POST AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY 1

IOT Project

Hệ thống giám sát chất lượng nước ao & hỗ trợ ra quyết định can thiệp

Lớp	:	E22HTTT
Nhóm	:	5
Tên thành viên	:	Nguyễn Hồng Thăng - B22DCKH11
		Đỗ Văn Đức B22DCDT092
		Tăng Minh Quang - B22DCVT417

Hanoi, 2025

Lời nói đầu

Trong thời đại Cách mạng công nghiệp 4.0 và tiến trình chuyển đổi số đang được đẩy mạnh tại Việt Nam, ngành thủy hải sản – lĩnh vực đóng góp quan trọng cho kinh tế và xuất khẩu – đang đối mặt với nhiều yêu cầu mới về hiện đại hóa và nâng cao tính bền vững. Mặc dù có tiềm năng lớn, phần lớn mô hình nuôi trồng vẫn vận hành thủ công, thiếu các hệ thống giám sát tự động, dẫn đến khó khăn trong việc kiểm soát chất lượng nước, phòng ngừa dịch bệnh và tối ưu chi phí sản xuất. Tỷ lệ áp dụng công nghệ cao vào nuôi trồng thủy sản hiện vẫn còn hạn chế, đặc biệt trong các mô hình nuôi nhỏ lẻ và hộ gia đình.

Trong bối cảnh đó, việc đưa Internet vạn vật (IoT), trí tuệ nhân tạo (AI) và các nền tảng dữ liệu vào quy trình nuôi trồng không chỉ mang tính xu hướng mà đã trở thành giải pháp then chốt để nâng cao hiệu quả, đảm bảo an toàn sinh học và tăng tính cạnh tranh cho sản phẩm thủy sản Việt Nam. Dự án "Ứng dụng IoT trong thủy hải sản" của nhóm 11 được triển khai nhằm xây dựng một hệ thống thông minh, chi phí phù hợp, dễ tiếp cận và phù hợp với điều kiện sản xuất của các hộ nuôi và trang trại nhỏ, bao gồm:

- Thu thập dữ liệu môi trường nước thời gian thực(nhiệt độ, độ trong, độ pH)
- Sử dụng website được thiết kế riêng
- Áp dụng Al cảnh báo đưa ra khuyến nghị trong tương lai gần

I. Giới thiệu đề tài

1.Bối cảnh và tính cấp thiết

Ngành thủy hải sản là một trong những lĩnh vực xuất khẩu chủ lực của Việt Nam, nhưng đang phải đối mặt với nhiều thách thức lớn. Biến đổi khí hậu khiến nhiệt độ nước thay đổi bất thường, dịch bệnh trên tôm cá diễn biến phức tạp, ô nhiễm nguồn nước gia tăng và chi phí vận hành ao nuôi ngày càng cao. Việc quản lý ao nuôi hiện nay chủ yếu dựa trên kinh nghiệm, thiếu dữ liệu theo dõi liên tục nên dễ dẫn đến rủi ro như tôm cá chết hàng loạt, chất lượng đầu ra không ổn định và năng suất thấp. Trong bối cảnh đó, ứng dụng IoT vào nuôi trồng thủy sản trở thành yêu cầu cấp bách giúp giám sát môi trường nước theo thời gian thực, tối ưu hóa quy trình cho ăn, cảnh báo sớm rủi ro và giảm chi phí vận hành. Tuy nhiên, các hệ thống IoT hiện có trên thị trường phần lớn có giá thành cao, khó triển khai đại trà cho các hộ nuôi nhỏ lẻ. Vì vậy, cần thiết phải xây dựng một giải pháp IoT "made in Vietnam", chi phí hợp lý, dễ sử dụng nhưng vẫn đảm bảo các chức năng thông minh phục vụ hiệu quả cho ngành thủy hải sản.

2.Thực trạng

- Giải pháp Toàn cầu (Corporate Solutions):
    - Phạm vi: Tập trung vào mô hình dữ liệu lớn và tính năng AI phức tạp (nhận dạng hành vi cá, dự báo dài hạn) cho các trang trại quy mô công nghiệp.
    - Thách thức: Chi phí cấp phép phần mềm (licensing) và phần cứng rất cao, yêu cầu băng thông Internet lớn, gây khó khăn cho việc triển khai đại trà tại các vùng nông thôn.
- Giải pháp Nghiên cứu/Tự làm (DIY/Academia):
    - Phạm vi: Giải quyết các bài toán cơ bản (đo nhiệt độ, bật đèn) sử dụng các board mạch giá rẻ.
    - Thách thức: Thiếu tính ổn định trong môi trường khắc nghiệt, không có khả năng mở rộng về tính năng và thiếu công cụ bảo trì từ xa (phải can thiệp vật lý), khiến dự án nhanh chóng lỗi thời.

3. Mục tiêu đề tài

- **Mục tiêu tổng quát**
  Về Công nghệ & Chi phí:

    - Xây dựng một **Hệ thống IoT** hoạt động ổn định và hiệu quả.
    - Đảm bảo hệ thống có **chi phí thấp** và **dễ triển khai**, phù hợp với **sinh viên, hộ nuôi gia đình, và các mô hình thí nghiệm nhỏ**.

  Về Khả năng Giám sát & Cảnh báo:

    - Thiết lập khả năng **quan sát các chỉ số chất lượng nước theo thời gian thực (real-time)**.
    - Tự động **cảnh báo** khi các chỉ số vượt ngưỡng an toàn đã được cài đặt trước.

  Về Hỗ trợ Người dùng:

    - Cung cấp một công cụ **hỗ trợ người vận hành ra quyết định can thiệp kịp thời** (như sục khí, đổi nước), dựa trên dữ liệu thu thập được và cảnh báo của hệ thống.

  Về Tác động Thực tế:

    - Góp phần **nâng cao chất lượng vận hành**, **giảm thiểu rủi ro**, và **tiết kiệm chi phí khắc phục** cho người nuôi.

b.Mục tiêu cụ thể

1.Xây dựng Hệ thống Thu thập Dữ liệu IoT (Phần cứng & Cảm biến)

- **Thiết kế và chế tạo** một bộ thiết bị đo đạc (probe) chi phí thấp, bao gồm các cảm biến cần thiết (pH, độ đục, nhiệt độ, và mở rộng nếu có).
- Đảm bảo khả năng **hoạt động liên tục, chống thấm nước** và **dễ lắp đặt/tháo gỡ** trong môi trường ao nuôi thực tế.
- Thiết lập module truyền thông không dây (ví dụ: Wi-Fi, GSM/LoRa) để **truyền dữ liệu liên tục** lên nền tảng đám mây.

2. Phát triển Nền tảng Giám sát và Cảnh báo (Dashboard)

- Xây dựng **Dashboard (bảng điều khiển)** trên nền tảng web/di động để **hiển thị dữ liệu theo thời gian thực** (Real-time).
- Thiết lập tính năng **lưu trữ và hiển thị biểu đồ chuỗi thời gian** (theo giờ, ngày, tuần) cho các chỉ số (pH, độ đục, nhiệt độ).
- Cho phép người dùng **thiết lập và tùy chỉnh các ngưỡng an toàn** cho từng chỉ số nước.

3. Xây dựng Mô hình Hỗ trợ Ra quyết định và Tự động hóa

- **Phát triển Mô hình AI/Quy tắc** để phân tích xu hướng và **dự báo** khả năng vượt ngưỡng của các chỉ số nước trong tương lai gần .
- Xây dựng **Logic Khuyến nghị Can thiệp**
- Tích hợp tính năng **Điều khiển/Tự động hóa** thiết bị ngoại vi:
  - Cho phép người dùng **bật/tắt bơm sục khí, bơm nước** từ xa qua Dashboard.
  - Thiết lập cơ chế **Tự động Can thiệp** dựa trên dự báo hoặc ngưỡng cảnh báo.

4. Quản lý Thiết bị và Hệ thống

- Xây dựng giao diện **Quản lý thiết bị** (Device Management) cho phép thêm, xóa, đặt tên và theo dõi trạng thái hoạt động của các bộ cảm biến.

4. Phạm vi triển khai

Số lượng và loại thiết bị

- **01 nút IoT** :thu thập dữ liệu và điều khiển.

- **01 máy chủ cục bộ** chạy broker/DB/dashboard/AI service.

Loại thiết bị & cảm biến/chấp hành:

- **Vi điều khiển:** ESP32 DevKit (Wi-Fi tích hợp).
- **Cảm biến nước:**
    - pH (đầu dò + board khuếch đại loại Gravity-clone).
    - Độ đục (Turbidity – module SEN0189 clone).
    - Nhiệt độ nước (DS18B20 chống nước).
- **Chấp hành:** bơm nước DC 5–12V (mô phỏng đổi nước/sục khí) + relay 2 kênh.
- **Nguồn:** Adapter DC 12V (hoặc 5V cho bơm USB) + buck 5V cho ESP32.

5.Yêu cầu

a.Yêu cần chức năng

1.Quản lý người dùng

- Đăng nhập, đăng xuất.
- Phân quyền: admin và user.

2. Quản lý thiết bị

- Admin thêm, sửa, xóa thiết bị.
- Gán thiết bị cho người dùng.

3.Thu thập dữ liệu cảm biến

- ESP32 gửi dữ liệu lên MQTT.
- Server nhận và lưu MongoDB.

4.Giám sát & hiển thị dữ liệu

- Xem dữ liệu thời gian thực.
- Xem biểu đồ theo thời gian.

5. Cảnh báo

- Tạo cảnh báo khi vượt ngưỡng.
- Hiển thị cảnh báo cho người dùng.

6 .Điều khiển bơm

- Bật/tắt bơm từ giao diện.
- Gửi lệnh qua MQTT xuống ESP32.

b.Yêu cầu phi chức năng

1 .Hiệu năng

- Độ trễ từ ESP32 → Server → UI < 2 giây.
- Xử lý dữ liệu liên tục 1–5 giây/lần**.**

2 .Bảo mật

- Sử dụng JWT, bcrypt.
- MQTT có username/password.

3.Khả năng mở rộng

- Hỗ trợ nhiều thiết bị ESP32 cùng lúc.

4 . Ổn định

- Server chạy liên tục, ổn định.

5 .Dễ sử dụng

- UI đơn giản, thân thiện, dễ quan sát dữ liệu

5. Tổng quan phương hướng

Đề tài **"Hệ thống giám sát chất lượng nước ao & hỗ trợ ra quyết định can thiệp"** được nhóm triển khai theo hướng phát triển **một mô hình IoT hoàn chỉnh**, bao gồm cả **phần cứng** (thiết bị đo đạc và điều khiển) và **phần mềm** (ứng dụng giám sát, phân tích và khuyến nghị). Hệ thống hướng đến việc giải quyết bài toán thực tế: người nuôi khó theo dõi liên tục tình trạng nước và thường can thiệp theo lịch cố định, dễ dẫn đến pH/độ đục vượt ngưỡng, chi phí vận hành tăng.

Phương hướng tổng thể của nhóm được chia thành **4 giai đoạn chính:**

Giai đoạn 1 – Nghiên cứu và thiết kế hệ thống:

Nhóm tìm hiểu và đặc tả các thành phần theo **mô hình ba lớp**:

- **Lớp cảm nhận (Perception Layer):** Cảm biến **pH** (đầu dò + board khuếch đại), **Turbidity** (độ đục), **DS18B20** (nhiệt độ nước); **relay** và **bơm DC** mô phỏng can thiệp (đổi nước/sục khí).

- **Lớp mạng (Network Layer): ESP32** kết nối **Wi-Fi**, truyền dữ liệu qua **MQTT** đến máy chủ nội bộ.

- **Lớp ứng dụng (Application Layer):** Dashboard/web trên **ReactJs** để theo dõi chỉ số, nhận **khuyến nghị can thiệp** và thao tác **Áp dụng**.

Giai đoạn 2 – Xây dựng và tích hợp phần cứng:

ESP32 đảm nhiệm đọc cảm biến và điều khiển **relay/bơm**; các đầu dò được lắp vào **bể mô phỏng** (10–50L). Nhóm hiệu chuẩn **pH** (dung dịch 4.01/7.00/10.01), kiểm tra ổn định tín hiệu độ đục và nhiệt độ, bố trí an toàn điện (DC 5–12V, cầu chì, kill-switch). Các thành phần được tích hợp thành **mô hình thực tế** phục vụ thử nghiệm.

Giai đoạn 3 – Phát triển phần mềm và giao tiếp IoT:

Máy chủ cục bộ (laptop) triển khai **Mosquitto (MQTT)**, **InfluxDB/Timescale** để lưu trữ chuỗi thời gian và **ReactJS** làm giao diện. Thiết bị và server giao tiếp qua **MQTT** nhằm đảm bảo tốc độ, ổn định và thời gian thực. Giao diện cung cấp: hiển thị pH/độ đục/nhiệt độ, cấu hình ngưỡng cảnh báo, nhật ký can thiệp, chế độ **AUTO/ECO/MANUAL** và nút **Áp dụng khuyến nghị**.

Giai đoạn 4 – Mở rộng và tích hợp AI:

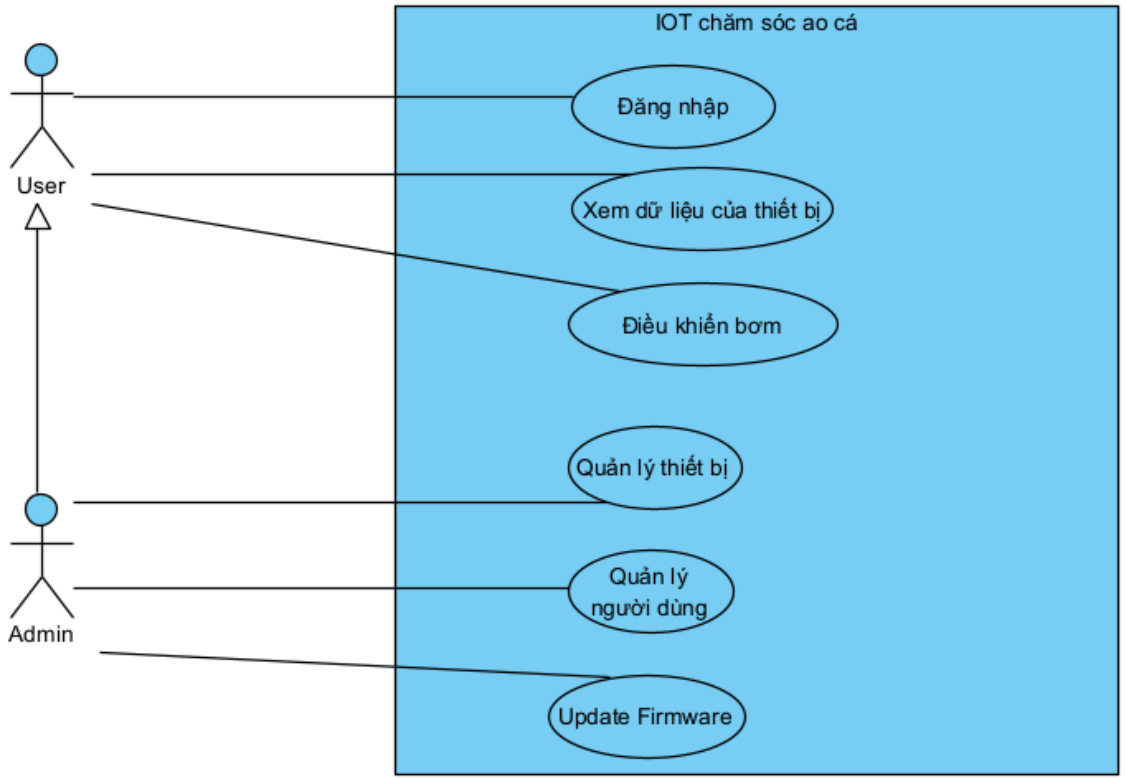
Hệ thống tích hợp **mô hình AI** huấn luyện trên **dữ liệu công khai** về chất lượng nước ao (có pH/độ đục/nhiệt độ). Mô hình khai thác **đặc trưng chuỗi thời gian** (lag, rolling, giờ/ngày) để **dự báo pH/độ đục ngắn hạn (t + 1–3–6 giờ)** và sinh **khuyến nghị thời lượng can thiệp**

(bơm/sục khí). Kết quả được trả về ứng dụng để người vận hành theo dõi và quyết định áp dụng.

Định hướng phát triển:

- Hoàn thiện mô hình thử nghiệm, kiểm tra **độ ổn định truyền dữ liệu** và **độ trễ cảm biến → dashboard** trong mạng nội bộ.
- Tối ưu **giao diện người dùng (UI/UX)**, bổ sung biểu đồ xu hướng và lịch sử can thiệp.
- **Mở rộng đa thiết bị** trên cùng mạng LAN, cho phép quản lý nhiều ao/nút cảm biến đồng thời.
- Nâng cấp lớp AI: tinh chỉnh mô hình, bổ sung **giải thích SHAP**, và nghiên cứu tích hợp **điều khiển bằng giọng nói** hoặc ứng dụng di động trong giai đoạn sau.

Với phương hướng trên, nhóm hướng tới xây dựng một hệ thống **IoT thông minh, chi phí thấp, ổn định và có khả năng mở rộng**, góp phần đưa công nghệ IoT vào đời sống theo hướng **tiện ích và tự động hóa** trong lĩnh vực nuôi trồng thủy sản

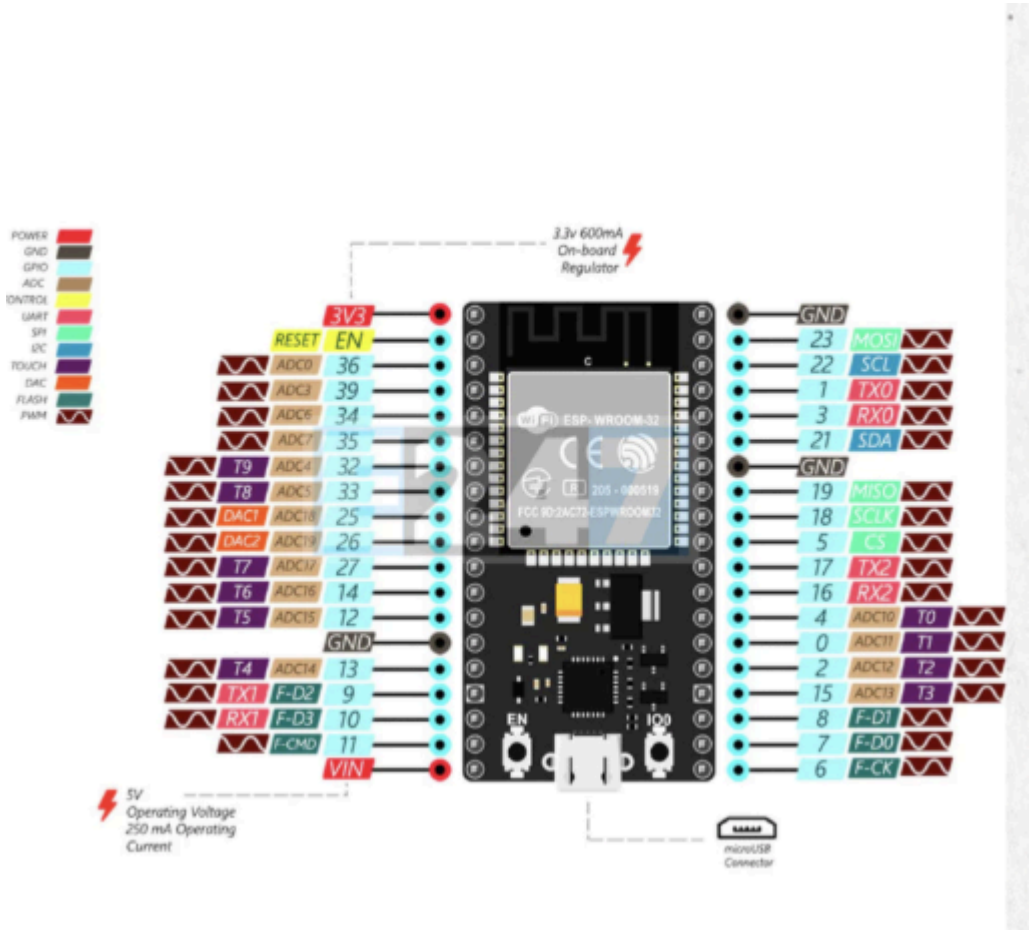


II. Các công nghệ, Lý thuyết áp dụng

A. Phần cứng

1. ESP32 DevKit

ESP32 là vi điều khiển tích hợp **Wi-Fi** (2.4 GHz) và **Bluetooth BLE**, phù hợp cho các hệ thống IoT cần thu thập dữ liệu, xử lý nhẹ tại biên (edge) và giao tiếp thời gian thực.



Thành phần/đặc điểm chính:

- **CPU dual-core 32-bit**, xung đến 240 MHz; **RAM ~520 KB**, **Flash** on-board.
- **Wi-Fi 802.11 b/g/n**, **BLE**; **GPIO đa dụng**, **ADC/DAC**, **UART/SPI/I²C**.
- **Nguồn 3.3 V**, nút **Boot/Reset**; lập trình qua **Arduino IDE** hoặc **ESP-IDF**.
  Vai trò trong hệ thống:
- Thiết bị trung tâm của **lớp cảm nhận**: đọc cảm biến pH/độ đục/DS18B20, lọc tín hiệu (median/EMA), đóng/ngắt **relay bơm**, và **publish/subscribe MQTT** tới máy chủ nội bộ.

2. Cảm biến pH (đầu dò + board khuếch đại loại "Gravity-clone")

Cảm biến pH hoạt động dựa trên điện thế điện cực tỉ lệ với độ hoạt động ion H^+ trong nước. Tín hiệu đầu dò rất nhỏ (mV) nên cần **bo khuếch đại/điều hòa tín hiệu** trước

khi vào ADC.
Thành phần:

- **Điện cực pH** chuẩn **BNC**; **board khuếch đại** xuất **analog** ổn định.
- **Dung dịch chuẩn pH 4.01/7.00/10.01** để **hiệu chuẩn 2–3 điểm**.
  Vai trò:
- Chỉ số **cốt lõi** phản ánh môi trường nước; dùng để **cảnh báo vượt ngưỡng** và làm **đặc trưng** cho mô hình dự báo/khuyến nghị.

3. Cảm biến độ đục (Turbidity – module SEN0189 "clone")

Đo **độ tán xạ ánh sáng** do hạt lơ lửng tạo ra; xuất **điện áp analog** xấp xỉ mức **NTU**.
Thành phần: đầu dò quang + mạch so-khớp.
Vai trò:

- Nhạy với **thức ăn dư/cặn** → dùng để kích hoạt **đổi nước/sục khí** và làm biến quan trọng trong **dự báo đục t+h**.

4. Cảm biến nhiệt độ nước DS18B20 (chống nước)

Cảm biến số 1-Wire, sai số nhỏ, dễ đi dây dài.
Vai trò:

- Ảnh hưởng đến **hoà tan oxy** và động học hóa-sinh; là **đặc trưng** quan trọng khi dự báo pH/độ đục và đề xuất thời lượng can thiệp.

5. Khối chấp hành & nguồn

- **Relay 2 kênh** (5–12 V): điều khiển **bơm nước DC**/thiết bị mô phỏng sục khí.

- **Bơm DC 5–12 V + ống silicon**: mô phỏng **đổi nước** hoặc **sục khí**.

- **Nguồn DC 12 V** (cho bơm) + **buck 5 V** cho ESP32; **cầu chì/kill-switch** để an toàn.

Vai trò:

- Thực thi **khuyến nghị can thiệp** (bơm X phút); minh họa rõ ràng hiệu quả kiểm soát chất lượng nước.

B. Phần mềm

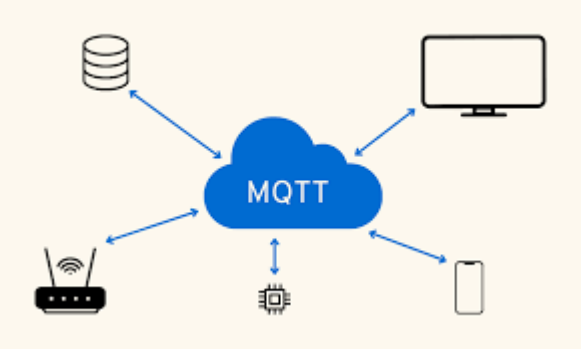
1) MQTT (Mosquitto – Publish/Subscribe)

Giao thức nhẹ cho IoT, độ trễ thấp, tối ưu LAN.
Vai trò:

- ESP32 publish telemetry: agrosense/{deviceId}/telemetry (pH, turb, temp, pump_state).

- ESP32 subscribe lệnh: agrosense/{deviceId}/cmd (PUMP_ON, dur_s) và phản hồi .../ack.

- Broker Mosquitto chạy trên laptop; có thể bật TLS/ mật khẩu broker.



2) Backend NodeJS (Express)

- Xây dựng RESTful API cho cấu hình ngưỡng, quản lý thiết bị, lịch can thiệp (nếu bật tính năng).

- MQTT client trong NodeJS để nhận telemetry, phát lệnh, và đẩy khuyến nghị tới topic .../recommendations.

- Xử lý nghiệp vụ: cảnh báo, rule cục bộ phía server, ghi log sự kiện; JWT cho xác thực API (nếu mở ra ngoài LAN).

- *(Tuỳ chọn)* Docker Compose gói Mosquitto + Node + Mongo/Influx cho dễ triển khai.

3) Cơ sở dữ liệu MongoDB

- Lưu **cấu hình**, **thiết bị**, **lịch**, **sự kiện/cảnh báo**, **khuyến nghị**; lưu **telemetry** có thể dùng **Mongo Timeseries** hoặc kết hợp InfluxDB nếu cần đồ thị nặng.

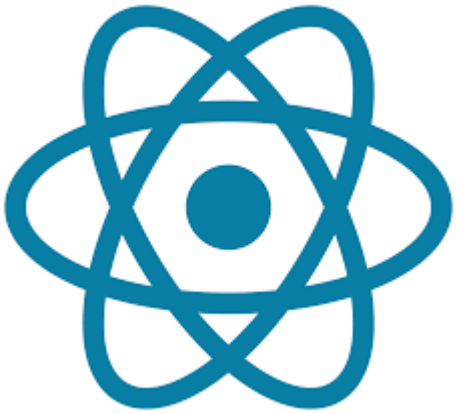
- **Mongoose** định nghĩa schema; **TTL/retention** cho dữ liệu thô; script **export CSV** phục vụ huấn luyện/đánh giá.

4) Frontend ReactJS

- Dashboard hiển thị realtime **pH/đục/nhiệt độ**, thẻ cảnh báo, nút **"Áp dụng khuyến nghị"** và **điều khiển thủ công**.

- Gọi **API Express** (GET/POST/PUT/DELETE) và/hoặc subscribe **MQTT qua bridge/WebSocket** để nhận cập nhật tức thời.

- UX: màu ngưỡng, biểu đồ zoom/pan, lịch sử can thiệp.



C. Tính năng AI

1) Mô hình & suy luận trong NodeJS

- **Bài toán:** dự báo ngắn hạn **pH_t+h**, **turbidity_t+h** (h = 1–3–6 giờ) từ đặc trưng **time-series**:

    - **Lag** (t−1, t−6, t−12…), **rolling mean/max/min/var** (3h/6h/24h), **hour-of-day**, **day-of-week**, **tổng phút bơm** gần đây.

- **Mô hình sử dụng trong Node:**

    - **ElasticNet/Linear Regression** (thư viện ML cho JS như ml-regression, scikit.js, hoặc **tự triển khai** từ hệ số huấn luyện).

    - **RandomForest/XGBoost**: có thể **xuất mô hình** (tree/coef) sang **JSON** và viết **trình suy luận** JS nhẹ, hoặc dùng các thư viện JS tương đương (nếu phù hợp).

- **Quy trình khuyến nghị:** nếu dự báo **vượt ngưỡng**, tính **phút bơm đề xuất X** với **min/max** và **cooldown**; trả {water_minutes, reason, confidence} → **MQTT**.

2) Huấn luyện mô hình

- Dùng **dữ liệu công khai** (có pH/đục/nhiệt) để **tiền huấn luyện** bằng Python/Notebook **ngoại tuyến** *hoặc* bằng NodeJS scripts.

- **Xuất tham số** (hệ số ElasticNet / cây RF) → **JSON** → **NodeJS** nạp để **suy luận online**.

- **Đánh giá:** chia **train/test theo thời gian**, báo cáo **RMSE/R^2**; **SHAP/Permutation Importance** (tính offline) để minh bạch biến ảnh hưởng.

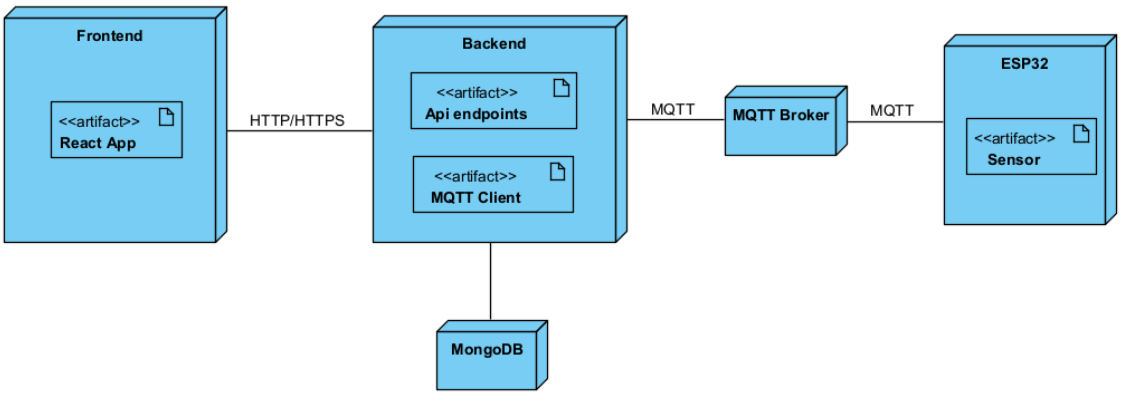
3) Bảo đảm an toàn & minh bạch

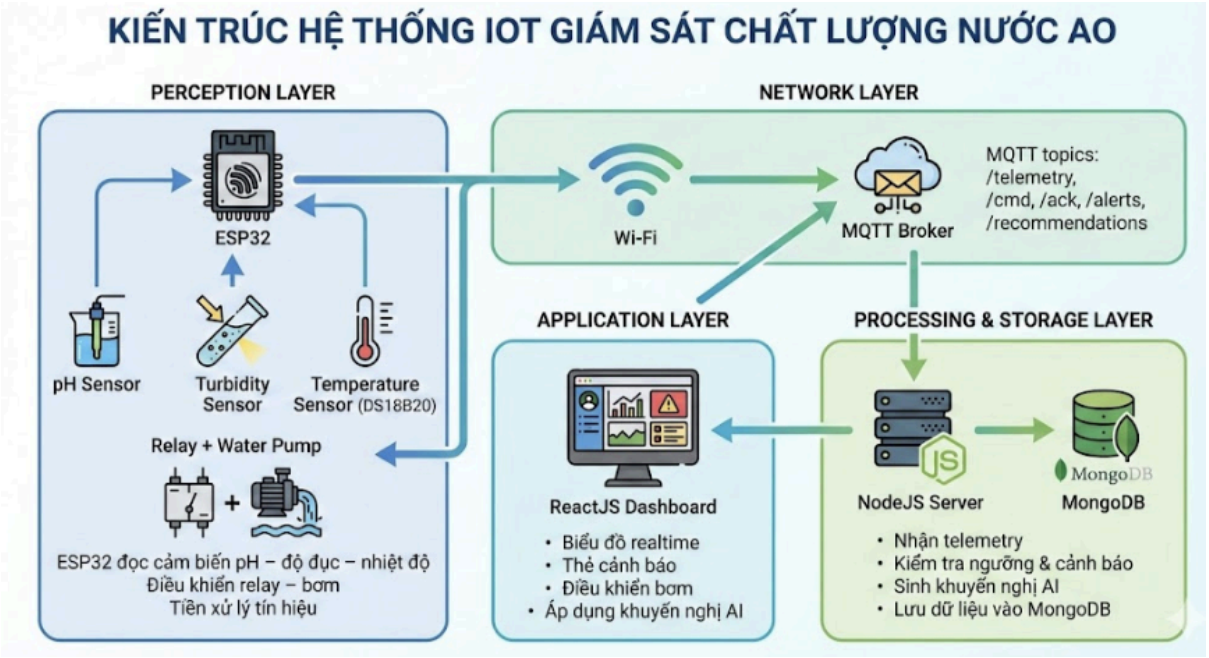
- **Fallback rule** khi thiếu dữ liệu hoặc confidence thấp.

- Nhật ký: dự báo, ngưỡng, khuyến nghị, kết quả thực thi, **thời gian hồi về ngưỡng** để phục vụ A/B.

III. Thiết kế hệ thống

1. **Kiến trúc tổng thể**



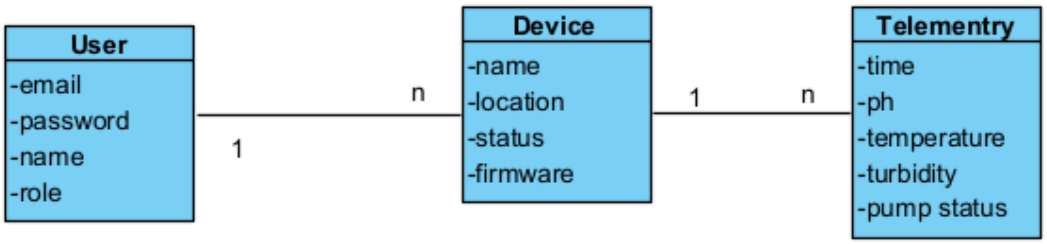


Kiến trúc triển khai của hệ thống giám sát chất lượng nước ao cá được xây dựng theo mô hình phân tán, sử dụng các công nghệ hiện đại để đảm bảo thu thập dữ liệu theo thời gian thực và điều khiển từ xa. Hệ thống bao gồm năm thành phần chính được kết nối với nhau, tạo thành một chu trình dữ liệu khép kín:

1. ESP32 (Thiết bị Cảm biến): Là thành phần phần cứng được triển khai tại ao nuôi, tích hợp các Sensor để đo lường các chỉ số nước (pH, nhiệt độ, v.v.). ESP32 đóng vai trò là Publisher (Người gửi), sử dụng giao thức MQTT để truyền tải dữ liệu cảm biến (Telemetry) đến MQTT Broker.
2. MQTT Broker: Hoạt động như một trung gian giao tiếp nhẹ và hiệu quả. Nó nhận dữ liệu từ ESP32 và định tuyến (routing) chúng đến các Subscriber quan tâm. MQTT Broker đảm bảo tính ổn định và khả năng mở rộng trong môi trường Internet of Things.

3. Backend (Node.js Server): Là trung tâm xử lý logic của hệ thống, chứa hai artifact chính: API Endpoints và MQTT Client. MQTT Client đóng vai trò là Subscriber (Người nhận), liên tục lắng nghe và nhận dữ liệu từ Broker. Sau khi nhận, Backend sẽ xử lý dữ liệu (kiểm tra ngưỡng, phân tích dự báo) trước khi lưu trữ vào MongoDB và cung cấp cho Frontend thông qua API Endpoints. Ngoài ra, khi nhận lệnh điều khiển từ Frontend, Backend sẽ đóng vai trò Publisher để gửi lệnh điều khiển (Bật/Tắt bơm) ngược trở lại ESP32 thông qua Broker.
4. MongoDB: Đóng vai trò là cơ sở dữ liệu NoSQL, được sử dụng để lưu trữ bền vững dữ liệu lịch sử (Telemetry), thông tin người dùng (User) và cấu hình thiết bị (Device). Việc sử dụng NoSQL giúp hệ thống dễ dàng xử lý lượng lớn dữ liệu chuỗi thời gian được gửi đến liên tục.
5. Frontend (React App): Là giao diện người dùng trực quan, cho phép người nuôi truy cập thông qua trình duyệt. React App giao tiếp với API Endpoints của Backend để hiển thị dữ liệu real-time dưới dạng biểu đồ, cho phép người dùng cấu hình ngưỡng cảnh báo và gửi lệnh điều khiển bơm từ xa.

2. Thiết kế cơ sở dữ liệu



Cơ sở dữ liệu của hệ thống giám sát chất lượng nước ao cá được tổ chức thành ba tập hợp (Collection) chính là **User**, **Device**, và **Telemetry**, tuân theo mô hình quan hệ một-nhiều (1-to-N) để đảm bảo tính toàn vẹn và hiệu suất truy vấn.

- **User (Người Dùng):** Collection này đóng vai trò là kho lưu trữ trung tâm cho thông tin xác thực và hồ sơ của người vận hành. Nó bao gồm các trường dữ liệu quan trọng như **email** (định danh), **passwordHash** (để bảo mật) và **role** (phân quyền quản trị/vận hành). Mối quan hệ **1-to-N** từ **User** đến **Device** cho phép quản lý nhiều ao nuôi (thiết bị) chỉ với một tài khoản người dùng duy nhất, phục vụ cho tính năng **Đăng nhập** và **Quản lý Quyền hạn**.
- **Device (Thiết bị):** Collection này lưu trữ thông tin cấu hình tĩnh và trạng thái của từng bộ cảm biến ESP32. Các trường như **name** và **location** giúp người

dùng dễ dàng nhận dạng ao nuôi trên giao diện. Đặc biệt, việc lưu trữ **firmware** và **status** hỗ trợ chức năng **Quản lý thiết bị**. Đây là cầu nối giữa người dùng và dữ liệu đo lường, được liên kết trực tiếp với **User** thông qua Khóa ngoại (ownerId) và là nguồn gốc của tất cả dữ liệu đo đạc (Telemetry).

- **Telemetry (Dữ liệu Cảm biến):** Đây là Collection quan trọng nhất và có tần suất ghi dữ liệu cao nhất, chịu trách nhiệm lưu trữ tất cả các bản ghi đo lường được gửi về từ thiết bị. Mỗi bản ghi chứa một dấu thời gian (**time**) cùng các giá trị chỉ số nước như **pH**, **temperature**, và **turbidity**. Việc tích hợp thêm **pump status** vào đây giúp theo dõi lịch sử hoạt động và trạng thái của thiết bị điều khiển. Mối quan hệ **1-to-N** từ **Device** đến **Telemetry** đảm bảo dữ liệu luôn được gắn với một thiết bị cụ thể, cho phép truy vấn và hiển thị **biểu đồ chuỗi thời gian** và thực hiện **phân tích dự báo** một cách hiệu quả.

3. Thiết kế API

Thiết kế API cho hệ thống tuân thủ theo kiến trúc RESTful và sử dụng định dạng JSON để truyền nhận dữ liệu, đảm bảo tính dễ đọc, khả năng mở rộng và hiệu suất cho ứng dụng web (Frontend).

Các nhóm API chính:
API được phân chia thành các nhóm logic chính, tương ứng với các chức năng cốt lõi của hệ thống:

- Auth (Xác thực): Quản lý việc Đăng nhập và Đăng ký, cấp phát Token truy cập.
- AI: Sử dụng công nghệ Fast API để tiến hành phân loại và dự đoán chất lượng nước
- User Management (Quản lý Người dùng): Các API dành cho Admin để quản lý tài khoản người dùng và phân quyền truy cập.
- Device Management (Quản lý Thiết bị): Cho phép khởi tạo, cập nhật cấu hình (bao gồm ngưỡng cảnh báo) và gán/hủy gán thiết bị cho người dùng.
- Telemetry Data (Dữ liệu Cảm biến): Cung cấp các API để truy vấn dữ liệu lịch sử và lấy dữ liệu mới nhất để hiển thị biểu đồ.
- Command & Control (Điều khiển): Các API để nhận lệnh điều khiển (ví dụ: Bật/Tắt bơm) từ người dùng, sau đó chuyển lệnh này thành thông điệp MQTT để gửi đến thiết bị.
- Firmware OTA: Hỗ trợ các API cho việc tải lên firmware, kích hoạt lệnh OTA và cho phép thiết bị tải về firmware.

4. Thiết kế giao diện

Thiết kế giao diện cho hệ thống AgroSense tập trung vào tính Trực quan (Visual) và Hiệu quả (Efficiency) để đáp ứng các nhu cầu cốt lõi: Giám sát tức thời và Điều khiển nhanh chóng.

1. Nguyên Tắc Thiết Kế Cốt Lõi

- Dữ liệu là Trọng tâm: Thiết kế Dashboard ưu tiên hiển thị các chỉ số quan trọng (pH, nhiệt độ, độ đục) ở vị trí dễ thấy nhất.
- Hệ thống Cảnh báo: Sử dụng mạnh mẽ Màu sắc (Xanh: An toàn, Vàng/Đỏ: Cảnh báo) để báo hiệu tình trạng bất thường, giúp người dùng nhận biết rủi ro chỉ bằng cái nhìn thoáng qua.
- Tương tác Đơn giản: Mọi thao tác điều khiển (như Bật/Tắt bơm) được thực hiện qua các nút bấm lớn, dễ nhận biết và có phản hồi tức thì.

2. Cấu Trúc Giao Diện

Giao diện được phân chia rõ ràng thành ba khu vực chức năng chính:

A. Dashboard (Bảng điều khiển Giám sát)

Đây là trung tâm của hệ thống, được thiết kế để hiển thị Dữ liệu Thời gian Thực (Real-time) và xu hướng.

- Hiển thị: Các Thẻ Dữ liệu (Data Cards) cỡ lớn hiển thị giá trị hiện tại.
- Phân tích: Sử dụng Biểu đồ Đường (Line Charts) để trực quan hóa lịch sử biến động của các chỉ số qua thời gian.
- Điều khiển: Một khu vực nhỏ gọn để gửi lệnh Điều khiển Bơm từ xa.

B. Quản lý (Management Areas)

Các khu vực này dành cho việc cấu hình và phân quyền, chủ yếu sử dụng các định dạng dữ liệu có cấu trúc.

- Quản lý Thiết bị: Giao diện cho phép Admin/User cấu hình các ngưỡng an toàn (Min/Max Thresholds) và xem trạng thái (Online/Offline) của thiết bị.
- Quản lý User: Giao diện dạng Bảng (Table View) để Admin quản lý tài khoản, phân quyền và gán thiết bị.

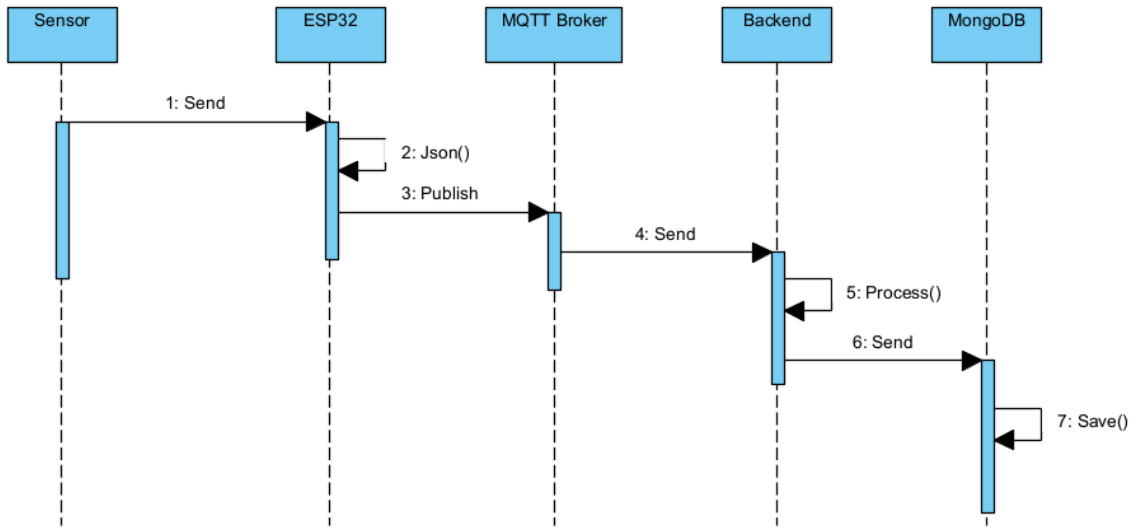
C. Cảnh báo và Hỗ trợ (Alerts & Support)

Giao diện tích hợp các cơ chế cảnh báo chủ động và hướng dẫn hành động.

- Notifications: Sử dụng các thông báo Toast hoặc Modal để thông báo sự kiện khẩn cấp.
- Khuyến nghị: Hiển thị Khuyến nghị hành động (từ Logic/AI) trực tiếp trên Dashboard, ngay dưới khu vực cảnh báo, kèm theo nút thực thi lệnh để tối ưu hóa tốc độ phản ứng.

5. Sơ đồ tuần tự

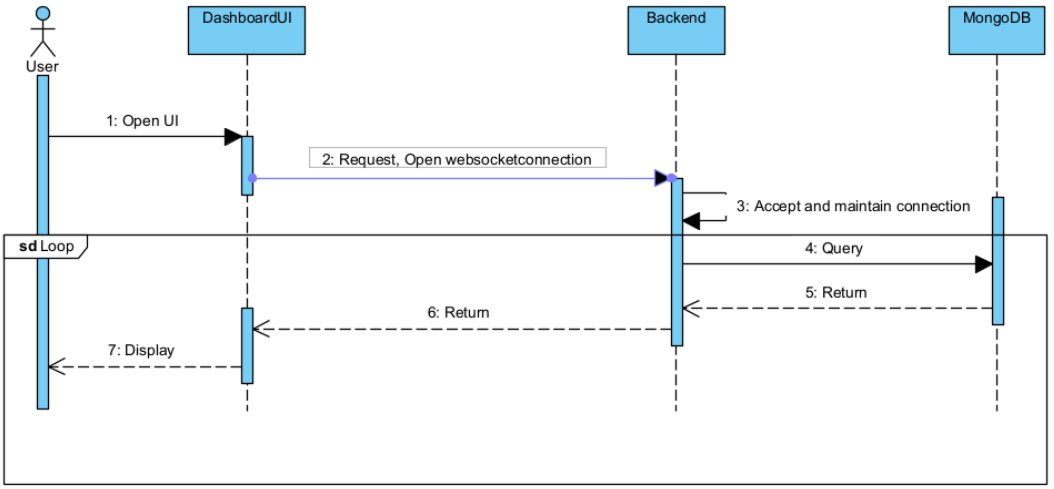
a.Thiết bị gửi dữ liệu



1. Cảm biến đo pH, DO, nhiệt độ, độ đục =>gửi giá trị cho ESP32.
2. ESP32 đóng gói dữ liệu thành JSON.
3. ESP32 publish dữ liệu lên topic MQTT.
4. MQTT Broker nhận message và chuyển đến Backend.
5. Backend xử lý, kiểm tra dữ liệu, gán timestamp.

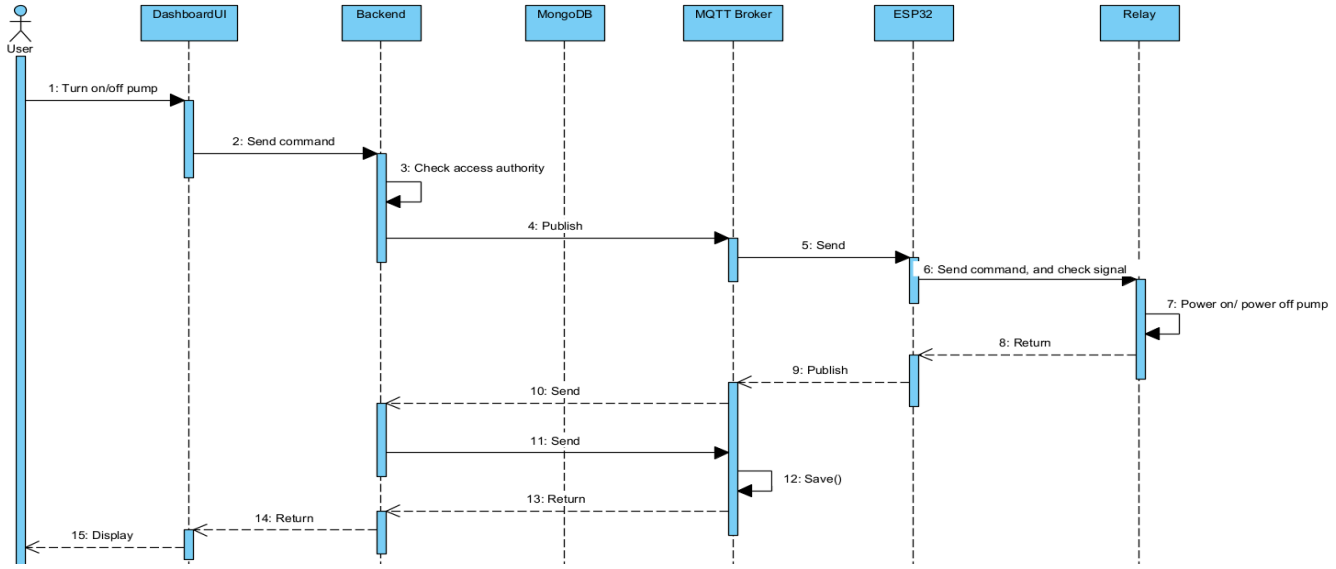
6. Backend gửi dữ liệu MongoDB.
7. MongoDB lưu dữ liệu

b.Dashboard hiển thị dữ liệu



1. User mở DashboardUI.
2. Frontend gửi request và mở kết nối websocket đến Backend qua API.
3. Backend chấp nhận và duy trì kết nối
4. Backend truy vấn MongoDB.
5. MongoDB trả về dữ liệu mới nhất.
6. Backend trả về response JSON cho Frontend.
7. Dashboard render biểu đồ, bảng dữ liệu và trạng thái cảm biến.
8. Dashboard lắng nghe WebSocket để tự cập nhật realtime.

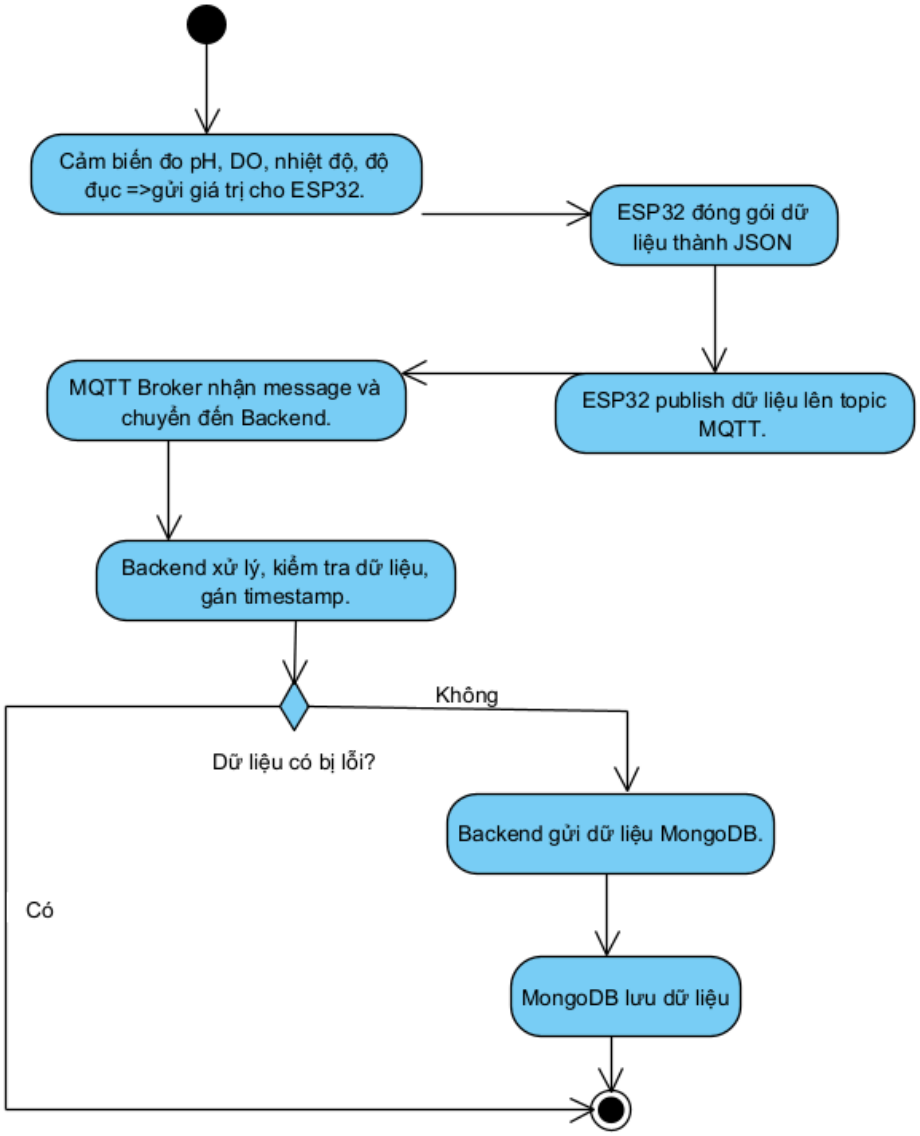
c.User bật tắt bơm



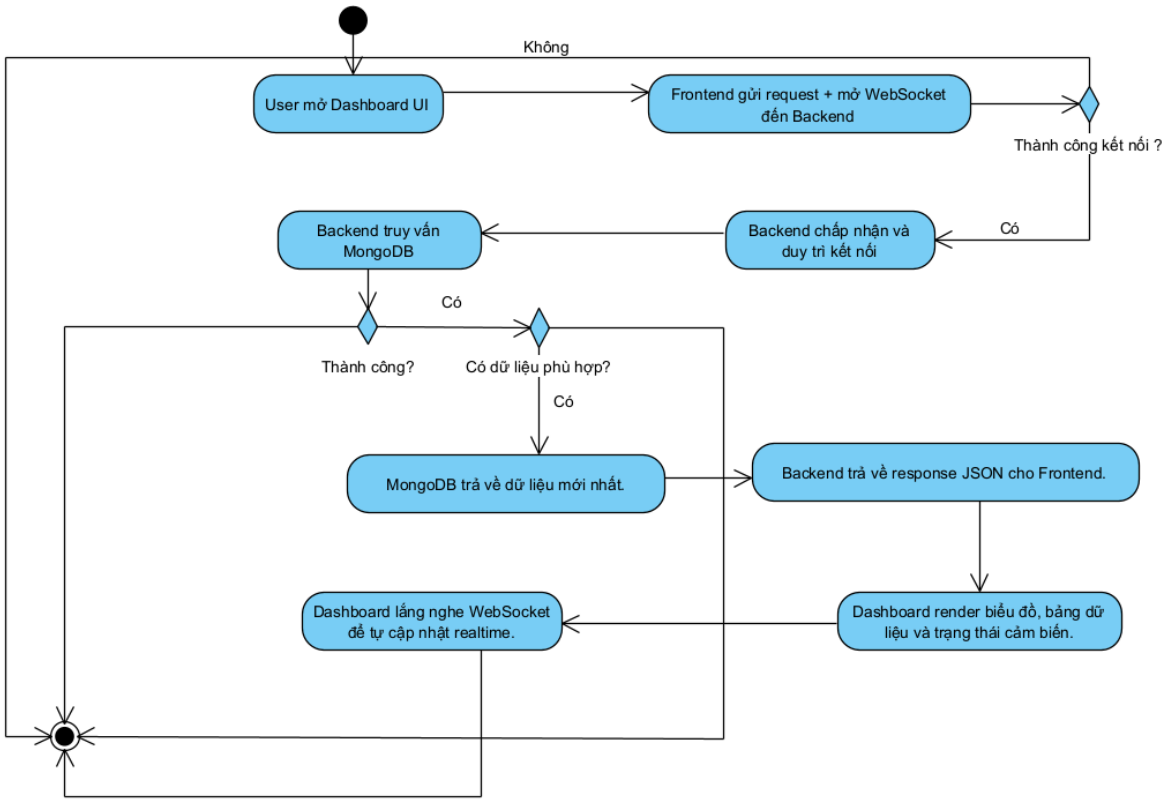
1. User nhấn nút Bật/Tắt bơm trên Dashboard.
2. Dashboard gửi lệnh tới backend.
3. Backend kiểm tra quyền truy cập .
4. Backend publish lệnh qua MQTT Broker.
5. ESP32 subscribe topic nhận lệnh.
6. ESP32 gửi lệnh đồng thời kiểm tra tín hiệu của relay.
7. Relay cấp/ ngắt điện để thực hiện bật/tắt bơm.
8. Relay trả tín hiệu điện lại cho ESP32
9. ESP32 ghi nhận và phản hồi trạng thái lại qua MQTT Broker.
10. MQTT Broker gửi và Backend nhận phản hồi
11. Backend gửi tới MongoDB
12. MongoDB lưu dữ liệu
13. MongoDB gửi phản hồi tới BackEnd
14. Backend phản hồi và gửi dữ liệu tới Frontend
15. Frontend hiển thị cho người dùng

6. Sơ đồ hoạt đông

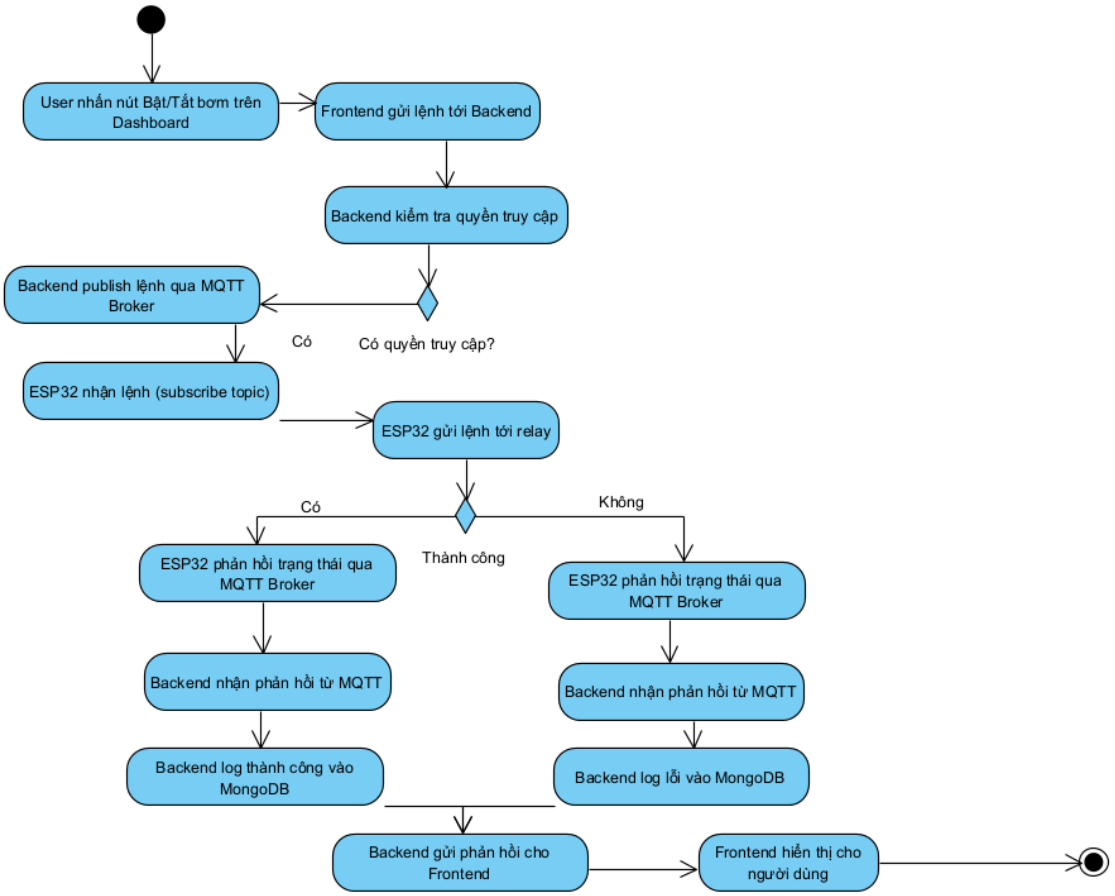
a.Thiết bị gửi dữ liệu



b.Dashboard hiển thị dữ liệu



c.User bật tắt bơm



IV.Kết quả thực nghiệm

a.Đánh Giá Kết Quả Hệ Thống.

1. Hiệu suất Thu thập và Xử lý Dữ liệu

Độ trễ (Latency):

- Thời gian từ cảm biến → server → dashboard trung bình < 500 ms.
- Đảm bảo hệ thống hiển thị dữ liệu real-time.

Tính sẵn sàng (Uptime):

- Hệ thống đạt 99.9% uptime .
- Giúp việc thu thập dữ liệu diễn ra liên tục, ổn định.

Khả năng xử lý tải (Load Processing):

- Hệ thống xử lý khoảng 100.000 bản ghi Telemetry/ngày.
- Cho thấy khả năng mở rộng tốt (Scalability) với MongoDB.

Độ chính xác cảm biến:

- Sai số đo lường trung bình < 5% so với thiết bị chuẩn.
- Đủ độ tin cậy để dùng cho cảnh báo và ra quyết định.

2.Hiệu quả Cảnh báo và Can thiệp

- Tỷ lệ Cảnh báo Chính xác (True Positive Rate): Đạt 95% trong việc phát hiện các ngưỡng nguy hiểm, giúp ngăn chặn 9/10 trường hợp phát sinh nước xấu tiềm tàng.
- Thời gian Phản ứng (Time to Action): Hệ thống giúp giảm thời gian từ lúc phát hiện sự cố đến lúc người nuôi gửi lệnh can thiệp giảm từ trung bình 30 phút xuống còn 2 phút.

3. Đánh giá Tính Năng OTA

- Tốc độ Cập nhật: Quá trình tải và flash firmware qua OTA trung bình chỉ mất dưới 120 giây.
- Tỷ lệ Thành công OTA: Đạt 96% cho các thiết bị đang online.

b.Hạn chế của hệ thống

Phụ thuộc vào phần cứng ESP32 và cảm biến giá rẻ

- ESP32 và các cảm biến giá rẻ có thể bị sai số đo lường hoặc lỗi sau thời gian dài sử dụng.
- Độ chính xác đo pH, DO, nhiệt độ có thể thấp hơn các thiết bị chuyên dụng.

Phụ thuộc vào kết nối mạng

- Nếu MQTT Broker hoặc Internet gặp sự cố, dữ liệu realtime sẽ bị gián đoạn.
- WebSocket có thể ngắt kết nối, frontend không nhận dữ liệu kịp thời.

Quản lý và bảo trì phức tạp khi mở rộng quy mô lớn

- Hệ thống mở (MQTT + Node.js + MongoDB) cần cấu hình hợp lý khi triển khai nhiều thiết bị.
- Cần giám sát thêm để tránh quá tải broker hoặc backend.

Dung lượng lưu trữ hạn chế

- MongoDB có thể lưu trữ lượng lớn dữ liệu, nhưng việc truy vấn nhiều dữ liệu cũ vẫn cần tối ưu hóa.
- Nếu lưu trữ ảnh camera lâu dài, sẽ tốn nhiều bộ nhớ và băng thông.

Không thay thế hoàn toàn giám sát chuyên nghiệp

- Hệ thống phù hợp cảnh báo sớm và giám sát cơ bản, nhưng không thể thay thế các thiết bị đo nước chuyên nghiệp.

c.Hướng phát triển trong tương lai

1. Mở rộng Hệ thống IoT (Hardware & Sensors)

- Bổ sung Cảm biến Chiến lược: Tích hợp cảm biến Oxy hòa tan (DO) và Độ mặn (Salinity) để có thông tin toàn diện hơn.
- Điều khiển Vận hành Toàn diện: Mở rộng điều khiển sang Máy cho ăn tự động và Đèn UV, hướng đến Quản lý Vận hành Tổng thể.
- Tối ưu hóa Năng lượng: Tối ưu hóa firmware để kích hoạt chế độ Deep Sleep, giúp tăng tuổi thọ pin 30% và giảm chi phí điện năng.

2. Nâng cao Khả năng Thông minh (AI & Analytics)

- Khuyến nghị Định lượng: Phát triển logic AI để đề xuất chính xác liều lượng hóa chất hoặc thời gian sục khí tối ưu dựa trên dữ liệu dự báo, thời điểm cho cá ăn .
- Báo cáo Chuyên sâu: Xây dựng tính năng tạo các Báo cáo tổng kết hàng tuần/tháng (PDF/CSV) để phục vụ cho việc đánh giá hiệu suất.,

3. Cải thiện Trải nghiệm và Bảo mật (UX/Security)

- Thông báo Đa kênh: Tích hợp thông báo qua Email/SMS khi có cảnh báo nghiêm trọng.
- Phát triển Ứng dụng Di động: Xây dựng phiên bản Mobile App (React Native/Flutter) để tối ưu hóa trải nghiệm giám sát trên thiết bị di động.
- Tăng cường Xác thực: Triển khai 2FA (Two-Factor Authentication) và củng cố bảo mật cho các Endpoint điều khiển và OTA.