**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**NGUYỄN KHÁNH THIỆN**

**KHÓA 15**

**HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**

**KHOÁ LUẬN**

**TỐT NGHIỆP ĐẠI HỌC**

**CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN QUẢN LÝ**

**XÂY DỰNG WEBSITE BÁN MỸ PHẨM ONLINE**

**HÀ NỘI, NĂM 2022**

**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**NGUYỄN KHÁNH THIỆN**

**KHÓA 15**

**HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**

**KHOÁ LUẬN**

**TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH: HỆ THỐNG THÔNG TIN**

**MÃ SỐ: 52480104**

**XÂY DỰNG WEBSITE BÁN MỸ PHẨM ONLINE**

***Cán bộ hướng dẫn: 4//, GVC, TS. Nguyễn Mạnh Hùng***

**HÀ NỘI, NĂM 2022**

|  |  |
| --- | --- |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **KHOA: CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN: HỆ THỐNG THÔNG TIN** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập -Tự do – Hạnh phúc** |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ và tên: **Nguyễn Khánh Thiện** Lớp: HTTTQL15 Khóa:15

Ngành: Hệ thống thông tin Chuyên ngành: Hệ thống thông tin quản lý

1. Tên đề tài: **Xây dựng website bán mỹ phẩm online**
2. Các số liệu ban đầu:

Dựa trên sự khảo sát quá trình bán mỹ phẩm, em đã tổng hợp và có được:

- Quy trình quản lý người dùng.

- Quy trình quản lý sản phẩm.

- Quy trình quản lý thanh toán hoá đơn.

1. Nội dung bản thuyết minh: bao gồm các chương:

Chương 1. Khảo sát hệ thống

Chương 2. Phân tích hệ thống

Chương 3. Thiết kế hệ thống

Chương 4. Xây dựng chương trình

1. Nội dung các bản vẽ:
   * Chương 1. Gồm các mô hình mô tả hệ thống, mô hình tiến trình nghiệp vụ, biểu đồ hoạt động.
   * Chương 2. Gồm các mô hình phân tích chức năng và phân tích dữ liệu.
   * Chương 3. Gồm các mô hình thiết kế hệ thống.
   * - Chương 4. Gồm các hình chụp giới thiệu phần mềm và giao diện phần mềm Hệ thống quản lý bán mỹ phẩm.
2. Cán bộ hướng dẫn:

Cán bộ hướng dẫn : Nguyễn Mạnh Hùng .

Học hàm, học vị : GVC, TS

Đơn vị : K12

Hướng dẫn toàn bộ.

Ngày giao: 29/10/2021 Ngày hoàn thành: 15/02/2022

*Hà Nội, ngày … tháng … năm….*

**Chủ nhiệm bộ môn** **Cán bộ hướng dẫn**

(Ký, ghi rõ họ tên, học hàm, học vị)

***GV, TS. Nguyễn Mạnh Hùng***

**Sinh viên thực hiện**

Đã hoàn thành và nộp đồ án ngày … tháng … năm ….

(Ký và ghi rõ họ tên )

Thiện

Nguyễn Khánh Thiện

MỤC LỤC

[LỜI NÓI ĐẦU 1](#_Toc95817112)

[Chương 1 KHẢO SÁT HỆ THỐNG 1](#_Toc95817113)

[1.1. Mô tả hệ thống 1](#_Toc95817114)

[**1.1.1. Nhiệm vụ cơ bản** 1](#_Toc95817115)

[**1.1.2. Cơ cấu tổ chức** 2](#_Toc95817116)

[**1.1.3. Quy trình xử lý** 2](#_Toc95817117)

[1.2.Mô hình hóa hệ thống 4](#_Toc95817118)

[**1.2.1.Mô hình tiến trình nghiệp vụ** 4](#_Toc95817119)

[**1.2.2.Biểu đồ hoạt động** 5](#_Toc95817120)

[Chương 2 PHÂN TÍCH HỆ THỐNG 9](#_Toc95817121)

[2.1. Phân tích chức năng nghiệp vụ 9](#_Toc95817122)

[**2.1.1. Mô hình hóa chức năng nghiệp vụ** 9](#_Toc95817123)

[**2.1.2. Sơ đồ luồng dữ liệu** 11](#_Toc95817124)

[**2.1.1** **Đặc tả tiến trình nghiệp vụ** 14](#_Toc95817125)

[2.2 Phân tích dữ liệu nghiệp vụ 19](#_Toc95817126)

[**2.2.1.Mô hình dữ liệu ban đầu** 19](#_Toc95817127)

[**2.2.1.1.Xác định kiểu thực thể, kiểu thuộc tính** 19](#_Toc95817128)

[**2.2.1.2.Mô hình thực thể liên kết mở rộng** 20](#_Toc95817129)

[**2.2.2. Chuẩn hóa dữ liệu** 22](#_Toc95817130)

[**2.2.3. Đặc tả dữ liệu** 28](#_Toc95817131)

[Chương 3 THIẾT KẾ HỆ THỐNG 34](#_Toc95817132)

[3.1. Thiết kế cơ sở dữ liệu 34](#_Toc95817133)

[3.2. Thiết kế kiểm soát 35](#_Toc95817134)

[3.2.1. Thiết kế nhóm người dùng 35](#_Toc95817135)

[3.2.2. Phân hạn nhóm người dùng về tiến trình 36](#_Toc95817136)

[3.3. Thiết kế cơ sở dữ liệu 37](#_Toc95817137)

[3.3.1. Mô hình dữ liệu hệ thống 37](#_Toc95817138)

[**3.3.2. Đặc tả bảng dữ liệu** 38](#_Toc95817139)

[Chương 4 XÂY DỰNG CHƯƠNG TRÌNH 43](#_Toc95817140)

[4.1. Mô tả hệ thống 43](#_Toc95817141)

[**4.1.1. Ngôn ngữ sử dụng** 43](#_Toc95817142)

[**4.2. Giới thiệu sản phẩm** 69](#_Toc95817143)

[KẾT LUẬN 76](#_Toc95817144)

[TÀI LIỆU THAM KHẢO 78](#_Toc95817145)

[1. Tài liệu internet 78](#_Toc95817146)

[1.2 2. Tài liệu sách 78](#_Toc95817147)

# **LỜI NÓI ĐẦU**

Sự phát triển mạnh mẽ của tin học làm cho máy tính trở thành phương tiện không thể thiếu được trong mọi lính vực đời sống. Hơn lúc nào hết tính năng của máy tính được khai thác một cách triệt để. Nếu như những năm trước máy tính ở nước ta được sử dụng chủ yếu như là một công cụ để soạn thảo văn bản thông thường, hoặc các công việc lập trình, quản lý phức tạp, xử lý dữ liệu bảng biểu, thương mại, khoa học… thì giờ đây, cùng với sự vươn xa của mạng Internet trên lãnh thổ Việt Nam máy tính còn là phương tiên có thể ngồi trên bàn làm việc cá nhân tại gia đình mà trao đổi thông tin liên lạc đi khắp toàn cầu.

Hệ thống thông tin được đề cập đến trong đồ án này là một phần mềm quản lý bán mỹ phẩm, giúp quản lý quá trình bán hàng trong ngành mỹ phẩm được thuận tiện hơn, tối ưu hoá thời gian công sức nhân công cũng như là gia tăng thu nhập so với quá trình làm thủ công như trước. Đó là lý do công nghệ thông tin ngày nay được sử dụng rộng rãi và phát triển trong cuộc sống ngày nay.

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô và bạn bè.

Em xin chân thành cảm ơn thầy Nguyễn Mạnh Hùng – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo đồ án tốt nghiệp này trong thời gian qua.

Bài báo cáo thực hiện trong khoảng thời gian 1 kì học. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót , em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Em xin chân thành cảm ơn!

**Chương 1  
KHẢO SÁT HỆ THỐNG**

## **1.1. Mô tả hệ thống**

### **1.1.1. Nhiệm vụ cơ bản**

Các nhiệm vụ cơ bản của hệ thống là:

- Quản lý sản phẩm và danh mục sản phẩm.

- Quản lý quá trình mua sản phẩm và thanh toán sản phẩm.

- Quản lý quá trình bình luận và đánh giá sản phẩm.

### **1.1.2. Cơ cấu tổ chức**

Gồm có:

+ Bộ phận quản trị (ADMIN)

+ Bộ phận người dùng (USER)

Nhiệm vụ của các bộ phận là *:*

+ Bộ phận quản trị (ADMIN)

- Quản lý thêm, sửa, xoá sản phẩm.

- Quản lý thêm, sửa, xoá danh mục sản phẩm.

- Quản lý hoá đơn mua sản phẩm.

- Quản lý nhà cung cấp.

- Quản lý thương hiệu.

- Quản lý quyền tài khoản hệ thống người dùng.

- Quản lý thống kê.

+ Bộ phận người dùng (USER)

- Xem và theo dõi sản phẩm.

- Mua và thanh toán sản phẩm.

- Xem và theo dõi đơn hàng.

- Quản lý giỏ hàng.

- Đăng ký, đăng nhập hệ thống sản phẩm.

### **1.1.3. Quy trình xử lý**

**1.1.3.1. Quy trình thêm sản phẩm, danh mục sản phẩm vào hệ thống**

- Quản trị viên đăng nhập vào hệ thống ứng dụng web được cung cấp.

- Quản trị viên chuyển qua màn hình quản lý để tiến hành thêm sản phâm.

- Quản trị viên nhập các thông tin cần thiết để tạo sản phẩm:

+ Tên sản phẩm.

+ Mô tả sản phẩm.

+ Giá.

+ Số lượng.

+ Ảnh sản phẩm.

+ Thương hiệu

+ Nhà cung cấp.

- Quản trị viên nhập các thông tin cần thiết để tạo danh mục sản phẩm:

+ Tên danh mục sản phẩm.

+ Ảnh danh mục sản phẩm.

**1.1.3.2. Quy trình quản lý nhà cung cấp, thương hiệu**

- Quản trị viên đăng nhập vào hệ thống ứng dụng web được cung cấp.

- Quản trị viên chuyển qua màn hình quản lý để tiến hành thêm nhà cung cấp, thương hiệu

- Quản trị viên nhập các thông tin cần thiết để thêm nhà cung cấp:

+ Tên nhà cung cấp.

+ Ảnh nhà cung cấp (nếu có).

- Quản trị viên nhập các thông tin cần thiết để thêm thương hiệu:

+ Tên thương hiệu.

+ Ảnh thương hiệu (nếu có).

**1.1.3.2. Quy trình quản lý quyền người dùng**

- Quản trị viên đăng nhập vào hệ thống ứng dụng web được cung cấp.

- Quản trị viên chuyển qua màn hình quản lý để tiến hành điều chỉnh quyền người dùng.

**1.1.3.4. Quy trình quản lý mua sản phẩm**

- Người dùng đăng nhập vào sản phẩm của hệ thống để tiến hành

- Người dùng chọn sản phẩm muốn mua để thêm vào giỏ hàng.

- Người dùng tiến hành thanh toán:

+ Nhập đầy đủ thông tin cá nhân, địa chỉ của người nhận hàng.

+ Chọn hình thức thanh toán để tiến hành mua sản phẩm.

- Sau khi thanh toán thành công người dùng sẽ nhận đc đơn hàng về sản phẩm mới mua để theo dõi.

**1.1.3.5. Quy trình bình luận, đánh giá về sản phẩm**

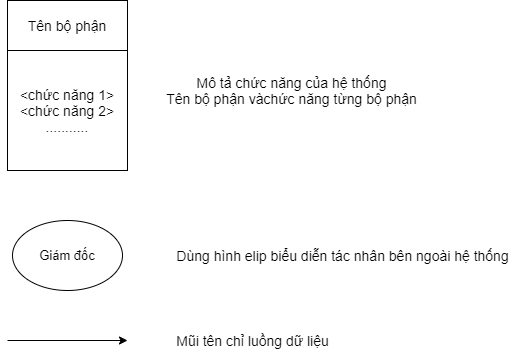
- Người dùng bắt buộc phải đăng nhập để có thể xem và bình luận của sản phẩm.

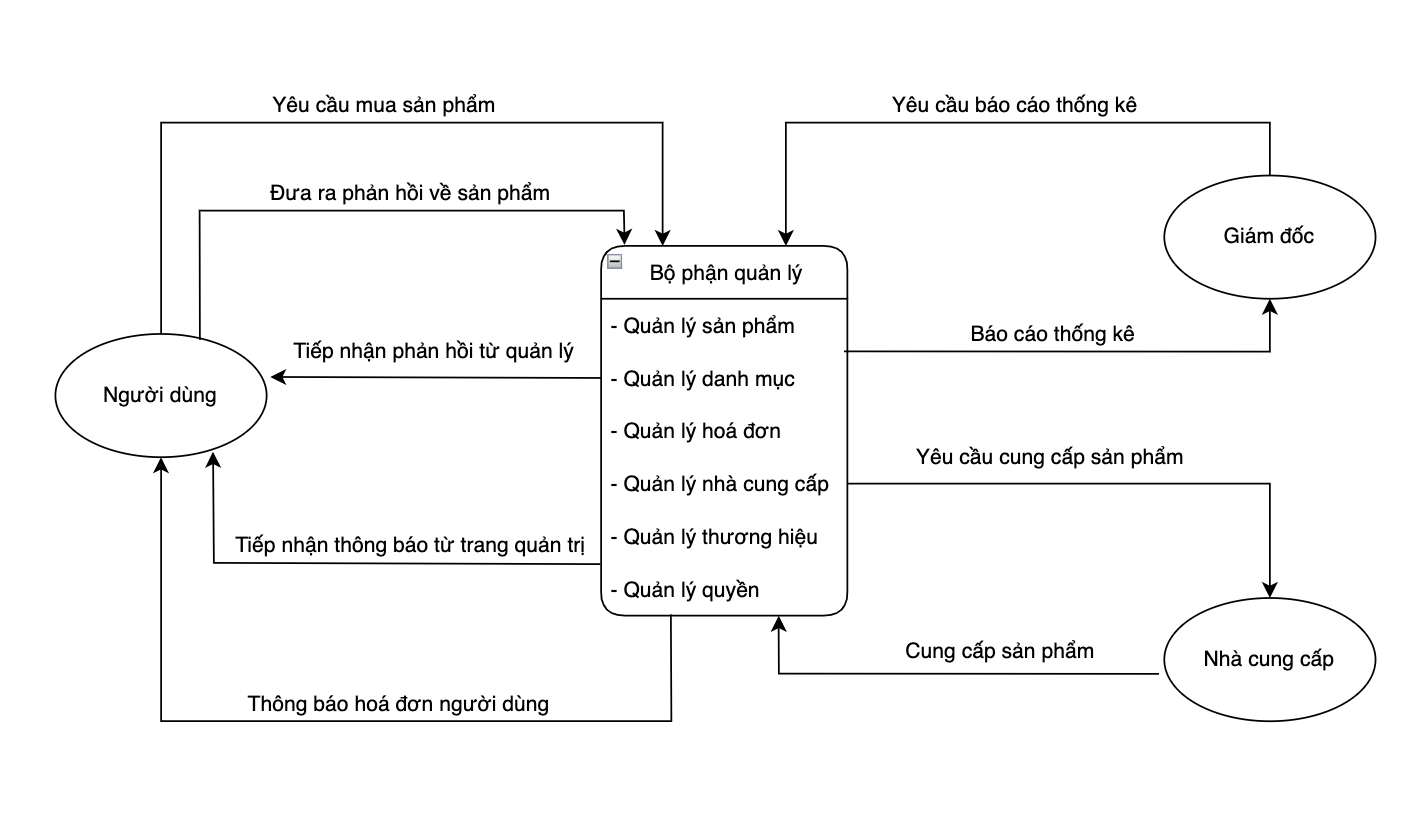
- Người dùng bắt buộc phải đăng nhập và mua sản phẩm để có thể đánh giá về sản phẩm.

- Sau khi người dùng bình luận và đánh gía về sản phẩm, người quản trị sẽ tiếp nhận ý kiến của ngừoi dùng.

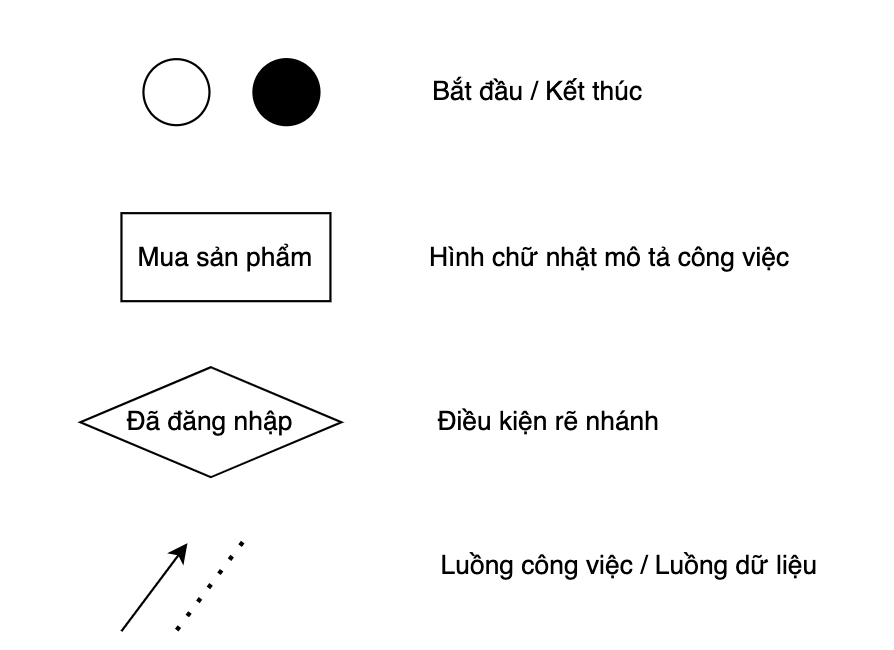
**1.2.Mô hình hóa hệ thống**

**1.2.1.Mô hình tiến trình nghiệp vụ**

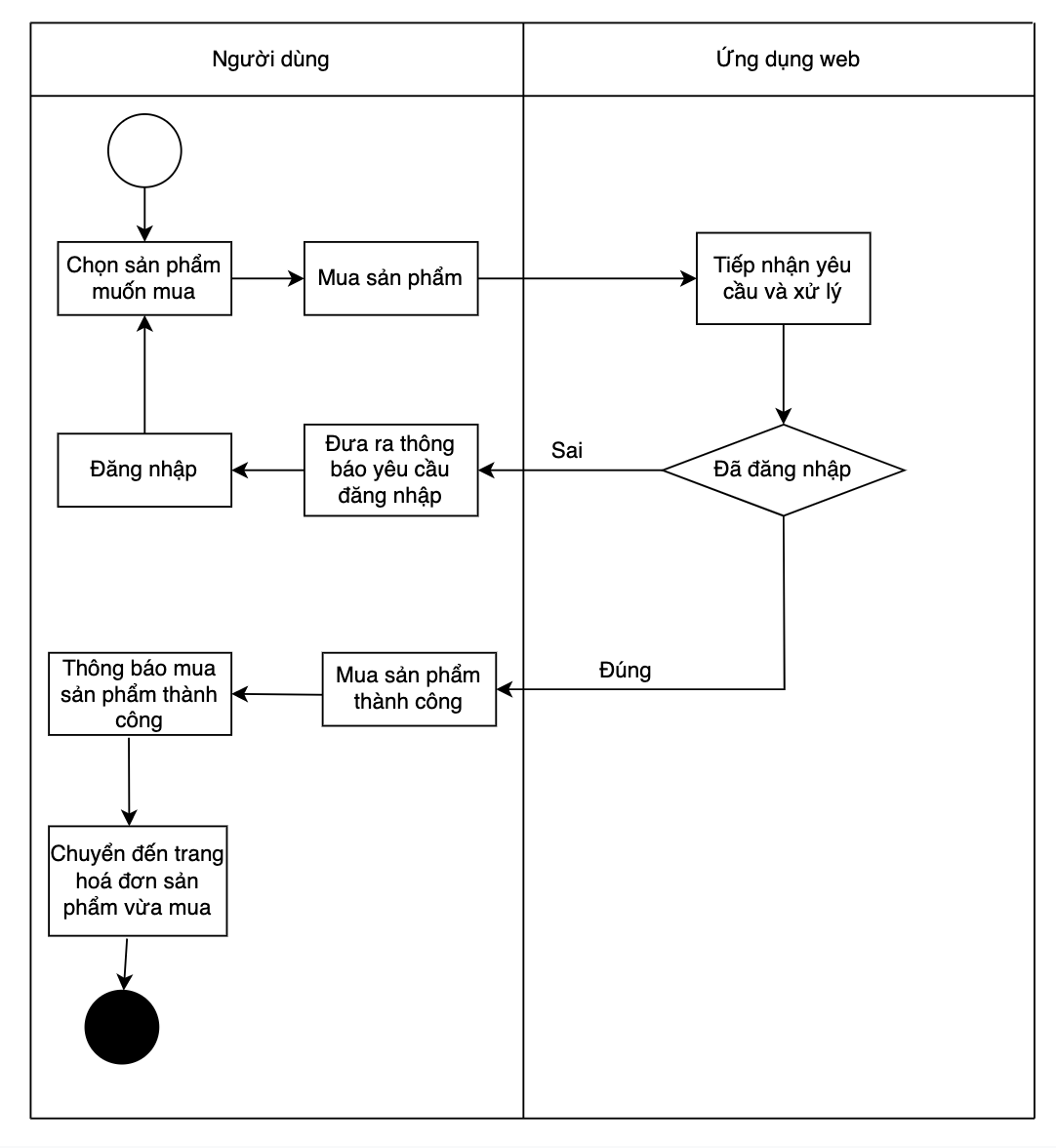
****

****

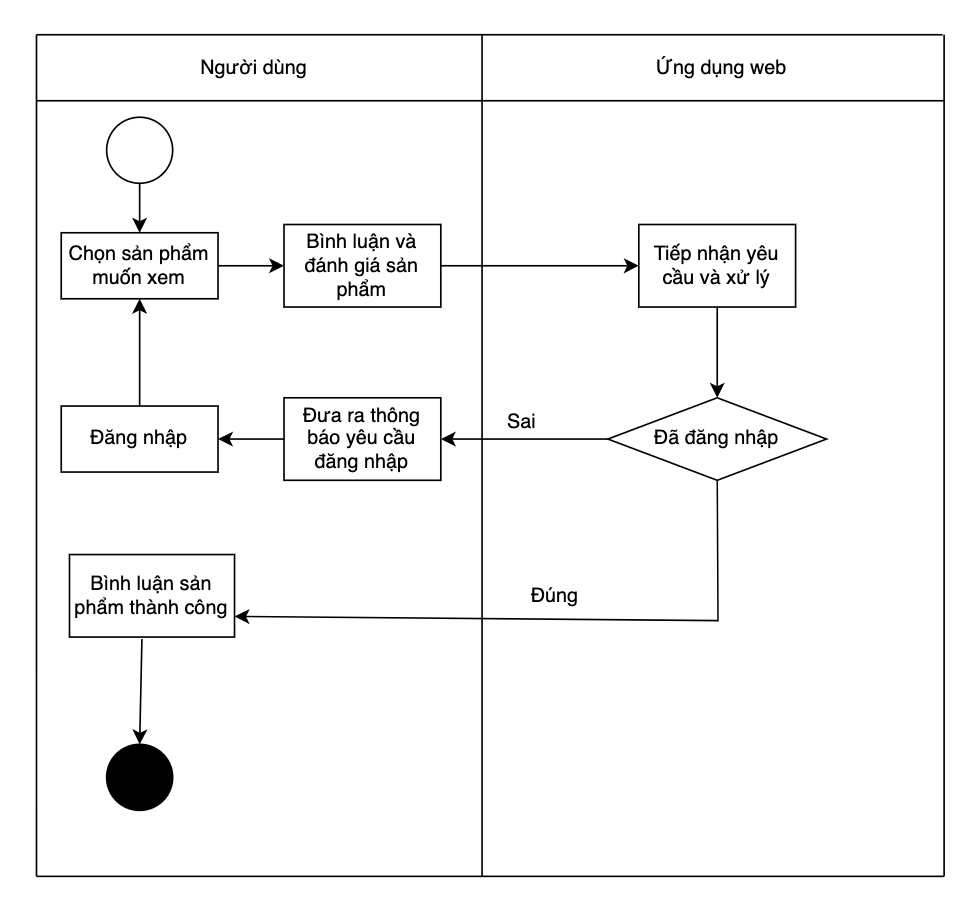
### **1.2.2.Biểu đồ hoạt động**



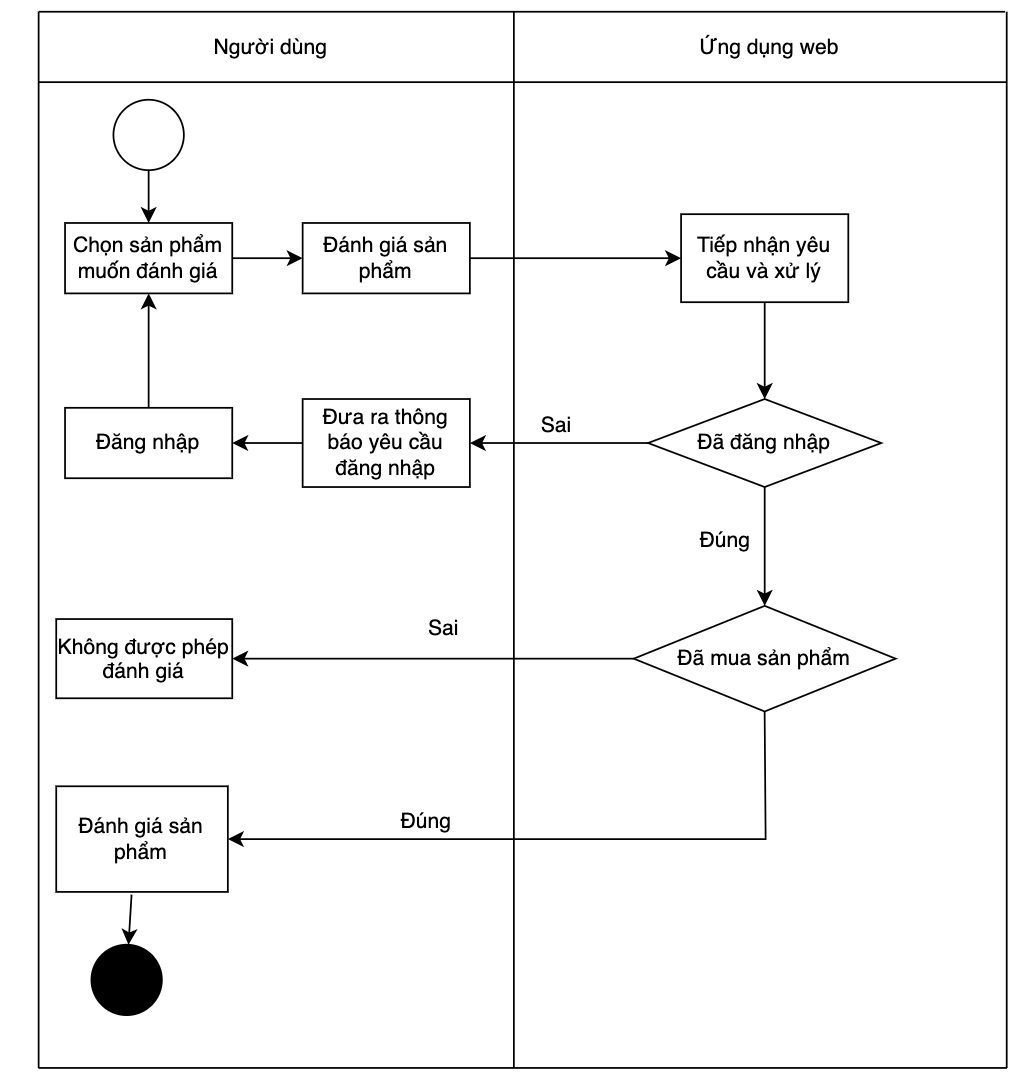
**a. Mua sản phẩm**

****

**b.Bình luận sản phẩm**

****

**c. Đánh giá sản phẩm**

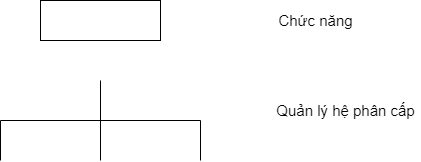
****

**Chương 2  
PHÂN TÍCH HỆ THỐNG**

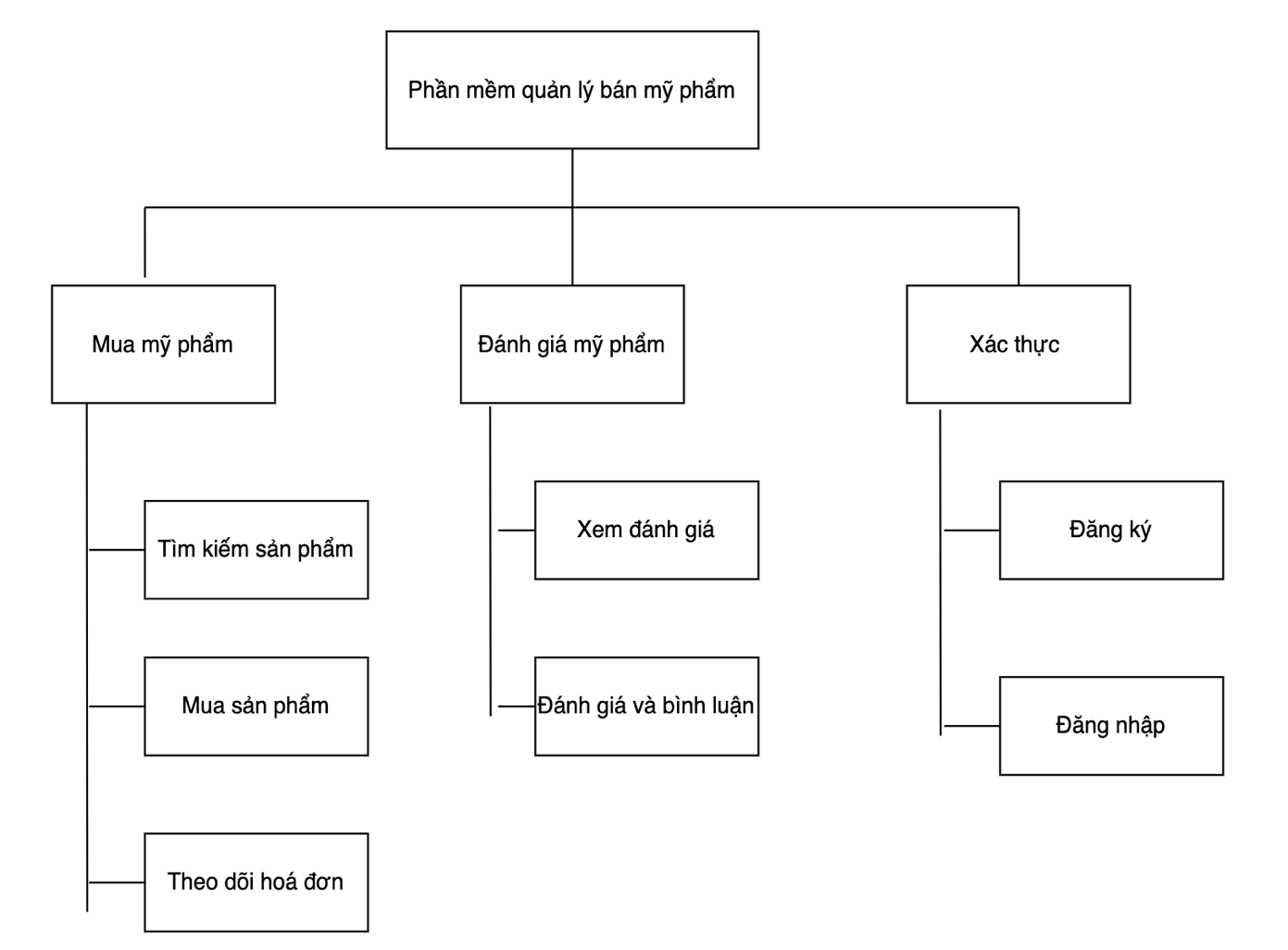
## **2.1. Phân tích chức năng nghiệp vụ**

### **2.1.1. Mô hình hóa chức năng nghiệp vụ**

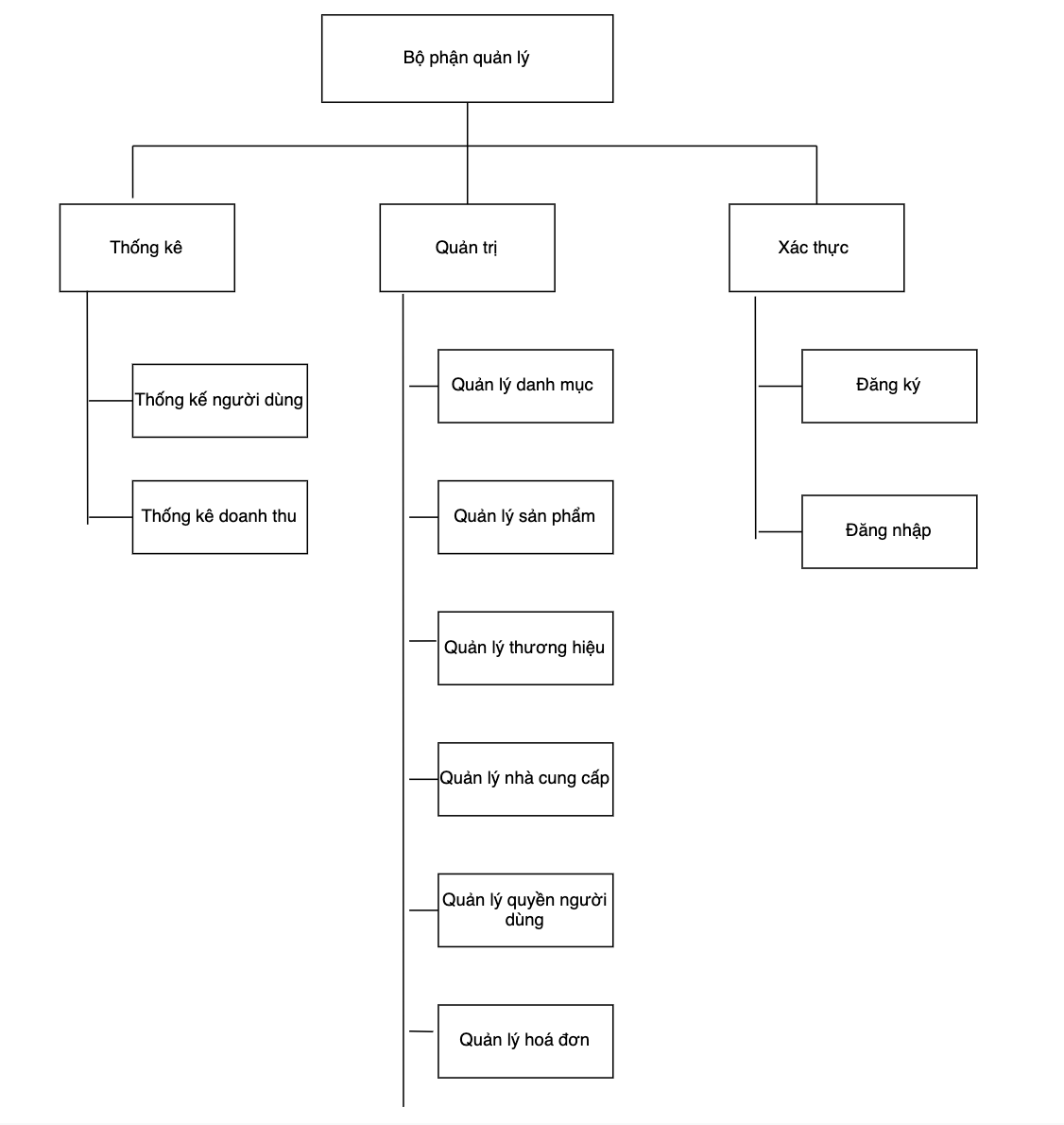
**2.1.1.1.Kí hiệu sử dụng**



**2.1.1.2. Sơ đồ phân rã chức năng dành cho người dùng web app**

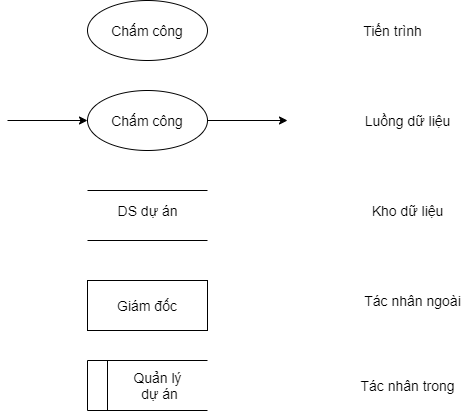


**2.1.1.3. Sơ đồ phân rã chức năng dành cho người dùng web admin**

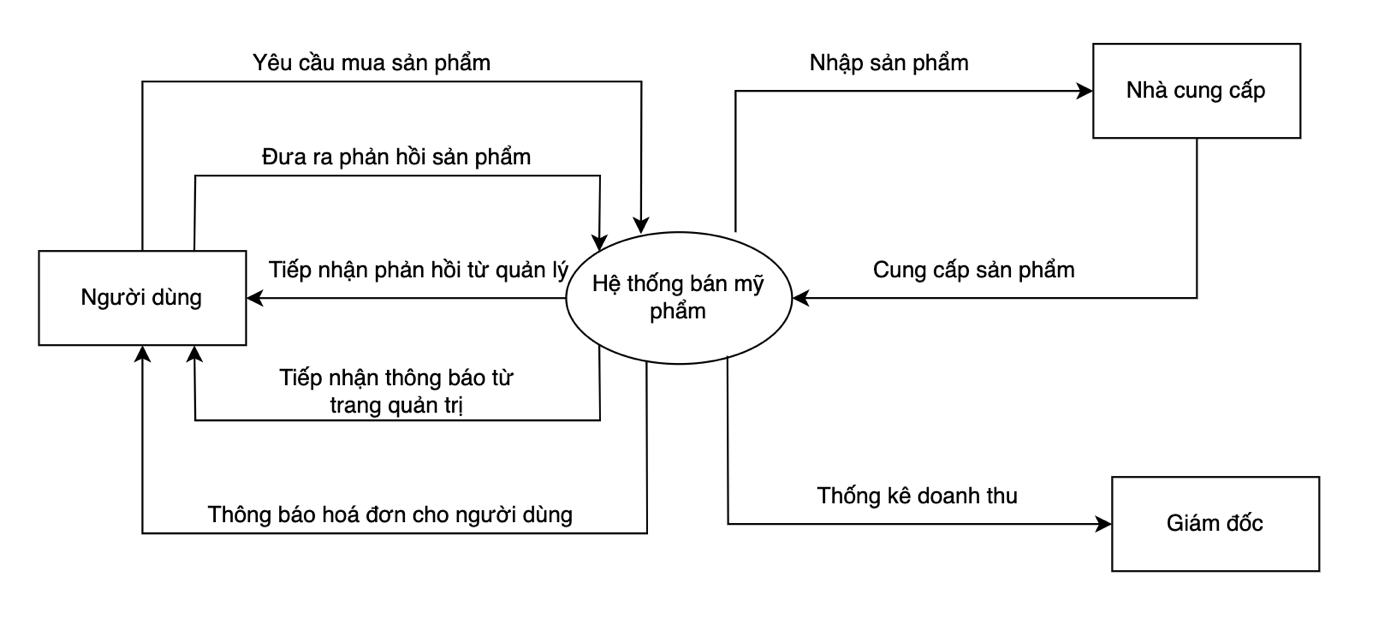


### **2.1.2. Sơ đồ luồng dữ liệu**

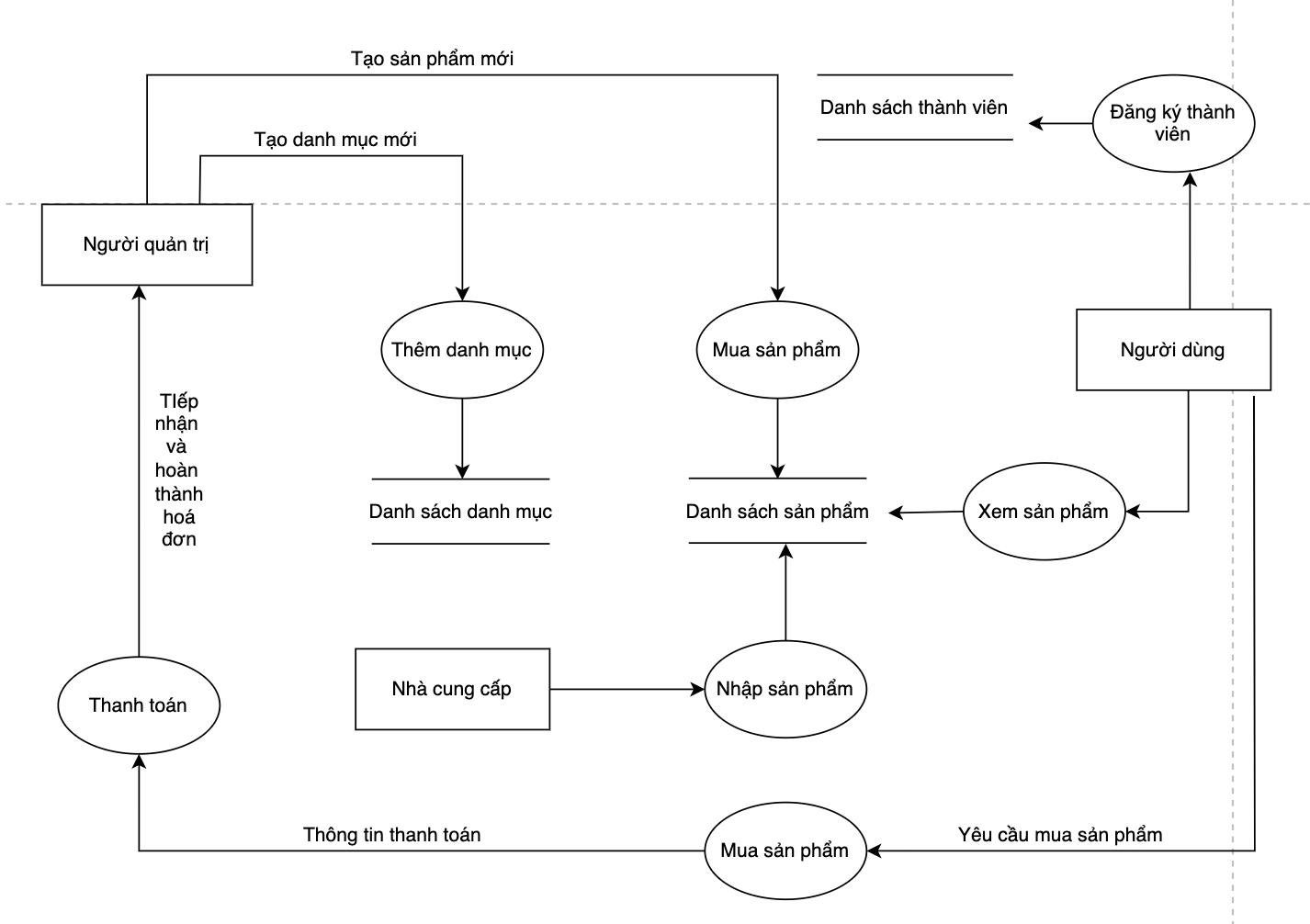
**2.1.2.1. Kí hiệu sử dụng**



**2.1.2.2. Sơ đồ luồng dữ liệu (DFD) mức khung cảnh**

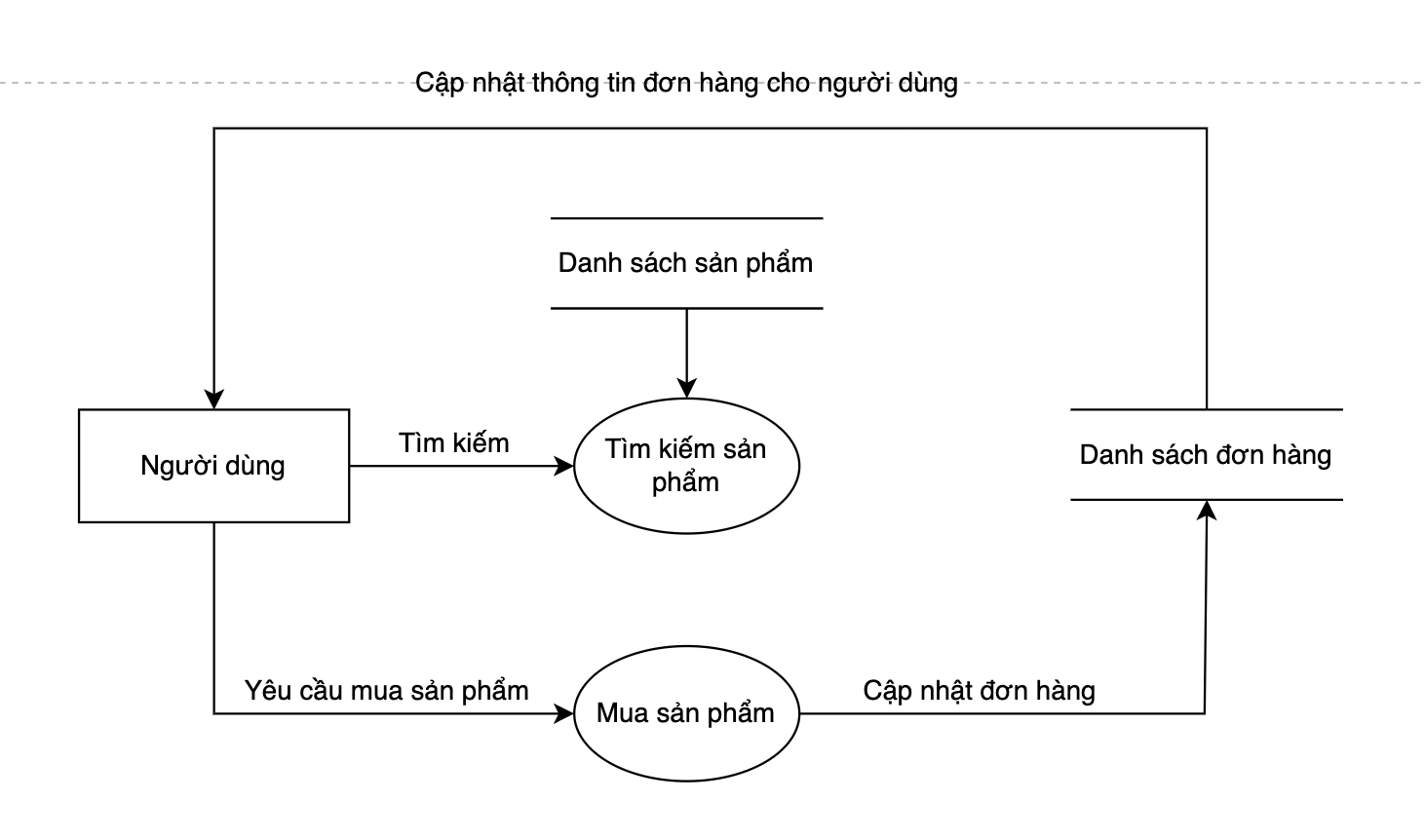
****

**2.1.2.3. DFD mức đỉnh**

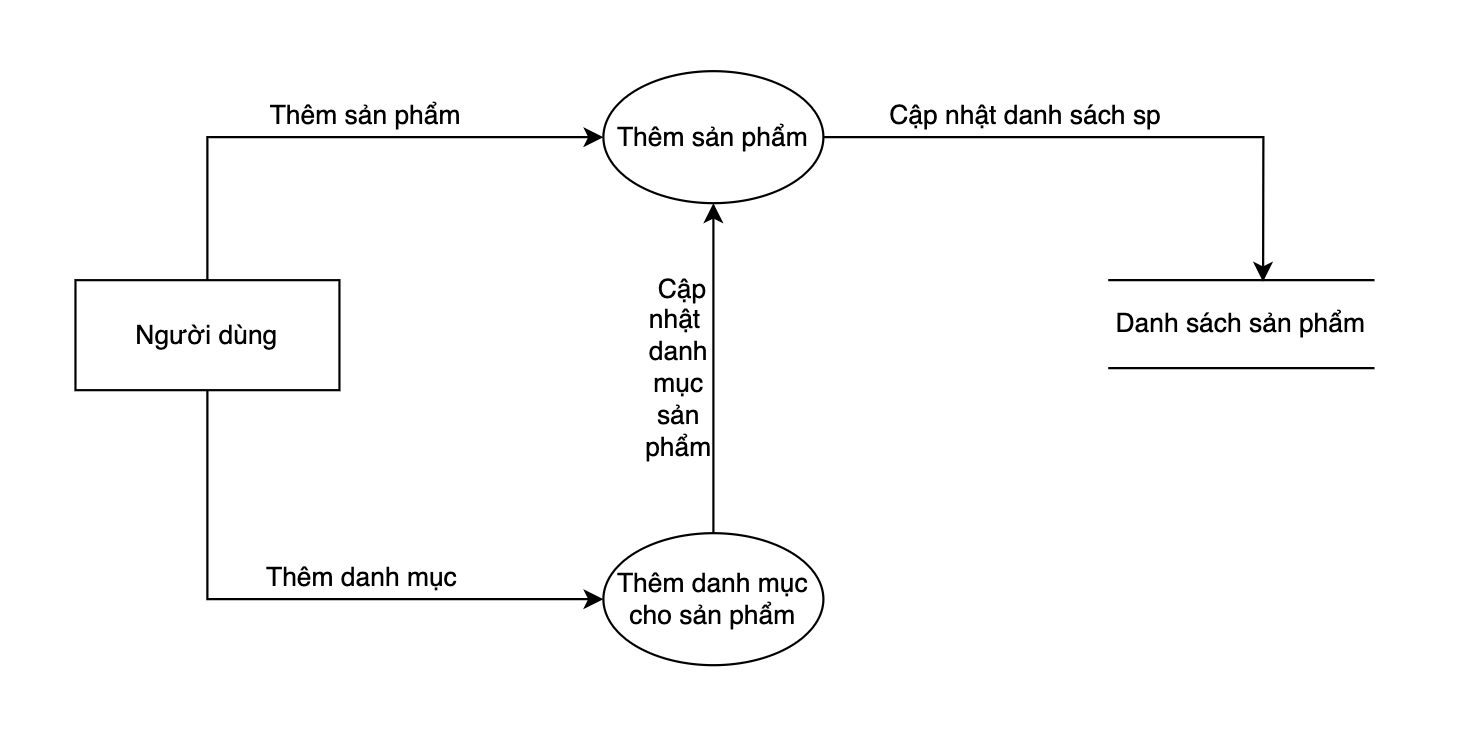


**2.1.3.4. DFD mức dưới đỉnh**

**Mua sản phẩm**

****

**Thêm sản phẩm**

****

* + 1. **Đặc tả tiến trình nghiệp vụ**

**2.1.3.1. Đăng ký tài khoản**

* Người dùng tiến hành đăng ký tài khoản và tiến hành đăng nhập để có thể tiến hành đặt mua sản phẩm.
* Đầu vào:

- Các thông tin các nhân của người dùng

+ email

+ password

+ Họ tên

+ Ngày sinh

+ Số điện thoại

- Các module cần thực hiện

+ Module nhập thông tin cá nhân của người dùng.

+ Module xác nhận thông tin vừa nhập.

+ Module thông báo cho người dùng yêu cầu nhập lại tài khoản người dùng nếu như tài khoản đó đã có người đăng ký

+ Module lưu thông tin khách hang vừa đăng ký vào Database

* Đầu ra:

- Thông báo cho người dùng và yêu cầu nhập lại tài khoản và mật khẩu nếu như email đó đã tồn tại trong Database

- Thông báo kết quả đăng ký

* Nội dung xử lý:

- Nếu: không nhập đúng validate các trường

- Thì: hiển thị thông báo sai validate

- Nếu: nhập đúng validate các trường

- Thì: đăng ký tài khoản thành công và hiển thị thông báo đăng ký thành công

**2.1.3.2. Quản lý và thay đổi thông tin cá nhân**

* Người dùng có thể xem thông tin cá nhân và thay đổi
* Đầu vào:

- Người dùng chọn mục quản lý thông tin cá nhân

- Các module cần thực hiện

+ Module xem thông tin cá nhân

+ Module thay đổi thông tin cá nhân

* Đầu ra:

- Hiển thị thông tin cá nhân của người dùng

- Khách hàng có thể thay đổi thông tin cá nhân

- Thông báo khi người dùng thay đổi thông tin cá nhân thành công

* Nội dung xử lý:

- Nếu: không nhập đúng validate các trường

- Thì: hiển thị thông báo sai validate

- Nếu: nhập đúng validate các trường

- Thì: đổi thông tin cá nhân thành công và hiển thị thông báo thay đổi thông tin thành công

**2.1.3.3. Thay đổi mật khẩu**

* Người dùng có thể thay đổi mật khẩu
* Đầu vào:

- Khách hàng chọn mục thay đổi mật khẩu

- Các module cần thực hiện

+ Module thay đổi mật khẩu

+ Module thông báo lỗi khi nhập mật khẩu không trùng khớp

* Đầu ra:

- Người dùng có thể thay đổi mật khẩu

- Thông báo cho người dùng khi nhập mật khẩu không trùng khớp

- Thông báo cho người dùng khi thay đổi mật khẩu thành công

* Nội dung xử lý:

- Nếu: không nhập đúng validate các trường

- Thì: hiển thị thông báo sai validate

- Nếu: nhập đúng validate các trường

- Thì: thay đổi mật khẩu tài khoản thành công và hiển thị thông báo thay đổi mật khẩu thành công

**2.1.3.4. Xem thông tin sản phẩm**

* Cho phép xem các thông tin liên quan đến sản phẩm
* Đầu vào:

- Khóa học được chọn

- Các quá trình thực hiện

+ Module chọn danh mục có chứa sản phẩm đó.

+ Module chọn sản phẩm

* Đầu ra:

- Các thông tin về sản phẩm đó

* Nội dung xử lý:

- Click vào sản phẩm muốn xem và đến trang chi tiết sản phẩm

**2.1.3.5. Mua sản phẩm**

* Cho phép người dùng tiến hành mua sản phẩm
* Đầu vào:

- Sản phẩm được chọn mua

- Các quá trình thực hiện

+ Module mua sản phẩm

+ Module thông báo khi mua sản phẩm thành công

+ Module thông báo khi không đủ tiền để mua

Đầu ra:

- Các thông tin về sản phẩm đó

* Nội dung xử lý:

- Click vào sản phẩm muốn mua điều hướng sang trang chi tiết sản phẩm.

- Vào chi tiết sản phẩm bấm thanh toán để đặt mua

**2.1.3.6. Thanh toán hoá đơn**

* Cho phép người dùng thanh toán mua sản phẩm
* Đầu vào:

- Sản phẩm muốn mua được chọn

- Các quá trình thực hiện

+ Module hiển thị thông tin địa chỉ người nhận

+ Module hiển thị thông tin xác nhận sản phẩm

+ Module xác nhận thanh toán

- Đầu ra:

- Thông báo thành công và hiển thị chi tiết hoá đơn đó

* Nội dung xử lý:

- Nếu: chưa điền đủ thông tin bắt buộc

- Thì: sẽ không cho phép thanh toán hoá đơn sản phẩm vừa chọn

- Nếu: Đã điền đủ thông tin để đặt mua

- Thì: Cho phép thanh toán sản phẩm đã chọn mua

**2.1.3.7. Theo dõi hoá đơn**

* Cho phép người dùng theo dõi hoá đơn mua hàng
* Đầu vào:

- Hoá đơn của sản phẩm đã mua

- Các quá trình thực hiện

+ Module hiển thị thông tin lịch sử giao dịch

+ Module hiển thị thông tin chi tiết hoá đơn

- Đầu ra:

- Chi tiết hoá đơn và trạng thái hoá đơn

* Nội dung xử lý:
* Click vào lịch sử giao dịch, xem và theo dõi chi tiết hoá đơn.

**2.1.3.6. Xem bình luận và đánh giá sản phẩm**

* Cho phép người dùng xem bình luận và đánh giá sản phẩm
* Đầu vào:

- Sản phẩm muốn xem bình luận và đánh giá

- Các quá trình thực hiện

+ Module hiển thị thông tin sản phẩm

+ Module hiển thị thông tin lượt bình luận

+ Module đánh giá sản phẩm

- Đầu ra:

- Đánh giá sản phẩm đã chọn

* Nội dung xử lý:

- Nếu: Chưa đăng nhập

- Thì: sẽ không cho phép bình luận và đánh giá sản phẩm đang xem.

- Nếu: Đã đăng nhập nhưng chưa mua sản phẩm lần nào.

- Thì: chỉ cho xem bình luận về sản phẩm chứ không được đánh giá cũng như bình luận trực tiếp về sản phẩm.

## **2.2 Phân tích dữ liệu nghiệp vụ**

### **2.2.1.Mô hình dữ liệu ban đầu**

### **2.2.1.1.Xác định kiểu thực thể, kiểu thuộc tính**

Gợi ý 1: Gợi ý từ mẫu biểu

Gợi ý 2: Xác định thực thể dựa vào các nguồn thông tin khác :

+Tài nguyên

**+**Giao dịch

* Các thực thể và thuộc tính đã xác định được là:

**Sản phẩm** (Mã sản phẩm, Tên sản phẩm, Chi tiết sản phẩm, Giá, Trạng thái, Lượt đánh giá, số sao, tồn kho).

**Chi tiết đánh giá sản phẩm** (Mã đánh giá, Số sao, Bình luận)

**Danh mục sản phẩm** (Mã danh mục, Tên danh mục, Mã danh mục cha).

**Ảnh sản phẩm** (Mã ảnh sản phẩm, Đường dẫn ảnh).

**Chương trình giảm giá** (Mã chương trình, Tên chương trình, Phần trăm giảm, Trạng thái kích hoạt).

**Thương hiệu** (Mã thương hiệu, Tên thương hiệu, Ảnh thương hiệu).

**Nhà cung cấp** (Mã nhà cung cấp, Tên nhà cung cấp, Ảnh nhà cung cấp).

**Đơn hàng** (Mã đơn hàng, Tên đơn hàng, Trạng thái đơn hàng, Tổng tiền).

**Chi tiết đơn hàng** (Mã chi tiết đơn hàng, Giá cả, Số lượng).

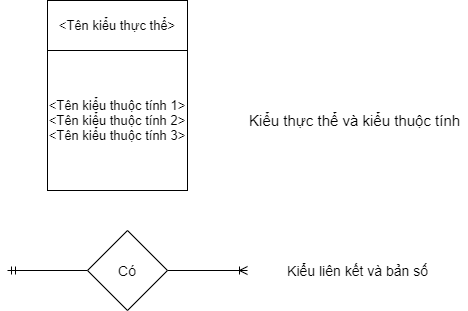
**Người Dùng** (Mã người dùng, Tên người dùng, Email người dùng, SĐT người dùng, Mật khẩu người dùng, Quyền).

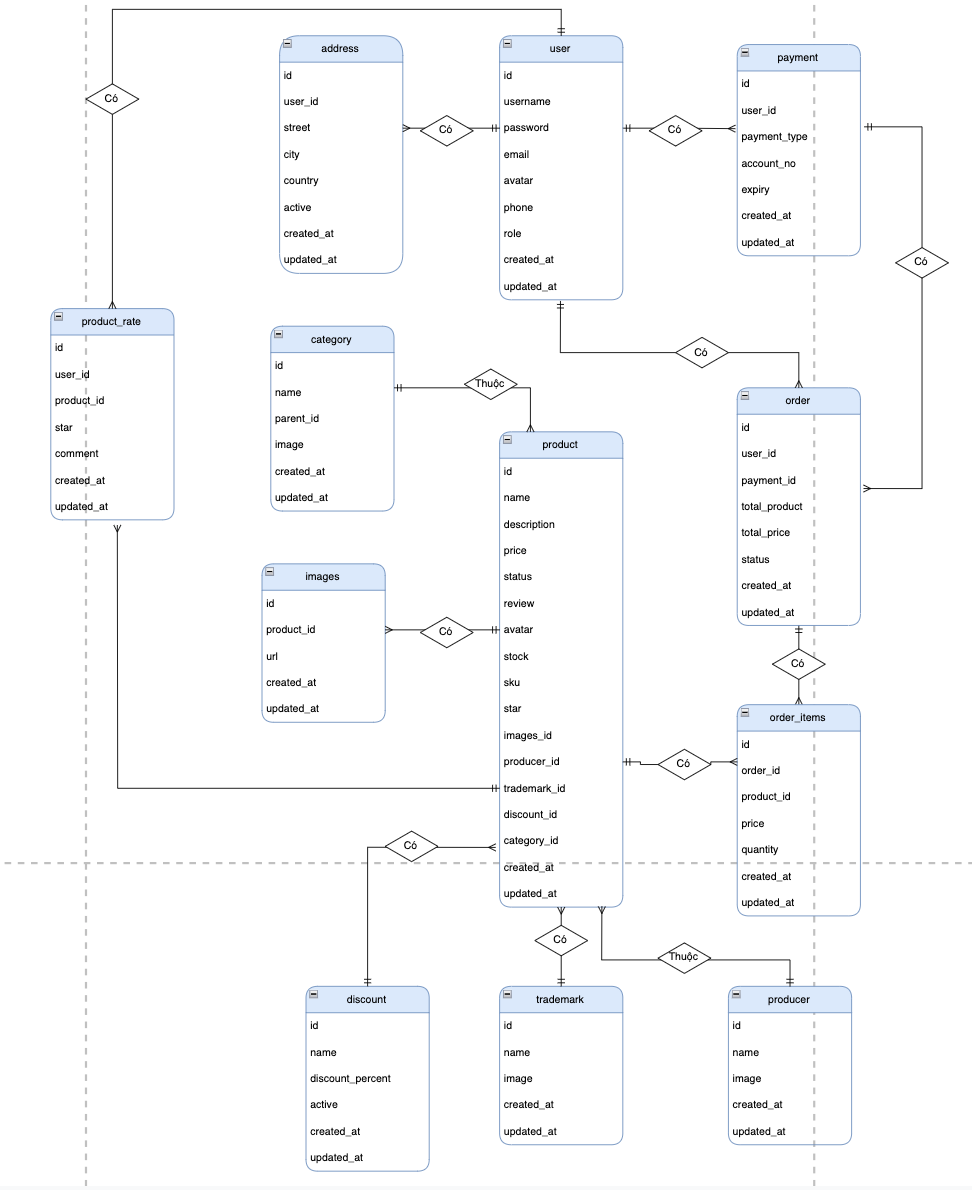
**Địa Chỉ** (Mã địa chỉ, Chi tiết, Thành phố, Quốc gia).

**Thanh toán** (Mã thanh toán, Số tài khoản, Loại thanh toán, Hạn sử dụng).

### **2.2.1.2.Mô hình thực thể liên kết mở rộng**

**Ký hiệu**

****

****

### **2.2.2. Chuẩn hóa dữ liệu**

* Chuyển đổi từ ERD mở rộng về ERD kinh điển
* Khử kiểu thuộc tính đa trị
* Khử kiểu thuộc tính không sơ đẳng
* Xác định khóa chính cho các thực thể
* Chuyển đổi từ ERD kinh điển về ERD hạn chế
* Lược bỏ tên, bản số tối thiểu của kiểu liên kết (áp dụng quy tắc 4)
* Xử lý kiểu liên kết 1-1 (áp dụng quy tắc 5)
* Xử lý kiểu liên kết n-n (áp dụng quy tắc 6):
* Xác định kiểu thuộc tính kết nối, khóa ngoài (áp dụng quy tắc 7)
* Xác định khóa chính cho từng thực thể (áp dụng quy tắc 8)
* Chuyển đổi từ ERD hạn chế về mô hình quan hệ
* Mã hóa kiểu thực thể thành bảng quan hệ/ Mã hóa kiểu thuộc tính thành trường dữ liệu

Thực thể: Khách hàng

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Người dùng | User |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã người dùng | Id |
| Tên người dùng | Username |
| Mật khẩu | Password |
| Email | Email |
| Ảnh đại diện | Avatar |
| Số điện thoại | Phone |
| Quyền người dùng | Role |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Địa chỉ

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Địa chỉ | Address |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã địa chỉ | Id |
| Địa chỉ | Street |
| Thành phố | City |
| Quốc gia | Country |
| Mã người dùng | User\_id |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Phương thức thanh toán

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Phương thức thanh toán | Payment |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã phương thức | Id |
| Loại phương thức | Payment\_type |
| Số thẻ | Account\_no |
| Ngày hết hạn | Expiry |
| Mã người dùng | User\_id |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Sản phẩm

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Sản phẩm | Product |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã sản phẩm | Id |
| Tên sản phẩm | Name |
| Mô tả sản phẩm | Description |
| Giá sản phẩm | Price |
| Trạng thái sp | Status |
| Lượt đánh giá | Review |
| Ảnh bìa sản phẩm | Avatar |
| Số lượng tồn kho | Stock |
| Sku sản phẩm | Sku |
| Sao sản phẩm | Star |
| Mã nhà cung cấp | Producer\_id |
| Mã thương hiệu | Trademark\_id |
| Mã giảm giá | Discount\_id |
| Mã danh mục | Category |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Danh mục

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Danh mục | **ProjectUser** |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã danh mục | Id |
| Tên danh mục | Name |
| Mã danh mục cha | Parent\_id |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Ảnh sản phẩm

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Ảnh sản phẩm | **Images** |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã ảnh | Id |
| Đường dẫn | Url |
| Mã sản phẩm | Product\_id |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Chương trình giảm giá

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Chương trình giảm giá | Discount |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã chương trình giảm giá | Id |
| Tên chương trình | Name |
| Phần trăm giảm | Discount\_percent |
| Trạng thái sử dụng | Active |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Thương hiệu

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Thương hiệu | Trademark |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã thương hiệu | Id |
| Tên thương hiệu | Name |
| Ảnh thương hiệu | Image |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Nhà cung cấp

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Nhà cung cấp | Producer |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã nhà cung cấp | Id |
| Tên nhà cùng cấp | Name |
| Ảnh nhà cung cấp | Image |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Hoá đơn

|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Hoá đơn | Order |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã hoá đơn | Id |
| Mã người dùng | User\_id |
| Mã thanh toán | Payment\_id |
| Tổng sản phẩm | Total\_product |
| Tổng giá | Total\_price |
| Trạng thái đơn hàng | Status |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Chi tiết hoá đơn

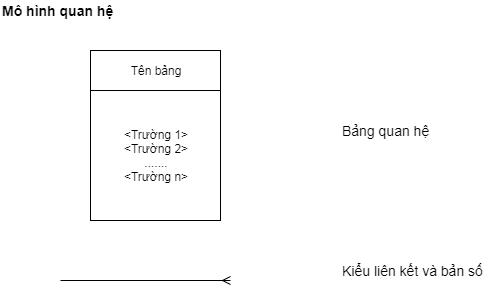
|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Chỉ tiết hoá đơn | Order\_items |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã chi tiết | Id |
| Mã sản phẩm | Product\_id |
| Mã hoá đơn | Order\_id |
| Giá sản phẩm | Price |
| Số lượng | Quantity |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

Thực thể: Chi tiết đánh giá sản phẩm

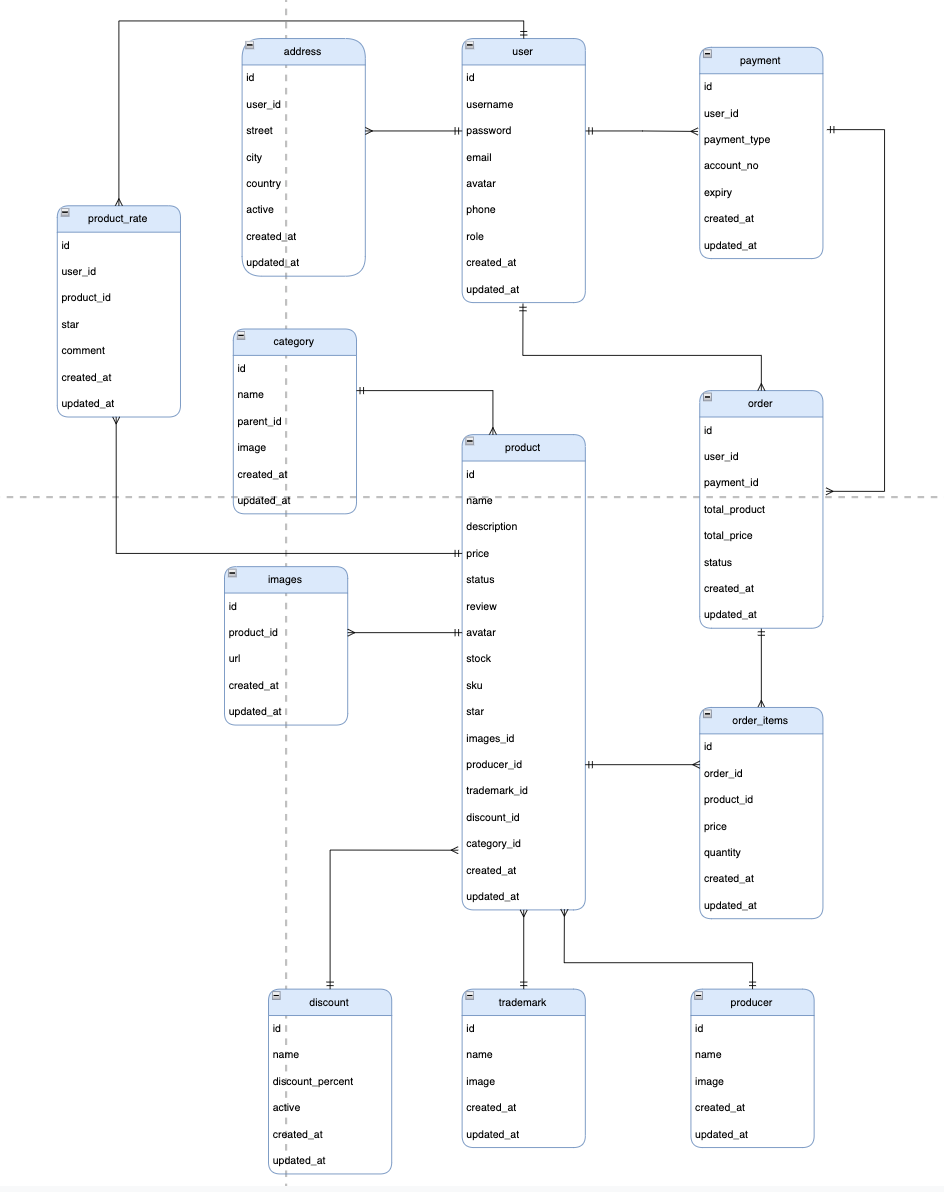
|  |  |
| --- | --- |
| **Kiểu thực thể** | **Bảng quan hệ** |
| Chỉ tiết đánh giá sản phẩm | Product\_rate |
| **Kiểu thuộc tính** | **Trường dữ liệu** |
| Mã chi tiết đánh giá | Id |
| Mã sản phẩm | Product\_id |
| Mã người dùng | User\_id |
| Số sao | Star |
| Bình luận | Comment |
| Thời gian tạo | Created\_at |
| Thời gian cập nhật | Updated\_at |

* Mô hình quan hệ

🡪Kí hiệu

****

🡪Mô hình



### 

### **2.2.3. Đặc tả dữ liệu**

* **User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã người dùng |
| 2 |  |  | Username | C(30) | Tên người dùng |
| 3 |  |  | Password | C(50) | Mật khẩu |
| 4 |  |  | Email | C(50) | Email người dùng |
| 5 |  |  | Avatar | C(50) | Ảnh đại diện |
| 6 |  |  | Phone | C(30) | SĐT người dùng |
| 7 |  |  | Role | C(20) | Quyền ngừoi dùng |
| 8 |  |  | Created\_at | C(30) | Thời gian tạo |
| 9 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Address**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã địa chỉ |
| 2 |  |  | Street | C(30) | Địa chỉ |
| 3 |  |  | City | C(30) | Thành phố |
| 4 |  |  | Country | C(30) | Quốc gia |
| 5 |  |  | Status | C(20) | Trạng thái địa chỉ |
| 6 |  | X | User\_id | N | Mã người dùng |
| 7 |  |  | Created\_at | C(30) | Thời gian tạo |
| 8 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Payment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã phương thức |
| 2 |  |  | Payment\_type | C(20) | Loại thanh toán |
| 3 |  |  | Account\_no | C(20) | Số tài khoản |
| 4 |  |  | Expiry | C(20) | Hạn sử dụng |
| 5 |  | X | User\_id | N | Mã người dùng |
| 6 |  |  | Created\_at | C(30) | Thời gian tạo |
| 7 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Category**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | C(10) | Mã danh mục |
| 2 |  |  | Name | C(30) | Tên danh mục |
| 3 |  |  | Parent\_id | N | Mã danh mục cha |
| 4 |  |  | Image | C(100) | Ảnh danh mục |
| 5 |  |  | Created\_at | C(30) | Thời gian tạo |
| 6 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Images**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã ảnh sản phẩm |
| 2 |  |  | Url | C(30) | Đường dẫn ảnh |
| 3 |  | X | Product\_id | N | Mã sản phẩm |
| 5 |  |  | Created\_at | C(30) | Thời gian tạo |
| 6 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Product**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã sản phẩm |
| 2 |  |  | Name | C(30) | Tên sản phẩm |
| 3 |  |  | Description | C(100) | Mô tả sản phẩm |
| 4 |  |  | Price | N(30) | Giá sản phẩm |
| 5 |  |  | Status | C(30) | Trạng thái sản phẩm |
| 6 |  |  | Stock | C(30) | Số lượng tồn kho |
| 7 |  |  | Star | C(30) | Số sao sản phẩm |
| 8 |  |  | Sku | C(30) | Sku sản phẩm |
| 9 |  |  | Created\_at | C(30) | Thời gian tạo |
| 10 |  |  | Updated\_at | C(30) | Thời gian cập nhật |
| 11 |  | X | Producer\_id | N | Mã nhà cung cấp |
| 12 |  | X | Trademark\_id | N | Mã thương hiệu |
| 13 |  | X | Discount\_id | N | Mã giảm giá |
| 14 |  | X | Category\_id | N | Mã danh mục |

* **Discount**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã giảm giá |
| 2 |  |  | Name | C(50) | Tên giảm giá |
| 3 |  |  | Discount\_percent | C(50) | Phần trăm giảm giá |
| 4 |  |  | Active | N(1) | Trạng thái kích hoạt |
| 5 |  |  | Created\_at | C(30) | Thời gian tạo |
| 6 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Trademark**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N(10) | Mã thương hiệu |
| 2 |  |  | Name | C(30) | Tên thương hiệu |
| 3 |  |  | Image | C(250) | Ảnh thương hiệu |
| 4 |  |  | Created\_at | C(30) | Thời gian tạo |
| 5 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Producer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã nhà cung cấp |
| 2 |  |  | Name | C(30) | Tên nhà cung cấp |
| 3 |  |  | Image | C(250) | Ảnh nhà cung cấp |
| 4 |  |  | Created\_at | C(30) | Thời gian tạo |
| 5 |  |  | Updated\_at | C(30) | Thời gian cập nhật |

* **Order**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | RequestId | N | Mã hoá đơn |
| 2 |  |  | Status | N | Trạng thái đơn hàng |
| 3 |  |  | Total\_price | N(50) | Tổng giá sản phẩm |
| 4 |  |  | Total\_product | N(50) | Tổng sản phẩm |
| 5 |  |  | Created\_at | C(30) | Thời gian tạo |
| 6 |  |  | Updated\_at | C(30) | Thời gian cập nhật |
| 7 |  | X | User\_id | N | Mã người dùng |
| 8 |  | X | Payment\_id | N | Mã thanh toán |

* **Order Items**

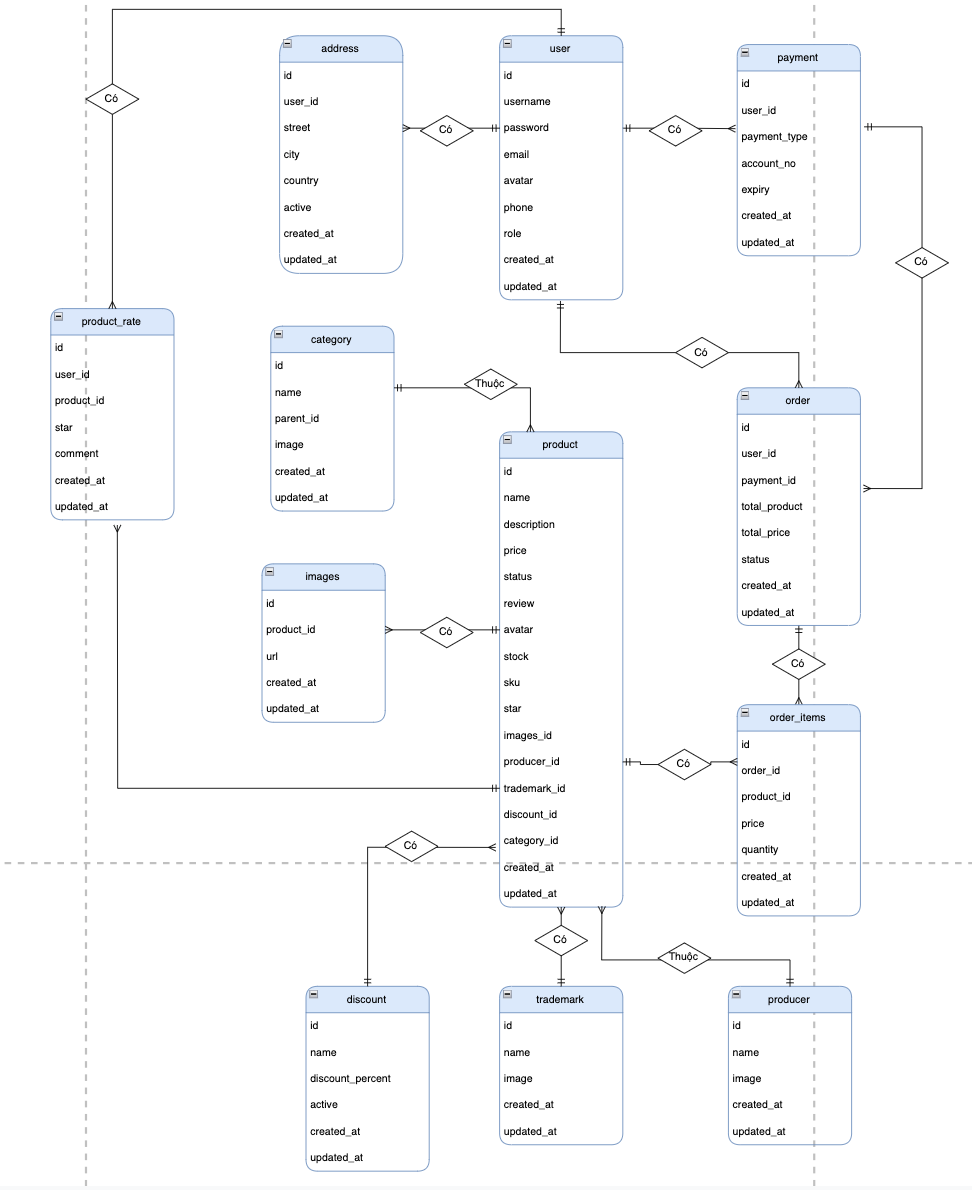
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã chi tiết hoá đơn |
| 2 |  |  | Price | N(10) | Giá sản phẩm |
|  |  |  | Quantity | N(10) | Số lượng sản phẩm |
| 5 |  |  | Created\_at | C(30) | Thời gian tạo |
| 6 |  |  | Updated\_at | C(30) | Thời gian cập nhật |
| 4 |  | X | Product\_id | C(10) | Mã sản phẩm |
|  |  | X | Order\_id | C(10) | Mã hoá đơn |

* **Product\_Rate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Khoá chính | Khoá ngoại | Tên trường | Kiểu dữ liệu | Diễn giải |
| 1 | X |  | Id | N | Mã đánh giá sản phẩm |
| 2 |  |  | Comment | N(10) | Bình luận |
|  |  |  | Star | N(10) | Số sao |
| 5 |  |  | Created\_at | C(30) | Thời gian tạo |
| 6 |  |  | Updated\_at | C(30) | Thời gian cập nhật |
| 4 |  | X | Product\_id | C(10) | Mã sản phẩm |
|  |  | X | User\_id | C(10) | Mã người dùng |

**Chương 3  
THIẾT KẾ HỆ THỐNG**

## **3.1. Thiết kế cơ sở dữ liệu**

****

**3.2. Thiết kế kiểm soát**

**3.2.1. Thiết kế nhóm người dùng**

* Dựa vào cơ cấu tổ chức của hệ thống gồm có 2 nhóm người dùng là:
  + Nhóm 1: nhóm người dùng ứng dụng web app
  + Nhóm 2: nhóm người quản trị
* Nhóm quản trị gồm: quản lý khóa học, quản lý danh mục khóa học, quản lý người dùng, thống kê
* Mỗi người dùng có quyền thay đổi mật khẩu và thông tin cá nhân của mình, ảnh đại diện, xem các thống kê cá nhân và các chứng chỉ đạt được, trừ “tên đăng nhập”

**3.2.2. Phân hạn nhóm người dùng về tiến trình**

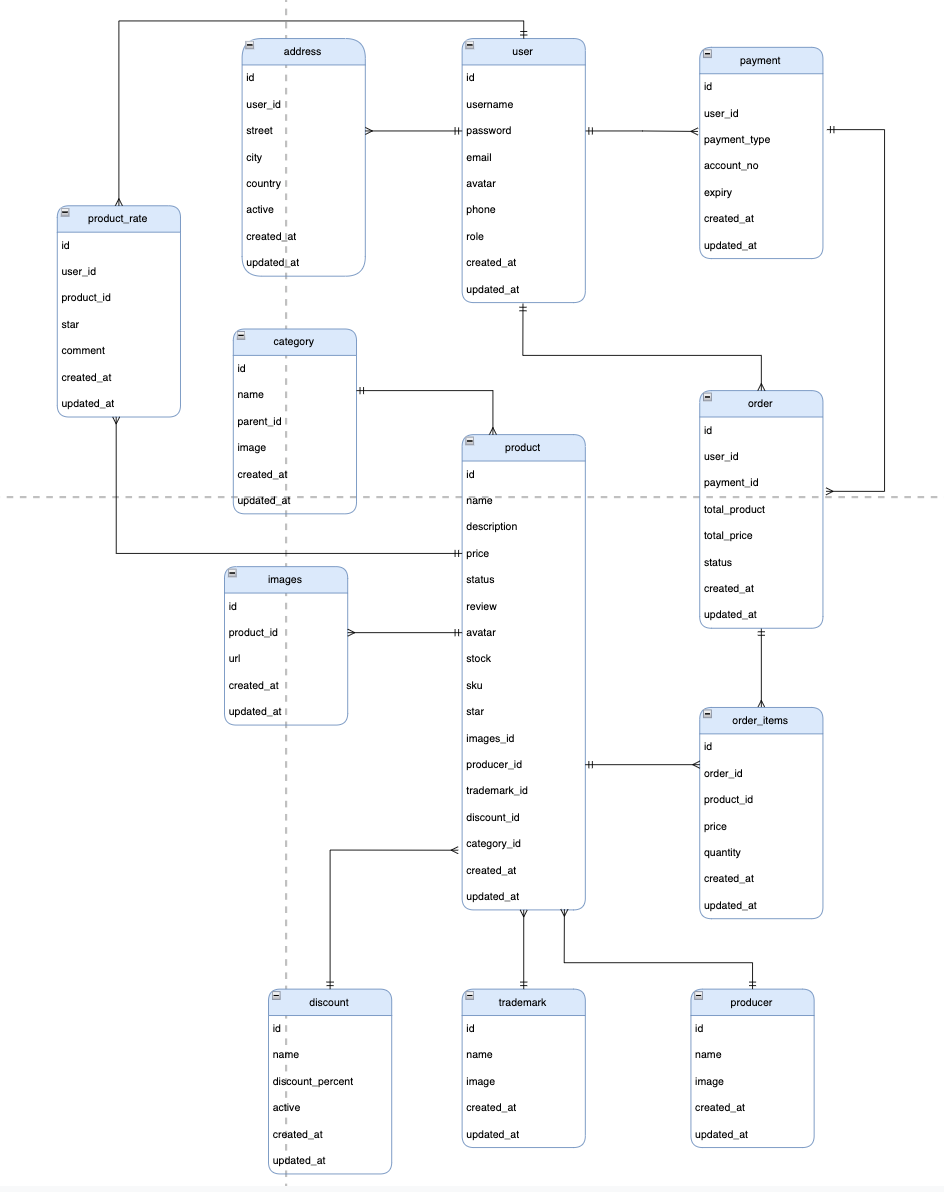
Ký hiệu: A: Active; not A: Not Active

|  |  |  |  |
| --- | --- | --- | --- |
| Tiến trình |  | | |
| Quản trị | Người dùng |
| Đăng ký | Not A | A |
| Đăng nhập | A | A |
| Tìm kiếm sản phẩm | Not A | A |
| Mua Sản phẩm | Not A | A |
| Thay Đổi mật khẩu | Not A | A |
| Thay Đổi thông tin cá nhân | Not A | A |
| Xem thông tin đơn hàng | A | A |
| Thêm sản phẩm mới | A | Not A |
| Thêm danh mục sản phẩm mới | A | Not A |
| Thêm nhà cung cấp mới | A | Not A |
| Xem địa chỉ nhận hàng | A | A |
| Quản lý thương hiệu sản phẩm | A | Not A |
| Quản lý đơn hàng | A | A |
| Chỉnh sửa quyền tài khoản | A | Not A |
| Hiển thị thống kê các thông tin về website | A | Not A |

Bảng 1. Phân hạn nhóm người dùng về tiến trình

**3.3. Thiết kế cơ sở dữ liệu**

**3.3.1. Mô hình dữ liệu hệ thống**



### **3.3.2. Đặc tả bảng dữ liệu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 1 | | 2.Tên bảng: User | | 3.Bí danh: User | |
| 4. Mô tả: Lưu trữ thông tin về người dùng | | | | | |
| 5. Mô tả chi tiết các cột | | | | | |
| Số | Tên cột | Mô tả | Kiểu dữ liệu | Khuôn dạng | N |
| 1 | Id | Mã người dùng | N(10) | Số nguyên | x |
| 2 | Username | Tên người dùng | C(30) | Chữ cái | x |
| 3 | Password | Mật khẩu người dùng | C(30) | Chữ cái |  |
| 4 | Email | Email người dùng | C(30) | Chữ cái |  |
| 5 | Avatar | Ảnh đại diện người dùng | C(100) | Chữ cái |  |
| 6 | Phone | Số điện thoại người dùng | C(100) | Chữ cái |  |
| 7 | Role | Quyền người dùng | C(8) | Chữ cái |  |
| 8 | Created\_at | Thời gian tạo | D(20) | Dd/mm/yyyy |  |
| 9 | Updated\_at | Thời gian cập nhật | D(20) | Dd/mm/yyyy |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 2 | | 2.Tên bảng: Address | | | 3.Bí danh: Address | | |
| 4. Mô tả: Lưu trữ thông tin về địa chỉ người dùng | | | | | | | |
| 5. Mô tả chi tiết các cột | | | | | | | |
| Số | Tên cột | Mô tả | | Kiểu dữ liệu | Khuôn dạng | | N |
| 1 | Id | Mã địa chỉ | | N(10) | Số nguyên | | x |
| 2 | Street | Địa chỉ | | C(100) | Chữ cái | | x |
| 3 | City | Thành phố | | C(100) | Chữ cái | |  |
| 4 | Country | Đất nước | | C(100) | Chữ cái | |  |
| 5 | Active | Trạng thái sử dụng | | N(1) | 0 hoặc 1 | |  |
| 6 | Created\_at | Thời gian tạo | | D(20) | Dd/mm/yyyy | |  |
| 7 | Updated\_at | Thời gian cập nhật | | D(20) | Dd/mm/yyyy | |  |
| 5. Khóa ngoài | | | | | | | |
| Số | Tên | | Cột khóa ngoài | | | Quan hệ với bảng | |
| 1 | User\_id | | User\_id | | | User | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 3 | | 2.Tên bảng: Payment | | | | | 3.Bí danh: Payment | |
| 4. Mô tả: Lưu trữ thông tin về phương thức thanh toán | | | | | | | | |
| 5. Mô tả chi tiết các cột | | | | | | | | |
| Số | Tên cột | Mô tả | | Kiểu dữ liệu | Khuôn dạng | | | N |
| 1 | Id | Mã phương thức | | N(10) | Số nguyên | | | x |
| 2 | Payment\_type | Loại thanh toán | | C(100) | Chữ cái | | | x |
| 3 | Account\_no | Số tài khoản thẻ | | C(100) | Chữ cái | | |  |
| 4 | Exipry | Ngày hết hạn | | C(10) | Chữ cái | | |  |
| 6 | Created\_at | Thời gian tạo | | D(20) | Dd/mm/yyyy | | |  |
| 7 | Updated\_at | Thời gian cập nhật | | D(20) | Dd/mm/yyyy | | |  |
| 5. Khóa ngoài | | | | | | | | |
| Số | Tên | | Cột khóa ngoài | | | Quan hệ với bảng | | |
| 1 | User\_id | | User\_id | | | User | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 4 | | 2.Tên bảng: Product | | | | 3.Bí danh: Product | |
| 4. Mô tả: Lưu trữ thông tin về sản phẩm | | | | | | | |
| 5. Mô tả chi tiết các cột | | | | | | | |
| Số | Tên cột | Mô tả | | Kiểu dữ liệu | | Khuôn dạng | N |
| 1 | Id | Mã sản phẩm | | N(10) | | Số nguyên | x |
| 2 | Name | Tên sản phẩm | | C(100) | | Chữ cái | x |
| 3 | Description | Chi tiết sản phẩm | | C(250) | | Chữ cái |  |
| 4 | Price | Giá sản phẩm | | N(100) | | Số nguyên |  |
| 5 | Status | Trạng thái sản phẩm | | C(200) | | Chữ cái |  |
| 6 | Review | Số lượng đánh giá | | N(100) | | Số nguyên |  |
| 7 | Avatar | Ảnh bìa sản phẩm | | C(200) | | Chữ cái |  |
| 8 | Stock | Số lượng tồn kho | | N(100) | | Số nguyên |  |
| 9 | Sku | Sku sản phẩm | | C(100) | | Chữ cái |  |
| 10 | Star | Số sao sản phẩm | | N(20) | | Số nguyên |  |
| 11 | Created\_at | Thời gian tạo | | D(20) | | Dd/mm/yyyy |  |
| 12 | Updated\_at | Thời gian cập nhật | | D(20) | | Dd/mm/yyyy |  |
| 5. Khóa ngoài | | | | | | | |
| Số | Tên | | Cột khóa ngoài | | Quan hệ với bảng | | |
| 1 | Images\_id | | Images\_id | | Images | | |
| 2 | Producer\_id | | Producer\_id | | Producer | | |
| 3 | Trademark\_id | | Trademark\_id | | Trademark | | |
| 4 | Discount\_id | | Discount\_id | | Discount | | |
| 5 | Category\_id | | Category\_id | | Category | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 5 | | 2.Tên bảng: Category | | 3.Bí danh: Category | |
| 4. Mô tả: Lưu trữ thông tin về danh mục sản phẩm | | | | | |
| 5. Mô tả chi tiết các cột | | | | | |
| Số | Tên cột | Mô tả | Kiểu dữ liệu | Khuôn dạng | N |
| 1 | Id | Mã danh mục | N(10) | Số nguyên | x |
| 2 | Name | Tên danh mục | C(100) | Chữ cái |  |
| 3 | Parent\_id | Mã danh mục cha | N(10) | Số nguyên |  |
| 4 | Image | Ảnh danh mục | C(100) | Chữ cái |  |
| 5 | Created\_at | Thời gian tạo | D(20) | Dd/mm/yyyy |  |
| 6 | Updated\_at | Thời gian cập nhật | D(20) | Dd/mm/yyyy |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 6 | | 2.Tên bảng: Discount | | 3.Bí danh: Discount | |
| 4. Mô tả: Lưu trữ thông tin về chương trình khuyễn mãi | | | | | |
| 5. Mô tả chi tiết các cột | | | | | |
| Số | Tên cột | Mô tả | Kiểu dữ liệu | Khuôn dạng | N |
| 1 | Id | Mã chương trình khuyến mãi | N(10) | Số nguyên | x |
| 2 | Name | Tên chương trình khuyến mãi | C(100) | Chữ cái |  |
| 3 | Discount\_percent | Phần trăm giảm giá | N(10) | Số nguyên |  |
| 4 | Active | Trạng thái sử dụng | N(1) | 0 hoặc 1 |  |
| 5 | Created\_at | Thời gian tạo | D(20) | Dd/mm/yyyy |  |
| 6 | Updated\_at | Thời gian cập nhật | D(20) | Dd/mm/yyyy |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 7 | | 2.Tên bảng: Images | | | | 3.Bí danh: Images | |
| 4. Mô tả: Lưu trữ thông tin về ảnh sản phẩm | | | | | | | |
| 5. Mô tả chi tiết các cột | | | | | | | |
| Số | Tên cột | Mô tả | | Kiểu dữ liệu | | Khuôn dạng | N |
| 1 | Id | Mã ảnh sản phẩm | | N(10) | | Số nguyên | x |
| 2 | Url | Ảnh sản phẩm | | C(100) | | Chữ cái |  |
| 3 | Created\_at | Thời gian tạo | | D(20) | | Dd/mm/yyyy |  |
| 4 | Updated\_at | Thời gian cập nhật | | D(20) | | Dd/mm/yyyy |  |
| 5. Khóa ngoài | | | | | | | |
| Số | Tên | | Cột khóa ngoài | | Quan hệ với bảng | | |
| 1 | Product\_id | | Product\_id | | Product | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 8 | | 2.Tên bảng: Trademark | | 3.Bí danh: Trademark | |
| 4. Mô tả: Lưu trữ thông tin về thương hiệu sản phẩm | | | | | |
| 5. Mô tả chi tiết các cột | | | | | |
| Số | Tên cột | Mô tả | Kiểu dữ liệu | Khuôn dạng | N |
| 1 | Id | Mã thương hiệu | N(10) | Số nguyên | x |
| 2 | Name | Tên thương hiệu | C(100) | Chữ cái |  |
| 3 | Image | Ảnh thương hiệu | C(100) | Ảnh thuơng hiệu |  |
| 4 | Created\_at | Thời gian tạo | D(20) | Dd/mm/yyyy |  |
| 5 | Updated\_at | Thời gian cập nhật | D(20) | Dd/mm/yyyy |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 9 | | 2.Tên bảng: Producer | | 3.Bí danh: Producer | |
| 4. Mô tả: Lưu trữ thông tin về nhà cung cấp | | | | | |
| 5. Mô tả chi tiết các cột | | | | | |
| Số | Tên cột | Mô tả | Kiểu dữ liệu | Khuôn dạng | N |
| 1 | Id | Mã nhà cung cấp | N(10) | Số nguyên | x |
| 2 | Name | Tên nhà cung cấp | C(100) | Chữ cái |  |
| 3 | Image | Ảnh nhà cung cấp | C(100) | Chữ cái |  |
| 4 | Created\_at | Thời gian tạo | D(20) | Dd/mm/yyyy |  |
| 5 | Updated\_at | Thời gian cập nhật | D(20) | Dd/mm/yyyy |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 10 | | 2.Tên bảng: Order | | | | 3.Bí danh: Order | |
| 4. Mô tả: Lưu trữ thông tin về hoá đơn sản phẩm | | | | | | | |
| 5. Mô tả chi tiết các cột | | | | | | | |
| Số | Tên cột | Mô tả | | Kiểu dữ liệu | | Khuôn dạng | N |
| 1 | Id | Mã hoá đơn | | N(10) | | Số nguyên | x |
| 2 | Total\_product | Tổng sản phẩm | | N(10) | | Số nguyên |  |
| 3 | Total\_price | Tổng giá sản phẩm | | N(10) | | Số nguyên |  |
| 4 | Status | Trạng thái hoá đơn | | C(100) | | Chữ cái |  |
| 5 | Created\_at | Thời gian tạo | | D(20) | | Dd/mm/yyyy |  |
| 6 | Updated\_at | Thời gian cập nhật | | D(20) | | Dd/mm/yyyy |  |
| 5. Khóa ngoài | | | | | | | |
| Số | Tên | | Cột khóa ngoài | | Quan hệ với bảng | | |
| 1 | User\_id | | User\_id | | User | | |
| 2 | Payment\_id | | Payment\_id | | Payment | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 11 | | 2.Tên bảng: Order\_items | | | | 3.Bí danh: Order\_items | |
| 4. Mô tả: Lưu trữ thông tin chi tiết về hoá đơn | | | | | | | |
| 5. Mô tả chi tiết các cột | | | | | | | |
| Số | Tên cột | Mô tả | | Kiểu dữ liệu | | Khuôn dạng | N |
| 1 | Id | Mã chi tiết hoá đơn | | N(10) | | Số nguyên | x |
| 2 | Price | Giá sản phẩm | | N(10) | | Số nguyên |  |
| 3 | Quantity | Số lượng sản phẩm | | N(10) | | Số nguyên |  |
| 4 | Created\_at | Thời gian tạo | | D(20) | | Dd/mm/yyyy |  |
| 5 | Updated\_at | Thời gian cập nhật | | D(20) | | Dd/mm/yyyy |  |
| 5. Khóa ngoài | | | | | | | |
| Số | Tên | | Cột khóa ngoài | | Quan hệ với bảng | | |
| 1 | Product\_id | | Product\_id | | Product | | |
| 2 | Order\_id | | Order\_id | | Order | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Số hiệu: 12 | | 2.Tên bảng: Product\_rate | | | | 3.Bí danh: Product\_rate | |
| 4. Mô tả: Lưu trữ thông tin chi tiết đánh giá sản phẩm | | | | | | | |
| 5. Mô tả chi tiết các cột | | | | | | | |
| Số | Tên cột | Mô tả | | Kiểu dữ liệu | | Khuôn dạng | N |
| 1 | Id | Mã chi tiết đánh giá sản phẩm | | N(10) | | Số nguyên | x |
| 2 | Star | Số sao đánh gía | | N(10) | | Số nguyên |  |
| 3 | Comment | Comment sản phẩm | | C(250) | | Chữ cái |  |
| 4 | Created\_at | Thời gian tạo | | D(20) | | Dd/mm/yyyy |  |
| 5 | Updated\_at | Thời gian cập nhật | | D(20) | | Dd/mm/yyyy |  |
| 5. Khóa ngoài | | | | | | | |
| Số | Tên | | Cột khóa ngoài | | Quan hệ với bảng | | |
| 1 | Product\_id | | Product\_id | | Product | | |
| 2 | User\_id | | User\_id | | User | | |

# 

# **Chương 4 XÂY DỰNG CHƯƠNG TRÌNH**

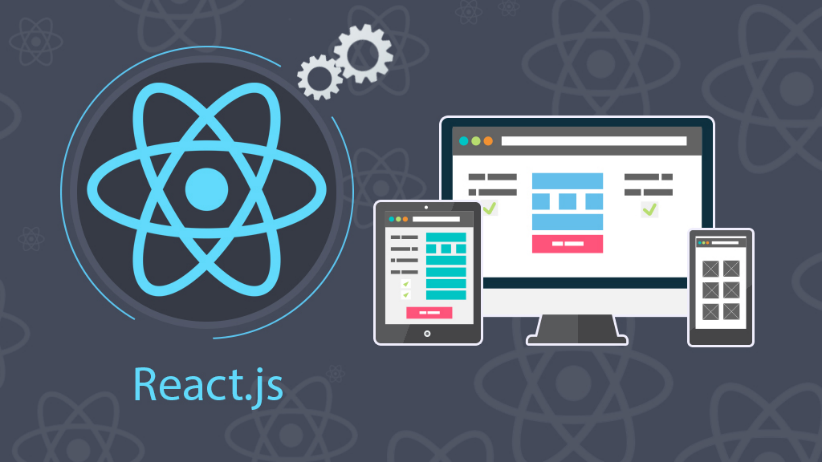
## **4.1. Mô tả hệ thống**

### **4.1.1. Ngôn ngữ sử dụng**

* + - 1. **Ngôn ngữ sử dụng chính: javascript**
* Giới thiệu:
  + Javascipt là gì: Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... thậm chí các trình duyệt trên thiết bị di động cũng có hỗ trợ.

Nếu bạn đã biết qua về HTML5 thì bạn thấy có các khái niệm như sessionStore hay localStore, đây là hai đối tượng được tạo nên từ Javascript nên rõ ràng trong HTML5 cũng có sử dụng nó. Với những ứng dụng đó thì bạn thấy javascript không thể chết trong các ứng dụng website được.

* + Một số thư viện, framework nổi tiếng hiện nay:
    - Angular
    - Nodejs: NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng. NodeJs được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau.
    - Reactjs: ReactJS là một thư viện javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Đây là 1 library cho phép nhúng code HTML trong code javascript nhờ vào JSX. Tích hợp giữa Javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.
    - Vuejs
    - Jquery
    - Nestjs
      1. **Ngôn ngữ sử dụng phía frontend ( ReactJs)**
* Trang chủ**:** https://reactjs.org/
* Giới thiệu**:**
* **Tổng quan về ReactJS**



React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javasscript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.

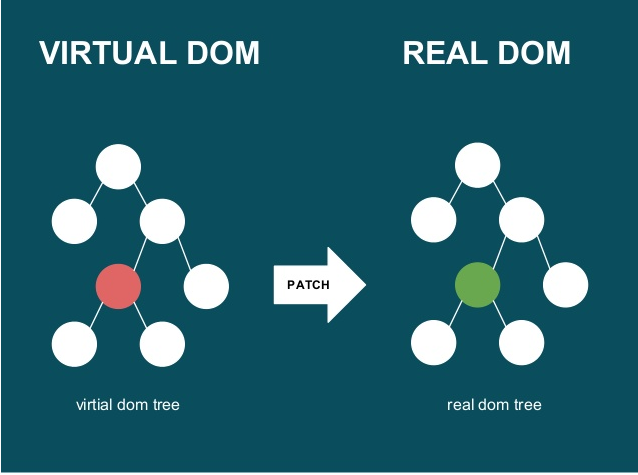
* **Giới thiệu về React**
  + **Component**

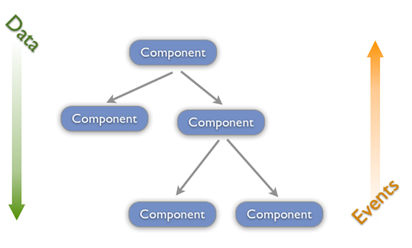
React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo. Cấu trúc của Component:



* + **Virtual DOM**

Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.



* + One-way binding

React sử d

Sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi. Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.

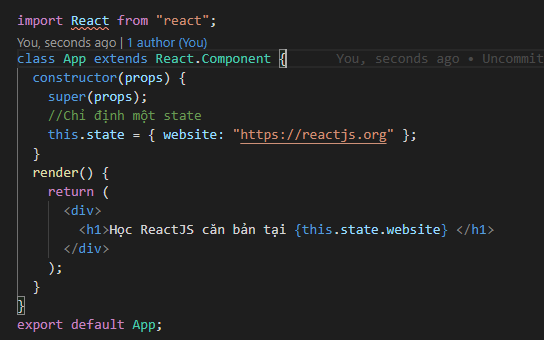
* + **JSX (JavaScript XML)**

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng (tham khảo tại [https://jsx.github.io/](https://jsx.github.io/" \t "_blank))

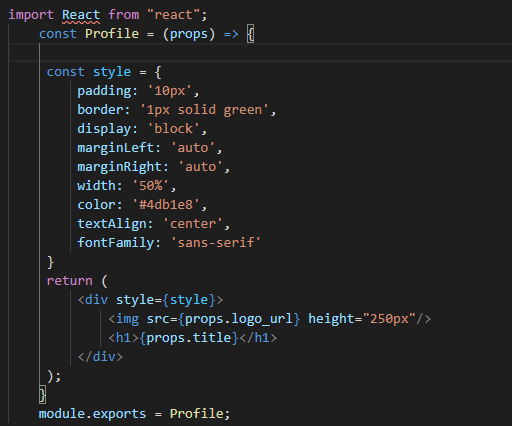
* + **State**

**State** là một object có thể được sử dụng để chứa dữ liệu hoặc thông tin về components. State có thể được thay đổi bất cứ khi nào mong muốn. Khác với props bạn có thể truyền props sang các components khác nhau thì state chỉ tồn tại trong phạm vi của components chứa nó, mỗi khi state thay đổi thì components đó sẽ được render lại.

Trong các dự án React, state được dùng để phản hồi các yêu cầu từ người dùng, hay lưu trữ một dữ liệu nào đó trong components.



* + **Props**
* Chúng ta sử dụng props để gửi dữ liệu đến component.
* Mọi component được coi là một hàm javascript thuần khiết (Pure Function).
* Trong ReactJS, props tương đương với các tham số của hàm javascript thuần khiết.
* Props là bất biến (không thể thay đổi được). Bởi vì điều này được phát triển trong khái niệm về các hàm thuần khiết. Trong các hàm thuần khiết, chúng ta không thể thay đổi dữ liệu của các tham số. Vì vậy, cũng không thể thay đổi dữ liệu của prop trong ReactJS.
* Một lớp cơ bản trong ReactJS



* Import và sử dụng một trong các component ReactJS ở trên như bên dưới



* + **Router**

Routing là cơ chế trong single-page giúp ta chuyển đổi qua lại giữa các component mà không cần phải reload lại browser. Để sử dụng được routing trong ReactJS, ta cần package hỗ trợ đó là React-router-dom.

**Cài đặt Route cho ứng dụng React**

Để cài đặt toàn bộ Router cho một ứng dụng react thì chúng ta cần cài bộ thư viện có tên *react-router-dom*.

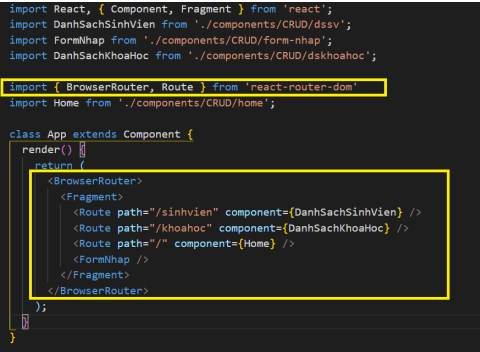
Các thành phần trong React-router-dom:

* + **BrowserRouter**

BrowserRouter là component sẽ bao toàn bộ ứng dụng để có thể sử dụng được routing. Nó sử dụng History API có trong HTML5 để theo dõi lịch sử bộ định tuyến của bạn. Nếu **KHÔNG** bọc <BrowserRouter> ngoài file gốc (thường là App.js) của project thì sẽ không thể sử dụng routing được

* + **Route**

Route được định nghĩa theo một ánh xạ (mapping) giữa một URL và một component. Thông thường, <Route > sẽ chứa 2 props quan trọng đó là *path* và *component.*



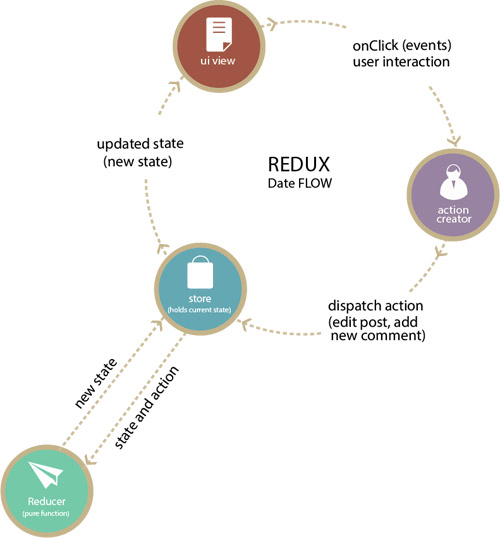
* **Tổng quan về Redux**

Redux là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng. Được dựa trên nền tảng tư tưởng của kiến trúc Flux do Facebook giới thiệu, do vậy Redux thường là bộ đôi kết hợp hoàn hảo với React (React Js và React Native).

Nguyên Lý hoạt động

* Nguyên lý đầu tiên của Redux là tất cả mọi thứ mà thay đổi trong ứng dụng, bao gồm dữ liệu và trạng thái giao diên – UI state, được lưu trong đối tượng gọi là state hoặc state tree.Ttrong quá trình hoạt động của ứng dụng, bị phụ thuộc khá nhiều yếu tố: dữ liệu từ máy chủ ban đầu, thao tác của người dùng (nhập dữ liệu, click menu, button...), dữ liệu cập nhật từ máy chủ, dữ liệu được tính toán trong ứng dụng (Ví dụ: tính toán số dư tài khoản dựa trên biến động của tỉ giá)...những yếu tố này còn gọi là nguồn dữ liệu. Những nguồn dữ liệu này đến từ những nơi khác nhau, bất kỳ khi nào, khiến cho ứng dụng của chúng ta rất khó kiểm soát, chúng tác động đến những thành phần đơn lẻ, hoặc nhiều thành phần trên ứng dụng, hoặc hiệu ứng dây chuyền. Chính vì sự phức tạp đó, là vấn đề mà Redux muốn giải quyết, tất cả các nguồn dữ liệu cần phải được quản lý và tạo thành một nguồn duy nhất, tin cậy.
* Nguyên lý thứ 2: state chỉ được phép đọc, Cách duy nhất để thay đổi State của ứng dụng là phát một Action (là 1 object mô tả những gì xảy ra). Trạng thái của ứng dụng không được phép thay đổi “trực tiếp”, trạng thái cũng chỉ là một đối tượng mà thôi, nên việc thay đổi là được. Tuy nhiên, với Redux hay Flux thì trạng thái chỉ thay đổi khi và chỉ khi có một sự kiện xảy ra, giống như ra trận thì chỉ được phép nghe lời từ chỉ huy, mọi tin tức tình báo đều được gởi tới chỉ huy, nếu không có lệnh từ chỉ huy thì tất cả không được phép manh động.
* Nguyên lý thứ 3: sử dụng pure function nhận tham số là state trước đó và action và trả về state tiếp theo. Hàm này gọi là reducer, (pure function là hàm trả về giá trị phụ thuộc duy nhất vào giá trị của tham số, pure function có tập các tham số thì giá trị trả về cũng là một tập tương tự như thế). Việc thay đổi trạng thái của ứng dụng, được thực hiện thông qua các hàm thuần tuý. Đưa vào giá trị sự kiện, trạng thái hiện tại và hàm trả về trạng thái tiếp theo. Dù tương lai ứng dụng của bạn có thể rất lớn, nhưng các hàm reducer này thì chỉ cần nhỏ gọn thay đổi trên từng lá của cây trạng thái, và chúng hoàn toàn có thể kết hợp với nhau tạo thành chuỗi sự kiện. Ví dụ: người click vào menu (một sự kiện => thay đổi trạng thái), sau đó router cũng cần thay đổi để phù hợp với ngữ cảnh.

Redux data folow



Data follow in Redux

* UI gọi tới storedispatch (action)
* Reducer store gọi reducer nhận state hiện tại và action vừa được gửi đi
* Root reducer có thể kết hợp nhiều reducer trong state tree để đưa ra kết quả - state tiếp theo của ứng dụng. Sử dụng combineReducers().
* Store lưu toàn bộ state được trả về bởi root reducer.

Tiểu về Data flow trong react

Trong một ứng dụng thực tế, chúng ta có nhiều thành phần hơn, và chúng ta sẽ khó mà hiểu hết được các hoạt động của Redux nếu chỉ nhìn vào 4 thành phần cơ bản trên. Chúng ta sẽ tìm hiểu kĩ hơn thông qua các hình tượng minh họa sau.

Các nhân vật

Tưởng tượng ứng dụng của chúng ta là một văn phòng, có các thành viên làm việc với nhau để giải quyết công việc chung. Cùng điểm qua các nhân vật trong phòng trước khi xem các họ tương tác với nhau để giải quyết công việc:

**The Action creators**

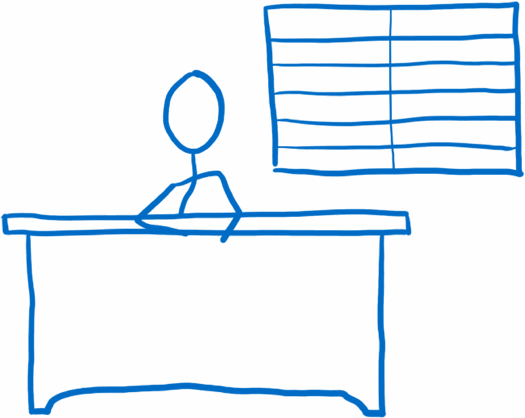


Đầu tiên là The Action Creators. Anh ta giữ nhiệm vụ tạo ra các Action, là bước đầu tiên trong luồng mà các thay đổi và tương tác đều đi qua. Bất cứ khi nào trạng thái của app hay là render của view thay đổi thì đầu tiên là một hành động sẽ được tạo ra.

Hãy hình dung anh này như một anh chàng đánh máy, anh ta biết bạn cần truyền đạt điều gì và cần phải đánh ra văn bản theo định dạng nào cho mọi người đều hiểu được.

The Action Creator tạo ra một action là formated object chứa type và thông tin của action đó. Type thường sẽ là một hằng số được định nghĩa trước, kiểu như INCREASE hay DECREASE.

**The Store**



Tiếp theo là The Store. Hãy hình dung đây là ông sếp đầy quyền lực, toàn bộ các thao tác với State tree (getState, updateState, registerListener...) đều do ông này quản lý.

Cái ông này làm nhiều việc nhỉ (?) Không thực ra ông ấy chỉ quản lý trạng thái của State tree thôi. Khi nhận được Action ông ấy sẽ đi hỏi The reducers xem State sẽ thay đổi ra sao chứ không tự làm.

Nếu đã biết qua Flux thì hẳn là bạn nhận ra ông này nhận trực tiếp Action mà không thông qua Dispatcher. Đó là bởi vì trong Redux, nhờ tiếp nhận tư tưởng của Functional Programming, ông sếp này tự biết cách hiểu Action và tự điều phối nó (dispatch()), nên không cần thuê thêm anh Dispatcher.

**The Reducers**



Tiếp theo là The Reducers. Khi The Store muốn biết State thay đổi như thế nào, ông ấy sẽ gọi cho The Reducers. Ở đây có một ông là Root Reducer nữa, ông này sẽ chịu trách nhiệm cắt ra State cần thay đổi dựa trên keys mà The Store gửi cho và đưa nó cho Reducer biết cách xử lý.

Hãy hình dung đây là một nhóm các thanh niên cuồng Photocopy (yaoming). Họ không thích làm rối tung những thứ họ được đưa cho, nên họ tạo ra bản sao của chúng và thực hiện thay đổi trên bản sao đó.

Đây là một trong những ý tưởng quan trọng của Redux. State không được thao tác trực tiếp. Thay vào đó, mỗi phần được sao chép và sau đó tất cả các phần được kết hợp thành một đối tượng trạng thái mới.

Các reducer gửi bản sao của chúng cho root reducer, và root reducer sẽ ghép các bản sao với nhau để tạo State mới. Sau đó, root reducer sẽ gửi các State mới trở lại Store và Store sẽ sử dụng nó như State chính thức mới.

Nếu ta có một ứng dụng nhỏ, ta chỉ có thể có một reducer làm cho một bản sao của toàn bộ State và và thay đổi nó. Hoặc nếu ta có một ứng dụng lớn, ta có thể có một Cây Reducers (Tree of Reducers) . Đây là một sự khác biệt giữa Flux và Redux. Trong Flux, các Store không nhất thiết phải kết nối với nhau và có một cấu trúc phẳng. Trong Redux, Reducers là một hệ thống cấp bậc, hệ thống cấp bậc này có thể có nhiều mức độ cần thiết, giống như các hệ thống cấp bậc component.

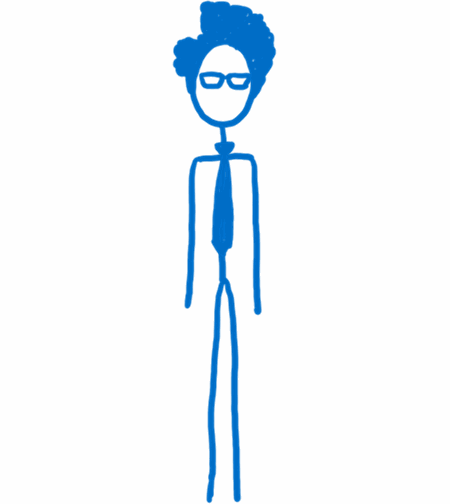
**The views: smart and dumb components**



Trong Redux có 2 khái khái niệm: smart and dumb components.

* Smart component: có thể gọi là containers
  + Hãy hình dung đây là anh quản lý của nhóm nhỏ, anh ta phụ trách các Action. Khi các thành viên dưới anh ta (dumb components) cần phát 1 action, anh ta sẽ gửi action cho các thành viên dưới dạng props, các thành viên chỉ cần coi đó là các callback mà không quan tâm nó là cái gì.
  + Anh ta không thích ăn diện (không có css).
  + Khi có việc cần thay đổi (DOM) thì anh ta sẽ sắp xếp các thành viên dưới làm chứ hiếm khi tự làm.
* Dumb components: có thể gọi là components
  + Hãy hình dung đây là mấy thanh niên học việc, thiên lôi chỉ đâu đánh đấy.
  + Mấy thanh niên này không phụ thuộc trực tiếp vào các Action, vì được anh quản lý đưa cho rồi. Điều này có nghĩa là mấy thanh niên này có thể đưa sang bộ phận khác làm cũng đc, miễn là có anh quản lý đưa "hàng" cho xài.
  + Mấy thanh niên này thì đẹp trai, tóc tai vuốt vuốt các thứ (có css riêng), nhưng đôi khi bị cấp trên bắt mặc theo ý sếp (nhận props style) thì vẫn phải chịu.

**The view layer binding**



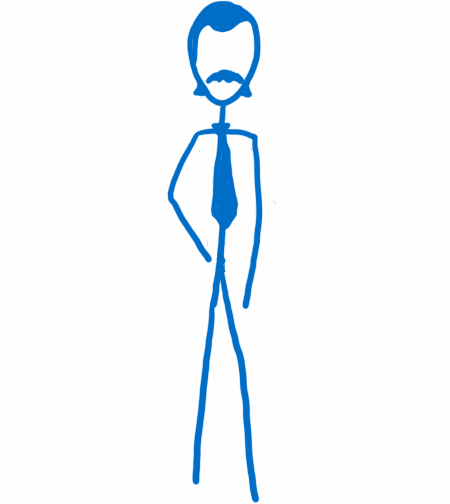
Để The Store giao tiếp đc với The views, chúng ta cần một ai đó kết nối họ lại với nhau, và chúng ta có The view layer binding, với React anh ta tên là react-redux.

Hãy hình dung anh này là nhân viên IT chịu trách nhiệm đảm bảo mạng trong công ty thông suốt để các components kết nối được với Store. Đồng thời cũng quản lý một đống thông tin kỹ thuật mà trong văn phòng chả ai hiểu (yaoming)

The view layer binding cung cấp 3 khái niệm:

* The Provider component: là thành phần bao quanh components tree. Giúp các components con kết nối với Store dễ dàng thông qua connect()
* connect() : là function cung cấp bởi The view layer binding như react-redux. Nếu một component muốn nhận được update State, nó phải tự bao lại bằng connect(). Sau đó, connect function sẽ thiết lập tất cả các hệ thống liên kết cho nó, bằng cách sử dụng selector
* selector : Đây là function mà bạn viết. Nó chỉ rõ phần nào của State mà component cần như properties.

**The root component**



Cuối cùng là The root component. Tất cả các React app đều có The root component. Là những component cao nhất của hệ thống component. Ở Redux thì nó đảm nhận nhiều trách nhiệm hơn.

Hãy hình dung đây là một giám đốc C-level (CEO, COO....). Ông giám đốc này sẽ tạo ra The Store và chỉ định The Reducers nào được sử dụng, tập hợp The view layer binding cùng với The views.

Sau khi chỉ định và tập hợp các thành phần trong team, giám đốc của chúng ta sẽ để cho các bộ phận bên dưới tự hoạt động.

Phối hợp giữa các nhân vật kể trên

Qua hết các nhân vật rồi, giờ cùng xem họ tương tác với nhau như thế nào để giải quyết công việc

#### Setup

Các bộ phận cần được nối với nhau. Việc này xảy ra lần đầu vào app.

**1. Bảo ông Store sẵn sàng**: ông giám đốc Root component tạo ra Store, chỉ cho ông Store dùng Root Reducer nào thông qua createStore(). Ông Root Reducer thì đã có team sẵn rồi, được tập hợp lại thông qua combineReducers()



**2. Chuẩn bị liên lạc giữa ông Store và các bộ phận khác.**

Root component bao các subcomponents với provider component (The view layer binding) và tạo kết nối giữa Store với các Provider.

Provider tạo ra 1 mạng cơ bản để cập nhật các components. Smart Components kết nối vào mạng bằng connect(), điều này đảm bảo họ nhận được cập nhật State.



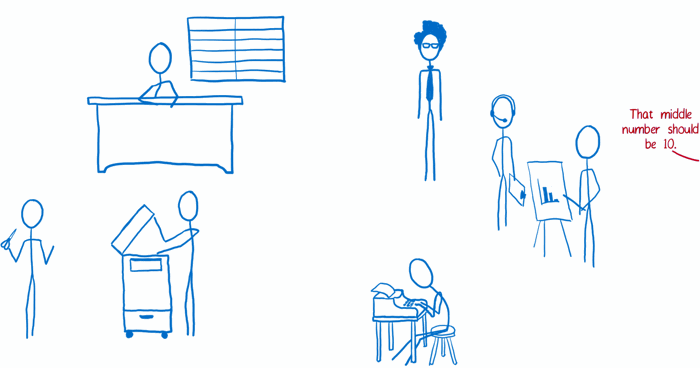
**3. Chuẩn bị các actions callback**

Để các Dump Components làm việc với Action dễ dàng hơn, các Smart Components có thể chuẩn bị các action callback thông qua bindActionCreators(). Bằng cách này, họ chỉ có cần truyền các callback cho Dump Components. Các Actions sẽ được tự động gửi đi sau khi nó được định dạng.

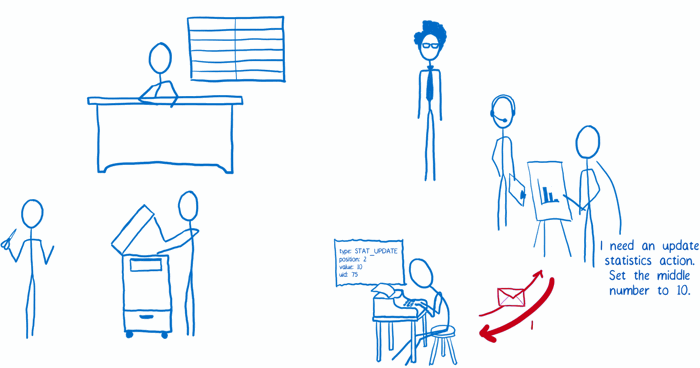


Flow

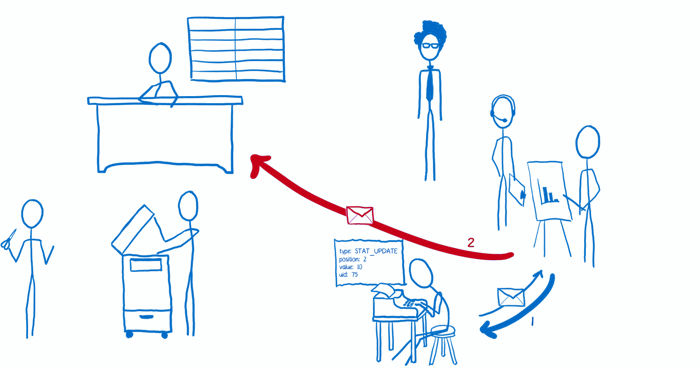
Sau phần setup thì web app đã sẵn sàng nhận input từ người dùng. Khi này trigger một action bằng cách để người dùng tạo ra một sự thay đổi.



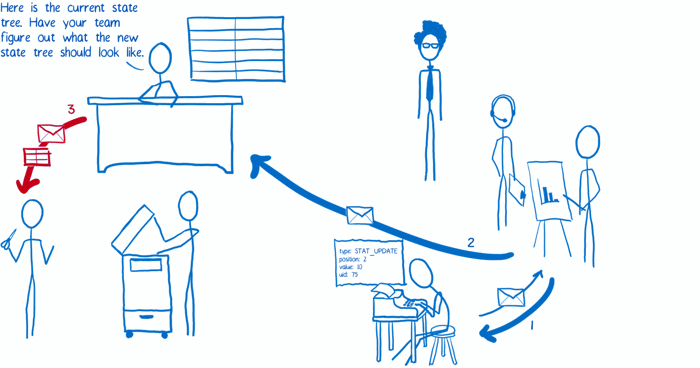
1. The View yêu cầu 1 action. Action Creator định dạng (format) yêu cầu và gửi lại



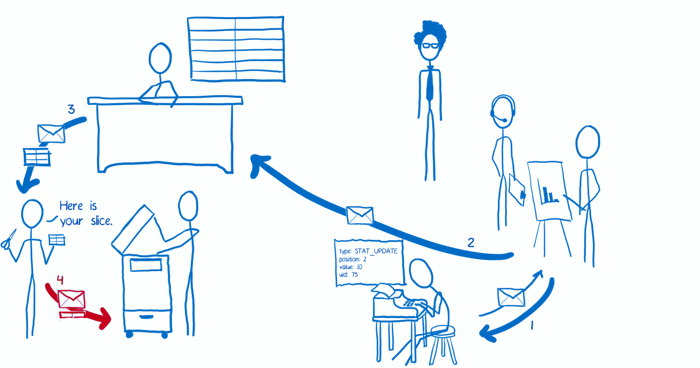
1. Action được gửi tự động (nếu bindActionCreators() đã được chuẩn bị) hoặc The View sẽ gửi



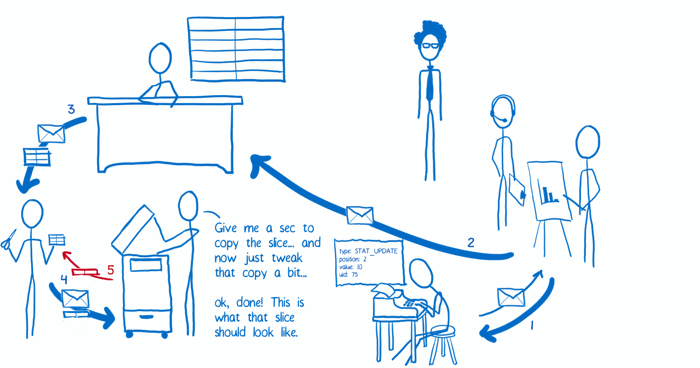
1. The Store nhận Action sau đó gửi State tree hiện tại và Action cho Root Reducer.



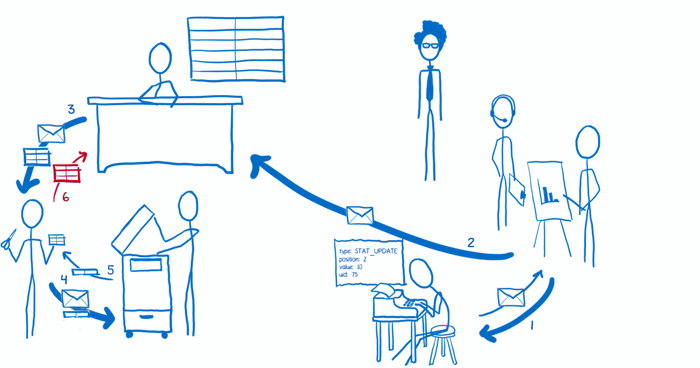
1. Root Reducer chia State ra thành nhiều phần và gửi cho từng subreducers biết cách xử lý chúng.



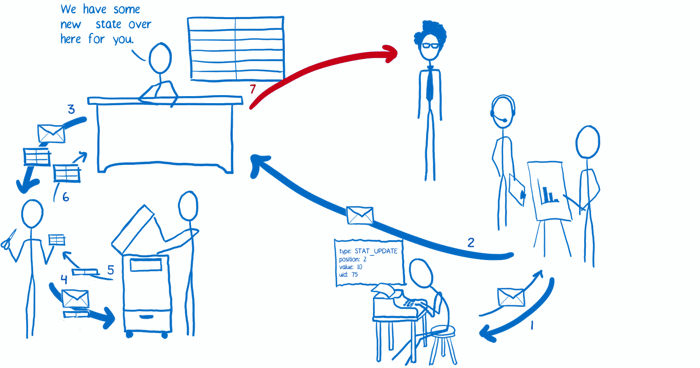
1. Subreducers tạo ra 1 bản copy từ phần nhận được và thay đổi trên bản copy. Sau đó gửi lại bản copy cho Root Reducer



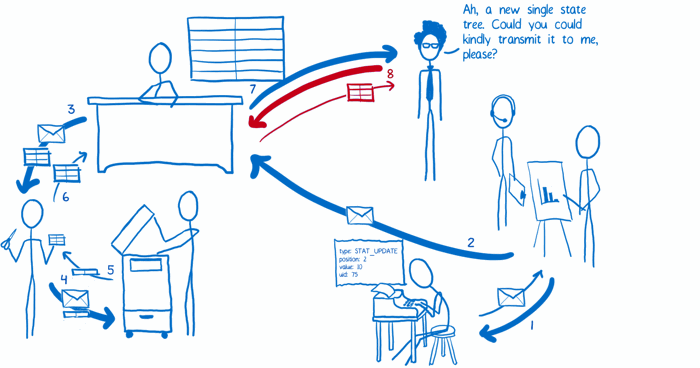
1. Khi tất cả subreducers trả về các phần copies, Root Reducer ghép chúng lại tạo thành 1 update State tree và gửi lại cho Store. Store thay thế State tree cũ bằng State tree mới.



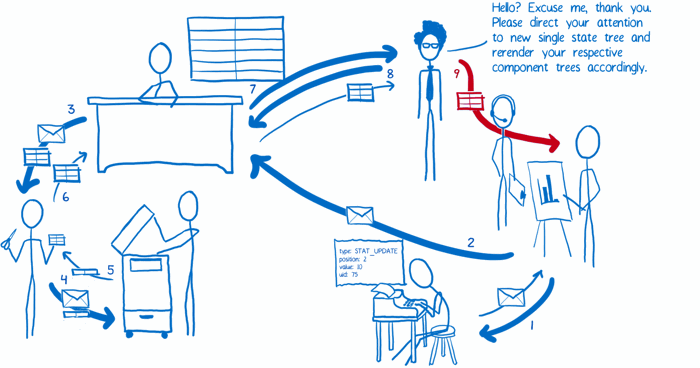
1. Store nói với The view layer binding là có State mới.



1. The view layer binding báo Store gửi State mới cho mình



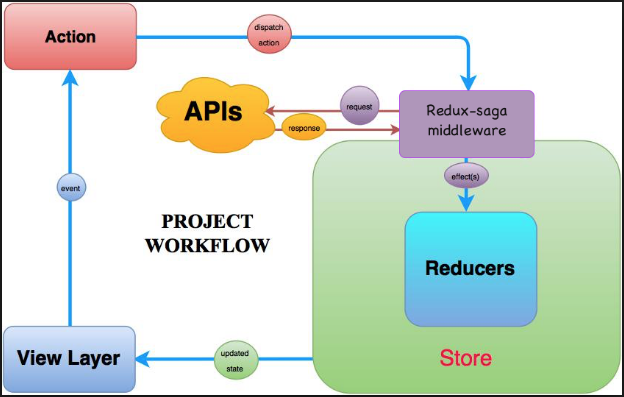
1. The view layer binding kích hoạt render view



Redux-Saga

Redux-Saga là một thư viện redux middleware, giúp quản lý những side effect trong ứng dụng redux trở nên đơn giản hơn. Bằng việc sử dụng tối đa tính năng [Generators](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Statements/function*" \t "_blank) (function\*) của ES6, nó cho phép ta viết async code nhìn giống như là synchronos.

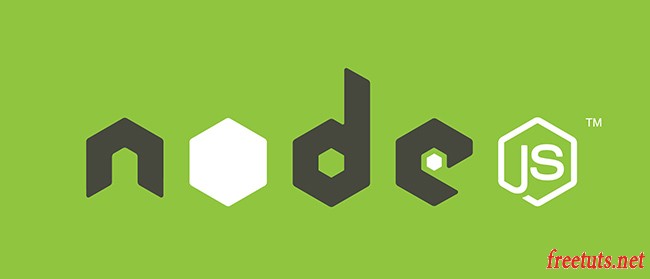
Đối với logic của saga, ta cung cấp một hàm cho saga, chính hàm này là hàm đứng ra xem xét các action trước khi vào store, nếu là action quan tâm thì nó sẽ thực thi hàm sẽ được thực thi, nếu bạn biết khái niệm hook thì hàm cung cấp cho saga chính là hàm hook. Điều đặc biệt của hàm hook này nó là một generator function, trong generator function này có yield và mỗi khi yield ta sẽ trả về một plain object. Object trả về đó được gọi Effect object. effect object này đơn giản chỉ là một object bình thường nhưng chứa thông tin đặc biệt dùng để chỉ dẫn middleware của Redux thực thi các hoạt động khác ví dụ như gọi một hàm async khác hay put một action tới store. Để tạo ra effect object đề cập ở trên thì ta gọi hàm từ thư viện của saga là redux-saga/effects.



* + - 1. **Ngôn ngữ sử dụng phía backend: javascript** 
         1. **Javascript: NodeJS và Framework ExpressJs**

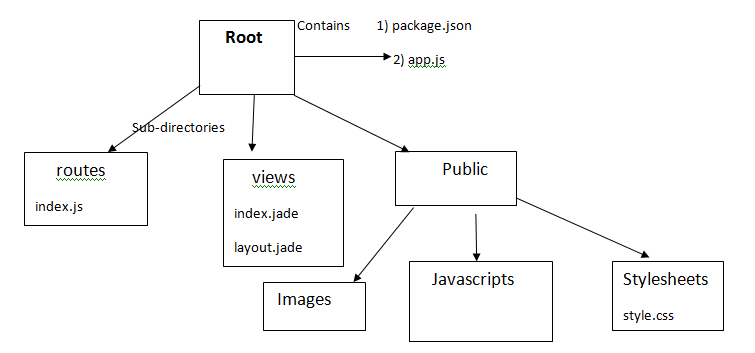
NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.



Khi nói đến NodeJS thì phải nghĩ tới vấn đề Realtime. Realtime ở đây chính là xử lý giao tiếp từ client tới máy chủ theo thời gian thực. Giống như khi bạn lướt Facebook thì mỗi khi bạn comment hay like một topic nào đó thì ngay lập tức chủ topic và những người đã comment trên đó sẽ nhận được thông báo là bạn đã comment.

**Expressjs** là một framework được xây dựng trên nền tảng của **Nodejs**. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. **Expressjs** hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.



*Tổng hợp một số chức năng chính của Expressjs như sau:*

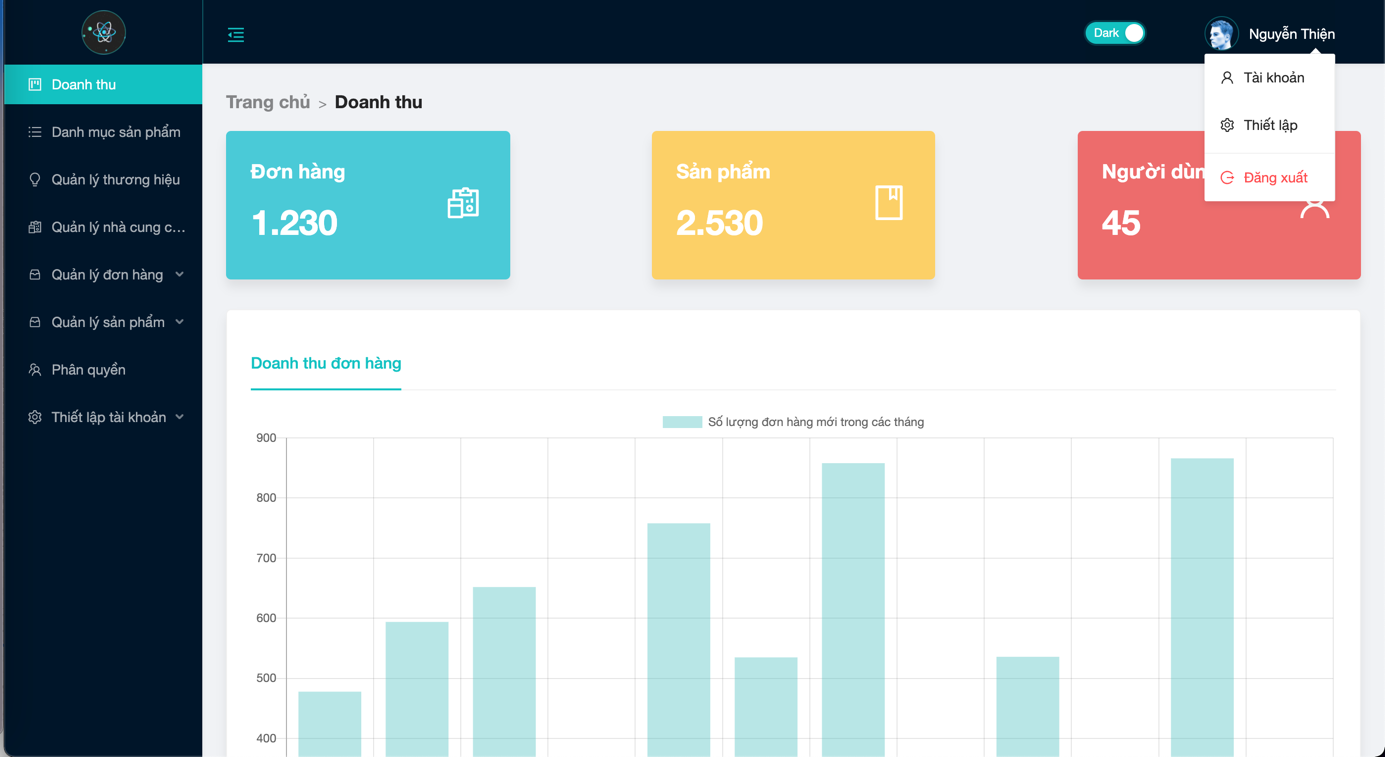
* Thiết lập các lớp trung gian để trả về các HTTP request.
* Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.

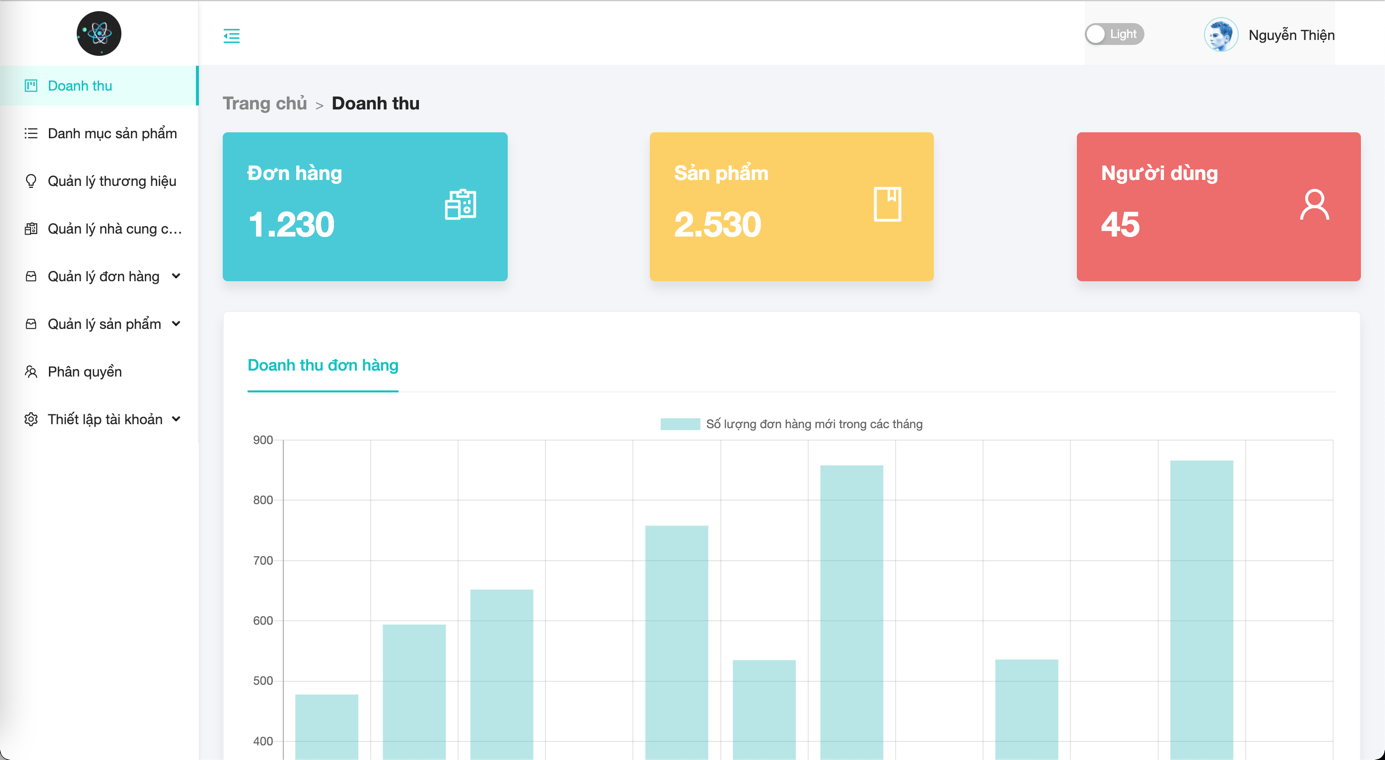
### **4.2. Giới thiệu sản phẩm**

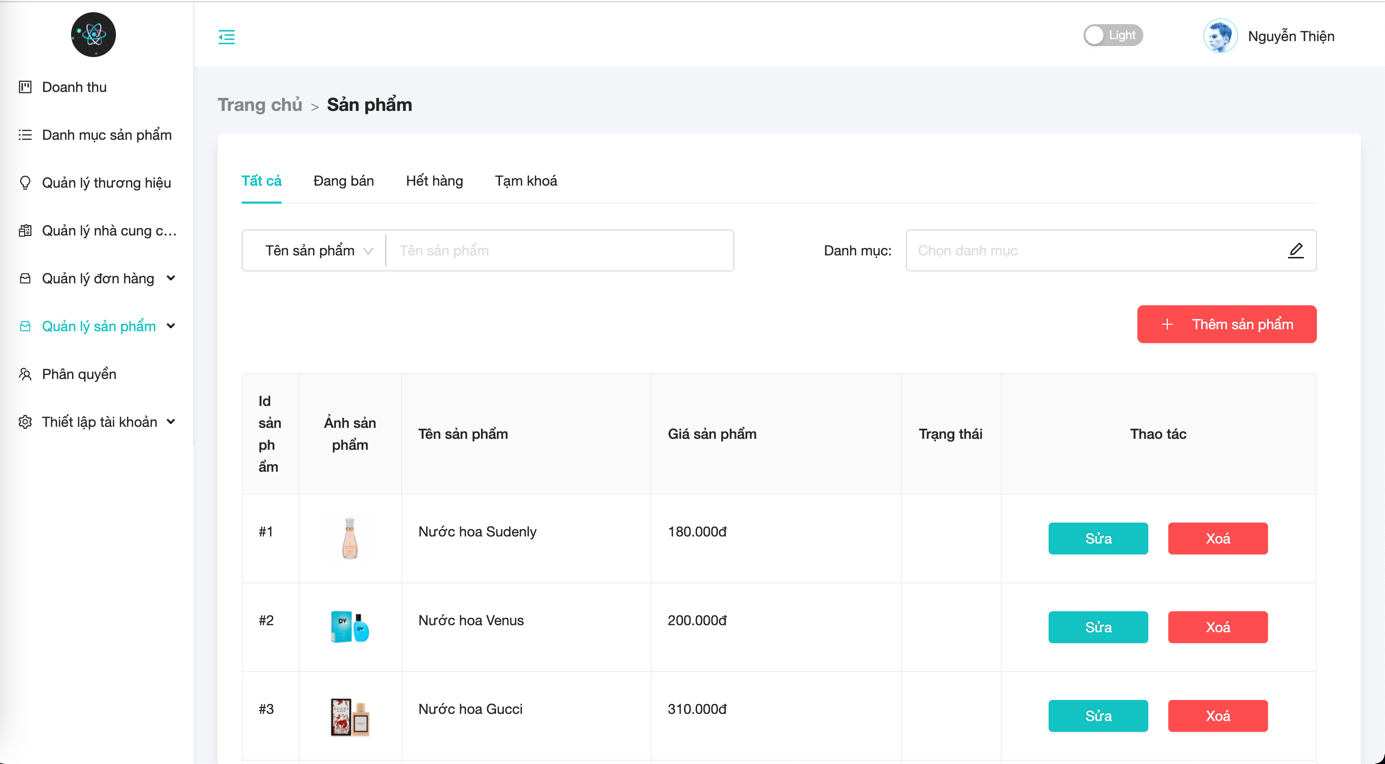
- Mô tả: Ứng dụng được xây dựng trên nền tảng công nghệ ReactJs, Redux, Ant Design bên client kết hợp với NestJs, Postgresql bên backend để giúp người dùng có thể trải nghiệm hệ thống 1 cách nhanh và hiệu quả nhất.

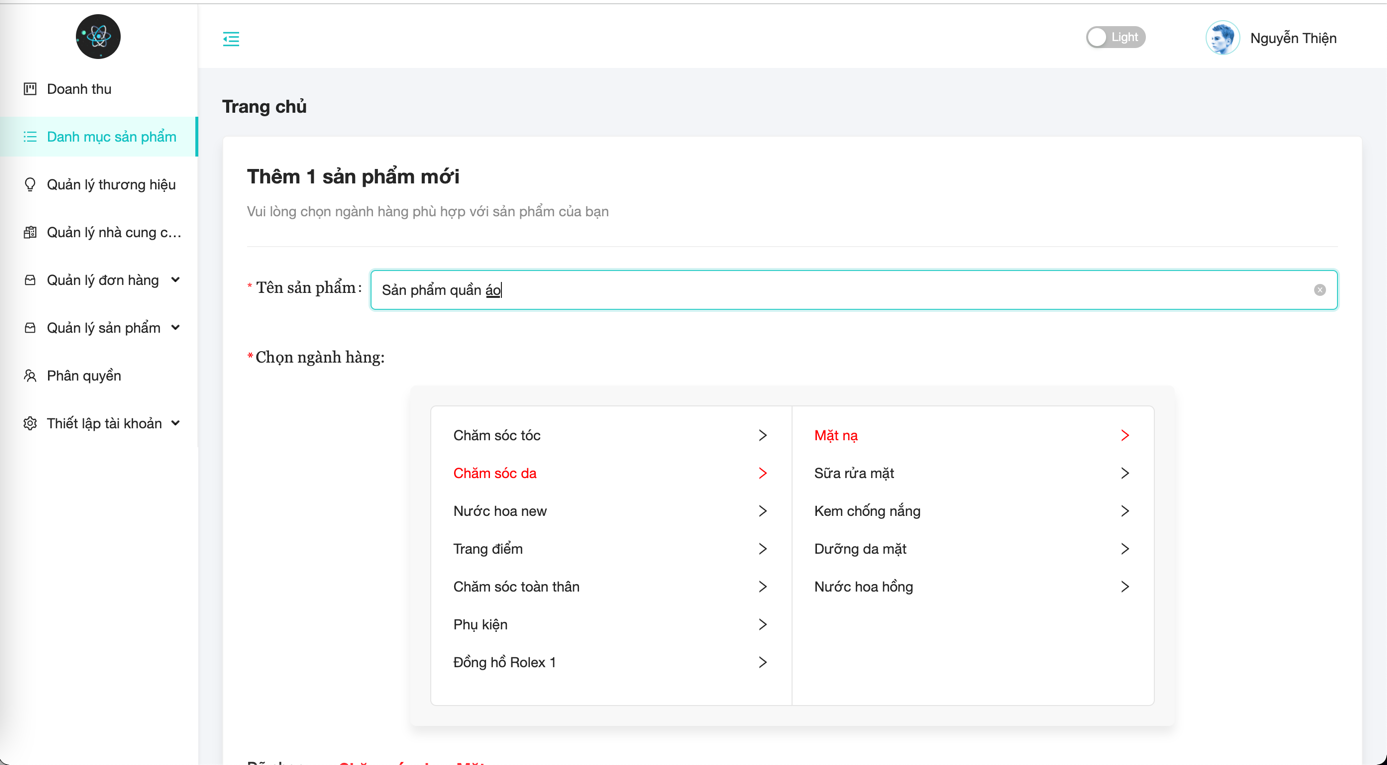
- Một số hình ảnh về sản phẩm:

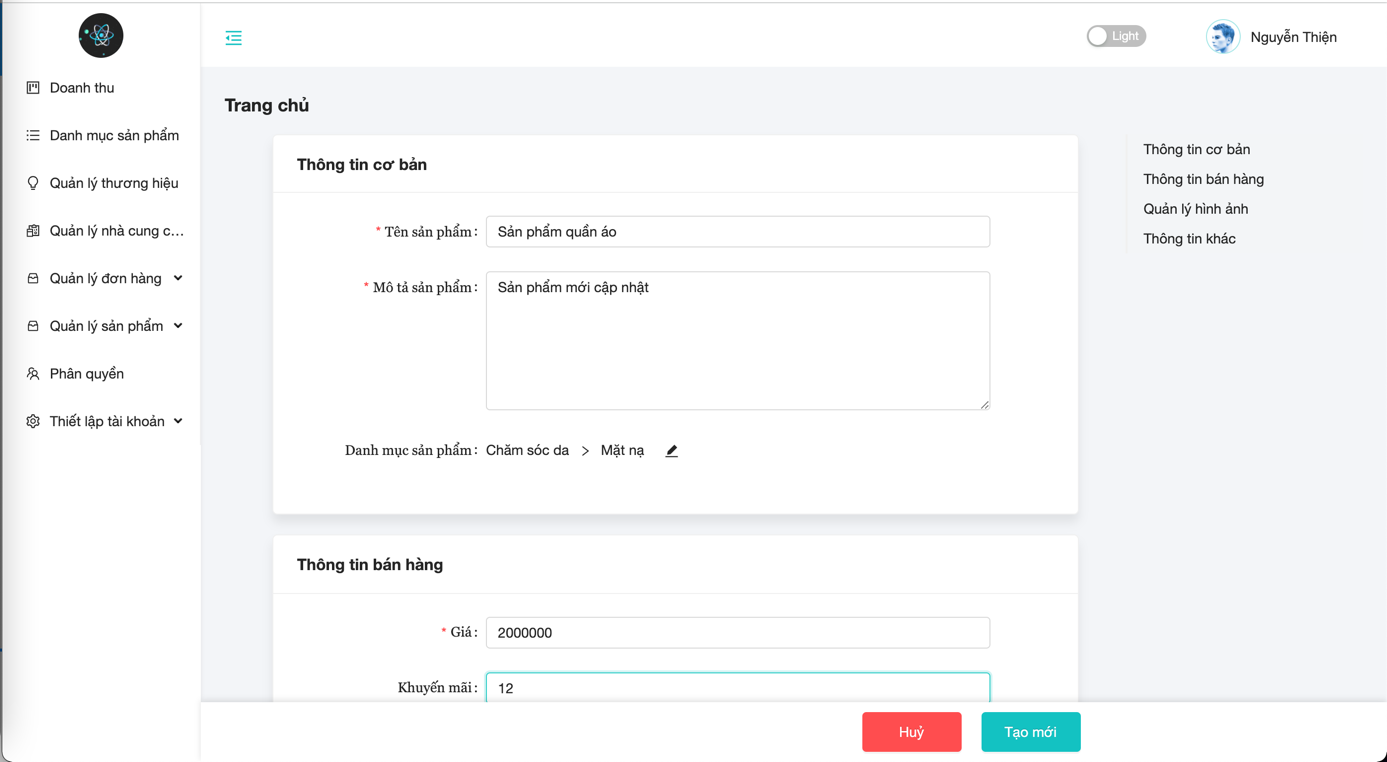
+ Trang quản trị:

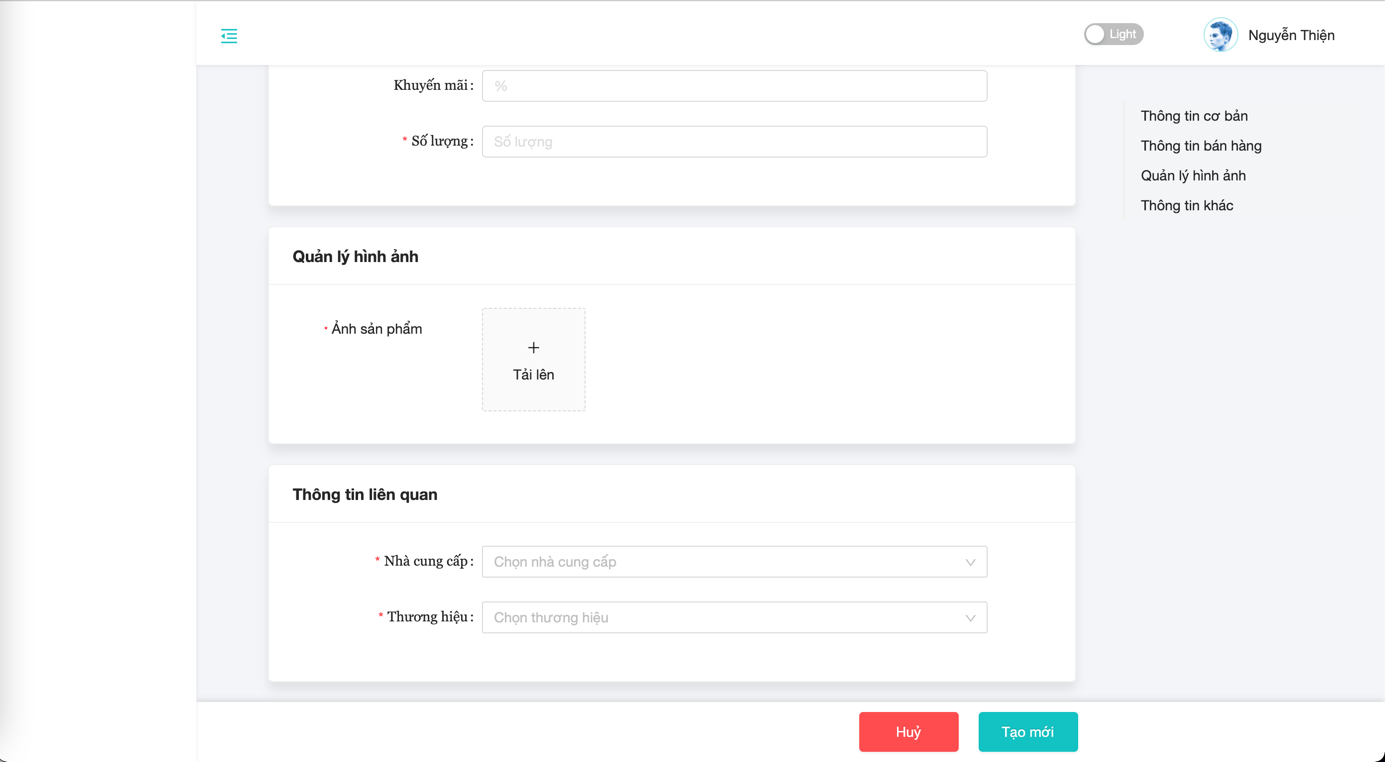


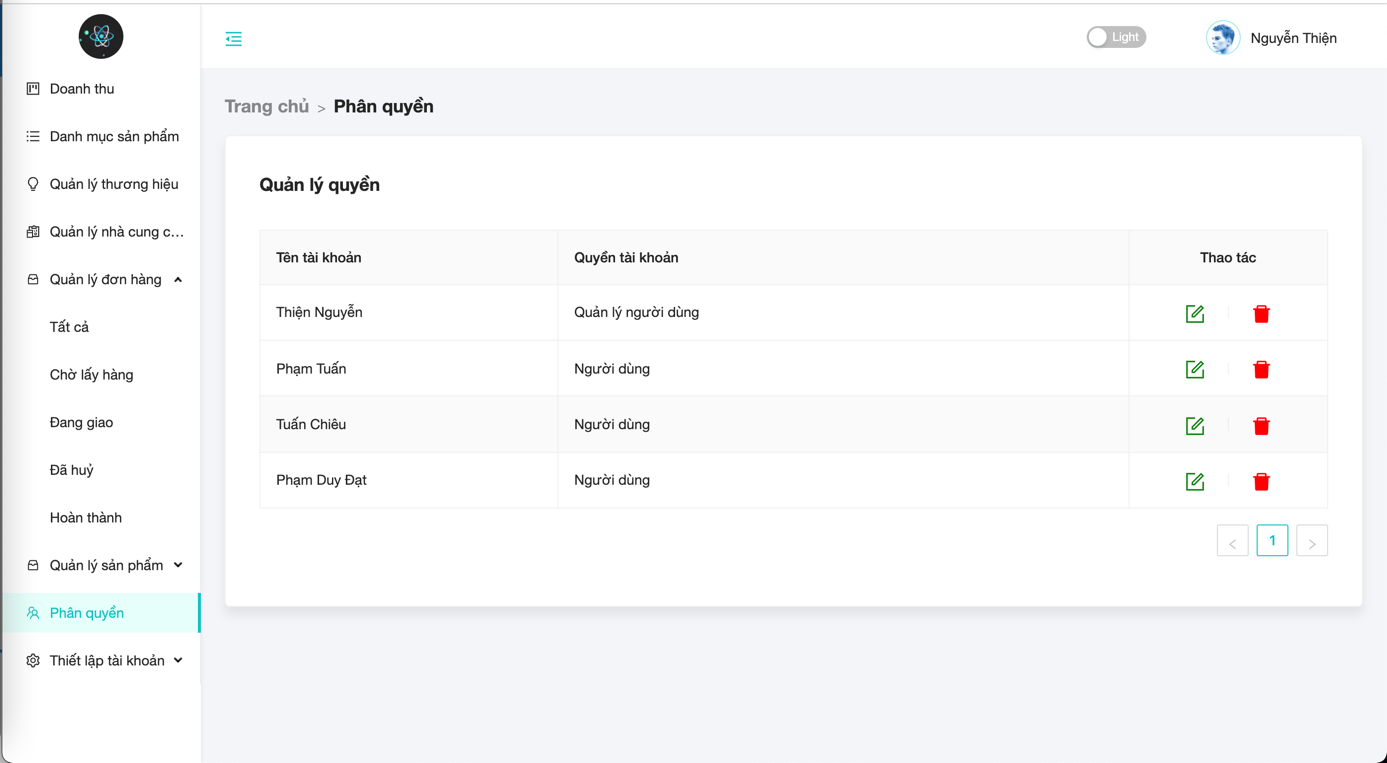


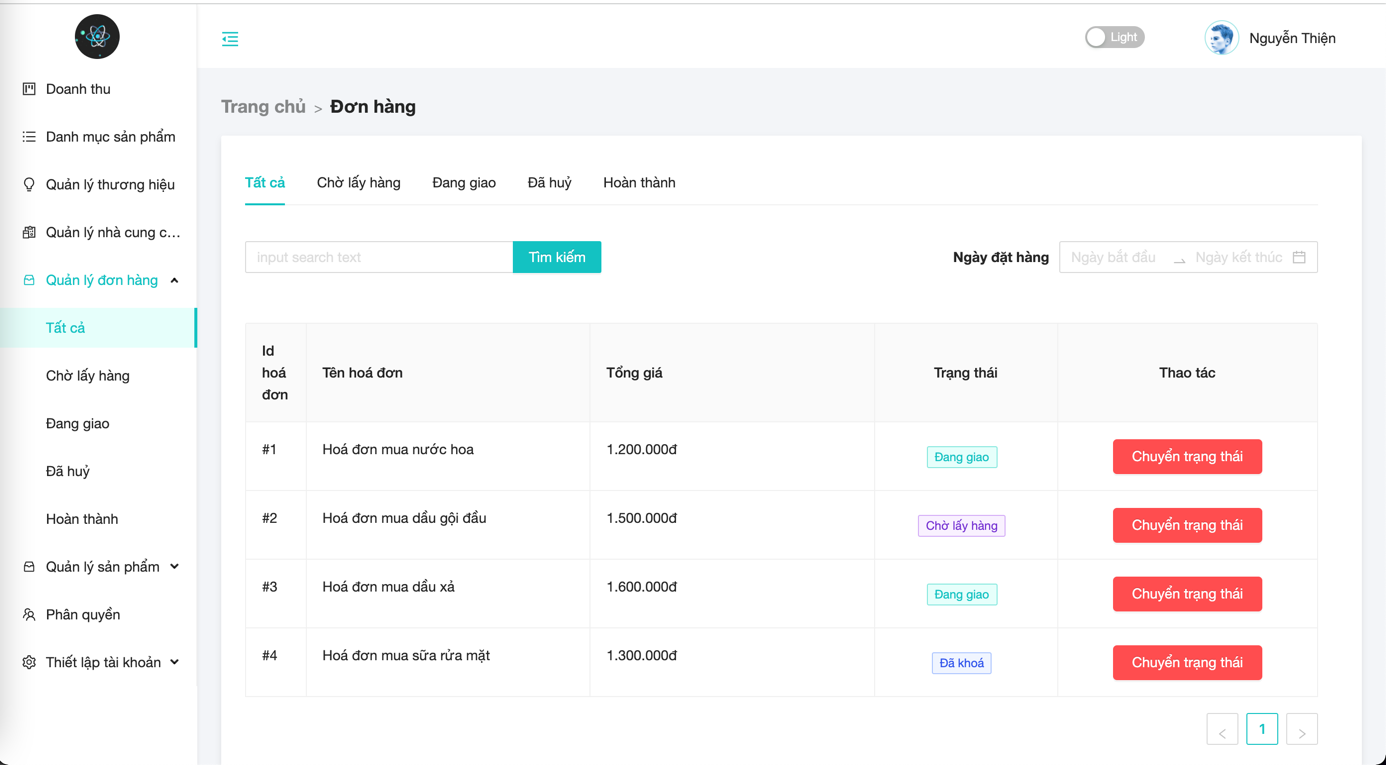




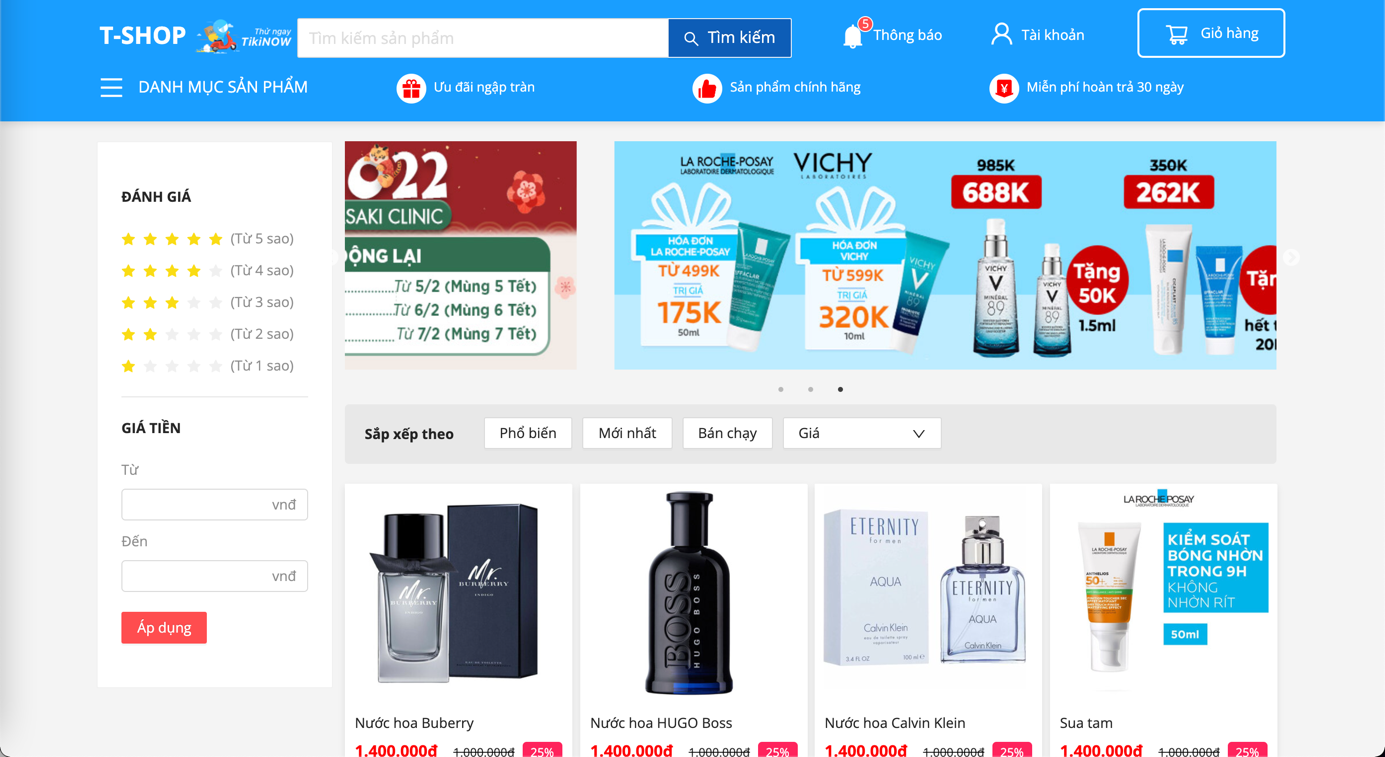


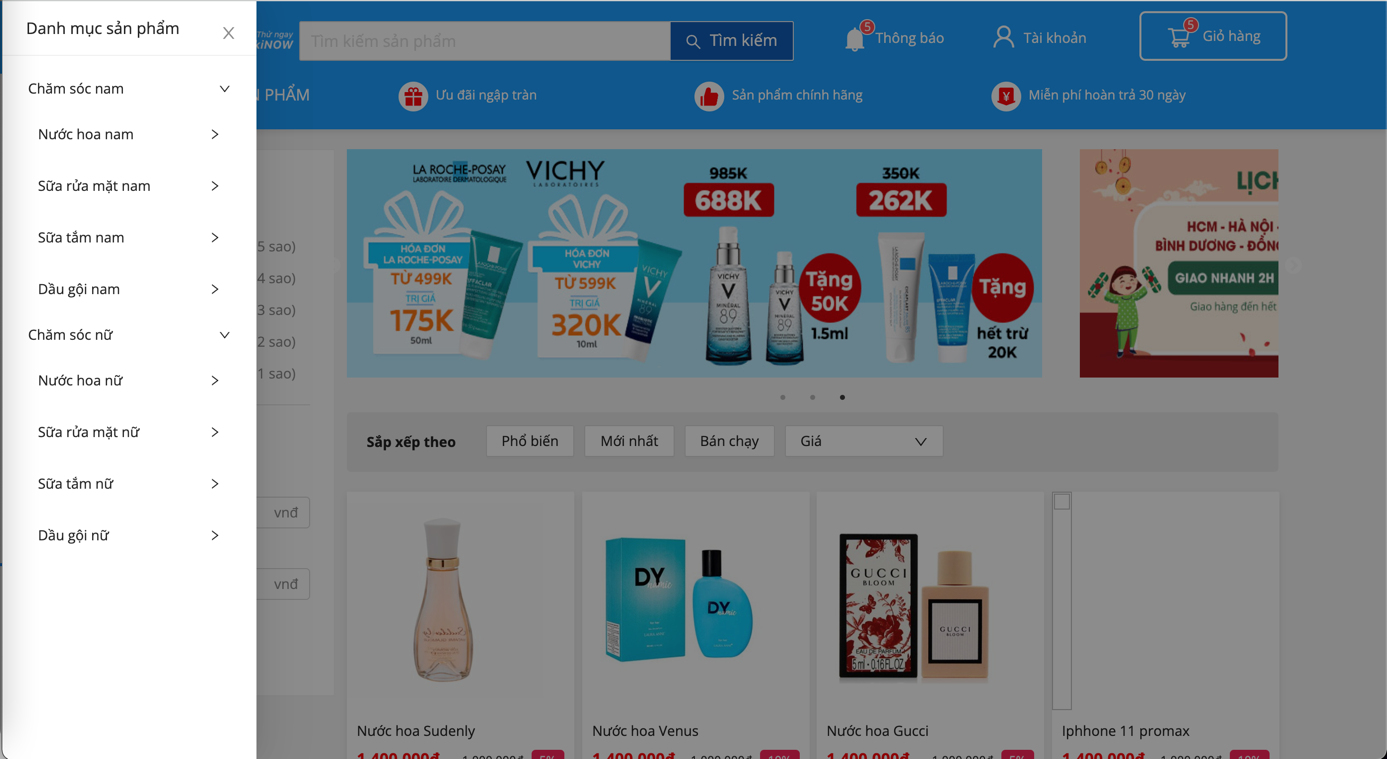


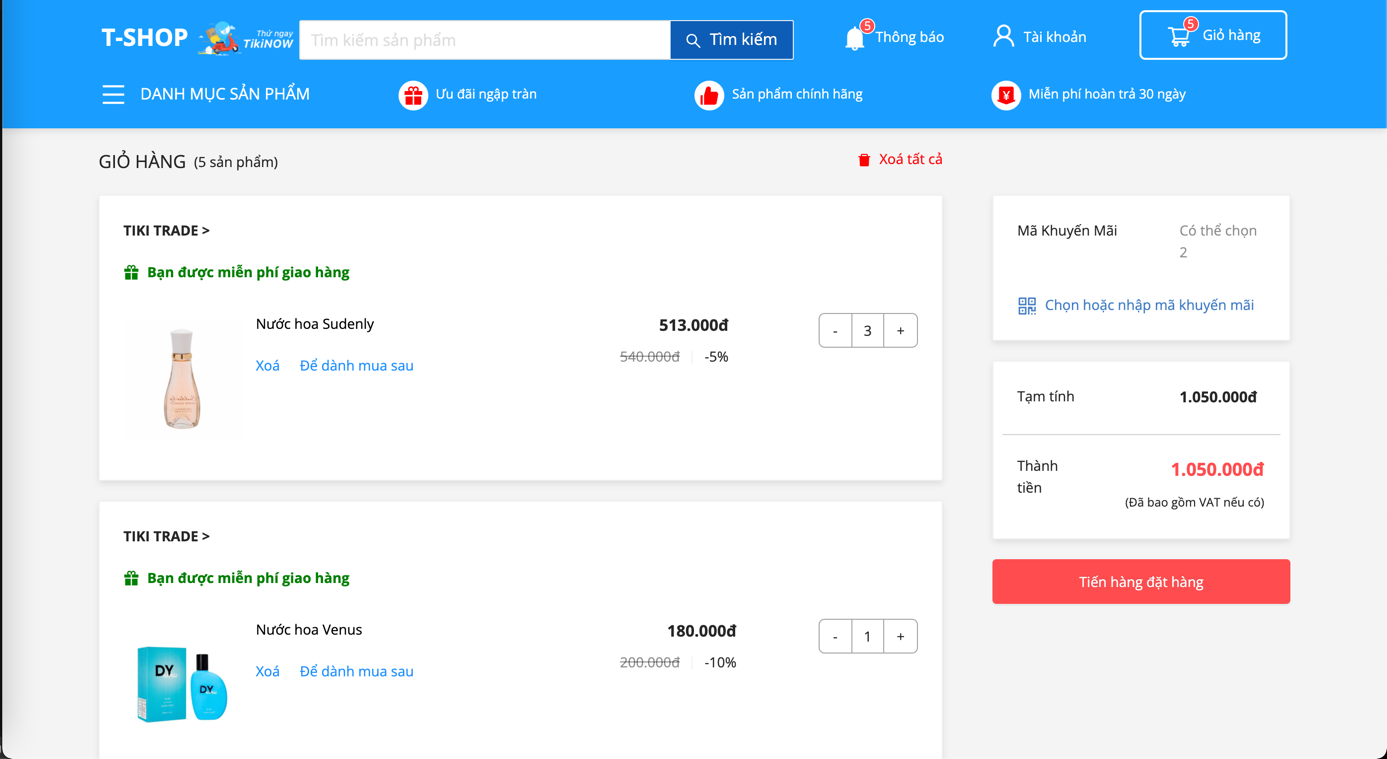


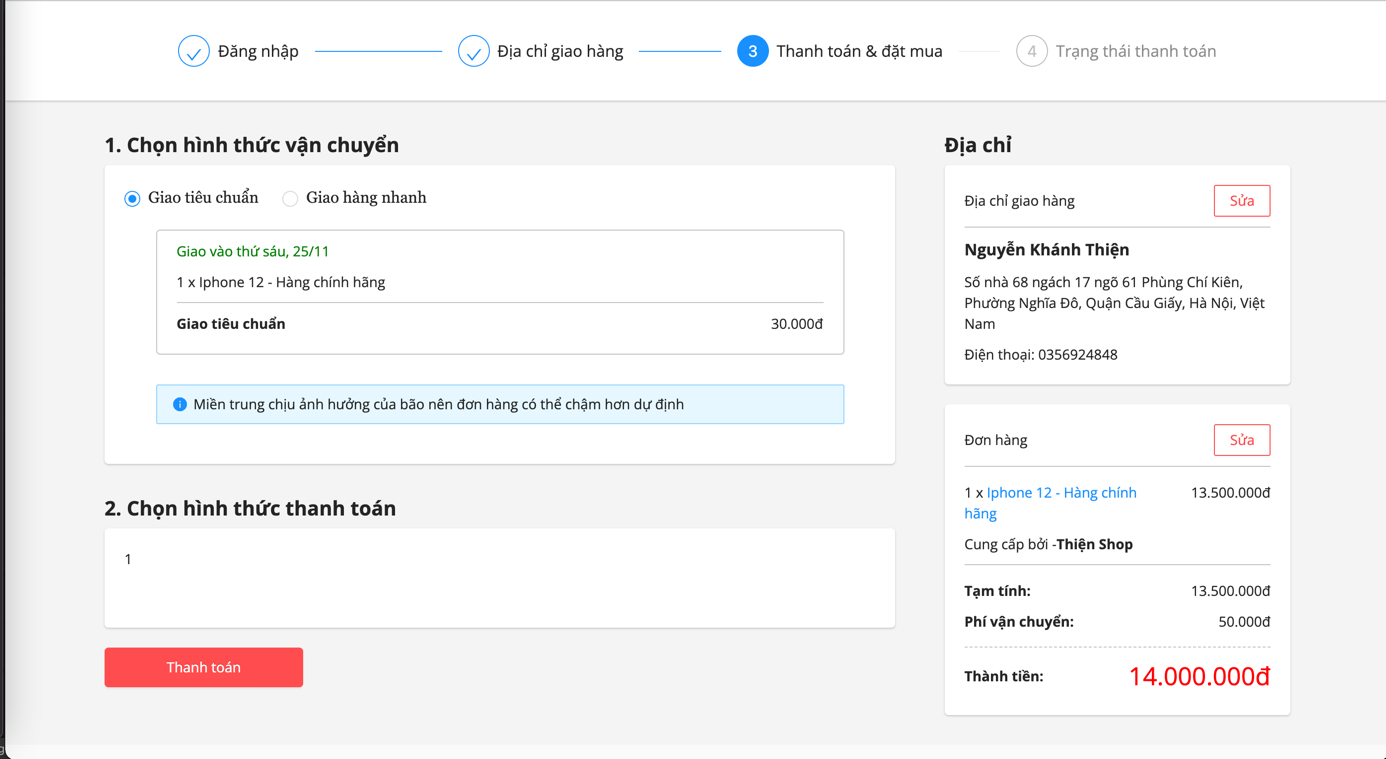


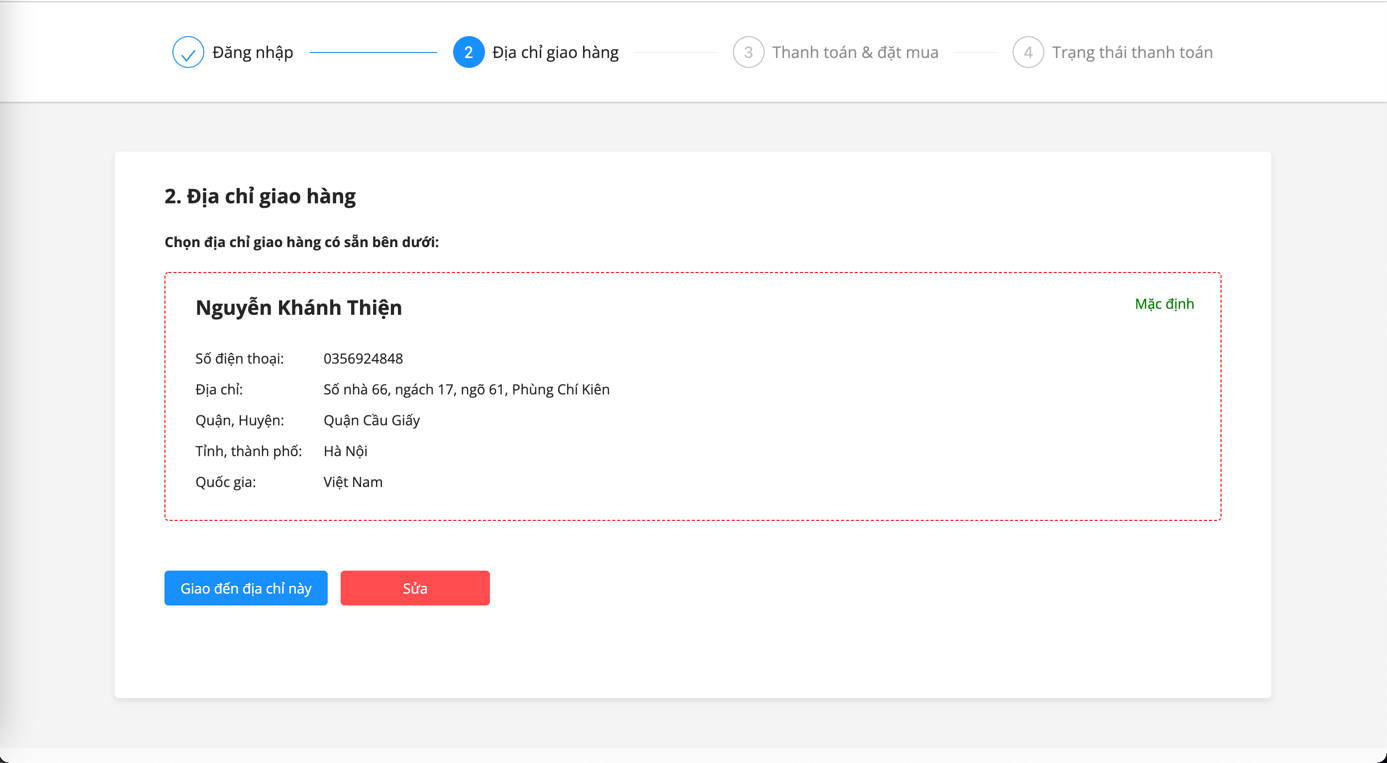
+ Trang sản phẩm:

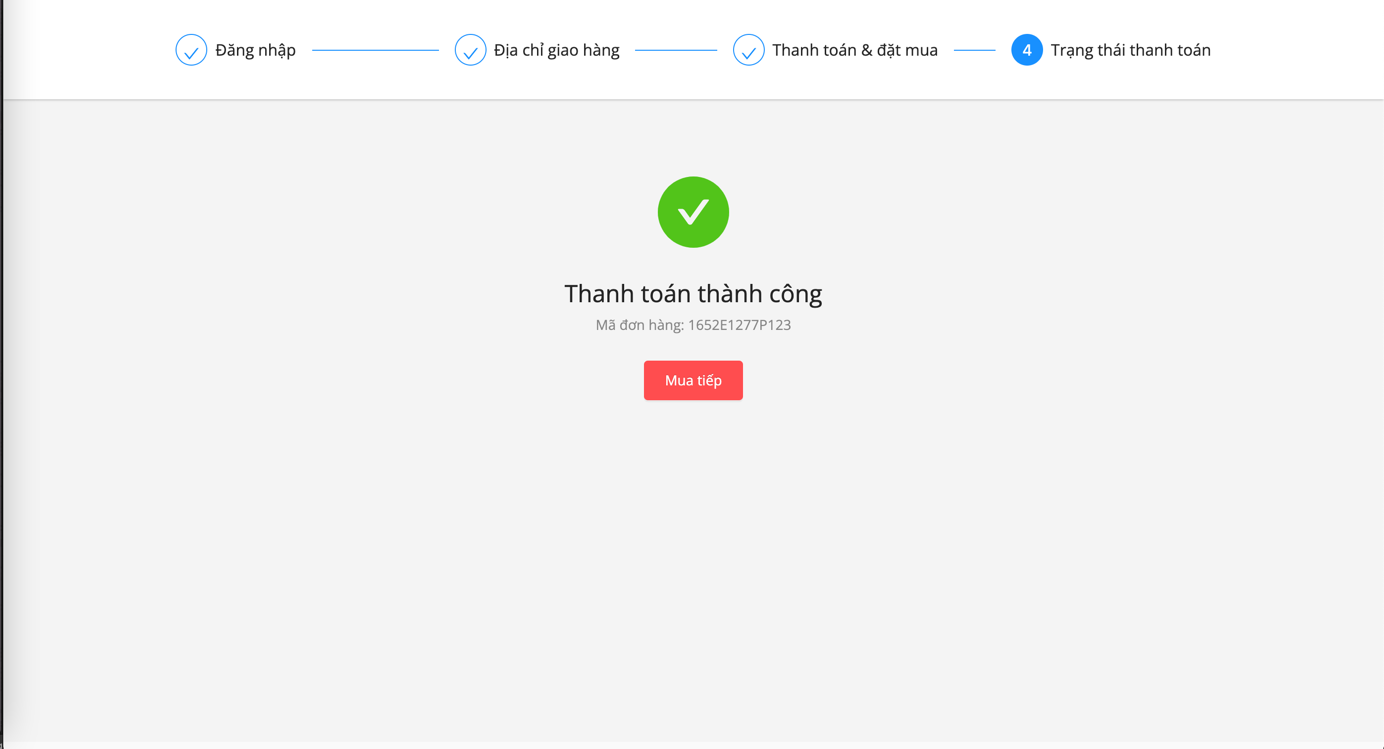












# 

# **KẾT LUẬN**

Sau khi tìm hiểu và nghiên cứu đề tài, sau đây là các vấn đề em đã đạt được khi thực hiện khoá luận:

**Về lý thuyết:**

Nắm được lý thuyết về xây dựng hệ thống website bán mỹ phẩm

Nắm được quy trình xây dựng, phát triển phần mềm

**Về chương trình:**

Cơ bản xây dựng được một hệ thống ứng dụng đáp ứng đầy đủ nhu cầu của doanh nghiệp và người dùng bao gồm 2 phần: ứng dụng web app, hệ thống website quản trị.

**Kiến nghị:**

Do bước đầu làm quen với nghiên cứu và thời gian có hạn nên trong quá trình nghiên cứu và biên soạn, khoá luận sẽ không tránh khỏi những thiếu sót. Kính mong các thầy cô góp ý để giúp em hoàn thiện được kiến thức cũng như tạo điều kiện giúp đỡ để em có thể hoàn thành nhiệm vụ được giao. Em xin chân thành cảm ơn!

**TÀI LIỆU THAM KHẢO**

**1. Tài liệu internet**

[1] <https://reactjs.org/>

[2] <https://nodejs.org/en/>

[3] <https://ant.design/docs/react/introduce>

[4] <https://redux-saga.js.org/>

[5] <https://redux.js.org/>

[6] <https://styled-components.com/>

[7] <https://viblo.asia/p/redux-cho-nguoi-moi-bat-dau-part-1-introduction-ZjleaBBZkqJ>

[8] <https://expressjs.com/>

## **2. Tài liệu sách**

[1] Don't Make Me Think! A Common Sense Approach to Web Usability: Steve Krug

[2] You don’t know JS: Kyle Simpson

[3] Designing with Web Standards: Jeffrey Zeldman và Ethan Marcotte