**Danh sách các phần đã chỉnh sửa của nhóm:**

* Chỉnh sửa quan hệ <<include>> thành <<extend>> ở Usecase quản lý sản phẩm: mục xóa sản phẩm.
* Bổ sung phần liên kết với các đối tác, hóa đơn ở phần vẽ mô hình kiến trúc.
* Bổ sung thành phần kết nối đối tác, hóa đơn ở phần vẽ biểu đồ UML.

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI TP. HỒ CHÍ MINH**

**Khoa CNTT 2**

**🙠🕮🙢**

A picture containing logo

Description automatically generated

**BÁO CÁO**

**ĐỒ ÁN MÔN HỌC**

**Đề tài:**

**WEBSITE KINH DOANH CÁC MẶT HÀNG THỜI TRANG**

**Môn học** : Kiến trúc và thiết kế phần mềm

**Giảng viên** : Dương Thanh Thảo

**Nhóm** : 2

**Danh sách**

1. Phạm Dung Bắc N19DCCN019
2. Lê Bá Minh Nhật N19DCCN128
3. Hoàng Hoài Nam N19DCCN114
4. Nguyễn Quốc Anh N19DCCN007
5. Chu Văn Mạnh N19DCCN105

*TP. Hồ Chí Minh, tháng 07 năm 2023*

**MỤC LỤC**

[**I.** **Giới thiệu** 5](#_Toc139793470)

[1.1. Giới thiệu đề tài 5](#_Toc139793471)

[1.2. Mô tả yêu cầu nghiệp vụ 5](#_Toc139793472)

[**II.** **Biểu đồ usecase** 7](#_Toc139793473)

[2.1. Biểu đồ usecase tổng quát 7](#_Toc139793474)

[2.2. Usecase đăng nhập 7](#_Toc139793475)

[2.3. Usecase Đăng ký 9](#_Toc139793476)

[2.4. Usecase chỉnh sửa thông tin cá nhân 11](#_Toc139793477)

[2.5. Usecase Quên mật khẩu 13](#_Toc139793478)

[2.6. Usecase tìm kiếm 15](#_Toc139793479)

[2.7. Usecase xem chi tiết sản phẩm 16](#_Toc139793480)

[2.8. Usecase Đăng xuất 17](#_Toc139793481)

[2.9. Usecase đổi mật khẩu 19](#_Toc139793482)

[2.10. Usecase quản lý giỏ hàng 21](#_Toc139793483)

[2.11. Usecase quản lý địa chỉ 23](#_Toc139793484)

[2.12. Usecase Đặt hàng 25](#_Toc139793485)

[2.13. Usecase quản lý đánh giá 27](#_Toc139793486)

[2.14. Usecase theo dõi đơn hàng 29](#_Toc139793487)

[2.15. Usecase quản lý sản phẩm 31](#_Toc139793488)

[2.16. Usecase quản lý đơn hàng 33](#_Toc139793489)

[2.17. Usecase quản lý nhãn hiệu 34](#_Toc139793490)

[2.18. Usecase quản lý loại sản phẩm 36](#_Toc139793491)

[2.19. Usecase quản lý chương trình khuyến mãi 38](#_Toc139793492)

[2.20. Usecase quản lý phiếu nhập 39](#_Toc139793493)

[2.21. Usecase quản lý phiếu xuất 41](#_Toc139793494)

[2.22. Usecase quản lý tồn kho 43](#_Toc139793495)

[2.23. Usecase quản lý người dùng 44](#_Toc139793496)

[2.24. Usecase thống kê 46](#_Toc139793497)

[**III.** **Thiết kế kiến trúc.** 48](#_Toc139793498)

[3.1. Kiến trúc lựa chọn 48](#_Toc139793499)

[3.2. Lý do chọn kiến trúc. 49](#_Toc139793500)

[3.3. Vẽ biểu đồ kiểu kiến trúc, mô hình kiến trúc được chọn: 49](#_Toc139793501)

[**IV.** **Công nghệ sử dụng** 50](#_Toc139793502)

[4.1. Về ngôn ngữ lập trình: 50](#_Toc139793503)

[4.2. Hệ quản trị cơ sở dữ liệu: 51](#_Toc139793504)

[4.3. Công cụ lập trình: 52](#_Toc139793505)

[4.4. Công nghệ lập trình phía server: 54](#_Toc139793506)

[4.5. Công nghệ lập trình phía client: 54](#_Toc139793507)

[4.6. Công nghệ quản lý source code: 55](#_Toc139793508)

[**V.** **Mô hình thành phần** 56](#_Toc139793509)

[**VI.** **Các design pattern sử dụng và demo chương trình** 58](#_Toc139793510)

[6.1. Các design pattern được sử dụng: 58](#_Toc139793511)

[6.2. Demo chương trình 63](#_Toc139793512)

1. **Giới thiệu** 
   1. Giới thiệu đề tài

Ngày nay, cùng với sự phát triển của đất nước ngành Công nghệ thông tin đã có những bước phát triển mạnh mẽ không ngừng và tin học đã trở thành chiếc chìa khóa dẫn đến thành công cho nhiều cá nhân trong nhiều lĩnh vực, hoạt động. Với những ứng dụng của mình, ngành Công nghệ thông tin đã góp phần mang lại nhiều lợi ích mà không ai có thể phủ nhận được. Đặc biệt là trong lĩnh vực quản lý kinh tế, tin học đã góp phần tạo ra sự thay đổi nhanh chóng cho bộ mặt xã hội. Nhất là khi việc tin học hóa vào công tác quản lý, bán hàng là một trong những yêu cầu cần thiết đối với các doanh nghiệp hiện nay.

Từ những phần mềm quản lý giúp cho công việc bán hàng của cửa hàng trở nên nhanh chóng và dễ dàng, ngày nay công nghệ thông tin đã phát triển mạnh mẽ hơn nữa, Website ra đời không chỉ đáp ứng cho các khách hàng đến trực tiếp cửa hàng mua sản phẩm mà nó còn phục vụ cho những khách hàng ở xa. Không những thế, việc giới thiệu, quảng bá hình ảnh về cửa hàng được mở rộng trên quy mô lớn giúp cho việc kinh doanh, buôn bán, trao đổi tin tức thuận lợi hơn rất nhiều, mang lại hiệu quả kinh tế cao hơn và tạo uy tín trong lòng khách hàng.

Chính vì lợi ích mà Website mang lại mà các cửa hàng đã áp dụng nó để phát triển công việc kinh doanh và giới thiệu cửa hàng của mình. Trong số đó, thời trang cũng là một ngành sử dụng nhiều đến Website. Do vậy, nhóm đã nghiên cứu kiến thức từ nhiều nguồn trên Internet và học phần Kiến trúc và thiết kế phần mềm của cô Dương Thanh Thảo, nhóm đã xây dựng nên hệ thống “**Website bán mặt hàng thời trang**” với ưu điểm thân thiện với người dùng và đáp ứng được các chức năng cơ bản của một website bán hàng.

Các phần tiếp theo sau đây là báo cáo của nhóm về bản thiết kế hệ thống Website bán mặt hàng thời trang, qua đó, nhóm em mong nhận được sự đánh giá và góp ý từ cô Dương Thanh Thảo.

* 1. Mô tả yêu cầu nghiệp vụ

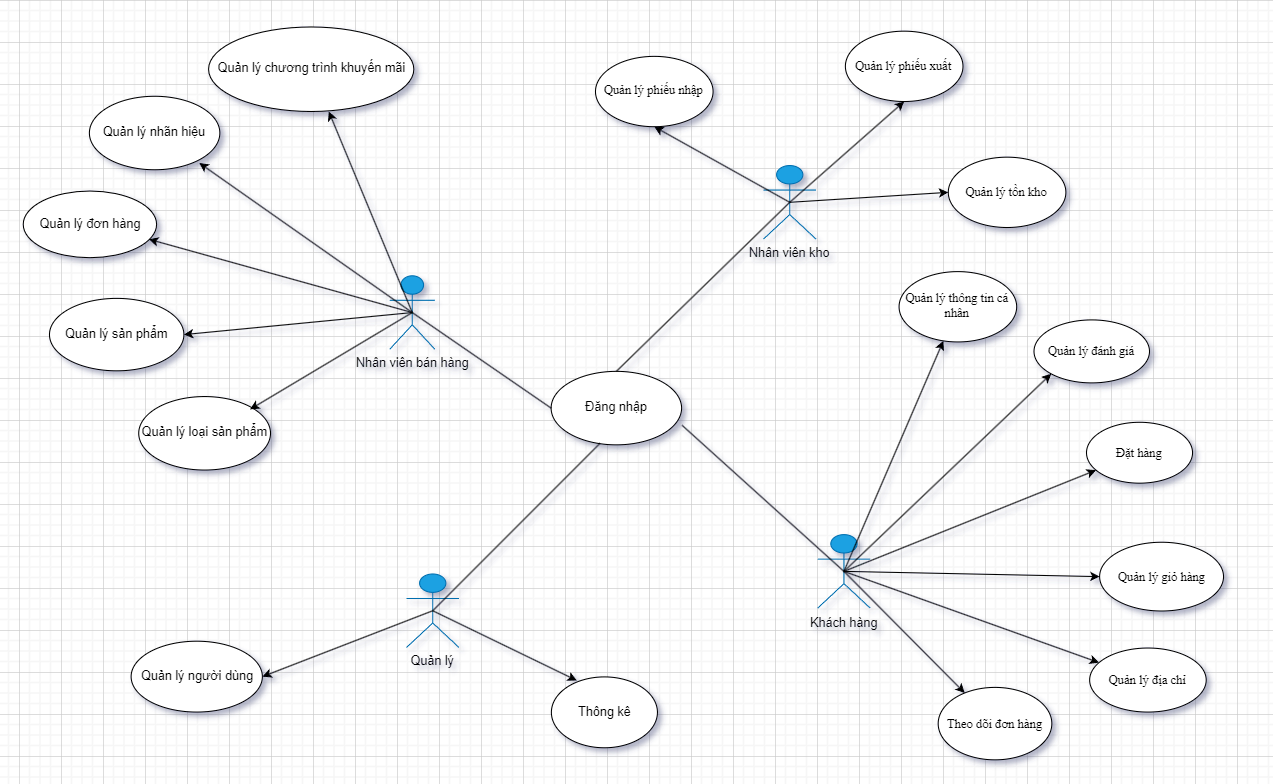
Khi truy cập vào hệ thống website kinh doanh các mặt hàng thời trang, hệ thống cho phép tra cứu, xem chi tiết sản phẩm mà không cần đăng nhập. Nếu muốn sử dụng các chức năng khác của hệ thống theo phân quyền của tài khoản thì yêu cầu người dùng phải đăng nhập vào hệ thống trước, nếu không thì phải đăng ký tài khoản.

Hệ thống gồm 2 site: site khách hàng và site quản trị trang web.

* + 1. Site quản trị trang web
* Quản lý sau khi đăng nhập có thể sử dụng các chức năng:
  + Quản lý người dùng gồm:
    - * Phân quyền tài khoản người dùng.
      * Xoá tài khoản
  + Lập báo cáo thống kê: thống kê doanh thu theo tuần, tháng, năm, thống kê sản phẩm bán chạy, thống kê đơn hàng, thống kê người dùng.
* Nhân viên bán hàng sau khi đăng nhập có thể sử dụng các chức năng:
  + - Quản lý sản phẩm gồm:
      * Thêm sản phẩm mới
      * Cập nhập thông tin sản phẩm nếu có thay đổi.
      * Xoá sản phẩm.
      * Tìm kiếm sản phẩm.
    - Quản lý nhãn hiệu gồm:
      * Thêm mới nhãn hiệu.
      * Chỉnh sửa nhãn hiệu.
      * Xoá nhãn hiệu.
      * Tìm kiếm nhãn hiệu.
    - Quản lý loại sản phẩm gồm:
      * Thêm mới loại sản phẩm.
      * Chỉnh sửa loại sản phẩm.
      * Thêm mới loại sản phẩm.
      * Tìm kiếm loại sản phẩm.
    - Quản lý đơn hàng gồm:
      * Xem chi tiết đơn hàng.
      * Duyệt đơn hàng.
      * Tìm kiếm đơn hàng.
    - Quản lý chương trình khuyến mãi gồm:
      * Thêm chương trình khuyến mãi
      * Xóa chương trình khuyến mãi
      * Chỉnh sửa chương trình khuyến mãi
      * Tìm kiếm chương trình khuyến mãi
    - Nhận thông tin đánh giá sản phẩm từ khách hàng và phản hồi lại khách hàng khi có yêu cầu.
* Nhân viên kho sau khi đăng nhập có thể sử dụng các chức năng:
  + Tạo phiếu nhập hàng khi hàng hóa được nhập về.
  + Tạo phiếu xuất hàng tương ứng với các đơn hàng được đặt.
  + Quản lý tồn kho các sản phẩm.

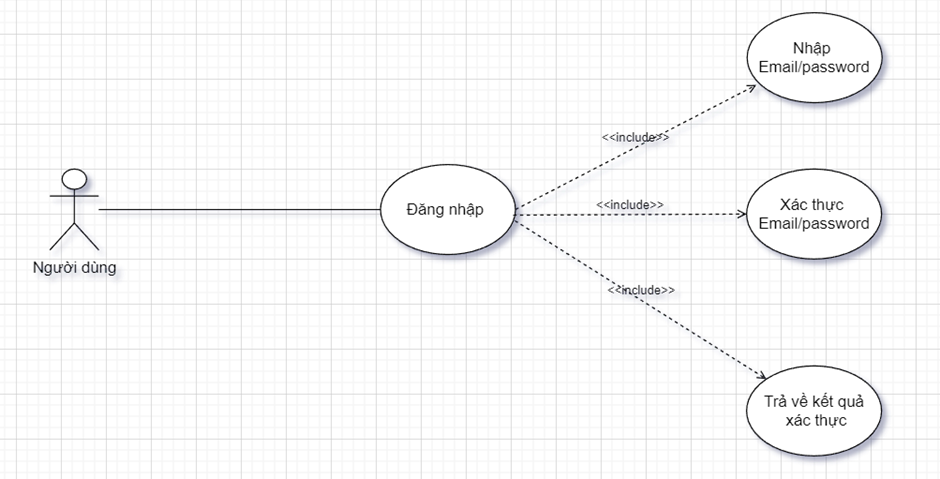
* + 1. Site khách hàng:
* Khách hàng gồm khách hàng vãng lai và khách hàng đã đăng ký tài khoản có thể sử dụng các chức năng sau:
  + - Tìm kiếm, xem danh sách và chi tiết sản phẩm.
* Khách hàng đã đăng ký tài khoản có thêm các chức năng sau:
  + Đổi mật khẩu khi có nhu cầu, quên mật khẩu.
  + Quản lý thông tin cá nhân như số điện thoại, họ tên ,…
  + Khi đặt hàng, khách hàng có thể chọn phương thức nhận hàng như loại vận chuyển, thời gian nhận hàng, địa chỉ nhận hàng ,…
  + Đánh giá sản phẩm sau khi đã nhận sản phẩm và thanh toán thành công.
  + Theo dõi trạng thái đơn hàng, yêu cầu hủy đơn hàng, đã nhận được hàng, xem chi tiết đơn hàng.
  + Quản lý giỏ hàng như thêm mới sản phẩm, chỉnh sửa, xóa sản phẩm khỏi giỏ hàng.
  + Tạo đơn hàng từ giỏ hàng.
  + Quản lý địa chỉ nhận hàng như thêm mới địa chỉ, xóa địa chỉ, chỉnh sửa địa chỉ đã có, đặt địa chỉ làm mặc định.

1. **Biểu đồ usecase**
   1. Biểu đồ usecase tổng quát



*Hình 2.1. Usecase tổng quát*

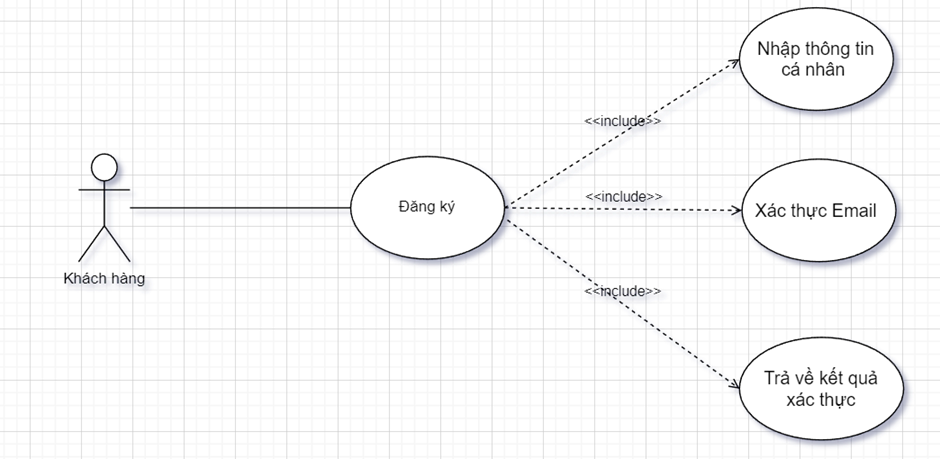
* 1. Usecase đăng nhập



*Hình 2.2. Usecase đăng nhập*

|  |  |
| --- | --- |
| Tên usecase | Đăng nhập |
| Actor | Người dùng. |
| Mô tả | Tác nhân sử dụng usecase này để thực hiện đăng nhập vào hệ thống |
| Điều kiện kích hoạt | Chọn chức năng đăng nhập ở hệ thống |
| Tiền điều kiện | Tác nhân chưa đăng nhập hệ thống. |
| Hậu điều kiện | Nếu thành công: Hệ thống sẽ hiển thị giao diện chính. Người dùng có thể thực hiện các chức năng theo đúng quyền hạn của mình.  Nếu thất bại: Hệ thống sẽ đưa ra thông báo “Thông tin đăng nhập không hợp lệ” và yêu cầu đăng nhập lại. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị giao diện đăng nhập  2. Người dùng nhập tài khoản bao gồm email và mật khẩu.  3. Hệ thống kiểm tra và xác nhận thông tin đăng nhập.  4. Hiển thị giao diện chính.  5. Kết thúc Use case. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Người dùng hủy yêu cầu đăng nhập.  2. Hệ thống trở về giao diện khi chưa đăng nhập.  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Người dùng nhập thông tin sai.  2. Hệ thống hiển thị thông báo lỗi.  3. Kết thúc Use case. |
| Các yêu cầu đặc biệt | Không có |

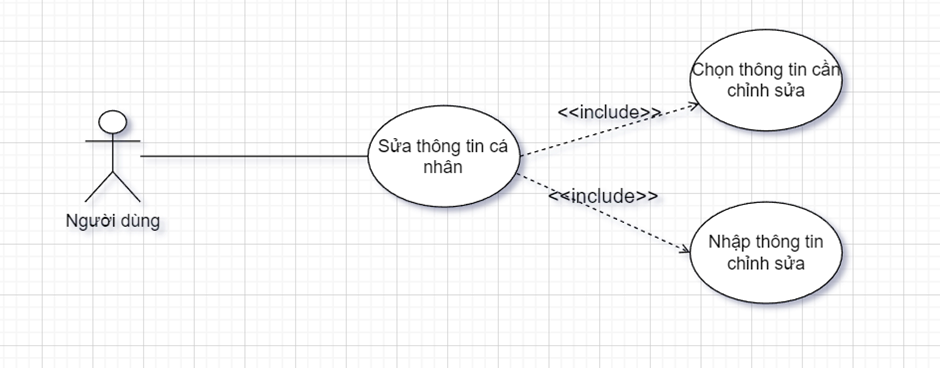
* 1. Usecase Đăng ký



*Hình 2.3. Usecase Đăng ký*

|  |  |
| --- | --- |
| Tên Usecase | Đăng ký |
| Actor | Khách hàng. |
| Mô tả | Tác nhân sử dụng use case này để thực hiện đăng ký tài khoản truy cập vào hệ thống. |
| Điều kiện kích hoạt | Chọn chức năng đăng ký ở hệ thống. |
| Tiền điều kiện | Tác nhân chưa đăng nhập hệ thống. |
| Hậu điều kiện | Nếu thành công: Hệ thống sẽ hiển thị giao diện đăng ký thành công. Người dùng có thể thực hiện chức năng đăng nhập để đăng nhập vào hệ thống.  Nếu thất bại: Hệ thống sẽ đưa ra thông báo “Thông tin đăng ký không hợp lệ” . |
| Luồng sự kiện chính | 1. Hệ thống hiển thị giao diện đăng ký  2. Người dùng nhập các thông tin cá nhân.  3. Hệ thống kiểm tra thông tin đăng ký và xác thực qua email.  4. Hệ thống trả về kết quả xác thực email.  5. Hệ thống thông báo đăng ký thành công và lưu thông tin vào sơ sở dữ liệu.  6. Kết thúc Use case. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Người dùng hủy yêu cầu đăng ký.  2. Hệ thống trở về giao diện chính.  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Người dùng nhập thông tin sai.  2. Hệ thống hiển thị thông báo lỗi.  3. Kết thúc Use case. |
| Các yêu cầu đặc biệt | Không có |

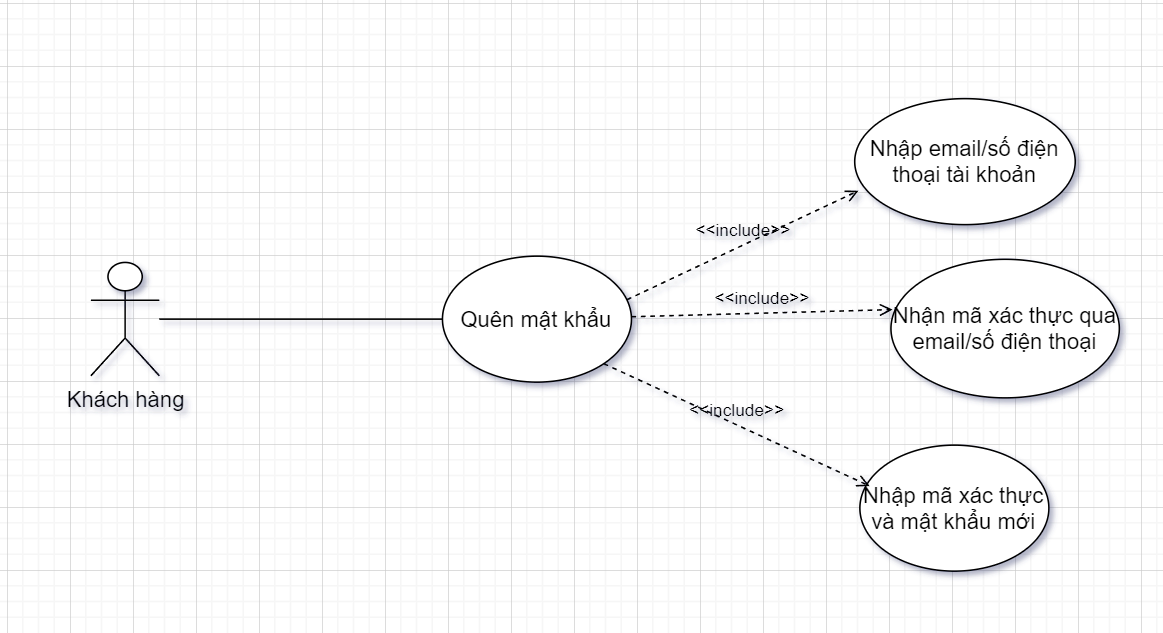
* 1. Usecase chỉnh sửa thông tin cá nhân



*Hình 2.4. Usecase chỉnh sửa thông tin cá nhân*

|  |  |
| --- | --- |
| Tên Usecase | Sửa thông tin cá nhân |
| Actor | Người dùng. |
| Mô tả | Tác nhân sử dụng use case này để thực hiện sửa thông tin cá nhân như: họ tên, giới tính ,... |
| Điều kiện kích hoạt | Chọn chức năng sửa thông tin cá nhân ở hệ thống |
| Tiền điều kiện | Tác nhân đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Thông tin cập nhật thành công vào hệ thống. Hệ thống thông báo được cập nhật thành công.  Nếu thất bại: Hệ thống sẽ đưa ra thông báo “Thông tin cập nhật không hợp lệ”. |
| Luồng sự kiện chính | 1.Người dùng chọn chức năng sửa thông tin cá nhân.  2. Hệ thống hiển thị giao diện sửa thông tin  3. Người dùng nhập thông tin cần sửa vào và nhấn lưu.  4. Hệ thống kiểm tra và xác nhận thông tin vừa nhập.  5. Hiển thị thông báo cập nhật thông tin thành công và lưu thông tin vào cơ sở dữ liệu.  6. Kết thúc Use case. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Người dùng hủy yêu cầu sửa thông tin.  2. Hệ thống bỏ qua và trở về giao diện chính.  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Người dùng nhập thông tin không hợp lệ.  2. Hệ thống hiển thị thông báo lỗi.  3. Kết thúc Use case. |
| Các yêu cầu đặc biệt | Không có |

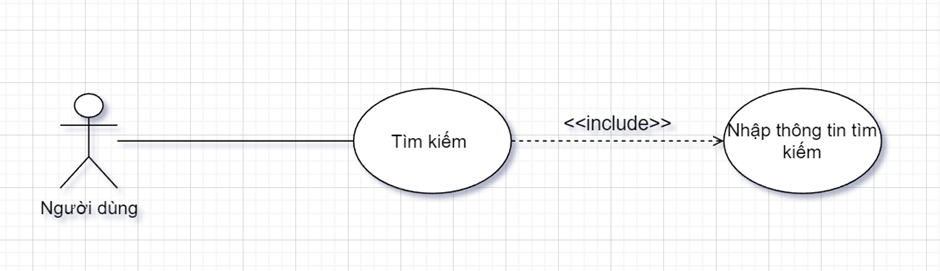
* 1. Usecase Quên mật khẩu



*Hình 2.5. Usecase Quên mật khẩu*

|  |  |
| --- | --- |
| Tên Usecase | Quên mật khẩu |
| Actor | Người dùng. |
| Mô tả | Tác nhân sử dụng use case này để được đổi mật khẩu mới vì quên mật khẩu cũ của tài khoản trong hệ thống. |
| Điều kiện kích hoạt | Chọn chức năng quên mật khẩu ở hệ thống. |
| Tiền điều kiện | Tác nhân chưa vào hệ thống. |
| Hậu điều kiện | Đổi được mật khẩu mới và đăng nhập vào hệ thống thành công |
| Luồng sự kiện chính | 1. Tác nhân chọn quên mật khẩu.  2. Tác nhân nhập email/số điện thoại tài khoản.  3. Hệ thống gửi mã xác nhận qua email/số điện thoại và mã xác thực chỉ có tác dụng trong vòng 2 phút.  4. Tác nhân nhập mã xác nhận và mật khẩu m.  5. Hệ thống thông báo kết quả.  6. Kết thúc Usecase. |
| Luồng sự kiện phụ | 1. Người dùng nhập mã xác nhận không đúng  2. Hệ thống thông báo lỗi sai mã xác nhận.  3. Kết thúc Usecase |
| Các yêu cầu đặc biệt | Không có |

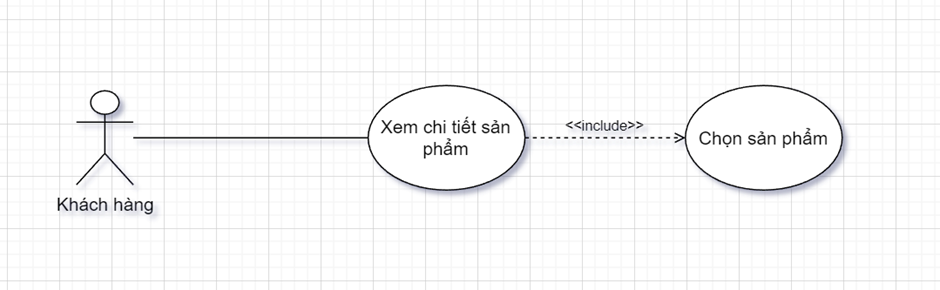
* 1. Usecase tìm kiếm



*Hình 2.6. Usecase tìm kiếm*

|  |  |
| --- | --- |
| Tên Usecase | Tìm kiếm |
| Actor | Người dùng. |
| Mô tả | Tác nhân sử dụng use case này để thực hiện tìm kiếm trong hệ thống. |
| Điều kiện kích hoạt | Chọn chức năng tìm kiếm ở hệ thống. |
| Tiền điều kiện | Tác nhân đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Tìm kiếm thành công các đối tượng tượng ứng. |
| Luồng sự kiện chính | 1. Tác nhân chọn vào thanh tìm kiếm.  2. Tác nhân nhập thông tin cần tìm kiếm và nhấn Enter  3. Hệ thống trả về các kết quả tương ứng.  4. Tác nhân tiếp tục tìm kiếm hoặc thực hiện các chức năng khác.  5. Kết thúc Usecase |
| Luồng sự kiện phụ | 1. Không tìm thấy kết quả tương ứng  1. Hệ thống thông báo không tìm thấy kết quả  3. Kết thúc Usecase |
| Các yêu cầu đặc biệt | Không có |

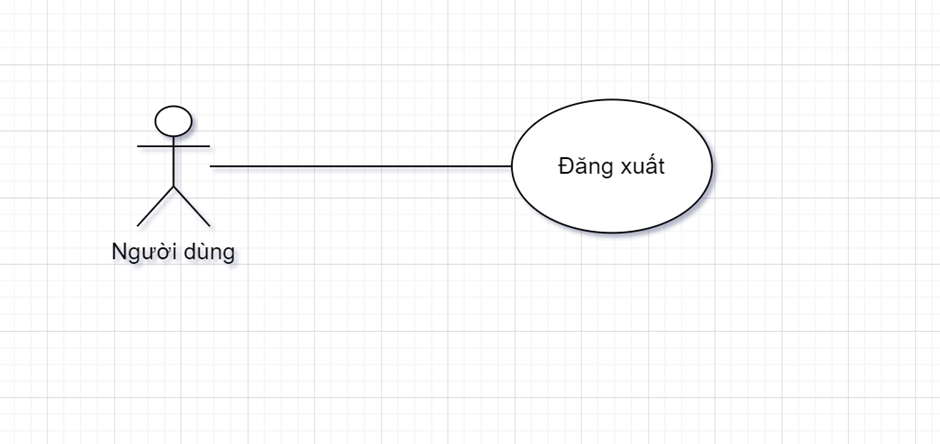
* 1. Usecase xem chi tiết sản phẩm



*Hình 2.7. Usecase xem chi tiết sản phẩm*

|  |  |
| --- | --- |
| Tên Usecase | Xem chi tiết sản phẩm |
| Actor | Khách hàng. |
| Mô tả | Tác nhân sử dụng use case này để xem chi tiết sản phẩm. |
| Điều kiện kích hoạt | Khách hàng truy cập vào hệ thống và chọn một sản phẩm để xem chi tiết |
| Luồng sự kiện chính | 1. Khách hàng truy cập vào hệ thống  2. Khách hàng chọn một sản phẩm để xem chi tiết.  3. Hệ thống trả về giao diện chi tiết sản phẩm  4. Kết thúc usecase. |
| Luồng sự kiện phụ | Không có |
| Các yêu cầu đặc biệt | Không có |

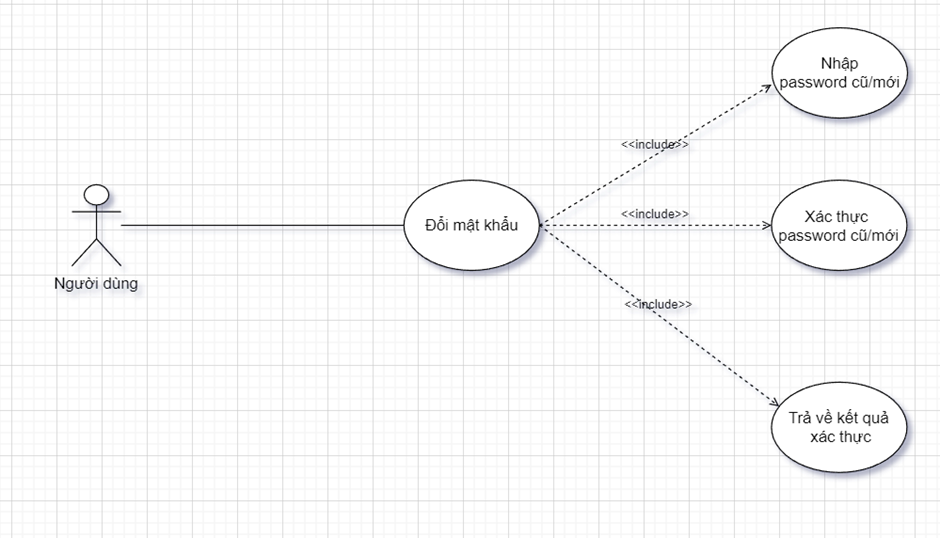
* 1. Usecase Đăng xuất



*Hình 2.8. Usecase Đăng xuất*

|  |  |
| --- | --- |
| Tên Usecase | Đăng xuất |
| Actor | Người dùng. |
| Mô tả | Tác nhân sử dụng use case này để thực hiện đăng xuất khỏi hệ thống. |
| Điều kiện kích hoạt | Chọn chức năng đăng xuất ở hệ thống. |
| Tiền điều kiện | Tác nhân đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Hệ thống sẽ hiển thị giao diện khi chưa đăng nhập.  Nếu thất bại: Hệ thống sẽ đưa ra thông báo lỗi. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị giao diện đăng xuất.  2. Người dùng chọn đăng xuất.  3. Hệ thống tiến hành đăng xuất tài khoản người dùng.  4. Hiển thị giao diện khi chưa đăng nhập.  5. Kết thúc Use case. |
| Luồng sự kiện phụ | 1. Người dùng đăng xuất thất bại.  2. Hệ thống hiển thị thông báo lỗi.  3. Kết thúc Use case. |
| Các yêu cầu đặc biệt | Không có |

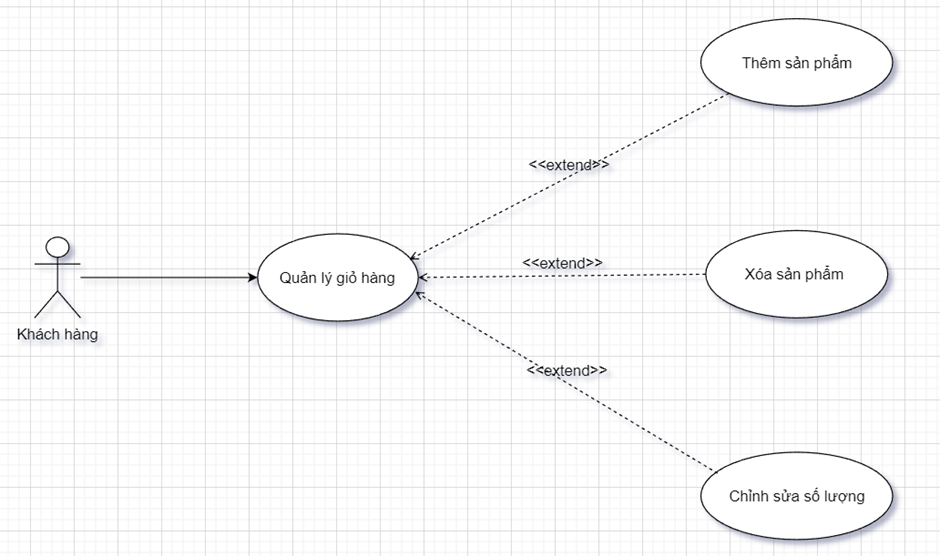
* 1. Usecase đổi mật khẩu



*Hình 2.9. Usecase đổi mật khẩu*

|  |  |
| --- | --- |
| Tên Usecase | Đổi mật khẩu |
| Actor | Người dùng. |
| Mô tả | Tác nhân sử dụng use case này để thực hiện đổi mật khẩu. |
| Điều kiện kích hoạt | Chọn chức năng đổi mật khẩu ở hệ thống |
| Tiền điều kiện | Tác nhân đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Mật khẩu của người dùng thay đổi thành công . Hệ thống hiển thị thông báo đổi mật khẩu thành công và chuyển đến form “Đăng nhập” cho phép người dùng đăng nhập lại.  Nếu thất bại: Hệ thống thông báo lỗi và mật khẩu của người dùng không được thay đổi. |
| Luồng sự kiện chính | 1. Người dùng chọn chức năng thay đổi mật khẩu.  2. Hệ thống hiển thị form thay đổi mật khẩu.  3. Người dùng nhập lại mật khẩu cũ và mật khẩu mới và xác nhận mật khẩu mới vào textbox.  4. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào và lưu thông tin vào cơ sở dữ liệu.  5. Kết thúc Use case. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Người dùng hủy yêu cầu thay đổi mật khẩu.  2. Hệ thống không thực hiện chức năng thay đổi mật khẩu.  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Người dùng nhập thông tin không hợp lệ.  2. Hệ thống thông báo thông tin không hợp lệ.  3. Kết thúc Use case. |
| Các yêu cầu đặc biệt | Không có |

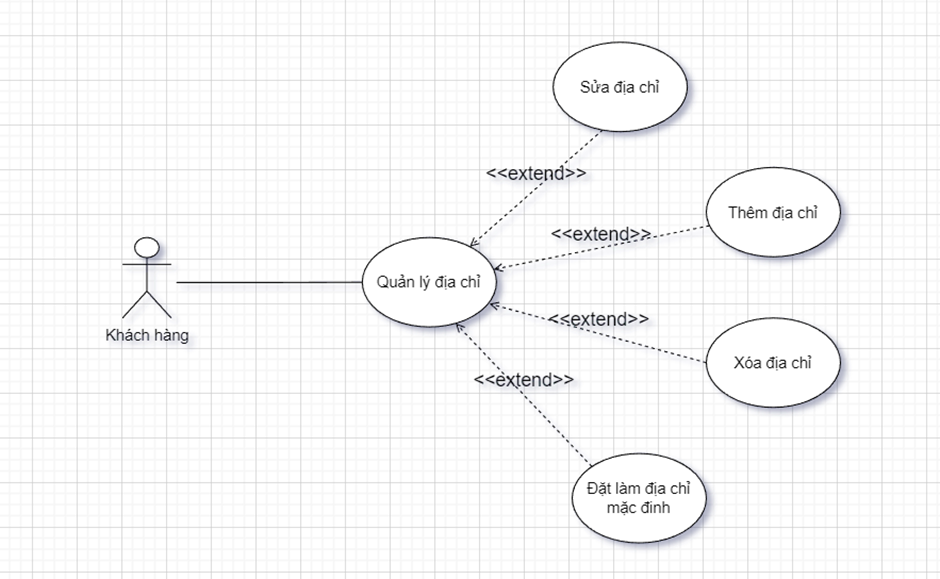
* 1. Usecase quản lý giỏ hàng



*Hình 2.10. Usecase quản lý giỏ hàng*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý giỏ hàng |
| Actor | Khách hàng |
| Mô tả | Khách hàng sử dụng Use case này để thực hiện chức năng quản lý giỏ hàng như: Thêm sản phẩm vào giỏ hàng, cập nhật  số lượng sản phẩm trong giỏ hàng, xóa sản phẩm trong giỏ hàng. |
| Điều kiện kích hoạt | Khách hàng chọn chức năng quản lý giỏ hàng. |
| Tiền điều kiện | Khách hàng phải đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Hệ thống thông báo và thông tin giỏ hàng được cập nhật thành công vào hệ thống.  Nếu thất bại: Hệ thống trả về thông báo lỗi, thông tin không được cập nhật thành công. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng giỏ hàng.  2. Tác nhân chọn một trong các chức năng: thêm, xóa, thay đổi thông tin.  3. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào.  4. Hệ thống lưu thông tin vào cơ sở dữ liệu.  5. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc quản lý giỏ hàng.  2. Hệ thống bỏ qua và trở về giao diện chính.  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Tác nhân thực hiện chức năng không hợp lệ,  2. Hệ thống hiển thị thông báo lỗi.  3. Kết thúc Use case. |
| Yêu cầu đặc biệt | Không có |

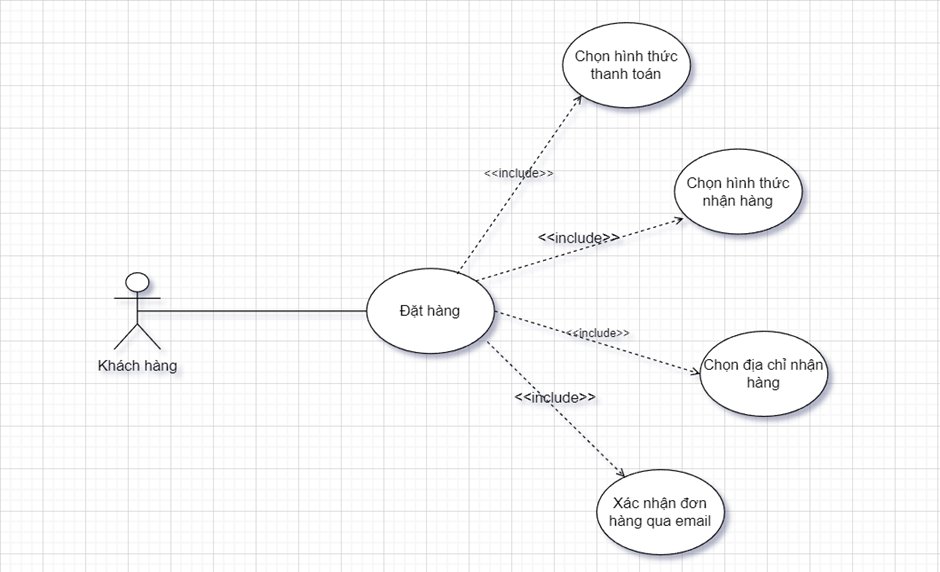
* 1. Usecase quản lý địa chỉ



*Hình 2.11. Usecase quản lý địa chỉ*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý địa chỉ |
| Actor | Khách hàng |
| Mô tả | Khách hàng sử dụng Use case này để thực hiện chức năng quản lý địa chỉ như : Thêm, chỉnh sửa, xóa, đặt làm mặc định. |
| Điều kiện kích hoạt | Khách hàng chọn chức năng quản lý địa chỉ. |
| Tiền điều kiện | Khách hàng  đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị giao diện quản lý địa chỉ và cung cấp các chức năng cần thiết. |
| Luồng sự kiện chính | 1. Khách hàng chọn chức năng quản lý địa chỉ.  2. Khách hàng chọn một trong các chức năng như : Thêm, chỉnh sửa, xóa, đặt làm mặc định  3. Hệ thống kiểm tra và cập nhật vào cơ sở dữ liệu.  4. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Khách hàng hủy bỏ chức năng quản lý địa chỉ.   2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Khách hàng thực hiện chức năng không hợp lệ.  2. Hệ thống hiển thị thông báo lỗi và quay lại bước 1 trong luồng sự kiện chính  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

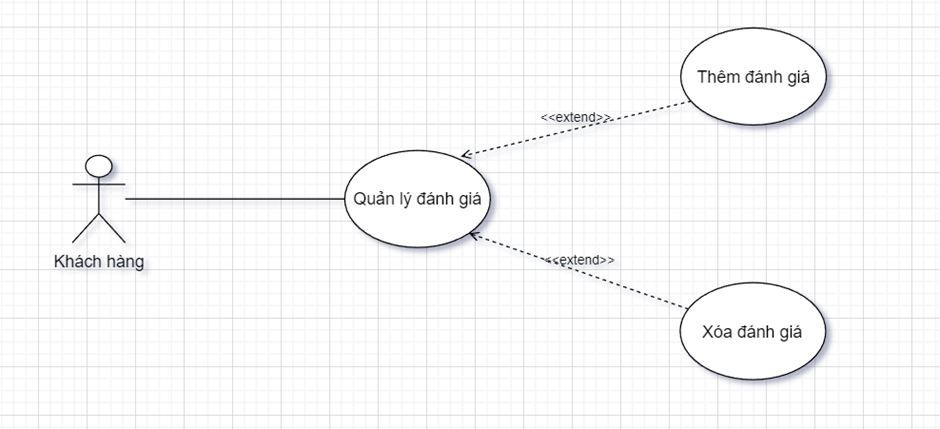
* 1. Usecase Đặt hàng



*Hình 2.12. Usecase Đặt hàng*

|  |  |
| --- | --- |
| Tên Usecase | Đặt hàng |
| Actor | Khách hàng |
| Mô tả | Tác nhân sử dụng Usecase này để thực hiện chức năng đặt hàng. |
| Điều kiện kích hoạt | Tác nhân đang tiến hành mua hàng. |
| Tiền điều kiện | Tác nhân phải đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Khách hàng đặt đơn hàng thành công.  Nếu thất bại: Hệ thống trả về thông báo lỗi, khách hàng đặt đơn hàng không thành công. |
| Luồng sự kiện chính | 1. Tác nhân chọn sản phẩm muốn mua từ giỏ hàng và tiến hành đặt hàng.  2. Tác nhân chọn, nhập các thông tin yêu cầu và tiến hành đặt hàng.  3. Tác nhân xác nhận thông tin đơn hàng qua email.  4. Hệ thống kiểm tra sự hợp lệ và thông báo.  5. Hệ thống lưu thông tin vào cơ sở dữ liệu.  6. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc đặt hàng.  2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Tác nhân xác nhận đơn hàng không thành công  2. Hệ thống hủy bỏ đơn hàng và thông báo cho khách hàng  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

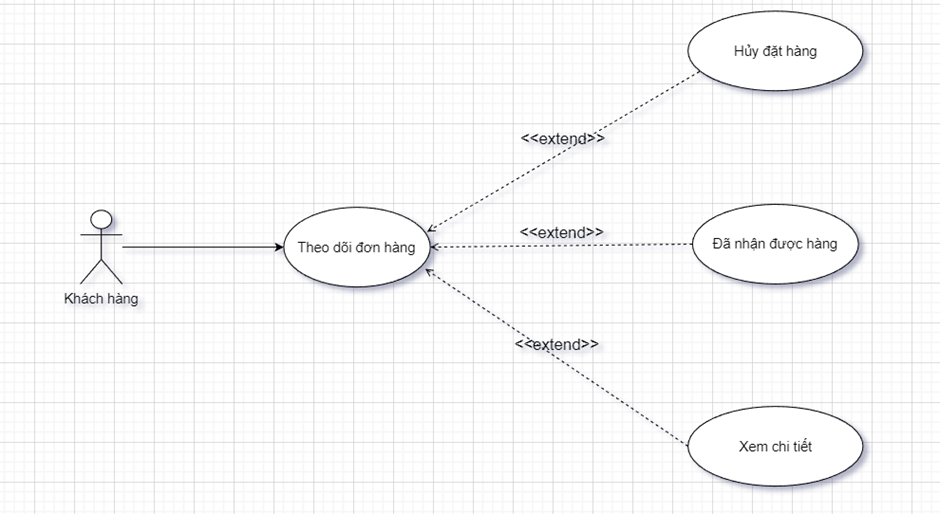
* 1. Usecase quản lý đánh giá



*Hình 2.13. Usecase quản lý đánh giá*

|  |  |
| --- | --- |
| Tên usecase | Quản lý đánh giá |
| Actor | Khách hàng |
| Mô tả | Tác nhân sử dụng usecase này để thực hiện chức năng quản lý đánh giá như: Thêm, xóa đánh giá. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý đánh giá. |
| Tiền điều kiện | Tác nhân đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị giao diện quản lý đánh giá và cung cấp các chức năng cần thiết. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý đánh giá.  2. Tác nhân chọn một trong các chức năng như : Thêm, xóa đánh giá.  3. Hệ thống kiểm tra và cập nhật vào cơ sở dữ liệu.  4. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Khách hàng hủy bỏ chức năng quản lý đánh giá.   2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Thông tin khách hàng nhập vào không hợp lệ.  2. Hệ thống hiển thị thông báo lỗi và quay lại bước 1 trong luồng sự kiện chính  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

* 1. Usecase theo dõi đơn hàng



*Hình 2.14. Usecase theo dõi đơn hàng*

|  |  |
| --- | --- |
| Tên Usecase | Theo dõi đơn hàng |
| Actor | Khách hàng |
| Mô tả | Tác nhân sử dụng Use case này để thực hiện chức năng theo dõi đơn hàng như: Xem chi tiết, hủy đặt hàng, đã nhận hàng. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng theo dõi đơn hàng. |
| Tiền điều kiện | Tác nhân đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị giao diện theo dõi đơn hàng và cung cấp các chức năng cần thiết. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng theo dõi đơn hàng.  2. Tác nhân chọn một trong các chức năng như: xem chi tiết đơn hàng, huy đặt hàng và đã nhận được hàng  3. Hệ thống kiểm tra và cập nhật vào cơ sở dữ liệu.  4. Use case kết thúc. |
| Luồng sự kiện phụ | 1. Khách hàng hủy bỏ chức năng theo dõi đơn hàng.   2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case. |
| Yêu cầu đặc biệt | Không có. |

* 1. Usecase quản lý sản phẩm

A diagram of a diagram

Description automatically generated

*Hình 2.15. Usecase quản lý sản phẩm*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý sản phẩm |
| Actor | Nhân viên bán hàng |
| Mô tả | Nhân viên bán hàng sử dụng Use case này để thực hiện chức năng quản lý / sản phẩm như: Thêm mới sản phẩm, cập nhật thông tin sản phẩm, xóa sản phẩm, tìm kiếm sản phẩm. |
| Điều kiện kích hoạt | Nhân viên bán hàng chọn chức năng quản lý sản phẩm. |
| Tiền điều kiện | Nhân viên bán hàng phải đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Thông tin sản phẩm được cập nhật thành công vào hệ thống.  Nếu thất bại: Hệ thống trả về thông báo lỗi, thông tin không được cập nhật thành công. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý sản phẩm.  2. Tác nhân chọn một trong các chức năng: thêm, xóa, thay đổi thông tin, tìm kiếm.  3. Tác nhân chọn lưu thông tin.  4. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào.  5. Hệ thống lưu thông tin vào cơ sở dữ liệu.  6. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc quản lý sản phẩm.  2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Thông tin tác nhân nhập vào không hợp lệ  2. Hệ thống hiển thị thông báo lỗi  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

* 1. Usecase quản lý đơn hàng

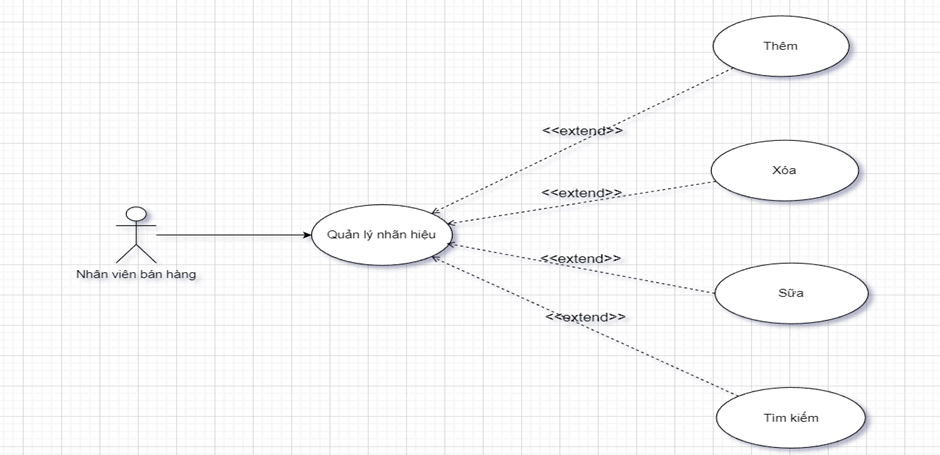
A diagram of a diagram

Description automatically generated

*Hình 2.16. Usecase quản lý đơn hàng*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý đơn hàng |
| Actor | Nhân viên bán hàng |
| Mô tả | Tác nhân sử dụng Usecase này để thực hiện chức năng quản lý đơn hàng như: Xem chi tiết đơn hàng, duyệt đơn hàng, tìm kiếm đơn hàng. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý đơn hàng. |
| Tiền điều kiện | Tác nhân phải đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Thông tin đơn hàng được cập nhật thành công vào hệ thống.  Nếu thất bại: Hệ thống trả về thông báo lỗi, thông tin không được cập nhật thành công. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý đơn hàng.  2. Tác nhân chọn một trong các chức năng: Xem chi tiết đơn hàng, duyệt đơn hàng, tìm kiếm đơn hàng.  3. Tác nhân chọn lưu thông tin.  4. Hệ thống kiểm tra sự hợp lệ của thông tin và thông báo.  5. Hệ thống lưu thông tin vào cơ sở dữ liệu.  6. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc quản lý đơn hàng.  2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Thao tác của tác nhân gây ra lỗi.  2. Hệ thống hiển thị thông báo lỗi  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

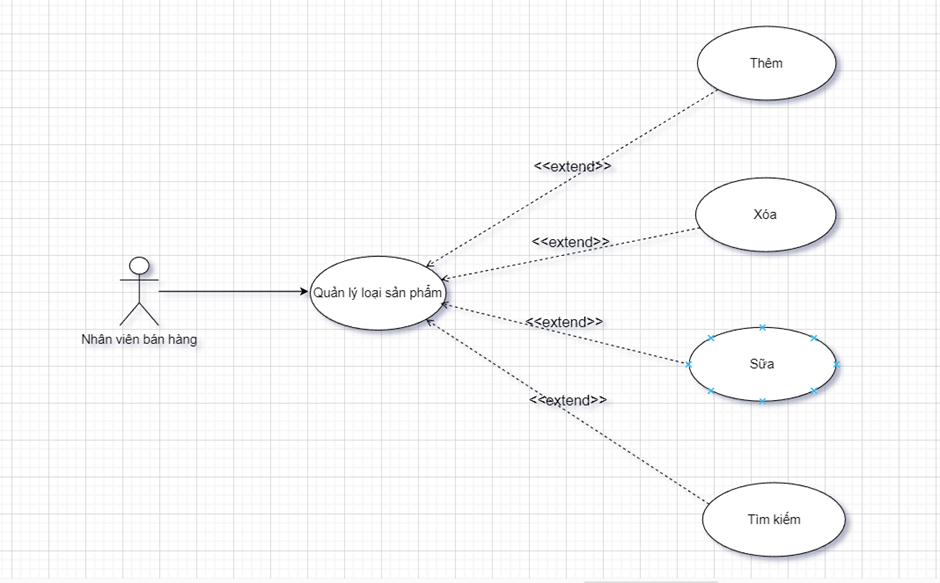
* 1. Usecase quản lý nhãn hiệu



*Hình 2.17. Usecase quản lý nhãn hiệu*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý nhãn hiệu |
| Actor | Nhân viên bán hàng |
| Mô tả | Tác nhân sử dụng Use case này để thực hiện chức năng quản lý nhãn hiệu như: Thêm, xóa, sửa, tìm kiếm nhãn hàng. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý nhãn hiệu. |
| Tiền điều kiện | Tác nhân đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị giao diện quản lý nhãn hiệu và cung cấp các chức năng cần thiết. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý nhãn hiệu.  2. Tác nhân chọn một trong các chức năng như: Thêm, chỉnh sửa, xóa, tìm kiếm  3. Hệ thống kiểm tra và cập nhật vào cơ sở dữ liệu.  4. Use case kết thúc. |
| Luồng sự kiện phụ | 1. Khách hàng hủy bỏ chức năng quản lý nhãn hiệu.   2. Hệ thống bỏ qua và trở về giao diện chính.  3. Kết thúc Use case. |
| Yêu cầu đặc biệt | Không có |

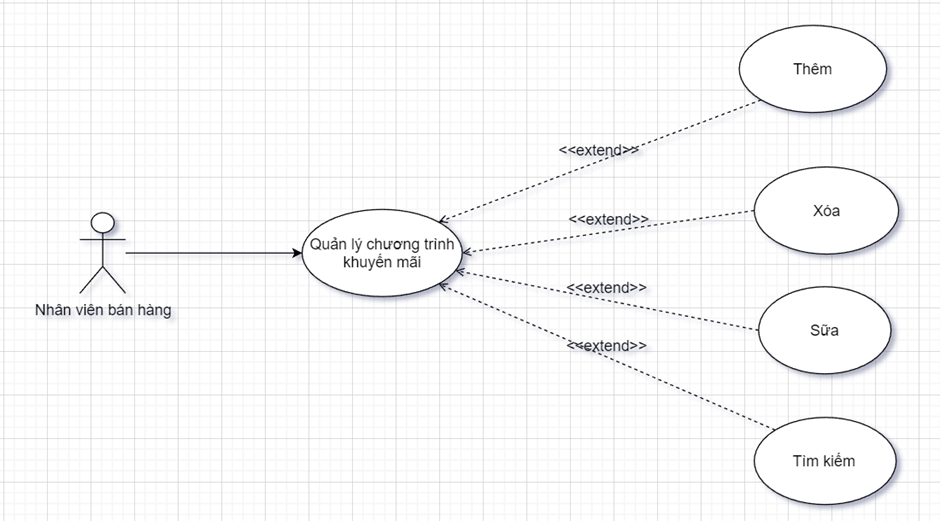
* 1. Usecase quản lý loại sản phẩm



*Hình 2.18. Usecase quản lý loại sản phẩm*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý loại sản phẩm |
| Actor | Nhân viên bán hàng |
| Mô tả | Tác nhân sử dụng Use case này để thực hiện chức năng quản lý loại sản phẩm như: Thêm, xóa, sửa, tìm kiếm loại sản phẩm. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý loại sản phẩm. |
| Tiền điều kiện | Tác nhân đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị giao diện quản lý loại sản phẩm và cung cấp các chức năng cần thiết. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý loại sản phẩm.  2. Tác nhân chọn một trong các chức năng như: Thêm, chỉnh sửa, xóa, tìm kiếm  3. Hệ thống kiểm tra và cập nhật vào cơ sở dữ liệu.  4. Usecase kết thúc. |
| Luồng sự kiện phụ | 1. Khách hàng hủy bỏ chức năng quản lý loại sản phẩm.  2. Hệ thống bỏ qua và trở về giao diện chính.  3. Kết thúc Usecase. |
| Yêu cầu đặc biệt | Không có |

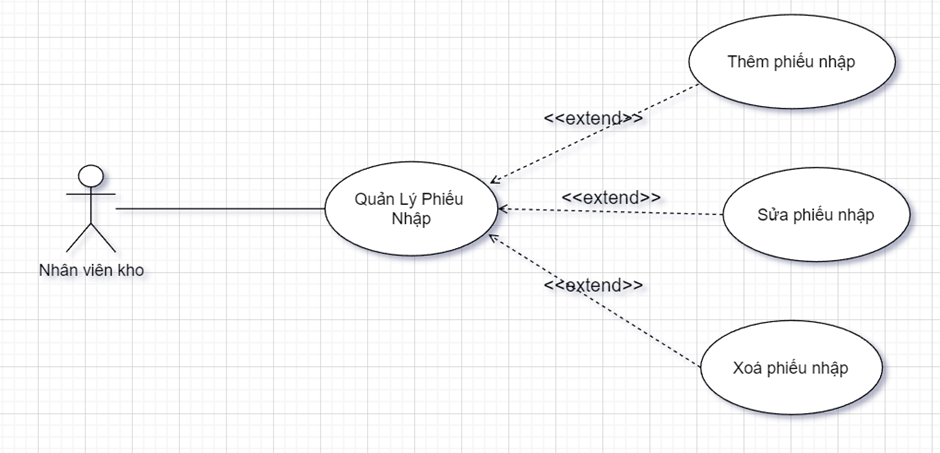
* 1. Usecase quản lý chương trình khuyến mãi



*Hình 2.19. Usecase quản lý chương trình khuyến mãi*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý chương trình khuyến mãi |
| Actor | Nhân viên bán hàng |
| Mô tả | Tác nhân sử dụng Use case này để thực hiện chức năng quản lý chương trình khuyến mãi như: Thêm, xóa, sửa, tìm kiếm. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý chương trình khuyến mãi. |
| Tiền điều kiện | Tác nhân đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị giao diện quản lý chương trình khuyến mãi và cung cấp các chức năng cần thiết. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý chương trình khuyến mãi.  2. Tác nhân chọn một trong các chức năng như: Thêm, chỉnh sửa, xóa, tìm kiếm  3. Hệ thống kiểm tra và cập nhật vào cơ sở dữ liệu.  4. Use case kết thúc. |
| Luồng sự kiện phụ | 1. Khách hàng hủy bỏ chức năng quản lý chương trình khuyến mãi.   2. Hệ thống bỏ qua và trở về giao diện chính.  3. Kết thúc Usecase. |
| Yêu cầu đặc biệt | Không có |

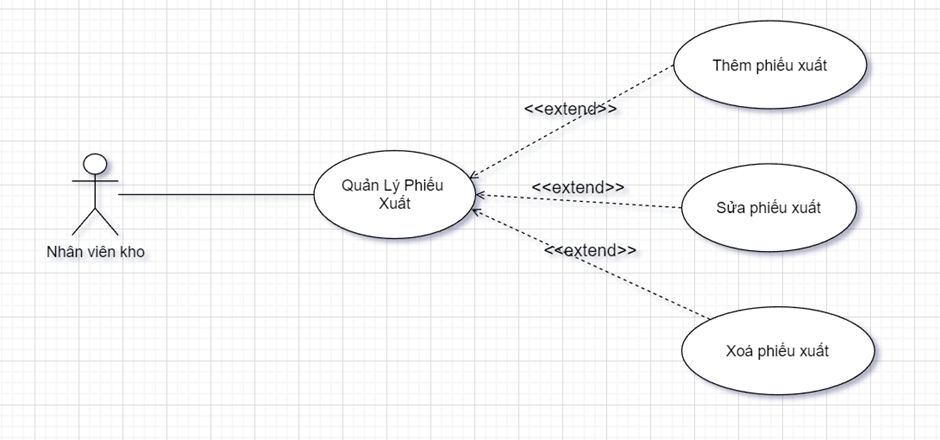
* 1. Usecase quản lý phiếu nhập



*Hình 2.20. Usecase quản lý phiếu nhập*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý phiếu nhập |
| Actor | Nhân viên kho |
| Mô tả | Tác nhân sử dụng Use case này để thực hiện chức năng quản lý phiếu nhập như: Thêm mới phiếu nhập, sửa phiếu nhập, xóa phiếu nhập. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý phiếu nhập. |
| Tiền điều kiện | Tác nhân phải đăng nhập vào hệ thống dưới quyền nhân viên kho. |
| Hậu điều kiện | Nếu thành công: Thông tin phiếu nhập được cập nhật thành công vào hệ thống.  Nếu thất bại: Hệ thống trả về thông báo lỗi, thông tin không được cập nhật thành công. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý phiếu nhập.  2. Tác nhân chọn một trong các chức năng: thêm, sửa, xóa phiếu nhập.  3. Tác nhân chọn lưu thông tin.  4. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào và thông báo.  5. Hệ thống lưu thông tin vào cơ sở dữ liệu.  6. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc quản lý phiếu nhập.  2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Thông tin tác nhân nhập vào không hợp lệ  2. Hệ thống hiển thị thông báo lỗi  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

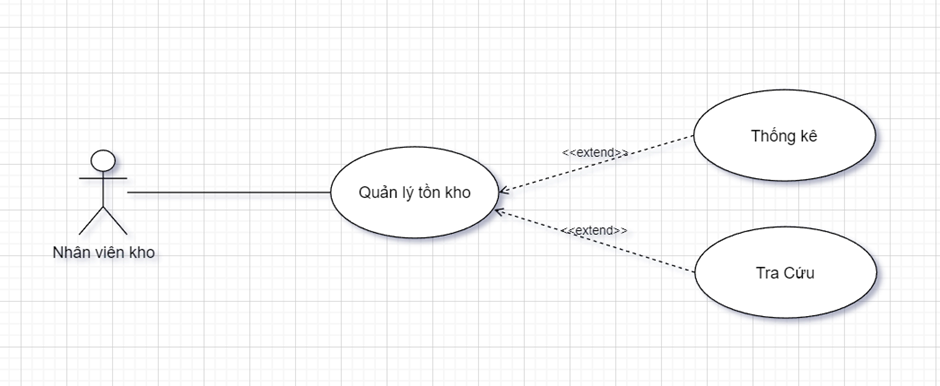
* 1. Usecase quản lý phiếu xuất



*Hình 2.21. Usecase quản lý phiếu xuất*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý phiếu xuất |
| Actor | Nhân viên kho |
| Mô tả | Tác nhân sử dụng Use case này để thực hiện chức năng quản lý phiếu xuất như: Thêm mới phiếu xuất, sửa phiếu xuất, xóa phiếu xuất. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý phiếu xuất. |
| Tiền điều kiện | Tác nhân phải đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Thông tin phiếu xuất được cập nhật thành công vào hệ thống.  Nếu thất bại: Hệ thống trả về thông báo lỗi, thông tin không được cập nhật thành công. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý phiếu xuất.  2. Tác nhân chọn một trong các chức năng: thêm, xóa, sửa phiếu xuất.  3. Tác nhân chọn lưu thông tin.  4. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào và thông báo.  5. Hệ thống lưu thông tin vào cơ sở dữ liệu.  6. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc quản lý phiếu xuất.  2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Thông tin tác nhân nhập vào không hợp lệ  2. Hệ thống hiển thị thông báo lỗi  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

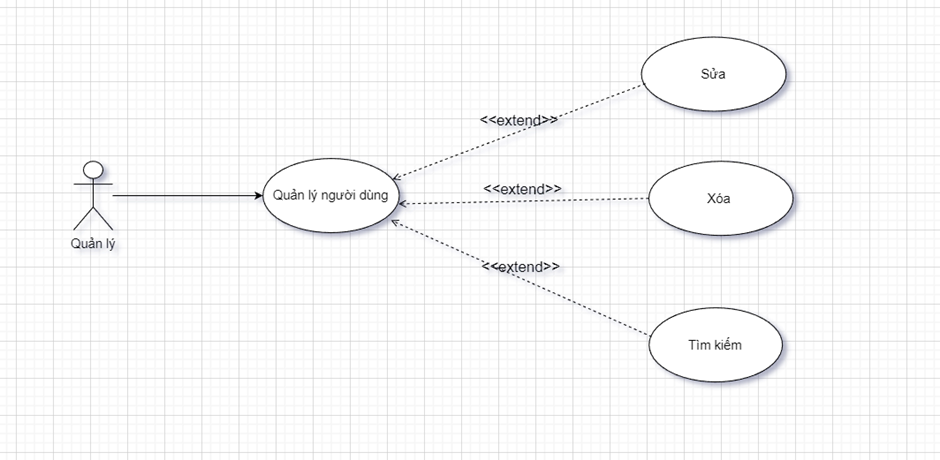
* 1. Usecase quản lý tồn kho



*Hình 2.22. Usecase quản lý tồn kho*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý tồn kho |
| Actor | Nhân viên kho |
| Mô tả | Tác nhân sử dụng Use case này để thực hiện chức năng quản lý tồn kho như: Tra cứu, thống kê. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng quản lý tồn kho. |
| Tiền điều kiện | Tác nhân phải đăng nhập vào hệ thống. |
| Hậu điều kiện | Nhân viên kho chọn một trong các chức năng quản lý kho. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý kho.  2. Tác nhân chọn một trong các chức năng: thống kê, tra cứu.  3. Tác nhân chọn lưu thông tin.  4. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào.  5. Hệ thống lưu thông tin vào cơ sở dữ liệu.  6. Use case kết thúc. |
| Luồng sự kiện phụ | 1. Tác nhân hủy bỏ việc quản lý tồn kho.  2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case. |
| Yêu cầu đặc biệt | Không có |

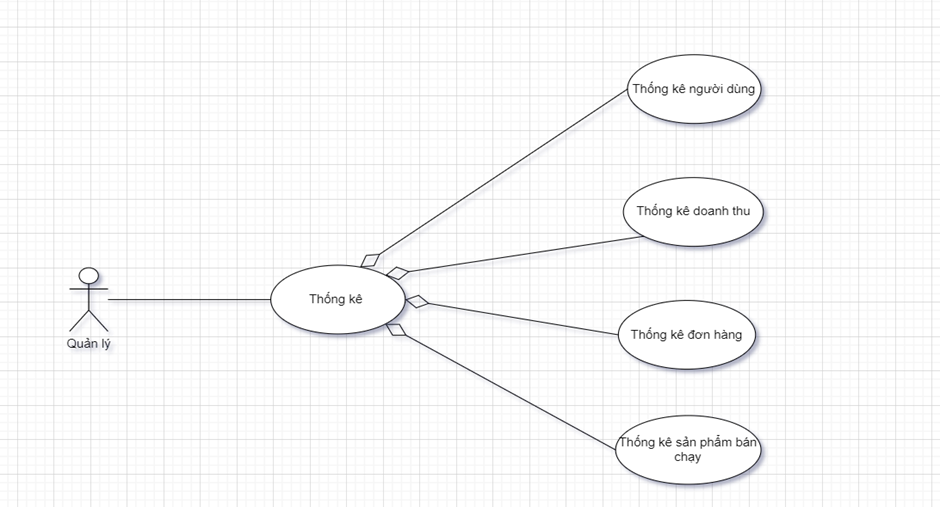
* 1. Usecase quản lý người dùng



*Hình 2.23. Usecase quản lý người dùng*

|  |  |
| --- | --- |
| Tên Usecase | Quản lý người dùng |
| Actor | Quản lý |
| Mô tả | Quản lý sử dụng Use case này để thực hiện chức năng quản lý người dùng như: Thêm, cập nhật, xóa, tìm kiếm. |
| Điều kiện kích hoạt | Quản lý chọn chức năng quản lý người dùng |
| Tiền điều kiện | Quản lý phải đăng nhập vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Hệ thống thông báo thành công và thông tin người dùng được cập nhật thành công vào hệ thống.  Nếu thất bại: Hệ thống trả về thông báo lỗi, thông tin không được cập nhật thành công. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng quản lý người dùng.  2. Tác nhân chọn một trong các chức năng: thêm, xóa, cập nhật, tìm kiếm.  3. Tác nhân chọn lưu thông tin.  4. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào.  5. Hệ thống lưu thông tin vào cơ sở dữ liệu.  6. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc quản lý người dùng  2. Hệ thống bỏ qua và trở về giao diện chính  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Thông tin tác nhân nhập vào không hợp lệ  2. Hệ thống hiển thị thông báo lỗi  3. Kết thúc Use case |
| Yêu cầu đặc biệt | Không có |

* 1. Usecase thống kê



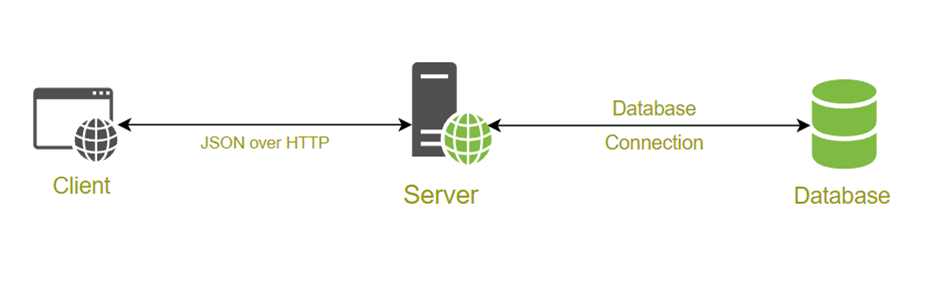
*Hình 2.24. Usecase thống kê*

|  |  |
| --- | --- |
| Tên Usecase | Thống kê |
| Actor | Quản lý |
| Mô tả | Tác nhân sử dụng use case này để thực hiện chức năng thống kê như: Thống kê người dùng, thống kê doanh thu, thống kê hóa đơn, thống kê sản phẩm bán chạy. |
| Điều kiện kích hoạt | Tác nhân chọn chức năng thống kê trong hệ thống. |
| Tiền điều kiện | Tác nhân phải đăng nhập tài khoản quản lý vào hệ thống. |
| Hậu điều kiện | Nếu thành công: Quản lý xem được các loại thống kê.  Nếu thất bại: Hệ thống thông báo lỗi, trở về giao diện chính. |
| Luồng sự kiện chính | 1. Tác nhân chọn chức năng thống kê doanh thu.  2. Tác nhân chọn một trong các chức năng thống kê: Thống kê người dùng, thống kê doanh thu, thống kê hóa đơn, thống kê sản phẩm bán chạy.  4. Hệ thống hiển thị danh sách thống kê.  5. Use case kết thúc. |
| Luồng sự kiện phụ | Dòng sự kiện phụ 1  1. Tác nhân hủy bỏ việc xem thống kê.  2. Hệ thống bỏ qua và trở về giao diện chính.  3. Kết thúc Use case.  Dòng sự kiện phụ 2  1. Tác nhân chọn vào chức năng bị lỗi.  2. Hệ thống hiển thị thông báo lỗi.  3. Kết thúc Use case. |
| Yêu cầu đặc biệt | Không có |

1. **Thiết kế kiến trúc.**
   1. Kiến trúc lựa chọn

Hệ thống thiết kế theo mô hình Client - Server theo kiến trúc 3 tầng.

* Tầng client:
  + Giao diện người dùng (UI): Đây là phần tương tác trực tiếp với người dùng, hiển thị thông tin và thu thập dữ liệu từ người dùng.
  + Xử lý logic giao diện: Nắm giữ logic giao diện người dùng, xử lý sự kiện và truyền dữ liệu giữa UI và tầng Server.
* Tầng Sever:
  + Logic nghiệp vụ (Business Logic): Xử lý logic nghiệp vụ, thực hiện các quy trình và xử lý yêu cầu từ tầng Client.
  + Quản lý dữ liệu (Data Management): Tương tác với tầng Cơ sở dữ liệu để lưu trữ, xử lý và truy xuất dữ liệu.
* Tầng Database:
  + Hệ quản trị cơ sở dữ liệu (DBMS): Quản lý cơ sở dữ liệu và cung cấp các khả năng truy vấn và lưu trữ dữ liệu.
  1. Lý do chọn kiến trúc.
* Tạo điều kiện dễ dàng khi phát triển, bảo trì: lập trình viên dễ phát triển và bảo trì ứng dụng
* Sử dụng máy tính hiệu quả: sử dụng máy tính phù hợp từng tầng
* Cải tiến hiệu năng: có thể nhân rộng máy tính ở tầng Application Server, Database (dùng cho load balancing), mỗi tầng được chuyên môn hóa, dễ tối ưu hóa
* Nâng cao tính bảo mật: có cơ chế bảo mật cho tầng ứng dụng, tầng dữ liệu phía sau an toàn hơn, dữ liệu chạy tốc độ cao, dễ thao tác.
* Hạn chế đầu tư: dễ dàng thêm bớt máy tính trong hệ thống
* Đa dạng kiểu máy client: mobile, PC, PDAs, …
  1. Vẽ biểu đồ kiểu kiến trúc, mô hình kiến trúc được chọn:



*Hình 3.1. Biểu đồ kiến trúc*

A screenshot of a computer

Description automatically generated

*Hình 3.2. Mô hình kiến trúc*

1. **Công nghệ sử dụng**

## 4.1. Về ngôn ngữ lập trình:

Ngôn ngữ sử dụng: Java là một ngôn ngữ bậc cao, được ra mắt lần đầu vào những năm 1990 bởi tập đoàn Sun Microsystem. Hiện tại thì Java thuộc sở hữu của tập đoàn Oracle**.**



*Hình 4.1. Ngôn ngữ java*

Ưu điểm của ngôn ngữ Java:

* Là ngôn ngữ lập trình cấp cao dễ học và dễ hiểu.
* Là ngôn ngữ hướng đối tượng làm tăng khả năng phát triển mã dễ dàng và tăng hiệu quả.
* Là ngôn ngữ an toàn vì Java không sử dụng con trỏ.
* Quản lý bộ nhớ hiệu quả.
* Hỗ trợ đa luồng, người dùng có thể thực hiện cùng lúc nhiều chương trình.
* Có nhiều tính năng: tự động thu gom rác, không sử dụng con trỏ, xử lý ngoại lệ,…

## 4.2. Hệ quản trị cơ sở dữ liệu:

Hệ quản trị cơ sở dữ liệu được chọn là MySQL

****

*Hình 4.2. Hệ quản trị cơ sở dữ liệu MySQL*

Sử dụng hệ quản trị csdl MySQL bởi các lý do sau:

* Mã nguồn mở tự do và hỗ trợ 24/7.
* Sử dụng dễ dàng: MySQL là cơ sở dữ liệu tốc độ cao và ổn định, công cụ này dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp hệ thống lớn các hàm tiện ích.
* Tính bảo mật cao: MySQL phù hợp với các ứng dụng có truy cập cơ sở dữ liệu trên internet vì nó sở hữu nhiều tính năng bảo mật, thậm chí là bảo mật cấp cao.
* Đa tính năng: MySQL có thể hỗ trợ hàng loạt các chức năng SQL từ hệ quản trị cơ sở dữ liệu quan hệ trực tiếp và cả gián tiếp.
* Khả năng mở rộng và mạnh mẽ: Công cụ MySQL có khả năng xử lý khối dữ liệu lớn và có thể mở rộng khi cần thiết.
* Tương thích trên nhiều hệ điều hành: MySQL tương thích để chạy trên nhiều hệ điều hành, như Novell NetWare, Windows \* Linux \*, nhiều loại UNIX \* (như Sun \* Solaris \*, AIX và DEC \* UNIX), OS / 2, FreeBSD \*,.... MySQL cũng cung cấp phương tiện mà các máy khách có thể chạy trên cùng một máy tính với máy chủ hoặc trên một máy tính khác (giao tiếp qua mạng cục bộ hoặc Internet).
* Cho phép khôi phục: MySQL cho phép các transaction được khôi phục, cam kết và phục hồi sự cố.

## 4.3. Công cụ lập trình:

Công cụ lập trình được sử dụng là **IntelliJ IDEA** là một trình IDE dùng để lập trình Java.



*Hình 4.3. Công cụ lập trình IntelliJ IDEA*

Các lợi ích khi dùng IntelliJ IDEA:

* Cho phép các nhà phát triển tập trung phát triển và quản lý tất cả các tác vụ thông thường.
* Cho phép viết, gỡ lỗi, tái cấu trúc, kiểm tra và tìm hiểu mã của bạn mà không bị ảnh hưởng.
* Xử lý liền mạch cơ sở mã hỗn hợp của Java, Ruby, Groovy, Python và Scala.
* Tự động duy trì chất lượng mã.
* Theo dõi và sửa lỗi trên tất cả các cấp độ – từ các câu đến kiến trúc tổng thể.
* Tạo mã “sạch”, nhanh chóng thực hiện mã trong thời gian ngắn nhất.
* Được thiết kế để làm việc trên tất cả các quy mô – từ cá nhân đến doanh nghiệp.
* Hỗ trợ tất cả các ngôn ngữ, công nghệ và framework chính.
* Làm việc với các hệ thống điều khiển phiên bản phổ biến và TeamCity, server tích hợp liên tục

  Công cụ lập trình thứ 2 là **Visual Studio Code** để lập trình giao diện:



*Hình 4.4. Công cụ lập trình Visual Studio Code*

Visual Studio Code là một công cụ soạn thảo mã nguồn do Microsoft phát triển, được giới thiệu lần đầu năm 2015 và chính thức phát hành năm 2016. VSCode có thể cài đặt và sử dụng trên cả Windows, MacOS và Linux. VSCode là mã nguồn mở và nó hoàn toàn miễn phí.

Lợi ích khi sử dụng:

* Dung lượng thấp.
* Phát triển mạnh mẽ trên 3 hệ điều hành phổ biến của thế giới.
* Thiết kế giao diện thân thiện và gần gũi với người dùng.
* Quá trình mã hóa, gỡ lỗi được tối ưu nhất. Visual Studio Code giúp bạn tiết kiệm thời gian bởi nó được thiết kế các phím tắt mở ra chức năng hay thêm dòng lệnh… để tránh gây cản trở khi đang code.
* Kiến trúc ưu việt và có thể khai thác mở rộng. Visual Studio Code được xây dựng phát triển dựa trên Electron kết hợp với các hệ thống ngôn ngữ công nghệ thông minh như: JavaScript, Node.js cùng khả năng vận hành vô cùng mạnh mẽ của chính nó.
* Cộng đồng hỗ trợ vô cùng lớn.

## 4.4. Công nghệ lập trình phía server:

Công nghệ sử dụng: Spring Boot là một dự án khá 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, tạo server cấu hình spring mvc, thì giờ đây với Spring Boot, chúng ta có thể tạo dự án Spring một cách nhanh chóng và cấu hình cũng đơn giản hơn. Dưới đây là một số tính năng nổi bật của Spring Boot:

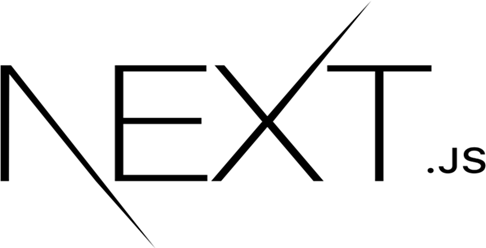
* Tạo các ứng dụng Spring độc lập.
* Các giúp việc cấu hình Maven đơn giản hơn.
* Tự động cấu hình Spring khi cần thiết.
* Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML.



*Hình 4.5. Công nghệ lập trình Spring Boot*

## 4.5. Công nghệ lập trình phía client:

Công nghệ sử dụng: **HTML5**, **CSS3**, **Javascript**, **Boostrap**, **NextJS**

****

*Hình 4.6. Công nghệ Next.js*

Next.js là một framework dùng để phát triển các ứng dụng React theo tư tưởng Isomorphic (Universal), được phát triển bởi [Zeit](https://zeit.co/). Nhờ Next.js chúng ta có thể dễ dàng tạo được một ứng dụng React theo tư tưởng Isomorphic, có bao gồm chức năng Server-Side Rendering (SSR).

Về cơ bản thì với việc SSR thì NextJS giới thiệu là sẽ cung cấp cho chúng ta một số thứ như:

* Hiệu năng tốt hơn so với ứng dụng CSR
* Mang lại khả năng SEO tốt hơn mà CSR không có như là việc chia sẻ bài viết.

Lợi ích khi dùng **NextJS:**

* Ứng dụng NextJS tải nhanh hơn đáng kể so với ứng dụng React do được render phía Server.
* Hỗ trợ các tính năng cho static web.
* Đối với những ai đã có kinh nghiệm làm việc với React thì việc tiếp tập NextJS sẽ là một việc dễ dàng.
* Tự động code splitting cho các page nhằm tối ưu hóa performance khi load trang.
* Dễ dàng xây dựng các API internal thông qua các API routes tích hợp sẵn và tạo các endpoint API.
* Hỗ trợ tích hợp cho route cho page, CSS, JSX và TypeScript.
* Nhanh chóng thêm các plugin để tùy chỉnh NextJS theo nhu cầu của trang cụ thể.

## 4.6. Công nghệ quản lý source code:

Công nghệ sử dụng: **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.

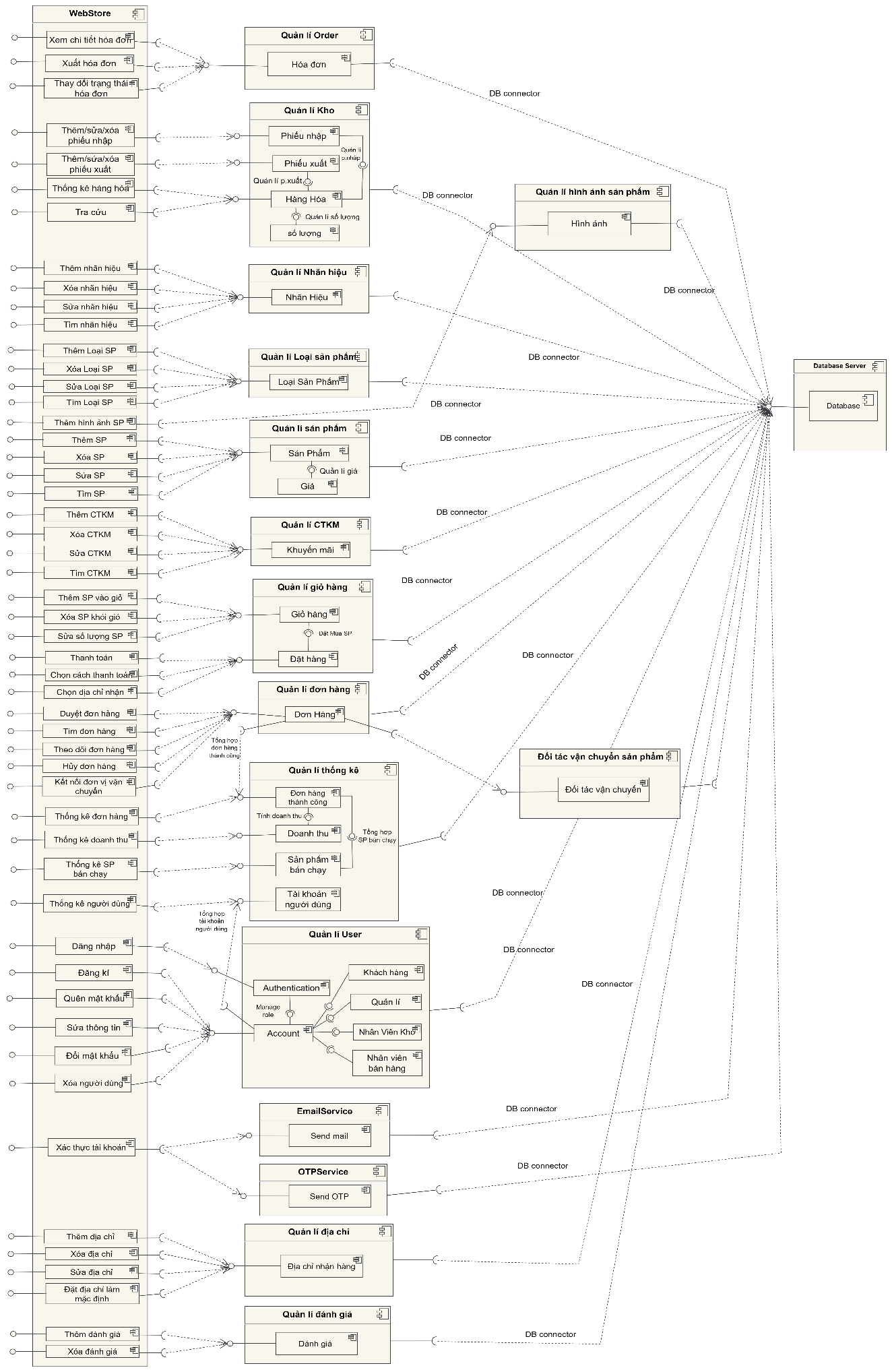


*Hình 4.7. Công nghệ quản lý source code Github*

Lý do sử dụng **github**:

* Có thể sẽ giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch)
* Git dễ sử dụng, có độ bảo mật an toàn thông tin cao và nhanh chóng.
* Có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ trên git.
* Dễ dàng trong việc deployment sản phẩm.
* Dễ dàng theo dõi các thay đổi trong code của bạn qua các phiên bản.
* GitHub tạo 1 môi trường khuyến khích giúp cải thiện coding skill bằng cách theo dõi và so sánh những thay đổi thường xuyên.
* Quản lý source code dễ dàng: Tại đây, bạn có thể coi lại quá trình mình đã làm việc thông qua các comment sau mỗi lần commit. Và cái hay ở đây, là nhiều người có thể cùng làm một repository.
* Github là một kho tài nguyên tuyệt vời: Bạn có thể theo dõi, tìm kiếm những open-source projects theo đúng technology pattern mà bạn ưa thích. Github hỗ trợ code search không kể nó ở dưới dạng một project riêng biệt hay là website.
* Bạn sẽ không còn lo ngại thay đổi làm mất mã nguồn từ kho chứa bởi hệ thống có thể khôi phục hoàn toàn các phiên bản cũ khi cần.

1. **Mô hình thành phần**



*Hình 5.1. Mô hình UML*

Giải thích:

Chức năng của quản lý:

* Đăng nhập vào hệ thống
* Quản lý người dùng: Sửa, xóa, tìm kiếm người dùng
* Quản lý thống kê: thống kê người dùng, thống kê doanh thu, thống kê đơn hàng, thống kê sản phẩm bán chạy.

Chức năng của nhân viên bán hàng:

* Đăng ký, đăng nhập vào hệ thống, sửa thông tin cá nhân, đổi mật khẩu
* Quản lý sản phẩm: thêm, xóa, sửa thông tin, tìm kiếm sản phẩm
* Quản lý đơn hàng: xem, duyệt đơn hàng, tìm kiếm đơn hàng
* Quản lý nhãn hiệu: thêm, xóa, sửa thông tin, tìm kiếm nhãn hiệu
* Quản lý loại sản phẩm: thêm, xóa, sửa thông tin, tìm kiếm loại sản phẩm
* Quản lý CTKM: thêm, xóa, sửa thông tin, tìm kiếm CTKM
* Quản lý order: xem chi tiết hóa đơn, thay đổi trạng thái hóa đơn, xuất hóa đơn.

Chức năng của nhân viên kho:

* Đăng ký, đăng nhập vào hệ thống, sửa thông tin cá nhân, đổi mật khẩu
* Quản lý phiếu nhập: thêm, xóa, sửa phiếu nhập
* Quản lý phiếu xuất: thêm, xóa, sửa phiếu xuất
* Quản lý tồn kho: thống kê số lượng, tra cứu hàng hóa

Chức năng của khách hàng:

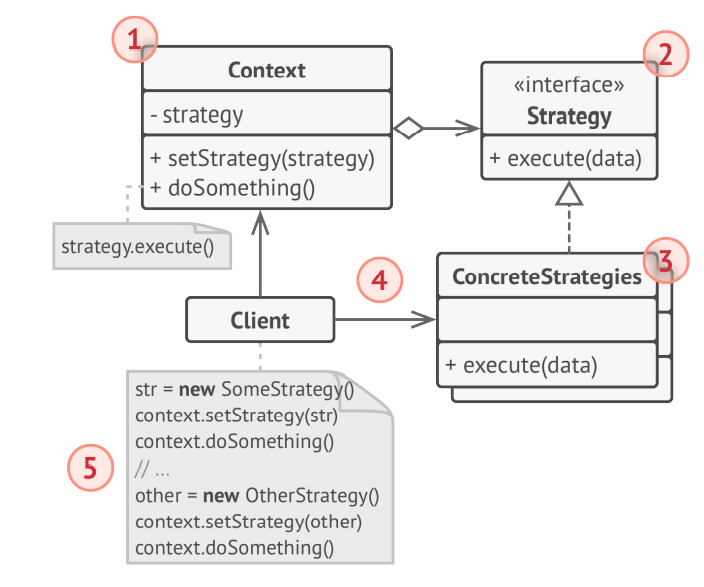
* Đăng ký, đăng nhập vào hệ thống, sửa thông tin cá nhân, đổi mật khẩu
* Quản lý giỏ hàng: thêm sản phẩm vào giỏ hàng, cập nhật số lượng sản phẩm trong giỏ hàng, xóa sản phẩm trong giỏ hàng.
* Quản lý địa chỉ: thêm, xóa, sửa, đặt địa chỉ làm mặc định.
* Quản lý đánh giá: thêm, xóa đánh giá.
* Thanh toán đơn hàng, theo dõi đơn hàng

Các hệ thống con kết nối với Database. Ngoài ra có kết nối với các đối tác thứ 3.

1. **Các design pattern sử dụng và demo chương trình**

## 6.1. Các design pattern được sử dụng:

* + 1. Strategy design pattern
  1. Giới thiệu
* Là design pattern thuộc thuộc nhóm Behavior Pattern
* Mục đích: Định nghĩa một tập hợp các thuật toán giống nhau, encapsulate chúng và khiến chúng có thể thay thế cho nhau. Strategy làm cho phần thuật toán độc lập khỏi client sử dụng nó.
  1. Cấu trúc

****

*Hình 6.1. Cấu trúc Strategy design pattern*

Các thành phần trong mô hình:

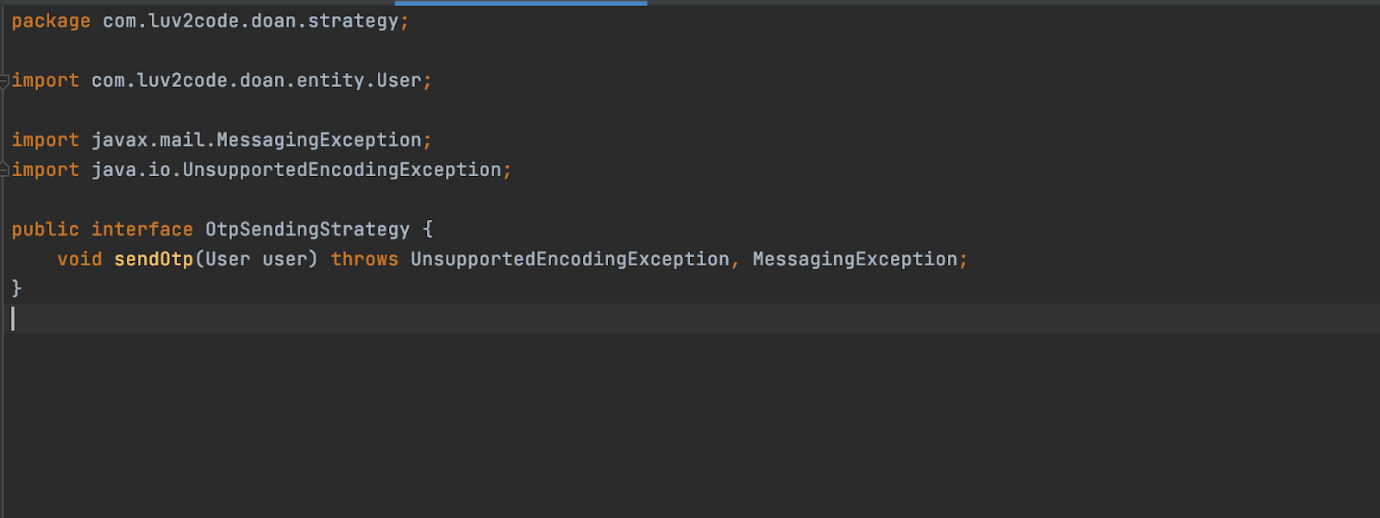
* *Context:* Class sử dụng các strategy object và chỉ giao tiếp với các strategy object thông qua interface
* *Strategy:* Cung cấp một interface chung cho context giao tiếp với các strategy object
* *Concrete Strategy:* Implement các thuật toán khác nhau cho context sử dụng
* *Client:* Có trách nhiệm tạo ra các strategy object và truyền vào cho context sử dụng
  1. Áp dụng

Strategy design pattern được sử dụng để gửi mã xác thực qua email hoặc qua số điện thoại.

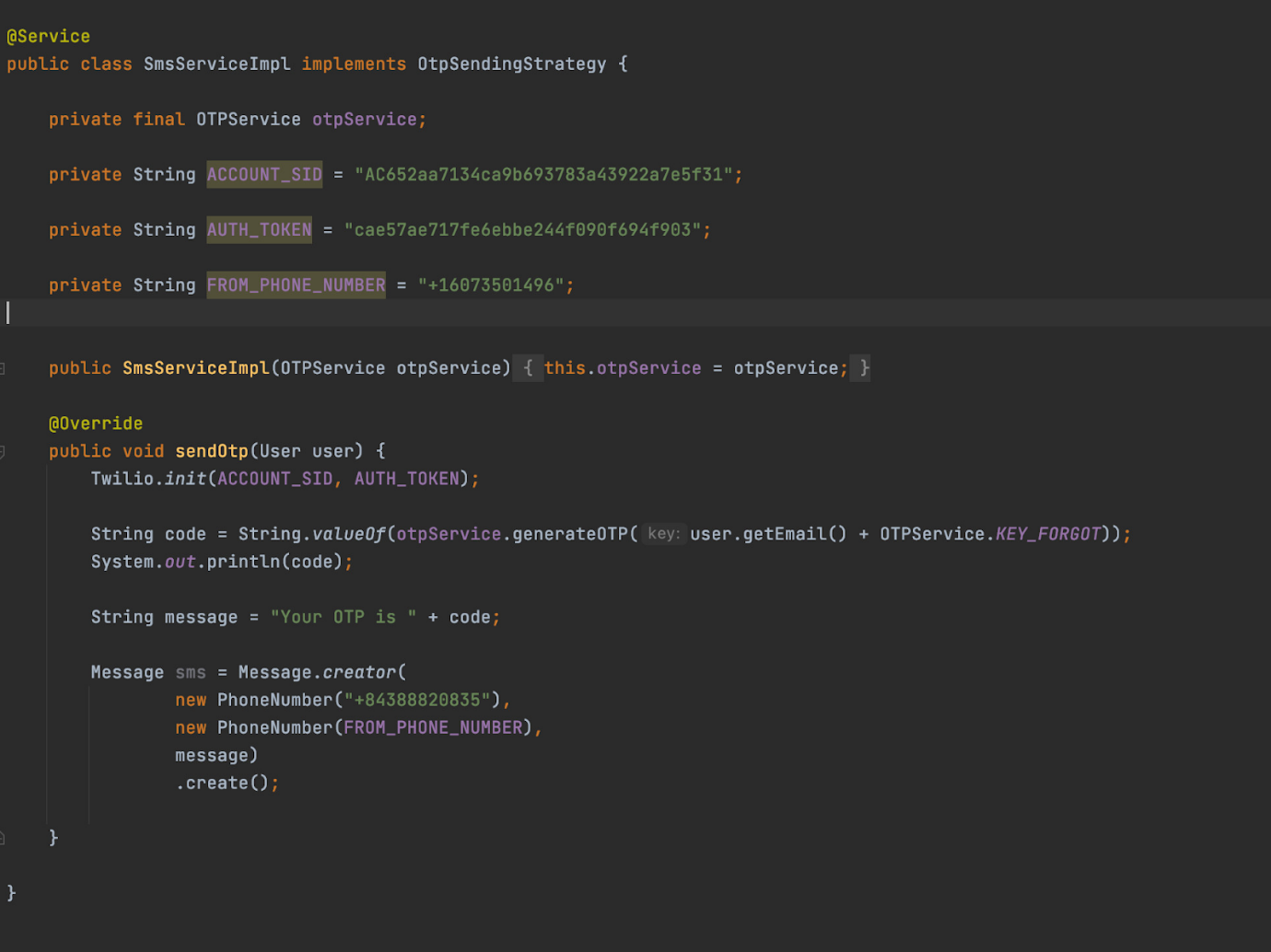
Tạo Context:



Tạo Strategy:

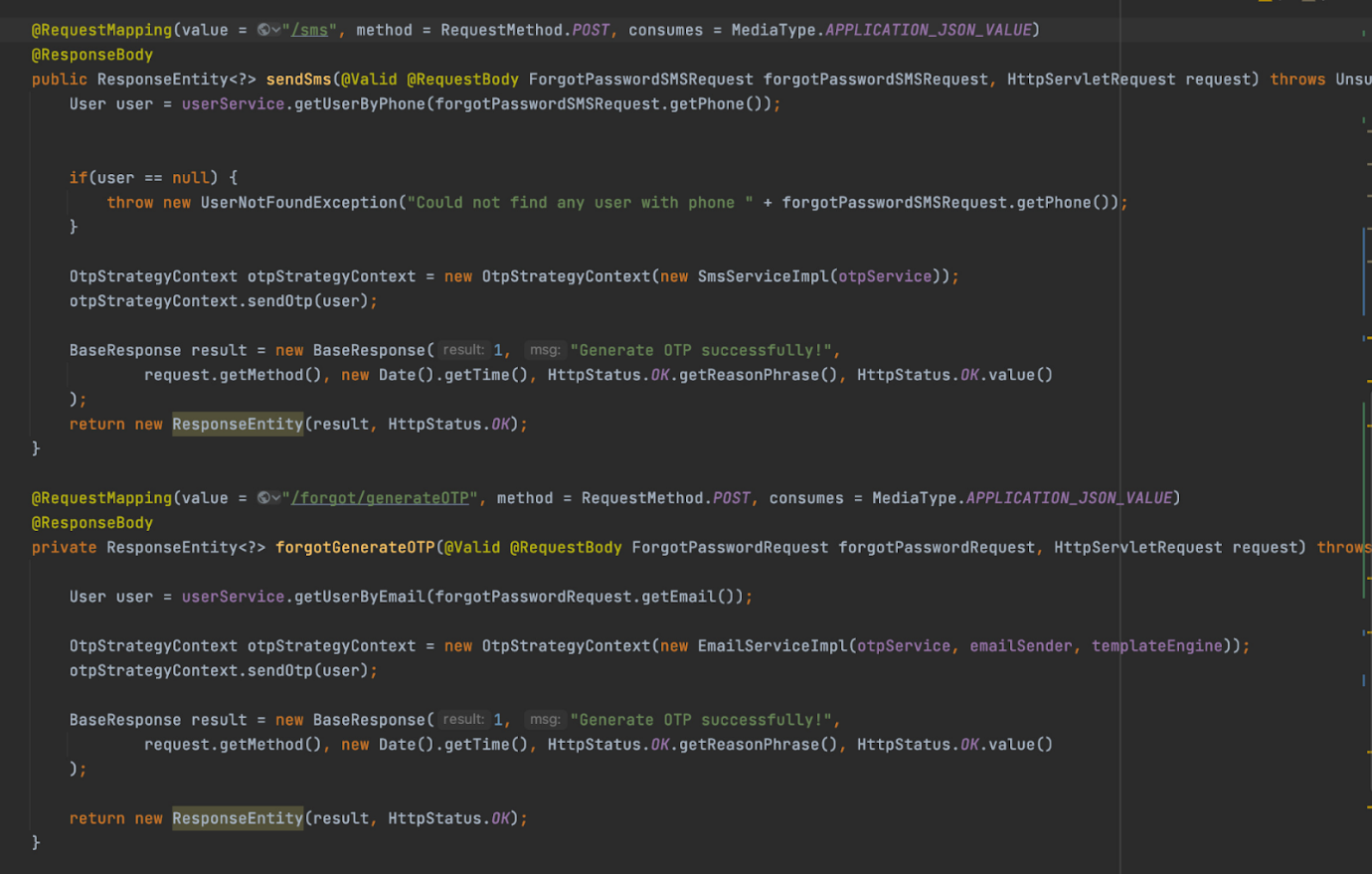


Tạo các concrete strategy:





Tạo Client:

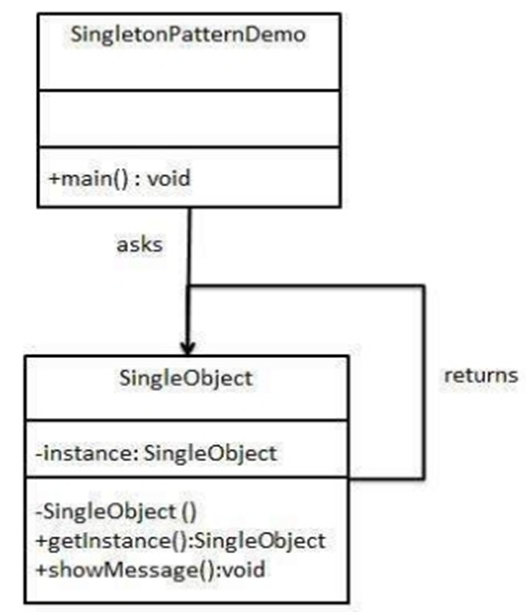


* + 1. Singleton design pattern

1. Giới thiệu

* Là design pattern thuộc thuộc nhóm Creational Design Pattern
* Mục đích: được sử dụng để đảm bảo rằng một class chỉ có một đối tượng duy nhất trong suốt quá trình thực thi của ứng dụng.

1. Cấu trúc

****

*Hình 6.2. Cấu trúc Singleton design pattern*

* Class Singleton: lớp này chứa phương thức để truy cập đến đối tượng duy nhất của lớp. Nó có thể được khởi tạo một lần và được lưu trữ trong bộ nhớ cho đến khi ứng dụng kết thúc.
* Static method: phương thức tĩnh của lớp Singleton, thường được gọi là “getInstance”, là nơi mà đối tượng duy nhất được tạo ra. Nếu đối tượng đã được tạo, phương thức sẽ trả về đối tượng đó; nếu không, nó sẽ tạo một đối tượng mới và trả về nó.
* Hàm tạo (constructor): hàm tạo của lớp Singleton thường là private hoặc protected để đảm bảo rằng không ai có thể tạo ra một đối tượng mới của lớp bên ngoài.
* Biến tĩnh (static variable): biến tĩnh được sử dụng để lưu trữ đối tượng Singleton trong bộ nhớ

1. Áp dụng

Singleton Design Pattern được sử dụng viết logging, tạo một đối tượng LoggingService duy nhất.





## 6.2. Demo chương trình