

Software Project Technical Document

I. Giới Thiệu

1. Bối cảnh
2. Mục tiêu dự án
3. Thông tin dự án
4. Phạm vi
5. Đối tượng sử dụng
6. Giới hạn dự án

II. Kế hoạch quản lý dự án

1. Cơ cấu và trách nhiệm
2. Work Breakdown Structure & Ước lượng (Scope & Estimation)
3. Milestone & KPI
4. Quản lý tiến độ, báo cáo
5. Quản lý rủi ro
6. Quản lý chất lượng
7. Giao tiếp trong Project
8. Quản lý cấu hình dự án phần mềm

III. Đặc tả yêu cầu phần mềm

1. Tổng quan về dự án HomeTrack
2. Yêu cầu người dùng
 - 2.1. Tác nhân
 - 2.2. Use case
 - 2.2.1. Quản lý tài khoản
 - 2.2.2. Quản lý đồ vật
 - 2.2.3. Tổ chức vị trí lưu trữ
 - 2.2.4. Tìm kiếm & Lọc
 - 2.2.5. Thống kê & Insights
 - 2.2.6. Quản lý gói dịch vụ
 - 2.2.7. Kiểm duyệt nội dung
 - 2.2.8. Giám sát & Báo cáo
 - 2.2.9. Dashboard/ Home Screen
 3. Yêu cầu chức năng
 - 3.1. Quản lý tài khoản (UC01, UC02)
 - 3.2. Quản lý đồ vật (UC03, UC04)
 - 3.3. Tổ chức lưu trữ vị trí (UC05, UC06)
 - 3.4. Tìm kiếm & Lọc (UC07, UC08)
 - 3.5. Thống kê & Insight (UC9)
 - 3.6. Quản lý gói dịch vụ (UC10, UC11, UC12)
 - 3.7. Kiểm duyệt nội dung (FR7.1, FR7.2)
 - 3.8. Giám sát & Báo cáo (UC14)
 - 3.9. Dashboard/Home Screen (UC19)
 4. Yêu cầu phi chức năng (Non-Functional Requirements)
 5. Quy tắc nghiệp vụ (Business Rules)
 6. Phân loại mức độ ưu tiên

IV. Bản mô tả thiết kế phần mềm

1. Kiến trúc phần mềm
 - 1.1 Presentation Layer
 - 1.2 Application & Business Layer
 - 1.3 Data Layer
2. Thiết kế CSDL
 - 2.1. Bảng Users
 - 2.2. Bảng Items
 - 2.3. Bảng Locations
 - 2.4. Bảng Tags
 - 2.5. Bảng Items-Tags
 - 2.6. Bảng Subscriptions
 - 2.7. Bảng Search-log

2.8. Bảng Stats-reports
2.9. Bảng Admin-logs
2.10. Bảng moderation_queue
3. Thiết kế giao diện người dùng
3.1 Screens Flow
3.2 Screen Descriptions
3.3 Screen Authorization
4. Mô tả các API có trong hệ thống
V. Tài liệu kiểm thử phần mềm
1. Phạm vi kiểm thử
2. Chiến lược kiểm thử
3. Các trường hợp kiểm thử
4. Báo cáo kết quả kiểm thử
VI. Tài liệu hướng dẫn triển khai phần mềm
1. Yêu cầu hệ thống
2. Cách cài đặt hệ thống
3. Cấu hình phụ trợ
4. Khởi động và truy cập

I. Giới Thiệu

1. Bối cảnh

Trong thời đại công nghệ số phát triển mạnh, nhu cầu quản lý tài sản cá nhân một cách hiệu quả trở nên ngày càng quan trọng. Nhiều người dùng gặp khó khăn ghi nhớ vị trí lưu trữ các vật dụng trong không gian sống và làm việc, dẫn đến việc thất lạc, tìm kiếm mất thời gian và thiếu tổ chức. Bên cạnh đó, sự gia tăng của xu hướng "smart living" đặt ra yêu cầu cho các hệ thống hỗ trợ người dùng sắp xếp và tra cứu thông tin đồ vật nhanh chóng, thuận tiện và thông minh hơn.

2. Mục tiêu dự án

Dự án *HomeTrack* hướng đến việc xây dựng một hệ thống phần mềm hỗ trợ người dùng quản lý và theo dõi vị trí của các đồ vật cá nhân trong không gian sống hoặc làm việc. Hệ thống cho phép người dùng lưu trữ thông tin đồ vật, tổ chức không gian chứa (phòng, kệ, ngăn...), tìm kiếm thông minh và truy xuất nhanh chóng thông tin khi cần thiết. Ngoài ra, hệ thống còn cung cấp các tính năng nâng cao như để xuất tag bằng AI, thống kê sử dụng và phân quyền theo gói dịch vụ.

3. Thông tin dự án

- Tên dự án: HomeTrack – Hệ thống quản lý và định vị đồ dùng cá nhân.
- Loại dự án phần mềm: Web application, Mobile application.

4. Phạm vi

Hệ thống *HomeTrack* bao gồm các thành phần sau:

- Ứng dụng di động cho người dùng cuối.
- Giao diện web dành cho quản trị viên hệ thống, cũng như người dùng.
- API backend phục vụ các thao tác lưu trữ, xử lý và truy xuất dữ liệu.
- Tích hợp các dịch vụ phụ trợ như object storage, AI tagging, xác thực người dùng bằng JWT.

Các tính năng chính được cung cấp bao gồm:

- Quản lý tài khoản: Đăng ký, đăng nhập/dăng xuất, quản lý profile người dùng
- Quản lý đồ vật: Thêm mới đồ vật với các thuộc tính (tên, ảnh, mô tả, tag), chỉnh sửa/xóa vị trí đồ vật. Hệ thống hỗ trợ lưu ảnh vào object storage và có AI gợi ý tag.
- Tổ chức đơn vị lưu trữ: Cho phép tạo/sửa/xóa phòng và container (kệ, ngăn kéo) trong mỗi phòng. Hệ thống hỗ trợ gắn đồ vật vào vị trí cụ thể.
- Tìm kiếm và lọc: Tìm kiếm cơ bản theo tên hoặc tag, và tìm kiếm nâng cao cho Premium User (lọc theo màu, tag).
- Thống kê và Insights: Cung cấp API thống kê số lượt thêm/sửa, tìm kiếm đồ vật, tính toán top đồ vật được tìm kiếm nhiều nhất.
- Quản lý gói dịch vụ: Giới hạn số lượng đồ vật cho tài khoản Basic, quy trình yêu cầu nâng cấp gói. Admin có quyền quản lý trạng thái gói dịch vụ của người dùng.
- Công cụ quản trị dành cho admin: kiểm duyệt nội dung, theo dõi logs và phân tích hệ thống.

5. Đối tượng sử dụng

Hệ thống *HomeTrack* phục vụ ba nhóm đối tượng chính:

- Người dùng cá nhân (Basic/Premium): Những người muốn quản lý đồ vật cá nhân trong không gian sống một cách khoa học và tiện lợi.

- Người dùng nâng cao (Premium):** Người có nhu cầu sử dụng các tính năng mở rộng như thống kê sử dụng, tìm kiếm nâng cao, và số lượng đồ vật lưu trữ lớn hơn.
- Quản trị viên hệ thống (Admin):** Nhân sự vận hành hệ thống có trách nhiệm kiểm duyệt nội dung, quản lý người dùng và theo dõi hoạt động của toàn hệ thống.

6. Giới hạn dự án

LI-1: Không hỗ trợ phần cứng IoT hoặc Bluetooth tracker - Hiện tại HomeTrack chỉ quản lý vị trí dựa trên dữ liệu do người dùng nhập tay (phòng, kệ, ngăn kéo), hệ thống không kết nối trực tiếp với các thiết bị phần cứng.

LI-2: Chỉ lưu trữ metadata ảnh, không xử lý ảnh độ phân giải cao - Hệ thống giới hạn dung lượng ảnh tối đa 5 MB và chỉ lưu trữ đường dẫn URL. Không có cơ chế xử lý ảnh gốc với độ phân giải vượt quá giới hạn, do đó chất lượng khi phóng to có thể giảm.

LI-3: AI gợi ý tag ở mức cơ bản - Tính năng gợi ý thẻ phân loại (tag) sử dụng mô hình nhận diện ảnh bên thứ ba ở mức free tier, chỉ đủ để gợi ý một số danh mục chung. Độ chính xác và phạm vi thẻ gợi ý có thể không đầy đủ hoặc sai lệch với nhu cầu chuyên biệt.

LI-4: Không hỗ trợ hoạt động offline toàn phần - Ứng dụng di động yêu cầu kết nối Internet để thêm/sửa/xóa đồ vật và đồng bộ dữ liệu. Không có cơ chế lưu tạm (cache) đầy đủ cho các thao tác ngoại tuyến; người dùng không thể thực hiện thao tác khi mất mạng.

LI-5: Không chia sẻ vị trí giữa nhiều tài khoản - Mỗi vật dụng và không gian lưu trữ chỉ thuộc quyền sở hữu và quản lý của một user duy nhất. Hệ thống chưa hỗ trợ cấp quyền truy cập hoặc chia sẻ kho đồ giữa nhiều user hoặc nhóm (family sharing).

LI-6: Không tích hợp băn đồ địa lý (GPS) - HomeTrack chỉ mô hình hóa không gian theo sơ đồ phòng/kệ/ngăn kéo. Không có chức năng định vị theo GPS, do đó không thể dùng cho việc định vị đồ vật di động hay ngoài không gian nội bộ.

LI-7: Giới hạn báo cáo phân tích cơ bản - Các báo cáo dành cho user Premium và Admin chỉ bao gồm thống kê đơn giản (top items, lượt tìm kiếm, số đồ tạo mới theo tháng). Không có phân tích nâng cao như forecasting, clustering hành vi, hoặc dashboard BI phức tạp.

LI-8: Không hỗ trợ đa ngôn ngữ (ngoài Tiếng Việt) - Ứng dụng hiện chỉ có giao diện và tài liệu hướng dẫn bằng Tiếng Việt. Không có sẵn bản dịch hoặc hỗ trợ đa ngôn ngữ cho người dùng quốc tế.

LI-9: Không bao gồm module thanh toán thực tế - Giao diện nâng cấp gói Premium mô phỏng luồng "Mua gói", nhưng không tích hợp trực tiếp cổng thanh toán (VNPay, Stripe, PayPal).

II. Kế hoạch quản lý dự án

1. Cơ cấu và trách nhiệm

Vai trò	Tên thành viên	Trách nhiệm chính
Project Sponsor	All	Phê duyệt phạm vi; chốt deliverables
Project Manager	Nguyễn Tiến Duy	Lập kế hoạch, theo dõi tiến độ, điều phối nhóm
Business Analyst	All	Thu thập & xác nhận yêu cầu
Technical Lead	Nguyễn Tiến Duy	Thiết kế kiến trúc; giám sát code quality
Designer	Trương Hoàng Thiện	Thiết kế figma giao diện web, mobile app.
Frontend Developer	Trương Hoàng Thiện, Đoàn Trọng Đại, Lê Thanh Đạt	Xây dựng giao diện web, mobile app.
Backend Developer	Nguyễn Tiến Duy, Phan Thiên Lộc, Võ Trần Khiêm	Xây dựng backend
QA Engineer	Võ Trần Khiêm	Thiết kế & thực thi test cases
DevOps Engineer	Nguyễn Tiến Duy	Thiết lập CI/CD, giám sát, deploy.

2. Work Breakdown Structure & Ước lượng (Scope & Estimation)

WBS	Task	Owner	Effort Est. (man-days)	Actual
1	Initiation		7	6

1.1	Xác định phạm vi dự án	PM	2	1
1.2	Thu thập yêu cầu sơ bộ	BA	3	4
1.3	Lập Project Charter	PM	2	1
2	Planning		14	
2.1	Lập kế hoạch quản lý dự án (WBS, timeline)	PM	5	
2.2	Đánh giá rủi ro & biện pháp ứng phó	PM/BA	4	
2.3	Chuẩn bị môi trường DevOps (CI/CD, repo)	DevOps	5	
3	Analysis & Design		28	
3.1	Mô hình Use Case & SRS	BA	6	
3.2	Thiết kế ERD & Database schema	TL/BE	4	
3.3	Thiết kế kiến trúc backend (API contract)	TL / BE	4	
3.4	Mockup giao diện mobile	UI/UX Designer	4	
3.5	Mockup giao diện mobile	UI/UX Designer	4	
3.6	Định nghĩa các test case sơ bộ	QA	6	
4	Implementation		40	
4.1	Phát triển module Auth & User Management (backend)	BE	6	
4.2	Phát triển API Items & Locations	BE	6	
4.3	Phát triển Frontend Web (React)	FE	8	
4.4	Phát triển Mobile App (React Native)	FE	8	
4.5	Tích hợp AI Tagging	TL/BE	6	
4.6	Cấu hình Object Storage	DevOps	6	
5	Testing & QA		16	
5.1	Integration test (backend)	QA / BE Dev	6	

5.2	UI & Usability test (web + mobile)	QA / FE & Mobile	6	
5.3	Performance & security test	QA / DevOps	4	
6	Deployment & Closure		10	
6.1	Triển khai staging & demo cho thầy	DevOps / PM	4	
6.2	Go-Live production	DevOps	3	
6.3	Hoàn thiện tài liệu hướng dẫn và nộp bài	PM / BA	3	
Tổng			120	

- **PM:** Project Manager
- **BA:** Business Analyst
- **TL:** Technical Lead
- **BE Dev:** Backend Developer
- **FE Dev:** Frontend Developer
- **Mobile Dev:** Mobile Developer
- **QA:** Quality Assurance Engineer
- **DevOps:** DevOps Engineer

Lưu ý: cột **Actual** để trống, cập nhật sau khi hoàn thành task để theo dõi chênh lệch so với ước lượng ban đầu.

3. Milestone & KPI

Milestone	Ngày dự kiến	Chỉ số đo lường (KPI)
M1: Hoàn thành Draft SRS & Architecture	26/05/2025	100% Use Case & FR được review và ký duyệt 0 ý kiến mở (open issues)
M2: Hoàn thiện Thiết kế Dữ liệu (ERD)	30/05/2025	ERD & bảng schema đã duyệt 100% migration scripts chạy thành công
M3: Hoàn thành API Contract	05/06/2025	100% endpoint được định nghĩa API spec đạt độ phủ test ≥ 90%
M4: Hoàn thiện UI Mockups Web & Mobile	09/06/2025	5 màn hình chính có mockup UX review score ≥ 4/5
M5: Triển khai Module Auth & User	13/06/2025	100% story points cho UC01–UC02 hoàn thành Unit test cover ≥ 80%
M6: Triển khai Module Items & Locations	19/06/2025	100% story points cho UC03–UC06 hoàn thành API latency < 500 ms
M7: Tích hợp Search & Dashboard Stats	23/06/2025	Tìm kiếm cơ bản & nâng cao pass smoke test Dashboard load < 500 ms
M8: Triển khai Admin & Moderation	27/06/2025	100% story points cho UC11–UC13 hoàn thành Defect escape rate ≤ 2%

M9: Hoàn thành Integration Test & Bugfix	28/06/2025	0 critical bug còn tồn đọng Test coverage toàn hệ thống ≥ 85%
M10: Release MVP & Go-Live	29/06/2025	Deploy thành công (0 rollback)

4. Quản lý tiến độ, báo cáo

- Sprint Review (mỗi nửa tuần): báo cáo tiến độ task, nêu các vấn đề gặp phải/tồn đọng, thu feedback. Tạo ra sprint report mỗi sprint theo form meeting note.
 - So sánh “Planned vs. Done” (story points, task list)
 - Đánh dấu task hoàn thành, đang làm dở, hoặc chưa được bắt đầu.
 - Liệt kê blocker, kỹ thuật, tài nguyên hoặc thông tin thiếu thốn.
 - Đề xuất phương án giải quyết hoặc hỗ trợ cần thiết.
- Change Control: Mọi thay đổi về yêu cầu nghiệp vụ, thiết kế hệ thống đều phải qua form Change Request, đánh giá ảnh hưởng về scope/time.

5. Quản lý rủi ro

ID	Rủi ro	Xác suất	Ảnh hưởng
R1	Phân công công việc không đồng đều, một số thành viên thiếu trách nhiệm	Medium	High
R2	Thiếu kinh nghiệm hoặc kỹ năng về công nghệ chính (React, React Native, .NET)	High	Medium
R3	Thời gian thực hiện gấp gáp, trùng lịch học hoặc thi cử	High	High
R4	Yêu cầu giáo viên hướng dẫn thay đổi giữa chứng	Medium	Medium
R5	Môi trường phát triển (máy, phần mềm) không ổn định.	Medium	Medium
R6	Không đủ thời gian để kiểm thử toàn diện (unit, integration)	High	Medium
R7	Xung đột merge code, mất dữ liệu do quản lý mã nguồn kém	Medium	Medium
R8	Khó khăn trong triển khai demo (hosting, domain, SSL)	Medium	Low
R9	Tài liệu (SRS, design) không hoàn thiện, gây hiểu nhầm khi làm việc nhóm	Medium	Medium
R10	Thiếu phản hồi hoặc hướng dẫn kịp thời từ giảng viên	Low	Medium

- Xác suất:** Low (thấp), Medium (trung bình), High (cao)
- Ảnh hưởng:** Low (tác động nhẹ), Medium (tác động vừa phải), High (tác động lớn)

6. Quản lý chất lượng

- Definition of Done (DoD) cho từng user story: Code reviewed ≥ 2 devs, integration test, UAT passed.
- Automation CI/CD pipeline: Build → SonarQube → OWASP ZAP → Deploy.
- Code Quality Gates: SonarQube để giáp sát code smells, security checks.

7. Giao tiếp trong Project

Hình thức giao tiếp	Mục đích	Người tham gia	Tần suất/thời gian	Công cụ sử dụng	Người chịu trách nhiệm
Hợp Sprint Planning	Lên kế hoạch cho Sprint mới	Tất cả thành viên nhóm	Trước mỗi Sprint (2 ngày/lần)	Google Meet, Confluence	Project Manager

Sprint Review	Trình bày sản phẩm Sprint	Tất cả thành viên nhóm	Cuối mỗi Sprint	Google Meet, Github, Confluence	Team Leader
Sprint Retrospective	Đánh giá lại cách làm việc	Tất cả thành viên nhóm	Sau Sprint Review	Google Meet, Confluence	Project Manager
Trao đổi kỹ thuật nhanh	Thảo luận issue nhỏ, fix bug	Dev liên quan	Bất kỳ khi nào	Zalo, GitHub issue	Người phát hiện
Code Review	Kiểm tra chất lượng code	≥ 2 dev khác	Khi có pull request	GitHub Pull Request	Reviewer
Gửi báo cáo tiến độ	Tổng hợp kết quả, vướng mắc	Các thành viên gửi cho Teach Lead	Mỗi Sprint	Google Docs	Team Leader
Cập nhật tài liệu	Đồng bộ tài liệu thiết kế/code	Mọi thành viên	Sau mỗi Sprint	Google Docs, Confluence	QA

8. Quản lý cấu hình dự án phần mềm

- Document Management: Google Docs, Confluence.
- Source Code Management: được quản lý trên Github.
- Công cụ và hạ tầng sử dụng:

Category	Tools/Infrastructure	Explain
Technology	React, React Native (FE), C#/.NET (BE)	Công nghệ chính cho frontend/backend
Database	PostgreSQL	Lưu dữ liệu hệ thống
IDEs/Editors	VS Code, Visual Studio	Công cụ viết code
Diagramming	DrawIO, PlantUML	Dùng vẽ biểu đồ (UML, ERD...)
Design UI	Figma	Dùng để thiết kế Frontend
Documentation	MS Office online, Google Docs, Confluence	Viết và lưu tài liệu
Version Control	GitHub	Lưu code, theo dõi version
Deployment Tool	Docker	Triển khai hệ thống
Project Management	Confluence	Quản lý công việc, phân chia task

III. Đặc tả yêu cầu phần mềm

1. Tổng quan về dự án HomeTrack

HomeTrack là một hệ thống quản lý và hỗ trợ người dùng định vị, theo dõi và tổ chức các vật dụng cá nhân trong không gian sống, đặc biệt hữu ích trong bối cảnh nhà ở sinh viên, ký túc xá, hoặc gia đình. Mục tiêu của hệ thống là giải quyết các vấn đề thường gặp như mất đồ, thất lạc vật dụng, hoặc khó khăn trong việc quản lý tài sản cá nhân.

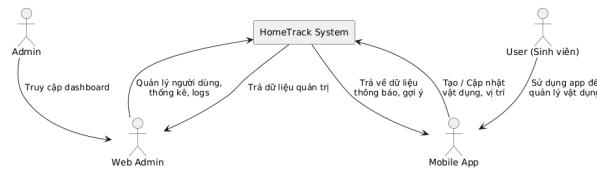
Hệ thống bao gồm hai thành phần chính: **ứng dụng di động cho người dùng cuối (mobile app)** và **bảng điều khiển quản trị viên (web admin dashboard)**. Người dùng có thể chụp ảnh, phân loại và ghi chú vị trí của các vật dụng trong nhà. Ngoài ra, hệ thống còn hỗ trợ chức năng tìm kiếm nhanh, cảnh báo khi vật dụng không ở đúng vị trí và phân tích thói quen sử dụng đồ đạc.

Trong giai đoạn MVP (Minimum Viable Product), HomeTrack tập trung vào các tính năng cốt lõi:

- Ghi nhận và theo dõi vị trí vật dụng.
- Hỗ trợ tìm kiếm bằng hình ảnh hoặc từ khóa
- Thông báo khi có sự thay đổi bất thường.

Hệ thống được thiết kế linh hoạt để hỗ trợ khả năng mở rộng trong các giai đoạn phát triển sau, không bao gồm tích hợp với thiết bị IoT, không hỗ trợ nhiều người dùng trong một không gian, và để xuất vị trí tối ưu cho các vật dụng dựa trên hình ảnh.

HomeTrack không chỉ giúp sinh viên tiết kiệm thời gian tìm kiếm vật dụng mà còn xây dựng thói quen sống ngắn nắp, chủ động hơn trong việc quản lý không gian cá nhân.



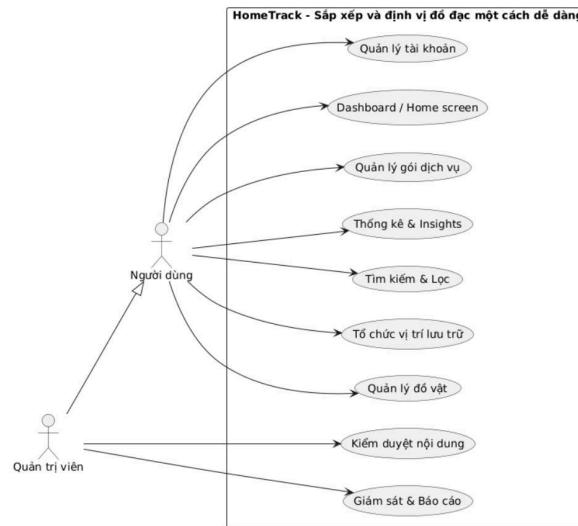
2. Yêu cầu người dùng

2.1. Tác nhân

ID	Actor	Description
1	Admin	Quản trị hệ thống, có quyền quản lý người dùng, theo dõi nhật ký và thống kê
2	Premium user	Người dùng trả phí, có quyền truy cập tính năng nâng cao (tìm kiếm nâng cao, theo dõi vị trí nâng cao)
3	Basic user	Người dùng miễn phí, có thể quản lý danh sách vật dụng cơ bản và xem lại lịch sử thao tác

2.2. Use case

Sơ đồ Use case tổng quát:



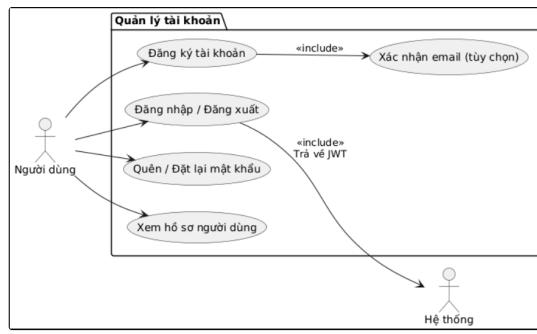
Hình 2.2.a. Use case HomeTrack – Hệ thống quản lý và định vị đồ đạc một cách dễ dàng

Mô tả Use case

Sơ đồ Use case tổng quát:

- Người dùng có thể thực hiện các chức năng: Quản lý tài khoản, xem dashboard, quản lý gói dịch vụ, thống kê, tìm kiếm, tổ chức vị trí lưu trữ, và quản lý đồ vật.
- Quản trị viên kế thừa quyền của người dùng, đồng thời có thêm chức năng: Kiểm duyệt nội dung và Giám sát & Báo cáo.

2.2.1. Quản lý tài khoản



Hình 2.2.b. Use case Quản lý tài khoản

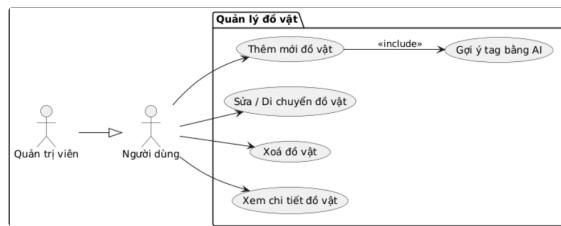
Mô tả Use case

Các thành phần chính:

- Tác nhân: Người dùng và Hệ thống.
- Các chức năng chính:
 - Đăng ký tài khoản (include) → Xác nhận email (tùy chọn).
 - Đăng nhập / Đăng xuất (include) → Trả về JWT (token xác thực).
 - Quên / Đặt lại mật khẩu.
 - Xem hồ sơ người dùng.

Sơ đồ cho thấy Hệ thống là bên xử lý logic như trả JWT (JSON Web Token), còn người dùng tương tác với các chức năng thông qua giao diện.

2.2.2. Quản lý đồ vật



Hình 2.2.c. Use case Quản lý đồ vật

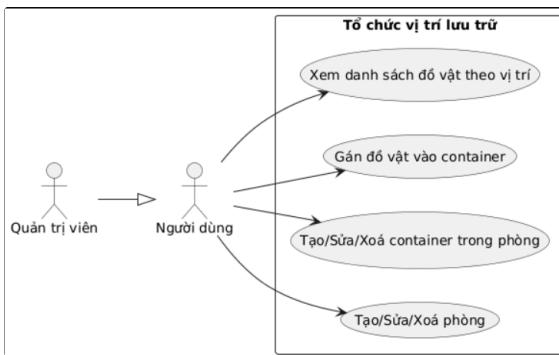
Mô tả Use Case

Thành phần chính:

- Tác nhân: Người dùng (được kế thừa từ Quản trị viên).
- Chức năng:
 - Thêm mới đồ vật (include) → Gợi ý tag bằng AI.
 - Sửa / Di chuyển đồ vật.
 - Xoá đồ vật.
 - Xem chi tiết đồ vật.

Sơ đồ cho thấy người dùng có thể thao tác đầy đủ với đồ vật trong hệ thống, và có thể nhận hỗ trợ thông minh từ AI khi thêm mới.

2.2.3. Tổ chức vị trí lưu trữ



Hình 2.2.d. Use case Tổ chức vị trí lưu trữ

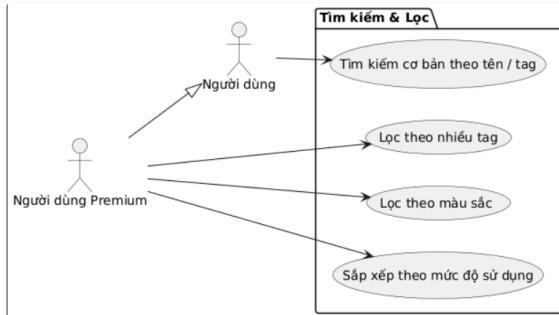
Mô tả Use Case

Thành phần chính:

- Tác nhân: Người dùng (được kế thừa từ Quản trị viên).
- Chức năng:
 - Xem danh sách đồ vật theo vị trí.
 - Gán đồ vật vào container.
 - Tạo/Sửa/Xoá container trong phòng.
 - Tạo/Sửa/Xoá phòng.

Người dùng có toàn quyền tổ chức hệ thống lưu trữ theo phòng và container, giúp quản lý và định vị đồ vật một cách hiệu quả.

2.2.4. Tìm kiếm & Lọc



Hình 2.2.e. Use case Tim kiem & Loc

Mô tả Use Case

Thành phần chính:

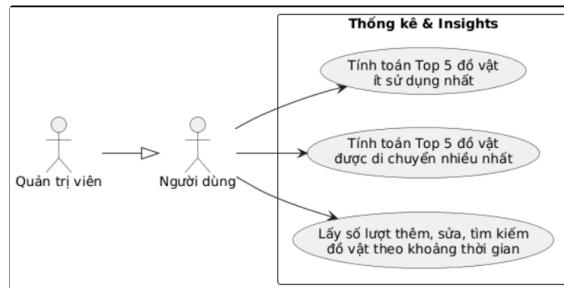
- Tác nhân:
 - Người dùng: Có quyền Tìm kiếm cơ bản theo tên / tag.
 - Người dùng Premium (kế thừa từ Người dùng): Có thêm quyền nâng cao.

Các chức năng:

- Tìm kiếm cơ bản theo tên / tag (cho mọi người dùng).
- Lọc theo nhiều tag, lọc theo màu sắc, sắp xếp theo mức độ sử dụng (chỉ dành cho người dùng Premium).

Sơ đồ cho thấy tính năng nâng cao được phân quyền theo loại tài khoản.

2.2.5. Thống kê & Insights



Hình 2.2.f. Thống kê & Insights

Mô tả Use Case

Thành phần chính:

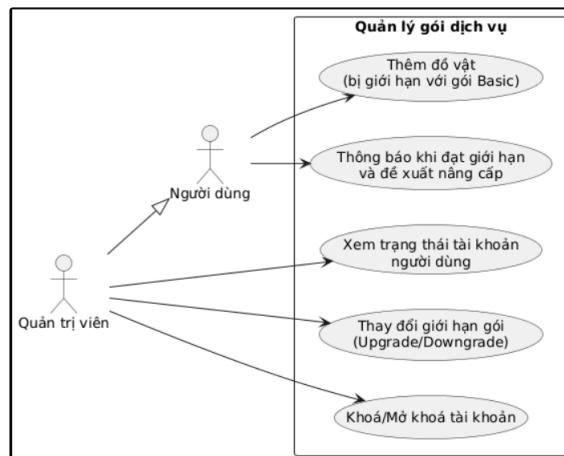
- Tác nhân: Người dùng (bao gồm cả Quản trị viên).

Các chức năng thống kê:

- Tính toán Top 5 đồ vật ít sử dụng nhất.
- Tính toán Top 5 đồ vật được di chuyển nhiều nhất.
- Lấy số lượt thêm, sửa, tìm kiếm đồ vật theo khoảng thời gian.

Sơ đồ thể hiện các tính năng phân tích hành vi và hoạt động của người dùng nhằm cung cấp thông tin hữu ích để tối ưu quản lý đồ vật.

2.2.6. Quản lý gói dịch vụ



Hình 2.2.g. Use case Quản lý gói dịch vụ

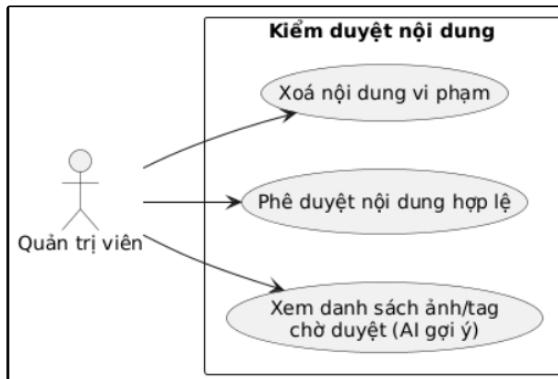
Mô tả Use Case

Thành phần chính:

- Người dùng:
 - Có thể thêm đồ vật (có giới hạn với gói Basic).
 - Nhận thông báo khi đạt giới hạn và đề xuất nâng cấp.
- Quản trị viên:
 - Xem trạng thái tài khoản người dùng.
 - Thay đổi giới hạn gói dịch vụ (nâng cấp/hạ cấp).
 - Khóa/Mở khóa tài khoản người dùng.

Sơ đồ thể hiện rõ vai trò của quản trị viên trong việc quản lý tài khoản và gói dịch vụ, cũng như các giới hạn áp dụng cho người dùng.

2.2.7. Kiểm duyệt nội dung



Hình 2.2.h. Use case Kiểm duyệt nội dung

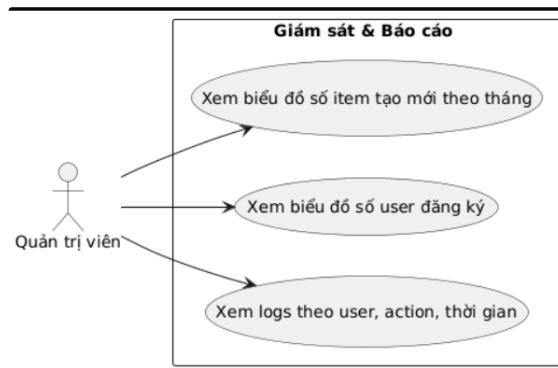
Mô tả Use Case

Thành phần chính:

- Tác nhân: Quản trị viên.
- Các chức năng:
 - Xóa nội dung vi phạm theo chính sách.
 - Phê duyệt nội dung hợp lệ để hiển thị trên hệ thống.
 - Xem danh sách ảnh/tag chờ duyệt (danh sách do AI đề xuất).

Sơ đồ thể hiện rõ vai trò của quản trị viên trong việc đảm bảo chất lượng nội dung qua các thao tác duyệt hoặc xóa dựa trên đánh giá nội dung.

2.2.8. Giám sát & Báo cáo



Hình 2.2.k. Use case Giám sát & Báo cáo

Mô tả Use Case

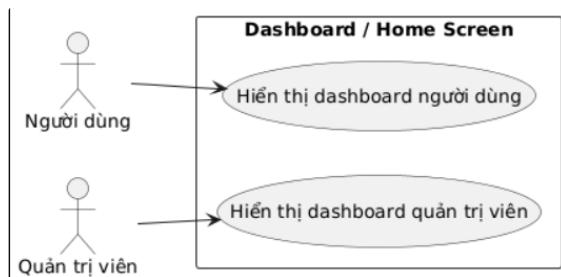
Thành phần chính:

- Tác nhân: Quản trị viên.
- Các chức năng:
 - Xem biểu đồ số item tạo mới theo tháng để theo dõi mức độ hoạt động.
 - Xem biểu đồ số user đăng ký để đánh giá mức độ thu hút người dùng.
 - Xem logs theo user, action, thời gian nhằm phục vụ công tác kiểm tra và truy vết.

Sơ đồ thể hiện vai trò của quản trị viên trong việc theo dõi hoạt động hệ thống và đưa ra quyết định dựa trên dữ liệu.

2.2.9. Dashboard/ Home Screen

∅



Hình 2.2.1. Use case Dashboard/ Home Screen

Mô tả Use Case

Biểu đồ này chỉ ra hai tác nhân (actor) chính:

- Người dùng (Người dùng): Tương tác với hệ thống để "Hiển thị dashboard người dùng".
- Quản trị viên (Quản trị viên): Tương tác với hệ thống để "Hiển thị dashboard quản trị viên".

Tóm lại, biểu đồ này minh họa cách mà người dùng và quản trị viên tương tác với hệ thống để truy cập vào các giao diện dashboard tương ứng của họ.

3. Yêu cầu chức năng ∅

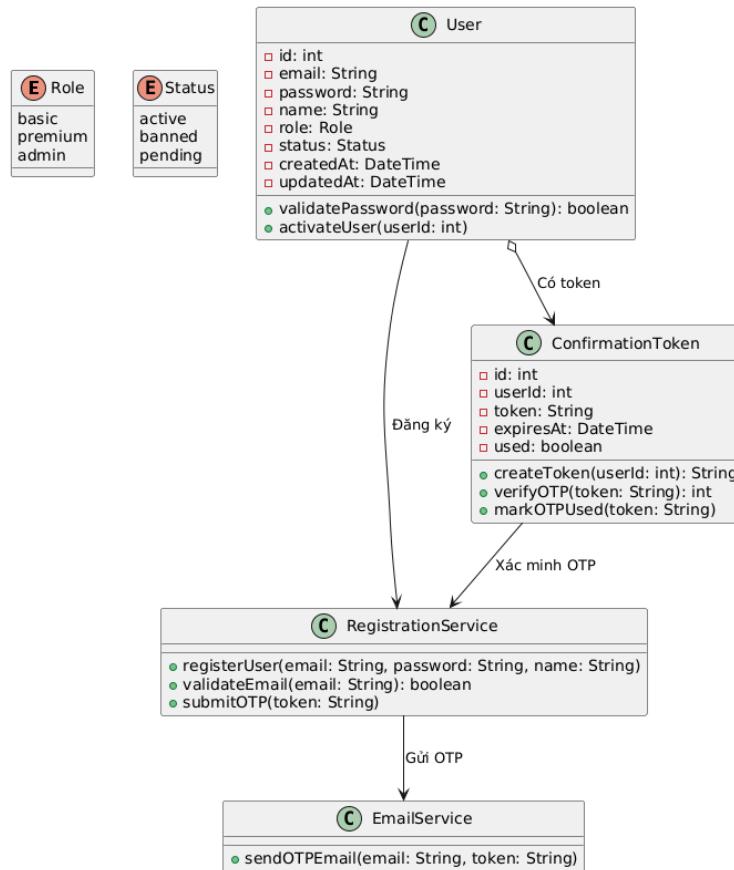
3.1. Quản lý tài khoản (UC01, UC02) ∅

UC01: Đăng ký tài khoản

ID	Mô tả yêu cầu chức năng
FR1.1	Người dùng có thể đăng ký bằng email và mật khẩu; email phải hợp lệ, mật khẩu ≥ 8 ký tự.
FR1.2	Hệ thống gửi email xác nhận (tùy chọn).

Class diagram:

Bảng vẽ



Phân tích

- Các thành phần và thuộc tính
 - enum Role { basic, premium, admin }:**
 - Định nghĩa vai trò người dùng (role), mặc định basic cho tài khoản mới.
 - enum Status { active, banned, pending }:**
 - Định nghĩa trạng thái tài khoản (status).
 - pending cho tài khoản mới, active sau khi xác nhận OTP (FR1.2).
 - class User:**
 - Đại diện:** Bảng users trong database.
 - Thuộc tính:**
 - id: int: Khóa chính.
 - email: String, password: String, name: String: Thông tin đăng ký (FR1.1).
 - role: Role, status: Status: Vai trò và trạng thái.
 - createdAt: DateTime, updatedAt: DateTime: Thời gian tạo và cập nhật.
 - Phương thức:**
 - +validatePassword(password: String): boolean: Kiểm tra mật khẩu ≥ 8 ký tự (FR1.1).
 - +activateUser(userId: int): Cập nhật status = 'active' sau khi xác nhận OTP (FR1.2).
 - Ánh xạ schema:** Bảng users với các cột id, email, password, name, role, status, createdAt, updatedAt.
 - class RegistrationService:**
 - Đại diện:** Dịch vụ xử lý logic đăng ký và xác nhận OTP.
 - Phương thức:**
 - +registerUser(email: String, password: String, name: String): Đăng ký tài khoản mới (FR1.1).
 - +validateEmail(email: String): boolean: Kiểm tra email hợp lệ (FR1.1).
 - +submitOTP(token: String): Xác minh OTP và kích hoạt tài khoản (FR1.2).
 - Vai trò:** Điều phối quy trình đăng ký, tạo OTP, gửi email, và xác minh.
 - class EmailService:**
 - Đại diện:** Dịch vụ gửi email.

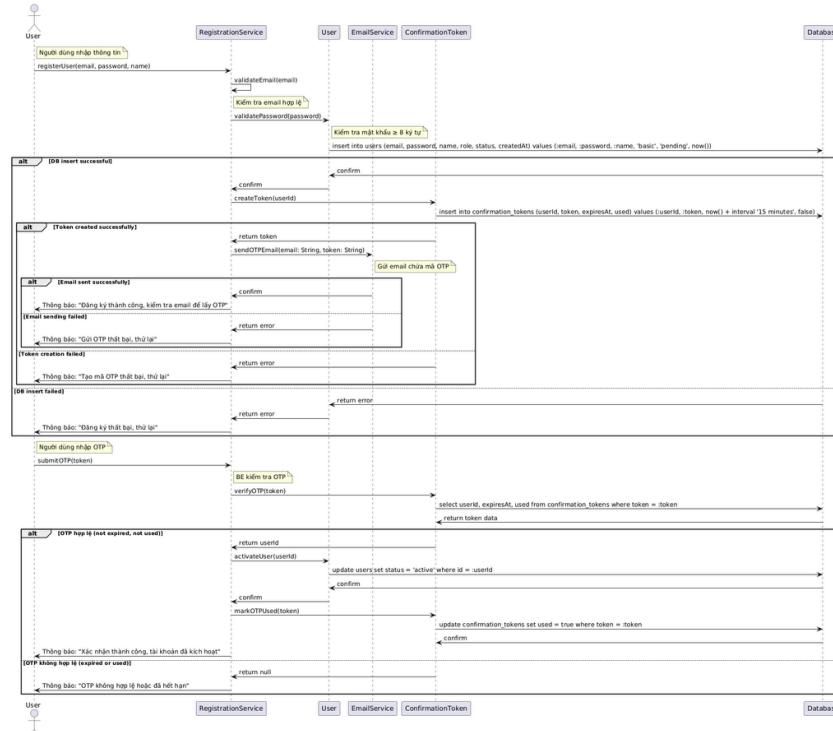
- **Phương thức:**
 - +sendOTPEmail(email: String, token: String): Gửi email chứa mã OTP (FR1.2).
- **Vai trò:** Đảm bảo OTP được gửi đến người dùng với format input rõ ràng (email: String, token: String).
- **class ConfirmationToken:**
 - **Đại diện:** Bảng confirmation_tokens (lưu OTP xác nhận).
 - **Thuộc tính:**
 - id: int: Khóa chính.
 - userId: int: Liên kết với tài khoản người dùng.
 - token: String: Mã OTP.
 - expiresAt: DateTime: Thời gian hết hạn (15 phút).
 - used: boolean: Trạng thái sử dụng (mặc định false, chuyển thành true sau xác minh).
- **Phương thức:**
 - +createToken(userId: int): String: Tạo OTP mới.
 - +verifyOTP(token: String): int: Kiểm tra OTP (hợp lệ, chưa hết hạn, chưa sử dụng), trả userId.
 - +markOTPUsed(token: String): Đánh dấu used = true.
- **Ánh xạ schema:** Bảng confirmation_tokens với các cột id, userId, token, expiresAt, used.

2. Quan hệ giữa các thành phần

- **User → RegistrationService:**
 - Người dùng tương tác với RegistrationService để đăng ký và xác minh OTP.
 - RegistrationService gọi User để kiểm tra mật khẩu và kích hoạt tài khoản.
- **User o→ ConfirmationToken:**
 - Mỗi tài khoản (User) có thể có nhiều ConfirmationToken (OTP).
 - Quan hệ "composition" (o→) vì ConfirmationToken phụ thuộc vào User.
- **RegistrationService → EmailService:**
 - RegistrationService gọi EmailService để gửi email chứa OTP.
- **ConfirmationToken → RegistrationService:**
 - RegistrationService gọi ConfirmationToken để tạo, xác minh, và đánh dấu OTP.
- **Đáp ứng FR:**
 - **FR1.1:** Hỗ trợ đăng ký với kiểm tra email và mật khẩu.
 - **FR1.2:** Hỗ trợ gửi email xác nhận qua EmailService.

Sequence Diagram

Bảng vẽ



Phân tích

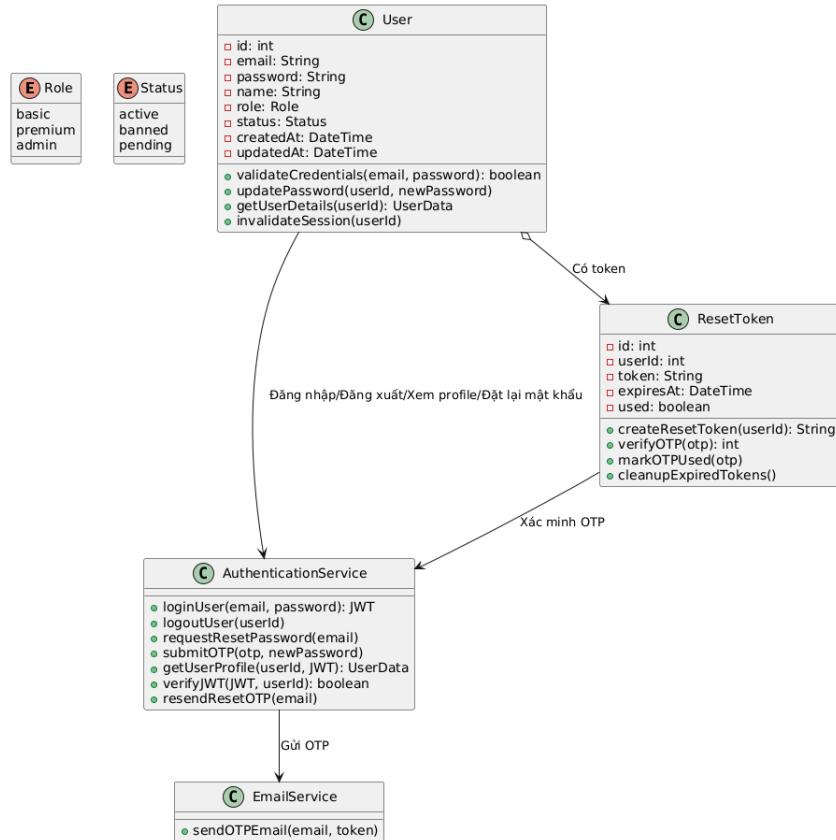
- Các bước chính
 - Đăng ký tài khoản:
 - Người dùng gửi thông tin (email, password, name) qua UserActor đến RegistrationService.
 - RegistrationService.validateEmail(email) kiểm tra email hợp lệ (FR1.1).
 - User.validatePassword(password) kiểm tra mật khẩu ≥ 8 ký tự (FR1.1).
 - Lưu tài khoản vào Database với status = 'pending' (FR1.2).
 - Tạo và gửi OTP:
 - Nếu lưu thành công, RegistrationService gọi ConfirmationToken.createToken(userId) để tạo OTP (hết hạn 15 phút, used = false).
 - Nếu tạo token thành công, RegistrationService gọi EmailService.sendOTPEmail(email: String, token: String) để gửi OTP qua email.
 - Thông báo tùy theo kết quả:
 - Email gửi thành công: "Đăng ký thành công, kiểm tra email để lấy OTP".
 - Email gửi thất bại: "Gửi OTP thất bại, thử lại".
 - Tạo token thất bại: "Tạo mã OTP thất bại, thử lại".
 - Lưu DB thất bại: "Đăng ký thất bại, thử lại".
 - Xác minh OTP:
 - Người dùng nhập OTP qua UserActor, gửi đến RegistrationService.submitOTP(token).
 - RegistrationService gọi ConfirmationToken.verifyOTP(token) để kiểm tra OTP (hợp lệ, chưa hết hạn, chưa sử dụng).
 - Nếu hợp lệ:
 - RegistrationService gọi User.activateUser(userId) để cập nhật status = 'active'.
 - Gọi ConfirmationToken.markOTPUsed(token) để đánh dấu used = true.
 - Thông báo: "Xác nhận thành công, tài khoản đã kích hoạt".
 - Nếu không hợp lệ: Thông báo "OTP không hợp lệ hoặc đã hết hạn".
 - Đáp ứng FR:
 - FR1.1: Quy trình kiểm tra email hợp lệ, mật khẩu ≥ 8 ký tự.
 - FR1.2: Gửi email xác nhận (tùy chọn) với token.

UC02: Đăng nhập/Đăng xuất (FR1.3, FR1.4, FR1.5, FR1.6)

ID	Mô tả yêu cầu chức năng
FR1.3	Hệ thống cho phép đăng nhập/đăng xuất, trả về JWT (JSON Web Token)
FR1.4	Đăng nhập, đăng xuất.
FR1.5	Quên, đặt lại mật khẩu.
FR1.6	Xem chi tiết user profile.

Class diagram

Bảng vẽ



Phân tích

- **enum Role { basic, premium, admin }:**
 - Định nghĩa vai trò người dùng (role), áp dụng cho User.
- **enum Status { active, banned, pending }:**
 - Định nghĩa trạng thái tài khoản (status), bao gồm pending (chờ xác nhận), active, banned.
- **class User:**
 - **Đại diện:** Bảng users trong database.
 - **Thuộc tính:**
 - id: int: Khóa chính.
 - email: String, password: String, name: String: Thông tin người dùng.
 - role: Role, status: Status: Vai trò và trạng thái.
 - createdAt: DateTime, updatedAt: DateTime: Thời gian tạo và cập nhật.
 - **Phương thức:**
 - +validateCredentials(email, password): boolean: Kiểm tra email, mật khẩu, và status = 'active' (FR1.3, FR1.4).
 - +updatePassword(userId, newPassword): Cập nhật mật khẩu (FR1.5).
 - +getUserDetails(userId): UserData: Lấy dữ liệu (email, name, role) (FR1.6).
 - +invalidateSession(userId): Hủy phiên đăng nhập (FR1.3, FR1.4).
 - **Ánh xạ schema:** Bảng users với các cột tương ứng.
- **class AuthenticationService:**
 - **Đại diện:** Dịch vụ xử lý logic đăng nhập, đăng xuất, đặt lại mật khẩu, và xem profile.
 - **Phương thức:**
 - +loginUser(email, password): JWT: Đăng nhập và trả về JWT (FR1.3, FR1.4).
 - +logoutUser(userId): Đăng xuất (FR1.3, FR1.4).
 - +requestResetPassword(email): Yêu cầu đặt lại mật khẩu (FR1.5).
 - +submitOTP(otp, newPassword): Xác minh OTP và cập nhật mật khẩu (FR1.5).
 - +getUserProfile(userId, JWT): UserData: Lấy và trả dữ liệu profile sau khi kiểm tra JWT (FR1.6).
 - +verifyJWT(JWT, userId): boolean: Kiểm tra tính hợp lệ của JWT và khớp userId (FR1.6).
 - +resendResetOTP(email): Gửi lại OTP mới (FR1.5).
 - **Vai trò:** Điều phối toàn bộ quy trình, tích hợp với User và ResetToken.
- **class EmailService:**

- **Đại diện:** Dịch vụ gửi email.
- **Phương thức:**
 - +sendOTPEmail(email, token): Gửi email chứa OTP (FR1.5).
- **Vai trò:** Hỗ trợ gửi OTP cho đặt lại mật khẩu.
- **class ResetToken:**
 - **Đại diện:** Bảng reset_tokens (lưu OTP cho đặt lại mật khẩu).
 - **Thuộc tính:**
 - id: int: Khóa chính.
 - userId: int: Liên kết với tài khoản người dùng.
 - token: String: Mã OTP (6-8 chữ số).
 - expiresAt: DateTime: Thời gian hết hạn (10 phút).
 - used: boolean: Trạng thái sử dụng (mặc định false, chuyển thành true sau khi sử dụng).
 - **Phương thức:**
 - +createResetToken(userId): String: Tạo OTP mới (6-8 chữ số, hết hạn 10 phút).
 - +verifyOTP(otp): int: Kiểm tra OTP (hợp lệ, chưa hết hạn, chưa sử dụng), trả userId.
 - +markOTPUsed(otp): Đánh dấu used = true.
 - +cleanupExpiredTokens(): Xóa OTP hết hạn.
 - **Ánh xạ schema:** Bảng reset_tokens với các cột userId, token, expiresAt, used.

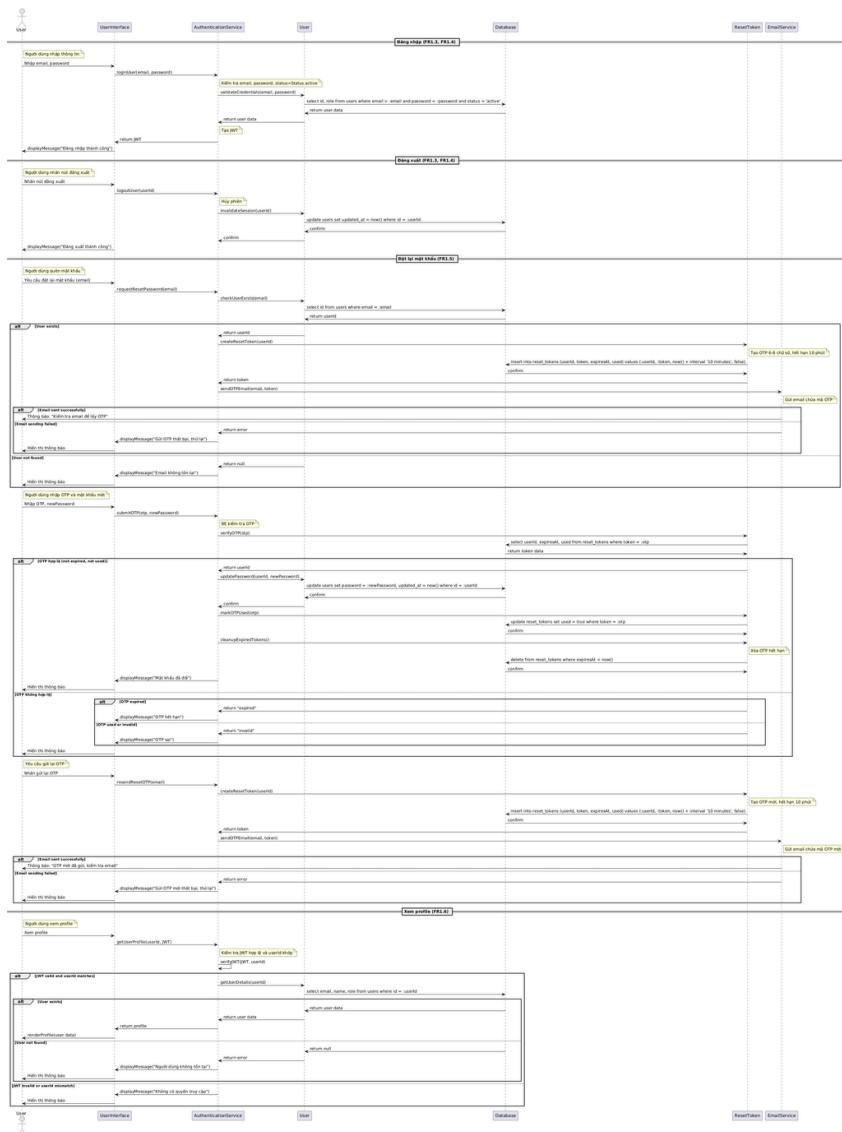
Quan hệ

User → AuthenticationService:

- Người dùng tương tác với AuthenticationService cho đăng nhập, đăng xuất, xem profile, và đặt lại mật khẩu.
 - AuthenticationService gọi User để xác thực, cập nhật mật khẩu, lấy dữ liệu, hoặc hủy phiên.
- **User → ResetToken:**
 - Mỗi tài khoản (User) có thể có nhiều ResetToken (OTP).
 - Quan hệ "composition" (→) vì ResetToken phụ thuộc vào User.
- **AuthenticationService → EmailService:**
 - AuthenticationService gọi EmailService để gửi OTP qua email.
- **ResetToken → AuthenticationService:**
 - AuthenticationService gọi ResetToken để tạo, xác minh, và quản lý OTP.

Sequence Diagram

Bảng vẽ



Phân tích

Đăng nhập (FR1.3, FR1.4)

- Người dùng nhập email, password qua UserInterface.
- AuthenticationService kiểm tra User.validateCredentials (email, password, status = 'active') từ Database.
- Nếu hợp lệ, tạo JWT và trả về UserInterface.
- Thông báo: "Đăng nhập thành công".

Đăng xuất

- Người dùng nhấn nút đăng xuất qua UserInterface.
- AuthenticationService gọi User.invalidateSession và cập nhật updated_at trong Database.
- Thông báo: "Đăng xuất thành công"

Đặt lại mật khẩu

Người dùng yêu cầu đặt lại mật khẩu (nhập email) qua UserInterface.

- AuthenticationService kiểm tra User.checkUserExists trong Database.
- Nếu tồn tại:
 - ResetToken.createResetToken tạo OTP (6-8 chữ số, hết hạn 10 phút), lưu vào reset_tokens (used = false).
 - EmailService.sendOTPEmail gửi OTP qua email.
 - Nếu gửi thành công: Thông báo "Kiểm tra email để lấy OTP".
 - Nếu gửi thất bại: Thông báo "Gửi OTP thất bại, thử lại".
- Nếu không tồn tại: Thông báo "Email không tồn tại".

- Người dùng nhập OTP, mật khẩu mới qua UserInterface.
- AuthenticationService gọi ResetToken.verifyOTP kiểm tra (hợp lệ, chưa hết hạn, chưa sử dụng).
 - Nếu hợp lệ: User.updatePassword, đánh dấu used = true, xóa OTP hết hạn, thông báo "Mật khẩu đã đổi".
 - Nếu không hợp lệ:
 - Hết hạn: Thông báo "OTP hết hạn".
 - Sai: Thông báo "OTP sai".
- Yêu cầu gửi lại OTP:
 - Người dùng nhấn gửi lại qua UserInterface.
 - AuthenticationService.resendResetOTP tạo OTP mới, gửi qua EmailService.
 - Nếu thành công: Thông báo "OTP mới đã gửi, kiểm tra email".
 - Nếu thất bại: Thông báo "Gửi OTP mới thất bại, thử lại".

Xem profile

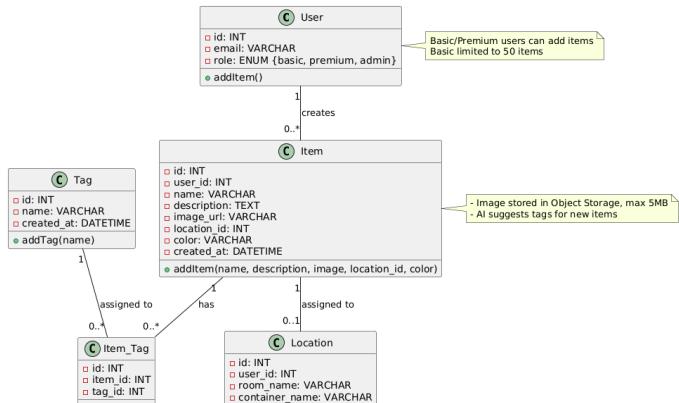
- Người dùng yêu cầu xem profile qua UserInterface (gửi userId, JWT).
- AuthenticationService.verifyJWT kiểm tra JWT hợp lệ và userId khớp.
 - Nếu hợp lệ:
 - User.getUserDetails lấy dữ liệu (email, name, role) từ Database.
 - Nếu tồn tại: UserInterface hiển thị profile.
 - Nếu không tồn tại: Thông báo "Người dùng không tồn tại".
 - Nếu không hợp lệ: Thông báo "Không có quyền truy cập".

3.2. Quản lý đồ vật (UC03, UC04)

UC03: Thêm mới đồ vật

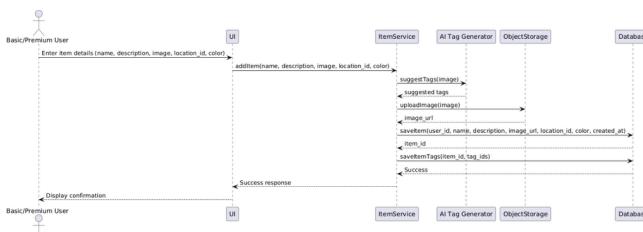
ID	Mô tả chức năng
FR2.1	Người dùng thêm mới đồ vật với các thuộc tính: tên (bắt buộc), mô tả (tùy chọn), tag (tùy chọn), ảnh (tối đa 5MB). Có AI gợi ý tag
FR2.2	Hệ thống lưu ảnh vào File System và trả URL

- Class diagram:



+ Xác định cấu trúc dữ liệu (User, Item, Location, Tag, Item_Tag) để lưu thông tin đồ vật (tên, mô tả, hình ảnh, vị trí, màu sắc, thẻ) + Giúp lập trình viên hiểu các thực thể và mối quan hệ để thiết kế cơ sở dữ liệu và logic thêm mới, đảm bảo hỗ trợ lưu trữ hình ảnh trên Object Storage và gợi ý thẻ bằng AI.

- Sequence diagram:

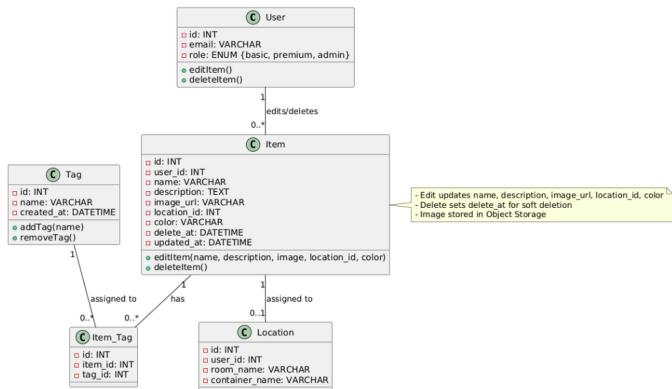


Mô tả quy trình thêm đồ vật, từ nhập liệu qua UI đến lưu vào cơ sở dữ liệu và Object Storage

UC04: Chỉnh sửa / Xóa đồ vật

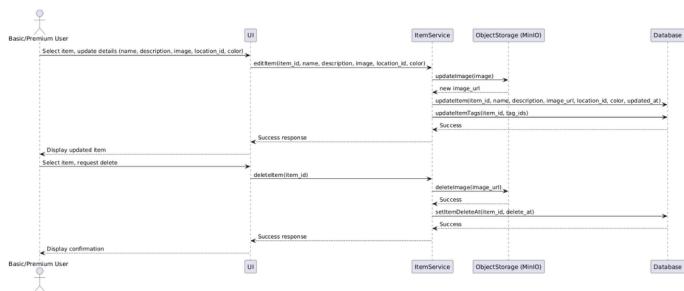
ID	Mô tả chức năng
FR2.3	Cho phép sửa thông tin và di chuyển đồ vật tới vị trí khác
FR2.4	Xoá đồ vật; sau khi xoá, dữ liệu ảnh và metadata cũng bị xoá theo cấu hình retention
FR2.5	Xem chi tiết đồ vật

- Class diagram



Cung cấp cấu trúc để chỉnh sửa/xóa đồ vật, bao gồm cập nhật image_url, location_id, và xóa mềm qua delete_at. Định nghĩa các thuộc tính và quan hệ cần thiết để lập trình chức năng chỉnh sửa/xóa, đảm bảo dữ liệu nhất quán.

- Sequence diagram



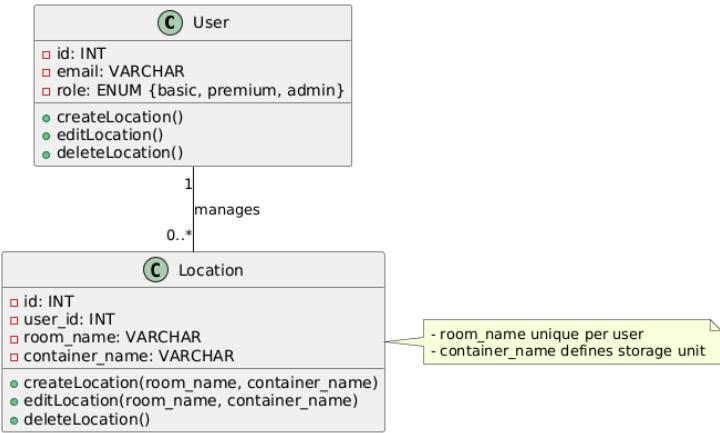
Mô tả luồng chỉnh sửa (cập nhật dữ liệu, hình ảnh) và xóa (đặt delete_at, xóa hình ảnh)

3.3. Tổ chức lưu trữ vị trí (UC05, UC06) ☈

UC05: Tổ chức không gian lưu trữ

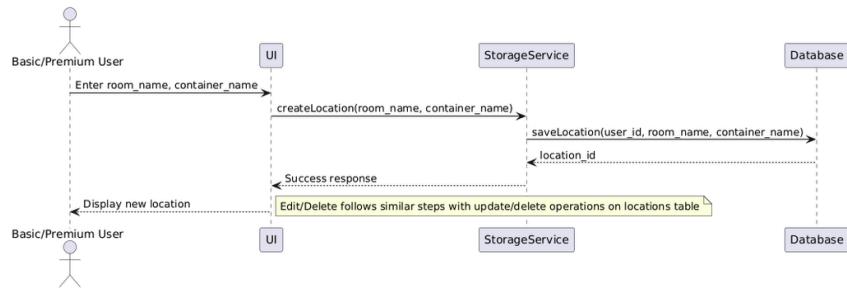
ID	Mô tả chức năng
FR3.1	Cho phép tạo/sửa/xoá phòng , mỗi phòng có tên duy nhất trên user đó.
FR3.2	Trong mỗi phòng, cho phép tạo/sửa/xoá container (kệ, ngăn kéo) với tên và vị trí (ví dụ: "Tầng dưới", "Ngăn 1").

- Class diagram



Định nghĩa User và Location để quản lý phòng và container. Giúp thiết kế cơ sở dữ liệu cho không gian lưu trữ, đảm bảo tên phòng duy nhất và hỗ trợ tạo/sửa/xóa

- Sequence diagram

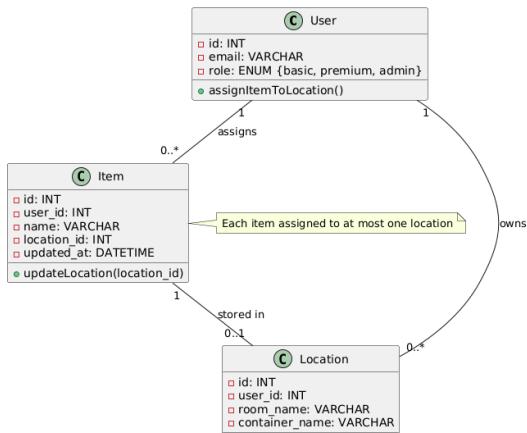


Mô tả quy trình tạo/sửa/xóa vị trí lưu trữ

UC06: Gán đồ vật vào vị trí

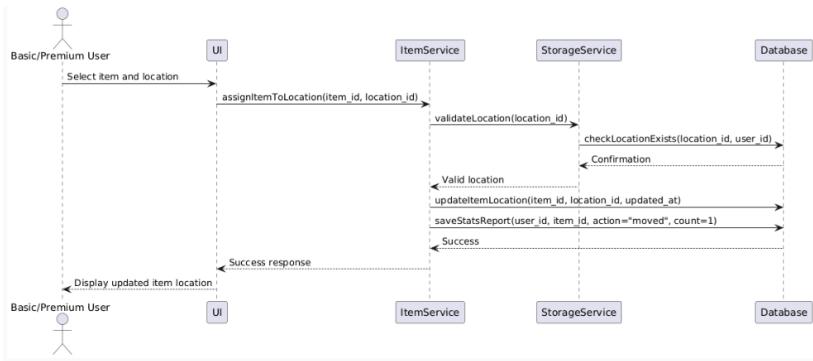
ID	Mô tả chức năng
FR3.3	Mỗi đồ vật chỉ được gán vào một container tại một thời điểm; hiển thị danh sách đồ vật theo phòng và container

- Class diagram



Mô tả User, Item, Location và quan hệ gán location_id. Định rõ cấu trúc để gán đồ vật vào vị trí, hỗ trợ lập trình logic cập nhật vị trí

- Sequence diagram



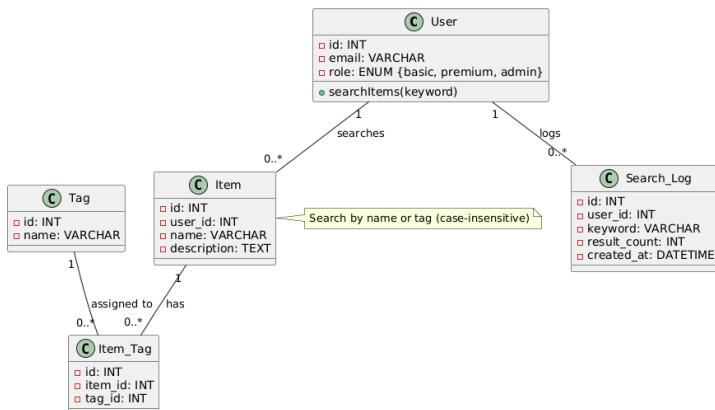
Mô tả quy trình xác thực và cập nhật vị trí, ghi log hành động "moved"

3.4. Tìm kiếm & Lọc (UC07, UC08) ⚡

UC07: Tìm kiếm cơ bản

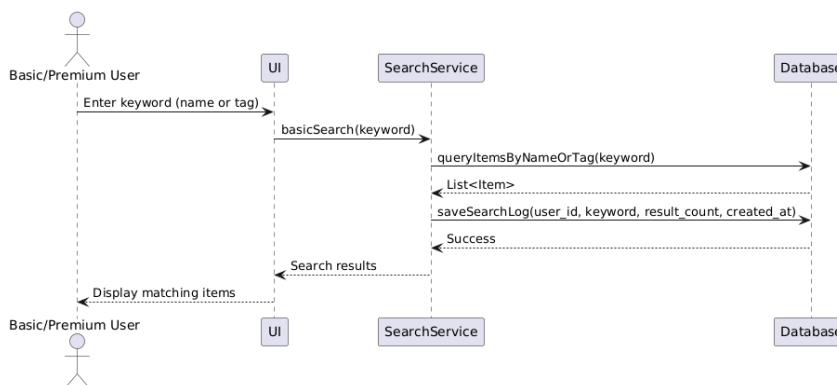
ID	Mô tả chức năng
FR4.1	Tìm kiếm cơ bản: trả về đồ vật có tên hoặc tag chứa từ khóa (case-insensitive)

- Class diagram



Bao gồm User, Item, Tag, Item_Tag, Search_Log để tìm kiếm theo tên/thẻ. Định nghĩa cấu trúc dữ liệu cho tìm kiếm cơ bản, hỗ trợ lưu lịch sử tìm kiếm.

- Sequence diagram

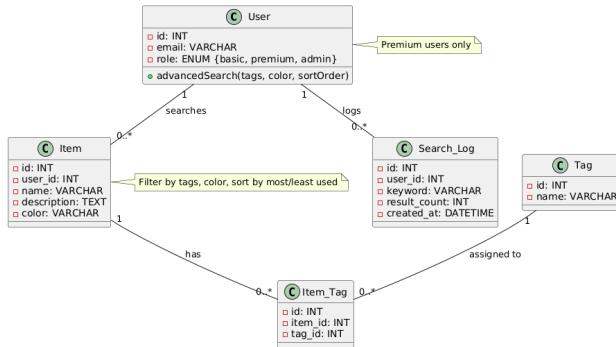


Mô tả luồng tìm kiếm và lưu log vào Search_Log. Hướng dẫn lập trình API tìm kiếm nhanh, đảm bảo ghi nhận lịch sử để phân tích sau này.

UC08: Tìm kiếm nâng cao

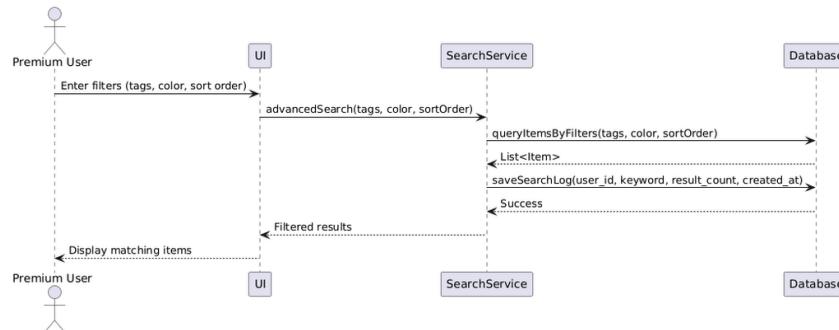
ID	Mô tả chức năng
FR4.2	Tìm kiếm nâng cao (Premium): <ul style="list-style-type: none"> Lọc theo nhiều tag cùng lúc (tags IN [...]) Lọc theo màu sắc (dựa trên thuộc tính màu) Sắp xếp theo “most recently used” hoặc “least used”

- Class diagram



Tương tự UC07, thêm hỗ trợ lọc theo màu sắc và sắp xếp, dành cho Premium User. Cung cấp cấu trúc để triển khai tìm kiếm nâng cao, hỗ trợ các bộ lọc phức tạp.

- Sequence diagram



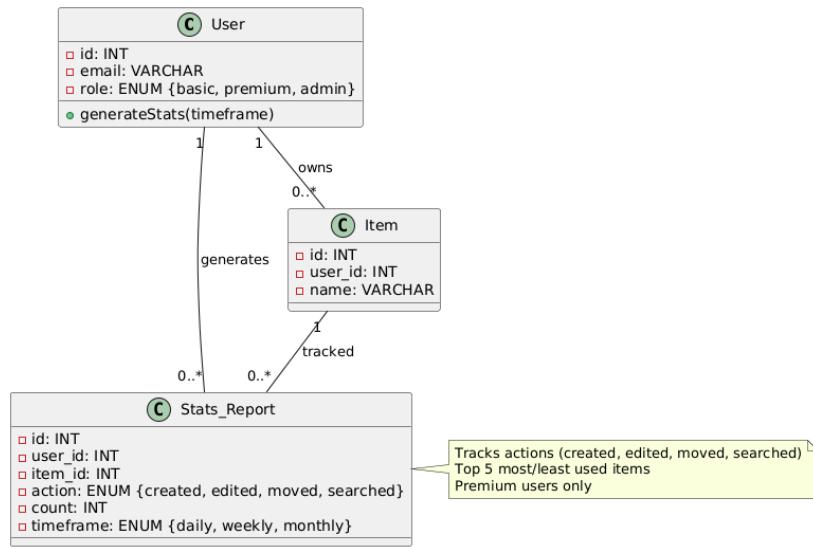
Mô tả quy trình lọc theo thẻ, màu sắc, thứ tự, và lưu log. Hướng dẫn lập trình API tìm kiếm nâng cao, đảm bảo đáp ứng yêu cầu Premium User.

3.5. Thống kê & Insight (UC9) ☘

UC09: Thống kê sử dụng

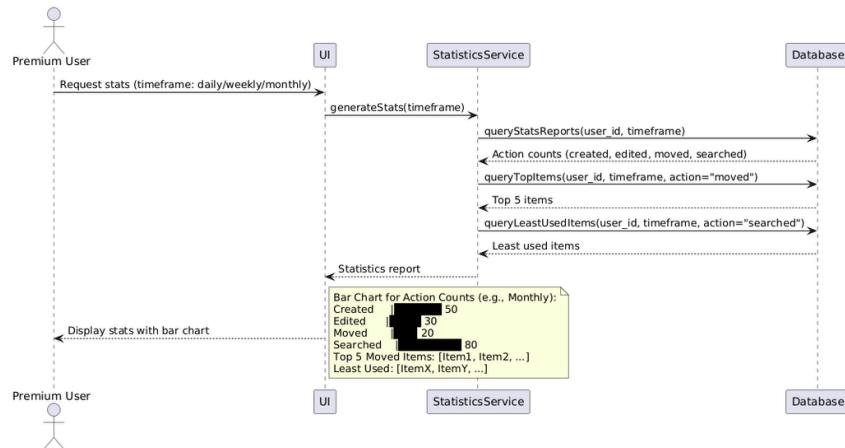
ID	Mô tả chức năng
FR5.1	Cung cấp API trả về số lượt thêm, sửa, tìm kiếm đồ vật trong khoảng thời gian (ngày/tuần/tháng)
FR5.2	Tính toán top 5 đồ vật được di chuyển nhiều nhất, top 5 đồ vật ít sử dụng nhất

- Class Diagram



Bao gồm User, Item, Stats_Report để tạo số liệu về hành động và xếp hạng đồ vật. Định nghĩa cấu trúc dữ liệu để tạo báo cáo thống kê, hỗ trợ phân tích hành vi người dùng.

- Sequence diagram

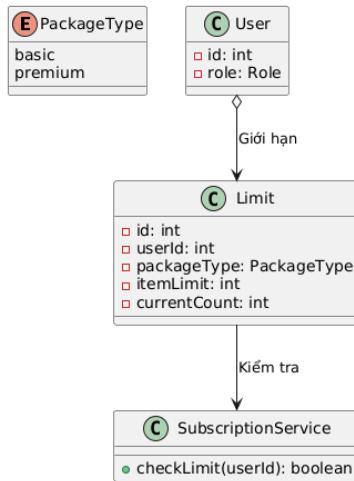


Mô tả quy trình truy vấn stats_reports để hiển thị số liệu và biểu đồ thanh. Hướng dẫn lập trình API thống kê, cung cấp dữ liệu trực quan cho Premium User.

3.6. Quản lý gói dịch vụ (UC10, UC11, UC12) ☀

- UC10 : Giới hạn số lượng đồ vật (FR6.1)

ID	Mô tả yêu cầu chức năng
FR6.1	Giới hạn số lượng đồ vật với tài khoản Basic (khoảng 50 item).

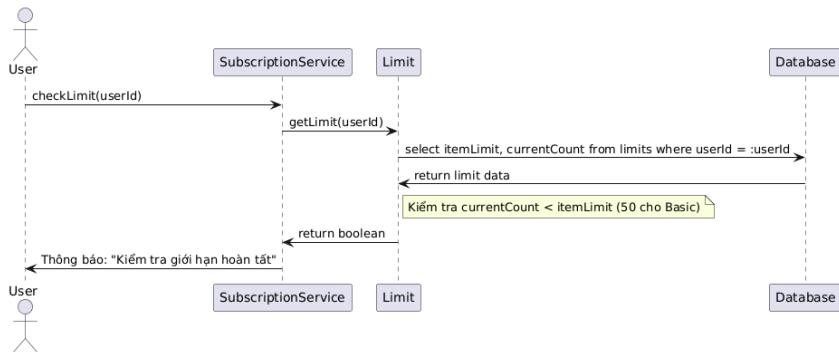


Phân tích

- **enum PackageType { basic, premium }**: Định nghĩa loại gói (package_type).
- **class User**: Đại diện bảng users.
 - **Thuộc tính**: id, role.
- **class Limit**: Đại diện bảng limits (quản lý giới hạn item).
 - **Thuộc tính**: id, userId, packageType, itemLimit (50 cho Basic), currentCount.
- **class SubscriptionService**:
 - **Phương thức**: +checkLimit(userId): boolean kiểm tra currentCount so với itemLimit (FR6.1).
- **Quan hệ**:
 - User → Limit: Mỗi người dùng có giới hạn.
 - Limit → SubscriptionService: Dịch vụ kiểm tra giới hạn.
- **Ánh xạ schema**: users, limits.
- **Đáp ứng FR**:
 - **FR6.1**: Giới hạn 50 item cho tài khoản Basic. Hỗ trợ đăng ký/hủy gói (người dùng) và tạo/cập nhật gói (admin).

Sequence Diagram

Bảng vẽ



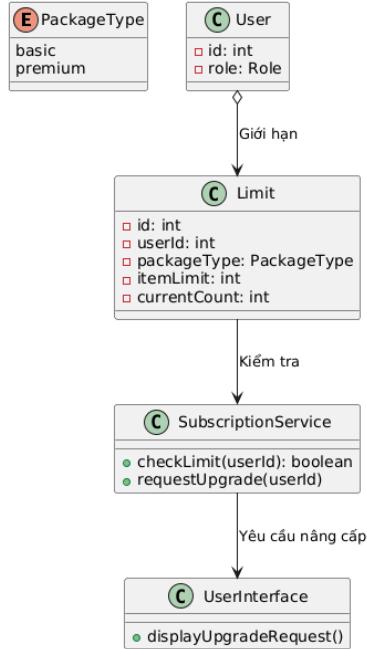
Phân tích

- **Quy trình kiểm tra giới hạn**:
 - a. **Người dùng yêu cầu kiểm tra**: UserActor gọi SubscriptionService để kiểm tra giới hạn.
 - b. **Lấy dữ liệu giới hạn**: SubscriptionService gọi Limit để lấy thông tin từ bảng limits. Ánh xạ: userId, itemLimit, currentCount.
 - c. **Kiểm tra giới hạn**: Limit so sánh currentCount với itemLimit (50 cho Basic, FR6.1).
 - d. **Trả kết quả**: Limit trả về true (chưa vượt) hoặc false (đã vượt), SubscriptionService thông báo “Kiểm tra giới hạn hoàn tất”.
- **Ánh xạ schema**: limits.
- **Đáp ứng FR**:
 - **FR6.1**: Kiểm tra giới hạn 50 item cho tài khoản Basic.

UC11: Yêu cầu nâng cấp (FR6.2)

ID	Mô tả yêu cầu chức năng
FR6.2	Khi chạm giới hạn, yêu cầu nâng cấp qua UI.

- **class Diagram**

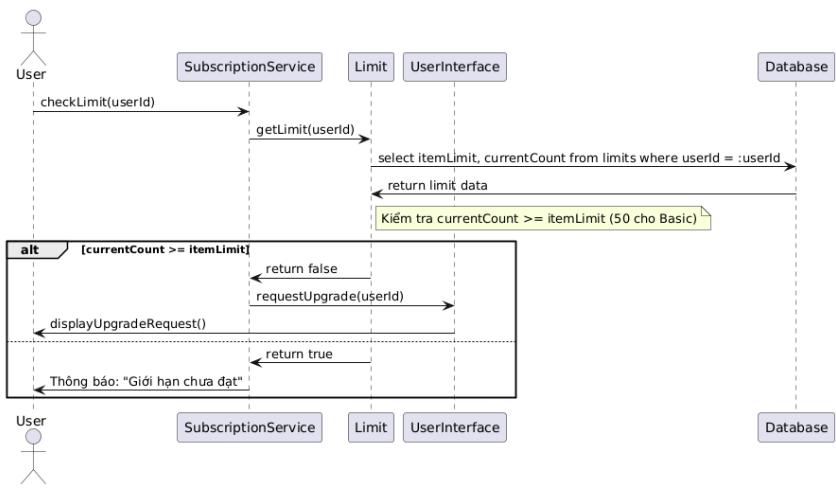


Phân tích

- **enum PackageType { basic, premium }**: Định nghĩa loại gói (package_type).
- **class User**: Đại diện bảng users.
 - **Thuộc tính**: id, role.
- **class Limit**: Đại diện bảng limits.
 - **Thuộc tính**: id, userId, packageType, itemLimit, currentCount.
- **class SubscriptionService**:
 - **Phương thức**:
 - +`checkLimit(userId): boolean`: Kiểm tra giới hạn.
 - +`requestUpgrade(userId)`: Yêu cầu nâng cấp nếu vượt giới hạn (FR6.2).
- **class UserInterface**:
 - **Phương thức**: `+displayUpgradeRequest()` hiển thị thông báo nâng cấp (FR6.2).
- **Quan hệ**:
 - User → Limit: Người dùng có giới hạn.
 - Limit → SubscriptionService: Dịch vụ kiểm tra giới hạn.
 - SubscriptionService → UserInterface: Dịch vụ yêu cầu giao diện hiển thị nâng cấp.
- **Ánh xạ schema**: users, limits.
- **Đáp ứng FR**:
 - **FR6.2**: Yêu cầu nâng cấp qua UI khi vượt giới hạn.

Sequence Diagram

Bảng vẽ



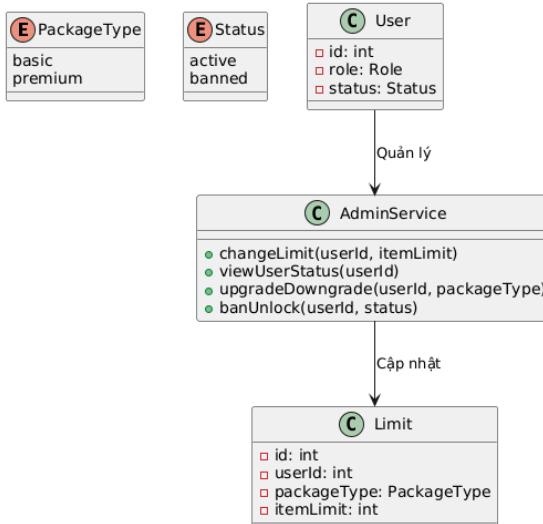
Phân tích

- Quy trình yêu cầu nâng cấp:
 - a. Người dùng yêu cầu kiểm tra:** UserActor gọi SubscriptionService để kiểm tra giới hạn.
 - b. Lấy dữ liệu giới hạn:** SubscriptionService gọi Limit để lấy thông tin từ bảng limits. Ánh xạ: userId, itemLimit, currentCount.
 - c. Kiểm tra giới hạn:** Limit kiểm tra nếu currentCount >= itemLimit (50 cho Basic).
 - d. Nếu vượt giới hạn:**
 - Limit trả về false.
 - SubscriptionService gọi UserInterface để hiển thị thông báo nâng cấp (FR6.2).
 - e. Nếu chưa vượt:** Thông báo "Giới hạn chưa đạt".
- Ánh xạ schema:** limits.
- Đáp ứng FR:**
 - FR6.2:** Hiển thị yêu cầu nâng cấp qua UI khi vượt giới hạn.
- UC12: Quản lý giới hạn bởi Admin (FR6.3)

ID	Mô tả yêu cầu chức năng
FR6.3	Admin có thể thay đổi giới hạn, xem trạng thái user, upgrade/downgrade hoặc ban/unlock.

- Class Diagram

Bảng vẽ

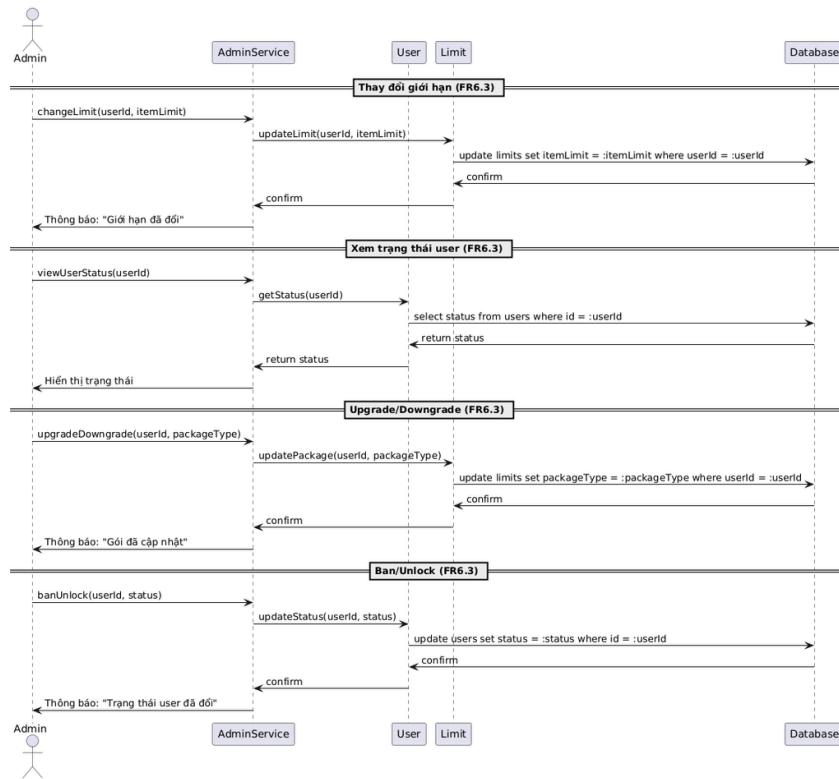


Phân tích

- **enum PackageType { basic, premium }**: Định nghĩa loại gói (package_type).
- **enum Status { active, banned }**: Định nghĩa trạng thái (status).
- **class User**: Đại diện bảng users.
 - **Thuộc tính**: id, role, status.
- **class Limit**: Đại diện bảng limits.
 - **Thuộc tính**: id, userId, packageType, itemLimit.
- **class AdminService**:
 - **Phương thức**:
 - +changeLimit(userId, itemLimit): Thay đổi giới hạn (FR6.3).
 - +viewUserStatus(userId): Xem trạng thái user (FR6.3).
 - +upgradeDowngrade(userId, packageType): Nâng cấp/hạ cấp gói (FR6.3).
 - +banUnlock(userId, status): Khóa/mở khóa user (FR6.3).
- **Quan hệ**:
 - User → AdminService: Admin quản lý user.
 - AdminService → Limit: Admin cập nhật giới hạn.
- **Ánh xạ schema**: users, limits.
- **Đáp ứng FR**:
 - **FR6.3**: Admin quản lý giới hạn, trạng thái, gói, và khóa/mở khóa.

Sequence Diagram

Bảng vẽ



Phân tích

- **Quy trình thay đổi giới hạn:**
 - Admin yêu cầu: AdminActor gọi AdminService để thay đổi itemLimit.
 - Cập nhật DB: AdminService gọi Limit để cập nhật bảng limits. Ánh xạ: userId, itemLimit.
 - Thông báo: "Giới hạn đã đổi".
- **Quy trình xem trạng thái user:**
 - Admin yêu cầu: AdminActor gọi AdminService để xem trạng thái.
 - Lấy dữ liệu: AdminService gọi User để lấy status từ bảng users. Ánh xạ: userId, status.
 - Hiển thị: Trả trạng thái cho admin.

- **Quy trình upgrade/downgrade:**
 - a. **Admin yêu cầu:** AdminActor gọi AdminService để thay đổi gói.
 - b. **Cập nhật DB:** AdminService gọi Limit để cập nhật packageType trong bảng limits. Ánh xạ: userId, packageType.
 - c. **Thông báo:** “Gói đã cập nhật”.
- **Quy trình ban/unlock:**
 - a. **Admin yêu cầu:** AdminActor gọi AdminService để thay đổi trạng thái.
 - b. **Cập nhật DB:** AdminService gọi User để cập nhật status trong bảng users. Ánh xạ: userId, status.
 - c. **Thông báo:** “Trạng thái user đã đổi”.
- **Ánh xạ schema:** users, limits.
- **Đáp ứng FR:**
 - **FR6.3:** Admin quản lý giới hạn, trạng thái, gói, và khóa/mở khóa.

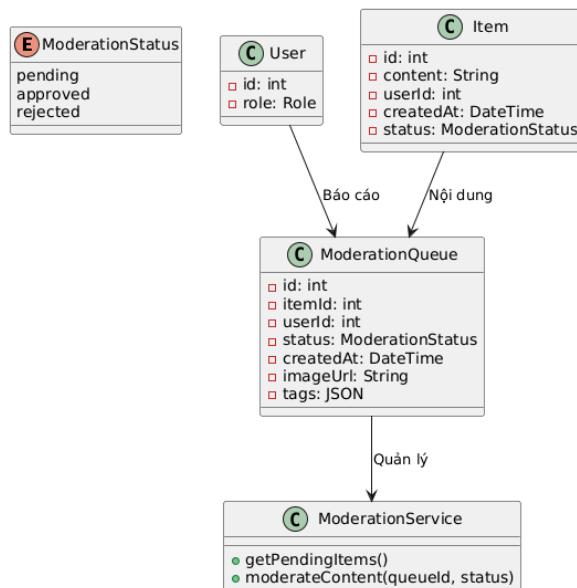
3.7. Kiểm duyệt nội dung (FR7.1, FR7.2) ↗

- **UC13: Kiểm duyệt nội dung**

ID	Mô tả yêu cầu chức năng
FR7.1	Admin có danh sách ảnh/tag đang chờ duyệt (do AI lọc ra).
FR7.2	Admin phê duyệt hoặc xoá nội dung vi phạm.

- Class Diagram

Bảng vẽ



Phân tích

- **enum ModerationStatus { pending, approved, rejected }:** Định nghĩa trạng thái kiểm duyệt (status).
- **class User:** Đại diện bảng users.
 - **Thuộc tính:** id, role.
- **class Item:** Đại diện bảng items.
 - **Thuộc tính:** id, content, userId, createdAt, status.
- **class ModerationQueue:** Đại diện bảng moderation_queue.
 - **Thuộc tính:** id, itemId, userId, status, createdAt, imageUrl, tags (FR7.1).
- **class ModerationService:**
 - **Phương thức:**
 - +`getPendingItems()`: Lấy danh sách nội dung chờ duyệt (FR7.1).
 - +`moderateContent(queueId, status)`: Phê duyệt hoặc xoá nội dung (FR7.2).
- **Quan hệ:**
 - User → ModerationQueue: Người dùng báo cáo nội dung.
 - Item → ModerationQueue: Nội dung được đưa vào hàng đợi kiểm duyệt.

- ModerationQueue → ModerationService: Dịch vụ quản lý kiểm duyệt.

- Ánh xạ schema:** users, items, moderation_queue (thêm imageUrl, tags).

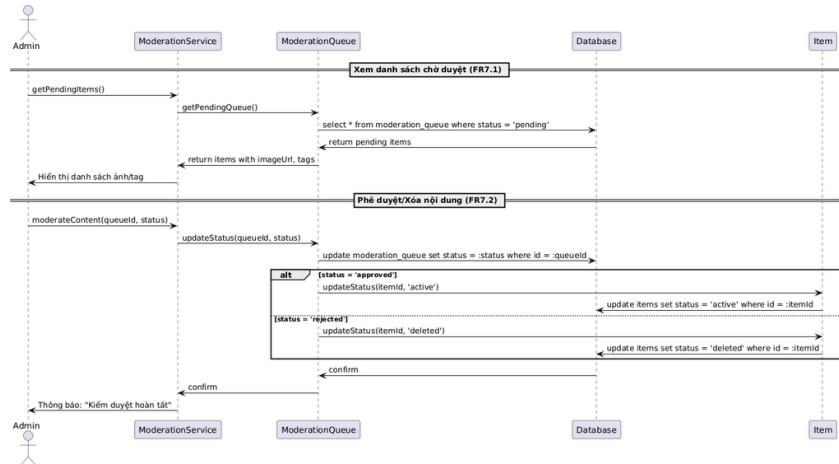
- Đáp ứng FR:**

- FR7.1:** Hỗ trợ danh sách ảnh/tag chờ duyệt.
- FR7.2:** Hỗ trợ phê duyệt/xóa nội dung.

Đáp ứng UC13: Hỗ trợ giám sát hệ thống.

Sequence Diagram

Bảng vẽ



Phân tích

- Quy trình xem danh sách chờ duyệt (FR7.1):**

- Admin yêu cầu:** AdminActor gọi ModerationService để lấy danh sách nội dung chờ duyệt.
- Lấy dữ liệu:** ModerationService gọi ModerationQueue để truy vấn bảng moderation_queue với status = 'pending'. Ánh xạ: status, imageUrl, tags.
- Hiển thị:** Trả danh sách ảnh/tag cho admin.

- Quy trình phê duyệt/xóa nội dung (FR7.2):**

- Admin yêu cầu:** AdminActor gọi ModerationService để phê duyệt hoặc xóa.
- Cập nhật hàng đợi:** ModerationService gọi ModerationQueue để cập nhật status trong bảng moderation_queue. Ánh xạ: status, id.
- Cập nhật nội dung:**

- Nếu status = 'approved': Cập nhật status = 'active' trong bảng items.
- Nếu status = 'rejected': Cập nhật status = 'deleted' trong bảng items.

- Thông báo:** "Kiểm duyệt hoàn tất".

- Ánh xạ schema:** moderation_queue, items.

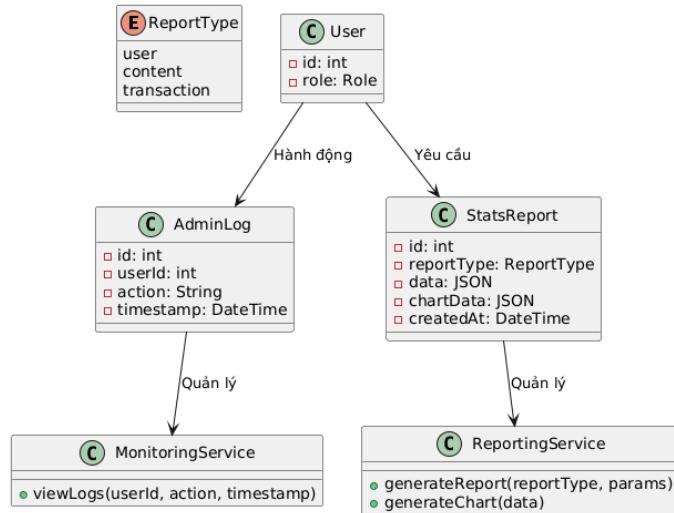
- Đáp ứng FR:**

- FR7.1:** Hiển thị danh sách ảnh/tag chờ duyệt.
- FR7.2:** Phê duyệt hoặc xóa nội dung vi phạm.

3.8. Giám sát & Báo cáo (UC14)

- Class Diagram**

Bảng vẽ

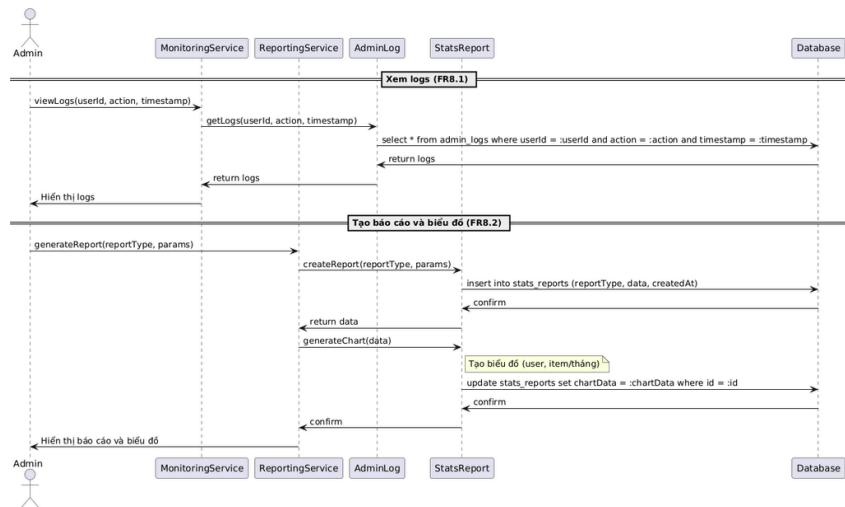


Phân tích

- enum ReportType { user, content, transaction }: Định nghĩa loại báo cáo (report_type).
- class User: Đại diện bảng users.
 - Thuộc tính: id, role.
- class AdminLog: Đại diện bảng admin_logs (lưu log hành động).
 - Thuộc tính: id, userId, action, timestamp (FR8.1).
- class StatsReport: Đại diện bảng stats_reports (lưu báo cáo).
 - Thuộc tính: id, reportType, data, chartData (FR8.2), createdAt.
- class MonitoringService:
 - Phương thức: +viewLogs(userId, action, timestamp) xem logs (FR8.1).
- class ReportingService:
 - Phương thức:
 - +generateReport(reportType, params): Tạo báo cáo.
 - +generateChart(data): Tạo biểu đồ (FR8.2).
- Quan hệ:
 - User → AdminLog: Người dùng tạo hành động.
 - AdminLog → MonitoringService: Dịch vụ quản lý log.
 - User → StatsReport: Người dùng yêu cầu báo cáo.
 - StatsReport → ReportingService: Dịch vụ quản lý báo cáo.
- Ánh xạ schema: users, admin_logs (thêm timestamp), stats_reports (thêm chartData).
- Đáp ứng FR:
 - FR8.1: Hỗ trợ xem logs.
 - FR8.2: Hỗ trợ tạo biểu đồ.

Sequence Diagram

Bảng vẽ



Phân tích

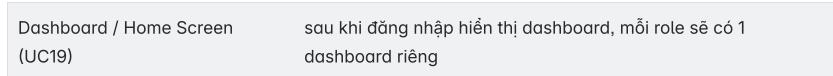
- Quy trình xem logs (FR8.1):

- Admin yêu cầu: AdminActor gọi MonitoringService để xem logs.
- Lấy dữ liệu: MonitoringService gọi AdminLog để truy vấn bảng admin_logs với các tiêu chí userId, action, timestamp. Ánh xạ: userId, action, timestamp.
- Hiển thị: Trả danh sách logs cho admin.

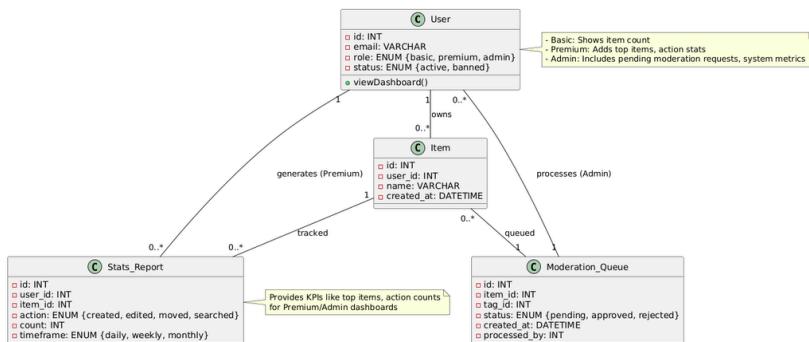
- Quy trình tạo báo cáo và biểu đồ (FR8.2):

- Admin yêu cầu: AdminActor gọi ReportingService để tạo báo cáo.
 - Lưu báo cáo: ReportingService gọi StatsReport để lưu vào bảng stats_reports. Ánh xạ: reportType, data, createdAt.
 - Tạo biểu đồ: ReportingService gọi StatsReport để tạo biểu đồ (số user, số item/tháng), cập nhật chartData trong bảng stats_reports.
 - Hiển thị: Trả báo cáo và biểu đồ cho admin.
- Ánh xạ schema: admin_logs, stats_reports.
 - Đáp ứng FR:
 - FR8.1: Xem logs theo userId, action, timestamp.
 - FR8.2: Tạo và hiển thị biểu đồ (số user, số item/tháng).

3.9. Dashboard/Home Screen (UC19) ☺

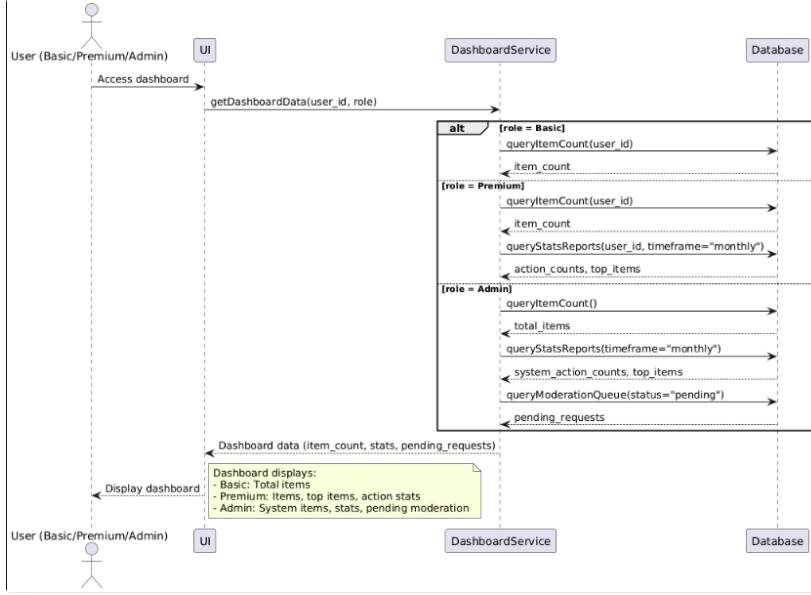


- Class diagram



Bao gồm User, Item, Stats_Report, Moderation_Queue để hiển thị KPI theo vai trò. Định nghĩa cấu trúc dữ liệu cho dashboard, hỗ trợ hiển thị số liệu phù hợp với Basic, Premium, Admin.

- Sequence diagram



Mô tả luồng truy vấn dữ liệu (số đồ vật, thống kê, yêu cầu kiểm duyệt) theo vai trò, triển khai giao diện dashboard, đảm bảo hiển thị đúng thông tin cho từng loại người dùng

4. Yêu cầu phi chức năng (Non-Functional Requirements) ☺

NFR1. Hiệu năng

- NFR1.1:** Hệ thống phải xử lý yêu cầu hiển thị danh sách đồ vật (khoảng 100 item) trong vòng < 1 giây với kết nối Internet ổn định.
- NFR1.2:** Thời gian phản hồi của API trung bình phải nhỏ hơn 500ms cho các thao tác CRUD cơ bản.
- NFR1.3:** Hệ thống hỗ trợ tối thiểu 1.000 người dùng đồng thời mà không bị gián đoạn dịch vụ.

NFR2. Tính sẵn sàng & Độ tin cậy

- NFR2.1:** Tỷ lệ uptime của hệ thống không được dưới 99.5% mỗi tháng.

NFR3. Bảo mật

- NFR3.1:** Tất cả thông tin đăng nhập, mật khẩu, token đều phải được mã hóa theo chuẩn bcrypt/SHA256.
- NFR3.2:** Hệ thống sử dụng JWT (JSON Web Token) cho cơ chế xác thực và phân quyền.
- NFR3.3:** Hỗ trợ xác thực thiết bị, đặt lại mật khẩu qua email có mã OTP.

NFR4. Khả năng mở rộng

- NFR4.1:** Hệ thống được thiết kế theo mô hình module hóa, cho phép mở rộng từng thành phần độc lập.
- NFR4.2:** Dữ liệu lưu trữ (ảnh, metadata) được tách biệt.

NFR5. Khả năng sử dụng

- NFR5.1:** Giao diện được thiết kế phù hợp cho thiết bị di động và web, tuân theo nguyên tắc UX chuẩn.
- NFR5.2:** Người dùng có thể hoàn thành thao tác “thêm đồ vật mới” trong tối đa 4 bước đơn giản.
- NFR5.3:** Hệ thống hỗ trợ thông báo lỗi/feedback rõ ràng, có trạng thái loading và xử lý ngoại lệ giao diện.

NFR6. Khả năng bảo trì

- NFR6.1:** Mã nguồn được tổ chức rõ ràng theo module, có comment và tuân thủ coding convention.
- NFR6.2:** Hệ thống sử dụng CI/CD tự động (Jenkins) để kiểm thử và triển khai.

NFR6.3: Dùng SonarQube để giám sát **code quality**, cảnh báo code smells và security risks.

5. Quy tắc nghiệp vụ (Business Rules) ☺

ID	Tên quy tắc	Mô tả chi tiết
BR01	Xác thực người dùng	Mỗi tài khoản người dùng phải được xác thực bằng email hợp lệ. Người dùng chỉ có thể truy cập hệ thống sau khi đăng nhập thành công và nhận được token hợp lệ (JWT).
BR02	Quy tắc phân quyền	Hệ thống phân quyền theo 3 vai trò: Basic User, Premium User, Admin. Mỗi vai trò chỉ được truy cập các chức năng tương ứng.

BR03	Giới hạn số lượng đồ vật (gói dịch vụ)	<ul style="list-style-type: none"> Basic User được tạo tối đa 50 đồ vật. Premium User không bị giới hạn số lượng đồ vật.
BR04	Quy tắc định danh đồ vật	Mỗi đồ vật phải có tên duy nhất trong không gian lưu trữ của một người dùng. Mỗi đồ vật chỉ tồn tại trong một container tại một thời điểm.
BR05	Quy tắc gán vị trí	Mỗi đồ vật bắt buộc phải được gán vào một container thuộc một phòng. Container phải thuộc user đó và có tên duy nhất trong mỗi phòng.
BR06	Gợi ý tag bằng AI	Khi thêm đồ vật, nếu người dùng không nhập tag, hệ thống sẽ sử dụng dịch vụ AI nội bộ để gợi ý tag dựa trên ảnh.
BR07	Lưu trữ hình ảnh	Tất cả hình ảnh được lưu trữ qua Object Storage. Kích thước ảnh tối đa là 5MB. Chỉ định dạng PNG, JPG và JPEG được chấp nhận.
BR08	Cơ chế kiểm duyệt nội dung	Nếu AI phát hiện ảnh hoặc tag chứa nội dung bất thường, hệ thống đưa vào hàng chờ duyệt. Chỉ Admin mới có quyền phê duyệt hoặc xóa nội dung vi phạm
BR09	Quy tắc đặt lại mật khẩu	Người dùng phải cung cấp email chính xác và xác minh OTP trong vòng 10 phút để thực hiện đặt lại mật khẩu.
BR10	Quy tắc thống kê & báo cáo	Hệ thống ghi log tất cả thao tác tạo, chỉnh sửa, xóa đồ vật để phục vụ chức năng thống kê và phân tích hành vi người dùng.
BR11	Dashboard phân theo vai trò	Sau khi đăng nhập, hệ thống hiển thị dashboard tương ứng với vai trò người dùng: Basic, Premium hoặc Admin.
BR12	Retention dữ liệu	Khi người dùng xóa đồ vật, hệ thống chỉ thực sự xóa metadata và hình ảnh sau thời gian retention cấu hình mặc định là 7 ngày.

6. Phân loại mức độ ưu tiên

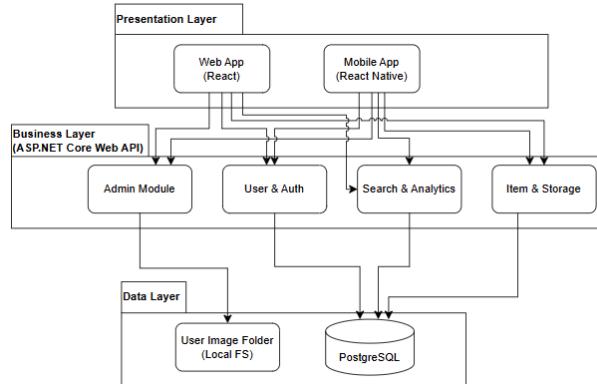
Mức độ ưu tiên	ID yêu cầu chức năng	Mô tả tóm tắt
Must have	FR1.1 – FR1.3	Đăng ký, xác thực, đăng nhập bằng JWT
	FR2.1 – FR2.4	Thêm, chỉnh sửa, xóa đồ vật và lưu ảnh
	FR3.1 – FR3.3	Tổ chức và quản lý vị trí đồ vật
	FR4.1	Tìm kiếm cơ bản (tên, tag, không phân biệt hoa thường)
	FR6.1 – FR6.2	Giới hạn tài khoản Basic, yêu cầu nâng cấp khi vượt giới hạn
Should have	FR4.2	Tìm kiếm nâng cao theo tag, màu, mức sử dụng
	FR5.1 – FR5.2	Thống kê hành vi người dùng
	FR8.1 – FR8.2	Admin xem logs và biểu đồ tổng quan hệ thống
Could have	FR7.1 – FR7.2	Kiểm duyệt nội dung AI lọc
	AI Assistance (gợi ý tag/màu)	Gợi ý thông minh khi thêm đồ vật
	Notifications	Gửi thông báo tới người dùng khi xảy ra sự kiện quan trọng (OTP).

IV. Bản mô tả thiết kế phần mềm

1. Kiến trúc phần mềm

Hệ thống được phát triển theo kiến trúc Monolithic 3-tier, bao gồm:

- Tầng trình bày (Presentation Layer): Gồm 2 giao diện
 - Web App: ReactJS (hoặc Next.js), người dùng truy cập qua trình duyệt.
 - Mobile App: React Native, cài đặt trên điện thoại.
- Tầng logic nghiệp vụ (Business Layer): [ASP.NET Core, an open-source web development framework | .NET](#) Core Web API.
- Tầng dữ liệu (Data Layer):
 - CSDL chính: PostgreSQL (lưu trữ metadata, thông tin người dùng, đồ vật, container...)
 - Lưu trữ tệp: Hệ thống thư mục nội bộ để lưu ảnh của người dùng.



1.1 Presentation Layer

Thành phần	Công nghệ	Chức năng & Mô tả
Web UI	React	<ul style="list-style-type: none"> Giao diện SPA cho admin Các màn: Đăng nhập, Dashboard, Giám sát & Thống kê, Quản lý nội dung, Cấu hình hệ thống, Quản lý người dùng. Kết nối API qua Axios/Fetch, xử lý loading, error.
Mobile UI	React Native	<ul style="list-style-type: none"> Ứng dụng mobile. Màn hình: tải ảnh, Thêm/sửa/xóa đồ vật, Tìm kiếm, Xem chi tiết, Profile.

Giao tiếp:

- Tất cả các yêu cầu đến Backend thông qua HTTP/HTTPS, JSON payload.
- Mỗi request kèm header **Authorization: Bearer <JWT>** sau khi đăng nhập.

1.2. Application & Business Layer

Module	Chức năng chính
Auth & User	<ul style="list-style-type: none"> Đăng ký, đăng nhập, sinh & validate JWT Đặt lại mật khẩu qua email/OTP Refresh token flow Xem/Sửa hồ sơ cá nhân Đổi mật khẩu Quản lý gói subscription
Item & Storage	<ul style="list-style-type: none"> CRUD đồ vật; upload ảnh Gọi ý tag bằng AI (call external AI service) Tạo/Sửa/Xóa phòng và container Gán/Di chuyển đồ vật giữa các container.
Search & Analytics	<ul style="list-style-type: none"> Tìm kiếm cơ bản và nâng cao (tag, màu, tần suất) Ghi log hành động (create/edit/delete/search) Tính toán báo cáo: top items, lượt thao tác.
Admin Module	<ul style="list-style-type: none"> Quản lý gói Basic/Premium

	<ul style="list-style-type: none"> Giới hạn số lượng đồ vật cho Basic Kiểm duyệt ảnh/tag chờ duyệt Dashboard tổng quan hoạt động
--	---

1.3. Data Layer

Loại dữ liệu	Lưu Trữ	Ghi chú
Metadata (user, items, tag, locations,...)	PostgreSQL	<ul style="list-style-type: none"> Thiết kế schema với ràng buộc PK/FK, index on user_id, item_id, full-text index cho search.
Ảnh đồ vật	File System (Server-local)	<ul style="list-style-type: none"> Thư mục cấu trúc: <code>./uploads/{userId}/{filename}</code> Lưu đường dẫn tương đối trong DB (image_path). Kích thước tối đa 5MB, kiểm tra file type (PNG/JPEG/JPG).

2. Thiết kế CSDL

2.1. Bảng Users

Lưu thông tin tài khoản người dùng và phân quyền.

Mô tả: Bảng `users` lưu trữ thông tin tài khoản người dùng trong hệ thống. Mỗi người dùng có thể có một hoặc nhiều đồ vật và có quyền truy cập khác nhau tùy thuộc vào vai trò (Basic, Premium, Admin).

Liên kết:

- `user_id` trong bảng `items` và `locations` tham chiếu đến `id` trong bảng `users`.
- `user_id` trong bảng `subscriptions`, `search_logs`, và `admin_logs` tham chiếu đến `id` trong bảng `users`.

Field Name	Type	Description	Unique	Not null	PK/FK
id	INT	Mã người dùng, khóa chính	yes	yes	PK
email	VARCHAR	Địa chỉ email người dùng	no	yes	
password	VARCHAR	Mật khẩu người dùng	no	no	
firstname	VARCHAR	Tên người dùng	no	no	
lastname	VARCHAR	Họ người dùng	no	no	
role	ENUM('basic', 'premium', 'admin')	Phân quyền của người dùng	no	no	
status	ENUM('active', 'banned')	Trạng thái tài khoản	no	no	
refresh_token	VARCHAR	Mã token làm mới để xác thực	no	no	
created_at	DATETIME	Thời gian tạo tài khoản	no	no	
updated_at	DATETIME	Thời gian cập nhật tài khoản	no	no	

2.2. Bảng Items

Lưu thông tin về các đồ vật được quản lý bởi người dùng.

Mô tả: Bảng `items` lưu trữ thông tin về các đồ vật được người dùng quản lý. Mỗi đồ vật có thể có tên, mô tả, hình ảnh, và được gán vào một vị trí cụ thể.

Liên kết:

- `user_id` tham chiếu đến `id` trong bảng `users` (mỗi đồ vật thuộc về một người dùng).
- `location_id` tham chiếu đến `id` trong bảng `locations` (mỗi đồ vật được lưu trữ tại một vị trí cụ thể).
- **Bảng item_tags** liên kết với bảng `tags` qua `item_id` và `tag_id`.

Field Name	Type	Description	Unique	Not null	PK/FK
<code>id</code>	INT	Mã đồ vật, khóa chính	yes	yes	PK
<code>user_id</code>	INT	Mã người dùng (liên kết bảng users)	no	no	FK
<code>name</code>	VARCHAR	Tên đồ vật	no	no	
<code>description</code>	TEXT	Mô tả đồ vật	no	no	
<code>image_url</code>	VARCHAR	Đường dẫn ảnh đồ vật	no	no	
<code>location_id</code>	INT	Mã vị trí (liên kết bảng locations)	no	no	FK
<code>delete_at</code>	DATETIME	Thời gian xóa (nếu có)	no	no	
<code>created_at</code>	DATETIME	Thời gian tạo đồ vật	no	no	
<code>updated_at</code>	DATETIME	Thời gian cập nhật đồ vật	no	no	

2.3. Bảng Locations

Lưu thông tin về các vị trí (phòng, kệ, ngăn kéo) nơi đồ vật được tổ chức.

Mô tả: Bảng `locations` lưu trữ thông tin về các vị trí lưu trữ đồ vật của người dùng, bao gồm các phòng, kệ, và ngăn kéo.

Liên kết:

- `user_id` tham chiếu đến `id` trong bảng `users` (mỗi vị trí lưu trữ thuộc về một người dùng).
- `id` trong bảng `locations` được tham chiếu trong bảng `items` qua `location_id`.

Field Name	Type	Description	Unique	Not null	PK/FK
<code>id</code>	INT	Mã vị trí lưu trữ	yes	yes	PK
<code>user_id</code>	INT	Mã người dùng (liên kết bảng users)	no	no	FK
<code>room_name</code>	VARCHAR	Tên phòng	no	no	
<code>container_name</code>	VARCHAR	Tên kệ/ngăn kéo	no	no	

2.4. Bảng Tags

Lưu các tag dùng để phân loại đồ vật.

Mô tả: Bảng `tags` lưu trữ các tag mà người dùng có thể gán cho đồ vật để phân loại chúng. Mỗi tag có thể được sử dụng cho nhiều đồ vật khác nhau.

Liên kết:

- **Bảng item_tags** có mối quan hệ với bảng `tags` qua `tag_id` và `item_id`.

Field Name	Type	Description	Unique	Not null	PK/FK
id	INT	Mã tag, khóa chính	yes	yes	PK
name	VARCHAR	Tên tag	no	no	
created_at	DATETIME	Thời gian tạo tag	no	no	

2.5. Bảng Items-Tags

Bảng quan hệ giữa các đồ vật và tags.

Mô tả: Đây là bảng quan hệ giữa các đồ vật và tags. Mỗi dòng trong bảng này đại diện cho một mối quan hệ giữa một đồ vật và một tag.

Liên kết:

- **item_id** tham chiếu đến **id** trong bảng **items**.
- **tag_id** tham chiếu đến **id** trong bảng **tags**.

Field Name	Type	Description	Unique	Not null	PK/FK
id	INT	Mã quan hệ giữa đồ vật và tag	yes	yes	PK
item_id	INT	Mã đồ vật (liên kết bảng items)	no	no	FK
tag_id	INT	Mã tag (liên kết bảng tags)	no	no	FK

2.6. Bảng Subscriptions

Lưu thông tin về gói dịch vụ mà người dùng đã đăng ký (Basic/Premium).

Mô tả: Bảng **subscriptions** lưu trữ thông tin về các gói dịch vụ mà người dùng đã đăng ký (Basic hoặc Premium). Thông tin này giúp quản lý các quyền lợi người dùng có được trong hệ thống.

Liên kết:

- **user_id** tham chiếu đến **id** trong bảng **users** (mỗi gói dịch vụ thuộc về một người dùng).

Field Name	Type	Description	Unique	Not null	PK/FK
id	INT	Mã đăng ký gói dịch vụ, khóa chính	yes	yes	PK
user_id	INT	Mã người dùng (liên kết bảng users)	no	no	FK
type	ENUM('basic','premium')	Loại gói dịch vụ	no	no	
start_date	DATETIME	Ngày bắt đầu gói dịch vụ	no	no	
end_date	DATETIME	Ngày kết thúc gói dịch vụ	no	no	

2.7. Bảng Search-log

Lưu lại các truy vấn tìm kiếm của người dùng.

Mô tả: Bảng **search_logs** lưu lại các tìm kiếm mà người dùng thực hiện trong hệ thống. Dữ liệu này có thể được dùng để thống kê các từ khóa tìm kiếm phổ biến hoặc phân tích hành vi người dùng.

Liên kết:

- `user_id` tham chiếu đến `id` trong bảng `users`.

Field Name	Type	Description	Unique	Not null	PK/FK
<code>id</code>	INT	Mã nhặt ký tìm kiếm	yes	yes	PK
<code>user_id</code>	INT	Mã người dùng (liên kết bảng users)	no	no	FK
<code>keyword</code>	VARCHAR	Từ khóa tìm kiếm	no	no	
<code>result_count</code>	INT	Số kết quả trả về	no	no	
<code>created_at</code>	DATETIME	Thời gian thực hiện tìm kiếm	no	no	

2.8. Bảng Stats-reports

Lưu thông tin về các báo cáo sử dụng đồ vật (thống kê cho Premium và Admin).

Mô tả: Bảng `stats_reports` lưu trữ các báo cáo thống kê sử dụng đồ vật, giúp Premium và Admin theo dõi hoạt động của người dùng (thêm, sửa, di chuyển đồ vật, v.v.).

Liên kết:

- `user_id` tham chiếu đến `id` trong bảng `users`.
- `item_id` tham chiếu đến `id` trong bảng `items`.

Field Name	Type	Description	Unique	Not null	PK/FK
<code>id</code>	INT	Mã báo cáo thống kê	yes	yes	PK
<code>user_id</code>	INT	Mã người dùng (liên kết bảng users)	no	no	FK
<code>item_id</code>	INT	Mã đồ vật (liên kết bảng items)	no	no	FK
<code>action</code>	ENUM('create','edited','moved','searched')	Hành động thực hiện	no	no	
<code>count</code>	INT	Số lần hành động xảy ra	no	no	
<code>timeframe</code>	ENUM('daily','weekly','monthly')	Khoảng thời gian báo cáo	no	no	

2.9. Bảng Admin-logs

Lưu lại các hành động của Admin trong hệ thống.

Mô tả: Bảng `admin_logs` lưu lại các hành động quản trị của Admin trong hệ thống, giúp theo dõi các thay đổi và hoạt động liên quan đến quản lý tài khoản và hệ thống.

Liên kết:

- `admin_id` tham chiếu đến `id` trong bảng `users` (chỉ Admin mới có quyền thao tác này).

Field Name	Type	Description	Unique	Not null	PK/FK

<code>id</code>	INT	Mã nhặt kí quản trị	yes	yes	PK
<code>admin_id</code>	INT	Mã Admin (liên kết bảng users)	no	no	FK
<code>action</code>	TEXT	Mô tả hành động quản trị	no	no	
<code>timestamp</code>	DATETIME	Thời gian thực hiện hành động	no	no	

2.10. Bảng moderation_queue

Lưu trữ các đồ vật hoặc nội dung cần được kiểm duyệt bởi quản trị viên trước khi hiển thị hoặc thay đổi trạng thái.

Mô tả: Bảng `moderation_queue` chứa thông tin về các mục (ví dụ: đồ vật mới tạo, đồ vật được báo cáo, tag mới) đang chờ xử lý bởi quản trị viên. Bảng này giúp theo dõi trạng thái kiểm duyệt (chờ, chấp thuận, từ chối) và người đã xử lý.

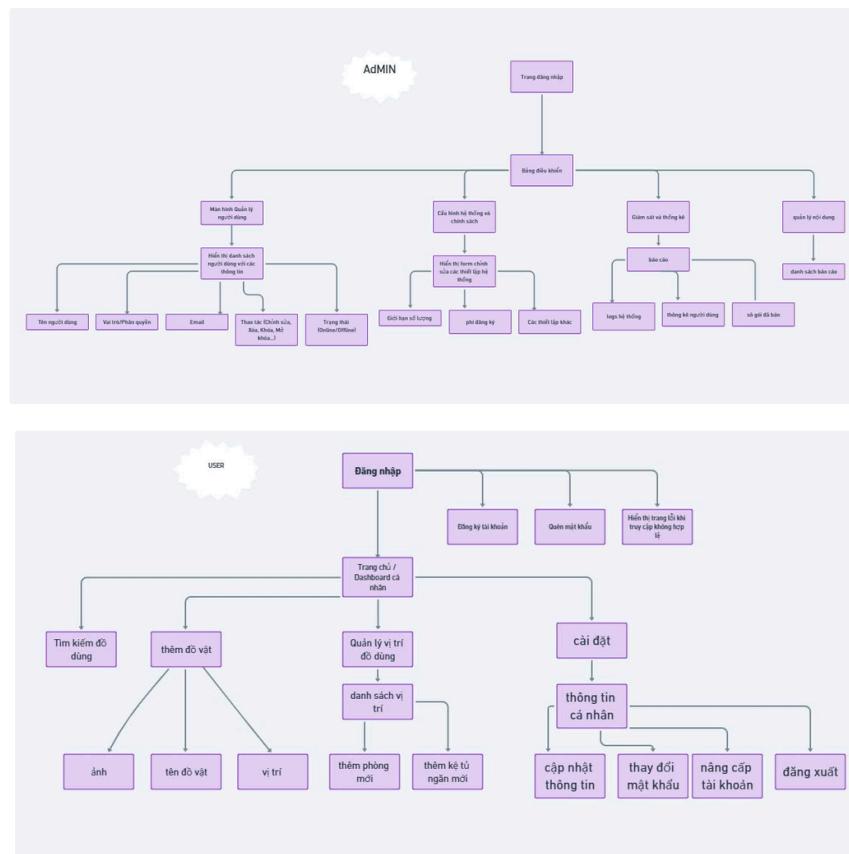
Liên kết:

- `item_id` tham chiếu đến id trong bảng items (mục cần kiểm duyệt).
- `tag_id` tham chiếu đến id trong bảng tags (có thể là NULL nếu mục kiểm duyệt không phải là tag).
- `processed_by` tham chiếu đến id trong bảng users (Admin đã xử lý yêu cầu).

Field Name	Type	Description	Unique	Not null	PK/FK
<code>id</code>	INT	Mã hàng đợi kiểm duyệt, khóa chính	yes	yes	PK
<code>item_id</code>	INT	Mã đồ vật (liên kết bảng items)	no	no	FK
<code>tag_id</code>	INT	Mã tag (liên kết bảng tags), có thể null	no	no	FK
<code>status</code>	ENUM('pending','approved','rejected')	Trạng thái kiểm duyệt	no	no	
<code>created_at</code>	DATETIME	Thời gian tạo yêu cầu kiểm duyệt	no	no	
<code>processed_by</code>	INT	Mã Admin (liên kết bảng users) đã xử lý	no	no	FK
<code>processed_at</code>	DATETIME	Thời gian Admin xử lý yêu cầu	no	no	

3. Thiết kế giao diện người dùng

3.1 Screens Flow



3.2 Screen Descriptions

	Tính Năng	Màn Hình	Mô Tả
1	Dashboard	Dashboard	Trang chủ tóm tắt nhanh các chỉ số quan trọng như số đồ dùng gần đây, phòng đã tạo, request chờ duyệt, thông báo. Có phiên bản riêng cho User (Basic/Premium) và Admin.
2	Hồ sơ cá nhân (Profile)	Trang thông tin cá nhân	Hiển thị thông tin cá nhân của user như email, gói đăng ký, tên, đổi mật khẩu, đổi tên, thêm ảnh đại diện
3	Chi tiết người dùng	quản lý người dùng	Khi Admin click vào 1 đồ vật trong "tìm kiếm đồ dùng", hiển thị Tên, email, số điện thoại Vai trò hoặc loại tài khoản (Basic, Premium, Admin...) Ngày đăng ký, trạng thái tài khoản (hoạt động, bị khóa). Các quyền hạn, lịch sử hoạt động Thông tin bổ sung như địa chỉ, gói dịch vụ đang sử dụng

4	Chi tiết đồ dùng (Item Detail)	tim kiếm đồ vật	Khi User chọn 1 đồ dùng từ kết quả tìm kiếm tên món đồ Mô tả chi tiết Hình ảnh lớn của món đồ số lượng Vị trí lưu trữ (phòng, kệ, kho...)
5	Quên mật khẩu & Đặt lại mật khẩu	Màn hình Quên mật khẩu	Màn hình hỗ trợ người dùng quên mật khẩu và đặt lại mật khẩu qua email, đi kèm màn Đăng nhập/Đăng ký.
6	Quản lý tài khoản người dùng	quản lý người dùng	Quản lý tất cả người dùng trong hệ thống, bao gồm xem và lọc tài khoản người dùng.
7	Theo dõi trạng thái đăng ký	quản lý người dùng	Phân biệt và theo dõi tài khoản Cơ bản và Premium.
8	Xử lý yêu cầu nâng cấp/hạ cấp	quản lý người dùng	Quản lý các yêu cầu nâng cấp hoặc hạ cấp tài khoản của người dùng.
9	Khóa/mở khóa tài khoản	quản lý người dùng	Quản lý việc khóa hoặc mở khóa tài khoản người dùng khi vi phạm chính sách.
10	Cấu hình phí đăng ký	cấu hình hệ thống và chính sách	Thiết lập và điều chỉnh phí cũng như chính sách liên quan đến việc đăng ký tài khoản.
11	Kiểm tra nhật ký hệ thống	giám sát và thống kê	Theo dõi và kiểm tra các hoạt động ghi lại trong hệ thống.
12	Báo cáo hiệu suất	giám sát và thống kê	Xem các báo cáo về hiệu suất hoạt động của hệ thống.
13	Phân tích sử dụng	giám sát và thống kê	Đánh giá hành vi và dữ liệu sử dụng của người dùng.
14	Kiểm duyệt nội dung	quản lý nội dung	Phê duyệt hoặc loại bỏ các nội dung không phù hợp do người dùng gửi lên, bao gồm ảnh và thè.
15	Đăng ký/Đăng nhập	Đăng ký/Đăng nhập	Cho phép người dùng đăng ký tài khoản mới hoặc đăng nhập vào hệ thống.
16	Quản lý thông tin cá nhân	thông tin cá nhân	Cập nhật thông tin cá nhân và theo dõi trạng thái tài khoản.
17	thêm đồ dùng	thêm đồ vật	Thêm mới và quản lý đồ dùng, bao gồm tên, mô tả, hình ảnh và thè.
18	Quản lý vị trí đồ dùng	danh sách vị trí	Xác định và quản lý vị trí thực tế của đồ dùng trong các phòng và thiết bị lưu trữ.
19	Tìm kiếm đồ dùng	Tim kiếm đồ dùng	Tim kiếm đồ dùng theo tên, thè hoặc màu sắc.
20	Nâng cấp tài khoản	Nâng cấp tài khoản	Nâng cấp tài khoản

3.3 Screen Authorization ↗

Screen	Admin	User Basic	User Premium
--------	-------	------------	--------------

1	Danh sách người dùng	x		
2	Trạng thái đăng ký	x		
3	Yêu cầu nâng cấp tài khoản	x		
4	Quản lý khóa tài khoản	x		
5	Thiết lập phí đăng ký	x		
6	Quản lý chính sách	x		
7	Nhật ký hệ thống	x		
8	Báo cáo hiệu suất	x		
9	Phân tích sử dụng	x		
10	Kiểm duyệt nội dung	x		
11	Đăng ký/Đăng nhập	x	x	x
12	Quản lý thông tin cá nhân	x	x	x
13	Tạo và quản lý đồ dùng	x	x	x
14	Quản lý vị trí đồ dùng	x	x	x
15	Tìm kiếm đồ dùng	x	x	x
16	Nâng cấp tài khoản	x	x	
17	Tìm kiếm nâng cao	x		x
18	Phân tích chi tiết	x		x

4. Mô tả các API có trong hệ thống

V. Tài liệu kiểm thử phần mềm

1. Phạm vi kiểm thử

Xác định giới hạn và nội dung sẽ được kiểm thử cho HomeTrack phiên bản MVP:

Chức năng sẽ kiểm thử:

- Quản lý tài khoản: đăng ký, đăng nhập/dăng xuất, reset mật khẩu, xem profile.
- Quản lý đồ vật: CRUD đồ vật, upload ảnh, gợi ý tag AI, xem chi tiết.
- Tổ chức vị trí lưu trữ: tạo/sửa/xóa phòng & container, gán/di chuyển đồ vật.
- Tìm kiếm & lọc: tìm kiếm cơ bản, tìm kiếm nâng cao (Premium).
- Thống kê & Insights: API trả báo cáo usage, top items.
- Quản lý gói dịch vụ: giới hạn Basic, nâng cấp/downgrade.
- Kiểm duyệt & Admin: xem queues, phê duyệt/xóa, xem logs, biểu đồ.
- Dashboard: load dữ liệu đúng theo role, thời gian phản hồi chấp nhận được.

Ngoại trừ:

- Tính năng AI Assistance (nâng cao gợi ý tag/màu).
- Tích hợp cổng thanh toán thực tế.

2. Chiến lược kiểm thử

Cấp độ	Mục Tiêu	Công cụ/Phương pháp
--------	----------	---------------------

Unit Test	Kiểm tra logic tập trung trong từng hàm, module	xUnit (C#)
Integration Test	Kiểm tra tương tác giữa các module (API ↔ DB, API ↔ FileSystem)	Swagger
UI Test	Kiểm tra flow cơ bản trên Web & Mobile	Test bằng tay
Performance Test	Kiểm tra thời gian phản hồi API, load 100 request/giây	K6
Security Test	Kiểm tra lỗ hổng cơ bản (SQLi, XSS, auth bypass)	OWASP ZAP
UAT (User Acceptance Testing)	Người dùng nội bộ test tính trải nghiệm, tính đúng đắn	Checklist + Feedback form

3. Các trường hợp kiểm thử ↴

TC_ID	Tên Test Case	Mô Tả	Bước kiểm thử	Kỳ vọng	Loại
TC01	Đăng ký thành công	Đăng ký với email & mật khẩu hợp lệ	1. Mở màn hình Đăng ký 2. Nhập email hợp lệ, mật khẩu ≥ 8 ký tự 3. Submit	Hiển thị “Đăng ký thành công”, email OTP được gửi	Functional
TC01	Đăng nhập không thành công	Sai mật khẩu	1. Mở màn hình Đăng nhập 2. Nhập email đúng, mật khẩu sai 3. Submit	Hiển thị lỗi “Email hoặc mật khẩu không đúng”	Functional
TC01	Thêm đồ vật với ảnh > 5MB	Upload ảnh vượt kích thước	1. Mở màn hình Thêm đồ vật 2. Chọn ảnh > 5MB 3. Submit	Hiển thị lỗi “Kích thước ảnh tối đa 5MB”	Functional
TC01	Tìm kiếm cơ bản	Tim theo tên	1. Đăng nhập Basic user 2. Vào Search 3. Nhập keyword tồn tại 4. Submit	Danh sách trả về bao gồm item chứa keyword	Functional
TC01	Tìm kiếm nâng cao	Lọc tag & màu	1. Đăng nhập Premium 2. Vào Search nâng cao 3. Chọn 2 tag, 1 màu 4. Submit	Danh sách kết quả thỏa mãn các điều kiện lọc	Functional
TC01	API thống kê sử dụng	Lấy báo cáo theo tháng	Gọi GET /api/stats/usage? period=monthly với token Admin	Trả về JSON với số lượt create/edit/search ngày tháng qua	Integration
TC01	UI Dashboard Admin	Load chart	1. Login Admin 2. Vào Dashboard 3. Quan sát biểu đồ	Biểu đồ user đăng ký, item tạo mới hiển thị đúng số liệu	UI

TC01	Load 100 concurrent search requests	Kiểm tra hiệu năng	Sử dụng K6 để send 100 request / giây vào endpoint /api/search	95% request phản hồi < 500ms	Performance
TC01	SQL Injection	Bảo mật endpoint login	Gửi payload ' OR '1='1 ' vào mật khẩu	API trả về lỗi 401, không có SQL injection.	Security

4. Báo cáo kết quả kiểm thử ↴

Chưa có dữ liệu

VI. Tài liệu hướng dẫn triển khai phần mềm ↴

Chưa làm xong nên chưa có hướng dẫn chi tiết

1. Yêu cầu hệ thống ↴

2. Cách cài đặt hệ thống ↴

3. Cấu hình phụ trợ ↴

4. Khởi động và truy cập ↴