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



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**--------------------**

**ĐỒ ÁN MÔN CÔNG NGHỆ J2EE**

**Đề tài: WEBSITE BÁN NHẠC MP3**

**Giảng viên hướng dẫn:**

**Ths. NGUYỄN TRÁC THỨC**

**Sinh viên thực hiện:**

**NGUYỄN VĂN NGUYỆN – 13520567**

**HUỲNH NGỌC THẮNG – 13520775**

**TRẦN MINH TUẤN - 12520486**

***Năm học 2007 - 2008***

***TPHCM, 31/05/2017***

**Mục lục**

[PHẦN MỘT: GIỚI THIỆU 4](#_Toc484122343)

[1. TỔNG QUAN VỀ J2EE 4](#_Toc484122344)

[1.1.1 JavaEE là gì? 4](#_Toc484122345)

[1.1.2 Kiến trúc của JavaEE 4](#_Toc484122346)

[1.1.3 Các Container trong JavaEE 6](#_Toc484122347)

[1.1.4 Các công nghệ trong JavaEE 6](#_Toc484122348)

[1.1.5 Nhận xét về JavaEE 7](#_Toc484122349)

[2. TỔNG QUAN VỀ PROJECT 7](#_Toc484122350)

[PHẦN HAI: CÔNG NGHỆ ÁP DỤNG 7](#_Toc484122351)

[1. TỔNG QUAN VỀ MVC 7](#_Toc484122352)

[1.1.1 Mô hình MVC là gì? 7](#_Toc484122353)

[1.1.2 Các thành phần trong MVC 7](#_Toc484122354)

[1.1.3 Sự tương tác giữa các lớp trong mô hình MVC 8](#_Toc484122355)

[1.1.4 Quy trình hoạt động mô hình MVC trong dự án Website 8](#_Toc484122356)

[1.1.5 Ưu nhược điểm của mô hình MVC? 9](#_Toc484122357)

[2. TỔNG QUAN VỀ STRUTS FRAMEWORK 9](#_Toc484122358)

[2.1 Struts Framework là gì? 9](#_Toc484122359)

[2.2 Các kiến trúc lập trình 10](#_Toc484122360)

[2.3 Các thành phần cơ bản của Struts 10](#_Toc484122361)

[2.4 Chu kì sống của Struts 11](#_Toc484122362)

[3. JSP 11](#_Toc484122363)

[3.1 Giới thiệu JSP 11](#_Toc484122364)

[3.2 Thành phần của JSP 12](#_Toc484122365)

[3.3 Kiến trúc của JSP trong ứng dụng Web 13](#_Toc484122366)

[3.4 Chu trình sống của JSP 14](#_Toc484122367)

[3.5 Ưu nhược điểm của JSP 15](#_Toc484122368)

[4. SERVLET 15](#_Toc484122369)

[4.1 Servlet là gì? 15](#_Toc484122370)

[4.2 Kiến trúc của Servlet 15](#_Toc484122371)

[4.3 Hoạt động của Servlet 16](#_Toc484122372)

[4.4 Servlet Package 16](#_Toc484122373)

[4.5 Ưu nhược điểm của Servlet 16](#_Toc484122374)

[5. HTML 17](#_Toc484122375)

[5.1 HTML là gì? 17](#_Toc484122376)

[5.2 HTML được xử lý như thế nào? 17](#_Toc484122377)

[5.3 Cấu trúc một đoạn HTML  <p>Đây là một đoạn văn bản trong HTML.</p> 17](#_Toc484122378)

[5.4 HTML đóng vai trì gì trong website 17](#_Toc484122379)

[6. CSS 18](#_Toc484122380)

[6.1 CSS là gì? 18](#_Toc484122381)

[6.2 Cấu trúc một đoạn CSS 19](#_Toc484122382)

[6.3 Cách chèn CSS vào trang HTML: 19](#_Toc484122383)

[7. JAVASCRIPT 21](#_Toc484122384)

[7.1 Javascript là gì? 21](#_Toc484122385)

[7.2 Đặc điểm của Javascript 21](#_Toc484122386)

[7.3 Cách nhúng Javascript 21](#_Toc484122387)

[7.4 Ưu điểm của Javascript 21](#_Toc484122388)

[8. AJAX 22](#_Toc484122389)

[8.1 Ajax là gì? 22](#_Toc484122390)

[8.2 Hoạt động của Ajax 22](#_Toc484122391)

[8.3 Sử dụng Ajax khi nào và ở đâu? 23](#_Toc484122392)

[8.4 Trình duyệt hỗ trợ Ajax 23](#_Toc484122393)

[8.5 Ứng dụng Ajax phổ biến 23](#_Toc484122394)

[8.6 Để sử dụng tốt Ajax? 23](#_Toc484122395)

[8.7 Ưu nhược điểm của Ajax 23](#_Toc484122396)

[PHẦN BA: THIẾT KẾ VÀ CÀI ĐẶT 24](#_Toc484122397)

[1. KIẾN TRÚC HỆ THỐNG 24](#_Toc484122398)

[2. SƠ ĐỒ USE – CASE 25](#_Toc484122399)

[3. SƠ ĐỒ SEQUENCE 25](#_Toc484122400)

[3.1 Chức năng Upload nhạc 25](#_Toc484122401)

[3.2 Chức năng Comment – Share 25](#_Toc484122402)

[3.3 Chức năng tìm kiếm nhạc 26](#_Toc484122403)

[4. GIAO DIỆN ỨNG DỤNG 26](#_Toc484122404)

[NHÂN XÉT VÀ KẾT LUẬN 26](#_Toc484122405)

[HƯỚNG PHÁT TRIỂN 27](#_Toc484122406)

[PHỤ LỤC 27](#_Toc484122407)

[TÀI LIỆU THAM KHẢO 27](#_Toc484122408)

# **PHẦN MỘT: GIỚI THIỆU**

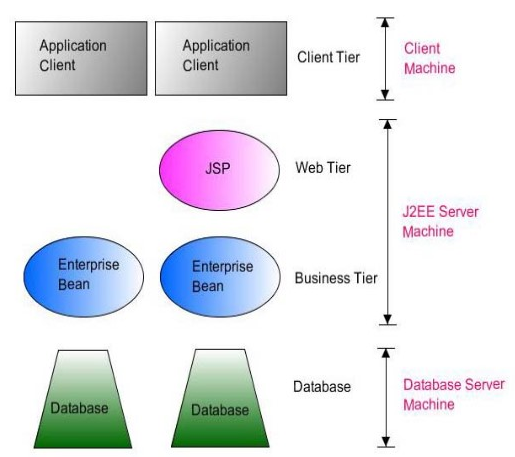
## **TỔNG QUAN VỀ J2EE**

### JavaEE là gì?

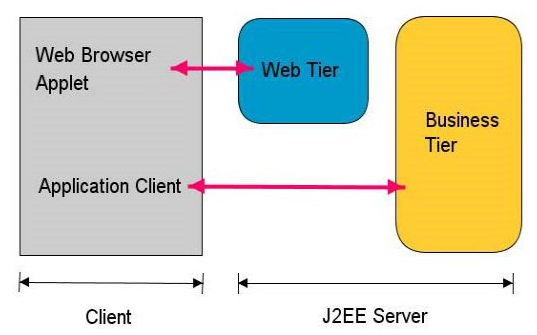
Java 2 Platform Enterprise Edition viết tắt là J2EE, định nghĩa một chuẩn để phát triển những ứng dụngthương mại đa tầng (multitier enterprise applications). J2EE platform chứa đựng những tính năng tốt của Java 2 Platform như “viết một lần, chạy ở mọi nơi”, JDBC (Java Database Connectivity) truy cập 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.

### Kiến trúc của JavaEE

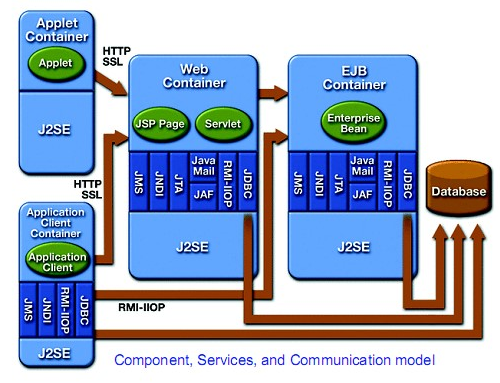
* J2EE nền tảng sử dụng một mô hình ứng dụng phân tán đa tầng.



* 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



### Các Container trong JavaEE

* **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 JavaEE

* **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.

### Nhận xét về JavaEE

## **TỔNG QUAN VỀ PROJECT**

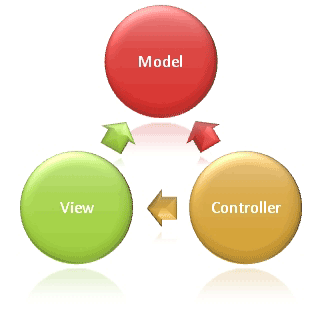
# **PHẦN HAI: CÔNG NGHỆ ÁP DỤNG**

## **TỔNG QUAN VỀ MVC**

### Mô hình MVC là gì?

Mô hình MVC là một chuẩn mô hình và đóng vai trò quan trọng trong quá trình xây dựng – phát triển – vận hành và bảo trì một hệ thống hay một ứng dụng – phần mềm. Nó tạo ra một mô hình 3 lớp Model – View – Controller tách biệt và tương tác nhau, giúp các chuyên gia có thể dễ dàng dựa vào mô hình để trao đổi và xử lý những nghiệp vụ một cách nhanh chóng. Đây là một mô hình đã xuất hiện từ những năm 70 của thế kỷ 20 tại phòng thí nghiệm Xerox PARC ở Palo Alto, nó không phụ thuộc vào môi trường, nền tảng xây dựng hay ngôn ngữ phát triển. Chúng ta có thể áp dụng mô hình MVC vào các dự án trong môi trường Windows, Linux… và sử dụng bất kỳ ngôn ngữ nào như PHP, ASP, JSP…

### Các thành phần trong MVC



Mô hình MVC được chia làm 3 lớp xử lý gồm Model – View – Controller :

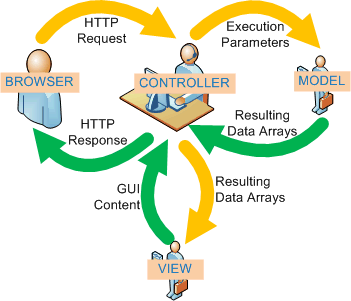
* **Model** : là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql… ); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu…
* **View** : là nới chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh… nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.
* **Controller** : là nới tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View

### Sự tương tác giữa các lớp trong mô hình MVC

* **Controller – View** sẽ lấy những hình ảnh, nút bấm…hoặc hiển thị dữ liệu được trả ra từ Controller để người dùng có thể quan sát và thao tác. Trong sự tương tác này cũng có thể không có dữ liệu được lấy từ Model và khi đó nó chỉ chịu trách nhiệm hiển thị đơn thuần như hình ảnh, nút bấm…
* **Controller – Model** là luồng xử lý khi controller tiếp nhận yêu cầu và các tham số đầu vào từ người dùng, controller sẽ sử dụng các lớp/hàm trong Model cần thiết để lấy ra những dữ liệu chính xác.
* **View – Model** có thể tương tác với nhau mà không qua Controller, nó chỉ đảm nhận hiển thị dữ liệu chứ không phải qua bất kỳ xử lý nghiệp vụ logics nào. Nó giống như các vùng dữ liệu hiển thị tĩnh trên các website như block slidebar…

Mô hình MVC có thể áp dụng cho nhiều loại dự án khác nhau : dự án website, dự án ứng dụng – phần mềm,… nhưng trong phạm vi bài viết chúng ta sẽ chỉ đi tìm hiểu mô hình này được áp dụng như nào trong các dự án xây dựng và phát triển website

### Quy trình hoạt động mô hình MVC trong dự án Website



Mô hình trên là thể hiện mô hình MVC trong các dự án website và sẽ hoạt động theo quy trình sau:

* Người dùng sử dụng một BROWSER trình duyệt web bất kỳ (Firefox, Chrome, IE,…) để có thể gửi những yêu cầu (HTTP Request) có thể kèm theo những dữ liệu nhập tới những CONTROLLER xử lý tương ứng. Việc xác định Controllerr xử lý sẽ dựa vào một bộ Routing điều hướng.
* Khi CONTROLLER nhận được yêu cầu gửi tới, nó sẽ chịu trách nhiệm kiểm tra yêu cầu đó có cần dữ liệu từ MODEL hay không? Nếu có, nó sẽ sử dụng các class/function cần thiết trong MODEL và nó sẽ trả ra kết quả( Resulting Arrays), khi đó CONTROLLER sẽ xử lý giá trị đó và trả ra VIEW để hiển thị. CONTROLLER sẽ xác định các VIEW tương ứng để hiển thị đúng với yêu cầu.
* Khi nhận được dữ liệu từ CONTROLLER, VIEW sẽ chịu trách nhiệm xây dựng các thành phẩn hiển thị như hình ảnh, thông tin dữ liệu… và trả về GUI Content để CONTROLLER đưa ra kết quả lên màn hình BROWSER.
* BROWSER sẽ nhận giá trị trả về( HTTP Response) và sẽ hiển thị với người dùng. Kết thúc một quy trình hoạt động.

### Ưu nhược điểm của mô hình MVC?

#### Ưu điểm

* Các dự án có thể áp dụng ngay mô hình MVC mà không phụ thuộc môi trường, nền tảng xây dựng hay ngôn ngữ lập trình phát triển;
* Quy hoạch các class/ function vào các thành phần riêng biệt Controller – Model – View, khi đó sẽ dễ dàng xây dựng – phát triển – quản lý – vận hành và bảo trì một dự án, tạo sự rõ ràng, trong sáng trong quá trình phát triển dự án, kiểm soát được các luồng xử lý và tạo ra các thành phần xử lý nghiệp vụ chuyên biệt hóa.
* Tạo thành mô hình chuẩn cho nhiều dự án, các chuyên gia sẽ tiếp cận – tìm hiểu những dự án đó một cách nhanh chóng và hiệu quả . Nếu bạn nắm rõ mô hình MVC của một dự án nào đó, thì khi tiếp cận với một dự án khác mà bạn chưa từng biết hoặc tiếp xúc, nhưng nó lại được xây dựng với mô hình MVC thì sẽ không khó khăn gì mà cực kỳ dễ dàng. Học một nhưng có thể hiểu và sử dụng được mười.
* Giúp các chuyên gia lập trình, nhà quản lý, nhà đầu tư, PM… có thể hiểu được dự án hoạt động ra sao hoặc giúp các lập trình viên dễ dàng quản lý – phát triển dự án. Nó không phải ngôn ngữ, nhưng khi họ cùng nhìn vào nó thì sẽ tự hiểu nó là gì, khi đó họ có thể trao đổi các yêu cầu và bàn bạc công việc.
* Đây là một mô hình chuẩn, nó tối ưu nhất hiện nay so với nhiều mô hình khác và được sử dụng trong nhiều dự án và nhiều lĩnh vực, đặc biệt trong công nghệ sản xuất ứng dụng – phần mềm. Các lập trình viên sử dụng mô hình chuẩn MVC để có thể dễ dàng phân phối và chuyển giao công nghệ.
* Đây là mô hình đơn giản, xử lý những nghiệp vụ đơn giản, và dễ dàng triển khai với các dự án nhỏ.

#### Nhược điểm

Yêu cầu về chuyên môn khá cao, có kiến thức vững về các mô hình chuẩn;Khó triển khai với những dự án yêu cầu phúc tạp hơn. Hiện nay đang có một khái niệm mô hình mới đó là HMVC đang dần thay thế cho MVC

## **TỔNG QUAN VỀ STRUTS FRAMEWORK**

### 2.1 Struts Framework là gì?

Struts là một framework phục vụ việc phát triển các ứng dụng Web trên Java. Sử dụng mẫu thiết kế Model-View-Controller (MVC), Struts giải quyết rất nhiều các vấn đề liên quan đến các ứng dụng Web hướng business đòi hỏi hiệu năng cao sử dụng Java servlet và JSP. Struts cơ bản định hình lại cách các Web programmer nghĩ về và cấu trúc một ứng dụng Web.

Structs được phát triển bởi Craig McClanahan và được bão trợ bởi Apache, nhóm Jakarta

### 2.2 Các kiến trúc lập trình

#### 2.2.1. Kiến trúc truyền thống (Còn gọi là mô hình 1 - Model 1).

Kiến trúc Model 1 hết sức đơn giản. Tóm lại là 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, kiến trúc này không phù hợp cho việc phát triển các ứng dụng quy mô lớn vì chắc chắn một 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. Ví dụ, ngoài giao diện HTML, bạn có thể muốn thêm một giao diện WML cho truy nhập wireless. Trong trường hợp này, việc sử dụng Model 1 yêu cầu sự lặp lại không cần thiết của business logic trong các trang sử dụng HTML và WML.

#### 2.2.2. Kiến trúc MVC (Còn gọi là mô hình 2 - Model 2)

Model 2, hay cách gọi thông dụng ngày nay, Model-View-Controller (MVC), giải quyết nhiều vấn đề của Model 1 với việc cung cấp một 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 và 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 một cách hòan hảo.

Có một điều khá lý thú ở đây là tôi đã từng cho rằng việc trong một ứng dụng Web có thể có nhiều Controller thì ứng dụng đó cũng sử dụng MVC. Có nghĩa là cứ một view component ta sẽ có một Controller cho nó. Tuy nhiên điều này là hòan tòan sai với mẫu thiết kế MVC.

### 2.3 Các thành phần cơ bản của Struts

#### 2.3.1. Framework cơ sở

Framework cơ sở cung cấp các tính năng MVC cốt lõi. Nền tảng của framework cơ sở là Controller servlet: ActionServlet. Phần còn lại của framework cơ sở bao gồm các lớp cơ sở mà ứng dụng của bạn sẽ extend và các class tiện ích. Nổi bật nhát trong các lớp cơ sở là lớp Action và lớp ActionForm. Lớp Action được sử dụng bởi ActionServlet để sử lý các request xác định. Lớp ActionForm được sử dụng để capture dữ liệu từ các HTML form và được sử dụng để chuyển dữ liệu trở lại View để generate response.

#### 2.3.2. Thư viện thẻ JSP

Struts đưa ra các thư viện thẻ để hỗ trợ việc lập trình View logic trong JSP. Các thư viện thẻ JSP cho phép các JSP author sử dụng các thẻ giống HTML.

* HTML: Sử dụng để generate các THML form tương tac với Struts API
* Bean: Sử dụng để làm việc với các Java bean object trong JSP
* Logic: Sử dụng để implement các logic điều kiện đơn giản trong JSP
* Nested: Sử dụng để cho phép các mức nested tùy ý

#### 2.3.3. Tiles plugin

Tiles là một JSP template framework giúp thuận tiện cho việc tái sử dụng mã HTML.

#### 2.3.4. Validator plugin

Validator là một framework với khả năng thực hiện xác nhận tính hợp lệ của dữ liệu ở cả phía server và client.

### 2.4 Chu kì sống của Struts

Trước khi đi vào chi tiết các lớp, nghía qua một chút về tiến trình thực hiện của một ứng dụng Web sử dụng Struts cũng có lẽ không thừa.

* Trình duyệt tạo một request tới ứng dụng Struts sẽ được xử lý bới ActionServlet (Controller).
* ActionServlet (Controller) populate subclass của ActionForm (View), do ta viết, với dữ liệu form HTML và invoke method validate() của nó.
* ActionServlet (Controller) execute subclass của Action do ta viết (Controller).
* Subclass của Action (Controller) đó sẽ tương tác với model component và chuẩn bị dữ liệu cho để hiển thị.
* Sau đó subclass của Action (Controller) đó sẽ chuyển điều khiển tới JSP (View).
* JSP (View) sử dụng dữ liệu nhận được để generate response về cho trình duyệt.

## **JSP**

### 3.1 Giới thiệu JSP

* JSP (viết tắt của tiếng Anh JavaServer Pages) còn được biết đến với một cái tên khác là Java Scripting Preprocessor - tạm dịch là "Bộ tiền xử lý văn lệnh Java" - là một công nghệ Java cho phép các nhà phát triển tạo nội dung HTML, XML hay một số định dạng khác của trang web một cách năng động, trong khi hồi âm yêu cầu của trình khách. Công nghệ này cho phép người ta nhúng mã Java và một số hành động xử lý đã được định trước (pre-defined actions) vào trong nội dung tĩnh của trang.
* JSP là cách đơn giản hóa hơn cho Servlet. Nếu như đối với Servlet phải viết mã Java và biên dịch bằng tay trước khi sử dụng với trình chủ Web server thì JSP không cần điều này. JSP viết mã Java tương tự Servlet nhưng cho phép trộn lẫn Java với các thẻ HTML để tạo ra Servlet xử lý các yêu cầu mà trình khác gửi đến trang.
* JSP đơn giản và dễ sử dụng hơn Servlet mặc dù sau khi biên dịch và thực thi chúng chỉ là một.

### 3.2 Thành phần của JSP

#### 3.2.1 Thẻ root

Thẻ này sẽ chứa các thuộc tính, thông tin của trang jsp.

#### 3.2.2 Comment

Cũng như trang html, trong jsp cũng cho phép chúng ta có thể comment. Để comment chúng ta dùng kí hiệu này:

**<! your comment -->**

#### 3.2.3 Declaration

Chúng ta có thể khai báo biến hoặc phương thức của java ngay trong trang jsp như một trang ngôn ngữ kịch bản. Nhưng nếu như khai báo quá nhiều trong trang thì sẽ bị nhầm lẫn giữa code jsp và code java. Cú pháp khai báo là

**<%! your code %>**

#### 3.2.4 Expression

Thẻ biểu thức JSP được sử dụng để đánh giá một biểu thức và định hướng các output đến một trình duyệt web phù hợp.

Cú pháp khai báo là: **<%= your code %>**

#### 3.2.5 Scriptlet Tag

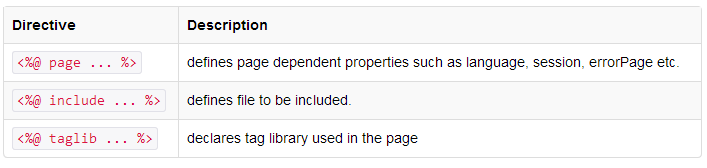
Scriptlet Tag cho phép bạn viết mã java trong trang JSP. Cú pháp của Scriptlet Tag như sau :

**<% <i> mã java </ i>%>**

#### 3.2.6 Derective Element

Một JSP directive cung cấp thông tin đặc biệt cũng như các phương tiện của trang jsp.

Directive Elements gồm có 3 thẻ như hình dưới:

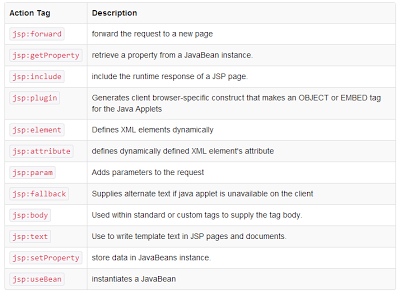


#### 3.2.7 Standard Tag (Action Element)

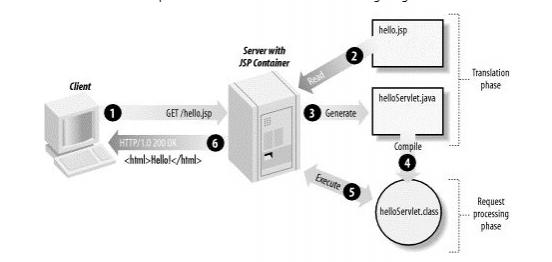
- JSP cung cấp Standard Tag(Action Element) để sử dụng trong các trang JSP của bạn. Các thẻ được sử dụng để loại bỏ hoặc loại bỏ mã scriptlet từ trang JSP của bạn bởi vì scriplet mã được về mặt kỹ thuật không được khuyến cáo hiện nay.

- Standard tags bắt đầu với tiền tố **jsp <code>: </ code>.** Có rất nhiều JSP Standard Action tag được sử dụng để thực hiện một số nhiệm vụ cụ thể.

- Sau đây là một số JSP Standard Action Tag thường dùng:



### 3.3 Kiến trúc của 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

### 3.4 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:

#### 3.4.1 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 dich 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.

#### 3.4.2 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.

#### 3.4.3 Khởi tạo

Khi nạp mã trang thành công, Web server sẽ gọi đến phương thức khỏi 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.

#### 3.4.4 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.

#### 3.4.5 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.

### 3.5 Ưu nhược điểm của JSP

#### 3.5.1 Ưu điểm

* Cho phép chúng ta thiết kế giao diện web dễ dàng hơn.
* Jsp có thể cho phép chúng ta tạo nên những trang web động.
* Có thể viết một nơi và chạy bất cứ nơi nào.

#### 3.5.2 Nhược điểm

* Nhìn chung thì việc thiết kế giao diện bằng JSP vẫn còn nhiều khó khăn.
* Tiêu tốn dung lượng lưu trữ phía server gấp đôi.
* Lần đầu tiên truy cập vào trang JSP sẽ mất nhiều thời gian chờ.

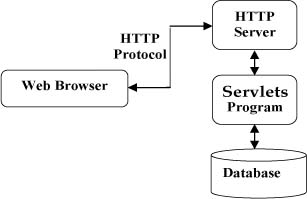
## **SERVLET**

### 4.1 Servlet là gì?

Java Servlets là một chương trình, nó chạy trên một Web Server hoặc Application server và hoạt động như tầng trung gian giữa request đến từ web browser hoặc là các Http client khác và database hoặc ứng dụng trên HTTP server

Sử dụng Servlet, bạn có thể thu thập dữ liệu đầu vào của người dùng thông qua page form, hiển thị các bạn ghi từ database hoặc nguồn khác, và tạo các trang web động

### 4.2 Kiến trúc của Servlet



### 4.3 Hoạt động của Servlet

* Đọc dữ liệu gửi bởi client (trình duyệt). Bao gồm một HTML form trên trang web hoặc cũng có thể là từ một applet hoặc một custom HTTP client khác
* Đọc các dữ liệu HTTP được gửi lên từ client (browser). Điều này bao gồm cả cookies, user-agent, media types và compression schemes, v.v
* Xử lý dữ liệu và tạo ra các kết quả. Quá trình này có thể yêu cầu kết nối với một cơ sở dữ liệu, thực hiện một RMI hoặc gọi CORBA, gọi một dịch vụ Web, hoặc tính toán phản ứng trực tiếp
* Gửi dữ liệu rõ ràng (ví dụ, tài liệu) cho khách hàng (trình duyệt). Tài liệu này có thể được gửi trong một loạt các định dạng, bao gồm văn bản (HTML hay XML), nhị phân (ảnh GIF), Excel, vv
* Gửi phản hồi HTTP ngầm cho client (browser). Điều này bao gồm nói với các browser hoặc các client khác về loại tài liệu đang được trả lại (ví dụ, HTML), thiết lập các tập tin cookie và các thông số bộ nhớ đệm, và các nhiệm vụ khác tương tự như vậy.

### 4.4 Servlet Package

* Java servlets là các lớp Java chạy bởi một máy chủ web mà có một thông dịch viên hỗ trợ Java Servlet specification.
* Servlet có thể được tạo ra bằng cách sử dụng javax.servlet và javax.servlet.http packages, đó là một phần tiêu chuẩn của JavaEE , một phiên bản mở rộng của thư viện Java hỗ trợ các dự án phát triển quy mô lớn.
* Các lớp này thực hiện các đặc tả kĩ thuật Java Servlet và JSP.Java servlet đã được tạo ra và biên dịch giống như bất kỳ lớp Java khác. Sau khi bạn cài đặt các gói servlet và thêm chúng vào Classpath máy tính của bạn, bạn có thể biên dịch servlet với trình biên dịch Java của JDK hoặc bất kỳ trình biên dịch hiện hành khác.

### 4.5 Ưu nhược điểm của Servlet

* 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 tình 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.

## **HTML**

### 5.1 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 (thi 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 ngọn (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.

### 5.2 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).

### 5.3 Cấu trúc một đoạn HTML **<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.

### 5.4 HTML đóng vai trì 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.

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.

## **CSS**

### 6.1 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.

### 6.2 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;

   .....

}

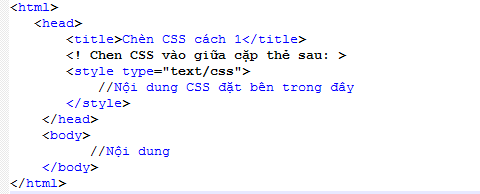
Nghĩa là nó sẽ được khai báo bằng vùng chọn, sau đó các thuộc tính và giá trị sẽ nằm bên trong cặp dấu ngoặc nhọn {}. Mỗi thuộc tính sẽ luôn có một giá trị riêng, giá trị có thể là dạng số, hoặc các tên giá trị trong danh sách có sẵn của CSS. Phần giá trị và thuộc tính phải được cách nhau bằng dấu hai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

### 6.3 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.

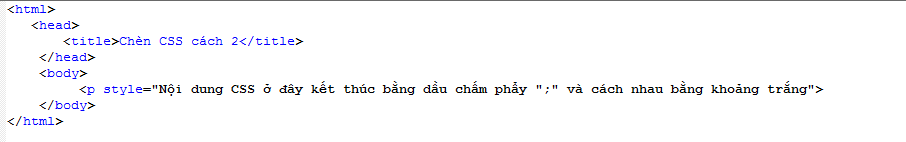
**Ví dụ:**



**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:**



**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 này 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:

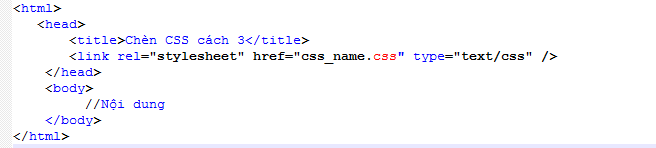
<link rel=”stylesheet” href=”Đường dẫn đên file .css” type=”text/css” />

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:**

****

## **JAVASCRIPT**

### 7.1 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 Javascipt.

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

### 7.2 Đặc điểm của Javascript

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

### 7.3 Cách nhúng Javascript

#### 7.3.1 Viết trực tiếp trong file HTML

<script type=”text/javascript”>

//Các lệnh Javascript

</script>

#### 7.3.2 Nhúng file Javascript từ bên ngoài vào file HTML

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

#### 7.3.3 Viết trực tiếp trong thẻ HTML

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

### 7.4 Ưu điểm của Javascript

* 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.

## **AJAX**

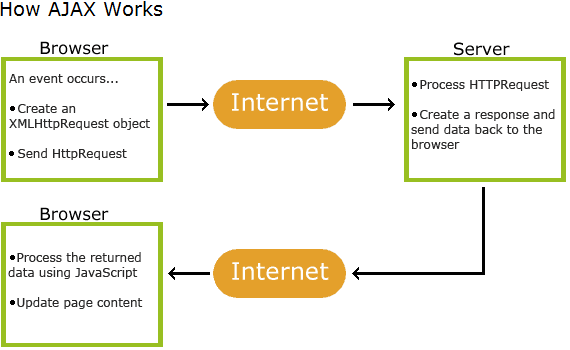
### 8.1 Ajax là gì?

Nội dung định nghĩa của Garrett về AJAX trong bài báo của mình như sau:AJAX không phải là một công nghệ. Nó là tập hợp của nhiều công nghệ với thế mạnh của riêng mình để tạo thành một sức mạnh mới. AJAX bao gồm:

* Thể hiện Web theo tiêu chuẩn XHTML và CSS.
* Nâng cao tính năng động và tương tác bằng DOM.
* Trao đổi và xử lý dữ liệu bằng XML và XSLT.
* Truy cập dữ liệu theo kiểu bất đồng bộ (asynchronous) bằng XMLHttpRequest.
* Và tất cả các kỹ thuật trên được liên kết lại với nhau bằng JavaScript.

Trong các thành phần cấu thành trên,điểm mấu chốt của AJAX nằm ở XMLHttpRequest.Tiếp đến chúng ta phân tích những ưu nhược điểmcủa ứng dụng sử dụng AJAX.

### 8.2 Hoạt động của Ajax



AJAX cho phép tạo ra một AJAX Engine nằm giữa giao tiếp này. Khi đó, các yêu cầu gửi (resquest) và nhận (response) do AJAX Engine thực hiện. Thay vì trả dữ liệu dưới dạng HTML và CSS trực tiếp cho trình duyệt, Web server có thể gửi trả dữ liệu dạng XML và AJAX Engine sẽ tiếp nhận, phân tách và chuyển hóa thành XHTML + CSS cho trình duyệt hiển thị. Việc này được thực hiện trên client nên giảm tải rất nhiều cho server, đồng thời người sử dụng cảm thấy kết quả xử lý được hiển thị tức thì mà không cần nạp lại trang. Mặt khác, sự kết hợp của các công nghệ Web như CSS và XHTML làm cho việc trình bày giao diện trang Web tốt hơn nhiều và giảm đáng kể dung lượng trang phải nạp. Đây là những lợi ích hết sức thiết thực mà AJAX đem lại.

### 8.3 Sử dụng Ajax khi nào và ở đâu?

* Khi cần tự động update nhưng loại bỏ Resfresh (ví dụ: Bảng tỷ số, chứng khoán, đấu giá, dự báo thời tiết...)
* Tối thiểu tần số tự động update để giảm tải cho server, cung cấp link refresh dễ nhận biết cho người dùng
* Ý thức việc sử dụng Back/Forward button, hoặc chia sẻ URL của user
* Loại bỏ sự phụ thuộc giữa thông tin và các trang web (ví dụ Google Maps, Ajax pagination), sử dụng XML và raw Data
* Kết hợp Ajax với Traditional HTML để hỗ trợ tối ưu hóa máy tìm kiếm
* Đơn giản hóa các thao tác thêm, sửa, xóa, thông báo
* Giúp người sử dụng nhận biết các action đang được thực hiện (Sử dụng Ajax indicator image : Loading/Saving/Success
* Tránh tạo các ứng dụng hoàn toàn bằng Ajax

### 8.4 Trình duyệt hỗ trợ Ajax

* Apple Safari từ 1.2 trở lên
* Konqueror
* Microsoft Internet Explorer từ 4.0 trở lên
* Mozilla/Mozilla Firefox từ 1.0 trở lên
* Netscape từ 7.1 trở lên
* Opera từ 8.0 trở lên
* GreenBrowser 4.3
* Google Chrome tất cả các phiên bản.

### 8.5 Ứng dụng Ajax phổ biến

* Google Suggest ,Google Maps,Google Groups
* Orkut, Gmail

### 8.6 Để sử dụng tốt Ajax?

* Nắm vững nội dung một trang HTML
* Hiểu rõ vai trò “trình thông dịch” của web browser
* Sử dụng javascript để truy cập các điều khiển HTML
* Hiểu về cấu trúc tài liệu XML
* Hiểu về cơ chế truyền thông giữa web server với ajax engine.

### 8.7 Ưu nhược điểm của Ajax

#### 8.7.1 Ưu điểm

* Truy cập bất đồng bộ
* Đa dạng trong tương tác
* CÓ thể tạo các ứng dụng phức tạp
* Giảm tải cho server
* Tốc độ response nhanh hơn
* Người dùng tiếp tục làm các thao tác khi đợi
* Là công nghệ của Brower, độc lập với Web Server

#### 8.7.2 Nhược điểm

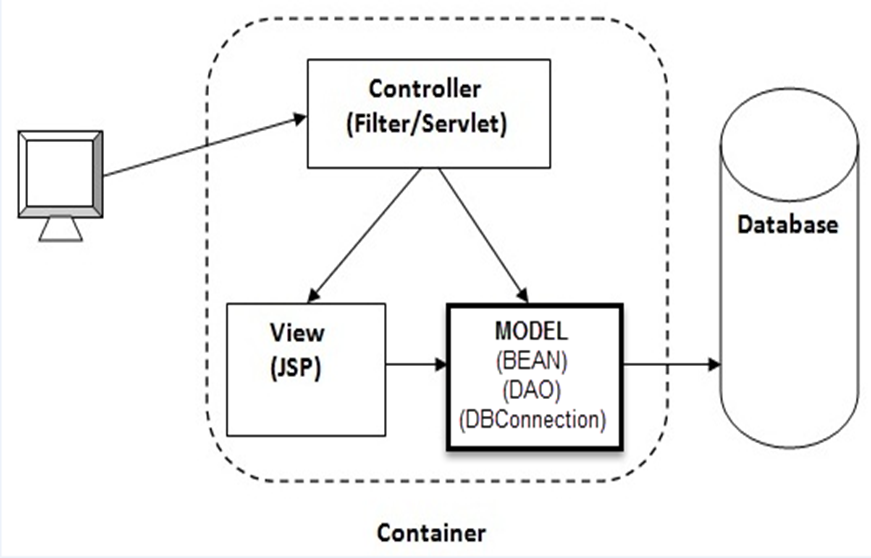
* Các trang web được tạo động không được ghi vào bộ lưu lịch sử lướt web của trình duyệt, do đó nút "back" (quay lui) của trình duyệt sẽ mất tác dụng quay lại trang thái trước đó của trang sử dụng Ajax, thay vào đó sẽ quay lại trang web trước đó mà người dùng ghé thăm. Để khắc phục có thể dùng các IFrame không hiển thị để gây ra sự thay đổi trong lịch sử trình duyệt và thay đổi phần neo của URL (bằng mã a #) khi chạy Ajax và theo dõi những sự thay đổi của nó.
* Việc cập nhật các trang web động cũng gây khó khăn cho người dùng trong việc bookmark (đánh dấu địa chỉ yêu thích) một trạng thái nào đó của ứng dụng. Cũng có những cách khắc phục cho vấn đề này, một số trong đó sử dụng mã xác định đoạn (fragment identifier) URL (phần URL ở sau dấu '#') để lưu vết, và cho phép người dùng đánh dấu và quay lại một trạng thái nào đó của ứng dụng.
* Do hầu hết các web crawler không thực thi mã JavaScript, các ứng dụng web sẽ cung cấp một phương thức thay thế để truy cập nội dung thông thường được truy cập bằng Ajax, để cho phép các máy tìm kiếm lập chỉ mục chúng.
* Bất kỳ người dùng nào có trình duyệt không hỗ trợ Ajax hay JavaScript, hoặc đơn giản là đã bị vô hiệu hóa JavaScript, sẽ đương nhiên không thể sử dụng Ajax. Tương tự, các thiết bị như điện thoại di động, PDA, và thiết bị đọc màn hình (screen reader) có thể không hỗ trợ JavaScript hay đối tượng XMLHttp được yêu cầu. Ngoài ra, các thiết bị đọc màn hình nếu có thể sử dụng Ajax đi nữa cũng vẫn có thể không đọc chính xác các nội dung động.
* Chế độ same origin policy (chế độ gốc đơn điệu) có thể không cho phép sử dụng Ajax thông qua các tên miền,[6] mặc dù W3C đã có một đồ án sơ thảo để cho phép điều này.
* Việc thiếu các chuẩn cơ bản của Ajax đồng nghĩa với việc không có nhiều sự chọn lựa thực tiễn tốt nhất để kiểm tra các ứng dụng Ajax. Các công cụ kiểm thử cho Ajax thường không hiểu các mô hình sự kiện, mô hình dữ liệu và giao thức của Ajax.
* Mở ra một cách thức khác cho việc tấn công của các đoạn mã độc mà những nhà phát triển web có thể không kiểm thử hết được.

# **PHẦN BA: THIẾT KẾ VÀ CÀI ĐẶT**

## **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

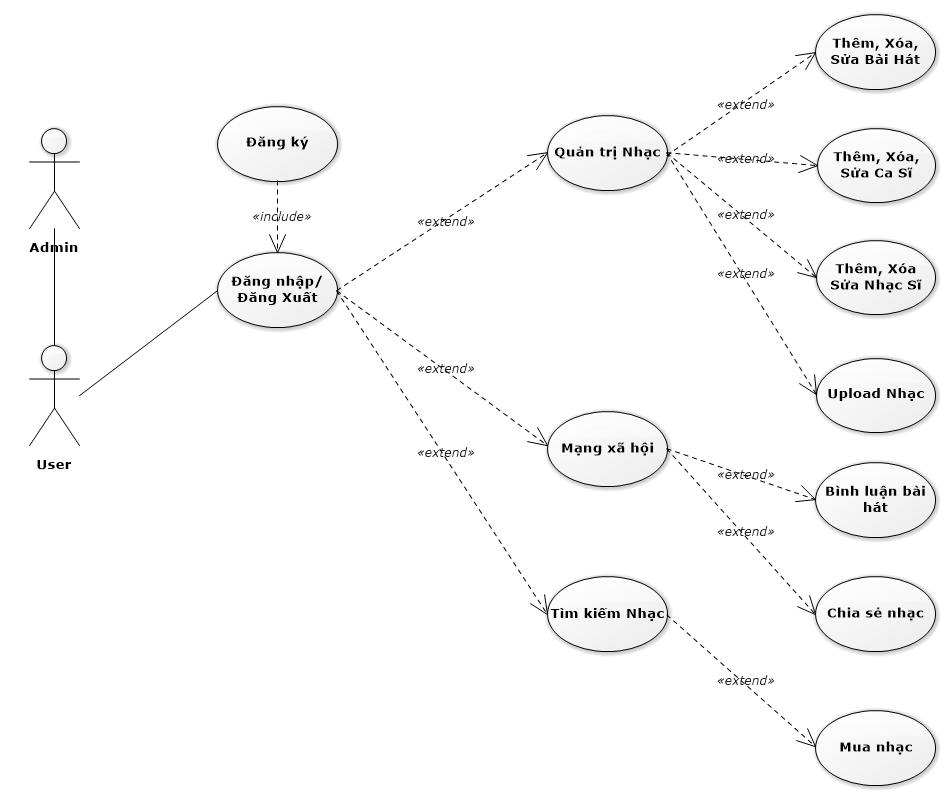


* Controller do Servlet đảm nhiệm nên MVC dễ dàng viết và chỉnh sửa code java
* View do JSP đảm nhiệm nên dễ dàng thiết kế và chỉnh sửa giao diện

Phân tách rõ ràng 3 tầng DAL, BAL, Presentation

## **SƠ ĐỒ USE – CASE**

### 2.1 Sơ đồ Use case



### 2.2 Danh sách các Actor

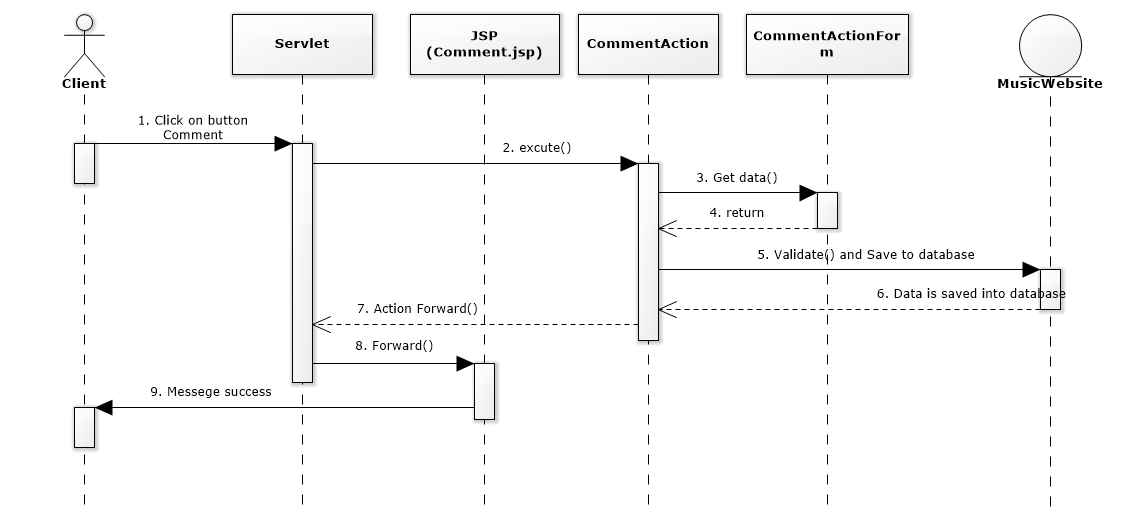
### 2.3 Danh sách các Use Case

## **SƠ ĐỒ SEQUENCE**

### 3.1 Chức năng Upload nhạc

### 3.2 Chức năng Comment – Share

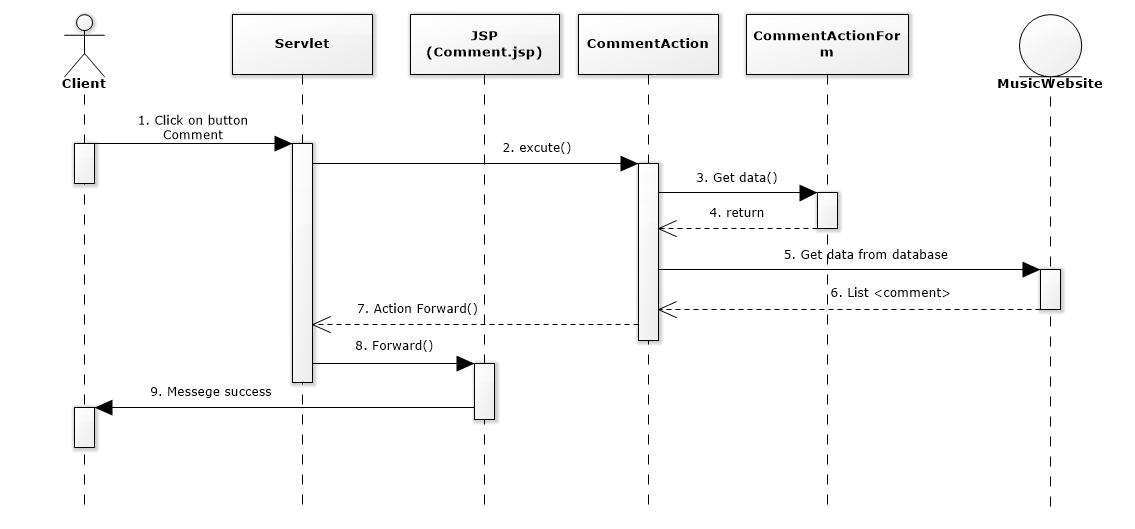
#### 3.2.1 Lưu comment theo User



Mô tả Sequence Diagram:

|  |  |
| --- | --- |
| **Luồng** | **Diễn giải** |
| 1 | User nhập comment vào và nhấn button Comment trên giao diện |
| 2 | Servlet gọi phương thức excute() |
| 3 | Action gọi lấy dữ liệu từ Action Form |
| 4 | ActionForm trả dữ liệu về cho Action |
| 5 | Action kiểm tra dữ liệu và lưu xuống database |
| 6 | Thông báo lưu comment thành công |
| 7 | Action forward kết quả cho Servlet |
| 8 | Servlet forward kết quả cho trang JSP |
| 9 | Hiển thị thông báo cho user |

#### 3.2.2 Load Comment theo bài hát



Mô tả Sequence Diagram:

|  |  |
| --- | --- |
| **Luồng** | **Diễn giải** |
| 1 | User nhập comment vào và nhấn button Comment trên giao diện |
| 2 | Servlet gọi phương thức excute() |
| 3 | Action gọi lấy dữ liệu từ Action Form |
| 4 | ActionForm trả dữ liệu về cho Action |
| 5 | Action lấy data từ cơ sở dữ liệu |
| 6 | List comment với user tương ứng |
| 7 | Action forward kết quả cho Servlet |
| 8 | Servlet forward kết quả cho trang JSP |
| 9 | Hiển thị thông báo cho user |

### 3.3 Chức năng tìm kiếm nhạc

## **GIAO DIỆN ỨNG DỤNG**

# **NHÂN XÉT VÀ KẾT LUẬN**

* Nhóm em phần nào cũng đã hoàn thiện và đáp ứng yêu cầu của đồ án
* Có một số khó khăn trong việc lập trình cũng như viết báo cáo (Báo cáo chưa hoàn thiện cho lắm và có một số chỗ sai sót)
* Bên cạnh đó, còn một số vấn đề cần được khắc phục như:
  + Các lỗi tiềm ẩn trong phần mềm
  + Giao diện chương trình chưa đẹp cho lắm
  + Phần mềm chỉ chạy trên các máy tính có cài đặt đầy đủ chương trình hỗ trợ nên khó phát triển rộng rãi

# **HƯỚNG PHÁT TRIỂN**

* Sửa các lỗi hiện có
* Phát triển thêm việc kết nối giữa Client và Server (dưới sự hỗ trợ của Window Server) để triển khai ứng dụng rộng rãi dễ dàng hơn.

# **PHỤ LỤC**

* Bảng phân công công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ tên** | **MSSV** | **Mức độ hoàn thành** | **Công việc được giao** |
| **1** | Nguyễn Văn Nguyện | 13520567 |  |  |
| **2** | Huỳnh Ngọc Thắng | 13520775 |  |  |
| **3** | Trần Minh Tuấn | 13520567 |  |  |

Nhóm xin chân thành gửi lời cảm ơn đến Giảng Viên đã dạy và hướng dẫn chúng em trong thời gian vừa qua để hoàn thành tốt đồ án này một cách tốt nhất và hoàn thiện nhất,xin chân thành cảm ơn!

# **TÀI LIỆU THAM KHẢO**

[1]. https://struts.apache.org/

[2]. https://vi.wikipedia.org/wiki/Struts

[3]. Struts 2 Design and Programming Second Edition by Kudi Kurniawan

[4] Google, YouTube

[5] Một số nguồn tài liệu khác