BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

⎯⎯⎯⎯⎯⎯✧⎯⎯⎯⎯⎯⎯

A yellow and red logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**XÂY DỰNG WEBSITE QUẢN LÝ VÀ ĐẶT PIZZA**

**CHO CỬA HÀNG PIZZING**

|  |  |
| --- | --- |
| GVHD: | TS. Phạm Văn Hiệp |
| Sinh viên:  Mã sinh viên: | Nguyễn Đức Xuân  2021605089 |
| Lớp: | CNTT05 |
| Khóa: | 16 |

*Hà Nội – 2025*

# LỜI NÓI ĐẦU

Đầu tiên em xin gửi lời cảm ơn đến các thầy, cô giảng viên của trường Đại học Công nghiệp Hà Nội nói chung và các thầy cô khoa Công nghệ thông tin nói riêng đã tận tình giúp đỡ, giảng dạy truyền đạt cho em những kiến thức, kinh nghiệm quý giá trong suốt thời gian học tập tại trường.

Và xem xin gửi lời cảm ơn đến Ts. Phạm Văn Hiệp đã tận tình giúp đỡ, hướng dẫn em trong quá trình thực hiện đồ án tốt nghiệp. Trong quá trình học tập và làm việc với thầy, em đã không ngừng tiếp thu thêm nhiều kiến thức bổ ích và tinh thần làm việc. Đây là những điều rất bổ ích cho em trong quá trình học tập và làm việc sau này.

Trong quá trình thực hiện đồ án tốt nghiệp, em khó tránh khỏi sai sót, rất mong thầy có thể thông cảm cho em. Em rất mong nhận được những đóng góp, nhận xét của các thầy cô để bài báo cáo đồ án tốt nghiệp của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI NÓI ĐẦU 2](#_Toc194252810)

[MỤC LỤC 3](#_Toc194252811)

[DANH MỤC HÌNH ẢNH 5](#_Toc194252812)

[DANH MỤC BẢNG 6](#_Toc194252813)

[MỞ ĐẦU 7](#_Toc194252814)

[1. Lý do chọn đề tài 7](#_Toc194252815)

[2. Giới thiệu chung 8](#_Toc194252816)

[2.1 Mục tiêu 8](#_Toc194252817)

[2.2 Phạm vi 8](#_Toc194252818)

[2.3 Bố cục 8](#_Toc194252819)

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN 9](#_Toc194252820)

[1.1 Giới thiệu tổng quan 9](#_Toc194252821)

[1.2 Nghiên cứu hiện trạng và giải quyết vấn đề 9](#_Toc194252822)

[1.2.1 Hiện trạng của cửa hàng 9](#_Toc194252823)

[1.2.2 Cơ cấu tổ chức 10](#_Toc194252824)

[1.2.3 Mô tả hoạt động của các bộ phận 10](#_Toc194252825)

[1.2.4 Xác định yêu cầu 10](#_Toc194252826)

[1.3 Giới thiệu về công nghệ, ngôn ngữ sử dụng 10](#_Toc194252827)

[1.3.1 Giới thiệu về ngôn ngữ Java 10](#_Toc194252828)

[1.3.2 Giới thiệu về mô hình Restful API 11](#_Toc194252829)

[1.3.3 Tổng quan về ngôn ngữ 12](#_Toc194252830)

[1.3.4 Cơ sở dữ liệu MySQL 13](#_Toc194252831)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 15](#_Toc194252832)

[2.1 Khảo sát hệ thống 15](#_Toc194252833)

[2.1.1 Mục tiêu 15](#_Toc194252834)

[2.1.2 Giải pháp 15](#_Toc194252835)

[2.1.3 Tổng quan về hệ thống 18](#_Toc194252836)

[2.1.4 Hoạt động của hệ thống 18](#_Toc194252837)

[2.2 Xác định các tác nhân của hệ thống, ca sử dụng 19](#_Toc194252838)

[2.2.1 Các tác nhân 19](#_Toc194252839)

[2.2.2 Các ca sử dụng (usecase) 20](#_Toc194252840)

[2.3 Các biểu đồ 21](#_Toc194252841)

[2.3.1 Biểu đồ use case tổng quát 21](#_Toc194252842)

[2.3.2 Biểu đồ use case phân rã 21](#_Toc194252843)

[2.4 Mô tả chi tiết các Usecase 23](#_Toc194252844)

[2.5 Xây dựng biểu đồ trình tự 33](#_Toc194252845)

[2.6 Thiết kế biểu đồ lớp chi tiết 34](#_Toc194252846)

[2.7 Cơ sở dữ liệu 35](#_Toc194252847)

[2.7.1 Sơ đồ dữ liệu quan hệ 35](#_Toc194252848)

[2.7.2 Chi tiết các bảng dữ liệu 36](#_Toc194252849)

[2.8 Thiết kế giao diện 40](#_Toc194252850)

[CHƯƠNG 3: CÀI ĐẶT VÀ TRIỂN KHAI 41](#_Toc194252851)

[3.1 Giới thiệu về công cụ và môi trường cài đặt 41](#_Toc194252852)

[3.1.1 Công cụ lập trình Intelij, Visual Studio Code 41](#_Toc194252853)

[3.1.2 Môi trường Java, TypeScript 41](#_Toc194252854)

[3.1.3 Thư viện Java Spring Boot, Angular 41](#_Toc194252855)

[3.2 Trang màn hình giao diện của khách hàng (Front-end) 41](#_Toc194252856)

[3.3 Trang màn hình giao diện của người quản trị (Back-end): 41](#_Toc194252857)

[3.4 Một số giao diện khác 41](#_Toc194252858)

[3.5 Kiểm thử 41](#_Toc194252859)

[3.5.1 Lập kế hoạch kiểm thử 41](#_Toc194252860)

[3.5.2 Thiết kế testcase 41](#_Toc194252861)

[3.5.3 Thực thi testcase 41](#_Toc194252862)

[3.5.4 Báo cáo kiểm thử 41](#_Toc194252863)

[KẾT LUẬN 41](#_Toc194252864)

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG

# MỞ ĐẦU

# Lý do chọn đề tài

Trong nhiều năm trở lại đây, xu hướng số hóa toàn cầu ở mọi lĩnh vực, mọi ngành nghề đã và đang diễn ra mạnh mẽ, tạo ra những thay đổi sâu rộng trong cách con người làm việc, kinh doanh và xử lý các vấn đề. Sự phát triển của công nghệ thông tin, trí tuệ nhân tạo và dữ liệu lớn không chỉ thúc đẩy hiệu suất, mà còn mở ra những cơ hội đột phá trong quản lý, vận hành và trải nghiệm khách hàng. Trong bối cảnh đó, việc ứng dụng các giải pháp số hóa trở thành một yếu tố quan trọng giúp doanh nghiệp nâng cao năng lực cạnh tranh, tối ưu hóa quy trình và thích ứng với xu thế phát triển hiện đại.

Nhận thấy tiềm năng này, nhiều cửa hàng đã kết hợp, áp dụng mô hình bán hàng, quản lý online nhằm giúp khách hàng dễ dàng tìm kiếm, trao đổi thông tin, giao dịch một cách tiện lợi, nhanh chóng, ít tiêu hao chi phí và tiêu biểu là những doanh nghiệp, của hàng kinh doanh trong lĩnh vực thực phẩm, đồ ăn nhanh trong đó có cửa hàng kinh doanh pizza: PizZing.

Cửa hàng PizZing hiện đang phục vụ một lượng lớn khách hàng với nhu cầu đặt hàng nhanh chóng, tiện lợi. Tuy nhiên, phương thức quản lý và đặt hàng thủ công hoặc qua các nền tảng trung gian còn nhiều hạn chế như mất thời gian, dễ xảy ra sai sót và khó kiểm soát thông tin đơn hàng. Đề tài: **"Xây dựng website quản lý và đặt pizza cho cửa hàng PizZing"** nhằm đáp ứng nhu cầu số hóa trong quá trình bán hàng, quản lý, quảng bá, tiếp cận khách hàng cho cửa hàng và nhu cầu tìm kiếm thông tin, mua sắm cho người mua hàng. Với website là một trong những công cụ hữu hiệu đáp ứng những yêu cầu và thuận tiện cho quá trình quản lý và mua hàng, giúp khách hàng dễ dàng tiếp cận dịch vụ mọi lúc, mọi nơi. Do đó, việc xây dựng một website quản lý và đặt pizza chuyên biệt cho cửa hàng không chỉ giúp khách hàng dễ dàng lựa chọn, đặt hàng trực tuyến mà còn hỗ trợ cửa hàng trong việc quản lý đơn hàng, doanh thu một cách hiệu quả hơn.

Và dưới sự hướng dẫn tận tâm của giảng viên **Ts.Phạm Văn Hiệp**, em đã thực hiện đồ án ""**Xây dựng website quản lý và đặt pizza cho cửa hàng PizZing**" với mục tiêu giúp chủ cửa hàng có thể quản lý và giới thiệu sản phẩm của mình một cách dễ dàng thông qua nền tảng trực tuyến, khách hàng có thể đặt pizza, đặt bàn từ xa mà không cần phải đến tận nơi, và chủ cửa hàng có thể dõi dõi toàn bộ quá trình vận hành của doanh nghiệp 1 cách suôn sẻ mọi lúc mọi nơi.

# Giới thiệu chung

## 2.1 Mục tiêu

* Xây dựng các chức năng cơ bản cho một hệ thống thương mại điện tử.
* Website có khả năng tương thích, hiẻn thị và hoạt động tốt trên tất cả các thiết bị hiện có hiện nay và có thể nâng cấp trong tương lai.
* Website đảm bảo, đáp ứng các chức năng cần có, cung cấp nhu cầu quản lý, đặt hàng trực tuyến
* Nắm được các công nghệ thiết kế web hiện đại, thực tế: Java, TypeScript, các thư viện: Java Spring Boot, Angular…

## 2.2 Phạm vi

PizZing là một cửa hàng kinh doanh pizza với hình thức phục vụ tại chỗ và bán mang đi. Hiện tại, cửa hàng hoạt động chủ yếu dựa trên phương thức đặt hàng trực tiếp tại quầy hoặc qua điện thoại.

Hệ thống website này dựa vào các chức năng cần thiết và được xây dựng cho cửa hàng PizZing.

## 2.3 Bố cục

Kết cấu báo cáo gồm 4 phần :

- Phần 1: Chương 1: Giới thiệu tổng quan

- Phần 2: Chương 2: Phân tích thiết kế hệ thống

- Phần 3: Chương 3: Cài đặt và triển khai

- Phần 4 : Kết luận

## 

# CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

## **Giới thiệu tổng quan**

Dự án xây dựng hệ thống website quản lý và đặt pizza nhằm giải quyết bài toán đặt hàng và quản lý đơn hàng trực tuyến cho cửa hàng PizZing, giúp khách hàng dễ dàng đặt pizza mọi lúc, mọi nơi mà không cần đến trực tiếp cửa hàng. Đồng thời, đây cũng là cơ hội để cửa hàng mở rộng mô hình kinh doanh và tiếp cận nhiều khách hàng tiềm năng hơn.

Mục tiêu của hệ thống là xây dựng một nền tảng đặt hàng trực tuyến thuận tiện, giúp khách hàng dễ dàng lựa chọn sản phẩm, đặt hàng nhanh chóng và thanh toán linh hoạt. Dự án sẽ tập trung vào việc thiết kế giao diện thân thiện với người dùng và xây dựng hệ thống quản lý đơn hàng hiệu quả.

Hệ thống website hướng tới đối tượng chính là khách hàng có nhu cầu đặt pizza trực tuyến. Người dùng có thể lựa chọn loại pizza, kích thước, combo đi kèm và thanh toán trực tuyến, giúp tối ưu trải nghiệm mua hàng và nâng cao hiệu suất hoạt động của cửa hàng.

## **Nghiên cứu hiện trạng và giải quyết vấn đề**

### Hiện trạng của cửa hàng

PizZing là một cửa hàng kinh doanh pizza với hình thức phục vụ tại chỗ và bán mang đi. Hiện tại, cửa hàng hoạt động chủ yếu dựa trên phương thức đặt hàng trực tiếp tại quầy hoặc qua điện thoại. Tuy nhiên, việc quản lý đơn hàng, nguyên liệu và doanh thu vẫn còn thực hiện thủ công hoặc thông qua các ứng dụng đơn giản, dẫn đến một số hạn chế như:

* Thời gian xử lý đơn hàng lâu, dễ xảy ra sai sót khi ghi nhận thông tin khách hàng.
* Không có hệ thống theo dõi và quản lý doanh thu hiệu quả.
* Chưa có nền tảng trực tuyến chính thức để khách hàng đặt hàng, dẫn đến phụ thuộc vào các ứng dụng giao hàng trung gian với mức phí cao.

### Cơ cấu tổ chức

Cửa hàng PizZing hiện tại được tổ chức theo các số bộ phận với các chức năng đơn giản như: bộ phận bán hàng gồm bộ phận quản lý, bộ phận bếp, bộ phận phục vụ, bộ phận thu ngân, bộ phận chăm sóc khách hàng.

### Mô tả hoạt động của các bộ phận

* Bộ phận quản lý: Chịu trách nhiệm giám sát hoạt động kinh doanh, quản lý doanh thu, kiểm soát chất lượng dịch vụ và ra quyết định chiến lược.
* Bộ phận bếp: Bao gồm các đầu bếp và nhân viên bếp chuyên chế biến pizza, đảm bảo chất lượng món ăn.
* Bộ phận phục vụ và giao hàng: Tiếp nhận đơn hàng trực tiếp tại cửa hàng, hỗ trợ khách hàng và đảm bảo giao hàng đúng thời gian.
* Bộ phận thu ngân: Quản lý thanh toán, ghi nhận doanh thu và xử lý các vấn đề liên quan đến hóa đơn.
* Bộ phận chăm sóc khách hàng: Tiếp nhận phản hồi, giải đáp thắc mắc và hỗ trợ khách hàng khi cần.

### Xác định yêu cầu

Dựa trên hiện trạng của cửa hàng, yêu cầu đối với hệ thống website bao gồm:

* Hỗ trợ quản lý danh mục sản phẩm.
* Theo dõi đơn hàng, phục vụ đặt bàn và khách ăn tại bàn
* Cung cấp giỏ hàng và thanh toán trực tuyến cho khách hàng.
* Theo dõi và quản lý đơn hàng, giao nhận hàng online.
* Hệ thống chăm sóc khách hàng trực tuyến qua live chat hoặc email.

## **Giới thiệu về công nghệ, ngôn ngữ sử dụng**

### Giới thiệu về ngôn ngữ Java

Java là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Sun Microsystems vào năm 1995 và hiện thuộc sở hữu của Oracle. "Viết một lần, chạy mọi nơi", Java cho phép các chương trình được biên dịch một lần và chạy trên nhiều nền tảng khác nhau thông qua máy ảo Java (JVM). Ngôn ngữ Java được sử dụng rộng rãi trong nhiều lĩnh vực như ứng dụng web, phần mềm doanh nghiệp, thiết bị nhúng và các ứng dụng di động trên nền tảng Android.

Java sở hữu nhiều tính năng nổi bật như tính hướng đối tượng, giúp tổ chức mã nguồn một cách linh hoạt và dễ bảo trì. Java hỗ trợ thu gom rác tự động giúp quản lý bộ nhớ hiệu quả, có khả năng bảo mật cao và chạy được trên nhiều nền tảng khác nhau nhờ vào JVM. Bên cạnh đó, Java cung cấp một thư viện phong phú, hỗ trợ nhiều chức năng như xử lý tập tin, mạng, giao tiếp cơ sở dữ liệu và giao diện người dùng. Cộng đồng phát triển Java lớn mạnh cũng là một lợi thế khi học tập và phát triển phần mềm.

Ưu điểm của Java bao gồm cú pháp rõ ràng, dễ học, phù hợp với người mới bắt đầu cũng như lập trình viên chuyên nghiệp. Khả năng mở rộng và tái sử dụng mã của Java giúp phát triển ứng dụng quy mô lớn một cách hiệu quả. Bên cạnh đó, Java có khả năng hoạt động ổn định và đáng tin cậy trên nhiều nền tảng khác nhau. Ngoài ra, nhờ vào hệ sinh thái phong phú và tài liệu hướng dẫn đa dạng, lập trình viên dễ dàng tiếp cận và giải quyết vấn đề khi làm việc với Java.

Tuy vậy, Java cũng có một số nhược điểm như tốc độ xử lý không nhanh bằng các ngôn ngữ biên dịch trực tiếp như C hoặc C++, do phải chạy thông qua máy ảo. Ứng dụng viết bằng Java thường tiêu tốn nhiều bộ nhớ hơn, đặc biệt trong các hệ thống lớn. Các thư viện giao diện người dùng như Swing hoặc AWT có phần lỗi thời, giao diện không hiện đại và khó tùy biến. Ngoài ra, Java thường yêu cầu viết nhiều mã lệnh hơn cho cùng một chức năng so với các ngôn ngữ lập trình cấp cao hơn như Python, khiến việc phát triển đôi khi kém linh hoạt hơn.

### Giới thiệu về mô hình Restful API

Restful API là một kiểu kiến trúc được sử dụng phổ biến trong việc xây dựng các dịch vụ web, cho phép các hệ thống khác nhau giao tiếp với nhau thông qua giao thức HTTP. REST là viết tắt của “Representational State Transfer” – một chuẩn thiết kế đơn giản, nhẹ và dễ mở rộng. Restful API sử dụng các phương thức HTTP cơ bản như GET, POST, PUT, DELETE để thao tác với tài nguyên được đại diện bởi các URL. Mỗi tài nguyên thường được định danh duy nhất bằng một đường dẫn và phản hồi dưới dạng dữ liệu như JSON hoặc XML.

Restful API có một số đặc điểm chính như không trạng thái (stateless), tức là mỗi yêu cầu từ client gửi đến server đều độc lập và không lưu trữ trạng thái giữa các phiên làm việc. Ngoài ra, Restful API thường được thiết kế theo nguyên tắc phân chia rõ ràng giữa client và server, giúp hệ thống dễ mở rộng, bảo trì và triển khai độc lập. Dữ liệu thường được truyền qua định dạng JSON, gọn nhẹ, dễ đọc và dễ xử lý trong hầu hết các ngôn ngữ lập trình hiện nay.

Ưu điểm của Restful API là đơn giản, dễ hiểu, dễ triển khai và có thể sử dụng trên nhiều nền tảng khác nhau. Nhờ sử dụng giao thức HTTP phổ biến, Restful API có thể hoạt động tốt với hầu hết các trình duyệt, thiết bị di động và các hệ thống backend. Việc sử dụng định dạng dữ liệu như JSON giúp giảm kích thước gói tin, tăng tốc độ xử lý. Hơn nữa, Restful API dễ dàng tích hợp với các framework và thư viện hiện đại, giúp tăng hiệu quả phát triển phần mềm.

Tuy nhiên, Restful API cũng có một số nhược điểm như không hỗ trợ tốt các tương tác phức tạp hoặc yêu cầu duy trì kết nối lâu dài giữa client và server. Vì không lưu trạng thái, nên client cần gửi đầy đủ thông tin trong mỗi lần gọi, dẫn đến dữ liệu bị lặp và tiêu tốn băng thông hơn trong một số trường hợp. Ngoài ra, việc đảm bảo bảo mật như xác thực, phân quyền hoặc chống giả mạo (CSRF, XSS) đòi hỏi lập trình viên phải tự triển khai thêm các cơ chế bổ sung để đảm bảo an toàn cho hệ thống.

### Tổng quan về ngôn ngữ

Giao diện người dùng: Website sử dụng Angular – một framework phát triển bởi Google – kết hợp với ngôn ngữ TypeScript để xây dựng phần giao diện người dùng. Angular cho phép phát triển các ứng dụng web một cách cấu trúc và có tổ chức, hỗ trợ cơ chế component giúp tái sử dụng và quản lý mã nguồn hiệu quả. TypeScript, với khả năng kiểm tra kiểu tĩnh và cú pháp rõ ràng, giúp phát hiện lỗi sớm trong quá trình phát triển, đồng thời tăng cường khả năng bảo trì và mở rộng mã nguồn. Việc kết hợp Angular và TypeScript giúp giao diện website trở nên tương tác cao, phản hồi nhanh và thân thiện với người dùng.

Xử lý server: Ở phía máy chủ, website sử dụng ngôn ngữ lập trình Java cùng với IDE IntelliJ IDEA và framework Spring Boot – một nền tảng phát triển ứng dụng web hiện đại, nổi bật với khả năng tự động cấu hình, tích hợp sẵn nhiều thư viện cần thiết và dễ triển khai. Spring Boot hỗ trợ mạnh mẽ việc xây dựng các API RESTful, quản lý cơ sở dữ liệu, bảo mật và kiểm soát truy cập. Với Java – một ngôn ngữ hướng đối tượng ổn định, dễ bảo trì – việc phát triển backend trở nên rõ ràng, dễ tổ chức, đồng thời bảo đảm hiệu năng và tính bảo mật cho hệ thống. IntelliJ IDEA hỗ trợ lập trình viên với tính năng thông minh như gợi ý mã, kiểm tra lỗi thời gian thực và công cụ debug mạnh mẽ, giúp tăng tốc quá trình phát triển.

Sự kết hợp giữa Angular ở phía frontend và Spring Boot ở phía backend tạo nên một kiến trúc phân lớp rõ ràng, hiện đại và hiệu quả. Angular đảm nhận phần giao diện và trải nghiệm người dùng, trong khi Spring Boot chịu trách nhiệm xử lý logic nghiệp vụ, truy xuất dữ liệu và đảm bảo an toàn hệ thống. Hai công nghệ này giao tiếp với nhau thông qua các API RESTful, giúp tách biệt hoàn toàn frontend và backend, từ đó dễ dàng nâng cấp, bảo trì và mở rộng từng phần mà không ảnh hưởng đến toàn bộ hệ thống. Đây là sự kết hợp tối ưu, phù hợp với xu hướng phát triển web hiện đại và các hệ thống doanh nghiệp quy mô lớn.

### Cơ sở dữ liệu MySQL

MySQL MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, được phát triển ban đầu bởi công ty MySQL AB và hiện nay thuộc sở hữu của Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để quản lý, truy xuất và thao tác dữ liệu trong các bảng có quan hệ với nhau. Nhờ tính ổn định, dễ sử dụng và hiệu năng cao, MySQL trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất thế giới, được sử dụng rộng rãi trong các ứng dụng web, đặc biệt là trong mô hình LAMP (Linux, Apache, MySQL, PHP/Perl/Python).

MySQL cung cấp nhiều tính năng mạnh mẽ như hỗ trợ giao dịch (transaction), khóa ngoại (foreign key), chỉ mục (index), và khả năng mở rộng với các công cụ lưu trữ khác nhau như InnoDB hay MyISAM. Hệ thống còn hỗ trợ sao lưu, phục hồi dữ liệu, phân quyền người dùng và kiểm soát truy cập. MySQL cũng tương thích với nhiều hệ điều hành như Windows, Linux và macOS, đồng thời có thể tích hợp với nhiều ngôn ngữ lập trình như Java, PHP, Python, C#, giúp lập trình viên dễ dàng xây dựng và quản lý hệ thống dữ liệu.

Ưu điểm của MySQL bao gồm tốc độ truy vấn nhanh, hiệu suất ổn định và khả năng xử lý khối lượng dữ liệu lớn. Là một phần mềm mã nguồn mở, MySQL miễn phí cho hầu hết các mục đích sử dụng và có cộng đồng người dùng rộng lớn cung cấp hỗ trợ, tài liệu và công cụ hữu ích. Ngoài ra, MySQL có giao diện quản lý đồ họa như phpMyAdmin, giúp người dùng dễ dàng thao tác với cơ sở dữ liệu mà không cần viết nhiều câu lệnh SQL thủ công.

Tuy nhiên, MySQL cũng có những nhược điểm như khả năng hỗ trợ các tính năng nâng cao còn hạn chế so với các hệ quản trị cơ sở dữ liệu thương mại như Oracle Database hay Microsoft SQL Server. Trong các hệ thống yêu cầu giao dịch phức tạp hoặc dữ liệu phi quan hệ, MySQL có thể không phải là lựa chọn tối ưu. Ngoài ra, mặc dù hỗ trợ chuẩn SQL, nhưng một số hành vi xử lý truy vấn trong MySQL có thể khác biệt so với các hệ thống khác, đòi hỏi lập trình viên cần tìm hiểu kỹ khi xây dựng các hệ thống lớn và phức tạp.

Trong hệ thống website cho cửa hàng PizZing, MySQL giúp quản lý đơn hàng, sản phẩm, tài khoản người dùng và giao dịch.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## **Khảo sát hệ thống**

### 2.1.1 Mục tiêu

Mục đích của việc khảo sát người dùng cho đề tài **Xây dựng website quản lý và đặt pizza cho cửa hàng Pizzing** là nhằm đánh giá và phân tích các yếu tố ảnh hưởng đến hiệu quả sử dụng, mức độ hài lòng, cũng như khả năng đặt hàng trực tuyến của khách hàng đối với hệ thống. Qua việc khảo sát, nhóm phát triển có thể nắm bắt nhu cầu thực tế của người dùng, từ đó xây dựng các chức năng phù hợp, thuận tiện trong quá trình đặt pizza, lựa chọn kích cỡ, loại nhân, đồ uống đi kèm và theo dõi đơn hàng.

Bên cạnh đó, khảo sát hệ thống còn giúp kiểm tra các yêu cầu về tính bảo mật, tính ổn định và khả năng mở rộng của website, đảm bảo rằng dữ liệu người dùng được bảo vệ an toàn và các thao tác quản lý đơn hàng, menu hay doanh thu diễn ra trơn tru. Việc khảo sát kỹ lưỡng là bước quan trọng nhằm tạo ra một website có giao diện thân thiện, dễ sử dụng và đáp ứng được nhu cầu thực tế của cả khách hàng lẫn nhân viên cửa hàng, góp phần nâng cao chất lượng dịch vụ và trải nghiệm mua hàng trực tuyến tại Pizzing.

### 2.1.2 Giải pháp

* Khảo sát trực tiếp
* **Phiếu khảo sát 1:**

Người khảo sát: Nguyễn Đức Xuân

Người được khảo sát: Một người dùng

|  |  |
| --- | --- |
| Câu hỏi | Câu trả lời |
| Bạn đã từng đặt pizza qua các website trực tuyến chưa? (Có/Không) | Có |
| Nếu có, bạn thấy trải nghiệm của mình như thế nào? (Rất tốt/Tốt/Bình thường/Kém/Rất kém) | Bình thường |
| Các chức năng bạn mong muốn có trên website đặt pizza? | Tìm kiếm sản phẩm dễ dàng  Lựa chọn kích thước, loại vỏ tùy chọn  Đặt hàng và thanh toán an toàn, tiện lợi  Thanh toán an toàn và tiện lợi  Giao hàng nhanh chóng và đúng hẹn  Hỗ trợ khách hàng tận tình |

* **Phiếu khảo sát 2:**

Người khảo sát: Nguyễn Đức Xuân

Người được khảo sát: Trần Văn Minh (Quản lý 1 cửa hàng pizza)

|  |  |
| --- | --- |
| Câu hỏi | Câu trả lời |
| Bạn có yêu cầu gì về bảo mật của website? | Cần có chức năng đăng nhập/đăng ký tài khoản cho người dù  Quản lý tài khoản người dùng: chỉnh sửa, khóa/mở tài khoản, phân quyền |
| Bạn có yêu cầu gì về quản lý sản phẩm và danh mục của website? | Thêm, sửa, xóa các loại pizza và combo  Quản lý kích thước, loại vỏ và giá tiền |
| Bạn có yêu cầu gì về quản lý đơn hàng và thanh toán của website? | Xem và xác nhận đơn hàng  Cập nhật trạng thái đơn hàng (đang xử lý, đang giao, hoàn tất...)  Thống kê đơn hàng theo ngày, tuần, tháng. |
| Bạn có yêu cầu gì về thống kê và báo cáo của website? | Doanh thu, số lượng đơn hàng, sản phẩm bán chạy.  Báo cáo lỗi hệ thống và phản hồi từ người dùng. |
| Bạn có yêu cầu gì về Tối ưu hóa hiệu suất và tốc độ tải trang? | Tối ưu hóa tốc độ tải trang, mọi xử lý được cập nhật nhanh chóng, chính xác.  Tối ưu hóa mã nguồn: Đảm bảo mã nguồn hệ thống được viết gọn gàng, tối ưu hóa và tương thích trên nhiều thiết bị. |
| Bạn có yêu cầu gì về hỗ trợ khách hàng? | Hệ thống hỗ trợ trực tuyến: Cung cấp khả năng chat trực tuyến giữa người dùng và quản lý để hỗ trợ người dùng.  Trang FAQ và hỗ trợ thông tin: Tạo trang FAQ để giải đáp các câu hỏi thường gặp của người dùng. |

**Kết quả khảo sát:**

Mục tiêu của việc khảo sát trong đề tài **Xây dựng website quản lý và đặt pizza cho cửa hàng Pizzing** là nhằm thu thập thông tin, ý kiến và mong muốn từ người dùng và quản trị viên. Cụ thể, khảo sát hướng đến các mục tiêu sau:

Hiểu rõ nhu cầu và mong muốn của khách hàng khi sử dụng một website đặt pizza trực tuyến, bao gồm các tính năng, chức năng và trải nghiệm người dùng mà họ kỳ vọng.

Đánh giá mức độ hài lòng của người dùng đối với các website đặt đồ ăn hiện nay, từ đó rút ra điểm mạnh và điểm yếu để cải thiện giao diện, tốc độ, tính năng và độ tiện dụng của hệ thống.

Xác định các tính năng cần thiết mà người dùng mong muốn được tích hợp, chẳng hạn như đặt pizza theo yêu cầu (chọn topping, kích cỡ), thanh toán an toàn, theo dõi đơn hàng, và hỗ trợ trực tuyến.

Xây dựng cơ sở dữ liệu phản hồi từ người dùng và quản trị viên, nhằm hỗ trợ quá trình đánh giá hiệu quả hệ thống sau khi triển khai, đồng thời làm căn cứ để phát triển thêm các tính năng mới hoặc điều chỉnh chiến lược vận hành phù hợp với nhu cầu thực tế.

### 2.1.3 Tổng quan về hệ thống

Hệ thống **website quản lý và đặt pizza cho cửa hàng Pizzing** là một nền tảng thương mại điện tử giúp kết nối khách hàng với cửa hàng một cách nhanh chóng và thuận tiện. Thông qua hệ thống, khách hàng có thể lựa chọn loại pizza, combo, kích cỡ, topping và thực hiện thanh toán trực tuyến hoặc khi nhận hàng. Đồng thời, hệ thống còn hỗ trợ nhân viên trong việc xử lý đơn hàng và giúp quản trị viên giám sát toàn bộ hoạt động kinh doanh.

Hệ thống được thiết kế dựa trên mô hình client-server, vận hành trên trình duyệt web và tương thích với nhiều thiết bị (PC, điện thoại, tablet), hướng đến mục tiêu tối ưu hóa trải nghiệm người dùng và nâng cao hiệu quả vận hành của cửa hàng.

### Hoạt động của hệ thống

Hệ thống vận hành theo quy trình logic với các chức năng chính tương ứng cho từng vai trò:

* **Người dùng (khách hàng):**
  + Đăng ký, đăng nhập tài khoản.
  + Tìm kiếm và lựa chọn pizza, combo theo nhu cầu (kích thước, loại pizza)
  + Thêm sản phẩm vào giỏ hàng, tiến hành đặt hàng và thanh toán trực tuyến.
  + Theo dõi trạng thái đơn hàng (đang xử lý, đang giao, đã hoàn tất).
  + Liên hệ hỗ trợ và gửi đánh giá sau khi nhận hàng.
* **Nhân viên:**
  + Nhận thông báo khi có đơn hàng mới.
  + Kiểm tra chi tiết đơn hàng, xác nhận và chuyển trạng thái đơn hàng.
  + Cập nhật tiến trình vận đơn, phối hợp giao hàng hoặc xác nhận hoàn tất đơn.
  + Báo cáo các sự cố đơn hàng hoặc cập nhật thông tin giao hàng khi cần.
  + Nhận và xử lý đặt bàn
* **Quản trị viên (Admin):**
  + Quản lý thông tin sản phẩm (pizza, combo, loại pizza...), thêm/sửa/xóa.
  + Quản lý người dùng và nhân viên: phân quyền, chỉnh sửa, khóa/mở tài khoản.
  + Quản lý đơn hàng: giám sát tiến trình, can thiệp khi có vấn đề.
  + Xem thống kê: số lượng đơn hàng, sản phẩm bán chạy, doanh thu theo ngày/tháng.
  + Theo dõi và xử lý lỗi hệ thống, phản hồi từ người dùng.

Hệ thống được xây dựng theo mô hình client-server, đảm bảo hoạt động ổn định, tốc độ truy xuất nhanh và khả năng mở rộng linh hoạt trong tương lai.

## **Xác định các tác nhân của hệ thống, ca sử dụng**

### Các tác nhân

Hệ thống chia thành 3 tác nhân chính tương tác trực tiếp với các chức năng của hệ thống:

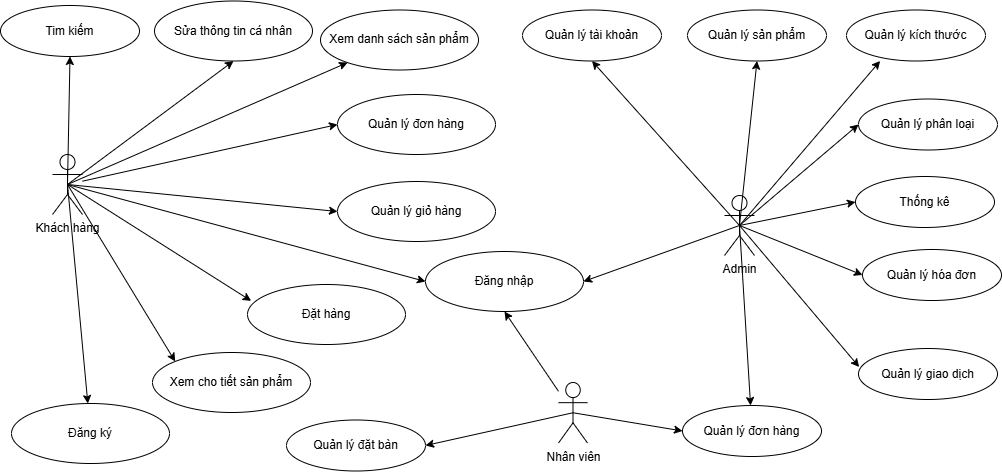
* **Người dùng (khách hàng):** Thực hiện các thao tác đặt hàng, lựa chọn loại pizza, combo, thanh toán và theo dõi đơn hàng.
* **Nhân viên:** Phụ trách tiếp nhận đơn hàng từ hệ thống, xác nhận và xử lý đơn, thực hiện các bước như nhận đơn đặt bàn, chuẩn bị đơn, cập nhật trạng thái giao hàng và hoàn tất đơn hàng.
* **Quản trị viên (Admin):** Quản lý toàn bộ hệ thống bao gồm người dùng, nhân viên, sản phẩm, danh mục, đơn hàng, và theo dõi thống kê doanh thu, hiệu suất hoạt động, cũng như xử lý các vấn đề kỹ thuật.

### Các ca sử dụng (usecase)

* Khách hàng:
* Đăng nhập
* Đăng ký
* Tìm kiếm
* Sửa thông tin cá nhân
* Xem danh sách sản phẩm
* Xem chi tiết sản phẩm
* Quản lý giỏ hàng: xóa sản phẩm khỏi giỏ hàng, thêm sản phẩm vào giỏ hàng, xem giỏ hàng
* Quản lý đặt hàng: thanh toán, theo dõi đơn hàng
* Đặt hàng
* Nhân viên:
* Đăng nhập
* Quản lý đơn hàng: cập nhật trạng thái đơn hàng, Xem danh sách đơn hàng, In hóa đơn.
* Quản lý đặt bàn: xem sơ đồ đặt bàn, cập nhật tình trạng bàn
* Admin:
* Đăng nhập
* Quản lý tài khoản: Thêm tài khoản, Xóa tài khoản, Thay đổi thông tin tài khoản
* Quản lý sản phẩm: Thêm sản phẩm, Sửa sản phẩm, Xóa sản phẩm
* Quản lý kích thước: Thêm kích thước, Sửa kích thước, Xóa kích thước
* Quản lý phân loại: Thêm phân loại, Sửa phân loại, Xóa phân loại
* Quản lý đơn hàng: Thêm đơn hàng, Cập nhật đơn hàng
* Quản lý giao dịch: Theo dõi giao dịch
* Quản lý thành viên: Thêm thành viên, Sửa thành viên, Xóa thành viên
* Quản lý hóa đơn Xác nhận hóa đơn
* Thống kê: Thống kê doanh thu, Thống kê tài khoản, Thống kê thành viên, Thống kê đơn hàng, Thống kê sản phẩm

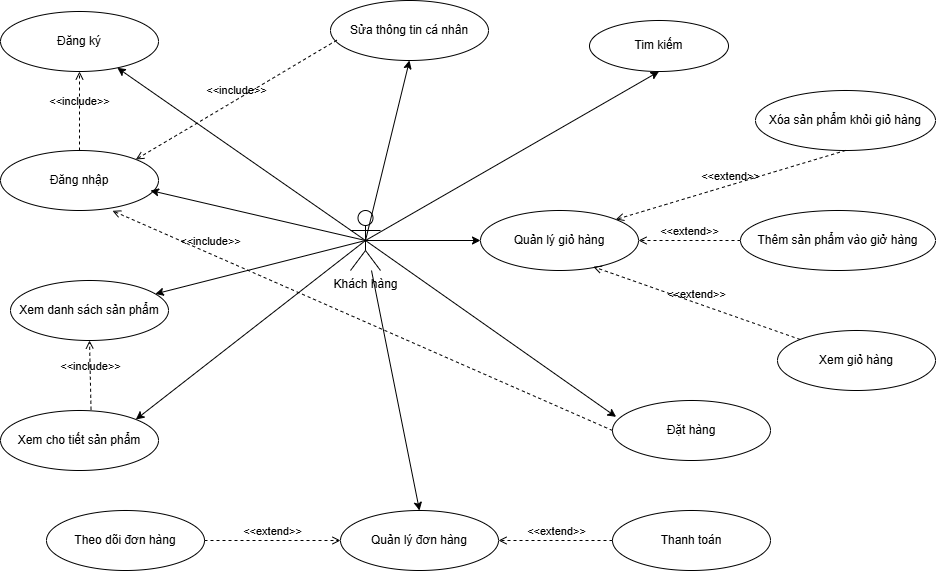
## **Các biểu đồ**

### Biểu đồ use case tổng quát

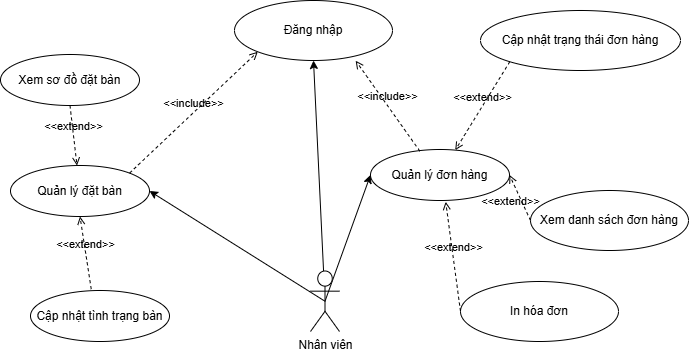


### Biểu đồ use case phân rã

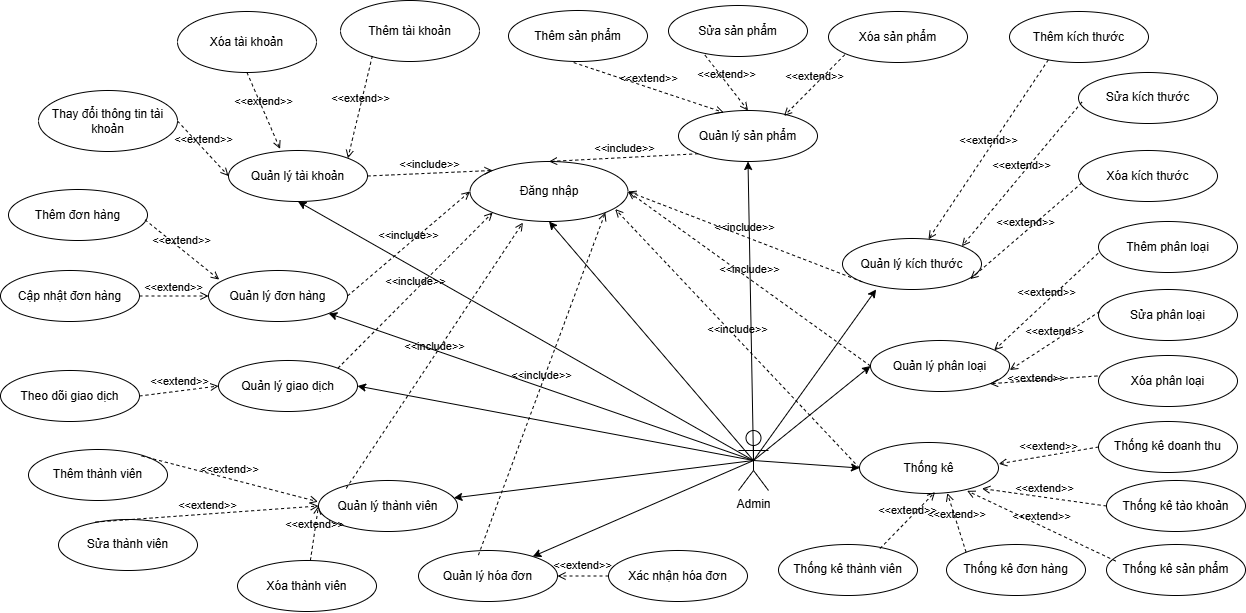
* Khách hàng



* Nhân viên



* Admin



## **Mô tả chi tiết các Usecase**

2.4.1 Usecase Đăng ký

- Mô tả vắn tắt: Use case cho phép khách đăng ký tài khoản trên hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào nút đăng ký trên màn hình. Hệ thống sẽ hiển thị màn hình đăng ký.
2. Khách hàng nhập thông tin trên form đăng ký gồm Số điện thoại, Password, Họ tên, Email, Ngày sinh, Địa chỉ rồi kích nút đăng ký. Hệ thống thêm một bản ghi mới vào trong bảng USER trong cơ sở dữ liệu.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
     1. Tại bước 2 trong luồng cơ bản, khi nhập thông tin không thoả mãn, hệ thống thông báo lỗi. Use case kết thúc.
     2. 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 có.
* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

2.4.2 Usecase Đăng nhập

- Mô tả vắn tắt: Use case này cho phép người dùng đăng nhập để xác định quyền truy cập vào hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:
    1. Use case này bắt đầu khi người dùng bấm vào nút “Đăng nhập” trên góc phải màn hình.

2) Hệ thống yêu cầu người dùng nhập các thông tin bao gồm: Tài khoản và Mật khẩu.

1. Người dùng nhập các thông tin: Tên đăng nhập và Mật khẩu và bấm nút “Đăng nhập”.
2. Hệ thống kiểm tra tên đăng nhập và mật khẩu đã nhập và chuyển hướng sang trang chủ.
3. Use case kết thúc.
   * Luồng rẽ nhánh:
     + 1. Sai tên đăng nhập hoặc mật khẩu: tại bước 3 trong luồng cơ bản nếu người dùng nhập tên đăng nhập hoặc mật khẩu sai, hệ thống hiển thị thông báo lỗi. Người dùng có thể nhập lại hoặc bỏ qua thao tác, khi đó use case kết thúc.
     1. Đăng ký: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Đăng ký” use case kết thúc.
     2. Quên mật khẩu: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Quên mật khẩu” use case kết thúc.
     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 có.
* Tiền điều kiện: Không có.
* Hậu điều kiện: Nếu use case thành công, người dùng sẽ đăng nhập được vào hệ thống. Nếu không trạng thái của hệ thống không thay đổi.
* Điểm mở rộng: Không có.

2.4.3 Usecase Tìm kiếm

Use case cho phép khách hàng tìm kiếm pizza.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi khách hàng kích vào ô tìm kiếm trên thanh công cụ, nhập tên sản phẩm và nhấn biểu tượng kính lúp ở bên cạnh. Hệ thống sẽ lấy các thông tin về sản phẩm bao gồm Name từ bảng PIZZA và Price từ bảng PIZZA, Image từ bảng PIZZAIMAGE và hiển thị lên màn hình.
2. Use case kết thúc.
   * Luồng rẽ nhánh:
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 một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện:

Không có

* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có

2.4.4 Usecase Xem chi tiết sản phẩm

* Mô tả vắn tắt: Use case cho phép khách hàng xem thông tin chi tiết về sản phẩm.
* Luồng sự kiện:
  + Luồng cơ bản:
    - 1. Use case này bắt đầu khi khách hàng kích nút “Sản phẩm” trong danh mục. Hệ thống sẽ lấy thông tin về danh sách các sản phẩm.
      2. Khách hàng kick vào ảnh của sản phẩm bất kỳ. Hệ thống sẽ hiển thị đầy đủ các thông tin (Name, Price, Description) từ bảng PRODUCTS, Image từ bảng PIZZAIMAGE, (Name, Price) từ bảng SIZE, BASETYPE của sản phẩm được chọn.
      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 một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có

* Tiền điều kiện: Không có
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có

2.4.5 Usecase Sửa thông tin cá nhân

- Mô tả vắn tắt: Use case cho phép khách hàng sửa thông tin cá nhân trong tài khoản.

* Luồng sự kiện:
  + Luồng cơ bản:

1) Use case này bắt đầu khi khách hàng kích vào nút “Tài khoản” trên màn hình. Hệ thống sẽ hiển thị màn hình thông tin cá nhân.

* + 1. Khách hàng nhập thông tin cần chỉnh sửa rồi kích nút "Cập nhật". Hệ thống cập nhật lại thông tin trên bảng User trong cơ sở dữ liệu.
    2. Use case kết thúc.
  + Luồng rẽ nhánh:

1) Tại bước 2 ở luồng cơ bản, nếu khách hàng nhập sai định dạng, hệ thống thông báo lỗi. Use case kết thúc.

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 một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Không có
* Tiền điều kiện: Khách hàng đăng nhập thành công
* Hậu điều kiện: Không có
* Điểm mở rộng: Không có.

2.4.6 Usecase Quản lý sản phẩm

- Mô tả vắn tắt: Use case này cho phép quản trị viên xem, thêm, sửa thông tin, xóa các sản phẩm trong bảng PIZZA.

* Luồng sự kiện:
  + Luồng cơ bản:
    - 1. Use case này bắt đầu khi quản trị viên kích vào nút “Quản lý sản phẩm” trên thanh Menu quản trị. Hệ thống lấy thông tin chi tiết của các sản phẩm từ bảng PIZZA, PIZZAIMAGE trong cơ sở dữ liệu và hiển thị lên màn hình.
      2. Thêm mới sản phẩm:
  1. Quản trị viên kích vào nút “Thêm sản phẩm”. Hệ thống hiển thị “Pizza Form” yêu cầu quản trị viên nhập các thông tin sản phẩm.
  2. Quản trị viên nhập các thông tin sản phẩm và kích vào nút “Thêm sản phẩm”. Hệ thống tạo ra id lưu các thông tin vào bảng PIZZA, PIZZAIMAGE và hiển thị danh sách các sản phẩm lên màn hình.
     + 1. Sửa thông tin sản phẩm:
          1. Quản trị viên kích vào nút “Cập nhật” trên dòng sản phẩm. Hệ thống hiển thị một màn hình với các thông tin cũ của sản phẩm.
          2. Quản trị viên sửa thông tin của sản phẩm và kích vào nút “Cập nhật”. Hệ thống sẽ cập nhật thông tin sản phẩm trong bảng PIZZA, PIZZAIMAGE và hiển thị danh sách sản phẩm đã cập nhật.
     1. Xóa sản phẩm:
        + 1. Quản trị viên kích vào nút “Xóa” trên dòng sản phẩm. Hệ thống sẽ xóa thông tin sản phẩm trong bảng PIZZA, PIZZAIMAGE và hiển thị danh sách sản phẩm còn lại.
     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 website sẽ hiển thị một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Quản trị viên phải đăng nhập vào hệ thống trước thực hiện quản lý sản phẩm.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

2.4.7 Usecase Đặt hàng

- Mô tả vắn tắt: Use case này cho phép khách hàng đặt mua sản phẩm .

* Luồng sự kiện:
  + Luồng cơ bản:

Use case này bắt đầu khi khách hàng kick vào nút “Giỏ hàng”. Hệ thống lấy thông tin của sản phẩm (Name, Price) từ bảng PRODUCT. Image\_url từ bảng IMAGES, Price từ bảng , SIZE, BASETYPE bảng ORDERS, ORDER\_DETAIL bảng trong cơ sở dữ liệu và hiển thị lên màn hình.

Khách hàng chọn sản phẩm muốn mua và kick vào nút “Mua hàng” trong giỏ hàng sản phẩm. Hệ thống hiển thị form nhập thông tin nhận hàng.

Khách hàng nhập đầy đủ các trường thông tin bắt buộc (Địa chỉ, Họ Tên, SDT, Email, chọn phương thức thanh toán)., ghi chú nếu ai có nhu cầu thì nhập vào, sau đó khách hàng nhấn vào nút “Mua hàng”. Hệ thống hiển thị thông báo mua hàng thành công

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 website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Người dùng phải đăng nhập vào hệ thống trước khi thực hiện đặt hàng.

* Hậu điều kiện:

Không có.

* Điểm mở rộng: Không có.

2.4.8 Usecase Quản lý giỏ hàng

- Mô tả vắn tắt: Use case này cho phép khách hàng quản lý giỏ hàng .

* Luồng sự kiện:
  + Luồng cơ bản:
    - 1. Xem giỏ hàng:

Use case này bắt đầu khi khách hàng kick vào nút “Giỏ hàng” trong trang chủ. Hệ thống sẽ hiển thị danh sach các sản phẩm có trong giỏ hàng từ bảng PIZZA, PIZZAIMAGE, SIZE, BASETYPE.

* + - 1. Sửa giỏ hàng:

Khách hàng chỉnh sửa số lượng sản tùy ý. Hệ thống sẽ cập nhật lại thông tin sản phẩm, size, loại trong localstorage và hiển thị lên màn hình sau khi sửa.

* + - 1. Xóa giỏ hàng:

Khách hàng nhấn vào biểu tượng thùng rác. Hệ thống sẽ xóa thông tin sản phẩm vừa xóa ra khỏi localstorage và hiển thỉa màn hình các sản phẩm còn lại 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 website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có

* Hậu điều kiện:

Không có.

* Điểm mở rộng: Không có.

2.4.9 Usecase Quản lý đơn hàng

- Mô tả vắn tắt: Use case này cho phép quản lý quản lý đơn hàng .

* Luồng sự kiện:
  + Luồng cơ bản:
    - 1. Xem danh sách đơn hàng:

Use case này bắt đầu khi khách hàng kick vào nút “Quản lý đơn hàng” trong trang chủ quản lý. Hệ thống sẽ hiển thị danh sach các sản phẩm có trong giỏ hàng từ bảng ORDERS, ORDERDETAIL.

* + 1. Thêm đơn hàng:

Use case này bắt đầu khi khách hàng kick vào nút “Thêm đơn hàng ” trong màn hình “Quản lý đơn hàng”. Hệ thống sẽ hiển thị form điền thông tin đơn hàng và các mặt hàng vào form, nhấn “Thêm”, dữ liệu được lưu vào bảng ORDERS, ORDERDETAIL.

* + 1. Sửa đơn hàng:

Nhân viên chỉnh sửa thông tin đơn hàng bao gồm: số điện thoại, địa chỉ người nhận hàng, thời gian muốn nhận hàng, trạng thái đơn hàng, xác nhận hoàn thành đơn hàng và lưu vào bảng ORDERS.

* + 1. Xóa đơn hàng:

Nhân viên nhấn vào biểu tượng thùng rác. Hệ thống sẽ xóa thông tin sản phẩm vừa xóa ra khỏi bảng ORDERDETAIL và ORDERS và hiển thị màn hình các đơn hàng còn lại trong danh sách đơn 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 website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Nhân viên phải đăng nhập vào hệ thống trước khi thực hiện quản lý đơn hàng.

* Hậu điều kiện:

Không có.

* Điểm mở rộng: Không có.

2.4.10 Usecase Quản lý thành viên

- Mô tả vắn tắt: Use case này cho phép admin quản lý đơn hàng .

* Luồng sự kiện:
  + Luồng cơ bản:
    - 1. Xem danh sách thành viên:

Use case này bắt đầu khi khách hàng kick vào nút “Quản lý thành viên” trong trang chủ quản lý. Hệ thống sẽ hiển thị danh sách thành viên hiện có từ bảng USER, MEMBERSHIP.

* + - 1. Thêm thành viên:

Use case này bắt đầu khi khách hàng kick vào nút “Thêm thành viên” trong màn hình “Quản lý thành viên”. Hệ thống sẽ hiển thị form điền thông tin thành viên vào form, nhấn “Thêm”, dữ liệu được lưu vào bảng MEMBERSHIP.

* + 1. Sửa thành viên:

Admin chỉnh sửa thông tin thành viên bao gồm: tên, số điện thoại, địa chỉ, ngày sinh... và lưu vào bảng MEMBERSHIP.

* + 1. Xóa thành viên:

Admin nhấn vào biểu tượng thùng rác. Hệ thống sẽ xóa thông tin sản phẩm vừa xóa ra khỏi bảng MEMBERSHIP và hiển thị màn hình các thành viên còn lại trong giỏ hàng danh sách thành viên.

* + 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 website sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Admin phải đăng nhập vào hệ thống trước khi thực hiện quản lý thành viên.

* Hậu điều kiện:

Không có.

* Điểm mở rộng: Không có.

## **Xây dựng biểu đồ trình tự**

2.5.1 Usecase Đăng ký



2.5.2 Usecase Đăng nhập



2.5.3 Usecase Tìm kiếm



2.5.4 Usecase Xem chi tiết sản phẩm



2.5.5 Usecase Sửa thông tin cá nhân



2.5.6 Usecase Quản lý sản phẩm





2.5.7 Usecase Đặt hàng



2.5.8 Usecase Quản lý giỏ hàng



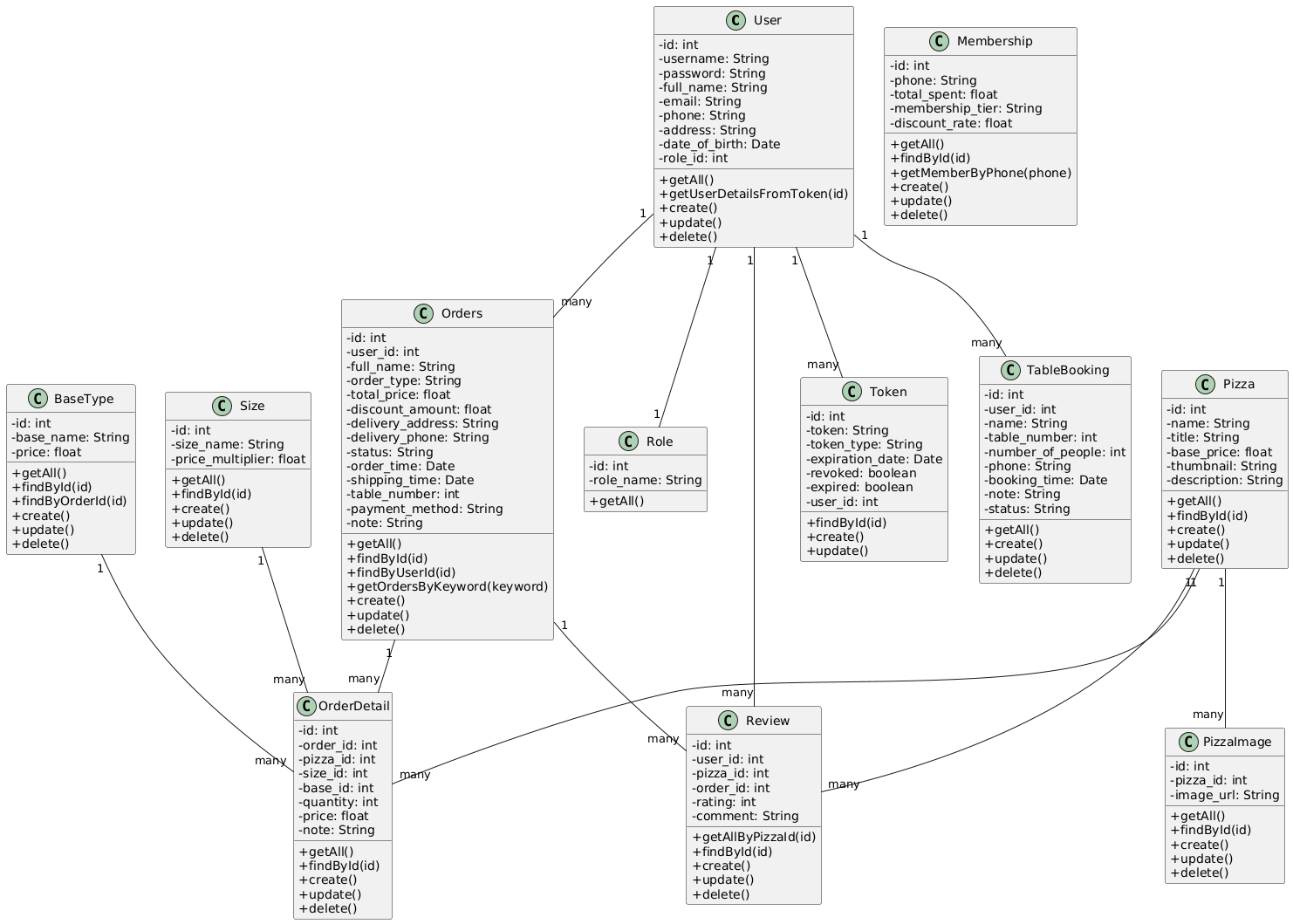
2.5.9 Usecase Quản lý đơn hàng



2.5.10 Usecase Quản lý thành viên

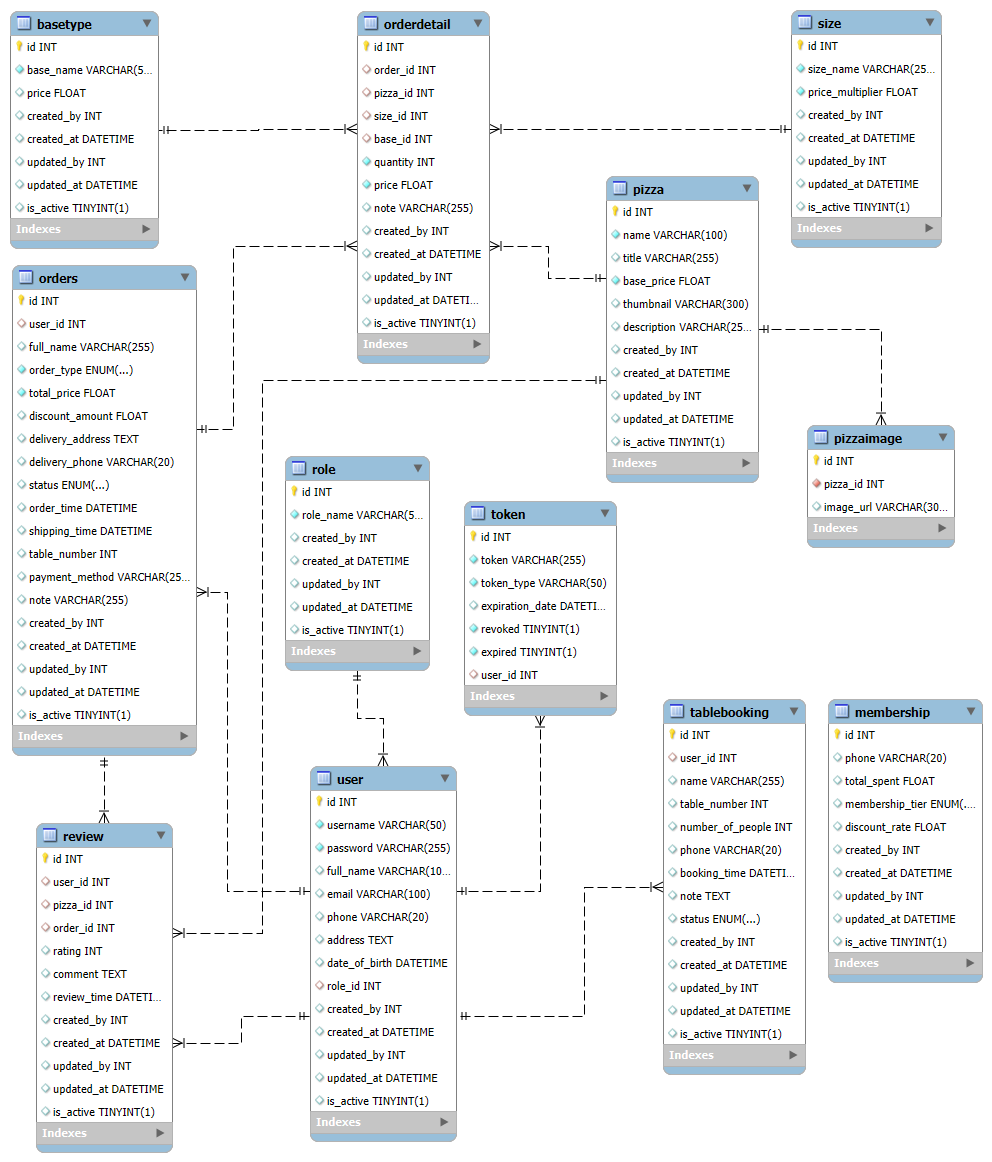


## **Thiết kế biểu đồ lớp chi tiết**

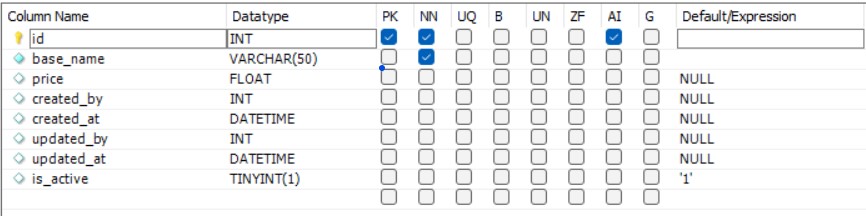


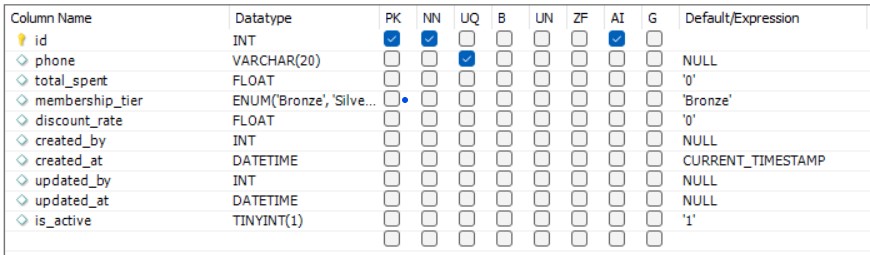
## **Cơ sở dữ liệu**

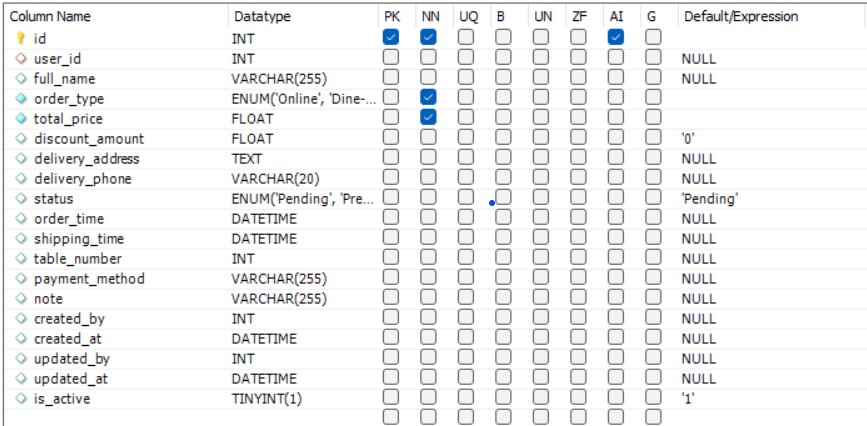
### Sơ đồ dữ liệu quan hệ



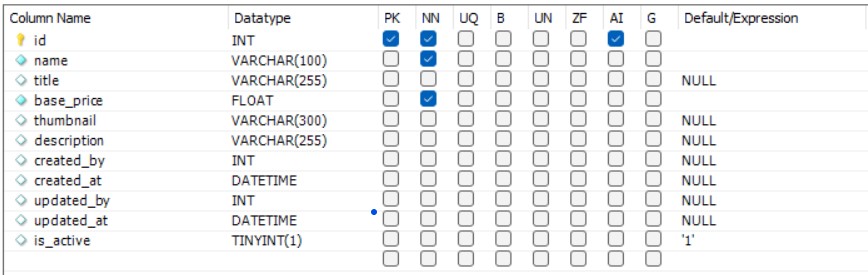
### Chi tiết các bảng dữ liệu

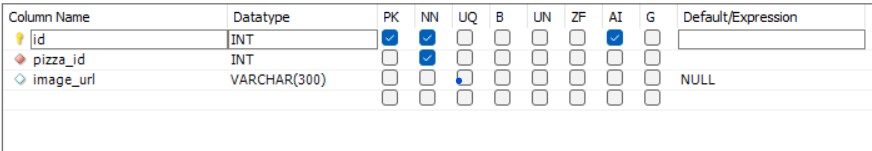


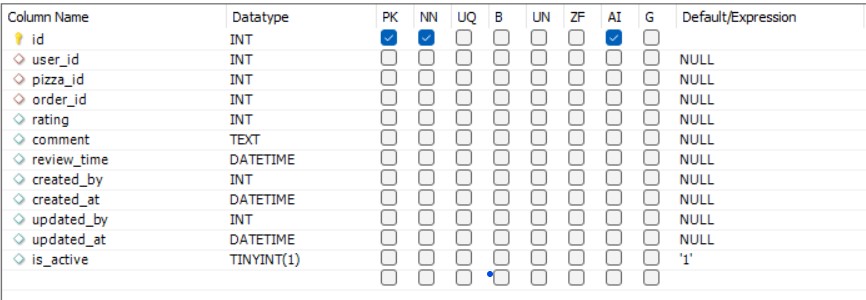


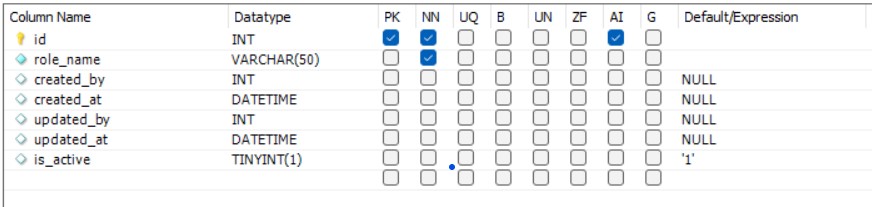


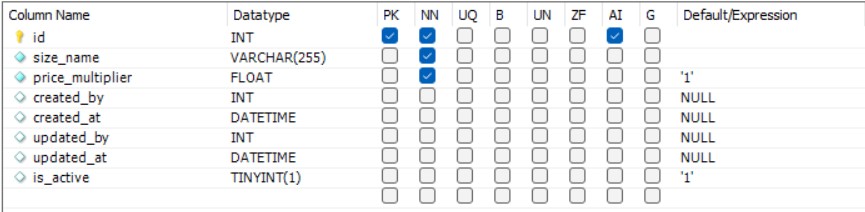


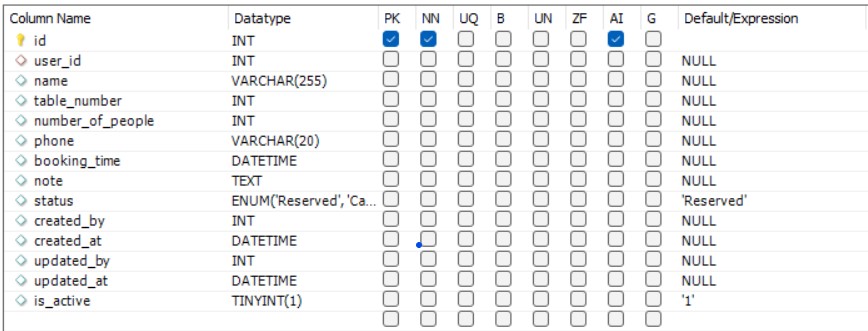


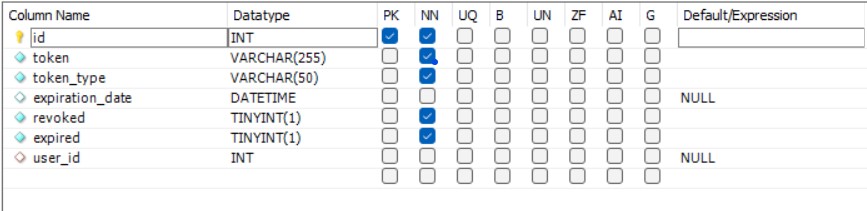


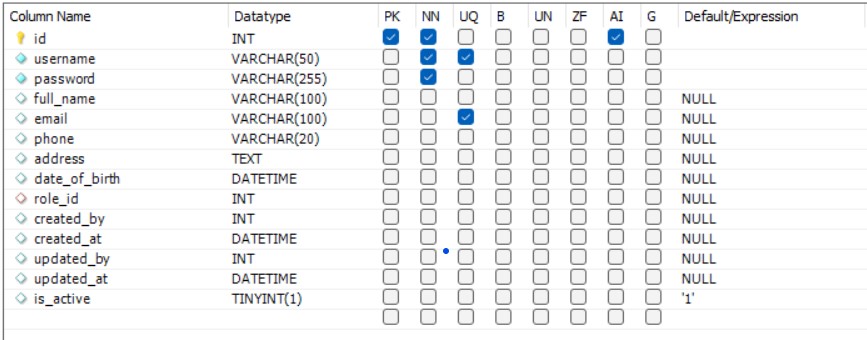




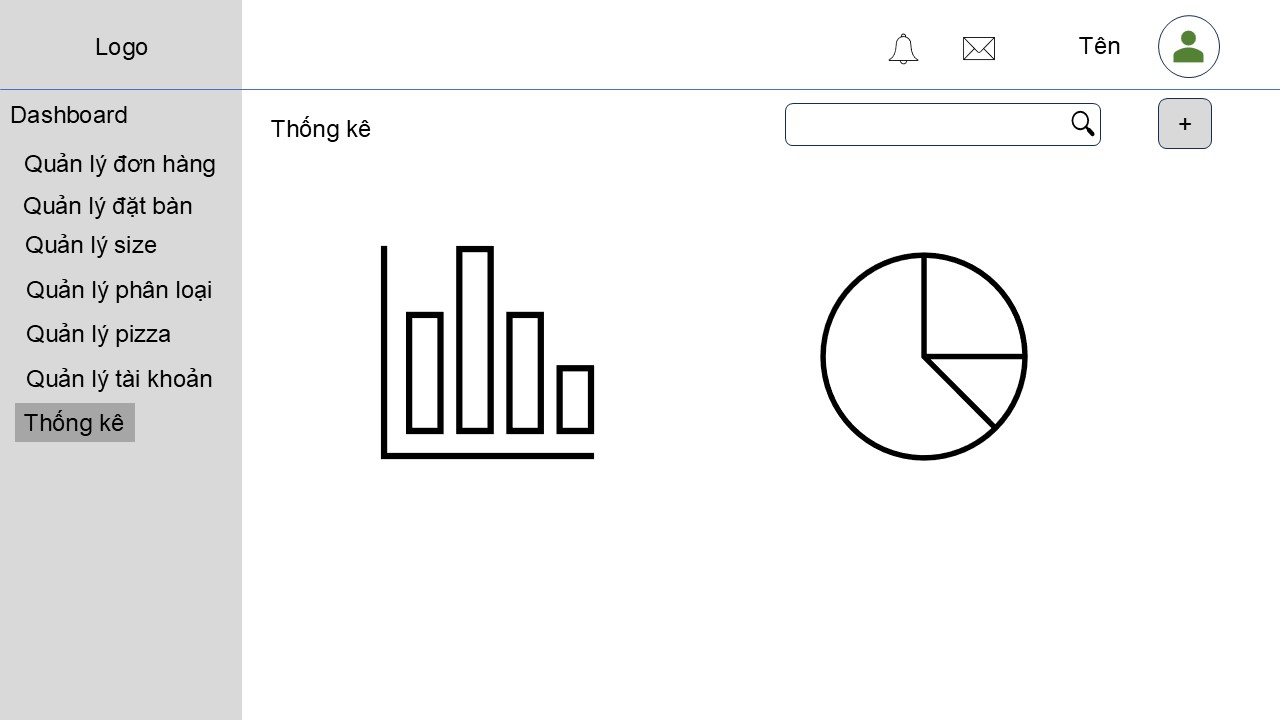
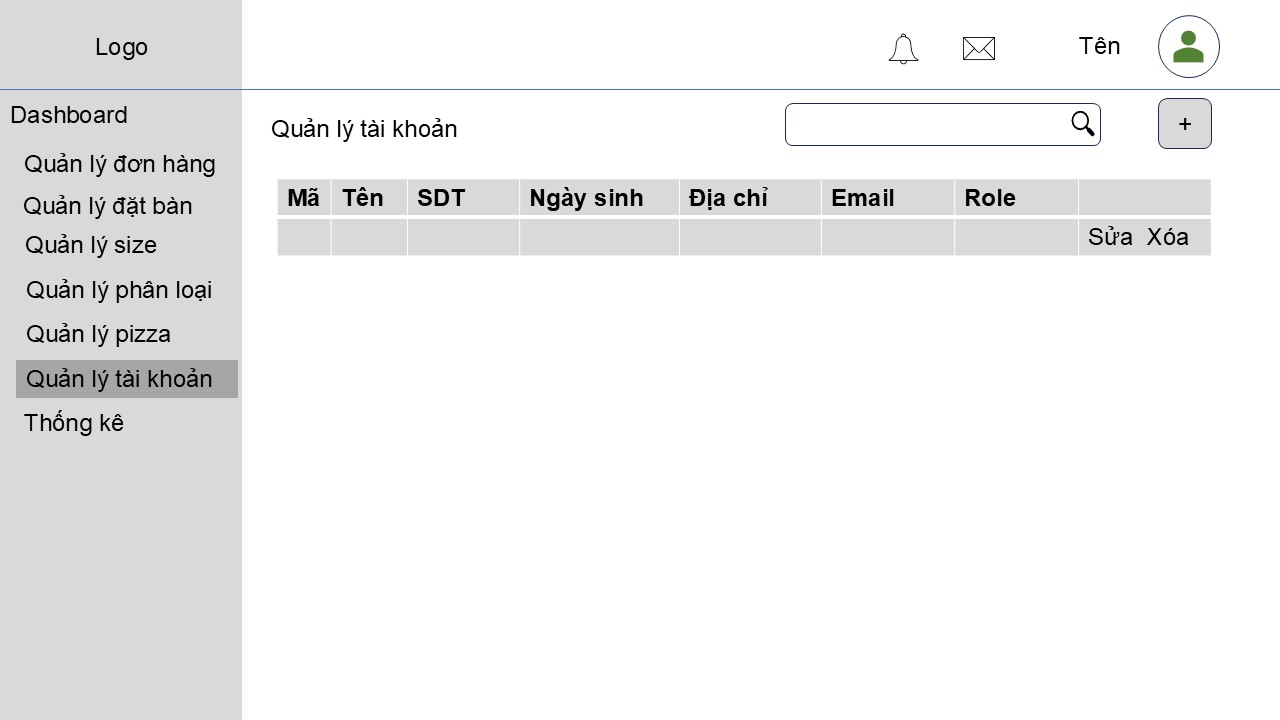
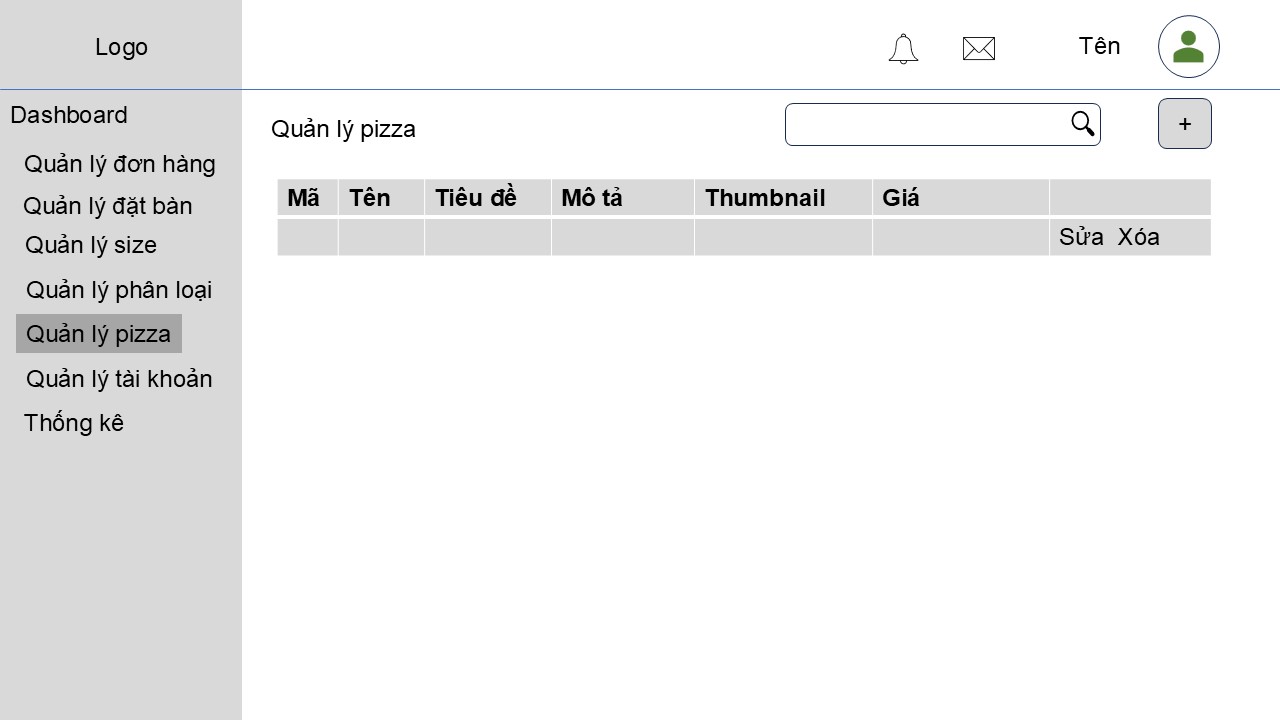
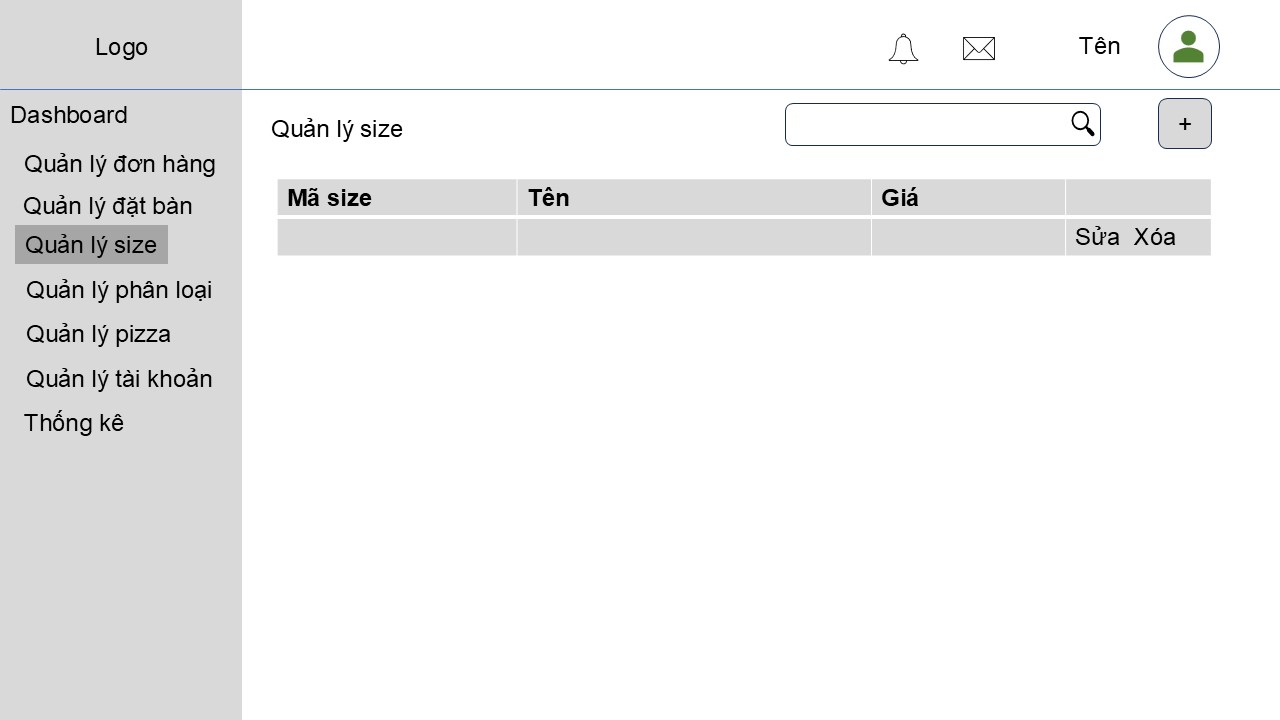
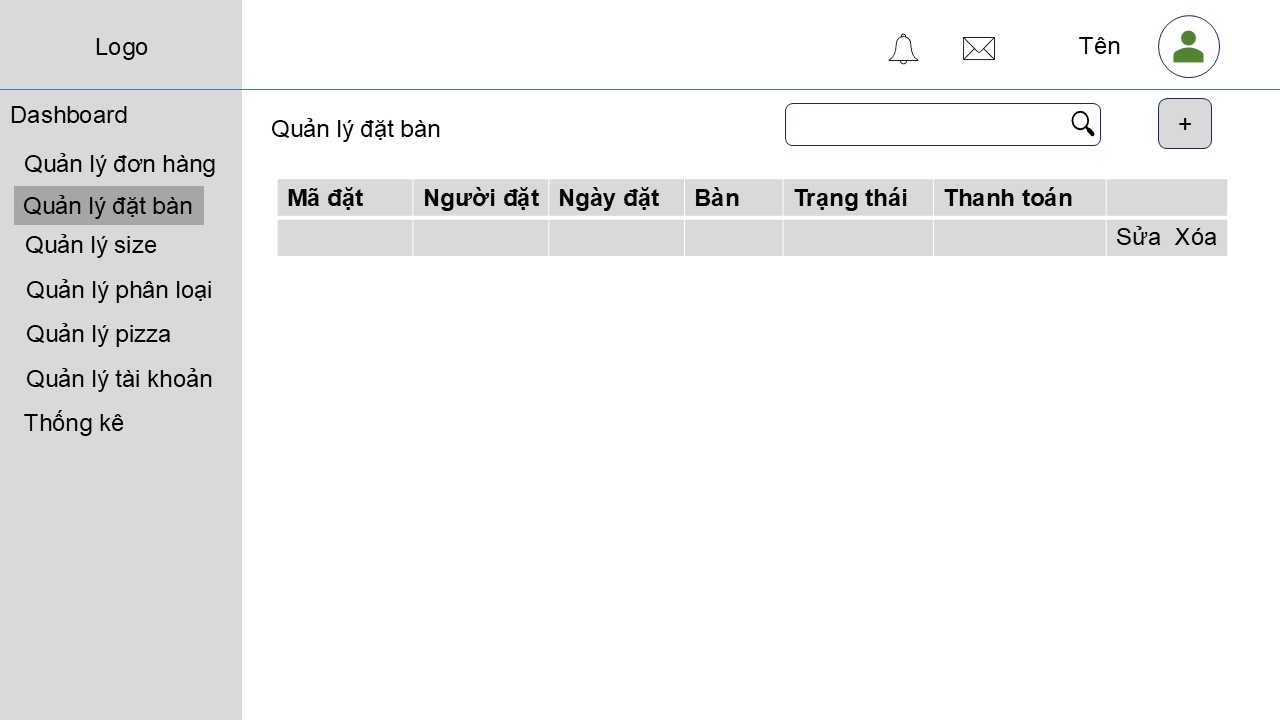
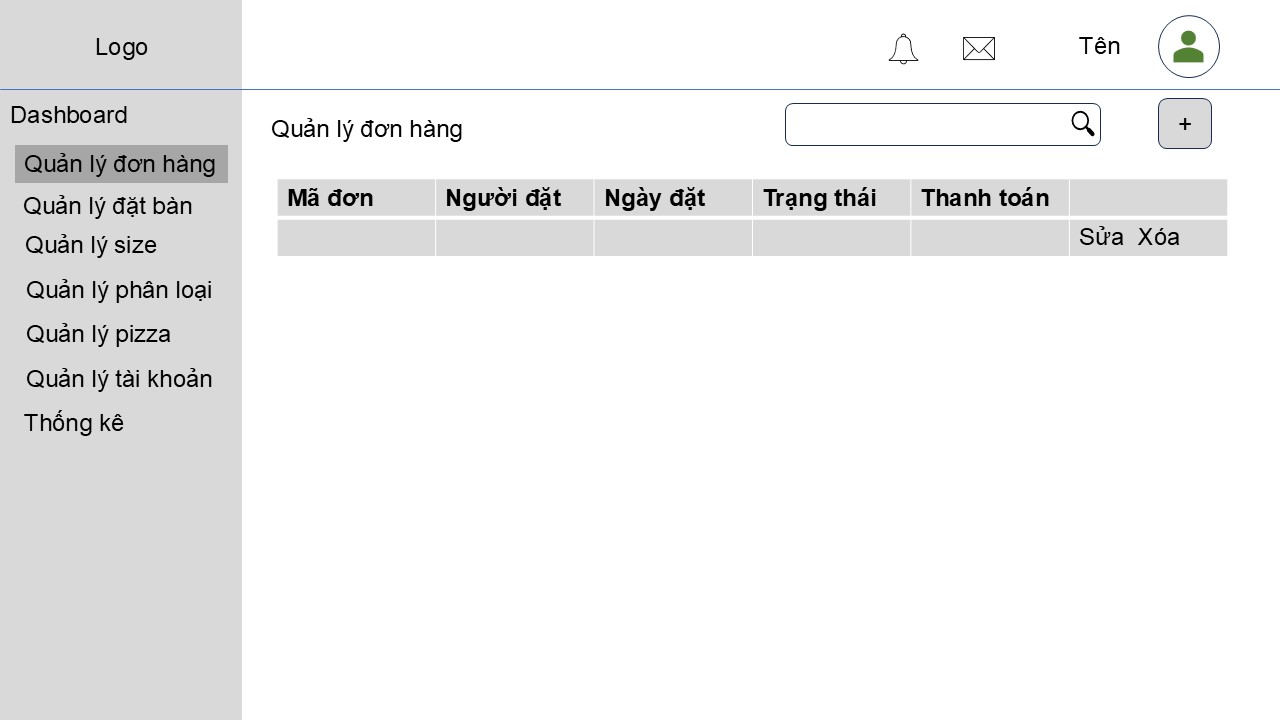
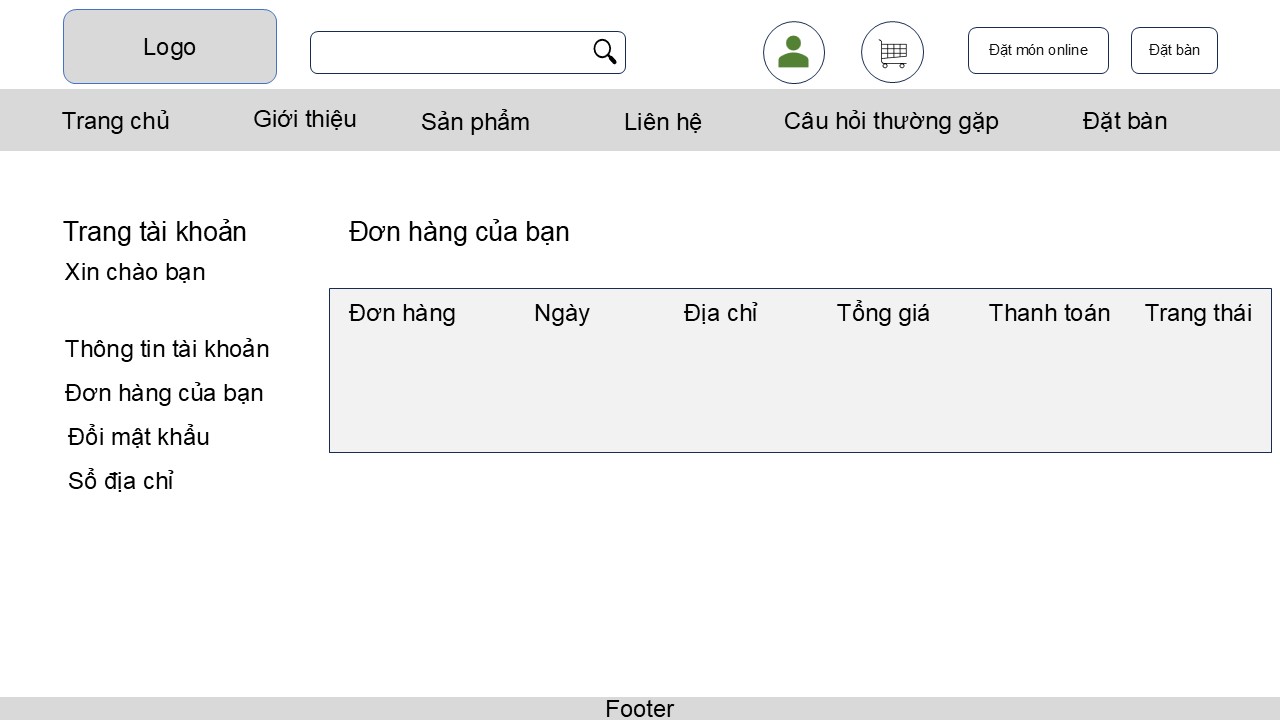
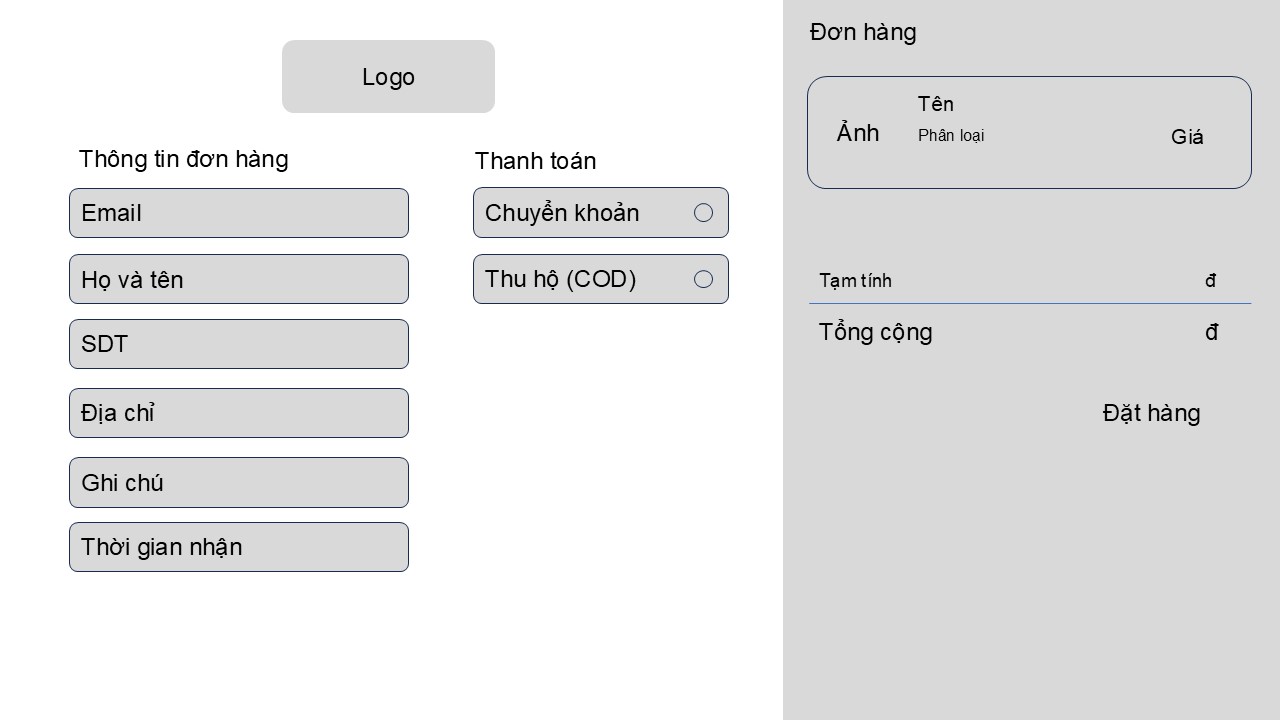
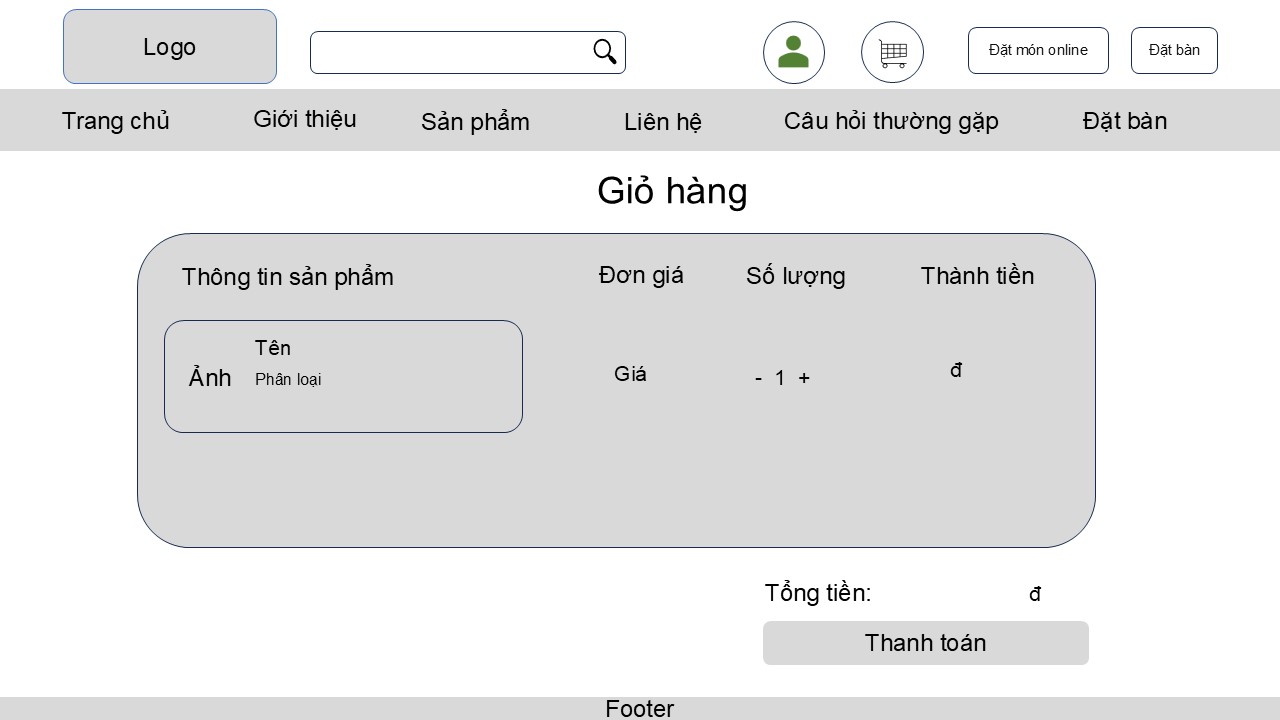
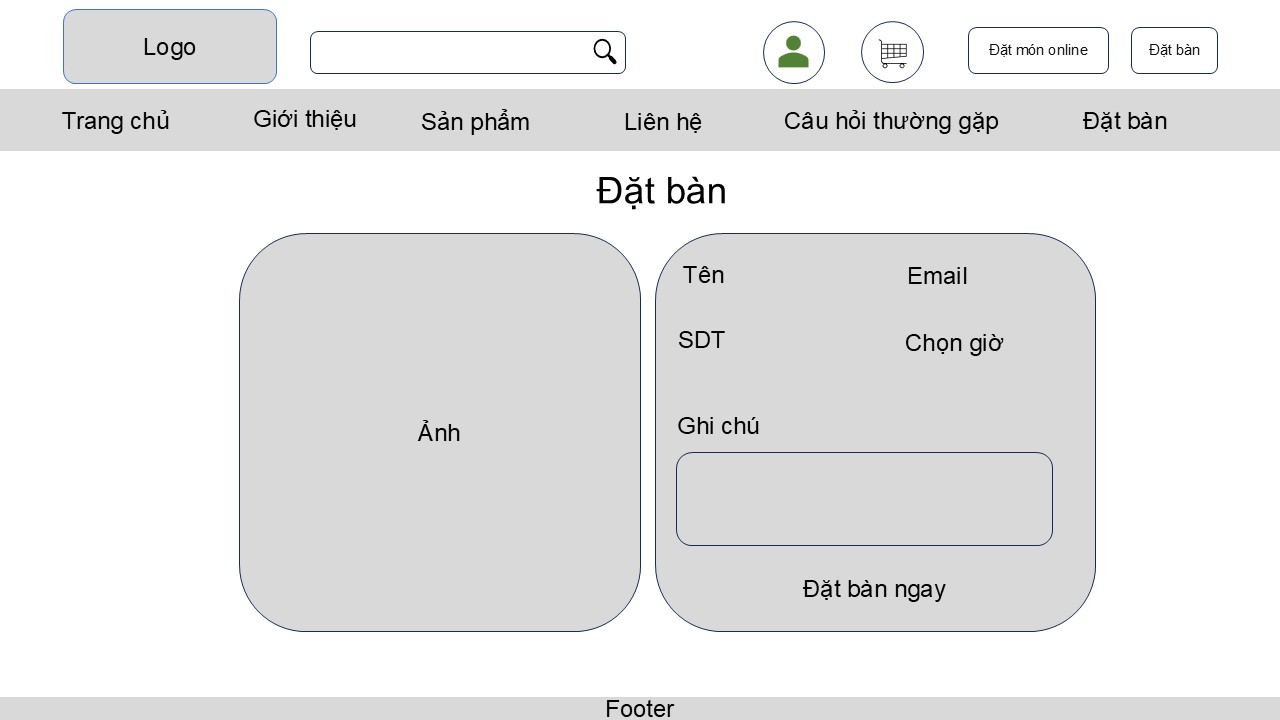
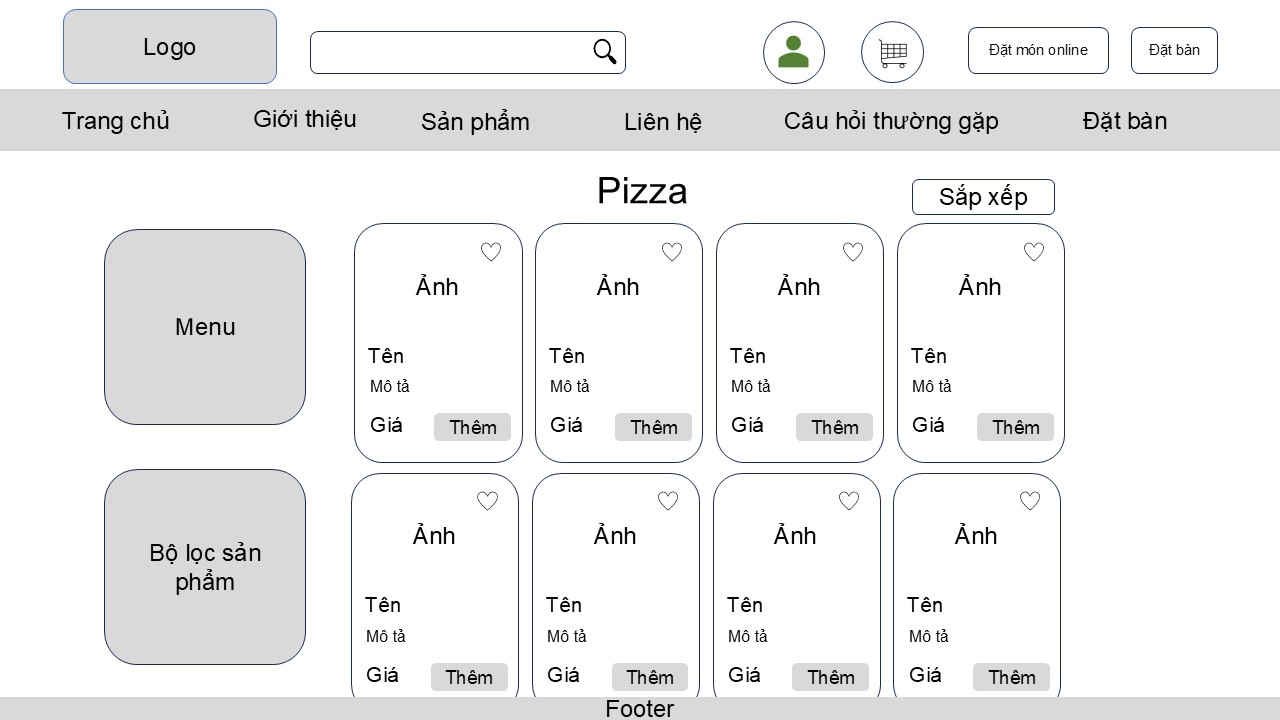
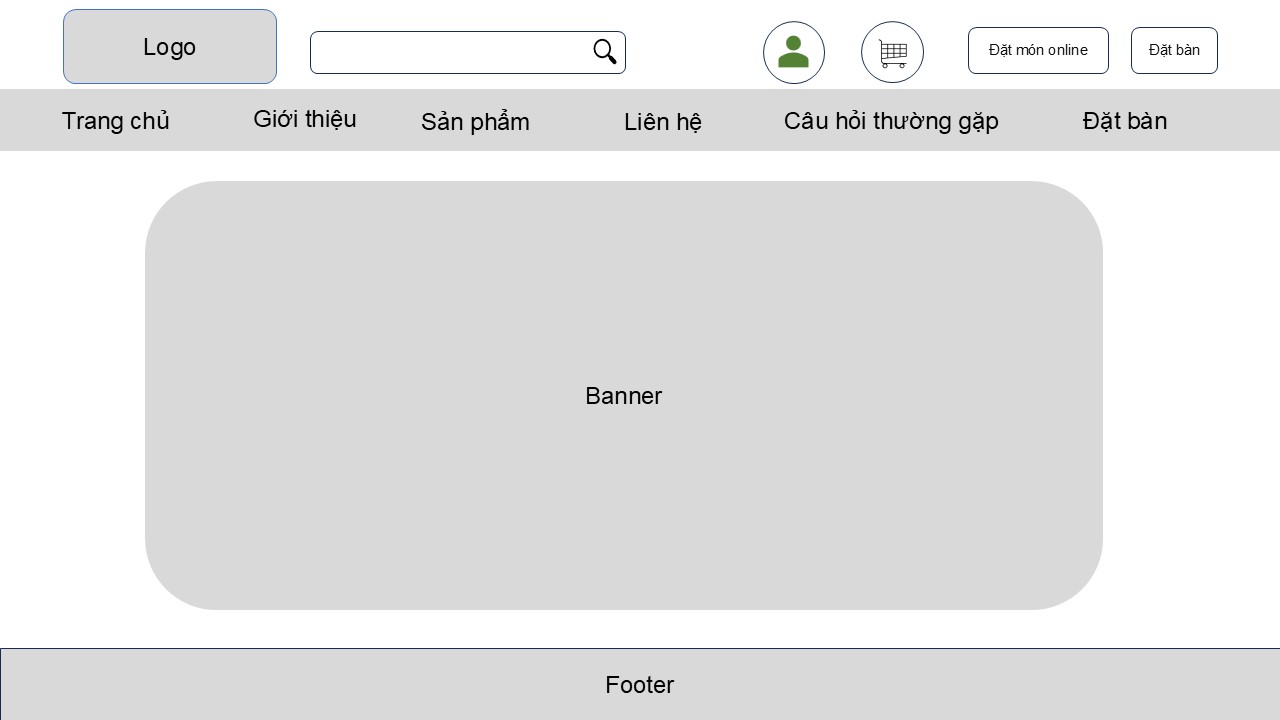








## **Thiết kế giao diện**



# 

# CHƯƠNG 3: CÀI ĐẶT VÀ TRIỂN KHAI

## **3.1 Giới thiệu về công cụ và môi trường cài đặt**

### 3.1.1 Công cụ lập trình Intelij, Visual Studio Code

### 3.1.2 Môi trường Java, TypeScript

### 3.1.3 Thư viện Java Spring Boot, Angular

## **3.2 Trang màn hình giao diện của khách hàng (Front-end)**

## **3.3 Trang màn hình giao diện của người quản trị (Back-end):**

## **3.4 Một số giao diện khác**

## **3.5 Kiểm thử**

### 3.5.1 Lập kế hoạch kiểm thử

### 3.5.2 Thiết kế testcase

### 3.5.3 Thực thi testcase

### 3.5.4 Báo cáo kiểm thử

# KẾT LUẬN

**TÀI LIỆU THAM KHẢO**