Abstract of project presented to the School of Technology Management in fulfillment of the requirements for the degree of Bachelor in Computer Science.

**FACEBOOK AGGREGATOR FOR THE BLIND COMMUNITY**

By

**MOHAMMAD NAVEED**

**June 2010**

As for the abstract, it usually encompasses four (4) elements:  
  
1. Statement of problem of the research or project - issues addressed  
2. method used for implementation  
3. results and finding  
4. conclusion

World’s biggest social networking website Facebook.com currently has more than 600 million active users worldwide. It is being used every minute for different purposes, knowledge sharing, information gathering, personal sharing, and finding childhood friends etc, but only by sighted people, not by the blind or visual impaired people.

Facebook website pages contain many graphical and complex contents and they are arranged in a manner that the screen reader software is not able to read correctly in order to be understood by the blind user, it is almost impossible for the blind to use all the needed features of the Facebook.

This dissertation has covered a full report on the Facebook Aggregator system developed for the blinds to use social networking website Facebook easily.

Facebook Aggregator aggregate the necessary Facebook features, such as Status updates, YouTube video sharing, Emails and Messages etc, by using the Facebook Application Programming Interfaces (API), .NET Framework, and the JAWS screen reader software with some accessibility added.

Facebook Aggregator managed to extract, manipulate, and present the data in such a way that is easily read by the JAWS screen reader software.

# Acknowledgment

First and foremost I thank to ALLAH for helping me all the way from the very first day of my life.

Right after that I offer my sincerest gratitude to my supervisor, Mr. Amjad Hanesh, who has supported me throughout my studies and thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my bachelor's degree to his encouragement and effort and without him this project and thesis, too, would not have been completed or written. Besides my academic studies, I have learned more than a lot from him, which includes RFID technology, Dot Net Framework, Mobile development, CCTV camera programming, Arduino programming, and FTIR Multi Touch Screen development etc. One simply could not wish for a better and friendlier supervisor and the Italian restaurant he takes me to.

In my daily work I have been blessed and helped with a friendly and cheerful group of college staff, and fellow students, Miss. Privena (School of Technology Management Coordinator), Khoo Siaw Loo, Arfat Malik, and Waqas Rauf.

Beyond Computer Science, Awais Sarwar Malik has been a companionable housemate for many years as well as a colleague and possesses the greatest quality in a housemate. He shared all my stress I get from doing the studies by hanging out with me and doing fun.

More than anything else, I thank my parents for supporting me, praying for me, throughout all my studies at University, buying me the car, and travelling all the way from Libya to Malaysia just to see and encourage me.

Finally, I thank to my love “jagu” for giving me everything that led to this success.

# Declaration

I declare this project work in my original work except for the quotations and citation which has been accordingly acknowledged. I also declare this project work has not been previously, and is not currently, submitted for any other degree at Binary University College.

(Signature)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**MOHAMMAD NAVEED**

# List of Figures

[Figure 1: Web Accessibility components relation 4](#_Toc265504844)

[Figure 2: Use Case Diagram 23](#_Toc265504845)

[Figure 3: Requirement Diagram 27](#_Toc265504846)

[Figure 4: Generic overview of Facebook Aggregator 35](#_Toc265504847)

[Figure 5: Level 0 Data Flow Diagram 35](#_Toc265504848)

[Figure 6: Level 1 Data Flow Diagram 36](#_Toc265504849)

[Figure 7: All-in-one Architecture 39](#_Toc265504850)

[Figure 8: Flat Architecture Model 39](#_Toc265504851)

[Figure 9: Hub-and-spoke (or daisy) pattern 40](#_Toc265504852)

[Figure 10: Facebook Aggregator horizontal top bar 41](#_Toc265504853)

[Figure 11: Facebook Aggregator paging interface 41](#_Toc265504854)

[Figure 12: Wall post FBML tag 45](#_Toc265504855)

[Figure 13: Facebook Aggregator unit folder 49](#_Toc265504856)

[Figure 14: Black Box testing 50](#_Toc265504857)

[Figure 15: Request Application Specific Help from Facebook form 54](#_Toc265504858)

# List of Tables

[Table 1: Actors Grid 23](#_Toc265504859)

[Table 2: Use Cases Grid 24](#_Toc265504860)

[Table 3: Use Case Diagram Summary 24](#_Toc265504861)

[Table 4: Operating Environment 27](#_Toc265504862)

[Table 5: Roadmap changes of Facebook APIs 30](#_Toc265504863)

[Table 6: Five levels of testing 49](#_Toc265504864)

[Table 7: Test Case Planning Format 51](#_Toc265504865)

[Table 8: Test cases for Facebook Aggregator 51](#_Toc265504866)

**TABLE OF CONTENTS**

[Abstract ii](#_Toc265545223)

[Acknowledgment iii](#_Toc265545224)

[Declaration iv](#_Toc265545225)

[List of Figures v](#_Toc265545226)

[List of Tables vi](#_Toc265545227)

[CHAPTER ONE INTRODUCTION 1](#_Toc265545228)

[1.1: Introduction 1](#_Toc265545229)

[1.2: Social Networking 1](#_Toc265545230)

[1.3: Web Accessibility 2](#_Toc265545231)

[1.3.1: The importance of web accessibility 2](#_Toc265545232)

[1.3.2: Making a Web site accessible 3](#_Toc265545233)

[1.3.3: Essential Components of Web Accessibility 3](#_Toc265545234)

[1.4: Assistive Technology 4](#_Toc265545235)

[1.4.1: JAWS screen reader 5](#_Toc265545236)

[1.5: Problem Statement 5](#_Toc265545237)

[1.6: Facebook Aggregator 5](#_Toc265545238)

[1.7: Objectives 6](#_Toc265545239)

[1.8: Layout of project write-up 6](#_Toc265545240)

[CHAPTER TWO LITERATURE REVIEW 7](#_Toc265545241)

[2.1: Web Accessibility 7](#_Toc265545242)

[2.1.1: History of Web Accessibility 8](#_Toc265545243)

[2.2: The needs that Web accessibility aims to address include: 9](#_Toc265545244)

[2.3: Different Disabilities that Can Affects Web Accessibility 10](#_Toc265545245)

[2.4: Main barriers that blinds have faced when browsing web 11](#_Toc265545246)

[2.5: User with blindness using the web 11](#_Toc265545247)

[2.6: Barriers that people with low vision have encountered on the Web 12](#_Toc265545248)

[2.7: List of some well accessible websites 12](#_Toc265545249)

[2.7: Assistive Technologies 14](#_Toc265545250)

[2.7.1: Alternative keyboards or switches 14](#_Toc265545251)

[2.7.2: Braille and refreshable Braille 14](#_Toc265545252)

[2.7.3: Scanning software 15](#_Toc265545253)

[2.7.4: Screen magnifiers 15](#_Toc265545254)

[2.7.5: Screen readers 15](#_Toc265545255)

[2.7.6: Speech recognition 15](#_Toc265545256)

[2.7.7: Speech synthesis 16](#_Toc265545257)

[2.7.8: Tabbing through structural elements 16](#_Toc265545258)

[2.7.9: Text browsers 16](#_Toc265545259)

[2.7.10: Visual notification 16](#_Toc265545260)

[2.7.11: Voice browsers 16](#_Toc265545261)

[2.8: Facebook.com 16](#_Toc265545262)

[2.8.1: Web Accessibility and Facebook 18](#_Toc265545263)

[2.9: The Gap 19](#_Toc265545264)

[CHAPTER THREE REQUIREMENT AND ANALYSIS 20](#_Toc265545265)

[3.1: Requirements Analysis in general 20](#_Toc265545266)

[3.1.1: Stakeholder identification 21](#_Toc265545267)

[3.1.2: Stakeholder interviews 21](#_Toc265545268)

[3.2: Software Requirements Specification 22](#_Toc265545269)

[3.2.1: USE CASE Diagram 23](#_Toc265545270)

[3.2.2: Requirement Diagram 27](#_Toc265545271)

[3.2.3: Operating Environment 27](#_Toc265545272)

[3.2.4: Constraints and Dependencies 28](#_Toc265545273)

[3.2.4.1: Dependency on Facebook APIs 28](#_Toc265545274)

[3.2.4.2: Dependency on Facebook Developer Toolkit library 31](#_Toc265545275)

[3.2.5: System Features requirement 32](#_Toc265545276)

[3.2.5.1: Feature 1: Status update 32](#_Toc265545277)

[3.2.5.2: Feature 2: Comment posting 32](#_Toc265545278)

[3.2.5.3: Feature 3: Comment extraction 32](#_Toc265545279)

[3.2.5.4: Feature 4: Wall 32](#_Toc265545280)

[3.2.5.5: Feature 5: Friends and their profile 32](#_Toc265545281)

[3.2.5.6: Feature 6: Photo extraction 32](#_Toc265545282)

[3.2.5.7: Feature 7: Email and Messages 33](#_Toc265545283)

[3.2.5.8: Feature 8: Weblink post 33](#_Toc265545284)

[3.2.5.9: Feature 9: Video post 33](#_Toc265545285)

[3.2.5.10: Feature 10: Login authentication 33](#_Toc265545286)

[3.2.5.11: Feature 11: Video extraction 33](#_Toc265545287)

[3.2.5.12: Feature 12: Text posting 33](#_Toc265545288)

[CHAPTER FOUR DESIGN 34](#_Toc265545289)

[4.1: Design in Software Engineering 34](#_Toc265545290)

[4.3: Flow of the data 35](#_Toc265545291)

[4.3: Website Architecture 37](#_Toc265545292)

[I ensure the following so JAWS screen reader can maximally utilized. 38](#_Toc265545293)

[4.3.1: Architecture Model 39](#_Toc265545294)

[4.3.1.1: All-in-one model 39](#_Toc265545295)

[4.3.1.2: Flat model 39](#_Toc265545296)

[4.3.1.3: Hub-and-spoke / Daisy model 40](#_Toc265545297)

[4.4: Navigation Design 40](#_Toc265545298)

[4.4.1: Horizontal top bar 40](#_Toc265545299)

[4.4.2: Paging 41](#_Toc265545300)

[CHAPTER FIVE IMPLEMENTATION 42](#_Toc265545301)

[5.1: Implementation in Software Engineering 42](#_Toc265545302)

[5.2: Before writing the Facebook Aggregator 42](#_Toc265545303)

[5.3: Selecting the right tools, programming language, technologies, and IDEs 43](#_Toc265545304)

[5.3.1: APIs and Toolkits 43](#_Toc265545305)

[5.3.2: Adoption of Facebook Developer Toolkit 44](#_Toc265545306)

[5.3.3: Using FQL (Facebook Query Language) 44](#_Toc265545307)

[5.3.4: Using FBML / XFBML 44](#_Toc265545308)

[5.3.5: FQL, Facebook Toolkit, and ASP.net 46](#_Toc265545309)

[CHAPTER SIX TESTING 47](#_Toc265545310)

[6.1: Testing in Software Engineering 47](#_Toc265545311)

[6.2: Goals and Types of Testing 47](#_Toc265545312)

[6.2.1: System Testing 47](#_Toc265545313)

[6.2.1.1: Integration Testing 48](#_Toc265545314)

[5.2.1.2: Release Testing 48](#_Toc265545315)

[6.2.2: Component Testing 48](#_Toc265545316)

[5.2.3: Interface Testing 49](#_Toc265545317)

[6.3: Black box testing 50](#_Toc265545318)

[6.4: Test Cases 51](#_Toc265545319)

[CHAPTER SEVEN FUTURE WORK/MAINTENANCE 55](#_Toc265545320)

[7.1: Recommendations and Features 55](#_Toc265545321)

[7.1.1: Photo and video uploads 55](#_Toc265545322)

[7.1.2: Notes 55](#_Toc265545323)

[7.1.3: Marketplace 55](#_Toc265545324)

[7.1.4: Events 56](#_Toc265545325)

[7.1.5: Groups 56](#_Toc265545326)

[7.1.6: Chat 56](#_Toc265545327)

[7.1.6: Notifications 56](#_Toc265545328)

[7.1.7: Feedbacks and Recommendations 56](#_Toc265545329)

[References 58](#_Toc265545330)

[APPENDIX A COMPLETE SOURCE CODE 61](#_Toc265545331)

# CHAPTER ONE INTRODUCTION

## 1.1: Introduction

Facebook.com currently has more than 600 million active users worldwide. It is being used every minute for different purposes, knowledge sharing, information gathering, personal sharing, and finding childhood friends etc, but only by sighted people, not by the blind or visual impaired people. Contents on Facebook.com are not very well presented in order to be readable by the JAWS screen reader, it is almost impossible for the blind to use all the needed features of the Facebook. Facebook Aggregator aggregate the necessary Facebook features, such as Status updates, YouTube video sharing, Emails and Messages etc, by using the Microsoft Facebook Developer Toolkit (the Facebook Application Programming Interfaces (API) client library), ASP.NET, and the JAWS screen reader software. I this dissertation, we will go through all the step that were involved in the development process of the system.

## 1.2: Social Networking

Social Networking, It's the way the 21st century communicates today. Social networking is the grouping of individuals into specific groups, like small rural communities or a neighborhood subdivision, if you will. Although social networking is possible in person, especially in the workplace, universities, and high schools, it is most popular online. This is because unlike most high schools, colleges, or workplaces, the internet is filled with millions of individuals who are looking to meet other people.

When it comes to online social networking, websites are commonly used. These websites are known as social sites. Social networking websites function like an online community of internet users. Depending on the website in question, many of these online community members share common interests in hobbies, religion, or politics. Once you are granted access to a social networking website you can begin to socialize. This socialization may include reading the profile pages of other members and possibly even contacting them.

The friends that you can make are just one of the many benefits to social networking online. Another one of those benefits includes diversity because the internet gives individuals from all around the world access to social networking sites. This means that although you are in the United States, you could develop an online friendship with someone in Denmark or India. Not only will you make new friends, but you just might learn a thing or two about new cultures or new languages and learning is always a good thing.

The main types of social networking services are those which contain category places (such as former school-year or classmates), means to connect with friends (usually with self-description pages) and a recommendation system linked to trust. Popular methods now combine many of these, with Facebook, MySpace and Twitter widely used worldwide.

## 1.3: Web Accessibility

Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging. [1]

Web accessibility encompasses all disabilities that affect access to the Web, including visual, auditory, physical, speech, cognitive, and neurological disabilities. The document "How People with Disabilities Use the Web" describes how different disabilities affect Web use and includes scenarios of people with disabilities using the Web. [1]

Millions of people have disabilities that affect their use of the Web. Currently most Web sites and Web software have accessibility barriers that make it difficult or impossible for many people with disabilities to use the Web. As more accessible Web sites and software become available, people with disabilities are able to use and contribute to the Web more effectively.

Web accessibility also benefits people without disabilities. For example, a key principle of Web accessibility is designing Web sites and software that are flexible to meet different user needs, preferences, and situations. This flexibility also benefits people without disabilities in certain situations, such as people using a slow Internet connection, people with "temporary disabilities" such as a broken arm, and people with changing abilities due to aging. The document "Developing a Web Accessibility Business Case for Your Organization" describes many different benefits of Web accessibility, including benefits for organizations. [1]

### 1.3.1: The importance of web accessibility

The Web is an increasingly important resource in many aspects of life: education, employment, government, commerce, health care, recreation, social networking and more. It is essential that the Web be accessible in order to provide equal access and equal opportunity to people with disabilities. An accessible Web can also help people with disabilities more actively participate in society

The Web offers the possibility of unprecedented access to information and interaction for many people with disabilities. That is, the accessibility barriers to print, audio, and visual media can be much more easily overcome through Web technologies. [1]

### 1.3.2: Making a Web site accessible

It can be simple or complex, depending on many factors such as the type of content, the size and complexity of the site, and the development tools and environment.

Many accessibility features are easily implemented if they are planned from the beginning of Web site development or redesign. Fixing inaccessible Web sites, like Facebook, can require significant effort, especially sites that were not originally "coded" properly with standard XHTML markup, and sites with certain types of content such as multimedia.

### 1.3.3: Essential Components of Web Accessibility

It is essential that several different components of Web development and interaction work together in order for the Web to be accessible to people with disabilities. These components include:

* Content - the information in a Web page or Web application, including:
* Natural information such as text, images, and sounds
* Code or markup that defines structure, presentation, etc.
* Web browsers, media players, and other "user agents"
* Assistive technology, in some cases - screen readers, alternative keyboards, switches, scanning software, etc.
* Users' knowledge, experiences, and in some cases, adaptive strategies using the Web
* Developers - designers, coders, authors, etc., including developers with disabilities and users who contribute content
* Authoring tools - software that has been used to develop the website
* Evaluation tools - Web accessibility evaluation tools, HTML validators, CSS validators, etc. [2]

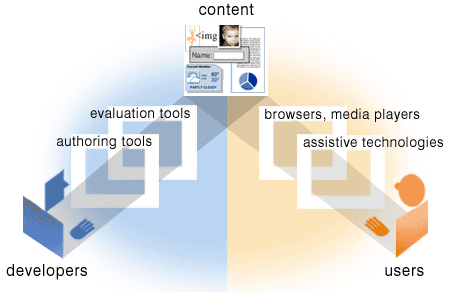


Figure 1: Web Accessibility components relation

http://www.w3.org/WAI/intro/relate.png

Web developers usually use authoring tools and evaluation tools to create Web content.

People ("users") use Web browsers, media players, assistive technologies, or other "user agents" to get and interact with the content.

## 1.4: Assistive Technology

Assistive Technology is technology used to perform those functions that might otherwise difficult or impossible to achieve, for the community of disables. It can include mobility devices such as walkers and wheelchairs, as well as hardware, software, and peripherals that assist people with disabilities in operating computers or other devices for information technologies. A good example could be people with limited function using special keyboards with large keys, or text-to-speech-engine like JAWS allows blind community to interact with computers by reading the text on the screen through a computer-generated voice, for them, JAWS is used in Facebook Aggregator as a screen reading component. Other examples could be the built-in accessibility features offered by some leading operating systems, for instance magnifier, narrator in almost all versions of Microsoft 's Windows. Assistive technology covers numerous aspects of disable person's life, to accomplish daily living tasks, from communication to education, work or recreation activities, in essence, helping them to achieve tasks independently and enhance their quality of life. Technically, Assistive Technology includes mechanical, non-mechanical and non-electronic aids, electronic, microprocessor based equipment, specialized instructional materials, services, and strategies that can either to assist them in learning, making the environment more accessible. [3]

### 1.4.1: JAWS screen reader

JAWS is screen reader is a software application that attempts to identify and interpret what is being displayed on the screen. Its purpose is to make personal computers using Microsoft Windows accessible to blind and visually impaired users. It accomplishes this by providing the user with access to the information displayed on the screen via text-to-speech or by means of Braille display and allows for comprehensive keyboard interaction with the computer.

It also allows users to create custom scripts using the JAWS Scripting Language, which can alter the amount and type of information which is presented by applications, and ultimately makes programs that were not designed for accessibility (such as programs that do not use standard Windows controls) usable through JAWS. [4]

## 1.5: Problem Statement

Since September 2006, anyone over the age of 12 with a valid e-mail address can become a Facebook user. Facebook's is target audience more for an adult demographic than a youth demographic. Users can add friends and send them messages, share pictures and videos, and update their personal profiles to notify friends about themselves. Additionally, users can join networks organized by workplace, school, or college. The website currently has more than 600 million active users worldwide. Facebook is being used every minute, every moment but not for blinds or visual impaired people. Because Facebook contains many graphical contents and the pages are full of different contents and they are arranged in a manner that the screen reader software does not read. There are blind users on Facebook but it is almost impossible for them to use all the needed features of the Facebook.

## 1.6: Facebook Aggregator

All of us know about Facebook but what is Facebook Aggregator? In general internet terms, a news aggregation website is a website where headlines are collected, In computing, a feed aggregator, also known as a feed reader, RSS reader or simply aggregator, is client software or a Web application which aggregates syndicated web contents such as news headlines, blogs, vlogs, podcasts in a single location *for easy viewing.* [5]

*“For easy viewing? Not necessarily every person in the world can view.*

Facebook Aggregator is a Web-Based system which aggregates the Facebook contents and features, not for easy viewing but for easy accessing so the blinds can also access the Facebook easily. Facebook Aggregator is the combination of Web Accessibility and Assistive technology.

## 1.7: Objectives

The objective of the project is to develop an aggregation website using ASP.NET and Facebook Developer Toolkit for blind Facebook users that provide an easy navigation and easy access to the Facebook with the help of JAWS screen reader, the system will be known as Facebook Aggregator.

## 1.8: Layout of project write-up

The dissertation is divided into seven chapters, briefly described as follow:

* Chapter Two: will demonstrate the literature review on traditional tactile for the blind and the problems that blind community is facing. The purpose of this chapter is to highlight the problems with available websites in terms of web accessibility for the blinds.
* Chapter Three: Analysis and Requirements Specification reveals the purpose, goal, scope of the Facebook Aggregator and preliminary investigation. It also clearly specifies the functional and non functional requirements, which identifies the complete specification of requirements for the system development.
* Chapter Four: System Design presents that how the system process flow and the design of the system process for the development of the system.
* Chapter Five: Implementation presents that how the system is being written and developed according to the design that has been developed to lead the implementation.
* Chapter Six: System testing presents the testing strategy and test case to test the Facebook Aggregator and test results of the system.

# CHAPTER TWO LITERATURE REVIEW

In the process of completing of this project, the developers have taken upon to properly research the literature that already existing for the Facebook, Assistive technology and Web Accessibility for the Blind. This is to get a better understanding on the subject before going further into the research and design phase of the development.

As such this paper will detail the sources in which the developer refer to and the knowledge gathered by the process. The main point of this research consists of example, and type of preexisting system that currently in used for the purpose of student information.

## 2.1: Web Accessibility

Web accessibility refers to the practice of making websites usable by people of all abilities and disabilities. When sites are correctly designed, developed and edited, all users can have equal access to information and functionality. For example, when a site is coded with semantically meaningful HTML, with textual equivalents provided for images and with links named meaningfully, this helps blind users using text-to-speech, like JAWS, software and/or text-to-Braille hardware. When text and images are large and/or enlargeable, it is easier for users with poor sight to read and understand the content. When links are underlined (or otherwise differentiated) as well as colored, this ensures that color blind users will be able to notice them. When clickable links and areas are large, this helps users who cannot control a mouse with precision. When pages are coded so that users can navigate by means of the keyboard alone, or a single switch access device alone, this helps users who cannot use a mouse or even a standard keyboard. When videos are closed captioned or a sign language version is available, deaf and hard of hearing users can understand the video. When flashing effects are avoided or made optional, users prone to seizures caused by these effects are not put at risk. And when content is written in plain language and illustrated with instructional diagrams and animations, users with dyslexia and learning difficulties are better able to understand the content. When sites are correctly built and maintained, all of these users can be accommodated while not impacting on the usability of the site for non-disabled users. [6]

### 2.1.1: History of Web Accessibility

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect." -- Tim Berners-Lee, World Wide Web Consortium (W3C) Director and inventor of the World Wide Web said. [7]

The World Wide Web Consortium was created in October 1994 by Tim Berners-Lee, with a mission to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability with a strong emphasis on usability for people with disabilities. The W3C has more than 400 Member organizations from around the world and has earned international recognition for its contributions to the growth of the Web.

The Web Accessibility Initiative (WAI), in coordination with the W3C and organizations around the world, is pursuing accessibility of the Web through five primary areas of work: technology, guidelines, tools, education & outreach, and research & development. The work of the W3C and WAI proceeded on a parallel though not necessarily a collaborative path with the emerging standards of the member countries, which of course include the United States.

Prior to the late 1980's, computers were generally easy for most people with disabilities to use. People who were blind, for example, could use the same word processing software packages as everyone else. Instead of relying on monitors, they used assistive technology -- called "screen readers" -- to read in a synthesized voice all the text and punctuation that a sighted person would read on the computer monitor. All who used early word processors used keyboard commands to interact with the software. To print a document, for instance, one would simultaneously hit the "control" and "P" keys -- something that could be done as easily by blind people as others.

As technology grew more sophisticated, many changes that generally made it easier for nondisabled people to use computers often created barriers for people with disabilities. For instance, software that required someone to issue commands by "pointing and clicking" using a computer mouse became inaccessible to those who could not see icons. Although the solutions were simple and inexpensive, little thought was given to preserving accessibility. For example, if word processing software allows the user to choose between entering "control-P" to print or clicking on a printer icon, then blind people can use the print function as easily as everyone else. [8]

The first web revolution happened on the 14th January 1997: HTML 3.2 was issued from a consortium called the W3C, a consortium not yet known to web developers. As written in the document's introduction, HTML 3.2 is the W3C's specification for HTML, developed in early 1996, together with vendors including IBM, Microsoft, Netscape Communications Corporation, Novell, SoftQuad, Spyglass, and Sun Microsystems. HTML 3.2 added widely deployed features such as tables, applets and text flow around images, while providing full backwards compatibility with the existing standard HTML 2.0. Ten years later, and this is all history, but in those early years were the first web developers: guys and students (especially inside universities, where the web was available before it was available to the general public) started to develop their first pages using built-in web browser editors. [8]

HTML 4.0 was the first HTML version that had notes for accessibility:

As the Web community grows and its members diversify in their abilities and skills, it is crucial that the underlying technologies be appropriate to their specific needs. HTML has been designed to make Web pages more accessible to those with physical limitations. HTML 4.0 developments inspired by concerns for accessibility include:

* Better distinction between document structure and presentation, thus encouraging the use of style sheets instead of HTML presentation elements and attributes.
* Better forms, including the addition of access keys, the ability to group form controls semantically, the ability to group SELECT options semantically, and active labels.
* The ability to markup a text description of an included object (with the OBJECT element).
* A new client-side image map mechanism (the MAP element) that allows authors to integrate image and text links.
* The requirement that alternate text accompanies images included with the IMG element and image maps included with the AREA element.
* Support for the title and lang attributes on all elements.
* Support for the ABBR and ACRONYM elements.
* A wider range of target media (TTY, Braille, etc.) for use with style sheets.
* Better tables, including captions, column groups, and mechanisms to facilitate non-visual rendering.
* Long descriptions of tables, images, frames, etc. [9]

## 2.2: The needs that Web accessibility aims to address include:

* **Visual:** Visual impairments including blindness, various common types of low vision and poor eyesight, various types of color blindness;
* **Motor/Mobility:** e.g. difficulty or inability to use the hands, including tremors, muscle slowness, loss of fine muscle control, etc., due to conditions such as Parkinson's Disease, muscular dystrophy, cerebral palsy, stroke;
* **Auditory:** Deafness or hearing impairments, including individuals who are hard of hearing;
* **Seizures:** Photoepileptic seizures caused by visual strobe or flashing effects.
* **Cognitive/Intellectual:** Developmental disabilities, learning disabilities (dyslexia, dyscalculia, etc.), and cognitive disabilities of various origins, affecting memory, attention, developmental "maturity," problem-solving and logic skills, etc. [10]

## 2.3: Different Disabilities that Can Affects Web Accessibility

Abilities can vary from person to person, and over time, for different people with the same type of disability. People can have combinations of different disabilities, and combinations of varying levels of severity. Following is a list of some disabilities and their relation to accessibility issues on the Web: [11]

* visual disabilities
  + blindness
  + low vision
  + color blindness
* hearing impairments
  + deafness
  + hard of hearing
* cognitive and neurological disabilities
  + dyslexia and dyscalculia
  + attention deficit disorder
  + intellectual disabilities
  + memory impairments
  + mental health disabilities
  + seizure disorders
  + multiple disabilities
  + aging-related conditions

*The Facebook Aggregator targets the people with visual disabilities, specifically blindness and low vision.*

To access the Web, many individuals who are blind rely on screen reader software that reads text on the screen (monitor), JAWS or other screen reader, and outputs this information to a speech synthesizer or refreshable Braille display. Some people who are blind use text-based browsers such as Lynx, or voice browsers, instead of a graphical user interface browser plus screen reader. They use rapid navigation strategies such as tabbing through the headings or links on Web pages rather than reading every word on the page in sequence.

## 2.4: Main barriers that blinds have faced when browsing web

* images that do not have alternative text
* complex images (e.g., graphs or charts) that are not adequately described
* video that is not described in text or audio
* forms that cannot be tabbed through in a logical sequence or that are poorly labeled
* browsers and authoring tools that lack keyboard support for all commands
* browsers and authoring tools that do not use standard applications programmer interfaces for the operating system they are based in
* non-standard document formats that may be difficult for their screen reader to interpret
* flash contents [11]

## 2.5: User with blindness using the web

User uses the Web to find new restaurants to go to with friends and classmates. She has low vision and is deaf. She uses a screen magnifier to enlarge the text on Web sites to a font size that she can read. When screen magnification is not sufficient, she also uses a screen reader to drive a refreshable braille display, which she reads slowly.

At home, User browses local Web sites for new and different restaurants. She uses a personal style sheet with her browser, which makes all Web pages display according to her preferences. Her preferences include having background patterns turned off so that there is enough contrast for her when she uses screen magnification. This is especially helpful when she reads on-line sample menus of appealing restaurants.

A multimedia virtual tour of local entertainment options was recently added to the Web site of the city in which User lives. The tour is captioned and described -- including text subtitles for the audio, and descriptions of the video -- which allows her to access it using a combination of screen magnification and braille. The interface used for the virtual tour is accessible no matter what kind of assistive technology she is using -- screen magnification, her screen reader with refreshable braille, or her portable braille device. The user forwards the Web site address to friends and asks if they are interested in going with her to some of the restaurants featured on the tour.

She also checks the public transportation sites to find local train or bus stops near the restaurants. The Web site for the bus schedule has frames without meaningful titles, and tables without clear column or row headers, so she often gets lost on the site when trying to find the information she needs. The Web site for the local train schedule, however, is easy to use because the frames on that Web site have meaningful titles, and the schedules, which are laid out as long tables with clear row and column headers that she uses to orient herself even when she has magnified the screen display. [11]

## 2.6: Barriers that people with low vision have encountered on the Web

* Web pages with absolute font sizes that do not change (enlarge or reduce) easily
* Web pages that, because of inconsistent layout, are difficult to navigate when enlarged, due to loss of surrounding context [11]
* Web pages, or images on Web pages, that have poor contrast, and whose contrast cannot be easily changed through user override of author style sheets
* text presented as images, which prevents wrapping to the next line when enlarged

## 

## 2.7: List of some well accessible websites

Banks and building societies

* http://www.co-operativebank.co.uk (UK Bank)

Corporations

* http://www.unilever.com/ (goods industries)

Cultural (art, music, literature)

* http://www.onedayfilms.com (cinema)
* http://www.literarymoose.info/ (literature)
* http://www.savethechildren.net (children's charity)

Education and academic institutions

* http://www.manchester.ac.uk/ (UK university)
* http://www.anu.edu.au (AU university)
* http://theatre.msu.edu/ (drama)

Electronic equipment and gadgets:

* http://www.boot.com (UK drugstore)
* http://www.specsavers.com (opticians)

Kids related

* http://www.pearlsforteengirls.com (girls)

News and media:

* http://www.bbc.com (UK media)
* http://www.sky.co.uk (UK media)

Online shopping:

* http://www.amazon.com/text (books)
* http://www.waitrose.com (groceries)

Food and beverage:

* http://www.farehamwinecellar.co.ukl (wine)
* http://www.germanbeerguide.co.uk/ (beer)
* http://www.morrisons.co.uk (supermarket chain

Online tools:

* http://www.wikipedia.org (encyclopedia)
* http://www.google.com (search)

Politics:

* http://europa.eu.int/index\_en.htm (EU)
* http://www.whitehouse.gov (USA)

Private and public transport:

* http://www.opt.dtup.sa.gov.au/ (Public Transport - AU)
* http://www.transportarchive.org.uk/ (Transport History - UK)

Science and nature:

* http://www.iop.org (physics)
* http://www.treesbyapex.com (tree surgeon)

Sports:

* http://www.tankwilliams.com (US football)
* http://www.golf.uk.net/

Information technology related:

* http://www.mozilla.org (browser)
* http://www.opera.com (browser) [12]

## 2.7: Assistive Technologies

Assistive technologies are products used by people with disabilities to help accomplish tasks that they cannot accomplish otherwise or could not do easily otherwise. When used with computers, assistive technologies are sometimes referred to as adaptive software or hardware.

Some assistive technologies are used together with graphical desktop browsers, text browsers, voice browsers, multimedia players, or plug-ins. Some accessibility solutions are built into the operating system, for instance the ability to change the system font size, or configure the operating system so that multiple-keystroke commands can be entered with a sequence of single keystrokes.

Adaptive strategies are techniques that people with disabilities use to assist in using computers or other devices. For example someone who cannot see a Web page may tab through the links on a page as one strategy for helping browse the content. [11]

Following is a list of the assistive technologies and adaptive strategies described below:

### 2.7.1: Alternative keyboards or switches

When typing, holding the hands and wrists in a neutral work posture--where the hands are extended straight without significant bending at the wrist-- is thought to reduce the risk of musculoskeletal problems. Computer users sometimes use awkward or non-neutral work postures when working on the traditional keyboard. [13]

Alternate keyboards or switches are hardware or software devices used by people with physical disabilities, which provide an alternate way of creating keystrokes that appear to come from the standard keyboard. Examples include keyboard with extra-small or extra-large key spacing, key guards that only allow pressing one key at a time, on-screen keyboards. Web-based applications that can be operated entirely from the keyboard, with no mouse required, support a wide range of alternative modes of input. [13]

### 2.7.2: Braille and refreshable Braille

Braille is a system using six to eight raised dots in various patterns to represent letters and numbers that can be read by the fingertips. Braille systems vary greatly around the world. Some "grades" of braille include additional codes beyond standard alpha-numeric characters to represent common letter groupings (e.g., "th," "ble" in Grade II American English braille) in order to make braille more compact. An 8-dot version of braille has been developed to allow all ASCII characters to be represented. Refreshable or dynamic braille involves the use of a mechanical display where dots (pins) can be raised and lowered dynamically to allow any braille characters to be displayed. Refreshable braille displays can be incorporated into portable braille devices with the capabilities of small computers, which can also be used as interfaces to devices such as information kiosks. [11]

### 2.7.3: Scanning software

Scanning software is adaptive software used by individuals with some physical or cognitive disabilities that highlights or announces selection choices (e.g., menu items, links, phrases) one at a time. A user selects a desired item by hitting a switch when the desired item is highlighted or announced. [11]

### 2.7.4: Screen magnifiers

Screen magnification is software used primarily by individuals with low vision that magnifies a portion of the screen for easier viewing. At the same time screen magnifiers make presentations larger, they also reduce the area of the document that may be viewed, removing surrounding context. Some screen magnifiers offer two views of the screen: one magnified and one default size for navigation. [11]

### 2.7.5: Screen readers

Software used by individuals who are blind or who have dyslexia that interprets what is displayed on a screen and directs it either to speech synthesis for audio output, or to refreshable braille for tactile output. Some screen readers use the document tree (i.e., the parsed document code) as their input. Older screen readers make use of the rendered version of a document, so that document order or structure may be lost (e.g., when tables are used for layout) and their output may be confusing. [11]

### 2.7.6: Speech recognition

Speech (or voice) recognition is used by people with some physical disabilities or temporary injuries to hands and forearms as an input method in some voice browsers. Applications that have full keyboard support can be used with speech recognition.

### 2.7.7: Speech synthesis

Speech synthesis or speech output can be generated by screen readers or voice browsers, and involves production of digitized speech from text. People who are used to using speech output sometimes listen to it at very rapid speeds.

### 2.7.8: Tabbing through structural elements

Some accessibility solutions are adaptive strategies rather than specific assistive technologies such as software or hardware. For instance, for people who cannot use a mouse, one strategy for rapidly scanning through links, headers, list items, or other structural items on a Web page is to use the tab key to go through the items in sequence. People who are using screen readers -- whether because they are blind or dyslexic -- may tab through items on a page, as well as people using voice recognition.

### 2.7.9: Text browsers

Text browsers such as Lynx are an alternative to graphical user interface browsers. They can be used with screen readers for people who are blind. They are also used by many people who have low bandwidth connections and do not want to wait for images to download.

### 2.7.10: Visual notification

Visual notification is an alternative feature of some operating systems that allows deaf or hard of hearing users to receive a visual alert of a warning or error message that might otherwise be issued by sound.

### 2.7.11: Voice browsers

Voice browsers are systems which allow voice-driven navigation, some with both voice-input and voice-output, and some allowing telephone-based Web access.

## 2.8: Facebook.com

Facebook is a social networking website launched in February 2004 and operated and privately owned by Facebook, Inc. Users can add people as friends and send them messages, and update their personal profiles to notify friends about themselves. Additionally, users can join networks organized by workplace, school, or college. The website's name stems from the colloquial name of books given to students at the start of the academic year by university administrations in the US with the intention of helping students to get to know each other better. Anyone age 13 or older can become a Facebook user. [14]

Zuckerberg began writing code for a new website in January 2004. He was inspired, he said, by an editorial in The Harvard Crimson about the Facemash incident. "It is clear that the technology needed to create a centralized Website is readily available," the paper observed. "The benefits are many. On February 4, 2004, Zuckerberg launched "Thefacebook", originally located at thefacebook.com "Everyone’s been talking a lot about a universal face book within Harvard," Zuckerberg told The Harvard Crimson. "I think it’s kind of silly that it would take the University a couple of years to get around to it. I can do it better than they can, and I can do it in a week.] "When Mark finished the site, he told a couple of friends. And then one of them suggested putting it on the Kirkland House online mailing list, which was three hundred people," according to roommate Dustin Moskovitz. "And, once they did that, several dozen people joined, and then they were telling people at the other houses. By the end of the night, we were...actively watching the registration process. Within twenty-four hours, we had somewhere between twelve hundred and fifteen hundred registrants."

Membership was initially restricted to students of Harvard College, and within the first month, more than half the undergraduate population at Harvard was registered on the service. Eduardo Saverin (business aspects), Dustin Moskovitz (programmer), Andrew McCollum (graphic artist), and Chris Hughes soon joined Zuckerberg to help promote the website. In March 2004, Facebook expanded to Stanford, Columbia, and Yale. This expansion continued when it opened to all Ivy League and Boston area schools, and gradually most universities in Canada and the United States. Facebook incorporated in the summer of 2004 and the entrepreneur Sean Parker, who had been informally advising Zuckerberg, became the company's president. In June 2004, Facebook moved its base of operations to Palo Alto, California. The company dropped. The from its name after purchasing the domain name facebook.com in 2005 for $200,000. [14]

Facebook launched a high school version in September 2005, which Zuckerberg called the next logical step. At that time, high school networks required an invitation to join. Facebook later expanded membership eligibility to employees of several companies, including Apple Inc. and Microsoft. [15]

The media and users often compare Facebook to MySpace, but one significant difference between the two websites is the level of customization. MySpace allows users to decorate their profiles using HTML and Cascading Style Sheets (CSS), that’s what made MySpace more accessible than Facebook, while Facebook only allows plain text.

### 2.8.1: Web Accessibility and Facebook

According to article “Facebook Commits to Making Social Networking More Accessible for Visually Challenged Users” by Robin Wauters on Apr 7, on www.techcrunch.com Facebook is working together with the American Foundation for the Blind (AFB) to make its social networking service more accessible to users who are blind or visually impaired. In a company blog post, the non-profit organization’s President, Carl Augusto, explains which problems visually challenged users encounter when surfing the web and how they’re able to overcome these issues with the help of website publishers, developers and designers. [16]

This makes sense from Facebook’s perspective too, of course. There are a lot of visually impaired people on the planet, and they want to connect to their friends, relatives.

Facebook's mobile version of the site is HTML only and is compatible with Internet Explorer 7 (and above) and other supported browsers, like Firefox 2 (and above) and Safari 3. [16]

According to The American Foundation for the Blind Answers, “*Cluttered web pages with many links can also complicate usage for a person who is blind. MySpace and Friendster sometimes have more than 100 links on each page that loads—which makes for an overwhelming experience. While sighted users might quickly scan web pages for the most important information, screen reader users generally have to listen to web pages from start to finish, top to bottom, left to right. On sites like MySpace or Friendster, this can mean going through a lot of content before finding the desired link Unlabeled links cause problems for people using a screen reader because although the screen reader will read the link, it sounds like gibberish. AFB evaluators found a few cases of poorly labeled or unlabeled links on the sites, but were pleased to find that most of the main features—messaging, adding friends, browsing, editing, and commenting—are accessible. Because these sites are free, most of them contain several online ads. Ads make it more cumbersome for screen reader users because they have to scroll down the page, and go through the scattered ads, before they can find what they're looking for.”* [17]

Facebook’s mobile version is readable by screen reader but the main issue is still there that there are extra unnecessary links, and some functions are not available, like user cannot post a link or a video. Because Facebook mobile is not meant to be for the blinds so it does not provide the exact Web Accessibility.

## 2.9: The Gap

Making the Facebook more accessible for the blinds is the gap that was filled through this theses and study.

# CHAPTER THREE REQUIREMENT AND ANALYSIS

## 3.1: Requirements Analysis in general

Requirements Analysis is the process of understanding the customer needs and expectations from a proposed system or application and is a well-defined stage in the Software Development Life Cycle. [18]

Requirements are a description of how a system should behave or a description of system properties or attributes. It can alternatively be a statement of ‘what’ an application is expected to do. [18]

Requirements analysis includes three types of activity:

* Eliciting requirements: the task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering.
* Analyzing requirements: determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.
* Recording requirements: Requirements might be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications. [18]

Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. Historically, this has included such things as holding interviews, or holding focus groups (more aptly named in this context as requirements workshops) and creating requirements lists. More modern techniques include prototyping, and use cases. Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced. [18]

In general, requirements are partitioned into functional requirements and non-functional requirements. Functional requirements are associated with specific functions, tasks or behaviors the system must support (e.g. the system must output the data in xml form, or the system must provide the login function). While non-functional requirements are constraints on various attributes of these functions or tasks. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements (e.g. the system must be easily accessible, the quality must follow a particular standard). Because non-functional requirements tend to be stated in terms of constraints on the results of tasks which are given as functional requirements (e.g., constraints on the speed or efficiency of a given task)?

To gather the functional and non-functional requirements, stakeholder identification and stakeholder interviews are the primary steps.

### 3.1.1: Stakeholder identification

In more recent times there has been a focus on identifying who the ‘users’ or ‘customers’ of an application are. Referred to broadly as the ‘stake holders’, these indicate the group or groups of people who will be directly or indirectly impacted by the new application. Stakeholders are persons or organizations which have a valid interest in the system. They may be affected by it either directly or indirectly. A major new emphasis in the 1990s was a focus on the identification of *stakeholders*.

As the stakeholders of Facebook Aggregator, blind community carries the highest amount of people who will directly get affected or interacted with the system. Malaysian Association of Blinds is considered to be stakeholders of the system as well, for requirement gathering the stakeholder interview was conducted from the blinds in the MAB.

### 3.1.2: Stakeholder interviews

Stakeholder interviews are a common technique used in requirement analysis. These interviews may reveal requirements not previously envisioned as being within the scope of the project, and requirements may be contradictory. However, each stakeholder will have an idea of their expectation or will have visualized their requirements. [19]

More than three meetings were organized in order to gather the system requirements. A list of Facebook features (only those features that were provided by the Facebook API service) with short description was explained to the blind community and the questions like following were asked from more than twenty blinds:

* Which Facebook features do you use the most
* Which Facebook features do you interact with the least
* Where do you often get stuck while using Facebook on the web
* Does JAWS reads the contents in sequence
* Does the ads on the Facebook created problem for the JAWS to reads the contents
* How difficult is the navigation from one page to another
* How difficult is to update the status and other accessing the text based features, like email
* Is it easy or difficult to post the comment on the post

Based on questions and answers, we were able to determine that which Facebook feature needs to be added in the first version of the system to prove the concept.

## 3.2: Software Requirements Specification

A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional requirements.

We will discuss the Use case diagram on the next page.

### 3.2.1: USE CASE Diagram

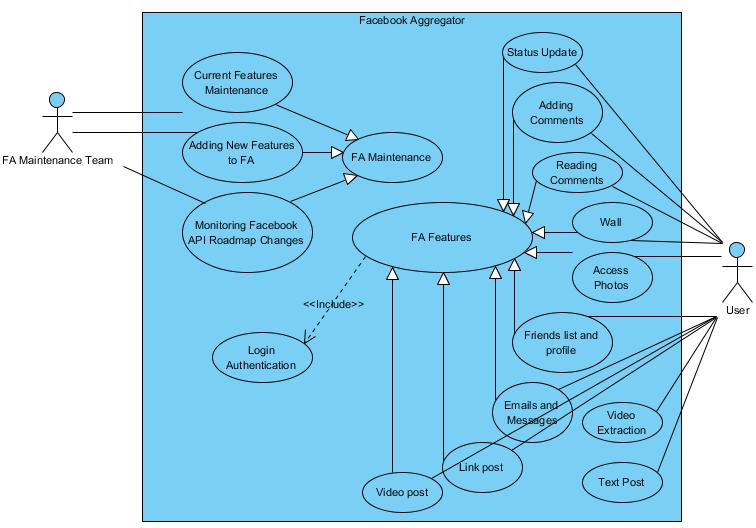
****

Figure 2: Use Case Diagram

Table 1: Actors Grid

|  |  |  |
| --- | --- | --- |
| ID | Name | Related use cases |
| 1 | FA Maintenance Team | Current Features Maintenance, Adding New Features to FA, Monitoring Facebook API Roadmap Changes |
| 2 | User | Status Update, Adding Comments, Reading Comments, Wall, Access Photos, Friends list and profile, Emails and Messages, Link post, Video post |

Table 2: Use Cases Grid

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Primary actors | Supporting actors |
| 1 | FA Maintenance |  |  |
| 2 | Current Features Maintenance | FA Maintenance Team |  |
| 3 | Adding New Features to FA | FA Maintenance Team |  |
| 4 | Monitoring Facebook API Roadmap Changes | FA Maintenance Team |  |
| 5 | FA Features |  |  |
| 6 | Status Update | User |  |
| 7 | Adding Comments | User |  |
| 8 | Reading Comments | User |  |
| 9 | Wall | User |  |
| 10 | Friends list and profile | User |  |
| 11 | Access Photos | User |  |
| 12 | Emails and Messages | User |  |
| 13 | Link post | User |  |
| 14 | Video post | User |  |
| 15 | Login Authentication | User |  |
| 16 | Video extraction | User |  |
| 17 | Text posting | User |  |

Table 3: Use Case Diagram Summary

|  |  |
| --- | --- |
| Name | Documentation |
| Image1.png [FA Maintenance Team](#MA0pDpSGAqACA1AL) | Maintains the Facebook Aggregator website, correct faults (if any), add new features to Facebook Aggregator, in order to improve the performance and other attributes of Facebook Aggregator. |
| Image2.png [Current Features Maintenance](#Ji0JDpSGAqACAzOd) | Currently available contents and features of Facebook Aggregator. e.g. link post, status update etc. |
| Image2.png [FA Maintenance](#sAfxDpSGAqACAyrk) | Facebook maintenance in general. |
| Image2.png [Adding New Features to FA](#V3TJDpSGAqACA0Gc) | Adding new features to the Facebook Aggregator. e.g. feedback form, Facebook online chat etc... |
| Image2.png [Monitoring Facebook API Roadmap Changes](#nd3JDpSGAqACA0Yy) | Keep the eyes on any changes in Facebook APIs made by the Facebook Developer team so the Facebook Aggregator's source code can easily be modified in order to work with the latest API version |
| Image1.png [User](#rQWpDpSGAqACA1CD) | Blind Facebook user who is not comfortable with the Facebook accessibility |
| Image2.png [Status Update](#z6rZDpSGAqACA371) | User's status is plain text that appears on the friends wall. Status could be user's current activity, current location, quote or any other text. |
| Image2.png [FA Features](#iGWZDpSGAqACA3eV) | A feature that has been extracted, added, or adopted from the Facebook to the Facebook Aggregator. i.e. status update, adding comment, wall, reading comment are the examples of the features |
| Image2.png [Adding Comments](#j5nZDpSGAqACA4Cr) | Adding comments on the post or photo. Post could be video, status, or any other activity posted by the friend. |
| Image2.png [Reading Comments](#_GvZDpSGAqACA4Jd) | Reading the comments, through JAWS, that has been made on the friends photos, videos, status etc. |
| Image2.png [Access Photos](#0qu5DpSGAqACA5TJ) | A photo of friends or of the user’s him/her self. |
| Image2.png [Friends list and profile](#eH05DpSGAqACA429) | List of the user's friends and their profile information. |
| Image2.png [Emails and Messages](#E615DpSGAqACA5r1) | Inbox and Sent Facebook emails |
| Image2.png [Link post](#NeT5DpSGAqACA5zR) | Posting of the website link. |
| Image2.png [Video post](#5zCFDpSGAqACA6p9) | Posting of the video link. i.e. youtube video |
| Image2.png [Wall](#AKA5DpSGAqACA4QH) | The home page of the user where all the friend's, group's, page's, activities appears. |
| Image2.png [Login Authentication](#TiPlDpSGAqACAxm3) | Facebook login that is required to use any Facebook feature. |
| Image2.png [Video](#AKA5DpSGAqACA4QH) Extraction | Information (link, description, voice, etc) of the video posted by the friend |
| Image2.png [Text](#TiPlDpSGAqACAxm3) post | Text message that a user posts on friend’s wall. |
| Image3.png [Facebook Aggregator](#ovOhDpSGAqACAyo2) | Facebook Aggregator, a system that extracts the data from Facebook, using Facebook APIs, manipulate the data and present in way that is easily accessible by the blinds. |

### 3.2.2: Requirement Diagram

## C:\Documents and Settings\Bura\Desktop\Requirement Diagram.JPG

Figure 3: Requirement Diagram

### 3.2.3: Operating Environment

In computing, an operating environment is the environment in which users run application software, whether by a command-line interface (such as in MS-DOS or the UNIX shell) or a graphical user interface (such as in the Macintosh operating system or a web browser). Table 4 has briefly shows the operating environment of the Facebook Aggregator. [20]

Table 4: Operating Environment

|  |  |  |
| --- | --- | --- |
| **No** | **Requirement** | **Reason** |
| 1 | JAWS Screen Reader | To read the contents of web page |
| 2 | Internet Explorer | Internet Explorer web browser best work with JAWS |
| 3 | Facebook Developer Toolkit  (API) | To access the functionality of Facebook |
| 4 | Windows XP or Vista | To run JAWS easily |
| 5 | ASP.NET with VB code behind | To develop the system |

## 

### 3.2.4: Constraints and Dependencies

Facebook Aggregator is dependent on both, the Facebook open API and the Facebook Developer Toolkit library. [21] Facebook Aggregator extracts and posts the data to/from Facebook server through Facebook’s open API service using the Facebook Developer Toolkit library [22] and that is the main dependency of the system. However Facebook Aggregator is directly affected by the Facebook Developer Toolkit and indirectly by the Facebook APIs. How? We will see in the below section:

#### 3.2.4.1: Dependency on Facebook APIs

The system if completely dependent on the APIs as the system is using the classes, methods, and functions from the APIs. Any modification or changes made in the API, by the API developer, affect the whole system that is depending on the API. [23] The API developer might shut down the open API service completely or for the specific period of time, means your system also has to be shut down. There are few reasons that force the API developers or the API owner to shut down the service: [24]

* No control: anyone, anywhere, can use the API, and while this may sound like the goal of web services, it drastically limits your response if abuse requests begin pouring in. If those requests are coming from an application on a single machine, it is easy enough to recognize the requests and block them at the firewall. But should an application that behaves poorly reach wide distribution, you will have a very difficult time dealing with the requests.
* No encryption: All requests and responses are visible to anyone between the requesting server and the API server. [25]
* Can't contact developer: Because anyone anywhere can access the API without any prior registration, you are left without any method of directly engaging developers using the API. You may want to contact developers in situations where their application is being abusive, when changes are being made to the API that will affect their application, or to seek suggestions on how to improve the API itself.
* Abuse: Unfortunately today, systems with little or no security or authentication make prime targets for abuse by some less ethical elements out there. Even if you feel that the risk is minimal, you may end up surprised at what others can take advantage of.

In very few cases the API providers shut down the API service but changes in the API are being done almost every day. To keep Facebook Aggregator updated with the latest Facebook API, Developer Roadmaps has to be read every day or atheist three times a week so the downtime can be avoided. In the below table, I will show the example of the changes made in the Facebook API by the Facebook and these changes directly affect the Facebook Aggregator because Facebook Aggregator in completely dependent to the Facebook API. [23]

Table 5: Roadmap changes of Facebook APIs

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Changes** | **Timing** | **Last Updated** |
| Requests | Invite requests (requests with the invite parameter set as "true") will be transitioned to a new notification area. Other types of requests should utilize our improved[Share](http://wiki.developers.facebook.com/index.php/Roadmap_Share) dialogs. | July 2010  [30 days after launching[Share](http://wiki.developers.facebook.com/index.php/Roadmap_Share) enhancements] | *May 13, 2010* |
| Invites | Invites will be transitioned to a new notification area. | June 2010 | *May 13, 2010* |
| Share | Users will be able to share application content with one another via the Inbox as well as the Stream. | June 2010 | *May 13, 2010* |
| Profile | Application tabs will be the only way to integrate into the Profile. | Mid 2010 | *May 3, 2010* |
| Canvas | Formatting on canvas pages will change to better highlight an application's brand. | Mid 2010 | *May 3, 2010* |
| Photos | We're slightly changing a few photo sizes, including the maximum photo size and profile pictures. | Mid 2010 | *May 3, 2010* |
| Enhancements to Data Permissions | We're refreshing privacy settings to give users clear control over how they share data with applications. | June 1, 2010 | *April 30, 2009* |

#### 3.2.4.2: Dependency on Facebook Developer Toolkit library

This toolkit (also known as the Facebook .NET SDK) is provided as a Facebook Client Library similar to Facebook's PHP Client Library or Facebook's JavaScript Client Library. The goal is to enable .NET developers to quickly and easily leverage the various features of the Facebook Platform. Facebook Developer Toolkit is nothing but the bunch to DLL libraries that serves as a third party component between Facebook Aggregator and the Facebook API. [26] In the implementation chapter, I will explain Facebook Developer Toolkit in detail.

As explained in the above section *3.2.4.1* that the Facebook Aggregator is dependent on the Facebook API, similarly the Facebook Developer Toolkit is directly dependent on the Facebook API. I mentioned above, *Facebook Aggregator extracts and post the data to/from Facebook server through Facebook’s open API service using the Facebook Developer Toolkit library;* it means that Facebook Aggregator is not directly affected by the Facebook Developer Toolkit and indirectly by the Facebook APIs because it is dealing with the Facebook APIs using the Facebook Developer Toolkit.

Facebook Aggregator does not have the direct link to the Facebook API if Facebook Developer Toolkit tracks every single change in the Facebook API and modify the Toolkit accordingly in a way that the clients (Facebook Aggregator on any other system that is using FDT) does not have to do too much of modification, minor modification in the client system will still be taking place even if Facebook Developer Toolkit is keep updating on the time, because the system is directly affected by any change made in the Facebook Developer Toolkit.

If worst case scenario is that when Facebook Developer Toolkit developers does not modify and update the toolkit library then the client developers who are using the toolkit will have to program and modify the toolkit DLL files manually to meet the change that has been made in Facebook API, here the client developer (i.e. Facebook Aggregator developer) will have to deal directly with the Facebook APIs. [27]

### 3.2.5: System Features requirement

Facebook features that Facebook Aggregator encompasses. Below section will illustrate the required features of the system. The feature requirements fall under the function requirements because they interact directly with the user. [28]

#### 3.2.5.1: Feature 1: Status update

The "status" feature allows users to inform their friends and the Facebook community of their current whereabouts and actions. A user's most recent status update appears at the top of their profile.

#### 3.2.5.2: Feature 2: Comment posting

Comment that a user can write on the friend’s photo, posted link, posted video, and status.

#### 3.2.5.3: Feature 3: Comment extraction

Feature to read the comment that has been made by the user himself or friends on the post, photo, link, or status. Comment appears at the bottom of the post in the listbox. Each item in the listbox is considered one comment.

#### 3.2.5.4: Feature 4: Wall

The Wall is a space on each user's home page that allows friends to post messages for the user to see while displaying the time and date the message was written. Wall also contains the latest posts made by the friends and the user can comment on the posts.

#### 3.2.5.5: Feature 5: Friends and their profile

Listing friends of the user. From there user will be redirected to the friend’s profile information page pressing enter key. The profile information contains only the basic and important fields like name, sex, country, movies, music etc, because the inclusion of all fields will result the profile page too overloaded. Total fields of user profile information are more than fifty; therefore only less than fifteen basic fields have been added in the Facebook Aggregator.

#### 3.2.5.6: Feature 6: Photo extraction

Information associated to the photo in been added to a listbox. Each item in the listbox serves as one attribute of the picture, first item is the photo index, second item is the photo caption, thirds item contains the photo tags and so on. Listbox shows the information of single picture at once, to show the next picture the listbox clears the items in the list and then reloads the information of the second photo in the album. This information is being read by JAWS.

#### 3.2.5.7: Feature 7: Email and Messages

Facebook email system to allow users send personal and private messages to friends. Inbox and sent messages are being extracted from the Facebook in text form, without HTML tags or any other graphical content.

#### 3.2.5.8: Feature 8: Weblink post

Feature to allow the user to share the links of the website with their friends by simply typing the website address in the text field and pressing enter, the website link will appear on friend’s wall and they can also comment on the link posted by the user.

#### 3.2.5.9: Feature 9: Video post

Feature to allow the user to post the videos, with the text from user, from YouTube, dailymotion, and other video sharing website on their wall, which is accessible by the friends. Videos cannot be placed in categories, whereas photos are sorted by albums.

#### 3.2.5.10: Feature 10: Login authentication

A simple login feature that allows the user signs in to the website.

#### 3.2.5.11: Feature 11: Video extraction

Extracts the information associated with the video posted by the friends or the user and output the information in text form. The information includes the video itself, link of the video, comments made on the video, description of the video, and the text that was added by the poster of the video while posting it.

#### 3.2.5.12: Feature 12: Text posting

The feature to allow the user to write text on the wall of specific friend. The update appears to all the friend’s wall that user A have posted “some text” on user B’s wall.

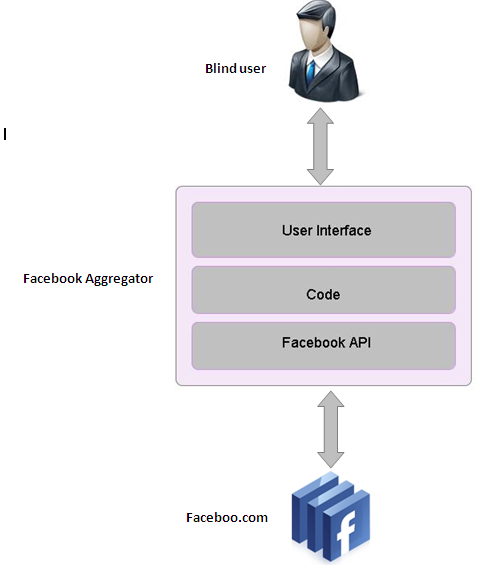
# CHAPTER FOUR DESIGN

## 4.1: Design in Software Engineering

A software design is a meaningful engineering representation of some software product that is to be built. It can be traced to the customer’s requirements and can be assessed for quality against predefined criteria. There are four major areas of concern in design: data, architecture, interfaces, and components. [29]

The design process is very important, because the emphasis in design is on quality, therefore it provides the representation of software that can be assessed for quality. Furthermore, this is the only phase in which the customer’s requirements can be accurately translated into a finished software product or system. Hence, software design serves as the foundation for all software engineering steps that follow regardless of which process model is being employed. [30]

The software specifications will be transformed into designing models, during the design process, to describe the details of the data structures, system architecture, interface and components. Also for quality review will be assessed of the each design product before moving to the next phase of software development. At the end of this process a design specification document is produced, which is composed of the design models that describe the data, architecture, interfaces and components. [31]



**JAWS Screen Reader**

Figure 4: Generic overview of Facebook Aggregator

## 4.3: Flow of the data

****

Figure 5: Level 0 Data Flow Diagram

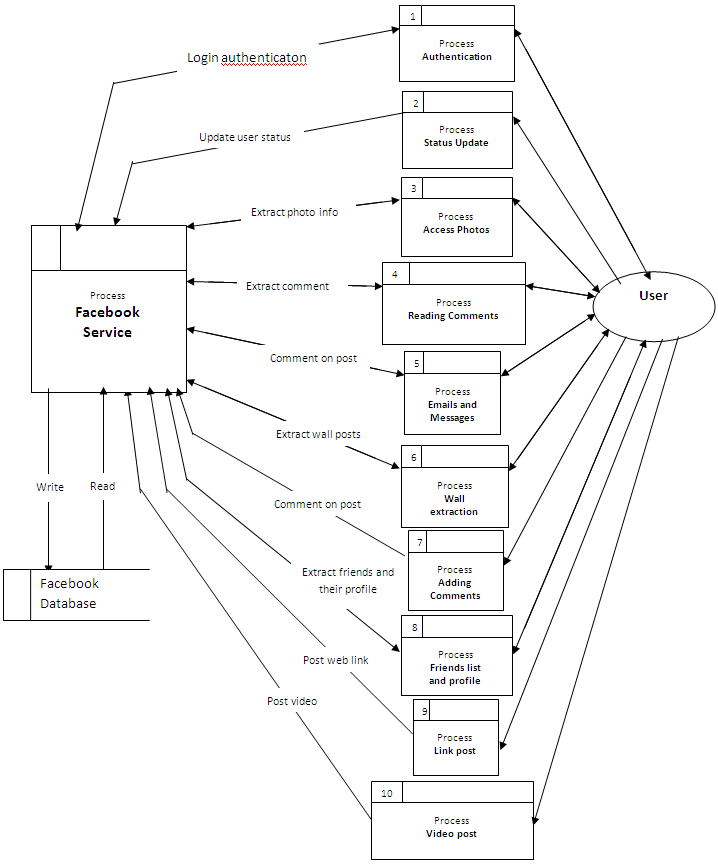
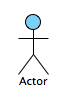


Figure 6: Level 1 Data Flow Diagram

## 4.3: Overall sequence diagram

## Much like the class diagram. In Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.

**Facebook Aggregator**



User

**Front end UI**

**(IE + JAWS)**

**Facebook .NET API**

**Input**

**User request**

**API.Query(FQL statement)**

**Facebook REST server**

**FQL Query request**

**XML data response**

**Code behind**

**XML data**

**Manipulated, Accessible data format**

## 4.3: Website Architecture

Website architecture is an approach to the design and planning of websites which, like architecture itself, involves technical, aesthetic and functional criteria. As in traditional architecture, the focus is properly on the user and on user requirements. This requires particular attention to web content, a business plan, usability, interaction design, and information architecture and web design.

"Website architecture" has the potential to be a term used for the intellectual discipline of organizing website content. "Web design", by way of contrast, describes the practical tasks, part-graphic and part-technical, of designing and publishing a website.

Before I got started with anything, I had to keep in mind few design guidelines to lead the project design. The design of the website should utilize the maximum functionality of the JAWS screen reader in order to make web well accessible. [32]

#### I ensure the following so JAWS screen reader can maximally utilized.

* Ensure that there is adequate contrast between the page background   
  and foreground.
* Ensure that color alone is not used to convey information.
* Ensure that the language is specified for the screen reader software   
  and that any changes in langue are clearly identified.
* Ensure that images have a text only version if the image is complex   
  enough to warrant a description. If the image is not complex,   
  we ensure that there is a simple alternate description of the image
* Ensure that every lines of text make sense when read out aloud,   
  by listening to all text on all pages.
* Test each page with applicable software tools for handicap access   
  such as A-Prompts, Bobby or other applicable test tool.
* Guarantee that each page is fully accessible.
* Ensure proper usage of punctuation at end of sentences or lines of text.
* Ensure that there are no large blocks of text.
* Ensure that there are no "long" lines of text, to minimize the need   
  for excessive eye or head movement.
* Form fields are labeled and easy to navigate.
* Ensure that when tables are used, they are summarized; cells are properly   
  labeled and readable when linearized.

As Java Script is largely not accessible for text to speech software or screen readers, I avoid the use of Java design to “present any kind of information”.

### 4.3.1: Architecture Model

There are quite a few website architecture models that can be adopted, depends on the type of website you are developing. Following are some of the architecture models; I will briefly explain three models and then the model that has been used to develop Facebook Aggregator

Website Architecture Models:

#### 4.3.1.1: All-in-one model



Figure 7: All-in-one Architecture

*http://www.webdesignfromscratch.com/snippets/ia\_diagram\_allinone.gif*

This is the simplest possible model. Everything goes on a single Home page.

#### 4.3.1.2: Flat model



Figure 8: Flat Architecture Model

http://www.webdesignfromscratch.com/snippets/ia\_diagram\_monocline2.gif

A flat pattern is where all pages are arranged as peers, and everyone is accessible from every other one. This is very common for simple sites, where there are a few standard topics, such as: Home, About Us, Contact Us, Products. [32]

*I will be using flat model for Facebook Aggregator so the accessing and navigation can be simpler and easier*

#### 4.3.1.3: Hub-and-spoke / Daisy model

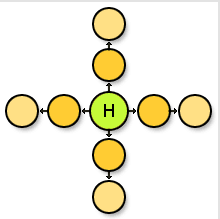


Figure 9: Hub-and-spoke (or daisy) pattern

http://www.webdesignfromscratch.com/snippets/ia\_diagram\_hub\_and\_spoke.gif

This model is useful for multiple, distinct linear workflows. A good example may be an email application, where you will return to your inbox at several points, e.g. after reading a message, after sending a message, or after adding a new contact.

## 4.4: Navigation Design

Navigation design is the design of moving from one page, content, or area of the website to another. It arranges that how the user will be navigating from one page to another more easily. As of Facebook Aggregator is meant to be for blinds, no Graphical User Interface is required for the navigation but the user will navigate using the keyboard TAB key and Arrow keys.

### 4.4.1: Horizontal top bar

Horizontal navigation bars are very neat. Because top-down flow is slightly better to left-right flow, it’s natural for high-level navigation to sit above content.

The obvious problem is when a top bar gets wider than the page. It’s normal for web pages to extend vertically, and users are used to scrolling vertically. If you use text-based navigation, and it’s resizable, your navigation could get bigger or smaller depending on the user’s browser settings. [32]

Horizontal bars are therefore appropriate where the number of items is known in advance, there are not going to be any more items added, and there is enough width to accommodate all items safely in the target screen resolution.

*Horizontal top bars are used in Facebook Aggregator but without drop-down menus or sub menus within menu as the navigation bar is text based.*

****

Figure 10: Facebook Aggregator horizontal top bar

### 4.4.2: Paging

Paging is a mechanic that will be familiar to all web users. This is where you get a piece of content that spans several pages. You are given standard tools that let you navigate back, forwards, or jump directly to specific pages. In Facebook Aggregator the list of 100 results is shown upon the request and there are more results for the requested query, first 100 results are shown in the listbox control and, like paging, a button of “Next 100 friends” is placed on the page to show the next 100 results, as in paging the page navigates to the next page in order to show rest of the results but here the story is different. When “Next 100 friends” button is pressed, the system does not navigate to the other page but postback the same page with new result items in the listbox control. [32]



Figure 11: Facebook Aggregator paging interface

*So the design of the Facebook Aggregator is very simple made in order to achieve the goal which is to make the website “easily” accessible.*

# CHAPTER FIVE IMPLEMENTATION

## 5.1: Implementation in Software Engineering

In software implementation stage, we build the components of the system, or we can say that implementation is the stage of start writing the system or start doing programming, based on the given architecture documentation from the design phase and the requirement document from the analysis phase. Hence, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For instance, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability guideline. [33] Therefore, the architecture document should give guidance, sometimes, which is found in the requirement document. This phase deals with issues of programming, quality, performance, libraries, and debugging. The end deliverable is the product itself and the source code of the system.

## 5.2: Before writing the Facebook Aggregator

An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system. Many implementations exist for a given specification or standard, as mentioned above. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages. [34]

The implementation stage is a stage where you have to be very clear about each and every requirement of the customer because software implementation is a collaborative effort between the software vendor and the customer. Secondly, clear and open communication is essential. Of course, customer needs to communicate their objectives to the software vendor. But even more important, they must listen to what the software vendor tells you. The biggest sources of failure are the misunderstandings that develop between what the customer expects and what the software vendor can deliver. Be on your guard and learn the software's capabilities and limitations. The software vendor may not volunteer the product's failings unless customer does enough inquiring and it is possible that vendor’s power of influence may convince customer to purchase a product that may not fully address the needs.

In Facebook Aggregator’s case, we listed down quite a few functionalities of the Facebook, the social networking website, and demonstrated them to the blinds in Malaysian Association of Blinds (MAB) and asked them to select the functionalities that are mostly used by the blinds. After they refined the list of the featured of the Facebook (Photos, Videos, Status updating, Emailing etc), the list was given back to us and the work had been started with respect of the requirements of the Blinds. Off course we added few other things that we found can also be beneficial for the Blinds.

## 5.3: Selecting the right tools, programming language, technologies, and IDEs

The next and important step is to select the right tools, programming language, technologies and Integrated Development Environment. Below I will explain briefly that **what** and **why** I have used to implement the system and I will also be telling **what I** have **not** used and **why not** used. [35]

### 5.3.1: APIs and Toolkits

There were two methods available, *when the system was built*, to use the Facebook APIs to extract the functionalities of the Facebook and integrate, manipulate, and organize them to form Facebook Aggregator. First method was to use the API that uses a REST-like interface. This means that the Facebook method calls are made over the internet by sending HTTP GET or POST requests to the Facebook API REST server (http://api.facebook.com/restserver.php). Nearly any computer language can be used to communicate over HTTP with the REST server. [36]

The second method was to use the SDK library (Java SDK or Microsoft SDK, depends on the programming language that you use). Clarity Consulting Inc developed the original Facebook Developer Toolkit for the Microsoft Visual Studio Express Team. This method was used to develop the Facebook Aggregator. [36]

After the Facebook Aggregator was developed, right after that I visited the Facebook developer’s page for the REST APIs and I saw the Facebook said that “*The REST API is the previous version of the Graph API. It enables you to interact with Facebook web site programmatically via simple HTTP requests. If you are new to the Facebook Platform, we recommend you use our new Graph API instead.”* [36]

Facebook had introduces new method called Graph API to access the Facebook server. In the future Facebook Aggregator might be upgraded to the new APIs method that has just been released by the Facebook. *Sometime upgrading can lead to system crash.*

### 5.3.2: Adoption of Facebook Developer Toolkit

This toolkit (also known as the Facebook .NET SDK) is provided as a Facebook Client Library similar to Facebook's PHP Client Library or Facebook's JavaScript Client Library. The goal is to enable .NET developers to quickly and easily leverage the various features of the Facebook Platform. This toolkit has evolved over time with input from the community and from Microsoft. The latest release (v3.0) cleans up some architecture inconsistencies and provides an asynchronous interface to improve using the toolkit from Silverlight and from WPF.

The main entry point is the API (Facebook.Rest.Api) class in the Facebook.dll assembly. This class wraps the Facebook REST API and provides an easy to use interface for calling the different methods currently available in the Facebook API. The toolkit also provides samples and tools for helping develop Facebook applications in the various .NET platforms including: Asp.NET, Silverlight, WPF and Winforms. Additionally, we’ve provided all the source code for the API, components, controls and samples for you to explore. [37]

Facebook Developer Toolkit was adopted for the Facebook Aggregator because it was the only Facebook development kit that was built specifically for .NET platform; I am personally more comfortable with Visual Basic programming syntax and that’s what brought me to select Facebook Development Toolkit. To use the REST API, the request has to send and get using httpwebrequest and httpwebresponce and to call any function using REST API, a lot of code has to be written and that looked quite long and difficult as compare to using the SDK toolkit .After making the choice between selecting the REST APIs and Facebook Developer Toolkit, the second step was to select the right method from the available API methods/approaches to access, read, and write the data to Facebook. [36]

There are more than three different methods and approaches to do the same task, i.e. retrieving photo, updating status etc. The main API methods to read and write data were:

### 5.3.3: Using FQL (Facebook Query Language)

Facebook Query Language, or FQL, allows you to use a SQL-style interface to more easily query the same Facebook social data that you can access through other Facebook API methods. As I have used FQL to read and write data from/to Facebook Aggregator to Facebook, I will be explaining FQL in detail later in this chapter. [38]

### 5.3.4: Using FBML / XFBML

Facebook Markup Language (FBML) enables you to build full Facebook Platform applications that deeply integrate into a user's Facebook experience. You can hook into several Facebook integration points, including the profile, profile actions, Facebook canvas, News Feed and Mini-Feed. FBML is an evolved subset of HTML with some elements removed, and others which have been added that are specific to Facebook. FBML syntax is quite similar to HTML as they share the same “tag” pattern. Below is the example of FBML tag to extract the wall post, could be a text from the friend, video link from a friend or any valid post. [36]

<fb:wallpost uid="10" t="1180502413">

This tag exists outside <fb:wall> tags.

<br />

It has the t attribute set to 1180502413

</fb:wallpost>

<fb:wall>

<fb:wallpost uid="11">

This tag exists inside <fb:wall> tags.

It has no t attribute set

</fb:wallpost>

</fb:wall>: *The above FBML tag results the below*



Figure 12: Wall post FBML tag

http://img211.imageshack.us/img211/2830/fbwallpostkx5.jpg

What happened actually is that the FBML tag has retrieved the user control and data , of parameters specified in the FBML tags, from the Facebook server, over the internet, and added the control to the web page. In FQL or other methods, you have to extract the data manually from the Facebook server and then display it whichever way you want you. There are many FBML tags/controls such as dialog boxes (like msgboxes), comment boxes (interface to add comment), wall post (above example).

Calling methods of Facebook Developer Toolkit and let the toolkit do the rest. Meaning to say that each and every method to read and write the data has to be called from the API SDK, i.e. to retrieve the friend list, I have to call the method friends.getList, put the returned into a variable and then execute it further.

### 5.3.5: FQL, Facebook Toolkit, and ASP.net

*FQL + Facebook Developer Toolkit + ASP .NET = Facebook Aggregator*

ASP.NET (Visual Studio 2008) programming platform is used to develop Facebook Aggregator.

Small data were extracted from the Facebook by calling Facebook Development Toolkit methods and functions, i.e. extracting the name of one, or two friends. But when it comes to extract the data, which needs repetitions and loops to get the final result, FQL was the only best way to do that. It worked this way, Facebook Development Toolkit has a method that sends the FQL query to the Facebook server and returns back the result in XML format and then we can manipulate the XML text whichever way we want.

Dim query = String.Format("SELECT uid, name FROM user WHERE uid = ‘55678’”)

Dim result = api.Fql.Query(query)

The above FQL example extracts the name of the user who has the ID ‘55678’ and saved the result in the declared variable. Each Facebook user has a unique ID that worked as a primary key to extract the information of the user.

# CHAPTER SIX TESTING

## 6.1: Testing in Software Engineering

There are two basic classes of software testing, black box testing and white box testing. General testing process for large system development starts with the testing of individual program units such as functions, classes or objects. These are then integrated into sub-system and systems, and the interactions of these units were tested. Finally after delivery of the system, the customer may carry out a series of acceptance tests to check that the system performs as specified. [39]

Whereas, for smaller system or for system that are developed through scripting or reuse, there are often fewer distinct stages in the process.

The two fundamental testing activities are component testing, testing the parts of the system – and system testing, testing the system as a whole. [39]

## 6.2: Goals and Types of Testing

Basically, there are two distinct goals of the software testing process:

* To demonstrate to the developer and the customer that the software meets its requirements.
* To discover faults or defects in the software where the behavior of the software is incorrect, undesirable or does not conform to its specification.

The first goal, where you expect the system to perform correctly using a given set of test cases that reflect the systems expected use, leads to validation testing. The second goal leads to defect testing, where the test cases are designed to expose defects. The main types of testing approaches are defined below: [39]

### 6.2.1: System Testing

System testing involves integrating two or more components that implement system functions or features and then testing this integrated system. For most complex systems, there are two distinct phases to system testing – Integration Testing and Release Testing. As for the Facebook Aggregator, the system had to go through Integration and Release testing both.

#### 6.2.1.1: Integration Testing

Integration testing is mostly concerned with finding defects in the system, where the test team has access to the source code of the system. If the problem is discovered, the team goes through the source code to find the components that have to be debugged. [39]

Integration testing was done after every unit or feature being added to the system. For example, if the current has three features (adding comments, reading wall, posting link), when the fourth feature, unit, or component if being attached or added to the system, integration testing had to be done throughout although there are still many units or components to be added to the system.

#### 5.2.1.2: Release Testing

In release testing that version of the system is tested that could be released to users or customers. The test team here validates if the system meets its requirements and also ensures system dependability. It is usually black-box testing where the test team is simply concerned with demonstrating the system does or does not work properly. If problems discovered are then reported to the development team whose job is to debug the program. Acceptance Testing is key aspect of release testing, where the customers are involved in release testing. If the release is good enough, the customer may then accept it for use.

After all the units were integrated and combined together to form a complete system, a release test was ran to make sure that system’s components are not affecting the other components after integrating them. Few feedbacks from the users were collected, according to the feedback the system has to go through the debugging and tested again. [39]

### 6.2.2: Component Testing

Also known as Unit Testing is the process of testing individual components in the system, to expose faults in these components, and the software developers are responsible for this testing. There are different types of component that may be tested at this stage:

1. Individual functions or methods within an object
2. Object classes that have several attributes and methods
3. Composite components made up of several different objects or function. These composite components have a defined interface that is used t access their functionality.

Facebook Aggregator’s was created in components and units. A new ASP.NET project was created for every unit and each project was put in to the separate folder. Below is the screenshot of each unit.

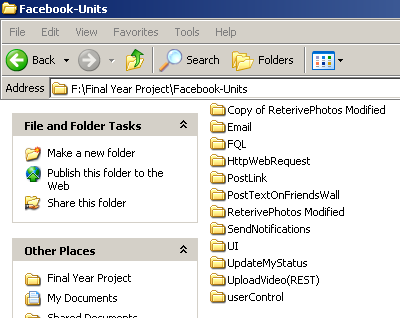


Figure 13: Facebook Aggregator unit folder

Above are the units of the Facebook Aggregator which later were integrated as a whole system and were tested using the system testing method.

### 5.2.3: Interface Testing

One of the most important tests carried out during any Quality Assurance process is Interface Testing, because human being interacts with hardware through an interface. Therefore, to validate the functionality and the accuracy of the interface to the written specifications, interfaces testing are to be carried out. Interface testing is very important because this ensures that customers will not face any problem in using the software once the software is commercially available for them.

But the case here is different; Facebook Aggregator does not deal with the fancy or good looking Graphical User Interface as it is meant to be for blinds. Interface of the Facebook Aggregator deals with making the contents of the website easily accessible. The TAB indexes for the links and buttons had to be precisely defined because with the blind press TAB to navigate through the page, the TABING upon TAB key pressed must be in sequence.

Table 6: Five levels of testing



## 

## 6.3: Black box testing

Black box testing, also called functional testing and behavioral testing, focuses on determining whether or not a program does what it is supposed to do based on its functional requirements. Black box testing attempts to find errors in the external behavior of the code in the following categories (1) incorrect or missing functionality; (2) interface errors; (3) errors in data structures used by interfaces; (4) behavior or performance errors; and (5) initialization and termination errors. Through this testing, we can determine if the functions appear to work according to specifications. However, it is important to note that no amount of testing can unequivocally demonstrate the absence of errors and defects the code.



Figure 14: Black Box testing

http://2.bp.blogspot.com/\_fOOSCGT3XIw/Sa4cVsX7T7I/AAAAAAAAAAM/i\_YOWD6xHIQ/s320/BlackBoxTesting.gif

## 6.4: Test Cases

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not.

The format of the test case design is very important. I will use a particular format for our test cases, as shown in Table

Table 7: Test Case Planning Format



Table 8: Test cases for Facebook Aggregator

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Description | Expected Results | Actual Results |
| 1 | User pressed the login button on the home page | Facebook login page must appear | Facebook login page appeared.  **Test case passed** |
| 2 | User entered the username and password. Test case 1 has successfully completed | User should be redirected to the home page, from where he/she started (Test case 1) | User redirected back to the home page. **Test case passed** |
| 3 | User pressed the “I have signed in, take me to the home page” link. | User must be redirected to the user’s home page, also called Wall, and get the friend’s posts in a listbox. | User’s home page appeared with the friend’s, liked pages’, and groups’ posts in a listbox (Wall).  **Test case passed** |
| 4 | User has selected YouTube video post item, (using tab and arrow keys from the keyboard) from the wall’s listbox and pressed enter | User will be redirected to the post page where the video will be played and the information, links of the video will appear in the form of text and hyperlinks | Page with video and the associated information is appeared.  **Test case passed** |
| 5 | User navigated to the “Friends” webpage by pressing on the “Friends” link from the home page. | List of first hundred,  0 – 100, friends will appear on the “Friends” page | List of hundred friends appeared in less than four seconds time.  **Test case passed** |
| 6 | After the completion of test case 5, user clicked on the “Next 100 friends” link on the “Friends” page | List will reload the next 100 friends upon the page postback | List has successfully reloaded with 100 friends, range from 100 – 200.  **Test case passed** |
| 7 | User has continuously pressed the “Next 100 friends” button to the point where there is no more friends to be shown in the list, or a user does not have a friend added at all and he navigated to the “Friends” page | List of friends will not add any item in it and a label above the list will notify the user that he/she has “0” friends to be added in the list | List appears empty with the label “0 friends”, and JAWS read that correctly.  **Test case passed** |
| 8 | User has selected an item from the list of 100 friends and pressed enter. | Three link buttons will appear on the right side of the friends list, link to the profile of the selected user, link to the photo albums of the selected user, link to post the text on the selected friend’s wall | Links appeared in less than one second.  **Test case passed** |
| 9 | User has navigated to the to the “Photos” web page, selected an album from his/her own albums using keyboard keys, and pressed enter. | The information (caption, index, tags, comments etc) of the very first photo in the album will be added in the photos listbox as items, photo index as item 0, photo caption as item 1, photo link as item 2, and so on. Pressing the “next” button is required to go to the next photo. | The information  associated to the photo has been added to the photos listbox successfully  **Test case passed** |
| 10 | User has navigated to the “Inbox” page to check the inbox send emails. | List of twenty inbox emails will appear at first. List of sent emails will be added upon the press of “Next” button | “User does not have the permission to this feature” error occurred.  **Test case failed** |

Ten test cases were developed to accomplish the Black Box testing for the Facebook Aggregator.

The last test case in the above table has failed for a reason. In order to use the “email” feature of the Facebook, through the APIs, the application (Facebook Aggregator) owner has to send a letter to the Facebook Inc to whitelist the application and the whitelisting process takes up to 2 months.

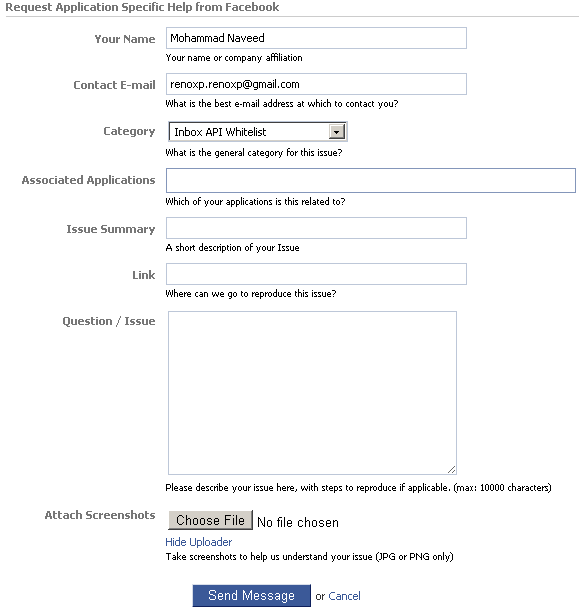


Figure 15: Request Application Specific Help from Facebook form

http://www.facebook.com/developers/developer\_help.php?category=Inbox

*“You can set global permissions for your social widget. Decide whether you want a blacklist or a whitelist for viewing content. In blacklist mode, any users you specify cannot see the content of the social widget. In whitelist mode, only the users you specify can see the content of the social widget”* Facebook Developers Wiki.

# CHAPTER SEVEN FUTURE WORK/MAINTENANCE

As for the future work on the Facebook Aggregator, few tasks, ideas, features, and functionalities are queued up to be added in the system. Not all the Facebook features have been added to the Facebook Aggregator but will be added in the future to the Facebook Aggregator.

## 7.1: Recommendations and Features

Not all the Facebook features has been added to the Facebook Aggregator as the system is the proof of concept and does not encompass the full features. To develop the Facebook Aggregator with the full features of the Facebook need more time and developing team.

The features that are going to be added to the Facebook Aggregator in the future are the following:

### 7.1.1: Photo and video uploads

Users can add their videos and photos with the service by uploading video and photos, from the local computer and/or using a webcam recording feature. Additionally, users can "tag" their friends in the video. Videos cannot be placed in categories, whereas photos are sorted by albums.

### 7.1.2: Notes

Facebook Notes was introduced on 22 August 2006, a blogging feature that allowed tags and embeddable images. Users were later able to import blogs from Xanga, LiveJournal, Blogger, and other blogging services. Facebook aggregator has the function to notify the user, on user’s wall, when any of the friends has published the note. It notifies the note title, published date/time, but the note reading, writing, and publishing features will fall under the future work.

### 7.1.3: Marketplace

Facebook Marketplace allowing users to post and search free classified ads within the following categories: For Sale, Housing, Jobs, and Others. Ads can be posted in either available or wanted format. The market place is available for all Facebook users and is currently free.

### 7.1.4: Events

Facebook events are a way to let friends know about upcoming events in their community and to organize social gatherings. Events require an event name, network, host name, event type, start and end time, location, and a guest list of friends invited. Events can be open, closed, or secret. When setting up an event the user can choose to allow friends to upload photos or videos to the event.

### 7.1.5: Groups

Groups are used for discussions and events etc. Groups are a way of enabling a number of people to come together online to share information and discuss specific subjects. They are increasingly used by clubs, companies and public sector organizations to engage with stakeholders, employees, members, service users, shareholders or customers.

### 7.1.6: Chat

Users are only able to chat with their Facebook friends and on a one-to-one basis, although a user may chat with multiple friends simultaneously through separate chat interfaces.

### 7.1.6: Notifications

Notifications of the more important events, e.g. someone sharing a link on the user's wall or commenting on a post the user previously commented on, someone commented on the user’s photo or video etc, keeping the user up to speed with events as they are occurring.

### 7.1.7: Feedbacks and Recommendations

A feedback form will be added to the Facebook Aggregator to allow visitors to give the feedbacks, opinions, and suggestions on the website by filling in a form online. This kind of suggestions and feedbacks from the users helps a lot in the post-release testing.

It must be understood that all testing measurements involves sampling. A sample is a small division of a large and usually indefinable population or people. During development, the user sample sizes available to most design teams are extremely small: usually on the order of 3 to 10 users. Similarly, testing procedures will usually focus on a small number of user tasks -- presumably those adjudged to be the most critical ones. Typically, a testing session will last up to 60 minutes and test the execution of 4 to 5 tasks, whereas the product will typically support a large and possibly indefinable number of tasks.

However, there is usually no way of being able to test with large samples until the product is released.

Facebook Aggregator web site will be launched in a Beta version state to a limited but usually sizeable sample of users, and testing can be carried out on that basis, to achieve a well-tested final version.

There is a great need to continue testing and measurement after the product has been released, to see the true picture of how the product is performing in relation to the usability goals set for it. Data from such an activity will be used in a number of ways. It can:

* Give advance warning of where user support will be most needed;
* Indicate what are the good sales points of the product;
* Prioritize bug fixes and improvements;
* Be used as benchmarks for future releases;
* Feed into the requirements specification of the next version.

# References

|  |  |
| --- | --- |
| [1] | S. L. Henry, "Introduction to Web Accessibility," WC3, September 2005. [Online]. Available: http://www.w3.org/WAI/intro/accessibility. [Accessed June 2010]. |
| [2] | S. L. Henry, "Essential Components of Web Accessibility," W3C, August 2005. [Online]. Available: http://www.w3.org/WAI/intro/components.php. [Accessed July 2010]. |
| [3] | P. V. M. N. a. J. H. Jos Vander Sloten, Assistive Technologies: New Challenges for Education, Antwerp: Springer Berlin Heidelberg, 2008. |
| [4] | "Freedom Scientific - JAWS for Windows Screen Reading Software," Freedom Scientific, [Online]. Available: http://www.freedomscientific.com/products/fs/jaws-product-page.asp. |
| [5] | E. Heald, "Google News and newspaper publishers: allies or enemies?," World Association of Newspapers, 11 March 2009. [Online]. Available: http://www.editorsweblog.org/analysis/2009/03/google\_news\_and\_newspaper\_publishers\_all.php. [Accessed 20 June 2010]. |
| [6] | "Web Accessibility - George Mason University," George Mason University, [Online]. Available: http://accessibility.gmu.edu/webaccessibility/. [Accessed July 2010]. |
| [7] | "World Wide Web Access: Disability Discrimination Act Advisory Notes," [Online]. Available: http://www.hreoc.gov.au/disability\_rights/standards/www\_3/www\_3.html. [Accessed July 2010]. |
| [8] | M. VanRanden, "A BRIEF HISTORY OF WEB ACCESSIBILITY INTIATIVES AND LAWS," August 2005. [Online]. Available: http://www.suite101.com/article.cfm/recovery\_after\_amputation/46713. [Accessed March 2010]. |
| [9] | W3C, "Introduction to HTML 4," W3C, [Online]. Available: http://www.w3.org/TR/REC-html40/intro/intro.html. [Accessed January 2010]. |
| [10] | "Accessibility - Community Mozilla," Mozilla, [Online]. Available: http://guides.mozilla.org/Accessibility. [Accessed March 2010]. |
| [11] | J. Brewer, "How People with Disabilities Use the Web," W3C, May 2005. [Online]. Available: http://www.w3.org/WAI/EO/Drafts/PWD-Use-Web/. [Accessed March 2010]. |
| [12] | hgilbert, "List of Best Accessible Websites - SitePoint Forums," September 2006. [Online]. Available: http://www.sitepoint.com/forums/showthread.php?t=218076. [Accessed March 2010]. |
| [13] | U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES, "http://www.cdc.gov/niosh/pdfs/97-148.pdf," [Online]. Available: http://www.cdc.gov/niosh/pdfs/97-148.pdf. [Accessed March 2010]. |
| [14] | Wikipedia, "Facebook - Wikipedia, the free encyclopedia," Wikipedia, 2010. [Online]. Available: http://en.wikipedia.org/wiki/Facebook. [Accessed March 2010]. |
| [15] | 2010. [Online]. Available: http://www.openwiki.com/ow.asp?FacebookNews. [Accessed March 2010]. |
| [16] | R. Wauters, "Facebook Commits To Making Social Networking More Accessible For Visually Challenged Users," Techcrunch, April 2009. [Online]. Available: http://techcrunch.com/2009/04/07/facebook-commits-to-making-social-networking-more-accessible-for-visually-challenged-users/. [Accessed April 2010]. |
| [17] | A. F. f. t. Blind, "Are Social Networking Sites Accessible to People with Vision Loss? - American Foundation for the Blind," American Foundation for the Blind, [Online]. Available: http://www.afb.org/Section.asp?SectionID=57&TopicID=167&DocumentID=3153. [Accessed June 2010]. |
| [18] | Outsource2india, "Requirements Analysis: Process of requirements gathering and requirement definition," Outsource2india, [Online]. Available: http://www.outsource2india.com/software/RequirementAnalysis.asp. [Accessed March 2010]. |
| [19] | Wikipedia, 2010. [Online]. Available: http://en.wikipedia.org/wiki/Requirements\_analysis. [Accessed March 2010]. |
| [20] | M. BROWN, "Collaborative Production Management in the process industries.," MATRIKON INC, November 004. [Online]. Available: http://www.thefreelibrary.com/Collaborative+Production+Management+in+the+process+industries.-a0148008394. [Accessed March 2010]. |
| [21] | B. Konrad, "How to Retrieve User Data from Facebook Connect in ASP.NET - Devtacular," Devtacular, 2009. [Online]. Available: http://devtacular.com/articles/bkonrad/how-to-retrieve-user-data-from-facebook-connect-in-aspnet/. [Accessed January 2010]. |
| [22] | K. Wong, "COMPUTER BEACON | ASP.NET FBML and Facebook Connect Application Tutorial," July 2009. [Online]. Available: http://fbtutorial.computerbeacon.net/section2/step1.aspx. [Accessed January 2010]. |
| [23] | Facebook, "Developer Roadmap - Facebook Developer Wik," Facebook, 2010. [Online]. Available: http://wiki.developers.facebook.com/index.php/Developer\_Roadmap. [Accessed December 2009]. |
| [24] | g. delgado, "Advantages and disadvantages of open api," free article, July 2006. [Online]. Available: http://e-articles.info/e/a/title/Advantages-and-Disadvantages-of-Open-API/. [Accessed March 2010]. |
| [25] | Facebook, "Authentication - Facebook," Facebook, 2010. [Online]. Available: http://developers.facebook.com/docs/authentication/. [Accessed March 2010]. |
| [26] | Microsoft, "Facebook SDK Overview," Microsoft, [Online]. Available: http://msdn.microsoft.com/en-us/windows/ee388574.aspx. [Accessed March 2010]. |
| [27] | D. Orenstein, "Application Programming Interface," Computer World, January 2000. [Online]. Available: http://www.computerworld.com/s/article/43487/Application\_Programming\_Interface. [Accessed March 2010]. |
| [28] | G. R. Allen, "System Features," striweb, December 2002. [Online]. Available: http://striweb.si.edu/drr/book/SFTEPprospectus.pdf. [Accessed March 2010]. |
| [29] | Adrian Als & Charles Greenidge, "Design - Concepts and Principles," The University of the West Indies, October 2004. [Online]. Available: http://www.cavehill.uwi.edu/staff/eportfolios/paulwalcott/courses/comp2145/2009/design\_-\_concepts\_and\_principles.htm. [Accessed March 2010]. |
| [30] | Q. Zhang, "International Journal of Quality & Reliability Management," vol. 18, 2001. |
| [31] | B. Kitchenham, "SOFTWARE QUALITY: THE ELUSIVE TARGET," 1996. |
| [32] | Scratchmedia Limited, "Information Architecture models: Guide to web site navigation patterns," Scratchmedia Limited, 2009. [Online]. Available: http://www.webdesignfromscratch.com/website-architecture/ia-models/. [Accessed 2010]. |
| [33] | J. Donn Le Vie, "Writing Software Requirements Specifications," December 2009. [Online]. Available: http://www.techwr-l.com/techwhirl/magazine/writing/softwarerequirementspecs.html. [Accessed March 2010]. |
| [34] | G. Dubinovskiy, "Facebook Application Development - .NET Software Development - Generate Facebook Content using FBML (part 1," 2009. |
| [35] | A. L. Tharp, "Selecting the “right” programming language," 1982. |
| [36] | Facebook, "API - Facebook," Facebook, 2009. [Online]. Available: http://wiki.developers.facebook.com/index.php/API. [Accessed March 2010]. |
| [37] | Facebook, "Microsoft SDK for Facebook Platform - Facebook," Facebook, 2009. [Online]. Available: http://wiki.developers.facebook.com/index.php/Microsoft\_SDK\_for\_Facebook\_Platform. [Accessed 2010]. |
| [38] | Facebook, "Facebook Query Language (FQL) - Facebook Developers," Facebook, 2010. [Online]. Available: http://developers.facebook.com/docs/reference/fql/. [Accessed 2010]. |
| [39] | A. Bertolino, "Software testing research and practice," 2003. |

# APPENDIX A COMPLETE SOURCE CODE

**F:\Final Year Project\UI Modified (Without PopUps)\Default.aspx.vb 1**

'Importing Facebook Developer Toolkit 3.0 dll libraries Imports System

Imports System.Collections Imports System.Web.UI Imports Facebook.Rest Imports Facebook.Schema Imports Facebook.Session Imports Facebook.Utility Imports System.Xml

Partial Public Class \_Default Inherits System.Web.UI.Page

'Facebook Aggregator application and secret key declaration

Private Const APPLICATION\_KEY As String = "8e7457a8c77749e6dbb15f4776a9937a

"

Private Const SECRET\_KEY As String = "fde8f78a7d62c9e30da5d889c5319b4d"

Dim postItem As String

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load



On Error Resume Next

\_connectSession = New ConnectSession(APPLICATION\_KEY, SECRET\_KEY)

Dim con As New fbconnect

con.sessionSecret = \_connectSession.SessionSecret con.apiKey = \_connectSession.ApplicationKey con.secretKey = \_connectSession.ApplicationSecret con.uid = \_connectSession.UserId

con.sessionKey = \_connectSession.SessionKey

'If Not Page.IsPostBack And \_connectSession.IsConnected() Then ' MsgBox(" ")

'End If

If Not \_connectSession.IsConnected() Then LinkFriends.Visible = False LinkHome.Visible = False LinkInbox.Visible = False LinkProfile.Visible = False LinkPostLink.Visible = False LinkButton1.Visible = False LinkButton2.Visible = False LinkButton3.Visible = False ListBox1.Visible = False LinkLogout.Visible = False LinkLogin.Visible = True LinkVerify.Visible = True

lblUser.Text = "You are not loggged in" LinkLogin.Focus()

Else

\_facebookAPI = New Api(\_connectSession)

Dim us As user = \_facebookAPI.Users.GetInfo() lblUser.Text = "Logged in as " & us.name LinkVerify.Visible = False

If \_facebookAPI.Users.HasAppPermission(Enums.ExtendedPermissions. publish\_stream) And \_facebookAPI.Users.HasAppPermission(Enums. ExtendedPermissions.read\_stream) And \_facebookAPI.Users.HasAppPermission (Enums.ExtendedPermissions.read\_mailbox) And \_facebookAPI.Users. HasAppPermission(Enums.ExtendedPermissions.manage\_mailbox) Then



hasPermissions = True

Else

hasPermissions = False End If

If hasPermissions = True Then

If LinkFriends.Visible = False Then LinkFriends.Visible = True If LinkHome.Visible = False Then LinkHome.Visible = True

F:\Final Year Project\UI Modified (Without PopUps)\Default.aspx.vb 2

If LinkInbox.Visible = False Then LinkInbox.Visible = True

If LinkProfile.Visible = False Then LinkProfile.Visible = True If LinkPostLink.Visible = False Then LinkPostLink.Visible =

True

If LinkButton1.Visible = False Then LinkButton1.Visible = True If LinkButton2.Visible = False Then LinkButton2.Visible = True If LinkButton3.Visible = False Then LinkButton3.Visible = True If ListBox1.Visible = False Then ListBox1.Visible = True

If LinkLogout.Visible = False Then LinkLogout.Visible = True 'If LinkLogin.Visible = True Then LinkLogin.Visible = False

If ListBox1.Items.Count < 1 Then

postsToBeShown = 0

wallStream(postsToBeShown)

ListBox1.Visible = True

End If

Else

Response.Redirect("PermissionsPopUp.aspx")

End If

End If

ListBox1.Attributes.Add("onkeypress", "javascript:keyhandler()")

End Sub

Protected Sub LinkFriends\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkFriends.Click



Response.Redirect("Friends.aspx") End Sub

Protected Sub LinkProfile\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkProfile.Click



Response.Redirect("MyProfile.aspx") End Sub

Protected Sub LinkPostLink\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkPostLink.Click



Response.Redirect("PostToMyWall.aspx") End Sub

Protected Sub LinkInbox\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkInbox.Click



Response.Redirect("Message.aspx") End Sub

Private Sub wallStream(ByVal ofset As Integer) ' returns update status, posts, links, notes



On Error Resume Next

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dim | query As | | String = | String.Format("SELECT | | | | actor\_id, type, | target\_id | |
| , message,created\_time,post\_id | | | | | FROM | stream WHERE source\_id in (SELECT | | | |  |
| target\_id | | FROM connection | | WHERE | source\_id='" | | & \_connectSession.UserId & | | | "') |
| AND type | | <> 237 | AND type | <> 247 AND | | type <> | 259 limit " & (ofset | | + 10) | & " |
| offset | " | & ofset, \_connectSession.SessionKey) | | | | | |  |  |  |



Dim result As String = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

ReadWallXML(xmlDoc) End Sub

Public Sub ReadWallXML(ByVal xml As XmlDocument) On Error Resume Next

ListBox1.Items.Clear() Dim d As DateTime

'Try

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

F:\Final Year Project\UI Modified (Without PopUps)\Default.aspx.vb 3

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_ objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

If sLastElement = "actor\_id" Or sLastElement = "target\_id" Then sValue = getName(sValue)

End If

If sLastElement = "type" And sValue = "56" Then sValue = " to "

End If

If sLastElement = "type" And sValue = "46" Then sValue = " updated status: "

End If

If sLastElement = "type" And sValue = "80" Then sValue = " posted a link: "

End If

If sLastElement = "type" And sValue = "66" Or sValue = "236"

Then

sValue = " wrote a note: " End If

If sLastElement = "created\_time" Then

d = New DateTime(1970, 1, 1)

d = d.AddSeconds(sValue + currentOffset.TotalSeconds) 'post = post & sLastElement & " " & d.ToString & "<BR>" 'sValue = String.Format("{0:dd MMMM hh:mm:ss tt}", d) & "

<BR>"

sValue = " posted at " & String.Format("{0:dd MMMM hh:mm

tt}", d)

End If

postItem = postItem & " " & sValue

If sLastElement = "created\_time" Then

ListBox1.Items.Add(postItem) postItem = ""

End If

If sLastElement = "post\_id" Then ListBox1.Items.Item(ListBox1.Items.Count - 1).Value =

sValue

postItem = "" End If sLastElement = "" sCategory = ""

End If

Loop

F:\Final Year Project\UI Modified (Without PopUps)\Default.aspx.vb 4

objXMLTR.Close() 'Catch Ex As Exception

' Label3.Text = "The following error occurred: " & Ex.Message 'End Try

End Sub

Private Function getName(ByVal id As Long) As String On Error Resume Next

Dim query As String = String.Format("select name from profile where id = '" & id & "'", \_connectSession.SessionKey)



Dim result As String = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

Return readXMLUName(xmlDoc) End Function

Public Function readXMLUName(ByVal xml As XmlDocument) As String On Error Resume Next

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_ objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

'Label3.Text = Label3.Text & sCategory & vbCrLf & sLastElement & ": " & sValue



Return sValue

sLastElement = "" sCategory = ""

End If

Loop

objXMLTR.Close()

End Function

Private Sub Button1\_Click(ByVal sender As Object, ByVal e As System. EventArgs) Handles Button1.Click



F:\Final Year Project\UI Modified (Without PopUps)\Default.aspx.vb 5

On Error Resume Next

Response.Redirect("StreamPostPopUp.aspx?field1=" & (ListBox1. SelectedItem.Value))

End Sub

Protected Sub LinkLogin\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkLogin.Click



'Page.Title = sender.ToString & " " & e.ToString Response.Redirect("https://login.facebook.com/login.php?api\_key=



8e7457a8c77749e6dbb15f4776a9937a&auth\_token=&v=1.0") 'Response.Redirect("http://www.google.com")

End Sub

Protected Sub LinkLogout\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkLogout.Click



'MsgBox("You are loggin out, you will be redirected to the official facebook home page")



'Response.Redirect(\_facebookAPI.LogOffUrl)

End Sub

Protected Sub LinkButton2\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton2.Click

On Error Resume Next

If Not postsToBeShown > 50 Then postsToBeShown = postsToBeShown + 10 wallStream(postsToBeShown)

End Sub

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton1.Click

On Error Resume Next

If Not postsToBeShown < 0 Then postsToBeShown = postsToBeShown - 10 wallStream(postsToBeShown)

End Sub

Protected Sub LinkVerify\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkVerify.Click



End Sub

Protected Sub LinkHome\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkHome.Click



End Sub

Protected Sub LinkButton3\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton3.Click



postsToBeShown = 0 wallStream(postsToBeShown)

**F:\Final Year Project\UI Modified (Without PopUps)\Friends.aspx.vb 1**

Imports Facebook

Imports Facebook.Schema

Imports Facebook.Rest

Imports System.Xml

Partial Public Class Friends

Inherits System.Web.UI.Page

'Private Shared ReadOnly \_Instance As New Friends

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load



'Dim i As Integer

On Error Resume Next

If Page.IsPostBack Then Else

loadFriends(friendsToBeShown.ToString) friendsLabel()

End If

ListFriends.Attributes.Add("onkeypress", "javascript:keyhandler()") If ListFriends.Items.Count > 0 Then

'OpenPopUp(LinkFriendPhotos, "FriendPhotosPOPUP.aspx?field1=" & selectedFriendUid(), "", 880, 650)



End If End Sub

Public Function selectedFriendUid() As Long On Error Resume Next

Dim selected As String

'selected = \_facebookAPI.Friends.GetUserObjects().Item(ListFriends. SelectedIndex).uid



selected = ListFriends.Items.Item(ListFriends.SelectedIndex).Value Return selected

End Function

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click



On Error Resume Next LinkFriendPhotos.Visible = True LinkPostToWall.Visible = True LinkViewProfile.Visible = True LinkFriendWall.Visible = True LinkViewProfile.Focus()

LinkFriendPhotos.Text = "Open photos of " & ListFriends.Items.Item (ListFriends.SelectedIndex).Text



LinkPostToWall.Text = "Write on wall of " & ListFriends.Items.Item (ListFriends.SelectedIndex).Text



LinkViewProfile.Text = "Open profile of " & ListFriends.Items.Item (ListFriends.SelectedIndex).Text



LinkFriendWall.Text = "Open wall of " & ListFriends.Items.Item (ListFriends.SelectedIndex).Text



'MsgBox(ListFriends.SelectedIndex) End Sub

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton1.Click

On Error Resume Next

If Not ListFriends.Items.Count < 1 Then friendsToBeShown = friendsToBeShown + 100 loadFriends(friendsToBeShown.ToString) friendsLabel()

End If ListFriends.Focus()

End Sub

F:\Final Year Project\UI Modified (Without PopUps)\Friends.aspx.vb 2

Protected Sub LinkButton2\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton2.Click

On Error Resume Next

friendsToBeShown = friendsToBeShown - 100

If friendsToBeShown < 0 Then friendsToBeShown = 0 loadFriends(friendsToBeShown.ToString) friendsLabel()

ListFriends.Focus() End Sub

Private Sub friendsLabel() On Error Resume Next

If friendsToBeShown = 0 Then

Label1.Text = "List 1 contains " & ListFriends.Items.Count & "



friends" Else

Label1.Text = "List " & (friendsToBeShown / 100) + 1 & " contains " & ListFriends.Items.Count & " friends"



End If End Sub

Private Sub loadFriends(ByVal offset As String) On Error Resume Next

Dim query As String = String.Format("select name, uid from user where uid IN (SELECT uid2 FROM friend WHERE uid1='" & \_connectSession.UserId & "') limit 100 offset " & offset, \_connectSession.SessionKey)



Dim result As String = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

ReadFriendsXML(xmlDoc)

End Sub

Public Sub ReadFriendsXML(ByVal xml As XmlDocument) On Error Resume Next

Dim \_friends As String Dim i As Integer ListFriends.Items.Clear() 'Try

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_

F:\Final Year Project\UI Modified (Without PopUps)\Friends.aspx.vb 3

objXMLTR.NodeType = XmlNodeType.CDATA Then bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

'\_friends = \_friends & " " & sValue If sLastElement = "name" Then

ListFriends.Items.Add(sValue) '\_friends = ""

'i = i + 1 End If

If sLastElement = "uid" Then ListFriends.Items.Item(ListFriends.Items.Count - 1).Value =



sValue

End If sLastElement = "" sCategory = ""

End If

Loop objXMLTR.Close()

End Sub

Protected Sub LinkHome\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkHome.Click



Response.Redirect("Default.aspx") End Sub

Protected Sub LinkProfile\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkProfile.Click



Response.Redirect("MyProfile.aspx") End Sub

Protected Sub LinkFriends\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkFriends.Click



End Sub

Protected Sub LinkPostLink\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkPostLink.Click



Response.Redirect("PostToMyWall.aspx") End Sub

Protected Sub LinkInbox\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkInbox.Click



Response.Redirect("Message.aspx") End Sub

Protected Sub LinkViewProfile\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkViewProfile.Click



Response.Redirect("ProfilePOPUP.aspx?field1=" & selectedFriendUid()) End Sub

Protected Sub LinkPostToWall\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkPostToWall.Click



Response.Redirect("PostToFriendsWallPOPUP.aspx?field1=" & selectedFriendUid())

End Sub

Protected Sub LinkFriendPhotos\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkFriendPhotos.Click

'Response.Redirect("FriendPhotosPOPUP.aspx?field1=" & selectedFriendUid

())

Response.Redirect("Albums.aspx?field1=" & selectedFriendUid())

End Sub

Protected Sub LinkFriendWall\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkFriendWall.Click



MsgBox("This feature is not available yet") End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\Albums.aspx.vb 1**

Imports System.Xml

Partial Public Class Albums Inherits System.Web.UI.Page 'Dim z As Integer

Dim who As Long

'Dim albumSize As Integer

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

who = Request.QueryString("field1") If Page.IsPostBack Then

Else

reteriveAlbums()

' ListBox1.Focus() End If

ListBox1.Attributes.Add("onkeypress", "javascript:keyhandler()") End Sub

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click



z = 0

rawAlbum = ListBox1.SelectedItem.Text Response.Redirect("FriendPhotosPOPUP.aspx?field1=" & who) 'MsgBox(ListBox1.Items.Item(ListBox1.SelectedIndex).Value())

End Sub

Public Sub reteriveAlbums() On Error Resume Next ListBox1.Items.Clear() Dim query1 As String

query1 = String.Format("select size, name, aid from album where owner = '" & who & "'", \_facebookAPI.Session.UserId)



Dim result As String

result = \_facebookAPI.Fql.Query(query1)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result) ReadAlbumsXML(xmlDoc)

End Sub

Public Sub ReadAlbumsXML(ByVal xml As XmlDocument) On Error Resume Next

Dim \_friends As String Dim i As Integer ListBox1.Items.Clear() 'Try

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then

F:\Final Year Project\UI Modified (Without PopUps)\Albums.aspx.vb 2

If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_ objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

'\_friends = \_friends & " " & sValue If sLastElement = "size" Then

albumSize = sValue '\_friends = ""

'i = i + 1 End If

If sLastElement = "name" Then ListBox1.Items.Add(sValue & "," & albumSize) '\_friends = ""

'i = i + 1 End If

If sLastElement = "aid" Then ListBox1.Items.Item(ListBox1.Items.Count - 1).Value =

sValue

End If sLastElement = "" sCategory = ""

End If

Loop objXMLTR.Close()

End Sub

Protected Sub LinkHome\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkHome.Click



Response.Redirect("Default.aspx") End Sub

Protected Sub LinkProfile\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkProfile.Click



Response.Redirect("MyProfile.aspx") End Sub

Protected Sub LinkFriends\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkFriends.Click



Response.Redirect("Friends.aspx") End Sub

Protected Sub LinkPostLink\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkPostLink.Click



Response.Redirect("PostToMyWall.aspx") End Sub

Protected Sub LinkInbox\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkInbox.Click



Response.Redirect("Message.aspx")

End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\FriendPhotosPOPUP.aspx.vb 1**

Imports System.Xml

Partial Public Class FriendPhotosPOPUP Inherits System.Web.UI.Page

'Dim z As Integer Dim who As Long

'Dim albumSize As Integer

Dim u As Facebook.Schema.photo 'Dim albumSize As Integer

Dim photoInfo As String

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

who = Request.QueryString("field1") 'If Page.IsPostBack Then

'Else 'reteriveAlbums() ' ListBox1.Focus()

'MsgBox("rawAlbum " & rawAlbum) 'MsgBox("z " & z)

'MsgBox("who " & who) reterivePhoto() ListBox2.Focus()

'End If

'ListBox2.Attributes.Add("onkeypress", "javascript:keyhandler2()") End Sub

Public Sub reterivePhoto()

On Error Resume Next

ListBox2.Items.Clear()

Dim query1 As String

Dim albumNameAndSize() As String

albumNameAndSize = rawAlbum.Split(",")

'albumNameAndSize(0) = Replace(albumNameAndSize(0), " ", "") albumSize = albumNameAndSize(1)

'MsgBox(albumNameAndSize(0)) 'MsgBox(albumNameAndSize(1))

query1 = String.Format("SELECT caption,pid,link FROM photo WHERE aid IN ( SELECT aid FROM album WHERE owner='" & who & "' and name ='" &



albumNameAndSize(0) & "') limit 1 offset " & z, \_facebookAPI.Session. UserId)



Dim result As String

result = \_facebookAPI.Fql.Query(query1)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result) ReadPhotosXML(xmlDoc)

'ListBox2.Items.Add(getTag(5029226980740270577))

ListBox2.Focus()

End Sub

Public Function getComment(ByVal pid As String) As String

On Error Resume Next

Dim comments As String

Dim query As String

query = String.Format("select fromid,text from comment where object\_id in (select object\_id from photo where pid='" & pid & "') limit 10", \_facebookAPI.Session.UserId)



Dim result As String

F:\Final Year Project\UI Modified (Without PopUps)\FriendPhotosPOPUP.aspx.vb 2

result = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

comments = ReadXMLComments(xmlDoc)

Return comments

End Function

Public Function ReadXMLComments(ByVal xml As XmlDocument) As String On Error Resume Next

Dim comments As String 'Try

Dim sr As New IO.StringReader(xml.OuterXml) 'Label3.Text = ""

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

If sLastElement = "fromid" Then

comments = comments & " " & getName(sValue) End If

If sLastElement = "text" Then

comments = comments & " commented " & sValue & "~" End If

'Label3.Text = Label3.Text & "<B>" & sCategory & "<BR>" & sLastElement & "</B>" & ": " & sValue '& "<BR>" & tagss



End If

Loop

Return comments objXMLTR.Close() 'Catch Ex As Exception

'Page.Title = "The following error occurred: " & Ex.Message 'End Try

End Function

Public Function getTag(ByVal pid As String) As String

On Error Resume Next

F:\Final Year Project\UI Modified (Without PopUps)\FriendPhotosPOPUP.aspx.vb 3

Dim tags As String

Dim query As String

query = String.Format("SELECT text FROM photo\_tag WHERE pid= '" & pid & "'", \_facebookAPI.Session.UserId)



Dim result As String

result = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

tags = ReadXMLTags(xmlDoc)

Return tags

End Function

Public Function ReadXMLTags(ByVal xml As XmlDocument) As String On Error Resume Next

Dim tags As String 'Try

Dim sr As New IO.StringReader(xml.OuterXml) 'Label3.Text = ""

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value tags = tags & sValue & ", "

'Label3.Text = Label3.Text & "<B>" & sCategory & "<BR>" & sLastElement & "</B>" & ": " & sValue '& "<BR>" & tagss



End If

Loop

Return tags 'Label3.Text = tags objXMLTR.Close() 'Catch Ex As Exception

'Page.Title = "The following error occurred: " & Ex.Message ' End Try

End Function

F:\Final Year Project\UI Modified (Without PopUps)\FriendPhotosPOPUP.aspx.vb 4

Public Sub ReadPhotosXML(ByVal xml As XmlDocument) On Error Resume Next

'Try

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_ objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

photoInfo = photoInfo & sLastElement & " : " & sValue If ListBox2.Items.Count < 1 Then

ListBox2.Items.Add("Photo " & (z + 1)) End If

If sLastElement = "pid" Then currentPID = sValue

ListBox2.Items.Add("Tags: " & (getTag(sValue).Remove(getTag (sValue).Length - 2)))

Else

ListBox2.Items.Add(photoInfo) End If

photoInfo = "" sLastElement = "" sCategory = ""

End If

Loop

If ListBox2.Items.Count < 1 Then ListBox2.Items.Add("Comments") Dim commentss() As String

commentss = getComment(currentPID).Split("~")

Dim ci As Integer

For ci = 0 To commentss.Length - 1

ListBox2.Items.Add(commentss(ci))

Next

objXMLTR.Close() 'Catch Ex As Exception

F:\Final Year Project\UI Modified (Without PopUps)\FriendPhotosPOPUP.aspx.vb 5

' Label3.Text = "The following error occurred: " & Ex.Message 'End Try

End Sub

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton1.Click



On Error Resume Next z = z + 1

'MsgBox(z)

If z = albumSize Then z = 0 reterivePhoto()

End Sub

Protected Sub LinkButton2\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton2.Click



On Error Resume Next z = z - 1

If z < 0 Then z = albumSize - 1 reterivePhoto()

End Sub

Protected Sub Button2\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button2.Click



'\_facebookAPI.Comments.Add("1247637108978", "hello")

End Sub

Private Function getName(ByVal id As Long) As String On Error Resume Next

Dim query As String = String.Format("select name from profile where id = '" & id & "'", \_connectSession.SessionKey)



Dim result As String = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

Return readXMLUName(xmlDoc) End Function

Public Function readXMLUName(ByVal xml As XmlDocument) As String On Error Resume Next

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

|  |  |  |  |
| --- | --- | --- | --- |
| F:\Final Year Project\UI Modified (Without PopUps)\FriendPhotosPOPUP.aspx.vb | | | 6 |
|  | sLastElement = | objXMLTR.Name |  |
|  | ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_ | |  |
|  | objXMLTR.NodeType = XmlNodeType.CDATA Then | |  |
|  | bNested = False |  |  |
|  | sCategory = "<P>" & sCategory | |  |
|  | sCategory = "" |  |  |
|  | sValue = objXMLTR.Value | |  |
|  | 'Label3.Text = | Label3.Text & sCategory & vbCrLf & sLastElement |  |
| & ": | " & sValue |  |  |
|  | Return sValue |  |  |
|  | sLastElement = | "" |  |
|  | sCategory = "" |  |  |



End If

Loop

objXMLTR.Close()

End Function

Protected Sub LinkButton3\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton3.Click



Response.Redirect("Albums.aspx?field1=" & who) End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\keyPres.js 1**

function keyhandler(e) { if (document.layers)

Key = e.which;

else

Key = window.event.keyCode; if (Key == '13')

//window.document.getElementById("TextBox1").value = Key; window.document.getElementById('Button1').click();

//var myTextField = document.getElementById('TextBox1'); //alert(myTextField.value()) //alert(document.getElementById('TextBox1').value()); //alert("Key pressed! ASCII-value: " + Key); //PageMethods.getKey(Key); //window.document.getElementById("TextBox1").value;

}

**F:\Final Year Project\UI Modified (Without PopUps)\Message.aspx.vb 1**

Imports Facebook.Schema

Imports Facebook.Rest

Imports Facebook

Imports Facebook.Utility

Partial Public Class Message

Inherits System.Web.UI.Page

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

If ListThreads.Items.Count < 1 Then loadMessages(0) ListThreads.Focus() ListThreads.SelectedIndex = 0

Else

ListThreads.Focus() End If

On Error Resume Next

'OpenPopUp2(Button1, "Messages.aspx", "", 500, 500)

ListThreads.Attributes.Add("onkeypress", "javascript:keyhandler()")

End Sub

Public Sub OpenPopUp2(ByVal opener As System.Web.UI.WebControls.WebControl, ByVal PagePath As String, ByVal windowName As String, ByVal width As



Integer, ByVal height As Integer) On Error Resume Next

Dim clientScript As String Dim windowAttribs As String

'Building Client side window attributes with width and height. 'Also the the window will be positioned to the middle of the screen windowAttribs = "width=" & width & "px," & \_

"height=" & height & "px," & \_ "left='+((screen.width -" & width & ") / 2)+'," & \_ "top='+ (screen.height - " & height & ") / 2+'"

'Building the client script- window.open, with additional parameters clientScript = "window.open('" & PagePath & "','" & windowName & "','"

& windowAttribs & "');return false;"

'regiter the script to the clientside click event of the 'opener' control



opener.Attributes.Add("onClick", clientScript) End Sub

Public Function selectedThread() As Long On Error Resume Next

Dim selected As String

'selected = \_facebookAPI.Friends.GetUserObjects().Item(ListFriends. SelectedIndex).uid



selected = ListThreads.SelectedIndex Return selected

End Function

Private Sub loadMessages(ByVal folder As Integer)

On Error Resume Next

ListThreads.Items.Clear()

Dim fri As Integer = 0

For Each f As thread In \_facebookAPI.Message.GetThreadsInFolder(folder, \_connectSession.UserId, 20, TL)

'ListBox2.Items.Add(\_facebookAPI.Users.GetInfo(f.messages.message

(0).author\_id).name & ": " & f.subject)

If String.IsNullOrEmpty(f.subject) Then

F:\Final Year Project\UI Modified (Without PopUps)\Message.aspx.vb 2

If f.unread Then

ListThreads.Items.Add(ListThreads.Items.Count + (TL + 1) & " (Unread) " & " No Subject ")

Else

ListThreads.Items.Add(ListThreads.Items.Count + (TL + 1) &



" No Subject ") End If

Else

If f.unread Then

ListThreads.Items.Add(ListThreads.Items.Count + (TL + 1) & " (Unread) " & " " & (f.subject))

Else

ListThreads.Items.Add(ListThreads.Items.Count + (TL + 1) & " " & (f.subject))



End If End If

Next

If folder = 0 Then Label1.Text = "Inbox"

ElseIf folder = 1 Then Label1.Text = "Sent"

End If

threadFolder = folder ListThreads.Focus()

End Sub

Protected Sub LinkHome\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkHome.Click



Response.Redirect("Default.aspx") End Sub

Protected Sub LinkInbox\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkInbox.Click



loadMessages(0) End Sub

Protected Sub LinkSent\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkSent.Click



loadMessages(1) End Sub

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton1.Click

On Error Resume Next

If ListThreads.Items.Count > 1 Then TL = TL + 20

If Label1.Text = "Inbox" Then loadMessages(0)

Else

loadMessages(1) End If

Else

'LinkButton1.Visible = False End If

End Sub

Protected Sub LinkButton2\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton2.Click



On Error Resume Next TL = TL - 20

If TL < 0 Then 'LinkButton2.Visible = False

Else

If Label1.Text = "Inbox" Then

F:\Final Year Project\UI Modified (Without PopUps)\Message.aspx.vb 3

loadMessages(0)

Else

loadMessages(1) End If

End If

'LinkButton1.Visible = True End Sub

Private Sub Button1\_Click(ByVal sender As Object, ByVal e As System. EventArgs) Handles Button1.Click



On Error Resume Next listThreadIndex = selectedThread() Response.Redirect("Messages.aspx")

End Sub

Protected Sub LinkProfile\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkProfile.Click



Response.Redirect("MyProfile.aspx") End Sub

Protected Sub LinkFriends\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkFriends.Click



Response.Redirect("Friends.aspx") End Sub

Protected Sub LinkPostLink\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkPostLink.Click



Response.Redirect("PostToMyWall.aspx") End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\Messages.aspx.vb 1**

Imports Facebook.Schema

Partial Public Class Messages Inherits System.Web.UI.Page

'Dim localZone As TimeZone = TimeZone.CurrentTimeZone

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

If Not Page.IsPostBack Then

'Dim requestedMessage As Integer 'requestedMessage = Request.QueryString("field1") Label1.ReadOnly = True

Dim f As thread

Dim rec() As Long = Nothing Dim d As DateTime

f = \_facebookAPI.Message.GetThreadsInFolder(threadFolder, \_connectSession.UserId, 20, TL).Item(listThreadIndex)

Label1.Text = ""

For i = 0 To f.message\_count - 1 d = New DateTime(1970, 1, 1)

d = d.AddSeconds(f.messages.message(i).created\_time + currentOffset.TotalSeconds)



Label1.Text = Label1.Text & (\_facebookAPI.Users.GetInfo(f. messages.message(i).author\_id).name & " " & String.Format("{0:dd MMMM hh:mm :ss tt}", d)) & vbCrLf & (f.messages.message(i).body) & vbCrLf & vbCrLf



'rec = f.recipients.uid.ToArray

Next

'Label1.Text = Label1.Text & vbCrLf & listThreadIndex

End If

End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\MyStatus.aspx.vb 1**

Public Partial Class MyStatus Inherits System.Web.UI.Page Dim sts As String

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

sts = \_facebookAPI.Users.GetInfo().status.message If String.IsNullOrEmpty(sts) Then

Label1.Text = "No current status"

Else

Label1.Text = "Current status: " & sts End If

Button2.Attributes.Add("OnClick", "self.close()") End Sub

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton1.Click



On Error Resume Next \_facebookAPI.Users.SetStatus("") Response.Redirect(Request.Url.ToString)

End Sub

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click



On Error Resume Next

If Not String.IsNullOrEmpty(TextBox1.Text) Then \_facebookAPI.Users.SetStatus(TextBox1.Text)

End If Response.Redirect(Request.Url.ToString)

End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\PermissionsPopUp.aspx.vb 1**

Imports Facebook

Imports Facebook.Schema

Partial Public Class PermissionsPopUp

Inherits System.Web.UI.Page

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

'HyperLink1.NavigateUrl = String.Format("http://www.facebook.com/ authorize.php?api\_key={0}&v=2.0&ext\_perm={1}", \_connectSession. ApplicationKey, Enums.ExtendedPermissions.publish\_stream)



'MsgBox(HyperLink1.NavigateUrl.ToString) 'Response.Redirect("PermissionsPopUp.aspx")

If Not \_facebookAPI.Users.HasAppPermission(Enums.ExtendedPermissions. publish\_stream) Then

LinkButton1.Visible = True

Else

LinkButton1.Visible = False End If

If Not \_facebookAPI.Users.HasAppPermission(Enums.ExtendedPermissions. read\_stream) Then



LinkButton2.Visible = True

Else

LinkButton2.Visible = False End If

If Not \_facebookAPI.Users.HasAppPermission(Enums.ExtendedPermissions. read\_mailbox) Then

LinkButton3.Visible = True

Else

LinkButton3.Visible = False End If

If Not \_facebookAPI.Users.HasAppPermission(Enums.ExtendedPermissions. manage\_mailbox) Then

LinkButton4.Visible = True

Else

LinkButton4.Visible = False End If

OpenPopUp(LinkButton1, String.Format("http://www.facebook.com/authorize

.php?api\_key={0}&v=2.0&ext\_perm={1}", \_connectSession.ApplicationKey, Enums

.ExtendedPermissions.publish\_stream), "", 600, 400)

OpenPopUp(LinkButton2, String.Format("http://www.facebook.com/authorize

.php?api\_key={0}&v=2.0&ext\_perm={1}", \_connectSession.ApplicationKey, Enums

.ExtendedPermissions.read\_stream), "", 600, 400)

OpenPopUp(LinkButton3, String.Format("http://www.facebook.com/authorize

.php?api\_key={0}&v=2.0&ext\_perm={1}", \_connectSession.ApplicationKey, Enums

.ExtendedPermissions.read\_mailbox), "", 600, 400)

OpenPopUp(LinkButton4, String.Format("http://www.facebook.com/authorize

.php?api\_key={0}&v=2.0&ext\_perm={1}", \_connectSession.ApplicationKey, Enums

.ExtendedPermissions.manage\_mailbox), "", 600, 400)

End Sub

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton1.Click



End Sub

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click



Response.Redirect("Default.aspx") End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\PostLinkPopUp.aspx.vb 1**

Imports System.Net

Imports System.Net.Sockets

Partial Public Class PostLinkPopUp

Inherits System.Web.UI.Page

Dim ur As System.Uri

Dim x As String

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load



Button2.Attributes.Add("OnClick", "self.close()") End Sub

Protected Sub Button1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click



On Error Resume Next

x = TextLink.Text

If Left(x, 7) <> "http://" Then x = "http://" & x

Else End If

ur = New System.Uri(x)

'MsgBox(ur.ToString)

If Not String.IsNullOrEmpty(TextDescription.Text) Then \_facebookAPI.Links.Post(\_facebookAPI.Users.GetInfo().uid, ur,

TextDescription.Text)

Label1.Text = "Link Posted, click close to get back to Post To Wall

Page"

TextLink.Visible = False TextDescription.Visible = False Button1.Visible = False

Else

MsgBox("please provide description") End If

End Sub

Public Shared Function UrlIsValid(ByVal smtpHost As String) As Boolean

Dim br As Boolean = False Try

Dim ipHost As IPHostEntry = Dns.GetHostEntry(smtpHost) br = True

Catch se As SocketException br = False

End Try Return br

End Function End Class

**F:\Final Year Project\UI Modified (...PopUps)\PostToFriendsWallPOPUP.aspx.vb 1**

Imports Facebook.Rest

Imports Facebook

Imports Facebook.Schema

Partial Public Class PostToFriendsWallPOPUP

Inherits System.Web.UI.Page

Dim who As String

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

Page.Title = ("Post to friend")

who = Request.QueryString("field1")

Label1.Text = "Type something to post to your friends wall. Post will be sent upon pressing the POST button and you will be redirected to your friends page"



TextBox1.Focus() Button1.Attributes.Add("OnClick", "self.close()")

End Sub

Private Sub Button1\_Click(ByVal sender As Object, ByVal e As System. EventArgs) Handles Button1.Click

On Error Resume Next

Dim body As String = TextBox1.Text

\_facebookAPI.Stream.Publish(body, Nothing, Nothing, who, \_facebookAPI. Users.GetInfo().uid)



End Sub End Class

**F:\Final Year Project\UI Modified (Without PopUps)\PostToMyWall.aspx.vb 1**

Public Partial Class PostToMyWall

Inherits System.Web.UI.Page

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load



OpenPopUp(LinkPostLinkMy, "PostLinkPopUp.aspx", "", 500, 270) End Sub

Protected Sub LinkHome\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkHome.Click



Response.Redirect("Default.aspx") End Sub

Protected Sub LinkProfile\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkProfile.Click



Response.Redirect("MyProfile.aspx") End Sub

Protected Sub LinkFriends\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkFriends.Click



Response.Redirect("Friends.aspx") End Sub

Protected Sub LinkInbox\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkInbox.Click



Response.Redirect("Message.aspx") End Sub

End Class

**F:\Final Year Project\UI Modified (Without PopUps)\StreamPostPopUp.aspx.vb 1**

Imports System.Xml

Partial Public Class StreamPostPopUp

Inherits System.Web.UI.Page

Dim postID As String

'Dim localZone As TimeZone = TimeZone.CurrentTimeZone Protected WithEvents frame1 As System.Web.UI.HtmlControls. HtmlGenericControl



Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System. EventArgs) Handles Me.Load

On Error Resume Next

postID = Request.QueryString("field1") postID = Replace(postID, " ", "") readPost()

End Sub

Private Sub readPost() ' returns update status, posts, links, notes On Error Resume Next

Dim query As String = String.Format("SELECT actor\_id, attachment, target\_id, message,created\_time FROM stream WHERE post\_id ='" & postID & "'", \_connectSession.SessionKey)



Dim result As String = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

ReadXML(xmlDoc)

'ListBox1.Items.Add("Comments") Dim commentss() As String

commentss = getComment(postID).Split("~")

Dim ci As Integer

For ci = 0 To commentss.Length - 1

ListBox1.Items.Add(commentss(ci))

Next

End Sub

Public Sub ReadXML(ByVal xml As XmlDocument)

On Error Resume Next Dim d As DateTime

Dim linkHref As HyperLink Dim post As String

'Try

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to

F:\Final Year Project\UI Modified (Without PopUps)\StreamPostPopUp.aspx.vb 2

'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_ objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

If sLastElement = "source\_url" Then frame1.Attributes("src") = sValue

End If

If sLastElement = "type" Then frame1.Visible = True

End If

If sLastElement = "type" And sValue <> "video" Then 'frame1.Visible = True 'frame1.Attributes("src") = sValue

'Else 'frame1.Attributes("src") = "" frame1.Visible = False

End If

If sLastElement = "target\_id" Then sLastElement = "To"

post = post & sLastElement & " " & \_facebookAPI.Users. GetInfo(sValue).name & "<BR>"

End If

If sLastElement = "actor\_id" Then sLastElement = ""

post = post & sLastElement & " " & \_facebookAPI.Users. GetInfo(sValue).name & "<BR>"

End If

If sLastElement = "message" Then sLastElement = "Message"

post = post & sLastElement & " " & sValue & "<BR>" End If

If sLastElement = "description" Then sLastElement = "Description"

post = post & sLastElement & " " & sValue & "<BR>" End If

If sLastElement = "caption" Then sLastElement = "Source"

post = post & sLastElement & " " & sValue & "<BR>" End If

If sLastElement = "name" Then linkNameTemp = sValue

End If

If sLastElement = "href" Then hrefIndex = hrefIndex + 1 linkHref = New HyperLink linkHref.ID = postID & hrefIndex linkHref.Text = linkNameTemp linkHref.NavigateUrl = sValue linkHref.Target = "\_blank" linkNameTemp = ""

End If

If sLastElement = "created\_time" Then

sLastElement = "Posted at "

'sValue = "Posted Time"

d = New DateTime(1970, 1, 1)

F:\Final Year Project\UI Modified (Without PopUps)\StreamPostPopUp.aspx.vb 3

d = d.AddSeconds(sValue + currentOffset.TotalSeconds) 'post = post & sLastElement & " " & d.ToString & "<BR>" post = post & sLastElement & " " & String.Format("{0:dd



MMMM hh:mm tt}", d) & "<BR>" End If sLastElement = "" sCategory = ""

End If

Loop

Dim lblPost As Label lblPost = New Label lblPost.ID = postID lblPost.Text = post On Error Resume Next

PlaceHolder1.Controls.Add(lblPost)

PlaceHolder1.Controls.Add(linkHref) linkHrefDone = False objXMLTR.Close()

'Catch Ex As Exception

' Label3.Text = "The following error occurred: " & Ex.Message 'End Try

End Sub

Public Function getComment(ByVal pid As String) As String On Error Resume Next

Dim comments As String ListBox1.Items.Clear() Dim query As String

query = String.Format("select fromid,text from comment where post\_id = '" & pid & "' limit 10", \_facebookAPI.Session.UserId)



Dim result As String

result = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

comments = ReadXMLComments(xmlDoc)

Return comments

End Function

Public Function ReadXMLComments(ByVal xml As XmlDocument) As String Dim comments As String

Try

Dim sr As New IO.StringReader(xml.OuterXml) 'Label3.Text = ""

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to 'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > "

F:\Final Year Project\UI Modified (Without PopUps)\StreamPostPopUp.aspx.vb 4

sCategory = sCategory & sLastElement End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or objXMLTR. NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

If sLastElement = "fromid" Then

comments = comments & " " & getName(sValue) End If

If sLastElement = "text" Then

comments = comments & " commented " & sValue & "~" End If

'Label3.Text = Label3.Text & "<B>" & sCategory & "<BR>" & sLastElement & "</B>" & ": " & sValue '& "<BR>" & tagss



End If

Loop

Return comments objXMLTR.Close()

Catch Ex As Exception

Page.Title = "The following error occurred: " & Ex.Message End Try

End Function

Private Function getName(ByVal id As Long) As String On Error Resume Next

Dim query As String = String.Format("select name from profile where id = '" & id & "'", \_connectSession.SessionKey)



Dim result As String = \_facebookAPI.Fql.Query(query)

Dim xmlDoc As New XmlDocument

xmlDoc.LoadXml(result)

Return readXMLUName(xmlDoc)

End Function

Public Function readXMLUName(ByVal xml As XmlDocument) As String On Error Resume Next

Dim sr As New IO.StringReader(xml.OuterXml)

'Dim m As New XmlDocument

'm.LoadXml(FileName)

Dim objXMLTR As New XmlTextReader(sr)

Dim sCategory As String

Dim bNested As Boolean

Dim sLastElement As String

Dim sValue As String

'Read method loops through the XML stream Do While objXMLTR.Read

'Output elements and values

'Look at output in browser and compare to menu.xml file to

F:\Final Year Project\UI Modified (Without PopUps)\StreamPostPopUp.aspx.vb 5

'see exactly what is being done

If objXMLTR.NodeType = XmlNodeType.Element Then If bNested = True Then

If sCategory <> "" Then sCategory = sCategory & " > " sCategory = sCategory & sLastElement

End If

bNested = True

sLastElement = objXMLTR.Name

ElseIf objXMLTR.NodeType = XmlNodeType.Text Or \_ objXMLTR.NodeType = XmlNodeType.CDATA Then

bNested = False

sCategory = "<P>" & sCategory sCategory = ""

sValue = objXMLTR.Value

'Label3.Text = Label3.Text & sCategory & vbCrLf & sLastElement & ": " & sValue



Return sValue

sLastElement = "" sCategory = ""

End If

Loop

objXMLTR.Close()

End Function

Protected Sub LinkButton1\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles LinkButton1.Click



'LinkButton1.Enabled = False \_facebookAPI.Stream.AddComment(postID, TextBox1.Text) TextBox1.Text = ""

MsgBox("Comment Posted") End Sub

End Class