 ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

****

BÁO CÁO ĐỒ ÁN

**Lập trình trên thiết bị di động**

Đề tài:

**App bán hàng**

GVHD: ***Thầy Huỳnh Tuấn Anh***

*Lớp:SE346.K11*

*Nhóm sinh viên thực hiện:*

* Hoàng Thùy Trang – 16521277
* Hoàng Đức Minh – 16520733

Tp. Hồ Chí Minh, tháng 12 năm 2019

****

BÁO CÁO ĐỒ ÁN

**Lập trình trên thiết bị di động**

Đề tài:

**App bán hàng**

GVHD: *Thầy Huỳnh Tuấn Anh*

*Lớp: SE346.K11*

*Nhóm sinh viên thực hiện:*

* Hoàng Thùy Trang – 16521277
* Hoàng Đức Minh – 16520733

Tp. Hồ Chí Minh, tháng 12 năm 2019

# LỜI CẢM ƠN

Quá trình học tập và rèn luyện tại khoa Công nghệ phần mềm - Trường Đại học Công nghệ Thông tin, chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể áp dụng vào thực hiện các Project, các bài tập lớn. Chúng em xin gửi lời cảm ơn chân thành đến quý Thầy, Cô khoa Công nghệ phần mềm đã truyền đạt cho chúng em những kiến thức bổ ích ấy để chúng em có thể hoàn thành tốt đồ án môn học của mình.

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc nhất tới thầy Huỳnh Tuấn Anh, là người trực tiếp hướng dẫn, truyền đạt kiến thức môn Lập trình trên thiết bị di động để chúng em có thể hoàn thành tốt nhất đồ án môn học này.

Trong quá trình thực hiện đồ án không khỏi mắc phải những sai sót, kính mong nhận được sự góp ý của thầy và các bạn để nhóm có thể hoàn thiện đồ án của mình hơn nữa, đó cũng là kinh nghiệm quý giá để mỗi người trong nhóm áp dụng vào những đồ án sau này và công việc trong tương lai. Xin chân thành cảm ơn!

Tp. Hồ Chí Minh, tháng 12 năm 2019

# NHẬN XÉT CỦA GIẢNG VIÊN

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

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc26858309)

[NHẬN XÉT CỦA GIẢNG VIÊN 4](#_Toc26858310)

[MỤC LỤC 5](#_Toc26858311)

[CHƯƠNG 1: TỔNG QUÁT VỀ ĐỀ TÀI 7](#_Toc26858312)

[1.1. Đặt vấn đề. 7](#_Toc26858313)

[1.2. Yêu cầu của hệ thống bán hàng. 8](#_Toc26858314)

[1.2.1. Nhu cầu người sử dụng. 8](#_Toc26858315)

[1.2.2. Với người quản trị. 8](#_Toc26858316)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 9](#_Toc26858317)

[2.1. Nền tảng NodeJs 9](#_Toc26858318)

[2.1.1. Khái niệm 9](#_Toc26858319)

[2.1.2. Các đặc tính của NodeJs 9](#_Toc26858320)

[2.1.3. Những điều mà NodeJs có thể làm 10](#_Toc26858321)

[2.1.4. Khi nào nên sử dụng NodeJs 10](#_Toc26858322)

[2.1.5. Các thành phần của NodeJs 11](#_Toc26858323)

[2.1.6. Tại sao nên sử dụng NodeJs 11](#_Toc26858324)

[2.2. MongoDB 12](#_Toc26858325)

[2.2.1. Khái niệm 12](#_Toc26858326)

[2.2.2. Cách thức hoạt động 13](#_Toc26858327)

[2.2.3. Các khái niệm cơ bản 13](#_Toc26858328)

[2.2.4. Ưu, nhược điểm 15](#_Toc26858329)

[2.2.5. Khi nào nên sử dụng và không nên sử dụng 15](#_Toc26858330)

[2.3. Android 16](#_Toc26858331)

[2.3.1. Android, nền tảng di động phổ biến nhất thế giới 16](#_Toc26858332)

[2.3.2. Framework phát triển mạnh mẽ 16](#_Toc26858333)

[2.3.3. Thị trường mở để phân phối các ứng dụng của bạn 17](#_Toc26858334)

[2.3.4. Kiến trúc android 18](#_Toc26858335)

[2.3.5. Cấu trúc một Project Android 21](#_Toc26858336)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 29](#_Toc26858337)

[3.1. Kiến trúc hệ thống 29](#_Toc26858338)

[3.2. Thiết kế 29](#_Toc26858339)

[3.2.1. Cơ sở dữ liệu 29](#_Toc26858340)

[3.2.2. Mô hình phân rã chức năng 33](#_Toc26858341)

[3.2.3. Use case 34](#_Toc26858342)

[3.2.4. Sequence Diagram 55](#_Toc26858343)

[3.2.5. Class Diagram 76](#_Toc26858344)

[CHƯƠNG 4: HIỆN THỰC ỨNG DỤNG 85](#_Toc26858345)

[4.1. App bán hàng 85](#_Toc26858346)

[4.1.1. Màn hình trang chủ 85](#_Toc26858347)

[4.1.2. Màn hình danh mục 87](#_Toc26858348)

[4.1.3. Màn hình tài khoản 88](#_Toc26858349)

[4.1.4. Màn hình danh sách sản phẩm 93](#_Toc26858350)

[4.1.5. Màn hình chi tiết sản phẩm 94](#_Toc26858351)

[4.1.6. Màn hình giỏ hàng 96](#_Toc26858352)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 98](#_Toc26858353)

[5.1. Những công việc làm được 98](#_Toc26858354)

[5.2. Hạn chế 98](#_Toc26858355)

[5.3. Kết luận 99](#_Toc26858356)

[Tài liệu tham khảo 99](#_Toc26858357)

CHƯƠNG 1: TỔNG QUÁT VỀ ĐỀ TÀI

## 1.1. Đặt vấn đề.

Hiện nay với sự phát triển không ngừng của công nghệ thì nhu cầu của con người ngày càng tăng. Trong đó mua hàng online là một trong những nhu cầu không thể thiếu của con người.

*App bán hàng* là một app bán hàng mà nhóm thực hiện để đáp ứng nhu cầu này.

Mục tiêu xây dựng trang app này nhắm giúp cho khách hàng có thể mua hàng trực tiếp từ xa thông qua mạng internet. Khách hàng ở nhà hay tại cửa hàng vẫn có thể dễ dàng tham khảo thông tin sản phẩm mình tìm, so sánh giá cả các mặt hàng và lựa chọn cho mình loại sản phẩm phù hợp nhu cầu của mình, giúp công việc mua sắm một cách nhanh chóng, tiện lợi, tiết kiệm thời gian, đáp ứng được nhu cầu thực tế. Hệ thống tìm kiếm dễ dàng, giao diện thân thiện. Chỉ cần đăng nhập vào hệ thống với tài khoản đã có hay chỉ cần vài thao tác đăng kí đơn giản là khách hàng có thể tự do chọn mua và tạo đơn đặt hàng tại hệ thống.

## 1.2. Yêu cầu của hệ thống bán hàng.

### 1.2.1. Nhu cầu người sử dụng.

* Nhu cầu của khách hàng khi truy cập vào app là tìm kiếm các sản phẩm. Do đó yêu cầu của chương trình là phải đáp ứng được những nhu cầu đó, sao cho khách hàng có thể tìm kiếm nhanh chóng và hiệu quả các loại sản phẩm mà họ muốn và cần mua.
* Chương trình phải có tính đa dạng và hấp dẫn nhằm thu hút sự quan tâm của nhiều người về công ty mình.
* Giao diện phải dễ dùng, hấp dẫn và quan trọng là làm sao cho khách thấy những thông tin cần tìm cũng như thông tin liên quan.
* Điều quan trọng trong mua bán qua mạng là phải đảm bảo an toàn tuyệt đối những thông tin liên quan đến người dùng trong quá trình đặt mua hay thanh toán cũng được đảm bảo hàng được chuyển giao đúng nơi, đúng lúc.

### 1.2.2. Với người quản trị.

App đòi hỏi người quản trị phải thường xuyên theo dõi các thông tin về hàng hóa, xử lý đúng yêu cầu, đúng chức năng do mình nhập vào và thao tác dễ dàng với công việc quản lý dữ liệu:

* Được phép chỉnh sửa, xóa những thông tin sai, không phù hợp.
* Theo dõi quá trình mua bán.
* Theo dõi thông tin khách hàng nhập vào khi mua hàng, phải đảm bảo tính an toàn, bảo mật, chính xác.
* Theo dõi, xử lý các đơn đặt hàng và cập nhật các thông tin liên quan đến đơn đặt hàng của khách.
* Có thể xóa tất cả các cơ sở dữ liệu sau một thời gian xác định.

# CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

## 2.1. Nền tảng NodeJs

### 2.1.1. Khái niệm

* Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.
* Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Dù sao thì chúng ta cũng nên biết qua một chút chút lịch sử của thứ mà chúng ta đang học một chút chứ nhỉ? =))
* Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.
* Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.
* Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

### 2.1.2. Các đặc tính của NodeJs

* Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* Chạy rất nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* Không đệm: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.

### 2.1.3. Những điều mà NodeJs có thể làm

* NodeJS có thể tạo nội dung trang động
* NodeJS rất bá đạo, nó có thể: Tạo, mở, đọc, ghi, xóa và đóng tập tin ngay trên máy chủ
* NodeJS nó có thể thu thập được dữ liệu
* NodeJS có thể truy vấn, thêm, sửa, xóa dữ liệu trong nhiều hệ Cơ sở dữ liệu như: MySql, MongoDB, Microsoft SQL Server, PostgreSQL.

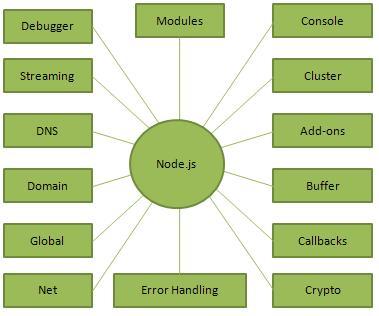
### 2.1.4. Khi nào nên sử dụng NodeJs

Khi chúng ta muốn xây dựng các ứng dụng sau thì nên sử dụng NodeJs:

* Websocket server: Các máy chủ web socket như là Online Chat, Game Server…
* Fast File Upload Client: là các chương trình upload file tốc độ cao.
* Ad Server: Các máy chủ quảng cáo.
* Cloud Services: Các dịch vụ đám mây.
* RESTful API: đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
* Any Real-time Data Application: bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.

### 2.1.5. Các thành phần của NodeJs

NodeJs bao gồm các thành phần quan trọng sau:



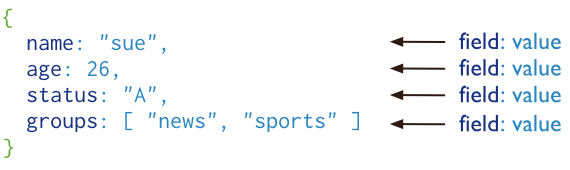
### 2.1.6. Tại sao nên sử dụng NodeJs

* Các ứng dụng Nodejs được viết bằng javascript, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ Javascript, Ryan Dahl: “Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”
* Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.
* Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí. Chúng ta có thể thấy cộng đồng Nodejs lớn như thế nào tại đây, các package đều hoàn toàn free: [https://www.npmjs.com/](https://www.npmjs.com/" \t "_blank)
* Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

## 2.2. MongoDB

### 2.2.1. Khái niệm

MongoDB là một mã nguồn mở và là một tập tài liệu dùng cơ chế NoSQL để truy vấn, nó được viết bởi ngôn ngữ C++. Chính vì được viết bởi C++ nên nó có khả năng tính toán với tốc độ cao chứ không giống như các hệ quản trị CSDL hiện nay. Mỗi một table (bảng dữ liệu) trong SQL sử dụng thì trong MongoDB gọi là collection (tập hợp) Mỗi một record (bản ghi) trong MongoDB được gọi là document (tài liệu). Một bản ghi của MongoDB được lưu trữ dưới dạng document (tài liệu), nó được ghi xuống với cấu trúc field (trường) và value (giá trị). Nó giống như là một đối tượng JSON có dạng như sau:



Điều đó giúp chúng ta có thể dễ dàng ép kiểu sang array (mảng) để lập trình các ứng dụng một cách dễ dàng hơn. Nói một cách dễ hiểu thì mỗi một bản ghi của MongoDB là một mảng dữ liệu riêng biệt bao gồm các cặp key, value khác nhau do đó cách lưu trữ của MongoDB là phi cấu trúc dữ liệu.

### 2.2.2. Cách thức hoạt động



* MongoDB hoạt động dưới một tiến trình ngầm service luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.
* Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng. Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

### 2.2.3. Các khái niệm cơ bản

#### 2.2.3.1. Database

Database chính là tập chứa các collection trong MongoDB. Mỗi database sẽ có một tập file riêng của mình trên file system của hệ thống. Một MongoDB server thường chứa nhiều database trên đó.

#### 2.2.3.2. Collection

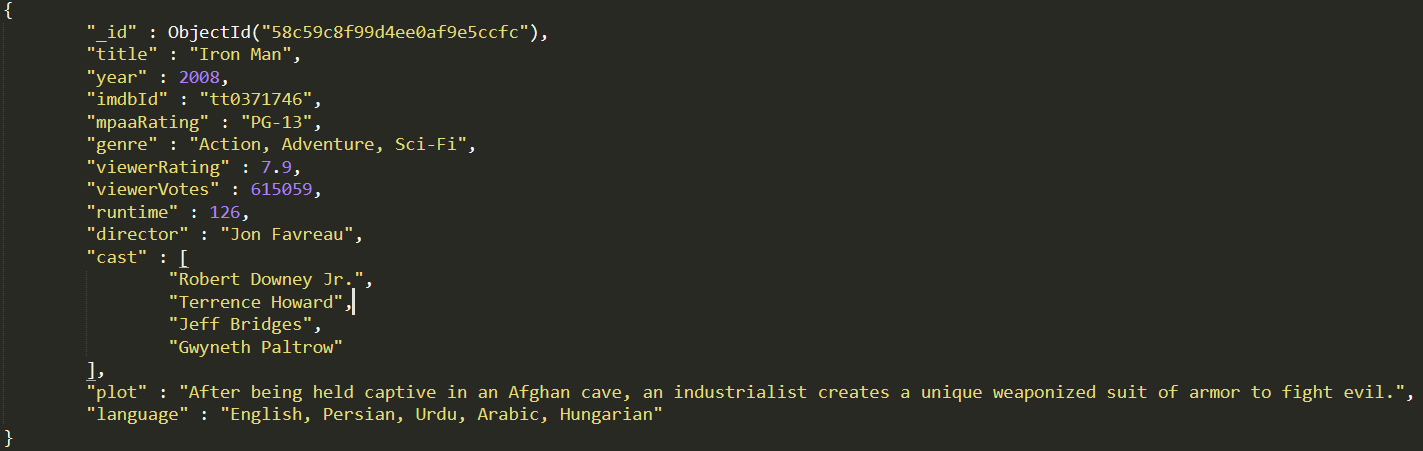
Tương tự như Table trong MySQL, Collection là một tập chứa các MongoDB Document. Một điểm khác so với các RDBMS khác đó chính là Collection không bắt buộc một schema cố định nào cả. Các document trong cùng một collection có thể có nhiều field khác nhau. Nhưng thường thì các document trong một collection sẽ có một số field chính tương đồng nhau và có liên quan với nhau.

#### 2.2.3.4. Document

Tương tự như Table trong MySQL, Collection là một tập chứa các MongoDB Document. Một điểm khác so với các RDBMS khác đó chính là Collection không bắt buộc một schema cố định nào cả. Các document trong cùng một collection có thể có nhiều field khác nhau. Nhưng thường thì các document trong một collection sẽ có một số field chính tương đồng nhau và có liên quan với nhau.

#### 2.2.3.5. JSON

* Các document của MongoDB sử dụng format JSON (JavaScript Object Notation), đây là một chuẩn lưu trữ, trao đổi dữ liệu đơn giản và gọn nhẹ. Với ưu điểm dễ đọc, dễ hiểu, đa phần các ngôn ngữ lập trình phổ biến hiện nay đều hỗ trợ JSON như: C, C++, C#, Java, JavaScript, Perl, Python,….
* Dữ liêu trong JSON được lưu trữ dưới dạng key/value. Một key sẽ tương ứng với 1 value. Value ở đây có thể là một mảng, một chuỗi, một số int, double, mảng hoặc object…
* Ví dụ:



### 2.2.4. Ưu, nhược điểm

#### 2.2.4.1. Ưu điểm

* Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
* Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

#### 2.2.4.2. Nhược điểm

* Không ràng buộc, toàn vẹn nên không ứng dụng được cho các mô hình giao dịch yêu cầu độ chính xác cao.
* Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
* Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
* Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

### 2.2.5. Khi nào nên sử dụng và không nên sử dụng

#### 2.2.5.1. Nên sử dụng khi

* Sử dụng document để lưu trữ dữ liệu, dễ dàng thêm bớt, mở rộng dữ liệu tùy ý.
* Ứng dụng có tính chất Insert nhiều (write-intensive)
* Cần cơ chế Replication và High Availabity.
* Cần cơ chế Sharding (sẽ nói rõ trong phần sau).

#### 2.2.5.2. Không nên sử dụng khi

* Hệ thống cần xử lý nhiều transaction như các giao dịch mua bán, chuyển tiền, ngân hàng. Đối với các trường hợp này thì SQL phù hợp hơn hẳn!
* Cần sử dụng đến các lệnh JOIN.

## 2.3. Android

### 2.3.1. Android, nền tảng di động phổ biến nhất thế giới

Android là nền tảng của hàng trăm triệu thiết bị di động tại hơn 190 quốc gia trên thế giới. Trong các nền tảng di động (mobile platform), đây là nền tảng được cài đặt nhiều nhất và có tốc độ phát triển rất nhanh - mỗi ngày, hàng triệu người dùng bật thiết bị Android của họ lên lần đầu tiên rồi bắt đầu tìm kiếm các ứng dụng, trò chơi cùng những nội dung số khác.

Android cung cấp cho bạn một nền tảng tốt nhất toàn cầu nhằm tạo ra các ứng dụng và trò chơi cho người dùng Android ở mọi nơi, cùng với đó là một thị trường mở để phân phối chúng ngay tức thì.



### 2.3.2. Framework phát triển mạnh mẽ

Android cung cấp mọi thứ bạn cần để xây dựng và trải nghiệm ứng dụng ở mức tốt nhất. Nó cũng cung cấp một mô hình ứng dụng đơn, cho phép bạn triển khai rộng rãi ứng dụng của mình tới hàng trăm triệu người dùng trên một loạt thiết bị - từ điện thoại cho tới máy tính bảng và còn hơn thế nữa.

Android cũng đưa ra những công cụ tạo ứng dụng có giao diện đẹp mắt và tận dụng lợi thế từ khả năng phần cứng có sẵn trên từng thiết bị. Nó cũng tự động thích nghi với giao diện người dùng (user interface - UI) để đạt được sự tối ưu nhất trên từng thiết bị, trong khi vẫn đưa ra nhiều điều khiển (control) như bạn muốn, thông qua giao diện người dùng trên các loại thiết bị khác nhau.

Ví dụ, bạn có thể tạo một ứng dụng hệ nhị phân sao cho ứng dụng này được tối ưu hóa cho cả điện thoại lẫn các dạng máy tính bảng. Bạn có thể khai báo giao diện người dùng của mình một cách gọn nhẹ trong tập các tài nguyên XML, một tập dành cho những thành phần chung đối với tất cả các dạng, trong khi những tập khác dùng cho việc tối ưu hóa theo đặc trưng của điện thoại hoặc máy tính bảng. Khi chạy, Android áp dụng đúng tập tài nguyên dựa trên kích thước màn hình, mật độ, vị trí,…

Nhằm giúp bạn phát triển hiệu quả, các công cụ phát triển Android ([Android Development](http://developer.android.com/tools/index.html) [Tools](http://developer.android.com/tools/index.html) - ADT) cung cấp cho bạn một môi trường phát triển tích hợp (Integrated Development Environment - IDE) đầy đủ cho Java với những tính năng tiên tiến để phát triển, gỡ lỗi (debugging) và đóng gói (packing) các ứng dụng Android. Sử dụng IDE, bạn có thể phát triển ứng dụng trên mọi thiết bị Android có sẵn, hoặc tạo các thiết bị ảo (virtual device) giả lập bất kỳ cấu hình phần cứng nào.

Có 1,5 tỷ lượt tải về mỗi tháng và con số này vẫn tiếp tục tăng. Hãy đưa ứng dụng của bạn ra trước mặt hàng triệu người dùng theo quy mô của Google.

### 2.3.3. Thị trường mở để phân phối các ứng dụng của bạn

Google Play là thị trường hàng đầu cho việc bán và phân phối các ứng dụng Android. Khi phát hành một ứng dụng trên Google Play, bạn đã chạm được vào nền tảng khổng lồ được cài đặt của Android.

Là một thị trường mở, Google Play cho phép bạn kiểm soát cách thức bán các sản phẩm của mình. Bạn có thể phát hành ở mọi thời điểm bạn muốn, với tần suất tùy ý và tới những khách hàng bạn quan tâm. Bạn cũng có thể phân phối rộng khắp trên mọi thị trường và thiết bị hay tập trung vào những phân khúc, thiết bị cụ thể, hoặc theo phạm vi của khả năng phần cứng.

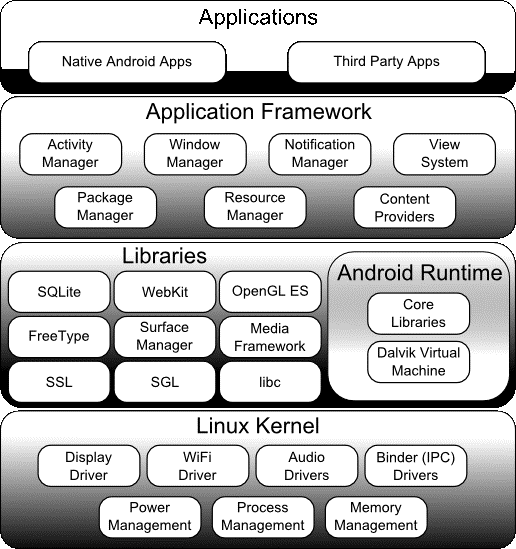
Bạn có thể thu lợi nhuận theo cách hợp lý nhất cho doanh nghiệp của mình - mất phí hoặc miễn phí, với các sản phẩm nhúng trong ứng dụng (in-app) hoặc đăng ký theo dõi (subscription) - đảm bảo đạt được sự cam kết và thu nhập tốt nhất. Bạn cũng có thể kiểm soát hoàn toàn giá cả cho ứng dụng của mình và các sản phẩm nhúng trong ứng dụng, đồng thời có thể thiết lập hoặc thay đổi giá với bất cứ loại tiền tệ được hỗ trợ vào mọi thời điểm.

Ngoài sự phát triển của nền tảng khách hàng, Google Play giúp bạn xây dựng tầm nhìn và cam kết thông qua ứng dụng và thương hiệu của bạn. Khi ứng dụng của bạn dần trở nên phổ biến, Google Play xếp cho chúng vị trí cao hơn trong biểu đồ xếp hạng “đứng đầu” mỗi tuần, tiến hành xếp hạng và đảm bảo những vị trí tốt nhất cho ứng dụng trong kho.

Được cài đặt sẵn trong hàng trăm triệu thiết bị Android trên thế giới, Google Play có thể là một phương tiện phát triển cho doanh nghiệp của bạn.

### 2.3.4. Kiến trúc android

Android được cấu trúc theo hình thức một hàng đợi phần mềm (software stack) bao gồm các ứng dụng (applications), hệ điều hành (operating system), môi trường thực thi (run-time environment), middleware, các dịch vụ (services), các thư viện (libraries). Hình sau minh họa kiến trúc của Android với các tầng (layer) và các thành phần tương ứng trong mỗi tầng trong một Android software stack:



#### 2.3.4.1. Linux Kernel

Cung cấp một mức trừu tượng giữa các thiết bị phần cứng và các tầng trên của stack. Dựa trên Linux phiên bản 2.6, kernel cung cấp đa nhiệm, các dịch vụ hệ thống mức thấp như bộ nhớ, tiến trình, các drivers như wifi, audio, v.v.

#### 2.3.4.2. Android Runtime

* Khi chúng ta thực thi một ứng dụng trong Android Studio, nó sẽ được biên dịch thành một mã bytecode trung gian (định dạng DEX). Khi ứng dụng được tải lên thiết bị, ART sẽ sử dụng một tiến trình gọi là Ahead-of-Time (AOT) để chuyển mã bytecode thành định dạng mã để bộ xử lý có thể hiểu được. Định dạng mã này được gọi là Executable and Linkable Format (ELF). Mỗi lần ứng dụng Android tải lên thiết bị, phiên bản thực thi ELF đã tạo lần đầu sẽ chạy mà không cần phải chuyển sang mã bytecode, do đó, ứng dụng thực thi nhanh hơn và làm tăng tuổi thọ pin của các thiết bị.
* Cách tiếp cận AOT ngược với cách biên dịch Just-in-Time (JIT) trong các phiên bản Andorid cũ hơn. Biên dịch JIT sẽ chuyển mã bytecode sang máy ảo mỗi lần ứng dụng Android được tải lên thiết bị, do đó, sẽ xử lý chậm hơn rất nhiều so với cách biên dịch AOT.

#### 2.3.4.3. Libraries

Android cung cấp một tập các thư viện rất đa dạng dùng trong đồ họa, cơ sở dữ liệu, giao tiếp mạng, v.v. Các thư viện chủ yếu dự trên Java và C/C++. Một số thư viện chủ yếu gồm: android.app, android.database, android.net, android.opengl, v.v. Chúng ta sẽ làm quen với các thư viện Android qua các bài sau.

#### 2.3.4.4. Aplplication Framework

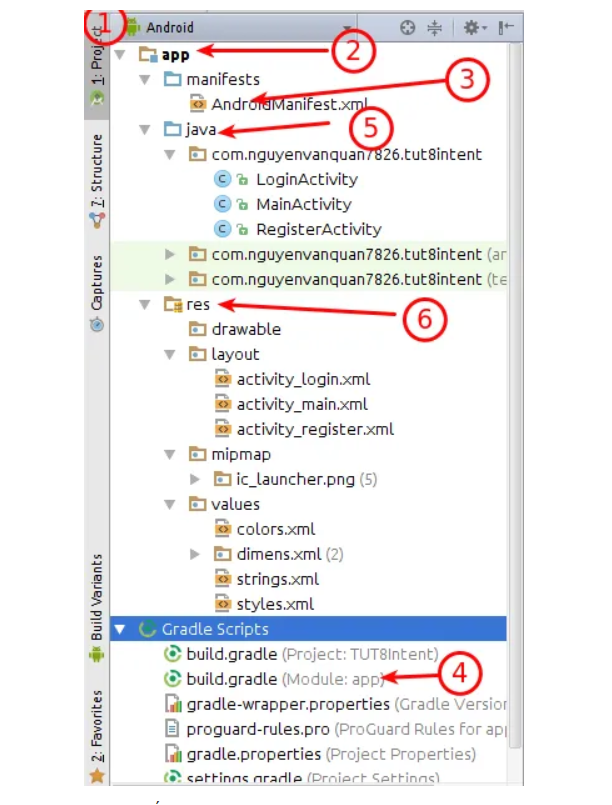
Application Framework là một tập các dịch vụ định hình nên môi trường trong đó các ứng dụng Android thực thi và được quản lý. Một số framework chủ yếu trong Android như Activity Manager, Content Providers, Resource Manager, v.v.

#### 2.3.4.5. Applications

Là tầng cao nhất trong stack gồm các ứng dụng Android và các ứng dụng bên thứ ba do người dùng cài đặt trên thiết bị.

### 2.3.5. Cấu trúc một Project Android

#### 2.3.5.1. Cấu trúc theo chế độ android



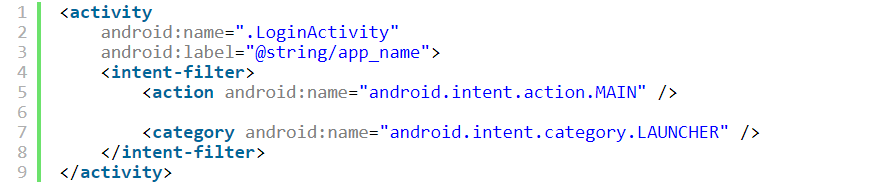
**(1) Chế độ xem Android**

**(2) Module app**: Mặc định khi chúng ta tạo Project mới thì sẽ tự động sinh ra thư mục app này. Nó là một module, Trong một project có thể có rất nhiều module, các module này có thể là các project ứng dụng khác nhau, liên quan đến nhau hoặc là các module thư viện.

**(3) File AndroidManifest.xml**

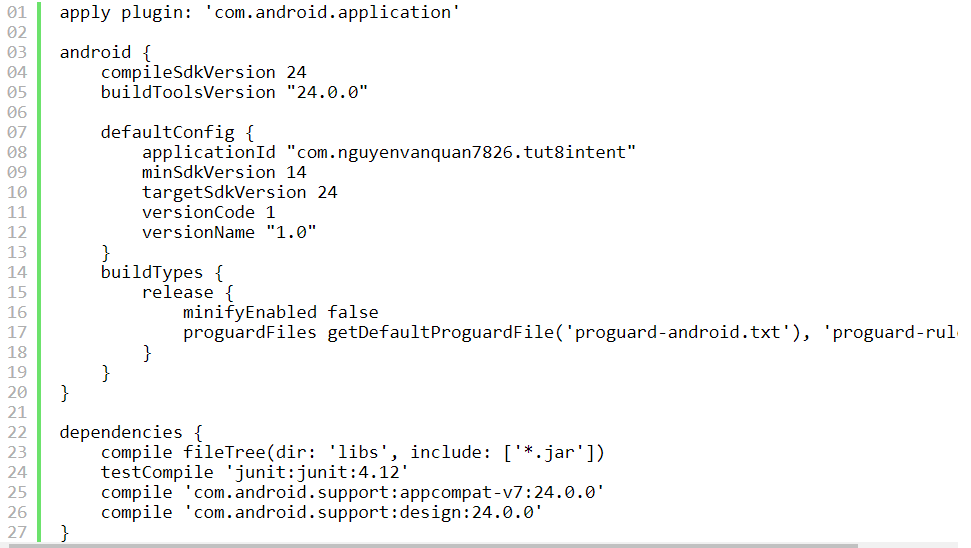


* Đầu tiên là package=”com.nguyenvanquan7826.tut8intent”, đây là package của module. nó cũng chính là id của ứng dụng. Mỗi ứng dụng đưa lên store của google sẽ phải có giá trình package khác nhau, không có 2 ứng dụng nào có thể trùng package.
* 3 dòng uses-permission là 3 dòng mình thêm vào, nó thể hiện các quyền mà ứng dụng yêu cầu từ người dùng. Khi các bạn cài bất kỳ một ứng dụng nào, đâu tiên thường hiện ra phần permisstion yêu cầu bạn đồng ý thì mới cài đó. Các bạn đọc 3 yêu cầu này sẽ nhận ra là yêu cầu lấy danh sách tài khoản trong điện thoại, lấy thông tin tài khoản, và lấy danh bạ.
* Thẻ application bao gồm toàn bộ cài đặt về cách hoạt động của ứng dụng. Cụ thể là có những Activity nào trong ứng dụng, loại activity của mỗi cái, có những service (các dịch vụ chạy ngầm) nào chạy trong ứng dụng. Mỗi thẻ activity sẽ là khai báo cho một activity. Nếu Activity hoặc service nào mà không khai báo ở đây thì sẽ không thể dùng được. Các bạn để ý ở đây chúng ta có 3 Activity là LoginActivity, RegisterActivity, MainActivity.
* Khai báo một Activity. Mặc định khi các bạn tạo Activity như hôm trước mình hướng dẫn thì nó sẽ tự động thêm khai báo vào file AndroidManifest này.



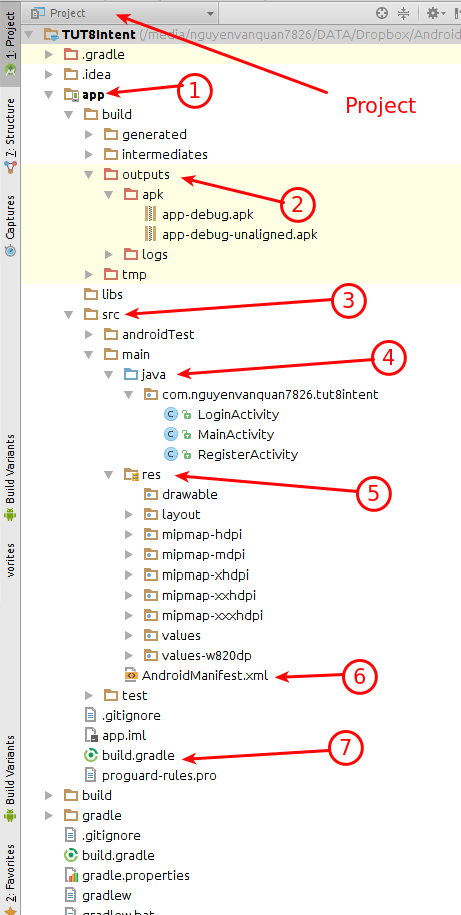
+ **android:name** là tên Activity, ở đây các bạn thấy có dấu chấm (.) đằng trước nó biểu thị là nằm trong package, tức là nằm trong com.nguyenvanquan7826.tut8intent, nếu bạn tạo thêm 1 package con packtemp và đặt một Activity tên là Temp thì sẽ khai báo là android:name=”.packtemp.Temp” tức là nó viết com.nguyenvanquan7826.tut8intent.packtemp.Temp.  
+ **android:label** chính là tiêu đề của Activity, nó sẽ hiển thị trên thanh trên cùng.  
+ **/ intent-filter** là bộ lọc thông báo các điều kiện của Activity, ở đây thông báo nó là activity chính android.intent.action.MAIN và được mở đầu tiên khi mở ứng dụng android.intent.category.LAUNCHER.

**(4) File build.gradle**



* Dòng đầu tiên apply plugin: ‘com.android.application’ thể hiện module này là một ứng dụng. Tức nó là một ứng dụng có thể chạy trên điện thoại. Nếu chúng ta muốn tạo một module có chức năng như một thư viện, có thể dùng cho nhiều Project khác nhau thì nó là apply plugin: ‘com.android.library’.
* Tiếp theo dòng compileSdkVersion và buildToolsVersion lần lượt là phiên bản SDK và bản Tool để tạo ra ứng dụng. Cái này mặc định dùng SDK bao nhiêu thì nó tự động lấy bản mới nhất.
* Dòng applicationId “com.nguyenvanquan7826.tut8intent”, nó sẽ giống trong AndroidManifest, chính là cái id của ứng dụng.
* Tiếp dòng minSdkVersion 14 và targetSdkVersion 24 lần lượt là phiên bản Api Android nhỏ nhất có thể chạy và phiên bản chạy tốt nhất đối với Project này. Khi tạo Project thì nó có cho phép bạn chọn đó. Ở đây phiên bản api nhỏ nhất có thể chạy là api 14 tức Android 4.0 trở lên có thể chay, nếu máy nào dùng phiên bản nhỏ hơn sẽ không thể chạy được ứng dụng này. Bây giờ hầu hết lên trên 4 rồi nên các bạn để min là api 14 (Android 4.0) hoặc api 16 (Android 4.1).
* Dòng versionCode 1 và versionName “1.0” lần lượt là phiên bản code và tên phiên bản ứng dụng. Khi chúng ta cập nhật ứng dụng thì phải sửa phiên bản code lớn hơn phiên bản trước, tên phiên bản phải khác đi thì mới đưa lên store được.

#### 2.3.5.2. Cấu trúc theo chế độ Project



**(1) Module app**

**(2) Phần output**: Phần này là phần chứa file output khi chúng ta debug. Các file apk debug được sinh ra tại đây, các bạn có thể copy cho vào các máy điện thoại khác nhau và chạy luôn. Nhưng lưu ý vì là bản debug thì sẽ không cho lên store được. Mình sẽ có bài hướng dẫn cho lên store riêng.

**(3) Thư mục source:**

**(4) Chứa mã nguồn java**

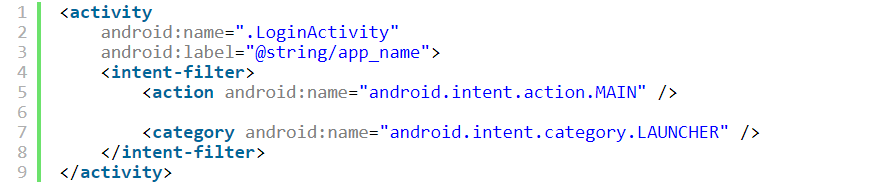
**(5) file res:**

* Thư mục drawable sẽ chứa các file ảnh, file xml cấu hình,… Khi các bạn tạo icon từ công cụ hệ thống Android Studio sẽ sinh ra nhiều loại folder của drawable, tương ứng với các loại kích thước màn hình.
* Thư mục layout chứa các layout rồi nhé.
* Thư mục mipmap, chỉ chứa file ảnh là icon của ứng dụng

**(6) File AndroidManifest.xml**

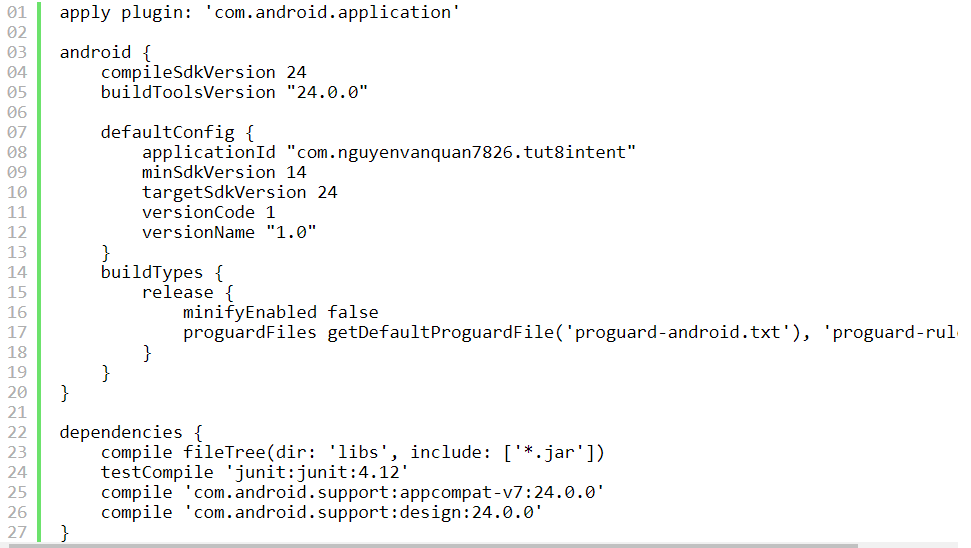


* Đầu tiên là package=”com.nguyenvanquan7826.tut8intent”, đây là package của module. nó cũng chính là id của ứng dụng. Mỗi ứng dụng đưa lên store của google sẽ phải có giá trình package khác nhau, không có 2 ứng dụng nào có thể trùng package.
* 3 dòng uses-permission là 3 dòng mình thêm vào, nó thể hiện các quyền mà ứng dụng yêu cầu từ người dùng. Khi các bạn cài bất kỳ một ứng dụng nào, đâu tiên thường hiện ra phần permisstion yêu cầu bạn đồng ý thì mới cài đó. Các bạn đọc 3 yêu cầu này sẽ nhận ra là yêu cầu lấy danh sách tài khoản trong điện thoại, lấy thông tin tài khoản, và lấy danh bạ.
* Thẻ application bao gồm toàn bộ cài đặt về cách hoạt động của ứng dụng. Cụ thể là có những Activity nào trong ứng dụng, loại activity của mỗi cái, có những service (các dịch vụ chạy ngầm) nào chạy trong ứng dụng. Mỗi thẻ activity sẽ là khai báo cho một activity. Nếu Activity hoặc service nào mà không khai báo ở đây thì sẽ không thể dùng được. Các bạn để ý ở đây chúng ta có 3 Activity là LoginActivity, RegisterActivity, MainActivity.
* Khai báo một Activity. Mặc định khi các bạn tạo Activity như hôm trước mình hướng dẫn thì nó sẽ tự động thêm khai báo vào file AndroidManifest này.



+ **android:name** là tên Activity, ở đây các bạn thấy có dấu chấm (.) đằng trước nó biểu thị là nằm trong package, tức là nằm trong com.nguyenvanquan7826.tut8intent, nếu bạn tạo thêm 1 package con packtemp và đặt một Activity tên là Temp thì sẽ khai báo là android:name=”.packtemp.Temp” tức là nó viết com.nguyenvanquan7826.tut8intent.packtemp.Temp.  
+ **android:label** chính là tiêu đề của Activity, nó sẽ hiển thị trên thanh trên cùng.  
+ **/ intent-filter** là bộ lọc thông báo các điều kiện của Activity, ở đây thông báo nó là activity chính android.intent.action.MAIN và được mở đầu tiên khi mở ứng dụng android.intent.category.LAUNCHER.

**(7) File build.gradle**



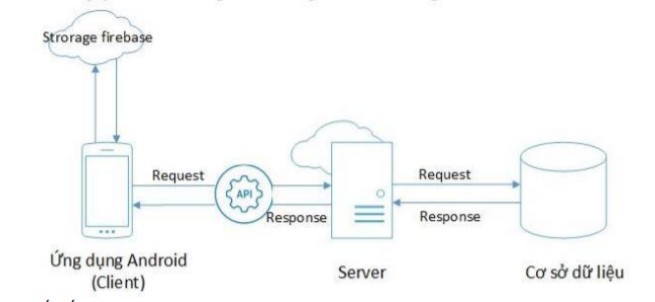
* Dòng đầu tiên apply plugin: ‘com.android.application’ thể hiện module này là một ứng dụng. Tức nó là một ứng dụng có thể chạy trên điện thoại. Nếu chúng ta muốn tạo một module có chức năng như một thư viện, có thể dùng cho nhiều Project khác nhau thì nó là apply plugin: ‘com.android.library’.
* Tiếp theo dòng compileSdkVersion và buildToolsVersion lần lượt là phiên bản SDK và bản Tool để tạo ra ứng dụng. Cái này mặc định dùng SDK bao nhiêu thì nó tự động lấy bản mới nhất.
* Dòng applicationId “com.nguyenvanquan7826.tut8intent”, nó sẽ giống trong AndroidManifest, chính là cái id của ứng dụng.
* Tiếp dòng minSdkVersion 14 và targetSdkVersion 24 lần lượt là phiên bản Api Android nhỏ nhất có thể chạy và phiên bản chạy tốt nhất đối với Project này. Khi tạo Project thì nó có cho phép chúng ta chọn đó. Ở đây phiên bản api nhỏ nhất có thể chạy là api 14 tức Android 4.0 trở lên có thể chạy, nếu máy nào dùng phiên bản nhỏ hơn sẽ không thể chạy được ứng dụng này. Bây giờ hầu hết lên trên 4 rồi nên chúng ta để min là api 14 (Android 4.0) hoặc api 16 (Android 4.1).
* Dòng versionCode 1 và versionName “1.0” lần lượt là phiên bản code và tên phiên bản ứng dụng. Khi chúng ta cập nhật ứng dụng thì phải sửa phiên bản code lớn hơn phiên bản trước, tên phiên bản phải khác đi thì mới đưa lên store được.

## 2.4. Cloudinary

## 2.5. Heroku

# CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

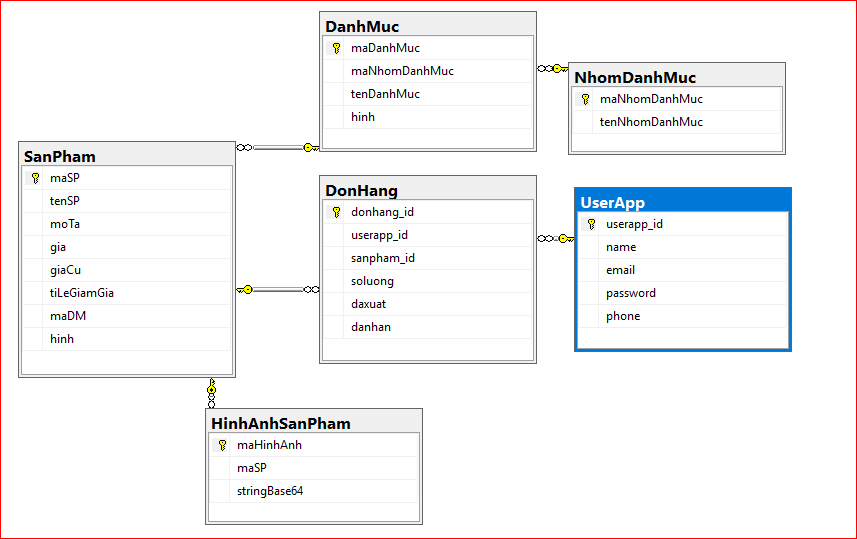
## 3.1. Kiến trúc hệ thống



## 3.2. Thiết kế

### 3.2.1. Cơ sở dữ liệu

#### Sơ đồ logic



#### Danh sách các bảng dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Diễn giải** |
| 1 | UserApp | Người dùng |
| 2 | DanhMuc | Danh mục sản phẩm |
| 3 | NhomDanhMuc | Nhóm danh mục sản phẩm |
| 4 | SanPham | Sản phảm |
| 5 | ĐonHang | Đơn hàng |
| 6 | HinhAnhSanPham | Hình ảnh sản phẩm |

#### Mô tả từng bảng dữ liệu

#### 3.2.1.3.1. Bảng UserApp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | userapp\_id | Int(20) | Khóa chính | Mã user |
| 2 | name | Varchar2(50) | Not null | Tên đăng nhập |
| 3 | email | Varchar2(50) | Not null | email |
| 4 | password | Varchar(50) | Not null | Mật khẩu |
| 5 | phone | Int(20) |  | Số điện thoại |

#### 3.2.1.3.2. Bảng DanhMuc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | maDanhMuc | Int(20) | Khóa chính | Mã danh mục |
| 2 | maNhomDanhMuc | Int(20) | Not null | Mã nhóm danh mục |
| 3 | tenDanhMuc | Varchar(20) | Not null | Tên danh mục |
| 4 | hinh | Varchar2 |  | Hình ảnh |

#### 3.2.1.3.3. Bảng NhomDanhMuc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | maNhomDanhMuc | Int(20) | Khóa chính | Mã nhóm danh mục |
| 2 | tenNhomDanhMuc | Varchar2(50) | Not null | Tên nhóm danh mục |

#### 3.2.1.3.4. Bảng SanPham

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | maSanPham | Int(20) | Khóa chính | Mã sản phẩm |
| 2 | tenSanPham | Varchar2(50) | Not null | Tên sản phẩm |
| 3 | moTa | Varchar2 | Not null | Mô tả |
| 4 | gia | int |  | giá |
| 5 | giaCu | int |  | Giá cũ |
| 6 | tiLeGiamGia | int |  | Tỉ lệ giảm giá |
| 7 | maDM | int |  | Mã danh mục |
| 8 | hinh | Varchar2 |  | Hình ảnh |

#### 3.2.1.3.5. Bảng DonHang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | donHang\_id | Int | Khóa chính | Mã hiệu sản phẩm |
| 2 | Userapp\_id | Varchar(255) | Not null | Tên hiệu sản phẩm |
| 3 | sanPham\_id | Varchar(100) | Not null | Tình trạng |
| 4 | soLuong | int |  | Số lượng |
| 5 | daXuat | int |  | Đã xuất |
| 6 | daNhan | int |  | Đã nhận |

#### 3.2.1.3.6. Bảng HinhAnhSanPham

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | maHinhAnh | Int | Khóa chính | Mã hình ảnh |
| 2 | maSanPham | Int | Not null | Mã sản phẩm |

### 3.2.2. Mô hình phân rã chức năng

**Quản lý shop bán hàng SHTN Shop**

*Đăng nhập, đăng ký*

*Quản lý sản phẩm*

*Quản lý đặt hàng*

*Quản lý mua hàng*

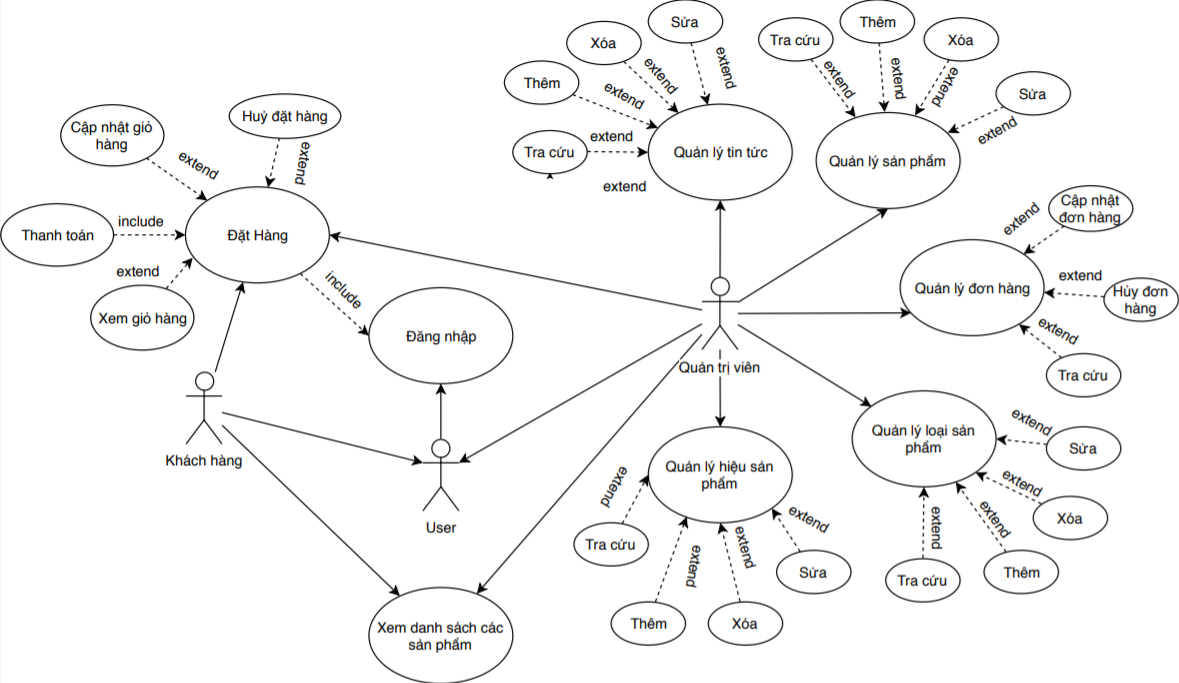
*Quản lý Loại sản phẩm*

*Quản lý Hiệu sản phẩm*

*Quản lý đơn hàng*

### 3.2.3. Use case

#### 3.2.3.1 Sơ đồ Use case



#### 3.2.3.2. Đặc tả use case và Activity Diagram

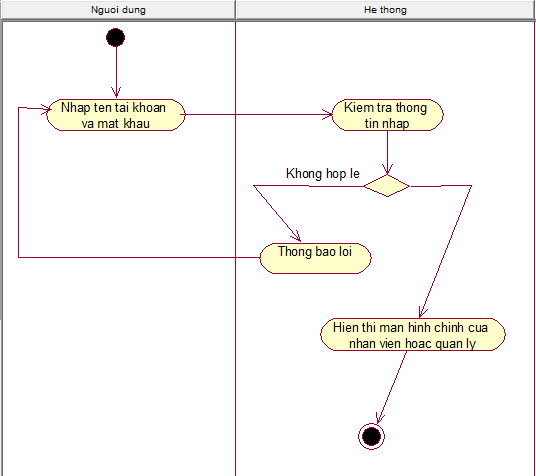
#### 3.2.3.2.1. Quản lý đăng nhập, đăng ký

**- Tên Use case:** Đăng nhập

**-** **Mô tả**: Use case cho phép thực hiện chức năng đăng nhập, đăng ký. Người dùng nhập tên tài khoản và mật khẩu để có thể vào sử dụng hệ thống

**- Activity Diagram:**

* *Đăng nhập:*

****

**-** **Mô tả** **dòng sự kiện**:

* **Dòng sự kiện chính :**
* Hệ thống yêu cầu nhập username và password.
* Người dùng nhập tên tài khoản và mật khẩu
* Hệ thống xác nhận thông tin đăng nhập để cho người dùng đăng nhập vào hệ thống.
* **Các dòng sự kiện khác:**

Nếu actor nhập sai username và password, xuất hiện thông báo lỗi. Actor có thể chọn bắt đầu lại Dòng sự kiện chính(nhập thông tin lại từ đầu)hoặc không đăng nhập nữa.

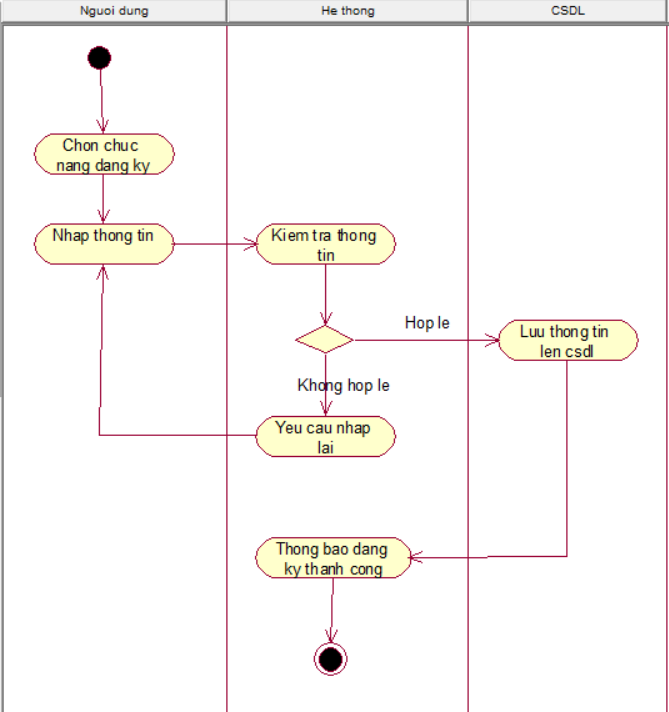
**- Các yêu cầu đặc biệt**: không có

**- Trạng thái hệ thống khi bắt đầu thực hiện usecase**: không có

**- Trạng thái hệ thống sau khi thực hiện usecase**: Nếu login thành công, Actor được đăng nhập vào hệ thống. Nếu không thì hệ thống vẫn giữ nguyên trạng thái cũ

**-** **Điểm mở rộng:** không có

* *Đăng ký:*



**- Mô tả dòng sự kiện:**

* Người dùng chọn chức năng đăng ký
* Người dùng nhập thông tin
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì lưu thông tin sản phẩm lên cơ sở dữ liệu, và thông báo đăng ký thành công
* Nếu không hợp lệ thì hệ thống yêu cầu người dùng nhập lại thông tin

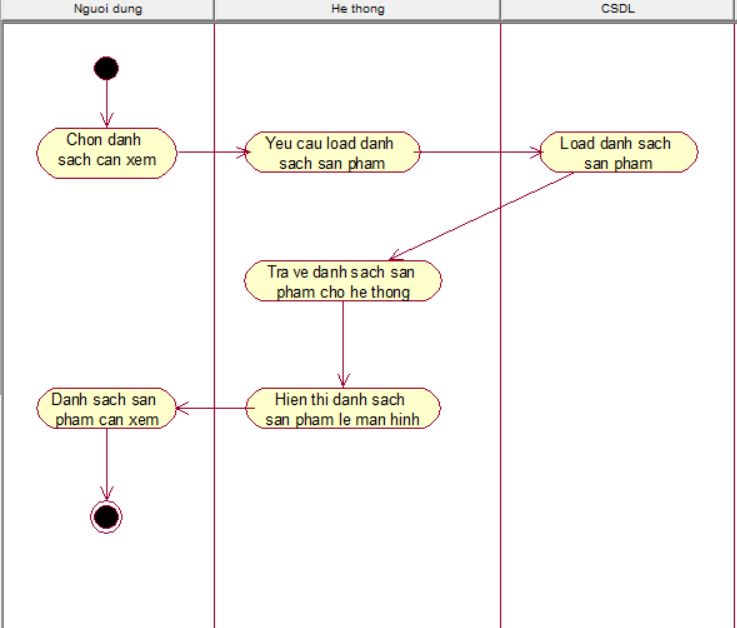
**- Yêu cầu tiên quyết:** Không có

#### 3.2.3.2.2. Xem danh sách sản phẩm

**-** **Tên Use case:** Xem danh sách sản phẩm

**-** **Mô tả:** Cho phép người dùng xem danh sách sản phẩm

**- Activity Diagram:**



**- Mô tả dòng sự kiện:**

* Người dùng chọn danh sách sản phẩm cần xem
* Hệ thống yêu cầu load danh sách sản phẩm
* CSDL trả về danh sách sản phẩm để hệ thống hiển thị lên cho người dùng

**- Yêu cầu tiên quyết:** Không có

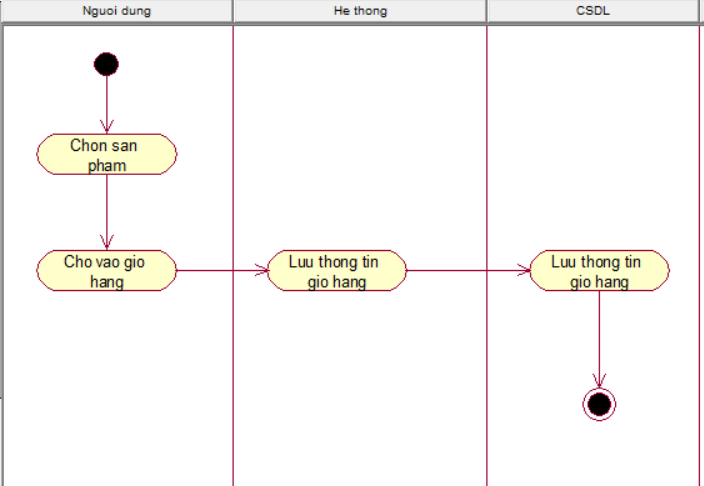
#### 3.2.3.2.3. Đặt hàng

**-** **Tên Use case:** Đặt hàng

**-** **Mô tả:** Cho phép người dùng chọn sản phẩm vào giỏ hàng, xem giỏ hàng, đặt hàng

**- Activity Diagram:**

* *Chọn sản phẩm vào giỏ hàng:*

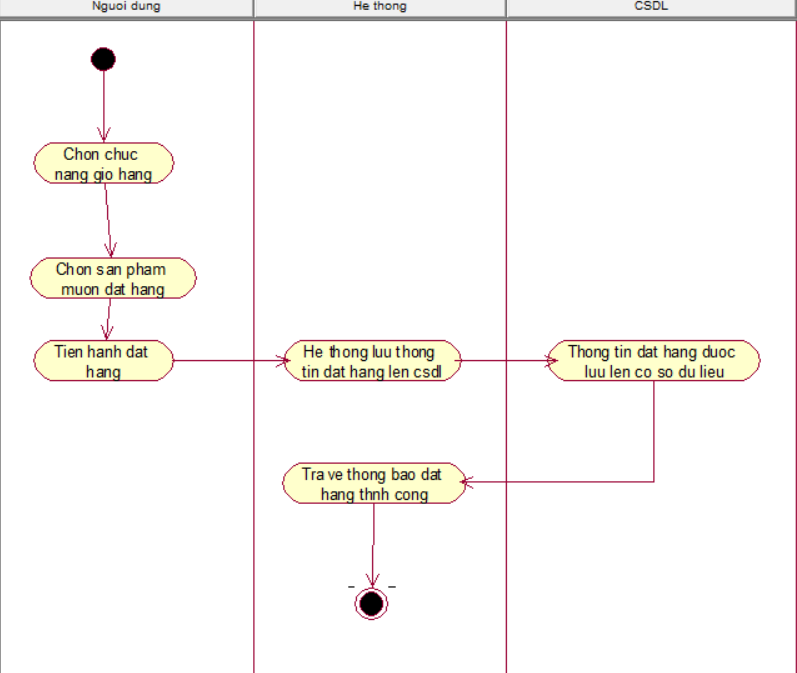


**- Mô tả dòng sự kiện:**

* Người dùng chọn sản phẩm cần cho vào giỏ hàng
* Hệ thống lưu thông tin giỏ hàng lên csdl.

**- Yêu cầu tiên quyết: Không có.**

* *Đặt hàng*



**- Mô tả dòng sự kiện:**

* Người dùng chọn chức năng giỏ hàng
* Người dùng chọn sản phẩm muốn đặt hàng
* Tiến hành đặt hàng
* Hệ thống lưu thông tin đặt hàng lên csdl
* Thông báo đặt hàng thành công

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

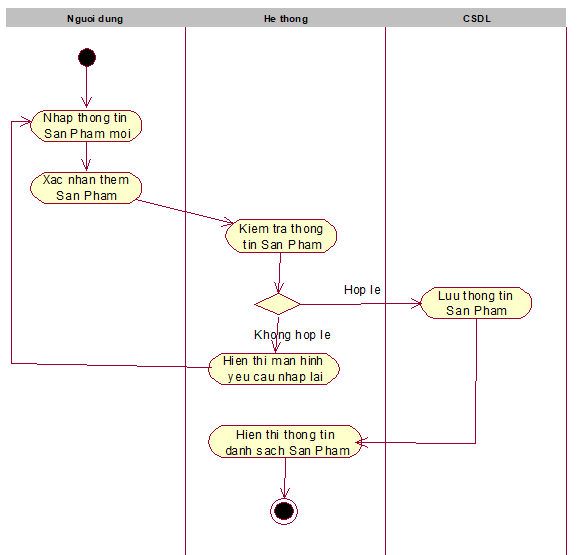
#### 3.2.3.2.4. Quản lý sản phẩm

**-** **Tên Use case:** Thêm xóa, sửa, tra cứu sản phẩm

**-** **Mô tả:** Cho phép người dùng thêm sản phẩm, xóa sản phẩm, sửa thông tin sản phẩm, tra cứu sản phẩm.

**- Activity Diagram:**

*Thêm Sản phẩm:*

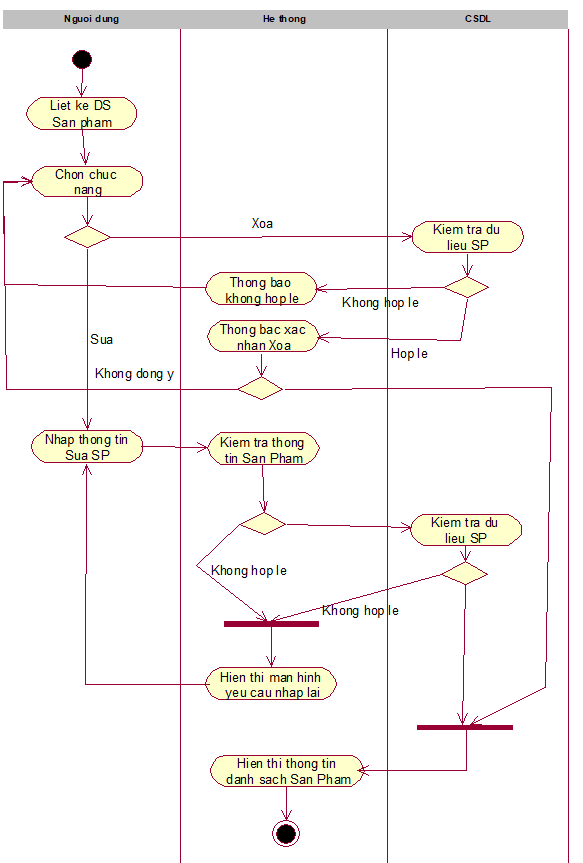


**- Mô tả dòng sự kiện:**

* Hệ thống yêu cầu nhập thông tin sản phẩm mới.
* Người dùng nhập thông tin sản phẩm và xác nhận thêm sản phẩm mới.
* Hệ thống kiểm tra thông tin sản phẩm
* Nếu hợp lệ thì lưu thông tin sản phẩm lên cơ sở dữ liệu, và hiển thị thông tin danh sách sản phẩm
* Nếu không hợp lệ thì hệ thống yêu cầu người dùng nhập lại thông tin sản phẩm

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Xoá sửa Sản phẩm:*

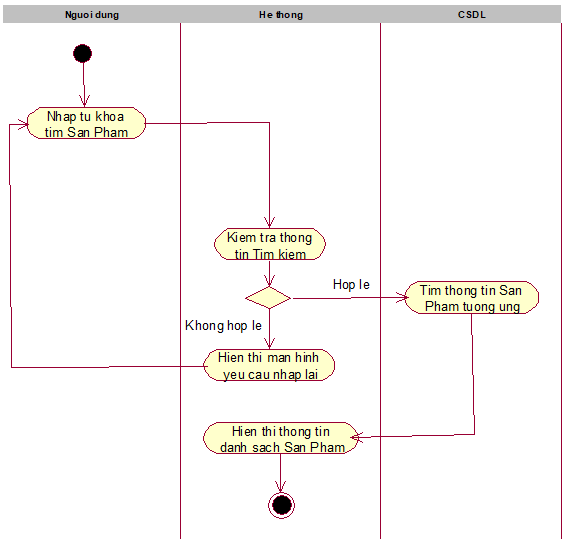


**- Mô tả dòng sự kiện:**

* Người dùng chọn chức năng liệt kê danh sách sản phẩm
* Người dùng chọn chức năng muốn thực hiện
* Nếu chọn xóa thì hệ thống sẽ kiểm tra dữ liệu
* Nếu hợp lệ thì xóa sản phẩm khỏi cơ sở dữ liệu và thông báo xóa thành công
* Nếu không hợp lệ thì thông báo xóa không thành công và yêu cầu chọn lại chức năng.
* Nếu chọn sửa thì hệ thống yêu cầu nhập thông tin cần sửa
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì update lại thông tin sản phẩm và thông báo sửa thành công
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Tra cứu Sản Phẩm:*



**- Mô tả dòng sự kiện:**

* Người dùng nhập thông tin cần tra cứu
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì tiến hành tìm kiếm thông tin tương ứng và hiển thị thông tin ra màn hình
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin.

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

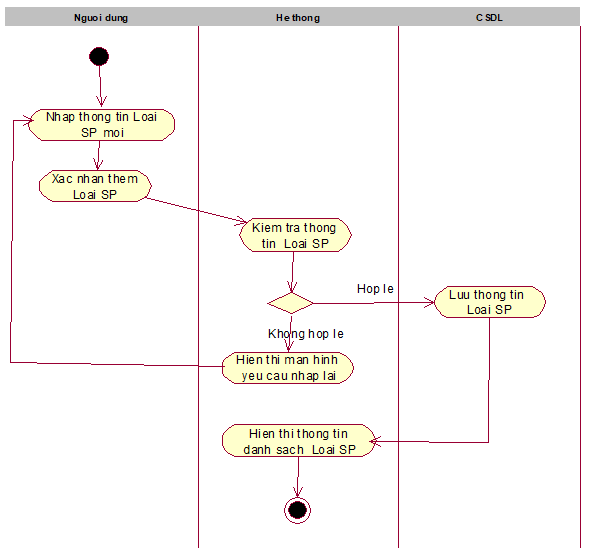
#### 3.2.3.2.5. Quản lý loại sản phẩm

**- Tên Use case:** Thêm, xóa, sửa, tra cứu loại sản phẩm

**- Mô tả:** Cho phép người dùng thêm loại sản phẩm, xóa loại sản phẩm, sửa thông tin loại sản phẩm, tra cứu loại sản phẩm.

**- Activity Diagram:**

*Thêm loại Sản Phẩm:*

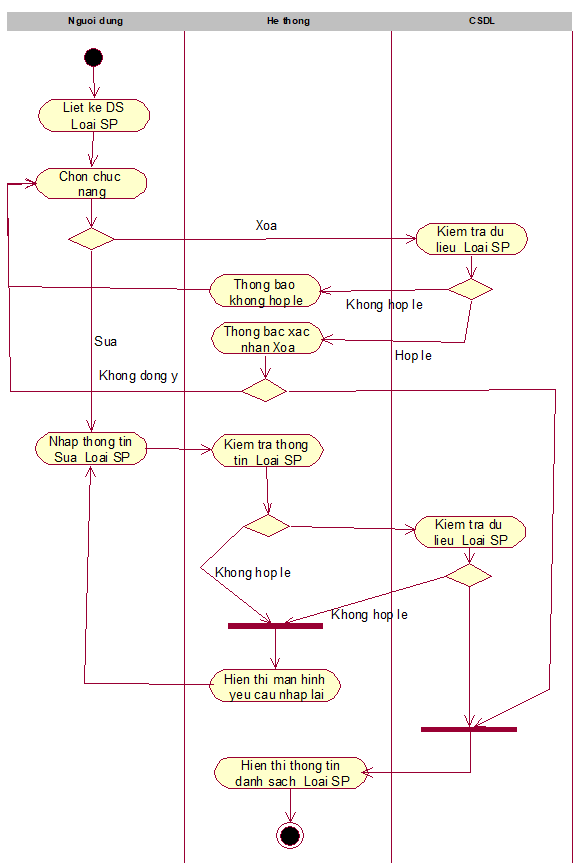


**- Mô tả dòng sự kiện:**

* Hệ thống yêu cầu nhập thông tin loại sản phẩm mới.
* Người dùng nhập thông tin loại sản phẩm và xác nhận thêm loại sản phẩm mới.
* Hệ thống kiểm tra thông tin loại sản phẩm
* Nếu hợp lệ thì lưu thông tin loại sản phẩm lên cơ sở dữ liệu, và hiển thị thông tin danh sách loại sản phẩm
* Nếu không hợp lệ thì hệ thống yêu cầu người dùng nhập lại thông tin loại sản phẩm

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Xóa,Sửa loại sản phẩm:*

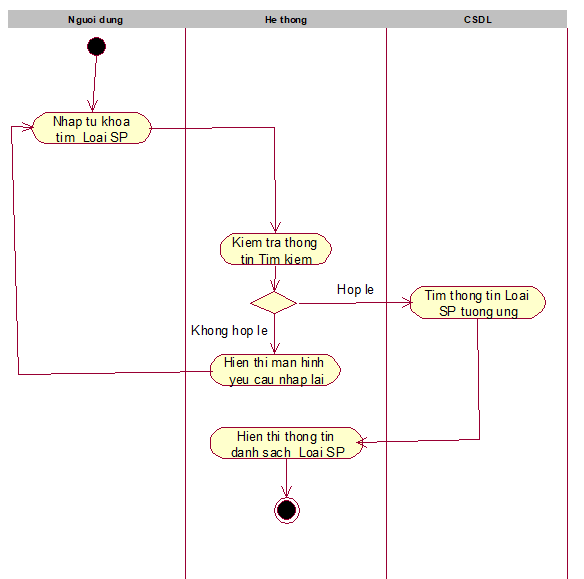


**- Mô tả dòng sự kiện:**

* Người dùng chọn chức năng liệt kê danh sách loại sản phẩm
* Người dùng chọn chức năng muốn thực hiện
* Nếu chọn xóa thì hệ thống sẽ kiểm tra dữ liệu
* Nếu hợp lệ thì xóa loại sản phẩm khỏi cơ sở dữ liệu và thông báo xóa thành công
* Nếu không hợp lệ thì thông báo xóa không thành công và yêu cầu chọn lại chức năng.
* Nếu chọn sửa thì hệ thống yêu cầu nhập thông tin cần sửa
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì update lại thông tin loại sản phẩm và thông báo sửa thành công
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Tra cứu loại sản phẩm:*



**- Mô tả dòng sự kiện:**

* Người dùng nhập thông tin cần tra cứu
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì tiến hành tìm kiếm thông tin tương ứng và hiển thị thông tin ra màn hình
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin.

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

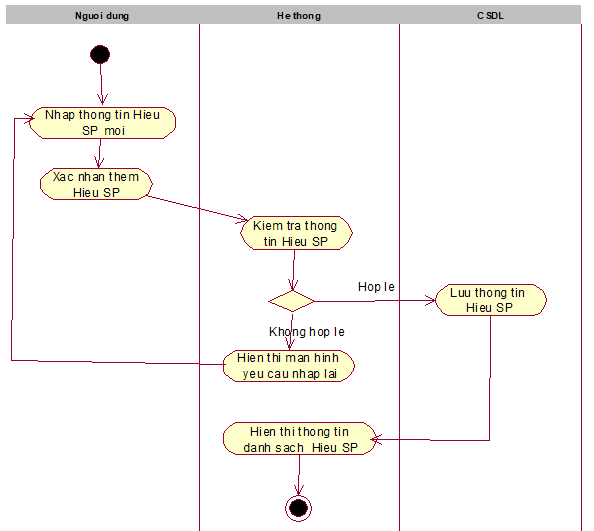
#### 3.2.3.2.6. Quản lý Hiệu sản phẩm

**- Tên Use case:** Thêm, xóa, sửa, tra cứu hiệu sản phẩm

**- Mô tả:** Cho phép người dùng thêm hiệu sản phẩm, xóa hiệu sản phẩm, sửa thông tin hiệu sản phẩm, tra cứu hiệu sản phẩm.

**- Activity Diagram:**

*Thêm Hiệu sản phẩm:*



**- Mô tả dòng sự kiện:**

* Hệ thống yêu cầu nhập thông tin hiệu sản phẩm mới.
* Người dùng nhập thông tin hiệu sản phẩm và xác nhận thêm hiệu sản phẩm mới.
* Hệ thống kiểm tra thông tin hiệu sản phẩm
* Nếu hợp lệ thì lưu thông tin hiệu sản phẩm lên cơ sở dữ liệu, và hiển thị thông tin danh sách hiệu sản phẩm
* Nếu không hợp lệ thì hệ thống yêu cầu người dùng nhập lại thông tin hiệu sản phẩm

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Xóa, sửa Hiệu sản phẩm:*

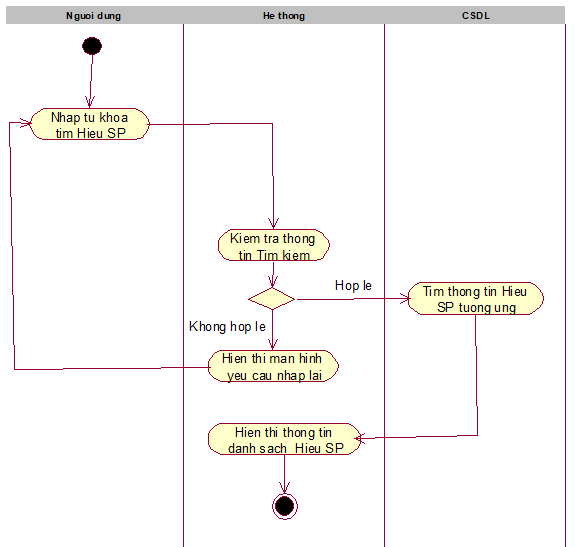


**- Mô tả dòng sự kiện:**

* Người dùng chọn chức năng liệt kê danh sách hiệu sản phẩm
* Người dùng chọn chức năng muốn thực hiện
* Nếu chọn xóa thì hệ thống sẽ kiểm tra dữ liệu
* Nếu hợp lệ thì xóa hiệu sản phẩm khỏi cơ sở dữ liệu và thông báo xóa thành công
* Nếu không hợp lệ thì thông báo xóa không thành công và yêu cầu chọn lại chức năng.
* Nếu chọn sửa thì hệ thống yêu cầu nhập thông tin cần sửa
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì update lại thông tin hiệu sản phẩm và thông báo sửa thành công
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Tra cứu Hiệu sản phẩm:*



**- Mô tả dòng sự kiện:**

* Người dùng nhập thông tin cần tra cứu
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì tiến hành tìm kiếm thông tin tương ứng và hiển thị thông tin ra màn hình
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin.

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

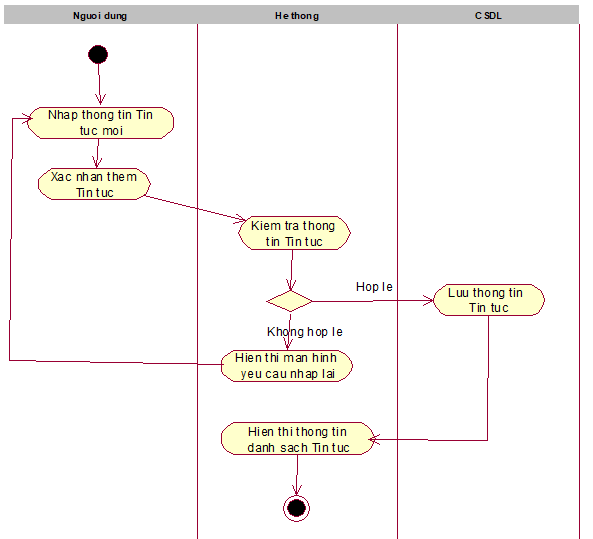
#### 3.2.3.2.7. Quản lý Tin tức

**- Tên Use case:** Thêm, xóa, sửa, tra cứu tin tức

**- Mô tả:** Cho phép người dùng thêm tin tức, xóa tin tức, sửa thông tin tin tức, tra cứu tin tức.

**- Activity Diagram:**

*Thêm Tin tức :*

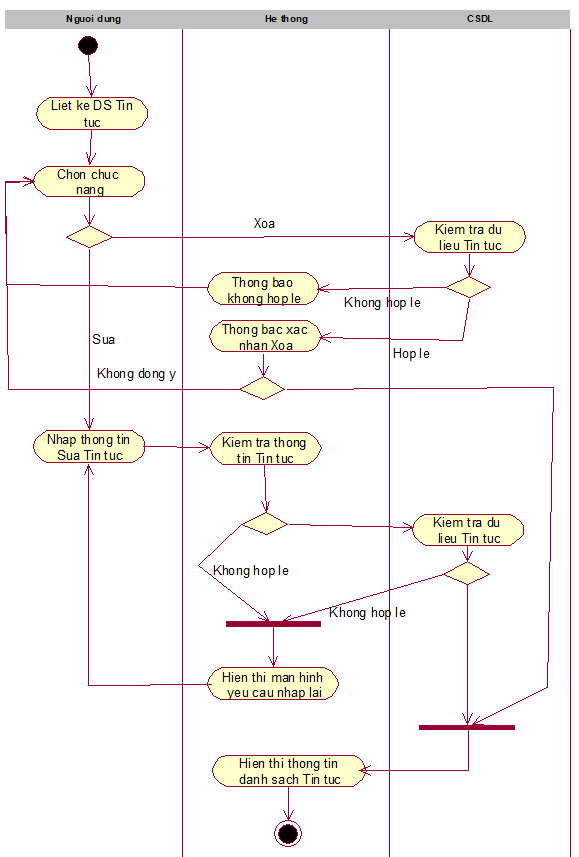


**- Mô tả dòng sự kiện:**

* Hệ thống yêu cầu nhập thông tin tin tức.
* Người dùng nhập thông tin tin tức và xác nhận thêm tin tức mới.
* Hệ thống kiểm tra thông tin tin tức
* Nếu hợp lệ thì lưu thông tin tin tức lên cơ sở dữ liệu, và hiển thị thông tin danh sách tin tức
* Nếu không hợp lệ thì hệ thống yêu cầu người dùng nhập lại thông tin tin tức

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Xóa, sửa Tin tức:*

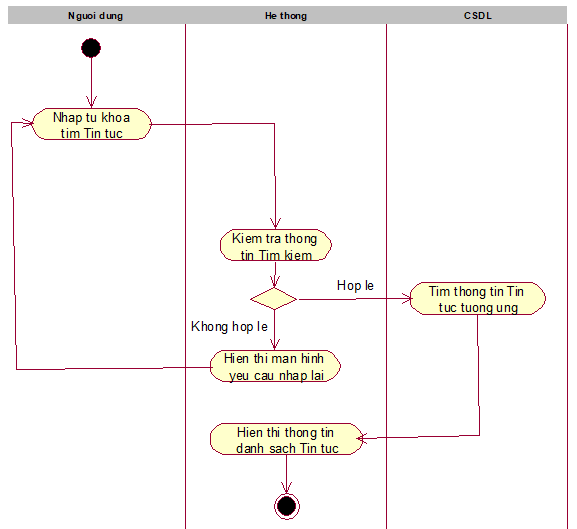


**- Mô tả dòng sự kiện:**

* Người dùng chọn chức năng liệt kê danh sách tin tức
* Người dùng chọn chức năng muốn thực hiện
* Nếu chọn xóa thì hệ thống sẽ kiểm tra dữ liệu
* Nếu hợp lệ thì xóa tin tức khỏi cơ sở dữ liệu và thông báo xóa thành công
* Nếu không hợp lệ thì thông báo xóa không thành công và yêu cầu chọn lại chức năng.
* Nếu chọn sửa thì hệ thống yêu cầu nhập thông tin cần sửa
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì update lại thông tin tin tức và thông báo sửa thành công
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

*Tra cứu Tin tức:*



**- Mô tả dòng sự kiện:**

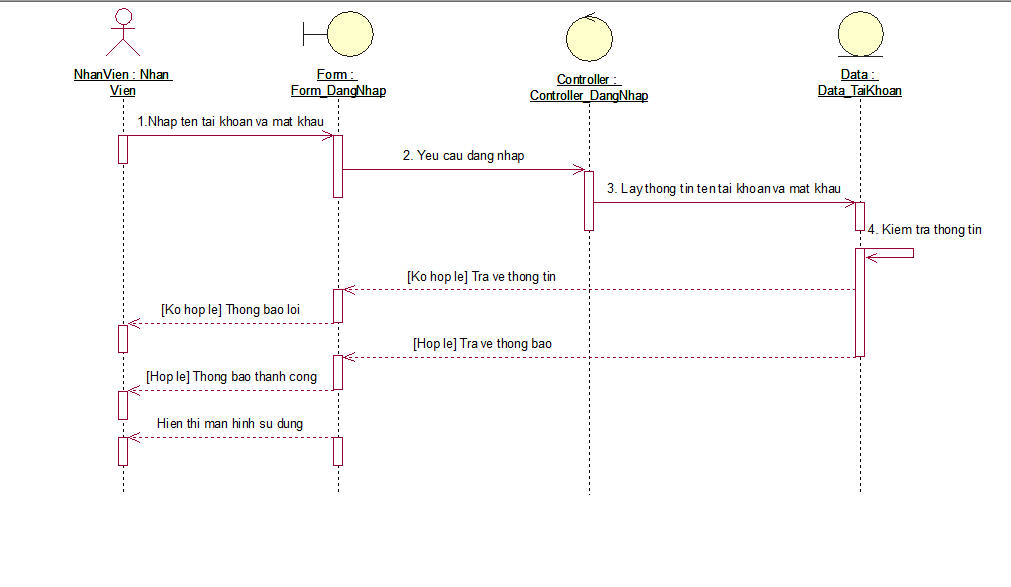
* Người dùng nhập thông tin cần tra cứu
* Hệ thống kiểm tra thông tin
* Nếu hợp lệ thì tiến hành tìm kiếm thông tin tương ứng và hiển thị thông tin ra màn hình
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin.

**- Yêu cầu tiên quyết:** Người dùng đã đăng nhập.

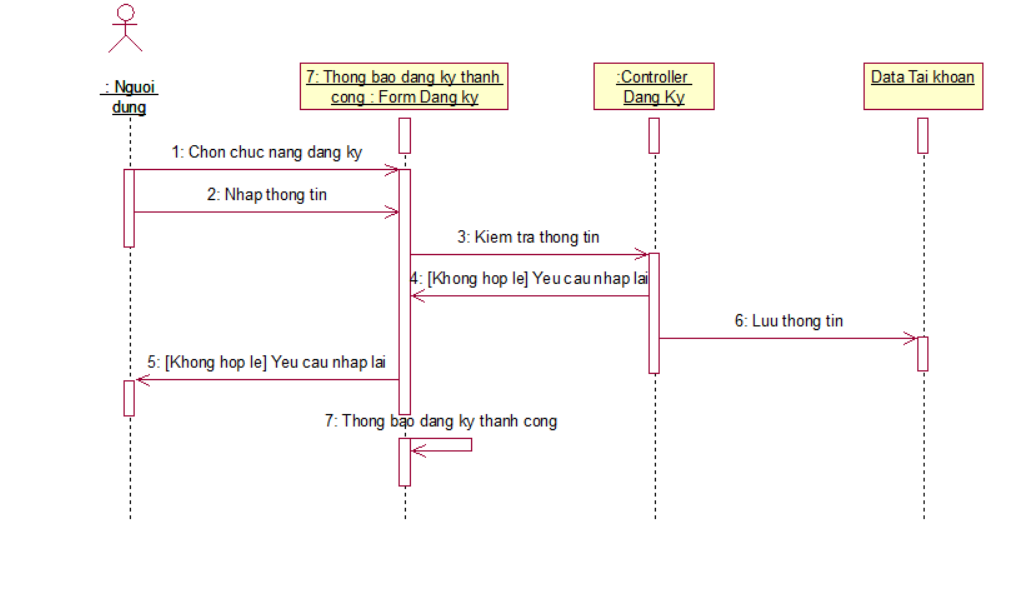
### 3.2.4. Sequence Diagram

#### 3.2.4.1. Đăng nhập, đăng ký

* *Đăng nhập*

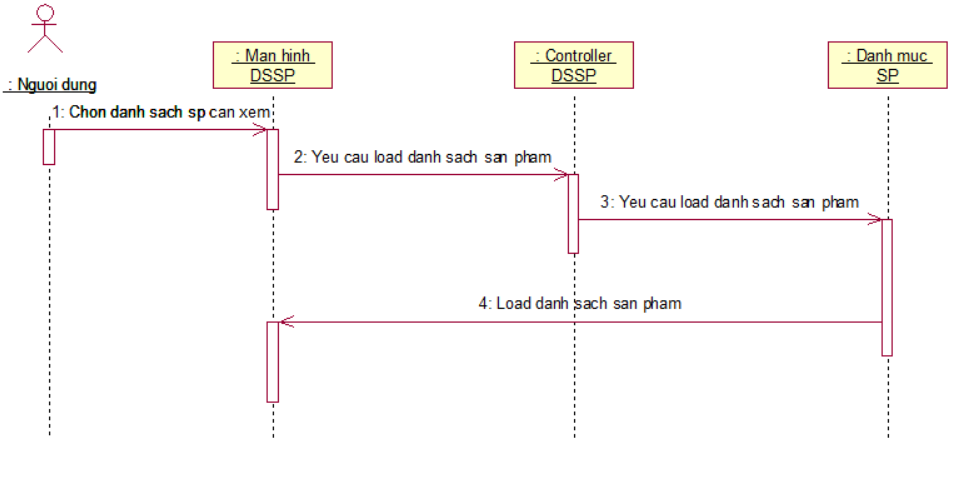


* Các thành phần trong hệ thống trong chức năng đăng nhập: Form\_DangNhap, Controller\_DangNhap, Data\_TaiKhoan.
* Khái quát các bước trong chức năng đăng nhập theo sơ đồ:
* Người dùng yêu cầu giao diện đăng nhập, nhập các thông tin tài khoản.
* Form\_DangNhap gửi yêu cầu đăng nhập lên Controller\_DangNhap.
* Controller\_DangNhap lấy thông tin tên đăng nhập và mật khẩu gửi lên Data\_TaiKhoan.
* Data\_TaiKhoan kiểm tra thông tin.
* Nếu hợp lệ thì trả về thông báo đăng nhập thành công và hiển thị màn hình sử dụng.
* Nếu không hợp lệ thì trả về thông báo đăng nhập không thành công.
* *Đăng ký*



* Các thành phần trong hệ thống trong chức năng đăng ký: Form Dang ky, Controller\_Dang Ky, Data\_TaiKhoan.
* Khái quát các bước trong chức năng đăng ký theo sơ đồ:
* Người dùng yêu cầu giao diện đăng ký, nhập các thông tin.
* Form\_Dang Ky gửi yêu cầu kiểm tra lên Controller\_Dang Ky.
* Controller\_Dang Ky kiểm tra thông tin
* Nếu hợp lệ thì trả về thông báo đăng ký thành công và hiển thị màn hình sử dụng.
* Nếu không hợp lệ thì yêu cầu nhập lại thông tin.

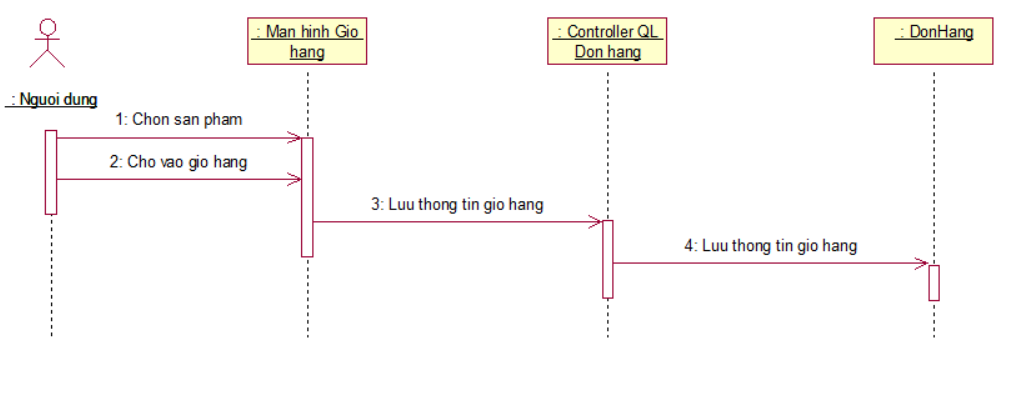
#### 3.2.4.2.2. Xem danh sách sản phẩm



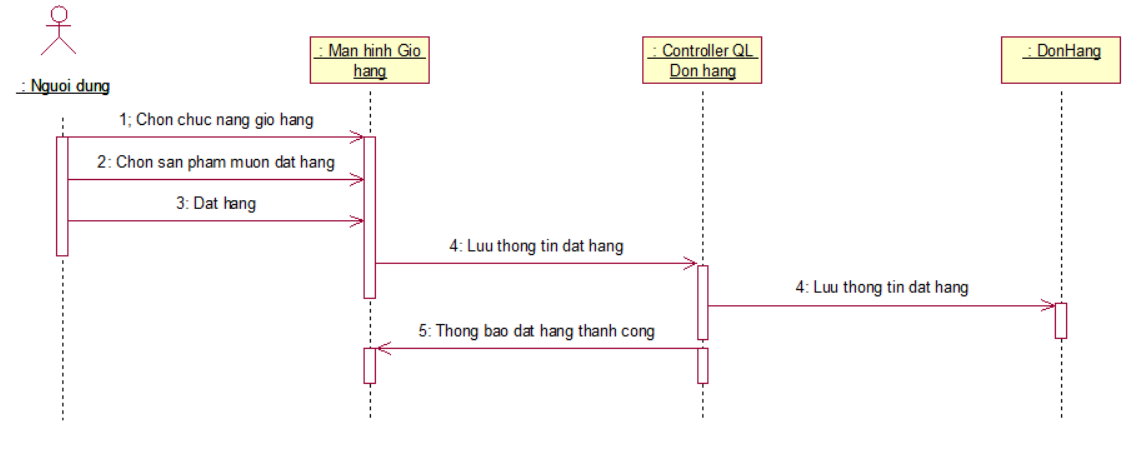
* Các thành phần trong hệ thống trong chức năng đăng ký: Man hinh DSSP, Controller DSSP, Cơ sở dữ liệu Danh muc SP.
* Khái quát các bước trong chức năng đăng ký theo sơ đồ:
* Người dùng chọn danh sách sản phẩm cần xem
* Màn hình danh sách sản phẩm yêu cầu load thông tin danh sách sản phẩm
* Cơ sở dữ liệu danh mục sản phẩm load thông tin danh mục lên màn hình danh sách sản phẩm.

#### 3.2.4.3. Đặt hàng

* *Chọn sản phẩm vào giỏ hàng*



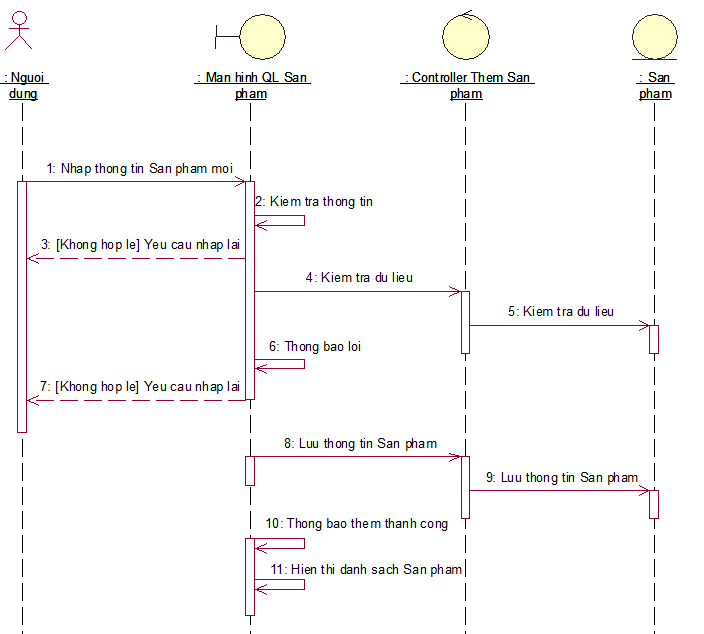
* Các thành phần trong hệ thống trong chức năng đăng ký: Man hinh Gio hang, Controller QL Don hang, Cơ sở dữ liệu Don hang.
* Khái quát các bước trong chức năng đăng ký theo sơ đồ:
* Người dùng chọn sản phẩm để cho vào giỏ hàng
* Controller QL Don hang lưu thông tin giỏ hàng lên cơ sở dữ liệu DonHang
* *Đặt hàng*



* Các thành phần trong hệ thống trong chức năng đăng ký: Man hinh Gio hang, Controller QL Don hang, Cơ sở dữ liệu Don hang.
* Khái quát các bước trong chức năng đăng ký theo sơ đồ:
* Người dùng chọn chức năng giỏ hàng, chọn sản phẩm muốn đặt hàng, tiến hành đặt hàng.
* Hệ thống lưu thông tin đặt hàng lên CSDL.
* Thông báo đặt hàng thành công.

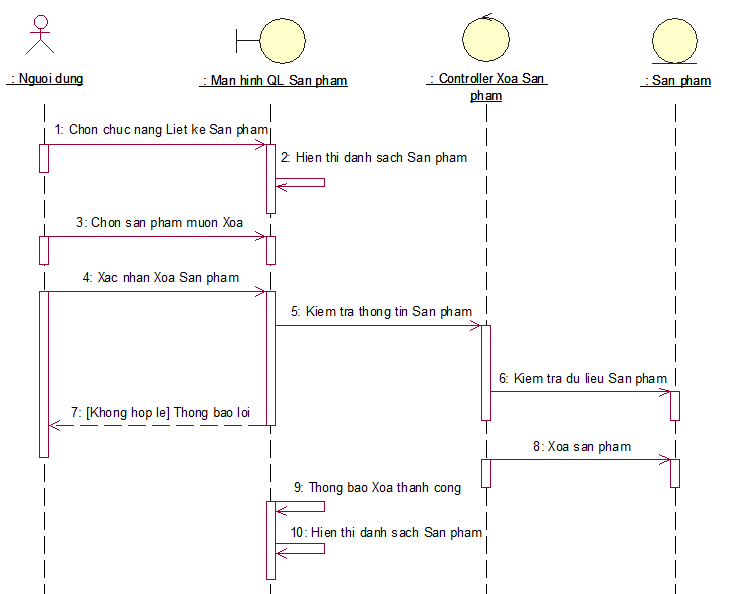
#### 3.2.4.2. Quản lý Sản Phẩm

*Thêm Sản Phẩm:*



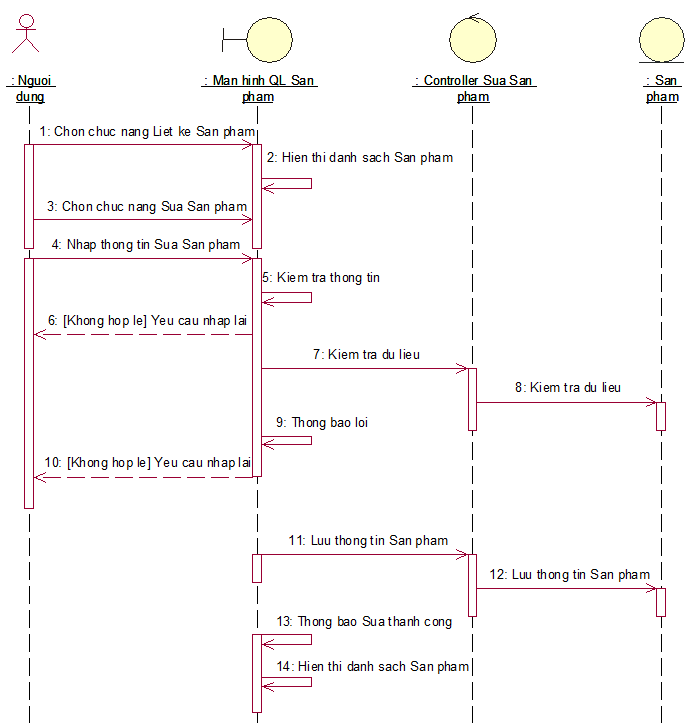
* Các thành phần trong hệ thống trong chức năng thêm sản phẩm: Màn hình QL San Pham, Controller Them San pham, Cơ sở dữ liệu San Pham.
* Khái quát các bước trong chức năng thêm sản phẩm theo sơ đồ:
* Người dùng nhập thông tin sản phẩm cần thêm.
* Màn hình QL San Pham kiểm tra thông tin sản phẩm.
* Nếu không hợp lệ yêu cầu nhập lại.
* Nếu hợp lệ thì bắt đầu kiểm tra dữ liệu
* Nếu dữ liệu không đúng thì yêu cầu nhập lại.
* Nếu dữ liệu hợp lệ thì lưu thông tin sản phẩm lên cơ sở dữ liệu và thông báo thêm sản phẩm thành công, hiển thị thông tin sản phẩm.

*Xoá Sản phẩm:*



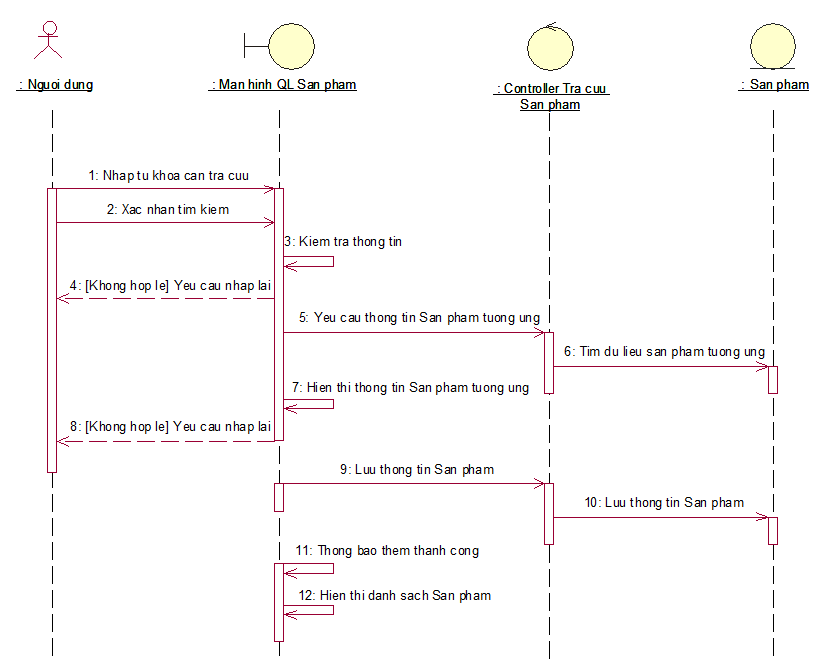
* Các thành phần trong hệ thống trong chức năng xóa sản phẩm: Màn hình QL San Pham, Controller Xoa San pham, Cơ sở dữ liệu San Pham.
* Khái quát các bước trong chức năng xóa sản phẩm theo sơ đồ:
* Người dùng chọn chức năng liệt kê sản phẩm, hiển thị danh sách sản phẩm lên màn hình. Người dùng chọn sản phẩm muốn xóa, xác nhận xóa sản phẩm
* Màn hình QL San Pham sẽ gửi yêu cầu kiểm tra thông tin sản phẩm.
* Controller Xoa San pham sẽ kiểm tra thông tin sản phẩm, thực hiện xóa thông tin sản phẩm trên cơ sở dữ liệu San pham.
* Thông báo xóa thành công.

*Sửa thông tin Sản phẩm:*



* Các thành phần trong hệ thống trong chức năng sửa sản phẩm: Màn hình QL San Pham, Controller Sua San pham, Cơ sở dữ liệu San Pham.
* Khái quát các bước trong chức năng sửa sản phẩm theo sơ đồ:
* Người dùng chọn chức năng liệt kê sản phẩm. Màn hình QL San Pham hiển thị danh sách sản phẩm lên màn hình.
* Người dùng chọn sản phẩm cần sửa, nhập thông tin cần sửa.
* Màn hình QL San Pham kiểm tra thông tin sản phẩm đã hơp lệ chưa.
* Nếu không hợp lệ yêu cầu nhập lại thông tin sản phẩm.
* Nếu hợp lệ thì Controller Sua San pham kiểm tra dữ liệu.
* Nếu không hợp lệ thì thông báo không hợp lệ, yêu cầu nhập lại
* Nếu hợp lệ thì lưu thông tin sản phẩm lên cơ sở dữ liệu sản phẩm, thông báo sửa thông tin sản phẩm thành công, Hiển thị danh sách sản phẩm.

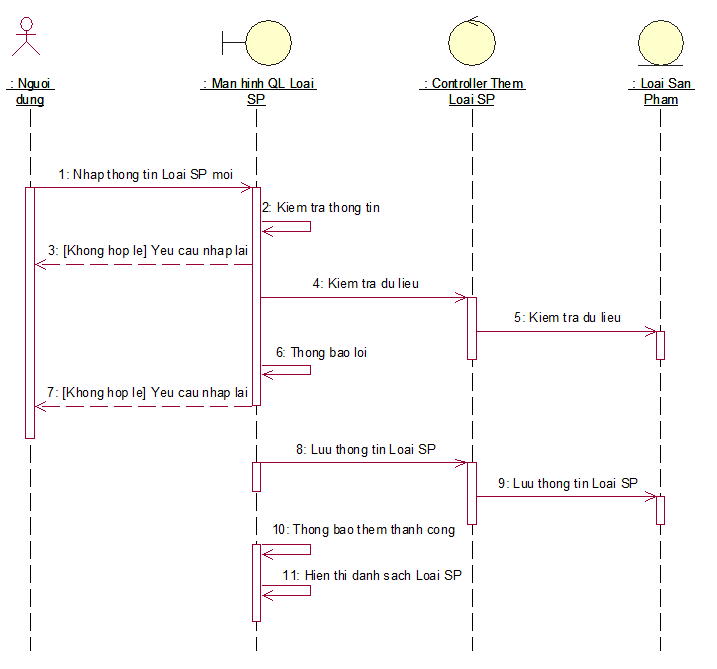
*Tra cứu sản phẩm:*



* Các thành phần trong hệ thống trong chức năng tra cứu sản phẩm: Màn hình QL San Pham, Controller Tra Cuu San pham, Cơ sở dữ liệu San Pham.
* Khái quát các bước trong chức năng tra cứu sản phẩm theo sơ đồ:
* Người dùng nhập từ khóa cần tìm kiếm.
* Màn hình QL San Pham kiểm tra thông tin có hợp lệ không.
* Nếu không hợp lệ thì yêu cầu nhập lại.
* Nếu hợp lệ thì Màn hình QL San Pham gửi yêu cầu thông tin sản phẩm tương ứng lên Controller Tra Cuu San pham. Controller Tra Cuu San pham gửi yêu cầu tìm dữ liệu sản phẩm tương ứng lên cơ sở dữ liệu sản phẩm.
* Cơ sở dữ liệu sản phẩm trả về thông tin sản phẩm tương ứng cho Controller Tra Cuu San pham, Controller Tra Cuu San pham trả về thông tin sản phẩm tương ứng cho Màn hình QL San Pham, Màn hình QL San Pham hiển thị thông tin sản phẩm tương ứng lên màn hình.

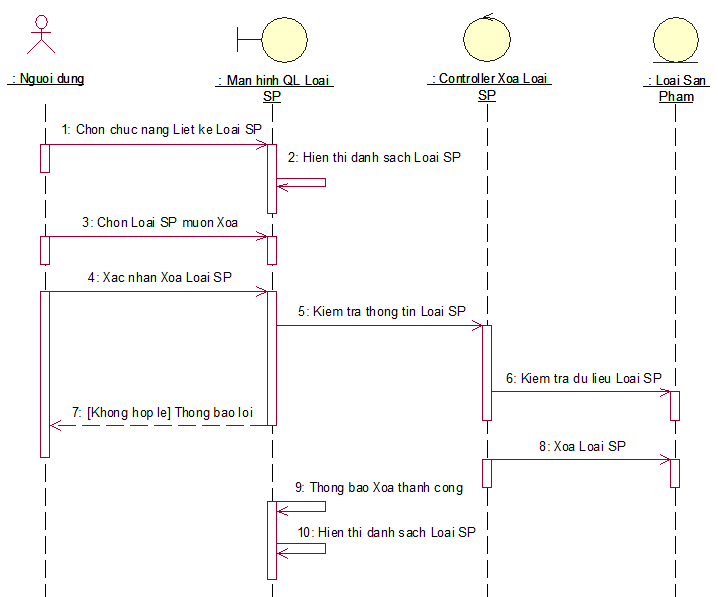
#### 3.2.4.3. Quản lý Loại Sản Phẩm

*Thêm Loại sản phẩm:*



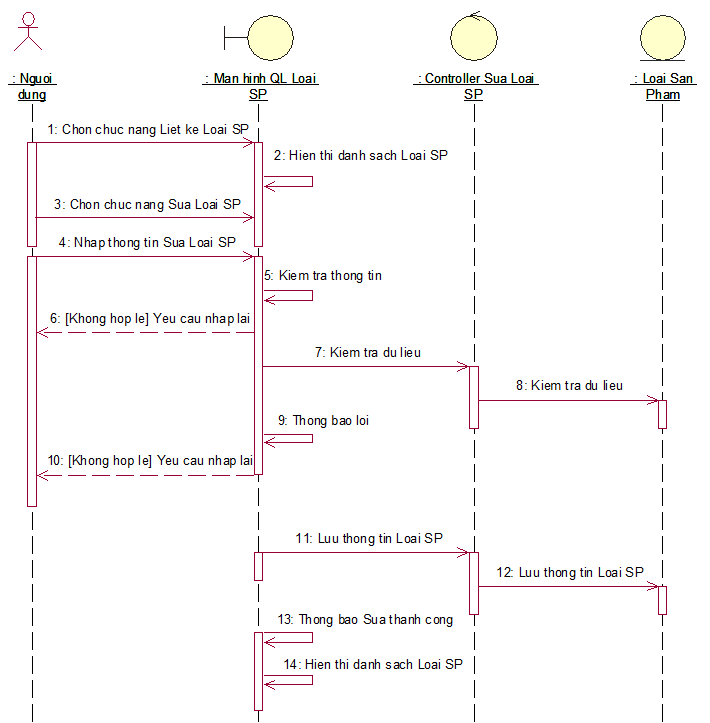
* Các thành phần trong hệ thống trong chức năng thêm loại sản phẩm: Màn hình QL Loai SP, Controller Them Loai SP, Cơ sở dữ liệu Loai San Pham.
* Khái quát các bước trong chức năng thêm loại sản phẩm theo sơ đồ:
* Người dùng nhập thông tin loại sản phẩm cần thêm.
* Màn hình QL Loai SP kiểm tra thông tin loại sản phẩm.
* Nếu không hợp lệ yêu cầu nhập lại.
* Nếu hợp lệ thì bắt đầu kiểm tra dữ liệu
* Nếu dữ liệu không đúng thì yêu cầu nhập lại.
* Nếu dữ liệu hợp lệ thì lưu thông tin loại sản phẩm lên cơ sở dữ liệu và thông báo thêm loại sản phẩm thành công, hiển thị thông tin loại sản phẩm.

*Xóa loại sản phẩm:*



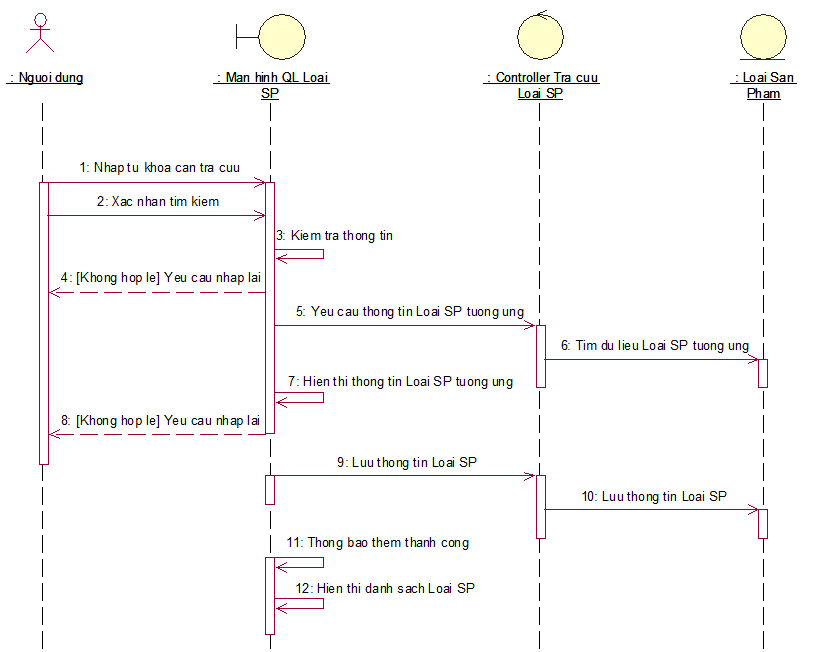
* Các thành phần trong hệ thống trong chức năng xóa loại sản phẩm: Màn hình QL Loai SP, Controller Xoa Loai SP, Cơ sở dữ liệu Loai San Pham.
* Khái quát các bước trong chức năng xóa loại sản phẩm theo sơ đồ:
* Người dùng chọn chức năng liệt kê Loai SP, hiển thị danh sách Loai SP lên màn hình. Người dùng chọn loại sản phẩm muốn xóa, xác nhận xóa loại sản phẩm.
* Màn hình QL Loai SP sẽ gửi yêu cầu kiểm tra thông tin loại sản phẩm.
* Controller Xoa Loai SP sẽ kiểm tra thông tin loại sản phẩm, thực hiện xóa thông tin loại sản phẩm trên cơ sở dữ liệu Loai San pham.
* Thông báo xóa thành công.

*Sửa Loại sản phẩm:*



* Các thành phần trong hệ thống trong chức năng sửa loại sản phẩm: Màn hình QL Loai SP, Controller Sua Loai SP, Cơ sở dữ liệu Loai San Pham.
* Khái quát các bước trong chức năng sửa loại sản phẩm theo sơ đồ:
* Người dùng chọn chức năng liệt kê loại sản phẩm. Màn hình QL Loai SP hiển thị danh sách loại sản phẩm lên màn hình.
* Người dùng chọn loại sản phẩm cần sửa, nhập thông tin cần sửa.
* Màn hình QL Loại SP kiểm tra thông tin loại sản phẩm đã hợp lệ chưa.
* Nếu không hợp lệ yêu cầu nhập lại thông tin loại sản phẩm.
* Nếu hợp lệ thì Controller Sua Loại SP kiểm tra dữ liệu.
* Nếu không hợp lệ thì thông báo không hợp lệ, yêu cầu nhập lại
* Nếu hợp lệ thì lưu thông tin loại sản phẩm lên cơ sở dữ liệu loại sản phẩm, thông báo sửa thông tin loại sản phẩm thành công, Hiển thị danh sách loại sản phẩm.

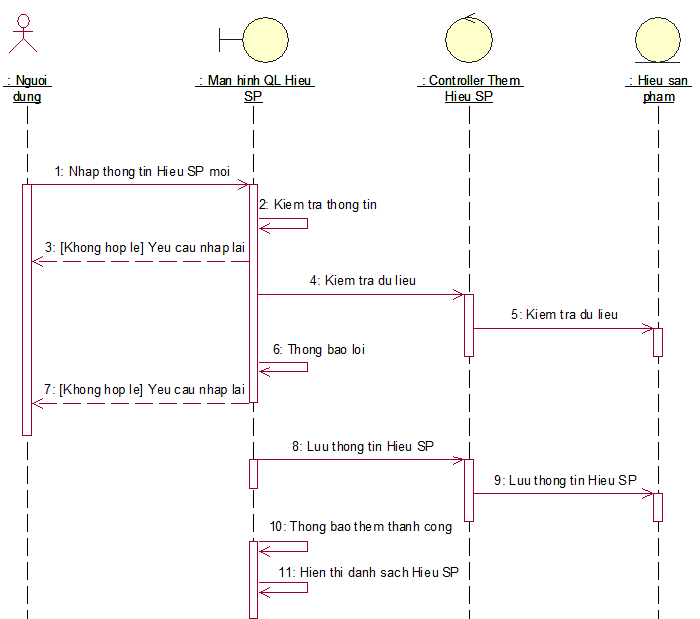
*Tra cứu loại sản phẩm:*



* Các thành phần trong hệ thống trong chức năng tra cứu loại sản phẩm: Màn hình QL Loai SP, Controller Tra Cuu LoaiSP, Cơ sở dữ liệu Loai San Pham.
* Khái quát các bước trong chức năng tra cứu loại sản phẩm theo sơ đồ:
* Người dùng nhập từ khóa cần tìm kiếm.
* Màn hình QL Loai SP kiểm tra thông tin có hợp lệ không.
* Nếu không hợp lệ thì yêu cầu nhập lại.
* Nếu hợp lệ thì Màn hình QL Loai SP gửi yêu cầu thông tin loại sản phẩm tương ứng lên Controller Tra Cuu Loai SP. Controller Tra Cuu Loai SP gửi yêu cầu tìm dữ liệu loại sản phẩm tương ứng lên cơ sở dữ liệu loại sản phẩm.
* Cơ sở dữ liệu loại sản phẩm trả về thông tin loại sản phẩm tương ứng cho Controller Tra Cuu Loai SP, Controller Tra Cuu Loai SP trả về thông tin loại sản phẩm tương ứng cho Màn hình QL Loai SP, Màn hình QL Loai SP hiển thị thông tin loại sản phẩm tương ứng lên màn hình.

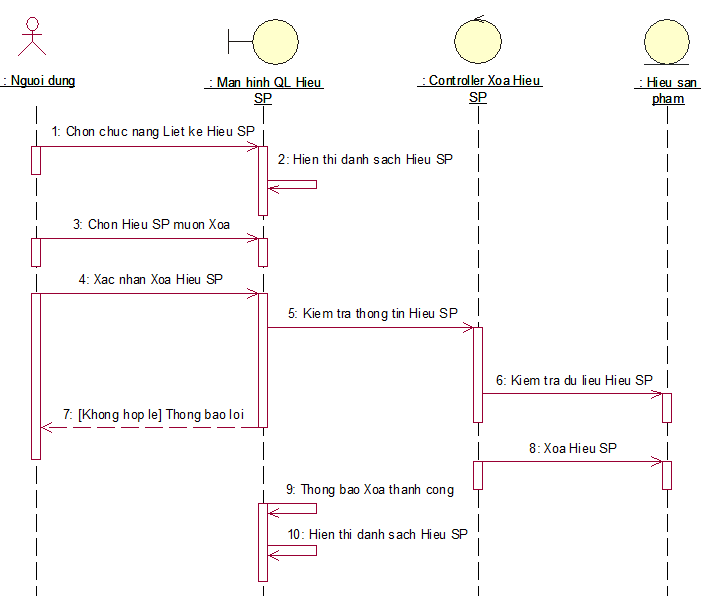
#### 3.2.4.4. Quản lý Hiệu sản phẩm

*Thêm Hiệu sản phẩm:*

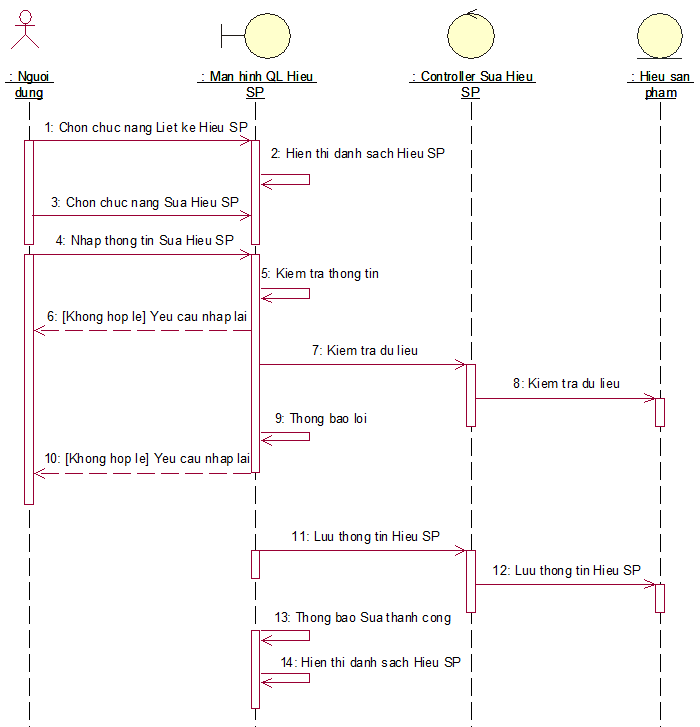


* Các thành phần trong hệ thống trong chức năng thêm hiệu sản phẩm: Màn hình QL Hieu SP, Controller Them Hieu SP, Cơ sở dữ liệu Hieu San Pham.
* Khái quát các bước trong chức năng thêm hiệu sản phẩm theo sơ đồ:
* Người dùng nhập thông tin hiệu sản phẩm cần thêm.
* Màn hình QL Hieu SP kiểm tra thông tin hiệu sản phẩm.
* Nếu không hợp lệ yêu cầu nhập lại.
* Nếu hợp lệ thì bắt đầu kiểm tra dữ liệu
* Nếu dữ liệu không đúng thì yêu cầu nhập lại.
* Nếu dữ liệu hợp lệ thì lưu thông tin hiệu sản phẩm lên cơ sở dữ liệu và thông báo thêm hiệu sản phẩm thành công, hiển thị thông tin hiệu sản phẩm.

*Xóa Hiệu sản phẩm:*

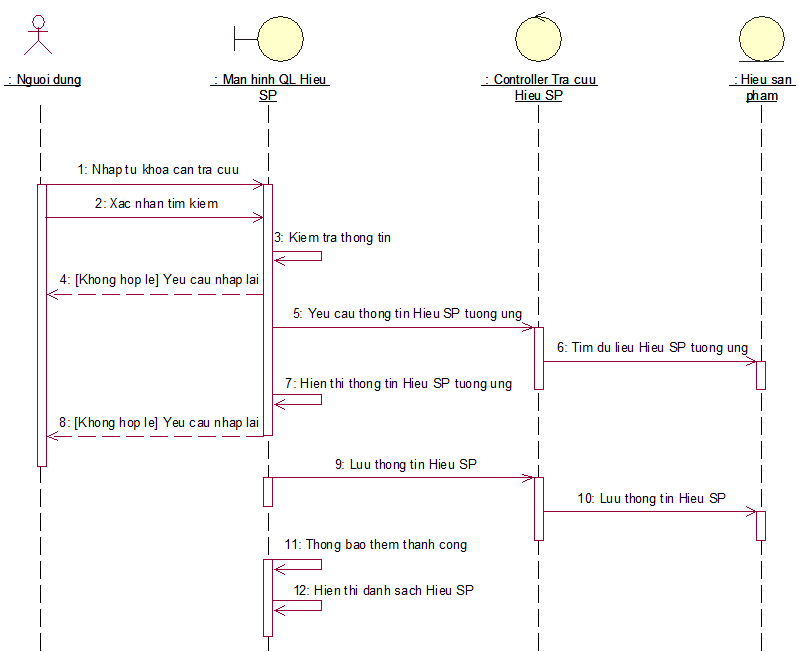


* Các thành phần trong hệ thống trong chức năng xóa hiệu sản phẩm: Màn hình QL Hieu SP, Controller Xoa Hieu SP, Cơ sở dữ liệu Hieu San Pham.
* Khái quát các bước trong chức năng xóa hiệu sản phẩm theo sơ đồ:
* Người dùng chọn chức năng liệt kê Hieu SP, hiển thị danh sách Hieu SP lên màn hình. Người dùng chọn hiệu sản phẩm muốn xóa, xác nhận xóa hiệu sản phẩm.
* Màn hình QL Hieu SP sẽ gửi yêu cầu kiểm tra thông tin hiệu sản phẩm.
* Controller Xoa Hieu SP sẽ kiểm tra thông tin hiệu sản phẩm, thực hiện xóa thông tin hiệu sản phẩm trên cơ sở dữ liệu Hieu San pham.
* Thông báo xóa thành công.

*Sửa Hiệu sản phẩm:*

* Các thành phần trong hệ thống trong chức năng sửa hiệu sản phẩm: Màn hình QL Hieu SP, Controller Sua Hieu SP, Cơ sở dữ liệu Hieu San Pham.
* Khái quát các bước trong chức năng sửa hiệu sản phẩm theo sơ đồ:
* Người dùng chọn chức năng liệt kê hiệu sản phẩm. Màn hình QL Hieu SP hiển thị danh sách hiệu sản phẩm lên màn hình.
* Người dùng chọn hiệu sản phẩm cần sửa, nhập thông tin cần sửa.
* Màn hình QL Hieu SP kiểm tra thông tin hiệu sản phẩm đã hợp lệ chưa.
* Nếu không hợp lệ yêu cầu nhập lại thông tin hiệu sản phẩm.
* Nếu hợp lệ thì Controller Sua Hieu SP kiểm tra dữ liệu.
* Nếu không hợp lệ thì thông báo không hợp lệ, yêu cầu nhập lại
* Nếu hợp lệ thì lưu thông tin hiệu sản phẩm lên cơ sở dữ liệu hiệu sản phẩm, thông báo sửa thông tin hiệu sản phẩm thành công. Hiển thị danh sách hiệu sản phẩm.

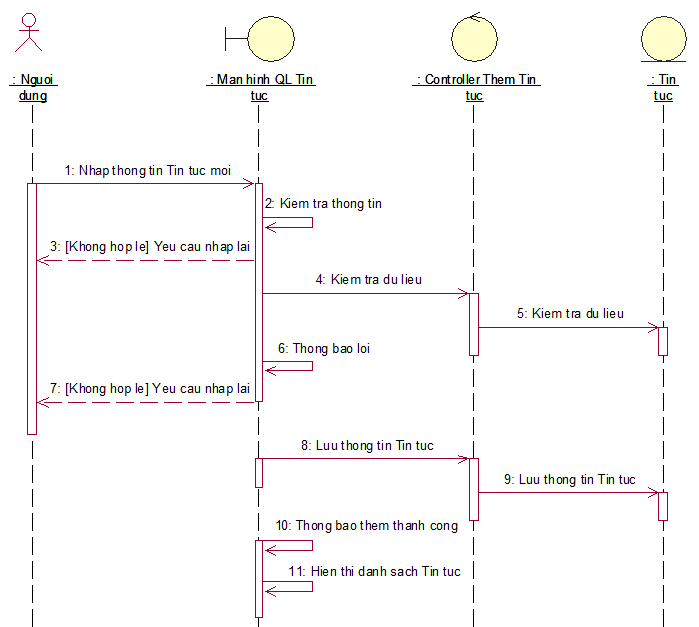
*Tra cứu Hiệu sản phẩm:*



* Các thành phần trong hệ thống trong chức năng tra cứu hiệu sản phẩm: Màn hình QL Hieu SP, Controller Tra Cuu Hieu SP, Cơ sở dữ liệu Hieu San Pham.
* Khái quát các bước trong chức năng tra cứu hiệu sản phẩm theo sơ đồ:
* Người dùng nhập từ khóa cần tìm kiếm.
* Màn hình QL Hieu SP kiểm tra thông tin có hợp lệ không.
* Nếu không hợp lệ thì yêu cầu nhập lại.
* Nếu hợp lệ thì Màn hình QL Hieu SP gửi yêu cầu thông tin hiệu sản phẩm tương ứng lên Controller Tra Cuu Hieu SP. Controller Tra Cuu Hieu SP gửi yêu cầu tìm dữ liệu hiệu sản phẩm tương ứng lên cơ sở dữ liệu hiệu sản phẩm.
* Cơ sở dữ liệu hiệu sản phẩm trả về thông tin hiệu sản phẩm tương ứng cho Controller Tra Cuu Hieu SP, Controller Tra Cuu Hieu SP trả về thông tin hiệu sản phẩm tương ứng cho Màn hình QL Hieu SP, Màn hình QL Hieu SP hiển thị thông tin hiệu sản phẩm tương ứng lên màn hình.

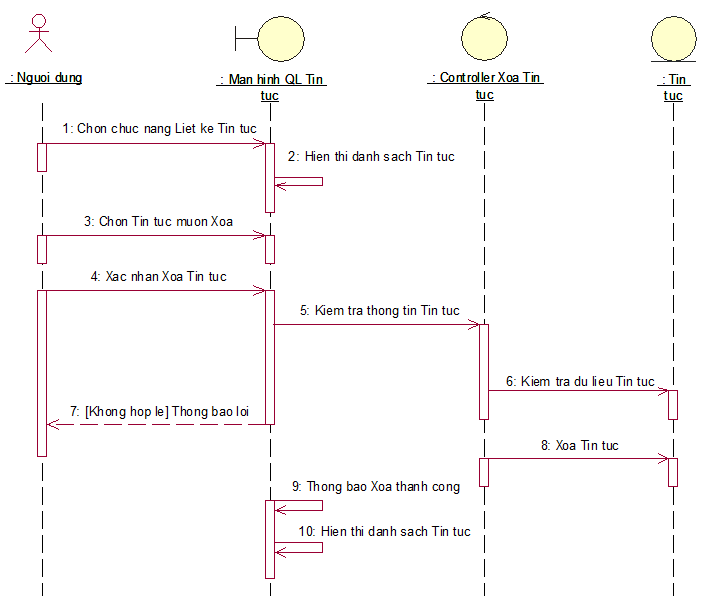
#### 3.2.4.5. Quản lý Tin tức

*Thêm thông tin Tin tức:*



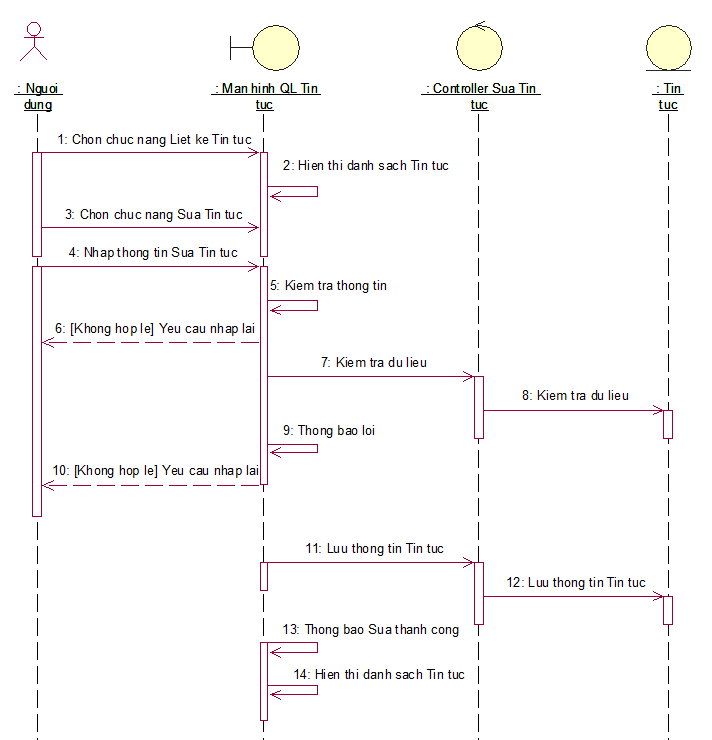
* Các thành phần trong hệ thống trong chức năng thêm tin tức: Màn hình QL Tin tức, Controller Them Tin tức, Cơ sở dữ liệu Tin tức.
* Khái quát các bước trong chức năng thêm tin tức theo sơ đồ:
* Người dùng nhập thông tin tin tức cần thêm.
* Màn hình QL Tin tức kiểm tra thông tin tin tức.
* Nếu không hợp lệ yêu cầu nhập lại.
* Nếu hợp lệ thì bắt đầu kiểm tra dữ liệu
* Nếu dữ liệu không đúng thì yêu cầu nhập lại.
* Nếu dữ liệu hợp lệ thì lưu thông tin tin tức lên cơ sở dữ liệu và thông báo thêm tin tức thành công, hiển thị thông tin tin tức.

*Xóa thông tin Tin tức:*



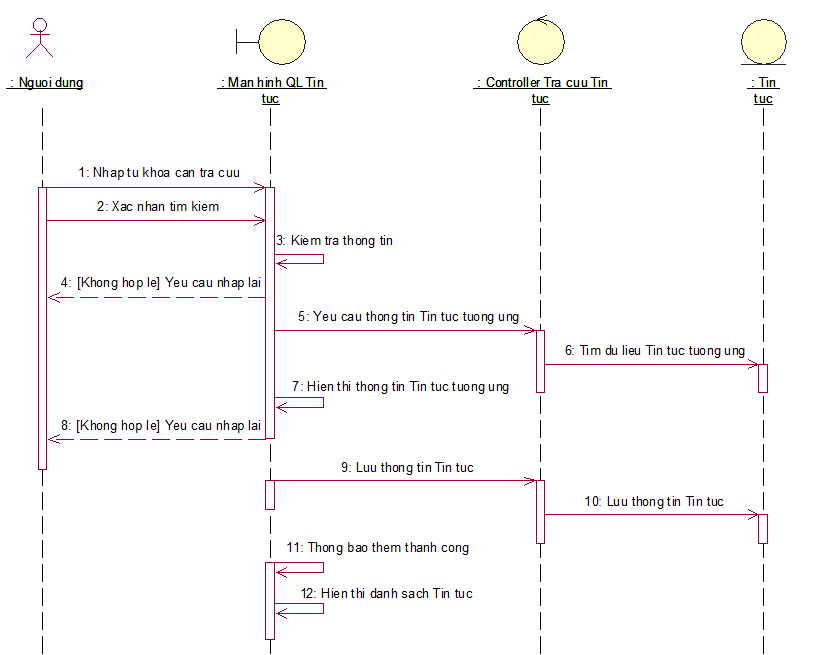
* Các thành phần trong hệ thống trong chức năng xóa tin tức: Màn hình QL Tin tức, Controller Xoa Tin tuc, Cơ sở dữ liệu Tin tuc.
* Khái quát các bước trong chức năng xóa tin tức theo sơ đồ:
* Người dùng chọn chức năng liệt kê Tin tuc, hiển thị danh sách Tin tuc lên màn hình. Người dùng chọn tin tức muốn xóa, xác nhận xóa tin tức.
* Màn hình QL Tin tức sẽ gửi yêu cầu kiểm tra thông tin tin tức.
* Controller Xoa Tin tuc sẽ kiểm tra thông tin tin tức, thực hiện xóa thông tin tin tức trên cơ sở dữ liệu Tin tuc.
* Thông báo xóa thành công.

*Sửa thông tin Tin tức :*



* Các thành phần trong hệ thống trong chức năng sửa tin tức: Màn hình QL Tin tuc, Controller Sua Tin tuc, Cơ sở dữ liệu Tin tuc.
* Khái quát các bước trong chức năng sửa tin tức theo sơ đồ:
* Người dùng chọn chức năng liệt kê tin tức. Màn hình QL Tin tuc hiển thị danh sách tin tức lên màn hình.
* Người dùng chọn tin tức cần sửa, nhập thông tin cần sửa.
* Màn hình QL Tin tuc kiểm tra thông tin tin tức đã hợp lệ chưa.
* Nếu không hợp lệ yêu cầu nhập lại thông tin tin tức.
* Nếu hợp lệ thì Controller Sua Tin tuc kiểm tra dữ liệu.
* Nếu không hợp lệ thì thông báo không hợp lệ, yêu cầu nhập lại
* Nếu hợp lệ thì lưu thông tin tin tức lên cơ sở dữ liệu tin tức, thông báo sửa thông tin tin tức thành công. Hiển thị danh sách tin tức.

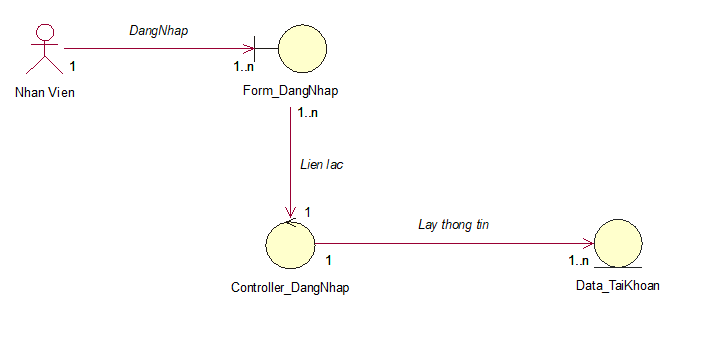
*Tra cứu thông tin Tin tức:*



* Các thành phần trong hệ thống trong chức năng tra cứu tin tức: Màn hình QL Tin tuc, Controller Tra Cuu Tin tuc, Cơ sở dữ liệu Tin tuc.
* Khái quát các bước trong chức năng tra cứu tin tức theo sơ đồ:
* Người dùng nhập từ khóa cần tìm kiếm.
* Màn hình QL Tin tuc kiểm tra thông tin có hợp lệ không.
* Nếu không hợp lệ thì yêu cầu nhập lại.
* Nếu hợp lệ thì Màn hình QL Tin tuc gửi yêu cầu thông tin tin tức tương ứng lên Controller Tra Cuu Tin tuc. Controller Tra Cuu Tin tuc gửi yêu cầu tìm dữ liệu tin tức tương ứng lên cơ sở dữ liệu tin tức.
* Cơ sở dữ liệu tin tức trả về thông tin tin tức tương ứng cho Controller Tra Cuu Tin tuc, Controller Tra Cuu Tin tuc trả về thông tin tin tức tương ứng cho Màn hình QL Tin tuc, Màn hình QL Tin tuc hiển thị thông tin tin tức tương ứng lên màn hình.

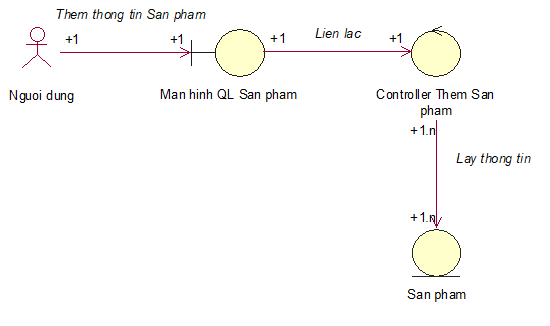
### 3.2.5. Class Diagram

#### 3.2.5.1. Đăng nhập

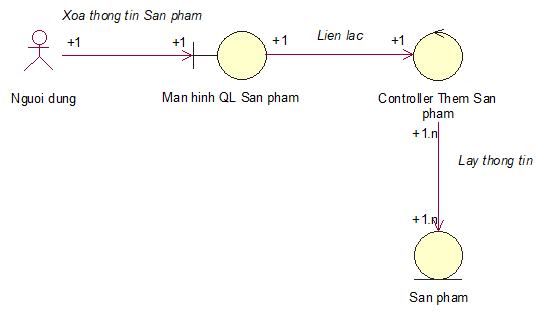


#### 3.2.5.2. Quản lí sản phẩm

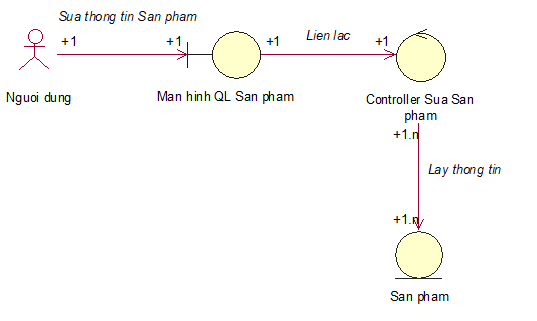
*Thêm Sản phẩm:*



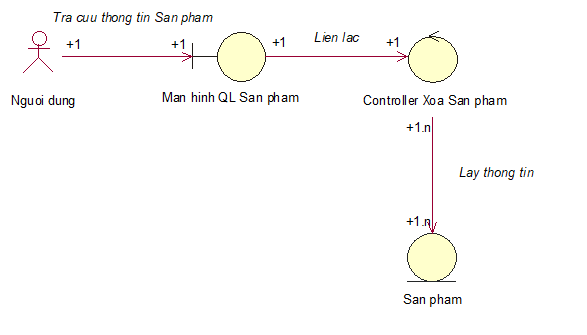
*Xóa Sản phẩm:*



*Sửa Sản phẩm:*



*Tra cứu Sản phẩm:*

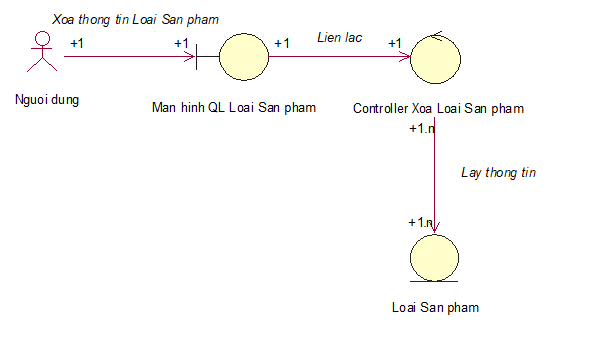


#### 3.2.5.3. Quản lí loại sản phẩm

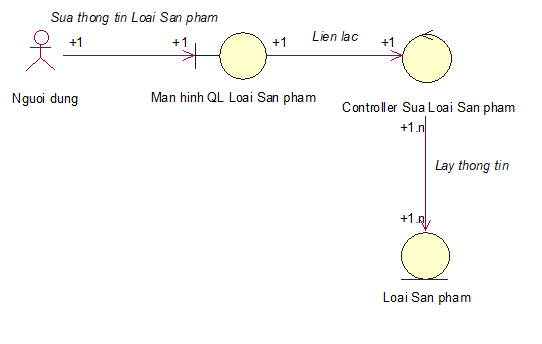
*Thêm Loại sản phẩm:*



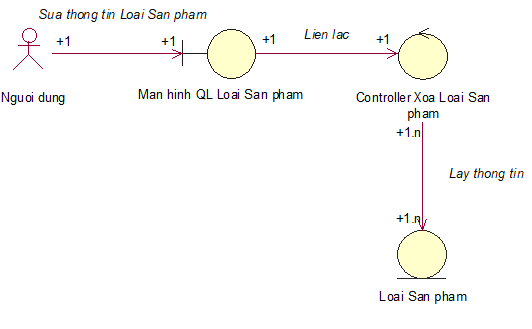
*Xóa Loại sản phẩm:*



*Sửa Loại sản phẩm:*

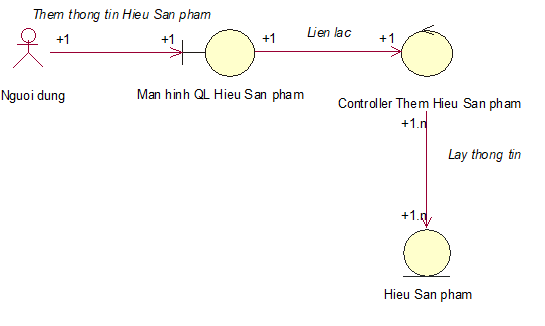


*Tra cứu Loại sản phẩm:*

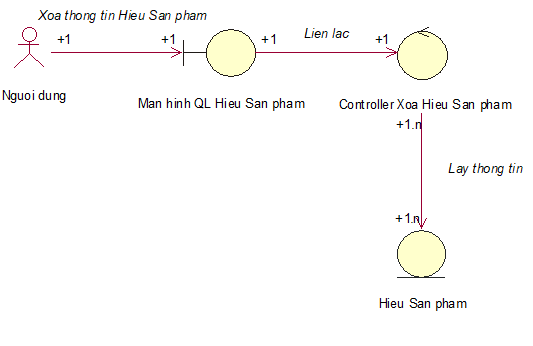


#### 3.2.5.4. Quản lí hiệu sản phẩm

*Thêm Hiệu sản phẩm:*



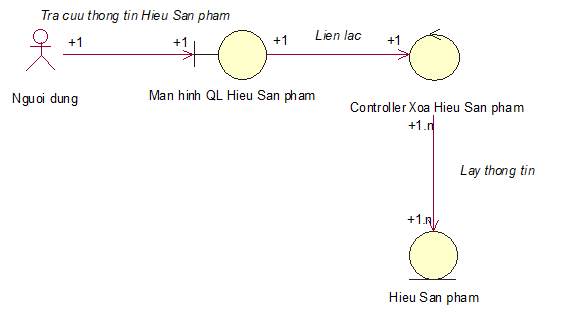
*Xóa Hiệu sản phẩm:*



*Sửa Hiệu sản phẩm:*

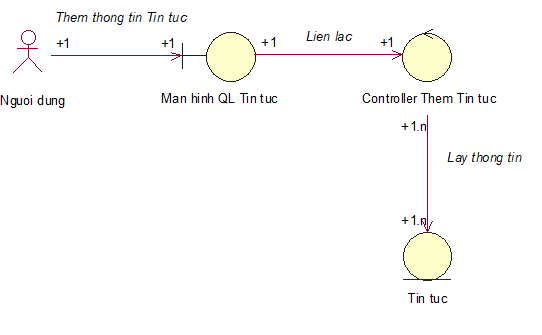


*Tra cứu Hiệu sản phẩm:*

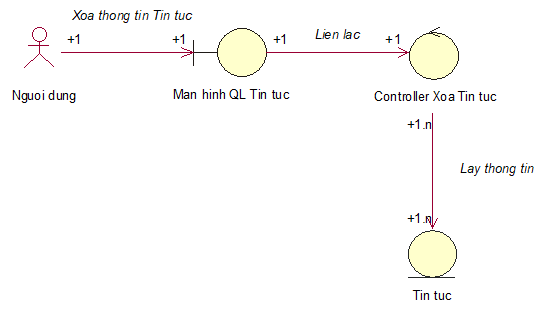


#### 3.2.5.5. Quản lí tin tức

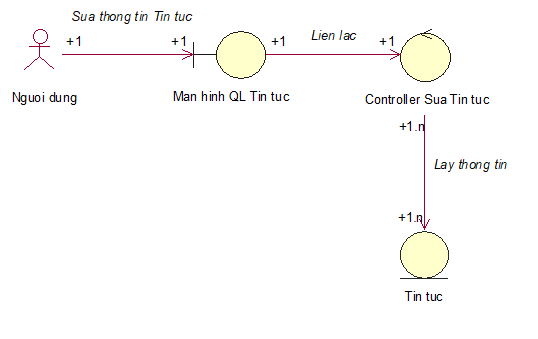
*Thêm Tin tức:*



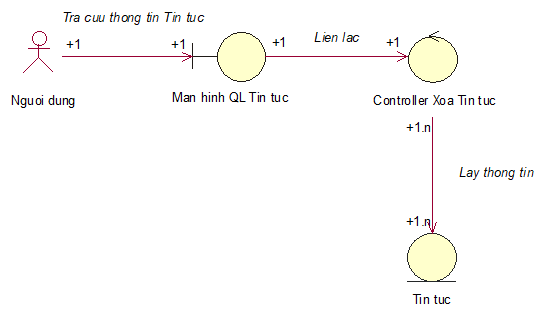
*Xóa Tin tức:*



*Sửa Tin tức:*



*Tra cứu Tin tức:*

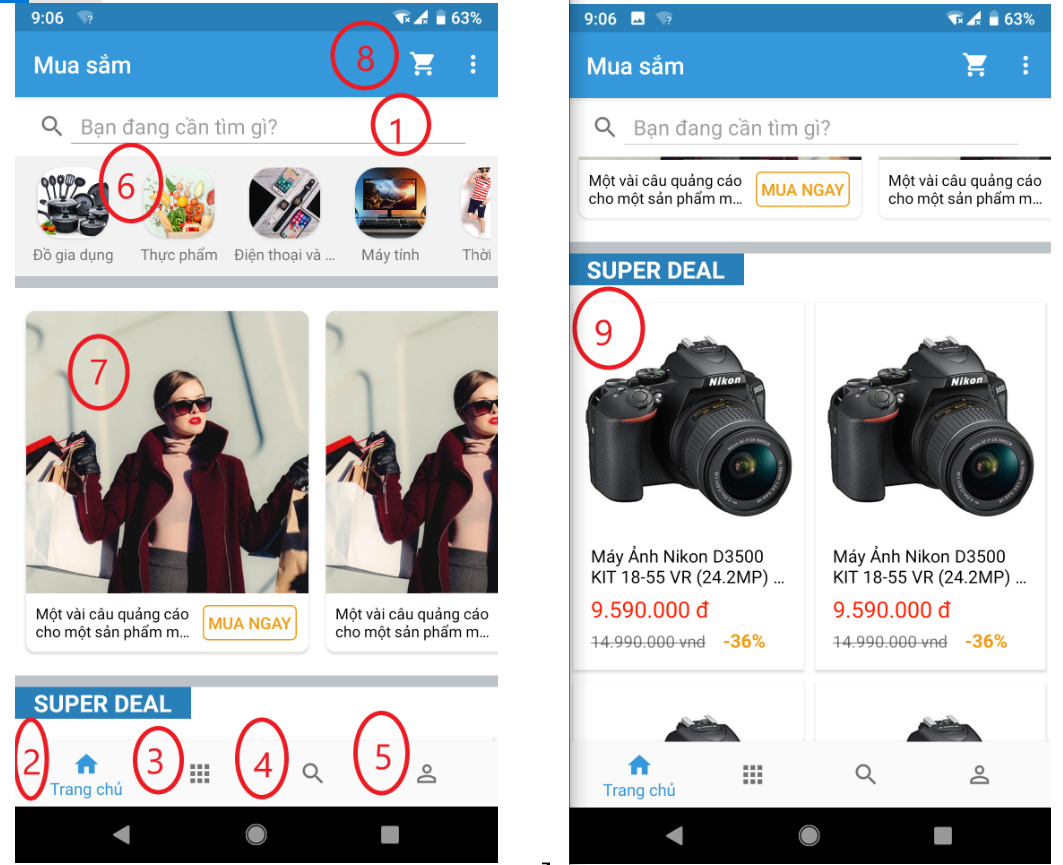


# CHƯƠNG 4: HIỆN THỰC ỨNG DỤNG

## 4.1. App bán hàng

### 4.1.1. Màn hình trang chủ

* Khi đường truyền internet ổn định hoặc kết nối đến server không bị lỗi



* Điều hướng và tương tác người dùng:

(1): Thanh tìm kiếm, khi nhấn vào sẽ chuyển sang màn hình tìm kiếm.

(2): Button trang chủ, khi nhấn vào sẽ hiện trang chủ của app.

(3): Button danh mục sản phẩm, khi nhấn vào sẽ đi đến màn hình danh mục sản phẩm.

(4): Button tìm kiếm, khi nhấn vào sẽ hiện trang tìm kiếm.

(5): Khi nhấn vào sẽ điều hướng đến trang quản lý tài khoản.

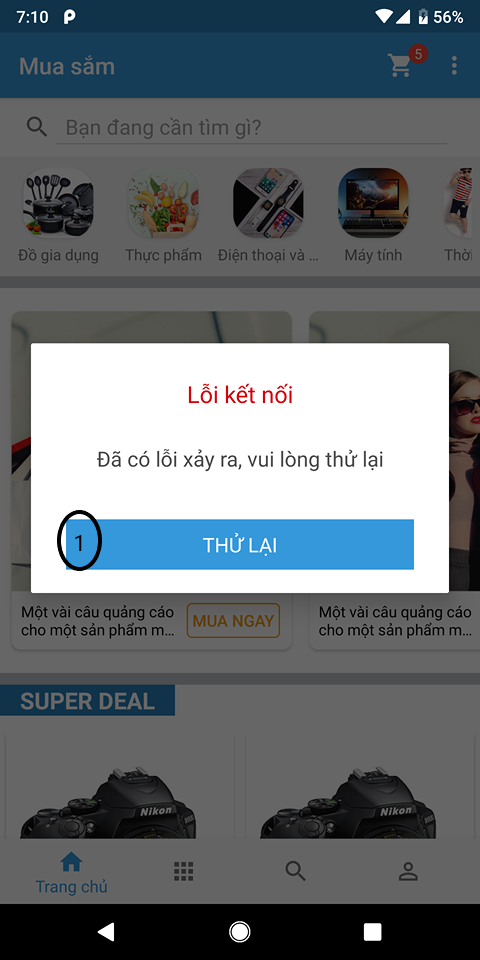
(6): Khi nhấn vào sẽ đi đến màn hình danh sách sản phẩm.

(7): Khi nhấn vào sẽ đi đến màn hình giỏ hàng với sản phẩm đã chọn

(8): Khi người dùng click vào, sẽ chuyển đến màn hình giỏ hàng để tiến hành mua hàng.

(9): Khi người dùng click vào sẽ đi đến màn hình Chi tiết sản phẩm

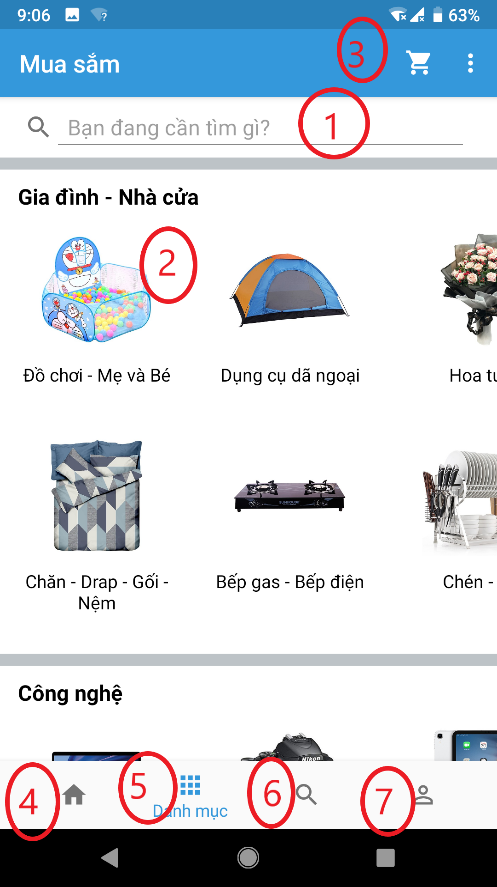
* Khi đường truyền internet không được ổn định và kết nối đến server bị lỗi thì ứng dụng sẽ hiện ra thông báo lỗi kết nối và yêu cầu người dùng thử lại



* Điều hướng và tương tác người dùng:

(1): người dùng nhấn thử lại kết nối đến ứng dụng.

### 4.1.2. Màn hình danh mục



* Điều hướng và tương tác người dùng:

(1): Thanh tìm kiếm, khi nhấn vào sẽ chuyển sang màn hình tìm kiếm.

(2): Khi người dùng nhấn vào sẽ chuyển sang màn hình danh sách sản phẩm.

(3): Khi người dùng click vào, sẽ chuyển đến màn hình giỏ hàng để tiến hành mua hàng.

(4): Button trang chủ, khi nhấn vào sẽ hiện trang chủ của app.

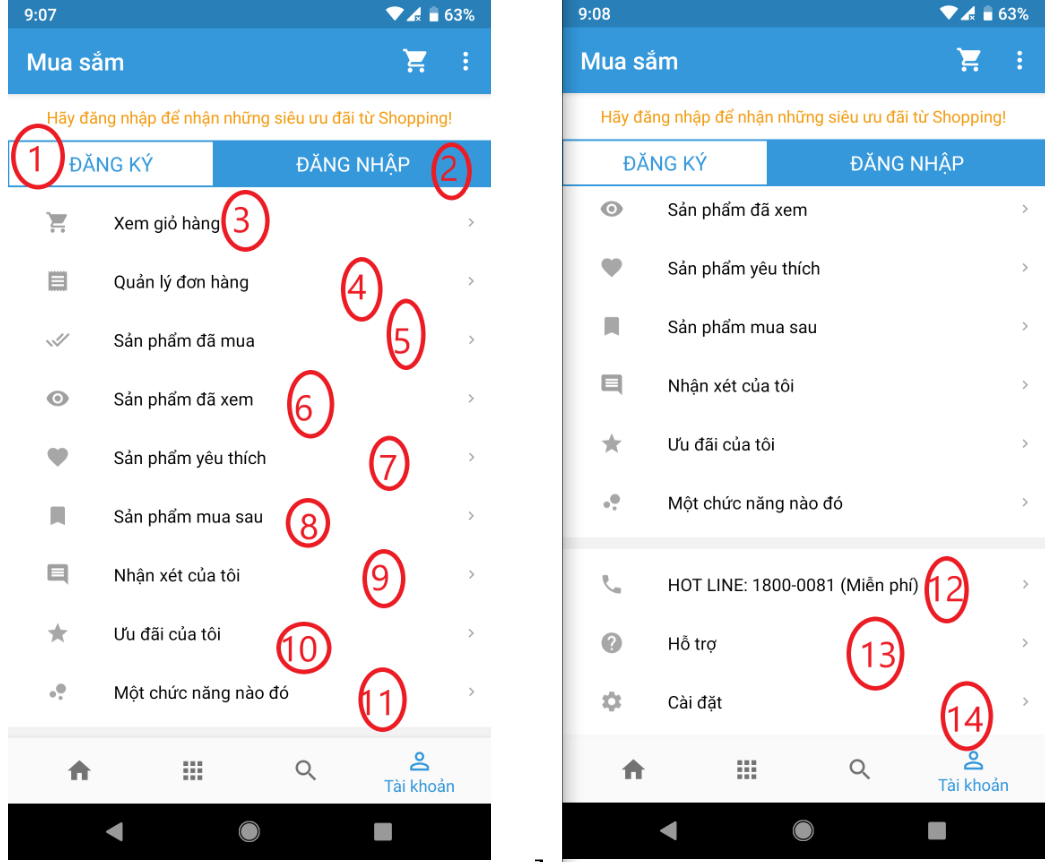
(5): Button danh mục sản phẩm, khi nhấn vào sẽ đi đến màn hình danh mục sản phẩm

(6): Button tìm kiếm, khi nhấn vào sẽ hiện trang tìm kiếm.

(7): Khi nhấn vào sẽ điều hướng đến trang quản lý tài khoản

### 4.1.3. Màn hình tài khoản

* **Màn hình tài khoản chưa đăng nhập:**



* Điều hướng và tương tác người dùng:

(1): Khi nhấn vào sẽ chuyển đến màn hình đăng kí.

(2): Khi nhấn vào sẽ chuyển đến màn hình đăng nhập.

(3): Khi nhấn vào sẽ chuyển đến màn hình giỏ hàng.

(4): Khi nhấn vào sẽ chuyển đến màn hình quản lý đơn hàng.

(5): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm người dùng đã mua.

(6): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm mà người dùng đã xem.

(7): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm àm người dùng ưa thích.

(8): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm mua sau.

(9): Khi nhấn vào sẽ chuyển đến màn hình nhận xét.

(10): Khi nhấn vào sẽ chuyển đến màn hình ưu đãi.

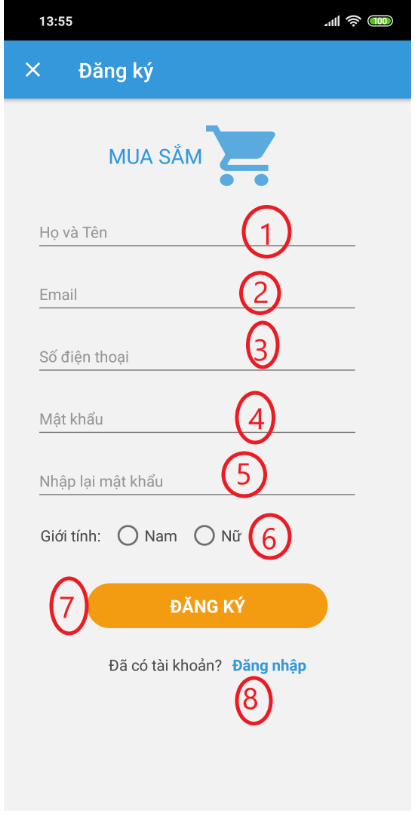
(11): Khi nhấn vào sẽ chuyển đến màn hình chức năng khác.

(12): Khi nhấn vào sẽ chuyển đến màn hình tổng đài chăm sóc khách hàng.

(13): Khi nhấn vào sẽ chuyển đến màn hình hỗ trợ.

(14): Khi nhấn vào sẽ chuyển đến màn hình cài đặt tài khoản.

* **Màn hình đăng kí**



* Điều hướng và tương tác người dùng:

(1): Người dùng nhập họ tên

(2): Người dùng nhập email

(3): Người dùng nhập số điện thoại

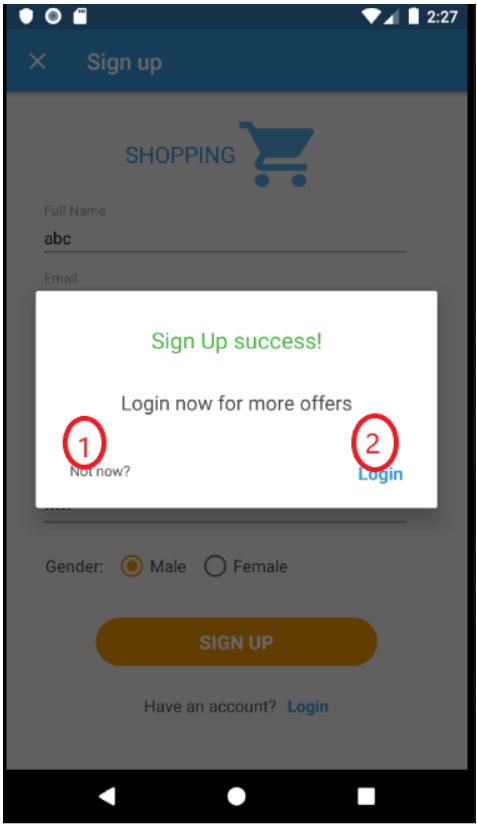
(4): Người dùng nhập mật khẩu

(5): Người dùng nhập lại mật khẩu

(6): Người dùng chọn giới tính

(7): Sau khi người dùng nhấn đăng ký. Ứng dụng sẽ kiểm tra các dữ liệu mà người dùng nhập vào.

* Trường hợp thông tin được nhập đúng và đầy đủ thì sẽ hiện ra thông báo đã đăng kí thành công cho người dùng.

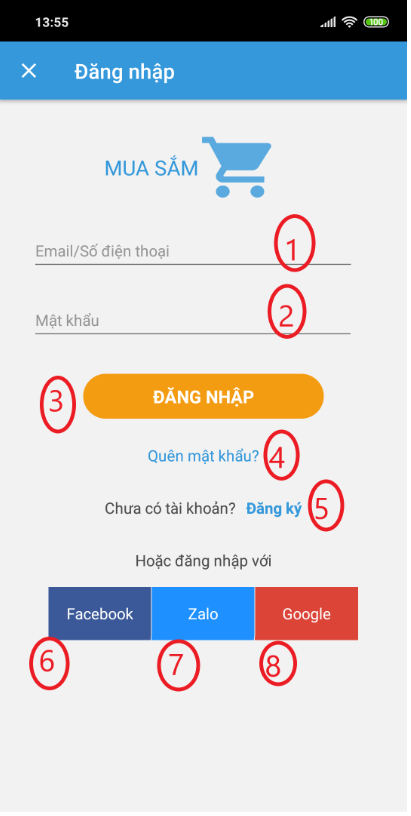


* Thông báo đăng kí thành công:

(1): Khi nhấn vào sẽ tắt thông báo

(2): Khi nhấn vào sẽ đi đến màn hình đăng nhập

* Trường hợp thông tin sai thì người dùng sẽ nhập lại thông tin.
* **Màn hình đăng nhập**



* Điều hướng và tương tác người dùng:

(1): Người dùng nhập email hoặc số điện thoại

(2): Người dùng nhập mật khẩu

(3): Sau khi người dùng nhấn vào nút đăng nhập. Ứng dụng sẽ kiểm tra thông tin mà người dùng nhập vào.

* Nếu đúng thì tiến hành đăng nhập và đi đến màn hình tài khoản.
* Nếu sai thì yêu cầu người dùng nhập lại.

(4): Khi người dùng nhấn vào, sẽ đi đến màn hình rết lại mật khẩu

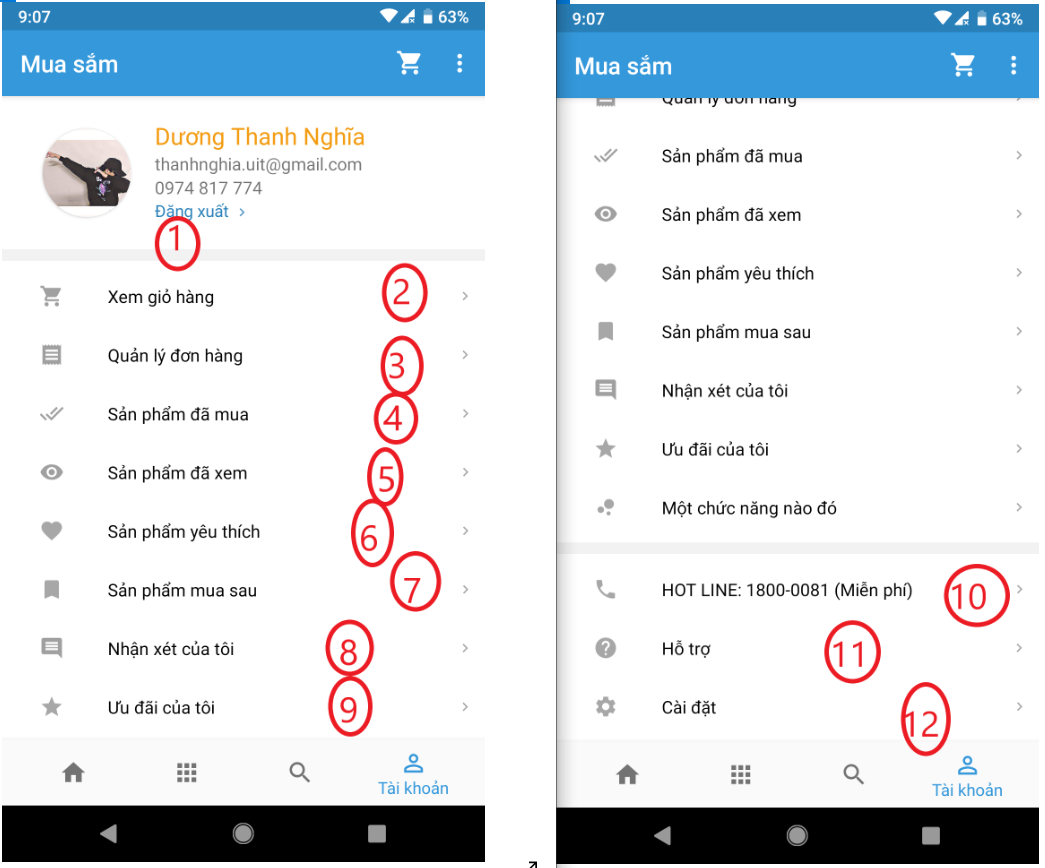
(5): Khi nhấn vào sẽ đi đến màn hình đăng ký

(6): Đăng nhập với facebook

(7): Đăng nhập với Zalo

(8): Đăng nhập với Google.

* **Màn hình tài khoản đã đăng nhập:**



* Điều hướng và tương tác người dùng:

(1): Khi nhấn vào thì ứng dụng sẽ đăng xuất khoản ngừi dùng và hiển thị màn hình tài khoản chưa đăng nhập.

(2): Khi nhấn vào sẽ chuyển đến màn hình giỏ hàng.

(3): Khi nhấn vào sẽ chuyển đến màn hình quản lý đơn hàng.

(4): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm người dùng đã mua.

(5): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm mà người dùng đã xem.

(6): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm àm người dùng ưa thích.

(7): Khi nhấn vào sẽ chuyển đến màn hình sản phẩm mua sau.

(8): Khi nhấn vào sẽ chuyển đến màn hình nhận xét.

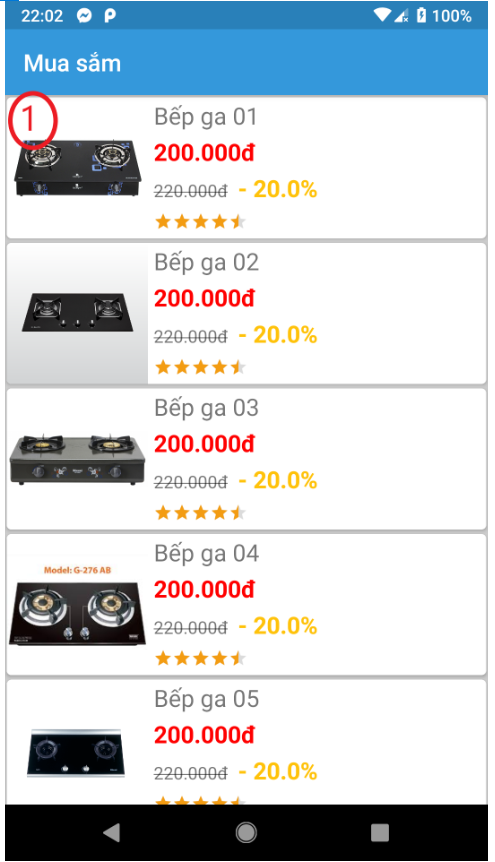
(9): Khi nhấn vào sẽ chuyển đến màn hình ưu đãi.

(10): Khi nhấn vào sẽ chuyển đến màn hình tổng đài chăm sóc khách hàng.

(11): Khi nhấn vào sẽ chuyển đến màn hình hỗ trợ.

(12): Khi nhấn vào sẽ chuyển đến màn hình cài đặt tài khoản.

### 4.1.4. Màn hình danh sách sản phẩm

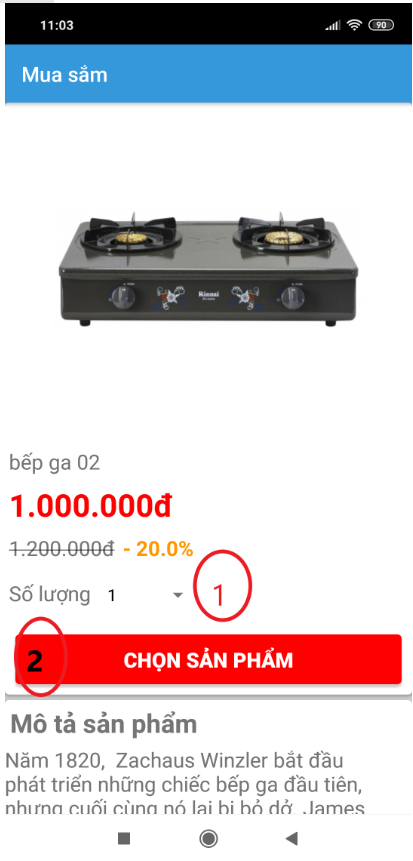


* Điều hướng và tương tác người dùng:

(1): Khi nhấn vào sẽ chuyển đến màn hình chi tiết sản phẩm tương ứng.

* Các chi tiết:
  + - * + Hình ảnh sản phẩm.
        + Tên sản phẩm.
        + Giá mới (Được làm màu đỏ để nổi bật hơn).
        + Giá cũ chưa giảm (Được làm màu xám và có gạch ngang ở giữa).
        + Phần trăm đã được giảm (Được làm màu nổi để thu hút sự chú ý).
        + Đánh giá của khách hàng về sản phẩm (Được hiển thị theo 5 sao cho trực quan hơn).

### 4.1.5. Màn hình chi tiết sản phẩm



* Hiển thị thông tin chi tiết của sản phẩm tương ứng.
* Điều hướng và tương tác người dùng:

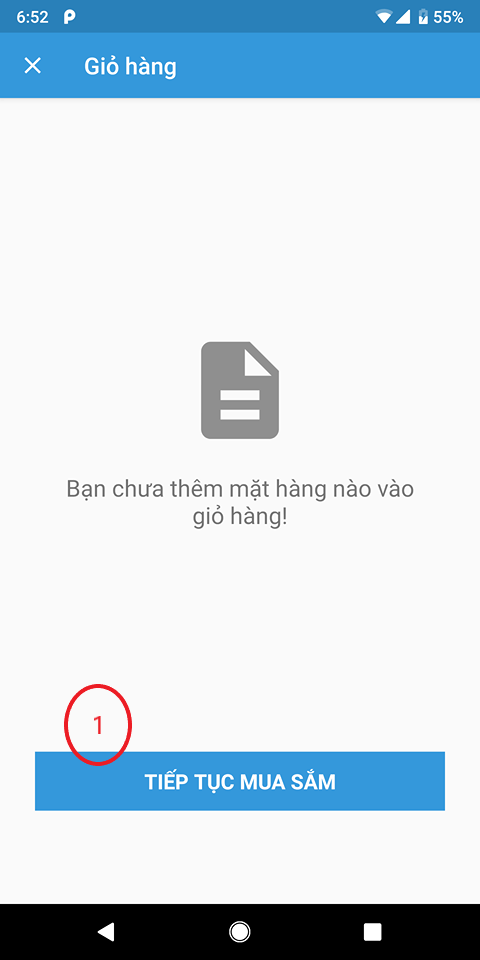
(1): Người dùng chọn số lượng sản phẩm

(2): Khi nhấn vào nút Chọn sản phẩm thì sẽ thêm sản phẩm vào giỏ hàng

* Các chi tiết
  + - * + Hình ảnh sản phẩm.
        + Tên sản phẩm.
        + Giá mới (Được làm màu nổi bật để thu hút sự chú ý).
        + Giá cũ chưa giảm (Được làm màu xám và có gạch ngang).
        + Phần trăm đã được giảm (Được làm màu nổi bật).
        + Số lượng: người dùng có thể nhập số lượng sản phẩm muốn mua.
        + Nút bấm Chọn sản phẩm: Khi bấm vào sẽ thêm sản phẩm vào giỏ hàng (Được làm màu nổi bật để thu hút sự chú ý).
        + Mô tả sản phẩm: Mô tả chi tiết cách sử dụng và các đặc điểm của sản phẩm.

### 4.1.6. Màn hình giỏ hàng

* Khi chưa có hàng trong giỏ

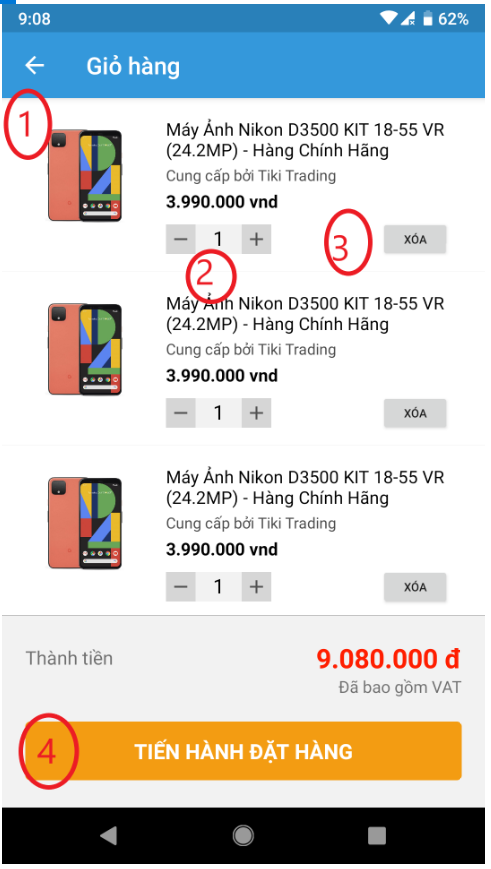


Khi chưa có mặt hàng nào trong giỏ hàng thì khi người dùng nhấn vào nút giỏ hàng thì ứng dụng sẽ hiện ra màn hình thông báo chưa có mặt hàng nào trong giỏ hàng, yêu cầu người dùng tiếp tục mua hàng.

* Điều hướng và tương tác người dùng:

(1): Sau khi nhấn vào nút Tiếp tục mua hàng thì ứng dụng sẽ đi đến màn hình trước đó để người dùng tiếp tục mua hàng.

* Khi trong giỏ hàng đã có sản phẩm



* Điều hướng và tương tác người dùng:

(1): Khi nhấn vào sẽ chuyển đến màn hình chi tiết sản phẩm.

(2): Chọn số lượng sản phẩm

(3): Khi nhấn vào sẽ xóa sản phẩm khỏi giỏ hàng

(4): Sau khi người dùng nhấn vào nút Tiến hành đặt hàng ứng dụng sẽ chuyển đến màn hình thông tin đặt hàng

* Tổng quan về màn hình:
  + - * + Hình ảnh mặt hàng: Khi nhấn vào sẽ chuyển đến màn hình chi tiết sản phẩm.
        + Tên mặt hàng: Khi nhấn vào sẽ chuyển đến màn hình chi tiết sản phẩm.
        + Tên nhà cung cấp.
        + Giá sản phẩm: Giá đã giảm (Được tô đậm để dễ quan sát).
        + Số lượng mỗi mặt hàng: Có nút tăng giảm để thay đổi số lượng sản phẩm muốn mua.
        + Nút xóa: Xóa sản phẩm khỏi giỏ hàng.
* Chi tiết tổng quan của giỏ hàng:
  + - * + Tổng tiền: Được làm size lớn, màu đỏ và tô đậm để nổi bật hơn.
        + Nút tiến hành đặt hàng: Khi nhấn vào sẽ chuyển đến màn hình thông tin đặt hàng.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Những công việc làm được

* Đã phân tích được hiện trạng hệ thống để đưa ra được sơ đồ phân cấp các chức năng của hệ thống mới, đưa ra được các biểu đồ mức ngữ cảnh, mức đỉnh và mức dưới đỉnh để phục vụ cho quá trình xây dựng hệ thống mới của nhà hàng.
* Thiết kế được mô hình cơ sở dữ liệu cho website.
* Thiết kế giao diện app thân thiện, dễ sử dụng cho người dùng.
* Thiết kế được giao diện giỏ hàng.

## 5.2. Hạn chế

* Do kinh nghiệm thực tế chưa có nhiều do vậy quá trình phân tích hệ thống cho app của cửa hàng còn nhiều chỗ chưa đúng với thực tế hay chưa đảm bảo tính đúng đắn. Vấn đề này em xin phép hoàn thiện thêm trong quá trình phát triển hệ thống sau đó.
* Giao diện app còn chưa được đẹp mắt.
* Ngôn ngữ và phần mềm soạn thảo là còn mới mẻ nên còn nhiều chức năng, chưa vận dụng và kiểm soát được.

## 5.3. Kết luận

* Với sự nỗ lực của các thành viên trong nhóm và sự giúp đỡ, chỉ bảo tận tình của **Thầy Huỳnh Tuấn Anh** hướng dẫn cuối cùng nhóm cũng hoàn thành xong đề tài. Tuy vậy,Với những thuận lợi và khó khăn trong quá trình làm việc, bài làm về cơ bản đã hoàn thành nhưng không thể tránh khỏi sai sót.

# Tài liệu tham khảo

[1] Slide bài giảng môn Lập trình trên thiết bị di động trên courses.uit.edu.vn

[2] Khóa học lập trình di động trên howkteam.vn

[3] Khóa học lập trình di động trên CodersX:

[3.1]

<https://www.youtube.com/watch?v=fF3yOUHXtKU&list=PLkY6Xj8Sg8-s-m-qFBQFoeNSfpCTBiwMU&fbclid=IwAR3G5bUAcHgUqdVXWX7jv1VFoQcXUV97HVE4vJNtcZwv-1kiDqtBIbHrs7Q>

[3.2]

<https://www.youtube.com/watch?v=Oo9MAd40lbI&list=PLkY6Xj8Sg8-uPZnTdScfuH0xD-O6Kb-V-&fbclid=IwAR1fdTJlOKHanSOHTVB1J43hgKkp_uY2V-piCuT1dQM03cuCvIq1yDp8f5g>