

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN 1**



**BÁO CÁO IOT VÀ ỨNG DỤNG**

**ĐỀ TÀI: HỆ THỐNG THÙNG RÁC  
IOT PHÂN QUYỀN VÀ CẢNH BÁO ĐẦY**

Giảng viên hướng dẫn : Kim Ngọc Bách

Nhóm thực hiện: 14

Nhóm học phần : 06

Thành viên thực hiện:

B22DCCN830 – Đinh Công Thịnh

B22DCCN842 – Nguyễn Như Thuật

B22DCCN758 – Nguyễn Anh Tuấn

B22DCCN013 – Đỗ Nhật Anh

*Hà Nội, 2025*

## Mục lục

I.	Giới thiệu đề tài.....	3
a.	Giới thiệu đề tài .....	3
b.	Mục tiêu Chi tiết của Đề tài.....	3
c.	Thu thập Yêu cầu từ Stakeholders.....	5
II.	Cơ sở lý thuyết .....	8
a.	Mô hình truyền thông Publish/Subscribe (Pub/Sub).....	8
b.	Giao thức MQTT (Message Queuing Telemetry Transport).....	8
c.	Nguyên lý hoạt động của cảm biến .....	9
d.	Phân quyền người dùng & xác thực .....	9
e.	Giao tiếp Client-Server & Realtime .....	9
f.	Cơ sở dữ liệu và logging .....	9
g.	Kiến trúc hệ thống IoT .....	9
h.	Nguyên lý hệ thống nhúng (Embedded Systems) .....	10
i.	Tự động hóa và IoT (Internet of Things).....	10
III.	Công nghệ sử dụng.....	10
IV.	Phân tích yêu cầu.....	11
a.	Yêu cầu chức năng .....	11
b.	Yêu cầu phi chức năng .....	16
V.	Phân tích ràng buộc .....	19
a.	Ràng buộc môi trường.....	19
b.	Ràng buộc pháp lý & bảo mật.....	19
c.	Ràng buộc tài nguyên thiết bị.....	19
d.	Ràng buộc hiệu năng .....	20
e.	Ràng buộc mở rộng .....	20
VI.	Mô hình hoá.....	20
a.	Ưu tiên hoá .....	20
b.	Mô hình .....	22
VII.	Phân chia công việc .....	23

## I. Giới thiệu đề tài

### a. Giới thiệu đề tài

Quản lý rác thải là một vấn đề trọng yếu trong phát triển đô thị. Phương pháp truyền thống thường dẫn đến việc thu gom không đúng lúc, gây ô nhiễm môi trường, tốn kém chi phí vận hành, và thiếu dữ liệu để tối ưu hóa quy trình. Đề tài đề xuất một giải pháp thông minh dựa trên IoT để hiện đại hóa quy trình này.

Hệ thống **Thùng rác IoT Phân quyền và Cảnh báo đầy** không chỉ tự động hóa hoạt động mở nắp mà còn tích hợp khả năng **giám sát mức đầy realtime** và **phân quyền người dùng** thông qua thẻ RFID. Điều này giúp:

- Tăng tính vệ sinh.
- Tối ưu hóa lịch trình thu gom dựa trên dữ liệu mức đầy thực tế.
- Cung cấp tính năng kiểm soát truy cập và ghi lại lịch sử hoạt động chi tiết.

### b. Mục tiêu Chi tiết của Đề tài

#### i. Hệ thống IoT nhằm giải quyết vấn đề gì?

Hệ thống giải quyết đồng thời ba mục tiêu cốt lõi trong quản lý rác thải đô thị, nhằm thay thế phương pháp thu gom truyền thống kém hiệu quả:

- **Giám sát (Monitoring):**
  - **Giám sát mức đầy theo thời gian thực (realtime)** của thùng rác bằng cảm biến siêu âm **HC-SR04**.
  - **Ghi lại lịch sử hoạt động chi tiết** (thời gian quét thẻ, mở nắp, cảnh báo).
- **Tự động hóa (Automation):**
  - **Tự động hóa quy trình mở/đóng nắp** bằng Servo **SG90** khi người dùng được xác thực hợp lệ bằng thẻ **RFID** hoặc được phát hiện bằng cảm biến **HC-SR04**.

- **Tự động cảnh báo** ( $\geq 80\%$  đầy) để kích hoạt quy trình thu gom.
- **Tối ưu hóa (Optimization):**
  - **Tối ưu hóa lịch trình thu gom** dựa trên dữ liệu mức đầy thực tế, thay vì lịch trình cố định.
  - **Tăng tính vệ sinh và giảm chi phí vận hành.**

ii. Phạm vi triển khai

Phạm vi của đề tài tập trung vào việc xây dựng một hệ thống **mô hình hóa và chứng minh tính khả thi (Proof of Concept - PoC)**:

- **Quy mô:** Hệ thống mẫu (Prototype) với **01 thiết bị Thùng rác IoT** hoàn chỉnh.
- **Môi trường:** Triển khai trong môi trường phòng thí nghiệm (Lab) hoặc khu vực thử nghiệm nội bộ, mô phỏng các điều kiện sử dụng thực tế tại khu dân cư hoặc khuôn viên trường.
- **Số lượng thiết bị:** Bao gồm **01 đơn vị ESP32, 01 RFID RC522, 01 Servo SG90, 01 HC-SR04** (cảm biến mức đầy), và hệ thống Backend/Frontend tập trung.
- **Phân quyền:** Bao gồm ít nhất 2 vai trò người dùng (**Admin** và **User**) theo mô hình **RBAC (Role-Based Access Control)**

iii. Đặt Tiêu chí Thành công (KPIs)

Các tiêu chí này là thước đo định lượng cho sự thành công của đề tài:

Tiêu chí	Mục tiêu Cụ thể	Chứng minh Kỹ thuật
Độ chính xác	Độ chính xác của phép đo mức đầy (HC-SR04).	Sai số đo mức đầy < 5% so với đo thủ công.
Độ trễ (Latency)	Thời gian phản hồi cho lệnh điều khiển mở nắp.	Thời gian phản hồi lệnh điều khiển từ xa < 500ms.
Độ tin cậy (Reliability)	Đảm bảo truyền nhận các thông điệp quan trọng.	Sử dụng QoS 1 cho các thông điệp Cảnh báo, Lệnh điều khiển, Log để đảm bảo không mất gói tin.
Bảo mật	Khả năng xác thực người dùng và thiết bị.	Xác thực thẻ RFID qua Backend Server. Xác thực (username/password) và Phân quyền Topic cho MQTT Broker.

c. Thu thập Yêu cầu từ Stakeholders

Phần này phân tích các yêu cầu chức năng và phi chức năng cần phải đáp ứng, dựa trên vai trò của các bên liên quan đối với hệ thống thùng rác IoT.

i. Người dùng cuối (End-User: Người đổ rác)

Yêu cầu	Mô tả Chi tiết (Mong muốn)	Đã đáp ứng (Tính năng)
Thao tác đơn giản	Khả năng đổ rác nhanh chóng, không chạm.	Mở nắp tự động khi quét thẻ hợp lệ hoặc phát hiện người dùng (HC-SR04).
Phản hồi	Biết được thao tác quét thẻ thành công hay thất bại.	Hệ thống cần cung cấp phản hồi hình ảnh/âm thanh (LED/Buzzer) sau khi quét thẻ.
Vệ sinh	Giảm thiểu tiếp xúc với thùng rác.	Tăng tính vệ sinh nhờ cơ chế mở nắp tự động.

ii. Quản lý/Doanh nghiệp (Admin, Đơn vị Quản lý Rác)

Yêu cầu	Mô tả Chi tiết (Hiệu quả, Báo cáo)	Đã đáp ứng (Tính năng)
Cảnh báo tức thì	Cần nhận được thông báo khi thùng rác gần đầy để lên kế hoạch thu gom.	Cảnh báo mức đầy ( $\geq 80\%$ ) được gửi qua hệ thống/Dashboard. (Mở rộng: Chatbot Telegram cho phép Admin gửi lệnh /status).
Báo cáo/Thống	Cần dữ liệu để tối ưu hóa lộ	Logging chi tiết mọi sự kiện (mở nắp, quét thẻ, cảnh báo) vào cơ sở

kê	trình thu gom và đánh giá hiệu suất.	dữ liệu MySQL. Dữ liệu mức đầy realtime hiển thị trên Web Dashboard.
Kiểm soát Truy cập	Kiểm soát những ai được phép sử dụng thùng rác.	Phân quyền người dùng (RBAC) bằng thẻ RFID, ghi lại lịch sử hoạt động chi tiết.

iii. Kỹ thuật/IT (Nhóm Phát triển, Quản trị Hệ thống)

Yêu cầu	Mô tả Chi tiết (Tích hợp, Giao thức, Bảo mật)	Đã đáp ứng (Công nghệ)
Tích hợp/Mở rộng	Hệ thống phải dễ dàng tích hợp thiết bị mới và mở rộng quy mô.	Kiến trúc Pub/Sub (MQTT) trên nền tảng Mosquitto Broker đảm bảo giao tiếp phi đồng bộ và tính mở rộng.
Giao thức hiệu quả	Cần giao thức nhẹ, tối ưu cho thiết bị IoT có tài nguyên hạn chế.	Sử dụng MQTT, một giao thức nhẹ, tối ưu cho IoT.
Xử lý lỗi	Cần biết trạng thái của thiết bị khi bị ngắt kết nối.	Triển khai tính năng Last Will and Testament (LWT) để tự động cảnh báo trạng thái offline của ESP32.

#### iv. Phương pháp thu thập yêu cầu

Phương pháp được đề xuất áp dụng là kết hợp để có cái nhìn toàn diện:

- Phỏng vấn/Quan sát thực tế:
  - Mục đích: Hiểu rõ hành vi đồ rác của người dùng và các vấn đề "Điểm đau" (Pain Points) của quy trình thu gom rác hiện tại (ví dụ: rác tràn, mùi hôi, thu gom lãng phí).
  - Áp dụng: Áp dụng cho mục tiêu tự động hóa (mở nắp không chạm) và tối ưu hóa (giám sát mức đầy).
- Phân tích Tài liệu/Yêu cầu Kỹ thuật:
  - Mục đích: Xác định các yêu cầu phi chức năng về hiệu năng, bảo mật, và độ tin cậy.
  - Áp dụng: Xác định các tiêu chuẩn như QoS 0/QoS 1, thời gian phản hồi lệnh điều khiển ( $< 500\text{ms}$ ), và mô hình RBAC.

## II. Cơ sở lý thuyết

### a. Mô hình truyền thông Publish/Subscribe (Pub/Sub)

- i. Khái niệm: Mô hình giao tiếp phi đồng bộ, trong đó publisher gửi message lên topic, subscriber nhận message khi đăng ký topic đó.
- ii. Ưu điểm: giảm tải kết nối trực tiếp, tăng tính mở rộng, hỗ trợ realtime.
- iii. Ứng dụng: Truyền dữ liệu giữa ESP32  $\leftrightarrow$  MQTT Broker  $\leftrightarrow$  Server  $\leftrightarrow$  Web/App.

### b. Giao thức MQTT (Message Queuing Telemetry Transport)

- i. Đặc điểm: giao thức nhẹ, tối ưu cho IoT, sử dụng TCP/IP.
- ii. Thành phần: Publisher, Subscriber, Broker (Mosquitto/CloudMQTT).
- iii. Cơ chế QoS:



- QoS 0 – at most once (nhANH, chấp nhận mất gói)
- QoS 1 – at least once (đảm bảo đến ít nhất 1 lần)
- QoS 2 – exactly once (hiếm dùng, tốn tài nguyên)
- iv. Ứng dụng: gửi dữ liệu cảm biến, lệnh điều khiển, cảnh báo, log sự kiện.
- c. Nguyên lý hoạt động của cảm biến
  - i. HC-SR04 (Siêu âm): đo khoảng cách dựa trên thời gian truyền – phản xạ sóng âm.  
→ Dùng để tính mức đầy thùng:  

$$fill\_level = \left(1 - \frac{distance}{bin\_height}\right) \times 100\%$$
  - ii. RC522 (RFID): đọc UID thẻ qua giao tiếp SPI, dùng cho xác thực người dùng.
  - iii. Servo SG90: điều khiển góc quay để mở/đóng nắp rác (thường  $0^\circ \leftrightarrow 90^\circ$ ).
- d. Phân quyền người dùng & xác thực
  - i. Khái niệm: Hệ thống có nhiều mức quyền (Admin, User).
  - ii. Cơ chế: Mỗi UID thẻ RFID được ánh xạ đến người dùng trong DB.
  - iii. Lý thuyết: Mô hình RBAC (Role-Based Access Control) – phân quyền theo vai trò.
- e. Giao tiếp Client-Server & Realtime
  - i. Client: ESP32 và Web/App.
  - ii. Server: Node.js backend.
  - iii. Kết nối realtime: WebSocket (ws://broker:8080) qua MQTT.js để cập nhật tức thì.
- f. Cơ sở dữ liệu và logging
  - i. MySQL: mô hình quan hệ, dùng để lưu user, log, cấu hình.
  - ii. Chuẩn hóa dữ liệu (Normalization) – tránh dư thừa, dễ truy vấn.
  - iii. Timestamp & logging: Ghi lại sự kiện giúp truy vết hệ thống, hỗ trợ phân tích.
- g. Kiến trúc hệ thống IoT

- i. Gồm 3 lớp:
  - ii. Perception Layer: cảm biến, vi điều khiển (ESP32, HC-SR04, RC522)
  - iii. Network Layer: truyền thông MQTT qua Internet
  - iv. Application Layer: Web/App dashboard, xử lý backend
  - v. Mục tiêu: Kết nối thiết bị vật lý ↔ Dữ liệu ↔ Người dùng.
- h. Nguyên lý hệ thống nhúng (Embedded Systems)
  - i. ESP32: vi điều khiển có WiFi/Bluetooth, lập trình qua Arduino IDE.
  - ii. Chức năng chính: đọc cảm biến, publish dữ liệu MQTT, nhận lệnh điều khiển.
- i. Tự động hóa và IoT (Internet of Things)
  - i. IoT concept: kết nối thiết bị vật lý qua Internet để giám sát, điều khiển, cập nhật firmware.
  - ii. Đặc điểm: realtime, tiết kiệm năng lượng, dễ mở rộng.
  - iii. Ứng dụng: tự động mở nắp, cảnh báo đầy, thống kê người dùng.

### III. Công nghệ sử dụng

Thành phần	Công nghệ	Vai trò / Ghi chú
<b>Vi điều khiển</b>	ESP32	Thiết bị IoT chính, xử lý cảm biến, gửi/nhận MQTT
<b>Cảm biến</b>	HC-SR04, RC522, Servo SG90	Đo khoảng cách, nhận diện RFID, mở/đóng nắp
<b>Giao thức IoT</b>	MQTT (Mosquitto / CloudMQTT)	Truyền dữ liệu realtime giữa thiết bị ↔ server ↔ app
<b>Backend Server</b>	Node.js + Express + MQTT.js	Xử lý logic, xác thực RFID, lưu log

<b>Cơ sở dữ liệu</b>	MySQL	Lưu thông tin user, log, cấu hình
<b>Frontend Web</b>	React.js + MQTT.js	Giao diện realtime hiển thị dữ liệu và điều khiển
<b>UI Library</b>	Material UI / Ant Design	Thiết kế giao diện trực quan
<b>Biểu đồ realtime</b>	Recharts / Chart.js	Hiển thị % đầy thùng rác, biểu đồ mức rác
<b>WebSocket</b>	ws://broker:8080	Duy trì kết nối realtime cho Web/App
<b>Cloud Hosting (tùy chọn)</b>	CloudMQTT / AWS IoT Core	Nếu triển khai online thay cho self-hosted broker
<b>Ngôn ngữ lập trình</b>	C++ (Arduino IDE) / JavaScript (Node.js, React)	Lập trình nhúng và web
<b>Kiến trúc hệ thống</b>	Publish-Subscribe (Pub/Sub)	Mô hình kết nối dữ liệu phi đồng bộ
<b>Phân quyền</b>	RBAC (Role-Based Access Control)	Quản lý user bằng role admin/user

#### IV. Phân tích yêu cầu

##### a. Yêu cầu chức năng

##### i. Chức năng thu thập và truyền dữ liệu mức đầy

Hệ thống đảm bảo mỗi thùng rác IoT đều có khả năng tự động đo lường mức đầy theo thời gian thực bằng cảm biến siêu âm HC-SR04 (hoặc tùy chọn thêm Loadcell để đo trọng lượng). Cảm biến HC-SR04 có dải đo từ 2 cm đến 400 cm, sai số kỹ thuật  $\pm 3$  mm, được đặt ở nắp

thùng để đo khoảng cách từ cảm biến đến bề mặt rác. Thiết bị phải tính toán phần trăm mức đầy theo công thức:

$$fill\_level = \left(1 - \frac{distance}{bin\_height}\right) \times 100\%$$

Sau khi đo, thiết bị gửi dữ liệu về hệ thống thông qua giao thức MQTT theo chu kỳ mặc định 10 giây/lần. Ngoài ra, để tránh lãng phí băng thông, thiết bị gửi thêm một gói dữ liệu phụ nếu mức đầy thay đổi trên 5% so với lần gửi trước. Trong trường hợp phát hiện sự bất thường (ví dụ giá trị đo bị sai lệch quá lớn), thiết bị cần tự động lặp lại việc đo tối đa 3 lần trước khi báo lỗi.

#### ii. Chức năng xác thực người dùng và phân quyền truy cập

Mỗi thùng rác hỗ trợ xác thực người dùng thông qua đầu đọc RFID RC522, hoạt động ở tần số 13.56 MHz, có khả năng đọc UID thẻ trong khoảng cách 0–3 cm. Khi người dùng quét thẻ, thiết bị gửi UID lên hệ thống backend để kiểm tra quyền trong cơ sở dữ liệu. Thời gian từ lúc quét thẻ đến khi nhận được phản hồi xác thực phải đảm bảo  $\leq 300$  mili giây.

Hệ thống phân loại người dùng thành ít nhất 3 nhóm:

- User (người sử dụng thông thường): chỉ có quyền mở nắp để bỏ rác.
- Nhân viên thu gom / bảo trì: chỉ được xem những thùng đầy hoặc gặp lỗi.
- Quản trị viên (Admin): toàn quyền điều khiển từ xa, cấu hình ngưỡng cảnh báo và thêm hoặc xóa thiết bị.

Toàn bộ các lần xác thực đều phải được ghi vào log hoạt động, bao gồm: UID người dùng, timestamp, kết quả xác thực (thành công / bị từ chối).

#### iii. Chức năng điều khiển mở nắp (local & remote)

Thùng rác hỗ trợ hai chế độ mở nắp: **tự động mở khi người dùng đến gần** (nếu bật cảm biến chuyển động hoặc cảm biến siêu âm hướng ngang) hoặc **chỉ mở khi có xác thực hợp lệ**. Động cơ sử dụng là Servo SG90, có góc quay  $0^{\circ}$ – $90^{\circ}$ , mô-men xoắn 1.8 kg/cm, đủ để nâng nắp có tải trọng nhỏ. Sau khi xác thực hợp lệ, nắp phải được mở hoàn toàn trong thời gian  $\leq 500\text{ms}$ .

Đối với điều khiển từ xa, Admin hoặc hệ thống giám sát có thể gửi lệnh mở hoặc khóa nắp thông qua MQTT. Lệnh điều khiển phải được gửi với QoS 1 để đảm bảo không bị mất gói. Nếu nắp đang mở **quá 5 giây mà không có chuyển động**, hệ thống phải **tự động đóng lại** để tiết kiệm điện năng và tránh mùi thoát ra.

iv. Chức năng giám sát trạng thái thiết bị theo thời gian thực

Hệ thống cung cấp giao diện giám sát tập trung (Dashboard) hiển thị toàn bộ các thùng rác đang hoạt động. Mỗi thiết bị phải có mã định danh riêng (ví dụ: BIN\_01, BIN\_02, hoặc theo định dạng vị trí như T2\_HALLWAY\_03) để dễ dàng phân loại. Trạng thái của mỗi thùng phải được cập nhật theo thời gian thực, bao gồm:

- Mức đầy hiện tại theo phần trăm (% Full).
- Trạng thái nắp (Open / Closed).
- Trạng thái kết nối (Online / Offline, với quy tắc nếu không nhận được dữ liệu trong 30 giây thì coi như mất kết nối).
- Lần cập nhật cuối (timestamp dạng ISO 8601, ví dụ “2025-10-13T21:05:33+07:00”).

Nếu có từ **3 thùng liên tiếp đạt mức đầy  $\geq 80\%$** , giao diện cần tự động chuyển sang chế độ “**Cảnh báo ưu tiên cao**”, làm **nổi bật các thùng này** bằng màu đỏ hoặc nhấp nháy để nhân viên thu gom dễ dàng nhận diện.

v. Chức năng cảnh báo và gửi thông báo

Khi một thùng đạt ngưỡng cảnh báo đầy (mặc định là 80% nhưng phải được phép cấu hình lại theo từng vị trí), hệ thống phải lập tức phát thông báo bằng ít nhất một trong các phương thức sau:

- Thông báo trực tiếp trên Dashboard.
- Gửi tin nhắn qua Telegram Bot hoặc Email tới nhóm nhân viên thu gom.
- Bật tín hiệu cảnh báo tại chỗ (tùy chọn) như buzzer hoặc đèn LED nhấp nháy.

Trong trường hợp mức đầy tiếp tục tăng lên  $\geq 95\%$  mà vẫn chưa được xử lý, hệ thống phải đánh dấu trạng thái thiết bị là “Khẩn cấp – Cần thu gom ngay” và có thể tự động gọi chuyển hướng thiết bị hiển thị lên đầu danh sách ưu tiên.

Ngoài cảnh báo đầy, hệ thống cũng phải cảnh báo khi:

- Thùng bị mở nắp quá thời gian quy định (ví dụ  $> 60$  giây).
- Phát hiện cảm biến lỗi / giá trị bất thường liên tiếp 5 lần.
- Thiết bị không gửi dữ liệu quá 30 giây  $\rightarrow$  đánh dấu Offline.

vi. Chức năng ghi log và lưu trữ lịch sử hoạt động

Mọi sự kiện quan trọng đều được ghi lại dưới dạng log trong cơ sở dữ liệu. Tối thiểu phải lưu lại các bản ghi với các trường dữ liệu như sau:

Loại sự kiện	Dữ liệu cần lưu	Ví dụ
Quét thẻ RFID	UID, loại người dùng, kết quả xác thực, thời gian	UID=0xA4C3B2, ROLE=User, RESULT=Accepted, TIME=2025-10-13 21:10:05
Mở nắp thủ công/từ xa	Người thực hiện (hoặc Thiết bị), nguồn lệnh, thời gian	ACTION=Open, SOURCE=Remote, BY=Admin01

Cảnh báo mức đầy	Thiết bị, giá trị đo, thời gian cảnh báo	BIN_02, LEVEL=82%, TIME=...
Mất kết nối / khôi phục	Thiết bị, trạng thái, thời gian	BIN_03 → Offline

Các bản ghi này được lưu trữ tối thiểu 30 ngày trên hệ thống và có thể xuất ra dưới dạng CSV hoặc Excel nếu cần phục vụ công tác phân tích định kỳ.

vii. Chức năng thống kê và tối ưu hoạt động thu gom

Hệ thống cung cấp giao diện thống kê cho phép người quản trị xem lại lịch sử sử dụng theo từng khoảng thời gian. Thống kê cần hỗ trợ tối thiểu các dạng sau:

- Thống kê theo tần suất vứt rác / số lần mở nắp mỗi ngày.
- Biểu đồ mức đầy của từng thùng theo thời gian để xác định chu kỳ rác đầy.
- Xếp hạng các thùng hoạt động nhiều nhất / ít nhất, từ đó có thể đề xuất việc thay đổi vị trí hoặc điều chỉnh lịch thu gom.

Nếu hệ thống phát hiện một thùng có xu hướng đầy trong thời gian ngắn hơn 50% so với trung bình, hệ thống phải tự động đánh dấu là “Vị trí sử dụng cao” và gợi ý tăng tần suất kiểm tra.

viii. Chức năng mở rộng và quản lý thiết bị động

Hệ thống hỗ trợ thêm thùng rác mới dễ dàng mà không cần lập trình lại toàn bộ. Thiết bị mới chỉ cần:

- Được gán mã định danh duy nhất (ID) và
- Đăng ký topic MQTT đúng định dạng quy ước, ví dụ:  
`bin/{ID}/level`.

Ngay khi thiết bị gửi gói tin đầu tiên lên hệ thống, hệ thống phải tự động nhận diện và thêm vào danh sách thiết bị đang quản lý, đưa vào nhóm “Thiết bị mới”. Quản trị viên có thể bổ sung thông tin như vị trí, ngưỡng cảnh báo riêng, chế độ hoạt động ngay từ giao diện.

Đồng thời cập nhật firmware từ xa (OTA) qua mạng.

b. Yêu cầu phi chức năng

i. Yêu cầu về hiệu năng (Performance Requirements)

Hệ thống đảm bảo hoạt động mượt mà ngay cả khi có nhiều thiết bị gửi dữ liệu đồng thời. Cụ thể:



- Thời gian phản hồi từ lúc quét thẻ RFID đến khi nắp mở không được vượt quá 500 mili giây trong điều kiện mạng Wi-Fi ổn định (độ trễ < 50ms).
- Độ trễ truyền dữ liệu từ thiết bị lên Dashboard hiển thị không quá 1 giây. Nếu vượt quá, hệ thống phải đánh dấu là “Cảnh báo chậm trễ”.
- Tần suất gửi dữ liệu tối thiểu phải đạt 6 gói/phút cho mỗi thiết bị (tương ứng chu kỳ 10 giây/lần), và hệ thống phải xử lý được tối thiểu 100 gói dữ liệu/phút mà không gây nghẽn hoặc mất mát thông tin.
- Khi số lượng thiết bị vượt quá 30 thùng, hệ thống phải tự động kích hoạt cơ chế giảm tần suất gửi dữ liệu còn 20 giây/lần để đảm bảo ổn định (adaptive rate control).

#### ii. Yêu cầu về bảo mật (Security Requirements)

Vì hệ thống có liên quan đến xác thực người dùng và điều khiển thiết bị từ xa nên cần tuân thủ các yêu cầu bảo mật sau:

- Kết nối MQTT phải yêu cầu xác thực bằng username/password. Password mặc định phải dài tối thiểu 8 ký tự, bao gồm ít nhất 1 chữ hoa + 1 chữ thường + 1 ký tự số.
- Thiết bị chỉ được phép publish và subscribe trong phạm vi topic của chính nó, ví dụ bin/01/#. Nếu thiết bị cố tình truy cập topic khác, hệ thống phải từ chối kết nối ngay lập tức.
- Nếu triển khai ngoài mạng nội bộ, hệ thống phải hỗ trợ mã hóa SSL/TLS (port 8883) hoặc chạy trong VPN nội bộ để tránh rò rỉ dữ liệu.
- Các lệnh điều khiển từ xa (Open/Close) phải được hệ thống đánh dấu bằng mã định danh lệnh (CommandID) để tránh tình trạng xử lý lặp lại nếu gói tin bị gửi lại do QoS 1.

#### iii. Yêu cầu về độ tin cậy và khả năng phục hồi (Reliability & Fault Tolerance)

Hệ thống được thiết kế để đảm bảo không bị gián đoạn ngay cả khi một số thiết bị gặp lỗi. Cụ thể:

- Nếu một thiết bị không gửi tín hiệu trong hơn 30 giây, hệ thống phải tự động đánh dấu trạng thái là “Offline” và ghi log sự kiện DeviceLostConnection.
- Thiết bị phải có cơ chế tự kết nối lại MQTT tối đa 5 lần liên tiếp, mỗi lần cách nhau 3 giây trước khi báo lỗi.
- Các trạng thái quan trọng như mức đầy gần nhất, trạng thái nắp hiện tại phải được lưu dưới dạng Retained Message trên Broker, để khi hệ thống khởi động lại vẫn hiển thị chính xác dữ liệu cuối.
- Trong trường hợp lỗi cơ sở dữ liệu, hệ thống vẫn phải ưu tiên hiển thị trạng thái realtime từ MQTT thay vì tệp liệt hoàn toàn.

iv. Yêu cầu về khả năng mở rộng (Scalability Requirements)

Hệ thống có khả năng hoạt động ổn định với đa dạng quy mô, từ một thùng đơn lẻ đến một hệ thống lớn gồm hàng chục thùng trong nhiều tầng hoặc nhiều khu vực.

- Kiến trúc hỗ trợ mở rộng theo chiều ngang (horizontal scaling): có thể thêm nhiều broker hoặc server xử lý mà không phải thay đổi firmware thiết bị.
- Hệ thống có thể thiết kế theo nguyên tắc cấu hình tự động (auto discovery): thiết bị mới chỉ cần gửi dữ liệu lần đầu là được tự động thêm vào danh sách giám sát.
- Nếu số lượng thiết bị vượt 50 thùng, hệ thống phải gợi ý chuyển sang các dịch vụ CloudMQTT, EMQX Cloud hoặc AWS IoT Core.

v. Yêu cầu về tối ưu năng lượng và băng thông (Energy & Bandwidth Efficiency)

Mô hình chạy bằng pin thiết bị phải được tối ưu để tiêu thụ năng lượng thấp:

- Thiết bị phải hỗ trợ Deep Sleep Mode và chỉ thức dậy khi cần gửi dữ liệu hoặc nhận lệnh, giúp kéo dài thời gian sử dụng pin lên tối thiểu 24 giờ liên tục.
- Dung lượng mỗi gói MQTT không được vượt quá 200 byte, tổng lưu lượng truyền của mỗi thiết bị không vượt quá 50KB/ngày để tiết kiệm băng thông và chi phí cloud.
- Nếu thiết bị không nhận lệnh trong thời gian dài thì phải tạm ngưng việc subscribe các lệnh điều khiển để tiết kiệm năng lượng (Idle Mode).

## V. Phân tích ràng buộc

### a. Ràng buộc môi trường

- Nhiệt độ: 0–50°C, độ ẩm 20–90% (không ngưng tụ).
- Nhiều Wi-Fi 2.4GHz: khoảng cách ESP32–router  $\leq 50\text{m}$ ; giải pháp auto-reconnect sau 5s, chọn kênh Wi-Fi ít nhiễu.
- Nguồn điện: 5V–2A, công suất  $\sim 5\text{W}$ ; cần ổn định, tránh nhiễu cảm biến.
- Cảm biến: HC-SR04 đặt giữa nắp; lấy trung bình 3–5 lần đo; RC522 đặt cạnh nắp, servo tại bản lề.

### b. Ràng buộc pháp lý & bảo mật

- Wi-Fi: tuân thủ Thông tư 26/2019/TT-BTTTT (2400–2483.5MHz, công suất  $\leq 100\text{mW}$ ).
- RFID 13.56MHz chuẩn ISO/IEC 14443A, không cần cấp phép.
- MQTT: dùng broker Mosquitto/HiveMQ, xác thực user/pass, mã hóa TLS/SSL (port 8883).
- Phân quyền topic: ESP32 chỉ publish “smartbin/#”, subscribe “lid/command”, “rfid/response”.
- Bảo mật dữ liệu cá nhân: mã hóa UID RFID, log tự xóa sau 6 tháng.

### c. Ràng buộc tài nguyên thiết bị

- ESP32: Dual-core 240MHz, RAM 520KB  $\rightarrow$  cần code tối ưu.
- Buffer offline tối đa 50 message ( $\sim 20\text{KB}$ ).
- QoS: level (0), lệnh mở nắp/RFID/cảnh báo (1).
- Băng thông:  $\sim 17\text{KB/phút/thiết bị}$  (1 message/10s).

d. Ràng buộc hiệu năng

- Độ trễ mở nắp <500ms (100ms mạng + 200ms xử lý + 200ms hiển thị).
- Dashboard cập nhật <1s; cảm biến gửi dữ liệu mỗi 10s hoặc khi thay đổi >5%.
- Mất kết nối: tự reconnect sau 5s, lưu 50 message trong EEPROM, dùng LWT & Retained message để báo trạng thái.

e. Ràng buộc mở rộng

- MQTT Broker hỗ trợ 1000 kết nối, hệ thống thiết kế cho 10–50 thiết bị.
- Cấu trúc topic: smartbin/{device\_id}/{category}/{action}.
- Giới hạn 3–4 cấp để dễ quản lý & mở rộng.

VI. Mô hình hoá

a. Ưu tiên hoá

Must-have:

Nội dung yêu cầu	Loại	Ghi chú
Đo mức đầy thùng rác định kỳ (10s) và gửi lại khi thay đổi >5%.	Chức năng	Thành phần đo lường chính của hệ thống.
Gửi dữ liệu qua MQTT (smartbin/{id}/level) với QoS = 1.	Chức năng	Đảm bảo dữ liệu truyền ổn định và không mất gói.
Xác thực thẻ RFID qua backend, phản hồi $\leq 300$ ms.	Chức năng	Liên quan bảo mật và tính khả dụng.
Mở nắp sau xác thực; tự đóng sau 5 giây nếu không hoạt động.	Chức năng	Hành vi cơ bản của thùng rác thông minh.
Ghi log toàn bộ sự kiện (scan, open, alert, offline) vào CSDL $\geq 30$ ngày.	Chức năng	Phục vụ truy vết và báo cáo vận hành.
Dashboard hiển thị realtime: % đầy, trạng thái nắp, kết nối, thời gian.	Chức năng	Yêu cầu cốt lõi để giám sát hệ thống.
Hỗ trợ cập nhật firmware từ xa (OTA) qua mạng	Chức năng	Bắt buộc để bảo trì, vá lỗi, nâng cấp thiết bị.

Hiệu năng: phản hồi RFID $\leq 500$ ms; dashboard cập nhật $\leq 1$ s.	Phi chức năng	Đảm bảo trải nghiệm người dùng mượt.
Độ tin cậy: reconnect tối đa 5 lần; offline $> 30$ s $\rightarrow$ trạng thái “Offline”.	Phi chức năng	Đảm bảo hệ thống luôn giám sát được.
Bảo mật: MQTT có username/password, hỗ trợ TLS, phân quyền topic.	Phi chức năng	Ngăn truy cập trái phép và rò rỉ dữ liệu.

Should-have:

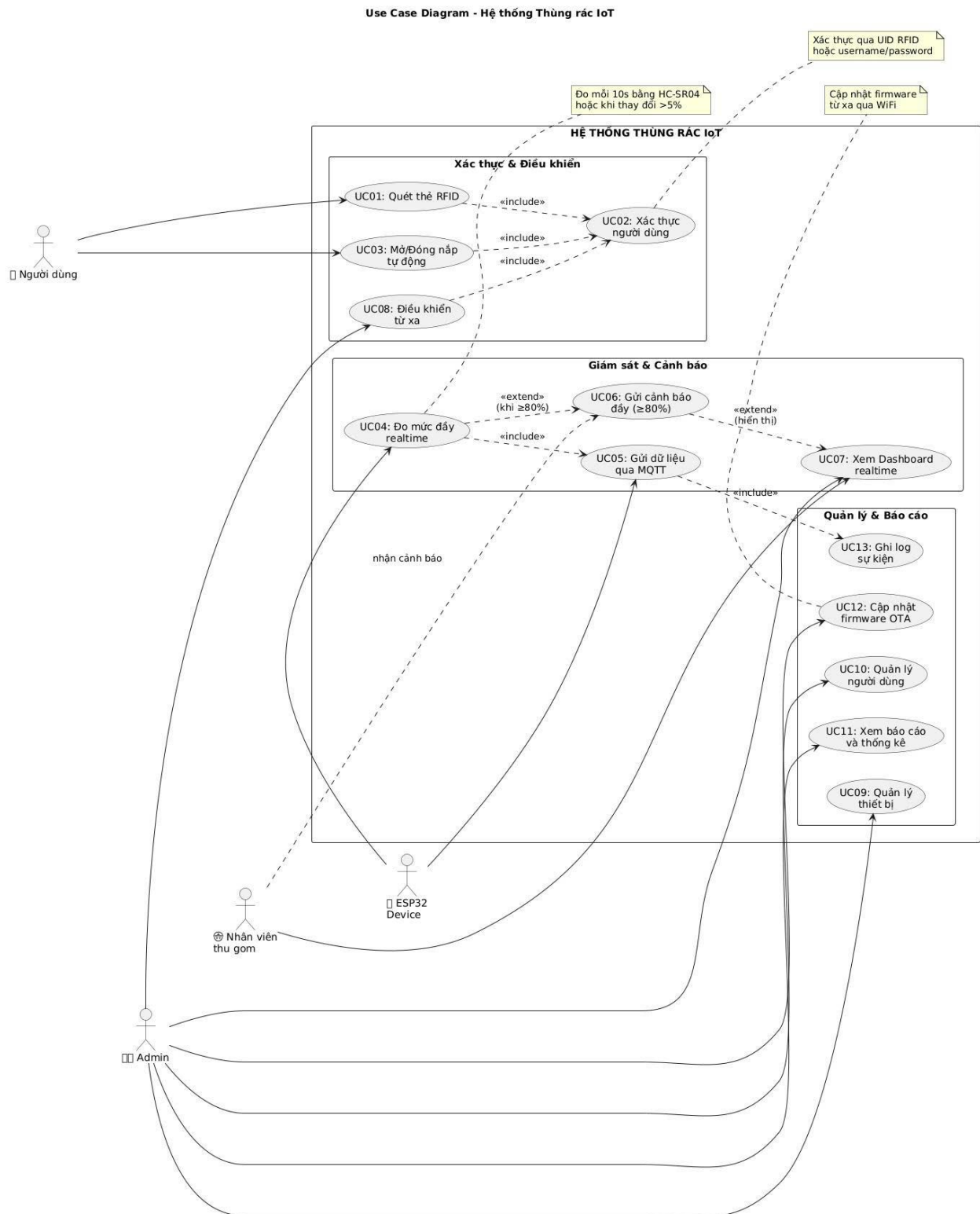
Nội dung yêu cầu	Loại	Ghi chú
Cảnh báo tự động khi $\geq 80\%$ , nâng “Khẩn cấp” khi $\geq 95\%$ .	Chức năng	Cải thiện hiệu quả vận hành.
Thiết bị tự đăng ký khi publish lần đầu (auto-register).	Chức năng	Giúp triển khai nhanh hơn.
Xuất báo cáo CSV/Excel theo khoảng thời gian tùy chọn.	Chức năng	Hỗ trợ thống kê và lưu trữ.
Khả năng mở rộng: hỗ trợ 10–50 thiết bị, gợi ý chuyển cloud khi $> 50$ .	Phi chức năng	Dự phòng mở rộng quy mô.
Tiết kiệm năng lượng: Deep Sleep, giới hạn payload $\leq 200$ bytes.	Phi chức năng	Tăng hiệu suất và tiết kiệm năng lượng.

Nice-to-have:

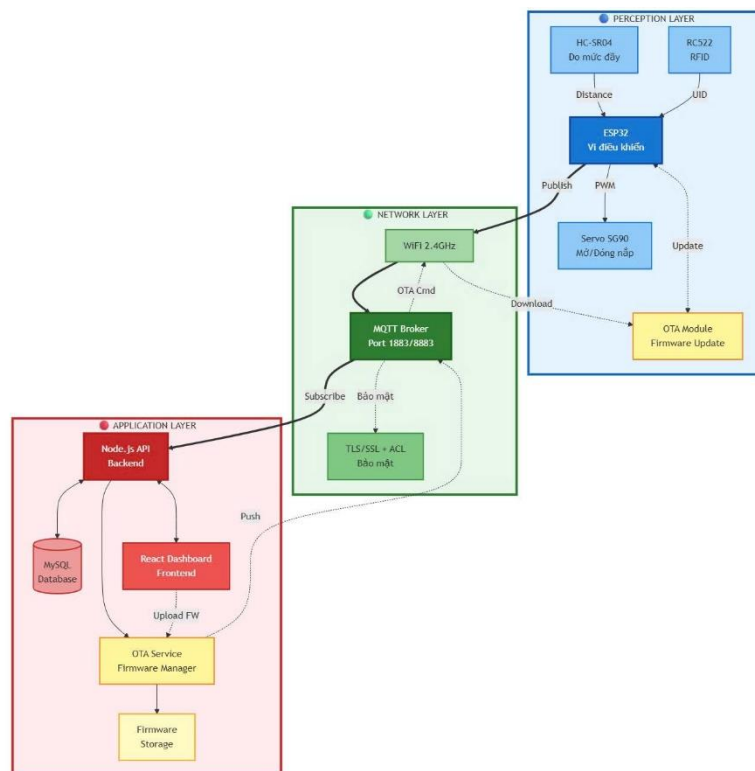
Nội dung yêu cầu	Loại	Ghi chú
Tích hợp thông báo nâng cao qua Telegram Bot / Email.	Chức năng	Cảnh báo tiện lợi, không bắt buộc.
Phân tích dữ liệu lịch sử, dự đoán xu hướng (Machine Learning cơ bản).	Chức năng	Tăng khả năng thông minh hệ thống.
Dashboard hỗ trợ đa ngôn ngữ (VN/EN).	Chức năng	Nâng cao trải nghiệm người dùng.

## b. Mô hình

### i. Use Case



ii. 3 layers



VII. Phân chia công việc

Họ và tên	Mã sinh viên	Công việc
Nguyễn Anh Tuấn	B22DCCN758	Lập trình chức năng bằng Arduino cho ESP và viết báo cáo
Đình Công Thịnh	B22DCCN830	Lập trình chức năng bằng Arduino cho ESP và viết báo cáo
Đỗ Nhật Anh	B22DCCN013	Lập trình giao thức kết nối MQTT, lập trình OTA, lắp thiết bị
Nguyễn Như Thuật	B22DCCN842	Lập trình Front-End, xử lý dữ liệu từ Client, từ cảm biến