TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.Hồ Chí Minh

**KHOA CÔNG NGHỆ THÔNG TIN**



TP.Hồ Chí Minh - Năm 2021

**BÁO CÁO BÀI TẬP LỚN LẬP TRÌNH WWW (JAVA)**

Website giới thiệu, bán quần áo trẻ em trực tuyến

SVTH: Đồng Đức Khang 18041671

Nguyễn Nhật Huy 17083971

Hà Quốc Cường 17016541

Trần Đỗ Ngọc Tân 16022171

**Mục lục**

[**Chương 1: GIỚI THIỆU** 4](#_Toc73375921)

[**1.Tổng quan** 4](#_Toc73375922)

[**2.Mục tiêu đề tài** 4](#_Toc73375923)

[**3.Phạm vi đề tài** 5](#_Toc73375924)

[**4.Mô tả yêu cầu chức năng** 5](#_Toc73375925)

[**Chương 2 cơ sở lý thuyết** 6](#_Toc73375926)

[**1.Spring** 6](#_Toc73375927)

[**2.Spring Security** 6](#_Toc73375928)

[**3.Spring MVC** 7](#_Toc73375929)

[**Phát triển Web Application đơn giản với Spring MVC** 7](#_Toc73375930)

[**Tạo file cấu hình** 9](#_Toc73375931)

[**Tạo Spring MVC Controller** 10](#_Toc73375932)

[**Tạo JSP Views** 11](#_Toc73375933)

[**Triển khai web application** 11](#_Toc73375934)

[**Chương 3: phân tích** 13](#_Toc73375935)

[**3.1.Use case tổng quát** 13](#_Toc73375936)

[**3.2.Class Diagram** 14](#_Toc73375937)

[**3.3.Use case Đăng kí tài khoản** 14](#_Toc73375938)

[**3.4.Use case Xem danh sách sản phẩm** 17](#_Toc73375939)

[**3.5.Use case xem thông tin chi tiết của sản phẩm:** 20](#_Toc73375940)

[**3.6. use case Xem danh sách các đơn hàng** 23](#_Toc73375941)

[**3.7.Use case xóa sản phẩm** 24](#_Toc73375942)

[**3.8 Use case xem thông tin chi tiết từng loại sản phẩm** 26](#_Toc73375943)

[**3.9.Use case Thêm sản phẩm vào giỏ hàng:** 28](#_Toc73375944)

[**3.10 Use case Cập nhật số lượng của mặt hàng trong đơn hàng** 30](#_Toc73375945)

[**3.11 Use case Xem chi tiết các đơn hàng** 33](#_Toc73375946)

[**3.12.Use case Xem giỏ hàng:** 35](#_Toc73375947)

[**3.13.Use case Đăng nhập:** 37](#_Toc73375948)

[**3.14. Use case Thanh toán** 39](#_Toc73375949)

[**3.15.Use case Cập nhật số lượng của mặt hàng trong đơn hàng trực tuyến** 41](#_Toc73375950)

[**3.16 Use case Xem chi tiết các đơn hàng** 44](#_Toc73375951)

[**3.17 Use case Cập nhật thông tin loại sản phẩm** 47](#_Toc73375952)

[**3.18 Use case Cập nhật thông tin sản phẩm** 49](#_Toc73375953)

[**3.19.Xem chi tiết đơn hàng** 50](#_Toc73375954)

[**3.20.Xem danh sách các đơn hàng** 50](#_Toc73375955)

[**3.21.Use case Tìm kiếm sản phẩm** 53](#_Toc73375956)

[**Chương 4:Thiết kế và Hiện thực** 55](#_Toc73375957)

[**4.1.Giao diện đăng ký:** 55](#_Toc73375958)

[**4.2.Giao diện đăng nhập:** 56](#_Toc73375959)

[**4.3.Giao diện trang chủ** 56](#_Toc73375960)

[**4.4.Giao diện giỏ hàng:** 57](#_Toc73375961)

[**4.5.Giao diện thanh toán:** 57](#_Toc73375962)

[**4.6.Giao diện hóa đơn:** 57](#_Toc73375963)

[**4.7. Giao diện Quản lý sản phẩm:** 58](#_Toc73375964)

[**4.8. Giao diện chi tiết sản phẩm** 58](#_Toc73375965)

[**4.9.Giao diện quản lý khách hàng:** 58](#_Toc73375966)

[**4.10.Giao diện Quản lý hóa đơn:** 59](#_Toc73375967)

[**Chương 5: KẾT LUẬN** 59](#_Toc73375968)

[**5.1.Kết quả đạt được** 59](#_Toc73375969)

[**5.2.Hạn chế của đồ án** 59](#_Toc73375970)

[**5.3.Hướng phát triển** 59](#_Toc73375971)

# **Chương 1: GIỚI THIỆU**

## **1.Tổng quan**

Kinh doanh bán hàng trực tuyến ngày càng trở thành xu hướng hiện nay . Không chỉ cá nhân, ngay cả những doanh nghiệp, cửa hàng thời trang khi đứng trước áp lực cạnh tranh ngày càng gay gắt cũng như chuyển mình bằng cách thiết kế web bán hàng nói chung và thiết kế web bán quần áo trẻ em nói riêng nhằm khai thác tối đa thị trường trực tuyến đầy tiềm năng .

Giờ đây không chỉ cá nhân mà ngay cả những doanh nghiệp, cửa hàng thời trang đứng trước áp lực cạnh tranh ngày càng gay gặt này đòi hỏi phải chuyển mình bằng việc thiết kế web bán hàng nói chung và thiết kế web bán quần áo trẻ em nói riêng nhằm khai thác tối đa thị trường kinh doanh trực tuyến đầy tiềm năng này.

Với sự phát triển của internet , ngành thời trang quần áo trẻ em trở nên phát triển mạnh mẽ , giúp doanh nghiệp, cá nhân tiếp cận khách hàng cực kỳ lớn . Các bà mẹ , ông bố cũng ngày càng mua quần áo online cho trẻ nhỏ càng nhiều . Đó là lý do tại sao mỗi ngày hàng trăm cửa hàng thời trang ra đời và đặc biệt là cửa hàng thời trang online cho trẻ em. Từ đó, nhóm giới thiệu đến website bán quần áo trẻ em trực tuyến.

## **2.Mục tiêu đề tài**

* Quản lý thông tin tài khoản khách hàng
* Thiết kế giao diện website trực quan, dễ sử dụng giúp người dung dễ dàng lựa chọn và mua đươc sản phẩm họ ưa thích
* Tạo website bao gồm 3 loại người dùng tương tác: người dùng có tài khoản (customer,employee), người quản trị hệ thống (admin).
* Tạo các chức năng dành cho người dùng có tài khoản (customer,employee) ngoài ra người dùng có tài khoản (customer) còn có thể:
  + Khách hàng có thể đăng ký tài khoản mới, có thể đăng nhập vào tài khoản
* Xử lý thanh toán (chức năng này thực hiện khi giỏ hàng đã có sản phẩm và người dùng đăng nhập thành công vào hệ thống): cập nhật thông tin vào CSDL + gửi email + thông báo đăng ký đặt hàng thành công với các thông tin kèm theo. Sau khi xử lý thành công, Session được xóa về null.
* Tạo các chức năng dành cho người quản trị hệ thống (admin) có thể thực hiện được chức năng như một người dùng có tài khoản (customer). Ngoài ra, chức năng khác dành cho người quản trị hệ thống (admin) - Phần Back-End:
* Tìm kiếm thông tin về sản phẩm/loại sản phẩm, các đơn đặt sản phẩm.
* Quản lý thông tin sản phẩm/loại sản phẩm:
* Xem danh sách sản phẩm/loại sản phẩm.
* Xem chi tiết từng sản phẩm/loại sản phẩm.
* Xóa sản phẩm/loại sản phẩm trong trường hợp sản phẩm chưa có trong đơn hàng nào hoặc loại sản phẩm chưa có sản phẩm nào.
* Thêm mới, cập nhật thông tin sản phẩm/loại sản phẩm.
* Quản lý thông tin đơn hàng trực tuyến:
* Xem danh sách các đơn hàng (sắp xếp theo ngày mua).
* Xem chi tiết đơn hàng.
* Cập nhật số lượng của mặt hàng trong đơn hàng trực tuyến.

## **3.Phạm vi đề tài**

* Đồ án được thực hiện dựa trên cơ sở phạm vi môn học Lập trình WWW:
* Dùng công nghệ:
* SQL Server
* Spring Security
* Spring MVC
* Đồ án thực hiện những mục tiêu:
* Đăng ký tài khoản
* Xem danh sách sản phẩm
* Xem chi tiết của từng sản phẩm từ danh sách sản phẩm
* Thêm sản phẩm vào giỏ hàng
* Xem giỏ hàng
* Xử lý thanh toán
* Tìm kiếm thông tin về sản phẩm/loại sản phẩm, tài khoản người dùng, các đơn đặt sản phẩm
* Quản lý thông tin sản phẩm/loại sản phẩm
* Quản lý thông tin đơn hàng trực tuyến

## **4.Mô tả yêu cầu chức năng**

* Quy trình dành cho khách hàng:
* Đăng nhập tài khoản vào website quần áo trẻ em trực tuyến
* Tìm kiếm, tham khảo thông tin sản phẩm quần áo, dịch vụ, khuyến mại
* Khách hàng lựa chọn cách thức giao dịch
* Đơn hàng của khách hàng sẽ được chuyển thông tin đến người bán. Tùy vào thỏa thuận giữa khách hàng và người bán mà website có thể hỗ trợ quá trình vận chuyển hàng hóa và thanh toán đảm bảo.
* Khách hàng nhận sản phẩm.
* Khách hàng thắc mắc, khiếu nại người bán (nếu có) qua tổng đài hỗ trợ của website.

# **Chương 2 cơ sở lý thuyết**

## **1.Spring**

- Các tính năng cốt lõi của Spring Framework có thể được sử dụng để phát triển bất kỳ ứng dụng Java nào, nhưng có những phần mở rộng để xây dựng các ứng dụng web trên nền tảng Java EE. Spring framework nhắm mục tiêu làm cho việc phát triển J2EE dễ sử dụng hơn và thúc đẩy các phương pháp lập trình tốt bằng cách cho phép mô hình lập trình dựa trên POJO( Plain Old Java Object).

## **2.Spring Security**

- Spring Security cung cấp các dịch vụ bảo mật toàn diện cho các ứng dụng doanh nghiệp có nền tảng Java EE.

Spring Security cung cấp 2 cơ chế cơ bản:

* Authentication (xác thực): là tiến trình thiết lập một principal. Principal có thể hiểu là một người, hoặc một thiết bị, hoặc một hệ thống nào đó có thể thực hiện một hành động trong ứng dụng của bạn.
* Authorization (phân quyền) hay Access-control: là tiến trình quyết định xem một principal có được phép thực hiện một hành động trong ứng dụng của bạn hay không. Trước khi diễn tiến tới Authorization, principal cần phải được thiết lập bởi Authentication.
* Các thành phần cốt lõi:
* Security, SecurityContext và Authentication

SecurityContext là interface cốt lõi của Spring Security, lưu trữ tất cả các chi tiết liên quan đến bảo mật trong ứng dụng. Khi chúng ta kích hoạt Spring Security trong ứng dụng thì SecurityContext cũng sẽ được kích hoạt theo.

Chúng ta sẽ không truy cập trực tiếp vào SecurityContext, thay vào đó sẽ sử dụng lớp SecurityContextHolder. Lớp này lưu trữ security context hiện tại của ứng dụng, bao gồm chi tiết của principal đang tương tác với ứng dụng. Spring Security sẽ dùng một đối tượng Authentication để biểu diễn thông tin này.

* UserDetails và UserDetailsService

Trong đoạn code trên, chúng ta có được một principal từ đối tượng Authentication. Principal đơn giản chỉ là một đối tượng và sẽ được ép kiểu sang UserDetails.

UserDetails là một interface cốt lõi của Spring Security. Nó đại diện cho một principal nhưng theo một cách mở rộng và cụ thể hơn:

* getAuthorities(): trả về danh sách các quyền của người dùng.
* getPassword(): trả về password đã dùng trong qúa trình xác thực.
* getUsername(): trả về username đã dùng trong qúa trình xác thực
* isAccountNonExpired(): trả về true nếu tài khoản của người dùng chưa hết hạn.
* isAccountNonLocked(): trả về true nếu người dùng chưa bị khóa.
* isCredentialsNonExpired(): trả về true nếu chứng thực (mật khẩu) của người dùng chưa hết hạn.
* isEnabled(): trả về true nếu người dùng đã được kích hoạt GrantedAuthority.

- Phương thức getAuthorities() sẽ trả về một tập hợp các đối tượng GrantedAuthority. Một GrantedAuthority là một quyền được ban cho principal. Các quyền đều có tiền tố là ROLE\_, ví dụ như ROLE\_ADMIN, ROLE\_MEMBER...

## **3.Spring MVC**

MVC là một trong những modules quan trọng nhất của Spring Framework. Nó xây dựng trên bộ chứa Spring IoC mạnh mẽ và sử dụng rộng rãi các tính năng của container để đơn giản hóa cấu hình của nó.

Model-View-Controller (MVC) là một mẫu thiết kế phổ biến trong thiết kế UI. Nó tách rời logic business từ các UI bằng cách tách biệt các vai trò của model, views và controller trong một ứng dụng. Các models có trách nhiệm đóng gói dữ liệu ứng dụng cho các views để trình bày. Views chỉ nên trình bày dữ liệu này, không bao gồm bất kỳ logic business nào.Controller có trách nhiệm nhận yêu cầu từ người dùng và gọi các services back-end cho gia công kinh doanh. Sau khi xử lý, các services back-end có thể trả về một số dữ liệu để hiển thị. Controller thu thập dữ liệu này và chuẩn bị các models để trình bày. Ý tưởng cốt lõi của mẫu MVC là tách logic business khỏi các UI để cho phép chúng thay đổi độc lập mà không ảnh hưởng lẫn nhau.

Trong một ứng dụng Spring MVC, các models thường bao gồm các domain object được xử lý bởi lớp service và được duy trì bởi persistence layer. Các views thường là các templates JSP được viết bằng Java Standard Tag Library (JSTL). Tuy nhiên, cũng có thể định nghĩa các chế độ xem là file PDF, file Excel, service web RESTful, hoặc thậm chí giao diện Flex, giao diện cuối cùng thường được mệnh danh Internet applications (RIAs) phong phú.

Sau khi kết thúc chương này, bạn sẽ có thể phát triển các ứng dụng web Java bằng Spring MVC. Bạn cũng sẽ hiểu các controller chung và view types của Spring MVC, bao gồm cả những gì đã trở thành việc sử dụng các chú thích để tạo controller trên thực tế kể từ khi phát hành Spring 3.0. Hơn nữa, bạn sẽ hiểu các nguyên tắc cơ bản của Spring MVC, sẽ là nền tảng cho các chủ đề nâng cao hơn được đề cập trong các chương sắp tới.

### **Phát triển Web Application đơn giản với Spring MVC**

Thành phần trung tâm của Spring MVC là bộ điều khiển phía trước. Trong Spring MVC Application đơn giản nhất, trình điều khiển là servlet duy nhất bạn cần định cấu hình trong bộ mô tả triển khai web Java (nghĩa là tệp web.xml hoặc một ServletContainerInitializer). Một MVC Spring controller thường được gọi là một servlet bộ điều phối thực hiện một trong các mẫu thiết kế Java EE lõi được gọi là Bộ điều khiển phía trước. Nó hoạt động như bộ điều khiển phía trước của Spring MVC framework và mọi yêu cầu web phải đi qua nó để có thể quản lý toàn bộ quy trình xử lý yêu cầu.

Khi một yêu cầu web được gửi đến ứng dụng Spring MVC, trước tiên bộ điều khiển sẽ nhận được yêu cầu. Sau đó nó tổ chức các components khác nhau được cấu hình trong ngữ cảnh hoặc annotation của ứng dụng Web Spring có trong bộ điều khiển chính nó, tất cả cần thiết để xử lý yêu cầu.

Hình phía dưới cho thấy luồng xử lý yêu cầu chính trong Spring MVC.

  
Để xác định một lớp controller trong Spring, một lớp phải được đánh dấu bằng @Controll hoặc @RestControll annotation. Khi một lớp @Controll annotation (tức là một lớp trình điều khiển) nhận được một yêu cầu, nó sẽ tìm một phương thức xử lý thích hợp để xử lý yêu cầu. Điều này đòi hỏi một lớp controller ánh xạ mỗi yêu cầu tới một phương thức xử lý bằng một hoặc nhiều ánh xạ xử lý. Để làm như vậy, một phương thức lớp controller được decorated với @RequestMapping annotation, biến chúng thành các phương thức xử lý. Signature cho các phương thức xử lý này như bạn có thể mong đợi từ bất kỳ lớp standard nào đã kết thúc mở. Bạn có thể chỉ định một tên tùy ý cho một phương thức xử lý và xác định một loạt các method arguments. Tương tự, một phương thức xử lý có thể trả về bất kỳ chuỗi giá trị nào (ví dụ: String hoặc void), tùy thuộc vào ứng dụng logic nó đáp ứng. Khi book progresses, bạn sẽ gặp các method arguments khác nhau có thể được sử dụng trong các phương thức xử lý bằng cách sử dụng @RequestMapping annotation. Dưới đây chỉ là một danh sách hợp lệ các loại argument :

* HttpServletRequest or HttpServleResponse
* Yêu cầu parameters của arbitrary type bằng @RequestParam annotation
* Các thuộc tính modules của arbitrary type bằng @ModelAttribution annotation
* Giá trị Cookie được bao gồm trong yêu cầu đến bằng @CookieValue annotation
* Map hoặc ModelMap, cho phương thức xử lý để thêm các thuộc tính cho model
* Errors hoặc BindingResult cho phương thức xử lý để truy cập ràng buộc và kết quả xác nhận cho comment object
* SessionStatus, để phương thức xử lý thông báo hoàn thành session processing

Khi lớp controller đã chọn một phương thức xử lý thích hợp, nó sẽ gọi phương thức xử lý logic với yêu cầu. Thông thường, logic của controller gọi các service phía sau để xử lý yêu cầu. Ngoài ra, logic của phương thức xử lý có khả năng thêm hoặc xóa thông tin khỏi nhiều input arguments (ví dụ: httpServletRequest, Map, Error hoặc SessionStatus) sẽ tạo thành một phần của luồng Spring MVC.

Sau khi một phương thức xử lý đã xử lý xong yêu cầu, nó ủy quyền điều khiển cho view, đó là đại diện là phương thức xử lý giá trị trả về. Để cung cấp một cách tiếp cận linh hoạt, một phương thức xử lý giá trị trả về không có giá trị không biểu thị cách thực hiện của view (ví dụ: user.jsp hoặc report.pdf) nhưng thay vào đó là logical view( ví dụ: user hoăc report).

Một phương thức xử lý giá trị trả về có thể là một chuỗi, biểu thị logical view name hoặc void, trong trong trường hợp đó, logical view name mặc định được xác định trên cơ sở phương thức xử lý hoặc controller name.

Để truyền thông tin từ controller đến view, nó không liên quan khi phương thức Handler xử lý trả về logical view chuỗi hoặc void vì các input argument của phương thức xử lý sẽ có sẵn cho view. Ví dụ, nếu một phương thức xử lý lấy các Map object và SessionStatus làm tham số đầu vào sửa đổi nội dung bên trong phương thức xử lý các object tương tự này sẽ có thể truy cập được vào view được trả về bởi phương thức xử lý.

Khi lớp controller nhận được một view, nó sẽ phân giải logical view name thành một view cụ thể (ví dụ: user.jsp hoặc report.pdf) bằng trình phân giải view. Một trình phân giải view là một bean được định cấu hình trong ngữ cảnh ứng dụng web thực hiện giao diện ViewResolver. Trách nhiệm của nó là để trả về view cụ thể (HTML, JSP, PDF hoặc loại khác) cho logical view name.

Khi lớp controller đã phân giải view name thành view implementation, theo view implementation thiết kế, nó hiển thị các object (ví dụ: httpServletRequest, Map, Error hoặc SessionStatus) thông qua phương thức xử lý controller. Trách nhiệm của view là hiển thị các object được thêm vào trong phương thức xử lý logic cho user.

Một ứng dụng web được phát triển với Spring MVC được thiết lập giống như một ứng dụng web Java tiêu chuẩn, ngoại trừ việc bạn phải thêm một vài file cấu hình và các thư viện cần thiết dành riêng cho Spring MVC.

Java EE xác định cấu trúc thư mục hợp lệ của ứng dụng web Java được tạo thành từ kho lưu trữ web (file WAR). Ví dụ: bạn phải cung cấp một mô tả triển khai web (ví dụ: web.xml) trong root WEB-INF hoặc một hoặc nhiều lớp triển khai ServletContainerInitializer. Các class file và file JAR cho ứng dụng web này phải được đặt trong thư mục WEB-INF / class và WEB-INF / lib tương ứng.  
Các file bên ngoài thư mục WEB-INF có thể truy cập trực tiếp đến người dùng thông qua URL, vì vậy các file CSS và file hình ảnh phải được đặt ở đó. Khi sử dụng Spring MVC, các file JSP hoạt động như các templates. Chúng được đọc bởi framework để tạo dynamic content, do đó các file nên được đặt trong thư mục WEB-INF để ngăn chặn truy cập trực tiếp vào chúng. Tuy nhiên, một số servers ứng dụng không cho phép các file trong WEB-INF được đọc bởi một ứng dụng web trong nội bộ. Trong trường hợp đó, chỉ có thể đặt chúng bên ngoài thư mục WEB-INF.

### **Tạo file cấu hình**

Mô tả web deployment (web.xml hoặc ServletContainerInitializer là điều cần thiết file cấu hình cho một ứng dụng web Java). Trong file này, bạn xác định servlets cho ứng dụng của bạn và web như thế nào yêu cầu mapped đến chúng. Đối với ứng dụng Spring MVC, bạn chỉ phải xác định một DispatcherServlet instance là front controller cho Spring MVC, mặc dù bạn được phép định nghĩa nhiều hơn một nếu được yêu cầu.

Trong các ứng dụng lớn, có thể thuận tiện khi sử dụng nhiều phiên bản DispatcherServlet. Điều này cho phép các phiên bản DispatcherServlet được chỉ định cho các URL cụ thể, giúp việc quản lý mã dễ dàng hơn và cho phép các thành viên trong nhóm làm việc dựa trên logic ứng dụng mà không cần phải

hiểu theo cách khác.

Trong CourtServletContainerInitializer này, bạn xác định một servlet loại DispatcherServlet. Đây là servlet class cốt lõi trong Spring MVC nhận các yêu cầu web và gửi chúng đến xử lý. Bạn đặt tên servlet này thành court và map tất cả các URL bằng dấu gạch chéo (/), với dấu gạch chéo thể hiện thư mục gốc. Lưu ý rằng mẫu URL có thể được đặt thành các mẫu chi tiết hơn. Trong các ứng dụng lớn hơn, có thể có ý nghĩa hơn khi ủy quyền các mẫu trong số các máy chủ khác nhau, nhưng để đơn giản, tất cả các URL trong ứng dụng được ủy quyền cho các court duy nhất.Để phát hiện ra CourtServletContainerInitializer, bạn cũng phải thêm một file có tên javax.servlet.ServletContainerInitializer vào thư mục META-INF / services. Nội dung của file phải là tên đầy đủ của CourtServletContainerInitializer. Tập tin này được tải bởi servlet container và được sử dụng để bootstrap ứng dụng.

com.apress.springrecipes.court.web.CourtServletContainerInitializer

Cuối cùng, thêm CourtConfiguration class, đây là @Configuration class đơn giản.

### **Tạo Spring MVC Controller**

Một lớp controller dựa trên annotation có thể là một lớp tùy ý không thực hiện giao diện cụ thể hoặc mở rộng một lớp cơ sở cụ thể. Bạn có thể annotation nó với @Controll annotation. Có thể có một hoặc nhiều phương thức xử lý được xác định trong controller để xử lý một hoặc nhiều hành động. Signature của các phương thức xử lý đủ linh hoạt để chấp nhận một loạt các arguments.

@RequestMapping annotation có thể được áp dụng cho cấp độ lớp hoặc cấp độ phương thức. Chiến lược mapping đầu tiên là mapping một mẫu URL cụ thể đến một lớp controller và sau đó là phương thức HTTP riêng biệt đến từng phương pháp xử lý.Trình điều khiển này tạo một đối tượng java.util.Date để lấy ngày hiện tại và sau đó thêm nó vào đầu vào.

Model object là một thuộc tính để view có thể hiển thị nó. Vì vậy, bạn đã kích hoạt quét annotation trên gói com.apress.springrecipes.court, các annotation cho lớp controller được phát hiện khi triển khai.

@Controll annotation định nghĩa lớp là controller Spring MVC controller.@RequestMapping annotation thú vị hơn vì nó chứa các thuộc tính và có thể được khai báo ở cấp phương thức lớp hoặc trình xử lý. Giá trị đầu tiên được sử dụng trong lớp này ("/ welcome") - được sử dụng để chỉ định URL trên đó mà controller có thể thực hiện được, nghĩa là mọi yêu cầu nhận được trên /welcome URL đều được tham gia bởi lớp WelcomeContoder.

Khi lớp yêu cầu được tham gia, nó gọi đến phương thức xử lý HTTP GET mặc định được khai báo trong controller. Lý do cho hành vi này là mọi yêu cầu ban đầu được thực hiện trên một URL đều thuộc loại HTTP GET. Vì vậy, khi controller tham gia yêu cầu trên / welcome URL, sau đó nó sẽ ủy quyền cho phương thức xử lý HTTP GET mặc định để xử lý.

@RequestMapping annotation (phương thức = RequestMethod.GET) được sử dụng để decorate phương thức như bộ điều khiển phương thức xử lý HTTP GET mặc định của controller. Nó đáng để đề cập đến nếu không có phương thức xử lý HTTP GET mặc định nào được khai báo, một ServletException được đưa ra. Do đó, nó rất quan trọng đối với Spring MVC controller để có tối thiểu một tuyến URL và phương thức xử lý HTTP GET mặc định.

Một biến thể khác của cách tiếp cận này có thể là khai báo cả hai giá trị tuyến URL và phương thức xử lý HTTP GET mặc định trong @RequestMapping annotation được sử dụng ở cấp phương thức.

Thuộc tính value chỉ ra URLphương thức xử lý được mapping và thuộc tính phương thức định nghĩa phương thức xử lý là controller phương thức HTTP GET mặc định. Cuối cùng, cũng có một số annotation tiện lợi như @ GetMapping, @PostMapping, v.v., để giảm thiểu cấu hình. @GetMapping annotation làm cho lớp ngắn hơn một chút và có thể dễ đọc hơn. Tuy nhiên, một controller có thể gọi các service back-end để xử lý nghiệp vụ.

### **Tạo JSP Views**

Spring MVC hỗ trợ nhiều kiểu view cho các công nghệ khác nhau. Chúng bao gồm các file, HTML, PDF, Excel (XLS), nguồn cấp dữ liệu XML, JSON, Atom và RSS, JasperReports và các view của bên thứ ba khác.

Trong một ứng dụng Spring MVC, các view là các mẫu phổ biến nhất được viết bằng JSTL. Khi mà DispatcherServlet, được định nghĩa trong một ứng dụng, file webDB của Google, nhận được một view name được trả về từ một trình xử lý, nó phân giải logical view name thành một view implementation để hiển thị. Ví dụ, bạn có thể cấu hình bean InternalResourceViewResolver, trong trường hợp này trong CourtConfiguration, của một trang web ứng dụng bối cảnh của ứng dụng để phân giải tên xem thành các file trong thư mục / WEB-INF / jsp /.

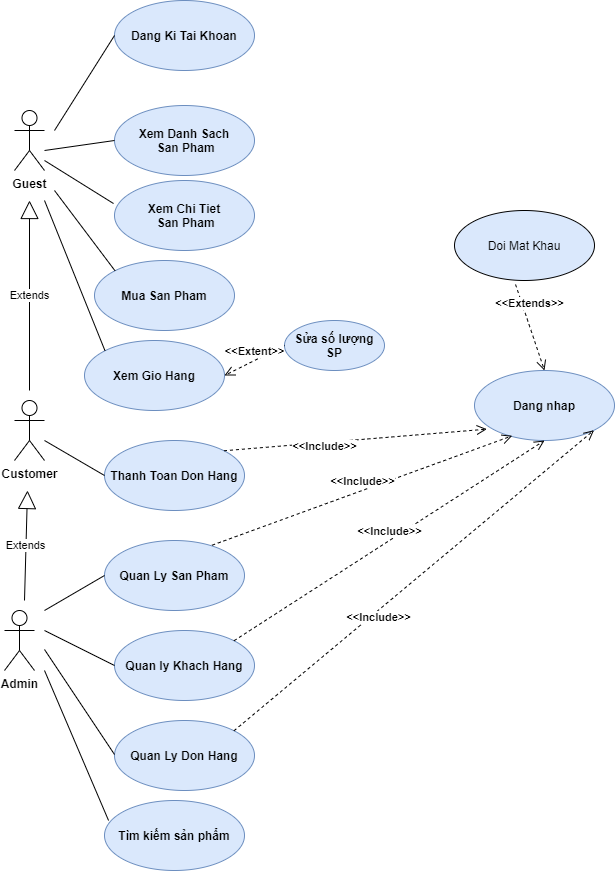
### **Triển khai web application**

Trong quy trình phát triển ứng dụng Web, nên cài đặt ứng dụng local Java EE server đi kèm với một web container cho mục đích thử nghiệm và gỡ lỗi. Để dễ dàng cấu hình và triển khai, chúng tôi đã chọn Apache Tomcat 8.5.x làm web container.

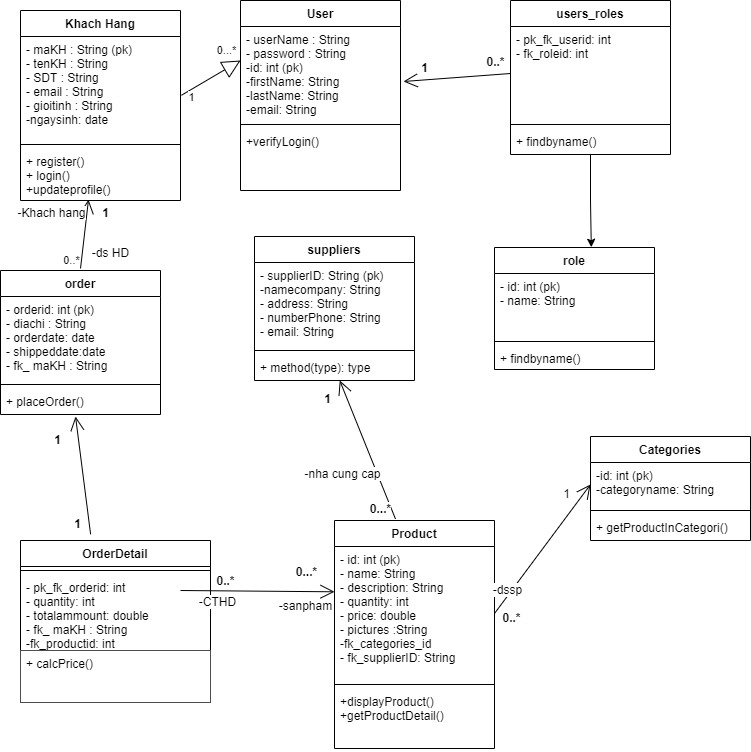
Thư mục triển khai cho web container này nằm trong thư mục webapps. Theo mặc định, Tomcat lắng nghe trên port 8080 và triển khai các ứng dụng vào một ngữ cảnh có cùng tên của một ứng dụng WAR.

# **Chương 3: phân tích**

## **3.1.Use case tổng quát**



## **3.2.Class Diagram**

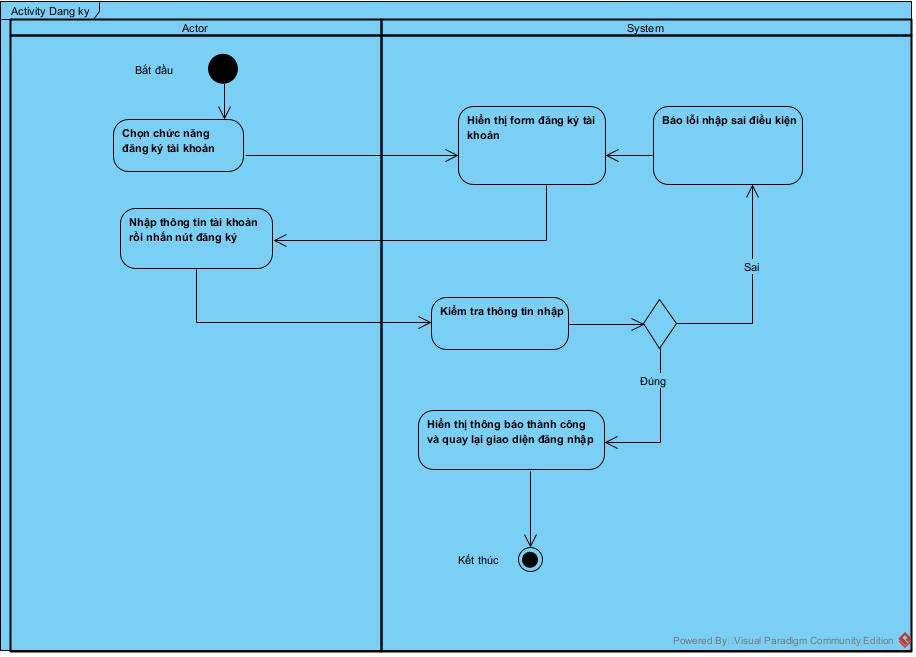


## **3.3.Use case Đăng kí tài khoản**

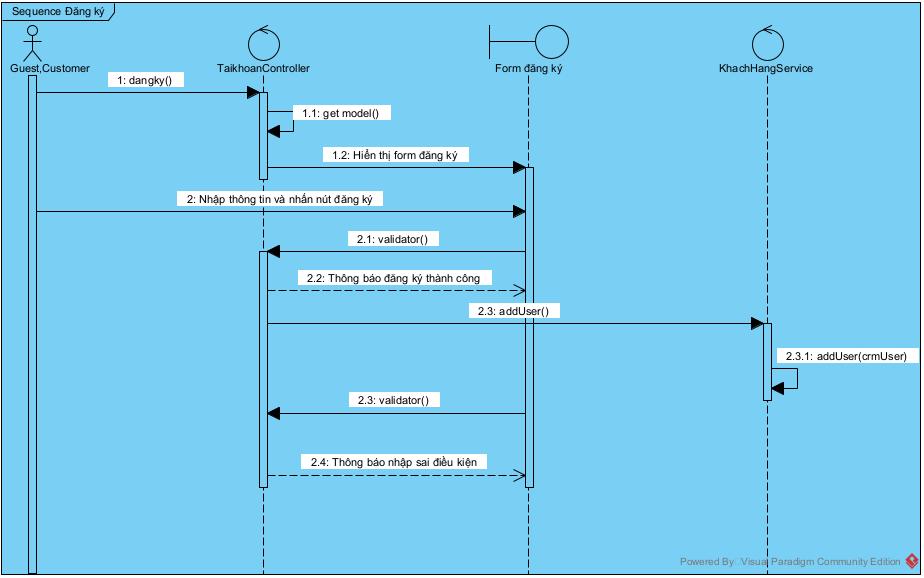
**3.3.1 Đặc tả use case Đăng ký**

|  |  |  |  |
| --- | --- | --- | --- |
| Actor(Tác nhân) | Customer,employee,admin | | |
| Mô tả | Cho phép người dùng đăng kí tài khoản | | |
| Hậu điều kiện | Hệ thống gửi thông tin lên server, gmail của khách hàng và lưu thông tin vào cơ sở dữ liệu | | |
|  |  | Actor | System |
| Luồng sự kiện chính | 1 | Chọn chức năng đăng kí tài khoản |  |
| 2 |  | Hiển thị form đăng ký tài khoản |
| 3 | Nhập thông tin |  |
| 4 | Chọn chức năng đăng kí |  |
| 5 |  | Kiểm tra thông tin |
| 6 |  | Hiển thị thông báo thành công và quay lại giao diện đăng nhập |
| Luồng sự kiện phụ | 5.1 |  | Nếu thông tin không hợp lệ, hệ thông đưa ra thông báo nhập sai |
| 5.2 |  | Hệ thống quay lại bước 2 |

3.3.2.Sơ đồ Activity Đăng ký:



3.3.3. Sequence Đăng ký:

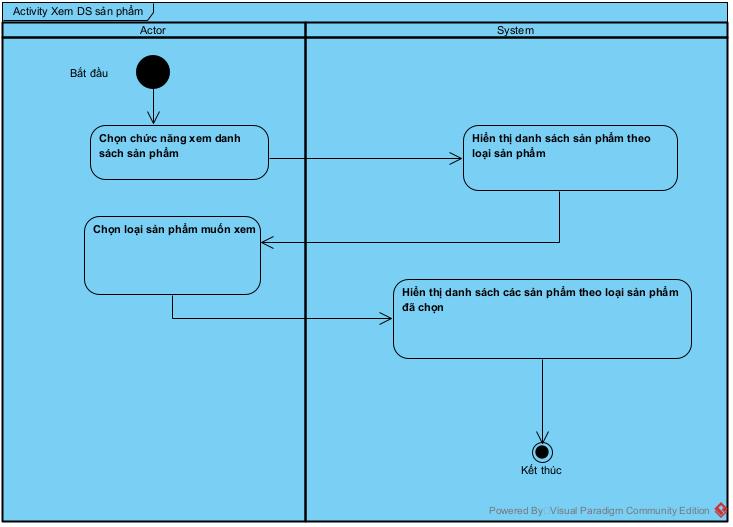


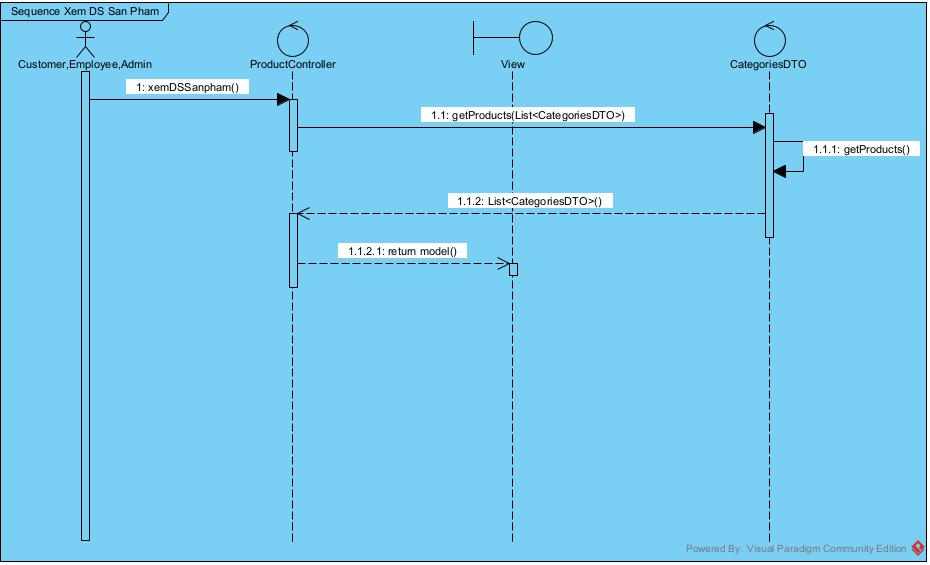
## **3.4.Use case Xem danh sách sản phẩm**

3.4.1 Đặc tả use case

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | customer, admin, employee | | |
| Mô tả | Cho phép người dùng xem danh sách sản phẩm | | |
| Luồng sự kiện chính |  | Actor | System |
|  | 1 | Chọn chức năng xem danh sách sản phẩm trên menu |  |
| 2 |  | Hiển thị danh sách theo từng loại sản phẩm |
| 3 | Chọn loại sản phẩm cần xem |  |
| 4 |  | Hiển thị giao diện danh sách các sản phẩm của loại đã chọn |

3.4.2.Sơ đồ Activity Xem DS sản phẩm:



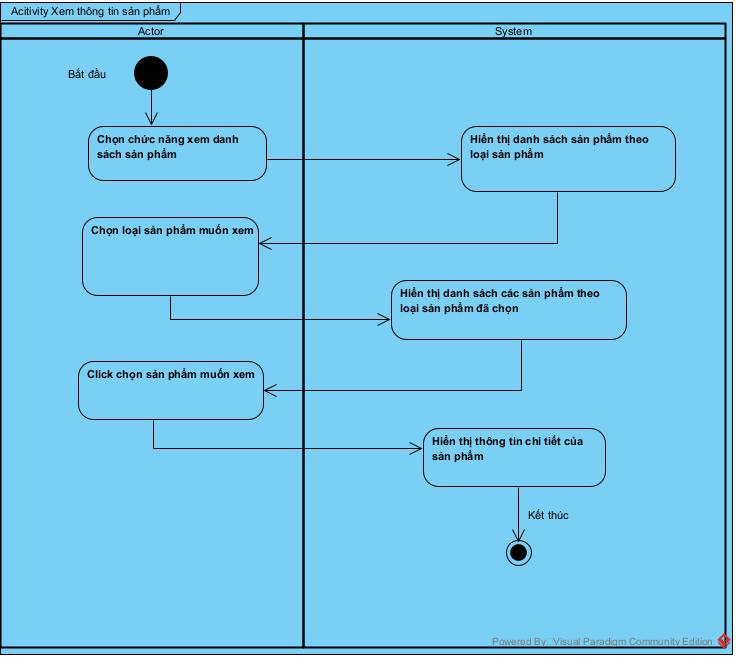
3.4.3.Sequence xem DS sản phẩm:  


## **3.5.Use case xem thông tin chi tiết của sản phẩm:**

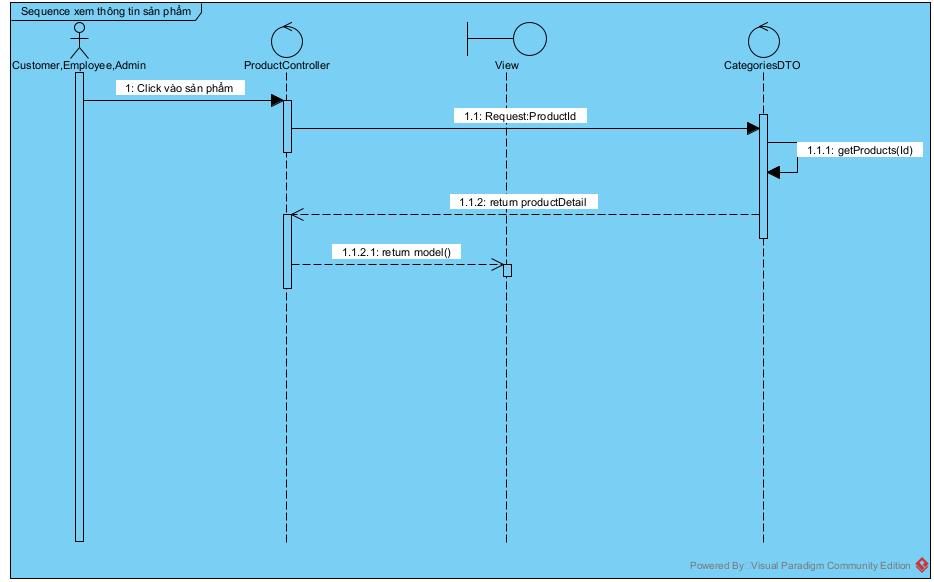
3.5.1.Đặc tả use case

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | customer,employee, admin | | |
| Mô tả | Cho phép người dùng xem thông tin chi tiết của sản phẩm | | |
| Luồng sự kiện chính |  | Actor | System |
|  | 1 | Chọn chức năng xem danh sách sản phẩm trên menu |  |
| 2 |  | Hiển thị danh sách theo từng loại sản phẩm |
| 3 | Chọn loại sản phẩm cần xem |  |
| 4 |  | Hiển thị giao diện danh sách các sản phẩm của loại đã chọn |
| 5 | Chọn sản phẩm cần xem |  |
| 6 |  | Hiện thị giao diện thông tin chi tiết của sản phẩm đã chọn |
| 7 | Xem thông tin chi tiết của sản phẩm |  |

3.5.2.Activity xem thông tin chi tiết của sản phẩm:



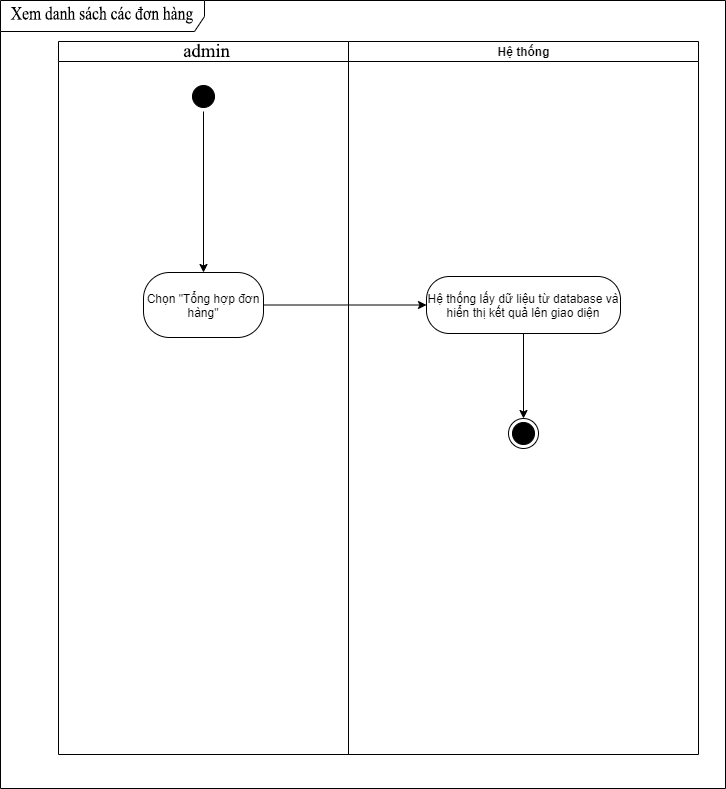
3.5.3.Sequence Xem thông tin chi tiết sản phẩm

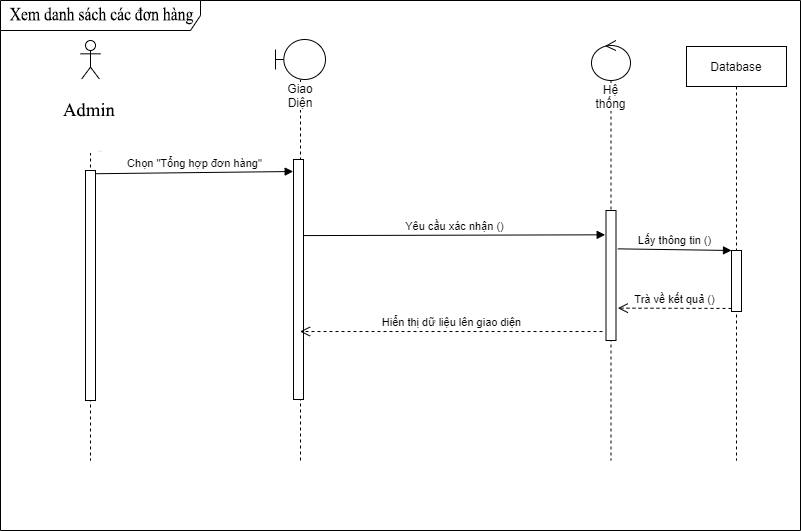


## **3.6. use case Xem danh sách các đơn hàng**

3.6.1. Đặc tả use case Xem danh sách các đơn hàng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Admin,Employee,Customer | | |
| Mô tả | Cho phép người dùng xem danh sách các đơn hàng | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống | | |
|  |  | Tác nhân | Hệ thống |
| Luồng sự kiện chính | 1 | Chọn chức năng “Tổng hợp đơn hàng” |  |
| 2 |  | Hiển thị form danh sách đơn hàng |

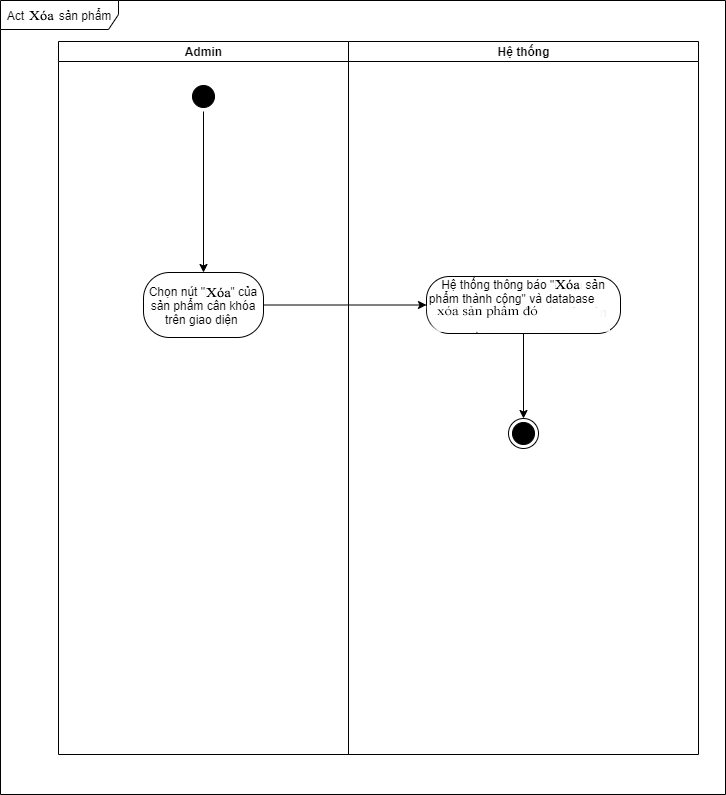
3.6.2.Activity xem danh sách đơn hàng

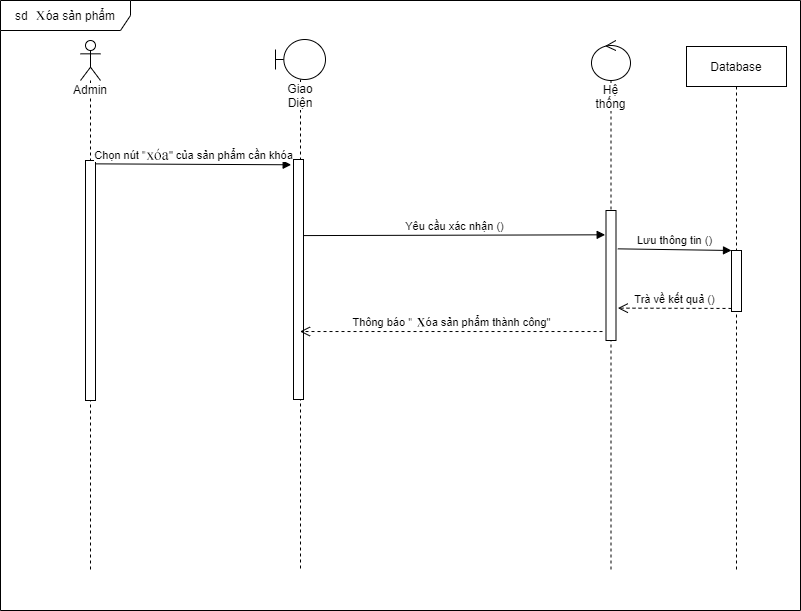
3.6.3.Sequence Xem danh sách các đơn hàng

## **3.7.Use case xóa sản phẩm**

3.7.1.Đặc tả use case Xóa sản phẩm

|  |  |
| --- | --- |
| Mô tả: | Admin, người dùng có gian hàng có thể xóa một sản phẩm có trong hệ thống |
| Tác nhân: | Admin, employee,customer |
| Điều kiện trước: | Đăng nhập vào hệ thống thành công  Chọn chức năng “Quản lý sản phẩm” (Admin)  Chọn chức năng “Danh sách sản phẩm” (Người dùng có gian hàng) |
| Điều kiện sau: | Hệ thống thông báo xóa sản phẩm thành công  Hệ thống xóa sản phẩm khỏi cơ sở dữ liệu |
| Luồng sự kiện chính (Basic flows) | 1. Chọn nút “Xóa” của sản phẩm cần xóa có trên giao diện 2. Hệ thống hiển thị thông báo xóa sản phẩm thành công và database xóa sản phẩm đó. |
| Luồng sự kiện phụ (Alternative Flows): |  |

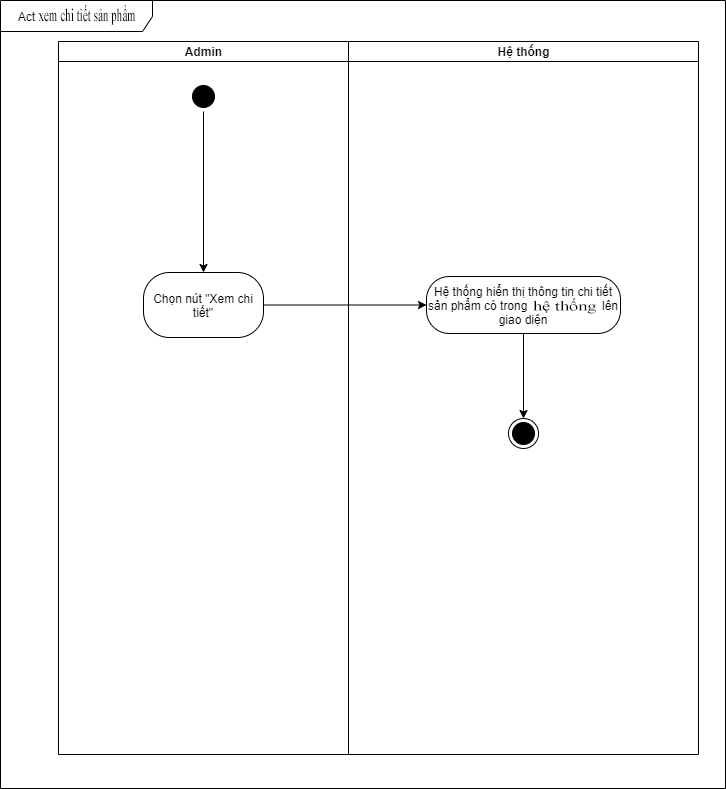
3.7.2.Activity Xóa sản phẩm:

3.7.3.Sequence xóa sản phẩm 

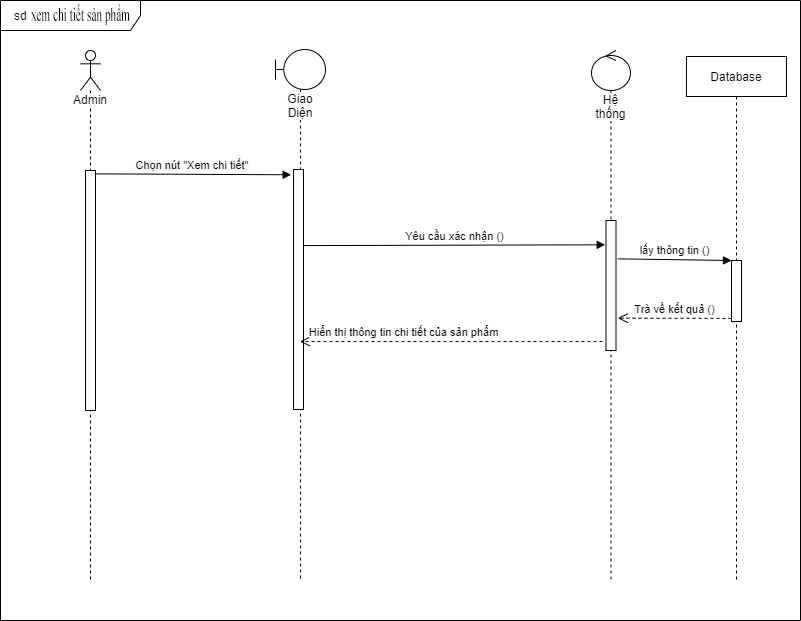
## **3.8 Use case xem thông tin chi tiết từng loại sản phẩm**

3.8.1.Đặc tả use case xem thông tin chi tiết từng loại sản phẩm

|  |  |
| --- | --- |
| Mô tả: | Admin xem chi tiết sản phẩm có trong hệ thống |
| Tác nhân: | Admin |
| Điều kiện trước: | Admin đăng nhập vào hệ thống thành công |
| Điều kiện sau: | Hiển thị tất cả các thông tin chi tiết sản phẩm |
| Luồng sự kiện chính (Basic flows) | 1. Admin chọn nút “Xem chi tiết” 2. Hệ thống thị tất cả các thông tin chi tiết sản phẩm lên giao diện |
| Luồng sự kiện phụ (Alternative Flows): |  |

3.8.2.Activity xem thông tin chi tiết loại sản phẩm

3.8.3.Sequence xem thông tin chi tiết loại sản phẩm

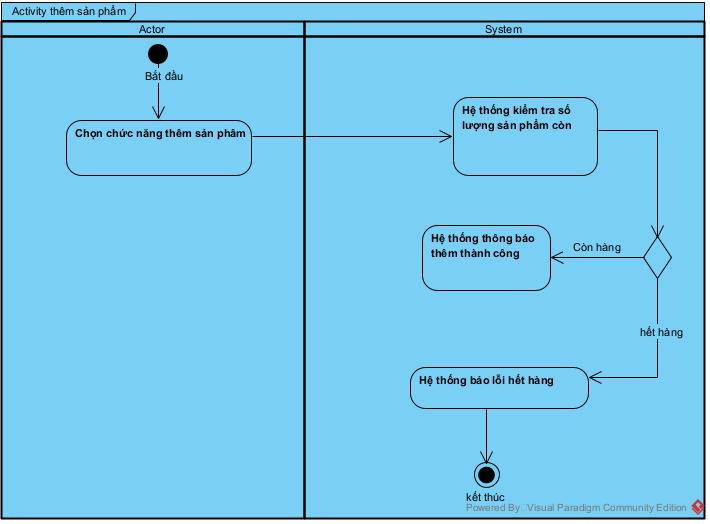


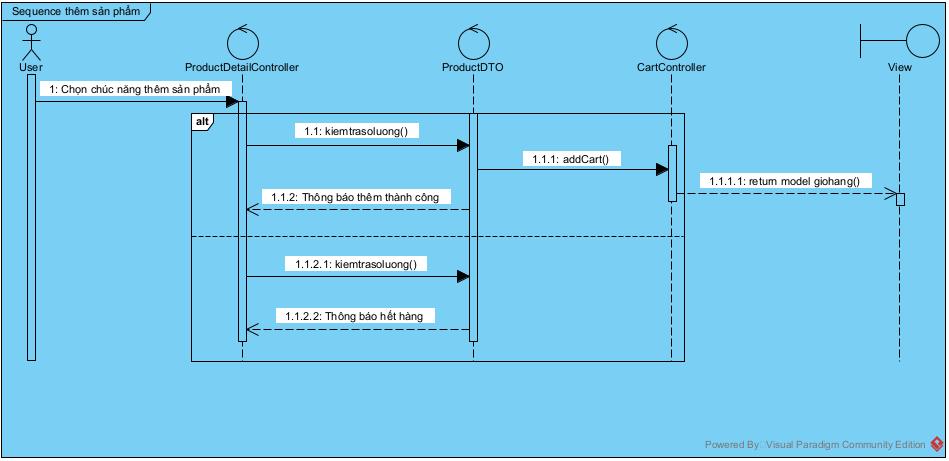
## **3.9.Use case Thêm sản phẩm vào giỏ hàng:**

3.9.1.Đặc tả

|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Employee,Customer,Admin | | |
| Mô tả | Cho phép khách hàng them sản phẩm đã mình muốn vào giỏ hàng | | |
|  |  | Actor | System |
| Luồng sự kiện chính | 1 | Chọn chức năng thêm sản phẩm |  |
| 2 |  | Hệ thống kiểm tra số lượng sản phẩm còn |
| 3 |  | Hệ thống thông báo thêm thành công |
| Luồng sự kiện phụ | 4 |  | Hệ thống báo lỗi hết hàng |

3.9.2.Activity thêm sản phẩm:

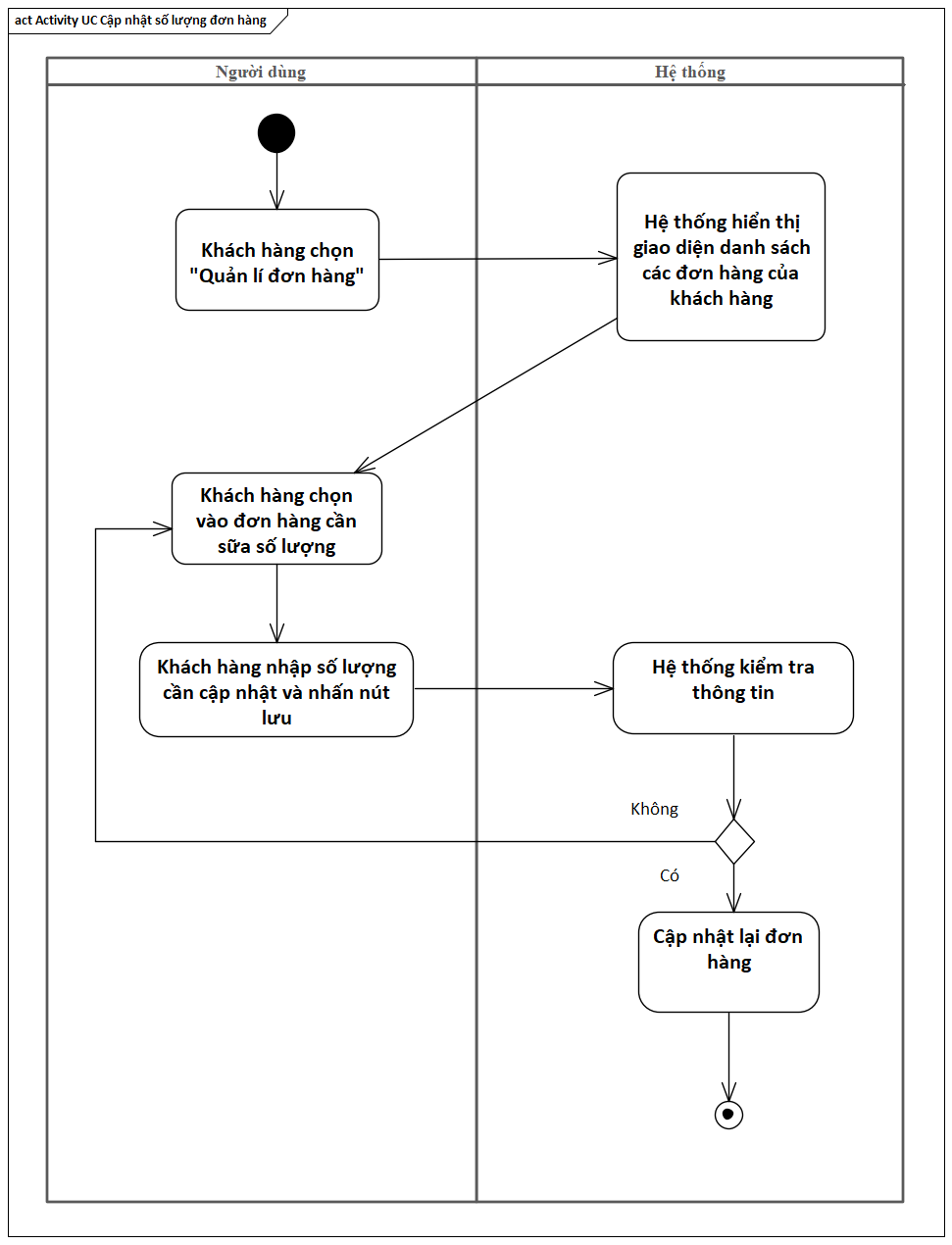


3.9.3.Sequence thêm sản phẩm:  


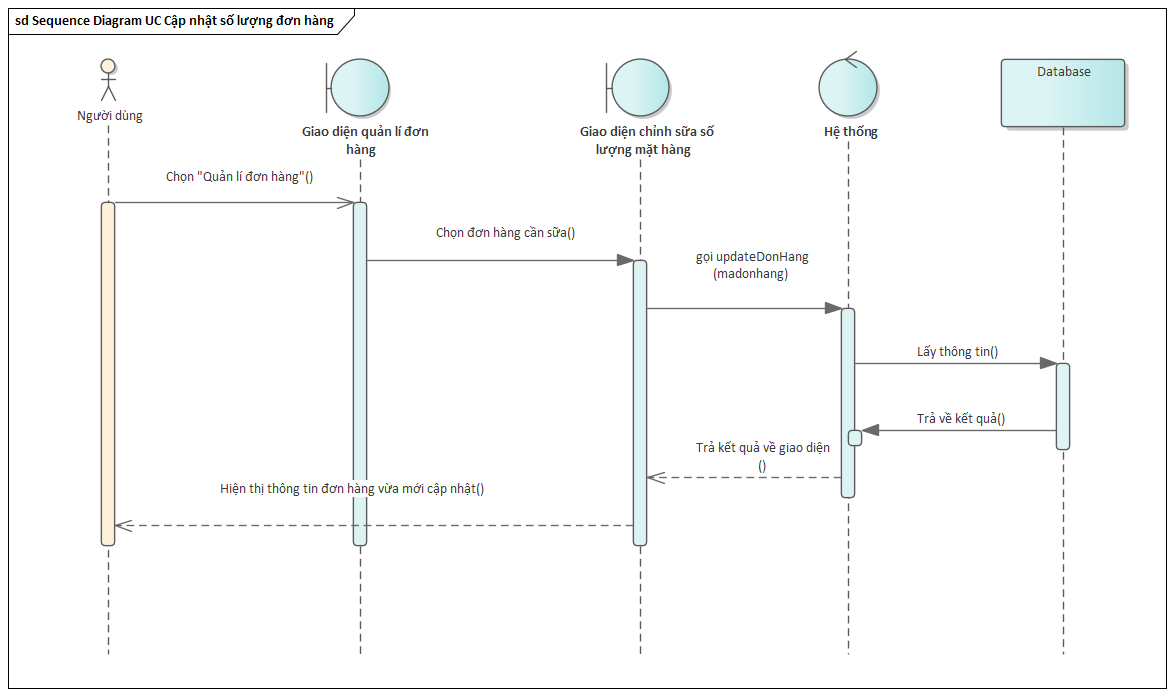
## **3.10 Use case Cập nhật số lượng của mặt hàng trong đơn hàng**

3.10.1. Đặc tả Use case Cập nhật số lượng của mặt hàng trong đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Khách hàng | | |
| Mô tả | Cho phép khách hàng cập nhật số lượng của mặt hàng đã thêm vào giỏ hàng | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống  Người dùng phải có đơn hàng trước | | |
|  |  | Tác nhân | Hệ thống |
| Luồng sự kiện chính | 1 | Người dùng chọn chức năng “Đơn hàng” |  |
| 2 |  | Hệ thống hiển thị giao diện đơn hàng |
| 3 | Người dùng chọn vào đơn hàng cần cập nhật số lượng |  |
| 4 | Người dùng nhập số lượng sản phẩm cần cập nhật nhấn nút “Lưu” |  |
|  |  | Hệ thống lưu đơn hàng mới vào database |
| Luồng sự kiện phụ |  | | |

3.10.2.Activity Cập nhật số lượng của mặt hàng trong đơn hàng

3.10.3.Sequence Cập nhật số lượng của mặt hàng trong đơn hàng

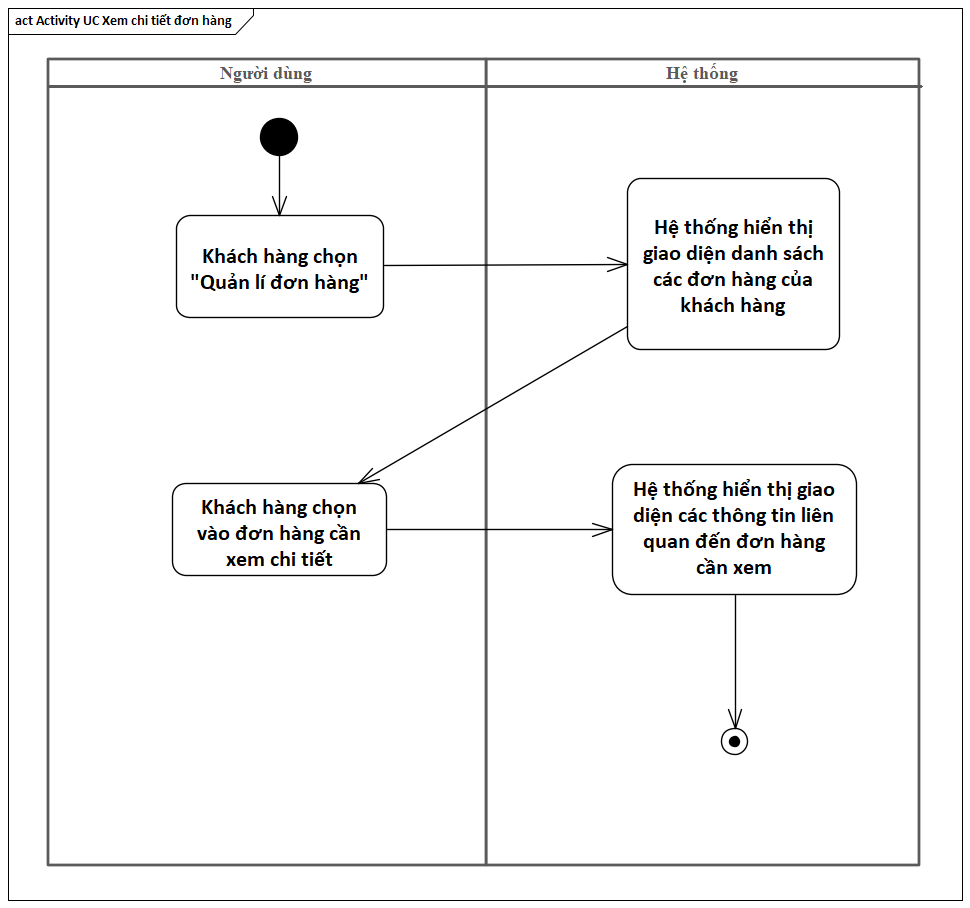


## **3.11 Use case Xem chi tiết các đơn hàng**

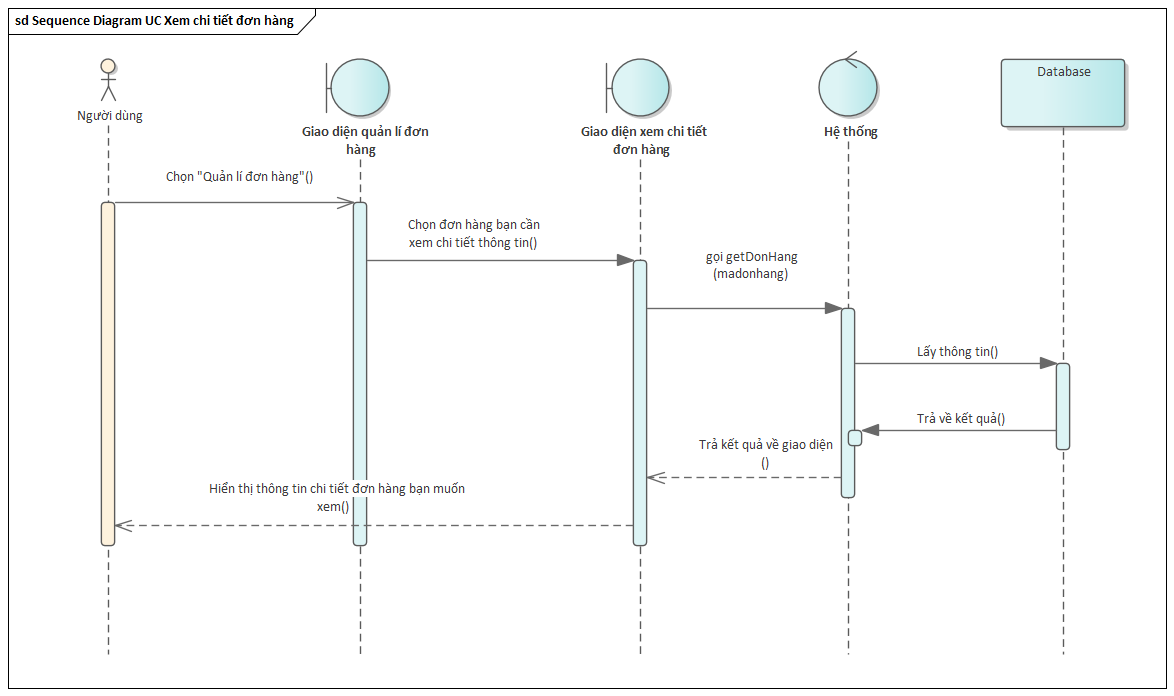
3.11.1.Đặc tả Xem chi tiết các đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Khách hàng | | |
| Mô tả | Cho phép khách xem được các thông tin đơn hàng đã đặt | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống  Người dùng phải có đơn hàng trước | | |
|  |  | Tác nhân | Hệ thống |
| Luồng sự kiện chính | 1 | Người dùng chọn vào đơn hàng cần xem chi tiết |  |
| 2 |  | Hệ thống lấy dữ liệu từ database |
| 3 |  | Hệ thống hiển thị các thông tin liên quan đến đơn hàng bạn đã đặt |
| Luồng sự kiện phụ | Không có đơn hàng nào bạn đặt | | |

3.11.2.Activity Xem chi tiết các đơn hàng



3.11.3.Sequence Xem chi tiết các đơn hàng

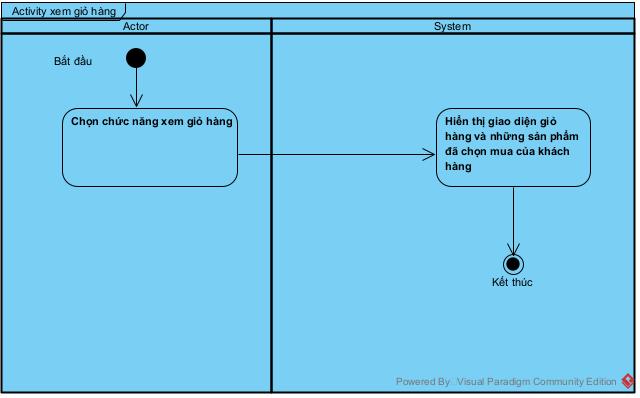


## **3.12.Use case Xem giỏ hàng:**

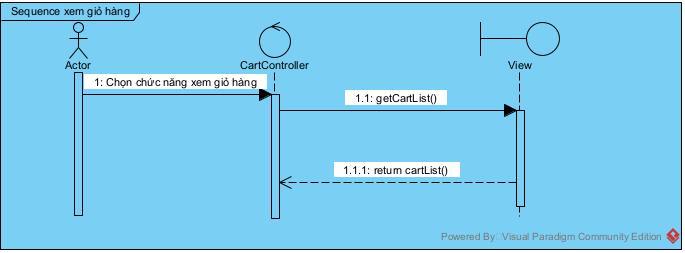
3.12.1.Đặc tả

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Guest, customer, admin | | |
| Mô tả | Cho phép khách hàng xem giỏ hàng của mình | | |
| Hậu điều kiện | Hệ thống hiển thị và cập nhật những sản phẩm mà khách hàng đã chọn mua | | |
|  |  | Actor | System |
| Luồng sự kiện chính | 1 | Chọn chức năng xem giỏ hàng |  |
| 2 |  | Hiển thị giao diện giỏ hàng và thông tin những sản phẩm đã chọn mua của khách hàng |

3.12.2.Activity xem giỏ hàng



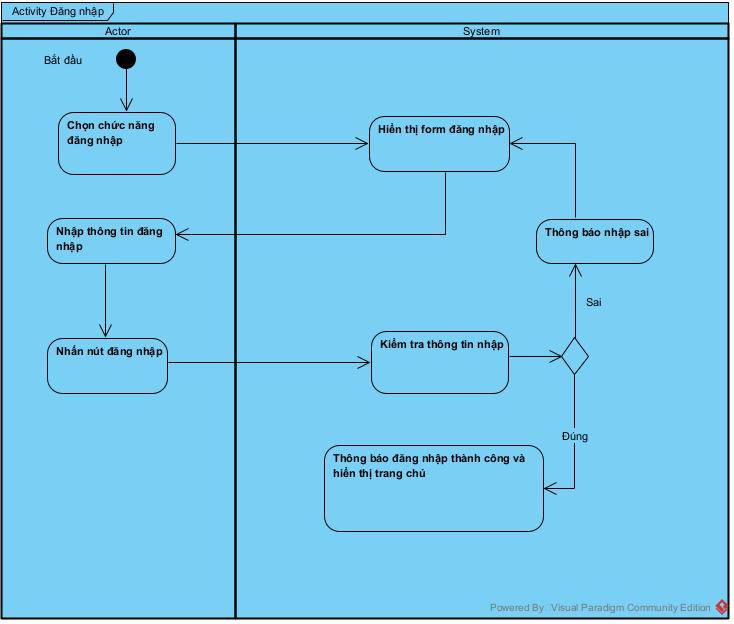
3.12.3.Sequence Xem giỏ hàng:



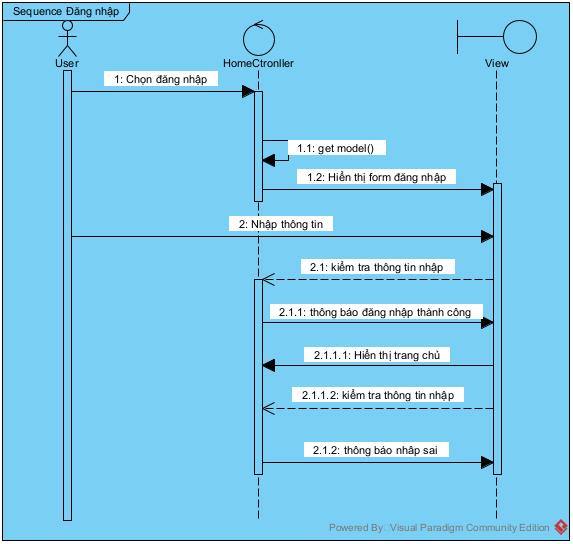
## **3.13.Use case Đăng nhập:**

3.13.1.Đặc tả

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Customer, Admin, Employee | | |
| Mô tả | Cho phép người dùng thanh toán trực tuyến | | |
| Tiền diều kiện | Người dùng đã có tài khoản | | |
|  |  | Actor | System |
| Luồng sự kiện chính | 1 | Chọn chức năng đăng nhập |  |
| 2 |  | Hiển thị form đăng nhâp |
| 3 | Nhập thông tin đăng nhập |  |
| 4 | Nhấn nút đăng nhập |  |
| 5 |  | Kiểm tra thông tin nhập |
| 6 |  | Thông báo đăng nhập thành công và hiển thị trang chủ |
| Luồng sự kiện phụ | 5.1 |  | Thông báo đăng nhập sai |
|  | 5.2 |  | Quay lại bước 2 |

3.13.2.Activity Đăng nhập

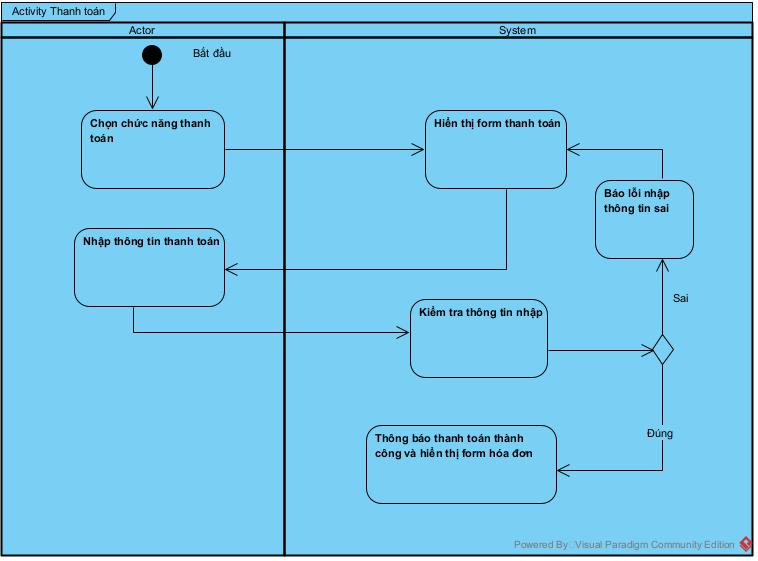
3.13.3.Sequence Đăng nhập:



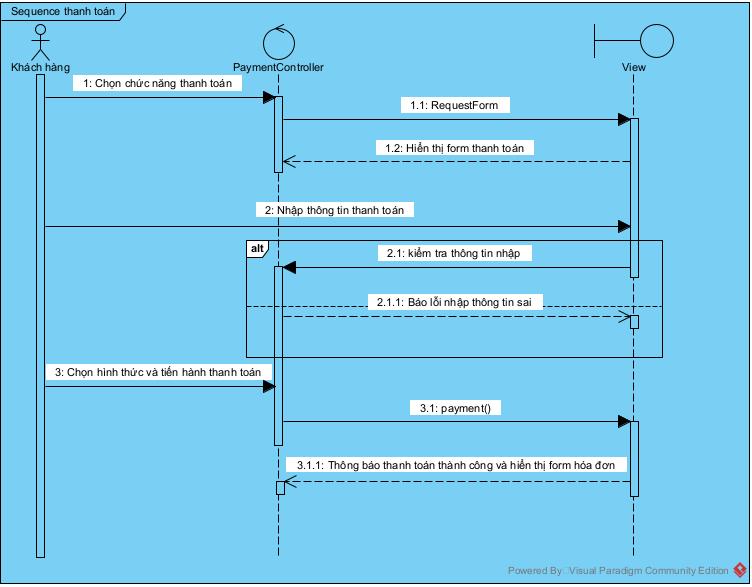
## **3.14. Use case Thanh toán**

3.14.1.Đặc tả:

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Customer | | |
| Mô tả | Cho phép người dùng thanh toán | | |
| Tiền điều kiện | Người dùng đã đăng nhập tài khoản và đã vào giao diện giỏ hàng | | |
|  |  | Actor | System |
| Luồng sự kiện chính | 1 | Chọn chức năng thanh toán |  |
| 2 |  | Hiển thị form thanh toán |
| 3 | Nhập thông tin thanh toán |  |
| 4 |  | Kiểm tra thông tin nhập |
| 5 | Chọn hình thức thanh toán và tiến hành thanh toán |  |
| 6 |  | Thông báo thanh toán thành công và hiển thị form hóa đơn |
| Luồng sự kiện phụ | 4.1 |  | Báo lỗi nhập thông tin sai |
| 4.2 |  | Quay lại bước 2 |

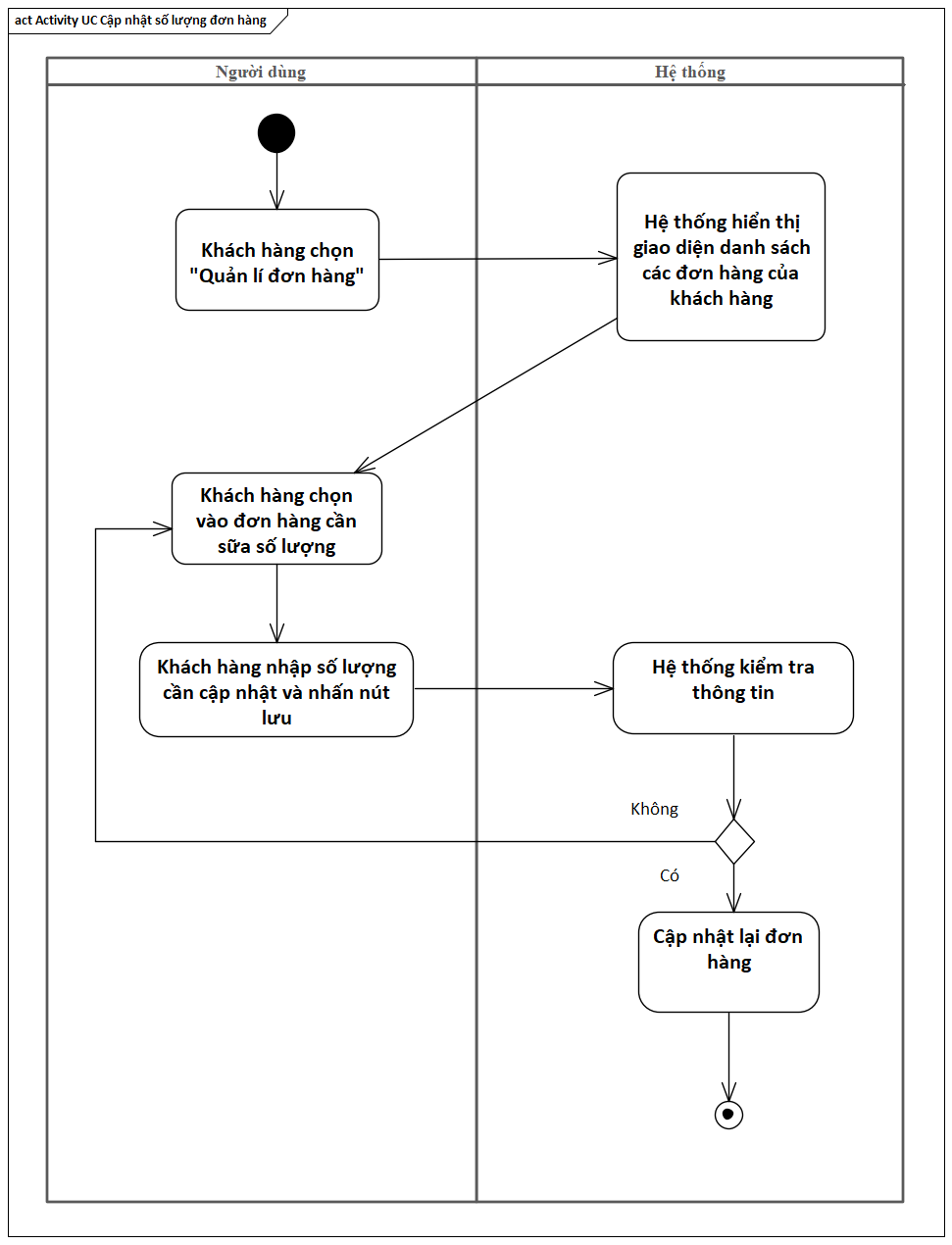
3.14.2.Activity thanh toán  


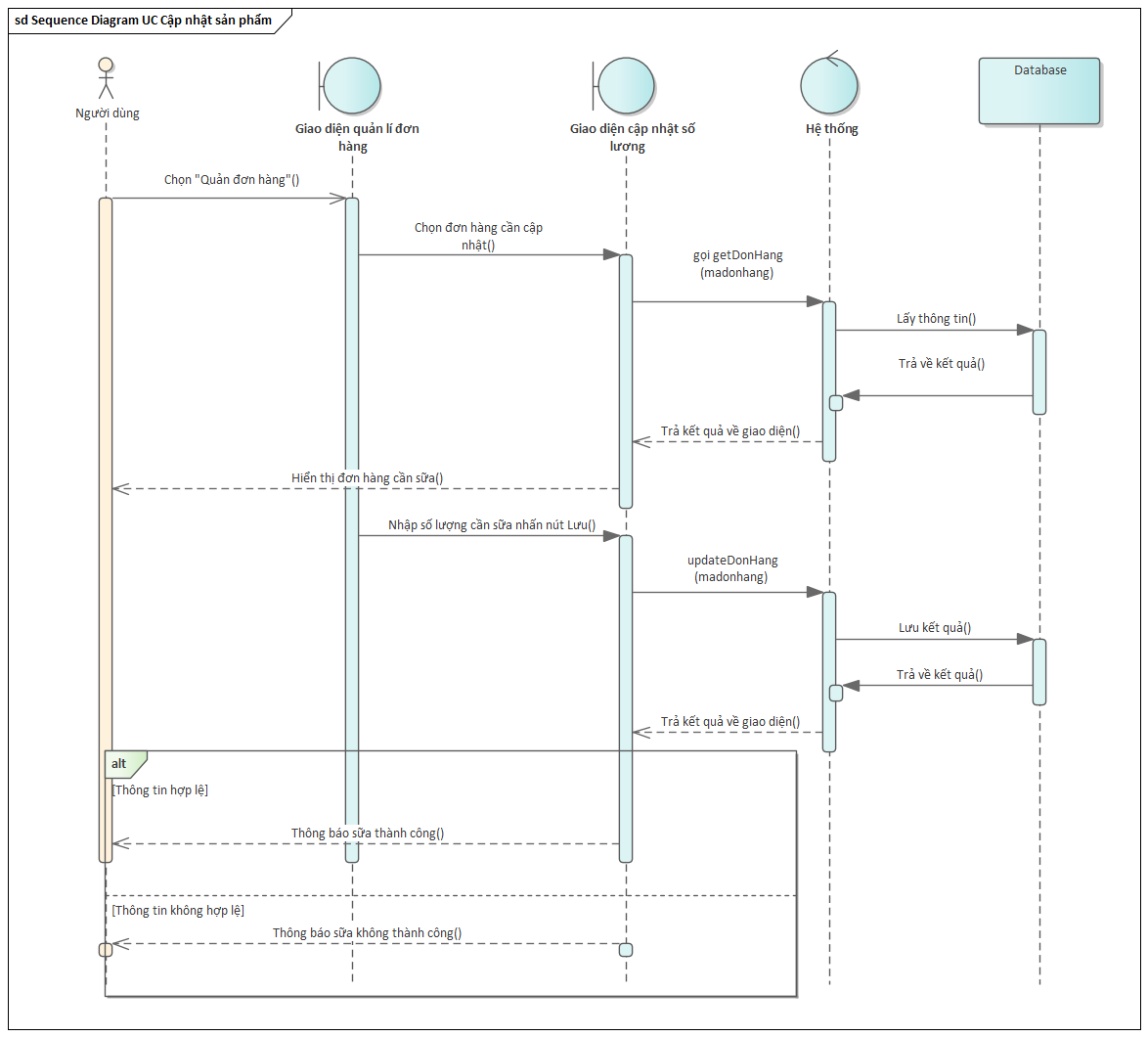
3.14.3.Sequence thanh toán:



## **3.15.Use case Cập nhật số lượng của mặt hàng trong đơn hàng trực tuyến**

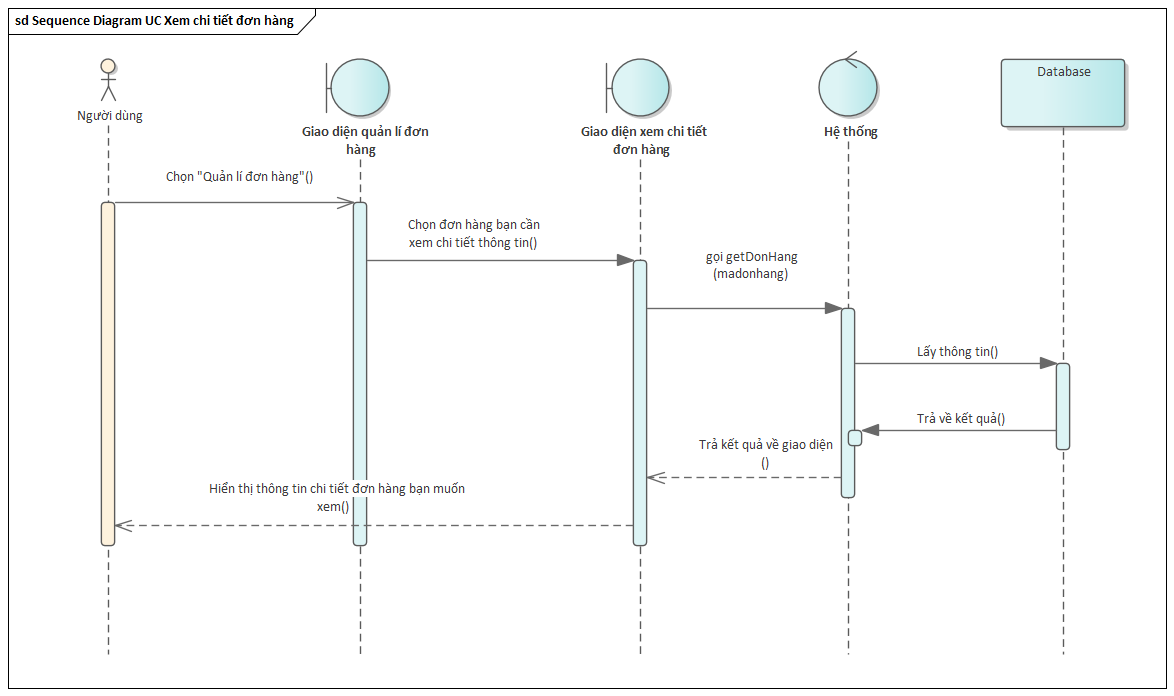
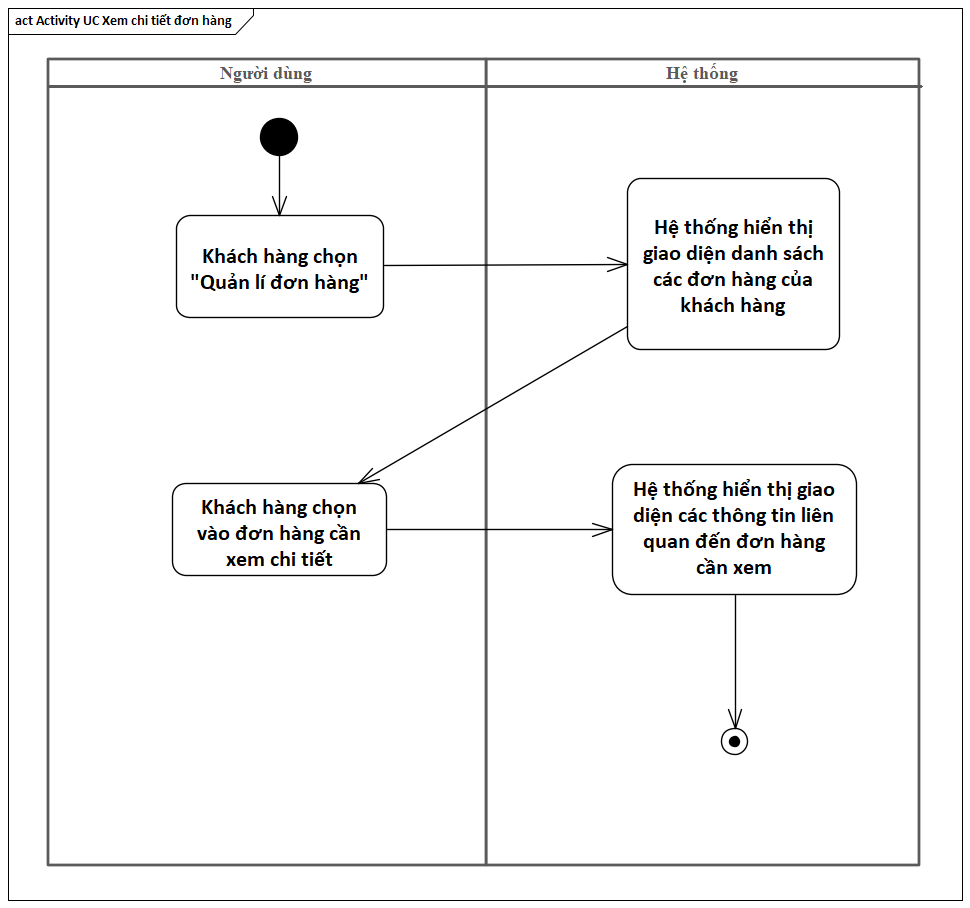
|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Khách hàng | | |
| Mô tả | Cho phép khách hàng cập nhật số lượng của mặt hàng đã thêm vào giỏ hàng | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống  Người dùng phải có đơn hàng trước | | |
| Luồng sự kiện chính |  | Tác nhân | Hệ thống |
|  | 1 | Người dùng chọn chức năng “Đơn hàng” |  |
| 2 |  | Hệ thống hiển thị giao diện đơn hàng |
|  | 3 | Người dùng chọn vào đơn hàng cần cập nhật số lượng |  |
|  | 4 | Người dùng nhập số lượng sản phẩm cần cập nhật nhấn nút “Lưu” |  |
|  |  |  | Hệ thống lưu đơn hàng mới vào database |
| Luồng sự kiện phụ |  | | |





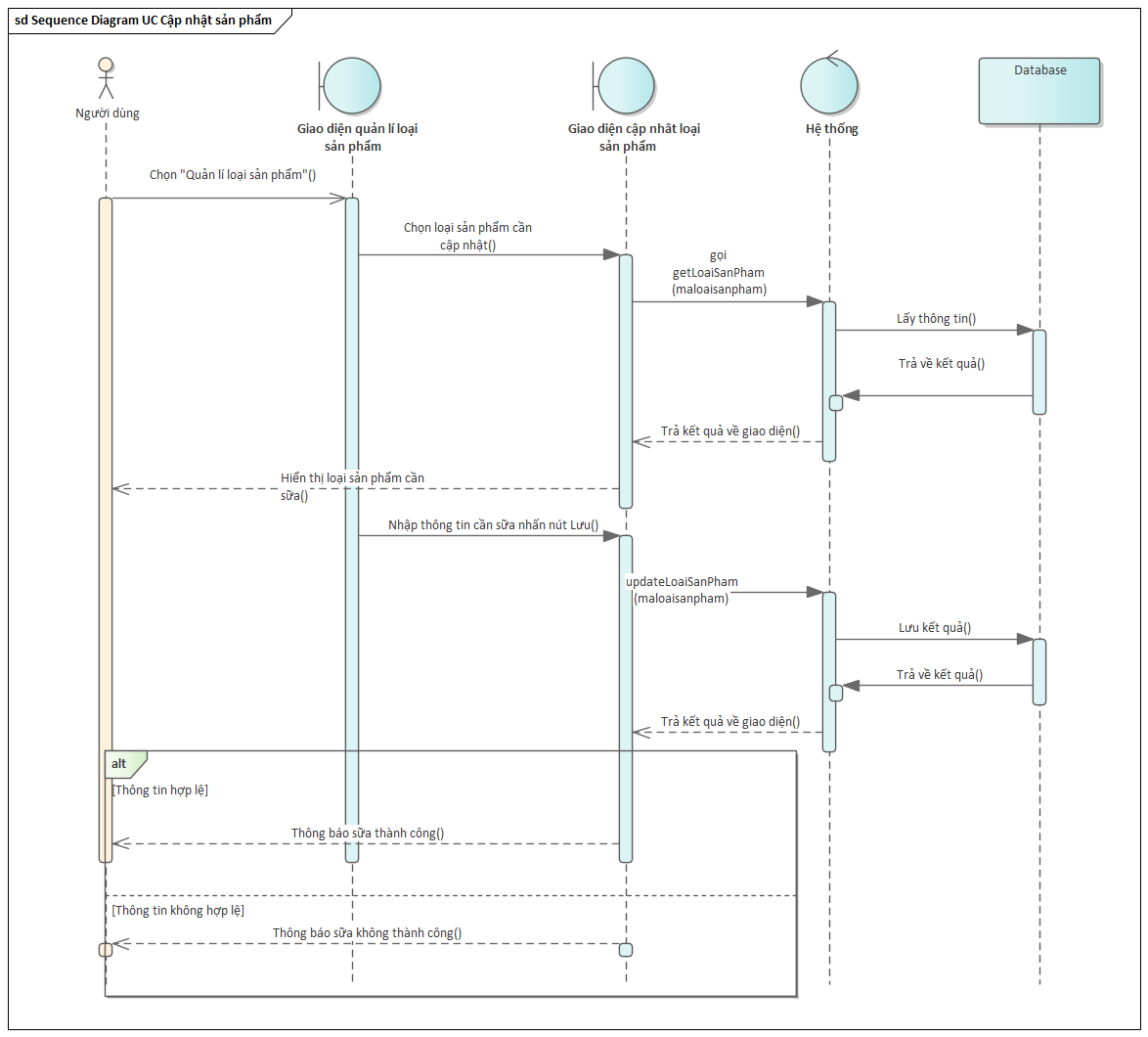
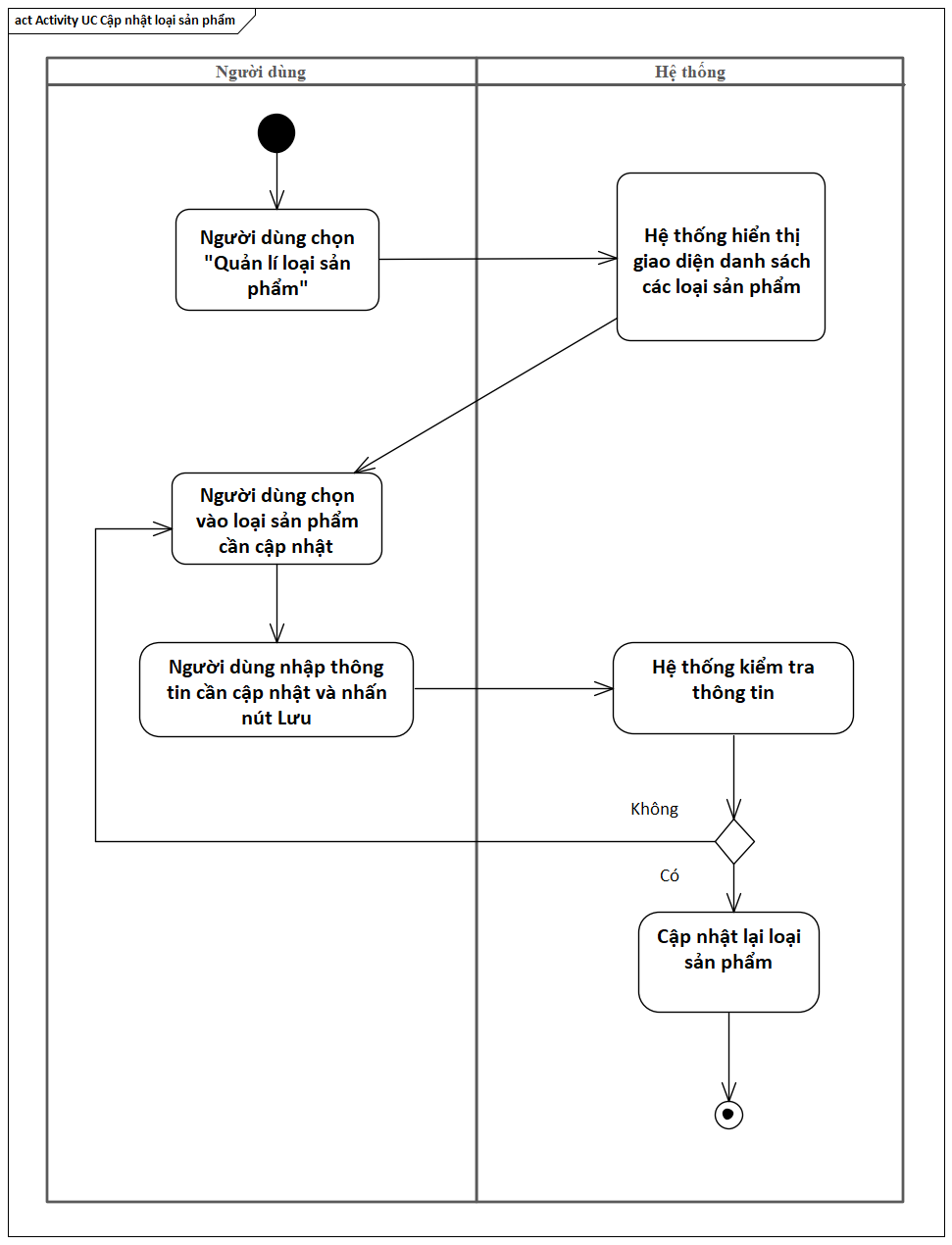
## **3.16 Use case Xem chi tiết các đơn hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Khách hàng | | |
| Mô tả | Cho phép khách xem được các thông tin đơn hàng đã đặt | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống  Người dùng phải có đơn hàng trước | | |
| Luồng sự kiện chính |  | Tác nhân | Hệ thống |
|  | 1 | Người dùng chọn vào đơn hàng cần xem chi tiết |  |
| 2 |  | Hệ thống lấy dữ liệu từ database |
|  | 3 |  | Hệ thống hiển thị các thông tin liên quan đến đơn hàng bạn đã đặt |
| Luồng sự kiện phụ | Không có đơn hàng nào bạn đặt | | |



## **3.17 Use case Cập nhật thông tin loại sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Admin | | |
| Mô tả | Cho phép admin cập nhật thông tin của loại sản phẩm | | |
| Tiền điều kiện | Admin đã đăng nhập vào hệ thống  Đã có loại sản phẩm trong hệ thống | | |
| Luồng sự kiện chính |  | Tác nhân | Hệ thống |
|  | 1 | Người dùng chọn chức năng “Quản lí loại sản phẩm” |  |
| 2 |  | Hệ thống hiển thị giao diện quản lí loại sản phẩm |
|  | 3 | Người dùng chọn vào loại sản phẩm cần cập nhật |  |
|  | 4 | Người dùng nhập thông tin cần cập nhật nhấn nút “Lưu” |  |
|  |  |  | Hệ thống cập nhật loại sản phẩm vào database |
| Luồng sự kiện phụ |  | | |



## **3.18 Use case Cập nhật thông tin sản phẩm**

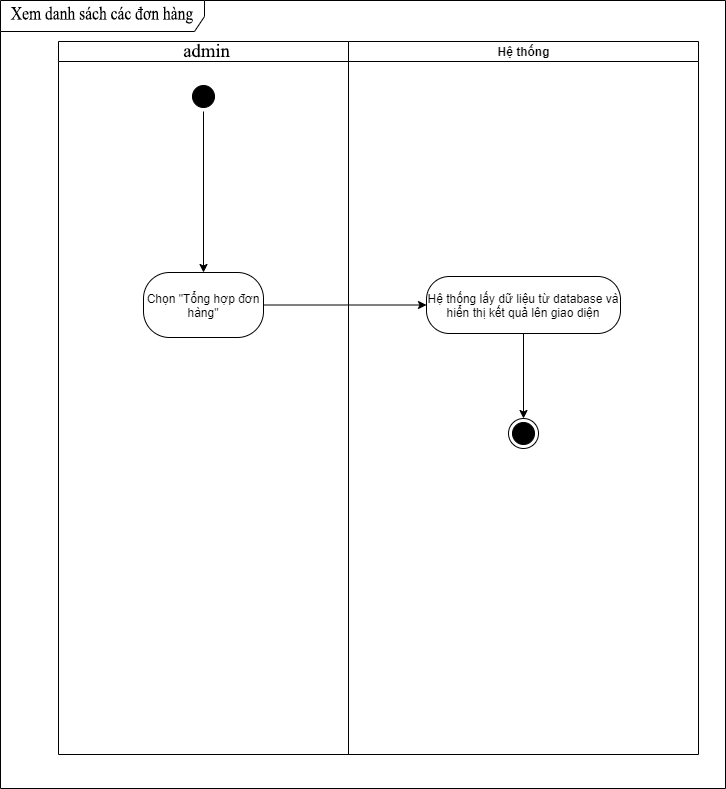
|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Admin | | |
| Mô tả | Cho phép admin cập nhật thông tin của sản phẩm | | |
| Tiền điều kiện | Admin đã đăng nhập vào hệ thống  Đã có sản phẩm trong hệ thống | | |
| Luồng sự kiện chính |  | Tác nhân | Hệ thống |
|  | 1 | Người dùng chọn chức năng “Quản lí sản phẩm” |  |
| 2 |  | Hệ thống hiển thị giao diện quản lí sản phẩm |
|  | 3 | Người dùng chọn vào sản phẩm cần cập nhật |  |
|  | 4 | Người dùng nhập thông tin cần cập nhật nhấn nút “Lưu” |  |
|  |  |  | Hệ thống cập nhật sản phẩm vào database |
| Luồng sự kiện phụ |  | | |

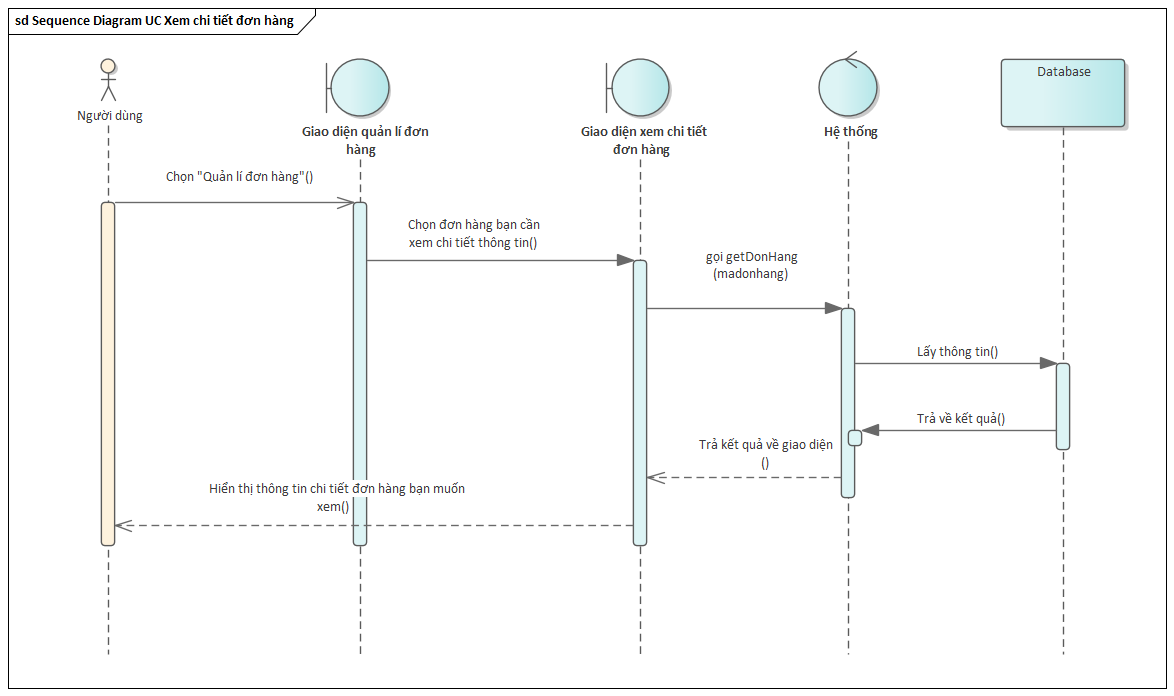
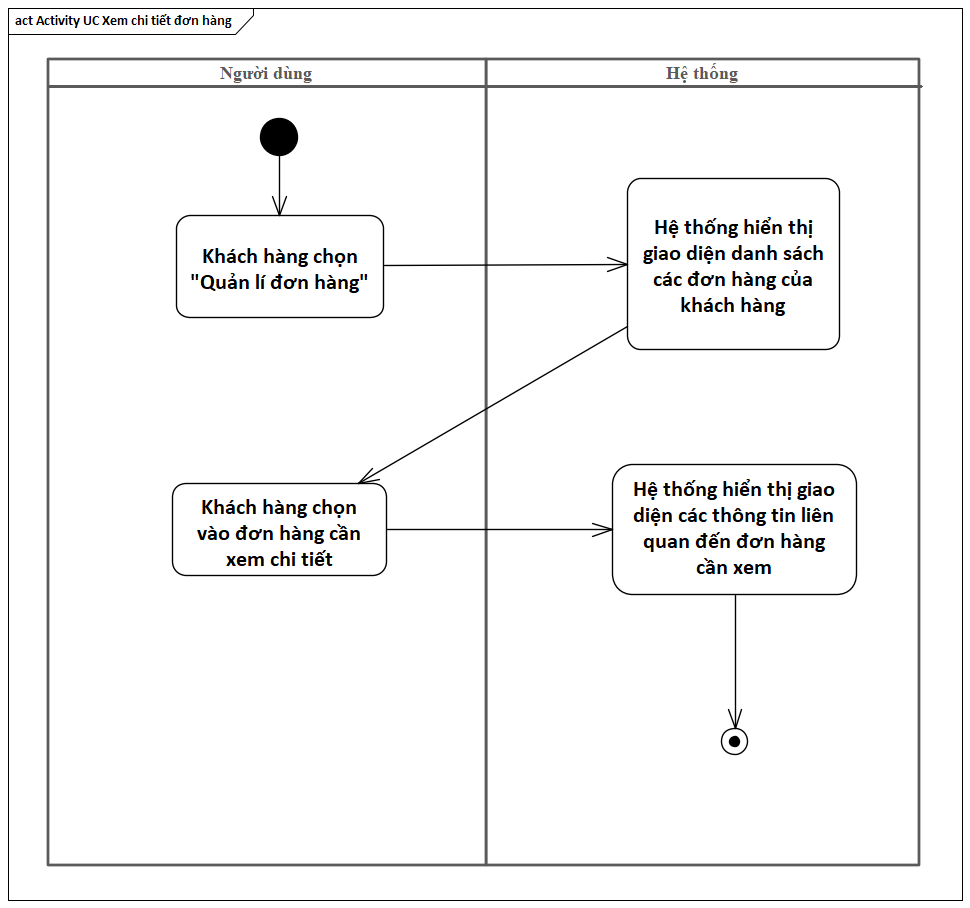
## **3.19.Xem chi tiết đơn hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Admin,Customer,Employee | | |
| Mô tả | Cho phép người dùng xem chi tiết đơn hàng | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống | | |
|  |  | Actor | Hệ thống |
| Luồng sự kiện chính | 1 | Chọn chức năng danh sách các đơn hàng |  |
| 2 |  | Hiển thị form danh sách đơn hàng |
| 3 | Chọn đơn hàng cần xem chi tiết |  |
| 4 |  | Hiển thị form thông tin chi tiết của đơn hàng đã chọn |

## **3.20.Xem danh sách các đơn hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| Tác nhân | Admin | | |
| Mô tả | Cho phép người dùng xem danh sách các đơn hàng | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống | | |
| Luồng sự kiện chính |  | Tác nhân | Hệ thống |
|  | 1 | Chọn chức năng “Tổng hợp đơn hàng” |  |
| 2 |  | Hiển thị form danh sách đơn hàng |

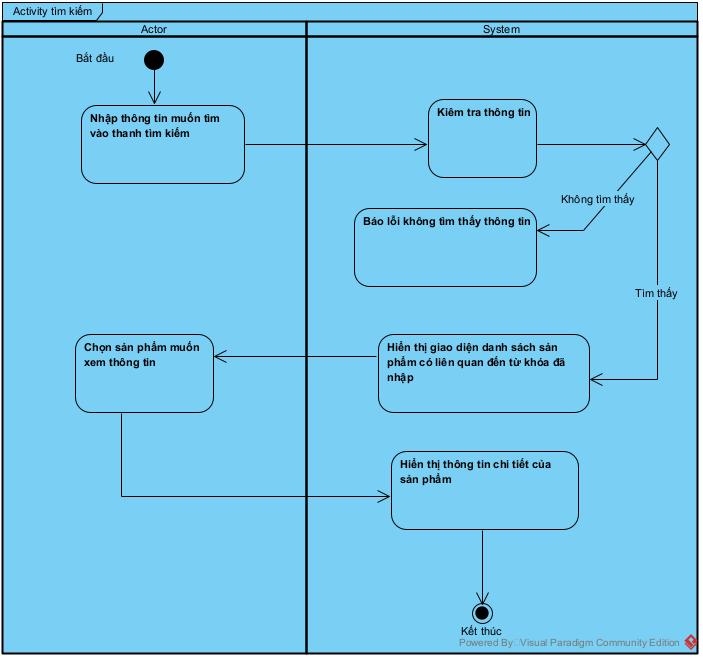


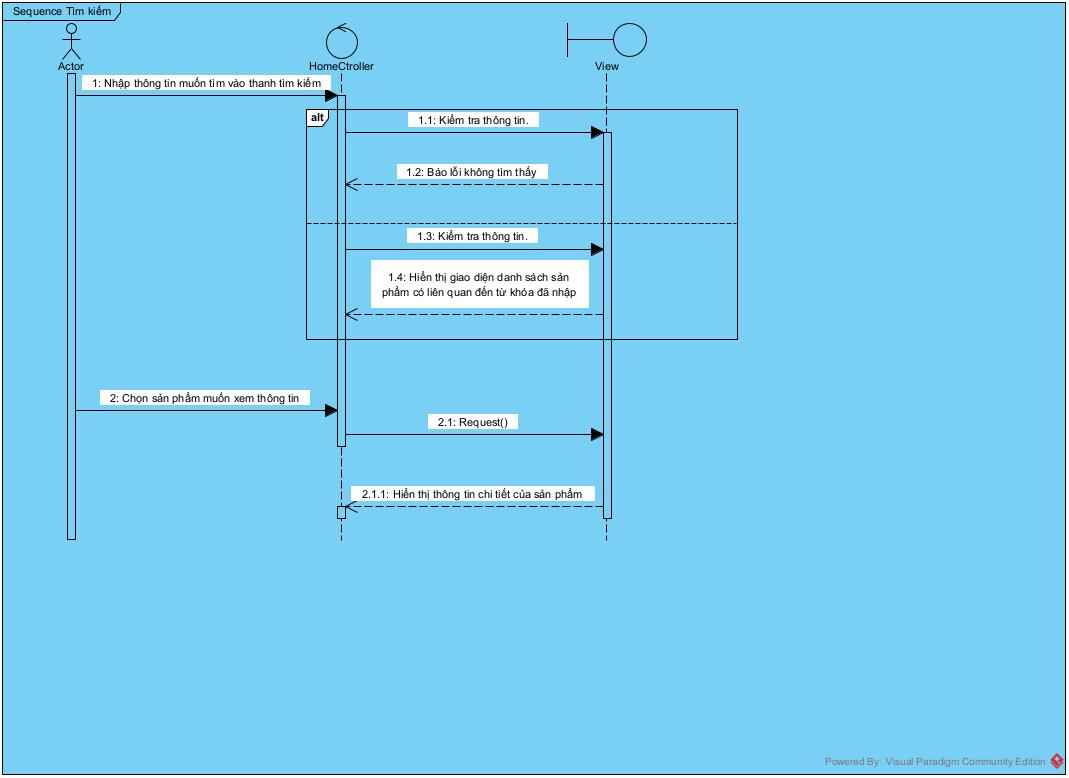


## **3.21.Use case Tìm kiếm sản phẩm**

3.21.1.Đặc tả

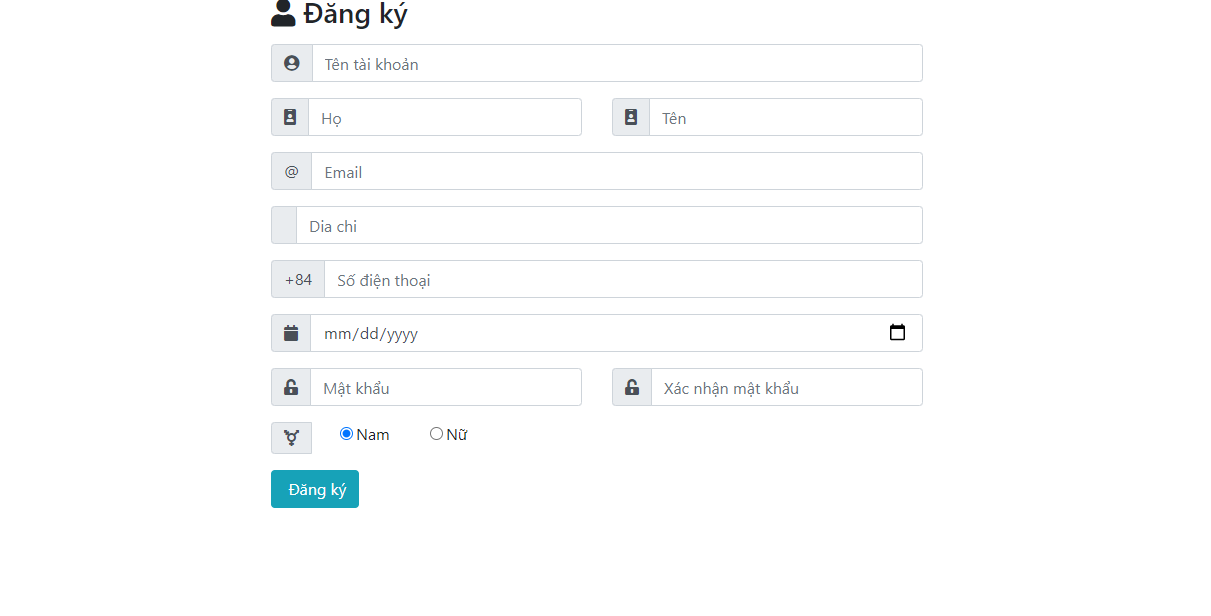
|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Customer,Employee,Admin | | |
| Mô tả | Cho phép người dùng tìm kiếm sản phẩm hoặc loại sản phẩm | | |
|  |  | Actor | System |
| Luồng sự kiện chính | 1 | Nhập thông tin muốn tìm vào thanh tìm kiếm |  |
| 2 |  | Kiểm tra thông tin. |
| 3 |  | Hiển thị giao diện danh sách sản phẩm có liên quan đến từ khóa đã nhập |
| 4 | Chọn sản phẩm muốn xem thông tin |  |
| 5 |  | Hiển thị thông tin chi tiết của sản phẩm |
| Luồng sự kiện phụ | 2.1 |  | Báo lỗi không tìm thấy |

3.21.2.Activity Tìm kiếm

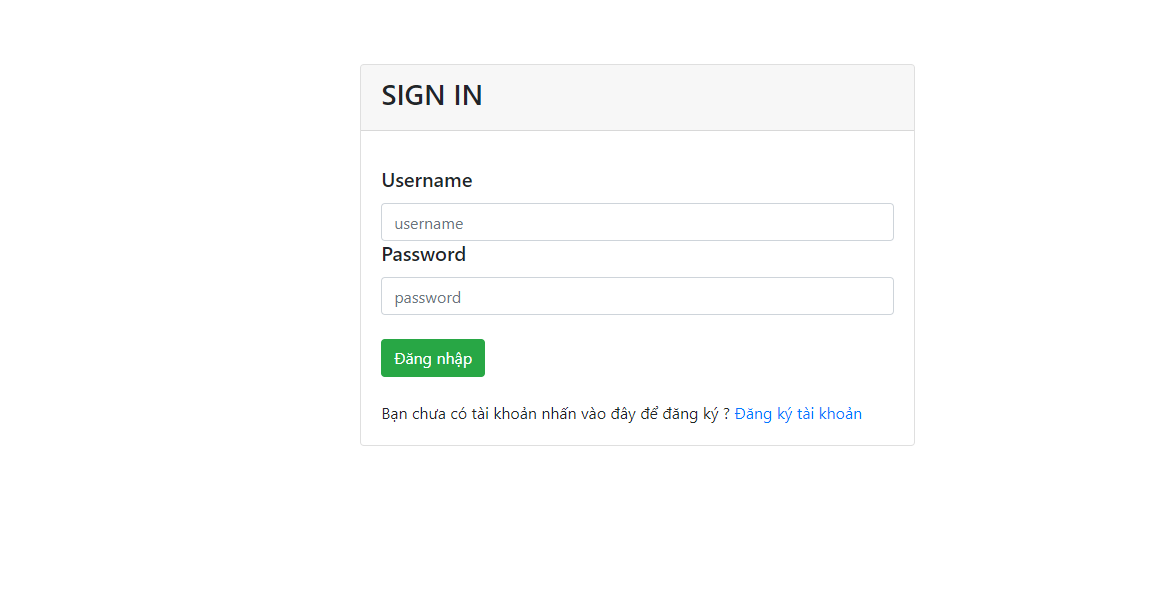
3.21.3.Sequence tìm kiếm

# **Chương 4:Thiết kế và Hiện thực**

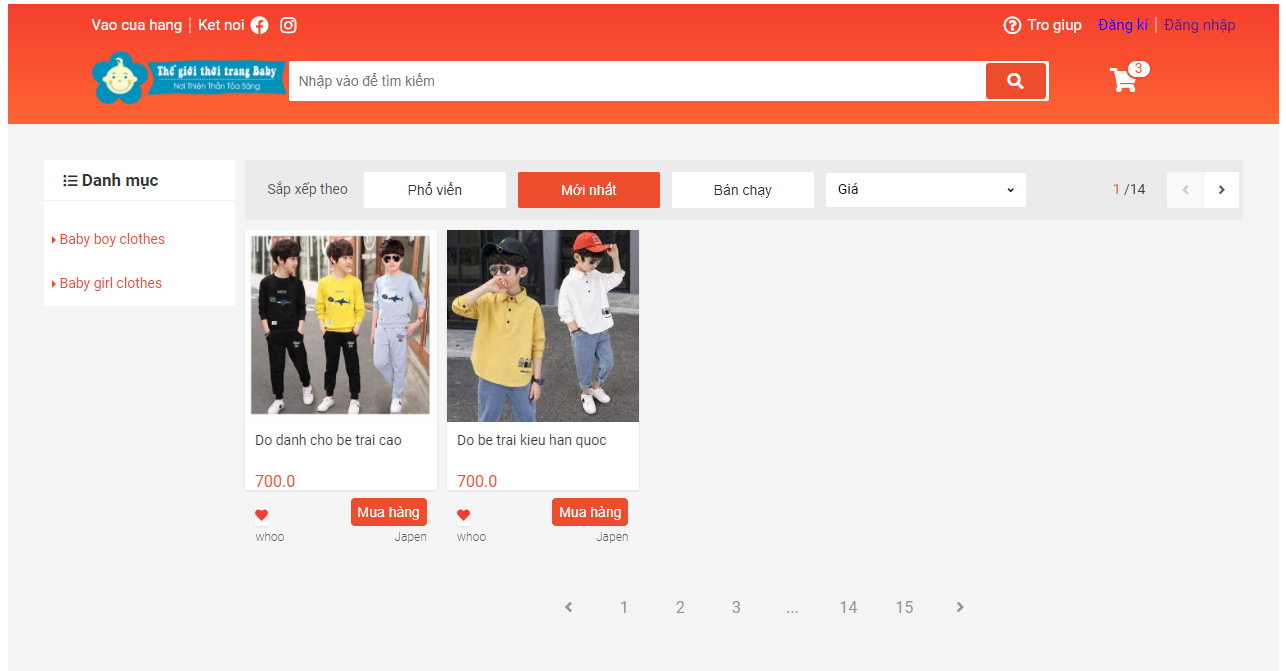
## **4.1.Giao diện đăng ký:**



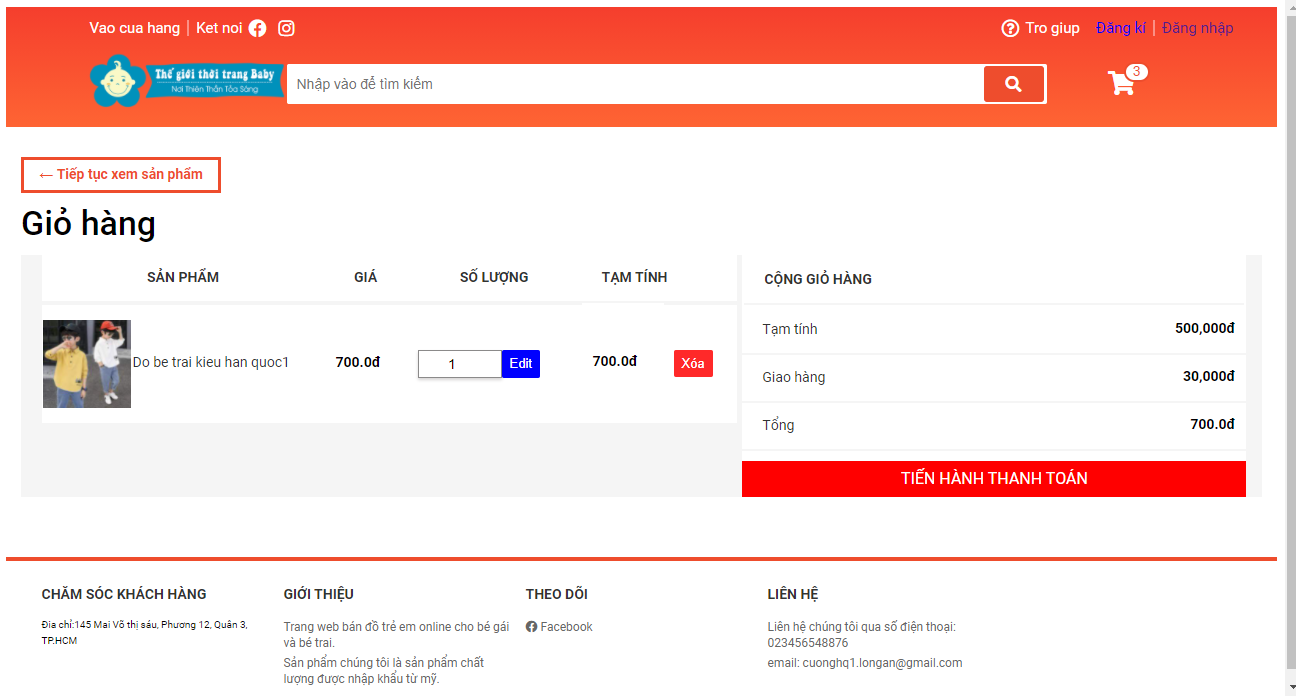
## **4.2.Giao diện đăng nhập:**



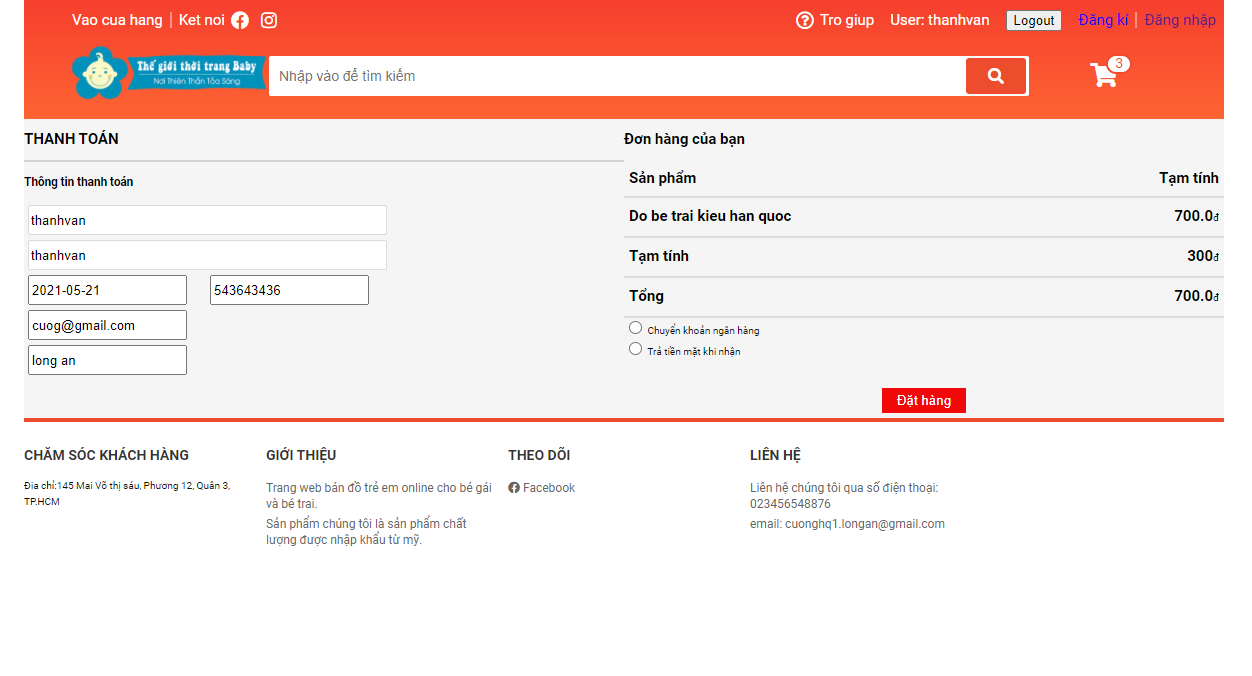
## **4.3.Giao diện trang chủ**



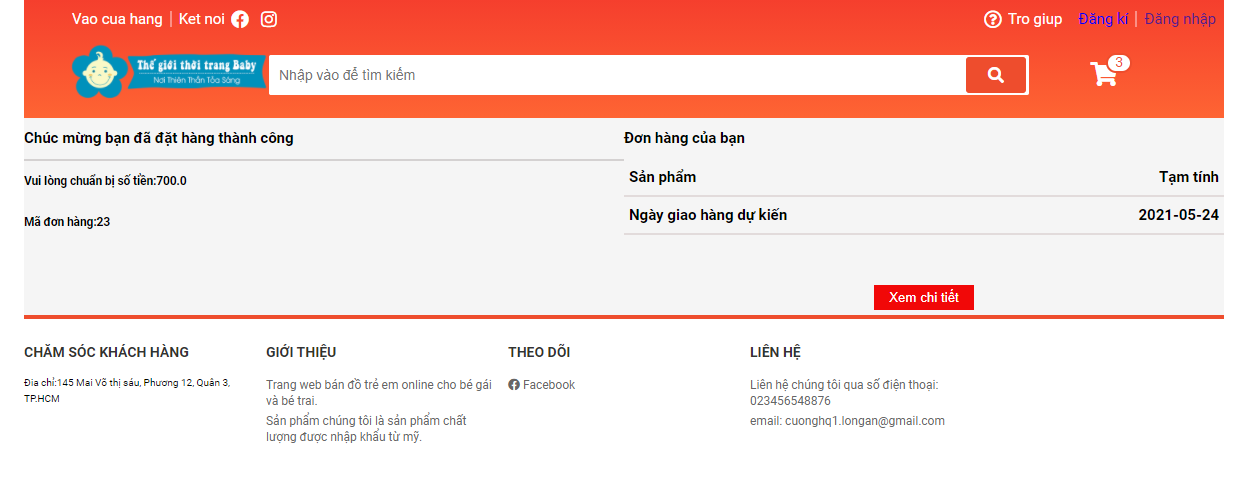
## **4.4.Giao diện giỏ hàng:**



## **4.5.Giao diện thanh toán:**

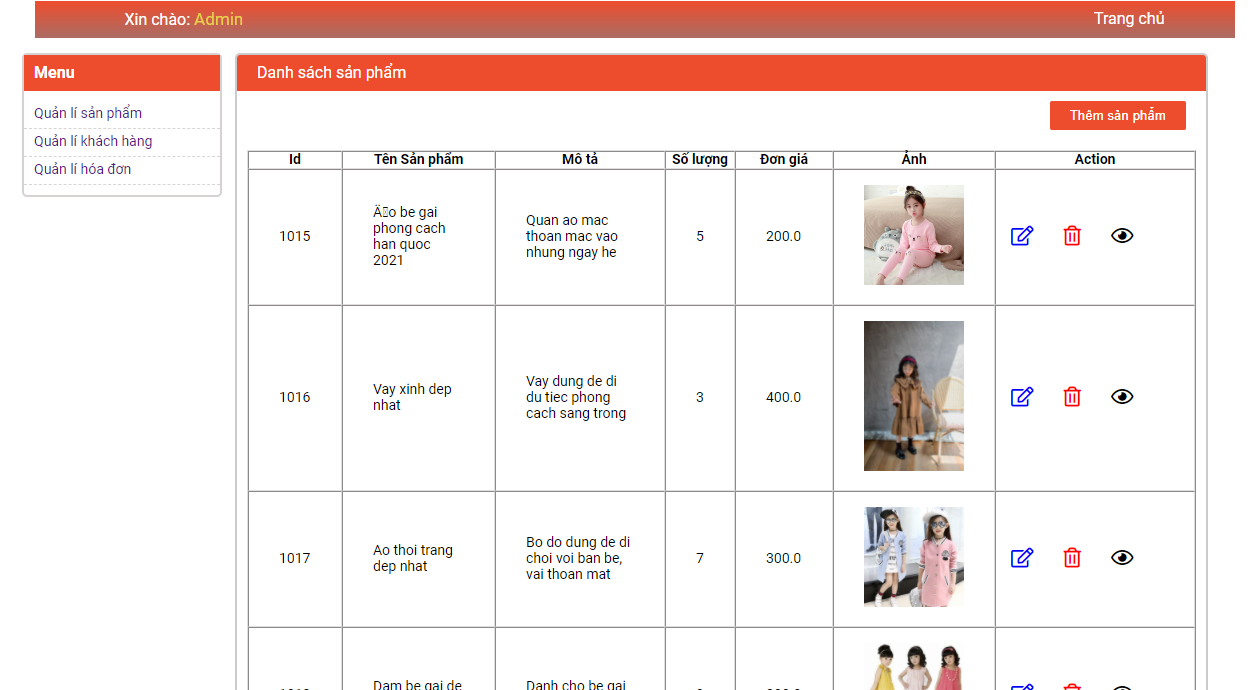


## **4.6.Giao diện hóa đơn:**

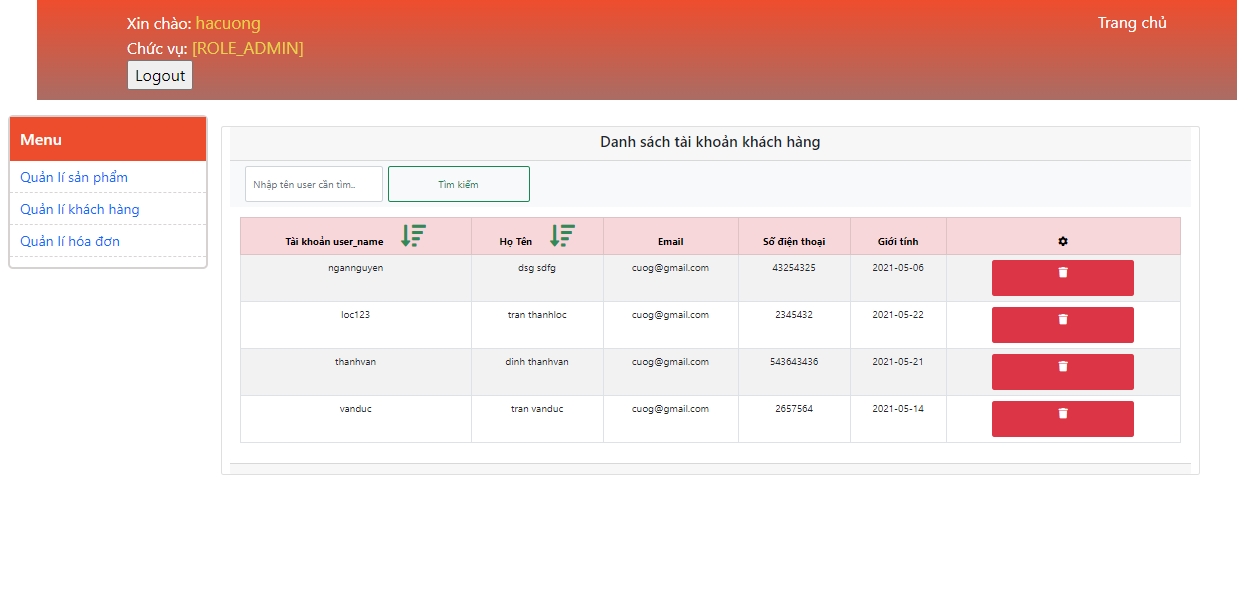


## **4.7. Giao diện Quản lý sản phẩm:**

## **4.8. Giao diện chi tiết sản phẩm**



## **4.9.Giao diện quản lý khách hàng:**



## **4.10.Giao diện Quản lý hóa đơn:**

# **Chương 5: KẾT LUẬN**

## **5.1.Kết quả đạt được**

* Sau thời gian thực hiện đề tài, website đã hoàn thành và đạt được một số kết quả sau:
* Tạo ra website để giới thiệu, bán quần áo thời trang trẻ em trực tuyến. Giúp các bậc phụ huynh dễ dàng lựa chọn và mua sắm sản phẩm thời trang cho con mình.
* Ứng dụng web hỗ trợ cho phụ huynh trong việc mua sắm và lựa chọn quần áo cho con cái thông qua website trực tuyến giúp tiết kiệm thời gian di chuyển và giảm thời gian làm thủ tục thanh toán.
* Giao diện của website thân thiện, dễ sử dụng

## **5.2.Hạn chế của đồ án**

* Thiếu kinh phí
* Một số chức năng còn thiếu sót như: gửi mail xác nhận, đánh giá sản phẩm, quên mật khẩu

## **5.3.Hướng phát triển**

* Đầu tư kinh phí
* Cố gắng bổ sung thêm những chức năng còn thiếu sót
* Khắc phục tối đa những lỗi phát sinh
* Phát triển trên app điện thoại
* Hoàn thiện website hơn giúp website trở nên dễ sử dụng hơn, hoạt động mượt mà ổn định hơn và công cụ tìm kiếm sản phẩm của website cho ra kết quả chính xác hơn để hỗ trợ người dung có thể dễ dàng, nhanh chóng có thể lựa chọn và mua được sản phẩm mình ưa thích.