

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO BÀI TẬP LỚN

MÔN: HỆ THỐNG HƯỚNG DỊCH VỤ

XÂY DỰNG HỆ THỐNG QUẢN LÝ TÀI SẢN THÔNG MINH
(AI INVENTORY SYSTEM)

Giảng viên hướng dẫn: TS Lê Khánh Trình

Nhóm thực hiện: 5

Thành viên

Nguyễn Văn Hưng 22028118

Phạm Văn Quyền 23021678

MỤC LỤC

CHƯƠNG I: XÁC ĐỊNH BÀI TOÁN	3
CHƯƠNG II: PHÂN TÍCH YÊU CẦU NGƯỜI DÙNG	4
2.1. Yêu cầu chức năng	4
2.2. Yêu cầu phi chức năng	4
CHƯƠNG III: THIẾT KẾ HỆ THỐNG	5
3.1. Biểu đồ use-case	5
3.1.1. Xác định các tác nhân	5
3.1.2. Xác định các use-case	5
3.1.3. Biểu đồ use-case người quản trị / Cán bộ (Staff)	5
3.1.4. Biểu đồ use-case phân hệ AI (Tương tự Sensor)	6
3.2. BIỂU ĐỒ HOẠT ĐỘNG	7
3.3. BIỂU ĐỒ TRÌNH TỰ	8
3.3.1. Biểu đồ trình tự ca sử dụng toàn bộ hệ thống	8
3.3.2. Biểu đồ trình tự ca sử dụng Đăng nhập (Login)	8
3.3.3. Biểu đồ trình tự ca sử dụng Xem danh sách tài sản	8
3.3.4. Biểu đồ trình tự ca sử dụng Thêm tài sản mới	8
3.3.5. Biểu đồ trình tự ca sử dụng Cập nhật tài sản	9
3.3.6. Biểu đồ trình tự ca sử dụng Xóa tài sản	9
3.3.7. Biểu đồ trình tự ca sử dụng Tìm kiếm tài sản	9
3.3.8. Biểu đồ trình tự ca sử dụng Upload Video (Gửi dữ liệu quét)	9
3.3.9. Biểu đồ trình tự ca sử dụng AI xử lý (Sensor nhận diện)	9
3.3.10. Biểu đồ trình tự ca sử dụng Xem chi tiết kết quả quét	10
3.4. BIỂU ĐỒ CỘNG TÁC	10
3.4.1. Biểu đồ cộng tác ca sử dụng Đăng nhập	10
3.4.2. Biểu đồ cộng tác ca sử dụng Thêm tài sản	10
3.5. BIỂU ĐỒ LỚP (CLASS DIAGRAM)	10
3.5.1. Mô tả	10
3.5.2. Những lớp/đối tượng quan trọng	11
3.5.3. Các lớp Service và Interface	12
3.6. BIỂU ĐỒ TRANG THÁI (STATE DIAGRAM)	12
3.6.1. Biểu đồ trạng thái của Tài sản (Asset)	12
3.6.2. Biểu đồ trạng thái của Phiên quét (Scan Session)	12
3.7. BIỂU ĐỒ THÀNH PHẦN (COMPONENT DIAGRAM)	12
3.8. BIỂU ĐỒ TRIỂN KHAI (DEPLOYMENT DIAGRAM)	12
CHƯƠNG IV: KẾT QUẢ CÀI ĐẶT VÀ GIAO DIỆN	13
4.1. Môi trường cài đặt	13
4.2. Giao diện chương trình	13
KẾT LUẬN	14

CHƯƠNG I: XÁC ĐỊNH BÀI TOÁN

Trong bối cảnh hiện đại hóa công tác quản lý tại Đại học Công nghệ (UET), việc kiểm soát cơ sở vật chất (bàn ghế, máy chiếu, máy tính, điều hòa...) tại các giảng đường G2, GD2, GD3 là vô cùng cấp thiết. Hiện trạng quản lý thủ công bộc lộ nhiều yếu điểm: tốn nhân lực kiểm kê, dữ liệu không cập nhật tức thời, khó tra cứu lịch sử hỏng hóc.

Hệ thống **AI Inventory System** được xây dựng nhằm giải quyết bài toán trên thông qua công nghệ Web và AI (Computer Vision). Hệ thống cho phép:

- Quản lý tập trung danh sách tài sản, vị trí, tình trạng.
- Tự động hóa quy trình kiểm kê bằng cách quay video/chụp ảnh phòng học, hệ thống tự động nhận diện và đếm thiết bị.
- Cung cấp báo cáo trực quan cho ban quản lý.

CHƯƠNG II: PHÂN TÍCH YÊU CẦU NGƯỜI DÙNG

2.1. Yêu cầu chức năng

Hệ thống cần đáp ứng các nhóm chức năng sau:

- Quản trị hệ thống (Admin):** Đăng nhập, quản lý tài khoản người dùng (cán bộ), cấu hình tham số hệ thống.
- Quản lý tài sản (Staff):**
 - Thêm mới, cập nhật, xóa tài sản.
 - Tìm kiếm, tra cứu thông tin tài sản theo mã, tên, vị trí (Giảng đường).
- Kiểm kê thông minh (AI Scan):**
 - Tải lên video/hình ảnh hiện trường.
 - Xử lý nhận diện vật thể (Object Detection) tự động.
 - Lưu trữ lịch sử quét và kết quả nhận diện.
 - So khớp kết quả quét với dữ liệu tồn kho.
- Báo cáo thống kê:** Xem biểu đồ thống kê số lượng, giá trị, tình trạng tài sản.

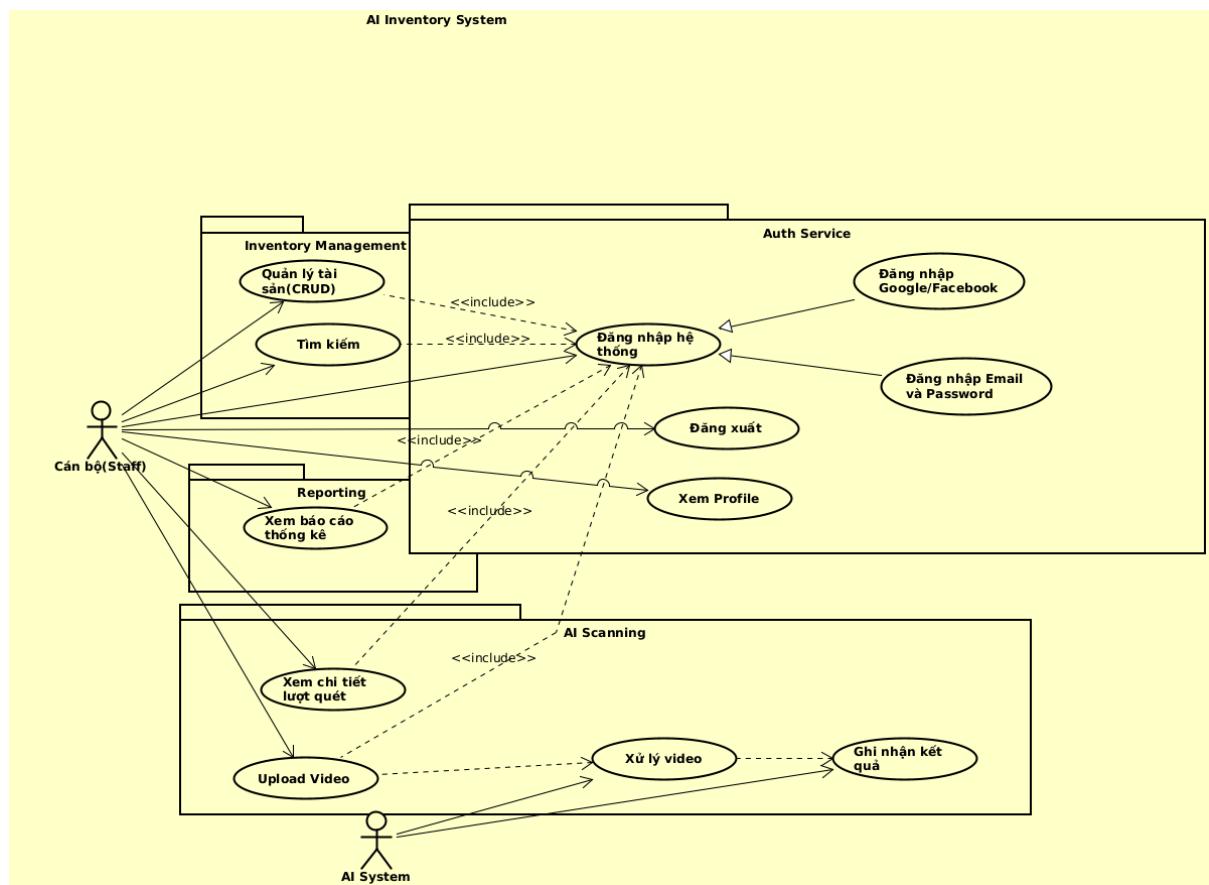
2.2. Yêu cầu phi chức năng

- Hiệu năng:** Thời gian phản hồi cho các thao tác quản lý < 1 giây. Thời gian xử lý AI cho video 1 phút < 30 giây.
- Bảo mật:** Mật khẩu được mã hóa. Sử dụng cơ chế Access Token và Refresh Token để bảo vệ phiên làm việc.
- Giao diện:** Thân thiện, hỗ trợ Tiếng Việt, hoạt động tốt trên trình duyệt Web (Chrome, Edge).

4. Độ tin cậy: AI nhận diện chính xác > 85% đối với các thiết bị tiêu chuẩn (màn hình, bàn ghế).

CHƯƠNG III: THIẾT KẾ HỆ THỐNG

3.1. Biểu đồ use-case



Hình 0. Biểu đồ use-case chung cho toàn bộ hệ thống

3.1.1. Xác định các tác nhân

Hệ thống bao gồm các tác nhân chính tương tác trực tiếp:

ST T	Tác nhân	Mô tả
1	Cán bộ (Staff)	Người dùng trực tiếp sử dụng giao diện web để quản lý tài sản, tải lên dữ liệu kiểm kê và khai thác báo cáo.

2	AI Service	Tác nhân hệ thống (tương tự như Sensor), hoạt động tự động để nhận diện vật thể từ video và gửi kết quả về hệ thống trung tâm.
---	-------------------	--

3.1.2. Xác định các use-case

Hệ thống bao gồm các Use-case chính được chia theo phân hệ:

Dựa trên phân tích yêu cầu, các Use-case được xác định và phân nhóm như sau:

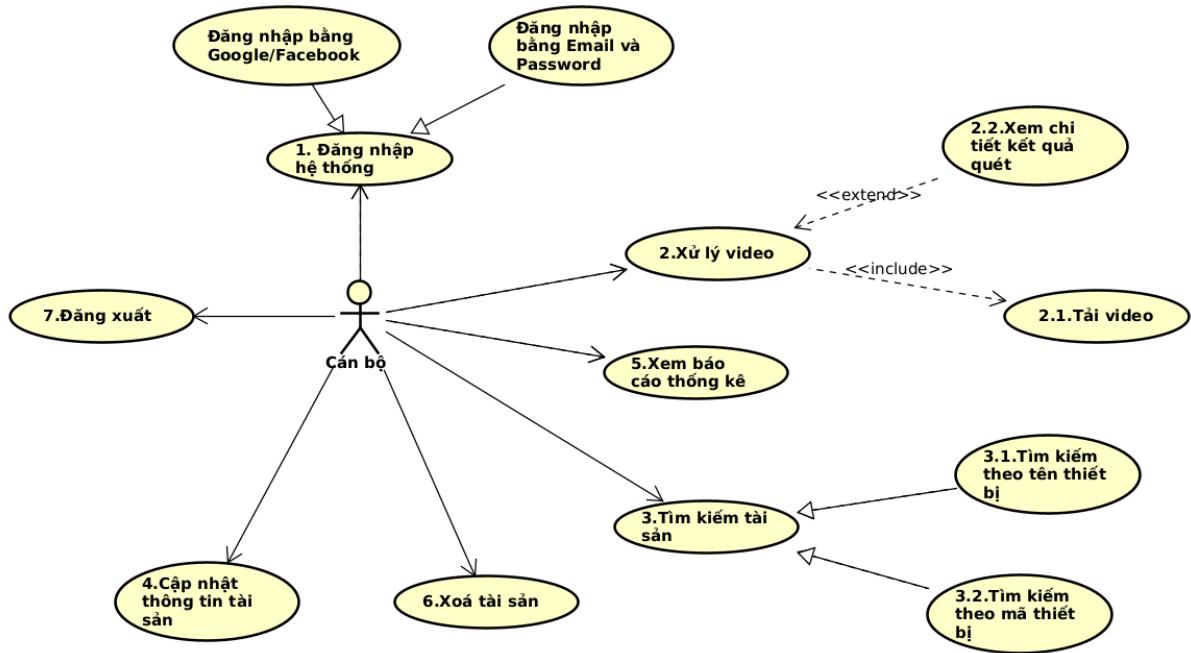
Nhóm Use-case của Cán bộ:

1. Đăng nhập hệ thống.
2. Xử lý video: Tải video, xem chi tiết kết quả,xem báo cáo thống kê.
3. Quản lý tài sản: Cập nhật thông tin tài sản,xoá tài sản,tìm kiếm tài sản.

Nhóm Use-case của AI Service:

1. Nhận dữ liệu video (Receive Data).
2. Nhận diện thiết bị (Detect Objects).
3. Gửi kết quả nhận diện (Send Results).

3.1.3. Biểu đồ use-case Cán bộ (Staff)



Hình 1. Biểu đồ usecase cho Cán bộ

Dưới đây là các bảng đặc tả chi tiết cho các Use-case quan trọng

Tên Use-case	Đăng nhập hệ thống
Tác nhân	Cán bộ
Mục đích	Truy cập vào hệ thống để thực hiện các quyền quản lý.
Luồng sự kiện chính	1. Cán bộ truy cập trang web. 2. Hệ thống hiển thị Form đăng nhập. 3. Cán bộ nhập Email và Mật khẩu. 4. Hệ thống xác thực thông tin với CSDL.

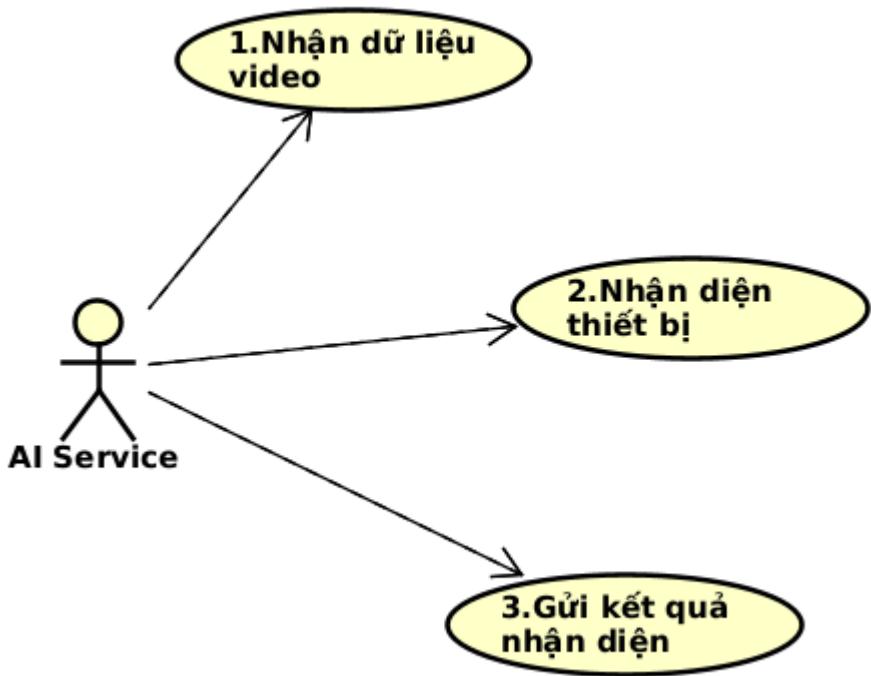
	5. Hệ thống cấp Access Token và chuyển hướng vào Dashboard.
Luồng ngoại lệ	<ul style="list-style-type: none"> - Sai thông tin: Hệ thống báo lỗi và yêu cầu nhập lại. - Token hết hạn: Hệ thống tự động logout hoặc refresh token.

Tên Use-case	Thêm mới tài sản
Tác nhân	Cán bộ
Mục đích	Nhập liệu thủ công một thiết bị mới vào kho.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Cán bộ nhấn nút "Thêm mới". 2. Nhập thông tin: Tên, Loại, Vị trí, Giá trị. 3. Nhấn "Lưu". 4. Hệ thống validate và lưu vào Database. 5. Cập nhật lại danh sách hiển thị.

Tên Use-case	Tải lên video kiểm kê (Upload Scan)
---------------------	--

Tác nhân	Cán bộ
Mục đích	Cung cấp dữ liệu đầu vào (video quay phòng học) để AI xử lý.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Cán bộ chọn chức năng "Quét AI". 2. Hệ thống hiển thị màn hình Upload. 3. Cán bộ chọn file Video và chọn Vị trí (Giảng đường/Phòng). 4. Cán bộ nhấn "Bắt đầu". 5. Hệ thống upload file và tạo bản ghi trạng thái "Processing". 6. Hệ thống thông báo thành công.
Luồng ngoại lệ	<ul style="list-style-type: none"> - File sai định dạng: Hệ thống từ chối và báo lỗi.

3.1.4. Biểu đồ Use-case AI Service



Hình 2. Biểu đồ usecase cho AI Service

Tên Use-case	Nhận dữ liệu video (Receive Data)
Tác nhân	AI Service
Mục đích	Lấy file video từ hệ thống lưu trữ để chuẩn bị xử lý.
Tiền điều kiện	Có bản ghi Scan mới với trạng thái "Processing".

Luồng sự kiện chính	<ol style="list-style-type: none"> 1. AI Service quét định kỳ hoặc nhận trigger từ Inventory Service. 2. AI Service tải file video từ đường dẫn được cung cấp. 3. Kiểm tra tính toàn vẹn của file video.
----------------------------	---

Tên Use-case	Thực hiện nhận diện (Detect Objects)
Tác nhân	AI Service
Mục đích	Phân tích hình ảnh để tìm kiếm và đếm thiết bị.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. AI Service cắt video thành các frame hình ảnh. 2. Load mô hình YOLLO v11 đã huấn luyện. 3. Chạy thuật toán nhận diện trên từng frame. 4. Lọc bỏ các kết quả trùng lặp hoặc độ tin cậy thấp. 5. Tổng hợp danh sách thiết bị cuối cùng.

Tên Use-case	Gửi kết quả nhận diện (Send Results)
---------------------	---

Tác nhân	AI Service
Mục đích	Cập nhật kết quả vào cơ sở dữ liệu để Cán bộ có thể xem được.
Luồng sự kiện chính	<p>1. AI Service đóng gói kết quả dạng JSON (Loại thiết bị, Tọa độ, Số lượng).</p> <p>2. Gọi API PUT /scans/:id của Inventory Service.</p> <p>3. Inventory Service cập nhật trạng thái bản ghi thành "Completed".</p> <p>4. Kết thúc phiên xử lý.</p>

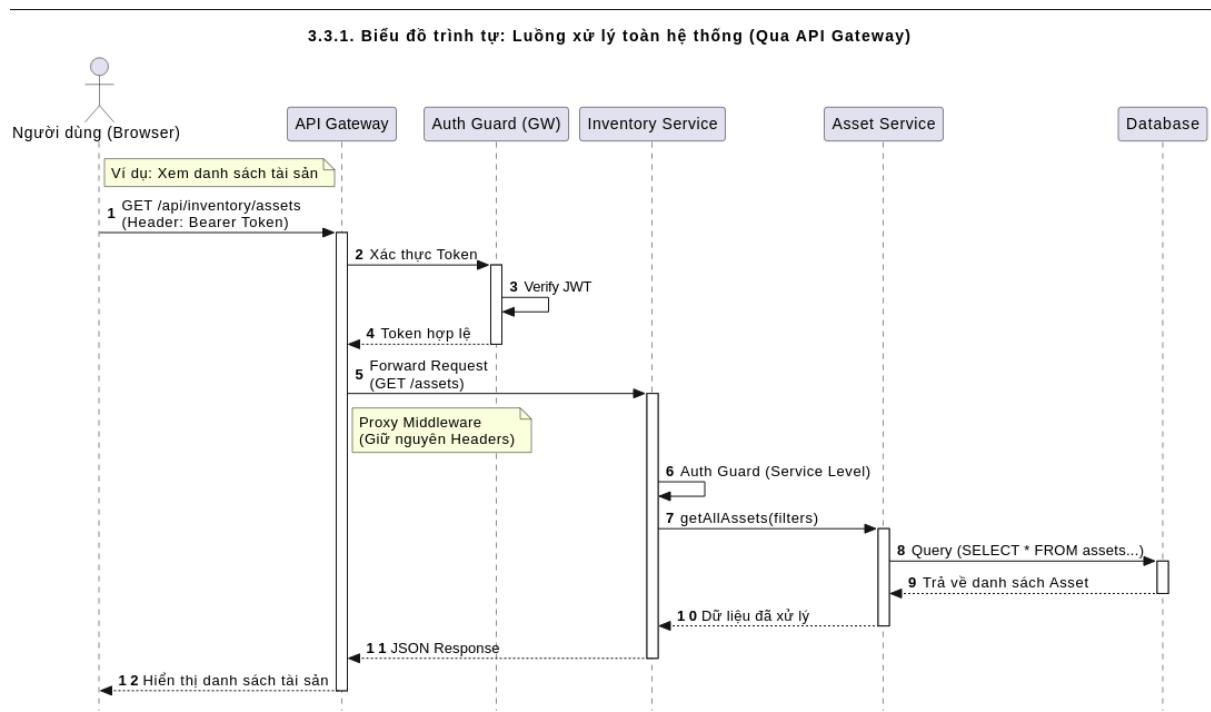
3.2. BIỂU ĐỒ HOẠT ĐỘNG

Mô tả luồng hoạt động "Quy trình Quét AI":

1. Bắt đầu.
2. Cán bộ chọn chức năng "Quét mới".
3. Hệ thống hiển thị form Upload.
4. Cán bộ chọn File và Vị trí (Location).
5. Cán bộ nhấn "Upload".
6. Hệ thống kiểm tra định dạng file?
 - Sai: Thông báo lỗi -> Quay lại B3.
 - Đúng: Lưu file -> Chuyển trạng thái "Processing".
7. AI Service đọc file -> Phân tích từng frame -> Đếm vật thể.
8. AI Service trả kết quả JSON.
9. Hệ thống cập nhật trạng thái "Completed".
10. Kết thúc.

3.3. BIỂU ĐỒ TRÌNH TỰ

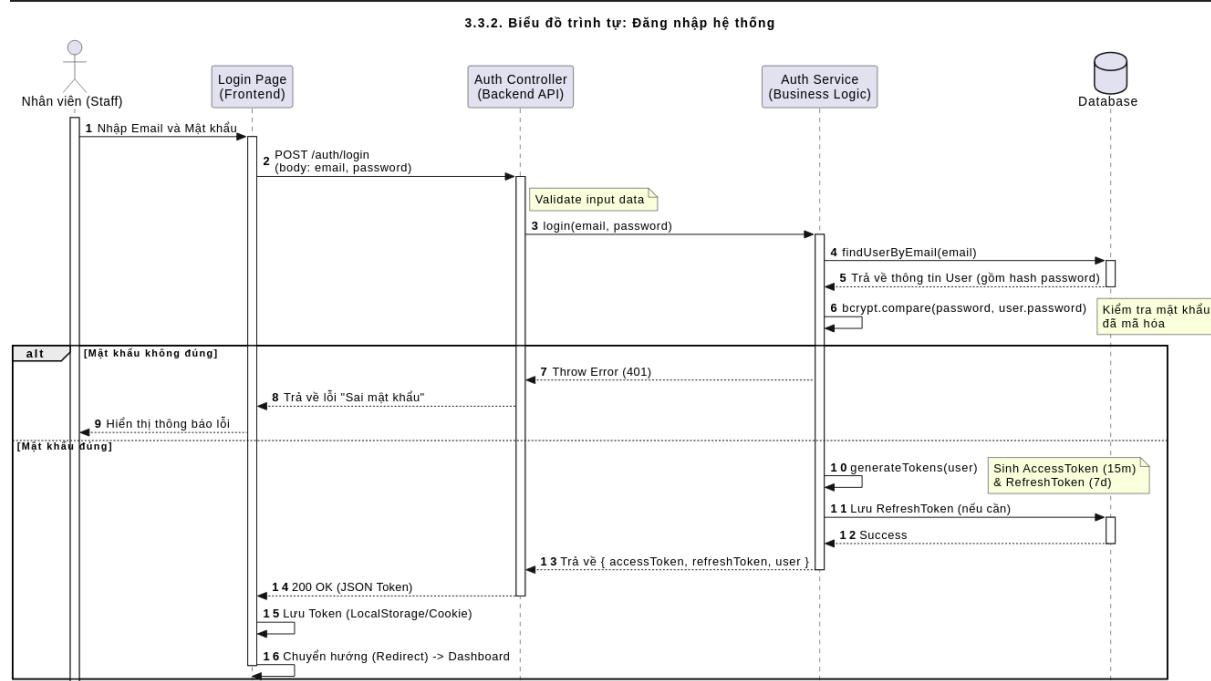
3.3.1. Biểu đồ trình tự ca sử dụng toàn bộ hệ thống



Hình 3. Biểu đồ trình tự Luồng xử lý API Gateway

Mô tả sự tương tác giữa: **Người dùng (Browser) -> API Gateway -> Inventory Service -> Database.**

3.3.2. Biểu đồ trình tự ca sử dụng Đăng nhập (Login)



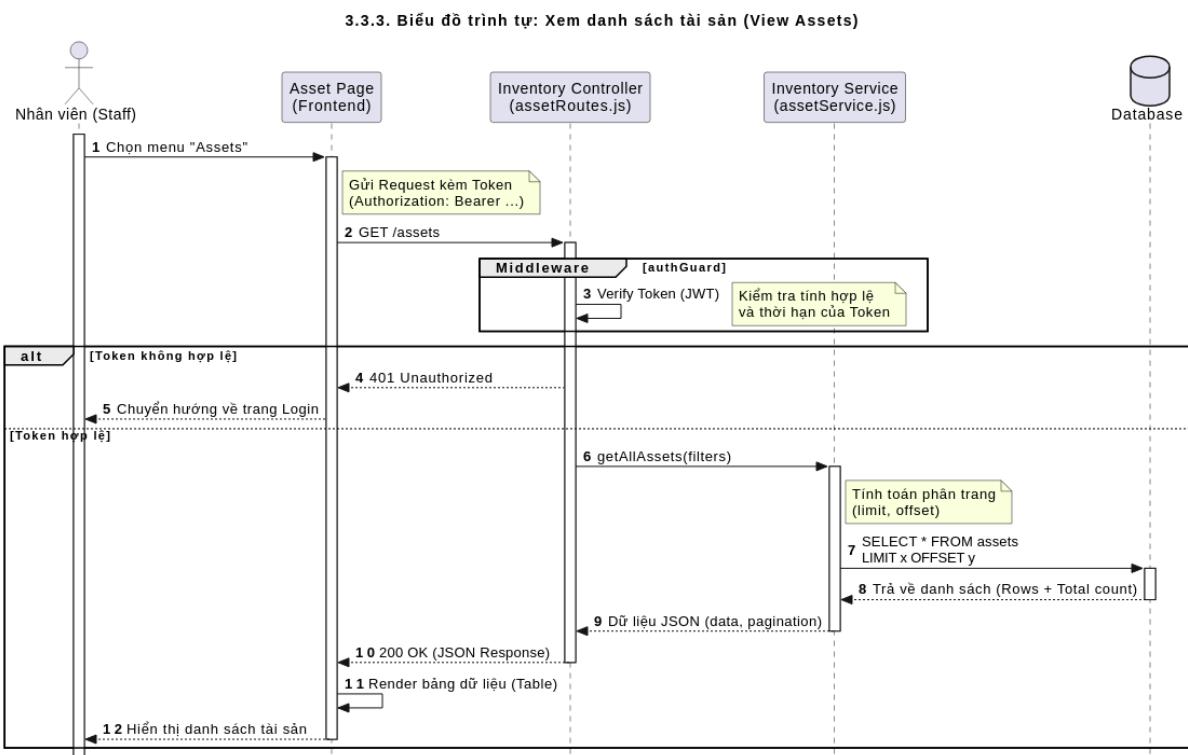
Hình 4. Biểu đồ trình tự Đăng nhập hệ thống

- **Tác nhân:** Staff.

- **Đối tượng:** Login Page, Auth Controller, Auth Service, Database.
- **Luồng xử lý:**

1. Staff nhập Email/Pass tại Login Page.
2. Page gửi POST /login tới Auth Controller.
3. Auth Controller gọi `verifyUser()` tại Auth Service.
4. Auth Service truy vấn bảng `User` trong Database.
5. Database trả về thông tin user.
6. Auth Service so khớp mật khẩu (Bcrypt).
7. Nếu đúng, sinh `AccessToken` và `RefreshToken`.
8. Trả về Token cho Login Page.
9. Login Page lưu Token và chuyển hướng sang Dashboard.

3.3.3. Biểu đồ trình tự ca sử dụng Xem danh sách tài sản

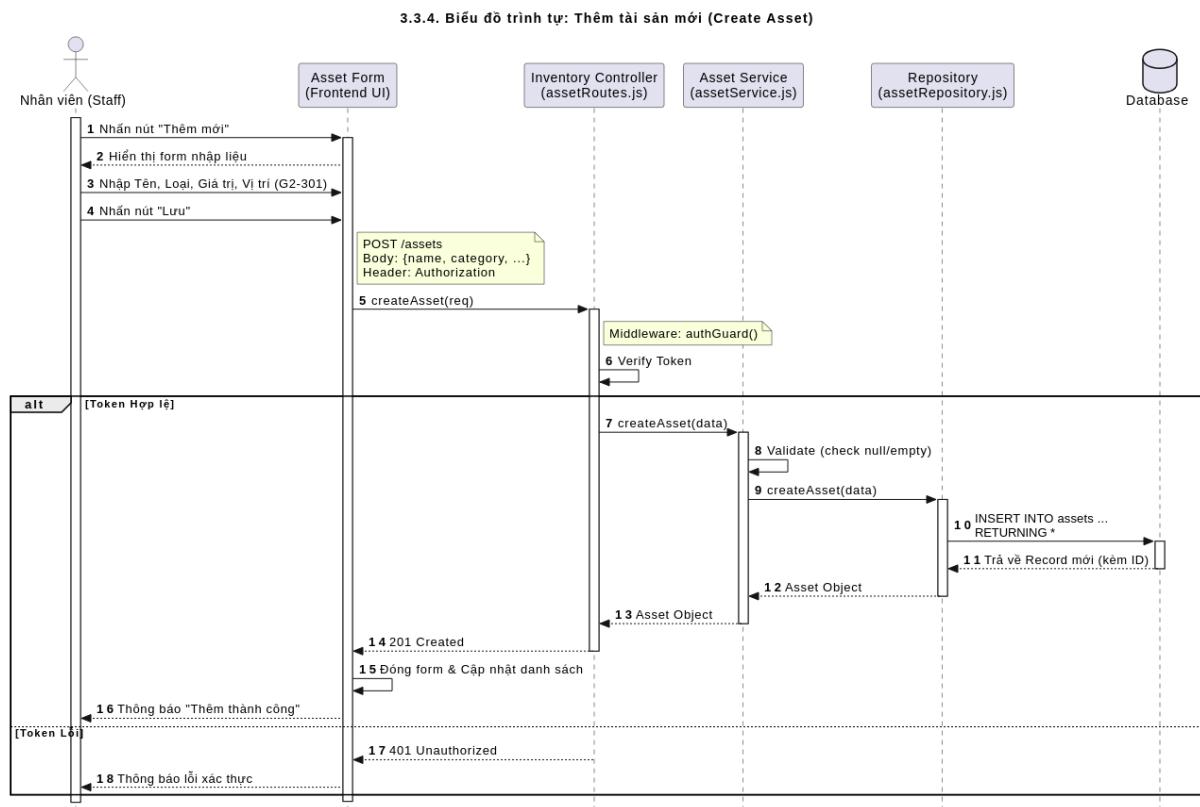


Hình 5. Biểu đồ trình tự Xem danh sách tài sản

- **Tác nhân:** Staff.
- **Luồng xử lý:**

1. Staff chọn menu "Assets".
2. Asset Page gửi GET /assets (kèm token) tới Inventory Service.
3. Inventory Service xác thực Token.
4. Inventory Service truy vấn bảng `Assets` (Select *).
5. Database trả về danh sách.
6. Service trả JSON về Frontend.
7. Asset Page hiển thị bảng dữ liệu.

3.3.4. Biểu đồ trình tự ca sử dụng Thêm tài sản mới



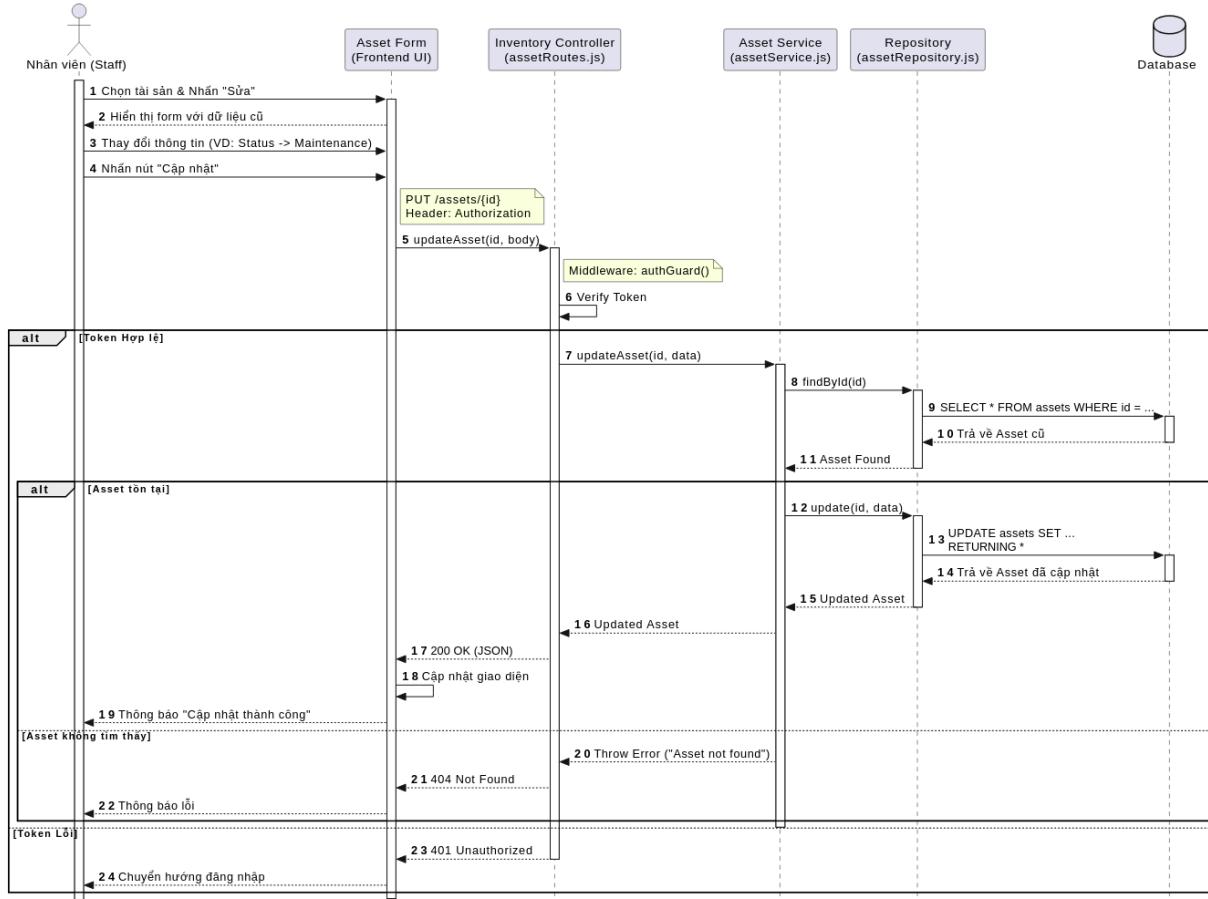
Hình 6. Biểu đồ trình tự Thêm tài sản mới

- **Tác nhân:** Staff.
- **Luồng xử lý:**

1. Staff nhấn "Thêm mới" trên giao diện.
2. Nhập thông tin: Tên, Loại, Giá trị, Vị trí (G2-301).
3. Nhấn "Lưu".
4. Frontend gửi POST /assets tới Inventory Service.
5. Service validate dữ liệu (không được để trống).
6. Service gọi lệnh Insert vào Database.
7. Database trả về ID tài sản mới tạo.
8. Hệ thống thông báo "Thêm thành công".

3.3.5. Biểu đồ trình tự ca sử dụng Cập nhật tài sản

3.3.5. Biểu đồ trình tự: Cập nhật tài sản (Update Asset)



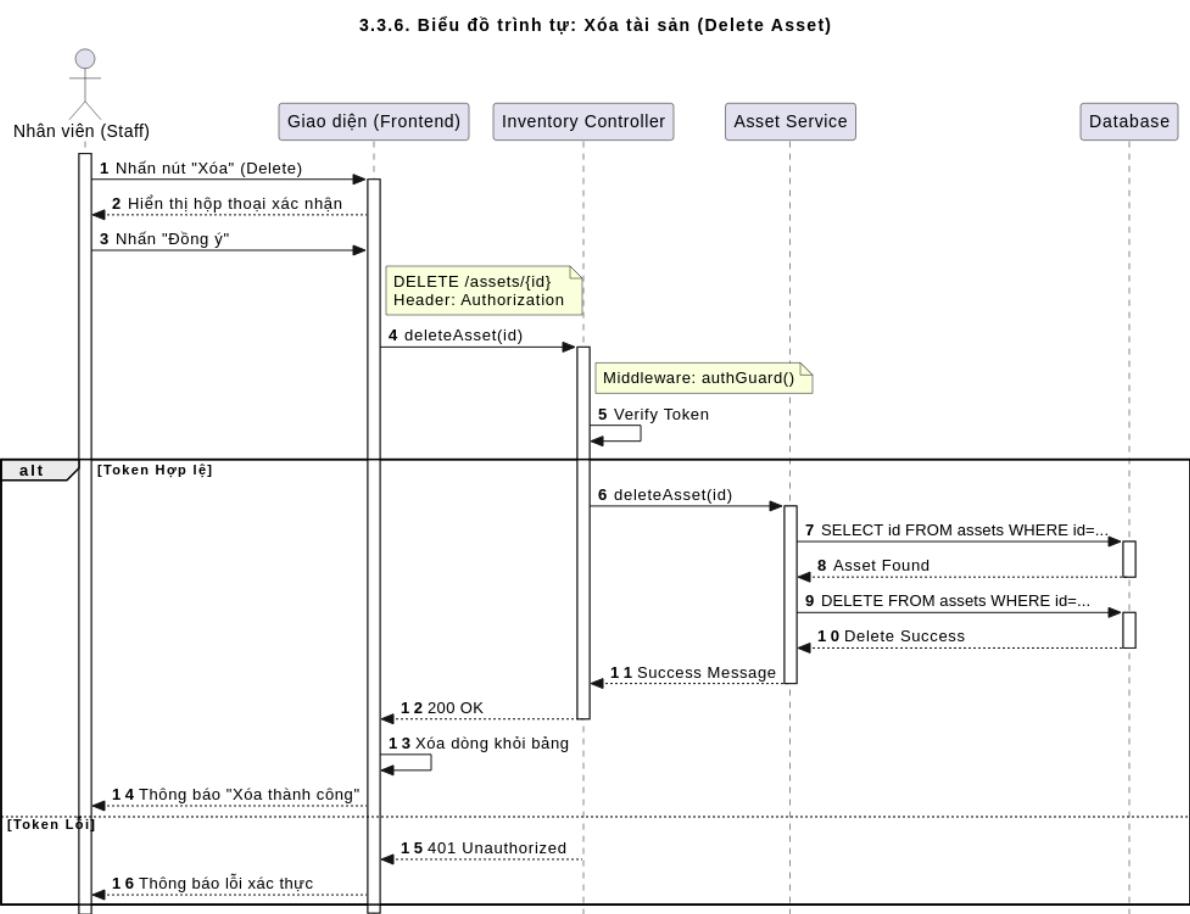
Hình 7. Biểu đồ trình tự Cập nhật tài sản

Tác nhân: Staff.

Luồng xử lý:

1. Staff nhấn nút "Sửa" (icon bút chì) trên dòng tài sản cần cập nhật.
2. Hệ thống hiển thị form với thông tin hiện tại (Tên, Loại, Trạng thái...).
3. Staff thay đổi thông tin (ví dụ: đổi Trạng thái sang "Maintenance").
4. Staff nhấn "Lưu".
5. Frontend gửi PUT /assets/{id} tới Inventory Service.
6. Service xác thực Token và kiểm tra tài sản có tồn tại không.
7. Service gọi lệnh Update vào Database.
8. Database trả về thông tin tài sản đã cập nhật.
9. Hệ thống thông báo "Cập nhật thành công".

3.3.6. Biểu đồ trình tự ca sử dụng Xóa tài sản



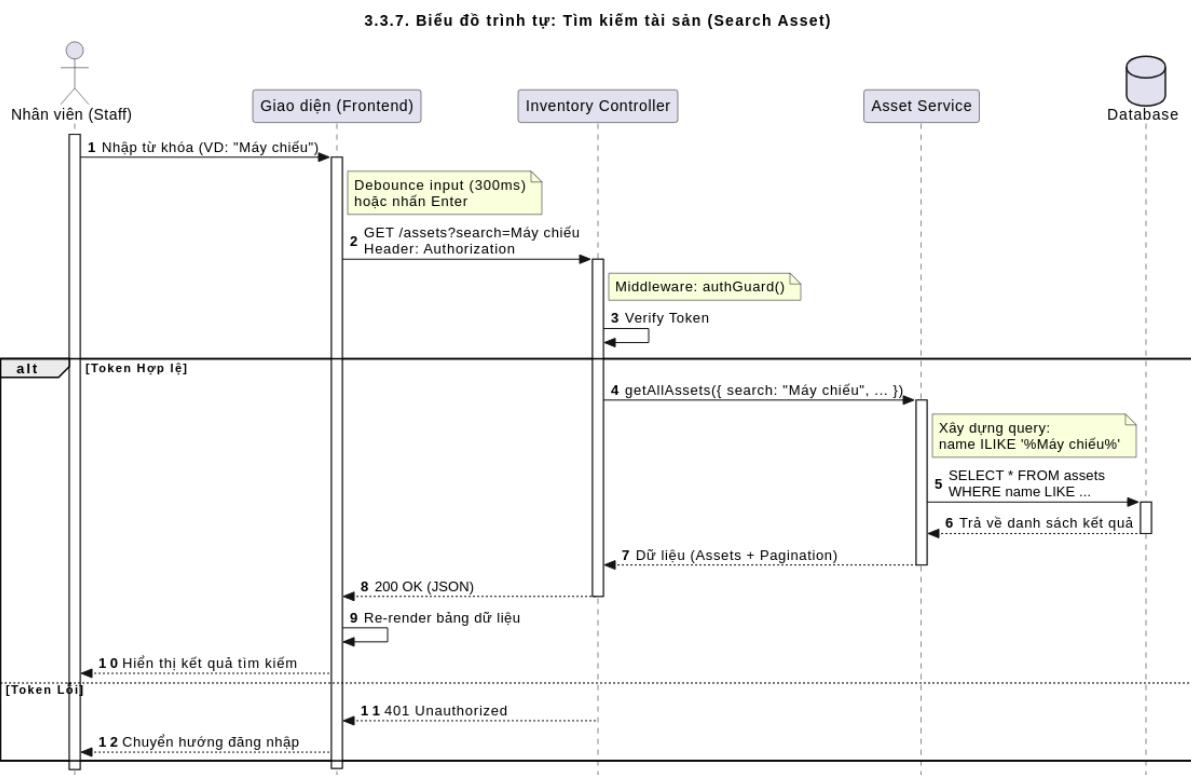
Hình 8. Biểu đồ trình tự Xóa tài sản

Tác nhân: Staff.

Luồng xử lý:

1. Staff nhấp nút "Xóa" (icon thùng rác) trên dòng tài sản cần xóa.
2. Hệ thống hiển thị cảnh báo "Bạn có chắc chắn muốn xóa?".
3. Staff nhấp "Đồng ý".
4. Frontend gửi DELETE /assets/{id} tới Inventory Service.
5. Service xác thực Token và kiểm tra tài sản có tồn tại không.
6. Service gọi lệnh Delete trong Database.
7. Database xóa bản ghi và trả về xác nhận.
8. Hệ thống thông báo "Xóa thành công" và tải lại danh sách.

3.3.7. Biểu đồ trình tự ca sử dụng Tìm kiếm tài sản



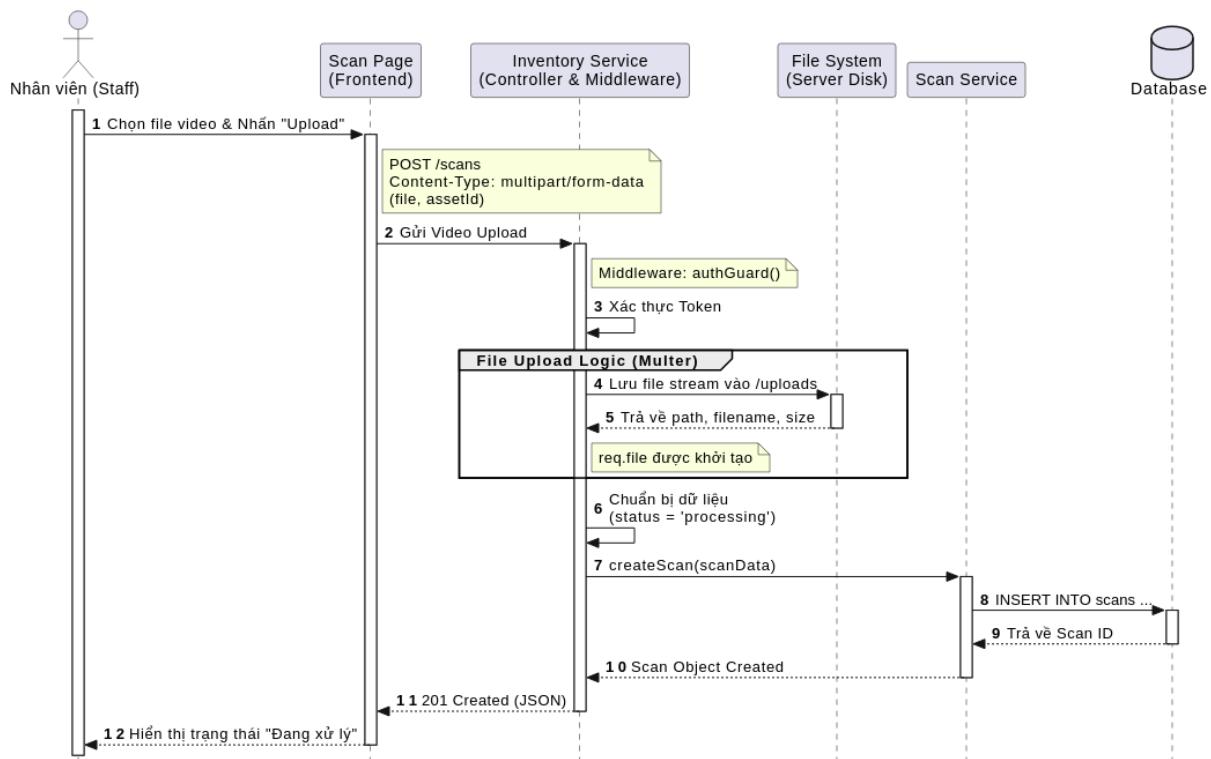
Hình 9. Biểu đồ trình tự Tìm kiếm tài sản

- **Tác nhân:** Staff.
- **Luồng xử lý:**

1. Staff nhập từ khóa vào ô Search (ví dụ: "Máy chiếu").
2. Frontend gửi request GET /assets?search=Máy chiếu.
3. Inventory Service tạo câu truy vấn SQL **LIKE %Máy chiếu%**.
4. Database trả về kết quả lọc.
5. Hệ thống hiển thị kết quả lọc trên Grid.

3.3.8. Biểu đồ trình tự ca sử dụng Upload Video (Gửi dữ liệu quét)

3.3.8. Biểu đồ trình tự: Upload Video Scan

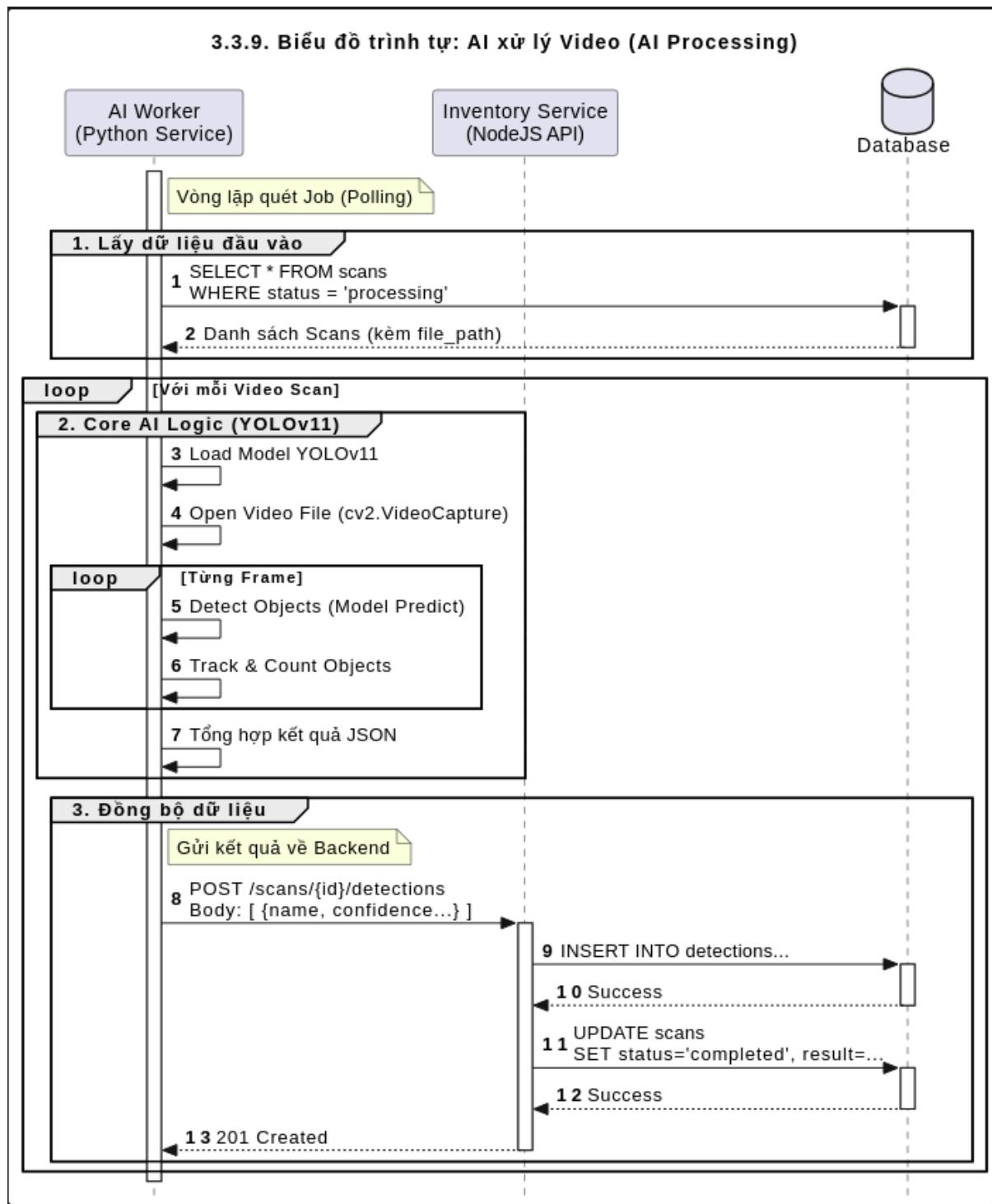


Hình 10. Biểu đồ trình tự Upload Video Scan

- **Tác nhân:** Staff.
- **Đối tượng:** Scan Page, Inventory Service, Scan Repository, File System.
- **Luồng xử lý:**

1. Staff chọn file video và nhấn Upload.
2. Frontend gửi Multipart form-data tới Inventory Service.
3. Service lưu file vào thư mục `/uploads` trên Server (File System).
4. Service tạo bản ghi mới trong bảng `Scans` với status='processing'.
5. Trả về thông báo "Đang xử lý" cho người dùng.

3.3.9. Biểu đồ trình tự ca sử dụng AI xử lý

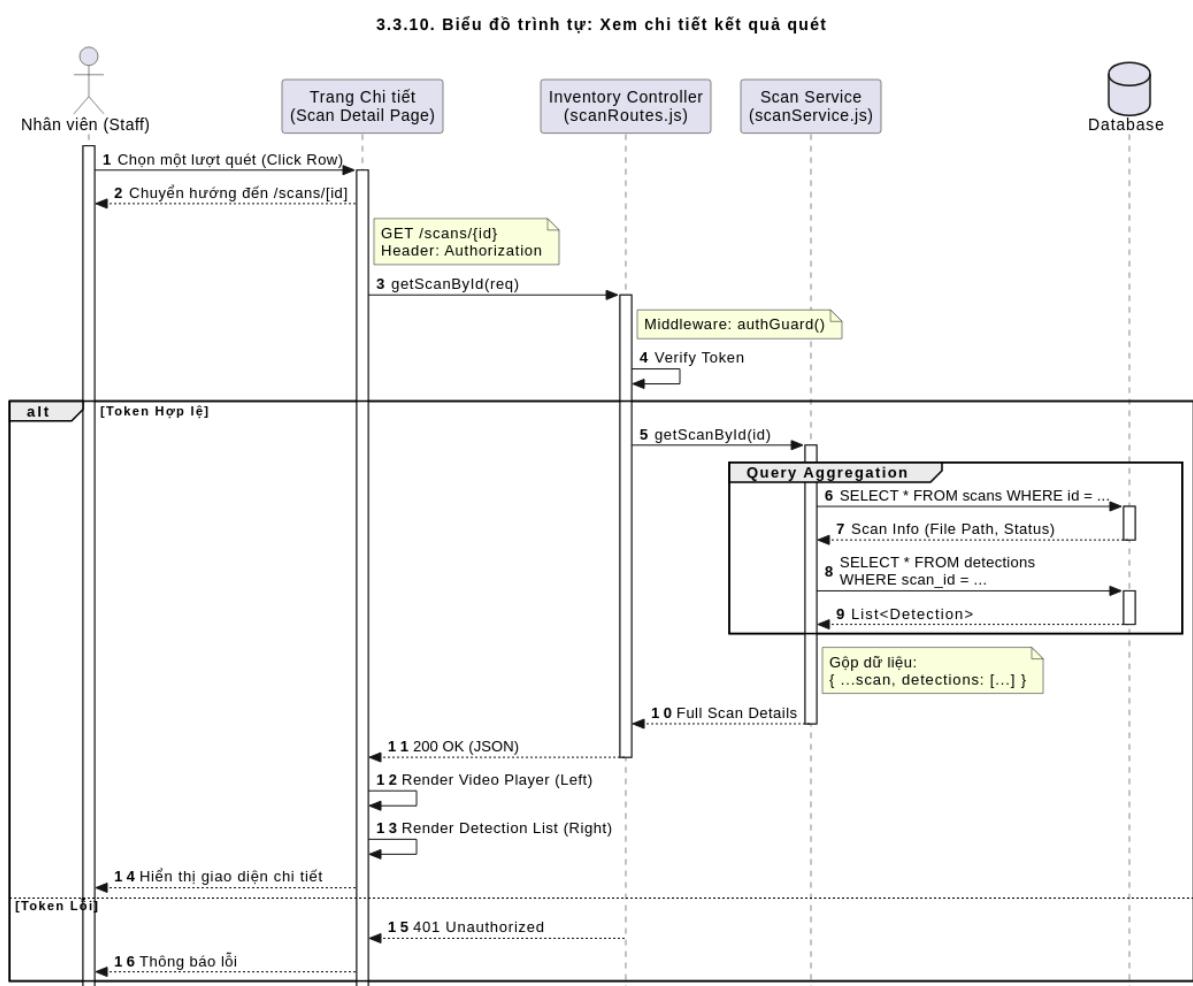


Hình 11. Biểu đồ trình tự: AI xử lý video

- **Tác nhân:** AI Service (Hệ thống ngầm).
- **Luồng xử lý:**
 1. AI Service quét DB tìm các bản ghi có status='processing'.
 2. AI Service lấy đường dẫn file video.
 3. Load mô hình YOLOv11.
 4. Thực hiện Detect trên từng frame.

5. Tổng hợp kết quả (đếm số lượng).
6. AI Service gọi API `updateScanResult` của Inventory Service.
7. Inventory Service cập nhật DB (status='completed', result_data=JSON).

3.3.10. Biểu đồ trình tự ca sử dụng Xem chi tiết kết quả quét



Hình 12. Biểu đồ trình tự: Xem chi tiết kết quả quét

- **Tác nhân:** Staff.

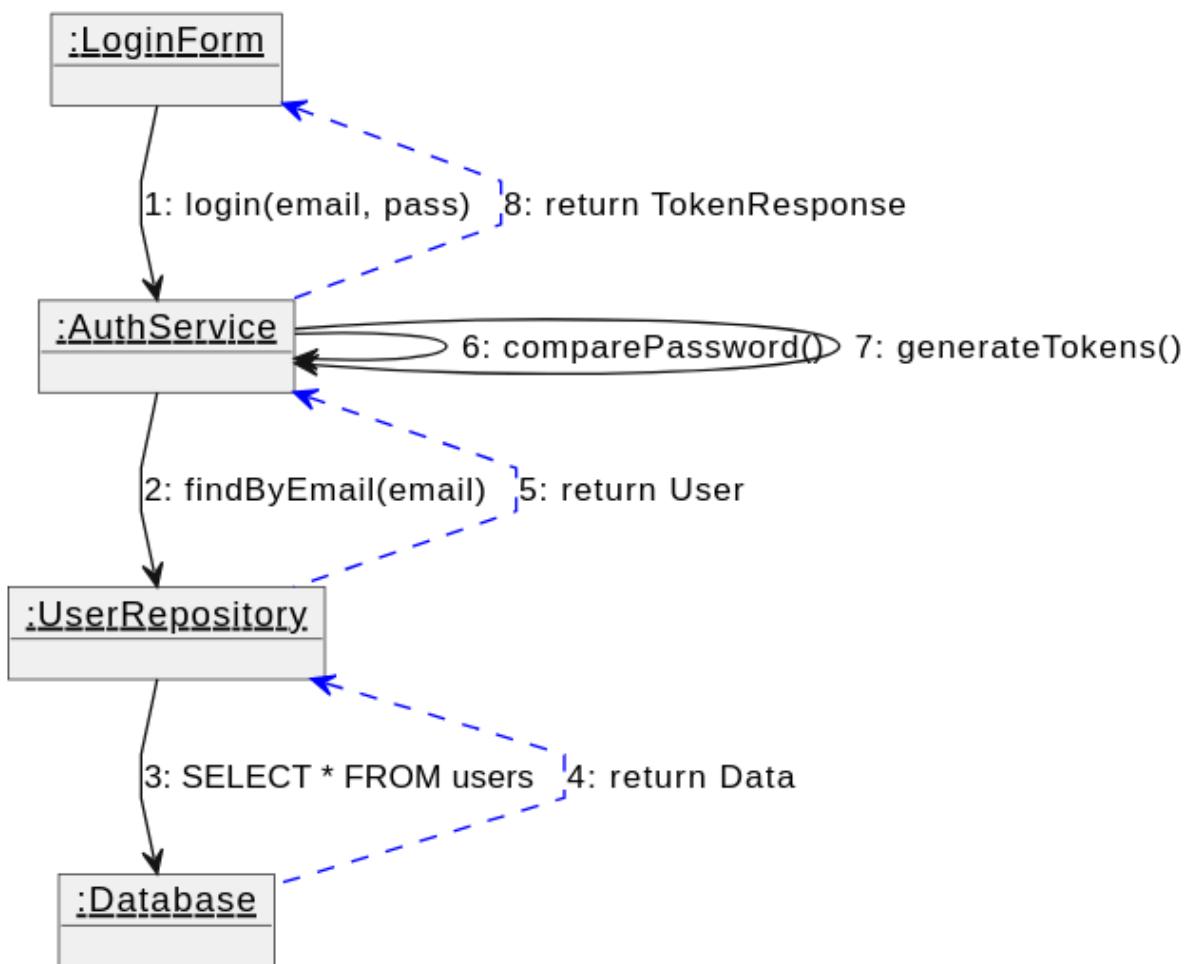
- **Luồng xử lý:**

1. Staff chọn một mục trong lịch sử quét.
2. Frontend gọi GET /scans/:id.
3. Service trả về thông tin chi tiết + danh sách vật thể nhận diện.
4. Frontend hiển thị Video bên trái và Danh sách thiết bị bên phải.

3.4. BIỂU ĐỒ CỘNG TÁC

3.4.1. Biểu đồ cộng tác ca sử dụng Đăng nhập

Biểu đồ cộng tác: Đăng nhập

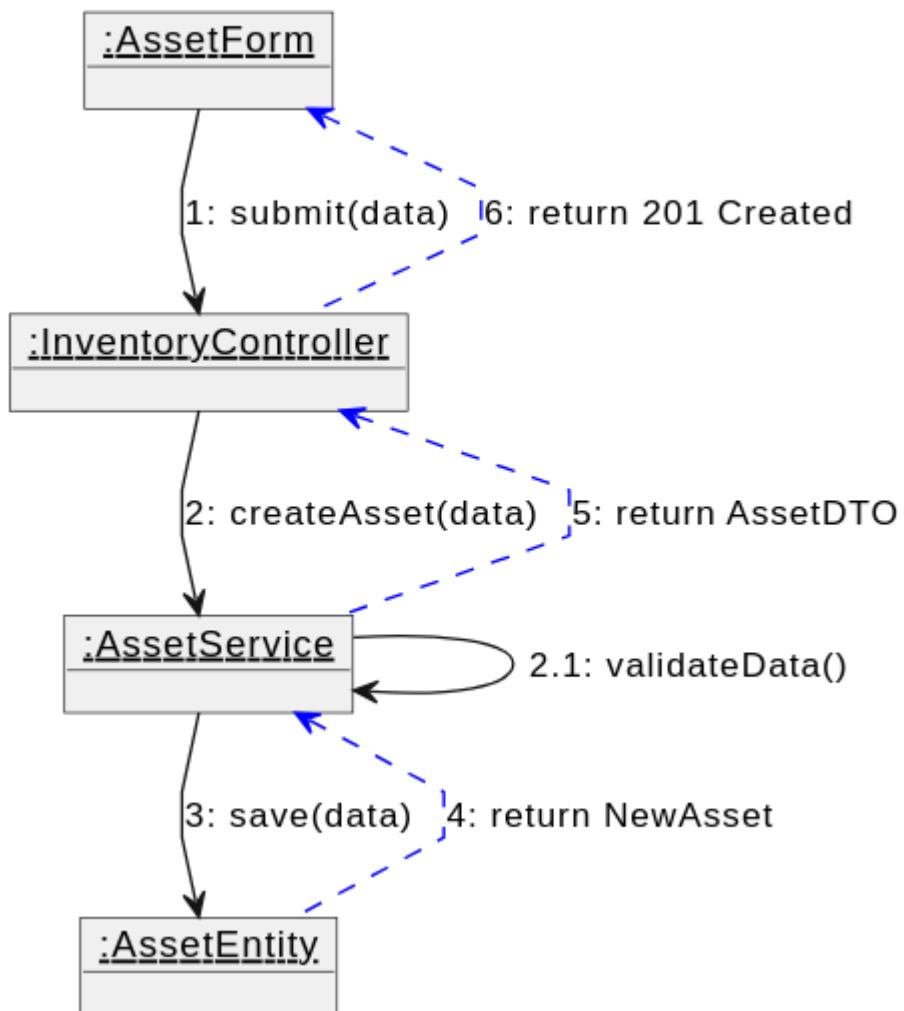


Hình 13. Biểu đồ cộng tác: Đăng nhập

Mô tả liên kết giữa: `LoginForm` <--> `AuthService` <--> `UserRepository` <--> `Database`.

3.4.2. Biểu đồ cộng tác ca sử dụng Thêm tài sản

3.4.2. Biểu đồ cộng tác: Thêm tài sản

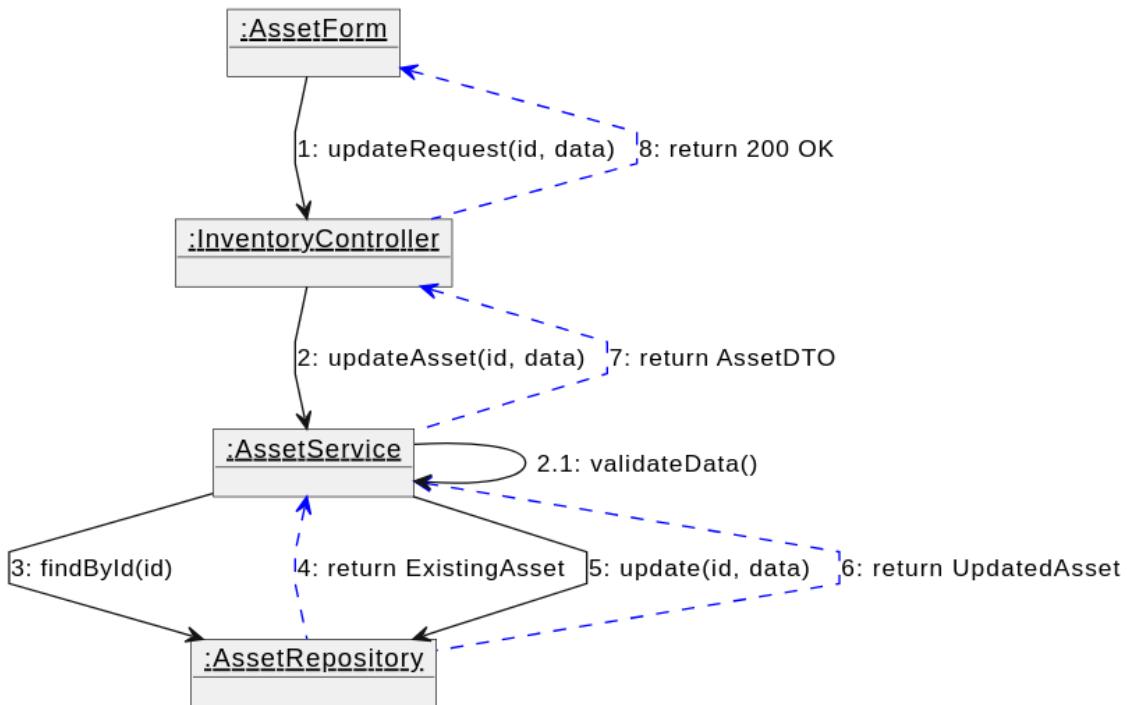


Hình 14. Biểu đồ cộng tác: Thêm tài sản

Mô tả liên kết giữa: `AssetForm` <--> `InventoryController` <--> `AssetService` <--> `AssetEntity`.

3.4.3. Biểu đồ cộng tác ca sử dụng Cập nhật tài sản (Update Asset)

3.4.3. Biểu đồ cộng tác: Cập nhật tài sản



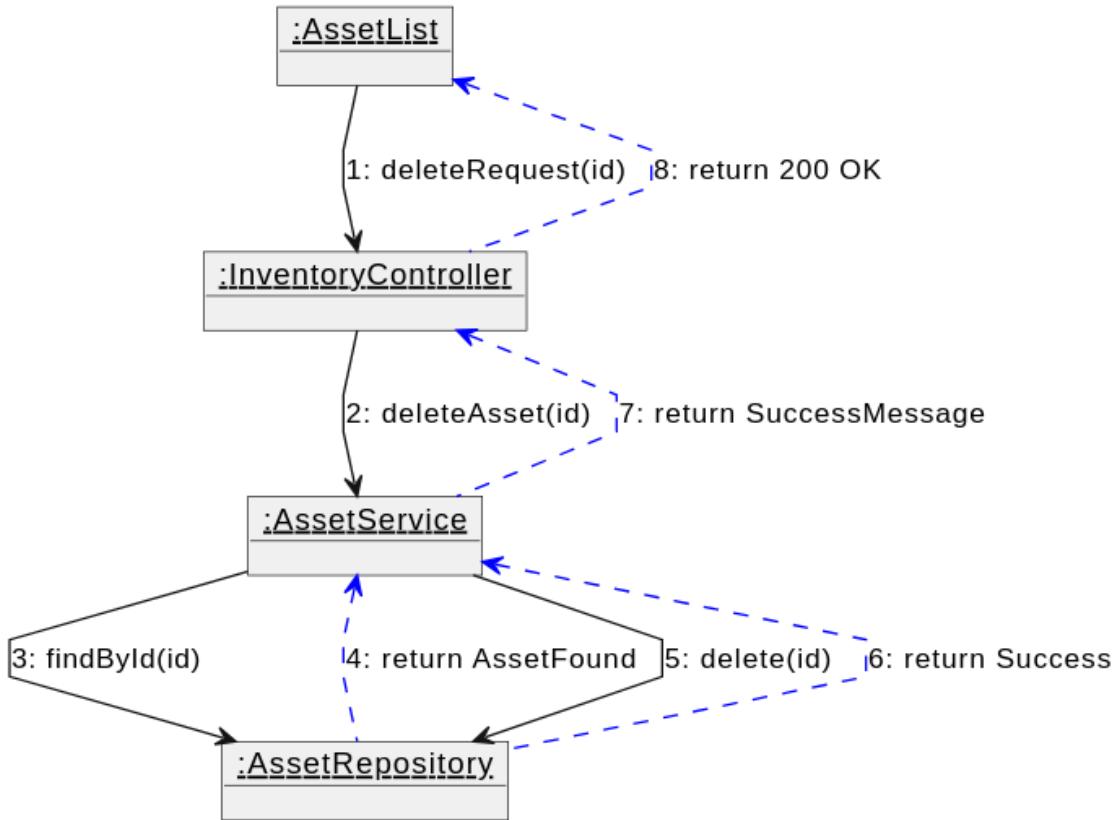
Hình 15. Biểu đồ cộng tác: Cập nhật tài sản

Các đối tượng:

- :AssetForm: Giao diện sửa đổi.
- :InventoryController: Tiếp nhận request PUT.
- :AssetService: Logic nghiệp vụ (Kiểm tra tồn tại + Cập nhật).
- :AssetRepository: Thao tác DB.

3.4.4. Biểu đồ cộng tác ca sử dụng Xóa tài sản (Delete Asset)

3.4.4. Biểu đồ cộng tác: Xóa tài sản



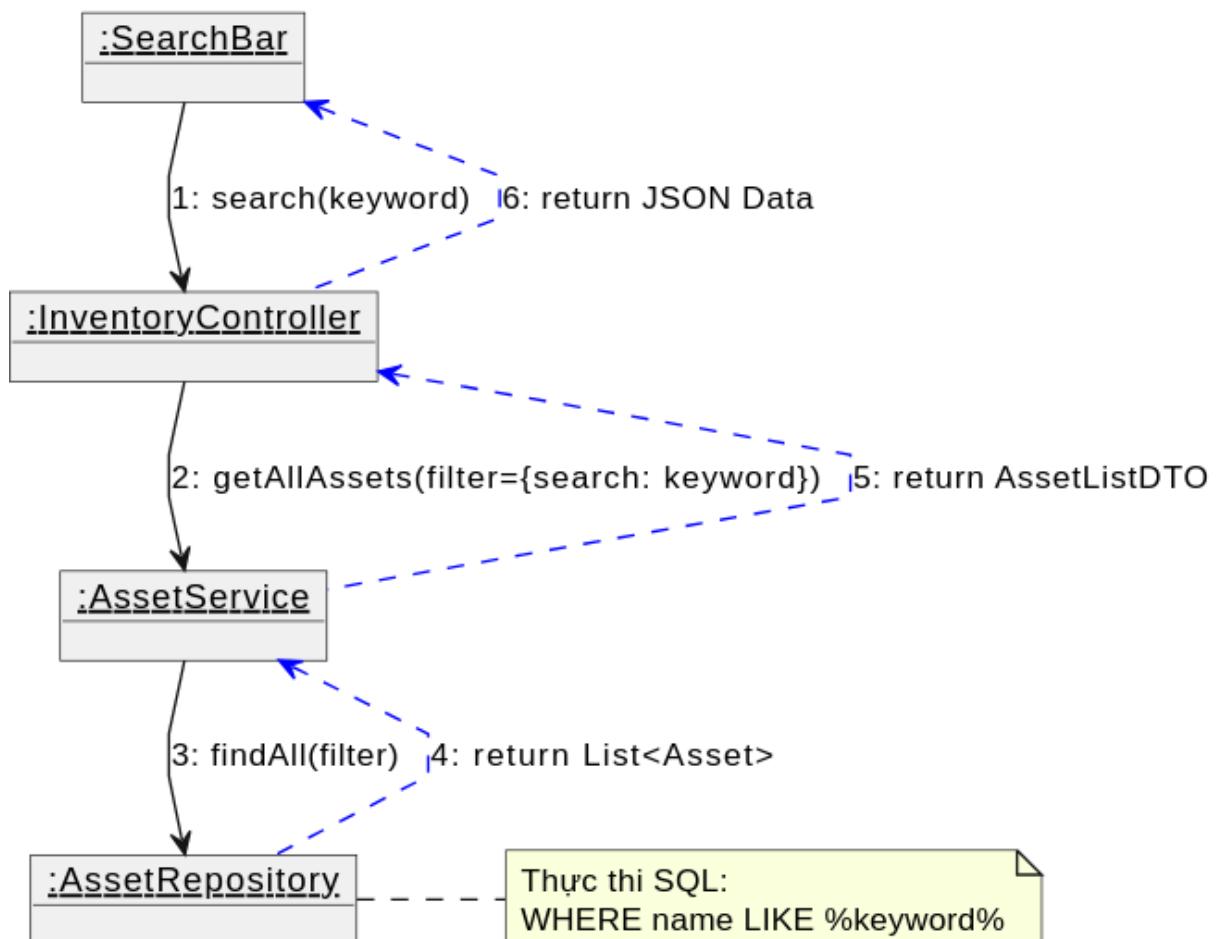
Hình 16. Biểu đồ cộng tác: Xóa tài sản

Các đối tượng:

- :AssetList: Giao diện danh sách (nơi chứa nút xóa).
- :InventoryController: Tiếp nhận request DELETE.
- :AssetService: Logic kiểm tra và xóa.
- :AssetRepository: Thao tác DB

3.4.5. Biểu đồ cộng tác ca sử dụng Tìm kiếm tài sản (Search Asset)

3.4.5. Biểu đồ cộng tác: Tìm kiếm tài sản



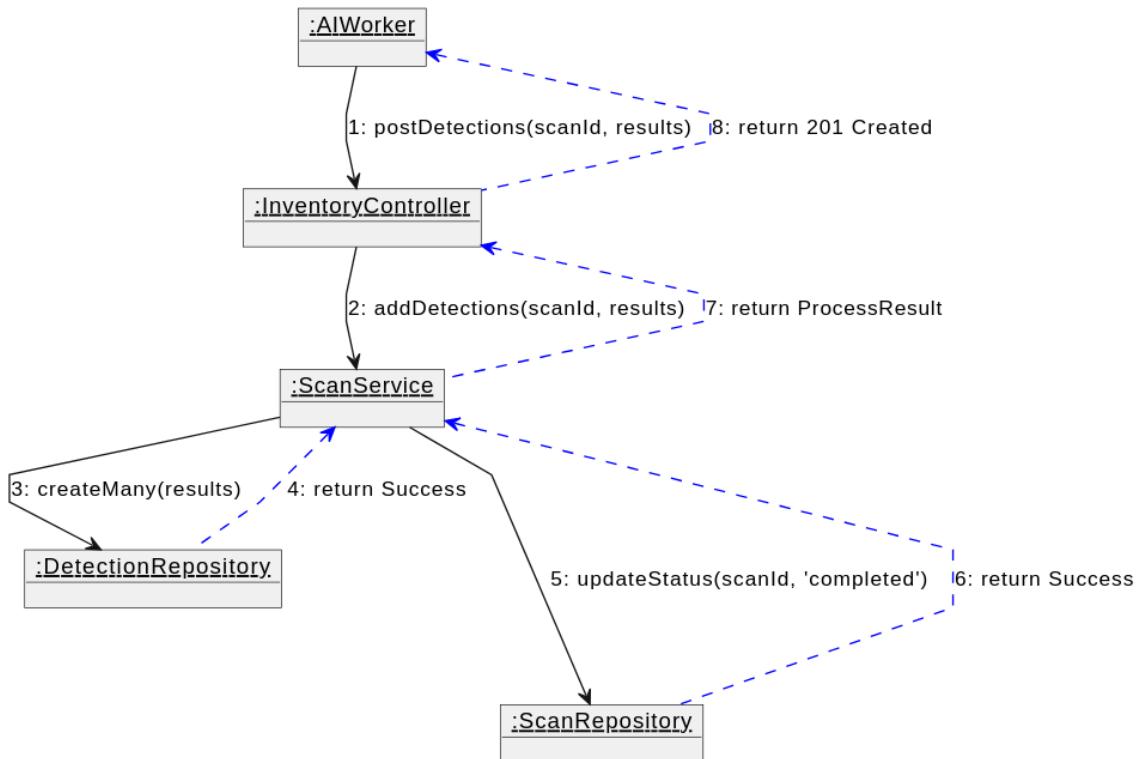
Hình 17. Biểu đồ cộng tác: Tìm kiếm tài sản

Các đối tượng:

- `:SearchBar`: Thanh tìm kiếm trên giao diện.
- `:InventoryController`: Tiếp nhận request GET với query param.
- `:AssetService`: Xử lý tham số lọc.
- `:AssetRepository`: Tạo câu truy vấn SQL (LIKE/ILIKE)

3.4.6. Biểu đồ cộng tác: Quy trình AI Xử lý & Trả kết quả (AI Processing)

3.4.6. Biểu đồ cộng tác: AI Cập nhật kết quả (Callback)



Hình 18. Biểu đồ cộng tác: AI cập nhật kết quả(Callback)

Các đối tượng:

- **:AIWorker**: Service chạy ngầm (Python).
- **:InventoryController**: API nhận kết quả (POST /scans/:id/detections).
- **:ScanService**: Logic cập nhật trạng thái Scan và lưu Detection.
- **:ScanRepository & :DetectionRepository**: Thao tác DB

3.5. BIỂU ĐỒ LỚP (CLASS DIAGRAM)

3.5.1. Mô tả

Hệ thống được thiết kế theo mô hình 3 lớp (3-Layer Architecture) kết hợp Microservices. Các lớp chính bao gồm Controller (giao tiếp API), Service (xử lý nghiệp vụ) và Repository (truy xuất dữ liệu).

3.5.2. Những lớp/đối tượng quan trọng

3.5.2.1. Lớp Asset (Tài sản)

- **Thuộc tính:**
 - id: UUID
 - name: String
 - category: String
 - location: String (Lưu trữ dạng "G2 - 301")
 - status: Enum (Active, Maintenance, Inactive)
 - value: Decimal
- **Phương thức:**
 - create()
 - updateInfo()
 - markAsMaintenance()

3.5.2.2. Lớp ScanSession (Phiên quét)

- **Thuộc tính:**
 - id: UUID
 - scanCode: String
 - imageUrl: String
 - scanStatus: Enum (Processing, Completed, Failed)
 - detectedItems: JSON
- **Phương thức:**
 - processVideo()
 - saveResult()

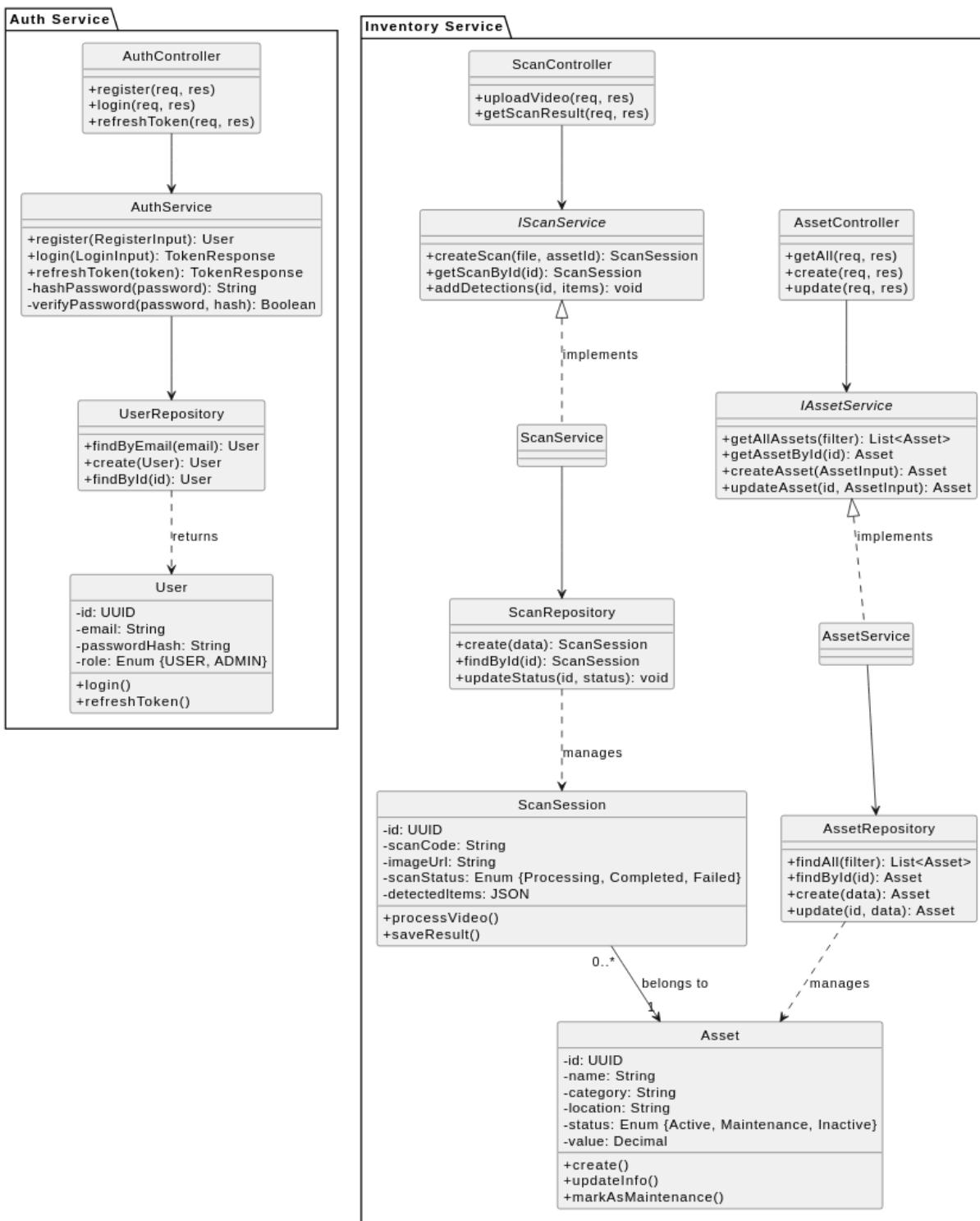
3.5.2.3. Lớp User (Người dùng)

- **Thuộc tính:**
 - id: UUID
 - email: String
 - passwordHash: String
 - role: String
- **Phương thức:**
 - login()
 - refreshToken()

3.5.3. Các lớp Service và Interface

- **IAssetService:** Định nghĩa các phương thức quản lý tài sản.
- **IScanService:** Định nghĩa các phương thức xử lý quét.
- **AuthService:** Xử lý logic đăng nhập, mã hóa mật khẩu.

3.5. Biểu đồ lớp: Kiến trúc 3 lớp & Microservices

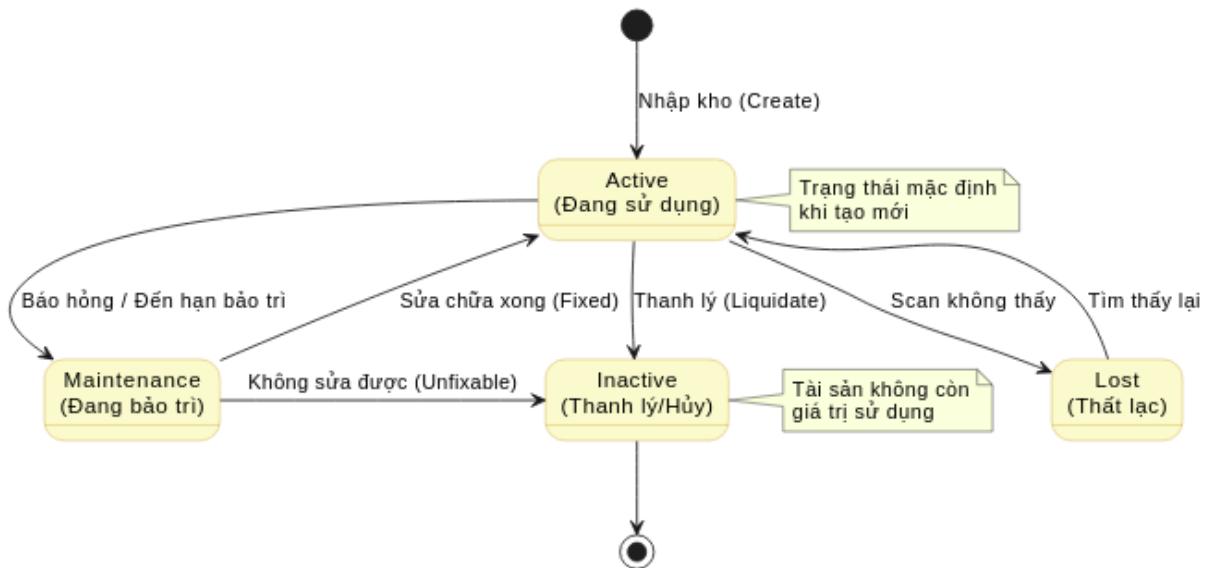


Hình 19. Biểu đồ lớp của hệ thống

3.6. BIỂU ĐỒ TRẠNG THÁI (STATE DIAGRAM)

3.6.1. Biểu đồ trạng thái của Tài sản (Asset)

3.6.1. Biểu đồ trạng thái: Vòng đời Tài sản (Asset)



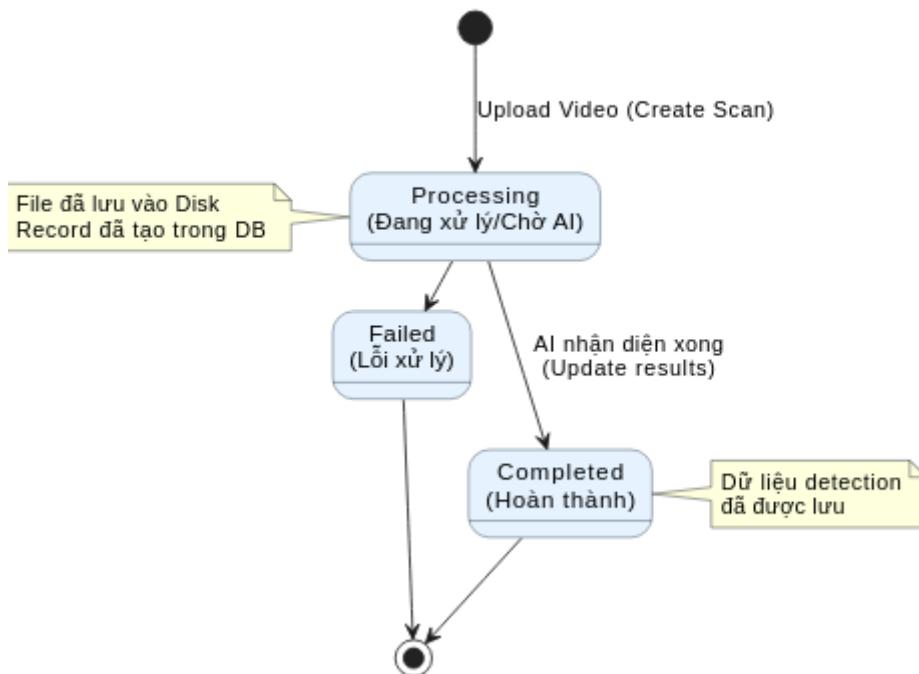
Hình 20. Biểu đồ trạng thái: Vòng đời tài sản

Mô tả vòng đời của một tài sản trong hệ thống:

- **Start** -> **Active** (Mới nhập kho).
- **Active** -> **Maintenance** (Khi báo hỏng/cần sửa).
- **Maintenance** -> **Active** (Sửa xong).
- **Maintenance** -> **Inactive** (Không sửa được/Thanh lý).
- **Active** -> **Inactive** (Thanh lý).
- **End**.

3.6.2. Biểu đồ trạng thái của Phiên quét (Scan Session)

3.6.2. Biểu đồ trạng thái: Phiên quét (Scan Session)



Hình 21. Biểu đồ trạng thái: Phiên quét

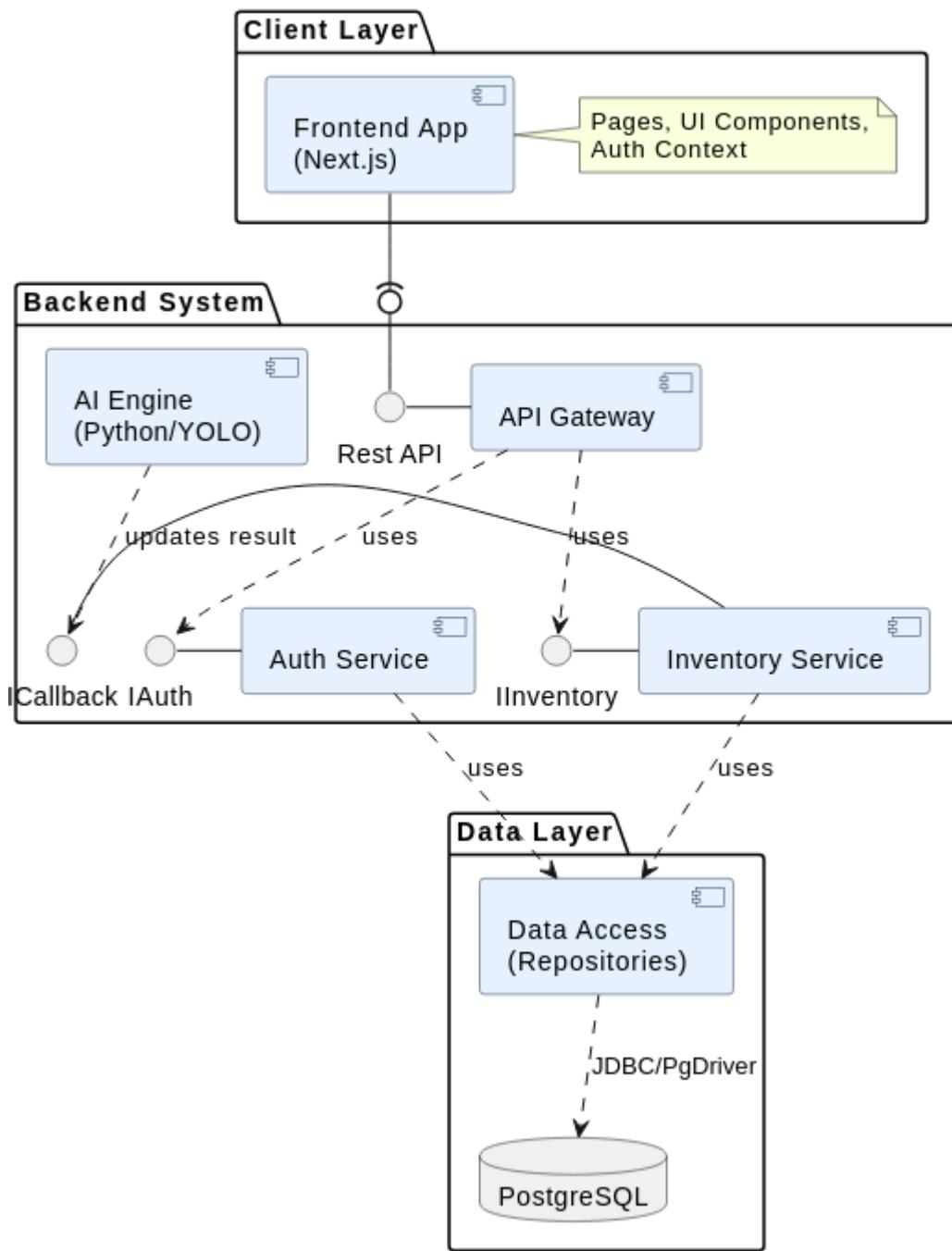
- **Start** -> **Processing** (Khi vừa upload video).
- **Processing** -> **Completed** (AI chạy xong và trả kết quả).
- **Processing** -> **Failed** (Nếu file lỗi hoặc AI crash).
- **End**.

3.7. BIỂU ĐỒ THÀNH PHẦN (COMPONENT DIAGRAM)

Hệ thống bao gồm các thành phần phần mềm (Component) chính:

1. **Frontend Component:** Chứa các module giao diện (Next.js Pages, Components).
2. **Auth Service Component:** Module xác thực, quản lý Token.
3. **Inventory Service Component:** Core logic, chứa các API quản lý tài sản.
4. **AI Engine Component:** Module xử lý ảnh (Python/YOLO).
5. **Data Access Component:** Các thư viện kết nối Database (pg library).

3.7. Biểu đồ thành phần (Component Diagram)



Hình 22. Biểu đồ thành phần

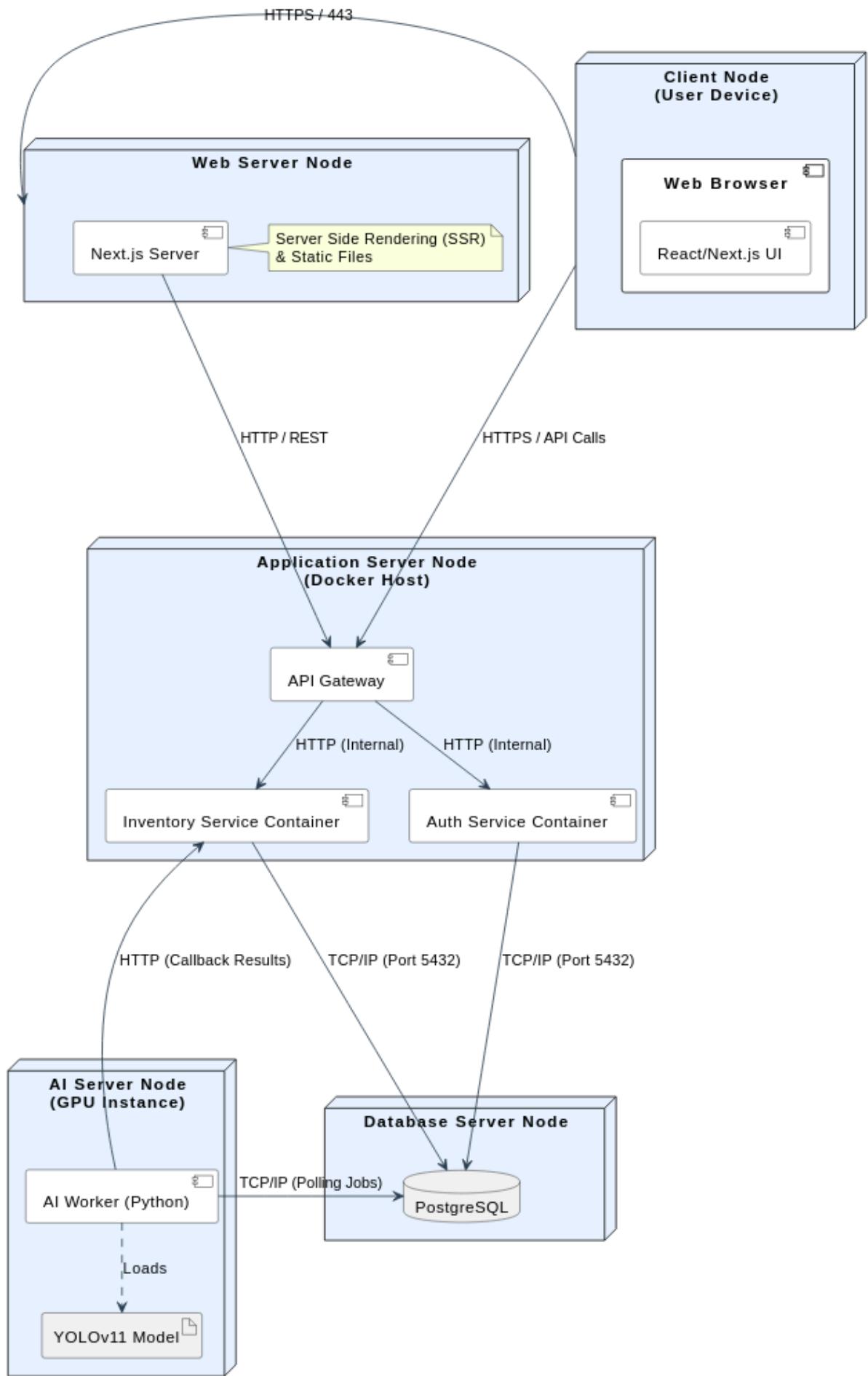
3.8. BIỂU ĐỒ TRIỀN KHAI (DEPLOYMENT DIAGRAM)

Mô tả cách hệ thống được cài đặt vật lý:

- **Client Node:** Máy tính/Điện thoại của cán bộ (Chạy trình duyệt Web).
- **Web Server Node:** Máy chủ chạy Frontend (Next.js).
- **Application Server Node:** Máy chủ chạy Backend API (Node.js Container).

- **AI Server Node:** Máy chủ có GPU chạy AI Service.
- **Database Server Node:** Máy chủ chạy PostgreSQL. (*Kết nối giữa các node là giao thức HTTP/HTTPS và TCP/IP*).

3.8. Biểu đồ triển khai hệ thống AI Inventory



Hình 23. Biểu đồ triển khai hệ thống

CHƯƠNG IV: KẾT QUẢ CÀI ĐẶT VÀ GIAO DIỆN

4.1. Môi trường cài đặt

- **Frontend:** Framework Next.js, thư viện UI Shaden.
- **Backend:** Node.js, Express Framework.
- **Database:** PostgreSQL trên nền tảng Neon Tech.
- **AI:** Python 3.9, thư viện Ultralytics YOLOv8.

4.2. Giao diện chương trình

Hình 1: Giao diện Đăng nhập hệ thống (*Dán ảnh màn hình Login*) Mô tả: Cho phép người dùng đăng nhập bằng tài khoản được cấp. Hỗ trợ ghi nhớ phiên đăng nhập.

Hình 2: Giao diện Dashboard (Tổng quan) (*Dán ảnh màn hình Dashboard*) Mô tả: Hiển thị tổng số tài sản, số lượt quét trong tháng, biểu đồ biến động.

Hình 3: Giao diện Quản lý Tài sản (*Dán ảnh màn hình danh sách Assets*) Mô tả: Danh sách tài sản dạng bảng, hỗ trợ tìm kiếm theo tên và lọc theo Giảng đường (G2, GD2...).

Hình 4: Giao diện Thêm mới Tài sản (*Dán ảnh Popup thêm mới*) Mô tả: Form nhập liệu chi tiết, tách biệt trường Giảng đường và Phòng để chuẩn hóa dữ liệu.

Hình 5: Giao diện Upload Video Quét AI (*Dán ảnh màn hình Upload*) Mô tả: Cho phép kéo thả file video quay tại phòng học để hệ thống xử lý.

Hình 6: Giao diện Chi tiết Kết quả Quét (*Dán ảnh màn hình Scan Detail*) Mô tả: Hiển thị video gốc và danh sách các vật thể (bàn, ghế, máy chiếu) mà AI đã nhận diện được kèm độ tin cậy.

Hình 7: Giao diện Báo cáo (*Dán ảnh màn hình Reports*) Mô tả: Các biểu đồ phân tích sâu về tình trạng cơ sở vật chất của nhà trường.

KẾT LUẬN

Hệ thống **AI Inventory System** đã được phân tích, thiết kế và cài đặt bám sát theo quy trình phát triển phần mềm hướng đối tượng. Với việc áp dụng kiến trúc

Microservices và công nghệ AI tiên tiến, hệ thống giải quyết triệt để bài toán quản lý tài sản thủ công tại Đại học Công nghệ.

Tuy nhiên, hệ thống vẫn còn một số hạn chế về tốc độ xử lý video độ phân giải cao và cần thêm dữ liệu để huấn luyện AI chính xác hơn với các thiết bị đặc thù. Hướng phát triển tiếp theo sẽ là tối ưu hóa thuật toán và phát triển ứng dụng di động (Mobile App).