**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC THÀNH ĐÔNG**

----------\*\*\*----------



**ĐỒ ÁN TỐT NGHIỆP**

**NGHIÊN CỨU CÔNG NGHỆ ASP.NET VÀ FACEBOOK API**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn  Sinh viên thực hiện  Lớp  Ngành | : **HOÀNG ANH TUẤN**  : **TRẦN THỊ NHUNG**  **: 3C\_CNTT6**  : **Công nghệ thông tin** |

***Hải Dương, Tháng 10 năm 2017***

# LỜI CẢM ƠN

Tri thức là một biển cả mênh mông, mỗi một con người cụ thể muốn nắm bắt tri thức để làm hành trang cho cuộc đời, không loại trừ là phải học tập. Chúng ta sinh ra và lớn lên trong sự nuôi dưỡng của cha mẹ và hạnh phúc biết bao được thầy cô “gieo mầm tri thức” để mai này khi trưởng thành là một công dân tốt và có cơ hội cống hiến cho xã hội.

Để đạt được thành quả và sự hiểu biết như ngày hôm nay là nhờ phần lớn công ơn thầy cô trường Đại học Thành Đông, đặc biệt là sự dạy dỗ nhiệt tình của các thầy cô khoa Công Nghệ Thông Tin đã hết lòng truyền đạt những kiến thức, kinh nghiệm quý báu trong suốt quá trình học tập vừa qua tại trường để chúng em làm hành trang chuẩn bị hòa nhập vào cuộc sống. Cùng với sự hướng dẫn nhiệt tình của các anh chị trong công ty VCCORP đã giúp em củng cố kiến thức đã học tại trường và thấy rõ sự phong phú giữa lý thuyết với thực tế.

Để hoàn thành báo cáo này, em xin trân trọng có lời cảm ơn chân thành đến ban lãnh đạo công ty VCCORP.Đặc biệt xin chân thành cảm ơn đến thầy giáo **Hoàng Anh Tuấn**, người đã trực tiếp hướng dẫn em hoàn thành đề tài này.

Vì thời gian thực tập có hạn, cũng như kiến thức bản thân còn nhiều hạn chế nên không sao tránh khỏi những sai sót, kính mong được sự nhận xét, phê bình, góp ý của các thầy cô trường Đại học Thành Đông, để em có điều kiện học hỏi và phấn đấu hơn nữa.

Em xin chân thành cảm ơn!

*Hải Dương, tháng 10 năm 2017*

**Sinh viên thực hiện**

**Trần Thị Nhung**

# MỤC LỤC

[LỜI CẢM ƠN I](#_Toc496082146)

[MỤC LỤC II](#_Toc496082147)

[DANH MỤC SƠ ĐỒ HÌNH V](#_Toc496082148)

[DANH MỤC CỤM TỪ VIẾT TẮT VI](#_Toc496082149)

[LỜI MỞ ĐẦU VII](#_Toc496082150)

[1. Lý do chọn đề tài VII](#_Toc496082151)

[2. Mục tiêu nghiên cứu VII](#_Toc496082152)

[3. Phương pháp nghiên cứu VII](#_Toc496082153)

[4. Phạm vi nghiên cứu VII](#_Toc496082154)

[5. Bố cục của đề tài VIII](#_Toc496082155)

[6. Hướng tiếp cận VIII](#_Toc496082156)

[CHƯƠNG I: TỔNG QUAN VỀ ASP.NET 1](#_Toc496082157)

[1.1. ASP.NET là gì? 1](#_Toc496082158)

[1.2. Lịch sử ra đời của ASP.NET 1](#_Toc496082159)

[1.3. Mô hình ASP.NET 1](#_Toc496082160)

[a. **Framework Class Library (FCL)** 2](#_Toc496082161)

[b. Commom Language Runtime(CLR) 2](#_Toc496082162)

[d. Các điều khiển trên Asp.net 3](#_Toc496082163)

[1.4. Vòng đời của ASP.NET 3](#_Toc496082164)

[a. NET Application Life Cycle: 4](#_Toc496082165)

[b. ASP.NET Page Life Cycle: 4](#_Toc496082166)

[c. ASP.NET Page Life Cycle Events: 5](#_Toc496082167)

[1.5. Ứng dụng ASP.NET 6](#_Toc496082168)

[a. Windows Forms (Winforms) 6](#_Toc496082169)

[b. Web Forms 7](#_Toc496082170)

[1.6. Tính bảo mật ASP.NET qua các phiên bản 8](#_Toc496082171)

[1.7. Công cụ phát triển ASP.NET 10](#_Toc496082172)

[CHƯƠNG II : TỔNG QUAN VỀ API 11](#_Toc496082173)

[2.1 Khái niệm về API 11](#_Toc496082174)

[2.2 Mô tả về API 11](#_Toc496082175)

[2.3 Các ứng dụng của API 13](#_Toc496082176)

[2.4 Tìm hiểu về FACEBOOK API 14](#_Toc496082177)

[2.4.1. Facebook API là gì 14](#_Toc496082178)

[2.4.2. Cách thức hoạt động của Facebook API 15](#_Toc496082179)

[2.4.3. Cách thức làm việc của Facebook API 16](#_Toc496082180)

[2.5 Tìm hiểu về WEB API 16](#_Toc496082181)

[2.5.1. Web API là gì 16](#_Toc496082182)

[2.5.2. RESTful Service 18](#_Toc496082183)

[2.5.3. Nguyên tắc cơ bản để tạo ra RESTful Service 18](#_Toc496082184)

[a. Sử dụng các phương thức HTTP một cách rõ ràng. 18](#_Toc496082185)

[b. Phi trạng thái (Stateless). 19](#_Toc496082186)

[c. Hiển thị cấu trúc thư mục như URI. 20](#_Toc496082187)

[d. Chuyển đổi JSON, XML hoặc cả hai. 20](#_Toc496082188)

[CHƯƠNG III: PHÂN TÍCH THIẾT KẾ VỚI UML 22](#_Toc496082189)

[3.1 Biểu đồ usecase 22](#_Toc496082190)

[3.1.1. Biểu đồ phân cấp chức năng 22](#_Toc496082191)

[3.2. Biểu đồ luồng xử lý 24](#_Toc496082192)

[3.2.1. Chức năng người dùng đăng nhập 25](#_Toc496082193)

[3.2.2. Chức năng người dùng Chia sẻ bài viết 26](#_Toc496082194)

[3.2.3. Chức năng người dùng Chia sẻ Hình ảnh 26](#_Toc496082195)

[3.2.4. Chức năng người dùng Chia sẻ Video 27](#_Toc496082196)

[3.2.5. Chức năng người dùng Gửi thông báo 27](#_Toc496082197)

[3.2.6. Chức năng người dùng đăng xuất 28](#_Toc496082198)

[3.3. Biểu đồ tuần tự 28](#_Toc496082199)

[3.3.1. Chức năng người dùng đăng nhập 28](#_Toc496082200)

[3.3.2. Chức năng Chia sẻ bài viết 29](#_Toc496082201)

[3.3.3. Chức năng Chia sẻ hình ảnh 30](#_Toc496082202)

[3.2.4. Chức năng gửi thông báo 30](#_Toc496082203)

[3.2.5. Chức năng Xem danh sách bạn bè 31](#_Toc496082204)

[3.2.6. Chức năng Đăng xuất 32](#_Toc496082205)

[CHƯƠNG IV: THIẾT KẾ GIAO DIỆN VÀ KIỂM THỬ 33](#_Toc496082206)

[4.1. Lựa chọn môi trường cài đặt 33](#_Toc496082207)

[4.1.1. Công nghệ .Net và Microsoft Visual Studio 2013 33](#_Toc496082208)

[a. Microsoft .NET 33](#_Toc496082209)

[b. Công cụ Visual Studio 2013 34](#_Toc496082210)

[4.1.2. Ngôn ngữ ASP.NET 35](#_Toc496082211)

[4.1.3. Ngôn ngữ C# 35](#_Toc496082212)

[4.1.4. Giao diện Bootstrap 36](#_Toc496082213)

[4.1.5. Bộ cài đặt Facebook SDK 37](#_Toc496082214)

[4.2. Hướng dẫn cài đặt 38](#_Toc496082215)

[4.3. Hướng dẫn cách sử dụng 38](#_Toc496082216)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 43](#_Toc496082217)

[1. Kết quả đạt được của đề tài 43](#_Toc496082218)

[2. Ưu điểm và nhược điểm 43](#_Toc496082219)

[3. Hướng phát triển của đề tài 43](#_Toc496082220)

[CHÚ THÍCH 44](#_Toc496082221)

[TÀI LIỆU THAM KHẢO 44](#_Toc496082222)

# DANH MỤC SƠ ĐỒ HÌNH

Hình 1.1 Mô hình framework của ASP.NET 1

Hình 1.2 Mô hình xử lý trang của ASP.NET 3

Hình 1.3 Hình Sự kiện của Page ASP.NET 5

Hình 2.1 Hoạt động của API 11

Hình 2.2. Ứng dụng của API trong website 13

Hình 2.3 Cách thức hoạt động của Facebook API 15

Hình 2.4 Cách thức làm việc của Facebook API 16

Hình 2.5 Mô hình ASP.NET API 17

Hình 3.1: Sơ đồ phân cấp chức năng của hệ thống 22

Hình 3.2: Usecase Ứng dụng Facebook API 24

Hình 3.3. Luồng xử lý Chức năng Đăng nhập 25

Hình 3.4. Luồng xử lý Chức năng Chia sẻ bài viết 26

Hình 3.5. Luồng xử lý Chức năng Chia sẻ Hình ảnh 26

Hình 3.6. Luồng xử lý Chức năng Chia sẻ Video 27

Hình 3.7. Luồng xử lý Chức năng Gửi thông báo 27

Hình 3.8. Luồng xử lý Chức năng Đăng xuất 28

Hình 3.9. Biểu đồ tuần tự chức năng Người dùng đăng nhập 28

Hình 3.10. Biểu đồ tuần tự chức năng Chia sẻ bài viết 29

Hình 3.11. Biểu đồ tuần tự chức năng Chia sẻ hình ảnh 29

Hình 3.12. Biểu đồ tuần tự chức năng Gửi thông báo 30

Hình 3.13. Biểu đồ tuần tự chức năng Danh sách bạn bè 30

Hình 3.14. Biểu đồ tuần tự chức năng Đăng xuất 31

Hình 4.1. Màn hình Trang chủ 37

Hình 4.2. Màn hình Đăng nhập 38

Hình 4.3. Màn hình Thống kê 38

Hình 4.4. Màn hình Chia sẻ bài viết 38

Hình 4.5. Màn hình Chia sẻ hình ảnh 39

Hình 4.6. Màn hình Chia sẻ Video 39

Hình 4.7. Màn hình trên Fanpage 40

Hình 4.8. Màn hình Thông tin người dùng 40

Hình 4.9. Màn hình Gửi thông báo 41

Bảng 1.1 Bảng giai đoạn chu kỳ vòng đời của ASP.NET 5

Bảng 1.2 Bảng Sự kiện của page life cycle ASP.NET 6

Bảng 1.3 Bảng Phiên bản của ASP.NET 8

Bảng 1.4 Công cụ hỗ trợ phát triển ASP.NET 10

# DANH MỤC CỤM TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Từ đầy đủ** |
| ASP.NET | Active Server Pages. NET |
| CLR | Common Language Runtime |
| FCL | **Framework Class Library** |
| API | Application Programming Interface |
| Life Cycle | Vòng đời |
| RESTFul | Representational State Transfer |
| JSON | JavaScript Object Noattion |
| XML | Extensible Markup Language |
| MSIL | Microsoft Intermediate Language |

# LỜI MỞ ĐẦU

## 1. Lý do chọn đề tài

Những năm gần đây, mạng xã hội đã trở thành một nhu cầu khá phổ biến của nhiều người,đặc biệt là giới trẻ trên toàn thế giới. Nó được coi như cuộc sống ảo của con người trong xã hội hiện đại. Mạng xã hội được sử dụng như một diễn đàn để trao đổi thông tin, giải trí, kết nối bạn bè, ngoài ra nó còn như một kênh thông tin cung cấp những tin tức cập nhật hàng ngày. Tuy nhiên ở Việt Nam hiện nay, ở khía cạnh là một kênh thông tin, ảnh hưởng của mạng xã hội chưa thực sự nổi trội.

Trong những năm gần đây, Facebook đã nhanh chóng trở thành mạng xã hội số một thế giới và chính các ứng dụng trên Facebook đã góp một phần không nhỏ để tạo nên điều đó. Những ứng dụng giải trí,quảng cáo trên Facebook có tính cộng đồng cao, thu hút người sử dụng.

Theo ý kiến thống kê được thì có tới 65% người dùng sử dụng ứng dụng để giải trí, 20% ý kiến cho rằng ứng dụng thật sự phong phú và chỉ hơn 3% ý kiến cho rằng ứng dụng phức tạp và khó sử dụng. Với thống kê như trên, có thể dễ dàng nhận ra được sức hút của các ứng dụng trên mạng Facebook và việc tạo dựng các ứng dụng mới trên Facebook là khá cần thiết. Vì vậy em chọn đề tài “Nghiên cứu công nghệ về ASP.NET và FACEBOOK API” làm đề tài thực tập tốt nghiệp.

## 2. Mục tiêu nghiên cứu

* Tìm hiểu công nghệ ASP.NET
* Tìm hiểu về Facebook API.
* Facebook Platform : Nền tảng của Facebook.
* Xây dựng ứng dụng trên nền Facebook.

## 3. Phương pháp nghiên cứu

* Nghiên cứu tài liệu
* Nghiên cứu các đối tượng Facebook cung cấp.
* Phân tích.

## 4. Phạm vi nghiên cứu

* Visual Studio 2013
* Ngôn ngữ ASP.NET
* Công cụ phân tích hệ thống hướng đối tượng: UML

## 5. Bố cục của đề tài

Mở đầu

- Chương I: **Tổng quan về ASP.NET**. Chương này tập trung giới thiệu khái quát về công nghệ ASP.NET, mô hình và ứng dụng và tính bảo mật của ASP.NET qua các phiên bản.

- Chương II: **Tổng quan về API**. Chương này tập trung giới thiệu khái quát về mạng xã hội nói chung và mạng Facebook nói riêng, lịch sử phát triển và phương thức hoạt động của mạng Facebook. Cuối cùng là một số thống kê về mạng Facebook so với các mạng xã hội khác.

- Chương III**: Phân tích thiết kế UML**.Chương này sẽ mô tả ứng dụng thực tế xây dựng được bằng việc sử dụng các đối tượng mà Facebook cung cấp.

- Chương IV**: Thiết kế giao diện và Kiểm thử**: Chương này sẽ mô tả ứng dụng thực tế xây dựng được bằng việc sử dụng các đối tượng mà Facebook cung cấp.

Kết luận

## 6. Hướng tiếp cận

**Đối tượng sử dụng**: Theo thống kê mới nhất năm 2017, hiện có khoảng 35 triệu người dùng mạng xã hội ở Việt Nam, hơn 1/3 dân số đang sở hữu một tài khoản của Facebook.

**Hướng phát triển của Facebook**: Trong 5 năm tới Facebook sẽ không chỉ có địa chỉ facebook.com mà còn gồm nhiều hệ thống khác, Facebook cố gắng mở rộng số lượng người dùng, phát triển các hệ thống site. Bên cạnh đó Facebook có một số lượng lớn các thông tin cá nhân trên mạng. Nếu sử dụng khéo léo các thông tin này, rất có thể Facebook sẽ tạo ra một hình thái mạng xã hội khác, tại đó bạn bè cung cấp cho nhau những thông tin họ cần. Trong các năm tiếp theo có thể facebook sẽ phát triển theo hướng này. Khi đó Facebook sẽ chuyển từ một trang web xã hội sang một trang web cung cấp các dịch vụ đáp ứng nhu cầu của người sử dụng. Hiện nay tính năng này đang được thử nghiệm.

# CHƯƠNG I: TỔNG QUAN VỀ ASP.NET

# 1.1. ASP.NET là gì?

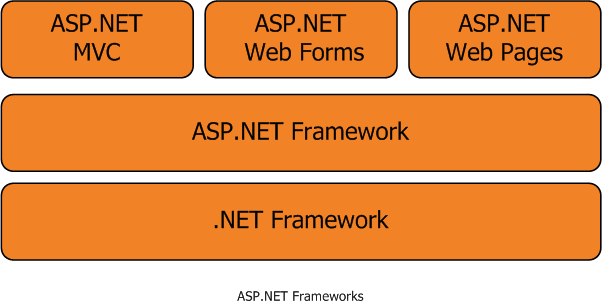
**ASP.NET** là một nền tảng ứng dụng web (*web application framework*) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. Lần đầu tiên được đưa ra thị trường vào tháng 2 năm 2002 cùng với phiên bản 1.0 của.NET framework, là công nghệ nối tiếp của Microsoft's Active Server Pages(ASP).

**ASP.NET** được biên dịch dưới dạng Common Language Runtime (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi .NET language.

# 1.2. Lịch sử ra đời của ASP.NET

Sau khi phát hành phiên bảo Internet Information Service 4.0 vào năm 1997, hãng Microsoft bắt đầu nghiên cứu một mô hình ứng dụng web để giải quyết những bất tiện của ASP, đặc biệt là việc tách riêng biệt phần thể hiện và phần nội dung cũng như cách viết mã (code) rõ ràng hơn. Mark Anders, quản lý nhóm IIS và Scott Guthrie, gia nhập Microsoft vào năm 1997 sau khi tốt nghiệp Đại học Duke, được giao nhiệm vụ định hình mô hình cần phát triển. Những thiết kế ban đầu được thực hiện trong vòng 2 tháng bởi Anders và Guthrie, Guthrie đã viết mã prototype đầu tiên trong khoảng thời gian nghỉ lễ Giáng sinh năm 1997.

# 1.3. Mô hình ASP.NET



*Hình 1.1 Mô hình framework của ASP.NET*

ASP.NET là một phần của .NET Framework để xây dựng trang ASP.NET bạn cần thêm vào các đặc tính của .net framework. .Net Framework chứa đựng hai phần Framework Class Library (FCL) và Commom Language Runtime (CLR).

1. **Framework Class Library (FCL)**

Là một thư viện chuẩn và Microsoft .NET Framework thực hiện thư viện chuẩn theo quy định của cơ sở hạ tầng Common Language . FCL là một tập hợp các lớp, giao diện và các loại giá trị sử dụng lại. Các Base Class Library (BCL) là cốt lõi của FCL và cung cấp các chức năng cơ bản nhất, trong đó bao gồm các lớp trong không gian System, System.CodeDom,System.Collections,System.Diagnostics,System.IO,System.Resources và System.Text. Ví dụ một vài lớp của .Net Famework:

* **Lớp File**: cho phép bạn tạo file, sửa, xoá hay kiểm tra sự tồn tại của file trên đĩa cứng…
* **Lớp Graphics**: Cho phép bạn làm việc với nhiều kiểu của ảnh, bạn cũng có thể tạo ra các ảnh từ các phương thức trên lớp này.
* **Lớp SmtpClient**: Cho phép bạn gửi thư.
* **Hiểu về Namespaces**: hơn 13 nghìn lớp trong Net framework. Đây là một con số rất lớn, Microsoft đã chia các lớp cùng xử lý về một vấn đề gì đó vào các không gian tên chung hay namespaces.

### Commom Language Runtime(CLR)

* Khi bạn viết ứng dụng bằng ngôn ngữ C#, VB.NET hay bằng một ngôn ngữ bất kỳ trên nền .Net Framework mã của bạn sẽ được không bao giờ biên dịch trực tiếp thành mã máy. Thay vào đó chúng được biên dạng sang ngôn ngữ đặc tả MSIL (Microsoft intermediate Language).
* MSIL nhìn rất giống với ngôn ngữ hướng đối tượng Assembly, nhưng không giống kiểu ngôn ngữ Assembly. MSIL là ngôn ngữ bậc thấp và phụ thuộc vào Platform.
* Khi ứng dụng của bạn thực thi, mã MSIL là “just in time” biên dịch sang mã máy bởi JITTER (just in time compiler).

Như vậy khi bạn viết các lớp trên .Net bằng bất kỳ ngôn ngữ nào khi bạn biên dịch sang Assembly bạn đều có thể sử dụng Assembly đó cho các ngôn ngữ khác.

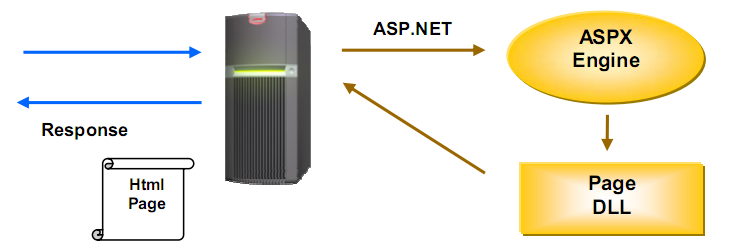
1. **Assembly**

Một Assembly là một file dll trên đĩa cứng của bạn, nơi mà lưu trữ các lớp của .NET, ví dụ tất cả các lớp trong ASP.NET Framework đều nằm trong Assembly System.web.dll. Trước khi sử dụng các lớp trong dll bạn cần tạo một tham chiếu đến file dll này.

### Các điều khiển trên Asp.net

Các điều khiển asp.net là phần quan trọng nhất trong ASP.NET Framework. một Control ASP.NET là một lớp mà thực thi trên server và đưa ra nội dung trên trình duyệt. ASP.NET có hơn 70 control mà bạn có thể sử dụng trong xây dựng ứng dụng web của bạn và cơ bản nó chia ra các nhóm control sau:

* **Standard control**: bao gồm các điều khiển đưa ra các thành phần chuẩn của form như: Label, Button, TextBox…
* **Validator Control**: là các control cho phép bản kiểm tra tính hợp lệ của các control cho phép nhập giá trị trên form.
* **Rich Control**: là những điều khiển như FileUpload, Calendar…
* **Data Control:** là các điều khiển cho phép thao tác với dữ liệu
* **Navigation Control**: là những điều khiển giúp bạn dễ dàng di chuyển giữa các trang trong website.
* **Login control**: Là các điều khiển về bảo mật của ứng dụng cho phép bạn đưa ra các form đăng nhập, thay đổi mật khẩu…
* **HTML Control**: cho phép bạn chuyển các điều khiển của HTML thành các điều khiển có thể làm việc trên server.



*Hình 1.2 Mô hình xử lý trang của ASP.NET*

# 1.4. Vòng đời của ASP.NET

Vòng đời (life cycle): Là một quá trình từ việc bắt đầu khởi tạo cho đến lúc kết thúc một ứng dụng.

Vòng đời của một trang ASP.NET được chưa thành 2 nhóm:

* Application Life Cycle
* Page Life Cycle

### NET Application Life Cycle:

Application life cycle có thể chia thành các giai đoạn sau:

Người dùng gửi một yêu cầu truy cập vào dữ liệu của ứng dụng. Trình duyệt sẽ gửi yêu cầu này đến Web Server. Các sự kiện sau đây sẽ được khởi tạo:

* Một đối tượng cũ lớp Application Manager được tạo.
* Một đối tượng của lớp Hosting Environment được tạo để cung cấp thông tin về nguồn dữ liệu.
* Các thành phần đầu của ứng dụng sẽ được biên dịch.

Các đối tượng như HttpContext, HttpRequest và HttpResponse được khởi tạo và cài đặt. Một thể hiện của đối tượng HttpApplication được tạo và gắn cho yêu cầu. Các yêu cầu được xử lý bởi lớp HttpApplication, các sự kiện khác nhau được kích hoạt với lớp này để xử lí các yêu cầu.

### b. ASP.NET Page Life Cycle:

Khi một trang được yêu cầu, nó sẽ được load vào bộ nhớ của Server, xử lí và gửi lại Browser. Sau đó nó sẽ bị giải phóng khỏi bộ nhớ. Vào mỗi bước, các phương thức và sự kiện luôn có sẵn, ta có thể viết lại cách xử lí tương ứng cho mỗi ứng dụng.

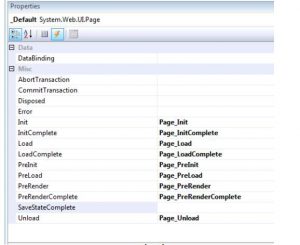
Các giai đoạn chung trong vòng đời của trang ASP.NET

|  |  |
| --- | --- |
| **Giai đoạn** | **Mô tả** |
| Page request  (Yêu cầu trang) | Yêu cầu trang xuất hiện trước khi chu kỳ đời trang bắt đầu. Khi người dùng yêu cầu trang, ASP.NET sẽ xác định liệu trang cần phải được phân tích cú pháp và biên soạn hay liệu phiên bản lưu trong bộ nhớ cache của trang có thể được gửi đáp ứng mà không cần chạy trang. |
| Start  (Khởi đầu) | Trong giai đoạn bắt đầu, các thuộc tính của trang như Request và Response được đặt. Ở giai đoạn này, trang cũng xác định yêu cầu là một postback hay một yêu cầu mới và đặt thuộc tính IsPostBack . Trang cũng đặt thuộc tính UICulture . |
| Initialization  (Khởi tạo) | Trong quá trình khởi tạo trang, các điều khiển trên trang có sẵn và thuộc tính UniqueID của từng điều khiển được thiết lập. Trang chủ và chủ đề cũng được áp dụng cho trang nếu có. Nếu yêu cầu hiện tại là postback, dữ liệu postback chưa được tải và kiểm soát các giá trị thuộc tính chưa được khôi phục lại với các giá trị từ trạng thái xem. |
| Load  (Tải) | Trong quá trình tải, nếu yêu cầu hiện tại là một postback, các thuộc tính điều khiển được nạp với các thông tin thu được từ trạng thái xem và trạng thái kiểm soát. |
| Postback event handling  (Xử lý sự kiện Postback) | Nếu yêu cầu là một postback, điều khiển sự kiện điều khiển được gọi. Sau đó, phương thức Validate của tất cả các điều khiển validator được gọi, đặt thuộc tính IsValid của điều khiển trình kiểm soát cá nhân và của trang. |
| Rendering  (Kết xuất đồ họa) | Trước khi hiển thị, xem trạng thái được lưu cho trang và tất cả các điều khiển. Trong giai đoạn rendering, trang gọi phương thức Render cho mỗi điều khiển, cung cấp một trình soạn thảo văn bản ghi đầu ra của nó tới đối tượng OutputStream của thuộc tính Response của trang . |
| Unload  (Dỡ bỏ) | Các sự kiện Unload được nâng lên sau khi trang đã được rendered đầy đủ, gửi đến khách hàng, và đã sẵn sàng để được loại bỏ. Tại thời điểm này, các thuộc tính của trang Response và Request được dỡ xuống và dọn dẹp được thực hiện. |

*Bảng 1.1. Bảng giai đoạn chu kỳ vòng đời ASP.NET*

### c. ASP.NET Page Life Cycle Events:

Trong mỗi vòng đời của trang, sẽ có một số sự kiện được tạo ra, có thể xử lí theo mục đích cụ thể.



*Hình 1.3. Hình Sự kiện của Page ASP.NET*

|  |  |
| --- | --- |
| **Sự kiện** | **Mô tả** |
| Initialization | Trong giai đoạn này, các property IsPostback được thiết lập. Page sẽ xác định request là Postback (old request) hay đây là lần đầu tiên truy cập và page (new request).  Các control của page được tạo và mỗi 1 control đều được set UniqueID. Nếu request hiện tại là một postback thì dữ liệu chưa được load và giá trị của các control cũng chưa được khôi phục từ view state. |
| Loading | Ở giai đoạn này, nếu request là một Postback thì nó sẽ load data lên từ view state. |
| Rendering | Trước khi render, View State lưu lại page và các control của page. trong suốt giao đoạn này, page gọi đến phương thức render cho mỗi control, cung cấp một trình soạn thảo văn bản ghi output của page tới OutputStream một thuộc tính Response của page. |
| Uploading | Unload được gọi sau khi page đã render xong, gửi kết quả render đến client và kết thúc 1 life cycle. Tại thời điển này các thuộc tính của page cũng như các response, request đều không được load nữa. |

*Bảng 1.2. Bảng Sự kiện của page life cycle ASP.NET*

## 1.5. Ứng dụng ASP.NET

Khi bạn bắt đầu phát triển các ứng dụng với một giao diện người dùng, bạn có thể sử dụng Windows Forms hoặc Web Forms. Cả hai đều có đầy đủ thiết kế thời gian hỗ trợ trong môi trường phát triển, và có thể cung cấp một giao diện người dùng phong phú và chức năng ứng dụng tiên tiến để giải quyết các vấn đề gặp phải.

### a. Windows Forms (Winforms)

**Windows Forms** là thuật ngữ chỉ việc phát triển các ứng dụng giao diện người dùng bằng cách sử dụng các thành phần xây dựng sẵn (build in component) còn được gọi là các điều khiển (controls). Hay nói cách khác Windows Forms là một API cho phép tạo GUI(Graphical User Interface) cho các ứng dụng chạy trên desktop.

Các điều khiển này dùng để hiển thị thông tin cho người dùng cũng như cho người dùng nhập thông tin vào. Windows Forms ra đời đáp ứng nhu cầu tạo ứng dụng nhanh (Rapid Application Development).

Windows Form cho phép người phát triển tạo ra các giao diện người dùng sử dụng các thành phần (components). Các thành phần này được gọi là các điều khiển (controls). Những điều khiển này cho phép chúng ta thu thập thông tin từ người dùng cũng như trình bày các thông tin để người dùng có thể xem.

Một form được chạy trên một máy tính cục bộ (local machine) và một form có thể truy cập đến các tài nguyên khác nhau như bộ nhớ, các thư mục, tệp tin, cơ sở dữ liệu…

Do đó Windows Forms phù hợp với các ứng dụng desktop như các ứng dụng quản lý thông tin, các ứng dụng tương tác trực tiếp với người dùng.

Vai trò của Windows Forms:

* Các Form có thể chứa các điều khiển (các thành phần) khác nhau.
* Xử lý dữ liệu được nhập bởi người dùng.
* Hiển thị (trình bày) các thông tin tới người dùng.
* Kết nối đến các nguồn CSDL khác nhau trên các máy tính cục bộ hoặc máy tính khác.

### b. Web Forms

* ASP.NET Web Forms là nền tảng dành cho các lập trình viên yêu thích mô hình lập trình hướng sự kiện (event driven development) và muốn tăng hiệu suất bằng cách sử dụng các controls. ASP.NET Web Forms hỗ trợ nền tảng lập trình ở phía máy chủ và cung cấp nhiều các server side controls, chúng đóng gói HTML, JavaScript và CSS bên trong, giúp bạn có thể thiết kế ứng dụng một cách dễ dàng và nhanh chóng bằng cách kéo thả các control.
* ASP.NET Web Forms còn cung cấp tính năng data binding nên bạn có thể dễ dàng phát triển các ứng dụng nặng về việc truy xuất và hiển thị dữ liệu.
* Với ASP.NET Web Forms, bạn có thể dễ dàng quản lý trạng thái của các control nhờ vào cơ chế post-back và view state. Mọi thay đổi, tương tác trên form của người dùng đối với trang web sẽ được lưu lại trạng thái và bạn có thể xử lý chúng bằng mã lệnh từ phía server sau mỗi lần người dùng post-back (gửi trả lại trang) cho server.

## 1.6. Tính bảo mật ASP.NET qua các phiên bản

Nó là rất phổ biến cho nhà phát triển truy cập từ xa chia sẻ tệp tin, Dịch vụ Web, cơ sở dữ liệu hoặc các tài nguyên từ các ứng dụng ASP.NET. Này thường là một tệp máy chủ hoặc máy chủ cơ sở dữ liệu tách biệt với máy chủ Web khi ứng dụng đang chạy. Tuy nhiên, để làm việc này hoạt động như mong đợi, có một số cân nhắc bảo mật rất quan trọng để biết về ASP.NET quá trình nhận dạng, xác thực và cấp phép. Chúng tôi sẽ tìm hiểu các khái niệm quan trọng và các giải pháp. Tuỳ chọn nào sẽ hoạt động tốt nhất cho tình huống cụ thể sẽ phụ thuộc vào yêu cầu ứng dụng kiến trúc và bảo mật của bạn.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Ghi chú** | **Các tính năng mới liên quan đến ASP.NET** |
| 16/01/2002 | 1,0 | Phiên bản đầu tiên | Phát triển ứng dụng Web hướng đối tượng hỗ trợ thừa kế , đa hình và các tính năng OOP tiêu chuẩn khác |
| phát hành cùng với Visual Studio.NET | Các nhà phát triển không còn buộc phải sử dụng Server.CreateObject (...), vì vậy việc bảo vệ liên kết sớm và kiểu an toàn là có thể. |
|  | Dựa trên lập trình Windows ; nhà phát triển có thể sử dụng các thư viện lớp DLL và các tính năng khác của máy chủ Web để xây dựng nhiều ứng dụng mạnh mẽ hơn nhiều so với việc tạo dựng HTML (ví dụ xử lý ngoại lệ ) |
| 24/4/ 2003 | 1.1 | được phát hành cùng với Windows Server 2003 | Điều khiển di động |
| phát hành cùng với Visual Studio.NET 2003 | Tự động xác nhận đầu vào |
| 7/11/2005 | 2 | tên mã Whidbey | Kiểm soát dữ liệu mới (GridView, FormView, DetailsView) |
| được phát hành cùng với Visual Studio 2005 và Visual Web Developer Express | Kỹ thuật mới cho truy cập dữ liệu khai báo (SqlDataSource, ObjectDataSource, XmlDataSource controls) |
| và SQL Server 2005 | Điều khiển điều hướng |
|  | Trang chính |
| Kiểm soát đăng nhập |
| Chủ đề |
| Skins |
| Phần Web |
| Dịch vụ cá nhân hóa |
| Tổng hợp trước |
| Kỹ thuật bản địa hóa mới |
| Hỗ trợ bộ xử lý 64-bit |
| Mô hình lớp nhà cung cấp |
| 21/11/ 2006 | 3,0 |  | Windows Presentation Foundation (WPF) |
| Windows Workflow Foundation (WF) |
| Windows Communication Foundation , có thể sử dụng ASP.NET để lưu trữ các dịch vụ |
| Windows CardSpace , sử dụng ASP.NET cho vai trò đăng nhập |
| 19/11/ 2007 | 3,5 | Phát hành với Visual Studio 2008 và Windows Server 2008 | Kiểm soát dữ liệu mới (ListView, DataPager) |
| ASP.NET AJAX được bao gồm như một phần của khuôn khổ |
| Hỗ trợ cho các đường dẫn HTTP và các nguồn cấp dữ liệu cung cấp. |
| WCF hỗ trợ cho RSS, JSON, POX và một phần Trust |
| Tất cả các thay đổi.NET Framework 3.5 , như LINQ vv |
| 11/8/2008 | 3.5 Gói Dịch vụ 1 | Được phát hành với Visual Studio 2008 Gói Dịch vụ 1 | Kết hợp dữ liệu động ASP.NET |
| Hỗ trợ kiểm soát lịch sử trình duyệt trong một ứng dụng ASP.NET AJAX |
| Khả năng kết hợp nhiều tệp JavaScript vào một tệp để tải về hiệu quả hơn |
| Không gian tên mới System.Web.Abstractions và System.Web.Routing |
| 12/4/2010 | 4,0 | Mở rộng song song và các tính năng khác của .Net Framework 4 | Hai thuộc tính mới được thêm vào trong lớp Page là MetaKeyword và MetaDescription. |
| 15/8/2012 | 4,5 | Được phát hành với Visual Studio 2012 và Windows Server 2012 cho Windows 8 |  |
| Mở rộng song song và các tính năng khác của .NET Framework 4.5 |
| 17/10/ 2013 | 4.5.1 | Được phát hành với Visual Studio 2013 [22] choWindows Server 2012 R2 và Windows 8.1 | Bootstrap 3.0 |
| Web API 2: OAuth 2.0, cải tiến OData , CORS |
| MVC 5: Định tuyến thuộc tính, bộ lọc xác thực và ghi đè bộ lọc |
| EF 6 |
| Tín hiệu |
| OWIN |
| 5/5/2014 | 4.5.2 |  | Độ tin cậy cao hơn HTTP tiêu đề kiểm tra và các phương pháp sửa đổi |
| Cách mới để sắp xếp các tác vụ công việc của nhân viên không đồng bộ nền |
| 20/7/2015 | **4,6** | Phát hành với Visual Studio 2015 và Xem trước của EF 7 cho Windows Server 2016 và Windows 10 | Hỗ trợ HTTP / 2 khi chạy trên Windows 10 |
| Các API trả về nhiệm vụ async khác |
| 18/11/ 2015 | 5 RC1 | Phiên bản này sau đó được tách ra khỏi ASP.NET và đưa vào một dự án mới gọi là ASP.NET Core , có phiên bản bắt đầu ở 1.0. [25] | Một dự án hoàn toàn mới với các nguyên lý và mục tiêu phát triển khác nhau |

*Bảng 1.3. Phiên bản của ASP.NET*

## 1.7. Công cụ phát triển ASP.NET

Một số gói phần mềm sẵn có để phát triển ứng dụng ASP.NET:

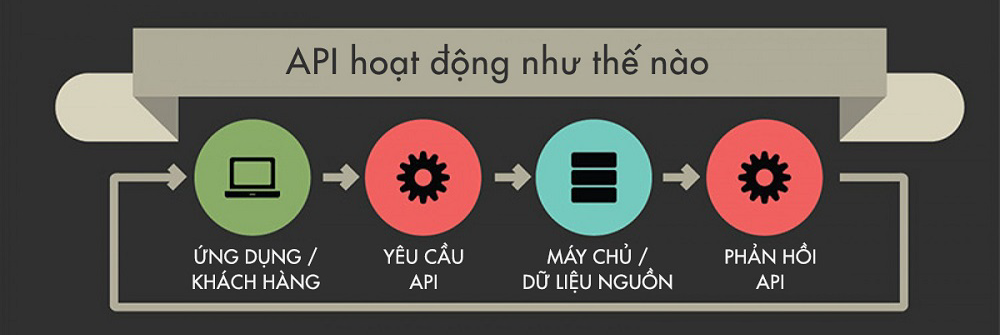
|  |  |  |
| --- | --- | --- |
| **Phần mềm** | **Nhà phát triển** | **Cấp giấy phép** |
| Microsoft Visual Studio | Microsoft | Miễn phí và thương mại |
| Microsoft Visual Web Developer Express | Microsoft | Đăng ký |
| CodeGear Delphi | Công nghệ Embarcadero | Thương mại |
| Macromedia HomeSite | Adobe Systems | Thương mại |
| Microsoft Expression Web | Microsoft | Miễn phí |
| Microsoft SharePoint Designer | Microsoft | Miễn phí |
| MonoDevelop | Xamarin và cộng đồng Mono | Miễn phí mã nguồn mở |
| SharpDevelop | Nhóm ICSharpCode | Miễn phí mã nguồn mở |
| Adobe Dreamweaver | Adobe Systems | Thương mại |

*Bảng 1.4. Công cụ hỗ trợ phát triển ASP.NET*

# CHƯƠNG II : TỔNG QUAN VỀ API

## 2.1 Khái niệm về API

API là viết tắt của Application Programming Interface (Giao diện chương trình ứng dụng) là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng.



*Hình 2.1. Hoạt động của API*

Chẳng hạn, một chương trình máy tính có thể dùng các hàm API của hệ điều hành để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng hiện thực API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính. Đây là phần mềm hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các lập trình viên có thể rút ra từ đó để tạo nên các tính năng giao tiếp người- máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện. Hệ giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn.

## Mô tả về API

Một trong các mục đích chính của một API là cung cấp khả năng truy xuất đến một tập các hàm hay dùng - ví dụ, hàm để vẽ các cửa sổ hay các icon trên màn hình. Các API, cũng như hầu hết các interfaces là trừu tượng (abstract). Phần mềm mà muốn cung cấp truy xuất đến chính nó thông qua các API cho sẵn, phải hiện thực API đó. Trong nhiều tình huống, một API thường là một phần của bộ SDK, hay software development kit. Một bộ SDK có thể bao gồm một API cũng như các công cụ/phần cứng, vì thế hai thuật ngữ này không thay thế cho nhau được.

Có nhiều mô hình thiết kế khác nhau cho các APIs. Interfaces nhằm là cách thực thi nhanh nhất thường gồm các tập các hàm, thủ tục, biến và các cấu trúc dữ liệu. Tuy nhiên, các mô hình khác vẫn tồn tại, như bộ thông dịch dùng để ước giá biểu thức trong ECMAScript/JavaScript.

Một API tốt thường cung cấp một "hộp đen" hay là một lớp trừu tượng (abstraction layer) bao bọc nó, nhằm đảm bảo là nhà lập trình không thể biết cách hiện thực cụ thể bên trong của mỗi hàm trong API.

Điều này làm cho việc thiết kế lại hay cải tiến hàm của API đó trở nên dễ dàng hơn vì nó không làm đổ ổ các đoạn mã khác mà có sử dụng các hàm đó.

Có hai dòng chính sách đối với việc công bố các APIs:

* Một số công ty bảo vệ APIs của họ một cách mạnh mẽ. Ví dụ, Sony thường chỉ cung cấp API chính thức của PlayStation 2 cho các nhà phát triển PlayStation có đăng kí. Điều này là vì Sony muốn giới hạn những người có thể viết trò chơi trên PlayStation 2, và muốn thu lợi nhuận từ những người này càng nhiều càng tốt. Đây thường là chính sách đối với các công ty mà họ không thu lợi từ việc bán các hiện thực API của họ. Tuy nhiên, PlayStation 3 là công bố hoàn toàn APIs.
* Một số công ty thì cung cấp miễn phí APIs. Ví dụ, Microsoft công bố hầu như hoàn toàn thông tin về các API, để cho các phần mềm có thể được viết chạy trên nền Windows. Việc bán của các phần mềm hãng thứ 3 đồng thời với việc phải mua Hệ điều hành Microsoft Windows. Đây thường là các công ty thu lợi nhuận từ việc bán các hiện thực API.

Một số APIs, chẳng hạn các API là chuẩn cho một hệ điều hành, được hiện thực dưới dạng các thư viện mã độc lập được phân phối kèm theo hệ điều hành. Một số khác thì đòi hỏi nhà sản xuất phần mềm phải tích hợp API trực tiếp vào trong chương trình. Microsoft Windows APIs đi kèm theo hệ điều hành cho phép mọi người có thể sử dụng chúng. Phần mềm cho các hệ thống nhúng như thiết bị chơi trò chơi thường thuộc vào loại tích hợp vào trong ứng dụng. Trong khi các tài liệu API chính thức của PlayStation là nên đọc, nhưng nó chẳng giúp ích gì nếu ta chẳng có các hiện thực của nó, dưới dạng một thư viện độc lập hay bộ phát triển phần mềm.

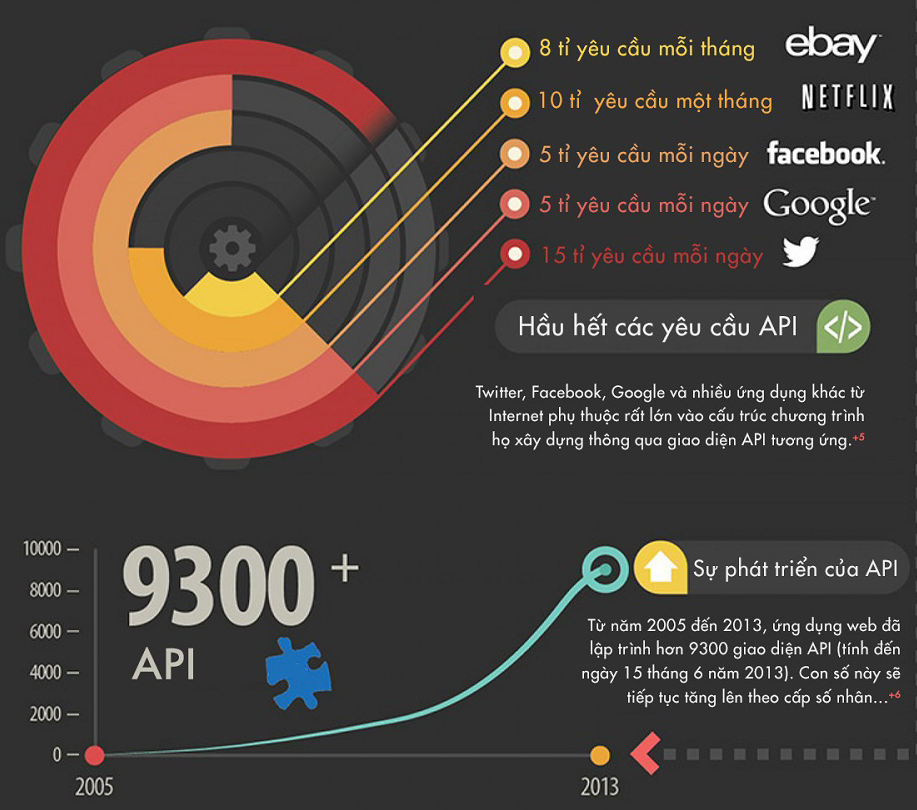
Một API mà cho phép truy xuất và sử dụng tự do được gọi là "mở". Các APIs được cung cấp bởi phần mềm mở (như mọi phần mềm được phân phối theo giấy phép đăng kí GNU), là mở theo đúng nghĩa, vì mọi người có thể xem mã nguồn của phần mềm và tìm ra API. Mặc dù việc tham khảo hiện thực vẫn tồn tại cho một API (như với Microsoft Windows cho Win32 API), thì việc tạo thêm các hiện thực bổ sung vẫn có thể diễn ra. Ví dụ, hầu hết các Win32 API có thể được cung cấp từ hệ thống UNIX dùng phần mềm tên là Wine.

## 2.3 Các ứng dụng của API

**a. Ứng dụng trong website**

Lập trình ứng dụng API cho thiết kế website được tạo ra nhằm hỗ trợ tốt nhất cho người dùng và kỹ thuật viên lập trình có thể tương tác vào website và viết ứng dụng theo nhu cầu của họ.

Dễ hiểu nhất thì **Facebook** là một ví dụ, những nút like và share bài viết của **Facebook** trên các trang web tin tức chính là những **API** do **Facebook** cung cấp. Lập trình viên chỉ việc lấy nó về và chèn vào đoạn mã nguồn của website, việc đó đồng nghĩa chúng ta đã tạo ra một cổng tương tác giữa các website đó và **Facebook**. Khi bạn share bài viết của một website nó sẽ hiện thông tin của bài đó trên trang **Facebook** cá nhân của bạn.



*Hình 2.2. Ứng dụng của API trong website*

**b. Ứng dụng trong Game**

Các hàm API dành để lập trình Game được giới thiệu trong bản MIDP 2.0, những hàm này là phương tiện để phát triển game với nhiều phần đồ họa. Các hàm API dành cho Game là một phần của gói **javax.microedition.lcdui.game**

Lớp GameCanvas gần như giống lớp Canvas, GameCanvas cung cấp nền tảng để tạo giao diện người dùng, nhưng trong trường hợp này chỉ cho việc tạo games. GameCanvas chứa một vùng nhớ tách rời với vùng nhớ màn hình cho mỗi thể hiện và cung cấp các phương tiện tích hợp để xác định tình trạng các phím trò chơi.

1. **Ứng dụng trong Quảng Cáo**

API quảng cáo trên Facebook là một phần trong nhóm API Facebook lớn hơn của chúng tôi và cho phép bạn tạo cũng như quản lý quảng cáo trên Facebook theo kế hoạch.

Thông tin chung về việc sử dụng API Tiếp thị, quyền truy cập, lập phiên bản, v.v. Trường hợp sử dụng chính của API Tiếp thị là thông tin chi tiết về quảng cáo, quản lý đối tượng và quản lý quảng cáo:

**Quản lý đối tượng**: quản lý dữ liệu bên thứ nhất của bạn, tìm khách hàng mới và thu hút lại khách hàng hiện có thông qua nhắm mục tiêu Đối tượng tùy chỉnh trên Facebook. Bắt đầu trên API Quản lý đối tượng và Tham khảo đối tượng tùy chỉnh.

**API Quản lý quảng cáo**: Bộ API đầy đủ, từ tạo và quản lý quảng cáo, báo cáo và quản lý đối tượng. Bắt đầu trên API Quản lý quảng cáo và Hướng dẫn API Quản lý quảng cáo.

## 2.4 Tìm hiểu về FACEBOOK API

### 2.4.1. Facebook API là gì

Facebook API là một nền tảng để xây dựng những ứng dụng cho các thành viên của mạng xã hội Facebook. API cho phép các ứng dụng sử dụng các kết nối xã hội và các thông tin hồ sơ để làm cho các ứng dụng liên quan tới nhau nhiều hơn, và để phổ biến những hoạt động tới nguồn cung cấp tin và trang hồ sơ của Facebook, tùy thuộc vào cài đặt cá nhân của người dùng. API cung cấp những lời gọi để lấy thông tin về người sử dụng, nhóm người dùng, bạn bè, thông báo, sự kiện và nguồn cung cấp dữ liệu…Cũng có thể dùng lời gọi API để cập nhật và lấy thông tin hồ sơ.

API sử dụng giao thức RESTful và các hồi đáp được trả lại dưới dạng XML.

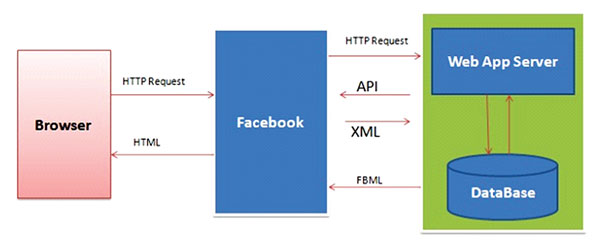
Có 3 ứng dụng được sử dụng nhiều nhất của Facebook API.

* **Facebook javascript SDK**: Hỗ trợ lập trình bằng ngôn ngữ javascript dành cho các nhà phát triển.
* **Facebook social plugin**: Tích hợp facebook like, share, comment, fanpage vào website. Tại giao diện website, người dùng tương tác fanpage, truy xuất hình ảnh và nội dung bạn đã post trước đây.
* **Facebook login**: Đăng nhập vào tài khoản website, forum thay vì phải khai báo như truyền thống.

### 2.4.2. Cách thức hoạt động của Facebook API

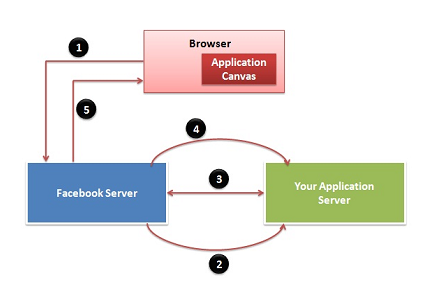
**API** là nền tảng do Facebook cung cấp cho người viết ứng dụng để dễ dàng trong việc tạo ứng dụng và đảm bảo người viết ứng dụng không can thiệp quá sâu vào hệ thống của Facebook. API cho phép thực thi nền tảng thông qua các phương thức được định nghĩa. Thông qua các lời gọi API, người tạo ứng dụng có thể lấy thông tin user, groups, photo…mà họ cần.

Facebook gửi phương thức POST tới máy chủ Facebook API. Nó bao gồm một số các thông số yêu cầu như api\_key của ứng dụng.Sesion\_key của người dùng đưa ra yêu cầu.Bên cạnh đó Facebook còn thêm vào tham số fb\_sig để thông báo ứng dụng đưa ra yêu cầu, bằng cách này tất cả các lời gọi API sẽ được đảm bảo, Facebook có thể xác minh các yêu cầu được gửi từ một ứng dụng đã được chấp thuận. Thông tin mà Facebook sẽ trả lại là một tài liệu XML.



*Hình 2.3 Cách thức hoạt động của Facebook API*

### 2.4.3. Cách thức làm việc của Facebook API



*Hình 2.4 Cách thức làm việc của Facebook API*

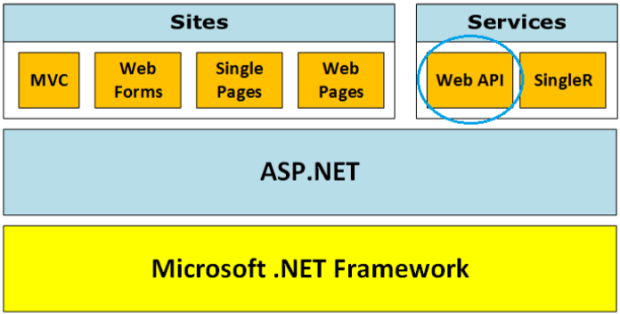
* Trình duyệt của người dùng yêu cầu <http://apps.facebook.com/>.... Địa chỉ này chỉ tới một cụm máy chủ trong trung tâm dữ liệu của Facebook. Những server này sẽ phân tích các yêu cầu, xác định các ứng dụng tương ứng, sau đó tìm kiếm Url callback mà nhà phát triển ứng dụng cung cấp và thực hiện gọi tới Url đó.
* Máy chủ của Facebook sẽ gửi yêu cầu tới máy chủ của người tạo ứng dụng. Yêu cầu này người dùng hoàn toàn không biết.
* Server chứa ứng dụng tạo một lời gọi API tới server của Facebook bằng cách sử dụng FQL thông qua phương thức fql.query() của API hoặc trực tiếp bằng các hàm API.
* Máy chủ chứa ứng dụng sẽ trả về FBML tới máy chủ của Facebook. Kết quả đạt được cuối cùng là 1 tài liệu FBML. Tài liệu này sẽ được gửi trả lại server của Facebook để thực thi.
* Tài liệu FBML sẽ được chuyển sang dạng HTML và phục vụ cho người sử dụng. Đây là bước cuối cùng trong quá trình thực thi và trả về kết quả cho trình duyệt.

## 2.5 Tìm hiểu về WEB API

### 2.5.1. Web API là gì

Web API là **công nghệ khá mới của hãng Microsoft để xây dựng dịch vụ thành phần phân tán.** ASP.NET Web Api là một framework giúp cho việc xây dựng HTTP service một cách dễ dàng. Chúng có thể phát triển cho nhiều clients khác nhau như trình duyệt, mobile app. Web api là một nền tảng để phát triển các ứng dụng dựa trên Restful service trong .Net

Trên nền tảng Net 4.0 trở lên có xuất hiện một công nghệ mới đó là tạo Webservices bằng Web Api. Web giúp xây dựng Rest full services trên nền tảng .net, nó tự động trả dữ liệu có kiều phù hợp: Json, XML.Web Api sẽ giúp bạn chuyển kiểu dữ liệu trả về theo thuộc tính Accept trong header của Request và bạn không cần quan tâm đến điều đó.



*Hình 2.5 Mô hình ASP.NET API*

* **Đặc điểm của Web API (.NET 4.0 trở lên):**
* Đây là một framework mới giúp cho việc xây dựng các HTTP service rất đơn giản và nhanh chóng.
* Open Source và có thể được sử dụng bởi bất kì client nào hỗ trợ XML, JSON
* Hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content formats
* Có thể host trong ứng dụng hoặc trên IIS
* Kiến trúc lý tưởng cho các thiết bị có băng thông giới hạn như smartphone, tablet
* Định dạng dữ liệu có thể là JSON, XML hoặc một kiểu dữ liệu bất kỳ
* **Ưu điểm của Web API:**
* Cấu hình đơn giản hơn nhiều so với WCF.
* Hiệu suất (performance) cao.
* Hỗ trợ RESTful đầy đủ. Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test, …
* Mã nguồn mở (Open source).

### 2.5.2. RESTful Service

REST là viết tắt của cụm từ Representational State Transfer (đôi khi còn được viết là ReST) là một kiểu kiến trúc được sử dụng trong việc giao tiếp giữa các máy tính (máy tính cá nhân và máy chủ của trang web) trong việc quản lý các tài nguyên trên internet. REST được sử dụng rất nhiều trong việc phát triển các ứng dụng Web Services sử dụng giao thức HTTP trong giao tiếp thông qua mạng internet. Các ứng dụng sử dụng kiến trúc REST này thì sẽ được gọi là ứng dụng phát triển theo kiểu RESTful.

RESTful Service: Là một dịch vụ web đơn giản sử dụng HTTP và tính chất của REST.

Nó tuân thủ theo 4 nguyên tắc thiết kế cơ bản sau:

* Sử dụng các phương thức HTTP một cách rõ ràng.
* Phi trạng thái (Stateless)
* Hiển thị cấu trúc thư mục như URI
* Chuyển đổi linh hoạt JavaScript Object Notation (JSON) và XML hoặc cả hai.

### 2.5.3. Nguyên tắc cơ bản để tạo ra RESTful Service

### a. Sử dụng các phương thức HTTP một cách rõ ràng.

**REST** đặt ra một quy tắc thông qua các phương thức của **HTTP**. Thông thường ý định đó bao gồm lấy dữ liệu, chèn dữ liệu, cập nhập dữ liệu hoặc xóa dữ liệu. Vậy khi bạn muốn thực hiện một trong các ý định trên hãy lưu ý các quy tắc sau:.

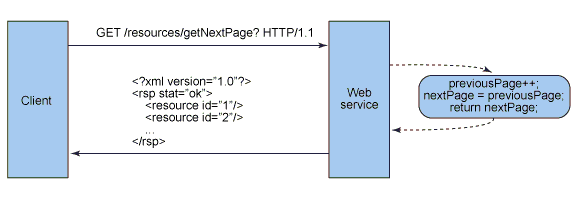
* Để tạo một tài nguyên trên máy chủ, bạn cần sử dụng phương thức **POST(**HttpPost).
* Để truy xuất một tài nguyên, sử dụng **GET(**HttpGet).
* Để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó, sử dụng **PUT(**HttpPut).
* Để huỷ bỏ hoặc xoá một tài nguyên, sử dụng **DELETE(**HttpDelete).

Chú ý rằng các nguyên tắc ở trên là không bắt buộc, thực tế bạn có thể sử dụng phương thức **GET** để yêu cầu lấy dữ liệu, chèn, sửa hoặc xóa dữ liệu trên Server. Tuy nhiên **REST** đưa ra các nguyên tắc ở trên mục đích đưa mọi thứ trở lên rõ ràng và dễ hiểu.

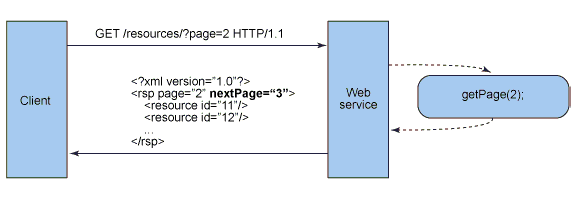
### Phi trạng thái (Stateless).

Có nghĩa là nó không lưu giữ thông tin của client. Chẳng hạn bạn vừa gửi yêu cầu để xem trang thứ 2 của một tài liệu, và bây giờ bạn muốn xem trang tiếp theo (sẽ là trang 3). **REST** không lưu trữ lại thông tin rằng trước đó nó đã phục vụ bạn trang 2. Điều đó có nghĩa là **REST** không quản lý phiên làm việc (Session).

Hình dưới đây minh họa một ứng dụng có lưu trữ trạng thái, nó biết người dùng đang xem trang số mấy. Và người dùng chỉ cần yêu cầu ***"Trang Tiếp theo"*** để nhận được trang mong muốn.



Với các thiết kế phi trạng thái, Client phải gửi yêu cầu rõ ràng, bao gồm số thự tự của trang cần xem.



Như vậy, các thành phần máy chủ phi trạng thái ít phức tạp hơn để thiết kế, viết và phân bổ thông qua máy chủ được cân bằng tải. Dịch vụ phi trạng thái không chỉ hoạt động tốt hơn, nó còn chuyển hầu hết vai trò duy trì trạng thái sang ứng dụng ở máy khách. Trong một dịch vụ mạng **RESTful**, máy chủ chịu trách nhiệm đưa ra các phản hồi và cung cấp một cách thức cho phép Client duy trì trạng thái ứng dụng của chính nó.

### Hiển thị cấu trúc thư mục như URI.

**REST** đưa ra một cấu trúc để người dùng có thể truy cập vào tài nguyên của nó thông qua các **URL**, tài nguyên ở đây là tất cả những cái mà bạn có thể gọi tên được (Video, ảnh, báo cáo thời tiết,..) Bạn cần tạo ra các REST serivce để nó trả về cho người dùng các nguồn tài nguyên tương ứng.

Các địa chỉ **REST service** cần phải thật trực quan đến mức người dùng dễ đoán. Hãy nghĩ về một địa chỉ (URI) giống như một gợi ý rõ ràng, dễ đoán rằng nó đang trỏ tới cái gì và cung cấp tài nguyên gì. Tóm lại, cấu trúc của một URI nên được đơn giản, có thể dự đoán, và dễ hiểu.

Cấu trúc địa chỉ của RESTful service:

* Giấu các đuôi tài liệu mở rộng của bản gốc trong máy chủ (.jsp, .php, .asp).
* Để mọi thứ là chữ thường (thực ra là không phân biệt, nhưng cũng nên tuân thủ để khỏi phải nhớ HOA-thường lung tung).
* Thay thế các khoảng trống bằng gạch chân hoặc gạch nối (một trong hai loại).
* Tránh các chuỗi yêu cầu.
* Thay vì sử dụng mã (404 Not Found) khi yêu cầu địa chỉ cho một phần đường dẫn thì luôn luôn cung cấp một trang mặc định hoặc tài nguyên như một phản hồi.

### Chuyển đổi JSON, XML hoặc cả hai.

Khi Client gửi một yêu cầu tới web service nó thường được truyền tải dưới dạng XML hoặc JSON và thông thường nhận về với hình thức tương tự.

Đôi khi Client cũng có thể chỉ định kiểu dữ liệu nhận về mà nó mong muốn (JSON, hoặc XML,..), các chỉ định này được gọi là các kiểu MINE, nó được gửi kèm trên phần HEADER của request.

Dưới đây là các kiểu MINE phổ biến thường sử dụng với **REST** service.

| **MIME-Type** | **Content-Type** |
| --- | --- |
| **JSON** | application/json |
| **XML** | application/xml |
| **XHTML** | application/xhtml+xml |

* Là một bản tóm tắt các thuộc tính của những thứ trong mô hình dữ liệu hệ thống.
* Định dạng dữ liệu mà ứng dụng và trao đổi dịch vụ trong mức đáp ứng yêu cầu/ phản hồi hoặc trong phần thân của HTTP.
* Các chủ thể trong mô hình dữ liệu có liên quan với nhau.
* Cấu trúc dịch vụ sao cho nó tận dụng được phần đầu chấp nhận HTTP có sẵn bên trong – một loại MIME

2.6. Facebook Platform là gì?

Facebook Platform cung cấp một framework cho người phát triển để tạo ra các ứng dụng tương tác và tích hợp các tính năng cốt lõi của Facebook. Được giới thiệu vào ngày 24/05/2007 và đến ngày 09/12/2009 đã có hơn 500.000 ứng dụng hoạt động trên Facebook Platform. Facebook platform bao gồm 4 thành phần: FBML, API,FQL và FBJS.

Facebook Platform hỗ trợ hai kiểu OAuth 2.0 cho việc đăng nhập người dùng: **Server-side** và **Client-side**. Server-side được sử dụng trong trường hợp cần gọi tới Graph API từ client, như JavaScript trên trình duyệt hoặc ứng dụng mobile.

Facebook Oauth 2.0 diễn ra qua 3 bước sau: User authentication, App Authorization và App authentication.

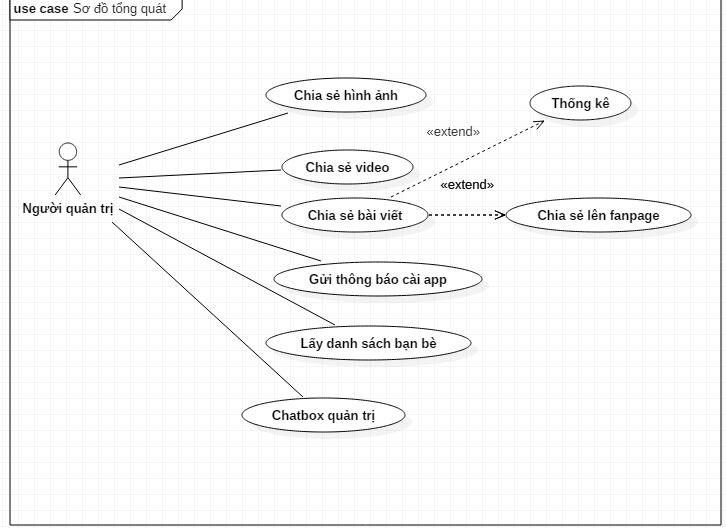
CHƯƠNG III: PHÂN TÍCH THIẾT KẾ VỚI UML

**3.1 Biểu đồ usecase**

### 3.1.1. Biểu đồ phân cấp chức năng

**a. Biểu đồ phân cấp chức năng là gì ?**

Là một loại biểu đồ diễn tả sự phân rã dần dần các chức năng từ đại thể đến chi tiết. Mỗi nút trong biểu đồ là một chức năng, và quan hệ duy nhất giữa các chức năng, diễn tả bởi các cung nối liền các nút, là quan hệ bao hàm. như vậy biểu đồ phân cấp chức năng tạo thành một cấu trúc cây



*Hình 3.1: Sơ đồ phân cấp chức năng tổng quát*

**b**. **Đặc điểm của biểu đồ phân cấp chức năng**

- Cho một cách nhìn khái quát, dễ hiểu, từ đại thể đến chi tiết về các chức năng, nhiệm vụ cần thực hiện (thường ở mức diễn tả logic)

- Rất dễ thành lập, bằng cách phân rã dần các chức năng từ trên xuống

- Có tính chất tĩnh, bởi chúng chỉ cho thấy các chức năng mà không cho thấy trình tự xử lý.

- Thiếu vắng sự trao đổi thông tin giữa các chức năng.

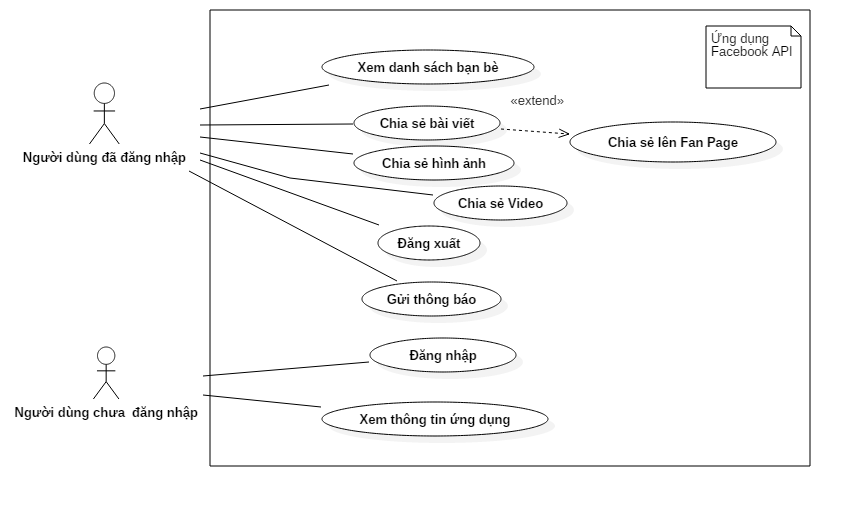
Vì những lý do trên nên biểu đồ phân cấp chức năng thường được sử dụng làm mô hình chức năng trong bước đầu phân tích, hoặc cho các hệ thống đơn giản. Nếu hệ thống phức tạp thì biểu đồ phân cấp chức năng là quá sơ lược và còn thiếu sót nêu trên nên không thể châm trước được. Khi đó chúng ta thường dùng biểu đồ luồng dữ liệu.

* **Danh sách các Actor**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Ý nghĩa** |
| 1 | Người dùng  đã đăng nhập | Hiển thị màn hình quản trị trong tài khoản nhà phát triển ứng dụng |
| 2 | Người dùng  chưa đăng nhập | Chir xem thông tin trang front end ở bên ngoài |

* **Danh sách các biểu đồ tuần tự**

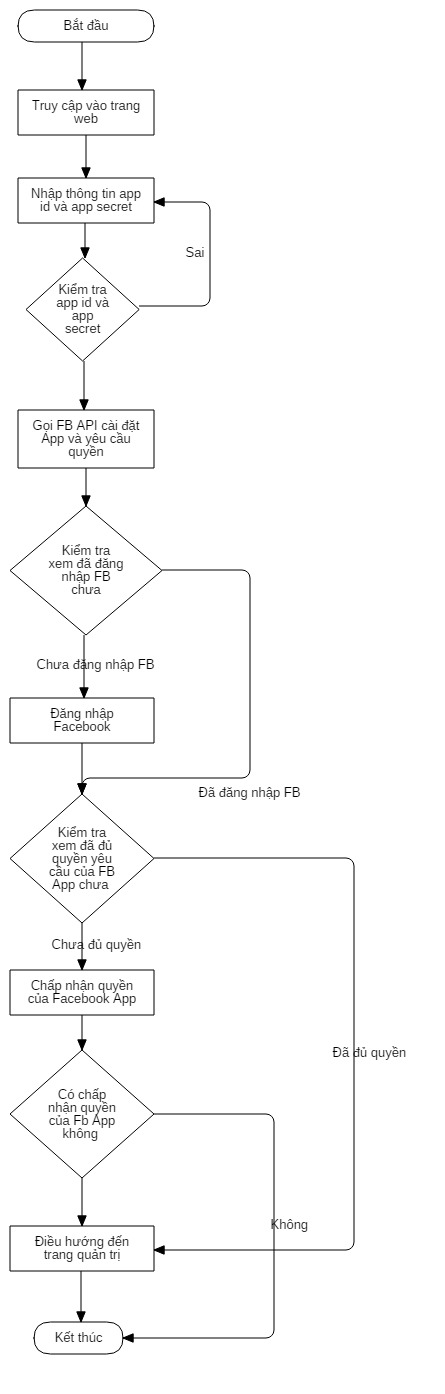
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Sơ đồ**  **Tuần tự** | **Ý nghĩa** |
| 1 | Đăng nhập | UC này mô tả chức năng đăng nhập vào hệ thống của người dùng |
| 2 | Chia sẻ bài viết | UC này mô tả chức năng chia sẻ bài viết của người dùng |
| 3 | Chia sẻ hình ảnh | UC này mô tả chức năng chia sẻ hình ảnh của người dùng |
| 4 | Gửi thông báo | UC này mô tả chức năng gửi thông báo đến danh sách bạn bè của người dùng |
| 4 | Đăng xuất | UC này mô tả chức năng đăng xuất khỏi hệ thống của người dùng |
| 5 | Xem danh sách bạn bè | UC này mô tả chức năng xem danh sách bạn bè của người dùng |
| 6 | Xem thông tin ứng dụng | UC này mô tả chức năng xem thông tin ứng dụng tiện ích của người dùng |



*Hình 3.2: Usecase Ứng dụng Facebook API*

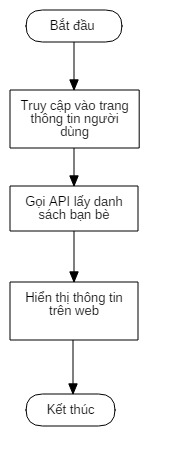
**3.2. Biểu đồ luồng xử lý**

### 3.2.1. Chức năng người dùng đăng nhập



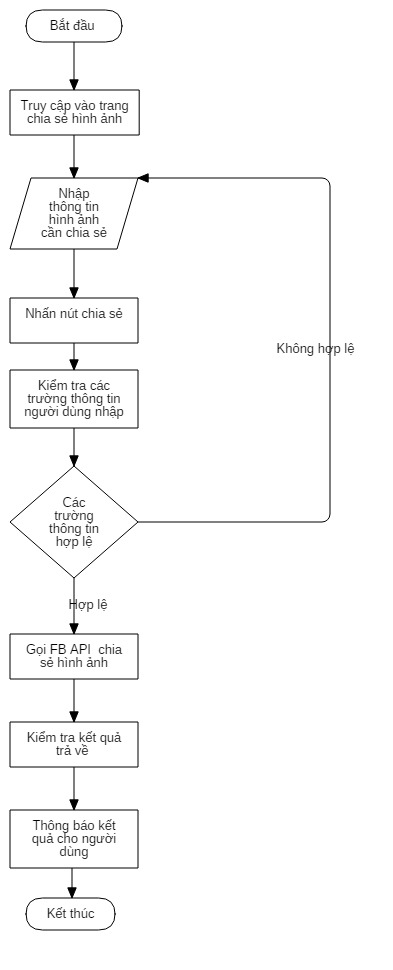
*Hình 3.3: Luồng xử lý Chức năng đăng nhập*

### 3.2.2. Chức năng người dùng Chia sẻ bài viết



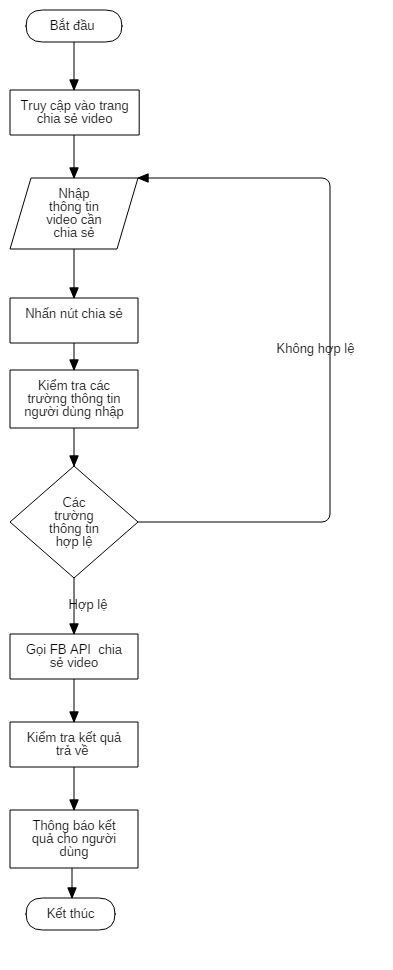
*Hình 3.4: Luồng xử lý Chức năng Chia sẻ bài viết*

### 3.2.3. Chức năng người dùng Chia sẻ Hình ảnh



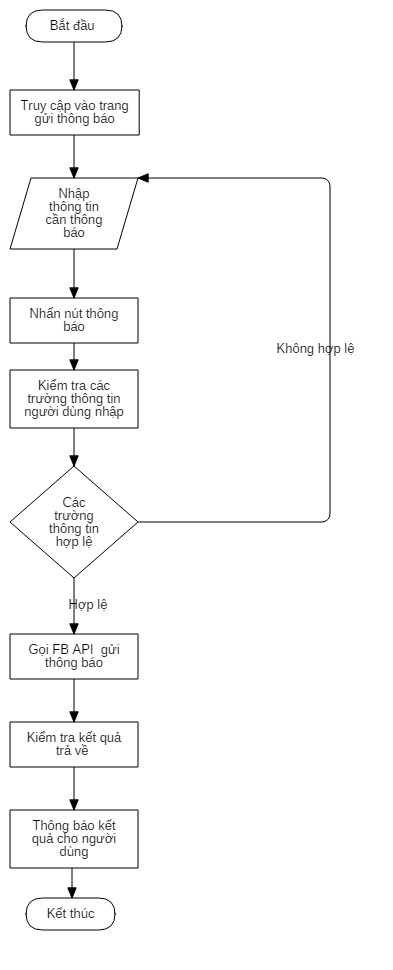
*Hình 3.5: Luồng xử lý Chức năng Chia sẻ Hình ảnh*

### 3.2.4. Chức năng người dùng Chia sẻ Video



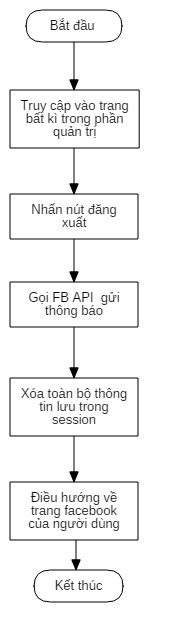
*Hình 3.6: Luồng xử lý Chức năng Chia sẻ Video*

### 3.2.5. Chức năng người dùng Gửi thông báo



*Hình 3.7: Luồng xử lý Chức năng Gửi thông báo*

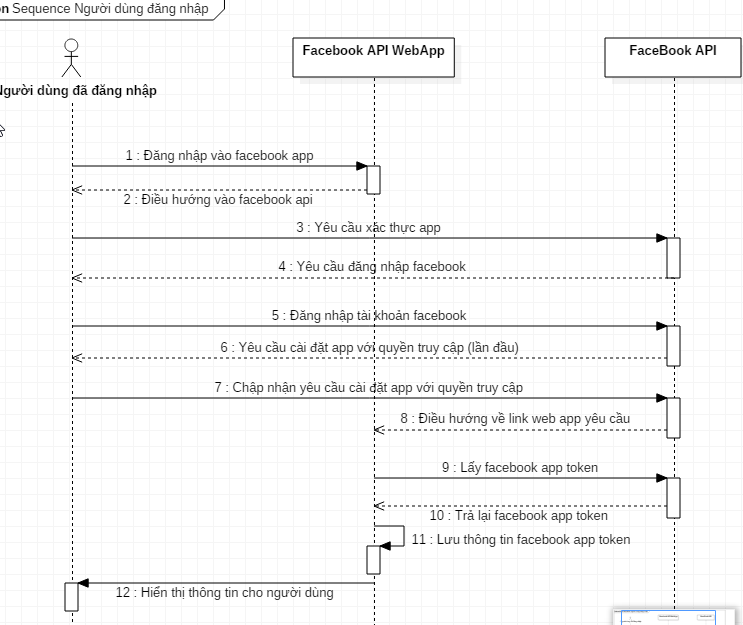
### 3.2.6. Chức năng người dùng đăng xuất



*Hình 3.8: Luồng xử lý Chức năng Đăng xuất*

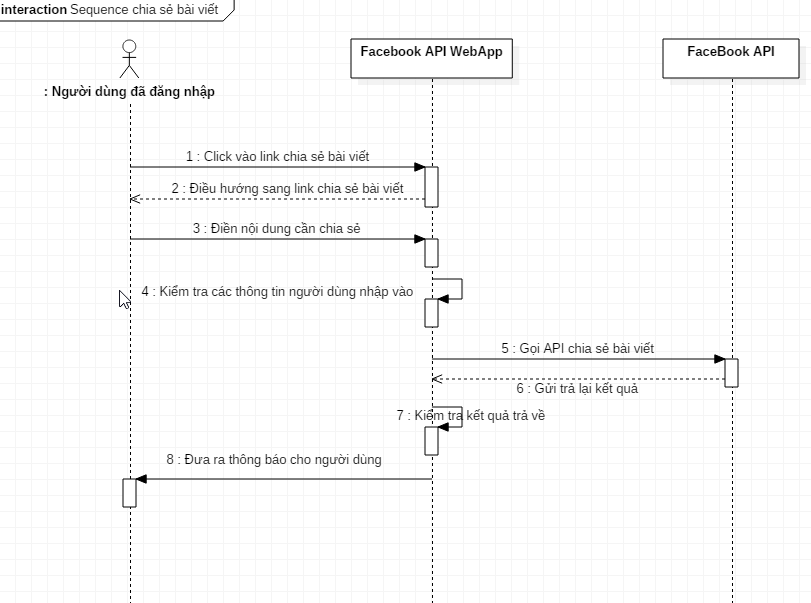
**3.3. Biểu đồ tuần tự**

### 3.3.1. Sơ đồ tuần tự “Đăng nhập”



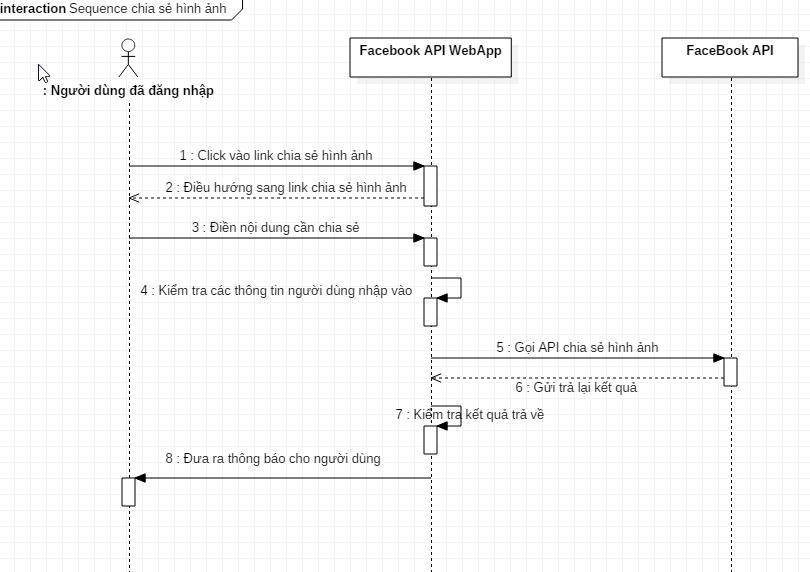
*Hình 3.8: Sơ đồ tuần tự Đăng nhập*

### 3.3.2. Sơ đồ tuần tự “Chia sẻ bài viết”



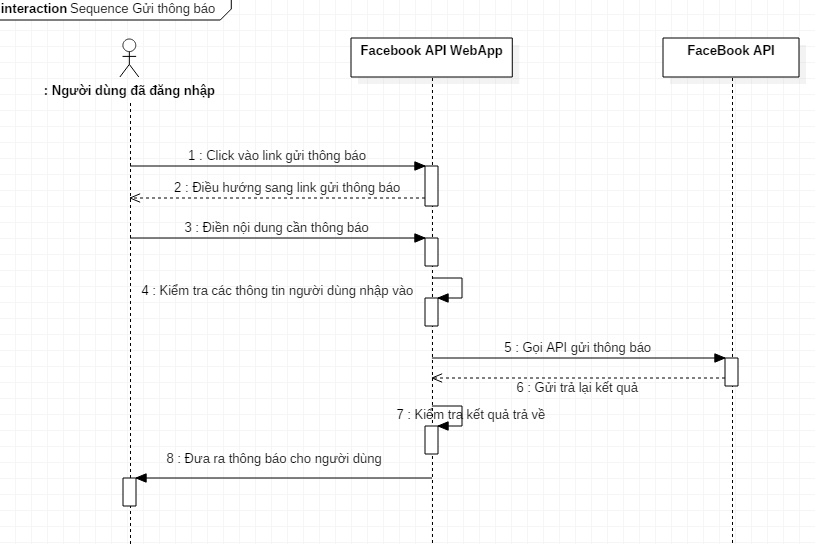
*Hình 3.9: Sơ đồ tuần tự Chia sẻ bài viết*

### 3.3.3. Sơ đồ tuần tự “Chia sẻ hình ảnh”



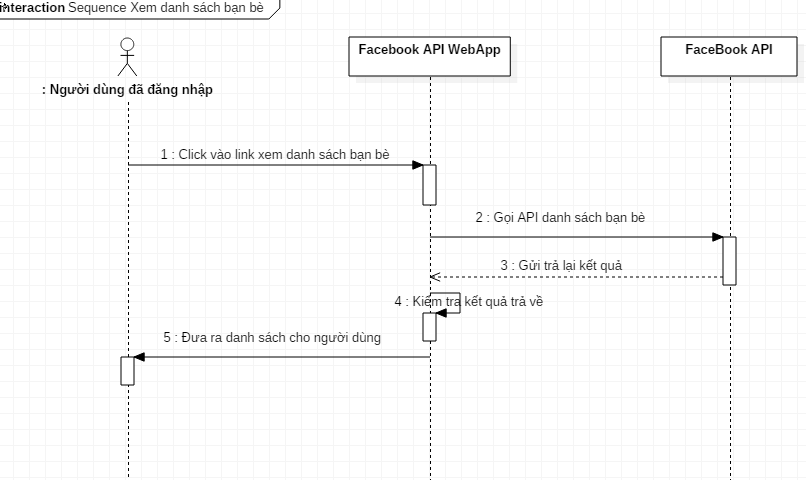
*Hình 3.10: Sơ đồ tuần tự Chia sẻ hình ảnh*

### 3.2.4. Sơ đồ tuần tự “gửi thông báo”



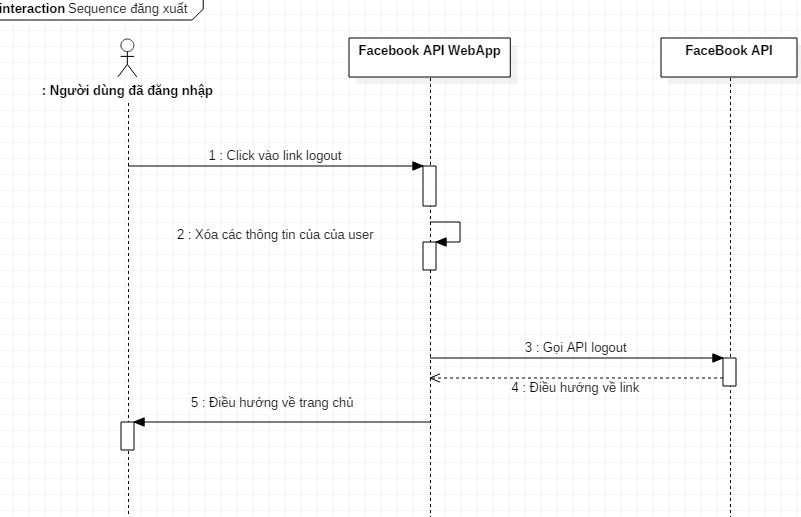
*Hình 3.11: Sơ đồ tuần tự Gửi thông báo*

### 3.2.5. Sơ đồ tuần tự “Xem danh sách bạn bè”



*Hình 3.12: Sơ đồ tuần tự Xem danh sách bạn bè*

### 3.2.6. Sơ đồ tuần tự “Đăng xuất”



*Hình 3.13: Sơ đồ tuần tự Đăng xuất*

CHƯƠNG IV: THIẾT KẾ GIAO DIỆN VÀ KIỂM THỬ

## 4.1. Lựa chọn môi trường cài đặt

* Công cụ phát triển Visual Studio 2013
* Môi trường phát triển .Net Framework 4.5
* Ngôn ngữ lập trình ASP.NET.
* Thiết kế giao diện dùng Bootstrap.
* Sử dụng bộ cài đặt Facebook SDK

### 4.1.1. Công nghệ .Net và Microsoft Visual Studio 2013

### a. Microsoft .NET

* Thư viện lớp .NET Framework là một tập hợp những kiểu dữ liệu được dùng lại và được kết hợp chặt chẽ với Comman Language Runtime. Thư viện lớp là hướng đối tượng cung cấp những kiểu dữ liệu là mã nguồn được quản lý của chúng ta có thể dẫn xuất. Điều này không chỉ làm cho những kiểu dữ liệu của .NET Framework dễ sử dụng mà còn làm giảm thời gian liên quan đến việc học đặc tính mới của .NET Framework. Thêm vào đó, các thành phần của các hãng thứ ba có thể tích hợp với những lớp trong .NET Framework.
* Cũng như mong đợi của người phát triển với thư viện lớp hướng đối tượng, kiểu dữ liệu. NET Framework cho phép người phát triển thiết lập nhiều mức độ thông dụng của việc lập trình, bao gồm các nhiệm vụ như: quản lý chuỗi, thu thập hay chọn lọc dữ liệu, kết nối với cơ cở dữ liệu, và truy cập tập tin. Ngoài những nhiệm vụ thông dụng trên. Thư viện lớp còn đưa vào những kiểu dữ liệu để hỗ trợ cho những kịch bản phát triển chuyên biệt khác. Ví dụ người phát triển có thể sử dụng .NET Framework để phát triển những kiểu ứng dụng và dịch vụ như sau:
* Ứng dụng Console.
* Ứng dụng giao diện GUI trên Windows (Windows Forms).
* Ứng dụng ASP.NET.
* Dịch vụ XML Web.
* Dịch vụ Windows.
* Trong đó những lớp Windows Forms cung cấp một tập hợp lớn các kiểu dữ liệu nhằm làm đơn giản việc phát triển các ứng dụng GUI chạy trên Windows. Còn nếu như viết các ứng dụng ASP.NET thì có thể sử dụng các lớp Web Forms trong thư viện .NET Framework.

### b. Công cụ Visual Studio 2013

* Microsoft Visual Studio. Net là một bộ sản phẩm gồm: VB.NET, C#, Visual C++.NET và Visual J#.NET. Thêm vào đó là Integrated Development Environment (IDE) giúp lập trình dễ đàng, thoải mái. IDE không những cung cấp mọi công cu lập trình cần thiết mà còn giúp kiểm tra nguồn mã hay tạo giao diện Windows trực quan, truy tìm các tập tin liên hệ đến dự án.
* VS 2013 được cài đặt trên hệ điều hành : Có sẵn Internet Explorer 10 và được hỗ trợ trên Windows 7 , Server 2008 R2 , Windows 8 , Server 2012 , Windows 8.1 và Server 2012 R2 ….
* Bộ xử lý xung nhịp 1.6 GHz hoặc cao hơn
* 1 GB RAM (1.5 GB RAM nếu chạy trên máy ảo)
* Ổ cứng còn trống 10 GB
* Tốc độ của ổ đĩa cứng 5400 RPM
* Độ phân giải màn hình tối thiểu 1024 x 768, hỗ trợ Direct X 9
* Kết nối Internet
* **Môi trường phát triển .Net framework 4.5**

.NET Framework 4.5 chỉ được hỗ trợ trên Windows Vista hoặc mới hơn.

.NET Framework 4.5 sử dụng Common Language Runtime 4.0, với một số tính năng thời gian chạy bổ sung.

.NET Framework 4.5 được hỗ trợ trên Windows Vista , Server 2008 , Windows 7 , Server 2008 R2 , Windows 8 , Server 2012 , Windows 8.1 và Server 2012 R2 .

Các ứng dụng sử dụng .NET Framework 4.5 cũng sẽ chạy trên các máy tính với .NET Framework 4.6 được cài đặt, hỗ trợ hệ điều hành bổ sung.

.NET Framework 4.5 chạy thông qua ngôn ngữ ASP.NET:

* Hỗ trợ các loại biểu mẫu HTML5 mới .
* Hỗ trợ cho các mô hình kết nối trong Web Forms . Các tính năng này cho phép bạn ràng buộc kiểm soát dữ liệu trực tiếp với các phương pháp truy cập dữ liệu và tự động chuyển đổi dữ liệu người dùng nhập vào và từ các loại dữ liệu . NET Framework .
* Hỗ trợ JavaScript không phô trương trong các kịch bản xác thực phía máy khách.
* Cải thiện xử lý của khách hàng kịch bản thông qua bó và minification để cải thiện hiệu suất trang.
* Các chương trình mã hoá tích hợp từ thư viện Anti-XSS (trước đây là thư viện bên ngoài) để bảo vệ chống lại các cuộc tấn công mã chéo .
* Hỗ trợ giao thức WebSocket .
* Hỗ trợ đọc và viết các yêu cầu và phản hồi HTTP không đồng bộ .
* Hỗ trợ cho các module không đồng bộ và xử lý.
* Hỗ trợ mạng lưới phân phối nội dung (CDN) dự phòng trong kiểm soát ScriptManager.

### 4.1.2. Ngôn ngữ ASP.NET

* ASP.NET là một nền tảng ứng dụng web (web application framework) được phát triển và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web.
* ASP.NET được biên dịch dưới dạng Common Language Runtime (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi .NET language.
* Xử lý lỗi time (thời gian) chạy được cải thiện đáng kể, sử dụng xử lý ngoại lệ bằng cách sử dụng các khối try –catch.
* ASP.NET sử dụng các khả năng đa ngôn ngữ của .NET Common Language Runtime (CLR), cho phép các trang web được mã hóa trong VB.NET, C#,J#...
* Khả năng cache toàn bộ trang hoặc chỉ là một phần của nó để cải thiện hiệu suất.
* Khả năng sử dụng đúng thiết kế hướng đối tượng cho các trang lập trình và điều khiển.

### 4.1.3. Ngôn ngữ C#

* C# là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường.Microsoft phát triển C# dựa trên C++ và Java. C# là ngôn ngữ có được sự cân bằng giữa C++, Visuabasic, Dephi và Java.
* C# hỗ trợ giao diện interface, nó được xem như một cam kết với một lớp cho những dịch vụ mà giao diện quy định. Trong ngôn ngữ C#, một lớp chỉ có thể kế thừa từ duy nhất một lớp cha, tức là không cho đa kế thừa như trong ngôn ngữ C++, tuy nhiên một lớp có thể thực thi nhiều giao diện.Khi một lớp thực thi một giao diện thì nó sẽ cung cấp chức năng thực thi giao diện.
* Ngôn ngữ C# cung cấp những đặc tính hướng thành phần (component-oriented), như là những thuộc tính, những sự kiện.Lập trình hướng thành phần được hỗ trợ bởi CLR cho phép lưu trữ metadata với mã nguồn cho một lớp.Metadata mô tả cho một lớp, bao gồm những phương thức và những thuộc tính của nó, cũng như những sự bảo mật cần thiết và những thuộc tính khác.
* Mã nguồn chứa đựng những logic cần thiết để thực hiện những chức năng của nó. Do vậy, một lớp được biên dịch như là một khối self-contained, nên môi trường hosting biết được cách đọc metadata của một lớp và mã nguồn cần thiết mà không cần những thông tin khác để sử dụng nó.
* C# là một ngôn ngữ đơn giản: Loại bỏ những phức tạp có trong Java hay C++ như macro, template, đa kế thừa, virtual base class.
* Giống về diện mạo cú pháp C và C++ nhưng được cải tiến đơn giản hơn
* C# là một ngôn ngữ hiện đại: Có đầy đủ tính năng : Xử lý ngoại lệ, thu gom bộ nhớ tự động, kiểu dữ liệu an toàn, bảo mật mã nguồn.
* C# là ngôn ngữ hướng đối tượng: Đóng gói (Encapsulation), Kế thừa (Inheritance), Đa hình (polymorphism).
* C# là ngôn ngữ mạnh mẽ và mềm dẻo, ít từ khóa., hướng module.

### 4.1.4. Giao diện Bootstrap

* Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn
* Hỗ trợ thiết kế giao diện web đáp ứng chuẩn RWD (Responsive Web Desgin) để có thể chạy được trên tất cả các thiết bị như Desktop, Mobile hay Tablet.
* Giúp thiết kế giao diện website nhanh chóng nhờ các CSS class được xây dựng sẵn.
* 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.

### 4.1.5. Bộ cài đặt Facebook SDK

Facebook SDK chính là Facebook Client Library tương tự như Facebook PHP Library và Facebook JavaScript Library nhằm giúp nhà phát triển .NET nhanh chóng và dễ dàng tận dụng các tính năng của Facebook Platform. Bộ thư viện này được phát triển bởi cộng đồng và Microsoft.

Điểm bắt đầu chính là lớp API (Facebook.Rest.Api) thuộc Facebook.dll. Lớp này chứa Facebook REST API và cung cấp Interface để gọi các phương thức khác có sẵn trong Facebook API. Microsoft cũng cung cấp các ví dụ và công cụ giúp phát triển ứng dụng Facebook trên nhiều công nghệ của .NET gồm: ASP.NET, Silverlight, WPF và WinForms.

Bộ công cụ gồm các thành phần chính:

**\* Facebook.dll** – Đây là DLL chính được dùng bởi tất cả các ứng dụng, có tất cả các sự kiện giao tiếp với ứng dụng Facebook. DLL này cũng hỗ trợ ứng dụng XAML (Silverlight và WPF) cải thiện nền tảng Facebook trong việc liên kết và lưu đệm dữ liệu.

**\* Facebook.Silverlight.dll** – Đây là phiên bản hỗ trợ Silverlight của DLL chính được dùng bởi tất cả ứng dụng Silverlight, có tất cả sự kiện giao tiếp với ứng dụng Facebook. DLL này cũng hỗ trợ ứng dụng XAML (Silverlight và WPF) cải thiện nền tảng Facebook trong việc liên kết và lưu đệm dữ liệu. REST API trong DLL này chỉ hỗ trợ không đồng bộ (Asynchronous).

**\* Facebook.Web.dll** – DLL này được dùng trong ứng dụng Web, chức năng chính là hỗ trợ giao tiếp dữ liệu giữa ứng dụng Facebook và ứng dụng Web (cả FBML và IFrame).

**\* Facebook.Web.Mvc.dll** – Cung cấp hỗ trợ xây dứng ứng dụng dùng ASP.NET MVC. Được tách từ Facebook.Web.dll.

**\* Facebook.Winforms.dll** – Cung cấp hỗ trợ viết ứng dụng Facebook dùng Windows Form. Các API giao tiếp và các điều khiển hiển thị dữ liệu từ Facebook.

## 4.2. Hướng dẫn cài đặt

* Cài đặt .NET Framework 4.5: Download tại địa chỉ sau:

<https://www.microsoft.com/en-us/download/details.aspx?id=30653>

* Cài đặt Visual Studio 2013: Download tại địa chỉ sau:

<https://www.visualstudio.com/en-us/news/releasenotes/vs2013-community-vs>

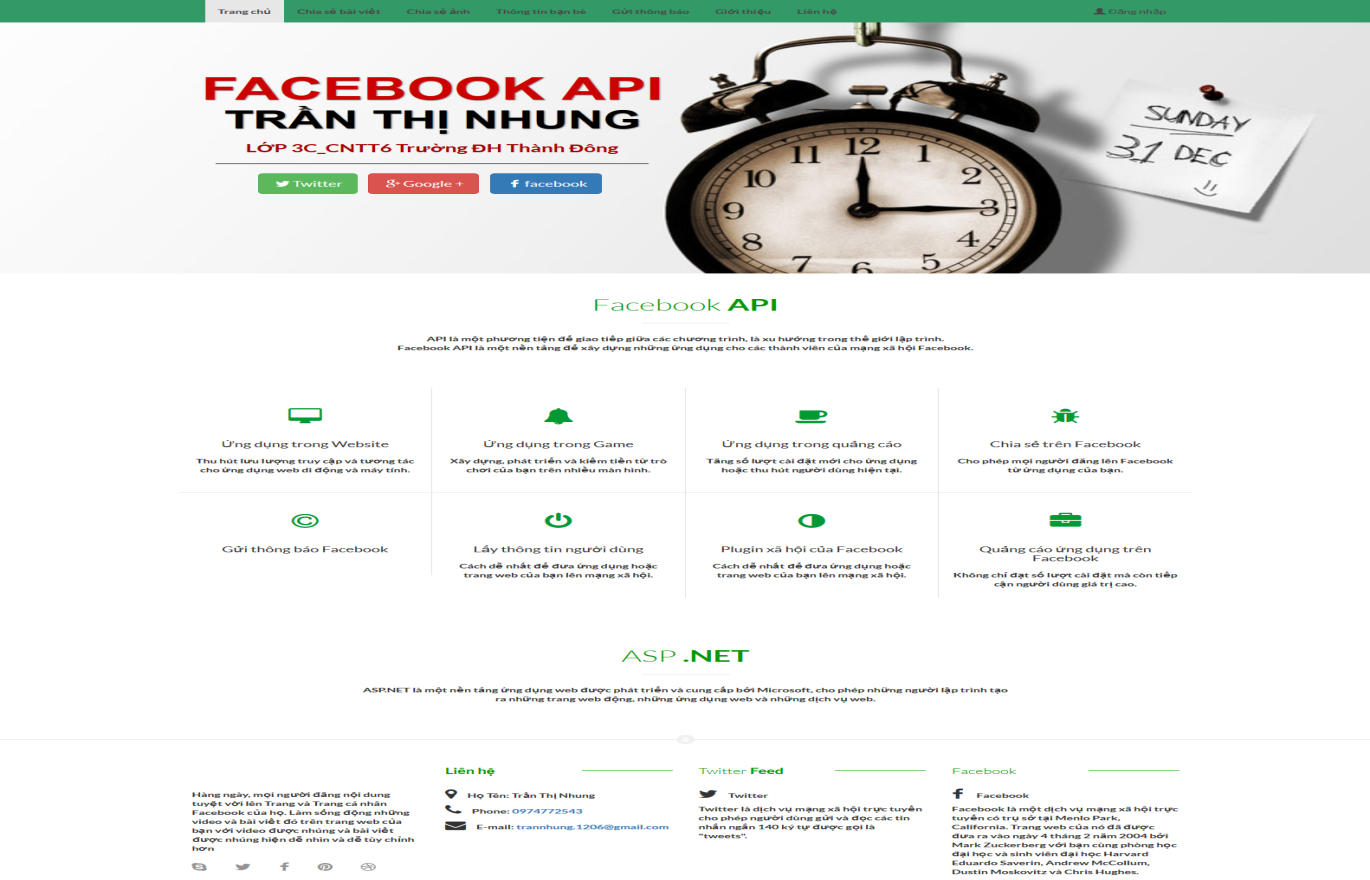
* Cài đặt và cấu hình Facebook SDK : Download tại địa chỉ sau:

<http://microsoft.github.io/winsdkfb/>

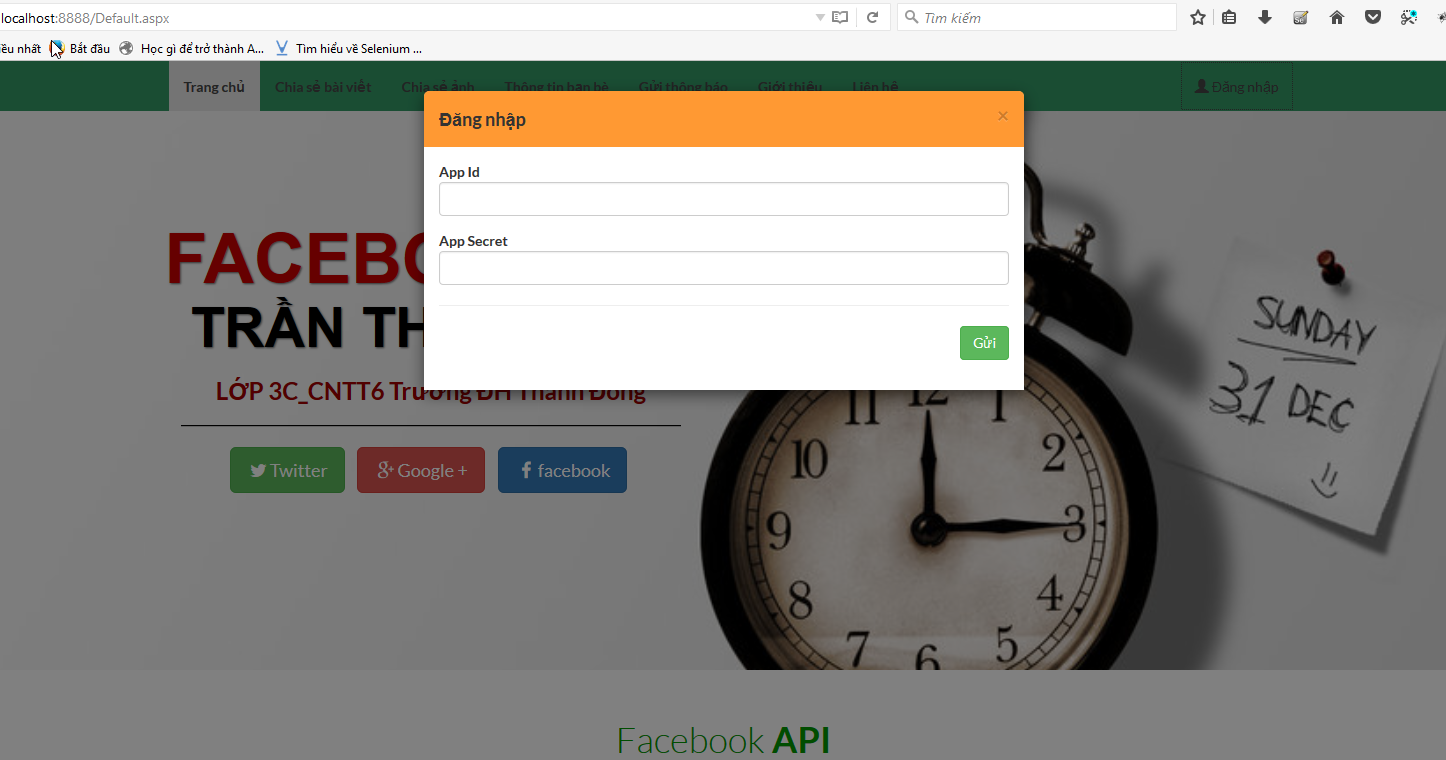
Hướng dẫn cài đặt Facebook SDK cho .NET:

<https://blogs.windows.com/buildingapps/2013/04/18/facebook-sdk-for-net-featured-in-the-facebook-technology-partner-program/>

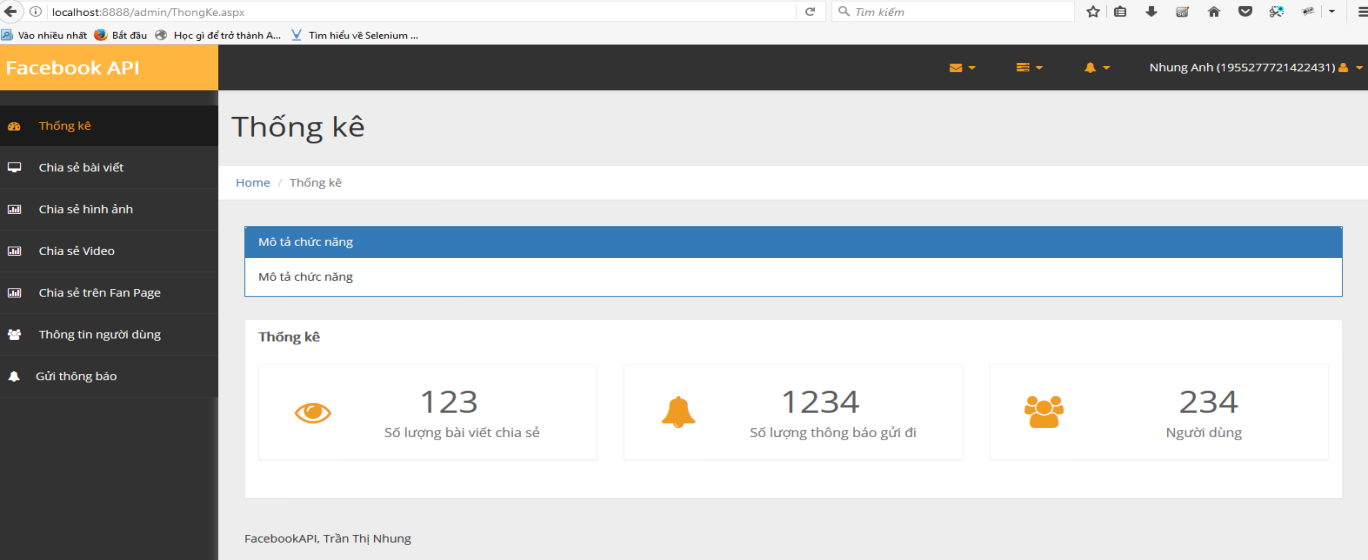
## 4.3. Hướng dẫn cách sử dụng



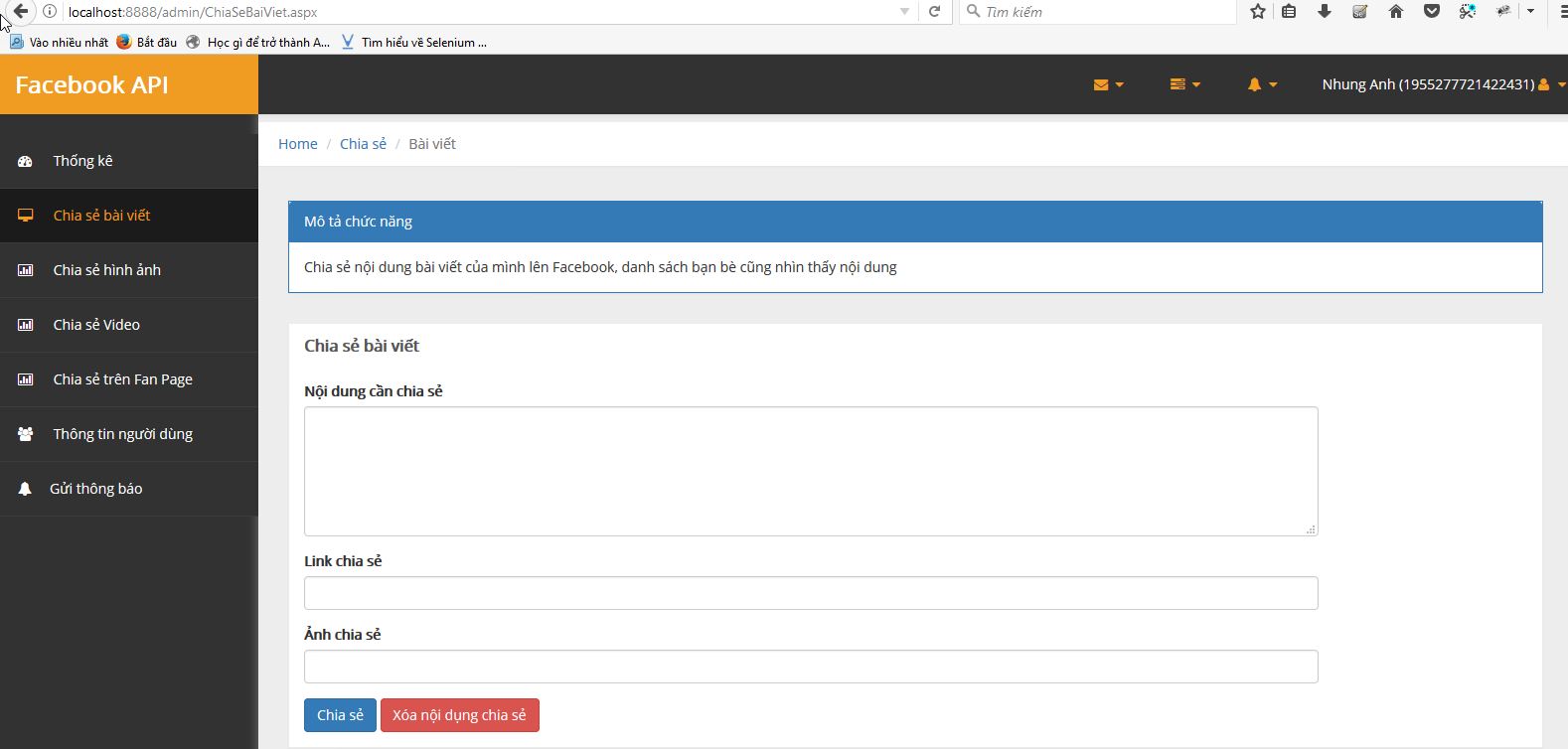
*Hình 4.1. Màn hình Trang chủ*



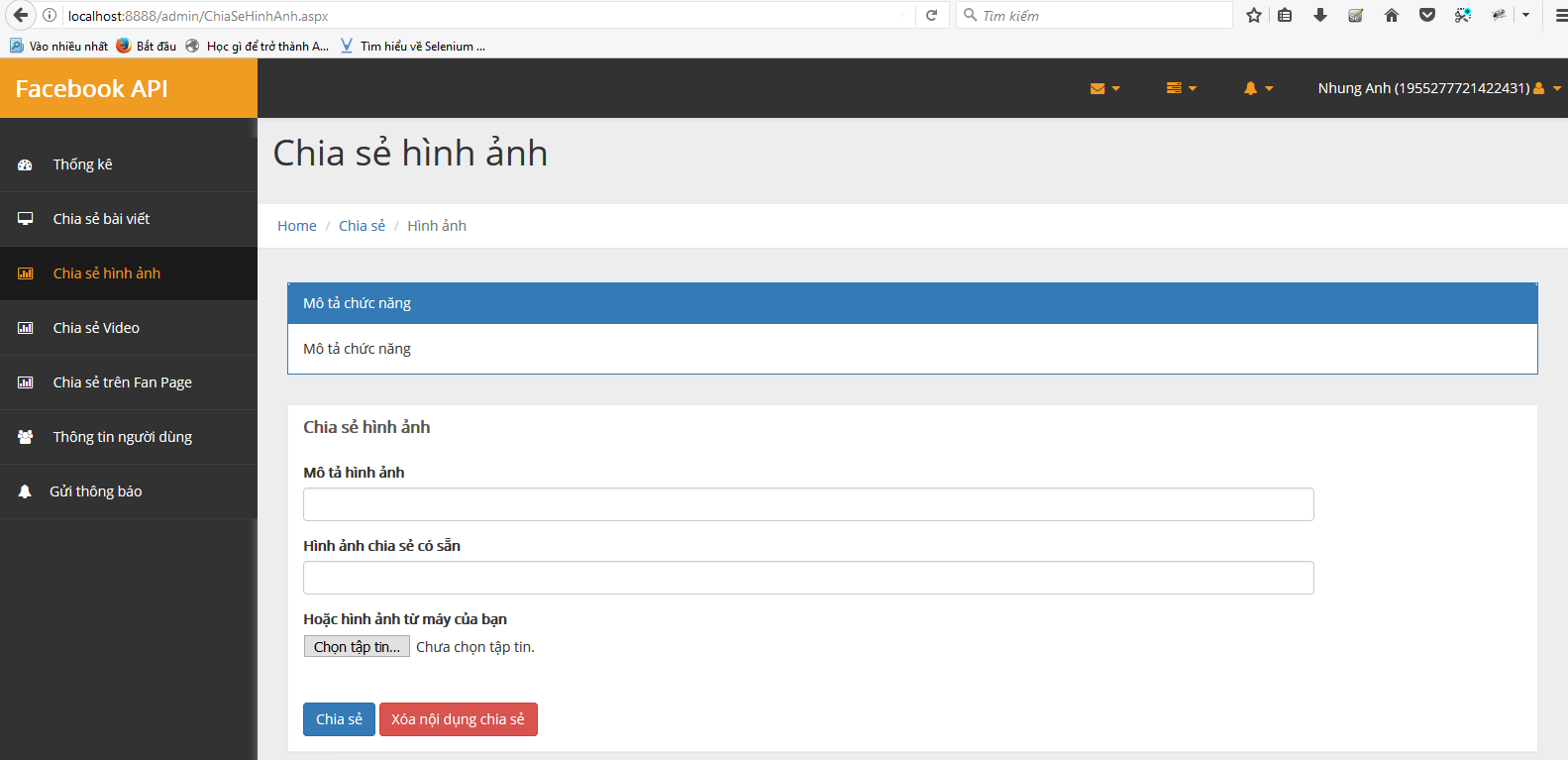
*Hình 4.2 : Màn hình Đăng nhập*



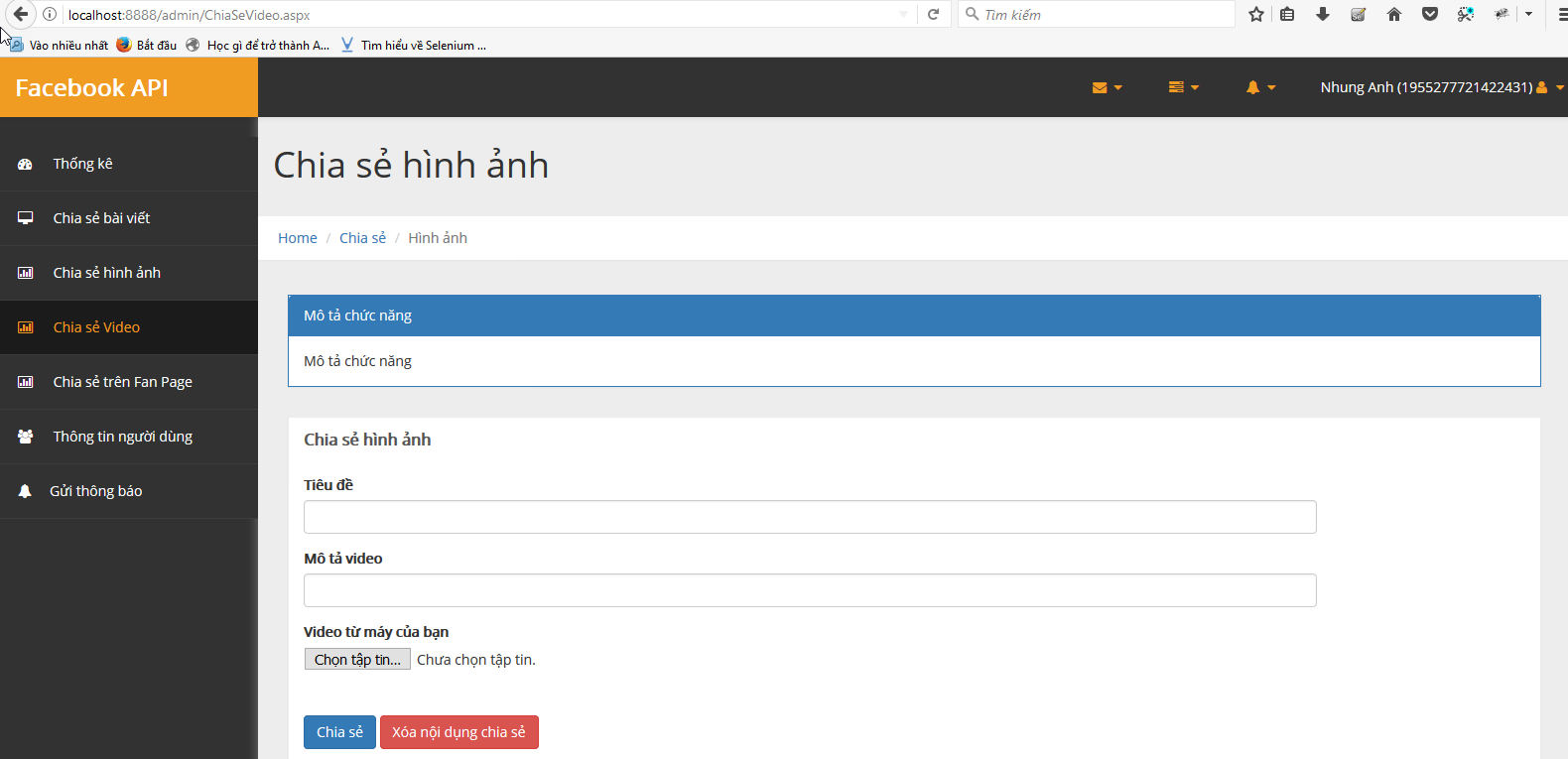
*Hình 4.3 : Màn hình Thống kê*



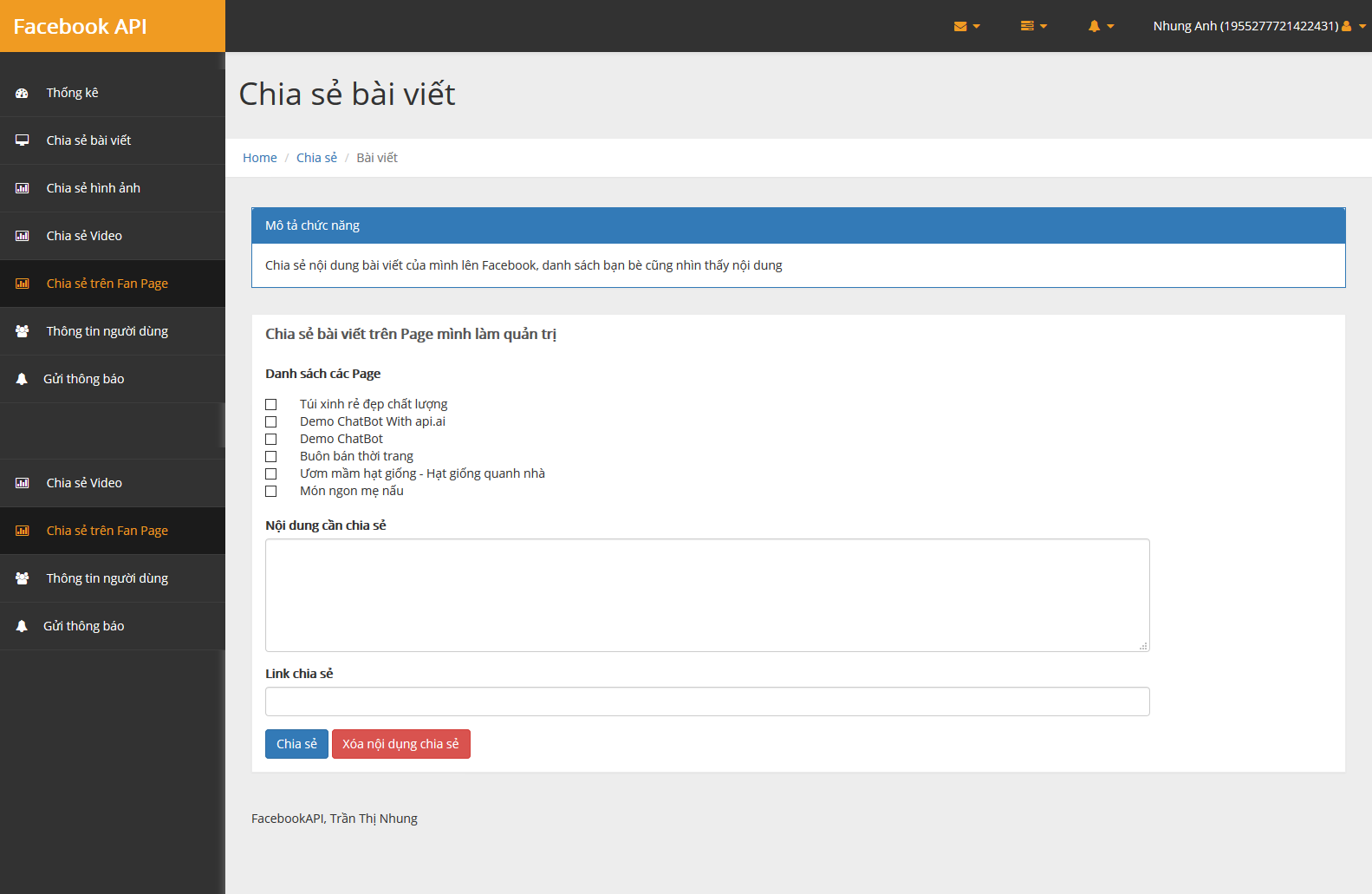
*Hình 4.4 : Màn hình Chia sẻ bài viết*



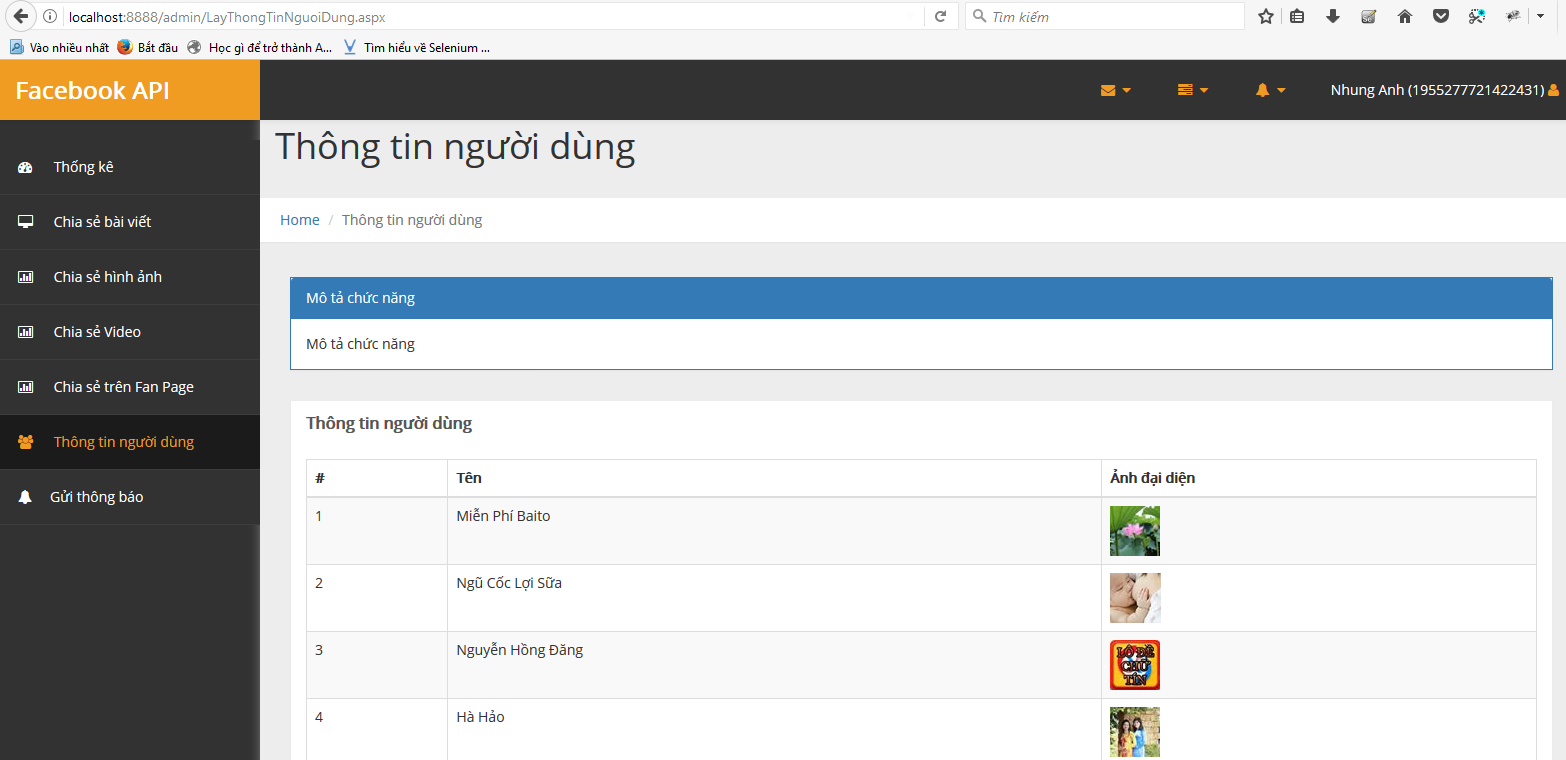
*Hình 4.5 : Màn hình Chia sẻ hình ảnh*



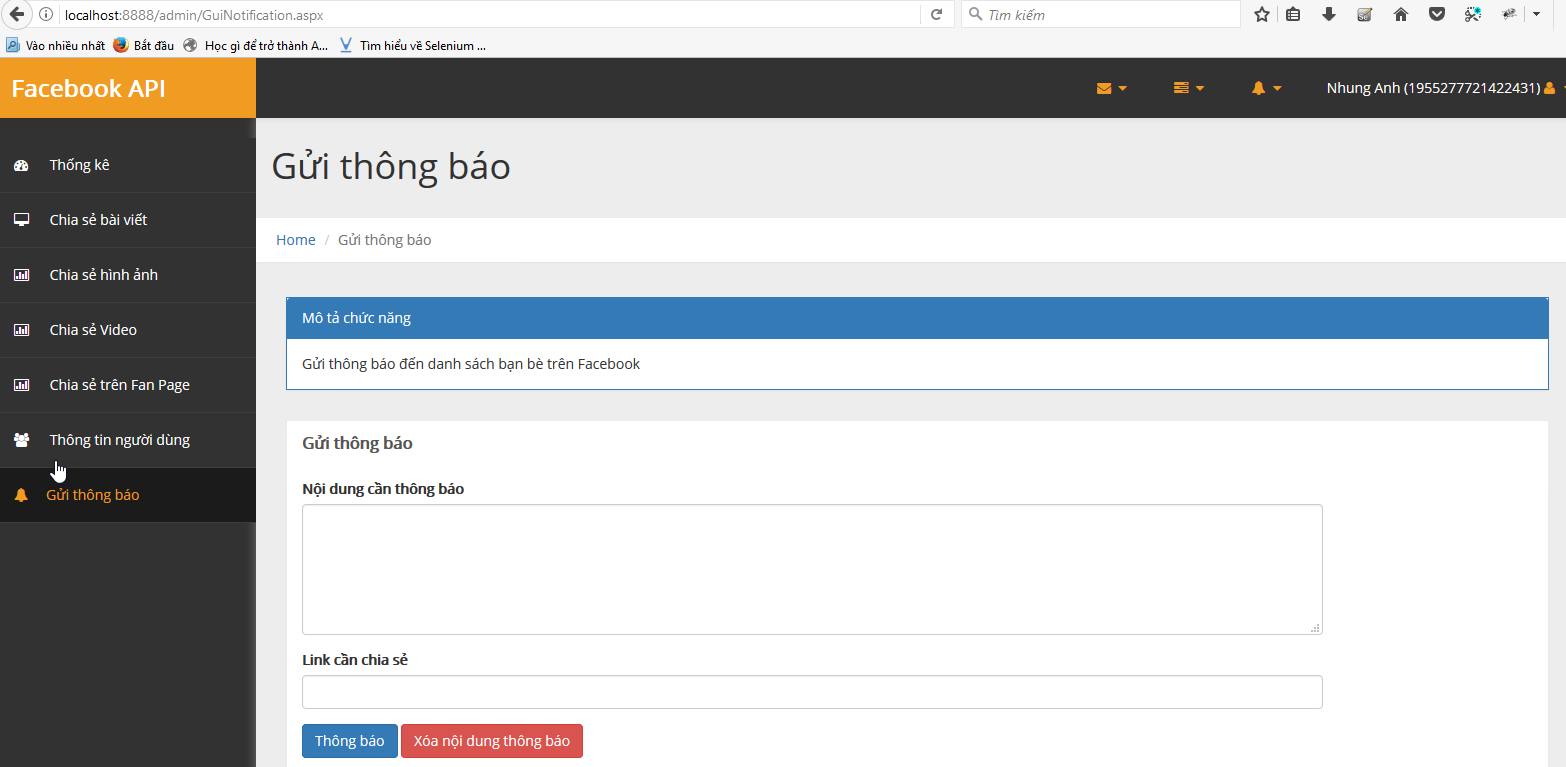
*Hình 4.6 : Màn hình Chia sẻ Video*



*Hình 4.7 : Màn hình Chia sẻ trên Fanpage*



*Hình 4.8 : Màn hình Thông tin người dùng*



*Hình 4.9 : Màn hình Gửi thông báo*

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. Kết quả đạt được của đề tài

Qua quá trình tìm hiểu và nghiên cứu đề tài này, bước đầu em đã tìm hiểu được cơ chế hoạt động của Facebook, thấy được sự khác biệt so với cơ chế hoạt động của các website thông thường. Em cũng đã trình bày về Facebook Platform – nền tảng của Facebook. Đây cũng chính là thành phần quan trọng nhất giúp người phát triển viết ứng dụng có thể tạo dựng một các dễ dàng. Đi sâu tìm hiểu về nền tảng của Facebook, em đã bước đầu hiểu được cách thức tạo dựng ứng dụng trên nền Facebook, cách sử dụng chúng thay thế các thẻ HTML thông thường và cách lấy dữ liệu bằng các hàm API và FQL.

Sau một thời gian nghiên cứu và thực hiện đề tài em đã hoàn thành cơ bản đề tài với kết quả sau:

* Tìm hiểu các cách thức facebook api hoạt động.
* Viết ứng dụng gọi facebook api để thực hiện một số thao tác hay dùng
* Tìm hiểu facebook app, cách thức hoạt động , tính ứng dụng của facebook app.

Tuy nhiên sau khi viết ứng dụng em có nhận thấy còn một số điểm hạn chế như:

* Khi sử dụng API để lấy dữ liệu từ Facebook, dữ liệu trả về còn chậm.
* Câu lệnh FQL không hỗ trợ đầy đủ, còn hạn chế ví dụ như không cho join nhiều bảng mà phải truy xuất từng bảng, muốn join nhiều bảng phải sử dụng câu lệnh FQL lồng nhau….
* FBJS,CSS không được Facebook nhận một cách nhanh chóng.
* Quyền truy cập của nhà phát triển hiện tại bị hạn chế nhiều permission để phát triển ứng dụng.

Qua bài này em hi vọng đã góp phần giải quyết vấn đề tạo dựng ứng dụng trên Facebook, giúp những người có ý định viết ứng dụng trên Facebook dễ dàng tiếp cận và hiểu sâu hơn về cách thức tạo ứng dụng cũng như trên nền tảng mà Facebook cung cấp, từ đó có thể dễ dàng tạo ra các ứng dụng linh hoạt và hiệu quả hơn.

## 2. Ưu điểm và nhược điểm

**a. Ưu điểm**

* Tính kết nối cao, chia sẻ cập nhật thông tin hàng ngày mà vẫn bảo vệ quyền riêng tư cá nhân.
* Ứng dụng được thiết kế khoa học, thao tác nhanh gọn, hỗ trợ Tiếng việt tốt.
* Dễ dàng lưu trữ và chia sẻ hình ảnh, Video clip…
* Khả năng kết nối, giao lưu với các thành viên trên các lãnh thổ khác nhau.
* Là mạng xã hội phổ biến hiện nay.

**b. Nhược điểm**

* Người dùng không tùy chỉnh được template để có trang riêng cho mình.
* Tài khoản facebook phải được đăng ký quyền của Developers (Nhà phát triển). Nếu là người mới dùng Facebook sẽ gặp khó khăn về thao tác.

## 3. Hướng phát triển của đề tài

* Ứng dụng sẽ có những chức năng giúp người dùng thao tác đơn giản hơn : chia sẻ tới nhiều trang cùng, đặt lịch thực hiện các chia sẻ…
* Thống kê giúp người dùng biết được các thông của page, app.
* Viết một số mẫu câu cho chat bot có thể tự trả lời.

# TÀI LIỆU THAM KHẢO

**Sách:**

* + 1. Hasin Hayder và Dr.Mark Alexander, *Learning Facebook Application Development*, Nxb Pack, 5-2008.
    2. Jay Goldman, *Facebook Cookbook*, Nxb O’Reilly Media, 10-2008.
    3. Wayne Graham, *Facebook API Developers Guide*, Nxb Apress, 2008.
    4. Jesse Stay, *FBML Essentials*, Nxb O’Reilly Media, 7 – 2008.
    5. Scott W.Ambler, *Introduction to Microsoft ASP.NET*, 4-2003.
    6. Jesse Stay ,*Microsoft .NET Framework SDK Documentation*, 7-2003.
    7. Arthur M.Langer, *Analysis and Design of Information System,* 2008.
    8. Sparks Geoffrey, *Database Modelling in UML,* 8-2011
    9. Yanic Inghelbrecht, *Introduction to UML Sequence Diagrams ,* 2008
    10. McLaughlin et al, *UML interaction Overview Diagrams,* 12-2008

**Web:**

* + 1. Facebook Developer, nguồn: <https://developers.facebook.com/>
    2. Wikipedia, nguồn: <https://en.wikipedia.org/wiki/ASP.NET>