Helsinki Metropolia University of Applied Sciences
Degree Programme in Information Technology

Markus Sammallahti

Facebook Application Development

Bachelor's Thesis. 24 March 2010
Supervisor: Markku Karhu, Principal Lecturer

Helsinki Metropolia University of Applied Sciences Abstract

Author	Markus Sammallahti
Title	Facebook Application Development
Number of Pages	46
Date	24 March 2010
Degree Programme	Information Technology
Degree	Bachelor of Engineering
Supervisor	Markku Karhu, Principal Lecturer

The purpose of this thesis was to gather high-level technical information about Facebook application development. The primary study targets were Facebook platform components: how they could be used in platform applications and what kind of data is available for the applications to use. During the project two applications were built to obtain first-hand knowledge about the development process and available tools.

At the beginning of the project the platform was studied by reading available online documentation. Information about all platform components was gathered, and the purpose and capabilities of each component was mapped out in detail. Later in the project this information was used to create experimental applications. During the development process one aim was to discover possible problems with the platform components, and special circumstances regarding the platform were investigated.

The Facebook platform was found to be extremely powerful in developing a social application. The tools that the platform provides proved to be very useful and easy to use. It was possible to build fully functional social applications quickly by combining all platform components. User data provided by the platform was also found to be very useful. It was noted that the user data and large numbers of users are the most important asset of the platform.

As all development platforms, the Facebook platform was also found to have advantages and disadvantages. The platform offers many powerful components that save a large amount of time in the development process, but at the same time it lays down very stringent development restrictions. While developing a Facebook application a choice between some platform components and development tools had to be made. Other problems with the platform components proved to be trivial and easy to solve.

All project objectives were achieved, and the project can be considered successful. The comprehensive information about Facebook application development that was gathered can be used in future projects.

Keywords	Facebook, application development, social media, fbml, fbjs, fql

Metropolia Ammattikorkeakoulu Insinöörityön tiivistelmä

Tekijä Otsikko Sivumäärä Aika	Markus Sammallahti Facebook-sovelluskehitys 46 sivua 24.3.2010
Koulutusohjelma	Tietotekniikan koulutusohjelma
Tutkinto	insinööri (AMK)
Ohjaava opettaja	yliopettaja Markku Karhu

Tämän insinöörityön tarkoituksena oli kerätä korkean tason teknistä tietoa Facebook-sovellusten kehittämisestä. Tärkeimmät tutkimuskohteet olivat Facebook-ohjelmistoalustan osat, niiden käyttö sovelluksissa ja sovelluksien käytössä oleva data. Työn aikana luotiin kaksi sovellusta, joiden avulla saatiin ensikäden tietoa sovelluskehityksen vaiheista ja käytettävissä olevista työkaluista.

Työn alussa ohjelmistoalustaan tutustuttiin lukemalla Internetissä tarjolla olevaa dokumentaatiota. Tietoa alustan eri komponenttien käyttötarkoituksista ja käyttömahdollisuuksista tutkittiin erittäin huolellisesti. Kerätyn tiedon perusteella luotiin myöhemmin kaksi testisovellusta joiden avulla yritettiin etsiä mahdollisia ongelmatilanteita ja kartoittaa alustan erikoispiirteitä sovelluskehityksen näkökulmasta.

Facebook-ohjelmistoalustan havaittiin olevan erittäin tehokas apu sosiaalisten verkkosovellusten kehittämisessä. Alustan tarjoamat työkalut todettiin käyttökelpoisiksi ja helppokäyttöisiksi. Täysiverisen sosiaalisen verkkosovelluksen nopea luominen oli mahdollista yhdistelemällä kaikkia alustan eri komponentteja. Alustan tarjoamat käyttäjätiedot ja suuri käyttäjämäärä todettiin tärkeimmiksi käytettävissä oleviksi resursseiksi.

Facebook-ohjelmistoalustalla todettiin olevan hyviä ja huonoja puolia kuten yleensä kaikilla alustoilla. Kyseinen alusta tarjoaa monia aikaa säästäviä sovelluskomponentteja, mutta samalla rajoittaa käytettävissä olevia ohjelmointityökaluja. Facebook-sovelluksia luotaessa jouduttiin tekemään ratkaisuja alustan ominaisuuksien ja kehitystyökalujen välillä. Muiden alustan käytön tuomien ongelmien havaittiin olevan helposti ratkaistavissa.

Projektin aikana saavutettiin kaikki tälle insinöörityölle asetetut tavoitteet ja tulosta voidaan pitää onnistuneena. Facebook-sovellusten kehittämisestä kerättiin kattavasti tietoa, jota voidaan käyttää hyväksi tulevissa projekteissa.

Hakusanat	Facebook, ohjelmistokehitys, sosiaalinen media, fbml, fbjs, fql

Abbreviations and Terms

AJAX	Asynchronous JavaScript And XML, a technique used in client side web applications.
API	Application Programming Interface, an interface implemented by a software program to interact with other software.
CSS	Cascading Style Sheets, a style sheet language used to describe presentation semantics of a document.
DOM	Document Object Model, a convention for representing and interacting with objects in markup language documents.
FBML	Facebook Markup Language, a markup language used in Facebook applications.
FBJS	Facebook JavaScript, a scripting language used in Facebook applications.
FQL	Facebook Query Language, a database computer language used in Facebook applications.
HTML	Hypertext Markup Language, a markup language used in web documents.
HTTP	Hypertext Transfer Protocol, an application layer protocol for distributed hypermedia information systems.
IM	Instant Messaging, real-time text-based communication.
JavaScript	A scripting language used in client-side web development.
JSON	JavaScript Object Notation, text-based computer data interchange format.
PHP	PHP Hypertext Preprocessor, a scripting language used in server-side web development.

REST	Representational State Transfer, a style of software architecture for distributed hypermedia systems.
SQL	Structured Query Language, a database computer language used in relational database management systems.
UI	User Interface, a place where interaction between humans and machines occur.
URL	Uniform Resource Locator, an identifier that specifies where a resource is available.
Web 2.0	A term used for web applications with user created content.
XFBML	A subset of FBML elements available in iframe applications.
XML	Extensible Markup Language, a set of rules for encoding documents electronically.

Contents

1 Introduction

Social networks have become the most dominant part of the Internet, whose users spend more time on social networking websites than on any other type of websites. This makes social networks to be very intriguing from an advertiser's point of view. A large amount of personal data is available and more targeted ads can be displayed to users. Previously social networks had simple profiles and only advertising space could be bought, but now special applications can be built and integrated into a social network. These applications can be created for virtually any desired purpose.

Facebook applications have become hugely popular and valuable. The platform that the applications are running on is extremely interesting and is the main reason behind the popularity. The platform provides a wide range of tools and vast numbers of user data to be used in the applications.

The purpose of this project was to study this entirely new platform and to get familiar with it through hands-on experience. Before the project was launched only a little literature was available, and most of it was extremely technical. The goal was to study what can or cannot be done on higher level, and to find as many problematic situations as possible. During the process two different kinds of applications were built to get first hand information about the platform, the application development process and the available tools.

This thesis discusses Facebook application development and Facebook platform features. The most important parts of the platform and the development process are covered with information about general problems and platform specific circumstances. This thesis concentrates only on Facebook platform applications. Desktop applications and Facebook connect features are not discussed, because they can be considered to be different environments. The attractiveness of a social network from an advertiser's point of view is lost in these segments.

2 Facebook

Facebook is a social networking website owned and operated by Facebook Inc [1]. Since its launch in 2004, Facebook has practically been taking over the Internet [2]. Facebook is the most used social network and the second most popular website on the Internet right after Google.com [3]. Facebook currently has over 400 million users which is more than 23 percent of all Internet users [2; 4]. Not only is the number of users large, but the amount of time spent on the website is also high. An average user spends over 30 minutes on the service during one session [5].

Facebook's main competitor is MySpace, which has the same level of features as Facebook. Facebook is also competing with other services and social networks, but their feature set is less comprehensive. Facebook has almost every feature common to other social networking and web 2.0 services. Whereas some popular services like Flickr and Twitter are built around one simple context like photo sharing or status updates, Facebook combines all these different features under one service. For example Facebook allows users to share photos and links, write blogs and to communicate in many different ways. [6]

A wide range of features and usability has made Facebook hugely popular. And after allowing third parties to access user data and to build applications that are integrated into the service, the popularity has been skyrocketing. Enormous and worldwide user space, advertisement possibilities and extensibility have made Facebook extremely valuable. Facebook has been estimated to be worth $15 billion, which makes it the most valuable social networking website. [2; 7]

3 Platform

The Facebook platform was introduced on May 27, 2007. The platform provides a wide range of tools for software developers to build social network applications. The platform can be used to integrate external, already existing resources into Facebook, and to create applications using Facebook features and data. User data can be accessed through the Facebook platform, but users always have to allow every individual application to access their data. [8; 9]

There are many reasons for the release of the Facebook platform. By providing powerful tools for independent software developers, the service will obtain more content and new ways to interact. These new and different kinds of features attract more people to join the service, and as the user space increases so does the amount of data about the users. Social networks are mainly valued for the number of active users, and the amount and accuracy of the available user data. This is because the vast amounts of data attract advertisers, collaborating companies and developers. [7]

The platform provides basic and easy-to-use components that are essential to social network applications. This makes application development much faster and more reliable, since these components have great numbers of users. Software developers can create extremely powerful social network applications by merging these platform components with an ordinary online application or create an entirely new application by only using platform components.

Because raw user data is accessible for applications, it also makes the platform very interesting to other operators. The more one particular platform has users, the more intriguing it is for advertisers. The platform allows companies to gain information about user demographics while interacting with current and potential customers. In this way the platform can be seen as a PR tool and a new way of marketing. From a software developer's point of view good development tools make a platform even more attractive. But by using the Facebook platform, powerful social network applications, automatically available to a great numbers of users, can be created. And since the

market for applications already exists, only a small amount of time and effort needs to be put into marketing the application. [7; 10]

3.1 Components

The Facebook platform can be divided into four main components, Facebook API, Facebook Markup Language (FBML), FBJS and Facebook Query Language (FQL). Each component has a different role on the platform, even though some of the features do overlap.

3.1.1 Facebook API

Facebook API uses a REST-like interface, which means that method calls are made by sending HTTP requests over the Internet. Almost every modern programming language can be used to send HTTP requests, and this makes the API available to the largest possible number of platforms. The official API client library is only provided in PHP and JavaScript languages, but there are multiple unofficial client libraries created by the application developer community. [11]

The Facebook API is the most important component of the platform. It is used to authenticate the application and the users using the application, which is required for the other platform components to work. The API can be used to access users' Facebook related data and to modify it. All platform applications have access to users' data such as friends, networks and photos. Personal data like this is the base of a fully featured social network application. Because an application can access personal Facebook data, every application and user making use of the application is authenticated to ensure information security. [9]

The API is also the only way to trigger other platform features that Facebook provides to all applications. The main features are alerts, notifications and requests. Some of these features need application specific components like templates or queries to be registered before they can be used. These components are registered also with the API. The registered items are cached on the Facebook server, so that they can be used afterwards with the features triggered by the API. Some of the features provided by

Facebook have application specific daily request limits. These limits are set to prevent spamming and making sure that every application has the resources it needs.

The API has a large number of methods that can be used in an application, and the methods are divided into groups based on their type of use. The groups are administration, authentication, data retrieval, publishing, photos and events. [11]

3.1.1.1 Administrative methods

Administrative methods are used to control application specific settings. Getting the application's allocation limits to platform integration points or managing banned users can be done by using these methods. Basically everything related to an application's permissions is managed through administrative methods. For example the *admin.banUsers* method can be used to ban users from using an application, the method taking an array of user IDs to be banned as an argument. [11; 12]

3.1.1.2 Authentication methods

Authentication methods are rarely used in an application development process. They are mainly used internally in the client library to manage session information and to validate parameters received from Facebook. Authentication methods can still be used in an application, if session related special functionality is required. For example the *auth.revokeAuthorization* method can be used to deauthorize an application on behalf of the current user. This means that the user cannot use the application before it is authorized again, nor can the application access the user's data. [11]

3.1.1.3 Data retrieval methods

Data retrieval methods are used to access Facebook data. These methods are crucial for a social network application, and may be the most important part of the API. Everything related to getting and setting user related Facebook data is done using data retrieval methods. These methods can be used to get information about the user, his friends and groups. [11]

With the *friends.getMutualFriends* method a list of users can be fetched that are mutual friends of two different users. This method takes a user id as an argument, and matches friends of that user with the users of the user currently using the application. The method returns an array of user IDs that are friends with both users. [13]

3.1.1.4 Publishing methods

Facebook allows content to be created to users' profile pages through an application. Profile entries like notes, links and news feed stories can be created and modified using publishing methods. These methods can also be used to add a comment to, or to remove a comment previously created by a user from any item published on Facebook. In a nutshell, publishing methods control user output outside of an application. [11]

For example, the *users.setStatus* method can be used to update the status of a user currently using an application. The method takes a text string as an argument and sets it as the user's current status message. [14]

3.1.1.5 Photos API methods

While photos can be considered as profile content, API methods related to photo management have their own special group. With these methods information about a user's albums and photos can be accessed from an application. Also albums can be created and photos uploaded using the photo API methods, but no photo-related content can be removed or modified. [11]

With the *photos.createAlbum* method a new album can be created into the user's photo library. The method takes a text string as an argument, which is set as the name of the created album, and returns an array containing a newly created album's meta data. [15]

3.1.1.6 Events API methods

One of Facebook's main features is events. With this feature users can organize and share information about upcoming events, and event methods enable interaction with Facebook events from an application. Event methods include actions from creating an event, to inviting people to an event. [11]

With the *events.getMembers* method a list of people that are associated with an event can be fetched. The method takes an event id as an argument and returns an array of user IDs related to the event grouped by the current RSVP status. [16]

3.1.2 FBML

The Facebook Markup Language is the markup language used in Facebook application development. FBML is an evolved subset of HTML, and like HTML it is used to define a document structure and how it is displayed. Some standard HTML elements have been removed from FBML, and many Facebook specific elements have been added. A few additional element attributes have also been added to enable Facebook application specific actions. These extra attributes are deployed to pursue higher abstraction. [17]

While HTML is strictly used to describe what the document looks like, FBML has also tags to control what will be included in the final output. This is possible, because FBML is processed on a Facebook server and the developer can use tags to control the process. During the process FBML is actually turned into a combination of HTML elements, CSS style definitions and JavaScript code. [18]

FBML has plenty of Facebook specific tags that allow software developers to integrate the application with Facebook data and easily adopt the look and feel of Facebook within an application. Some FBML tags even create fully functional widget-like components that do not need any contribution from a developer. These are the same exact components that are used in the main service, and a social application could be easily built by using only FBML components. These additional FBML tags have been divided into five groups based on their purpose. The groups are social, sanitization, design, component and control. [18]

3.1.2.1 Social data tags

Social data tags provide the primary information for a social network application. Social data tags can be used to get and format data about all social networks' components like users, networks and photos. The output of social data tags depend on a user viewing a document. The user might not have permissions, or has limited permissions to view the

data. Depending on the permissions, full, partial or no information will be displayed. [18]

For example a *fb:profile-pic* tag can be used to display the current profile picture of a given user. Social data tags provide a level of abstraction to the development process, because decision making is handed over to the platform in parts that are insignificant to application operations. Social data tags are very useful, and the abstraction that they provide makes the development process more straightforward and faster.

3.1.2.2 Sanitization tags

Sanitization tags are used to add embedded media to the application. Normally HTML tags like *object* and *embed* would be used to add media, and sanitization tags are created to replace these universal tags that can be a security threat. With *object* and *embed* tags almost anything can be embedded into a HTML document. These dangerous tags are replaced with a tag for every type of media that is allowed to embed. For example with a *fb:swf* tag a Shockwave Flash object can be embedded into an application. [18]

3.1.2.3 Design tags

Design tags do not have any data related functions. They are created to easily adopt Facebook appearance and style in an application. Design tags are simply transformed into HTML elements with style definitions similar to those Facebook uses. This way an application blends in with other parts of the service, and provides a familiar interface for users. By using design tags the developer can save time in the design process and make an application more usable. [18]

The best example of design tags are *fb:tabs* and *fb:tab-item*. Since applications often have multiple sections, a *fb:tabs* tag can be used to create an interface to navigate between different section. In order to render the tabs, a *fb:tab-item* tag is needed for each desired section along with the *fb:tabs* tag.

3.1.2.4 Component tags

Component tags are the most interesting part of FBML. The component tags provide fully functional widget-like application parts that are maintained by Facebook. Things like a discussion board can be easily embedded into an application, with virtually no trouble at all. The component tags can be used to quickly turn an ordinary web application into a highly social application. [18]

For example with a *fb:comments* tag, a commenting functionality can be easily added to an application and enable users to communicate. Component tags are really valuable to a developer, because they look good and work really well, not to mention the time that is saved in the development process.

3.1.2.5 Control tags

With control tags some level of programmability can be easily added into an application. Instead of providing a programming language like general-purpose control structures, the control tags are used to control application output based on Facebook data and users' permission settings. This feature is very useful, since the user data and settings are changing constantly. [18]

By using control tags the developer hands over the decision making to Facebook, and makes sure that everything is displayed correctly based on current user settings. For example, with a *fb:if-is-group-member* tag some part of an application can be displayed only to members of a particular group, while non-members receive some other content.

3.1.3 FBJS

JavaScript is a widely used client side programming language in web development, and FBJS is Facebook's way of implementing JavaScript support on the platform. Since many developers are familiar using JavaScript, working with FBJS does not raise a great issue. The most challenging problem with JavaScript is that it is not cross-browser compatible, and Facebook has redressed this issue in FBJS by introducing new methods and events. [19]

Since Facebook has fixed the browser compatibility issue by altering the way DOM properties are accessed, another far more extensive problem has emerged. Changing the DOM tree manipulation practice means that none of the cross-browser JavaScript software libraries are compatible with FBJS. This is an extremely unpleasant issue, because not only has every one of these software libraries already addressed the browser incompatibility issue, but they also provide other really powerful features.

3.1.3.1 Sandboxing

What FBJS actually does is process the FBJS code turning it into JavaScript code that works with the browser that will be using it. During the processing identifiers like function names and variable names are prefixed with an application identifier. For example the variable *user_name* turns into *app123_user_name*. The same operation is done to function names and function calls. [19]

This way Facebook sandboxes the JavaScript that applications produce. Every application has its own namespace, and applications cannot access each other's code. This feature increases security, and allows different developers to use the same names in variables and functions. But when some of the standard JavaScript functions are disabled in FBJS, they are also prefixed. For example no *alert* function is found in FBJS, so an application prefix is added to it. This makes code debugging somewhat harder, because the prefix is added to a function call but the function is never declared.

3.1.3.2 DOM manipulation

The way DOM object properties are accessed has been modified in FBJS. To reckon with object oriented programming style, setters and getters have been created to access common DOM object properties like title and name. When using setter and getter methods the code becomes a little cleaner and easier to read. And because these methods are chainable, multiple method calls can be made to one object in one simultaneous chain of commands. Setters and getters are the most significant modifications that have been made to FBJS. [19]

3.1.3.3 Content setters

Since DOM object properties have setters, special element content setters have also been created. DOM objects have three different content setters to handle all different kinds of content formats supported by the platform. To support platform specific FBML elements, *setInnerFBML* can be used to set pre-processed content to an element. *setInnerXHTML* is used when used content has HTML elements that do not require pre-processing and can be created dynamically. If content is to be set to a plain text a *setTextValue* can be used. [19]

3.1.3.4 CSS manipulation

DOM object style properties can be accessed with *setStyle* and *getStyle* methods. In addition to these CSS style attribute-related methods, a few CSS class manipulation methods have been created. It is really common to create CSS style classes and link DOM objects to those classes. *addClassName* and *removeClassName* functions can be used to dynamically add a style class to or remove it from an element. Or a *toggleClassName* can be used to add a style class to or remove it from an object depending on current state. To check if an element has been added to a class, a *hasClassName* function can be used. These CSS style methods are an effortless way to increase application interactivity by quickly altering elements styles. [19]

3.1.3.5 Events

Events are basic operations to respond to user actions in JavaScript. FBJS supports events, and has also added a few methods to make event handling easier. Basic methods for listing and removing all objects' event listeners have been added, as have a method to prevent default element behavior and another method to break the event triggering chain. These extra event functions make event handling somewhat easier, especially removing of all event listeners at once. [19]

3.1.3.6 AJAX

AJAX is a technique that is widely used in web applications to increase usability and add interactivity. FBJS provides an AJAX object that can be used to easily add AJAX

functionality to Facebook applications. The AJAX object is cross-browser compatible like other FBJS features, but not as general-purpose as AJAX implementations in other JavaScript libraries. Because Facebook uses a special markup language and has a strict security policy, the AJAX object is built to support these requirements. To implement the security policy, an AJAX request can be forced to require a user to log in before it can be executed. JSON type responses are often used in AJAX implementations, and the FBJS AJAX object also has support for parsing JSON response. In addition to JSON parsing, the response can be also parsed as FBML to support the platform's special markup language. [19]

3.1.3.7 Dialogs

Facebook relies heavily on popup dialogs in its data presentation. FBJS has a Dialog object that can be used to create Facebook style dialogs in an application. Dialogs are really easy to use and make the application look more like the main service. Another reason for providing the Dialog object is, that regular JavaScript alert and confirm dialogs are disabled in FBJS, and these interactive features needed a reasonable replacement. Facebook dialogs look attractive, and they are a convenient way to ask for a confirmation or display small amounts of data. Dialogs can be created dynamically, or they can be displayed by using pre-rendered FBML blocks. [19]

3.1.3.8 Animation library

With FBJS Facebook also ships an animation library. This animation library can be used to increase the user friendliness of an application by making things work more smoothly. The animation library offers standard UI operations like hiding and revealing and element, but it also has methods for tweening an object's style properties, which means transition from one color to another, or moving an element smoothly. The animation library provides a convenient tool to make an application more usable and polished. There is also a version of the animation library that can be used outside of Facebook. [20]

3.1.4 FQL

The Facebook Query Language provides a consistent interface to user data. FQL is a SQL-like language, and everyone familiar with SQL can easily use it. FQL does not support as complex queries as SQL, but basic operations are identical. While the Facebook API is inconsistent and slow on some parts, FQL is a more powerful way to access user data. But FQL cannot completely replace the API, since it only allows reading of user data. [21]

With FQL, only the data that is needed can be fetched rather than large collections that the API returns. Even though FQL does not support cross-table queries, fewer API requests are needed to get the same data with FQL than by using API data retrieval methods. For example, getting a list of groups for multiple users requires multiple requests when the API is used. Exactly the same data can me fetched with one request using FQL. When getting less data in a simpler format, parsing becomes easier and the application faster and more straightforward. Also, because FQL compared with the API, provides a more consistent interface to user data, the code becomes cleaner and easier to manage.

FQL queries can be executed by using the Facebook API. The execution can be done one query at a time, by running multiple queries at once, or by preloading queries. [21]

3.1.4.1 Query structure

FQL queries are made in a simple form of *SELECT [fields] FROM [table] WHERE [conditions]*. Sorting clauses ORDER BY and LIMIT are optional. Every query has to be indexable and can only be made against one table. Indexable means that the WHERE clause must have a condition against a column in a queried table that represents a unique identification. Such values are user ID, group ID and event ID, for example. [21]

Even though queries can be made only against one table, subqueries are allowed to be used in a WHERE clause. By using the IN keyword in a WHERE clause, cross table queries are possible. [21]

3.1.4.2 Available data and functions

All data that Facebook makes available can be accessed, but it is only readable with FQL. There are three tables, *application*, *developer* and *metrics* that can be used to read application settings. Table *translation* can be used to make an application multi-lingual. It provides text strings in many available languages. All other tables like *friend*, *event* and *profile* hold user related data. [22]

FQL mainly supports string functions. These are very basic functions like joining two strings, calculating the length of a string and converting a string to uppercase. The only non-string related functions are *now*, to get current timestamp, and *rand*, to get a random number. Widely used SQL functions like *sum*, *count* and *average* are not available, since these functions are applied to record groups which FQL does not support. [23]

3.1.4.3 Preloading FQL

What preloading an FQL query means, is that a developer tells Facebook beforehand what queries will be executed in an application. This is done by binding an application URL pattern with an FQL query. For example a developer can define that if a URL includes the word friends, an FQL query that gets a list of the user's friends will be executed. This same operation could be done using a regular FQL query executed via the API, or using a dedicated API method. But by using a preloaded FQL query, no API calls are needed and the application will have the same data. FQL preloading makes an application run faster since less API calls are required. [24]

Every FQL query desired to be preloaded, has to be registered as a preload rule with a unique identifying name. Every rule has a regular expression pattern that is used to match a URL and an FQL query to be executed. This way multiple preload rules can match one URL. FQL preloads are registered with the *admin.setAppProperties* API method. [24]

3.2 Competition

None of the other social networks has a development platform like Facebook. Many other services do not have any kind of development features or have only ones with limited operations allowed. Twitter is one platform that allows third party applications to be integrated into the main service, but it is not comparable to the level of the Facebook platform. The platform that Facebook offers is very unique, because it has many social network elements integrated and a large available user space.

The only development platform that even comes close to Facebook platform is OpenSocial, which is a platform that provides tools for basic social network development. It is not bound to any particular service and an application created using OpenSocial can be used with all services that support the platform. OpenSocial has not been adopted by many services, so the user space is fairly limited making OpenSocial somewhat less interesting to various operators.

3.3 Privacy

When interacting on social networks, people share vast amounts of personal data. The fact of data being personal makes maintaining a sense of privacy important, and therefore Facebook has set up a comprehensive privacy policy. Facebook users have versatile privacy settings, whereby they can manage how their information is shared within Facebook and other services. Users can set visibility limits on many different levels, such as friends, groups, networks, applications and third parties. For example, for a photo album, a user can set that only a group of his friends, all of his friends or friends and friends of friends can see it. These settings can be applied to almost any user content and social integration point. For example, the user cannot set privacy information to a single photo, but only to a photo album. The user can also set that his data is visible to third parties, but cannot select which external services can access his data.

When users make strict privacy settings it can affect Facebook platform applications. The data an application can see is filtered through users' privacy settings, and might not reveal the real circumstances. For example a user can restrict that applications his

friends are using cannot access his information. This setting affects almost all Facebook platform components. When the application requests a list of the user's friends, only the friends who allow the platform applications to access their information are listed.

3.4 Developer roadmap

Facebook has released a developer roadmap, which provides information about short term platform changes. The roadmap is a great way to let developers know what will be happening to the platform in a near future. It is very important for a developer to stay up to date with the roadmap, because otherwise some functions that would be instantaneously unusable could be created.

3.5 Platform pitfalls

3.5.1 API issues

There are some minor glitches with the API. The main usability problem is inconsistency throughout the API. Some of the method names are totally different from each other, even if they perform similar actions or vice versa. For example, the *users.getInfo* method fetches metadata about users and the *photos.getAlbums* method acts likewise for photo albums.

Also the return values of some of the methods are incoherent. A method can be called with a single value or an array of values, but in both cases the method returns a two-dimensional array. It would make much more sense if a method could only be called with an array, or if the return value was a single-dimensional or two-dimensional array depending on the argument type used.

Some of the API requests take a rather long time, and may cause a timeout if too many requests are made. While using the API, method calls should be limited only to the ones really needed. To help to deal with the timeout problem, some of the API methods can return the same data about multiple subjects with one request. Also by using FQL queries, the number of API requests can be lowered.

The problem with timeouts is definitely a serious issue. Debugging timeout related problems can be difficult, because sometimes a timeout is hard to identify. It may seem that data sent to the API could be incorrect, or a problem with parsing the return data occurs, but instead, execution is stopped because of a timeout. It is important for the developer to identify these kinds of situations and be aware of time consuming API operations.

3.5.2 FBML problems

While FBML tags are extremely useful, they have a few drawbacks also. The biggest problem is, that all FBML tags are only available to FBML type applications. Only a small subset of FBML tags called XFBML is available in iframe type applications. Another problem with some FBML tags is nonworking parameters. All available parameters are listed in documentation, but they have no effect on tag output. This is not as big a problem with tags as it is with documentation.

One minor problem with component tags is that the component data can only be accessed by using API requests or FQL. And since these components communicate only with Facebook, it is hard for a developer to know when new data is added to the application's components. This is not a serious issue, but there might be situations where all data in an application needs to be kept track of.

3.5.3 Platform changes

The platform is changing quite frequently and application developers need to stay informed about the changes. Many of the changes can be considered as major changes, because they change the way applications implement social communication. For example only within two years Facebook has changed twice the way an application publishes user actions. Frequently changing platform components require constant updating of applications, and increase development times.

3.5.4 Communication

Communication by Facebook related to the platform changes is somewhat poor. The only information available is shared through Facebook developer wiki pages and

postings on the developer application and developer website. The main problem is that the information is scattered to many different places, and is hard to find. The best way for a developer to get up-to-date information about platform changes is to subscribe to these information channels or visit them frequently. A platform development roadmap is also available, where developers can see some long term information about platform changes. The developers should be informed more clearly and effectively when a major change to the API is made, because it may cause an application to stop working.

3.5.5 Documentation

There are lots of problems with the platform documentation. The biggest problem is that the information is scattered all over different places. There is a developer website that has guides, tools and platform rules, and there also exists a developer wiki that has detailed information and usage examples about the platform components. In order to successfully build a properly working application, information needs to be gathered from multiple locations that are not even listed anywhere. In some cases the information differs between two places.

The developer website is well organized and has extremely valuable information and tools. Some of the information in the developer website should be on the wiki pages and vice versa. Otherwise the developer website is in much better shape than the Facebook developer wiki. The wiki is not organized in a reasonable way, and finding an answer to a trivial question might take an excessive amount of time. The wiki pages should be organized in a consistent manner, and a naming scheme for pages should be enforced. Currently some related pages might not be linked at all, and valuable information is hard to find.

4 Facebook application

4.1 Anatomy of an application

A Facebook application is a large collection of distinct items that work together. Since interacting among peers is the main idea in social applications, most of the application parts are related to communication. By providing many integration points to Facebook, many different kinds of applications can be built.

4.1.1 Canvas

The canvas is the main part of an application and it is what separates an application from others. It is the part of the application which is rendered inside a Facebook page, as if it were a part of the main service. All main actions should be made in the canvas section, while all other parts of the applications are connected to it. Other application parts are either triggered through user actions on the canvas page, or contents of other application parts are altered based on actions made in the canvas section. [25]

4.1.2 Application profile

The application profile is like a profile or an About-page of an application. This is where users go to get information like who is using the application or read user reviews. Like a normal profile page, the application profile has a wall where users can write comments. The application profile can be used to give detailed information about the application, and communicate with users on a discussion board. This is a good place to get feedback from users and tell them about new application features. [25]

4.1.3 Profile box

The profile box is a part of an application which can be placed on a user’s profile page. An application user can opt-in to add an application box to his profile, or remove it later. Because the profile page has information related only to one user, the profile box should also contain information from one user only. Therefore the profile box is an efficient way to display information about the user's actions made in the application. A

profile box can be thought as an advertisement for an application, it is a convenient way to get more people to use the application. [25]

4.1.4 Profile tab

A profile tab is like a bigger version of a profile box. Where as a profile box has a small area on a user's profile page, with a profile tab a whole page in the user's profile can be used to display application data. A profile tab can be considered as something between a canvas page and a profile box. It is a large area to display application data, but with contents from only one user. [25]

4.1.5 Notifications

Notifications are Facebook's way to inform users that something closely related to them has happened within their network. An application can notify application users about new content, or an application user can notify his friends about something he did within the application. Notifications are an easy way to get users to come back to the application canvas page, and also to get new users for the applications by sending user-to-user notifications. To prevent spamming, there is a daily limit for invites set by Facebook. [25]

4.1.6 Alerts

An alert is like an extended notification, which is sent from Facebook to users via email. Alerts can have more content than notifications and they are sent to the personal email account provided. This is a good way to tell users that are not currently online in Facebook that something has happened and they should log in. Alerts have also a daily limit. [25]

4.1.7 Invites

Inside of an application, users can interact only with other users using the application. Users can send invites to their friends in order to get them to use the application. The invite function, one way or other, is crucial to an application. Invites are the best way to

get more users into an application by taking advantage of a word-of-mouth effect. Like notifications, invites have a daily limit. [25]

4.1.8 News feed

News feed stories are a crucial part of an application. Users can publish information about their actions in an application through news feed. Unlike notifications and invites that are sent to a specific set of users, news feed stories are visible to everyone who can see the user's profile. News feed stories do not have a daily sending limit, but the user has to authorize all publications. News feed stories can contain attachments like photos or videos, which makes them more versatile than notifications. Because of the special content and wide audience, news feed stories are a perfect implementation of the word-of-mouth effect. [25]

4.2 Application type

There are two different ways to build a Facebook application. An application can be FBML or iframe type, the application type defining how the canvas page of the application is rendered in Facebook. The type of the application limits which Facebook platform components are available for use in an application, and more importantly what software libraries can be used in the development process. [26]

When deciding which application type to use, the developer must carefully think what Facebook platform features are needed in the application and what software libraries are used in the development process. The application type only affects the canvas part of the application. Because all other parts of the application are only triggered or altered from the canvas, they can be created in the same way regardless of the application type.

4.2.1 FBML

FBML type applications have full platform support, meaning that all features of the platform can be used in an application without any problems. FBML type applications are heavily platform-integrated, and can be quickly built using available platform components. FBML applications are always processed on a Facebook server, which might make them slower than iframe applications. [26]

While FBML type applications can use all of the features the Facebook platform offers, the use of some popular software libraries becomes nearly impossible. Since the Facebook platform does not support all HTML elements and uses a modified version of JavaScript, all libraries that create client-side executable code cannot be used. [26]

If an application is built from scratch, FBML application is a quick and easy way to build the application. Therefore it is especially suitable for new Facebook application developers. However, current web development tools do not have support for FBML or FBJS, so debugging FBML and FBJS syntax might create problems.

4.2.2 Iframe

An iframe application is basically a traditional website displayed inside an iframe element at the Facebook end. Since the application is inside of an iframe, no output processing is done on Facebook servers. This means that platform features like FBML and FBJS are not available or have very limited support. For example, very usable features like Facebook dialogs cannot be used. FBML elements that can be used in an iframe application are referred to as XFBML tags in documentation. FBJS is not available at all for iframe applications, but Facebook JavaScript client library can be used. [26]

Because no processing is done on Facebook servers, iframe applications tend to work faster than FBML applications. This is important especially in situations where a great deal of AJAX is used, because AJAX requests can be sent directly to an application server instead of sending them through a Facebook server. Because data for XFBML tags is populated after a page loads, iframe applications might feel less user friendly if large numbers of XFBML tags are used. [26]

While the support of the Facebook platform is limited, more software libraries can on the other hand be used in the application development. An iframe application is a good choice in a situation where an application already exists or no FBML support is needed. When building an iframe type application, developers can use tools that they are already familiar with. Therefore, while some FBML components are very useful they can quickly be replicated with familiar tools by experienced developers.

Since there are no limitations on what HTML elements and software libraries can be used for an iframe application, only minor or no changes at all are needed to integrate a previously developed web application. The only changes necessary are application authentication and possible user authentication. By using iframe type applications, software developers can quickly integrate existing web applications with Facebook.

4.3 Application data location

Since Facebook does not host the applications, they have to be located on a third-party server. For each request made to Facebook application page, a request is made to a third-party server to get the application data. Facebook simply fetches the data from the external server, processes it and integrates it into the service.

4.3.1 Application URL schema

An application can have more than one page, and the data on each page can be different. Facebook has to know somehow what data on the external server corresponds to the URL provided. Thus, a simple URL schema has been created to fetch the correct application data for every possible application URL.

The Facebook application URL is transformed into a callback URL using the canvas callback URL provided by the developer. If the requested application URL is *http://apps.facebook.com/example/info.php?id=12* and canvas callback URL is *http://www.example.com/application/* the created callback URL is *http://www.example.com/application/**info.php?id=12***. The only thing that happens in the transformation is that the canvas URL, which is always in the form of *http://apps.facebook.com/example/*, is replaced with the canvas callback URL. [27]

By building the callback URL this way, a request can be made to any subfolder and HTTP GET parameters can also be sent with the request. When providing the canvas callback URL, it is important to include a slash at the end, because the canvas URL always has it. If the last slash is missing, the callback request might be sent to the wrong address. [27]

4.3.2 Application request process

The integration of application data from an external server into a Facebook canvas is a fairly simple process. Because there are two application types, which display the application data in different ways, there are two different kinds of chains of events.

4.3.2.1 FBML application

In an FBML application users always use Facebook canvas URLs to access an application. When the Facebook server receives the request it processes possible preloaded FQL queries and sends a POST request to a canvas callback URL with all required and requested data. This is where the actual application starts to do its part. Possible API calls are made and application output is sent back to the Facebook server. When the Facebook server receives the application data it processes all FBML and JSON entries and builds the final output that is sent to the user's client. [26]

4.3.2.2 Iframe application

When a user accesses an iframe application using the Facebook canvas URL, the Facebook server returns a document which has only the Facebook chrome and an iframe element with a callback canvas URL assigned to it. When the user's client receives this response, iframe contents are fetched from an application server. This is where the actual application starts to work, and makes API calls to the Facebook server possible. The application server sends the output directly to the iframe element on the user's client. If the output document contains any XFBML elements they are processed at this point, and appropriate data is fetched from the Facebook server. After this point the user can reload only the iframe contents or the whole page. [26]

4.4 Application directory

Facebook has an application directory where application developers can exhibit their creations. Submitting an application to the directory is not required, but it is an efficient way to get the word out. The application directory has currently over 500 000 applications listed [28]. The applications can be browsed based on popularity and newness. Applications are also divided into multiple categories based on the type of use,

which makes it easier to find a particular kind of application. The application directory is an excellent place to find new, not so popular applications and to do research for possible competition.

The application directory has some information about each listed application. This includes a logo, a short description and a possible rating for the application, if the application has been rated by users. From the listing users are redirected to the About-page of an application, where users can find more information and add the application to their profile. It is advisable to submit the application to the directory. This way the developer can get more publicity for the application and information about the popularity of the application compared to other applications.

5 Development

Facebook application development is very similar to traditional web development. The only differences are special languages used by the platform, multiple integration points and the application request process. Working with the platform languages is pretty simple, they just replace very similar languages that are normally used. The biggest difference is the multiple integration points, which makes the application more scattered. The request process does not interfere with the development process, but only in debugging situations.

5.1 Development requirements

Before development can be started, a few required things need to be done. Facebook requires all developers to register themselves. This is done in Facebook by accessing the developer application, which is a place where all application configurations are made. For every application an API key & application secret pair has to be registered. They are automatically created when an application is set up. The API key and application secret are used to identify the application, which allows it to access Facebook resources. [29]

Because Facebook does not host the applications, an online resource has to be available for the application. The application data has to be on a server that Facebook can access in order to integrate it into the service. A client library has to be used in order to identify the application and use the platform resources. Client libraries are available for download on the developer wiki pages. [29]

5.2 Development tools

Facebook provides some very useful tools that help in the development process and also in information logging. A test console is available for FBML markup because it is not supported by any editors, and the API also has its own test console. Statistical tools are available for use in debugging of the application request process and for studying of application reliability.

5.2.1 FBML test console

The FBML test console is a tool where application developers can debug their FBML markup. This is a powerful tool to test that the data sent to Facebook server is well formed and valid. FBML is rather hard to debug within the application itself, and development environments do not have support for FBML tags. The test console provides a preview of a tag output with a number of different attributes that can be easily changed to verify correct operation in every situation.

5.2.2 API test console

As with the FBML test console, the API test console can be used to test and debug the Facebook API functionality. The test console has all API methods available for testing, and automatically provides input fields for all parameters needed for a method that is being tested.

Different API requests can be tested in the API test console as if the requests were actually sent from an application. This way the debug data is valid for every unique application. Debugging can be done without committing any changes to Facebook data, while getting valid return values. Since all operations performed in an application can be tested with actual data beforehand, it makes application development and debugging much faster and the application more reliable.

5.2.3 Statistics

Facebook offers statistical tools for developers to examine application traffic and resource usage. Statistical tools have many different sections that can help to make an application better and work faster, or they can be used just to gather information about application usage.

A usage view can be used to determine how many API calls have been made over a period of time, or what the average canvas page request time at a particular moment has been. This is really important information in order to make the application work faster.

A features view is an easy way to see how much each part of an application has traffic. And more importantly, it tells if the canvas page has had runtime or parsing errors.

5.3 Guiding principles

Facebook has published a list of principles and policies that needs to be followed when developing an application. The list of rules is fairly long, but the main message is one of trustworthiness and great user experience. This means not to mislead or spam users, and to respect their privacy. Also giving the control to users and promoting social interaction are important issues. The complete list of policies can be found on the developer website. [30]

5.4 Application configuration

The application setup process is really easy and straightforward. There are only a few settings that are needed to make an application work. Even though the basic setup process is simple, there is a large set of configuration options that can be used to tune the application functionality.

5.4.1 Required settings

First of all, an application must have a unique name. The name cannot contain the word face in it, because all application names containing the word face are reserved for Facebook's own use. Another required setting that is visible to users is the canvas page URL. This URL must also be unique, and normally something close to the application name. These two settings are the only values that are visible to users, so they should reflect the purpose of an application. [29]

A canvas callback URL has to be provided in order for Facebook to know where to fetch application data. When providing the canvas callback URL, it is important to include a slash at the end because the canvas URL always has it. If the last slash is missing, the callback request might be sent to wrong the address. Other required options are developer contact email and user support address. These addresses are prefilled with the email address of the user who creates the application. [29]

5.4.2 Useful optional settings

The most important optional setting is the render method. This setting defines if an application is an FBML or iframe type application. By default all applications are FBML type.

Post-authorize and post-remove callback URLs are extremely useful in user tracking. When a user authorizes or removes an application, a request with user information is sent to either one of these URLs, if the URL is set. This way specific time of application authorization and removal can be recorded.

Another useful URL setting is the post-authorize redirect URL. If provided, right after authorization a user is redirected to this URL instead of the URL he used to enter the application. This option is extremely useful, if it is wished that a user visits a certain application page before using the application.

One setting that is useful in development stage is sandbox mode. When an application is set to be in sandbox mode, only users that are defined as developers of the application can access it.

5.5 Application type

The chosen application type has the most significant impact on the development process. As mentioned before, the application type restricts the use of either some Facebook platform features or external JavaScript libraries. Some functionalities might need to be recreated, since all normally available solutions are not usable.

Popular software libraries like jQuery cannot be used in FBML applications. This is really unfortunate, because many libraries like jQuery provide a wide range of useful functions. An FBML type application restricts the developer to use a small set of tools, but also provides powerful platform related tools.

5.6 Publishing an application

As mentioned before, applications can be submitted to the Facebook application directory. To make sure that an application is listed and the submission process runs smoothly, the application has to work correctly and cope with all application directory requirements.

5.6.1 Publication requirements

Facebook has set some requirements for applications before they can be listed into the application directory. To make sure that an application is an actual working application, it must have the canvas page URL set. The application also must have a description and a logo, since these are the main things displayed in the application directory. Before an application can be accepted it must also have users. Five total users or 10 monthly active users are required. An active user is a user that interacts with an application, and need not to have authorized the application. The interaction can happen for example through profile boxes or tabs. [29]

While not required, Facebook recommends to add the application into a category. This makes it easier to find the application, which is convenient for the developer and other users. An application can be listed into two categories.

5.6.2 Publishing procedure

The publishing procedure is pretty straightforward. When an application fulfills all platform requirements, it can then be submitted for a verification from the developer application. The verification process takes a few days, since every application submitted is checked. The status of the verification process can be seen also from the developer application.

If something is wrong with the application, a developer will be informed and asked to do the required changes to the application. A problem may occur if the application does not fulfill the Facebook policies, which include the copyright policy, the application content policy and the features and functionality policy. This means that an application cannot be confusing or misleading, and it cannot infringe on anyone’s copyrights nor

have adult content. And if users are allowed to upload content through an application, the application must have a copyright policy viewable for users. The application can be resubmitted for verification after the changes required by Facebook have been made.

5.7 Development problems

During the development process multiple non-application specific issues were raised. Some of these problems were related to the platform functionality and some were issues with operational approaches of basic HTML elements.

5.7.1 Application type restrictions

The most difficult problem in the development process was the choice between Facebook features and software libraries when deciding on application type. An enormous amount of time should be spent on planning the application type, because it has such an immense impact on the development process. Depending on the application type, some platform components or software libraries could not be used and the functionalities had to be dropped or re-created.

When creating a FBML type application, a drag-and-drop feature was abandoned because the software library could not be used. This choice made the user interface less friendly, but the feature was not significant and it would have taken ample time to re-create it. In an iframe application a *fb:comments* tag could not be used, and the same functionality was re-created because interaction in a social application is a fundamental feature.

5.7.2 FQL preloading

Using the FQL preload feature in an iframe application is more complicated than in an FBML application. This is because page requests aren't always sent to a Facebook server, which would trigger an FQL query and provide the data. By default FQL preloading works only if requests are made to a Facebook server. If requests are made straight to an application server, preloaded data can be accessed only with the JavaScript client library. This means that data fetched with preloaded FQL cannot be used on the server side.

5.7.3 Iframe links

When building an iframe type application a few problems with links within the application was found. Since the application canvas is displayed in an iframe, the links by default refer to the iframe element instead of the browser window. This problem is at worst when a user refreshes the browser page. The address in the browser's address bar might be different from the address that was used to fetch the data currently in the iframe. Thus the contents of the iframe could not be set back to previous state after a reload.

This problem can be solved by setting the value of the target attribute in links to *_top*. This way when a link is clicked the whole page is loaded instead of only the iframe contents. This makes the application work more slowly, because the whole page is always reloaded rather than just the iframe. But this way the problem with FQL preloads is also solved, because requests are always sent to a Facebook server. Facebook has released an iframe URL feature that deals with this particular problem, but the feature is currently in early beta state. [31]

5.7.4 File uploads

Because of the application request process a problem was encountered when creating a file upload feature. When a POST request with a file upload information is sent to a Facebook canvas URL, the file upload information is not passed to the request that is made to the application canvas callback URL. This means that file uploads can be made only when communicating directly with the application server and leaving the Facebook site. The problem only exists in an FBML type application, because an iframe application already communicates directly with the application server.

Uploading a file using AJAX is not possible, since the FBJS AJAX object does not support sending form data. One solution to the problem is to create a popup window that is directly connected to the application server, but this approach is rather clumsy. It makes the application less user friendly and the connection to the Facebook look and feel is lost. The best solution is to handle the file upload using an iframe in a Facebook

dialog. This way a direct connection to the application server is made by using the iframe while application style is retained with the dialog.

5.7.5 User tracking

Post-authorize and post-remove URLs provide an easy way to keep track of users making use of an application. A problem arises when a developer wants to know how a user entered the application first time. This is no important issue, but surely the information could come in handy if the application is to be used for marketing purposes.

Users can enter the application by an invitation or notification, clicking on a link in a friend's news feed, or from the application directory. And, obviously, a user can enter the application URL directly to a browser. It is very time consuming to build a system that can tell which way the user entered the application. And even so, the system cannot be totally sure about the entry point, because the HTTP_REFERER parameter is not available for use at the application server. This is because the first application entry is done through a Facebook server due to authorization, and after it, the Facebook server sends a modified POST request to the application server. Invites, notifications and news feed stories have an application URL attached, and this URL could contain a special identifier revealing the entry point. But this method only covers some of the possible ways to enter the application.

5.7.6 Application submission

Minor problems might arise in application submission. Because the platform policy list is very precise and long, some policies might not have been followed. For example if an application has an upload feature of any kind, a copyright policy has to be made available for application users. These are only minor issues that can be addressed easily, and they are pointed out by Facebook staff.

6 Conclusions

Facebook applications are powerful tools that can be used in many different ways. Many extremely popular applications built for entertaining, advertising and profiting purposes have been released. The applications can be simple and quick to build, or very complex and yet easy to create. The Facebook platform offers multiple ways to build an application, and a wide range of useful tools to use in development process.

Facebook application development doesn't differ much from traditional web development, the main difference is the perspective. Facebook applications are built for social purposes, as tools to interact with or as something to interact for. What really sets Facebook applications apart is access to personal data. Because the Facebook platform provides personal data to be used, highly interesting applications can be built.

Many problems were discovered in the Facebook platform and in the development process. The platform components have some inconsistencies and usability issues, but more serious problems are found in the documentation. The documentation is not organized and is scattered to multiple outlets, which is why it becomes a major issue. Facebook also develops its services at a very rapid pace, which causes the documentation not always to be up-to-date and applications to require constant maintenance. Other problems with development are fairly easy to solve, and only require some time to be addressed.

The main objective of this project was to gain information about the possibilities of the Facebook platform. The capabilities of the platform were mapped out in a detailed manner, which revealed many shortcomings. First-hand information was also gathered by building two experimental applications. All project objectives are covered in this thesis, and the project can be considered successful.

While this thesis provides comprehensive information about Facebook application development, there is plenty of room for future studies. Only the main parts of Facebook platform have been discussed, because the platform is constantly evolving with new features being introduced all the time. The knowledge that was gained during this project can be used in future assignments. The knowledge of the platform

environment and its special functionalities, will make the following development projects more successful. This means more powerful applications utilizing all platform features to the full.

References

1 Facebook, Inc. News - The New York Times [online]. The New York Times; 27 May 2009. URL: http://topics.nytimes.com/top/news/business/companies/facebook_inc. Accessed 10 March 2010.

2 Facebook | Timeline [online]. Facebook Inc. URL: http://www.facebook.com/press/info.php?timeline. Accessed 10 March 2010.

3 Alexa Top 500 Global Sites [online]. Alexa Internet Inc. URL: http://www.alexa.com/topsites. Accessed 10 March 2010.

4 World Internet Usage Statistics News and World Population Stats [online] Miniwatts Marketing Group; 12 February 2010. URL: http://www.internetworldstats.com/stats.htm. Accessed 10 March 2010.

5 facebook.com - Site Info from Alexa [online]. Alexa Internet Inc. URL: http://www.alexa.com/siteinfo/facebook.com#trafficstats. Accessed 10 March 2010.

6 Social Networking Websites Review 2010 - TopTenREVIEWS [online]. TopTenREVIEWS. URL: http://social-networking-websites-review.toptenreviews.com. Accessed 10 March 2010.

7 BBC NEWS | Technology | 15 reasons Facebook may be worth $15bn [online]. BBC; 25 October 2007. URL: http://news.bbc.co.uk/2/hi/7061398.stm. Accessed 10 March 2010.

8 Facebook Developers | Facebook Platform Launches [online]. Facebook Inc; 28 May 2007. URL: http://developers.facebook.com/news.php?blog=1&story=21. Accessed 10 March 2010.

9 Authorizing Applications - Facebook Developer Wiki [online]. Facebook Inc; 4 February 2010. URL: http://wiki.developers.facebook.com/index.php/Authorizing_Applications. Accessed 10 March 2010.

10 Bickel, Bob. Bickel Blog: Why Develop a Facebook Application? [online]. Bob Bickel; 18 April 2008. URL: http://bobbickel.blogspot.com/2008/04/why-develop-facebook-application.html. Accessed 10 March 2010.

11 API - Facebook Developer Wiki [online]. Facebook Inc; 26 February 2010. URL: http://wiki.developers.facebook.com/index.php/API. Accessed 10 March 2010.

12 Admin.banUsers - Facebook Developer Wiki [online]. Facebook Inc; 10 August 2009. URL: http://wiki.developers.facebook.com/index.php/Admin.banUsers. Accessed 10 March 2010.

13 Friends.getMutualFriends - Facebook Developer Wiki [online]. Facebook Inc; 4 November 2009. URL: http://wiki.developers.facebook.com/index.php/Friends.getMutualFriends. Accessed 10 March 2010.

14 Users.setStatus - Facebook Developer Wiki [online]. Facebook Inc; 4 November 2009. URL: http://wiki.developers.facebook.com/index.php/Users.setStatus. Accessed 10 March 2010.

15 Photos.createAlbum - Facebook Developer Wiki [online]. Facebook Inc; 2 March 2010. URL: http://wiki.developers.facebook.com/index.php/Photos.createAlbum. Accessed 10 March 2010.

16 Events.getMembers - Facebook Developer Wiki [online]. Facebook Inc; 4 November 2009. URL: http://wiki.developers.facebook.com/index.php/Events.getMembers. Accessed 10 March 2010.

17 FBML - Facebook Developer Wiki [online]. Facebook Inc; 16 December 2009. URL: http://wiki.developers.facebook.com/index.php/FBML. Accessed 10 March 2010.

18 FBMLspec - Facebook Developer Wiki [online]. Facebook Inc; 12 September 2009. URL: http://wiki.developers.facebook.com/index.php/FBMLspec. Accessed 10 March 2010.

19 FBJS - Facebook Developer Wiki [online]. Facebook Inc; 7 November 2009. URL: http://wiki.developers.facebook.com/index.php/FBJS. Accessed 10 March 2010.

20 FBJS/Animation - Facebook Developer Wiki [online]. Facebook Inc; 6 November 2009. URL: http://wiki.developers.facebook.com/index.php/FBJS/Animation. Accessed 10 March 2010.

21 FQL - Facebook Developer Wiki [online]. Facebook Inc; 1 October 2009. URL: http://wiki.developers.facebook.com/index.php/FQL. Accessed 10 March 2010.

22 FQL Tables - Facebook Developer Wiki [online]. Facebook Inc; 4 March 2010. URL: http://wiki.developers.facebook.com/index.php/FQL_Tables. Accessed 10 March 2010.

23 Functions and Operators - Facebook Developer Wiki [online]. Facebook Inc; 9 October 2009. URL: http://wiki.developers.facebook.com/index.php/Functions_and_Operators. Accessed 10 March 2010.

24 Preload FQL - Facebook Developer Wiki [online]. Facebook Inc; 13 February 2010. URL: http://wiki.developers.facebook.com/index.php/Preload_FQL. Accessed 10 March 2010.

25 Anatomy of a Facebook App - Facebook Developer Wiki [online]. Facebook Inc; 18 November 2009. URL: http://wiki.developers.facebook.com/index.php/Anatomy_of_a_Facebook_App. Accessed 10 March 2010.

26 Choosing between an FBML or IFrame Application - Facebook Developer Wiki [online]. Facebook Inc; 5 November 2009. URL: http://wiki.developers.facebook.com/index.php/Choosing_between_an_FBML_or_IFrame_Application. Accessed 10 March 2010.

27 Canvas Callback URL - Facebook Developer Wiki [online]. Facebook Inc; 4 December 2009. URL:

http://wiki.developers.facebook.com/index.php/Canvas_Callback_URL. Accessed 10 March 2010.

28 Facebook | Statistics [online]. Facebook Inc. URL: http://www.facebook.com/press/info.php?statistics. Accessed 10 March 2010.

29 Creating a Platform Application - Facebook Developer Wiki [online]. Facebook Inc; 4 March 2010. URL: http://wiki.developers.facebook.com/index.php/Creating_a_Platform_Application. Accessed 10 March 2010.

30 Facebook Developers - Platform Policy [online]. Facebook Inc; 10 March 2010. URL: http://developers.facebook.com/policy. Accessed 10 March 2010.

31 IFrame URL - Facebook Developer Wiki [online]. Facebook Inc; 11 September 2009. URL: http://wiki.developers.facebook.com/index.php/IFrame_URL. Accessed 10 March 2010.