**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\*\*\*\*\*\*\*\*\***

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**KHOA HỌC MÁY TÍNH**

Giảng viên hướng dẫn: Ths. Hồ Đắc Quán

Sinh viên thực hiện : Nguyễn Thành Luân

Mã sinh viên: 16026801 Lớp: DHKHMT12A

TP. Hồ Chí Minh 2020

**LỜI CẢM ƠN**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

**Họ và tên sinh viên :** Nguyễn Thành Luân

**Mã sinh viên  :** 16026801

1. Nhận xét chung :

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

|  |  |
| --- | --- |
|  | *Ngày ....... tháng ........ năm .........*  NGƯỜI HƯỚNG DẪN  *(Ký tên và đóng dấu)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

|  |  |
| --- | --- |
|  | *Ngày ....... tháng ........ năm .........*  **GIẢNG VIÊN HƯỚNG DẪN**  *(Ký và ghi rõ họ tên)* |

# NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

**Họ và tên sinh viên: Nguyễn Thành Luân**

**Mã sinh viên: 16026801**

1. **Nhận xét chung:**

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

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

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

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

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

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

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

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

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

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

*Ngày ....... tháng ........ năm .........*

GIẢNG VIÊN HƯỚNG DẪN

*(Ký và ghi rõ họ tên)*

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

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

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

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

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

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

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

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

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

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

|  |  |
| --- | --- |
|  | *Ngày ....... tháng ........ năm .........*  GIẢNG VIÊN HƯỚNG DẪN  *(Ký và ghi rõ họ tên)* |

# MỤC LỤC

[NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP 5](#_Toc40734109)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 6](#_Toc40734110)

[MỤC LỤC 7](#_Toc40734111)

[MỤC LỤC HÌNH ẢNH 9](#_Toc40734112)

[MỤC LỤC CÁC BẢNG BIỂU 10](#_Toc40734113)

[CHƯƠNG 1. GIỚI THIỆU CHUNG ĐƠN VỊ THỰC TẬP 10](#_Toc40734114)

[1.1 Thông tin về đơn vị thực tập 10](#_Toc40734115)

[1.2 **Thông tin về vị trí sinh viên thực tập** 11](#_Toc40734116)

[CHƯƠNG 2. VẤN ĐỀ MÀ SINH VIÊN THAM GIA GIẢI QUYẾT/THỰC HIỆN TẠI ĐƠN VỊ THỰC TẬP 12](#_Toc40734117)

[2.1 Tóm tắt vấn đề mà sinh viên tham gia giải quyết/thực hiện tại Đơn vị/Doanh nghiệp thực tập 12](#_Toc40734118)

[**2.2** **Tiến độ công việc thực hiện** 12](#_Toc40734119)

[2.3 TÓM TẮT CÔNG VIỆC THỰC HIỆN THEO TIẾN ĐỘ 14](#_Toc40734120)

[WEEK 1 15](#_Toc40734121)

[1.1 OOP 15](#_Toc40734122)

[1.2 Core Java 1 16](#_Toc40734123)

[1.3 Git 17](#_Toc40734124)

[1.4 Build with maven 19](#_Toc40734125)

[WEEK 2 21](#_Toc40734126)

[2.2 Structured Query Language(SQL) 23](#_Toc40734127)

[2.1 Packages 25](#_Toc40734128)

[2.2 Classes 26](#_Toc40734129)

[2.3 Methods(Phương thức) 27](#_Toc40734130)

[WEEK 3 30](#_Toc40734131)

[3.1 Cascading Style Sheets(CSS) 30](#_Toc40734132)

[3.2 Javascript 33](#_Toc40734138)

[3.3 JSON 33](#_Toc40734139)

[WEEK 4 36](#_Toc40734140)

[4.1 Unit test là gì? 36](#_Toc40734141)

[4.2 SonarLint/Qube and Rule Profiles 36](#_Toc40734142)

[4.3 Software Development Framework Using Scrum 37](#_Toc40734143)

[WEEK 5 40](#_Toc40734144)

[5.1 Jira Agile 40](#_Toc40734145)

[5.2 Web Services (REST & SOAP) 46](#_Toc40734146)

[10: 49](#_Toc40734147)

[5.1 Hypertext Markup Language(HTML) 50](#_Toc40734148)

[5.3 Bootstrap 52](#_Toc40734149)

[WEEK 6 53](#_Toc40734150)

[WEEK 7 53](#_Toc40734151)

[WEEK 8 53](#_Toc40734152)

[WEEK 9 53](#_Toc40734153)

[WEEK 10 53](#_Toc40734154)

[CHƯƠNG 3. NHẬN XÉT, ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP 54](#_Toc40734155)

[3.1. Các nhận xét, đánh giá thực trạng của quá trình làm việc. 54](#_Toc40734156)

# MỤC LỤC HÌNH ẢNH

[Hình 1 Mô hình OOP 14](#_Toc40738732)

[Hình 2 Mô hình hoạt động của DVCS 17](#_Toc40738733)

[Hình 3 Maven 18](#_Toc40738734)

[Hình 4 Spring IO Platform 22](#_Toc40738735)

[Hình 5 Java packages 25](#_Toc40738736)

[Hình 6 Constructor java 28](#_Toc40738737)

[Hình 7 HTML & CSS 30](#_Toc40738738)

[Hình 8 Cấu trúc CSS 31](#_Toc40738739)

[Hình 9 JSON example 32](#_Toc40738740)

[Hình 10 Sonar Qube 36](#_Toc40738741)

[Hình 11 Mô hình Scrum 37](#_Toc40738742)

[Hình 12 To do list 39](#_Toc40738743)

[Hình 13 Mô hình Jira 39](#_Toc40738744)

[Hình 14 Luồng công việc Jira 40](#_Toc40738745)

[Hình 15 Giao diện Jira 41](#_Toc40738746)

[Hình 16 Giao diện Jira 42](#_Toc40738747)

[Hình 17 Version Jira 43](#_Toc40738748)

[Hình 18 Jira version 43](#_Toc40738749)

[Hình 19 Jira calendar 44](#_Toc40738750)

[Hình 20 Javascript 50](#_Toc40738751)

[Hình 21 Bootstrap 51](file:///C:\training\probation\BaoCao.docx#_Toc40738752)

# MỤC LỤC CÁC BẢNG BIỂU

# CHƯƠNG 1. GIỚI THIỆU CHUNG ĐƠN VỊ THỰC TẬP

## Thông tin về đơn vị thực tập

Công Ty TNHH DXC Technology Services Việt Nam là chi nhánh của tập đoàn DXC Technology được hình thành vào tháng 4 năm 2017 từ sự sáp nhập giữa CSC và mảng Dịch vụ Doanh nghiệp của Hewlett Packard nterprise (HPE). Có chi nhánh ở hơn 70 quốc gia trên toàn thế giới, với hơn 6.000 đối tác, hơn 160.000 nhân viên đến từ khắp các quốc gia và vùng lãnh thổ, riêng ở Việt Nam có hơn 1.200 người là nhân viên làm việc tại DXC. Công ty DXC Technology Services có trụ sở chính tại Mỹ.

*Trụ sở chính*

1775 Tysons Blvd Tysons, VA 22102, USA

*Chi nhánh tại TP Hồ Chí Minh*

Địa chỉ: Tòa Etown 5, 364 Cộng Hòa, phường 13, quận Tân Bình, Tp.Hồ Chí Minh, Việt Nam.

Số điện thoại: 028 39248730

Website: <https://www.dxc.technology/vn_vn>

Với sự độc lập về công nghệ, đội ngũ nhân viên tài năng toàn cầu, liên minh chặt chẽ với các nhà đối tác công nghệ hàng đầu thế giới và luôn ứng dụng các giải pháp công nghệ thế hệ mới, DXC đảm bảo sẽ đem lại thành công mỹ mãn cho khách hàng.

Tại Việt Nam, DXC là một trong những công ty hàng đầu trong ngành CNTT, cung cấp các dịch vụ phần mềm và các dịch vụ CNTT cho khách hàng toàn cầu và các doanh nghiệp trong nước. DXC luôn tiên phong áp dụng những giải pháp thế hệ mới với các công nghệ hiện đại nhất trong các lĩnh vực như Ứng dụng Di động, Trí tuệ Nhân tạo, hiện đại hóa các ứng dụng, tư vấn và triển khai SAP ERP tại các ngành công nghiệp khác nhau như Thương mại, Bảo hiểm, Ngân hàng, Sản xuất, Giáo dục, Chăm sóc Sức khỏe và Khoa học Đời sống..

Với hơn hai thập kỷ hoạt động tại Việt nam, DXC Việt Nam có một đội ngũ kỹ sư hơn 1,100 nhân viên đầy kinh nghiệm với trình độ chuyên môn cao, năng động, liên tục cập nhật những kỹ thuật công nghệ mới, sẵn sàng đáp ứng mọi nhu cầu đa dạng của khách hàng. Cùng với sự kết hợp hoàn hảo các giải pháp kinh doanh sáng tạo và công nghệ hiện đại, quy trình phát triển chuẩn quốc tế và sự am hiểu sâu sắc về các lĩnh vực chuyên ngành, DXC chắc chắn sẽ đem lại sự hài lòng và thành công cho khách hàng - đó là cung cấp những dịch vụ phát triển ứng dụng chất lượng cao với chi phí hợp lý.

Là một bộ phận quan trọng trong hệ thống phát triển toàn cầu của DXC, DXC tại Việt Nam đã xây dựng được một hình ảnh xuất sắc của một trung tâm chất lượng cao. DXC luôn áp dụng chặt chẽ những qui trình phát triển ứng dụng một cách nhất quán trong quá trình phát triển phần mềm để đảm bảo chất lượng sản phẩm, là chìa khóa cho dịch vụ tuyệt hảo. DXC Việt Nam tự hào là công ty đầu tiên tại Việt Nam đạt chuẩn CMMI Cấp 5 - cấp cao nhất về quy trình quản lý chất lượng phát triển phần mềm của Viện Công Nghệ Phần Mềm Mỹ (SEI) và Chứng chỉ ISO 27001:2013 (phiên bản mới nhất của hệ thống quản lý an toàn thông tin).

***Cơ cấu tổ chức***

Công ty có các bộ phận sau: Bộ phận Phát Triển Giải pháp Bảo hiểm (Insurance), Bộ phận Phát triển Giải pháp Y tế và Khoa học Đời sống, Bộ phận các ứng dụng Java/IBM/Web và Hiện đại hoá các ứng dụng, Bộ phận Công nghệ Web, Cơ sở dữ liệu và Công nghệ mới, Bộ phận Di động và SAP, Bộ phận Dịch vụ Cơ sở hạ tầng, Bộ phận Kiểm định phần mềm, Bộ phận Quy trình và Công nghệ, Bộ phận Nhân sự, Phòng Kinh doanh.

## **Thông tin về vị trí sinh viên thực tập**

Vị trí công tác: Thực tập sinh cho vị trí front-end(Pointin)

Để đảm bảo sản phẩm luôn luôn được cải tiến và phát triển phiên bản hoàn thiện trước khi phân phối cho khách hàng ở các thị trường nghiêm ngặt và khó tính ở các quốc gia phát triền thì công ty luôn có một đội ngũ Lập trình viên vá lỗi, bảo trì và phát triển các chức năng của hệ thống.

Lập trình viên sẽ được tham gia vào giai đoạn bảo trì, vá lỗi và phát triển phần mềm.

* Đọc hiểu các tài liệu giải pháp, tài liệu thiết kế; phối hợp với các nhóm để hiểu rõ yêu cầu sản phẩm.
* Tiếp nhận các lỗi từ nhóm Kiểm thử. Thực hiện tái hiện lại lỗi.
* Phân tích, nắm rõ các lỗi, luồng đi của hệ thống, cách xử lí khi gặp sự cố.
* Thảo luận cách giải quyết với các thành viên trong team ở Việt Nam và nước ngoài.
* Thực hiện code vá lỗi.
* Viết báo cáo phân tích, cách xử lí lỗi, viết tài liệu về vấn đề đã giải quyết.
* Quản lí task, thời gian tiến độ dự án.

# CHƯƠNG 2. VẤN ĐỀ MÀ SINH VIÊN THAM GIA GIẢI QUYẾT/THỰC HIỆN TẠI ĐƠN VỊ THỰC TẬP

## Tóm tắt vấn đề mà sinh viên tham gia giải quyết/thực hiện tại Đơn vị/Doanh nghiệp thực tập

- Vá lỗi phát sinh trong trong sản phẩm.

- Sinh viên được training các kỹ năng về code: Java OOP, luồng đi của hệ thống, cách hệ thống làm việc, cách database configuration xử lí cho từng nghiệp vụ cụ thể.

- Các công nghệ có thể nâng cao hiệu suất của hệ thống.

- Được training các kĩ năng mềm qua các khoá học trực tuyến.

- Sinh viên được phép tham gia đóng góp ý kiến vào việc cải thiện/xây dựng quy trình phù hợp hơn với dự án phát triển.

- Sinh viên sẽ tham gia vào dự án phát triển phần mềm thực tế của công ty theo quy trình đã định sẵn.

## **Tiến độ công việc thực hiện**

Thực hiện 10-12 tuần. Thời gian bắt đầu từ: 03/02/2020 đến 30/05/2020

|  |  |
| --- | --- |
| Thời gian | Nội dung công việc theo tuần |
| Tuần 1 | Được training các kiến thức như   1. Bảo mật thông tin 2. Cách sử dụng các chương trình của công ty   Được training gồm có   1. OOP 2. Core Java 3. Git 4. Build with Maven   Thực hiện một số bài quiz cuối tuần |
| Tuần 2 | Được training các kiến thức như   1. Springboot 2. SQL   Thực hiện một số bài quiz cuối tuần |
| Tuần 3 | Được trainig các kiến thức như   1. HTML5 & CSS 1-2 2. JavaScript 3. JSON   Thực hiện một số bài quiz cuối tuần |
| Tuần 4 | Được training các kiến thức về linux như:   1. Unit Test with Junit 2. SonarLint/Qube and Rule Profiles 3. Software Development Framework Using Scrum   Thực hiện một số bài quiz cuối tuần |
| Tuần 5 | Được training các kiến thức như   1. JIRA Agile and Workshop 2. Web Services (REST & SOAP)   Thực hiện một số bài quiz cuối tuần |
| Tuần 6 | Bắt đầu vào dự án và được training các kiến thức như   1. Installation Environment 2. Work with commfw log. 3. Commfw stuff(mapping response, request, record xml.) 4. Copy book |
| Tuần 7 | Được training các kiến thức như   1. UOW- How BE handle a request. 2. Process Management |
| Tuần 8 | Được training các kiến thức như   1. How to run BNRs ( Batch and Online) 2. Run Basic 8 BNRs 3. BNRs- How BE handle. |
| Tuần 9 | Được training các kiến thức như   1. Work with UMT ( request, response) 2. Batch Job ( Daily Driver) |
| Tuần 10 | 1. Work with Utilities |
| Tuần 11 | Practise with real bug  Jira Stuff |

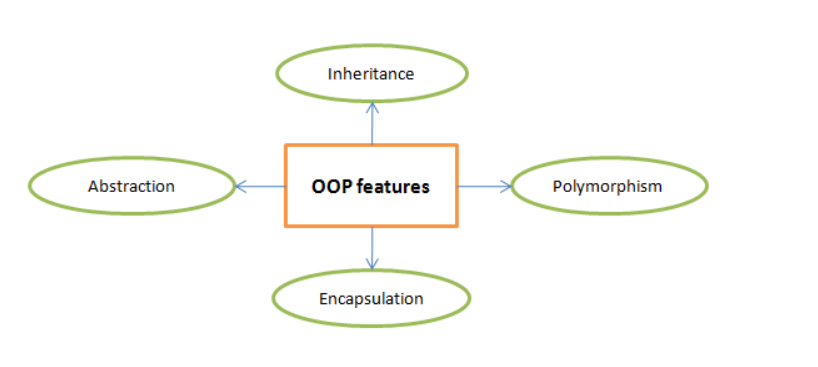
## 2.3 TÓM TẮT CÔNG VIỆC THỰC HIỆN THEO TIẾN ĐỘ

## WEEK 1

### 1.1 OOP

**Lập trình hướng đối tượng (OOP-Object-Oriented Programming)** là một phương pháp hay mô hình giúp tăng năng suất, đơn giản hóa việc bảo trì, dễ dàng mở rộng trong thiết kế phần mềm.

**Lập trình hướng đối tượng có 4 tính chất:**



Hình Mô hình OOP

**Kế thừa (Inheritance):**

Khi một đối tượng được truyền lại tất cả các thuộc tính và phương thức của đối tượng cha được gọi là kế thừa. Kế thừa giúp tái sử dụng lại mã nguồn. Nó được sử dụng cho đa hình lúc runtime.

**Đa hình (Polymorphism):**

Khi một nhiệm vụ được thực hiện bởi nhiều cách khác nhau, tính chất này được gọi là đa hình. Ví dụ có nhiều các để thuyết phục các khách hàng khác nhau, để vẽ một cái gì đó như hình tròn, hình chữ nhật, ...

Trong java, để áp dụng tính đa hình chúng ta sử dụng phương thức orverloading hoặc overriding

**Trừu tượng (Abstraction):**

Trừu tượng là sự ẩn đi những chi tiết bên trong và hiển thị ra các chức năng, tính chất này gọi là trừu tượng. Ví dụ: khi gọi điện thoại chúng ta không biết xử lý nội bộ thế nào, khi đi xe máy cũng vậy, mà chúng ta chỉ biết đến các chức năng thông qua giao tiếp bên ngoài.

Trong java, chúng ta áp dụng tính chất trừu tượng bằng cách sử dụng abstract class và interface

**Đóng gói (Encapsulation):**

Việc ràng buộc giữa code và data với nhau tạo thành một khối duy nhất được biết đến là đóng gói. Ví dụ: viên thuốc con nhộng được đóng gói với nhiều loại thuốc bên trong.

Một class trong java là một ví dụ về đóng gói. Java bean là một lớp được đóng gói hoàn toàn vì tất cả các dữ liệu thành viên là private.

**Thế mạnh của OOPs so với ngôn ngữ lập trình hướng thủ tục:**

1) Lập trình hướng đối tượng giúp việc phát triển và bảo trì dễ dàng hơn. Trong khi phương pháp lập trình hướng thủ tục là không dẽ dàng quản lý khi code lớn.

2) Lập trình hướng đối tượng có tính năng ẩn dấu thông tin, trong khi hướng thủ tục có thể truy cập dữ liệu toàn cục ở bất kỳ nơi nào

3) Lập trình hướng đối tượng cung cấp khả năng mô phỏng sự kiện thực tế hiệu quả hơn.

### 1.2 Core Java 1

**Tổng quan**

Ngôn ngữ lập trình Java được phát triển với tập đoàn Sun Microsystems với người sáng lập là James Gosling, được phát hành vào năm 1995

Ngôn ngữ lập trình Java:

* Hướng đối tượng
* Nền tảng độc lập
* Đa luồng
* Đối tượng:
  + State 🡪 Field
  + Behavior 🡪 Method
* Một lớp là một nhóm đối tượng có các thuộc tính chung. Nó là một mẫu hoặc thiết kế từ đó các đối tượng được tạo ra
* Interface: một bản thiết kế của một lớp. Nó chỉ có các phương thức trừu tượng. Interface là một kỹ thuật để thu được tính trừu tượng hoàn toàn và đa kế thừa trong Java. Interface trong Java cũng biễu diễn mối quan hệ IS-A. Nó không thể được khởi tạo giống như lớp trừu tượng.

**Lập trình hướng đối tượng có 4 tính chất:**

- Inheritance(thừa kế)

- Polymorphism(đa hình)

- Abstraction(trừu tượng)

- Encapsulation(đóng gói)

### 1.3 Git

**Git** là tên gọi của một **Hệ thống quản lý phiên bản phân tán** (*Distributed Version Control System –****DVCS***) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (**clone**) từ một kho chứa mã nguồn (**repository**), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (**commit**) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là **Working Tree**. Đại loại là như vậy.

[](https://thachpham.com/wp-content/uploads/2015/04/dvcs.png)

Hình Mô hình hoạt động của DVCS

Ngoài ra, có một cách hiểu khác về Git đơn giản hơn đó là nó sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó. Và một người khác có thể xem các thay đổi của bạn ở từng phiên bản,  họ cũng có thể đối chiếu các thay đổi của bạn rồi gộp phiên bản của bạn vào phiên bản của họ. Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một kho chứa mã nguồn.

Cơ chế lưu trữ phiên bản của Git là nó sẽ tạo ra một “*ảnh chụp*” (***snapshot***) trên mỗi tập tin và thư mục sau khi commit, từ đó nó có thể cho phép bạn tái sử dụng lại một ảnh chụp nào đó mà bạn có thể hiểu đó là một phiên bản. Đây cũng chính là lợi thế của Git so với các DVCS khác khi nó không “lưu cứng” dữ liệu mà sẽ lưu với dạng snapshot.

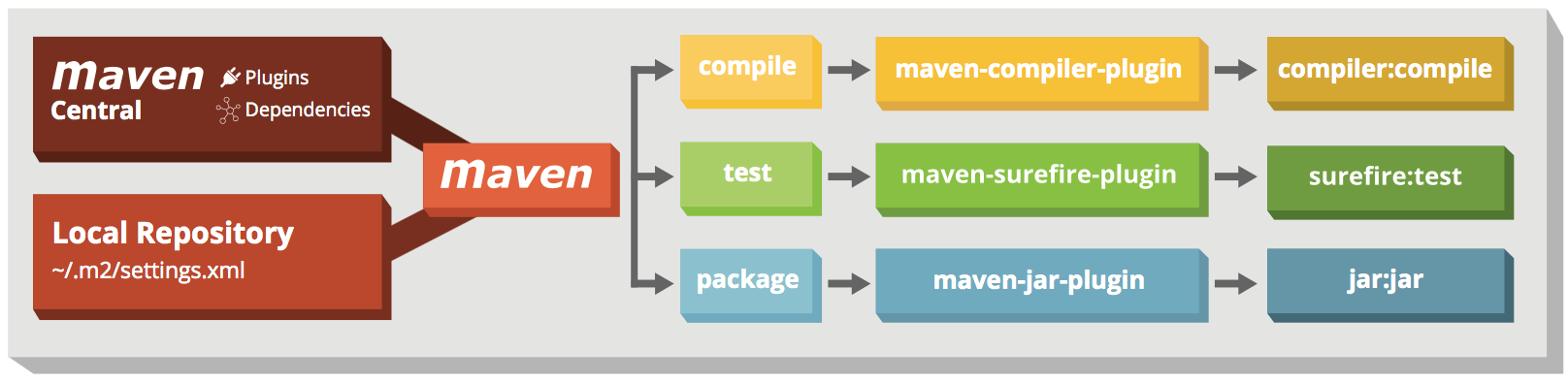
### Build with maven

Maven là một công cụ quản lý và thiết lập dự án.

* Quản lý thư viện cho cả dự án, quản lý version, bản phân phối
* Thiết lập tự động các quá trình validate, test, đóng gói, cài đặt, triển khai dự án…

Tóm lại nó đơn giản hóa và chuẩn hóa quá trình xây dựng một dự án. Nó xử lý biên dịch, phân phối, tài liệu, sự giao tiếp giữa các team phát triển thành 1 nhiệm vụ liền mạch.

Hơn hết, Maven được viết bằng ngôn ngữ Java và nó hoàn toàn free  😎

[](https://stackjava.com/wp-content/uploads/2019/06/maven-architecture.png)

Hình Maven

2. Mục tiêu chính của Maven

Mục tiêu chính của Maven là cung cấp cho nhà phát triển những điều sau:

* Cung cấp một mô hình toàn diện cho các dự án, giúp nó có thể tái sử dụng, bảo trì và dễ hiểu hơn.
* Cung cấp các plugins hoặc công cụ tương tác với mô hình khai báo này.

Maven cấu trúc dự án và nội dung được tuyên bố trong một tập tin pom.xml,  
gọi là mô hình đối tượng dự án (POM), là đơn vị cơ bản của  
toàn bộ hệ thống Maven. Trong các chương sau, chúng tôi sẽ giải thích chi tiết về POM.

3. Các tính năng của Maven

Để thực hiện được mục tiêu của mình, Maven có những tính năng sau:

* Thiết lập, cài đặt các dự án đơn giản phù hợp thực tiễn
* Sử dụng thống nhất trên tất cả các dự án + quản lý sự phụ thuộc và tự động update (Cái này hay dùng nhất: ví dụ có nhiều team, nhiều member cùng phát triển 1 project thì mọi người sẽ thống nhất qua file pom.xml về version, các thư viện… cần thêm thư viện nào, sửa version thì thực hiện hết trong đây, những người khác update không bị xung đột (impact)).
* Kho thư viện lớn và ngày càng mở rộng
* Khả năng mở rộng với việc dễ dàng viết các plugin trong Java hoặc các ngôn ngữ scripting
* Xây dụng dựa trên mô hình: Maven có thể xây dựng bất kì project nào thành các đầu ra được định nghĩa trước như war, jar, metadata.

Biến

Biến là tên của vùng nhớ được lưu trong bộ nhớ stack

**Có 3 kiểu biến trong java:** bao gồm biến cục bộ (hay còn gọi là biến local), biến toàn cục (biến instance) và biến tĩnh(biến static).

Kiểu dữ liệu

Kiểu dữ liệu nguyên thủy (primitive) và kiểu dữ liệu đối tượng

Bảng kiểu dữ liệu nguyên thủy như sau:

|  |  |
| --- | --- |
| Kiểu dữ liệu | Mô tả |
| byte | Dùng để lưu dữ liệu kiểu số nguyên có kích thước một byte (8 bít). Phạm vi biểu diễn giá trị từ -128 đến 127. Giá trị mặc định là 0. |
| char | Dùng để lưu dữ liệu kiểu kí tự hoặc số nguyên không âm có kích thước 2 byte (16 bít). Phạm vi biểu diễn giá trị từ 0 đến u\ffff. Giá trị mặc định là 0. |
| boolean | Dùng để lưu dữ liệu chỉ có hai trạng thái đúng hoặc sai (độ lớn chỉ có 1 bít). Phạm vi biểu diễn giá trị là {“True”, “False”}. Giá trị mặc định là False. |
| short | Dùng để lưu dữ liệu có kiểu số nguyên, kích cỡ 2 byte (16 bít). Phạm vi biểu diễn giá trị từ - 32768 đến 32767. Giá trị mặc định là 0. |
| int | Dùng để lưu dữ liệu có kiểu số nguyên, kích cỡ 4 byte (32 bít). Phạm vi biểu diễn giá trị từ -2,147,483,648 đến 2,147s,483,647. Giá trị mặc định là 0. |
| long | Dùng để lưu dữ liệu có kiểu số nguyên có kích thước lên đến 8 byte. Giá trị mặc định là 0L. |
| float | Dùng để lưu dữ liệu có kiểu số thực, kích cỡ 4 byte (32 bít). Giá trị mặc định là 0.0F. |
| double | Dùng để lưu dữ liệu có kiểu số thực có kích thước lên đến 8 byte. Giá trị mặc định là 0.00D |



*Kiểu dữ liệu đối tượng*

Trong java có 3 kiểu dữ liệu đối tượng:

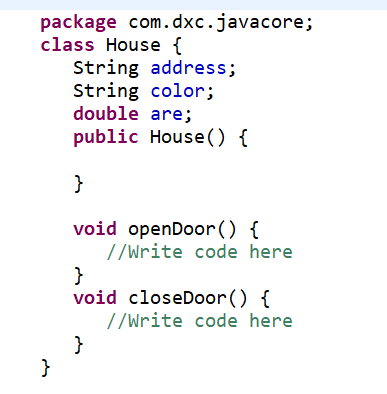
Table Kiểu đối tượng trong java

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Mô tả** |
| **Array** | Một mảng của các dữ liệu cùng kiểu. |
| **class** | Dữ liệu kiểu lớp đối tượng do người dùng định nghĩa. Chứa tập các thuộc tính và phương thức.. |
| **interface** | Dữ liệu kiểu lớp giao tiếp do người dùng định nghĩa. Chứa các phương thức của giao tiếp. |

## WEEK 2

Cấu trúc class cơ bản:

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Mô tả** |
| **Array** | Một mảng của các dữ liệu cùng kiểu. |
| **class** | Dữ liệu kiểu lớp đối tượng do người dùng định nghĩa. Chứa tập các thuộc tính và phương thức.. |
| **interface** | Dữ liệu kiểu lớp giao tiếp do người dùng định nghĩa. Chứa các phương thức của giao tiếp. |

****

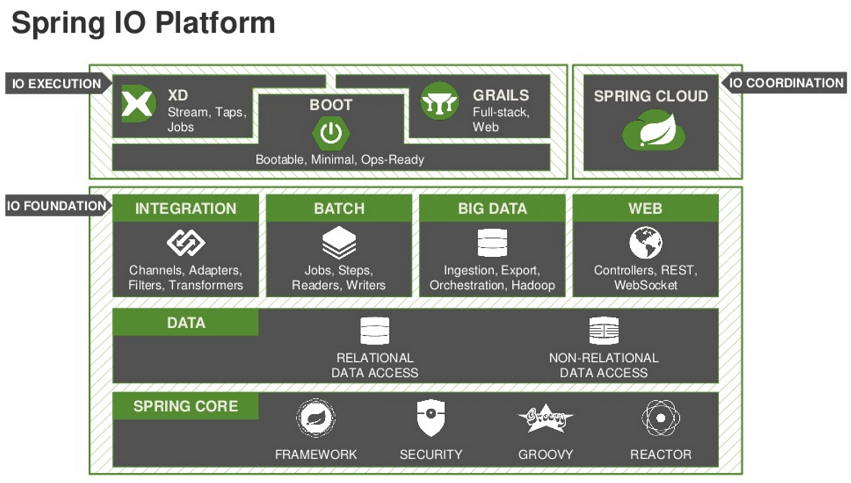
Packages(Gói)

Classes(Lớp)

Methods(Phương thức)

Constructors

Objects(Đối tượng)

2.1 SPRING BOOT 

Hình Spring IO Platform

**Spring Boot** là một dự án phát triển bởi **JAV**(ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.

Để phát triển một ứng dụng web cơ bản HelloWorld sử dụng Spring framework bạn sẽ cần ít nhất 5 công đoạn sau;

* Tạo một project sử dụng Maven với các dependency cần thiết của Spring MVC và Servlet API.
* Một tập tin web.xml để khai báo DispatcherServlet của Spring MVC.
* Một tập tin cấu hình của Spring MVC.
* Một class Controller trả về một trang “Hello World” khi có request đến.
* Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy.

Trong các công đoạn này, chỉ có công đoạn tạo một class Controller thì có thể khác cho các ứng dụng khác nhau vì mỗi ứng dụng có một yêu cầu khác nhau. Còn các công đoạn khác thì như nhau.

Giờ đây với Spring Boot, chúng ta có thể tạo dự án Spring một cách nhanh chóng và cấu hình cũng đơn giản **dùng Sublime Text** để phát triển luôn **khỏi cần cài đặt eclipse hay netbean** nặng bỏ bà.  
Dưới đây là một số tính năng nổi bật của Spring Boot:

* Tạo các ứng dụng Spring độc lập
* Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần phải deploy ra file WAR)
* Các starter dependency giúp việc cấu hình Maven đơn giản hơn
* Tự động cấu hình Spring khi cần thiết
* Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML …

### 2.2 Structured Query Language(SQL)

SQL là viết tắt của Structured Query Language có nghĩa là ngôn ngữ truy vấn có cấu trúc, là một ngôn ngữ máy tính để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ.

SQL là ngôn ngữ chuẩn cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres và SQL Server đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.

Ngoài ra, họ đang sử dụng các ngôn ngữ khác nhau, chẳng hạn như:

* MS SQL Server using T-SQL,
* Oracle using PL/SQL,
* MS Access version of SQL is called JET SQL (native format) etc.

Các câu lệnh SQL

Các lệnh SQL tiêu chuẩn để tương tác với cơ sở dữ liệu quan hệ là CREATE, SELECT, INSERT, UPDATE, DELETE và DROP. Các lệnh này có thể được phân thành các nhóm sau dựa trên bản chất của chúng

DDL - Ngôn ngữ định nghĩa dữ liệu(Data Definition Language)

Table Ngôn ngữ định nghĩa dữ liệu

|  |  |
| --- | --- |
| Lệnh | Mô tả |
| CREATE | Tạo ra một bảng mới, một view của một bảng, hoặc các đối tượng khác trong cơ sở dữ liệu. |
| ALTER | Sửa đổi một đối tượng cơ sở dữ liệu hiện có, chẳng hạn như một bảng. |
| DROP | Xoá toàn bộ một bảng, view của một bảng hoặc các đối tượng khác trong cơ sở dữ liệu. |

DML - Ngôn ngữ thao tác dữ liệu(Data Manipulation Language)

Table Ngôn ngữ thao tác dữ liệu

|  |  |
| --- | --- |
| Lệnh | Mô tả |
| SELECT | Lấy ra các bản ghi nhất định từ một hoặc nhiều bảng. |
| INSERT | Tạo một bản ghi.. |
| UPDATE | Chỉnh sủa bản nghi. |
| DELETE | Xóa bản ghi. |

DCL - Ngôn ngữ điều khiển dữ liệu (Data Control Language)

Table Ngôn ngữ điều khiển dữ liệu

|  |  |
| --- | --- |
| Lệnh | Mô tả |
| GRANT | Cung cấp một đặc quyền cho người dùng. |
| REVOKE | Lấy lại các đặc quyền được cấp từ người dùng. |

### 2.1 Packages

Package là một nhóm các kiểu tương tự của các lớp, giao diện và các package con.

**Package trong java** có thể được phân loại theo hai hình thức, package được dựng sẵn và package do người dùng định nghĩa.

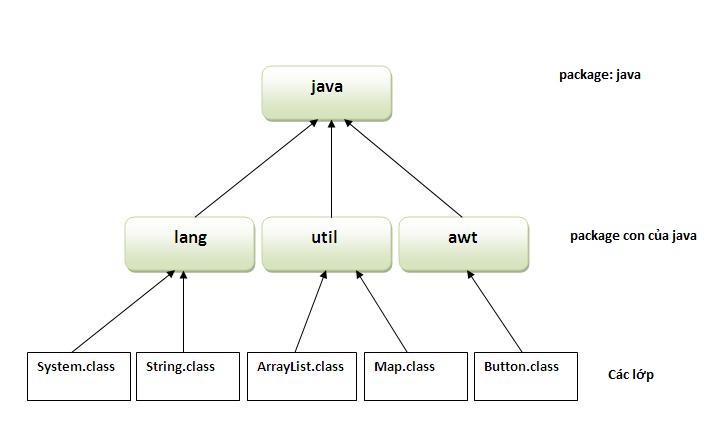
Có rất nhiều package được dựng sẵn như java, lang, AWT, javax, swing, net, io, util, sql, ...

Lợi thế của việc sử dụng package trong java

Package được sử dụng để phân loại lớp và interface giúp dễ dàng bảo trì.

Package cung cấp bảo vể truy cập

Package khắc phục được việc đặt trùng tên.



Hình Java packages

Cách đặt tên cho package:

* Tên package phải viết thường không được viết hoa
* Phải là danh từ

### 2.2 Classes

Modifier class *MyClass extends [MySuperClass implements YourInterface ]* {

// field, constructor, and

// method declarations

}

Các loại access modifiers: public, protected, default, private

Cách đặt tên lớp:

* Tên Class chỉ cho phép chứa các kí tự a đến z, A đên Z, 0 đến 9, và \_ (Underscore). Kí tự đặc biệt khác và space là không hợp lệ.
* Tên Class phải bắt đầu bằng kí tự chữ và dấu \_, không được bắt đầu bằng kí tự số
* Đội dài của tên class phải hợp lệ không quá dài cũng không quá ngắn, tên đúng với chức năng của class
* Tên class phải đặt bằng tiếng Anh phải là danh từ

### 2.3 Methods(Phương thức)

Xem xét ví dụ sau để giải thích cú pháp của một phương thức:

public static int tenPhuongThuc(int a, int b) {

// phan than phuong thuc

}

public static : Là modifier.

int: Kiểu trả về

tenPhuongThuc: Tên phương thức

a, b: Các tham số chính thức

int a, int b: Danh sách các tham số

Các phương thức cũng còn được biết như các Procedure (thủ tục) hoặc Function (hàm):

**Procedure:** Chúng không trả về bất kỳ giá trị nào.

**Function:** Chúng trả về giá trị.

Sự định nghĩa phương thức bao gồm một header và phần thân phương thức. Tương tự như sau:

modifier kieuTraVe tenPhuongThuc (Danh sach tham so) {

// Than phuong thuc

}

Cú pháp trên bao gồm:

**modifier:** Nó định nghĩa kiểu truy cập của phương thức và nó là tùy ý để sử dụng.

**kieuTraVe:** Phương thức có thể trả về một giá trị.

**tenPhuongThuc:** Đây là tên phương thức.

**Danh sach tham so:** Danh sách các tham số, nó là kiểu, thứ tự, và số tham số của một phương thức. Đây là tùy ý, phương thức có thể không chứa tham số nào.

**Than phuong thuc:** Phần thân phương thức định nghĩa những gì phương thức đó thực hiện với các lệnh.

* + 1. Constructors

Constructor trong java là một dạng đặc biệt của phương thức được sử dụng để khởi tạo các đối tượng.

Java Constructor được gọi tại thời điểm tạo đối tượng. Nó khởi tạo các giá trị để cung cấp dữ liệu cho các đối tượng, đó là lý do tại sao nó được gọi là constructor.

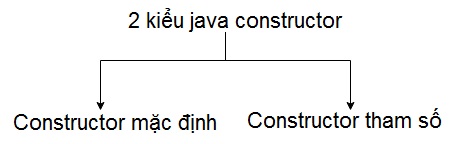
Các quy tắc tạo constructor trong java

Có 2 quy tắc cơ bản cho việc tạo constructor:

* Tên constructor phải giống tên lớp chứa nó.
* Constructor không có kiểu trả về tường minh.

**Các kiểu của java constructor:** có 2 kiểu của constructor

* Constructor mặc định (không có tham số truyền vào)
* Constructor tham số



Hình Constructor java

**Mục đích của constructor mặc định là gì?**

Constructor mặc định cung cấp các giá trị mặc định như 0, null, (tùy thuộc vào kiểu dữ liệu) ... tới đối tượng được khởi tạo.

Constructor tham số trong java

Một constructor có tham số truyền vào được gọi là constructor tham số.

Constructor tham số được sử dụng để cung cấp các giá trị khác nhau cho các đối tượng khác nhau.

**Sự khác nhau giữa constructor và phương thức trong java**

|  |  |
| --- | --- |
| **Constructor** | **Phương thức** |
| Constructor được sử dụng để khởi tạo trạng thái của một đối tượng. | Phương thức được sử dụng để thể hiện hành động của một đối tượng. |
| Constructor không có kiểu trả về. | Phương thức có kiểu trả về. |
| Constructor được gọi ngầm. | Phương thức được gọi tường minh. |
| Trình biên dịch Java tạo ra constructor mặc định nếu bạn không có constructor nào. | Phương thức không được tạo ra bởi trình biên dịch Java. |
| Tên của constructor phải giống tên lớp. | Tên phương thức có thể giống hoặc khác tên lớp. |

* + 1. Objects(Đối tượng)

 Đối tượng là thực thể mang tính vật lý cũng như mang tính logic, trong khi lớp chỉ là thực thể logic. Đối tượng có các trạng thái và các hành vi. Ví dụ: Một dog có trạng thái là color, name, breed (dòng dõi) và cũng có các hành vi: Wag (vẫy đuôi), bark (sủa), eat (ăn). Một đối tượng là một instance (ví dụ,trường hợp) của một lớp.

**Đối tượng trong Java**

Đó là một thực thể có trạng thái và hành vi, ví dụ như bàn, ghế, xe con, mèo, … Nó có thể mang tính vật lý hoặc logic. Ví dụ về logic đó là Banking system.

Một đối tượng có hai đặc trưng sau:

States(Trạng thái): biểu diễn dữ liệu (giá trị) của một đối tượng.

Behavior(Hành vi): biểu diễn hành vi (tính năng) của một đối tượng như gửi tiền vào, rút tiền ra, …

## WEEK 3

### 3.1 Cascading Style Sheets(CSS)

**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.

[](https://thachpham.com/wp-content/uploads/2015/04/html-css-webpage.png)

Hình HTML & CSS

*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.

Cấu trúc một đoạn CSS

[](https://thachpham.com/wp-content/uploads/2013/07/css_banner.jpg)

Hình Cấu trúc CSS

Một đoạn CSS bao gồm 4 phần như thế này:

|  |  |
| --- | --- |
| 01  02  03  04  05 | 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.

Về danh sách các thuộc tính của CSS bạn có thể xem qua [**CSS Reference của Mozilla**](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference).

### 3.2 Javascript

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... thậm chí các trình duyệt trên thiết bị di động cũng có hỗ trợ.

Nếu bạn đã biết qua về HTML5 thì bạn thấy có các khái niệm như sessionStore hay localStore, đây là hai đối tượng được tạo nên từ Javascript nên rõ ràng trong HTML5 cũng có sử dụng nó. Với những ứng dụng đó thì bạn thấy javascript không thể chết trong các ứng dụng website được.

Hay thậm chí có những ứng dụng Webgame người ta sử dụng javascript để xử lý các thao tác trên Client, nếu không có nó thì thông thường chọn Flash để xây dựng nhưng lại gặp vấn đề load chậm nên có một số người chọn Javascript để làm.

### 3.3 JSON



Hình JSON example

JSON Là Gì

JSON là một kiểu định dạng dữ liệu trong đó sử dụng văn bản thuần tuý, định dạng JSON sử dụng các cặp key - value để dữ liệu sử dụng .

Một ví dụ một tập tin có tên application\_info.json với nội dung như ở dưới đây sử dụng format kiểu JSON để lưu trữ thông tin của một ứng dụng phần mềm:

{

"name" : "PHPStorm",

"version" : "16.0.1",

"license" : "commercial"

}

Tập tin json có thể được lưu với bất kỳ phần mở rộng nào, tuy nhiên thông thường thì nó được lưu dưới phần mở rộng là .json hoặc .js.

JSON ban đầu được phát triển để dành phục vụ cho ứng dụng viết bằng JavaScript. Bản thân thuật ngữ JSON là viết tắt của cụm từ JavaScript Object Notation . Tuy nhiên vì JSON là một định dạng dữ liệu nên nó có thể được sử dụng bởi bất cứ ngôn ngữ nào àm không giới hạn với JavaScript.

Sử Dụng JSON

Một trường hợp khá phổ biến trong JavaScript mà dữ liệu được định dạng theo format JSON xuất hiện đó là trong các AJAX request. Ví dụ bạn lưu tập tin application\_info.json ở thư mục gốc của server (để khi truy cập vào URL http://localhost/application\_info.json thì server trả về nội dung của tập tin này) và sau đó bạn tạo một tập tin json\_ajax.html với nội dung như sau:

<html>

<head>

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

</head>

<body>

<script type="text/javascript">

$.ajax(

type: 'GET',

url: "http://localhost/application\_info.json",

success: **function**(response) {

console.log("Data type: " + (**typeof** response));

console.log("Application name: " + response.name);

},

);

</script>

</body>

</html>

Đoạn code trên sử dụng $.ajax() để gửi AJAX request lên server lấy về nội dung tập tin application\_info.json. Sau khi lấy về nội dung tập tin này thành công, dữ liệu sẽ được chuyển vào biến response. Nếu bạn mở developer console của trình duyệt lên (thông thường sử dụng phím F12) bạn sẽ thấy kiểu dữ liệu của biến response này được JavaScript object với các thuộc tính như name, version, license.

Sử Dụng JSON Với PHP

Giống như JavaSCript, PHP cũng hỗ trợ dữ liệu định dạng JSON.

**<?php**

$json\_string = <<<DOC

{

"name" : "PHPStorm",

"version" : "16.0.1",

"license" : "commercial"

}

DOC;

$arr = json\_decode($json\_string, **true**);

print\_r($arr);

Đoạn code trên sử dụng hàm json\_decode để decode dữ liệu được lưu trong biến $json\_string. Biến $json\_string lúc này có kiểu dữ liệu là String của PHP nhưng đồng thời chuỗi này cũng hỗ trợ định dạng JSON.

Kết quả trả về của đoạn code trên sẽ như sau:

**Array**

(

[name] => PHPStorm

[version] => 16.0.1

[license] => commercial

)

Tới đây chúng ta đã kết thúc việc tìm hiểu về kiểu format JSON và cách ứng dụng kiểu dữ liệu này trong hai ngôn ngữ là JavaScript và PHP.

## WEEK 4

### 4.1 Unit test là gì?

Trong kiểm thử phần mềm có 4 mức độ kiểm thử: Unit test ( kiểm thử mức đơn vị), Intergration test ( kiểm thử tích hợp), System test (kiểm thử hệ thống), Acceptance test (kiểm thử chấp nhận).

Unit test là mức độ kiểm thử nhỏ nhất trong quy trình kiểm thử phần mềm. Unit test kiểm thử các đơn vị nhỏ nhất trong mã nguồn như method, class, module...Do đó Unit test nhằm kiểm tra mã nguồn của các chương trình, các chức năng riêng rẽ hoạt động đúng hay không.

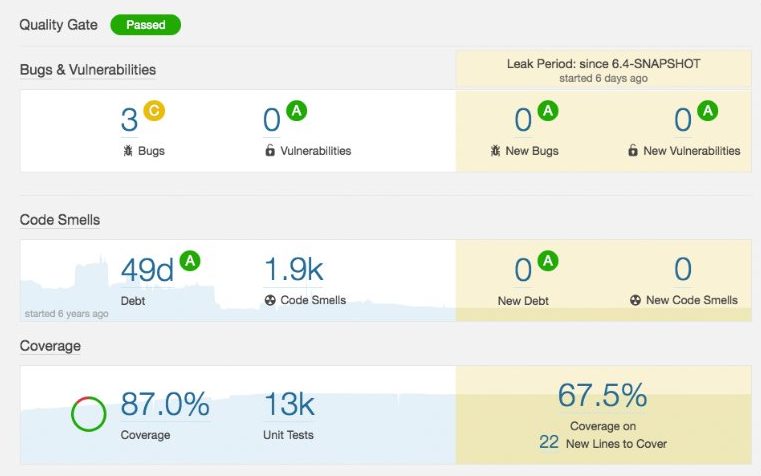
Unit testing được thực hiện bởi lập trình viên.

### 4.2 SonarLint/Qube and Rule Profiles

**SonarQube** là một nền tảng mã nguồn mở được phát triển bởi SonarSource để liên tục kiểm tra chất lượng code, review tự động với việc phân tích code để phát hiện lỗi, đoạn code không tốt, hoặc lỗ hổng bảo mật trên 20 ngôn ngữ lập trình.

Sonarqube hiện hỗ trợ các ngôn ngữ lập trình sau: Java (including Android), C#, PHP, JavaScript, C/C++, COBOL, PL/SQL, PL/I, ABAP, VB.NET, VB6, Python, RPG, Flex, Objective-C, Swift, Web and XML.

Nó được tích hợp vào công cụ phát triển Eclipse, Visual Studio, IntelliJ IDEA thông qua plugin SonarLint và tích hợp với các công cụ khác như LDAP, Active Directory, GitHub….



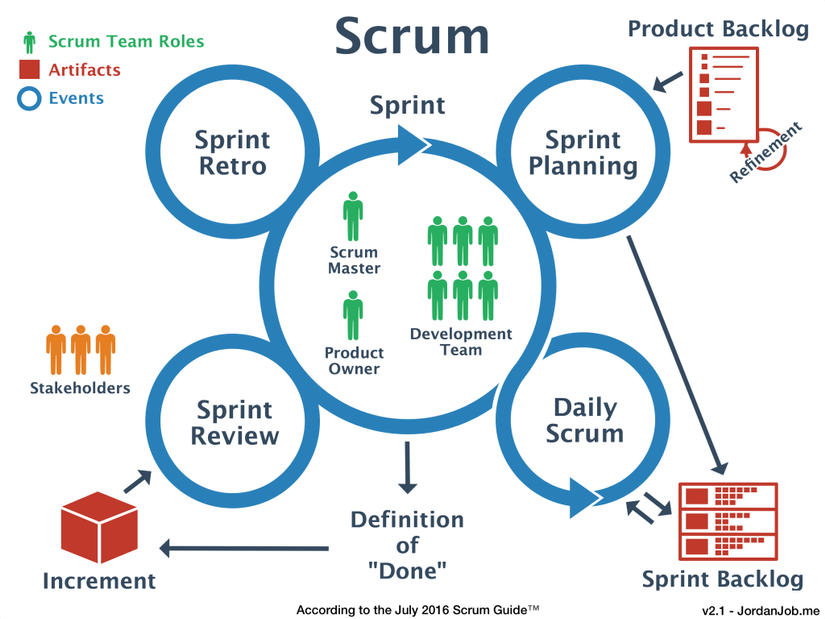
Hình Sonar Qube

### 4.3 Software Development Framework Using Scrum

Scrum là một Framework về quy trình và quản lý giúp giải quyết các vấn đề phức tạp, nhưng vẫn đảm bảo tính hiệu quả, sáng tạo và sản phẩm được tạo ra phải đạt được giá trị cao nhất. Bản thân Scrum là một Framework đơn giản nhằm giúp việc phối hợp hiệu quả nhất giữa các thành viên trong đội phát triển khi thực hiện những sản phẩm phức tạp. Với Scrum, sản phẩm được xây dựng trên 1 chuỗi các quy trình lặp lại (gọi là Sprint). Các sprint diễn ra đều đặn, mỗi một sprint là cơ hội để học hỏi điều chỉnh nhằm đạt được sự phù hợp và kết quả tốt nhất.

Nhắc đến Scrum là:

* Nhẹ nhàng
* Dễ hiểu
* Khó để quản lý và làm chủ



Hình Mô hình Scrum

Khi áp dụng Scrum, có 4 cuộc họp (Meetings or Ceremonies) quan trọng tạo nên cấu trúc trong mỗi Sprint như sau:

* **Sprint planning**: Cuộc họp lên kế hoạch của đội dự án, nhằm xác định những gì cần hoàn thành trong Spring sắp tới.
* **Daily stand-up**: Cũng được biết đến như “Daily Scrum”, một cuộc họp nhỏ 15 phút mỗi ngày để trao đổi công việc giữa đội phát triển.
* **Sprint demo**: Một cuộc họp chia sẻ, nơi mà các thành viên chỉ ra những gì họ đã làm được trong Sprint đó
* **Sprint retrospective**: Sự đánh giá, nhìn lại những điều đã làm được và chưa làm được của Sprint hiện tại, và đưa ra giải pháp hành động cho Sprint tiếp theo được tốt và hoàn thiện hơn.

Một nhóm phát triển Scrum thì sẽ có những thành phần hơi khác biệt so với mô hình truyền thống Waterfall, với 3 vai trò sau:

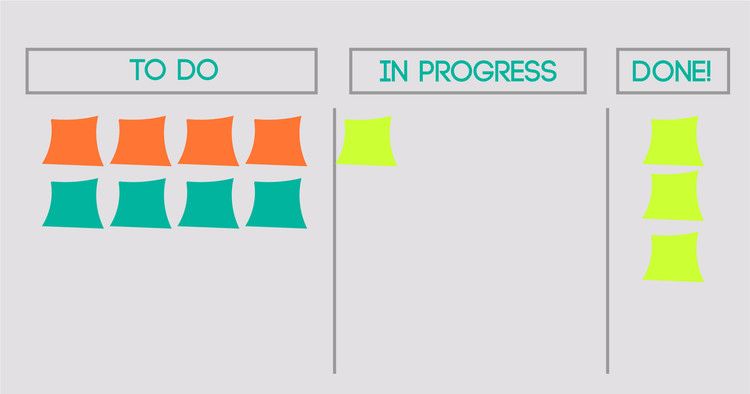
* Product Owner
* Scrum Master
* Development Team

Và bởi vì nhóm phát triển Scrum là đa chức năng, nên “Development Team” sẽ bao gồm Testers, Designers, và Ops Engineers.

**2. Tại sao nên dùng Scrum**

Nguồn gốc của Scrum xuất phát từ nghiên cứu những năm 1986 về quy trình phát triển sản phẩm thành công, tất cả đều có những đặc tính như yêu cầu mới và sự thay đổi yêu cầu thường xuyên và liên tục, thời gian để đưa sản phẩm ra thị trường rất ngắn. Trong nghiên cứu này, cách thức để đội phát triển làm việc với năng suất cao nhất được đưa ra so sánh với bộ môn bóng bầu dục và được gọi là Scrum. Tính minh bạch, kiểm tra, và thích nghi là 3 nền tảng cơ bản của Scrum. Và dưới đây là những lý do tại sao nên dùng Scrum.

* Scrum cho phép việc tự do triển khai
* Scrum dễ học và dễ sử dụng
* Scrum chấp nhận sự thay đổi
* Scrum làm giảm rủi ro khi xây dựng sản phẩm
* Scrum tối ưu hóa hiệu quả và nỗ lực của đội phát triển
* Scrum cho phép khách hàng sử dụng sản phẩm sớm hơn
* Scrum cải tiến liên tục

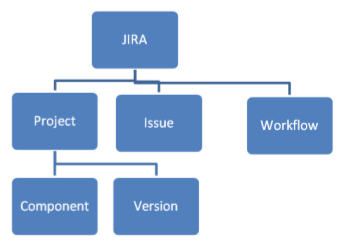


Hình To do list

## WEEK 5

### 5.1 Jira Agile

JIRA dựa trên 3 khái niệm:



Hình Mô hình Jira

1. Issue: Mọi task, lỗi, yêu cầu nâng cao; về cơ bản mọi thứ được tạo ra và theo dõi qua JIRA được coi là một vấn đề.
2. Dự án: tập hợp các issues
3. Luồng công việc: Luồng công việc đơn giản là một loạt các bước mà một vấn đề trải qua bắt đầu từ khi tạo thành đến khi hoàn thành.

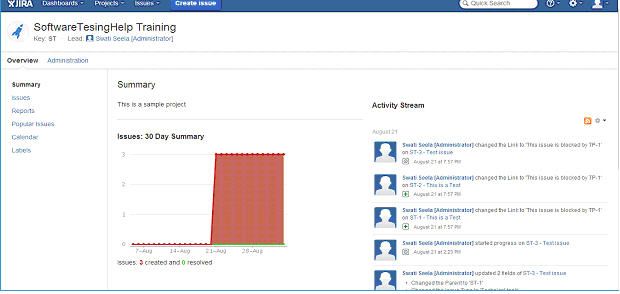
Luồng công việc từ khi issue được tạo , được làm và hoàn thành là :



Hình Luồng công việc Jira

**Giới thiệu giao diện JIRA**

Vì là bản dùng thử nên sau khi tải bản dùng thử, các bạn đăng ký account và đăng nhập = account đó, sau khi đăng nhập, bảng điều khiển được hiển thị cho người dùng.Trang bảng điều khiển cung cấp một ảnh chụp nhanh về mô tả của dự án bạn thuộc về; tóm tắt vấn đề và luồng hoạt động (các vấn đề được relate cho bạn, các vấn đề mà bạn đã tạo, v.v.).



Hình Giao diện Jira

Khi bạn đã tạo 1 danh sách các dự án , Để xem lại các dự án đã được tạo bạn chỉ cần vào chọn tên dự án từ danh sách thả xuống “Projects”. Một dự án là một tập hợp các issues. Mục số 6 trong danh sách phía trên - tính năng cho phép nhóm các vấn đề được hoàn thành với khái niệm này. Các dự án có các thành phần và phiên bản theo nó. Các thành phần không là gì ngoài các nhóm con trong một dự án dựa trên cơ sở chung. Ngoài ra, đối với cùng một dự án, các phiên bản khác nhau có thể được theo dõi từ JIRA.

**Mỗi dự án đều có các thuộc tính chính sau**:

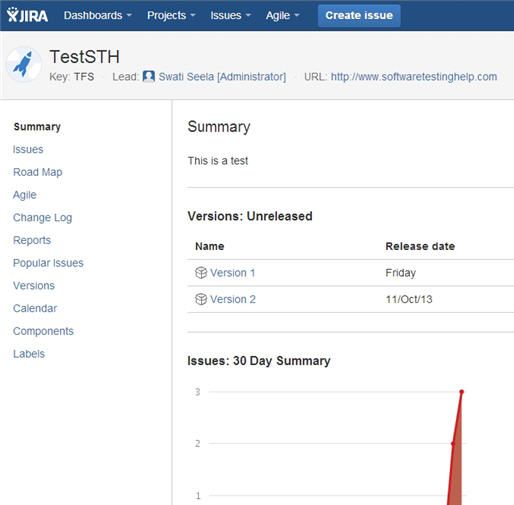
1. Tên - được chọn bởi quản trị viên.
2. Key-Nó là một định danh mà tất cả các tên issues theo dự án sẽ bắt đầu . Giá trị này được đặt trong quá trình tạo dự án và không thể sửa đổi sau đó ngay cả bởi quản trị viên.
3. Các thành phần
4. Phiên bản Ví dụ, lấy một ứng dụng dựa trên web; có 10 yêu cầu cần được phát triển. Sẽ có thêm 5 tính năng được thêm vào sau này. Bạn có thể chọn tạo dự án là “Thử nghiệm cho STH” phiên bản 1 và Phiên bản 2. Phiên bản 1 với 10 yêu cầu, phiên bản 2 với 5 yêu cầu mới.

Đối với phiên bản 1 nếu 5 yêu cầu thuộc về Module 1 và phần còn lại thuộc về module 2. Module 1 và module 2 có thể được tạo thành các đơn vị riêng biệt

**Ghi chú**: Tạo và quản lý dự án trong JIRA là một nhiệm vụ quản trị. Vì vậy,sẽ không bao gồm việc tạo dự án và sẽ tiếp tục thảo luận bằng cách sử dụng một dự án đã được tạo.

Lấy các chi tiết trong ví dụ trên, tôi đã tạo một dự án trong JIRA được gọi là "Kiểm tra cho STH", khóa là "TFS". Vì vậy, nếu tôi tạo ra một vấn đề mới, bộ định danh vấn đề sẽ bắt đầu với TFS và sẽ là "TSH-01". Phần này sẽ được giải thích rõ trong bài viết sau.

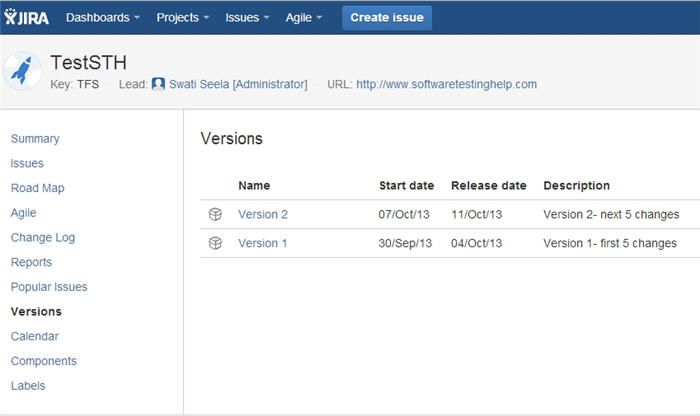
Sau đây là cách các chi tiết Dự án được hiển thị trong JIRA trong phần Sumary:



Hình Giao diện Jira

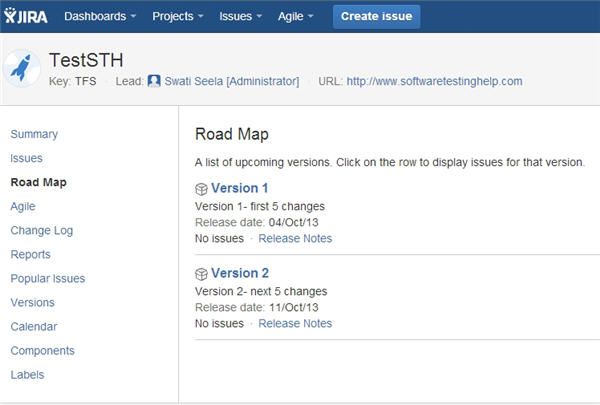
Xin lưu ý phần menu bên trái.

Khi tôi chọn tùy chọn "Components", nó sẽ hiển thị hai thành phần trong dự án:



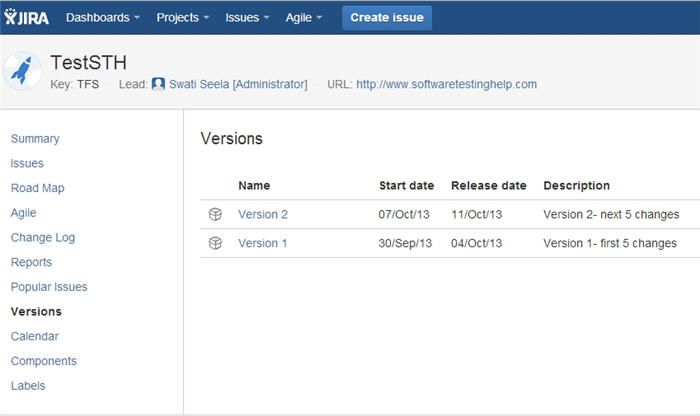
Hình Version Jira

Khi tôi chọn tùy chọn Version, các phiên bản trong dự án được hiển thị

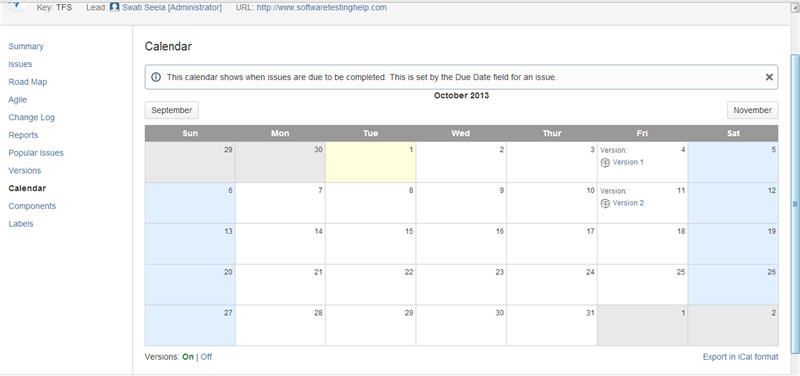


Hình Jira version

Chọn tùy chọn 'Roadmap', thông tin phiên bản được hiển thị cùng với các ngày đưa ra ý tưởng chung về các mốc quan trọng trong dự án.



Chọn tùy chọn 'Calendar ' để xem các ngày trong giai đoạn quan trọng nào đó:



Hình Jira calendar

Tại thời điểm này, không có issue nào được tạo cho dự án này. Nếu có, bạn sẽ có thể xem tất cả chúng bằng cách chọn "Issues" từ menu điều hướng bên trái.

### 5.2 Web Services (REST & SOAP)

**SOAP** là một giao thức truy cập Webservice được sử dụng trong một thời gian và hưởng tất cả các lợi ích của việc sử dụng lâu dai. Được phát triển bởi Micrisoft, SOAP không thực sự đơn giản như từ viết tắt của nó. Trong khi đó, **REST** là một chiến binh mới, tìm kiếm để sửa các vấn đề mà SOAP gặp phải và cung cấp các phương thức thực sự đơn giản để truy cập Webservice. Tuy nhiên, đôi khi SOAP thực sự dễ sử dụng hơn, đôi khi REST có vấn đề của riêng nó. Cả hai kỹ thuật đều có vấn đề cần xem xét khi quyết định sử dụng phương thức nào.

Cả **SOAP** và **REST** đêù có những điểm tương đồng trên giao thức HTTP, **SOAP** là một tổ hợp các pattern truyền tin nghiêm ngặt hơn **REST**. Quy định trong **SOAP** là rất quan trọng bởi vì nếu thiếu chúng, bạn không thể đạt được bất kì level tiêu chuẩn hóa nào. **REST** mang một kiến trức không yêu cầu quy trình và linh hoạt hơn. Cả SOAP và REST đều dựa trên những quy định đã tồn tại trong một thời gian dài mà mọi người đều đồng ý tuân theo do lợi ích mà nó mang lại

9.1.1 Tổng quan ngắn gọn về SOAP

SOAP dựa hoàn toàn vào XML để cũng cấp các services truyền tin. Microsoft ban đầu phát triển SOAP để thay thế cho các công nghệ cũ hơn không hoạt động tốt trên Internet như Distributed Component Object Model (DCOM) and Common Object Request Broker Architecture (CORBA). Những công nghệ này không thành công vì chúng dựa vào truyền tin nhị phân, cách truyền tin XML mà SOAP sử dụng làm việc tốt hơn qua Internet.

XML được sử dụng để tạo ra các requests và nhận responses trong SOAP, nó có thể trở nên cực kỳ phức tạp. Trong một số ngôn ngữ lập trình, bạn cần phải xây dựng các requests đó bằng tay, điều này trở nên mơ hồ vì SOAP không dung nạp được các lỗi. Tuy nhiên, một vài ngôn ngữ khác có thể sử dụng các phím tắt mà SOAP cung cấp; Có thể giúp bạn giảm bớt effort cần thiết để tạo request và phân tích responses. Thực tế, khi làm việc với .NET, bạn thậm chí chưa bao giờ gặp đến XML.

Một trong những tính năng quan trọng nhất của SOAP là tích hợp trong xử lý lỗi. Nếu có sự cố với requests của bạn, responses chứa thông tin lỗi mà bạn có thể sử dụng để khắc phục sự cố. Bạn có thể không sở hữu các Web service, tính năng này đặc biệt vô cùng quan trọng; Nếu không bạn sẽ thắc mắc tại sao mọi việc lại không thành công. Các báo cáo lỗi thậm chí cung cấp mã code chuẩn để có thể tự động hoá một số nhiệm vụ xử lý lỗi trong mã của bạn.

Một tính năng khác khá thú vị của SOAP là bạn không nhất thiết phải sử dụng nó với giao thức HyperText Transfer Protocol (HTTP). Có một chỉ định rõ ràng để sử dụng SOAP qua Simple Mail Transfer Protocol (SMTP) và không có lý do gì bạn không thể sử dụng nó trên các phương thức khác.

9.1.2 Tổng quan ngắn gọn về REST

Nhiều developers thấy rằng SOAP cồng kềnh và khó sử dụng. Ví dụ, làm việc với SOAP trong JavaScript có nghĩa là viết một tấn các dòng code để thực hiện các task cực kỳ đơn giản bởi vì bạn phải tạo cấu trúc XML mỗi lần.

REST cung cấp giải pháp thay thế nhẹ hơn. Thay vì sử dụng XML để tạo request, REST dựa vào một URL đơn giản. Trong một số trường hợp, bạn phải cung cấp thông tin bổ sung theo những cách đặc biệt, nhưng hầu hết các Web service sử dụng REST đều dựa hoàn toàn vào việc thu lại các thông tin cần thiết bằng phương pháp URL. REST có thể sử dụng bốn hình thái HTTP 1.1 khác nhau (GET, POST, PUT, và DELETE) để thực hiện các tasks.

Không giống như SOAP, REST không phải sử dụng XML để cung cấp response. Bạn có thể tìm các Web service dựa trên REST đưa ra dữ liệu trong Command Separated Value (CSV), JavaScript Object Notation (JSON) và Really Simple Syndication (RSS). Vấn đề là bạn có thểthu được output bạn cần trong một form dễ phân tích trong ngôn ngữ bạn cần cho ứng dụng của bạn.

9.1.3 Lựa chọn SOAP hay REST

Trước khi bạn dành hàng giờ đồng hồ để lựa chọn giữa SOAP và REST, hãy xem xét rằng một số Web service hỗ trợ một và một số khác. Trừ khi bạn có kế hoạch tạo ra Web service của riêng bạn, quyết định sử dụng giao thức nào đã được hình thành bởi bạn. Rất ít Web service, chẳng hạn như Amazon, hỗ trợ cả hai. Quyết định của bạn thường tập trung vào Web service nào phù hợp nhất với nhu cầu của bạn chứ không phải là sử dụng giao thức nào.

9.1.4 So sánh giữa SOAP và REST

SOAP chắc chắn là lựa chọn nặng ký để truy cập Web service. Nó cung cấp những ưu điểm sau đây khi so sánh với REST:

* Ngôn ngữ, nền tảng và phương tiện độc lập (REST yêu cầu sử dụng HTTP)
* Hoạt động tốt trong môi trường doanh nghiệp (REST giao tiếp thẳng trực tiếp)
* Tiêu chuẩn hoá
* Cung cấp khả năng mở rộng đáng kể trước khi xây dựng dưới dạng các tiêu chuẩn WS \*
* Tích hợp xử lý lỗi
* Tự động hóa khi sử dụng với một số sản phẩm ngôn ngữ

Bên cạnh đó, REST dễ sử dụng hơn và linh hoạt hơn. Nó có những lợi thế sau khi so sánh với SOAP:

* Không có tools đắt tiền nào yêu cầu tương tác với Web service
* Smaller learning curve
* Hiệu quả (SOAP sử dụng XML cho tất cả các truyền tin, REST có thể sử dụng định dạng truyền tin ngắn gọn hơn)
* Nhanh (không yêu cầu xử lý rộng rãi) Gần gũi hơn với các công nghệ Web khác trong triết lý design.

10: KẾT LUẬNQua đợt thực tập này em đã hoàn thành tương đối tốt các công việc được giao. Tuy bước đầu còn gặp nhiều khó khăn, bỡ ngỡ nhưng nhờ sự chỉ bảo, hướng dẫn tận tình của các anh đã giúp em học hỏi được nhiều thứ và rút ra được những bài học cho riêng mình. Về kiến thức em đã củng cố được rõ thêm rất nhiều về Entity Framework, OOP cùng với đó hiểu sâu hơn về mô hình MVC, các hướng tiếp cận EF approaches… Về kỹ năng thực hành em đã có tiến bộ rất nhiều về kỹ năng giao tiếp, mạnh dạn đưa ra câu hỏi khi gặp khó khăn. Đặc biệt là kỹ năng làm việc nhóm được cải thiện rõ rệt bao gồm kỹ năng quản lý, kỹ năng lên kế hoạch và kỹ năng lắng nghe đóng góp ý kiến. Cùng với đó là những kinh nghiệm thực tiễn mà em tích lũy được khi trải nghiệm thực tế như tác phong làm việc ,sự chuyên nghiệp,tính kỷ luật,tinh thần trách nhiệm và đặc biệt các mối quan hệ xã hội được cải thiện và mở rộng.

### 5.1 Hypertext Markup Language(HTML)

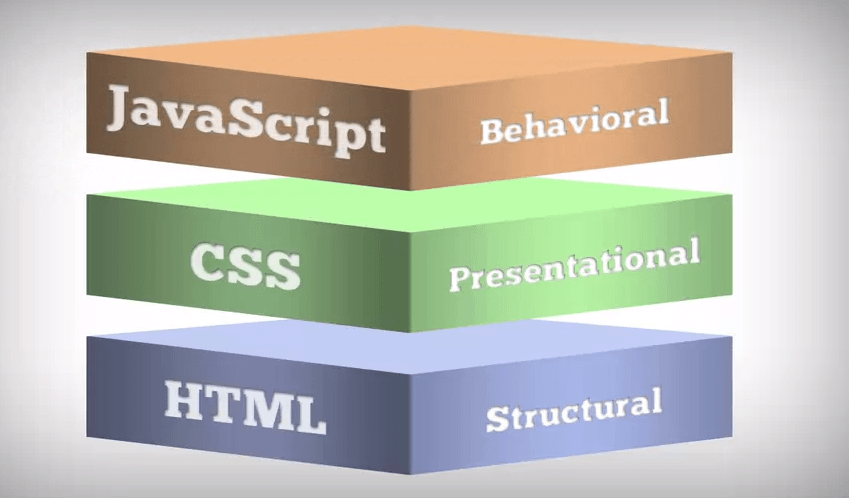
HTML là chữ viết tắt của cụm từ **H**yper**T**ext **M**arkup **L**anguage1) (dịch là Ngôn ngữ đánh dấu [siêu văn bản](http://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n)) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (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**.

**HTML đóng vai trò gì trong website?**

Như mình đã nói, 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.

[](https://thachpham.com/wp-content/uploads/2015/04/html-la-gi.png)

Hình Javascript

*Vai trò của từng ngôn ngữ trong website*

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 Thachpham.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).
* **PHP** – 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).
* **MySQL** – 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 bạn 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.

Nhân tiện đây mình cũng nói luôn, website có hai loại chính:

* **Website tĩnh** (static web) – Là một website không giao tiếp với máy chủ web để gửi nhận dữ liệu mà chỉ có các dữ liệu được khai báo sẵn bằng HTML và trình duyệt đọc.
* **Website động** (dynamic web) – Là một website sẽ giao tiếp với một máy chủ để gửi nhận dữ liệu, các dữ liệu đó sẽ gửi ra ngoài cho người dùng bằng văn bản HTML và trình duyệt sẽ hiển thị nó. Để một website có thể giao tiếp với máy chủ web thì sẽ dùng một số ngôn ngữ lập trình dạng server-side như PHP, ASP.NET, Ruby,..để thực hiện. Ví dụ như một website làm bằng WordPress là website động.

### 5.3 Bootstrap

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 tao 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 reponsive của bạn dễ dàng hơn và nhanh chóng hơn.  
Lịch sử Bootstrap  
Bootstrap là dược phát triển bởi Mark Otto và Jacob Thornton tại Twitter. Nó được xuất bản như là một mã nguồn mở vào tháng 8 năm 2011 trên GitHub. Tính ra đến thời điểm mình viết bài viết này nó cũng đã phát triển được 3 năm rồi. Bản bootstrap mới nhất bây giờ là bootstrap 4.



Hình Bootstrap

**-**  **Ưu điểm của Bootstrap**

Những điểm thuận lợi khi bản sử dụng bootstrap:

Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cẩn có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.  
Tính năng Responsive: Bootstrap’s xây dựng sẵn reponsive 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 hãm với IE phiên bản cũ vì thế việc IE9 hay IE8 đổ xuống không support là chuyện bình thường. Các bạn có dùng nên luy 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.

Bootstrap, ban đầu có tên là Twitter Blueprint, được phát triển bởi Mark Otto và Jacob Thornton tại Twitter như một khuôn khổ để khuyến khích sự nhất quán trên các công cụ nội bộ. Trước khi Bootstrap, các thư viện khác nhau được sử dụng để phát triển giao diện, dẫn đến sự thiếu nhất quán và gánh nặng bảo trì cao.

Sau một vài tháng phát triển bởi một nhóm nhỏ, nhiều nhà phát triển tại Twitter đã bắt đầu đóng góp cho dự án như một phần của Tuần lễ Hack, một tuần theo phong cách hackathon dành cho nhóm phát triển Twitter. Nó được đổi tên từ Twitter Blueprint thành Bootstrap, và được phát hành như một dự án nguồn mở vào ngày 19 tháng 8 năm 2011. Nó tiếp tục được duy trì bởi Mark Otto, Jacob Thornton và một nhóm nhỏ các nhà phát triển cốt lõi, cũng như một cộng đồng lớn người dùng.

## WEEK 6

## WEEK 7

## WEEK 8

## WEEK 9

## WEEK 10

WEEK 11

# CHƯƠNG 3. NHẬN XÉT, ĐÁNH GIÁ QUÁ TRÌNH THỰC TẬP

## 3.1. Các nhận xét, đánh giá thực trạng của quá trình làm việc.

Trong quá trình thực tập tuy bước đầu có nhiều bỡ ngỡ nhưng hoàn thành tương tối công việc được giao, nhờ sự chỉ dẫn tận tình của các anh nên em đã học hỏi được nhiều thứ từ đó rút ra kinh nghiệm cho mình. Về kiến thức em đã hiểu rõ hơn về mô hình MVC, Angular JS. kỹ năng làm việc nhóm được cải thiện rõ rệt bao gồm kỹ năng quản lý, kỹ năng lên kế hoạch và kỹ năng lắng nghe đóng góp ý kiến.