# TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

# KHOA CÔNG NGHỆ THÔNG TIN



# A blue and yellow logo Description automatically generated

**ĐỒ ÁN TỐT NGHIỆP**

ĐỀ TÀI

**XÂY DỰNG ỨNG DỤNG WEBSITE BÁN ĐỒ**

**THỂ THAO CHO CỬA HÀNG BLUE SKY**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | : TS.Nguyễn Hiếu Cường |
| Sinh viên thực hiện | : Nguyễn Văn Vũ |
| Mã sinh viên | : 201200421 |
| Lớp | : CNTT5\_K61 |
| Khóa | : 61 |

**Hà Nội – 2025**

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

# KHOA CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN TỐT NGHIỆP**

ĐỀ TÀI

**XÂY DỰNG ỨNG DỤNG WEBSITE BÁN ĐỒ**

**THỂ THAO CHO CỬA HÀNG BLUE SKY**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | : TS.Nguyễn Hiếu Cường |
| Sinh viên thực hiện | : Nguyễn Văn Vũ |
| Mã sinh viên | : 201200421 |
| Lớp | : CNTT5\_K61 |
| Khóa | : 61 |

**Hà Nội – 2025**

**LỜI CẢM ƠN**

Lời đầu tiên em xin chân thành cảm ơn quý thầy cô trong khoa công nghệ thông tin đã hỗ trợ chúng em trong suốt quá trình học tập, rèn luyện tại trường và khi chúng em thực hiện đề tài này.

Em xin gửi lời cảm ơn đến thầy TS**.** Nguyễn Hiếu Cường, người đã tận tình hướng dẫn, giúp đỡ, chỉ bảo em trong suốt thời gian thực hiện đồ án tốt nghiệp. Đồng thời em xin trân trọng cảm ơn những tình cảm quý báu mà các thầy cô trong Trường Đại học Giao Thông Vận Tải truyền đạt cho em những những kinh nghiệm, kỹ thuật và cách thức trong việc xây dựng đề tài này.

Tuy nhiên, do thời gian hạn chế, em chưa thể phát huy hết các ý tưởng và khả năng của mình trong việc áp dụng ngôn ngữ lập trình và kỹ thuật lập trình vào đề tài này. Trong thời gian thực hiện đề tài không tránh khỏi những thiếu sót. Em rất mong nhận được ý kiến đóng góp từ quý thầy cô và các bạn để hoàn thiện hơn.

Em xin chân thành cảm ơn!

**MỤC LỤC**

### LỜI CẢM ƠN 1

### MỤC LỤC 2

### DANH MỤC KÝ HIỆU VÀ CÁC TỪ VIẾT TẮT 4

### DANH MỤC CÁC HÌNH VẼ, CÁC HÌNH ẢNH 4

### DANH MỤC BẢNG BIỂU 7

### LỜI MỞ ĐẦU 8

### Lý do chọn đề tài 8

### Mục đích 8

### Đối tượng và phạm vi nghiên cứu 9

### Ý nghĩa khoa học và thực tiễn của đề tài 10

### Cấu trúc đồ án 11

### CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 11

### Giới thiệu chung 11

### Nghiên cứu hiện trang và giải quyết vấn đề 11

* + 1. **Hiện trạng của công ty** 11
    2. **Cơ cấu tổ chức** 12
    3. **Mô tả hoạt động của các bộ phận** 12
    4. **Xác định yêu cầu** 13
    5. **Yêu cầu về phi chức năng** 13

### Công nghệ sử dụng trong dự án 14

* + 1. **Giới thiệu một số công nghệ phổ biến** 14
    2. **Giới thiệu một số kiến trúc** 14
    3. **Công nghệ sử dụng** 18

### CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 26

### 2.1 Các yêu cầu phi chức năng 26

### 2.2 Các yêu cầu chức năng 27

### 2.3 Biểu đồ Use Case tổng quan 28

### 2.4 Thiết kế cơ sở dữ liệu 64

### CHƯƠNG 3: THỬ NGHIỆM VÀ TRIỂN KHAI HỆ THỐNG 71

**3.1 Công cụ và môi trường được sử dụng** 71

**3.1.1 Các phần mềm được sử dụng** 71

**3.1.2 Cài đặt phần mềm và môi trường** 71

**3.2 Kiểm thử chức năng hệ thống** 72

**3.2.1 Môi trường và phương pháp kiểm thử** 72

**3.2.2 Tiến hành kiểm thử** 72

**3.2.3 Hình ảnh** 76

### KẾT LUẬN 89

### TÀI LIỆU THAM KHẢO 90

**DANH MỤC KÝ HIỆU VÀ CÁC TỪ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **NGHĨA TIẾNG ANH** | **NGHĨA TIẾNG VIỆT** |
| 1 | UC | Use case | Tình huống sử dụng |
| 2 | DB | Database | Cơ sở dữ liệu |
| 3 | API | Application Programming Interface | Giao diện lập trình ứng dụng |
| 4 | HTTP | Hypertext Transfer Protocol | Giao thức Truyền tải Siêu văn bản |
| 5 | VOPC | View of a participating classes diagram | Biểu đồ lớp phân tích |
| 6 | TC | Test case | Trường hợp kiểm thử |
| 7 | TMĐT | E commerce | Thương mại điện tử |
| 8 | ADMIN | Administrator | Quản trị |
| 9 | HTML | HyperText Markup Language | Ngôn ngữ đánh dấu siêu văn bản |

**DANH MỤC CÁC HÌNH VẼ, CÁC HÌNH ẢNH**

Hình 1.1. Kiến trúc mô hình 3 lớp 15

Hình 1.2 Mô hình kiến trúc Microservices 17

Hình 1.3 Kiến trúc của Angular 19

Hình 1.4 Kiến trúc phân tầng Spring Boot 23

Hình 1.5 Kiến trúc luồng Spring Boot 23

Hình 1.6 Hệ quản trị cơ sở dữ liệu MySQL 24

Hình 2.1: Use case tổng quan toàn bộ hệ thống 29

Hình 2.2: Biểu đồ use case tổng quan khách hang 29

Hình 2.3 Biểu đồ use case tổng quan admin 30

Hình 2.4: Biểu đồ use case chính 30

Hình 2.5: Biểu đồ use case thứ cấp 31

Hình 2.6: Biểu đồ VOPC đăng nhập 47

Hình 2.7: Biểu đồ trình tự use case đăng nhập 47

Hình 2.8: Biểu đồ VOPC use case đăng ký 48

Hình 2.9: Biểu đồ trình tự use case đăng ký 48

Hình 2.10: Biểu đồ VOPC use case xem sản phẩm theo danh mục 49

Hình 2.11: Biểu đồ trình tự xem sản phẩm theo danh mục 50

Hình 2.12: Biểu đồ VOPC xem chi tiết một sản phẩm 50

Hình 2.13: Biểu đồ trình tự use case xem chi tiết một sản phẩm 51

Hình 2.14: Biểu đồ VOPC use case tìm kiếm sản phẩm 51

Hình 2.15: Biểu đồ trình tự use case tìm kiếm sản phẩm 52

Hình 2.16: Biểu đồ VOPC use case quản lý giỏ hàng 52

Hình 2.17: Biểu đồ trình tự use case quản lý giỏ hàng 53

Hình 2.18: Biểu đồ use case theo dõi đơn hàng 54

Hình 2.19: Biểu đồ trình tự use case theo dõi đơn hàng 54

Hình 2.20: Biểu đồ VOPC use case thanh toán 55

Hình 2.21: Biểu đồ trình tự use case thanh toán 55

Hình 2.22: Biểu đồ VOPC use case liên hệ 56

Hình 2.23: Biểu đồ use case liên hệ 56

Hình 2.24: Biểu đồ VOPC use case quản lý đơn hàng 57

Hình 2.25: Biểu đồ trình tự use case quản lý đơn hàng 58

Hình 2.26: Biểu đồ VOPC quản lý sản phẩm 59

Hình 2.27: Biểu đồ trình tự use case quản lý sản phẩm 60

Hình 2.28: Biểu đồ VOPC use case quản lý danh mục 61

Hình 2.29: Use case quản lý danh mục 61

Hình 2.30: Biểu đồ VOPC quản lý khách hàng 62

Hình 2.31: Biểu đồ trình tự use case quản lý khách hàng 63

Hình 2.32: Biểu đồ VOPC quản lý liên hệ 64

Hình 2.33: Biểu đồ trình tự use case quản lý liên hệ 64

Hình 2.34: Sơ đồ thực thể liên kết 65

Hình 3.1: Kiểm thử API đăng nhập 77

Hình 3.2: Kiểm thử API đăng ký 77

Hình 3.3: Kiểm thử API xem sản phẩm theo danh mục 78

Hình 3.4: Kiểm thử API xem chi tiết một sản phẩm 78

Hình 3.6: Kiểm thử API theo dõi đơn hàng 79

Hình 3.8: Kiểm thử API liên hệ 79

Hình 3.10: Kiểm thử API xem tất cả khách hàng (Admin) 80

Hình 3.12: Kiểm thử API xem tất cả sản phẩm (Admin) 80

Hình 3.13: Giao diện trang chủ 81

Hình 3.14: Giao diện đăng nhập khách hàng 82

Hình 3.15: Giao diện đăng ký tài khoản 82

Hình 3.16: Giao diện theo dõi đơn hàng 83

Hình 3.17: Giao diện chỉnh sửa thông tin cá nhân 83

Hình 3.18: Giao diện liên hệ với Admin 83

Hình 3.19: Giao diện xem sản phẩm theo danh mục 84

Hình 3.20: Giao diện giỏ hàng 84

Hình 3.21: Giao diện danh sách sản phẩm yêu thích 85

Hình 3.23: Giao diện quản lý khách hàng 85

Hình 3.24: Giao diện quản lý đơn hàng 85

Hình 3.25: Giao diện xem đánh giá 86

Hình 3.26: Giao diện quản lý danh mục 86

Hình 3.27: Giao diện quản lý liên hệ 87

Hình 3.28: Giao diện thống kê sản phẩm 87

Hình 3.29: Giao diện quản lý sản phẩm 88

Hình 3.30: Giao diện thêm sản phẩm 88

**DANH MỤC BẢNG BIỂU**

Bảng 2.1: Các tác nhân hệ thống 27

Bảng 2.2: Cấu trúc bảng notification 65

Bảng 2.3: Cấu trúc bảng roles 66

Bảng 2.4: Cấu trúc bảng user\_roles 66

Bảng 2.5: Cấu trúc bảng users 66

Bảng 2.6: Cấu trúc bảng categories 67

Bảng 2.7: Cấu trúc bảng products 67

Bảng 2.8: Cấu trúc bảng rates 68

Bảng 2.9: Cấu trúc bảng favotites 68

Bảng 2.10: Cấu trúc bảng carts 68

Bảng 2.11: Cấu trúc bảng order\_details 69

Bảng 2.12: Cấu trúc bảng orders 69

Bảng 2.13: Cấu trúc bảng order\_details 70

Bảng 2.14: Cấu trúc bảng contact 70

Bảng 2.15: Cấu trúc bảng images 70

Bảng 3.1: Kiểm thử API đăng nhập 72

Bảng 3.2: Kiểm thử API đăng ký 73

Bảng 3.3: Kiểm thử API quên mật khẩu 74

Bảng 3.4: Kiểm thử API quên mật khẩu 74

Bảng 3.5: Kiểm thử API tìm kiếm sản phẩm 75

Bảng 3.6: Kiểm thử API liên hệ 75

Bảng 3.7: Kiểm thử API thanh toán 76

# LỜI MỞ ĐẦU

## Lý do chọn đề tài

Trong thời đại công nghệ số ngày nay, việc phát triển các ứng dụng web chất lượng cao không chỉ đòi hỏi sự hiểu biết sâu rộng về lập trình mà còn yêu cầu kiến thức chuyên sâu về các công nghệ xây dựng web. Chính vì vậy, đề tài xây dựng một ứng dụng web kinh doanh bán đồ thể thao sẽ giúp sinh viên có cơ hội áp dụng và phát triển những kỹ năng quan trọng để chuẩn bị cho sự nghiệp sau này.

Thị trường bán lẻ trực tuyến đang ngày càng phát triển và đồ thể thao là một trong những lĩnh vực có nhu cầu mua sắm lớn, đặc biệt trong các mùa giải thể thao và dịp lễ.

Sử dụng Spring Boot mang lại sự ổn định, bảo mật và hiệu suất cao cho hệ thống. Việc nắm bắt hiệu quả của Spring Boot sẽ giúp phát triển kỹ năng làm việc với các công nghệ phổ biến trong ngành. Xây dựng một ứng dụng web thực tế đòi hỏi sự học hỏi liên tục và thử nghiệm các giải pháp và khái niệm như quản lý dữ liệu, bảo mật, hiệu suất và tối ưu hóa trải nghiệm người dùng.

Xây dựng một ứng dụng thương mại điện tử đòi hỏi sinh viên đối mặt với những thách thức thực tế. Điều này sẽ giúp sinh viên phát triển khả năng giải quyết vấn đề và quản lý dự án.

## Mục đích

***Mục tiêu chung:***

Xây dựng ứng dụng web bán đồ thể thao cho cửa hàng BlueSky sử dụng Spring Boot và Angular nhằm:

* Cung cấp cho khách hàng trải nghiệm mua sắm trực tuyến tiện lợi và hiệu quả.
* Nâng cao hiệu quả hoạt động kinh doanh của cửa hàng BlueSky.

***Mục tiêu cụ thể:***

* Về mặt chức năng: Ứng dụng web phải cung cấp đầy đủ các chức năng cần thiết cho việc mua sắm trực tuyến như:
* Sắp xếp sản phẩm theo danh mục, giá tiền, v.v.
* Xem chi tiết sản phẩm, hình ảnh, mô tả, đánh giá.
* Thêm sản phẩm vào giỏ hàng, thực hiện mua sản phẩm.
* Theo dõi đơn hàng, quản lý tài khoản.
* Ứng dụng web phải có khả năng xử lý các yêu cầu của khách hàng một cách nhanh chóng và hiệu quả.
* Về mặt hiệu quả kinh doanh:
* Tăng doanh thu bán hàng cho cửa hàng BlueSky.
* Tiết kiệm chi phí vận hành.
* Nâng cao hiệu quả quản lý bán hàng.

## Đối tượng và phạm vi nghiên cứu

***Đối tượng nghiên cứu:***

Khách hàng:

* Khách hàng tiềm năng của cửa hàng BlueSky.
* Khách hàng có nhu cầu mua sắm đồ thể thao trực tuyến.

Cửa hàng BlueSky:

* Cửa hàng bán đồ thể thao.
* Cửa hàng muốn mở rộng kênh bán hàng trực tuyến.

***Phạm vi nghiên cứu:***

Chức năng:

* Ứng dụng web phải cung cấp đầy đủ các chức năng cần thiết cho việc mua sắm trực tuyến như:
* Duyệt sản phẩm theo danh mục, thương hiệu, giá cả, v.v.
* Xem chi tiết sản phẩm, hình ảnh, mô tả, đánh giá.
* Thêm sản phẩm vào giỏ hàng, mua hàng.
* Theo dõi đơn hàng, quản lý tài khoản.
* Ứng dụng web phải có khả năng xử lý các yêu cầu của khách hàng một cách nhanh chóng và hiệu quả.

***Công nghệ:***

* Ứng dụng web được xây dựng dựa trên nền tảng Spring Boot và Angular.
* Ứng dụng web sử dụng các công nghệ web hiện đại như HTML5, CSS3, JavaScript.

***Dữ liệu:***

* Ứng dụng web sử dụng cơ sở dữ liệu MySQL để lưu trữ dữ liệu sản phẩm, khách hàng, đơn hàng, ...

***Kết quả dự kiến***

* Báo cáo đồ án tốt nghiệp.
* Phần mềm.

## Ý nghĩa khoa học và thực tiễn của đề tài

***Ý nghĩa khoa học:***

* Đóng góp vào kho tàng kiến thức về xây dựng ứng dụng web bán hàng trực tuyến.
* Đề xuất một mô hình ứng dụng web bán hàng trực tuyến hiệu quả cho các cửa hàng bán đồ thể thao.
* Góp phần nâng cao chất lượng dịch vụ bán hàng trực tuyến trong lĩnh vực bán đồ thể thao.

***Ý nghĩa thực tiễn:***

* Giúp cửa hàng BlueSky tăng doanh thu bán hàng và nâng cao hiệu quả kinh doanh.
* Mang đến cho khách hàng trải nghiệm mua sắm trực tuyến tiện lợi và hiệu quả.
* Có thể áp dụng cho các cửa hàng bán đồ thể thao khác muốn mở rộng kênh bán hàng trực tuyến.

## Cấu trúc đồ án

Ngoài phần Mở đầu và Kết luận, cấu trúc báo cáo gồm 3 chương:

Chương 1: Giới thiệu tổng quan.

Chương 2: Phân tích thiết kế hệ thống.

Chương 3: Thử nghiệm và đánh giá hệ thống

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## Giới thiệu chung

*Thương mại điện tử:* là hình thức mua bán hàng hóa, dịch vụ qua mạng internet. TMĐT đã trở nên phổ biến và phát triển mạnh mẽ trong những năm gần đây, mang lại nhiều lợi ích cho cả người mua và người bán.

TMĐT mang lại nhiều lợi ích đối với cả người mua và người bán. Đối với người mua, việc có thể mua sắm mọi lúc mọi nơi đem lại sự thuận tiện tối đa. Khả năng so sánh giá cả dễ dàng giúp họ tìm được sản phẩm với giá tốt nhất. Không chỉ vậy, mua sắm trực tuyến giúp họ tiết kiệm thời gian và chi phí di chuyển. Hơn nữa, việc có nhiều chương trình khuyến mãi khi mua hàng trực tuyến cũng là một động lực lớn để họ lựa chọn TMĐT.

So với việc mở cửa hàng truyền thống, tham gia vào TMĐT giúp người bán giảm thiểu chi phí mặt bằng và chi phí nhân viên. Không cần phải thuê hoặc mua một vị trí cửa hàng đắt đỏ, không gian lưu trữ lớn, hay chi trả lương cho một đội ngũ nhân viên đông đảo, người bán có thể chuyển đổi sang mô hình kinh doanh trực tuyến với chi phí thấp hơn và hiệu quả hơn.

Các mô hình TMĐT phổ biến bao gồm Business-to-Consumer (B2C) - doanh nghiệp bán hàng trực tiếp cho người tiêu dùng, Business-to-Business (B2B) - doanh nghiệp bán hàng cho doanh nghiệp khác và Consumer-to-Consumer (C2C) - cá nhân bán hàng cho cá nhân khác. Mỗi mô hình này đều có những đặc điểm và ưu điểm riêng, đáp ứng nhu cầu và mong muốn của cả người mua và người bán trong môi trường thương mại điện tử ngày nay.

## Nghiên cứu hiện trang và giải quyết vấn đề

### Hiện trạng của công ty

Cửa hàng giày thể thao BlueSky hiện đang hoạt động dưới hình thức bán trực tiếp tại cửa hàng và chưa có mặt trên nền tảng bán hàng trực tuyến. Tuy nhiên, việc chỉ hoạt động trực tiếp tại cửa hàng đem lại một số vấn đề tồn tại như:

* Giới hạn khách hàng: Phạm vi tiếp cận khách hàng bị hạn chế chỉ trong phạm vi địa lý của cửa hàng, không thể tiếp cận được khách hàng ở xa.
* Hạn chế về thời gian: Cửa hàng chỉ hoạt động trong giờ hành chính, không thể phục vụ khách hàng vào các khung giờ khác như buổi tối hoặc cuối tuần.
* Cạnh tranh: Sự cạnh tranh ngày càng gay gắt từ các đối thủ cũng như các cửa hàng trực tuyến khiến cho việc giữ chân khách hàng trở nên khó khăn hơn.

Để mở rộng thị trường, quảng bá sản phẩm và nâng cao chất lượng dịch vụ, cũng như tăng doanh thu, chủ cửa hàng quyết định xây dựng một trang web giới thiệu sản phẩm và bán giày thể thao trực tuyến. Trang web sẽ cung cấp thông tin chi tiết về các sản phẩm, giúp khách hàng dễ dàng lựa chọn và mua sắm.

Ý tưởng xây dựng website bán hàng:

* Mở rộng phạm vi tiếp cận: Xây dựng một website bán hàng trực tuyến sẽ giúp mở rộng phạm vi tiếp cận khách hàng, kể cả những người ở xa.
* Tăng cơ hội kinh doanh: Website bán hàng trực tuyến sẽ tạo ra cơ hội kinh doanh mới, tăng doanh số bán hàng và lợi nhuận cho cửa hàng.
* Cạnh tranh trực tuyến: Tham gia thị trường trực tuyến sẽ giúp cửa hàng cạnh tranh hiệu quả hơn với các đối thủ và thu hút được nhiều khách hàng mới.

### Cơ cấu tổ chức

Cửa hàng BlueSky hiện tại có cấu trúc tổ chức đơn giản bao gồm:

* Chủ cửa hàng: Người chịu trách nhiệm quản lý toàn bộ hoạt động của cửa hàng và đưa ra quyết định chiến lược.
* Nhân viên bán hàng: Được phân công công việc tại cửa hàng, chịu trách nhiệm tiếp xúc trực tiếp với khách hàng và bán sản phẩm.

### Mô tả hoạt động của các bộ phận

*Chủ cửa hàng*: Chủ cửa hàng đóng vai trò quan trọng trong việc quản lý toàn bộ hoạt động của cửa hàng và đưa ra các quyết định chiến lược để phát triển kinh doanh. Cụ thể:

* Quản lý cửa hàng: Đảm bảo hoạt động hàng ngày của cửa hàng diễn ra một cách trơn tru và hiệu quả. Điều này bao gồm việc quản lý kho hàng, quản lý tài chính và giải quyết các vấn đề phát sinh.
* Chiến lược kinh doanh: Phát triển và thực hiện các chiến lược kinh doanh dựa trên nghiên cứu thị trường và phản hồi từ khách hàng. Điều này có thể bao gồm việc xây dựng các chương trình khuyến mãi, quảng cáo và phát triển sản phẩm mới.
* Quản lý nhân viên: Đảm bảo nhân viên được đào tạo đầy đủ và có đủ kỹ năng để phục vụ khách hàng một cách chuyên nghiệp. Thúc đẩy tinh thần làm việc nhóm và đảm bảo môi trường làm việc tích cực...

*Nhân viên bán hàng*: Nhân viên bán hàng là những người tiếp xúc trực tiếp với khách hàng và đóng vai trò quan trọng trong việc tạo ra trải nghiệm mua sắm tích cực. Cụ thể:

* Tiếp xúc và tư vấn khách hàng: Chào đón khách hàng khi họ đến cửa hàng và cung cấp thông tin chi tiết về sản phẩm. Tư vấn khách hàng về các sản phẩm phù hợp với nhu cầu và mong muốn của họ.
* Thực hiện giao dịch bán hàng: Xử lý các giao dịch bán hàng một cách chuyên nghiệp và nhanh chóng. Đảm bảo việc thanh toán được thực hiện một cách an toàn và tiện lợi cho khách hàng.
* Xử lý phản hồi từ khách hàng: Lắng nghe và xử lý các phản hồi từ khách hàng một cách tích cực. Hỗ trợ khách hàng giải đáp các thắc mắc và giải quyết các vấn đề phát sinh liên quan đến sản phẩm hoặc dịch vụ.

### Xác định yêu cầu

Yêu cầu về chức năng:

* Ứng dụng web phải cung cấp đầy đủ các chức năng cần thiết cho việc mua sắm trực tuyến như:
* Phân loại sản phẩm theo danh mục, thương hiệu, giá cả, v.v.
* Xem chi tiết sản phẩm, hình ảnh, mô tả, đánh giá.
* Thêm sản phẩm vào giỏ hàng, thanh toán đơn hàng.
* Theo dõi đơn hàng, quản lý tài khoản.
* Ứng dụng web phải có khả năng xử lý các yêu cầu của khách hàng một cách nhanh chóng và hiệu quả.

### Yêu cầu về phi chức năng

* Giao diện ứng dụng web phải đẹp mắt, dễ sử dụng, thân thiện với người dùng.
* Ứng dụng web phải có khả năng tương thích với các thiết bị di động.
* Ứng dụng web phải đảm bảo an toàn thông tin cho khách hàng.
* Ứng dụng web có thể dễ dàng mở rộng thêm các chức năng mới trong tương lai.

## Công nghệ sử dụng trong dự án

### Giới thiệu một số công nghệ phổ biến

Trong lĩnh vực phát triển phần mềm và thương mại điện tử hiện nay, có một loạt các công nghệ được sử dụng phổ biến. Đối với việc xây dựng giao diện người dùng, các công nghệ như HTML/CSS (bao gồm SCSS), JavaScript, cùng với các framework như React, Angular và Vue.js đều được ưa chuộng. Đây là những công cụ mạnh mẽ giúp phát triển các trang web và ứng dụng web có giao diện thân thiện, tương tác và linh hoạt.

Trong phát triển backend, các công nghệ như Node.js, Express.js, Django (Python), Flask (Python), Ruby on Rails (Ruby) và ASP.NET đều được sử dụng rộng rãi. Các framework và ngôn ngữ này cung cấp các công cụ mạnh mẽ để xây dựng và quản lý các hệ thống backend phức tạp, cung cấp các API và xử lý logic kinh doanh.

Đối với cơ sở dữ liệu, các lựa chọn phổ biến bao gồm MS SQL Server, MongoDB (NoSQL), MySQL (Relational) và PostgreSQL (Object-Relational). Các hệ thống cơ sở dữ liệu này đều có điểm mạnh riêng, phù hợp với các yêu cầu về lưu trữ và truy xuất dữ liệu của các ứng dụng và hệ thống khác nhau.

Trong việc quản lý phiên bản, Git là công cụ được sử dụng rộng rãi nhất. Nó cung cấp các tính năng mạnh mẽ cho việc quản lý mã nguồn và hợp tác trong dự án phần mềm.

Cuối cùng, trong quá trình kiểm thử, các công cụ như Jest, Swagger và Postman được sử dụng để đảm bảo chất lượng và tính ổn định của ứng dụng. Chúng cung cấp các cơ chế kiểm thử tự động và kiểm tra tương tác của ứng dụng với các API và giao diện người dùng.

### Giới thiệu một số kiến trúc phổ biến

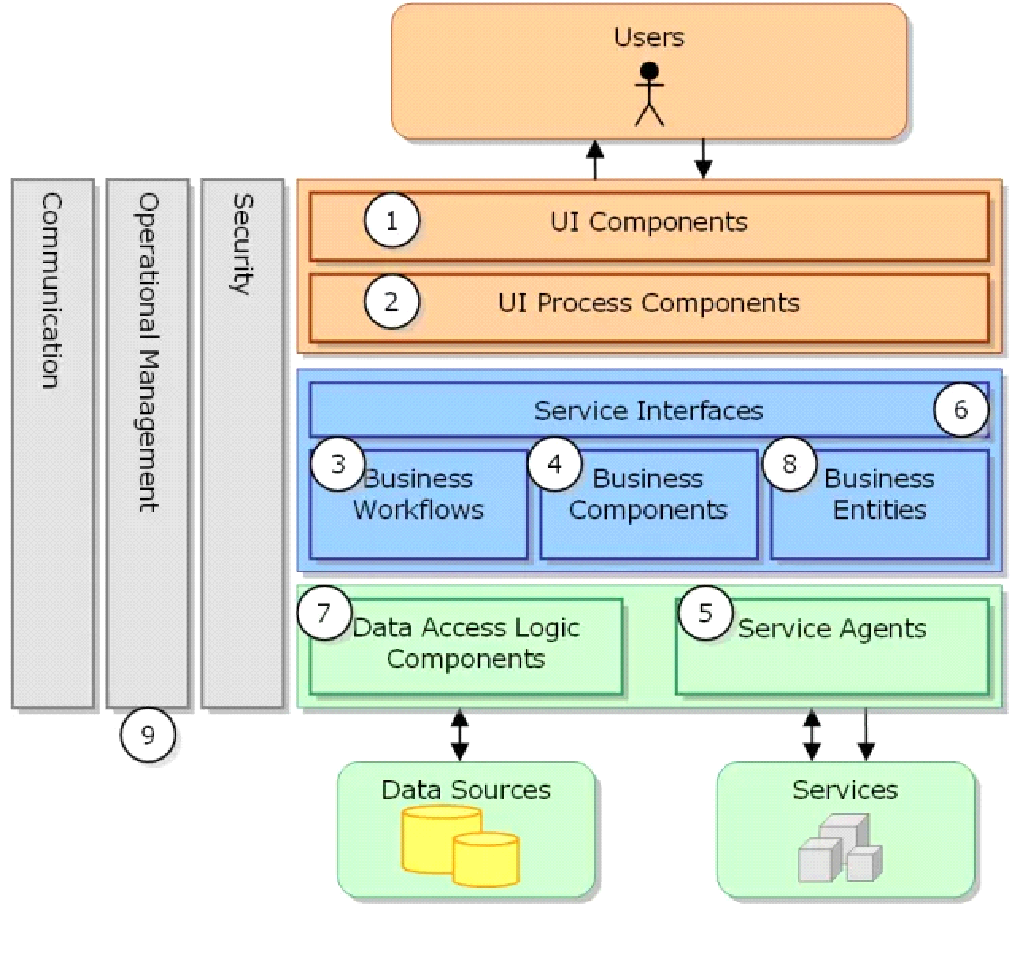
**Mô hình 3 lớp (three layer)**

Mô hình này phát huy hiệu quả nhất khi bạn xây dựng một hệ thống lớn, việc quản lý code và xử lý dữ liệu lỗi dễ dàng hơn

Ưu điểm:

* Phân loại rõ ràng các lớp có các nhiệm vụ khác nhau. Từ đó ta có thể quản lý và bảo trì dự án tốt hơn.
* Dễ dàng phân loại các hành động tại Business.
* Dễ dàng phân loại các hàm truy xuất tại cơ sở dữ liệu, phân loại hàm theo bảng.
* Ứng dụng được cho các project lớn ở bên ngoài.

Ảnh dưới minh họa kiến trúc mô hình 3 lớp:



*Hình 1.1. Kiến trúc mô hình 3 lớp*

**Presentation Layer (GUI)**

Có hai thành phần chính sau đây với những tác vụ cụ thể :

*UI Components*: gồm các thành phần tạo nên giao diện của ứng dụng (GUI). Chúng chịu trách nhiệm thu nhận và hiển thị dữ liệu cho người dùng: Ví dụ : textbox, button, combobox, …

*UI Process Components*: là thành phần chịu trách nhiệm quản lý các quá trình chuyển đổi giữa các UI…

Ví dụ:

* Sắp xếp quá trình kiểm tra thông tin khách hàng:
* Hiển thị màn hình tra cứu ID.
* Hiển thị màn hình thông tin chi tiết khách hàng tương ứng.
* Hiển thị màn hình liên lạc với khách hàng.

**Business Layer (BLL)**

Lớp này gồm 4 thành phần:

*Service Interface:* là thành phần giao diện lập trình mà lớp này cung cấp cho lớp Presentation sử dụng.

*Business Workflows:* chịu trách nhiệm xác định và điều phối các quy trình nghiệp vụ gồm nhiều bước và kéo dài. Những quy trình này phải được sắp xếp và thực hiện theo một thứ tự chính xác.

*Business Components:* chịu trách nhiệm kiểm tra các quy tắc nghiệp vụ, ràng buộc logic và thực hiện các công việc. Các thành phần này cũng thực hiện các dịch vụ mà Service Interface cung cấp và Business Workflows sẽ sử dụng nó.

*Business Entities:* thường được sử dụng như Data Transfer Objects (DTO). Bạn có thể sử dụng để truyền dữ liệu giữa các lớp (Presentation và Data Layer). Chúng thường là cấu trúc dữ liệu (DataSets, XML…) hay các lớp đối tượng đã được tùy chỉnh. Ví dụ: tạo 1 class Student lưu trữ các dữ liệu về tên, ngày sinh, mã sinh viên, lớp.

**Data Layer (DAL)**

*Data Access Logic Components*: chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu (Data Sources) như XML, file system,… Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì. Service Agents: giúp gọi và tương tác với các dịch vụ từ bên ngoài một cách dễ dàng và đơn giản.

**Kiến Trúc Mô Hình Microservices**

A diagram of a software application

AI-generated content may be incorrect.

*Hình 1.2 Mô hình kiến trúc Microservices*

**Micro-service**

Đặc trưng này được thể hiện ngay từ tên của kiến trúc. Nó là microservice chứ không phải là miniservice hay nanoservice. Trên thực tế không tồn tại mô hình kiến trúc cho miniservice hay nanoservice. Từ microservice được sử dụng để giúp người thiết kế có cách tiếp cận đúng đắn. Một ứng dụng lớn cần được chia nhỏ ra thành nhiều thành phần, các thành phần đó cần tách biệt về mặt dữ liệu (database) và phải đủ nhỏ cả về mặt kích cỡ và độ ảnh hưởng của nó trong hệ thống, khi thêm một microservice vào hệ thống cũng nên đảm bảo rằng nó đủ nhỏ để dễ dàng tháo gỡ, xóa bỏ khỏi hệ thống mà không ảnh hưởng nhiều tới các thành phần khác.

**Tính độc lập**

Các microservice hoạt động tách biệt nhau trong hệ thống, do vậy việc build một microservice cũng độc lập với việc build các microservice khác. Thông thường, để tiện cho việc phát triển và duy trì các microservice, người phát triển nên viết các built script khác nhau cho mỗi microservice.

Do tính tách biệt này mà mỗi microservice đều dễ dàng thay thế và mở rộng. Hơn thế nữa, nó còn giúp việc phát triển các microservice linh động hơn, các microservice có thể được phát triển bởi các team khác nhau, dùng các ngôn ngữ khác nhau và tiến độ phát triển dự án cũng nhanh hơn do không có sự phụ thuộc giữa các team, mỗi team có thể chủ động quản lý phần việc riêng của mình.

**Tính chuyên biệt**

Mỗi microservice là một dịch vụ chuyên biệt, có thể hoạt động độc lập, thông thường mỗi microservice đại diện cho một tính năng mà các công ty/ doanh nghiệp muốn cung cấp tới người dùng, do vậy người thiết kế hệ thống microservice cần hiểu rõ về hoạt động kinh doanh của công ty. Các đầu vào đầu ra và chức năng của mỗi microservice cần được định nghĩa rõ ràng.

**Phòng chống lỗi**

Kiến trúc microservice sinh ra là để dành cho các hệ thống từ lớn đến vô cùng lớn. Nó áp dụng phương pháp chia để trị, phương pháp này giúp việc áp dụng các công cụ, kỹ thuật cho việc giám sát, phòng chống lỗi phần mềm, lỗi hệ thống hiệu quả.

Khi một thành phần trong hệ thống bị lỗi, nó có thể được thay thế bằng các thành phần dự phòng một cách dễ dàng, trong quá trình thay thế thành phần bị lỗi, các thành phần khác vẫn hoạt động bình thường, do vậy hoạt động của toàn bộ hệ thống sẽ không hoặc ít bị gián đoạn.

### Công nghệ sử dụng

**Giới thiệu về Angular**

Giới thiệu về Angular

Angular là một framework mã nguồn mở miễn phí chuyên dụng cho việc thiết kế web. Angular được phát triển từ những năm 2009 được duy trì bởi Google và được xem là framework trình duyệt mạnh mẽ nhất chuyên dụng bởi các lập trình viên HTML cao cấp.

Kiến trúc của Angular

Angular là một framework dựa trên mô hình Model-View-Controller (MVC). Trong đó Model là thành phần trung tâm thể hiện hành vi của một ứng dụng và quản lý dữ liệu. View được tạo ra dựa trên thông tin của Model. Controller đóng vai trò trung gian giữa Model và View.

Trong một ứng dụng Angular có 8 thành phần:

* Module
* Template
* Component
* Metadata
* Service
* Dependency Injection
* Databinding
* Directive

A diagram of a business process

AI-generated content may be incorrect.

*Hình 1.3 Kiến trúc của Angular*

**Modules**

Một ứng dụng Angular có một module gốc, tên là AppModule, cung cấp cơ chế bootstrap để khởi chạy ứng dụng.

**Component**

Mỗi component trong ứng dụng xác định một lớp chứa dữ liệu và logic của ứng dụng. Một thành phần thường xác định một phần của giao diện người dùng (UI).

**Templates**

Angular template sẽ kết hợp các Angular markup với HTML để sửa đổi các phần tử HTML trước khi chúng được hiển thị.

Có hai loại liên kết dữ liệu:

* *Liên kết sự kiện*: Cho phép ứng dụng của bạn phản hồi thông tin đầu vào của người dùng trong môi trường đích bằng cách cập nhật dữ liệu ứng dụng của bạn.
* *Ràng buộc thuộc tính:* Cho phép người dùng thêm các giá trị được tính toán từ dữ liệu ứng dụng của bạn vào HTML.

**Metadata**

Metadata cho Angular biết cách xử lý một lớp. Nó được sử dụng để trang trí lớp để nó có thể cấu hình hành vi mong đợi của một lớp.

**Service**

Khi bạn có dữ liệu hoặc logic không được liên kết với view nhưng phải được chia sẻ giữa các thành phần, một lớp service sẽ được tạo. Lớp luôn được liên kết với decorator @Injectible.

**Ưu điểm**

* Custom Components:

Angular cho phép người dùng xây dựng các component của riêng họ và có thể đóng gói chức năng cùng với logic kết xuất thành các phần có thể tái sử dụng. Nó cũng hoạt động tốt với các component của web.

* Data Binding:

Angular cho phép người dùng dễ dàng di chuyển dữ liệu từ mã JavaScript sang view và phản hồi với các sự kiện của người dùng mà không cần phải viết bất kỳ mã nào theo cách thủ công.

* Dependency Injection:

Angular cho phép người dùng viết các service mô-đun và đưa chúng vào bất cứ nơi nào họ cần. Điều này cải thiện khả năng kiểm tra và khả năng tái sử dụng của các service giống nhau.

* Testing: Có thể dễ dàng Unit test mọi phần trong ứng dụng của mình.
* Comprehensive:

Angular là một full-fledged Framework và có thể cung cấp các giải pháp tiện ích cho giao tiếp máy chủ, định tuyến trong ứng dụng của bạn và hơn thế nữa.

* Browser Compatibility:

Angular cung cấp đa nền tảng và tương thích với nhiều trình duyệt. Một ứng dụng Angular thường có thể chạy trên tất cả các trình duyệt (Ví dụ: Chrome, Firefox) và hệ điều hành, chẳng hạn như Windows, MacOS và Linux.

**Giới thiệu về Spring Boot**

Khái niệm

Spring Boot là một dự án được xây dựng dựa trên Spring Framework. Cung cấp một cách dễ dàng hơn và nhanh hơn để thiết lập, định cấu hình và chạy cả ứng dụng đơn giản và ứng dụng dựa trên web.

Spring Boot là một module Spring cung cấp tính năng RAD cho Spring Framework được sử dụng để tạo ra một ứng dụng dựa trên Spring độc lập.

Lý do nên sử dụng Spring Boot:

* Phương pháp tiêm phụ thuộc được sử dụng trong Spring Boot.
* Chứa khả năng quản lý giao dịch cơ sở dữ liệu mạnh mẽ.
* Đơn giản hóa việc tích hợp với các khung công tác Java khác như JPA/ Hibernate ORM…

Ưu điểm, nhược điểm của Spring Boot

**Các ưu điểm chính của Spring Boot bao gồm:**

* Không sinh code cấu hình, đồng thời không yêu cầu phải cấu hình bằng XML.
* Phát triển các ứng dụng dựa trên Spring một cách tiết kiệm thời gian và dễ dàng.
* Tự động cấu hình tất cả các components cho một ứng dụng Spring cấp sản xuất.
* Các máy chủ nhúng được tạo sẵn (Tomcat, Jetty và Undertow), dẫn đến việc triển khai ứng dụng được tăng tốc và hiệu quả hơn.
* Endpoint HTTP, cho phép nhập các tính năng bên trong ứng dụng
* Nhiều lựa chọn bổ sung, hỗ trợ nhà phát triển làm việc với cơ sở dữ liệu được nhúng và trong bộ nhớ.
* Dễ dàng truy cập cơ sở dữ liệu và các dịch vụ hàng đợi như MySQL, Oracle, MongoDB, Redis, ActiveMQ và các dịch vụ khác.
* Tích hợp trơn tru với hệ sinh thái Spring.
* Cộng đồng lớn và rất nhiều hướng dẫn, tạo điều kiện cho giai đoạn làm quen.

**Bên cạnh đó Spring Boot có một số nhược điểm sau:**

* Thiếu kiểm soát: Do style cố định, Spring Boot tạo ra nhiều phụ thuộc không được sử dụng dẫn đến kích thước tệp triển khai lớn.
* Quá trình chuyển đổi dự án Spring cũ hoặc hiện có thành các ứng dụng Spring Boot nhiều khó khăn và tốn thời gian.

**Các tính năng của Spring Boot**

Spring Boot là một framework phát triển ứng dụng Java dựa trên nền tảng của Spring Framework, với những tính năng nổi bật giúp tạo ra các ứng dụng Spring độc lập và linh hoạt. Một số điểm đáng chú ý của Spring Boot bao gồm khả năng nhúng trực tiếp các máy chủ như Tomcat, Jetty, hoặc Undertow mà không cần phải triển khai ra file WAR, giúp giảm bớt thời gian và công sức trong quá trình triển khai ứng dụng.

Spring Boot cũng hỗ trợ các starter dependency, giúp cho việc cấu hình Maven trở nên đơn giản hơn bao giờ hết. Bằng cách sử dụng các starter dependency, người phát triển có thể dễ dàng thêm các tính năng và module vào ứng dụng mà không cần phải lo lắng về việc cấu hình chi tiết.

Một trong những điểm mạnh của Spring Boot là khả năng tự động chạy cấu trình Spring khi cần thiết, giúp cho việc phát triển và triển khai ứng dụng trở nên dễ dàng hơn. Nó cũng loại bỏ sự cần thiết phải sinh code cấu hình và không yêu cầu phải cấu hình bằng XML, thay vào đó sử dụng các annotation và convention-over-configuration để giảm thiểu sự phức tạp và tăng tính linh hoạt của ứng dụng.

**Kiến trúc Spring Boot**

Kiến trúc của Spring Boot là một trong những yếu tố quan trọng định hình cách mà ứng dụng được tổ chức và triển khai. Spring Boot tuân thủ một kiến trúc phân lớp phân cấp, giúp tạo ra một hệ thống có cấu trúc rõ ràng và dễ bảo trì.

Spring Boot là một framework Java phổ biến trong phát triển ứng dụng và nó tuân theo một kiến trúc phân lớp cấu trúc phân cấp. Có tổng cộng bốn tầng trong Spring Boot, mỗi tầng đóng vai trò quan trọng trong việc xây dựng và phát triển ứng dụng.

Tầng đầu tiên là lớp Presentation, còn được gọi là lớp giao diện. Lớp này chịu trách nhiệm tương tác trực tiếp với người dùng thông qua các giao diện người dùng (UI). Nó là nơi mà các yêu cầu từ người dùng được nhận và xử lý, sau đó gửi tới lớp Business để thực hiện các logic xử lý phù hợp.

Tiếp theo là lớp Business, hay lớp nghiệp vụ. Đây là nơi chứa các logic xử lý chính của ứng dụng. Các tác vụ như xử lý dữ liệu, tính toán và quyết định logic đều được thực hiện ở đây. Lớp Business thường giao tiếp với Lớp Persistence để truy xuất và cập nhật dữ liệu khi cần thiết.

Lớp Persistence, hoặc lớp truy cập dữ liệu, là tầng thứ ba trong kiến trúc. Lớp này chịu trách nhiệm truy cập và thao tác với cơ sở dữ liệu. Nó tương tác trực tiếp với các hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL, MongoDB, vv., để lưu trữ và truy xuất dữ liệu.

Cuối cùng là lớp Database, là nơi lưu trữ dữ liệu của ứng dụng. Đây có thể là cơ sở dữ liệu thực sự hoặc các bộ nhớ đệm dữ liệu như Redis. Lớp này là điểm cuối của ứng dụng và là nơi mà dữ liệu được lưu trữ và truy xuất từ.

Tổng thể, kiến trúc phân lớp trong Spring Boot giúp tách biệt các trách nhiệm và logic của ứng dụng, tạo ra một hệ thống có cấu trúc rõ ràng và dễ dàng bảo trì.

Hình dưới đây minh họa kiến trúc của Spring Boot:

A diagram of a business layer

AI-generated content may be incorrect.

*Hình 1.4 Kiến trúc phân tầng Spring Boot*

A diagram of a computer system

AI-generated content may be incorrect.

*Hình 1.5 Kiến trúc luồng Spring Boot*

**Hệ quản trị cơ sở dữ liệu MySQL**

MySQL chính là một hệ thống mã nguồn mở quản lý cơ sở dữ liệu. Cũng như tất cả các cơ sở dữ liệu quan hệ khác, MySQL cũng sẽ lưu trữ dữ liệu ở bên trong các bảng được tạo thành từ các cột và các hàng của nó.

SQL là một tập hợp các dữ liệu có cấu trúc. Đó có thể là tất cả mọi thứ từ danh sách mua sắm đơn giản đến nơi chứa lượng thông tin khổng lồ trong mạng công ty hay thư viện ảnh của bạn.

Bên cạnh đó, nó là một phần vô cùng quan trọng trong nhiều phần mềm phổ biến nhất để xây dựng và duy trì mọi thứ, từ các ứng dụng web dành cho khách hàng đến các dịch vụ B2B mạnh mẽ, dựa trên dữ liệu.

Với bản chất là nguồn mở, bộ tính năng phong phú của nó, kết hợp cùng với sự phát triển vượt bậc ngày càng tăng và hỗ trợ liên tục từ Oracle, có nghĩa là các tổ chức quan trọng trên internet như Facebook, Flickr, Twitter, Wikipedia và YouTube đều sử dụng các phần phụ trợ của MySQL.

A logo with a dolphin

AI-generated content may be incorrect.

*Hình 1.6 Hệ quản trị cơ sở dữ liệu MySQL*

**Các đặc điểm của MySQL**

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở, được cấp phép theo Giấy phép GNU (GPL), cho phép người dùng sử dụng, sửa đổi và phân phối nó một cách miễn phí. Với hiệu suất cao, MySQL có thể xử lý lượng truy vấn lớn một cách hiệu quả, đồng thời có khả năng mở rộng để đáp ứng nhu cầu của người dùng bằng cách thêm nhiều máy chủ vào cụm MySQL.

Điều đặc biệt là MySQL dễ cài đặt, cấu hình và sử dụng, làm cho việc triển khai và quản lý cơ sở dữ liệu trở nên đơn giản. Nó cũng hỗ trợ nhiều nền tảng hệ điều hành như Windows, Linux, macOS và Solaris, cùng với việc hỗ trợ nhiều ngôn ngữ lập trình phổ biến như PHP, Java, Python và C++.

Cuối cùng, MySQL có một cộng đồng người dùng và nhà phát triển lớn, luôn sẵn sàng hỗ trợ người dùng trong việc sử dụng MySQL, cung cấp sự đảm bảo và hỗ trợ cho người dùng trên toàn thế giới.

**Cách hoạt động của MySQL**

MySQL thực hiện kiến trúc client-server, trong đó client là chương trình gửi yêu cầu đến server và server là chương trình xử lý yêu cầu và trả về kết quả cho client. Quá trình này bao gồm các bước như sau: đầu tiên, client kết nối với server thông qua giao thức TCP/IP. Sau đó, client gửi thông tin xác thực như tên đăng nhập và mật khẩu đến server. Tiếp theo, client gửi yêu cầu đến server và server tiếp nhận yêu cầu và thực hiện xử lý. Kết quả được trả về cho client, sau đó client ngắt kết nối với server.

MySQL hỗ trợ nhiều loại dữ liệu khác nhau bao gồm số nguyên, số thập phân, chuỗi, ngày giờ và BLOB (Binary Large Object). Điều này cho phép người dùng lưu trữ và quản lý nhiều loại dữ liệu khác nhau một cách linh hoạt.

Để truy vấn và thao tác dữ liệu, MySQL sử dụng ngôn ngữ truy vấn SQL (Structured Query Language), là một ngôn ngữ tiêu chuẩn được sử dụng rộng rãi trong hệ quản trị cơ sở dữ liệu quan hệ. MySQL hỗ trợ nhiều câu lệnh và chức năng mạnh mẽ như JOIN, SUBQUERY, TRIGGERS, STORED PROCEDURES và VIEWS, giúp người dùng thực hiện các thao tác phức tạp một cách dễ dàng.

Để tăng hiệu suất và đảm bảo tính ổn định của cơ sở dữ liệu, MySQL cung cấp các tính năng như indexing, caching và replication. Indexing giúp tối ưu hóa tìm kiếm và truy vấn dữ liệu bằng cách tạo ra các chỉ mục cho các trường quan trọng trong bảng. Caching cho phép lưu trữ dữ liệu phổ biến trong bộ nhớ để giảm thiểu thời gian truy cập cơ sở dữ liệu.

Ngoài ra, MySQL cung cấp nhiều công cụ để quản lý và bảo trì cơ sở dữ liệu, bao gồm MySQL Workbench, MySQL Administrator và mysqldump. Các công cụ này giúp người dùng quản lý cơ sở dữ liệu một cách hiệu quả và thuận tiện, từ việc thiết lập, xây dựng cơ sở dữ liệu đến việc sao lưu và phục hồi dữ liệu.

**Giới thiệu về MySQL Workbench**

MySQL Workbench là một công cụ thiết kế và quản lý cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Oracle, cung cấp giao diện đồ họa thống nhất để hỗ trợ các kiến trúc sư cơ sở dữ liệu, nhà phát triển và quản trị viên (DBA). Công cụ này tích hợp các chức năng chính như mô hình hóa dữ liệu (data modeling), phát triển SQL, và quản trị server, giúp đơn giản hóa các tác vụ phức tạp như thiết kế sơ đồ ER (Entity-Relationship Diagram), tạo và tối ưu hóa truy vấn SQL, cũng như quản lý người dùng, sao lưu, và phục hồi dữ liệu. MySQL Workbench hỗ trợ nhiều nền tảng bao gồm Windows, Linux và macOS, và được sử dụng rộng rãi nhờ khả năng trực quan hóa, reverse/forward engineering, cũng như quản lý thay đổi cơ sở dữ liệu hiệu quả. Trong đồ án này, MySQL Workbench được sử dụng để thiết kế cấu trúc cơ sở dữ liệu, minh họa mối quan hệ giữa các bảng thông qua sơ đồ ER, và hỗ trợ kiểm tra, tối ưu hóa các truy vấn SQL, góp phần nâng cao tính chuyên nghiệp và hiệu quả của dự án.

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1 Các yêu cầu phi chức năng

**Thân thiện với người dùng**

Theo các nghiên cứu, 76% người tiêu dùng cho biết tính thân thiện của một trang web là yếu tố quan trọng nhất. Mục tiêu là giúp người mua hàng dễ dàng đạt được mục tiêu mua sắm của họ một cách nhanh chóng mà không gặp phải những phức tạp không cần thiết, có thể làm gián đoạn trải nghiệm mua hàng của họ. Với thời gian quý báu của người bán trực tuyến, việc cung cấp một trải nghiệm mua sắm dễ dàng nên là ưu tiên hàng đầu. Điều này có thể đạt được thông qua việc tạo ra các danh mục mua sắm rõ ràng, tích hợp các bộ lọc hiệu quả và khả năng so sánh sản phẩm tiện lợi.

Một chức năng tìm kiếm được cải thiện sẽ giúp người dùng nhanh chóng tìm thấy sản phẩm mong muốn chỉ bằng cách nhập tên sản phẩm. Hệ thống sẽ tự động hiển thị danh sách các sản phẩm phù hợp nhất, giúp tiết kiệm thời gian và công sức cho người dùng. Đồng thời, việc tạo ra một giao diện điều hướng trực quan và hiệu quả cũng là một yêu cầu thiết yếu. Giao diện này không chỉ dễ sử dụng mà còn giúp người dùng tìm thấy thông tin và sản phẩm họ cần một cách nhanh chóng và dễ dàng.

Ngoài ra, tự động điền thông tin vào các biểu mẫu cũng sẽ giúp tăng tính tiện ích và trải nghiệm người dùng. Khi thông tin cá nhân của khách hàng được tự động điền vào các biểu mẫu mua sắm, quy trình mua hàng sẽ trở nên mượt mà hơn, giảm thiểu thời gian và công sức của người mua. Nhờ vào những cải tiến này, trải nghiệm mua sắm trực tuyến sẽ trở nên thú vị và hiệu quả hơn, đáp ứng tốt hơn nhu cầu của người tiêu dùng hiện đại.

**Chức năng hiển thị hình ảnh**

Người mua hàng mong muốn được xem sản phẩm từ nhiều góc độ khác nhau và trong các bối cảnh sử dụng đa dạng. Họ muốn có khả năng phóng to hình ảnh để cảm nhận rõ ràng về sản phẩm. Tuy nhiên, điều quan trọng là phải đảm bảo rằng trang web của bạn có thể tải nhanh chóng ngay cả khi sử dụng hình ảnh chất lượng cao. Theo Adobe, hình ảnh không tải được hoặc mất quá nhiều thời gian để tải có thể làm tăng tỷ lệ thoát trang của người dùng lên đến 39%.

Do đó, việc tối ưu hóa hình ảnh sao cho vừa giữ được chất lượng cao, vừa đảm bảo tốc độ tải trang nhanh là yếu tố then chốt để giữ chân khách hàng và giảm tỷ lệ thoát trang. Một số kỹ thuật có thể áp dụng bao gồm nén hình ảnh, sử dụng định dạng hình ảnh hiệu quả để cải thiện hiệu suất tải trang.Phân tích yêu các cầu chức năng của phần mềm

## 2.2 Các yêu cầu chức năng

**Các tác nhân hệ thống**

|  |  |
| --- | --- |
| Người sử dụng | Mô tả |
| Người quản trị | Là người chủ kinh doanh mặt hàng của cửa hàng hoặc là quản lý cho cửa hàng/website, người quản trị chịu trách nhiệm điều khiển và duy trì, cập nhật thông tin của sản phẩm, danh mục sản phẩm… |
| Khách hàng | - Là người có như cầu muốn tìm kiếm các sản phẩm có mặt tại của hàng trực tuyến, khách hàng có thể xem sản phẩm, tìm kiếm, thanh toán đơn hàng.  - Khách hàng có thể quản lý thông tin cá nhân, thông tin lịch sử mua hàng. |

*Bảng 2.1: Các tác nhân hệ thống*

**Các chức năng**

* Quản lý người dùng:

Hệ thống cần cung cấp các chức năng quản lý người dùng như đăng ký và đăng nhập. Người dùng có thể tạo tài khoản mới và truy cập vào hệ thống bằng cách đăng nhập vào tài khoản của mình.

* Quản lý danh mục:

Quản trị viên cần có khả năng thêm, sửa, xóa và tìm kiếm danh mục sản phẩm. Họ có thể thêm danh mục mới cho sản phẩm, chỉnh sửa thông tin của danh mục, xóa các danh mục không cần thiết và tìm kiếm danh mục theo tên hoặc các thuộc tính khác.

* Quản lý sản phẩm:

Quản trị viên cần có khả năng thêm, sửa, xóa và tìm kiếm sản phẩm. Họ có thể thêm sản phẩm mới vào hệ thống, chỉnh sửa thông tin của sản phẩm, xóa các sản phẩm không cần thiết và tìm kiếm sản phẩm theo tên, mô tả hoặc các thuộc tính khác.

* Quản lý đơn hàng:

Quản trị viên cần có khả năng xác nhận đơn hàng của khách hàng, hủy đơn hàng nếu cần thiết và xác nhận thanh toán thành công của đơn hàng.

* Quản lý khách hàng:

Hệ thống cần cho phép quản trị viên thêm, sửa, xóa thông tin khách hàng.

* Quản lý liên hệ:

Quản trị viên cần có khả năng xem, xóa và phản hồi các yêu cầu liên hệ từ khách hàng.

* Thống kê:

Hệ thống cần cung cấp các chức năng thống kê doanh thu theo tháng/năm, thống kê sản phẩm bán chạy và thống kê hàng tồn kho.

* Tính năng khác:

Các tính năng khác bao gồm xem sản phẩm theo danh mục, xem chi tiết sản phẩm, liên hệ với cửa hàng, thanh toán, thêm vào giỏ hàng, theo dõi đơn hàng, quản lý giỏ hàng, đánh giá sản phẩm và thêm sản phẩm vào danh sách yêu thích.

## 2.3 Biểu đồ Use Case tổng quan

**Biểu đồ use case tổng quan**

A diagram of a person's relationship

AI-generated content may be incorrect.

*Hình 2.1: Use case tổng quan toàn bộ hệ thống*

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.2: Biểu đồ use case tổng quan khách hàng*

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.3 Biểu đồ use case tổng quan admin*

**Biểu đồ use case chính**

Ảnh minh họa biểu đồ use case chính:

A diagram of a person's work flow

AI-generated content may be incorrect.

*Hình 2.4: Biểu đồ use case chính*

**Biểu đồ use case thứ cấp**

Ảnh minh họa biểu đồ use case thứ cấp:

**A diagram of a group of text

AI-generated content may be incorrect.**

*Hình 2.5: Biểu đồ use case thứ cấp*

**Mô tả chi tiết các use case**

***Mô tả use case đăng ký***

**Tên Use case:** Đăng ký.

**Mô tả vắn tắt:** Use case này cho phép người dùng đăng ký tài khoản.

**Luồng sự kiện:**

* Luồng cơ bản:

Use case bắt đầu khi người dùng nhấn nút đăng ký trong mục đăng nhập/đăng ký. Hệ thống hiển thị form đăng ký (email, OTP, họ tên, mật khẩu, địa chỉ, số điện thoại). Người dùng nhập thông tin, hệ thống lưu vào bảng “users” và thông báo “Đăng ký thành công”. Use case kết thúc.

* Luồng rẽ nhánh:

Ở bước 2, nếu người dùng nhập sai mã OTP, email, mật khẩu hoặc số điện thoại, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại. Nếu mất kết nối cơ sở dữ liệu trong quá trình thực hiện, hệ thống 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:** Thông tin sẽ được lưu và bảng “users”

**Điểm mở rộng:** Không có

***Mô tả use case đăng nhập***

**Tên Use case: Đăng nhập.**

**Mô tả vắn tắt:** Use case cho phép người dùng đăng nhập để xác nhận quyền truy cập hệ thống.

**Luồng sự kiện:**

* Luồng cơ bản:

Use case bắt đầu khi người dùng nhấn nút “Đăng nhập” trên trang chủ. Hệ thống hiển thị form đăng nhập (email, password). Người dùng nhập thông tin và nhấn “Đăng nhập”. Hệ thống kiểm tra email, password trong bảng “users”, cho phép đăng nhập và thông báo “Đăng nhập thành công”. Use case kết thúc.

* Luồng rẽ nhánh:

Tại bước 2, nếu người dùng nhập sai email hoặc mật khẩu, hệ thống hiển thị thông báo lỗi và quay lại bước 2. Nếu mất kết nối cơ sở dữ liệu, hệ thống 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, thì người dùng sẽ truy cập vào hệ thống bằng tài khoản đăng nhập.

**Điểm mở rộng**: Đăng ký tài khoản

***Mô tả use case xem sản phẩm theo danh mục***

**Tên Use case:** Xem sản phẩm theo danh mục.

**Mô tả vắn tắt:** Use case này cho phép khách hàng xem sản phẩm theo danh mục.

**Luồng sự kiện:**

* Luồng cơ bản:

Use case bắt đầu khi khách hàng nhấp vào danh mục trong menu “Sản phẩm”. Hệ thống lấy category\_name từ bảng “categories” và hiển thị. Khi chọn danh mục, hệ thống lấy thông tin từ bảng “products”, “categories”, “rates”, hiển thị danh sách sản phẩm (name, price, category\_name, image, discount, rating). Use case kết thúc.

* Luồng rẽ nhánh:

Tại bất kỳ bước nào trong luồng cơ bản nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiện 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:** Thêm vào giỏ hàng, xem chi tiết, yêu thích.

***Mô tả use case xem chi tiết sản phẩm***

**Tên Use case:** Xem chi tiết sản phẩm.

**Mô tả vắn tắt:** Use case này cho phép người dùng xem thông tin chi tiết sản phẩm.

Luồng sự kiện:

* Luồng cơ bản:

Use case này bắt đầu khi người dùng chọn vào một sản phẩm. Hệ thống sẽ lấy ra số lượng đánh giá từ bảng “rates”, lấy ra số lượng yêu thích từ bảng “favorites”, lấy hình ảnh sản phẩm từ bảng “image”, lấy thông tin chi tiết về sản phẩm bao gồm: name, description, sold, quantity, discount, branch, material, size, image từ bảng “products” và hiển thị ra màn hình. 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 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:** Thêm vào giỏ hang

***Mô tả use case tìm kiếm sản phẩm***

**Tên Use case:** Tìm kiếm sản phẩm.

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

**Luồng sự kiện:**

* Luồng cơ bản:

Use case này bắt đầu khi khách hàng nhập từ khóa tìm kiếm vào thanh tìm kiếm trên trang web và chọn vào nút “Tìm kiếm” hoặc nhấn enter. Hệ thống sẽ lấy thông tin của các sản phẩm liên quan đến từ khóa đó từ bảng “products”, “categories”, “rates” gồm: image, name, price, discount, rating, category\_name và hiển thị danh sách sản phẩm cần tìm ra. Use case kết thúc.

* Luồng rẽ nhánh:

Tại bước 1 của luồng cơ bản, nếu sản phẩm cần tìm kiếm không có trong bảng “products”. Màn hình sẽ đưa ra thông báo “Không tìm thấy sản phẩm có từ khóa liên quan”.

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

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Không có

**Hậu điều kiện:** Không có

**Điểm mở rộng:** Không có

***Mô tả use case quản lý giỏ hàng***

**Tên Use case:** Quản lý giỏ hàng.

**Mô tả vắn tắt**: Use case này cho phép khách hàng thêm sản phẩm vào giỏ hàng, xem, cập nhật, xóa sản phẩm có trong giỏ hàng.

**Luồng sự kiện:**

* Luồng cơ bản:

Use case này bắt đầu khi khách hàng chọn một sản phẩm và kích vào nút “Thêm vào giỏ hàng”. Hệ thống lấy thông tin từ bảng “products” gồm: name, quantity, size, price” của sản phẩm đã chọn và thêm sản phẩm vào bảng “cart\_detail”. Sau đó hiển thị thông báo “Thêm sản giỏ hàng thành công”. Use case kết thúc.

Use case này bắt đầu khi khách hàng chọn vào nút giỏ hàng trên màn hình chính. Hệ thống lấy thông tin các bảng “products”, “cart\_details”, “carts” trong cơ sở dữ liệu và hiển thị ra màn hình thông tin sản phẩm trong giỏ hàng gồm: name, image, price, price, quantity.

Xóa sản phẩm:

Khách hàng chọn vào biểu tượng thùng rác của một sản phẩm trong giỏ hàng. Hệ thống sẽ hiển thị ra màn hình xác nhận xóa “Bạn có muốn xóa sản phẩm này ra khỏi cửa hàng”. Khi khách hàng ấn “Xóa”. Hệ thống sẽ xóa sản phẩm đó khỏi bảng “carts”, “cart\_details” và cập nhật lại số lượng trong bảng “product”.

Cập nhật số lượng sản phẩm:

Khách hàng thay đổi số lượng của một sản phẩm trong giỏ hàng. Hệ thống cập nhật lại số lượng trong bảng “product”. Use case kết thúc.

* Luồng rẽ nhánh:

Tại bước 1 trong luồng cơ bản nếu trong bảng “carts”, “cart\_details” đã hết sản phẩm, hoặc không có sản phẩm nào trong giỏ hàng thì hệ thống sẽ xóa toàn bộ thông tin trong giỏ hàng, hiển thị “Giỏ hàng trống”.

Tại bước 2 trong luồng cơ bản nếu khách hàng ấn “Không” hệ thống sẽ bỏ qua thao tác xóa.

Tại bất kỳ bước nào trong luồng cơ bản 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 đã đăng nhập thành công

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về giỏ hàng sẽ được cập nhập trong cơ sở dữ liệu.

**Điểm mở rộng:** Thanh toán, tiếp tục mua hàng

***Mô tả use case theo dõi đơn hàng***

**Tên Use case:** Theo dõi đơn hàng.

**Mô tả vắn tắt:** Use case này cho phép khách hàng xem chi tiết đơn đã đặt hàng và hủy đơn hàng.

**Luồng sự kiện:**

* Luồng cơ bản:

Ca sử dụng này bắt đầu khi khách hàng chọn vào mục “Đơn mua” trên menu. Hệ thống lấy các thông tin (address, phone, status, amount, order\_date ) từ bảng “orders” hiển thị ra màn hình.

Khách hàng chọn vào một đơn đã đặt. Hệ thống lấy thông tin chi tiết của hóa đơn từ bảng “order\_details”, “products”, “users” gồm các thông tin: name\_product, image, quantity, price, amount, name\_user, address, phone và hiển thị ra màn hình.

Khách hàng chọn vào biểu tượng “x” (hủy đơn hàng). Hệ thống hiển thị thông báo “Bạn có muốn hủy đơn hàng này”.

Khách hàng kích chọn “Hủy”. Hệ thống sẽ cập nhật lại trạng thái của đơn hàng trong bảng “Orders”. Use case kết thúc.

* Luồng rẽ nhánh:

Tại bước 1 trong luồng cơ bản nếu khách hàng chưa có đơn hàng nào đã đặt thì hệ thống sẽ hiển thị lên màn hình thông báo “Bạn chưa có đơn hàng nào”.

Tại bước 4 trong luồng cơ bản nếu khách hàng kích chọn “Không”. Hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các danh mục từ bảng “Orders”.

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

**Các yêu cầu đặc biệt:** Không có

**Tiền điều kiện:** Khách hàng đã đăng nhập và đã đặt hàng thành công.

**Hậu điều kiện:** Nếu khách hàng chọn hủy đơn hàng, hệ thống sẽ cập nhật lại trạng thái của đơn hàng.

**Điểm mở rộng:** Đánh giá

***Mô tả use case thanh toán đơn hàng***

**Tên Use case:** Thanh toán đơn hàng

**Mô tả vắn tắt**: Use case này cho phép khách hàng thanh toán các mặt hàng muốn mua có trong giỏ hàng.

**Luồng sự kiện:**

* Luồng cơ bản:

Use case này bắt đầu khi người dùng chọn vào giỏ hàng trên menu chính. Hệ thống sẽ truy cập và bảng “Products” lấy ra thông tin các sản phẩm đã chọn sau đó hiển thị ra màn hình chi tiết thông tin đơn hàng từ bảng “Orders” gồm: name, price, amount, image.

Người dùng chọn sản phẩm bất kì trong giỏ hàng và chọn vào nút “Thanh toán” bên dưới. Hệ thống lấy thông tin bảng “Users” hiển thị form yêu cầu khách hàng nhập thông tin gồm: name, address, phone, note để hoàn thành đơn hàng.

Người dùng nhấn chọn “Xác nhận thanh toán”. Hệ thống kiểm tra lại thông tin đơn hàng và lưu thông tin vào bảng Orders. Sau đó hiển thị thông báo “Đặt hàng thành công”. Use case kết thúc.

* Luồng rẽ nhánh:

Tại bước 2 trong luồng cơ bản, khi khách hàng nhập thông tin thanh toán không chính xác thì hệ thống hiển thị thông báo lỗi và quay lại bước 1 trong luồng cơ bản.

Tại bước 3 trong luồng cơ bản, khi khách hàng chọn “Không” thì hệ thống sẽ quay lại bước 1 để khách hàng thực hiện giao dịch mới.

Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được 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 và chọn sản phẩm vào giỏ hàng

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về đơn hàng sẽ được cập nhập trong cơ sở dữ liệu.

**Điểm mở rộng:** Xem đơn đặt hang

**Mô tả use case liên hệ**

**Tên Use case:** Liên hệ.

**Mô tả vắn tắt:** Use case này cho phép người dùng gửi liên hệ cho cửa hàng.

**Luồng sự kiện:**

* Luồng cơ bản:

Use case này bắt đầu khi người dùng click vào “Liên hệ” trên menu chính. Hệ thống sẽ hiển thị form liên hệ

Khách hàng nhập thông tin liên hệ (tên, email, tiêu đề, nội dung) và nhấn nút “Gửi”. Hệ thống lưu thông tin liên hệ vào bảng “contact” và hiển thị thông báo “Gửi 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 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

***Mô tả use case quản lý đơn hàng (Admin)***

**Tên Use case:** Quản lý đơn hàng người quản trị.

**Mô tả vắn tắt:** Use case này cho phép người quản trị tìm kiếm, xem, hủy đơn, xác nhận đơn và xác nhận đã thanh toán đơn hàng.

**Luồng các sự kiện**

* Luồng cơ bản

Use case này bắt đầu khi người quản trị chọn vào nút “Đơn hàng” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các đơn hàng gồm: order\_id, address, amount, order\_date, phone, status từ bảng “orders” trong cơ sở dữ liệu và hiển thị danh sách các đơn hàng lên màn hình.

Tìm kiếm đơn hàng:

Người quản trị nhập đơn hàng cần tìm trên cửa sổ danh sách các đơn hàng. Hệ thống lấy thông tin đơn hàng trong bảng “orders” có liên quan đến từ khóa và hiển thị ra màn hình.

Khi Admin click vào nút biểu tượng kiểm tra trên một dòng của đơn hàng. Hệ thống sẽ hiển thị ra thông tin chi tiết đơn hàng (price, quantity) từ bảng “order\_details” và (price, name, image) từ bảng “products” và (user\_name, address, phone) từ bảng “users” ra màn hình.

Xác nhận đơn hàng:

Khi Admin click vào nút “Xác nhận đơn”. Hệ thống sẽ hiển thị form yêu cầu xác nhận duyệt đơn hàng “Bạn muốn xác nhận đơn hàng này ?”.

Admin chọn vào nút “Xác nhận”. Hệ thống sẽ cập nhật lại trạng thái của đơn hàng được chọn trong bảng “orders” và hiển thị danh sách các đơn hàng.

Hủy đơn hàng:

Khi Admin click vào nút “Hủy đơn”. Hệ thống sẽ hiển thị form yêu cầu xác nhận hủy đơn hàng “Bạn muốn hủy đơn hàng này ?”.

Admin chọn vào nút “Hủy”. Hệ thống sẽ cập nhật lại trạng thái của đơn hàng được chọn trong bảng “orders” và hiển thị danh sách các đơn hàng.

Đã thanh toán:

Khi Admin click vào nút “Đã thanh toán”. Hệ thống sẽ hiển thị form yêu cầu xác nhận thanh toán đơn hàng “Bạn muốn xác nhận thanh toán đơn hàng này ?”.

Admin chọn vào nút “Xác nhận”. Hệ thống sẽ cập nhật lại trạng thái của đơn hàng được chọn trong bảng “orders” và hiển thị danh sách các đơn hàng. Use case kết thúc.

* Các luồng rẽ nhánh

Tại bước 4b, 5b, 6b trong luồng cơ bản nếu người quản trị chọn vào nút “Không” hệ thống sẽ bỏ qua thao tác Hủy đơn hoặc xác nhận đơn và hiển thị danh sách các đơn hàng từ bảng “orders”.

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về đơn hàng sẽ được cập nhập trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có

***Mô tả use case quản lý sản phẩm (Admin)***

**Tên Use case:** Quản lý sản phẩm.

**Mô tả vắn tắt:** Use case này cho phép người quản trị tìm kiếm, xem, thêm, sửa và xóa các danh mục.

**Luồng các sự kiện**

* Luồng cơ bản

Use case này bắt đầu khi người quản trị chọn vào nút “Sản phẩm” trên menu quản trị. Hệ thống lấy thông tin category\_name từ bảng “categorises” và các thông tin: product\_id, product\_name, price, discount, entered\_date, image từ bảng “products” trong cơ sở dữ liệu và hiển thị danh sách các sản phẩm lên màn hình.

Tìm kiếm sản phẩm:

Người quản trị nhập sản phẩm cần tìm trên cửa sổ danh sách các sản phẩm. Hệ thống lấy thông tin sản phẩm trong bảng “products” có liên quan đến từ khóa và hiển thị ra màn hình.

Thêm sản phẩm:

Người quản trị chọn vào nút “Thêm sản phẩm” trên cửa sổ danh sách các sản phẩm. Hệ thống lấy category\_name từ bảng “categories” và hiển thị màn hình thêm mới sản phẩm.

Người quản trị nhập thông tin sản phẩm mới (name, price, discout, image, category\_name, discription, branch, material) và chọn vào nút “Thêm”. Hệ thống sinh mã sản phẩm mới, lưu thông tin (name, price, discout, image, category\_name, discription, branch, material) vào bảng sản phẩm, các ảnh của sản phẩm vào bảng “image” và hiển thị danh sách sản phẩm đã được cập nhật. Use case kết thúc.

Sửa sản phẩm:

Người quản trị chọn vào biểu tượng “Sửa” trên một dòng danh mục. Hệ thống sẽ lấy thông tin cũ của sản phẩm được chọn gồm: tên sản phẩm, số lượng, đơn giá, thương hiệu, chất liệu, ảnh, khuyến mại, loại hàng, mô tả từ bảng “products”, “image”, “categories” và hiển thị lên màn hình.

Người quản trị nhập thông tin mới cho tên sản phẩm và chọn vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của sản phẩm được chọn trong bảng “products”, “image” và hiển thị danh sách sản phẩm đã được cập nhật.

Xóa sản phẩm

Người quản trị chọn vào biểu tượng thùng rác trên một dòng sản phẩm. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa “Bạn có muốn xóa sản phẩm…?”.

Người quản trị chọn vào nút “Xóa”. Hệ thống sẽ xóa sản phẩm được chọn khỏi bảng “products”, “image” và hiển thị danh sách các sản phẩm mà cửa hàng đã cập nhật. Use case kết thúc.

* Các luồng rẽ nhánh

Tại bước 3b hoặc 4b trong luồng cơ bản nếu người quản trị nhập thông tin danh mục không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc thoát ra ngoài để kết thúc.

Tại bước 5b trong luồng cơ bản nếu người quản trị chọn vào nút “Không” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các sản phẩm từ bảng “products”.

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về sản phẩm sẽ được cập nhập trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có

***Mô tả use case quản lý danh mục (Admin)***

**Tên Use case:** Quản lý danh mục.

**Mô tả vắn tắt:** Use case này cho phép người quản trị tìm kiếm, xem, thêm, sửa và xóa các danh mục.

**Luồng các sự kiện**

* Luồng cơ bản

Use case này bắt đầu khi người quản trị chọn vào nút “Danh mục sản phẩm” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các danh mục gồm: category\_id, category\_name từ bảng “categories” trong cơ sở dữ liệu và hiển thị danh sách các danh mục lên màn hình.

Tìm kiếm danh mục:

Người quản trị nhập danh mục cần tìm trên cửa sổ danh sách các danh mục. Hệ thống lấy thông tin danh mục trong bảng “categories” có liên quan đến từ khóa và hiển thị ra màn hình.

Thêm loại danh mục:

Người quản trị chọn vào nút “Thêm danh mục” trên cửa sổ danh sách các danh mục. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho danh mục: tên danh mục.

Người quản trị nhập thông tin của tên danh mục, tự sinh một mã danh mục mới và chọn vào nút “Thêm”. Hệ thống sẽ tạo một danh mục mới trong bảng “categories” và hiển thị danh sách các danh mục đã được thêm mới.

Sửa danh mục:

Người quản trị chọn vào nút “Sửa” trên một dòng danh mục. Hệ thống sẽ lấy thông tin cũ của danh mục được chọn gồm: tên danh mục từ bảng “categories” và hiển thị lên màn hình.

Người quản trị nhập thông tin mới cho tên danh mục và chọn vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của danh mục được chọn trong bảng “categories” và hiển thị danh sách các danh mục mà cửa hàng đã cập nhật.

Xóa danh mục:

Người quản trị chọn vào biểu tượng thùng rác trên một dòng danh mục. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa “Bạn có muốn xóa danh mục có tên…?”.

Người quản trị chọn vào nút “Xóa”. Hệ thống sẽ xóa danh mục được chọn khỏi bảng “categories” và hiển thị danh sách các danh mục mà cửa hàng đã cập nhật. Use case kết thúc.

* Các luồng rẽ nhánh

Tại bước 3b hoặc 4b trong luồng cơ bản nếu người quản trị nhập thông tin danh mục không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc thoát ra ngoài để kết thúc.

Tại bước 5b trong luồng cơ bản nếu người quản trị chọn vào nút “Không” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các danh mục từ bảng “categories”.

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về danh mục sẽ được cập nhập trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có.

***Mô tả use case quản lý khách hàng (Admin)***

**Tên Use case:** Quản lý khách hàng.

**Mô tả vắn tắt:** Use case này cho phép người quản trị tìm kiếm, xem, thêm, xóa các khách hàng.

**Luồng các sự kiện**

* Luồng cơ bản

Use case này bắt đầu khi người quản trị chọn vào nút “Khách hàng” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các khách hàng gồm: user\_id, address, image, email, phone, gender, register\_date từ bảng “users” trong cơ sở dữ liệu và hiển thị danh sách các khách hàng lên màn hình.

Tìm kiếm khách hàng:

Người quản trị nhập khách hàng cần tìm trên cửa sổ danh sách các khách hàng. Hệ thống lấy thông tin khách hàng trong bảng “users” có liên quan đến từ khóa và hiển thị ra màn hình.

Thêm khách hàng:

Người quản trị chọn vào nút “Thêm khách hàng” trên cửa sổ danh sách các khách hàng. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho khách hàng: user\_id, address, image, email, phone, gender, register\_date.

Người quản trị nhập thông tin của khách hàng và chọn vào nút “Thêm”. Hệ thống sẽ tạo một khách hàng mới trong bảng “users” và hiển thị danh sách các khách hàng đã được cập nhật. Use case kết thúc.

Xóa khách hàng

Người quản trị chọn vào biểu tượng thùng rác trên một dòng khách hàng. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa “Bạn có muốn xóa khách hàng có tên…?”.

Người quản trị chọn vào nút “Xóa”. Hệ thống sẽ xóa khách hàng được chọn khỏi bảng “users” và hiển thị danh sách các khách hàng đã cập nhật.

* Các luồng rẽ nhánh

Tại bước 3b trong luồng cơ bản nếu người quản trị nhập thông tin khách hàng không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc thoát ra ngoài để kết thúc.

Tại bước 4b trong luồng cơ bản nếu người quản trị chọn vào nút “Không” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các khách hàng từ bảng “users”.

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về khách hàng sẽ được cập nhập trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có

***Mô tả use case quản lý liên hệ (Admin)***

**Tên Use case:** Quản lý liên hệ.

**Mô tả vắn tắt:** Use case này cho phép người quản trị tìm kiếm, xem, phản hồi và xóa liên hệ.

**Luồng các sự kiện**

* Luồng cơ bản

Use case này bắt đầu khi người quản trị chọn vào nút “Liên hệ của khách hàng” trên menu quản trị. Hệ thống lấy thông tin chi tiết của các liên hệ gồm: id, content, name, date, email, title từ bảng “contact” trong cơ sở dữ liệu và hiển thị danh sách các liên hệ lên màn hình.

Tìm kiếm liên hệ:

Người quản trị nhập từ khóa cần tìm trên cửa sổ danh sách các liên hệ. Hệ thống lấy thông tin liên hệ trong bảng “contact” có liên quan đến từ khóa và hiển thị ra màn hình.

Xóa liên hệ:

Người quản trị chọn vào biểu tượng thùng rác trên một dòng liên hệ. Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa “Bạn có muốn xóa liên hệ này không?”.

Người quản trị chọn vào nút “Xóa”. Hệ thống sẽ xóa liên hệ được chọn khỏi bảng “contact” và hiển thị danh sách các liên hệ mà cửa hàng đã cập nhật.

Phản hồi liên hệ:

Người quản trị chọn vào nút “Gửi” trên một dòng liên hệ. Hệ thống hiển thị ra cửa sổ gửi email. Use case kết thúc.

* Các luồng rẽ nhánh

Tại bước 3b trong luồng cơ bản nếu người quản trị chọn vào nút “Không” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các liên hệ từ bảng “contact”.

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì thông tin về liên hệ sẽ được cập nhập trong cơ sở dữ liệu.

**Điểm mở rộng:** Không có

***Thống kê sản phẩm (Admin)***

**Tên Use case:** Thống kê các sản phẩm.

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem những sản phẩm bán chạy nhất, sản phẩm tồn kho.

**Luồng các sự kiện**

* Luồng cơ bản

Use case này bắt đầu khi người quản trị chọn vào nút “Sản phẩm bán chạy” trên menu quản trị. Hệ thống lấy thông tin của các sản phẩm bán chạy gồm: hình ảnh sản phẩm, mã sản phẩm, tên sản phẩm, số lượng bán được, danh mục của sản phẩm của các sản phẩm có lượt mua nhiều nhất và tự động lập biểu đồ thống kê 10 sản phẩm bán chạy nhất.

Use case này bắt đầu khi người quản trị chọn vào nút “Hàng tồn kho” trên menu quản trị. Hệ thống lấy thông tin của các sản phẩm bán chạy gồm: hình ảnh sản phẩm, mã sản phẩm, tên sản phẩm, số lượng bán được, danh mục của sản phẩm của các sản phẩm có lượt mua ít nhất và tự động lập biểu đồ thống kê 10 sản phẩm có lượt mua thấp nhất.

* Các luồng rẽ nhánh

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

**Các yêu cầu đặc biệt:** Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

**Tiền điều kiện:** Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

**Hậu điều kiện:** Nếu use case kết thúc thành công thì sẽ hiển thị biểu đồ thống kê.

**Điểm mở rộng:** Không có.

**Biểu đồ các use case**

***Use case đăng nhập***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case đăng nhập:

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.6: Biểu đồ VOPC đăng nhập

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.7: Biểu đồ trình tự use case đăng nhập*

***Use case đăng ký***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case đăng ký:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.8: Biểu đồ VOPC use case đăng ký*

**A diagram of a diagram

AI-generated content may be incorrect.**

*Hình 2.9: Biểu đồ trình tự use case đăng ký*

***Use case xem sản phẩm theo danh mục***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case xem sản phẩm theo danh mục:

A screenshot of a computer code

AI-generated content may be incorrect.

*Hình 2.10: Biểu đồ VOPC use case xem sản phẩm theo danh mục*

**A diagram with text and words

AI-generated content may be incorrect.**

*Hình 2.11: Biểu đồ trình tự xem sản phẩm theo danh mục*

***Use case xem chi tiết một sản phẩm***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case xem chi tiết một sản phẩm:

A diagram of a computer

AI-generated content may be incorrect.

*Hình 2.12: Biểu đồ VOPC xem chi tiết một sản phẩm*

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.13: Biểu đồ trình tự use case xem chi tiết một sản phẩm*

***Use case tìm kiếm sản phẩm***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case tìm kiểm sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.14: Biểu đồ VOPC use case tìm kiếm sản phẩm*

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.15: Biểu đồ trình tự use case tìm kiếm sản phẩm*

***Use case quản lý giỏ hàng***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case quản lý giỏ hàng:

A diagram of a person with text

AI-generated content may be incorrect.

*Hình 2.16: Biểu đồ VOPC use case quản lý giỏ hàng*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.17: Biểu đồ trình tự use case quản lý giỏ hàng*

***Use case theo dõi đơn hàng***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case theo dõi đơn hàng:

A screenshot of a diagram

AI-generated content may be incorrect.

*Hình 2.18: Biểu đồ use case theo dõi đơn hàng*

A diagram of a project

AI-generated content may be incorrect.

*Hình 2.19: Biểu đồ trình tự use case theo dõi đơn hàng*

***Use case thanh toán đơn hàng***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case thanh toán:

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.20: Biểu đồ VOPC use case thanh toán*

A diagram of a project

AI-generated content may be incorrect.

*Hình 2.21: Biểu đồ trình tự use case thanh toán*

***Use case liên hệ***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case liên hệ:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.22: Biểu đồ VOPC use case liên hệ*

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.23: Biểu đồ use case liên hệ*

***Use case quản lý đơn hàng (admin)***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case quản lý đơn hàng:

***A screenshot of a computer code

AI-generated content may be incorrect.***

*Hình 2.24: Biểu đồ VOPC use case quản lý đơn hàng*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.25: Biểu đồ trình tự use case quản lý đơn hàng*

***Use case quản lý sản phẩm (Admin)***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case quản lý sản phẩm:

A screenshot of a computer code

AI-generated content may be incorrect.

*Hình 2.26: Biểu đồ VOPC quản lý sản phẩm*

A screenshot of a computer screen

AI-generated content may be incorrect.

*Hình 2.27: Biểu đồ trình tự use case quản lý sản phẩm*

**Use case quản lý danh mục (Admin)**

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case quản lý danh mục:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.28: Biểu đồ VOPC use case quản lý danh mục*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.29: Use case quản lý danh mục*

***Use case quản lý khách hàng (Admin)***

Hình dưới minh họa biểu đồ VOPC và biểu đồ trình tự use case quản lý khách hàng:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.30: Biểu đồ VOPC quản lý khách hàng*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.31: Biểu đồ trình tự use case quản lý khách hàng*

***Use case quản lý liên hệ (Admin)***

Hình minh họa biểu đồ VOPC và biểu đồ trình tự use case quản lý liên hệ:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2.32: Biểu đồ VOPC quản lý liên hệ*

*A diagram of a project

AI-generated content may be incorrect.*

*Hình 2.33: Biểu đồ trình tự use case quản lý liên hệ*

## 2.4 Thiết kế cơ sở dữ liệu

**Sơ đồ thực thể liên kết**

Hình dưới minh họa sơ đồ thực thể liên kết:

A computer screen shot of a computer

AI-generated content may be incorrect.

*Hình 2.34: Sơ đồ thực thể liên kết*

**Cấu trúc các bảng**

**Notification (Thông báo đơn đặt hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | bigint | Primary key | Mã thông báo |
| message | varchar(255) |  | Nội dung thông báo |
| status | bit(1) |  | Trạng thái thông báo  1: đã xem  0: chưa xem |
| time | datetime |  | Thời gian thông báo |

*Bảng 2.2: Cấu trúc bảng notification*

**Roles (Phân quyền)**

*Bảng 2.3: Cấu trúc bảng roles*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int | Primary key | Mã quyền |
| name | varchar(20) |  | Tên quyền |

**User\_roles (Chi tiết phân quyền)**

*Bảng 2.4: Cấu trúc bảng user\_roles*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| user\_id | bigint | Primary key | Mã người dùng |
| role\_id | int | Primary key | Mã quyền người dùng |

**Users (Tài khoản)**

*Bảng 2.5: Cấu trúc bảng users*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| user\_id | bigint | Primary key | Mã người dùng |
| address | varchar(255) |  | Địa chỉ người dùng |
| email | varchar(255) |  | Email người dùng |
| gender | bit(1) |  | Giới tính người dùng  1: nam  0: nữ |
| image | varchar(255) |  | Ảnh đại diện |
| name | varchar(255) |  | Tên người dùng |
| password | varchar(255) |  | Mật khẩu |
| phone | varchar(255) |  | Số điện thoại |
| register\_date | date |  | Ngày đăng kí |
| status | bit(1) |  | Trạng thái tài khoản |
| token | varchar(255) |  | Mã token đăng nhập |

**Categories (Danh mục sản phẩm)**

*Bảng 2.6: Cấu trúc bảng categories*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| category\_id | bigint | Primary key | Mã danh mục sản phẩm |
| category\_name | varchar(255) |  | Tên danh mục sản phẩm |

**Products (Sản phẩm)**

*Bảng 2.7: Cấu trúc bảng products*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| product\_id | bigint | Primary key | Mã sản phẩm |
| escription | varchar(1000) |  | Mô tả sản phẩm |
| discount | int |  | Giảm giá |
| entered\_date | date |  | Ngày nhập sản phẩm |
| image | varchar(255) |  | Url ảnh sản phẩm |
| name | varchar(255) |  | Tên sản phẩm |
| price | double |  | Giá tiền |
| quantity | int |  | Số lượng |
| sold | int |  | Số lượng đã bán ra |
| status | bit(1) |  | Trạng thái  1: còn sản phẩm  0: hết sản phẩm |
| category\_id | bigint | Foreign key | Mã danh mục sản phẩm |
| size | varchar(255) |  | Size sản phẩm |

**Rates (Đánh giá sản phẩm)**

*Bảng 2.8: Cấu trúc bảng rates*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | bigint | Primary key | Mã đánh giá |
| comment | varchar(255) |  | Nội dung đánh giá |
| rate\_date | datetime |  | Ngày đánh giá |
| rating | double |  | Số sao đánh giá (1-5 sao) |
| order\_detail\_id | bigint | Foreign key | Mã đơn hàng |
| product\_id | bigint | Foreign key | Mã sản phẩm |
| user\_id | bigint | Foreign key | Mã người dùng |

**Favorites (Yêu thích)**

*Bảng 2.9: Cấu trúc bảng favotites*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| favorite\_id | bigint | Primary key | Mã yêu thích |
| product\_id | bigint | Foreign key | Mã sản phẩm |
| user\_id | bigint | Foreign key | Mã người dùng |

**Carts (Giỏ hàng)**

*Bảng 2.10: Cấu trúc bảng carts*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| cart\_id | bigint | Primary key | Mã giỏ hàng |
| address | varchar(255) |  | Địa chỉ giỏ hàng |
| phone | varchar(255) |  | Số điện thoại |
| user\_id | bigint | Foreign key | Mã người dùng |

**Order\_details (Chi tiết giỏ hàng)**

*Bảng 2.11: Cấu trúc bảng order\_details*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| cart\_detail\_id | bigint | Primary key | Mã chi tiết giỏ hàng |
| price | double |  | Giá |
| quantity | int |  | Số lượng |
| cart\_id | bigint | Foreign key | Mã giỏ hàng |
| product\_id | bigint | Foreign key | Mã sản phẩm |
| size | varchar(45) |  | Kích thước sản phẩm |

**Orders (Đơn hàng)**

*Bảng 2.12: Cấu trúc bảng orders*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| order\_id | bigint | primary key | Mã đơn hàng |
| address | varchar(255) |  | Địa chỉ đơn hàng |
| amount | double |  | Tổng tiền |
| order\_date | datetime |  | Ngày đặt hàng |
| phone | varchar(255) |  | Số điện thoại |
| status | int |  | Trạng thái  0: Chưa xác nhận đơn hàng  1: Đã xác nhận đơn hàng  2: Đã giao hàng thành công  3: Đã hủy đơn hàng |
| user\_id | bigint | foreign key | Mã người dùng |

**Order\_details (Chi tiết đơn hàng)**

*Bảng 2.13: Cấu trúc bảng order\_details*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| order\_detail\_id | bigint | Primary key | Mã chi tiết đơn hàng |
| price | double |  | Giá |
| quantity | int |  | Số lượng |
| order\_id | bigint | Foreign key | Mã đơn hàng |
| product\_id | bigint | Foreign key | Mã sản phẩm |
| size | varchar(45) |  | Kích thước sản phẩm |

**Contact (Liên hệ)**

*Bảng 2.14: Cấu trúc bảng contact*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int | Primary key | Mã liên hệ |
| email | varchar(45) |  | Email |
| title | varchar(255) |  | Tiêu đề |
| content | varchar(1000) |  | Nội dung |
| date | date |  | Ngày gửi |

**Images (Ảnh sản phẩm)**

*Bảng 2.15: Cấu trúc bảng images*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | int | Primary key | Mã ảnh |
| product\_id | bigint | Foreign key | Mã sản phẩm |
| file | varchar(255) |  | Đường dẫn ảnh |

# CHƯƠNG 3: THỬ NGHIỆM VÀ TRIỂN KHAI

# HỆ THỐNG

## 3.1 Công cụ và môi trường được sử dụng

### 3.1.1 Các phần mềm sử dụng.

* IntelliJ IDEA: Một môi trường phát triển tích hợp (IDE) cho Java, được sử dụng để phát triển ứng dụng Java và các dự án phát triển phần mềm khác.
* MySQL Workbench: Hệ quản trị cơ sở dữ liệu phổ biến, được sử dụng để lưu trữ và quản lý dữ liệu.
* Visual Studio Code: Một trình soạn thảo mã nguồn mở, nhẹ và mạnh mẽ được phát triển bởi Microsoft, thích hợp cho nhiều ngôn ngữ lập trình và kịch bản khác nhau.
* Node.js (phiên bản v16.13.2): Một nền tảng phát triển mã nguồn mở, được xây dựng trên Chrome's V8 JavaScript runtime, cho phép thực thi mã JavaScript nền tảng server-side

### 3.1.2 Cài đặt phần mềm và môi trường

* Bước 1: Cài đặt các phần mềm IntelliJ IDEA, MySQL và Visual Studio Code theo hướng dẫn từ trang web chính thức của mỗi phần mềm.
* Bước 2: Tải và cài đặt Node.js phiên bản v16.13.2 từ trang web chính thức của Node.js.
* Bước 3: Mở project với IntelliJ IDEA và mở project shop-sport-backend, sau đó chạy file start.
* Bước 4: Mở MySQL và chạy file bluesky.sql để thiết lập cơ sở dữ liệu.
* Bước 5: Mở command prompt và cài đặt Angular CLI bằng cách nhập lệnh:

npm install -g @angular/cli@12.0.5

* Bước 6: Sau khi cài đặt xong Angular CLI, mở project shop-sport-fe-admin và shop-sport-fe-customer bằng Visual Studio Code và chạy lệnh npm start.
* Bước 7: Sau khi chạy thành công, truy cập vào các địa chỉ URL sau:

<http://localhost:4200/> (dành cho shop-sport-fe- customer)

<http://localhost:4201/> (dành cho shop-sport-fe-admin)

## 3.2 Kiểm thử chức năng hệ thống

### 3.2.1 Môi trường và phương pháp kiểm thử

*Thiết lập môi trường kiểm thử:*

* Cấu hình cơ sở dữ liệu: MySQL Workbench (Local).
* Cấu hình máy chủ: AMD RYZEN 5 5600H ~3.3GHz, RAM 8GB-512GB. (Local)
* Hệ điều hành máy chủ: Window 11.
* Công cụ kiểm thử: POSTMAN, SWAGGER

*Phương pháp kiểm thử API:*

* Functionality testing: Xác nhận API hoạt động chính xác theo đúng chức năng mà nó được tạo ra.
* Usability testing: Xác nhận API có thể sử dụng một cách dễ dàng
* Reliability testing: Xác nhận việc gọi API và trả kết quả hoạt động ổn định và nhất quán.

### 3.2.2 Tiến hành kiểm thử

*Bảng 3.1: Kiểm thử API đăng nhập*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Trường hợp test** | **Kết quả mong đợi** | **Kết quả**  **thực tế** |
| TC\_01 | Username: không nhập  Password: không nhập | Đăng nhập thất bại | PASS |
| TC\_02 | Username: có nhập  Password: không nhập | Đăng nhập thất bại | PASS |
| TC\_03 | Username: không nhập   Password: có nhập | Đăng nhập thất bại | PASS |
| TC\_04 | Username: có nhập (đúng)  Password: có nhập (sai) | Đăng nhập thất bại | PASS |
| TC\_05 | Username: có nhập (sai)  Password: có nhập (đúng) | Đăng nhập thất bại | PASS |
| TC\_06 | Username: có nhập (đúng)  Pass: có nhập (đúng) | Đăng nhập thành công | PASS |

*Bảng 3.2: Kiểm thử API đăng ký*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Trường hợp test** | **Kết quả mong đợi** | **Kết quả**  **thực tế** |
| TC\_01 | Email: không  Password: không  OTP: không  User: không  Address: không  Phone: không | Đăng ký thất bại | PASS |
| TC\_02 | Không nhập đủ thông tin các trường | Đăng ký thất bại | PASS |
| TC\_03 | Mail: có nhập (đúng)  OTP: có nhập (sai) | Đăng ký thất bại | PASS |
| TC\_04 | Mail: có nhập (sai)  OTP: có nhập (sai) | Đăng ký thất bại | PASS |
| TC\_05 | Mail: có nhập (đúng)  OTP: có nhập (đúng)  Address: có nhập  Một trong 3 trường (name, phone, user): có nhập (sai) | Đăng ký thất bại | PASS |
| T06 | Mail: có nhập (đúng)  OTP: có nhập (đúng)  Address: có nhập  Pone: có nhập (đúng)  Name: có nhập (đúng)  User: có nhập (đúng) | Đăng ký thành công | PASS |

*Bảng 3.3: Kiểm thử API quên mật khẩu*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Trường hợp test** | **Kết quả mong đợi** | **Kết quả**  **thực tế** |
| TC\_01 | Email: không nhập | Email không hợp lệ! | PASS |
| TC\_02 | Email: có nhập (không tồn tại trong CSDL) | Email không hợp lệ! | PASS |
| TC\_03 | Email: có nhập (tồn tại trong CSDL) | Gửi mã về mail | PASS |

*Bảng 3.4: Kiểm thử API quên mật khẩu*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Trường hợp test** | **Kết quả mong đợi** | **Kếtquả thực tế** |
| TC\_01 | Password: không nhập  Confirm Password: không nhập | Mật khẩu phải trên 6 kí tự! | PASS |
| TC\_02 | Password: không nhập  Confirm Password: có nhập | Mật khẩu phải trên 6 kí tự! | PASS |
| TC\_03 | Password: có nhập (thiếu kí tự)  Confirm Password: có nhập (thiếu kí tự) | Mật khẩu phải trên 6 kí tự! | PASS |
| TC\_04 | Password: có nhập (đúng)  Confirm Password: có nhập (không khớp với password ở trên) | Mật khẩu không khớp | PASS |
| TC\_05 | Password: có nhập (đúng)  Confirm Password: có nhập (đúng) | Hiển thông báo quay lại trang đăng nhập | PASS |

*Bảng 3.5: Kiểm thử API tìm kiếm sản phẩm*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Trường hợp test** | **Kết quả mong đợi** | **Kết quả**  **thực tế** |
| TC\_01 | Không nhập | Hiển thị ra danh sách sản phẩm | PASS |
| TC\_02 | Nhập sai từ khóa | Hiển thị thông báo “Không tìm thấy sản phẩm có từ khóa...” | PASS |
| TC\_03 | Nhập đúng | Hiển thị sản phẩm liên quan đến từ khóa | PASS |

*Bảng 3.6: Kiểm thử API liên hệ*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Trường hợp test** | **Kết quả mong đợi** | **Kết quả**  **thực tế** |
| TC\_01 | Không nhập | Gửi thất bại. Hãy điền đầy đủ thông tin | PASS |
| TC\_02 | Một trong các trường không nhập | Gửi thất bại. Hãy điền đầy đủ thông tin | PASS |
| TC\_03 | Email: có nhập (sai)Title: có nhập (đúng)Name: có nhập (đúng)  Content: có nhập (đúng) | Gửi thất bại. Hãy điền đầy đủ thông tin | PASS |
| TC\_04 | Email: có nhập(đúng)Title: có nhập (đúng)  Name: có nhập (đúng)  Content: có nhập (đúng) | Gửi thành công | PASS |

*Bảng 3.7: Kiểm thử API thanh toán*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Trường hợp test** | **Kết quả mong đợi** | **Kết quả**  **thực tế** |
| TC\_01 | Một trong các trường không nhập | Hãy nhập đầy đủ thông tin | PASS |
| TC\_02 | Phone: có nhập (đúng)  district: có nhập  ward: có nhập  province: có nhập  note: không nhập  number: không nhập | Hiển thị cửa sổ “Bạn có muốn đặt đơn hàng này” | PASS |
| TC\_03 | Phone: có nhập (đúng)  district: có chọn  ward: có chọn  province: có chọn  note: có nhập  number: có nhập | Hiển thị cửa sổ “Bạn có muốn đặt đơn hàng này” | PASS |
| TC\_04 | Ấn “Đặt” | Hiển thị cửa sổ “Chúc mừng bạn đã đặt hàng thành công” | PASS |

### 3.2.3 Hình ảnh

**Kết quả kiểm thử**

Dưới đây là các hình ảnh minh họa kiểm thử API bằng công cụ postman:

Kiểm thử API đăng nhập:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.1: Kiểm thử API đăng nhập*

Kiểm thử API đăng ký:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.2: Kiểm thử API đăng ký*

Kiểm tra API xem sản phẩm theo danh mục:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.3: Kiểm thử API xem sản phẩm theo danh mục*

Kiểm thử API xem chi tiết một sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.4: Kiểm thử API xem chi tiết một sản phẩm*

Kiểm thử API theo dõi đơn hàng:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.6: Kiểm thử API theo dõi đơn hàng*

Kiểm thử API liên hệ:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.8: Kiểm thử API liên hệ*

Kiểm thử API xem tất cả khách hàng:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.10: Kiểm thử API xem tất cả khách hàng (Admin)*

Kiểm thử API xem tất cả sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.12: Kiểm thử API xem tất cả sản phẩm (Admin)*

**Giao diện hệ thống**

***Giao diện khách hàng***

Dưới đây là một số hình ảnh về giao diện website:

Giao diện trang chủ:

A screenshot of a website

AI-generated content may be incorrect.

*Hình 3.13: Giao diện trang chủ*

Giao diện đăng nhập và đăng ký:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.14: Giao diện đăng nhập khách hàng*

A screenshot of a chat

AI-generated content may be incorrect.

*Hình 3.15: Giao diện đăng ký tài khoản*

Giao diện theo dõi đơn hàng:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.16: Giao diện theo dõi đơn hàng*

Giao diện chỉnh sửa thông tin cá nhân:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.17: Giao diện chỉnh sửa thông tin cá nhân*

Giao diện liên hệ:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.18: Giao diện liên hệ với Admin*

Giao diện xem sản phẩm theo danh mục

A screenshot of a website

AI-generated content may be incorrect.

*Hình 3.19: Giao diện xem sản phẩm theo danh mục*

Giao diện giỏ hàng:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.20: Giao diện giỏ hàng*

Giao diện danh sách sản phẩm yêu thích:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.21: Giao diện danh sách sản phẩm yêu thích*

***Giao diện quản trị***

Giao diện quản lý khách hàng:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.23: Giao diện quản lý khách hàng*

Giao diện quản lý đơn hàng:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.24: Giao diện quản lý đơn hang*

Giao diện xem đánh giá:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.25: Giao diện xem đánh giá*

Giao diện quản lý danh mục sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.26: Giao diện quản lý danh mục*

Giao diện quản lý liên hệ:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.27: Giao diện quản lý liên hệ*

Giao diện thống kê sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.28: Giao diện thống kê sản phẩm*

Giao diện quản lý sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.29: Giao diện quản lý sản phẩm*

Giao diện thêm mới sản phẩm:

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 3.30: Giao diện thêm sản phẩm*

# KẾT LUẬN

Trong quá trình thực hiện đề tài “Xây dựng ứng dụng website để kinh doanh bán đồ thể thao cho cửa hàng Bluesky”, em đã thu được nhiều kiến thức quý báu cả về lý thuyết và ý nghĩa thực tiễn.

Về lý thuyết, em đã nắm vững và áp dụng sâu hơn kiến thức của Angular vào việc phát triển Front-end và Spring Boot Framework trong xử lý Back-end. Đồng thời, việc nắm bắt một số câu lệnh trong Git, sử dụng các công cụ quản lý database và Rational Rose để vẽ biểu đồ Use case, Biểu đồ lớp đã giúp hiểu rõ hơn về quy trình phát triển ứng dụng web.

Về ý nghĩa thực tiễn, em đã có cơ hội tiếp cận và học hỏi kiến thức từ những người có kinh nghiệm trong lĩnh vực phát triển phần mềm. Quá trình làm việc này đã giúp em phân chia công việc và phân bổ thời gian một cách hợp lý.

Kết quả đạt được:

* Xây dựng thành công ứng dụng web để kinh doanh bán đồ thể thao cho cửa hàng Bluesky.
* Học hỏi thêm được nhiều kiến thức và kỹ năng mới.
* Đã áp dụng được những kiến thức học được vào thực tế.
* Ứng dụng quy trình phát triển phần mềm vào dự án.

Đề xuất phát triển tiếp theo là để hệ thống “Xây dựng ứng dụng website để kinh doanh bán đồ thể thao cho cửa hàng Bluesky” có thể mang lại nhiều chức năng hơn, phù hợp với nhu cầu người tiêu dùng như:

* Tích hợp thanh toán trực tuyến.
* Hỗ trợ đa ngôn ngữ.
* Cải thiện UI/UX.
* Chương trình khách hàng thân thiết.

Hy vọng rằng hệ thống sẽ được sử dụng rộng rãi và mang đến những sản phẩm thông tin hữu ích đến người dùng.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | VŨ THỊ HƯƠNG TRÀ, HOÀNG THỊ KIM OANH (2023), Phát triển thương mại điện tử tại Việt Nam. |
| [2] | Aristeidis Bampakos (2022), Learning Angular. |
| [3] | Josephine Bush(2020), *Learn SQL Database Programming*, ACB Bookstore. |
| [4] | <https://stackoverflow.com/questions>. |
| [5] | <https://viblo.asia/questions>. |
| [6] | [https://topdev.vn](https://topdev.vn/). |
| [7] | [https://spring.io](https://spring.io/). |