**Câu 1 Quản lý giỏ hàng**

1. **User story**

Là một người dùng tôi muốn thêm các sản phẩm vào giỏ hàng để chuẩn bị thanh toán đặt hàng.

1. **Tiêu chí chấp nhận**

* Người dùng có thể chọn sản phẩm và số lượng mong muốn, sau đó thêm vào giỏ hàng thành công.
* Sau khi thêm, giỏ hàng hiển thị đúng sản phẩm và tổng số lượng.
* Người dùng có thể kiểm tra các sản phẩm đã thêm trong giỏ hàng và thấy tổng giá trị các sản phẩm có trong giỏ hàng.

1. **Độ phức tạp và mức độ ưu tiên**

* Độ phức tạp: Trung bình (3/5)

Vì:

* Yêu cầu về xử lý logic: Tính năng này yêu cầu xử lý logic để thêm sản phẩm vào giỏ hàng, bao gồm việc kiểm tra số lượng, cập nhật danh sách giỏ hàng, và tính toán tổng giá trị. So với các tính năng đơn giản khác (như hiển thị danh mục), tính năng này phức tạp hơn một chút do các bước xử lý bổ sung cần thiết để đảm bảo tính chính xác và nhất quán của giỏ hàng.
* Thiết kế giao diện và trải nghiệm người dùng: Thiết kế cần cung cấp giao diện thân thiện, dễ hiểu cho người dùng trong việc chọn số lượng sản phẩm và thêm vào giỏ hàng, đồng thời cho phép họ xem thông tin cập nhật về các sản phẩm đã thêm vào.
* Yêu cầu kết nối với cơ sở dữ liệu: Tính năng này cũng đòi hỏi thao tác với cơ sở dữ liệu để lưu thông tin sản phẩm và cập nhật khi có thay đổi từ người dùng.
* Mức độ ưu tiên: Cao

Vì:

* Trực tiếp ảnh hưởng đến trải nghiệm người dùng: Tính năng "Thêm sản phẩm vào giỏ hàng" là một trong những tính năng cốt lõi và có ảnh hưởng trực tiếp đến trải nghiệm người dùng khi mua sắm. Nếu tính năng này không hoạt động tốt, người dùng sẽ không thể tiếp tục quá trình mua hàng.
* Mức độ quan trọng trong trải nghiệm của người dùng: Việc thêm sản phẩm vào giỏ hàng là bước quan trọng để chuyển đổi từ người xem thành người mua. Nếu thiếu tính năng này, hệ thống mua sắm sẽ không hoàn thiện, ảnh hưởng lớn đến các mục tiêu kinh doanh và doanh thu.

1. **Kịch bản theo cấu trúc BDD**

* Scenario: Thêm sản phẩm vào giỏ hàng.
* Given người dùng đã chọn sản phẩm và số lượng mong muốn
* When người dùng nhấn nút "Thêm vào giỏ hàng"
* Then sản phẩm được thêm vào giỏ hàng và hiển thị chính xác số lượng, tổng giá trị trên giao diện giỏ hàng.

1. **Task cho user story**
   1. Front-end:
2. Thiết kế giao diện chọn sản phẩm và số lượng:

* Tạo UI trên trang sản phẩm với nút "Thêm vào giỏ hàng".
* Thêm trường nhập số lượng cho người dùng chọn (có thể là dropdown hoặc ô nhập với số lượng tối đa).

1. Tạo UI giỏ hàng:

* Tạo trang hoặc modal hiển thị giỏ hàng với danh sách sản phẩm đã thêm, số lượng, và giá từng sản phẩm.
* Hiển thị tổng số lượng sản phẩm và tổng giá trị của giỏ hàng.

1. Thêm và cập nhật giỏ hàng trên giao diện người dùng:

* Khi người dùng nhấn "Thêm vào giỏ hàng", giao diện cần hiển thị sản phẩm được thêm và cập nhật giỏ hàng theo thời gian thực.
* Tạo hiệu ứng hoặc thông báo nhỏ (ví dụ: "Đã thêm vào giỏ hàng") để xác nhận thao tác của người dùng.

1. Xử lý lỗi giao diện:

* Hiển thị thông báo khi người dùng cố gắng thêm số lượng vượt quá số lượng trong kho hoặc khi gặp lỗi kết nối.
  1. Back-end:

1. Xây dựng API thêm sản phẩm vào giỏ hàng:

* Tạo API endpoint POST /cart/add nhận dữ liệu từ front-end (gồm user\_id, product\_id, và quantity).
* Xác thực và xử lý dữ liệu đầu vào (kiểm tra định dạng, kiểm tra số lượng hợp lệ).

1. Xử lý logic thêm sản phẩm:

* Kiểm tra trong cơ sở dữ liệu để đảm bảo số lượng sản phẩm còn đủ trong kho.
* Nếu sản phẩm đã có trong giỏ hàng của người dùng, tăng số lượng hiện có; nếu không có, thêm sản phẩm mới vào giỏ hàng.

1. Cập nhật giỏ hàng trong cơ sở dữ liệu:

* Nếu hợp lệ, thêm hoặc cập nhật bản ghi trong bảng Cart với các thông tin user\_id, product\_id, quantity, và total\_price.
* Tính toán tổng giá trị của sản phẩm (dựa trên số lượng và giá sản phẩm) và lưu vào cơ sở dữ liệu.

1. Xử lý lỗi và phản hồi cho front-end:

* Trả về các mã lỗi nếu số lượng yêu cầu vượt quá số lượng có sẵn hoặc gặp lỗi trong quá trình lưu trữ.
* Gửi phản hồi JSON về trạng thái thao tác (thành công/thất bại) để front-end cập nhật giao diện.

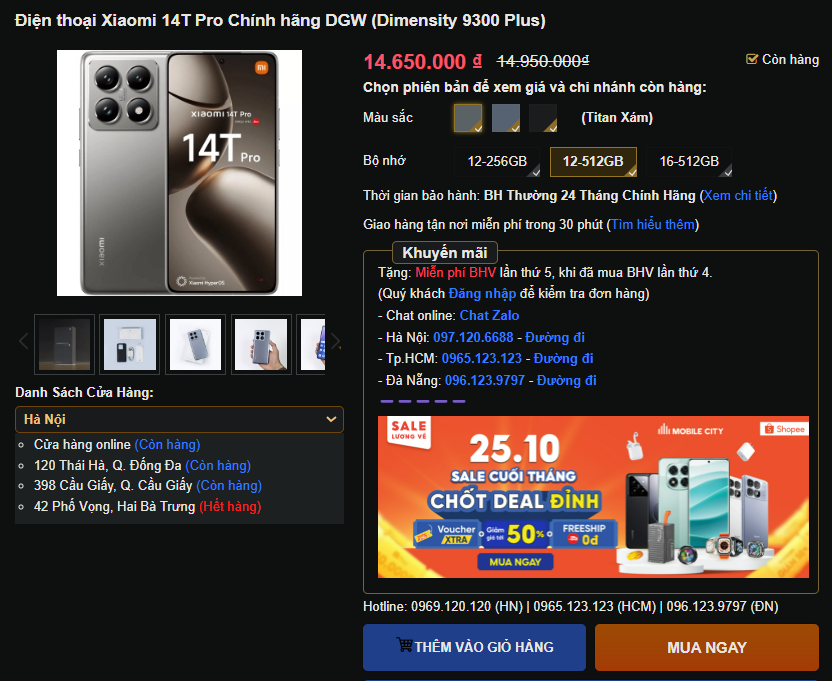
1. Kiểm tra bảo mật:

* Xác thực người dùng đã đăng nhập và có quyền thực hiện thao tác này.
* Đảm bảo người dùng chỉ có thể thêm sản phẩm vào giỏ hàng của họ.

1. Kiểm thử API:

* Kiểm thử các trường hợp (như: thêm sản phẩm hợp lệ; số lượng vượt quá kho; trường hợp sản phẩm đã có trong giỏ hàng)

1. **Thiết kế giao diện**



1. **Thiết kế logic**

* Xử lý yêu cầu từ front-end: Nhận yêu cầu thêm sản phẩm vào giỏ hàng với user\_id, product\_id, và quantity.
* Kiểm tra tồn kho: Kiểm tra số lượng sản phẩm còn trong kho.
* Cập nhật giỏ hàng:
* Nếu sản phẩm đã có trong giỏ hàng, tăng số lượng hiện có.
* Nếu sản phẩm chưa có, thêm sản phẩm mới với số lượng và giá trị tương ứng.
* Tính toán tổng giá trị: Cập nhật lại tổng giá trị giỏ hàng sau khi thêm sản phẩm.
* Trả phản hồi về front-end: Gửi dữ liệu cập nhật giỏ hàng hoặc mã lỗi nếu thao tác thất bại.

1. **Thiết kế Cơ sở dữ liệu**

* Bảng Cart:
* cart\_id (khóa chính)
* user\_id (khóa ngoại tham chiếu bảng User)
* product\_id (khóa ngoại tham chiếu bảng Product)
* quantity (số lượng sản phẩm trong giỏ)
* total\_price (tổng giá trị cho từng sản phẩm trong giỏ)
* Bảng Product:
* product\_id
* name
* price
* stock (số lượng tồn kho)

1. **Ca kiểm thử**

* Thêm sản phẩm với số lượng hợp lệ: Xác minh sản phẩm được thêm vào giỏ hàng thành công và hiển thị đúng số lượng.
* Thêm sản phẩm với số lượng vượt quá tồn kho: Xác minh thông báo lỗi được hiển thị khi số lượng sản phẩm vượt quá kho hàng.
* Thêm sản phẩm đã có trong giỏ hàng: Xác minh số lượng sản phẩm tăng lên trong giỏ hàng thay vì tạo bản ghi mới.
* Hiển thị giỏ hàng: Kiểm tra xem giỏ hàng hiển thị đúng sản phẩm, số lượng, và tổng giá trị.
* Thêm sản phẩm khi chưa đăng nhập: Xác minh người dùng được yêu cầu đăng nhập trước khi thêm sản phẩm vào giỏ hàng.

**Câu 2 Quản lý danh mục hàng**

1. **User Story**

Là một quản trị viên, tôi muốn thêm danh mục hàng mới để tổ chức các sản phẩm theo danh mục, giúp người dùng dễ dàng tìm kiếm.

1. **Tiêu chí chấp nhận**

* Quản trị viên có thể thêm danh mục hàng mới với tên danh mục duy nhất và mô tả ngắn.
* Sau khi thêm thành công, danh mục mới sẽ hiển thị trong hệ thống quản lý danh mục.
* Nếu tên danh mục trùng lặp hoặc bỏ trống, hệ thống hiển thị thông báo lỗi phù hợp.

1. **Độ phức tạp và mức độ ưu tiên**

* Độ phức tạp: Thấp - Trung bình (2) – yêu cầu đơn giản về xử lý và tương tác cơ sở dữ liệu.
* Mức độ ưu tiên: Trung bình – cần thiết để cấu trúc hệ thống sản phẩm nhưng không ảnh hưởng trực tiếp đến trải nghiệm người dùng mua hàng.

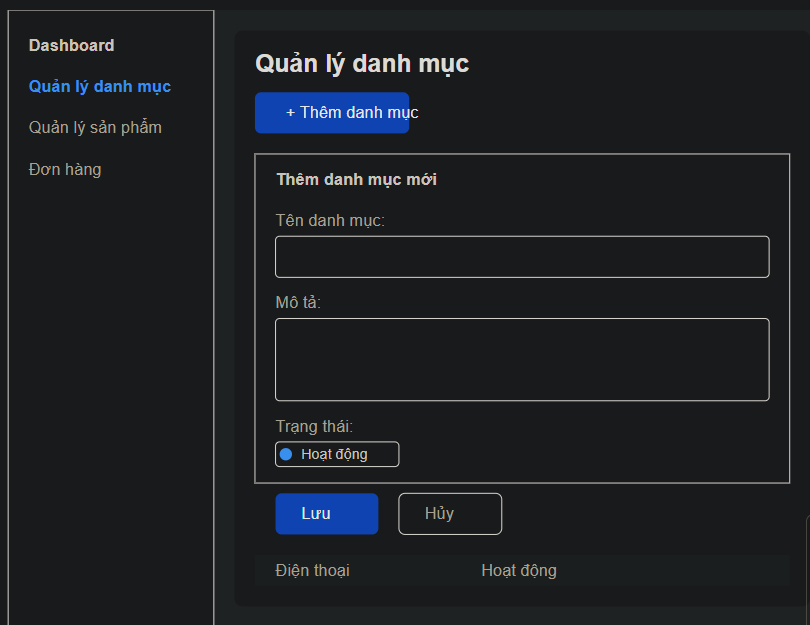
1. **Kịch bản BDD**

* **Scenario**: Thêm danh mục hàng mới.
  + **Given** quản trị viên đã đăng nhập vào hệ thống quản trị.
  + **When** quản trị viên điền thông tin danh mục mới và nhấn "Lưu".
  + **Then** danh mục mới được thêm vào cơ sở dữ liệu và hiển thị trong danh sách danh mục.

1. **Tasks cho User Story**

5.1 Front-end

1. Tạo giao diện thêm danh mục mới:
   * Tạo trang hoặc modal với form nhập gồm: tên danh mục (bắt buộc) và mô tả ngắn (tùy chọn).
   * Thêm nút "Lưu" để xác nhận tạo danh mục mới.
2. Xử lý sự kiện khi nhấn nút "Lưu":
   * Khi quản trị viên điền thông tin và nhấn "Lưu", gửi yêu cầu tới back-end để thêm danh mục mới.
3. Hiển thị thông báo lỗi nếu có:
   * Nếu tên danh mục trống hoặc trùng lặp, hiển thị thông báo lỗi ngay trên form.
4. Cập nhật danh sách danh mục sau khi thêm thành công:
   * Nếu thêm thành công, tự động cập nhật danh sách danh mục để hiển thị danh mục mới mà không cần tải lại trang.
   1. Back-end
   2. Xây dựng API thêm danh mục:
   * Tạo endpoint POST /categories/add để nhận yêu cầu thêm danh mục từ front-end.
   * API này sẽ nhận thông tin category\_name và description từ yêu cầu gửi đến.
   1. Xác thực và xử lý dữ liệu:
   * Kiểm tra category\_name không được để trống.
   * Kiểm tra category\_name là duy nhất, đảm bảo không trùng lặp với danh mục đã tồn tại trong cơ sở dữ liệu.
   1. Thêm danh mục mới vào cơ sở dữ liệu:
   * Nếu hợp lệ, thêm bản ghi mới vào bảng Categories với các thông tin category\_name và description.
   1. Xử lý lỗi và gửi phản hồi:
   * Nếu category\_name trống hoặc trùng lặp, trả về mã lỗi cùng thông báo thích hợp để front-end hiển thị cho quản trị viên.
   * Nếu thêm thành công, trả về phản hồi JSON xác nhận với dữ liệu danh mục mới để front-end cập nhật.
   1. Kiểm thử API:
   * Kiểm thử các trường hợp thêm danh mục mới, tên danh mục trùng lặp và tên danh mục để trống.
5. **Thiết kế Giao diện**

****

1. **Thiết kế Logic**
2. **Xử lý yêu cầu từ front-end**:
   * + - Khi quản trị viên nhấn "Lưu", front-end gửi yêu cầu POST /categories/add với dữ liệu danh mục tới back-end.
3. **Xác thực dữ liệu trên back-end**:
   * + - Kiểm tra tên danh mục không trống.
       - Kiểm tra tên danh mục là duy nhất (không trùng lặp trong cơ sở dữ liệu).
4. **Lưu danh mục vào cơ sở dữ liệu**:
   * + - Nếu hợp lệ, lưu danh mục vào bảng Categories với category\_name và description.
5. **Trả phản hồi về front-end**:
   * + - Nếu thao tác thành công, back-end gửi thông báo xác nhận và thông tin của danh mục mới.
       - Nếu có lỗi, back-end trả mã lỗi kèm thông báo để front-end hiển thị thông báo lỗi.
6. **Cập nhật giao diện**:

* Front-end nhận phản hồi thành công và cập nhật danh sách danh mục để hiển thị danh mục mới.
* Nếu có lỗi, hiển thị thông báo lỗi ngay dưới form để quản trị viên biết và sửa thông tin.

1. **Thiết kế Cơ sở Dữ liệu**

* **Bảng Categories**:
  + category\_id (khóa chính)
  + category\_name (tên danh mục, duy nhất, không được để trống)
  + description (mô tả danh mục)

1. **Ca kiểm thử cho User Story**
2. **Thêm danh mục với tên hợp lệ**:
   * Đảm bảo danh mục được thêm vào cơ sở dữ liệu và hiển thị trong danh sách danh mục sau khi thêm.
3. **Thêm danh mục khi tên danh mục trùng lặp**:
   * Kiểm tra thông báo lỗi hiển thị khi tên danh mục đã tồn tại.
4. **Thêm danh mục khi tên danh mục để trống**:
   * Đảm bảo hệ thống từ chối thao tác và hiển thị thông báo yêu cầu nhập tên danh mục.
5. **Thêm danh mục với mô tả tùy chọn**:
   * Đảm bảo danh mục được thêm thành công dù trường mô tả để trống.
6. **Hiển thị danh sách danh mục sau khi thêm**:
   * Kiểm tra danh sách danh mục tự động cập nhật sau khi thêm mới mà không cần tải lại trang.