**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**⸎⸎⸎⸎⸎**



**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN Thiết kế web nâng cao**

**Đề tài :** **Xây dựng website bán game**

Thành viên :Hoàng Ngọc Hưng – 23010239

Giáo viên hướng dẫn :TS.Nguyễn Thị Thu

Lớp tín chỉ : Thiết kế web nâng cao-1-3-24 (TH3)

**HÀ NỘI, THÁNG 6/2025**

Link ReadMe : https://github.com/hoanghung-oop/hung-myWebApp/blob/main/README.md

Link Repo của Project : https://github.com/hoanghung-oop/hung-myWebApp

# Lời mở đầu

Trong thời đại công nghệ phát triển, game console như PS5 đã trở thành một phần không thể thiếu trong đời sống giải trí. Nhận thấy nhu cầu mua bán game PS5 ngày càng tăng, em quyết định xây dựng website GameStore PS5 để cung cấp một nền tảng mua bán game uy tín, tiện lợi và đa dạng.

Dự án này không chỉ giúp người dùng dễ dàng tìm kiếm và mua các tựa game yêu thích mà còn hỗ trợ quản lý hiệu quả cho các nhà bán hàng. Báo cáo sẽ trình bày chi tiết quy trình phân tích, thiết kế, triển khai và định hướng phát triển của hệ thống.

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu về game PS5

PlayStation 5 (PS5) là thế hệ console mới nhất của Sony, mang đến trải nghiệm chơi game đỉnh cao với đồ họa 4K, tốc độ tải nhanh và nhiều tính năng đột phá. Các tựa game PS5 đa dạng từ hành động, phiêu lưu đến thể thao, đáp ứng mọi nhu cầu giải trí.

## 1.2. Mục tiêu của đề tài

* Xây dựng website bán game PS5 với giao diện thân thiện, dễ sử dụng.
* Hỗ trợ tìm kiếm, phân loại game theo thể loại, giá cả, nhà phát hành.
* Tích hợp hệ thống thanh toán trực tuyến an toàn.
* Quản lý kho game, đơn hàng và người dùng hiệu quả.

## 1.3. Phạm vi thực hiện

* Phát triển website trên nền tảng web, chưa triển khai ứng dụng di động.
* Tập trung vào các chức năng cơ bản: đăng ký/đăng nhập, mua hàng, quản lý sản phẩm.
* Chưa tích hợp cổng thanh toán thực tế (mô phỏng giao dịch).

## 1.4. Yêu cầu hệ thống

* Yêu cầu chức năng**:** Đăng ký, đăng nhập, tìm kiếm game, quản lý giỏ hàng, thanh toán.
* Yêu cầu phi chức năng**:** Giao diện thân thiện, bảo mật, tốc độ tải nhanh.

### 1.4.1. Yêu cầu chức năng

Đối với người dùng thông thường, có thể thực hiện các chức năng cơ bản sau:

* + Đăng ký tài khoản.
  + Đăng nhập và quản lý thông tin cá nhân.
  + Tìm kiếm.
  + Mua game.

Đối với người quản trị hệ thống, có quyền truy cập toàn bộ chức năng quản lý và cấu hình website, bao gồm:

* + Quản lý cơ sở dữ liệu, duyệt và xóa các.
  + Quản lý tài khoản người dùng.
  + Cấu hình nội dung trang chủ, danh mục và thông báo hệ thống.

### 1.4.2. Yêu cầu phi chức năng

* Giao diện thân thiện, trực quan, dễ sử dụng cho cả người dùng mới và người dùng có kinh nghiệm.
* Hệ thống hoạt động ổn định, xử lý chính xác các thao tác .
* Tính bảo mật cao, đảm bảo thông tin người dùng và không bị rò rỉ
* Khả năng mở rộng tốt, dễ dàng thêm tính năng như tích hợp cổng thanh toán, phân loại nâng cao, hoặc xuất sang định dạng khác trong tương lai.
* Tối ưu hiệu năng, tiêu tốn ít tài nguyên và phản hồi nhanh với các thao tác của người dùng.

## 1.5. Công nghệ sử dụng

### 1.5.1. Nền tảng Website

Dự án được xây dựng trên nền tảng web sử dụng ngôn ngữ PHP và framework Laravel – một trong những công nghệ phổ biến và mạnh mẽ trong phát triển ứng dụng web hiện nay. Laravel hỗ trợ tổ chức mã nguồn theo mô hình MVC, tích hợp hệ thống định tuyến rõ ràng, cơ chế ORM (Eloquent), bảo mật tốt và khả năng mở rộng cao.

Các công nghệ chính được sử dụng trong dự án:

* HTML (HyperText Markup Language):  
  Ngôn ngữ đánh dấu để xây dựng cấu trúc cơ bản của giao diện người dùng. HTML định nghĩa các phần tử như tiêu đề, đoạn văn bản, hình ảnh, liên kết,...
* CSS (Cascading Style Sheets):  
  Dùng để định dạng và trang trí các phần tử HTML, giúp website hiển thị đẹp mắt và thân thiện với người dùng.
* JavaScript:  
  Ngôn ngữ lập trình phía client, tạo ra các hiệu ứng động và xử lý sự kiện tương tác với người dùng trên trình duyệt.
* PHP:  
  Ngôn ngữ lập trình phía server dùng để xử lý logic ứng dụng, tương tác cơ sở dữ liệu và tạo phản hồi cho các yêu cầu từ trình duyệt.
* Laravel Framework:  
  Laravel là framework PHP hiện đại, hỗ trợ đầy đủ các tính năng như routing, middleware, authentication, validation, migration và tích hợp Eloquent ORM để làm việc với cơ sở dữ liệu.

### 1.5.2. Hệ quản trị cơ sở dữ liệu MySQL

Dự án sử dụng MySQL – một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến, dễ sử dụng và tương thích tốt với Laravel thông qua Eloquent ORM. MySQL cho phép quản lý dữ liệu người dùng, các thao tác quản trị hệ thống.

Một số đặc điểm nổi bật:

* Hỗ trợ ngôn ngữ SQL tiêu chuẩn.
* Tốc độ truy vấn nhanh, phù hợp cho hệ thống web có tải vừa và nhỏ.
* Tích hợp tốt với Laravel qua kết nối database trực tiếp và migration.
* Được cộng đồng rộng lớn hỗ trợ và thường xuyên cập nhật.

### 1.5.3. Mô hình kiến trúc MVC (Model–View–Controller)

Laravel áp dụng mô hình kiến trúc MVC nhằm phân tách rõ ràng ba thành phần chính của ứng dụng:

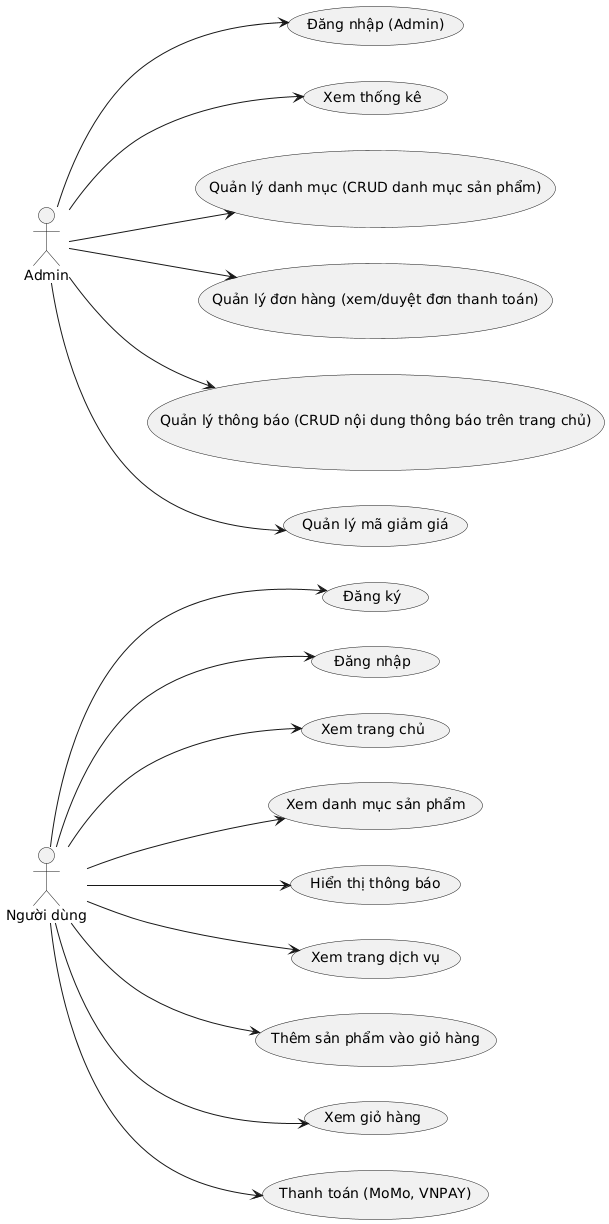
* Model: Quản lý dữ liệu và tương tác với cơ sở dữ liệu thông qua Eloquent ORM. Model đại diện cho các bảng dữ liệu và xử lý các thao tác như thêm, sửa, xóa hoặc truy vấn dữ liệu.
* View: Chịu trách nhiệm hiển thị giao diện người dùng, sử dụng Blade Template Engine để kết hợp HTML với logic hiển thị đơn giản, giúp tạo giao diện linh hoạt và dễ bảo trì.
* Controller: Điều phối luồng xử lý giữa Model và View. Controller tiếp nhận yêu cầu từ người dùng, gọi Model để xử lý dữ liệu và trả kết quả về View để hiển thị.

Việc áp dụng mô hình MVC giúp hệ thống dễ mở rộng, bảo trì hiệu quả và nâng cao khả năng tổ chức mã nguồn. Đây là một trong những điểm mạnh của Laravel, giúp tăng tốc quá trình phát triển ứng dụng web chuyên nghiệp.

# CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

## 2.1. Biểu đồ Use case

### 2.1.1 Use case tổng quát



Hình 2.1. Sơ đồ Use case tổng quát

Hệ thống bao gồm 2 tác nhân chính: Người dùng (User) và Quản trị viên (Admin). Sơ đồ Use-case tổng quát (Hình 2.1) thể hiện tập hợp các chức năng mà mỗi tác nhân có thể thực hiện, cũng như ranh giới hệ thống mà các chức năng đó nằm bên trong.

Tác nhân “Người dùng” có khả năng thực hiện các Use-case sau:

* + - Đăng ký: Tạo mới tài khoản người dùng, cung cấp email, mật khẩu và thông tin cá nhân.
    - Đăng nhập: Sử dụng tài khoản đã đăng ký
    - Xem trang chủ: Truy cập trang chủ, hiển thị banner, nội dung nổi bật.
    - Xem danh mục sản phẩm: Duyệt qua các danh mục sản phẩm.
    - Hiển thị thông báo: Xem các thông báo, khuyến mãi do Admin phát hành.
    - Xem trang dịch vụ: Truy cập thông tin chi tiết các gói dịch vụ của shop.
    - Thêm sản phẩm vào giỏ hàng: Chọn và thêm sản phẩm vào giỏ để chuẩn bị thanh toán.
    - Xem giỏ hàng: Kiểm tra lại các mục đã chọn, cập nhật số lượng hoặc xóa sản phẩm.
    - Thanh toán (MoMo, VNPAY): Chọn phương thức thanh toán online qua MoMo hoặc VNPAY, hoàn tất đơn hàng.

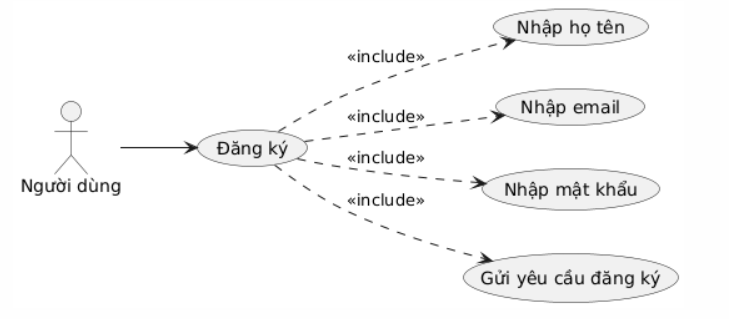
Tác nhân “Admin” chịu trách nhiệm quản lý toàn bộ hệ thống qua các Use-case:

* + - Đăng nhập (Admin): Xác thực quyền truy cập vào khu vực quản trị.
    - Xem thống kê: Khai thác báo cáo doanh thu, số đơn hàng, khách hàng, lượt truy cập.
    - Quản lý danh mục: Tạo, sửa, xóa các danh mục sản phẩm.
    - Quản lý đơn hàng: Xem, duyệt hoặc hủy đơn thanh toán của người dùng.
    - Quản lý thông báo: Tạo, sửa, xóa nội dung thông báo, khuyến mãi hiển thị trên trang chủ.
    - Quản lý mã giảm giá: Tạo, sửa, xóa và theo dõi tình trạng các coupon khuyến mãi.

### 2.1.2 Phân rã Use case

**Phân rã use case người dùng**

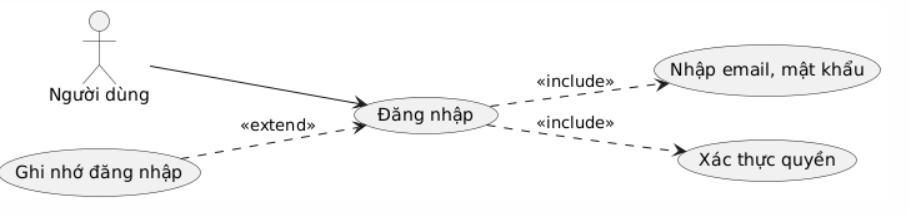
* + - Đăng ký



Hình 2.2. Phân rã use case đăng ký

|  |  |
| --- | --- |
| **Use case** | Đăng ký |
| **Tác nhân** | Người dùng |
| **Mục đích** | Cho phép khách hàng tạo tài khoản mới để truy cập các chức năng của hệ thống. |
| **Điều kiện trước** | Người dùng chưa có tài khoản.  Người dùng đang ở trang Đăng ký. |
| **Điều kiện sau** | Một bản ghi tài khoản mới được lưu trong cơ sở dữ liệu. Mật khẩu được băm an toàn. |

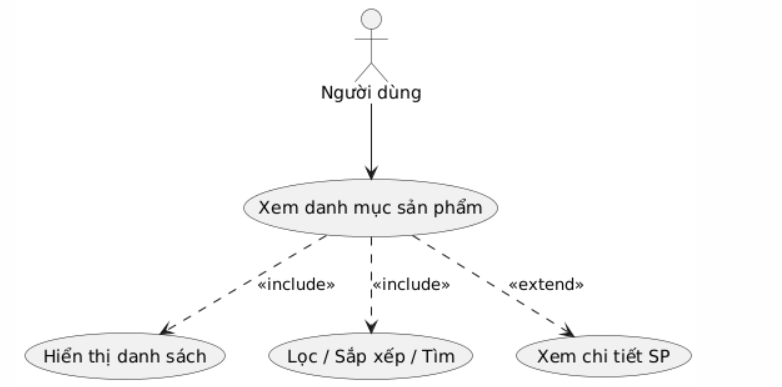
- Đăng nhập



Hình 2.3. Phân rã use case đăng nhập

|  |  |
| --- | --- |
| **Use case** | Đăng nhập |
| **Tác nhân** | Người dùng |
| **Mục đích** | Cho phép khách hàng tạo tài khoản mới để truy cập các chức năng của hệ thống. |
| **Điều kiện trước** | |  | | --- | | Người dùng đã có tài khoản và đang ở trang Đăng nhập |  |  | | --- | |  | |
| **Điều kiện sau** | Người dùng được xác thực thành công, phiên làm việc (session) được tạo, điều hướng vào trang chủ |

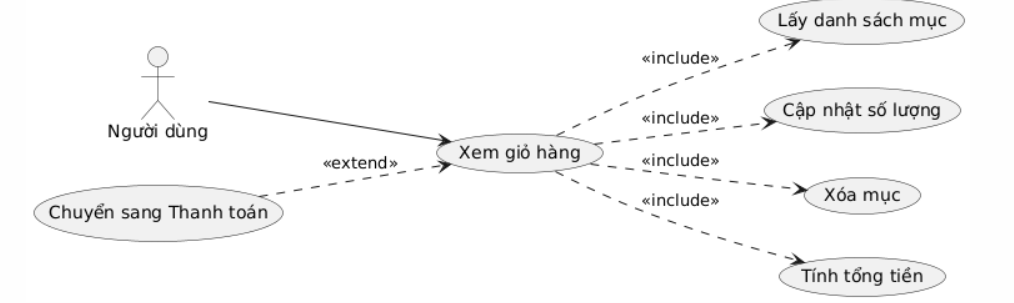
- Xem danh mục sản phẩm



Hình 2.4. Phân rã use case xem danh mục sản phẩm

|  |  |
| --- | --- |
| **Use case** | Xem danh mục sản phẩm |
| **Tác nhân** | Người dùng |
| **Mục đích** | Cho phép người dùng duyệt, lọc và tìm kiếm các sản phẩm có sẵn trong hệ thống |
| **Điều kiện trước** | |  |  |  | | --- | --- | --- | | |  | | --- | | Người dùng đã mở trang “Sản phẩm” (không nhất thiết phải đăng nhập) |  |  | | --- | |  | |  |  | | --- | |  | |
| **Điều kiện sau** | Danh sách sản phẩm được hiển thị theo bộ lọc/tìm kiếm; người dùng có thể chọn “Xem sản phẩm” để xem chi tiết |

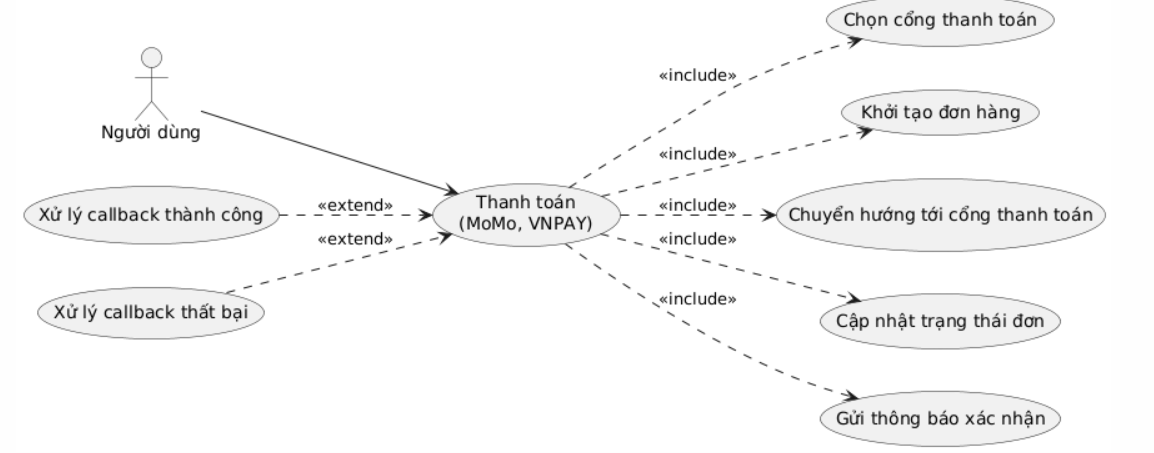
- Xem giỏ hàng:



Hình 2.5. Phân rã use case xem giỏ hàng

|  |  |
| --- | --- |
| **Use case** | Xem giỏ hàng |
| **Tác nhân** | Người dùng |
| **Mục đích** | |  | | --- | | Cho phép người dùng kiểm tra, điều chỉnh các mục trong giỏ trước khi thanh toán |  |  | | --- | |  | |
| **Điều kiện trước** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Đã có ít nhất một sản phẩm trong giỏ (hoặc truy cập trang giỏ hàng) |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |
| **Điều kiện sau** | Danh sách các mục được hiển thị với số lượng và giá từng mục; tổng tiền được tính toán; có thể khởi động luồng thanh toán |

- Thanh toán:

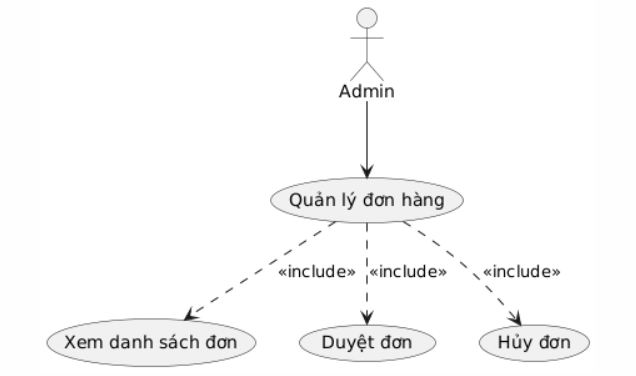


Hình 2.6. Phân rã use case xem danh mục thanh toán

|  |  |
| --- | --- |
| **Use case** | Thanh toán |
| **Tác nhân** | Người dùng |
| **Mục đích** | |  | | --- | | Cho phép người dùng hoàn tất đơn hàng bằng các phương thức: MoMo, VNPAY, COD hoặc chuyển khoản |  |  | | --- | |  | |
| **Điều kiện trước** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | - Giỏ hàng có ít nhất một sản phẩm  - Người dùng đã điền đầy đủ thông tin khách hàng |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |
| **Điều kiện sau** | - Đơn hàng được lưu vào hệ thống với trạng thái tương ứng (Đã thanh toán).  - Khách nhận được xác nhận thành công. |

**Phân rã use case admin**

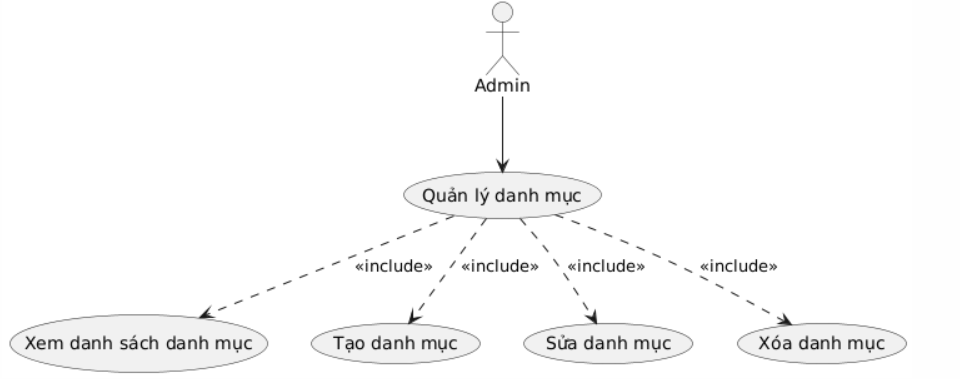
* + - Quản lý đơn hàng



Hình 2.7. Phân rã use case quản lý đơn hàng

|  |  |
| --- | --- |
| **Use case** | Quản lý đơn hàng |
| **Tác nhân** | |  | | --- | | Admin |  |  | | --- | |  | |
| **Mục đích** | |  |  |  | | --- | --- | --- | | |  | | --- | | Xem chi tiết, duyệt hoặc hủy các đơn thanh toán của người dùng |  |  | | --- | |  | |  |  | | --- | |  | |
| **Điều kiện trước** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Admin đã đăng nhập |  |  | | --- | |  | |  |  | | --- | |  | | |  |  | | --- | |  | |
| **Điều kiện sau** | |  | | --- | | Trạng thái đơn được cập nhật (Đã xác nhận/ Đã hủy) | |

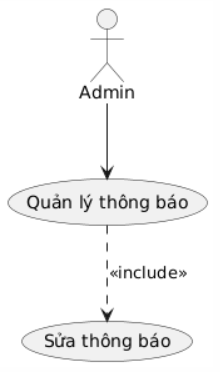
* + - Quản lý danh mục:



Hình 2.8. Phân rã use case quản lý danh mục sản phẩm

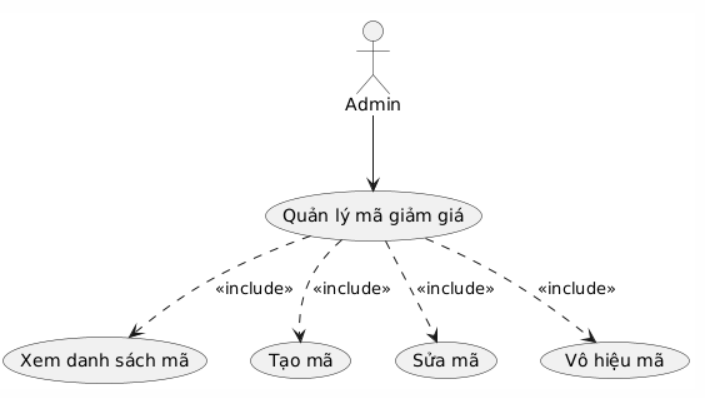
|  |  |
| --- | --- |
| **Use case** | Quản lý sản phẩm |
| **Tác nhân** | |  | | --- | | Admin |  |  | | --- | |  | |
| **Mục đích** | |  |  |  | | --- | --- | --- | | |  | | --- | | Tạo, sửa, xóa và xem danh sách các danh mục sản phẩm |  |  | | --- | |  | |  |  | | --- | |  | |
| **Điều kiện trước** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Admin đã đăng nhập |  |  | | --- | |  | |  |  | | --- | |  | | |  |  | | --- | |  | |
| **Điều kiện sau** | |  |  | | --- | --- | | |  | | --- | | Danh mục được tạo/sửa/xóa trong cơ sở dữ liệu và hiển thị lên web | | |

* + - Quản lý thông báo:



Hình 2.8. Phân rã use case quản lý thông báo

|  |  |
| --- | --- |
| **Use case** | Quản lý thông báo |
| **Tác nhân** | |  | | --- | | Admin |  |  | | --- | |  | |
| **Mục đích** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Tạo, sửa, xóa và bật/tắt hiển thị các thông báo khuyến mãi |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |
| **Điều kiện trước** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Admin đã đăng nhập |  |  | | --- | |  | |  |  | | --- | |  | | |  |  | | --- | |  | |
| **Điều kiện sau** | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | Thông báo được cập nhật lên trang chủ theo trạng thái chọn |   plantuml | | |

* + - Quản lý mã giảm giá: ****

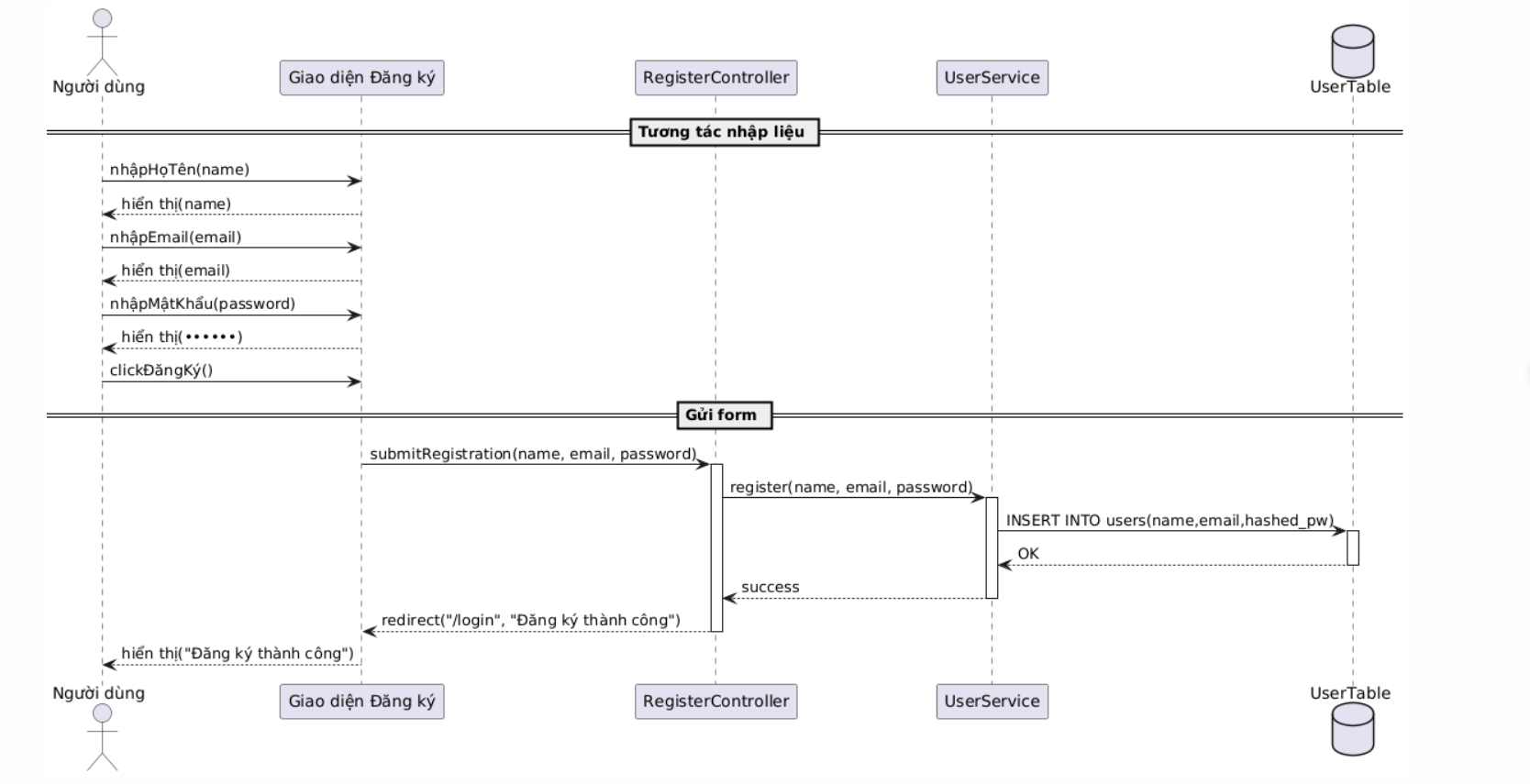
Hình 2.9. Phân rã use case quản lý mã giảm giá

|  |  |
| --- | --- |
| **Use case** | Quản lý mã giảm giá |
| **Tác nhân** | |  | | --- | | Admin |  |  | | --- | |  | |
| **Mục đích** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Tạo, sửa, xóa và giám sát việc sử dụng mã giảm giá |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |
| **Điều kiện trước** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Admin đã đăng nhập |  |  | | --- | |  | |  |  | | --- | |  | | |  |  | | --- | |  | |
| **Điều kiện sau** | |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | |  | | --- | | Mã giảm giá được cập nhật và số lượt sử dụng được ghi nhận | | | | |

## 2.2. Biểu đồ tuần tự

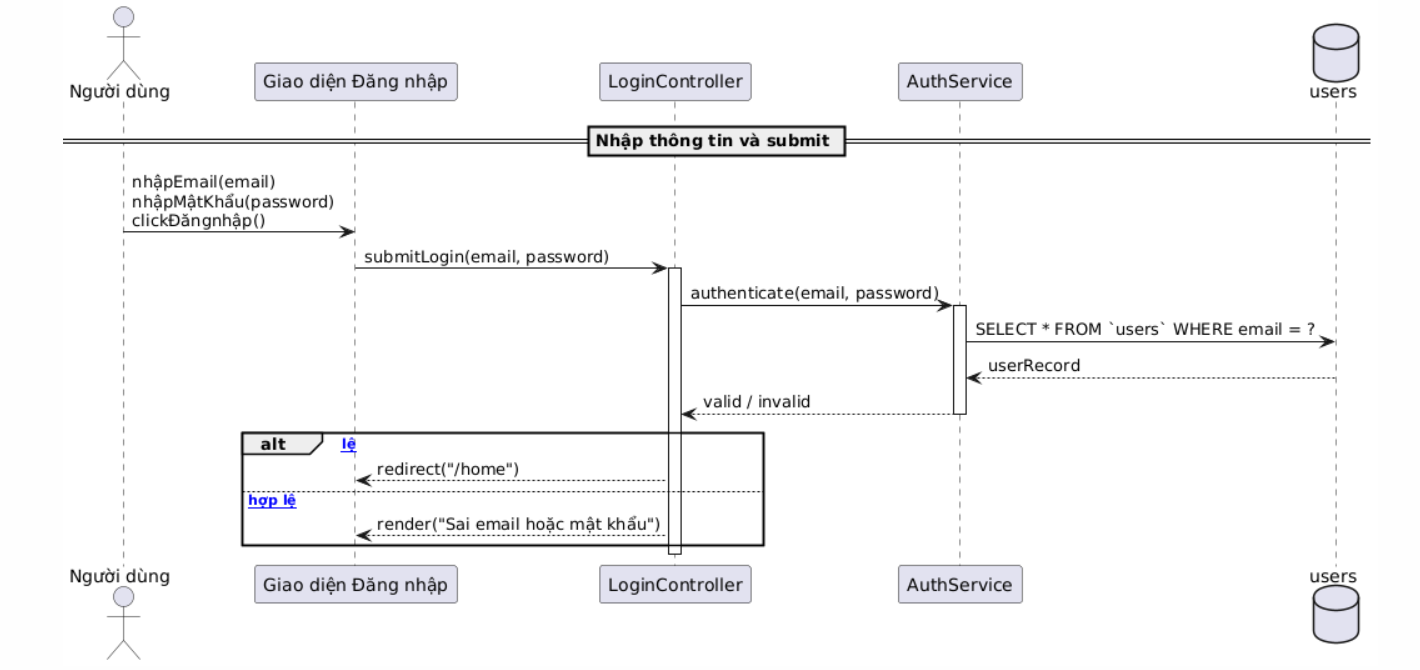
Phần này trình bày thứ tự tương tác giữa các thành phần chính trong ba nghiệp vụ then chốt: Đăng ký, Đăng nhập và Thanh toán

2.2.1. Biểu đồ tuần tự chức năng đăng ký



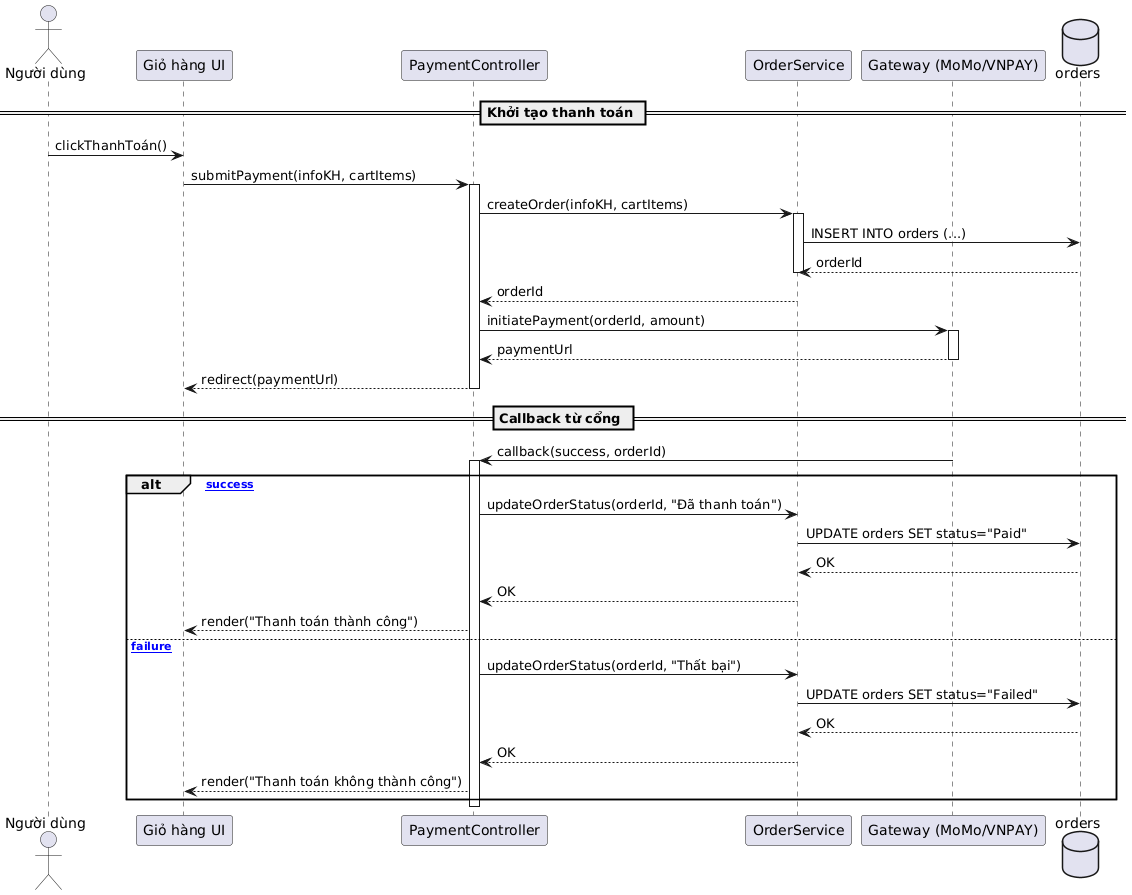
Hình 2.10. Biểu đồ tuần tự chức năng đăng ký

2.2.2. Biểu đồ tuần tự chức năng đăng nhập



Hình 2.11. Biểu đồ tuần tự chức năng đăng nhập

2.2.3. Biểu đồ tuần tự cho chức năng thanh toán



Hình 2.12. Biểu đồ tuần tự chức năng thanh toán

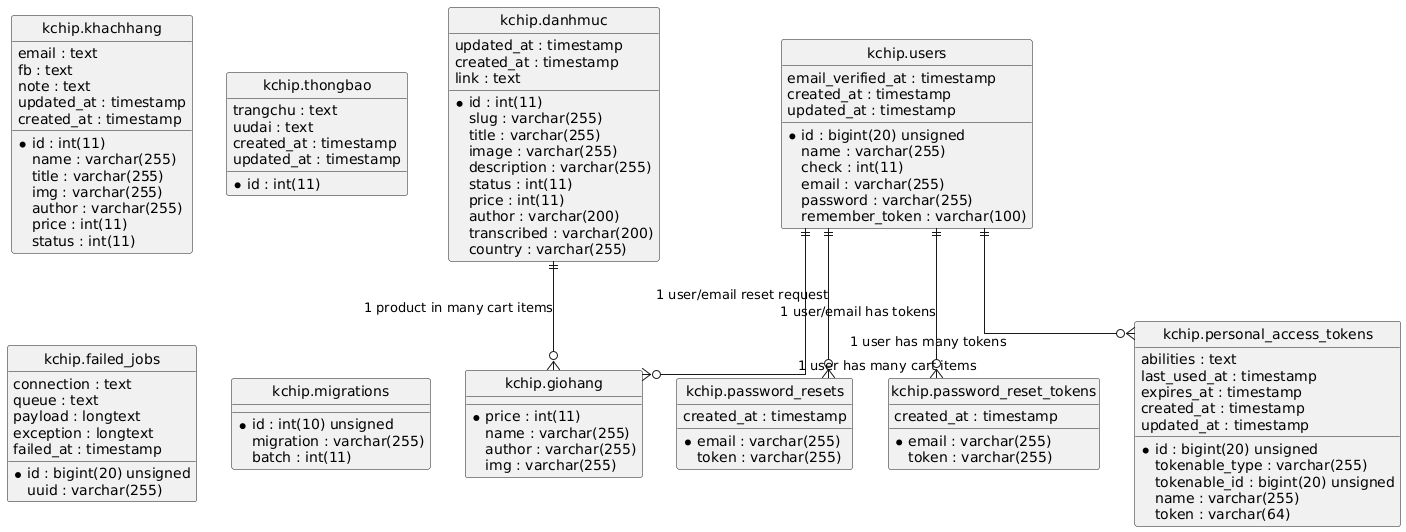
## 2.3. Thiết kế cơ sở dữ liệu

### 2.3.1 Sơ đồ kết nối bảng

Hình 2.38 thể hiện các quan hệ chính giữa các bảng nghiệp vụ trong cơ sở dữ liệu kchip. Trong đó:

* Bảng users là bảng trung tâm, lưu thông tin tài khoản người dùng (id, name, email, password, v.v.).
* Bảng danhmuc chứa danh sách sản phẩm (hoặc danh mục) với các trường (id, slug, title, price, author, v.v.).
* Bảng giohang lưu các mục người dùng thêm vào giỏ, liên kết theo khóa ngoại user\_id → users.id và danhmuc\_id → danhmuc.id.
* Bảng personal\_access\_tokens, password\_resets và password\_reset\_tokens dùng cho chức năng xác thực và đặt lại mật khẩu, đều tham chiếu users.id hoặc users.email.

Việc thiết lập các khóa ngoại này không chỉ đảm bảo tính toàn vẹn dữ liệu (ngăn chặn ghi hoặc xóa rác) mà còn hỗ trợ trực quan hóa sơ đồ ER trong công cụ quản lý cơ sở dữ liệu.



Hình 2.13. Sơ đồ kết nối bảng trong cơ sơ dữ liệu

### Cấu trúc bảng

**Bảng User**

Bảng users lưu trữ thông tin tài khoản của người dùng và admin. Cột check dùng để phân biệt quyền (0 = user, 1 = admin).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Khóa** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | PK | bigint unsigned | Khóa chính, tự động tăng |
| 2 | name |  | varchar(255) | Tên hiển thị của tài khoản |
| 3 | Check |  | int(11) | Phân biệt quyền (0 = user, 1 = admin) |
| 4 | Email |  | varchar(255) | Địa chỉ email dùng để đăng nhập |
| 5 | email\_verified\_at |  | timestamp NULL | Thời điểm xác thực email |
| 6 | password |  | varchar(255) | Mật khẩu đã băm |
| 7 | remember\_token |  | varchar(100) | Token “ghi nhớ đăng nhập” |
| 8 | created\_at |  | timestamp NULL | Thời điểm tạo bản ghi |
| 9 | updated\_at |  | timestamp NULL | Thời điểm cập nhật bản ghi |

Bảng 2.1. Cấu trúc bảng user

**Bảng danhmuc**

Bảng danhmuc chịu trách nhiệm lưu trữ thông tin cơ bản về các sản phẩm và danh mục được phân phối trên. Mỗi bản ghi đại diện cho một mục sản phẩm với đầy đủ metadata như đường dẫn ảnh, mô tả ngắn, giá, tác giả và liên kết đến nội dung demo hoặc video. Trường slug được dùng để tạo URL thân thiện với người dùng và tối ưu SEO. Thông qua cột status, hệ thống có thể linh hoạt bật/tắt hiển thị từng sản phẩm mà không cần xóa dữ liệu gốc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Khóa** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | PK | int | Khóa chính, AUTO\_INCREMENT |
| 2 | slug |  | varchar(255) | Định danh duy nhất dùng trong URL |
| 3 | title |  | varchar(255) | Tiêu đề (tên) sản phẩm |
| 4 | image |  | varchar(255) | Tên hoặc đường dẫn file ảnh đại diện |
| 5 | description |  | varchar(255) | Mô tả ngắn về sản phẩm |
| 6 | price |  | int | Giá sản phẩm |
| 7 | author |  | varchar(200) | Tác giả hoặc nhà xuất bản |
| 8 | transcribed |  | varchar(200) | Thể loại hoặc nội dung mô tả chi tiết |
| 9 | country |  | varchar(255) | Quốc gia sản xuất |
| 10 | link |  | text | Link demo hoặc video (ví dụ YouTube) |
| 11 | status |  | int | Trạng thái hiển thị (1 = Active, 0 = Hidden) |
| 12 | created\_at |  | timestamp | Ngày khởi tạo bản ghi |
| 13 | updated\_at |  | timestamp | Ngày cập nhật bản ghi |

Bảng 2.2. Cấu trúc bảng danhmuc

**Bảng giohang**

Bảng giohang lưu các sản phẩm mà người dùng đã thêm vào giỏ trước khi thanh toán. Mỗi bản ghi đại diện một dòng mục trong giỏ, sao chép các thông tin thiết yếu từ bảng danhmuc để tránh phụ thuộc trực tiếp khi giá hoặc tên sản phẩm thay đổi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên cột | Khóa | Kiểu dữ liệu | Mô tả |
| 1 | id | PK | int | Khóa chính, AUTO\_INCREMENT |
| 2 | price |  | int | Giá sản phẩm tại thời điểm thêm |
| 3 | name |  | varchar(255) | Tên sản phẩm |
| 4 | author |  | varchar(255) | Tác giả/nhà xuất bản của sản phẩm |
| 5 | img |  | varchar(255) | Đường dẫn hoặc tên file ảnh đại diện |

Bảng 2.3. Cấu trúc bảng giohang

**Bảng khachhang**

Bảng khachhang ghi lại thông tin khách đã đặt hàng, bao gồm địa chỉ, liên hệ và các chi tiết liên quan đến đơn hàng. Dữ liệu này dùng để xử lý và giao hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên cột | Khóa | Kiểu dữ liệu | Mô tả |
| 1 | id | PK | int | Khóa chính, AUTO\_INCREMENT |
| 2 | name |  | varchar(255) | Họ và tên người nhận |
| 3 | email |  | text | Địa chỉ email để gửi thông báo đơn hàng |
| 4 | fb |  | text | Link Facebook (nếu khách cung cấp) |
| 5 | note |  | text | Ghi chú đặc biệt (ví dụ: giờ giao hàng) |
| 6 | title |  | varchar(255) | Tên sản phẩm trong đơn |
| 7 | img |  | varchar(255) | Ảnh sản phẩm |
| 8 | author |  | varchar(255) | Tác giả/nhà xuất bản |
| 9 | price |  | int | Tổng tiền của sản phẩm |
| 10 | status |  | int | Trạng thái đơn (0=Chờ, 1=Đã duyệt, 2=Hủy) |
| 11 | created\_at |  | timestamp | Ngày tạo đơn |
| 12 | updated\_at |  | timestamp | Ngày cập nhật đơn |

Bảng 2.4. Cấu trúc bảng khachhang

**Bảng thongbao**

Bảng thongbao quản lý các nội dung hiển thị dưới dạng thông báo và khuyến mãi trên trang chủ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Khóa** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | PK | int | Khóa chính, AUTO\_INCREMENT |
| 2 | trangchu |  | text | Nội dung chính của thông báo |
| 3 | uudai |  | text | Nội dung khuyến mãi |
| 4 | created\_at |  | timestamp | Thời điểm tạo bản ghi |
| 5 | updated\_at |  | timestamp | Thời điểm cập nhật bản ghi |

Bảng 2.5. Cấu trúc bảng thongbao

**Bảng password\_resets**

Bảng lưu trữ các yêu cầu đặt lại mật khẩu (gửi link reset qua email).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Khóa** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | email | PK | varchar(255) | Email người yêu cầu reset |
| 2 | token |  | varchar(255) | Token sinh ra để reset |
| 3 | created\_at |  | timestamp | Thời điểm yêu cầu được tạo |

Bảng 2.6. Cấu trúc bảng password\_resets

**Bảng password\_reset\_tokens**

Bảng này (được Laravel 9+) thay thế password\_resets, lưu lại các token đặt lại password.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Khóa** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | email | PK | varchar(255) | Email người yêu cầu reset |
| 2 | token |  | varchar(255) | Token sinh ra để reset |
| 3 | created\_at |  | timestamp | Thời điểm yêu cầu được tạo |

Bảng 2.7. Cấu trúc bảng password\_reset\_tokens

**Bảng failed\_jobs**  
 Bảng ghi lại các job (tác vụ nền) thực thi thất bại để theo dõi và retry.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Khóa** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | PK | bigint unsigned | Khóa chính, AUTO\_INCREMENT |
| 2 | uuid |  | varchar(255) | Định danh duy nhất của job |
| 3 | connection |  | text | Thông tin kết nối queue |
| 4 | queue |  | text | Tên hàng đợi |
| 5 | payload |  | longtext | Dữ liệu truyền vào job |
| 6 | exception |  | longtext | Chi tiết ngoại lệ khi job thất bại |
| 7 | failed\_at |  | timestamp | Thời điểm job thất bại |

Bảng 2.8. Cấu trúc bảng failed\_jobs

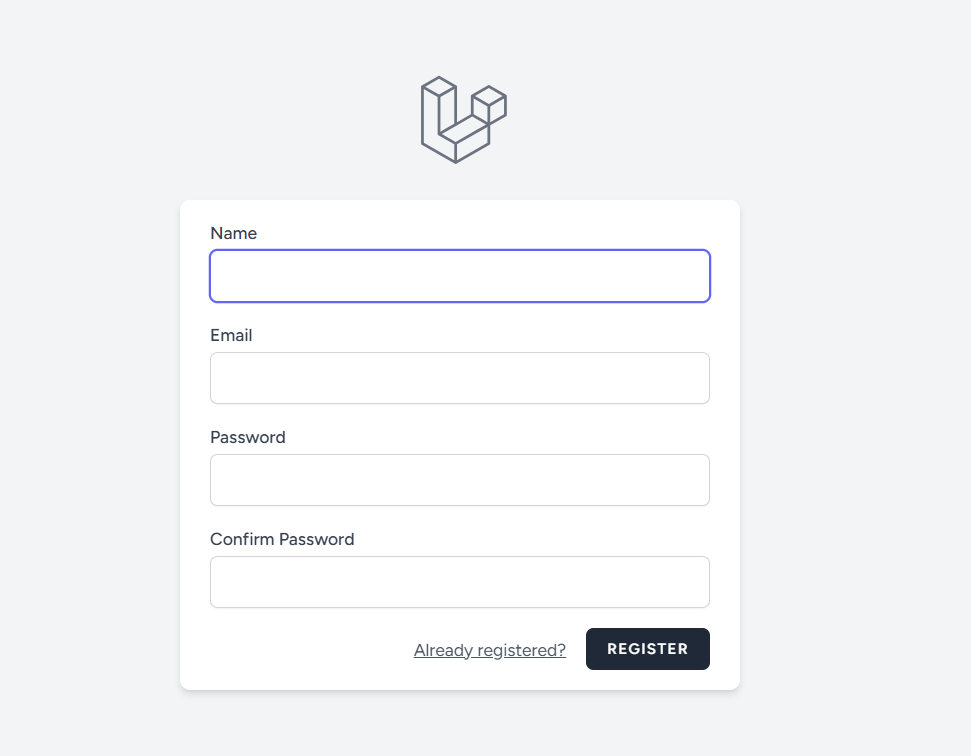
**Bảng migrations**  
 Bảng này do Laravel tạo tự động, ghi lại danh sách migration đã chạy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Khóa** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | PK | int unsigned | Khóa chính, AUTO\_INCREMENT |
| 2 | migration |  | varchar(255) | Tên file migration |
| 3 | batch |  | int | Thứ tự batch khi chạy migration |

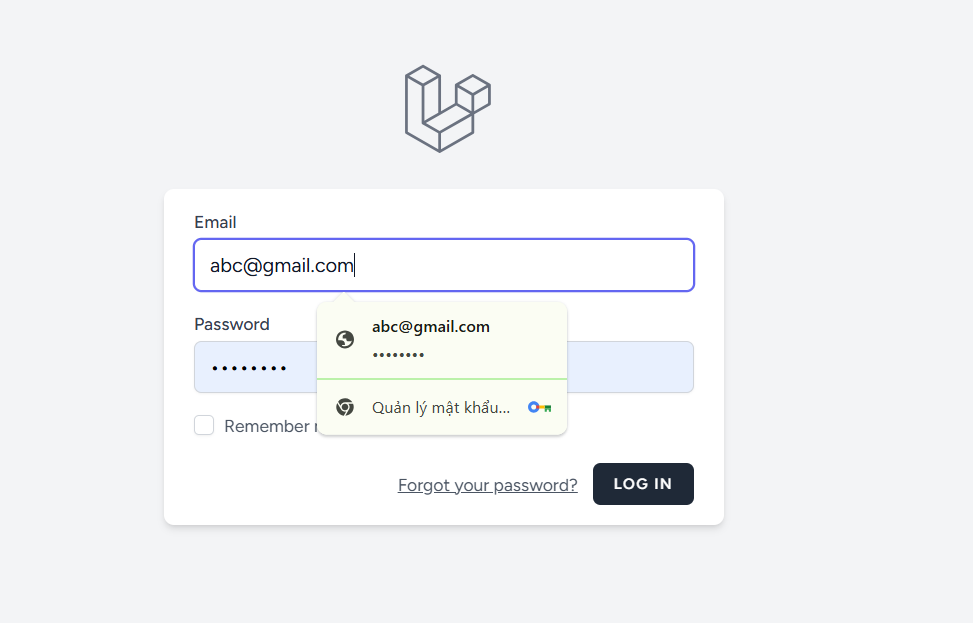
Bảng 2.9. Cấu trúc bảng migrations

# CHƯƠNG 3: THIẾT KẾ GIAO DIỆN

## 3.1. Giao diện dăng ký



## 3.2. Giao diện đăng nhập



## 5.3. Định hướng và đề xuất phát triển trong tương lai

Dựa trên những hạn chế đã nêu và tiềm năng mở rộng của hệ thống, dự án có nhiều hướng phát triển trong tương lai nhằm mở rộng tính năng và nâng cao chất lượng ứng dụng, cũng như chuẩn bị cho việc triển khai thực tế. Cụ thể, một số định hướng và đề xuất chính bao gồm:

* Mở rộng tính năng: Bổ sung các chức năng mới để tăng cường trải nghiệm người dùng và khả năng quản trị. Ví dụ: triển khai hệ thống phân quyền chi tiết cho phép quản lý vai trò và quyền hạn của người dùng linh hoạt hơn (phân biệt rõ các quyền của admin, kiểm duyệt viên, người bán , v.v.); hỗ trợ mô hình multi-tenant để nhiều cửa hàng hoặc nhiều nhóm người dùng có thể sử dụng chung nền tảng nhưng dữ liệu được phân tách độc lập; tích hợp chat nội bộ giữa người dùng với nhau (hoặc giữa người mua và người bán ) để trao đổi thông tin, hỗ trợ tư vấn; và phát triển tính năng gợi ý sản phẩm (recommendation) dựa trên hành vi người dùng (lịch sử tìm kiếm, mua hàng) nhằm cá nhân hóa trải nghiệm, giúp người dùng dễ dàng tiếp cận các sheet nhạc phù hợp với sở thích của mình.
* Nâng cao chất lượng: Tiến hành các cải tiến về mặt kỹ thuật để hệ thống hoạt động ổn định, hiệu quả và an toàn hơn. Cụ thể, cần bổ sung unit test và tích hợp test tự động cho các chức năng quan trọng, qua đó sớm phát hiện lỗi và đảm bảo mỗi lần nâng cấp không làm ảnh hưởng đến tính năng hiện có. Song song, tích hợp quy trình CI/CD (Continuous Integration/Continuous Deployment) giúp tự động hóa việc build, test và deploy, rút ngắn thời gian đưa tính năng mới vào hoạt động thực tế. Hệ thống cũng nên được tối ưu hiệu năng (ví dụ: tối ưu truy vấn database, sử dụng cache cho những dữ liệu truy cập thường xuyên, tối ưu tải trang) nhằm phục vụ tốt hơn khi số lượng người dùng và dữ liệu tăng cao. Cuối cùng, cần tăng cường bảo mật cho ứng dụng: bổ sung xác thực email và có thể cả xác thực hai bư Vớc cho tài khoản người dùng, sử dụng HTTPS/SSL bắt buộc, kiểm tra và vá các lỗ hổng tiềm ẩn (SQL injection, XSS, CSRF) và đảm bảo an toàn cho dữ liệu người dùng.
* Hướng triển khai thực tế: Để đưa dự án vào vận hành thực, cần chuẩn bị hạ tầng và quy trình quản trị hệ thống chuyên nghiệp. Trước hết, việc deploy ứng dụng lên môi trường sản xuất (VPS hoặc hosting) nên được thực hiện, kết hợp với cấu hình domain riêng cho website. Thiết lập chứng chỉ SSL để đảm bảo giao thức HTTPS an toàn cho người dùng khi trao đổi thông tin. Bên cạnh đó, cần xây dựng cơ chế sao lưu (backup) dữ liệu định kỳ nhằm phòng tránh mất mát dữ liệu trong các tình huống sự cố. Đồng thời, tích hợp hệ thống giám sát (monitoring) để theo dõi trạng thái server, lưu lượng truy cập và log lỗi của ứng dụng, giúp đội ngũ quản trị kịp thời phát hiện và xử lý các vấn đề trong quá trình vận hành. Những bước chuẩn bị này sẽ đảm bảo rằng hệ thống có thể phục vụ người dùng thực tế một cách ổn định, liên tục và an toàn.

**TÀI LIỆU THAM KHẢO**

[1] Sommerville, I., “Software Engineering”, 10th Edition, Pearson, 2016.

[2] Pressman, R. S. & Maxim, B. R., “Software Engineering: A Practitioner’s Approach”, 8th Edition, McGraw-Hill, 2014.

[3] Larman, C., “Applying UML and Patterns”, 3rd Edition, Prentice Hall, 2004.

[4] Laravel Documentation, “The PHP Framework for Web Artisans”, https://laravel.com/docs, Truy cập tháng 6/2025.

[5] W3Schools, “HTML/CSS/JS Tutorials”, https://www.w3schools.com, Truy cập tháng 6/2025.

[6] MySQL Documentation, “MySQL 8.0 Reference Manual”, Oracle, https://dev.mysql.com/doc, Truy cập tháng 6/2025.