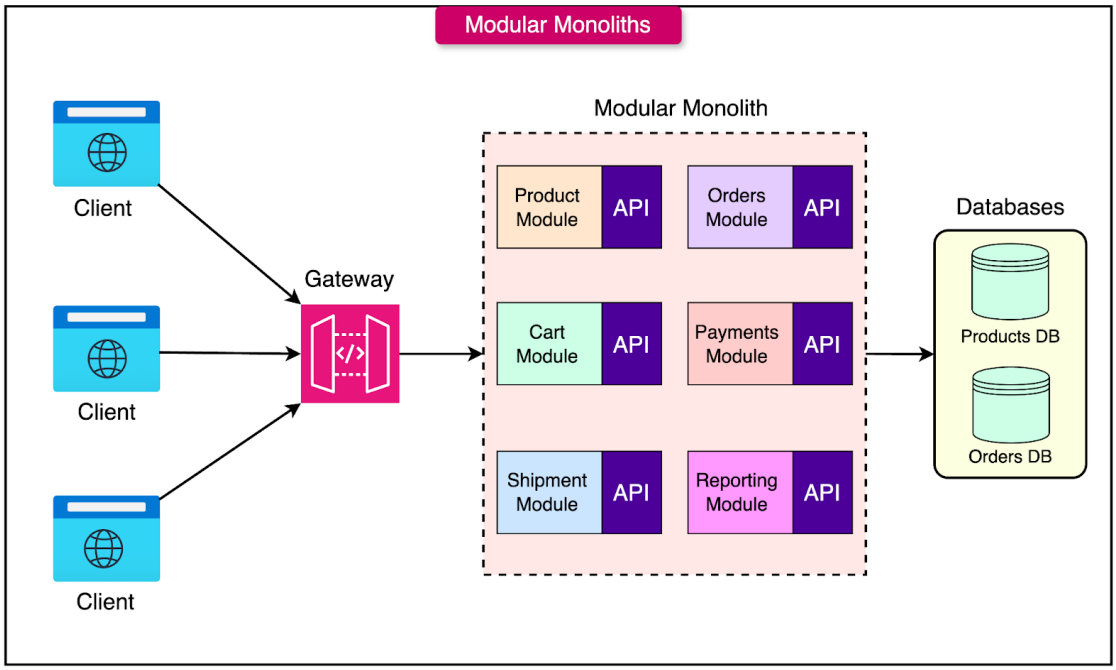
0.1 Thiết kế kiến trúc

0.1.1 Lựa chọn kiến trúc phần mềm

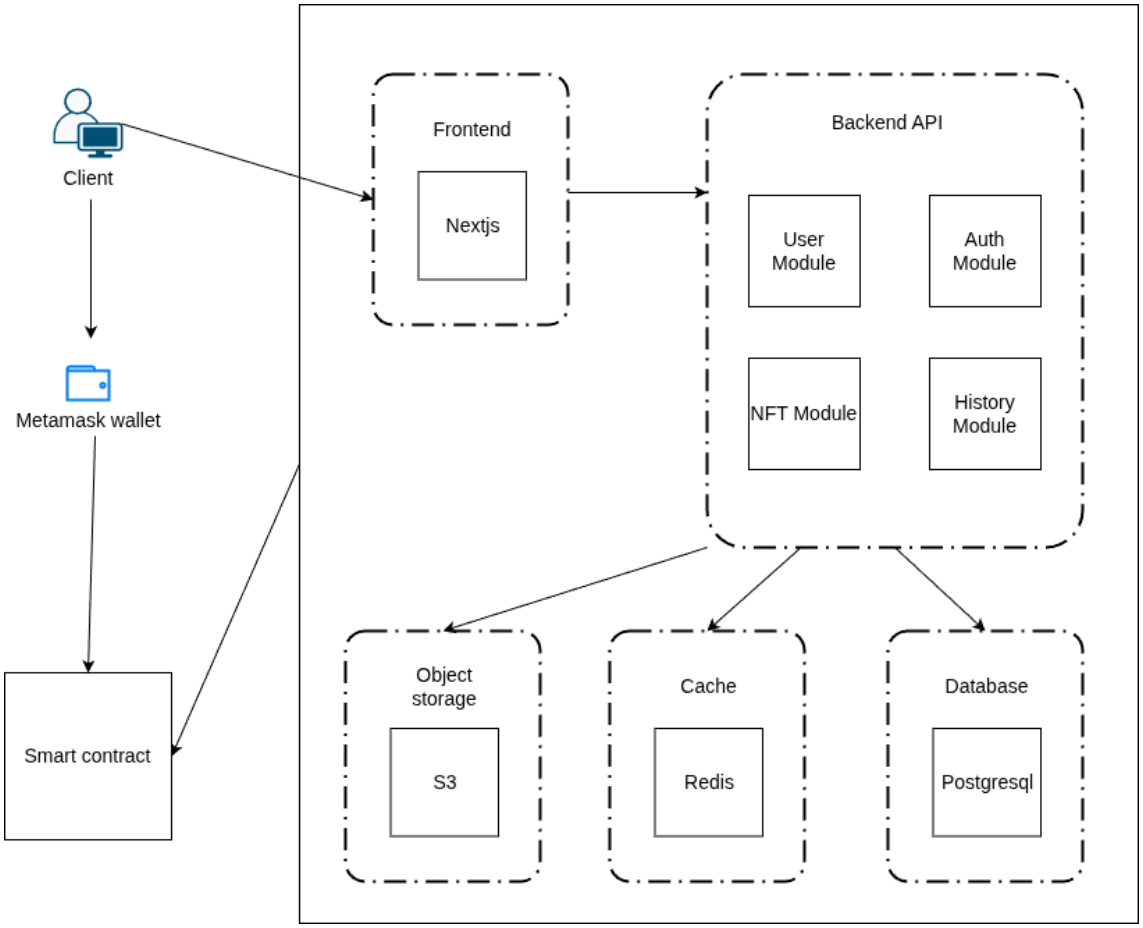
Kiến trúc phần mềm có thể được hiểu là cách tổ chức tổng thể của một hệ thống, bao gồm các thành phần chính, mối quan hệ giữa các thành phần và các nguyên tắc chi phối việc thiết kế và triển khai. Việc lựa chọn kiến trúc phù hợp giúp hệ thống có cấu trúc rõ ràng, dễ phát triển, dễ bảo trì và thuận lợi cho việc mở rộng trong tương lai. Trong thực tế, có nhiều phong cách kiến trúc được áp dụng tùy theo mục tiêu và bối cảnh như MVC/MVP trong ứng dụng hướng giao diện, kiến trúc phân lớp, Clean Architecture tập trung tách biệt business logic với chi tiết triển khai, hoặc Microservice hướng đến triển khai độc lập và khả năng mở rộng ở quy mô lớn.

Trong phạm vi đồ án này, em lựa chọn kiến trúc Modular Monolith. Đây là hướng tiếp cận trong đó hệ thống được triển khai như một ứng dụng thống nhất (monolith) nhằm đơn giản hóa quá trình triển khai và vận hành, đồng thời được chia thành các module theo nghiệp vụ (modular) để đảm bảo tính tổ chức, cô lập và dễ bảo trì. Mỗi module đóng vai trò quản lý tập trung các thành phần liên quan đến nghiệp vụ đó, nhờ vậy hạn chế tình trạng phụ thuộc chồng chéo và giúp mở rộng tính năng theo từng phần mà không ảnh hưởng đến các phần tính năng khác. Có thể xem Modular Monolith là sự cân bằng giữa monolith truyền thống và microservice. Không những phù hợp với điều kiện thời gian, nguồn lực và mục tiêu hoàn thiện các luồng nghiệp vụ của đồ án mà còn dễ dàng mở rộng nâng cấp khi hệ thống phát triển trong tương lai.



Hình 0.1: Ví dụ kiến trúc Modular monolith

Dựa trên kiến trúc đã lựa chọn, hệ thống NFT Marketplace được thiết kế theo mô hình client–server kết hợp blockchain được biểu thị bằng hình vẽ dưới đây.



Hình 0.2: Kiến trúc của hệ thống NFT marketplace

Ở phía client, người dùng thao tác trên giao diện web (Next.js/React) và tương tác với ví (Wallet Extension) để ký và gửi giao dịch blockchain. Các nghiệp vụ quan trọng như tạo, mua/bán NFT được thực hiện on-chain thông qua smart contract triển khai trên blockchain. Smart contract là thành phần nằm trên mạng blockchain, có nhiệm vụ thực thi logic giao dịch và phát sinh các sự kiện (events/logs) làm căn cứ xác minh giao dịch.

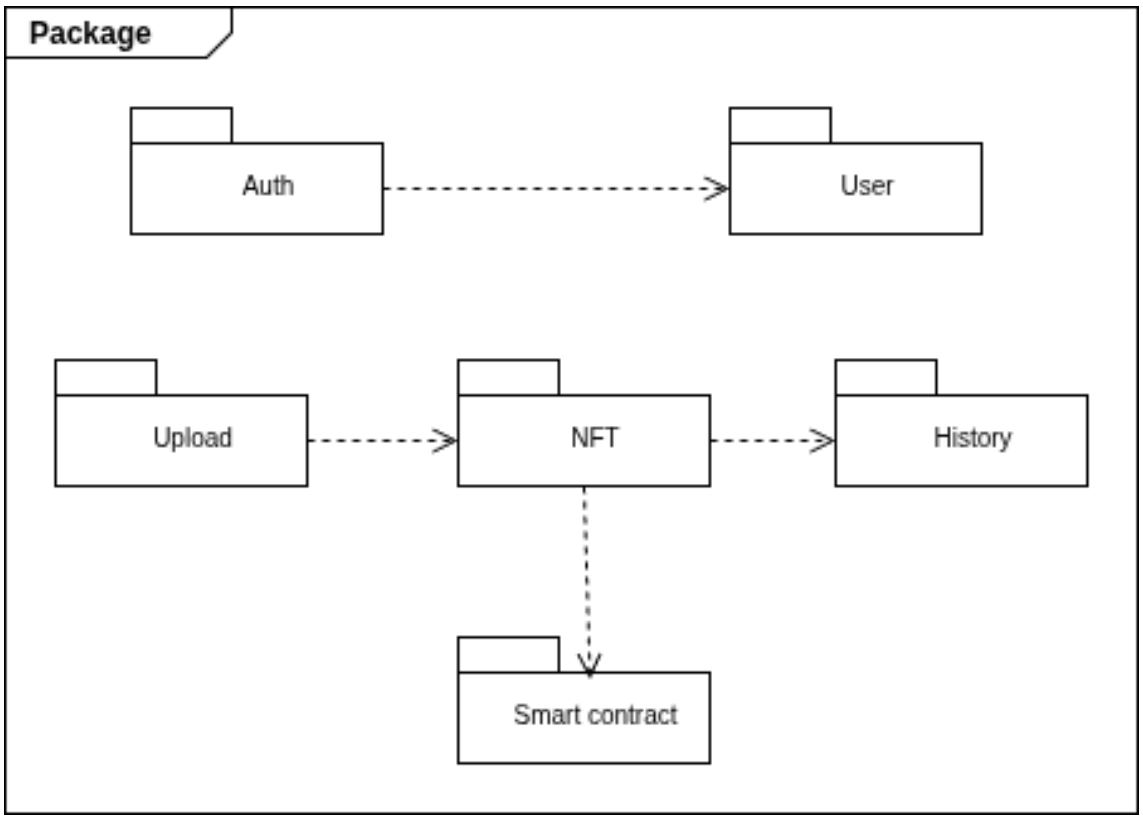
Ở phía server, backend được triển khai như một ứng dụng monolith và được chia thành các module nghiệp vụ chính gồm: Auth, User, NFT và History. Trong đó, Auth phụ trách xác thực và luồng kết nối ví; User quản lý thông tin hồ sơ người dùng; NFT xử lý nghiệp vụ liên quan NFT và trạng thái hiển thị; History ghi nhận các sự kiện và lịch sử giao dịch phục vụ hiển thị lịch sử và thống kê cơ bản. Với cách chia này, mỗi module có phạm vi trách nhiệm rõ ràng, dễ phát triển theo từng nhóm chức năng và thuận lợi cho việc bảo trì, mở rộng.

Về lưu trữ dữ liệu, hệ thống sử dụng PostgreSQL làm cơ sở dữ liệu chính cho dữ liệu nghiệp vụ, kết hợp Redis làm lớp cache để tối ưu các truy vấn đọc nhiều như danh sách NFT/collection, chi tiết NFT/collection và một số dữ liệu thống kê. Tài nguyên ảnh và metadata được lưu trên Amazon S3, đồng thời hệ thống có module upload hỗ trợ tải ảnh lên S3 để giảm tải cho backend và tối ưu tốc độ upload.

Một điểm thiết kế quan trọng của đồ án là cơ chế đồng bộ giữa dữ liệu on-chain và off-chain. Cụ thể, client thực hiện giao dịch blockchain qua ví và nhận txHash. Sau đó, client gọi API lên backend để lưu giữ liệu. Backend sẽ truy vấn blockchain thông qua RPC provider để kiểm tra giao dịch đã thành công hay chưa, có đúng smart contract và dữ liệu trong event có khớp với yêu cầu nghiệp vụ hay không. Chỉ khi xác minh hợp lệ, backend mới cập nhật dữ liệu vào PostgreSQL, ghi nhận lịch sử vào History module và cập nhật cache Redis khi cần. Cách làm này giúp giảm sai lệch trạng thái hiển thị off-chain so với trạng thái thực tế on-chain, đồng thời nâng cao tính tin cậy của hệ thống.

0.1.2 Thiết kế tổng quan

Với thiết kế theo hướng tách thành các module, mỗi module sẽ là một package riêng.



Hình 0.3: Biểu đồ tổng quan gói

Dưới đây là mô tả sơ lược về từng package:

Auth: Quản lý xác thực người dùng dựa trên ví. Package này chịu trách nhiệm kiểm tra chữ ký, tạo/duy trì token phiên đăng nhập.

User: Quản lý thông tin người dùng trong hệ thống. Package này lưu trữ và cập nhật hồ sơ người dùng.

NFT: Chịu trách nhiệm tạo, quản lý trạng thái và mua bán NFT.

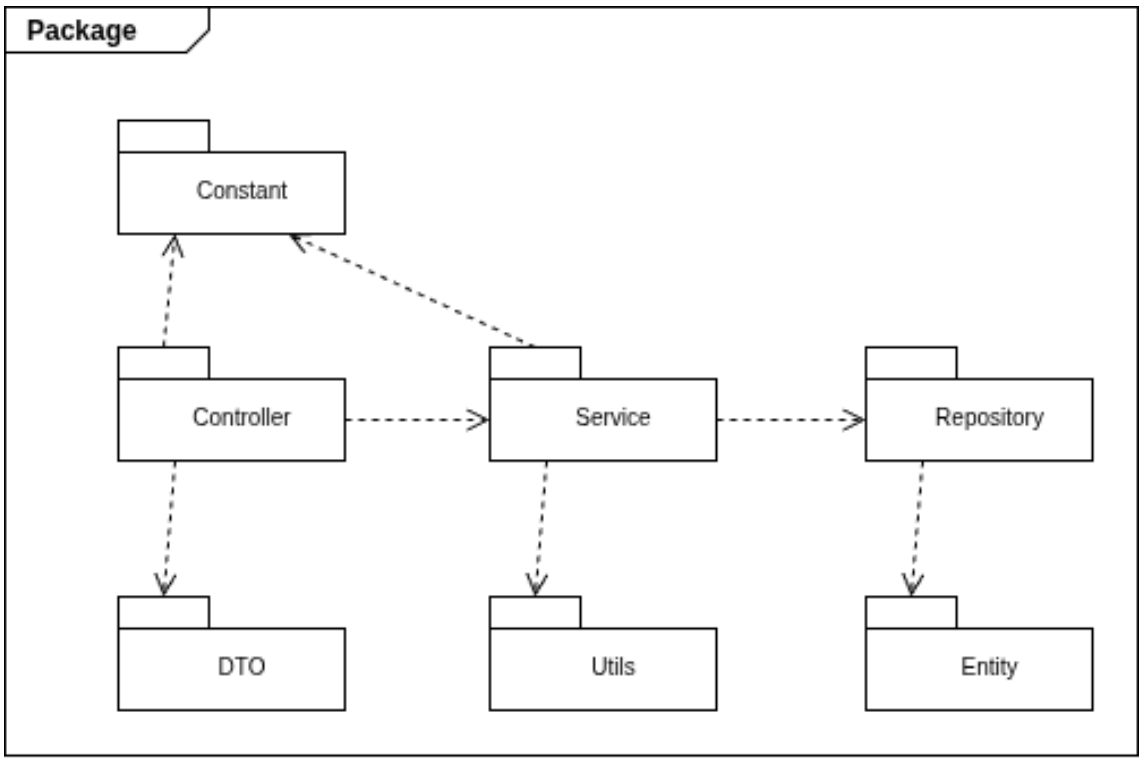
Upload: Quản lý upload và lưu trữ tài nguyên đa phương tiện.

History: Ghi nhận và truy vấn lịch sử hoạt động/giao dịch. Package này lưu các sự kiện khi phát sinh giao dịch trên smart contract, phục vụ màn hình lịch sử, thống kê cơ bản theo thời gian.

SmartContract: Giúp hệ thống tương tác với blockchain và smart contract.

0.1.3 Thiết kế chi tiết gói

Các package được tổ chức theo hướng module nên kiến trúc mỗi gói là tương tự nhau, dưới đây là chi tiết về package



Hình 0.4: Biểu đồ chi tiết gói

Controller: Đảm nhận vai trò tiếp nhận và xử lý các yêu cầu từ phía client thông qua các API. Nhiệm vụ chính của gói này là định tuyến yêu cầu đến đúng chức năng, kiểm tra/chuẩn hoá dữ liệu đầu vào và trả về phản hồi theo định dạng thống nhất. Controller không tập trung triển khai nghiệp vụ, mà đóng vai trò kết nối giữa client và hệ thống.

Service: Là nơi triển khai các nghiệp vụ cốt lõi của từng module. Gói này chịu trách nhiệm điều phối luồng xử lý: gọi các lớp truy cập dữ liệu, kết hợp các bước xử lý nghiệp vụ, áp dụng các quy tắc/điều kiện, và phối hợp với các module liên quan khi cần. Có thể xem service là tầng xử lý chính giúp đảm bảo nghiệp vụ được thực thi đúng và nhất quán.

Repository: Phụ trách tương tác với nguồn dữ liệu và các thành phần hạ tầng lưu trữ. Mục đích chính của gói này là đóng gói các thao tác đọc/ghi dữ liệu, giúp tầng service không phụ thuộc trực tiếp vào chi tiết triển khai như ORM, câu truy vấn SQL hay cơ chế lưu trữ cụ thể. Nhờ đó, hệ thống dễ bảo trì và dễ thay đổi công nghệ lưu trữ khi cần.

Entity: Mục đích của gói này là chuẩn hoá cấu trúc dữ liệu, quan hệ giữa các đối tượng, và làm nền tảng để các tầng phía trên thao tác một cách thống nhất, tránh việc xử lý dữ liệu rời rạc, thiếu nhất quán.

Utils: Chứa các thành phần hỗ trợ dùng chung trong phạm vi module. Mục đích

của utils là giảm trùng lặp mã nguồn, tăng khả năng tái sử dụng và giữ cho service tập trung vào nghiệp vụ chính.

DTO: Định nghĩa các cấu trúc dữ liệu dùng để trao đổi giữa client và server. Mục đích chính là đảm bảo cấu trúc dữ liệu rõ ràng, hỗ trợ kiểm tra tính hợp lệ của dữ liệu đầu vào và chuẩn hoá dữ liệu đầu ra. Việc tách DTO khỏi entity giúp tránh lộ cấu trúc nội bộ và tăng tính ổn định của API.

Constant: Tập hợp các hằng số và giá trị cấu hình logic được sử dụng xuyên suốt module. Mục đích của gói này là hạn chế việc cấu hình mã nguồn rải rác trong hệ thống, đảm bảo tính nhất quán và giúp việc thay đổi/điều chỉnh trở nên thuận tiện, an toàn hơn.

0.2 Thiết kế chi tiết

0.2.1 Thiết kế giao diện

Phần giao diện của hệ thống NFT Marketplace được thiết kế theo hướng responsive nhằm đảm bảo khả năng sử dụng tốt trên cả máy tính (desktop/laptop) và thiết bị di động (mobile). Mục tiêu của thiết kế là tạo trải nghiệm nhất quán, dễ thao tác, đồng thời hỗ trợ người dùng thực hiện các chức năng chính như connect wallet, tạo NFT, mua/bán NFT trên nhiều kích thước màn hình khác nhau.

Thông tin về màn hình mà ứng dụng hướng tới

Hệ thống hướng đến các thiết bị phổ biến trên thị trường hiện nay, bao gồm máy tính và điện thoại thông minh. Các độ phân giải dưới đây được lựa chọn do có tần suất sử dụng cao và đại diện tốt cho phần lớn thiết bị người dùng.

STT	Loại thiết bị	Kích thước phổ biến	Độ phân giải mục tiêu
1	Máy tính để bàn / Laptop	13” – 15.6”, 21” trở lên	1366×768; 1920×1080; 2560×1440
2	Điện thoại thông minh	5” – 6.9”	1280×720; 1920×1080

Bảng 1: Bảng mô tả thông tin về màn hình mà hệ thống hướng tới

Chuẩn hoá thiết kế giao diện

Để đảm bảo tính thống nhất giữa các màn hình và giảm sai lệch trong quá trình triển khai, đồ án đưa ra các quy tắc chuẩn hoá giao diện được thiết kế theo bố cục linh hoạt, thay đổi theo độ rộng màn hình (breakpoint). Các breakpoint được sử dụng theo nhóm phổ biến để đảm bảo tương thích tốt:

Nhóm màn hình	Độ rộng (tham khảo)	Nguyên tắc bố cục
Mobile	$\leq$ 576px	Bố cục một cột, ưu tiên nội dung chính, nút hành động rõ ràng, thao tác chạm thuận tiện
Tablet nhỏ	577–768px	Hai cột đơn giản, giảm mật độ thông tin
Tablet lớn / Laptop	769–1024px	Ba cột, bắt đầu hiển thị thêm khu vực lọc/sắp xếp khi phù hợp
Desktop	1025–1440px	Nhiều cột, danh sách NFT hiển thị dạng lưới (grid) để tăng khả năng quan sát
Desktop lớn	$>$ 1440px	Tăng khoảng trắng, giữ chiều rộng nội dung hợp lý, tránh dàn trải quá rộng

Bảng 2: Các breakpoint và nguyên tắc bố cục responsive

Nguyên tắc tổng quát:

Trên desktop/laptop, danh sách NFT và collection hiển thị theo dạng lưới (grid) để tăng khả năng quan sát.

Trên mobile, giao diện chuyển sang một cột theo chiều dọc; thẻ NFT được thiết kế đủ lớn để dễ thao tác.

Các thông tin trọng tâm như hình ảnh NFT, tên NFT/collection, giá, trạng thái và nút hành động được ưu tiên đặt ở vùng dễ nhìn nhằm giảm số thao tác cuộn.

Chuẩn hoá phối màu và hiển thị chữ

Giao diện sử dụng tông màu xám và trắng làm nền chủ đạo nhằm tạo cảm giác hiện đại, tối giản và làm nổi bật nội dung hình ảnh. Màu xanh nước biển được sử dụng làm màu nhấn cho các hành động quan trọng và các thành phần cần thu hút sự chú ý.

Quy ước màu chữ:

Chữ màu xám cho nội dung thông thường và mô tả phụ.

Chữ màu xanh nước biển cho nội dung cần nhấn mạnh như liên kết, trạng thái, hoặc điểm nổi bật.

Chữ màu trắng dùng trên nền tối hoặc trên nút hành động để đảm bảo độ tương phản.

Chuẩn hoá nút bấm và điều khiển

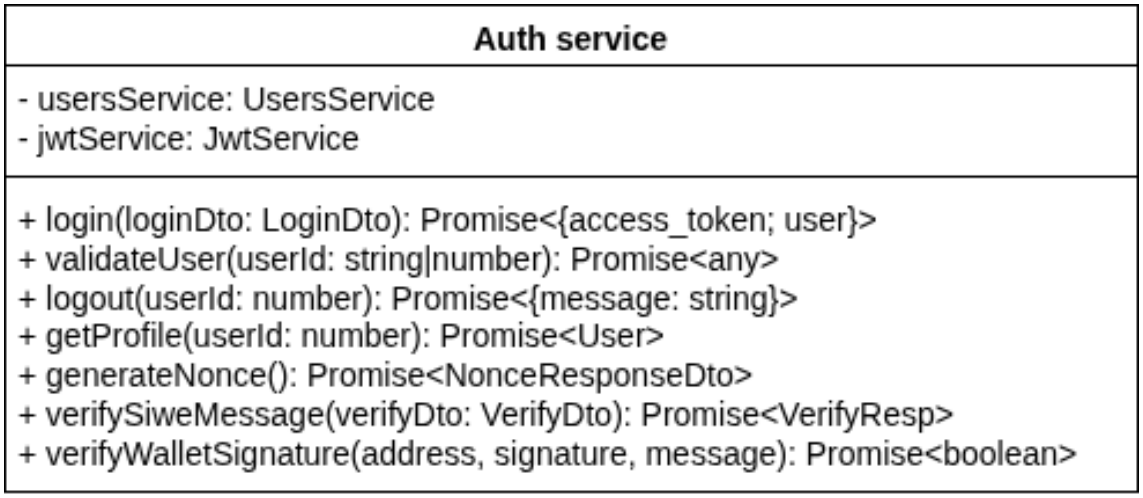
Hệ thống chuẩn hoá 2 loại nút chính:

Primary Button dùng cho hành động quan trọng như: Connect Wallet, *Create NFT*. **(2) Secondary Button** dùng cho hành động phụ/hỗ trợ như: *Cancel*.

Các nút được thiết kế thống nhất về kiểu dáng và trạng thái hiển thị: (1) Trạng thái bình thường: hiển thị rõ nội dung, dễ nhận biết. (2) Trạng thái hover/active (desktop): thay đổi sắc độ/viền để phản hồi thao tác. (3) Trạng thái disabled: giảm độ nổi bật để biểu thị không thể thao tác. (4) Trạng thái loading: hiển thị tiến trình khi hệ thống đang xử lý các thao tác như xác nhận giao dịch hoặc tải dữ liệu.

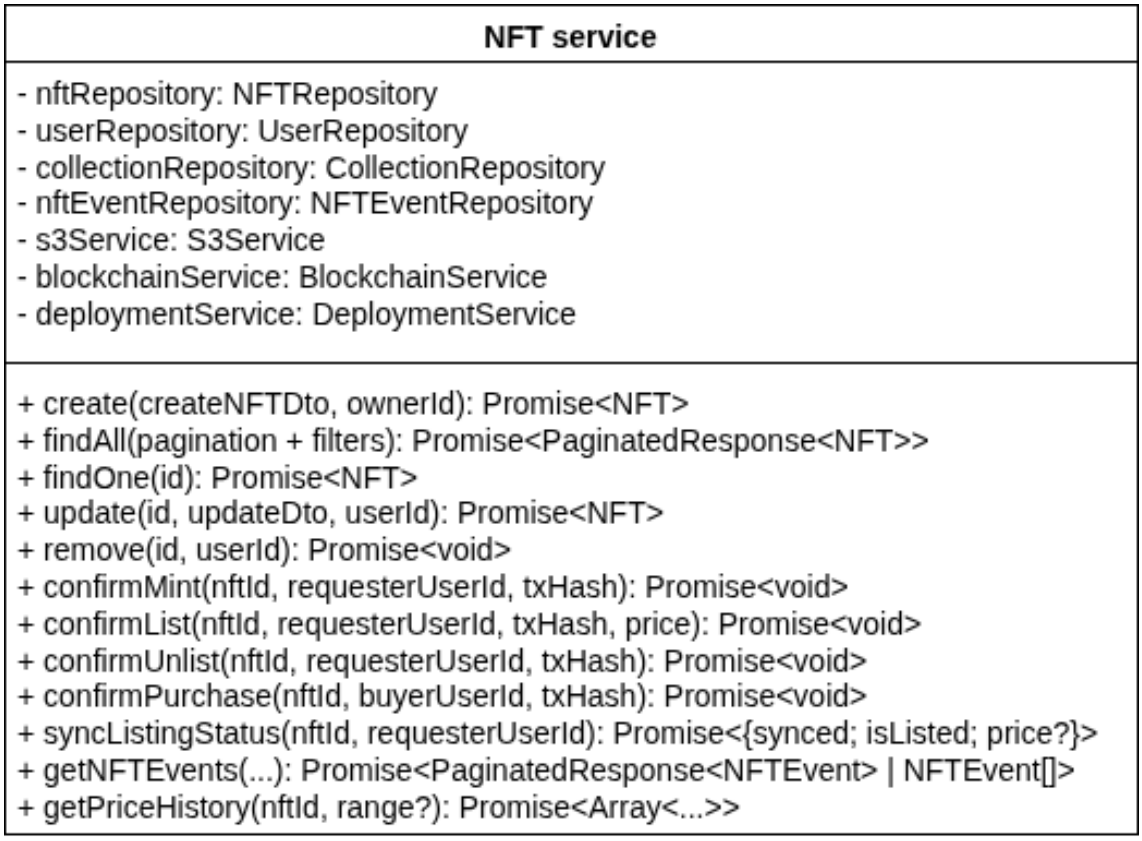
Minh họa thiết kế giao diện(đoạn này cần thêm ảnh vào, sau đọc lại nhớ bổ sung)

0.2.2 Thiết kế lớp



Hình 0.5: Biểu đồ Lớp Auth service

Lớp AuthService được sử dụng để thực hiện xác thực người dùng dựa trên địa chỉ ví và chữ ký (signature). Khi người dùng gửi yêu cầu đăng nhập, hệ thống chuẩn hoá địa chỉ ví để đảm bảo tính nhất quán dữ liệu. Nếu request có kèm chữ ký và thông điệp ký, backend sẽ tiến hành xác minh chữ ký nhằm đảm bảo người dùng thực sự sở hữu ví đó. Sau khi xác thực hợp lệ, hệ thống tìm hoặc tự động tạo mới tài khoản theo địa chỉ ví, sau đó phát hành JWT để người dùng sử dụng trong các request tiếp theo. Lớp cũng hỗ trợ xác thực theo chuẩn SIWE và cung cấp các chức năng phụ trợ như tạo nonce và truy vấn hồ sơ người dùng.



Hình 0.6: Biểu đồ Lớp NFT service

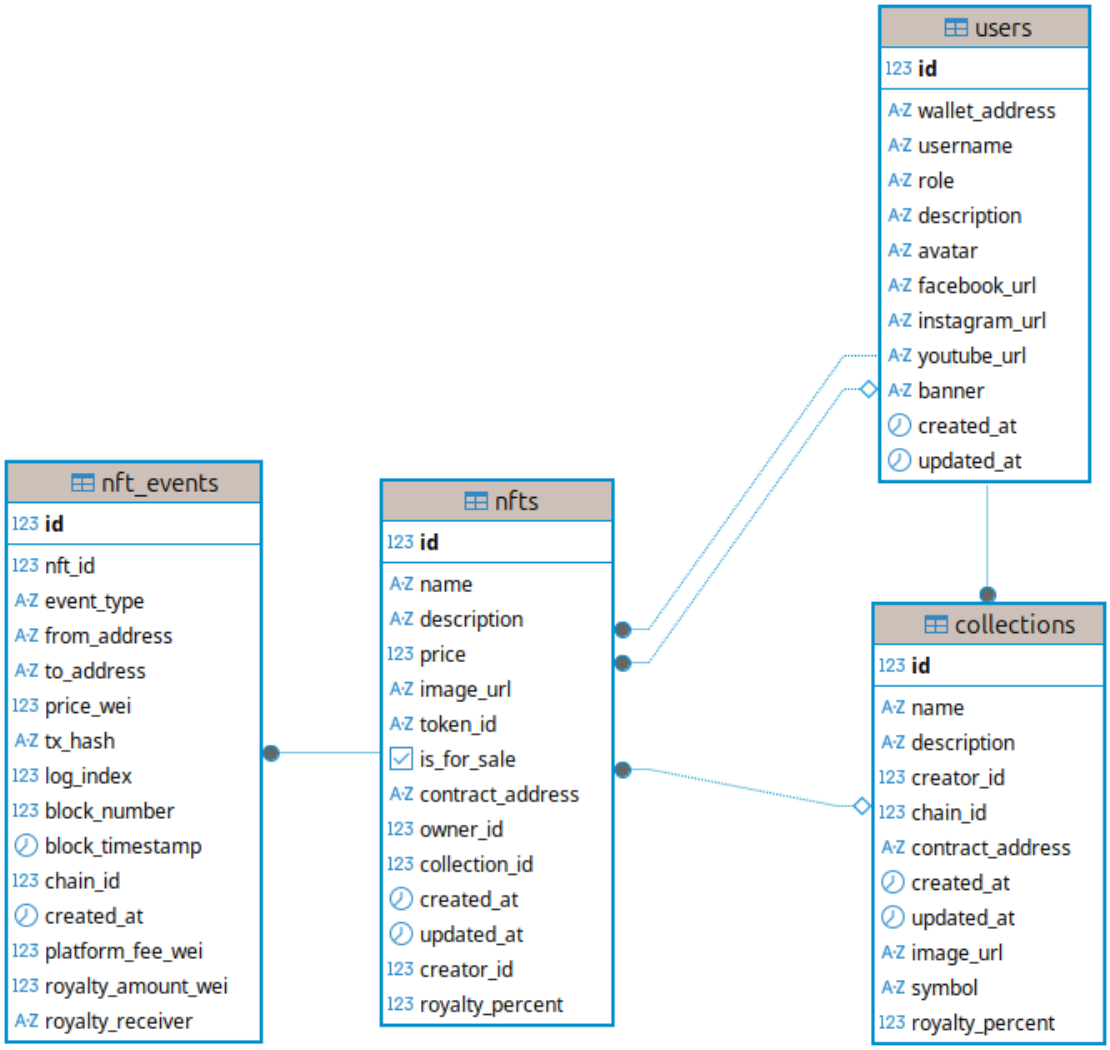
Lớp NFTService là lớp nghiệp vụ trung tâm của module NFT. Lớp này chịu trách nhiệm quản lý vòng đời của NFT trong hệ thống off-chain (tạo NFT, truy vấn, cập nhật, xoá), đồng thời xử lý các luồng nghiệp vụ gắn với blockchain như mint, list/unlist và purchase. Điểm quan trọng trong thiết kế là hệ thống không ghi nhận kết quả giao dịch chỉ dựa trên dữ liệu từ client, mà sử dụng txHash để truy vấn transaction và xác minh event phát sinh trên blockchain. Khi dữ liệu xác minh hợp lệ, backend mới cập nhật trạng thái NFT trong cơ sở dữ liệu và ghi nhận lịch sử giao dịch (event) để phục vụ các chức năng của history. Cách tiếp cận này giúp dữ liệu off-chain đồng bộ và có cơ sở tin cậy từ dữ liệu on-chain.

Biểu đồ trinh tự kết nối ví

Biểu đồ trinh mua NFT

0.2.3 Thiết kế cơ sở dữ liệu

Thiết kế sơ đồ thực thể liên kết:



Hình 0.7: Sơ đồ thực thể liên kết

Đặc tả bảng User:

Trường	**Kiểu dữ liệu**	**Mô tả**
id	BIGINT	Khóa chính (PK) định danh người dùng.
wallet_address	VARCHAR(42)	Địa chỉ ví (0x...), dùng để đăng nhập/định danh on-chain.
username	VARCHAR(100)	Tên hiển thị của người dùng trên hệ thống.
role	VARCHAR(30)	Vai trò người dùng (ví dụ: user/admin).
description	TEXT	Mô tả/bio của người dùng.
avatar	TEXT	URL ảnh đại diện.
facebook_url	TEXT	Liên kết Facebook (nếu có).
instagram_url	TEXT	Liên kết Instagram (nếu có).
youtube_url	TEXT	Liên kết Youtube (nếu có).
banner	TEXT	URL ảnh bìa (banner).

Trường	Kiểu dữ liệu	Mô tả
created_at	TIMESTAMP	Thời điểm tạo bản ghi.
updated_at	TIMESTAMP	Thời điểm cập nhật gần nhất.

Bảng 3: Chi tiết bảng người dùng

Đặc tả bảng Collection:

Trường	Kiểu dữ liệu	Mô tả
id	BIGINT	Khóa chính (PK) định danh collection.
name	VARCHAR(150)	Tên collection.
description	TEXT	Mô tả collection.
creator_id	BIGINT	Khóa ngoại (FK) → users.id, người tạo collection.
chain_id	INT	Mã mạng blockchain (ví dụ: 1, 56, 97...).
contract_address	VARCHAR(42)	Địa chỉ contract của collection trên blockchain.
created_at	TIMESTAMP	Thời điểm tạo bản ghi.
updated_at	TIMESTAMP	Thời điểm cập nhật gần nhất.
image_url	TEXT	URL ảnh đại diện collection.
symbol	VARCHAR(30)	Ký hiệu (symbol) của collection (ví dụ: BAYC).
royalty_percent	NUMERIC(5,2)	Phần trăm royalty áp dụng cho collection (ví dụ: 2.50).

Bảng 4: Chi tiết bảng Collection

Đặc tả bảng NFT:

Trường	Kiểu dữ liệu	Mô tả
id	BIGINT	Khóa chính (PK) định danh NFT trong DB.
name	VARCHAR(150)	Tên NFT.
description	TEXT	Mô tả NFT.
price	NUMERIC(38,0)	Giá hiện tại (có thể lưu theo wei hoặc theo đơn vị chuẩn tùy thiết kế).
image_url	TEXT	URL ảnh NFT (S3/IPFS/HTTP...).
token_id	BIGINT	TokenId on-chain của NFT.
is_for_sale	BOOLEAN	Trạng thái đang được niêm yết bán hay không.
contract_address	VARCHAR(42)	Địa chỉ contract chứa NFT.
owner_id	BIGINT	FK → users.id, owner hiện tại.
collection_id	BIGINT	FK → collections.id, NFT thuộc collection nào.

Trường	Kiểu dữ liệu	Mô tả
created_at	TIMESTAMP	Thời điểm tạo bản ghi.
updated_at	TIMESTAMP	Thời điểm cập nhật gần nhất.
creator_id	BIGINT	FK → users.id, người tạo/mint NFT.
royalty_percent	NUMERIC(5,2)	Royalty cho NFT.

Bảng 5: Chi tiết bảng NFT

Đặc tả bảng NFT event:

Trường	Kiểu dữ liệu	Mô tả
id	BIGINT	Khóa chính (PK) định danh sự kiện NFT.
nft_id	BIGINT	FK → nfts.id, NFT liên quan tới sự kiện.
event_type	VARCHAR(50)	Loại sự kiện (mint/list/buy/transfer/cancel...).
from_address	VARCHAR(42)	Địa chỉ ví nguồn (người gửi/seller).
to_address	VARCHAR(42)	Địa chỉ ví đích (người nhận/buyer).
price_wei	NUMERIC(38,0)	Giá giao dịch theo wei (nếu có).
tx_hash	VARCHAR(66)	Transaction hash trên blockchain.
log_index	INT	Log index trong transaction (phục vụ đối chiếu event log).
block_number	BIGINT	Số block chứa transaction.
block_timestamp	TIMESTAMP	Thời điểm block được ghi nhận.
chain_id	INT	Mã mạng blockchain phát sinh sự kiện.
created_at	TIMESTAMP	Thời điểm hệ thống lưu sự kiện vào DB.
platform_fee_wei	NUMERIC(38,0)	Phí nền tảng thu được (tính theo wei).
royalty_amount_wei	NUMERIC(38,0)	Số tiền royalty trả cho creator (theo wei).
royalty_receiver	VARCHAR(42)	Địa chỉ nhận royalty.

Bảng 6: Chi tiết bảng NFT event

0.3 Xây dựng ứng dụng

0.3.1 Thư viện và công cụ sử dụng

Các công cụ được sử dụng trong quá trình phát triển sản phẩm được liệt kê dưới bảng **??**:

STT	Mục đích	Tên công cụ	Phiên bản	URL
1	Ngôn ngữ + runtime backend	Node.js (LTS) + TypeScript	24.12.0; 5.9.3	https://nodejs.org/; https://www.typescriptlang.org/
2	Framework backend API	NestJS	11.1.10	https://nestjs.com/
3	Framework frontend web	Next.js	16.1.1	https://nextjs.org/
4	Hệ quản trị CSDL (SQL)	PostgreSQL	18.1	https://www.postgresql.org/
5	Cache / tăng tốc truy vấn	Redis (Open Source)	8.4.0	https://redis.io/
6	IDE phát triển	Visual Studio Code	1.107.1	https://code.visualstudio.com/
7	Triển khai/vận hành trên AWS	AWS CLI (v2)	2.32.24	https://aws.amazon.com/cli/

Bảng 7: Các công cụ chính sử dụng trong quá trình phát triển hệ thống

0.3.2 Kết quả đạt được(cần cập nhật lại bảng chỉ số thống kê)

Trong quá trình thực hiện đồ án, em đã xây dựng được một hệ thống NFT Marketplace có thể vận hành và trình diễn đầy đủ các luồng chức năng chính. Kết quả đạt được không chỉ dừng ở việc hoàn thiện mã nguồn, mà còn được đóng gói thành các sản phẩm cụ thể để thuận tiện cho việc triển khai, kiểm thử và demo. Các sản phẩm đóng gói bao gồm: ứng dụng giao diện người dùng (frontend) giúp người dùng thao tác tạo/bán/mua và khám phá NFT; dịch vụ backend API xử lý nghiệp vụ, quản lý dữ liệu và đồng bộ trạng thái sau giao dịch; smart contract triển khai các chức năng on-chain liên quan đến NFT; ngoài ra kèm theo các script cấu hình/migration phục vụ khởi tạo dữ liệu. Việc đóng gói theo từng thành phần giúp hệ thống có cấu trúc rõ ràng, dễ quản lý và có thể mở rộng trong tương lai.

Chỉ số thống kê	Frontend	Backend	Smart contract	Tổng
Số dòng code (LOC)	...	...	...	...
Số thư mục/module chính	...	...	...	...
Số file mã nguồn	...	...	...	...
Số lớp (class) / thành phần chính	...	...	...	...
Dung lượng toàn bộ mã nguồn (MB)	...	...	...	...
Dung lượng sản phẩm sau build (MB)	...	...	...	...

Bảng 8: Thống kê quy mô mã nguồn và sản phẩm đóng gói

0.3.3 Minh họa các chức năng chính

Sinh viên lựa chọn và đưa ra màn hình cho các chức năng chính, quan trọng, và thú vị nhất. Mỗi giao diện cần phải có lời giải thích ngắn gọn. Khi giải thích, sinh viên có thể kết hợp với các chú thích ở trong hình ảnh giao diện.

0.4 Kiểm thử

0.4.1 Mục tiêu và phạm vi kiểm thử

Phần này tập trung thiết kế và thực hiện kiểm thử cho các chức năng quan trọng nhất của hệ thống NFT Marketplace: đăng nhập kết nối ví; tạo NFT; niêm yết và mua/bán NFT.

0.4.2 Kỹ thuật kiểm thử đã sử dụng

Trong quá trình thiết kế test case, em sử dụng kết hợp các kỹ thuật sau:

- **Kiểm thử hộp đen (Black-box testing):** thiết kế test case dựa trên đầu vào/đầu ra và hành vi mong đợi theo yêu cầu.
- **Phân lớp tương đương (Equivalence Partitioning):** chia dữ liệu đầu vào thành nhóm hợp lệ/không hợp lệ để giảm số lượng test case nhưng vẫn đảm bảo bao phủ.
- **Giá trị biên (Boundary Value Analysis):** kiểm thử tại các ngưỡng quan trọng (ví dụ giá = 0, giá rất nhỏ, độ dài chuỗi tối đa,. . .).
- **Kiểm thử theo luồng (Scenario/Flow testing):** kiểm thử theo chuỗi hành động thực tế (login $\rightarrow$ tạo NFT $\rightarrow$ list $\rightarrow$ buy $\rightarrow$ cập nhật).
- **Kiểm thử API/kiểm thử tích hợp (API & Integration testing):** gọi API và đối chiếu dữ liệu phát sinh trong cơ sở dữ liệu.

0.4.3 Thiết kế các trường hợp kiểm thử

a, Chức năng 1: Đăng nhập bằng ví (SIWE)

Người dùng lấy nonce, ký thông điệp SIWE bằng ví, backend xác thực chữ ký và cấp access token.

Mã TC	Mục tiêu	Dữ liệu vào	Các bước thực hiện	Kết quả mong đợi
TC-LOGIN-01	Đăng nhập thành công	wallet hợp lệ, message hợp lệ, signature hợp lệ	(1) Lấy nonce (2) Ký SIWE (3) Gửi verify/login	Trả token; user được tạo mới(nếu chưa tồn tại trong hệ thống).
TC-LOGIN-02	Sai chữ ký	signature không khớp message	Gửi verify/login với signature sai	Trả lỗi và không cấp token.
TC-LOGIN-03	Nonce đã dùng/hết hạn	nonce cũ hoặc bị dùng lại	Dùng lại nonce và gửi verify/login	Bị từ chối và yêu cầu lấy nonce mới.

Mã TC	Mục tiêu	Dữ liệu vào	Các bước thực hiện	Kết quả mong đợi
TC-LOGIN-04	Sai chain/domain	chainId hoặc domain không đúng cấu hình	Gửi verify/login với chain/domain sai	Bị từ chối xác thực và trả lỗi.
TC-LOGIN-05	Wallet sai định dạng	wallet address sai chuẩn	Gửi verify/login với địa chỉ ví sai	Validate thất bại và trả lỗi.

Bảng 9: Thiết kế test case cho chức năng đăng nhập bằng ví (SIWE)

b, Chức năng 2: Tạo NFT

Người dùng nhập thông tin NFT (tên, mô tả, ảnh/metadata), backend lưu dữ liệu NFT và liên kết vào collection.

Mã TC	Mục tiêu	Dữ liệu vào	Các bước thực hiện	Kết quả mong đợi
TC-NFT-01	Tạo NFT hợp lệ	dữ liệu NFT hợp lệ	Gọi POST /nfts	Tạo thành công, trả về dữ liệu đúng, DB có bản ghi mới.
TC-NFT-02	Thiếu trường bắt buộc	name rỗng hoặc thiếu	Gọi POST /nfts	Trả lỗi, không tạo bản ghi.
TC-NFT-03	Collection không tồn tại	collectionId sai	Gọi POST /nfts	Trả lỗi, không tạo NFT.
TC-NFT-04	Ảnh/URL sai định dạng	imageUrl không hợp lệ	Gọi POST /nfts	Trả lỗi, thông báo rõ trường sai.
TC-NFT-05	Giá trị biên (tên)	name quá dài hoặc vượt giới hạn	Gọi POST /nfts	Trả lỗi validation và yêu cầu user nhập lại.

Bảng 10: Thiết kế test case cho chức năng tạo NFT

c, Chức năng 3: Niêm yết bán và cập nhật trạng thái sau giao dịch

User niêm yết NFT với giá bán; sau giao dịch mua trên ví, frontend gọi API để backend cập nhật dữ liệu dựa theo transaction event.

Mã TC	Mục tiêu	Dữ liệu vào	Các bước thực hiện	Kết quả mong đợi
TC-LIST-01	List NFT thành công	nft thuộc owner hiện tại, price $>$ 0	Gọi POST /listings	Chuyển trạng thái NFT listing sang active và cập nhật giá bán.
TC-LIST-02	Không phải chủ sở hữu	nft không thuộc user hiện tại	Gọi POST /listings	Trả lỗi và không tạo listing.
TC-LIST-03	Giá bán không hợp lệ (biên)	price = 0 hoặc price $<$ 0	Gọi POST /listings	Trả lỗi và không tạo listing.

Mã TC	Mục tiêu	Dữ liệu vào	Các bước thực hiện	Kết quả mong đợi
TC-BUY-01	Xác nhận mua thành công	txHash hợp lệ, buyer/seller hợp lệ	Gọi POST /transactions/confirm	Cập nhật thành công owner sang buyer, tạo bản ghi history/event tương ứng.
TC-BUY-02	Trùng txHash (idempotent)	txHash đã tồn tại trong DB	Gọi lại confirm với cùng txHash	Không tạo trùng; trả kết quả an toàn khi retry (idempotent).
TC-CANCEL-01	Hủy listing	listing đang active	Gọi POST /listings/cancel	Chuyển trạng thái listing sang unactive thành công.

Bảng 11: Thiết kế test case cho chức năng niêm yết và cập nhật trạng thái sau giao dịch

0.5 Triển khai(khi nào deploy có cấu hình server thì cập nhật vào đây)