# CHƯƠNG I: TỔNG QUAN

## 1.1 Giới thiệu đề tài

Văn phòng phẩm là các sản phẩm, vật phẩm và vật dụng dành cho văn phòng làm việc tại các công ty, trường học, bệnh viện, cơ quan, ban ngành đều cần dùng đến văn phòng phẩm. VD như bìa hồ sơ, giấy photo, các dụng cụ văn phòng như dao rọc giấy, kéo, băng keo, giấy note, bút viết các loại, kim bấm, bấm lỗ.

Trên thực tế hiện nay văn phòng phẩm có nghĩa rộng hơn nó bao gồm dụng cụ học và một số dụng cụ liên quan đến ngành mộc.

Theo phong cách truyển thống, chúng ta thường thấy các nhà sách thường tập trung trưng bày sản phẩm trên kệ trong đó hơn 70% là sách, 30% còn lại dành cho các sản phẩm văn phòng phẩm và đồ dùng học tập...

Xu hướng công nghệ phát triển khá mạnh trong những năm gần đây, hỗ trợ rất lớn cho tất các ngành hàng, không chỉ riêng về ngành hàng văn phòng phẩm, đã tạo ra xu thế mới thay đổi không ít trong cách nhìn nhận của người tiêu dùng. Sự xuất hiện của những chiếc điện thoại thông minh, máy tính bảng, laptop, máy tính bảng đọc sách...đã tạo nên một làn sóng mới, dẫn đến việc người dùng nhất là đọc giả sẽ tìm đọc những cuốn sách hay, những bài báo mới, những kiến thức về cuộc sống và xã hội được người tiêu dùng cập nhật thường xuyên, liên tục, không nhàm chán và điều quan trọng nữa là rất tiện lợi về thời gian và chi phí.

Chính vì vậy, ngành hàng kinh doanh văn phòng phẩm truyền thống ngày càng phát triển chậm dần, khó phát triển vì không đủ điều kiện đáp ứng được các yếu tố nhanh, gọn, tiện lợi, không mất quá nhiều thời gian đi lại, lựa chọn, thực tế có thể thấy trong những năm qua.

Để thay đổi cho phù hợp với xu thế hiện nay, bắt buộc ngành hàng văn phòng phẩm truyền thống phải có những nghiên cứu, có những bước đi mới phù hợp và thu hút, níu kéo lại khách hàng của mình bằng cách đổi mới tu duy nhìn nhận, đổi mới chương trình, đổi mới cách trình bày, đổi mới sự tư vấn và cung cách phục vụ khách hàng...và thậm chí là mở thêm kênh online để phục vụ khách hàng chuyên dùng mạng internet.

## 1.2 Mục tiêu của đề tài

* Xây dựng một hệ thống giúp khách hàng có thể đặt mua sản phẩm một cách nhanh chóng và tiện lợi, dễ dàng tìm kiếm sản phẩm trong cửa hàng, hỗ trợ thanh toán trực tuyến thông qua ví điện tử.
* Đối với quản lý cửa hàng hệ thống giúp quản lý sản phẩm, đơn hàng cũng như doanh thu, lợi nhuận của cửa hàng, cập nhật tình hình hoạt động của cửa hàng từ đó làm tiền đề phục vụ tốt nhất cho khách hàng và kinh doanh.

## 1.3 Kết quả phải đạt

* Hỗ trợ thanh toán trực tuyến.
* Khách hàng có thể lập giỏ hàng, theo dõi đơn hàng, đánh giá sản phẩm sau khi mua.
* Quản lý có thể quản lý khách hàng, nhân viên, sản phẩm, danh mục, thương hiệu và đơn hàng.
* Quản lý xem báo cáo thống kê doanh thu, lợi nhuận.
* Quản lý phân công nhân viên giao hàng và in hoá đơn.
* Nhân viên vận chuyển xác nhận giao hàng thành công.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Giới thiệu công cụ lập trình

### 2.1.1 Giới thiệu về IntelliJ IDEA

IntelliJ IDEA là một IDE Java để phát triển các phần mềm máy tính. Nó được phát triển bởi JetBrains, được cấp phép Apache 2 cho một phiên bản cộng đồng và một phiên bản thương mại độc quyền. Cả hai có thể được sử dụng cho phát triển các ứng dụng lập trình mang tính thương mại.

Các lợi ích của việc sử dụng 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.

### 2.1.2 Giới thiệu về Visual studio Code

Visual Studio Code (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.



## 2.2 Giới thiệu về hệ quản trị cơ sở dữ liệu Microsoft SQL Server

SQL Server hay còn gọi là Microsoft SQL Server, là một phần mềm quản trị cơ sơ dữ liệu quan hệ được phát triển bởi microsoft bao gồm: tạo, duy trì, phân tích dữ liệu... dễ dàng sử dụng để lưu trữ cho các dữ liệu dựa trên tiêu chuẩn RDBMS – Relational Database Management System.

SQL Server được xây dựng dựa trên SQL, được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn lên đến Tera – Byte cùng lúc phục vụ cho hàng ngàn user. SQL Server cung cấp đầy đủ các công cụ cho việc quản lý từ nhận diện GUI đến sử dụng ngôn ngữ cho việc truy vấn SQL.

SQL Server thường đi kèm với việc thực hiện riêng các ngôn ngữ SQL, T – SQL, cụ thể như sau: 

* **SQL Server Management Studio:** công cụ giao diện chính cho máy chủ, hỗ trợ cho môi trường 64-bit và 32 bit.
* **T – SQL:** là ngôn ngữ thuộc quyền sở hữu của Microsoft hay còn được gọi là Transact – SQL.

## 2.3 Giới thiệu về JAVA

### 2.3.1 Tổng quan về Java

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động. Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.J ava được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

### 2.3.2 Lợi ích của việc sử dụng Java

Ứng dụng viết bằng ngôn ngữ lập trình Java có thể chạy trên nhiều nền tảng khác nhau thông qua sự hỗ trợ của JVM (Java Virtual Machine).

Java là ngôn ngữ thuần hướng đối tượng, dễ đọc, cú pháp ràng buộc chặt chẽ. Hỗ trợ việc mở rộng, bảo trì phần mềm về sau.

Java hỗ trợ thực thi nhiều chương trình đồng thời, hỗ trợ đa luồng.

Java hỗ trợ nhiều tính năng đặc biệt như thu gom rác (Garbage collection), không sử dụng con trỏ (Pointer), hỗ trợ xử lý ngoại lệ (Exception).

### 2.3.3 Tổng quan về Java Spring Boot

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.

### 2.3.4 Tổng quan về JPA

JPA là viết tắt của Java Persistence API, là một tập hợp các tiêu chuẩn của Java để làm việc với cơ sở dữ liệu quan hệ.

JPA có thể xem như cầu nối giữa Java object và cơ sở dữ liệu quan hệ. Là một tập hợp các đặc tả (các interface), JPA không chứa bất kỳ phương thức thực thi nào, nó cần một JPA implementation triển khai tất cả các đặc tả mà nó định nghĩa. Những ORM tools như Hibernate, TopLink đều cung cấp trình triển khai cho JPA.

ORM viết tắt của object-relational-mapping, công nghệ cho phép chuyển đổi từ các object trong ngôn ngữ hướng đối tượng sang database quan hệ và ngược lại.

Hibernate là một trong những ORM tools phổ biến được sử dụng nhiều trong các ứng dụng Java. Từ phiên bản 3.2 trở về sau, hibernate đã cung cấp một JPA Implement và được sử dụng khá rộng rãi trong cộng đồng Java, trong chương trình này chúng em cũng sử dụng JPA được chính Hibernate cung cấp.

### 2.3.5 Tổng quan về Spring Security

Spring Security thực sự chỉ là một loạt các bộ lọc servlet giúp bạn thêm authentication và authorization vào ứng dụng web của mình.

Spring Security cũng tích hợp tốt với các framework như Spring Web MVC (hay Spring Boot), cũng như với các tiêu chuẩn như OAuth2 hoặc SAML. Và nó tự động tạo các trang login / logout và bảo vệ chống lại các hành vi khai thác thông tin như CSRF.

Việc sử dụng Spring Security sẽ đảm bảo tính xác thực và phân quyền trong hệ thống, đảm bảo hệ thống hoạt động an toàn và đúng với mục đích đề ra, Spring Security hỗ trợ tạo ra những JWT Token-là chìa khóa đã thực hiện mọi thao tác trong chương trình.

## 2.4 Giới thiệu về ReactJS

ReactJS là một thư viện JavaScript phổ biến nhất hiện nay được dùng để xây dựng giao diện của người dùng (UI – User interfaces). Được ra mắt vào năm 2013 phát triển bởi Facebook theo chuẩn SPA (Single Page Application) mang đến giải pháp hiệu quả, đơn giản trong việc xây dựng giao diện người dùng với tốc độ phản hồi nhanh mượt mà. ReactJS còn cho phép lập trình viên phân tách những thành phần phức tạp thành các phần nhỏ lẻ, đơn giản để giúp cho việc phát triển và bảo trì tái sử dụng hiệu quả hơn.

ReactJS có ưu điểm dễ tiếp cận, có khả năng tái sử dụng các component đã khai báo trước đó để phát triển các ứng dụng khác có cùng chức năng giúp cho tiến độ công việc nhanh chóng hơn và code không bị trùng lập, dễ dàng tích hợp và kiểm thử, cùng với nhiều thư viện css hỗ trợ như bootstrap… khiến giao diện đẹp mắt và nhanh chóng phát triển hơn.

ReactJS hoạt động dựa trên sự kết hợp HTML và Javascript. Chúng ta có thể chèn hoặc viết code vào thẳng HTML bằng Javascript với giải pháp này. Hầu hết các phần trong ReactJS đều được viết bởi JSX (Javascript XML) để giúp cho việc tạo các components dễ dàng hơn.

JSX là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào.

State trong ReactJS giống như một kho lưu trữ dữ liệu cho các component trong ReactJS. Nó chủ yếu được sử dụng để cập nhật component khi người dùng thực hiện một số hành động như nhấp vào nút, nhập một số văn bản, nhấn một số phím, …

ReactJS được cộng đồng lập trình đánh giá cao và có tương lai, với sự hỗ trợ từ Facebook đã giúp nó vượt qua tất cả các thư viện khác hiện có.

### 2.4.1 Các thành phần chính có trong một ReactJS

**Component**: là thành phần cơ bản của ReactJS. Component giúp chia nhỏ các thành phần UI độc lập với mục đích là dễ quản lý và tái sử dụng nó. ReactJS kiểm soát luồng dữ liệu thông qua states và props được dùng để render ra các component với dữ liệu động.

**State**: là thành phần lưu trữ các giá trị của component được xem như kho lưu trữ dữ liệu cho các component, khi có thay đổi thì component chứa state sẽ được render lại. Để có thể cập nhật state phải dùng phương thức setState và không thể gán trực tiếp như kiểu này this. state = {‘key’: ‘value’}.

**Props**: dùng để gửi dữ liệu đến các component con, props là bất biến (không thể thay đổi). Vì đã được phát triển trong khái niệm về các hàm thuần khiết. Với các hàm thuần khiết không thể thay đổi dữ liệu của các tham số. Do đó, không thể thay đổi dữ liệu của prop trong ReactJS.

**Hooks**: cho phép sử dụng state và các tính năng khác của ReactJS mà không phải dùng đến Class. Hooks giúp kết nối React state và lifecycle vào các components sử dụng hàm, với cách dùng đơn giản Hooks giúp giảm đáng kể số lượng code, dễ dàng sử dụng và khiến cho các component trở nên gọn nhẹ hơn.

Một số thư viện, package khác:

* + **React hook form**: là một thư viện giúp quản lý và xác thực các form trong React. Nó cung cấp các cách tiếp cận đơn giản và hiệu quả để xử lý các và kiểm tra các dữ liệu trong form trong ứng dụng React.
  + **React-query**: là một thư viện quản lý trạng thái và dữ liệu trong React. Nó được thiết kế để giúp bạn quản lý dữ liệu từ xa một cách dễ dàng và hiệu quả. Nó có thể xử lý việc tìm nạp, lưu vào bộ nhớ đệm, đồng bộ hoá và cập nhật trạng thái.
  + **Axios**: Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho việc xử lý XHR (XMLHttpRequests).

## 2.5 Giới thiệu về Restful API

REST là viết tắt của Representational State Transfer. RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (Representational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

Những thuận lợi khi xây dựng dịch vụ web với RESTful:

* Các RESTfull web services là độc lập đa nền tảng.
* Có thể được viết với bất cứ ngôn ngữ lập trình nào và có thể được thực thi trên bất cứ nền tảng nào đó.
* Cung cấp đa dạng các định dạng dữ liệu như JSON, HTML, XML hay text.
* Tốc độ xây dựng nhanh chóng hơn khi so sánh với SOAP – một trong những phương pháp truyên thống khi xây dựng web API.
* Có tính tái sử dụng cao.
* Trung lập về ngôn ngữ.

# CHƯƠNG III: KHẢO SÁT HIỆN TRẠNG

## 3.1 Hiện trạng thực tế

Ngày nay, với sự phát triển của công nghiệp hóa và hiện đại hóa, cuộc sống của chúng ta trở nên ngày càng bận rộn. Khả năng tìm kiếm, so sánh giá cả và mua sắm các sản phẩm văn phòng phẩm trở nên khó khăn và tốn thời gian đối với nhiều người. Thời gian làm việc kéo dài và cửa hàng thường đóng cửa khi người làm việc kết thúc ca làm, tạo ra rào cản cho việc mua sắm.

Đối với những người không có thời gian vào cửa hàng trong giờ làm việc hoặc nơi sống cách xa cửa hàng, việc mua sắm trở nên ngày càng khó khăn. Thêm vào đó, việc theo dõi các chương trình khuyến mãi và giảm giá tại cửa hàng trở nên khó khăn do thời gian hạn chế.

Việc quản lý danh sách sản phẩm và tồn kho cũng là một thách thức đối với các chủ cửa hàng. Khó khăn trong việc nắm bắt số lượng tồn kho hiện tại, doanh thu hàng tuần và hàng tháng, và xác định sản phẩm nào đang bị tồn cùng có thể dẫn đến sự không hiệu quả trong quản lý kinh doanh và đáp ứng nhu cầu của khách hàng một cách tốt nhất.

Tất cả những vấn đề này có thể dẫn đến giảm doanh thu và số lượng sản phẩm bán ra của cửa hàng. Để vượt qua những thách thức này, cửa hàng cần xem xét và áp dụng các giải pháp hiện đại như cửa hàng trực tuyến, dịch vụ giao hàng, và hệ thống quản lý tồn kho hiệu quả.

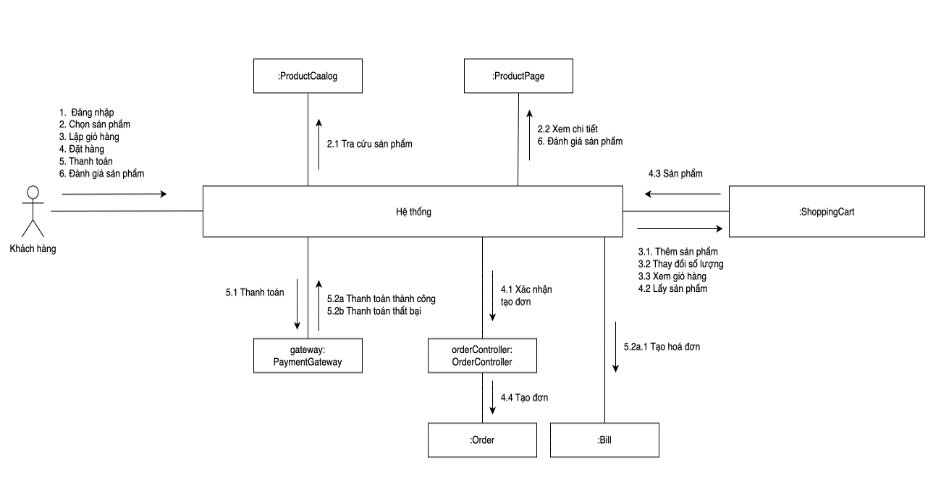
## 3.2 Giải pháp nâng cao hiệu quả cho việc sử dụng website bán hàng online

Những vấn đề đặt ra trên đây cho thấy, cần có một giải pháp để nâng cao hiệu quả hoạt động để tăng doanh thu và đáp ứng yêu cầu khách hàng một cách tốt nhất.

Một trong các giải pháp để nâng cao hiệu quả là công nghệ việc bán hàng và vận hành. Từ những vấn đề phát hiện, ta có thể có các giải pháp tương ứng sau đây:

|  |  |
| --- | --- |
| **Vấn đề** | **Giải pháp** |
| Thống kê doanh thu, lợi nhuận khó khăn đối với người quản lý | Chương trình tự động cập nhật số lượng tồn kho và doanh thu hàng tuần và hàng tháng. |
| Khó khăn trong việc theo dõi các chương trình khuyến mãi và giảm giá. | Chương trình cung cấp thông tin về các chương trình khuyến mãi và giảm giá đang diễn ra. Hơn nữa, hệ thống có thể tự động áp dụng các ưu đãi cho đơn hàng của họ. |
| Khó khăn trong việc tìm kiếm sản phẩm. | Chương trình có giao diện thân thiện và chức năng tìm kiếm tiện lợi. Khách hàng có thể truy cập trang web này bất kỳ lúc nào và tìm kiếm sản phẩm dễ dàng. |
| Khó khăn trong việc đến tận nơi để mua sản phẩm | Khách hàng có thể đặt hàng trực tuyến và chọn dịch vụ giao hàng đến địa chỉ của họ. |

Như vậy ta có thể phát triển website bán văn phòng phẩm cho phép người dùng có thể tìm đến mọi lúc mọi nơi, có thể đặt hàng một cách nhanh chóng và không mất quá nhiều thời gian cho việc tìm kiếm và mua sản phẩm cần thiết. Chương trình giúp khách hàng dễ tiếp cận và người quản lý thì có thể quản lý một cách hiệu quả hơn giúp doanh thu và số lượng bán ra sẽ được nâng cao hơn so với cách dùng truyền thống.



Hình 3.1: Sơ đồ cộng tác khách hàng

Khách hàng:

1) Đăng nhập vào ứng dụng.

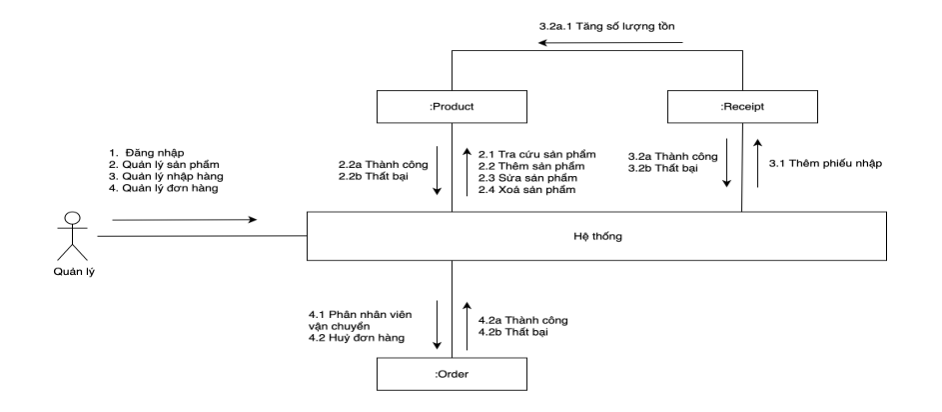
2) Xem danh sách sản phẩm và chi tiết từng sản phẩm.

3) Lập giỏ hàng (thêm sản phẩm, chỉnh sửa số lượng, xoá sản phẩm khỏi giỏ hàng).

4) Từ trang giỏ hàng, khách chọn các sản phẩm muốn mua và ấn xác nhận → sản phẩm từ giỏ hàng được chuyển vào đơn hàng.

5) Di chuyển đến trang thanh toán, khách chọn hình thức thanh toán và nhấn thanh toán → hệ thống báo thành công/thất bại.

6) Sau khi nhận được hàng khách hàng có thể đánh giá sản phẩm đã mua.

 Hình 3.2: Sơ đồ cộng tác quản lý

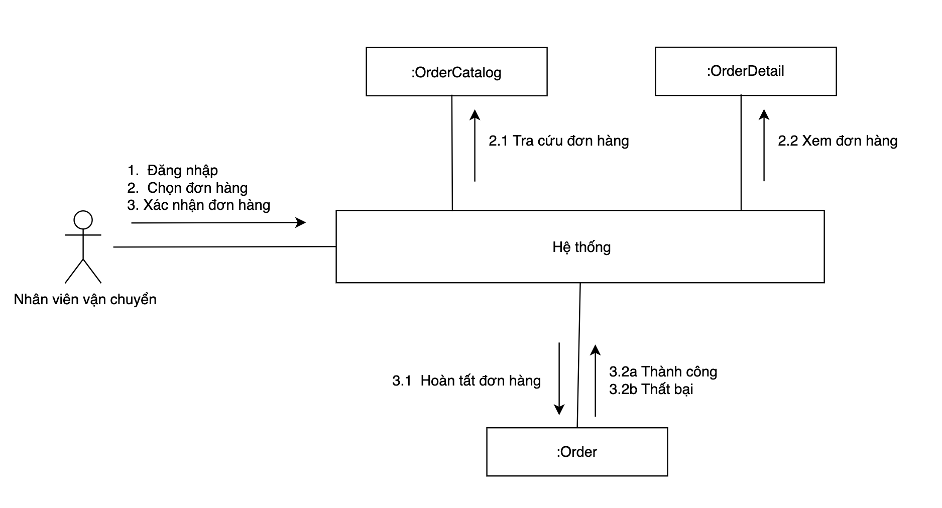
Quản lý:

1) Đăng nhập vào hệ thống

2) Quản lý sản phẩm (thêm, sửa, xóa) → hệ thống phản hồi thành công hay thất bại

3) Quản lý nhập hàng (thêm) → hệ thống phản hồi thành công hay thất bại

4) Quản lý đơn hàng (giao cho nhân viên vận chuyển, huỷ đơn hàng) → hệ thống phản hồi thành công hay thất bại



Hình 3.2: Sơ đồ cộng tác nhân viên vận chuyển

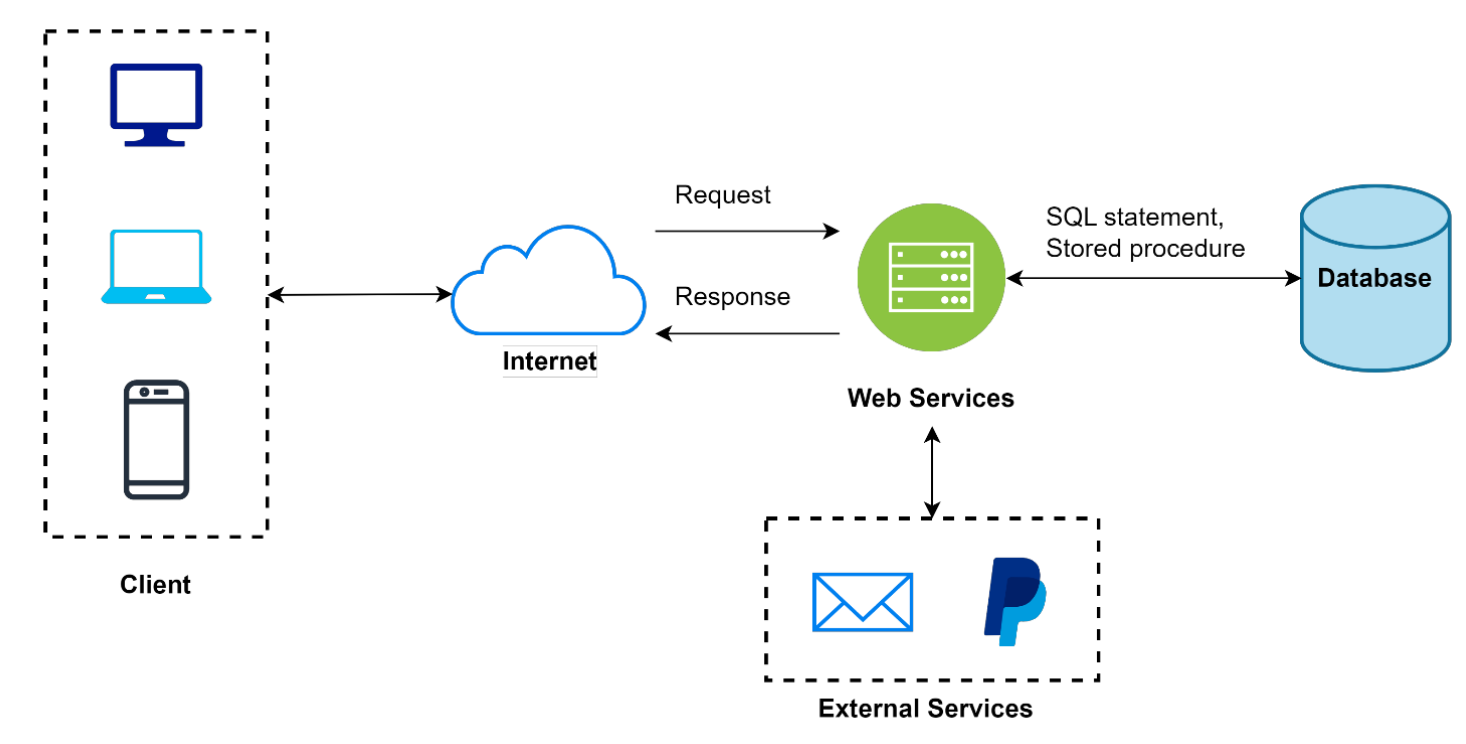
Nhân viên vận chuyển:

1) Đăng nhập vào hệ thống

2) Tra cứu đơn hàng và chọn đơn hàng

3) Quản lý đơn hàng (hoàn tất đơn hàng) → hệ thống phản hồi thành công hay thất bạị

## 3.3 Kiến trúc vật lý của hệ thống

**

Hình 3.3: Sơ đồ kiến trúc vật lý

## 3.4 Yêu cầu chức năng

### 3.4.1 Hệ thống cho phép quản lý:

* Quản lý nhân viên.
* Quản lý khách hàng.
* Quản lý sản phẩm.
* Lập báo cáo doanh thu.
* Quản lý đơn hàng, phân công nhân viên giao hàng.
* Lập hoá đơn.
* Quản lý nhập hàng.
* Quản lý thay đổi giá.
* Quản lý khuyến mãi.
* Quản lý poster quảng cáo.
* Quản lý danh mục sản phẩm.
* Quản lý thương hiệu sản phẩm.

### 3.4.2 Hệ thống cho phép nhân viên giao hàng:

* Xem danh sách đơn hàng được phân công.
* Xác nhận giao hàng thành công.

### 3.4.3 Hệ thống cho phép khách hàng:

* Đặt hàng, thanh toán trực tuyến.
* Theo dõi đơn hàng.
* Quản lý giỏ hàng.
* Cập nhật thông tin cá nhân.
* Đánh giá sản phẩm đã mua.

### 3.4.4 Hệ thống cho phép người dùng:

* Đăng nhập, đăng ký tài khoản.
* Tra cứu sản phẩm.

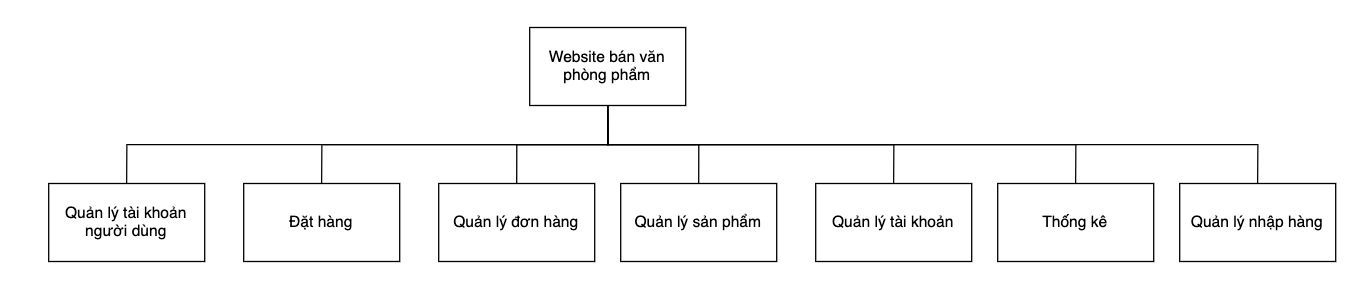
## 3.5 Yêu cầu phi chức năng

* Website đảm bảo tính an toàn, bảo mật thông tin cá nhân người dùng của hệ thống.
* Hệ thống cho phép người dùng truy cập hệ thống 24/7.
* Trang web cho phép tối đa 95% người dùng truy cập cùng một lúc. Hệ thống cho phép truy cập dữ liệu thời gian thực. Các tác vụ thực hiện tức thời trong thời gian ngừng cho phép chấp nhận dưới 30s.
* Hệ thống cho phép hỗ trợ đa trình duyệt phổ biến như Google Chrome, Cốc Cốc, Microsoft Edge…
* Thời gian trả về đối với các yêu cầu dưới 2 giây.
* Khi xảy ra các sự cố làm ngừng vận hành hệ thống, hệ thống phải đảm bảo phục hồi 90% trong vòng 1h và 100% trong vòng 24h.

# CHƯƠNG IV: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 4.1 Sơ đồ phân cấp chức năng

**4.1.1 Sơ đồ phân cấp chức năng mức đỉnh:**



Hệ thống nghiệp vụ được xét gồm các chức năng chính: Quản lý tài khoản người dùng, Đặt hàng, Quản lý đơn hàng, Quản lý sản phẩm, Quản lý nhập hàng, Quản lý tài khoản và Tổng hợp báo cáo. Mỗi chức này được phân rã thành một số chức năng nhỏ hơn mà thao tác thực hiện nó đủ đơn giản cho việc lập trình.

**4.1.2 Sơ đồ phân rã**

**4.1.2.1 Chức năng quản lý tài khoản người dùng**

A black background with white rectangles

Description automatically generated

Đăng ký: Khi khách hàng vào website, khách hàng có thể đăng ký tài khoản, khi đăng ký tài khoản khách hàng cần điền các thông tin cần thiết để tạo tài khoản ban đầu như: Họ và tên, email, số điện thoại, mật khẩu.

Đăng nhập: Khi khách hàng đã có tài tài khoản, khách hàng có thể tiến hành đăng nhập vào tài khoản.

Thêm xóa, sửa thông tin tài khoản: Khi khách hàng đã đăng nhập, khách hàng có thể tiến hành thêm các thông tin, chỉnh sửa những thông tin đã có sao cho phù hợp với mong muốn mỗi khách hàng.

Quên mật khẩu: Khách hàng khi đã có tài khoản trước đó có thể thực dùng chức năng quên mật khẩu để nhận được mật khẩu mới email đã đăng ký trước đó.

**4.1.2.2 Chức năng đặt hàng:**

A black and white screen with white rectangles

Description automatically generated

Tra cứu sản phẩm: Khi truy cập website, khách hàng có thể xem các chương trình khuyến mãi sản phẩm, các danh mục của sản phẩm cho khách hàng. Khách hàng có thể xem sản phẩm với hình ảnh mô tả, tên, giá, giá sau khuyến mãi (nếu có). Khách hàng có thể xem chi tiết thông tin sản phẩm.

Lập giỏ hàng: Khách hàng có thể lập giỏ hàng mà không cần đăng nhập trước, khách hàng có thể thay đổi sản phẩm, số lượng sản phẩm trong giỏ hàng. Giỏ hàng sẽ tính toán tổng giá sản phẩm người dùng cần thanh toán tương ứng với các sản phẩm và số lượng sản phẩm.

Tạo đơn hàng: Khách hàng có thể tạo đơn hàng sau khi đã đăng nhập. Đơn hàng sẽ lấy các thông tin có trong giỏ hàng khi người dùng ấn vào đặt hàng. Khách hàng có thể dùng thông tin của tài khoản đăng nhập để nhận hàng hoặc thay đổi thông tin người nhận hàng có đơn hàng đặt.

Thanh toán: Khách hàng có thể chọn hình thức thanh toán như thanh toán khi nhận hàng, thanh toán online qua Paypal.

**4.1.2.3 Chức năng quản lý sản phẩm:**

A black background with white rectangular objects

Description automatically generated

Thêm, xóa, sửa thông tin sản phẩm: Người quản trị có thể thêm, sửa các thông tin sản phẩm. Sản phẩm có thể xóa nếu chưa có ai từng đặt hàng. Người quản trị có thể vô hiệu sản phẩm nếu không muốn sản phẩm hiển thị trên website.

Thay đổi giá: người quản trị có thể thay đổi giá sản phẩm. Mỗi lần thay đổi giá thì hệ thống sẽ ghi nhận lại nhân viên thay đổi và thời gian thay đổi. Sản phẩm sẽ áp dụng giá gần nhất.

Thêm, xóa, sửa danh mục: Người quản trị có thể thực hiện thêm, sửa thông tin danh mục. Khi người quản trị xóa danh mục thì phải thỏa mãn điều kiện không có sản phẩm nào thuộc danh mục đó.

Thêm, xóa, sửa thương hiệu: Người quản trị có thể thực hiện thêm, sửa thông tin thương hiệu. Khi người quản trị xóa thương hiệu thì phải thỏa mãn điều kiện không có sản phẩm nào thuộc thương hiệu đó.

Thêm, xóa, sửa nhà cung cấp: Người quản trị có thể thực hiện thêm, sửa thông tin nhà cung cấp. Khi người quản trị xóa nhà cung cấp thì phải thỏa mãn điều kiện không có sản phẩm nào thuộc nhà cung cấp đó và chưa có đơn đặt hàng từ nhà cung cấp bị xóa.

Khuyến mãi: Mỗi khuyến mãi có nhiều sản phẩm khác nhau với các phần trăm khuyến mãi cho từng sản phẩm. Mỗi sản phẩm có thể thuộc nhiều chương trình khuyến mãi, một thời điểm chỉ có thể có một chương trình khuyến mãi chạy.

**4.1.2.4 Chức năng quản lý đơn hàng:**

A black and white background with white rectangles

Description automatically generated

Xác nhận đơn hàng: Người quản trị có thể xác nhận đơn hàng của khách hàng sau khi đặt để chuyển trạng thái đơn hàng sang đã xác nhận đơn hàng.

Phân công đơn hàng: Người quản trị có thể phân công người giao hàng sau khi đã xác nhận đơn hàng. Có thể chọn người giao hàng phù hợp với đơn hàng, hiển thị thông tin cơ bản của người giao hàng cho người quản trị chọn.

In hóa đơn: Người quản trị có thể xuất hóa đơn bán hàng ra file PDF để chuyển đến khách hàng.

Hủy đơn hàng: Người quản trị có thể hủy đơn hàng của khách hàng.

**4.1.2.5 Chức năng quản lý tài khoản:**

A black background with white rectangles

Description automatically generated

Xem thông tin khách hàng: Người quản trị có thể truy cập vào trang quản lý khách hàng, xem các thông tin của từng khách hàng.

Khóa tài khoản khách hàng: Người quản trị có thể khóa tài khoản khách hàng

Xem, sửa thông tin nhân viên: Người quản trị có thể xem, sửa thông tin nhân viên khi truy cập vào trang quản lý nhân viên của cửa hàng.

Khóa tài khoản nhân viên: Người quản trị có thể khóa tài khoản nhân viên.

**4.1.2.6 Chức năng thống kê:**

A black and white background with white rectangles

Description automatically generated

Thống kê doanh thu dạng biểu đồ: người quản trị có thể xem thống kê doanh thu dạng biểu đồ cột theo tháng.

Báo cáo doanh thu dạng bảng:  Người quản trị có thể xuất báo cáo doanh thu dạng bảng và xuất file PDF báo cáo doanh thu đó.

Thống kê số lượng đơn hàng: Người quản trị có thể xem số lượng đơn hàng đã đạt trong 30 ngày gần nhất.

Thống kê số lượng khách hàng mới: Người quản trị có thể xem số lượng khách hàng mới.

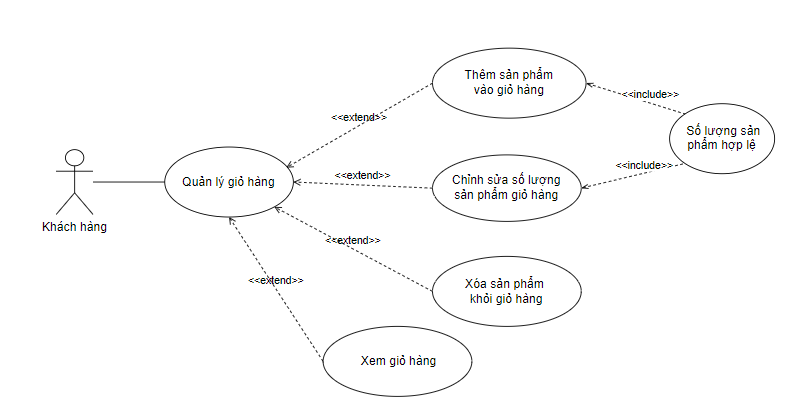
## 4.2 Lược đồ Usecase tổng quát

A diagram of a person's diagram

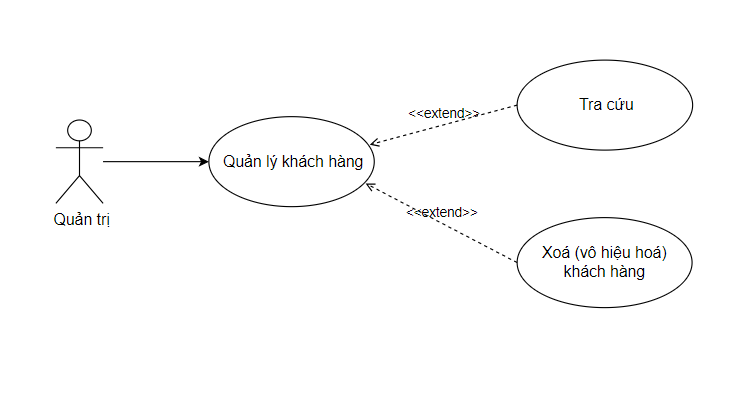
Description automatically generated

*Hình 4-2 Lược đồ usecase tổng quát*

### 4.2.1 Usecase quản lý giỏ hàng



### 4.2.2 Usecase quản lý khách hàng



### 4.2.3 Usecase theo dõi đơn hàng khách hàng

**A diagram of a diagram

Description automatically generated**

### 4.2.4 Usecase quản lý bình luận

**A diagram of a diagram

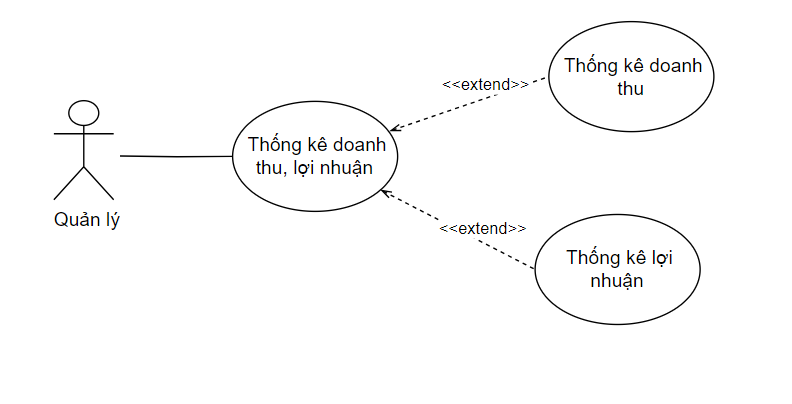
Description automatically generated**

### 4.2.5 Usecase quản lý thay đổi giá

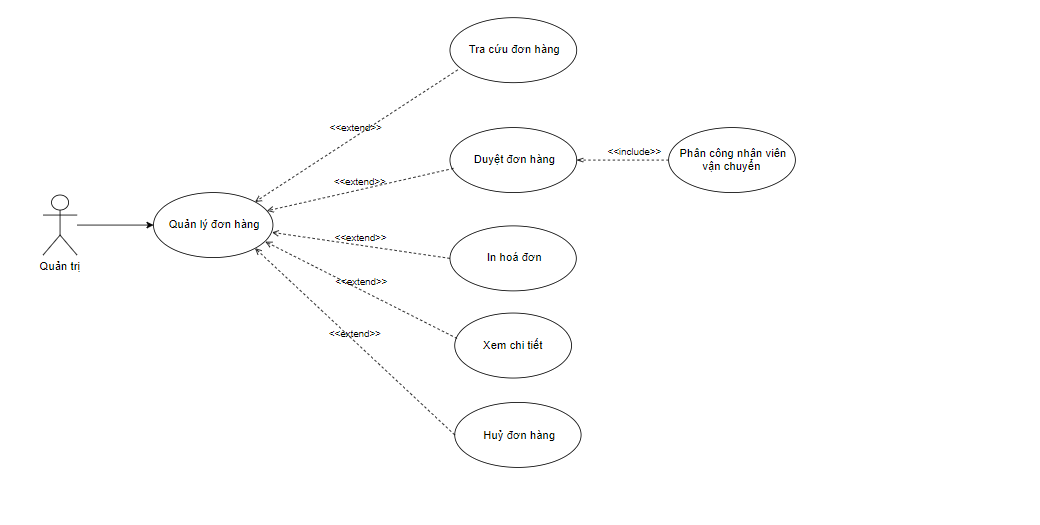
**A diagram of a diagram

Description automatically generated**

### 4.2.6 Usecase quản lý thống kê daonh thu, lợi nhuận

****

### 4.2.7 Usecase quản lý đơn hàng



### 4.2.8 Usecase quản lý nhập hàng

A diagram of a diagram

Description automatically generated

### 4.2.9 Usecase quản lý đơn hàng nhân viên

A diagram of a person

Description automatically generated

### 4.2.10 Usecase quản lý nhân viên

A diagram of a person with text

Description automatically generated

## 4.3 Mô hình dữ liệu quan hệ:

**4.3.1 Xác định các thực thể**

* **Role** (id\_role, name)
* **Account** (email, password, is\_active)
* **Staff** (id, name, phone, gender, address, avatar, created\_at)
* **Customer** (id, name, phone, gender, birthday, avatar, address, created\_at)
* **Supplier** (id, name, email, phone, address, website, created\_at, is\_active)
* **Category** (id, name, image, description, created\_at, is\_active)
* **Brand** (id, name, image, description, created\_at, is\_active)
* **Product** (id, name, description, in\_stock, created\_at, is\_active)
* **Image\_Product** (id, path)
* **The\_Order\_Status** (id, name)
* **The\_Order** (id, created\_at, payment\_id, received\_at, name, phone, address)
* **Invoice** (id, created\_at, name, tax\_code)
* **Return\_Product** (id, created\_at)
* **Purchase\_Order** (id, created\_at)
* **Receipt** (id, created\_at)
* **Promotion** (id, reason, end\_at, start\_at)
* **Review** (id, comment, vote, created\_at)
* **Poster** (id, name, type, is\_active, created\_at, image)

**4.3.2 Mô hình thực thể kết hợp ERD (Entity RelationShip Diagram)**

*A diagram of a product

Description automatically generated*

*Hình 4-3 Mô hình ERD bán hàng*

A diagram of a company

Description automatically generated

*Hình 4-4 Mô hình ERD nhập, trả hàng*

**4.3.3 Mô hình dữ liệu quan hệ**

* **Role (**id\_role, name**)**
* **Account** (email, password, *role\_id*, is\_active)
* **Staff** (id, *email*, name, phone, gender, address, avatar, created\_at)
* **Customer** (id, *email*, name, phone, gender, birthday, avatar, address, created\_at)
* **Supplier** (id, name, email, phone, address, website, created\_at, is\_active)
* **Supply\_Product** (*product\_id, supplier\_id*)
* **Category** (id, name, image, description, created\_at, is\_active)
* **Brand** (id, name, image, description, created\_at, is\_active)
* **Product** (id, name, description, in\_stock, created\_at, *brand\_id, category\_id*, is\_active)
* **Image\_Product** (id, path, *product\_id*)
* **The\_Order\_Status** (id, name)
* **The\_Order** (id, created\_at, *staff\_creator\_id, staff\_deliver\_id,* payment\_id, received\_at, *customer\_id*, *status\_id*, name, phone, address)
* **The\_Order\_Detail** (*order\_id, product\_id*, quantity, quantity\_return, unit\_price, *return\_id)*
* **Invoice** (id, created\_at, name, tax\_code, *the\_order\_id, staff\_id*)
* **Return\_Product** (id, created\_at*, invoice\_id, staff\_id*)
* **Purchase\_Order** (id, created\_at, *staff\_id, supplier\_id*)
* **Purchase\_Order\_Detail** (*purchase\_order\_id, product\_id*, quantity, unit\_price)
* **Receipt** (id, created\_at, *staff\_id, purchase\_order\_id*)
* **Receipt\_Detail** (*receipt\_id, product\_id*, unit\_price, quantity)
* **Promotion** (id, reason, end\_at, start\_at, *staff\_id*)
* **Promotion\_Detail** (*promotion\_id, product\_id*, percentage)
* **Review** (id, comment, vote, *customer\_id, product\_id*, created\_at)
* **Price\_History** (*product\_id, staff\_id*, apply\_at, price, created\_at)
* **Poster** (id, name, type, is\_active, created\_at, image, *staff\_id*)