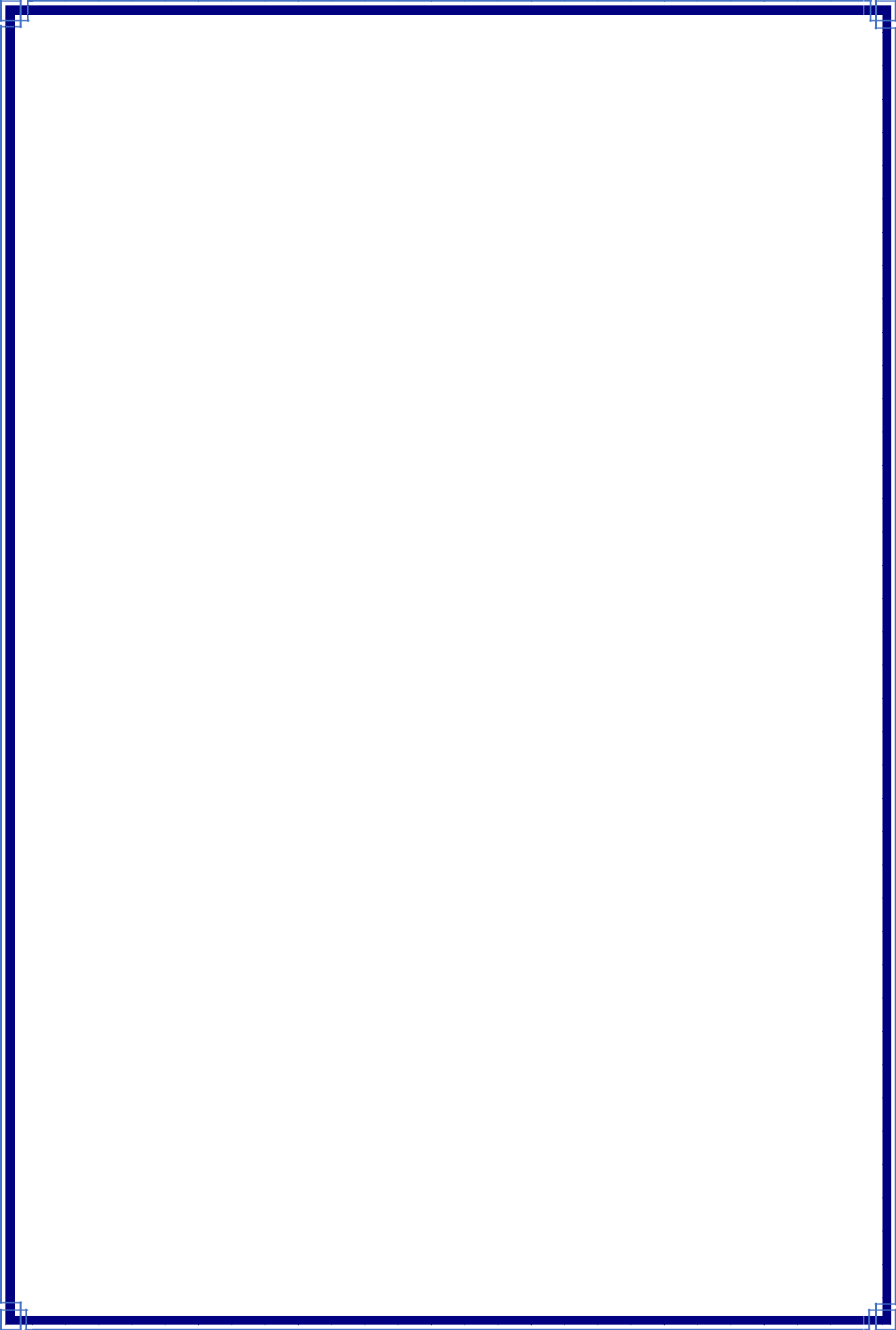
****

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

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

**Cơ Sở TP Hồ Chí Minh**

**----🙣🕮🙡----**



**BÁO CÁO TIẾN ĐỘ THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

**Đề tài: Xây dựng ứng dụng bán điện thoại**

**Giảng viên hướng dẫn**: **Nguyễn Thị Tuyết Hải**

**Sinh viên thực hiện: Lê Quang Hùng**

**Mã số sinh viên: N19DCCN070**

**Lớp: N19CQCNPM01-N**

**Khoá: 2019**

**Hệ: CHÍNH QUY**

*TP Hồ Chí Minh, tháng 7 năm 2023*

Muc luc

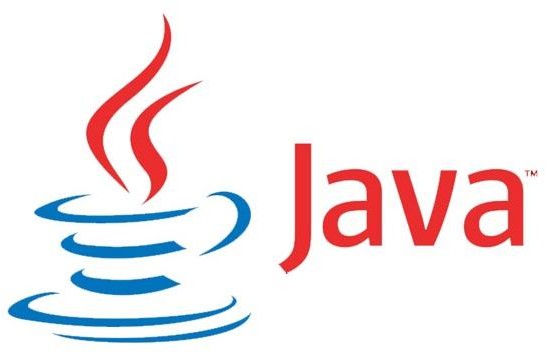
# 1. GIỚI THIỆU ĐỀ TÀI

* 1. **Tìm hiểu “Xây dựng ứng dụng bán điện thoại”:**

Trong thời công nghệ 4.0, điện thoại di động đang ngày càng trở nên phổ biến và là một phần không thể thiếu đối với cuộc sống của con người, hình thức mua hàng trên điện thoại di động cũng không là ngoại lệ. Hàng loạt các ứng dụng bán các mặt hàng phục vụ nhu cầu mua sắm được tạo ra và ngày càng phổ biến.

Dựa trên nhu cầu thực tế, tôi lựa chọn đề tài “Xây dựng ứng dụng bán điện thoại”. Phần mềm bán điện thoại được nhiều người bán và người mua sử dụng nhằm giúp cách mua và bán hàng dễ dàng và hiệu quả hơn. Chính những phần mềm bán điện thoại này cũng giúp tăng năng suất bán hàng lên gấp nhiều lần, dễ dàng quản lý, tăng năng suất làm việc, dễ dàng kiểm soát doanh thu, giúp tiết kiệm thời gian và chi phí.

* 1. **Nền tảng công nghệ:**
     + Website quản trị: Sử dụng HTML, CSS, Javascript, Framework ReactJS
     + Android App: Sử dụng Java, thư viện Glider, Retrofit..
     + RESTful Web Service: Sử dụng java, framework Spring Boot, Spring Security
  2. **Công cụ lập trình và công nghệ được sử dụng:**
     + Hệ quản trị cơ sở dữ liệu: SQL Server
     + Ngôn ngữ lập trình: Java, Javascript.
     + Xây dựng giao diện front-end: HTML, CSS, Javascript, ReactJS
     + Công nghệ xử lý back-end Java.
     + Framework Spring Boot, Spring MVC, Spring Security, ReactJS
     + Công cụ lập trình: Android studio, IntelliJ IDEA, Visual Studio Code.
  3. **Giới thiệu ngôn ngữ lập trình Java:**



*Hình 1. 1 Giới thiệu ngôn ngữ lập trình Java*

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak. Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó .

* 1. **Giới thiệu hệ quản trị cơ sở dữ liệu SQL SERVER:**

*Hình 1. 2 Giới thiệu hệ quản trị CSDL SQL Server*

- SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, database engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

- Năm 1989, phiên bản đầu tiên của SQL Server 1.0 ra đời được dùng cho các hệ điều hành 16 bit và được phát triển cho tới ngày nay. Cho đến ngày nay thì phiên bản mới nhất đó là SQL Server 2016 hỗ trợ bộ vi xử lý 6I.2 bit ra đời vào ngày 1 tháng 6 năm 2016.

* 1. **Các thư viện – framework hỗ trợ:**

### **React-Bootstrap** :

React-Bootstrap là một thư viện UI (User Interface) cho React được xây dựng dựa trên Bootstrap, một framework phổ biến cho thiết kế giao diện người dùng (UI) trên web. Thư viện này cho phép các nhà phát triển sử dụng các thành phần UI có sẵn của Bootstrap để xây dựng các ứng dụng React một cách nhanh chóng và dễ dàng. Với React-Bootstrap, bạn có thể tạo các trang web với các thành phần như buttons, forms, modals, navigation, alerts, và nhiều hơn nữa một cách đơn giản. Nó cũng có thể giúp bạn tạo ra các ứng dụng có thiết kế responsive (đáp ứng) và có khả năng tương thích trên các thiết bị khác nhau.

### **Chart.js:**

Chart.js là một thư viện JavaScript mã nguồn mở cho phép bạn tạo các biểu đồ và đồ thị tương tác trên trang web của bạn. Nó cung cấp các tính năng phức tạp và mạnh mẽ, bao gồm đồ thị cột, đồ thị đường, đồ thị vùng, biểu đồ tròn và nhiều hơn nữa.

Chart.js được phát triển bởi nhóm tác giả tại GitHub và được viết bằng JavaScript, vì vậy nó có thể dễ dàng tích hợp vào bất kỳ ứng dụng web nào. Nó cung cấp một cách linh hoạt để tạo ra các biểu đồ tương tác đẹp mắt mà không cần sử dụng các thư viện khác như jQuery.

Với Chart.js, bạn có thể tùy chỉnh các tính năng, màu sắc và kiểu dáng của biểu đồ của mình để phù hợp với nhu cầu của bạn. Nó cũng có thể tương tác với người dùng, ví dụ như khi họ di chuột qua các điểm dữ liệu để hiển thị các thông tin chi tiết. Chart.js cung cấp một API đơn giản và dễ hiểu, cho phép bạn tạo các biểu đồ tùy chỉnh với độ chính xác cao.

### **React-chartjs-2:**

React-Chartjs-2 là một thư viện phụ trợ cho React để hiển thị biểu đồ sử dụng thư viện Chart.js. Nó cho phép bạn dễ dàng tích hợp các biểu đồ phức tạp vào ứng dụng React của bạn một cách nhanh chóng và tiện lợi.

React-Chartjs-2 cung cấp một API để tạo các thành phần biểu đồ React được dựa trên các hình thức biểu đồ của Chart.js. Thư viện này cho phép bạn tùy chỉnh các biểu đồ của mình bằng cách cung cấp các thuộc tính và phương thức để thay đổi các tính năng, màu sắc và kiểu dáng của biểu đồ.

Với React-Chartjs-2, bạn có thể tạo các biểu đồ đường, biểu đồ cột, biểu đồ tròn, biểu đồ vùng và các loại biểu đồ khác. Bạn cũng có thể tạo ra các biểu đồ động với sự tương tác của người dùng, ví dụ như kéo thả hoặc phóng to/thu nhỏ.

### **Spring Security:**

Spring Security là một framework chuyện về bảo mật ứng dụng Java. Nó cung cấp các tính năng như xác thực và phân quyền truy cập, bảo vệ toàn vẹn dữ liệu, đăng nhập và đăng xuất người dùng, v.v.

Một ví dụ về sử dụng Spring Security là khi bạn muốn bảo vệ phần mềm của mình khỏi các cuộc tấn công từ bên ngoài, đồng thời cung cấp cho các người dùng truy cập vào phần mềm một cách an toàn và bảo mật.

### **Hibernate - JPA::**

Hibernate là một framework ORM (Object-Relational Mapping) được sử dụng để tương tác với cơ sở dữ liệu trong ứn g dụng Java. Hibernate cung cấp một cách thức thuận tiện để ánh xạ đối tượng của Java với các bảng trong cơ sở dữ liệu, giúp cho việc lập trình trở nên dễ dàng hơn và giảm thiểu việc phải viết nhiều mã lặp lại.

Hibernate được hỗ trợ rộng rãi bởi JPA (Java Persistence API), một đặc tả của Java EE (Enterprise Edition) được thiết kế để quản lý các đối tượng trong cơ sở dữ liệu. JPA cung cấp các chức năng tương tự như Hibernate bao gồm ánh xạ đối tượng và quản lý phiên đối tượng, đồng thời cung cấp các giao diện tự động cập nhật cơ sở dữ liệu.

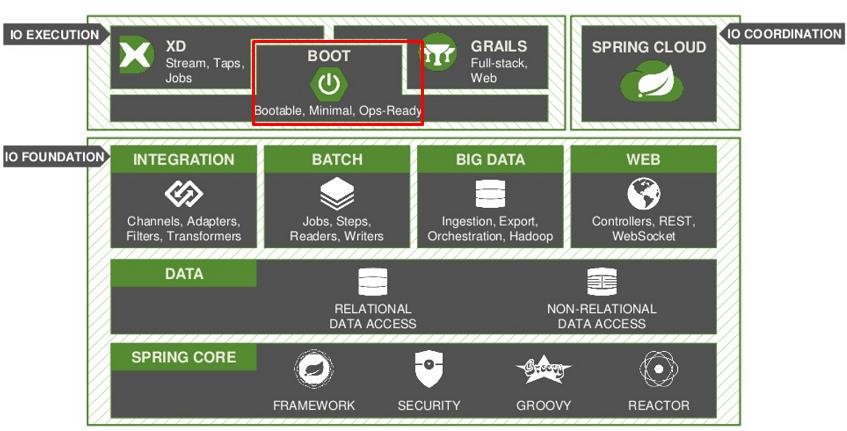
### **Lombok:**

Lombok là một thư viện Java được sử dụng để giảm thiểu mã lặp và giúp lập trình viên viết mã tối giản hơn. Nó cung cấp các chức năng cho phép người lập trình thêm các annotation cho lớp Java, giúp giảm thiểu code tiện ích và các phương thức getter/setter bằng cách sử dụng các annotation.

### **JavaScript:**

JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

### **Spring Boo**t:



*Hình 1. 3 Giới thiệu Spring Boot*

Spring Boot là một trong những thành viên của hệ sinh thái Spring framework. Nó giúp cho chúng ta đơn giản hóa công đoạn khởi tạo một dự án với Spring, thay vào đó các bạn chỉ tập trung vào việc phát triển business cho ứng dụng mà thôi. Dưới đây là một số ưu điểm của Spring Boot:

* + - * Tạo ứng dụng Spring độc lập, đóng gói dưới dạng một file JAR (stand- alone application).
      * Tối ưu công đoạn cấu hình cho ứng dụng Spring, không yêu cầu XML config…
      * Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR
      * Cung cấp nhiều plugin
      * Các starter dependency giúp việc cấu hình Maven đơn giản hơn

### **Spring MVC:**

Spring MVC là một trong những thành viên của hệ sinh thái Spring framework. Nó giúp các nhà phát triển viết các ứng dụng MVC nhanh hơn. Dưới đây là một số ưu điểm của Spring MVC:

Spring MVC là một framework MVC sử dụng một trình điều phối để gọi các phương thức trong controller và chuyển tiếp điều khiển đến một view (khung nhìn).

Spring MVC sử dụng tệp cấu hình bằng XML mà bạn có thể chỉnh sửa mà không cần biên dịch lại ứng dụng.

Spring MVC khởi tạo các lớp controller và gắn các bean với các đầu vào của người dùng.

Spring MVC tự động liên kết đầu vào của người dùng với đúng kiểu dữ liệu. Ví dụ, Spring MVC có thể tự động phân tích cú pháp một chuỗi và đặt thuộc tính kiểu float hoặc thập phân.

Spring MVC xác thực đầu vào của người dùng và chuyển hướng người dùng trở lại form đầu vào nếu xác thực không thành công. Xác thực đầu vào là tùy chọn và có thể được thực hiện theo chương trình hoặc theo khai báo. Trước đó, Spring MVC cung cấp trình xác thực tích hợp cho hầu hết các tác vụ bạn có thể gặp phải khi xây dựng một ứng dụng web.

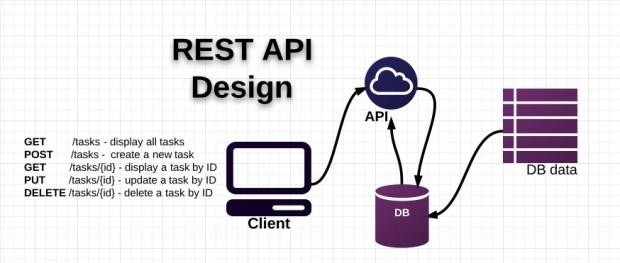
Spring MVC là một phần của Spring framework. Bạn nhận được mọi thứ mà Spring cung cấp.

Spring MVC hỗ trợ quốc tế hóa (internationalization) và bản địa hoá (localization). Điều này có nghĩa là bạn có thể hiển thị tin nhắn bằng nhiều ngôn ngữ tùy thuộc vào ngôn ngữ người dùng.

Spring MVC hỗ trợ nhiều công nghệ hiển thị (view). Hầu hết trường hợp bạn sẽ sử dụng JSP, nhưng các công nghệ khác, bao gồm Velocity và FreeMarker, cũng được hỗ trợ.

### **RESTful Web Service:**

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



*Hình 1. 4 Giới thiệu RESTful Web Service*

Diễn giải các thành phần:

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

REST (Representational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý

một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

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

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API. JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

### **Java Swing:**

Java Swing là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng window-based. Nó được xây dựng trên API AWT (Abstract Windowing Toolkit) và được viết hoàn toàn bằng Java.

Không giống như AWT, Java Swing cung cấp các thành phần không phụ thuộc vào nền tảng và nhẹ hơn.

Gói javax.swing cung cấp các lớp cho java swing API như JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser, v.v.

Java Swing được xem là một bộ công cụ Giao diện Người dùng Đồ họa (GUI) bao gồm các thành phần GUI

GUI (Giao diện người dùng đồ họa) trong Java là một trình tạo trải nghiệm trực quan dễ sử dụng cho các ứng dụng Java. Nó chủ yếu được làm bằng các thành phần đồ họa như nút, nhãn, cửa sổ, v.v. mà qua đó người dùng có thể tương tác với ứng dụng. GUI đóng một vai trò quan trọng để xây dựng giao diện dễ dàng cho các ứng dụng Java.

- Tính năng của Swing trong Java:

+ Độ tùy biến cao, các điều khiển xoay có thể được tùy chỉnh theo một cách rất dễ dàng và độc lập với biểu diễn bên trong.

+ Bộ điều khiển nâng cao phong phú như TabbedPane, Tree, thanh trượt, Table, và colorpicker.

+ Trọng lượng nhẹ, hầu như những thành phần trong Swing đều độc lập với API của hệ điều hành gốc do các điều khiển API Swing được kết xuất chủ yếu bằng cách sử dụng mã Java thuần túy thay vì các cuộc gọi hệ điều hành cơ bản.

### **JSON-Java:**

Thư viện JSON-Java còn được gọi là org.json cung cấp cho chúng ta các lớp được sử dụng để phân tích cú pháp và thao tác JSON trong Java.

Hơn nữa, thư viện này cũng có thể chuyển đổi giữa JSON, XML, Cookie, Mảng danh sách hoặc Văn bản được phân tách bằng dấu phẩy, v.v.

Tiện ích giúp dễ dàng lưu trữ các giá trị response dưới dạng JSON gửi về từ các request. Từ đó sử dụng những phương thức có sẳn để lấy giá trị của chúng, cũng như dễ dàng lưu vào ArrayList Java…

### **Android studio:**

Android Studio là một phần mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập emulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

*Hình 1. 5 Giới thiệu Android Studio*

Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân hoặc laptop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiết bị mobile (hoặc table) để tiến hành cài đặt ứng dụng và chạy kiểm thử (testing). Bạn thử tưởng tượng nếu với mỗi lần viết một dòng code bạn lại phải build ứng dụng, cài đặt trên điện thoại hoặc tablet và sau đó chạy thử thì sẽ vô cùng tốn thời gian và công sức.

Android Studio được phát triển để giải quyết các vấn đề này. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất..

# 2. KHẢO SÁT HIỆN TRẠNG

* 1. **Hiện trạng thực tế:**

Ngày nay khoa học công nghệ ngày càng phát triển, nhu cầu của con người ngày càng cao. Một trong những nhu cầu đó là việc mua bán hàng diễn ra thuận tiện Thị trường điện thoại di động phát triển mạnh mẽ, với nhu cầu mua bán điện thoại di động tăng cao. Việc xây dựng ứng dụng bán điện thoại di động giúp đáp ứng nhu cầu mua sắm của người dùng và khai thác tiềm năng kinh doanh trong lĩnh vực này. Với ứng dụng bán điện thoại di động, người dùng có thể mua điện thoại từ bất kỳ đâu, mọi lúc chỉ cần kết nối internet. Điều này mang lại sự tiện lợi và linh hoạt cho người dùng, không cần phải tốn thời gian và công sức đến các cửa hàng truyền thống. Ứng dụng bán điện thoại di động cung cấp thông tin chi tiết về sản phẩm, giúp người dùng nắm rõ thông tin kỹ thuật, đặc điểm và giá cả. Người dùng có thể thực hiện giao dịch nhanh chóng và dễ dàng mà không cần phải đến cửa hàng trực tiếp.

## 2.2 Phân tích yêu cầu người dùng:

### **Yêu cầu của khách hàng:**

* Giao diện dễ sử dụng và tính thẩm mỹ cao.
* Cho phép khách hàng đăng kí thành viên và đảm bảo tính bảo mật.
* Xem thông tin về tào khoản.
* Đặt hàng đơn giản nhanh chóng.
* Thông tin sản phẩm phong phú đa dạng.
* Luôn luôn cập nhật, giới thiệu những mẫu mới.
* Cho phép tìm kiếm sản phẩm nhanh chóng, chính xác theo tiêu chí.

### **Yêu cầu của quản lý:**

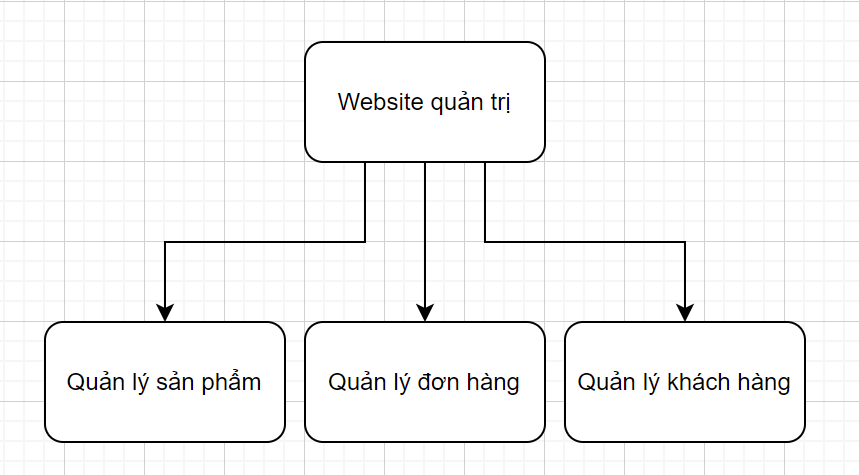
* Cập nhật thông tin hàng hoá trực tuyến :
* Dễ dàng cập nhật và thường xuyên thay đổi hình ảnh, thông tin chi tiết giá các mặt hàng bạn sẽ kinh doanh ở bất cứ đâu.
* Quản lý các đơn đặt hàng trực tuyến:
* Quản lý, lưu trữ các thông tin về đặt hàng và trạng thái của đơn hàng: đã giao hàng chưa, đã thanh toán chưa…
* Quản lý khách hàng:
* Lưu trữ các hoạt động gắn với các khách hàng.
* Mọi hoạt động gắn với khách hàng và đơn hàng đều có thể thực hiện từ xa, không phụ thuộc vào vị trí địa lý
  1. **Yêu cầu chức năng:**
     + Hệ thống chạy đúng, tốt, ổn định.
     + Giao diện thân thiện, dễ sử dụng.
     + Đáp ứng hầu hết các yêu cầu của người dùng.

**2.4 Yêu cầu hệ thống:**

Cung cấp phân quyền cho các loại người dùng có trách nhiệm: Quản lý, Khách hàng

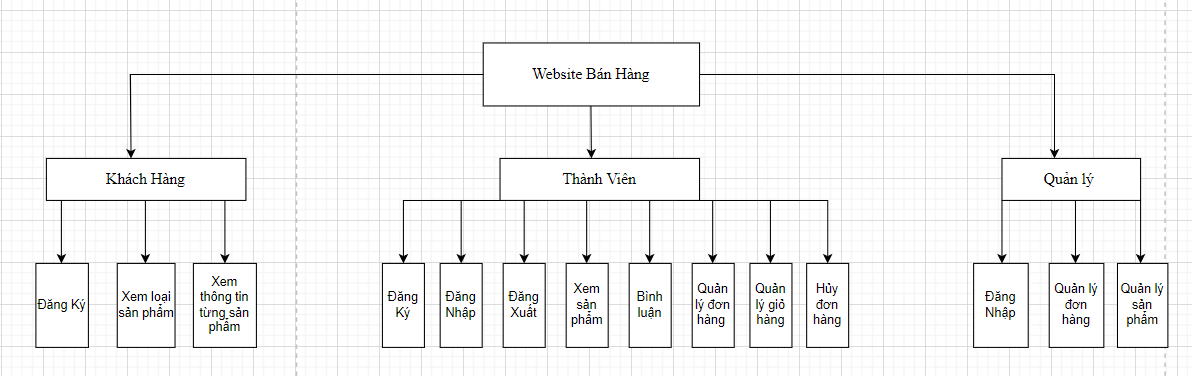
# 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Sơ đồ phân cấp chức năng:



*Sơ đồ 3. 1 Sơ đồ phân cấp chức năng*

## Sơ đồ phân rã chức năng:

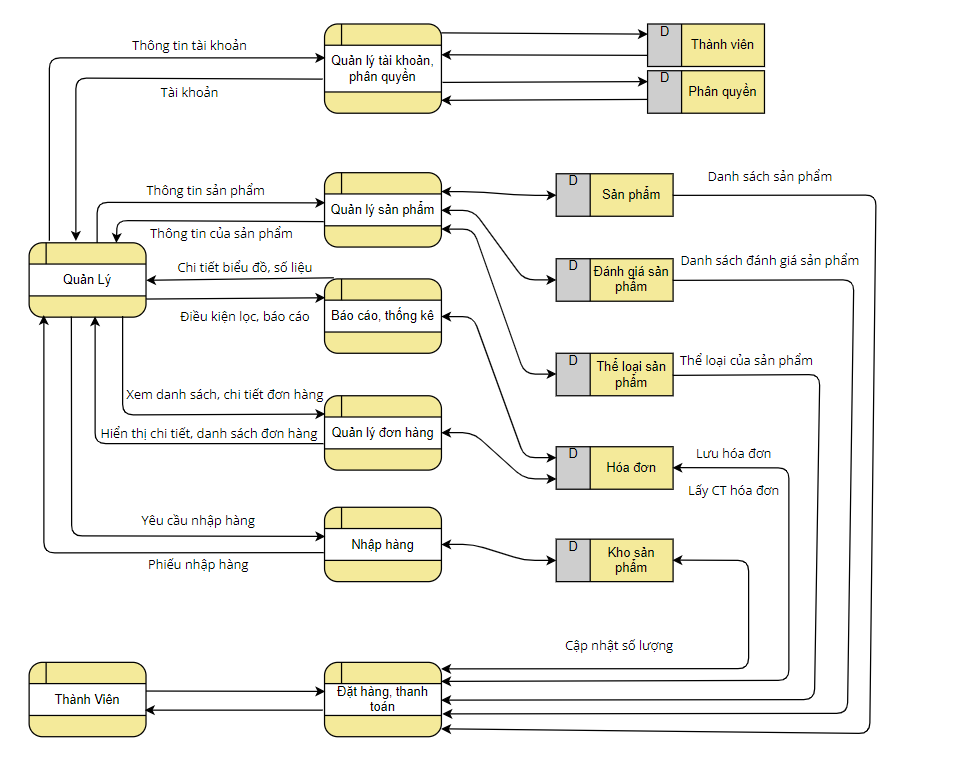
**

*Sơ đồ 3. 2 Sơ đồ phân rã chức năng*

* 1. **Mô hình luồng dữ liệu DFD (Data Flow Diagram)**

### **Biểu đồ DFD mức ngữ cảnh:**

*Sơ đồ 3. 3 Biểu đồ luồng dữ liệu mức ngữ cảnh*

1. **Mô hình luồng dữ liệu DFD phân rã ở mức 0 (chức năng chính):**

*Sơ đồ 3. 4 Biểu đồ luồng dữ liệu mức 0*

* 1. **Mô hình usecase:**

### **Xác định Actor:**

Dựa vào yêu cầu của bài toán, ta có các actor: khách hàng, thành viên, quản lý

### **Xác định các Use Case:**

Từ yêu cầu chức năng ứng với từng actor ta có thể xác định được các use case sau:

c

|  |  |
| --- | --- |
| **Actor** | **Use Case** |
| Khách hàng | Đăng ký, xem danh mục sản phẩm, xem thông tin từng sản phẩm |
| Thành viên | Đăng nhập, xem thông tin loại sản phẩm, xem thông tin từng sản phẩm, thêm sản phẩm vào giỏ hàng, xem danh sách sản phẩm trong giỏ hàng, xóa các sản phầm trong giỏ hàng, xác nhận đặt hàng và đặt hàng, xem thông tin của mình và đơn đặt hàng của mình. |
| Quản lý | Đăng nhập, quản lý sản phẩm: thêm, xóa, sửa sản phẩm, quản lý đơn hàng, quản lý tài khoản |

### **Xây dựng biểu đồ Use Case tổng quát:**

Actor Diagram

Description automatically generated:

* Khách hàng
* Thành viên
* Quản lí

Chức năng: Diagram

Description automatically generated with low confidence

* Xem danh mục sản phẩm
* Xem chi tiết sản phẩm
* Thêm sản phẩm
* Sửa sản phẩm
* Xóa sản phẩm
* Đăng kí
* Đăng nhập
* Quên mật khẩu
* Xem sản phẩm trong giỏ hàng
* Xóa sản phẩm khỏi giỏ hàng
* Đặt hàng
* Xem thông tin và đơn hàng

*Sơ đồ 3. 5 Biểu đồ use case tổng quát*

### **Đặc tả use case:**

**Bảng 1: Xem danh mục sản phẩm**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Xem danh mục sản phẩm |
| Actor | Quản lí/ Thành viên/ Khách hàng |
| Mô tả | Use case này mô tả việc xem danh mục sản phẩm của hệ thống |
| Điều kiện kích hoạt | Khi actor chọn chức năng xem danh mục sản phẩm từ hệ thống |
| Hậu điền kiện | Actor xem được danh sách sản phẩm của hệ thống |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng xem danh mục trên hệ thống  2. Hiển thị danh mục sản phẩm đang có trong hệ thống  3. Kết thúc use-case. |

**Bảng 2: Xem chi tiết sản phẩm**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Xem chi tiết sản phẩm |
| Actor | Quản lí/ Thành viên/ Khách hàng |
| Mô tả | Use case này mô tả việc xem chi tiết sản phẩm của hệ thống |
| Điều kiện kích hoạt | Khi actor chọn sản phẩm |
| Hậu điền kiện | Actor xem được chi tiết sản phẩm của hệ thống |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng xem sản phẩm  2. Hiển thị chi tiết sản phẩm  3. Kết thúc use-case. |

**Bảng 3: Thêm sản phẩm**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Thêm sản phẩm |
| Actor | Quản lí |
| Mô tả | Use case này mô tả việc thêm sản phẩm vào hệ thống |
| Điều kiện kích hoạt | Khi actor chọn chức năng thêm sản phẩm từ hệ thống |
| Tiền điều kiện | Actor đăng nhập vào hệ thống |
| Hậu điền kiện | Actor thêm được sản phẩm vào hệ thống |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng thêm sản phẩm trên hệ thống 2. Hiển thị biểu mẫu để thêm sản phẩm đang có vào hệ thống 3. Actor nhập thông tin cho sản phẩm mới, hệ thống sẽ kiểm tra tính đúng đắn. 4. Hệ thống kiểm tra thông tin, nếu mọi thứ đều hợp lệ thì sẽ thêm sản phẩm vào hệ thống 5. Kết thúc use-case. |

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Sửa sản phẩm |
| Actor | Quản lí |
| Mô tả | Use case này mô tả việc sửa sản phẩm của hệ thống |
| Điều kiện kích hoạt | Khi actor chọn chức năng sửa sản phẩm từ hệ thống |
| Tiền điều kiện | Actor đăng nhập vào hệ thống |
| Hậu điền kiện | Actor sửa được sản phẩm |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng sửa sản phẩm  2. Hiển thị sản phẩm cần sửa  3. Hệ thống sẽ hiển thị biểu mẫu để actor chỉnh sửa  4. Actor nhập thông tin vào trường cần thay đổi dữ liệu. Hệ thống sẽ kiểm tra tính đúng đắn và cập nhật sản phẩm  5. Kết thúc use-case. |

**Bảng 4: Sửa sản phẩm:**

**Bảng 5: Xóa sản phẩm:**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Xóa sản phẩm |
| Actor | Quản lí |
| Mô tả | Use case này mô tả việc xóa sản phẩm |
| Điều kiện kích hoạt | Khi actor chọn chức năng xóa sản phẩm |
| Tiền điều kiện | Actor đăng nhập vào hệ thống |
| Hậu điền kiện | Actor xóa được sản phẩm của hệ thống |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng xóa sản phẩm 2. Sau đó, actor chọn sản phẩm cần xóa và xác nhận. 3. Hệ thống tìm sản phẩm và xóa trong database 4. Kết thúc use-case. |

**Bảng 6: Đăng ký:**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Đăng kí |
| Actor | Khách hàng |
| Mô tả | Use case này mô tả việc đăng kí tài khoản vào hệ thống |
| Điều kiện kích hoạt | Khi actor chọn chức năng đăng kí trên trang chủ hệ thống |
| Hậu điền kiện | Actor có tài khoản để vào đăng nhập vào hệ thống |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng đăng kí 2. Nhập tên đăng nhập, password 3. Kết thúc use-case. |

**Bảng 7: Đăng nhập:**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Đăng nhập |
| Actor | Quản lí, thành viên |
| Mô tả | Use case này mô tả đăng nhập hệ thống |
| Điều kiện kích hoạt | Khi actor chọn chức năng đăng nhập từ trang chủ hệ thống |
| Tiền điều kiện | Actor có tài khoản trên hệ thống |
| Hậu điền kiện | Actor đăng nhập thành công |
| Luồng sự kiện chính | 1. Hệ thống yêu cầu nhập thông tin gồm tài khoản và mật khẩu để đăng nhập 2. Người dùng nhập thông tin để đăng nhập vào tài khoản. 3. Hệ thống kiểm tra tài khoản hợp lệ và thông báo thành công/thất bại cho actor 4. Kết thúc use-case. |
| Luồng sự kiện phụ | Trường hợp 1: Đăng nhập thành công:   1. Nếu đăng nhập thành công hệ thống dựa trên thông tin đăng nhập sẽ đồng thời phân quyền tùy theo loại nhân viên.   Trường hợp 2: Đăng nhập thất bại:   1. Người dùng có thể đăng nhập lại hoặc đăng ký tại trang. |

**Bảng 8 : Quên mật khẩu:**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Quên mật khẩu |
| Actor | Quản lí, thành viên |
| Mô tả | Use case này mô tả việc quên mật khẩu của hệ thống |
| Điều kiện kích hoạt | Khi actor quên mật khẩu đăng nhập vào hệ thống |
| Tiền điều kiện | Khi actor có tài khoản trong hệ thống |
| Hậu điền kiện | Actor thay đổi mật khẩu thành công |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng quên mật khẩu trên trang đăng nhập 2. Hiển thị biểu mẫu cho người dùng nhập email để nhận mail mã xác thực đổi mật khẩu 3. Người dùng nhập đúng mã xác thực 4. Người dùng nhập mật khẩu mới để đăng nhập vào hệ thống 5. Kết thúc use-case. |
| Luồng sự kiện phụ |  |

**Bảng 9: Xem sản phẩm trong giỏ hàng**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Xem sản phẩm trong giỏ hàng |
| Actor | Thành viên |
| Mô tả | Use case này mô tả việc xem sản phẩm trong giỏ hàng |
| Điều kiện kích hoạt | Khi actor chọn chức năng xem giỏ hàng từ hệ thống |
| Tiền điều kiện | Actor đăng nhập vào hệ thống |
| Hậu điền kiện | Actor xem được sản phẩm trong giỏ hàng |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng xem giỏ hàng 2. Hiển thị sản phẩm đang có trong giỏ hàng 3. Kết thúc use-case. |

**Bảng 10: Xóa sản phẩm trong giỏ hàng**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Xóa sản phẩm khỏi giỏ hàng |
| Actor | Thành viên |
| Mô tả | Use case này mô tả việc xóa sản phẩm trong giỏ hàng |
| Điều kiện kích hoạt | Khi actor chọn chức năng xóa sản phẩm từ giỏ hàng |
| Tiền điều kiện | Actor đăng nhập vào hệ thống |
| Hậu điền kiện | Actor xóa được sản phẩm trong giỏ hàng |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng xóa sản phẩm trong giỏ hàng 2. Sau đó, chọn sản phẩm cần xóa 3. Sản phẩm được xóa trong database 4. Kết thúc use-case. |

**Bảng 11: Đặt hàng**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Đặt hàng |
| Actor | Thành viên |
| Mô tả | Use case này mô tả việc đặt hàng của hệ thống |
| Điều kiện kích hoạt | Khi actor chọn chức năng đặt hàng từ hệ thống |
| Tiền điều kiện | Actor là thành viên |
| Hậu điền kiện | Actor đặt hàng trên hệ thống |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng giỏ hàng 2. Actor chọn chức năng đặt hàng   3. Kết thúc use-case. |

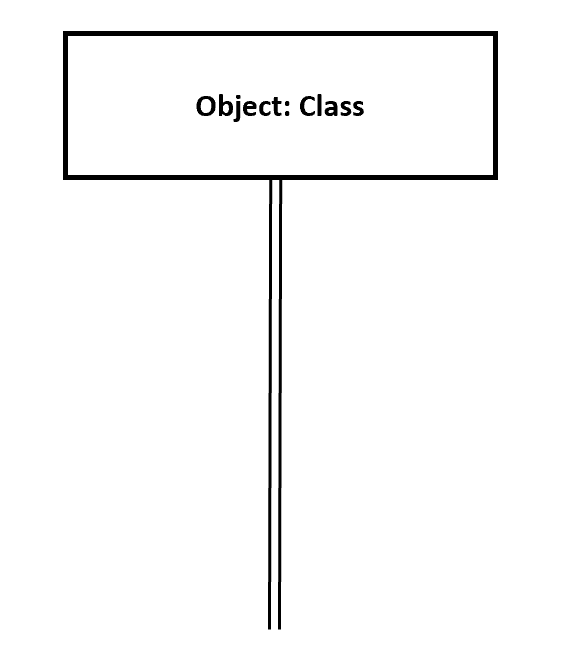
**Bảng 12: Xem thông tin và đơn hàng của họ**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên use-case | Xem thông tin và đơn hàng |
| Actor | Thành viên |
| Mô tả | Use case này mô tả việc xem thông tin đơn hàng |
| Điều kiện kích hoạt | Khi actor chọn chức năng xem thông tin đơn hàng |
| Tiền điều kiện | Actor đăng nhập vào hệ thống |
| Hậu điền kiện | Actor xem được thông tin đơn hàng |
| Luồng sự kiện chính | 1. Actor nhấn vào chức năng xem thông tin đơn hàng 2. Hiển thị thông tin của đơn hàng 3. Kết thúc use-case. |

### **Sơ đồ tuần tự:**

* Các thành phần của sơ đồ tuần tự
  + A picture containing rectangle

    Description automatically generatedĐối tượng (object or class): Biểu diễn bằng các hình chữ nhật
  + Đường đời đối tượng (Lifelines): Biểu diễn bằng các đường gạch thẳng đứng bên dưới đối tượng

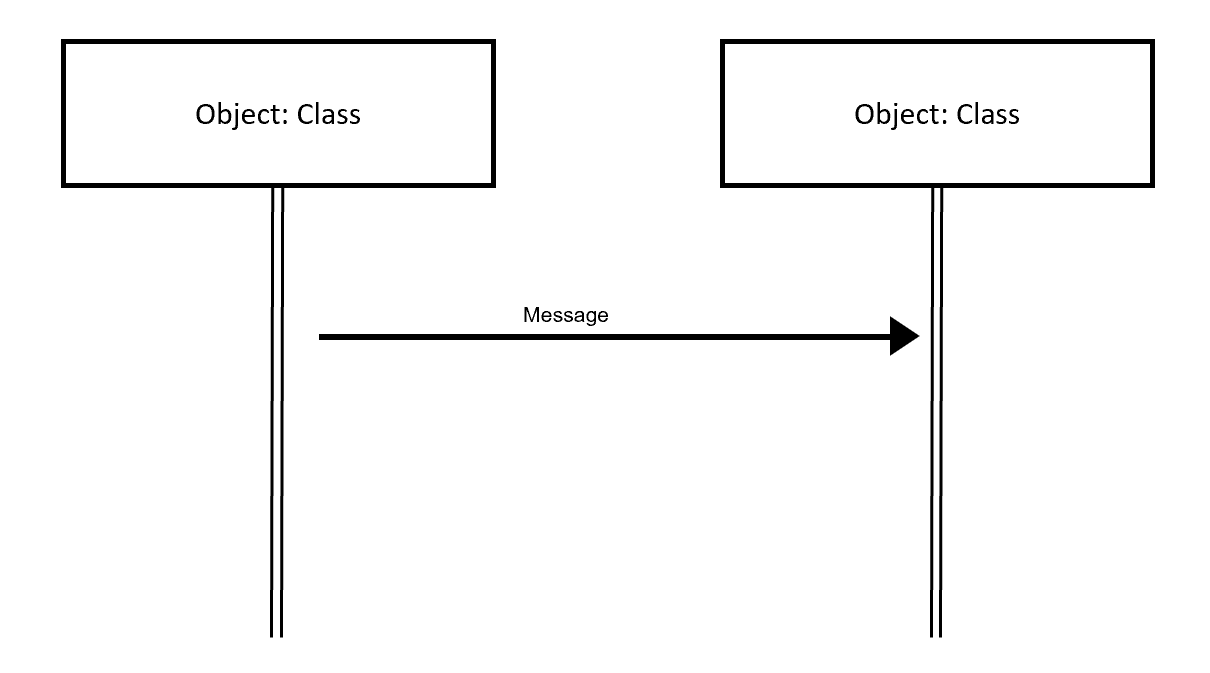


* + Thông điệp (Message) biểu diễn bằng các đường mũi tên
    - Có nhiều loại thông điệp:
      * Thông điệp đồng bộ: cần có 1 request trước hành động tiếp theo:

Text

Description automatically generated with medium confidence

* Thông điệp không đồng bộ: không cần có một request trước hành động tiếp theo



Rectangle

Description automatically generated with medium confidence

* Thông điệp chính mình: Là thông điệp mà đối tượng gửi cho chính nó để thực hiện các hàm nội tại

Shape, arrow

Description automatically generated

* Thông điệp trả lời hoặc trả về: Là thông điệp trả lời lại khi có request hoặc sau khi kiểm tra tính đúng đắn của một điều kiện nào đó này tin nhắn trả về là success hoặc fail

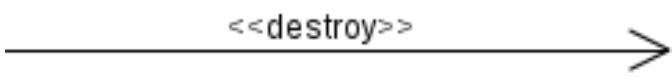


* Thông điệp tạo mới: Là thông điệp được trả về khi tạo mới một đối tượng

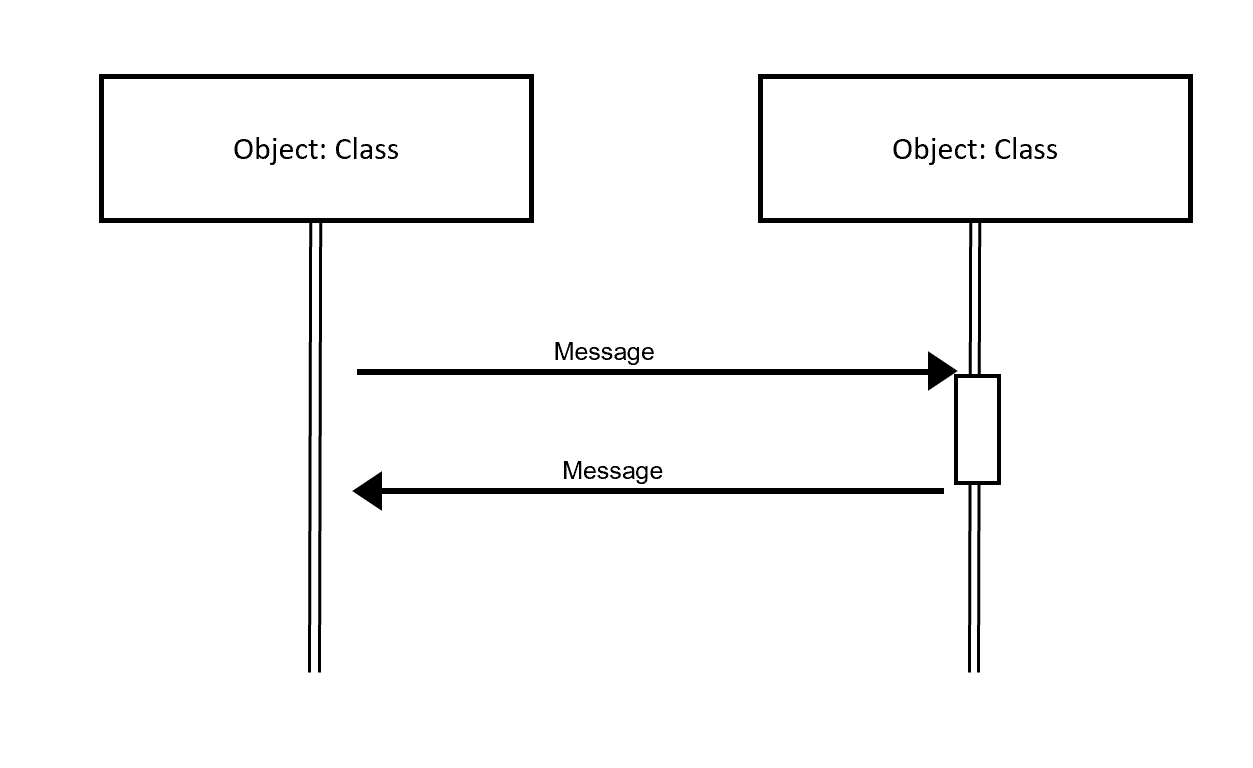
Text

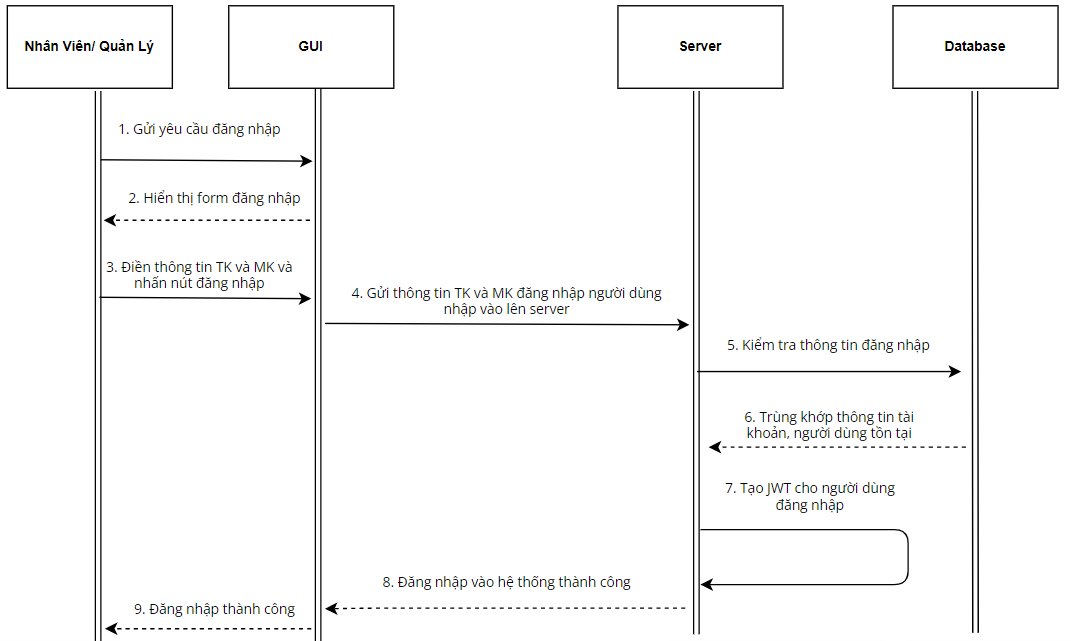
Description automatically generated with medium confidence

* Thông điệp xóa: Là thông điệp được trả lời khi xóa một đối tượng

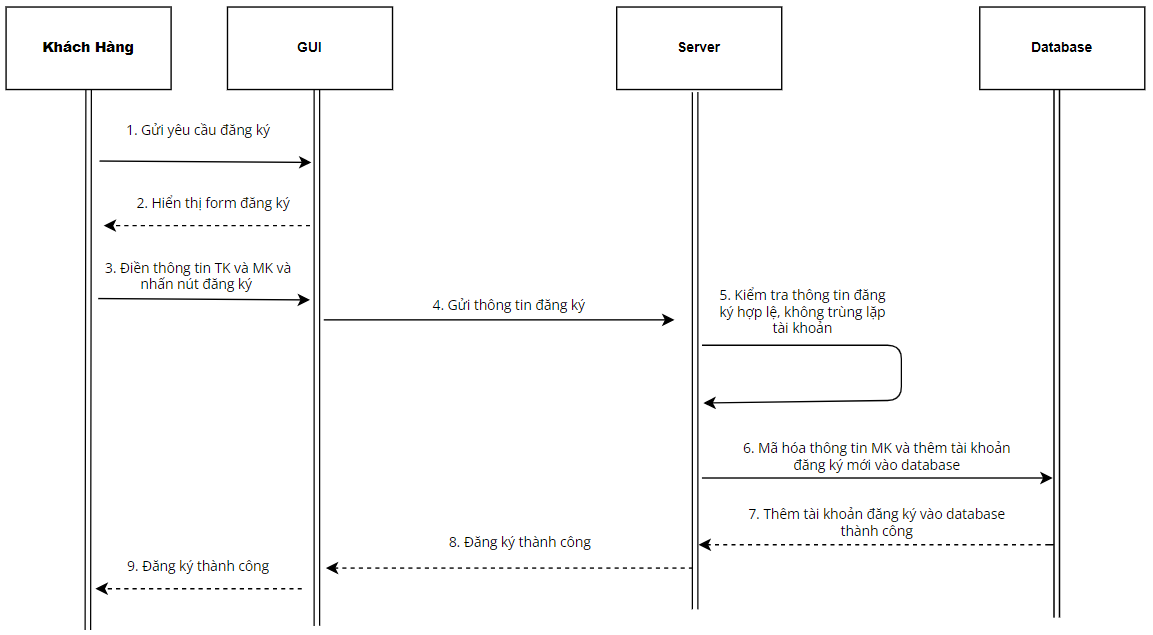


* Xử lí bên trong đối tượng (biểu diễn bằng các đoạn hình chữ nhật rỗng nối với các đường đời đối tượng

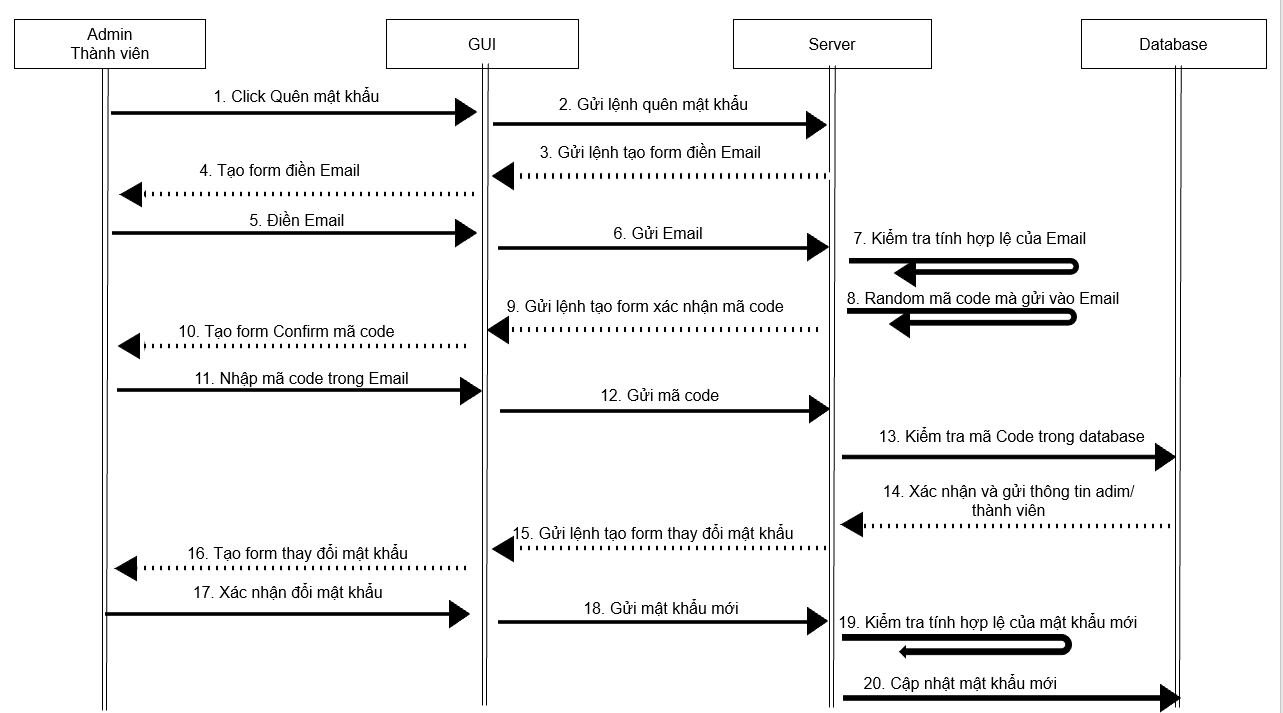


* **Đăng nhập**

*Sơ đồ 3. 6 Biểu đồ trình tự đăng nhập*

* ****Đăng Ký**

*Sơ đồ 3. 7 Biểu đồ trình tự đăng ký*

* ****Quên mật khẩu:**

*Sơ đồ 3. 8 Biểu đồ trình tự quên mật khẩu*

* *Diagram

  Description automatically generated***Xem chi tiết sản phẩm**

*Sơ đồ 3. 9 Biểu đồ trình tự xem chi tiết sản phẩm*

* *Diagram

  Description automatically generated***Xem danh mục sản phẩm:**

*Sơ đồ 3. 10 Biểu đồ trình tự xem danh mục sản phẩm*

* *Diagram

  Description automatically generated***Tìm kiếm sản phẩm:**

*Sơ đồ 3. 11 Biểu đồ tuần tự tìm kiếm sản phẩm*

* Diagram

  Description automatically generated**Xem thông tin cá nhân:**

*Sơ đồ 3. 12 Biểu đồ tuần tự xem thông tin cá nhân*

* *Graphical user interface, diagram

  Description automatically generated***Thêm sản phẩm vào giỏ hàng**

*Sơ đồ 3. 13 Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng*

* **Graphical user interface, application

  Description automatically generated Xóa sản phẩm khỏi giỏ hàng:**

*Sơ đồ 3. 14 Biểu đồ tuần tự xóa sản phẩm khỏi giỏ hàng*

* **Graphical user interface, diagram

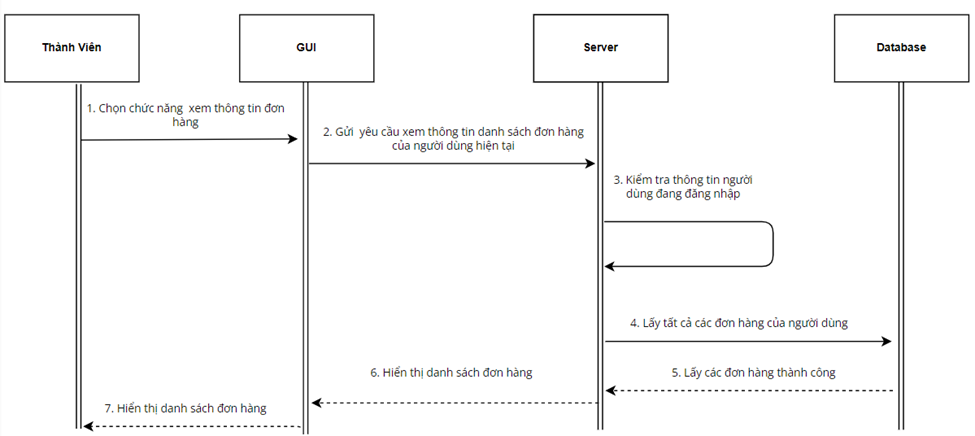
  Description automatically generatedXem các sản phẩm có trong giỏ hàng**

*Sơ đồ 3. 15 Biểu đồ tuần tự xem sản phẩm trong giỏ hàng*

* **Graphical user interface, diagram

  Description automatically generatedĐặt hàng:**

*Sơ đồ 3.16 Biểu đồ tuần tự đặt hàng*

* **Xem thông tin đơn hàng:**

*Sơ đồ 3. 17 Biểu đồ tuần tự xem đơn hàng*

* **A picture containing diagram

  Description automatically generatedThêm sản phẩm:**

*Sơ đồ 3. 18 Biểu đồ tuần tự thêm sản phẩm*

* **Diagram

  Description automatically generated with medium confidenceChỉnh sửa sản phẩm:**

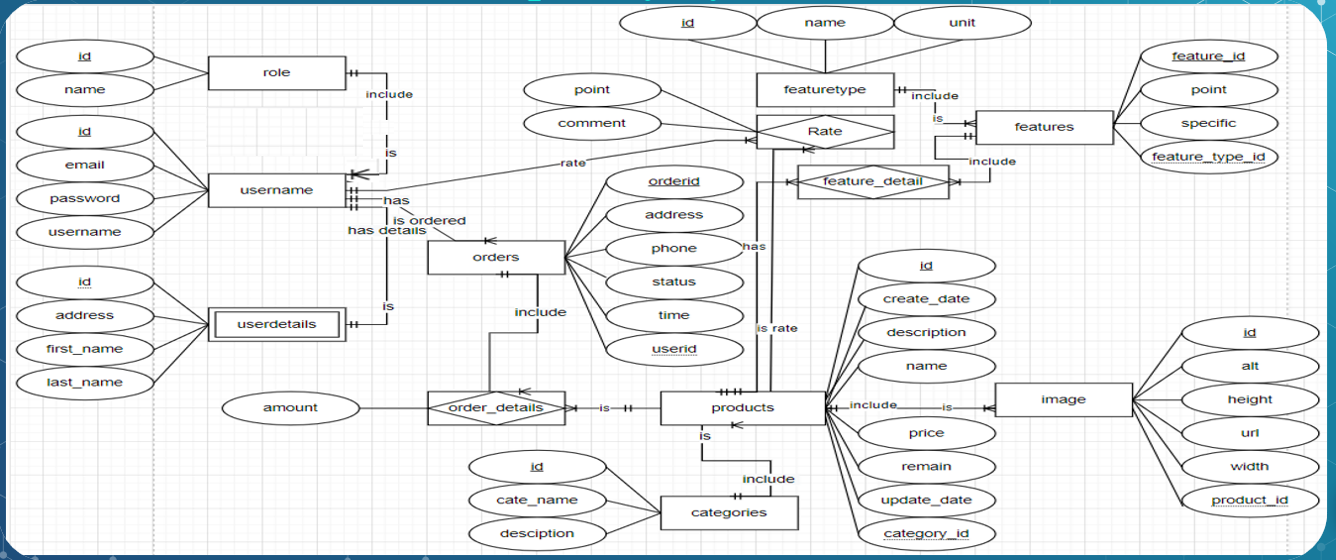
*Sơ đồ 3. 19 Biểu đồ tuần tự chỉnh sửa sản phẩm*

* **Xóa sản phẩm:**

**Diagram

Description automatically generated**

*Sơ đồ 3. 20 Biểu đồ tuần tự xóa sản phẩm*

* 1. **Xác định thực thể:**
  + Username(id, username, password, email).
  + Roles(id, name).
  + Features(feature\_id, point, specific).
  + FeatureType(id, name, unit).
  + Image(id, alt, height, url, width).
  + Products(id, name, price, description, create\_date, remain, update\_date).
  + Categories(id, name, description).
  + Orders(orderid, address, phone, status, time).
  + Rates(comment, point, sum\_price).
  1. **Mô hình ERD**:

*Hình 3. 21 Mô hình ERD*

1. **Mô hình cơ sở dữ liệu quan hệ:**

**Quy ước: Primary key:** (gạch chân) MABP, **Foreign key :** (in đậm) **MABP PK\_FK:** (vừa thuộc khóa chính, vừa là khóa ngoại, gạch chân) **MAPD, MAHP**

* + Username (id, username, password, email).
  + UserDetails(id, address, first\_name, last\_name).
  + User\_Roles(**user\_id, role\_id**).
  + Roles(id, name).
  + Features(feature\_id, point, specific, **features\_type\_id**).
  + Feature\_detail(**product\_id, feature\_id**)
  + FeatureType(id, name, unit).
  + Image(id, alt, height, url, width, **product\_id**).
  + Products(id, name, price, description, create\_date, remain, update\_date, **category\_id**).
  + Categories(id, name, description).
  + Orders(order\_id, address, phone, status, time, **user\_id**).
  + OrderDetails(**product\_id**, **order\_id**, amount).
  + Rates(comment, point, sum\_price, **product\_id, user\_id**).
* Thực thể Username: Thông tin user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | id | id | Bigint | Khóa chính | X |
| 2 | username | Tên đăng nhập | varchar(255) |  | X |
| 3 | password | Mật khẩu | varchar(255) |  | X |
| 4 | email | Email | varchar(255) |  | X |

*Bảng 3. 1 Bảng thực thể UserName*

* Thực thể User\_Roles: **user\_id + role\_id = Khóa chính**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | User\_id | Id User | bigint | Khóa ngoại | X |
| 2 | Role\_id | Id Role | bigint | Khóa ngoại | X |

*Bảng 3. 2 Bảng thực thể User\_Roles*

* Thực thể Roles: Thông tin các quyền

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | id | id | int | Khóa chính | X |
| 2 | name | Tên quyền | varchar(45) |  | X |

*Bảng 3. 3 Bảng thực thể Roles*

* Thực thể UserDetails: Thông tin chi tiết của User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | id | id | bigint | Khóa ngoại | X |
| 2 | address | Địa chỉ giao hàng của User | varchar(255) |  | X |
| 3 | first\_name | Họ của user | varchar(255) |  |  |
| 4 | last\_name | Tên của user | varchar(255) |  | X |

*Bảng 3. 4 Bảng thực thể UserDetails*

* Thực thể Features: Thông tin tính năng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | Feature\_id | Id | bigint | Khóa chính | X |
| 2 | Point | Điểm đánh giá | Bignit |  | X |
| 3 | Specific | Tính năng cụ thể | varchar (255) |  |  |
| 4 | Feature\_type\_id | Id loại tính năng | TINYINT(1) |  | X |

*Bảng 3. 5 Bảng thực thể Feature*

* Thực thể Feature\_detail: Chi tiết tính năng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | Product\_id | Id sản phẩm | bigint | Khóa ngoại | X |
| 2 | Feature\_id | Id tính năng | bigint | Khóa ngoại | X |

*Bảng 3. 6 Bảng thực thể Areas*

* Thực thể Feature\_Type: Loại tính năng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | id | Id loại tính năng | int | Khóa ngoại | X |
| 2 | Name | Tên loại tính năng | varchar(225) | Khóa ngoại | X |
| 3 | Unit | Số lượng | varchar(225) |  | X |

*Bảng 3. 7 Bảng thực thể Feature\_Type*

* Thực thể Product: Thông tin sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | id | Id sản phẩm | Bigint | Khóa chính | X |
| 2 | name | Tên sản phẩm | varchar(255) |  | X |
| 3 | price | Giá sản phẩm | Numeric(19,2) |  | X |
| 4 | description | Mô tả sản phẩm | varchar(255) |  |  |
| 5 | id\_category | Id danh mục sản phẩm | Bigint | Khóa ngoại | X |
| 6 | Remain | Số lượng tồn | Bigint |  |  |
| 7 | Update\_date | Ngày cập nhật | Datetime2(7) |  |  |
| 8 | Create\_date | Ngày tạo | Datetime2(7) |  |  |

*Bảng 3. 8 Bảng thực thể Product*

* Thực thể Categories: Thông tin chi tiết sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | id | Id danh mục sản phẩm | Bigint | Khóa chính | X |
| 2 | Cate\_name | Tên danh mục sản phẩm | varchar(225) |  |  |
| 3 | Description | Mô tả danh mục sản phẩm | varchar(225) |  |  |

*Bảng 3. 9 Bảng thực thể Categories*

* Thực thể Image: Thông tin hình ảnh sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | id | Id hình ảnh | Bigint | Khóa chính | X |
| 2 | Alt | Thể mô tả | varchar(225) |  |  |
| 3 | Height | Chiều cao của ảnh | Bigint |  |  |
| 4 | Url | Địa chỉ của ảnh | varchar(225) |  |  |
| 5 | Width | Độ rộng ảnh | varchar(225) |  |  |
| 6 | Product\_id | Id sản phẩm | Bigint | Khóa ngoại | X |

*Bảng 3. 10 Bảng thực thể Image*

* Thực thể Orders: Thông tin đơn hàng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | orderid | Id đơn hàng | Bigint | Khóa chính | X |
| 2 | Address | Địa chỉ giao hàng | varchar(225) |  | X |
| 3 | Phone | Số điện thoại | varchar(225) |  | X |
| 4 | Status | Trạng thái đơn hàng | varchar(225) |  | X |
| 5 | Time | Ngày giao hàng | Datetime2(7) |  | X |
| 6 | User\_id | Id người đặt hàng | Bigint | Khóa ngoại | X |

*Bảng 3. 11 Bảng thực thể Orders*

* Thực thể OrderDetails: Chi tiết đơn hàng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | Product\_id | Id sản phẩm | Bigint | Khóa phụ | X |
| 2 | Order\_id | Id đơn hàng | Bigint | Khóa phụ | X |
| 3 | Amount | Số lượng sản phẩm | Int |  | X |

*Bảng 3. 12 Bảng thực thể OrderDetails*

* Thực thể Rates: Thông tin đánh giá sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | Product\_id | Id sản phẩm | Bigint | Khóa ngoại | X |
| 2 | User\_id | Id người dùng | Bigint | Khóa ngoại | X |
| 3 | Comment | Lời đánh giá | Varchar(225) |  | X |
| 4 | Point | Điểm sản phẩm | Bigint |  | X |

*Bảng 3. 13 Bảng thực thể Rates*

## Diagram:

*Bảng 3. 14 Diagram*