**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC**

**CHUYÊN ĐỀ J2EE**

**SE325.H21**

**ĐỀ TÀI**

**XÂY DỰNG HỆ THỐNG BÁN CÁC THIẾT BỊ   
LUYỆN TẬP THỂ DỤC THỂ THAO**

**GV hướng dẫn*:* Th.S Nguyễn Trác Thức**

**Thực hiện bởi: Nhóm 12**

**Phạm Văn Huy – 12520176**

**Lê Nguyễn Quý – 13520683**

**Vũ Đức Huy - 12520179**

**Tp. Hồ Chí Minh, ngày 03 tháng 06 năm 2017**

# 

**MỤC LỤC**

[Phần 1: Giới thiệu 5](#_Toc484206718)

[1. Giới thiệu tổng quan về Java EE 5](#_Toc484206719)

[1.1 J2EE là gì? 5](#_Toc484206720)

[1.2 Kiến trúc của J2EE 5](#_Toc484206721)

[2. Giới thiệu về project sẽ phát triển 8](#_Toc484206722)

[2.1 Ý tưởng 8](#_Toc484206723)

[2.2 Các tính năng có trong project 9](#_Toc484206724)

[Phần 2: Công nghệ 9](#_Toc484206725)

[1. JSP là gì? 9](#_Toc484206726)

[2. Servlet là gì? 9](#_Toc484206727)

[3. JSP Hoạt Động Như Thế Nào 10](#_Toc484206728)

[4. Tính Năng Của JSP 10](#_Toc484206729)

[5. Tại Sao Sử Dụng JSP 10](#_Toc484206730)

[Phần 3: Thiết kế – Cài đặt 11](#_Toc484206731)

[1. Kiến trúc hệ thống 11](#_Toc484206732)

[1.1 MVC là gì? 11](#_Toc484206733)

[Tại sao cần mô hình MVC? 12](#_Toc484206734)

[1.2 13](#_Toc484206735)

[2. Sơ đồ use-case 13](#_Toc484206736)

[2.1. Sơ đồ use-case tổng quát cho cả hệ thống: 13](#_Toc484206737)

[a. Sơ đồ 13](#_Toc484206738)

[b. Mô tả: 13](#_Toc484206739)

[2.2 Sơ đồ use-case đăng nhập 15](#_Toc484206740)

[a. Sơ đồ 15](#_Toc484206741)

[b. Mô tả 15](#_Toc484206742)

[2.2 Sơ đồ use-case xem sản phẩm 16](#_Toc484206743)

[a. Sơ đồ 16](#_Toc484206744)

[b. Mô tả 16](#_Toc484206745)

[2.2 Sơ đồ use-case thanh toán 17](#_Toc484206746)

[a. Sơ đồ 17](#_Toc484206747)

[b. Mô tả 17](#_Toc484206748)

[2.2 Sơ đồ use-case quản lý sản phẩm 18](#_Toc484206749)

[a. Sơ đồ 18](#_Toc484206750)

[b. Mô tả 18](#_Toc484206751)

[3. Sơ đồ sequence 19](#_Toc484206752)

[3.1 Sơ đồ 19](#_Toc484206753)

[3.2 Mô tả 19](#_Toc484206754)

[4. Giao diện ứng dụng 20](#_Toc484206755)

[Phần 4: Kết luận – Hướng phát triển 20](#_Toc484206756)

[1. Kết luận 20](#_Toc484206757)

[1.1 Môi trường phát triển 20](#_Toc484206758)

[1.2 Kết luận 20](#_Toc484206759)

[2. Hướng phát triển 21](#_Toc484206760)

[3. Phụ lục: Hướng dẫn cài đặt và sử dụng 21](#_Toc484206761)

[4. Tài liệu tham khảo 21](#_Toc484206762)

[BẢNG PHÂN CÔNG TRONG QUÁ TRÌNH THỰC HIỆN 21](#_Toc484206763)

**ĐÁNH GIÁ CỦA GIÁO VIÊN**

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

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

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

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

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

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

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

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

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

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

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

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

# Phần 1: Giới thiệu

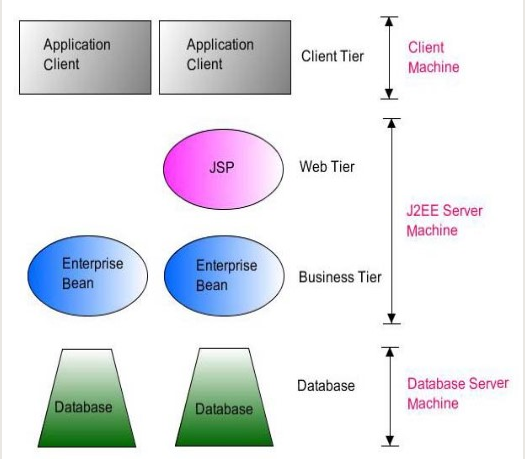
# 1. Giới thiệu tổng quan về Java EE

1.1 J2EE là gì?

Java 2 Platform Enterprise Edition viết tắt là J2EE, định nghĩa một chuẩn để phát triển những ứng dụng thương mại đa tầng (multitier enterprise applications). J2EE platform chứa đựng những tính năng tốt của Java 2 Platform như “viết một lần, chạy ở mọi nơi”, JDBC (Java Database Connectivity) truy cập cơ sở dữ liệu, CORBA kết hợp với những tài nguyên đã có… Được xây dựng trên nền tảng là Java 2 Platform, J2EE mở rộng hổ trợ cho nhiều công nghệ mới như Enterprise JavaBeans, Java Servlets, JavaServer Pages và XML.

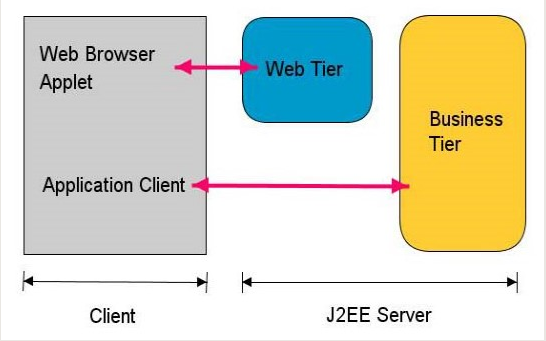
1.2 Kiến trúc của J2EE

J2EE nền tảng sử dụng một mô hình ứng dụng phân tán đa tầng.

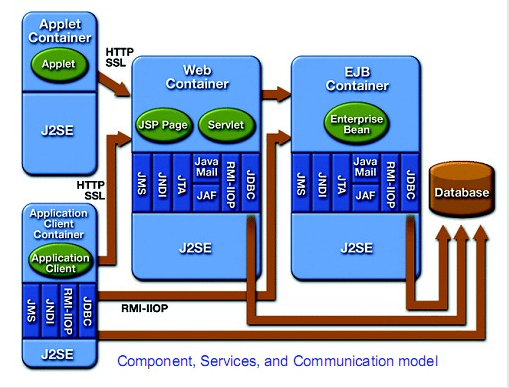


Trong mô hình ứng dụng J2EE có nhiều tầng:

Tầng khách hàng (client tier), tầng web (web tier),Tầng thương mại(business tier) và tầng hệ thống thông tin thương mại (enterprise information system tier). Tầng thương mại và tầng web nằm trên một máy chủ ứng dụng gọi là máy chủ ứng dụng (application server) hay máy chủ J2EE (J2EE server). Máy chủ J2EE cung cấp những dịch vụ cần thiết cho những thành phần (component) của tầng thương mại và tầng web.



Có hai loại khách hàng (client) là: Những khách hàng ứng dụng (application clients) và những khách hàng web (web client). Khách hàng web truy cập tới những thành phần trong tâng web là các lớp Java Servlet hay Java ServerPages. Những khách hàng ứng dụng là những ứng dụng độc lập, chúng truy cập trực tiếp tới các thành phần trong tầng thương mại.



**J2EE có các loại container như :**

Web Container: Quản lý sự thực thi của các thành phần web, và container của chúng chạy trong máy chủ J2EE (J2EE server).

EJB Container : Quản lý sự thực thi của các thành phần EJB, các thành phần EJB và container của chúng chạy trong J2EE server.

Application client Container : Quản lý sự thực thi các thành phần của ứng dụng client.

Applet Container : Quản lý sự thực thi của applets(ứng dụng ký sinh).

**Các công nghệ trong J2EE :**

Java Database Connectivity (JDBC) API: JDBC là một tập hợp các giao diện cho phép các ứng dụng Java truy cập vào cơ sở dữ liệu bất kỳ. API này có cùng một mục đích như ODBC (Open Database Connectivity) Microsoft.

Remote Method Invocation (RMI): RMI là một API cho phép các đối tượng Java để giao tiếp từ xa với các đối tượng khác.

Java IDL: IDL (Interface Definition Language) là một ngôn ngữ nền tảng tiêu chuẩn khai báo độc lập được sử dụng để xác định các giao diện đối tượng triển khai cung cấp và đối tượng khách hàng gọi. Java IDL cho phép bất kỳ đối tượng Java để giao tiếp với các đối tượng khác trong ngôn ngữ bất kỳ bằng phương tiện của IDL.

Enterprise Java Beans: Đóng gói các logic thương mại (business logic) và hổ trợ thực hiện giao tác và bảo mật.

Servlets và Java Server Pages (JSP): Công nghệ dùng trong xây dựng những ứng dụng web.

Java Message Service (JMS): Tập APIs cho phép ứng dụng tạo, gởi, nhận, và đọc thông điệp bất đồng bộ.

Java Transaction API (JTA): JTA cho phép những ứng dụng J2EE thực hiện những giao tác phân tán (distibuted transaction).

JavaMail: Cho phép thành phần java gởi và nhận mail.

Java API cho xử lý XML (Java API for XML Processing -JAXP): XML là một định dạng dữ liệu cho các tài liệu thay thế cấu trúc trên Web. JAXP cho phép các ứng dụng Java phân tích và chuyển đổi tài liệu XML. XML được sử dụng nhiều trong J2EE như là một định dạng dữ liệu.

Java Naming và Directory Interface (JNDI): JNDI là một giao thức cung cấp một tập API chuẩn cho phép truy cập tới những dịch vụ naming và directory.

Java Persistence API (JPA) : Cho phép ánh xạ giữa một lớp java với các cột trong một bảng.

# 2. Giới thiệu về project sẽ phát triển

2.1 Ý tưởng

Trong bối cảnh xã hội phát triển, con người dần chú ý hơn đến việc rèn luyện sức khỏe thông qua các hoạt động thể dục thể thao. Điển hình là từ năm 2013 đến nay, số lượng thanh thiếu niên cũng như người lớn tuổi đi đến phòng gym tăng mạnh. Trước nhu cầu đó, nhiều người đã mở phòng gym để kinh doanh. Chắc chắn họ sẽ cần một số lượng lớn các dụng cụ tập luyện thê dục thể thao. Hiểu rõ được điều, nên nhóm quyết định lập ra hệ thống bán các thiết bị luyện tập thể dục thể thao phục vụ cho phòng gym.

2.2 Các tính năng có trong project

Trang người dùng:

- Đăng ký: cho phép người dùng đăng ký tài khoản vào hệ thống.

- Đăng nhập: cho phép người dùng đăng nhập vào hệ thống.

- Hiển thị tất cả sản phẩm cho người dùng chọn lựa.

- Cho phép người dùng lựa chọn sản phẩm theo danh mục.

- Giỏ hàng: cho phép người dùng thêm các sản phẩm muốn mua vào giỏ hàng, có thể xóa chúng đi nếu không muốn mua nữa.

- Trang hóa đơn: cho phép người dùng nhìn lại những thông tin, giá tiền của các sản phẩm trước khi họ gởi đơn hàng đến hệ thống.

Trang Admin:

* Quản lý sản phẩm: cho phép Admin quản lý thêm, xóa, sửa các sản phẩm có trong kho.
* Quản lý loại sản phẩm: cho phép Admin quản lý thêm, xóa, sửa các loại sản phẩm có trong kho.

# Phần 2: Công nghệ

JSP, Servlet thuần túy và không sử dụng bất kỳ framework nào khác.

# 1. JSP là gì?

JSP là công nghệ được viết bằng ngôn ngữ lập trình Java và được dùng trong việc phát triển các ứng dụng web. Chương trình viết bằng JSP được chạy trên server và hỗ trợ tạo ra nội dung động trước khi gửi nội dung này về hiển thị ở phía client. JSP là viết tắt của cụm từ JavaSever Pages.

# 2. Servlet là gì?

Một Servlet là 1 Class của Java được chạy trên Server chuyên dụng, có nhiệm vụ xử lí các nghiệp vụ của Web Application. Một Servlet được cung cấp rất nhiều các Object và Class khác để làm việc, cho nên người ta gọi nó là Servlet Application Programing Interface (API). Do nó cũng là 1 Class Java cho nên có phần mở rộng là .java. Nhưng sau khi được biên dịch thành byte code có phần mở rộng .class thì nó mới chạy được ở trên Web Server.

# 3. JSP Hoạt Động Như Thế Nào

Một trang JSP bao gồm hai phần:

+ Các thẻ markup HTML và

+ Các thẻ JSP.

Khi chạy ứng dụng web, chính các thẻ JSP này sẽ tạo ra các trang HTML động được tùy biến cho từng request trước khi gửi về cho client.

# 4. Tính Năng Của JSP

Tương tự như các công nghệ dùng trên server như ASP.NET hay PHP, JSP có thể được dùng để:

+ Xử lý dữ liệu form được gửi lên server từ client.

+ Tạo các trang web với dữ liệu động thông qua việc kết nối tới cơ sở dữ liệu, lưu trữ và cập nhật dữ liệu trên database.

+ Xử lý việc đăng nhập, đăng xuất người dùng thông qua session.

+ Và rất nhiều tính năng khác.

# 5. Tại Sao Sử Dụng JSP

Trong giai đoạn đầu tiên của ngành lập trình web, Common Gateway Interface (hay CGI) là công nghệ được sử dụng rất phổ biến để tạo các trang web động. Mặc dù được sử dụng với mục đích giống với CGI, tuy nhiên JSP có những ưu điểm hơn CGI như:

+ Performance tốt hơn CGI. JSP cho phép chúng ta có thể nhúng các thẻ giúp tạo nội dung động vào trang HTML thay vì làm ngược lại như CGI là chèn mã HTML vào trong mã lệnh chương trình. Bạn sẽ tìm hiểu cách nhúng mã code chương trình trong trang HTML ở các bài học sau.

+ Các trang viết bằng JSP sẽ được compile (biên soạn) trước khi xử lý bởi web server. Điều này giúp cải thiện tốc độ xử lý request gửi tới server.

+ Mặc dù Java là ngôn ngữ mặc định khi sử dụng JSP, tuy nhiên chúng ta có thể sử dụng các ngôn ngữ khác như JavaScript, Perl hay VBScript.

+ Ngoài ra, JSP được phát triển dựa trên Java Servlets API do đó sử dụng JSP chúng ta có thể tích hợp với các Enterprise Java API như JDBD database, Enterprise JavaBeans (EJB)...

+ JSP đặc biệt thích hợp để phát triển các ứng dụng doanh nghiệp enterprise yêu cầu phải xử lý số lượng lớn các logic phức tạp đồi hỏi nhiều tài nguyên server.

# Phần 3: Thiết kế – Cài đặt

# 1. Kiến trúc hệ thống

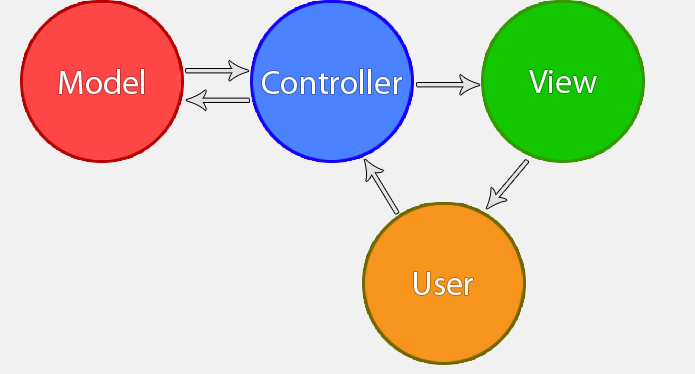
* Mô hình MVC.
  1. MVC là gì?
* MVC viết tắt của 3 từ đó là Model – View – Controller (MVC) là mẫu thiết kế nhằm mục tiêu chia tách phần Giao diện và Code để dễ quản lý, phát triển và bảo trì.
* Mỗi phần lại có có một nhiệm vụ xử lý khác nhau, đối với trong mô hình MVC trong java nói riêng và mô hình MVC nói chung thì :

**+ Model**: Tương tác và truy xuất dữ liệu đến database (cơ sở dữ liệu)

**+ View**: Giao diện mà người dùng có thể nhìn thấy, tuy nhiên thường view chỉ có một nhiệm vụ duy nhất là hiển thị dữ liệu. Trong Java web, view chính là jsp.

**+ Controller**: Nó có nhiệm vụ điều khiển tương tác giữa Model và View cũng như xử lý logic nghiệp vụ (Business). Có thể giải thích kỹ hơn nữa đối với trong Java thì controller lấy dữ liệu từ model sau đó gửi đến view. Trong [Java web](https://itphutran.com/hoc-java-web-voi-jspservlet) controller là file servlet.

Bên dưới là hình ảnh mô phỏng mô hình MVC:



Nhìn vào mô hình trên thì có thể giải thích rằng khi có một user (người dùng) truy cập vào một địa chỉ url, tất nhiên sẽ có một request đến và lúc này controller sẽ lấy yêu cầu của người dùng. Sau đó sẽ gửi yêu cầu đó đến model. Model sẽ tương tác với database nếu có (đa số sẽ truy xuất dữ liệu từ database), sau đó lấy đúng dữ liệu yêu cầu và gửi trả lại cho controller. Controller xử lý nghiệp vụ, logic business và gửi trả lại cho view. View thực hiện nhiệm vụ hiển thị dữ liệu đúng kết quả cho người dùng yêu cầu.

Tại sao cần mô hình MVC?

Ưu điểm của mô hình MVC:

+ Việc sử dụng MVC mang tính chất chuyên nghiệp trong lập trình web vì nó mang tính logic cao nhưng đơn giản, phân tách rõ ràng các chức năng cần làm của một ứng dụng, qua đó nhiều người có thể làm chung dự án, phát hiện và sửa lỗi, dễ nâng cấp tính năng đồng thời khi bảo trì cũng rất dễ dàng.

Nhược điểm của mô hình MVC:

+ MVC khi dùng trong các project nhỏ sẽ gây phức tạp vấn đề cần giải quyết

. Mặc dù, MVC tỏ ra lợi thế hơn nhiều so với cách lập trình thông thường. Nhưng MVC luôn phải nạp, load những thư viện đồ sộ để xử lý dữ liệu. Chính điều này làm cho mô hình trở nên chậm chạp hơn nhiều so với việc code tay thuần túy.

. MVC đòi hỏi người tiếp cận phải biết qua OOP, có kinh nghiệm tương đối cho việc thiết lập và xây dựng một ứng dụng hoàn chỉnh. Sẽ rất khó khăn nếu OOP của người sử dụng còn yếu.

. MVC tận dụng mảng là thành phần chính cho việc truy xuất dữ liệu. Nhất là với việc sử dụng active record để viết ứng dụng. Chúng luôn cần người viết phải nắm vứng mô hình mảng đa chiều.



# 2. Sơ đồ use-case

2.1. Sơ đồ use-case tổng quát cho cả hệ thống:

a. Sơ đồ



b. Mô tả:

- Đăng ký: Nếu khách hàng là người dùng mới, anh ta sẽ được chuyển đến trang đăng ký. Tại đây, anh ta sẽ điền các thông tin cá nhân cũng như tên tài khoản, mật khẩu anh ta muốn vào các form đã tạo sẵn.

- Đăng nhập: khách hàng có thể đăng nhập bằng tài khoản và mật khẩu anh ta đã tạo trước đây. Trang này sẽ kiểm tra thông tin đăng nhập khách hàng nhập vào có đúng hay không. Nếu không đúng, sẽ báo thông tin sai cho khách hàng biết.

- Xem sản phẩm: khách hàng có thể chọn xem sản phẩm theo danh mục. Hệ thống sẽ hiển thị thông tin theo tên danh mục mà người dùng đã chọn.

- Tìm kiếm sản phẩm: khách hàng nhập tên của sản phẩm cần tìm vào ô tìm kiếm và gởi yêu cầu. Hệ thống sẽ tìm kiếm tất cả các sản phẩm có trong kho. Nếu tìm thấy sẽ trả về cho người dùng sản phẩm cần tìm. Nếu không, sẽ có thông báo không tìm thấy.

- Xem giỏ hàng: khách hàng sẽ gởi yêu cầu để xem được giỏ hàng. Hệ thống sẽ trả về trang giỏ hàng cho khách hàng, bao gồm đơn giá của sản phẩm mà khách hàng chọn và tổng giá trị đơn hàng.

- Thêm sản phẩm vào giỏ hàng: Khi khách hàng tìm thấy sản phẩm anh ta muốn mua, anh ta sẽ thêm sản phẩm vào giỏ hàn. Hệ thống sẽ thêm sản phẩm vào giỏ hàng của anh ta và kiểm soát các sản phẩm ấy.

- Cập nhật thông tin khách hàng: khách hàng sẽ gởi yêu cầu để cập nhật các thông tin như họ tên, địa chỉ… Sau này khi mua hàng, địa chỉ nhận hàng của khách hàng cũng sẽ được đổi theo địa chỉ mà khách hàng đã cập nhật.

- Thanh toán: khách hàng sẽ gởi yêu cầu thanh toán đến hệ thống. Sau đó anh ta sẽ nhập thông tin thẻ ngân hàng của mình vào trang web của bên thanh toán trung gian. Nếu thông tin sai thì khách hàng sẽ phải nhập lại. Nếu thông tin đúng, việc thanh toán của khách hàng sẽ hoàn tất.

- Kiểm tra thông tin thẻ ngân hàng: Hệ thống kiểm tra của bên thanh toán trung gian sẽ kiểm tra xem thông tin thẻ ngân hàng mà khách hàng nhập có đúng hay không. Nếu không, khách hàng sẽ bị yêu cầu nhập lại thông tin thẻ ngân hàng.

- Cập nhật thông tin sản phẩm: Người quản lý hệ thống bán hàng sẽ thường xuyên cập nhật các thông tin của sản phẩm như giá, nhãn hàng… và hệ thống sẽ cập nhật các thông tin ấy vào cơ sở dữ liệu.

- Nhập hàng: người quản lý sẽ gởi yêu cầu nhập hàng đến hệ thống. Hệ thống sẽ nhận các thông tin do người quản lý cung cấp và lưu xuống cơ sở dữ liệu.

- Giao hàng: sau khi nhận được thông tin đơn hàng, nhân viên giao hàng sẽ giao hàng cho khách hàng trong vòng 4 – 7 ngày.

2.2 Sơ đồ use-case đăng nhập

a. Sơ đồ



b. Mô tả

- Đăng nhập: khách hàng có thể đăng nhập bằng tài khoản và mật khẩu anh ta đã tạo trước đây. Trang này sẽ kiểm tra thông tin đăng nhập khách hàng nhập vào có đúng hay không. Nếu không đúng, sẽ báo thông tin sai cho khách hàng biết.

- Đăng xuất: khách hàng đăng xuất ra khỏi hệ thống.

- Nhập mã captacha: khách hàng nhập mã captacha vào hệ thống.

- Nhập vào username: khách hàng nhập username vào hệ thống.

- Nhập vào password: khách hàng nhập password vào hệ thống.

- Trả lời câu hỏi bảo mật: trong trường hợp quên mật khẩu, khách hàng có thể dùng tính năng này để lấy lại mật khẩu của mình.

2.2 Sơ đồ use-case xem sản phẩm

a. Sơ đồ



b. Mô tả

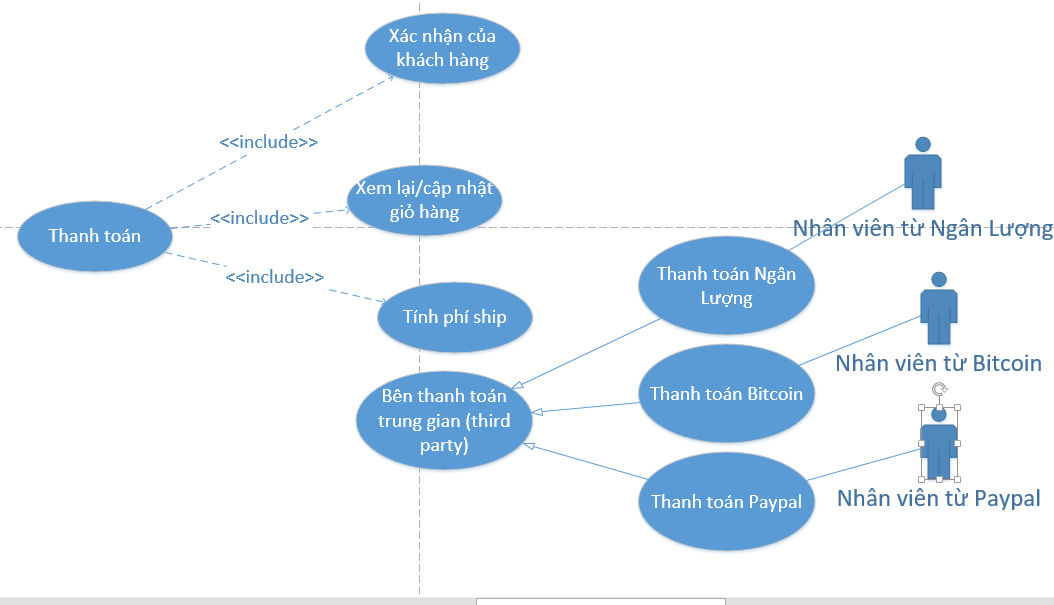
- Xem sản phẩm: cho phép khách hàng xem tất cả các sản phẩm hiện có trong hệ thống.

- Tìm kiếm sản phẩm: khách hàng nhập tên của sản phẩm cần tìm vào ô tìm kiếm và gởi yêu cầu. Hệ thống sẽ tìm kiếm tất cả các sản phẩm có trong kho. Nếu tìm thấy sẽ trả về cho người dùng sản phẩm cần tìm. Nếu không, sẽ có thông báo không tìm thấy.

- Xem sản phẩm theo danh mục: khách hàng chọn danh mục sản phẩm mình muốn xem. Sau khi hệ thống nhận được yêu cầu từ khách hàng, hệ thống sẽ hiển thị tất cả các sản phẩm thuộc danh mục sản phẩm mà khách hàng muốn tìm kiếm.

2.2 Sơ đồ use-case thanh toán

a. Sơ đồ



b. Mô tả

- Xác nhận của khách hàng: khách hàng sẽ xác nhận thông tin đơn hàng trước khi thanh toán.

- Xem lại/cập nhật giỏ hàng: khách hàng có thể quay lại giỏ hàng ban đầu để tùy chỉnh thêm, xóa, sửa sản phẩm.

- Tính phí ship: Sau khi xác định được địa chỉ của khách hàng, hệ thống sẽ xác định phí ship tùy theo nơi ở của khách hàng.

- Bên thanh toán trung gian: các bên thanh toán trung gian có thể là Ngân Lượng, Bitcoin, Paypal. Họ sẽ có hệ thống hỗ trợ việc kiểm tra thông tin, thanh toán của khách hàng.

2.2 Sơ đồ use-case quản lý sản phẩm

a. Sơ đồ



b. Mô tả

- Thêm sản phẩm: Hệ thống sẽ cho phép Administrator thêm sản phẩm vào hệ thống.

- Sửa sản phẩm: Hệ thống cho phép Administrator cập nhật các thông tin liên quan đến sản phẩm.

- Xóa sản phẩm: Hệ thống cho phép Administrator xóa sản phẩm trong hệ thống.

- Cập nhật hình ảnh: Administrator có thể cập nhật hình ảnh cho sản phẩm khi cần thiết.

# 3. Sơ đồ sequence

3.1 Sơ đồ

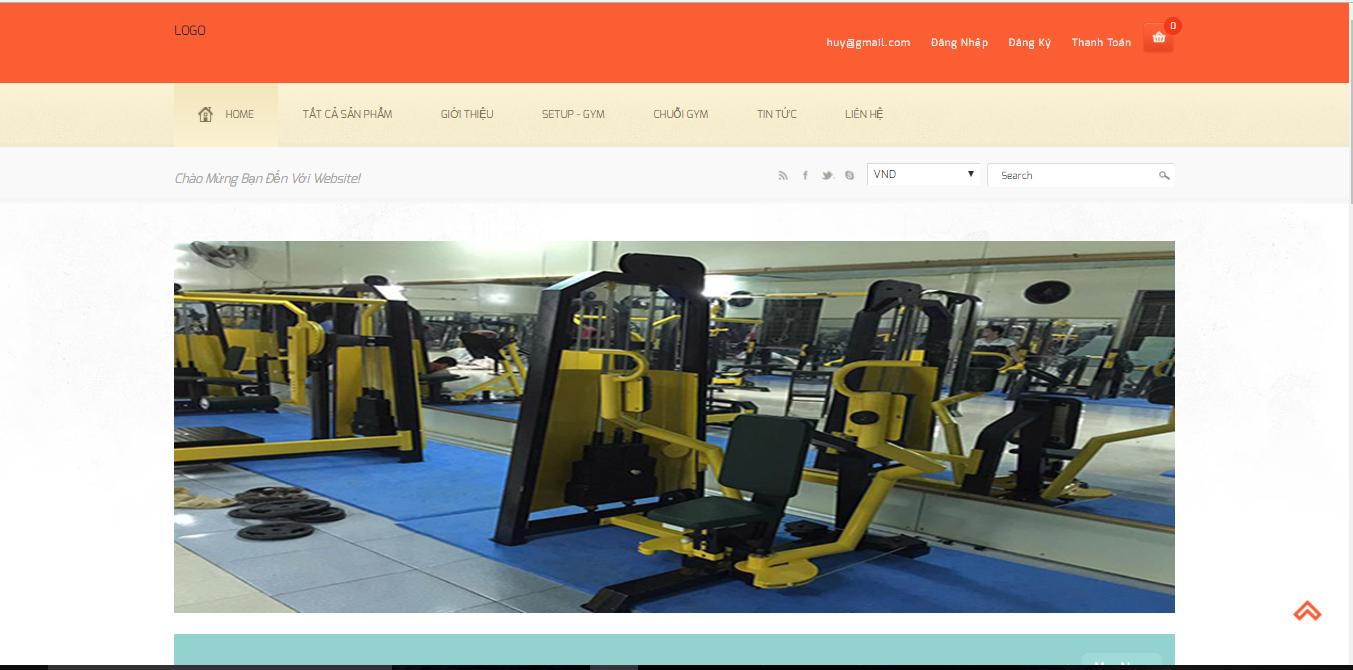


3.2 Mô tả

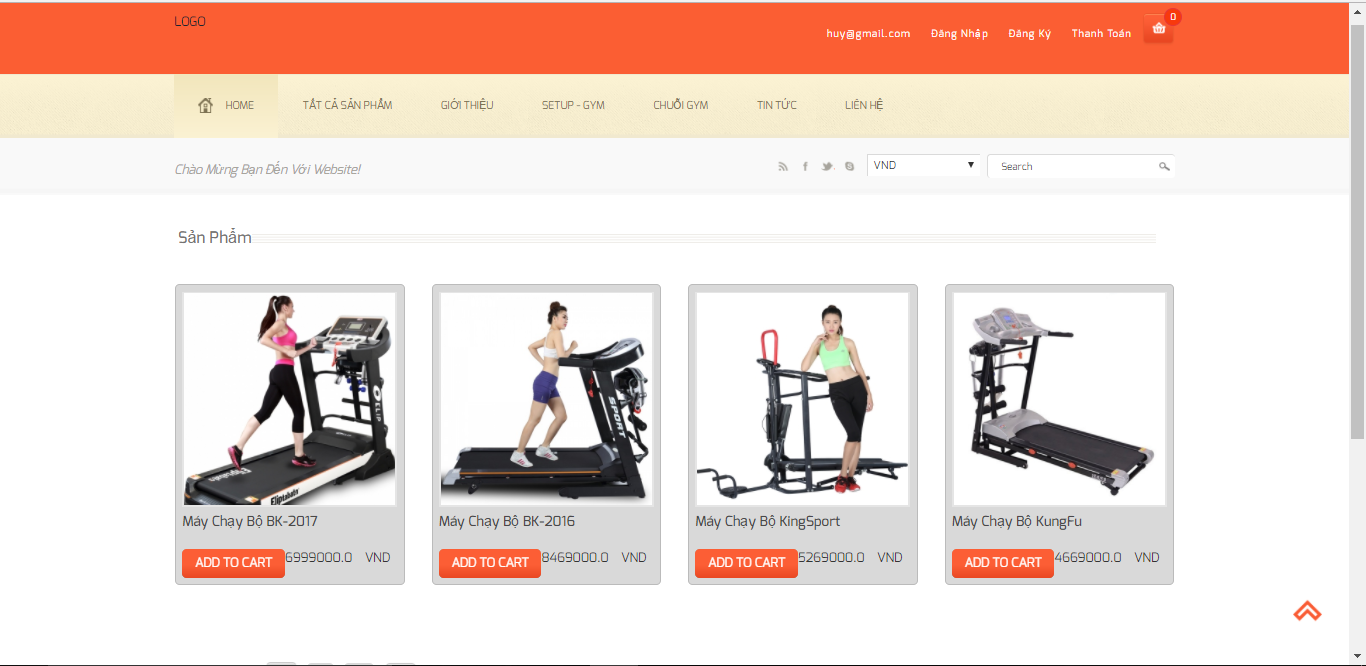
* Khách hàng đăng nhập vào hệ thống. Hệ thống kiểm tra và phản hồi với khách hàng.
* Khách hàng lựa chọn sản phẩm rồi thêm vào giỏ hàng.
* Nếu không thích sản phẩm nào, khách hàng có thể xóa khỏi giỏ hàng.
* Khách hàng có thể tùy chỉnh xem, xóa, sửa đối với giỏ hàng của mình.
* Khi khách hàng nhấn thanh toán, hệ thống sẽ nhận được đơn hàng.
* Nhân viên hoặc người quản lý hệ thống sẽ kiểm tra phương thức thanh toán cũng như thông tin đơn hàng của khách hàng.
* Bên thanh toán trung gian có trách nhiệm xác nhận thông tin thẻ credit card của khách hàng.
* Ngoài ra, người dùng cũng có thể đăng nhập thường xuyên vào hệ thống để cập nhật tình trạng đơn hàng. Quản lý hệ thống có trách nhiệm cập nhật tình trạng đơn hàng cho khách hàng.

# 4. Giao diện ứng dụng

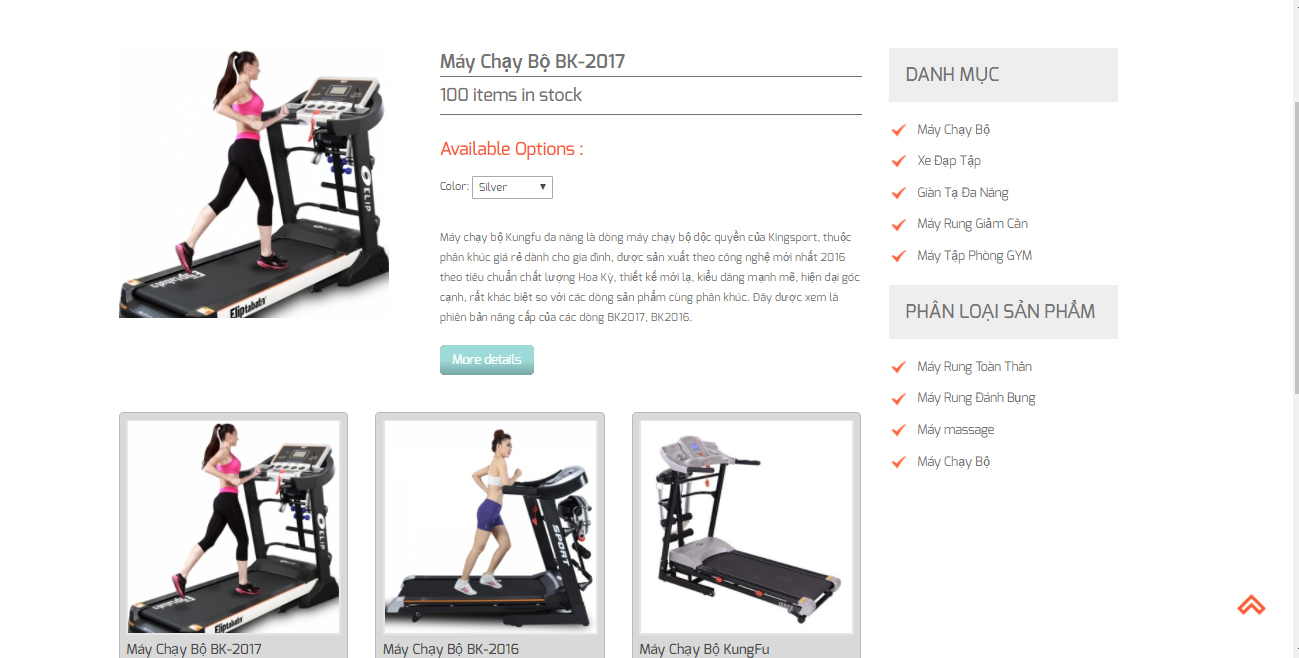
4.1. Trang Chủ.



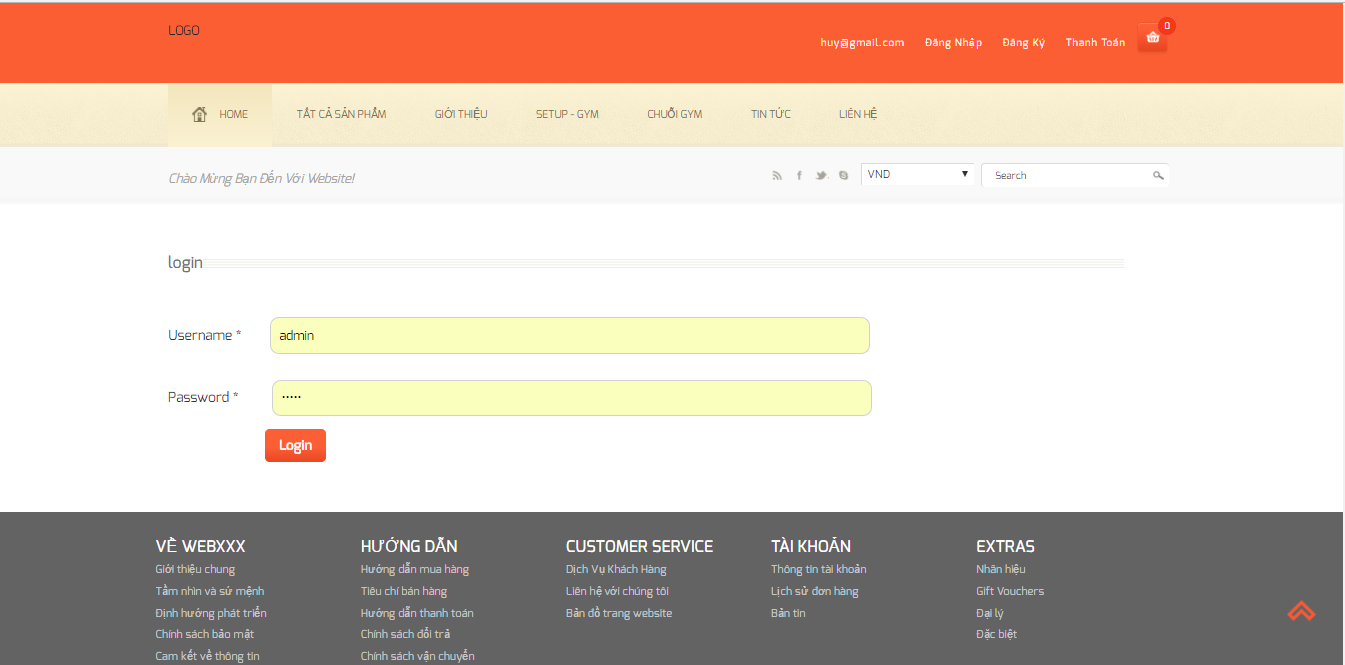
4.2. Danh mục sản phẩm.



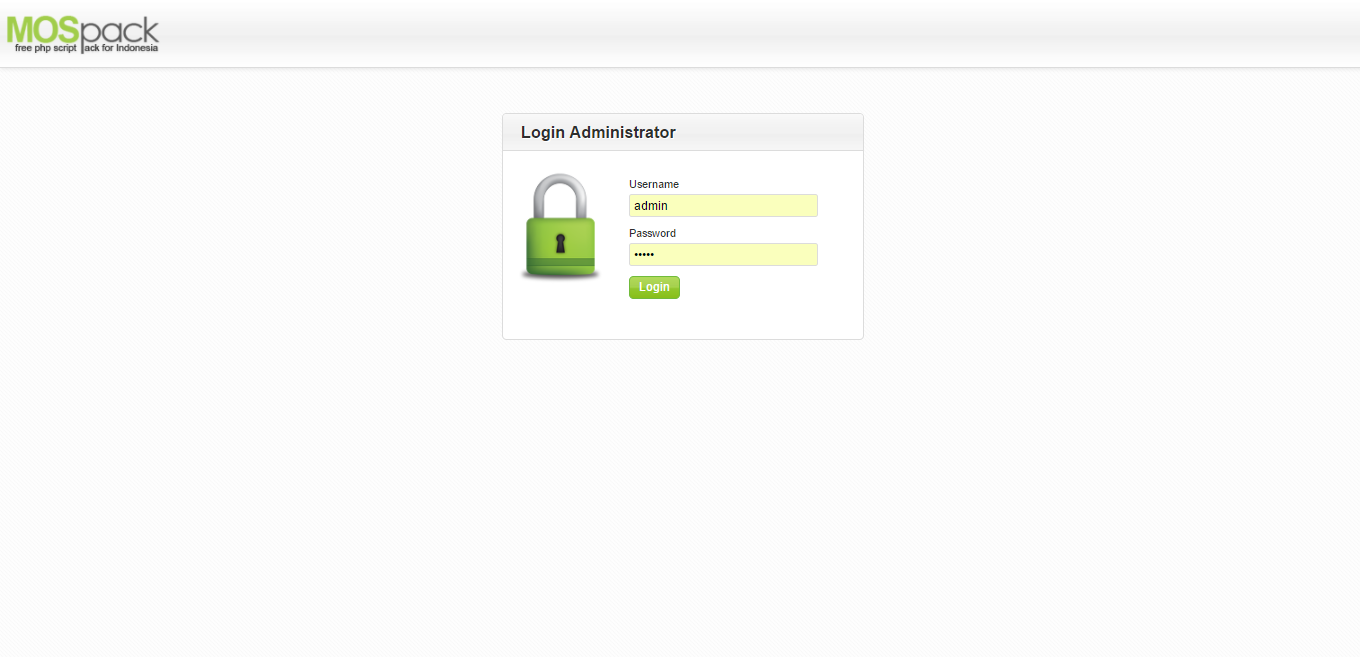
4.3. Chi tiết sản phẩm.



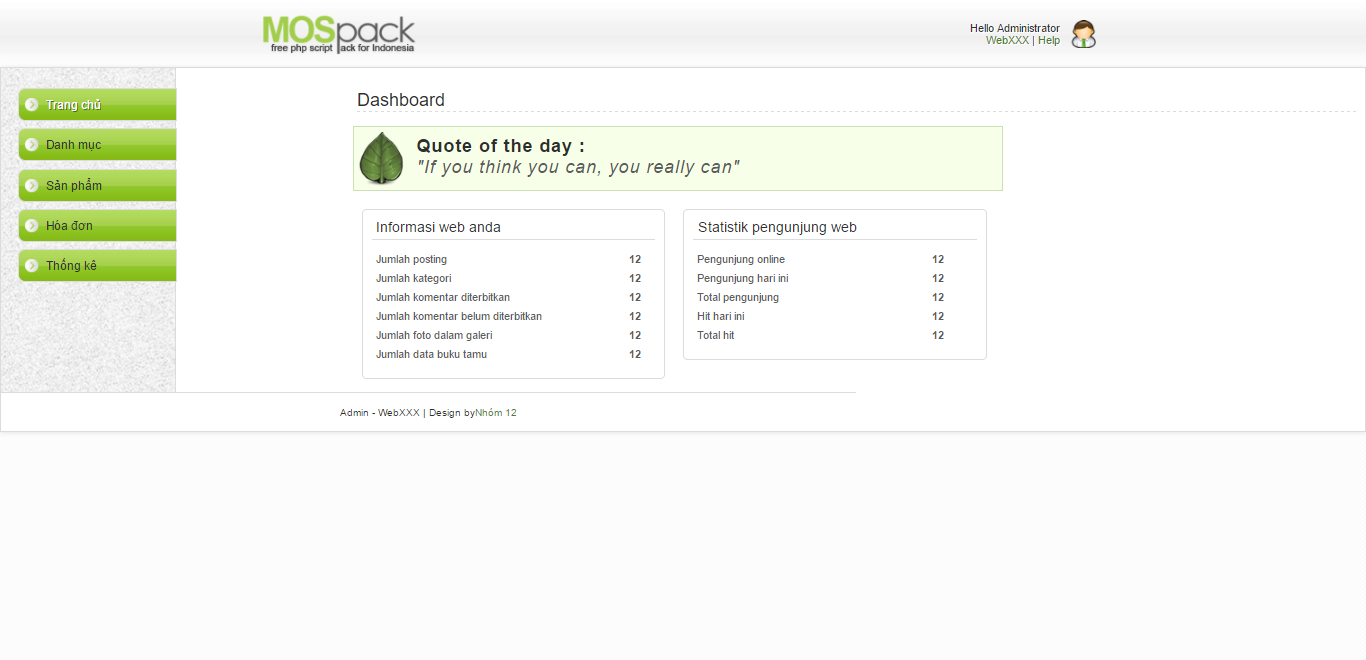
4.4. Đăng ký, đăng nhập.



4.5. Đăng nhập Admin.



4.6. Trang chủ Admin.



# Phần 4: Kết luận – Hướng phát triển

# 1. Kết luận

1.1 Môi trường phát triển

Môi trường: Website được viết với ngôn nữa Java, JSP, Servlet.

IDE: Netbean.

Hệ quản trị cơ sở dữ liệu: MySQL.

1.2 Kết luận

- Kết quả đạt được:

Hệ thống bán các thiết bị luyện tập thể dục thể thao về cơ bản đã đáp ứng đầy đủ các nhu cầu của 1 hệ thống bán hàng như: tìm kiếm sản phẩm, đặt hàng online, quản lý sản phẩm, hóa đơn, thành viên... Bên cạnh đó, hệ thống còn có giao diện thân thiện, quen thuộc với người dùng, giúp cho người dùng dễ dàng chọn lựa được các dụng cụ luyện tập thể dục thể thao ưng ý nhằm cải thiện sức khỏe của họ.

* Khuyết điểm:

+ Chương trình chưa có hệ thống thanh toán online từ ngân hàng, Bitcoin hay Ngân Lượng.

+ Một số tính năng về bảo mật hệ thống, chống hacker, DDOS cần phải được cải thiện tốt hơn.

# 2. Hướng phát triển

* Hoàn thiện phân tích thiết kế và xây dựng chương trình tốt hơn.
* Sử dụng lại hệ thống để ứng dụng vào các bài tóan bán hàng khác.
* Dự án quy mô nhỏ nên vẫn còn hạn chế trong việc phát triển hiện hành.
* Chương trình chưa được hoàn thiện một cách tối ưu nhưng đã đáp ứng được phần nào yêu cầu dự án đưa ra.

# 3. Phụ lục: Hướng dẫn cài đặt và sử dụng

# 4. Tài liệu tham khảo

<http://creately.com/blog/diagrams/use-case-diagram-tutorial/>

<http://o7planning.org/vi/10263/huong-dan-lap-trinh-java-jsp>

<http://creately.com/blog/diagrams/sequence-diagram-tutorial/>

# BẢNG PHÂN CÔNG TRONG QUÁ TRÌNH THỰC HIỆN

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Thành viên thực hiện** |
| **1** | **Phân tích bài toán** | **Vũ Đức Huy**  **Phạm Văn Huy**  **Lê Nguyễn Quý** |
| **2** | **Phân tích thiết kế cơ sở dữ liệu** | **Vũ Đức Huy**  **Phạm Văn Huy**  **Lê Nguyễn Quý** |
| **4** | **Chức năng đăng ký, đăng nhập** | **Vũ Đức Huy** |
| **5** | **Hiển thị sản phẩm theo danh mục** | **Lê Nguyễn Quý** |
| **5** | **Hiển thị danh mục sản phẩm** | **Lê Nguyễn Quý** |
| **6** | **Hiển thị chi tiết sản phẩm** | **Lê Nguyễn Quý** |
| **7** | **Kiểm thử chương trình** | **Vũ Đức Huy** |
| **8** | **Phân quyền người dùng** | **Vũ Đức Huy** |
| **9** | **Sơ đồ Usecase – đặc tả usecase** | **Lê Nguyễn Quý** |
| **10** | **Sequence Diagram** | **Lê Nguyễn Quý** |
| **11** | **Phân trang** | **Phạm Văn Huy** |
| **12** | **Trang quản trị Admin** | **Phạm Văn Huy** |
| **13** | **Chức năng Cart** | **Phạm Văn Huy** |
| **14** | **Chức năng bình luận facebook** | **Phạm Văn Huy** |
| **15** | **Thêm, xóa, sửa sản phẩm** | **Phạm Văn Huy** |
| **16** | **Thêm, xóa, sửa danh mục** | **Phạm Văn Huy** |
| **17** | **Thống kê số lượng sản phẩm** | **Phạm Văn Huy** |
| **18** | **Làm báo cáo** | **Lê Nguyễn Quý, Phạm Văn Huy, Vũ Đức Huy** |