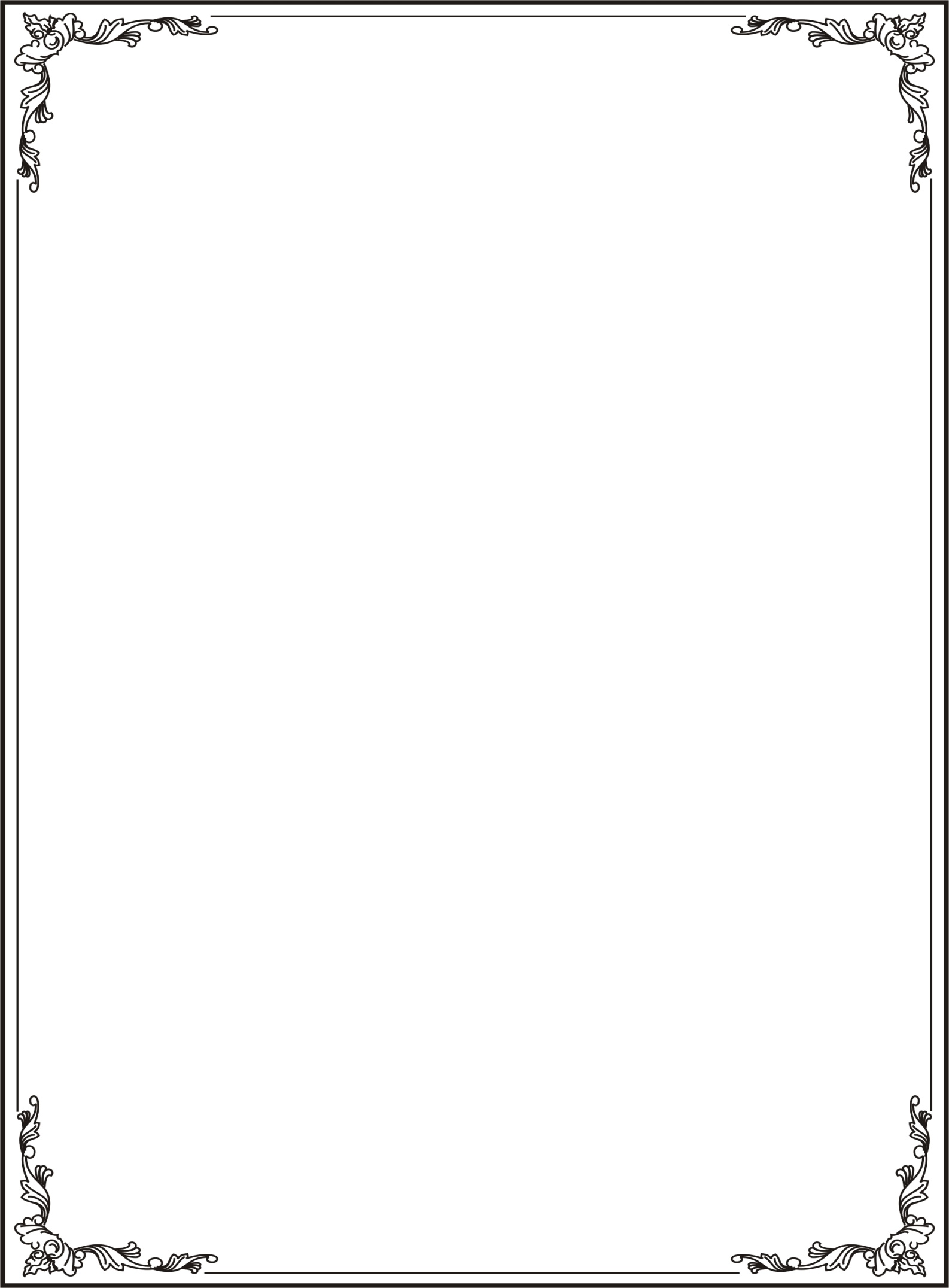
** TRƯỜNG CAO ĐẲNG THỰC HÀNH FPT POLYTECHNIC**

**XÂY DỰNG WEBSITE THỜI TRANG ONLINE**

**SUNRISE SHOP**



|  |  |
| --- | --- |
| **Giáo viên hướng dẫn** | **: Vũ Thị Thanh Huyền** |
| **Lớp** | **: PT14301** |
| **Tên nhóm** | **: Nhóm Sunrise** |
| **Sinh viên thực hiện** |  |
| 1. Nguyễn Ngô Anh Tuấn | : Trưởng nhóm |
| 1. Trần Thị Quỳnh | : Thành viên |
| 1. Trương Nguyễn Tiểu Long | : Thành viên |
| 1. Nguyễn Đức Tuấn Anh | : Thành viên |
| 1. Phạm Hồng Đạt | : Thành viên |

*Đà Nẵng, năm 2020*

**LỜI NÓI ĐẦU**

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ. Công nghệ Thông tin là một trong những ngành có vị thế dẫn đầu và có vai trò rất lớn trong sự phát triển chung. Các ứng dụng của công nghệ thông tin được áp dụng trong mọi lĩnh vực nghiên cứu khoa học cũng như trong mọi lĩnh vực của đời sống. Là một phần của Công nghệ Thông tin, Công nghệ web đang có được sự phát triển mạnh mẽ và phổ biến rất nhanh bởi những lợi ích mà nó mang lại cho cộng đồng là rất lớn.

Bằng việc lựa chọn và thực hiện đề tài “***Xây dựng website thời trang online – Sunrise Shop”***, nhóm em muốn xây dựng website có tính hữu dụng cao và đưa ra một giải pháp tốt nhằm giải quyết công việc bán hàng trực tuyến cho công ty. Website mang lại cho công ty rất nhiều lợi ich như: khả năng quảng cáo, phổ biến tên tuổi công ty rất nhanh trên môi trường mạng toàn cầu, việc kinh doanh sẽ mở 24/24, giảm thiểu chi phí nhân viên cho tiếp thị, quảng cáo, dễ dàng nhận phải hồi từ phía khách hàng, cơ hội mở rộng liên kết và hợp tác ở phạm vi quốc tế...v..v.

Website bán hàng giúp cho công ty cung cấp cho khách hàng những lựa chọn linh hoạt và tiện lợi trong việc tìm mua sản phẩm thông qua chức năng tìm kiếm và giỏ hàng. Các thông tin về sản phẩm được hiển thị chi tiết với từng sản phẩm, từ đó khách hàng dễ dàng nhận biết và lựa chọn được sản phẩm mong muốn. Về phía Công ty, hệ thống mang lại sự tiện dụng trong việc cập nhập và quản lý các thông tin cho website, tạo khả năng liên lạc nhanh và dễ dàng với khách hàng.

# LỜI CẢM ƠN

Trước tiên em xin gửi lời cảm ơn chân thành sâu sắc tới các thầy cô giáo trong trường Cao đẳng thực hành FPT-Polytechnic nói chung và các thầy cô giáo trong khoa Công nghệ thông tin – Ứng dụng phần mềm nói riêng đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt thời gian hai năm 4 tháng học vừa qua.

Đặc biệt em xin gửi lời cảm ơn đến cô giáo Vũ Thị Thanh Huyền đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn em trong suốt quá trình thực hiện dự án. Trong thời gian làm việc, em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc và hiệu quả, đây là những điều rất cần thiết cho em trong quá trình học tập và công tác sau này.

Sau cùng xin gửi lời cảm ơn chân thành tới gia đình, bạn bè đã động viên, đóng góp ý kiến và giúp đỡ em trong quá trình học tập, nghiên cứu và hoàn thành khoá luận tốt nghiệp.

Một lần nữa, em xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | Người thực hiên  (Ký và ghi rõ họ tên)  Nguyễn Ngô Anh Tuấn – Trần Thị Quỳnh – Trương Nguyễn Tiểu Long – Nguyễn Đức Tuấn Anh – Phạm Hồng Đạt |

**NHẬN XÉT CỦA GIẢNG VIÊN**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Giáo viên hướng dẫn ký, ghi rõ họ tên

**NHẬN XÉT CỦA HỘI ĐỒNG PHẢN BIỆN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

HĐ phản biện ký, ghi rõ họ tên

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Tên đầy đủ** | **Giải thích** |
| 1 | API | Application Programming Interface | Giao diện lập trình ứng dụng. |
| 2 | RESTful API | Representational state transfer | REST gửi một yêu cầu HTTP như GET, POST, DELETE,.. đến một URL để xử lý dữ liệu. |
| 3 | CSDL | Cơ sở dữ liệu | Là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) |
| 4 | JWT | JSON Web Token | JSON Web Token là một chuỗi mã hóa mà nguồn gốc ban đầu là một chuỗi JSON. Chuỗi thông tin dạng JSON bằng phương pháp mã hóa nào đó, nó trở thành 1 chuỗi ký tự lộn xộn nhìn vào sẽ rất khó hiểu |
|  |  |  |  |

# DANH MỤC HÌNH ẢNH

Hình 1: Mô hình triển khai tổng quan

Hình 2: Sơ đồ usecase hệ thống website Sunrise Shop

Hình 3: Sơ đồ usecase chi tiết của Manage Product

Hình 4: Sơ đồ usecase chi tiết của Manage Category

Hình 5: Sơ đồ usecase chi tiết của Manage User

Hình 6: Sơ đồ usecase chi tiết của Manage Statistical

Hình 7: Sơ đồ ERD của hệ thống

Hình 8: Sơ đồ Login Activity

Hình 9: Sơ đồ SignUp Activity

Hình 10: Sơ đồ Update Category Activity

Hình 11: Sơ đồ Checkout Activity

Hình 12: Chi tiết CSDL

Hình 13: Mockup giao diện Admin

Hình 14: Mockup giao diện người dùng

Hình 15: Giao diện dành cho Admin

Hình 16: Giao diện dành cho người dùng

Hình 17: Giao diện hiển thị sản phẩm

Hình 18: Giao diện trang đăng nhập

Hình 19: Giao diện đăng ký tài khoản

Hình 20: Giao diện trang giỏ hàng

Hình 21: Giao diện trang chi tiết sản phẩm

Hình 22: Giao diện trang quản lý tài khoản

Hình 23: Giao diện trang quản lý loại sản phẩm

Hình 24: Giao diện trang quản lý sản phẩm

Hình 25: Giao diện trang quản lý trạng thái đơn hàng

Hình 26: Giao diện Facebook chat

Hình 27: Giao diện quản lý thống kê

# GIỚI THIỆU THÀNH VIÊN

Thành viên thực hiện dự án gồm:

1. Nguyễn Ngô Anh Tuấn: Trưởng nhóm
2. Trần Thị Quỳnh
3. Trương Nguyễn Tiểu Long
4. Nguyễn Đức Tuấn Anh
5. Phạm Hồng Đạt

Bảng phân chia công việc:

|  |  |
| --- | --- |
| **Thành viên** | **Công việc** |
| Nguyễn Ngô Anh Tuấn | * Đăng nhập – đăng ký * Phân quyền và bảo mật * Trạng thái đơn hàng * Vẽ sơ đồ Classdiagram * Thống kê doanh thu |
| Trần Thị Quỳnh | * Thiết kế giao diện, đa ngôn ngữ * Quản lý sản phẩm * Trang chi tiết sản phẩm * Viết báo cáo |
| Trương Nguyễn Tiểu Long | * Lọc sản phẩm theo tên, giá * Quản lý users * Update profile * Thống kê sản phẩm bán chạy nhất * Vẽ sơ đồ usercase |
| Nguyễn Đức Tuấn Anh | * Quản lý loại sản phẩm * Giỏ hàng * Quên mật khẩu, gửi mail xác nhận mật khẩu * Vẽ sơ đồ Activity |
| Phạm Hồng Đạt | * Chức năng Add to cart * Phân trang sàn phẩm * Checkout * Vẽ sơ đồ ERD |

Contents

[I. TỔNG QUAN 12](#_Toc58430123)

[1. Hiện trạng 12](#_Toc58430124)

[2. Yêu cầu hệ thống 12](#_Toc58430129)

[2.1. Mô tả 12](#_Toc58430130)

[2.2. Chức năng 14](#_Toc58430131)

[2.3. Đối tượng sử dụng 14](#_Toc58430135)

[2.4. Môi trường hế thống 14](#_Toc58430136)

[2.5. Giao diện 14](#_Toc58430137)

[3. Khảo sát một số hệ thống tương tự 14](#_Toc58430138)

[II. THIẾT KẾ HỆ THỐNG 15](#_Toc58430139)

[1. Mô hình ứng dụng 15](#_Toc58430140)

[2. Use case 15](#_Toc58430141)

[2.1. Sơ đồ Use Case 15](#_Toc58430142)

[2.2. Mô tả Actor 18](#_Toc58430143)

[2.3. Mô tả Use Case 18](#_Toc58430144)

[2.4. Chi tiết Use Case 21](#_Toc58430145)

[2.4.1. Đăng ký tài khoản 21](#_Toc58430146)

[2.4.2. Xem giỏ hàng 21](#_Toc58430147)

[2.4.3. Xem và chỉnh sửa thông tin cá nhân 21](#_Toc58430148)

[2.4.4. Quản lý sản phẩm 22](#_Toc58430149)

[2.4.5. Xem chi tiết User 22](#_Toc58430150)

[2.4.6. Quản lý thống kê 22](#_Toc58430151)

[3. Thiết kế CSDL 23](#_Toc58430152)

[3.1. Sơ đồ ERD 23](#_Toc58430153)

[3.2. Mô tả ERD 23](#_Toc58430154)

[4. Sơ đồ Activity của ứng dụng 24](#_Toc58430155)

[4.1. Sơ đồ Login Activity 24](#_Toc58430156)

[4.2. Sơ đồ SignUp Activity 25](#_Toc58430157)

[4.3. Sơ đồ Update Category Activity 26](#_Toc58430158)

[4.4. Sơ đồ Checkout Activity 27](#_Toc58430159)

[III. THỰC HIỆN DỰ ÁN 28](#_Toc58430160)

[1. Các công nghệ sử dụng 31](#_Toc58430161)

[1.1. Back end 31](#_Toc58430162)

[1.2. Font end 31](#_Toc58430176)

[1.3. Hệ quản trị cơ sở dữ liệu 33](#_Toc58430185)

[1.4. Git (Version control system) 33](#_Toc58430192)

[2. Thiết kế CSDL 34](#_Toc58430193)

[2.1. Thiết kế CSDL 34](#_Toc58430194)

[2.2. Từ điển dữ liệu 35](#_Toc58430195)

[3. Thiết kế Model 37](#_Toc58430196)

[3.1. Model User 37](#_Toc58430197)

[3.2. Model ProductCategory 38](#_Toc58430198)

[3.3. Model ProductInfo 39](#_Toc58430199)

[3.4. Model ProductInOrder 39](#_Toc58430200)

[3.5. Model Order main 41](#_Toc58430201)

[3.6. Model Cart 42](#_Toc58430202)

[4. Thiết kế giao diện 43](#_Toc58430203)

[4.1. Phác thảo giao diện Admin 43](#_Toc58430204)

[4.2. Phác thảo giao diện người dùng (Products) 44](#_Toc58430205)

[4.3. Giao diện dành cho Admin 44](#_Toc58430206)

[4.4. Giao diện dành cho người dùng 45](#_Toc58430207)

[4.5. Giao diện hiển thị products 46](#_Toc58430208)

[IV. THỰC HIỆN VIẾT MÃ CHO TỪNG CHỨC NĂNG 46](#_Toc58430209)

[1. Triển khai hệ thống Rest-API 46](#_Toc58430210)

[2. Chức năng đăng nhập 48](#_Toc58430239)

[3. Chức năng đăng ký tài khoản 49](#_Toc58430240)

[4. Chức năng thêm sản phẩm vào giỏ hàng 49](#_Toc58430241)

[5. Chức năng hiển thị chi tiết sản phẩm 50](#_Toc58430242)

[6. Chức năng quản lý tài khoản 51](#_Toc58430243)

[7. Chức năng quản lý loại sản phẩm 52](#_Toc58430244)

[8. Chức năng quản lý sản phẩm 53](#_Toc58430245)

[9. Chức năng quản lý đơn hàng 54](#_Toc58430246)

[10. Chức năng Facebook chat 55](#_Toc58430247)

[11. Chức năng quản lý thống kê 55](#_Toc58430251)

[V. KẾT LUẬN 56](#_Toc58430253)

[1. Những điểm đã làm được trong dự án: 56](#_Toc58430254)

[2. Những vấn đề chưa giải quyết được trong dự án: 56](#_Toc58430255)

[3. Những thuận lợi và khó khăn trong quá trình làm dự án: 56](#_Toc58430256)

[VI. TÀI LIỆU THAM KHẢO 56](#_Toc58430257)

1. **TỔNG QUAN**
2. **Hiện trạng**

Trong thời đại thương mại điện tử đang ngày càng phát triển, việc ứng dụng website bán hàng đang ngày càng phổ biến. Với dự án “Xây dựng website thời trang online – Sunrise” sẽ mang tới nhiều lợi ích cho doanh nghiệp sử dụng:

* Tiết kiệm thời gian cho khách hàng cũng như nhà quản lý trong việc tham gia vào hoạt động mua bán, quảng bá sản phẩm.
* Khách hàng có thể tiết kiệm thời gian mua sắm mà không cần tới trực tiếp cửa hàng. Bên cạnh đó, khách hàng dễ dàng tìm kiếm sản phẩm theo tên, giá hoặc loại sản phẩm để lựa chọn được sản phẩm ưng ý với mức giá hợp lý.
* Giúp cho người quản lý dễ dàng hơn trong việc quản lý sản phẩm, khách hàng và thống kê đơn hàng.

1. **Yêu cầu hệ thống**
   1. **Mô tả**

Dịch vụ website sẽ mang lại cho khách hàng đầy đủ các thông tin về sản phẩm mà cửa hàng đang kinh doanh như tên sản phẩm, hình ảnh, giá thành cũng như hỗ trợ khách hàng nhanh chóng... Đồng thời khách hàng có thể thực hiện việc đặt mua sản phẩm ngay trên website thanh toán và nhận hàng ngay trong ngày khi đặt tại cơ sở cửa hàng gần nhất.

Về phía website thời trang Sunrise, quản trị viên sẽ dễ dàng tiếp nhận các đơn hàng và xử lý chúng ngay trên website. Bên cạnh đó, các thông tin về tài khoản khách hàng, thông tin chi tiết về sản phẩm, doanh thu đều được quản lý ngay tại trang quản trị.

* 1. **Chức năng**

### Khách hàng

Khách hàng là những người có nhu cầu mua sắm các sản phầm hiện đang có tại cửa hàng và đặt hàng trực tiếp trên trang web. Do đó, hệ thống phải có những yêu cầu sau đây:

* Trang web cung cấp chức năng đăng nhập, đăng ký để khách hàng có thể tương tác với trang web.
* Hiển thị các danh mục sản phẩm hiện đang có trong cửa hàng, ứng với đó là các danh sách sản phẩm của danh mục đó.
* Xem thông tin chi tiết về sản phẩm. Sau khi tìm được sản phẩm mình cần, khách hàng có thể xem thông tin chi tiết của sản phẩm đó và có thể thêm sản phẩm vào giỏ hàng của mình.
* Trang web cung cấp chức năng giỏ hàng. Chức năng này là không thể thiếu của mỗi trang web bán hàng. Với chức năng này, khách hàng có thể xem tất cả những sản phẩm mình đã chọn trước đó và tổng giá tiền mình cần thanh toán.
* Trang web cung cấp chức năng quản lý hồ sơ cá nhân. Đây là chức năng cực kì quan trọng của mỗi trang web bán hàng. Với chức năng này khách hàng có thể cập nhật thông tin của mình.

### Nhân viên

Nhân viên là người được quản lý các chức năng:

* Quản lý sản phẩm: Với chức năng này nhân viên có thể thêm mới, cập nhật, xóa, tìm kiếm sản phẩm.
* Quản lý loại sản phẩm: Với chức năng này nhân viên có thể thêm mới, cập nhật, xóa, tìm kiếm loại sản phẩm.
* Quản lý đơn hàng: Với chức năng này nhân viên có thể xác nhận và cập nhật trạng thái đơn hàng.

### Người quản trị

Người quản lý là người kiểm soát các hoạt động của hệ thống. Người quản lý được cung cấp tài khoản để đăng nhập vào trang web để thực hiện chức năng của mình:

* Quản lý sản phẩm: Với chức năng này người quản trị có thể thêm mới, cập nhật, xóa, tìm kiếm sản phẩm.
* Quản lý loại sản phẩm: Với chức năng này người quản trị có thể thêm mới, cập nhật, xóa, tìm kiếm loại sản phẩm.
* Quản lý tài khoản khách hàng: Với chức năng này người quản trị có thể xem danh sách tài khoản khách hàng.
* Quản lý đơn hàng: Với chức năng này người quản trị sẽ biết được trạng thái của đơn hàng và cập nhật trạng thái đơn hàng.
* Xem thống kê doanh thu của shop: Với chức năng này người quản trị có thể xem các thông tin như số lượng sản phẩm đã bán, tổng tiền sản phẩm đã bán theo tháng và theo năm.
  1. **Đối tượng sử dụng**

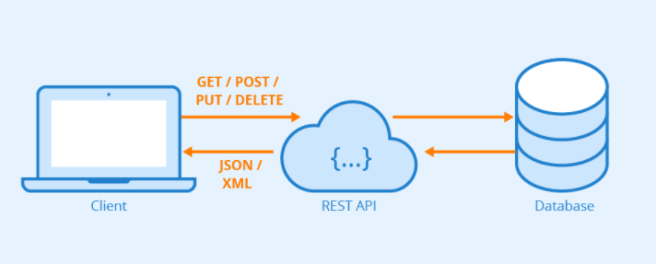
Với website thời trang Sunrise, hệ thống phục vụ cho 3 đối tượng chính: khách hàng, nhân viên và người quản trị.

* 1. **Môi trường hế thống**
* Một máy làm web server.
* Hệ điều hành Window 7 trở lên.
* Hệ cơ sở dữ liệu dùng là MySQL 2008 trở lên.
  1. **Giao diện**
* Giao diện người dùng phải được thiết kế đẹp mắt, có bố cục hợp lý, màu sắc hết hợp hài hòa để không gây rối mắt khi thao tác trên website. Giao diện phải đủ các chức năng cần thiết: Trang chủ, sản phẩm, giỏ hàng, giới thiệu, liên hệ, đăng nhập và đăng ký. Hiển thị đầy đủ các thông tin, nhưng không phức tạp, rối mắt.
* Giao diện người quản trị phải cung cấp đầy đủ các chức năng của hệ thống: danh sách users, danh sách sản phẩm, loại sản phẩm, danh sách các đơn hàng và thống kê doanh thu.

1. **Khảo sát một số hệ thống tương tự**

|  |  |  |
| --- | --- | --- |
| **#** | **IVY Moda** | **Yody** |
| Tính năng nổi bật | Giao diện trang chủ đơn giản, dễ sử dụng.  Hiển thị các loại sản phẩm  Tìm kiếm sản phẩm theo tên hoặc loại sản phẩm.  Hiển thị chi tiết sản phẩm,các sản phẩm tương tự.  Đặt hàng và thanh toán dễ dàng.  Có trang cập nhật tin tức và bộ sưu tập | Giao diện trang chủ thanh lịch, dễ sử dụng  Hiển thị chi tiết sản phẩm, loại sản phẩm  Dễ dàng tìm kiếm sản phẩm theo tên  Đặt hàng và thanh toán nhanh chóng.  Có trang cập nhật tin tức và bộ sưu tập mới |

1. **THIẾT KẾ HỆ THỐNG**
2. **Mô hình ứng dụng**

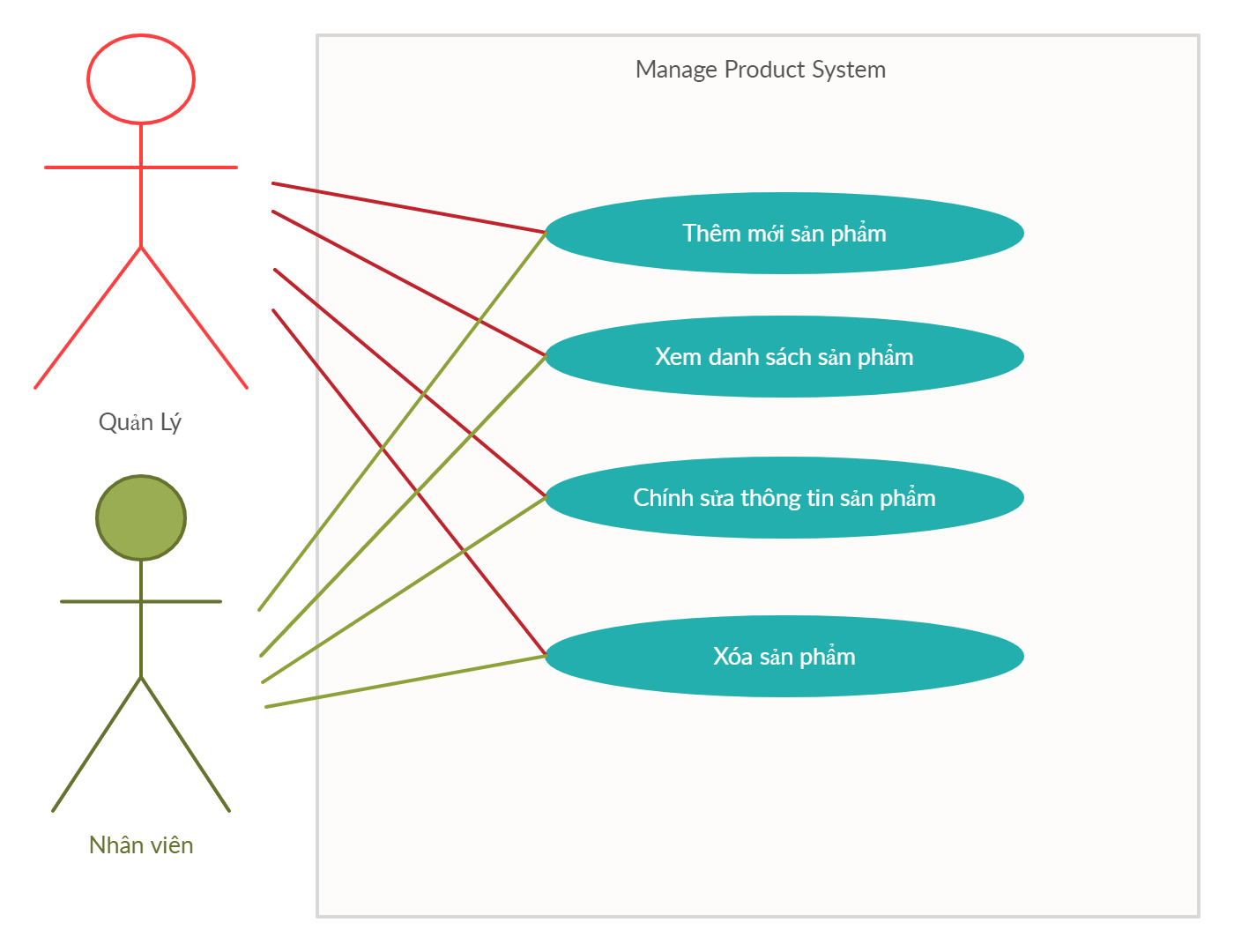


Hình 1: Mô hình triển khai tổng quan

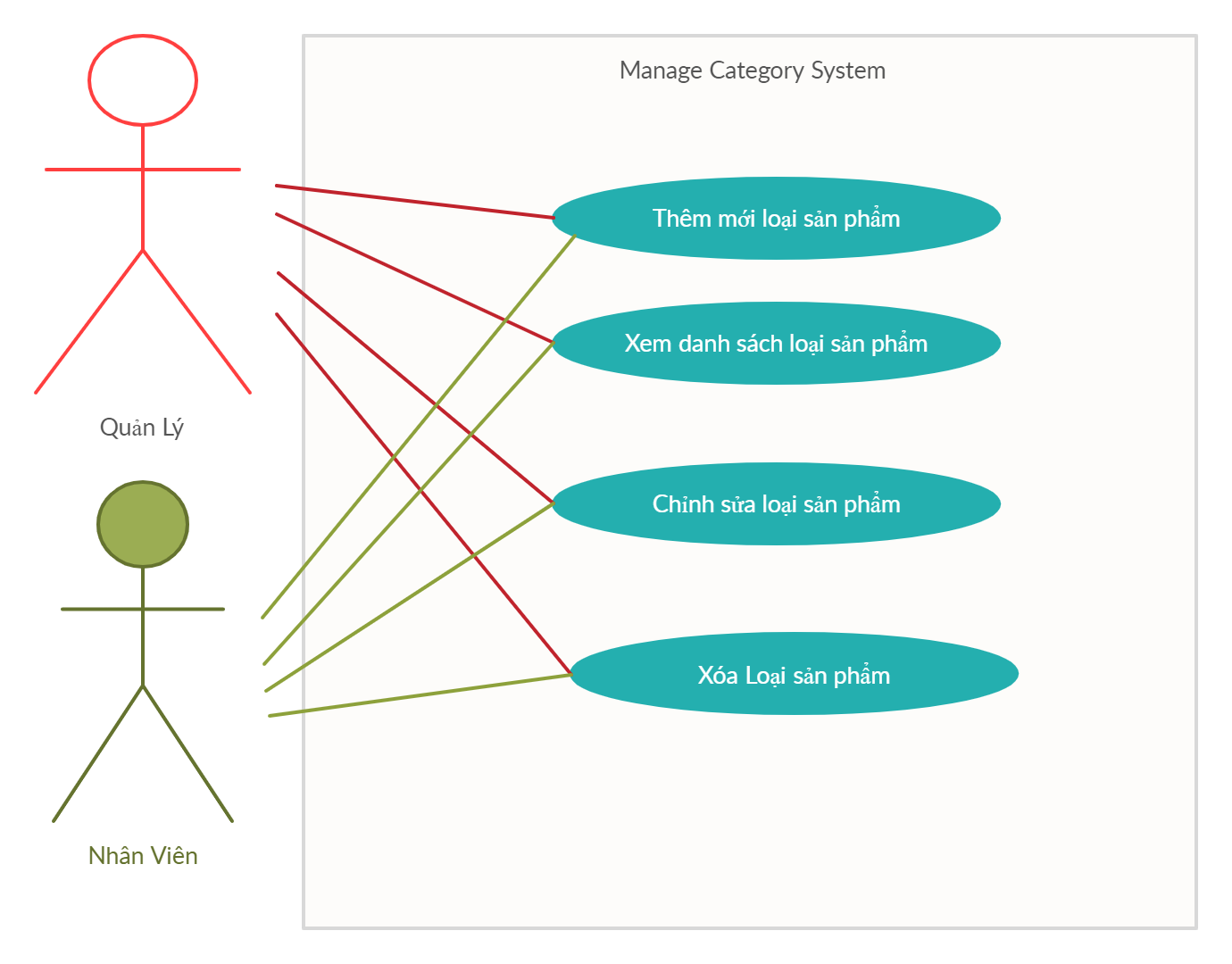
1. **Use case**
   1. **Sơ đồ Use Case**

****

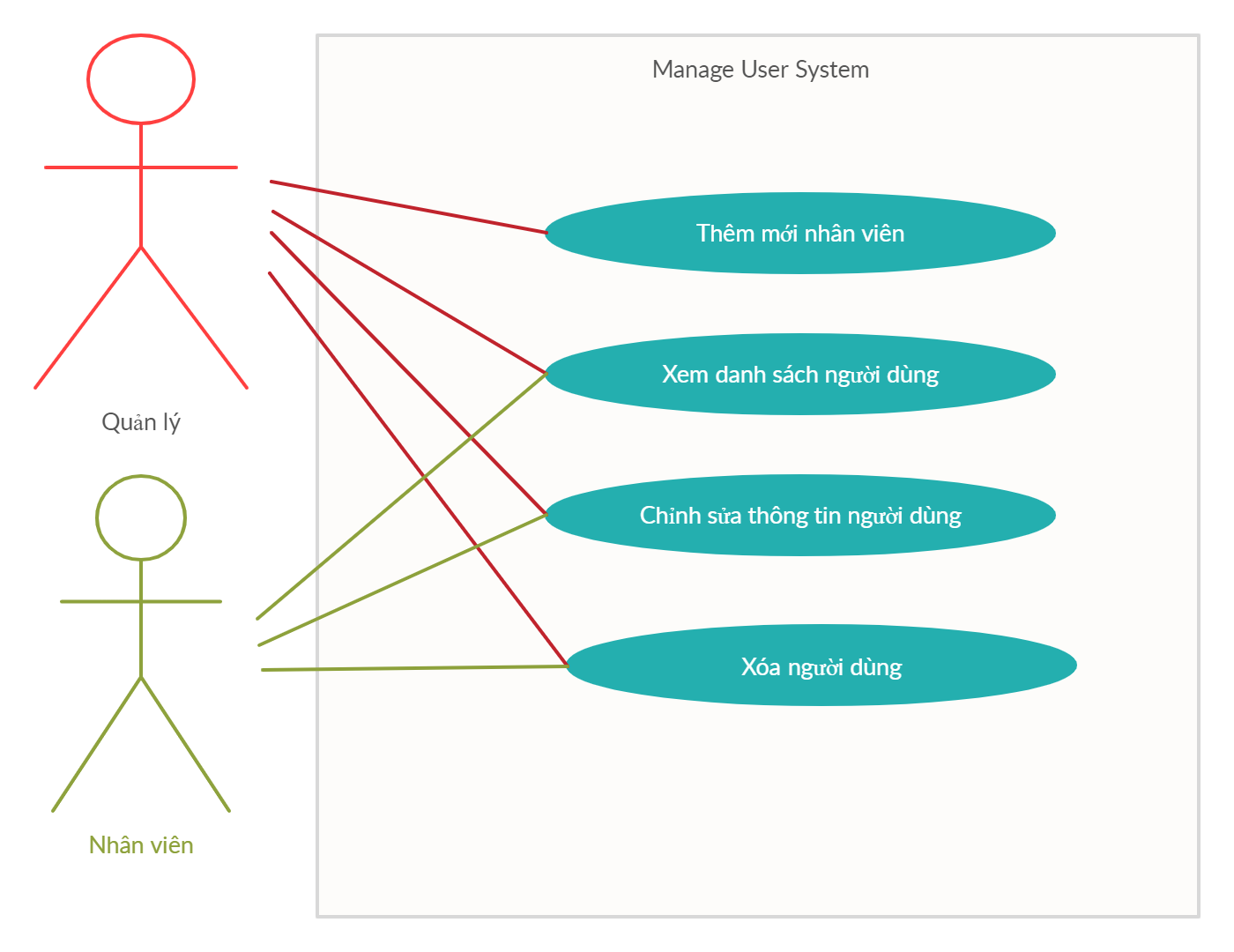
Hình 2: Sơ đồ usecase hệ thống website Sunriseshop



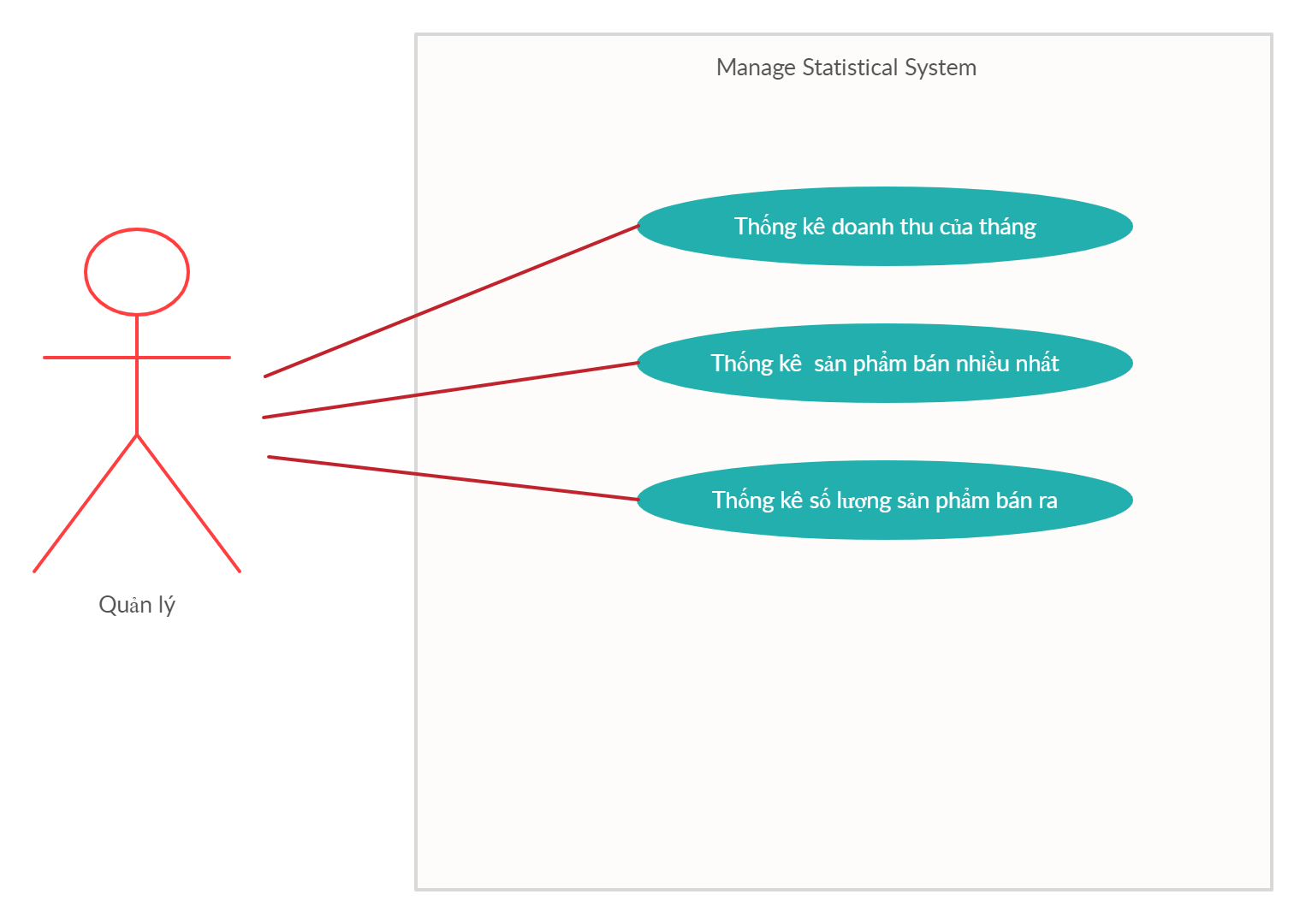
Hình 3: Sơ đồ usecase chi tiết của Manage Product



Hình 4: Sơ đồ usecase chi tiết của Manage Category



Hình 5: Sơ đồ usecase chi tiết của Manage User



Hình 6: Sơ đồ usecase chi tiết của Manage Statistical

* 1. **Mô tả Actor**

|  |  |  |
| --- | --- | --- |
| **STT** | **Actor Name** | **Definition & Interests** |
| 1 | Admin | * Có quyền sửa thông tin cá nhân, mật khẩu * Quản lý thêm nhân viên, sửa, xóa thông tin user có trong hệ thống * Quản lý thêm, sửa, xóa loại sản phẩm * Quản lý thêm, sửa, xóa các sản phẩm * Quản lý các đơn hàng có trong hệ thống * Quản lý thông kê của cửa hàng. |
| 2 | Staff | * Có quyền sửa thông tin cá nhân, mật khẩu * Có quyền xem thông tin user có trong hệ thống * Quản lý thêm, sửa, xóa loại sản phẩm * Quản lý thêm, sửa, xóa các sản phẩm * Quản lý các đơn hàng có trong hệ thống |
| 3 | User | * User có quyền xem, thêm và xóa các sản phẩm trong giỏ hàng của mình. * User có quyền xem các dịch vụ. * User có quyền xem các đơn hàng đã đặt của mình. * User có quyền hủy các đơn hàng đã đặt của mình. * User có quyền xem và chỉnh sửa thông tin cá nhân. * User có quyền check out để mua sản phẩm. |
| 4 | Visitor | * Visitor có thể đăng ký tài khoản để đăng nhập và mua hàng. * Visitor có quyền xem, thêm và xóa các sản phẩm trong giỏ hàng của mình. * Visitor có quyền xem các dịch vụ. |
|  |  |  |

* 1. **Mô tả Use Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Code** | **Tên UseCase** | **Mô tả** |
| 1 | UC01 | Resgister | Cho phép khách hàng mới đến trang web đăng ký tài khoản để thực hiện giao dịch. |
| 2 | UC02 | View Shoping Card | Hiển thị các sản phẩm và cho phép người dùng thêm các sản phẩm vào giỏ hàng |
| 3 | UC03 | Xem các sản phẩm | Khách hàng mới và khách hàng đã đăng ký tài khoản có thể xem danh sách các sản phẩm mà website hiển thị |
| 4 | UC04 | Thêm các sản phẩm vào giỏ hàng | Khách hàng mới và khách hàng đã đăng ký tài khoản có thể thêm các sản phẩm mình muốn mua vào giỏ hàng |
| 5 | UC05 | Xóa các sản phẩm trong giỏ hàng | Khách hàng mới và khách hàng đã đăng ký tài khoản có thể xóa các sản phẩm mình muốn mua trong giỏ hàng |
| 6 | UC06 | Chỉnh sửa các sản phẩm trong giỏ hàng | Khách hàng mới và khách hàng đã đăng ký tài khoản có thể chỉnh sửa các sản số lượng sản phẩm trong giỏ hàng của mình. |
| 7 | UC07 | View Service | Khách hàng có thể xem các dịch vụ mà shop cập nhập trên website |
| 8 | UC08 | Gửi mail | Khách hàng mới và khách hàng đã đăng ký tài khoản có thể gửi Email góp ý hay phàn nàn về dịch vụ của shop |
| 9 | UC09 | View & Edit UserProfile | Khách hàng đã đăng nhập có thể xem quản lý thông tin cá nhân của mình. |
| 10 | UC10 | Xem thông tin cá nhân | Khách hàng đã đăng nhập tài khoản của mình có thể xem tất cả thông tin cá nhân của tài khoản đó. |
| 11 | UC11 | Sửa đổi thông tin cá nhân | Khách hàng đã đăng nhập tài khoản của mình có thể sửa đổi thông tin cá nhân của mình |
| 12 | UC12 | Check out | Khách hàng đã đăng nhập tài khoản của mình có thể mua các sản phẩm trong giỏ hàng của mình |
| 13 | UC13 | View&Update Status Order | Khách hàng, nhân viên và quản lý đã đăng nhập tài khoản của mình có thể quản lý các đơn hàng. |
| 14 | UC14 | Xem các đơn hàng đã mua(Khách hàng) | Khách hàng đã đăng nhập tài khoản của mình có thể xem các đơn hàng đã mua. |
| 15 | UC15 | Hủy đơn hàng(Khách hàng) | Khách hàng đã đăng nhập tài khoản của mình có thể hủy đơn hàng đã đặt mua. |
| 16 | UC16 | Xem các đơn hàng đã mua(Quản lý và nhân viên) | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể xem tất cả các đơn hàng mà khách hàng đã đặt mua. |
| 17 | UC17 | Hủy đơn hàng(Quản lý và nhân viên) | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể Cancel hoặc Fisnish các đơn hàng mà khách hàng đã đặt. |
| 18 | UC18 | Manage Product | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể quản lý các sản phẩm trên website. |
| 19 | UC19 | Xem các sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể xem danh sách các sản phẩm. |
| 20 | UC20 | Thêm các sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể thêm mới sản phẩm vào danh sách sản phẩm trên website. |
| 21 | UC21 | Sửa thông tin sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể thay đổi một số thông tin của sản phẩm |
| 22 | UC22 | Xóa sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoảng của mình có thể xóa các sản phẩm trong danh sách các sản phẩm |
| 23 | UC23 | Manage Category | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể quản lý các loại sản phẩm trên website. |
| 24 | UC24 | Xem các loại sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể xem danh sách các loại sản phẩm. |
| 25 | UC25 | Thêm các loại sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể thêm mới các loại sản phẩm vào danh sách các loại sản phẩm trên website. |
| 26 | UC26 | Sửa thông tin các loại sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoảng của mình có thể thây đổi một số thông tin của các loại sản phẩm |
| 27 | UC27 | Xóa các loại sản phẩm | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể xóa các loại sản phẩm trong danh sách các sản phẩm |
| 28 | UC28 | Manage User | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể quản user của khách hàng |
| 29 | UC29 | Xem danh sách khách hàng | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể xem danh sách các user khách hàng. |
| 30 | UC30 | Thêm nhân viên | Quản lý đã đăng nhập tài khoản của mình có thể thêm mới nhân viên cho website |
| 32 | UC31 | Sửa thông tin Khách hàng | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể thay đổi một số thông tin của khách hàng. |
| 33 | UC33 | Xóa Khách hàng | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể xóa các user khách hàng trong danh sách user |
| 34 | UC34 | Manage Statistical | Quản lý đã đăng nhập tài khoản của mình có thể quản lý thống kê của shop. |

* 1. **Chi tiết Use Case**
     1. **Đăng ký tài khoản**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Resgister | **Code** | UC01 |
| **Mô tả** | Cho phép khách hàng mới đến trang web đăng ký tài khoản để thực hiện giao dịch. | | |
| **Actor** | Visister | **Actor** | Visister |
| **Điều Kiện** | Khách hàng chưa có tài khoản đăng nhập hoặc chưa đăng nhập vào hệ thống | | |
| **Thực hiện** | Khách hàng có thể điền các thông tin cơ bản để đăng ký tài khoảng để đăng nhập vào hệ thống | | |

* + 1. **Xem giỏ hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | View Shoping Cart | **Code** | UC02 |
| **Mô tả** | Hiển thị các sản phẩm và cho phép người dùng thêm các sản phẩm vào giỏ hàng | | |
| **Actor** | Visister&User | **Kích hoạt** | Visister&User vào chức năng quản lý giỏ hàng |
| **Điều Kiện** | Không có | | |
| **Thực hiện** | Visister&User thực hiện xóa, tăng và giảm số lượng các sản phẩm trong giỏ hàng | | |

* + 1. **Xem và chỉnh sửa thông tin cá nhân**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | View & Edit UserProfile | **Code** | UC09 |
| **Mô tả** | Khách hàng đã đăng nhập có thể xem quản lý thông tin cá nhân của mình. | | |
| **Actor** | User | **Kích hoạt** | User vào chức năng thông tin cá nhân. |
| **Điều Kiện** | Phải đăng nhập vào hệ thống | | |
| **Thực hiện** | User có thể xem và thay đổi một số thông tin cá nhân của mình | | |

* + 1. **Quản lý sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Manage Product | **Code** | UC18 |
| **Mô tả** | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể quản lý các sản phẩm trên website. | | |
| **Actor** | Admin&Staff | **Kích hoạt** | Admin&Staff vào chức năng quản lý sản phẩm |
| **Điều Kiện** | Phải đăng nhập tài khoản vào hệ thống với vai trò là Admin&Staff | | |
| **Thực hiện** | Admin&Staff có thể xem, thêm mới, sửa thông tin và xóa sản phẩm trong hệ thống | | |

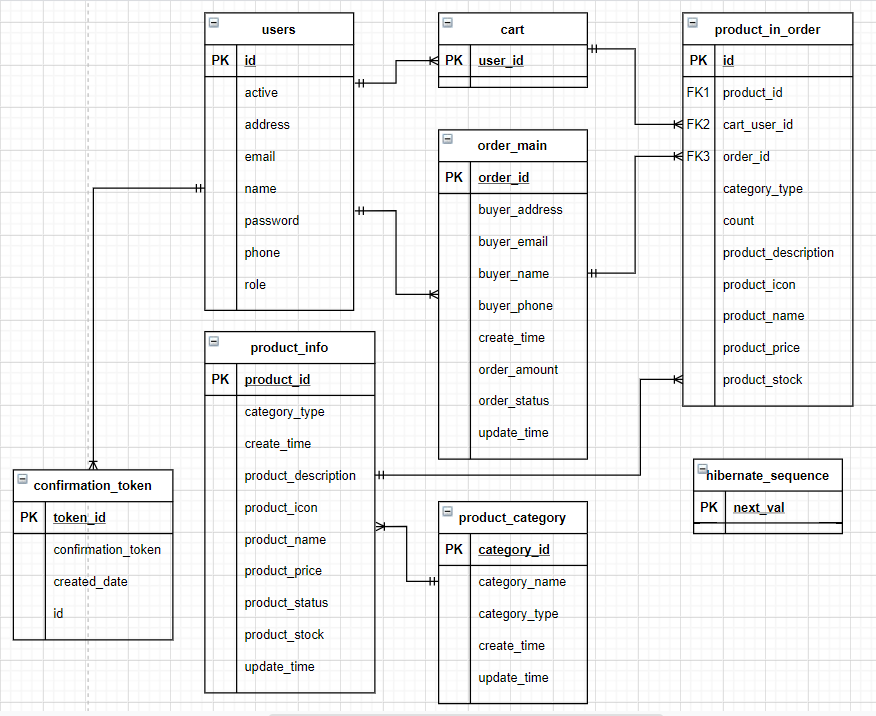
* + 1. **Xem chi tiết User**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Manage User | **Code** | UC28 |
| **Mô tả** | Quản lý và nhân viên đã đăng nhập tài khoản của mình có thể quản user của khách hàng | | |
| **Actor** | Admin&Staff | **Kích hoạt** | Admin&Staff vào chức năng quản lý User |
| **Điều Kiện** | Phải đăng nhập vào hệ thống với vai trò là Admin&Staff | | |
| **Thực hiện** | Admin&Staff có thể xem , sửa đổi thông tin và xóa User(Chỉ có Admin mới thêm được nhân viên) | | |

* + 1. **Quản lý thống kê**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | Manage Statistical | **Code** | UC34 |
| **Mô tả** | Quản lý đã đăng nhập tài khoản của mình có thể quản lý thống kê của shop. | | |
| **Actor** | Admin | **Kích hoạt** | Admin vào chức năng quản lý thống kê |
| **Điều Kiện** | Phải đăng nhập vào hệ thống với vai trò là Admin | | |
| **Thực hiện** | Admin có thể xem thống bảng thống kê doanh số của hệ thống | | |

1. **Thiết kế CSDL**
   1. **Sơ đồ ERD**



Hình 7: Sơ đồ ERD của hệ thống

* 1. **Mô tả ERD**

|  |  |
| --- | --- |
| **Cardinality** | **Mô tả** |
|  | Chỉ một và bắt buộc có một dữ liệu |
|  | Ít nhất phải có một và nhiều dữ liệu |

Về cơ bản thì relationship của ERD có 3 loại:

One-to-One: quan hệ 1-1

One-to-Many: quan hệ 1-nhiều

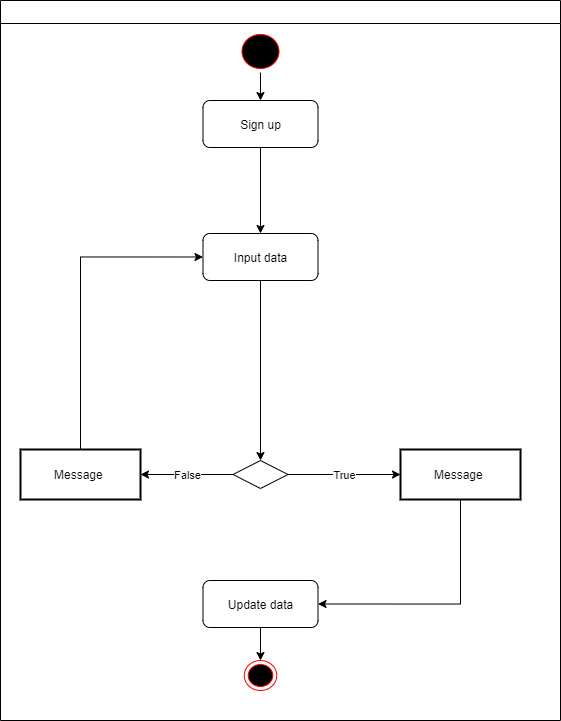
Many-to-Many: quan hệ nhiều-nhiều.

1. **Sơ đồ Activity của ứng dụng**
   1. **Sơ đồ Login Activity**



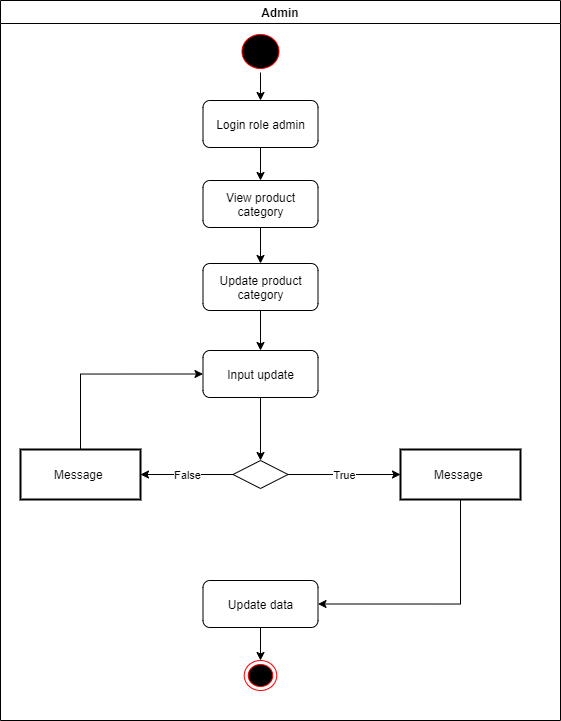
Hình 8: Sơ đồ Login Activity

* 1. **Sơ đồ SignUp Activity**



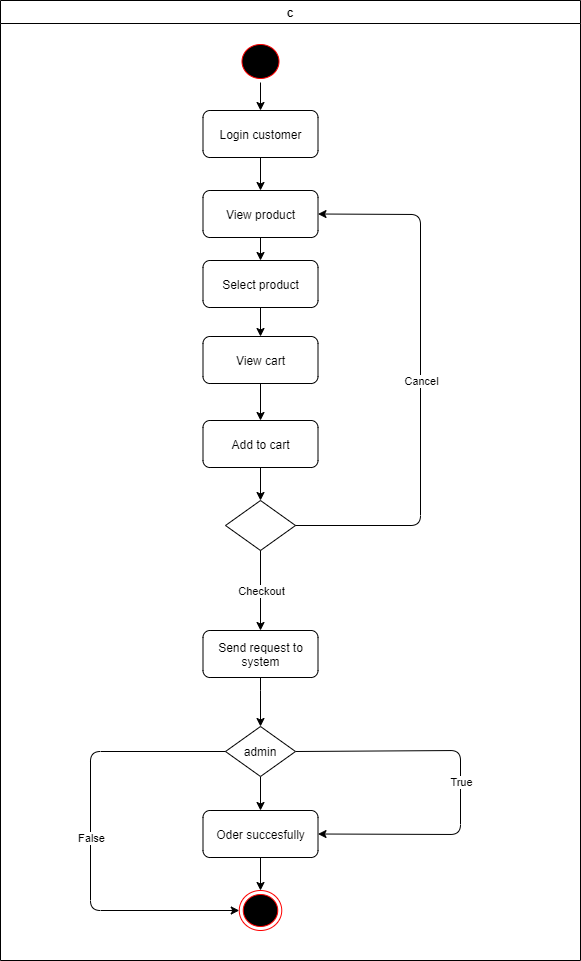
Hình 9: Sơ đồ SignUp Activity

* 1. **Sơ đồ Update Category Activity**



Hình 10: Sơ đồ Update Category Activity

* 1. **Sơ đồ Checkout Activity**



Hình 11: Sơ đồ Checkout Activity

1. **THỰC HIỆN DỰ ÁN**
2. **Các công nghệ sử dụng**
   1. **Back end**

**Spring Boot**



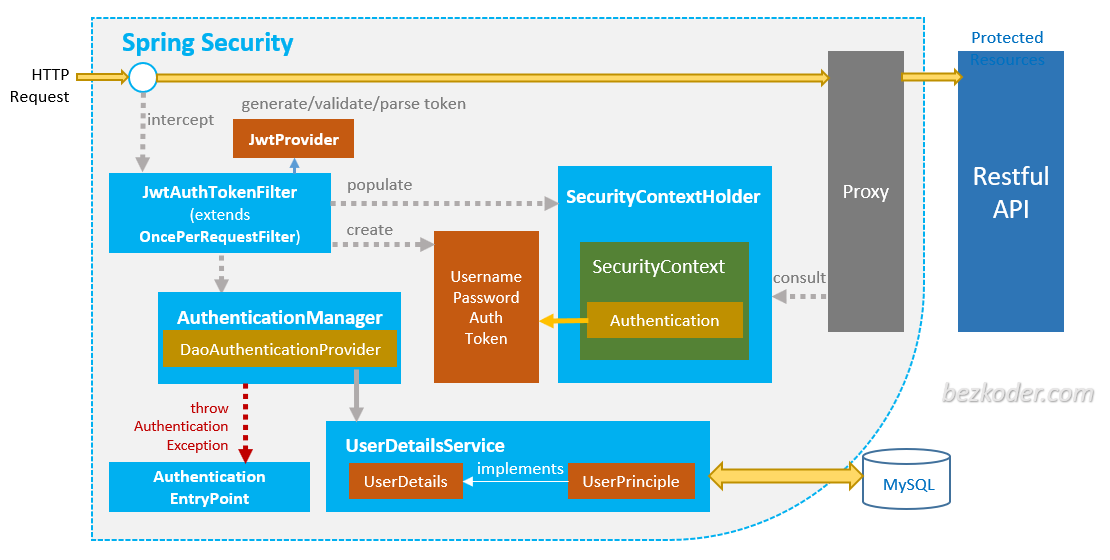
Spring Boot là một dự án nổi bật trong hệ sinh thái Spring Framework. Nếu như trước đây, công đoạn khởi tạo một dự án Spring khá vất vả từ việc khai báo các dependency trong file pom.xml cho đến cấu hình bằng XML hoặc annotation phức tạp, thì giờ đây với Spring Boot, chúng ta có thể tạo các ứng dụng Spring một cách nhanh chóng và cấu hình cũng đơn giản hơn.

* Tính năng nổi bật:
* Đóng gói ứng dụng Spring dưới dạng một file JAR (stand-alone application). Chúng ta có thể dễ dàng start ứng dụng Spring chỉ với câu lệnh quen thuộc java -jar.
* Tối ưu công đoạn cấu hình cho ứng dụng Spring, không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML.
* Cung cấp một loạt các tính năng phi chức năng phổ biến cho các dự án lớn như nhúng trực tiếp web server như Tomcat, Jetty,... vào ứng dụng, bảo mật, health check, ...
* Phiên bản: Hiện tại, Spring Boot có 2 version chính là:
* v1.5.x
* v2.x

**Java (Spring Restfull API)**

Đây là sơ đồ cho các lớp Spring Security / JWT được tách thành 3 lớp:

* HTTP
* Spring sercurity
* API REST



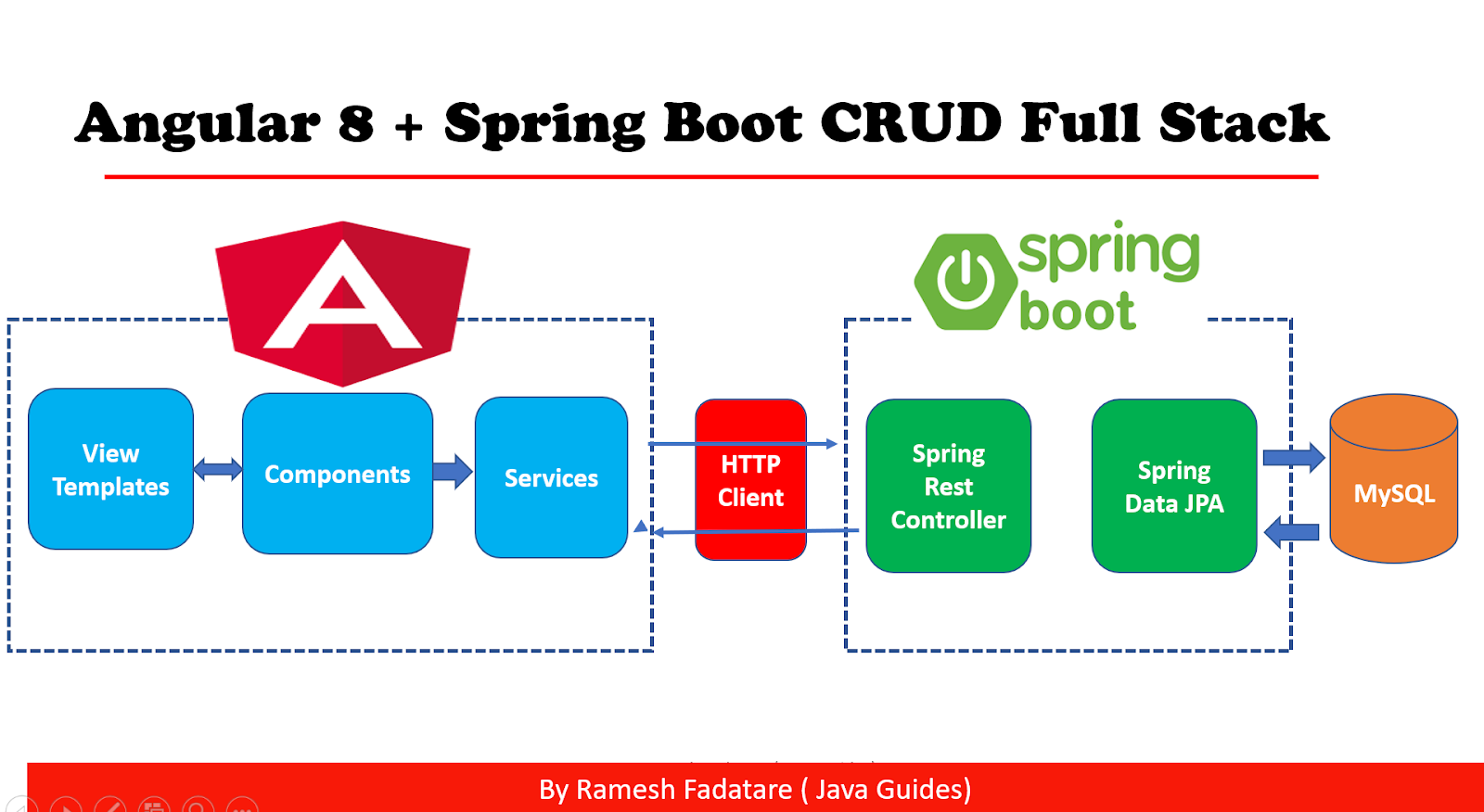
* SecurityContextHolder cung cấp quyền truy cập vào SecurityContext.
* SecurityContext giữ Thông tin xác thực và có thể là thông tin bảo mật theo yêu cầu cụ thể.
* Authentication cho thông tin, bao gồm GrantedAuthority phản ánh các quyền trên toàn ứng dụng được cấp cho người chính.
* UserDetails chứa thông tin cần thiết để xây dựng đối tượng Xác thực từ các DAO hoặc nguồn dữ liệu bảo mật khác.
* UserDetailsService giúp tạo UserDetails từ tên người dùng dựa trên chuỗi được sử dụng bởi AuthenticationProvider.
* JwtAuthTokenFilter (mở rộng OncePerRequestFilter) xử lý trước yêu cầu HTTP, từ Token, tạo Xác thực và đưa nó vào SecurityContext.
* JwtProvider xác thực, phân tích cú pháp chuỗi mã thông báo hoặc tạo chuỗi mã thông báo từ UserDetails.
* UsernamePasswordAuthenticationToken lấy tên người dùng / mật khẩu từ Login request và kết hợp thành một phiên bản của giao diện Xác thực.
* AuthenticationManager sử dụng DaoAuthenticationProvider (với sự trợ giúp của UserDetailsService & PasswordEncoder) để xác thực phiên bản
* UsernamePasswordAuthenticationToken, sau đó trả về phiên bản Authentication được điền đầy đủ khi xác thực thành công.
* SecurityContext được thiết lập bằng cách gọi SecurityContextHolder.getContext (). SetAuthentication(…) với đối tượng xác thực trả về ở trên.
* AuthenticationEntryPoint xử lý AuthenticationException.
* Quyền truy cập vào API Restful được bảo vệ bởi HTTPSecurity và được ủy quyền với biểu thức bảo mật.



**Hibernate** là một triển khai JPA, trong khi Spring Data JPA là một Trừu tượng truy cập dữ liệu JPA. Spring Data cung cấp một giải pháp để GenericDaothực hiện tùy chỉnh. Nó cũng có thể tạo các truy vấn JPA thay mặt bạn thông qua các quy ước tên phương thức.

Với Spring Data, bạn có thể sử dụng Hibernate, Eclipse Link hoặc bất kỳ nhà cung cấp JPA nào khác. Một lợi ích rất thú vị là bạn có thể kiểm soát ranh giới giao dịch một cách khai báo bằng cách sử dụng @Transactional chú thích.

* 1. **Font end**



**Angular CLI**: Angular cung cấp một số những tiện ích cho phép người dùng tạo và quản lý các project bằng command line. Nó tự động hóa các tác vụ như là tạo project, add thêm controllers mới... Nói chung là nên sử dụng Angular CLI vì nó giúp tạo và maintain các common patterns trong ứng dụng của chúng ta.

**Angular CLI** là một công cụ được phát triển đề chạy trên giao diện cửa sổ ứng dụng (command line interface hay CLI) nhằm giúp đỡ việc tạo dự án, quản lý tập tin trong dự án và thực hiện nhiều tác vụ khác nhau như test (kiểm thử), bundle và deploy dự án một cách nhanh chóng và hiệu quả.

**Bootstrap:**

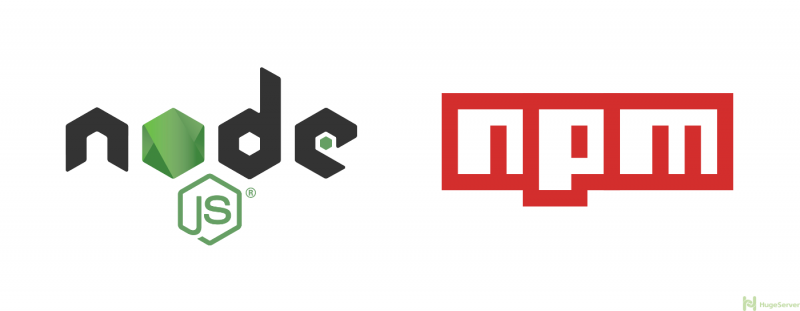


**Bootstrap** là front-end framework miễn phí ngày càng phổ biến hơn trong số các nhà phát triển front-end. Các nhà phát triển có thể dễ dàng sử dụng và tiết kiệm rất nhiều thời gian khi  khỏi phải viết code thủ công.

Bootstrap bao gồm những cái cơ bản như: row, column, typography, forms, buttons, tables, navigation, modals, image carousels… Trong bootstrap có thêm nhiều [component](https://topdev.vn/blog/component-la-gi/), Javascript hỗ trợ cho việc xây dựng reponsive cho web dễ dàng, thuận tiện và nhanh chóng hơn.

Framework này cũng rất linh hoạt và có thể đáp ứng hầu hết mọi nhu cầu phát triển front end web. Các khả năng tốt nhất của nó bao gồm, nhưng không giới hạn tính năng responsive web tối ưu trên tất cả các kích thước màn hình.

**NodeJs + Npm:**



**NPM** viết tắt của Node Package Manager là một công cụ tạo và quản lý các thư viện lập trình Javascript cho NodeJS. Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework.

* 1. **Hệ quản trị cơ sở dữ liệu**



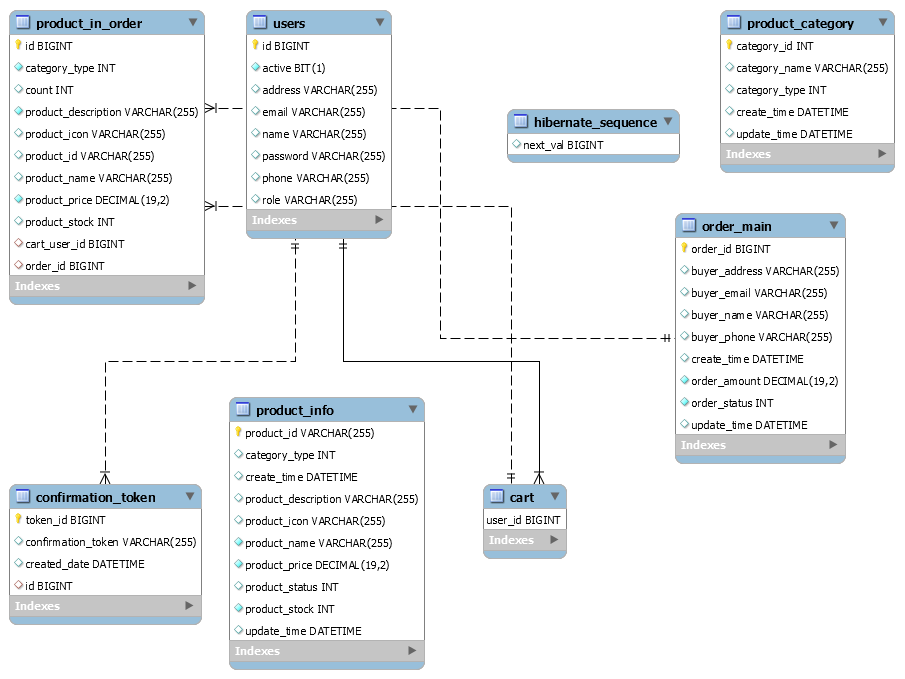
MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

* Ưu điểm:
* Bảo mật dữ liệu
* Hiệu suất cao
* Mã nguồn mở
  1. **Git (Version control system)**



**GitHub** là một dịch vụ cung cấp [kho lưu trữ mã nguồn](https://vi.wikipedia.org/w/index.php?title=Kho_l%C6%B0u_tr%E1%BB%AF_m%C3%A3_ngu%E1%BB%93n&action=edit&redlink=1) [Git](https://vi.wikipedia.org/wiki/Git_(ph%E1%BA%A7n_m%E1%BB%81m)) dựa trên nền web cho các dự án phát triển phần mềm. **GitHub** là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.Tính đến tháng 4 năm 2016, GitHub có hơn 14 triệu người sử dụng với hơn 35 triệu kho mã nguồn, làm cho nó trở thành máy chủ chứa mã nguồn lớn trên thế giới.

1. **Thiết kế CSDL**
   1. **Thiết kế CSDL**



Hình 12: Chi tiết CSDL

* 1. **Từ điển dữ liệu**
* Bảng User (Tài khoản):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Loại khoá** | **Các ràng buộc** | **Mô tả** |
| id | bigint | PK | NOT NULL | Mã người dùng |
| active | bit(1) |  | NOT NULL |  |
| address | varchar(255) |  | NOT NULL | Địa chỉ người dùng |
| email | varchar(255) |  | NOT NULL | Email người dùng |
| name | varchar(255) |  | NOT NULL | Tên người dùng |
| password | varchar(255) |  | NOT NULL | Mật khẩu người dùng |
| phone | varchar(255) |  | NOT NULL | SDT người dùng |
| role | varchar(255) |  | NOT NULL | Quyền người dùng |

* Bảng Product\_Category (Loại sản phẩm):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Loại khoá** | **Các ràng buộc** | **Mô tả** |
| category\_id | int | PK | NOT NULL | Mã loại sản phẩm |
| category\_name | varchar(255) |  | NOT NULL | Tên loại sản phẩm |
| category\_type | int |  | NOT NULL | Loại category |
| create\_time | datetime |  | NOT NULL | Ngày thêm loại SP |

* Bảng Product (Sản phẩm):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Loại khoá** | **Các ràng buộc** | **Mô tả** |
| product\_id | varchar(255) | PK | NOT NULL | Mã sản phẩm |
| category\_type | int |  | NOT NULL | Loại Category |
| create\_time | datetime |  | NOT NULL | Ngày thêm SP |
| product\_description | varchar(255) |  | NOT NULL | Mô tả sản phẩm |
| product\_icon | varchar(255) |  | NOT NULL | Hình ảnh sản phẩm |
| product\_name | varchar(255) |  | NOT NULL | Tên sản phẩm |
| product\_ price | decimal(19,2) |  | NOT NULL | Giá sản phẩm |
| product\_status | int |  | NOT NULL | Trạng thái sản phẩm |
| product\_stock | int |  | NOT NULL | Kho |
| update\_time | datetime |  | NOT NULL | Xếp hạng yêu thích |

* Bảng Product\_in\_order (Thêm vào giỏ hàng):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Loại khoá** | **Các ràng buộc** | **Mô tả** |
| id | bigint | PK | NOT NULL | Mã |
| category\_type | int |  | NOT NULL | Loại Category |
| count | int |  | NOT NULL | Số lượng sản phẩm |
| product\_description | varchar(255) |  | NOT NULL | Mô tả sản phẩm |
| product\_icon | varchar(255) |  | NOT NULL | Hình ảnh sản phẩm |
| product\_id | varchar(255) |  | NOT NULL | Mã sản phẩm |
| product\_name | varchar(255) |  | NOT NULL | Tên sản phẩm |
| product\_price | decimal(19,2) |  | NOT NULL | Giá sản phẩm |
| product\_stock | int |  | NOT NULL | Kho hàng |
| cart\_user\_id | bigint |  | NOT NULL | Mã khách hàng |
| order\_id | bigint |  | NOT NULL | Mã đơn hàng |

* Bảng Order\_main ():

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Loại khoá** | **Các ràng buộc** | **Mô tả** |
| order\_id | bigint | PK | NOT NULL | Mã đơn hàng |
| buyer\_address | varchar(255) |  | NOT NULL | Địa chỉ người dùng |
| buyer\_email | varchar(255) |  | NOT NULL | Email người dùng |
| buyer\_name | varchar(255) |  | NOT NULL | Tên người dùng |
| buyer\_phone | varchar(255) |  | NOT NULL | SDT người dùng |
| create\_time | datetime |  | NOT NULL | Ngày đặt hàng |
| order\_amount | decimal(19,2) |  | NOT NULL | Tổng tiền đơn hàng |
| order\_status | int |  | NOT NULL | Trạng thái đơn hàng |
| update\_time | datetime |  | NOT NULL | Ngày sửa đổi |

1. **Thiết kế Model**
   1. **Model User**

|  |
| --- |
| package me.zhulin.shopapi.entity;  import com.fasterxml.jackson.annotation.JsonIgnore; import lombok.Data; import lombok.NoArgsConstructor; import org.hibernate.annotations.NaturalId; import javax.persistence.\*; import javax.validation.constraints.NotEmpty; import javax.validation.constraints.NotNull; import javax.validation.constraints.Size; import java.io.Serializable;  @Entity @Data @Table(name = "users") @NoArgsConstructor public class User implements Serializable {  private static final long *serialVersionUID* = 4887904943282174032L;  @Id  @GeneratedValue(strategy = GenerationType.*AUTO*)  private Long id;  @NaturalId  @NotEmpty  private String email;  @NotEmpty  @Size(min = 3, message = "Length must be more than 3")  private String password;  @NotEmpty  private String name, phone, address;  @NotNull  private boolean active;  @NotEmpty  private String role = "ROLE\_CUSTOMER";  @OneToOne(mappedBy = "user", cascade = CascadeType.*ALL*, fetch = FetchType.*LAZY*)  @JsonIgnore *// fix bi-direction toString() recursion problem* private Cart cart;  @Override  public String toString() {  return "User{" +  "id=" + id +  ", email='" + email + '\'' +  ", password='" + password + '\'' +  ", name='" + name + '\'' +  ", phone='" + phone + '\'' +  ", address='" + address + '\'' +  ", active=" + active +  ", role='" + role + '\'' +  '}'; } \*\*\* getter setter \*\*\* |

* 1. **Model ProductCategory**

|  |
| --- |
| package me.zhulin.shopapi.entity;  import lombok.Data; import org.hibernate.annotations.DynamicUpdate; import org.hibernate.annotations.NaturalId; import javax.persistence.Entity; import javax.persistence.GeneratedValue; import javax.persistence.Id; import java.io.Serializable; import java.util.Date;  @Entity @Data @DynamicUpdate public class ProductCategory implements Serializable {  @Id  @GeneratedValue  private Integer categoryId;   private String categoryName;   @NaturalId  private Integer categoryType;   private Date createTime, updateTime;   public ProductCategory() {  }   public ProductCategory(String categoryName, Integer categoryType) {  this.categoryName = categoryName;  this.categoryType = categoryType;  }  \*\*\* getter setter \*\*\* |

* 1. **Model ProductInfo**

|  |
| --- |
| package me.zhulin.shopapi.entity;  import lombok.Data; import org.hibernate.annotations.ColumnDefault; import org.hibernate.annotations.CreationTimestamp; import org.hibernate.annotations.DynamicUpdate; import org.hibernate.annotations.UpdateTimestamp; import javax.persistence.Entity; import javax.persistence.Id; import javax.validation.constraints.Min; import javax.validation.constraints.NotNull; import java.io.Serializable; import java.math.BigDecimal; import java.util.Date;  @Entity @Data @DynamicUpdate public class ProductInfo implements Serializable {  @Id  private String productId;   @NotNull  private BigDecimal productPrice;   @NotNull  @Min(0)  private Integer productStock;   private String productDescription, productIcon, productName;   @ColumnDefault("0")  private Integer productStatus, categoryType   @CreationTimestamp  private Date createTime, updateTime;   \*\*\* getter setter \*\*\* |

* 1. **Model ProductInOrder**

|  |
| --- |
| package me.zhulin.shopapi.entity; import com.fasterxml.jackson.annotation.JsonIgnore; import lombok.Data; import lombok.NoArgsConstructor; import javax.persistence.\*; import javax.validation.constraints.Min; import javax.validation.constraints.NotEmpty; import javax.validation.constraints.NotNull; import java.io.Serializable; import java.math.BigDecimal; import java.util.Objects;  @Entity @Data @NoArgsConstructor public class ProductInOrder {  @Id  @GeneratedValue(strategy = GenerationType.*AUTO*)  private Long id;   @ManyToOne(fetch = FetchType.*LAZY*, cascade = CascadeType.*REMOVE*)   @JsonIgnore  private Cart cart;   @ManyToOne(fetch = FetchType.*LAZY*)  @JoinColumn(name = "order\_id")  @JsonIgnore  private OrderMain orderMain;   @NotEmpty  private String productId, productName, productDescription, productIcon;   @NotNull  private Integer categoryType;   @NotNull  private BigDecimal productPrice;   @Min(0)  private Integer productStock;   @Min(1)  private Integer count;   public ProductInOrder(ProductInfo productInfo, Integer quantity) {  this.productId = productInfo.getProductId();  this.productName = productInfo.getProductName();  this.productDescription = productInfo.getProductDescription();  this.productIcon = productInfo.getProductIcon();  this.categoryType = productInfo.getCategoryType();  this.productPrice = productInfo.getProductPrice();  this.productStock = productInfo.getProductStock();  this.count = quantity;  }  @Override  public String toString() {  return "ProductInOrder{" +  "id=" + id +  ", productId='" + productId + '\'' +  ", productName='" + productName + '\'' +  ", productDescription='" + productDescription + '\'' +  ", productIcon='" + productIcon + '\'' +  ", categoryType=" + categoryType +  ", productPrice=" + productPrice +  ", productStock=" + productStock +  ", count=" + count +  '}';  }  @Override  public boolean equals(Object o) {  if (this == o) return true;  if (o == null || getClass() != o.getClass()) return false;  if (!super.equals(o)) return false;  ProductInOrder that = (ProductInOrder) o;  return Objects.*equals*(id, that.id) &&  Objects.*equals*(productId, that.productId) &&  Objects.*equals*(productName, that.productName) &&  Objects.*equals*(productDescription, that.productDescription) &&  Objects.*equals*(productIcon, that.productIcon) &&  Objects.*equals*(categoryType, that.categoryType) &&  Objects.*equals*(productPrice, that.productPrice);  }  @Override  public int hashCode() {   return Objects.*hash*(super.hashCode(), id, productId, productName, productDescription, productIcon, categoryType, productPrice);  }  \*\*\* getter setter \*\*\* |

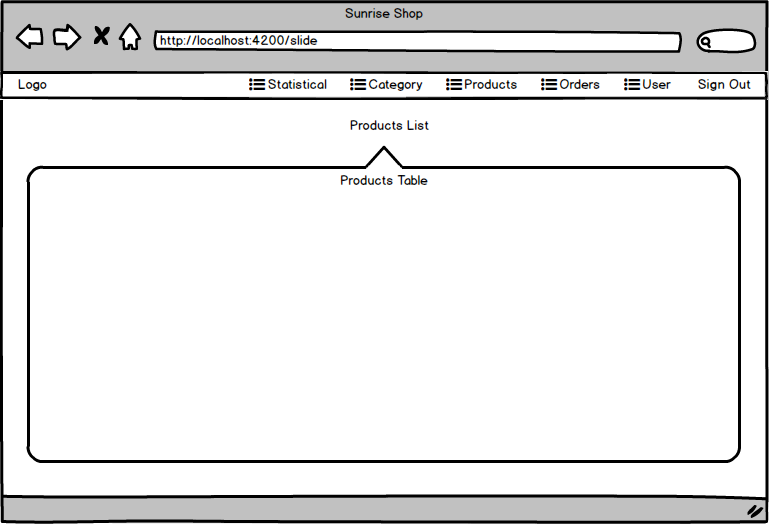
* 1. **Model Order main**

|  |
| --- |
| package me.zhulin.shopapi.entity;  import lombok.Data; import lombok.NoArgsConstructor; import org.hibernate.annotations.ColumnDefault; import org.hibernate.annotations.CreationTimestamp; import org.hibernate.annotations.DynamicUpdate; import org.hibernate.annotations.UpdateTimestamp; import javax.persistence.\*; import javax.validation.constraints.NotEmpty; import javax.validation.constraints.NotNull; import java.io.Serializable; import java.math.BigDecimal; import java.time.LocalDateTime; import java.util.HashSet; import java.util.Set;  @Entity @Data @NoArgsConstructor @DynamicUpdate public class OrderMain implements Serializable {  private static final long *serialVersionUID* = -3819883511505235030L;   @Id  @NotNull  @GeneratedValue(strategy = GenerationType.*AUTO*)  private Long orderId;   @OneToMany(cascade = CascadeType.*ALL*,  fetch = FetchType.*LAZY*,  mappedBy = "orderMain")  private Set<ProductInOrder> products = new HashSet<>();   @NotEmpty  private String buyerEmail, buyerName, buyerPhone, buyerAddress;   @NotNull  private BigDecimal orderAmount;   @NotNull  @ColumnDefault("0")  private Integer orderStatus;   @CreationTimestamp  private LocalDateTime createTime, updateTime;   public OrderMain(User buyer) {  this.buyerEmail = buyer.getEmail();  this.buyerName = buyer.getName();  this.buyerPhone = buyer.getPhone();  this.buyerAddress = buyer.getAddress();  this.orderAmount = buyer.getCart().getProducts().stream().map(item -> item.getProductPrice().multiply(new BigDecimal(item.getCount())))  .reduce(BigDecimal::add)  .orElse(new BigDecimal(0));  this.orderStatus = 0;  } \*\*\* getter setter \*\*\* |

* 1. **Model Cart**

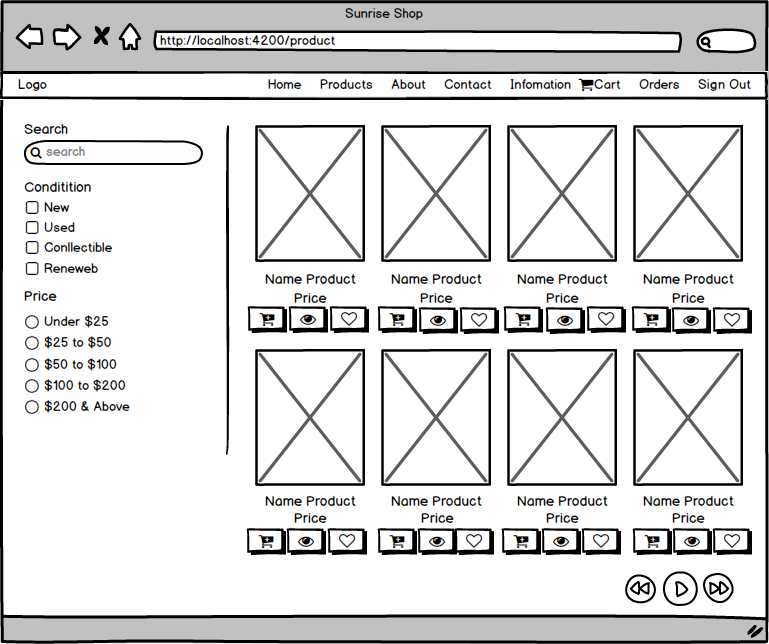
|  |
| --- |
| package me.zhulin.shopapi.entity; import com.fasterxml.jackson.annotation.JsonIgnore; import lombok.Data; import lombok.NoArgsConstructor; import javax.persistence.\*; import javax.validation.constraints.NotNull; import java.io.Serializable; import java.util.HashSet; import java.util.Set;  @Data @Entity @NoArgsConstructor public class Cart implements Serializable {  @Id  @NotNull  @GeneratedValue(strategy = GenerationType.*AUTO*)  private long cartId;   @OneToOne(fetch = FetchType.*LAZY*)  @MapsId  @JsonIgnore  private User user;  @OneToMany(cascade = CascadeType.*ALL*,  fetch = FetchType.*LAZY*, orphanRemoval = true,  mappedBy = "cart")  private Set<ProductInOrder> products = new HashSet<>();   @Override  public String toString() {  return "Cart{" + "cartId=" + cartId + ", products=" + products + '}';  }  \*\*\* getter setter \*\*\* |

1. **Thiết kế giao diện**
   1. **Phác thảo giao diện Admin**

****

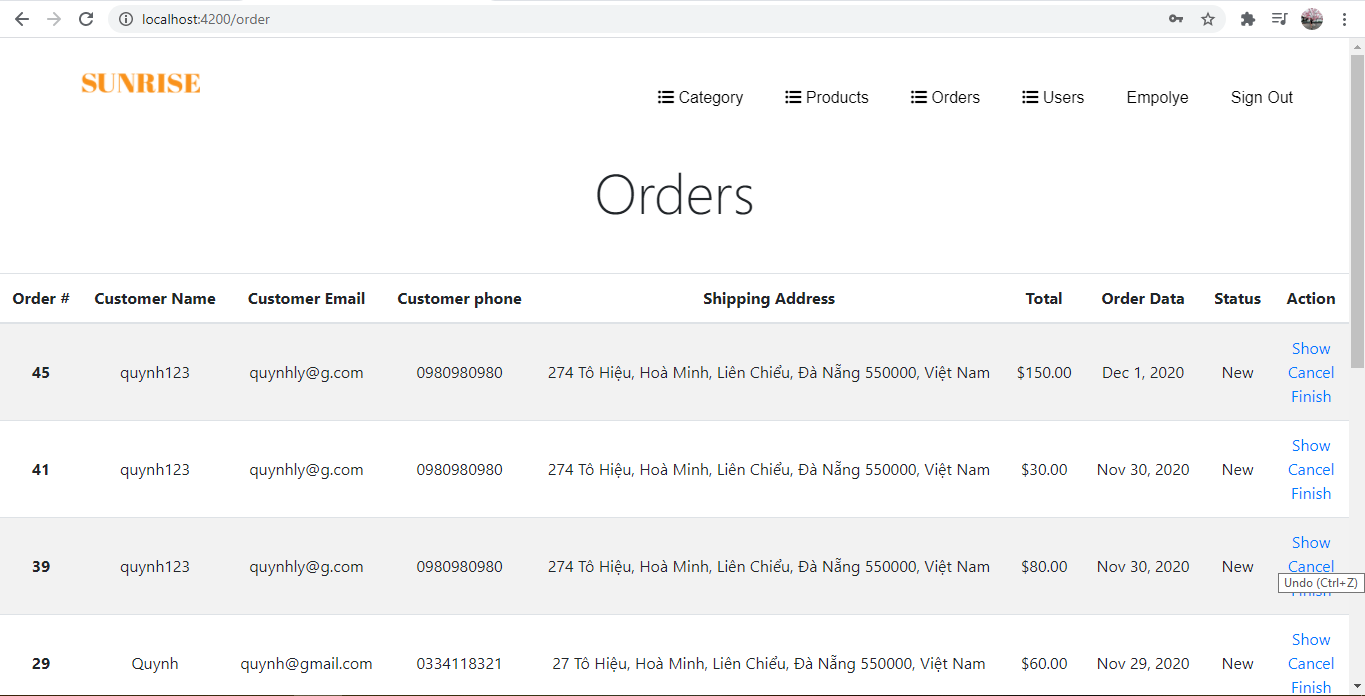
Hình 13: Mockup giao diện Admin

* 1. **Phác thảo giao diện người dùng (Products)**

****

Hình 14: Mockup giao diện người dùng

* 1. **Giao diện dành cho Admin**



Hình 15: Giao diện dành cho Admin

Giao diện quản lý Admin gồm có:

Logo của doanh nghiệp

Hiển thị thông tin của Admin

Chức năng quản lý tài khoản user

Chức năng quản lý loại sản phẩm

Chức năng quản lý sản phẩm

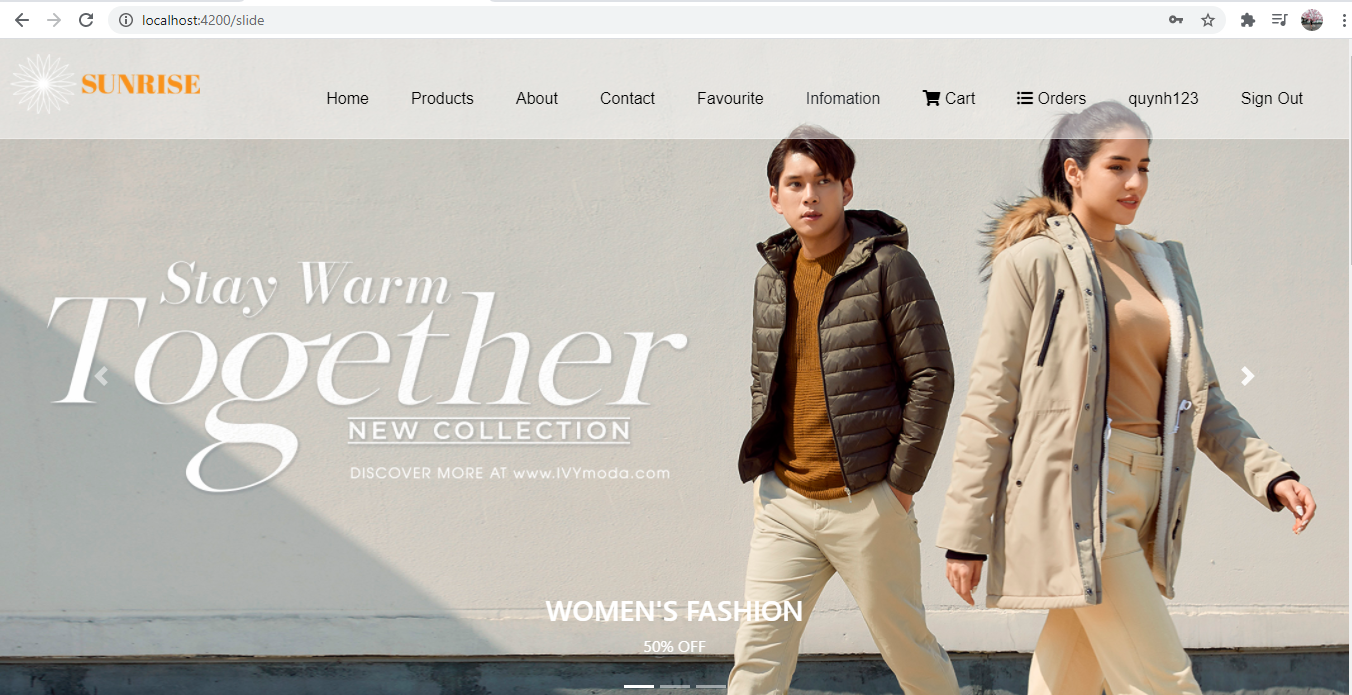
Quản lý bình luận của khách hàng

Quản lý trang thái đơn hàng

Quản lý thống kê đơn hàng, thống kê doanh thu

Mặc định sau khi Admin đăng nhập sẽ vào trang quản lý đơn hàng

* 1. **Giao diện dành cho người dùng**



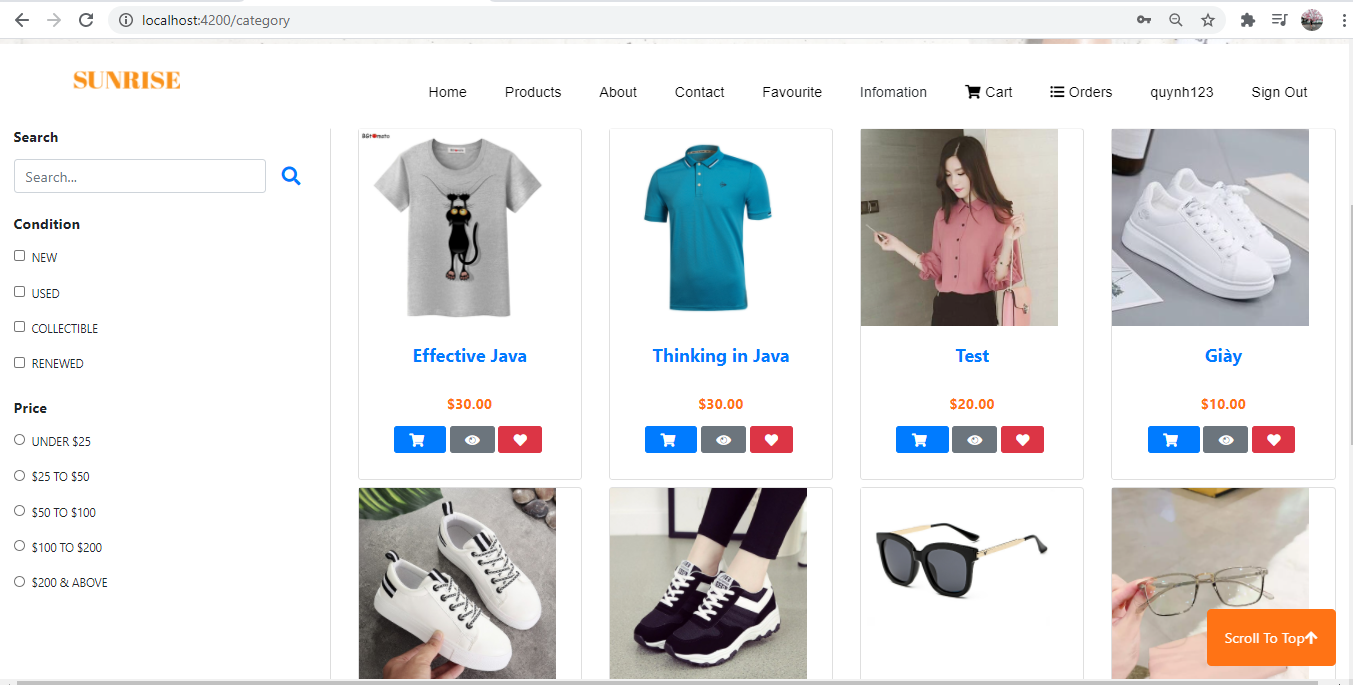
Hình 16: Giao diện dành cho người dùng

Giao diện sản phẩm gồm có:

Logo của doanh nghiệp

Thanh menu hiển thị các mục: trang chủ, trang sản phẩm, trang giới thiệu, trang liên hệ, trang thông tin, trang sản phẩm yêu thích, trang giỏ hàng, trang đăng nhập và chức năng đa ngôn ngữ.

* 1. **Giao diện hiển thị products**



Hình 17: Giao diện hiển thị products

Trang hiển sản phẩm: hiển thị tất cả các sản phẩm, hiển thị sản phẩm theo loại sản phẩm, phân trang sản phẩm.

Có các chức năng, thêm sản phẩm vào giỏ hàng, xem chi tiết sản phẩm, thêm sản phẩm yêu thích.

1. **THỰC HIỆN VIẾT MÃ CHO TỪNG CHỨC NĂNG**
   1. **Triển khai hệ thống Rest-API**



RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

* HTTP Request

REST hoạt động chủ yếu dựa vào giao thức HTTP.

GET (SELECT): Trả về một Resource hoặc một danh sách Resource.

POST (CREATE): Tạo mới một Resource.

PUT (UPDATE): Cập nhật thông tin cho Resource.

DELETE (DELETE): Xoá một Resource.

CONNECT: thiết lập một kết nối tới server theo URI.

OPTIONS: mô tả các tùy chọn giao tiếp cho resource.

TRACE: thực hiện một bài test loop – back theo đường dẫn đến resource.

* Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.
* RESTful API không sử dụng session và cookie.
* Status code:
* Khi chúng ta request một API nào đó thường thì sẽ có vài status code để nhận biết sau:

200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.

201 Created – Trả về khi một Resouce vừa được tạo thành công.

204 No Content – Trả về khi Resource xoá thành công.

304 Not Modified – Client có thể sử dụng dữ liệu cache.

400 Bad Request – Request không hợp lệ

401 Unauthorized – Request cần có auth.

403 Forbidden – bị từ chối không cho phép.

404 Not Found – Không tìm thấy resource từ URI

405 Method Not Allowed – Phương thức không cho phép với user hiện tại.

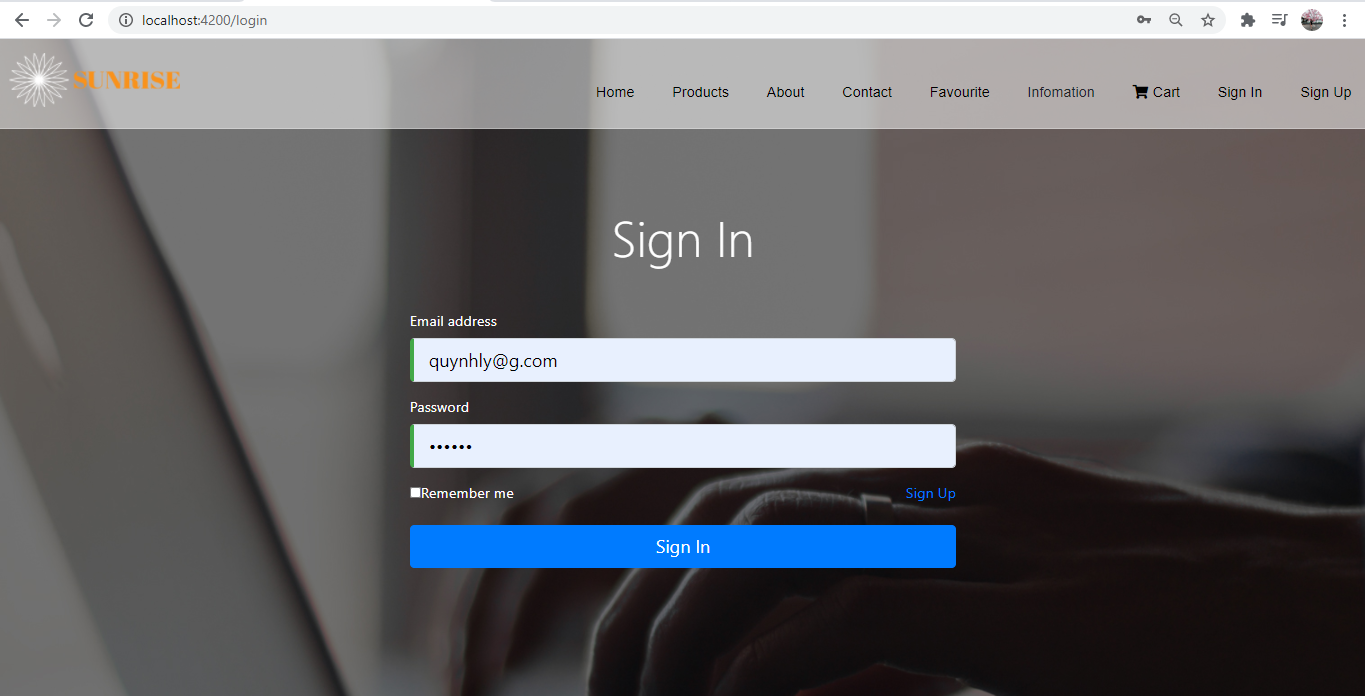
410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.

415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.

422 Unprocessable Entity – Dữ liệu không được xác thực

429 Too Many Requests – Request bị từ chối do bị giới hạn

* 1. **Chức năng đăng nhập**



Hình 18: Giao diện trang đăng nhập

Người dùng không cần đăng nhập vẫn có thể truy cập vào trang chủ của trang web, có thể sử dụng các chức năng xem sản phẩm, thêm sản phẩm yêu thích, thêm sản phẩm vào giỏ hàng, xem chi tiết và bình luận về sản phẩm.

Nếu người dùng muốn thực hiện mua sản phẩm thì bắt buộc người dùng phải đăng nhập vào hệ thống, trang chủ sẽ điều hướng tới trang login

Chức năng đăng nhập cho phép người dùng nhập email và mật khẩu đã đăng ký, ứng với 2 trường là Email và Password.

Yêu cầu người dùng nhập mật khẩu tối đa 8 ký tự, email phải được nhập đúng cú pháp.

Trường email và mật khẩu không được để trống.

Nếu nhập sai email hoặc mật khẩu đã đăng ký, ứng dụng sẽ hiển thị thông báo: “Email or password is invalid”.

* 1. **Chức năng đăng ký tài khoản**



Hình 19: Giao diện đăng ký tài khoản

Trong trường hợp người dùng chưa có tài khoản, người dùng có thể kích vào link Register here, hệ thống sẽ điều hướng tới form đăng ký tài khoản.

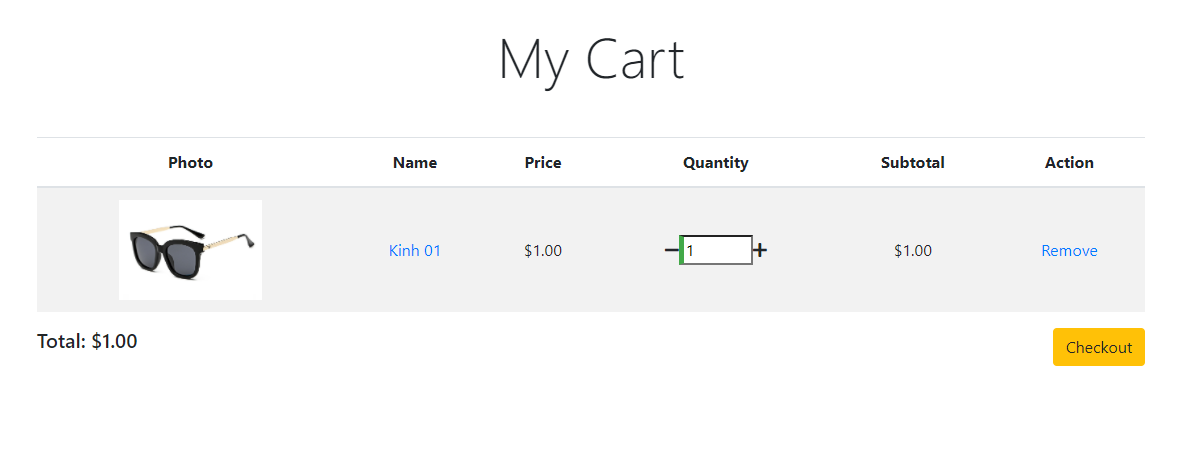
Form đăng ký với các trường bắt buộc: Your name, Phone Number, Your email, Password, Repassword.

Các trường trong form đăng ký không được bỏ trống.

Nếu trường không trùng với trường Password, hệ thống sẽ hiển thị thông báo yêu cầu người dùng phải nhập lại Repassword.

Sau khi người dùng đăng ký thành công, ứng dụng sẽ điều hướng tới trang login, để người dùng đăng nhập vào website lại từ đầu.

* 1. **Chức năng thêm sản phẩm vào giỏ hàng**



Hình 20: Giao diện trang giỏ hàng

Trong trường hợp người dùng đã lựa được sản phẩm ưng ý và thêm sản phẩm vào giỏ hàng.

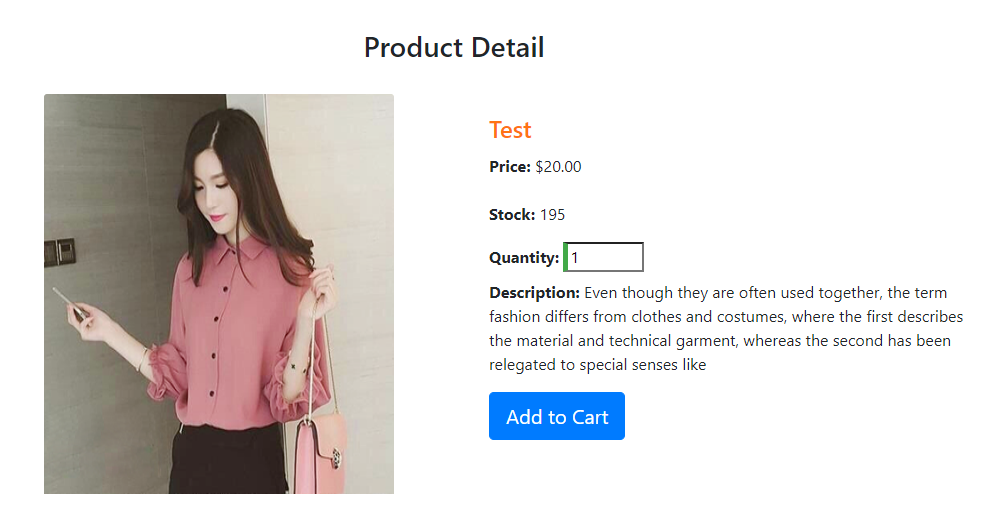
Trang giỏ hàng sẽ hiển thị hình ảnh sản phẩm, tên sản phẩm, giá tiền, số lượng và tổng tiền, action.

Nếu khách hàng muốn chọn mua thêm sản phẩm, có thể kích quay lại trang Products để lựa chọn thêm sản phẩm.

Khách hàng có thể điều chỉnh lại số lượng sản phẩm và tổng tiền trước khi tiến hành đặt hàng.

Khi khách hàng muốn đặt hàng, có thể kích vào nút Checkout để thực hiện bước tiếp theo.

* 1. **Chức năng hiển thị chi tiết sản phẩm**



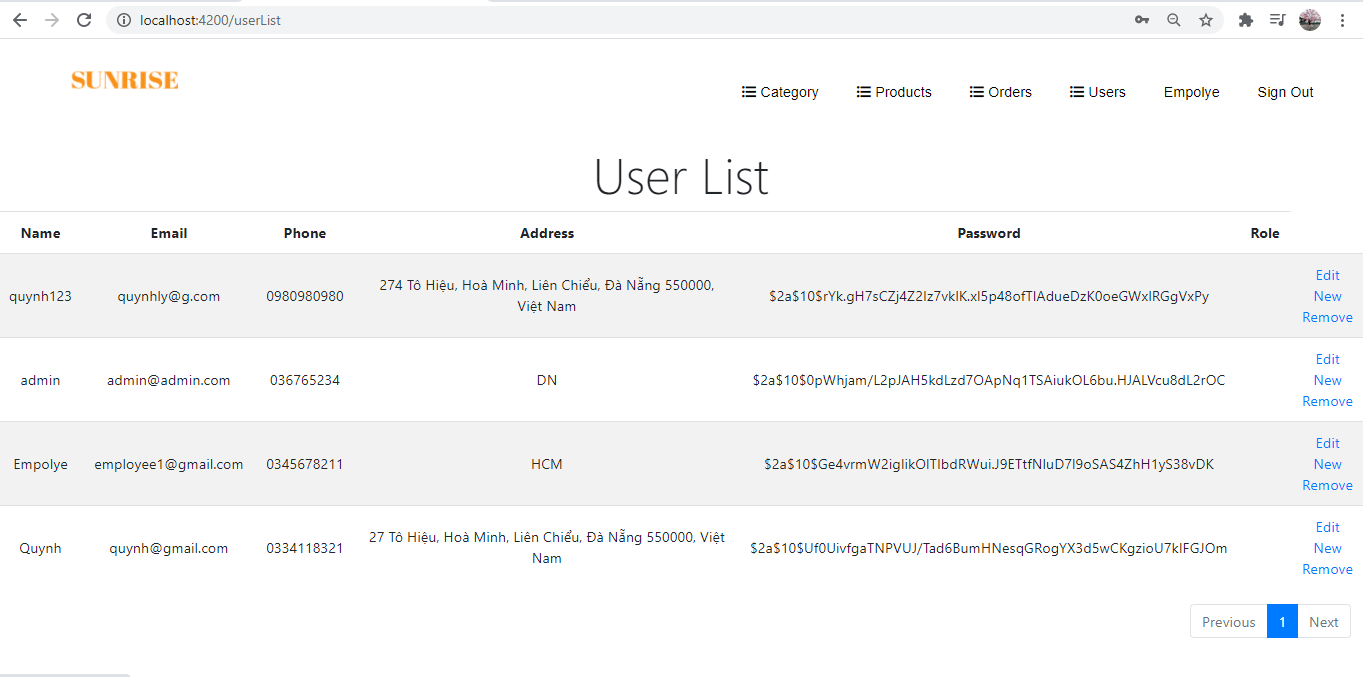
Hình 21: Giao diện trang chi tiết sản phẩm

Trang chi tiết sản phẩm, giúp cho người dùng biết được nhiều thông tin về sản phẩm hơn.

Ở trang chi tiết sản phẩm sẽ hiển thị: Hình ảnh sản phẩm, tên sản phẩm, mô tả sản phẩm, giá tiền, số lượng hàng còn, và các thông tin liên quan.

Trang chi tiết sản phẩm có chức năng thêm sản phẩm vào giỏ hàng và thêm sản phẩm yêu thích.

* 1. **Chức năng quản lý tài khoản**



Hình 22: Giao diện trang quản lý tài khoản

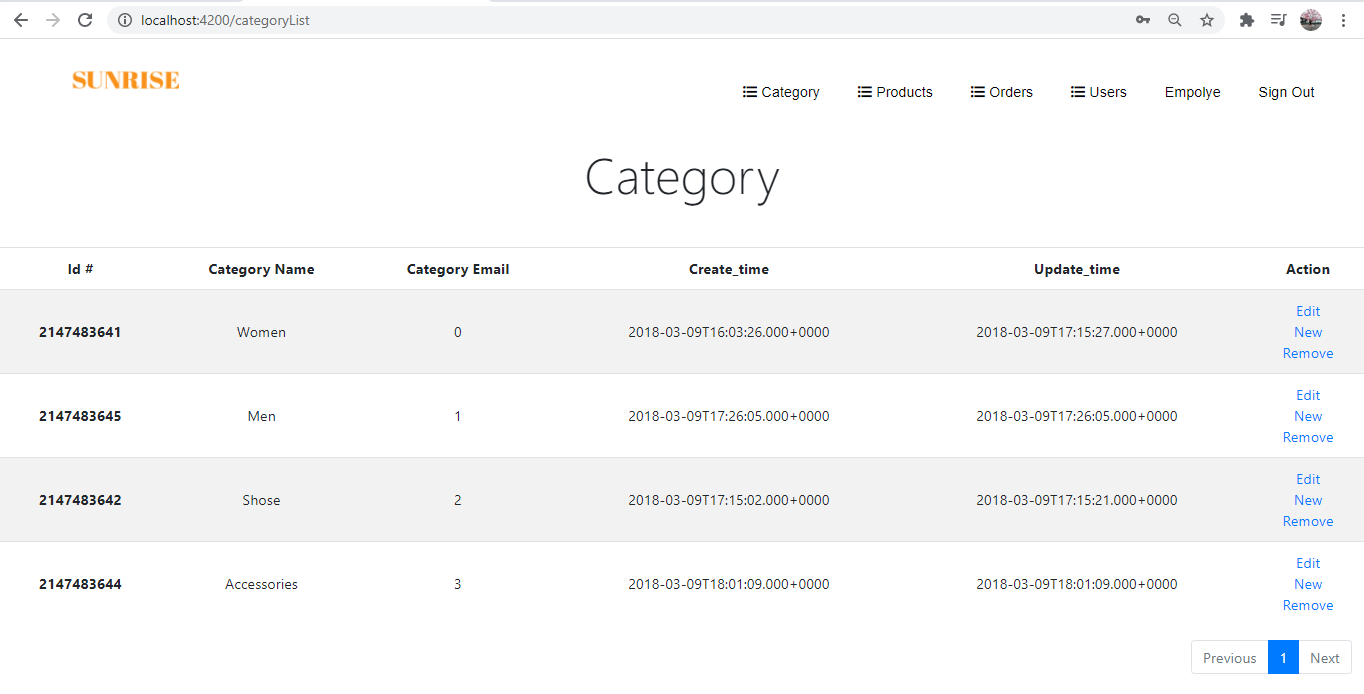
Trang quản lý tài khoản hiển thị danh sách tất cả users với các trường: name, email, phone, address, password, role.

Tại trang này nhân viên hoặc người quản trị viên có quyền thêm mới, xóa, cập nhật đề nhân viên hoặc người quản trị viên có thể dễ dàng quản lý tài khoản người dùng.

Nhân viên hoặc người quản trị viên có thể kích vào New để thêm mới 1 tài khoản.

Tương tự, nhân viên hoặc quản trị viên có thể chỉnh sửa 1 tài khoản khi kích vào edit ở cột Action và ấn vào Remove để xóa tài khoản nếu cần.

* 1. **Chức năng quản lý loại sản phẩm**



Hình 23: Giao diện trang quản lý loại sản phẩm

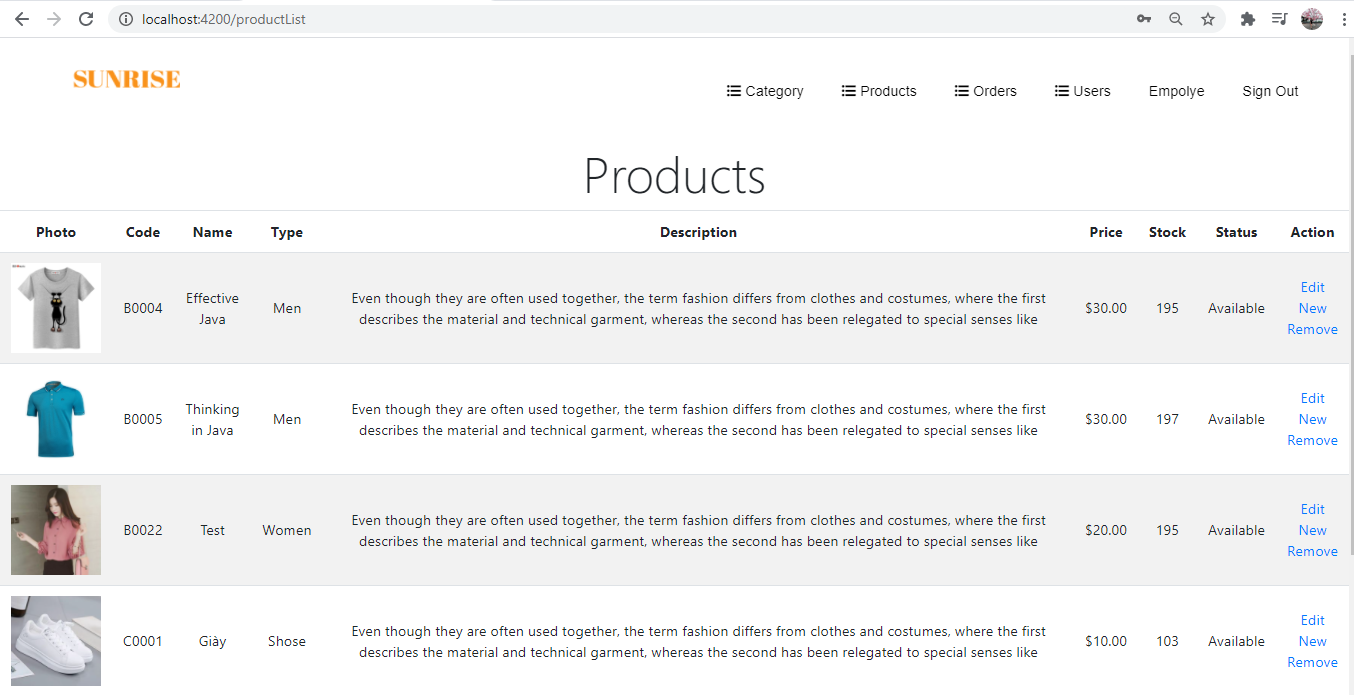
Trang quản lý loại sản phẩm hiển thị danh sách tất cả loại sản phẩm với các trường: id, category\_name, category\_type, create\_time, update\_time, action.

Tại trang này nhân viên hoặc người quản trị viên có quyền thêm mới, xóa, cập nhật đề nhân viên hoặc người quản trị viên có thể dễ dàng quản lý loại sản phẩm.

Nhân viên hoặc người quản trị viên có thể kích vào New để thêm mới 1 loại sản phẩm.

Tương tự, nhân viên hoặc quản trị viên có thể chỉnh sửa 1 loại sản phẩm khi kích vào edit ở cột Action và ấn vào Remove để xóa loại sản phẩm nếu cần.

* 1. **Chức năng quản lý sản phẩm**



Hình 24: Giao diện trang quản lý sản phẩm

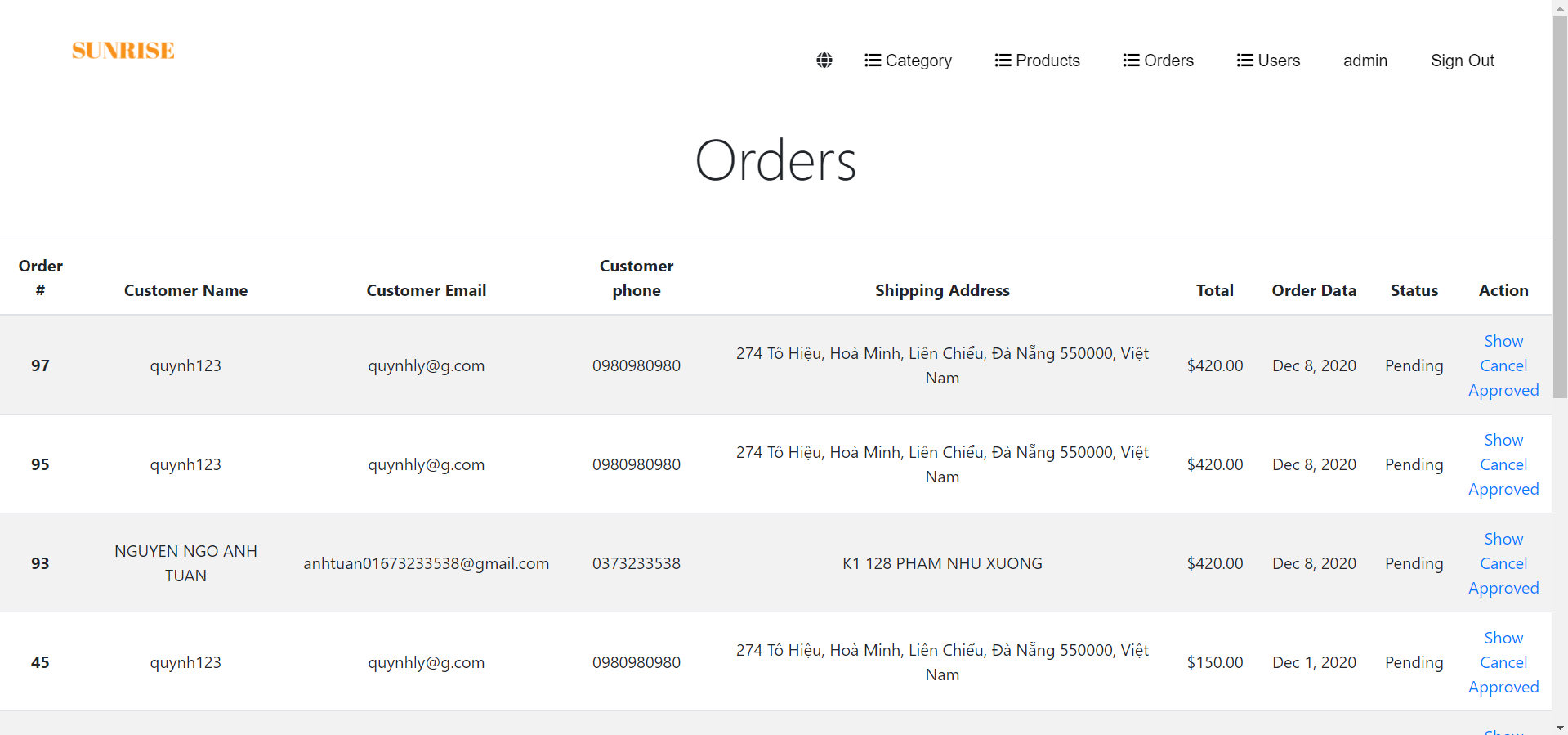
Trang quản lý sản phẩm hiển thị danh sách tất cả sản phẩm với các trường: photo, code, name, type,desciption, price, stock, status, action.

Tại trang này có các chức năng: thêm mới, xóa, cập nhật đề nhân viên và quản trị viên có thể dễ dàng quản lý sản phẩm.

Nhân viên hoặc người quản trị viên có thể kích vào New để thêm mới 1 sản phẩm.

Tương tự, nhân viên hoặc quản trị viên có thể chỉnh sửa 1 sản phẩm khi kích vào edit ở cột Action và ấn vào Remove để xóa sản phẩm nếu cần.

* 1. **Chức năng quản lý đơn hàng**



Hình 25: Giao diện trang quản lý trạng thái đơn hàng

Trang quản lý đơn hàng hiển thị danh sách tất cả các đơn hàng với các trường: id, customer name, customer email, customer phone, address, total, order date, status, action.

Tại trang này nhân viên hoặc người quản trị viên có quyền xem chi tiết đơn hàng mà khách hàng đặt khi ấn vào nút Show.

Có các action: Pending, Approved, Finish, Cancel để nhân viên hoặc người quản trị viên có thể cập nhật trạng thái đơn hàng theo từng giai đoạn.

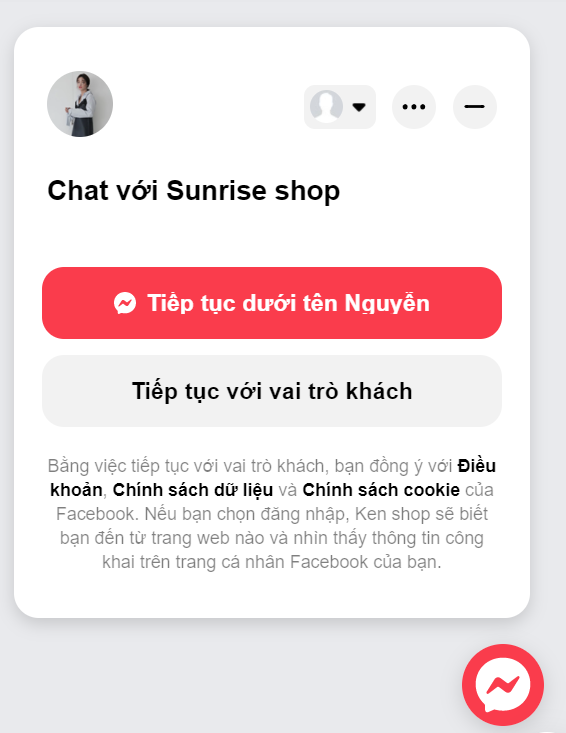
Khi nhân viên hoặc người quản trị viên nhận được đơn hàng của khách hàng, đơn hàng sẽ được hiển thị ở trạng thái Pending.

Sau khi nhân viên gọi điện xác nhận đơn hàng, đơn hàng đó sẽ được chuyển qua trạng thái Approved.

Khi khách hàng xác nhận đăt đơn hàng và bên cửa hàng vận chuyển đơn hàng đi, đơn hàng sẽ được chuyển qua trạng thái Finish.

Trường hợp nhân viên gọi điện xác nhận đơn hàng không được, hoặc khách hàng yêu cầu hủy đơn, nhân viên có thể ấn vào Cancel để xóa đơn hàng đó.

* 1. **Chức năng Facebook chat**

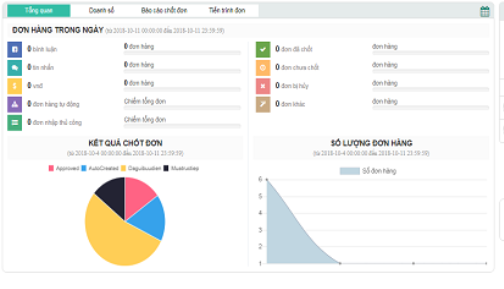


Hình 26: Giao diện Facebook chat

Người dùng cần phải đăng nhập bằng Facebook để chat trực tuyến với nhân viên giải đáp thắc mắc và hỗ trợ đặt sản phẩm.

Nếu người dùng không đăng nhập bằng Facebook thì có thể chat với nhân viên với vai trò khách thăm quan.

* 1. **Chức năng quản lý thống kê**



Hình 27: Giao diện quản lý thống kê

1. **KẾT LUẬN**
2. **Những điểm đã làm được trong dự án:**

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và thực hiện triển khai dự án, đề tài đã đạt được những kết quả sau:

Qua đề tài dự án đã cải thiện và củng cố kiến thức về cách viết tài liệu báo cáo, kỹ năng thiết kế sơ đồ usecase, sơ đồ hoạt động và xây dựng hệ thống theo hướng RESTful API.

Trên thực tế, đề tài website thời trang Sunrise đã được xây dựng hoàn chỉnh, với các chức năng:

* Đăng nhập, đăng ký và phân quyền
* Quản lý tài khoản nhân viên
* Thống kê doanh thu theo ngày, theo tháng
* Hiển thị sản phẩm và thông tin chi tiết sản phẩm
* Chức năng giỏ hàng và thanh toán

Với những gì mà dự án đạt được, website có thể được vận hành tốt trên Internet, dáp ứng tốt các yêu cầu cần có của một website thời trang.

1. **Những vấn đề chưa giải quyết được trong dự án:**

* Thanh toán trực tuyến
* Đánh giá 5 sao của khách hàng dành cho sản phẩm

1. **Những thuận lợi và khó khăn trong quá trình làm dự án:**

* Thuận lợi:
* Các thành viên trong nhóm có ý thức, tự giác trong công việc của mình và luôn giúp đỡ các thành viên khác
* Nhận được sự hướng dẫn nhiệt tình từ giáo viên hướng dẫn

1. **TÀI LIỆU THAM KHẢO**

* Các bài hướng dẫn mẫu làm báo cáo trên hệ thống LMS
* Tham khảo tài liệu trên website <https://stackoverflow.com>
* Tham khảo tài liệu trên website <https://www.w3schools.com>
* Tư liệu Spring Framework trên website <https://spring.io>
* Tài liệu trên website <https://o7planning.org>