

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN.....	2
1.1. Giới thiệu về công ty.....	2
1.1.1. Các cơ sở.....	3
1.1.2. Trung tâm.....	3
1.1.3. Lĩnh vực hoạt động	4
1.2. Chức năng, cơ cấu, nhân sự	4
1.2.1. Các lĩnh vực thực tập	4
1.2.2. Cơ cấu tổ chức	4
1.2.3. Dịch vụ hỗ trợ	5
1.2.4. Các công nghệ chính.....	5
1.2.5. Quy Trình Sản Xuất Phần Mềm	5
CHƯƠNG 2. KẾ HOẠCH KIẾN TẬP.....	7
2.1. Nhiệm vụ đang thực hiện tại công ty	7
2.2. Kế hoạch, thời gian thực tập	7
CHƯƠNG 3. NỘI DUNG TÌM HIỂU.....	11
3.1. Nội dung tìm hiểu các vấn đề liên quan trong quá trình thực tập.....	11
3.1.1. Quy trình làm web với trong công ty với các tài liệu và nội dung...	11
3.1.2. Các ngôn ngữ lập trình web như C# Asp.net MVC, JAVA	13
3.1.3. ASP.NET ENTITY FRAMEWORK.....	15
3.1.4. Lập trình hướng đối tượng trong ASP.NET MVC.....	18
3.1.5. Lập trình Front End với HTML5, CSS, JAVASCRIPT, JQUERY.....	23

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu về công ty

TMA Solutions (gọi tắt TMA) là một doanh nghiệp tư nhân tại Việt Nam, kinh doanh các dịch vụ liên quan đến phát triển phần mềm.

Vào tháng 3 năm 1997, bà Bùi Ngọc Anh thành lập công ty TMA với 6 kỹ sư tại phòng khách nhà bà. Vào thời điểm đó, một công ty công nghệ thông tin ở Canada có ý thuê nhà bà Ngọc Anh làm chi nhánh cho công ty, đồng thời cũng muốn nhờ bà hoàn thành giúp các thủ tục pháp lý để mở chi nhánh cho mình. Nhưng cuối cùng, vì lý do tài chính nên công ty này đã hủy hợp đồng, thế là bà Ngọc Anh, cùng với vốn kiến thức đã thu thập được trong thời gian qua đã cho ra đời công ty TMA.

Đến năm 1998, TMA đón nhận khách hàng đầu tiên từ Mỹ và Canada. Vào năm này, số lượng thành viên của TMA đã tăng lên gấp ba lần, tức 18 người.

Năm 1999, con số 18 người này lại một lần nữa tăng lên gấp 3, TMA đạt được mức 54 nhân viên, và dời trụ sở sang quận Phú Nhuận. Hiện nay, trụ sở này vẫn là trụ sở chính của công ty.

Vào năm 2000, TMA có thêm khách hàng mới từ Úc, Singapore, Ấn Độ và Nhật Bản

Năm 2001, TMA có thêm trụ sở mới, cũng ở quận Phú Nhuận. Đồng thời cũng có thêm khách hàng từ Nhật Bản.

Năm 2003, TMA thành lập trung tâm nghiên cứu và phát triển R&D. Số lượng nhân viên lúc này cũng đạt được 200 người.

Năm 2005, TMA thành lập trụ sở ở nước ngoài đầu tiên tại Canada. [cần dẫn nguồn] Xét về mặt các trụ sở trong nước, cùng với trụ sở tại đường Đặng Văn Ngữ (quận Phú Nhuận) được thành lập vào năm 2004, đến năm 2005, TMA lại mở thêm

một trụ sở mới trên đường Trần Hữu Trang (quận Phú Nhuận), đẩy số lượng trụ sở lên 5, góp phần mở rộng quy mô công ty hơn.

Năm 2006-2008, TMA thành lập thêm 3 chi nhánh mới ở Nhật bản, Mỹ và ở châu Âu. Vào lúc này, TMA bước đầu thâm nhập thị trường châu Âu với các khách hàng từ Đức, Pháp, Đan Mạch.

Trong khoảng năm 2009-2010, TMA hoàn thành thêm trụ sở thứ sáu của mình tại công viên phần mềm Quang Trung. Đồng thời cho ra đời Trung tâm Giải pháp Di Động TMA (TMA Mobile Solutions). Thêm vào đó, TMA mở thêm một chi nhánh mới tại Úc. Vào năm 2009, TMA cũng có mặt tại triển lãm CommunicAsia2009 ở Singapore. Ngày 29/7/2010, Trung tâm Nghiên cứu & Phát triển CNTT-TT (ICT R&D Center - iRDC) của TMA được thành lập nhằm phục vụ hợp tác phát triển công nghệ và sản phẩm mới (iRDC) Năm 2011, TMA thành lập Tech Lab và Trung tâm Thực tập Sinh viên (SDC) nhằm đào tạo và nâng cao trình độ các thể hệ sinh viên, đặc biệt là sinh viên CNTT. Vào năm này, TMA cũng thành lập Bảo tàng và Thư viện sách cũ tại Lab6 (công viên phần mềm Quang Trung)

Năm 2012, TMA tham gia triển lãm CommunicAsia2012 tại Singapore. Cùng năm, TMA cũng tham gia triển lãm CNTT CeBIT tại Hannover, Đức . Lúc này, số lượng nhân viên của TMA là 1200 người.

Năm 2013, số lượng nhân viên tại TMA là 1400 người.

Năm 2015, với các dự án không ngừng tăng lên, số lượng kỹ sư CNTT đang làm việc tại TMA là 1900 người.

1.1.1. Các cơ sở

- Việt Nam: Thành phố Hồ Chí Minh
- Canada: Ottawa
- Hoa Kỳ: Vermont
- Châu Âu: Galway, Ireland
- Úc: Melbourne

1.1.2. Trung tâm

- Phòng Trung tâm Đào Tạo TMA
- Phòng Trung tâm Phát triển Sinh Viên
- Phòng Trung tâm Nghiên cứu và Phát triển (R&D)

1.1.3. Lĩnh vực hoạt động

- Tích hợp hệ thống.
- Xuất khẩu phần mềm.
- Giải pháp phần mềm.
- Cung cấp các giải pháp, dịch vụ viễn thông
- Đào tạo sinh viên và nhân viên về kiến thức phần mềm và kỹ năng mềm.

1.2. Chức năng, cơ cấu, nhân sự

1.2.1. Các lĩnh vực thực tập

Chương trình thực tập của sinh viên tại TMA Solutions, Student Development Center (SDC) là cơ hội để sinh viên ngành CNTT có thể tiếp cận, trải nghiệm quy trình làm việc theo dự án và hiểu rõ hơn về môi trường làm việc tại TMA.

Nhằm tạo điều kiện cho tất cả sinh viên ngành CNTT có thể tham gia chương trình, hằng năm TMA tổ chức 3 đợt thực tập chính vào tháng 3, tháng 6 và tháng 9 với những chuyên ngành: Business Applications, Big Data, Automation Testing và Telecommunication & Networking.

Trong quá trình thực tập, sinh viên sẽ được tham gia vào một nhóm dự án thực tế để làm việc chung với các lập trình viên khác và đồng thời được hướng dẫn tận tình từ các trưởng dự án. Đây là cơ hội để các bạn áp dụng những kiến thức đã được học tại trường vào thực tế, đồng thời tìm hiểu thêm những công nghệ mới trong từng dự án.

Bên cạnh đó, trong thời gian thực tập tại TMA, sinh viên sẽ có cơ hội tham gia các khóa đào tạo về kỹ năng và chuyên môn để trau dồi kiến thức và tự hoàn thiện bản thân.

1.2.2. Cơ cấu tổ chức

- **Đội ngũ nhân sự**

- + Design: DC1, DC3, DC4, Phạm Tuấn Anh...
- + Quản lý: Lê Hồng Việt, Nguyễn Đăng Khoa...
- + CSKH: Võ Thị Nhung...
- + Kỹ thuật: DS1, DS3...
- + Lập trình: TC1, DC13, DC2, TC3, TC23, DEP...

1.2.3. Dịch vụ hỗ trợ

- Software Development
- TMA software testing
- TMA it managed services
- TMA production support
- TMA it managed services
- TMA visual design services
- User Experience/ User Interface Application Design, Web, Mobile - Tablet, Desktop Design

1.2.4. Các công nghệ chính

- Java
- C++
- .NET
- Mobile AI PHP/Ruby
- Cloud Embedded & IoT Telecom/Network
- Big data & analytics

1.2.5. Quy Trình Sản Xuất Phần Mềm

12/12

1. Quản lý mã nguồn (source code) có được áp dụng trong các dự án/ sản phẩm phần mềm trong công ty?

2. Đội ngũ chuyên viên phần mềm có thể tạo bản build cho các sản phẩm chỉ trong vòng 1 bước?
3. Các dự án có áp dụng tạo bản build hàng ngày vào giờ cố định?
4. Trong quá trình phát triển phần mềm, những lỗi sản phẩm được tìm thấy có được lưu giữ lại trong cơ sở dữ liệu về lỗi?
5. Trong dự án, đội ngũ chuyên viên phần mềm sẽ thường sửa lỗi trước khi viết code mới?
6. Các công việc trong dự án có được quản lý rõ ràng bằng cách phân loại theo mức độ ưu tiên, với số giờ ước lượng cần có để hoàn thành, cũng như số giờ còn lại để thực hiện...?
7. Các yêu cầu nghiệp vụ có được tài liệu hoá rõ ràng trong Đặc tả yêu cầu (spec)?
8. Đội ngũ nhân viên lập trình thường được làm việc trong môi trường yên tĩnh?
8. Các dự án có sự hỗ trợ đặc lực từ các công cụ phần mềm thương mại?
9. Các dự án/ sản phẩm có nhân viên kiểm tra chất lượng phần mềm (tester)?
10. Các ứng viên kỹ sư CNTT khi phỏng vấn vào công ty có phải kiểm tra viết code?
11. Các dự án có áp dụng kiểm thử khả dụng Hallway (hallway usability testing)?

CHƯƠNG 2. KẾ HOẠCH KIẾN TẬP

2.1. Nhiệm vụ đang thực hiện tại công ty

Thực hiện xây dựng website internship manager với asp.net mvc với sự hướng dẫn của leader Phương Võ thuộc phòng TC1

2.2. Kế hoạch, thời gian thực tập

TT	Thời gian	Công Việc
1	7/5	<ul style="list-style-type: none">– Tìm hiểu về các khái niệm (scrum master, leader, sprint, backlog) trong quá trình xây dựng một dự án thực tế– Tìm hiểu quy trình, và các tiến trình xây dựng dự án của công ty– Tìm hiểu trang web slack (để trao đổi nhóm khi làm dự án chung) , trello (quản lý và trao đổi với manager dự án)
2	8/5	<ul style="list-style-type: none">– Xây dựng nhóm với dự án InternShip Manager– Tìm hiểu về yêu cầu, chức năng, nhiệm vụ của dự án– Xây dựng User stories cho dự án (sprint 0)
3	9/5	<ul style="list-style-type: none">– Tiếp tục thực hiện xây dựng và hoàn thành sprint 0
4	10/5	<ul style="list-style-type: none">– Tiến hành xây dựng và thiết kế cơ sở dữ liệu cho web InternShip Manager với hệ quản trị cơ sở dữ liệu SQL.
5	11/5	<ul style="list-style-type: none">– Hoàn thành cơ sở dữ liệu InternShip Manager
6	14/5	<ul style="list-style-type: none">– Tiến hành thiết kế giao diện cho web InternShip Manager với 3 quyền chính– Thực hiện Sprint(1) với các chức năng<ul style="list-style-type: none">+ Login via account

		<ul style="list-style-type: none"> + Login via facebook + Login via Google + Project View with account is manager + Detail Project manager with account is manager
7	15/5	<p>– Tiến hành hoàn thành sprint(1) của dự án</p> <p>Kết quả:</p> <ul style="list-style-type: none"> + Trang Login <div data-bbox="493 602 1386 1203"> </div> <ul style="list-style-type: none"> + Trang View Project:

		<div><div>InternLogo</div><div>Project ManagerIntern Member ManagerInventory Manager</div></div> <div><div>Total number project: 4</div><div>Project Completed: 1</div><div>Project Processing: 3</div></div> <div><div>List Project</div><div><div>Add Project</div><div>Search Project</div></div><table><tr><th>Project Name</th><th>Leader</th><th>Start date</th><th>End date</th><th>Status</th><th></th></tr><tr><td>Intern-KHMH</td><td>abc</td><td>04/05/18 12:00:00 AM</td><td></td><td>Completed</td><td></td></tr><tr><td>Intern_Hhdl</td><td>abc</td><td>03/05/18 12:00:00 AM</td><td></td><td>Processing</td><td></td></tr><tr><td>Intern-Idd</td><td>abc</td><td>02/04/18 12:00:00 AM</td><td></td><td>Completed</td><td></td></tr></table><div>+ Trang Detail Project:</div><div><div>Project Name: Intern-KHMH</div><div>Team Leader: abc</div><div><div>Startdate : 4/5/2018</div><div>Enddate : undefined</div></div><div><div>Member:</div><div><div>Phạm Văn Khánh</div><div>Phạm Đức Minh</div><div>nguyen huy</div><div>hoc huy</div></div></div><div><div>Feature Member</div><table><tr><th>Feature</th><th>Member</th><th>Duration</th><th>Status</th><th></th></tr><tr><td>Manage Project</td><td>Phạm Văn Khánh</td><td>0</td><td>Processing</td><td></td></tr><tr><td>View Project</td><td>Phạm Văn Khánh</td><td>0</td><td>Completed</td><td></td></tr></table></div></div></div>	Project Name	Leader	Start date	End date	Status		Intern-KHMH	abc	04/05/18 12:00:00 AM		Completed		Intern_Hhdl	abc	03/05/18 12:00:00 AM		Processing		Intern-Idd	abc	02/04/18 12:00:00 AM		Completed		Feature	Member	Duration	Status		Manage Project	Phạm Văn Khánh	0	Processing		View Project	Phạm Văn Khánh	0	Completed	
Project Name	Leader	Start date	End date	Status																																					
Intern-KHMH	abc	04/05/18 12:00:00 AM		Completed																																					
Intern_Hhdl	abc	03/05/18 12:00:00 AM		Processing																																					
Intern-Idd	abc	02/04/18 12:00:00 AM		Completed																																					
Feature	Member	Duration	Status																																						
Manage Project	Phạm Văn Khánh	0	Processing																																						
View Project	Phạm Văn Khánh	0	Completed																																						
8	16/5	<div><div>– Hoàn thành các chức năng trên trang View project như : Add project, View Detail project khi click vào các Project trên list project trên trang view project, Remove Project, Edit Project</div></div>																																							
9	17/5	<div><div>– Hợp dự án với Manager Phương Võ</div><div>– Phân tích một số lỗi sai trong quá trình xây dựng dự án trên</div><div>– Tiếp tục hoàn thiện và cải thiện lỗi trên Sprint(1)</div></div>																																							

10	18/5	<ul style="list-style-type: none"> – Hoàn thành sửa lỗi Sprint(1) và tiếp tục xây dựng sprint(2) gồm các chức năng – Intern manager – Inventory Manager
----	------	--

CHƯƠNG 3. NỘI DUNG TÌM HIỂU

3.1. Nội dung tìm hiểu các vấn đề liên quan trong quá trình thực tập

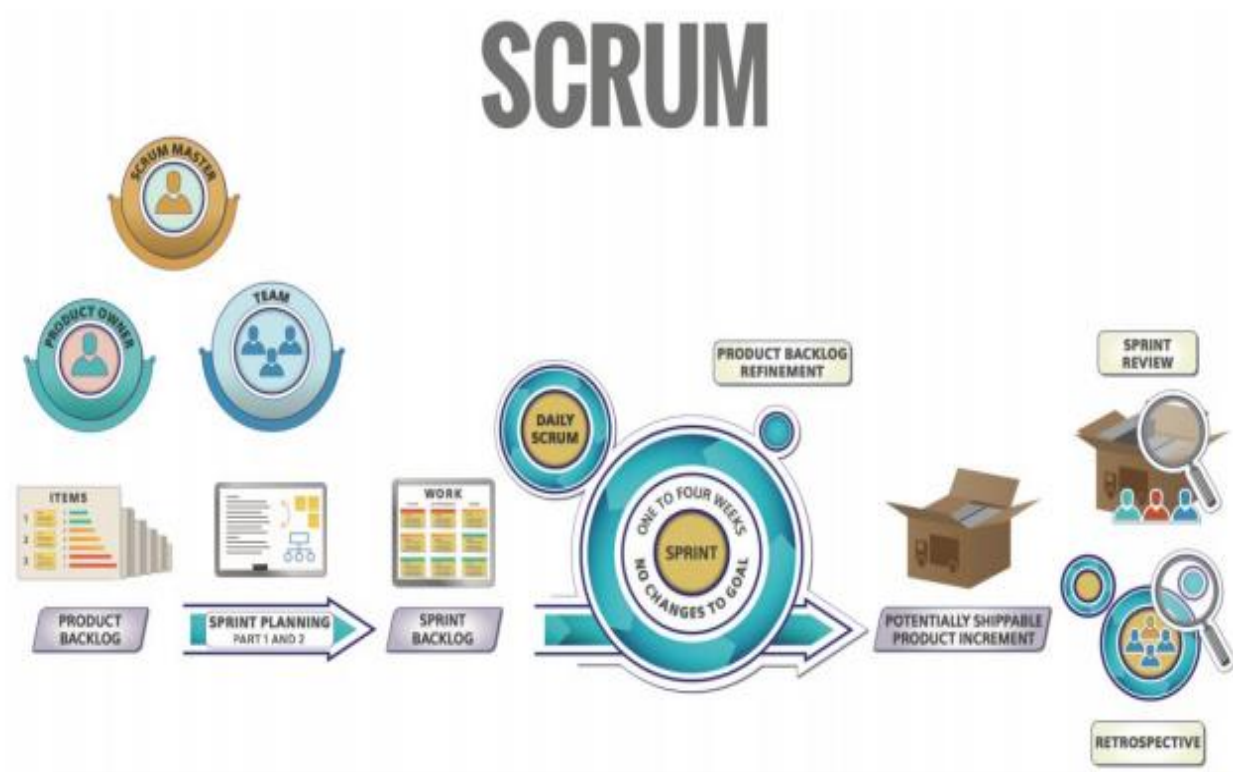
3.1.1. Quy trình làm web với trong công ty với các tài liệu và nội dung

– The Scrum Primer

+ Khái niệm về Scrum.

Scrum là một khung phát triển trong đó các nhóm liên-chức năng phát triển các sản phẩm hoặc dự án theo hình thức lặp và tăng trưởng. Scrum tổ chức quá trình phát triển thành các chu trình làm việc gọi là Sprint. Mỗi chu trình không kéo dài quá bốn tuần (phổ biến nhất là hai tuần) và diễn ra liên tiếp nhau mà không bị gián đoạn. Các Sprint được đóng khung thời gian, tức là chúng kết thúc vào một ngày nhất định cho dù công việc đã được hoàn thành hết hay chưa và không được phép kéo dài thêm. Thông thường thì các Nhóm Scrum¹ chọn một độ dài Sprint và sử dụng nó cho đến khi họ có thể cải tiến và sử dụng chu trình ngắn hơn. Bắt đầu mỗi Sprint, một Nhóm liên-chức năng được gọi là Nhóm Phát triển 2 (Development Team, bao gồm khoảng bảy người 3) lựa chọn các hạng mục (yêu cầu khách hàng) từ một danh sách ưu tiên. Nhóm Phát triển thống nhất một mục tiêu chung mà họ tin rằng có thể chuyển giao được vào cuối Sprint, một thứ gì đó phải thấy được và thực sự được “hoàn thành”. Trong suốt Sprint, không có hạng mục mới nào được thêm vào Scrum để dành các thay đổi cho Sprint tiếp theo, còn đối với Sprint ngắn hiện tại thì cố gắng tập trung vào một mục tiêu nhỏ, rõ ràng và tương đối ổn định. Hằng ngày Nhóm Phát triển họp với nhau trong một khoảng thời gian ngắn nhằm thanh tra tiến độ và điều chỉnh các bước cần thiết tiếp theo để hoàn thành công việc còn lại. Vào cuối Sprint, Nhóm Phát triển rà soát Sprint với các bên liên quan và trình bày phần mình đã xây dựng được. Mọi người nhận được các phản hồi và có thể đưa vào trong Sprint tiếp theo. Scrum nhấn mạnh vào sản phẩm chạy tốt ở cuối Sprint mà thực sự đã được “hoàn thành”; trong


trường hợp của phần mềm, điều này có nghĩa là một hệ thống đã được tích hợp, thực hiện kiểm thử đầy đủ, viết tài liệu cho người dùng cuối, và có khả năng chuyển giao



Hình 3.1 Tổng quan về Scrum

+ Các Vai trò trong Scrum

- ✚ Product Owner
- ✚ Nhóm Phát triển
- ✚ ScrumMaster
- ✚ Product Backlog
- ✚ Detailed appropriately
- ✚ Estimated
- ✚ Emergent
- ✚ Định nghĩa Hoàn thành
- ✚ Lập kế hoạch Sprint
- ✚ Scrum Hằng ngày

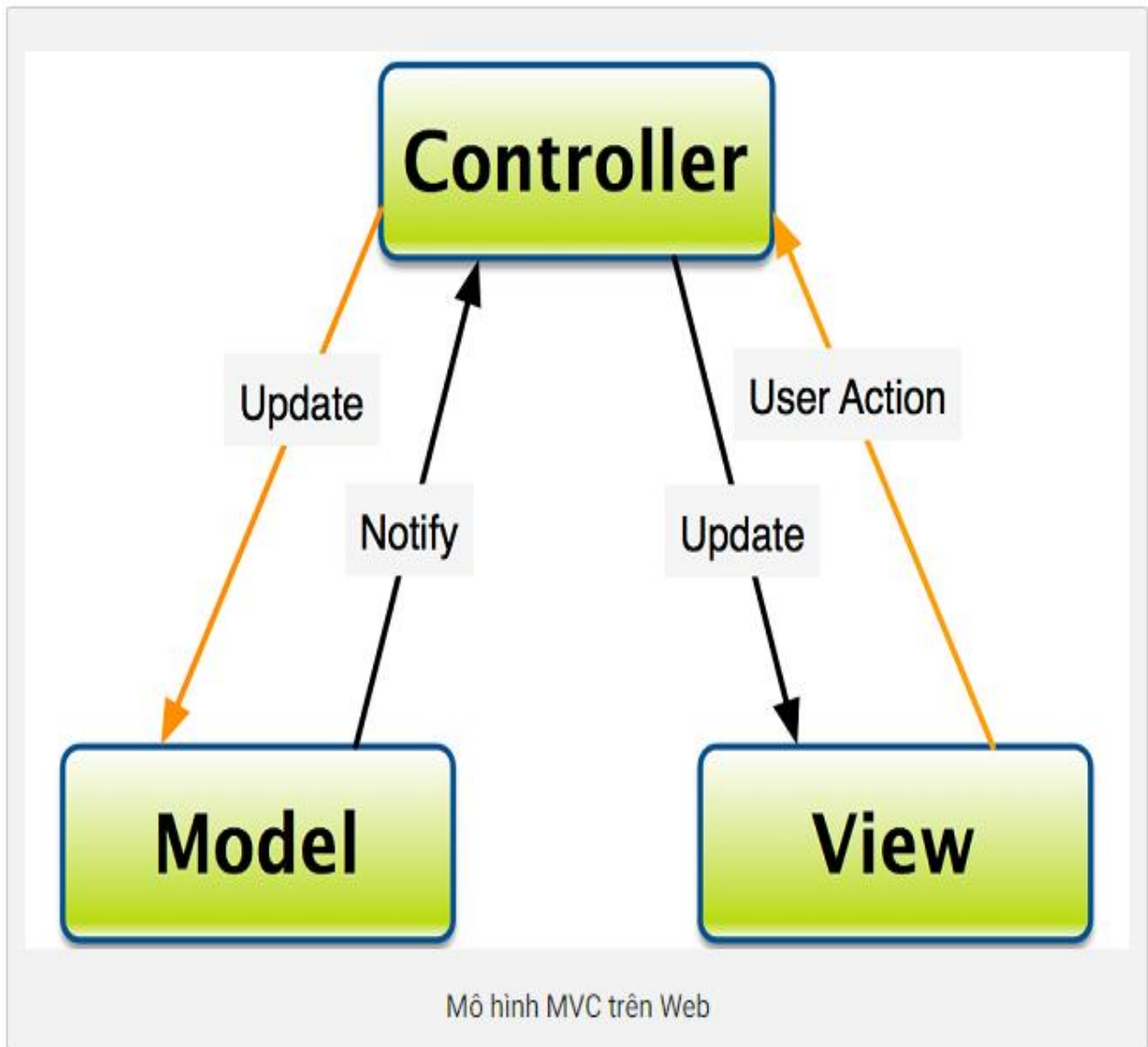
 Theo dõi tiến độ trong suốt Sprint

(Chi tiết xem trên The Scrum Primer)

3.1.2. Các ngôn ngữ lập trình web như C# Asp.net MVC, JAVA

– Mô hình MVC

- + M là Model: cấu trúc dữ liệu theo cách tin cậy và chuẩn bị dữ liệu theo lệnh của controller
- + V là View: Hiển thị dữ liệu cho người dùng theo cách dễ hiểu dựa trên hành động của người dùng
- + C là Controller: Nhận lệnh từ người dùng, gửi lệnh đến cho Model để cập nhập dữ liệu, truyền lệnh đến View để cập nhập giao diện hiển thị.



Hình 3.2 Mô hình MVC

- **Quy trình xử lý chung của web hiện đại thường như sau:**
 - + Người dùng gọi yêu cầu xử lý tại trang chủ.
 - + “Controller” nhận yêu cầu này và đưa lệnh xử lý yêu cầu đó. Các lệnh thực thi với phần “View” thì cập nhập hoặc phục vụ yêu cầu trang web, với “Model” thì để trình diễn logic. Ta giả sử lệnh yêu cầu có yếu tố logic.
 - + “Model” thực thi phần logic được lấy từ cơ sở dữ liệu và gửi trả lại phản hồi dựa trên hướng dẫn của “Controller”.

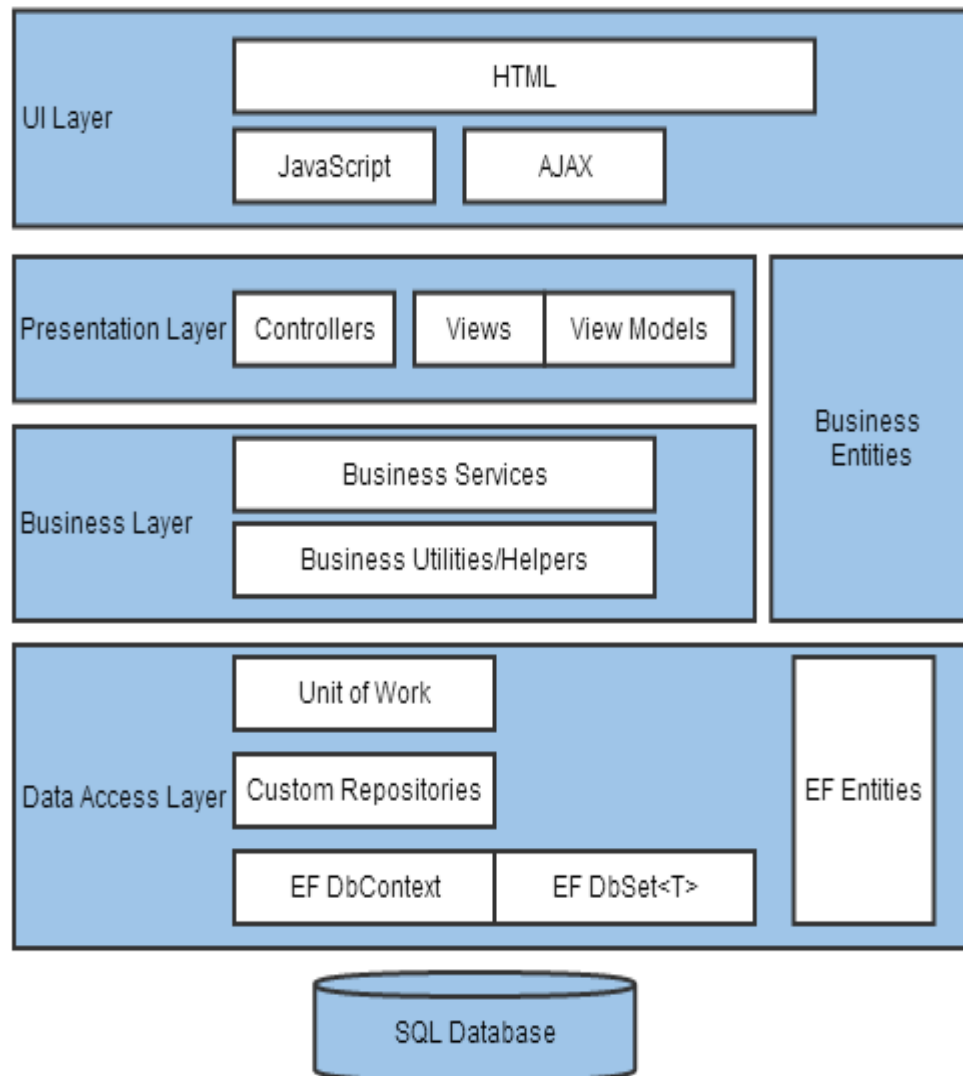
- + “Controller” truyền dữ liệu ra phần “View”, cập nhật giao diện hiển thị cho người dùng.

3.1.3. ASP.NET ENTITY FRAMEWORK

Entity Framework (EF) là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. EF cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của EF là giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. EF được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy EF là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đồng đạo của các nhà phát triển Web.

Vị trí của Entity Framework trong mô hình lập trình và phát triển Web
Chắc hẳn, bạn đã biết về mô hình Web 3 tầng (n tầng) khi đọc sách, giáo trình, bài giảng về nội dung lập trình Web. Tuy nhiên, vấn đề bạn cần phải hiểu rõ là: việc định rõ số lượng các tầng và các mối liên hệ giữa các tầng trong mô hình phát triển Web cũng rất đa dạng, tùy theo cách hiểu các lập trình viên và dự án Web. Do đó, việc hiểu Entity Framework nằm ở đâu trong mô hình Web 3 tầng cũng không hoàn toàn dễ dàng. Trong phần này, tôi có tham khảo từ Stephen M. Redd nhằm giúp các bạn hiểu rõ bản chất vấn đề hơn.

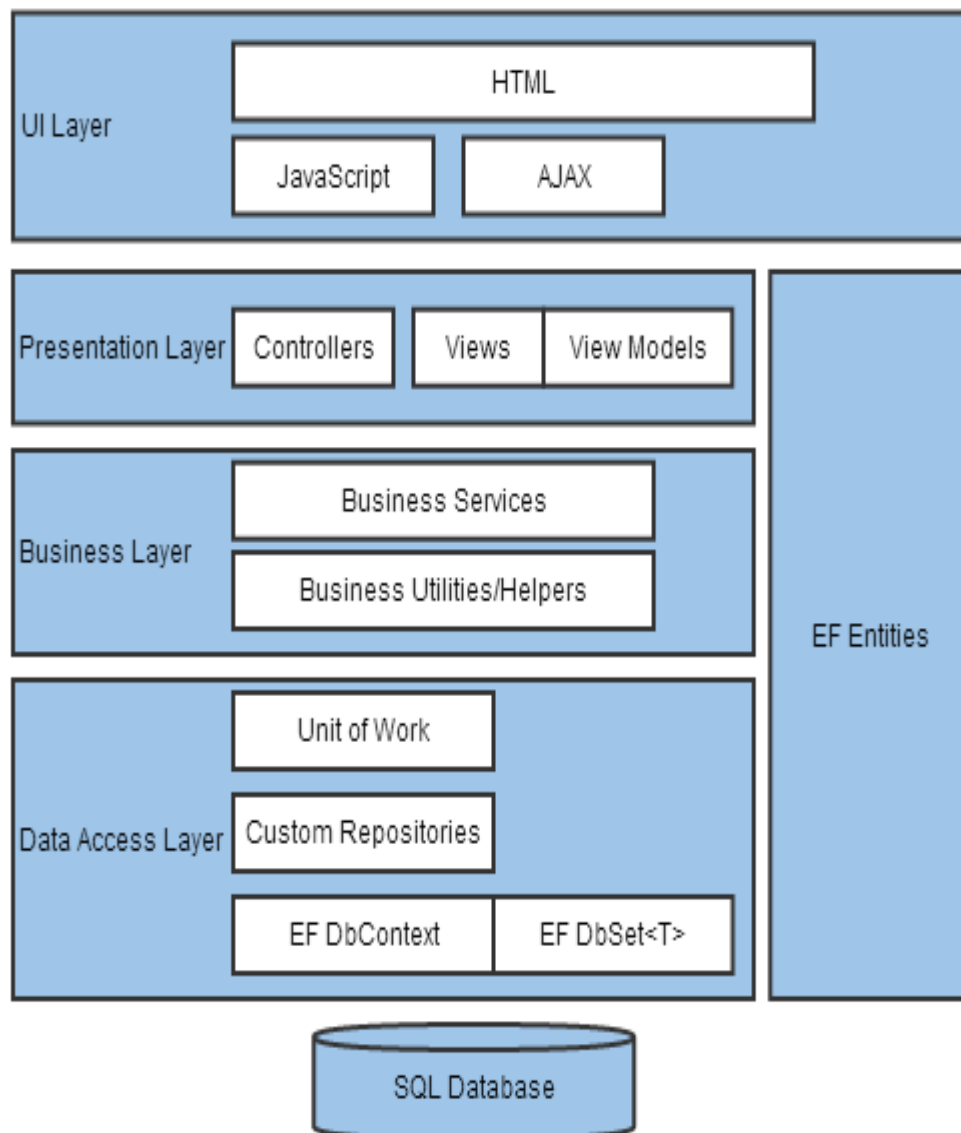
Đầu tiên, chúng ta có vị trí của EF trong mô hình dự án Web ASP.NET MVC.



Hình 3.3 Vị trí của EF trong web (1)

Trong mô hình trên, chúng ta thấy vị trí của EF nằm trọn trong tầng Data Access Layer, tức là EF đóng vai trò là nơi trung gian để tương tác dữ liệu từ database với các tầng cao hơn, chẳng hạn như Business Layer. Hay nói cách khác, EF đóng vai trò như là phiên bản mới của ADO.NET. Tuy nhiên vị trí của EF trong mô hình này

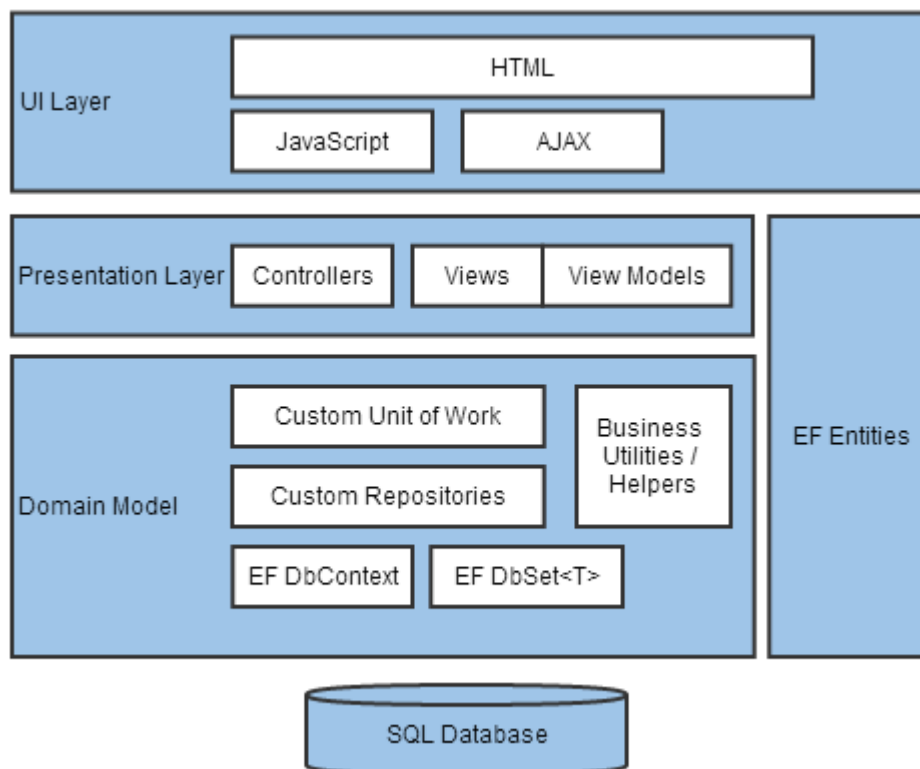
chỉ có trên lý thuyết. Thật sự, khi bắt tay xây dựng nhiều dự án Web, vị trí thường



Hình 3.4 Vị trí của EF trong web (2)

Trong hình trên, EF dường như có vị trí rất tự do, xuất hiện ở cả 3 tầng Data Access Layer, Business Layer, và cả Presentation Layer. Các bạn có thể hình dung EF như là 1 thư viện, hễ tầng nào cần thì chỉ cần gọi đến. Mô hình này xuất phát có thể là sự cầu thả của lập trình viên khi không thích tuân theo tiêu chuẩn quan hệ giữa các tầng hoặc cách giải thích khác là sự lỏng lẻo giúp cho việc lập trình thuận tiện hơn. Trong vài mô hình người ta gom tầng Data Access Layer và

Business Layer chỉ làm tầng Data Model, và vị trí EF cũng bao hàm tầng Data



Hình 3.5 vị trí của EF trong web (3)

Đến đây, bạn có thể hiểu 1 cách chung nhất EF có vị trí trung gian, đóng vai trò kết nối giữa cơ sở dữ liệu và các thành phần khác của 1 dự án Web khi cần đến. Ngoài ra, có nhiều cách hiểu về vị trí của EF ở đâu trong mô hình Web, bạn sẽ dần khám phá để đưa ra cách hiểu và định nghĩa riêng trong quá trình thiết kế và xây dựng dự án Web ASP.NET.

3.1.4. Lập trình hướng đối tượng trong ASP.NET MVC

Lập trình hướng đối tượng(OOP) là một trong những kỹ thuật lập trình rất quan trọng hiện nay. Nó được áp dụng ở hầu hết các ứng dụng thực tế xây dựng tại các doanh nghiệp. Hầu hết các ngôn ngữ lập trình và framework lập trình phổ biến hiện nay như Java, PHP, .NET, ruby đều hỗ trợ lập trình hướng đối tượng. Các lập trình viên đa phần đã được học về lập trình hướng đối tượng ở trường đại học

nhưng các nguyên lý cơ bản của lập trình hướng đối tượng đôi khi lại không nắm rõ dẫn đến sử dụng sai, không đúng triết lý của lập trình hướng đối tượng.

Lập trình hướng đối tượng là gì? Lập trình hướng đối tượng là một kỹ thuật lập trình cho phép lập trình viên tạo ra các đối tượng trong code trừu tượng hóa các đối tượng thực tế trong cuộc sống. Hướng tiếp cận này hiện đang rất thành công và đã trở thành một trong những khuôn mẫu phát triển phần mềm, đặc biệt là các phần mềm cho doanh nghiệp.

Khi phát triển ứng dụng sử dụng OOP, chúng ta sẽ định nghĩa các lớp (class) để mô hình các đối tượng thực tế. Trong ứng dụng các lớp này sẽ được khởi tạo thành các đối tượng và trong suốt thời gian ứng dụng chạy, các phương thức (method) của đối tượng này sẽ được gọi.

Lớp định nghĩa đối tượng sẽ như thế nào: gồm những phương thức và thuộc tính (property) gì. Một đối tượng chỉ là một thể hiện của lớp. Các lớp tương tác với nhau bởi các public API: là tập các phương thức, thuộc tính public của nó.

Lớp và đối tượng

OOP có 3 nguyên lý cơ bản chúng ta sẽ cùng tìm hiểu chi tiết sau đây đó là:

Tính đóng gói (Encapsulation)

Tính đóng gói tức là quy tắc yêu cầu trạng thái bên trong của một đối tượng được bảo vệ và tránh truy cập được từ code bên ngoài (tức là code bên ngoài không thể trực tiếp nhìn thấy và thay đổi trạng thái của đối tượng đó). Bất cứ truy cập nào tới trạng thái bên trong này bắt buộc phải thông qua một public API để đảm bảo trạng thái của đối tượng luôn hợp lệ bởi vì các public API chịu trách nhiệm thực hiện kiểm tra tính hợp lệ cũng như trình tự cập nhật trạng thái của đối tượng đó.

Nói chung trạng thái đối tượng không hợp lệ thường do: chưa được kiểm tra tính hợp lệ, các bước thực hiện không đúng trình tự hoặc bị bỏ qua nên trong OOP

có một quy tắc quan trọng cần nhớ đó là phải luôn khai báo các trạng thái bên trong của đối tượng là private và chỉ cho truy cập qua các public/protected method/property. Khi sử dụng các đối tượng ta không cần biết bên trong nó làm việc như thế nào, ta chỉ cần biết các public API là gì và điều này đảm bảo những gì thay đổi đối tượng sẽ được kiểm tra bởi các quy tắc logic bên trong, tránh đối tượng bị sử dụng không chính xác.

Nguyên lý đóng gói như thế này ở đâu ta cũng có thể bắt gặp ví dụ như thiết kế viên thuốc, chúng ta chỉ biết nó chữa bệnh này, bệnh kia và một số thành phần chính còn cụ thể bên trong nó có những gì thì hoàn toàn không biết.

Tính kế thừa (Inheritance)

Khi bắt đầu xây dựng ứng dụng chúng ta sẽ bắt đầu việc thiết kế các lớp, thông thường chúng ta sẽ thấy có trường hợp một số lớp dường như có quan hệ với những lớp khác, chúng có những đặc tính khá giống nhau. VD: 3 lớp AndroidPhone, iPhone, WindowsPhone

Mỗi lớp đều đại diện cho một loại smartphone khác nhau nhưng lại có những thuộc tính giống nhau. Thay vì sao chép những thuộc tính này, sẽ hay hơn nếu ta đặt chúng ở một nơi có thể dùng bởi những lớp khác. Điều này được thực hiện bởi tính kế thừa trong OOP: chúng ta có thể định nghĩa lớp cha – base class (trong trường hợp này là Smartphone) và có những lớp con kế thừa từ nó (derived class), tạo ra một mối quan hệ cha/con.

Bây giờ, các lớp con có thể kế thừa 3 thuộc tính từ lớp cha. Nếu các chức năng của lớp cha đã được định nghĩa đầy đủ thì lập trình viên sẽ không phải làm bất cứ việc gì ở lớp con. Còn nếu một lớp con muốn chức năng khác so với định nghĩa ở lớp cha thì nó có thể ghi đè (override) chức năng đã được định nghĩa trên lớp cha này.

Tính đa hình (Polymorphism)

Với đa số lập trình viên thì tính Kế thừa và Đóng gói trong OOP khá dễ hiểu còn tính Đa hình khi mới tiếp cận sẽ thấy khó hiểu hơn một chút. Tuy nhiên đây lại là một tính chất có thể nói là chứa đựng hầu hết sức mạnh của lập trình hướng đối tượng. Hiểu một cách đơn giản: Đa hình là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

Ví dụ như ở phần trên, mỗi một smartphone kế thừa từ lớp Smartphone nhưng có thể lưu trữ dữ liệu trên cloud theo những cách khác nhau:

AndroidPhone lưu trữ bằng Google Drive Iphone lưu trên iCloud WindowsPhone sử dụng SkyDrive.

Bởi vì tất cả đều là Smartphone nên nếu ta viết một hàm dùng kiểu Smartphone làm tham số thì khi gọi hàm ta có thể truyền vào một đối tượng kiểu AndroidPhone, Iphone hoặc WindowsPhone bởi vì chúng đều kế thừa từ lớp Smartphone nên được chấp nhận (hiểu nôm na một AndroidPhone, Iphone, WindowsPhone cũng là một Smartphone). Bên cạnh đó hàm này thậm chí không cần quan tâm smartphone nào được truyền vào do nó chỉ cần biết đối tượng đang xử lý ở đây là Smartphone với những public method/property đã được định nghĩa. Nếu các lớp con không định nghĩa lại (overrides) phương thức CloudStore() thì phương thức CloudStore() trên lớp cha (Smartphone) sẽ được gọi. Còn nếu lớp con override lại phương thức CloudStore() của lớp cha như ở hình trên thì phương thức CloudStore() trên lớp con sẽ được gọi mặc dù code trong hàm đang thao tác với đối tượng kiểu Smartphone.

Tính Đa hình như trên là một tính chất rất mạnh mẽ bởi vì nó mang lại cho code khả năng tổng quát hóa cao. Chúng ta không cần tạo ra phương thức cho mỗi kiểu kế thừa từ lớp cha Smartphone mà chỉ cần nhận một biến kiểu Smartphone và có thể làm việc với bất cứ lớp nào kế thừa từ nó. Điều duy nhất không làm được ở

đây là sử dụng những phương thức mà chỉ được khai báo trên các lớp con. VD: nếu ta có một phương thức trên lớp iPhone gọi là `OpenSiri()` nhưng không được khai báo trên lớp Smartphone, khi đó muốn gọi nó sẽ bắt buộc phải ép kiểu từ Smartphone sang iPhone trước khi gọi.

Interface

Đa hình dựa trên Kế thừa không phải bao giờ cũng là lựa chọn tốt nhất. Ta thấy rõ ràng rằng 3 lớp bên dưới (iPhone, Laptop, FingerprintScanner) đều là những thứ có thể truy cập được bằng vân tay nhưng chúng thực hiện theo những cách khác nhau. Những lớp này có chung một hành động tạm gọi là định danh bằng sinh trắc học – `BiometricAuth()`. Nếu ta cố gắng gộp cả 3 lớp này vào thành 1 lớp chung sẽ không hay vì rất khó để tìm ra điểm chung tổng quát của chúng ngoài việc có thể truy cập bằng vân tay.

Do vậy thay vì sử dụng Kế thừa ở đây, ta có thể sử dụng một kỹ thuật khác đó là Interface. Interface đơn giản là một giao ước chỉ ra rằng code của bạn sẽ thực thi và hỗ trợ một public API cụ thể nào đó. Tuy nhiên các public API này được thực hiện như thế nào thì không được chỉ ra trên Interface mà sẽ được chỉ ra trên lớp thực thi interface này. Về cơ bản giao ước là một danh sách các public method/property mà chắc chắn sẽ được thực thi trong lớp của bạn.

Áp vào VD trên, ta có thể tạo ra một interface là `IBiometricAuth` với một phương thức là `BiometricAuth()`. Tiếp theo cho các lớp iPhone, Laptop, FingerprintScanner thực thi interface `IBiometricAuth` này như hình sau

Vì mỗi lớp trên đều thực thi interface `IBiometricAuth` nên ta có thể đảm bảo rằng chúng đều có phương thức `BiometricAuth()` và khai báo của phương thức sẽ giống y như được định nghĩa trên interface `IBiometricAuth`. Tương tự Kế thừa dựa trên Đa hình, sử dụng Interface cho phép chúng ta khai báo phương thức nhận tham số kiểu `IBiometricAuth` nhưng chấp nhận bất cứ đối tượng nào truyền vào mà kiểu

của nó thực thi interface `IBiometricAuth` này. Các lớp thực thi `IBiometricAuth` không cần phải có chung lớp cha ngoại trừ interface `IBiometricAuth`. Trong phương thức ở trên, ta có thể gọi bất cứ phương thức nào đã được định nghĩa trên interface `IBiometricAuth` của đối tượng truyền vào mà không cần quan tâm kiểu thực sự của nó là gì: Không cần quan tâm nó là `iPhone`, `Laptop` hay `FingerprintScanner`, chỉ cần biết nó hỗ trợ interface `IBiometricAuth` là có thể gọi được phương thức `BiometricAuth()`, hết sức linh hoạt và mềm dẻo.

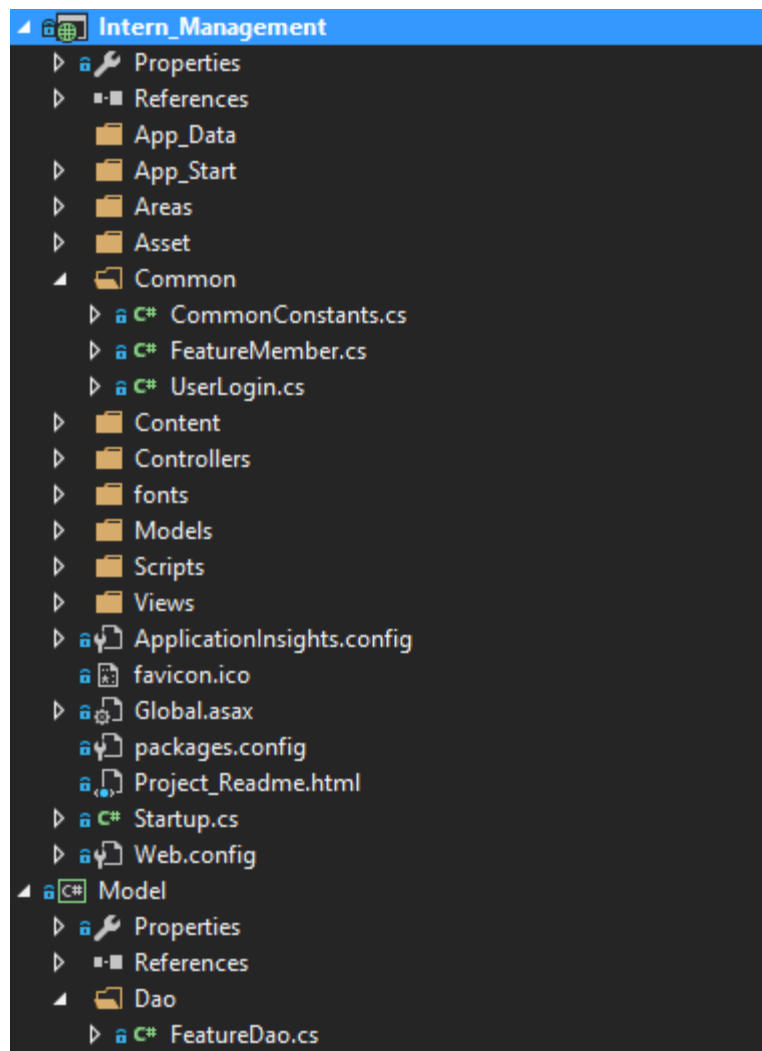
3.1.5. Lập trình Front End với HTML5, CSS, JAVASCRIPT, JQUERY...

Phần front-end của một trang web là phần tương tác với người dùng. Tất cả mọi thứ bạn nhìn thấy khi điều hướng trên Internet, từ các font chữ, màu sắc cho tới các menu xổ xuống và các thanh trượt, là một sự kết hợp của `HTML`, `CSS`, và `JavaScript` được điều khiển bởi trình duyệt máy tính của bạn.

Các kỹ năng và công cụ

Các lập trình viên front-end chịu trách nhiệm cho giao diện của một trang web và kiến trúc những trải nghiệm của người dùng. Để thực hiện được những mục tiêu đó, các lập trình viên front-end phải tinh thông 3 ngôn ngữ chính: `HTML`, `CSS`, và ngôn ngữ lập trình `JavaScript`. Ngoài việc thông thạo các ngôn ngữ đó, các lập trình viên front-end cần phải làm quen với các framework như `Bootstrap`, `Foundation`, `Backbone`, `AngularJS`, và `EmberJS`, để đảm bảo nội dung luôn hiển thị tốt trên mọi thiết bị khác nhau, và các thư viện như `jQuery` và `LESS`, đóng gói code vào trong một hình thức giúp tiết kiệm thời gian và hữu dụng hơn. Rất nhiều công việc dành cho lập trình viên front-end cũng yêu cầu kinh nghiệm với `Ajax`, một kỹ thuật được sử dụng rộng rãi bằng cách dùng `JavaScript` để cho phép các trang load một cách tự động bằng cách tải dữ liệu máy chủ ở phần background.

3.1.6. Quy trình và cách thức tổ chức thư mục trong lập trình của công ty.



Hình 3.4 Cách bố trí thư mục trong một dự án web asp.net mvc

KẾT LUẬN

Kiến tập là giai đoạn chuyển tiếp giữa môi trường học tập với môi trường làm việc thực tiễn tạo cho em có cơ hội tiếp xúc với công việc thực tế hơn gắn kết lý thuyết đã học ở môi trường đại học với công việc thực tiễn ngoài xã hội. Đợt kiến tập là một trong những trải nghiệm mới thử thách mới. Đợt kiến tập cho em học hỏi được nhiều kinh nghiệm trong việc làm nhóm và làm độc lập. Rèn luyện cho bản thân tính kiên nhẫn, bộc lộ được hết khả năng tư duy và sáng tạo của bản thân. Với những nội dung đã học ở trường việc thực tập giúp em áp dụng được các kiến thức đã học vào thực tế một cách sáng tạo và khoa học. Giúp em nâng cao được kiến thức khi làm việc ở 1 công ty lập trình, tìm hiểu được công việc ở các chuyên ngành khác nhau và vị trí công việc trong tương lai. Hiểu biết được kỹ năng chuyên môn cũng như kỹ năng mềm để hoàn thành tốt công việc được giao. Đợt kiến tập này giúp em nhận thức sâu hơn về công việc sau này cũng như bổ sung những khiếm khuyết trong những kỹ năng của bản thân. Hoàn thành đợt kiến tập là một trong những khởi đầu là tiền đề cho thực tập cũng như công việc sau này, là hành trang để em gói gọn vào đời. Bước chuẩn bị để em tự tin hơn. Em xin chân thành cảm ơn thầy Trần Vũ Đại và thầy Nguyễn Văn Khiêm đã hướng dẫn chúng em trong đợt kiến tập vừa qua