



TẬP ĐOÀN CÔNG NGHIỆP – VIỄN THÔNG QUÂN ĐỘI

BÁO CÁO MINI-PROJECT

Xây dựng trang web bán hàng tích hợp Viettel Money

Trần Quang Hưng

hungtran30112004@gmail.com

Chương trình Viettel Digital Talent 2025

Lĩnh vực: Software Engineering

Mentor: Nguyễn Đức Thịnh

Đơn vị: Tổng Công ty Dịch vụ số Viettel (VDS)

HÀ NỘI, 06/2025

Mục lục

I. Giới thiệu vấn đề	2
A. Yêu cầu đề bài	2
II. Kiến trúc hệ thống	3
A. Sơ đồ kiến trúc hệ thống	3
B. Công nghệ sử dụng	3
III.Cơ sở dữ liệu	4
A. Các bảng trong cơ sở dữ liệu	4
IV.Backend	6
A. Tổng quan	6
B. Gói category	7
C. Gói product	7
D. Gói order	8
E. Gói payment	9
F. Bảo mật và xác thực người dùng	10
V. Frontend	11
VI.Kết luận và hướng phát triển	14

I. Giới thiệu vắn đề

A. Yêu cầu đề bài

Xây dựng website bán hàng đơn giản kết nối thanh toán qua cổng thanh toán của Viettel Money, sử dụng frontend Node.js (hoặc các biến thể) cùng backend Spring Boot

- **Nghiên cứu và phân tích yêu cầu:**

- Xác định sản phẩm bán và đối tượng khách hàng (tự chọn)
- Tham khảo các giao diện web bán hàng trên thị trường
- Xác định các chức năng cơ bản cần có

- **Thiết kế giao diện người dùng (UI/UX):**

- Lên kế hoạch và thiết kế bố cục trang (wireframe)
- Đảm bảo tính tương thích với các thiết bị và trình duyệt khác nhau (responsive)

- **Phát triển giao diện:**

- Sử dụng các công nghệ web phổ biến như HTML, CSS, JavaScript để xây dựng giao diện
- Tích hợp các thành phần giao diện như biểu mẫu đăng ký, các yếu tố truyền thông (hình ảnh, video, biểu đồ, v.v.)

- **Phát triển hệ thống xử lý phía sau:**

- Xây dựng chức năng cho phép người bán đăng nhập, quản trị người dùng đơn giản (có thể dùng công cụ mã nguồn mở như Keycloak)
- Thiết kế cơ sở dữ liệu để lưu trữ thông tin hàng hóa, đơn hàng
- Tích hợp đầy đủ luồng thanh toán theo tài liệu đặc tả kết nối đã có

- **Kiểm thử và triển khai:**

- Kiểm thử chức năng và hiệu suất của trang web
- Đảm bảo tính bảo mật

II. Kiến trúc hệ thống

A. Sơ đồ kiến trúc hệ thống

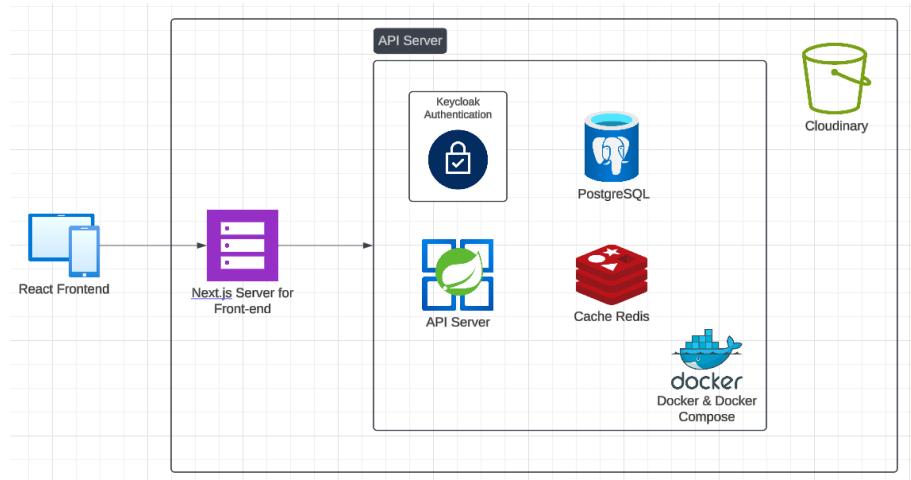


Figure 1: Thiết kế kiến trúc

B. Công nghệ sử dụng

- Front-end: React [1] với framework Next.js [2]
- Spring Boot: API server của ứng dụng
- Keycloak [3]: Xác thực và quản lý người dùng
- PostgreSQL [4]: Cơ sở dữ liệu
- Redis [5]: Caching
- Docker và Docker Compose [6]: đóng gói và điều phối các container của backend
- Cloudinary [7]: Lưu trữ đa phương tiện

III. Cơ sở dữ liệu

A. Các bảng trong cơ sở dữ liệu

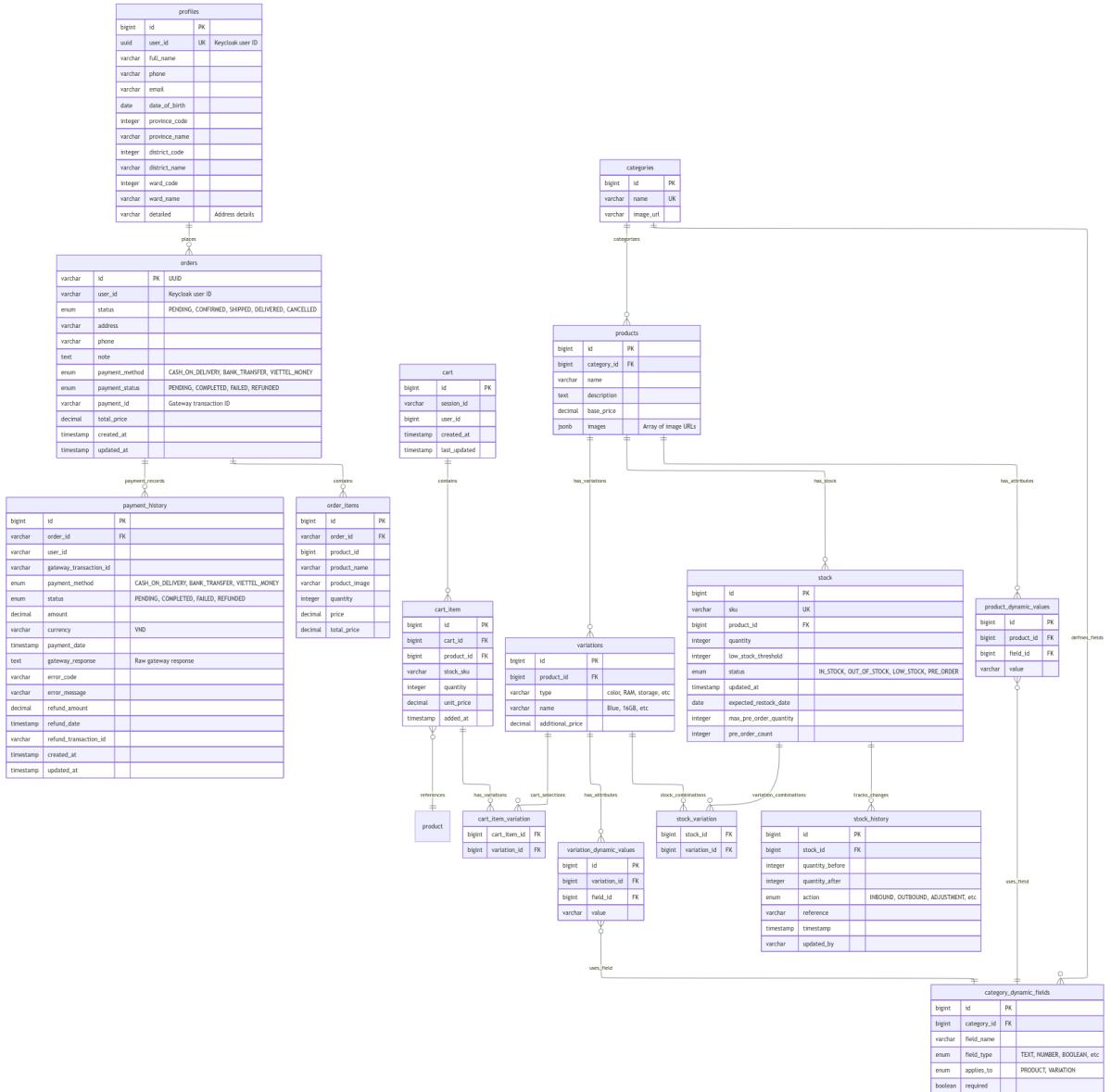


Figure 2: Thiết kế bảng

1. profiles

- Lưu thông tin cá nhân người dùng, gắn với tài khoản Keycloak qua `user_id`.
- Bao gồm họ tên, email, số điện thoại, ngày sinh và địa chỉ đầy đủ (tỉnh, huyện, xã, chi tiết).

2. cart và cart_item

- cart**: Đại diện cho một phiên giỏ hàng (session hoặc người dùng), lưu thời điểm tạo và cập nhật.

- **cart_item**: Lưu các sản phẩm trong giỏ hàng, bao gồm số lượng, đơn giá và SKU cụ thể.
- **cart_item_variation**: Liên kết các biến thể sản phẩm được chọn cho từng dòng trong giỏ hàng.

3. categories và category_dynamic_fields

- **categories**: Quản lý danh mục sản phẩm, gồm tên và ảnh đại diện.
- **category_dynamic_fields**: Định nghĩa các trường thuộc tính động tùy theo từng danh mục, áp dụng cho sản phẩm hoặc biến thể (ví dụ: màu sắc, RAM).

4. products, variations, và các bảng động

- **products**: Sản phẩm cơ bản, liên kết với danh mục, có mô tả, giá gốc và danh sách ảnh.
- **product_dynamic_values**: Lưu giá trị cụ thể của các trường động cho sản phẩm.
- **variations**: Biến thể sản phẩm (ví dụ: màu sắc, dung lượng), có thể thêm giá phụ.
- **variation_dynamic_values**: Giá trị của các trường động áp dụng cho biến thể.

5. stock, stock_variation, và stock_history

- **stock**: Theo dõi SKU tồn kho của sản phẩm hoặc tổ hợp biến thể, kèm trạng thái và nguồn cung cấp.
- **stock_variation**: Xác định tổ hợp biến thể tạo nên một SKU.
- **stock_history**: Ghi lại lịch sử nhập, xuất, điều chỉnh tồn kho, ai thực hiện và thời điểm.

6. orders và order_items

- **orders**: Đại diện cho đơn hàng, gồm thông tin giao hàng, trạng thái, phương thức thanh toán, tổng giá trị.
- **order_items**: Danh sách sản phẩm trong đơn hàng, lưu trữ thông tin tinh tại thời điểm đặt hàng (tên, ảnh, giá, số lượng).

7. payment_history

- Ghi nhận các giao dịch thanh toán qua cổng (COD, chuyển khoản, ví điện tử).
- Lưu trạng thái thanh toán, mã giao dịch, phản hồi từ cổng và thông tin hoàn tiền (nếu có).

IV. Backend

A. Tổng quan

Sơ đồ gói tổng quan của backend:

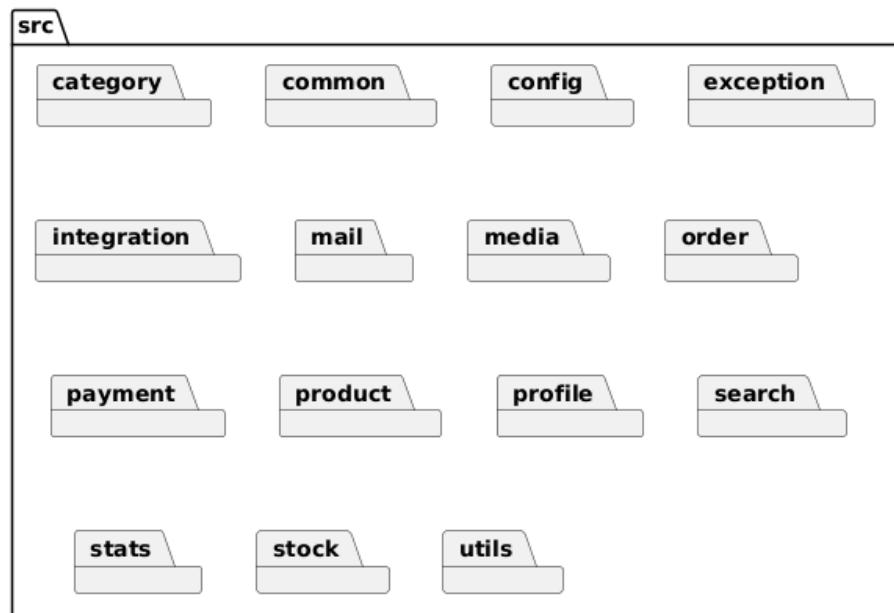


Figure 3: Sơ đồ gói tổng quan

Các gói được chia theo các module chức năng, bao gồm

- Các module liên quan tới nghiệp vụ như sản phẩm, danh mục, nhà kho... Tùy nhiệm vụ mà mỗi gói module lại được chia nhỏ thành các gói con như **entities** (các thực thể - dùng để tạo bảng), **dtos** (Data Transfer Object)
- Gói config: Chứa các cấu hình dành cho bảo mật hay các dịch vụ như Redis hay Cloudinary
- Gói exception: Chứa cách xử lý các ngoại lệ và mã lỗi khi có lỗi
- Gói common: Chứa một số lớp dùng chung.

B. Gói category

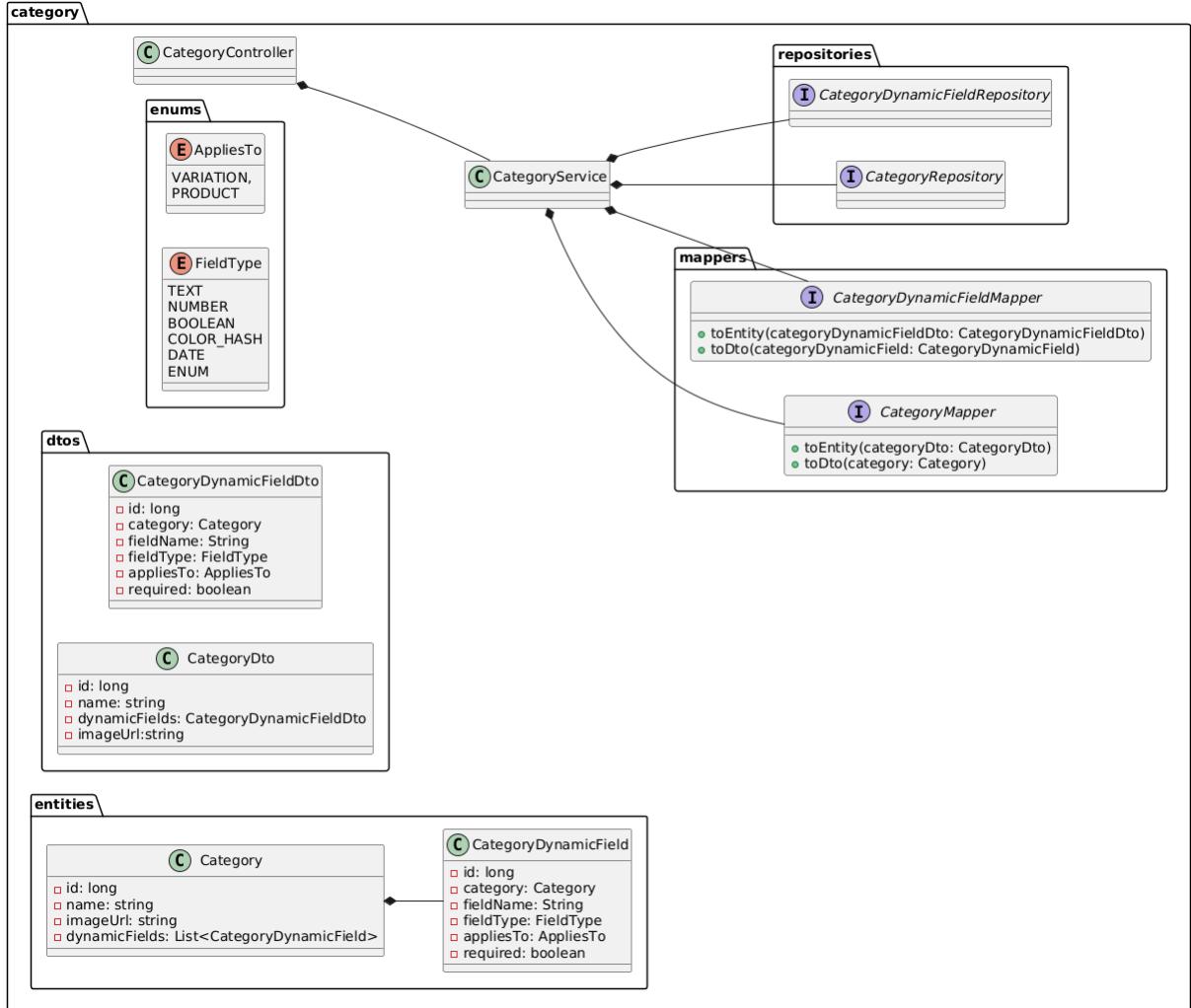


Figure 4: Sơ đồ gói Category

Gói **category** bao gồm các chức năng của danh mục sản phẩm, được chia thành từng lớp như **entity**, **dto**, **mapper**, **repository**, **service** và **controller**. Danh mục sản phẩm này cho phép định nghĩa các thuộc tính của mỗi danh mục sản phẩm. 4

C. Gói product

Gói **product** định nghĩa dữ liệu cần lưu trong database cũng như các dịch vụ liên quan. Mỗi sản phẩm nằm trong một danh mục, với các trường dữ liệu được định nghĩa trong khuôn khổ của Category đó. ??

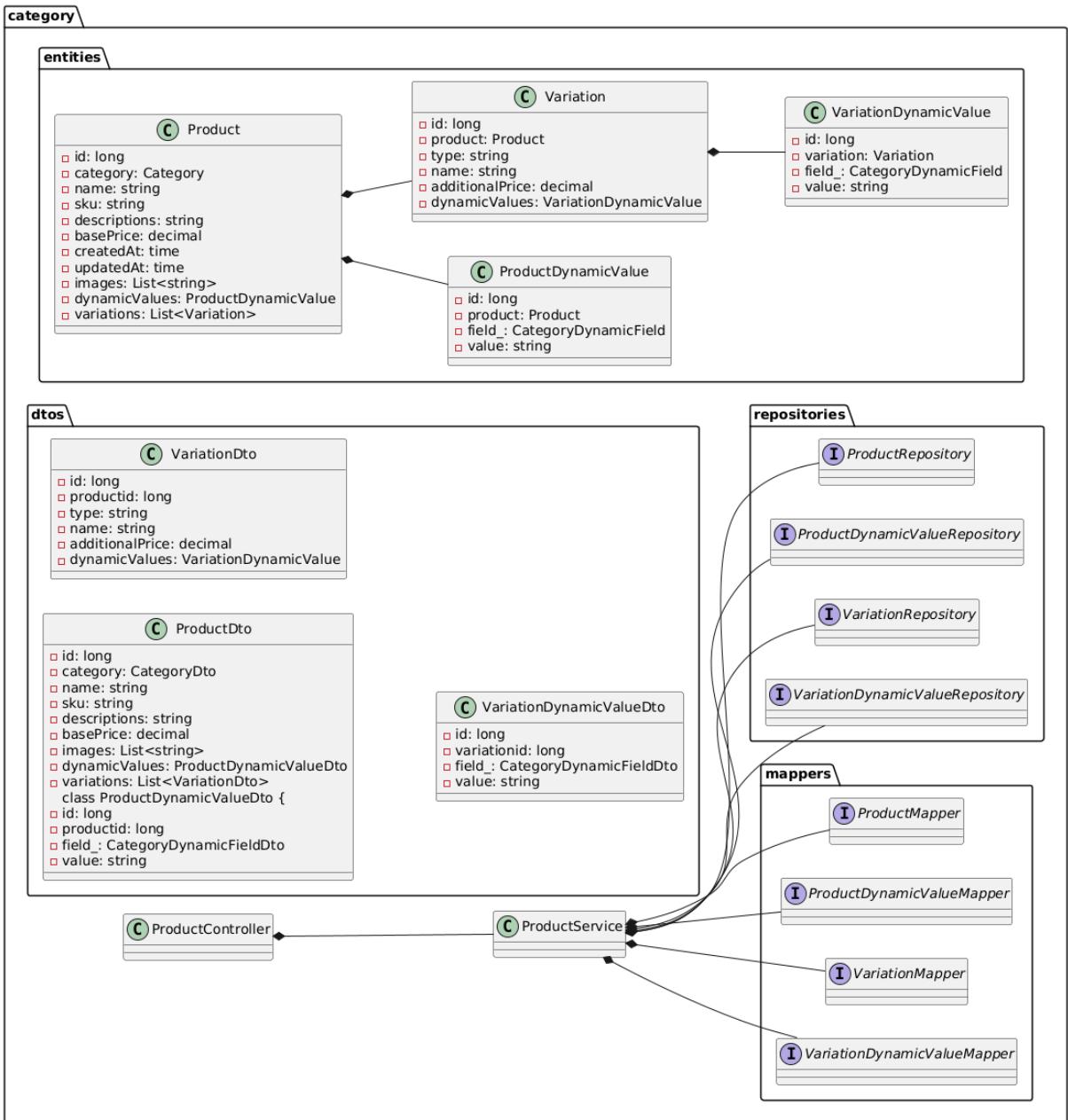


Figure 5: Sơ đồ gói product

D. Gói order

Gói order cho phép quản lý đơn hàng và trạng thái thanh toán của đơn hàng.

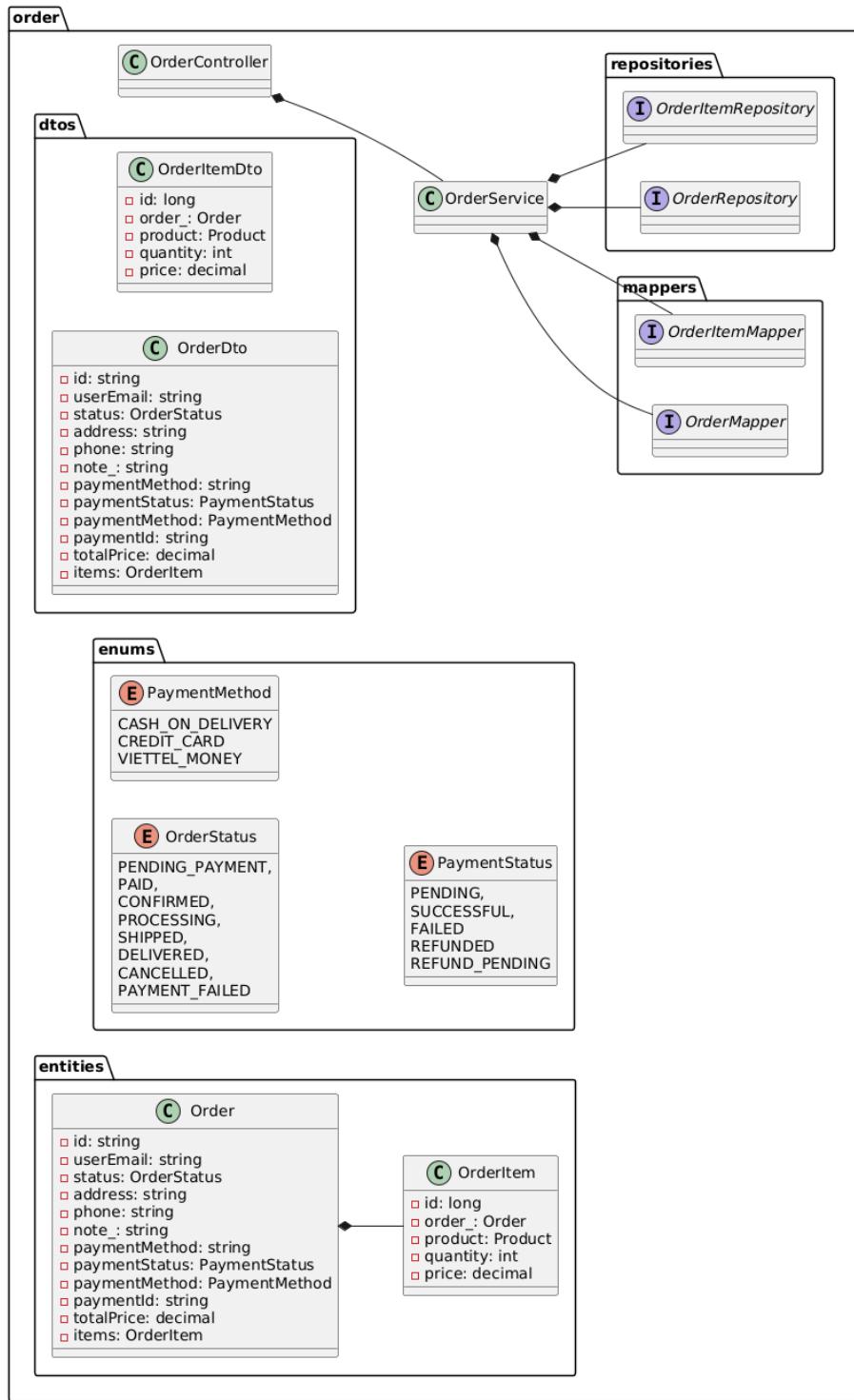


Figure 6: Sơ đồ gói Order

E. Gói payment

Gói payment là nơi tích hợp thanh toán với Viettel Money, cùng với API cung cấp để Viettel gọi tới cũng như phương thức bảo mật giao dịch hay lịch sử giao dịch.

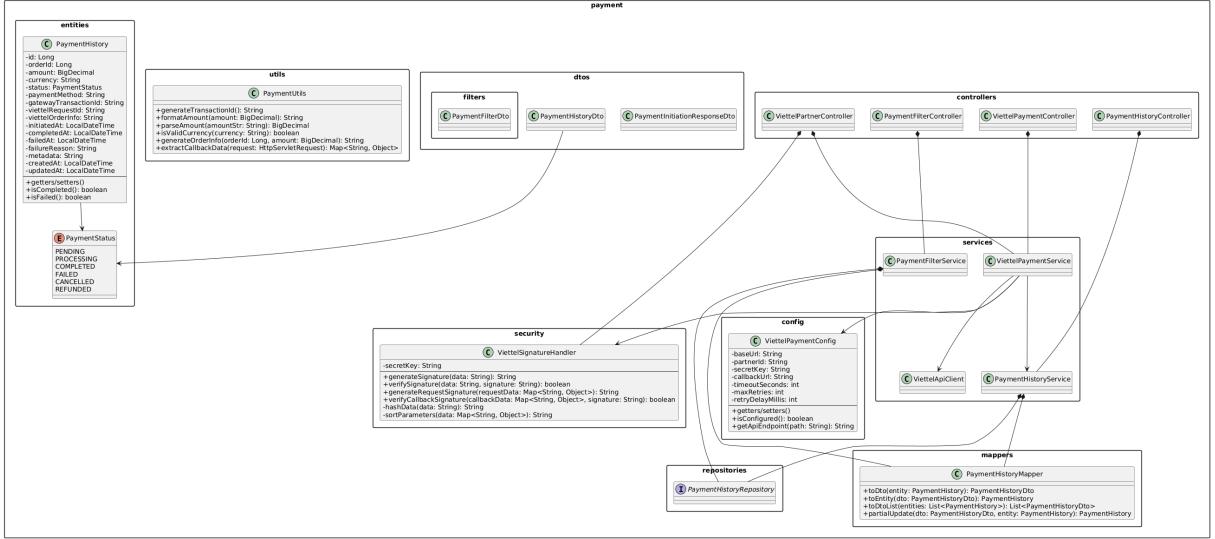


Figure 7: Gói payment

F. Bảo mật và xác thực người dùng

Về xác thực và quản lý người dùng, dự án sử dụng Keycloak để thực hiện. Keycloak cung cấp giao diện đăng nhập/đăng ký hoàn chỉnh, có thể sử dụng để đăng nhập bằng email/password hoặc thậm chí sử dụng OAuth thông qua các nhà cung cấp như Google hay Microsoft để đăng nhập website.

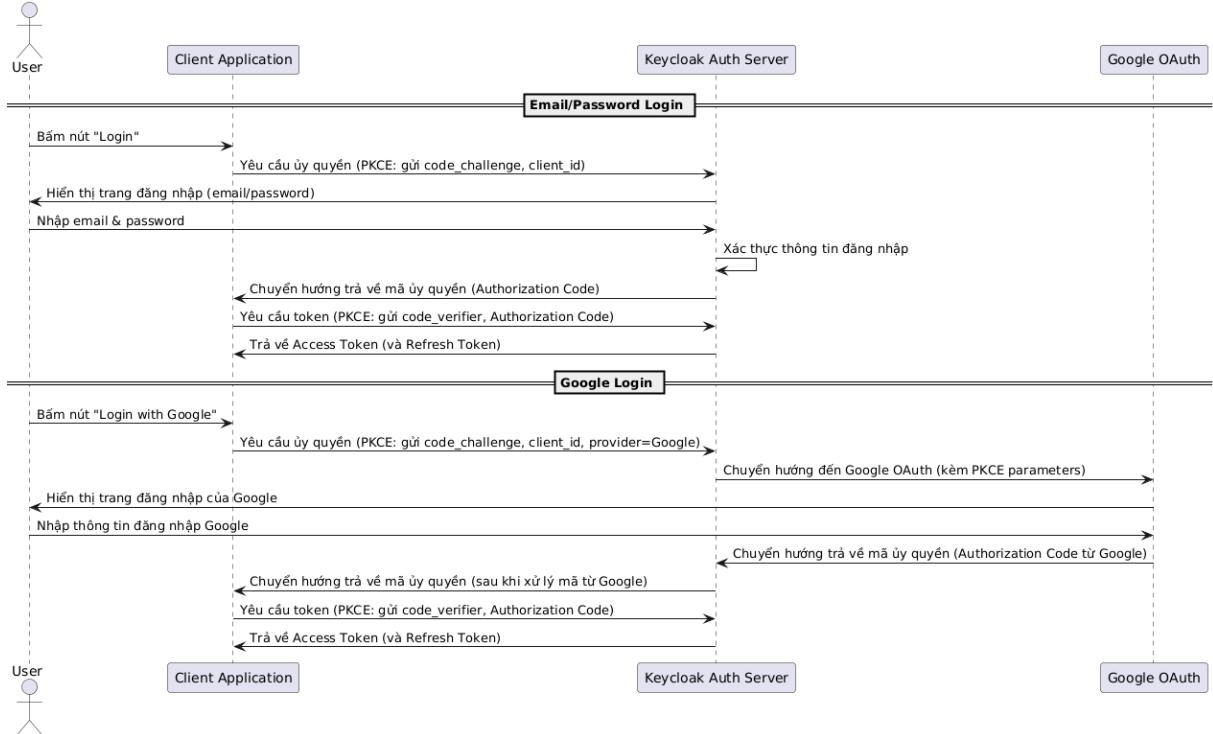


Figure 8: Luồng đăng nhập website với Keycloak

Project này sử dụng phương án xác thực với luồng PKCE với Keycloak và Next-auth của phía server Next.js. Ưu điểm của phương pháp đăng nhập này là có thể mở rộng thành SSO (Single Sign-on), dễ dàng triển khai khi mở rộng thành phần backend và frontend.

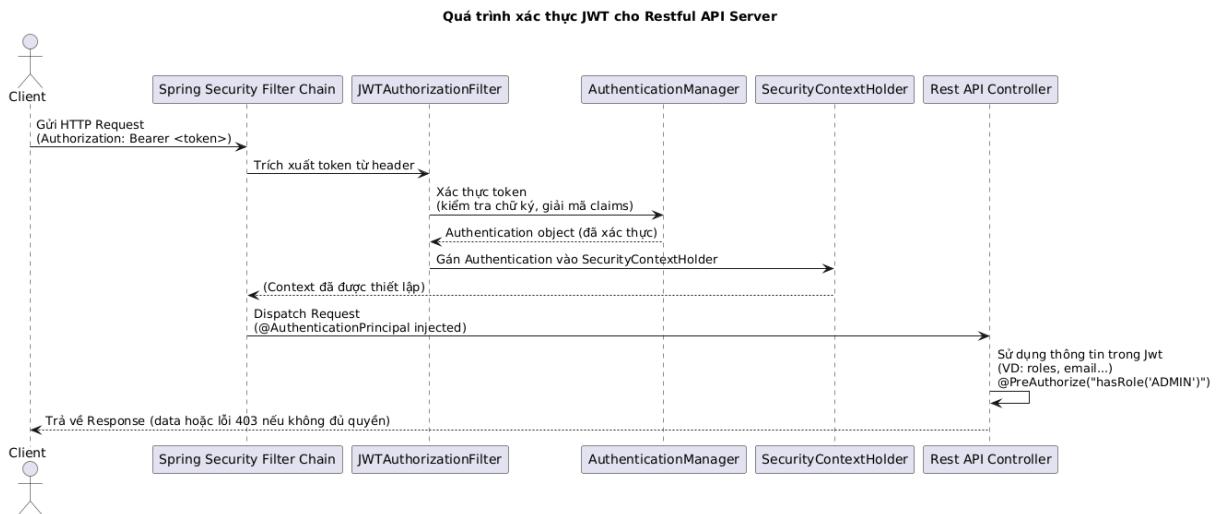


Figure 9: Enter Caption

Về API server, website sử dụng chuẩn RESTful API có bảo mật với JWT thông qua Bearer token. Server sẽ liên hệ với Keycloak để xác thực mã JWT trước khi thực hiện tác vụ đối với các endpoint cần được bảo mật.

Về tích hợp cổng thanh toán, website sử dụng chữ ký số ECDSA với SHA256 để ký request và xác minh các response. Các lớp như ViettelApiClient và ViettelPaymentService thực hiện quy trình ký số, kiểm tra tính toàn vẹn của dữ liệu gửi nhận và xử lý lỗi theo nguyên tắc fail-fast. Tuy nhiên, do thời gian có hạn cũng như các lỗi trong quản lý thời gian, tính năng này chưa được tích hợp đầy đủ vào website.

V. Frontend

Phần giao diện được lấy cảm hứng từ triết lý thiết kế của Material Design 3, với bộ màu sắc tươi mới và trẻ trung.

Người dùng có thể lướt xem các mặt hàng, danh mục sản phẩm thông qua giao diện người dùng. Tuy nhiên, để có thể mua hàng, người dùng cần phải đăng nhập vào website. Website cho phép người dùng đăng nhập bằng email/mật khẩu hoặc tài khoản Google.

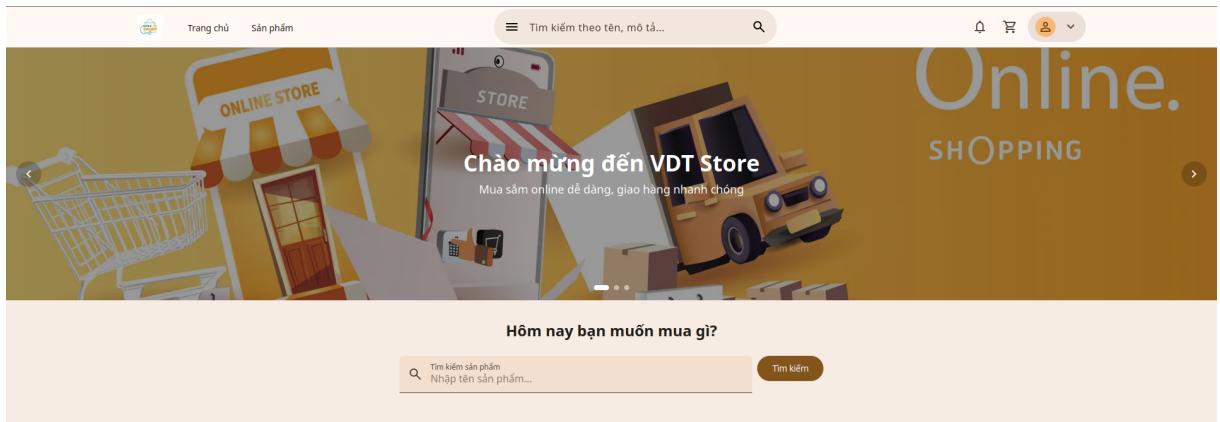


Figure 10: Trang chủ của ứng dụng

Dể có thể mua hàng, người dùng cần đăng nhập vào website. Keycloak cung cấp một giao diện đầy đủ cho việc đăng nhập / đăng ký tài khoản. Ở đây, website sẽ sử dụng 2 phương thức đăng nhập chính là đăng nhập với username/password và đăng nhập với tài khoản Google.

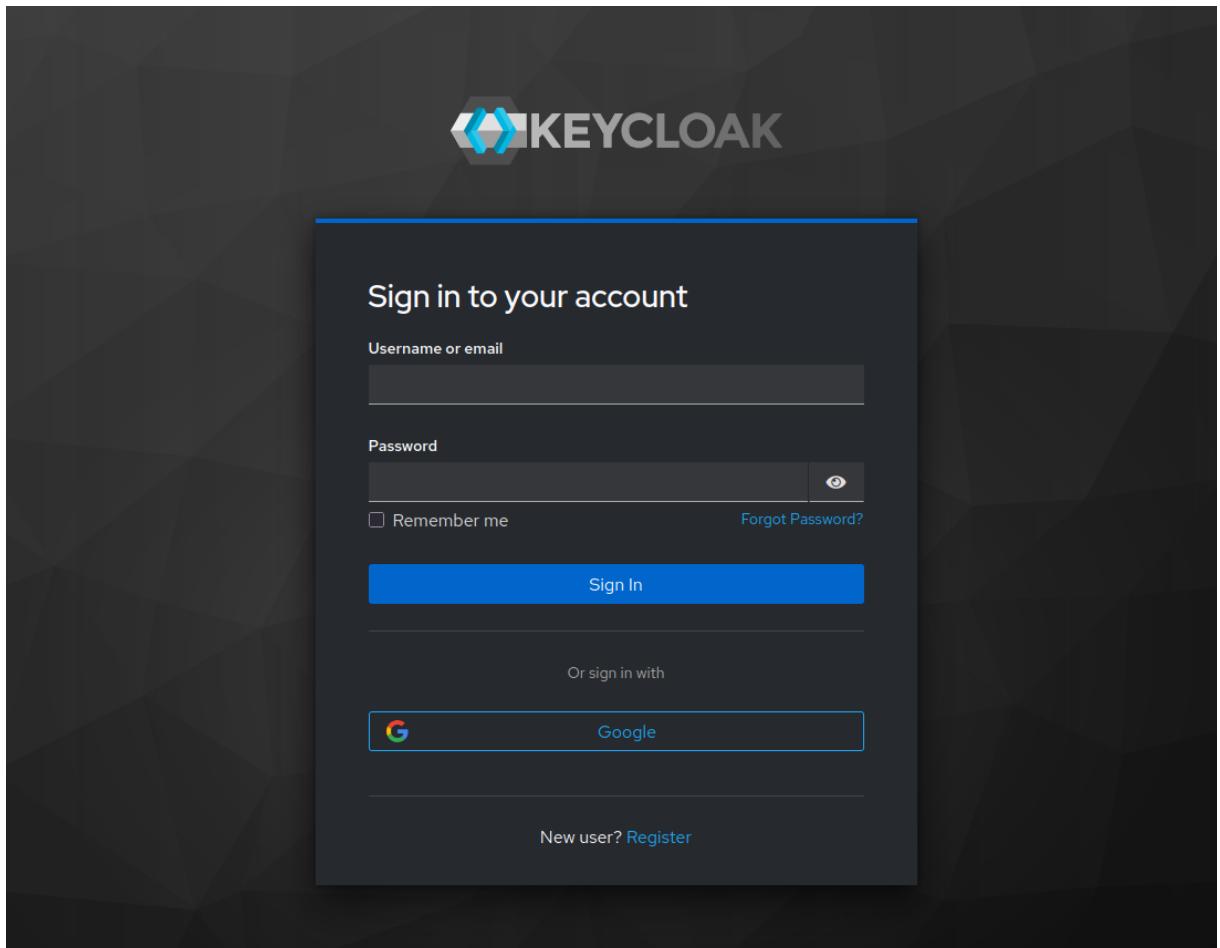


Figure 11: Đăng nhập/đăng ký

Từ trang chủ, người dùng có thể truy cập và các danh mục cũng như thông tin chi tiết sản phẩm. Nếu muốn mua, người dùng có thể ấn thêm sản phẩm vào giỏ

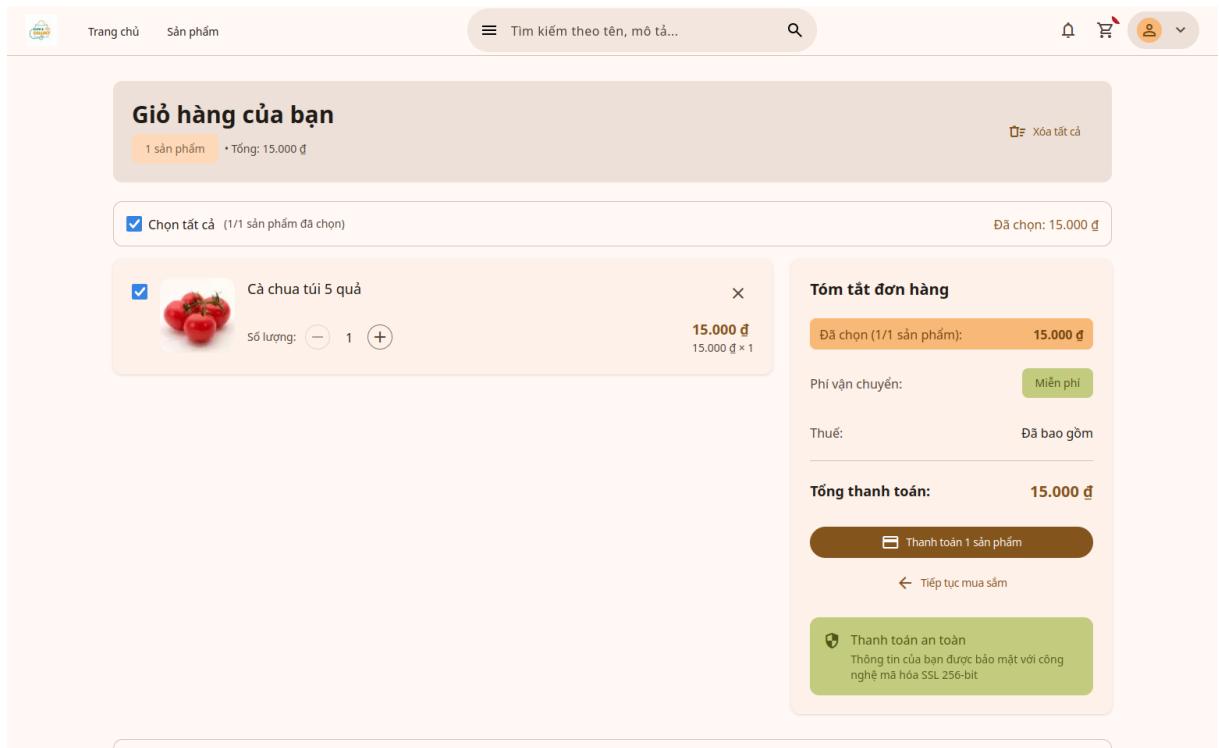


Figure 12: Giỏ hàng

Từ giỏ hàng, người dùng có thể chọn thanh toán, nhập thông tin giao hàng và thanh toán.

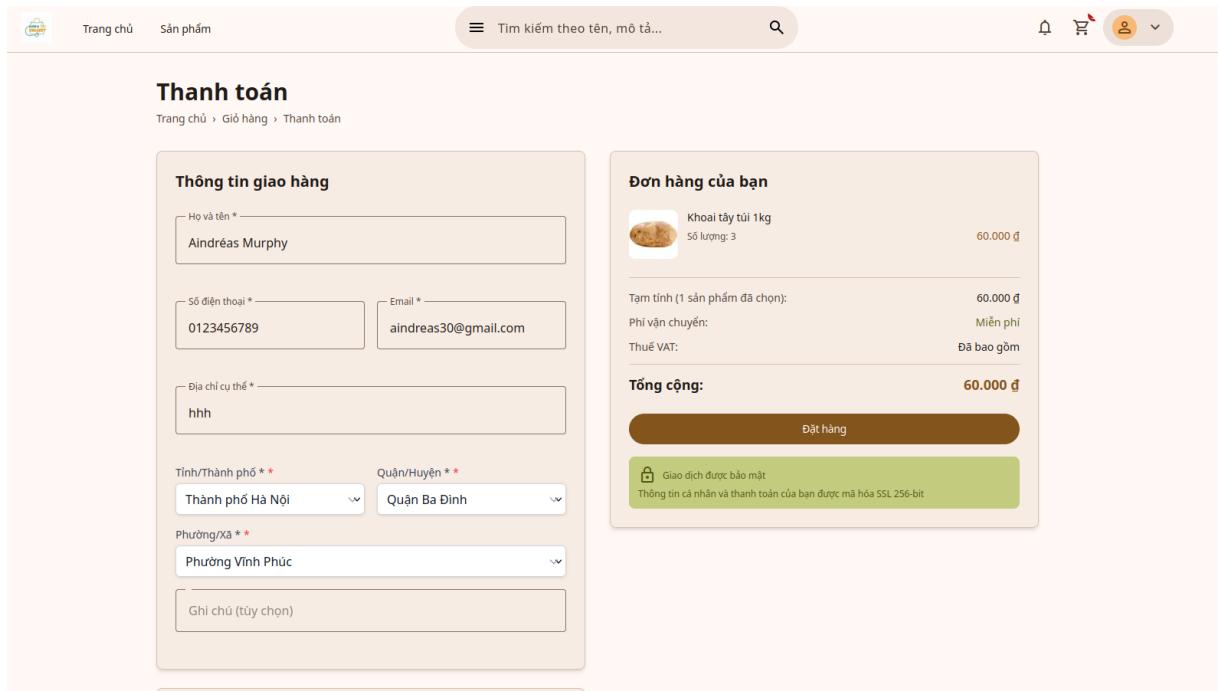


Figure 13: Trang thanh toán

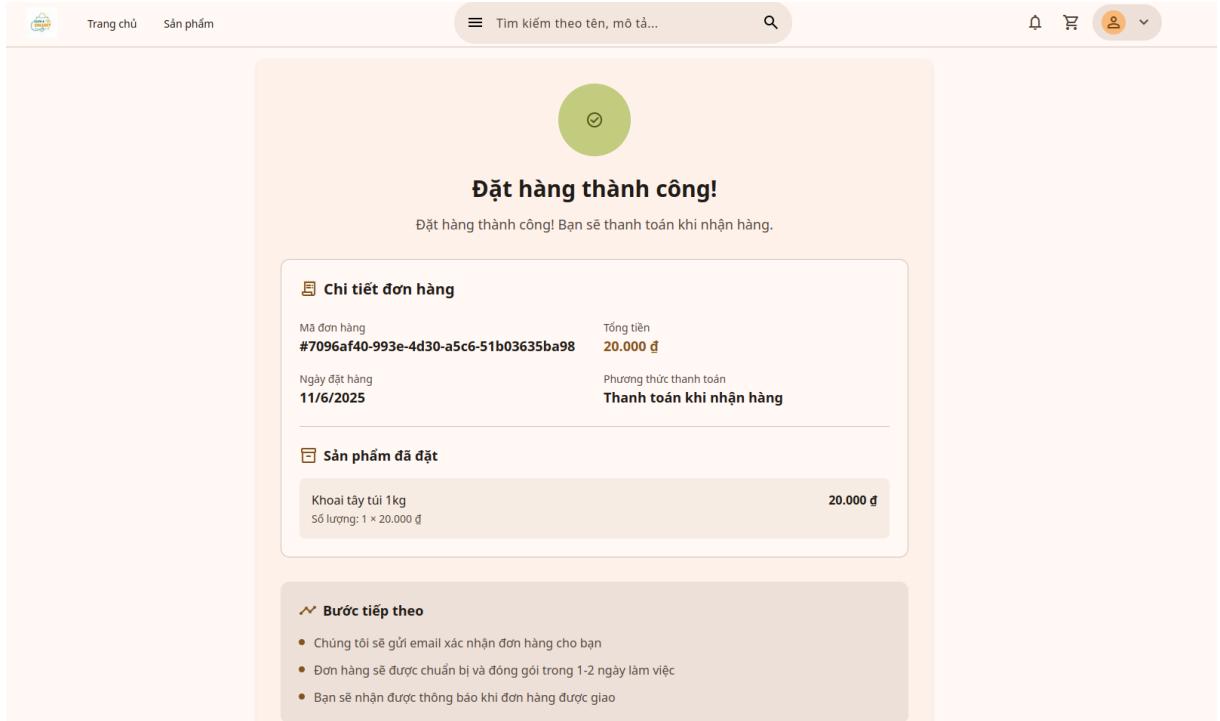


Figure 14: Đặt hàng thành công

Dối với chủ cửa hàng: Chủ cửa hàng có thẻ truy cập Keycloak để quản lý người dùng của mình. Ngoài ra, website cũng cung cấp giao diện cơ bản để chủ cửa hàng có thể quản lý danh mục, sản phẩm hay kho hàng.

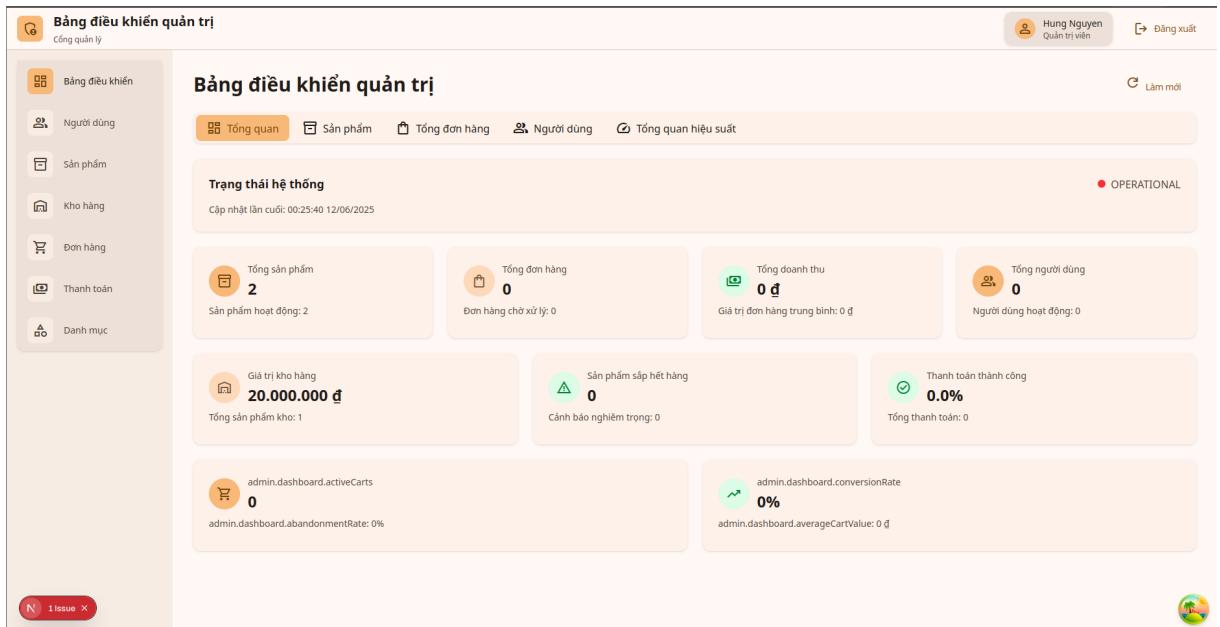


Figure 15: Trang quan trị

VI. Kết luận và hướng phát triển

Sau một tháng phát triển, sản phẩm đã tương đối hoàn thiện chức năng cơ bản của một ứng dụng bán hàng (tuy chưa tích hợp đầy đủ thanh toán với Viettel Money như yêu cầu).

Tuy nhiên, ứng dụng vẫn còn một số hạn chế cần cải thiện, như thiết kế cơ sở dữ liệu chưa tối ưu, giao diện nhiều chỗ chưa được trau chuốt kỹ, còn thiếu nhiều tính năng cần thiết cho một ứng dụng bán hàng.

Trong tương lai, ứng dụng có thể mở rộng với các tính năng như các chương trình giảm giá, xuất báo cáo dạng Excel hay xác thực đa yếu tố.

Tài liệu tham khảo

- [1] “React — react.dev.” <https://react.dev/>. [Accessed 08-06-2025].
- [2] “Next.js by Vercel.” <https://nextjs.org/>. [Accessed 10-06-2025].
- [3] “Keycloak.” <https://www.keycloak.org/>. [Accessed 10-06-2025].
- [4] “PostgreSQL - the world’s most advanced open source database.” <https://www.postgresql.org/>. [Accessed 10-06-2025].
- [5] “Redis - The Real-time data platform.” <https://redis.io/>. [Accessed 10-06-2025].
- [6] “Docker: Accelerated Container Application Development.” <https://www.docker.com/>. [Accessed 10-06-2025].
- [7] “Cloudinary.” <https://cloudinary.com/>. [Accessed 10-06-2025].
- [8] “Spring boot documentation.” <https://spring.io/projects/spring-boot>. [Accessed 10-06-2025].