

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Khoa Công Nghệ Phần Mềm



---

*ĐỒ ÁN MÔN*

*CHUYÊN ĐỀ J2EE*

---

**Lớp: SE325.H22**

**Đề tài: Website Bán Quần Áo**

**Version 1.0**

Giảng viên hướng dẫn:

Ths. Nguyễn Trác Thức

Sinh viên thực hiện:

- |                       |            |
|-----------------------|------------|
| 1. Tô Thành Thương    | – 13520862 |
| 2. Phạm Hoàng Hải Sơn | – 13520708 |
| 3. Nguyễn Văn Thuyền  | – 13520873 |

TP Hồ Chí Minh, Ngày 03 tháng 06 năm 2017

# MỤC LỤC

THÔNG TIN CHUNG.....	1
<b>1. MÔI TRƯỜNG DỰ KIẾN PHÁT TRIỂN: .....</b>	<b>1</b>
<b>2. THÔNG TIN VỀ NHÓM.....</b>	<b>1</b>
PHẦN 1: GIỚI THIỆU .....	2
<b>1. KHẢO SÁT HIỆN TRẠNG VỀ PROJECT SẼ PHÁT TRIỂN .....</b>	<b>2</b>
<b>2. TỔNG QUAN VỀ CÔNG NGHỆ J2EE.....</b>	<b>3</b>
PHẦN 2: CÔNG NGHỆ .....	6
<b>1. HTML .....</b>	<b>6</b>
1.1 HTML là gì? .....	6
1.2 HTML được xử lý như thế nào? .....	7
1.3 Cấu trúc một đoạn HTML.....	7
1.4 Dùng chương trình gì để tạo tập tin HTML.....	7
1.5 HTML đóng vai trò gì trong website .....	7
<b>2. CSS .....</b>	<b>8</b>
2.1 CSS là gì?.....	8
2.2 Cấu trúc một đoạn CSS.....	10
2.3 Cách chèn CSS vào trang HTML: .....	10
<b>3. Javascript.....</b>	<b>12</b>
3.1 Javascript là gì?.....	12
3.2 Đặc tính của ngôn ngữ javascript .....	13
3.3 Ngôn ngữ javascript.....	13
3.4 Cách nhúng javascript vào trong tập tin HTML .....	13
<b>4. JQuery.....</b>	<b>14</b>
4.1 JQuery là gì? .....	14
4.2 Đặc tính của jquery.....	14
4.3 Dùng JQuery để làm gì? .....	15
4.3 Sử dụng JQuery như thế nào?.....	15
<b>5. Bootstrap.....</b>	<b>16</b>
5.1 Bootstrap là gì? .....	16

5.2 Tại sao bạn nên sử dụng Bootstrap? .....	17
5.3 Lấy Bootstrap từ đâu?.....	18
<b>6. JSP.....</b>	<b>18</b>
6.1 JSP là gì?.....	18
6.3 Chu trình sống của JSP .....	20
6.4 Mối quan hệ giữa JSP và Servlet .....	22
6.5 Các tags trong JSP .....	22
<b>7. Servlet.....</b>	<b>30</b>
7.1 Servlets là gì? .....	30
7.2 Cấu trúc của Servlets .....	30
7.3 Nhiệm vụ của Servlets .....	31
7.4 Package trong Servlets .....	31
7.5 Vòng đời của Servlet.....	32
<b>8. MVC model 2.....</b>	<b>34</b>
8.1. JSP Model 1 là gì? .....	34
8.2. JSP Model 2 .....	35
PHẦN 3: THIẾT KẾ - CÀI ĐẶT .....	36
<b>1. KIẾN TRÚC HỆ THỐNG.....</b>	<b>36</b>
<b>2. USECASE.....</b>	<b>37</b>
a. SƠ ĐỒ USE-CASE.....	37
b. DANH SÁCH CÁC ACTOR .....	37
c. DANH SÁCH CÁC USE-CASE .....	38
d. ĐẶC TẢ USECASE.....	39
<b>2. SEQUENCE .....</b>	<b>50</b>
a. Đăng nhập .....	50
b. Đăng xuất.....	51
c. Đăng ký thành viên .....	53
d. Quản lý nhập hàng.....	53
e. Quản lý sản phẩm.....	54
f. Quản lý danh mục .....	56
g. Đặt hàng.....	57
h. Duyệt đơn hàng.....	58
<b>3. GIAO DIỆN WEBSITE.....</b>	<b>59</b>
HƯỚNG DẪN CHẠY PROJECT .....	67

<b>1. Môi trường phát triển .....</b>	<b>67</b>
<b>2. Chạy project.....</b>	<b>67</b>

# **THÔNG TIN CHUNG**

## **1. MÔI TRƯỜNG DỰ KIẾN PHÁT TRIỂN:**

Chúng em sẽ phát triển một web bán quần áo dùng công nghệ JSP/SERVLET

## **2. THÔNG TIN VỀ NHÓM**

Số thứ tự: Nhóm 4

STT	MSSV	Họ và tên	Điện thoại	Email
1	13520862	Tô Thành Thương	0986335807	thuongthanhto@gmail.com
2	13520708	Phạm Hoàng Hải Sơn	0949512609	Son0949512609@gmail.com
3	13520873	Nguyễn Văn Thuyền		

# **PHẦN 1: GIỚI THIỆU**

## **1. KHẢO SÁT HIỆN TRẠNG VỀ PROJECT SẼ PHÁT TRIỂN**

Trong những năm qua đời sống của người Việt ngày càng nâng cao và cải thiện không ngừng, xã hội phát triển, nhu cầu tự khẳng định phong cách, giá trị sống ngày là điều không thể thiếu. Nhu cầu cuộc sống đang chuyển từ “ăn no mặc ấm” sang “ăn ngon mặc đẹp”, chính vì thế nhu cầu làm đẹp của mọi người trong một xã hội tri thức là điều không thể thiếu, trang phục không chỉ che chắn, bảo vệ cơ thể mà còn chứa đựng thông điệp văn hóa, xã hội, thẩm mỹ...

Cuộc sống tấp nập, bận rộn, nhiều người không đủ thời gian để đi cửa hàng, chợ, siêu thị mua áo quần để chăm lo cho vẻ ngoài của mình, trong khi đó ăn mặc đẹp là nhu cầu không thể thiếu trong cuộc sống, thời trang phục vụ cho công việc, giao dịch, làm ăn, mặc một bộ trang phục đẹp khiến mọi người tự tin hơn với mọi người xung quanh và cũng là thể hiện cá tính riêng của mình.

Trước những cơ hội phát triển của nhu cầu ăn mặc của người Việt Nam hiện nay, nhóm đã chọn đề tài kinh doanh thời trang trực tuyến. Website thời trang đáp ứng nhu cầu mua quần áo online của đa số mọi người

Project sẽ diễn tả ít nhất 3 quy trình trong việc quản lý bán hàng của shop.

### **DANH SÁCH CÁC NGHIỆP VỤ TRONG HỆ THỐNG**

- Quản lý sản phẩm
- Quản lý danh mục
- Đặt hàng
- Quản lý đơn đặt hàng
- Quản lý nhập hàng

## 2. TỔNG QUAN VỀ CÔNG NGHỆ J2EE

### Khái niệm JSP

JavaServer Pages (JSP) là một kỹ thuật xử lý các yêu cầu của client ở máy chủ (server-side), do đó chúng ta không thể thực hiện bất kỳ hành động nào ở client. JSP cho phép chúng ta tách thành phần động của trang ra khỏi thành phần tĩnh HTML.

Các thành phần script của JSP

#### 1. Các biến được định nghĩa sẵn trong JSP

Để đơn giản hóa mã trong các biểu thức hay scriptlets trong JSP, người ta cung cấp cho chúng ta 9 đối tượng đã được định nghĩa trước. Đó là: request, response, out, session, application, config, pageContext, page, exception.

- Request: Cho phép chúng ta truy cập vào các tham số của request như loại request (GET, POST, ....)
- Response: Có phạm vi toàn trang (page)
- Session: Có phạm vi trong một phiên truyền (Session). Gọi về các phiên truyền được tự động

#### 2. Biểu thức trong JSP

Biểu thức trong JSP được dùng để chèn các giá trị trực tiếp vào luồng xuất. Nó có dạng sau:

`<%=Java expressions%>`

#### 3. Xử lý nhúng và chuyển hướng giữa các trang

Trong quá trình xử lý và chuyển dữ liệu cho form chúng ta có thể nhúng và triệu gọi những trang JSP khác với trang hiện hành

- a. Nhúng file vào trang với chỉ thị include

Chỉ thị `<%include%>` thường dùng để đem một nội dung file .html hay jsp bên ngoài vào trang hiện hành

- b. Sử dụng thẻ `<jsp:include>`

Chỉ thị `<%include%>` chỉ dùng để nhúng các mã nguồn tĩnh. Nếu muốn nhúng kết quả kết xuất từ các trang jsp, servlet hay .html khác vào trang hiện hành chúng ta sử dụng thẻ `<jsp:include>` với cú pháp sau:

```
<jsp:include page="filename" flush="true"/>
```

c. Chuyển tham số bằng thẻ `<jsp:param>`

Khi nhúng trang bằng thẻ `<jsp:include>` bạn có thể chuyển tham số cho trang được nhúng để nhận được kết xuất linh động hơn bằng cách sử dụng thẻ con `<jsp:param>`

d. Chuyển sang trang mới với `sendRedirect()`

Ngoài việc chuyển tiếp chúng ta có thể sử dụng phương thức `sendRedirect()` để chuyển hướng và triệu gọi trang JSP khác xử lý. Sự khác nhau giữa chuyển tiếp (forward) và chuyển hướng (redirect) đó là lệnh `sendRedirect()` chỉ thực hiện triệu gọi trang mà không chuyển tham số của trang hiện hành cho trang chuyển hướng

## **SERVLET**

### **1. Khái niệm**

Servlet là các thành phần đối tượng nhúng trên trình chủ Web server thực hiện xử lý yêu cầu và sinh ra các trang Web động trả về máy khách. Để sử dụng được Servlet chúng ta cần có các trình chủ hiểu Java và hỗ trợ triệu gọi Servlet như Apache, Jrun, Web Logic ...

### **2. Các phương thức xử lý cơ bản của Servlet**

Lớp `GenericServlet` là phần cài đặt tổng quát cho đặc tả giao tiếp (interface) mang tên Servlet. Theo đặc tả của giao tiếp Servlet do Sun đưa ra một servlet cơ bản cần có những phương thức phục vụ cho các nhu cầu khởi tạo (init), hoạt động và phục vụ (service), hủy (destroy), trả về thông tin cấu hình (`getServletConfig`), trả về thông tin của Servlet (`getServletInfo`).



## **Tag Libraries**

Là các action do người lập trình tạo ra (custom action), một custom action được triệu gọi bằng cách dùng custom tag. Một tag library là một tập hợp các custom tag.

## **JAVABEAN**

JavaBean là một component phần mềm có thể dùng lại được, có thể được thực hiện trực quan bằng môi trường phát triển tích hợp IDE

JavaBean là một đối tượng Java bất kỳ cài đặt interface serializable. Chúng ta thường truy xuất các đối tượng Java thông qua thuộc tính của nó, với JavaBean điều này cũng hoàn toàn tương tự nhưng thường JavaBean không sử dụng trực tiếp cú pháp Java để truy xuất thuộc tính. JavaBean sử dụng các hệ tương tự như HTML để định nghĩa Bean, thiết lập hoặc lấy về các giá trị của Bean.

## **JDBC và trình điều khiển truy xuất cơ sở dữ liệu**

JDBC là các trình đạo diễn cho phép bạn truy cập vào cơ sở dữ liệu của các hệ như Access, MS SQL Server, Oracle... bằng ngôn ngữ truy vấn SQL. Theo đặc tả của Sun có 4 kiểu JDBC sau:

- Truy xuất cơ sở dữ liệu JDBC-ODBC
- JDBC kết hợp trực tiếp với các trình điều khiển cơ sở dữ liệu.
- JDBC kết nối thông qua các ứng dụng mạng trung gian
- JDBC kết nối thông qua các trình điều khiển đặc thù ở xa

## **PHẦN 2: CÔNG NGHỆ**

Các công nghệ sử dụng trong đồ án

- HTML
- CSS
- JAVASCRIPT
- JQUERY
- BOOTSTRAP
- JSP
- SERVLET
- MVC Model 2

### **1. HTML**

#### ***1.1 HTML là gì?***

HTML là chữ viết tắt của cụm từ HyperText Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (thì thoảng mình sẽ ghi là một tập tin HTML). Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví <strong> dụ </strong> và ). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ <strong>Đây là chữ in đậm</strong>). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>).

Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.

## 1.2 HTML được xử lý như thế nào?

Khi một tập tin HTML được hình thành, việc xử lý nó sẽ do trình duyệt web đảm nhận. Trình duyệt sẽ đóng vai trò đọc hiểu nội dung HTML từ các thẻ bên trong và sẽ chuyển sang dạng văn bản đã được đánh dấu để đọc, nghe hoặc hiểu (do các bot máy tính hiểu).

## 1.3 Cấu trúc một đoạn HTML

Như mình đã nói ở trên, HTML sẽ được khai báo bằng các phần tử bởi các từ khóa. Nội dung nằm bên trong cặp từ khóa sẽ là nội dung bạn cần định dạng với HTML. Ví dụ dưới đây là một đoạn HTML khai báo một đoạn văn bản.

**<p>Đây là một đoạn văn bản trong HTML.</p>**

Ngoài ra, trong các thẻ còn có các thuộc tính, thuộc tính sẽ đặt bên trong thẻ mở đầu, mỗi thuộc tính sẽ có giá trị được đặt trong dấu ngoặc kép và cách nhau bởi dấu bằng (=) với tên thuộc tính. Ví dụ dưới đây là một thẻ có sử dụng thuộc tính

**<form action="https://thachpham.com"> </form>**

Một thẻ có thể sử dụng nhiều thuộc tính chứ không phải chỉ một thuộc tính.

## 1.4 Dùng chương trình gì để tạo tập tin HTML

HTML là một tập tin siêu văn bản nên bạn có thể dùng các chương trình soạn thảo văn bản không có chức năng định dạng văn bản để tạo ra một tập tin HTML. Trong Windows, bạn có thể dùng Notepad để tạo ra một tập tin HTML, còn trên Mac thì có thể dùng TextEdit và Vim trên các hệ điều hành Linux khác. Miễn là sau đó bạn phải lưu tập tin thành đuôi .html và sử dụng trình duyệt website để đọc nó.

HTML có thể được soạn thảo bởi bất kỳ một trình soạn thảo văn bản đơn giản.

Tuy nhiên Notepad lại quá đơn giản để sử dụng cho mục đích soạn thảo, nên từ các bài sau chúng ta sẽ dùng một chương trình khác tương tự để soạn thảo HTML. Và mình cũng không bao giờ khuyến khích bạn sửa một tập tin HTML bất kỳ bằng chương trình Notepad để tránh gặp các lỗi hiển thị ký tự tiếng Việt.

## 1.5 HTML đóng vai trò gì trong website

HTML là một ngôn ngữ đánh dấu siêu văn bản nên nó sẽ có vai trò xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình

ảnh,

video,

nhạc.

Điều đó không có nghĩa là chỉ sử dụng HTML để tạo ra một website mà HTML chỉ đóng một vai trò hình thành trên website. Ví dụ một website như mangthuvien.com sẽ được hình thành bởi:

- **HTML** – Xây dựng cấu trúc và định dạng các siêu văn bản.
- **CSS** – Định dạng các siêu văn bản dạng thô tạo ra từ HTML thành một bố cục website, có màu sắc, ảnh nền,....
- **Javascript** – Tạo ra các sự kiện tương tác với hành vi của người dùng (ví dụ nhấp vào ảnh trên nó sẽ có hiệu ứng phóng to).
- **Java** – Ngôn ngữ lập trình để xử lý và trao đổi dữ liệu giữa máy chủ đến trình duyệt (ví dụ như các bài viết sẽ được lưu trong máy chủ).
- **SQLServer** – Hệ quản trị cơ sở dữ liệu truy vấn có cấu trúc (SQL – ví dụ như các bài viết sẽ được lưu lại với dạng dữ liệu SQL).

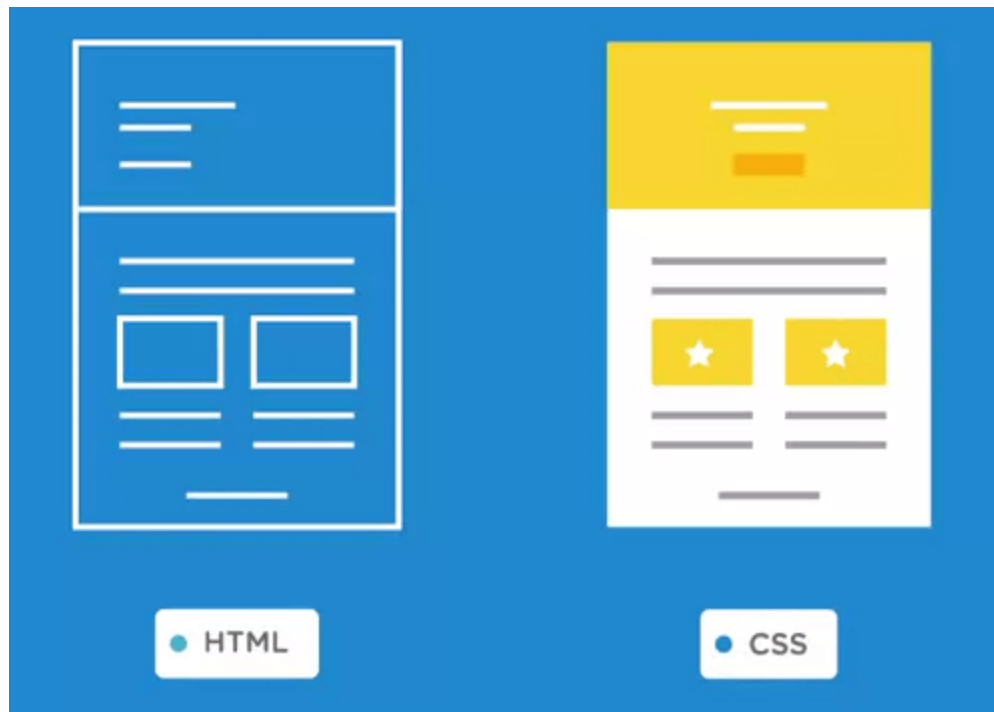
Nhưng ở đây, tạm thời chỉ cần quan tâm đến HTML mà thôi. Dễ hiểu hơn, bạn hãy nghĩ rằng nếu website là một cơ thể hoàn chỉnh thì HTML chính là bộ xương của cơ thể đó, nó như là một cái khung sườn vậy.

Như vậy, dù website thuộc thể loại nào, giao tiếp với ngôn ngữ lập trình nào để xử lý dữ liệu thì vẫn phải cần HTML để hiển thị nội dung ra cho người truy cập xem.

## 2. CSS

### 2.1 CSS là gì?

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,...thì CSS sẽ giúp chúng ta có thể thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,...rất nhiều.



CSS có vai trò trang trí thêm cho văn bản được viết bằng HTML.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

## 2.2 Cấu trúc một đoạn CSS



Một đoạn CSS bao gồm 4 phần như thế này:

```
vùng chọn {  
    thuộc tính : giá trị;  
    thuộc tính: giá trị;  
    .....  
}
```

Nghĩa là nó sẽ được khai báo bằng vùng chọn, sau đó các thuộc tính và giá trị sẽ nằm bên trong cặp dấu ngoặc nhọn { }. Mỗi thuộc tính sẽ luôn có một giá trị riêng, giá trị có thể là dạng số, hoặc các tên giá trị trong danh sách có sẵn của CSS. Phần giá trị và thuộc tính phải được cách nhau bằng dấu hai chấm, và mỗi một dòng khai báo thuộc tính sẽ luôn có dấu chấm phẩy ở cuối. Một vùng chọn có thể sử dụng không giới hạn thuộc tính.

## 2.3 Cách chèn CSS vào trang HTML:

Chúng ta có 3 cách để đưa nội dung của CSS vào trang web như sau:

**Cách 1:** Chèn nội dung CSS vào trong cặp thẻ <style> </style> trong phần <head></head> của trang web.

**Ví dụ:**

```
<html>
```

```

<head>
  <title>Chèn CSS cách 1</title>
  <! Chèn CSS vào giữa cặp thẻ sau: >
  <style type="text/css">
    //Nội dung CSS đặt bên trong đây
  </style>
</head>
<body>
  //Nội dung
</body>
</html>

```

**Cách 2:** Chèn trực tiếp vào bên trong thẻ HTML.

Chèn trực tiếp CSS vào bên trong các thẻ mở của HTML. Nếu có nhiều thuộc tính cần quy định, ta ngăn cách giữa chúng bằng khoảng trắng.

**Ví dụ:**

```

<html>
  <head>
    <title>Chèn CSS cách 2</title>
  </head>
  <body>
    <p style="Nội dung CSS ở đây kết thúc bằng dấu chấm phẩy ";" và cách nhau bằng khoảng trắng">
  </body>
</html>

```

**Cách 3:** Liên kết với một file \*.css bên ngoài.

Đây là cách được dùng nhiều nhất trong sử dụng CSS vì tính tiện dụng và linh hoạt của nó. Giả sử, bạn có một website gồm hàng trăm trang, khi bạn muốn thay đổi một

chút cách trình bày của website, thay vì gạch cạch đi gõ từng trang một, bạn chỉ cần sửa file CSS bên ngoài một chút, hàng trăm trang web của bạn đều được thay đổi.

Cú pháp: Thực hiện liên kết bằng thẻ <link> ( Không có tag đóng ), theo cú pháp sau:

```
<link rel="stylesheet" href="Đường dẫn đến file .css" type="text/css" />
```

Tag này được đặt trong cặp tag <head></head> của trang web, và không cần đặt trong tag <style></style>

Nếu bạn đặt file \*.css cùng thư mục với trang web, thì trong thuộc tính href bạn chỉ cần viết: href="ten\_file\_css.css".

Nếu file \*.css không cùng thư mục với trang web: Bạn cần chỉ trong đường dẫn tuyệt đối tới file \*.css đó trong thuộc tính href.

### **Ví dụ:**

```
<html>
  <head>
    <title>Chèn CSS cách 3</title>
    <link rel="stylesheet" href="css_name.css" type="text/css" />
  </head>
  <body>
    //Nội dung
  </body>
</html>
```

## **3. Javascript**

### **3.1 Javascript là gì?**

**Javascript** là một ngôn ngữ thông dịch, chương trình nguồn của nó được nhúng hoặc tích hợp vào tập tin HTML. Khi trang web được tải trong trình duyệt hỗ trợ javascript, Trình duyệt sẽ thông dịch và thực hiện các lệnh Javascript.

**JavaScript** là một trong những ngôn ngữ lập trình phổ biến nhất thế giới, nó là ngôn ngữ cho HTML, web, server, PC, laptop, tablet, smart phone...

- JavaScript là một ngôn ngữ kịch bản



- Ngôn ngữ kịch bản là một ngôn ngữ lập trình nhỏ.
- JavaScript là mã lập trình có thể được chèn vào các trang HTML.
- JavaScript được chèn vào các trang web có thể được chạy bởi tất cả các trình duyệt web hiện đại.
- JavaScript là một ngôn ngữ lập trình dễ học.

### **3.2 Đặc tính của ngôn ngữ javascript**

Javascript là một ngôn ngữ có đặc tính:

- Đơn giản.
- Động (Dynamic).
- Hướng đối tượng (Object Oriented).

### **3.3 Ngôn ngữ javascript**

Một trong những đặc tính quan trọng của ngôn ngữ javascript là khả năng tạo và sử dụng các đối tượng (Object). Các Object này cho phép người lập trình sử dụng để phát triển ứng dụng.

Trong javascript, các Object được nhìn theo 2 khía cạnh:

- Các Object đã tồn tại.
- Các Object do người lập trình xây dựng.

### **3.4 Cách nhúng javascript vào trong tập tin HTML**

Có 3 cách để nhúng Javascript vào trong tập tin HTML:

#### **Cách 1: Viết chương trình Javascript trực tiếp trong file HTML**

```
<script type="text/javascript">
```

```
//Các lệnh Javascript
```

```
</script>
```

#### **Cách 2: Sử dụng tập tin javascript bên ngoài :**

Cũng giống như CSS ngoại tuyến, chúng ta cũng có thể nhúng Javascript vào tập tin HTML bằng cách liên kết đến một tập tin bên ngoài, đây cũng là phương thức được sử dụng nhiều nhất.

Với phương pháp này, các lệnh Javascript sẽ được viết trong một file riêng biệt có phần mở rộng là **.js** (Ví dụ ta có tập tin my.js)

Để nhúng tập tin **demo.js** vào tập tin HTML ta sử dụng đoạn mã sau:

```
<script type="javascript" src="demo.js" type="text/javascript"></script>
```

### **Cách 3: Inline - viết trực tiếp trong thẻ HTML**

Inline nghĩa là bạn sẽ viết những đoạn mã Javascript trực tiếp trong thẻ HTML luôn. Như ví dụ dưới đây là mình đang viết dạng inline vì đoạn mã `alert(1)` được đặt trong sự kiện onclick của thẻ button

```
<input type="button" onclick="alert(1)" value="Click Me"/>
```

## **4. JQuery**

### **4.1 JQuery là gì?**

jQuery là một thư viện được xây dựng từ Javascript nhằm giúp lập trình viên xây dựng những chức năng có sử dụng Javascript trở nên dễ dàng hơn. jQuery được tích hợp nhiều module khác nhau từ module hiệu ứng cho đến module truy vấn selector. jQuery được sử dụng đến 99% trên tổng số website trên thế giới (trừ những website sử dụng JS Framework).

Các module chính của jQuery bao gồm:

- **Ajax** - xử lý Ajax
- **Attributes** - Xử lý các thuộc tính của đối tượng HTML
- **Effect** - xử lý hiệu ứng
- **Event** - xử lý sự kiện
- **Form** - xử lý sự kiện liên quan tới form
- **DOM** - xử lý Data Object Model
- **Selector** - xử lý luồng lách giữa các đối tượng HTML

### **4.2 Đặc tính của jquery**

Các đặc trưng cơ bản của jquery là:

- Thao tác HTML/DOM
- Thao tác CSS
- Phương thức sự kiện HTML
- Hiệu ứng và chuyển động
- Ajax

### 4.3 Dùng JQuery để làm gì?

- JQuery có thể cải thiện hiệu suất của ứng dụng
- JQuery tương thích hầu hết các trình duyệt
- JQuery giúp thực hiện UI mà không phải viết nhiều dòng code
- Dùng jquery bạn có thể tạo ra các hiệu ứng đẹp mắt cho website.
- Load trang nhanh hơn và không cần download jquery.
- Tiết kiệm băng thông vì không phải load jquery từ server.
- Có tính mềm dẻo hơn vì load file jquery tại các vị trí địa lý khác nhau gần nhất với người dùng thay vì load chỉ từ server của bạn.

### 4.3 Sử dụng JQuery như thế nào?

Có 2 cách thêm thư viện jquery:

- **Sử dụng local:** Vào trang chủ [jquery.com](http://jquery.com) để tải phiên bản jquery mới nhất. Sau đó bỏ file jquery bạn vừa tải xong vào thư mục web của bạn. Bây giờ bạn có thể thêm thư viện jquery vào file HTML như sau:

```
<html>
```

```
<head>
```

```
<title>The jQuery Example</title>
```

```
  <script type = "text/javascript" src = "jquery/jquery.min.js"></script>
```

```
  <script type = "text/javascript">
```

```
    $(document).ready(function(){
```

```
      document.write("Hello, World!");
```

```
    });
```

```
</script>
</head>
<body>
<h1>Hello</h1>
</body>
</html>
```

- **Sử dụng Google CDN:** Bạn thêm đoạn sau vào file HTML để thêm thư viện jquery vào

```
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript" src =
"http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
<script type = "text/javascript">
$(document).ready(function(){
    document.write("Hello, World!");
});
</script>
</head>
<body>
<h1>Hello</h1>
</body>
</html>
```

## 5. Bootstrap

### 5.1 Bootstrap là gì?

Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn.

Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế responsive của bạn dễ dàng hơn và nhanh chóng hơn.

## 5.2 Tại sao bạn nên sử dụng Bootstrap?

Những điểm thuận lợi khi bạn sử dụng bootstrap:

- **Rất dễ để sử dụng:** Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cần có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.
- **Tính năng Responsive:** Bootstrap's xây dựng sẵn responsive css trên các thiết bị phones, tablets, và desktops
- **Mobile:** Trong Bootstrap 3 mobile-first styles là một phần của core framework
- **Tương thích với trình duyệt:** Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera) nhưng lưu ý vì em IE vẫn rất hăm với IE phiên bản cũ vì thế việc IE9 hay IE8 đổ xuống không support là chuyện bình thường. Các bạn có dùng nên lưu ý điểm này theo như mình dùng làm dự án thì nó support tuyệt vời trên IE10 đổ lên nhé, thấp nhất có lẽ là IE9 thôi còn IE8 thì không nên xài nhé.

Được viết bởi những người có óc thẩm mỹ và tài năng trên khắp thế giới. Sự tương thích của trình duyệt với thiết bị đã được kiểm tra nhiều lần nên có thể tin tưởng kết quả mình làm ra và nhiều khi không cần kiểm tra lại. Vì vậy, giúp cho dự án của bạn tiết kiệm được thời gian và tiền bạc.

Chỉ cần biết sơ qua HTML, CSS, Javascript, JQuery là bạn có thể sử dụng Bootstrap để tạo nên một trang web sang trọng và đầy đủ. Nhưng lại không cần code quá nhiều CSS.

Với giao diện mặc định là màu xám bạc sang trọng, hỗ trợ các component thông dụng mà các website hiện nay cần có. Vì nó là opensource nên bạn có thể vào mã nguồn của nó để thay đổi theo ý thích của bản thân.

Do có sử dụng Grid System nên Bootstrap mặc định hỗ trợ Responsive. Bootstrap được viết theo xu hướng Mobile First tức là ưu tiên giao diện trên Mobile trước. Nên việc sử dụng Bootstrap cho website của bạn sẽ phù hợp với tất cả kích thước màn hình. Nhờ đó mà chúng ta không cần xây dựng thêm một trang web riêng biệt cho mobile.

Đội ngũ phát triển Bootstrap đã bổ sung thêm tính năng Customizer. Giúp cho designer có thể lựa chọn những thuộc tính, component phù hợp với project của họ. Chức năng này giúp ta không cần phải tải toàn bộ mã nguồn về máy.

### 5.3 Lấy Bootstrap từ đâu?

Có 2 cách để sử dụng bootstrap:

- Cách 1: bạn có thể tải bootstrap từ [getbootstrap.com](https://getbootstrap.com)
- Cách 2: bạn dẫn đường link trực tiếp từ CDN

MaxCDN:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/
/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

## 6. JSP

### 6.1 JSP là gì?

- JavaServer Pages (JSP) là một công nghệ để phát triển các Webpage mà hỗ trợ nội dung động, giúp các lập trình viên chèn java code vào trong các HTML page bằng việc sử dụng các JSP tag đặc biệt, hầu hết bắt đầu với `<%` và kết thúc với `%>`.
- Sự kết hợp các thành phần của các đối tượng HTML, XML, Servlet (mở rộng từ Servlet) và Java Code để tạo ra trang Web động

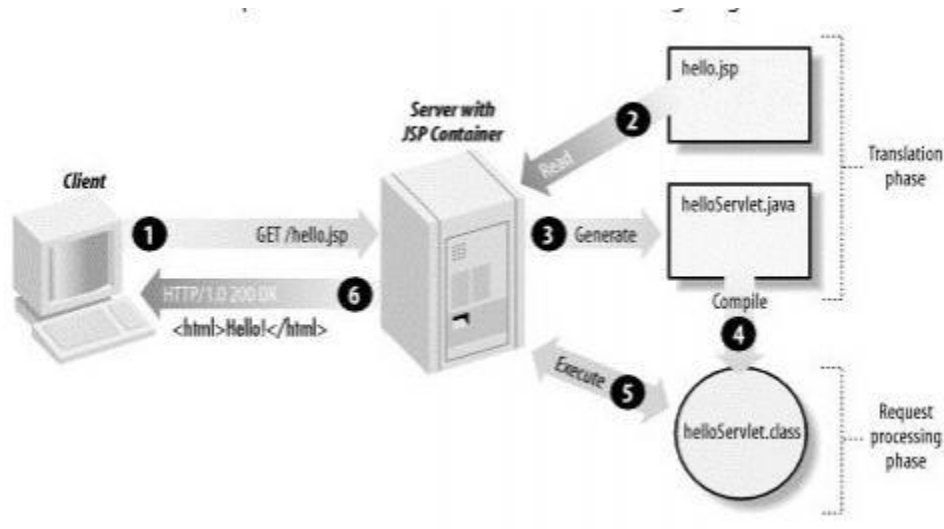
- Được xử lý trên server (server side)
- Được thực thi trên các web (application) server như Tomcat, JBoss, Sun, JOnAS, ...
- Sử dụng ngôn ngữ Java
- Ngôn ngữ lập trình có phân biệt chữ hoa – thường (case sensitive) và hỗ trợ các thành phần trong xử lý lập trình (JSP tag, scripting ..)
- Tách biệt giữa giao diện và phần xử lý
- **Ưu điểm**
  - Duy trì trạng thái kết nối giữa client và server.
  - Khả năng mở rộng cao.
  - Có tính uyển chuyển.
  - Tạo thread mới khi có request.
  - Làm việc trên nhiều platform (OS) khác nhau.
  - Có tính bảo mật tốt

**Ví dụ: Trang JSP đơn giản hiển thị ngày tháng hiện hành**

```
<html>
<head>
  <title>A simple date</title>
</head>
<body>
  The time on the server is <%= new java.util.Date() %>
</body>
</html>
```

=> Server xử lý thành phần của JSP để chuyển đổi thành dữ liệu tĩnh trên trang HTML để trình bày trên Web browser

## 6.2 Kiến trúc JSP trong ứng dụng web



Đây là cách thực nạp, dịch và thực thi trang của một trang JSP khi được triệu gọi trên Web Server

## 6.3 Chu trình sống của JSP

Trang JSP có chu trình sống xác định tính từ khi hệ thống đọc biên dịch trang JSP, gọi thực thi và loại bỏ trang ra khỏi bộ nhớ. Chu trình sống của trang JSP gồm có 5 giai đoạn sau:

### 6.3.1 Biên dịch trang

Khi trình duyệt yêu cầu trang JSP, Web server sẽ kiểm tra xem trang JSP đã được biên dịch hay chưa. Nếu chưa biên dịch hoặc đã biên dịch nhưng trang JSP mới vừa thay đổi mã nguồn thì Web Server sẽ thực hiện biên dịch trang JSP. Quá trình biên dịch JSP thực tế là chuyển trang JSP thành servlet. File biên dịch `.class` của trang chỉ diễn ra một lần. Nếu trang đã biên dịch và sau đó không bị thay đổi trong mã nguồn thì quá trình biên dịch sẽ không xảy ra nữa, do đó mà tốc độ thực thi sẽ nhanh hơn. Sau khi biên dịch, mã trang sẽ được nạp vào bộ nhớ để thực thi

Quá trình biên dịch trang JSP sẽ được diễn ra như sau:

**Bước 1:** Kiểm tra xem trang đã được dịch thành mã nguồn tương đương servlet hay chưa.



**Bước 2:** Nếu chưa được biên dịch thì trang JSP sẽ được biên dịch thành file nguồn .java theo cấu trúc của servlet. Gọi trình biên dịch javac biên dịch file nguồn .java thành file thực thi của servlet .class.

**Bước 3:** Nạp servlet đã biên dịch ở bước 2, thực thi trả kết quả về cho trình khách.

**Bước 4:** Nếu file JSP đã được biên dịch trước đó : thực hiện kiểm tra xem nội dung file .jsp có thay đổi không, Nếu có thì quay lại bước 2 biên dịch lại trang, nếu không thì quay lại bước 3.

### 6.3.2 Nạp trang

Kể từ giai đoạn này, quá trình nạp trang tương tự như servlet (trang JSP sau khi biên dịch có thể coi như một servlet). Chỉ có một điểm khác là servlet chỉ được nạp một lần trong khi mã trang JSP mặc dù đã biên dịch nhưng phải nạp lại nhiều lần mỗi khi web server nhận được yêu cầu trang từ trình duyệt.

### 6.3.3 Khởi tạo

Khi nạp mã trang thành công, Web server sẽ gọi đến phương thức khởi tạo trang. Và mặc dù JSP được biên dịch ra servlet nhưng phương thức khởi tạo cho trang JSP lại mang tên là jspInit() chứ không phải là init() như servlet.

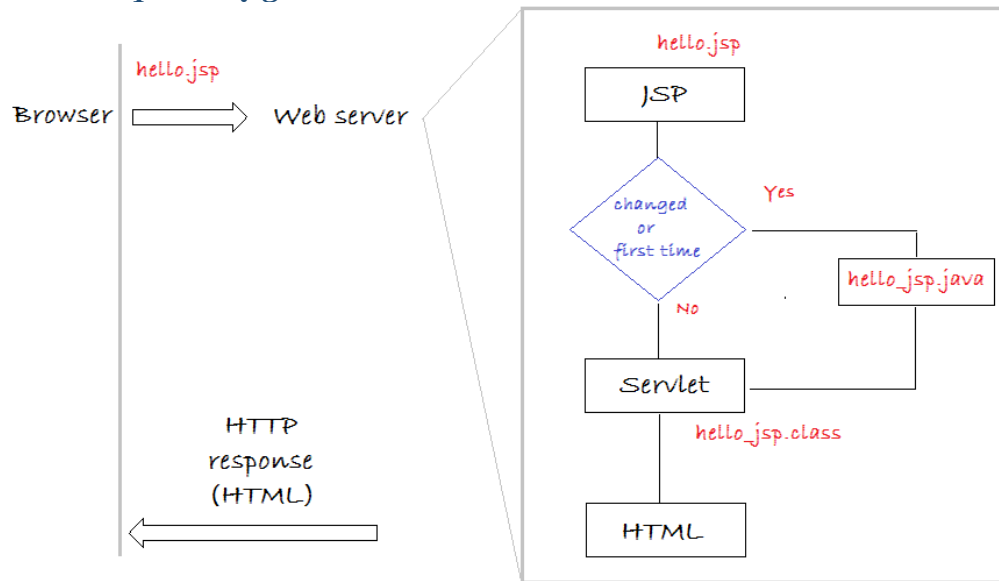
### 6.3.4 Thực thi

Sau quá trình khởi tạo, Web server sẽ gọi đến phương thức \_jspService (khác với servlet gọi đến doPost(), doGet() hoặc service()). Phương thức \_jspService sẽ chuyển đến hai lớp đối tượng HttpServletRequest và HttpServletResponse để đọc và ghi kết xuất trả về trình khách.

### 6.3.5 Dọn dẹp

Khi trang JSP đã thực thi xong, trình chủ Web Server sẽ gọi phương thức jspDestroy() để giải phóng mã trang khỏi bộ nhớ. Tương tự như trong Servlet, có thể cài đặt phương thức jspDestroy() thực hiện giải phóng vùng nhớ hoặc đóng kết nối trả về tài nguyên cho hệ thống.

## 6.4 Mối quan hệ giữa JSP và Servlet



Khi người dùng gửi một request tới một trang JSP. Ví dụ hello.jsp:

Trong lần đầu tiên Web Server sẽ chuyển trang hello.jsp thành file hello\_jsp.java và dịch nó thành file class hello\_jsp.class. Đây là một Servlet, nó sẽ tạo ra HTML trả về phía người dùng.

Trong các lần request thứ 2 trở đi nó sẽ kiểm tra file hello.jsp có gì thay đổi không, nếu không thay đổi nó gọi tới servlet ( hello\_jsp.class) và trả dữ liệu HTML về phía người dùng. Trong trường hợp có thay đổi nó sẽ tạo lại file hello\_jsp.java và biên dịch lại ra file hello\_jsp.class.

Như vậy khi bạn thay đổi file jsp bạn không cần phải chạy lại Web Server

## 6.5 Các tags trong JSP

- Tags
  - Interface
  - Functional
  - Encapsulation
  - Cấu trúc: bắt đầu là “<” và kết thúc là “>” bên trong tag chứa các nội dung, thành phần và thuộc tính
- Có 4 loại tags nhưng trong trang JSP cấu thành nên trang JSP

- Comments
- Directives
- Scripting elements
- Standard actions

### 6.5.1 Comments

- Chú thích, giải thích
- Servlet không biên dịch thành phần này
- Trang JSP chứa 02 loại comments là JSP và HTML
  - HTML comments được gán vào response để gửi đến server
  - JSP chỉ hiển thị trong source của JSP
- Không trình bày trên đối tượng kết xuất
- Cú pháp
  - **JSP comments:** `<%-- comments --%>`
  - Ví dụ: `<%-- chú thích của JSP --%>`
  - **HTML comments:** `<!-- comments -->`
  - Ví dụ: `<!-- chú thích HTML -->`

### 6.5.2 Directives

- Định hướng xử lý trên trang JSP và cách thức vận hành của servlet.
- Cung cấp các thông tin JSP container
- Xác định ngôn ngữ scripting được sử dụng (language)
  - Ví dụ: `<% @ page language = "java" ...%>`
- Xác định custom tags sử dụng (taglib)
  - Ví dụ: `<% @ taglib uri = "c:\..." prefix = "abc" %>`
- Include các trang JSP khác vào trang hiện hành (include)
  - Ví dụ: `<% @ include file = "c:\..." %>`
- Import thư viện sử dụng (import)
  - Ví dụ: `<% @ page .... import = "java.util.*, java.lang.*" %>`

- Xác định xuất khi có lỗi xảy ra trên trang JSP (error page) và các trang JSP được qui định là trang lỗi (isErrorPage).
  - Ví dụ: định hướng <% @ page ... errorpage = “/error.jsp” ... %>
  - trang lỗi <% @ page isErrorPage = “true” %>
- Scope: trên toàn trang JSP
- Không đưa ra trong kết xuất khi có request trên trang JSP
- Cú pháp
  - <% @ directiveName attribute value %>
- Có 3 loại directives:
  - Page
    - Include
    - Taglib

## Page Directive

- Các thuộc tính có hiệu lực trên phạm vi 1 trang
- <% @ page **attributes** %>

Thuộc tính	Mô tả
language	Xác định ngôn ngữ scripting dùng trên page. Mặc định là Java
extends	Thay đổi nội dung của servlet được phát sinh.
import	Cung cấp các thư viện hỗ trợ phương thức sử dụng trên trang
session	Xác định đối với JSP tham gia vào http session. Mặc định là true
buffer	Xác định kích thước của page buffer. Mặc định là 8KB
autoflush	Xóa sạch page buffer khi nó đầy. Mặc định là true
isThreadSafe	Xác định mức độ an toàn cho các thread trên page sao cho việc xử lý tối ưu, ngược lại các xử lý được xếp tuần tự (false). Giá trị mặc định là true.
info	Mô tả thông tin liên quan đến trang JSP
errorPage	Xác định trang báo lỗi khi có lỗi phát sinh
isErrorPage	Xác định trang xử lý lỗi
contentType	Xác định nội dung trình bày của trang (định dạng và ký tự). Mặc định là text/html

## INCLUDE & TAGLIB DIRECTIVE

- Include
  - Mở rộng nội dung của trang JSP hiện hành
  - Xác định URL của file include
  - Có thể include nhiều file trên cùng trang
  - `<%@ include file = “URL” %>`
- Taglib
  - Sử dụng các custom tags
  - Xác định URL chứa các thư viện tag và tên dùng để truy cập trên trang JSP
  - `<%@ taglib uri = “URL” prefix = “name” %>`

### 6.5.3 Scripting elements

- Các hoạt động xử lý trên server
- Cho phép nhúng trực tiếp code vào trang JSP
- Declarations:
  - Định nghĩa việc khai báo các biến và phương thức
  - `<%! Declaration; %>`
  - Ví dụ: `<%! String s = “Aptech” %>`
- Scriptlets
  - Block code (định nghĩa theo cú pháp của language scripting trong directive) nhúng trên trang JSP thực hiện xử lý tác vụ
  - `<% scriptlet %>`
  - Ví dụ:
- `<% for (int i =0 ; i<n; i++){`
  - `System.out.println(i + “.Đây là scriptlets.”); } %>`
- Expressions
  - Biểu thức lấy trị trả về, duy nhất hàng code

- `<%= expression %>`
- Ví dụ: `<%= i %>`

## VÍ DỤ

```
<%@ page import="java.util.Date" %>
```

```
<html>
```

```
<head>
```

```
<title>First JSP program</title>
```

```
</head>
```

```
<body>
```

```
<!-- myFirstProgram.jsp -->
```

```
<% out.println("Hello there!"); %><br>
```

```
<%= "Current date is " + new Date() %>
```

```
</body>
```

```
</html>
```

Directives - page

JSP Comments

Scriptlet

Expression

## VÍ DỤ

### • Main file

```
<html>
```

```
<head>
```

```
<title>Directive Includes JSP program</title>
```

```
</head>
```

```
<body>
```

```
<!-- include use directives -->
```

```
Current date is
```

```
<%@ include file = "myDate.jsp" %>
```

```
</body>
```

```
</html>
```

### • Include file

```
<%@ page import="java.util.Date" %>
```

```
<html>
```

```
<head>
```

```
<title>Date JSP program</title>
```

```
</head>
```

```
<body>
```

```
<%= new Date().toLocaleString() %>
```

```
</body>
```

```
</html>
```

JSP Comments

Directives - include

Directives - page

Expression

#### 6.5.4 Standard actions

- Là một directive đặc biệt được đóng gói cung cấp các phương thức hỗ trợ xử lý trên trang JSP
- Hoạt động runtime
- Chuyển điều khiển từ trang này sang trang khác

Actions	Mô tả
<jsp:useBean>	Truy cập thành phần của java Bean, khởi tạo instance của JB
<jsp:setProperty>	Thiết lập giá trị thuộc tính trong Java Bean
<jsp:getProperty>	Lấy giá trị thuộc tính trong Java Bean
<jsp:param>	Cung cấp tên và giá trị thuộc tính khi gọi trang JSP khác, được sử dụng trong <jsp:include>, <jsp:forward> và <jsp:plugin>
<jsp:include>	Includes output từ một file đến file khác
<jsp:forward>	Chuyển điều khiển từ 1 trang đến trang khác trên server
<jsp:plugin>	Thực thi applet hay Java Bean. Đây là dạng kết hợp với chương trình khác

#### IMPLICIT OBJECTS (IB)

- Không cần khai báo vẫn sử dụng được
- Sử dụng trong scriptlets và expression
- Cách thức truy cập: **ImplicitObject.method(params)**
- Có 4 loại
  - Kết hợp với Servlet
  - Nhập xuất trên JSP
  - Thông tin về context trên JSP
  - Điều khiển xử lý lỗi

## IB – KẾT HỢP VỚI SERVLET

Objects	Mô tả
page	<ul style="list-style-type: none"><li>- Đại diện cho servlets cùng các tham số khởi tạo lưu trữ trong config</li><li>- Dùng “this” để truy cập đối tượng này trên JSP</li><li>- Được implement từ java.lang.object</li><li>- &lt;%@ page info = “information” %&gt;</li><li>- Scope: page</li></ul>
config	<ul style="list-style-type: none"><li>- Trình bày các thông tin trong cấu hình servlet</li><li>- Được implement từ javax.Servlet.ServletConfig</li><li>- Truy cập đối tượng config.getInitParameter(“tên tham số”)</li><li>- Scope: page</li></ul>

## IB – NHẬP & XUẤT TRÊN JSP

Objects	Mô tả
request	<ul style="list-style-type: none"><li>- Đại diện cho dữ liệu nhập cho JSP được client chuyển cho server xử lý thông qua việc chuyển tham số cho services của servlet xử lý.</li><li>- Được implement từ javax.servlet.http.HttpServletRequest</li><li>- Truy cập đối tượng request.method(params)</li><li>- Scope: request</li><li>- Ví dụ : request.getParameter(“username”);</li></ul>
response	<ul style="list-style-type: none"><li>- Đại diện cho kết quả trả về gửi lại cho Web Browser sau khi đã nhận và xử lý request.</li><li>- Được implement từ javax.servlet.http.HttpServletResponse</li><li>- Truy cập đối tượng response.method(params)</li><li>- Scope: page</li><li>- Ví dụ : response.addCookie(cookie)</li></ul>
out	<ul style="list-style-type: none"><li>- Đại diện luồng xuất trên trang JSP</li><li>- Được implement từ javax.servlet.jsp.JspWriter (buffer của PrintWriter hỗ trợ của Servlet)</li><li>- Truy cập đối tượng out.method(params)</li><li>- Scope: page</li><li>- Ví dụ : out.println(“output stream”)</li></ul>



## IB – THÔNG TIN CONTEXT TRÊN JSP

Objects	Mô tả
session	<ul style="list-style-type: none"> <li>- Xác định dữ liệu trong session hiện hành và lưu trữ các thông tin liên quan đến session.</li> <li>- Được implement từ javax.servlet.http.HttpSession</li> <li>- Truy cập đối tượng session.method(params)</li> <li>- Scope: session</li> <li>- Ví dụ : session.setAttribute("username", "Aptech");</li> </ul>
application	<ul style="list-style-type: none"> <li>- Đại diện application và dữ liệu lưu trữ trong đối tượng này có thể sử dụng thông qua các trang JSP – trên toàn ứng dụng</li> <li>- Được implement từ javax.servlet.ServletContext</li> <li>- Truy cập đối tượng application.method(params)</li> <li>- Scope: application</li> <li>- Ví dụ : application.setAttribute("username", "Aptech");</li> </ul>
pageContext	<ul style="list-style-type: none"> <li>- Đại diện instance cho context của Pages (javax.jsp.PageContext)</li> <li>- Truy cập các thuộc tính và thành phần của page</li> <li>- Truy cập đối tượng pageContext.method(params)</li> <li>- Ví dụ: pageContext.getAttributes("username");</li> </ul>

## IB – XỬ LÝ LỖI

Objects	Mô tả
exception	<ul style="list-style-type: none"> <li>- Điều khiển các lỗi trên trang JSP</li> <li>- Được sử dụng trên các trang có khai báo các directives là isErrorPage</li> <li>- Được vận dụng từ javax.lang.Throwable</li> <li>- Các phương thức hỗ trợ <ul style="list-style-type: none"> <li>+ getMessage() : trả về message báo lỗi</li> <li>+ toString() : trả về class name cùng thông báo lỗi</li> </ul> </li> </ul>

## 7. Servlet

### 7.1 Servlets là gì?

Java Servlets là các chương trình chạy trên một Web server hoặc một Application server và thực hiện như là **một tầng trung gian** giữa một Yêu cầu từ một trình duyệt web hoặc HTTP client với các Database hoặc các ứng dụng trên HTTP server.

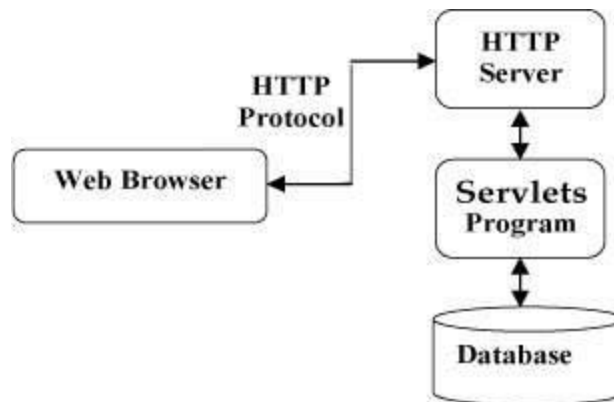
Sử dụng Servlets, bạn có thể thu thập Input từ người sử dụng thông qua các form trên trang web, hiển thị các bản ghi (record) từ một Database hoặc từ nguồn khác, và tạo các trang web động.

Java Servlets thường có chung mục đích: là các chương trình độc lập bởi việc sử dụng Common Gateway Interface (CGI). Nhưng so với CGI, thì Servlets có các lợi thế sau:

- Hiệu năng tốt hơn đáng kể.
- Servlets thực thi bên trong không gian địa chỉ của một Web server, không cần thiết phải tạo một tiến trình riêng biệt để xử lý mỗi yêu cầu từ Client.
- Servlets là độc lập trên nền tảng bởi vì chúng được viết bằng Java.
- Bởi vì viết bằng Java (mà Java cung cấp tính bảo mật cao trên Server) nên Servlets là đáng tin cậy.
- Tính năng đầy đủ của thư viện của các lớp trong Java là luôn luôn có sẵn cho Servlets. Nó có thể giao tiếp với Applet, Database hoặc phần mềm khác thông qua các Socket và kỹ thuật RMI mà bạn đã từng thấy.

### 7.2 Cấu trúc của Servlets

Sơ đồ dưới đây minh họa cấu trúc của Servlets trong một ứng dụng web:



### 7.3 Nhiệm vụ của Servlets

Servlets thực hiện các tác vụ chủ yếu sau:

- Đọc dữ liệu hiển thị (explicit) được gửi bởi Client (hoặc trình duyệt) bao gồm một HTML Form trên một trang web hoặc nó cũng có thể từ một Applet hoặc một chương trình Custom từ HTTP Client.
- Đọc dữ liệu yêu cầu HTTP ẩn (implicit) được gửi bởi Client (hoặc trình duyệt) bao gồm cookie, các loại media.
- Xử lý dữ liệu và cho ra kết quả. Tiến trình này có thể yêu cầu Database, đang thực thi một triệu hồi tới RMI hoặc CORBA, triệu hồi một Web Service, hoặc tính toán phản hồi một cách trực tiếp.
- Gửi dữ liệu hiển thị (ví dụ: tài liệu) tới các Client (hoặc trình duyệt). Tài liệu này có thể được gửi theo nhiều định dạng khác nhau, gồm text (HTML hoặc XML), nhị phân (hình ảnh GIF), Excel, .v.v.
- Gửi phản hồi HTTP ẩn tới các Client (hoặc trình duyệt), thông báo cho trình duyệt hoặc Client về kiểu của tài liệu được trả về (ví dụ: HTML), thiết lập các Cookie và Caching các tham số, cùng các tác vụ khác.

### 7.4 Package trong Servlets

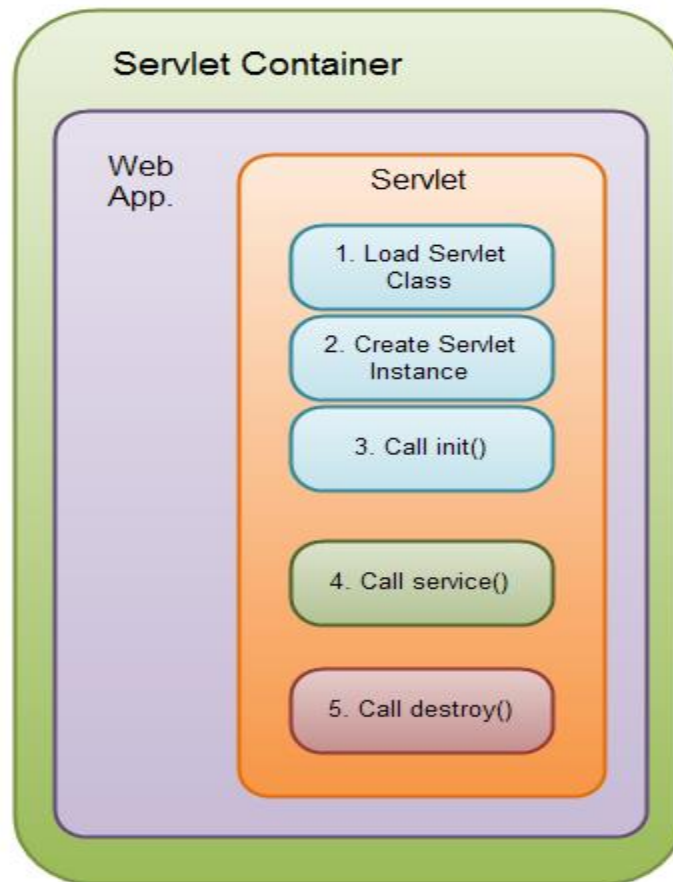
Java Servlets là các lớp trong Java chạy bởi một Web Server mà có một trình thông dịch hỗ trợ Java Servlets.

Servlets có thể được tạo bởi sử dụng các gói **javax.servlet** và **javax.servlet.http** là một phiên bản mở rộng của thư viện lớp Java để hỗ trợ các dự án phát triển có phạm vi lớn. Các lớp này triển khai Java Servlet và JSP.

Java Servlet đã được tạo và được biên dịch giống như các lớp khác trong Java. Sau khi bạn cài đặt các gói servlet và thêm chúng vào Classpath trong máy của bạn, bạn có thể biên dịch Servlet với bộ biên dịch JDK hoặc bất kỳ bộ biên dịch nào khác.

### 7.5 Vòng đời của Servlet

Hình dưới đây minh họa về vòng đời của một Servlet. Từ khi nó được tạo ra, xử lý các đòi hỏi từ người dùng, cho tới lúc nó bị hủy.



Có 5 bước:

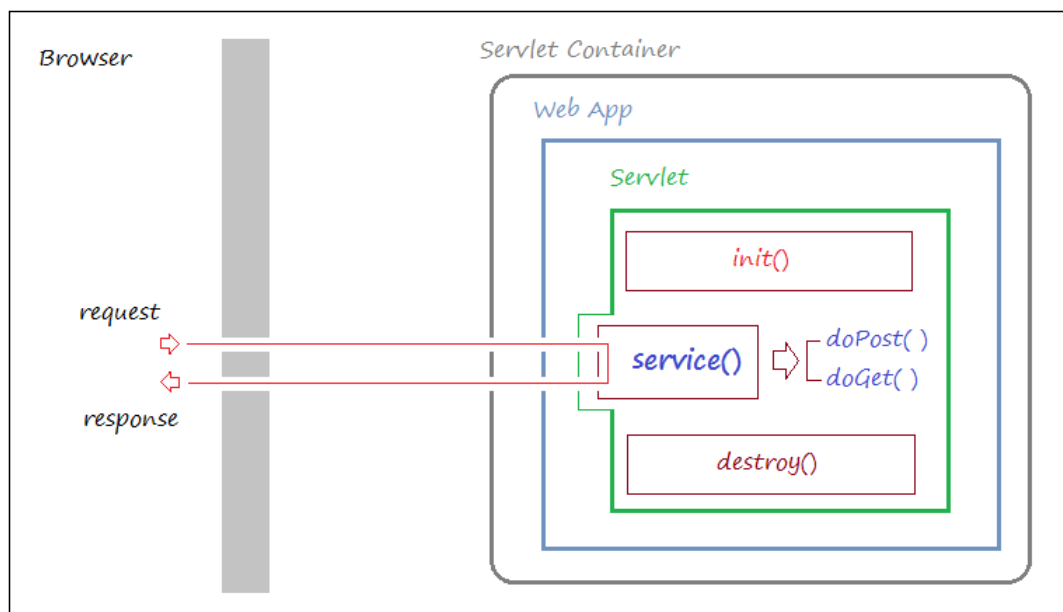
- Tải Servlet Class vào bộ nhớ.
- Tạo đối tượng Servlet.

- Gọi method servlets `init()`
- Gọi method servlets `service()`.
- Gọi method servlets `destroy()`.

Bước 1, 2 và 3 được thực thi một lần duy nhất, khi mà servlet được nạp lần đầu. Mặc định các servlet không được tải lên cho tới khi nó nhận một đòi hỏi đầu tiên từ người dùng. Bạn có thể buộc ServletContainer (Bộ chứa các servlet) tải các servlet khi nó khởi động.

Bước 4 được thực thi nhiều lần, mỗi khi có đòi hỏi từ phía người dùng tới servlet. Bước 5 nó được thực thi khi bộ chứa servlet (Servlet Container) trút bỏ (unloaded) servlet.

Bạn có thể xem hình minh họa tiếp theo để hiểu hơn về vòng đời của Servlet.



Khi yêu cầu (request) của người dùng tới Servlet, servlet sẽ gọi phương thức `service()` để phục vụ yêu cầu của người dùng, `service()` sẽ gọi một trong hai phương thức `doGet()` hoặc `doPost()`. Trong servlet của bạn, bạn cần ghi đè và xử lý tại các phương thức này.

Như vậy khi người dùng gửi yêu cầu một Servlet, servlet sẽ được tạo ra tại thời điểm có yêu cầu lần đầu tiên tới, đồng thời sẽ gọi phương thức `init()` của servlet để khởi tạo cho nó, `init()` được gọi duy nhất 1 lần. Phương thức `destroy()` dùng để hủy servlet, nó sẽ

được gọi một lần duy nhất khi gỡ bỏ triển khai (undeloy) ứng dụng web hoặc tắt (shutdown) web server.

## 8. MVC model 2

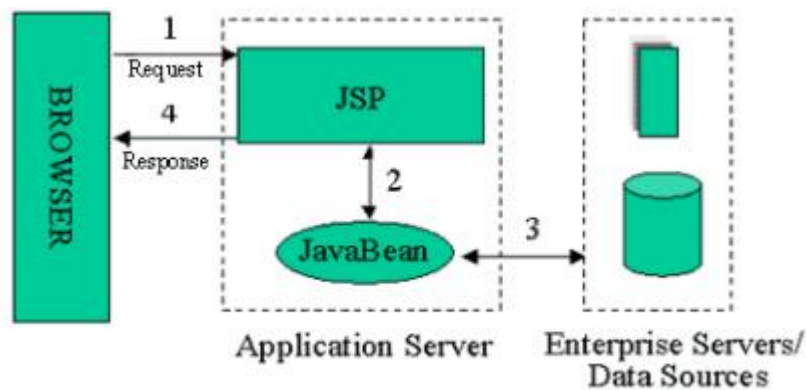
### 8.1. JSP Model 1 là gì?

Nó là một kiến trúc truyền thống (Còn gọi là mô hình 1 – Model 1)

Kiến trúc Model 1 hết sức đơn giản. Tóm lại tất cả mọi thứ đều được gói gọn trong Servlet hoặc JSP từ việc xử lý request, xác nhận tính hợp lệ của dữ liệu, điều quản business logic và generate response.

Mặc dù về khái niệm hết sức đơn giản và dễ hiểu nhưng kiến trúc này không phù hợp cho việc phát triển các ứng dụng quy mô lớn vì chắc chắn 1 số lượng lớn các tính năng sẽ bị lặp lại trong các trang JSP (lặp lại code). Cũng vậy, kiến trúc Model 1 tạo nên sự ràng buộc không cần thiết giữa business logic và presentation logic của ứng dụng.

Chúng ta có thể xem qua kiến trúc của nó:



Như trên mô hình chúng ta thấy thì mọi công việc về xử lý dữ liệu đều được gói gọn trong trang JSP từ việc báo lỗi đến trả về dữ liệu cho người dùng và điều này rất nguy hiểm vì nó dẫn tới việc lặp lại code không cần thiết.

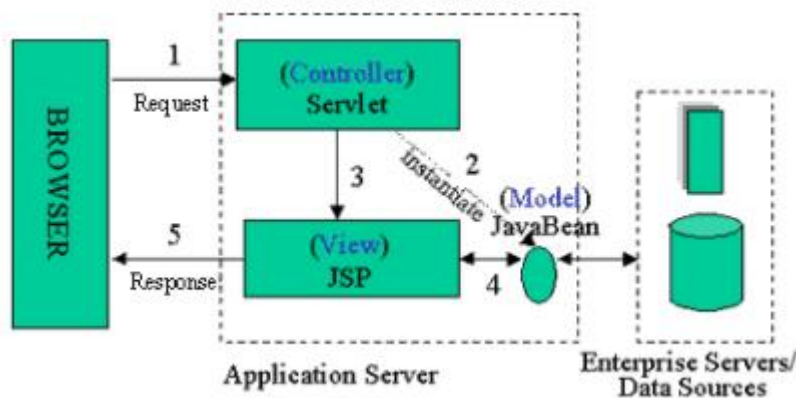
## 8.2. JSP Model 2

Sau khi nhận thấy khá nhiều điểm bất lợi của JSP Model 1 – java sun đã ra phiên bản tiếp theo nó JSP Model 2 hay còn gọi là mô hình MVC (Model – View - Controller)



JSP Model 2 ra đời và giải quyết nhiều vấn đề của JSP Model 1 với việc cung cấp 1 sự phân tách rõ ràng trong ứng dụng.

Trong kiến trúc MVC, một Servlet trung tâm, được gọi là Controller, tiếp nhận tất cả các request cho ứng dụng. Sau đó controller sẽ xử lý request và làm việc với model để chuẩn bị những dữ liệu cần thiết cho View (thường là JSP) và forward dữ liệu tới trang JSP.



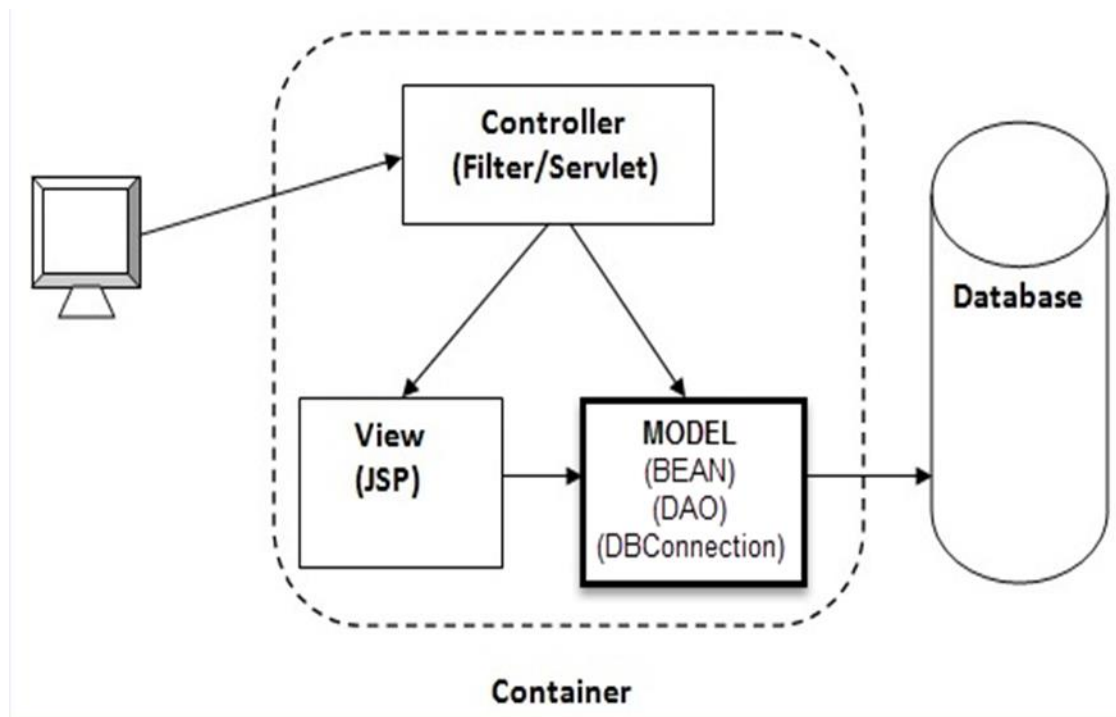
Sau đó JSP sử dụng các dữ liệu đã được chuẩn bị bởi Controller để generate một response về cho browser. Trong kiến trúc này, business logic và presentation logic được phân tách nhau. Việc tách business code là presentation code giúp ta có thể sử dụng nhiều giao diện cho ứng dụng, chúng có thể là web, wireless hay GUI. Thêm vào đó, việc phân tách này còn cung cấp việc tái sử dụng lại code 1 cách hoàn hảo.

# PHẦN 3: THIẾT KẾ - CÀI ĐẶT

## 1. KIẾN TRÚC HỆ THỐNG

### MVC Model 2 trong công nghệ J2EE:

- MVC (Model View Controller) là 1 design pattern giúp phân tách tầng Business Logic, tầng Presentation, tầng Data Access 1 cách riêng biệt
- View: tầng hiển thị giao diện, còn gọi là tầng presentation layer
- Model: tầng chứa các hàm thực thi, chứa kết nối cơ sở dữ liệu
- Controller: tầng kết nối 2 tầng view với model
- Hình dưới là mô hình MVC sử dụng trong lập trình JSP – Servlet

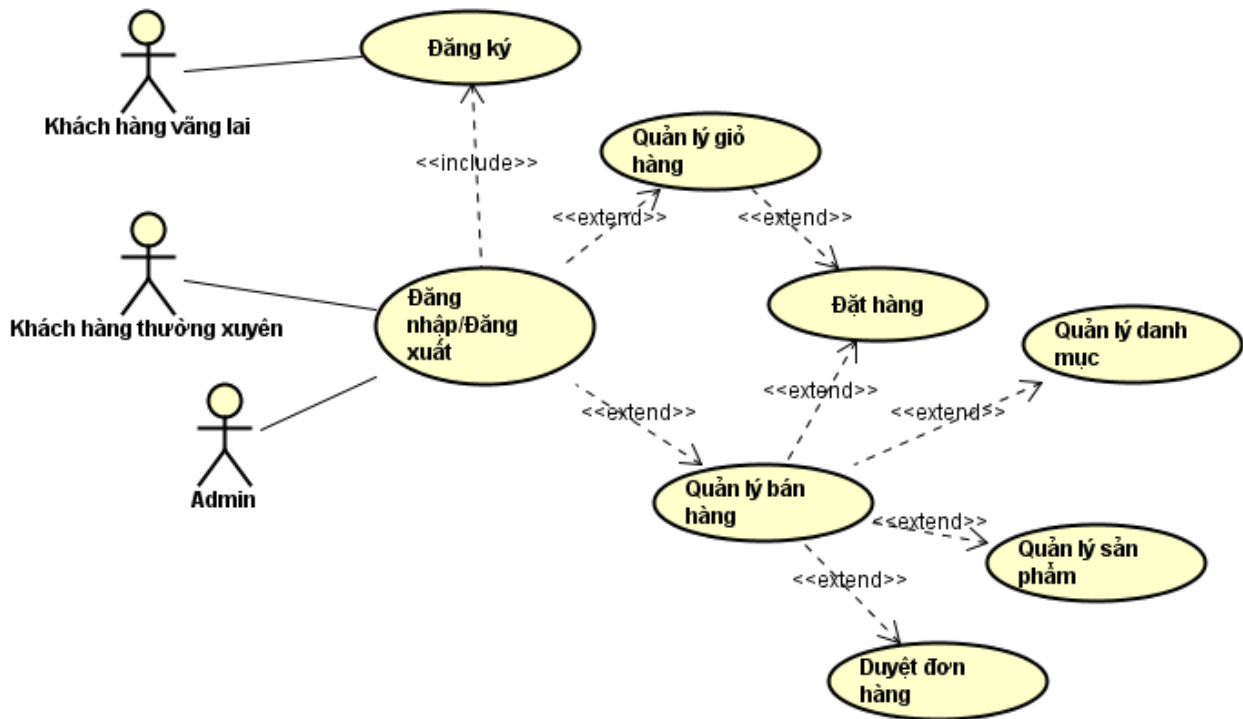


- Controller do Servlet đảm nhiệm nên MVC dễ dàng viết và chỉnh sửa code java
  - View do jsp đảm nhiệm nên dễ dàng thiết kế và chỉnh sửa giao diện
- Phân tách rõ ràng 3 tầng DAL, BAL, Presentation



## 2. USECASE

### a. SƠ ĐỒ USE-CASE



### b. DANH SÁCH CÁC ACTOR

STT	Tên Actor	Mô tả
1	Admin	Chịu trách nhiệm quản lý hệ thống
2	Khách hàng thành viên	Là khách hàng đã đăng ký tài khoản
3	Khách hàng tiềm năng	Là khách hàng vắng lai chưa có tài khoản nhưng vẫn có thể mua được sản phẩm bằng cách đăng ký tài khoản hoặc nhập thông tin đặt hàng

### c. DANH SÁCH CÁC USE-CASE

STT	Tên Use-case
1	Đăng nhập/Đăng xuất
2	Đăng ký thành viên
4	Quản lý nhập hàng
5	Quản lý sản phẩm
7	Quản lý danh mục
8	Quản lý thông tin nhân viên
9	Quản lý khách hàng thân thiết
10	Đặt hàng
11	Duyệt đơn hàng
14	Quản lý giỏ hàng

## d. ĐẶC TẢ USECASE

### i. Đăng nhập/Đăng xuất:

Đăng nhập:

<b>Tóm tắt:</b> Chức năng đăng nhập vào hệ thống
<b>Dòng sự kiện chính:</b> <ol style="list-style-type: none"><li>1) Người dùng bấm vào nút “Đăng nhập” ở phía trên bên phải màn hình</li><li>2) Hệ thống hiển thị form đăng nhập</li><li>3) Người dùng nhập tên và tài khoản (cả 2 trường này đều bắt buộc nhập) và nhấn nút “Đăng nhập”</li><li>4) Hệ thống kiểm tra thông tin đăng nhập (Dòng sự kiện khác:<ul style="list-style-type: none"><li>- Thông tin đăng nhập không đúng</li><li>- Trường Email Address hoặc Password bị bỏ trống)</li></ul>)</li><li>5) Người dùng đăng nhập vào hệ thống, hệ thống hiển thị trang chủ</li></ol>
<b>Dòng sự kiện phụ:</b> <ol style="list-style-type: none"><li>1) Thông tin đăng nhập không đúng: Hệ thống hiển thị thông báo tài khoản đăng nhập không hợp lệ và yêu cầu nhập lại.</li><li>2) Các trường “Email Address” và “Password” bị để trống: Hệ thống hiển thị thông báo “Vui lòng điền vào trường này” và yêu cầu nhập lại</li></ol>
<b>Tiền điều kiện:</b> <p>Actor: Tất cả các actor ngoại trừ khách hàng tiềm năng</p> <p>Điều kiện: Người dùng phải có tài khoản</p>
<b>Hậu điều kiện:</b> <p>Actor: Tất cả các actor ngoại trừ khách hàng tiềm năng</p> <p>Điều kiện: Người dùng phải có tài khoản</p>

Đăng xuất:

<b>Tóm tắt:</b> Chức năng đăng xuất khỏi hệ thống
<b>Dòng sự kiện chính:</b> 1) Người dùng bấm vào nút “Đăng xuất” ở phía trên bên phải màn hình 2) Người dùng đăng xuất khỏi hệ thống và trở về trang chủ
<b>Dòng sự kiện phụ:</b> Không có
<b>Tiền điều kiện:</b> Actor: Tất cả các actor ngoại trừ khách hàng tiềm năng Điều kiện: Người dùng đã đăng nhập
<b>Hậu điều kiện:</b> Người dùng đăng xuất khỏi hệ thống

## ii. Đăng ký thành viên:

<b>Tóm tắt:</b> Chức năng đăng ký tài khoản cho hệ thống
<b>Dòng sự kiện chính:</b> 1) Người dùng bấm nút “Đăng ký” ở góc trên bên phải màn hình 2) Hệ thống hiển thị form đăng ký 3) Người dùng nhập thông tin vào các trường và bấm nút “Xác nhận” 4) Hệ thống kiểm tra các thông tin đã nhập (Dòng sự kiện khác : - Một số trường bị bỏ trống - Thông tin trong các trường được nhập chưa đúng điều kiện) 5) Hệ thống tạo một tài khoản mới với thông tin vừa nhập và hiển thị thông báo “Thêm thành công”
<b>Dòng sự kiện phụ:</b> 1) Một số trường bị bỏ trống: Hiển thị thông báo trường đó không được bỏ trống và yêu cầu nhập lại 2) Thông tin các trường được nhập chưa đúng điều kiện : Hiển thị thông báo các trường đó chưa hợp lệ và yêu cầu nhập lại
<b>Tiền điều kiện:</b> Actor: Khách hàng tiềm năng Điều kiện: Không có

**Hậu điều kiện:**

Tài khoản mới được tạo cho khách hàng

**iii. Quản lý sản phẩm**

Thêm sản phẩm

**Tóm tắt:**

Thêm thông tin của các loại máy tính, linh kiện máy tính: tên sản phẩm, nhà sản xuất, giá bán, ...

**Dòng sự kiện chính:**

- 1) Vào trang quản trị và Chọn chức năng Quản lý sản phẩm
- 2) Bấm nút “Thêm” trong trang Quản lý sản phẩm
- 3) Người dùng nhập những thông tin cần thiết, trong đó có những thông tin bắt buộc và nhấn nút “Lưu”.
- 4) Hệ thống kiểm tra thông tin, nếu thông tin hợp lệ sẽ tiến hành bước tiếp theo.  
(Dòng sự kiện khác: Thông tin không hợp lệ.)
- 5) Hệ thống lưu dữ liệu và thông báo thành công.  
(Dòng sự kiện khác: Không thể cập nhật CSDL)

**Dòng sự kiện phụ:**

- 1) Thông tin không hợp lệ.  
Hệ thống hiển thị thông báo yêu cầu nhập lại thông tin.
- 2) Không thể cập nhật CSDL.  
Lỗi trong lúc cập nhật: Yêu cầu người dùng nhập lại thông tin, nếu vẫn bị lỗi nên liên hệ với nhà cung cấp phần mềm.  
Dữ liệu đã thêm có đã có trong CSDL: Hiển thị thông báo sản phẩm đã tồn tại trong hệ thống, Yêu cầu kiểm tra lại thông tin sản phẩm.

**Tiền điều kiện:**

Actor: Chủ cửa hàng.

Điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền sử dụng chức năng này.

**Hậu điều kiện:**

Người dùng thêm thông tin sản phẩm thành công.

Xóa sản phẩm

**Tóm tắt:**

Xóa thông tin của các loại máy tính, linh kiện máy tính.

**Dòng sự kiện chính:**

- 1) Vào trang quản trị và Chọn chức năng Quản lý sản phẩm
- 2) Chọn sản phẩm cần xóa và bấm nút “Xóa”.
- 3) Hệ thống tiến hành xóa dữ liệu và thông báo thành công.

**Dòng sự kiện phụ:**

**Tiền điều kiện:**

Actor: Chủ cửa hàng.

Điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền sử dụng chức năng này.

**Hậu điều kiện:**

Người dùng xóa sản phẩm thành công.

Cập nhật sản phẩm

**Tóm tắt:**

Cập nhật thông tin của các loại máy tính, linh kiện máy tính: tên sản phẩm, nhà sản xuất, giá bán, ...

**Dòng sự kiện chính:**

- 1) Vào trang quản trị và Chọn chức năng Quản lý sản phẩm
- 2) Nháy đúp vào sản phẩm cần cập nhật.
- 3) Người dùng cập nhật những thông tin cần thiết, trong đó có những thông tin không được phép cập nhật và nhấn nút “Lưu”.
- 4) Hệ thống kiểm tra thông tin, nếu thông tin hợp lệ sẽ tiến hành bước tiếp theo.

<p>(Dòng sự kiện khác: Thông tin không hợp lệ.)</p> <p>5) Hệ thống lưu dữ liệu và thông báo thành công.</p> <p>(Dòng sự kiện khác: Không thể cập nhật CSDL)</p>
<p><b>Dòng sự kiện phụ:</b></p> <p>1) Thông tin không hợp lệ.</p> <p>Hệ thống hiển thị thông báo yêu cầu nhập lại thông tin.</p> <p>2) Không thể cập nhật CSDL.</p> <p>Lỗi trong lúc cập nhật: Yêu cầu người dùng nhập lại thông tin, nếu vẫn bị lỗi nên liên hệ với nhà cung cấp phần mềm.</p>
<p><b>Tiền điều kiện:</b></p> <p>Actor: Chủ cửa hàng.</p> <p>Điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền sử dụng chức năng này.</p>
<p><b>Hậu điều kiện:</b></p> <p>Người dùng cập nhật thông tin sản phẩm thành công.</p>

#### iv. Quản lý danh mục

Thêm danh mục

<p><b>Tóm tắt:</b></p> <p>Thêm thông tin của các danh mục</p>
<p><b>Dòng sự kiện chính:</b></p> <p>1) Vào trang quản trị và Chọn chức năng Quản lý danh mục</p> <p>2) Bấm nút “Thêm” trong trang Quản lý danh mục</p> <p>3) Người dùng nhập những thông tin cần thiết, trong đó có những thông tin bắt buộc và nhấn nút “Lưu”.</p> <p>4) Hệ thống kiểm tra thông tin, nếu thông tin hợp lệ sẽ tiến hành bước tiếp theo.</p> <p>(Dòng sự kiện khác: Thông tin không hợp lệ.)</p> <p>5) Hệ thống lưu dữ liệu và thông báo thành công.</p> <p>(Dòng sự kiện khác: Không thể cập nhật CSDL)</p>

**Dòng sự kiện phụ:**

1) Thông tin không hợp lệ.

Hệ thống hiển thị thông báo yêu cầu nhập lại thông tin.

2) Không thể cập nhật CSDL.

Lỗi trong lúc cập nhật: Yêu cầu người dùng nhập lại thông tin, nếu vẫn bị lỗi nên liên hệ với nhà cung cấp phần mềm.

Dữ liệu đã thêm có đã có trong CSDL: Hiển thị thông báo nhà cung cấp đã tồn tại trong hệ thống, Yêu cầu kiểm tra lại thông tin nhà cung cấp.

**Tiền điều kiện:**

Actor: Chủ cửa hàng.

Điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền sử dụng chức năng này.

**Hậu điều kiện:**

Người dùng thêm thông tin danh mục thành công.

Xóa danh mục

**Tóm tắt:**

Xóa thông tin của các danh mục

**Dòng sự kiện chính:**

1) Vào trang quản trị và Chọn chức năng Quản lý danh mục

2) Chọn danh mục cần xóa và bấm nút “Xóa”.

3) Hệ thống tiến hành xóa dữ liệu và thông báo thành công.

(Dòng sự kiện khác: Không thể xóa danh mục)

**Dòng sự kiện phụ:**

1) Không thể xóa nhà cung cấp.

Dữ liệu cần xóa đang được sử dụng trong CSDL ở một nơi khác: Hiển thị thông báo không được xóa danh mục này.

**Tiền điều kiện:**

Actor: Chủ cửa hàng.



Điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền sử dụng chức năng này.
<b>Hậu điều kiện:</b> Người dùng xóa danh mục thành công.

Cập nhật danh mục

<b>Tóm tắt:</b> Cập nhật thông tin danh mục
<b>Dòng sự kiện chính:</b> <ol style="list-style-type: none"> <li>1) Vào trang quản trị và Chọn chức năng Quản lý danh mục</li> <li>2) Nháy đúp vào danh mục cần cập nhật.</li> <li>3) Người dùng cập nhật những thông tin cần thiết, trong đó có những thông tin không được phép cập nhật và nhấn nút “Lưu”.</li> <li>4) Hệ thống kiểm tra thông tin, nếu thông tin hợp lệ sẽ tiến hành bước tiếp theo. (Dòng sự kiện khác: Thông tin không hợp lệ.)</li> <li>5) Hệ thống lưu dữ liệu và thông báo thành công. (Dòng sự kiện khác: Không thể cập nhật CSDL)</li> </ol>
<b>Dòng sự kiện phụ:</b> <ol style="list-style-type: none"> <li>1) Thông tin không hợp lệ. Hệ thống hiển thị thông báo yêu cầu nhập lại thông tin.</li> <li>2) Không thể cập nhật CSDL. Lỗi trong lúc cập nhật: Yêu cầu người dùng nhập lại thông tin, nếu vẫn bị lỗi nên liên hệ với nhà cung cấp phần mềm.</li> </ol>
<b>Tiền điều kiện:</b> Actor: Chủ cửa hàng. Điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền sử dụng chức năng này.
<b>Hậu điều kiện:</b> Người dùng cập nhật thông tin danh mục thành công.

## v. Đặt hàng

### **Tóm tắt:**

Quản lý bán hàng: đặt hàng được dùng khi khách hàng tới cửa hàng để mua hàng.

### **Dòng sự kiện chính:**

- 1) Vào trang quản trị.
- 2) Chọn chức năng quản lý bán hàng
- 3) Hệ thống sẽ chuyển đến trang quản lý bán hàng
- 4) Hiển thị menu trong đó có chức năng bán hàng.
- 5) Nhân viên chọn chức năng bán hàng
- 6) Hệ thống sẽ chuyển đến trang bán hàng
- 7) Hệ thống hiển thị bảng danh sách đơn đặt hàng do nhân viên đặt trước đó.
- 8) Nhân viên chọn chức năng tạo mới đơn đặt hàng.
- 9) Hệ thống hiển thị form đặt hàng và nhân viên thực hiện việc đặt hàng cho khách hàng.
- 10) Nhân viên thực hiện submit form lên hệ thống
- 11) Hệ thống hiển thị thông báo thành công và tự động chuyển đến trang bán hàng và cập nhật lại bảng danh sách đơn đặt hàng.
- 12) Nhân viên thi hành chức năng xóa, sửa nếu thông tin đơn đặt hàng sai sót hoặc dư thừa.
- 13) Hệ thống thông báo thành công.
- 14) Nhân viên thoát khỏi trang quản lý bán hàng.

### **Dòng sự kiện phụ:**

- 1) Thi hành chức năng sửa không hợp lệ.
- 2) Hệ thống hiển thị thông báo thi hành chức năng không hợp lệ và yêu cầu phải nhập lại thông tin.
- 3) Thi hành chức năng xóa không hợp lệ.  
Hệ thống hiển thị thông báo xóa đơn đặt hàng không thành công và yêu cầu

<p>xem xét điều kiện thỏa mãn của quy trình xóa đơn đặt hàng ra khỏi hệ thống.</p> <p>4) Thi hành chức năng hiển thị thông tin không hợp lệ.</p> <p>Hệ thống hiển thị bảng danh sách thông tin đơn đặt hàng với các trường dữ liệu không phù hợp, thiếu sót hoặc không hiển thị bảng danh sách thông tin đơn đặt hàng.</p> <p>5) Không thể công nhật CSDL</p> <p>6) Lỗi trong lúc cập nhật: Yêu cầu người dùng nhập lại thông tin, nếu vẫn bị lỗi nên liên hệ với nhà cung cấp phần mềm.</p>
<p><b>Tiền điều kiện:</b></p> <p>Actor: Nhân viên.</p> <p>Điều kiện: Nhân viên phải đăng nhập vào hệ thống</p>
<p><b>Hậu điều kiện:</b></p> <p>Hệ thống sẵn sàng thi hành chức năng đặt hàng, xóa, sửa và hiển thị danh sách đơn đặt hàng.</p>

## vi. Duyệt đơn hàng

### **Tóm tắt:**

Quản lý bán hàng: duyệt đơn hàng gồm chức năng hiển thị danh đơn hàng, duyệt đơn hàng.

### **Dòng sự kiện chính:**

Vào trang quản trị.

- 1) Chọn chức năng quản lý bán hàng
- 2) Hệ thống sẽ chuyển đến trang quản lý bán hàng
- 3) Hiển thị menu trong đó có chức năng duyệt đơn hàng.
- 4) Nhân viên chọn chức năng duyệt đơn hàng.
- 5) Hệ thống sẽ chuyển đến trang duyệt đơn hàng.
- 6) Hệ thống hiển thị bảng danh sách đơn đặt hàng do nhân viên và khách hàng đã đặt trên trang web.
- 7) Mỗi dòng trong bảng danh sách đơn đặt hàng sẽ có một nút duyệt đơn hàng tương ứng.
- 8) Nhân viên thực thi chức năng duyệt đơn hàng
- 9) Hệ thống hiển thị thông báo thành công và tự động chuyển đến trang duyệt đơn hàng và cập nhật lại bảng danh sách đơn đặt hàng.
- 10) Nhân viên thoát khỏi trang duyệt đơn hàng.

### **Dòng sự kiện phụ:**

- 1) Thi hành chức năng duyệt đơn hàng không hợp lệ.  
Hệ thống hiển thị thông báo thi hành chức năng không hợp lệ và yêu cầu xem xét lại điều kiện thỏa mãn của quy trình duyệt đơn hàng.
- 2) Thi hành chức năng hiển thị bảng danh sách duyệt đơn hàng không hợp lệ.  
Hệ thống hiển thị bảng danh sách duyệt đơn hàng với các trường dữ liệu không phù hợp, thiếu sót hoặc không hiển thị bảng danh sách duyệt đơn hàng.

<p>3) Không thể công nhật CSDL</p> <p>Lỗi trong lúc cập nhật: Yêu cầu người dùng nhập lại thông tin, nếu vẫn bị lỗi nên liên hệ với nhà cung cấp phần mềm.</p>
<p><b>Tiền điều kiện:</b></p> <p>Actor: Nhân viên.</p> <p>Điều kiện: Nhân viên phải đăng nhập vào hệ thống</p>
<p><b>Hậu điều kiện:</b></p> <p>Hệ thống sẵn sàng thi hành chức năng duyệt đơn hàng và hiển thị danh sách duyệt đơn đặt hàng.</p>

## vii. Quản lý giỏ hàng

### Cập nhật giỏ hàng

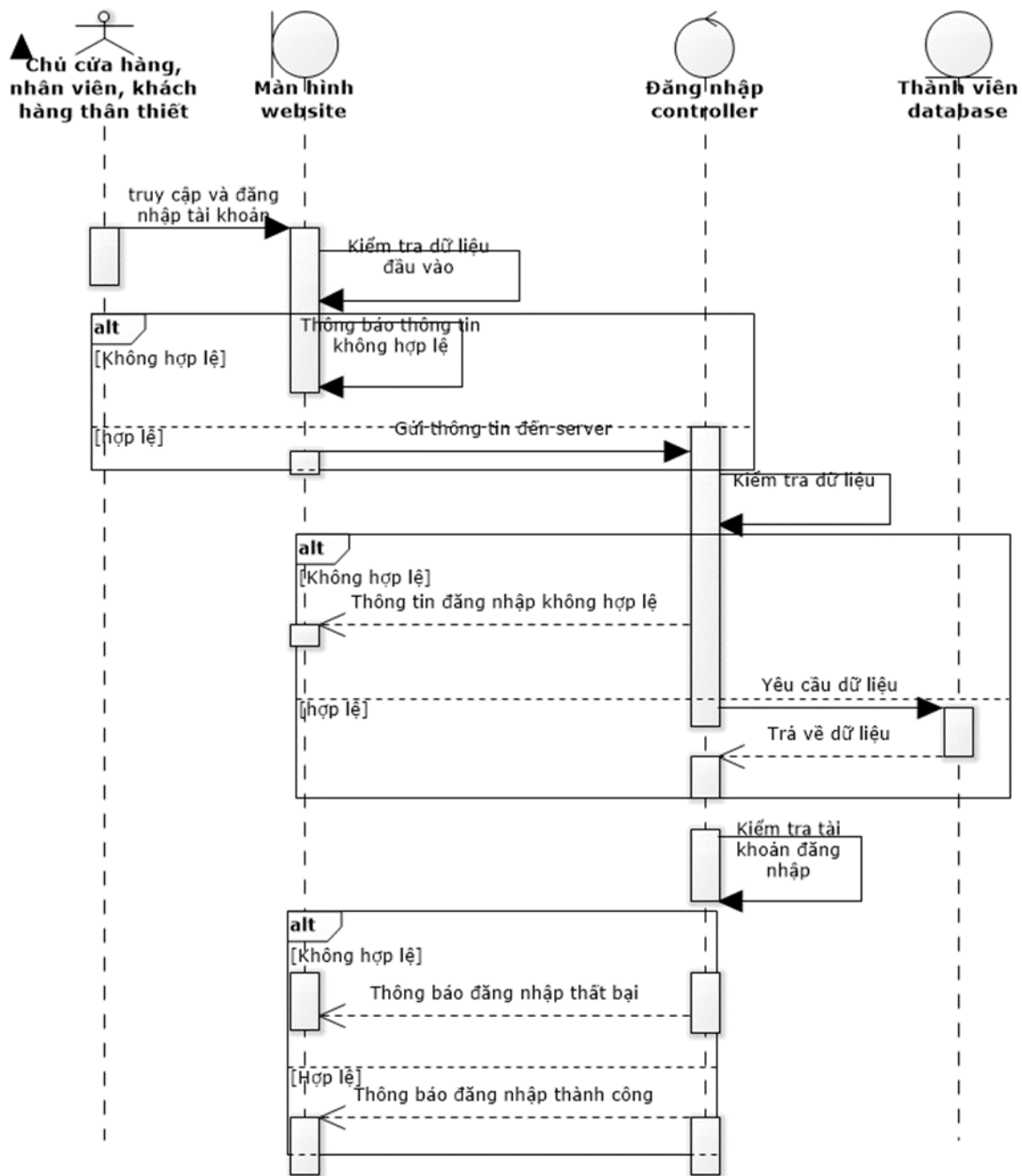
<p><b>Tóm tắt:</b> Cập nhật số lượng hàng trong giỏ hàng</p>
<p><b>Dòng sự kiện chính:</b></p> <ol style="list-style-type: none"> <li>1) Người dùng bấm vào mục giỏ hàng ở góc trên bên phải màn hình</li> <li>2) Hệ thống hiển thị giao diện giỏ hàng</li> <li>3) Người dùng bấm nút “Sửa” nằm bên phải sản phẩm muốn cập nhật</li> <li>4) Người dùng nhập số lượng sản phẩm mới vào mục số lượng và bấm nút “Cập nhật” (Dòng sự kiện phụ: Người dùng nhập số lượng không hợp lệ)</li> <li>5) Hệ thống thay đổi số lượng sản phẩm như người dùng đã nhập và thay đổi tổng tiền ở mục “Thành tiền”</li> <li>6) Quay về giao diện giỏ hàng</li> </ol>
<p><b>Dòng sự kiện phụ:</b></p> <ol style="list-style-type: none"> <li>1) Người dùng nhập số lượng không hợp lệ: Hệ thống hiển thị thông báo “Số lượng không hợp lệ” và yêu cầu nhập lại.</li> </ol>
<p><b>Tiền điều kiện:</b></p> <p>Actor: Khách hàng tiềm năng, khách hàng thành viên</p> <p>Điều kiện: Giỏ hàng đã có sản phẩm</p>
<p><b>Hậu điều kiện:</b></p> <p>Người dùng cập nhật giỏ hàng thành công</p>

### Xóa giỏ hàng

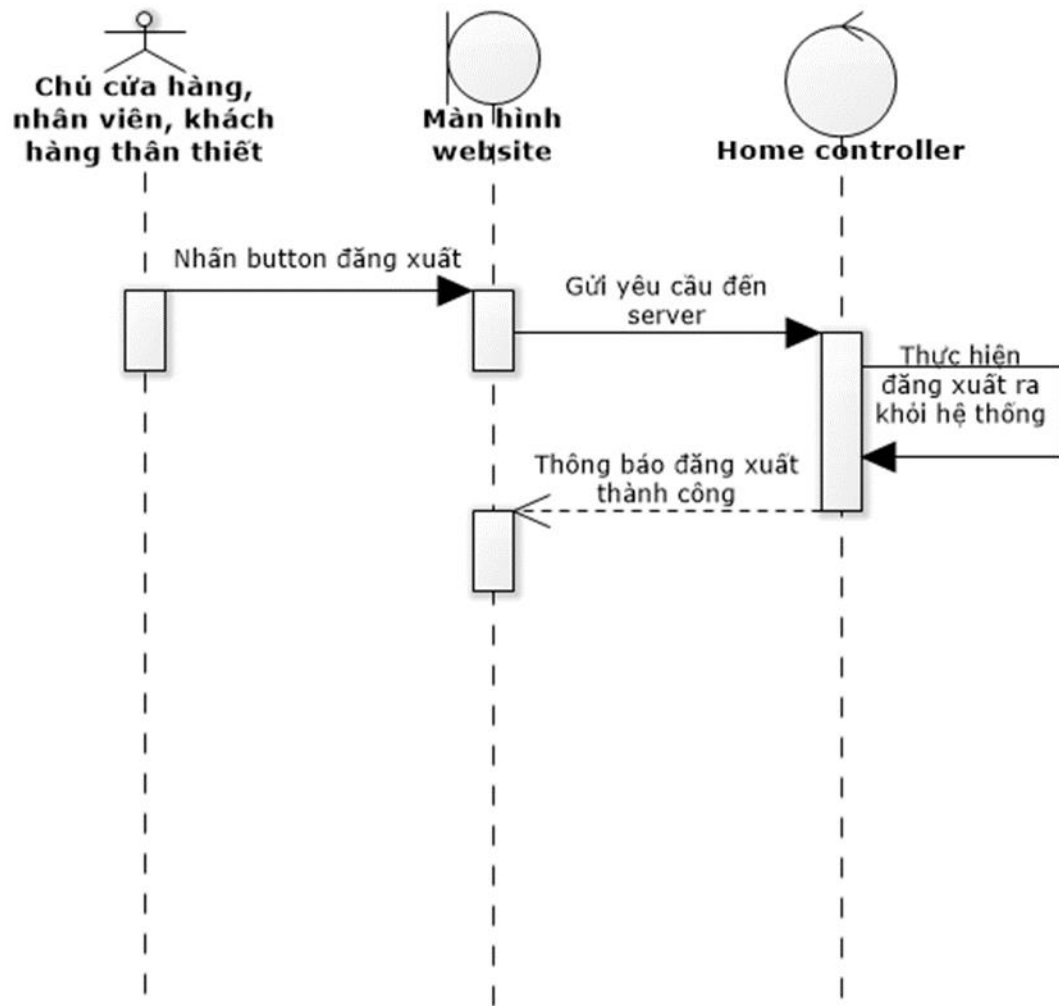
<b>Tóm tắt:</b> Chức năng xóa một sản phẩm ra khỏi giỏ hàng
<b>Dòng sự kiện chính:</b> <ol style="list-style-type: none"> <li>1) Người dùng bấm vào mục giỏ hàng ở góc trên bên phải màn hình</li> <li>2) Hệ thống hiển thị giao diện giỏ hàng</li> <li>3) Người dùng bấm nút “Xóa” nằm bên phải sản phẩm muốn cập nhật</li> <li>4) Hệ thống xóa sản phẩm khỏi giỏ hàng</li> <li>5) Quay về giao diện giỏ hàng</li> </ol>
<b>Dòng sự kiện phụ:</b>
<b>Tiền điều kiện:</b> Actor: Khách hàng tiềm năng, khách hàng thành viên Điều kiện: Giỏ hàng đã có sản phẩm
<b>Hậu điều kiện:</b> Sản phẩm bị loại bỏ khỏi giỏ hàng thành công

## 2. SEQUENCE

### a. Đăng nhập

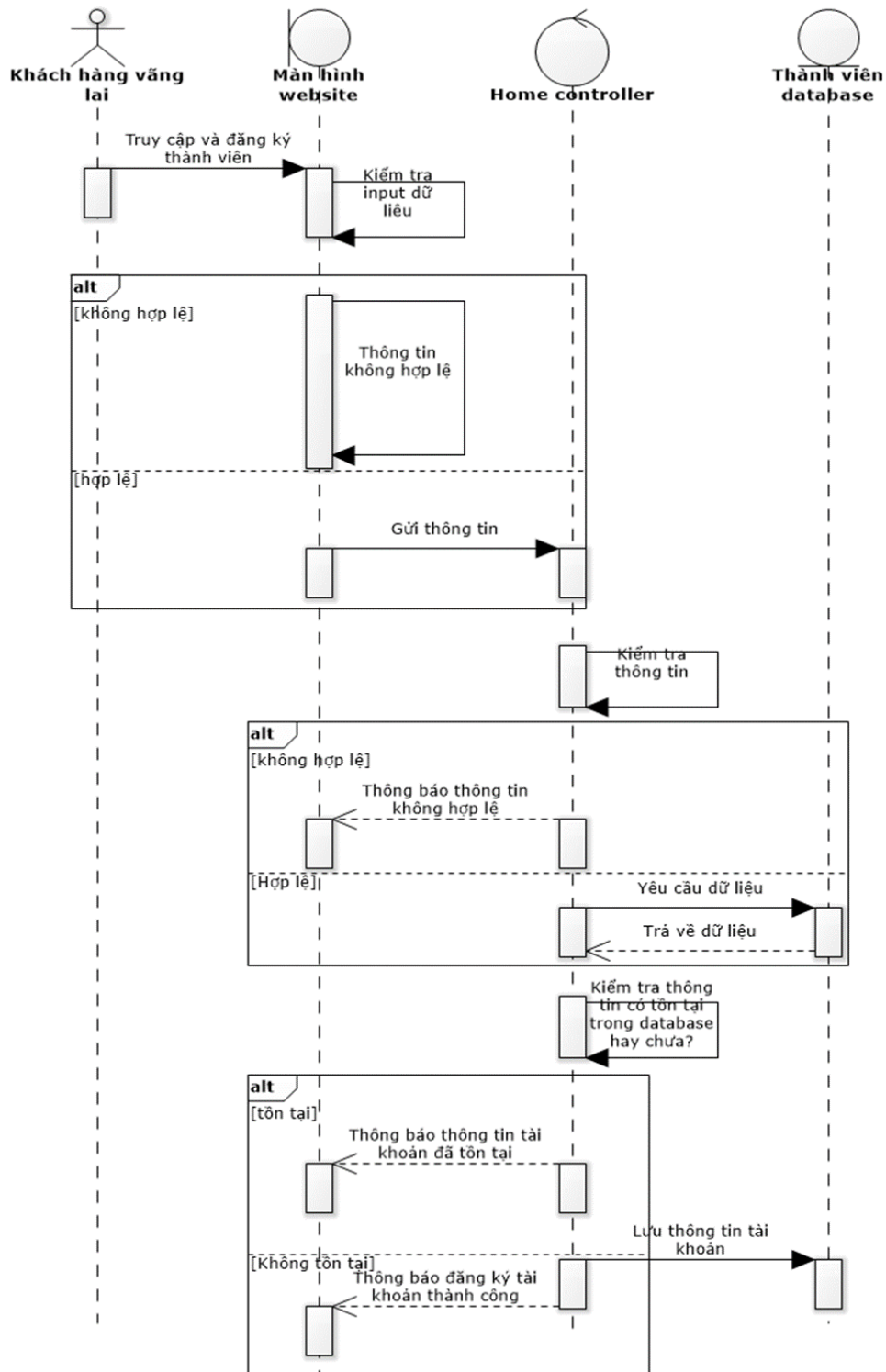


## b. Đăng xuất

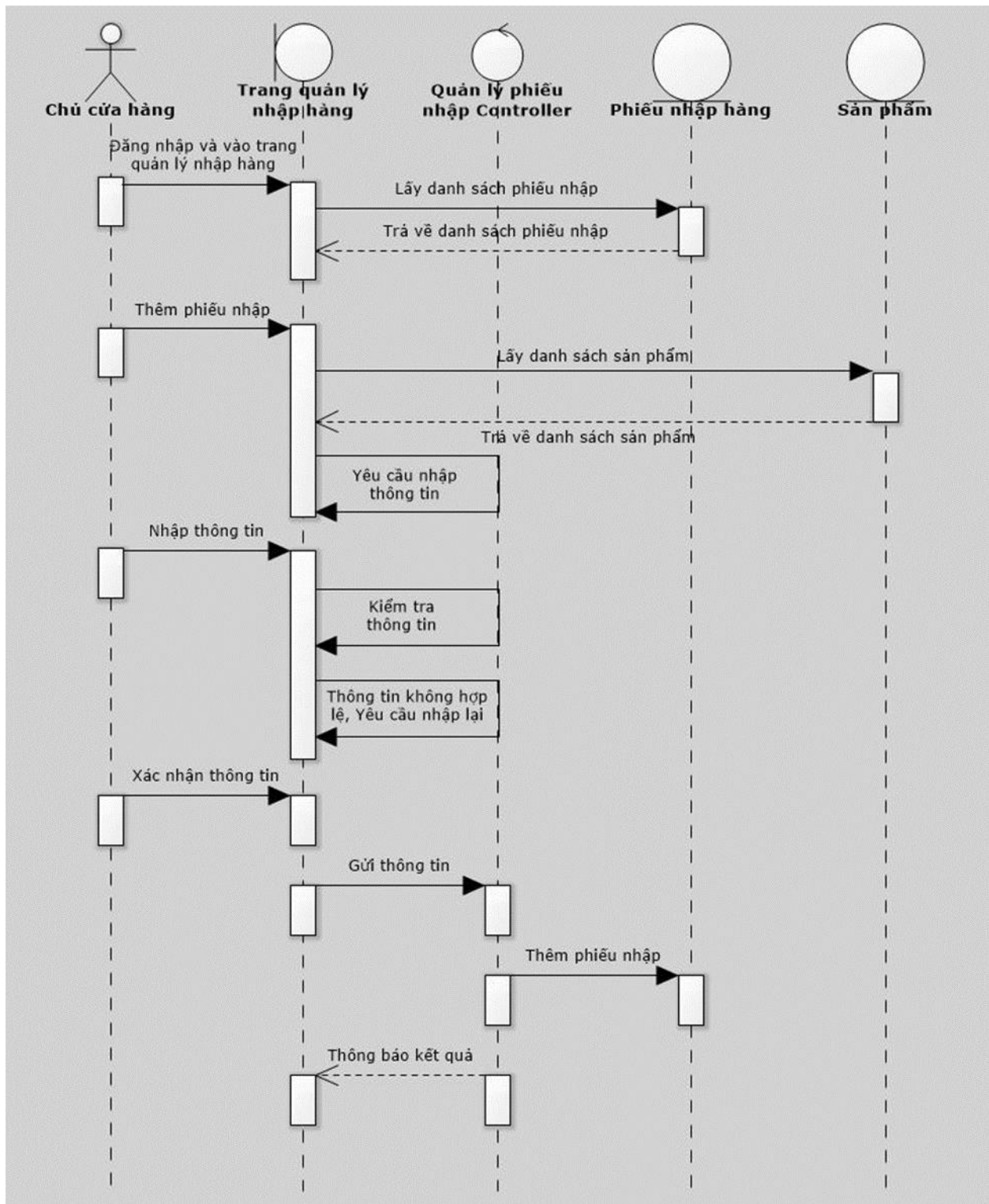




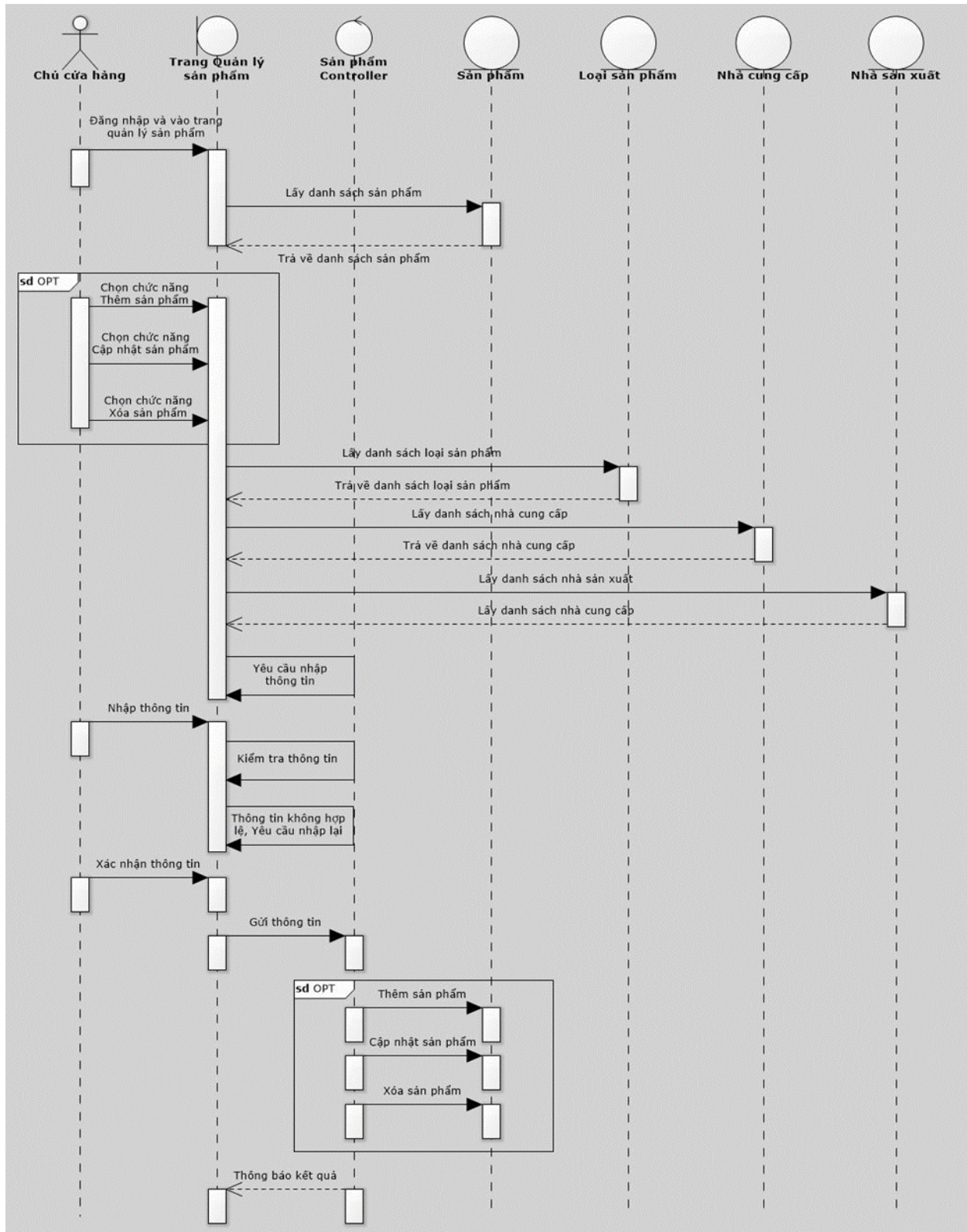
### c. Đăng ký thành viên



### d. Quản lý nhập hàng

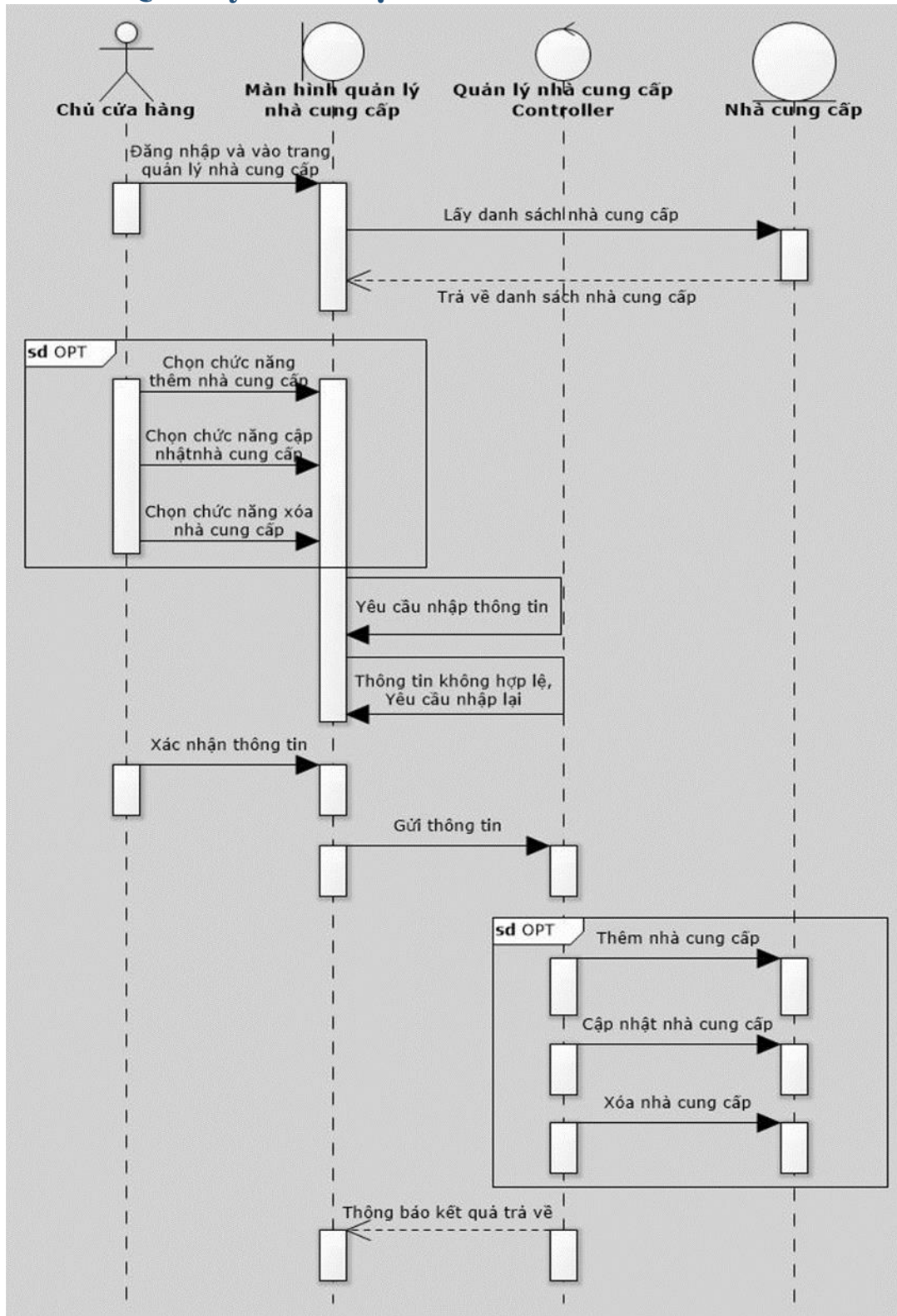


### e. Quản lý sản phẩm

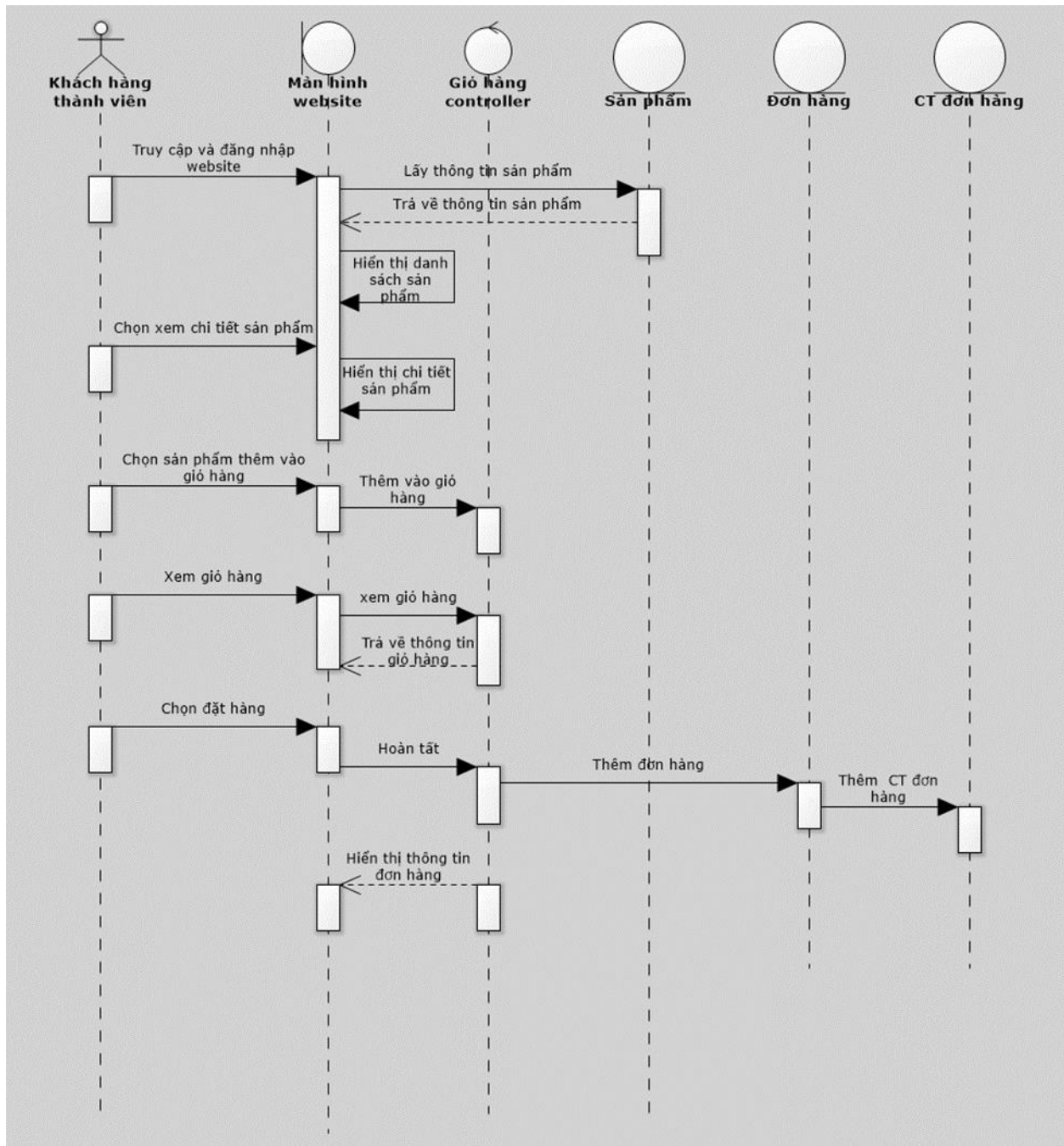




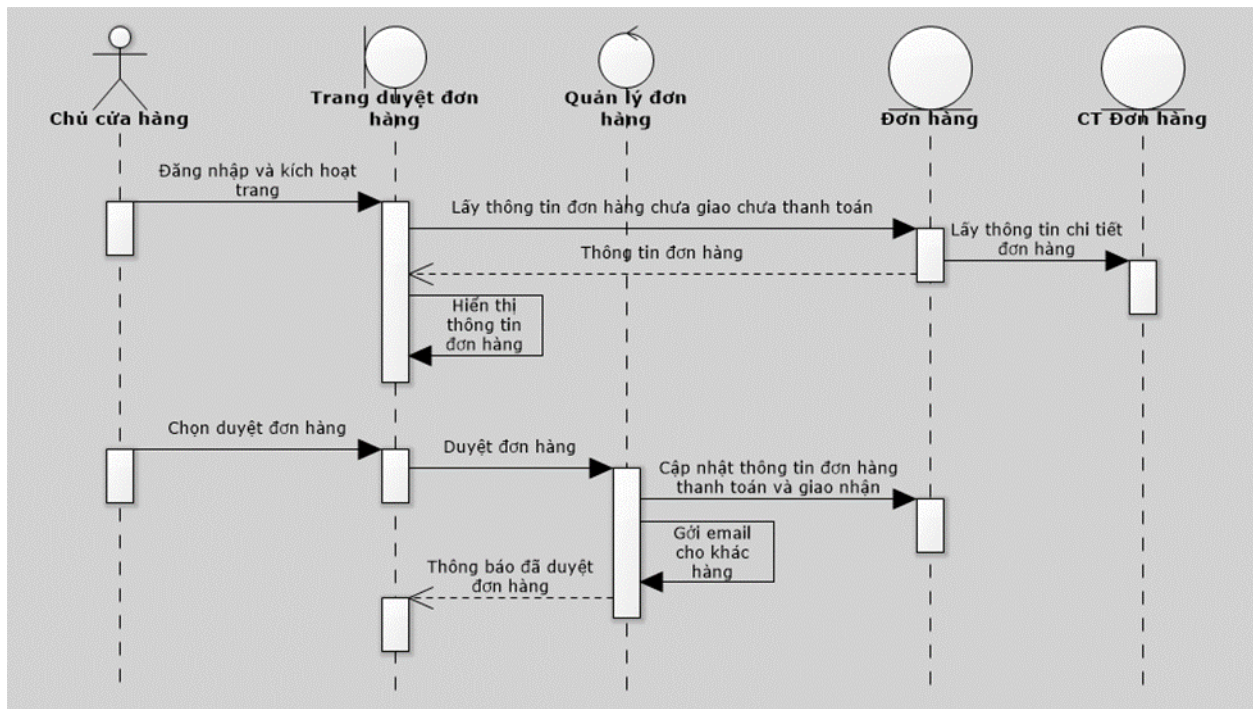
## f. Quản lý danh mục



## g. Đặt hàng



## h. Duyệt đơn hàng

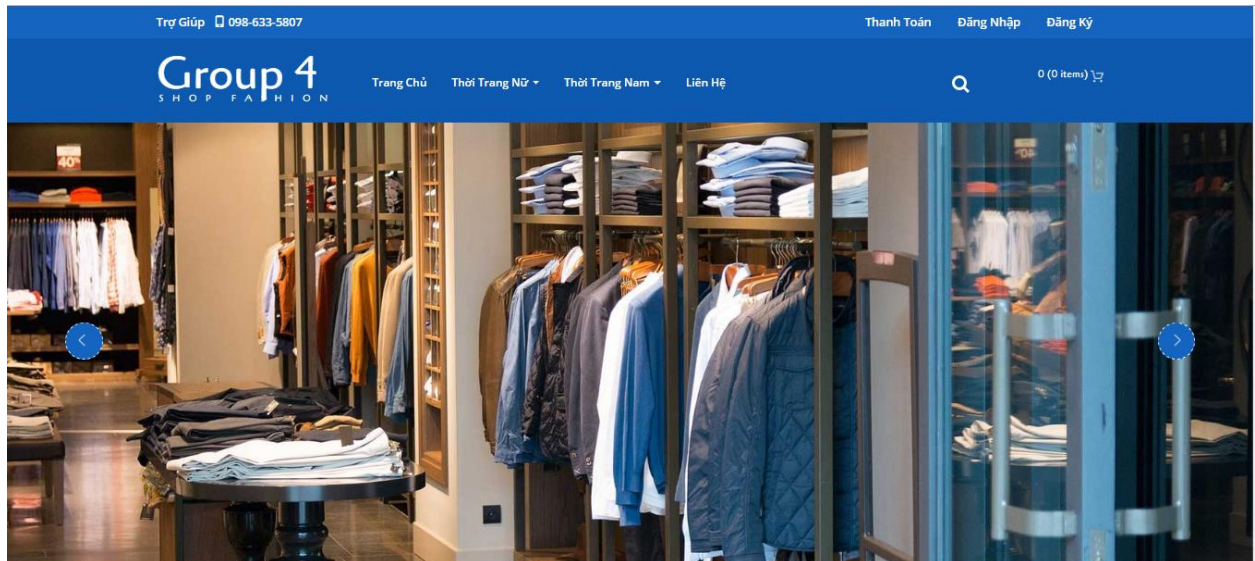




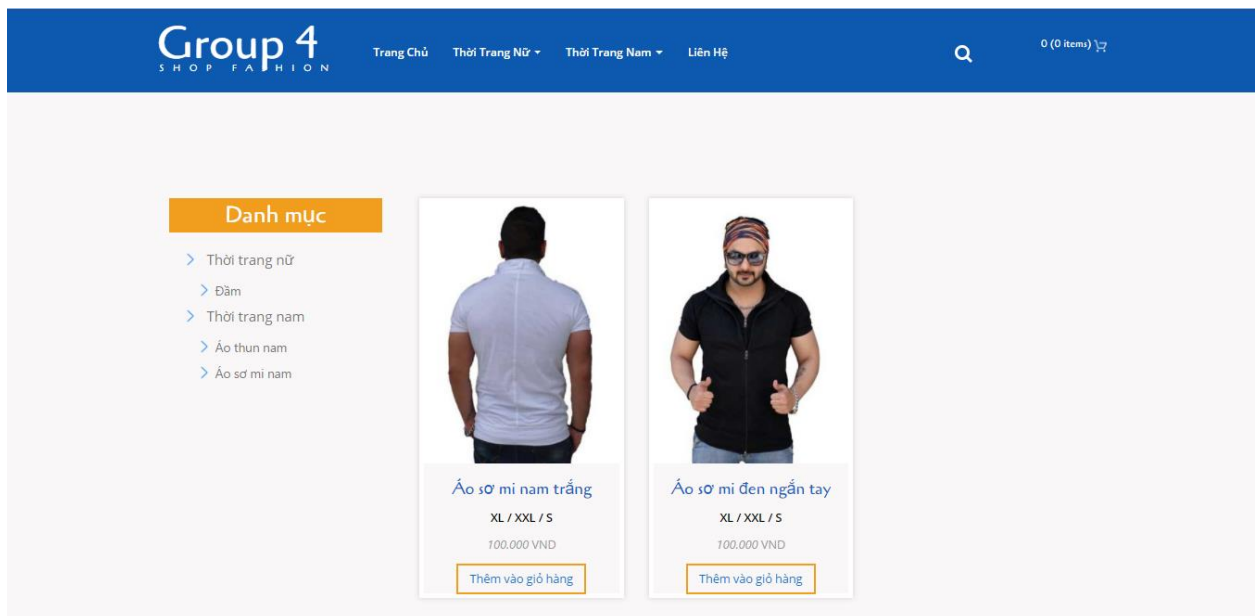
### 3. GIAO DIỆN WEBSITE

#### A. Phần giao diện bên ngoài

- Trang chủ



- Trang danh sách sản phẩm



- Trang chi tiết sản phẩm



Áo sơ mi nam trắng

100.000 VND

Mô tả ngắn : Áo sơ mi nam

XL / XXL / S

Add To Cart



- Trang đăng nhập

## Đăng nhập

✉️ thuongthanhto@gmail.com

🔒 .....

Đăng nhập

Đăng ký tài khoản mới

- Trang đăng ký

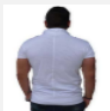


## Đăng ký



Đăng ký

### - Trang giỏ hàng

Mã SP	Tên SP	Hình ảnh	Số lượng	Đơn giá	Thành tiền	
1	Áo sơ mi nam trắng		2	100000 VNĐ	200000 VNĐ	Xóa

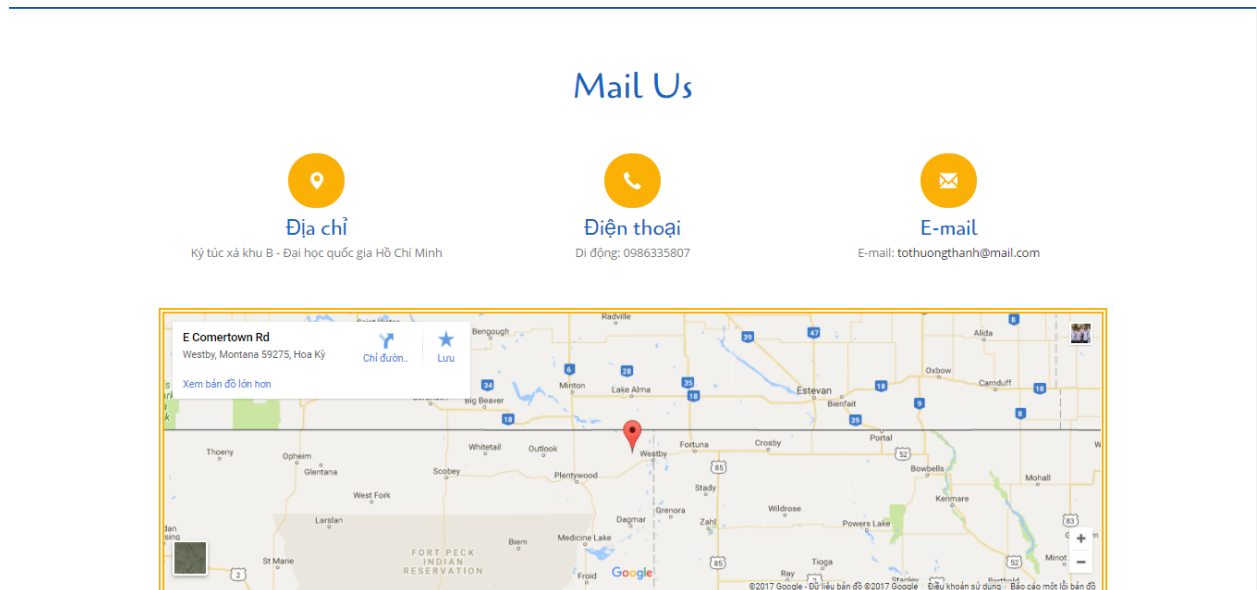
### Thông tin khách hàng

Địa chỉ:

Số điện thoại:

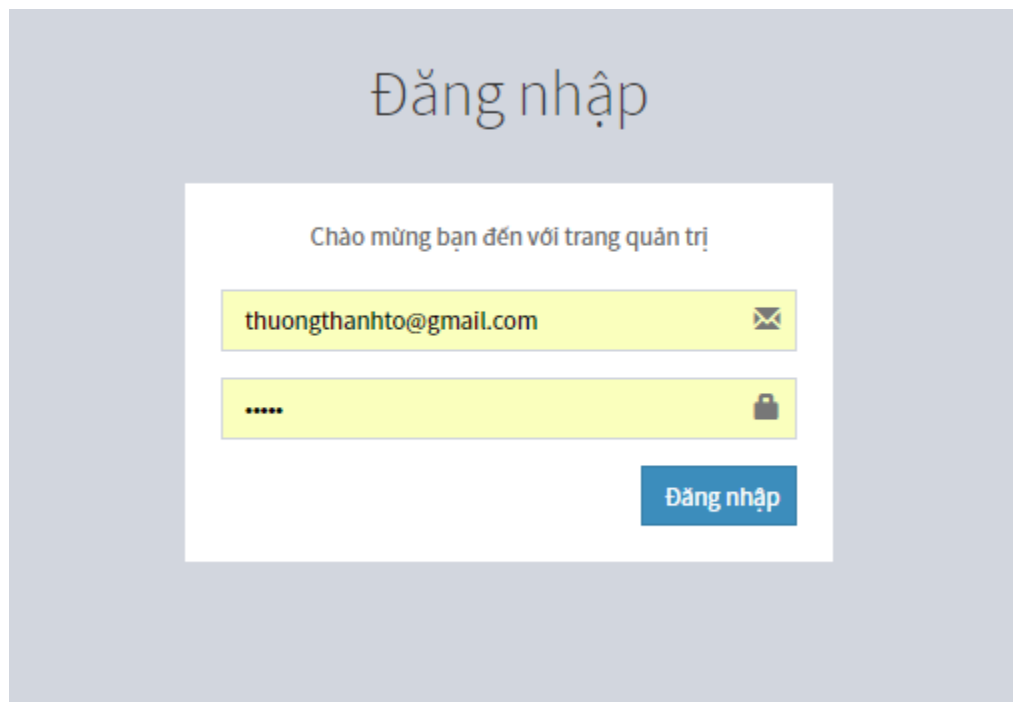
Đặt hàng

- Trang liên hệ

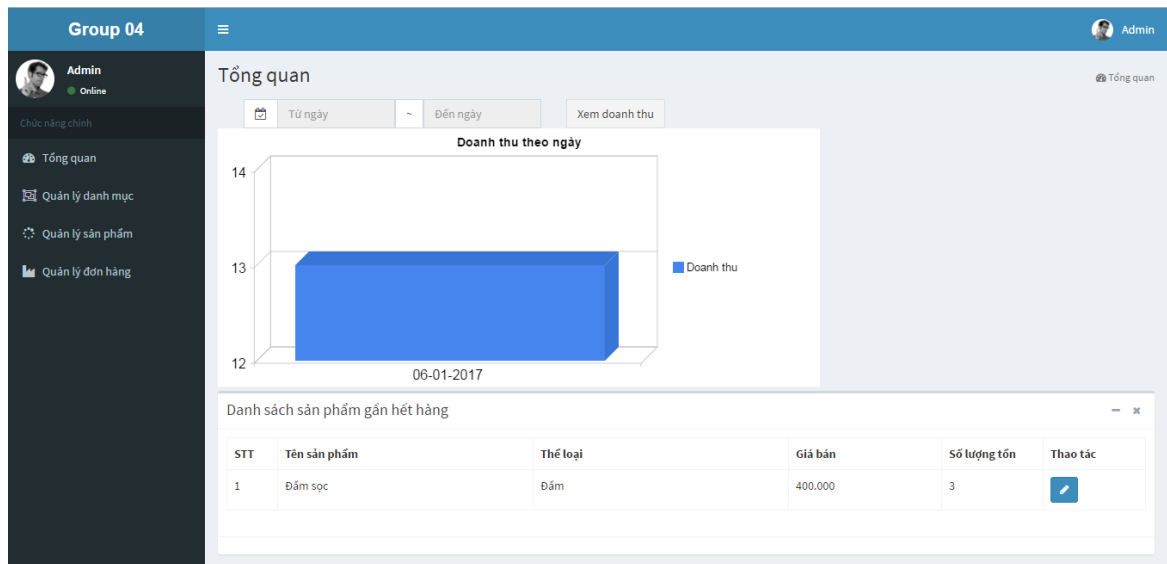


## B. Phần quản trị

- Đăng nhập



- Trang chủ admin



- Trang danh mục sản phẩm

**Group 04**

**Admin** Online

Chức năng chính

- Tổng quan
- Quản lý danh mục
- Quản lý sản phẩm
- Quản lý đơn hàng

**Danh mục sản phẩm**

Trang chủ > Danh mục sản phẩm

**Danh sách**

Từ khóa Tìm kiếm **Thêm mới**

STT	Mã danh mục	Tên danh mục	Nhóm danh mục	Thao tác
1	1	Áo thun nam	Nam	
2	2	Áo sơ mi nam	Nam	
3	3	Đầm	Nữ	

Trang: 1 - Tổng số bản ghi: 3

« 1 »

Copyright © 2014-2016 Almsaeed Studio. All rights reserved. Version 2.3.8

- Trang thêm danh mục sản phẩm

Group 04

Admin Online

Chức năng chính

Tổng quan

Quản lý danh mục

Quản lý sản phẩm

Quản lý đơn hàng

Danh mục sản phẩm

Trang chủ > Danh mục > Thêm mới

Thêm mới

Tên danh mục

Nhóm danh mục

☐ Nam ☐ Nữ

Hủy

Thêm

Copyright © 2014-2016 Almsaeed Studio. All rights reserved.

Version 2.3.8

- Trang danh sách sản phẩm

Group 04

Admin Online

Chức năng chính

Tổng quan

Quản lý danh mục

Quản lý sản phẩm

Quản lý đơn hàng

Danh mục sản phẩm

Trang chủ > Danh mục sản phẩm

Danh sách

Từ khóa

Tìm kiếm

Thêm mới

STT	Hình ảnh	Tên sản phẩm	Thể loại	Giá bán	Số lượng tồn	Thao tác
1		Đầm sọc	Đầm	400.000	3	
2		Áo sơ mi đen ngắn tay	Áo sơ mi nam	100.000	10	
3		Áo sơ mi nam trắng	Áo sơ mi nam	100.000	10	

- Trang thêm sản phẩm

**Sản phẩm**

Thông tin sản phẩm

Tên sản phẩm:

Danh mục:

Đơn giá:

Số lượng:

Hình ảnh:  Không có tệp nào được chọn

Mô tả:

- Trang danh sách hóa đơn

**Group 04**

**Quản lý hóa đơn**

Danh sách hóa đơn

Số hóa đơn	Khách hàng	Số điện thoại	Địa chỉ	Tổng tiền	Ngày giao	Thanh toán	Tình trạng	Tùy chọn
1463295350519	thuongthanhto@gmail.com	0986335807	123 - Q1 - TPHCM	13.0	2017-06-01 20:51:59.0	Chưa thanh toán	Đã hủy	<input type="button" value="v"/>
1463297481103	thuongthanhto@gmail.com	0986335807	123 - Q1 - TPHCM	7.0	2017-05-20 10:09:20.0	Chưa thanh toán	Chưa duyệt	<input type="button" value="v"/>

« 1 »

Copyright © 2014-2016 Almsaeed Studio. All rights reserved. Version 2.3.8

- Trang xem chi tiết hóa đơn

Group 04

Admin Online

Chức năng chính

Tổng quan

Quản lý danh mục

Quản lý sản phẩm

Quản lý đơn hàng

Quản lý hóa đơn

Trang chủ > Quản lý hóa đơn > Chi tiết hóa đơn

Chi tiết hóa đơn

Thông tin sản phẩm

Mã sản phẩm	Tên sản phẩm	Giá	Số lượng
1	Áo sơ mi nam trắng	2.0	1
2	Áo sơ mi đen ngắn tay	3.0	1
0	null	4.0	2

Quay lại

Thông tin hóa đơn

Số phiếu: 1463295350519  
Ngày giao: 2017-06-01 20:51:59.0  
Khách hàng: thuongthanhto@gmail.com  
Số điện thoại: 0986335807  
Địa chỉ: 123 - Q1 - TPHCM  
Tổng tiền: 13.0

Copyright © 2014-2016 Almsaeed Studio. All rights reserved.

Version 2.3.8

- Trang duyệt hóa đơn

Admin Online

Chức năng chính

Tổng quan

Quản lý danh mục

Quản lý sản phẩm

Quản lý đơn hàng

Quản lý hóa đơn

Trang chủ > Quản lý hóa đơn > Chi tiết hóa đơn

Chi tiết hóa đơn

Bạn có chắc chắn muốn duyệt hóa đơn này?

Thông tin sản phẩm

Mã sản phẩm	Tên sản phẩm sản phẩm	Giá	Số lượng
1	Áo sơ mi nam trắng	2.0	2
2	Áo sơ mi đen ngắn tay	3.0	1

Quay lại

Thông tin hóa đơn


Số phiếu: 1463297481103  
Ngày giao: 2017-05-20 10:09:20.0  
Khách hàng: thuongthanhto@gmail.com  
Số điện thoại: 0986335807  
Địa chỉ: 123 - Q1 - TPHCM  
Tổng tiền: 7.0

Duyệt

Copyright © 2014-2016 Almsaeed Studio. All rights reserved.

Version 2.3.8

- Trang hủy hóa đơn


**Admin**  
Online

Chức năng chính

- Tổng quan
- Quản lý danh mục
- Quản lý sản phẩm
- Quản lý đơn hàng

Trang chủ
>
Quản lý hóa đơn
>
Chi tiết hóa đơn

### Quản lý hóa đơn

#### Chi tiết hóa đơn

Bạn có chắc chắn muốn hủy hóa đơn này?

#### Thông tin sản phẩm

Mã sản phẩm	Tên sản phẩm	Giá	Số lượng
1	Áo sơ mi nam trắng	2.0	2
2	Áo sơ mi đen ngắn tay	3.0	1

#### Thông tin hóa đơn

Số phiếu
1463297481103

Ngày giao
2017-05-20 10:09:20.0

Khách hàng
thuongthanhto@gmail.com

Số điện thoại
0986335807

Địa chỉ
123 - Q1 - TPHCM

Tổng tiền
7.0

Quay lại
Hủy

Copyright © 2014-2016 Almsaeed Studio. All rights reserved.

Version 2.3.8

## HƯỚNG DẪN CHẠY PROJECT

- Môi trường phát triển
  - Netbean 8.2
  - MySQL workbench 6.3C
- Chạy project
  - Đầu tiên bạn mở MySQL workbench lên rồi tạo 1 database shop rồi mở file shop.sql lên rồi chạy file script này để tạo database
  - Mở project lên và chạy

Link source code: <https://github.com/thuongthanhto/J2EE.git>