**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐẶT ĐỒ ĂN NHANH THEO THỜI GIAN THỰC**

|  |  |
| --- | --- |
| CBHD: | Ts. Đặng Trọng Hợp |
| Sinh viên: | Nguyễn Tuấn Kiệt |
| Mã sinh viên: | 2021600870 |

**Hà Nội, 2024**

**LỜI CẢM ƠN**

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè đã giúp em hoàn thiện đồ án tốt nghiệp.

Trước tiên em xin gửi tới các thầy cô khoa Công nghệ thông tin trường Đại học Công Nghiệp Hà Nội lời chúc sức khỏe và lời cảm ơn chân thành. Với sự quan tâm, chỉ bảo của thầy cô đã tạo điều kiện cho em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp lần này.

Đặc biệt em xin gửi lời cảm ơn chân thành nhất đến thầy Đặng Trọng Hợp đã trực tiếp tận tình hướng dẫn cũng như nhận xét và giúp đỡ em trong suốt quá trình hoàn thiện đồ án.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không tránh khỏi những thiếu sót. Em rất mong nhận được sự giúp đỡ của thầy cô và các bạn để hoàn thiện hơn đồ án của mình.

Em xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng năm 2024  Sinh viên thực hiện |

**Mục lục**

[Lời mở đầu 1](#_Toc185036694)

[Danh mục hình ảnh 2](#_Toc185036695)

[CHƯƠNG 1. TỔNG QUAN 3](#_Toc185036696)

[1.1. Lý do chọn đề tài 3](#_Toc185036697)

[1.2. Đặt vấn đề 3](#_Toc185036698)

[1.3. Ý nghĩa và cơ sở lý luận 4](#_Toc185036699)

[1.3.1. Ý nghĩa của đề tài 4](#_Toc185036700)

[1.3.2. Cơ sở lý luận của đề tài 4](#_Toc185036701)

[1.4. Mục tiêu của đồ án 5](#_Toc185036702)

[1.5. Giới hạn và phạm vi của đồ án 5](#_Toc185036703)

[1.6. Kết quả đạt được 5](#_Toc185036704)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 7](#_Toc185036705)

[2.1. Các tác nhân của hệ thống 7](#_Toc185036706)

[2.2. Hoạt động của hệ thống 7](#_Toc185036707)

[2.2.1. Hoạt động đặt và giao nhận đồ ăn 7](#_Toc185036708)

[2.2.2. Báo cáo, thống kê 7](#_Toc185036709)

[2.2.3. Cập nhật thông tin hệ thống 7](#_Toc185036710)

[2.2.4. Hoạt động thời gian thực 8](#_Toc185036711)

[2.3. Các yêu cầu chức năng 8](#_Toc185036712)

[2.3.1. Phía người dùng: 8](#_Toc185036713)

[2.3.2. Phía nhân viên: 8](#_Toc185036714)

[2.3.3. Phía chủ cửa hàng: 8](#_Toc185036715)

[2.4. Biểu đồ use case 9](#_Toc185036716)

[2.4.1. Biểu đồ use case tổng quát 9](#_Toc185036717)

[2.4.2. Biểu đồ phân rã use case cho người dùng 10](#_Toc185036718)

[2.4.3. Biểu đồ phân rã use case cho nhân viên và chủ cửa hàng 12](#_Toc185036719)

[2.5. Biểu đồ thực thể liên kết mức vật lý 14](#_Toc185036720)

[2.6. Cơ sở dữ liệu 15](#_Toc185036721)

[CHƯƠNG 3: PHÂN TÍCH USE CASE 17](#_Toc185036722)

[3.1. Mô tả chi tiết các use case 17](#_Toc185036723)

[3.1.1. Use case Đăng nhập 17](#_Toc185036724)

[3.1.2. Use case Đăng ký 18](#_Toc185036725)

[3.1.3. Use case Quản lý giỏ hàng 19](#_Toc185036726)

[3.1.4. Use case xem chi tiết sản phẩm 22](#_Toc185036727)

[3.1.5. Use case đánh giá và phản hồi 22](#_Toc185036728)

[3.1.6. Use case quản lý thông tin cá nhân 23](#_Toc185036729)

[3.1.7. Use case Quản lý danh sách yêu thích 24](#_Toc185036730)

[3.1.8.Use case Xem thông tin cửa hàng 26](#_Toc185036731)

[3.1.9. Use case Tìm kiếm món ăn 27](#_Toc185036732)

[3.1.10. Use case đặt hàng 28](#_Toc185036733)

[3.1.11. Use case theo dõi đơn hàng 28](#_Toc185036734)

[3.1.12. Use case Quản lý đơn hàng theo trạng thái 29](#_Toc185036735)

[3.1.13. Use case Quản lý danh mục 30](#_Toc185036736)

[3.1.14. Use case Quản lý sản phẩm 33](#_Toc185036737)

[3.2. Phân tích use case 36](#_Toc185036738)

[3.1.1. Use case đăng nhập 36](#_Toc185036739)

[3.1.2. Use case Đăng ký 38](#_Toc185036740)

[3.1.3. Use case Quản lý thông tin cá nhân 40](#_Toc185036741)

[3.1.4. Quản lý giỏ hàng 42](#_Toc185036742)

[3.1.5. Quản lý đơn hàng theo trạng thái 43](#_Toc185036743)

[3.1.6. Xem chi tiết đơn hàng 45](#_Toc185036744)

[3.1.7. Quản lý danh mục 47](#_Toc185036745)

[3.1.8. Quản lý sản phẩm 49](#_Toc185036746)

[Chương 4: TRIỂN KHAI VÀ KIỂM THỬ PHẦN MỀM 51](#_Toc185036747)

[4.1. Lựa chọn công nghệ triển khai 51](#_Toc185036748)

[4.1.2. Flask 51](#_Toc185036749)

[4.1.2. MySQL 52](#_Toc185036750)

[4.1.3. SocketIO 52](#_Toc185036751)

[4.1.4. Streamlit 53](#_Toc185036752)

[4.1.5. Quản lý phiên bản (Version Control System - VCS): 53](#_Toc185036753)

[4.1.6. Android (Kotlin) 53](#_Toc185036754)

[4.1.7. Docker 54](#_Toc185036755)

[4.1.8. MongoDB 54](#_Toc185036756)

[4.2. Cài đặt chương trình 55](#_Toc185036757)

[4.2.1. Yêu cầu hệ thống 55](#_Toc185036758)

[4.2.2. Cài đặt chương trình 56](#_Toc185036759)

[4.3. Kiểm thử phần mềm 61](#_Toc185036760)

[4.3.1. Kiểm thử Unit 61](#_Toc185036761)

[4.3.2. Kiểm thử API 63](#_Toc185036762)

[Chương 4. Kết luận 64](#_Toc185036763)

[4.1. Tóm tắt kết quả nghiên cứu: 64](#_Toc185036764)

[4.2. Đánh giá với mục tiêu ban đầu: 64](#_Toc185036765)

[4.3. Đề xuất hướng phát triển tương lai 65](#_Toc185036766)

[Tài liệu tham khảo 66](#_Toc185036767)

# Lời mở đầu

Trong kỷ nguyên công nghệ số, cùng với việc phát triển thần tốc của thương mại điện tử hành vi tiêu dùng của con người đã có nhiều sự đổi mới. Đặc biệt nhu cầu đặt đồ ăn trực tuyến ngày càng tăng cao đòi hỏi các nhà hàng, quán ăn cũng phải có sự thay đổi để bắt kịp với sự thay đổi của thời đại, quán ăn phải tìm kiếm các giải pháp hiệu quả để đáp ứng nhu cầu đa dạng và nhanh chóng của khách hàng. Hệ thống đặt đồ ăn theo thời gian thực là cầu trả lời cho những yêu cầu đó.

Đề tài này nhằm mục tiêu giới thiệu và nghiên cứu về việc xây dựng một hệ thống đặt đồ ăn trực tuyến hoàn chỉnh cho các cửa hàng. Chúng ta sẽ tìm hiểu về những thách thức và cơ hội mà các nhà hàng hay hệ thống hiện có còn đang mắc phải, và cách một hệ thống có thể giúp cải thiện quá trình này. Bằng cách tận dụng công nghệ và các công cụ phần mềm hiện đại, chúng ta có thể tối ưu hóa quy trình đặt đồ ăn, từ việc xem các món ăn, đặt hàng và giao hàng đến thống kê và báo cáo để cải thiện chất lượng cho cửa hàng.

Trong các phần tiếp theo, chúng ta sẽ xem xét sâu hơn về sự quan trọng của hệ thống đặt đồ ăn trực tuyến và hiệu quả của nó mang lại cho cả nhà hàng, và khách hàng.

# Danh mục hình ảnh

[Hình 2. 1 Biểu đồ use case tổng quát 9](#_heading=h.1jlao46)

[Hình 2. 2 Biểu đồ phân rã use case cho người dùng 10](#_heading=h.2iq8gzs)

[Hình 2. 3 Biểu đồ phân rã use case cho nhân viên và chủ cửa hàng 12](#_heading=h.1yyy98l)

[Hình 2. 4 Biểu đồ thực thể liên kết mức vật lý 14](#_heading=h.2ce457m)

[Hình 2. 5 Cơ sở dữ liệu 15](#_heading=h.1qoc8b1)

[Hình 3. 1 Luồng cơ bản use case đăng nhập 36](#_heading=h.1kc7wiv)

[Hình 3. 2 Biểu đồ lớp phân tích use case đăng nhập 37](#_heading=h.44bvf6o)

[Hình 3. 3 3 Luồng cơ bản use case đăng ký 38](#_heading=h.2jh5peh)

[Hình 3. 4 Biểu đồ lớp phân tích use case đăng ký 39](#_heading=h.ymfzma)

[Hình 3. 5 Luồng cơ bản use case quản lý thông tin cá nhân 40](#_heading=h.3im3ia3)

[Hình 3. 6 Biểu đồ lớp phân tích use case quản lý thông tin cá nhân 41](#_heading=h.1xrdshw)

[Hình 3. 7 Luồng cơ bản use case quản lý giỏ hàng 42](#_heading=h.4hr1b5p)

[Hình 3. 8 Biểu đồ lớp phân tích use case quản lý giỏ hàng 43](#_heading=h.2wwbldi)

[Hình 3. 9 Luồng cơ bản use case quản lý đơn hàng theo trạng thái 43](#_heading=h.1c1lvlb)

[Hình 3. 10 Biểu đồ lớp phân tích use case quản lý đơn hàng theo trạng thái 44](#_heading=h.3w19e94)

[Hình 3. 11 Luồng cơ bản use case xem chi tiết đơn hàng 45](#_heading=h.2b6jogx)

[Hình 3. 12 Biểu đồ lớp phân tích use case xem chi tiết đơn hàng 46](#_heading=h.qbtyoq)

[Hình 3. 13 Luồng cơ bản use case quản lý danh mục 47](#_heading=h.3abhhcj)

[Hình 3. 14 Biểu đồ lớp phân tích use case quản lý danh mục 48](#_heading=h.49gfa85)

[Hình 3. 15 Luồng cơ bản use case quản lý sản phẩm 49](#_heading=h.2olpkfy)

[Hình 3. 16 Biểu đồ lớp phân tích use case quản lý sản phẩm 50](#_heading=h.13qzunr)

[Hình 4. 1 Mô hình Clean Architecture 51](#_heading=h.320vgez)

[Hình 4. 2 Mô hình MVVM 53](#_heading=h.1ulbmlt)

[Hình 4. 3 Cấu trúc thư mục test unit 61](#_heading=h.1maplo9)

[Hình 4. 4 Cấu trúc thư mục test Socket 62](#_heading=h.46ad4c2)

[Hình 4. 5 Câu lệnh chạy test unit 62](#_heading=h.2lfnejv)

[Hình 4. 6 Triển khai test Socket thông qua file ipynb 6](#_heading=h.10kxoro)2

[Hình 4. 7 Triển khai test API thông qua postman 63](#_heading=h.1zpvhna)

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Lý do chọn đề tài

Ngày nay, Công nghệ thông tin có vai trò hết sức quan trọng, là động lực thúc đẩy tăng trưởng kinh tế và kích thích đổi mới nền kinh tế toàn cầu nói chung và nền kinh tế nước ta nói riêng. Bước vào thời đại công nghiệp 4.0, công nghệ thông tin đã có các cải tiến vượt bậc. Ngày nay, các thiết bị điện tử đã không còn là những phương tiện quý hiếm, mà ngược lại đã trở thành phương tiện giải trí, công cụ làm việc tiện lợi của con người mọi lúc, mọi nơi. Việc phổ thông hóa các thiết bị điện tử đã mở ra cho nhân loại một kỷ nguyên mới – kỷ nguyên công nghệ thông tin.

Song song với sự phát triển vượt trội của ngành Công nghệ thông tin là sự phát triển của thương mại điện tử đặc biệt là trong bối cảnh đại dịch COVID-19, việc giãn cách xã hội thúc đẩy sự bùng nổ của hệ thống đặt đồ ăn trực tuyến. Hệ thống đặt đồ ăn trực tuyến làm tăng nhu cầu về sự tiện lợi do sự nhanh chóng và đơn giản của nó, tạo thói quen mới cho cộng đồng trong nhu cầu ăn uống.

Với tư cách là một sinh viên của khoa Công nghệ thông tin việc áp dụng công nghệ vào việc đặt đồ ăn trực tuyến để thấy được nhiều lợi ích nhờ việc áp dụng công nghệ thông tin vào thực tế. Chính vì vậy, em đã quyết định lựa chọn đề tài **“Xây dựng hệ thống đặt đồ ăn nhanh theo thời gian thực”**để thực hiện đồ án của mình với mục đích nghiên cứu hệ thống đặt đồ ăn nhanh, giúp cho việc đặt đồ được thuận tiện hơn nhờ việc áp dụng công nghệ thông tin. Việc phát triển một hệ thống đặt đồ ăn theo thời gian thực không chỉ đáp ứng nhu cầu thực tế mà còn bắt kịp xu hướng chuyển đổi số toàn cầu, giúp cho cửa hàng, khách hàng và cả nhân viên có sự kết nối thuận tiện và hiệu quả.

## 1.2. Đặt vấn đề

Mức sống của người dân Việt Nam ngày càng tăng cao, đi kèm với đó là những bộn bề nên việc đặt đồ ăn đã không trở nên quá xa lạ với nhiều người, đặc biệt với đối tượng là các nhân viên văn phòng, những người không có quá nhiều thời gian cho việc chuẩn bị cho mình một bữa ăn. Nhận biết được nhu cầu đó, trong những năm gần đây, một loạt các hệ thống đặt đồ ăn nhanh… ra đời để đáp ứng nhu cầu của người tiêu dùng.

Các hệ thống đặt đồ ăn nhanh sẽ giải quyết được các vấn để về sự minh bạch trong giá cả và thông tin của món ăn, bên cạnh đó là việc đa dạng các món ăn giúp người dùng có nhiều sự lựa chọn và so sánh giữa các món ăn. Bên cạnh đó là việc mở rộng khu vực kết nối giữa các hàng quán với người tiêu dùng khi khoảng cách vật lý hay vị trí địa lý đã không còn là trở ngại.

Tuy nhiên bên cạnh đó thì các hệ thống đặt đồ ăn hiện tại đang phục vụ cho quá nhiều cửa hàng gây ra sự chậm trễ trong thông tin cho người tiêu dùng hay cửa hàng. Đó cũng là ý tưởng cho đồ án của tôi được thai nghén và triển khai để giúp những cửa hàng đã có tệp khách hàng cố định được trải nghiệm dịch vụ tốt hơn với tốc độ nhanh hơn.

## 1.3. Ý nghĩa và cơ sở lý luận

### 1.3.1. Ý nghĩa của đề tài

Nâng cao hiệu suất quản lý: Giúp tối ưu hóa quản lý đơn hàng, từ lúc đặt hàng đến khi nhận hàng, giảm thiểu sai sót trong vận hành.

Tiết kiệm thời gian và nguồn lực: Tự động hóa quy trình giao nhận hàng cho nhân viên, tiết kiệm thời gian quản lý cho cửa hàng.

Cải thiện trải nghiệm của khách hàng: Cung cấp giao diện dễ sử dụng, giúp khách hàng theo dõi đơn hàng theo thời gian thực và nhận các thông báo kịp thời.

Tạo giá trị cho cửa hàng: nâng cao hiệu quả kinh doanh, tăng sự hài lòng của khách hàng và tạo lợi thế cạnh tranh cho cửa hàng trên thị trường.

### 1.3.2. Cơ sở lý luận của đề tài

**Quản lý đơn hàng và dịch vụ ăn uống:** Đề tài dựa trên cơ sở lý luận về quản lý đơn hàng và dịch vụ ăn uống, tập trung vào việc tối ưu hóa quy trình đặt hàng, xử lý và giao nhận trong môi trường kinh doanh thực tế. Các nguyên tắc như quản lý thời gian thực, phân phối tài nguyên và tăng cường trải nghiệm khách hàng sẽ được tích hợp vào hệ thống.

**Công nghệ thông tin và phát triển phần mềm:** Đề tài áp dụng kiến thức về công nghệ thông tin và phát triển phần mềm để xây dựng một hệ thống đặt đồ ăn trực tuyến hiệu quả. Các công nghệ như ngôn ngữ lập trình, cơ sở dữ liệu, API thời gian thực (real-time API) và ứng dụng di động sẽ được triển khai để đảm bảo tính tiện ích, khả năng tích hợp và khả năng mở rộng.

## 1.4. Mục tiêu của đồ án

1. Tìm hiểu các công nghệ: Android, Flask, Socket, Streamlit, MySQL, MongoDB, Docker, Server …
2. Tìm hiểu cấu trúc chung của hệ thống đặt đồ ăn nhanh.
3. Tìm hiểu các chức năng của một hệ thống đặt đồ ăn nhanh.
4. Đưa ra phân tích thiết kế cho hệ thống đặt đồ ăn.
5. Xây dựng được hệ thống đặt đồ ăn nhanh, hệ thống quản trị bằng cách sử dụng các công nghệ lập trình web, android, cơ sở dữ liệu và quản trị cơ sở dữ liệu.

## 1.5. Giới hạn và phạm vi của đồ án

Trong phạm vi đề tài này em sẽ nghiên cứu các vấn đề:

Nghiên cứu, khảo sát, thu thập dữ liệu về quy trình đặt đồ ăn của một hệ thống đặt đồ ăn trên thị trường, đi sâu vào nghiên cứu và phân tích một hệ thống thông tin để xây dựng hệ thống hoàn chỉnh.

Hệ thống được xây dựng bằng sử dụng các công nghệ lập trình web, android và cơ sở dữ liệu.

Kiểm thử và phát triển hệ thống.

Triển khai hệ thống trên môi trường thực tế.

## 1.6. Kết quả đạt được

Hệ thống đặt đồ ăn khi hoàn thành dự kiến đạt được các kết quả sau:

Phân tích, thiết kế chức năng của hệ thống cho từng vai trò như: đăng nhập, đăng ký tài khoản, xem thông tin sản phẩm, đặt hàng, giao hàng, …

Thiết kế cơ sở dữ liệu.

Hệ thống được xây dựng với các chức năng riêng biệt cho từng vai trò của người dùng trong hệ thống:

* Chủ cửa hàng: Xem báo cáo, thống kê, quản lý thông tin nhân viên, …
* Nhân viên: Quản lý trạng thái sản phẩm, số lượng sản phẩm đã làm,...
* Khách hàng: Xem sản phẩm theo thư mục, đặt hàng, xem trạng thái đơn hàng,…

Cơ sở dữ liệu sử dụng MySQL được quản lý bằng PHP Admin.

Sử dụng MongoDB tăng tốc độ cho việc phân tích dữ liệu phía admin.

Xây dựng kịch bản kiểm thử cho API.

Người sử dụng hệ thống: Chủ cửa hàng, Nhân viên bán hàng, Quản trị hệ thống, khách hàng.

Hoàn thành báo cáo chi tiết đồ án tốt nghiệp.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 2.1. Các tác nhân của hệ thống

* Tác nhân khách hàng: là những người khách hàng ghé thăm và đặt hàng,...
* Tác nhân nhân viên của cửa hàng : là người tham gia cập nhật trạng thái các đơn hàng cùng với xem kết quả công việc theo ngày tháng năm.
* Tác nhân chủ cửa hàng: là người có thể xem các báo cáo thống kê bán hàng, quản lý nhân viên,...
* Tác nhân người quản trị (Admin): tác nhân tham gia quản lý hệ thống và toàn quyền quản lý database của hệ thống.

## 2.2. Hoạt động của hệ thống

### 2.2.1. Hoạt động đặt và giao nhận đồ ăn

* Khách hàng: Có thể xem danh mục món ăn, thêm các sản phẩm vào giỏ hàng, danh sách yêu thích. Hoạt động đặt hàng thông qua giỏ hàng và theo dõi trạng thái đơn hàng, xác nhận đơn hàng và đánh giá.
* Nhân viên: Tiếp nhận đơn hàng trong thời gian thực, xác nhận đơn hàng và giao hàng, cập nhật trạng thái của đơn hàng.

### 2.2.2. Báo cáo, thống kê

* Chủ cửa hàng: Có thể xem thống kế doanh thu theo ngày tháng năm, hệ thống sẽ đưa ra các báo cáo liên quan đến doanh thu, sản phẩm bán chạy, nhân viên chăm chỉ nhất,...
* Nhân viên: Có thể xem thống kê số lượng đơn hàng đã làm, đã giao theo ngày tháng năm cụ thể.

### 2.2.3. Cập nhật thông tin hệ thống

* Nhân viên: Cập nhật thông tin trạng thái các đơn hàng.
* Chủ cửa hàng: Quản lý các thông tin liên quan đến sản phẩm, thư mục, nhân viên,...
* Người dùng: Quản lý thông tin các nhân, giỏ hàng và danh sách sản phẩm yêu thích.

### 2.2.4. Hoạt động thời gian thực

* Đồng bộ trạng thái đơn hàng giữa khách hàng và nhân viên.
* Gửi thông báo khi có thay đổi trong đơn hàng
* Theo dõi vị trí đơn hàng trong quá trình vận chuyển trên map.

## 2.3. Các yêu cầu chức năng

### 2.3.1. Phía người dùng:

* Đăng ký, đăng nhập, đăng xuất
* Quản lý thông tin cá nhân
* Tìm kiếm và xem món ăn
* Quản lý giỏ hàng
* Quản lý danh sách yêu thích
* Đặt và theo dõi đơn hàng
* Đánh giá và phản hồi
* Xem thông tin cửa hàng

### 2.3.2. Phía nhân viên:

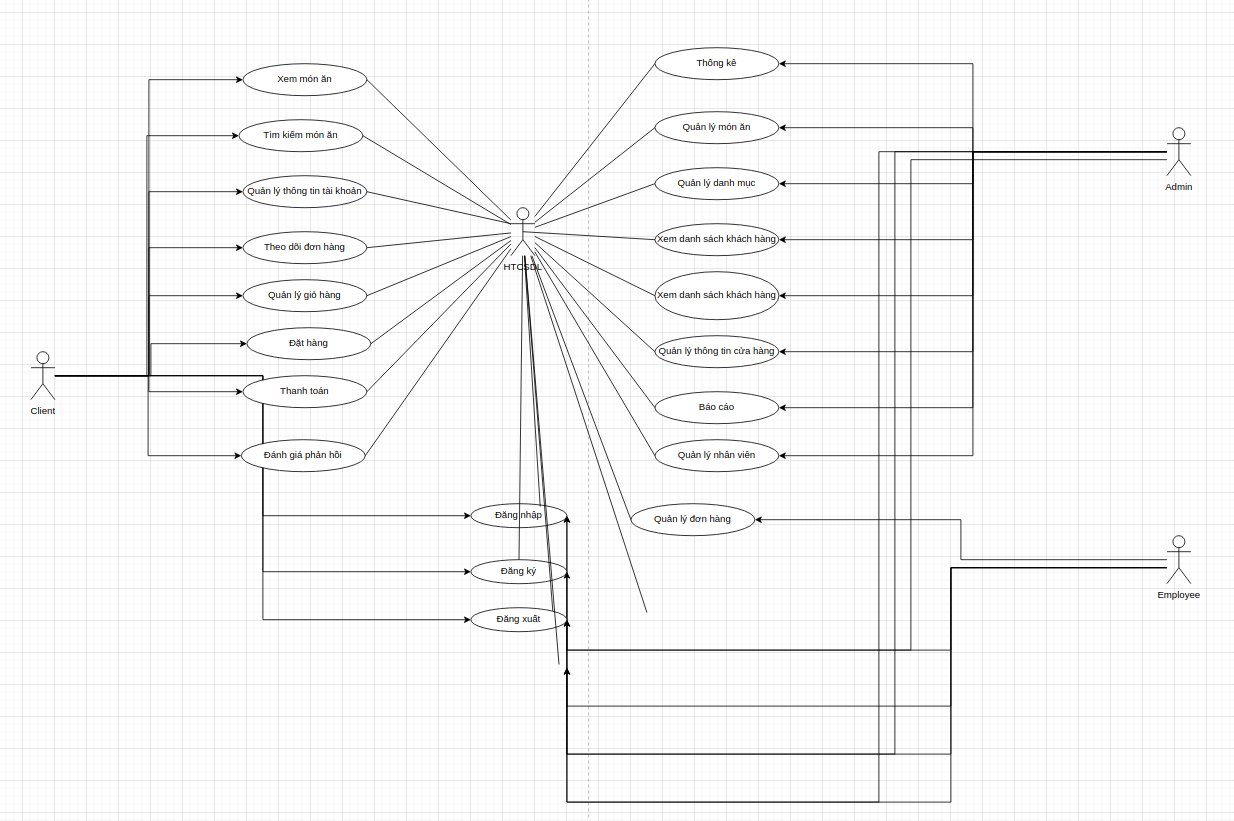
* Đăng ký, đăng nhập
* Quản lý đơn hàng theo trạng thái
* Xem số lượng đơn đã làm và đã giao
* Xem chi tiết đơn hàng

### 2.3.3. Phía chủ cửa hàng:

* Đăng nhập
* Quản lý danh mục
* Quản lý món ăn
* Xem thống kê khách hàng
* Quản lý nhân viên
* Báo cáo và thống kê
* Quản lý thông tin cửa hàng

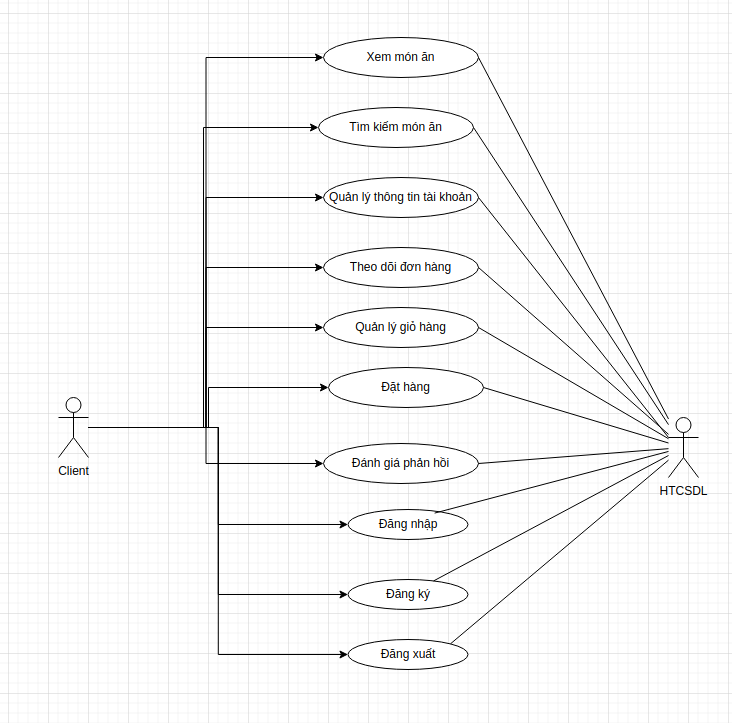
## 2.4. Biểu đồ use case

### 2.4.1. Biểu đồ use case tổng quát



*Hình 2. 1 Biểu đồ use case tổng quát*

### 2.4.2. Biểu đồ phân rã use case cho người dùng



*Hình 2. 2 Biểu đồ phân rã use case cho người dùng*

Use Case đăng nhập cho phép khách hàng truy cập hệ thống bằng tài khoản đã tạo.

Use Case đăng ký hỗ trợ khách hàng tạo tài khoản mới bằng cách nhập thông tin cá nhân như tên, email, số điện thoại và mật khẩu. Sau khi hoàn tất phiên làm việc, khách hàng có thể sử dụng

Use Case đăng xuất dùng để thoát khỏi hệ thống một cách an toàn.

Use Case xem món ăn để xem danh sách các món ăn với thông tin chi tiết như tên, hình ảnh, giá cả và mô tả.

Use Case tìm kiếm món ăn hỗ trợ tìm kiếm bằng từ khóa.

Use Case quản lý thông tin tài khoản cho phép khách hàng xem và chỉnh sửa thông tin cá nhân, bao gồm cập nhật tên, email, số điện thoại,...

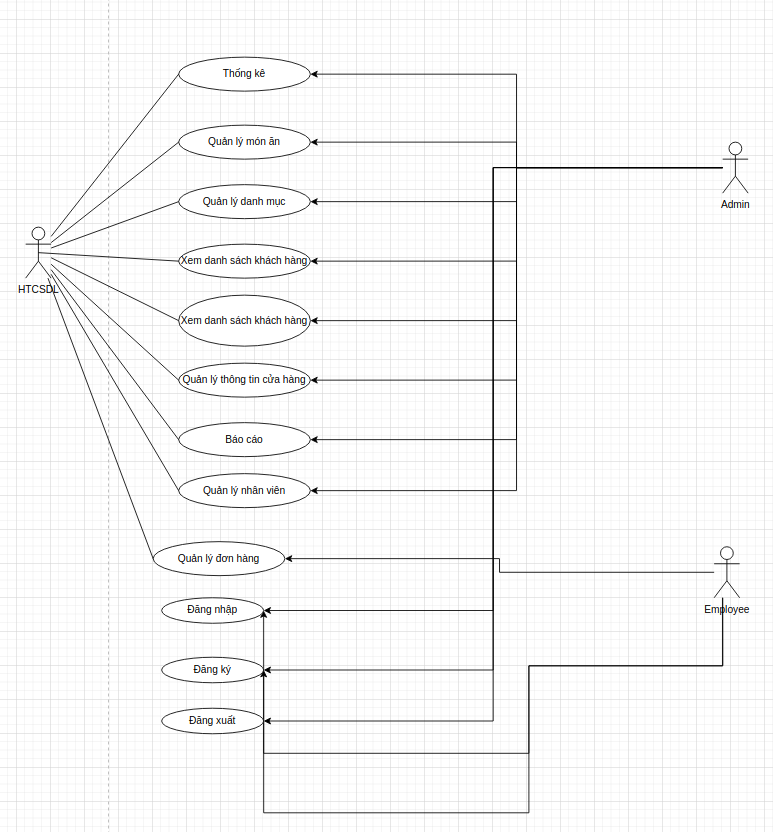
Use Case quản lý giỏ hàng, khách hàng có thể thêm, xóa hoặc chỉnh sửa các món ăn đã chọn trước khi đặt hàng.

Use Case đặt hàng giúp khách hàng xác nhận và thanh toán đơn hàng

Use Case theo dõi đơn hàng cho phép theo dõi trạng thái đơn hàng theo thời gian thực, từ khi chuẩn bị đến khi giao hàng thành công.

Use Case đánh giá phản hồi để đánh giá món ăn hoặc dịch vụ, để lại điểm số và bình luận về chất lượng sản phẩm hoặc trải nghiệm dịch vụ.

### 2.4.3. Biểu đồ phân rã use case cho nhân viên và chủ cửa hàng



*Hình 2. 3 Biểu đồ phân rã use case cho nhân viên và chủ cửa hàng*

Use Case đăng nhập cho phép khách hàng truy cập hệ thống bằng tài khoản đã tạo.

Use Case đăng ký hỗ trợ khách hàng tạo tài khoản mới bằng cách nhập thông tin cá nhân như tên, email, số điện thoại và mật khẩu. Sau khi hoàn tất phiên làm việc, khách hàng có thể sử dụng

Use Case đăng xuất dùng để thoát khỏi hệ thống một cách an toàn.

Use Case thống kê cung cấp báo cáo doanh số và dữ liệu bán hàng, giúp chủ cửa hàng theo dõi hiệu quả kinh doanh theo ngày, tuần, hoặc tháng.

Use Case quản lý món ăn cho phép thêm, sửa, xóa các món ăn, cập nhật thông tin về giá cả, hình ảnh và mô tả.

Use Case quản lý danh mục hỗ trợ tổ chức các món ăn theo từng danh mục như đồ uống, món chính, hay món tráng miệng.

Use Case xem danh sách nhân viên cho phép chủ cửa hàng theo dõi thông tin cá nhân và các hoạt động liên quan đến nhân viên.

Use Case xem danh sách khách hàng cho phép chủ cửa hàng theo dõi thông tin cá nhân và các hoạt động liên quan đến khách hàng.

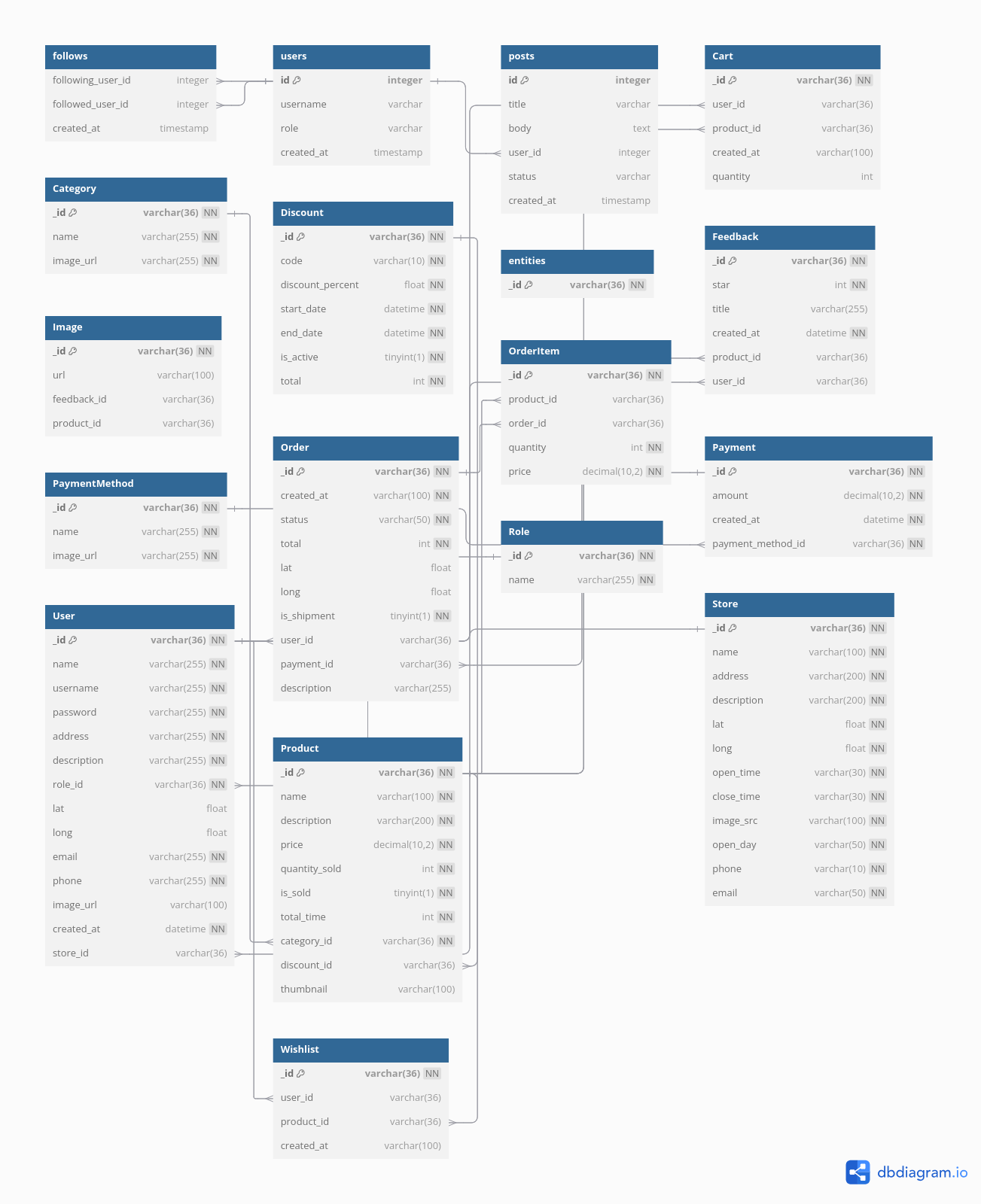
Use Case quản lý thông tin cửa hàng hỗ trợ cập nhật thông tin liên hệ, giờ hoạt động, và chính sách phục vụ.

Use Case quản lý đơn hàng cho phép theo dõi đơn hàng từ khách hàng

Use Case báo cáo cung cấp các tài liệu chi tiết về hiệu suất kinh doanh và hoạt động bán hàng.

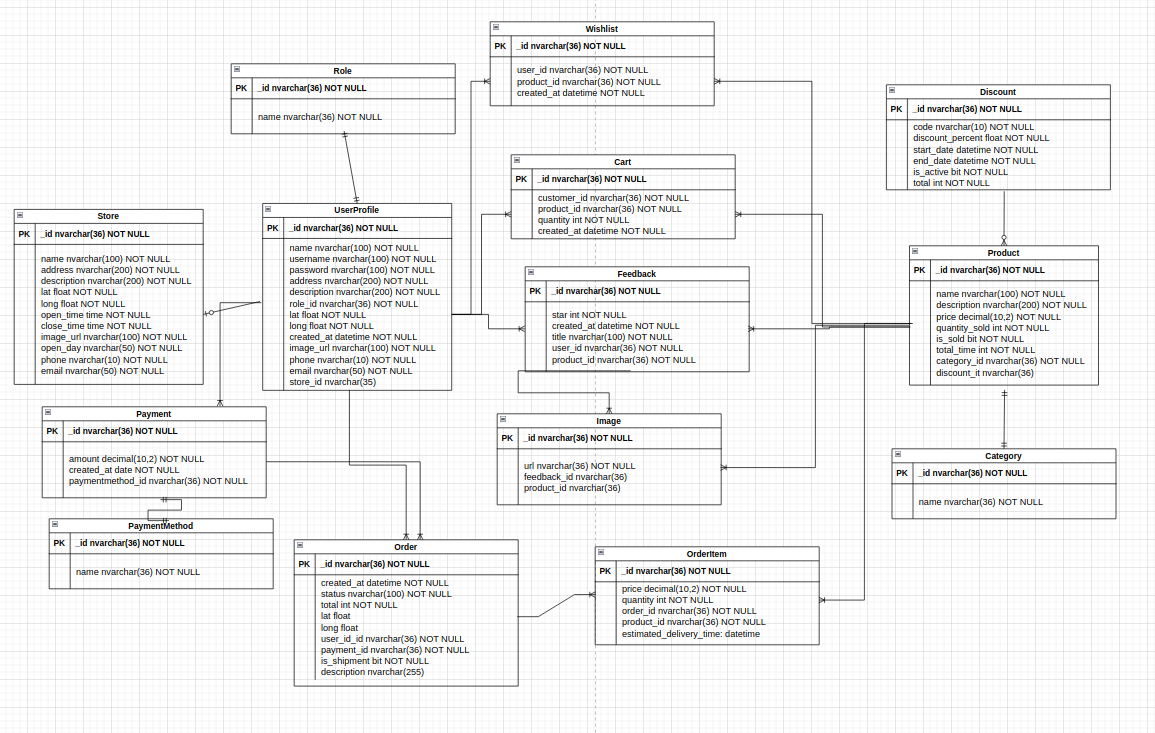
Use Case quản lý đơn hàng cho phép họ xác nhận đơn hàng, theo dõi trạng thái, và cập nhật thông tin khi đơn hàng được xử lý.

## 2.5. Biểu đồ thực thể liên kết mức vật lý

****

*Hình 2. 4 Biểu đồ thực thể liên kết mức vật lý*

## 2.6. Cơ sở dữ liệu

****

*Hình 2. 5 Cơ sở dữ liệu*

Giỏ hàng (Cart): Lưu trữ các sản phẩm mà người dùng đã thêm vào giỏ hàng. Mỗi giỏ hàng bao gồm các thông tin: mã giỏ hàng, mã người dùng, mã sản phẩm, ngày thêm vào, và số lượng. Một giỏ hàng có thể chứa nhiều sản phẩm.

Danh mục sản phẩm (Category): Gồm các thông tin như mã danh mục, tên danh mục, và mô tả. Một danh mục có thể chứa nhiều sản phẩm.

Giảm giá (Discount): Lưu thông tin về các chương trình giảm giá với các trường: mã giảm giá, tên giảm giá, phần trăm giảm và thời gian áp dụng.

Phản hồi (Feedback): Gồm mã phản hồi, mã người dùng, mã sản phẩm, nội dung phản hồi và đánh giá (từ 1 đến 5 sao). Một sản phẩm có thể nhận nhiều phản hồi từ người dùng.

Đơn hàng (Order): Gồm các thông tin: mã đơn hàng, mã người dùng, tổng giá trị, trạng thái, ngày đặt hàng, và ngày giao hàng. Một người dùng có thể tạo nhiều đơn hàng.

Chi tiết đơn hàng (OrderItem): Liên kết đơn hàng với sản phẩm, bao gồm mã chi tiết đơn hàng, mã đơn hàng, mã sản phẩm, số lượng và giá tại thời điểm mua.

Thanh toán (Payment): Gồm mã thanh toán, mã đơn hàng, số tiền thanh toán, ngày thanh toán, và phương thức thanh toán.

Phương thức thanh toán (PaymentMethod): Lưu thông tin về các phương thức thanh toán (mã phương thức, tên phương thức).

Sản phẩm (Product): Gồm mã sản phẩm, tên sản phẩm, giá, số lượng tồn kho, mô tả, mã danh mục và mã giảm giá (nếu có).

Cửa hàng (Store): Lưu thông tin về các cửa hàng cung cấp sản phẩm, gồm mã cửa hàng, tên cửa hàng, địa chỉ và số điện thoại.

Người dùng (User): Gồm mã người dùng, họ tên, email, mật khẩu, vai trò (mã vai trò) và các thông tin liên hệ.

Vai trò (Role): Lưu thông tin về các vai trò trong hệ thống, như quản trị viên, khách hàng, v.v., bao gồm mã vai trò và tên vai trò.

Danh sách yêu thích (Wishlist): Gồm mã danh sách yêu thích, mã người dùng và mã sản phẩm. Một người dùng có thể lưu nhiều sản phẩm vào danh sách yêu thích.

# CHƯƠNG 3: PHÂN TÍCH USE CASE

## 3.1. Mô tả chi tiết các use case

### 3.1.1. Use case Đăng nhập

- Mô tả: Use case cho phép người dùng đăng nhập vào hệ thống

- Tác nhân: Admin, Nhân viên, Khách hàng, Chủ cửa hàng

- Luồng sự kiện:

· Luồng cơ bản:

1) Use case này bắt đầu khi người dùng truy cập vào trang web, ứng dụng. Hệ thống hiển thị yêu cầu người dùng nhập tên và mật khẩu.

2) Người dùng nhập tên và mật khẩu sau đó click vào nút Đăng nhập. Hệ thống xác nhận tài khoản và mật khẩu có hợp lệ không, nếu hợp lệ thì đăng nhập thành công hệ thống chuyển màn hình đến trang chủ. Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản nếu người dùng không nhập tên hoặc mật khẩu, kiểu dữ liệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng cần nhập đúng thông tin để tiếp tục.

2) Tại bước 2 trong luồng cơ bản nếu kiểm tra tài khoản không có trong bảng User thì hệ thống sẽ hiển thị thông báo lỗi.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng đã có một tài khoản trong hệ thống.

- Hậu điều kiện: Không có.

- Điểm mở rộng: Không có.

### 3.1.2. Use case Đăng ký

- Mô tả: Use case cho phép khách hàng, nhân viên đăng ký tài khoản để sử dụng chức năng của hệ thống

- Tác nhân: khách hàng, nhân viên.

- Luồng sự kiện:

· Luồng cơ bản:

1) Use case này bắt đầu khi Khách hàng ấn vào nút đăng ký tài khoản. Hệ thống sẽ hiển thị lên màn hình yêu cầu nhập các thông tin về: email, mật khẩu, xác nhận mật khẩu…

2) Khách hàng nhập đầy đủ các thông tin hệ thống yêu cầu và ấn vào nút Đăng ký. Hệ thống kiểm tra thông tin rồi lưu thông tin tài khoản vào bảng User và tự động cấp quyền Khách hàng hoặc Nhân viên cho tài khoản đó. Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bước 2 trong luồng cơ bản nếu người dùng không nhập email hoặc mật khẩu, kiểu dữ liệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng cần nhập đúng thông tin để tiếp tục.

2) Tại bước 2 trong luồng cơ bản nếu kiểm tra email đã có trong bảng User thì hệ thống sẽ hiển thị thông báo lỗi.

3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.3. Use case Quản lý giỏ hàng

- Mô tả : Use case này cho phép khách hàng sử dụng giỏ hàng của cá nhân, khách hàng có thể thêm sản phẩm vào giỏ, điều chỉnh số lượng sản phẩm trong giỏ, xóa sản phẩm khỏi giỏ hàng, xem chi tiết sản phẩm có thể đặt hàng nếu muốn.

- Tác nhân : Khách hàng.

- Luồng sự kiện :

· Luồng cơ bản :

1) Use case bắt đầu khi khách hàng click vào giỏ hàng. Hệ thống hiển thị giỏ hàng của khách hàng và cung cấp các lựa chọn : Đặt hàng, Cập nhật số lượng sản phẩm, Xóa sản phẩm khỏi giỏ.

2) Nếu khách hàng chọn Thanh toán thì use case Đặt hàng được thực hiện.

3) Nếu khách hàng chọn Xoá thì use case Xóa sản phẩm khỏi giỏ hàng được thực hiện.

4) Nếu khách hàng chọn Cập nhật thì use case Cập nhật số lượng sản phẩm được thực hiện.

5) Nếu khách hàng chọn vào một sản phẩm thì use case Xem chi tiết sản phẩm được thực hiện và hệ thống chuyển hướng đến trang chi tiết sản phẩm ứng với sản phẩm khách hàng vừa chọn.

6) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Xóa sản phẩm khỏi giỏ:**

- Mô tả: Use case này cho phép khách hàng xóa sản phẩm khỏi giỏ hàng.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn biểu tượng xóa bên cạnh sản phẩm trong giỏ hàng thì hệ thống sẽ xóa sản phẩm tương ứng ra khỏi giỏ hàng.

2) Hệ thống cập nhật lại giỏ hàng và hiển thị lên màn hình cho người dùng.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Cập nhật số lượng sản phẩm:**

- Mô tả: Use case này cho phép khách hàng cập nhật số lượng sản phẩm trong giỏ hàng.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng tăng hay giảm số lượng hàng muốn mua.

2) Người dùng chọn biểu tượng tăng hoặc giảm bên cạnh sản phẩm trong giỏ hàng thì hệ thống tự động cập nhật số lượng sản phẩm và tổng tiền tương ứng trong giỏ hàng.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Thêm vào giỏ hàng:**

- Mô tả: Use case này cho phép khách hàng thêm sản phẩm vào giỏ hàng.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn thêm hàng vào giỏ hàng từ trang xem chi tiết hàng.

2) Hệ thống thêm hàng vào giỏ hàng giúp người dùng và thông báo “Thêm vào giỏ hàng thành công.

3) Nếu sản phẩm đã có trong giỏ hàng thì số lượng của sản phẩm đó trong giỏ hàng sẽ được cập nhật lại bằng đúng với số lượng sẵn có trong giỏ hàng cộng với số lượng vừa mới thêm.

4) Use case kết thúc.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.4. Use case xem chi tiết sản phẩm

- Mô tả : Use case này cho phép khách hàng xem chi tiết sản phẩm.

- Tác nhân : Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản :

1) Use case bắt đầu khi khách hàng muốn click vào sản phẩm. Hệ thống hiển thị thông tin sản phẩm. Khách hàng có thể xem chi tiết thông tin của sản phẩm.

2) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.5. Use case đánh giá và phản hồi

- Mô tả: Use case này cho phép khách hàng đánh giá sản phẩm.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng xem thông tin sản phẩm đã nhận trong trong màn hình quản lý đơn hàng. Hệ thống hiển thị thông tin chi tiết của đơn hàng.

2) Khách hàng nhập đánh giá bằng số sao, phản hồi, sau đó click vào nút Gửi. Hệ thống lưu đánh giá của khách hàng. Use case kết thúc.

3) Nếu khách hàng không nhập thông tin phản hồi mà gửi thì hệ thống sẽ mặc định đánh giá sản phẩm có chất lượng 5 sao. Use case kết thúc.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.6. Use case quản lý thông tin cá nhân

- Mô tả: Use case này cho phép khách hàng cập nhật thông tin tài khoản

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi khách hàng nhấn vào nút thông tin cá nhân. Hệ thống hiển thị thông tin cá nhân của khách hàng.

2) Khách hàng nhập thông tin muốn sửa và chọn nút Cập nhật. Hệ thống tự động cập nhật thông tin cá nhân mới và use case kết thúc.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Khách hàng đã đăng nhập vào tài khoản.

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.7. Use case Quản lý danh sách yêu thích

- Mô tả : Use case này cho phép khách hàng sử dụng danh sách sản phẩm yêu thích của cá nhân, khách hàng có thể thêm sản phẩm vào danh sách, xóa sản phẩm khỏi danh sách, xem chi tiết sản phẩm có thể đặt hàng nếu muốn.

- Tác nhân : Khách hàng.

- Luồng sự kiện :

· Luồng cơ bản :

1) Use case bắt đầu khi khách hàng click vào danh sách yêu thích. Hệ thống hiển thị danh sách yêu thích của khách hàng và cung cấp lựa chọn xóa sản phẩm khỏi danh sách yêu thích .

2) Nếu khách hàng chọn Xoá thì use case Xóa sản phẩm khỏi danh sách yêu thích được thực hiện.

3) Nếu khách hàng chọn vào một sản phẩm thì use case Xem chi tiết sản phẩm được thực hiện và hệ thống chuyển hướng đến trang chi tiết sản phẩm ứng với sản phẩm khách hàng vừa chọn.

6) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Xóa sản phẩm khỏi danh sách yêu thích:**

- Mô tả: Use case này cho phép khách hàng xóa sản phẩm khỏi danh sách yêu thích.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn biểu tượng xóa bên cạnh sản phẩm trong giỏ hàng thì hệ thống sẽ xóa sản phẩm tương ứng ra khỏi giỏ hàng.

2) Hệ thống cập nhật lại giỏ hàng và hiển thị lên màn hình cho người dùng.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Thêm vào danh sách yêu thích:**

- Mô tả: Use case này cho phép khách hàng thêm sản phẩm vào danh sách yêu thích.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn thêm hàng vào danh sách yêu thích từ trang danh sách sản phẩm hoặc trang chi tiết sản phẩm.

2) Hệ thống thêm hàng vào danh sách yêu thích giúp người dùng.

3) Use case kết thúc.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng phải đăng nhập vào hệ thống

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.8.Use case Xem thông tin cửa hàng

- Mô tả: Use case này cho phép khách hàng xem thông tin của cửa hàng.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi khách hàng nhấn vào biểu tượng cửa hàng. Hệ thống hiển thị thông tin của cửa hàng.

2) Use case kết thúc.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Khách hàng đã đăng nhập vào tài khoản.

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.9. Use case Tìm kiếm món ăn

- Mô tả: Use case này cho phép khách hàng tìm kiếm sản phẩm theo tên.

- Tác nhân: Khách hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case này bắt đầu khi khách hàng gõ tên sản phẩm trên thanh tìm kiếm và nhấn tìm kiếm. Hệ thống hiển thị các sản phẩm phù hợp với thông tin tìm kiếm của khách hàng.

2) Trường hợp người dùng bấm vào xem sản phẩm sẽ gọi tới use case "Xem chi tiết sản phẩm".

3) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.10. Use case đặt hàng

- Mô tả: Use case này cho phép khách hàng đặt hàng.

- Tác nhân: Khách hàng.

- Luồng sự kiện

· Luồng cơ bản

1) Use case bắt đầu khi khách hàng click vào biểu tượng giỏ hàng. Hệ thống hiển thị danh sách sản phẩm trong giỏ hàng

2) Khách hàng click vào nút đặt hàng. Hệ thống sẽ đặt hàng sản phẩm.

4) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Khách hàng đã đăng nhập vào hệ thống.

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.11. Use case theo dõi đơn hàng

- Mô tả: Use case này cho phép khách hàng theo dõi các đơn hàng theo trạng thái.

- Tác nhân: Khách hàng.

- Luồng sự kiện

· Luồng cơ bản

1) Use case bắt đầu khi khách hàng click vào biểu tượng đơn hàng. Hệ thống hiển thị danh sách các đơn hàng theo từng trạng thái.

2) Khách hàng click vào đơn hàng để xem chi tiết đơn hàng.

3) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Khách hàng đã đăng nhập vào hệ thống.

- Hậu điều kiện: Không

- Điểm mở rộng: Nếu đơn hàng đang được giao thì người dùng có thể xem vị trí của đơn hàng theo thời gian thực trên Google Map.

### 3.1.12. Use case Quản lý đơn hàng theo trạng thái

- Mô tả : Use case này cho phép nhân viên xem và cập nhật trạng thái của đơn hàng nếu muốn.

- Tác nhân : Nhân viên.

- Luồng sự kiện :

· Luồng cơ bản :

1) Use case bắt đầu khi nhân viên click vào đơn hàng. Hệ thống hiển thị danh sách đơn hàng của khách hàng theo từng trạng thái.

2) Nếu nhân viên chọn vào đơn hàng hệ thống sẽ chuyển sang màn hình chi tiết đơn hàng.

3) Nếu người dùng chọn nút cập nhật thông tin đơn hàng thì hệ thống sẽ tự động cập nhật thông tin đơn hàng.

6) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.13. Use case Quản lý danh mục

- Mô tả : Use case này cho chủ cửa hàng thêm sửa xóa danh mục của cửa hàng.

- Tác nhân : Chủ cửa hàng.

- Luồng sự kiện :

· Luồng cơ bản :

1) Use case bắt đầu khi chủ cửa hàng click vào danh mục sản phẩm. Hệ thống hiển thị danh sách danh mục sản phẩm và cung cấp các lựa chọn thêm, cập nhật, xóa.

2) Nếu chủ cửa hàng chọn thêm thì use case thêm thư mục được thực hiện.

3) Nếu chủ cửa hàng chọn Xoá thì use case Xóa thư mục khỏi giỏ hàng được thực hiện.

4) Nếu chủ cửa hàng chọn Cập nhật thì use case Cập nhật thư mục được thực hiện.

5) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Xóa danh mục:**

- Mô tả: Use case này cho phép chủ cửa hàng xóa danh mục.

- Tác nhân: Chủ cửa hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn nút xóa danh mục thì hệ thống sẽ xóa sản phẩm tương ứng.

2) Hệ thống cập nhật lại danh sách danh mục và hiển thị lên màn hình cho người dùng.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Cập nhật danh mục:**

- Mô tả: Use case này cho phép khách hàng cập nhật lại thông tin của các danh mục.

- Tác nhân: Chủ cửa hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng thay đổi thông tin danh mục trong danh sách.

2) Người dùng chọn nút cập nhật, hệ thống tự động cập nhật lại thông tin của toàn bộ danh mục.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Thêm danh mục:**

- Mô tả: Use case này cho phép chủ cửa hàng thêm danh mục.

- Tác nhân: Chủ cửa hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn thêm.

2) Hệ thống hiển thị form nhập thông tin danh mục. Người dùng bấm nút thêm hệ thống sẽ thêm danh mục và cập nhật lại danh mục.

3) Use case kết thúc.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

### 3.1.14. Use case Quản lý sản phẩm

- Mô tả : Use case này cho chủ cửa hàng thêm sửa xóa sản phẩm của cửa hàng.

- Tác nhân : Chủ cửa hàng.

- Luồng sự kiện :

· Luồng cơ bản :

1) Use case bắt đầu khi chủ cửa hàng click vào sản phẩm. Hệ thống hiển thị danh sách sản phẩm và cung cấp các lựa chọn thêm, cập nhật, xóa.

2) Nếu chủ cửa hàng chọn thêm thì use case thêm sản phẩm được thực hiện.

3) Nếu chủ cửa hàng chọn Xoá thì use case Xóa sản phẩm khỏi danh sách được thực hiện.

4) Nếu chủ cửa hàng chọn Cập nhật thì use case Cập nhật sản phẩm được thực hiện.

5) Use case kết thúc.

· Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Xóa sản phẩm:**

- Mô tả: Use case này cho phép chủ cửa hàng xóa sản phẩm.

- Tác nhân: Chủ cửa hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn nút xóa danh mục thì hệ thống sẽ xóa sản phẩm tương ứng.

2) Hệ thống cập nhật lại danh sách sản phẩm và hiển thị lên màn hình cho người dùng.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Cập nhật danh mục:**

- Mô tả: Use case này cho phép chủ cửa hàng cập nhật lại thông tin của các sản phẩm.

- Tác nhân: Chủ cửa hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng thay đổi thông tin sản phẩm trong danh sách.

2) Người dùng chọn nút cập nhật, hệ thống tự động cập nhật lại thông tin của toàn bộ sản phẩm.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

- Hậu điều kiện: Không

- Điểm mở rộng: Không

**Ø Use case Thêm sản phẩm:**

- Mô tả: Use case này cho phép chủ cửa hàng thêm sản phẩm.

- Tác nhân: Chủ cửa hàng.

- Luồng sự kiện:

· Luồng cơ bản

1) Use case bắt đầu khi người dùng chọn thêm.

2) Hệ thống hiển thị form nhập thông tin sản phẩm. Người dùng bấm nút thêm hệ thống sẽ thêm sản phẩm và cập nhật lại sản phẩm.

3) Use case kết thúc.

· Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- Các yêu cầu đặc biệt: Không

- Tiền điều kiện: Không

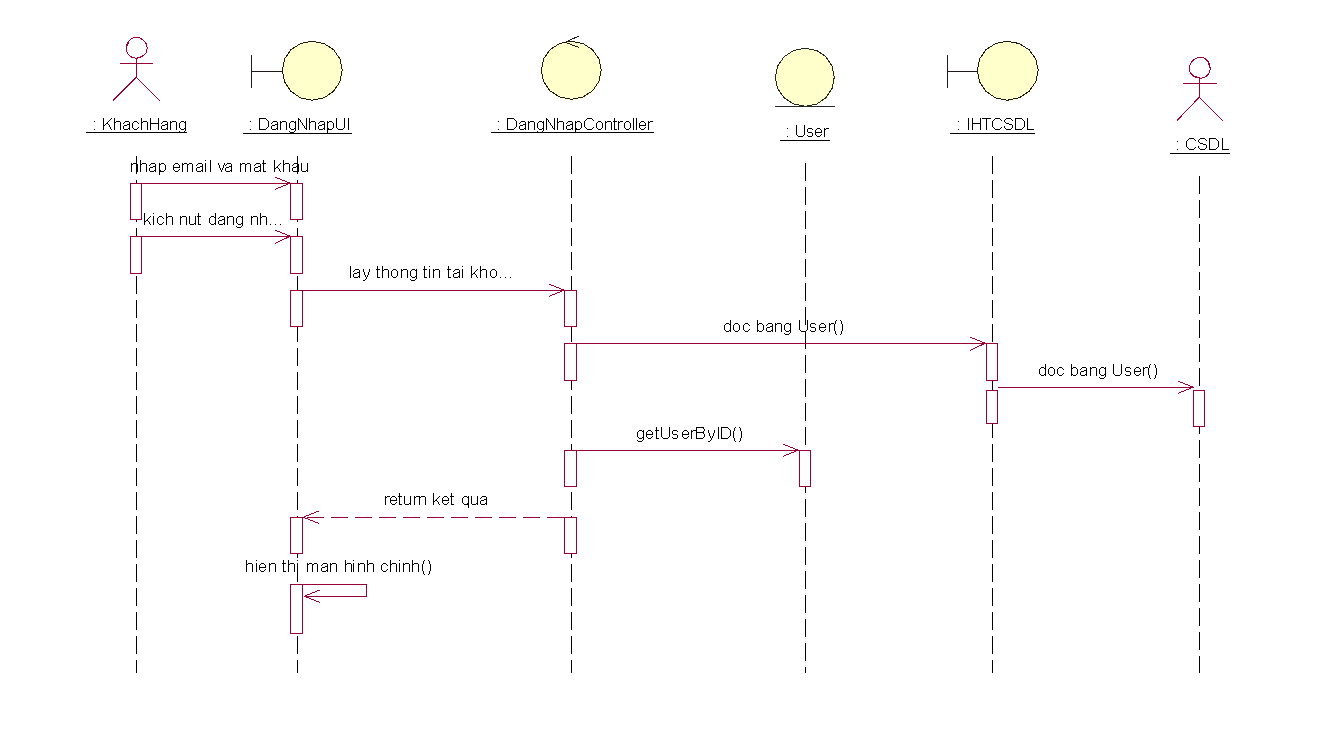
- Hậu điều kiện: Không

- Điểm mở rộng: Không

## 3.2. Phân tích use case

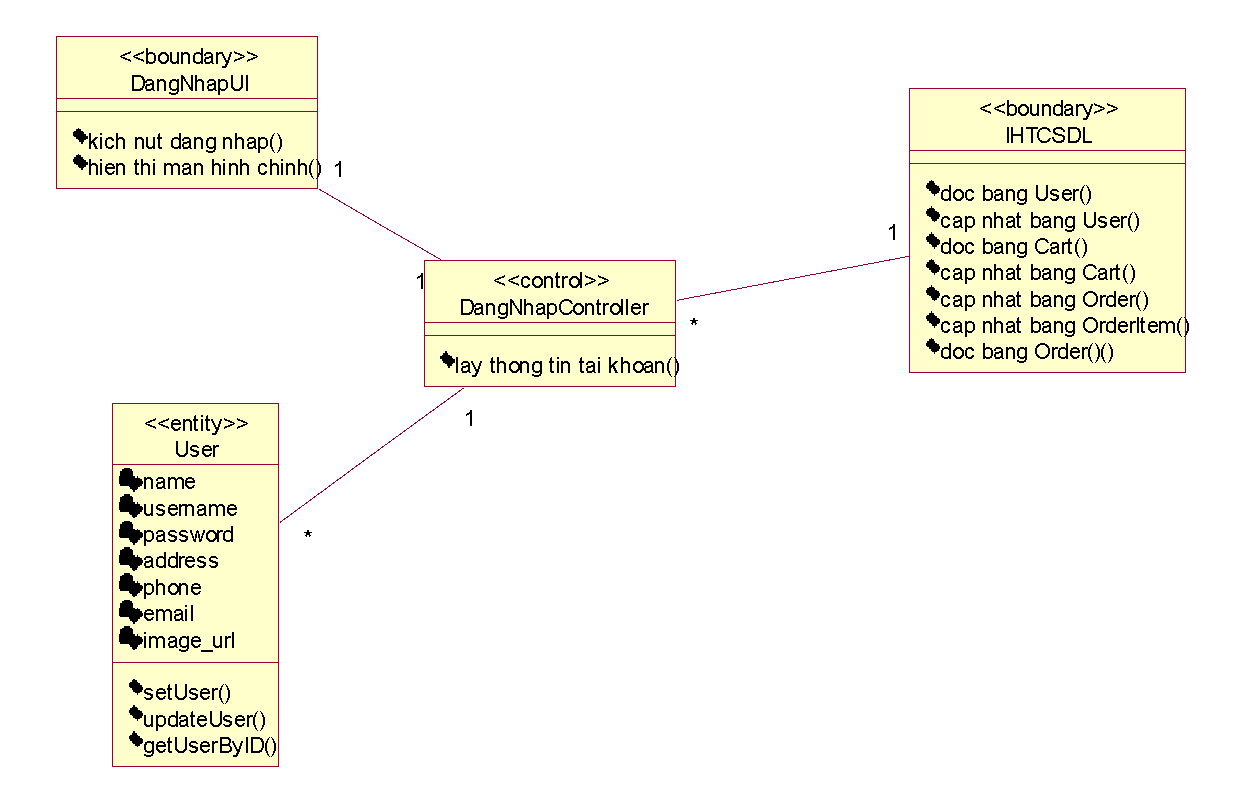
### 3.1.1. Use case đăng nhập

#### **3.1.1.1. Biểu đồ trình tự**

****

*Hình 3. 1 Luồng cơ bản use case đăng nhập*

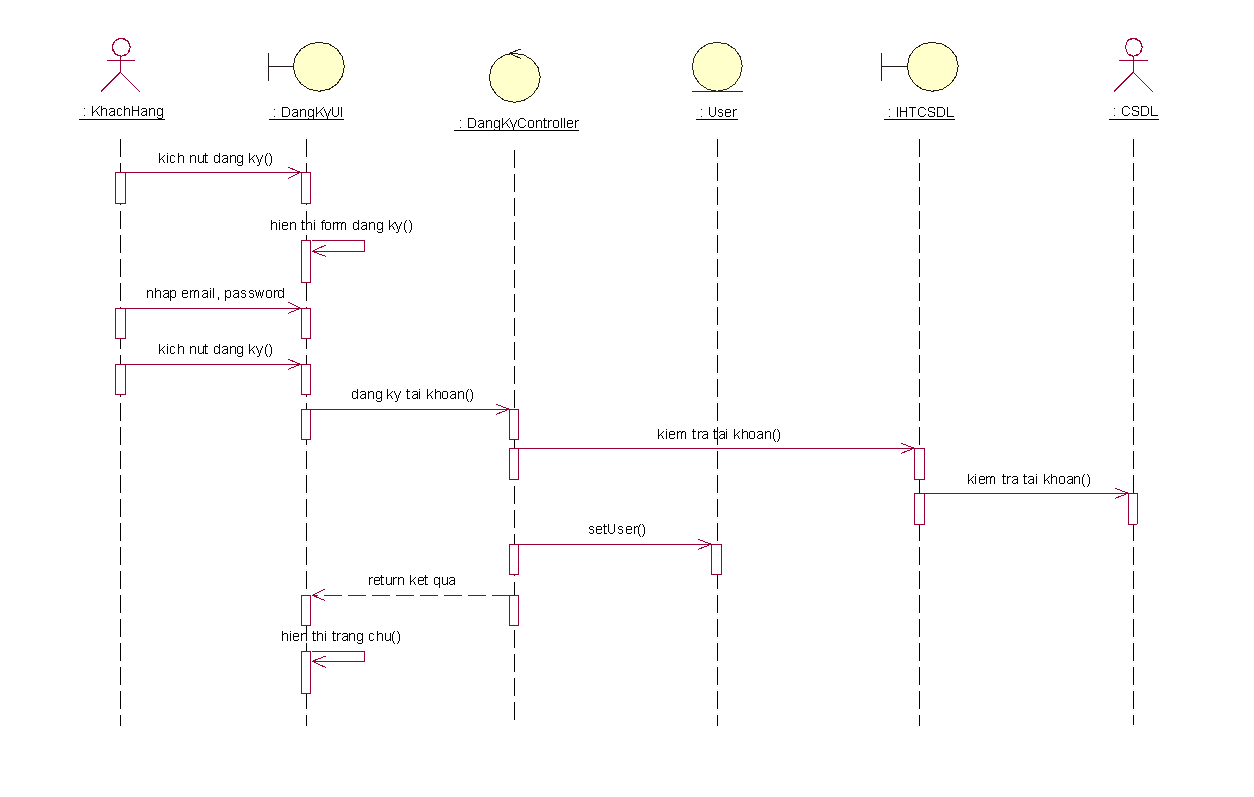
#### **3.1.1.2. Biểu đồ lớp phân tích**



*Hình 3. 2 Biểu đồ lớp phân tích use case đăng nhập*

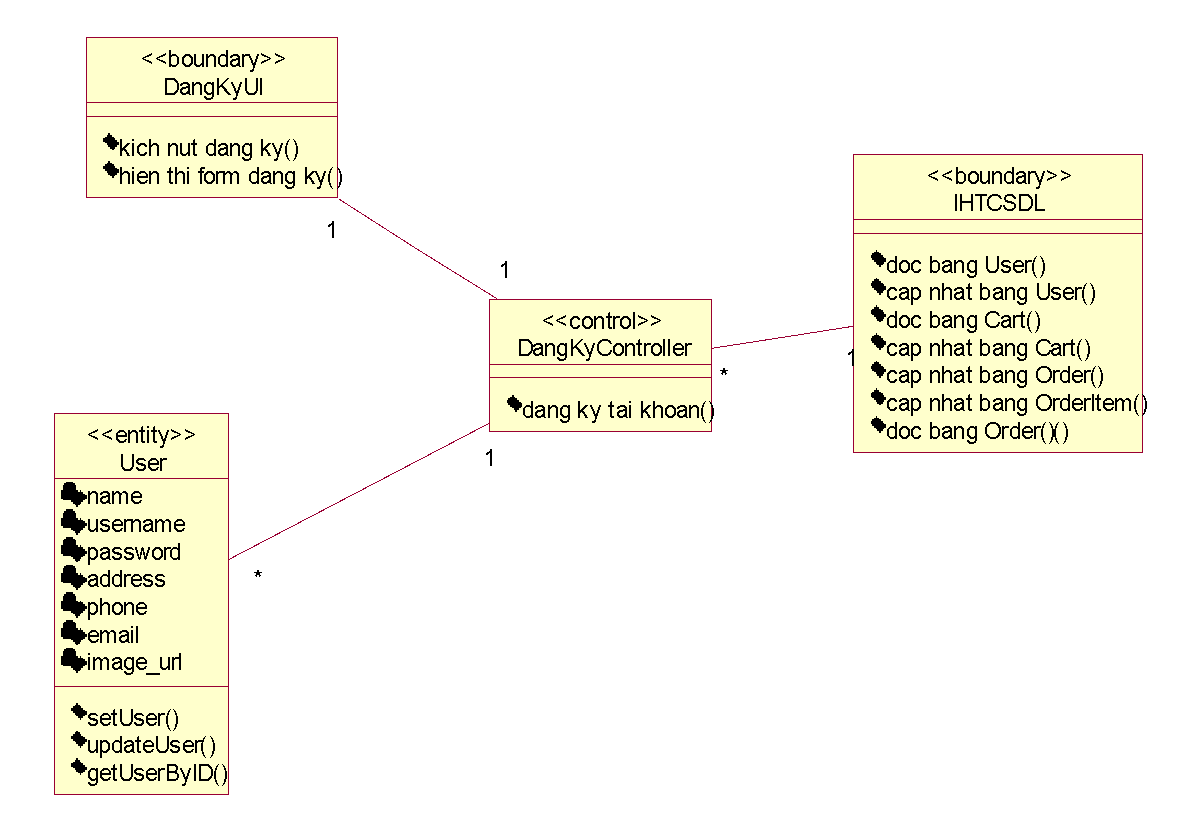
### 3.1.2. Use case Đăng ký

#### **3.1.2.1. Biểu đồ trình tự**



*Hình 3. 3 3 Luồng cơ bản use case đăng ký*

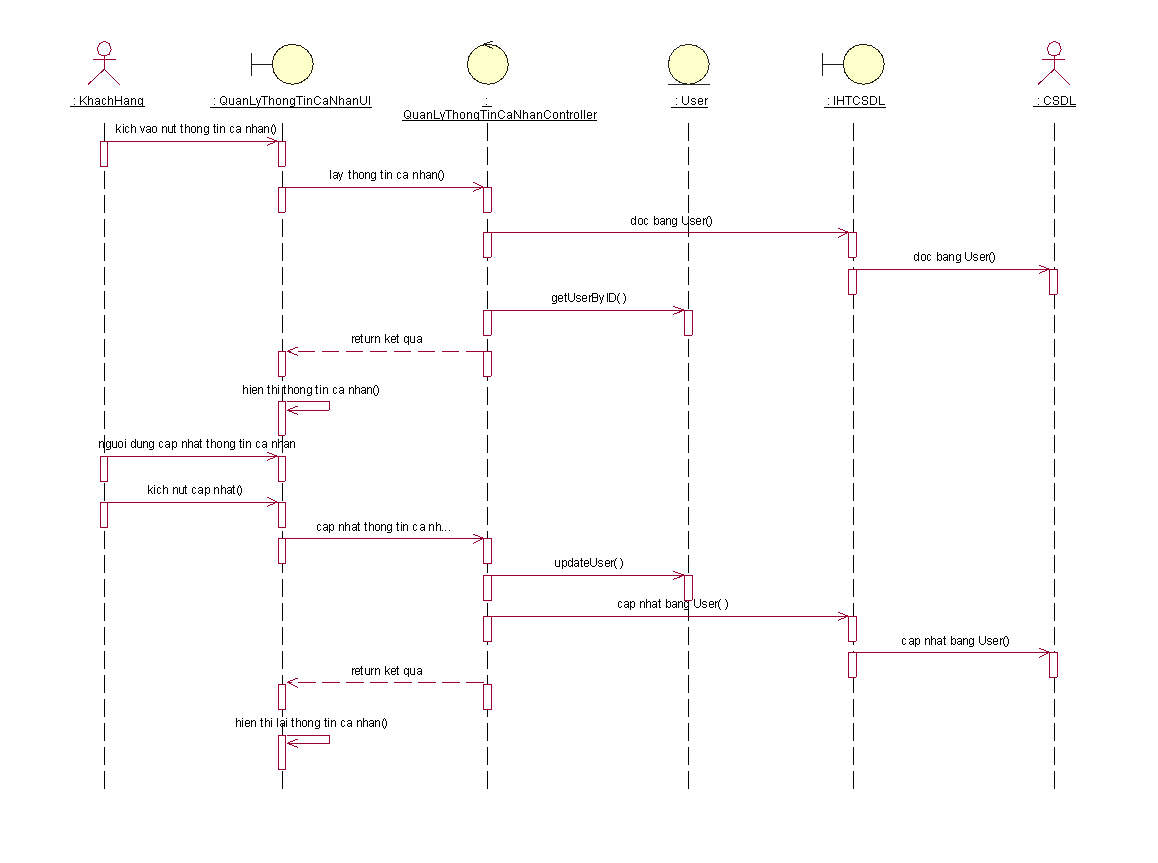
#### **3.1.2.2. Biểu đồ lớp phân tích**



*Hình 3. 4 Biểu đồ lớp phân tích use case đăng ký*

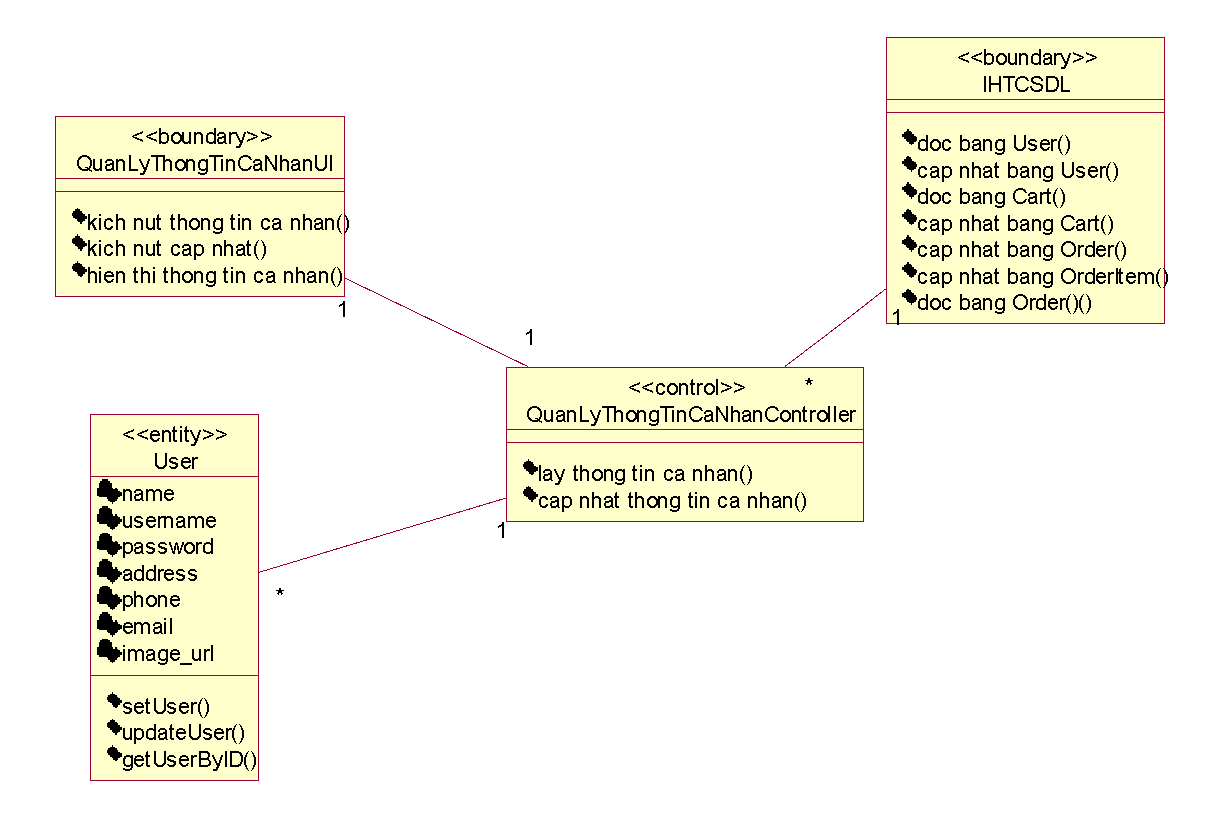
### 3.1.3. Use case Quản lý thông tin cá nhân

#### **3.1.3.1. Biểu đồ trình tự**



*Hình 3. 5 Luồng cơ bản use case quản lý thông tin cá nhân*

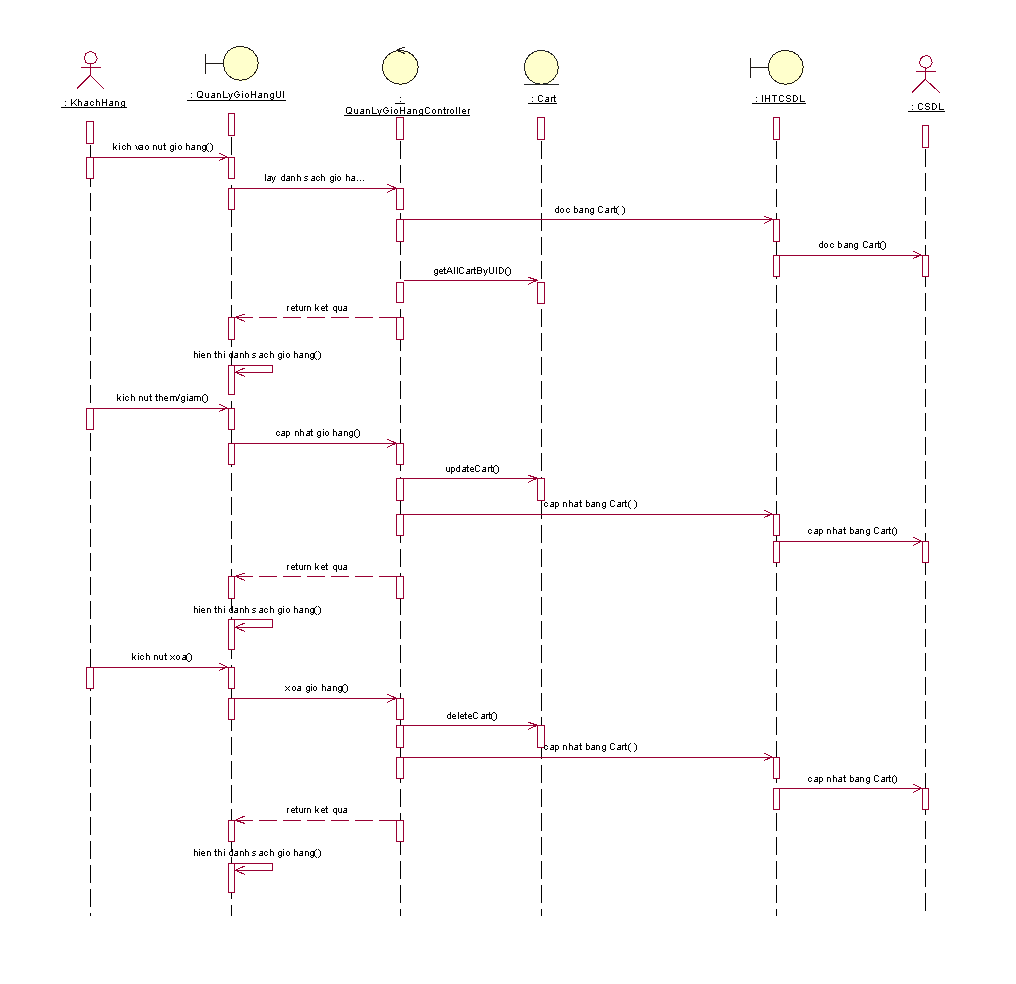
#### **3.1.3.2. Biểu đồ lớp phân tích**



*Hình 3. 6 Biểu đồ lớp phân tích use case quản lý thông tin cá nhân*

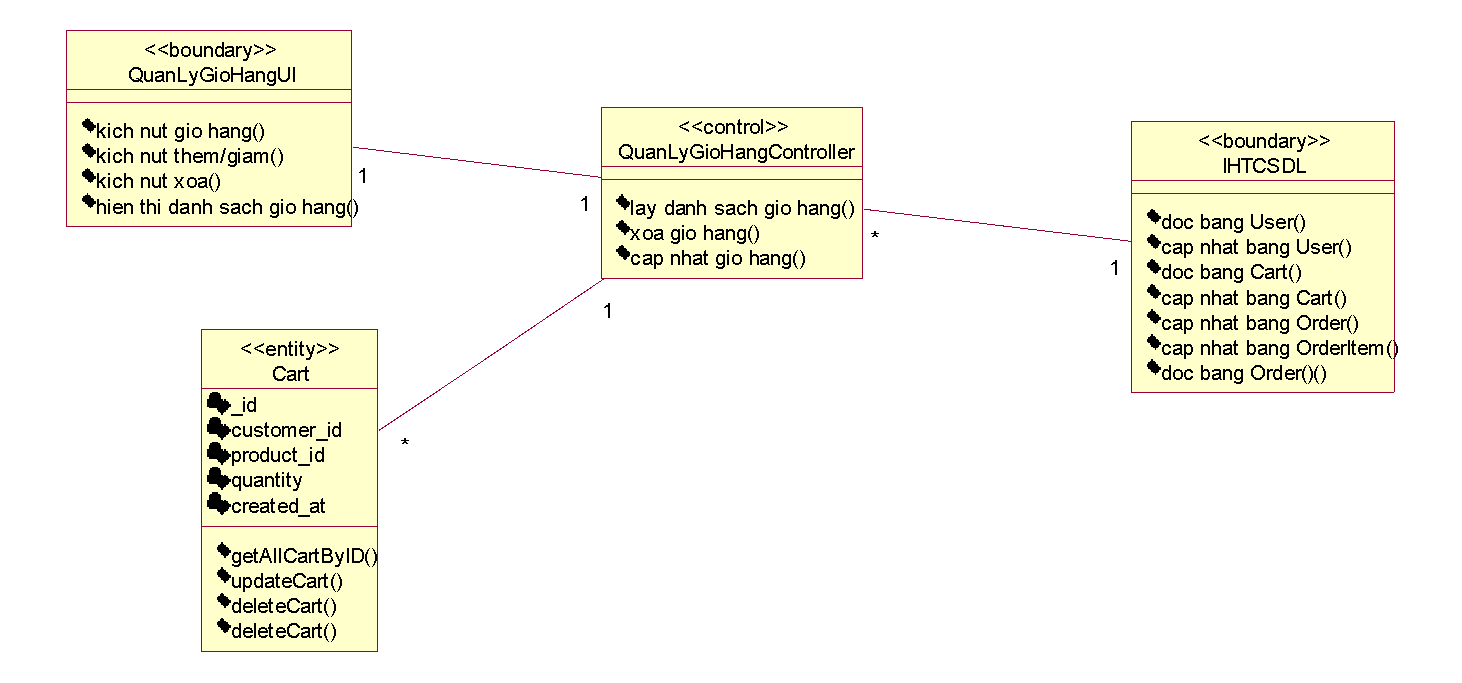
### 3.1.4. Quản lý giỏ hàng

#### **3.1.4.1. Biểu đồ trình tự**

****

*Hình 3. 7 Luồng cơ bản use case quản lý giỏ hàng*

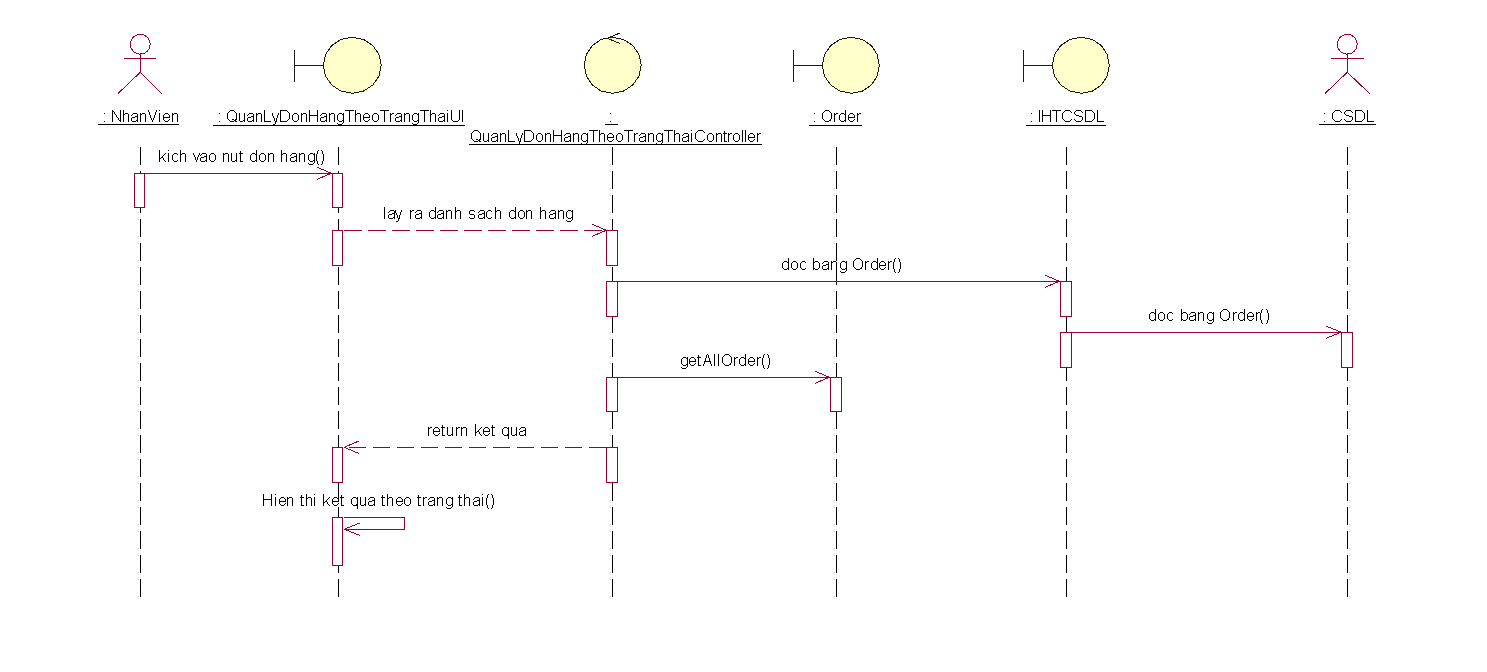
#### **3.1.4.2. Biểu đồ lớp phân tích**

****

*Hình 3. 8 Biểu đồ lớp phân tích use case quản lý giỏ hàng*

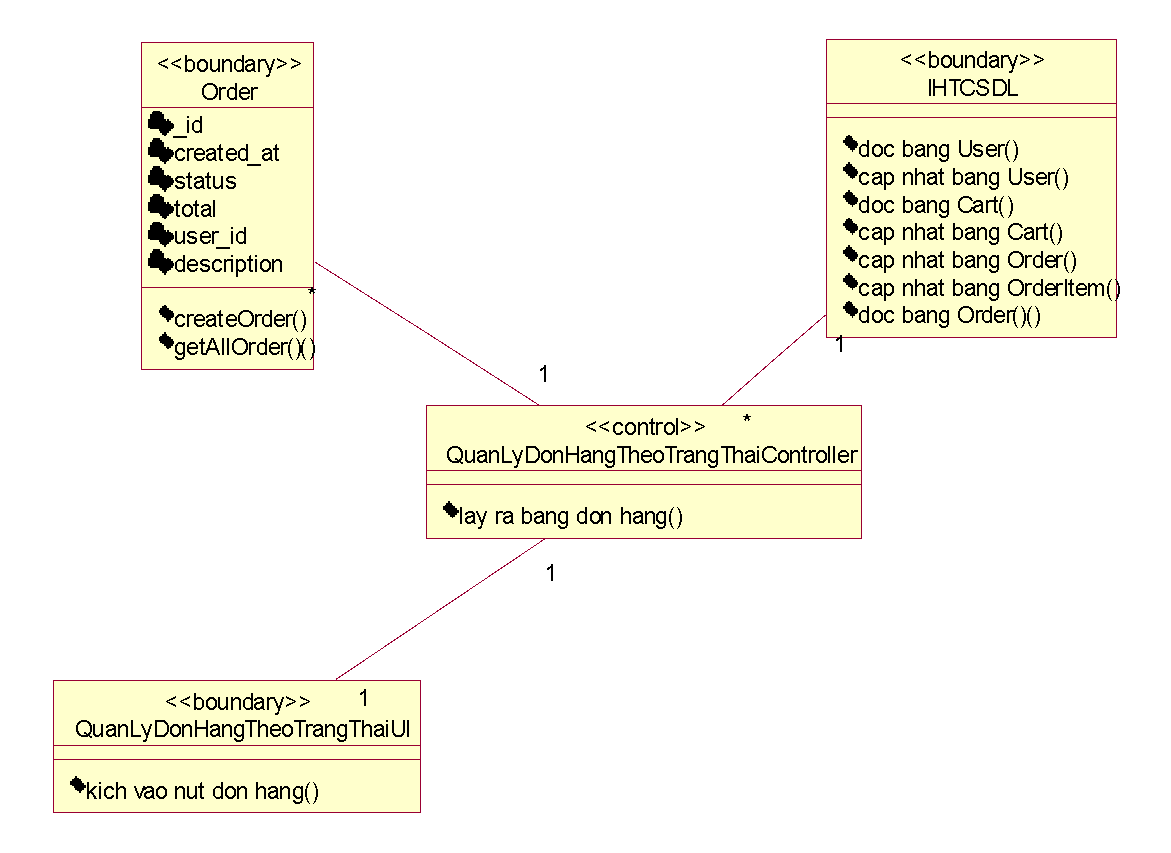
### 3.1.5. Quản lý đơn hàng theo trạng thái

#### **3.1.5.1. Biểu đồ trình tự**

****

*Hình 3. 9 Luồng cơ bản use case quản lý đơn hàng theo trạng thái*

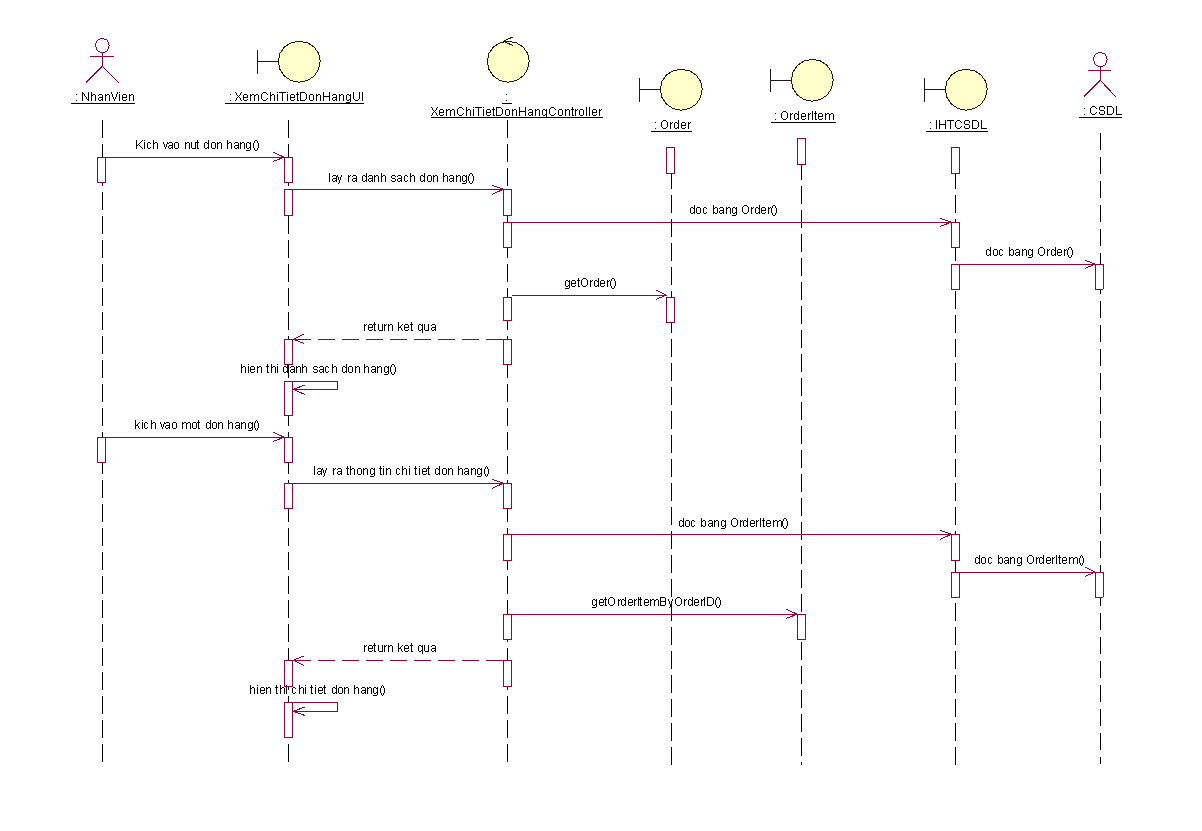
#### **3.1.5.2. Biểu đồ lớp phân tích**

****

*Hình 3. 10 Biểu đồ lớp phân tích use case quản lý đơn hàng theo trạng thái*

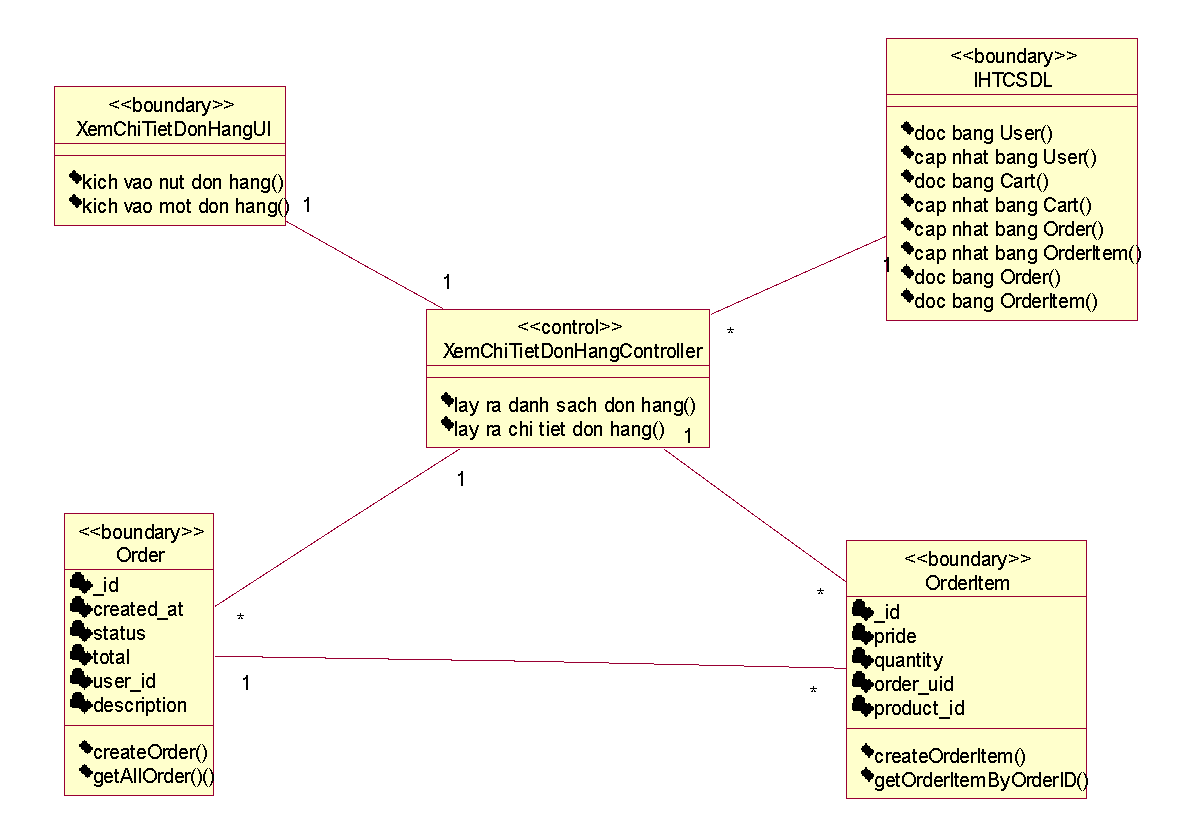
### 3.1.6. Xem chi tiết đơn hàng

#### **3.1.6.1. Biểu đồ trình tự**

****

*Hình 3. 11 Luồng cơ bản use case xem chi tiết đơn hàng*

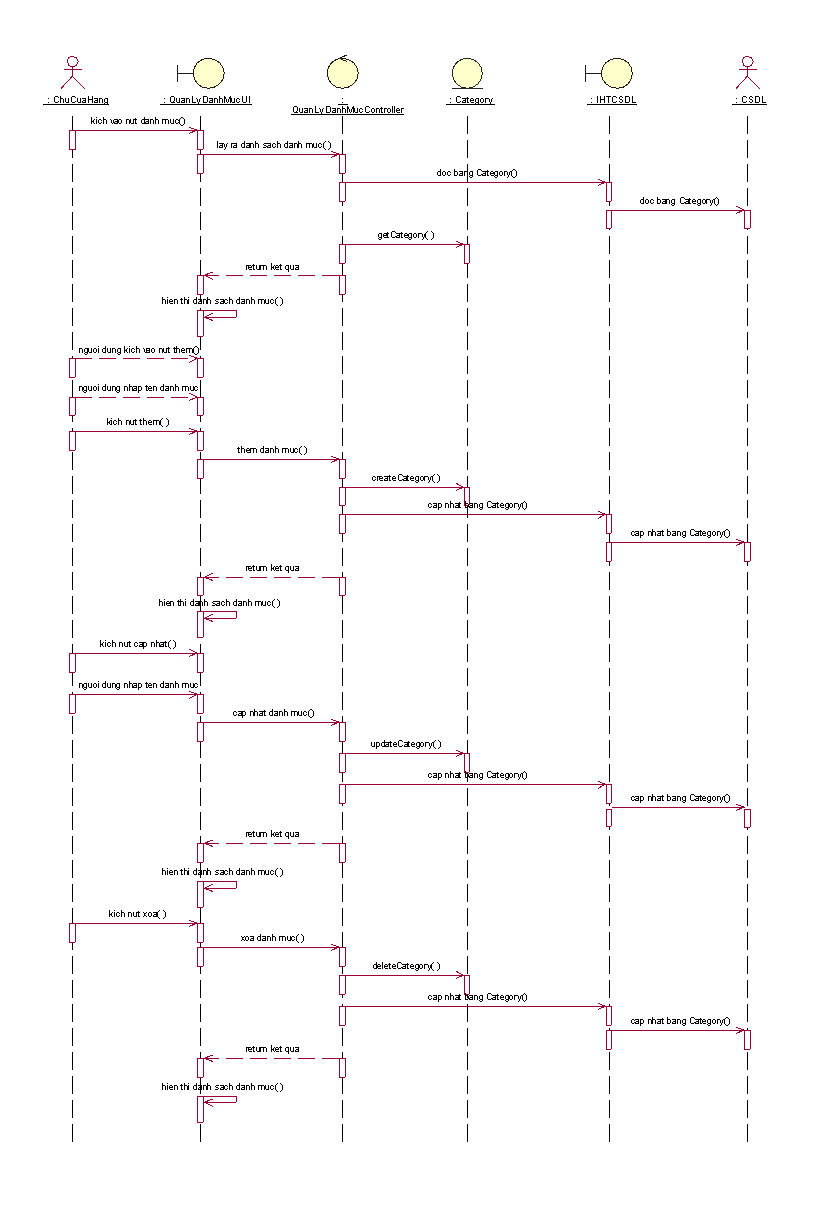
#### **3.1.6.2. Biểu đồ lớp phân tích**

****

*Hình 3. 12 Biểu đồ lớp phân tích use case xem chi tiết đơn hàng*

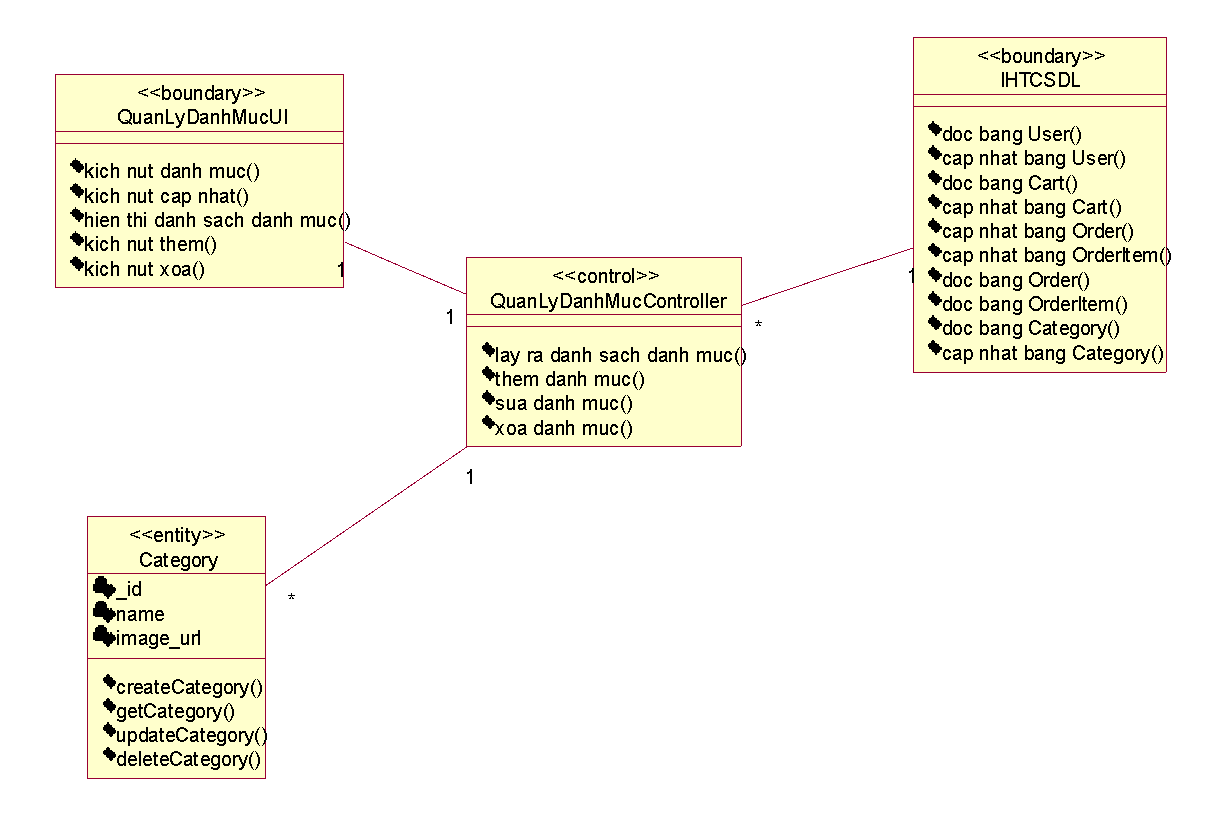
### 3.1.7. Quản lý danh mục

#### **3.1.7.1. Biểu đồ trình tự**

****

*Hình 3. 13 Luồng cơ bản use case quản lý danh mục*

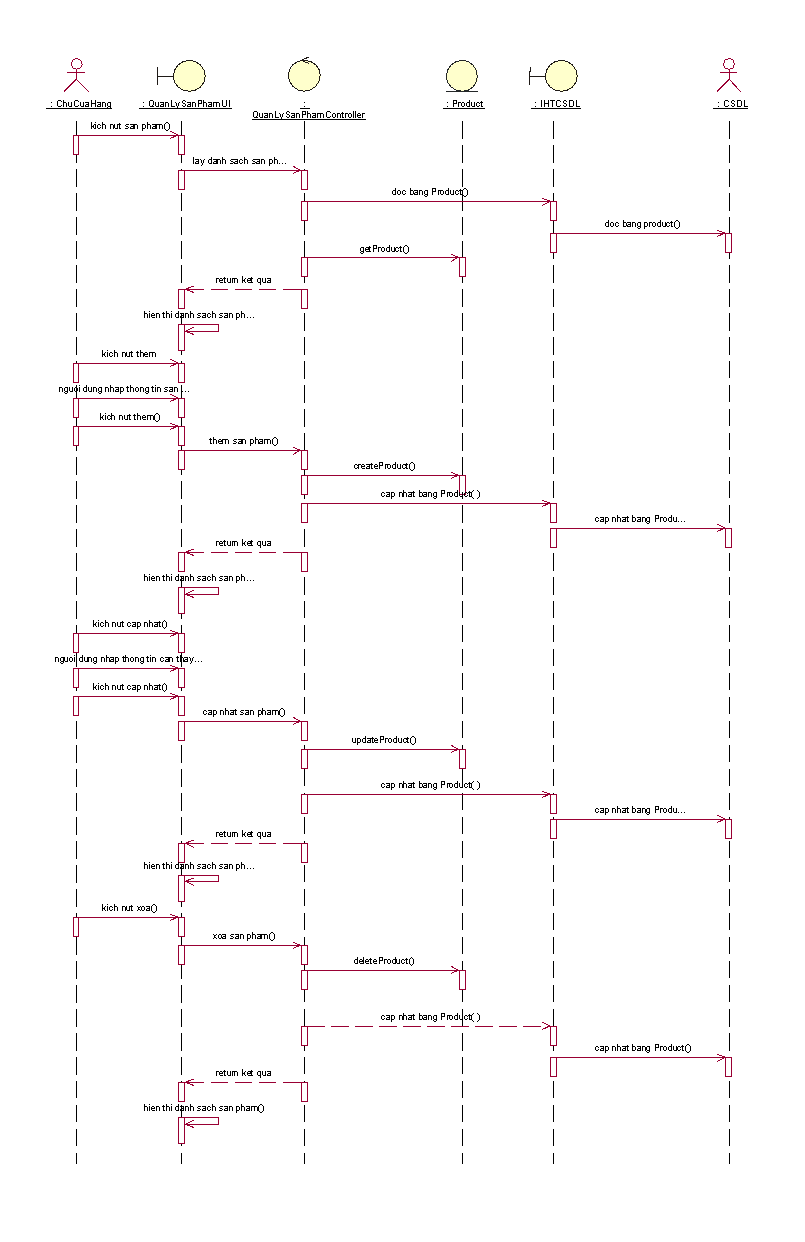
#### **3.1.7.2. Biểu đồ lớp phân tích**

****

*Hình 3. 14 Biểu đồ lớp phân tích use case quản lý danh mục*

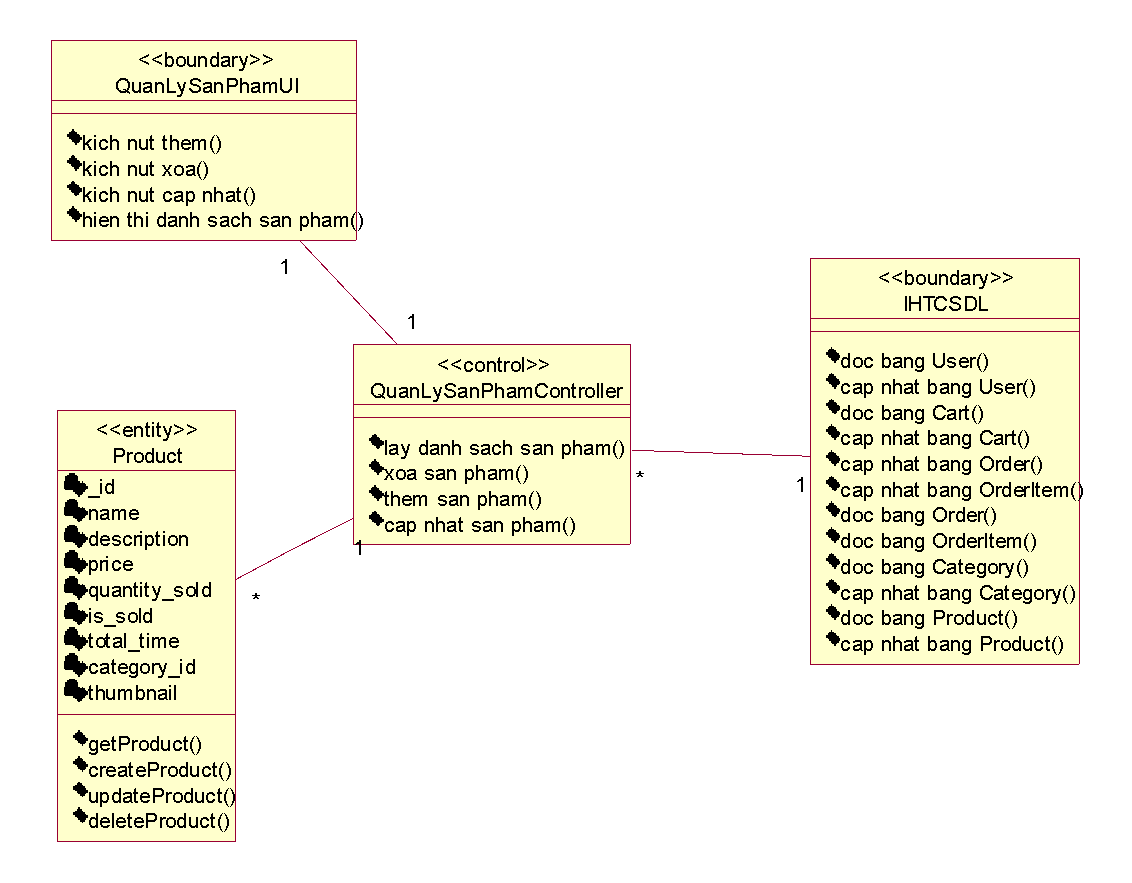
### 3.1.8. Quản lý sản phẩm

#### **3.1.8.1. Biểu đồ trình tự**

****

*Hình 3. 15 Luồng cơ bản use case quản lý sản phẩm*

#### **3.1.8.2. Biểu đồ lớp phân tích**

****

*Hình 3. 16 Biểu đồ lớp phân tích use case quản lý sản phẩm*

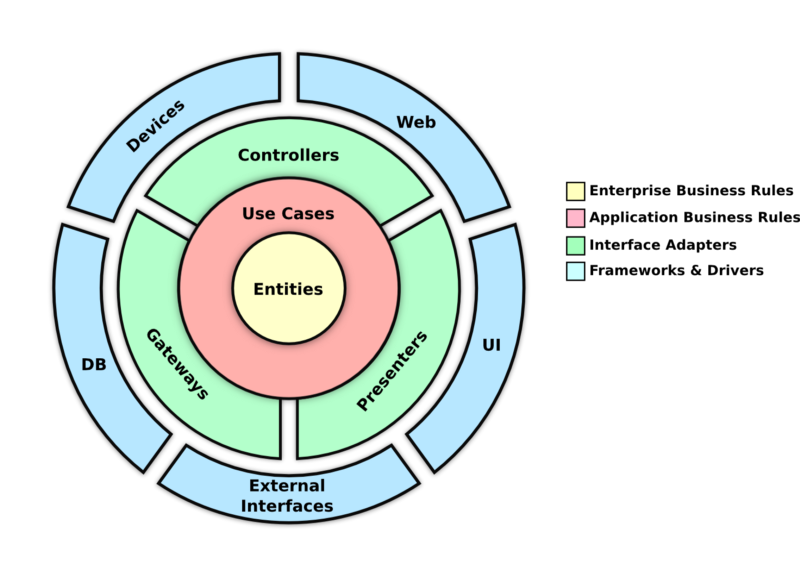
# Chương 4: TRIỂN KHAI VÀ KIỂM THỬ PHẦN MỀM

## 4.1. Lựa chọn công nghệ triển khai

### 4.1.2. Flask

Flask là một microframework, nói một cách dễ hiểu chính là một bộ khung ứng dụng được viết bằng Python. Do là microframework nên việc mở rộng rất dễ dàng tuy nhiên sẽ phụ thuộc vào các thư viện của python. Kết hợp flask với thư viện restx giúp việc phát triển API theo chuẩn restful trở nên nhanh chóng và dễ dàng với tài liệu kiểm thử rõ ràng.

Mẫu kiến trúc phần mềm được sử dụng là Clean Architecture được phát triển bởi Robert C.Martin hay còn được gọi là Uncle Bob nhằm tạo ra hệ thống dễ bảo trì và mở rộng dựa trên tư tưởng độc lập kết hợp với các nguyên lý thiết kế hướng đối tượng

Kiến trúc của Clean Architecture chia thành 4 layer với một quy tắc phụ thuộc. Các layer bên trong không nên biết bất kỳ điều gì về các layer bên ngoài. Điều này có nghĩa là nó có quan hệ phụ thuộc nên "hướng" vào bên trong.

*Hình 4. 1 Mô hình Clean Architecture*

*Entities*: là khái niệm dùng để mô tả các Business Logic. Đây là layer quan trọng nhất, là nơi bạn thực hiện giải quyết các vấn đề - mục đích khi xây dựng app. Layer này không chứa bất kỳ một framework nào, bạn có thể chạy nó mà không cần emulator. Nó giúp bạn dễ dàng test, maintain và develop phần business logic.

*Use case* : chứa các rule liên quan trực tiếp tới ứng dụng cục bộ (application-specific business rules).

*Interface Adapter* : tập hợp các adapter phục vụ quá trình tương tác với các công nghệ.

*Framework and Drivers* : chứa các công cụ về cơ sở dữ liệu và các framework, thông thường bạn sẽ không phải lập trình nhiều ở tầng này. Tuy nhiên cần chắc chắn về mức ưu tiên sử dụng các công cụ này trong project.

### 4.1.2. MySQL

**MySQL** là một **Relational Database Management System (RDBMS)** mã nguồn mở, được sử dụng rộng rãi để quản lý cơ sở dữ liệu. Nó cũng sử dụng ngôn ngữ **SQL (Structured Query Language)** để giao tiếp với cơ sở dữ liệu. MySQL thường được biết đến nhờ hiệu năng cao, dễ sử dụng và tích hợp tốt trong các ứng dụng web.

MySQL hoạt động theo kiến trúc **client-server**. Máy chủ (server) chịu trách nhiệm quản lý cơ sở dữ liệu và xử lý các yêu cầu, trong khi máy khách (client) gửi câu lệnh SQL đến server để thực thi và nhận kết quả trả về.

Nhìn chung, MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ và linh hoạt, phù hợp cho nhiều loại ứng dụng, từ các dự án cá nhân nhỏ đến các hệ thống doanh nghiệp lớn. Với khả năng hỗ trợ mã nguồn mở và hiệu suất cao, MySQL là lựa chọn phổ biến trong phát triển ứng dụng web và các hệ thống xử lý dữ liệu.

### 4.1.3. SocketIO

**Socket** là một cơ chế giao tiếp hai chiều giữa các ứng dụng trong mạng máy tính, cho phép trao đổi dữ liệu theo thời gian thực. Socket hoạt động dựa trên các giao thức truyền thông như TCP hoặc UDP, trong đó một bên đóng vai trò **máy chủ (server)** để lắng nghe kết nối, và bên còn lại là **máy khách (client)** thực hiện gửi yêu cầu. Socket thường được sử dụng trong các ứng dụng chat, truyền tải dữ liệu, trò chơi trực tuyến và các hệ thống thời gian thực khác.

### 4.1.4. Streamlit

Streamlit là framework được viết bằng python thường được dùng trong machine learning để xây dựng web app nhanh chóng chỉ với vài dòng code mà không cần phải biết nhiều về HTML, CSS hay JavaScript.

Streamlit hỗ trợ các giao diện cơ bản, biểu đồ, … Thường được sử dụng để làm dashboard để trực quan hóa các dữ liệu.

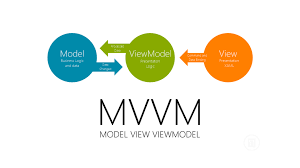
### 4.1.5. Quản lý phiên bản (Version Control System - VCS):

Sử dụng Git làm hệ thống quản lý phiên bản cho dự án. Git cho phép chúng tôi theo dõi sự thay đổi trong mã nguồn, quản lý nhánh phát triển (branch) và hợp nhất các phiên bản một cách hiệu quả. Repository Git sẽ được lưu trữ trên các nền tảng như GitHub hoặc GitLab để quản lý mã nguồn.

### 4.1.6. Android (Kotlin)

Kotlin là ngôn ngữ lập trình ứng dụng di động cho nền tảng Android được phát triển bởi JetBrains và được Google hỗ trợ chính thức vào năm 2017 để thay thế cho Java với ưu điểm ngắn gọn, dễ đọc, có khả năng tương thích ngược với java,...

Mẫu thiết kế phần mềm MVVM tách bạc giữa các thành phần của ứng dụng, mô hình có thiết kế như sau:



*Hình 4. 2 Mô hình MVVM*

Các thành phần trong mô hình MVVM bao gồm:

View: Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.

Model: Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.

ViewModel: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.

### 4.1.7. Docker

Docker là một nền tảng cho developers và sysadmin để develop, deploy và run application với container. Nó cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng và môi trường này được gọi là container. Khi cần deploy lên bất kỳ server nào chỉ cần run container của Docker thì application của bạn sẽ được khởi chạy ngay lập tức.

Lợi ích của docker:

* Không như máy ảo Docker start và stop chỉ trong vài giây.
* Bạn có thể khởi chạy container trên mỗi hệ thống mà bạn muốn.
* Container có thể build và loại bỏ nhanh hơn máy ảo.
* Dễ dàng thiết lập môi trường làm việc. Chỉ cần config 1 lần duy nhất và không bao giờ phải cài đặt lại các dependencies. Nếu bạn thay đổi máy hoặc có người mới tham gia vào project thì bạn chỉ cần lấy config đó và đưa cho họ.
* Nó giữ cho word-space của bạn sạch sẽ hơn khi bạn xóa môi trường mà ảnh hưởng đến các phần khác.

### 4.1.8. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL phổ biến, được thiết kế để lưu trữ dữ liệu dưới dạng document thay vì bảng như trong các hệ SQL truyền thống. Dữ liệu trong MongoDB được lưu trữ dưới dạng JSON hoặc BSON (Binary JSON), cho phép dễ dàng lưu các cấu trúc phức tạp. Điểm nổi bật của MongoDB là không cần schema cố định, giúp linh hoạt trong việc thay đổi cấu trúc dữ liệu mà không ảnh hưởng đến hệ thống. Hệ thống này hỗ trợ khả năng mở rộng ngang (horizontal scaling), rất phù hợp cho các ứng dụng lớn với lượng dữ liệu khổng lồ. Ngoài ra, MongoDB cung cấp các tính năng mạnh mẽ như truy vấn linh hoạt, chỉ mục (indexing), sao chép dữ liệu (replication), và sharding để đảm bảo hiệu suất và độ tin cậy cao trong môi trường phân tán. MongoDB được ứng dụng rộng rãi trong các lĩnh vực như phát triển web, big data, và hệ thống thời gian thực nhờ vào tính linh hoạt và hiệu năng cao của nó.

## 4.2. Cài đặt chương trình

### 4.2.1. Yêu cầu hệ thống

Để có thể triển khai được chương trình thì yêu cầu máy tính cần phải cài đặt một số IDE, công cụ sau:

* Docker Compose
* VSCode
* Android Studio
* Git
* Linux

**Docker Compose**Docker Compose là một công cụ giúp bạn định nghĩa và quản lý các ứng dụng container hóa. Với Docker Compose, bạn có thể dễ dàng cấu hình và triển khai nhiều container liên quan trong một dự án bằng cách sử dụng tệp docker-compose.yml. Công cụ này giúp đơn giản hóa việc phát triển và triển khai ứng dụng, đặc biệt là các ứng dụng có nhiều thành phần phụ thuộc.

**Visual Studio Code (VSCode)**Visual Studio Code là một trình soạn thảo mã nguồn phổ biến và mạnh mẽ, hỗ trợ nhiều ngôn ngữ lập trình và framework. Với hệ thống plugin phong phú, giao diện thân thiện, và tính năng như IntelliSense, gỡ lỗi trực tiếp, VSCode giúp tăng hiệu suất phát triển phần mềm đáng kể.

**Android Studio**Android Studio là môi trường phát triển tích hợp (IDE) chính thức được Google cung cấp để phát triển ứng dụng Android. IDE này cung cấp nhiều công cụ mạnh mẽ như trình giả lập Android, gợi ý mã thông minh, trình gỡ lỗi, và hỗ trợ tích hợp Gradle, giúp lập trình viên dễ dàng phát triển, kiểm thử và triển khai ứng dụng Android.

**Git**Git là một hệ thống quản lý mã nguồn phân tán, được sử dụng rộng rãi trong việc phát triển phần mềm. Công cụ này cho phép theo dõi các thay đổi trong mã nguồn, hỗ trợ làm việc nhóm và quản lý phiên bản hiệu quả. Git giúp lập trình viên dễ dàng hợp tác và duy trì mã nguồn một cách an toàn và đáng tin cậy.

### 4.2.2. Cài đặt chương trình

#### **4.2.2.1. Cài đặt hệ thống Backend và Admin web và PHPAdmin**

Bước 1: Tạo 1 file *.env* có nội dung như sau:

MYSQL\_USERNAME = YOUR\_DATA

APP\_HOST = YOUR\_DATA

MYSQL\_PASSWORD = YOUR\_DATA

MYSQL\_HOST = YOUR\_DATA

MYSQL\_PORT = YOUR\_DATA

MYSQL\_DB = YOUR\_DATA

AUTHENTICATION\_KEY=YOUR\_DATA

JWT\_SECRET\_KEY=YOUR\_DATA

JWT\_TOKEN\_LOCATION=[headers]

Trong đó bạn sẽ truyền value là các giá trị phù hợp với thiết bị, với database nhằm mục đích bảo vệ những thông tin cá nhân khi chia sẻ mã nguồn với người khác, tăng tính bảo mật cho phần mềm.

Bước 2: Tạo file *docker-compose.yml* có nội dung như sau:

version: '3.8'

services:

mongo:

image: mongo:latest

container\_name: mongo

ports:

- "27017:27017"

volumes:

- mongo-data:/data/db

mysql:

image: mysql:9.1.0

container\_name: datn-mysql

environment:

MYSQL\_ROOT\_PASSWORD: ${MYSQL\_PASSWORD}

MYSQL\_DATABASE: datn

ports:

- "3306:3306"

volumes:

- mysql\_data:/var/lib/mysql # Lưu dữ liệu của MySQL

restart: always

networks:

- flask\_network

redis:

image: redis:7.4.1

container\_name: datn-redis

ports:

- "6379:6379"

restart: always

networks:

- flask\_network

phpmyadmin:

image: phpmyadmin:latest

container\_name: phpmyadmin

environment:

PMA\_HOST: mysql

PMA\_USER: ${MYSQL\_USERNAME}

PMA\_PASSWORD: ${MYSQL\_PASSWORD}

ports:

- "8080:80"

depends\_on:

- mysql

networks:

- flask\_network

flask\_app:

image: datn-be

ports:

- "5001:5001"

build:

context: .

dockerfile: Dockerfile.backend

environment:

- MYSQL\_USERNAME = ${MYSQL\_USERNAME}

- APP\_HOST = ${APP\_HOST}

- MYSQL\_PASSWORD = ${MYSQL\_PASSWORD}

- MYSQL\_HOST = ${MYSQL\_HOST}

- MYSQL\_PORT = ${MYSQL\_PORT}

- MYSQL\_DB = ${MYSQL\_DB}

- AUTHENTICATION\_KEY = ${AUTHENTICATION\_KEY}

- JWT\_SECRET\_KEY = ${JWT\_SECRET\_KEY}

- JWT\_TOKEN\_LOCATION = ${JWT\_TOKEN\_LOCATION}

volumes:

- ./static:/app/static

depends\_on:

- redis

- mysql

- mongo

restart: unless-stopped

networks:

- flask\_network

frontend:

build:

context: .

dockerfile: Dockerfile.frontend # Chỉ định Dockerfile cho frontend

container\_name: datn\_admin

volumes:

- ./static:/app/static

environment:

- MYSQL\_USERNAME = ${MYSQL\_USERNAME}

- APP\_HOST = ${APP\_HOST}

- MYSQL\_PASSWORD = ${MYSQL\_PASSWORD}

- MYSQL\_HOST = ${MYSQL\_HOST}

- MYSQL\_PORT = ${MYSQL\_PORT}

- MYSQL\_DB = ${MYSQL\_DB}

- AUTHENTICATION\_KEY = ${AUTHENTICATION\_KEY}

- JWT\_SECRET\_KEY = ${JWT\_SECRET\_KEY}

- JWT\_TOKEN\_LOCATION = ${JWT\_TOKEN\_LOCATION}

ports:

- "8501:8501"

depends\_on:

- flask\_app

networks:

- flask\_network

networks:

flask\_network:

driver: bridge

volumes:

mysql\_data:

mongo-data:

Bước 2: Cài đặt chương trình

Cách 1: Triển khai không cần mã nguồn:

Người dùng sẽ cần chạy 3 lệnh

docker pull teikk/datn-be:lastest

docker pull teikk/datn-admin:lastest

docker compose up -d

Trong đó 2 lệnh đầu tiên dùng để pull image chứa chương trình về máy tính.

Lệnh cuối dùng để chạy chương trình dưới nên thông qua các mô tả, thông tin đã viết thông qua bước 1 và 2.

Cách 2:

Người dùng clone mã nguồn từ github: <https://github.com/22Teikk/DATN_BE.git>

Người dùng chạy 1 trong 2 lệnh sau để chạy chương trình:

bash init.sh

docker compose up --build -d

Trong đó cách chạy đầu tiên sẽ tự động chạy các câu lệnh trong file .sh để chạy chương trình, là cách đơn giản nhất để chạy các ứng dụng liên quan đến flask và streamlit.

Cách chạy thứ 2 sẽ chạy chương trình là các container trong docker thông qua việc tự động build các images, cách này sẽ dễ dàng hơn khi có thể tích hợp cho nhiều máy và nhiều môi trường khác nhau.

Bước 3: Sau khi chương trình được chạy thì tất cả sẽ được triển khai trên nền web với địa chỉ mặc định khi không thay đổi là:

* API: <http://localhost:5001/>
* Admin Web: <http://localhost:8501/>
* PHPAdmin: <http://localhost:8080>

Lưu ý: Trước khi cài đặt chương trình bạn cần kiểm tra xem các port cần dùng này của máy tính có đang chạy không. Nếu có thì bạn cần phải kill port đi trước khi cài đặt thông qua lệnh:

bash kill.sh <Tên Port>

#### **4.2.2.2. Cài đặt ứng dụng cho phía customer và employee**

Bước 1: Clone mã nguồn từ trang chủ của github

Phía Customer: https://github.com/22Teikk/DATN.git

Phía Employee: <https://github.com/22Teikk/DATN_Admin.git>

Bước 2: Mở file local.properties để thêm Google Map API Key vào để có thể sử dụng tính năng GoogleMap theo dõi đơn hàng theo thời gian thực.

MAP\_API\_KEY=YOUR\_API\_KEY

Bước 3: Mở Project bằng Android Studio, sau đó nhấn Run chương trình sẽ tự động được cài đặt trên thiết bị Android của bạn.

Lưu ý: Nếu bạn muốn chạy trên server chứa API của riêng bạn ở trên thì cần thay đổi lại đường dẫn trong file RemoteDataModule.kt

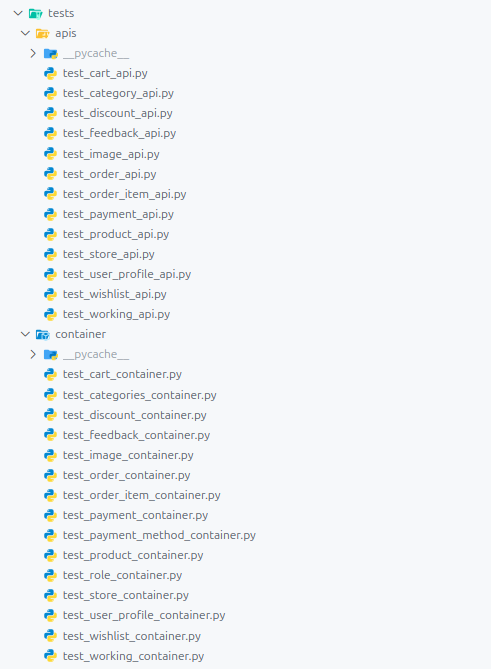
const val BASE\_URL = "http://94.237.64.46:5001/"

## 4.3. Kiểm thử phần mềm

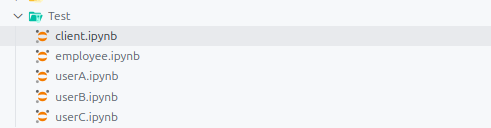
### 4.3.1. Kiểm thử Unit

Sử dụng sẵn thư viện pytest và ipynb để thực hiện kiểm thử Unit và socket cho hệ thống backend.

Cấu trúc thư mục test cho hệ thống backend

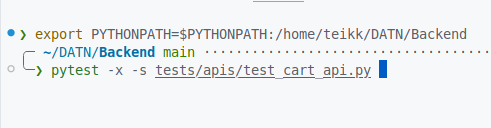


*Hình 4. 3 Cấu trúc thư mục test unit*



*Hình 4. 4 Cấu trúc thư mục test Socket*

Triển khai test thông qua pytest

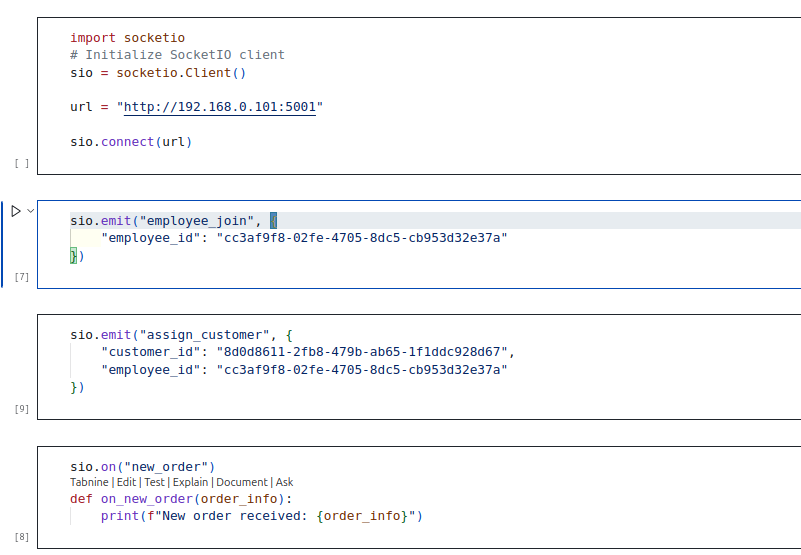


*Hình 4. 5 Câu lệnh chạy test unit*

Trong đó:

* export đường dẫn của thư mục cần test để python biết được vị trí của thư mục cần test
* Chạy lệnh pytest + tên file hoặc thư mục cần test để thực hiện test.

Triển khai test socket thông qua file ipynb

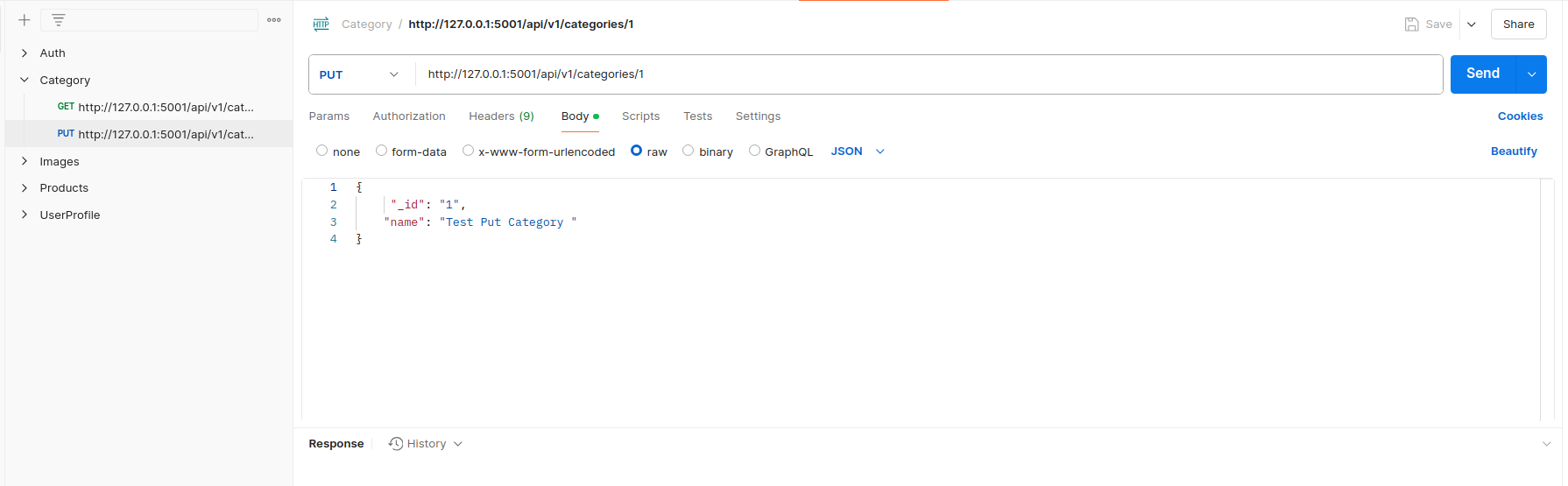


*Hình 4. 6 Triển khai test Socket thông qua file ipynb*

Để chạy file test socket thì ta sẽ click vào biểu tượng run trên từng cell.

### 4.3.2. Kiểm thử API

Sử dụng công cụ Postman để kiểm thử API theo kiến trúc restful



*Hình 4. 7 Triển khai test API thông qua postman*

Bấm nút send để có thể chạy test và nhận kết quả trả về.

# Chương 4. Kết luận

## 4.1. Tóm tắt kết quả nghiên cứu:

Đề tài "Xây dựng hệ thống đặt hàng theo thời gian thực" của em đã hoàn thiện và đạt được các kết quả quan trọng. Đề tài đã giúp em nghiên cứu và làm rõ quy trình nghiệp vụ của hệ thống bán hàng trực tuyến, từ việc phân tích các tài liệu, vẽ biểu đồ use case tổng quát, phân rã các use case chi tiết, và xây dựng các đặc tả yêu cầu cho từng use case. Hệ thống đã được phát triển với các biểu đồ hoạt động, trình tự, trạng thái, và biểu đồ lớp để từ đó thiết kế cơ sở dữ liệu hợp lý và tối ưu.

Hệ thống đạt được các yêu cầu cơ bản, đảm bảo các chức năng như quản lý món ăn, giỏ hàng, và đơn hàng dễ sử dụng, giúp người dùng dễ dàng quản lý và thao tác. Giao diện trang chủ được thiết kế bắt mắt, trực quan, dễ sử dụng và giao diện quản lý thân thiện với người dùng.

## 4.2. Đánh giá với mục tiêu ban đầu:

Mục tiêu ban đầu của đề tài là xây dựng một hệ thống đặt hàng trực tuyến có khả năng xử lý đơn hàng theo thời gian thực, đồng thời tối ưu hóa các chức năng và đảm bảo tính tiện dụng cho người dùng. Trong quá trình thực hiện, hệ thống đã hoàn thành các chức năng chính như đăng ký, đăng nhập, xem món ăn, tìm kiếm món ăn, quản lý giỏ hàng và đặt hàng. Tuy nhiên, do hạn chế về thời gian và kiến thức, hệ thống chưa hoàn thiện được các chức năng như quên mật khẩu, thanh toán online và xử lý các trường hợp ngoại lệ. Cấu trúc của chương trình vẫn còn một số điểm cần tối ưu hóa.

Tuy nhiên, kết quả đạt được vẫn tương đối đầy đủ và hoàn thiện so với mục tiêu ban đầu, tạo ra một nền tảng có thể phát triển và mở rộng thêm trong tương lai.

## 4.3. Đề xuất hướng phát triển tương lai

- Tăng cường bảo mật: Cần cải thiện cơ chế bảo mật hệ thống để ngăn chặn các rủi ro mất mát thông tin quan trọng của người dùng, bao gồm việc mã hóa dữ liệu và áp dụng các biện pháp bảo mật nâng cao.

- Quản lý quyền truy cập: Đề xuất xây dựng một cơ chế phân quyền mạnh mẽ hơn để chỉ những người có quyền mới có thể truy cập vào các dữ liệu nhạy cảm của hệ thống.

- Ứng dụng trí tuệ nhân tạo (AI): Sử dụng AI để phân tích và dự đoán nhu cầu của khách hàng, tối ưu hóa quy trình đặt hàng và phản hồi tự động với người dùng.

- Hoàn thiện chức năng quên mật khẩu và thanh toán online để hệ thống trở nên hoàn chỉnh hơn.

- Xây dựng cơ chế phân quyền linh hoạt hơn cho phép quản lý người dùng hiệu quả hơn.

- Phát triển chức năng chat trực tuyến để tăng cường giao tiếp giữa người bán và người mua.

# Tài liệu tham khảo

[1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, *Nhập môn công nghệ phần mềm*, NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.

[3] Martin, Robert C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.

[4] Trang hệ thống tham khảo: <https://shopeefood.vn/>

[5] Trang tài liệu của Streamlit: [https://docs.streamlit.io/](%20https:/docs.streamlit.io/)

[6] Trang tài liệu của Flask: [https://flask.palletsprojects.com/en/stable/](%20https:/flask.palletsprojects.com/en/stable/)

[7] Trang tài liệu của Docker: <https://docs.docker.com/>

[8] Trang tài liệu của Android: [https://developer.android.com/develop](%20https:/developer.android.com/develop%20)

[9] Trang tài liệu của SocketIO: [https://socket.io/docs/v4/](%20https:/socket.io/docs/v4/)

[10] Trang tin tức về lập trình <https://viblo.asia>