

LỜI CẢM ƠN

Để hoàn thành khóa luận này, tôi xin tỏ lòng biết ơn sâu sắc đến TS. Nguyễn Hải Châu, đã tận tình hướng dẫn trong suốt quá trình viết khóa luận tốt nghiệp.

Tôi cũng xin chân thành cảm ơn quý Thầy, Cô trong khoa Công Nghệ Thông Tin, đặc biệt là các Thầy, Cô trong bộ môn Hệ Thống Thông Tin, Trường Đại Học Công Nghệ - Đại Học Quốc Gia Hà Nội đã tận tình truyền đạt kiến thức trong 4 năm học tập. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu khóa luận mà còn là hành trang quý báu để tôi bước vào đời một cách vững chắc và tự tin.

Tôi cũng thầm biết ơn sự ủng hộ của gia đình, bạn bè – những người thân yêu luôn là chỗ dựa vững chắc cho tôi.

Cuối cùng, tôi xin kính chúc quý Thầy, Cô và gia đình dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

Hà Nội, ngày 20 tháng 5 năm 2010

Sinh viên thực hiện

Hoàng Thị Hiền

TÓM TẮT KHÓA LUẬN

Trong khóa luận này tôi đi sâu vào tìm hiểu cách thức xây dựng ứng dụng của mạng xã hội Facebook. Phần đầu tôi xin giới thiệu tổng quát về mạng xã hội nói chung và mạng Facebook nói riêng, một số thống kê mà tôi tập hợp được về tốc độ phát triển của các mạng xã hội hiện nay. Ngoài ra tôi cũng xin giới thiệu một cách khái quát cơ chế hoạt động của mạng Facebook. Tiếp đó, tôi sẽ trình bày một cách chi tiết về nền tảng của Facebook – công cụ được các nhà phát triển Facebook tạo ra để giúp phát triển ứng dụng trên nền Facebook. Nền tảng này bao gồm các đối tượng: API (*Application Programming Interface*), FBML (*Facebook Markup Language*), FQL (*Facebook Query Language*) và FBJS (*Facebook Javascript*). Cuối cùng tôi sẽ mô tả quá trình xây dựng ứng dụng trên Facebook có sử dụng nền tảng mà Facebook cung cấp như API, FBML, FBJS. Ứng dụng này sẽ được thêm vào thư mục ứng dụng của mạng Facebook và được sử dụng như một ứng dụng của trang Facebook.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ FACEBOOK.....	8
1.1. Giới thiệu khái quát về mạng xã hội.....	8
1.2. Giới thiệu về Facebook.....	8
1.2.1. Facebook là gì ?	9
1.2.2. Lịch sử phát triển.....	9
1.2.3. Cơ chế hoạt động của Facebook.	10
1.2.3.1. Mô hình ứng dụng web thông thường.....	10
1.2.3.2. Mô hình ứng dụng web Facebook.....	10
1.2.3.3. Cách thức làm việc	12
1.2.4. Các dạng ứng dụng trên Facebook hiện nay	12
1.2.5. Điểm mạnh và yếu của mạng Facebook.	13
1.2.6. Thống kê số lượng người dùng Facebook so với một số mạng xã hội khác..	14
1.3. Hướng phát triển của Facebook trong các năm tới.....	15
CHƯƠNG 2 : FACEBOOK PLATFORM.....	16
2.1. Facebook Platform là gì ?	16
2.2. Các đối tượng nền tảng của Facebook.....	16
2.2.1. Facebook API.....	16
2.2.1.1. Facebook API là gì ?	16
2.2.1.2. Facebook API làm việc như thế nào?.....	17
2.2.1.3. Các phương thức của API.	17
2.2.1.4. Data Store API.....	27
2.2.1.5. Thư viện khách.....	27
2.2.2. Ngôn ngữ đánh dấu Facebook – FBML.....	28

2.2.2.1.	FBML là gì ?	28
2.2.2.2.	Nhóm các thẻ FBML	28
2.2.2.3.	CSS	43
2.2.3.	Ngôn ngữ truy vấn – FQL	44
2.2.3.1.	FQL là gì ?	44
2.2.3.2.	Cấu trúc câu truy vấn FQL	44
2.2.3.3.	Các toán tử và các hàm hỗ trợ cơ bản.	45
2.2.3.4.	Các bảng và các trường được đánh chỉ mục.	46
2.2.4.	Ngôn ngữ tạo script – FBJS	48
2.2.4.1.	FBJS là gì ?	48
2.2.4.2.	FBJS làm việc như thế nào ?	48
2.2.4.3.	Ajax	49
CHƯƠNG 3 : XÂY DỰNG ỨNG DỤNG TRÊN NỀN FACEBOOK		50
1.1.	Mô tả ứng dụng	50
1.2.	Cài đặt	54

BẢNG CÁC TỪ VIẾT TẮT

AIP	<i>Application Programming Interface</i>
FBML	<i>Facebook Markup Language</i>
FQL	<i>Facebook Query Language</i>
FBJS	<i>Facebook Javascript</i>
CSS	<i>Cascading Style Sheets</i>

DANH SÁCH CÁC HÌNH

Hình 1.1	<i>Mô hình web thông thường</i>
Hình 1.2	<i>Mô hình web Facebook</i>
Hình 1.3	<i>Mô hình web Facebook sử dụng cache</i>
Hình 1.4	<i>Cách thức làm việc của Facebook</i>
Hình 1.5	<i>Bản đồ mạng xã hội</i>
Hình 2.1	<i>Cách thức làm việc của Facebook API</i>
Hình 2.2	<i>Dòng chảy thông tin từ máy chủ ứng dụng tới trình duyệt của người dùng</i>

GIỚI THIỆU

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 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 có 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 trong mạng Facebook và việc tạo dựng các ứng dụng mới trên Facebook là rất cần thiết.

Trong khóa luận này, tôi sẽ đi sâu vào tìm hiểu cách thức tạo dựng ứng dụng trên mạng xã hội, cụ thể là mạng Facebook. Tôi sẽ trình bày một cách chi tiết các đối tượng được các nhà phát triển Facebook tạo ra để hỗ trợ người viết ứng dụng. Bên cạnh đó, khóa luận cũng trình bày một cách khái quát về mạng xã hội hiện nay, một số thống kê mà tôi đã thu thập được và cách thức hoạt động của Facebook, nhằm giúp người xem có thể hiểu một phần cơ chế hoạt động của mạng xã hội này.

Nội dung chính của khóa luận bao gồm 3 chương, với nội dung cụ thể như sau:

Chương 1: Tổng quan về Facebook. Chương này tập trung giới thiệu một cách 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 2: Facebook Platform. Nội dung chính của chương là giới thiệu về nền tảng của Facebook, cụ thể là các đối tượng Application Programming Interface (API), Facebook Markup Language (FBML), Facebook Query Language (FQL). Đây là các đối tượng mà các nhà phát triển cung cấp để giúp phát triển ứng dụng một cách dễ dàng hơn.

Ngoài ra chương này còn giới thiệu một số các đối tượng khác cũng được Facebook cung cấp như Facebook Javascript (FBJS).

Chương 3: Xây dựng ứng dụng trên nền Facebook. Chương này sẽ mô tả ứng dụng thực tế xây dựng được bằng việc sử dụng các đối tượng mà Facebook cung cấp.

CHƯƠNG 1: TỔNG QUAN VỀ FACEBOOK

1.1. Giới thiệu khái quát về mạng xã hội.

1.1.1. Khái niệm

Mạng xã hội hay còn gọi là mạng xã hội ảo (social network) là dịch vụ kết nối các thành viên cùng sở thích trên Internet lại với nhau với nhiều mục đích khác nhau không phân biệt không gian và thời gian.

1.1.2. Lịch sử

Mở đầu cho kỷ nguyên mạng xã hội là vào năm 1995 với sự ra đời của trang Classmate.com. Tại thời điểm năm 1995 mạng xã hội này đã có tới 50 triệu người dùng. Đây là một con số kỷ lục khi mà trong thời kỳ đó Internet vẫn còn rất sơ khai. Hai năm sau là sự ra đời của 2 trang Care2.com và Opendiary.com nhằm đối trọng với Classmate. Nhưng tới năm 1999 mới đánh dấu sự bùng nổ về mạng xã hội với 5 trang mạng xã hội được ra đời: Xanga.com, Kiwibox.com, LiveJournal.com, BlackPlanet.com và Vampirefreaks.com. Năm 2003, MySpace ra đời với các tính năng như phim ảnh (*embedded video*) và nhanh chóng thu hút hàng chục ngàn thành viên mới mỗi ngày. Trong vòng một năm, MySpace trở thành mạng xã hội đầu tiên có nhiều lượt xem hơn cả Google và được tập đoàn News Corporation mua lại với giá 580 triệu USD^[1]. Vào tháng 2 năm 2004, sự ra đời của Facebook đánh dấu bước ngoặt mới cho hệ thống mạng xã hội trực tuyến với nền tảng lập trình "Facebook Platform" cho phép thành viên tạo ra những công cụ (*apps*) mới cho cá nhân mình cũng như các thành viên khác dùng. Facebook Platform nhanh chóng gặt hái được thành công vượt bậc, mang lại hàng trăm tính năng mới cho Facebook và đóng góp không nhỏ cho con số trung bình 19 phút mà các thành viên bỏ ra trên trang này mỗi ngày. Tiếp theo năm 2006 đánh dấu sự ra đời của Twitter.

Tới thời điểm hiện tại các trang web xã hội vẫn tiếp tục tăng. Đã có hàng trăm các mạng xã hội khác nhau như Facebook, Cramster, Internship Ratings, Twitter, Remember the Milk, Gradefund ... Tuy nhiên người dùng hầu như chỉ sử dụng các trang nổi tiếng như Facebook, Twitter, Myspace.

1.2. Giới thiệu về Facebook.

1.2.1. Facebook là gì ?

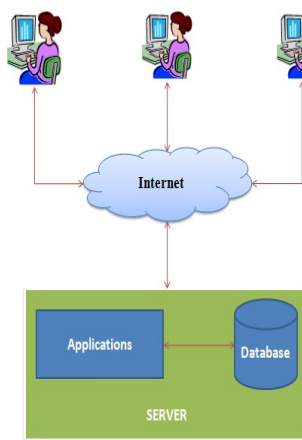
Facebook là một website mạng xã hội truy cập miễn phí do công ty Facebook, Inc điều hành và sở hữu tư nhân. Người dùng có thể tham gia các mạng lưới được tổ chức theo thành phố, nơi làm việc, trường học và khu vực để liên kết và giao tiếp với người khác. Mọi người cũng có thể kết bạn và gửi tin nhắn cho họ, và cập nhật trang hồ sơ cá nhân của mình để thông báo cho bạn bè biết về chúng. Tên của website nhắc tới những cuốn sổ lưu niệm dùng để ghi tên những thành viên của cộng đồng campus mà một số trường đại học và cao đẳng tại Mỹ đưa cho các sinh viên mới vào trường, phòng ban, và nhân viên để có thể làm quen với nhau tại khuôn viên trường.

1.2.2. Lịch sử phát triển.

Facebook mở đầu là một phiên bản Hot or Not của đại học Harvard với tên gọi Facemash. Mark Zuckerberg, khi đang học năm thứ hai tại Harvard, đã dựng nên Facemash vào ngày 28 tháng 10 năm 2003. Ngày 4 tháng 2 năm 2004, Zuckerman thành lập “The Facebook”, ban đầu đặt tại thefacebook.com. Việc đăng ký thành viên ban đầu giới hạn trong những sinh viên của đại học Harvard, và trong vòng một tháng đầu tiên, hơn một nửa số sinh viên tại Harvard đã đăng ký dịch vụ này. Eduardo Saverin (lĩnh vực kinh doanh), Dustin Moskovitz (lập trình viên), Andrew McCollum (nghệ sĩ đồ họa), và Chris Hughes nhanh chóng tham gia cùng với Zuckerberg để giúp quảng bá website. Vào tháng 3 năm 2004, Facebook mở rộng sang Stanford, Columbia, và Yale. Việc mở rộng tiếp tục khi nó mở cửa cho tất cả các trường thuộc Ivy League và khu vực Boston, rồi nhanh chóng đến hầu hết đại học ở Canada và Hoa Kỳ. Tháng 6 năm 2004, Facebook chuyển cơ sở điều hành đến Palo Alto, California. Công ty đã bỏ chữ *The* ra khỏi tên sau khi mua được tên miền facebook.com vào năm 2005. Tới tháng 9 năm 2005, Facebook ra mắt phiên bản trung, Zuckerberg gọi nó là một bước logic tiếp theo. Vào thời gian đó, các mạng của trường trung học bắt buộc phải được mời mới được gia nhập. Facebook sau đó mở rộng quyền đăng ký thành viên cho nhân viên của một vài công ty, trong đó có Apple Inc và Microsoft. Tiếp đó vào ngày 26 tháng 9 năm 2006, Facebook mở cửa cho mọi người trên 13 tuổi với một địa chỉ email hợp lệ. Vào tháng 10 năm 2008, Facebook tuyên bố nó đã thiết lập một trụ sở quốc tế tại Dublin, Ireland^[2].

1.2.3. Cơ chế hoạt động của Facebook.

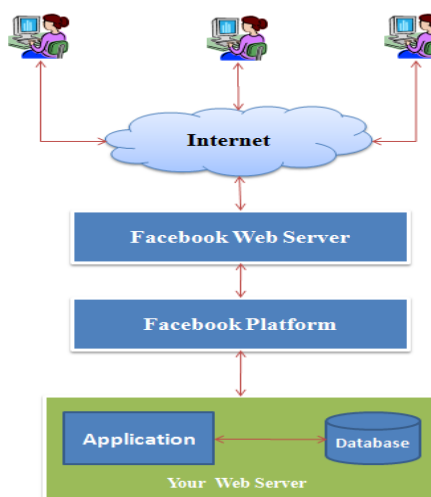
1.2.3.1. Mô hình ứng dụng web thông thường



Hình 1.1: Mô hình web thông thường

Ứng dụng và cơ sở dữ liệu được đặt trên một máy chủ Web. Ứng dụng sẽ chạy trên server và người dùng thực thi ứng dụng bằng cách sử dụng giao thức HTTP thông qua trình duyệt. Mô hình này sẽ làm việc hiệu quả nếu như server và client có 1 kết nối Internet ổn định.

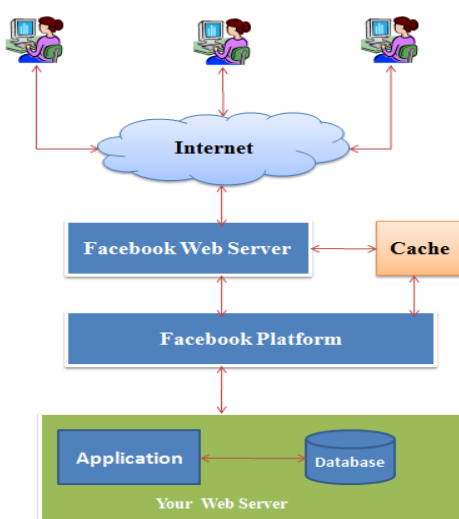
1.2.3.2. Mô hình ứng dụng web Facebook



Hình 1.2 : Mô hình web Facebook

Người dùng truy cập Facebook.com và ứng dụng thông qua trình duyệt và Internet. Tuy nhiên ứng dụng không được đặt tại máy chủ của Facebook mà được lưu trên máy chủ của chính người tạo ra ứng dụng đó. Facebook Platform cũng cung cấp một giao diện cho người viết ứng dụng.

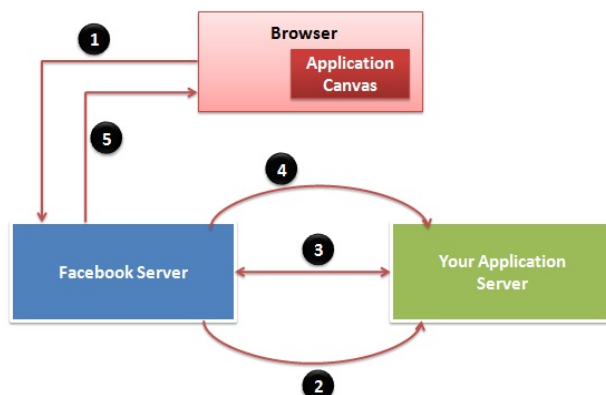
Với một máy chủ web bình thường trung bình chỉ chịu được từ 100 tới 500 truy vấn cùng một lúc. Tuy nhiên với Facebook trung bình mỗi giây phải hiện 600 nghìn hình ảnh cùng một lúc. Vấn đề đặt ra ở đây là làm thế nào để mỗi giây Facebook hiện được 600 ngàn hình ảnh ? Để giải quyết vấn đề này Facebook đã sử dụng vùng lưu trữ đệm (cache) và đây cũng chính là trái tim của hệ thống.



Hình 1.3: Mô hình web Facebook sử dụng cache

Facebook đã phát triển Haystack – một hệ thống quản lý các tập tin hình ảnh trung gian để cải tiến sự truy vấn. Trước đây, Facebook đã sử dụng hệ thống 2 tầng : một tầng chuyên cho việc upload hình ảnh và lưu trữ vào server. Tầng kia có nhiệm vụ tìm và lấy hình ảnh từ server để trả lời các truy vấn. Tuy vậy việc xử lý theo cách cổ điển này là không phù hợp với nhu cầu tăng rất nhanh số lượng người dùng của Facebook dẫn tới tình trạng nghẽn mạch Input / Output. Haystack sẽ quản lý và lưu trữ các hình ảnh trong vùng đệm và từ đây sẽ trả lời các truy vấn hình ảnh. Bên cạnh đó Facebook còn cải tiến các đoạn code trên webserver để giảm thiểu kích thước tập tin và thời gian tính toán. Ngôn ngữ lập trình cho từng bộ phận cũng dần được tối ưu hóa. Người truy cập sẽ được phân luồng ngẫu nhiên vào các ngân hàng dữ liệu để giảm tải và các truy vấn của họ sẽ được hệ thống cache trả lời nhanh chóng.

1.2.3.3. Cách thức làm việc



Hình 1.4: Cách thức làm việc của Facebook

- 1) Trình duyệt của người dùng yêu cầu <http://apps.facebook.com/myapp>. Địa chỉ này trở 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
- 2) 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.
- 3) 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.
- 4) 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.
- 5) 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.

1.2.4. Các dạng ứng dụng trên Facebook hiện nay

Sự phát triển ngày càng lớn mạnh của Facebook đã kéo theo sự ra đời của rất nhiều các ứng dụng. Theo thống kê hiện nay có khoảng 90.000 ứng dụng trên Facebook, với sự tham gia của 120 công ty.

Theo thống kê năm 2008 ứng dụng trên Facebook được chia làm các loại:

- | | | |
|--------------|------------------|------------|
| • Alerts | • File sharing | • Photo |
| • Business | • Food and Drink | • Politics |
| • Chat | • Gaming | • Sports |
| • Classified | • Just for fun | • Travel |
| • Dating | • Messaging | • Utility |
| • Education | • Mobile | • Video |
| • Events | • Money | |
| • Fashion | • Music | |

Hiện nay ứng dụng hấp dẫn người dùng trên Facebook vẫn là các game. Dẫn đầu là game Farmville với số lượng người dùng hàng tháng lên tới 82.580.911 người. Thứ 2 là ứng dụng Static FBML do Facebook phát triển với 46.827.021 người dùng. Đứng thứ 3 là Birthday Card với 41.904.049 người dùng hàng tháng. Tiếp theo là các ứng dụng Café world, Facebook for Iphone, Texas HoldEm Poker, ...

1.2.5. Điểm mạnh và yếu của mạng Facebook.

Ưu điểm

- Là mạng phổ biến hiện nay.
- Tính kết nối cao, chia sẻ cập nhật thông tin hàng ngày mà vẫn bảo vệ được quyền riêng tư.
- 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 ...

- Nhiều ứng dụng, game, quiz. Người dùng cũng có thể tạo ra ứng dụng cho riêng mình.
- Khả năng kết nối, giao lưu với các thành viên trên các lãnh thổ khác nhau.
- Tính năng viết blog cho phép soạn thảo dễ dàng.

Nhược điểm

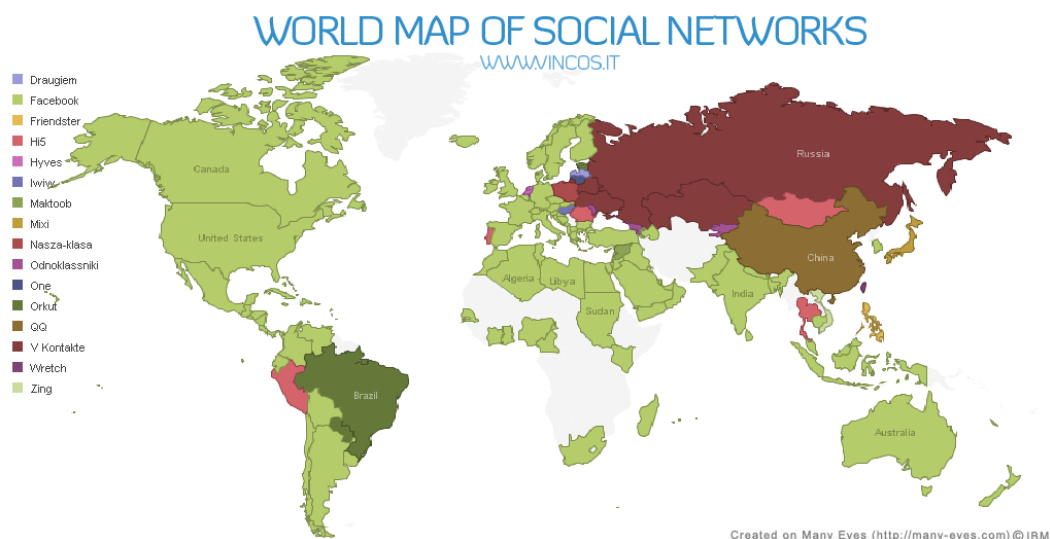
- Người dùng không tùy chỉnh được template để có trang riêng cho mình.
- Nếu là người mới dùng Facebook sẽ gặp khó khăn về thao tác.

1.2.6. Thống kê số lượng người dùng Facebook so với một số mạng xã hội khác

Theo số liệu thống kê tính tới tháng 6 /2007, Facebook có số lượng thành viên tăng từ 52 triệu lên 132 triệu trên toàn thế giới vào tháng 6 / 2008, tương đương với tỉ lệ tăng là 153%. Trang web mạng xã hội đứng thứ 2 là MySpace với 117 triệu thành viên nhưng trang này chỉ có tỉ lệ tăng trưởng là 3% trong cùng thời kỳ.

Khi xét riêng về tỉ lệ tăng trưởng, trang web mạng xã hội duy nhất tiên gần Facebook là Hi5 với tỉ lệ tăng là 100% từ năm 2007 tới năm 2008. Tuy nhiên Hi5 chỉ có 56 triệu thành viên.

Nếu xét riêng tỉ lệ tăng trưởng của Facebook trên thế giới còn đáng ngạc nhiên hơn. Các tỉ lệ này là 303% ở Châu Âu, 458% ở Châu Á Thái Bình Dương, 403% ở Trung Phi và ở Châu Mỹ Latin là 1055%. ^[3]



Hình 1.5: Bản đồ mạng xã hội tháng 12 năm 2009^[4]

Tại Việt Nam hiện nay số lượng người dùng Facebook đang tăng vượt bậc, Việt Nam là nước có số lượng người dùng Facebook tăng nhanh nhất thế giới theo thống kê tháng 7/2009 với tỉ lệ tăng là 26,49%. Tính tới ngày 20/7/2009 số lượng người dùng đã vượt qua mức 400.000, chiếm 0,18% lượng người dùng Facebook trên thế giới, đã tăng gấp 3 lần so với thời điểm 5/6/2009 (140.000). Tuy nhiên xét tới tháng 6/2009, theo website Vincos.com, Zing lại là mạng xã hội lớn nhất Việt Nam.

1.3. Hướng phát triển của Facebook trong các năm tới.

Tính tới thời điểm hiện tại, Facebook là mạng xã hội lớn nhất trên thế giới, thu hút hàng trăm triệu người. Số lượng thành viên gia tăng một cách nhanh chóng là một lợi thế giúp Facebook phát triển các tính năng mới. Theo như ông Zuckerberg, hiện đang là tổng giám đốc điều hành của Facebook, cho biết: 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 2 : FACEBOOK PLATFORM

=====*****=====

2.1. 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 với các tính năng cốt lõi của Facebook. Được giới thiệu vào ngày 24 tháng 5 năm 2007 và đến ngày 09 tháng 12 năm 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.

Khi tạo ra một ứng dụng từ Facebook Platform, có thể tích hợp một vài thành phần trong framework:

- Hiện thư mục ứng dụng.
- Cung cấp trang giới thiệu.
- Cung cấp cho người sử dụng những lựa chọn.
- Giới thiệu cho những người khác.
- Hiện thị trang Facebook Canvas.
- Tạo Left Nav.
- Hiện thị box thông tin.
- Tạo link liên kết tới profile.
- Thực thi và gửi thông báo tới nguồn tin .
- Gửi thông báo email.
- Tạo yêu cầu.
- Tạo tin nhắn đính kèm.
- Tích hợp với các thiết lập bảo mật của Facebook.

2.2. Các đối tượng nền tảng của Facebook.

2.2.1. Facebook API.

2.2.1.1. Facebook API là gì ?

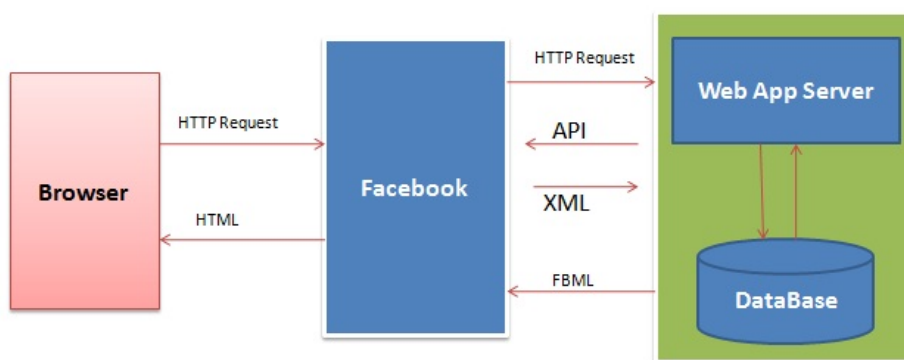
API là viết tắt của Application Programming Interface (giao diện chương trình ứng dụng). Facebook API là một nền tảng để xây dựng những ứng dụng cho các thành viên của mạng xã hội Facebook. API cho phép các ứng dụng sử dụng các kết

nổi 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 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.

2.2.1.2. Facebook API làm việc như thế nào?

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 về user, groups, photo,... mà họ cần.



Hình 2.1: Cách thức làm việc của Facebook API

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. Session_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.

2.2.1.3. Các phương thức của API.

Các phương thức API của Facebook được chia thành các nhóm một cách hợp lý:

- | | |
|------------------|---------------|
| ▪ Authentication | ▪ Fql |
| ▪ Events | ▪ Photos |
| ▪ FBML | ▪ Friends |
| ▪ Feed | ▪ Groups |
| ▪ Notifications | ▪ Error codes |
| ▪ Profile | ▪ Users |

▪ **Authentication**

Nhóm hàm này cung cấp 2 phương thức chính để đối phó với xác nhận người dùng. Phương thức facebook.auth.createToken() sẽ tạo ra một mã xác thực để thông qua cơ chế xác thực của Facebook. Sau khi đăng nhập, phương thức thứ 2 là facebook.auth.getSession() sẽ chứa mã thông báo, chỉ khi yêu cầu một auth_token cụ thể. Authentication cũng chính là trở ngại lớn cho các nhà phát triển ứng dụng trực tuyến.

❖ Auth.createToken()

- Mô tả: tạo ra một auth_token để vượt qua trang login.php của Facebook để sau đó tạo ra một phiên giao dịch khi người dùng đã đăng nhập.
- Tham số: api_key, sig, v
- Ví dụ:

```
$token = $facebook->api_client->
auth_createToken($api_key,md5($sig),"1.0","JSON");
```

❖ Auth.getSession()

- Mô tả: Trả về một khóa giao dịch bị ràng buộc với auth_token.
- Tham số: auth_token

- Ví dụ:

```
$session = $facebook->api_client-> auth_getSession ( $auth_token );
```

- **Events**

Nhóm này có 2 phương thức chính để lấy thông tin về sự kiện.

- ❖ Events.get()

- Mô tả: Sử dụng phương thức này để tìm tất cả các sự kiện của một người dùng, hoặc để truy vấn một tập các sự kiện của một danh sách các ID sự kiện (eids). Nếu 2 tham số uid và eids được cung cấp, phương thức sẽ trả về các sự kiện trong danh sách các eids có liên quan tới người sử dụng. Nếu không có tham số eids, phương thức trả lại tất cả sự kiện liên quan tới người dùng được chỉ định. Nếu uid được bỏ qua, phương thức trả về tất cả các sự kiện liên quan tới eids được cung cấp.

- Tham số: uid, eids, start_time, end_time, rsvp_status

- Ví dụ:

Lọc sự kiện của một người dùng:

```
$events = $facebook->apiclient->events_get(12345,  
null,null,null,null);
```

Lọc sự kiện của tập id sự kiện được cung cấp

```
$events = $facebook->api_client->events_get (null,  
array(14381739642,16044821668), null,null,null);
```

- ❖ Events.getMember()

- Mô tả: Trả về danh sách thành viên liên quan tới một sự kiện. Đây là một mảng đa chiều với chiều thứ nhất chứa tình trạng RSVP(attending, unsure, declined, not_replied) và chiều thứ 2 là uid của người dùng.
- Tham số: eid
- Ví dụ:

```
$members = $facebook->api_client->events_GetMembers ($eid);
```

▪ **Fbml**

Để phù hợp với một vài chi tiết mới hơn của FBML, Facebook đã cung cấp 2 phương thức API :

❖ Fbml.refreshImgSrc()

- Mô tả: Lấy và lưu trữ lại hình ảnh đã lưu cho các Url để sử dụng cho việc lấy hình ảnh ở các trang không phải trang vài nền thông qua các hàm API.
- Tham số: url
- Ví dụ:

```
$facebook->api_client->fbml.refreshImgSrc($url_image);
```

❖ Fbml.refreshRefUrl()

- Mô tả: Lấy và lưu trữ lại nội dung đã lưu cho các Url để sử dụng trong thẻ FBML <fb:ref>
- Tham số: url
- Ví dụ:

```
$facebook->api_client->fbml.refreshRefUrl  
("http://www.mysite.com/someurl.php");
```

▪ **Feed**

Để cập nhật lại bảng tin (feed) của người dùng, API có phương thức:

❖ Stream.publish()

- Mô tả: Phương thức này xuất bản một bài (post) vào luồng(stream)-tường của người dùng hiện tại hoặc người dùng cụ thể, hoặc tường của một người bạn hoặc trang Facebook, nhóm hoặc sự kiện được kết nối tới phiên giao dịch hiện tại hoặc người dùng cụ thể. Mặc định, lời gọi này được công bố tới tường của người dùng hiện tại. Những bài đăng này cũng xuất hiện trong luồng của những người có kết nối với cả điểm neo và điểm đích của bài đăng.
- Ví dụ:

```
$facebook->api_client->stream_publish($mess);
```

▪ Fql

Như đã biết, Facebook cung cấp các nhóm hàm API để lấy dữ liệu, bên cạnh đó, API còn cung cấp một nhóm hàm để giúp người viết ứng dụng sử dụng các câu truy vấn FQL để lấy dữ liệu từ cơ sở dữ liệu của Facebook.

❖ Fql.query()

- Mô tả: Thực hiện một truy vấn FQL. Với các truy vấn phức tạp, kết quả của một truy vấn ở bên trong truy vấn khác, để tránh lặp lại việc gọi hàm API nhiều lần, có thể sử dụng phương thức fql.multiquery()
- Tham số: query
- Ví dụ:

```
$query = "SELECT name FROM user
        WHERE uid IN (SELECT uid2 FROM friend WHERE uid1=211031);
$result=$facebook->api_client->fql_query($query);
```

❖ Fql.multiquery()

- Mô tả: Thực hiện một loạt truy vấn FQL trong một lời gọi và trả về dữ liệu cùng một lúc.
- Tham số: queries
- Ví dụ:

```
$query1 = "SELECT uid, name from user where uid in ($friends)";
$query2 = "SELECT pid, subject from photo_tag where pid in ($pids)";
$queries = '{
    "query1": "' . $query1 . '",
    "query2": "' . $query2 . '"
}';
$result = $facebook->api_client->fql_multiquery($queries);
$userInfo = $result[0]['fql_result_set'];
$photoTags = $result[1]['fql_result_set'];
```

▪ Friends

Khi phát triển ứng dụng, việc hiển thị danh sách bạn bè của người dùng là rất phổ biến. Để giải quyết vấn đề này API có các hàm giúp người viết ứng dụng lấy thông tin về bạn bè của người dùng

❖ Friends.areFriends()

- Mô tả: Trả lại mối liên quan giữa 2 người dùng (là bạn hoặc không là bạn của nhau).
- Tham số: uids1, uids2
- Ví dụ:

```
$areFriends = $facebook->api_client->friends_areFriends($user1,$user2);
```

❖ Friends.get()

- Mô tả: Trả lại id user của người dùng, những người là bạn của người dùng hiện hành.
- Ví dụ:

```
$friends = $facebook->api_client->friends_get();
```

❖ Friends.getAppUsers()

- Mô tả: Trả về mảng id của bạn bè, những người đã cài đặt ứng dụng
- Ví dụ:

```
$friends = $facebook->api_client->friends_getAppUsers();
```

❖ Friends.getList()

- Mô tả: Trả về mảng flids (friend list ids) của người dùng hiện tại
- Ví dụ:

```
$friendLists = $facebook->api_client->friends_getLists();
```

▪ **Groups**

Làm việc với nhóm cũng là một phần rất quan trọng của ứng dụng . API đã cung cấp các phương thức:

❖ Groups.get()

- Mô tả: Trả về các nhóm theo tiêu chí lọc như nhóm có liên kết với một người dùng , theo một danh sách gids hoặc theo cả 2 tiêu chí lọc trên.
- Ví dụ:

```
$groups = $facebook->api_client->groups_get($user);
```

❖ Groups.getMembers()

- Mô tả: Trả về danh sách thành viên của một nhóm. Kết quả trả về là một mảng đa chiều với chiều đầu tiên là kiểu thành viên (admin, member, officer, not_replied)
- Tham số: gid
- Ví dụ:

```
$members = $facebook->api_client->groups_getMembers($gid);
```

▪ **Notifications**

Facebook cho phép gửi và nhận các thông báo trong ứng dụng với các hàm API. Có 2 hàm chính:

❖ Notifications.get()

- Mô tả: Trả về một mảng đa chiều với các đối tượng là các loại thông báo: tin nhắn, giới thiệu nhóm, giới thiệu sự kiện, yêu cầu kết bạn, chia sẻ, poke)
- Ví dụ:

```
$notifications = $facebook->api_client->notifications_get();
```

❖ Notifications.getList()

- Mô tả: Trả về tất cả thông tin của người dùng hiện tại.
- Ví dụ:

```
$notifications = $facebook->api_client->notifications_getList();
```

▪ **Photos**

Với hơn 60 triệu bức ảnh được thêm vào mỗi tuần bởi người dùng Facebook, các nhà phát triển Facebook đã cung cấp một nhóm phương thức API để giúp các nhà phát triển ứng dụng dễ dàng thao tác với các ảnh của người dùng.

❖ Photos.addTag()

- Mô tả: Thêm một thẻ đi kèm thông tin cho một bức ảnh. Thẻ chỉ có thể thêm vào những bức ảnh đang chờ giải quyết thuộc sở hữu của người dùng cụ thể hoặc người dùng hiện hành. Ứng dụng với quyền mở rộng photo_upload có thể thêm thẻ vào bất kì bức ảnh nào thuộc quyền sở hữu của người dùng.
- Tham số: pid, tag_uid, tag_text, x, y
- Ví dụ:

```
$result = $facebook->api_client->photos_addTag($pid, $uid  
                                                , null, 50, 50, null, null);
```

❖ Photos.createAlbum()

- Mô tả: Tạo và trả về một album mới thuộc quyền sở hữu của một người dùng cụ thể hoặc người dùng hiện hành.
- Tham số: name
- Ví dụ:

```
$album = $facebook->api_client->photos_createAlbum('Test Album',  
                                                    'Testville', 'This is a test');
```

❖ Photos.get()

- Mô tả: Trả về tất cả các ảnh theo tiêu chí lọc. Có thể lọc theo 4 cách: được đánh dấu theo chủ đề đã quy định, có chứa trong album được định rõ bởi aid, bao gồm tập các ảnh được định rõ bởi pid hoặc cả 3 tiêu chí trên.
- Tham số: subj_id, aid, pids
- Ví dụ:

```
$photos = $facebook->api_client->photos_get(null, $pid, null);
```


❖ Photos.getAlbum()

- Mô tả: Trả về siêu dữ liệu về tất cả các album ảnh được tải lên bởi người dùng cụ thể. Phương thức này trả về thông tin của tất cả các album dựa theo tiêu chí lọc. Nó có thể được sử dụng để trả lại tất cả các album ảnh được tạo ra bởi người dùng, truy vấn một danh sách các album thông qua aids hoặc lọc theo cả 2 tiêu chí trên.

- Tham số: uid, aids

- Ví dụ:

```
$albums = $facebook->api_client->photos_getAlbums($uid);
```

❖ Photos.getTags()

- Mô tả: Trả về danh sách người dùng được đánh dấu của các hình ảnh cụ thể.

- Tham số: pids

- Ví dụ:

```
$result = $facebook->api_client->photos_getTags($pids);
```

❖ Photos.upload()

- Mô tả: upload một ảnh thuộc sở hữu của người dùng cụ thể hoặc người dùng hiện hành và trả về là một ảnh mới. Các loại ảnh hỗ trợ như: gif, jpg, png, psd, tiff, jp2, iff, wbmp, xbm.

- Tham số: path

- Ví dụ:

```
$result = $facebook->api_client->photos_upload($filename, $aid, $caption);
```

▪ **Profile**

Để dễ dàng tương tác với các thiết lập thông tin trong trang hồ sơ của người dùng, nhóm hàm này đã cung cấp 2 phương thức:

❖ Profile.setFBML()

- Mô tả: Thiết lập FBML cho hồ sơ của người dùng hoặc trang Facebook , bao gồm: nội dung của hộp thông tin trên nhãn Wall, hộp thông tin cho nhãn Boxes.
- Ví dụ: `$facebook->api_client->profile_setFBML(NULL, $user, 'profile', NULL, NULL, 'profile_main');`

❖ Profile.getFBML()

- Mô tả: Trả lại FBML của hộp thông tin hiện đang thiết lập cho cấu hình của người dùng.
- Ví dụ:
`$fbml = $facebook->api_client->getFBML($uid);`

▪ **Users**

Facebook API cung cấp một số phương pháp cho phép nhà phát triển ứng dụng truy cập vào một số thông tin người dùng :

❖ Users.getInfo()

- Mô tả: Trả về thông tin của người dùng cụ thể
- Tham số: uids, fields
- Ví dụ:

```
$uids = array('12345');  
$fields = array('first_name', 'last_name');  
$users = $facebook->api_client->users_getInfo($uids, $fields);
```

❖ Users.getLoggedInUser()

- Mô tả: Trả về uid của người dùng
- Ví dụ:

```
$currentUser = $facebook->api_client->users_getLoggedInUser();
```

▪ **Error codes**

Khi phát triển ứng dụng, rất có thể sẽ xảy ra lỗi, Facebook đã cung cấp một nhóm API giúp người phát triển dễ dàng tìm ra lỗi trong quá trình xây dựng ứng dụng. Giá trị trả lại cả hai dạng số và tin nhắn.

2.2.1.4. Data Store API

Data store API chủ yếu thiết kế để giúp các nhà phát triển ứng dụng nhanh chóng tìm ra một giải pháp lưu trữ mở rộng, có thể theo kịp với sự tăng nhanh của cơ sở dữ liệu của họ. API này cung cấp các phương thức cơ bản như tạo, đọc, xóa, cập nhật.

Data Store API bao gồm 3 nhóm cơ bản : Specialized Tables, Distributed Tables, Associations, được phân chia vào 5 API riêng biệt (User Preference, Object Data Definition, Object Data Access, Association Data Definition, and Association Data Access).

- Association : Theo phương pháp tiếp cận cơ sở dữ liệu truyền thống, sử dụng chỉ số để tăng tốc độ truy vấn. Tuy nhiên phương pháp này lại không hiệu quả trong một hệ thống phân phối vì không có thư mục tập trung để đánh chỉ mục. Facebook đã giải quyết vấn đề này với Association. Association có 2 trường và có thể là một chiều hoặc hai chiều
 - Một chiều: cho phép ứng dụng liên kết một thông điệp tới người dùng.
 - Hai chiều: Cho phép ứng dụng liên kết những người nhận thông điệp.
- Specialized Tables: Một trong những điểm thuận lợi của việc sử dụng Data Store API là bảng chuyên dụng, được thiết kế và tối ưu hóa để lưu các loại dữ liệu cho ứng dụng và hiện nay mới chỉ có User Preference được triển khai.

2.2.1.5. Thư viện khách.

Thư viện khách cung cấp một đối tượng API khách của Facebook, một đối tượng ủy quyền để chuyển lời gọi phương thức thành các yêu cầu đến các máy chủ của Facebook. Nó hỗ trợ các ngôn ngữ lập trình như: ActionScript, ASP.Net, ASP(VBScript), ColdFusion, C++, C#, D, Emacs Lisp, Lisp, Perl, PHP(4 và 5), Python, Ruby, VB.Net và Windows mobile.

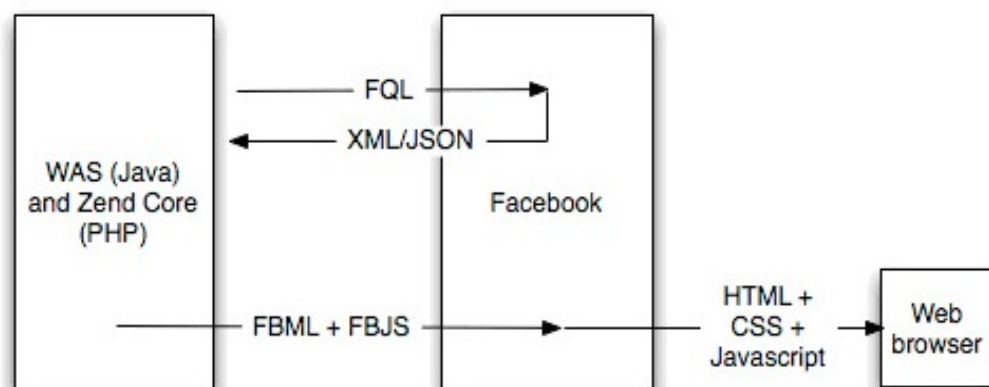
Thư viện khách PHP của Facebook bao gồm 2 đối tượng chính: Facebook (Facebook.php) và FacebookRestClient (facebook_api_php5_restlib.php) . Lớp FacebookRestClient tóm tắt những tương tác với API của Facebook. Lớp Facebook sử dụng các phương thức của lớp FacebookRestClient để tách ra các tương tác phổ biến với Facebook Platform.

2.2.2. Ngôn ngữ đánh dấu Facebook – FBML.

2.2.2.1. FBML là gì ?

FBML là ngôn ngữ đánh dấu do Facebook phát triển, là phần mở rộng của HTML, được sử dụng để hiển thị các trang bên trong trang vải nền Facebook. Hầu hết các thẻ FBML giống hệt với các thẻ HTML và dẫn đến nội dung trình duyệt giống nhau khi phân tích cú pháp, nhưng FBML cũng cung cấp nhiều thẻ hành vi và các thẻ trích ra thông tin từ các cơ sở dữ liệu Facebook và đưa nó vào trong nội dung của ứng dụng. FBML cũng hỗ trợ các yếu tố tạo style như CSS. Tuy nhiên FBML cũng bao gồm một số yếu tố mở rộng riêng cho Facebook.

FBML cho phép xây dựng một ứng dụng trên nền tảng Facebook một cách đầy đủ. Khi ứng dụng trả về FBML, đi qua proxy của Facebook, các thẻ FBML sẽ được render để trả về dạng HTML.



Hình 2.2: Dòng chảy thông tin từ máy chủ ứng dụng đến trình duyệt của người dùng

2.2.2.2. Nhóm các thẻ FBML

Các thẻ FBML được chia thành các nhóm:

- | | | |
|--------------------------------|----------------------|-----------------------------|
| ▪ User/Groups | ▪ MessageAttachments | ▪ Social Widgets |
| ▪ Tools | ▪ Editor Display | ▪ Additional Permissions |
| ▪ Status Messages | ▪ Misc | ▪ Dialog |
| ▪ PlatformInternationalization | ▪ Profile-specific | ▪ Notificationsand Requests |
| ▪ Visibility on Profile | ▪ Wall | ▪ Embedded Media |
| ▪ Page Navigation | ▪ Forms | |

▪ User/Groups

Làm việc với thông tin của người dùng và nhóm là phần rất quan trọng khi viết ứng dụng Facebook. Để dễ dàng tương tác với những người sử dụng ứng dụng, Facebook đã cung cấp nhóm user/groups cho người phát triển ứng dụng

Các thẻ cơ bản:

❖ <fb:name />

- Mô tả: Trả lại tên của người dùng, có thể dưới dạng link liên kết tới trang profile của người dùng.
- Thuộc tính: uid của người dùng. Ngoài ra còn có các tùy chọn như : firstnameonly, linked, lastnameonly, ..
- Ví dụ: <fb:name uid="12345" />

❖ <fb:user />

- Mô tả: tùy thuộc vào cài đặt bảo mật của người dùng , nội dung trả lại trong thẻ này chỉ hiện thị những người dùng có quyền được xem hồ sơ của người dùng mà họ quan tâm.
- Thuộc tính: uid của người dùng được bảo vệ nội dung
- Ví dụ
<fb:user id='4'>

The content rendered here will only display if you have permission to view Mark Zuckerberg's profile.

</fb:user>

❖ <fb:pronoun />

- Mô tả: Trả lại đại từ cho người dùng cụ thể
- Thuộc tính: uid của người dùng cần trả lại đại từ.
- Ví dụ:

```
<fb:name useyou="false" uid="683545112"/> just gave  
<fb:pronoun uid="actor" possessive="true"/> book to Fred.
```

⇒ Kết quả: Jesse Stay just gave *his* book to Fred

❖ <fb:profile-pic />

- Mô tả: Trả lại ảnh hiện tại của người dùng.
- Thuộc tính: uid của người dùng muốn hiện ảnh. Ngoài ra còn có các tùy chọn khác như : size, linked.
- Ví dụ:

```
<fb:profile-pic uid="4" size="square"/>
```

▪ Tools

Nhóm bao gồm tập các thẻ FBML, như một công cụ hỗ trợ người viết ứng dụng có thể xây dựng một phản ứng dụng một cách dễ dàng.

Các thẻ cơ bản:

❖ <fb:comments >

- Mô tả: Đây là thẻ để tạo ra Wall , là một thành phần giúp người dùng có thể viết bình luận lên trên đó.
- Thuộc tính: xid, canpost, candelete, numpost. Ngoài ra còn một số thuộc tính khác: callbackurl, returnurl, showform, send_notification_uid, publish_feed, simple, reverse.

- Ví dụ:

```
<fb:comments xid='titans_comments' canpost='true' candelete='true'  
  returnurl='http://apps.facebook.com/myapp/titans'>  
  <fb:title> talk about the Titans </title>  
</fb:comments>
```

❖ <fb: google-analytics>

- Mô tả: Đây là thẻ thêm Javascript để ứng dụng có thể sử dụng Google Analytics theo dõi việc sử dụng ứng dụng.
- Thuộc tính: uacct
- Ví dụ:

```
<fb:google-analytics uacct="UA-1279896-6" page="" />
```

❖ <fb:mobile>

- Mô tả: Nội dung hiện trong thẻ chỉ xuất hiện chỉ khi sử dụng di động hiện trang <http://m.facebook.com>. Thẻ này không làm việc trên Iphone của Apple.
- Ví dụ:

```
This text will not appear on the mobile phone.  
<fb:mobile>  
This text will appear on the mobile phone.  
</fb:mobile>
```

❖ <fb:random>

- Mô tả: Lấy ngẫu nhiên một mục bên trong thẻ dựa vào ‘weight’. Thẻ này chỉ lấy ngẫu nhiên. <fb:random> cho phép đầu vào là 1 tập các thẻ, trong đó có một hoặc nhiều được hiện thị ngẫu nhiên. Mỗi một thẻ có một ‘weight’.
- Ví dụ:

```
<fb:random> <fb:random-option> randomly selected text 1
  </fb:random-option> <fb:random-option> randomly selected
  text 2 </fb:random-option>
</fb:random>
```

▪ Status Messages

Đây là nhóm thẻ trả lại các thông báo cho người dùng dưới dạng các văn bản text.

❖ <fb:error>

- Mô tả: Trả lại thông báo lỗi. Nội dung trong thẻ được trả lại dưới dạng text.
- Thuộc tính: message
- Ví dụ:

```
<fb:error>
  This is the error message text.
</fb:error>
```

❖ <fb:explanation>

- Mô tả: Trả lại văn bản giải thích. Nội dung chứa trong thẻ được sử dụng như văn bản giải thích.
- Ví dụ:

```
<fb:explanation message="Explanation message" />
```

❖ <fb:success>

- Mô tả: Trả lại tin nhắn thành công
- Ví dụ:

```
<fb:success message="Success message" />
```

❖ <fb:message>

- Mô tả: Trả lại tiêu đề cho thông báo lỗi.
- Ví dụ:


```
<fb:message>
  This is the heading text for the message.
</fb:message>
```

▪ Visibility on Profile

Đây là nhóm thẻ mà Facebook cung cấp để giúp người phát triển hiển thị nội dung cụ thể tới người sử dụng dựa trên tình trạng hồ sơ của họ với ứng dụng. FBML cho phép phân biệt giữa chủ ứng dụng, người dùng, người sử dụng ứng dụng (người được cấp quyền đầy đủ để thực thi ứng dụng) và những người đã thêm ứng dụng vào tài khoản của họ.

❖ <fb:visible-to-owner>

- Mô tả: Chỉ hiển thị nội dung bên trong khi là chủ một trang hồ sơ hoặc admin của một trang. Thẻ này có thể chỉ được sử dụng trên trang hồ sơ của người dùng.
- Ví dụ:

```
<fb:visible-to-owner>
  Welcome back to your profile!
</fb:visible-to-owner>
```

❖ <fb:visible-to-friends>

- Mô tả: Sử dụng thẻ này để hiển thị nội dung bên trong thẻ trên hồ sơ của người dùng chỉ khi người xem là bạn của người dùng.
- Ví dụ:

```
<fb:visible-to-friends>Hello my friend!</fb:visible-to-friends>
```

❖ <fb:visible-to-app-users>

- Mô tả: Hiển thị nội dung kèm theo chỉ khi người xem đã được cấp quyền đầy đủ với ứng dụng
- Ví dụ:

```
<fb:visible-to-app-users>
  Welcome users!
</fb:visible-to-app-users>
```

▪ **Page Navigation**

Khi ứng dụng đã hoàn thành, nhà phát triển mong muốn tạo ra một chương trình chuyển hướng cho người dùng. Nhóm thẻ này là chính công cụ giúp thực hiện điều đó. Nó tạo ra một bảng điều khiển với các nút bấm, siêu liên kết và *help*.

❖ <fb:dashboard>

- Mô tả: Hiển thị bảng điều khiển chuẩn của Facebook có chứa các yếu tố như <fb:action>, <fb:create-button>, <fb:help>
- Ví dụ:

```
<fb:dashboard></fb:dashboard>
```

❖ <fb:header>

- Mô tả: Hiển thị tiêu đề
- Ví dụ:

```
<fb:header>My Photos</fb:header>
```

❖ <fb:tabs>

- Mô tả: Hiện nhóm các tab menu chuẩn của Facebook. Nội dung bên trong thẻ phải chứa ít nhất một <fb:tab-item>
- Ví dụ:

```
<fb:tabs>  
  <fb:tab-item href='' title='My Photos' selected='true' />  
  <fb:tab-item href='' title='Recent Albums' />  
</fb:tabs>
```

▪ **Editor display**

Để làm việc với các dữ liệu cần chỉnh sửa, các nhà phát triển Facebook đã tạo ra nhóm thẻ này để cho người viết ứng dụng dễ dàng thao tác với form chỉnh sửa. Form trả lại sẽ hiển thị ở 2 cột: cột bên trái là các label và cột bên phải là các trường nhập dữ liệu đầu vào. Nhược điểm của nhóm thẻ này là không hỗ trợ ajax. Nếu muốn sử dụng ajax người viết ứng dụng phải tự tạo ra các form riêng.

Ví dụ:

```

<fb:editor action="" labelwidth="100">
  <fb:editor-text label="Title" name="title" value=""/>
  <fb:editor-text label="Author" name="author" value=""/>
  <fb:editor-custom label="Status">
    <select name="state">
      <option value="0" selected>have read</option>
      <option value="1">am reading</option>
      <option value="2">want to read</option>
    </select>
  </fb:editor-custom>
  <fb:editor-textarea label="Comment" name="comment"/>
  <fb:editor-buttonset>
    <fb:editor-button value="Add"/>
    <fb:editor-button value="Recommend"/>
    <fb:editor-cancel />
  </fb:editor-buttonset>
</fb:editor>

```

⇒ Kết quả:

The screenshot shows a web form with the following elements:

- Title:** A single-line text input field.
- Author:** A single-line text input field.
- Status:** A dropdown menu with the selected option being "have read".
- Comment:** A multi-line text area.
- Buttons:** Three buttons labeled "Add", "Recommend", and "or Cancel" are positioned at the bottom right of the form.

▪ Misc

Đây là nhóm chứa tập hợp các thẻ FBML hỗn hợp.

❖ <fb:js-string>

- Mô tả: Cho phép gom các thẻ FBML thành một khối dưới dạng string để sử dụng trong FBJS. Khối thẻ này có thể đặt bất kỳ đâu mà không hiển thị lên trên trang.
- Thuộc tính: var
- Ví dụ:

```

<fb:js-string var="test_var">
  Hello <fb:name uid="loggedinuser"/>!
</fb:js-string>
<div id="test_div"></div>
<a href="#"
  onclick="document.getElementById('test_div').setInnerFBML(test_var)">
  Display test_div</a>

```

❖ <fb:title>

- Mô tả: Đặt lại tiêu đề của trang. Khi sử dụng bên trong thẻ <fb:comments>, đặt tiêu đề cho wall.
- Ví dụ:

```
<fb:title>This is the canvas page window title</fb:title>
```

❖ <fb:share-button>

- Mô tả: Hiển thị một nút chia sẻ trong hồ sơ cho các url cụ thể hoặc nội dung.
- Thuộc tính: class, href
- Ví dụ:

```
<fb:share-button class="url" href="http://apps.facebook.com/example" />
```

❖ <fb:time>

- Mô tả: Hiển thị ngày và giờ trong múi giờ của người dùng.
- Thuộc tính : t
- Ví dụ:

```
<fb:time t="1180502413" tz="America/New_York" />
```

▪ **Profile-specific**

Vì phải cung cấp những nội dung khác nhau tùy thuộc vào nơi người dùng truy cập ứng dụng, do vậy Facebook cung cấp một tập các thẻ để hiển thị nội dung bên trong hộp hồ sơ của người dùng.

Các thẻ cơ bản:

❖ <fb:wide>

- Mô tả: Nội dung đi kèm hiện thị chỉ khi hộp thông tin thuộc một cột rộng của hồ sơ
- Ví dụ:

```
<fb:wide>
  This will only appear in the wide column.
</fb:wide>
```

❖ <fb:narrow>

- Mô tả: Hiện thị nội dung chứa trong thẻ chỉ khi hộp thông tin trong cột hẹp của hồ sơ
- Ví dụ:

```
<fb:narrow>
  This text appears only in the narrow column.
</fb:narrow>
```

❖ <fb:user-table>

- Mô tả: Hiện thị một bảng, mỗi ô trong bảng chứa hình ảnh và tên của người dùng cụ thể. Bên trong thẻ này sử dụng thẻ <fb:user-item> để chỉ rõ người dùng. Thẻ này chỉ làm việc trên trang hồ sơ, không làm việc trên trang nền.
- Ví dụ:

```
<fb:user-table cols="3">
  <fb:user-item uid="12345" />
  <fb:user-item uid="123456" />
  <fb:user-item uid="123457" />
  <fb:user-item uid="123458" />
  <fb:user-item uid="123459" />
  <fb:user-item uid="123450" />
</fb:user-table>
```

❖ <fb:user-item>

- Mô tả: Trả lại một ô trong bảng, ô này có chứa hình ảnh và tên của người dùng cụ thể. Thẻ này phải được dùng trong thẻ <fb:user-item>

và chỉ hoạt động trên các trang hồ sơ, ứng dụng không thể sử dụng nó trên trang vãi nền.

- Thuộc tính: uid
- Ví dụ:

```
<fb:user-table cols="3">
  <fb:user-item uid="12345" />
  <fb:user-item uid="123456" />
  <fb:user-item uid="123457" />
  <fb:user-item uid="123458" />
  <fb:user-item uid="123459" />
  <fb:user-item uid="123450" />
</fb:user-table>
```

❖ <fb:subtitle>

- Mô tả: Định nghĩa tiêu đề phụ cho hộp thuộc tính. Nội dung đi kèm được thể hiện hoàn toàn như văn bản gốc. Thẻ này cũng có thể chứa thẻ <fb:action>
- Ví dụ:

```
<fb:subtitle seeallurl="" >
  <fb:action href="">Post a new fun graphic</fb:action>
  Displaying 10 images
</fb:subtitle>
```

▪ Wall

Đây là nhóm thẻ giúp nhà phát triển ứng dụng tạo ra các wall để người sử dụng có thể viết các bình luận

❖ <fb:wall>

- Mô tả: Giả lập một wall trên Facebook
- Ví dụ:

```
<fb:wall>
  <fb:wallpost uid='12345'> This is a wall post from uid 12345
</fb:wallpost>
</fb:wall>
```

❖ <fb:wallpost>

- Mô tả: Hiển thị wall dưới dạng post. Thẻ này có thể được sử dụng bên trong thẻ <fb:wall> . Nó cũng có thể chứa thẻ <fb:wallpost-action>
- Thuộc tính: uid
- Ví dụ:

```
<fb:wallpost uid="19235"> This tag </fb:wallpost>
```

❖ <fb:wallpost-action>

- Mô tả: Hiển thị một liên kết ở bên dưới wallpost
- Thuộc tính: href
- Ví dụ:

```
<fb:wall>  
  <fb:wallpost uid="12345">This is a wall post from uid 12345.  
    <fb:wallpost-action href="linktohere">link </fb:wallpost-action>  
  </fb:wallpost>  
</fb:wall>
```

▪ **Forms**

Làm việc với form thông tin là rất quan trọng trong việc phát triển bất kỳ ứng dụng trực tuyến. FBML có một số cấu trúc giúp cho người tạo ứng dụng dễ dàng trong việc thao tác với form thông tin.

Các thẻ cơ bản:

❖ <fb:request-form>

- Mô tả: Tạo một form để gửi yêu cầu. Đây là cách để người dùng ứng dụng gửi yêu cầu tới ứng dụng.
- Thuộc tính: type, content
- Ví dụ:

```

<fb:fbml>
  <fb:request-form action="index.php" method="POST"
    invite="true" type="YOUR APP NAME" content="Your text goes here" >
    <fb:multi-friend-selector showborder="false" actiontext="Invite your friends ">
  </fb:request-form>
</fb:fbml>

```

❖ <fb:request-form-submit>

- Mô tả: Tạo một nút submit với thẻ <fb:request-form>
- Thuộc tính: Có các tùy chọn như uid, label.
- Ví dụ:

```
<fb:request-form-submit />
```

❖ <fb:multi-friend-input>

- Mô tả: Tạo một khung văn bản tìm kiếm tên
- Ví dụ:

```

<form>
  <fb:multi-friend-input width="400px" border_color="#000"
    include_me="true" max="30" exclude_ids="4,1,2,3" prefill_ids="683545112"/>
</form>

```

❖ <fb:friend-selector>

- Mô tả: Hiện thị một form nhập, tự động thả xuống tên bạn bè của người dùng.
- Ví dụ:

```

<form>
<fb:friend-selector name="my_friend" idname="my_friend_id"
  include_me="true" exclude_ids="4,1,2" include_lists="true"/>
</form>

```

❖ <fb:submit>

- Mô tả: Tạo một nút submit cho form. Thẻ này thường sử dụng khi muốn thay nút submit thông thường bằng hình ảnh.
- Ví dụ:


```
<fb:submit>
  
</fb:submit>
```

▪ **Dialog**

Tập hợp các thẻ trong nhóm này hiện nay vẫn đang còn ở chế độ thử nghiệm. Nhưng cơ bản đây là một cơ chế tạo ra các hộp thoại cho ứng dụng để tương tác với người dùng.

❖ <fb:dialog>

- Mô tả: Là thẻ chứa của hộp thoại. Trong thẻ này bắt buộc phải có thẻ <fb:dialog-content>. Ngoài ra còn có thể có thẻ <fb:dialog-title> và <fb:dialog-button>
- Thuộc tính: id

❖ <fb:dialog-content>

- Mô tả: là thẻ chứa nội dung của hộp thoại

❖ <fb:dialog-title>

- Mô tả: là thẻ chứa tiêu đề của hộp thoại

❖ <fb:dialog-button>

- Mô tả: thẻ này giúp thêm các nút bấm vào bên dưới của hộp thoại

Ví dụ:

```
<fb:dialog id="fb_test">
  <fb:dialog-title>This is a test</fb:dialog-title>
  <fb:dialog-content>Content</fb:dialog-content>
  <fb:dialog-button type="button" value="Okay" close_dialog="
</fb:dialog>
```

▪ **Notifications and requests**

Đây là nhóm thẻ cung cấp các form yêu cầu, các form giới thiệu bằng cách sử dụng một tập các thẻ FBML. Ngoài ra nó còn cung cấp các thẻ hỗ trợ để tạo ra thông báo và chỉ hoạt động bên trong hàm của API

❖ <fb:notif-subject>

- Mô tả: Chỉ rõ chủ đề của email được gửi bằng hàm API: notification.send()

- Ví dụ:

```
<fb:notif-subject>This is the subject of the E-mail</fb:notif-subject>
```

❖ <fb:notif-page >

- Mô tả: Ghi rõ nội dung của thông báo, được hiển thị trên trang thông báo của người dùng. Nó chỉ làm việc bên trong lời gọi hàm API
- Ví dụ:

```
<fb:notif-page>  
  Add the <a href="http://apps.facebook.com/fbmlessentials">  
  FBML Essentials app!</a>  
</fb:notif-page>
```

❖ <fb:request-form>

- Mô tả: Hiển thị một form giới thiệu chuẩn của Facebook
- Thuộc tính: type, content
- Ví dụ:

```
<fb:request-form action="index.php" method="POST" invite="true"  
  type="FBML Essentials" content="Hello Friend. <fb:req-choice  
  url='http://apps.facebook.com/fbmlessentials' label='Go there!'/>">  
  <fb:multi-friend-selector showborder="false"  
    actiontext="Invite your friends to use FBML Essentials.">  
</fb:request-form>
```

❖ <fb:multi-friend-selector>

- Mô tả: Hiển thị đầy đủ bạn của người dùng với ảnh , tên
- Thuộc tính: actiontext
- Ví dụ:

```
<fb:request-form action="index.php" method="POST" invite="true"
  type="FBML Essentials" content="Hello Friend.">
  <fb:multi-friend-selector condensed="true" exclude_ids="1,4,5,6" />
  <fb:request-form-submit/>
</fb:request-form>
```

▪ **Embedded media**

Người phát triển ứng dụng cần nhúng media vào ứng dụng của họ như âm nhạc, video, game, ...Hiểu rõ điều này, các nhà phát triển Facebook đã tạo ra nhóm thẻ *embedded media* để hỗ trợ cho thao tác này.

❖ <fb:iframe>

- Mô tả: Chèn iframe vào ứng dụng.
- Thuộc tính: src

❖ <fb:photo>

- Mô tả: Hiển thị một ảnh trên Facebook.
- Thuộc tính: pid
- Ví dụ:

```
<fb:photo pid="12345" />
```

❖ <fb:mp3>

- Mô tả: Hiển thị công cụ nghe nhạc cơ bản
- Thuộc tính: src
- Ví dụ:

```
<fb:mp3 src="http://host.com/file.mp3"
  title="Song Name"
  artist="Song Artist" />
```

2.2.2.3. CSS

Mặc dù tạo ra FBML dựa trên nền tảng của HTML, thế nhưng các nhà phát triển Facebook lại không tạo ra ngôn ngữ hỗ trợ CSS cho người phát triển ứng dụng. CSS được dùng trong Facebook hoàn toàn tương tự như dùng trong các trang web thông thường. Cũng như hình ảnh, CSS được lưu trữ bởi máy chủ của Facebook, do

vậy nếu thay đổi nội dung file CSS, bắt buộc phải thay đổi tên của file nếu không máy chủ Facebook sẽ làm việc trên file cũ.

2.2.3. Ngôn ngữ truy vấn – FQL.

2.2.3.1. FQL là gì ?

Ngôn ngữ truy vấn của Facebook (FQL) là một ngôn ngữ dựa trên ngôn ngữ SQL, được các nhà phát triển Facebook tạo ra để giúp người viết ứng dụng truy xuất tới các bảng trong cơ sở dữ liệu bao gồm các bảng : user, friend, group, group_member, event, event_member, photo, album, and photo_tag..

2.2.3.2. Cấu trúc câu truy vấn FQL.

Do FQL dựa trên nền tảng là ngôn ngữ SQL chính vì vậy cấu trúc của FQL cũng tương tự như cấu trúc của SQL :

```
SELECT [ trường ] FROM [ bảng ] WHERE [ điều_kiện]
```

Trong FQL, các mệnh đề trong SQL như ORDER BY, LIMIT cũng được hỗ trợ:

```
SELECT [ trường ] FROM [ bảng ] WHERE [ điều_kiện]
```

```
ORDER BY[ trường] LIMIT [vị trí bắt đầu], [vị trí kết thúc]
```

Tuy dựa trên nền tảng của SQL nhưng FQL cũng có những điểm khác biệt. Đầu tiên là phải kể đến mệnh đề FROM. Nếu như trong SQL sau from có thể là một hoặc nhiều bảng để truy xuất dữ liệu nhưng trong FQL sau from chỉ có duy nhất một bảng. Như vậy trong FQL không hỗ trợ các kiểu join. Để có thể lấy dữ liệu từ nhiều bảng, cách phổ biến nhất là sử dụng các câu truy vấn con. Ví dụ nếu muốn lấy danh sách những người bạn của người dùng đang sử dụng ứng dụng câu truy vấn FQL có thể là:

```
SELECT uid FROM user WHERE has_added_app = 1 AND uid IN  
(SELECT uid2 FROM friend WHERE uid1=$user )
```

trong đó \$user là uid của người dùng hiện tại. Thứ hai, để các ứng dụng truy xuất cơ sở dữ liệu một cách trực tiếp, tất cả các truy vấn phải được đánh chỉ số, tránh cho việc có quá nhiều câu truy vấn gửi tới server. Facebook chỉ cho phép người tạo ứng dụng truy xuất trên 45 bảng dữ. Thứ ba, trong mệnh đề where, ít nhất phải có một

trường được đánh chỉ số. Ví dụ nếu muốn tìm tất cả những người dùng đang chia sẻ ngày sinh nhật của người dùng hiện tại:

```
SELECT uid FROM user WHERE strpos(birthday, "September 27") = 0  
AND uid IN (SELECT uid2 FROM friend WHERE uid1 = $user)
```

Tiếp theo, để giảm một lượng lớn dữ liệu trao đổi, FQL không hỗ trợ SELECT *. Cuối cùng, mệnh đề ORDER BY chỉ hỗ trợ một trường đơn duy nhất, không hỗ trợ nhiều trường như trong SQL.

2.2.3.3. Các toán tử và các hàm hỗ trợ cơ bản.

Như chúng ta đã biết, FQL là ngôn ngữ truy xuất cơ sở dữ liệu của Facebook dựa trên ngôn ngữ SQL. Mặc dù FQL không phải là chuẩn ANSI-SQL nhưng nó cũng có đầy đủ các toán tử đơn và các hàm để giúp người viết ứng dụng làm việc với cơ sở dữ liệu. FQL có các toán tử boolean (AND, OR, NOT), các toán tử so sánh (=, >, >=, <, <=, <>) và các toán tử toán học (+, -, *, /). FQL cũng hỗ trợ một số các hàm cơ bản, những hàm này rất quen thuộc với những nhà phát triển PHP:

- | | |
|---------------------------------|--|
| ▪ Now() | Trả lại thời gian hiện tại |
| ▪ Rand() | Tạo ra một số ngẫu nhiên |
| ▪ Strlen(string) | Trả lại độ dài của một chuỗi |
| ▪ Concat(string, ..) | Nối các chuỗi với nhau |
| ▪ Substr(string, start, length) | Trả lại chuỗi con |
| ▪ Strpos(haystack, needle) | Trả lại vị trí của needle trong haystack |
| ▪ Lower(string) | Chuyển đổi chuỗi sang mẫu tự thường |
| ▪ Upper(string) | Chuyển đổi chuỗi sang mẫu tự hoa |

Người viết ứng dụng có thể sử dụng trực tiếp các hàm hỗ trợ trong câu truy vấn. Ví dụ:

```
SELECT upper(concat(first_name, " ", substr(last_name, 0, 1), ".")), birthday
FROM user
WHERE sex = "female" AND uid IN (SELECT uid2
FROM friend
WHERE uid1 = $user) LIMIT 5;
```

2.2.3.4. Các bảng và các trường được đánh chỉ mục.

Như đã nói ở trên, Facebook cho phép người viết ứng dụng thao tác trên 45 bảng dữ liệu. Mỗi bảng dữ liệu sẽ có các trường được đánh chỉ số nhằm hỗ trợ cho việc sử dụng mệnh đề where trong câu truy vấn FQL. Dưới đây là danh sách một số các bảng và các trường được đánh chỉ mục:

- Album : Lưu các album được tạo trên ứng dụng Facebook photos. Các trường được đánh chỉ số:
 - Aid : id của album
 - Conver_pid: id của photo được sử dụng trên trang giới thiệu.
 - Ownerid: id của người tạo ra album
- Cookies: Trả lại cookies cho ứng dụng. Trường được đánh chỉ số:
 - Uid: id của người dùng
- Event: Lưu các sự kiện được tạo ra trong ứng dụng Facebook events. Trường được đánh chỉ số:
 - Eid: id của sự kiện
- Event member: Bảng mối quan hệ để lưu trạng thái của người dùng đặc biệt. Kiểm tra bảng này để tìm ra hồi đáp RSVP của người dùng cho một sự kiện. Các trường được đánh chỉ số:
 - Uid: id của người dùng
 - Edi: id của sự kiện
- Friend: Bảng mối quan hệ để lưu bạn bè của người dùng. Các trường được đánh chỉ số:
 - Uid1

- Uid2
- Friend_request: Truy vấn bảng này để xác định người dùng đã gửi yêu cầu kết bạn tới người dùng hiện tại hoặc để truy vấn một yêu cầu kết bạn đã được gửi từ người dùng hiện tại cho một người dùng cụ thể. Trường được đánh chỉ số:
 - Uid_to: id user của người dùng đã nhận được lời mời kết bạn
- Friendlist: Lưu danh sách bạn bè của người dùng hiện tại. Trường được đánh chỉ số:
 - Owner: Id user của người dùng hiện tại
- Friendlist_member: Truy vấn bảng này để xác định người dùng là thành viên của một danh sách bạn bè. Trường được đánh chỉ số:
 - Flid: id của danh sách bạn bè
- Group: Truy vấn bảng này trả về thông tin của nhóm. Trường được đánh chỉ số:
 - Gid: id của nhóm.
- Group_member: Truy vấn bảng này để trả lại thông tin về thành viên của nhóm. Các trường được đánh chỉ số:
 - Uid: id user của thành viên trong nhóm được truy vấn.
 - Gid: id của nhóm được truy vấn
- Page: Truy vấn bảng này trả lại thông tin về một trang Facebook. Các trường được đánh chỉ số:
 - Page_id: id của trang được truy vấn.
 - Name: tên của trang được truy vấn.
- Page_fan: Truy vấn bản này trả lại thông tin về người hâm mộ của một trang Facebook. Trường được đánh chỉ số:
 - Uid: id user của người hâm mộ

- Photo: Truy vấn bảng này trả lại thông tin về một bức ảnh. Các trường được đánh chỉ số:
 - Pid: id của bức ảnh
 - Aid: id của album chứa bức ảnh được truy vấn.
- Photo_tag: Truy vấn bảng này trả lại thông tin về những ảnh đã được đánh dấu bởi người sử dụng. Các trường được đánh chỉ số:
 - Pid: id của ảnh đang được truy vấn.
 - Subject: Với những người dùng được đánh dấu, sử dụng id user của các đối tượng được đánh dấu trong ảnh. Đối với các ảnh gắn với các sự kiện hoặc các nhóm, sử dụng eid , gid.
- User : Truy vấn bản này trả lại thông tin từ hồ sơ của người dùng. Trường được đánh chỉ số:
 - Uid: id của người được truy vấn.

2.2.4. Ngôn ngữ tạo script – FBJS

2.2.4.1. FBJS là gì ?

Trong ứng dụng web thông thường, Javascript là công cụ hỗ trợ các nhà phát triển tạo ra các script để thêm tương tác cho các trang HTML, giảm tải các yêu cầu tới server. Các nhà phát triển của Facebook hiểu rõ điều này. Họ tạo ra FBJS là ngôn ngữ dựa trên Javascript để tạo ra các script giúp cho người dùng xây dựng ứng dụng một cách dễ dàng hơn.

FBJS là một hình thức rất hạn chế của Javascript mà Facebook đưa ra để ngăn chặn các mã độc hại trong các ứng dụng. Vấn đề bảo mật được các nhà phát triển Facebook quan tâm hàng đầu, do vậy FBJS chính là công cụ để kiểm soát vấn đề này. Cũng như FBML, FBJS được phân tích và viết lại trước khi được trả lại cho người dùng.

2.2.4.2. FBJS làm việc như thế nào ?

FBJS cũng tương tự như javascript. Hầu hết các mã trong JavaScript sẽ không cần chỉnh sửa trong FBJS. Tuy nhiên, có một số các hàm, các đối tượng bị hạn chế

vì mục đích an ninh. Ví dụ đối tượng `object.setInnerHTML()` trong javascript là `document.getElementById('div').innerHTML = 'content'` khi chuyển sang FBJS sẽ là `document.getElementById('div').setInnerHTML('content')`.

2.2.4.3. Ajax

Facebook cung cấp một lớp Ajax giúp người phát triển có thể xử lý để thực hiện những lời gọi Ajax đơn giản. Lớp Ajax cung cấp các thuộc tính sau:

- `Ondone(data)` : Bộ quản lý sự kiện được gọi khi lời gọi Ajax được trả về. Dữ liệu có thể là đối tượng, xâu hoặc dạng FBML.
- `Onerror`: Bộ quản lý sự kiện được gọi khi lỗi được tạo ra bởi các lời gọi Ajax.
- `requireLogin`: Nếu thiết lập là true, người dùng phải đăng nhập trước khi một lời gọi Ajax được gửi tới máy chủ của ứng dụng. Khi đã đăng nhập, nó sẽ vượt qua giá trị `fb_sig` trở lại server của ứng dụng như một phần của lời gọi.

Ngoài các thuộc tính, lớp Ajax còn cung cấp các phương thức:

- `post(url, query)`: gửi lời gọi ajax tới một link với tham số truy vấn bất kì.
- `abort()`: hủy bỏ lời gọi ajax

CHƯƠNG 3 : XÂY DỰNG ỨNG DỤNG TRÊN NỀN FACEBOOK

=====*****=====

1.1. Mô tả ứng dụng

Hiện nay, Facebook đã cung cấp một chức năng xem ảnh của bạn bè của người dùng hiện tại, tuy nhiên có một điểm bất lợi là nếu người dùng muốn xem ảnh của nhiều người thì phải vào trang cá nhân tương ứng của họ. Điều này là khá mất công và tốn thời gian nếu người dùng cùng lúc muốn xem album ảnh của nhiều người. Từ thực tế đó, tôi đã phát triển một ứng dụng trên nền Facebook giúp người dùng có thể xem ảnh của bạn bè mình mà hoàn toàn không phải truy cập vào trang cá nhân của từng người.

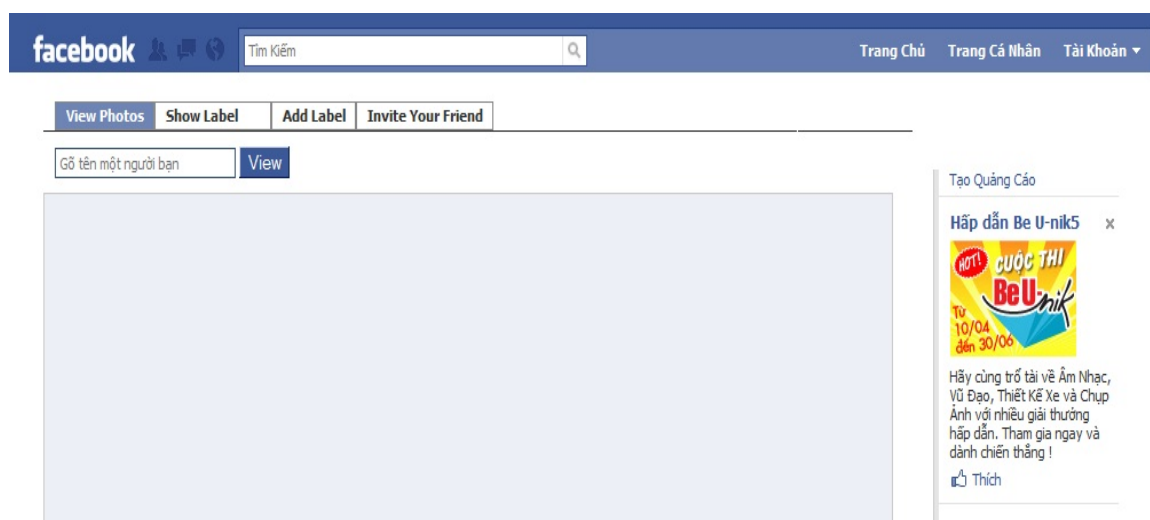
Ứng dụng xây dựng được là một ứng dụng dùng để hiển thị ảnh trong album của bạn của người dùng hiện tại đang đăng nhập vào Facebook. Ứng dụng này thực sự hữu ích nếu người dùng muốn xem nhiều album ảnh của bạn bè mà không muốn vào từng trang cá nhân của họ. Chức năng chính của ứng dụng là xem theo từng album ảnh của bạn bè người dùng. Với giao diện thân thiện, theo dạng trình chiếu slide, người dùng có thể dễ dàng xem ảnh bằng cách click chuột vào từng ảnh. Ngoài ra, người dùng có thể tạo nhãn riêng của mình theo chủ đề và thêm các ảnh yêu thích vào các chủ đề đó. Chức năng này là khá mềm dẻo khi người dùng muốn xem lại các ảnh mình thích theo các chủ đề mà mình đã chọn. Người dùng cũng có thể di chuyển các ảnh giữa các chủ đề hoặc có thể xóa đi nếu không thích. Dưới đây là danh sách các chức năng hiện có trong ứng dụng:

- View Photos: Đây là chức năng chính của ứng dụng, giúp người dùng xem album ảnh của bạn bè bằng cách nhập tên.
- Show Label: Hiển thị các chủ đề mà người dùng đã tạo ra, nếu muốn xóa một chủ đề, người dùng hoàn toàn có thể xóa.
- Add Label: Thêm chủ đề mà người dùng mong muốn. Sau khi thêm chủ đề mới, chủ đề đó sẽ được hiển thị ngay bên chức năng Show Label.
- Invite your friend: Chức năng này giúp người dùng có thể giới thiệu với bạn bè mình về ứng dụng View Photos bằng cách gửi một thông báo tới bạn mà người dùng chọn trong danh sách.

- Add : Chức năng này giúp người dùng có thể thêm ảnh trong album của bạn vào các chủ đề tùy chọn. Sau đó người dùng có thể xem ảnh đã thêm theo các chủ đề mà không phải mất công vào trang cá nhân của bạn bè họ. Một ảnh có thể được thêm vào nhiều chủ đề.
- Move : Di chuyển các ảnh giữa các chủ đề. Các ảnh được thêm vào từng chủ đề có thể di chuyển linh hoạt giữa các chủ đề. Mỗi lần di chuyển, ảnh sẽ không còn thuộc chủ đề đó nữa.

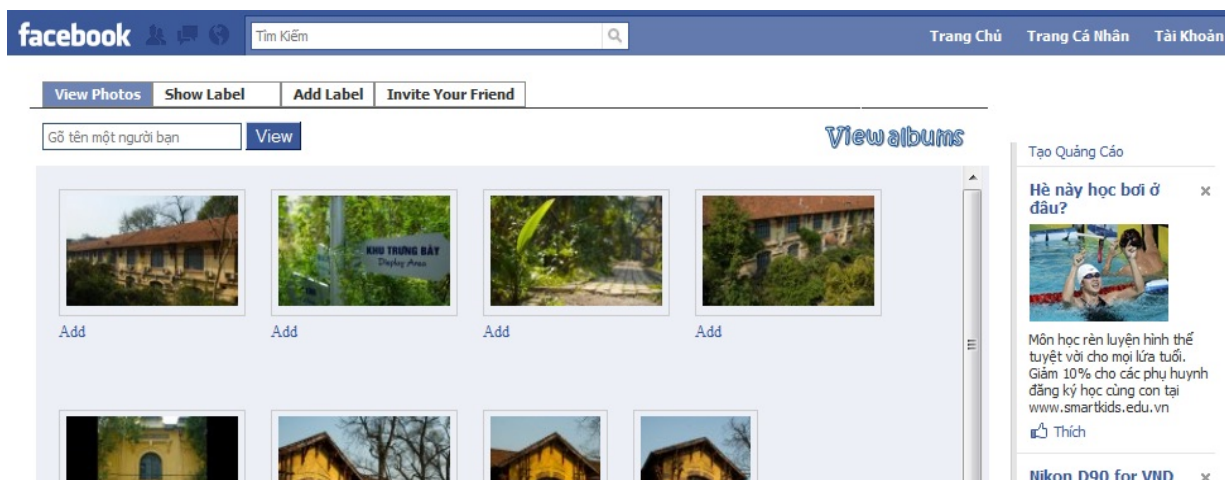
Ngoài ra còn một số chức năng khác như xóa chủ đề đã tạo, xóa ảnh trong chủ đề.

ViewPhoto là một ứng dụng được viết trên nền CentOS 5.4 x86_64 với ngôn ngữ lập trình được chọn là PHP. Ứng dụng có sử dụng các đối tượng mà Facebook cung cấp như API, FBML, FBJS, FQL. Bên cạnh đó, CSS cũng được sử dụng để tạo style cho ứng dụng. Dưới đây là một số hình ảnh về ứng dụng

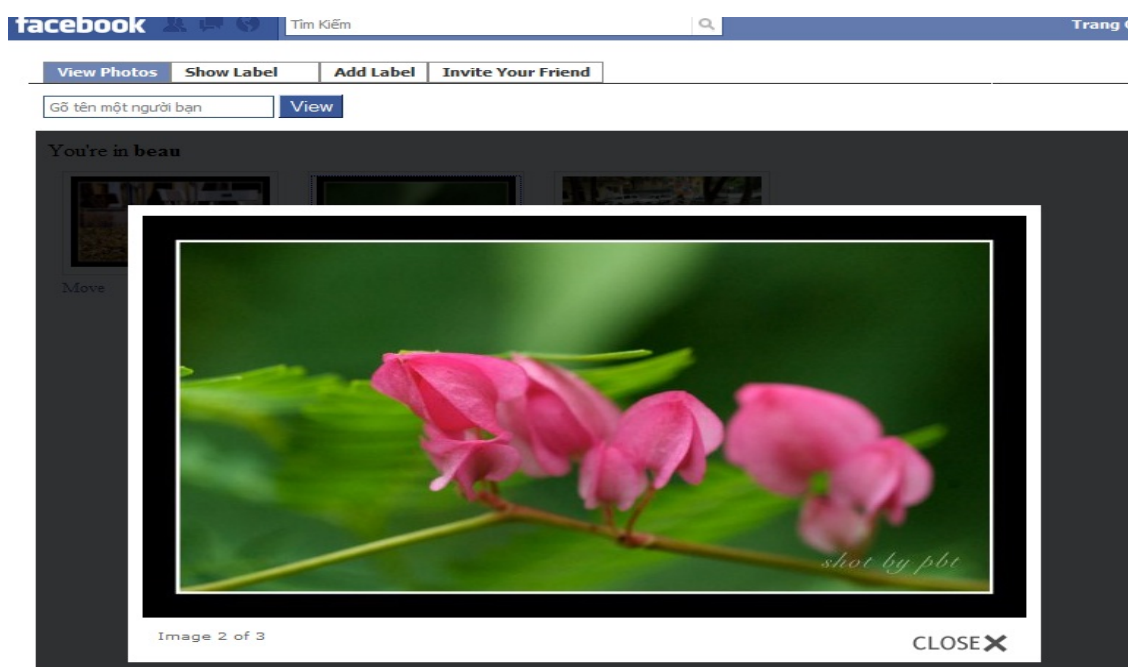


Giao diện chính

*Hiển thị nhãn**Thêm nhãn*



Hiển thị theo album

*Hiển thị theo nhãn**Giao diện xem ảnh*

1.2. Cài đặt

- Ứng dụng được đặt tại server có địa chỉ IP: 123.30.50.210.
- Địa chỉ ứng dụng trên Facebook : <http://apps.facebook.com/hienhoang/>

KẾT LUẬN

=====*****=====

Trong khóa luận này, bước đầu tôi 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. Tôi cũng đã trình bày về Facebook Platform – nền tảng của Facebook. Đây cũng chính là thành phần quan trọng nhất giúp người viết ứng dụng có thể tạo dựng ứng dụng một cách dễ dàng. Đi sâu tìm hiểu về nền tảng của Facebook, tôi đã bước đầu hiểu được cách thức tạo dựng ứng dụng trên nền Facebook, hiểu được cấu trúc các thẻ FBML, cách sử dụng chúng thay thế các thẻ HTML thông thường và cách lấy dữ liệu bằng các hàm API và FQL. Cuối cùng tôi đã xây dựng một ứng dụng cụ thể được viết trên nền Facebook. Đây là ứng dụng đã sử dụng nền tảng mà Facebook cung cấp như đối tượng API, FBML, FBJS và FQL. Tuy nhiên sau khi viết ứng dụng tôi 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, sau mỗi mệnh đề Where phải có trường được đánh chỉ số,
- FBJS, CSS không được Facebook nhận một cách nhanh chóng.

Trong khóa luận này tôi 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 nền tảng mà Facebook cung cấp, từ đó có thể dễ dàng tạo ra các ứng dụng linh hoạt và hiệu quả hơn.

PHỤ LỤC

Danh sách các file có trong chương trình:

PHP	CSS	Js
Thư mục Includes: config.php facebook.php facebookapi_php5_restlib.php Thư mục db config.php DBUtiliti.php Db.php DelPhoto.php Index.php Invite.php LabelToLabel.php Lightbox.php LightboxLabel.php PhotoByLabel.php PhotoIntoLabel.php ViewFriend.php	JQuery.lightbox-0.5.css Main.css Main_IE.css TabStyle.css	JQuery.lightbox-0.5.js JQuery.js ViewFriend.js Del_Move.js IndexPage.js

Index.php

```

<link rel="stylesheet" type="text/css" media="screen"
      href="http://123.30.50.210/~hienhoang/css/TabStyle.css?v=1.0" />
<link rel="stylesheet" type="text/css" media="screen"
      href="http://123.30.50.210/~hienhoang/css/Main.css?v=1.0" />
<!--[if IE]>
<link rel="stylesheet" type="text/css" media="screen"
      href="http://123.30.50.210/~hienhoang/css/Main_IE.css?v=1.0" />
<![endif]>
<script type="text/javascript" src="/js/Del_Move.js"></script>
<script src="http://123.30.50.210/~hienhoang/js/indexPage.js"> </script>
<?php
    include_once("./includes/config.php");
    include_once("./db/config.php");
    include_once("./db/DBUtiliti.php");
    connectDB($dbhost, $dbuser, $dbpass, $db);
?>
<div id='Main'>
<div id='Head'>
<div id='tab'>
    <ul id='itemTab'>
        <li id='first' value = '1'><a href='#' id='a1' onclick='selected_tab("1");'>View
            Photos</a></li>
        <li id='menuLabel'><a href='#'> Show Label </a>
    </ul>
<?php
$sql = "SELECT * FROM Label where uid=$user";
$result_showLB = mysql_query($sql);
$rows = mysql_num_rows($result_showLB);
for($i = 0; $i < $rows; $i++)
{
    $r = mysql_fetch_array($result_showLB);
    $name_label = $r['nameLabel'];
    $name_label = trim($name_label);
    $name_label = ucfirst($name_label);
    $idLabel = $r['idLabel'];
    echo "<li ><a href='#' onclick='\"photoByLabel(\".$idLabel.\");\" >$name_label</a>
        <span href='#' id='delete' onclick='\"delLabel();\"> <img
            src='http://123.30.50.210/~hienhoang/Images/del.ico' /></span>
        </li>";
    echo "<form action = 'index.php' id = 'del_Label' method = 'post' id = 'form_del'>
        <input type = 'hidden' value = \"\".$idLabel.\" name = 'hidden_idLabel' />
        </form>";
}
?>
</ul>

```

```

        </li>
        <li id='second' value='2'><a href='#' id='a2' onclick='selected_tab("2");'>Add Label</a></li>
        <li id='third' value='3'><a href='invite.php' >Invite Your Friend</a></li>
    </ul>
    <div id='menuShowLabel'></div>
</div>
<?php
    if(isset($_POST['hidden_idLabel']))
    {
        $idLabel = $_POST['hidden_idLabel'];
        $sql = "DELETE FROM Label WHERE idLabel='".$idLabel."'";
        $result = mysql_query($sql) or die('not delete');
    }
?>
<script>document.setLocation('http://apps.facebook.com/hienhoang/index.php');</script>
<?
    }
?>
<!-- form nhap ten friend -->
<div id='Friend'>
<?php
echo '<form action="index.php" id="friend_Form" method="post">
    <fb:friend-selector uid='.$user.' name="friendid" idname="friend_sel" include_me="true" />
    <button type="submit"> View</button>
</form>
';
?>
<!-- form add -->
</div>
<div id='addlabel'>
<form action='index.php' method='POST'>
    <fb:editor-text label=" name='label' value="" maxlength=50 />
    <button type="submit"> Add </button>
</form>
</div>
<?php
    if(isset($_POST['label']))
    {
        $label = $_POST['label'];
        $sql = "INSERT INTO Label VALUES (NULL, '".$label."', $user)";
        $result = mysql_query($sql);
    }
?>
    <script>document.setLocation('http://apps.facebook.com/hienhoang/index.php');</script>
<?php
    }
?>
<div id='Albums'><img src='http://123.30.50.210/~hienhoang/Images/viewalbums.png'
    onclick='shAlbum();' /></div>
</div>
<div id='Left' >
<div id = 'showAlbums'>

```

```

<!-- albums view here -->
<?php
if(!empty($_POST['friend_sel']))
{
    $fid = $_POST['friend_sel'];
    echo "<div class='TitleAlbums'>Albums's <fb:name uid=$fid /></div>";
    $albums = $facebook->api_client->photos_getAlbums($fid);
    echo "<input type='hidden' id='fid_hidden' value='".$fid.'" />";

    foreach($albums as $album)
    {
        $aid = $album['aid'];
        $nameAlbum = $album['name'];
        $photos = $facebook->api_client->photos_get(null,$aid,null);
        foreach ($photos as $photo)
        {
            $img = $photo['src'];
            echo" <div class='div_album'>
<div class='imgAlbum'>
    <a href='#' onclick=\"show('".$aid."');\"><img src=$img /></a>
</div>
<div class='nameAlbum'>
    <a href='#' onclick=\"show('".$aid."');\"><font
                                color='#3B5998'><b>$nameAlbum</b></font></a>
</div>
</div>";
            break;
        }
    }
}
?>
</div>
</div>
<!-- right -->
<div id='Right'>
    <div id='HeadRight'><div id='IconWait' ></div></div>
    <div id='Right_content'></div>
</div>
</div>

```

ViewFriend.php

```

<script type="text/javascript" src="http://123.30.50.210/~hienhoang/js/jquery.js"></script>
<script type="text/javascript" src="http://123.30.50.210/~hienhoang/js/jquery.lightbox-0.5.js"></script>
<link rel="stylesheet" type="text/css" href="http://123.30.50.210/~hienhoang/css/jquery.lightbox-0.5.css"
        media="screen" />
<link rel="stylesheet" type="text/css" media="screen"
        href="http://123.30.50.210/~hienhoang/css/Main.css?v=1.0" />
<!--[if IE]>
<link rel="stylesheet" type="text/css" media="screen"
        href="http://123.30.50.210/~hienhoang/css/Main_IE.css?v=1.0" />

```

```

<![endif]-->
<script type="text/javascript" src="./js/Del_Move.js"></script>
<script src='http://123.30.50.210/~hienhoang/js/indexPage.js'> </script>
<script type="text/javascript">
    $(function() {
        $('#gallery a').lightBox();
    });
</script>

<?php
include_once("./includes/config.php");
include_once("./db/config.php");
include_once("./db/DBUtiliti.php");
connectDB($dbhost, $dbuser, $dbpass, $db);

if(isset($_GET['fid']) && isset($_GET['aid']))
{
    $fid = $_GET['fid'];
    $aid = $_GET['aid'];
    /*
    $sql = "SELECT created,name FROM album WHERE aid=$aid ";
    $albums = $facebook->api_client->fql_query($sql);
    foreach($albums as $album)
    {
        //$nameAlbum = $album['name'];
        //echo "<div >Albums's $nameAlbum</div>";
        //echo $nameAlbum." -- ";
        echo $album;
        foreach($album as $al)
        echo $al;
    }
    */

    $sql = "SELECT pid,src_big,src,caption FROM photo WHERE aid=$aid AND owner=$fid";
    $photos = $facebook->api_client->fql_query($sql);
    echo "<div id='gallery'>";
    foreach($photos as $photo)
    {
        $pid = $photo['pid'];
        $img_big = $photo['src_big'];
        $img = $photo['src'];
        $caption = $photo['caption'];
        echo $caption;
        echo "<div class='div_album' >
            <div class='imgAlbum'>
                <a href=$img_big title='\".$caption.\"'><img src=$img /></a>
            </div>
            <div id='add\".$pid.\"' style='font-size:12px;color: #3B5998;cursor:pointer; width:
                25px;height: auto;margin-right: 2px;float:left;'
                onclick='\"showChooseLabel(\".$pid.\"');\"><span>Add</span>
        </div>";
    }
}

```

```

        </div>
        <div id = $pid style='display:none;'>
            <select id='label_choose'".$pid.'" name='label_choose'
onchange=\"addLabel(\".$pid.\"),\"style='background-color: #ECEFF5;' value='hihi'><option
selected> Go to</option>\";
            $sql = "SELECT * FROM Label where uid=$user";
            $result_showLB = mysql_query($sql);
            $rows = mysql_num_rows($result_showLB);
            for($i = 0; $i < $rows; $i++)
            {
                $r = mysql_fetch_array($result_showLB);
                $name_label = $r['nameLabel'];
                $idLabel = $r['idLabel'];
                echo "<option value= $idLabel>$name_label</option>";
            }
            echo"
                                </select>
                            </form>
                        </div>
                    </div>";
        }
        echo "<div class='alert' id='alert'>
            <div class='alert_content'>
                <div class='title_alert' > </div>
                <div class='divmess' id='mydiv'
                    style='border-bottom-width: 1px;
                    border-bottom-style: solid;
                    border-bottom-color: #cccccc;
                    background-color:#FFF;
                    width: 100%;
                    height: 40px;
                    text-align: center;
                    padding-top: 5px;'>
                </div>
                <div class='button_alert'>
                    <button style='width:75px;float: right;'
onclick=\"document.getElementById('alert').style.display='none'\" > OKey!</button>
                </div>
            </div>
        </div>";
        echo "</div>";
    }
?>

```

LightBox.php

```

<link rel="stylesheet" type="text/css" media="screen"
                                href="http://123.30.50.210/~hienhoang/css/Main.css?v=1.0" />
<?php
    $fid = $_GET['fid'];

```

```

        $aid = $_GET['aid'];
    ?>
    <fb:iframe src="http://123.30.50.210/~hienhoang/viewFriend.php?fid='<?echo $fid ?>'&aid='<?php echo
        $aid ?>'"
    frameborder="0" width=750 height=700 scrolling="auto"/>

```

PhotoByLabel.php

```

<script type="text/javascript" src="http://123.30.50.210/~hienhoang/js/jquery.js"></script>
<script type="text/javascript" src="http://123.30.50.210/~hienhoang/js/jquery.lightbox-0.5.js"></script>
<link rel="stylesheet" type="text/css" href="http://123.30.50.210/~hienhoang/css/jquery.lightbox-0.5.css"
    media="screen" />
<link rel="stylesheet" type="text/css" media="screen"
    href="http://123.30.50.210/~hienhoang/css/Main.css?v=1.0" />
<!--[if IE]>
<link rel="stylesheet" type="text/css" media="screen"
    href="http://123.30.50.210/~hienhoang/css/Main_IE.css?v=1.0" />
<![endif]-->
<script type="text/javascript" src=".js/Del_Move.js"></script>
<script src='http://123.30.50.210/~hienhoang/js/indexPage.js'> </script>
<script type="text/javascript">
    $(function() {
        $('#gallery a').lightBox();
    });
</script>

<?php
include_once("./includes/config.php");
include_once("./db/config.php");
include_once("./db/DBUtiliti.php");
connectDB($dbhost, $dbuser, $dbpass, $db);
//echo "hihi";
$label = $_GET['label'];
$label = substr($label,1,strlen($label)-2);
$sql = "SELECT * FROM Label WHERE idlabel= $label";
$rs = mysql_query($sql) or die (" not select");
$rows = mysql_num_rows($rs);
$r = mysql_fetch_array($rs);
$nameLabel = $r['nameLabel'];
echo "<div class='TitleAlbums'>You're in <b>$nameLabel</b> </div>";
echo "<input type='hidden' id='hidden_Idlabel' value=$label />";
echo "<input type='hidden' id='hidden_Namelabel' value='\". $nameLabel.\"' />";

$sql = "SELECT * FROM Photo,Label WHERE Photo.idlabel=$label AND uid=$user AND
    Photo.idlabel = Label.idlabel";
$result = mysql_query($sql) or die ("not select");
$rows = mysql_num_rows($result);
echo "<div id='gallery'>";

for($i = 0; $i< $rows; $i++)

```

```

{
    $r = mysql_fetch_array($result);
    $photo_id = $r['idphoto'];
    $fql = "SELECT src_big,src,caption FROM photo WHERE pid='".$photo_id."'";
    $photos = $facebook->api_client->fql_query($fql);
    foreach($photos as $photo)
    {
        $img_big = $photo['src_big'];
        $img = $photo['src'];
        $caption = $photo['caption'];

        echo "<div id='div_".$photo_id."' class='div_' style='height: 150px;margin: 10px;float:
left;'>";
        ?>
        <!--[if IE]>
            <style type='text/css'>
                .div_{
                    width: 130px;
                    height: 200px;
                    margin: 10px;
                    float: left;
                }
            </style>
        <![endif]-->

        <?php
        echo"
            <div class='imgAlbum' >
                <a href=$img_big title='".$caption."'><img src=$img /></a>
            </div>

            <div id='Move'."$photo_id."' style='font-size:12px;color: #3B5998;cursor:pointer;
width: 35px;height: auto;margin-right: 2px;float:left;'
                onclick=\"showLabel(\".$photo_id.\");\"><span>Move
|</span>
            </div>
            <div id='1'."$photo_id."' style='display:none; float: left;'>
                <select id='label'."$photo_id."' name='label'
onchange=\"moveLabel(\".$photo_id.\");\" style='background-color: #ECEFF5;z-
index=200;' value='hihi'>
                <option selected> Go to</option>";

            $sql_label = "SELECT * FROM Label where uid=$user AND idLabel != $label";
            $result_showLB = mysql_query($sql_label) or die("1");
            $_rows = mysql_num_rows($result_showLB);

            for($j = 0; $j < $_rows; $j++)
            {
                $r = mysql_fetch_array($result_showLB);
                $name_label = $r['nameLabel'];

```

```

        $idLabel = $r['idLabel'];
        echo "<option value= $idLabel>$name_label</option>";
    }

    echo"                </select>
            </div>
            <div id='Delete' ".$photo_id.'" style='font-size:12px;color: #3B5998;cursor:pointer; width:
            40px;height: auto;margin-right: 2px;float:left; margin-left: 5px;'
            onclick=\"DeletePhoto(\".$photo_id.\"');\"><span>Delete</span>
            </div>
            </div>";
        }
    }

    echo "
    <div class='alert' id='alert'>
        <div class='alert_content'>
            <div class='title_alert' > </div>
            <div class='divmess' id='divmess'></div>
            <div class='button_alert'>
                <button style='width:75px;float: right;'
                onclick=\"document.getElementById('alert').style.display='none'\" >
                OKey!</button>
            </div>
        </div>
    </div>";
    echo "</div>";

?>

```

photoIntoLabel.php

```

<?php

include_once("../db/config.php");
include_once("../db/DBUtiliti.php");
connectDB($dbhost, $dbuser, $dbpass, $db);

$pid = $_GET['pid'];
$idlabel = $_GET['idlabel'];
$sqlLabel = "SELECT nameLabel FROM Label WHERE idLabel=$idlabel";
$rs = mysql_query($sqlLabel) or die ("<font color=red>Err Label</font>");
$label = mysql_fetch_array($rs);
$nameLabel = $label['nameLabel'];
$sql = "SELECT * FROM Photo WHERE idPhoto='".$pid.'" AND idLabel='".$idlabel.'"";
$result = mysql_query($sql) or die("Err");
$rows = mysql_num_rows($result);

if($rows == 0)

```



```

    {
        $sql1 = "INSERT INTO Photo VALUES('$pid','$idlabel')";
        $result = mysql_query($sql1) or die("Err1");
        echo "Added to <b>$nameLabel</b>";
    }
    else{
        echo "<font color= 'red' >Don't add</font>";
    }
    closeDB();
?>

```

LightBoxLabel.php

```

<link rel="stylesheet" type="text/css" media="screen" href="css/Main.css?v=1.0" />

<?php
    $label = $_GET['label'];
    //echo "hi".$label;
?>
<fb:iframe src="http://123.30.50.210/~hienhoang/photoByLabel.php?label='<?echo $label ?>'"
    frameborder="0" width=750 height=700 scrolling="auto"/>

```

LabelToLabel.php

```

<?php

include_once("../db/config.php");
include_once("../db/DBUtiliti.php");

connectDB($dbhost, $dbuser, $dbpass, $db);

$pid = $_GET['pid'];
$idlabelTo = $_GET['idlabelTo'];
$idlabelFrom = $_GET['idlabelFrom'];

// $sqlLabel = "SELECT nameLabel FROM Label ";
$sqlLabel = "SELECT nameLabel FROM Label WHERE idLabel='".$idlabelTo."'";
$rs = mysql_query($sqlLabel) or die ("err Label");
$label = mysql_fetch_array($rs);
$nameLabelTo = $label['nameLabel'];

$sql = "SELECT * FROM Photo WHERE idPhoto='".$pid.'" AND idLabel='".$idlabelTo."'";
$result = mysql_query($sql) or die("Err");
$rows = mysql_num_rows($result);

if($rows == 0)
{
    $sql1 = "UPDATE Photo SET idLabel='".$idlabelTo.'" WHERE idphoto='".$pid."'";
    $result = mysql_query($sql1) or die("Err1");
}

echo " Moved to $nameLabelTo ";

```

```
closeDB();
```

```
?>
```

DelPhoto.php

```
<?php
if(isset($_GET['pid']))
{
    include_once("../db/config.php");
    include_once("../db/DBUtiliti.php");
    connectDB($dbhost, $dbuser, $dbpass, $db);

    $spid = $_GET['pid'];
    $sql = "DELETE FROM Photo WHERE idphoto='".$spid.'";
    $result = mysql_query($sql) or die('not delete');
    echo "Deleted successfully";
    closeDB();
}
?>
```

InviteYourFriend.php

```
<?php
include_once("../includes/config.php");

$sql = "SELECT uid FROM user
        WHERE uid IN (SELECT uid2 FROM friend WHERE uid1=$user) AND is_app_user";
// $app_users = $facebook->api_client->friends_getAppUsers();
$app_users = $facebook->api_client->fql_query($sql);
//loại danh sách người đang dùng ứng dụng
//echo $app_users;
foreach($app_users as $app_user)
{
    $app_user_id = $app_user['uid'];
    //echo $app_user_id." -- ";
}

$inviteText =
    <<<FBML
        You've been invited to View Photos!
        <fb:name uid="$user" firstnameonly="true" /> want to you joint View Photos
        <fb:req-choice
            url="http://apps.facebook.com/hienhoang/index.php?api_key=$appapikey&next=$user"
            label="Join View Photos">
    FBML;

?>
<fb:request-form action="http://apps.facebook.com/hienhoang/index.php" method="POST" invite="true"
                    type="View Photos" content="<?htmlentities($inviteText)?>">
<fb:multi-friend-selector showborder="false" actiontext="Invite your friends" exclude_ids="<?echo
                    $app_users?>" />
</fb:request-form>
```

Main.css

<pre> a{ padding: 3px; } input{ border:1px solid #BDC7D8; font-family:"lucida grande",tahoma,verdana,arial,sans-serif; font-size:11px; padding:3px; border-color:#3B5979; width: 150px; } button { padding:1px 3px 1px 4px; width: auto; cursor:pointer; background:#3b5998; color:#fff; border:1px solid #0E1f5b; border-left-color:#D9DFEA; border-top-color:#D9DFEA; } label{ padding-bottom: 10px; } #Main{ width: auto; height: auto; } #Head{ width: auto; height: 60px; } #Friend{ width: auto; height: auto; float: left; margin-left: 10px; } #path{ width: 200px; height: auto; float: left; border: 1px solid; } #addlabel{ </pre>	<pre> width: auto; height: auto; float: left; margin-left: 10px; display: none; } #Invite{ float: left; width: auto; height: auto; } #Albums{ width: 100% ; height: 25px; display: none; margin-left: 83%; } #menuShowLabel{ width: 100px; height: 25px; float: right; border: 1px solid; } #Albums img{ cursor:pointer; } #Back{ width: 100% ; height: 25px; display: none; margin-left: 90%; } #Back img{ cursor:pointer; } #Left{ width: 99%; height: 600px; margin-top: 3px; } #headShowLabel{ width: 100%; height: 20px; } #showLabel{ border: 1px solid #CCCCCC; width:130px; </pre>
--	--

<pre> height: 100%; margin-right: 2px; margin-left: 10px; float: left; font-size: 16px; } #showAlbums{ border: 1px solid #CCCCCC; background-color: #ECEFF5; width: 730px; height: 98%; padding: 5px; overflow: auto; } #TitleAlbums{ margin-top:20px; } .div_album{ width:auto; height: 150px; margin: 10px; float: left; } .imgAlbum{ border: 1px solid #CCCCCC; width:auto; height: auto; vertical-align: middle; margin-bottom: 3px; padding-top:4px; padding-right:8px; padding-bottom:3px; padding-left:2px; } .imgAlbum:hover{ border-color: #151B8D; } .imgAlbum img{ width: 100%; } .nameAlbum{ width:130px; height: auto; } #Right{ width: 100%; height: auto; float: left; </pre>	<pre> padding: 5px; } #HeadRight{ } #Right_content{ background-color: #ECEFF5; width: 100%; height: 90%; } #IconWait{ background- image:url(http://123.30.50.210/~hienhoa ng/Images/loader1.gif); width: 100px; height: 100px; display: none; margin-top:100px; margin-bottom: 100px; margin-left: 280px; float: left; } /* viewFriend */ .photo{ width: auto; height: 150px; margin-right: 10px; margin-left: 5px; float: left; } #gallery a{ text-decoration: none; } #gallery img{ border: none; } /* alert */ .alert{ display:none; position:absolute; top:1px; left:1px; width:auto; height: auto; padding: 10px; z-index: 2000; </pre>
--	---

```

-moz-border-radius: 5px; /* for Mozilla
browsers */
-webkit-border-radius: 5px; /* for webkit
browsers (Chrome, Safari) */
border-radius: 5px; /* for Opera and
others */
border: 1px solid gray;
background: #858585;
}
.alert_content{
width:400px;
height: auto;
background-color:#F2F2F2;
border-style:solid;
border-width:1px;
}
.title_alert{
width: 100%;

```

```

height: 30px;
background-color:#6D84B4;
}
.divmess{
border-bottom-width: 1px;
border-bottom-style: solid;
border-bottom-color: #cccccc;
background-color:#FFF;
width: 100%;
height: 40px;
text-align: center;
padding-top: 5px;
}
.button_alert{
width: 95%;
height: 20px;
padding: 5px;

```

TabStyle.css

```

#tab{
padding-left: 10px;
--float: left;
height: 20px;
margin-bottom: 10px;
border-bottom: 1px solid
#333333;
}
li{
border-color:#898989 #898989
#898989;
border-style: solid;
border-width: 1px;
float: left;
}
li a{
color: #333333;
display: inline-block;
font-weight: bold;
padding: 2px 8px 3px 9px;
}
#first {
background-color : #6D84B4;
}
#first:hover{
background-color : #6D84B4;
}
#first a{

```

```

color: #ffffff;
}
/* -- menu Label */
#menuLabel {
width: 100px;
height: 18px;
}
#menuLabel a{
}
#menuLabel ul {
background: none repeat scroll 0
0 #ECEFF5;
display : none;
width: 150px;
height: auto;
z-index: 200;
position: absolute;
opacity: 0.85;
}
#menuLabel:hover ul {
display: block;
}
#menuLabel ul li{
display: block;
float: none;
width: 150px;
height: 30px;
}
#menuLabel li a{

```

<pre> display: block; font-size: 12px; font-style: normal; padding: 10px 5px 5px 15px; text-align: left; width: 100px; float: left; } /* #menuLabel ul a:hover{ background : none repeat scroll 0 0 #949494; color: #fff; opacity: 1; } */ #menuLabel ul li:hover{ background : none repeat scroll 0 0 #fff; </pre>	<pre> opacity: 1; } #menuLabel ul li:hover span{ display: block; cursor: pointer; } #delete{ height: 20px; padding-top: 8px; float: left; display: none; width: 25px; } #label{ border: 1px solid; width: 100px; height: 25px; float: left; } </pre>
---	--

IndexPage.js

```

function delLabel()
{
    var form = document.getElementById('form_del');
    form.submit();
}
function show(aid){

    document.getElementById('Left').setStyle('display','none');
    document.getElementById('Albums').setStyle('display','block');
    document.getElementById('IconWait').setStyle('display','block');
    document.getElementById('Right_content').setStyle('display','none');
    var fid = document.getElementById('fid_hidden').getValue();
    var ajax = new Ajax();
    ajax.responseType = Ajax.FBML;
    ajax.ondone = function(data)
    {

        document.getElementById('Right_content').setStyle('display','block');
        document.getElementById('Right_content').setInnerFBML(data);
        document.getElementById('IconWait').setStyle('display','none');
    }
    ajax.post('http://123.30.50.210/~hienhoang/lightbox.php?fid='+ fid +'&aid='+ aid);
}

function showBigPhoto(currentP)
{

```

```

document.getElementById('IconWait').setStyle('display','block');
document.getElementById('Right_content').setStyle('display','none');
var aid = document.getElementById('aid_hidden').getValue();
var fid = document.getElementById('fid_hidden').getValue();
//var myDialog = new Dialog(Dialog.DIALOG_POP);
//myDialog.showMessage('Title Here','album: "+aid+" photo: "+currentP ,
button_confirm='Okay!');
var ajax = new Ajax();
ajax.responseType = Ajax.FBML;
ajax.ondone = function(data)
{

document.getElementById('Right_content').setStyle('display','block');
document.getElementById('Right_content').setInnerFBML(data);
document.getElementById('IconWait').setStyle('display','none');

}

ajax.post('http://123.30.50.210/~hienhoang/lightbox.php?fid='+fid+'&aid='+aid+'&currentP='+currentP);
}
function shAlbum()
{
document.getElementById('Left').setStyle('display','block');
document.getElementById('IconWait').setStyle('display','none');
document.getElementById('Right_content').setStyle('display','none');
document.getElementById('Albums').setStyle('display','none');
}
function selected_tab(tab)
{
var tab_first = document.getElementById('first').getValue();
var tab_second = document.getElementById('second').getValue();
if(tab == tab_first )
{
document.getElementById('first').setStyle('backgroundColor','#6D84B4');
document.getElementById('a1').setStyle('color','#ffffff');
document.getElementById('second').setStyle('backgroundColor','#FFFFFF');
document.getElementById('a2').setStyle('color','#333333');
document.getElementById('addlabel').setStyle('display','none');
document.getElementById('Friend').setStyle('display','block');
}
if(tab == tab_second)
{
document.getElementById('second').setStyle('backgroundColor','#6D84B4');
document.getElementById('a2').setStyle('color','#ffffff');
document.getElementById('first').setStyle('backgroundColor','#FFFFFF');
document.getElementById('a1').setStyle('color','#333333');
document.getElementById('addlabel').setStyle('display','block');
document.getElementById('Friend').setStyle('display','none');
}
}
}

```

```

function photoByLabel(label)
{
    //var myDialog = new Dialog(Dialog.DIALOG_POP);
    //myDialog.showMessage('Title Here',label, button_confirm='Okay!');

    document.getElementById('Left').setStyle('display','none');
    document.getElementById('IconWait').setStyle('display','block');
    document.getElementById('Right_content').setStyle('display','none');

    var ajax = new Ajax();
    ajax.responseType = Ajax.FBML;
    ajax.ondone = function(data)
    {

        document.getElementById('Right_content').setStyle('display','block');
        document.getElementById('Right_content').setInnerFBML(data);
        document.getElementById('IconWait').setStyle('display','none');
    }

    ajax.post('http://123.30.50.210/~hienhoang/lightboxLabel.php?label='+label);

}

```

Del_Move.js

```

var mz=false;
if(navigator.appName=="Netscape"){mz=true}
function getCursorPos(e)
{
    var cursor = {x:0, y:0};

    if(mz == true)//mozilla
    {
        //cursor.x = e.clientX;
        cursor.y = 300;//e.clientY;
    }
    else //IE
    {
        var de = document.documentElement;
        var db = document.body;
        cursor.x = event.clientX + (de.scrollLeft ? de.scrollLeft : db.scrollLeft);
        cursor.y = event.clientY + (de.scrollTop ? de.scrollTop : db.scrollTop);
    }

    return cursor;
}
function ajax(){
    var xmlhttp;
    try {
        xmlhttp=new XMLHttpRequest(); // Firefox, Opera 8.0+, Safari
    }
}

```



```

        catch (e)
        {
            // Internet Explorer
            try
            {
                xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
            }
            catch (e)
            {
                try
                {
                    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
                }
                catch (e)
                {
                    alert("Your browser does not support AJAX!");
                    return false;
                }
            }
        }
        return xmlhttp;
    }
}
function showChooseLabel(pid)
{
    document.getElementById(pid).style.display = 'block';
    document.getElementById('add'+pid).style.display = 'none';
    //document.getElementById('mydiv'+pid).style.display = 'none';
}
function addLabel(pid)
{
    var idlabel = document.getElementById('label_choose'+pid).value;
    //document.getElementById(pid).style.display = 'none';
    //document.getElementById('add'+pid).style.display = 'block';
    var xmlhttp = ajax();
    xmlhttp.onreadystatechange=function()
    {
        if(xmlhttp.readyState == 4 && xmlhttp.status == 200)
        {
            document.getElementById('mydiv').innerHTML = xmlhttp.responseText;
        }
    }
    xmlhttp.open("GET","photoIntoLabel.php?pid="+pid+"&idlabel="+idlabel,true);
    xmlhttp.send();
    document.getElementById(pid).style.display = 'none';
    document.getElementById('add'+pid).style.display = 'block';
    //document.getElementById('mydiv'+pid).style.display = 'block';
    /*
    var cursorPos = {x:0, y:0};
    cursorPos = getCursorPos;
    if(cursorPos.y > 300 )

```

```
{
    cursorPos.y = (cursorPos.y - 300);
}
else
    cursorPos.y = 300;
*/
DisplayAlert('alert',100,300);
document.getElementById('alert').focus();
}

function showLabel(pid){

    document.getElementById('l'+pid).style.display = 'block';
    document.getElementById('Move'+pid).style.display = 'none';
//    document.getElementById('divmess').style.display = 'none';
}

function moveLabel(pid)
{
    var idlabel_To = document.getElementById('label'+pid).value;
    var idlabel_From = document.getElementById('hidden_Idlabel').value;
    var nameLabel = document.getElementById('hidden_Namelabel').value;

    var xmlhttp = ajax();

    //alert(xmlhttp.readyState);
    xmlhttp.onreadystatechange=function()
    {
        if(xmlhttp.readyState == 4 && xmlhttp.status == 200)
        {

            document.getElementById('divmess').innerHTML = xmlhttp.responseText;
        }
        else
        {
            document.getElementById('divmess').innerHTML = 'please wait...';
        }
    }

    xmlhttp.open("GET","LabelToLabel.php?pid="+pid+"&idlabelTo="+idlabel_To+"&idlabelFrom="+idlabel_From,true);
    xmlhttp.send();
    document.getElementById('div_'+pid).style.display = 'none';
    document.getElementById('l'+pid).style.display = 'none';
    //document.getElementById('Move'+pid).style.display = 'block';
    //document.getElementById('divmess'+pid).style.display = 'block';
    /*
    var cursorPos = {x:0, y:0};
    cursorPos = getCursorPos(this);
    if(cursorPos.y > 300 )
```

```
{
    cursorPos.y = (cursorPos.y - 300);
}
else
    cursorPos.y = 300;
*/
DisplayAlert('alert',100,300);
document.getElementById('alert').focus();
}
function DeletePhoto(pid){
    var xmlhttp = ajax();
    xmlhttp.onreadystatechange=function()
    {
        if(xmlhttp.readyState == 4 && xmlhttp.status == 200)
        {

            document.getElementById('divmess').innerHTML = xmlhttp.responseText;
        }
        else
        {
            document.getElementById('divmess').innerHTML = 'please wait...';
        }
    }
    xmlhttp.open("GET","DelPhoto.php?pid="+pid,true);
    xmlhttp.send();
    document.getElementById('div_'+pid).style.display = 'none';
    document.getElementById('l'+pid).style.display = 'none';
    DisplayAlert('alert',130,170);
    document.getElementById('alert').focus();
}
function DisplayAlert(id,left,top) {
    document.getElementById(id).style.left=left+'px';
    document.getElementById(id).style.top=top+'px';
    document.getElementById(id).style.display='block';
}
```

CHÚ THÍCH

1. http://www.siliconbeat.com/entries/2005/07/18/web_20_peak_myspace_sells_for_580_million.html
2. <http://vi.wikipedia.org/wiki/Facebook>
3. http://www.pcworld.com/article/149732/facebook_continues_to_dominate_social_networking_sites.html
4. <http://techcrunch.com/2009/12/21/world-map-social-networks/>

TÀI LIỆU THAM KHẢO

Sách:

- [1] Hasin Hayder và Dr.Mark Alexander, Learning Facebook Application Development Nxb Packt, 5-2008.
- [2] Jay Goldman, Facebook Cookbook, Nxb O'Reilly Media, 10-2008
- [3] Jesse Stay, FBML Essentials, Nxb O'Reilly Media, 7 - 2008 .
- [4] Wayne Graham, Facebook API Developers Guide, Nxb Apress , 2008.

Web:

- [5] Facebook Developer Wiki . Nguồn: <http://wiki.developers.facebook.com/>
- [6] Facebook Platform Developer Forum, Nguồn: <http://forum.developers.facebook.com/>