|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  **--------------------------**    **BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**  **TÊN ĐỀ TÀI:**  **XÂY DỰNG WEBSITE BÁN ĐIỆN THOẠI CHO CỬA HÀNG SƠN THẢO.**   |  | | --- | | **GVHD:** ThS. Vũ Đức Huy | | **Sinh viên:** Phạm Văn Thành | | **Mã sinh viên:** 2021600986 | | **Lớp:** 2021DHCNTT01 – K16 |   **Hà Nội – Năm 2025** |

**LỜI CẢM ƠN**

Thưa thầy Vũ Đức Huy

Em muốn dành những lời cảm ơn chân thành nhất để bày tỏ sự biết ơn đến thầy về sự hỗ trợ và định hướng tận tâm của thầy trong suốt quá trình thực hiện đồ án tốt nghiệp của em. Quá trình nghiên cứu và hoàn thiện đồ án tốt nghiệp không hề dễ dàng, nhưng có thầy bên cạnh, em đã học được rất nhiều từ kiến thức chuyên môn, kỹ năng nghiên cứu, đến những giá trị đạo đức nghề nghiệp và tư duy nghiêm túc trong nghiên cứu khoa học.

Thầy không chỉ là người hướng dẫn cho công việc nghiên cứu mà còn là người đã truyền đạt cho em tinh thần trách nhiệm cao, lòng say mê và khát khao tiến bộ trong lĩnh vực của mình. Những lời động viên, nhận xét, góp ý từ thầy đã giúp em hoàn thiện và nâng cao chất lượng của đồ án.

Ngoài ra, em cũng muốn bày tỏ lòng biết ơn đến các thầy/cô trong khoa Công nghệ thông tin, trường Đại học công nghiệp Hà Nội đã tạo điều kiện tốt nhất cho em trong suốt quá trình thực hiện đồ án.

Cuối cùng em xin được gửi lời cảm ơn tới bố mẹ, gia đình, bạn bè … những người luôn đồng hành, động viên, khích lệ, tạo điều kiện, giúp đỡ em trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp này.

Trong quá trình hoàn thành đồ án tốt nghiệp không thể tránh khỏi thiếu sót, kính mong có sự góp ý từ thầy cô.

Chúc thầy cô luôn thành công trong công việc giảng dạy và nghiên cứu, và sẽ tiếp tục lan tỏa ngọn lửa đam mê cho các thế hệ sinh viên sau này.

Trân trọng, Phạm Văn Thành.

**PHẦN MỞ ĐẦU**

1. **Lý do chọn đề tài**

Ngày nay, việc mua sắm trực tuyến (online) đã trở thành một thói quen rất phổ biến với người tiêu dùng. Chỉ với một chiếc máy tính xách tay hay điện thoại có kết nối Internet, khách hàng có thể tìm kiếm, so sánh và đặt mua các sản phẩm mọi lúc mọi nơi. Lĩnh vực kinh doanh đồ điện tử nói chung và ngành kinh doanh điện thoại di động nói riêng là một trong những ngành cạnh tranh rất cao, sản phẩm thay đổi nhanh và liên tục, nên việc cập nhật thông tin và tiếp cận khách hàng kịp thời là vô cùng quan trọng.

Cửa hàng **Sơn Thảo** hiện đang chủ yếu bán hàng trực tiếp, chưa tận dụng được kênh bán hàng trực tuyến. Điều này khiến việc tiếp cận khách hàng từ xa gặp rất nhiều hạn chế, trong khi đó nhiều đối thủ cạnh tranh đã nhanh chóng chuyển sang kinh doanh online.

Vì vậy, xây dựng **Website bán điện thoại di động cho cửa hàng Sơn Thảo** sẽ giúp mở rộng thị trường, cung cấp thông tin đầy đủ và thuận tiện cho khách hàn, hỗ trợ thanh toán – đặt hàng trực tuyến, đồng thời nâng cao hình ảnh và cạnh tranh thương hiệu.

1. **Mục tiêu của đề tài**

Đề tài hướng đến việc xây dựng một website bán điện thoại di động cho cửa hàng Sơn Thảo bằng framework Laravel với các mục tiêu chính như sau:

* Thiết kế giao diện thân thiện, dễ sử dụng, giúp khách hàng dễ dàng tìm kiếm, so sánh và đặt mua sản phẩm trực tuyến.
* Xây dựng hệ thống quản lý sản phẩm, đơn hàng, khách hàng và khuyến mãi một cách hiệu quả.
* Tích hợp giỏ hàng, thanh toán trực tuyến và chức năng theo dõi trạng thái đơn hàng.
* Ứng dụng các tính năng bảo mật của Laravel để bảo vệ thông tin người dùng và giao dịch.
* Tối ưu hiệu suất và khả năng mở rộng của hệ thống, hỗ trợ SEO để tăng khả năng tiếp cận khách hàng.
* Tận dụng ưu điểm của Laravel như cấu trúc MVC, hệ thống route linh hoạt, Eloquent ORM và các gói mở rộng để phát triển nhanh, bảo trì dễ dàng.

1. **Nội dung nghiên cứu**

Nội dung đề tài tập trung vào việc phân tích, thiết kế và triển khai hệ thống website bán điện thoại di động Sơn Thảo. Phần frontend được xây dựng bằng Laravel Blade kết hợp JQuery để tạo giao diện thân thiện, dễ sử dụng. Phần backend sử dụng Laravel Framework để xử lý logic nghiệp vụ và kết nối với cơ sở dữ liệu MySQL nhằm quản lý thông tin sản phẩm, khách hàng và đơn hàng. Ngoài ra, hệ thống sẽ áp dụng Laravel Sanctum để xác thực và bảo mật, đồng thời tích hợp các tính năng hỗ trợ bán hàng trực tuyến.

Nội dung cụ thể bao gồm:

* **Xây dựng giao diện người dùng (Frontend)**

Thiết kế và phát triển các trang: trang chủ, danh mục sản phẩm, chi tiết sản phẩm, giỏ hàng, thanh toán, liên hệ, quản lý tài khoản khách hàng. Giao diện được xây dựng bằng Laravel Blade kết hợp HTML, CSS, JQuery, đảm bảo dễ sử dụng và tương thích trên nhiều thiết bị.

* **Xây dựng chức năng xử lý nghiệp vụ (Backend)**

Áp dụng mô hình MVC của Laravel để xử lý các chức năng: tìm kiếm và lọc sản phẩm, thêm/xoá/sửa sản phẩm trong giỏ hàng, đặt hàng, quản lý thông tin khách hàng và đơn hàng, áp dụng mã giảm giá hoặc khuyến mãi.

* **Tích hợp tính năng bảo mật và quản trị**

Sử dụng cơ chế xác thực và phân quyền của Laravel để quản lý quyền truy cập. Xây dựng trang quản trị (Admin) cho phép quản lý sản phẩm, danh mục, khách hàng, đơn hàng và nội dung website.

* **Triển khai và kiểm thử**
* Cài đặt và cấu hình hệ thống trên môi trường máy chủ (hosting hoặc VPS) hoặc môi trường cục bộ
* Sử dụng Postman để gửi và kiểm tra các request đến hệ thống, bao gồm: thêm/sửa/xoá sản phẩm, đặt hàng, đăng ký/đăng nhập tài khoản, quản lý đơn hàng.
* Kiểm thử luồng xử lý dữ liệu, phản hồi của server, mã trạng thái HTTP và tính chính xác của dữ liệu trả về.
* Đánh giá mức độ bảo mật và hiệu suất của các chức năng thông qua kết quả kiểm thử bằng Postman.

1. **Phạm vi của đề tài**

Đề tài tập trung xây dựng website bán điện thoại di động Sơn Thảo với các chức năng cơ bản phục vụ hoạt động bán hàng trực tuyến và quản trị hệ thống. Phạm vi nghiên cứu và triển khai bao gồm:

* **Về chức năng người dùng**
* Xem danh mục sản phẩm và chi tiết sản phẩm.
* Tìm kiếm và lọc sản phẩm theo tiêu chí.
* Quản lý giỏ hàng và đặt hàng trực tuyến.
* Đăng ký, đăng nhập và quản lý tài khoản cá nhân.
* **Về chức năng quản trị**
* Quản lý sản phẩm, danh mục và thông tin chi tiết.
* Quản lý khách hàng và đơn hàng.
* Quản lý mã giảm giá hoặc chương trình khuyến mãi.
* Phân quyền và quản lý tài khoản quản trị.
* **Về công nghệ**
* Sử dụng Laravel Framework với mô hình MVC để xây dựng hệ thống.
* Giao diện phát triển bằng Laravel Blade, HTML, CSS, JQuery.
* Cơ sở dữ liệu quản lý bằng MySQL.
* Áp dụng bảo mật bằng cơ chế xác thực, phân quyền của Laravel.
* **Về kiểm thử**
* Chỉ tập trung kiểm thử các request và luồng xử lý bằng Postman.
* Không triển khai kiểm thử giao diện người dùng trên trình duyệt ở phạm vi này.

1. **Bố cục của đề tài**

Ngoài hai phần Mở đầu và Kết luận, báo cáo đồ án tốt nghiệp còn có bốn chương như sau:

* Chương 1: Khảo sát hệ thống
* Chương 2: Phân tích và thiết kế hệ thống
* Chương 3: Xây dựng website
* Chương 4: Kiểm thử website

**CHƯƠNG 1: KHẢO SÁT HỆ THỐNG**

* 1. **Khảo sát nghiệp vụ**
     1. **Giới thiệu chung**

Cửa hàng điện thoại Sơn Thảo là một cửa hàng chuyên kinh doanh các sản phẩm điện thoại di động, máy tính bảng và phụ kiện đi kèm. Trong nhiều năm qua, hoạt động kinh doanh của cửa hàng chủ yếu dựa vào việc bán hàng trực tiếp tại cửa hàng. Nhân viên sẽ giới thiệu sản phẩm cho khách, tư vấn và tiến hành thanh toán tại quầy.

Tuy nhiên, thói quen mua sắm của người tiêu dùng đã thay đổi đáng kể trong những năm gần đây. Ngày càng nhiều khách hàng có xu hướng tìm hiểu và đặt mua sản phẩm trực tuyến thay vì đến tận nơi. Điều này tạo ra nhu cầu cấp thiết cho cửa hàng Sơn Thảo trong việc xây dựng một website bán hàng trực tuyến vừa để quảng bá thương hiệu, vừa để mở rộng tệp khách hàng và đồng thời quản lý hoạt động kinh doanh hiệu quả hơn.

* + 1. **Quy trình nghiệp vụ hiện tại của cửa hàng**

Hiện tại, hoạt động bán hàng hiện nay tại cửa hàng Sơn Thảo được tiến hành theo các bước cơ bản sau:

1. **Tiếp nhận khách hàng:** Khách hàng đến trực tiếp các cơ sở của cửa hàng, nhân viên cửa hàng tư vấn giới thiệu sản phẩm.
2. **Khách hàng lựa chọn sản phẩm:** Dựa trên nhu cầu sử dụng và tư vấn của nhân viên, khách hàng lựa chọn sản phẩm muốn mua.
3. **Thanh toán:** Nhân viên lập hoá đơn trên hoá đơn có sẵn và khách hàng tiến hành thanh toán trực tiếp.
4. **Giao hàng:** Khách hàng nhận sản phẩm tại quầy.
5. **Lưu trữ thông tin:** Cửa hàng lưu thông tin đơn hàng, khách hàng và doanh thu dưới dạng giấy tờ hoặc file Excel.

**Những hạn chế trong quy trình hiện tại:**

* Khách hàng ở xa hoặc không có thời gian đến cửa hàng rất khó mua sản phẩm.
* Thông tin sản phẩm chưa được quảng bá rộng rãi, phụ thuộc nhiều vào việc khách hàng quen biết hoặc tự tìm đến.
* Việc quản lý hóa đơn, doanh thu và hàng tồn kho bằng Excel hoặc giấy tờ thủ công thường mất nhiều thời gian, dễ nhầm lẫn và khó thống kê chính xác.
* Khả năng cạnh tranh kém so với các cửa hàng khác đã triển khai kênh bán hàng online.
  + 1. **Quy trình nghiệp vụ dự kiến khi sử dụng hệ thống mới**

Khi áp dụng hệ thống mới, quy trình bán hàng sẽ có nhiều thay đổi so với quy trình cũ và sẽ có hai hình thức song song là mua hàng trực tuyến và trực tiếp tại cửa hàng.

* + - 1. **Khi khách hàng mua online**

1. **Khách hàng tìm kiếm và tham khảo sản phẩm:** Khách hàng truy cập website, tìm kiếm sản phẩm theo nhu cầu sử dụng.
2. **Chọn sản phẩm và đặt hàng:** Khách hàng thêm sản phẩm vào giỏ hàng, kiểm tra lại thông tin đặt hàng và chọn phương thức thanh toán.
3. **Thanh toán:**

* Nếu khách hàng chọn COD: Thanh toán khi nhận hàng.
* Nếu khách hàng chọn thanh toán online: Thanh toán ngay sau khi đặt hàng.

1. **Xác nhận đơn hàng:** Thông tin đơn hàng sẽ được gửi lên hệ thống, nhân viên cửa hàng sẽ kiểm tra, xác nhận lại đơn hàng với khách hàng thông qua số điện thoại.
2. **Xuất hoá đơn:** Nhân viên cửa hàng tiến hành tạo, xuất hoá đơn bán hàng và gửi hoá đơn về cho khách hàng thông qua email hoặc số điện thoại.
3. **Giao hàng:** Cửa hàng sẽ đóng gói đơn hàng và giao sản phẩm cho khách hàng thông qua dịch vụ vận chuyển.
4. **Lưu trữ dữ liệu:** Thông tin khách hàng và đơn hàng sẽ được lưu lại trong hệ thống, hỗ trợ tra cứu và báo cáo.
   * + 1. **Khi khách hàng mua trực tiếp tại của hàng**
5. **Tiếp nhận khách hàng:** Nhân viên tư vấn giới thiệu sản phẩm.
6. **Khách hàng tham khảo sản phẩm:** Ngoài việc xem trực tiếp tại cửa hàng, nhân viên có thể hỗ trợ tra cứu thông tin sản phẩm ngay trên hệ thống (giá bán, khuyến mãi, số lượng tồn kho).
7. **Chọn mua sản phẩm:** Khách hàng chọn sản phẩm ưng ý và quyết định mua.
8. **Tạo hoá đơn:** Khi khách quyết định mua, nhân viên lập đơn hàng trên hệ thống.
9. **Thanh toán:** Khách thanh toán tại quầy (tiền mặt, chuyển khoản hoặc quét mã QR).
10. **Xuất hoá đơn và giao sản phẩm:** Hệ thống in hoá đơn bán hàng cho khách hàng và lưu hoá đơn vào hệ thống.
11. **Lưu trữ dữ liệu:** Thông tin khách hàng và đơn hàng sẽ được lưu lại trong hệ thống, hỗ trợ tra cứu và báo cáo.
    1. **Yêu cầu của hệ thống**
       1. **Yêu cầu chức năng**

Hệ thống website bán điện thoại cần đảm bảo cung cấp các chức năng chính cho từng đối tượng sử dụng như sau:

* + - 1. **Đối với khách hàng**
* Đăng ký tài khoản, đăng nhập và quản lý thông tin cá nhân.
* Tìm kiếm và xem chi tiết sản phẩm (theo hãng, mức giá, …).
* Xem hình ảnh, thông số kỹ thuật, giá bán, khuyến mãi của sản phẩm.
* Thêm sản phẩm vào giỏ hàng và đặt hàng trực tuyến.
* Theo dõi trạng thái đơn hàng (chờ xác nhận, đang giao, đã hoàn tất).
* Thực hiện thanh toán trực tuyến hoặc chọn phương thức thanh toán khi nhận hàng (COD).
* Liên hệ với cửa hàng qua form hỗ trợ hoặc thông tin liên hệ.
  + - 1. **Đối với nhân viên cửa hàng**
* Đăng nhập vào hệ thống bằng tài khoản được phân quyền.
* Quản lý danh mục sản phẩm (thêm, sửa, xóa, cập nhật tồn kho).
* Quản lý đơn hàng: tiếp nhận đơn hàng online, tạo đơn hàng cho khách mua trực tiếp, cập nhật trạng thái đơn hàng.
* Hỗ trợ khách hàng trong quá trình đặt mua (trả lời câu hỏi, cập nhật thông tin đơn).
  + - 1. **Đối với quản trị viên**
* Quản lý toàn bộ sản phẩm, đơn hàng và khách hàng.
* Quản lý nhân viên (tạo tài khoản, phân quyền, khóa/mở tài khoản).
* Thống kê và báo cáo doanh thu theo ngày, tháng, năm.
* Quản lý thông tin website (banner, tin tức, chính sách bảo hành, liên hệ).
* Theo dõi dữ liệu truy cập để cải thiện hiệu quả kinh doanh.
  + 1. **Yêu cầu phi chức năng**

Ngoài các chức năng chính, hệ thống cần đảm bảo các yêu cầu phi chức năng sau:

* **Tính dễ sử dụng:** Giao diện thân thiện, hỗ trợ tìm kiếm nhanh, bố cục rõ ràng.
* **Tính bảo mật:** Dữ liệu người dùng được bảo vệ, mật khẩu được mã hóa, phân quyền rõ ràng.
* **Tính toàn vẹn dữ liệu:** Thông tin sản phẩm, đơn hàng, khách hàng phải được lưu trữ tập trung, tránh trùng lặp hoặc mất dữ liệu.
* **Tính ổn định và hiệu suất:** Hệ thống hoạt động liên tục, có khả năng xử lý nhiều truy cập cùng lúc.
* **Khả năng mở rộng:** Hệ thống có thể bổ sung thêm chức năng mới (như khuyến mãi, tích điểm khách hàng thân thiết) mà không ảnh hưởng tới các chức năng hiện tại.
* **Tính tương thích:** Website hiển thị tốt trên nhiều thiết bị (máy tính, máy tính bảng, điện thoại).
  1. **Các yêu cầu về dữ liệu**
     1. **Nguồn dữ liệu hiện tại**

Hiện nay, dữ liệu của cửa hàng Sơn Thảo chủ yếu được lưu trữ dưới dạng:

* File Excel quản lý sản phẩm (tên sản phẩm, mã sản phẩm, hãng, số lượng tồn kho, giá bán).
* Hóa đơn bán hàng giấy hoặc file điện tử đơn giản.
* Thông tin khách hàng (họ tên, số điện thoại, địa chỉ) được lưu rời rạc, chưa tập trung.
* Thông tin nhân viên quản lý đơn hàng, sản phẩm cũng chưa được lưu trữ trong một hệ thống thống nhất.

Do chưa có hệ thống quản lý tập trung, việc thống kê doanh thu, theo dõi đơn hàng hay tra cứu sản phẩm còn mất nhiều thời gian, dễ sai sót và khó đảm bảo tính chính xác.

* + 1. **Các loại dữ liệu cần quản lý trong hệ thống**

Dựa trên quy trình nghiệp vụ, hệ thống website sẽ cần quản lý các loại dữ liệu sau:

* **Dữ liệu sản phẩm:**
* Mã sản phẩm.
* Tên sản phẩm.
* Hãng sản xuất.
* Thông tin chi tiết.
* Giá bán.
* Số lượng tồn kho.
* Hình ảnh.
* **Dữ liệu khách hàng:**
* Mã khách hàng.
* Tên khách hàng.
* Thông tin cá nhân.
* Thông tin liên hệ.
* Thông tin đăng nhập.
* **Dữ liệu đơn hàng:**
* Mã đơn hàng.
* Thông tin khách hàng.
* Danh sách sản phẩm đặt mua.
* Thông tin chi tiết đơn hàng.
* Trạng thái đơn hàng.
* **Dữ liệu nhân viên/Quản trị viên:**
* Mã nhân viên.
* Thông tin cá nhân.
* Thông tin liên hệ.
* Thông tin đăng nhập.
* Quyền hạn trong hệ thống.
* **Dữ liệu hệ thống:**
* Banner, tin tức, chính sách bán hàng, chính sách bảo hành.
* Nhật ký hoạt động (log) của hệ thống để phục vụ quản lý.
  1. **Kết luận chương 1**

Qua quá trình khảo sát, có thể thấy rằng hoạt động kinh doanh hiện tại của cửa hàng điện thoại Sơn Thảo còn gặp nhiều hạn chế: việc quản lý sản phẩm, đơn hàng và khách hàng chủ yếu thực hiện bằng phương pháp thủ công hoặc thông qua các file Excel rời rạc; khách hàng chủ yếu đến trực tiếp tại cửa hàng để mua sản phẩm nên phạm vi tiếp cận còn hạn chế; khả năng thống kê và báo cáo chưa thực sự hiệu quả.

Từ thực trạng đó, hệ thống website bán điện thoại cho cửa hàng Sơn Thảo được định hướng xây dựng nhằm giải quyết đồng thời hai hình thức bán hàng:

* **Bán hàng trực tuyến (online):** khách hàng có thể truy cập website để tìm kiếm sản phẩm, đặt hàng và thanh toán thuận tiện.
* **Bán hàng trực tiếp (offline):** nhân viên có thể sử dụng hệ thống để lập đơn hàng, quản lý kho và theo dõi dữ liệu một cách tập trung.

Để triển khai hệ thống, đề tài lựa chọn Laravel Framework – một nền tảng PHP mạnh mẽ, hỗ trợ mô hình MVC, có khả năng mở rộng, bảo mật tốt và dễ dàng tích hợp với các công nghệ hiện đại khác. Laravel đồng thời cung cấp nhiều công cụ hữu ích cho việc xây dựng website thương mại điện tử, như hệ thống quản lý người dùng, quản lý cơ sở dữ liệu (Eloquent ORM), hỗ trợ API, và cơ chế bảo mật (middleware, CSRF token).

Bên cạnh đó, chương này cũng đã làm rõ:

* **Khảo sát nghiệp vụ:** mô tả quy trình kinh doanh hiện tại và quy trình mới khi áp dụng hệ thống.
* **Yêu cầu hệ thống:** xác định các chức năng chính cho khách hàng, nhân viên và quản trị viên; các yêu cầu phi chức năng về bảo mật, hiệu năng và tính mở rộng.
* **Khảo sát dữ liệu:** phân tích các loại dữ liệu cần quản lý (sản phẩm, khách hàng, đơn hàng, nhân viên) và minh họa bằng sơ đồ luồng dữ liệu mức 0.

Những kết quả trên sẽ là cơ sở quan trọng để sang chương 2 – Phân tích và thiết kế hệ thống, trong đó các nghiệp vụ và dữ liệu đã khảo sát sẽ được mô hình hóa bằng các sơ đồ UML (Use Case, Class, Sequence, Activity, v.v.) cũng như mô hình dữ liệu để phục vụ cho quá trình cài đặt bằng Laravel.

**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**2.1. Giới thiệu chung**

Trong quá trình phát triển phần mềm, giai đoạn phân tích và thiết kế hệ thống giữ vai trò đặc biệt quan trọng. Đây là bước trung gian giữa khảo sát yêu cầu (Chương 1) và giai đoạn xây dựng – triển khai (Chương 3). Nếu như chương 1 tập trung tìm hiểu nghiệp vụ, yêu cầu chức năng và phi chức năng, thì chương 2 nhằm chuyển hóa những yêu cầu đó thành mô hình UML cụ thể, dễ hiểu, giúp lập trình viên và nhóm phát triển hiện thực hóa trong quá trình cài đặt.

**UML (Unified Modeling Language)** – Ngôn ngữ mô hình hóa thống nhất – được chọn để mô tả hệ thống. UML cung cấp nhiều loại biểu đồ khác nhau, mỗi loại phản ánh một khía cạnh:

* **Use Case Diagram (Biểu đồ Use Case):** cho biết hệ thống có những chức năng gì, ai sẽ sử dụng những chức năng đó.
* **Class Diagram (Biểu đồ lớp):** mô tả cấu trúc dữ liệu và quan hệ giữa các đối tượng.
* **Sequence Diagram (Biểu đồ tuần tự) và Collaboration Dìgram (Biểu đồ hợp tác):** mô tả luồng tương tác giữa các thành phần trong các kịch bản cụ thể.
* **State Diagram (Biểu đồ trạng thái) và Activity Diagram (Biểu đồ hoạt động):** biểu diễn trạng thái và hành vi hoạt động của hệ thống.
* **Component Diagram (Biểu đồ thành phần) và Deployment Diagram (Biểu đồ triển khai):** cho biết cách tổ chức thành phần và triển khai hệ thống.

Việc sử dụng UML trong phân tích và thiết kế hệ thống mang lại một số ưu điểm như sau:

* **Tính thống nhất:** hệ thống hoá, tổ chức lại các yêu cầu và quy trình thành mô hình logic, dễ theo dõi.
* **Tính trực quan:** các biểu đồ giúp người triển khai dễ dàng hình dung và mô tả lại hệ thống, tránh tình trạng mơ hồ khi triển khai.
* **Khả năng mở rộng:** khi có thay đổi về yêu cầu, các mô hình UML có thể được điều chỉnh nhanh chóng và đảm bảo tính linh hoạt của hệ thống.
* **Cơ sở hiện thực hoá:** từ các biểu đồ UML có thể dễ dàng chuyển đổi sang thiết kế cơ sở dữ liệu và triển khai mã nguồn.

Trong chương này, hệ thống website bán điện thoại Sơn Thảo sẽ được phân tích và thiết kế chi tiết thông qua 8 loại biểu đồ UML. Kết quả phân tích sẽ là nền tảng để xây dựng website ở Chương 3 và kiểm thử ở Chương 4.

**2.2. Biểu đồ Use Case**

Sau khi xác định các yêu cầu chức năng và phi chức năng của hệ thống ở Chương 1, bước tiếp theo là xây dựng biểu đồ Use Case nhằm mô tả tổng quan các chức năng mà hệ thống cần đáp ứng cũng như mối quan hệ với các tác nhân bên ngoài. Việc xây dựng biểu đồ Use Case giúp làm rõ phạm vi hoạt động của hệ thống trước khi tiến hành thiết kế chi tiết.

**2.2.1. Các tác nhân trong hệ thống**

Từ kết quả khảo sát, các tác nhân chính trong hệ thống được xác định như sau:

* **Khách hàng vãng lai:** Người truy cập website nhưng chưa đăng nhập. Nhóm tác nhân này chỉ có thể xem danh sách sản phẩm, tìm kiếm, lọc thông tin và gửi yêu cầu liên hệ.
* **Khách hàng đã đăng nhập:** Người dùng đã đăng ký và đăng nhập vào hệ thống. Ngoài các chức năng của khách vãng lai, khách hàng có thể thêm sản phẩm vào giỏ hàng, đặt hàng, thanh toán, áp dụng mã giảm giá và theo dõi trạng thái đơn hàng.
* **Nhân viên:** Người phụ trách quản lý sản phẩm và xử lý đơn hàng. Nhân viên có thể xác nhận đơn, cập nhật trạng thái giao hàng và hỗ trợ khách hàng khi cần thiết.
* **Quản trị viên:** Người có quyền quản lý cao nhất trong hệ thống. Ngoài việc thực hiện các chức năng của nhân viên, quản trị viên còn có thể phân quyền, quản lý nội dung website (banner, tin tức, chính sách) và xem các báo cáo thống kê doanh thu.
* **Hệ thống bên ngoài:** Bao gồm các dịch vụ tích hợp như cổng thanh toán trực tuyến, dịch vụ vận chuyển và dịch vụ Email/SMS được sử dụng để gửi thông báo cho khách hàng.

**2.2.2. Các use case chính**

Dựa trên các tác nhân đã xác định, hệ thống có các nhóm chức năng chính như sau:

* **Đối với khách hàng vãng lai:**
* Đăng ký tài khoản.
* Xem danh mục sản phẩm, xem chi tiết sản phẩm.
* Tìm kiếm và lọc sản phẩm.
* Liên hệ hỗ trợ.
* **Đối với khách hàng đã đăng nhập:**
* Đăng nhập, đăng xuất.
* Xem danh mục sản phẩm, xem chi tiết sản phẩm.
* Tìm kiếm và lọc sản phẩm.
* Quản lý giỏ hàng (thêm, sửa, xoá sản phẩm).
* Mua hàng và thanh toán.
* Đánh giá sản phẩm.
* Theo dõi trạng thái đơn hàng.
* Xem và chỉnh sửa thông tin cá nhân.
* Xem lịch sử mua hàng.
* Liên hệ hỗ trợ.
* **Đối với nhân viên:**

Quản lý sản phẩm (thêm, sửa, xóa, cập nhật tồn kho).

* Quản lý đơn hàng (xác nhận, cập nhật trạng thái, hoàn tất hoặc hủy đơn).
* Hỗ trợ khách hàng trong quá trình đặt mua**.**
* Đăng nhập, đăng xuất**.**
* **Đối với quản trị viên:**
* Quản lý nhân viên, phân quyền sử dụng hệ thống.
* Quản lý thông tin hiển thị trên website (banner, tin tức, chính sách).
* Quản lý khuyến mãi và mã giảm giá.
* Theo dõi báo cáo doanh thu, thống kê đơn hàng.
* Giám sát toàn bộ hoạt động của hệ thống.
* Đăng nhập, đăng xuất.
* **Đối với các hệ thống bên ngoài**:
* Cổng thanh toán: xử lý giao dịch trực tuyến.
* Dịch vụ Email/SMS: gửi thông báo xác nhận đơn hàng, trạng thái giao hàng và hỗ trợ bảo mật (OTP, đổi mật khẩu).

**2.2.3. Mô tả chi tiết một số use case chính**

***2.2.3.1. Mô tả use case Xem sản phẩm theo danh mục***

**Mô tả vắn tắt:** Use case này cho phép người dùng xem danh sách sản phẩm thuộc một danh mục cụ thể trên website.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Người dùng chọn một danh mục sản phẩm từ menu danh mục.
2. Hệ thống hiển thị danh sách các danh mục sản phẩm.
3. Người dùng chọn danh mục sản phẩm muốn xem.
4. Hệ thống hiển thị danh sách các sản phẩm theo danh mục đã chọn.

* Luồng rẽ nhánh:

1. Tại luồng cơ bản ở bước 3, nếu danh mục không có sản phẩm nào, hệ thống sẽ hiển thị thông báo “Không có sản phẩm trong danh mục này”.

**Tiền điều kiện:**

* Hệ thống đã có dữ liệu danh mục sản phẩm.
* Người dùng có thể truy cập website mà không cần đăng nhập.

**Hậu điều kiện:**

* Danh sách sản phẩm trong danh mục được hiển thị cho người dùng.
* Người dùng có thể điều hướng sang trang chi tiết sản phẩm khi chọn vào một sản phẩm cụ thể.

**Các yêu cầu đặc biệt:**

* Hệ thống cần hỗ trợ phân trang để tránh tải quá nhiều dữ liệu cùng một lúc.
* Tốc độ hiển thị danh sách sản phẩm phải đảm bảo trong thời gian ngắn.

**Điểm mở rộng:**

* Tìm kiếm, lọc sản phẩm trong phạm vi danh mục.

***2.2.3.2. Mô tả use case Xem chi tiết sản phẩm***

**Mô tả vắn tắt:** Use case này cho phép người dùng xem thông tin chi tiết của một sản phẩm, bao gồm hình ảnh, giá, mô tả, tình trạng và các thuộc tính liên quan.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Người dùng chọn một sản phẩm từ danh sách sản phẩm hoặc kết quả tìm kiếm.
2. Hệ thống hiển thị trang chi tiết sản phẩm với đầy đủ thông tin.
3. Người dùng có thể tiếp tục thao tác như thêm sản phẩm vào giỏ hàng, quay lại danh sách sản phẩm hoặc xem các sản phẩm liên quan.

* Luồng rẽ nhánh:

1. Tại luồng cơ bản ở bước 3, nếu người dùng chưa đăng nhập thì khi chọn vào các chức năng mua hàng hoặc thêm sản phẩm vào giỏ hàng thì sẽ báo lỗi “Vui lòng đăng nhập trước khi thực hiện thao tác này”.

**Tiền điều kiện:**

* Danh sách sản phẩm và dữ liệu chi tiết sản phẩm đã được lưu trong hệ thống.
* Người dùng có thể truy cập website bằng trình duyệt web, không bắt buộc đăng nhập.

**Hậu điều kiện:**

* Trang chi tiết sản phẩm được hiển thị đầy đủ và chính xác cho người dùng.
* Người dùng có thể lựa chọn các hành động tiếp theo như thêm sản phẩm vào giỏ hàng, quay lại danh sách hoặc xem sản phẩm liên quan.

**Các yêu cầu đặc biệt:**

* Trang chi tiết phải hiển thị nhanh để đảm bảo trải nghiệm người dùng.
* Hình ảnh sản phẩm cần có chất lượng cao, hỗ trợ xem nhiều góc độ.
* Nội dung chi tiết phải đồng bộ với thông tin trong kho hàng (giá cả, số lượng tồn).
* Hệ thống cần đảm bảo tính bảo mật khi hiển thị các dữ liệu động (tránh lỗi SQL Injection khi truy vấn theo ID sản phẩm).

**Điểm mở rộng:**

* Cho phép khách hàng đã đăng nhập viết đánh giá hoặc bình luận về sản phẩm.
* Hiển thị gợi ý “Mua kèm” hoặc “Sản phẩm tương tự” để tăng khả năng bán chéo.

***2.2.3.3. Mô tả use case Tìm kiếm sản phẩm***

**Mô tả vắn tắt:** Use case này cho phép người dùng tìm kiếm các sản phẩm trên hệ thống bằng các từ khoá liên quan.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Người dùng truy cập vào ô tìm kiếm trên website.
2. Người dùng nhập từ khóa (ví dụ: tên sản phẩm, mã sản phẩm, thương hiệu) và nhấn nút “Tìm kiếm”.
3. Hệ thống tiếp nhận từ khóa, thực hiện truy vấn dữ liệu sản phẩm trong cơ sở dữ liệu.
4. Hệ thống hiển thị danh sách kết quả phù hợp với từ khóa đã nhập, kèm theo các thông tin cơ bản: tên, giá, hình ảnh, tình trạng hàng.
5. Người dùng có thể:

* Lọc thêm theo các tiêu chí nâng cao (giá, thương hiệu, khuyến mãi...).
* Chọn một sản phẩm trong danh sách để xem chi tiết**.**
* Luồng rẽ nhánh:

1. Tại luồng cơ bản ở bước 2, nếu người dùng không nhập từ khóa mà nhấn tìm kiếm, hệ thống hiển thị thông báo “Vui lòng nhập từ khóa tìm kiếm” hoặc trả về danh sách sản phẩm nổi bật.
2. a, Tại luồng cơ bản ở bước 4, nếu từ khóa không trùng khớp với bất kỳ sản phẩm nào, hệ thống hiển thị thông báo “Không tìm thấy sản phẩm phù hợp”.

b, Nếu có nhiều kết quả, hệ thống hiển thị theo dạng phân trang để tránh quá tải dữ liệu.

**Tiền điều kiện:**

* Hệ thống đã có dữ liệu sản phẩm đầy đủ trong cơ sở dữ liệu.
* Người dùng có thể truy cập website mà không bắt buộc đăng nhập.

**Hậu điều kiện:**

* Kết quả tìm kiếm được hiển thị cho người dùng.
* Người dùng có thể chọn sản phẩm để xem chi tiết hoặc tiếp tục tìm kiếm/lọc thêm**.**

**Các yêu cầu đặc biệt:**

* Tìm kiếm phải hoạt động nhanh để đảm bảo trải nghiệm người dùng.
* Kết quả tìm kiếm phải được sắp xếp theo mức độ liên quan hoặc các tiêu chí phổ biến (bán chạy, giá, khuyến mãi).
* Hỗ trợ tìm kiếm không dấu và gần đúng (ví dụ: “dien thoai” vẫn ra “điện thoại”).

**Điểm mở rộng:**

* Gợi ý sản phẩm tương tự hoặc sản phẩm bán chạy liên quan đến từ khóa.
* Cho phép người dùng lưu lại lịch sử tìm kiếm.
* Gợi ý tìm kiếm dựa trên hành vi và lịch sử duyệt sản phẩm của người dùng**.**

***2.2.3.4. Mô tả use case Đặt hàng***

**Mô tả vắn tắt:** Use case này cho phép người dùng đã đăng nhập vào hệ thống tiến hành đặt mua sản phẩm sau khi chọn và thêm sản phẩm vào giỏ hàng.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Khách hảng thêm sản phẩm vào giỏ hàng hoặc trực tiếp truy cập vào giỏ hàng.
2. Hệ thống hiển thị danh sách sản phẩm trong giỏ cùng số lượng, đơn giá và tổng giá trị đơn hàng.
3. Khách hàng xác nhận sản phẩm muốn mua, có thể chỉnh sửa số lượng hoặc loại bỏ sản phẩm khỏi giỏ.
4. Khách hàng nhấn nút “Mua ngay”.
5. Hệ thống yêu cầu khách hàng cung cấp thông tin giao hàng (họ tên, số điện thoại, địa chỉ).
6. Khách hàng chọn phương thức thanh toán (COD hoặc thanh toán trực tuyến qua cổng thanh toán).
7. Hệ thống hiển thị màn hình xác nhận đơn hàng, bao gồm: danh sách sản phẩm, phí vận chuyển, khuyến mãi/giảm giá (nếu có), tổng số tiền phải thanh toán.
8. Khách hàng xác nhận đặt hàng.
9. Hệ thống ghi nhận thông tin đơn hàng, tạo mã đơn hàng và lưu trữ vào cơ sở dữ liệu.
10. Hệ thống gửi thông báo xác nhận đơn hàng cho khách hàng qua Email/SMS.

* Luồng rẽ nhánh:

1. Tại luồng cơ bản ở bước 2, nếu giỏ hàng trống, hệ thống hiển thị thông báo “Giỏ hàng của bạn đang trống” và không cho phép tiếp tục đặt hàng.
2. Tại luồng cơ bản ở bước 5, nếu khách hàng nhập thiếu thông tin giao hàng, hệ thống yêu cầu bổ sung trước khi tiếp tục.
3. Tại luồng cơ bản ở bước 6, nếu chọn thanh toán trực tuyến nhưng giao dịch thất bại (lỗi mạng, thẻ không hợp lệ…), hệ thống hiển thị thông báo lỗi và cho phép khách hàng chọn lại phương thức thanh toán khác hoặc thử lại.
4. Tại luồng cơ bản ở bước 8, nếu khách hàng hủy thao tác trước khi xác nhận, hệ thống không lưu đơn hàng và quay lại giỏ hàng.

**Tiền điều kiện:**

* Khách hàng đã đăng ký và đăng nhập vào hệ thống.
* Giỏ hàng có ít nhất một sản phẩm hợp lệ.

**Hậu điều kiện:**

* Đơn hàng được tạo và lưu vào hệ thống với trạng thái “Đang chờ xác nhận”.
* Khách hàng nhận được thông báo xác nhận đơn hàng.

**Các yêu cầu đặc biệt:**

* Hệ thống phải đảm bảo tính chính xác của dữ liệu đơn hàng (giá sản phẩm, số lượng, giảm giá, phí vận chuyển).
* Quá trình tạo đơn hàng cần diễn ra nhanh chóng, tránh lỗi mất dữ liệu khi kết nối không ổn định.
* Hệ thống cần có cơ chế kiểm tra số lượng tồn kho để tránh trường hợp đặt vượt quá số lượng hiện có.
* Thông tin khách hàng phải được mã hóa và bảo mật theo tiêu chuẩn an toàn dữ liệu.

**Điểm mở rộng:**

* Cho phép khách hàng sử dụng mã giảm giá hoặc voucher khi đặt hàng.
* Cho phép khách hàng chọn nhiều địa chỉ giao hàng đã lưu sẵn trong tài khoản.
* Tích hợp tính năng theo dõi trạng thái đơn hàng ngay sau khi đặt thành công.

***2.2.3.5. Mô tả use case Thanh toán***

**Mô tả vắn tắt:** Use case này cho phép khách hàng hoàn tất việc thanh toán đơn hàng đã đặt thông qua các phương thức hỗ trợ như COD (thanh toán khi nhận hàng) hoặc thanh toán trực tuyến (cổng thanh toán điện tử).

**Luồng sự kiện:**

* Luồng cơ bản (COD):

1. Sau khi tạo đơn hàng, khách hàng chọn phương thức thanh toán “Thanh toán khi nhận hàng (COD)”.
2. Hệ thống hiển thị thông tin đơn hàng: sản phẩm, tổng giá trị, phí vận chuyển.
3. Khách hàng xác nhận lựa chọn.
4. Hệ thống ghi nhận hình thức thanh toán là COD và cập nhật trạng thái đơn hàng thành “Chờ xử lý”.
5. Hệ thống gửi thông báo xác nhận đơn hàng cho khách hàng qua Email/SMS.

* Luồng cơ bản (Thanh toán trực tuyến):

1. Sau khi tạo đơn hàng, khách hàng chọn phương thức thanh toán “Trực tuyến”.
2. Hệ thống chuyển hướng khách hàng đến giao diện cổng thanh toán (VNPay, MoMo, PayPal…).
3. Khách hàng nhập thông tin thanh toán (số thẻ, mã OTP, ví điện tử…).
4. Cổng thanh toán xử lý giao dịch và trả kết quả cho hệ thống.
5. Nếu giao dịch thành công:

* Hệ thống cập nhật trạng thái đơn hàng thành “Đã thanh toán, chờ xử lý”.
* Hệ thống gửi thông báo xác nhận thành công qua Email/SMS.

1. Nếu giao dịch thất bại:

* Hệ thống thông báo lỗi cho khách hàng (ví dụ: thẻ không hợp lệ, số dư không đủ, kết nối lỗi).
* Cho phép khách hàng thử lại hoặc chọn phương thức thanh toán khác.
* Luồng rẽ nhánh:

1. Tại luồng cơ bản (Thanh toán trực tuyến) ở bước 2, nếu cổng thanh toán không phản hồi trong một khoảng thời gian nhất định, hệ thống hiển thị thông báo “Hệ thống bận, vui lòng thử lại sau”.
2. Tại luồng cơ bản (Thanh toán trực tuyến) ở bước 3, nếu khách hàng hủy giao dịch tại cổng thanh toán, hệ thống trả đơn hàng về trạng thái “Chờ thanh toán” và thông báo cho khách hàng.
3. Tại luồng cơ bản (Thanh toán trực tuyến) ở bước 4, nếu có sự cố mất kết nối giữa hệ thống và cổng thanh toán, trạng thái đơn hàng được giữ ở mức “Chưa xác định”, yêu cầu kiểm tra thủ công bởi nhân viên.

**Tiền điều kiện:**

* Khách hàng đã đăng nhập và có ít nhất một đơn hàng đang chờ thanh toán.
* Hệ thống đã tích hợp sẵn các cổng thanh toán trực tuyến.

**Hậu điều kiện:**

* Trường hợp COD: đơn hàng được lưu với trạng thái “Chờ xử lý” và chờ giao hàng.
* Trường hợp thanh toán trực tuyến:\
* Thành công → đơn hàng được lưu với trạng thái “Đã thanh toán”.
* Thất bại → đơn hàng vẫn giữ trạng thái “Chờ thanh toán”.

**Các yêu cầu đặc biệt:**

* Dữ liệu giao dịch phải được mã hóa, tuân thủ tiêu chuẩn bảo mật (SSL/TLS, PCI DSS).
* Thời gian xử lý giao dịch nhanh để tránh gây gián đoạn trải nghiệm.
* Hệ thống cần ghi log chi tiết mọi giao dịch để phục vụ kiểm tra và xử lý sự cố.
* Các trường hợp thanh toán lỗi cần có cơ chế hoàn tiền tự động hoặc thủ công.

**Điểm mở rộng:**

* Hỗ trợ thêm nhiều phương thức thanh toán khác (QR code, Internet Banking, ví điện tử).
* Tích hợp tính năng lưu phương thức thanh toán ưa thích cho lần sử dụng sau.
* Gửi hóa đơn điện tử cho khách hàng sau khi thanh toán thành công.

***2.2.3.6. Mô tả use case Xem lịch sử mua hàng***

**Mô tả vắn tắt:** Use case này cho phép khách hàng đã đăng nhập tra cứu danh sách các đơn hàng mà họ đã đặt trên hệ thống.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Khách hàng đăng nhập vào hệ thống.
2. Từ trang tài khoản cá nhân, khách hàng chọn chức năng “Lịch sử mua hàng”.
3. Hệ thống truy vấn dữ liệu các đơn hàng gắn với tài khoản khách hàng.
4. Hệ thống hiển thị danh sách đơn hàng theo thứ tự thời gian (gần nhất → xa nhất), kèm các thông tin cơ bản: mã đơn hàng, ngày đặt, tổng tiền, trạng thái.
5. Khách hàng có thể chọn một đơn hàng cụ thể để xem chi tiết (sản phẩm, số lượng, địa chỉ giao hàng, phương thức thanh toán, phí vận chuyển, khuyến mãi áp dụng).

* Luồng rẽ nhánh:

1. Tại luồng cơ bản ở bước 3, nếu tài khoản khách hàng chưa từng đặt hàng, hệ thống hiển thị thông báo: “Bạn chưa có đơn hàng nào”.
2. Tại luồng cơ bản ở bước 4, nếu dữ liệu đơn hàng bị lỗi hoặc không thể truy xuất, hệ thống hiển thị thông báo: “Không thể tải dữ liệu, vui lòng thử lại sau”.
3. Tại luồng cơ bản ở bước 5, nếu đơn hàng đã bị hủy, hệ thống hiển thị chi tiết kèm trạng thái “Đã hủy” cùng lý do hủy.

**Tiền điều kiện:**

* Khách hàng đã có tài khoản và đăng nhập vào hệ thống.
* Cơ sở dữ liệu lưu trữ thông tin đơn hàng đầy đủ và chính xác.

**Hậu điều kiện:**

* Danh sách đơn hàng gắn với tài khoản khách hàng được hiển thị.
* Khách hàng có thể lựa chọn xem chi tiết từng đơn hàng hoặc quay lại trang cá nhân.

**Các yêu cầu đặc biệt:**

* Dữ liệu đơn hàng phải được hiển thị chính xác, đảm bảo tính toàn vẹn thông tin.
* Giao diện cần hỗ trợ phân trang nếu khách hàng có số lượng đơn hàng lớn.
* Thông tin nhạy cảm (ví dụ địa chỉ, số điện thoại, phương thức thanh toán) chỉ hiển thị cho chính chủ tài khoản.

**Điểm mở rộng:**

* Cho phép khách hàng tải hóa đơn hoặc biên lai điện tử cho từng đơn hàng.
* Cung cấp bộ lọc tìm kiếm theo trạng thái đơn hàng (đang giao, đã giao, đã hủy).
* Cho phép khách hàng đặt lại (Re-order) các sản phẩm từ đơn hàng cũ.

***2.2.3.7. Mô tả use case Đăng ký***

**Mô tả vắn tắt:** Use case này cho phép người dùng đăng ký tài khoản mới trên hệ thống**.**

**Luồng sự kiện:**

* Luồng cơ bản:

1. Người dùng truy cập vào trang đăng ký tài khoản.
2. Hệ thống hiển thị biểu mẫu đăng ký với các trường thông tin:

* Họ và tên.
* Ngày sinh.
* Số điện thoại.
* Email.
* Mật khẩu và xác nhận mật khẩu.
* Đăng ký nhận tin khuyến mãi.

1. Người dùng nhập đầy đủ thông tin và nhấn nút “Đăng ký”.
2. Hệ thống kiểm tra tính hợp lệ của dữ liệu (định dạng email, độ dài mật khẩu, số điện thoại hợp lệ…).
3. Nếu hợp lệ, hệ thống lưu thông tin vào cơ sở dữ liệu và tạo tài khoản mới.
4. Người dùng nhận thông báo đăng ký thành công và có thể đăng nhập để sử dụng các chức năng nâng cao.

* Luồng rẽ nhánh:

1. Tại luồng cơ bản ở bước 3, nếu người dùng nhập thiếu thông tin bắt buộc, hệ thống thông báo lỗi và yêu cầu bổ sung.
2. Tại luồng cơ bản ở bước 4, nếu số điện thoại đã tồn tại trong hệ thống, hệ thống hiển thị thông báo: “Số điện thoại đã được sử dụng, vui lòng sử dụng số điện thoại khác để đăng ký”.
3. Tại luồng cơ bản ở bước 4, nếu mật khẩu và xác nhận mật khẩu không trùng khớp, hệ thống thông báo lỗi.

**Tiền điều kiện:**

* Người dùng chưa có tài khoản hoặc muốn tạo tài khoản mới trong hệ thống.
* Hệ thống hoạt động bình thường và có kết nối đến dịch vụ Email/SMS.

**Hậu điều kiện:**

* Tài khoản khách hàng mới được tạo trong hệ thống.
* Người dùng có thể đăng nhập bằng thông tin đã đăng ký.

**Các yêu cầu đặc biệt:**

* Mật khẩu phải đáp ứng yêu cầu bảo mật (tối thiểu 6 ký tự).
* Thông tin cá nhân phải được lưu trữ và xử lý theo quy định bảo mật dữ liệu.

**Điểm mở rộng:**

* Cho phép người dùng đăng ký nhanh bằng tài khoản Google/Zalo.

***2.2.3.8. Mô tả use case Đăng nhập***

**Mô tả vắn tắt:** Use case này cho phép người dùng đăng nhập vào hệ thống bằng tài khoản đã đăng ký.

**Luồng sự kiện:**

* Luồng cơ bản:

1. Người dùng truy cập trang đăng nhập.
2. Hệ thống hiển thị biểu mẫu đăng nhập gồm:

* Số điện thoại.
* Mật khẩu.

1. Người dùng nhập thông tin và nhấn nút “Đăng nhập”.
2. Hệ thống kiểm tra tính hợp lệ của dữ liệu (không bỏ trống, định dạng số điện thoại hợp lệ).
3. Hệ thống so khớp thông tin đăng nhập với dữ liệu trong cơ sở dữ liệu.
4. Hệ thống cho phép truy cập vào giao diện và chức năng tương ứng.

* Luồng rẽ nhánh:

1. Tại luồng cơ bản ở bước 3, nếu người dùng để trống trường bắt buộc, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.
2. Tại luồng cơ bản ở bước 5, nếu số điện thoại không tồn tại trên hệ thống, hệ thống hiển thị thông báo: *“Tài khoản không tồn tại”*.
3. Tại luồng cơ bản ở bước 5, nếu mật khẩu không đúng, hệ thống hiển thị thông báo: *“Mật khẩu không chính xác”*.

**Tiền điều kiện:**

* Người dùng đã có tài khoản hợp lệ trong hệ thống.
* Hệ thống đang hoạt động và có kết nối với dịch vụ Email/SMS (nếu sử dụng OTP).

**Hậu điều kiện:**

* Người dùng được xác thực và có thể truy cập vào hệ thống với quyền hạn phù hợp.
* Thông tin phiên đăng nhập (session/token) được lưu để duy trì trạng thái đăng nhập.

**Các yêu cầu đặc biệt:**

* Mật khẩu phải được mã hóa khi lưu trữ và khi truyền tải (HTTPS, hash + salt).
* Giao diện đăng nhập phải thân thiện, hỗ trợ đa nền tảng (web, mobile).

**Điểm mở rộng:**

* Cho phép đăng nhập nhanh bằng tài khoản Google/Zalo.

**2.2.4. Ý nghĩa của biểu đồ use case**

Thông qua việc xây dựng biểu đồ Use Case, các chức năng mà hệ thống cần đáp ứng cùng với phạm vi hoạt động của từng tác nhân đã được xác định rõ ràng. Đây là cơ sở quan trọng để tiếp tục triển khai các bước phân tích dữ liệu, thiết kế lớp và thiết kế quy trình xử lý trong các phần tiếp theo.

**2.3. Biểu đồ lớp (Class Diagram)**

Biểu đồ lớp là một trong những công cụ quan trọng của UML, dùng để mô tả cấu trúc tĩnh của hệ thống thông qua các lớp đối tượng, thuộc tính, phương thức và mối quan hệ giữa các lớp. Biểu đồ lớp đóng vai trò làm cầu nối giữa phân tích nghiệp vụ và thiết kế cơ sở dữ liệu, đồng thời cũng là nền tảng cho việc hiện thực hóa các chức năng trong lập trình.

**2.3.1. Các lớp chính trong hệ thống**

Dựa trên kết quả khảo sát và phân tích yêu cầu, hệ thống website bán điện thoại di động bao gồm các lớp chính sau:

* **User:** quản lý thông tin người dùng, bao gồm tài khoản, mật khẩu, email, số điện thoại. Đây là lớp cha chung cho các loại người dùng.
* **Customer:** kế thừa từ lớp User, bổ sung các thông tin về khách hàng như địa chỉ giao hàng, lịch sử mua hàng.
* **Staff:** kế thừa từ lớp User, đại diện cho nhân viên bán hàng, có thêm quyền xử lý đơn hàng và quản lý sản phẩm.
* **Admin:** cũng kế thừa từ lớp User, có toàn quyền quản lý hệ thống, bao gồm phân quyền tài khoản, quản lý nhân viên và giám sát hoạt động.
* **Category:** lớp danh mục sản phẩm, bao gồm tên danh mục, mô tả, và danh sách sản phẩm thuộc danh mục đó.
* **Product:** lớp sản phẩm, chứa các thông tin chi tiết như mã sản phẩm, tên, giá, số lượng tồn kho, hãng sản xuất, hình ảnh minh họa.
* **Cart:** lớp giỏ hàng, lưu trữ danh sách sản phẩm mà khách hàng chọn trước khi đặt hàng.
* **Order:** lớp đơn hàng, bao gồm mã đơn, ngày đặt, trạng thái, thông tin khách hàng, tổng tiền.
* **OrderItem:** chi tiết của đơn hàng, liên kết giữa đơn hàng và sản phẩm, lưu thông tin số lượng, đơn giá của từng sản phẩm trong đơn.
* **Promotion:** lớp khuyến mãi hoặc mã giảm giá, bao gồm điều kiện áp dụng, ngày hiệu lực, giá trị giảm.

**2.3.2. Quan hệ giữa các lớp**

Các lớp trong hệ thống có quan hệ mật thiết với nhau, cụ thể:

* Lớp Customer có quan hệ 1–n với Order (một khách hàng có thể có nhiều đơn hàng).
* Lớp Order có quan hệ 1–n với OrderItem và mỗi OrderItem liên kết đến một Product.
* Lớp Category có quan hệ 1–n với Product (mỗi danh mục chứa nhiều sản phẩm).
* Lớp Order có thể liên kết với Coupon nếu khách hàng áp dụng mã giảm giá.
* Các lớp Staff và Admin kế thừa từ User, đồng thời có quan hệ với Order và Product thông qua các chức năng quản lý.

**2.3.3. Ý nghĩa của biểu đồ lớp**

Biểu đồ lớp thể hiện rõ ràng cấu trúc của hệ thống và các mối quan hệ giữa dữ liệu. Đây là cơ sở quan trọng để:

* Thiết kế cơ sở dữ liệu quan hệ (mapping các lớp sang bảng trong MySQL).
* Hiện thực hóa các lớp trong lập trình hướng đối tượng (OOP) với Laravel.
* Đảm bảo tính thống nhất, tránh trùng lặp dữ liệu và giúp dễ dàng mở rộng trong tương lai.

**2.4. Biểu đồ tuần tự (Sequence Diagram)**

Biểu đồ tuần tự được sử dụng để mô tả luồng tương tác giữa các đối tượng trong hệ thống theo trình tự thời gian. Thông qua biểu đồ này có thể hình dung được cách các thành phần phối hợp với nhau để thực hiện một chức năng cụ thể.

Trong hệ thống website bán điện thoại di động Sơn Thảo, có nhiều kịch bản nghiệp vụ quan trọng. Dưới đây là ba kịch bản tiêu biểu được mô hình hóa bằng biểu đồ tuần tự.

**2.4.1. Kịch bản *Đặt hàng trực tuyến (COD hoặc Online)***

* Các đối tượng tham gia: Khách hàng, Giao diện Website, Giỏ hàng, Đơn hàng, Thanh toán, Email/SMS.
* Trình tự tương tác:

1. Khách hàng đăng nhập và thêm sản phẩm vào giỏ hàng.

2. Hệ thống hiển thị giỏ hàng, cho phép khách cập nhật số lượng, áp dụng mã giảm giá.

3. Khách xác nhận đặt hàng, nhập thông tin giao hàng.

4. Hệ thống tạo đơn hàng mới ở trạng thái “Chờ xác nhận”.

5. Nếu chọn COD → đơn hàng được lưu và thông báo xác nhận gửi qua Email/SMS.

6. Nếu chọn thanh toán Online → hệ thống gọi đến cổng thanh toán, chờ phản hồi rồi mới cập nhật trạng thái đơn.

**2.4.2. Kịch bản *Thanh toán trực tuyến***

* Các đối tượng tham gia: Khách hàng, Giao diện Website, Module Thanh toán, Cổng thanh toán.
* Trình tự tương tác:

1. Sau khi xác nhận đơn hàng, khách chọn phương thức thanh toán trực tuyến.

2. Hệ thống tạo yêu cầu thanh toán (số tiền, mã đơn, checksum).

3. Khách được chuyển hướng sang cổng thanh toán.

4. Khách nhập thông tin và thực hiện giao dịch.

5. Cổng thanh toán phản hồi kết quả về hệ thống.

6. Hệ thống xác minh tính hợp lệ, sau đó cập nhật trạng thái đơn hàng.

Thông báo kết quả được gửi tới khách qua Email/SMS.

**2.4.3. Kịch bản *Quản lý đơn hàng bởi nhân viên***

* Các đối tượng tham gia: Nhân viên, Giao diện quản trị, Module Đơn hàng, Email/SMS,
* Trình tự tương tác:

1. Nhân viên đăng nhập hệ thống và xem danh sách đơn hàng mới.

2. Nhân viên chọn đơn cần xử lý, hệ thống hiển thị chi tiết.

3. Nhân viên xác nhận đơn → hệ thống trừ tồn kho sản phẩm.

4. Nhân viên nhập thông tin vận chuyển, hệ thống gửi dữ liệu sang dịch vụ giao hàng.

5. Trạng thái đơn hàng được cập nhật sang “Đang giao”.

6. Hệ thống gửi thông báo cho khách hàng qua Email/SMS.

**2.4.4. Ý nghĩa của biểu đồ tuần tự**

Các biểu đồ tuần tự giúp làm rõ:

* Thứ tự thực hiện các bước trong từng quy trình nghiệp vụ.
* Cách các đối tượng (khách hàng, nhân viên, hệ thống con) phối hợp với nhau.
* Điểm tích hợp với hệ thống bên ngoài (cổng thanh toán, dịch vụ vận chuyển).

Nhờ đó, quá trình thiết kế và lập trình có định hướng rõ ràng hơn, đảm bảo đúng yêu cầu đã khảo sát.

**2.5. Biểu đồ hợp tác (Collaboration Diagram)**

Biểu đồ hợp tác (hay còn gọi là biểu đồ cộng tác hoặc biểu đồ giao tiếp) được sử dụng để mô tả cách các đối tượng trong hệ thống phối hợp với nhau thông qua việc trao đổi thông điệp. Nếu như biểu đồ tuần tự nhấn mạnh vào trình tự thời gian, thì biểu đồ hợp tác lại tập trung vào mối quan hệ và sự liên kết giữa các đối tượng trong quá trình thực hiện một chức năng.

Trong hệ thống website bán điện thoại di động Sơn Thảo, biểu đồ hợp tác giúp hình dung rõ ràng vai trò của từng đối tượng và cách chúng cộng tác để hoàn thành một tác vụ.

**2.5.1. Ví dụ kịch bản *Đặt hàng trực tuyến***

**Các đối tượng tham gia trong kịch bản này bao gồm:**

* Khách hàng: tác nhân khời tạo yêu cầu mua hàng.
* Giao diện website: nhận yêu cầu từ khách hàng và chuyển đến các module liên quan.
* Giỏ hàng (Cart): lưu tạm danh sách sản phẩm khách hàng đã chọn.
* Đơn hàng (Order): quản lý thông tin đơn hàng mới được tạo.
* Thanh toán (Payment): xử lý thông tin giao dịch thanh toán.
* Dịch vụ Email/SMS: gửi thông báo xác nhận đơn hàng đến khách hàng.

**Quá trình cộng tác diễn ra như sau:**

1. Khách hàng thao tác trên giao diện để chọn sản phẩm và thêm vào giỏ hàng.
2. Giao diện website gửi yêu cầu đến đối tượng Giỏ hàng để lưu thông tin sản phẩm.
3. Khi khách xác nhận mua, Giao diện chuyển dữ liệu sang Đơn hàng để tạo mới đơn hàng.
4. Đơn hàng gọi đến đối tượng Thanh toán để xử lý phương thức thanh toán (COD hoặc Online).
5. Sau khi đơn hàng được tạo thành công, hệ thống gọi dịch vụ Email/SMS để gửi thông báo xác nhận cho khách.

**2.5.2. Ví dụ kịch bản *Quản lý đơn hàng của nhân viên***

**Các đối tượng tham gia trong kịch bản này bao gồm:**

* Nhân viên: người thực hiện thao tác quản trị.
* Giao diện quản trị: nhận yêu cầu và điều hướng.
* Đơn hàng: lưu trữ và cập nhật trạng thái đơn hàng.
* Kho sản phẩm: cập nhật số lượng tồn kho.
* Dịch vụ vận chuyển: nhận thông tin giao hàng.
* Dịch vu Email/SMS: thông báo kết quả đến cho khách hàng.

**Quá trình cộng tác:**

1. Nhân viên chọn đơn hàng cần xử lý trên Giao diện quản trị.
2. Giao diện gọi đến đối tượng Đơn hàng để hiển thị chi tiết.
3. Khi xác nhận đơn, Đơn hàng yêu cầu Kho sản phẩm trừ số lượng tồn kho.
4. Đơn hàng sau đó gửi dữ liệu vận chuyển đến dịch vụ giao hàng.
5. Trạng thái đơn được cập nhật và hệ thống gọi dịch vụ Email/SMS để thông báo cho khách.

**2.5.3. Ý nghĩa của biểu đồ hợp tác**

Biểu đồ hợp tác cho thấy rõ:

* Cách các đối tượng trong hệ thống có mối liên kết với nhau.
* Các thông điệp trao đổi giữa đối tượng để hoàn thành một quy trình nghiệp vụ.
* Hệ thống không chỉ dựa vào một thành phần riêng lẻ, mà cần sự phối hợp đồng bộ giữa nhiều đối tượng.

Nhờ việc xây dựng biểu đồ hợp tác, việc lập trình các module chức năng sẽ có định hướng rõ ràng hơn, tránh sự rời rạc hoặc thiếu liên kết trong thiết kế hệ thống.

**2.6. Biểu đồ trạng thái (State Diagram)**

Biểu đồ trạng thái được sử dụng để mô tả vòng đời của một đối tượng trong hệ thống, thông qua các trạng thái mà đối tượng đó có thể trải qua và các sự kiện dẫn đến sự thay đổi trạng thái. Trong hệ thống website bán điện thoại di động Sơn Thảo, đối tượng được chọn để xây dựng biểu đồ trạng thái là Đơn hàng (Order), vì đây là đối tượng trung tâm phản ánh toàn bộ quá trình mua bán.

**2.6.1. Các trạng thái chính của đơn hàng**

**Một đơn hàng trong hệ thống có thể trải qua các trạng thái như sau:**

* Khởi tạo: đơn hàng vừa được khách hàng khởi tạo, chưa được xác nhận.
* Chờ xác nhận: đơn hàng đã ghi nhận nhưng chưa được nhân viên xử lý.
* Đã xác nhận: nhân viên đã kiểm tra và xác nhận tính hợp lệ của đơn hàng.
* Đang xử lý: đơn hàng được chuẩn bị, đóng gói và sẵn sàng giao.
* Đang giao hàng: đơn hàng đã được chuyển cho đơn vị vận chuyển.
* Hoàn tất: khách hàng đã nhận sản phẩm, giao dịch hoàn thành.
* Huỷ bỏ: đơn hàng bị hủy bởi khách hàng hoặc nhân viên do một số lý do (hết hàng, khách đổi ý...).

**2.6.2. Các sự kiện kích hoạt thay đổi trạng thái**

* Khách hàng bấm “Đặt hàng” → đơn hàng chuyển sang Chờ xác nhận.
* Nhân viên xác nhận đơn → đơn hàng chuyển sang Đã xác nhận.
* Nhân viên tiến hành đóng gói → đơn hàng chuyển sang Đang xử lý.
* Hệ thống cập nhật thông tin từ đơn vị vận chuyển → đơn hàng sang trạng thái Đang giao hàng.
* Khách hàng nhận hàng thành công → đơn hàng sang trạng thái Hoàn tất.
* Nếu có sự cố (hết hàng, khách hủy đơn, không nhận hàng) → đơn hàng sang trạng thái Hủy bỏ.

**2.6.3. Ý nghĩa của biểu đồ trạng thái**

**Việc mô tả trạng thái của đơn hàng có ý nghĩa quan trọng:**

* Giúp theo dõi vòng đời của đơn hàng một cách rõ ràng.
* Hỗ trợ nhân viên và khách hàng nắm bắt tiến trình xử lý.
* Làm cơ sở để xây dựng các chức năng quản lý, thống kê và báo cáo.
* Đảm bảo sự nhất quán dữ liệu khi trạng thái thay đổi.

**2.7. Biểu đồ hoạt động (Activity Diagram)**

Biểu đồ hoạt động là loại biểu đồ UML dùng để mô tả luồng công việc (workflow) hoặc các bước xử lý của một quy trình trong hệ thống. Biểu đồ này giúp hình dung một cách trực quan các hoạt động diễn ra, thứ tự thực hiện và những rẽ nhánh điều kiện có thể xảy ra.

Trong hệ thống website bán điện thoại di động Sơn Thảo, một quy trình nghiệp vụ quan trọng cần được biểu diễn bằng biểu đồ hoạt động là xử lý đơn hàng.

**2.7.1. Quy trình xử lý đơn hàng**

**Các bước cơ bản của quy trình được mô tả như sau:**

1. **Khách hàng đặt hàng:** khách chọn sản phẩm, thêm vào giỏ hàng và xác nhận mua.
2. **Hệ thống kiểm tra thông tin đơn hàng:** bao gồm số lượng tồn kho, thông tin giao hàng và phương thức thanh toán.
3. **Điều kiện xử lý:**

* Nếu thông tin hợp lệ → hệ thống tạo đơn hàng mới.
* Nếu thông tin không hợp lệ (sai địa chỉ, hết hàng, thanh toán thất bại) → thông báo lỗi gửi về khách hàng.

1. **Nhân viên xác nhận đơn hàng:** nhân viên kiểm tra và xác nhận tính hợp lệ.
2. **Đóng gói sản phẩm:** đơn hàng đã xác nhận sẽ được chuyển sang bước đóng gói.
3. **Giao hàng:** hệ thống gửi thông tin sang đơn vị vận chuyển để tiến hành giao hàng.
4. **Cập nhật trạng thái đơn hàng:** khi giao hàng thành công, hệ thống chuyển trạng thái sang “Hoàn tất” và gửi thông báo đến khách hàng. Nếu giao hàng thất bại → trạng thái chuyển sang “Hủy”.

**2.7.2. Ý nghĩa của biểu đồ hoạt động**

* Thể hiện rõ ràng trình tự các bước xử lý đơn hàng từ khi khách đặt đến khi hoàn tất.
* Giúp nhận diện các điểm quyết định, nơi cần kiểm tra điều kiện trước khi chuyển sang bước tiếp theo.
* Hỗ trợ nhân viên hiểu được luồng công việc chuẩn, từ đó giảm thiểu sai sót trong quá trình xử lý.
* Là cơ sở để thiết kế chi tiết luồng xử lý trong hệ thống phần mềm.

**2.8. Biểu đồ thành phần (Component Diagram)**

**2.8.1. Khái niệm**

Biểu đồ thành phần (Component Diagram) trong UML được sử dụng để mô tả cấu trúc vật lý của hệ thống, thể hiện cách các module phần mềm (components) được tổ chức và quan hệ giữa chúng. Mỗi thành phần trong hệ thống có thể là một gói chức năng, một module phần mềm hoặc một dịch vụ bên ngoài.

Biểu đồ thành phần (Component Diagram) trong UML được sử dụng để mô tả cấu trúc vật lý của hệ thống, thể hiện cách các module phần mềm (components) được tổ chức và quan hệ giữa chúng. Mỗi thành phần trong hệ thống có thể là một gói chức năng, một module phần mềm hoặc một dịch vụ bên ngoài.

**2.8.2. Các thành phần chính của hệ thống**

Trong hệ thống website bán điện thoại di động Sơn Thảo, các thành phần chính có thể xác định như sau:

1. **Thành phần giao diện người dùng (FrontEnd):**

* Được xây dựng bằng Laravel Blade, HTML, CSS, JQuery.
* Chức năng: hiển thị thông tin sản phẩm, giỏ hàng, đơn hàng; giao tiếp với người dùng thông qua trình duyệt web hoặc thiết bị di động.

1. **Thành phần điều khiển (Controller Layer):**

* Xử lý yêu cầu từ người dùng và điều hướng đến các dịch vụ liên quan.
* Bao gồm các controller của Laravel (ProductController, OrderController, CartController, UserController...).

1. **Thành phần nghiệp vụ (Business Logic):**

* Chứa các lớp và service đảm nhiệm xử lý nghiệp vụ chính, như quản lý đơn hàng, quản lý khuyến mãi, xử lý thanh toán, quản lý kho hàng.
* Đây là lớp trung gian giữa giao diện và dữ liệu.

1. **Thành phần dữ liệu (Model & ORM):**

* Được xây dựng bằng Eloquent ORM của Laravel.
* Quản lý việc ánh xạ dữ liệu giữa các bảng trong cơ sở dữ liệu và các lớp trong hệ thống.

1. **Thành phần cơ sở dữ liệu (Database):**

* Sử dụng hệ quản trị cơ sở dữ liệu MySQL.
* Lưu trữ toàn bộ dữ liệu của hệ thống: thông tin sản phẩm, khách hàng, đơn hàng, thanh toán, vận chuyển.

1. **Thành phần quản trị hệ thống (Admin Panel):**

* Cho phép quản trị viên và nhân viên truy cập vào các chức năng quản lý sản phẩm, đơn hàng, khách hàng, khuyến mãi, thống kê.
* Giao diện quản trị được phát triển tách biệt với giao diện khách hàng.

1. **Thành phần dịch vụ tích hợp (External Services):**

* Cổng thanh toán: xử lý các giao dịch thanh toán online.
* Dịch vụ vận chuyển: nhận thông tin giao hàng và phản hồi trạng thái.
* Dịch vụ Email/SMS: gửi thông báo xác nhận đơn hàng, trạng thái giao hàng, mã OTP.

**2.8.3. Quan hệ giữa các thành phần**

* Giao diện người dùng (Frontend) giao tiếp với Controller Layer thông qua các request HTTP.
* Controller Layer gọi đến Business Logic để xử lý nghiệp vụ.
* Business Logic sử dụng Model & ORM để tương tác với cơ sở dữ liệu.
* Business Logic cũng gọi tới các dịch vụ tích hợp (Payment, Shipping, Email/SMS) thông qua API.
* Admin Panel kết nối trực tiếp với Controller Layer và Business Logic để quản lý toàn bộ hệ thống.

**2.8.4. Ý nghĩa của biểu đồ thành phần**

* Thể hiện rõ cấu trúc tổng thể của hệ thống, phân tách thành từng module với trách nhiệm riêng.
* Giúp dễ dàng bảo trì và mở rộng: khi cần thay đổi một chức năng, chỉ cần tác động vào thành phần liên quan.
* Là cơ sở để lập trình viên triển khai đúng kiến trúc định sẵn, tránh việc phát triển rời rạc.
* Hỗ trợ tích hợp các dịch vụ ngoài một cách mạch lạc, đảm bảo hệ thống hoạt động đồng bộ.

**2.9. Biểu đồ triển khai (Deployment Diagram)**

**2.9.1. Khái niệm**

Biểu đồ triển khai (Deployment Diagram) được sử dụng để mô tả cách các thành phần phần mềm được cài đặt và hoạt động trên hạ tầng phần cứng. Biểu đồ này thể hiện rõ kiến trúc vật lý của hệ thống, bao gồm các nút (nodes) như máy chủ, thiết bị người dùng, và mối quan hệ giữa chúng.

Trong hệ thống website bán điện thoại di động Sơn Thảo, biểu đồ triển khai giúp hình dung cách ứng dụng Laravel được triển khai trong thực tế, từ phía người dùng truy cập cho đến máy chủ xử lý và các dịch vụ bên ngoài.

**2.9.2. Các thành phần triển khai chính**

* **Thiết bị khách hàng (Client Devices):**
* Bao gồm máy tính cá nhân, laptop, smartphone hoặc máy tính bảng.
* Người dùng sử dụng trình duyệt web để truy cập vào website.
* **Máy chủ web/ứng dụng (Web & Application Server):**
* Cài đặt Laravel Framework.
* Chịu trách nhiệm xử lý logic nghiệp vụ, nhận request từ trình duyệt và trả về response.
* Chứa các thành phần: Frontend (Blade Template), Controller, Service, ORM (Eloquent).
* **Máy chủ cơ sở dữ liệu:**
* Chạy MySQL để lưu trữ dữ liệu của hệ thống (sản phẩm, khách hàng, đơn hàng, thanh toán, vận chuyển, khuyến mãi).
* Kết nối trực tiếp với Application Server thông qua kết nối bảo mật.
* **Các dịch vụ bên ngoài (External Services):**
* Cổng thanh toán trực tuyến: xử lý giao dịch online.
* Dịch vụ vận chuyển: nhận thông tin giao hàng và phản hồi trạng thái vận đơn.
* Dịch vụ Email/SMS: gửi thông báo tới khách hàng (xác nhận đơn, OTP, trạng thái đơn hàng).

**2.9.3. Quan hệ giữa các nút triển khai**

* Thiết bị khách hàng kết nối tới Web/Application Server thông qua Internet (HTTP/HTTPS).
* Application Server xử lý yêu cầu và truy cập Database Server để lấy hoặc lưu dữ liệu.
* Application Server đồng thời giao tiếp với các dịch vụ bên ngoài qua API để xử lý thanh toán, giao hàng và gửi thông báo.

**2.9.4. Ý nghĩa của biểu đồ triển khai**

* Cung cấp cái nhìn tổng quan về cách hệ thống hoạt động trong môi trường thực tế.
* Giúp xác định rõ các nút phần cứng và phần mềm liên quan trong quá trình triển khai.
* Hỗ trợ việc đánh giá hiệu suất và bảo mật: phân tách server ứng dụng và server cơ sở dữ liệu giúp tăng tính an toàn và dễ mở rộng.
* Tạo nền tảng cho việc triển khai thật trên hosting, VPS hoặc cloud server.

**2.10. Kết luận chương 2**

Trong chương 2, hệ thống website bán điện thoại di động Sơn Thảo đã được phân tích và thiết kế một cách chi tiết dựa trên các yêu cầu khảo sát từ chương 1. Quá trình phân tích được thực hiện bằng ngôn ngữ mô hình hóa UML, với việc xây dựng đầy đủ các biểu đồ:

* **Biểu đồ Use Case:** xác định rõ các tác nhân và chức năng chính mà hệ thống cần cung cấp.
* **Biểu đồ lớp:** mô tả cấu trúc dữ liệu và mối quan hệ giữa các lớp đối tượng trong hệ thống.
* **Biểu đồ tuần tự và biểu đồ hợp tác:** thể hiện luồng tương tác giữa các đối tượng trong từng kịch bản nghiệp vụ cụ thể.
* **Biểu đồ trạng thái và biểu đồ hoạt động:** làm rõ vòng đời của đối tượng quan trọng (đơn hàng) và luồng công việc khi xử lý đơn.
* **Biểu đồ thành phần và biểu đồ triển khai:** mô tả cấu trúc vật lý, cách các module phần mềm và dịch vụ bên ngoài được tổ chức, triển khai trên hạ tầng.

Việc phân tích và thiết kế hệ thống bằng UML đã giúp mô hình hóa toàn diện cả về cấu trúc lẫn hành vi, đảm bảo hệ thống được xây dựng đúng định hướng, dễ bảo trì và có khả năng mở rộng. Đây sẽ là nền tảng quan trọng để triển khai quá trình **xây dựng website** trong chương 3 và **kiểm thử hệ thống** ở chương 4.

**CHƯƠNG 3: XÂY DỰNG WEBSITE**

**3.1. Công nghệ sử dụng**

Hệ thống website được xây dựng nhằm đáp ứng nhu cầu quản lý sản phẩm/dịch vụ một cách hiệu quả, thân thiện với người dùng, đồng thời dễ bảo trì và mở rộng trong tương lai. Các công nghệ chính được sử dụng bao gồm:

**3.1.1. Ngôn ngữ lập trình**

**3.1.1.1. Phía Backend**

Hệ thống được xây dựng dựa trên PHP 8.4 kết hợp với Laravel 12 – một trong những framework PHP hiện đại và phổ biến nhất hiện nay. Việc sử dụng Laravel mang lại nhiều lợi ích cho quá trình phát triển phần mềm:

* **Phát triển nhanh chóng và dễ bảo trì:** Laravel cung cấp cấu trúc dự án rõ ràng theo mô hình **MVC (Model-View-Controller)**, giúp tách biệt phần logic xử lý dữ liệu, giao diện và điều hướng. Điều này làm cho việc phát triển, mở rộng và bảo trì hệ thống trở nên dễ dàng hơn.
* **Quản lý routing linh hoạt:** Laravel hỗ trợ định tuyến (routing) một cách dễ dàng, cho phép gán các URL tới các controller hoặc closure, đồng thời hỗ trợ các route group, middleware và tên route, giúp tổ chức hệ thống URL một cách logic và rõ ràng.
* **Bảo mật cao:** Laravel tích hợp sẵn các tính năng bảo mật như xác thực người dùng (Authentication), ủy quyền (Authorization), CSRF protection, hashing mật khẩu và xác thực hai lớp (2FA). Điều này giúp giảm thiểu các lỗ hổng bảo mật phổ biến trên web.
* **Quản lý session và middleware:** Laravel cung cấp các middleware để xử lý các request HTTP trước khi vào controller, hỗ trợ kiểm tra quyền truy cập, logging, hoặc các kiểm tra bảo mật khác. Session được Laravel quản lý linh hoạt, có thể lưu trữ trên file, database hoặc cache.
* **Tích hợp ORM Eloquent:** Eloquent ORM của Laravel cho phép thao tác cơ sở dữ liệu một cách trực quan thông qua các mô hình (model). Các thao tác CRUD (Create, Read, Update, Delete) có thể thực hiện dễ dàng mà không cần viết nhiều câu SQL thuần túy. Hệ thống còn hỗ trợ relationship giữa các bảng, query builder mạnh mẽ, giúp tăng tốc quá trình phát triển.
* **Hệ sinh thái mạnh mẽ:** Laravel đi kèm với nhiều gói hỗ trợ như Laravel Passport (xác thực API), Laravel Horizon (quản lý queue), Laravel Telescope (debug) và Laravel Mix (build frontend). Điều này giúp phát triển hệ thống đồng bộ từ backend tới frontend.

**3.1.1.2. Phía FrontEnd**

Phần giao diện người dùng (frontend) của hệ thống được xây dựng dựa trên HTML5, CSS3, JavaScript, jQuery và Bootstrap 5. Sự kết hợp này mang lại trải nghiệm người dùng trực quan, hiện đại, responsive và có khả năng tương tác động với server thông qua AJAX.

* **HTML5 & CSS3:**
* HTML5 cung cấp cấu trúc trang web rõ ràng, hỗ trợ các thẻ semantic như <header>, <footer>, <section> và <article> giúp cải thiện khả năng truy cập và SEO.
* CSS3 cho phép tạo các hiệu ứng trực quan, animation, transition, gradient, shadow, giúp giao diện trở nên sinh động và chuyên nghiệp.
* Sử dụng Flexbox và CSS Grid để bố trí layout phức tạp, dễ dàng tạo giao diện responsive, thích ứng với mọi kích thước màn hình từ desktop, tablet đến mobile.
* **Bootstrap 5:**
* Grid system của Bootstrap 5 cho phép chia layout thành các cột, hàng, giúp thiết kế giao diện responsive nhanh chóng.
* Cung cấp sẵn các component như button, navbar, modal, tooltip, popover, card, form… giúp xây dựng giao diện đồng nhất và tiết kiệm thời gian phát triển.
* Utility classes tích hợp sẵn cho phép điều chỉnh margin, padding, typography, màu sắc mà không cần viết CSS thủ công, giúp code sạch và dễ bảo trì.
* **jQuery:**
* jQuery giúp thao tác DOM, xử lý sự kiện (click, hover, input) và animation dễ dàng, rút ngắn đáng kể lượng code so với JavaScript thuần.
* Hỗ trợ các plugin mở rộng, ví dụ như datepicker, carousel, hoặc modal nâng cao, giúp tăng tốc phát triển giao diện.
* Kết hợp với AJAX, jQuery cho phép gửi request tới backend mà không cần reload toàn bộ trang, cải thiện trải nghiệm người dùng.
* **AJAX:**
* AJAX (Asynchronous JavaScript and XML) được sử dụng để gửi và nhận dữ liệu từ server một cách bất đồng bộ, giúp các thao tác như tìm kiếm, lọc sản phẩm, thêm giỏ hàng, hay load danh sách dữ liệu diễn ra mượt mà.
* Ví dụ: Khi người dùng chọn bộ lọc sản phẩm, jQuery sẽ gửi request AJAX tới Laravel, nhận dữ liệu JSON và render lại bảng hoặc danh sách sản phẩm mà không reload trang.
* Kết hợp với Bootstrap modals hoặc alerts, AJAX giúp hiển thị thông báo thành công hoặc lỗi một cách trực quan, nâng cao UX.
* **Tương tác người dùng nâng cao:**
* Sử dụng jQuery và Bootstrap, hệ thống có thể hiển thị modal, tooltip, popover, dropdown, animation và các hiệu ứng chuyển trang mượt mà.
* Các thao tác form, validation, autocomplete, hoặc pagination đều có thể thực hiện thông qua AJAX, giúp người dùng không bị gián đoạn trong quá trình tương tác.
* **Responsive & UX/UI:**
* Giao diện thiết kế chuẩn responsive, đảm bảo hiển thị tối ưu trên desktop, tablet, smartphone.
* UX được tối ưu bằng các thông báo trực quan, form dễ điền, menu điều hướng rõ ràng, modal hướng dẫn thao tác và alert thông báo trạng thái hành động.
* Bootstrap 5 kết hợp AJAX và jQuery giúp trải nghiệm người dùng mượt mà, giảm số lần reload trang và tăng tốc độ tương tác với hệ thống.

**3.1.2. Cơ sở dữ liệu**

Hệ thống sử dụng MySQL 9.4 làm hệ quản trị cơ sở dữ liệu (DBMS) chính. MySQL là một trong những cơ sở dữ liệu quan hệ phổ biến, nổi bật với hiệu năng cao, khả năng mở rộng tốt và độ ổn định trong việc quản lý dữ liệu lớn.

**3.1.2.1. Lý do sử dụng MySQL 9.4**

* Hiệu năng cao và ổn định: MySQL được tối ưu cho các truy vấn dữ liệu nhanh, hỗ trợ các chỉ mục (index), cache và các cơ chế tối ưu hóa query phức tạp, giúp hệ thống xử lý hàng nghìn request đồng thời mà vẫn duy trì tốc độ ổn định.
* Tương thích tốt với Laravel: Laravel có tích hợp sẵn Eloquent ORM, giúp thao tác cơ sở dữ liệu MySQL trở nên trực quan và dễ dàng thông qua các model mà không cần viết nhiều câu SQL thủ công.
* Hỗ trợ các tính năng nâng cao: MySQL 9.4 hỗ trợ các loại dữ liệu phong phú (INT, VARCHAR, TEXT, DATETIME, JSON…), transaction, foreign key, stored procedure, trigger, view… giúp thiết kế cơ sở dữ liệu linh hoạt và an toàn.

**3.1.2.2. Thiết kế cơ sở dữ liệu**

Hệ thống được thiết kế với các bảng tối ưu hóa cho thao tác CRUD (Create, Read, Update, Delete), đảm bảo quản lý dữ liệu hiệu quả cho các nghiệp vụ chính như:

* **Người dùng (Users):** Lưu thông tin tài khoản, quyền hạn, trạng thái hoạt động, lịch sử đăng nhập.
* **Sản phẩm/Dịch vụ (Products/Services):** Bao gồm các thông tin chi tiết như tên, mô tả, giá, hình ảnh, trạng thái tồn kho/dịch vụ, danh mục liên quan.
* **Đơn hàng (Orders):** Ghi nhận các thông tin đơn hàng, trạng thái thanh toán, chi tiết sản phẩm/dịch vụ, thông tin khách hàng, lịch sử thay đổi trạng thái.
* **Các bảng liên quan khác:** Bao gồm bảng danh mục, bảng chi tiết sản phẩm/dịch vụ, bảng giỏ hàng, bảng log hoặc lịch sử giao dịch, đảm bảo dữ liệu có quan hệ rõ ràng và dễ mở rộng.

**3.1.2.3. Tối ưu hoá cơ sở dữ liệu**

* **Chỉ mục (Index):** Được thiết lập trên các cột thường xuyên truy vấn, như user\_id, product\_id, order\_id, giúp truy vấn nhanh hơn.
* **Quan hệ (Relationship):** Các bảng được thiết kế với foreign key, đảm bảo tính toàn vẹn dữ liệu và hỗ trợ Eloquent ORM dễ dàng khai báo các mối quan hệ như one-to-many, many-to-many hoặc one-to-one.
* **Chuẩn hóa dữ liệu (Normalization):** Các bảng được chuẩn hóa để giảm trùng lặp dữ liệu, đồng thời đảm bảo tính nhất quán, nhưng vẫn kết hợp hợp lý với denormalization ở một số bảng để tăng hiệu năng khi cần.
* **Hỗ trợ transaction:** MySQL cho phép sử dụng transaction để đảm bảo các thao tác thay đổi dữ liệu phức tạp (ví dụ: tạo đơn hàng, trừ tồn kho, cập nhật trạng thái) được thực hiện đồng bộ và an toàn.

**3.1.2.4. Tích hợp Laravel**

* **Eloquent ORM:** Mỗi bảng cơ sở dữ liệu được ánh xạ thành một model trong Laravel, giúp thao tác CRUD trở nên trực quan, ví dụ: $user->orders()->create($data) để tạo đơn hàng cho người dùng.
* **Migration:** Laravel hỗ trợ migration để quản lý thay đổi cấu trúc cơ sở dữ liệu, đảm bảo việc triển khai và nâng cấp hệ thống trên môi trường khác nhau dễ dàng và an toàn.
* **Seeder và Factory:** Hỗ trợ khởi tạo dữ liệu thử nghiệm hoặc dữ liệu mặc định cho hệ thống, giúp kiểm thử và phát triển nhanh hơn.

Nhờ thiết kế cơ sở dữ liệu tối ưu, kết hợp cùng MySQL 9.4 và Eloquent ORM, hệ thống có khả năng quản lý dữ liệu phức tạp, truy xuất nhanh chóng và mở rộng linh hoạt, đáp ứng tốt các nghiệp vụ như quản lý sản phẩm, dịch vụ, người dùng và đơn hàng.

**3.1.3. Framework và thư viện hỗ trợ**

Hệ thống được phát triển dựa trên một số framework và thư viện mạnh mẽ, giúp tăng tốc phát triển, đảm bảo bảo mật và cải thiện trải nghiệm người dùng. Các công nghệ chính bao gồm Laravel 12, Bootstrap 5 và jQuery.

**3.1.3.1. Laravel 12**

Laravel 12 là framework PHP hiện đại, được sử dụng để xây dựng backend của hệ thống. Laravel hỗ trợ lập trình theo mô hình MVC (Model-View-Controller), giúp tách biệt rõ ràng phần logic, giao diện và routing. Một số chức năng nổi bật:

* **Routing:** Quản lý và định tuyến các request HTTP đến controller tương ứng. Laravel hỗ trợ route group, middleware và route naming, giúp tổ chức URL rõ ràng và logic.
* **Middleware:** Giúp xử lý request trước khi đến controller, ví dụ như kiểm tra xác thực người dùng, phân quyền truy cập, logging hoặc lọc dữ liệu.
* **Bảo mật:** Laravel tích hợp các cơ chế bảo mật mạnh mẽ như CSRF protection, hashing mật khẩu, xác thực hai lớp (2FA) và bảo vệ SQL injection.
* **Xác thực người dùng (Authentication & Authorization):** Laravel cung cấp hệ thống login, register, reset password, quản lý role và permission, giúp kiểm soát quyền truy cập.
* **Xử lý email:** Laravel hỗ trợ gửi email thông qua SMTP, Mailgun, hoặc các dịch vụ khác, giúp thông báo cho người dùng về đơn hàng, đăng ký, hay thông tin quan trọng.
* **Session:** Quản lý session linh hoạt, lưu trữ trên file, database hoặc cache, dùng để lưu trạng thái người dùng hoặc thông tin tạm thời.
* **ORM Eloquent:** Giúp thao tác cơ sở dữ liệu MySQL dễ dàng thông qua các model, hỗ trợ các quan hệ (one-to-many, many-to-many), query builder mạnh mẽ, giảm thiểu việc viết SQL thủ công.

**3.1.3.2. Bootstrap 5**

Bootstrap 5 là framework CSS phổ biến, được dùng để xây dựng giao diện responsive, hiện đại và đồng bộ:

* **Grid system:** Cho phép thiết kế bố cục linh hoạt theo dạng lưới, dễ dàng điều chỉnh trên nhiều kích thước màn hình từ desktop, tablet đến smartphone.
* **Component sẵn có:** Bao gồm navbar, button, card, modal, form, alert, tooltip… giúp xây dựng giao diện nhanh và đồng nhất.
* **Utility classes:** Cho phép thay đổi margin, padding, color, typography, shadow… mà không cần viết CSS thủ công, giúp code sạch và dễ bảo trì.
* **Responsive design:** Bootstrap 5 hỗ trợ media query và responsive utilities, đảm bảo trải nghiệm người dùng mượt mà trên mọi thiết bị.
* **Tương thích với JavaScript:** Nhiều component của Bootstrap 5 tích hợp sẵn các hiệu ứng động và tương tác (modal, dropdown, carousel…), giúp nâng cao UX mà không cần viết quá nhiều code JS.

**3.1.3.3. jQuery**

jQuery là thư viện JavaScript được sử dụng để tương tác động trên frontend, hỗ trợ giao tiếp giữa người dùng và backend:

* **DOM manipulation:** Giúp thao tác với các phần tử HTML nhanh chóng, ví dụ thay đổi nội dung, thêm/xóa element, thay đổi class, style.
* **Xử lý sự kiện:** Hỗ trợ click, hover, submit, focus… dễ dàng, giúp tạo các hiệu ứng tương tác trực quan.
* **AJAX:** Cho phép gửi request bất đồng bộ tới backend (Laravel) để load dữ liệu, submit form hoặc xử lý thao tác mà không cần reload trang, nâng cao trải nghiệm người dùng.
* **Animation & hiệu ứng UI:** Hỗ trợ fade, slide, toggle, animate… giúp giao diện sống động và dễ tương tác.
* **Tương thích Bootstrap:** jQuery kết hợp tốt với các component của Bootstrap 5 như modal, dropdown, carousel, giúp triển khai các tính năng UI phức tạp nhanh chóng.

Nhờ sự kết hợp giữa Laravel 12, Bootstrap 5 và jQuery, hệ thống vừa đảm bảo backend mạnh mẽ, bảo mật, dễ quản lý, vừa có frontend đẹp mắt, responsive và tương tác động, đáp ứng đầy đủ các nhu cầu nghiệp vụ cũng như trải nghiệm người dùng hiện đại.

**3.1.4. Công cụ hỗ trợ**

Để phát triển và quản lý hệ thống một cách hiệu quả, nhóm phát triển sử dụng nhiều công cụ hỗ trợ khác nhau, từ quản lý thư viện, môi trường lập trình, cho đến quản lý cơ sở dữ liệu và phiên bản mã nguồn. Các công cụ chính bao gồm:

**3.1.4.1. Composer**

* Composer là công cụ quản lý thư viện và package PHP.
* Giúp cài đặt, cập nhật và quản lý các gói (package) mà Laravel và dự án PHP yêu cầu.
* Cho phép khai báo các dependency trong file composer.json, đảm bảo các thư viện cần thiết được tự động cài đặt trên môi trường phát triển và triển khai.
* Ví dụ: Laravel, PHPMailer, Guzzle… đều có thể cài đặt nhanh chóng qua Composer mà không phải tải thủ công.

**3.1.4.2. NPM/Yarn**

* NPM (Node Package Manager) và Yarn là các công cụ quản lý package frontend.
* Sử dụng để cài đặt và quản lý các thư viện như Bootstrap 5, jQuery, các plugin JavaScript hoặc các công cụ build frontend như Webpack, Vite.
* Giúp tự động cập nhật, quản lý version, đảm bảo tính nhất quán giữa các môi trường phát triển và triển khai.
* Cho phép tích hợp với các workflow frontend hiện đại, ví dụ: compile SCSS/SASS, minify JS, bundle module.

**3.1.4.3. Git**

* Git là hệ thống quản lý phiên bản mã nguồn phân tán.
* Giúp theo dõi lịch sử thay đổi mã nguồn, hỗ trợ quay lại phiên bản trước đó khi cần thiết.
* Tối ưu cho làm việc nhóm, cho phép nhiều lập trình viên cùng phát triển trên một dự án thông qua các branch, merge, pull request.
* Kết hợp với các nền tảng như GitHub hoặc GitLab, Git giúp quản lý code, review, backup và triển khai CI/CD hiệu quả.

**3.1.4.4. DBeaver / Sequel Ace**

* DBeaver và Sequel Ace là các công cụ quản lý cơ sở dữ liệu MySQL.
* Cho phép thực hiện các thao tác như:
* Tạo, sửa, xóa bảng và cột trong cơ sở dữ liệu.
* Thực hiện truy vấn SQL (SELECT, INSERT, UPDATE, DELETE).
* Backup và restore dữ liệu, kiểm tra dữ liệu thử nghiệm.
* Quản lý quan hệ giữa các bảng, index, foreign key.
* Hỗ trợ trực quan hóa dữ liệu, giúp lập trình viên dễ dàng kiểm tra và tối ưu cơ sở dữ liệu.

**3.1.4.5. Visual Studio Code (VS Code)**

* Visual Studio Code là IDE chính được sử dụng trong dự án.
* Hỗ trợ lập trình PHP, JavaScript, CSS, HTML và tích hợp nhiều plugin hữu ích, ví dụ:
* PHP Intelephense (hỗ trợ gợi ý code, kiểm tra lỗi PHP).
* Laravel Blade Snippets (hỗ trợ cú pháp Blade của Laravel).
* Prettier, ESLint (tự động format và kiểm tra code frontend).
* GitLens (hiển thị lịch sử commit, tracking code).
* Cho phép debugging, quản lý terminal, tích hợp Git và chạy các lệnh Composer/NPM trực tiếp từ IDE, giúp phát triển nhanh, tiện lợi và đồng bộ.

Nhờ sự kết hợp của các công cụ trên, nhóm phát triển có thể:

* Tối ưu lập trình frontend và backend (VS Code, Bootstrap, jQuery).
* Quản lý cơ sở dữ liệu và dữ liệu thử nghiệm (DBeaver, Sequel Ace).
* Quản lý mã nguồn và làm việc nhóm hiệu quả (Git).

Kết quả là dự án được phát triển nhanh chóng, ổn định, dễ bảo trì và triển khai trên nhiều môi trường khác nhau.

**3.1.5. Môi trường phát triển**

**3.2. Giao diện website**

**3.3. Mô tả giao diện**