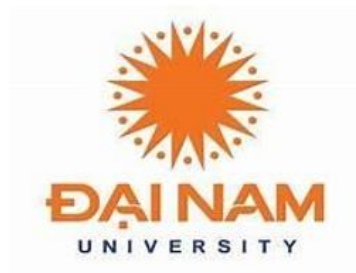


**TRƯỜNG ĐẠI HỌC ĐẠI NAM  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG BÁN ĐỒ TRANG SỨC  
CHO CỬA HÀNG KIRA STORE TRÊN NỀN  
TẢNG ASP.NET CORE WEB API VÀ REACTJS**

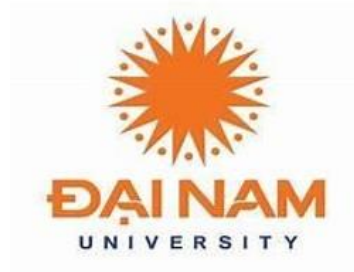
**SINH VIÊN THỰC HIỆN : ĐỖ THỊ XƯA**

**MÃ SINH VIÊN : 1451020278**

**KHOA : CÔNG NGHỆ THÔNG TIN**

**Hà Nội - 2024**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ THỊ XỬA**

**XÂY DỰNG HỆ THỐNG BÁN ĐỒ TRANG SỨC  
CHO CỬA HÀNG KIRA STORE TRÊN NỀN TẢNG  
ASP.NET CORE WEB API VÀ REACTJS**

**CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN**  
**MÃ SỐ : 74.80.201**

**NGƯỜI HƯỚNG DẪN : ThS. Lê Trung Hiếu**

**Hà Nội - 2024**

## LỜI CAM ĐOAN

Em Đỗ Thị Xưa, dưới đây xin ký tên xác nhận rằng toàn bộ nội dung báo cáo và mã nguồn trong xây dựng đồ án tốt nghiệp đề tài: “Xây dựng hệ thống bán đồ trang sức cho cửa hàng kira store trên nền tảng ASP.NET Core Web API và ReactJS” là kết quả nghiên cứu và làm việc độc lập của em.

Em cũng xin cam kết rằng không có sự sao chép hay làm dụng tài nguyên từ nguồn nào khác mà không được chỉ rõ. Mọi tài liệu tham khảo của bên thứ ba đều được em trích dẫn đầy đủ.

Trong quá trình xây dựng trang web, em đã tự tin và chân thành với sự hiểu biết và kỹ năng của mình. Khi thử sức với những điều mới trong học tập thì mọi quyết định và lựa chọn thiết kế đều dựa trên kiến thức và kinh nghiệm của em trong lĩnh vực này.

Hy vọng rằng trang web này sẽ mang lại giá trị và tiện ích cho người sử dụng và là một bước tiến quan trọng trong sự phát triển cá nhân và chuyên môn của em.

Ngày      , tháng      năm 2024

Sinh viên ký tên

## LỜI CẢM ƠN

Sau thời gian học tập tại trường, sinh viên được hệ thống lại toàn bộ lý thuyết chuyên ngành và được tham gia kiến tập một số khâu nghiệp vụ cơ bản của các kiến thức cơ bản của các kiến thức đã được học. Được sự cho phép của khoa Công nghệ thông tin Trường đại học Đại Nam, được sự quan tâm của, chỉ đạo của quý thầy cô trong khoa Công nghệ thông tin và các thầy cô bộ môn trong Trường Đại học Đại Nam, em đã bắt đầu quá trình thực hiện đồ án tốt nghiệp của mình. Khoảng thời gian làm đồ án tuy ngắn ngủi nhưng em đã học hỏi, có được những trải nghiệm từ những yêu cầu, thách thức trong quá trình làm. Thời gian này đã cho em những bài học kinh nghiệm quý báu, những kỹ năng cần thiết về ngành Công nghệ thông tin để em tự tin bước vào môi trường làm việc sau này.

Vì bài đồ án được thực hiện trong phạm vi thời gian ngắn và hạn chế về mặt kiến thức chuyên môn, thế nên bài báo cáo của em không thể tránh khỏi những sai sót. Đồng thời bản thân bài báo cáo là kết quả của quá trình tổng hợp và thu thập từ việc tham gia xây dựng một hệ thống website thực tế, đồng thời phản ánh những bài học mà em đã rút ra trong quá trình học tập và làm việc. Em rất mong có được những ý kiến đóng góp của thầy, cô để bài báo cáo và bản thân em hoàn thiện hơn.

Qua bài báo cáo này, em xin cảm ơn thầy ThS. Lê Trung Hiếu - giảng viên khoa Công nghệ thông tin, trường Đại học Đại Nam đã giúp đỡ và chỉ dẫn tận tình, tạo điều kiện để em hoàn thành học phần đồ án tốt nghiệp của mình.

Em xin chân thành cảm ơn.

## DANH MỤC KÝ HIỆU VÀ TỪ NGỮ VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết tắt đầy đủ
1	API	Application Programming Interface
2	ASP.NET	Active Server Pages .NET
3	CSDL	Cơ sở dữ liệu
4	CSS	Cascading Style Sheets
5	DOM	Document Object Model
6	HTTP	HyperText Transfer Protocol
7	IDE	Integrated Development Environment
8	MVC	Model-View-Controller
9	RDBMS	Relational Database Management System
10	REST	Representational State Transfer
11	SSAS	SQL Server Analysis Services
12	SSIS	SQL Server Integration Services
13	SSRS	SQL Server Reporting Services
14	UI	User Interface
15	Web API	Application Programming Interface

## DANH MỤC HÌNH ẢNH

Hình 1.1. Sơ đồ mô tả bài toán ( Trang người dùng) .....	7
Hình 1.2. Sơ đồ mô tả bài toán (Trang quản trị).....	8
Hình 2.1. Biểu đồ Database Diagram.....	20
Hình 2.2. Usecase Tổng quát hệ thống.....	21
Hình 2.3. Use case trang khách hàng .....	22
Hình 2.4. Biểu đồ phân rã của use case Xem sản phẩm.....	23
Hình 2.5. Biểu đồ phân rã của use case Quản lý đơn hàng .....	23
Hình 2.6. Biểu đồ phân rã use case Quản lý sản phẩm .....	24
Hình 2.7. Biểu đồ phân rã use case Quản lý thông tin nhà cung cấp. ....	24
Hình 2.8. Biểu đồ use case phân rã của use case Quản lý danh mục sản phẩm.....	25
Hình 2.9. Biểu đồ use case phân rã Quản lý thông tin khách hàng .....	25
Hình 2.10. Biểu đồ phân rã use case Quản lý nhân viên.....	26
Hình 2.11. Biểu đồ phân rã use case Quản lý giỏ hàng.....	26
Hình 2.12. Biểu đồ lớp thực thể .....	27
Hình 2.13. Biểu đồ VOPC use case Đăng nhập .....	28
Hình 2.14. Biểu đồ VOPC use case Quản lý sản phẩm.....	29
Hình 2.15. Biểu đồ lớp VOPC Quản lý danh mục .....	29
Hình 2.16. Biểu đồ tuần tự của use case Đăng nhập .....	30
Hình 2.17. Biểu đồ tuần tự use case Đăng ký .....	30
Hình 2.18. Biểu đồ tuần tự use case Đặt hàng.....	31
Hình 2.19. Biểu đồ tuần tự use case Xem sản phẩm .....	31
Hình 2.20. Biểu đồ tuần tự use case Thêm sản phẩm.....	32
Hình 2.21. Biểu đồ tuần tự use case Xóa sản phẩm. ....	32
Hình 2.22. Biểu đồ tuần tự use case Sửa thông tin sản phẩm .....	33
Hình 2.23. Biểu đồ trạng thái của lớp User .....	33
Hình 2.24. Biểu đồ trạng thái lớp Đăng nhập.....	34
Hình 2.25. Biểu đồ trạng thái lớp Tìm kiếm .....	34
Hình 2.26. Biểu đồ trạng thái của lớp Thêm sản phẩm.....	35
Hình 2.27. Biểu đồ tuần tự use case Quản lý nhân viên.....	36

Hình 2.28. Biểu đồ tuần tự use case Tìm kiếm sản phẩm .....	36
Hình 2.29. Biểu đồ tuần tự use case Đăng nhập.....	37
Hình 2.30. Biểu đồ tuần tự Xem sản phẩm (theo loại/ danh mục).....	37
Hình 3.1. Truy cập Google Dialogflow .....	41
Hình 3.2: Tạo Agent .....	42
Hình 3.3. Tạo Intents .....	42
Hình 3.4. Sinh mã để tích hợp .....	43
Hình 3.5. Giao diện trang chủ .....	43
Hình 3.6. Trang sản phẩm.....	44
Hình 3.7. Kết quả tìm kiếm .....	44
Hình 3.8. Giao diện đặt hàng.....	45
Hình 3.9. Giao diện chat với trợ lý AI.....	46
Hình 3.10. Giao diện Đăng nhập .....	46
Hình 3.11. Giao diện đăng ký tài khoản .....	47
Hình 3.12. Giao diện Menu điều khiển của Admin.....	47
Hình 3.13. Giao diện Quản lý sản phẩm .....	48
Hình 3.14. Giao diện Danh sách loại sản phẩm .....	48
Hình 3.15. Giao diện Quản lý nhân viên.....	49
Hình 3.16. Giao diện Quản lý thông tin khách hàng .....	49
Hình 3.17. Giao diện Quản lý thông tin nhà cung cấp .....	50
Hình 3.18. Giao diện quản lý đơn hàng.....	50
Hình 3.19. Giao diện Thống kê báo cáo .....	51

## **DANH MỤC BẢNG BIỂU**

Bảng 2.1. Bảng User (Người dùng).....	18
Bảng 2.2. Bảng Product (Sản phẩm) .....	18
Bảng 2.3. Bảng Category (Bảng Danh mục).....	19
Bảng 2.4. Bảng Supplier (Bảng nhà cung cấp) .....	19
Bảng 2.5. Bảng OrderDetail (Bảng chi tiết đơn đặt hàng) .....	19
Bảng 2.6. Bảng Order (Bảng Đặt hàng) .....	20



## MỤC LỤC

MỞ ĐẦU .....	1
1. Giới thiệu đề tài .....	1
2. Lí do chọn đề tài .....	1
3. Mục đích nghiên cứu .....	2
4. Phương pháp nghiên cứu .....	3
5. Phạm vi nghiên cứu .....	3
6. Kết cấu của đồ án.....	4
Chương 1 YÊU CẦU BÀI TOÁN.....	5
1.1. Yêu cầu bài toán .....	5
1.1.1. Yêu cầu chức năng.....	5
1.1.2. Yêu cầu phi chức năng .....	5
1.2. Mô tả bài toán và phân tích yêu cầu .....	6
1.2.1. Mô tả bài toán.....	6
1.2.2. Phân tích yêu cầu.....	8
1.3. Công nghệ sử dụng .....	9
1.3.1. ReactJS .....	9
1.3.2. TypeScript.....	11
1.3.3. Tailwind CSS .....	12
1.3.4. ASP.NET Web API .....	12
1.3.5. SQL Server .....	14
1.3.6. DialogFlow .....	16
Tiểu kết .....	17
Chương 2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	18
2.1. Thiết kế và xây dựng CSDL .....	18
2.1.1. Xác định thực thể và quan hệ .....	18
2.2.2. Biểu đồ Database Diagram .....	20

2.2. Thiết kế hệ thống .....	21
2.2.1. Các biểu đồ usecase (Use Case Diagram) .....	21
2.2.2. Biểu đồ lớp thực thể (Class Diagram) .....	27
2.2.3. Biểu đồ lớp phân tích (VOPC) .....	28
2.2.4. Biểu đồ tuần tự (Sequence Diagram) .....	30
2.2.5. Biểu đồ trạng thái (State Machine Diagram).....	33
2.2.6. Biểu đồ hoạt động (Activity Diagram).....	36
Tiểu kết .....	38
Chương 3 XÂY DỰNG, TRIỂN KHAI VÀ TÍCH HỢP .....	39
3.1. Xây dựng và triển khai .....	39
3.1.1. FrontEnd .....	39
3.1.2. BackEnd .....	40
3.1.3. Tích hợp.....	41
3.2. Kết quả xây dựng giao diện người dùng .....	43
3.2.1. Trang chủ .....	43
3.2.2. Trang sản phẩm.....	44
3.2.3. Trang tìm kiếm sản phẩm .....	44
3.2.4. Trang đặt hàng .....	45
3.2.5. Chat với trợ lý ảo (AI).....	45
3.2.6. Đăng nhập.....	46
3.2.7. Đăng ký tài khoản.....	47
3.2.8. Menu điều khiển của Admin.....	47
3.2.9. Quản lý sản phẩm .....	48
3.2.13. Quản lý danh mục sản phẩm .....	48
3.2.17. Quản lý nhân viên.....	49
3.2.21. Quản lý thông tin khách hàng.....	49
3.2.22. Quản lý thông tin nhà cung cấp.....	50

3.2.23. Quản lý đơn hàng .....	50
3.2.24. Thống kê báo cáo.....	51
3.3. Kết quả xây dựng các chức năng hệ thống.....	51
Tiểu kết .....	51
KẾT LUẬN .....	53

## MỞ ĐẦU

### 1. Giới thiệu đề tài

Trong thời đại ngày nay, sự phát triển nhanh chóng của công nghệ thông tin đã ảnh hưởng rất lớn đến lĩnh vực kinh doanh. Để đáp ứng những yêu cầu ngày càng cao về chất lượng kinh doanh, việc xây dựng một hệ thống hiệu quả là không thể phủ nhận. Điều này đặt ra nhu cầu cần thiết về sự kết hợp giữa công nghệ thông tin và kinh doanh để tạo nên hiệu quả tốt cho chủ doanh nghiệp và khách hàng.

Đề án "Xây dựng hệ thống bán đồ trang sức cho cửa hàng Kira Store trên nền tảng ASP.NET Core Web API và ReactJS" ra đời nhằm đáp ứng những thách thức này. Đề án hướng đến việc xây dựng một hệ thống thông tin linh hoạt và trực quan, giúp quản lý sản phẩm và khách hàng một cách có hiệu quả nhất.

Đặc biệt, mục tiêu của đề án là tập trung vào việc phát triển một hệ thống bán hàng trực quan, làm tăng tính hiệu quả và tiện lợi trong quá trình ra quyết định. Bằng cách này, em hy vọng tạo ra một trang web mạnh mẽ để hỗ trợ quản lý việc kinh doanh trực tuyến đáp ứng được yêu cầu của người sử dụng dịch vụ này.

### 2. Lí do chọn đề tài

Đề tài "Xây dựng hệ thống bán đồ trang sức cho cửa hàng Kira Store trên nền tảng ASP.NET Core Web API và ReactJS" được chọn vì các lý do chính sau:

Trước tiên, nhu cầu của thị trường đối với ngành công nghiệp trang sức đang phát triển mạnh mẽ. Xa xỉ vốn là một cuộc chơi dài hạn, các thương hiệu phải tìm cách nắm bắt được các hành vi mua sắm hiện tại của người tiêu dùng. Họ đang đi theo xu hướng nào, muốn gì và thể hiện được gì của bản thân qua những lần mua sắm trực tuyến. Đồng thời các nhãn hàng cũng phải cung cấp cho họ những trải nghiệm mua sắm phù hợp và được cá nhân hoá [1]. Vậy nên việc xây dựng một hệ thống bán hàng trực tuyến chuyên nghiệp sẽ giúp cho cửa hàng Kira Store mở rộng được khả năng tiếp cận khách hàng, không còn bị giới hạn bởi vị trí địa lý. Điều này không chỉ gia tăng doanh số bán hàng mà còn nâng cao nhận diện thương hiệu của cửa hàng.

Thứ hai, hệ thống này sẽ giúp tăng hiệu quả quản lý. Thay vì quản lý thủ công, việc sử dụng hệ thống tự động sẽ giúp cửa hàng quản lý nhân viên, đơn hàng, khách hàng và các báo

cáo thống kê doanh số một cách chính xác và hiệu quả hơn. Điều này giúp giảm thiểu sai sót và tiết kiệm thời gian, đồng thời cung cấp dữ liệu kịp thời và chính xác để hỗ trợ ra quyết định.

Thứ 3, sự thuận lợi của công nghệ. ASP.NET Core và ReactJS là hai công nghệ mạnh mẽ và phổ biến trong phát triển ứng dụng web hiện nay. Sử dụng các công nghệ này đảm bảo hệ thống có hiệu suất cao, bảo mật tốt và dễ dàng mở rộng trong tương lai. ASP.NET Core Web API cung cấp khả năng xây dựng các dịch vụ web mạnh mẽ, trong khi ReactJS mang lại trải nghiệm người dùng mượt mà và tương tác cao. Việc sử dụng công nghệ và Internet để kinh doanh trang sức mang lại nhiều lợi ích, bao gồm tiếp cận dễ dàng đến người tiêu dùng, quản lý hàng hóa hiệu quả, và tạo ra một trải nghiệm mua sắm thuận tiện và an toàn cho khách hàng.

Cuối cùng, một trải nghiệm người dùng tốt là yếu tố quyết định thành công của một hệ thống bán hàng trực tuyến. ReactJS với khả năng tạo ra các ứng dụng web có giao diện tương tác cao và phản hồi nhanh sẽ giúp thu hút và giữ chân khách hàng. Một giao diện người dùng thân thiện không chỉ nâng cao trải nghiệm mua sắm mà còn khuyến khích khách hàng quay lại mua sắm nhiều lần.

Với những lý do trên, việc lựa chọn đề tài này có thể mang lại nhiều cơ hội và lợi ích kinh doanh đáng kể cho các nhà nghiên cứu và doanh nghiệp quan tâm đến lĩnh vực kinh doanh trang sức.

### **3. Mục đích nghiên cứu**

Đề tài "Xây dựng hệ thống bán đồ trang sức cho cửa hàng Kira Store trên nền tảng ASP.NET Core Web API và ReactJS" nhằm giải quyết một cách cụ thể và hiệu quả các vấn đề mà Kira Store đang gặp phải trong việc kinh doanh trực tuyến. Các mục tiêu chính của đề tài bao gồm:

- Tạo ra một nền tảng trực tuyến cho Kira Store, giúp mở rộng phạm vi tiếp cận khách hàng và tăng doanh số. Hệ thống sẽ cho phép khách hàng dễ dàng truy cập, tìm kiếm và mua sắm các sản phẩm trang sức một cách thuận tiện.
- Sử dụng ASP.NET Core Web API để xây dựng phần backend của hệ thống, đảm bảo hiệu suất cao và khả năng mở rộng. Phần frontend sẽ được xây dựng bằng ReactJS, cung cấp một giao diện người dùng tương tác và thân thiện. Sự kết hợp này giúp tạo ra một hệ thống mạnh mẽ, linh hoạt và dễ bảo trì.

- Giao diện người dùng được thiết kế thân thiện, dễ sử dụng, giúp khách hàng dễ dàng tìm kiếm và chọn lựa sản phẩm. Hệ thống sẽ tích hợp các tính năng như tìm kiếm thông minh, gợi ý sản phẩm, và chat với trợ lý ảo AI, mang lại trải nghiệm mua sắm tốt nhất cho khách hàng.

#### 4. Phương pháp nghiên cứu

Đề tài "Xây dựng hệ thống bán đồ trang sức cho cửa hàng Kira Store trên nền tảng ASP.NET Core Web API và ReactJS" nhằm giải quyết các thách thức mà Kira Store đang gặp phải trong việc phát triển doanh nghiệp trực tuyến. Phương pháp nghiên cứu được thiết kế để cung cấp một hệ thống hiệu quả, linh hoạt và dễ bảo trì để tăng cường phạm vi tiếp cận khách hàng và doanh số bán hàng.

- *Nghiên cứu yêu cầu và vấn đề của Kira Store*: tiến hành cuộc trao đổi và phân tích yêu cầu từ phía Kira Store để hiểu rõ các thách thức và yêu cầu cụ thể của họ trong kinh doanh trực tuyến.
- *Xây dựng yêu cầu và phân tích hệ thống*: dựa trên thông tin thu thập được, xác định yêu cầu cụ thể cho hệ thống và phân tích các chức năng cần thiết để giải quyết vấn đề của Kira Store.
- *Lựa chọn công nghệ*: chọn ASP.NET Core Web API và ReactJS làm nền tảng để xây dựng hệ thống, dựa trên yêu cầu và các yếu tố như hiệu suất và dễ bảo trì.
- *Phát triển Backend và Frontend* : phát triển phần backend sử dụng ASP.NET Core Web API và phần frontend sử dụng ReactJS, đảm bảo tính tương thích và tương tác giữa hai phần của hệ thống.

#### 5. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng một hệ thống bán hàng trực tuyến toàn diện và hiệu quả cho Kira Store, từ việc phát triển cơ sở hạ tầng cho đến việc trải nghiệm khách hàng. Phạm vi bao gồm các khía cạnh:

- *Xây dựng nền tảng trực tuyến cho Kira Store*: tạo ra một hệ thống trực tuyến hoàn chỉnh để Kira Store có thể mở rộng phạm vi tiếp cận khách hàng và tăng doanh số bán hàng. Đồng thời phát triển một giao diện trực tuyến dễ sử dụng và thân thiện, giúp khách hàng dễ dàng truy cập, tìm kiếm và mua sắm các sản phẩm trang sức.

- Sử dụng công nghệ ASP.NET Core Web API và ReactJS.
- Tích hợp các tính năng tiên tiến: Thiết kế giao diện người dùng thân thiện và dễ sử dụng, tích hợp các tính năng như tìm kiếm thông minh, lọc sản phẩm và chat với trợ lý ảo AI để cải thiện trải nghiệm mua sắm. Phát triển tính năng tương tác giữa khách hàng và hệ thống thông qua các công nghệ mới như trợ lý ảo AI để hỗ trợ khách hàng tốt hơn trong quá trình mua sắm.

## **6. Kết cấu của đồ án**

Đồ án này gồm có 5 phần:

- Phần 1 – Mở đầu: Tổng quan về đề tài. Phần này sẽ chủ yếu tập trung vào giới thiệu các vấn đề chung liên quan đến đề tài đồ án như lý do chọn đề tài, phương pháp nghiên cứu, phạm vi nghiên cứu.
- Phần 2 – Chương 1: Yêu cầu bài toán. Chương này sẽ đi sâu phân tích cụ thể các yêu cầu đặt ra của bài toán, bao gồm các yêu cầu về chức năng và phi chức năng, giới thiệu về các công nghệ sử dụng trong đồ án.
- Phần 3 – Chương 2: Phân tích và thiết kế hệ thống. Quan trọng, chương này sẽ trình bày về các thiết kế cơ sở dữ liệu và thiết kế hệ thống.
- Phần 4 – Chương 3: Xây dựng, triển khai và tích hợp: Tập trung vào quá trình xây dựng hệ thống, bao gồm xây dựng CSDL, giao diện người dùng và các chức năng hệ thống.
- Phần 5 – Kết luận: Tóm tắt và đánh giá kết quả sản phẩm của đồ án, bao gồm những thành công cũng như là thách thức, khó khăn đã gặp phải trong quá trình nghiên cứu và triển khai đồ án.

## Chương 1

### YÊU CẦU BÀI TOÁN

#### 1.1. Yêu cầu bài toán

Yêu cầu chức năng và phi chức năng được mô tả trên đặt ra cơ sở cho việc xây dựng hệ thống website trực quan hiệu quả và linh hoạt, đáp ứng đầy đủ nhu cầu của đồ án xây dựng website kinh doanh trang sức. Điều này sẽ tạo ra một công cụ mạnh mẽ để theo dõi và quản lý thông tin với sự thuận lợi và hiệu quả.

##### 1.1.1. Yêu cầu chức năng

Website kinh doanh trang sức cần phải đáp ứng một loạt các yêu cầu chức năng để cung cấp trải nghiệm mua sắm trực tuyến thuận tiện và hấp dẫn cho người dùng. Điều này bao gồm việc cung cấp thông tin chi tiết và hình ảnh chất lượng cao về sản phẩm, cho phép người dùng tìm kiếm sản phẩm dễ dàng, quản lý giỏ hàng và đặt hàng an toàn. Ngoài ra, website cũng cần tối ưu hóa công cụ tìm kiếm để thu hút và giữ chân khách hàng. Bằng cách này, website không chỉ giúp doanh nghiệp tiếp cận khách hàng một cách hiệu quả mà còn tạo ra một môi trường mua sắm trực tuyến an toàn và đáng tin cậy cho người tiêu dùng.

Các yêu cầu chức năng cho website như sau:

Trang chủ: Hiển thị sản phẩm nổi bật và thông tin về thương hiệu để thu hút sự chú ý của người dùng ngay từ lần đầu tiên truy cập vào website.

Trang danh mục sản phẩm: Liệt kê các danh mục sản phẩm theo tiêu chí như loại trang sức để giúp người dùng dễ dàng tìm kiếm, truy cập và chọn mua sản phẩm.

Trang sản phẩm: Cung cấp thông tin chi tiết và hình ảnh chất lượng cao về từng sản phẩm, bao gồm giá cả, mô tả sản phẩm, thương hiệu...

Giỏ hàng: Cho phép người dùng xem và chỉnh sửa giỏ hàng của họ trước khi tiến hành đặt hàng, bao gồm thêm/xóa sản phẩm, điều chỉnh số lượng và tính được tổng số tiền.

Tìm kiếm: Tích hợp công cụ tìm kiếm mạnh mẽ, cho phép người dùng tìm kiếm sản phẩm theo từ khóa, các thuộc tính khác.

##### 1.1.2. Yêu cầu phi chức năng

Các yêu cầu phi chức năng giúp hệ thống hoạt động dễ dàng và trở nên thuận tiện hơn. Do đó, hệ thống cần có các yêu cầu phi chức năng sau:



Tốc độ tải trang nhanh: Tối ưu hóa website để tải trang nhanh chóng, giảm thiểu thời gian chờ đợi và cải thiện trải nghiệm người dùng.

Đáp ứng thiết bị di động: Thiết kế website sao cho hoạt động mượt mà trên các thiết bị di động như điện thoại thông minh và máy tính bảng, giúp người dùng tiện lợi khi truy cập từ mọi nơi.

Bảo mật thông tin người dùng: Đảm bảo bảo mật thông tin cá nhân và tài khoản của người dùng thông qua việc sử dụng mã hóa dữ liệu và các biện pháp bảo mật phù hợp.

Đễ dàng tương tác với người dùng: Cung cấp giao diện người dùng thân thiện, dễ sử dụng và dễ hiểu, giúp người dùng tương tác và thực hiện các thao tác một cách dễ dàng.

## **1.2. Mô tả bài toán và phân tích yêu cầu**

### *1.2.1. Mô tả bài toán*

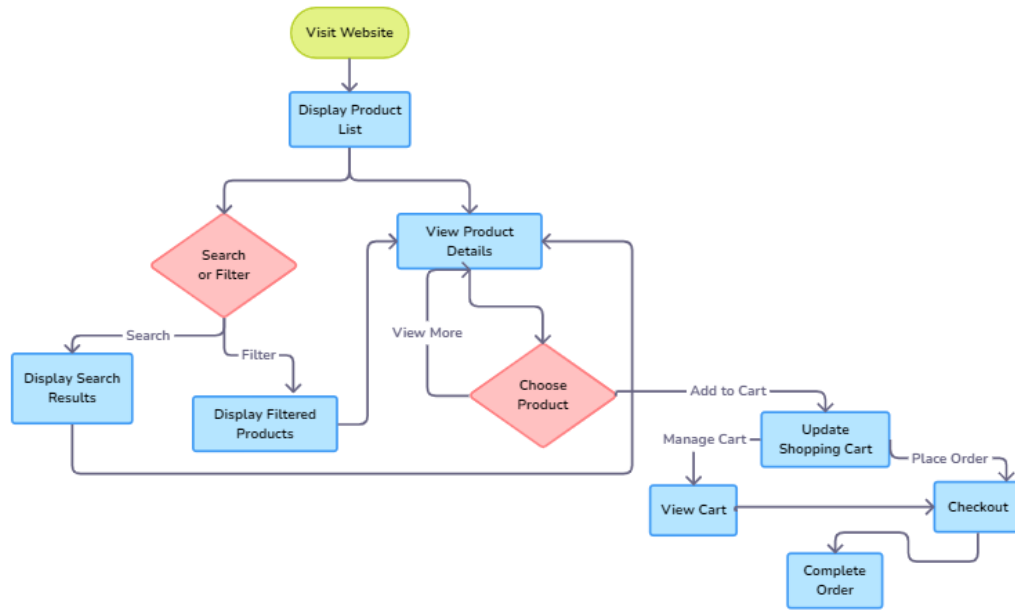
Khi người dùng (Mã người dùng, tên người dùng, giới tính, địa chỉ, số điện thoại, email, role, mật khẩu) truy cập vào trang web, trang web sẽ hiển thị danh sách các sản phẩm (Mã sản phẩm, tên sản phẩm, giá sản phẩm, mô tả sản phẩm, hình ảnh minh họa) và người dùng có thể tìm kiếm, lọc theo danh mục loại sản phẩm (Mã loại sản phẩm, tên loại sản phẩm) hoặc xem chi tiết các sản phẩm đó.

Người dùng có thể lựa chọn các sản phẩm yêu thích hoặc muốn mua (Mã đơn hàng, số lượng, đơn giá, mã giỏ hàng, mã sản phẩm) vào giỏ hàng (Mã giỏ hàng, ngày đặt hàng, tổng thành tiền). Khách hàng có thể quản lý giỏ hàng (tăng giảm số lượng sản phẩm có trong giỏ hàng) và đặt hàng.

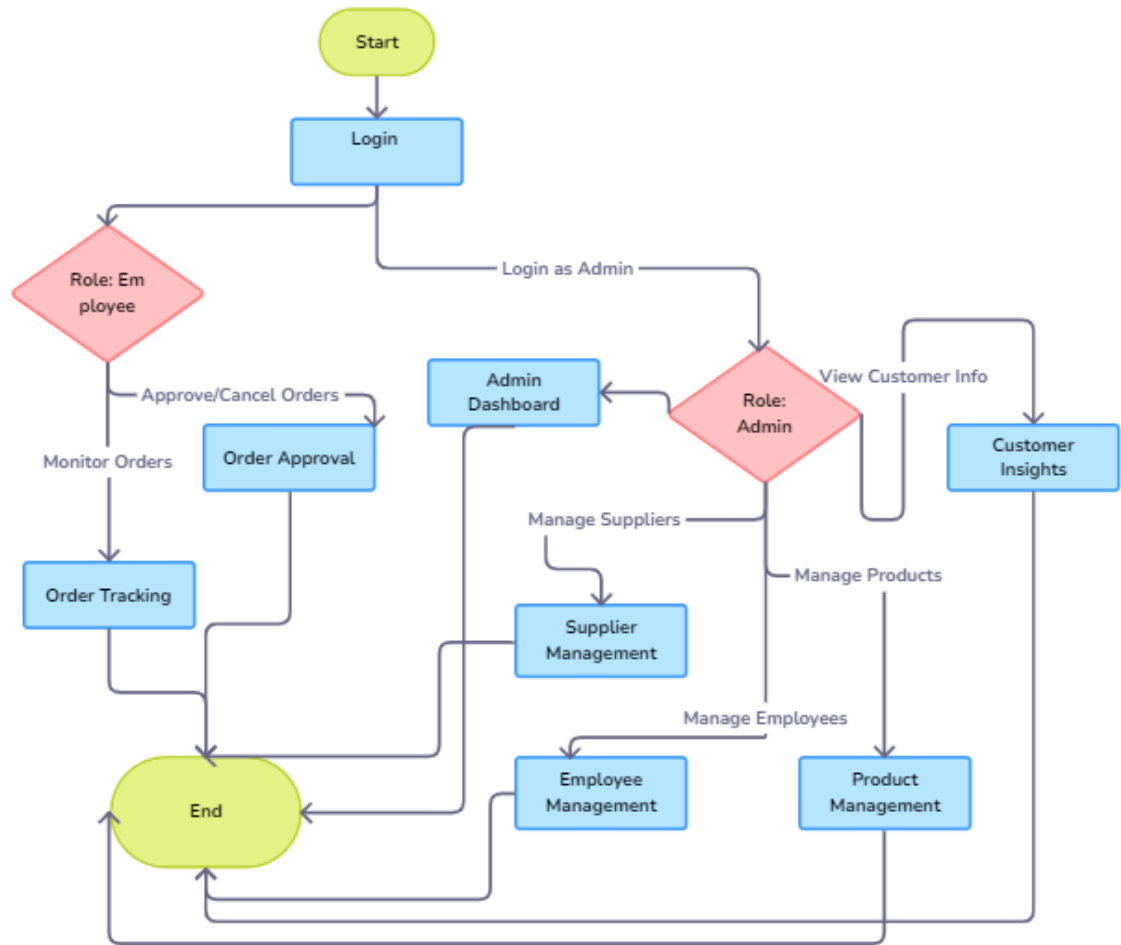
Sau khi đăng nhập tài khoản nhân viên, nhân viên có quyền quản lý (thêm, sửa, xóa) sản phẩm. Nhân viên có quyền theo dõi được tình trạng đơn hàng đã đặt của khách hàng. Đồng thời thì nhân viên cũng sẽ là người kiểm duyệt đơn hàng (Huỷ đơn hoặc Duyệt đơn).

Còn khi đăng nhập tài khoản Admin (chủ cửa hàng), admin sẽ chuyển tới trang dashboard hệ thống về sản phẩm, doanh thu (có trực quan hoá dữ liệu – biểu đồ), nhân viên. Admin có quyền quản lý nhân viên (bao gồm tài khoản và thông tin cá nhân của nhân viên), quản lý thông tin nhà cung cấp, quản lý sản phẩm, xem một số thông tin cơ bản của khách hàng.

Ta có thể khái quát hoá lại mô tả bài toán bằng sơ đồ sau:



Hình 1. 1. Sơ đồ mô tả bài toán (Trang người dùng)



Hình 1. 2. Sơ đồ mô tả bài toán (Trang quản trị)

### 1.2.2. Phân tích yêu cầu

Sau khi nghiên cứu phân mô tả bài toán và phân tích các yêu cầu bài toán, từ đó rút ra được hệ thống cần có các yêu cầu sau:

#### Khách hàng

Đăng nhập  
 Đăng ký tài khoản  
 Xem danh sách sản phẩm theo các loại danh mục sản phẩm, ...  
 Tìm kiếm sản phẩm theo tên sản phẩm, loại sản phẩm  
 Quản lý giỏ hàng  
 Chat với trợ lý ảo

**Nhân viên**

Đăng nhập tài khoản nhân viên Quản lý đơn hàng (Duyệt đơn hoặc Huỷ đơn) Đăng xuất tài khoản
---

**Admin**

Quản lý thông tin nhân viên, đăng ký tài khoản nhân viên Quản lý sản phẩm Quản lý loại sản phẩm Quản lý nhà cung cấp sản phẩm Báo cáo thống kê doanh thu Quản lý thông tin khách hàng Quản lý đơn hàng Đăng xuất
---

**Sản phẩm**

Loại sản phẩm Nguồn cung cấp sản phẩm Số lượng sản phẩm tồn kho
---

**1.3. Công nghệ sử dụng***1.3.1. ReactJS*

ReactJS là một thư viện JavaScript declarative, hiệu quả và linh hoạt cho việc xây dựng giao diện người dùng. React cho phép bạn tạo những giao diện (UI) phức tạp từ những đoạn code nhỏ và độc lập. Những đoạn code này được gọi là “components”. ReactJS phù hợp với đa dạng các thể loại website khác nhau. Có lẽ vì thế mà nó được các nhà phát triển lựa chọn để xây dựng website.

Tất nhiên, React nó phải mang trong mình một số ưu điểm cần thiết để phát triển ứng dụng web trên các nền tảng ứng dụng khác nhau. Một số lợi ích có thể kể đến như:

Hiệu suất cao: React được thiết kế để tối ưu hóa hiệu suất bằng cách sử dụng virtual DOM, giúp giảm thiểu số lượng các thao tác DOM thực sự cần thực hiện, từ đó cải thiện tốc độ và hiệu suất của ứng dụng.

Tính tái sử dụng cao: React khuyến khích việc sử dụng các thành phần tái sử dụng, giúp giảm thiểu việc viết mã lặp lại và tăng khả năng tái sử dụng mã nguồn.

Hỗ trợ cộng đồng mạnh mẽ: React là một dự án mã nguồn mở được phát triển và duy trì bởi Facebook và cộng đồng lập trình viên toàn cầu. Sự hỗ trợ từ cộng đồng có thể giúp công ty chứng khoán giải quyết các vấn đề nhanh chóng và hiệu quả.

Độ linh hoạt và mở rộng: React cung cấp một cách tiếp cận linh hoạt cho việc xây dựng ứng dụng, cho phép các nhà phát triển tích hợp React vào các dự án hiện có mà không cần phải tái thiết kế toàn bộ ứng dụng.

Cộng sinh với các công nghệ khác: React có thể dễ dàng kết hợp với các thư viện và framework khác như Redux để quản lý trạng thái ứng dụng, hoặc React Router để quản lý định tuyến.

Hỗ trợ cho ứng dụng đa nền tảng: React Native, một phiên bản của React, cho phép phát triển ứng dụng di động cho cả IOS và Android bằng cách sử dụng JavaScript, giúp giảm thiểu chi phí và thời gian phát triển cho các ứng dụng di động.

Nhưng bên cạnh đó thì ReactJS cũng mang trong mình một số nhược điểm:

- ReactJS chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác.
- Khi muốn tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
- React khá nặng nếu so với các framework khác React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.
- Khó tiếp cận cho người mới học web.

#### ❖ Cách Phát Triển Frontend với ReactJS

Phát triển giao diện người dùng bằng ReactJS bao gồm việc tạo các component, quản lý trạng thái (state management), và giao tiếp với API. Component là các khối xây dựng của giao diện React, giúp chia nhỏ giao diện thành các phần tử độc lập và có thể tái sử dụng. Việc quản lý trạng thái ứng dụng được thực hiện thông qua các công cụ như Redux hoặc Context API, giúp duy trì trạng thái nhất quán và dễ dàng quản lý. Giao tiếp với API được thực hiện thông qua các phương thức HTTP như GET, POST, PUT, DELETE, giúp gửi và nhận dữ liệu từ backend. ReactJS với các thư viện như Axios hoặc Fetch API hỗ trợ việc này, đảm bảo dữ liệu được xử lý và hiển thị một cách hiệu quả.

### 1.3.2. TypeScript

TypeScript là một ngôn ngữ lập trình được phát triển và duy trì bởi Microsoft. Nó là một siêu tập hợp cú pháp nghiêm ngặt của JavaScript và thêm tính năng kiểu tĩnh tùy chọn vào ngôn ngữ. TypeScript được thiết kế để phát triển các ứng dụng lớn và biên dịch sang JavaScript.

TypeScript mang trong mình một số ưu điểm như:

- Thuận tiện và hoàn toàn miễn phí: tất cả những số liệu, thông số của bạn sẽ dễ dàng được lấy hơn nhờ IDE và trình biên dịch. TypeScript hỗ trợ quá trình tìm kiếm giúp bạn tiết kiệm thời gian kiểm tra lại code, không cần thông qua bất cứ một ai để có thể tìm thông tin dữ liệu, ngoài ra TypeScript làm giảm phần trăm va chạm lỗi trong thời gian vận hành.
- Thao tác nhanh chóng và đơn giản: Ngôn ngữ TypeScript có thao tác khá đơn giản, tiết kiệm thời gian hơn nhưng lại đem đến kết quả tốt đến bất ngờ, nó khắc phục tình trạng xuất hiện lỗi và dễ đọc hơn.
- Giảm tỷ lệ mắc lỗi trong hệ thống: Nhờ vào việc cảnh báo lỗi ngay khi viết code, nên tỷ lệ mắc lỗi trong hệ thống khi sử dụng TypeScript khá thấp, TypeScript sẽ trả lại giá trị null hoặc gợi ý thay đổi chỉnh sửa. Mỗi lần chỉnh sửa sau khi được TypeScript báo lỗi thì phần trăm hệ thống hoạt động mà không mắc phải lỗi là rất cao, có thể dễ dàng thấy được TypeScript giúp người dùng tiết kiệm không ít thời gian để sửa lỗi.

Tuy nhiên, bất kỳ ngôn ngữ nào cũng có điểm yếu và hạn chế của nó. Typescript cũng vậy. Nó có một số điểm hạn chế như sau:

- Để có thể vận hành với một tệp TypeScript với đuôi .js trên nền tảng Node.js, người dùng bắt buộc phải dùng trình biên dịch để có thể sử dụng.
- Các bước để thiết lập khá công kềnh. Trước khi có thể sử dụng được TypeScript, bạn cần đảm bảo rằng máy chủ Node.js, trình thử nghiệm và webpack đều có thể hoạt động với TypeScript, nếu không thì bạn sẽ không sử dụng được.
- Bên cạnh đó, mỗi khi người apply thêm bất kỳ library nào như Redux, React và Styled-Component thì bạn cũng phải thêm Typedef vào.

### 1.3.3. Tailwind CSS

Tailwind CSS là một utility-first CSS framework nó hỗ trợ phát triển xây dựng nhanh chóng giao diện người dùng, nó cũng có điểm chung giống như Bootstrap & điểm làm nó nổi bật hơn cả đó là chúng ta có thể tùy biến phát triển css theo cách mà chúng ta định nghĩa ra.

Tailwind CSS có rất nhiều ưu điểm. Những ưu điểm nhìn thấy rõ nhất là:

- Người dùng có thể chẳng phải dùng đến 1 dòng css nào nhưng vẫn có thể ra được giao diện như mong muốn.
- Style, màu sắc, font chữ hiện đại phù hợp với phong cách hiện đại.
- Cách đặt tên các class rất dễ hiểu, mỗi 1 class đại diện cho một thuộc tính. Tailwind có gần như đủ 85% thuộc tính css.
- Sử dụng flex nên rất dễ chia layout.
- Dễ cài đặt, dễ sử dụng, document của Tailwind rất dễ hiểu. Tailwind CSS phù hợp cho các dự án nhỏ, người dùng tùy biến nhiều, cần làm nhanh giao diện.

Nhưng bên cạnh đó thì khi sử dụng tailwind thì chúng ta phải sử dụng số lượng class cực kỳ nhiều, số class sẽ tương ứng với số thuộc tính mà người dùng muốn cài đặt. Và đôi khi dùng font-size hay color vẫn còn phải custom lại bằng css riêng.

### 1.3.4. ASP.NET Web API

Web API hay ASP.NET Web API là một framework dùng để xây dựng hoặc lập trình các ứng dụng, dịch vụ web HTTP. Nó có dạng là một RESTfull API hiện đại, hội tụ các điều kiện của REST cũng như các tiêu chuẩn tương tự, được tối ưu cho các dịch vụ trực tuyến cũng

như ứng dụng web hiện nay. Được phát triển bởi Microsoft, Web API là một phần của bộ ASP.NET và cung cấp một cách dễ dàng và linh hoạt để xây dựng các API RESTful. Trong đó:

API: là một tập hợp các giao diện cho phép các ứng dụng giao tiếp với nhau. API định nghĩa cách thức mà các thành phần phần mềm tương tác.

REST: là một kiến trúc được sử dụng phổ biến để thiết kế các dịch vụ web. REST sử dụng các phương thức HTTP như GET, POST, PUT, DELETE để thực hiện các thao tác CRUD (Create, Read, Update, Delete).

Lợi ích khi sử dụng ASP.NET Web API:

- ASP.NET Web API có khả năng mở rộng để phục vụ nhiều loại ứng dụng khác nhau.
- Có thể xử lý số lượng lớn yêu cầu đồng thời, điều này rất quan trọng đối với dịch vụ cần phục vụ cho nhiều người cùng một lúc.
- Hỗ trợ thiết kế kiến trúc microservices, giúp phân tách các chức năng riêng lẻ, dễ dàng quản lý và mở rộng từng phần một cách độc lập.
- Kiến trúc RESTfull của web API giúp mã nguồn người đọc dễ bảo trì. Các thành phần như controller, model, routing,... được phân tách rõ ràng, giúp quản lý mã nguồn hiệu quả.
- Dễ dàng tích hợp tốt với các nền tảng công nghệ khác, tạo ra một hệ sinh thái linh hoạt và mạnh mẽ:

+ Front-End Frameworks: Tích hợp mượt mà với các framework front-end như Angular, React, và Vue.js thông qua các RESTful endpoints.

+ Mobile Applications: Dễ dàng phục vụ dữ liệu cho các ứng dụng di động (Android, iOS) thông qua các API endpoints.

+ Cloud Services: Tương thích tốt với các dịch vụ đám mây như Azure, AWS, giúp triển khai và quản lý các ứng dụng trên đám mây hiệu quả.

+ Databases: Tích hợp dễ dàng với nhiều hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, và NoSQL databases như MongoDB.



- ASP.NET Web API được tối ưu hóa để có hiệu suất cao. Hỗ trợ lập trình bất đồng bộ async/await giúp cải thiện hiệu suất và khả năng phản hồi của ứng dụng.

#### ❖ Cách phát triển API bằng ASP.NET Core

Trong quá trình phát triển API bằng ASP.NET Core, các bước chính bao gồm việc tạo các endpoint, xử lý dữ liệu, xác thực và phân quyền. Đầu tiên, việc thiết kế và xây dựng các endpoint cho phép giao tiếp giữa server và client. Các endpoint này được tạo ra dựa trên yêu cầu chức năng của hệ thống, như quản lý sản phẩm, giỏ hàng, và người dùng. Tiếp theo, xử lý dữ liệu là một bước quan trọng, bao gồm việc thao tác với cơ sở dữ liệu, thực hiện các truy vấn, và xử lý logic nghiệp vụ. Để đảm bảo an toàn, xác thực và phân quyền được triển khai nhằm kiểm soát quyền truy cập và bảo vệ dữ liệu người dùng. ASP.NET Core cung cấp các công cụ mạnh mẽ như Identity và JWT để thực hiện các nhiệm vụ này một cách hiệu quả.

#### 1.3.5. SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft từ năm 1989 và vẫn được sử dụng rộng rãi cho đến nay. Nó được thiết kế để xử lý và quản lý dữ liệu lưu trữ trên máy chủ, cung cấp các dịch vụ hỗ trợ cho việc truy xuất, cập nhật, và quản lý dữ liệu một cách hiệu quả.

Mục đích ban đầu của SQL server chính là dùng để lưu trữ và duy trì csdl. Lúc này chúng ta có thể hình dung rằng SQL server giống như một cái kho(hub) và các dữ liệu được thêm vào dồn dập trong cái kho đó. Nhưng dữ liệu ở đây không phải là được lưu trữ một cách bừa bãi và không theo một quy luật nào cả. Để tiện cho các truy vấn của người dùng khi tìm kiếm dữ liệu trong hub này, dữ liệu khi được đưa vào SQL server được lưu trữ một cách có chủ đích. Người dùng sẽ tận dụng các công cụ lưu trữ hiện có để phân loại và sắp xếp dữ liệu cho hợp lý[2].

Các thành phần chính của SQL Server:

- Database Engine là thành phần cốt lõi của SQL Server, chịu trách nhiệm xử lý các yêu cầu truy vấn dữ liệu, quản lý giao dịch và thực thi các thủ tục lưu trữ (stored procedures). Nó bao gồm:

- Relational Engine: Xử lý các yêu cầu truy vấn SQL, tối ưu hóa truy vấn và thực thi các lệnh.

- Storage Engine: Quản lý lưu trữ dữ liệu, xử lý các tác vụ liên quan đến đọc/ghi dữ liệu trên đĩa.

- SSMS là một công cụ tích hợp (IDE) dùng để quản lý, cấu hình, và phát triển các thành phần của SQL Server. Nó cung cấp giao diện đồ họa và các công cụ để tạo, sửa đổi, và quản lý cơ sở dữ liệu.

- SSIS là một công cụ ETL (Extract, Transform, Load) dùng để di chuyển và biến đổi dữ liệu giữa các hệ thống khác nhau. Nó cho phép tạo các quy trình xử lý dữ liệu phức tạp, bao gồm việc trích xuất dữ liệu từ nhiều nguồn, biến đổi dữ liệu và tải vào cơ sở dữ liệu đích.

- SSAS là một công cụ phân tích dữ liệu và xử lý phân tích trực tuyến (OLAP). Nó cho phép tạo các mô hình phân tích phức tạp, cung cấp khả năng xử lý dữ liệu lớn và tạo các báo cáo phân tích đa chiều.

SQL Server của Microsoft mang lại nhiều lợi ích cho các tổ chức và doanh nghiệp, giúp họ quản lý và khai thác dữ liệu một cách hiệu quả. Dưới đây là một số lợi ích chính khi sử dụng SQL Server:

- SQL Server được thiết kế để xử lý lượng dữ liệu lớn với tốc độ nhanh. Nó cung cấp các công cụ tối ưu hóa truy vấn, giúp tăng cường hiệu suất truy vấn và giảm thời gian phản hồi.

- Tính năng tối ưu hóa tự động của SQL Server giúp điều chỉnh hiệu suất hệ thống một cách liên tục mà không cần can thiệp thủ công nhiều từ người quản trị.

- Có thể mở rộng từ các hệ thống nhỏ đến các hệ thống lớn với hàng ngàn người dùng đồng thời, đáp ứng nhu cầu tăng trưởng của doanh nghiệp.

- SQL Server tích hợp dễ dàng với các sản phẩm khác của Microsoft như Azure, Power BI, và các công cụ Office, giúp tạo ra một hệ sinh thái hoàn chỉnh cho quản lý và phân tích dữ liệu.

- SQL Server cung cấp các tùy chọn sao lưu và phục hồi linh hoạt, bao gồm full backup, differential backup, và transaction log backup, giúp bảo vệ dữ liệu và đảm bảo khả năng phục hồi sau sự cố.
- Cung cấp giao diện đồ họa giúp quản lý cơ sở dữ liệu dễ dàng và hiệu quả, hỗ trợ cả quản lý, phát triển và gỡ lỗi.

### 1.3.6. DialogFlow

Dialogflow (tiền thân là API.AI) là một dịch vụ do Google cung cấp nhằm giúp các lập trình viên dễ dàng hơn khi lập trình các sản phẩm có giao tiếp giữa người dùng với sản phẩm thông qua các đoạn hội thoại bằng văn bản (text) hoặc giọng nói (voice) [3].

DialogFlow là một công cụ xây dựng ChatBot miễn phí và có những hiệu quả nhất định. Nhưng trước khi tìm hiểu về DialogFlow thì chúng ta cần tìm hiểu xem ChatBot là gì?

Chatbot là một chương trình hỗ trợ Trí tuệ nhân tạo có thể trả lời các câu hỏi tự động. Một số ứng dụng nhắn tin phổ biến mà bạn có thể đã biết bao gồm Facebook Messenger và WeChat. Mục đích chính là thu hút khách hàng và thu thập thông tin có giá trị [4]. Chúng ta cần có một con ChatBot để hỗ trợ chăm sóc khách hàng. Vì hầu hết các câu hỏi của khách hàng đều thuộc các câu hỏi dạng: bạn bán loại sản phẩm nào, cửa hàng bạn ở đâu?, cửa hàng có mở cửa vào cuối tuần hay không?, tôi có thể đặt hàng online hay không? .... Chúng đều là những câu hỏi rất dễ trả lời, nhưng sự lặp đi lặp lại của khối lượng câu hỏi đến từ khách hàng khiến cho nhân viên sẽ phải dành rất nhiều thời gian để xử lý. Điều này vừa tốn nhân lực, vừa tốn thời gian mà hiệu quả chưa chắc đã cao. Chính vì vậy chúng ta cần phải có một hệ thống chat thông minh để xử lý việc này. Và công cụ DialogFlow sẽ là công cụ cho chúng ta giải quyết vấn đề này.

Google Dialogflow là một nền tảng xử lý (NLP) ngôn ngữ tự nhiên tiên tiến bậc nhất cho phép các nhà phát triển phần mềm kinh doanh xây dựng các tình huống đàm thoại hoặc auto chatbot. Từ đó mà chúng hiểu và trả lời mọi truy vấn từ người dùng theo cách tự nhiên giống như con người [5]. Một số tính năng của Google Dialogflow có thể kể đến như:

- Hỗ trợ đa ngôn ngữ, tăng tương tác giao tiếp với các người dùng, đối tác nước ngoài.
- Phân tích dữ liệu về hiệu suất và tương tác khách hàng.

- Tích hợp đa kênh (chỉ cần xây trên 1 kênh, người dùng có thể triển khai trên các kênh kỹ thuật số và trung tâm liên lạc).
- Cung cấp thư viện agents dựng sẵn được chuẩn bị cho các trường hợp phổ biến nhằm giúp tối ưu hóa thời gian sản xuất.
- AI nâng cao (tận dụng các mô hình Machine Learning được phát triển bởi Google Research).
- Cung cấp những câu hỏi bổ sung (giải quyết những vấn đề trong cuộc đối thoại, sau đó trở lại luồng chính nhanh chóng).

Chính vì có những tính năng nổi trội ở trên mà Google Dialogflow sẽ đem lại cho chúng ta những lợi ích ít nhất định, đặc biệt là trong sự phát triển của nền kinh tế. Đối với các doanh nghiệp thuộc mọi quy mô, Google Dialogflow cung cấp bộ công cụ hấp dẫn để nâng cao trải nghiệm của khách hàng, hợp lý hóa hoạt động và thu được thông tin chi tiết có giá trị. Trong phạm vi của hệ thống bán hàng này thì sẽ áp dụng Chatbots dịch vụ khách hàng. Một chatbot có sự hỗ trợ của Dialogflow có thể xử lý các câu hỏi cơ bản, giúp con người xử lý các vấn đề phức tạp hơn. Chatbot có thể hiểu các câu hỏi về trạng thái đơn hàng, tính năng sản phẩm và thậm chí điều hướng người dùng đến các tài nguyên hữu ích.

## **Tiểu kết**

Chương 1 đã trình bày chi tiết về yêu cầu của bài toán và công nghệ sử dụng trong đồ án. Đầu tiên, chương đã nêu rõ các yêu cầu của bài toán, yêu cầu chức năng và phi chức năng nhằm xác định các tiêu chí cụ thể mà hệ thống cần đáp ứng. Tiếp theo, phần mô tả bài toán và phân tích yêu cầu đã cung cấp một cái nhìn khách quan về bài toán cần phải giải quyết, cùng với việc phân tích chi tiết từng yêu cầu. Điều này giúp xác định rõ ràng mục tiêu và phạm vi của đồ án, đảm bảo rằng mọi khía cạnh của bài toán đều được xem xét và xử lý. Cuối cùng, chương này đã liệt kê các công nghệ sẽ được sử dụng trong quá trình phát triển hệ thống, bao gồm ReactJS, TypeScript, Tailwind CSS, ASP.NET Web API, SQL Server, và DialogFlow. Việc chọn lựa các công nghệ này đảm bảo rằng hệ thống sẽ được xây dựng trên nền tảng hiện đại, đáp ứng tốt các yêu cầu đã đặt ra và đảm bảo hiệu quả trong quá trình phát triển.

## Chương 2

### PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

#### 2.1. Thiết kế và xây dựng CSDL

##### 2.1.1. Xác định thực thể và quan hệ

*Bảng 2.1. Bảng User (Người dùng)*

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PRIMARY KEY	Khóa chính
2	UserName	nvarchar(255)	NOT NULL	Tên người dùng
4	Address	nvarchar(255)	NOT NULL	Địa chỉ
5	Gender	nvarchar(50)	NOT NULL	Giới tính
6	Email	nvarchar(255)	NOT NULL	Email
7	Phone	nvarchar(25)	NOT NULL	Điện thoại
8	Role	nvarchar(255)	NOT NULL	Quyền
9	Password	nvarchar(255)	NOT NULL	Mật khẩu

*Bảng 2.2. Bảng Product (Sản phẩm)*

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PRIMARY KEY	Khóa chính
2	ProductName	nvarchar(255)	NOT NULL	Tên sản phẩm
3	CategoryId	int	NOT NULL	Khoá ngoại
4	SupplierId	int	NOT NULL	Khoá ngoại
5	ProductDescription	nvarchar(max)	ALLOW NULL	Mô tả sản phẩm
6	ProductImg	nvarchar(255)	ALLOW NULL	Ảnh sản phẩm
7	ProductQuantity	int	NOT NULL	Số lượng
8	ProductPrice	int	NOT NULL	Giá sản phẩm

*Bảng 2.3. Bảng Category (Bảng Danh mục)*

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PRIMARY KEY	Khóa chính
2	CategoryName	nvarchar(255)	NOT NULL	Tên danh mục
3	CategoryDescription	nvarchar(255)	NOT NULL	Mô tả danh mục
4	dateCreated	date	NOT NULL	Ngày tạo

*Bảng 2.4. Bảng Supplier (Bảng nhà cung cấp)*

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PRIMARY KEY	Khóa chính
2	SupplierName	nvarchar(255)	NOT NULL	Tên nhà cung cấp
3	Address	nvarchar(255)	NOT NULL	Địa chỉ
4	Phone	nvarchar(25)	NOT NULL	Số điện thoại
5	Email	nvarchar(255)	NOT NULL	Email
6	StartDate	date	NOT NULL	Ngày bắt đầu hợp tác
7	EndDate	date	NOT NULL	Ngày kết thúc hợp tác

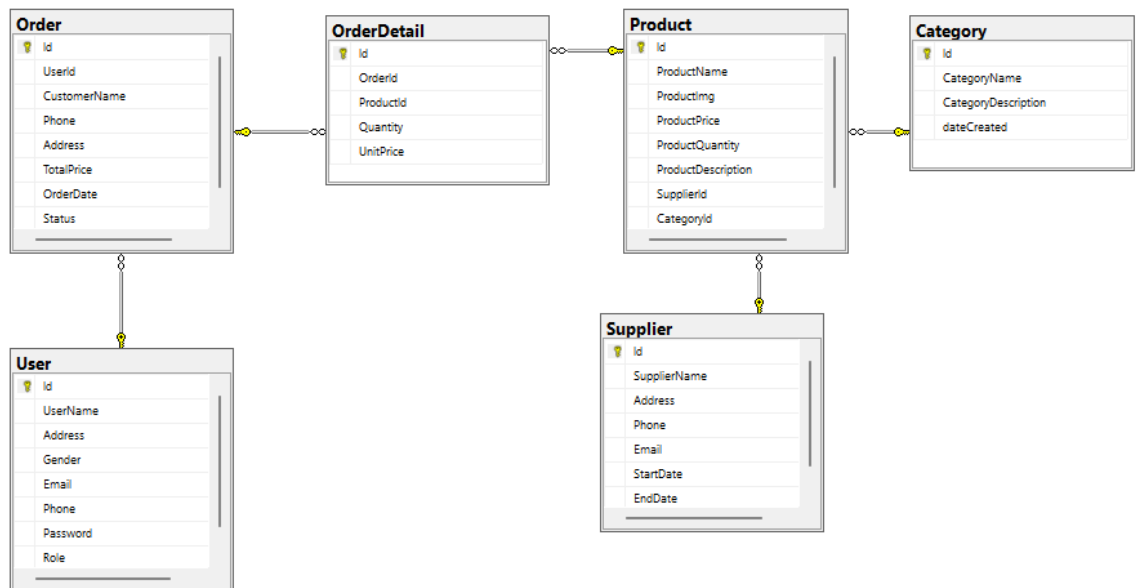
*Bảng 2.5. Bảng OrderDetail (Bảng chi tiết đơn đặt hàng)*

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PRIMARY KEY	Khóa chính
2	Quantity	int	NOT NULL	Số lượng
3	UnitPrice	int	NOT NULL	Đơn giá
4	OrderId	int	NOT NULL	Khoá ngoại
5	ProductId	int	NOT NULL	Khoá ngoại

Bảng 2.6. Bảng Order (Bảng Đặt hàng)

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PRIMARY KEY	Khóa chính
2	UserId	int	NOT NULL	Khoá ngoại
3	CustomerName	nvarchar(MAX)	NOT NULL	Tên khách hàng
4	Phone	nvarchar(25)	NOT NULL	Điện thoại đặt hàng
5	Address	nvarchar(255)	NOT NULL	Địa chỉ nhận
6	TotalPrice	float	NOT NULL	Tổng giá
7	OrderDate	nvarchar(30)	NOT NULL	Ngày đặt hàng
8	Status	nvarchar(255)	NOT NULL	Trạng thái đặt hàng

### 2.2.2. Biểu đồ Database Diagram

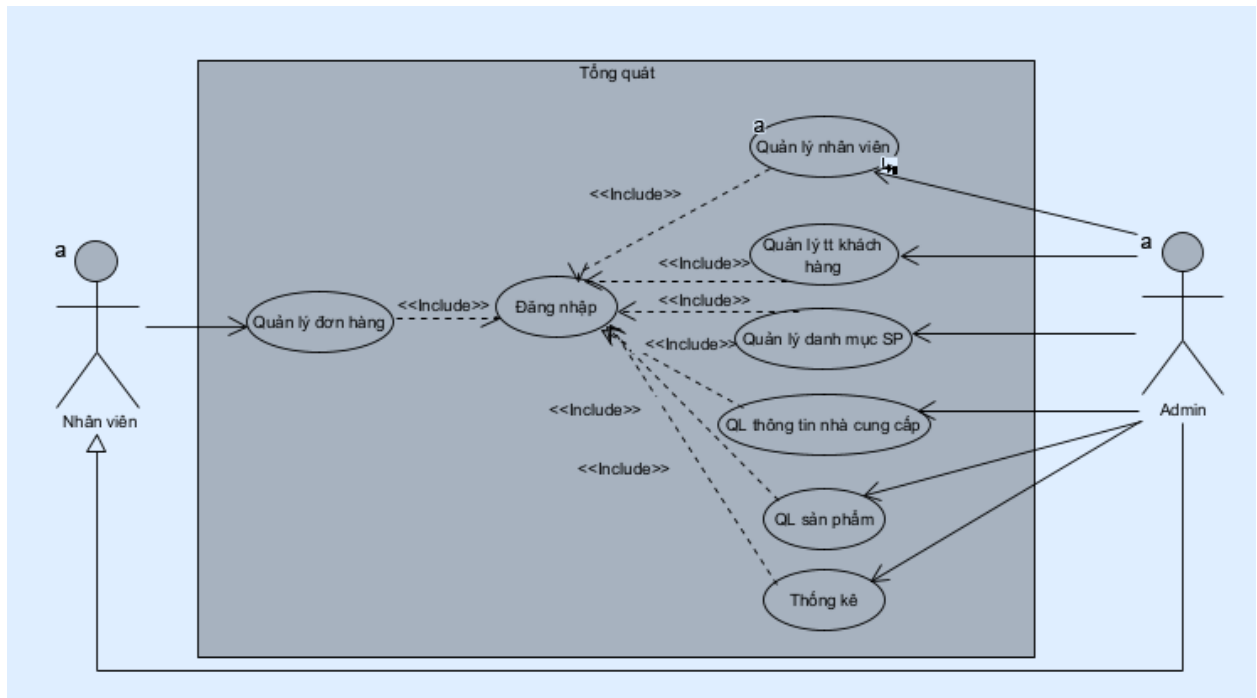


Hình 2.1. Biểu đồ Database Diagram

## 2.2. Thiết kế hệ thống

### 2.2.1. Các biểu đồ usecase (Use Case Diagram)

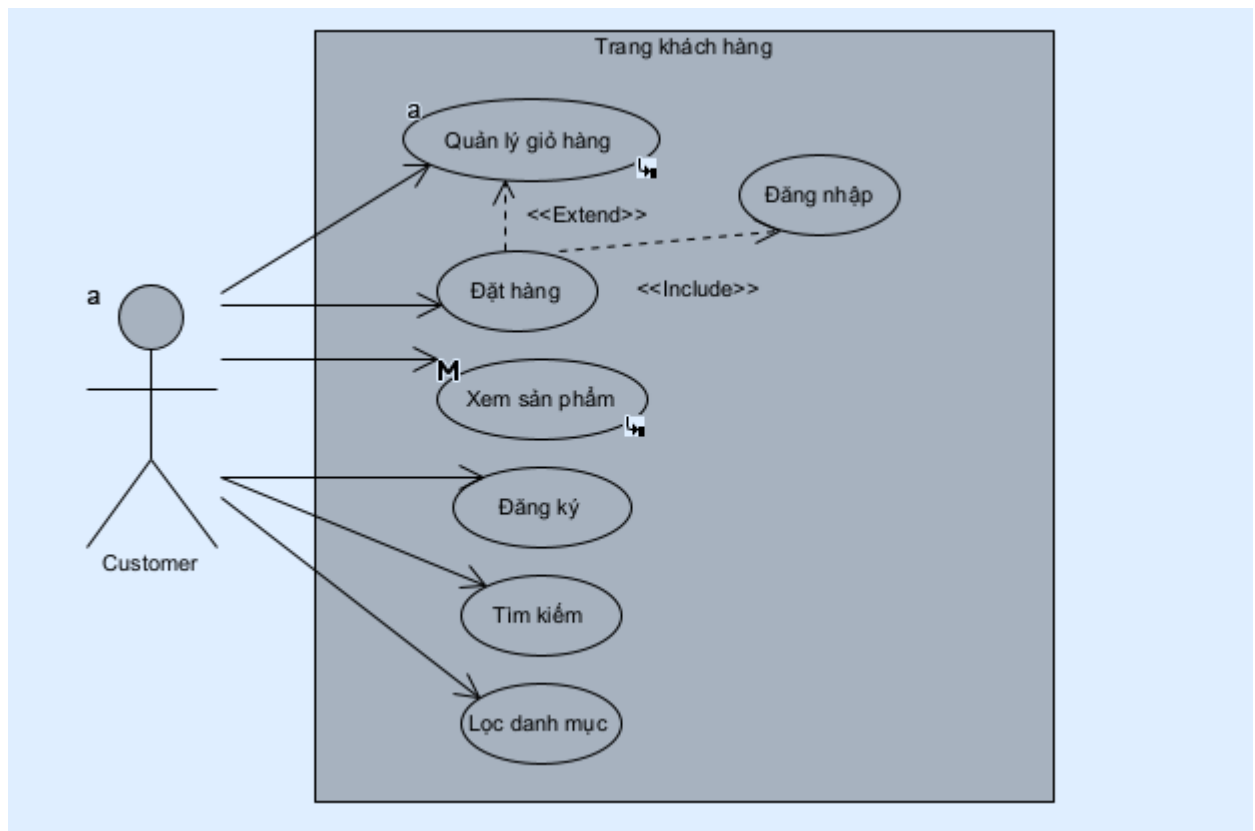
- Use Case Tổng quát. Hình 2.1 này cho người dùng thấy được tổng quan các chức năng của hệ thống. Hệ thống có sự phân quyền truy cập của Admin và nhân viên. Admin có chức năng quản lý mọi mặt của hệ thống và nhân viên có chức năng quản lý đơn hàng.



Hình 2.2. Usecase Tổng quát hệ thống

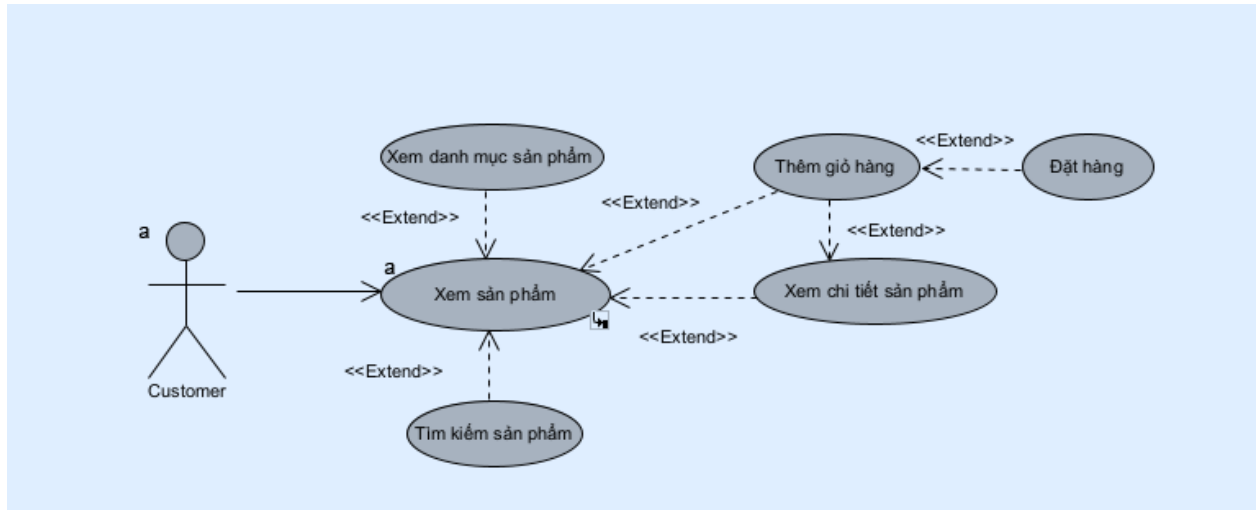


- Biểu đồ tổng quát trang khách hàng. Hình 2.3 là biểu đồ use case thể hiện khách hàng có thể thực hiện các chức năng chính như: đăng nhập, xem sản phẩm, đặt hàng,...+



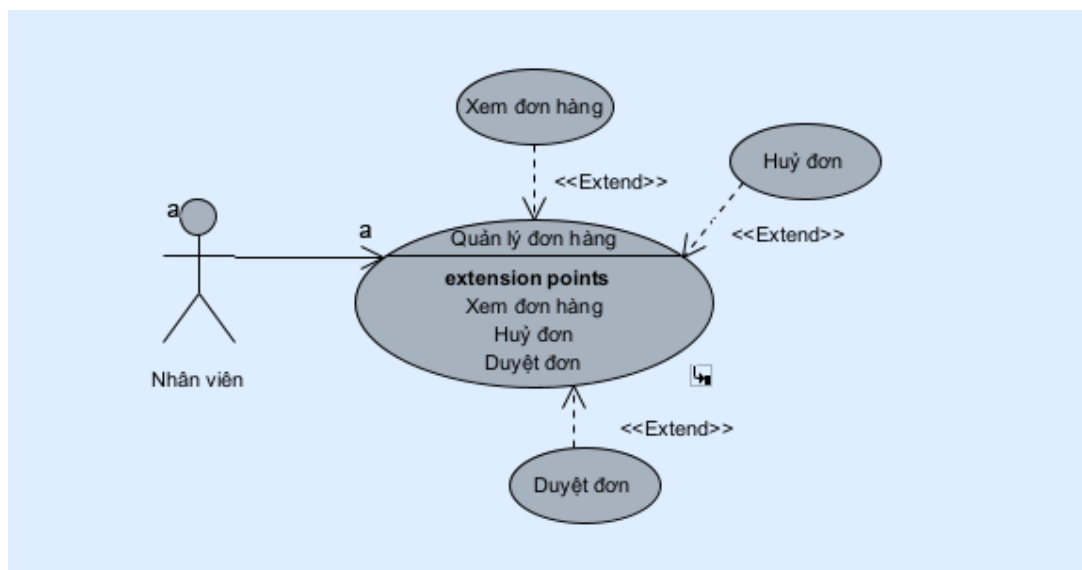
Hình 2.3. Use case trang khách hàng

- Biểu đồ phân rã của use case Xem sản phẩm. Hình 2.4 thể hiện biểu đồ phân rã của use case Xem sản phẩm của khách hàng. Biểu đồ sẽ gồm các use case phân rã: Tìm kiếm sản phẩm, Xem danh mục sản phẩm, Thêm giỏ hàng, Đặt hàng, ...



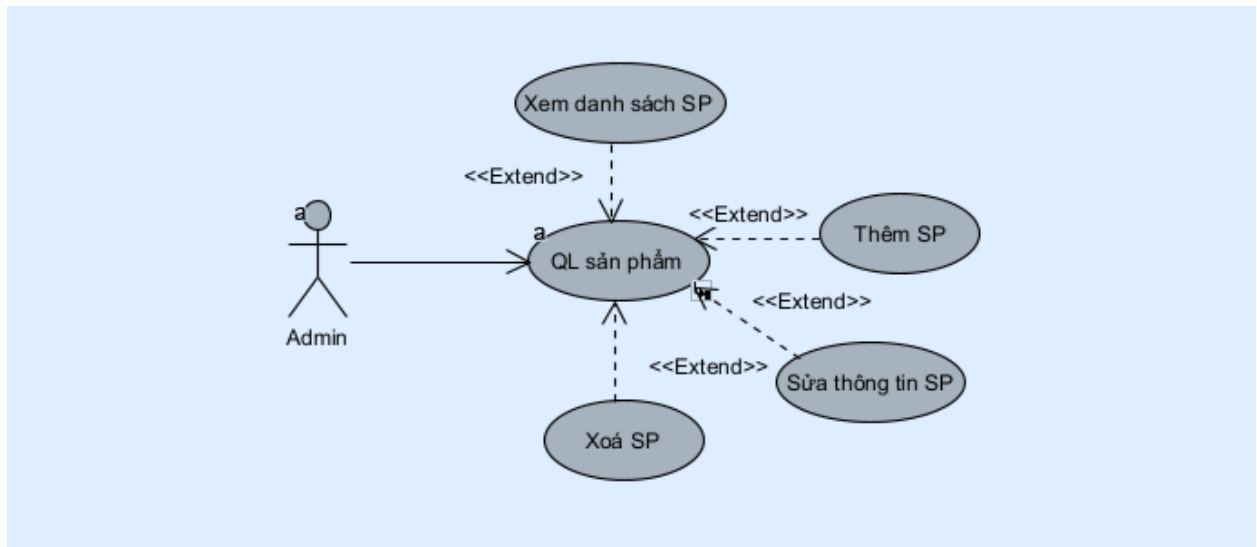
Hình 2.4. Biểu đồ phân rã của use case Xem sản phẩm

- Biểu đồ phân rã của use case Quản lý đơn hàng. Hình 2.5 thể hiện biểu đồ phân rã của use case Quản lý đơn hàng, bao gồm có các use case phân rã là: Xem đơn hàng, Huỷ đơn và Duyệt đơn.



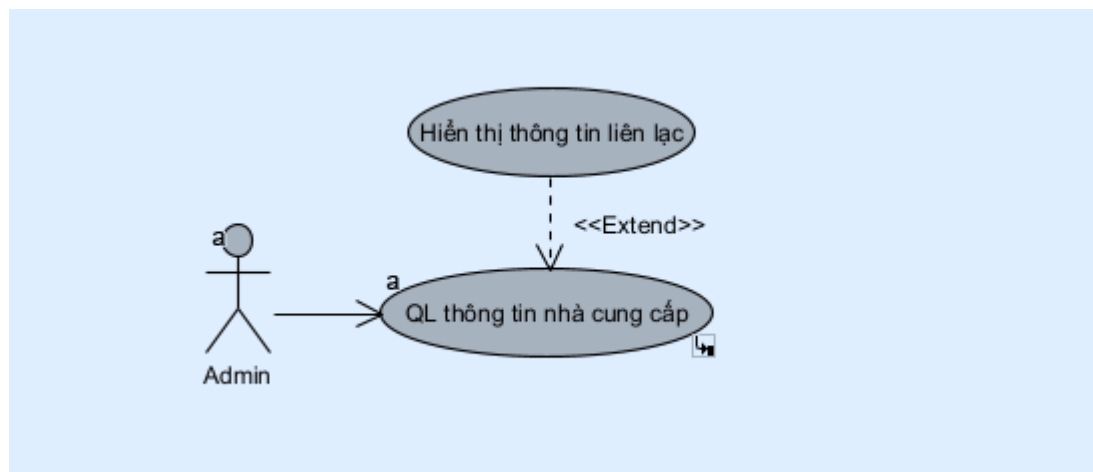
Hình 2.5. Biểu đồ phân rã của use case Quản lý đơn hàng

- Biểu đồ phân rã của use case Quản lý sản phẩm. Hình 2.6 thể hiện được các use case phân rã của chức năng Quản lý sản phẩm gồm: Xem danh sách sản phẩm, thêm sản phẩm, xoá sản phẩm, sửa thông tin sản phẩm.



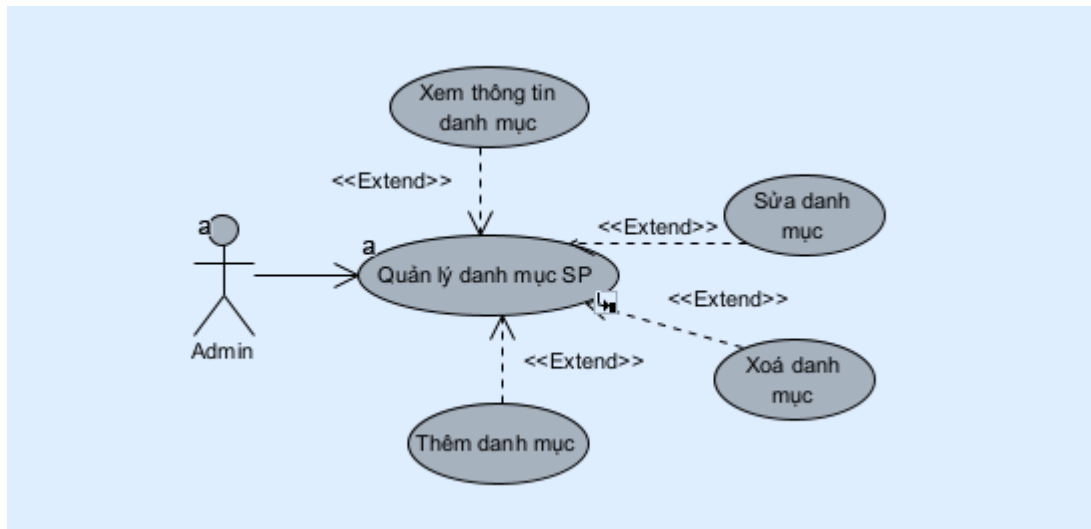
Hình 2.6. Biểu đồ phân rã use case Quản lý sản phẩm

- Biểu đồ phân rã của use case Quản lý thông tin nhà cung cấp, gồm có use case phân rã là Hiển thị thông tin liên lạc của nhà cung cấp.



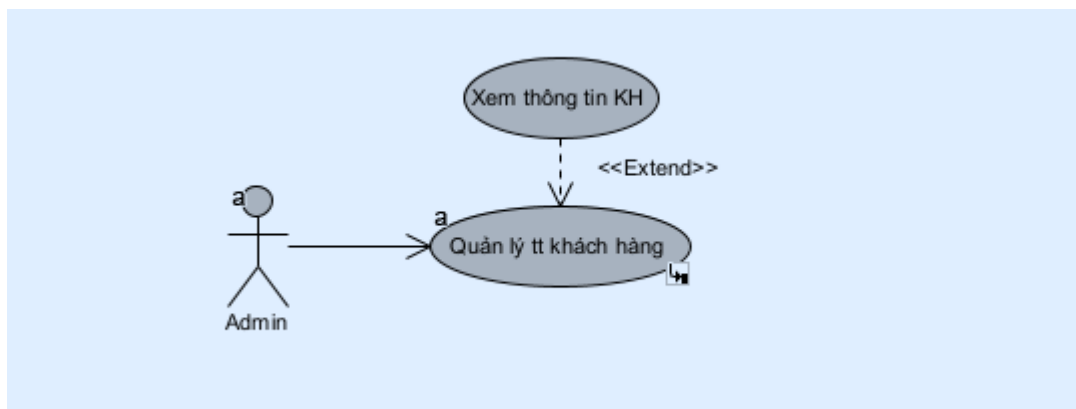
Hình 2.7. Biểu đồ phân rã use case Quản lý thông tin nhà cung cấp.

- Biểu đồ phân rã của use case Quản lý danh mục sản phẩm bao gồm có các use case phân rã là: Xem thông tin danh mục, Sửa danh mục, Xoá danh mục và Thêm danh mục.



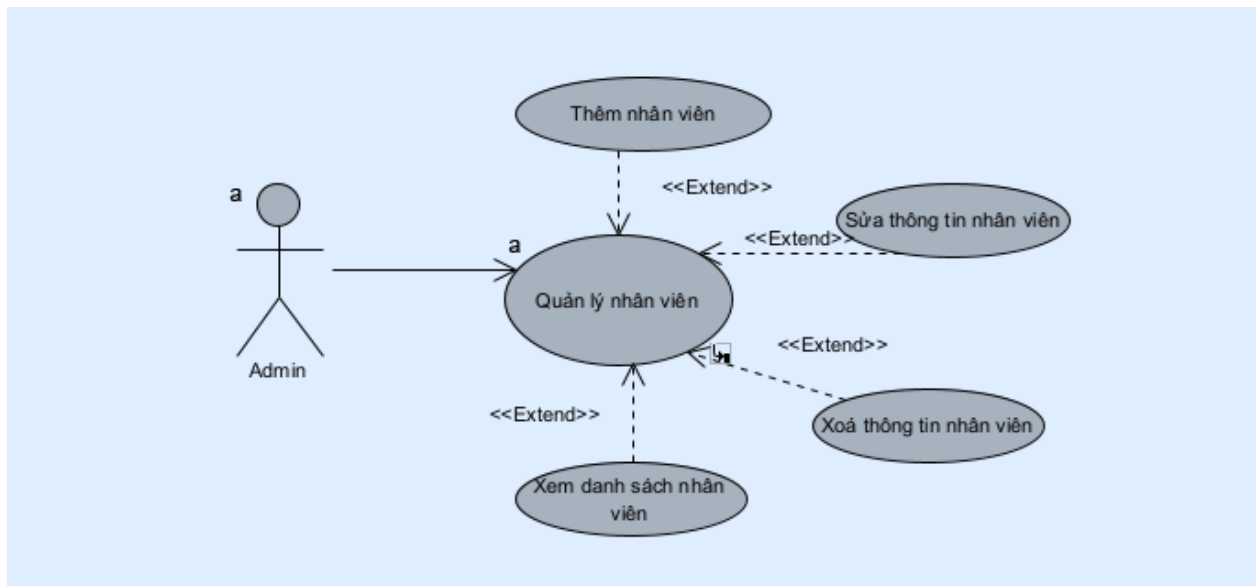
Hình 2.8. Biểu đồ use case phân rã của use case Quản lý danh mục sản phẩm

- Biểu đồ use case phân rã Quản lý thông tin khách hàng chỉ gồm có một use case phân rã là Xem thông tin khách hàng.



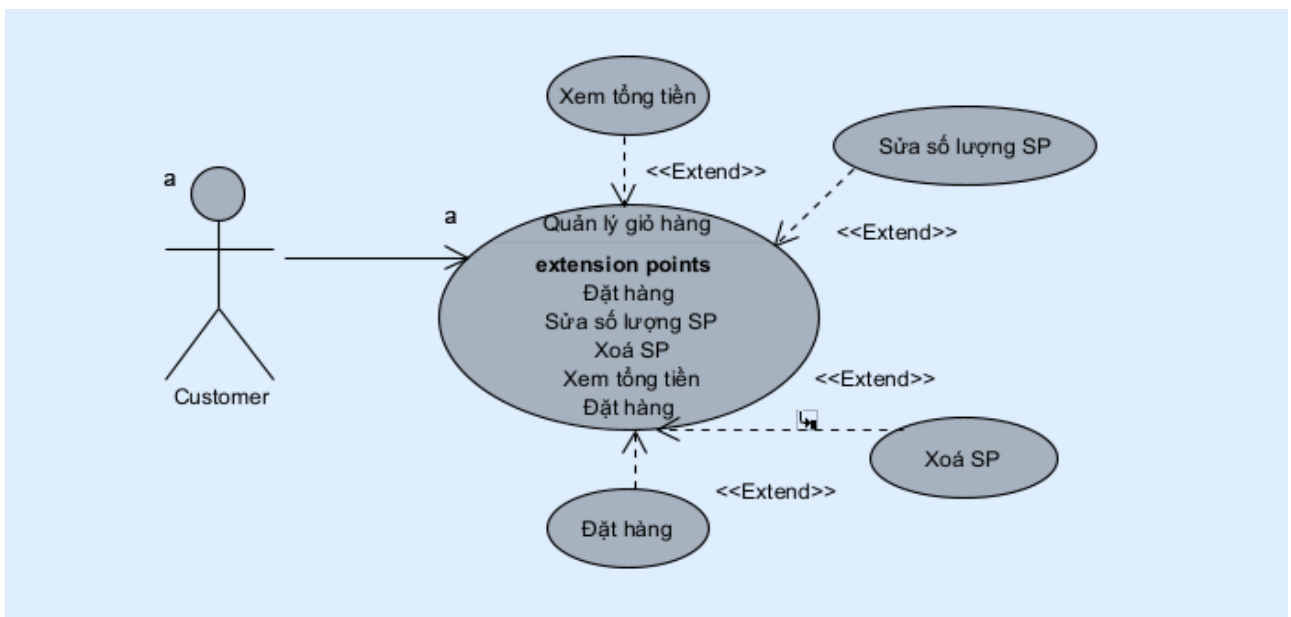
Hình 2.9. Biểu đồ use case phân rã Quản lý thông tin khách hàng

- Biểu đồ phân rã use case Quản lý nhân viên được thể hiện trong hình 2.10 bao gồm có các use case phân rã như: Xem danh sách nhân viên, Xoá thông tin nhân viên, Sửa thông tin nhân viên, Thêm nhân viên.



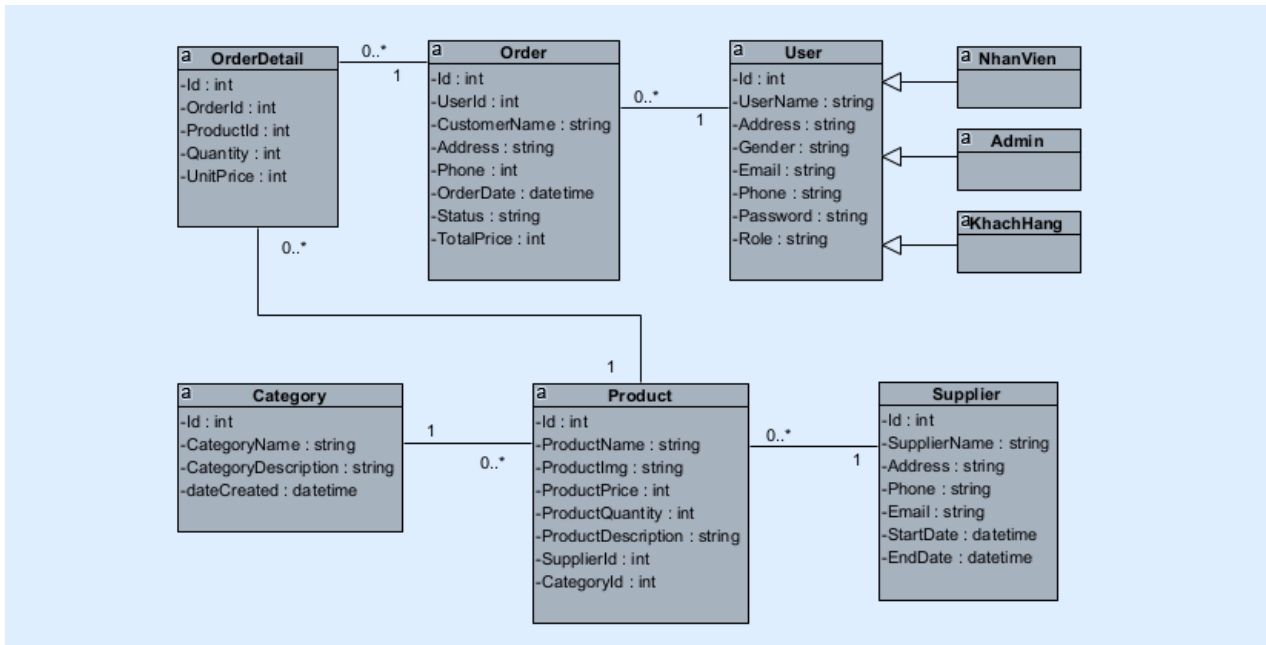
Hình 2.10. Biểu đồ phân rã use case Quản lý nhân viên

- Hình 2.11 thể hiện biểu đồ phân rã use case Quản lý giỏ hàng bao gồm có các use case phân rã như sau: Xem tổng tiền, Sửa số lượng SP, Xoá sản phẩm và Đặt hàng.



Hình 2.11. Biểu đồ phân rã use case Quản lý giỏ hàng

### 2.2.2. Biểu đồ lớp thực thể (Class Diagram)



Hình 2.12. Biểu đồ lớp thực thể

Các thực thể chính trong hệ thống bao gồm:

- Sản phẩm (Product):

Bao gồm những thuộc tính như: id, ProductName, ProductDescription, ProductPrice, ProductImg, ProductQuantity.

- Đơn hàng (Order):

Bao gồm những thuộc tính như: id, CustomerName, Address, Phone, OrderDate, Status, TotalPrice.

- Chi tiết đơn hàng (OrderDetail):

Bao gồm những thuộc tính như: Quantity, UnitPrice

- User (Người dùng):

Bao gồm những thuộc tính như: id, UserName, Address, Gender, Email, Phone, Password, Role.

- Loại sản phẩm (Category):

Thuộc tính: id, CategoryName, CategoryDescription, dateCreated.

- Nhà cung cấp (Supplier):

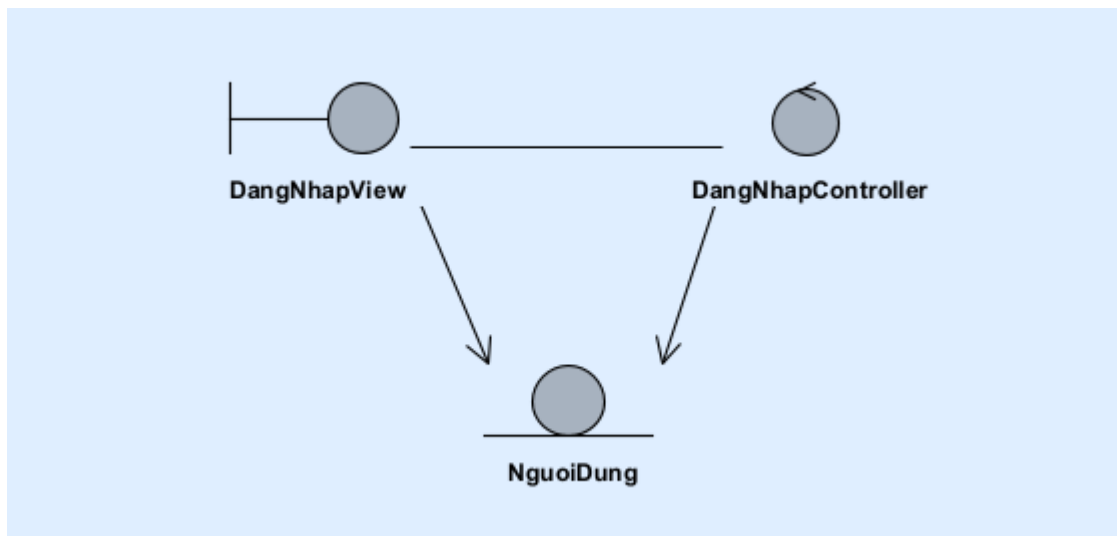
Thuộc tính: id, SupplierName, Address, Phone, Email, StartDate, EndDate.

❖ Môi quan hệ giữa các thực thể:

- Một User (role = Khách hàng) có thể có nhiều Đơn hàng.
- Một Đơn hàng bao gồm nhiều Chi tiết đơn hàng.
- Mỗi Chi tiết đơn hàng liên quan đến Sản phẩm.
- Một Loại sản phẩm có thể có nhiều Sản phẩm.
- Mỗi Sản phẩm thì chỉ thuộc về một Nhà cung cấp.
- Một User (role = Nhân viên or Admin) có thể xử lý nhiều Đơn hàng.

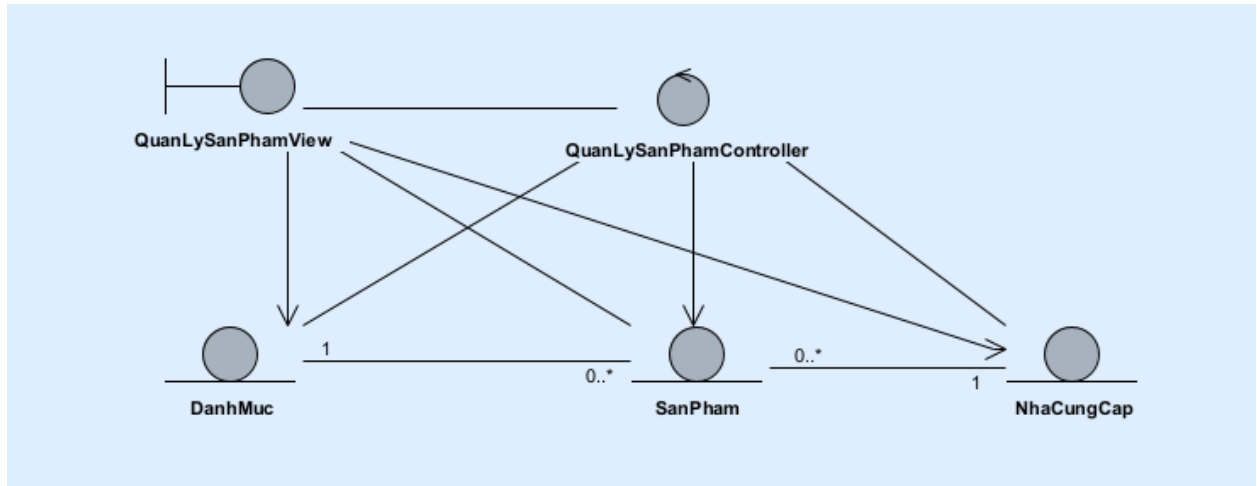
### 2.2.3. Biểu đồ lớp phân tích (VOPC)

Biểu đồ lớp phân tích của use case Đăng nhập gồm các lớp: lớp boundary DangNhapView, lớp control DangNhapController và một lớp thực thể User.



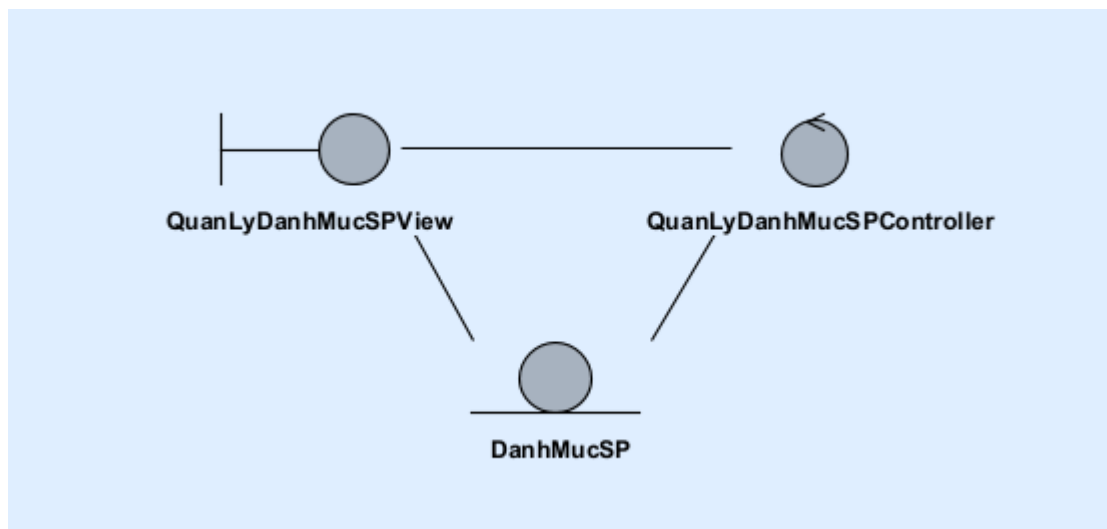
Hình 2.13. Biểu đồ VOPC use case Đăng nhập

Biểu đồ lớp phân tích use case Quản lý sản phẩm gồm các lớp: lớp boundary QuanLySanPhamView, lớp control QuanLySanPhamController và các lớp thực thể DanhMuc, SanPham và NhaCungCap.



Hình 2.14. Biểu đồ VOPC use case Quản lý sản phẩm

Biểu đồ lớp phân tích Quản lý danh mục sản phẩm gồm có: lớp boundary QuanLyDanhMucSPView, lớp control QuanLyDanhMucSPController và lớp thực thể DanhMucSP.

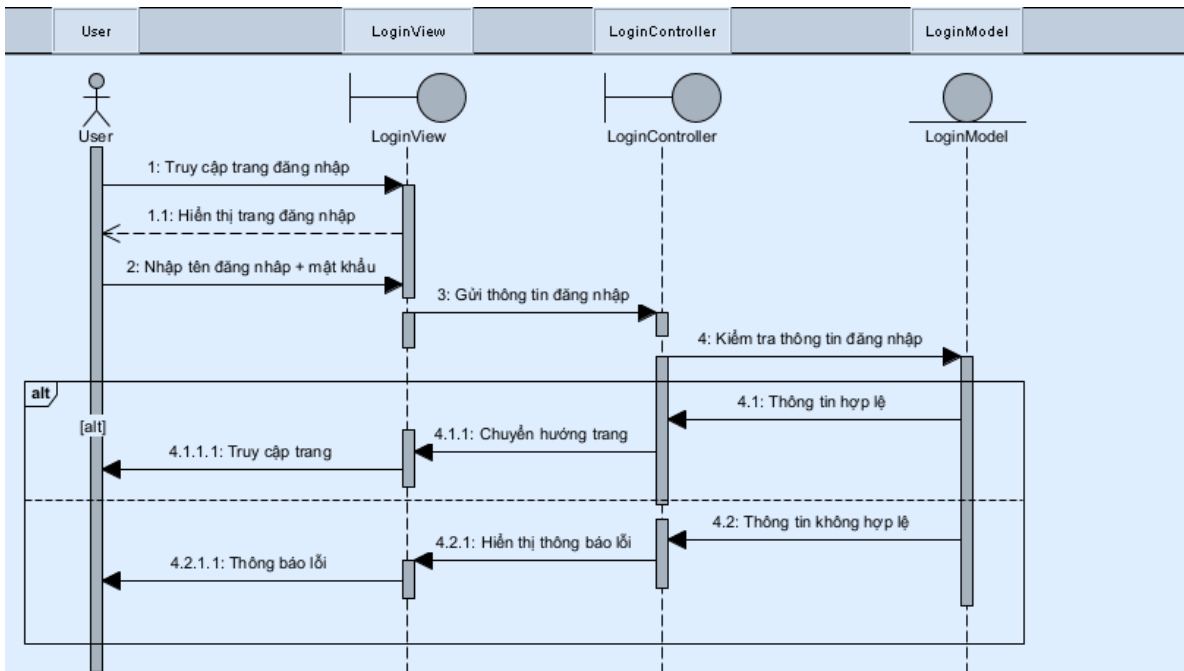


Hình 2.15. Biểu đồ lớp VOPC Quản lý danh mục



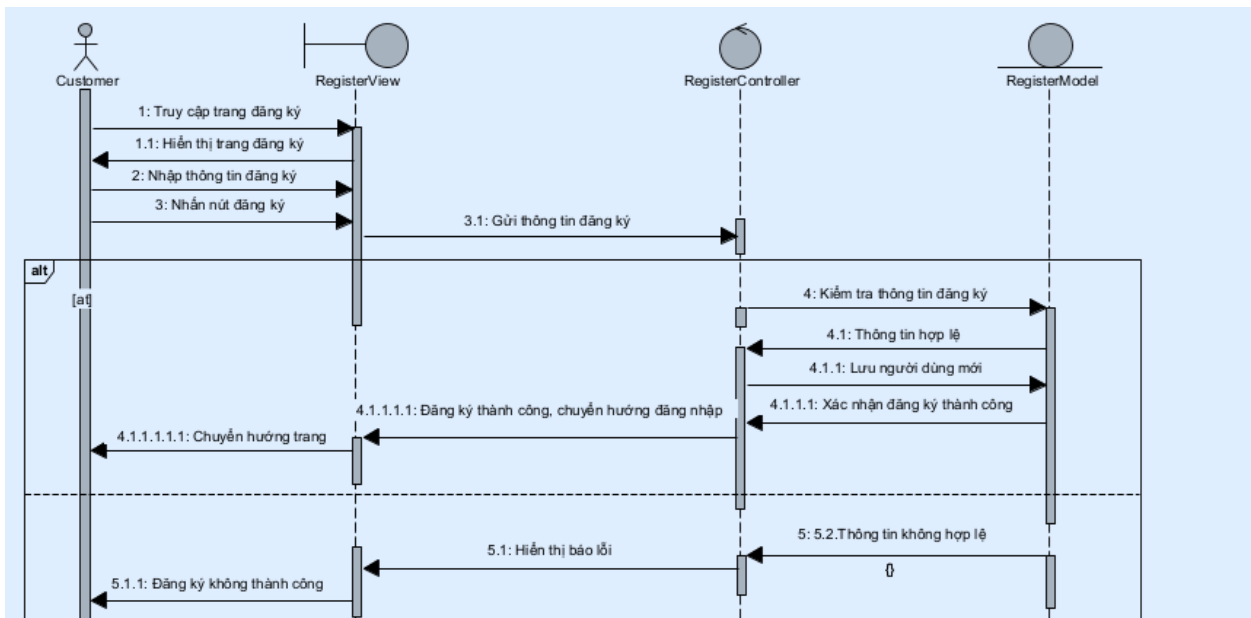
### 2.2.4. Biểu đồ tuần tự (Sequence Diagram)

- Biểu đồ tuần tự của use case Đăng nhập.



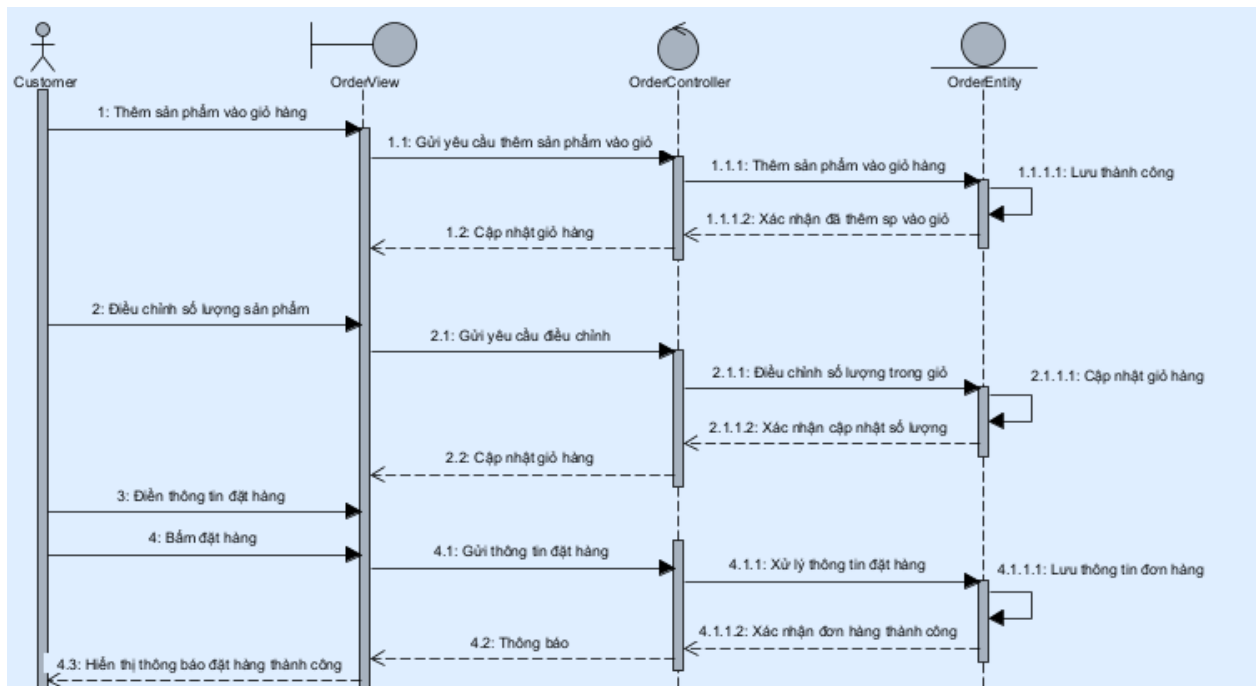
Hình 2.16. Biểu đồ tuần tự của use case Đăng nhập

- Biểu đồ tuần tự use case Đăng ký



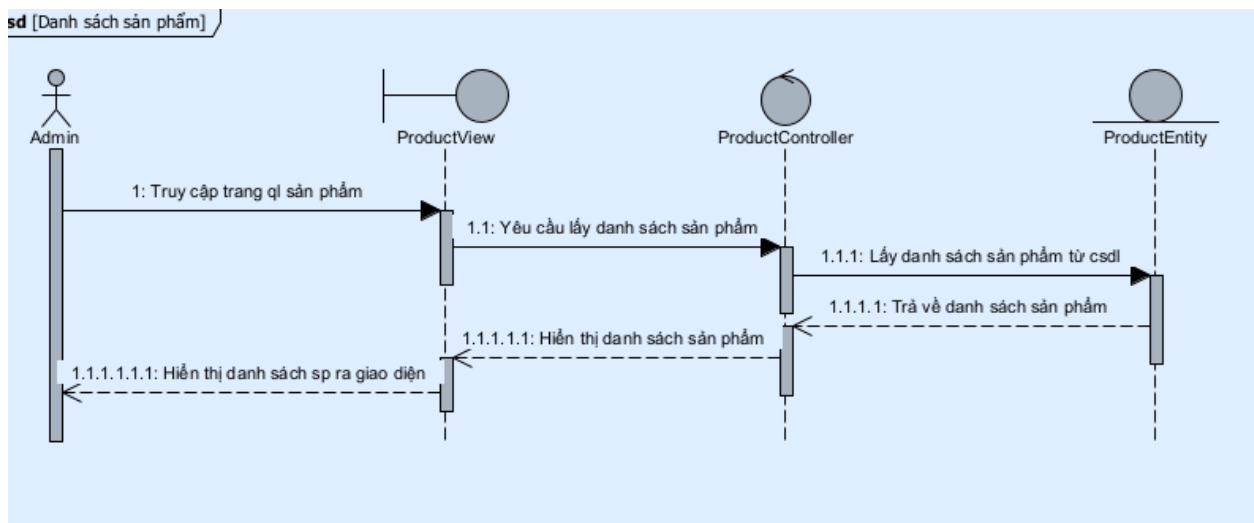
Hình 2.17. Biểu đồ tuần tự use case Đăng ký

- Biểu đồ tuần tự use case Đặt hàng



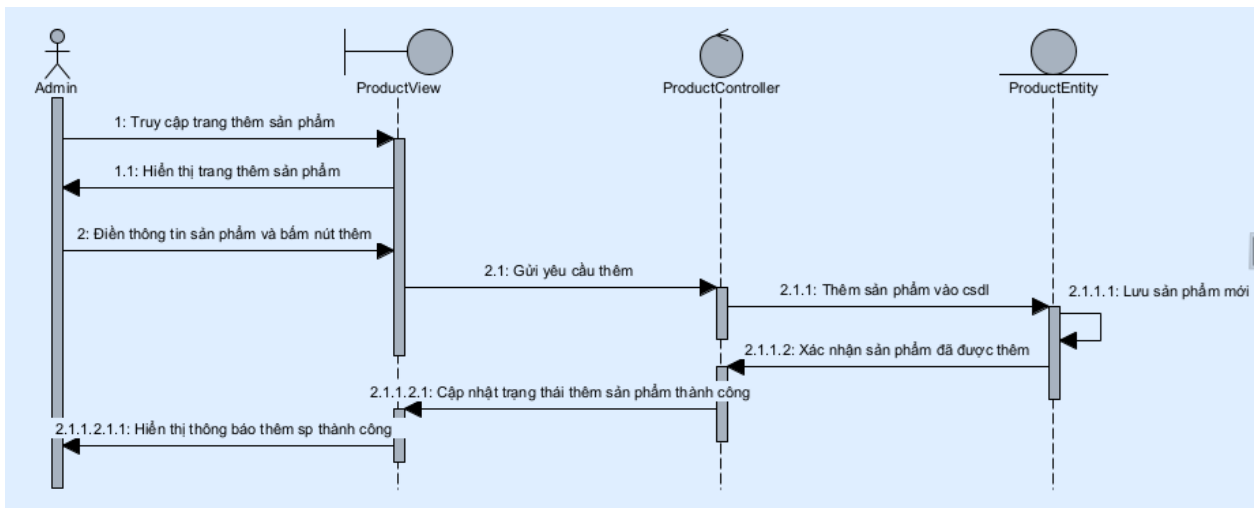
Hình 2.18. Biểu đồ tuần tự use case Đặt hàng

- Biểu đồ tuần tự use case Xem sản phẩm



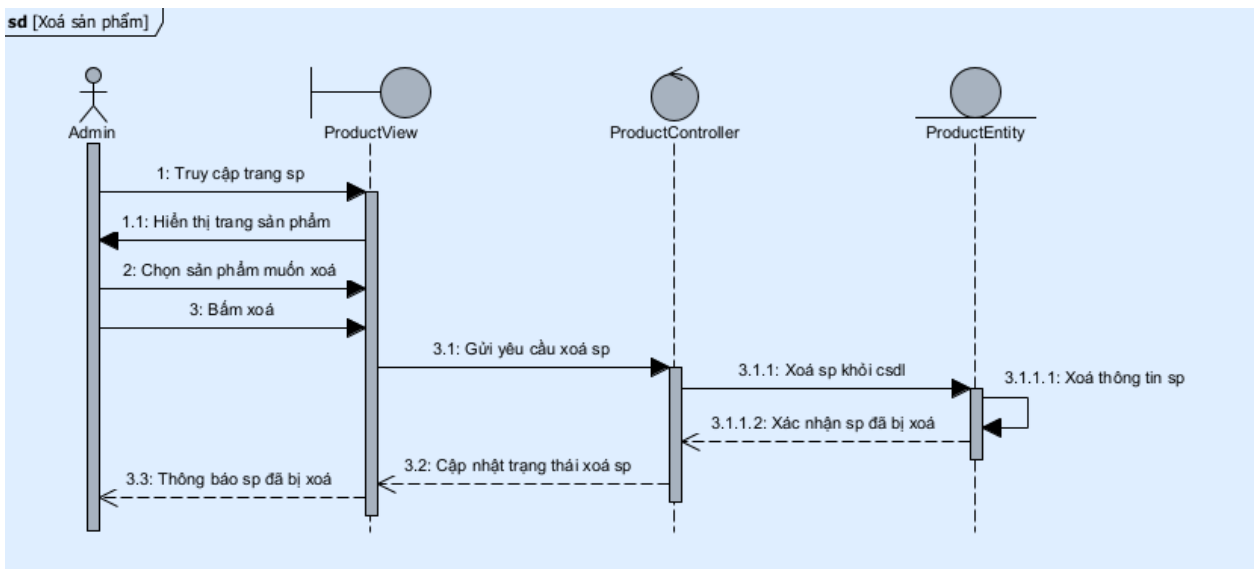
Hình 2.19. Biểu đồ tuần tự use case Xem sản phẩm

- Biểu đồ tuần tự use case Thêm sản phẩm.



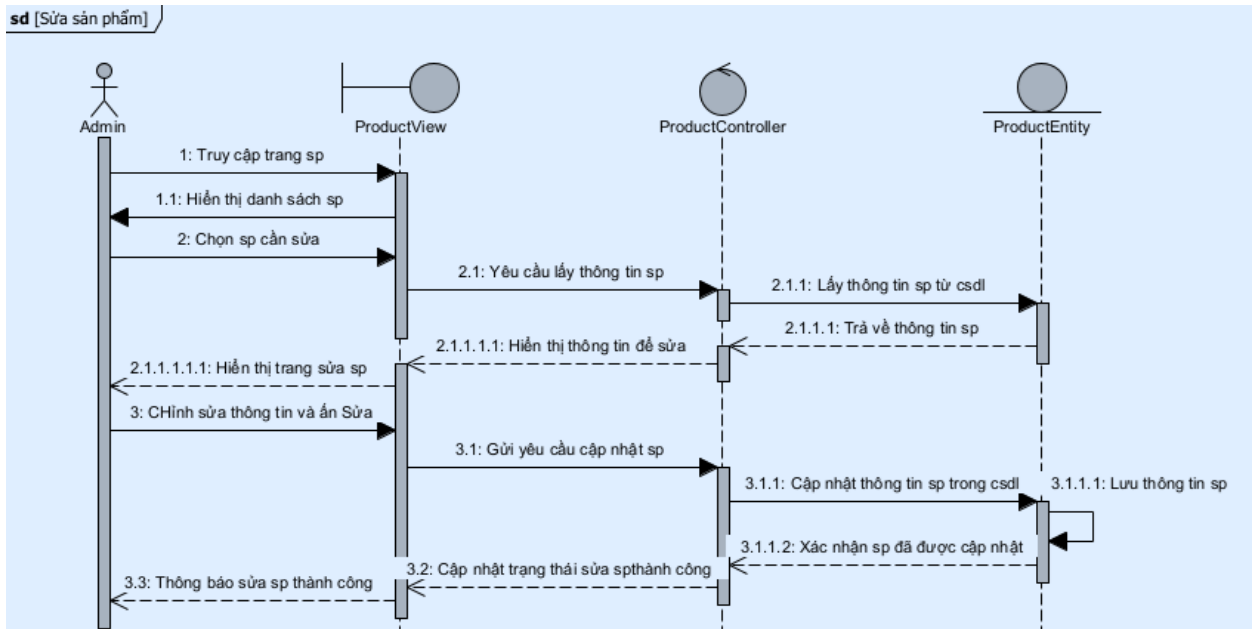
Hình 2.20. Biểu đồ tuần tự use case Thêm sản phẩm.

- Biểu đồ tuần tự use case Xóa sản phẩm.



Hình 2.21. Biểu đồ tuần tự use case Xóa sản phẩm.

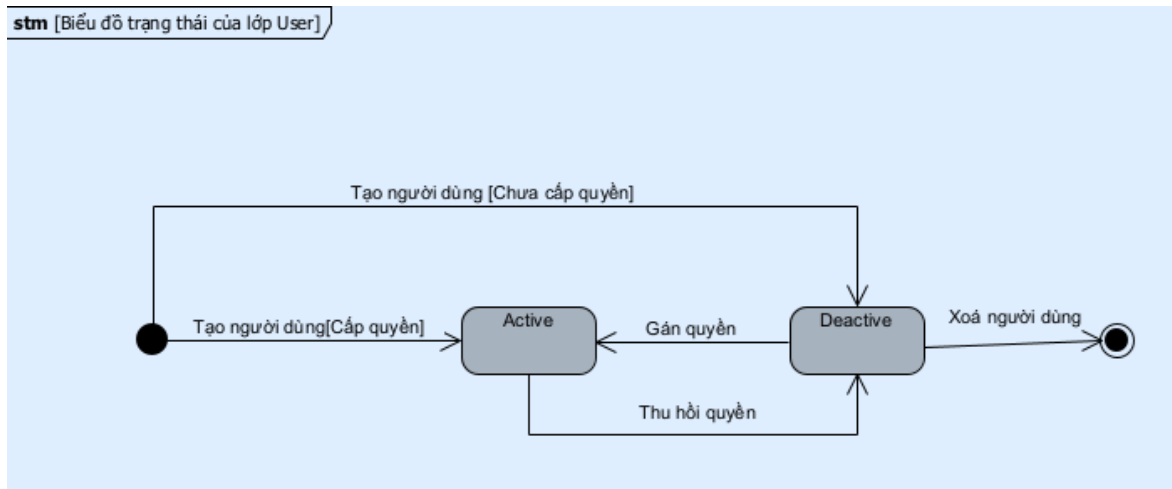
- Biểu đồ tuần tự use case Sửa thông tin sản phẩm.



Hình 2.22. Biểu đồ tuần tự use case Sửa thông tin sản phẩm

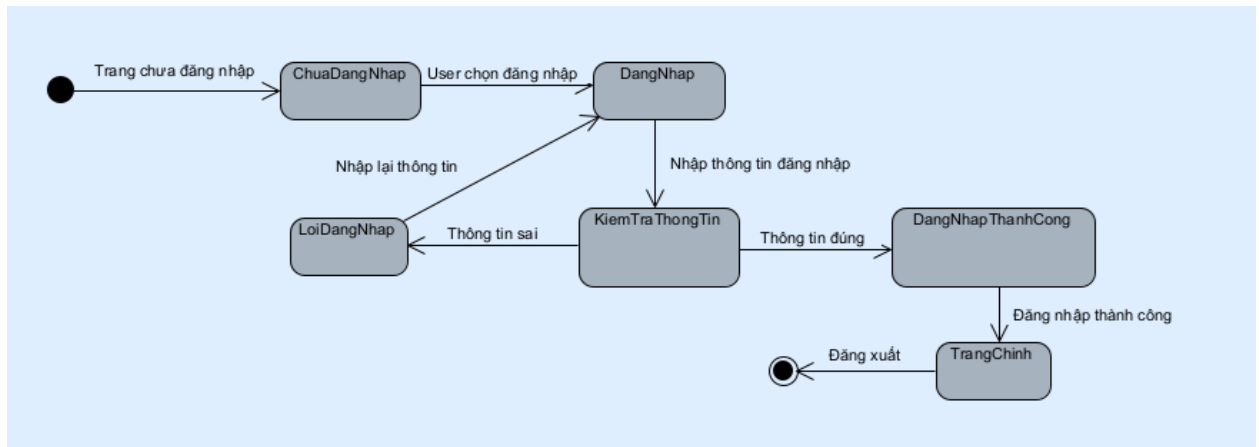
### 2.2.5. Biểu đồ trạng thái (State Machine Diagram)

Biểu đồ trạng thái của lớp User (role = Nhân viên). Đối tượng lớp này có thể có 2 trạng thái khác nhau gồm: hoạt động (active) và ngưng hoạt động (deactive) và các chuyển trạng thái có thể xảy ra giữa các trạng thái như hình dưới đây.



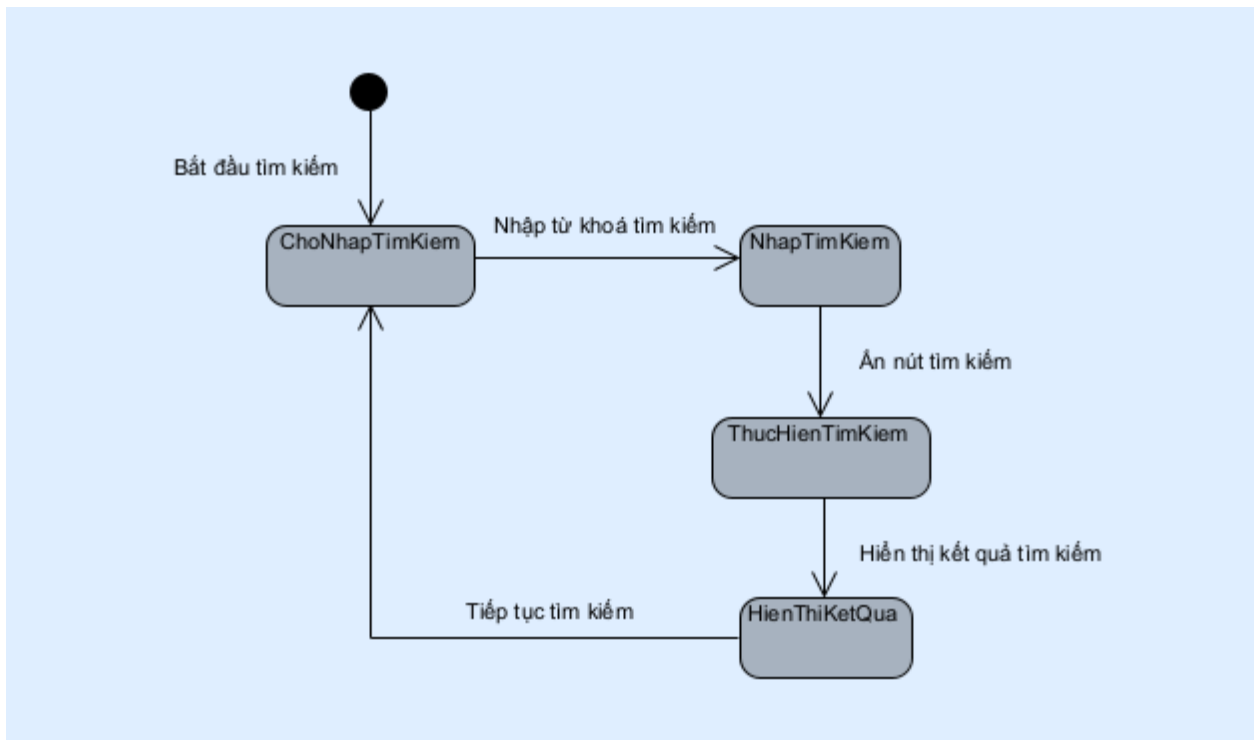
Hình 2.23. Biểu đồ trạng thái của lớp User

Biểu đồ trạng thái của lớp Đăng nhập có thể có các trạng thái khác nhau như: chưa đăng nhập, đăng nhập, kiểm tra thông tin, lỗi đăng nhập, đăng nhập thành công.



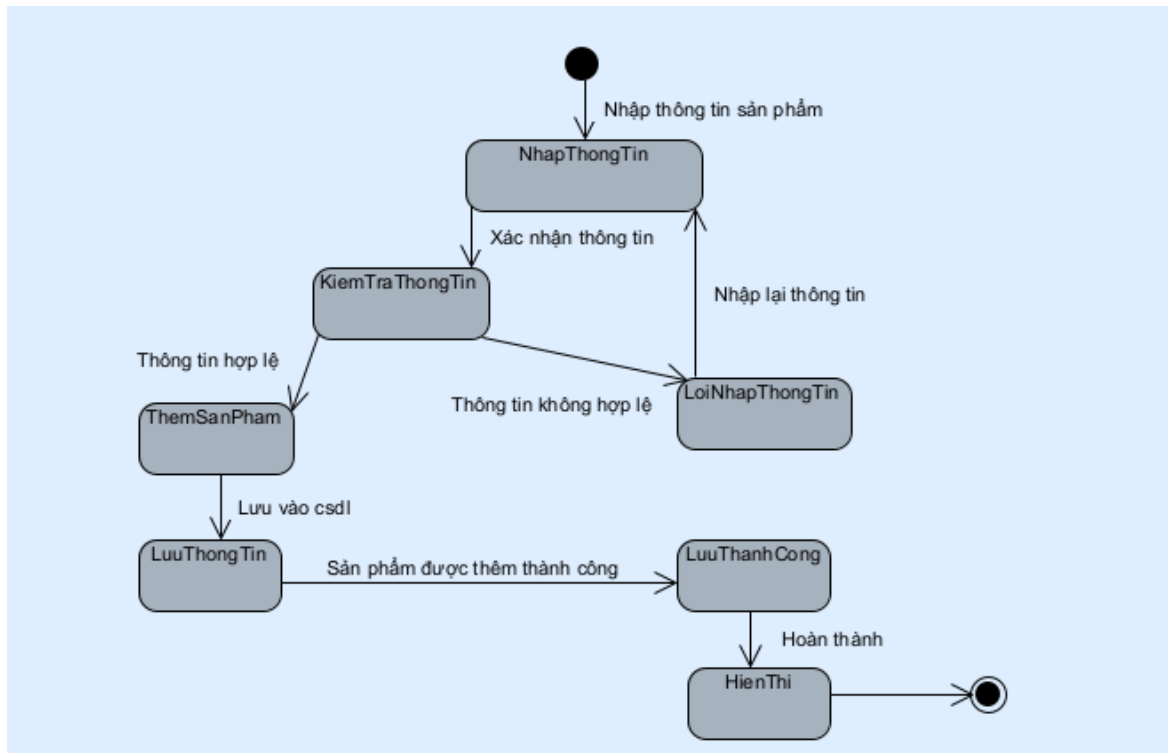
Hình 2.24. Biểu đồ trạng thái lớp Đăng nhập

Biểu đồ trạng thái của lớp Tìm kiếm có các trạng thái sau: chờ nhập tìm kiếm, nhập tìm kiếm, thực hiện tìm kiếm và hiển thị kết quả.



Hình 2. 25. Biểu đồ trạng thái lớp Tìm kiếm

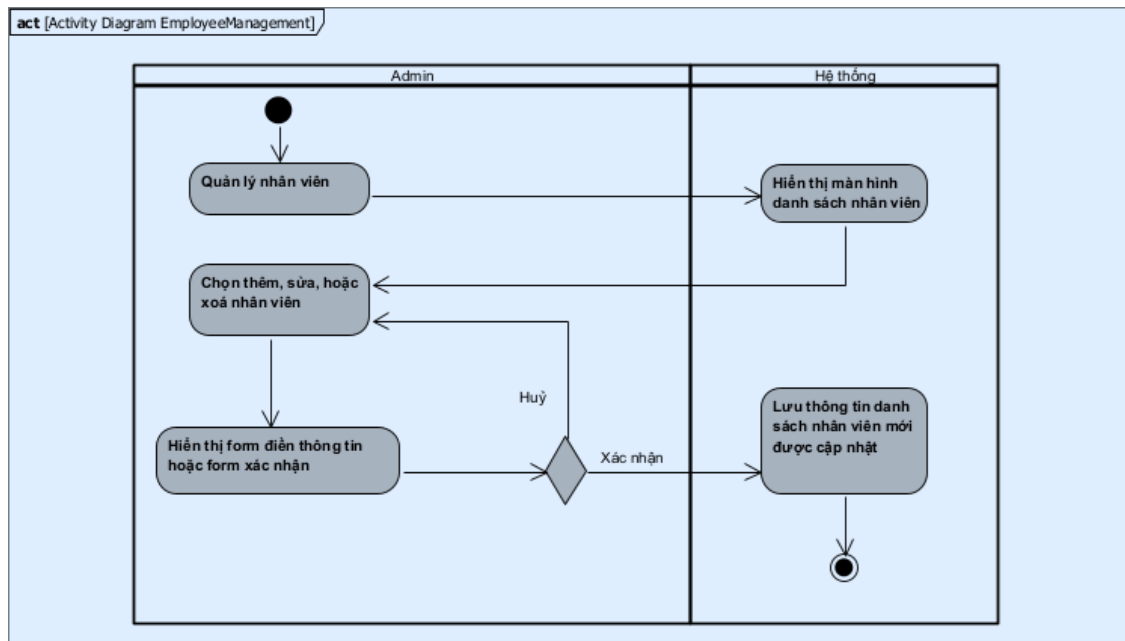
Biểu đồ trạng thái của lớp Thêm sản phẩm có thể có các trạng thái như sau: nhập thông tin, kiểm tra thông tin đầu vào, thêm sản phẩm hoặc lỗi nhập thông tin, lưu thông tin sản phẩm, lưu thành công và hiển thị.



Hình 2.26. Biểu đồ trạng thái của lớp Thêm sản phẩm

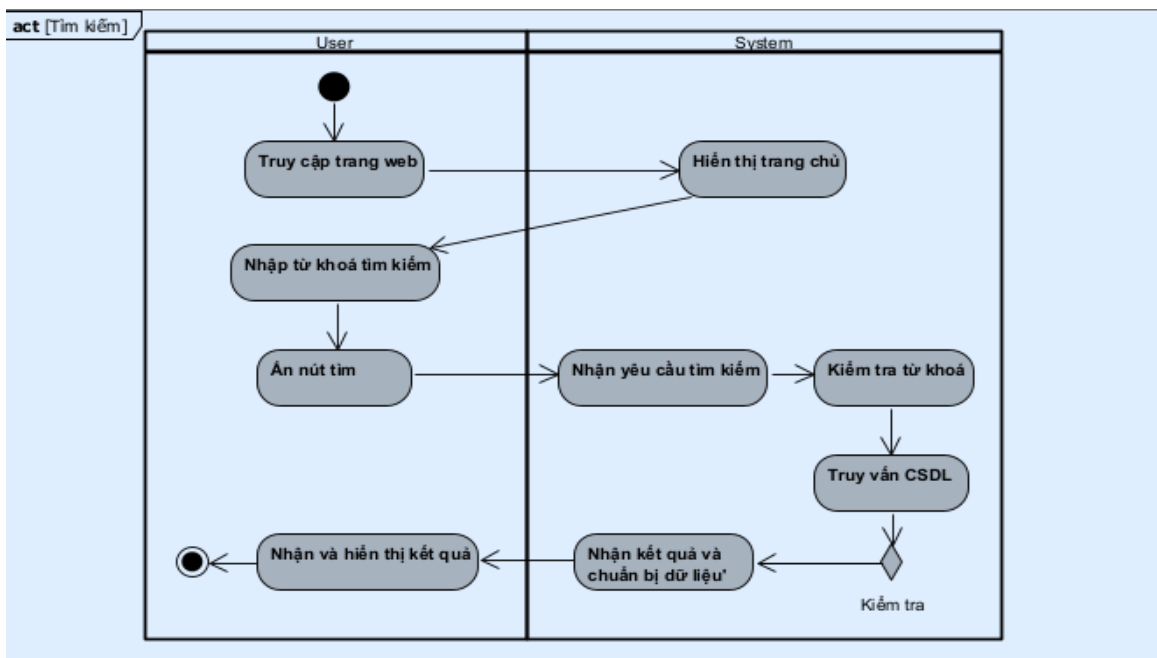
### 2.2.6. Biểu đồ hoạt động (Activity Diagram)

Biểu đồ tuần tự use case Quản lý nhân viên



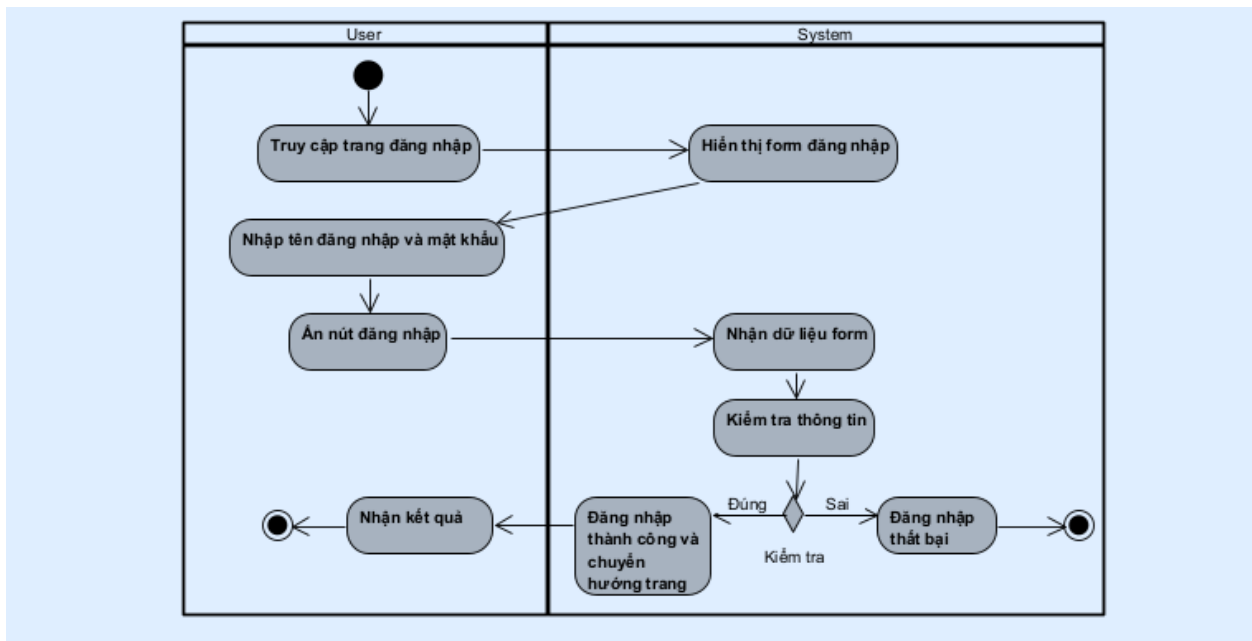
Hình 2.27. Biểu đồ tuần tự use case Quản lý nhân viên

Biểu đồ tuần tự use case Tìm kiếm sản phẩm.



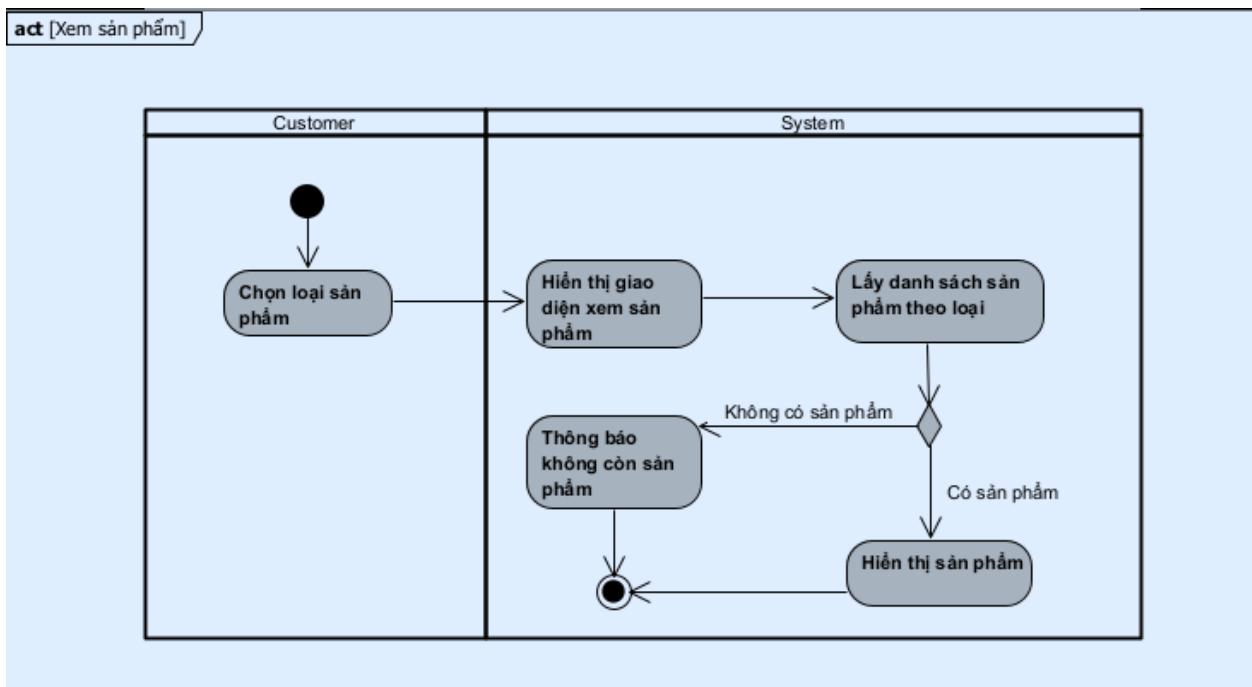
Hình 2.28. Biểu đồ tuần tự use case Tìm kiếm sản phẩm

### Biểu đồ tuần tự use case Đăng nhập



Hình 2.29. Biểu đồ tuần tự use case Đăng nhập

### Biểu đồ tuần tự Xem sản phẩm (theo loại/ danh mục)



Hình 2.30. Biểu đồ tuần tự Xem sản phẩm (theo loại/ danh mục)



**Tiểu kết**

Chương 2 đã trình bày quá trình thiết kế và xây dựng cơ sở dữ liệu, bao gồm việc xác định các thực thể và quan hệ, cùng với biểu đồ Database Diagram để minh họa cấu trúc. Ngoài ra, chương này cũng giới thiệu các biểu đồ UML như Use Case Diagram, Class Diagram, Sequence Diagram, State Machine Diagram và Activity Diagram để mô tả chi tiết các khía cạnh khác nhau của hệ thống, từ chức năng người dùng đến cấu trúc lớp và luồng các công việc, các module tạo nền tảng vững chắc cho việc phát triển hệ thống.

## Chương 3

### XÂY DỰNG, TRIỂN KHAI VÀ TÍCH HỢP

#### 3.1. Xây dựng và triển khai

##### 3.1.1. FrontEnd

Khởi tạo project bằng lệnh `npm create vite@latest my-vite-app -- --template react-ts` và cài đặt Tailwind CSS bằng lệnh `npm install -D tailwindcss postcss autoprefixer`. Sau đó chúng ta có thể bắt đầu cấu hình project của mình.

Các thư mục để quản lý project như:

- “node\_modules/ ” :Thư mục này chứa tất cả các dependencies (phụ thuộc) mà bạn đã cài đặt thông qua npm.
- “public/ ” : Thư mục này chứa các tài nguyên tĩnh (static assets) như hình ảnh, biểu tượng (favicon), và các file tĩnh khác mà bạn muốn đưa trực tiếp vào ứng dụng mà không qua quá trình biên dịch.
- “src/ ” : Đây là thư mục chứa mã nguồn chính của chương trình. Bao gồm:
  - “assets/”: Thư mục này dùng để lưu trữ các tài nguyên tĩnh như hình ảnh, biểu tượng, fonts, v.v. trong mã nguồn của bạn.
  - “components/”: Thư mục này chứa các thành phần React (React components). Bạn có thể tạo thêm các thành phần con tại đây để tổ chức mã nguồn của mình tốt hơn.
  - “App.tsx”: Đây là thành phần gốc của ứng dụng React của bạn. Tất cả các thành phần khác sẽ được nhúng vào đây.
  - “index.css”: Đây là file CSS chính của ứng dụng. Bạn sẽ thêm các chỉ thị của Tailwind CSS vào đây.
  - “main.tsx ”: File này là entry point của ứng dụng React. Nó khởi tạo và gắn kết ứng dụng React vào DOM.
- “.gitignore” : File này chứa danh sách các file và thư mục mà bạn muốn Git bỏ qua.

- “index.html” : Đây là file HTML chính của ứng dụng. Vite sử dụng file này như là template để nhúng ứng dụng React.
- “package.json” : File này chứa thông tin về project và danh sách các dependencies. Nó cũng định nghĩa các script để chạy, build và test ứng dụng.
- “postcss.config.cjs” : File này là cấu hình cho PostCSS, một công cụ để biến đổi CSS với các plugin JavaScript. Tailwind CSS sử dụng PostCSS để biên dịch các chỉ thị CSS của nó.
- “tailwind.config.js” : File này chứa cấu hình cho Tailwind CSS. Ta thể tùy chỉnh các thiết lập của Tailwind CSS trong file này như màu sắc, font chữ, breakpoints, v.v.
- “tsconfig.json”: File này chứa cấu hình cho TypeScript. Bạn có thể tùy chỉnh các thiết lập của TypeScript trong file này như phiên bản ECMAScript, các đường dẫn alias, strict mode, v.v.
- “vite.config.ts”: File này chứa cấu hình cho Vite. Bạn có thể tùy chỉnh các thiết lập của Vite trong file này như alias, plugins, server settings, v.v.

### 3.1.2. BackEnd

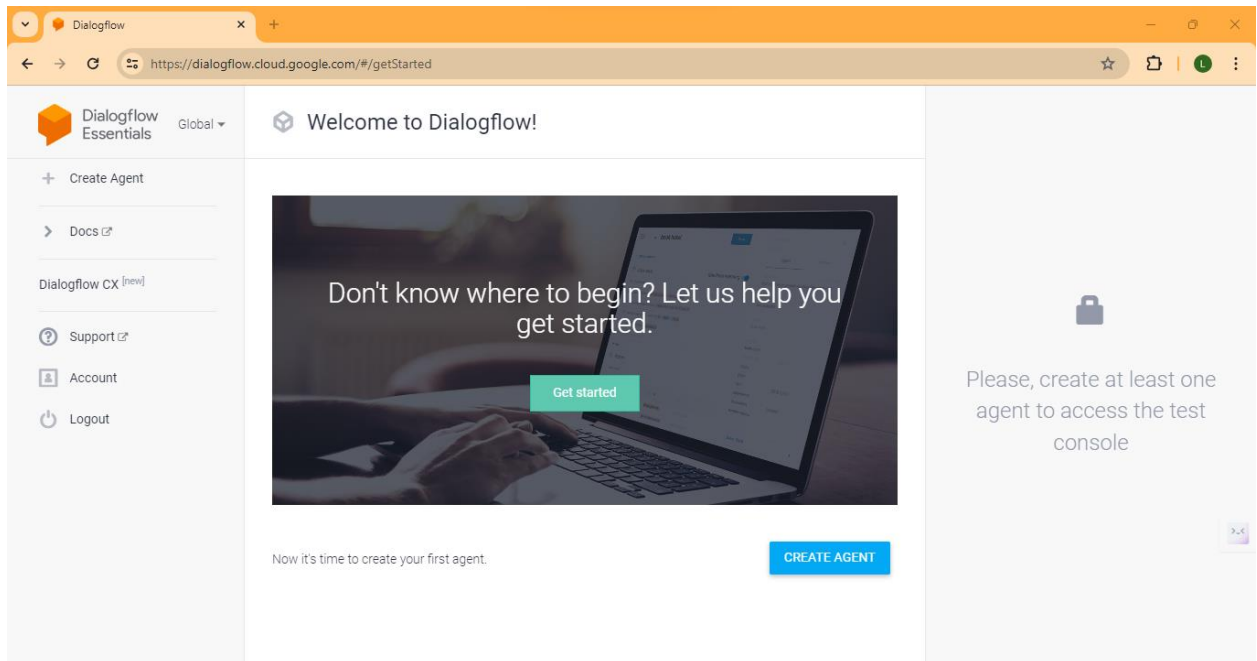
Khởi tạo solution để quản lý cấu trúc project:

- Kira.Domain: Đây là nơi chứa các thực thể (entities) đại diện cho các bảng trong cơ sở dữ liệu. Nó sẽ định nghĩa các interface repository để truy xuất và thao tác với dữ liệu.
- Kira.Application: Xử lý các yêu cầu từ người dùng và thực hiện các quy trình nghiệp vụ. Nó chứa các dịch vụ nghiệp vụ (business services) và các Data Transfer Objects (DTOs). Đặc biệt sẽ Quản lý các cấu hình ánh xạ (mappings) giữa các lớp domain và DTOs.
- Kira.Infrastructure: Chứa các lớp liên quan đến truy cập cơ sở dữ liệu như DbContext và các repository triển khai. Quản lý các file migration để duy trì và cập nhật cấu trúc cơ sở dữ liệu. Và xử lý tích hợp với các dịch vụ bên ngoài và quản lý hạ tầng kỹ thuật.
- Kira.API: Đây là project API chính, nơi triển khai các package liên quan đến ASP.NET Core, bao gồm cả Entity Framework Core. Nó chứa các controller của ASP.NET Core, quản lý cấu hình ứng dụng và tài nguyên tĩnh. Cung cấp điểm tiếp xúc giữa người dùng với hệ thống thông qua các API và xử lý các yêu cầu HTTP và trả về các phản hồi tương ứng.

### 3.1.3. Tích hợp

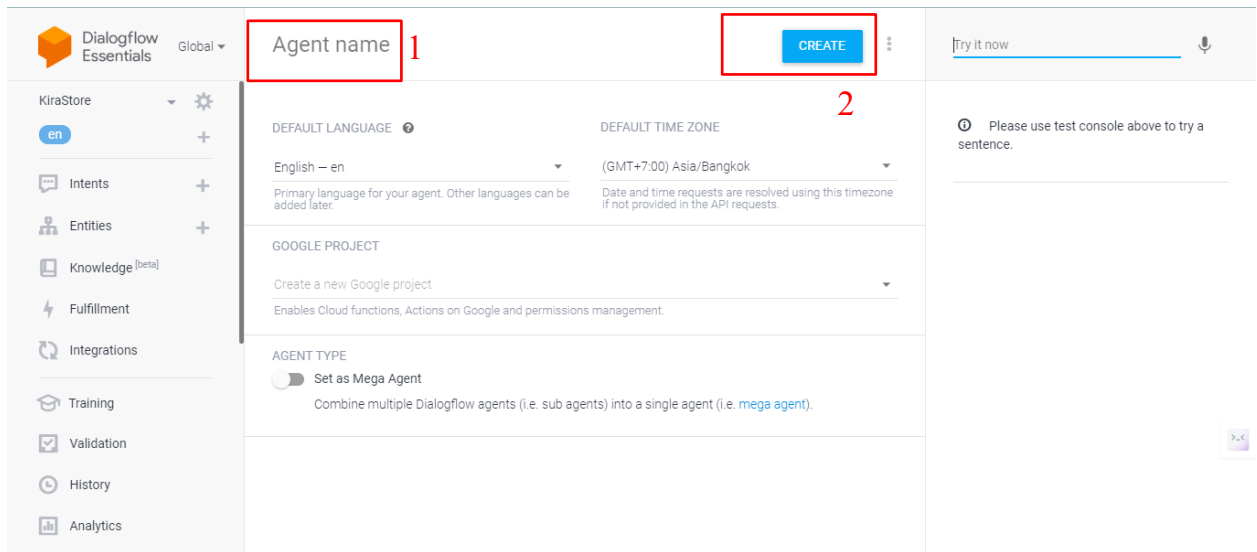
Trong đồ án này em đã tích hợp ứng dụng Google Dialogflow cho hệ thống Chatbot của website. Để tích hợp được ứng dụng Chatbot của Google Dialogflow thì chúng ta có thể thực hiện một số bước cơ bản sau:

- Bước 1: Truy cập vào <https://dialogflow.cloud.google.com/#/getStarted> để có thể bắt đầu sử dụng Google Dialogflow.



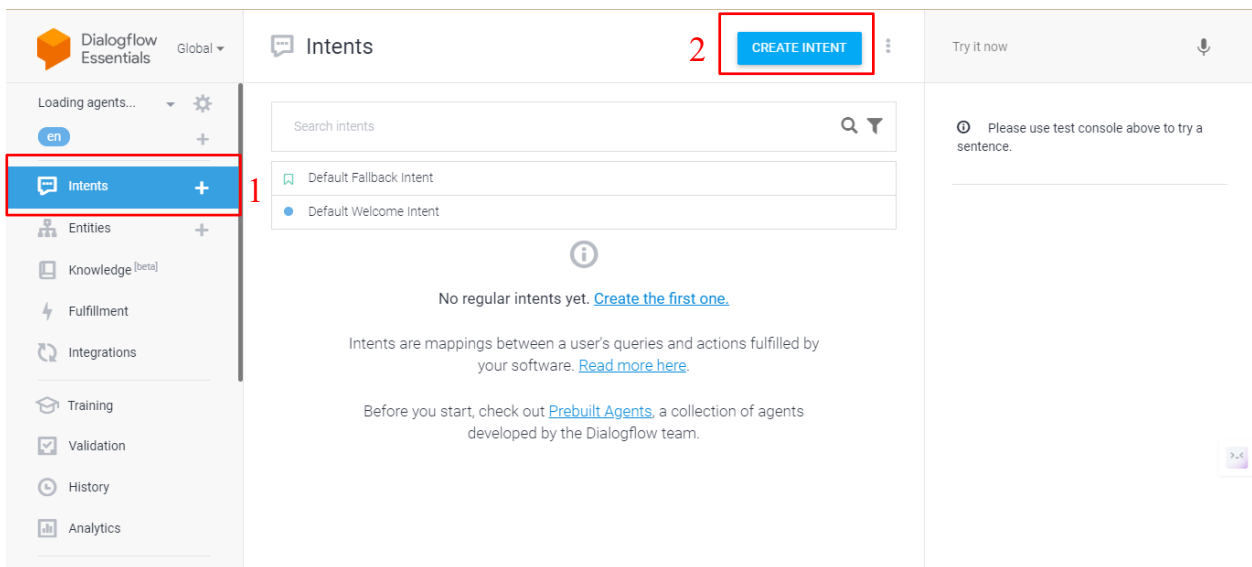
Hình 3. 1. Truy cập Google Dialogflow

- Bước 2: Chọn Create Agent và đặt tên cho Agent của mình phù hợp với project



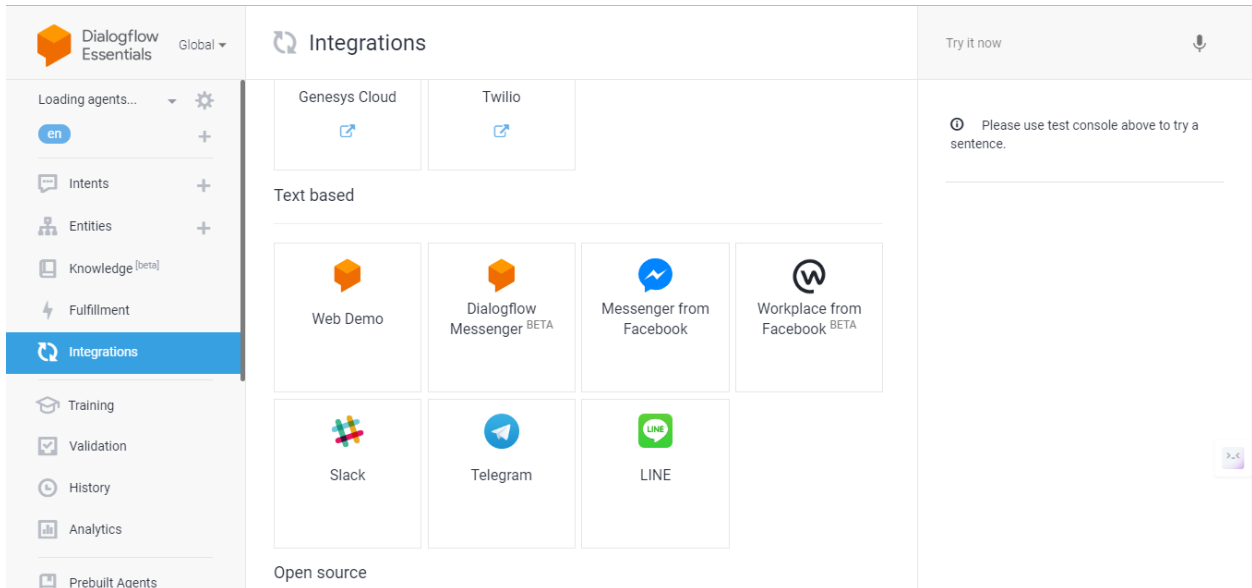
Hình 3.2: Tạo Agent

- Bước 3: Tạo các Intents (các chủ đề giao tiếp giữa chatbot và người dùng)



Hình 3. 3. Tạo Intents

- Bước 4: Cuối cùng chúng ta cũng có thể chọn Integrations để tạo ra mã tích hợp vào đồ án. Ở bước này tùy thuộc vào mục đích sử dụng mà chúng ta sẽ chọn loại tương ứng.

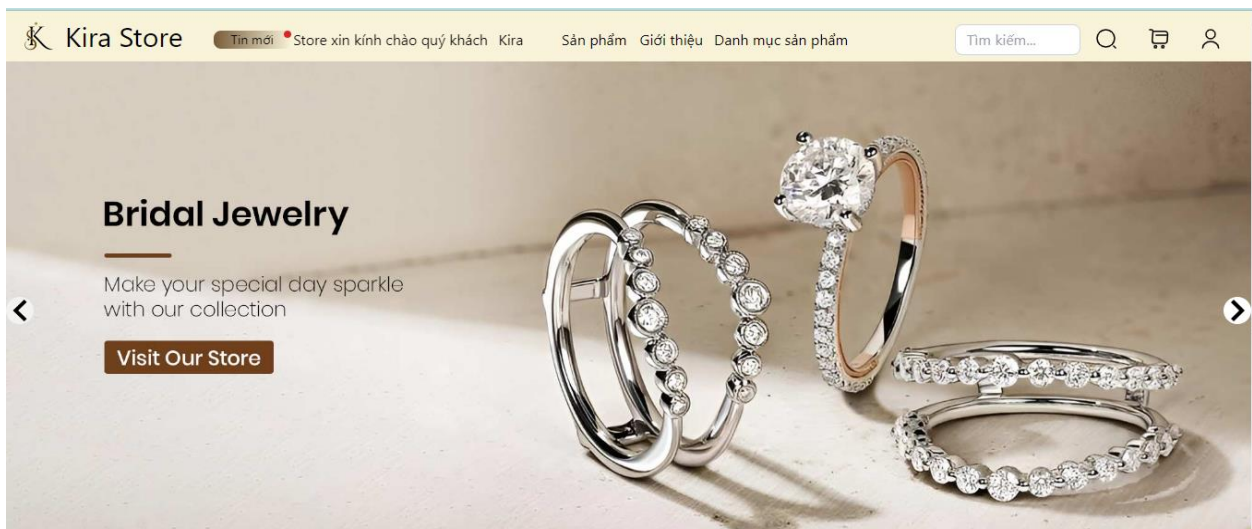


Hình 3. 4. Sinh mã để tích hợp

## 3.2. Kết quả xây dựng giao diện người dùng

### 3.2.1. Trang chủ

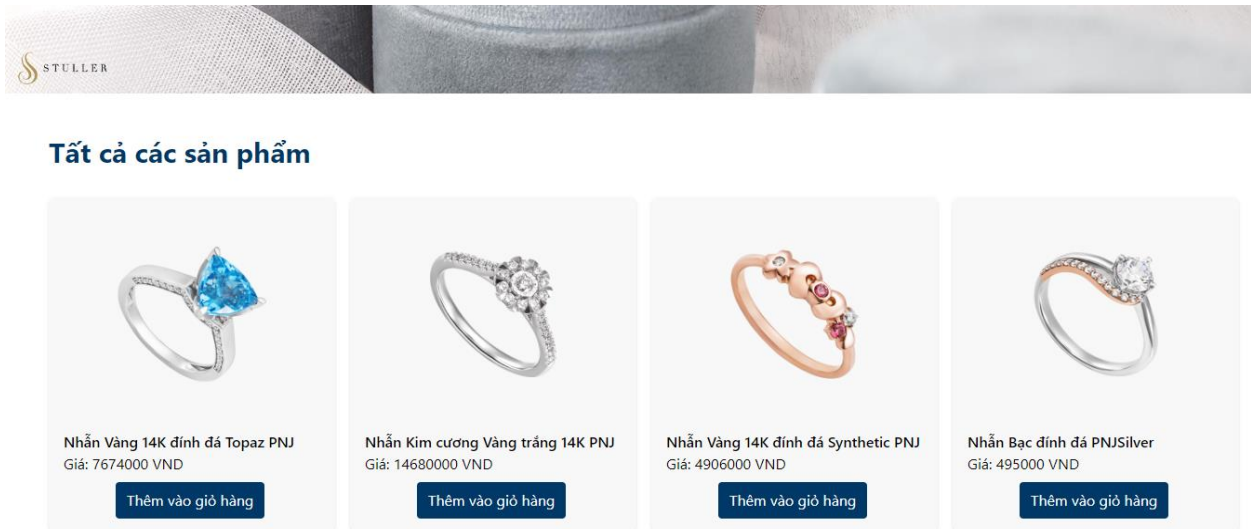
Cung cấp giao diện chính với các thông tin giới thiệu về website, các sản phẩm và điều hướng đến các trang khác như trang sản phẩm, trang đăng nhập, đăng ký.



Hình 3.5. Giao diện trang chủ

### 3.2.2. Trang sản phẩm

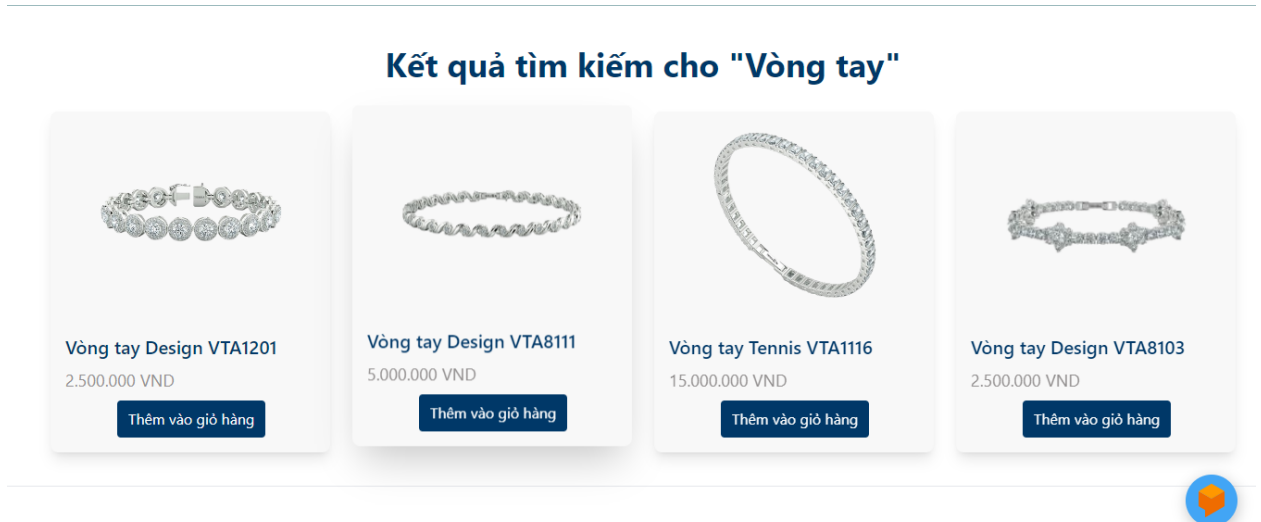
Hiện thị danh sách sản phẩm có sẵn trên website và cho phép khách hàng có thể thêm sản phẩm vào giỏ hàng.



Hình 3.6. Trang sản phẩm

### 3.2.3. Trang tìm kiếm sản phẩm

Khi người dùng muốn tìm một mặt hàng nào đó, chỉ cần gõ từ khoá vào ô tìm kiếm thì sẽ tìm thấy những sản phẩm có từ khoá đó.



Hình 3.7. Kết quả tìm kiếm

### 3.2.4. Trang đặt hàng

Giao diện cho phép người dùng chọn sản phẩm và thực hiện đặt hàng. Khách hàng có thể:

- Chọn sản phẩm, thêm vào giỏ hàng.
- Cập nhật số lượng sản phẩm trong giỏ hàng.
- Xác nhận đặt hàng.



#### Giỏ hàng

<p><b>Nhẫn Vàng 14K đính đá Topaz PNJ</b>  Đơn giá: 7674000 VND  Số lượng: <input type="text" value="1"/> <input type="button" value="+"/> <input type="button" value="-"/>  Xoá</p>	<p><b>Thông tin đặt hàng</b></p> <p>Họ và tên:  <input type="text" value="Xưa Đổ"/></p> <p>Địa chỉ nhận hàng:  <input type="text" value="bacgiang"/></p> <p>Số điện thoại:  <input type="text" value="0379722711"/></p> <p><input type="button" value="Đặt hàng"/></p>
<p><b>Nhẫn Kim cương Vàng trắng 14K PNJ</b>  Đơn giá: 14680000 VND  Số lượng: <input type="text" value="1"/> <input type="button" value="+"/> <input type="button" value="-"/>  Xoá</p>	

Tổng tiền: 22354000 VND

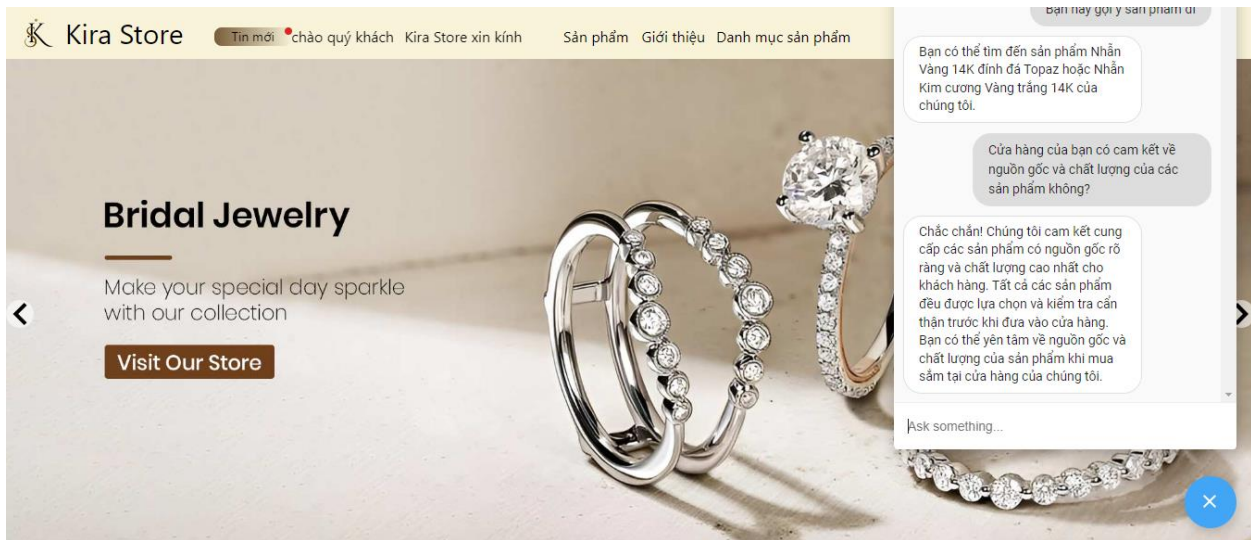
Hình 3.8. Giao diện đặt hàng

### 3.2.5. Chat với trợ lý ảo (AI)

Hỗ trợ người dùng thông qua giao tiếp với trợ lý AI để giải đáp thắc mắc và hỗ trợ mua sắm.

- Trả lời câu hỏi của khách hàng về sản phẩm, quy trình mua hàng, và các vấn đề khác liên quan đến dịch vụ.
- Hỗ trợ tìm kiếm sản phẩm và cung cấp thông tin theo yêu cầu của khách hàng.
- Hướng dẫn khách hàng thực hiện các thao tác trên website.



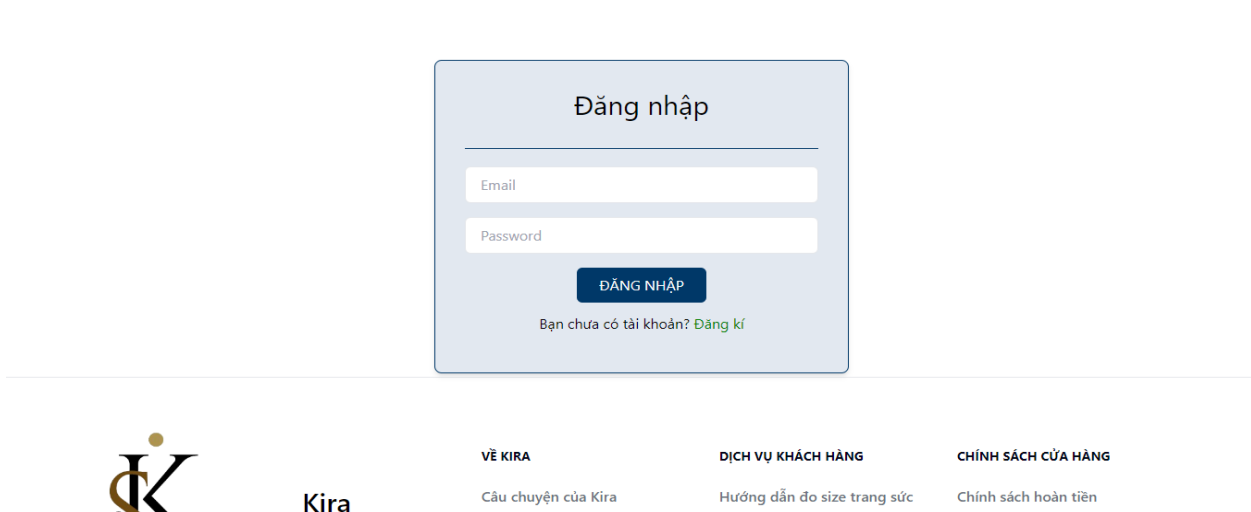


Hình 3.9. Giao diện chat với trợ lý AI

### 3.2.6. Đăng nhập

Giao diện cho phép người dùng hiện tại đăng nhập vào hệ thống.

- Nhập tên đăng nhập và mật khẩu.
- Xác thực thông tin người dùng và cho phép truy cập vào tài khoản cá nhân.

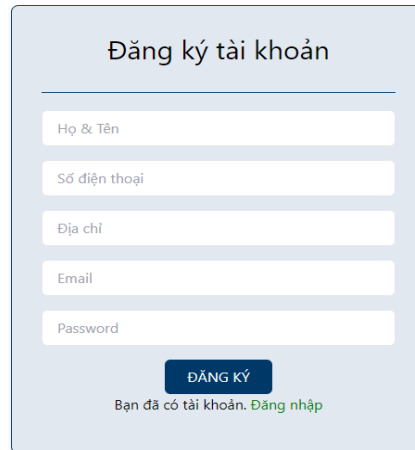


Hình 3.10. Giao diện Đăng nhập

### 3.2.7. Đăng ký tài khoản

Giao diện cho phép người dùng mới tạo tài khoản trên hệ thống.

- Nhập thông tin cá nhân bao gồm tên, email, mật khẩu, và các thông tin khác.
- Xác thực thông tin và tạo tài khoản mới.



Hình 3.11. Giao diện đăng ký tài khoản

### 3.2.8. Menu điều khiển của Admin

Giao diện chính cho quản trị viên quản lý toàn bộ hệ thống. Admin có thể truy cập các chức năng quản lý sản phẩm, loại sản phẩm, nhân viên, khách hàng, nhà cung cấp, đơn hàng, và thống kê báo cáo.



Hình 3.12. Giao diện Menu điều khiển của Admin

### 3.2.9. Quản lý sản phẩm

Hiện thị danh sách tất cả sản phẩm có sẵn trong hệ thống và điều hướng đến các chức năng thêm, sửa, xóa sản phẩm.

Mã sản phẩm	Tên sản phẩm	Ảnh sản phẩm	Mô tả	Số lượng	Đơn giá	Loại sản phẩm	Nhà cung cấp	Action
1	Nhẫn Vàng 14K đính đá Topaz PNU		Với sự kết hợp giữa sắc xanh tinh tế của đá Topaz cùng vẻ lấp lánh của những viên đá trắng, PNU mang đến tuyệt tác nhẫn vàng đá Topaz như những người phụ nữ kiêu hãnh cùng gu thời trang thời thượng của mình.	20	7674000	Nhẫn nữ	Công ty cổ phần Vàng Bạc Đá quý Phú Nhuận	
2	Nhẫn Kim cương Vàng trắng 14K PNU		Kim cương vốn là món trang sức mang đến niềm kiêu hãnh và cảm hứng thời trang bất tận. Sở hữu riêng cho mình món trang sức kim cương chính là điều mà ai cũng mong muốn. Chiếc nhẫn được chế tác từ vàng 14K cùng điểm nhấn kim cương với 57 giác cắt chuẩn xác, tạo nên món trang sức đầy sự sang trọng và đẳng cấp.	10	14680000	Nhẫn nữ	Công ty cổ phần Vàng Bạc Đá quý Phú Nhuận	
3	Nhẫn Vàng 14K đính đá Synthetic PNU		Với sự kết hợp giữa sắc xanh tinh tế của đá Topaz cùng vẻ lấp lánh của những viên đá trắng, PNU mang đến tuyệt tác nhẫn vàng đá Topaz như những người phụ nữ kiêu hãnh cùng gu thời trang thời thượng của mình.	20	4906000	Nhẫn nữ	Công ty cổ phần Vàng Bạc Đá quý Phú Nhuận	
4	Nhẫn Bạc đính đá PNUSilver		Với kiểu dáng thời thượng cùng những viên đá đính xung quanh bề mặt chiếc nhẫn trên chất liệu bạc 92.5, PNUSilver mang đến chiếc nhẫn với vẻ đẹp trẻ trung nhưng không kém phần phá cách, giúp các cô gái trông thật nổi bật	10	495000	Nhẫn nữ	Công ty cổ phần Vàng Bạc Đá quý Phú Nhuận	

Hình 3.13. Giao diện Quản lý sản phẩm

### 3.2.13. Quản lý danh mục sản phẩm

Hiện thị danh sách tất cả loại sản phẩm có sẵn trong hệ thống và điều hướng đến các chức năng thêm, sửa, xóa loại sản phẩm.

Mã loại	Tên loại sản phẩm	Mô tả loại sản phẩm	Ngày tạo	Hành động
1	Dây chuyền nữ	Mỗi chiếc dây chuyền nữ mang một câu chuyện riêng, phản ánh cá tính và phong cách	5/6/2024	
2	Lắc tay nữ	Lắc tay nữ, trong thế giới của trang sức, là một biểu tượng của sự tinh tế, phong cách và sự thanh lịch.	5/6/2024	
3	Lắc chân nữ	Lắc chân nữ là một loại trang sức tinh tế và quyền rũ được thiết kế để đeo quanh mắt cá chân	5/6/2024	
4	Nhẫn nữ	Những chiếc nhẫn nữ là biểu tượng của sự tinh tế và thanh lịch, với sự chăm chút tỉ mỉ trong từng đường nét và chi tiết.	5/6/2024	
5	Nhẫn đôi	Những chiếc nhẫn đôi không chỉ là một biểu tượng của tình yêu và sự cam kết, mà còn là biểu hiện của sự kết hợp hoàn hảo giữa vẻ đẹp và ý nghĩa.	5/6/2024	
6	Khuyên tai	Mỗi chiếc khuyên tai đều là một tác phẩm nghệ thuật riêng biệt, được tạo ra từ sự kết	5/6/2024	

Hình 3.14. Giao diện Danh sách loại sản phẩm

### 3.2.17. Quản lý nhân viên

Màn hình hiển thị danh sách và quản lý thông tin nhân viên.

- Xem danh sách nhân viên đang có trong hệ thống.
- Điều hướng đến các chức năng thêm, sửa, xóa nhân viên.

Mã nhân viên	Họ và tên	Địa chỉ	Giới tính	Email	Số điện thoại	Chức vụ	Mật khẩu	Hành động
9	Nguyễn Văn Tâm	Hà Nội	Nam	nguyenvana@gmail.com	0975224090	Nhân Viên	123456	Sửa  Xóa
10	Dương Xuân Tuấn	Việt Yên	Nam	duongxuantuan@gmail.com	04356273	Nhân Viên	123456	Sửa  Xóa
11	Trịnh Quốc	Hà Đông - Hà Nội	Nam	trinhquoc@gmail.com	0359678444	Nhân Viên	123456	Sửa  Xóa
12	Đỗ Thị Thanh Thuý	Việt Yên Bắc Giang	Nữ	dothuy@gmail.com	0379722733	Nhân Viên	123456	Sửa  Xóa

Hình 3.15. Giao diện Quản lý nhân viên

### 3.2.21. Quản lý thông tin khách hàng

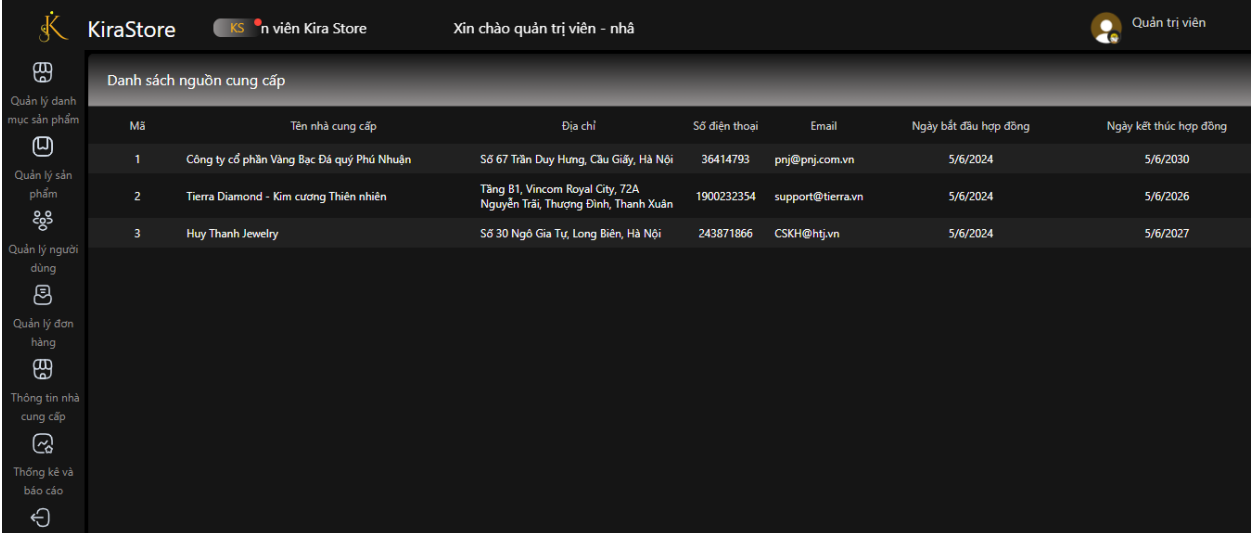
Màn hình hiển thị và quản lý thông tin khách hàng.

Mã nhà khách hàng	Tên khách hàng	Địa chỉ email	Số điện thoại
6	Đỗ Thị Xua	dothixua02@gmail.com	0345622788
7	Trịnh Lan Anh	lananh@gmail.com	0456733118

Hình 3.16. Giao diện Quản lý thông tin khách hàng

### 3.2.22. Quản lý thông tin nhà cung cấp

Hiển thị và quản lý thông tin nhà cung cấp.

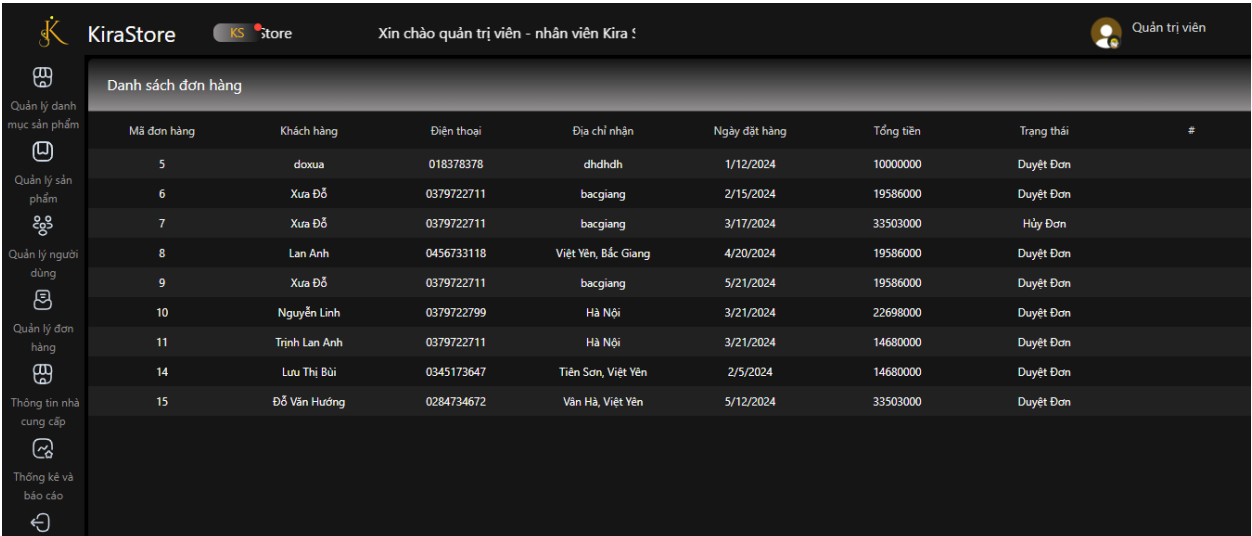


Mã	Tên nhà cung cấp	Địa chỉ	Số điện thoại	Email	Ngày bắt đầu hợp đồng	Ngày kết thúc hợp đồng
1	Công ty cổ phần Vàng Bạc Đá quý Phú Nhuận	Số 67 Trần Duy Hưng, Cầu Giấy, Hà Nội	36414793	pnj@pnj.com.vn	5/6/2024	5/6/2030
2	Tierra Diamond - Kim cương Thiên nhiên	Tầng B1, Vincom Royal City, 72A Nguyễn Trãi, Thượng Đình, Thanh Xuân	1900232354	support@tierra.vn	5/6/2024	5/6/2026
3	Huy Thanh Jewelry	Số 30 Ngô Gia Tự, Long Biên, Hà Nội	243871866	CSKH@htj.vn	5/6/2024	5/6/2027

Hình 3.17. Giao diện Quản lý thông tin nhà cung cấp

### 3.2.23. Quản lý đơn hàng

Hiển thị và quản lý các đơn hàng. Nhân viên và Admin đều có thể xem danh sách đơn hàng và cập nhật trạng thái đơn hàng (Huỷ đơn, Duyệt đơn).

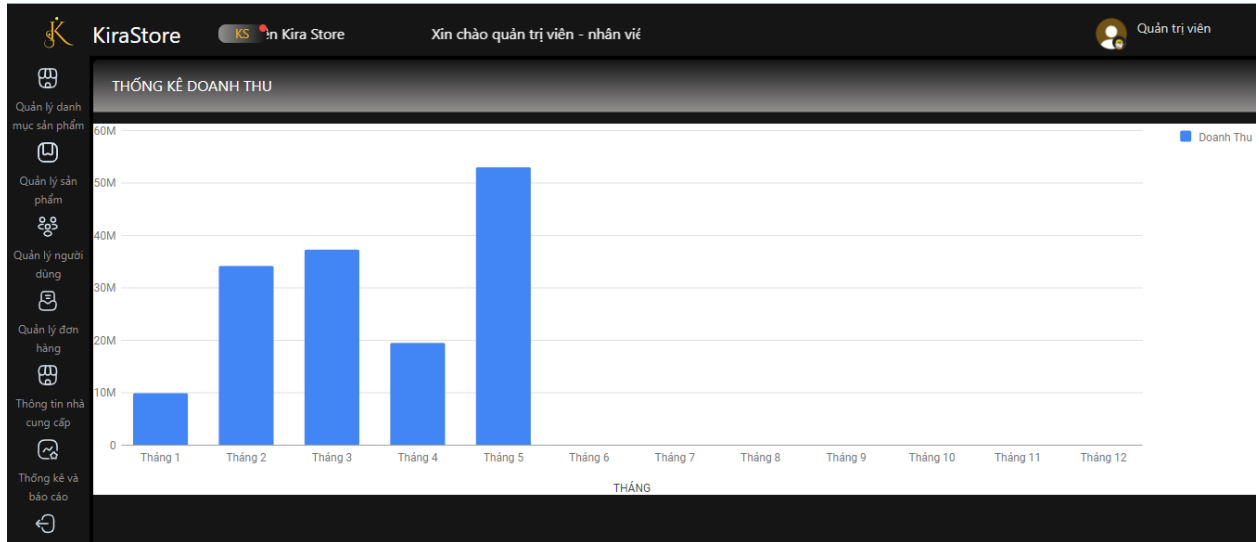


Mã đơn hàng	Khách hàng	Điện thoại	Địa chỉ nhận	Ngày đặt hàng	Tổng tiền	Trạng thái	#
5	doxua	018378378	dhhdh	1/12/2024	10000000	Duyệt Đơn	
6	Xua Đỏ	0379722711	bacgiang	2/15/2024	19586000	Duyệt Đơn	
7	Xua Đỏ	0379722711	bacgiang	3/17/2024	33503000	Hủy Đơn	
8	Lan Anh	0456733118	Việt Yên, Bắc Giang	4/20/2024	19586000	Duyệt Đơn	
9	Xua Đỏ	0379722711	bacgiang	5/21/2024	19586000	Duyệt Đơn	
10	Nguyễn Linh	0379722799	Hà Nội	3/21/2024	22698000	Duyệt Đơn	
11	Trịnh Lan Anh	0379722711	Hà Nội	3/21/2024	14680000	Duyệt Đơn	
14	Lưu Thị Bội	0345173647	Tiên Sơn, Việt Yên	2/5/2024	14680000	Duyệt Đơn	
15	Đỗ Văn Hướng	0284734672	Vân Hà, Việt Yên	5/12/2024	33503000	Duyệt Đơn	

Hình 3.18. Giao diện quản lý đơn hàng

### 3.2.24. Thống kê báo cáo

Màn hình thể hiện doanh thu theo các tháng trong năm của cửa hàng. Từ đây Admin có thể xem được doanh thu cụ thể các tháng là bao nhiêu.



Hình 3.19. Giao diện Thống kê báo cáo

### 3.3. Kết quả xây dựng các chức năng hệ thống

Sau quá trình nghiên cứu và thực hiện làm đồ án, hệ thống đã đạt được một số chức năng chính và đảm bảo rằng hệ thống hoạt động một cách hiệu quả, cung cấp các tính năng cần thiết cho cả người dùng và quản trị viên, từ việc mua sắm, quản lý sản phẩm đến các hoạt động quản trị hệ thống.

Chức năng cho người dùng: Truy cập trang chủ, xem và đặt hàng sản phẩm, tìm kiếm sản phẩm, chat với trợ lý AI, đăng nhập và đăng ký tài khoản.

Chức năng cho quản trị viên: Quản lý sản phẩm, loại sản phẩm, nhân viên, khách hàng, nhà cung cấp, đơn hàng và thực hiện thống kê báo cáo.

### Tiểu kết

Chương 3 đã trình bày chi tiết quá trình xây dựng, triển khai và tích hợp hệ thống. Đầu tiên, phần xây dựng và triển khai hệ thống được chia thành hai phần chính: FrontEnd và BackEnd, cùng với việc tích hợp giữa chúng với trợ lý ảo AI. Tiếp theo, chương này đã giới thiệu các kết quả xây dựng giao diện người dùng, bao gồm các trang chính như trang chủ, trang sản phẩm, trang tìm kiếm sản phẩm, trang đặt hàng, và các trang quản lý cho Admin như

quản lý sản phẩm, danh mục sản phẩm, nhân viên, khách hàng, nhà cung cấp, đơn hàng và thống kê doanh số. Cuối cùng, kết quả xây dựng các chức năng hệ thống đã được trình bày, cho thấy hệ thống đã hoàn thiện cả về mặt giao diện người dùng và các chức năng quản lý, đảm bảo đáp ứng đầy đủ các yêu cầu bài toán đặt ra ban đầu.

## KẾT LUẬN

Đồ án “XÂY DỰNG HỆ THỐNG BÁN ĐỒ TRANG SỨC CHO CỬA HÀNG KIRA STORE TRÊN NỀN TẢNG ASP.NET CORE WEB API VÀ REACTJS” đã đạt được các mục tiêu chính đề ra, bao gồm cung cấp một nền tảng thân thiện với người dùng cho việc mua sắm trực tuyến và một hệ thống quản trị hiệu quả cho quản trị viên. Hệ thống này bao gồm đầy đủ các chức năng từ quản lý sản phẩm, đơn hàng, nhân viên đến việc hỗ trợ khách hàng thông qua chat với trợ lý AI. Giao diện người dùng được thiết kế trực quan, dễ sử dụng và đáp ứng các yêu cầu cơ bản của người dùng và quản trị viên.

Một trong những ưu điểm nổi bật của hệ thống là giao diện người dùng thân thiện, giúp người dùng dễ dàng tìm kiếm và mua sắm sản phẩm. Hệ thống cung cấp đầy đủ các chức năng cần thiết cho cả người dùng và quản trị viên, từ quản lý sản phẩm, đơn hàng đến hỗ trợ khách hàng. Tính năng trợ lý AI hỗ trợ người dùng thông qua chat, giúp giải đáp thắc mắc và hướng dẫn mua sắm, từ đó nâng cao trải nghiệm người dùng. Bên cạnh đó, giao diện quản trị viên cung cấp các công cụ quản lý mạnh mẽ, giúp quản trị viên dễ dàng theo dõi và quản lý toàn bộ hệ thống.

Tuy nhiên, hệ thống vẫn còn tồn tại một số nhược điểm cần khắc phục. Đầu tiên, khi số lượng người dùng và sản phẩm tăng lên, hệ thống có thể gặp phải các vấn đề về hiệu suất. Do đó, cần tối ưu hóa cơ sở dữ liệu và mã nguồn để đảm bảo hệ thống hoạt động mượt mà. Cuối cùng, hiện tại hệ thống chưa có tính năng tích hợp mạnh mẽ với các nền tảng và dịch vụ bên ngoài như mạng xã hội, các dịch vụ thanh toán đa dạng và các hệ thống quản lý kho hàng.

Trong tương lai, để nâng cao hiệu quả hoạt động, hệ thống cần tập trung vào một số hướng phát triển quan trọng. Trước tiên là cải thiện bảo mật bằng cách tăng cường các biện pháp như mã hóa dữ liệu, xác thực hai yếu tố và kiểm tra bảo mật định kỳ để đảm bảo an toàn cho thông tin người dùng và hệ thống. Ngoài ra, phát triển tính năng tích hợp với các dịch vụ thanh toán đa dạng, mạng xã hội và các hệ thống quản lý kho hàng sẽ tăng tính linh hoạt và tiện dụng của hệ thống.

Tóm lại, hệ thống bán đồ trang sức cho cửa hàng Kira Store trên nền tảng ASP.NET Core Web Api và ReactJS quản lý và bán hàng trực tuyến đã hoàn thành tốt các mục tiêu đề ra, cung cấp một nền tảng vững chắc cho việc mua sắm và quản lý sản phẩm. Mặc dù còn một



số nhược điểm cần khắc phục, nhưng với các kế hoạch phát triển trong tương lai, hệ thống sẽ ngày càng hoàn thiện và đáp ứng tốt hơn nhu cầu của người dùng và quản trị viên. Việc liên tục cải tiến và cập nhật sẽ đảm bảo rằng hệ thống luôn ở mức độ hiện đại và hiệu quả, giúp doanh nghiệp phát triển mạnh mẽ trong thị trường trực tuyến ngày càng cạnh tranh.

## DANH MỤC TÀI LIỆU THAM KHẢO

### Danh mục các Website tham khảo:

[1] <https://vneconomy.vn/du-suc-mua-con-yeu-nganh-hang-xa-xi-lac-quan-ve-nam-2024.htm> (truy cập ngày 05/04/2024).

[2] <https://viettelidc.com.vn/tin-tuc/toan-tap-ve-sql-server-cho-nguoi-moi-bat-dau> (truy cập ngày 07/04/2024)

[3] <https://dialogflowvn.wordpress.com/2017/11/18/dialogflow-cac-khai-niem/> (truy cập ngày 10/04/2024)

[4] <https://respond.io/vi/blog/dialogflow-tutorial> (truy cập ngày 10/04/2024)

[5] <https://gcs.vn/google-dialogflow-la-gi/> (truy cập ngày 10/04/2024)