**TUẦN 1:**

1. **Giới thiệu**
   1. **Mục tiêu**

* Cung cấp nền tảng bán hàng trực tuyến cho phép khách hàng dễ dàng mua sắm, thanh toán, theo dõi và đánh giá sản phẩm.
* Tạo công cụ quản trị tập trung cho quản trị viên trong việc quản lý sản phẩm, đơn hàng và phản hồi của khách hàng.
* Nâng cao trải nghiệm khách hàng thông qua các chức năng thông báo, theo dõi đơn hàng và hệ thống đánh giá minh bạch.
* Hỗ trợ cửa hàng mở rộng hoạt động, quản lý hiệu quả dữ liệu khách hàng, doanh thu và sản phẩm trong tương lai.
  1. **Phạm** vi
* Người dùng ( User/ khách hàng/ customer)
* Đăng ký, đăng nhập và quản lý tài khoản cá nhân.
* Duyệt và tìm kiếm sản phẩm theo danh mục.
* Thêm sản phẩm vào giỏ hàng, đặt hàng và lựa chọn hình thức thanh toán (trực tuyến qua MOMO hoặc thanh toán khi nhận hàng).
* Theo dõi trạng thái đơn hàng (Chờ duyệt, Đang giao, Đã giao).
* Nhận thông báo về đơn hàng (tuỳ chọn).
* Để lại bình luận và đánh giá sản phẩm, quản lý bình luận cá nhân.
* Quản lý hồ sơ cá nhân
* Quản trị viên ( Admin)
* Đăng nhập hệ thống với quyền quản trị.
* Thực hiện CRUD sản phẩm (thêm, sửa, xóa, xem).
* Quản lý đơn hàng: duyệt đơn, xác nhận thanh toán, cập nhật trạng thái giao hàng.
* Quản lý hệ thống thanh toán.
* **Quản lý khách hàng**: Xem thông tin và lịch sử mua hàng của khách hàng
* Quản lý bình luận: xem toàn bộ bình luận, chỉnh sửa hoặc xóa bình luận vi phạm.
* Xem báo cáo: doanh thu, số lượng sản phẩm bán ra.
* Nền tảng triển khai
* Hệ thống hoạt động trên nền web.
* Có khả năng mở rộng và tích hợp thêm các phương thức thanh toán, vận chuyển, marketing trong tương lai.
  1. **Yêu cầu phi chức năng**
* **Hiệu năng:** Thời gian phản hồi khi người dùng tìm kiếm sản phẩm nhanh.
* **Tính bảo mật:** Hệ thống phải mã hóa mật khẩu người dùng và đảm bảo an toàn thông tin thanh toán qua Momo.
* **Tính dễ sử dụng:** Giao diện phải thân thiện, dễ thao tác cho cả người dùng lần đầu.
* **Khả năng mở rộng:** Hệ thống có thể dễ dàng mở rộng để tích hợp thêm nhiều phương thức thanh toán hoặc số lượng sản phẩm lớn hơn trong tương lai.

1. **Danh sách Actor và Use Case**

|  |  |  |
| --- | --- | --- |
| Actor  Use Case | Customer | Admin |
| Login | - Đăng nhập vào hệ thống để sử dụng chức năng.  - Luồng chính:  + Người dùng nhập username và password.  + Hệ thống kiểm tra thông tin (include: Check user/password).  + Nếu sai → extend: Show error. Nếu đúng → vào hệ thống. | |
| View | - Xem danh sách sản phẩm, thông tin cửa hàng.  - Luồng chính:  + Customer chọn chức năng View.  + Hệ thống hiển thị thông tin sản phẩm. |  |
| OrderFlower | - Đặt mua hoa.  - Luồng chính:  + Customer chọn sản phẩm (include: Product). + Customer thêm vào giỏ hàng.  + Customer xác nhận đơn hàng. |  |
| Pay/Checkout | - Thanh toán đơn hàng đã đặt.  - Luồng chính:  + Customer chọn phương thức thanh toán.  + Có thể: Pay with money / Pay with QR( MoMo).  + Hệ thống xác nhận giao dịch. | - Xác nhận thanh toán |
| Product |  | - Quản lý sản phẩm (thêm, sửa, xóa). - Luồng chính:  + Admin truy cập chức năng Product.  + Thực hiện thêm/sửa/xóa thông tin sản phẩm. |
| Analyst |  | - Phân tích dữ liệu, xem báo cáo doanh thu, đơn hàng.  - Luồng chính:  + Admin chọn chức năng Analyst.  + Hệ thống tổng hợp dữ liệu đơn hàng, hiển thị báo cáo. |

**TUẦN 2**

1. **Biểu đồ Use Case**

- Với tác nhân là “ Khách hàng”

- Sơ đồ này thể hiện mối quan hệ giữa các tác nhân và các use case chính của hệ thống

- Với tác nhân là “ Quản trị viên”

-Đăng nhập (Log in)

-Quản lý sản phẩm (Product Management)

-Quản lý khách hàng (Customer Management)

-Quản lý đơn hàng( Order Management)

-Quản lý bình luận(Comment Management)

-Thống Kê Đơn Hàng (Analyst)

1. **Viết kịch bản chi tiết**

**- Use Case 1: Đăng nhập**

**--- Primary Scenario**

* Người dùng (hoặc Admin) mở trang đăng nhập.
* Nhập tên đăng nhập và mật khẩu.
* Hệ thống kiểm tra thông tin.
* Nếu hợp lệ → đăng nhập thành công, chuyển đến trang chủ (dashboard).

**--- Alternative Scenario**

* Nếu sai tên đăng nhập/mật khẩu → hệ thống hiển thị thông báo lỗi.
* Nếu tài khoản bị khóa → hệ thống từ chối đăng nhập và báo cho người dùng liên hệ quản trị.

**- Use Case 2: Xem sản phẩm (View Product)**

**--- Primary Scenario**

* Khách hàng đăng nhập.
* Khách hàng truy cập danh mục sản phẩm.
* Hệ thống hiển thị danh sách sản phẩm (tên, giá, hình ảnh, số lượng tồn).
* Khách hàng chọn sản phẩm để xem chi tiết.

**--- Alternative Scenario**

* Nếu không tìm thấy sản phẩm → hệ thống báo "Sản phẩm không tồn tại".

**- Use Case 3: Đặt hàng (OrderFlower)**

**--- Primary Scenario**

* Khách hàng đăng nhập.
* Khách hàng chọn sản phẩm và thêm vào giỏ hàng.
* Khách hàng mở giỏ hàng và nhấn “Đặt hàng”.
* Hệ thống tạo đơn hàng ở trạng thái "Chờ duyệt".

**--- Alternative Scenario**

* Nếu giỏ hàng rỗng → hệ thống không cho đặt hàng.
* Nếu sản phẩm không đủ số lượng → hệ thống thông báo lỗi và yêu cầu điều chỉnh.

**- Use Case 4: Thanh toán (Pay/Checkout)**

**--- Primary Scenario**

* Khách hàng mở đơn hàng.
* Khách hàng chọn phương thức thanh toán:
* Thanh toán khi nhận hàng → hệ thống ghi nhận, đơn hàng chờ admin duyệt.
* Thanh toán qua MoMo (QR Code) → hệ thống hiển thị thông tin sản phẩm cần thanh toán.
* Khách hàng thực hiện thanh toán.
* Hệ thống cập nhật trạng thái đơn hàng thành “Đã thanh toán - Chờ duyệt”.

**--- Alternative Scenario**

* Nếu khách hàng thoát giỏ hàng trước khi thanh toán → đơn hàng chưa được tính mua.

**- Use Case 5: Quản lý sản phẩm (Admin – Product Management)**

**--- Primary Scenario**

* Admin đăng nhập.
* Admin vào mục Quản lý sản phẩm.
* Admin có thể thực hiện:
* Thêm sản phẩm (Create).
* Chỉnh sửa sản phẩm (Update).
* Xóa sản phẩm (Delete).
* Xem danh sách sản phẩm (Read).

**- Use Case 6: Quản lý đơn hàng (Admin – Order Management)**

**- Primary Scenario**

* Admin đăng nhập.
* Admin mở danh sách đơn hàng.
* Admin duyệt đơn và thay đổi trạng thái:

- Chờ duyệt → Đang giao.

- Đang giao → Đã giao.

* Hệ thống lưu thay đổi và thông báo cho khách hàng.

**--- Alternative Scenario**

* Nếu thanh toán chưa xác nhận → Admin từ chối duyệt đơn.
* Nếu đơn bị hủy → Admin cập nhật trạng thái thành “Cancelled”.

**Use Case 7: Đánh giá sản phẩm (Customer Review)**

**--- Primary Scenario**

* Khách hàng đăng nhập.
* Khách hàng mở lịch sử đơn hàng.
* Chọn sản phẩm đã mua và để lại đánh giá (sao + bình luận).
* Hệ thống lưu đánh giá và hiển thị công khai.

**--- Alternative Scenario**

* Nếu bình luận vi phạm (spam, ngôn ngữ xấu) → Admin có thể chỉnh sửa hoặc xóa.

**TUẦN 3**

1. Biểu đồ lớp

<https://drive.google.com/file/d/1iQPvfDTDjqYsZb_7EpJj_6R9rOtktXpV/view?usp=sharing>

1. Lập trình
2. **File CustomerController.php ( Quản lý khách hàng)**

**- Vị trí**: app/Http/Controllers/CustomerController.php

**- Chức năng**: Quản lý khách hàng (CRUD) và xem đơn hàng của từng khách hàng.

- Các thành phần chính:

* index(): Liệt kê khách hàng.
* create(): Hiển thị form thêm mới.
* store(): Thêm khách hàng sau khi validate.
* edit(): Hiển thị form chỉnh sửa.
* update(): Cập nhật thông tin khách hàng.
* destroy(): Xóa khách hàng.
* orders(): Xem danh sách đơn hàng của khách hàng.



1. **ProductController.php (Quản lý sản phẩm)**

**- Vị trí**: app/Http/Controllers/ProductController.php

**- Chức năng**: Quản lý sản phẩm (CRUD), hiển thị sản phẩm cho người dùng, quản lý đánh giá và phân tích sản phẩm.

- Các thành phần chính:

* **index()**: Liệt kê sản phẩm (dành cho admin).
* **create()**: Hiển thị form thêm mới sản phẩm.
* **store()**: Thêm sản phẩm sau khi validate và upload ảnh.
* **edit()**: Hiển thị form chỉnh sửa sản phẩm.
* **update()**: Cập nhật thông tin sản phẩm.
* **destroy()**: Xóa sản phẩm.
* **show()**: Xem chi tiết sản phẩm.
* **shop()**: Hiển thị danh sách sản phẩm cho người dùng, có tìm kiếm và lọc theo giá.
* **suggest()**: Gợi ý sản phẩm cùng danh mục.
* **storeReview()**: Thêm đánh giá sản phẩm.
* **updateReview()**: Cập nhật đánh giá sản phẩm.
* **deleteReview()**: Xóa đánh giá sản phẩm.
* **analytics()**: Phân tích số liệu bán hàng và đánh giá (chỉ dành cho admin).



1. **OrderController.php (Quản lý đơn hàng)**

- Vị trí: app/Http/Controllers/OrderController.php  
- Chức năng: Quản lý đơn hàng (CRUD, duyệt trạng thái) và hỗ trợ khách hàng theo dõi, hủy hoặc xác nhận đơn hàng.

- Các thành phần chính:

* index(): Liệt kê và lọc đơn hàng (theo trạng thái, thời gian).
* create(): Hiển thị form tạo đơn hàng (chọn khách hàng, sản phẩm).
* store(): Lưu đơn hàng mới và chi tiết sản phẩm.
* updateStatus(): Cập nhật trạng thái đơn hàng (admin).
* show(): Hiển thị chi tiết đơn hàng.
* history(): Khách hàng xem lịch sử đơn hàng (lọc theo trạng thái, thời gian).
* cancel(): Khách hàng hủy đơn hàng (nếu còn pending/processing).
* confirmDelivery(): Khách hàng xác nhận đã nhận hàng.
* destroy(): Xóa đơn hàng (admin).
* update(): Chuẩn resource, cập nhật trạng thái đơn hàng.



1. **CartController ( Quản lý giỏ hàng)**

- Vị trí: app/Http/Controllers/CartController.php

- Chức năng: Quản lý đơn hàng (CRUD, duyệt trạng thái) và hỗ trợ khách hàng theo dõi, hủy hoặc xác nhận đơn hàng.

- Các thành phần chính:

* index(): Liệt kê và lọc đơn hàng (theo trạng thái, thời gian).
* create(): Hiển thị form tạo đơn hàng (chọn khách hàng, sản phẩm).
* store(): Lưu đơn hàng mới và chi tiết sản phẩm.
* updateStatus(): Cập nhật trạng thái đơn hàng (admin).
* show(): Hiển thị chi tiết đơn hàng.
* history(): Khách hàng xem lịch sử đơn hàng (lọc theo trạng thái, thời gian).
* cancel(): Khách hàng hủy đơn hàng (nếu còn pending/processing).
* confirmDelivery(): Khách hàng xác nhận đã nhận hàng.
* destroy(): Xóa đơn hàng (admin).
* update(): Chuẩn resource, cập nhật trạng thái đơn hàng.



1. **CheckoutController.php (Xử lý thanh toán)**

- Vị trí: app/Http/Controllers/CheckoutController.php

- Chức năng: Xử lý thanh toán (COD, Chuyển khoản, MoMo), quản lý thông tin giao hàng, tạo đơn hàng từ giỏ hàng hoặc "Mua ngay", và xử lý callback từ cổng thanh toán.

- Các thành phần chính:

* show(): Hiển thị trang thanh toán (lấy giỏ hàng hoặc giỏ “Mua ngay”, điền sẵn thông tin giao hàng).
* process(): Xử lý đặt hàng, lưu đơn hàng và chi tiết sản phẩm, cập nhật thông tin khách hàng.
* execPostRequest(): Hàm tiện ích gửi request POST đến API (sử dụng cURL).
* momo\_payment(): Tạo đơn hàng và gửi request đến cổng thanh toán MoMo, sau đó chuyển hướng đến trang thanh toán MoMo.
* momoIpn(): Nhận callback từ MoMo (IPN), xác thực và cập nhật trạng thái đơn hàng (pending → paid).



1. **AdminController.php (Quản trị hệ thống)**

- Chức năng: Quản trị hệ thống, hiển thị dashboard, thống kê, quản lý sản phẩm, khách hàng và đơn hàng.

- Các phương thức chính:

* index(): Hiển thị trang admin chính.
* construct(): Middleware xác thực, chỉ cho phép user đã đăng nhập.
* dashboard(): Hiển thị trang tổng quan gồm thống kê sản phẩm, khách hàng, đơn hàng, người dùng; tìm kiếm sản phẩm; lọc đơn hàng theo trạng thái, thời gian; tính toán các chỉ số doanh thu và số lượng đơn hàng.
* updateOrderStatus(): Cập nhật trạng thái đơn hàng (chỉ admin).  
  destroyOrder(): Xóa đơn hàng (chỉ admin).  
  destroyProduct(): Xóa sản phẩm (chỉ admin, kèm xóa ảnh nếu tồn tại).

****

**TUẦN 4**

1. **Sequence Diagram - Quản lý sản phẩm**

* **Thêm sản phẩm mới**

A screenshot of a diagram

AI-generated content may be incorrect.

Amin

Amin

* **Cập nhật sản phẩm**

A screenshot of a computer

AI-generated content may be incorrect.

Amin

Amin

* **Xóa sản phẩm**

A diagram of a company

AI-generated content may be incorrect.

Amin

Amin

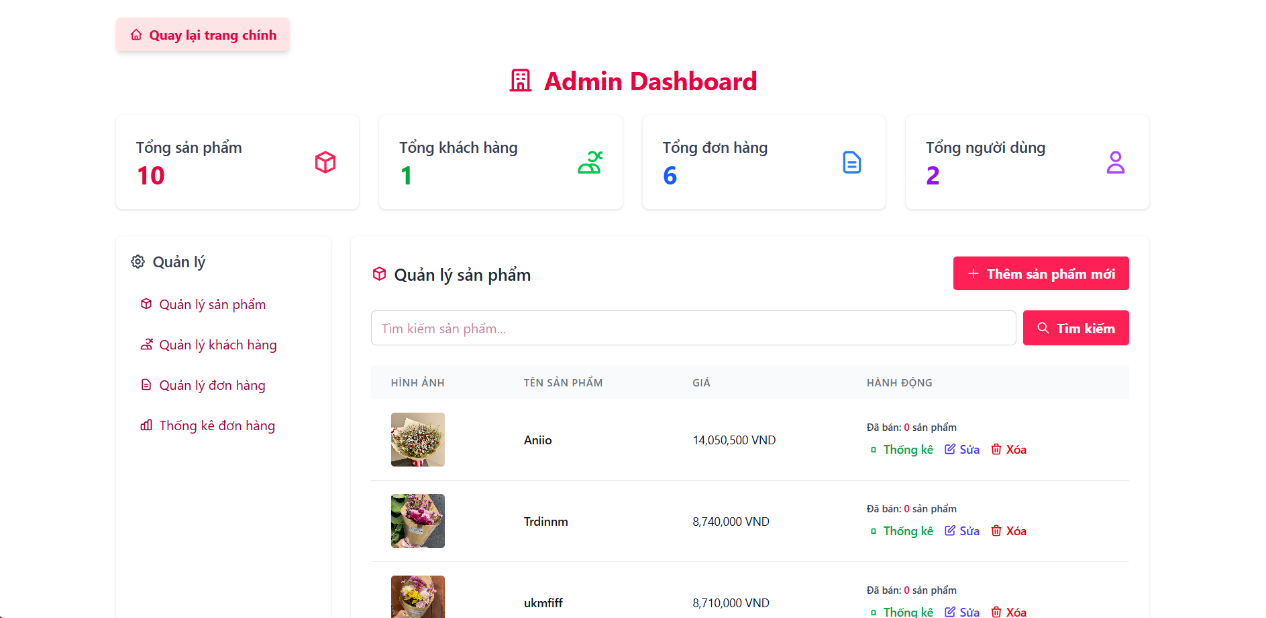
* **Xem chi tiết sản phẩm**

A diagram with text and arrows

AI-generated content may be incorrect.

Amin

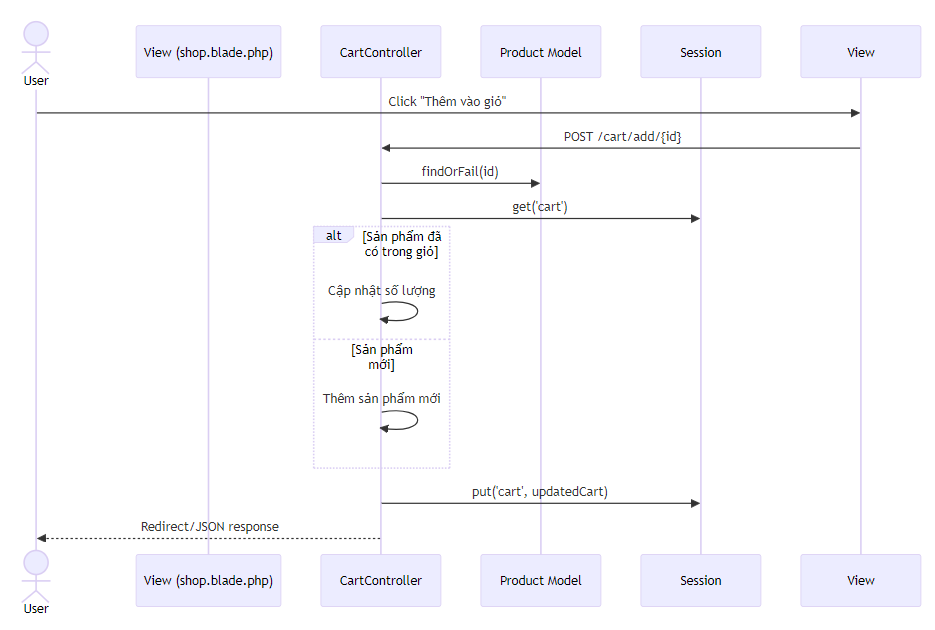
Amin



*(Giao diện Quản lý sản phẩm)*

1. **Sequence Diagram – Order Sản phẩm**

* **Thêm sản phẩm vào giỏ hàng**



* **Thanh toán**

A screenshot of a computer

AI-generated content may be incorrect.

* **Xử lý đơn hàng**

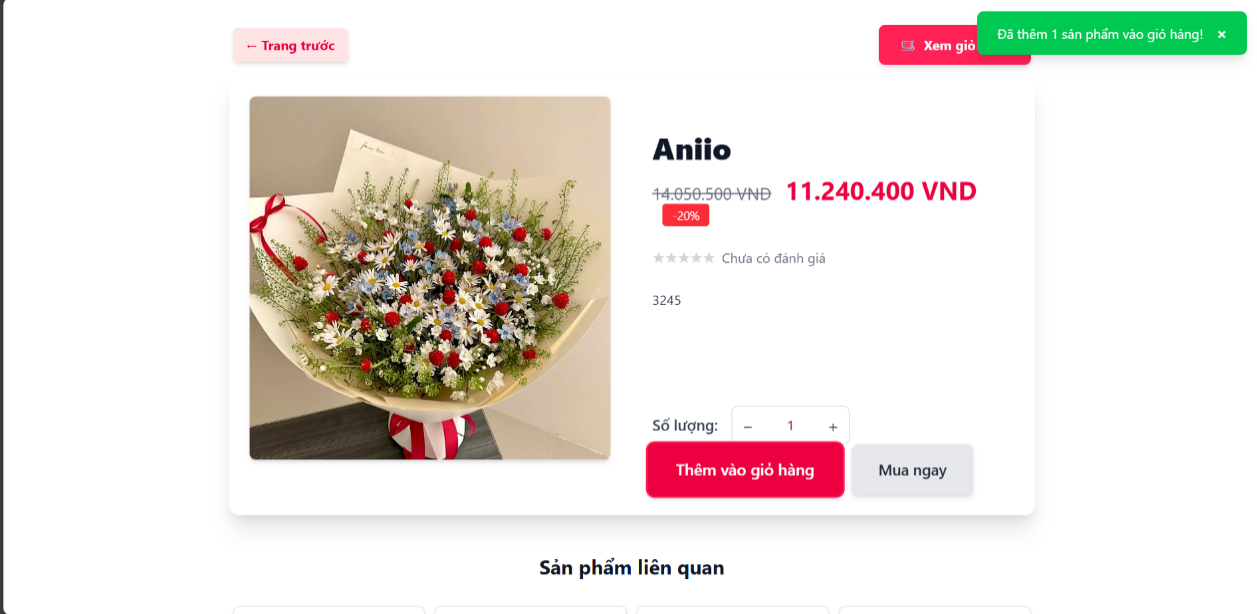
A diagram of a software application

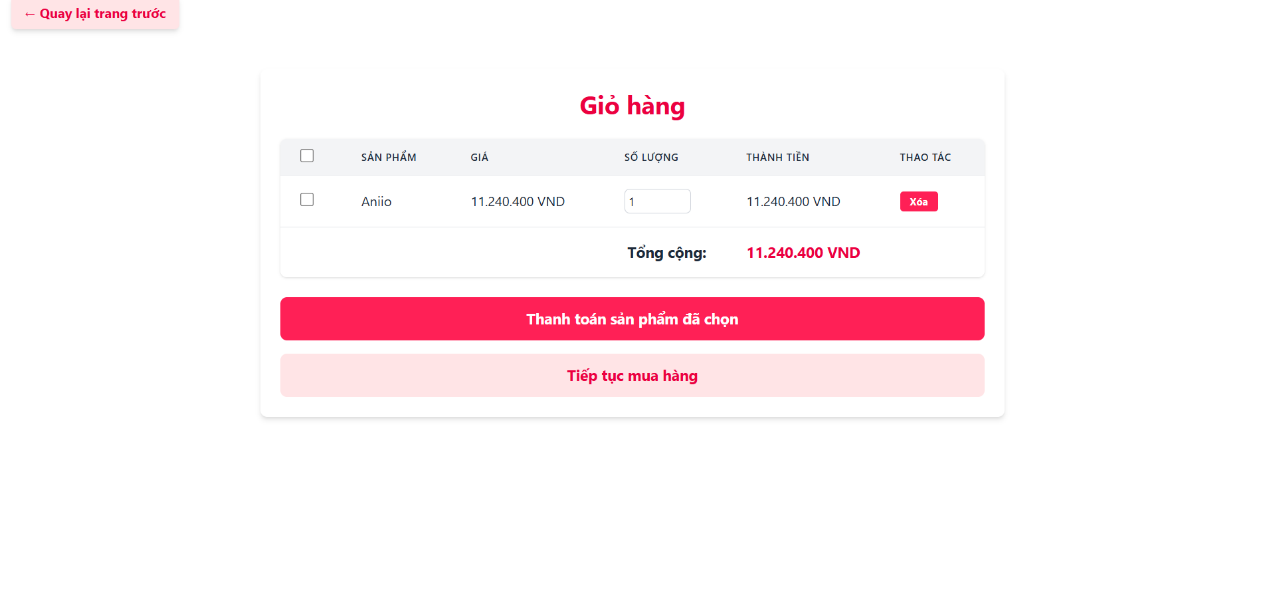
AI-generated content may be incorrect.

**-** **Sequence Diagram Quy trình Thanh toán Chi tiết**

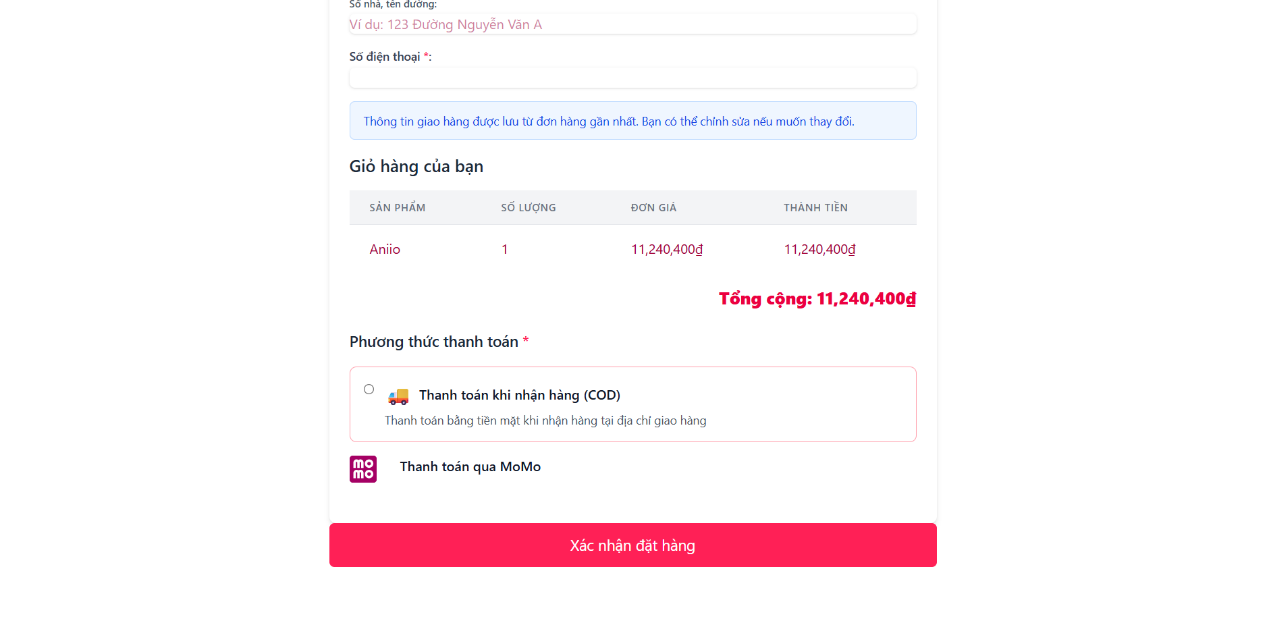
A screenshot of a computer program

AI-generated content may be incorrect.

****

****

***(Giao diện thêm giỏ hàng)***

******

***A screenshot of a qr code

AI-generated content may be incorrect.***

***(Thanh toán đơn hàng)***

1. **Sequence Diagram – Quy trình đánh giá sản phẩm**

* **Thêm đánh giá**

A diagram of a company

AI-generated content may be incorrect.

* **Cập nhật đánh giá**

A diagram of a product

AI-generated content may be incorrect.

* **Xóa đánh giá**

A diagram of a product control system

AI-generated content may be incorrect.

* **Hiển thị đánh giá**

A diagram of a product

AI-generated content may be incorrect.

**4. Sequence Diagram – Thống kê báo cáo**

* **Thống kê sản phẩm**

A diagram of a company

AI-generated content may be incorrect.

* **Thống kê đơn hàng**

A diagram of a computer program

AI-generated content may be incorrect.

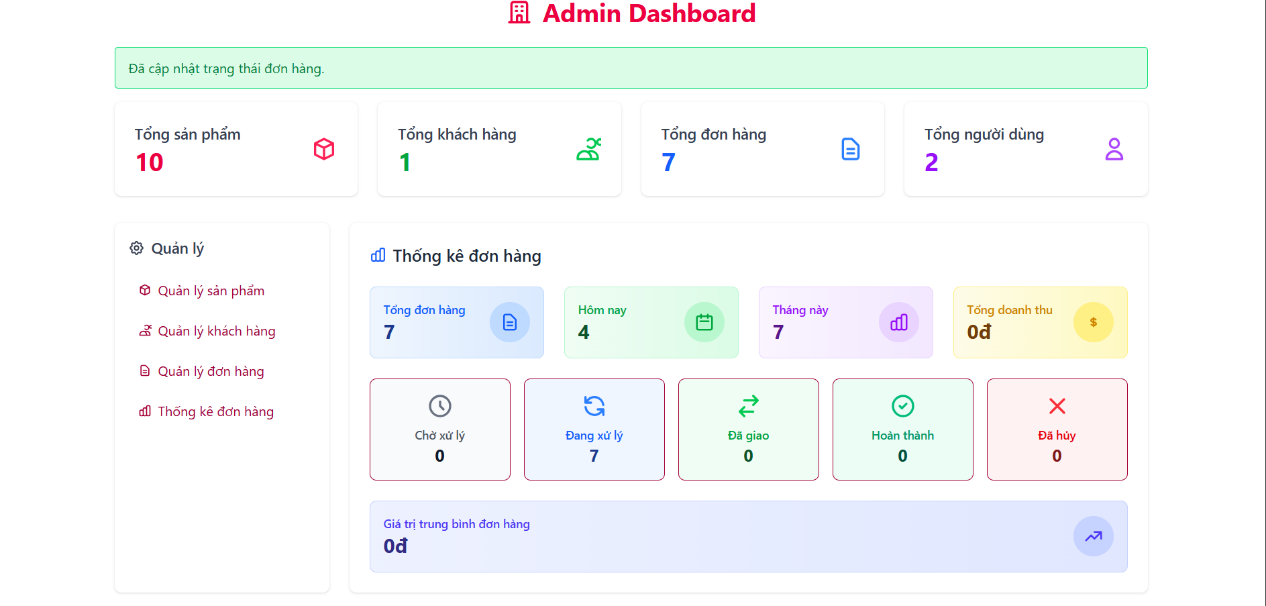
* **Thống kê doanh thu**

A diagram of a process

AI-generated content may be incorrect.

* **Thống kê đánh giá**

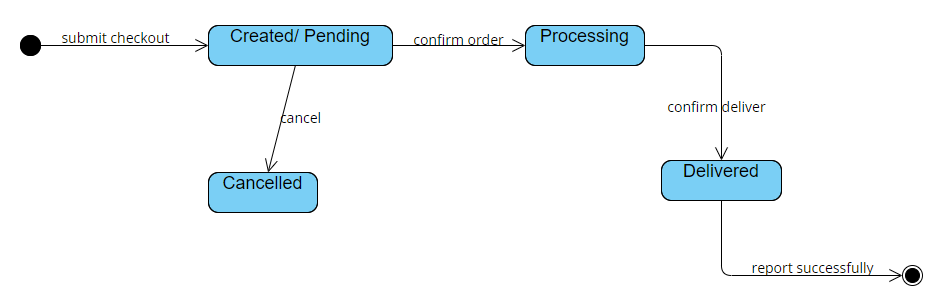
A diagram of a product

AI-generated content may be incorrect.

***(Giao diện thống kê đơn hàng)***

**Tuần 5**

1. Biểu đồ trạng thái State Machine Diagram

****

Trong hệ thống, đơn hàng được quản lý qua 4 trạng thái chính. Các trạng thái này phản ánh vòng đời xử lý của một đơn hàng từ khi khởi tạo cho đến khi hoàn tất hoặc bị hủy.

## 1. Pending (Chờ xử lý)

* Mô tả: Đơn hàng vừa được tạo sau khi khách hàng checkout, chưa được xác nhận.
* Chuyển đổi:
  + Sang *Processing* khi quản trị viên xác nhận đơn.
  + Sang *Cancelled* nếu khách hàng hủy.

## 2. Processing (Đang xử lý)

* Mô tả: Đơn hàng đã được xác nhận và đang trong quá trình xử lý.
* Chuyển đổi:
  + Sang *Delivered* khi đơn hàng được xác nhận đã giao thành công.

## 3. Delivered (Đã nhận)

* Mô tả: Đơn hàng đã được giao và xác nhận hoàn tất.
* Chuyển đổi:
  + Đây là trạng thái cuối cùng, không còn chuyển sang trạng thái khác.

## 4. Cancelled (Đã hủy)

* Mô tả: Đơn hàng bị hủy bởi khách hàng hoặc quản trị viên trước khi hoàn tất.
* Chuyển đổi:
  + Đây là trạng thái cuối cùng, không còn xử lý tiếp.

## *Minh họa trong code*

Các trạng thái trên được định nghĩa trong phương thức getStatuses() của lớp Order:

public static function getStatuses()

{

    return [

        'pending' => 'Chờ xử lý',

        'processing' => 'Đang xử lý',

        'delivered' => 'Đã nhận',

        'cancelled' => 'Đã hủy'

    ];

}

1. **Thuộc tính quản lý trạng thái**

* status: Thuộc tính chính lưu trữ trạng thái hiện tại của đơn hàng. Trạng thái có thể mang các giá trị:
* pending: Đơn hàng chờ xử lý.
* processing: Đơn hàng đang được xử lý.
* delivered: Đơn hàng đã được giao/nhận.
* cancelled: Đơn hàng đã bị hủy.
* getStatusNameAttribute: Thuộc tính động, trả về tên trạng thái bằng tiếng Việt, giúp hiển thị thân thiện với người dùng. Ví dụ: pending → *Chờ xử lý*.
* getStatusColorAttribute: Thuộc tính động, quy định màu sắc hiển thị cho từng trạng thái, nhằm tăng tính trực quan. Ví dụ: pending → *vàng*, processing → *xanh dương*, delivered → *xanh lá*, cancelled → *đỏ*.

## Phương thức quản lý trạng thái

* getStatuses(): Trả về danh sách tất cả các trạng thái của đơn hàng dưới dạng cặp key-value (giá trị trong cơ sở dữ liệu và tên hiển thị tiếng Việt).
* getStatusNameAttribute(): Xác định và trả về tên tiếng Việt của trạng thái hiện tại.
* getStatusColorAttribute(): Xác định và trả về lớp màu CSS phù hợp với trạng thái.
* Các phương thức đề xuất bổ sung cho xử lý nghiệp vụ.
  + isPending(): Kiểm tra đơn hàng có đang ở trạng thái *pending* hay không.
  + markAsProcessing(): Chuyển đơn hàng từ *pending* sang *processing*.
  + markAsDelivered(): Chuyển đơn hàng từ *processing* sang *delivered*.
  + cancel(): Hủy đơn hàng nếu trạng thái hiện tại là *pending* hoặc *processin*