**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------**

**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 748021**

**KHÓA 2019 - 2024**

***ĐỀ TÀI:***

**NGHIÊN CỨU PHÁT TRIỂN PHẦN MỀM THEO HƯỚNG CHIA NHỎ DỊCH VỤ (MICROSERVICE) ỨNG DỤNG XÂY DỰNG WEBSITE SHOPPING ONLINE**

|  |  |
| --- | --- |
| **SINH VIÊN THỰC HIỆN:** | **NGUYỄN THỊ KIM ANH** |
| **LỚP:** | **19CN3** |

**HÀ NỘI 11-2023**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------**

**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 748021**

**KHÓA 2019 - 2024**

***ĐỀ TÀI:***

**NGHIÊN CỨU PHÁT TRIỂN PHẦN MỀM THEO HƯỚNG CHIA NHỎ DỊCH VỤ (MICROSERVICE) ỨNG DỤNG XÂY DỰNG WEBSITE SHOPPING ONLINE**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **ThS.TRẦN ĐỨC THỊNH** |
| **SINH VIÊN THỰC HIỆN:** | **NGUYỄN THỊ KIM ANH** |
| **LỚP:** | **19CN3** |

**HÀ NỘI 11-2023**

MỤC LỤC

[CHƯƠNG I. CƠ SỞ LÝ THUYẾT 3](#_Toc152230637)

[1.1 Phần mở đầu: 3](#_Toc152230638)

[1.1.1. Lý do chọn đề tài 3](#_Toc152230639)

[1.1.2 Phạm vi và đối tượng của đề tài 3](#_Toc152230640)

[1.1.3. Khảo sát yêu cầu 4](#_Toc152230641)

[1.2. Cơ sở lý thuyết 5](#_Toc152230642)

[1.2.1. Tổng quan về ngôn ngữ Java 5](#_Toc152230643)

[1.2.2. Giới thiệu về MICROSERVICE 7](#_Toc152230644)

[1.2.3. Giới thiệu HTML, CSS, JavaScript, Bootstrap, Spring framework Angular Framework 10](#_Toc152230645)

[CHƯƠNG 2: THIẾT KẾ HỆ THỐNG 17](#_Toc152230646)

[2.1. Mô hình triển khai và sơ đồ hệ thống 17](#_Toc152230647)

[2.1.1. Mô hình triển khai 17](#_Toc152230648)

[2.1.2.Sơ đồ hệ thống 17](#_Toc152230649)

[2.2.Yêu cầu hệ thống 19](#_Toc152230650)

[2.2.1: Sơ đồ Usecase 19](#_Toc152230651)

[2.2.2: Mô tả Usecase: 19](#_Toc152230652)

[2.2b.1. UC1 Đăng nhập 23](#_Toc152230653)

[2.2b.2. UC2.1 Đăng xuất 23](#_Toc152230654)

[2.2b.3. UC02.2 Quên mật khẩu 24](#_Toc152230655)

[2.2b.4. UC2.3 Xem thông tin cá nhân 24](#_Toc152230656)

[2.2b.5. UC2.4 Cập nhật thông tin cá nhân 25](#_Toc152230657)

[2.2b.6. UC2.5.1 Xem danh sách đơn hàng đã đặt 25](#_Toc152230658)

[2.2b.7. UC2.5.2 Hủy đơn hàng 26](#_Toc152230659)

[2.2b.8. UC2.5.3 Đánh giá và bình luận 26](#_Toc152230660)

[2.2b.9. UC2.5.4 Xem chi tiết đơn hàng 27](#_Toc152230661)

[2.2b.10. UC3.1 Thêm sản phẩm vào yêu thích 27](#_Toc152230662)

[2.2b.11. UC3.2 Xem chi tiết sản phẩm 27](#_Toc152230663)

[2.2b.12. UC3.3 Xem danh sách sản phẩm 28](#_Toc152230664)

[2.2b.13. UC3.3.1 Liệt kê sản phẩm theo loại 28](#_Toc152230665)

[2.2b.14. UCUC3.3.2 Liệt kê sản phẩm theo đánh giá 29](#_Toc152230666)

[2.2b.15. UC3.3.3 Liệt kê sản phẩm theo ký tự 29](#_Toc152230667)

[2.2b.16. UC3.4 Giỏ hàng 30](#_Toc152230668)

[2.2b.17. UC3.4.1 Thêm sản phẩm vào giỏ hàng 30](#_Toc152230669)

[2.2b.18. UC3.4.2 Cập nhật số lượng sản phẩm 31](#_Toc152230670)

[2.2b.19. UC3.4.3 Xóa sản phẩm ra khỏi giỏ hàng 31](#_Toc152230671)

[2.2b.20. UC3.4.4 Đặt hàng 31](#_Toc152230672)

[2.2b.21. UC3.4.4.1 Xác nhận đơn hàng 32](#_Toc152230673)

[2.2b.22. UC3.4.4.2 Xem chi tiết đơn hàng vừa đặt 32](#_Toc152230674)

[2.2b.23. UC4.1 Xem danh sách 33](#_Toc152230675)

[2.2b.24. UC4.2 Tìm kiếm 33](#_Toc152230676)

[2.2b.25. UC4.3 Thêm mới 34](#_Toc152230677)

[2.2b.26. UC4.4 Cập nhật 34](#_Toc152230678)

[2.2b.27. UC4.5 Xem thông tin 35](#_Toc152230679)

[2.2b.28. UC4.6 Xóa 35](#_Toc152230680)

[2.2b.29. UC5 Thống kê 36](#_Toc152230681)

[2.2b.30. UC5.1 Thống kê doanh thu 36](#_Toc152230682)

[2.2b.31. UC5.2 Xem thống kê kho hàng 37](#_Toc152230683)

[2.2b.32. UC5.3 Thống kê khách hàng 37](#_Toc152230684)

[2.2b.33. UC5.4 Thống kê đánh giá 38](#_Toc152230685)

[2.2b.34. UC5.5 Thống kê đơn hàng 38](#_Toc152230686)

[2.2b.35. UC5.5.1 Quản lý đơn hàng 38](#_Toc152230687)

[TÀI LIỆU THAM KHẢO 40](#_Toc152230688)

**LỜI NÓI ĐẦU**

Đề tài “Nghiên cứu phát triển phần mềm theo hướng chia nhỏ dịch vụ (Microservice) ứng dụng xây dựng website Shopping online” là nội dung em chọn để nghiên cứu và làm đồ án tốt nghiệp sau 5 năm theo học chương trình đại học chuyên ngành công nghệ thông tin tại trường Trường Đại Học Kiến Trúc Hà Nội.

Để hoàn thành quá trình nghiên cứu và hoàn thiện đồ án này, em xin gửi lời cảm ơn chân thành và sâu sắc đến các quý thầy cô trong khoa Công nghệ thông tin, cũng như tất cả các thầy cô Trường Đại Học Kiến Trúc Hà Nội đã truyền đạt kiến thức, kinh nghiệm quý báu của mình cho em trong suốt quá trình học tập và nghiên cứu tại trường.

Đặc biệt em xin gửi lời cảm ơn đến thầy ThS. Trần Đức Thịnh thuộc khoa Công nghệ thông tin – Đại học Kinh doanh và Công nghệ Hà Nội. Thầy đã trực tiếp chỉ bảo và hướng dẫn em trong suốt quá trình nghiên cứu để em có thể hoàn thiện đồ án này.Mặc dù đã nỗ lực hoàn thành đồ án này nhưng không thể tránh khỏi những sai sót. Rất mong nhận được sự góp ý của quý Thầy cô và các bạn.

# CHƯƠNG I. CƠ SỞ LÝ THUYẾT

## Phần mở đầu:

### 1.1.1. Lý do chọn đề tài

Hiện nay, với sự phát triển vượt bậc của công nghệ và để bắt kịp xu hướng của công nghệ 4.0 thì tất cả các mặt hàng kinh doanh đều phải được buôn bán theo hình thức trực tiếp và qua hình thức mua bán online để phục vụ nhu cầu khách hàng tốt hơn và mang lại doanh thu hiệu quả hơn.

Hướng tới yêu cầu đó em lựa chọn đề tài xây dựng website quản lý bán máy tính để giúp chon nhu cầu mua sắm của khách hàng được thuận lợi hơn, Những khách hàng không có thời gian tới trực tiếp cửa hàng xem sản phẩm thì có thể tranh thủ mọi lúc mọi nơi truy cập vào trang web của cửa hàng để có thể xem toàn bộ thông tin của sản phẩm và tiến hành đặt hàng online thông qua website.

### 1.1.2 Phạm vi và đối tượng của đề tài

- Phạm vi của đề tài: Là một Website với giao diện thân thiện với người tiêu dùng, dễ dàng thực hiện các thao tác, quản lý hàng hóa, thống kê, đơn hàng cho người sử dụng. Website cung cấp đủ các chức năng xem, mua hàng, đặt hàng từ xa, bình luận, đánh giá sản phẩm nông sản cho người tiêu dùng.

- Đối tượng mà đề tài hướng đến là tất cả các cá nhân, cửa hàng nhỏ, hay chi nhánh cửa hàng máy tính của các tập đoàn kinh doanh mặt hàng điện tử công nghệ lớn.

### 1.1.3. Khảo sát yêu cầu

#### - Về mặt thiết bị và phần mềm

- Hệ điều hành hỗ trợ cho chương trình là Windows 10, Windows 11.

- Hệ cơ sở dữ liệu được dùng là MySQL, SQL SERVER.

- Các phần mềm khác dùng để lập trình và thiết kế phần mềm như nodeJS, Eclipse, Spring tool, Visual studio code, Star UML, Xamp, SQL Workbench….

#### - Yêu cầu về phần mềm

Hệ thống gồm có 2 phần:

* **Phần dành cho người quản lý (chủ cửa hàng):**

Chủ cửa hàng là người tương tác với khách hàng, những người mà có nhu cầu tìm kiếm, xem, mua hay đặt hàng nông sản từ cửa hàng…. Và cũng là người có quyền thực hiện các thao tác quản lý trên Website về sản phẩm, đơn hàng, doanh thu,…. Người này được cấp một username và password để đăng nhập vào hệ thống thực hiện những chức năng của mình bao gồm:

Thứ nhất: Xem, thêm, sửa, xóa thông tin liên quan đến sản phẩm, việc này giúp quản lý sản phẩm một cách thuận lợi.

Thứ hai: Xem doanh thu, kho hàng việc này giúp cho chủ cửa hàng có thể biết tình trạng cửa hàng trong thời gian xem xét.

Thứ ba: Xem và tương tác với các đơn hàng như hủy đơn, xác nhận đơn hàng,… để thực hiện giao dịch từ xa với khách hàng.

Thứ tư: Xem danh sách khách hàng, khách hàng tiềm năng của cửa hàng.

Ngoài các chức năng nêu trên thì phần mềm phải được trình bày sao cho dễ hiểu, giao diện mang tính dễ dùng và làm sao cho chủ cửa hàng dễ dàng thực hiện được những mong muốn của khách hàng một cách nhanh nhất. Nhằm đưa chất lượng phục vụ của cửa hàng lên cao, tạo sự tín nhiệm của khách hàng.

* **Phần dành cho khách hàng:**

Khách hàng là người xem và mua sản phẩm được trình bày trên Website, và để thuận tiện cho việc phân biệt giữa các khách hàng cũng như giỏ hàng riêng của từng người khi thực hiện mua hàng hóa tại cửa hàng. Website phải có các chức năng sau:

- Thứ nhất: Đăng ký tài khoản khách hàng, để chủ cửa hàng nhận biết các khách hàng của mình cũng như đơn hàng của họ, và cho khách hàng có thể đặt hàng từ xa.

- Thứ hai: Xem thông tin, bình luận, đánh giá của sản phẩm nhằm để khách hàng hiểu rõ về sản phẩm mình muốn mua, cũng như nhận xét của khách hàng khác.

- Thứ ba: Thêm sản phẩm vào giỏ hàng, cũng như đặt hàng để giúp khách hàng có thể mua sản phẩm từ xa mà không cần tới tại cửa hàng.

- Thứ tư: Chức năng bình luận, đánh giá để bày tỏ thái độ, nhận xét của khách hàng về sản phẩm của cửa hàng giúp cho chủ cửa hàng nâng cao chất lượng sản phẩm cũng như tiếp cận nhu cầu của người mua hàng.

Ngoài các chức năng trên, thì trang dành cho khách hàng phải có thêm các chức năng như đăng nhập, chỉnh sửa thông tin cá nhân, tìm kiếm, lọc sản phẩm,… Giao diện cho khách hàng phải thân thiện, dễ sử dụng, bắt mắt để gắt kết với cửa hàng.

## 1.2. Cơ sở lý thuyết

### 1.2.1. Tổng quan về ngôn ngữ Java

- Ngôn ngữ Java là một ngôn ngữ lập trình phổ biến được phát triển vào cuối những năm 1990 bởi Sun Microsystems (nay là một phần của Oracle Corporation). Đây là một ngôn ngữ lập trình có cú pháp rõ ràng, dễ đọc và dễ hiểu, được sử dụng rộng rãi trong việc phát triển ứng dụng từ ứng dụng di động cho đến ứng dụng máy tính để bàn và các hệ thống phức tạp.



*Hình 1.1 Ngôn ngữ Java*

- Các đặc trưng của Java

+ Tài nguyên học tập chất lượng cao

Java đã ra mắt từ lâu nên các nhà phát triển mới có thể tiếp cận rất nhiều tài nguyên học tập. Tài liệu chi tiết, những cuốn sách toàn diện và các khóa học sẽ hỗ trợ nhà phát triển trong quá trình học hỏi. Ngoài ra, người mới làm quen có thể bắt đầu viết mã bằng Java Cơ bản trước khi chuyển sang Java Nâng cao.

+ Các chức năng và thư viện sẵn có

Khi sử dụng Java, nhà phát triển không cần phải viết mọi chức năng mới từ đầu. Thay vào đó, Java cung cấp một hệ sinh thái phong phú gồm các chức năng và thư viện sẵn có để phát triển hàng loạt ứng dụng đa dạng.

+ Sự hỗ trợ tích cực của cộng đồng

Java có rất nhiều người dùng hoạt động và một cộng đồng có thể hỗ trợ nhà phát triển khi họ đối mặt với các thách thức trong việc viết mã. Phần mềm nền tảng Java cũng được duy trì và cập nhật thường xuyên.

+ Công cụ phát triển chất lượng cao

Java cung cấp nhiều công cụ khác nhau để hỗ trợ chỉnh sửa tự động, gỡ lỗi, thử nghiệm, triển khai và quản lý thay đổi. Những công cụ này khiến việc lập trình bằng Java tiết kiệm thời gian và chi phí.

+ Độc lập với nền tảng

Mã Java có thể chạy trên bất kỳ nền tảng cơ sở nào như Windows, Linux, iOS hoặc Android mà không cần viết lại. Đây là điều khiến ngôn ngữ này trở nên đặc biệt mạnh mẽ trong môi trường hiện nay khi chúng ta muốn chạy ứng dụng trên nhiều thiết bị.

+ Bảo mật

Người dùng có thể tải mã Java không tin cậy từ trên mạng xuống và chạy mã này trong môi trường bảo mật để nó không thể gây hại. Mã không tin cậy sẽ không thể lây nhiễm vi-rút cho hệ thống máy chủ và cũng không thể đọc hoặc ghi tệp từ ổ cứng. Bạn cũng có thể cấu hình nhiều cấp độ và hạn chế bảo mật trong Java.

### 1.2.2. Giới thiệu về MICROSERVICE

Microservices là một phương pháp đặc biệt được sử dụng trong quá trình phát triển phần mềm. Phương pháp này tập trung chủ yếu vào việc xây dựng những module đơn chức năng với những giao diện và cách thức hoạt động được xác định rõ ràng.

Phương pháp Microservices hỗ trợ nhiều lợi ích cho những nhóm Agile hay DevOps. Những công ty công nghệ hàng đầu thế giới hiện nay như eBay, Amazon, Paypal, Twitter,… đều đã tiến hành phát triển chuyển từ những kiến trúc nguyên khối sang dạng Microservices.



*Hình 1.2 Microservices*

Khái niệm này giải quyết những vấn đề hạn chế của hệ thống nguyên khối thông qua việc sử dụng những module hóa càng nhiều càng tốt. Microservices ở dạng đơn nhất có thể giúp xây dựng ứng dụng dưới dạng bộ các dịch vụ đơn lẻ nhỏ, mỗi dịch vụ này chạy với quy trình riêng và có khả năng triển khai độc lập. Những dịch vụ này có thể được triển khai bằng các ngôn ngữ lập trình khác nhau. Chúng thường được liên kết thông qua API và có thể tận dụng nhiều giải pháp tương tự như đã phát triển trong web và restfulAPI.

Microservices được sử dụng ngày càng phổ biến tại nhiều doanh nghiệp chuyên về phần mềm nhờ những lợi ích mà phương pháp này mang lại như:

- Mã nguồn tinh gọn:

Chính vì hệ thống được cấu hình từ những dự án nhỏ và từng dự án này đều rất cơ bản và chỉ tập trung vào một hay một vài những nghiệp vụ chính. Từ đó, code base và mức độ phức tạp của chúng đều không cao. Nhờ vậy mà nó hỗ trợ những tính năng hoạt động mượt mà và dễ dàng hơn trong công tác bảo trì hay mở rộng.

- Bảo mật mã nguồn được tối ưu hóa

Việc những nhân viên ở từng dự án chỉ được phép truy cập vào mã nguồn của dự án đang phụ trách sẽ đảm bảo cho khả năng kiểm soát thông tin, dữ liệu dự án tổng tốt hơn. Từ đó, nếu có vấn đề bảo mật phát sinh, phạm vi truy vết ảnh hưởng sẽ được thu hẹp.

- Tồn tại độc lập

Với khả năng tồn tại và hoạt động một cách độc lập, giúp cho nguy cơ ảnh hưởng lẫn nhau được giảm đến mức tối đa. Tức là, trong toàn hệ thống dịch vụ gồm nhiều dự án riêng, nếu một trong những dự án đó bị ảnh hưởng thì những dự án khác của dịch vụ vẫn sẽ hoạt động một cách bình thường.

- Khả năng mở rộng độc lập

Tùy vào nhu cầu sử dụng, người phụ trách có thể tiến hành mở rộng quy mô của riêng đối với dịch vụ trong hệ thống. Tức là, mỗi dịch vụ trong hệ thống sẽ có khả năng mở rộng và phát triển riêng mà vẫn đảm bảo không ảnh hưởng đến những dịch vụ khác.

- Kiến trúc của Microservices

Không có một định nghĩa chính thức về kiến trúc chung của Microservices hay Microservices architecture, cũng như không có những trình bày tiêu chuẩn nào về khái niệm này. Tuy nhiên, hầu hết những Microservices đều có những đặc điểm chung như:

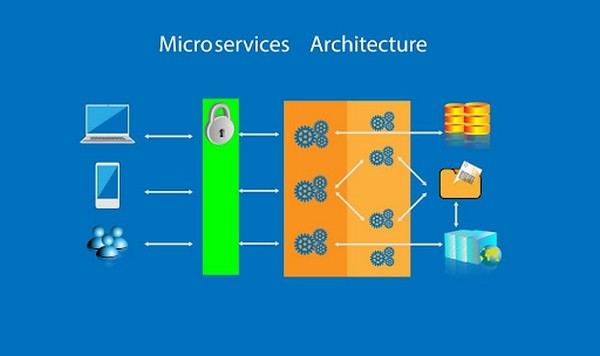
+ Phần mềm phát triển dựa trên phương pháp Microservices có thể được chia thành nhiều các thành phần dịch vụ khác nhau.

+ Phương thức hoạt động của Microservices tương tự như hệ thống UNIX như nhận yêu cầu, xử lý yêu cầu và đưa ra phản hồi.

+ Microservices có phương pháp quản trị phi tập trung nhằm tạo ra những công cụ hỗ trợ giải quyết những vấn đề khó khăn.

+ Hướng đến sự toàn diện nên Microservices được xây dựng nhằm tránh rủi ro từ thất bại

+ Cấu trúc của Microservices là thiết kế tiến hóa, thường được các doanh nghiệp ưu tiên sử dụng.



*Hình1.3 Cấu trúc Microservices*

### 1.2.3. Giới thiệu HTML, CSS, JavaScript, Bootstrap, Spring framework Angular Framework

**- HTML** (HyperText Markup Language): Ngôn ngữ đánh dấu được sử dụng để xây dựng cấu trúc và nội dung của trang web. HTML định nghĩa các phần tử và cấu trúc của trang, bao gồm văn bản, hình ảnh, đường liên kết, biểu mẫu, …

**- CSS** (Cascading Style Sheets): Ngôn ngữ định dạng được sử dụng để tùy chỉnh và thiết kế giao diện của trang web. CSS định nghĩa cách mà các phần tử HTML được hiển thị trên trình duyệt, bao gồm màu sắc, font chữ, khoảng cách, vị trí,…

**- JavaScript** (JS): Ngôn ngữ lập trình phổ biến được sử dụng chủ yếu cho việc tạo ra các tương tác động và thay đổi trên trang web. JavaScript có thể thực hiện các hoạt động đa dạng như thêm/sửa/xóa nội dung, xử lý sự kiện, thay đổi giao diện người dùng, …

**- Bootstrap** là một framework CSS mã nguồn mở, cung cấp các công cụ và giao diện sẵn có giúp tạo ra các trang web thân thiện với di động, linh hoạt và đẹp mắt. Bootstrap cung cấp các lớp CSS, các thành phần giao diện người dùng và các bố cục được chuẩn hóa giúp phát triển web nhanh chóng.

**- Spring framework** là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)

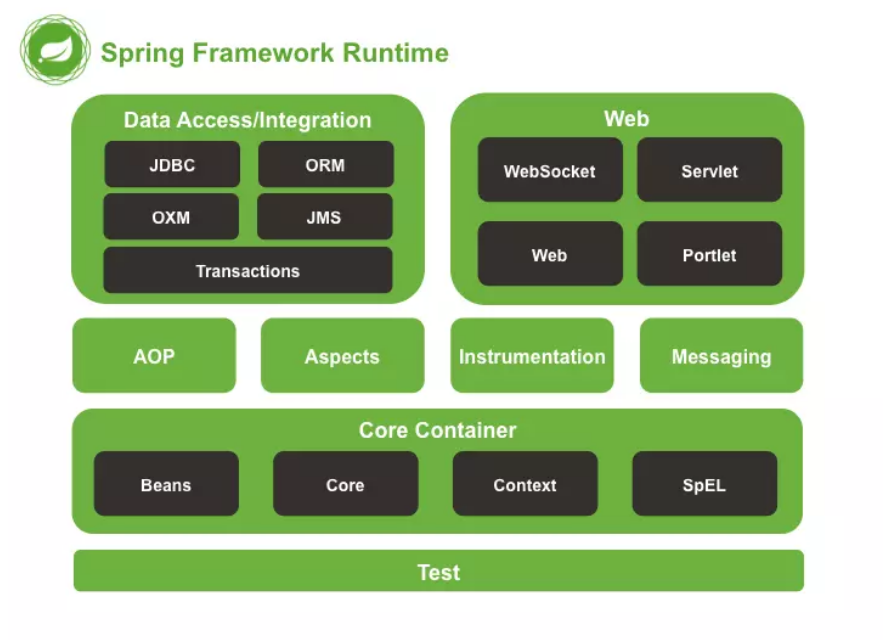
Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lơn.

Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.

Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

* Kiến trúc, các module của Spring Framework

Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng 1 trong các module đó.

Dưới đây là kiến trúc tổng thể của Spring Framework.

*Hình 1.4 Kiến trúc tổng thể của Spring Framework*

Test

Tầng này cung cấp khả năng hỗ trợ kiểm thử với JUnit và TestNG.

Spring Core Container

Bao gồm các module spring core, beans, context và expression languate (EL)

Spring core, bean cung cấp tính năng IOC và Dependency Injection.

Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB, JMX.

Expression Language được mở rộng từ Expresion Language trong JSP. Nó cung cấp hỗ trợ việc setting/getting giá trị, các method cải tiến cho phép truy cập collections, index, các toán tử logic…

AOP, Aspects and Instrumentation

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

Data Access / Integration

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database Web

Hay còn gọi là Spring MVC

Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

* Các lợi ích của Spring Framework.

Spring cho phép lập trình viên sử dụng POJOs. Việc sử dụng POJOs giúp ta không phải làm việc với EJB, ứng dụng, các luồng chạy, cấu hình… đơn giản hơn rất nhiều.

Spring được tổ chức theo kiểu mô đun. Số lượng các gói và các lớp khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì cần và không cần quan tâm đến phần còn lại.

Spring hỗ trợ sử dụng khá nhiều công nghệ như ORM Framework, các logging framework, JEE, các thư viện tạo lịch trình (Quartz và JDK timer)…

Module Web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ các tính năng giúp thay thế các web framework khác như Struts.

* Một số dự án khác của Spring.

Cũng dựa trên các nguyên tắc thiết kế cơ bản của spring core. Spring còn phát triển nhiều project con như:

+ Spring MVC

Spring MVC được thiết kế dành cho việc xây dựng các ứng dụng nền tảng web.

+ Spring Security

Cung cấp các cơ chế xác thực (authentication) và phân quyền (authorization) cho ứng dụng .

+ Spring Boot

Spring Boot là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng.

+ Spring Batch

Dự án này giúp chúng ta dễ dàng tạo các lịch trình (scheduling) và tiến trình (processing) cho các công việc xử lý theo mẻ (batch job).

+ Spring Social

Dự án này sẽ kết nối ứng dụng của bạn với các API bên thứ ba của Facebook, Twitter, Linkedin … (ví dụ đăng nhập bằng facebook, google+ …)

+ Spring IO

+ Spring Cloud

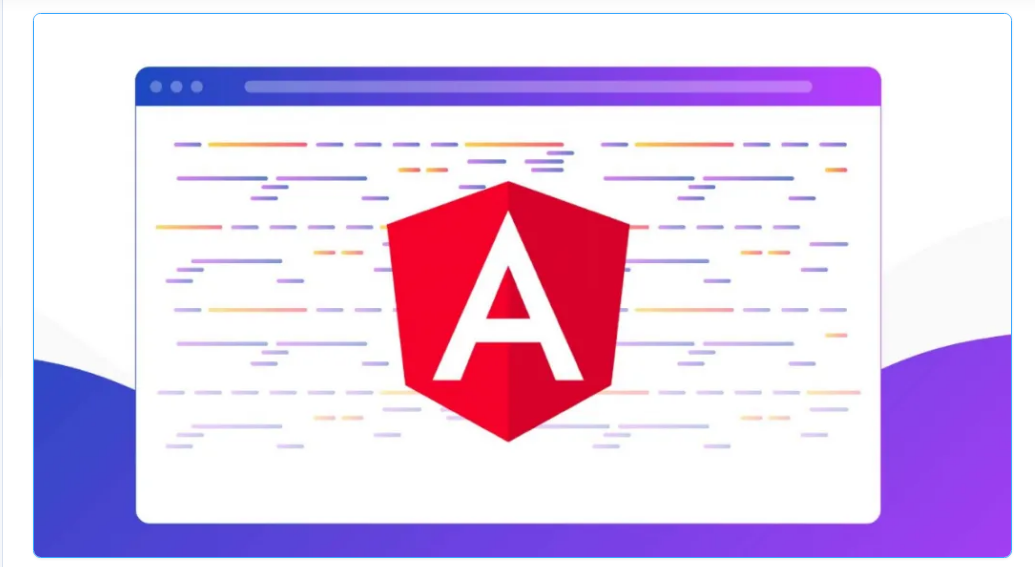
+ Spring Mobile

+ Spring for Android

+ Spring Session

**- Angular Framework**

Angular là một mã nguồn mở viết bằng TypeScript và được sử dụng để thiết kế giao diện web (front – end). Angular được xây dựng, phát triển từ những năm 2009 và đang duy trì cho đến nay bởi Google. Đây được xem là framework front end mạnh mẽ và chuyên dụng dành cho các lập trình viên sử dụng HTML cao cấp.



*Hình 1.5 Angular Framework*

Angular được ứng dụng rộng rãi để xây dựng các project Single Page Application (ứng dụng trang đơn). Hiện tại, version stable của Angular là Angular 9 (released on February 7, 2020) với TypeScript 3.6 và 3.7.

Hiện nay, Angular được các công ty lớn lựa chọn sử dụng như: Upwork, Forbes, General Motors,… Đây sẽ là cơ hội việc làm rất lớn nếu bạn sử dụng thành thạo Angular. Tuy nhiên, trước đó bạn cần nắm vững các kiến thức nền tảng về JavaScript, CSS và HTML, cách làm việc với kiến trúc Model-View-Controller (MVC).

Lịch sử phát triển của Angular:

Phần mềm này được xây dựng, phát triển và cho ra đời vào năm 2009 bởi Misko Hevery cùng với một người bạn khác là Adam Abrons. Angular được coi là một dự án riêng cho đến khi Misko Hevery tham gia vào dự án Google Feedback với tư cách là lập trình viên bán thời gian. Trong thời gian khoảng 6 tháng, Misko cùng 2 người khác nữa đã viết lên 17.000 dòng mã khác nhau cho dự án Google Feedback.

Ưu và nhược điểm của Angular

Mỗi mã nguồn đều có những ưu – nhược điểm nhất định và Angular cũng vậy. Cụ thể sau đây là những ưu – nhược điểm của mã nguồn này mà bạn cần nắm.

**Ưu điểm:**

+ Angular được các chuyên gia đánh giá cao, mã nguồn này giúp các Single Page Application làm việc dễ dàng, nhanh chóng.

+ Nhờ khả năng Binding data lên trên các nền tảng HTML nên code front-end thường rất thân thiện với người dùng.

+ Có thể thuận tiện Unit Test.

+ Component có thể tái sử dụng dễ dàng hơn.

+ Angular có khả năng hỗ trợ cho các lập trình viên có thể viết code được ít hơn cùng với nhiều chức năng hơn. Từ đó giúp tiết kiệm thời gian lập trình và tăng hiệu suất công việc.

+ AngularJS tương thích với nhiều nền tảng khác nhau. Bạn có thể dùng được trên nhiều loại trình duyệt khác nhau cả trong máy tính và thiết bị điện thoại di động. **Nhược điểm:**

+ Tính bảo mật: Bản chất của Angular là một framework front-end. Thông thường, tính bảo mật của front-end thường không cao bằng back-end. Chính vì thế, bạn cần xây dựng một hệ thống kiểm tra dữ liệu sao cho việc trả về được tốt nhất khi sử dụng API.

+ Khả năng an toàn: Website có thể trở nên không an toàn nếu bạn sử dụng một số trình duyệt sở hữu tính năng Disable JavaScript.iều nền tảng khác nhau. Bạn có thể dùng được trên nhiều loại trình duyệt khác nhau cả trong máy tính và thiết bị điện thoại di động.

**Angular có các đặc trưng nổi bật sau đây:**

+ Angular có khả năng tạo ra các ứng dụng client-side dựa trên mô hình Model-View-Controller (MVC).

+ Angular được các lập trình viên sử dụng để có thể phát triển dựa trên JavaScript.

+ Các mã JavaScript có thể dễ dàng tự động xử lý sao cho phù hợp với các trình duyệt nhất nhờ khả năng tương thích cao của Angular.

+ Angular được sử dụng rộng rãi khi có mã nguồn mở và miễn phí.

# CHƯƠNG 2: THIẾT KẾ HỆ THỐNG

## 2.1. Mô hình triển khai và sơ đồ hệ thống

### 2.1.1. Mô hình triển khai

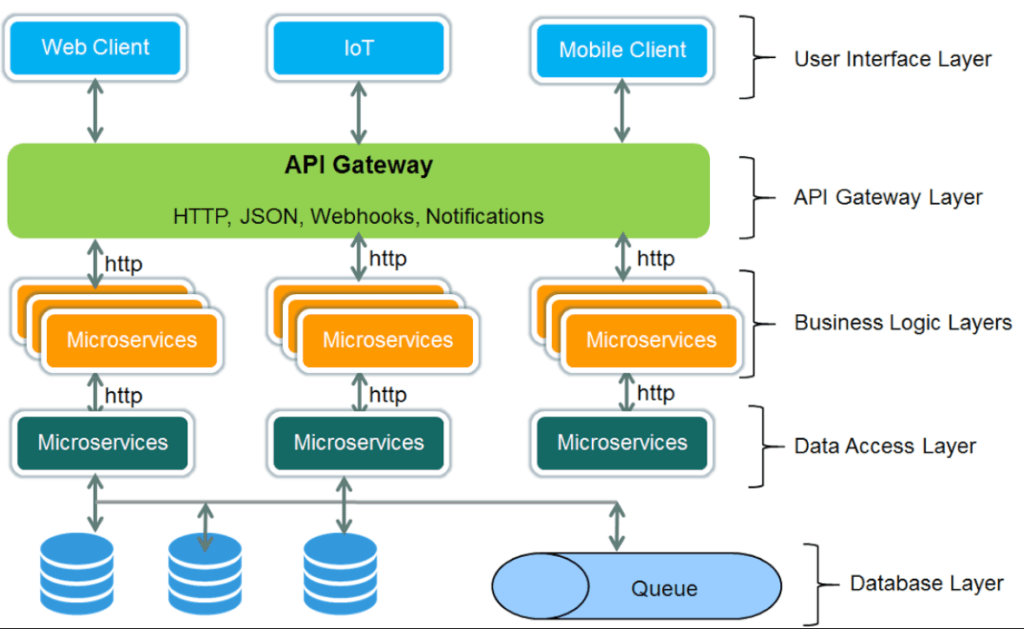
**-** Mô hình phát triển dự án mà em hướng đến đó là mô hình thác nước

- Lý do chọn mô hình:

* Thích hợp cho các dự án đơn giản và nhỏ.
* Hiểu rõ các yêu cầu.
* Dễ hiểu, dễ quản lí.
* Không cần đề ra các mốc thực hiện thuận lợi cho quá trình vừa thực tập vừa làm dự án tốt nghiệp.

### 2.1.2.Sơ đồ hệ thống

Sơ đồ kiến trúc hệ thống mà em làm áp dựng vào xây dựng website là mô hình Microservice



- Lý do chọn mô hình:

- Cho phép lập trình viên linh động hơn trong việc lựa chọn ngôn ngữ, công cụ và nền tảng để phát triển và triển khai các microservice (tuy nhiên trong một hệ thống, việc lựa chọn các ngôn ngữ khác nhau để phát triển các microservice không được khuyến khích)

- Một microservice có thể được phát triển bởi một team nhỏ. Do vậy việc quản lý sẽ dễ dàng hơn.

- Dễ dàng thực hiện tự động tích hợp và tự động triển khai (CI-CD) bằng cách sử dụng một số công cụ như Jenkins, Hudson …

- Mỗi microservice có kích thước nhỏ, giúp cho các lập trình viên dễ tiếp cận, đọc hiểu source code. Do vậy các thành viên mới tham gia team sẽ hòa nhập và đóng góp cho team nhanh hơn.

- Các microservice khởi động nhanh giúp quá trình phát triển, kiểm thử cũng nhanh hơn.

- Dễ dàng mở rộng và tích hợp với các dịch vụ của bên thứ ba.

- Cô lập lỗi tốt hơn, khi một microservice bị lỗi và ngừng hoạt động thì các microservice khác vẫn có thể hoạt động bình thường. Với mô hình nguyên khối, một lỗi nhỏ có thể làm cả hệ thống ngừng hoạt động.

- Khi cần thay đổi một thành phần, thì chỉ cần sửa đổi, cập nhật và triển khai lại thành phần đó chứ không cần triển khai lại toàn bộ hệ thống.

## Tác nhân (Actor)

Người quản lý: Là người quản trị hệ thống.

Khách hàng: Là khách xem đã có tài khoản.

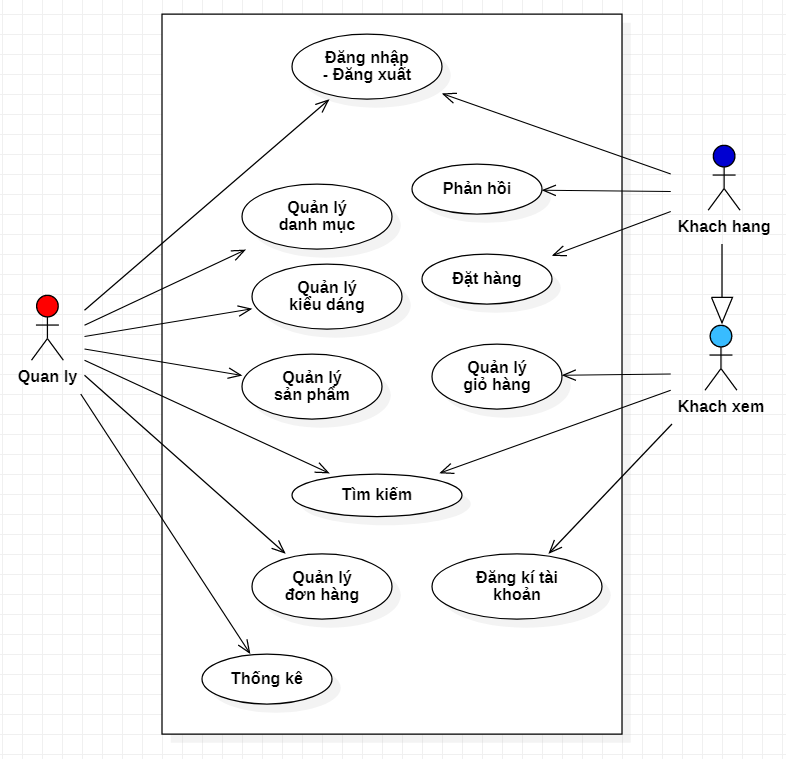
Khách xem: Là thành viên chưa có tài khoản.

**Bảng 3. 1. Tác nhân và các chức năng**

| **Tác nhân** | **Use case** |
| --- | --- |
| **Khách xem** | * Liên hệ với cửa hàng * Đặt hàng * Xem: Xem chi tiết sản phẩm, Xem các danh mục sản phẩm, Xem các kiểu dáng sản phẩm * Tìm kiếm * Đăng ký |
| **Khách hàng** | * Liên hệ với cửa hàng * Đặt hàng * Xem: Xem chi tiết sản phẩm, Xem các danh mục sản phẩm, Xem các kiểu dáng sản phẩm * Tìm kiếm * Xác thực tài khoản * Đăng nhập * Đăng xuất |
| **Quản lý** | * Xác thực tài khoản * Đăng nhập * Đăng xuất * Thống kê |

## Biểu đồ UseCase

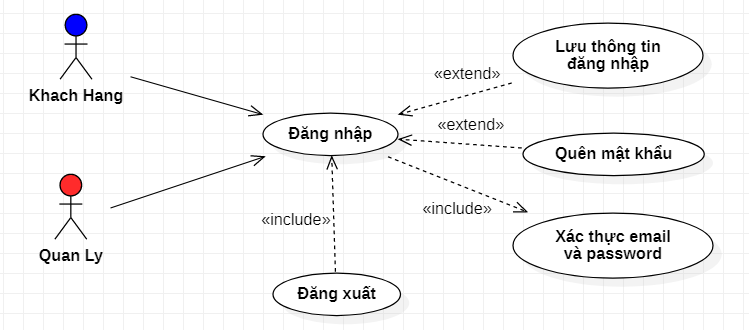
### Biểu đồ Use Case tổng quát



***Hình 3. 1. Use case tổng quát.***

### Biểu đồ Use Case chi tiết

#### Use Case Đăng nhập – Đăng xuất



***Hình 3. 2. Use Case Đăng nhập – Đăng xuất***

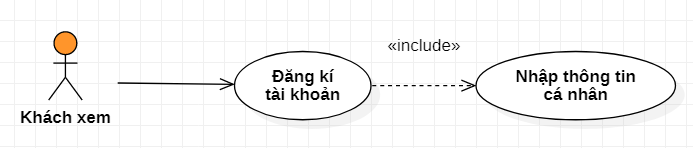
**Bảng 3. 2. Đặc tả Use case Đăng nhập**

|  |  |
| --- | --- |
| **Tên UC** | **Đăng nhập** |
| Tác nhân | Quản lý, khách hàng |
| Mục đích | Đăng nhập hệ thống |
| Mô tả | Người sử dụng đăng nhập vào hệ thống thông qua “Email” và “Password” |
| Tiền điều kiện | Đã có tài khoản |
| Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Đăng nhập” 2. Hệ thống hiển thị form “Đăng nhập” 3. Người sử dụng nhập “Email” và “Password”, gửi thông tin đến hệ thống 4. Hệ thống kiểm tra. Nếu đúng thì cho phép truy cập, và sử dụng. Nếu sai hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin. |
| Luồng sự kiện phụ | 4.1 Nhập sai định dạng “Email” hoặc “Password”, yêu cầu nhập lại  4.2 Nhập thiếu thông tin Email hoặc Password, yêu cầu nhập đầy đủ  4.3 Bỏ trống thông tin Email, Password, yêu cầu nhập đầy đủ  4.4. Nhập sai Email, Password, đăng nhập thất bại |

**Bảng 3. 3. Đặc tả Use case Đăng xuất.**

|  |  |
| --- | --- |
| **Tên UC** | **Đăng xuất** |
| Tác nhân | Quản lý, khách hàng |
| Mục đích | Đăng xuất hệ thống |
| Mô tả | Người sử dụng thoát khỏi hệ thống |
| Tiền điều kiện | Đã đăng nhập |
| Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Đăng xuất” 2. Hệ thống chuyển về “Trang chủ” |

#### Use Case Đăng ký tài khoản

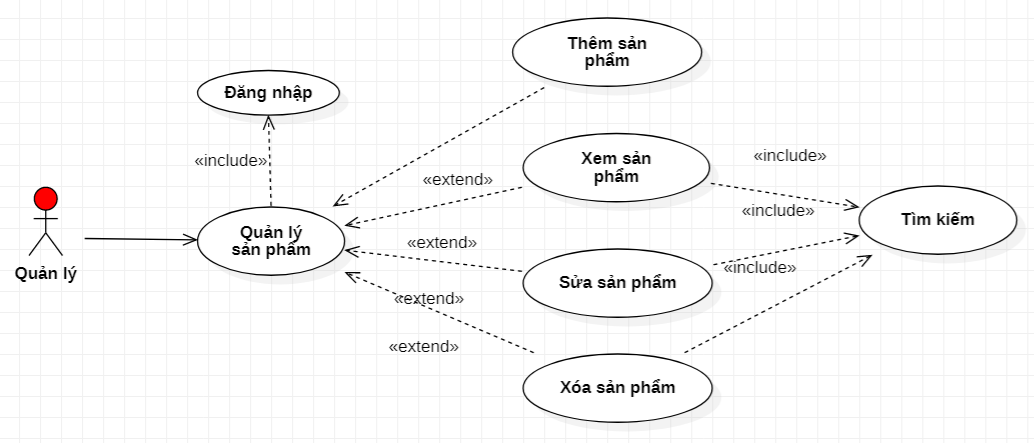


***Hình 3. 3. Use Case Đăng ký tài khoản.***

**Bảng 3. 4. Đặc tả Use case Đăng ký tài khoản**

|  |  |
| --- | --- |
| **Tên UC** | **Đăng kí tài khoản** |
| Tác nhân | Khách xem |
| Mục đích | Đăng ký hệ thống |
| Mô tả | Người dùng đăng ký làm thành viên của hệ thống |
| Tiền điều kiện | Khách xem chưa có tài khoản hệ thống |
| Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Đăng ký”. 2. Hệ thống hiển thị form “Đăng ký”. 3. Người sử dụng nhập thông tin vào form “Đăng ký tài khoản” 4. Hệ thống lưu thông tin khách hàng vào CSDL và chuyển sang form “Thông tin”. 5. Hệ thống lưu thông tin khách hàng và CSDL và chuyển sang trang “Sản phẩm” |
| Luồng sự kiện phụ | 3.1 Nhập sai định dạng Email, password, yêu càu nhập lại  3.2 Nhập thiếu thông tin: email, usename, password, confirm password, yêu câgu nhập đầy đủ  3.3. Nhập password và confirm password không trùng nhau, yêu càu nhập lại  3.4 Thông tin tài khoản đã tồn tại trong CSDL, thông báo tài khoản đã tồn tại  4.1 Nhập sai định dạng thông tin của form, yêu cầu nhập lại  4.2 Nhập thiếu thông tin form, yêu cầu nhập đầy đủ  4.3 Bỏ trống thông tin, yêu cầu nhập đầy đủ |

#### Use Case Quản lý sản phẩm



***Hình 3. 6 Use Case Quản lý sản phẩm.***

**Bảng 3. 11. Đăc tả Use case Thêm sản phẩm**

|  |  |
| --- | --- |
| **Tên UC** | **Them san pham** |
| Tác nhân | Quản lý |
| Mục đích | Thêm sản phẩm |
| Mô tả | Nhân viên thêm sản phẩm |
| Tiền điều kiện | Đã đăng nhập |
| Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Thêm sản phẩm” 2. Hệ thống hiển thị form “Thêm sản phẩm” 3. Người sử dụng điền đầy đủ các thông tin 4. Hệ thống kiểm tra dữ liệu. Nếu đúng thông báo thành công và lưu vào CSDL. Ngược lại, yêu cầu nhập lại thông tin |
| Luồng sự kiện phụ | 4.1. Sai định dạng thông tin: số lượng, giá sản phẩm chứa chữ cái. Yêu cầu nhập lại  4.2. Tên sản phẩm trùng với sản phẩm đã lưu trong CSDL, thông báo trùng lặp  4.3. Bỏ trống thông tin, yêu cầu nhập đầy đủ |

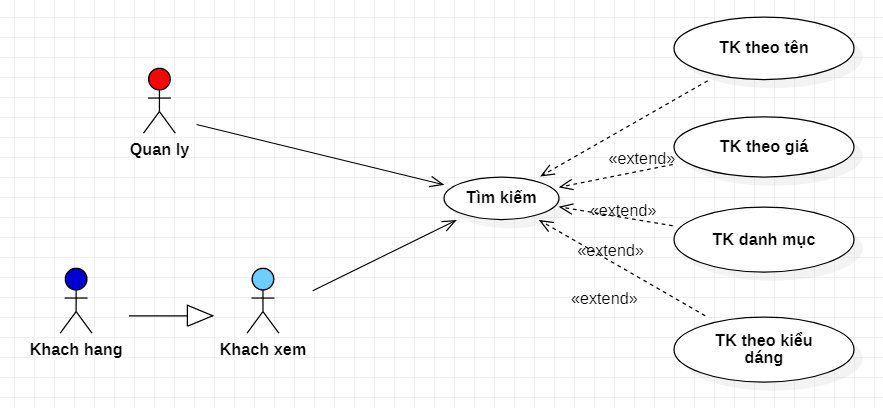
**Bảng 3. 12. Đăc tả Use case Sửa sản phẩm**

|  |  |
| --- | --- |
| **Tên UC** | **Sua san pham** |
| Tác nhân | Quản lý |
| Mục đích | Sửa thông tin sản phẩm |
| Mô tả | Quản lý sửa thông tin sản phẩm |
| Tiền điều kiện | Đã đăng nhập |
| Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Sửa sản phẩm” 2. Hệ thống hiển thị form “Sửa sản phẩm” 3. Người sử dụng điền đầy đủ các thông tin 4. Hệ thống kiểm tra dữ liệu. Nếu đúng thông báo thành công và lưu vào CSDL. Ngược lại, yêu cầu nhập lại thông tin |
| Luồng sự kiện phụ | 4.1. Sai định dạng thông tin sản phẩm, yêu cầu nhập lại  4.2. Để trống thông tin sản phẩm, yêu cầu nhập đầy đủ |

***Bảng 3. 13. Đăc tả Use case Xóa sản phẩm***

|  |  |
| --- | --- |
| **Tên UC** | **Xoa san pham** |
| Tác nhân | Quản lý |
| Mục đích | Xóa sản phẩm |
| Mô tả | Quản lý xóaphẩm |
| Tiền điều kiện | Đã đăng nhập |
| Luồng sự kiện chính | 1. Người sử dụng chọn chức năng “Xóa sản phẩm” 2. Hệ thống kiểm tra dữ liệu. Nếu đúng thông báo thành công và xóa. Ngược lại, yêu cầu kiểm tra lại |
| Luồng sự kiện phụ | 2.1. Sản phẩm tồn tại sản phẩm thì không được xóa. |

#### Use Case Tìm kiếm

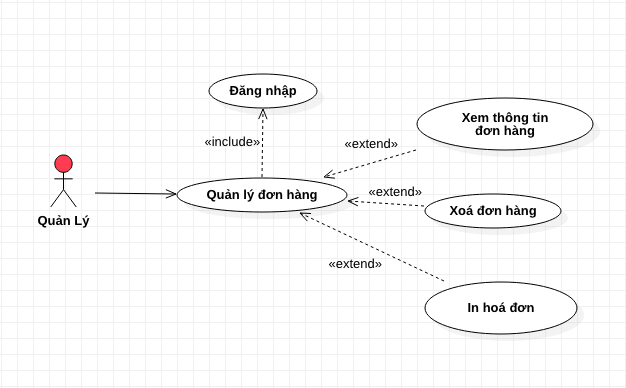


***Hình 3. 7 Use Case Tìm kiếm.***

**Bảng 3. 14. Đăc tả Use case Tìm kiếm sản phẩm**

|  |  |
| --- | --- |
| **Tên UC** | **Tim kiem san pham** |
| Tác nhân | Quản lí, khách hàng, khách xem |
| Mục đích | Tìm kiếm sản phẩm |
| Mô tả | Quản lý, người dùng tìm kiếm sản phẩm |
| Luồng sự kiện chính | 1. Người sử dụng nhập từ khóa sản phẩm 2. Hệ thống kiểm tra thông tin và hiển thị danh sách sản phẩm |
| Luồng sự kiện phụ | 2.1. Nhập sai tên sản phẩm, thông báo không có kết quả |

#### Use Case Quản lý đơn hàng

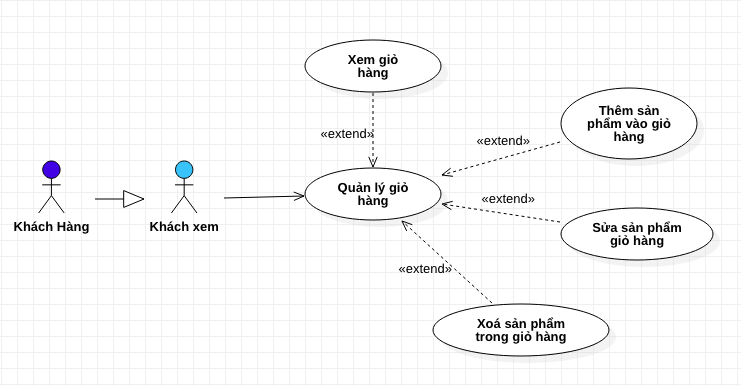


***Hình 3. 8 Use Case Quản lý đơn hàng.***

**Bảng 3. 15. Đăc tả Use case Quản lý đơn hàng**

|  |  |
| --- | --- |
| **Tên UC** | **Quản lý đơn hàng** |
| Tác nhân | Quản lý |
| Mục đích | Xem, xóa, in hoá đơn |
| Mô tả | Quản lý đơn đơn hàng |
| Tiền điều kiện | Đã đăng nhập |
| Luồng sự kiện chính | 1. Actor chọn chức năng “Quản lí đơn hàng”. 2. Actor có thể chọn các tùy chọn: xem, xóa, in hoá đơn.   - Xem thông tin đơn hàng:  1. Actor chọn “Xem”  2. Hệ thống hiển thị thông tin đơn hàng  - Xoá đơn hàng:   * + - 1. Actor chọn “Xoá”       2. Hệ thống hiển thị thông báo xác nhận xoá. Nếu đồng ý, thực hiện xoá và thông báo xoá thành công. Ngược lại quay trở về trang “Quản lý đơn hàng”   - In hoá đơn:   1. Actor chọn “Tải xuống” 2. Hệ thống thực hiện xuất thông tin đơn hàng thành file excel. |
| Luồng sự kiện phụ |  |

#### Use Case Quản lý giỏ hàng

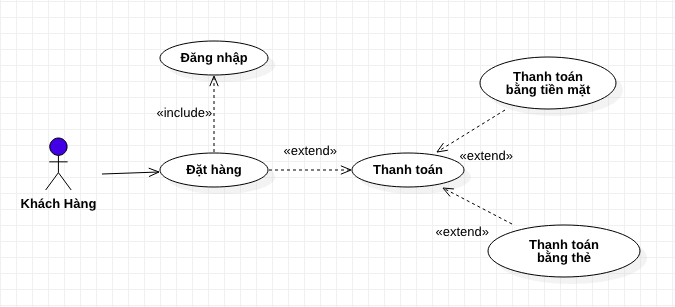


***Hình 3. 9 Use Case Quản lý giỏ hàng.***

**Bảng 3. 16. Đăc tả Use case Quản lý giỏ hàng**

|  |  |
| --- | --- |
| **Tên UC** | **Quản lý giỏ hàng** |
| Tác nhân | Khách hàng, khách xem |
| Mục đích | Xem, thêm, sửa số lượng, xoá sản phẩm trong giỏ hàng |
| Mô tả | Quản lý giỏ hàng |
| Tiền điều kiện |  |
| Luồng sự kiện chính | * Thêm sản phẩm vào giỏ hàng:  1. Từ trang chính hoặc trang “Chi tiết sản phẩm" Actor chọn “Thêm vào giỏ”. 2. Hệ thống sẽ thực hiện hiển thị sản phẩm trong giỏ hàng.  * Xem sản phẩm trong giỏ hàng:  1. Từ trang chính, actor nhấn vào biểu tượng giỏ hàng. 2. Hệ thống thực hiện yêu cầu, hiển thị trang “Giỏ hàng”, hiển thị các sản phẩm có trong giỏ.  * Sửa số lượng sản phẩm trong giỏ hàng:  1. Từ trang chính, actor nhấn vào biểu tượng giỏ hàng 2. Hệ thống thực hiện yêu cầu, hiển thị trang “Giỏ hàng” 3. Actor nhập thay đổi số lượng sản phẩm có trong giỏ và nhấn “Thay đổi” 4. Hệ thống thực hiện yêu cầu, cập nhật lại số lượng sản phẩm trong giỏ.  * Xoá sản phẩm trong giỏ hàng:  1. Trong trang “Giỏ hàng”, actor nhấn “Xoá” sản phẩm muốn xoá 2. Hệ thống đưa ra thông báo, nếu chấp nhận thì thực hiện xoá sản phẩm, ngược lại thì giữ nguyên. |
| Luồng sự kiện phụ | 7.1. Người dùng nhập số âm, thông báo số lượng phải lớn hơn 1. |

#### Use Case Đặt hàng

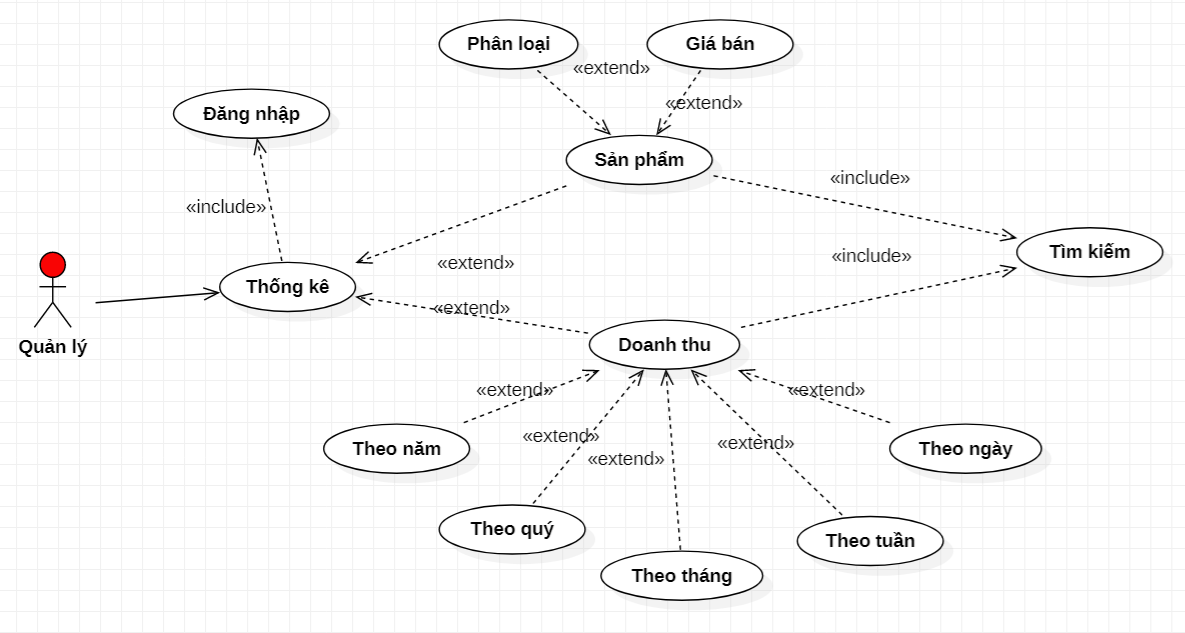


***Hình 3. 10 Use Case Đặt hàng.***

**Bảng 3. 17. Đăc tả Use case Đặt hàng**

|  |  |
| --- | --- |
| **Tên UC** | **Đặt hàng** |
| Tác nhân | Khách hàng |
| Mục đích | Đặt hàng thanh toán sản phẩm |
| Tiền điều kiện | Đã đăng nhập |
| Luồng sự kiện chính | 1. Tại giao diện “Giỏ hàng”, actor chọn “Đặt hàng” 2. Hệ thống thực hiện yêu cầu, hiển thị giao diện thanh toán. 3. Actor chọn hình thức thanh toán 4. Actor chọn địa chỉ vận chuyển theo địa chỉ đăng ký hoặc thay đối địa chỉ vận chuyển 5. Lưu vào database |
| Luồng sự kiện phụ | 4.1 Người dùng thay đổi địa chỉ vận chuyển không đúng định dạng: số điện thoại chứa chữ, tên thành phố chứa chữ số. Yêu cầu nhập lại |

#### Use Case Thống kê



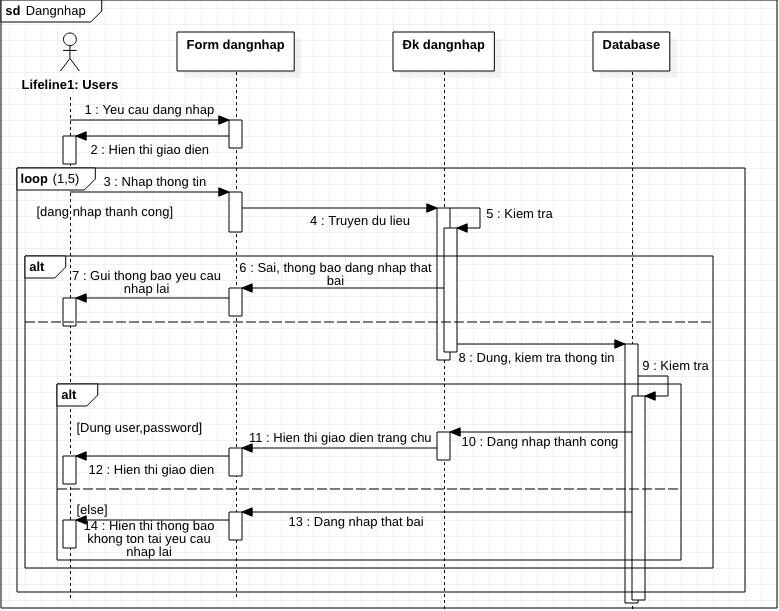
***Hình 3. 11 Use Case Thống kê.***

**Bảng 3. 18. Đăc tả Use case Thống kê**

|  |  |
| --- | --- |
| **Tên UC** | **Thống kê** |
| Tác nhân | Quản lý |
| Mục đích | Thống kê |
| Tiền điều kiện | Đã đăng nhập |
| Luồng sự kiện chính | 1. Hệ thống hiện thị form đăng nhập. 2. Actor nhập email và pass. 3. Hệ thống kiểm tra thông tin tài khoản. 4. Actor chọn chức năng “Thống kê”. 5. Actor có thể chọn các tùy chọn: thống kê số lượng sản phẩm, thông kê doanh thu. |

## Biểu đồ tuần tự

### Biểu đồ tuần tự Đăng nhập



***Hình 3. 12 Biểu đồ tuần tự Đăng nhập.***

- Loop (1,5): Thực hiện nhập email và mật khẩu cho đến khi đúng định dạng và thông tin trong CSDL. Thực hiện nhập tối thiểu 1 lần và tối đa 5 lần

- 5-Kiểm tra định dạng: Kiểm tra email đủ @ và không quá 23 kí tự, Kiểm tra mật khẩu ít nhất chứ 8 kí tự, kiểm tra có bỏ trống hay không.

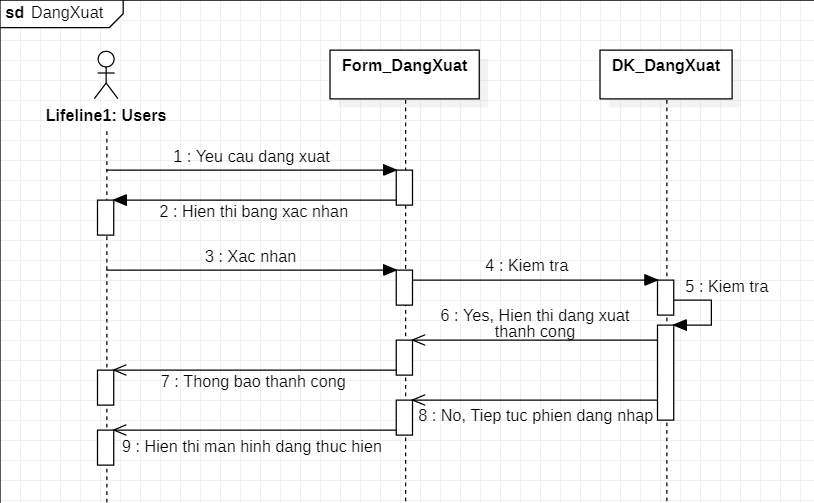
- 9-Kiểm tra: Kiểm tra thông tin đăng nhập, tồn tại tài khoản đăng nhập (có thông tin trong CSDL), tài khoản đang đăng nhập chỉ ở trên một thiết bị

- Alt: Kiểm tra định dạng và thông tin, trả về kết quả tại nhánh đúng và sai

- Form dangnhap: Hiển thị giao diện đăng nhập

- ĐK dangnhap: Kiểm tra định dạng dữ liệu, xử lý yêu cầu

- Database: Kiểm tra, lưu giữ, truy xuất dữ liệu



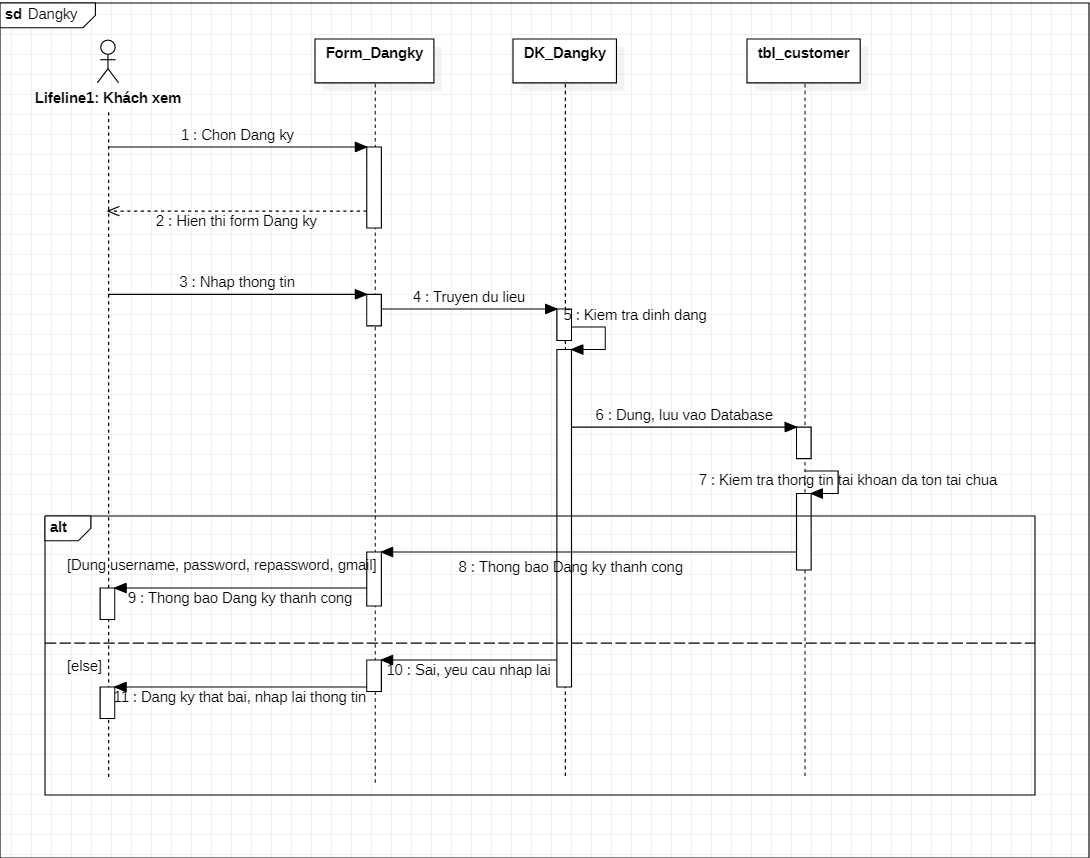
***Hình 3. 13 Biểu đồ tuần tự Đăng xuất.***

- 5- Kiểm tra xác nhận yêu cầu đăng xuất

- Form\_Dangxuat: Hiển thị giao diện đăng xuất

- DK\_Dangxuat: Thực hiện yêu cầu đăng xuất

### Biểu đồ tuần tự Đăng kí tài khoản



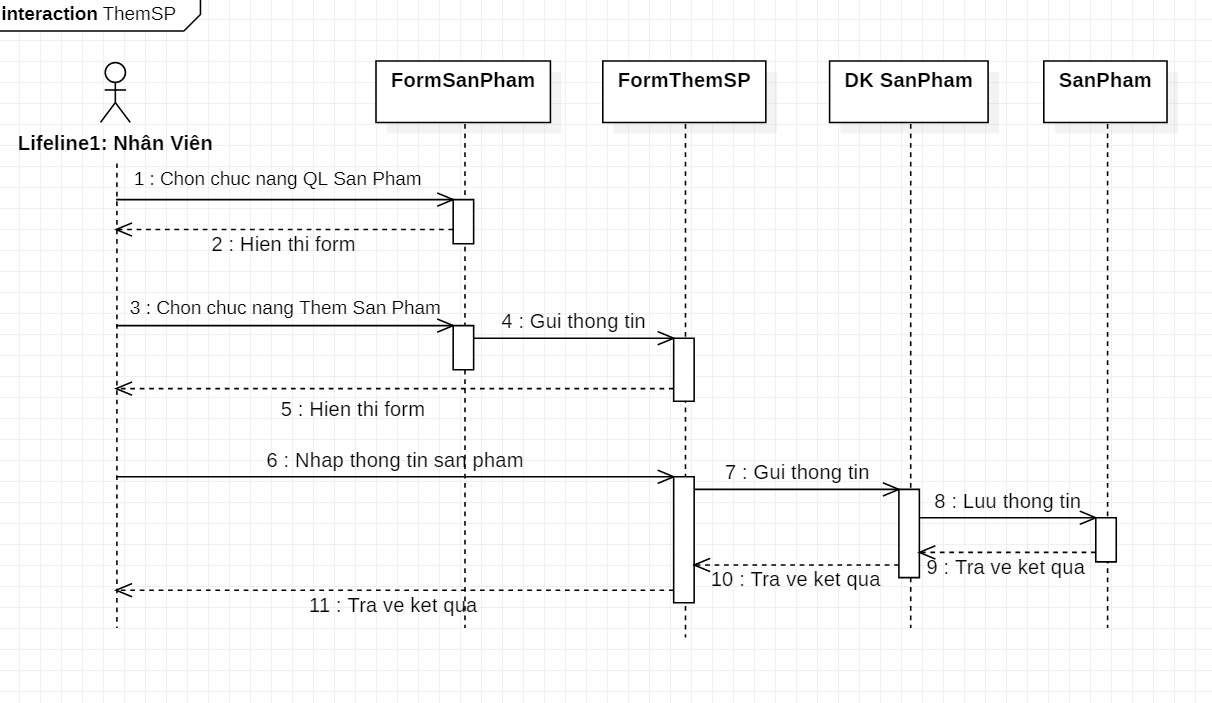
***Hình 3. 14 Biểu đồ tuần tự Đăng kí tài khoản.***

- 5-Kiểm tra định dạng: Kiểm tra thông tin email phải chứa @ và dài không quá 23 kí tự, kiểm tra có bỏ trống hay không.

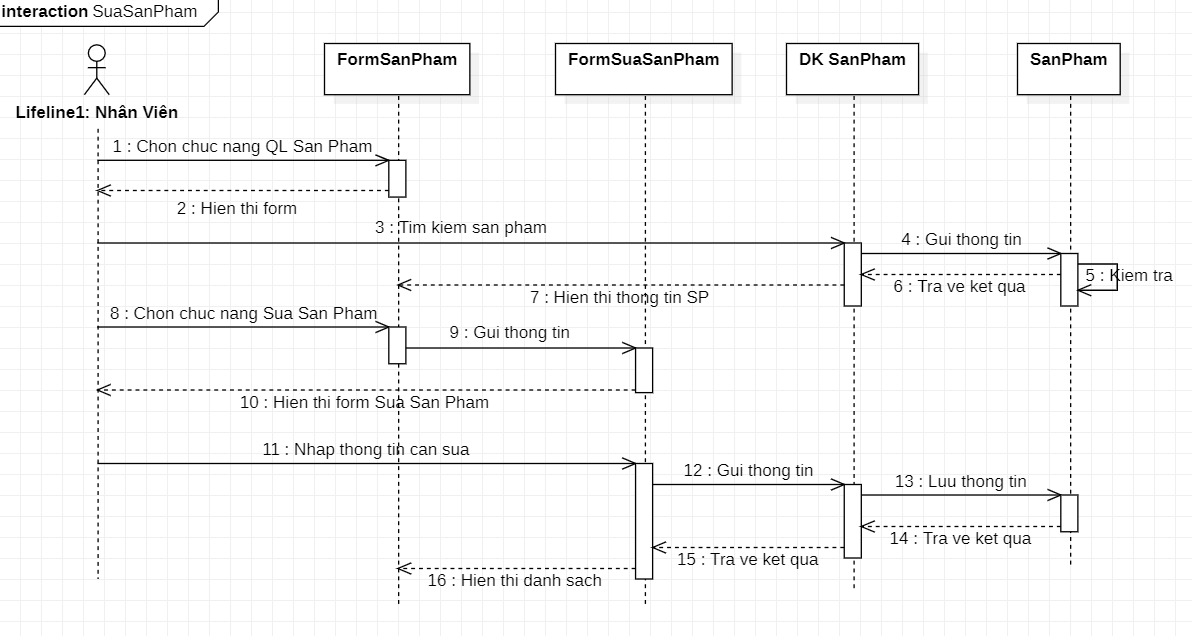
- 7-Kiểm tra tài khoản: Kiểm tra tài khoản đã có dữ liệu được lưu chưa (tồn tại tài khoản)

- Alt: Trả về kết quả theo nhánh đúng hoặc sai

### Biểu đồ tuần tự Quản lý sản phẩm

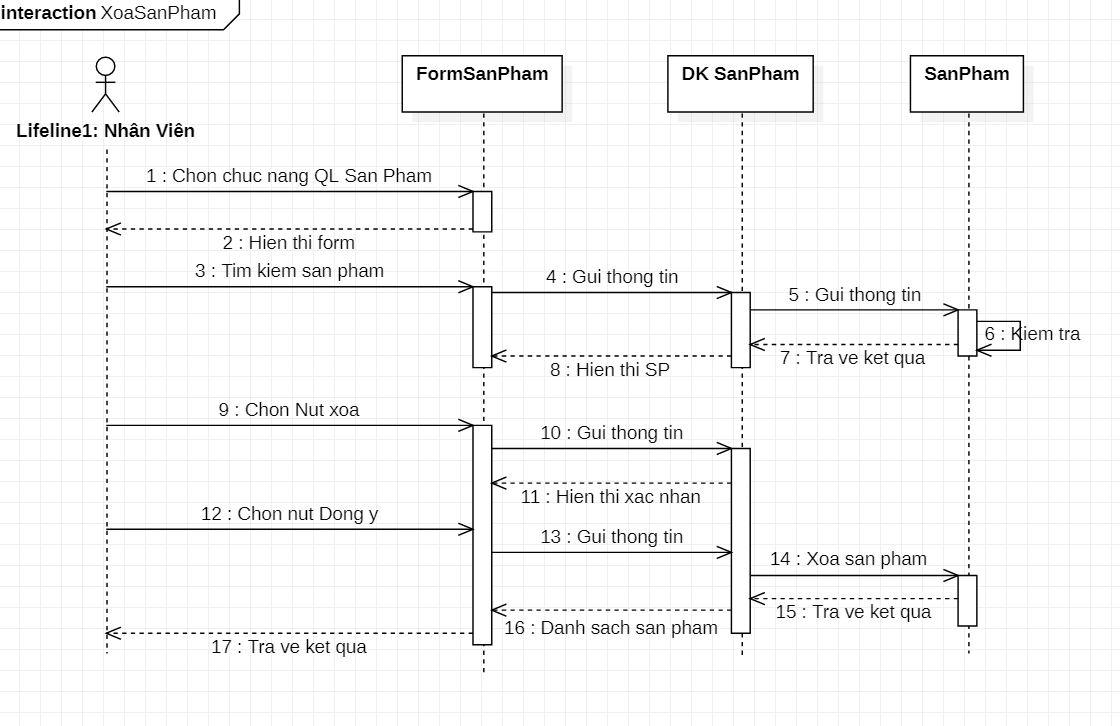


***Hình 3. 21 Biểu đồ tuần tự Thêm Sản Phẩm.***



***Hình 3. 22 Biểu đồ tuần tự Sửa Sản Phẩm.***

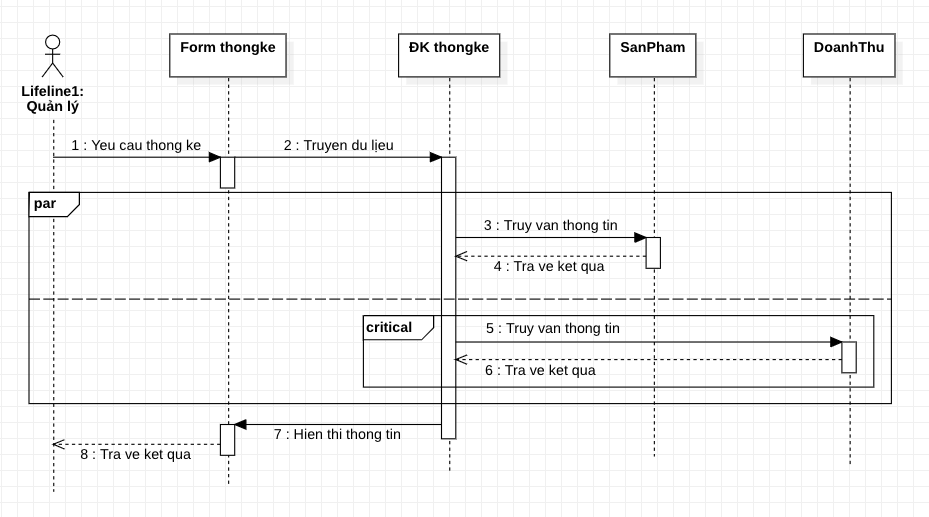
- 5-Kiểm tra: Kiểm tra, truy xuất dữ liệu sản phẩm



***Hình 3. 23 Biểu đồ tuần tự Xóa Sản Phẩm.***

- 6-Kiểm tra: Kiểm tra, truy xuất dữ liệu sản phẩm

### Biểu đồ tuần tự Thống kê

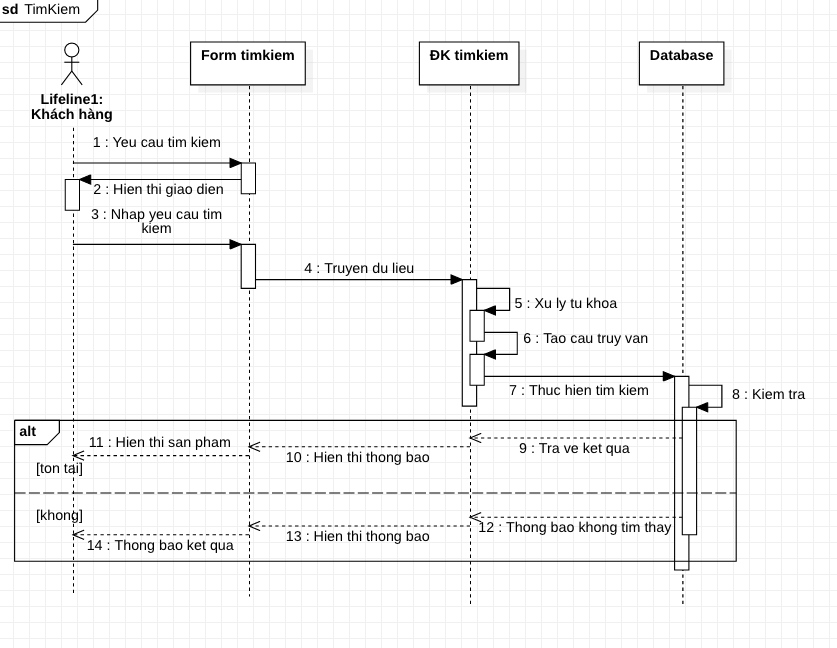


***Hình 3. 24 Biểu đồ tuần tự Thống kê.***

- Par: Thực hiện yêu cầu dữ liệu không theo thứ tự, có thể thực hiện yêu cầu dữ liệu bất kì của sản phẩm hoặc doanh thu mà không cần xét đến thứ tự thực hiện.

- Critical: Luôn luôn thực hiện truy vấn thông tin doanh thu và trả về kết quả theo doanh thu theo thứ tự.

### Biểu đồ tuần tự Tìm kiếm

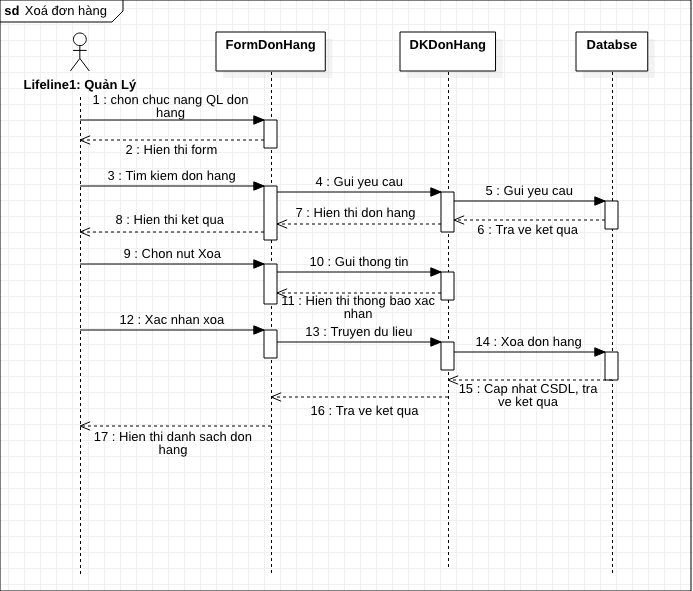


***Hình 3. 25 Biểu đồ tuần tự Tìm kiếm.***

- 8-Kiểm tra: So sánh với dữ liệu trên CSDL

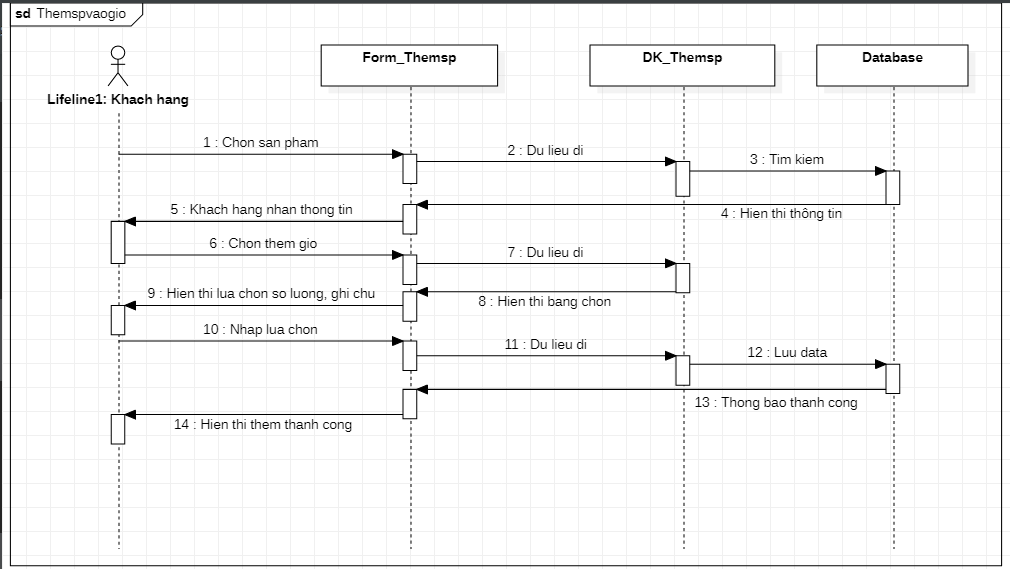
- Alt: Trả về kết quả theo trường hợp có kết quả tìm kiếm hoặc không

### Biểu đồ tuần tự Quản lý đơn hàng

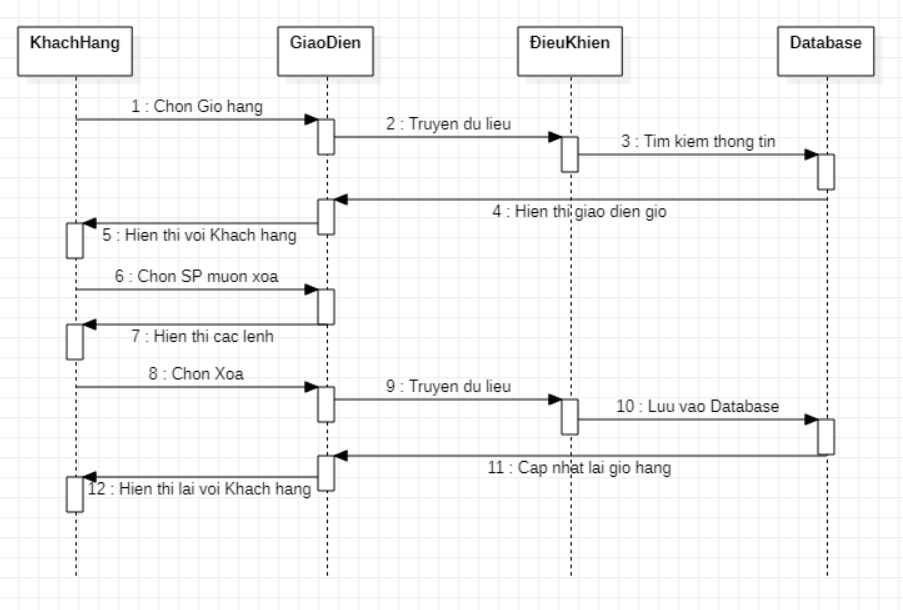


***Hình 3. 26 Biểu đồ tuần tự Xóa đơn hàng.***

### Biểu đồ tuần tự Quản lý giỏ hàng

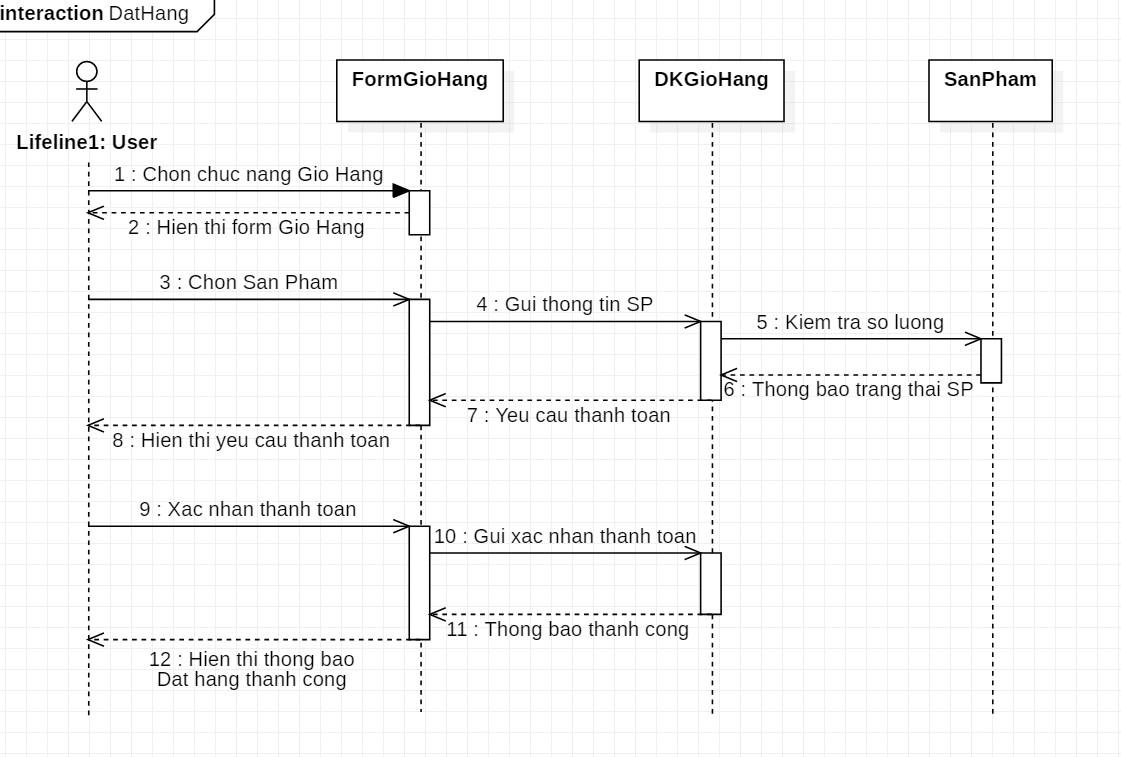


***Hình 3. 27 Biểu đồ tuần tự Thêm sản phẩm.***



***Hình 3. 28 Biểu đồ tuần tự Xóa sản phẩm trong giỏ hàng.***

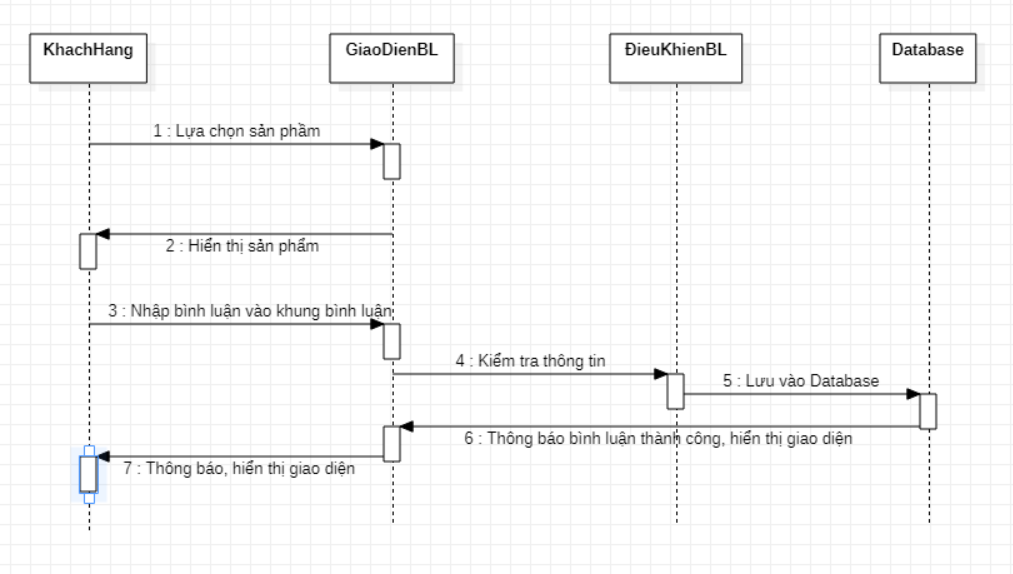
### Biểu đồ tuần tự Đặt hàng



***Hình 3. 29 Biểu đồ tuần tự Đặt hàng.***

- 5-Kiểm tra số lượng: Kiểm tra xem sản phẩm còn hàng trong kho hay không

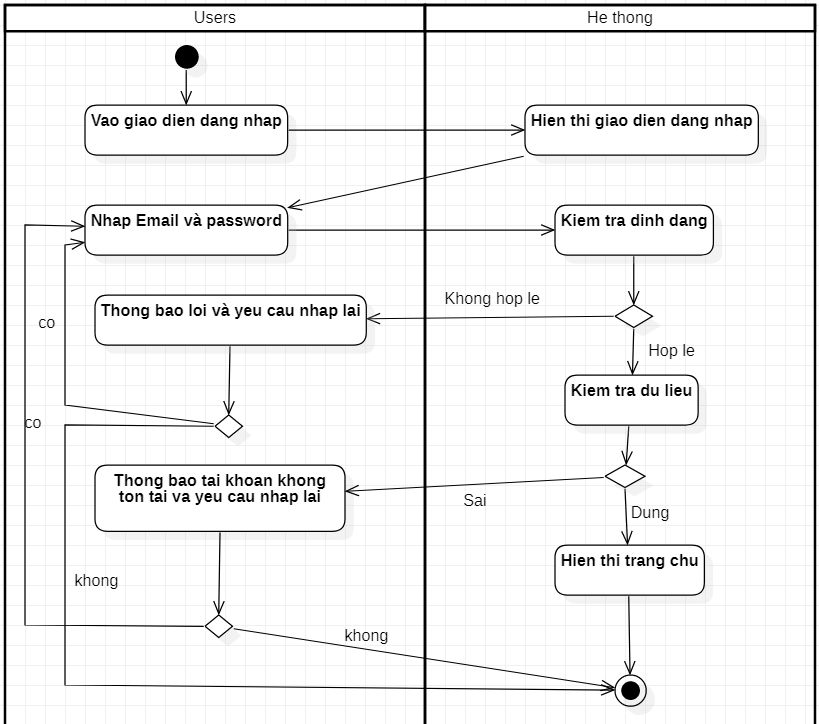
### Biểu đồ tuần tự Bình luận phản hồi



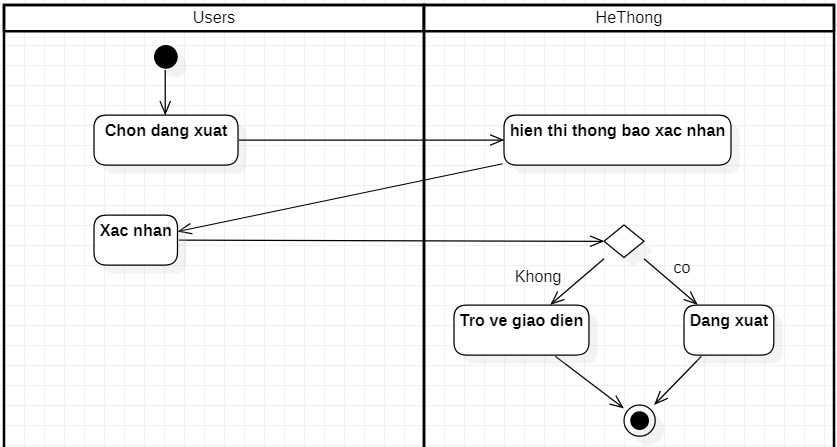
***Hình 3. 30 Biểu đồ tuần tự Bình luận, phản hồi.***

## Biểu đồ hoạt động

### Biểu đồ hoạt động Đăng nhập

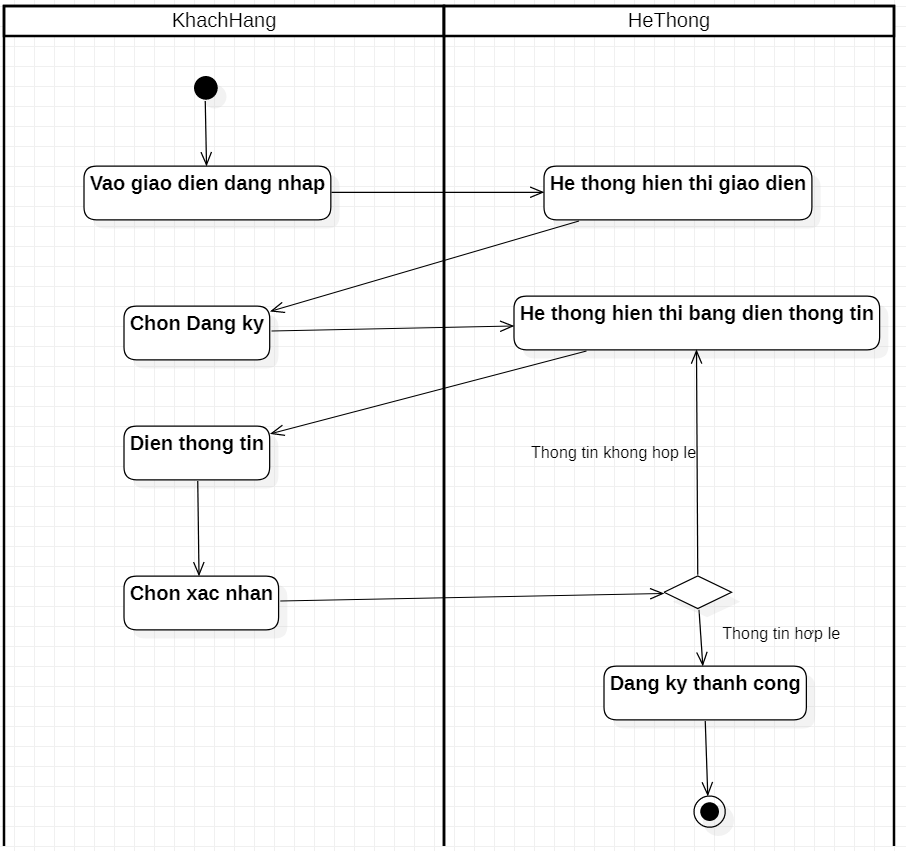


***Hình 3. 31 Biểu đồ hoạt động Đăng nhập.***



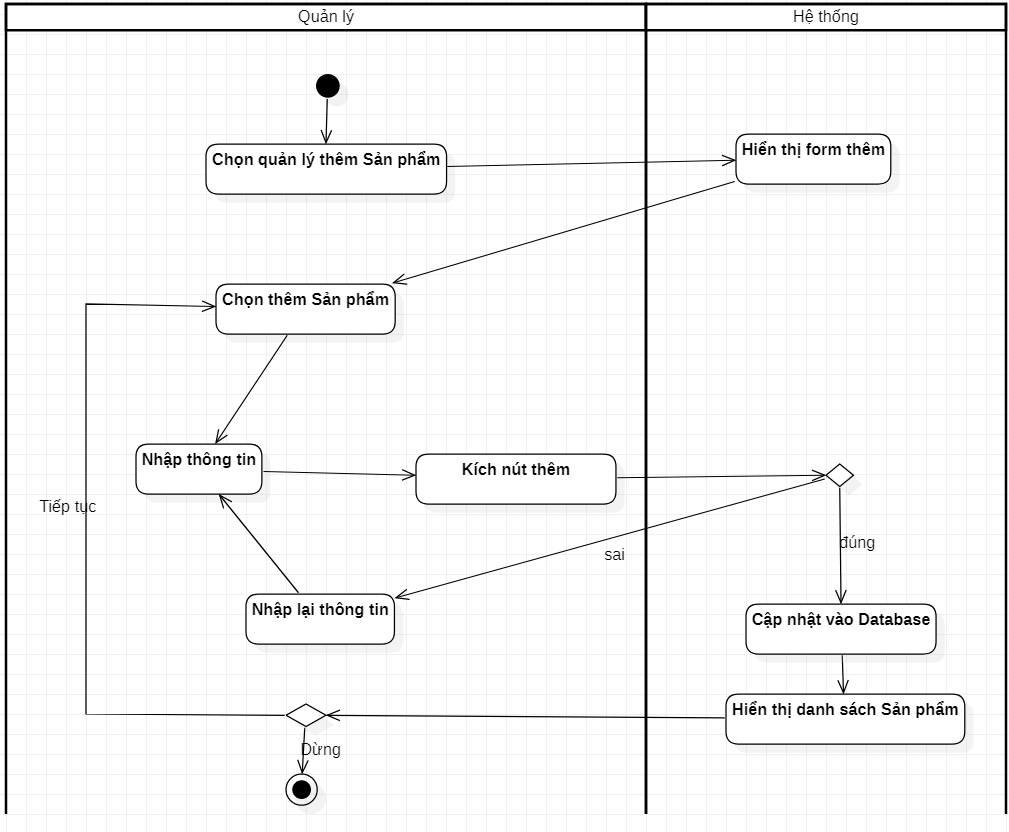
***Hình 3. 32 Biểu đồ hoạt động Đăng xuất.***

### Biểu đồ hoạt động Đăng kí tài khoản

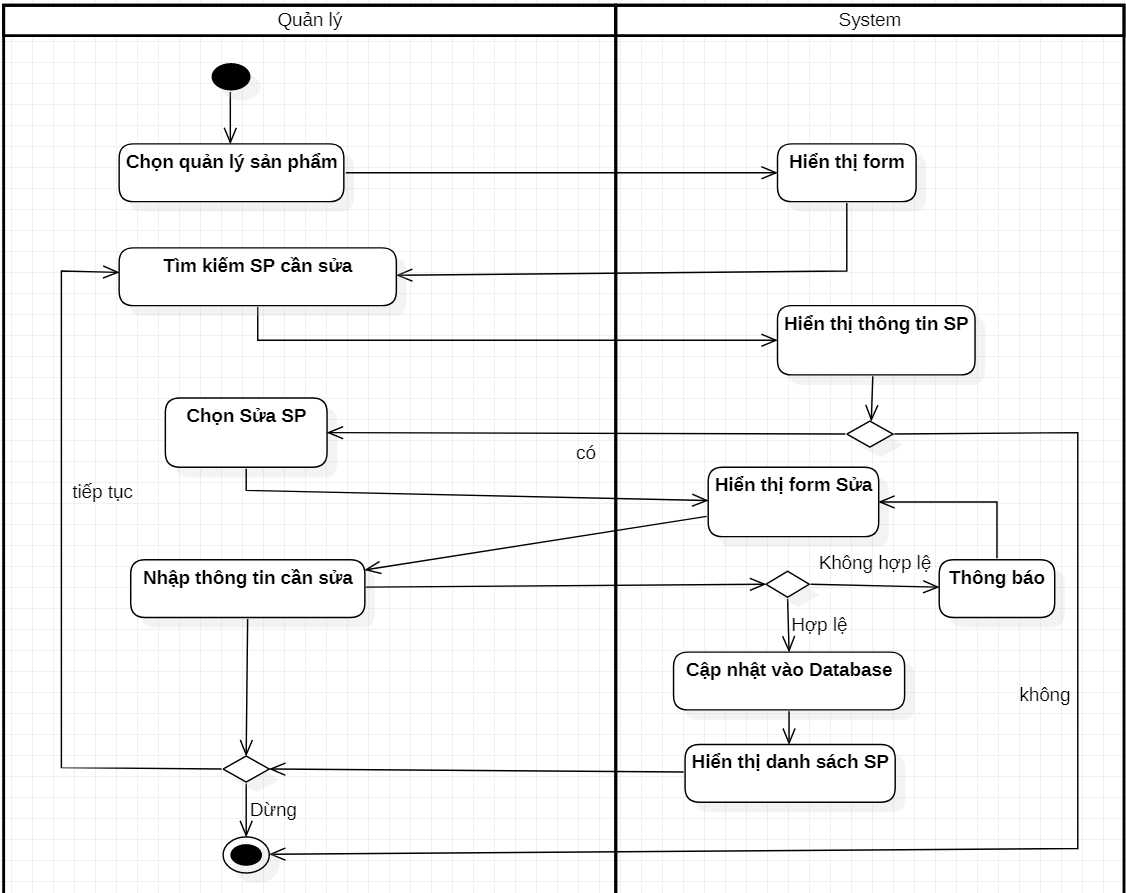


***Hình 3. 33 Biểu đồ hoạt động Đăng kí tài khoản.***

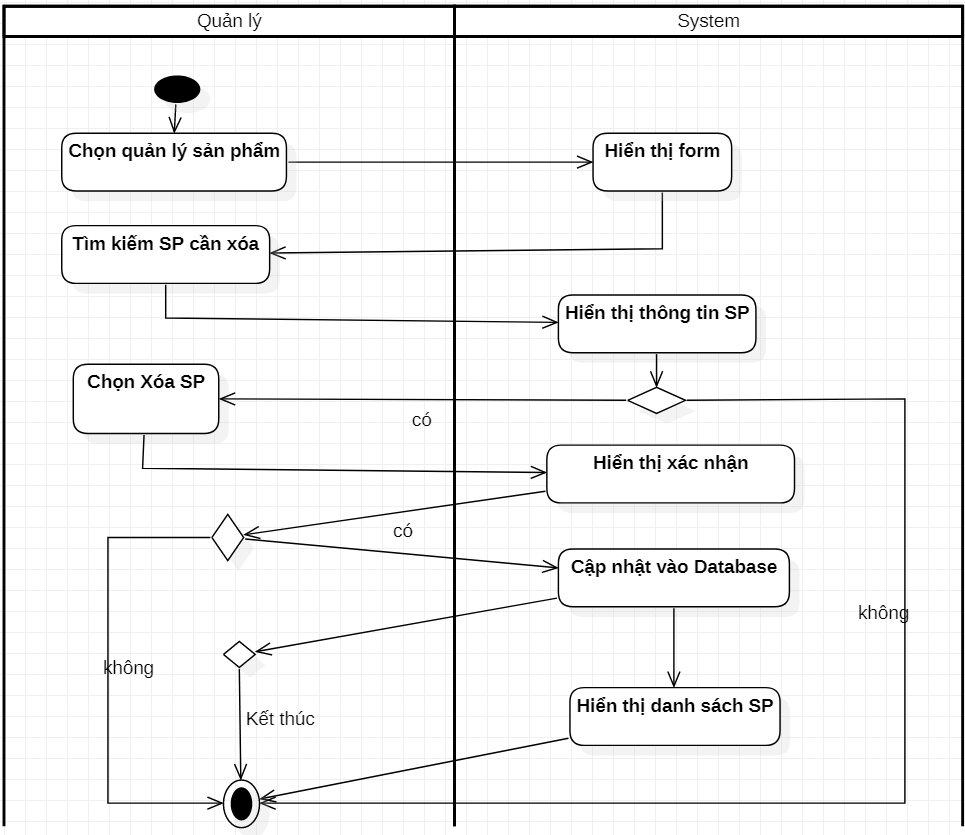
### Biểu đồ hoạt động Quản lý sản phẩm



***Hình 3. 40 Biểu đồ hoạt động Thêm Sản Phẩm.***

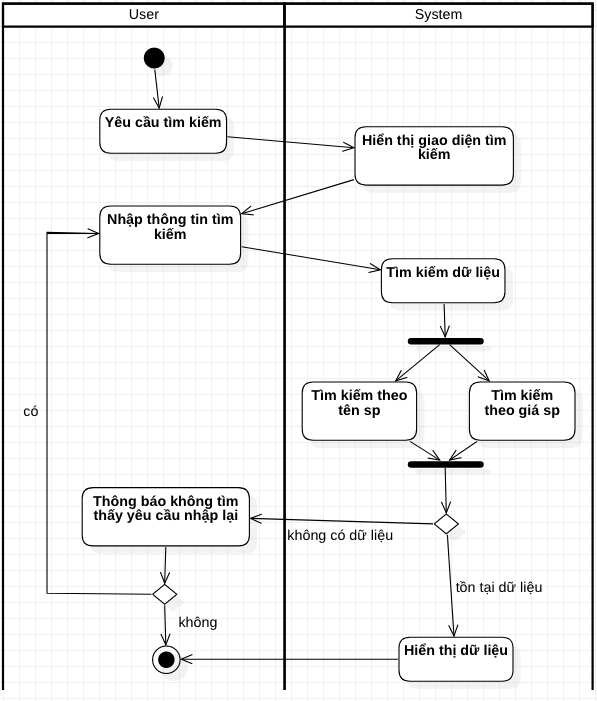


***Hình 3. 41 Biểu đồ hoạt động Sửa Sản Phẩm.***



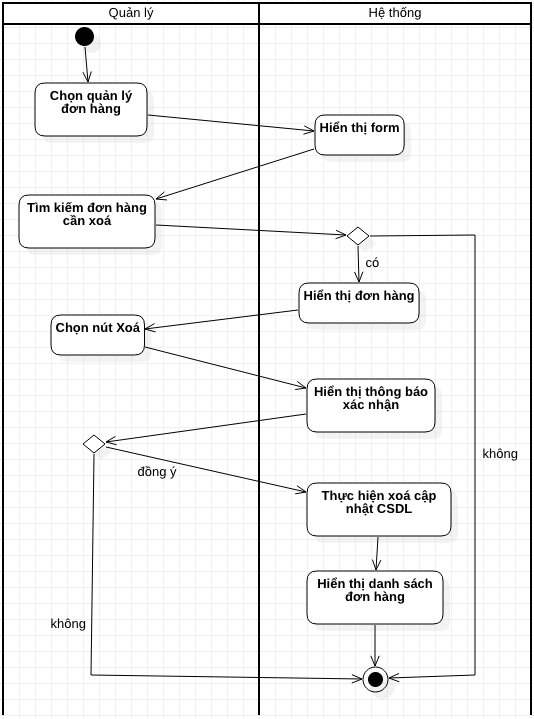
***Hình 3. 42 Biểu đồ hoạt động Xóa Sản Phẩm.***

### Biểu đồ hoạt động Tìm kiếm



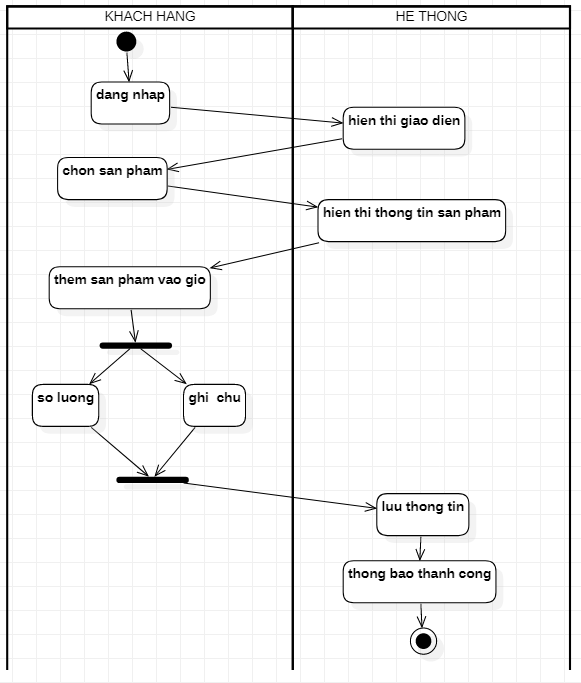
***Hình 3. 43 Biểu đồ hoạt động Tìm kiếm.***

### Biểu đồ hoạt động Quản lý đơn hàng

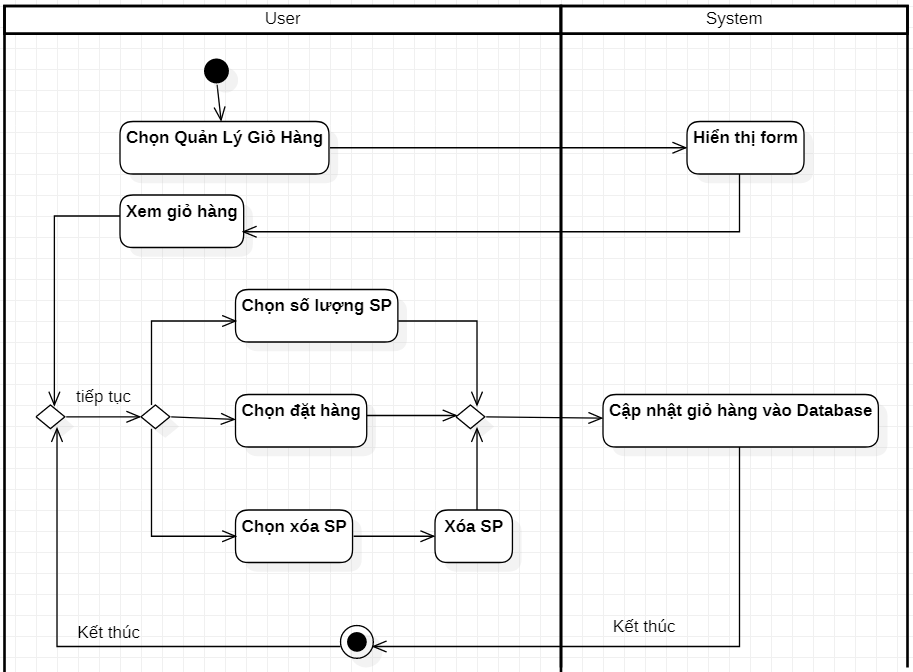


***Hình 3. 44 Biểu đồ hoạt động Quản lý đơn hàng.***

### Biểu đồ hoạt động Quản lý giỏ hàng

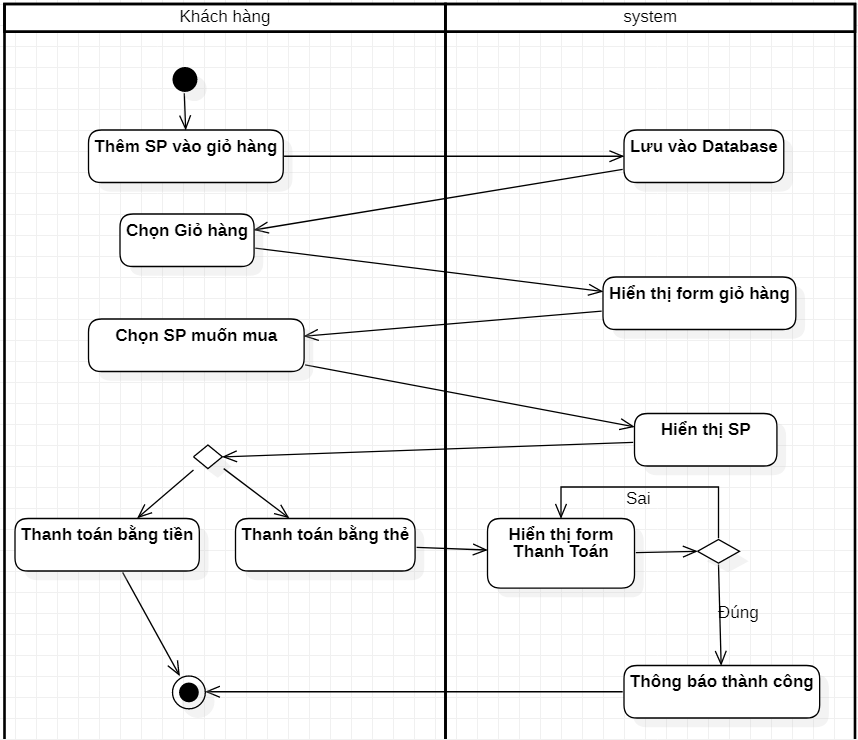


Hình 3. 45 Biểu đồ hoạt động Thêm sản phẩm.



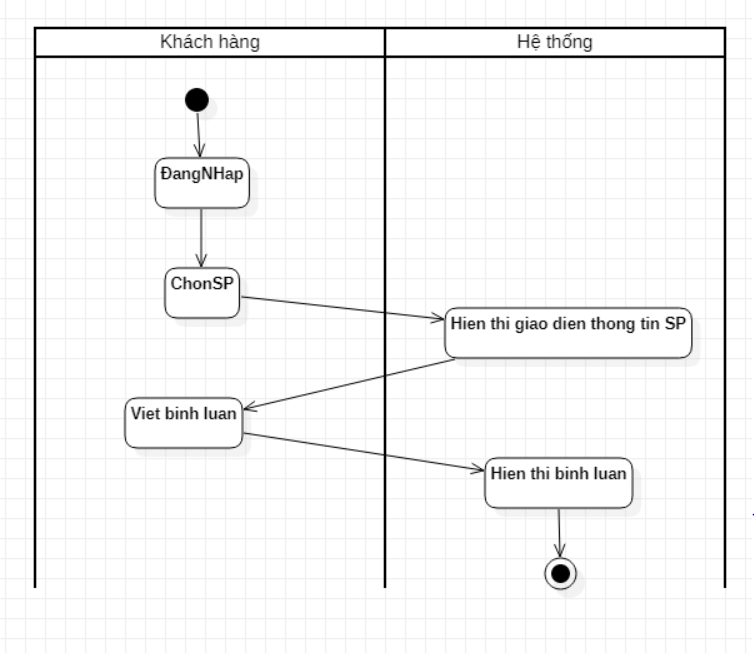
***Hình 3. 46 Biểu đồ hoạt động Xóa sản phẩm trong giỏ hàng.***

### Biểu đồ hoạt động Đặt hàng



***Hình 3. 47 Biểu đồ hoạt động Đặt hàng.***

### Biểu đồ hoạt động Bình luận phản hồi



Hình 3. 48 Biểu đồ hoạt động Bình luận phản hồi.

# TÀI LIỆU THAM KHẢO