

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

XÂY DỰNG WEBSITE BÁN ĐỒ ĐIỆN TỬ

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Bùi Văn Kiên** |
| **Sinh viên thực hiện:** | **Nguyễn Ngọc Anh - B20DCCN059** |
|  | **Nguyễn Công Khánh – B20DCCN375** |
|  | **Nguyễn Hải Phong – B20DCCN495** |
| **Niên khoá:** | **2020-2025** |
| **Hệ đào tạo:** | **Đại học chính quy** |
|  |  |

**Hà Nội – 2024**

## NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

## (CỦA GIẢNG VIÊN HƯỚNG DẪN)

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**Điểm: ...............................(**bằng chữ )

**Đồng ý/ Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp.

.................., ngày …. tháng …. năm …..

# CÁN BỘ GIẢNG VIÊN HƯỚNG DẪN

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(CỦA GIẢNG VIÊN PHẢN BIỆN)**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**.............................................................................................................................................**

**Điểm: ...............................(**bằng chữ )

Đồng ý/ Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp.

..................., ngày …. tháng …. năm …..

**CÁN BỘ GIẢNG VIÊN PHẢN BIỆN**

# LỜI CẢM ƠN

Trước hết, nhóm em xin gửi lời cảm ơn sâu sắc đến quý thầy cô của Học viện Công nghệ Bưu chính Viễn thông, đặc biệt là các thầy cô khoa Công nghệ thông tin I. Chính nhờ sự hướng dẫn tận tình và kiến thức quý báu mà các thầy cô đã truyền đạt, nhóm em mới có thể hoàn thành tốt quá trình học tập và nghiên cứu tại trường.

Nhóm em cũng xin bày tỏ lòng biết ơn đặc biệt đến thầy giáo Bùi Văn Kiên, người đã trực tiếp chỉ dẫn và hỗ trợ nhóm trong những lúc gặp khó khăn, giúp nhóm em có định hướng và hoàn thành đồ án đúng tiến độ.

Bên cạnh đó, nhóm em chân thành cảm ơn gia đình, bạn bè và các anh chị khóa trên đã luôn đồng hành, động viên và hỗ trợ nhóm em trong suốt hành trình thực hiện đồ án này.

Mặc dù đã nỗ lực hết mình, nhưng vì thời gian và kinh nghiệm còn hạn chế nên chắc chắn đồ án vẫn còn nhiều thiếu sót. Nhóm em rất mong nhận được những góp ý và chỉ dẫn quý báu từ các thầy cô để có thể hoàn thiện hơn.

Nhóm em xin chân thành cảm ơn!

Hà Nội, tháng 12 năm 2024

Nhóm sinh viên thực hiện

**Nguyễn Ngọc Anh**

**Nguyễn Công Khánh**

**Nguyễn Hải Phong**

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Ký hiệu** | **Giải nghĩa tiếng Anh** | **Giải nghĩa tiếng Việt** |
| UML | Unified Modeling Language | Mẫu thiết kế Luồng dữ liệu một chiều |
| UI | User Interface | Giao diện người dùng |
| CRUD | Create, Read, Update, và Delete | Tạo, Đọc, Cập nhập, Xóa |
| ORM | Object-Relational Mapping | Ánh xạ Đối tượng-Quan hệ |
| UC | Use Case | Trường hợp sử dụng |
| IDE | Intergrated Development Environment | Môi trường phát triển tích hợp |
| CSDL |  | Cơ sở dữ liệu |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| URL | Uniform Resource Locator | Đường dẫn liên kết đến website |

# LỜI MỞ ĐẦU

Trong thời đại mà công nghệ ngày càng phát triển mạnh mẽ, thương mại điện tử đã trở thành một phần thiết yếu trong cuộc sống hàng ngày, đóng vai trò không chỉ là phương thức giao dịch mà còn là động lực thúc đẩy sự sáng tạo và đổi mới trong kinh doanh. Đặc biệt, lĩnh vực kinh doanh thiết bị điện tử trực tuyến đang ngày càng được chú trọng, khi nhu cầu của người tiêu dùng về các sản phẩm công nghệ không ngừng tăng cao.

Website bán thiết bị điện tử mang đến nhiều lợi ích nổi bật cho cả người mua lẫn người bán. Người mua có thể dễ dàng tiếp cận các sản phẩm đa dạng, từ điện thoại di động, laptop, phụ kiện công nghệ đến các thiết bị gia dụng thông minh, mọi lúc, mọi nơi chỉ cần có kết nối internet. Đối với doanh nghiệp, website là cầu nối để mở rộng thị trường, nâng cao khả năng cạnh tranh và xây dựng thương hiệu. Hơn thế nữa, các tính năng hiện đại như tìm kiếm thông minh, đánh giá sản phẩm, và phương thức thanh toán an toàn giúp tối ưu hóa trải nghiệm người dùng, đồng thời đáp ứng tốt nhu cầu ngày càng đa dạng của khách hàng.

Xuất phát từ thực tế đó, đề tài “Xây dựng Website Bán Thiết Bị Điện Tử” được chọn nhằm góp phần giải quyết nhu cầu phát triển kênh thương mại điện tử chuyên biệt cho ngành hàng này. Hệ thống không chỉ tập trung vào việc cung cấp sản phẩm mà còn hướng đến xây dựng một nền tảng thân thiện, tiện lợi và an toàn, góp phần thúc đẩy sự phát triển bền vững của thương mại điện tử trong lĩnh vực thiết bị điện tử.

# CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN VÀ CÔNG NGHỆ LIÊN QUAN

## **1.1 Giới thiệu bài toán**

Trong thời đại số hóa ngày nay, người tiêu dùng ngày càng ưa chuộng việc mua sắm trực tuyến nhờ những lợi ích vượt trội mà nó mang lại. Website bán linh kiện điện tử cho phép người dùng dễ dàng tìm kiếm, lựa chọn và mua sắm các linh kiện cần thiết bất kỳ lúc nào, 24/7, mà không bị ràng buộc bởi thời gian hay địa điểm. Cùng với đó, dịch vụ giao hàng tận nơi giúp tiết kiệm thời gian và công sức cho người mua, đặc biệt là khi nhu cầu về các sản phẩm công nghệ ngày càng tăng cao.

Website bán linh kiện điện tử không chỉ là kênh cung cấp sản phẩm đa dạng, từ vi xử lý, bo mạch chủ, card đồ họa đến các phụ kiện chuyên dụng khác, mà còn giúp người bán mở rộng phạm vi kinh doanh một cách nhanh chóng, vượt qua các giới hạn địa lý. Với tính năng tìm kiếm thông minh, đánh giá sản phẩm và gửi phản hồi, người mua dễ dàng tìm thấy sản phẩm phù hợp với nhu cầu của mình. Ngoài ra, hệ thống website còn chú trọng đến bảo mật thanh toán, dữ liệu khách hàng, các ưu đãi cạnh tranh, và quản lý tài khoản hiệu quả, nhằm nâng cao trải nghiệm người dùng.

Nhìn chung, website bán linh kiện điện tử không chỉ là một nền tảng giao dịch mà còn là một giải pháp tích hợp nhiều tính năng ưu việt, đáp ứng nhu cầu ngày càng đa dạng của người mua và người bán. Trên cơ sở đó, đề tài Xây dựng Website Bán Linh Kiện Điện Tử được chọn nhằm góp phần hỗ trợ quá trình mua bán linh kiện điện tử trực tuyến một cách thuận tiện, nhanh chóng và hiệu quả.

## **1.2. Khảo sát**

### 1.2.1. Phần Mua Sắm Của Khách Hàng

Phần Mua Sắm Của Khách Hàng là nơi người tiêu dùng có thể tìm kiếm, xem thông tin chi tiết và mua sắm các loại linh kiện điện tử phù hợp với nhu cầu. Dưới đây là mô tả chi tiết về các tính năng và chức năng quan trọng trong phần này:

Tính Năng Tìm Kiếm: Hỗ trợ khách hàng dễ dàng tìm thấy các linh kiện mong muốn. Người dùng có thể nhập từ khóa, tên linh kiện (ví dụ: "CPU Intel i7"), hoặc mã sản phẩm để tìm kiếm nhanh chóng và chính xác.

Danh Mục và Bộ Lọc: Các sản phẩm được sắp xếp theo danh mục rõ ràng, chẳng hạn như "Vi xử lý," "Bo mạch chủ," "Card đồ họa," "RAM," "Ổ cứng," và "Nguồn điện." Bộ lọc cho phép thu hẹp kết quả tìm kiếm dựa trên các tiêu chí như giá, thương hiệu, khả năng tương thích, hiệu năng, và đánh giá từ người dùng.

Trang Sản Phẩm Chi Tiết: Mỗi sản phẩm có trang thông tin riêng biệt, bao gồm mô tả chi tiết, thông số kỹ thuật, hình ảnh chất lượng cao, đánh giá từ người dùng, và các tùy chọn khác như số lượng, phiên bản, hoặc cấu hình.

Giỏ Hàng và Thanh Toán: Người mua có thể thêm các linh kiện cần thiết vào giỏ hàng để kiểm tra trước khi tiến hành thanh toán. Quá trình thanh toán an toàn, hỗ trợ nhiều phương thức như chuyển khoản, thẻ tín dụng, ví điện tử, hoặc thanh toán khi nhận hàng (COD).

Ưu Đãi và Khuyến Mãi: Cung cấp các chương trình khuyến mãi hấp dẫn như giảm giá theo mùa, combo linh kiện giá rẻ, hoặc ưu đãi dành riêng cho khách hàng thân thiết. Thông báo khuyến mãi được hiển thị nổi bật trên trang chủ, trang sản phẩm, hoặc thông qua thông báo cá nhân.

Hệ Thống Đánh Giá và Nhận Xét: Người mua có thể đọc các đánh giá thực tế từ khách hàng khác về sản phẩm, kèm theo hình ảnh và trải nghiệm sử dụng thực tế. Những thông tin này giúp khách hàng có cái nhìn khách quan hơn trước khi đưa ra quyết định mua sắm.

Dịch Vụ Giao Hàng và Theo Dõi Đơn Hàng: Hỗ trợ giao hàng nhanh chóng trên toàn quốc với nhiều tùy chọn vận chuyển. Khách hàng có thể theo dõi trạng thái đơn hàng của mình trực tiếp trên website, biết được lịch trình giao hàng và thời gian dự kiến nhận hàng.

Trang Cá Nhân và Lịch Sử Mua Sắm: Mỗi khách hàng có một tài khoản cá nhân, nơi họ có thể quản lý thông tin cá nhân, theo dõi các đơn hàng đã đặt, lưu lại các sản phẩm yêu thích, và xem lịch sử mua sắm để dễ dàng mua lại các linh kiện đã từng mua.

Những tính năng này được thiết kế nhằm mang lại sự tiện lợi, an toàn, và trải nghiệm mua sắm tốt nhất cho khách hàng khi tìm kiếm và mua linh kiện điện tử.

### 1.2.2. Phần Quản Trị Viên Của Website Bán Linh Kiện Điện Tử

Phần quản trị viên của website bán linh kiện điện tử không được công khai và được thiết kế như một giao diện riêng biệt dành cho những người quản trị hệ thống và điều hành các quy trình kinh doanh. Quản trị viên đóng vai trò quan trọng trong việc duy trì hoạt động trơn tru của hệ thống với các nhiệm vụ chính như sau:

Quản Lý Tài Khoản Người Dùng: Theo dõi và kiểm soát tài khoản của khách hàng và nhân viên, bao gồm việc kích hoạt, khóa tài khoản hoặc xử lý các vấn đề liên quan đến thông tin người dùng. Điều này đảm bảo rằng chỉ những người có quyền hạn mới có thể truy cập và quản lý hệ thống.

Quản Lý Sản Phẩm Linh Kiện Điện Tử: Thêm, sửa đổi, hoặc xóa các linh kiện điện tử khỏi hệ thống. Quản trị viên cũng có thể cập nhật thông tin chi tiết về sản phẩm như mô tả, giá cả, hình ảnh, thông số kỹ thuật, và các chương trình khuyến mãi đặc thù cho từng loại linh kiện (ví dụ: loại chip, nhà sản xuất, tương thích với các thiết bị khác).

Cấu Hình Hệ Thống: Thiết lập các cấu hình quan trọng của hệ thống như cài đặt thanh toán, chính sách vận chuyển, mức độ bảo mật, và giao diện người dùng. Đặc biệt, các tính năng liên quan đến việc lọc và tìm kiếm linh kiện theo các thông số kỹ thuật sẽ được tùy chỉnh để phù hợp với nhu cầu của khách hàng.

Quản Lý Đơn Hàng: Theo dõi trạng thái đơn hàng, xử lý các yêu cầu đổi/trả hàng, đặc biệt là các đơn hàng liên quan đến linh kiện điện tử có thể yêu cầu kiểm tra kỹ lưỡng về tính tương thích và chất lượng. Quản trị viên cũng đảm bảo rằng các đơn hàng được giao đúng hạn và đúng yêu cầu của khách hàng.

Theo Dõi Bảo Mật: Đảm bảo hệ thống luôn an toàn bằng cách giám sát các hoạt động đăng nhập bất thường, quản lý quyền truy cập, và thường xuyên kiểm tra hệ thống để phát hiện các lỗ hổng bảo mật. Bảo vệ dữ liệu sản phẩm và thông tin khách hàng là ưu tiên hàng đầu.

Quản Lý Dữ Liệu: Truy cập và phân tích dữ liệu kinh doanh, bao gồm thống kê doanh thu, số lượng đơn hàng, xu hướng mua sắm linh kiện điện tử, và các dữ liệu liên quan đến hiệu suất sản phẩm. Những thông tin này hỗ trợ quản trị viên trong việc ra quyết định chiến lược nhằm phát triển kinh doanh.

Quản Lý Nội Dung: Quản lý các nội dung trên website như bài viết hướng dẫn, banner quảng cáo cho các linh kiện mới, hoặc thông báo khuyến mãi. Đảm bảo giao diện luôn hấp dẫn và cập nhật với các thông tin kỹ thuật mới nhất về sản phẩm, giúp khách hàng dễ dàng tiếp cận và hiểu rõ về sản phẩm.

Quản Lý Kho Hàng: Theo dõi tồn kho các linh kiện điện tử, cập nhật số lượng hàng hóa, quản lý nhập xuất kho, và đảm bảo luôn có đủ sản phẩm để đáp ứng nhu cầu của khách hàng. Điều này giúp duy trì sự ổn định trong cung cấp sản phẩm và tránh tình trạng hết hàng.

Hỗ Trợ Khách Hàng: Quản trị viên quản lý các yêu cầu hỗ trợ từ khách hàng, xử lý các câu hỏi về sản phẩm, cung cấp thông tin kỹ thuật, và đảm bảo sự hài lòng của khách hàng đối với dịch vụ. Điều này bao gồm việc phản hồi nhanh chóng và chính xác các thắc mắc, giúp xây dựng lòng tin và sự trung thành từ phía khách hàng.

Phần quản trị viên được thiết kế với các quyền hạn và công cụ phù hợp để tối ưu hóa quy trình vận hành của website bán linh kiện điện tử, từ đó mang lại trải nghiệm tốt nhất cho người dùng cuối và đảm bảo sự phát triển bền vững của kinh doanh trực tuyến trong lĩnh vực linh kiện điện tử.

## **1.3. Giải pháp công nghệ cho bài toán**

### 1.3.1 Kiến trúc hệ thống

Hệ thống được phát triển dựa trên mô hình 3 lớp, bao gồm Presentation Layer (lớp giao diện), Business Logic Layer (lớp xử lý nghiệp vụ), và Data Access Layer (lớp truy cập dữ liệu) [2]:

Lớp giao diện (Presentation Layer): Đây là lớp đảm nhận nhiệm vụ giao tiếp trực tiếp với người dùng. Nó bao gồm các thành phần giao diện để nhập liệu, hiển thị dữ liệu, và kiểm tra tính chính xác của dữ liệu trước khi gửi tới lớp logic. Hai thành phần chính của lớp này là:

Thành phần giao diện người dùng (UI Components): Bao gồm các yếu tố xây dựng giao diện hệ thống, chịu trách nhiệm thu nhận dữ liệu từ người dùng và hiển thị kết quả, chẳng hạn như textbox, button, combobox, ...

Thành phần quản lý giao diện (UI Process Components): Đảm nhận việc điều phối và chuyển đổi giữa các giao diện khác nhau trong hệ thống.

Lớp xử lý nghiệp vụ (Business Logic Layer): Lớp này được chia thành hai nhiệm vụ chính:

Xử lý dữ liệu: Nhận dữ liệu từ lớp giao diện, thực hiện các xử lý cần thiết trước khi chuyển xuống lớp truy cập dữ liệu và lưu trữ trong hệ quản trị cơ sở dữ liệu.

Kiểm tra và xử lý nghiệp vụ: Đảm bảo dữ liệu đáp ứng các ràng buộc, tính toàn vẹn, và hợp lệ. Đồng thời, thực hiện các tính toán, xử lý nghiệp vụ và trả kết quả lại cho lớp giao diện.

Lớp truy cập dữ liệu (Data Access Layer): Chức năng chính của lớp này là giao tiếp với hệ quản trị cơ sở dữ liệu. Nó thực hiện các tác vụ liên quan đến lưu trữ và truy vấn dữ liệu như tìm kiếm, thêm mới, cập nhật, và xóa dữ liệu.

### 1.3.2. Lập trình giao diện với ReactJS

Giao diện hệ thống được xây dựng dựa trên **ReactJS**, một thư viện JavaScript mã nguồn mở do Facebook phát triển. ReactJS được thiết kế để tạo ra giao diện người dùng (UI) hiệu suất cao thông qua việc sử dụng các thành phần (component). Với triết lý "Learn Once, Write Anywhere", ReactJS dễ dàng tích hợp vào các ứng dụng web và di động lớn nhỏ.

#### ****Đặc điểm nổi bật:****

* **Component-Based:**  
  ReactJS cho phép chia giao diện thành các thành phần độc lập, nhỏ gọn. Mỗi thành phần có thể tái sử dụng và dễ bảo trì, giúp cải thiện năng suất phát triển và giảm lỗi trong quá trình triển khai.
* **Virtual DOM:**  
  React sử dụng Virtual DOM để tối ưu hóa hiệu suất. Khi có sự thay đổi dữ liệu, React chỉ cập nhật các phần tử thay đổi thay vì làm mới toàn bộ giao diện, giúp ứng dụng nhanh và mượt hơn.
* **One-Way Data Binding:**  
  Dữ liệu trong ReactJS được truyền từ cha xuống con (props) theo một hướng duy nhất, giúp dễ dàng kiểm soát luồng dữ liệu và phát hiện lỗi trong ứng dụng.

#### ****Thư viện hỗ trợ:****

* **React Router:**  
  Công cụ mạnh mẽ để quản lý định tuyến (routing) giữa các trang hoặc thành phần giao diện trong ứng dụng.
  + Chuyển đổi nhanh chóng giữa các trang mà không cần tải lại trình duyệt.
  + Hỗ trợ định tuyến động, phù hợp cho các ứng dụng có nhiều cấp độ phức tạp.
  + Dễ dàng cấu hình các đường dẫn như /home, /product/:id, hoặc thậm chí các đường dẫn bảo vệ chỉ dành cho người dùng đã đăng nhập.
* **Redux:**  
  Redux được sử dụng để quản lý trạng thái toàn cục trong ứng dụng React, đặc biệt hữu ích khi ứng dụng có nhiều thành phần chia sẻ trạng thái chung như giỏ hàng, thông tin người dùng, hoặc dữ liệu sản phẩm.
  + Lưu trữ và quản lý trạng thái tập trung.
  + Giảm sự phụ thuộc lẫn nhau giữa các thành phần, giúp mã nguồn rõ ràng và dễ mở rộng.
  + Cung cấp công cụ DevTools để kiểm tra và debug trạng thái trực quan.
* **Material-UI:**  
  Thư viện giao diện người dùng phổ biến dựa trên triết lý Material Design của Google, cung cấp các thành phần thiết kế sẵn với phong cách hiện đại và chuyên nghiệp.
  + Bao gồm các component như Buttons, Tables, Forms, AppBars,... giúp giảm thời gian thiết kế giao diện.
  + Hỗ trợ tùy chỉnh giao diện để phù hợp với nhu cầu của từng ứng dụng cụ thể.
  + Hỗ trợ thiết kế responsive, tương thích với nhiều kích thước màn hình.

#### ****Tích hợp ReactJS trong ứng dụng:****

* ReactJS được kết hợp với các thư viện và công cụ khác như Axios hoặc Fetch API để giao tiếp với backend (RESTful API).
* Sử dụng **component lifecycle** hoặc **hooks (useEffect, useState)** để quản lý luồng dữ liệu và trạng thái trong ứng dụng.
* Tối ưu hiệu suất bằng cách sử dụng lazy-loading và code-splitting, đặc biệt trong các dự án lớn.

### 1.3.3. Lập trình backend với Java Spring Boot

Phần backend được phát triển bằng **Spring Boot**, một framework mạnh mẽ trong hệ sinh thái Spring của Java. Spring Boot giúp đơn giản hóa quá trình phát triển các ứng dụng backend nhờ vào việc tự động cấu hình và tích hợp sẵn các công cụ cần thiết.

#### ****Đặc điểm nổi bật:****

* **Cấu hình tự động:**  
  Spring Boot đi kèm với các cấu hình mặc định, giúp giảm thiểu thời gian cấu hình thủ công. Điều này đặc biệt hữu ích trong giai đoạn phát triển nhanh hoặc triển khai ứng dụng mới.
* **RESTful API:**  
  Spring Boot hỗ trợ xây dựng dịch vụ RESTful API dễ dàng với các annotation như @RestController, @RequestMapping,...
  + Đơn giản hóa việc tạo ra các endpoint để giao tiếp với frontend.
  + Hỗ trợ chuyển đổi dữ liệu tự động giữa các định dạng như JSON và XML.
* **Quản lý dependency:**  
  Sử dụng Maven hoặc Gradle, Spring Boot giúp quản lý các thư viện và phiên bản một cách hiệu quả. Điều này đảm bảo tính nhất quán và giảm thiểu xung đột giữa các thư viện.
* **Khả năng mở rộng:**  
  Hỗ trợ kiến trúc microservices, giúp dễ dàng mở rộng và quản lý hệ thống lớn với nhiều thành phần độc lập.

#### ****Thư viện hỗ trợ:****

* **Spring Data JPA:**  
  Spring Data JPA đơn giản hóa việc thao tác với cơ sở dữ liệu thông qua ORM (Object Relational Mapping).
  + Hỗ trợ tự động sinh các câu lệnh SQL từ các phương thức định nghĩa sẵn như findById, save, delete,...
  + Dễ dàng kết nối với các hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL, hoặc MongoDB.
  + Hỗ trợ tích hợp Hibernate, cho phép xây dựng truy vấn phức tạp với JPQL hoặc Native Query.
* **Spring Security:**  
  Một công cụ mạnh mẽ để bảo vệ ứng dụng, đảm bảo tính bảo mật trong xác thực và phân quyền.
  + Hỗ trợ xác thực bằng username/password, OAuth2, JWT.
  + Phân quyền truy cập linh hoạt dựa trên vai trò (roles) hoặc quyền (permissions).
  + Bảo vệ API khỏi các tấn công phổ biến như CSRF, XSS, và SQL Injection.
* **Spring Web:**  
  Cung cấp các tính năng để xây dựng dịch vụ RESTful và xử lý yêu cầu HTTP.
  + Hỗ trợ @RestController, @GetMapping, @PostMapping để định nghĩa các endpoint.
  + Tích hợp dễ dàng với các thư viện bên ngoài để xử lý yêu cầu phức tạp.

#### ****Tích hợp với ReactJS:****

* **RESTful API:**  
  Spring Boot cung cấp các endpoint RESTful, giúp giao diện ReactJS lấy và gửi dữ liệu một cách dễ dàng.
  + Dữ liệu được trao đổi qua định dạng JSON.
  + Sử dụng các thư viện như Axios hoặc Fetch để gửi các yêu cầu HTTP từ ReactJS.
* **Bảo mật:**  
  Spring Security bảo vệ các API và chỉ cho phép truy cập từ các ứng dụng hoặc người dùng hợp lệ. JWT (JSON Web Token) thường được sử dụng để bảo mật các yêu cầu giữa frontend và backend.
* **Triển khai:**  
  Backend Spring Boot và frontend ReactJS có thể được triển khai riêng biệt trên các máy chủ khác nhau, hoặc gộp chung bằng cách build ReactJS thành static files và nhúng vào thư mục resources/static của Spring Boot.

#### ****Ưu điểm của kết hợp ReactJS và Spring Boot:****

* Tách biệt rõ ràng giữa giao diện người dùng và logic xử lý phía server, giúp ứng dụng dễ bảo trì và mở rộng.
* Tăng tính linh hoạt khi phát triển, vì backend và frontend có thể được xây dựng và nâng cấp độc lập.
* Dễ dàng mở rộng hệ thống để hỗ trợ các tính năng phức tạp hoặc tích hợp với các dịch vụ bên thứ ba.

## 1.4. Xác định yêu cầu của hệ thống

### 1.4.1. Mô tả hệ thống

Xây dựng 1 website bán đồ điện tử, gồm có những chức năng cần thiết cho các nhu cầu cơ bản của hoạt động mua sắm mà khách hàng và người bán cần. Bao gồm quản lý các hoạt động mua sắm của khách hàng vào các chức năng quản lý bán hàng của người bán và quản trị viên.

Người dùng sẽ được sử dụng các chức năng như đăng ký, đăng nhập, quản lý thông tin của người dùng. Ngoài ra, website còn có các chức năng khác cho khách hàng như tìm kiếm sản phẩm, xem chi tiết sản phẩm, thêm sản phẩm vào danh sách yêu thích, thêm vào giỏ hàng, đặt mua hàng,… Nhân viên và quản trị viên có thể quản lí thông tin khách hàng, quản lý đơn hàng, quản lý thông tin sản phẩm,… Quản trị viên có thể quản lý thông tin của nhân viên, quản lý các đợt giảm giá, quản lý banner của của website.

### 1.4.2. Xác định chức năng

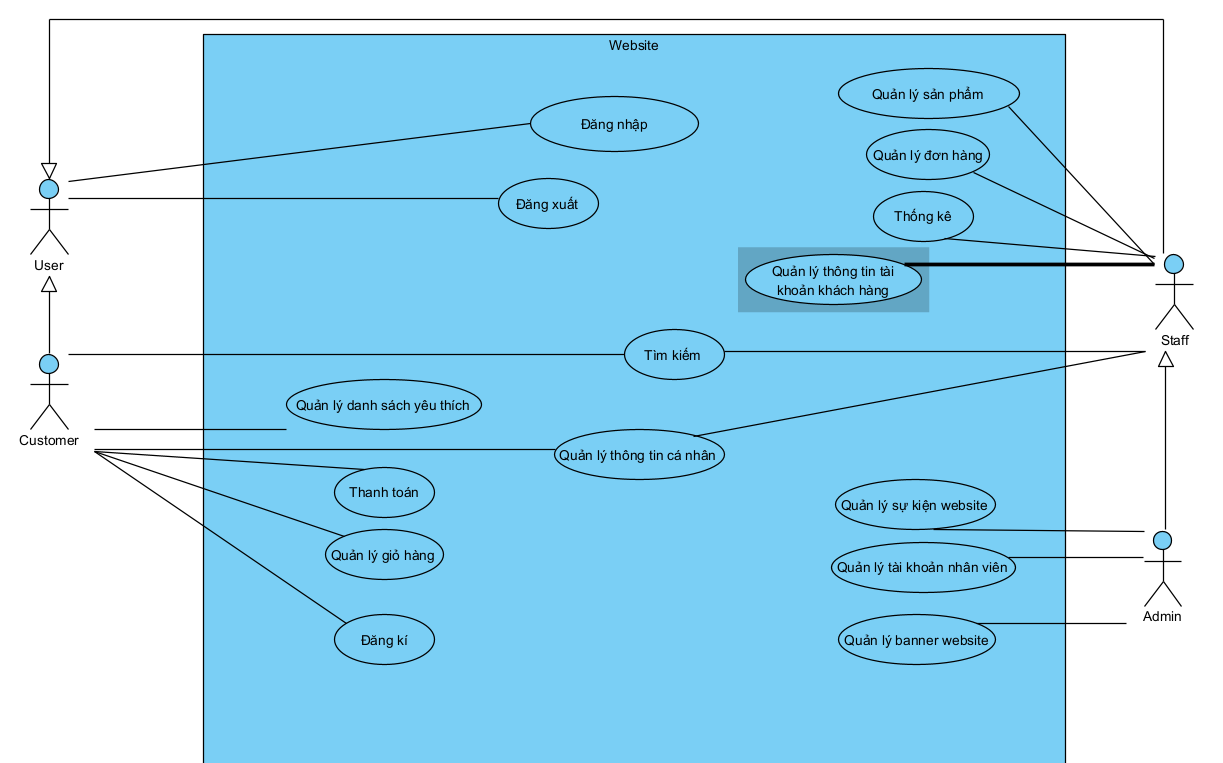
* Chức năng của người dùng:
  + Đăng nhập/ đăng ký tài khoản
  + Quản lý tài khoản (Quên mật khẩu, cập nhập thông tin tài khoản,… )
* Chức năng của nhân viên (Staff):
  + Quản lý tài khoản khách hàng
  + Quản lý đơn hàng
  + Thống kê
* Chức năng của quản trị viên (Admin) gồm chức năng của nhân viên và:
  + Quản lý tài khoản nhân viên
  + Quản lý thông tin các sự kiện trong cửa hàng (khuyến mãi,…)
  + Quản lý banner của website
* Chức năng của khách hàng
  + Quản lý thông tin tài khoản
  + Tìm kiếm sản phẩm
  + Quản lý giỏ hàng
  + Quản lý danh sách sản phẩm yêu thích
  + Đánh giá sản phẩm
  + Quản lý đơn hàng

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

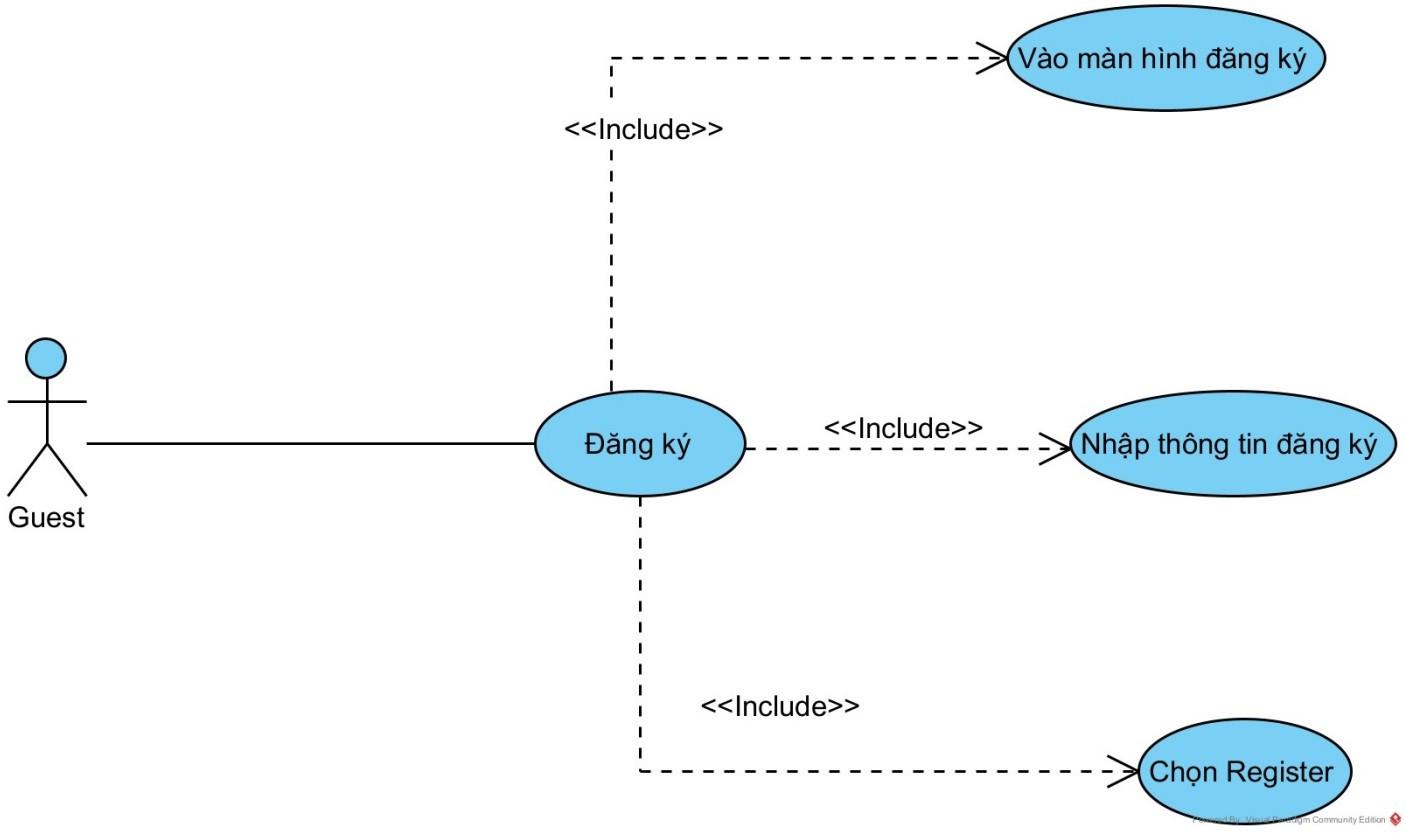
Trong chương này, hệ thống được triển khai phân tích và thiết kế theo phương pháp hướng đối tượng, áp dụng các biểu đồ theo tiêu chuẩn ngôn ngữ UML. Nội dung chính bao gồm: biểu đồ use case tổng quát, các biểu đồ phân rã use case, kịch bản, biểu đồ lớp phân tích, thiết kế mô hình lưu trữ dữ liệu, biểu đồ tuần tự và biểu đồ lớp thiết kế.

## 2.1. Biểu đồ use case và kịch bản

### 2.1.1. Biểu đồ use case tổng quát

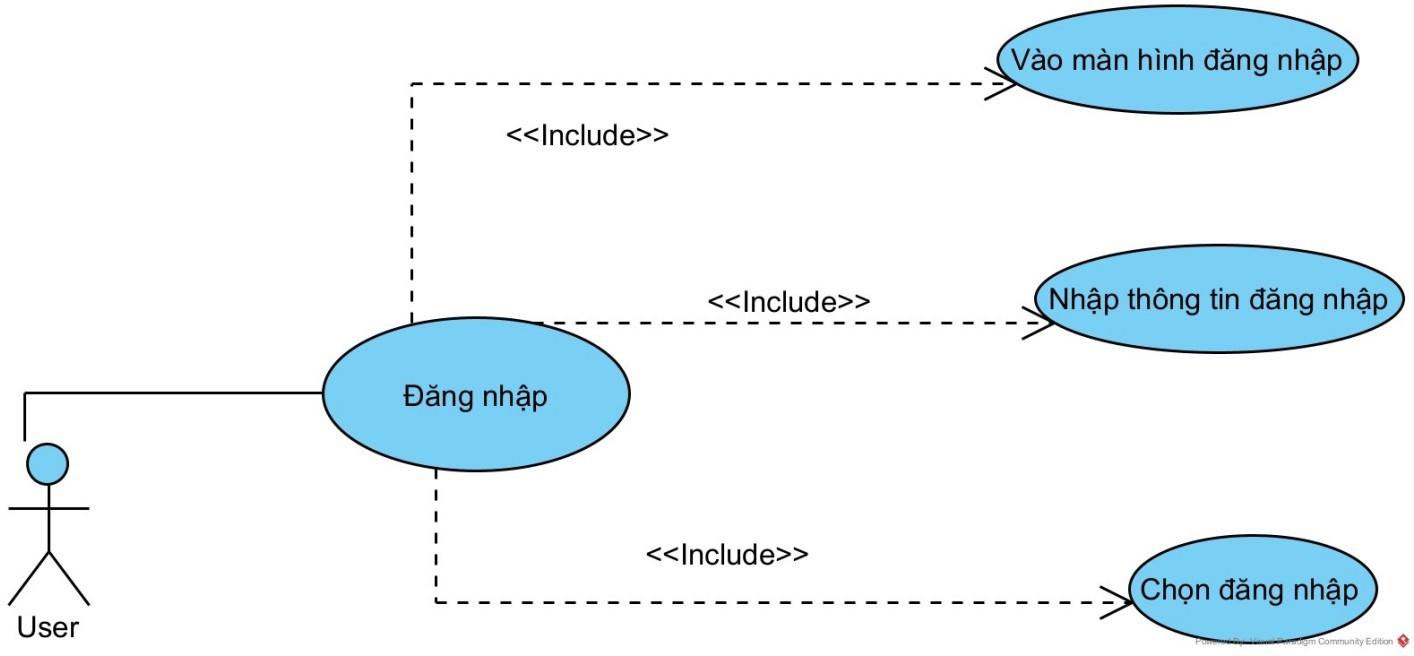
****

## Biểu đồ phân rã và kịch bản UC Đăng ký

****

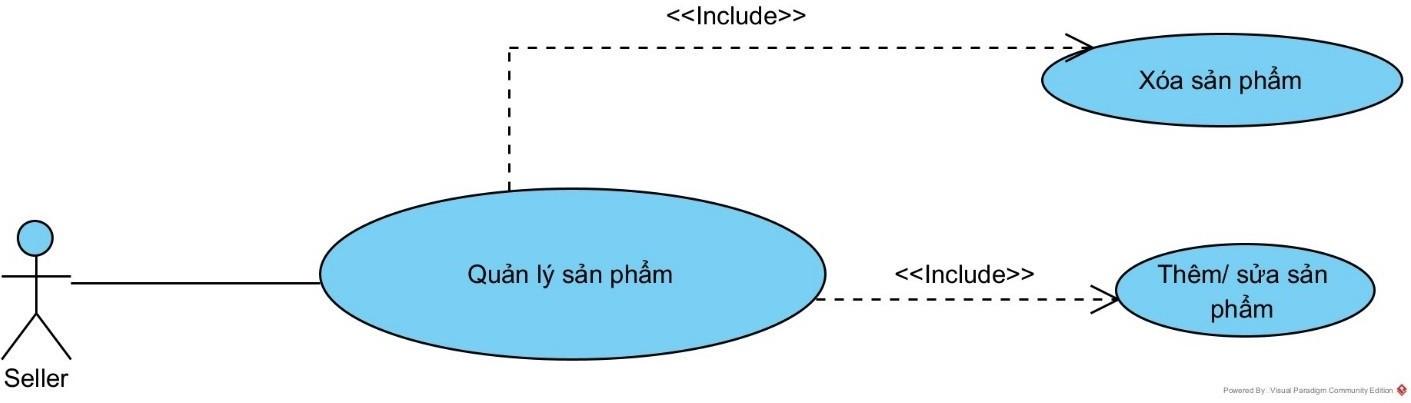
|  |  |
| --- | --- |
| **Tên Use Case** | **Đăng ký** |
| **Tác nhân chính** | Khách (Guest) |
| **Tiền điều kiện** | Khách khởi chạy hệ thống |
| **Hậu điều kiện** | Khách đăng ký tài khoản thành công. |
| **Chuỗi sự kiện chính** | 1. Khách khởi chạy hệ thống. 2. Hệ thống hiển thị trang chủ và nút đăng ký. 3. Khách bấm nút đăng ký 4. Hệ thống hiển thị màn hình đăng ký. 5. Khách nhập thông tin đăng ký (tài khoản, mật khẩu, xác thực mật khẩu) và ấn nút đăng ký 6. Hệ thống thông báo đăng ký tài khoản thành công. |
| **Ngoại lệ**   1. Hệ thống thông báo tài khoản đã tồn tại, mật khẩu và xác thực mật khẩu phải giống nhau. Và yêu cầu nhập lại    1. Quay lại bước 5 của chuỗi sự kiện chính. | |

## Biểu đồ phân rã và kịch bản UC Đăng nhập

****

|  |  |
| --- | --- |
| **Tên Use Case** | **Đăng nhập** |
| **Tác nhân chính** | Người dùng (User) |
| **Tiền điều kiện** | * Người dùng đến trang chủ. |
| **Hậu điều kiện** | * Người dùng đăng nhập tài khoản thành công. |
| **Chuỗi sự kiện chính**   1. Người dùng bấm nút đăng nhập trên trang chủ. 2. Hệ thống hiển thị màn hình đăng nhập. 3. Người dùng nhập tài khoản (usename), mật khẩu (password) 4. Hệ thống thông báo đăng nhập thành công. | |
| **Ngoại lệ**   1. Hệ thống thông báo sai tài khoản, mật khẩu. Và yêu cầu nhập lại    1. Quay lại bước 5 của chuỗi sự kiện chính. | |

## Biểu đồ phân rã và kịch bản UC quản lý sản phẩm

****

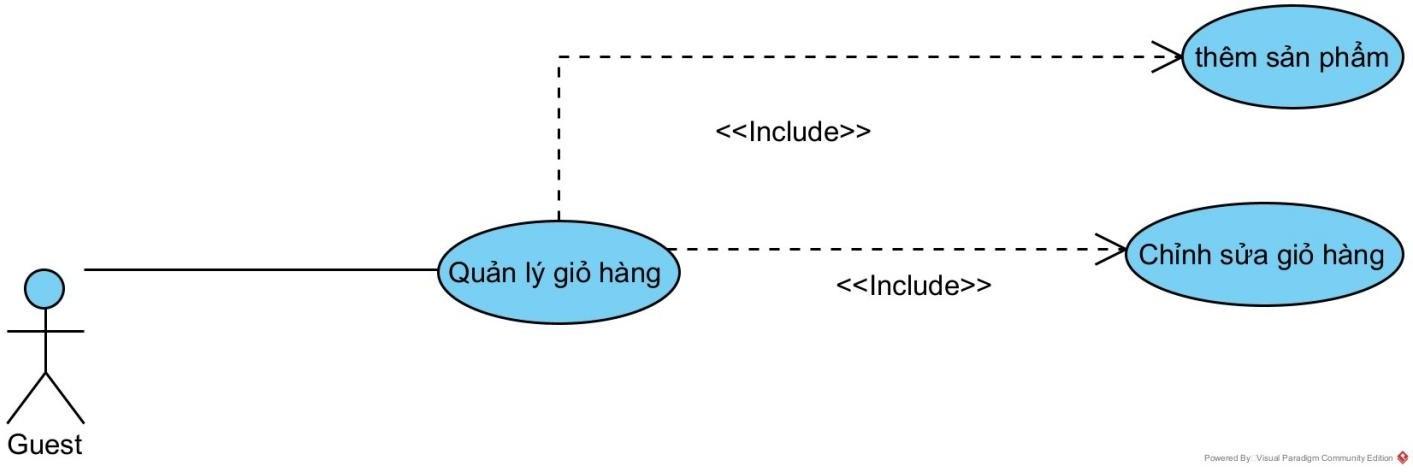
## Kịch bản UC thêm/sửa sản phẩm

|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm/sửa sản phẩm** |
| **Tác nhân chính** | Nhân viên (Staff) |
| **Tiền điều kiện** | * Đăng nhập thành công * Vào được trang quản lý sản phẩm |
| **Hậu điều kiện** | * Nhân viên thêm/ sửa sản phẩm thành công |
| **Chuỗi sự kiện chính**   1. Nhân viên nhập từ khóa trên thanh tìm kiếm và ấn tìm kiếm. 2. Hệ thống tìm kiếm các sản phẩm và hiện lên danh sách tìm kiếm. 3. Nhân viên ấn nút thêm mới sản phẩm hoặc ấn chỉnh sửa sản phẩm. 4. Hệ thống hiển thị trang điền thông tin sản phẩm. Gồm có các thông tin của sản phẩm và phần giảm giá của sản phẩm. 5. Nhân viên điền thông tin và bấm nút “save”. 6. Hệ thống kiểm tra tính đúng đắn của thông tin đã nhập. 7. Hệ thống thông báo thêm sản phẩm thành công. Sau đó lưu thông tin vào cơ sở dữ liệu. | |
| **Ngoại lệ**   1. Hệ thống thông báo nhập đầy đủ các trường và quay lại trang điền thông tin.    1. Quay lại bước 7 của chuỗi sự kiện chính. | |

## Kịch bản UC xóa sản phẩm

|  |  |
| --- | --- |
| **Tên Use Case** | **Xóa sản phẩm** |
| **Tác nhân chính** | Staff |
| **Tiền điều kiện** | * Đăng nhập thành công * Chuyển đến trang quản lý sản   phẩm |
| **Hậu điều kiện** | * Nhân viên xóa sản phẩm thành công |
| **Chuỗi sự kiện chính**   1. Nhân viên chọn 1 sản phẩm và ấn nút xóa sản phẩm 2. Hệ thống hiển thị xác nhận xóa sản phẩm 3. Nhân viên bấm xác nhận. 4. Hệ thống thông báo xóa sản phẩm thành công. | |
| **Ngoại lệ**   1. Nhân viên ấn hủy    1. Hệ thống chuyển đến trang quản lý sản phẩm | |

## Biểu đồ phân rã và kịch bản UC quản lý giỏ hàng

****

## Kịch bản UC thêm sản phẩm giỏ hàng

|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm sản phẩm giỏ hàng** |
| **Tác nhân chính** | Khách hàng |
| **Tiền điều kiện** | * Chuyển đến trang chủ. |
| **Hậu điều kiện** | * Khách thêm sản phẩm giỏ hàng thành công |
| **Chuỗi sự kiện chính** | |

**Ngoại lệ**

Hệ thống thông báo quá số lượng cho phép, vui lòng nhập lại.

Quay lại bước 3 của kịch bản chính.

Khách chọn 1 sản phẩm trên trang chủ và ấn xem chi tiết.

Hệ thống hiện thông tin sản phẩm.

Khách chỉnh sửa số lượng sản phẩm và bấm nút thêm vào giỏ hàng.

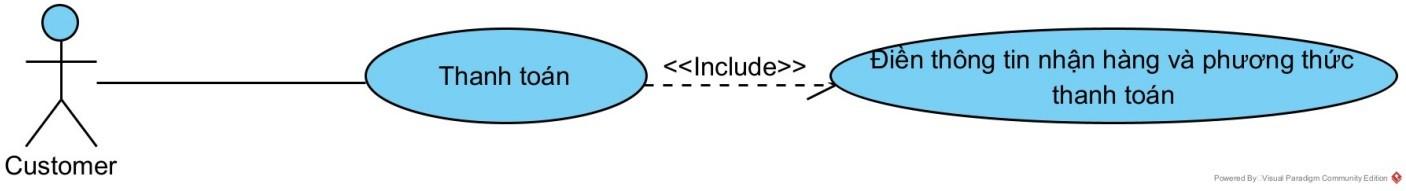
Hệ thống kiểm tra tính đúng đắn của số lượng sản phẩm giỏ hàng.

Thông báo thêm sản phẩm thành công và lưu giỏ hàng

## Kịch bản UC chỉnh sửa giỏ hàng

|  |  |
| --- | --- |
| **Tên Use Case** | **Chỉnh sửa số lượng sản phẩm** |
| **Tác nhân chính** | Khách (Guest) |
| **Tiền điều kiện** | * Chuyển đến trang chủ. |
| **Hậu điều kiện** | * Khách chỉnh sửa giỏ hàng thành công |
| **Kích hoạt** |  |
| **Chuỗi sự kiện chính**   1. Khách ấn nút giỏ hàng trên trang chủ. 2. Hệ thống chuyển đến trang giỏ hàng và hiện thông tin giỏ hàng. 3. Khách chỉnh sửa sản phẩm của giỏ hàng (sửa số lượng hoặc xóa sản phẩm). 4. Hệ thống thông báo chỉnh sửa sản phẩm thành công. | |
| **Ngoại lệ**   1. hệ thống thông báo số lượng sản phẩm không phù hợp.    1. quay lại bước 1 trong chuỗi sự kiện chính. | |

## Biểu đồ phân rã và kịch bản UC thanh toán

****

|  |  |
| --- | --- |
| **Tên Use Case** | **Thanh toán** |
| **Tác nhân chính** | Khách hàng (Customer) |
| **Tiền điều kiện** | * Đăng nhập thành công * Chuyển đến trang giỏ hàng |
| **Hậu điều kiện** | * Khách hàng thanh toán thành công |

**Ngoại lệ**

3. Khách hàng điền địa chỉ nhận hàng và chọn phương thức thanh toán là thanh toán trực tuyến

Hệ thống chuyển đến trang thanh toán của zaloPay.

Khách hàng tiến hành thanh toán

Quay trở lại bước 4 của chuỗi sự kiện chính.

**Chuỗi sự kiện chính**

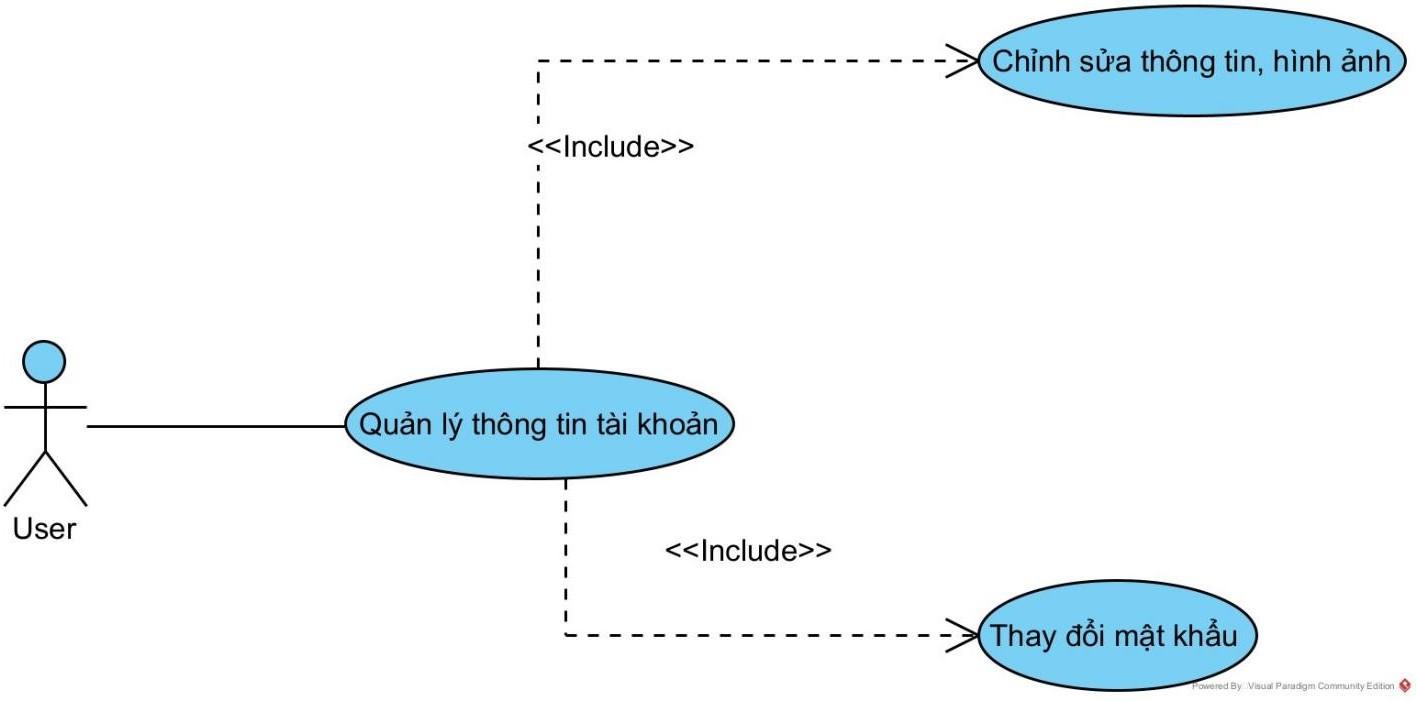
Khách hàng ấn thanh toán trên trang giỏ hàng

Hệ thống hiển thị trang điền thông tin thanh toán, gồm có địa chỉ nhận hàng và phương thức thanh toán

Khách hàng điền địa chỉ nhận hàng và chọn phương thức thanh toán là Thanh toán khi nhận hàng.

Hệ thống lưu trạng thái của đơn hàng và chuyển về trang danh sách đã mua.

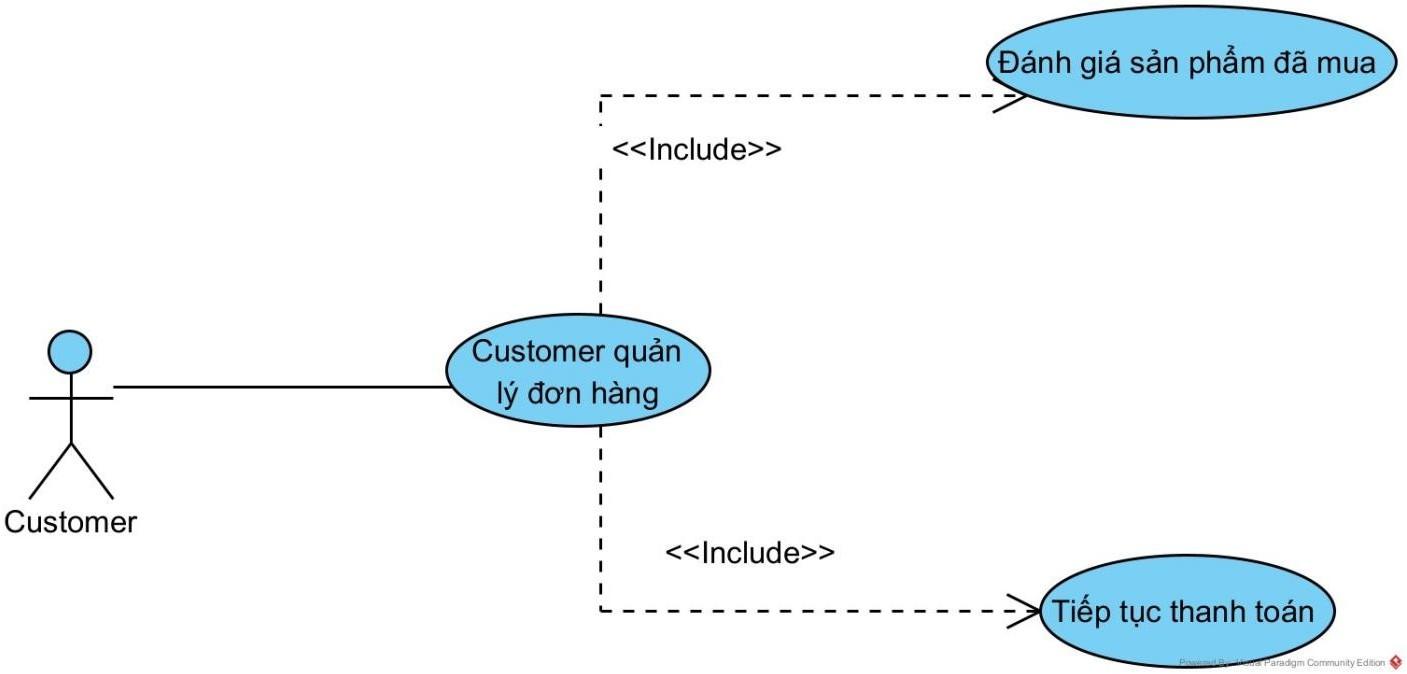
## Biểu đồ phân rã và kịch bản UC Quản lý thông tin tài khoản

****

|  |  |
| --- | --- |
| **Tác nhân chính** | Người dùng (User) |
| **Tiền điều kiện** | * Đăng nhập thành công * Chuyển đến trang chủ |
| **Hậu điều kiện** | * Người dùng chỉnh sửa thông tin tài khoản thành công |
| **Chuỗi sự kiện chính**   1. Khách hàng ấn vào nút thông tin tài khoản. 2. Hệ thống hiển thị thông tin của người dùng, gồm có thông tin, hình ảnh của người dùng, thông tin mật khẩu của người dùng 3. Người dùng chỉnh sửa thông tin và lưu 4. Hệ thống lưu các thông tin đã chỉnh sửa. | |
| **Ngoại lệ**   1. Hệ thống thông báo nhập đầy đủ các trường cần thiết.    1. Quay lại bước 3 của chuỗi sự kiện chính   4. Hệ thống thông báo mật khẩu hiện tại không đúng. | |

4.1. Quay lại bước 3a của chuỗi sự kiện chính

## Biểu đồ phân rã và kịch bản UC Khách hàng quản lý đơn hàng

****

## Kịch bản UC tiếp tục thanh toán.

|  |  |
| --- | --- |
| **Tác nhân chính** | Khách hàng(Customer) |
| **Tiền điều kiện** | * Đăng nhập thành công * Chuyển đến trang chủ |
| **Hậu điều kiện** | * Khách hàng thanh toán thành công |
| **Chuỗi sự kiện chính**   1. Khách hàng chọn nút quản lý đơn hàng trên trang chủ 2. Hệ thống hiện giao diện quản lý đơn hàng với danh sách đơn hàng 3. Khách hàng chọn 1 đơn hàng chưa thanh toán và ấn nút xem chi tiết. 4. Hệ thống hiện trang chi tiết đơn hàng với các sản phẩm của đơn hàng và nút tiếp tục thanh toán. 5. Khách hàng ấn nút tiếp tục thanh toán. 6. Hệ thống chuyển đến trang thanh toán của ZaloPay. 7. Khách hàng tiến hành thanh toán. 8. Hệ thống cập nhật trạng thái đơn hàng thành đã thanh toán và quay trở lại trang quản lý đơn hàng. | |
| **Ngoại lệ**  6. Hệ thống chuyển đến trang thanh toán của ZaloPay và hiện thông báo quá thời gian thanh toán | |

Khách hàng chọn quay lại.

Hệ thống cập nhật trạng thái đơn hàng thành Hủy và quay trở lại trang quản lý đơn hàng.

Hệ thống chuyển đến trang chi tiết đơn hàng và thông báo đơn hàng đã được thanh toán.

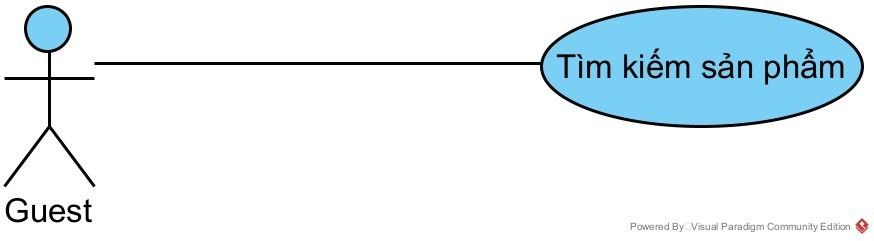
Khách hàng ấn nút quay lại.

Hệ thống chuyển đến trang quản lý đơn hàng.

## Kịch bản UC đánh giá sản phẩm đã mua

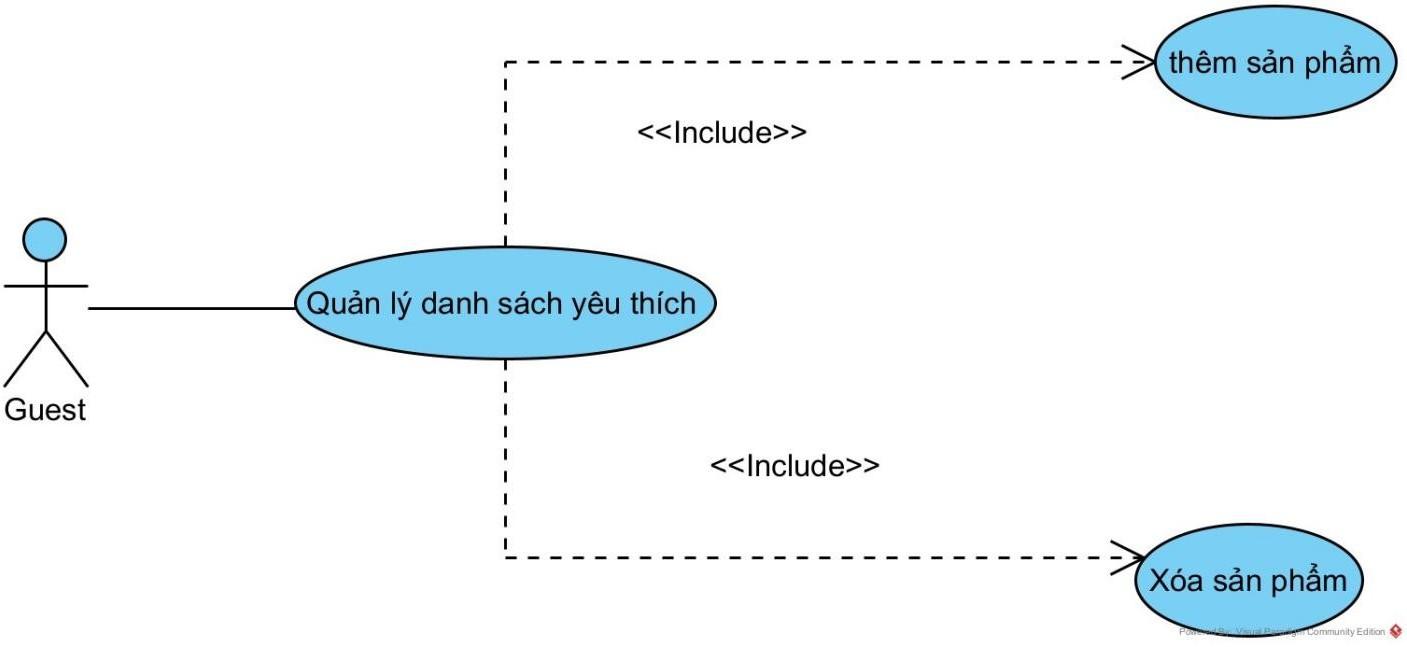
|  |  |
| --- | --- |
| **Tác nhân chính** | Khách hàng |
| **Tiền điều kiện** | * Khách hàng đăng nhập thành công * Chuyển đến trang quản lý đơn   hàng. |
| **Hậu điều kiện** | * Khách hàng đánh giá sản phẩm thành công. |
| **Chuỗi sự kiện chính**   1. Khách hàng chọn 1 đơn hàng và ấn xem chi tiết. 2. Hệ thống chuyển đến trang chi tiết đơn hàng, mỗi sản phẩm của đơn hàng có nút đánh giá. 3. Khách hàng chọn 1 sản phẩm và ấn nút đánh giá. 4. Hệ thống hiện phần đánh giá sản phẩm. 5. Khách hàng tiến hành đánh giá sản phẩm và ấn lưu đánh giá sản phẩm. 6. Hệ thống lưu đánh giá sản phẩm của khách hàng và thông báo đánh giá sản phẩm thành công. | |
| **Ngoại lệ**   1. Hệ thống thông báo “chưa nhập đầy đủ các trường”.    1. Quay lại bước 5 của chuỗi sự kiện chính. | |

## Biểu đồ phân rã và kịch bản UC Tìm kiếm



|  |  |
| --- | --- |
| **Tác nhân chính** | Khách (Guest) |
| **Tiền điều kiện** | * Chuyển đến trang chủ |
| **Hậu điều kiện** | * Khách tìm kiếm sản phẩm thành công |
| **Chuỗi sự kiện chính**   1. Khách hàng chọn tìm kiếm trên trang chủ. 2. Hệ thống hiển thị giao diện tìm kiếm. 3. Khách chọn thông tin sản phẩm muốn tìm kiếm. Gồm tên, loại sản phẩm, vị trí cửa hàng, khoảng giá, độ đánh giá của sản phẩm. 4. Hệ thống hiện các sản phẩm theo thông tin khách đã chọn. | |
| **Ngoại lệ** | |

## Biểu đồ phân rã và kịch bản UC quản lý danh sách yêu thích



## Kịch bản UC thêm sản phẩm vào danh sách yêu thích

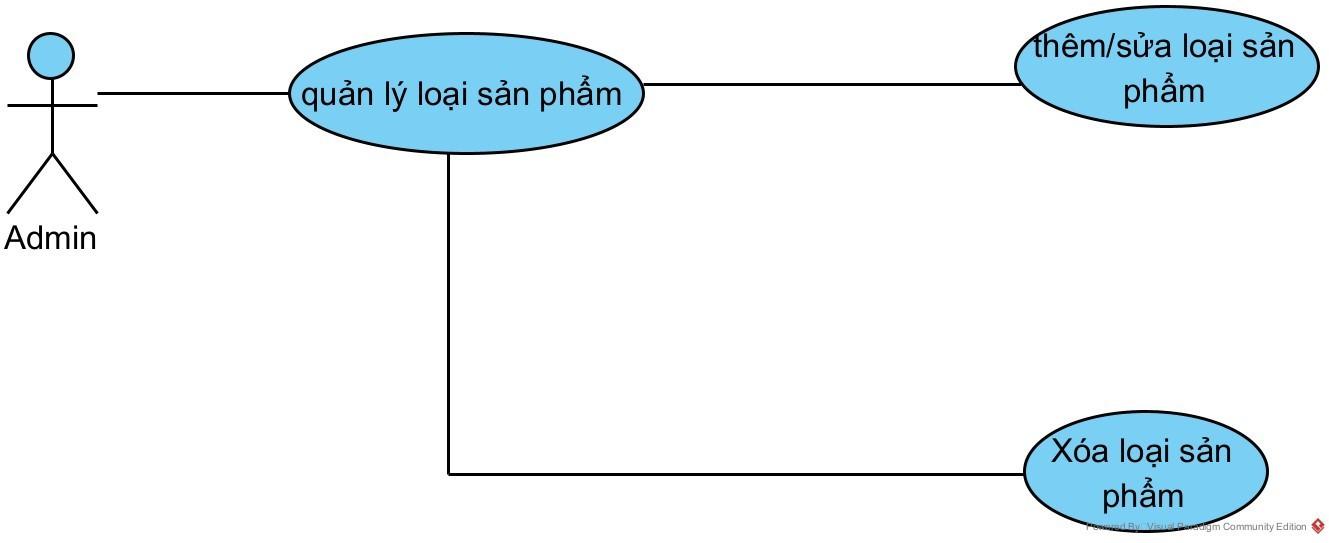
|  |  |
| --- | --- |
| **Tên Use Case** | **Thêm sản phẩm** |
| **Tác nhân chính** | Khách hàng (Guest) |
| **Tiền điều kiện** | * Chuyển đến trang thông tin chi tiết sản phẩm |
| **Hậu điều kiện** | * Khách hàng thêm sản phẩm vào danh sách yêu thích thành công |
| **Chuỗi sự kiện chính**   1. Khách hàng bấm nút thêm vào danh sách yêu thích. 2. Hệ thống thông báo thêm sản phẩm thành công. | |
| **Ngoại lệ** | |

## Kịch bản UC xóa sản phẩm yêu thích

|  |  |
| --- | --- |
| **Tên Use Case** | **Xóa sản phẩm giỏ hàng** |
| **Tác nhân chính** | Khách hàng (Guest) |
| **Tiền điều kiện** | * Chuyển đến trang chủ |
| **Hậu điều kiện** | * Khách hàng xóa sản phẩm yêu thích thành công |
| **Chuỗi sự kiện chính**   1. Khách hàng ấn nút Wishlist trên trang chủ. 2. Hệ thống hiển thị trang sản phẩm yêu thích với các sản phẩm trong giỏ hàng. 3. Khách hàng chọn 1 sản phẩm và ấn nút xóa sản phẩm. 4. Hệ thống thông báo xóa sản phẩm thành công. | |

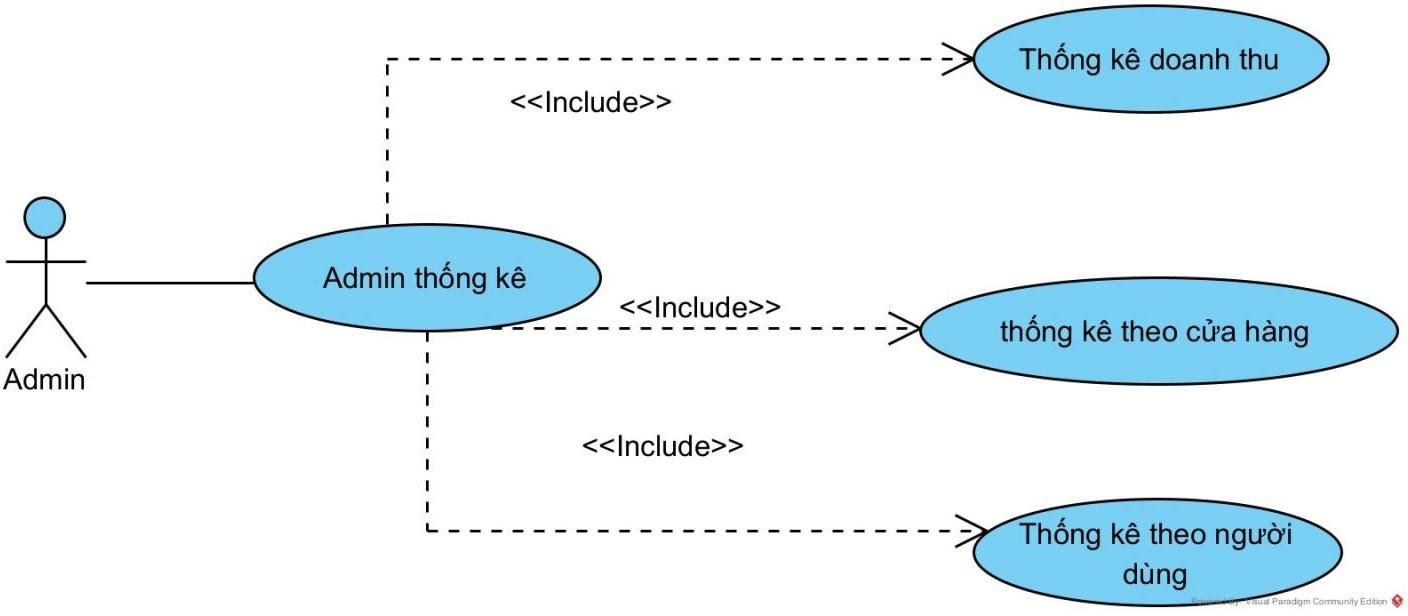
**Ngoại lệ**

## Biểu đồ phân rã và kịch bản UC quản lý sản phẩm

****

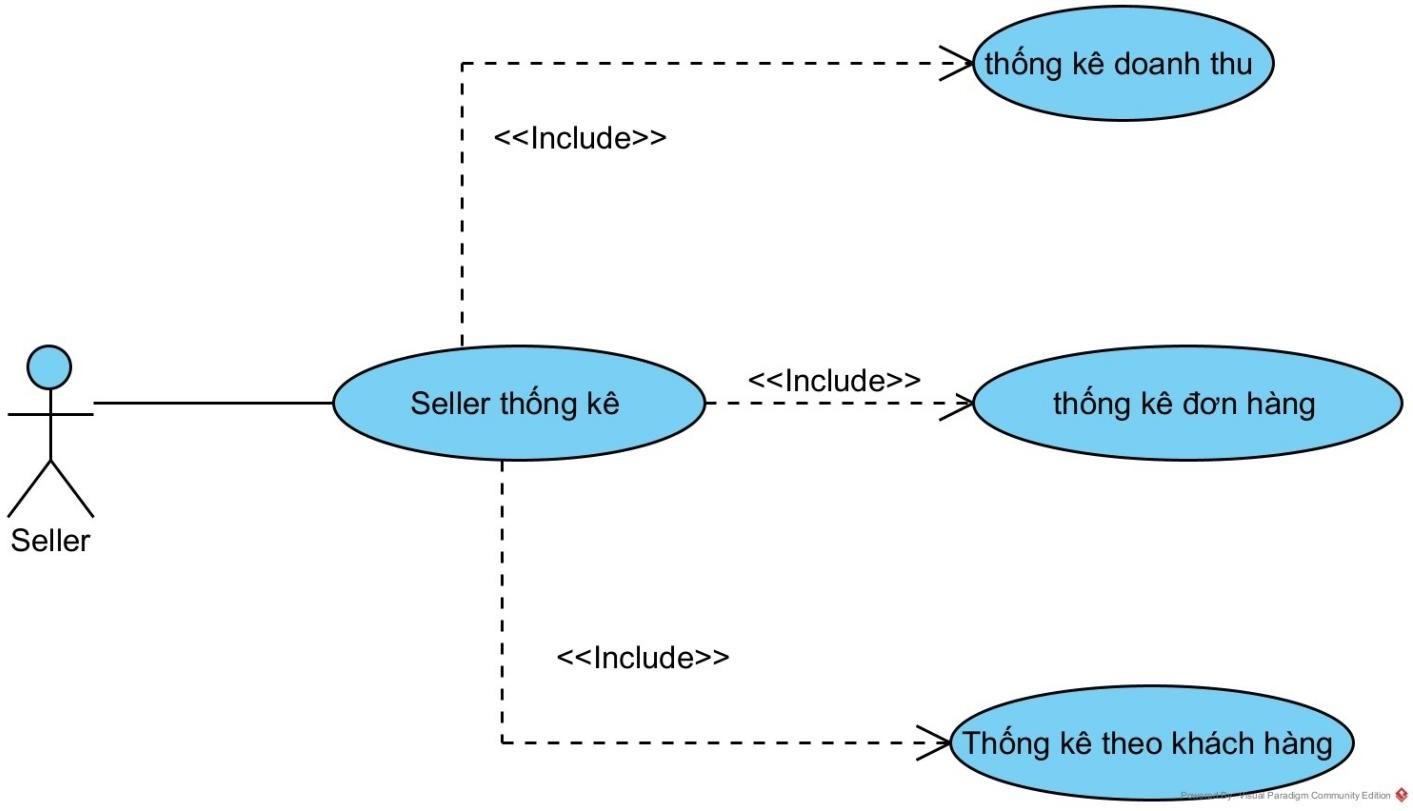
|  |  |
| --- | --- |
| **Tên Use Case** | **Quản lý loại sản phẩm** |
| **Tác nhân chính** | Quản trị viên (Admin) |
| **Tiền điều kiện** | * Đăng nhập tài khoản admin thành công * Vào được trang quản lý loại sản phẩm |
| **Hậu điều kiện** | * Nhân viên thêm/ sửa/ xóa loại sản phẩm thành công |
| **Chuỗi sự kiện chính**   1. Quản trị viên ấn nút thêm/ sửa/ xóa loại sản phẩm trên trang quản lý loại sản phẩm. 2. Hệ thống hiển thị trang thông tin loại sản phẩm. 3. Nhân viên thêm/ sửa/ xóa loại sản phẩm và lưu 4. Hệ thống thông báo thêm/ sửa/ xóa loại sản phẩm thành công. | |
| **Ngoại lệ**   1. Hệ thống thông báo nhập đầy đủ các trường hoặc tên loại sản phẩm đã tồn tại, sau đó quay lại trang điền thông tin.    1. Quay lại bước 3 của chuỗi sự kiện chính. | |

## Biểu đồ phân rã và kịch bản UC admin thống kê



|  |  |
| --- | --- |
| **Tác nhân chính** | Quản trị viên |
| **Tiền điều kiện** | * Đăng nhập tài khoản quản trị viên thành công * Chuyển đến trang quản lý của quản trị viên. |
| **Hậu điều kiện** | * Quản trị viên xem được thống kê báo cáo. |
| **Chuỗi sự kiện chính**   1. Quản trị viên chọn chức năng thống kê trên trang quản lý. 2. Hệ thống hiện trang thống kê của quản trị viên. Gồm có nút chọn ngày bắt đầu và kết thúc. 3. Quản trị viên chọn thời gian muốn thống kê và ấn thống kê. 4. Hệ thống thống kê dữ liệu doanh thu, thống kê theo cửa hàng, thống kê theo người dùng. 5. Hệ thống hiện dữ liệu cho quản trị viên. | |
| **Ngoại lệ** | |

## Biểu đồ phân rã và kịch bản UC staff thống kê



|  |  |
| --- | --- |
| **Tác nhân chính** | staff |
| **Tiền điều kiện** | * Đăng nhập tài khoản của nhân viên thành công * Chuyển đến trang quản lý của nhân viên. |
| **Hậu điều kiện** | * Nhân viên xem được thống kê báo cáo. |
| **Chuỗi sự kiện chính**   1. Nhân viên chọn chức năng thống kê. 2. Hệ thống chuyển đến trang thống kê của nhân viên. Gồm có ô chọn thời gian bắt đầu và kết thúc. 3. Nhân viên nhập thời gian thống kê và ấn thống kê. 4. Hệ thống thống kê doanh thu, thống kê số lượng đơn hàng, thống kê theo khách hàng của của hàng. 5. Hệ thống hiển thị thông tin thống kê. | |
| **Ngoại lệ** | |

## Xây dựng biểu đồ lớp

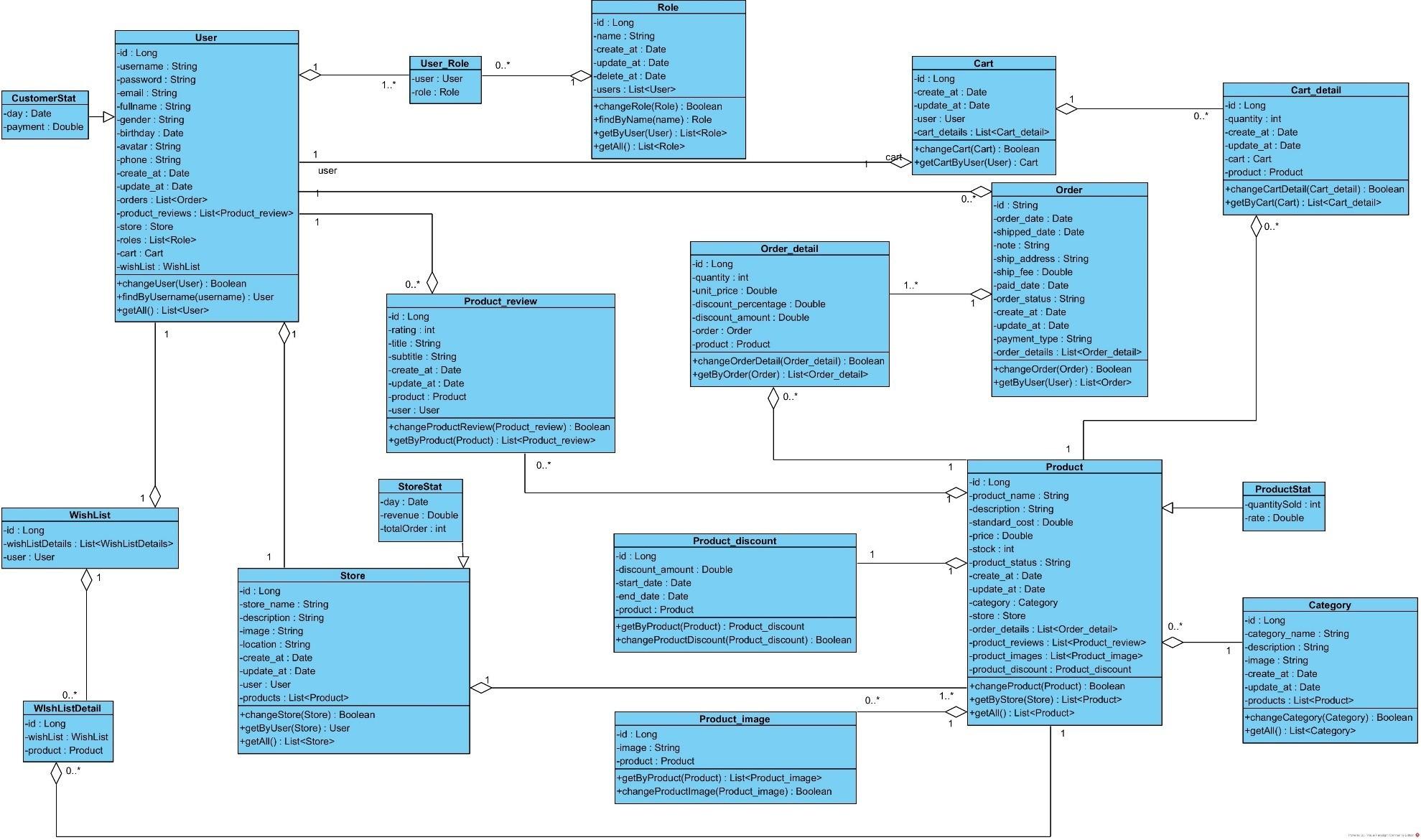
* Trích xuất danh từ
  + Danh từ liên quan đến người: Khách, Người dùng, Khách hàng, Quản trị viên.
  + Danh từ liên quan đến thông tin: hệ thống, web sàn thương mại điện tử, tài khoản, quyền, sản phẩm, giỏ hàng, loại sản phẩm, giá, cửa hàng, vị trí cửa hàng, danh sách yêu thích, đơn hàng.
* Đánh giá, lựa chọn danh từ, xác định lớp thực thể hoặc thuộc tính
  + Danh từ trừu tượng: hệ thống, web sàn thương mại điện tử
  + Danh từ liên quan đến người:
    - Khách: không cần quản lý -> bỏ.
    - Người dùng -> lớp User.
    - Khách hàng, quản trị hệ thống: là 1 loại của User
  + Danh từ thông tin:
    - Tài khoản: là thông tin của lớp User.
    - Quyền -> lớp Role.
    - Sản phẩm -> lớp Product.
    - Giỏ hàng -> lớp Cart.
    - Sản phẩm giỏ hàng -> lớp CartDetail
    - Loại sản phẩm -> lớp Category
    - Giá: là thông tin của lớp Product.
    - Danh sách yêu thích -> lớp WishList.
    - Đơn hàng-> lớp Order.
    - Sản phẩm của đơn hàng -> lớp OrderDetail.
    - Thống kê chi tiêu của khách hàng -> lớp CustomerStat.
* Quan hệ số lượng giữa các thực thể
  + 1 Product có trong nhiều Order, 1 Order có nhiều Product. Nên mối quan hệ là n-n. Chúng ta sẽ tạo 1 lớp giữa chúng là Order\_detail.
  + 1 Product có nhiều Order\_detail, 1 Order\_detail thuộc 1 Product. Nên mối quan hệ Product- Order\_detail là 1-n.
  + 1 Order có nhiều Order\_detail, 1 Order\_detail thuộc 1 Order. Nên mối quan hệ Order - Order\_detail là 1-n.
  + 1 User thì sẽ đánh giá nhiều Product và 1 Product sẽ được đánh giá bởi nhiều User. Nên mối quan hệ User- Product là n-n. Chúng ta sẽ tạo 1 lớp thực thể giữa chúng là: Product\_review.
  + 1 Product có nhiều Product\_review, 1 Product\_review thuộc 1 Product. Nên mối quan hệ Product - Product\_review là 1-n.
  + 1 User có nhiều Product\_review, 1 Product\_review thuộc 1 User. Nên mối quan hệ User- Product\_review là 1-n.
  + 1 User thì có thể đặt nhiều Order và 1 Order sẽ thuộc 1 User. Nên mối quan hệ User- Order là 1-n.
  + 1 User thì có 1 Cart và 1 Cart sẽ thuộc 1 User. Nên mối quan hệ User-Cart là 1-1.
  + 1 User có 1 WishList và 1 WishList thuộc 1 User. Nên mối quan hệ User-WishList là 1-1.
  + 1 Product có thuộc 1 Category, 1 Category có nhiều Product. Nên mối quan hệ Category-Product là 1-n.
  + 1 Product có trong nhiều Cart, 1 Cart có nhiều Product. Nên mối quan hệ là n-n. Chúng ta sẽ tạo 1 lớp giữa chúng là Cart\_detail.
  + 1 Product thuộc nhiều Cart\_detail. 1 Cart\_detail có 1 Product. Nên mối quan hệ Product- Cart\_detail là 1-n.
  + 1 Product có trong nhiều WishList. 1 WishList có nhiều Product. Mối quan hệ là n-

n. Chúng ta sẽ tạo 1 lớp giữa chúng là WishListDetail.

* + 1 Product thuộc nhiều WishListDetail. 1 WishListDetail có 1 Product. Mối quan hệ giữa Product- WishListDetail là 1-n.
  + 1 Cart có nhiều Cart\_detail. 1 Cart\_detail thuộc 1 Cart. Nên mối quan hệ Cart - Cart\_detail là 1-n.
  + 1 WishList có nhiều WishListDetail. 1 WishListDetail thuộc 1 WishList. Nên mối quan hệ WishList- WishListDetail là 1-n.

Ngoài các thực thể kể trên, đồ án đề xuất 1 số lớp chứa các thuộc tính dẫn xuất hoặc giống như thuộc tính của các thực thể để tạo điều kiện thuận lợi cho việc lấy dữ liệu và hiện thị giao diện. Các lớp này bao gồm:

* + Lớp Product\_image để lưu trữ những hình ảnh của sản phẩm
    - 1 Product sẽ có nhiều Product\_image, 1 Product\_image sẽ thuộc 1 Product. Nên mối quan hệ Product- Product\_image là 1-n.
  + Lớp Product\_discount để lưu trữ thông tin giảm giá
    - 1 Product sẽ có 1 Product\_discount, 1 Product\_discount sẽ thuộc 1 Product. Nên mối quan hệ Product- Product\_discount là 1-1.
* Biểu đồ lớp thực thể chi tiết cho hệ thống



### 1. Lớp CustomerStat

* **Định nghĩa**: Thống kê chi tiêu của khách hàng theo ngày.
* **Thuộc tính**:
  + day: Ngày thống kê.
  + payment: Tổng số tiền chi tiêu của khách hàng trong ngày đó.
* **Phương thức**:
  + getStatsByDay(Date day): List<CustomerStat> – Lấy dữ liệu thống kê chi tiêu theo ngày.
  + addStat(CustomerStat stat): Boolean – Thêm dữ liệu thống kê mới.
  + updateStat(CustomerStat stat): Boolean – Cập nhật dữ liệu thống kê chi tiêu.
  + deleteStat(Date day): Boolean – Xóa dữ liệu thống kê theo ngày.

### 3. Lớp ProductStat

* **Định nghĩa**: Thống kê số lượng bán và đánh giá sản phẩm.
* **Thuộc tính**:
  + productstat\_id: Mã định danh thống kê.
  + quantity\_sold: Số lượng sản phẩm đã bán.
  + rate: Tỷ lệ đánh giá trung bình.
* **Phương thức**:
  + getStatByProduct(Product product): ProductStat – Lấy thống kê theo sản phẩm.
  + addStat(ProductStat stat): Boolean – Thêm dữ liệu thống kê mới.
  + updateStat(ProductStat stat): Boolean – Cập nhật dữ liệu thống kê.
  + deleteStat(String productstat\_id): Boolean – Xóa dữ liệu thống kê sản phẩm.

### 4. Lớp Cart\_detail

* **Định nghĩa**: Thông tin sản phẩm trong giỏ hàng.
* **Thuộc tính**:
  + cart\_detail\_id: Mã định danh.
  + quantity: Số lượng sản phẩm.
  + create\_at: Ngày tạo.
  + update\_at: Ngày cập nhật.
  + cart: Thông tin giỏ hàng liên kết.
  + product: Thông tin sản phẩm liên kết.
* **Phương thức**:
  + changeCartDetail(Cart\_detail detail): Boolean – Thêm, sửa, xóa sản phẩm trong giỏ hàng.
  + getByCart(Cart cart): List<Cart\_detail> – Lấy danh sách sản phẩm trong giỏ hàng.

### 5. Lớp Cart

* **Định nghĩa**: Thông tin về giỏ hàng.
* **Thuộc tính**:
  + cart\_id: Mã định danh.
  + create\_at: Ngày tạo.
  + update\_at: Ngày cập nhật.
  + customer: Thông tin khách hàng.
  + cart\_details: Danh sách sản phẩm trong giỏ.
* **Phương thức**:
  + changeCart(Cart cart): Boolean – Thêm, sửa giỏ hàng.
  + getCartByCustomer(Customer customer): Cart – Lấy giỏ hàng của khách hàng.

### 6. Lớp Category

1. **Định nghĩa**: Thông tin loại sản phẩm.
2. **Thuộc tính**:
   1. category\_id: Mã định danh.
   2. category\_name: Tên loại sản phẩm.
   3. description: Mô tả loại sản phẩm.
   4. image: Hình ảnh sản phẩm.
   5. create\_at: Ngày tạo.
   6. update\_at: Ngày cập nhật.
   7. products: Danh sách sản phẩm trong loại.
3. **Phương thức**:
   1. changeCategory(Category category): Boolean – Thêm, sửa loại sản phẩm.
   2. getAll(): List<Category> – Lấy tất cả các loại sản phẩm.

### 7. Lớp Order\_detail

* **Định nghĩa**: Thông tin sản phẩm trong đơn hàng.
* **Thuộc tính**:
  + order\_detail\_id: Mã định danh.
  + quantity: Số lượng sản phẩm.
  + unit\_price: Đơn giá sản phẩm.
  + discount\_percentage: Phần trăm giảm giá.
  + discount\_amount: Số tiền giảm giá.
  + order: Thông tin đơn hàng.
  + product: Thông tin sản phẩm.
* **Phương thức**:
  + changeOrderDetail(Order\_detail detail): Boolean – Thêm, sửa sản phẩm trong đơn hàng.
  + getByOrder(Order order): List<Order\_detail> – Lấy danh sách sản phẩm trong đơn hàng.

### 8. Lớp Order

* **Định nghĩa**: Thông tin về đơn hàng.
* **Thuộc tính**:
  + order\_id: Mã định danh.
  + order\_date: Ngày đặt hàng.
  + shipped\_date: Ngày giao hàng.
  + note: Ghi chú.
  + ship\_address: Địa chỉ giao hàng.
  + ship\_fee: Phí giao hàng.
  + paid\_date: Ngày thanh toán.
  + order\_status: Trạng thái đơn hàng.
  + payment\_type: Phương thức thanh toán.
  + customer: Thông tin khách hàng.
  + order\_details: Danh sách sản phẩm trong đơn hàng.
* **Phương thức**:
  + changeOrder(Order order): Boolean – Thêm, sửa đơn hàng.
  + getByCustomer(Customer customer): List<Order> – Lấy danh sách đơn hàng theo khách hàng.

### 9. Lớp Product

* **Định nghĩa**: Thông tin sản phẩm.
* **Thuộc tính**:
  + product\_id: Mã định danh.
  + product\_name: Tên sản phẩm.
  + description: Mô tả sản phẩm.
  + standard\_cost: Giá gốc.
  + price: Giá bán.
  + stock: Số lượng sản phẩm.
  + product\_status: Trạng thái sản phẩm.
  + category: Thông tin loại sản phẩm.
  + product\_discounts: Thông tin giảm giá.
  + product\_images: Danh sách hình ảnh sản phẩm.
  + product\_reviews: Danh sách đánh giá.
* **Phương thức**:
  + changeProduct(Product product): Boolean – Thêm, sửa, xóa sản phẩm.
  + getByCategory(Category category): List<Product> – Lấy danh sách sản phẩm theo loại.
  + getAll(): List<Product> – Lấy tất cả sản phẩm.

### 10. Lớp Customer

* **Định nghĩa**: Thông tin khách hàng.
* **Thuộc tính**:
  + customer\_id: Mã định danh.
  + username: Tên đăng nhập.
  + password: Mật khẩu.
  + fullname: Họ và tên.
  + avatar: Ảnh đại diện.
  + phone: Số điện thoại.
  + orders: Danh sách đơn hàng của khách hàng.
* **Phương thức**:
  + changeCustomer(Customer customer): Boolean – Thêm, sửa, xóa khách hàng.
  + findByUsername(String username): Customer – Tìm khách hàng theo tên đăng nhập.
  + getAll(): List<Customer> – Lấy tất cả khách hàng.

11. Lớp Staff

* Định nghĩa: Thông tin về nhân viên.
* Thuộc tính:
  + staff\_id: Mã định danh nhân viên.
  + username: Tên đăng nhập.
  + password: Mật khẩu.
  + fullname: Họ và tên.
  + email: Email liên hệ.
  + phone: Số điện thoại.
  + role: Vai trò của nhân viên (như quản lý, nhân viên bán hàng, v.v.).
  + status: Trạng thái hoạt động của nhân viên (đang làm việc, nghỉ việc, v.v.).
* Phương thức:
  + changeStaff(Staff staff): Boolean – Thêm, sửa, xóa nhân viên.
  + findByUsername(String username): Staff – Tìm nhân viên theo tên đăng nhập.
  + getAll(): List<Staff> – Lấy tất cả nhân viên.

12. Lớp Banner

* Định nghĩa: Thông tin về banner quảng cáo.
* Thuộc tính:
  + banner\_id: Mã định danh banner.
  + title: Tiêu đề banner.
  + image: Đường dẫn ảnh của banner.
  + create\_at: Ngày tạo banner.
  + update\_at: Ngày cập nhật banner.
  + status: Trạng thái hoạt động của banner (hiển thị hoặc ẩn).
* Phương thức:
  + addBanner(Banner banner): Boolean – Thêm banner mới.
  + updateBanner(Banner banner): Boolean – Cập nhật thông tin banner.
  + deleteBanner(String banner\_id): Boolean – Xóa banner theo mã định danh.
  + getAll(): List<Banner> – Lấy tất cả các banner.

13. Lớp Event

* Định nghĩa: Thông tin về sự kiện.
* Thuộc tính:
  + event\_id: Mã định danh sự kiện.
  + title: Tiêu đề sự kiện.
  + description: Mô tả sự kiện.
  + start\_date: Ngày bắt đầu sự kiện.
  + end\_date: Ngày kết thúc sự kiện.
  + status: Trạng thái hoạt động của sự kiện (đang diễn ra, đã kết thúc).
* Phương thức:
  + addEvent(Event event): Boolean – Thêm sự kiện mới.
  + updateEvent(Event event): Boolean – Cập nhật thông tin sự kiện.
  + deleteEvent(String event\_id): Boolean – Xóa sự kiện theo mã định danh.
  + getAll(): List<Event> – Lấy tất cả các sự kiện.

14. Lớp ProductDiscount

* Định nghĩa: Thông tin giảm giá của sản phẩm.
* Thuộc tính:
  + productdiscount\_id: Mã định danh giảm giá.
  + discount\_percentage: Phần trăm giảm giá.
  + start\_date: Ngày bắt đầu giảm giá.
  + end\_date: Ngày kết thúc giảm giá.
  + product: Sản phẩm được áp dụng giảm giá.
* Phương thức:
  + addProductDiscount(ProductDiscount discount): Boolean – Thêm giảm giá mới cho sản phẩm.
  + updateProductDiscount(ProductDiscount discount): Boolean – Cập nhật thông tin giảm giá.
  + deleteProductDiscount(String productdiscount\_id): Boolean – Xóa thông tin giảm giá theo mã định danh.
  + getDiscountByProduct(Product product): List<ProductDiscount> – Lấy các giảm giá theo sản phẩm.

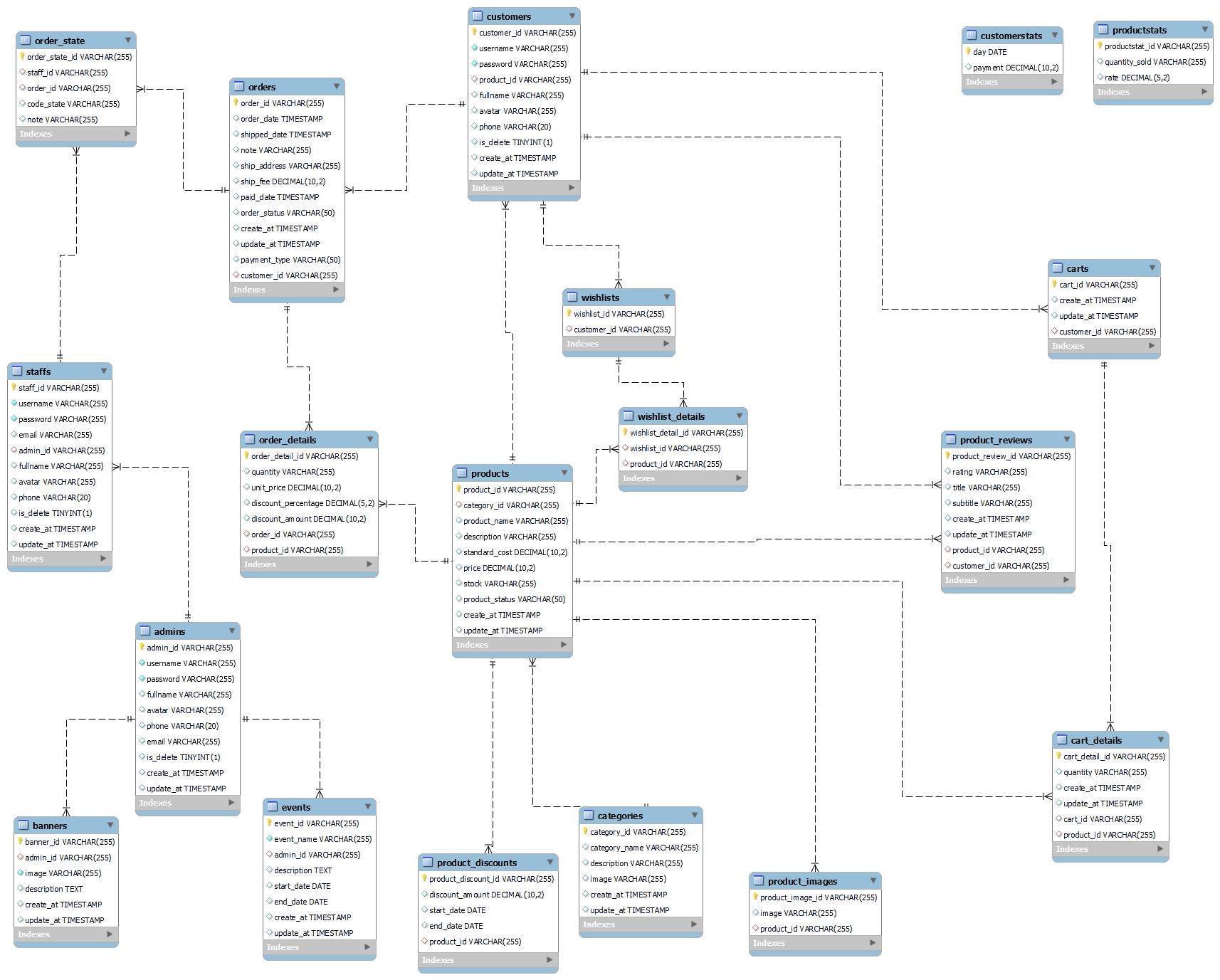
15. Lớp ProductImage

* Định nghĩa: Thông tin hình ảnh của sản phẩm.
* Thuộc tính:
  + productimage\_id: Mã định danh hình ảnh.
  + url: Đường dẫn URL của ảnh.
  + product: Sản phẩm liên kết với ảnh.
* Phương thức:
  + addProductImage(ProductImage image): Boolean – Thêm hình ảnh cho sản phẩm.
  + deleteProductImage(String productimage\_id): Boolean – Xóa hình ảnh theo mã định danh.
  + getImagesByProduct(Product product): List<ProductImage> – Lấy danh sách hình ảnh theo sản phẩm.

16. Lớp ProductReview

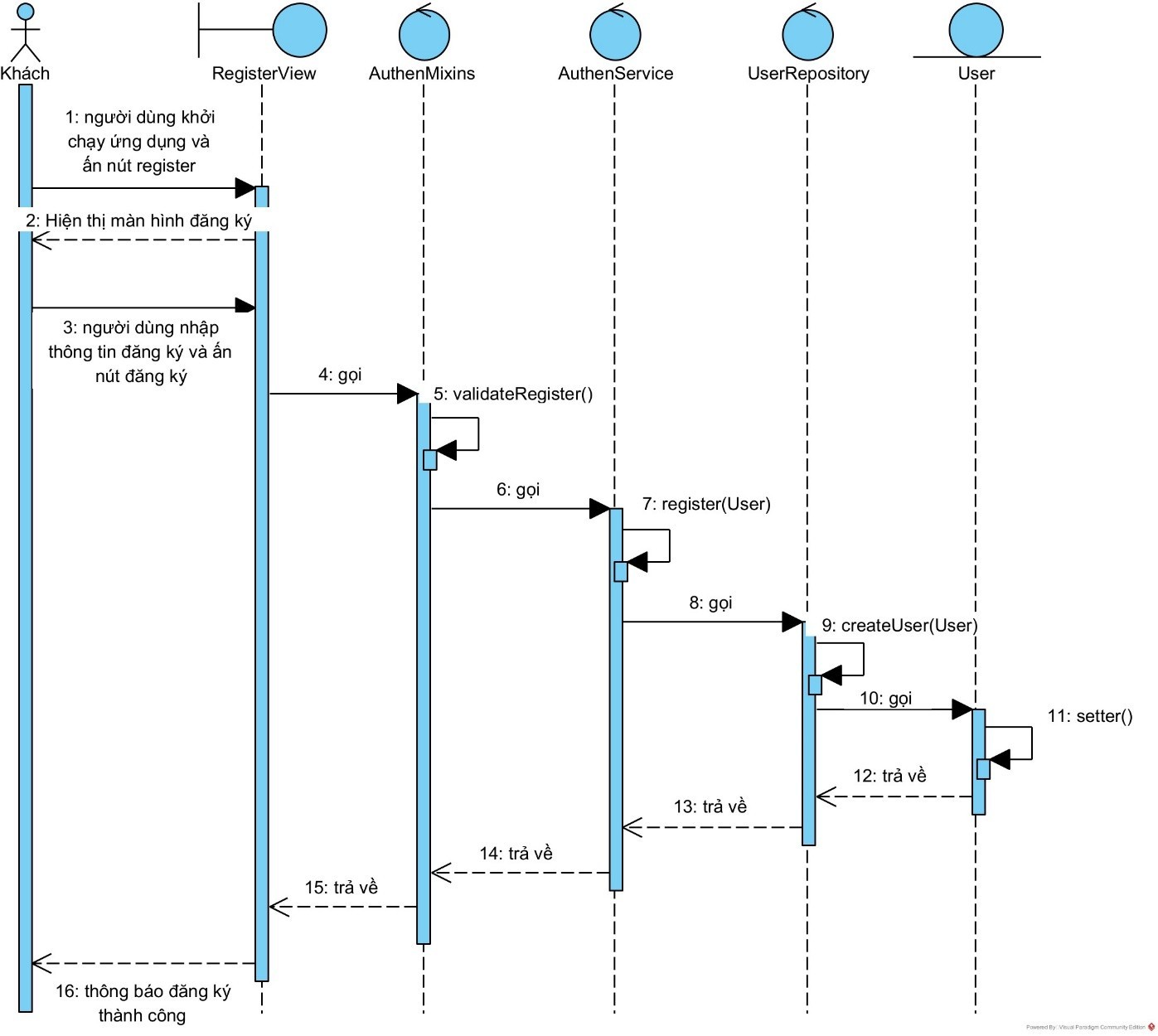
* Định nghĩa: Đánh giá sản phẩm.
* Thuộc tính:
  + review\_id: Mã định danh đánh giá.
  + rating: Xếp hạng (1 đến 5 sao).
  + comment: Bình luận đánh giá.
  + review\_date: Ngày đánh giá.
  + customer: Thông tin khách hàng đánh giá.
  + product: Thông tin sản phẩm được đánh giá.
* Phương thức:
  + addReview(ProductReview review): Boolean – Thêm đánh giá mới.
  + deleteReview(String review\_id): Boolean – Xóa đánh giá theo mã định danh.
  + getReviewsByProduct(Product product): List<ProductReview> – Lấy danh sách đánh giá theo sản phẩm.

Cơ sở dữ liệu toàn hệ thống:

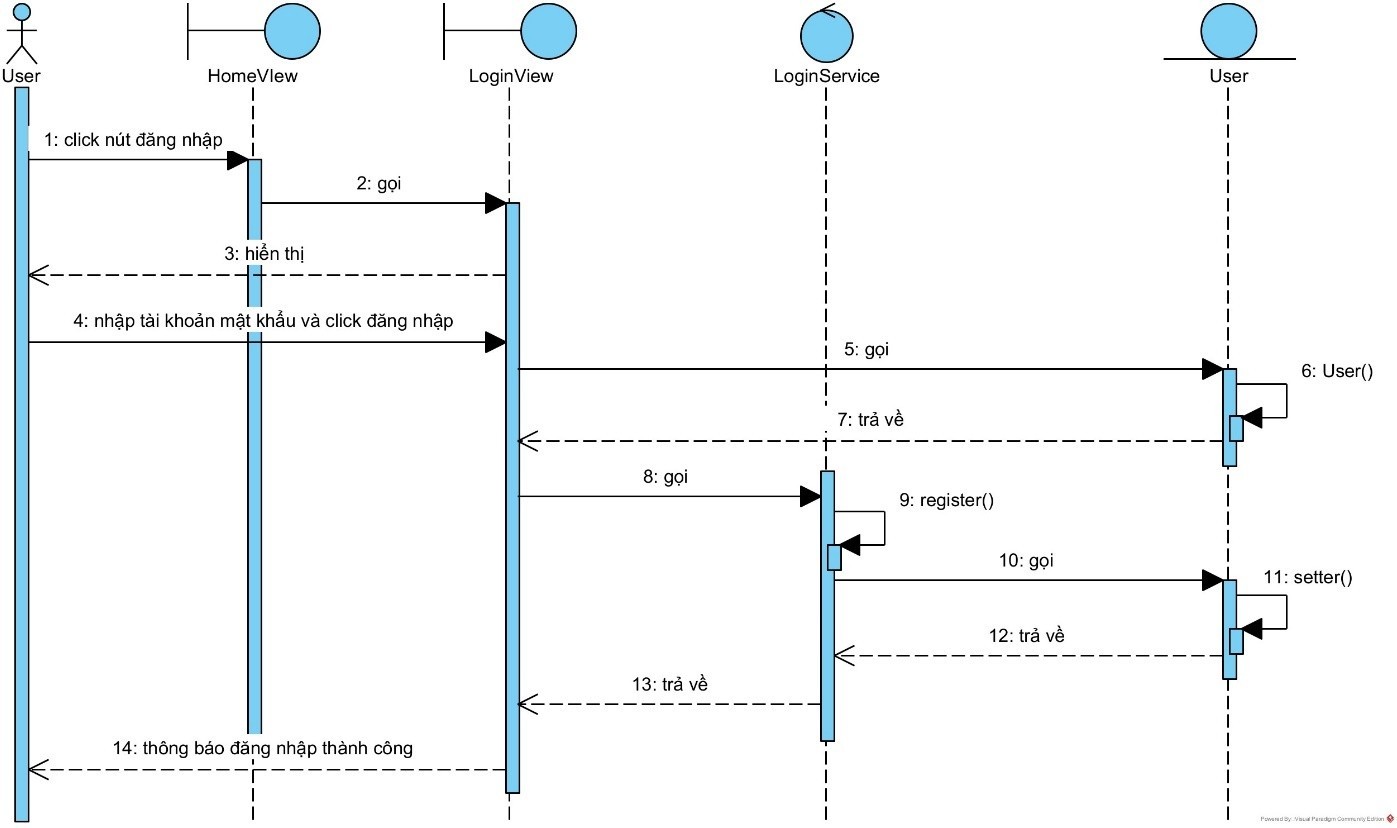


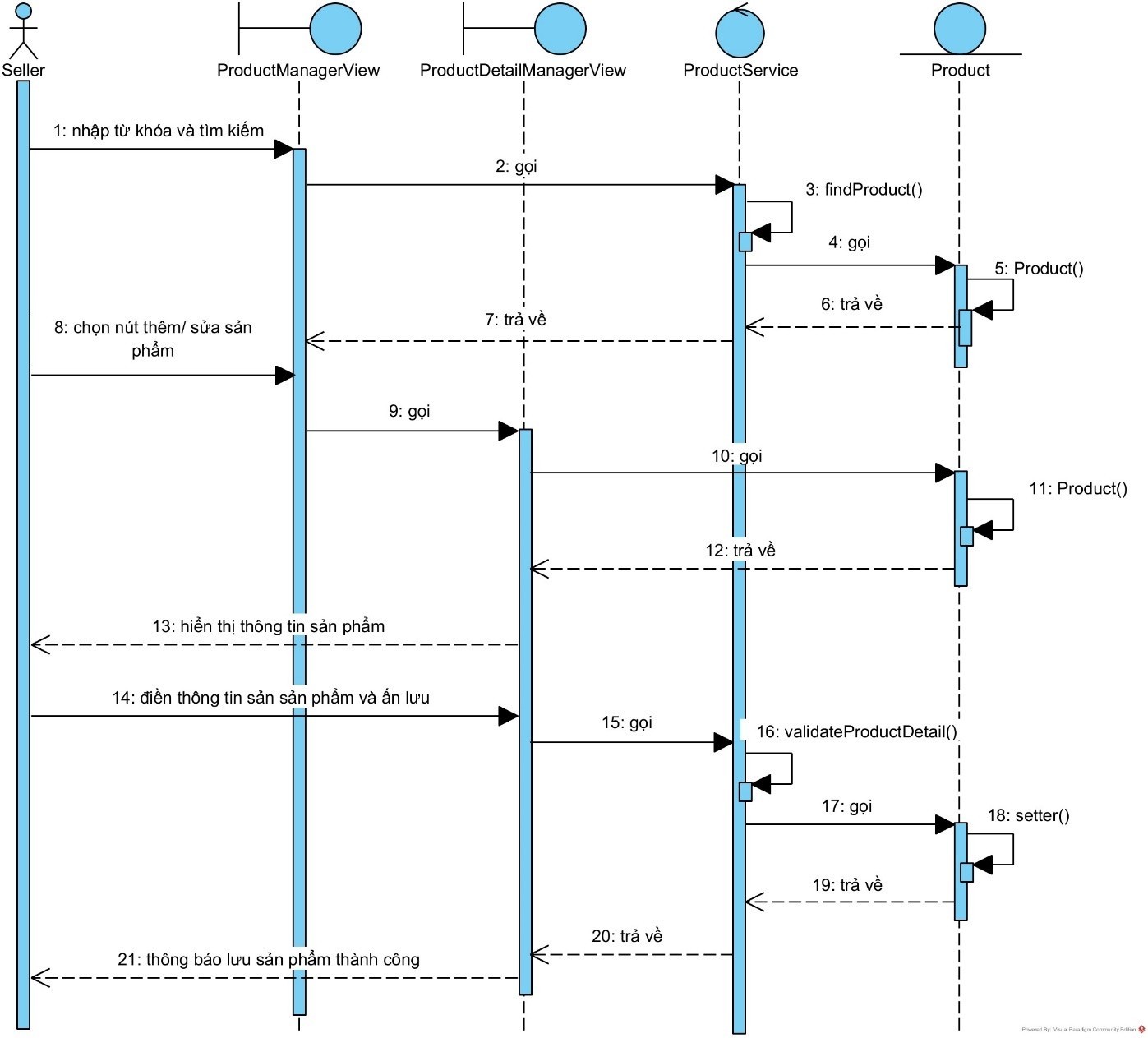
**Biều đồ tuần tự**

**1. Chức năng Đăng ký**

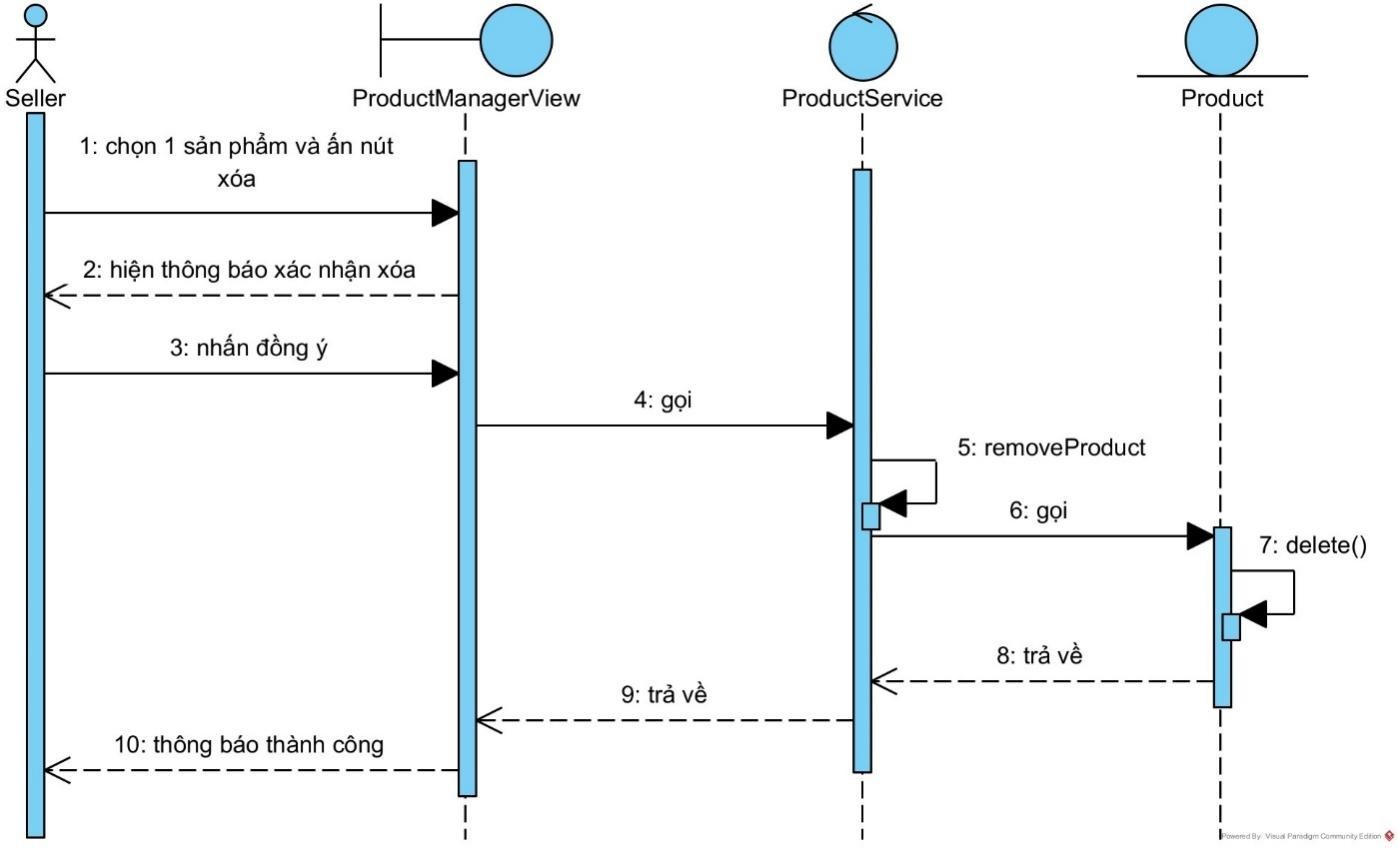
****

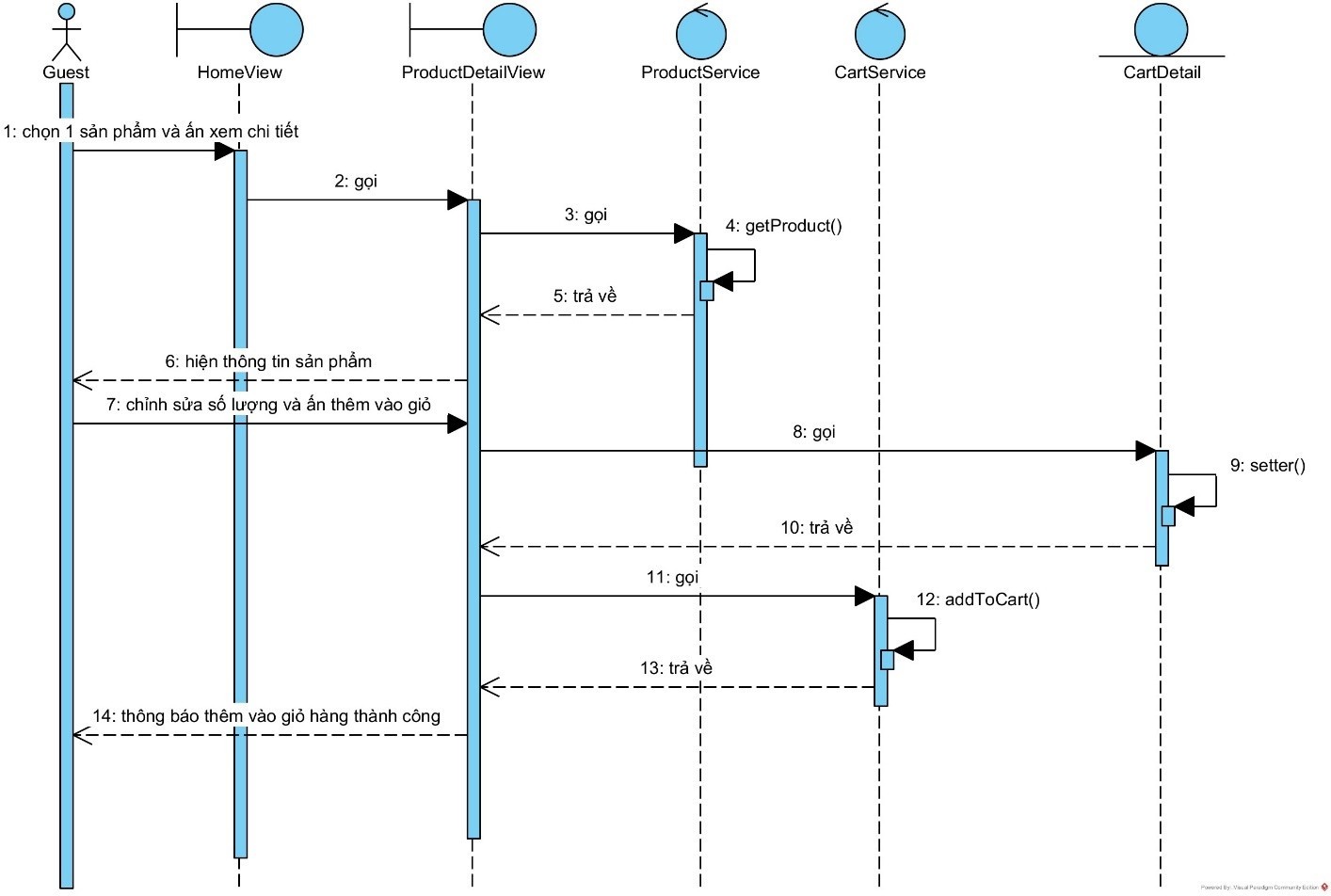
**2. Chức năng đăng nhập**

3. Chức năng thêm sửa sản phẩm

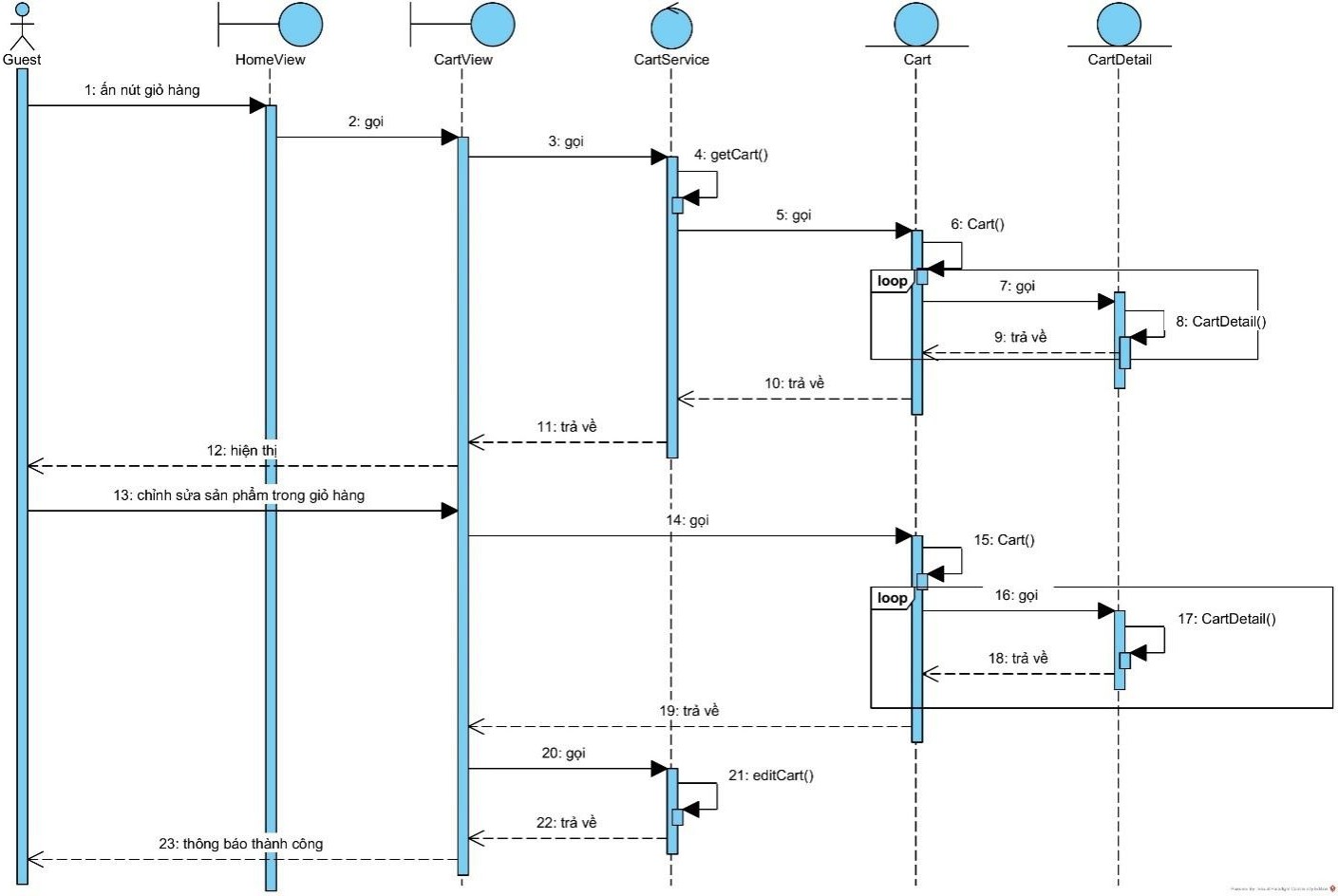


4. Chức năng xoá sản phẩm

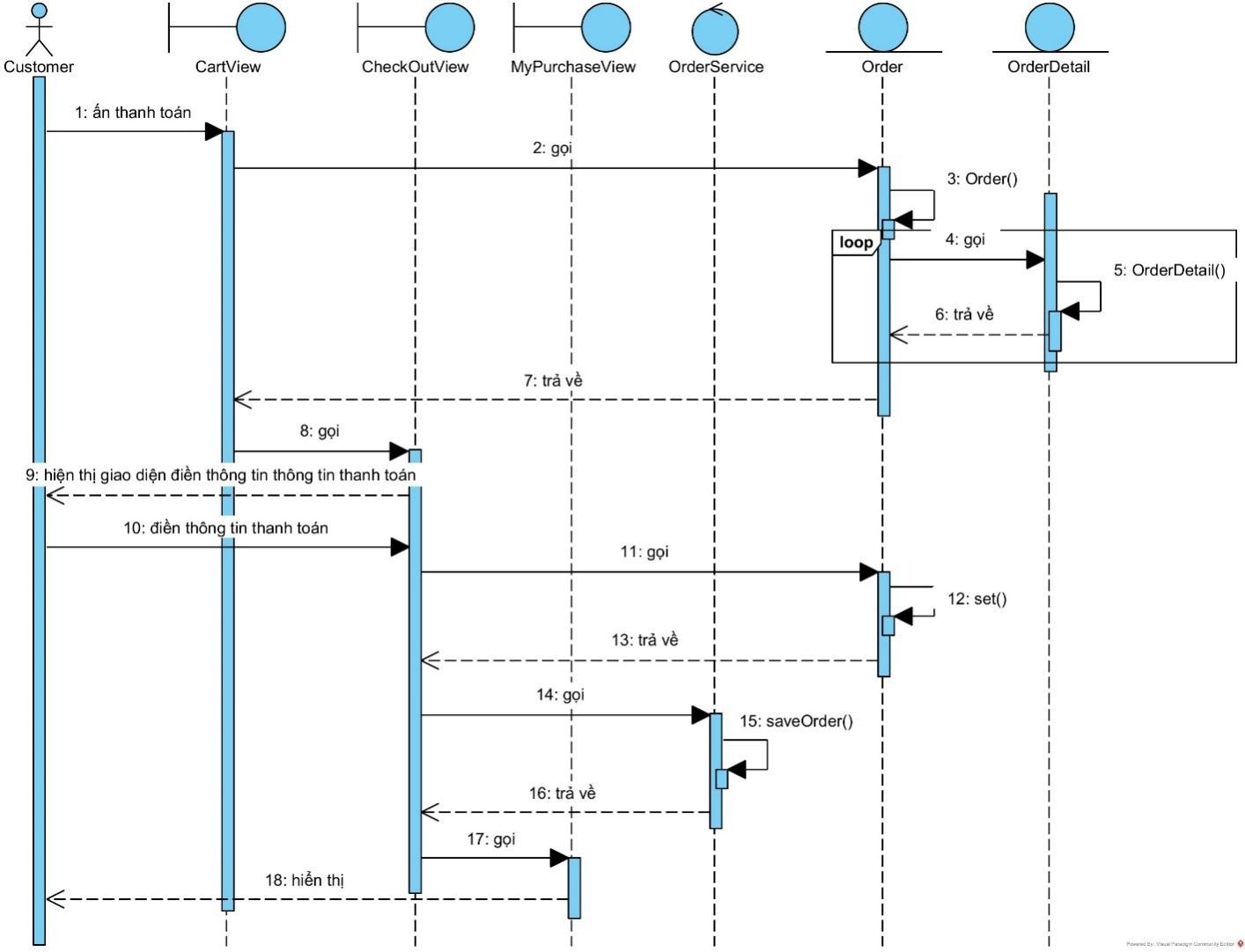
5. Chức năng quản lý giỏ hàng



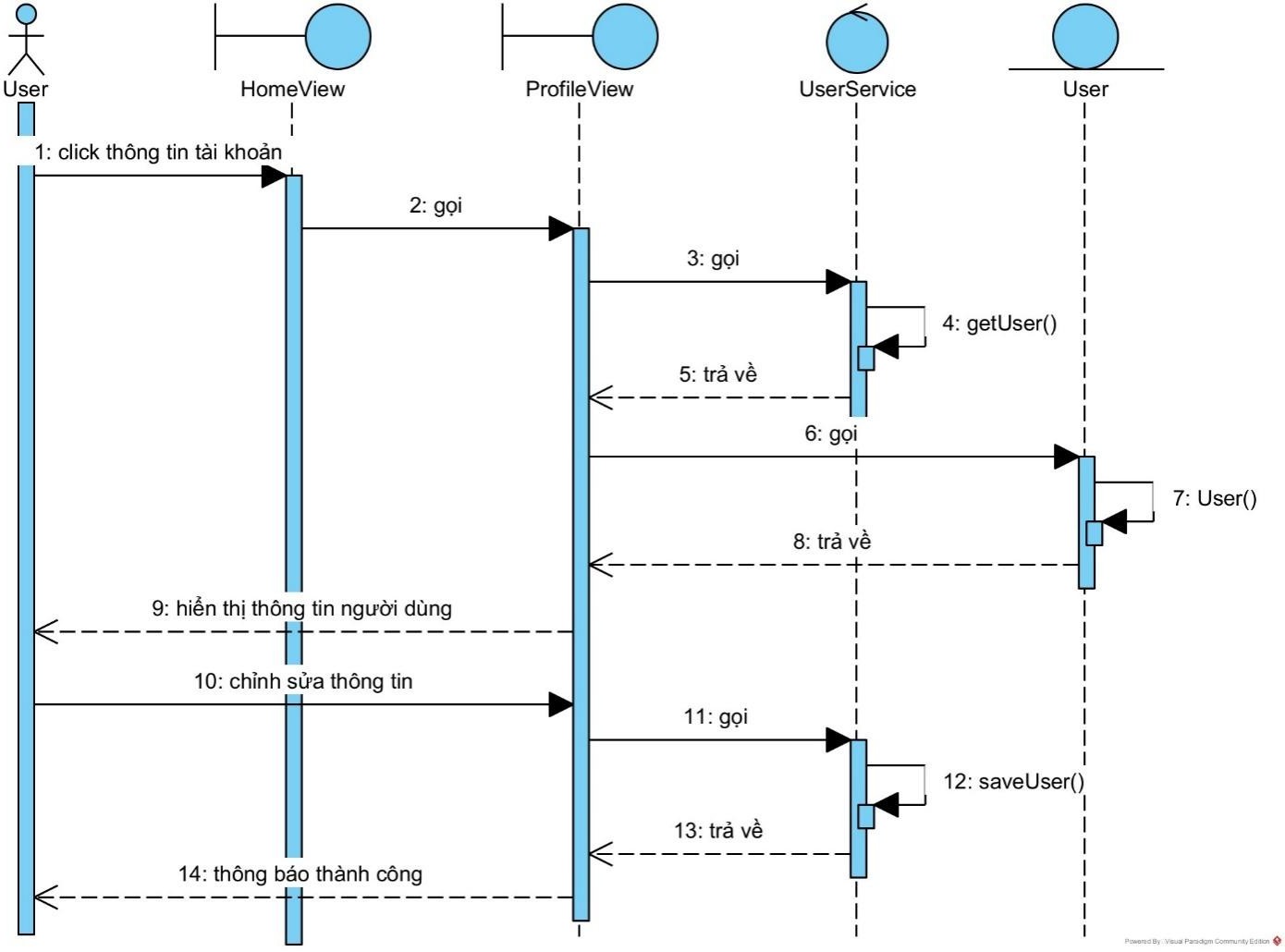
6. Chức năng chỉnh sửa giỏ hàng



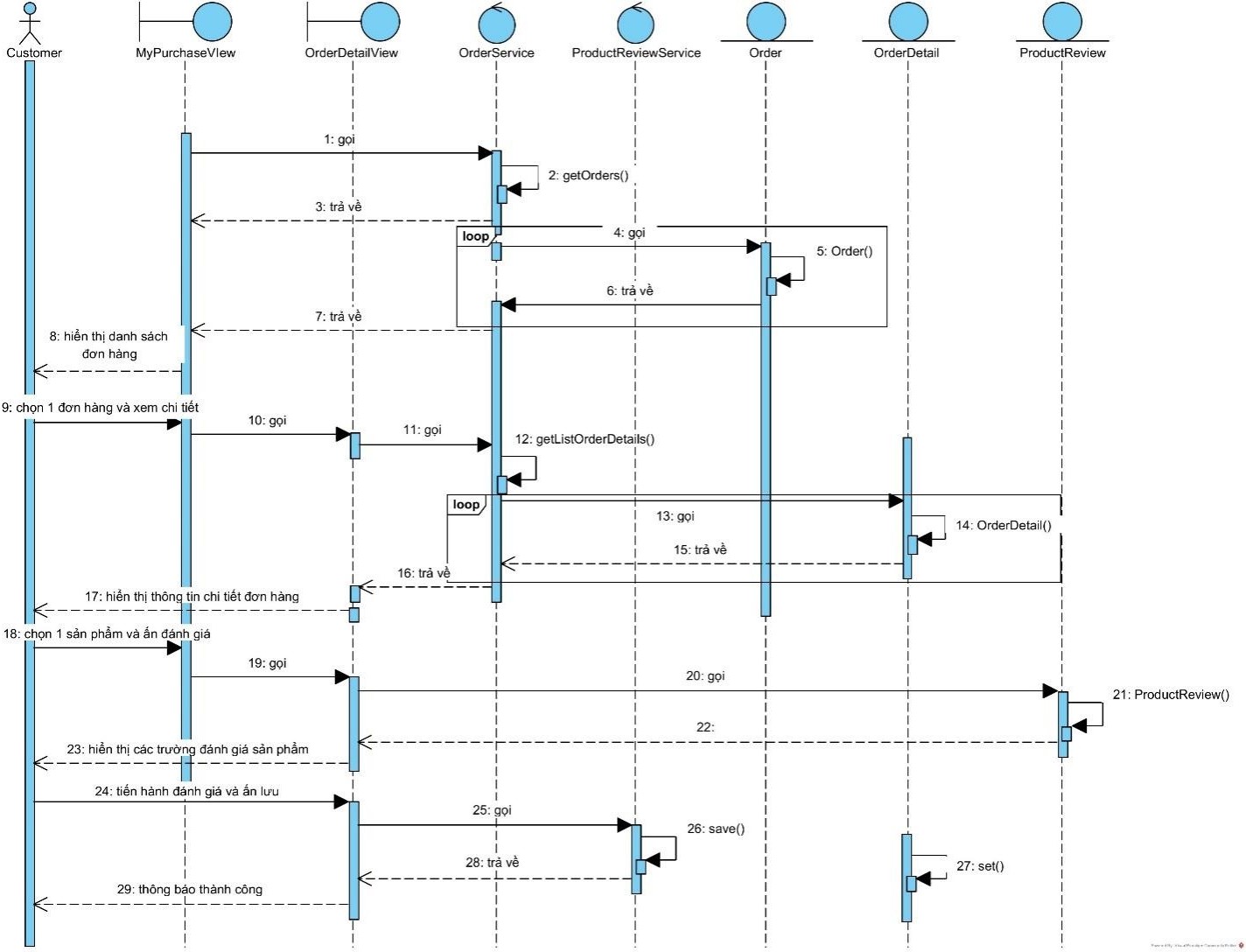
7. Chức năng thanh toán



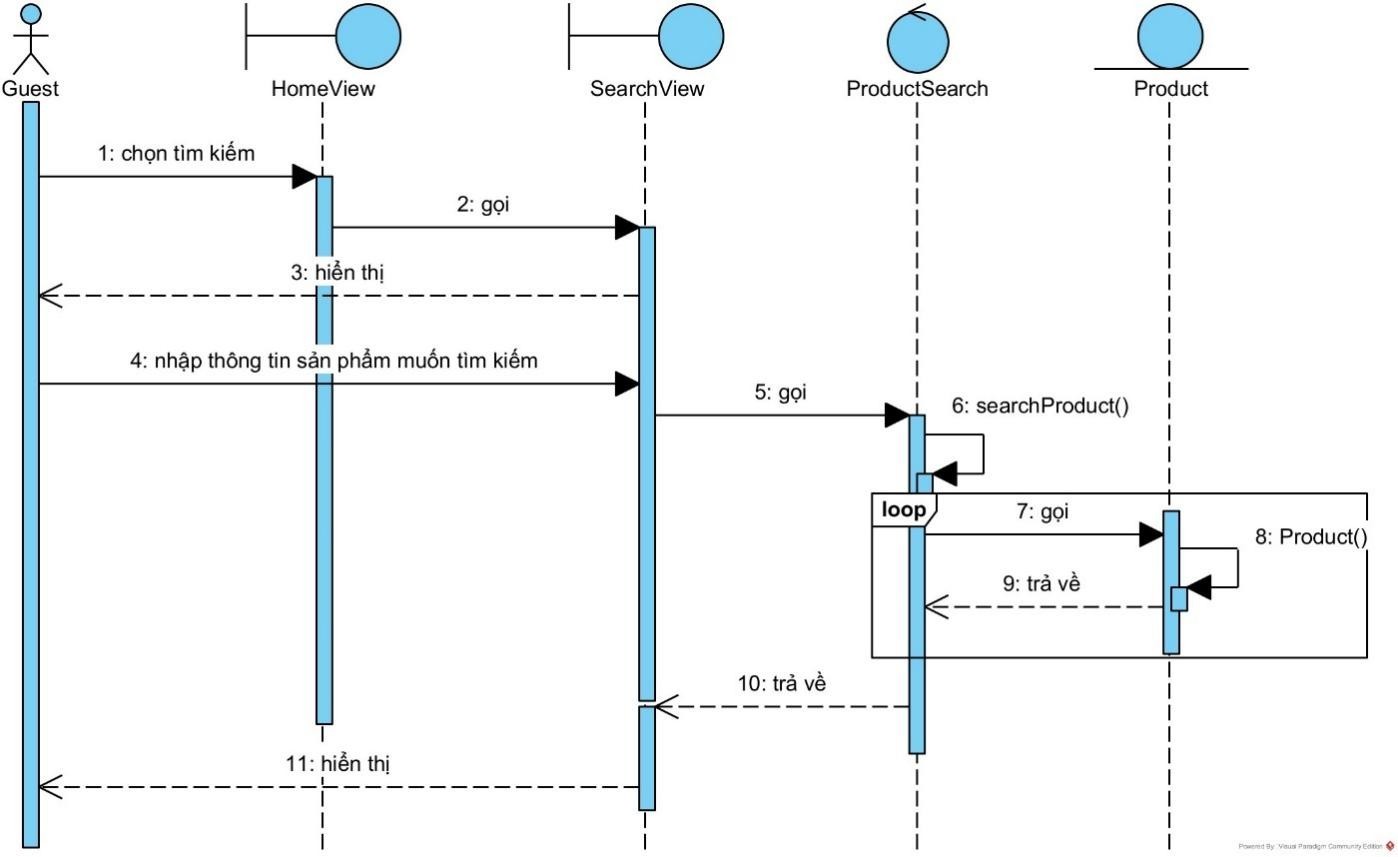
8. Chức năng quản lý thông tin tài khoản



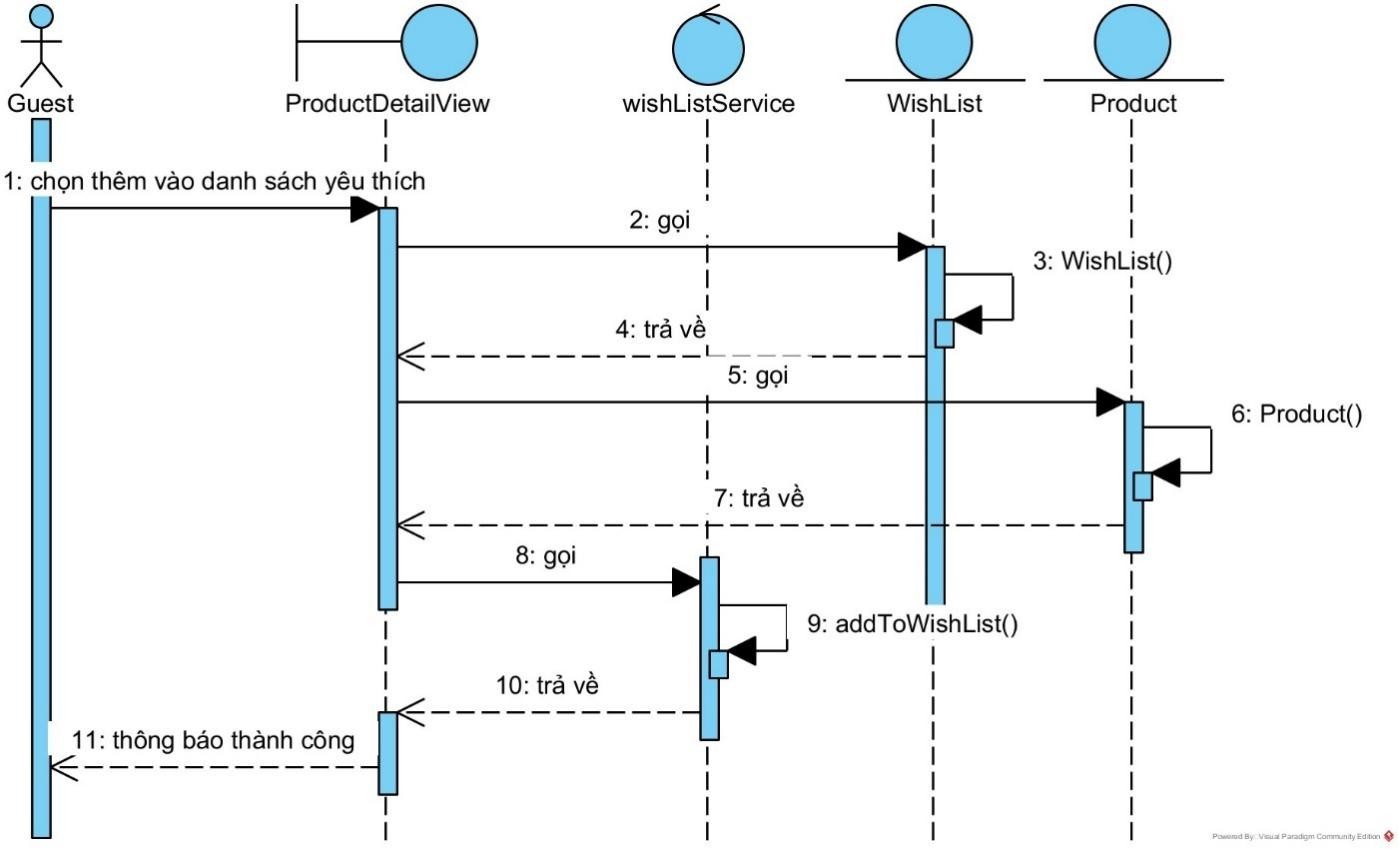
9. Chức năng đánh giá sản phẩm đã mua



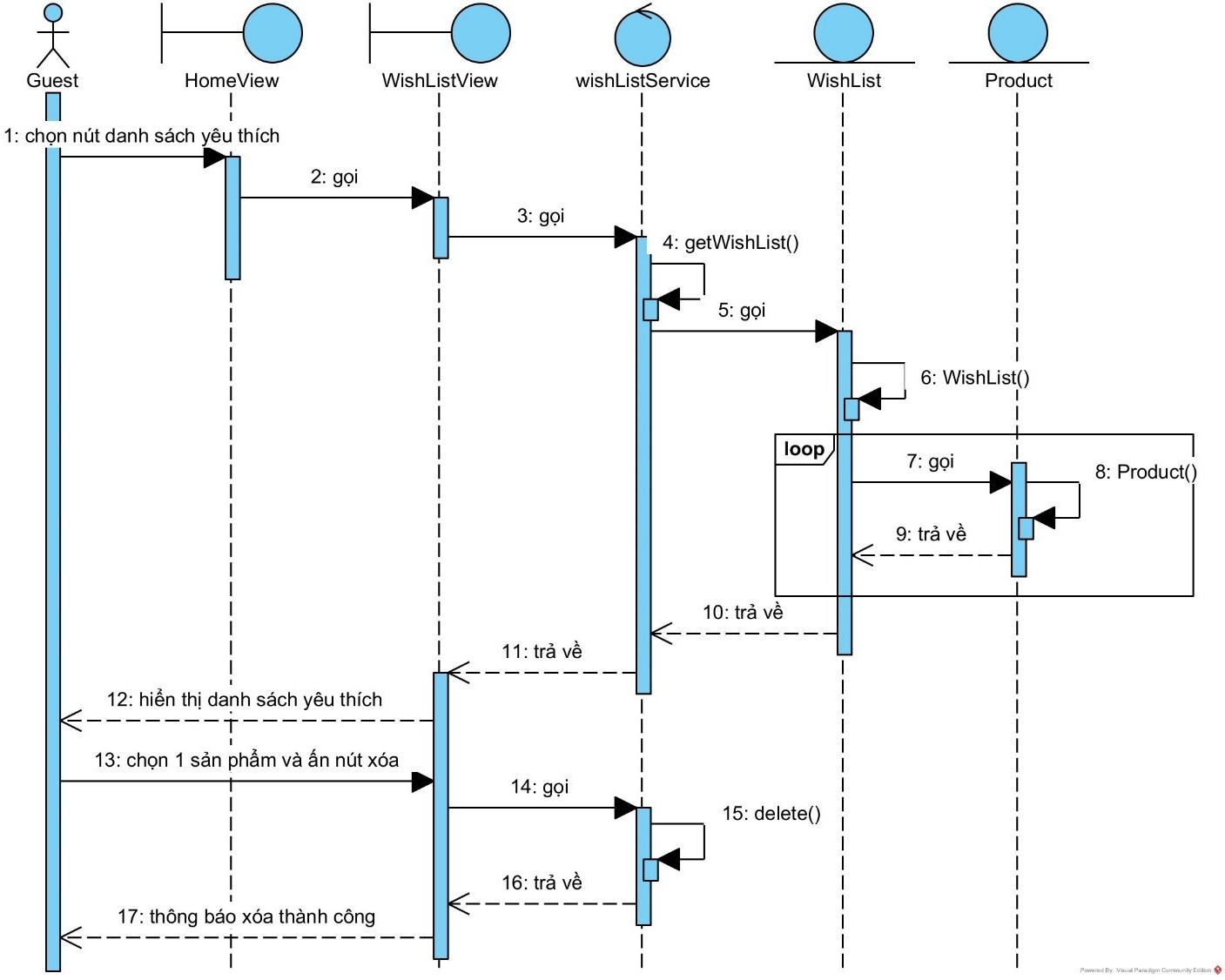
10. Chức năng tìm kiếm



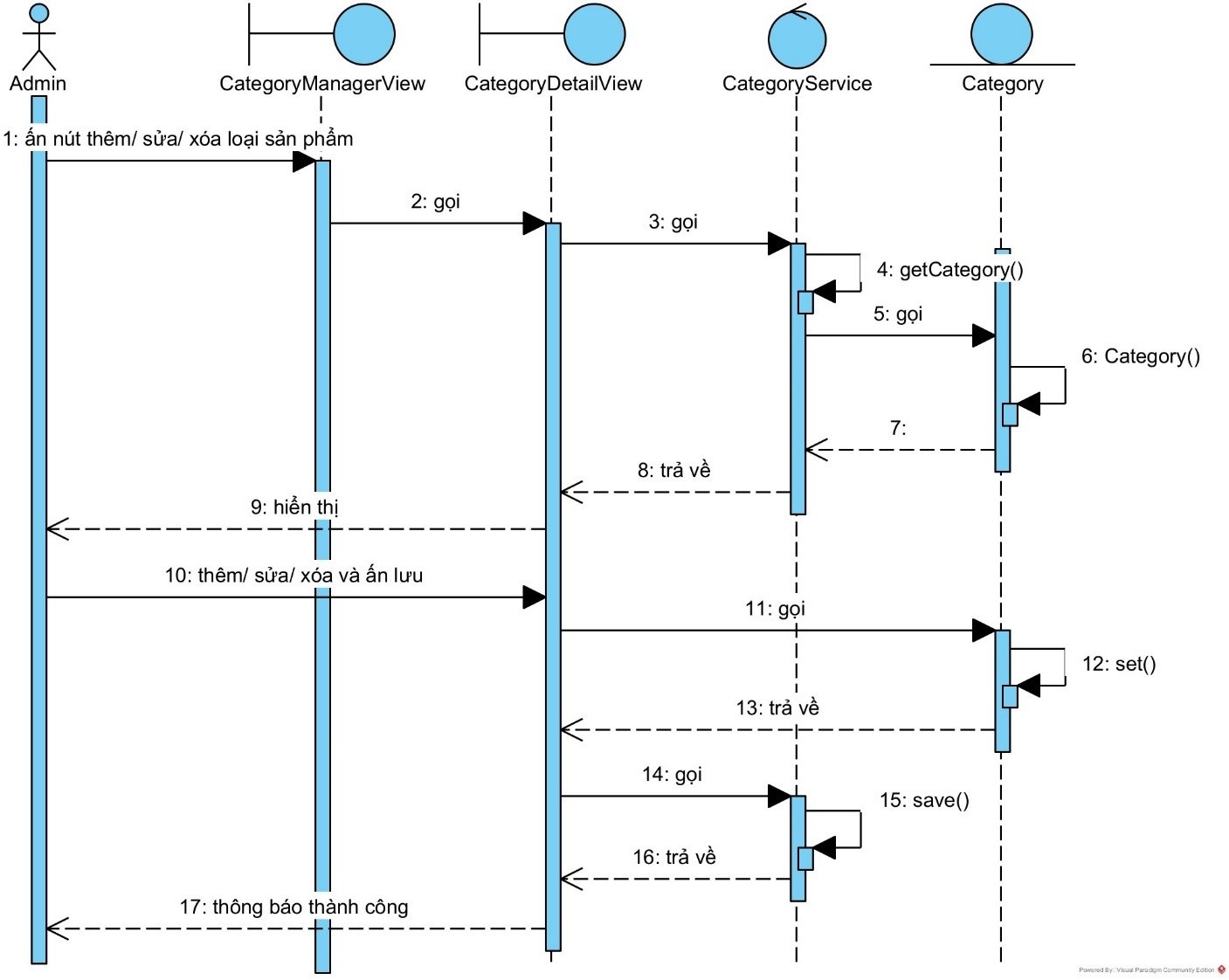
11. Chức năng thêm sản phẩm vào danh sách yêu thích



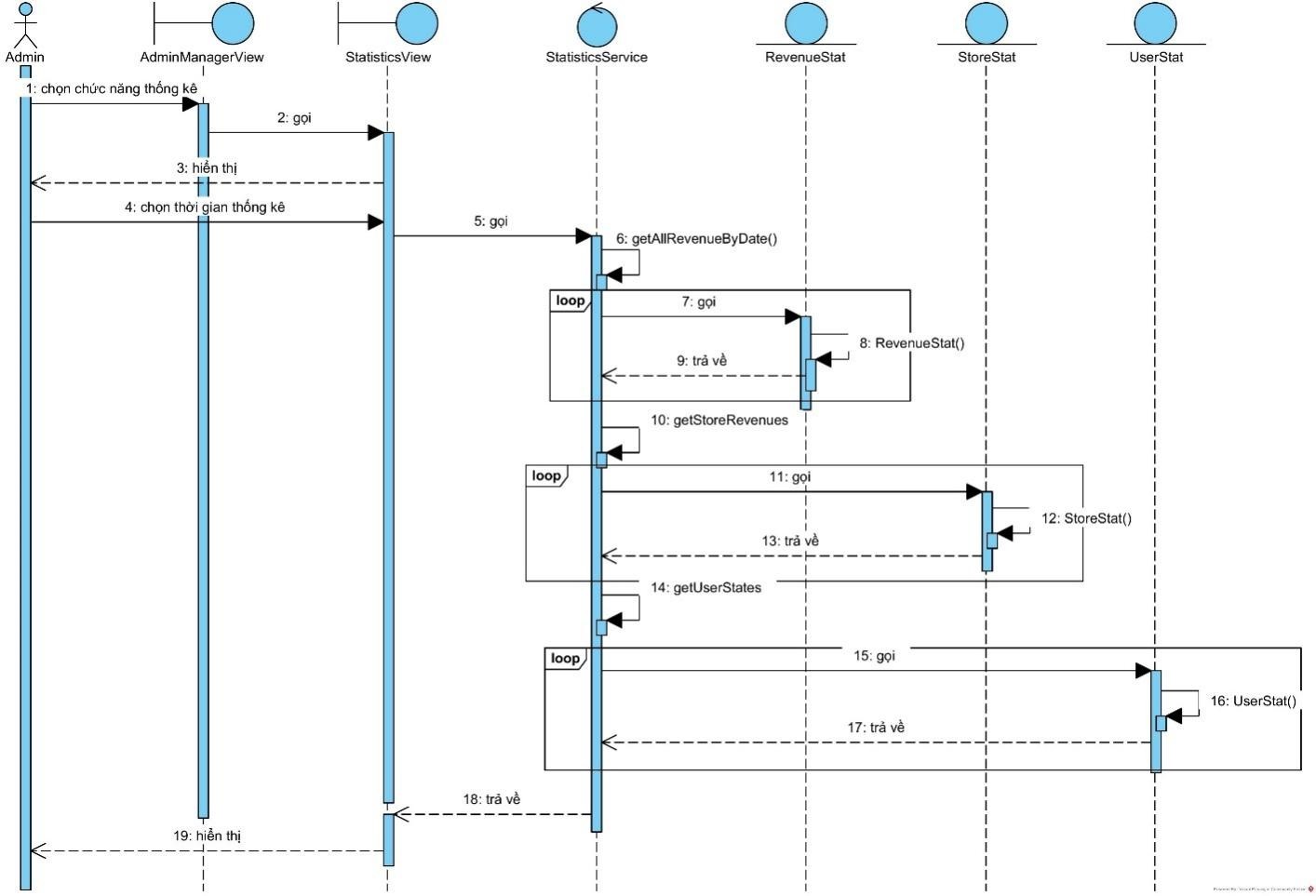
12. Chức năng xoá sản phẩm khỏi danh sách yêu thích



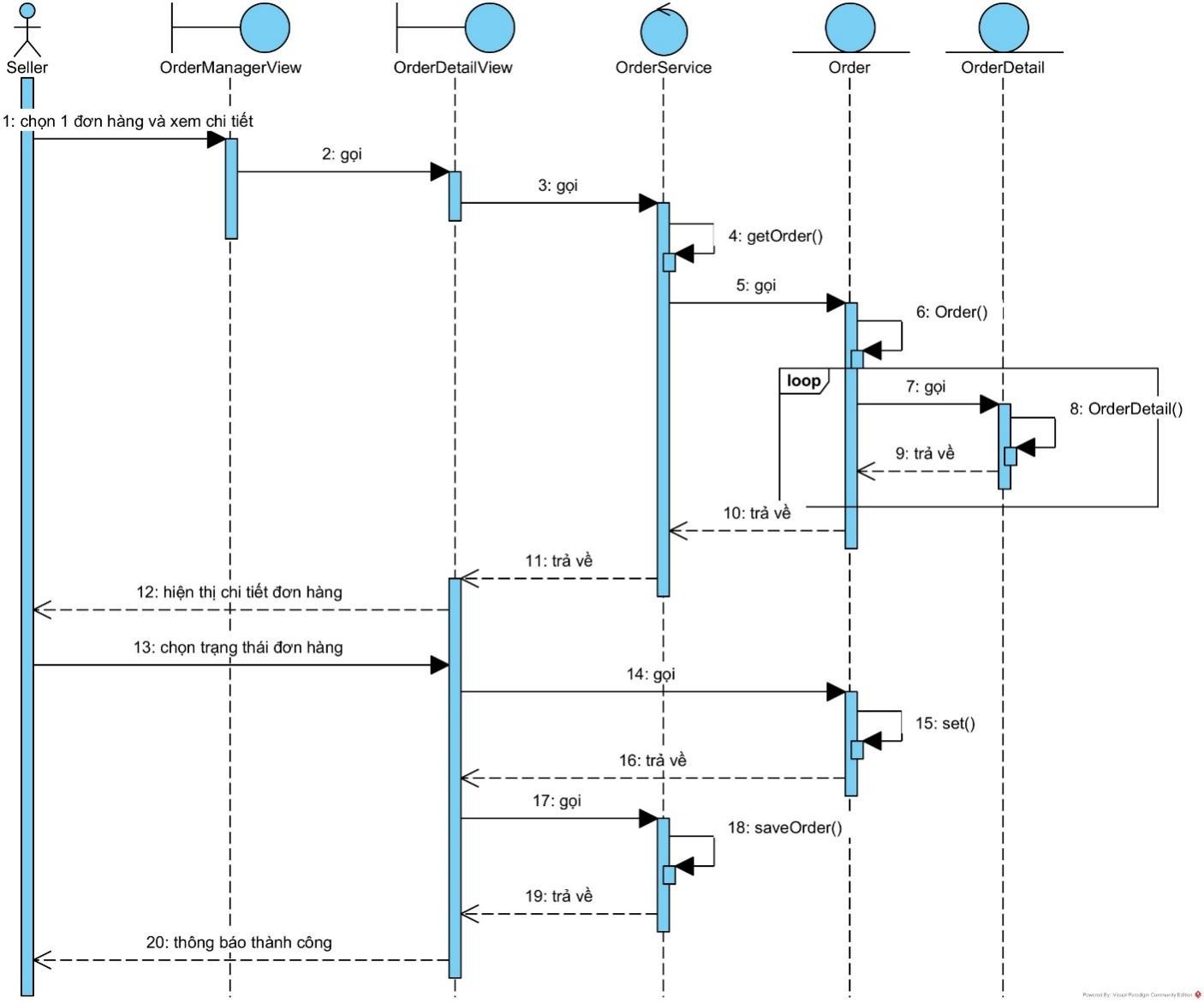
13. Chức năng quản lý sản phẩm



14. Chức năng thống kê



15. Chức năng quản lý đơn hàng



16.

# CHƯƠNG 3. CÀI ĐẶT VÀ THỬ NGHIỆM HỆ THỐNG

## 3.1. Môi trường triển khai hệ thống

## 3.2. Giao diện hệ thống