**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**Tài liệu công nghệ**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ THÔNG TIN DU LỊCH**

**Sinh viên thực hiện:**

**Trần Lê Quế Ngọc B1304707**

**Lê Như Ý B1304751**

**Khóa: K39**

**Cần Thơ, 04/2017**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



**LUẬN VĂN TỐT NGHIỆP**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**Tài liệu công nghệ**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ THÔNG TIN DU LỊCH**

**Người hướng dẫn: Sinh viên thực hiện:**

**TS/ Ths. Võ Huỳnh Trâm Trần Lê Quế Ngọc B1304707**

**Lê Như Ý B1304751**

**Khóa: K39**

***Cần Thơ, 04/2017***

Mục lục

[LỜI GIỚI THIỆU 1](#_Toc482544925)

[Bảng chú thích thuật ngữ 2](#_Toc482544926)

[I. Spring MVC Framework 4](#_Toc482544927)

[1. Đôi nét về Spring Framework 4](#_Toc482544928)

[2. Lịch sử ra đời 5](#_Toc482544929)

[3. Lịch sử phiên bản 6](#_Toc482544930)

[4. Đặc điểm nổi bật 6](#_Toc482544931)

[5. Các mô-dun 9](#_Toc482544932)

[5. 1 Spring Core 11](#_Toc482544933)

[5. 2 Spring Bean 11](#_Toc482544934)

[5. 3 Dependency Injection (DI) 12](#_Toc482544935)

[5. 4 Spring Context 12](#_Toc482544936)

[5. 5 Spring Expression Language (SpEL) 12](#_Toc482544937)

[5. 6 Spring MVC 12](#_Toc482544938)

[5. 7 Spring Data 12](#_Toc482544939)

[5. 8 Spring Security 12](#_Toc482544940)

[5. 9 Spring Boot 12](#_Toc482544941)

[5. 10 Spring Batch 13](#_Toc482544942)

[5. 11 Spring Integration 13](#_Toc482544943)

[5. 12 Spring XD 13](#_Toc482544944)

[5. 13 Spring Social 13](#_Toc482544945)

[6. Ưu điểm khi dùng Spring Framework 13](#_Toc482544946)

[II. Hibernate Framework 14](#_Toc482544947)

[1. Đôi nét về Hibernate Framework 14](#_Toc482544948)

[2. Giới thiệu về Hibernate 14](#_Toc482544949)

[3. Kiến trúc Hibernate 15](#_Toc482544950)

[4. Ưu điểm của Hibernate 15](#_Toc482544951)

[III. Maven 17](#_Toc482544952)

[1. Đôi nét về Maven 17](#_Toc482544953)

[2. Cách thức hoạt động của Maven 17](#_Toc482544954)

[3. Ưu điểm của Maven 18](#_Toc482544955)

[IV. MySQL 19](#_Toc482544956)

[1. Đôi nét về MySQL 19](#_Toc482544957)

[2. MySQL là gì? 19](#_Toc482544958)

[3. SQL là gì? 19](#_Toc482544959)

[4. Tính năng nổi bật 20](#_Toc482544960)

[5. Ưu điểm 20](#_Toc482544961)

[V. HTML, CSS, JavaScript 21](#_Toc482544962)

[1. HTML 21](#_Toc482544964)

[2. CSS 21](#_Toc482544965)

[3. JavaScript 22](#_Toc482544966)

[4. Các lợi thế của việc sử dụng JavaScript 22](#_Toc482544967)

[VI. Bootstrap 23](#_Toc482544968)

[1. Đôi nét về Bootstrap 23](#_Toc482544969)

[2. Lịch sử Bootstrap 23](#_Toc482544970)

[3. Các đặc điểm chính 23](#_Toc482544971)

[4. Ưu điểm 24](#_Toc482544972)

[TÀI LIỆU THAM KHẢO 25](#_Toc482544973)

# LỜI GIỚI THIỆU

Xây dựng một ứng dụng web hay web application (gọi tắt là web app) giúp chúng ta có thể tiếp cận hệ thống thông qua mạng như Internet hay Intranet đang là xu hướng và hướng đi của nhiều lập trình viên Java 2 Platform, Enterprise Edition (gọi tắt là J2EE). Ứng dụng web với khả năng cập nhật và bảo trì ứng dụng mà không phải phân phối và cài đặt phần mềm trên hàng ngàn máy tính là lý do chính cho sự phổ biến của nó. Ứng dụng web được dùng để hiện thực Web mail, bán hàng trực tuyến, web du lịch, đấu giá trực tuyến, wiki, diễn đàn thảo luận, Weblog, Massively Multiplayer Online Role - Playing Game (MMORPG), hệ quản trị nội dung, phần mềm quản lý nguồn nhân lực và nhiều chức năng khác. Trong đó việc xây dựng một “Hệ thống quản lý thông tin du lịch” đang là một trong những nhu cầu cấp thiết phục vụ cho ngành du lịch đang phát triển như vũ bão ở nước ta.

Để xây dựng một ứng dụng web tốt, ít tốn thời gian, chi phí phát triển và bảo trì thì việc ứng dụng những công nghệ kỹ thuật mới và kết hợp chúng lại với nhau đó là yếu tố hàng đầu, quyết định chất lượng và khả năng triển khai của một ứng dụng web. Những công nghệ phổ biến được sử dụng để xây dựng một web app như: Laravel (PHP), Django (Python), Ruby on Rails, ASP.NET, Spring MVC (Java)… cho server. HTML (HyperText Markup Language), CSS (Cascading Style Sheets), JavaScript, Ajax (Asynchronous JavaScript and XML), Bootstrap, jQuery (JavaScript Framework Library - commonly used in Ajax development), MooTools (JavaScript Framework Library - commonly used in Ajax development), Dojo Toolkit (JavaScript Framework Library - commonly used in Ajax development)… cho client. Và các ORM Framework: Athena Framework, EclipseLink, Enterprise Objects Framework, Hibernate.

Trong đó sự sự kết hợp giữa các công nghệ: Spring MVC Framework, Hibernate, Maven, HTML, CSS, JavaScript, Bootstrap và MySQL một trong những lựa chọn hoàn hảo và tối ưu cho việc phát triển ứng dụng web trên nền tảng J2EE.

# Bảng chú thích thuật ngữ

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ/ Từ viết tắt** | **Giải thích** |
| 1 | Maven | Công cụ quản lý và build project |
| 2 | MySQL | Hệ quản trị cơ sở dữ liệu |
| 3 | CSDL | Cơ sở dữ liệu |
| 4 | Hibernate | ORM Framework |
| 5 | Spring | Bộ khung phát triển ứng dụng Java |
| 6 | J2EE | Java 2 Platform, Enterprise Edition – Một nền lập trình dành cho việc phát triển ứng dụng phân tán kiến trúc đa tầng |
| 7 | EJB | Enterprise Java Beans – Một kiến trúc phát triển triển khai trên J2EE |
| 8 | J2SE | Java Standard Edition – Một nền tảng thực thi cho các ứng dụng Java |
| 9 | MVC | Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm |
| 10 | JDBC | Java Database Connectivity – Một API tiêu chuẩn dùng để tương tác với các loại cơ sở dữ liệu quan hệ |
| 11 | HTML | HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản |
| 12 | CSS | Cascading Style Sheets – Tập tin định kiểu theo tầng |
| 13 | JavaScript | Là một ngôn ngữ lập trình của HTML và Web |
| 14 | Bootstrap | Bootstrap là một thư viện HTML, CSS và JavaScript |
| 15 | Websocket | Công nghệ hỗ trợ giao tiếp hai chiều giữa client và server |
| 16 | IoC | Inversion of Control – Đảo ngược điều khiển |
| 17 | JTA | Java Transaction API – Một giao tiếp lập trình ứng dụng của J2EE |
| 18 | JDBC | Java Database Connectivity – Một API tiêu chuẩn dùng để tương tác với các loại cơ sở dữ liệu quan hệ |
| 19 | JDO | Java Data Objects |
| 20 | iBATIS | Là một Persistence Framework mã nguồn mở |
| 21 | DAO | Data Access Object – Đối tượng truy cập dữ liệu |
| 22 | JPA | Java Persistence API – Một đặc tả Java cho việc ánh xạ giữa các đối tượng Java tới CSDL quan hệ |
| 23 | ORM | Object Relational Mapping – Phương pháp lập trình chuyển đổi từ mô hình dữ liệu sang mô hình đối tượng |
| 24 | RDBMS | Relational Database Management System – Hệ thống quản lý cơ sở dữ liệu quan hệ |
| 25 | IDE | Integrated Development Environment – Môi trường phát triển tích hợp |

# Spring MVC Framework

## Đôi nét về Spring Framework



Spring Framework là một bộ khung ứng dụng và bộ chứa đảo ngược điều khiển cho nền tảng Java. Chức năng tính của bộ khung này có thể áp dụng cho bất kỳ ứng dụng Java nào, tuy vậy, nhiều bản mở rộng dành cho việc xây dựng ứng dụng nền web cũng được phát triển trên nền tảng J2EE. Tuy bộ khung không ấn định lên một đối tượng cụ thể nào, Spring Framework đã trở nên phổ biến trong cộng đồng Java như một phương án song hành, thay thế, hoặc thậm chí bổ sung cho mô hình Enterprise Java Beans (EJB).

Spring Framework cung cấp các giải pháp cho nhiều thách thức kỹ thuật mà các nhà phát triển phần mềm Java và các tổ chức muốn tạo ứng dụng dựa trên nền tảng Java phải đối mặt với Java platform. Bởi vì số lượng lớn chức năng của Spring Framework, việc phân loại những bộ phận cơ bản kiến tạo nên bộ khung là một việc rất khó. Spring Framework không chỉ được liên kết riêng biệt với Java Enterprise platform mặc dù sự tích hợp sâu của Spring trong Java EE là một yếu tố quan trọng cho sự phổ dụng, nổi tiếng của nó.

Có lẽ Spring Framework nổi tiếng nhất với sự cung cấp một cách hiệu quả các tính năng cần thiết để tạo ra các ứng dụng thương mại phức tạp, vượt lên khỏi các mô hình lập trình từng chiếm vị trí ưu thế trong lịch sử ngành công nghiệp phần mềm. Tiếp theo, nó cũng được ghi nhận về việc đưa các chức năng vốn ít được quen biết vào thực tế, có khi còn vượt cả ra bên ngoài nền tảng Java nữa.

Thành quả của việc này là nó đã tạo nên một bộ khung, cung cấp một mô hình nhất quán và làm cho nó có thể áp dụng vào hầu hết các loại ứng dụng được tạo trên nền tảng Java ngày nay. Bộ khung Spring được coi như thực thi một cách làm việc dựa trên những thói quen thực hành tốt nhất và những tiêu chuẩn công nghiệp, và đồng thời tạo cơ hội cho những lĩnh vực trong Java sử dụng nó.

## Lịch sử ra đời

Vào năm 2002, Spring Framework phát hành phiên bản đầu tiên bởi Rod Johnson. Việc xây dựng các ứng dụng doanh nghiệp trở nên đơn giản và dễ dàng hơn. Rod Johnson đã cho xuất bản một cuốn sách được phổ biến rộng rãi có nhan đề "J2EE Development without EJB". Điều này đã cho Spring Framework rất nhiều động lực để thay thế EJB.

Ngày nay, Spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring.

Phần đầu tiên của Spring Framework ban đầu được Rod Johnson viết vào năm 2000, trong khi ông đang làm một chuyên viên cố vấn cho các khách hàng trong thị trường kinh tế tài chính tại Luân Đôn. Trong lúc đang viết quyển Expert One-on-one J2EE Design And Development (Programmer to programmer) (Thiết kế và kiến tạo Java EE - Tương giao cá nhân một đối một chuyên ngành (Lập trình viên với lập trình viên)), ông đã mở rộng thêm phần mã nguồn, hòng biểu đạt ý tưởng của mình về mối tương quan của các trình ứng dụng với những phần khác nhau của J2EE và cho rằng người ta có thể đơn giản hóa mối tương quan của chúng và mối liên hệ ấy có thể trở nên cố định, bất biến hơn là những gì mà các lập trình viên và các công ty vốn quen sử dụng tại thời điểm đó.

Trước đây, vào năm 2001, những mô hình lập trình cho các trình ứng dụng cho mạng WEB được đại đa số sử dụng đều do Java Servlet API và EJB cung cấp. Cả hai đều là đặc tả do Sun Microsystems kiến tạo với sự cộng tác của các nhà buôn bán phần mềm và các bên quan tâm và chúng rất được mến chuộng trong cộng đồng Java. Những trình ứng dụng không dùng cho mạng WEB như các ứng dụng trình khách (Client-based application) hay các ứng dụng tác vụ (Batch application) đều có thể sử dụng các công cụ từ mã nguồn mở, các công cụ trên thị trường hoặc từ các dự án khác.

## Lịch sử phiên bản

Phiên bản đầu tiên của Spring được viết bởi Rod Johnson, cùng lúc đó ông cũng cho ra cuốn sách Expert One-on-One J2EE Design and Development vào tháng 10/2002. Ban đầu, bộ khung được phát hành dưới Giấy phép Apache 2.0 vào tháng 06/2003. Cột mốc đầu tiên, bản 1.0, được phát hành tháng 03/2004 và lần lượt các mốc phát hành kế tiếp vào tháng 09/2004 và tháng 03/2005. Phiên bản Spring 1.2.6 nhận Giải năng suất Jolt (Jolt Productivity award) và một Giải Cải tiến JAX (JAX Innovation Award) vào năm 2006. Spring Framework 4.0 được phát hành vào tháng 12/2013. Cải tiến đáng chú ý của phiên bản này gồm có hỗ trợ J2SE 8, Groovy 2, một số phần của J2EE 7 và Websocket.

Spring Framework 4.2.0 được phát hành ngày 31/07/2015 và ngay sau đó được nâng cấp lên phiên bản 4.2.1, được phát hành ngày 01/09 cùng năm. Phiên bản này được mô tả là "Tương thích với Java 6, 7 và 8, chú trọng vào những cải tiến cốt lõi và tính năng web hiện đại".

Spring Framework 4.3 được phát hành ngày 10/06/2016. Phiên bản 4.3.0.RC1 có hiệu lực. Nó "sẽ là thế hệ cuối cùng cho những yêu cầu của Spring 4 (Java 6+, Servlet 2.5+), chuẩn bị cho phiên bản mở rộng 4.3.x với thời gian hỗ trợ đến năm 2019".

Spring 5 được thông báo rằng sẽ dựng trên Reactive Streams tương thích với Reactor Core.

## Đặc điểm nổi bật

Spring Framework bao gồm bộ quản lý cài đặt dựa trên Java Beans, áp dụng lý thuyết đảo ngược quyền khống chế (IoC) làm cho việc mắc nối các chương trình ứng dụng trở nên nhanh chóng và dễ dàng hơn. Việc sử dụng IoC trong Spring Framework còn được ám chỉ đến với cái tên Dependency Injection (DI) (Nghĩa đen: Bơm hay tiếp thành phần phụ thuộc, từ bên ngoài vào, lúc cần thiết, thay vì phải tự tạo từ bên trong). Nguyên lý của IoC sử dụng trong Spring Framework là một kỹ thuật nhằm ngoại biên hóa (externalize) việc kiến tạo và quản lý những thành phần phụ thuộc. Lấy ví dụ trường hợp lớp "Foo" phụ thuộc vào một trường hợp cá biệt của lớp "Bar" để thi hình một nhiệm vụ nào đấy. Theo phương pháp truyền thống, "Foo" phải tự kiến tạo trường hợp cá biệt của lớp "Bar", dùng hành động "new", hoặc phải tiếp nhận trường hợp cá biệt của lớp này từ một lớp chế xuất nào đấy (factory class). Nếu sử dụng tiếp cận của IoC thì một tiến trình ngoại biên sẽ phải cung cấp trường hợp của "Bar" (hoặc một tiểu lớp của nó) cho "Foo" trong thời gian chạy (runtime). Đây chính là nguyên nhân tại sao "DI" là một thành ngữ diễn tả tính năng của Spring Framework rõ hơn "IoC".

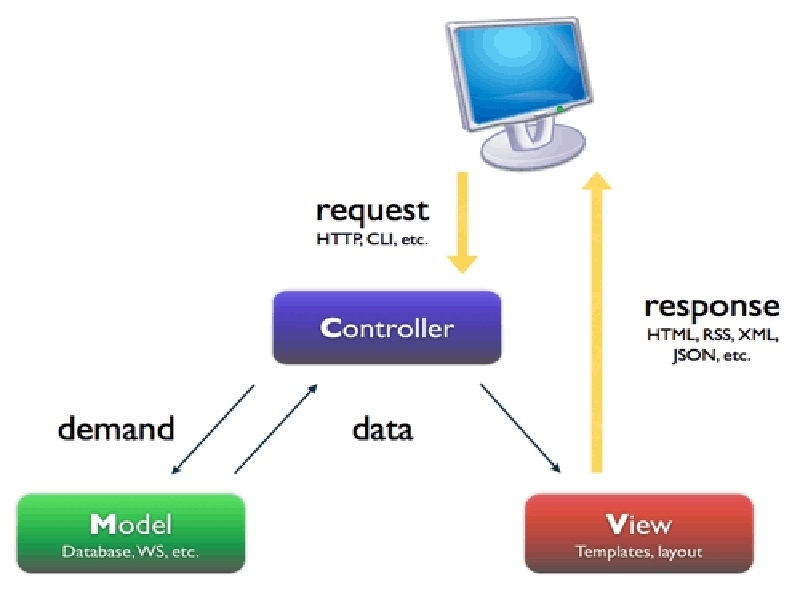
Bao gồm một mẫu hình chế xuất trừu tượng (abstract factory pattern) EJB nòng cốt, có thể sử dụng trong bất cứ môi trường nào, từ những khung chứa (containers) applet cho đến J2EE.

Một lớp trừu tượng chung quản lý giao dịch cơ sở dữ liệu (database transaction), cho phép các bộ phận quản lý giao dịch được bổ trợ theo lối lắp ráp (pluggable) và tạo điều kiện thuận lợi cho việc phân định ranh giới các giao dịch (demarcate transactions) mà không cần phải xử lý với những vấn đề ở tầng dưới (low-level issues).

Bao gồm một số các chiến lược phổ thông đã được cài sẵn bên trong (Built-in generic strategies) dành cho JTA và một đơn thể nguồn dữ liệu JDBC. Đối lập với những giao dịch do khung chứa quản lý (Container-Managed Transactions - CMT) đơn thuần trong JTA hoặc EJB, hỗ trợ giao dịch trong bộ khung Spring không đòi hỏi môi trường J2EE.

Tầng liên nối với cơ sở dữ liệu (JDBC abstraction layer) cung cấp một cơ chế tầng cấp đầy ý nghĩa đối với những ngoại lệ (exception hierarchy) (không còn phải dựa vào việc nắn bóp mã nguồn của nhà sản xuất phần mềm như đối với SQL Exception nữa), đơn giản hóa việc xử lý lỗi và cũng đồng thời thuyên giảm số lượng dòng lệnh mà lập trình viên cần phải viết. Lập trình viên sẽ không bao giờ phải viết khối 'finally' trong các dòng mã sử dụng JDBC nữa.

Tích hợp với Hibernate, JDO và iBATIS SQL Maps: trên phương diện cầm giữ tài nguyên (resource holders), hỗ trợ phần thực thi DAO và các chiến lược giao dịch (transaction strategies). Đứng hàng đầu trong việc hỗ trợ Hibernate với rất nhiều tính năng IoC tiện dụng, giải quyết nhiều nan đề trong việc tích hợp với Hibernate. Tất cả những tính năng này đều phục vụ cho cơ chế tầng cấp giao dịch chung và tầng cấp ngoại lệ DAO của Spring.



Mô hình MVC

Khung hình mô hình - giao diện - điều khiển (Model – View - Controller, hay MVC) linh hoạt dành cho ứng dụng mạng, được xây dựng trên tính năng nền tảng của Spring. Các nhà xây dựng phần mềm (lập trình viên) có quyền khống chế cao độ trong khung hình này thông qua những giao diện chiến lược (strategy interfaces), đồng thời điều tiết với những kỹ thuật đa giao diện (multiple view technologies) như JSP, FreeMarker, Velocity, Tiles, iText, và POI. Tầng trung gian của Spring có thể dễ dàng kết hợp với một tầng mạng sử dụng bất cứ một khung hình MVC mạng nào đó (web MVC framework), như Struts, WeBWork hoặc Tapestry.

Khung hình lập trình định dạng (Aspect-oriented programming - hay AOP) nhằm cung cấp các dịch vụ như quản lý giao dịch (transaction management). AOP cung cấp khả năng thực thi lôgic đan kết (crosscutting logic) - tức những lôgic tác động đến rất nhiều phần của chương trình ứng dụng - tại cùng một chỗ (nhóm lại) và để cho lôgic ấy áp dụng xuyên suốt chương trình ứng dụng của mình một cách tự động.

Vì những khó khăn trong việc kiến lập một cấu trúc mạch lạc để thiết kế phần mô hình (Model part) trong chương trình ứng dụng, khi áp dụng mô hình MVC (design pattern) (ví như Struts), khả năng Spring tạo điều kiện cho người sử dụng làm việc một cách dễ dàng có nghĩa là các nhà xây dựng có thể nhanh chóng điều chỉnh lại những tiếp cận không thành công của mình để lợi dụng tầng lớp JDBC trừu tượng của Spring Framework.

## Các mô-dun

Spring Framework được coi như là một tổ hợp của nhiều khung hình nhỏ, hay nói cách khác, nó là tổ hợp của khung hình trong khung hình. Đại bộ phận các khung hình này được thiết kế để hoạt động biệt lập với các khung hình khác, song lại tạo nên nhiều chức năng tốt hơn khi được phối hợp làm việc với nhau. Những khung hình này được phân loại theo các khối nền cơ bản mà những chương trình phần mềm phức tạp điển hình thường sử dụng:

* Khung chứa đảo ngược khống chế (Inversion of Control container): Sự cài đặt các thành phần của chương trình ứng dụng và quản lý chu trình của các đối tượng trong Java (lifecycle management of Java objects).
* Khung hình lập trình định dạng (Aspect-Oriented Programming Framework hay AOP Framework): Spring AOP module tích hợp chức năng lập trình hướng khía cạnh vào Spring Framework thông qua cấu hình của nó. Spring AOP module cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong bất kỳ ứng dụng nào sử dụng Spring. Với Spring AOP chúng ta có thể tích hợp declarative transaction management vào trong ứng dụng mà không cần dựa vào thành phần EJB.
* Spring AOP module cũng đưa lập trình metadata vào trong Spring. Sử dụng cái này chúng ta có thể thêm annotation vào source code để hướng dẫn Spring nơi và làm thế nào để liên hệ với aspect.
* Khung hình truy cập dữ liệu (Data Access Framework): Tầng JDBC và DAO đưa ra một cây phân cấp exception để quản lý kết nối đến database, điều khiển exception và thông báo lỗi được ném bởi vendor của database. Tầng exception đơn giản điều khiển lỗi và giảm khối lượng code mà chúng ta cần viết như mở và đóng kết nối. Module này cũng cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong ứng dụng Spring.
* Khung hình quản lý giao dịch (Transaction Processing/ Transaction Management Framework): Sự đồng hòa các API quản lý giao dịch và sự điều hợp quản lý giao dịch đối với các đối tượng Java dùng phương pháp cài đặt.
* Khung hình mô hình - giao diện - điều khiển (Model – View - Controller Framework): Khung hình dựa trên nền tảng HTTP và Servlet cung cấp rất nhiều móc điểm có thể mở rộng và cải biên (extension and customization).
* Khung hình truy cập từ xa (Remote Access Framework): Xuất nhập các đối tượng Java trên mạng lưới truyền thông hỗ trợ các giao thức dựa trên nền HTTP, sử dụng phong cách RPC (Remote Procedure Call) thông qua cài đặt, như RMI (Java Remote Method Invocation - khởi động phương pháp từ xa trong Java), CORBA và dịch vụ mạng (web services - SOAP).
* Khung hình xác thực và ủy quyền (Authentication and Authorization Framework): Sự điều hợp các quá trình xác thực và ủy quyền thông qua cài đặt hỗ trợ nhiều tiêu chuẩn công nghiệp và những tiêu chuẩn được sử dụng phổ biến, cũng như các giao thức, công cụ và các thói quen thực hành thông dụng.
* Khung hình quản lý từ xa (Remote Management Framework): Sự quản lý và quảng bá các đối tượng Java đối với các bố trí địa phương và bố trí ngoại biên thông qua JMX (Java Management Extension - quản lý mở rộng của Java cho phép các phần mềm không trực thuộc Java có thể lắp ráp vào và sử dụng được trong nền tảng Java).
* Khung hình truyền thông điệp (Messaging Framework): Đăng ký các đối tượng lắng nghe thông điệp dùng để tiếp thu và sử dụng thông điệp từ các hàng đợi thông điệp (message queue) thông qua JMS (Java Message Service - Dịch vụ thông điệp của Java), cải tiến việc truyền gửi thông điệp dùng tiêu chuẩn trong API của JMS.
* Khung hình kiểm thử phần mềm (Software Testing Framework): Những chủ thể (lớp) hỗ trợ việc viết các kiểm thử đơn vị và các kiểm thử tích hợp (unit tests and integration tests).

Để ngăn chặn sự phức tạp trong phát triển các ứng dụng, Spring Framework thường dựa trên các quan điểm như sau:

* Đơn giản hóa công việc phát triển thông qua việc sử dụng các đối tượng Java đơn giản hay còn được gọi là POJO (Plain Old Java Object).
* Nới lỏng ràng buộc giữa các thành phần thông qua việc sử dụng Dependency Injection và viết các interface.
* Tiếp cận lập trình khai báo bằng cách sử dụng các quy tắc (convention) và các khía cạnh (aspect) chung.
* Giảm thiểu các mã nghi thức và soạn sẵn (boilerplate) thông qua việc sử dụng các khuôn mẫu (template) và các khía cạnh.

### **Spring Core**

Spring Core chính là thành phần trung tâm, cốt lõi của Spring Framework. Đây chính là nền tảng để xây dựng nên các thành phần khác.

Lược đồ dưới đây mô tả mối quan hệ giữa Spring Core với các thành phần khác trong Spring Framework:



### **Spring Bean**

Spring Bean là trung tâm của Spring Core và là trái tim của một ứng dụng Spring. Trái ngược với EJB, Spring Framework được thiết kế từ lõi bằng cách sử dụng các POJO hay các Spring Bean. Spring Bean có thể được hiểu là các đối tượng Java đơn giản. Điều này khá tương đồng với nguyên lý thiết kế "Đơn trách nhiệm" (single responsibility) của Robert C Martin trong lập trình hướng đối tượng.

### **Dependency Injection (DI)**

Dependency Injection (có thể dịch tiêm các thành phần phụ thuộc) là một sức mạnh nổi bật của Spring Framework. Trong các phiên bản gốc của EJB người ta không thấy sự xuất hiện của DI, nhưng có lẽ do nhận thấy sự thành công từ Spring khi sử dụng DI, các phiên bản EJB 3.x đã cung cấp các annotation để thực hiện inject các dependency. DI là một mẫu thiết kế phần mềm mà các đối tượng phụ thuộc sẽ được inject vào một lớp nào đó. DI là một implementation cụ thể của khái niệm IoC.

### **Spring Context**

Spring Context mang mọi thứ lại với nhau. Spring Context kế thừa các tính năng của Spring Bean và bổ sung các hỗ trợ cho internationalization (ví dụ như các resource bundle), event propagation, resource loading ... Ngoài ra, Spring Context cũng hỗ trợ các tính năng của J2EE như EJB, JMX và truy cập từ xa cơ bản. Interface Application Context là tiêu điểm của Spring Context.

### **Spring Expression Language (SpEL)**

Spring Expression Language là một ngôn ngữ ngắn gọn giúp cho việc cấu hình Spring Framework trở nên linh hoạt hơn.

### **Spring MVC**

Spring MVC được thiết kế dành cho việc xây dựng các ứng dụng nền tảng web. Đây là một dự án chúng ta không thể bỏ qua khi xây dựng các ứng dụng Java web.

### **Spring Data**

Cung cấp một cách tiếp cận đúng đắn để truy cập dữ liệu từ cơ sở dữ liệu quan hệ, phi quan hệ, map-reduce và thậm chí còn hơn thế nữa.

### **Spring Security**

Spring Security cung cấp các cơ chế xác thực (authentication) và phân quyền (authorization) cho ứng dụng của bạn.

### **Spring Boot**

Spring Boot là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng.

### **Spring Batch**

Spring Batch giúp chúng ta dễ dàng tạo các lịch trình (scheduling) và tiến trình (processing) cho các công việc xử lý theo mẻ (batch job).

### **Spring Integration**

Spring Integration là một implementation của Enterprise Integration Patterns (EIP). Dự án này thiết kế một kiến trúc hướng thông điệp hỗ trợ việc tích hợp các hệ thống bên ngoài.

### **Spring XD**

Mục tiêu của Spring XD chính là đơn giản hóa công việc phát triển các ứng dụng Big Data.

### **Spring Social**

Spring Social sẽ kết nối ứng dụng của bạn với các API bên thứ ba của Facebook, Twitter, Linkedin...

## Ưu điểm khi dùng Spring Framework

* Xác định Template trước: Spring Framework cung cấp những template cho Hibernate, Struts, JDBC, v.v. Vì vậy, lập trình viên không cần phải viết quá nhiều code và có thể bỏ qua những bước cơ bản.
* Dễ dàng kiểm thử: DI làm cho việc kiểm thử phần mềm, ứng dụng được tiến hành dễ dàng hơn. Nếu như ứng dụng EJB yêu cầu máy chủ phải chạy lại ứng dụng thì Spring Framework thì không.
* Dung lương nhỏ: Spring Framework có dung lượng nhỏ nhờ vào mô hình POJO. Spring Framework không bắt lập trình viên phải kế thừa các lớp hay thực hiện qua bất kì giao diện nào. Đó là lí do Spring Framework được coi là bất khả xâm phạm.
* Phát triển nhanh: DI tạo nên tính năng đặc biệt của Spring Framework giúp hỗ trợ các framework khác và giúp cho việc phát triển ứng dụng J2EE dễ dàng hơn.
* Hỗ trợ khai báo: Spring Framework cung cấp những hỗ trợ khai báo về bộ nhớ đệm, xác nhận, giao dịch và định dạng.
* Tính trừu tượng: Nó cung cấp sự trừu tượng hóa cao cho các chi tiết J2EE như JMS, JDBC, JPA và JTA.

# Hibernate Framework

## Đôi nét về Hibernate Framework

****

Framework trong phần mềm là một khái niệm dùng để chỉ những “cấu trúc dùng để hỗ trợ đã được định nghĩa sẵn” mà trong đó những dự án phần mềm khác có thể sử dụng nó để phát triển. Một framework bao gồm những program hỗ trợ, core library và một ngôn ngữ lập trình để giúp phát triển và gắn những thành phần khác nhau của ứng dụng phần mềm lại với nhau. Hibernate là một trong những ORM Framework. Hibernate là một framework cho persistence layer. Như vậy sử dụng Hibernate framework giúp bạn phát triển ứng dụng nhanh và chỉ chú tâm vào những layer khác mà không cần chú tâm nhiều đến persistence layer (chịu trách nhiệm giao tiếp với data layer) nữa. Hibernate giúp lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh chóng. Hibernate cho phép truy vấn dữ liệu bằng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc bằng SQL thuần.

## Giới thiệu về Hibernate

Hibernate là một giải pháp ORM cho Java , một dự án open source chuyên nghiệp, một framework cho pesistence layer và là một thành phần cốt lõi của bộ sản phẩm JBoss Enterprise Middleware System (JEMS). JBoss, như chúng ta đã biết là một đơn vị của Red Hat, chuyên cung cấp các dịch vụ 24/7 về hỗ trợ chuyên nghiệp, tư vấn và huấn luyện sẵn sàng hỗ trợ bạn trong việc dùng Hibernate…

Hibernate ánh xạ các lớp Java với các bảng trong cơ sở dữ liệu và ánh xạ giữa các kiểu dữ liệu trong Java với các kiểu dữ liệu SQL. Hibernate giúp giảm thiểu các công việc liên quan đến nhiệm vụ xử lý dữ liệu thông thường trong phát triển ứng dụng.

Hibernate nằm giữa các đối tượng Java truyền thống và cơ sở dữ liệu để giải quyết tất cả các công việc trong lớp persistence dựa trên mô hình kỹ thuật ORM.

## Kiến trúc Hibernate

Hibernate sử dụng các file cấu hình để cung cấp các dịch vụ và đối tượng persistence cho ứng dụng.



Hibernate dùng rất nhiều loại Java API có sẵn như JDBC, JTA, JNDI (Java Naming and Directory Interface). JDBC cung cấp các chức năng làm việc với CSDL quan hệ ở mức thô sơ, trừu tượng, cho phép bất kỳ một CSDL nào dùng JDBC driver đều được hỗ trợ bởi Hibernate. JNDI và JTA cho phép Hibernate được tích hợp với các server ứng dụng J2EE.

## Ưu điểm của Hibernate

* Hibernate chịu trách nhiệm ánh xạ giữa các lớp Java đến các bảng trong CSDL dùng các file XML mà không cần phải viết bất kỳ dòng code nào.
* Cung cấp các API đơn giản để lưu trữ, truy xuất trực tiếp các đối tượng Java và CSDL.
* Nếu có bất kỳ thay đổi nào trong CSDL thì chỉ cần thay đổi file XML.
* Cung cấp đầy đủ các tiện ích, tính năng truy vấn dữ liệu đơn giản, hiệu quả.
* Thao tác, xử lý được các quan hệ phức tạp của các đối tượng trong CSDL.
* Giảm thiểu sự truy cập đến CSDL đến mức thấp nhất nhờ có chiến lược tìm, nạp thông minh.
* Hibernate hỗ trợ hầu hết các RDBMS quan trọng như: HSQL Database Engine, MySQL, Oracle, Microsoft SQL Server Database,…
* Hibernate hỗ trợ một số công nghệ sau: XDoclet Spring, J2EE, Eclipse plug-ins, Maven.

# Maven

## Đôi nét về Maven

****

Maven là một công cụ để quản lý project và việc build một cách tự động. Nó phục vụ chính cho các dự án viết bằng Java nhưng cũng có thể hỗ trợ cho các dự án khác viết bằng C#, Ruby hay Scala.

Apache Maven là chương trình quản lý dự án được sử dụng nhiều bởi các nhà phát triển Java bởi nó thực hiện các công việc như khởi tạo project, biên dịch, đóng gói, chạy test… một cách tự động và nhanh chóng. Nó đặc biệt hữu dụng với các dự án tầm trung và lớn.

Maven thường được so sánh với Apache Ant, nhưng Maven lại hoạt động khác hẳn với Apcache Ant.

Mỗi dự án sẽ được mô tả trong một file có định dạng XML có tên "pom.xml". File này sẽ chứa một số thông tin như tên dự án, các thư mục tài nguyên liên quan, các thư viện sử dụng… Khi bắt đầu biên dịch, trình Maven sẽ đọc file này và tự động tải các thư viện cần thiết từ repositories của nó thông qua mạng hoặc nơi người dùng đã định nghĩa.

## Cách thức hoạt động của Maven

Maven dùng khái niệm Project Object Model (POM) để mô tả việc build project, các thành phần phụ thuộc và các module. Nó định nghĩa trước các target cho việc khai báo công việc(task), trình biên dịch, đóng gói và thứ tự hoạt động để mọi việc diến ra tốt nhất.

Trong mỗi dự án Maven tạo ra một file .pom, trong file này định nghĩa ra những task như task khi chạy test, task khi build và khi chạy Maven sẽ dựa vào những định nghĩa này để thao tác với dự án.

## Ưu điểm của Maven

* Làm cho việc xây dựng hệ thống một cách dễ dàng hơn.
* Khi sử dụng Maven, người dùng không cần phải biết bộ máy phía dưới, Maven che chắn rất tốt các chi tiết cụ thể của hệ thống.
* Cung cấp cho người dùng một hệ thống xây dựng thống nhất.
* Maven cung cấp rất nhiều thông tin hữu ích mà một phần là được trích từ file POM của dự án Java đã build.
* Cung cấp thông tin dự án một cách chuyên nghiệp.
* Đưa ra hướng dẫn cụ thể nhất cho việc phát triển thực nghiệm.
* Maven giữ source code thử nghiệm của người dùng trong một nơi riêng.
* Maven cũng nhằm mục đích hỗ trợ trong công việc quản lý dự án, phát hiện và theo dõi vấn đề.
* Hướng dẫn về cách bố trí cây thư mục của dự án để người dùng tìm hiểu dễ dàng hơn.
* Cho phép tự động cập nhật những tính năng mới.
* Tự động hóa toàn bộ quá trình: Khởi tạo, cập nhật thư viện, build & test, release và tạo tài liệu mô tả cho dự án.
* Tự động hóa quá trình test với các plugin: Toàn bộ quá trình test được tự động, từ đó các lỗi được phát hiện sớm hơn. Thậm chí lỗi có thể phát hiện ở giai đoạn phát triển, do các nhà phát triển đều có thể chạy các test case một cách dễ dàng bằng Maven.
* Quản lý các phụ thuộc trong project một các ưu việt: Tự động cập nhật, mở rộng dễ dàng và đóng gói.
* Phân chia một dự án lớn thành các module nhỏ. Từ đó cho phép làm việc đồng thời trên các module khác nhau mà vẫn tạo được tính thống nhất.
* Không phụ thuộc vào IDE: Cho phép chuyển đổi từ Eclipse sang Netbeans,.. một cách dễ dàng.
* Hệ thống plugin phong phú trong đó có cả Ant.

# MySQL

## Đôi nét về MySQL

****

MySQL là hệ quản trị CSDL mã nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là CSDL tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau như: Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL là một trong những ví dụ rất cơ bản về Hệ quản trị CSDL quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP, Perl và nhiều ngôn ngữ lập trình khác.

## MySQL là gì?

MySQL là một chương trình dùng để quản lý hệ thống CSDL. Do đó có thể truy cập lấy dữ liệu một cách dễ dàng, MySQL hỗ trợ hầu hết các ngôn ngữ lập trình. Chính vì lẽ đó mà mã nguồn mở phổ biến nhất hiện nay là WordPress lại sử dụng MySQL làm cơ sở dữ liệu mặc định.

## SQL là gì?

SQL là viết tắt của Structured Query Language, là ngôn ngữ truy vấn mang tính cấu trúc. SQL được thiết kế để quản lý dữ liệu trong một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS). SQL là ngôn ngữ cơ sở dữ liệu, được sử dụng để tạo, xóa trong cơ sở dữ liệu, lấy các hàng và sửa đổi các hàng.

## Tính năng nổi bật

MySQL là một trong những phần mềm quản trị CSDL dạng server based, hệ này gần giống với SQL Server của Microsoft.

MySQL là phần mềm quản lý dữ liệu thông qua CSDL, mỗi một CSDL đều có bảng quan hệ chứa dữ liệu riêng.

MySQL có cơ chế quản lý sử dụng riêng, mỗi người sử dụng đều có thể quản lý một hoặc nhiều cơ sở dữ liệu khác nhau. Mỗi người dùng đều có một tên truy cập (username) và mật khẩu (password) để đăng nhập và truy xuất đến CSDL.

## Ưu điểm

* MySQL là CSDL có tốc độ khá cao, ổn định và dễ sử dụng, có thể hoạt động được trên nhiều hệ điều hành. Với tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có sử dụng CSDL trên hệ thống internet.
* MySQL hoàn toàn được sử dụng miễn phí. MySQL không chỉ dừng lại ở hỗ trợ cho PHP và Perl mà nó còn hỗ trợ cho nhiều ngôn ngữ khác.
* Khả năng mở rộng và tính linh hoạt. Ngoài ra, MySQL còn chạy được trên hầu hết hệ điều hành hiện nay.
* Máy chủ cơ sở dữ liệu MySQL cung cấp khả năng mở rộng tối đa, có khả năng xử lý các ứng dụng nhúng sâu chỉ với 1 MB để chạy các kho dữ liệu khổng lồ chứa terabyte thông tin.
* Hiệu năng cao: Dù một hệ thống xử lý giao dịch tốc độ cao hay một trang web có dung lượng lớn, phục vụ hàng tỷ truy vấn một ngày thì MySQL vẫn có thể đáp ứng nhanh chóng.
* Bảo vệ dữ liệu mạnh mẽ: MySQL cung cấp các tính năng bảo mật đặc biệt bảo vệ dữ liệu tuyệt đối, chỉ những người có thẩm quyền mới có quyền truy cập vào máy chủ cơ sở dữ liệu, đảm bảo rằng dữ liệu nhạy cảm được bảo vệ khỏi việc truy cập trái phép.
* Quản lý dễ dàng: MySQL cung cấp một bộ công cụ quản lý và chuyển đổi đồ họa cho phép quản trị hệ thống (Database Administrator – DBA) quản lý, khắc phục sự cố và kiểm soát hoạt động của nhiều máy chủ MySQL từ một máy trạm duy nhất.

# HTML, CSS, JavaScript

### 

## HTML

HTML là ngôn ngữ đánh dấu siêu văn bản, được sử dụng trong các tổ chức công nghệ truyền thông. HTML đã trở thành một chuẩn cho trang web trên Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản mới nhất: HTML5.

HTML được lưu trữ như là các tập tin văn bản - trên các máy chủ, để người dùng có thể truy cập thông qua mạng internet - ngoài ra tập tin HTML còn được lưu trữ tại máy tính cá nhân - sử dụng trình duyệt để xem dữ liệu hiển thị như thế nào và sử dụng một editor (Phầm mềm soạn thảo mã như notepad, notepad++, codelobster, dreamweaver,...) để soạn mã HTML.

Các file HTML chứa thẻ đánh dấu, là các chỉ thị cho chương trình về cách hiển thị, xử lý văn bản ở dạng thuần túy. Phần đuôi mở rộng của tập tin HTML thường có dạng .htm hay .html - ngoài ra còn có phtml (Nhúng mã nguồn php), cshtml (Nhúng mã nguồn ASP.NET - C#).

## CSS

CSS 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). 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.

Phương thức hoạt động của CSS là 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.

## JavaScript

JavaScript là một ngôn ngữ lập trình của HTML và WEB, được sử dụng phổ biến nhất như là một phần của các trang web, tạo các trang web động. JavaScript là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

JavaScript được biết đến đầu tiên với tên Mocha và sau đó là LiveScript, nhưng Hãng Netscape thay đổi tên của nó thành JavaScript, bởi vì sự phổ biến như là một hiện tượng của Java lúc bấy giờ. JavaScript xuất hiện lần đầu trong Netscape 2.0 năm 1995 với tên LiveScript. Core đa năng của ngôn ngữ này đã được nhúng vào Netscape, IE, và các trình duyệt khác.

## Các lợi thế của việc sử dụng JavaScript

* Sự tương tác Server ít hơn: Bạn có thể xác nhận đầu vào (input) người sử dụng trước khi gửi trang tới Server. Điều này làm tiết kiệm lưu lượng tải ở Server, nghĩa là Server của bạn tải ít hơn.
* Phản hồi ngay lập tức tới khách truy cập: Họ không phải chờ cho một trang web tải lại để thấy xem nếu họ đã quên hay nhập sai cái gì đó.
* Khả năng tương tác tăng lên: Có thể tạo các giao diện mà phản hồi lại khi người sử dụng rê chuột qua hoặc kích hoạt từ bàn phím.
* Giao diện phong phú hơn: Có thể sử dụng JavaScript để thêm các thành phần như Drag và Drop (DnD), các con trượt (Slider) cung cấp một giao diện giàu tính năng khi khách truy cập vào webiste.

# Bootstrap

## Đôi nét về Bootstrap

******

Bootstrap là một framework cho phép thiết kế website tương thích với mọi kích thước hiển thị của trình duyệt nhanh hơn và dễ dàng hơn.

Bootstrap được xây dựng dựa trên nền tảng HTML templates, CSS templates và các plugin Javascript tạo ra những thành phần cơ bản có sẵn như: kiểu chữ, biểu mẫu, biểu bảng, nút,… và nhiều thứ khác nữa.

## Lịch sử Bootstrap

Bootstrap được phát triển bởi Mark Otto và Jacob Thornton tại Twitter. Bootstrap được xuất bản như là một mã nguồn mở vào tháng 8/2011 trên GitHub. Bản mới nhất bây giờ là Bootstrap 3.3.7.

## Các đặc điểm chính

* Nền tảng tối ưu: Trong Bootstrap đã tạo sẵn một thư viện để lưu trữ mà các nhà thiết kế có thể sử dụng và tuỳ ý chỉnh sửa theo mục đích cá nhân. Điều này giúp cho việc phát triển website trở nên nhanh chóng bởi vì có thể lựa chọn một mẫu có sẵn phù hợp và thêm màu sắc, hình ảnh, video... là đã có ngay giao diện đẹp, tiết kiệm được thời gian và công sức.
* Tương tác tốt với điện thoại thông minh: Bootstrap hỗ trợ tính tương thích và viết theo xu hướng ưu tiên giao diện điện thoại trước, điều này cải thiện đáng kể hiệu suất trang web khi có người dùng truy cập bằng điện thoại.
* Giao diện đầy đủ, sang trọng: Giao diện của Bootstrap có màu xám bạc rất sang trọng và hỗ trợ gần như đầy đủ các thành phần mà một website hiện đại cần có. Cầu trúc HTML rõ ràng giúp bạn nhanh chóng nắm bắt được cách sử dụng và phát triển. Không những vậy, Bootstrap còn giúp website hiển thị tốt khi co dãn màn hình windows.
* Dễ dàng tuỳ biến: Để phù hợp cho nhiều loại website, Bootstrap cũng hỗ trợ thêm tính năng tùy chỉnh (customizer), có thể thay đổi gần như tất cả những thuộc tính của Bootstrap để phù hợp với yêu cầu của người sử dụng. Nếu những tuỳ chỉnh này vẫn không đáp ứng được yêu cầu cá nhân thì có thể chỉnh sửa trực tiếp trên mã nguồn của Bootstrap.

## Ưu điểm

* Rất dễ để sử dụng: Bootstrap được xây dựng dựa trên nền tảng HTML, CSS và Javascript nên chỉ cẩn có kiến thức cơ bản về ba ngôn ngữ trên là có thể sử dụng Bootstrap tốt.
* Tính năng tương thích (Responsive): Bootstrap cung cấp sẵn reponsive css phù hợp với các thiết bị như điện thoại, máy tính, tablet,…
* Tương thích với nhiều trình duyệt: Bootstrap tương thích với hấu hết các trình duyệt như: Chrome, Firefox, Internet Explorer, Safari, and Opera,… chỉ ngoại trừ IE 9 trở xuống.

# TÀI LIỆU THAM KHẢO

1. Ebook: Spring in Action, Fourth Edition Covers Spring 4
2. Ebook: Hibernate in Action
3. <https://vi.wikipedia.org/wiki/Spring_Framework>
4. <http://projects.spring.io/spring-framework/>
5. <https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>
6. <https://docs.jboss.org/hibernate/orm/3.2/api/>
7. <https://en.wikipedia.org/wiki/MySQL>
8. <https://dev.mysql.com/doc/refman/5.7/en/>
9. https://bootstrapdocs.com/v3.3.6/docs/