



**City University of Hong Kong
Department of Computer Science**

**BSCS Final Year Project 2005-2006
Final Report**

Project Title
**A Flash-based
Web Builder**

(Volume 1 of 1)

Student Name : **Leung Chi Yiu Vicker**
Student No. : **50378521**
Programme Code : **BSCCS**

Supervisor : **Dr. Andy Chun**
1st Reader : **Dr. Helena Wong**
2nd Reader : **Ms. Florence Mong**
Date : **15th, April, 2006**

For Official Use Only

Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

A Flash-based Web Builder

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name :

Signature :

Student ID :

Date :

Acknowledgement

The construction of a building will never be a one man job, so as this project. I would like to grab this opportunity to thank all the people who have participated or helped me throughout this year.

First of all I have to thank my supervisor Dr. Andy Chun giving me the opportunity to develop my virtual idea to real product. Without his clear guidance and valuable suggestions, I won't be able to have a channel to explore so many different areas and locate the best path to finish this project with fun. Dr. Helena Wong and Ms. Florence Mong also give their valuable time in this project, thank you.

Also I would like to express my appreciation to my friend Mr. Tim Shiu who has provided me so many suggestions on the program structure and user interface design. Special thanks to Mr. Warenix Wong, Ms. Soki Ho and Ms. Christy Chung participated in my user evaluation processes.

Last but not least, who have supported me in my whole life and accepted every fault of mine. My success will always belong to you two, mum and dad.

Abstract

Internet, personal web sites and online communities are very common nowadays. More people have their own web spaces to build their paradise. They want to put Flash contents in their web due to its excellent performance in providing attractive web multimedia experience.

However it is not easy to build Flash, and there are nearly no handy tools to assist. That is why the idea of a Flash-based web builder emerged. The builder is a Rich Internet Application (RIA) with the features that most users will be used in web building. The ultimate goal of the builder is to allow everyone able to build their own Flash-based website in an interactive interface with fun, and most importantly with good usability and easy to maintain.

To ensure the idea worth, several ways are analyzed which includes accessibility, usability, performance, resource consumption and most importantly the user experience. At last the future scope of the application can be drawn based on the analysis.



If we can make a HTML-based website simply by drag and drop, why we cant make a Flash-based website using the same approach? That is why the idea of building FLABER arises.

Table of Content

Student Final Year Project Declaration	2
Acknowledgement	3
Abstract	4
Table of Content	5
1. INTRODUCTION	8
1.1. Existing Problems	9
1.2. Project Objectives	10
1.3. Project Scope.....	10
1.4. Project Importance	10
2. BACKGROUND RESEARCH	11
2.1. Web Development Platform	12
2.2. Rich Internet Application Technology	13
2.3. Flash RIA Development Technology	16
2.4. Data Storage Approach	19
2.5. Weblog Application.....	21
2.6. Web-based Authoring Application	23
2.7. Flash-based Content Management System.....	26
3. METHODOLOGY	28
3.1. System Platform.....	29
3.2. Application Framework.....	30
3.2.1. Action Mode and Edit Mode.....	30
3.2.2. Authentication in Flash.....	31
3.2.3. XML Data Structure.....	32
3.3. Object Oriented Design.....	33
3.3.1. Class Design.....	33
3.3.2. Observer Design Pattern	35
3.3.3. Singleton Design Pattern	36

3.4.	FLABER Features and Screenshots	37
3.4.1.	Action Mode	37
3.4.2.	Administrator Authentication	38
3.4.3.	Edit Mode	39
3.4.4.	Web Properties	40
3.4.5.	Page Properties	42
3.4.6.	Page Elements	43
4.	RESULTS	46
4.1.	Feature Set Evaluation	47
4.2.	Performance Evaluation	48
4.3.	User Evaluation by Warenix Wong	51
4.4.	User Evaluation by Soki Ho	53
4.5.	User Evaluation by Christy Chung	55
5.	DISCUSSION	56
5.1.	Explanation on Performance Evaluation	57
5.2.	Comparison with Related Applications	58
5.3.	Benefits	59
5.4.	Limitations	59
5.5.	Applications	59
6.	CONCLUSION	60
6.1.	Critical Review	61
6.2.	Achievement	62
6.3.	Future Extension	63
	REFERENCES	65
	Cited References	66
	Additional References (Books)	68
	Additional References (Webs)	68
	APPENDICES	69
	Appendices A. User Documentation of FLABER 1.0 RC2	70
	Appendices B. Statistical Data of FLABER at SourceForge	78
	Appendices C. Copy of Email by Vesselin Drangajov dated Apr 4 th , 2006	80
	Appendices D. Project Progress Log	81



When a baby is a web admin...



1. INTRODUCTION

Many people have their own personal web sites and they are trying to put Flash contents inside their web pages. They have tried traditional web builder or authoring simply using Macromedia Flash. However there are many problems within which are mentioned in the coming section.



1.1. Existing Problems

The traditional web builders usually are What You See Is What You Get (WYSIWYG) editors. User can build their web pages simply by dragging and customizing the elements provided by the builder. Popular examples include Microsoft FrontPage and Macromedia Dreamweaver. The main problem of these editors is that they can only provide HyperText Markup Language (HTML) and Cascade Style Sheet (CSS) editing. They are not capable to create any Flash contents. Although Macromedia Dreamweaver has the function to make Flash buttons, it is still not possible to make a pure Flash-based web page.

The second approach to build Flash-based web pages is authoring using Macromedia Flash series. This is the most direct and common method. Macromedia Flash also has a WYSIWYG editing environment. Users can build their Flash content as simple as drawing using the paintbrush, but this is only limited to building animations. If the users want to build web pages which involve many interactions and navigations, advanced programming using ActionScripts is a must which may be a barrier to Flash beginners.

In recent year a new term came out which is Flash Content Management System. (Flash CMS) It is a simple Flash application which by means of a series of form filling and choice making, it will result a Flash-based web page. This is a very new topic, so the applications available right now are very immature and the functions provided are very limited. For example it is not possible to place objects according to users' will. Furthermore the user interface is not well designed and usability / accessibility issues are not concerned. As a result, this approach is not so popular yet.

1.2. Project Objectives

In order to overcome the problems stated in the previous section, this research is proposed. A Flash-based Rich Internet Application (Flash RIA) will be developed which allows the user to build Flash-based web pages. The application is named FLABER which means "FLAsh-based web BuildER". Traditional web builders will be analyzed in order to ensure that FLABER will have a common set of features available. The user interface (UI) will manipulate the common Windows applications' to enhance user experience. Web standards provided by W3C and Adobe will be followed to achieve usability and accessibility. Finally, several phases of user testing will be done to get comments to enhance the application. The ultimate objective is to allow web owners building their own Flash-based web pages with fun even they don't have any knowledge of Flash.

1.3. Project Scope

- ◆ Analyze web builders, weblog applications and Flash Content Management System
- ◆ Study and compare web technology chain
- ◆ Study and compare Rich Internet Application (RIA) implementation approaches
- ◆ Develop a Flash-based web builder, that is FLABER
- ◆ Evaluate FLABER in accessibility, usability, performance and user experience
- ◆ Review the weaknesses of FLABER and seek for further improvement

1.4. Project Importance

If FLABER is a success, more people will be able to build their Flash-based web sites. The advantage is that we can enjoy a better multimedia experience provided by Flash when browsing the internet. Together with the experience, usability and accessibility issues are concerned too which provide additional advantages. For example, search engines can retrieve the web contents easier which facilitate better information sharing in the net. Lastly, it will enhance users' experience in building a web, escaping from bored coding to funny playing.

2. BACKGROUND RESEARCH

Before the project starts, it is necessary to do a series of background research works. First, to know what is the current environment look like. Second, to study what are the problems, pros and cons of the current environment. Lastly, based on the research result, select the most suitable method to implement the new application.

The following are all the background researches that have been done. The first four items are different kinds of technologies that can be used when implementing FLABER, while the last three are existing products that are related to the research topic.



2.1. Web Development Platform

Web development platform refers to the environment which the web applications are deployed onto. The platform includes the combination of different kinds of operating system, web server, database and programming language. The most common combinations are Linux, Apache, MySQL, PHP (LAMP) and Windows, Internet Information Services, Microsoft SQL Server, ASP (WISA).

The major difference between the two chains is that LAMP is an open source chain which is totally free to be used while WISA is based on Microsoft technologies which charges. When comparing in deeper level, LAMP is considered to be a more secure, faster and flexible environment than WISA. [1] In terms of application purposes, LAMP is more suitable for light weight non commercial products, while WISA is better for enterprise. And finally if popularity is considered, according to the latest statistics of 2006 March from Netcraft [2], Apache is 48.19% more popular than IIS.

Actually there are even more combinations like Windows, IIS, Microsoft SQL Server, PHP (WIMP), Linux, Apache, Postgre SQL, PERL (LAPP), etc which are not analyzed here.

As FLABER is just a light web application and by considering the fact that personal web owners using LAMP is more popular, so it is chosen as the development platform.

2.2. Rich Internet Application Technology

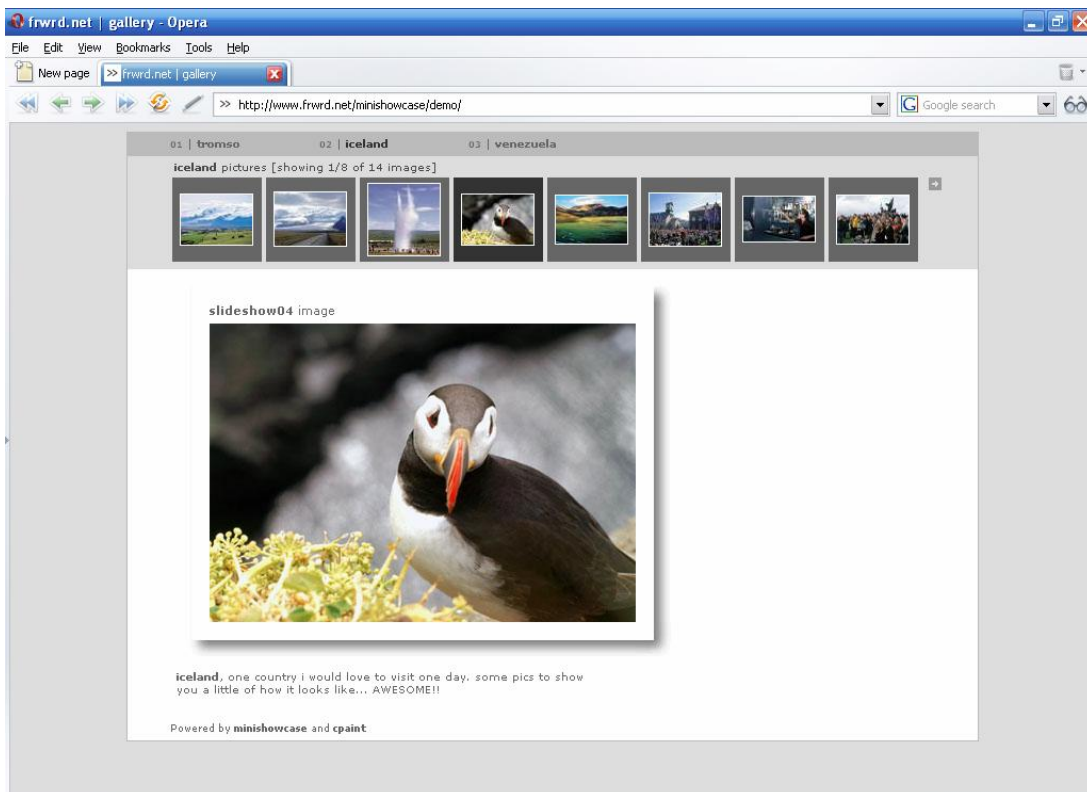
A rich internet application (RIA) is a kind of web application which has a well-developed user interface providing marvelous visual experiences and interactions to the visitors. Moreover, it is lighter in development effort than the traditional HTML web applications. [3] Examples of RIA technologies include Ajax, Java and Flash. [4]

Ajax is quite a new term in the web application world. It is mainly the combination of the traditional web presentation layer XHTML and CSS, interactive layer Document Object Model (DOM) wrapped by JavaScript. [5] Because only JavaScript is necessary to transform a traditional web page to RIA and no additional plug-in is required on the client side, Ajax is increasing in popularity recently. However the compatibility issue becomes the largest obstacle in developing Ajax application because different web browsers may have different rendering method on CSS and JavaScript.

Java applet is used for nearly a decade, which have a fixed standard and is well known in the industry. But the main point is that the Java Runtime Environment (JRE) is not the default built-in of most of the web browsers. This makes Java applet not as popular as Flash which have a 97.7% of internet-enabled PC coverage. [6]

Besides browser coverage, Flash also gives more freedom to the authors to create more creative products. Further the loading time is comparatively faster than Java. And because Flash player is independent on the web browser, the author can always assure that their product will behave exactly the same under different platforms.

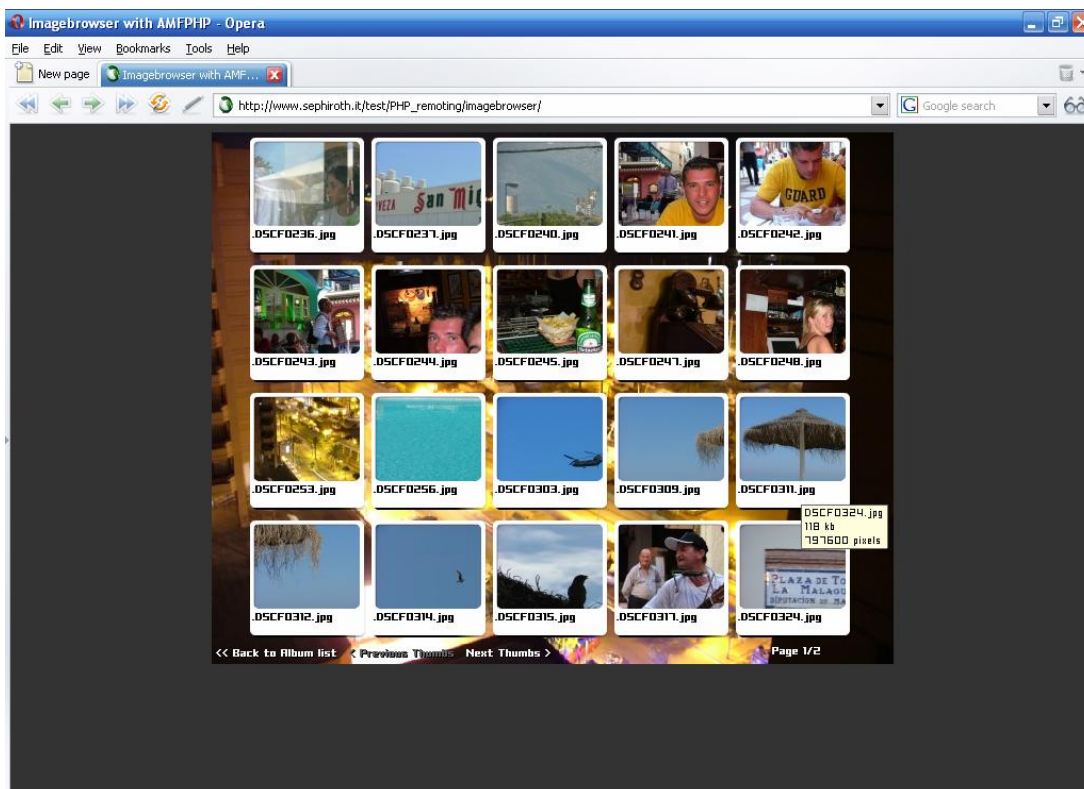
After considering the rich experience provided by Flash and the highest coverage among the three. It is chosen to be the technology used in implementing FLABER.



Screenshot 2.1 - An image gallery developed using Ajax - minishowcase [7]



Screenshot 2.2 - An image gallery developed using Java - ImageGallery [8]



Screenshot 2.3 - An image gallery developed using Flash - Image Browser [9]

2.3. Flash RIA Development Technology

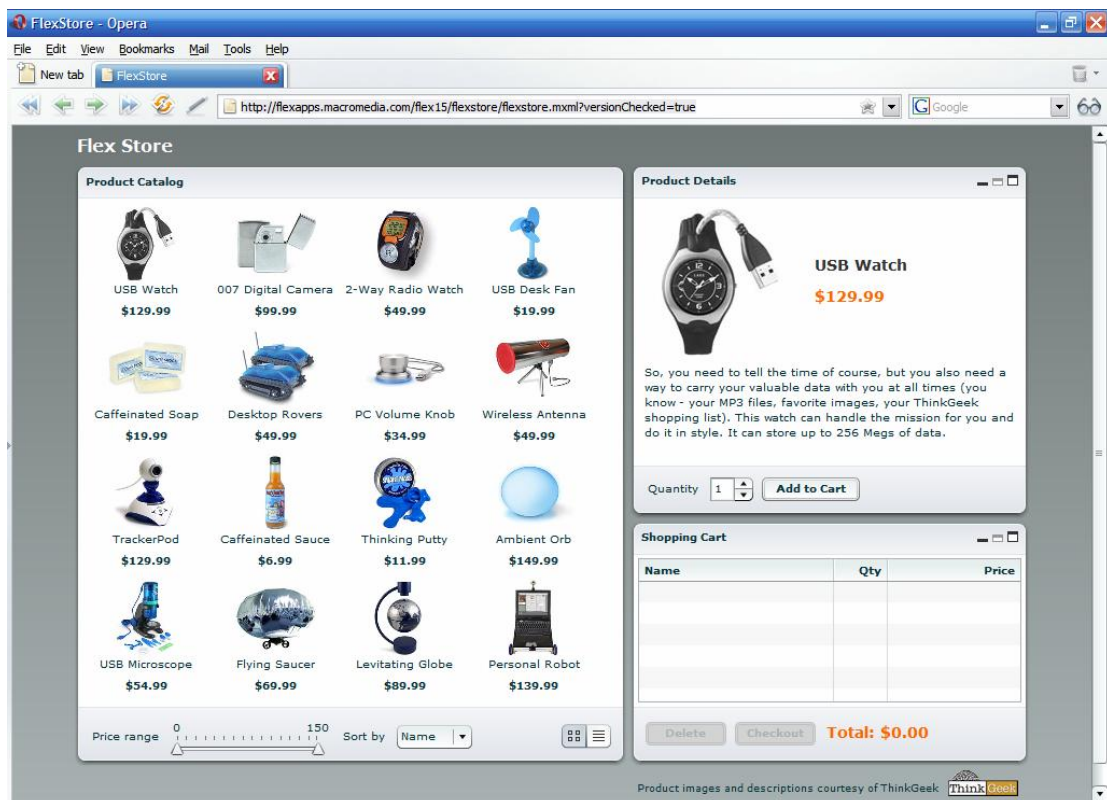
When developing Flash-based RIA, there are three major technologies including Adobe Flash [10], Adobe Flex [11] and OpenLaszlo [12]. The brief comparisons of them are stated below.

Adobe Flash, the primary authoring tool to make Flash contents (binary SWF files). The main programming language used is a ECMAScript-based language called ActionScripts. The major advantage using Adobe Flash over the other two is that it can fully utilize the power of Flash using ActionScripts, providing more advanced features like movieclip filtering and page transitions. Furthermore, the linkage between the objects inside the movie and the objects' class is more rigid.

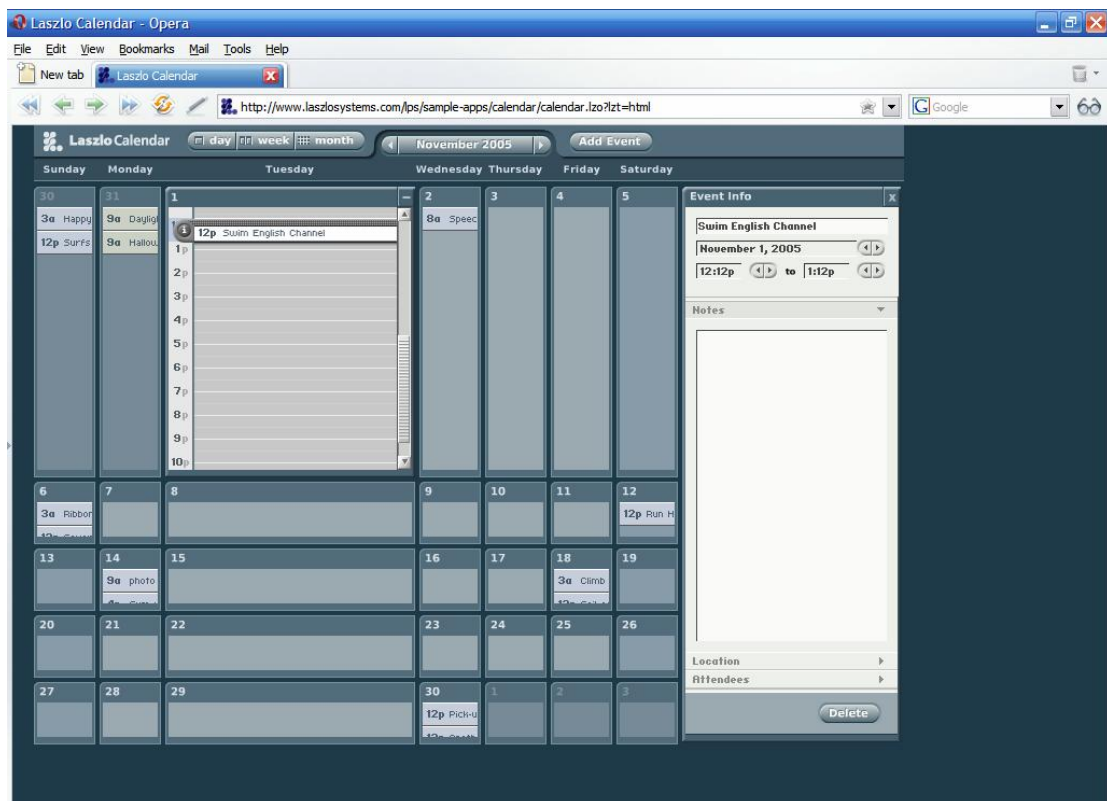
Adobe Flex and OpenLaszlo have a very similar working principle which is compiling their own specific language into binary SWF files. So users actually can skip the process of coding ActionScripts, replace it with some simpler tags like XML. This approach can ease the development effort, however sacrificing the development flexibility. Moreover, additional server setup is required to run the product which is not common in current web hosting services. So Macromedia Flash is chosen to be the RIA technology used in developing FLABER.



Screenshot 2.4 - An example of Adobe Flash developed RIA - Neovision [13]



Screenshot 2.5 - An example of Adobe Flex developed RIA - Flex Store [14]



Screenshot 2.6 - An example of OpenLaszlo developed RIA - Laszlo Calendar [15]

2.4. Data Storage Approach

When users try to use FLABER, there will be a large amount of data produced which includes the web pages' contents, system configurations, log files, etc. The method used to store these data is very crucial which will totally affect the system performance.

Usually there are three kinds of web data storage techniques, simple text file, eXtensible Markup Language (XML) and database. Using simple text file as a data store is the simplest method among all, because additional programming effort to interpret the data is not necessary. But due to the simplicity, the data is not structurally stored.

XML can be seen as a more structural text file. The data is stored inside some HTML like tags called nodes. And the nodes are linked with each other using the parent and child concepts of a tree. As there are standards defined for XML, the well defined structure allows a single XML to be ported into different places as a data source. For example, XML containing web contents can be ported to PHP application resulting HTML-based web pages. While ported to Flash application resulting Flash-based web pages. However, XML is not very efficient when data searching or editing are required.

To overcome the data processing problem in XML, database will be used in some situation. Database is a table like structure with each data record stored in a single row. Structured Query Language (SQL) is used to perform data manipulation. Usual database will be able to process thousands of records within several milliseconds. The only problem using database as data storage is that it consumes much server resources, because the data query cannot be cached which requires continuous communication between the server and the database.

In order to get a balance between the different methods, FLABER will mainly use XML for static contents, e.g. system configuration and web pages' contents. And database will be used for frequently modified contents and indexing. For example, the headings and keywords of the web pages' contents will be stored to facilitate topic searching.

```

&NavItemMC1_style=global
&NavItemMC1_x=20
&NavItemMC1_y=20
&NavItemMC1_text=Home
&NavItemMC1_url=page/index.xml
&NavItemMC2_style=global
&NavItemMC2_x=130
&NavItemMC2_y=20
&NavItemMC2_text=Techologies
&NavItemMC2_url=page/page2.xml

```

Figure 2.7 - An example data for a navigation bar in simple text file format

```

<NavItemMC style="global">
  <x>20</x>
  <y>20</y>
  <text>Home</text>
  <url>page/index.xml</url>
</NavItemMC>
<NavItemMC style="global">
  <x>130</x>
  <y>20</y>
  <text>Technologies</text>
  <url>page/page2.xml</url>
</NavItemMC>

```

Figure 2.8 - An example data for a navigation bar in XML format

Database Table - NavigationMenu

itemID	style	x	y	text	url
1	global	20	20	Home	page/index.xml
2	global	130	20	Technologies	page/page2.xml

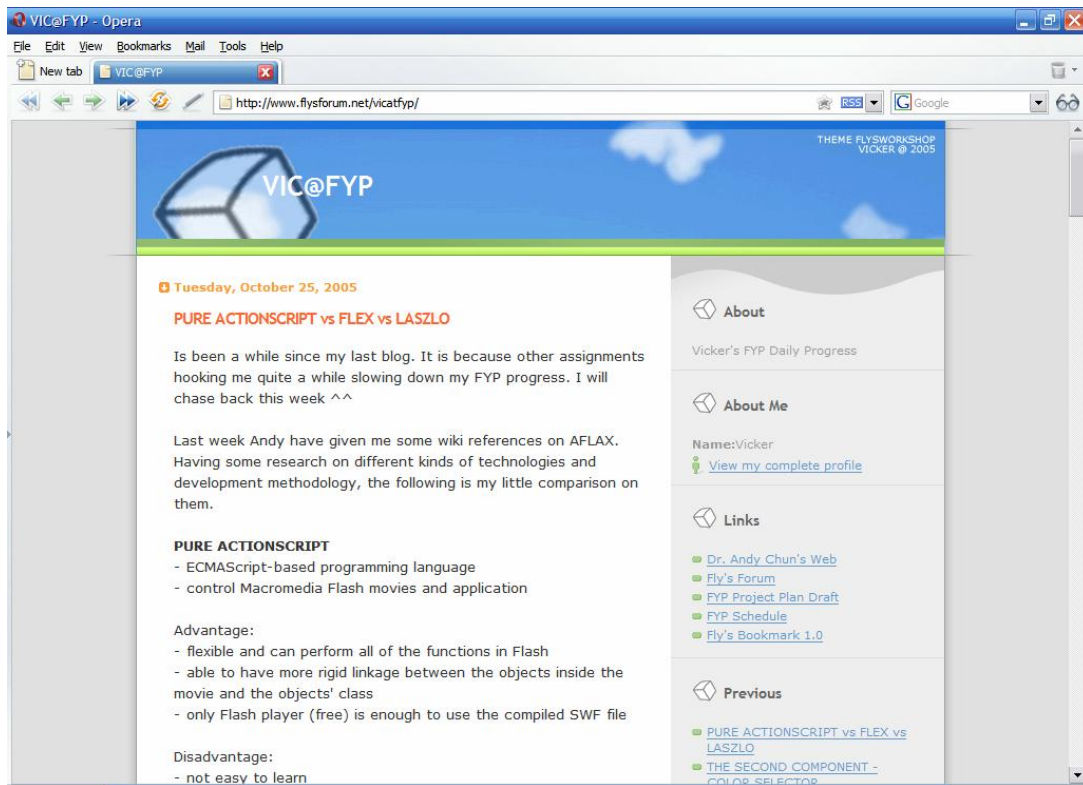
Figure 2.9 - An example data for a navigation bar in database format

2.5. Weblog Application

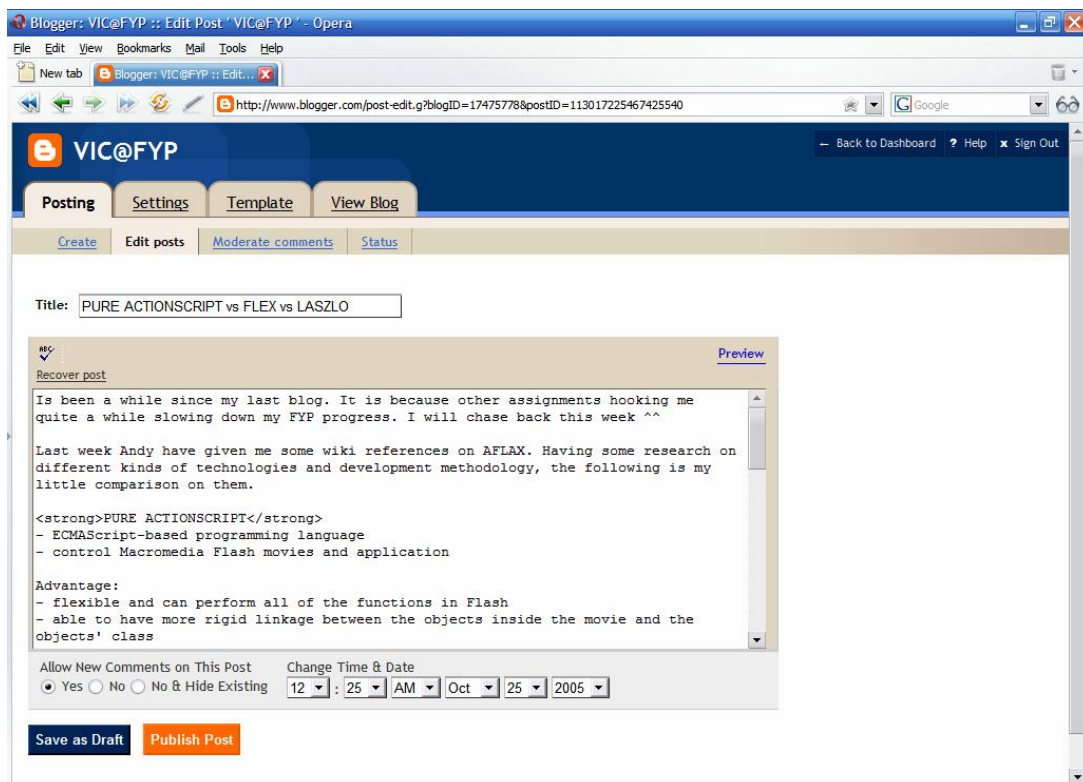
Weblog is an online personal publication system which is capable to create posts and allowing visitors to make comments. Users usually will update the weblog daily or weekly. The main difference between personal web page and weblog is that weblog itself have a predefined template, adding and editing of contents can be done simply by filling some forms. [16] Weblog is also distinguished as a personal content management system (CMS). Examples of weblog services include Blogger [17] and Xanga [18].

The process to create a post in weblog is very simple. First, users have to sign in to go into an administrative backend interface. Second, select "create new post" action. Third, typing in the necessary information in an editor and then save the content. Finally, browse the weblog page to check the updates.

The frontend (weblog page) and backend (administrative page) concept is well defined in weblog system, and both of them can be accessed through a browser. The advantage is that the owner can edit the weblog wherever internet access is available. This concept will be used in FLABER in order to achieve this convenience.



Screenshot 2.10 - A weblog done by Blogger [19]



Screenshot 2.11 - The administrative backend of Blogger [17]

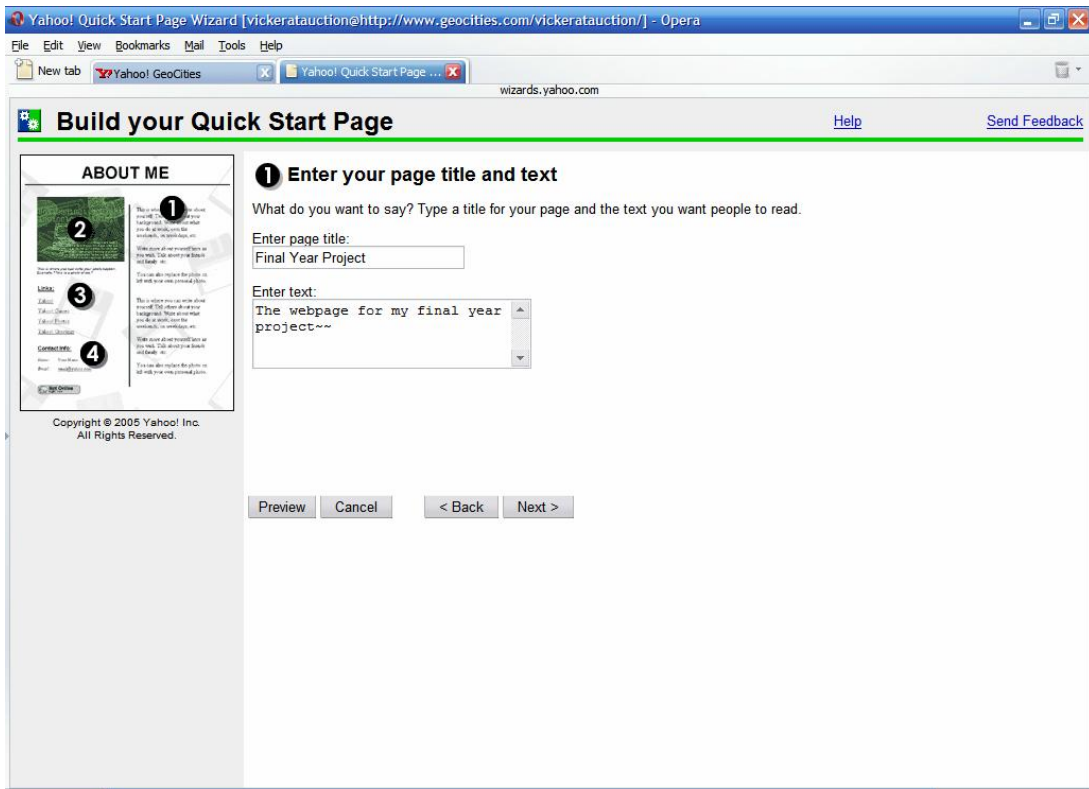
2.6. Web-based Authoring Application

Web owners usually have two kinds of web building methods, using simple text editors or web authoring applications. Web authoring application can be seen as a handy tool which assists the users to build a web page without caring about the codings. [20] It can be further divided into web templates editors and What You See Is What You Get (WYSIWYG) editors.

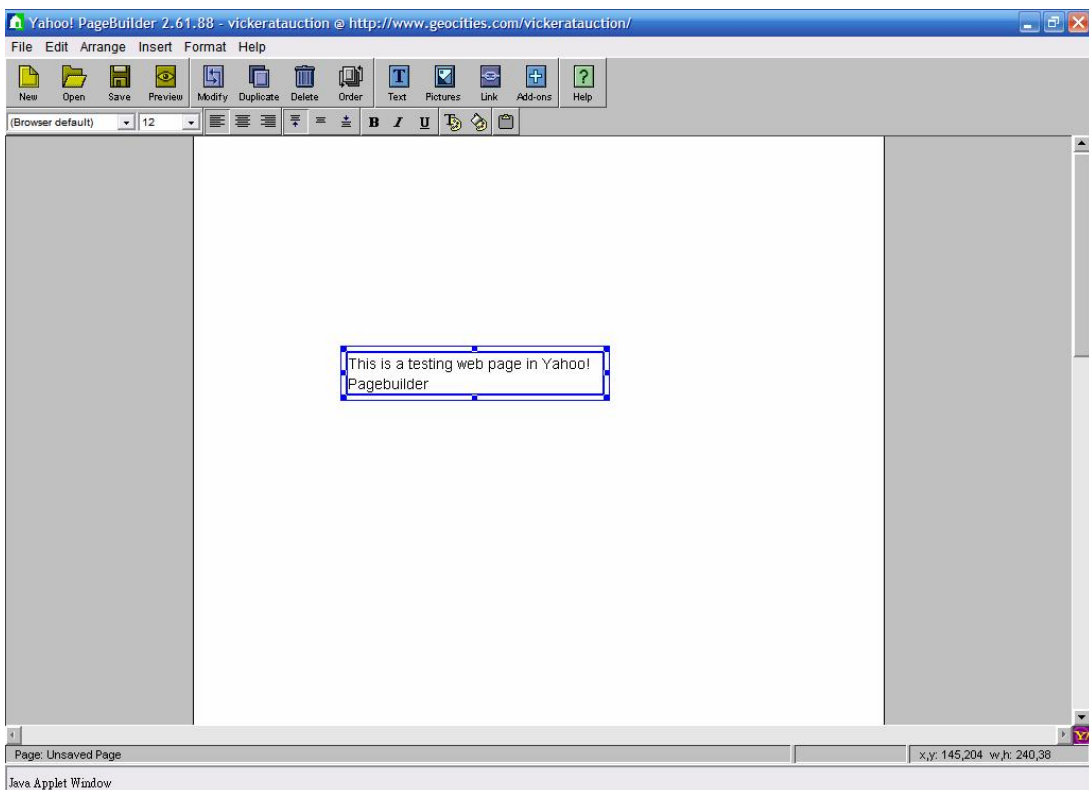
Web templates editors give a set of predefined template for the users to choose. After selection, the users can fill in their own contents into the template to produce their own web pages. One of the examples is the Yahoo! PageWizard. [21] The advantage of template editing is that the users do not need any design ideas. Furthermore, the authoring time is the fastest among all the methods. This is very suitable for web beginners.

WYSIWYG editors allow the users building their web by dragging web elements from the tool boxes to the document with some additional customization. Examples include the Yahoo! PageBuilder [21] and Google Page Creator [22]. Comparing with web templates editors, this approach provides more flexibility and freedom to the users. For example, it is possible to define the size and position of every element. However additional training is necessary in order to make full use of the editor, so it is more suitable for intermediate and expert users.

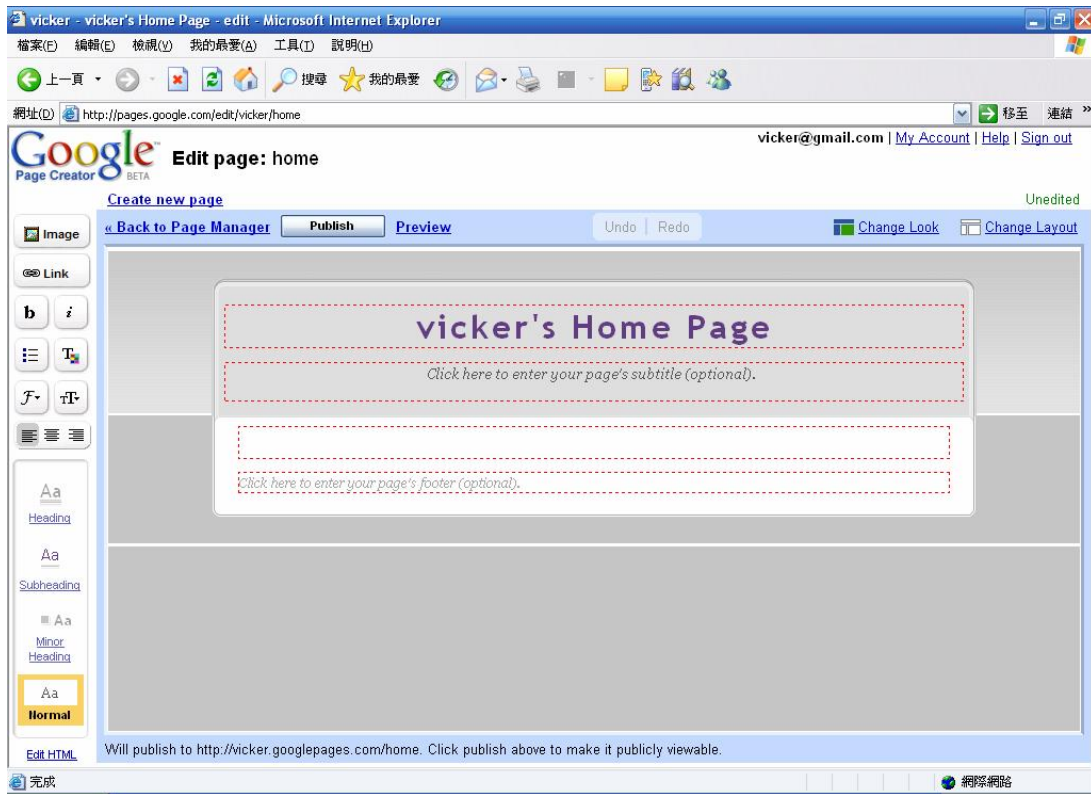
As free editing is the basic feature where wizard can be seen as an add-on, WYSIWYG will be implemented first in FLABER.



Screenshot 2.12 - The Yahoo! PageWizard provided by Geocities [21]



Screenshot 2.13 - The Yahoo! PageBuilder provided by Geocities [21]

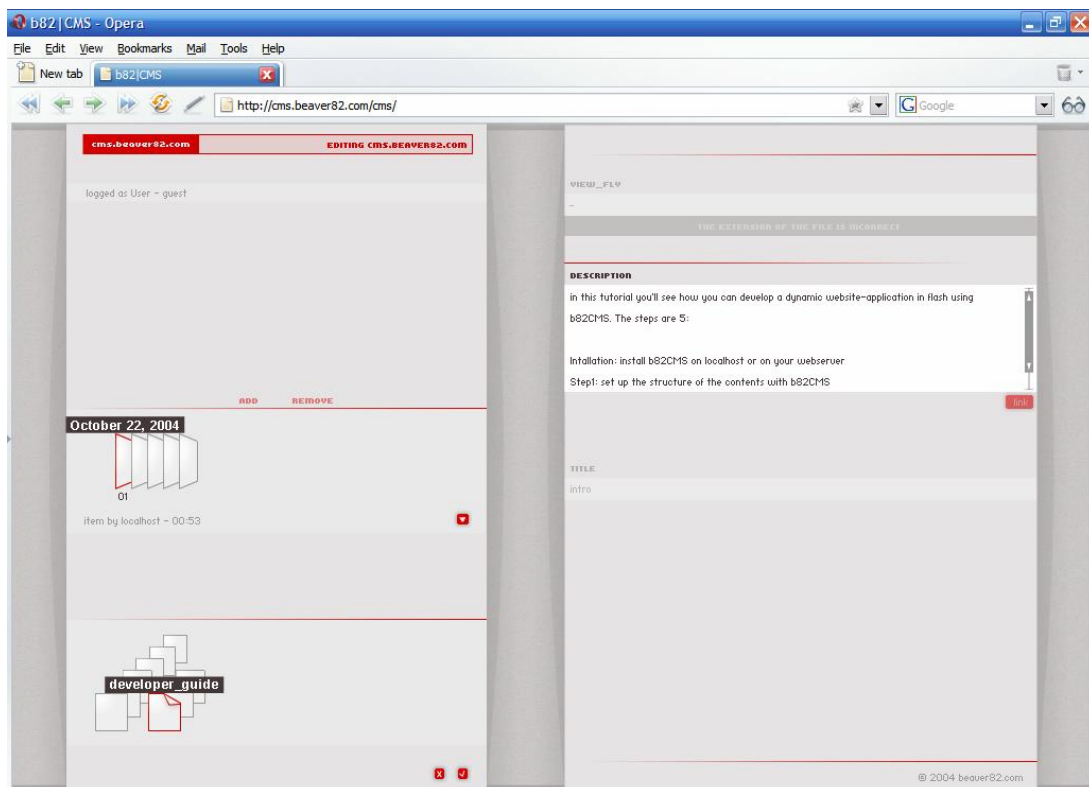


Screenshot 2.14 - The Google Page Creator provided by Google [22]

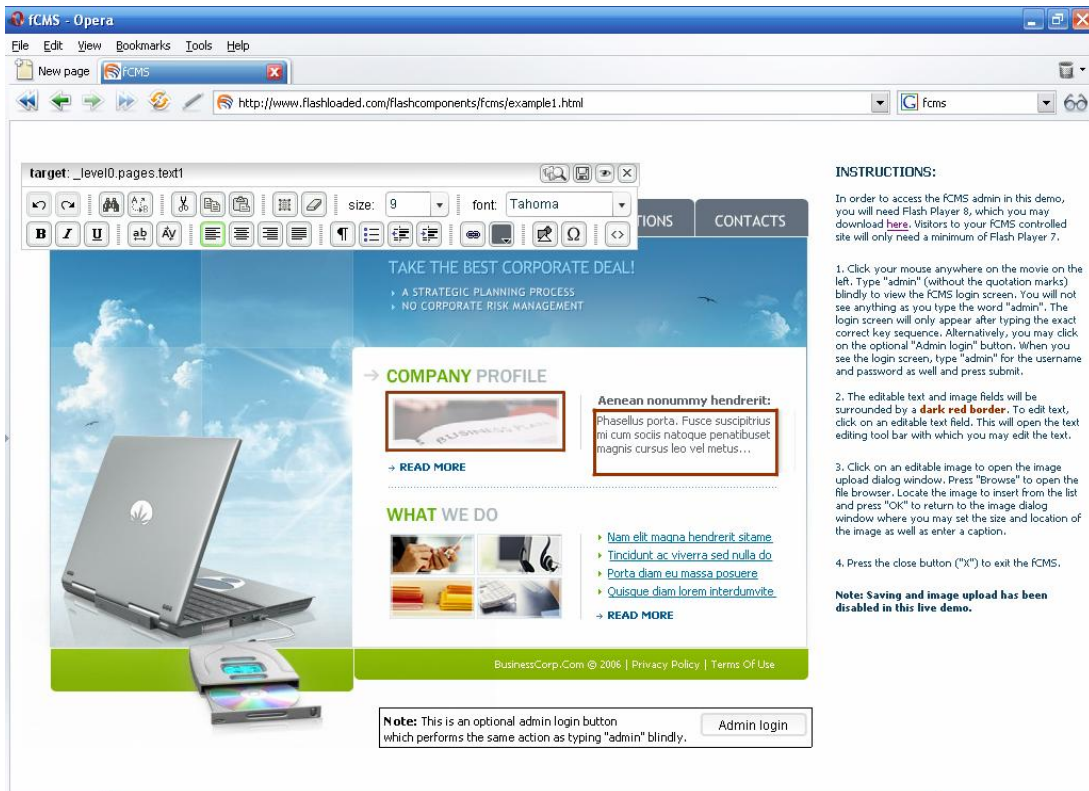
2.7. Flash-based Content Management System

Flash-based content management system (Flash CMS) is a brand new topic recent year. As more companies put Flash contents into the web sites. Every time when the content changes, they have to find someone who knows Flash to edit it which is totally not efficient. Because of this, Flash CMS is developed to facilitate the working process. Since it is still a new topic, the products available in the market are limited and very expensive. The more attractive Flash CMS examples are b82CMS [23], fCMS [24] and Flash Blocks [25].

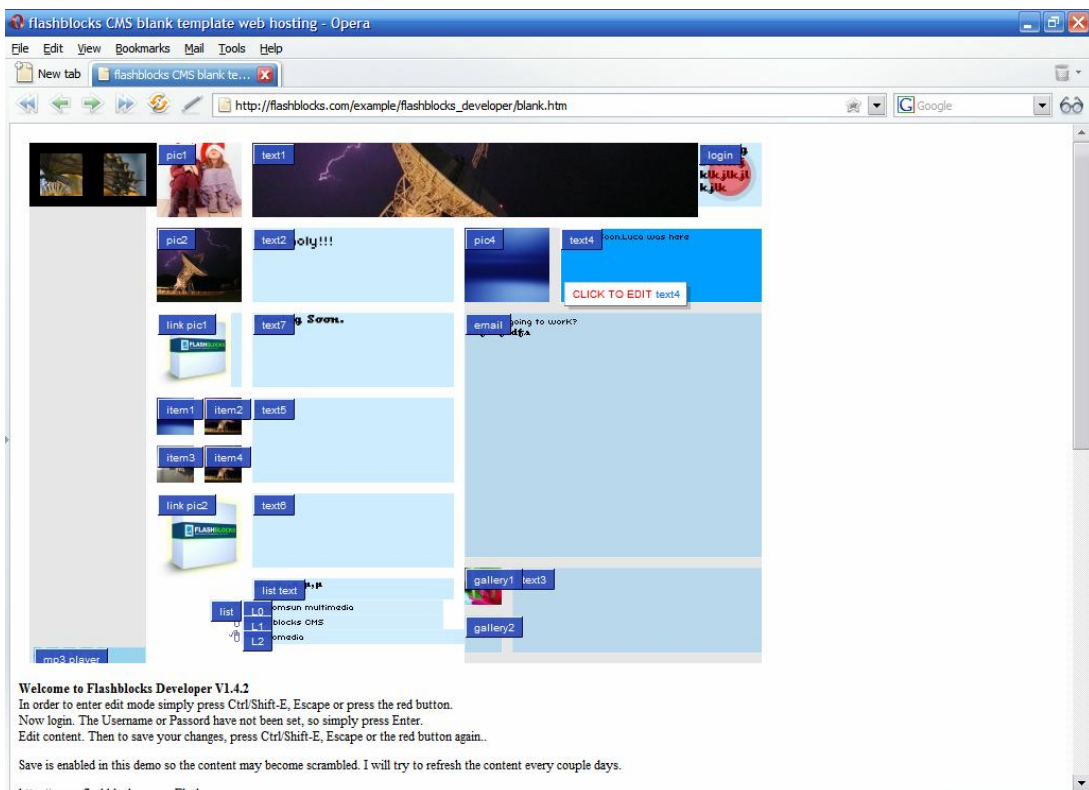
Because FLABER will have similar working principles as the Flash CMS, so it is very valuable to study these applications. Actually they have several common defects. First, the interface is not easy to use and it lacks consistency. Second, the accessibility concern is not considered. For example, it is not possible to view the web in non Flash format. Most importantly is that the authoring platform is very limited, users are not allowed to freely move and edit the page elements. The last point is that an intermediate level of Flash knowledge is needed in order to use their products, proved by a user's support question in the Flash Blocks forum dealing with some font issue. [26]



Screenshot 2.15 - The administrative interface for b82CMS [23]



Screenshot 2.16 - The administrative interface for fCMS [24]



Screenshot 2.17 - The administrative interface for Flash Blocks [25]

3. METHODOLOGY

After a series of background research works, it is important to decide how the new system FLABER is deployed and developed to reach the objectives stated before. The following sections will mainly focusing on these issues.

In the first three sections, the low level details of the system are mentioned, including the development platform, application structure and design approaches. And in the remaining section, the high level features of the system are described. Together will let you have a clear picture on the application - FLABER.



3.1. System Platform

FLABER is implemented in a Linux, Apache, MySQL, PHP (LAMP) environment. The main application structure is illustrated in diagram 3.1. The whole application can be mainly divided into three sections, administrative backend, data storage and user frontend.

Administrative back-end is implemented using Flash Rich Internet Application (Flash RIA), featuring those most web developers required. This includes creating, editing, linking and deleting of pages, administrative login and numerous page elements. The main purpose is to allow the web owners developing their web contents through a friendly user interface.

The data produced will be processed by a series of server-side PHP programs and at last stored in XML documents as data storage. Server-side will also provide some information to the administrative backend, e.g. authentication data, folder structure, etc.

User front-end is implemented also using Flash RIA. It will use the set of data previously created by the administrative backend to create the page view just like browsing a normal web site.

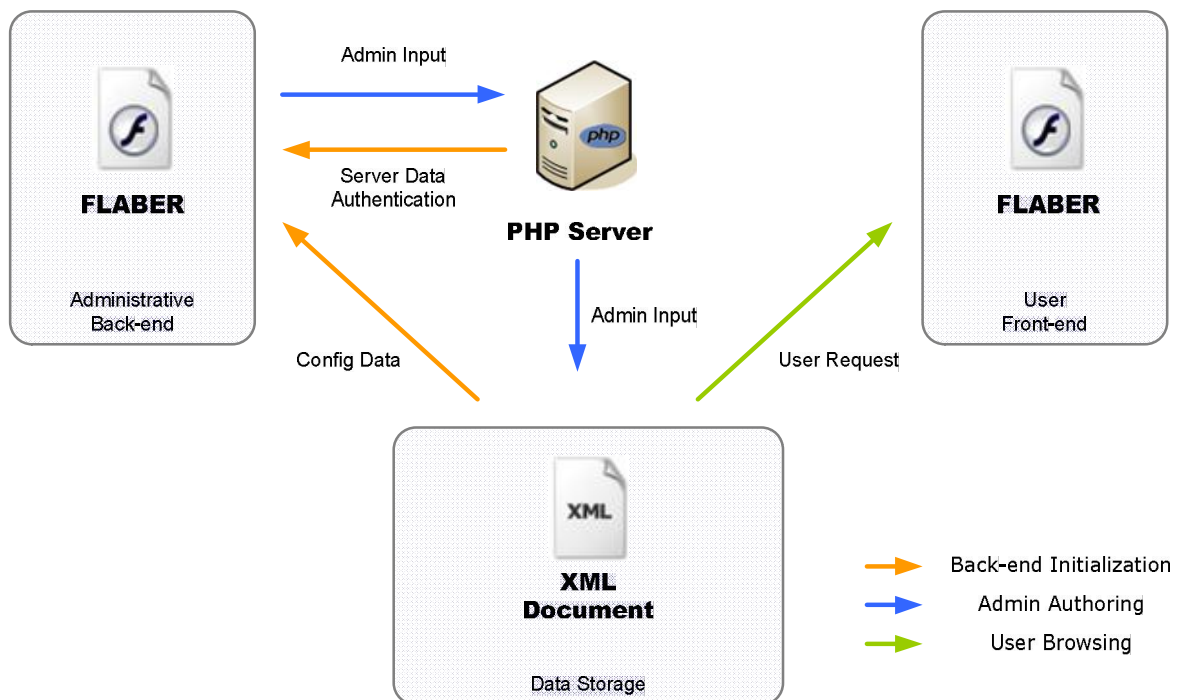


Diagram 3.1 - Brief system platform design of FLABER

3.2. Application Framework

3.2.1. Action Mode and Edit Mode

From the last subsection, I have briefly introduced the administrative back-end and user front-end. In the early stage of development, it is planned to implement two separate Flash programs. However when a more rigid structure comes out, I realized that the difference of the two user interface is just the presence of editing functions available. Further it is more direct and convenient for the user to edit the content without additional loading required, so in the first prototype, the feature of “Action Mode” and “Edit Mode” came out. And in release 1, admin authentication is required to switch between the two modes so that only the web administrator is allowed to edit the page content.



Screenshot 3.2 - Behavior of the navigation menu in Action Mode (left) and Edit Mode (right)

3.2.2. Authentication in Flash

In Flash development, security is always a very critical part to be considered. Normally developers will try to set a password variable within the Flash application, so that the authentication part can be processed immediately. When Flash decompilers [27] are widely spread in the market which allows anyone viewing the source code in a Flash application, this traditional approach is no longer secure.

Another popular way is getting the password from an external file, e.g. text / XML / server-side script. It seems that this approach is better than the previous but it is not. Because when Flash try to load the external file, the URL must be written inside the source, which means the hacker can directly browse to the URL to get the password.

In order to make the authentication 100% secure, the password must be checked on the server-side instead of client side (like the previous examples). In FLABER, when the user tries to login, the password is sent to PHP for authentication. The result is returned to Flash and the corresponding screen response / error message will be created. This implementation not only provides the best security, it also gives a sharp boundary between the role of Flash (view) and PHP (control). The only disadvantage is that additional loading time is required for the execution of the PHP script.

There are still alternative which make use of an obfuscator [28] changing all the variable names to dummies, so that the hacker wont be able to understand the source code. Unfortunately the URLs wont be altered, that means complete hacking is just a matter of time.

```
var tileName = ["Grass", "Bush"];  
for (var i = 0; i < 3; i++)  
{  
    eval ("create" + tileName[i})();  
}
```

```
var #2 = ["#3", "#4"];  
var #0 = 0;  
while (#0 < 3) {  
    eval ("#1" + #2[#0})();  
    #0++;  
}
```

Figure 3.3 - Sample source code before (left) and after (right) obfuscation

3.2.3. XML Data Structure

Another important process inside FLABER is considering how the data is read and saved. In the background research section, I have already considered using XML as the data storage. However how the structure should be designed in order to achieve portability and future extension?

XHTML format is used as a reference here for the design of FLABER's XML data format. In XHTML, every node is assigned to a page element, node attributes are used to define the element's properties. In FLABER, similar approach is used as shown in figure 3.4.

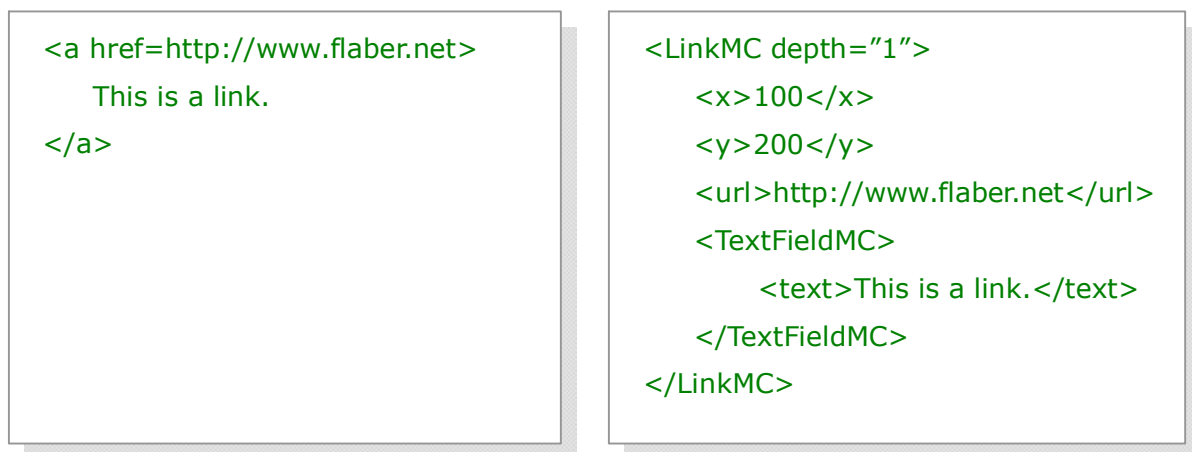


Figure 3.4 - Sample source code for a hyperlink using XHTML (left) and FLABER (right)

The nodes are using some plain English as node names so that it can be easily recognized by web developers. This helps some advanced users to develop their Flash web pages simply by typing codes like traditional XHTML web pages.

This XML design also facilitates the future extension of FLABER, the XHTML exporter which allows the user create XHTML version of their web site in FLABER simply by one click. (Details will be mentioned in chapter 6.3 - Future Extensions)

3.3. Object Oriented Design

3.3.1. Class Design

Class design for applications in Flash is quite different from those in C++ and others. For example in C++, you want to design a moving car. You will probably write the car class with all the moving functions and also a function for drawing the car. How about Flash? If using similar approach, it will become a tragedy. Let's say when the user tries to click the car image, it is rather impossible for the image calling back the car class for specific actions. (It is still possible using a dummy reference to the class, however not the best approach)

Actually when designing class in Flash, we will try to do the opposite. That is we will make the car image first and then applying a class to it. [29] In this way, the image will become the "parent" and all the member functions can work properly in an object oriented approach.

In FLABER, all the classes can be divided into 3 major packages, global, dialogue and page element. (Detailed structure is illustrated by figure 3.5)

Global package consists of all the classes which will be used or extended by other packages. E.g. TooltipMC class is responsible for displaying tip boxes for any page contents. MCFilters and MCTransitions are made to tackle any special image and animation effects.

Dialogue package contains the classes used to create dialogue boxes for the user to input data. They also contain their own update functions so that the user input can be saved or exported back to the global environment.

Page element package includes all the elements which will be shown on the stage, e.g. textfield, image, links, etc. These classes contain their own XML parsers so that their objects can load and save the XML data assigned.

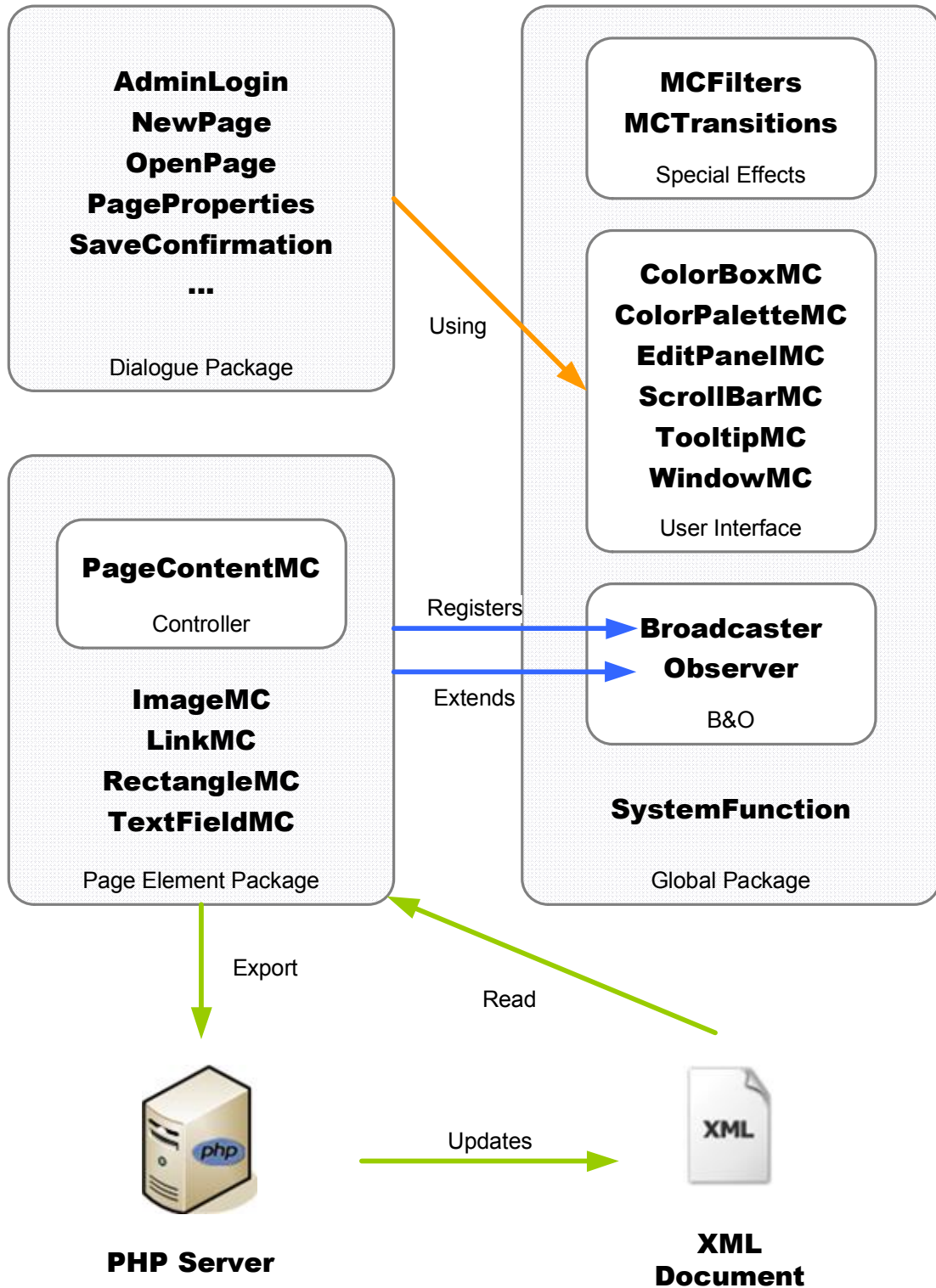


Figure 3.5 - Detailed class structure of FLABER

3.3.2. Observer Design Pattern

When developing FLABER, I have found a very troublesome problem about the Action Mode and Edit Mode. (Details in Chapter 3.2.1 - Action Mode and Edit Mode) Since the page elements will behave differently under different mode, it is important to let the page elements know what the current mode is.

In prototype version, I tried to use global variable to indicate current mode. When the page elements have to perform an action, e.g. user have mouse over the navigation item, the object will try to check the value of the global variable. This approach works peacefully at the very beginning. However I found that the resource consumption dramatically increases when more elements present on stage. That is why I try to find another approach.

After a series of book searching, I have found the observer design pattern in the book Essential ActionScript 2.0 [30]. The idea of the pattern is that we will create a broadcaster object with an array and a broadcast function. And then extend the page elements as observers, registered the observers to the broadcaster. So every time when the mode is changed, the broadcaster will execute the broadcast function to all the registered observers and the corresponding action will be executed.

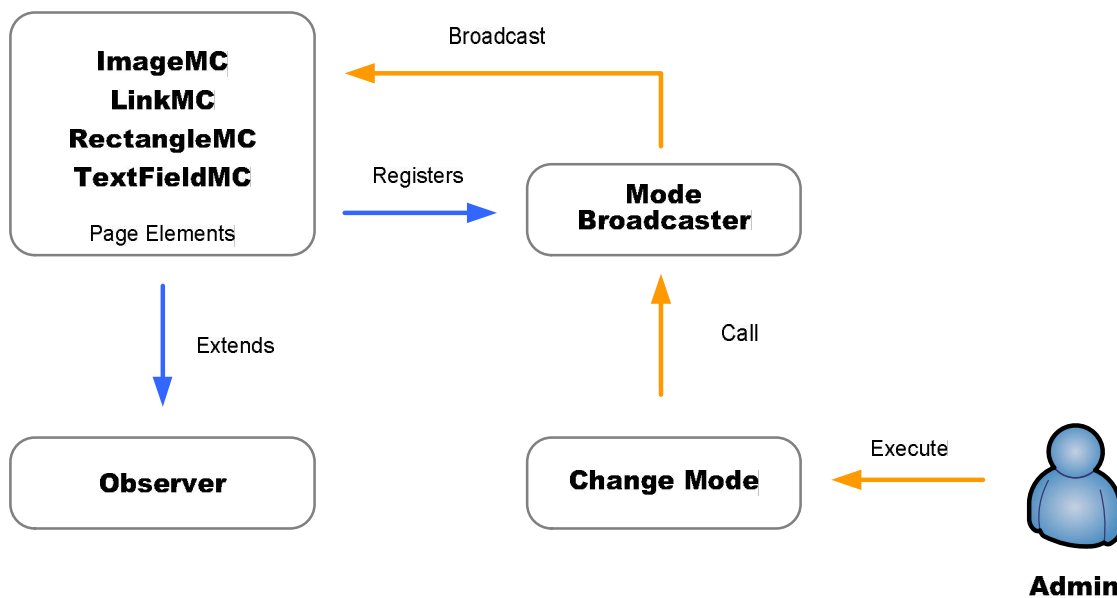


Figure 3.6 - Design of the observer pattern

3.3.3. Singleton Design Pattern

In many applications, there are usually classes that will have one and only one instances. For example in FLABER, tool tip class, status message class will only have one instance. It is not reasonable to have two tool tips appear at the same time, so as the status message bar. That is why it is necessary to ensure that these classes will not be created more than once.

The singleton design pattern is made to tackle this issue. The logic is whenever a new instance is being made, it will make a check on a static member variable. If this variable is set, it means an instance has been made before and no new instance will be created.

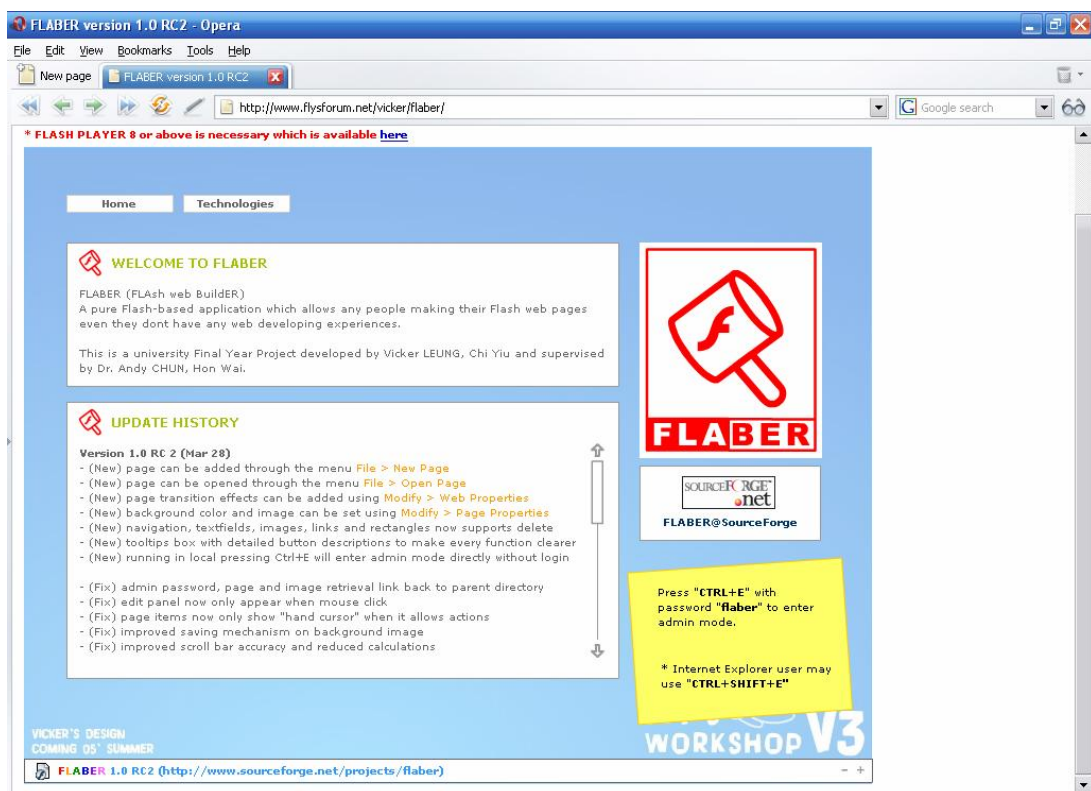
```
Class Singleton {  
    private static var _instance:Singleton = null;  
  
    private function Singleton () {  
    }  
  
    public static function getInstance ():Singleton {  
        if (Singleton._instance == null) {  
            Singleton._instance = new Singleton ();  
        }  
        return Singleton._instance;  
    }  
}  
  
var singleton_obj:Singleton = Singleton.getInstance ();
```

Figure 3.7 - Sample code on implementing singleton design pattern

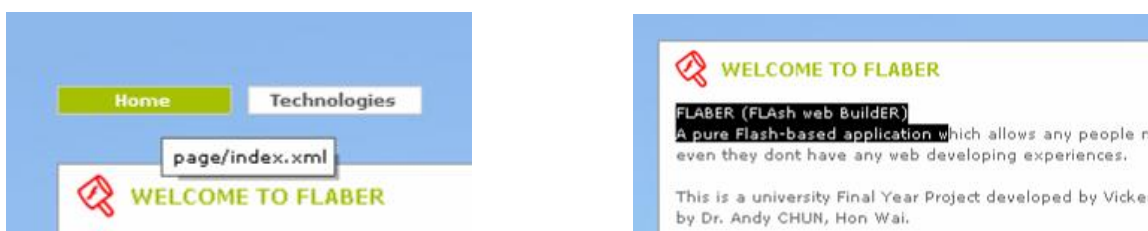
3.4. FLABER Features and Screenshots

3.4.1. Action Mode

Action mode is the first view when visitors browsing FLABER. It looks like a simple web site embedded with a Flash movie. The page contents are similar as the traditional web pages done in XHTML. Mouse over tool tip, text highlighting and status message bar are available to simulate the normal browser environment.



Screenshot 3.8 - First look of FLABER in action mode

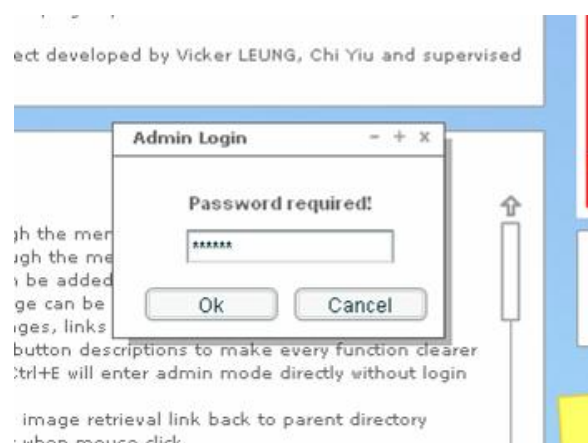


Screenshot 3.9 - Navigation menu tool tip (left) and text highlighting (right)

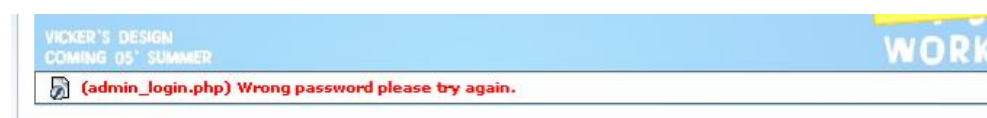
3.4.2. Administrator Authentication

In action mode, user can press “Ctrl+E” or “Ctrl+Shift+E” in any page to bring up the administrator login box. If the user provides the correct password, FLABER will make the menu bar available to the user. Otherwise error message will be prompted. Making use of the menu bar, administrator can do all kinds of editing to the web site.

Browser compatibility issue is considered in the key press combination design. Since “Ctrl+E” will bring up the search function in Internet Explorer, so an alternative “Ctrl+Shift+E” is made. Furthermore pressing “Enter” in the login box will trigger the form submission, which is a user interface design concern.



Screenshot 3.10 - Administrator login box



Screenshot 3.11 - Wrong password status message

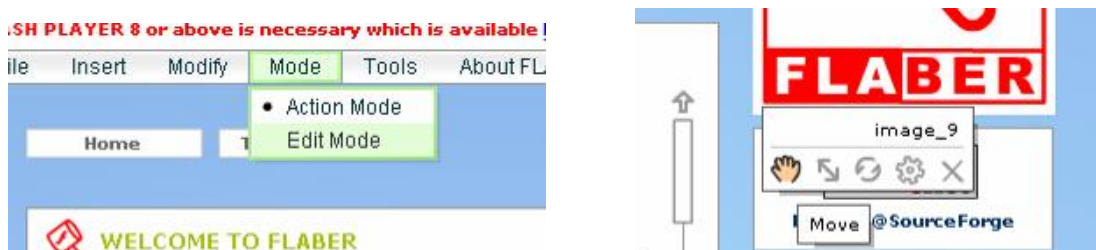


Screenshot 3.12 - Main menu bar with all the editing functions

3.4.3. Edit Mode

In order to edit the page content, administrator has to switch from action mode to edit mode which is available in the menu bar. After the mode switching, all the page elements become editable which will be highlighted when user try to mouse over them.

If the user left clicks on the item, an editing panel will pop out near the item providing a fast access to some of the major functions like move, resize, rotate, properties and delete. Edit panel ensures that all the functions are centralized in an area, assisting the user during the design process.



Screenshot 3.13 - Mode changing menu (left) and the edit panel of an image (right)

Some of the page elements like text fields do not support rotation. Instead of removing the function completely from the edit panel, it is lightened so that the whole edit panel is kept consistent in design. This is one of the Shneiderman’s Eight Golden Rules of Interface Design [31].



Screenshot 3.14 - Edit panel of a text field which shows a lightened rotate function

3.4.4. Web Properties

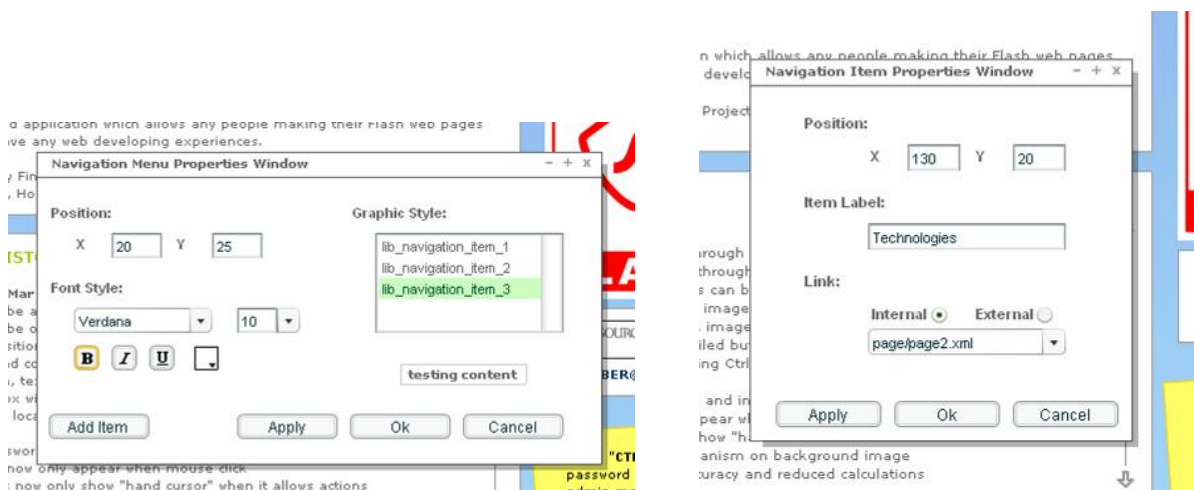
In terms of the whole web site, there are a lot of features available for the user to design. This includes the index page, navigation menu and page transition effect.

Index page indicates which page will be loaded when FLABER is first started. In traditional web site development, the index page's name is fixed to "index". Since it is server independent in FLABER, giving this setting available will provide the user a more flexible environment.



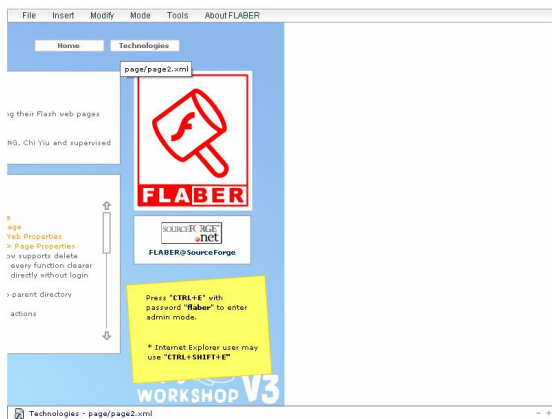
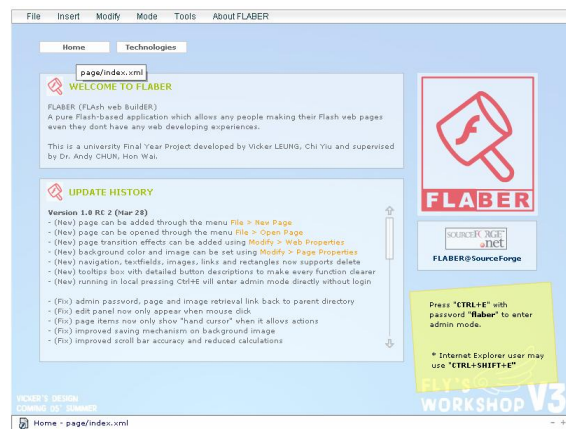
Screenshot 3.15 - Index page setting field within web properties

Usually a web site will bear a consistent navigation menu for the visitor to browse through the web site. For this reason, navigation menu setting is available in web properties. User can define whether a navigation menu is used, and advanced settings are available in the navigation menu's edit panel.



Screenshot 3.16 - Navigation menu (left) and navigation item (right) properties window

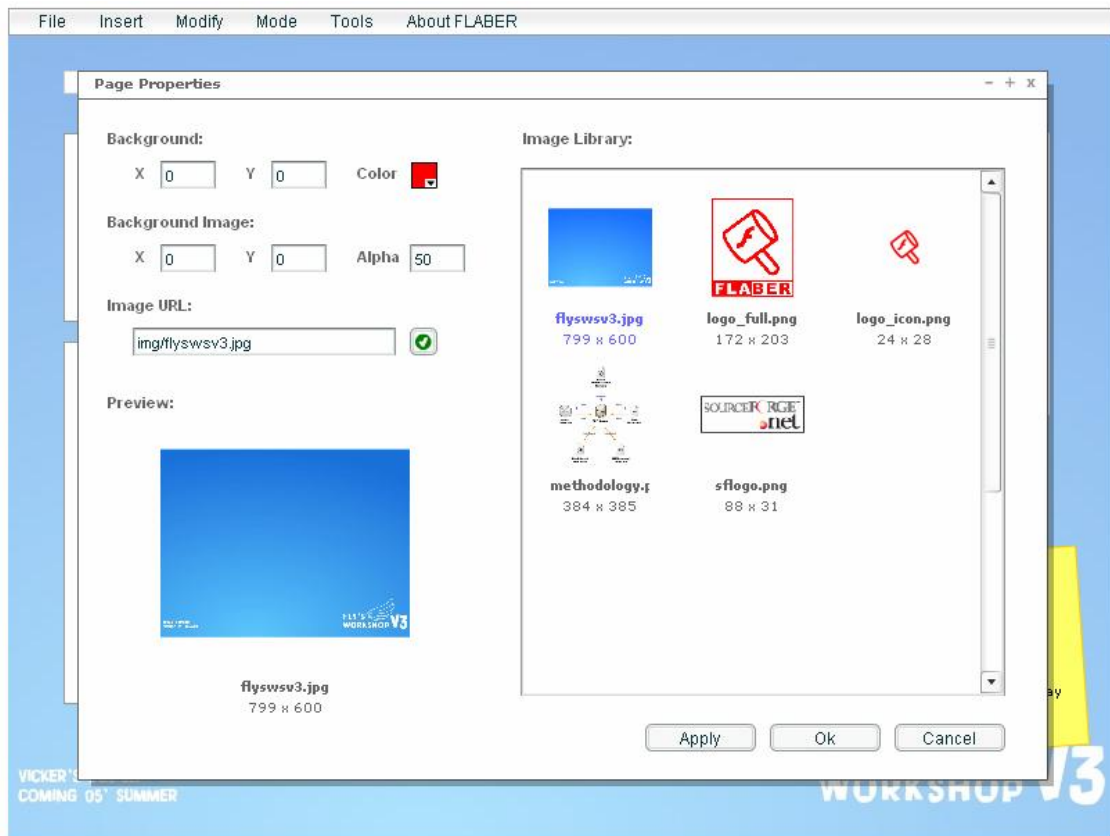
Some web designers will try to use javascript or java to achieve some fancy animation effect when a page flip occurs. Making use of the power of Flash player version 8, page transition effects are available for the user to choose. Currently five effects are supported including iris, fade, fly, pixel dissolve and wipe. Hopefully more will be available in the future.



Screenshot 3.17 - Different kinds of page transition effects, iris (top left), fade (top right), fly (bottom left) and pixel dissolve (bottom right)

3.4.5. Page Properties

Working in pairs with web properties is the page properties which the target is a single page instead of the whole web site. User can set the background color and background image within the page properties window. Images placed in server side is retrieved and showed in terms of thumbnails, allowing the user to have a glance of the images available.

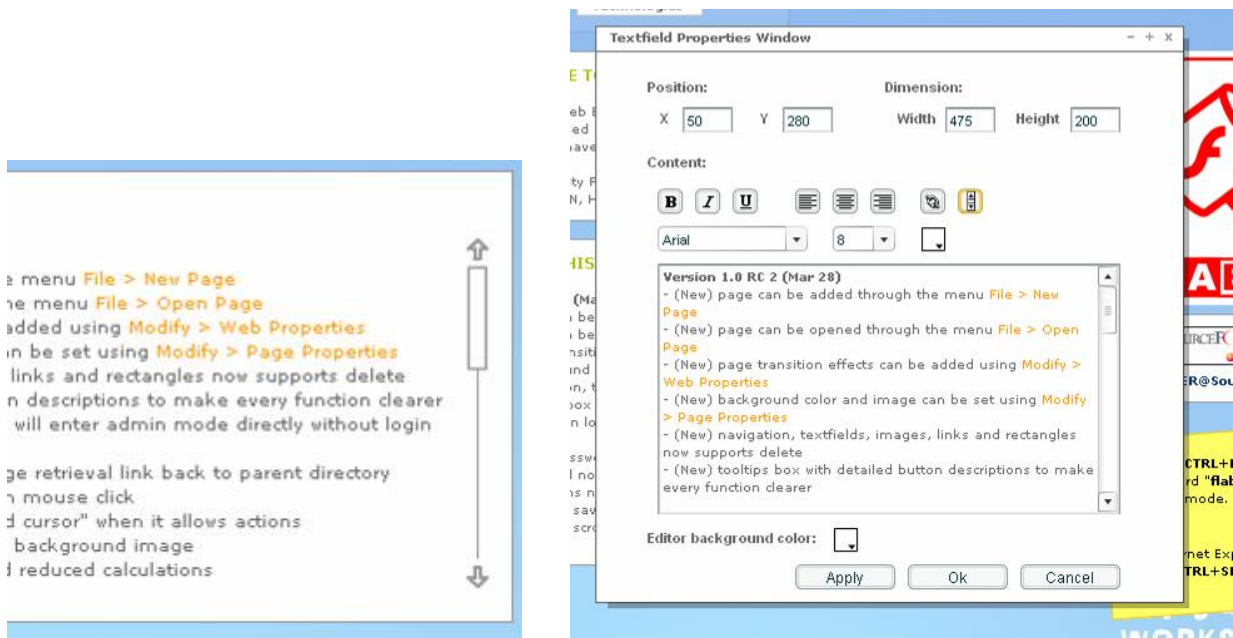


Screenshot 3.18 - Page properties window with background image selected

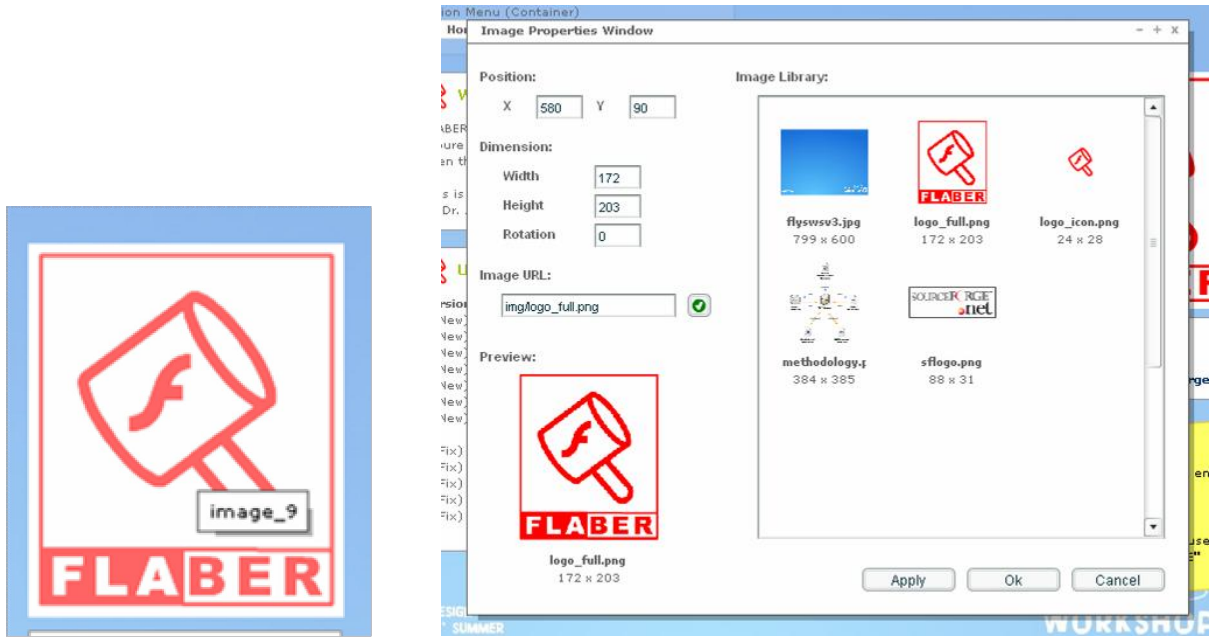
3.4.6. Page Elements

Page elements are the objects placed on the web pages as page contents. Taking reference of traditional XHTML, FLABER currently supports four kinds of page elements including text field, image, link and rectangle. It is pretty easy to understand the first three elements which are important to make a web page. For rectangle, it is mainly used as page design or decorations, e.g. a rectangle box surrounding a text field will results a text box area.

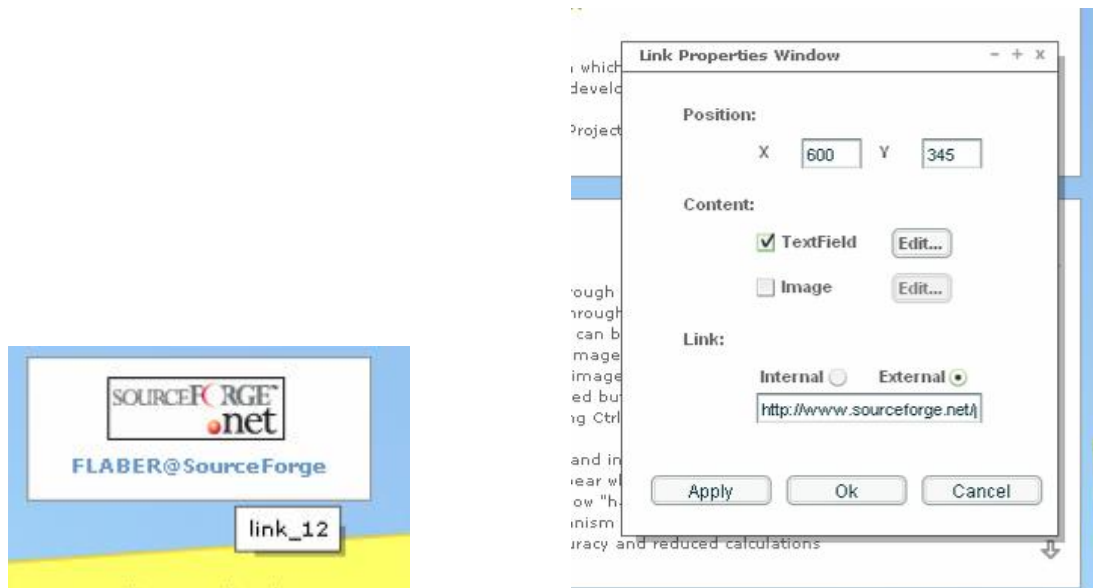
Each page element have its own properties window for the user to manipulate the objects, screenshots are available in the following for illustration purpose.



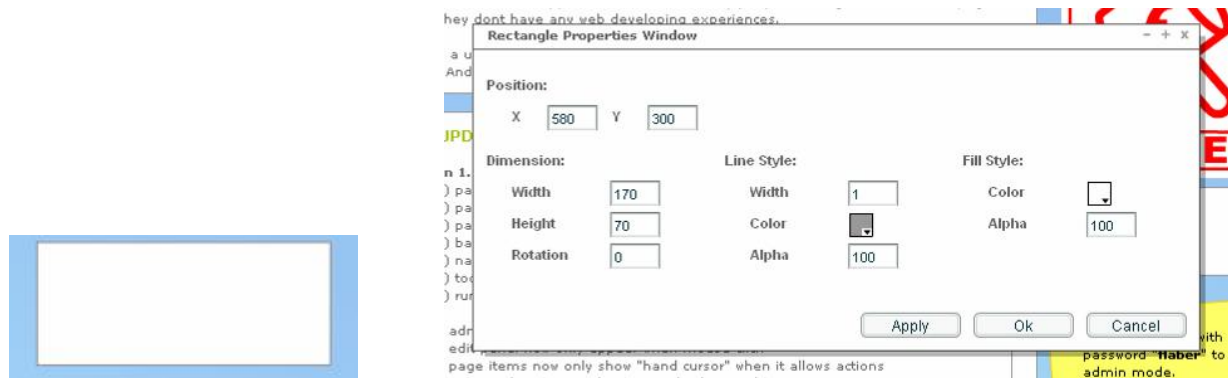
Screenshot 3.19 - Sample text field with scroll bar (left) and text field properties window (right)



Screenshot 3.20 - Sample image (left) and image properties window (right)



Screenshot 3.21 - Sample link (left) and link properties window (right)



Screenshot 3.22 - Sample rectangle (left) and rectangle properties window (right)



Screenshot 3.23 - A rotated rectangle with a text field combined to act like a sticky note

4. RESULTS

In order to evaluate whether FLABER is a success or failure, a series of post development work have to be done. This includes feature set evaluation, performance evaluation and user evaluation.

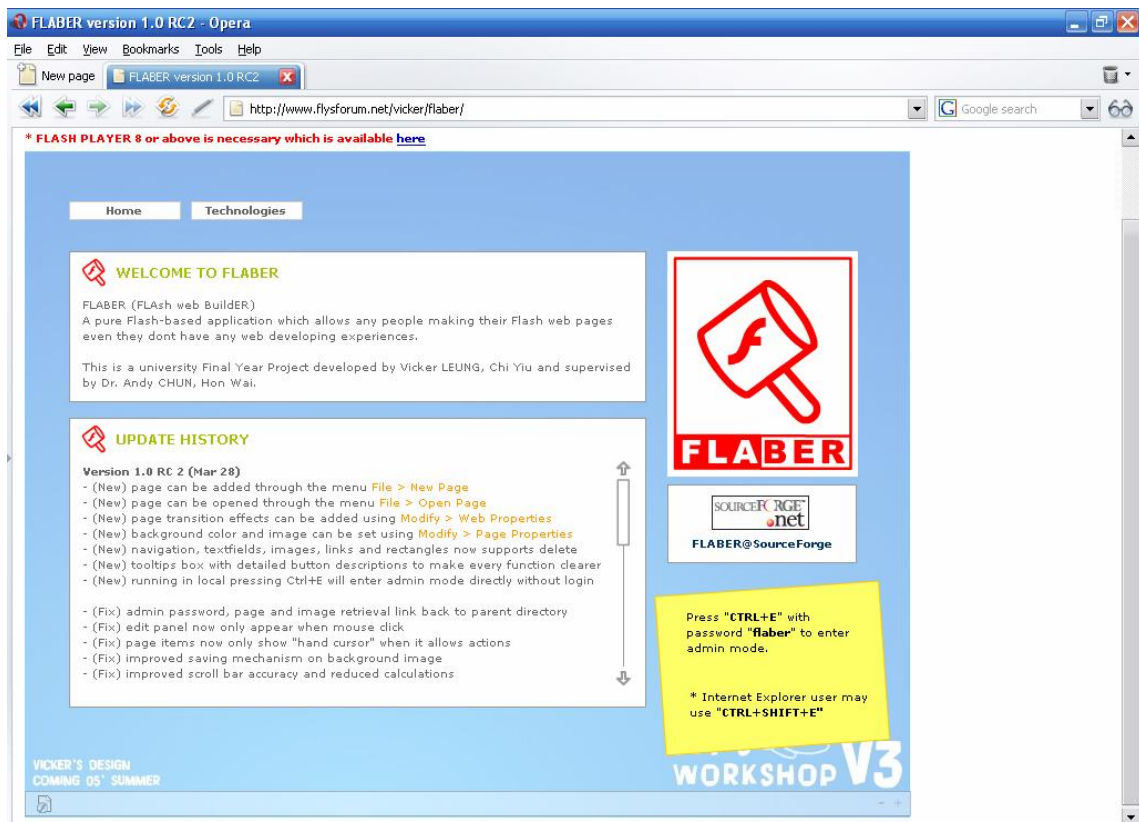
The beginning of this chapter will be a feature set evaluation followed by the performance evaluation using some sample web pages. And in the remaining section, three user testing are reported illustrating how Flash developer, college student and primary student use and think about FLABER.



4.1. Feature Set Evaluation

To ensure that FLABER has implemented all the features necessary to build a web site, it is used to build up a web site for testing purpose. The theme of this testing web site is the official web for FLABER deployed in SourceForge and FLYSforum. This web is used to provide the basic information about FLABER, latest updates and the system architecture. Moreover it is used as live demonstration.

The web is built successfully within two hours which consists of a total of two pages. Basically all the required elements can be done successfully. However some of the features are not achievable, which includes bullet point, un-filled rectangle and rotated text. Bullet point is very important when writing a list of items in a clear manner. Un-filled rectangle and rotated text can give more decorative features to the web site.



Screenshot 4.1 - Official FLABER web site deployed in FLYSforum [32]

4.2. Performance Evaluation

Resource consumption and page loading time is a very crucial factor of a successful web application. In order to test whether FLABER have a good performance comparing with the traditional web page, a FLABER version of the official web site of City University of Hong Kong Department of Computer Science (CityU CS) is made. It is used to compare with the original web site in terms of file size and loading time.

The first testing is done using web page analyzer provided by WebSiteOptimization.com [33]. A summary of the report is illustrated in the following tables.

	Official web of CityU CS [34]	FLABER version of CityU CS [35]
HTML:	9089	2261
HTML Images:	95029	0
CSS Images:	14687	0
Total Images:	109716	0
Javascript:	0	0
CSS:	0	0
Multimedia:	0	128618
Other:	0	0
Total: (in byte)	118805	130879

Table 4.2 - Comparison of the page size between the web page in standard XHTML and FLABER

	Official web of CityU CS [34]	FLABER version of CityU CS [35]
14.4K	92.08 seconds	101.44 seconds
28.8K	46.04 seconds	50.72 seconds
33.6K	39.46 seconds	43.47 seconds
56K	23.68 seconds	26.08 seconds
ISDN 128K	7.25 seconds	7.99 seconds
T1 1.44Mbps	0.63 seconds	0.69 seconds

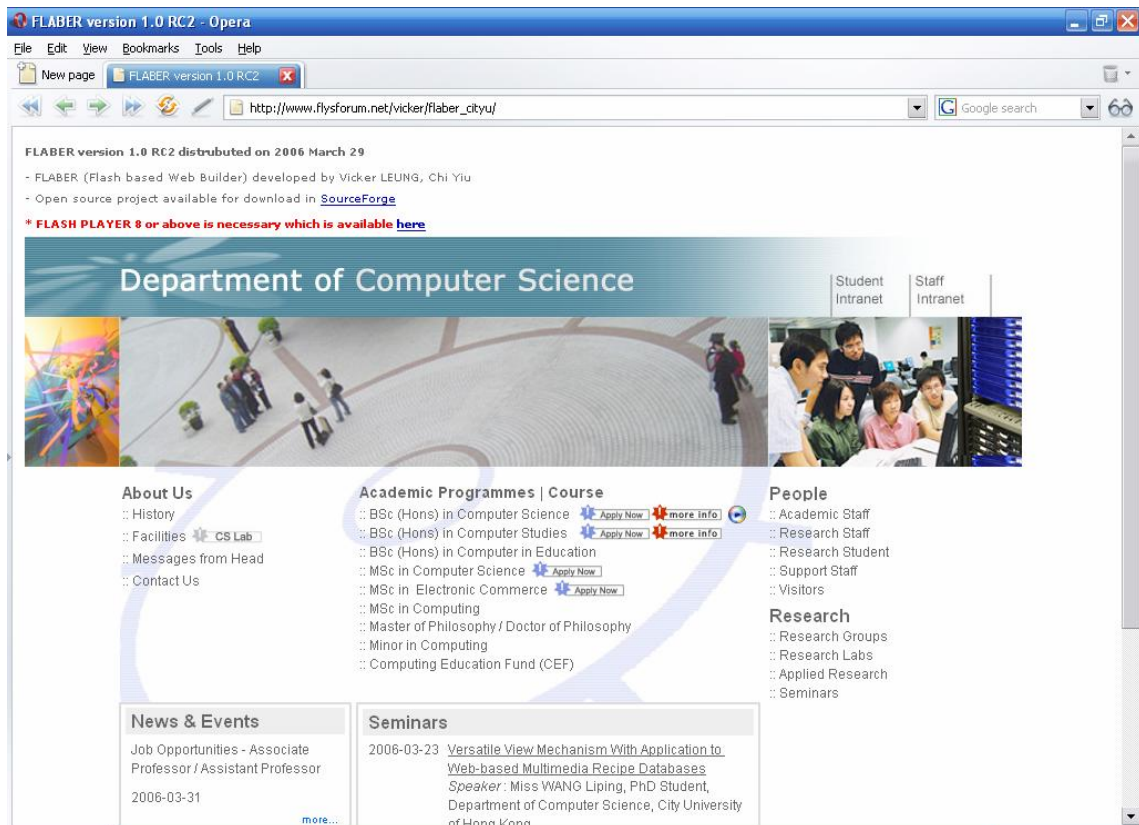
Table 4.3 - Comparison of the page load time between the web page in standard XHTML and FLABER

A second testing is done manually to check the actual time required when a user is browsing using their internet browsers. The time is counted from the user pressing the “Enter” key to the complete loading of the page content. The results are listed in the table below.

	Official web of CityU CS [34]	FLABER version of CityU CS [35]
First Visit (without any cached item)	2.81 seconds	5.04 seconds
Revisiting (with cached image)	1.35 seconds	1.37 seconds

Table 4.4 - Comparison of the page load time between the web page in standard XHTML and FLABER

From the resulting data, it is obvious that although FLABER have a large resource consumption than the traditional XHTML approach. This difference is actually not significant. The detailed explanation on the results will be mentioned in the discussion section in the next chapter.



Screenshot 4.5 - FLABER version of CityU CS for performance evaluation [35]

4.3. User Evaluation by Warenix Wong

User experience and their feedback is another important factor in product evaluation. Because FLABER have a great relationship with Adobe Flash, it is very important to hear what an experienced Flash developer thinks about FLABER. That is why Warenix WONG Chi Fung, a twenty-one years old university student with five years Flash development experience is invited to be one of the user to evaluate FLABER.

He is requested to build a free topic web site using FLABER. During the testing progress, his immediate feedback, comment and query are noted. The following are the points suggested by him. (Translated from Chinese)

- ◆ Depth control interface is absent
- ◆ Image upload is not possible
- ◆ Don't have the traditional "Save As..." function
- ◆ Bullet points for text is important which is not available
- ◆ Only rectangle shape is not enough
- ◆ Can drawing tools present for user to draw something?
- ◆ If grouping of item is possible, it will give great convenience to others
- ◆ Wallpaper can not tile like Microsoft Windows
- ◆ Fine adjustment to resizing is difficult
- ◆ Image loading should show loading progress
- ◆ No tool tips available in the text editing panel
- ◆ Moving, resizing is not direct manipulation like Microsoft Windows
- ◆ Some form labels in the user interface is not properly named

Afterwards, his general comment about FLABER is given as below.

Easy to use and learn. Fast to achieve result. Working with FLABER has great fun. Most of the functions that required in building a web are implemented. However some complicated functions like scripting can not be done. Suggested to add more available shapes or even drawing tools which allow the user to design more. If compared FLABER with Flash, I will choose Flash to build up web pages because it allows me to add ActionScripts.



Screenshot 4.6 - "Warenix in FLABER" [36] which is done in around two hours

4.4. User Evaluation by Soki Ho

In the previous section, a Flash developer is chosen to evaluate FLABER. But how will a user with no Flash experience think about FLABER? Soki HO Sin Yi is a fifteen years old secondary five college student with no Flash experience and two years of Adobe Dreamweaver experience. She was invited to do the same user evaluation as Warenix. The following are her feedbacks during the evaluation. (Once again it is translated from Chinese)

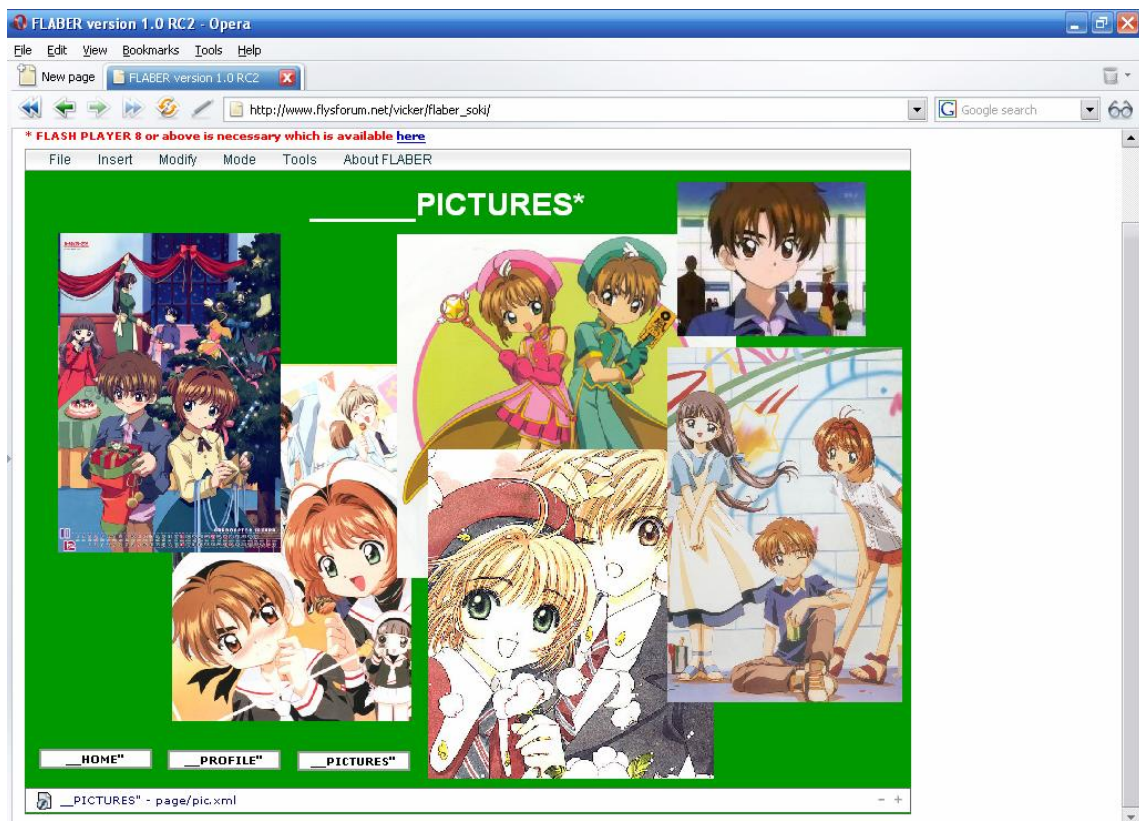
- ◆ Don't have loading progress when opening new page is confusing
- ◆ It is not convenient to use the edit panel to do the moving, traditional moving approach in Microsoft Windows is proposed
- ◆ Sometimes cannot locate the new created item, making them larger will be better
- ◆ It is not possible to make image semi-transparent

The general comment is stated as follow.

The feature set is not enough, especially it is disappointed that music cannot be added. However using FLABER makes the web building process much easier and it is not necessary to care about the HTML. This is obvious when trying to put several images overlapping each other. When using Dreamweaver, this task makes me crazy. Because Flash is too difficult to be used, I will choose FLABER to build up my web page.



Screenshot 4.7 - "Soki in FLABER" [37] which is done within one and a half hour



Screenshot 4.8 - A sample page with overlapping images which can be done in FLABER

4.5. User Evaluation by Christy Chung

Internet is spreading to kids nowadays, even primary students have their own email address and personal web pages. The largest obstacle to them during web development mainly is to tackle the complex user interface in the authoring software. A ten years old primary four student Christy CHUNG Lok Ling is invited to play with FLABER. She is asked to build up her favorite Pokemon web site under guidance. (Explaining those English user interface) The following are some of the points observed during the evaluation progress.

- ◆ She have great trouble with inserting new items due to the English interface
- ◆ She knows how to do the typing and have fun playing with drag and resize
- ◆ She found difficulties to locate the newly created text

Her general comment is given as below. (Arranged and translated from Chinese)

It is very funny playing around with FLABER. But why the size of FLABER is so small, why not making it like the size of the browser so that I can add even more images.



Screenshot 4.9 - "Christy in FLABER" [38] which is done within an hour

5. DISCUSSION

Last chapter I have provided several testing for FLABER, in terms of the feature set, performance and user experience. So what is the reason behind the findings?

In the chapter, I will explain the data collected in the performance evaluation section followed by comparison between FLABER and other related applications. And at last I will state the benefits, limitations and applications of FLABER.



5.1. Explanation on Performance Evaluation

In chapter 4.2, it is clear to observe that the page size of a web page created by FLABER (130879 bytes) is slightly larger than the page created by XHTML (118805 bytes).

This can be explained by the fact that in the page produced by FLABER, it contains some additional positioning information which is not present in XHTML. This information mainly includes the x and y coordinates of every page element. Although it contains only 20 characters or less which is not significant, the large amount of page elements will produce a multiplication effect resulting the difference of 12074 bytes. Luckily this 12KB is not very significant in the current broadband environment.

The manual performance testing shows an interesting result that during the first visit, FLABER (5.04 seconds) is much slower than XHTML (2.81 seconds). However in revisiting, FLABER (1.37 seconds) stands very close with XHTML (1.35 seconds).

This is because during the first visit, additional page loading time is required to load the shockwave flash file (swf) of FLABER. After the first visit the swf file is cached in the browser, that is why during revisiting, the speed of FLABER is much faster. The remaining loading time is used in loading the XML data file.

5.2. Comparison with Related Applications

After the user evaluation section described in the last chapter, Warenix and Soki is further invited to use some applications related to FLABER, which includes b82cms, fCMS and FlashBlocks. (Mentioned in chapter 2.7) And they are requested to give some comparison between those applications and FLABER.

The following is a summarization of all the comments.

- ◆ FLABER is comparatively more flexible, allowing the user to design and develop freely. Other applications are so rigid that it seems like a template editing environment instead of web authoring environment.
- ◆ b82cms have a great animated user interface design which is very vivid and attractive, and this is lacked in FLABER.
- ◆ fCMS have a much larger feature set in font editing which is great. However sometimes it is too much which will make the user interface too crowded and distracting. The loading time of the editing panel is too long.
- ◆ FlashBlock have developed some extensions like mp3 player and form filling which enrich the application very much. The user interface is not designed well and without a standard which is not easy to use.

5.3. Benefits

The major benefit that FLABER provided is that any web developers can build web pages even they don't have any web developing experience. Since the web pages are Flash-based, they can enjoy the great multimedia experience provided without the worries that different browser will have different behavior. Although it is Flash-based, it can be crawled by any search engines since all the data is stored in XML.

And of course having fun is one of the elements that FLABER provides 😊.

5.4. Limitations

A limitation which is very critical is that PHP enabled server is a must in the deploying environment of FLABER. FLABER can be used to browse locally but not editing. Flash player version 8 or above is essential for FLABER to operate normally.

5.5. Applications

FLABER can be used in different areas for different purposes. It can be used by any web owners to build up their own web pages like blog, online diaries or even photo gallery. It can be used by some small industries to build up their company web sites for marketing purposes. It can be used even by teachers to build up some presentation slides for educational purposes with the page transition ability of FLABER.

So FLABER can be used very widely and freely, depending on user's imagination.

6. CONCLUSION

This is the last chapter of this report which is used to wrap up the whole final year report.

First I will give a self review regarding this one year final year project. And then I will state out all the achievements made by FLABER. Lastly I will suggest the future development and extension of FLABER as a full stop of this report.



6.1. Critical Review

First of all I really want to take a deep breathe and say that “Woo... I did it. I withstand this whole year!” Many things are my very first time within this final year project. For example, the first time I write an application a whole year from scratch. The first time I write Flash ActionScripts in object-oriented way which is my weaknesses. The first time I release my work to the public and more. Although it is not easy to walk this one year, it is valuable.

The greatest failure in this project is that it is not within schedule. The main reason of this is that I have overestimated my time available and underestimated the time required for debugging during development. I have not considered well in the time allocated to other course and assignment, that is why starting from the milestone of the first prototype release, all the remaining tasks are out of schedule. Luckily, all the tasks required and stated are completed at last. Next time if I have another project to take up, I will try my very best to put more effort in scheduling.

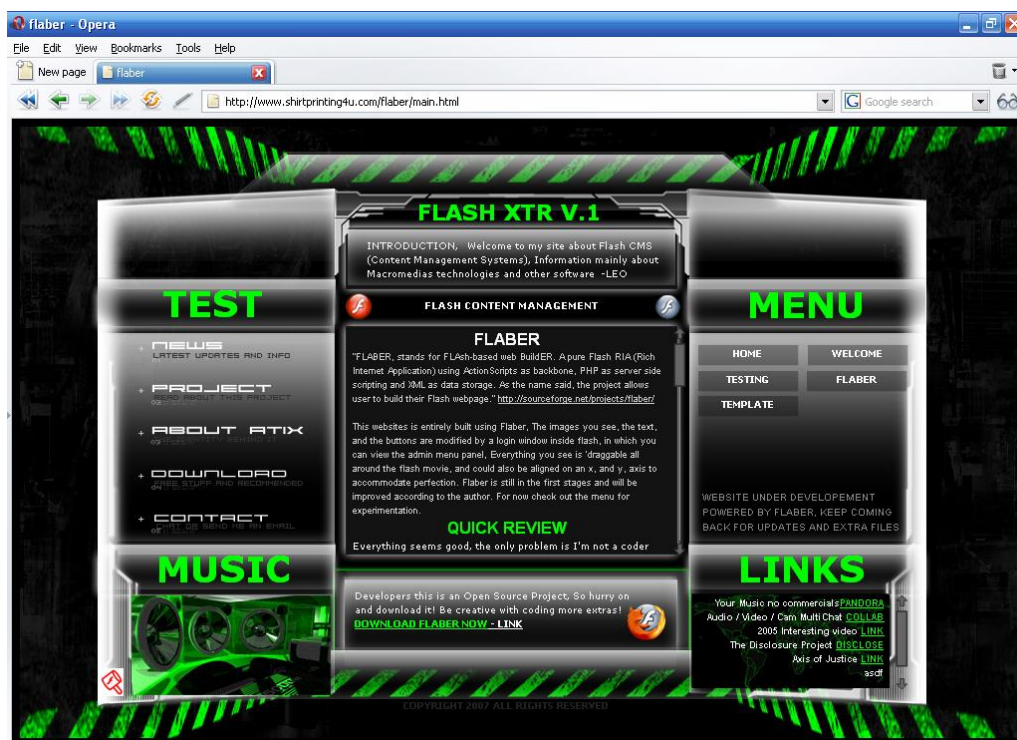
During this project, I have made many new relationships with other people. First of all is my great supervisor Dr. Andy CHUN. Others include Vesselin Drangajov (a web hosting company manager from Europe), John Koch (developer relations manager from Adobe Systems), and many FLABER supporters around the world. I learned much from all of them by sharing my ideas and listening from them. These relationships are more valuable than anything else which I will keep it forever.

6.2. Achievement

In order to keep track how the public react with the idea of FLABER. On November 6th, 2005, FLABER is first hosted in SourceForge as one of the web development project under General Public License (GPL). [38] Originally the traffic and feedback are not very active, however starting from the first public release of FLABER, the traffic grows a lot. Till now there are a total of 2,002 page hits and 424 downloads, ranking 1,555 out of 133,024 projects hosted in SourceForge. (Detailed statistics are provided in Appendices)

On April 4th, 2006, I received an email from Vesselin Drangajov (a web hosting company manager from Europe) who have great interests in FLABER, and want to ask for my permission allowing him to distribute FLABER to his customers. (Copy of the email is provided in Appendices)

I also receive quite a number of comments and supports from peoples around the world. One of the supporter atixive from SourceForge has even made a marvelous web site using FLABER which really utilizes the full power of it.



Screenshot 6.1 - Flash XTR [39] which is done by the first FLABER supporter - atixive

6.3. Future Extension

The great response and feedbacks from the public really pushes me a lot. So I have already decided to further develop FLABER even after this final year project. Actually there are numerous extensions and improvements available for FLABER.

The first one is the fully developed XHTML converter. Not all the users like to browse a web page in Flash, some of them like to browse in XHTML. Furthermore, search engines do have a better crawling support with HTML files than XML files. That is why if FLABER can export XHTML at the same time with a single click. It will bring up large convenience and usability to the internet world.

The second item that can extend FLABER further is the MySQL database storage for page indexing. Right now all the data is stored in XML format which is not easy to do any data manipulation, e.g. searching for a specific topic or word. If a MySQL database is plugged into FLABER, page indexing will become possible which allows functions like searching.

The last item that is greatly requested by the public is the possibilities of installing components to FLABER just like plug-ins. Examples include the music player, guestbook, user registration, and even more. If the development of component is a success, developers around the world will be able to contribute to FLABER by developing their own components which will be a both win situation to FLABER and the end users.

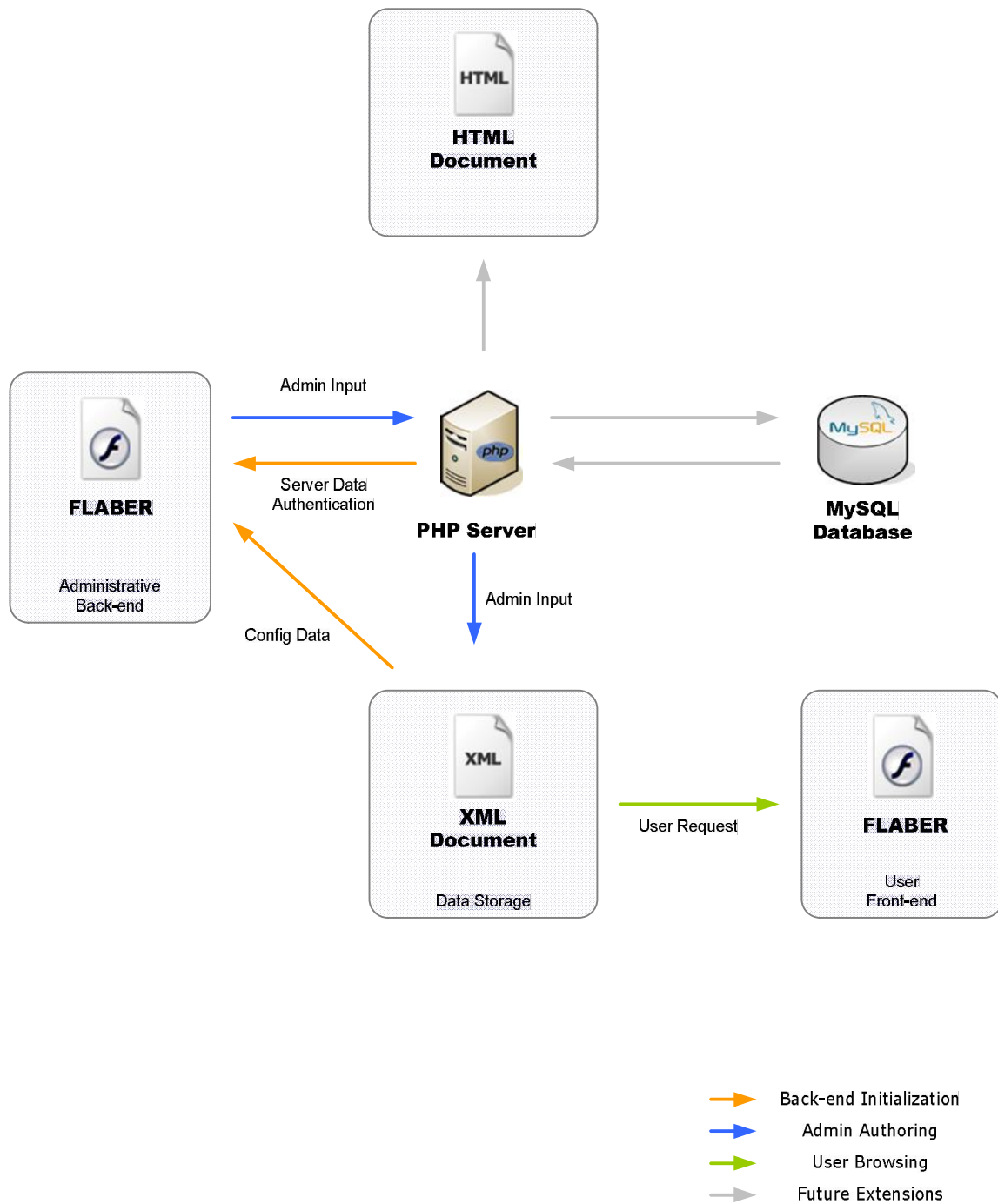


Diagram 6.2 - Possible future system design of FLABER

REFERENCES



Cited References

- [1] A Comparison of two major dynamic web platforms (LAMP vs. WISA). [Online]. Penry, A. Available: <http://www.shawnolson.net/a/302>
- [2] Netcraft Web Server Survey March 2006 - By Server. [Online]. Netcraft LTD. Available: <http://survey.netcraft.com/Reports/200603/byserver/index.html>
- [3] A Look at Rich Internet Application. [Online]. O'Rourke, C. Available: <http://www.carlsbadcubes.com/theodore/doc/richinetapp.pdf>
- [4] Rich Internet Applications and AJAX - Selecting the best product. [Online]. Domenig, M. Available: <http://www.javalobby.org/articles/ajax-ria-overview>
- [5] Ajax: A New Approach to Web Applications. [Online]. Garrett, J. J. Available: <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [6] Macromedia Flash content reaches 97.7% of Internet viewers. [Online]. Adobe Systems Incorporated. Available: http://www.macromedia.com/software/player_census/flashplayer
- [7] minishowcase. [Online]. frwr.net. Available: <http://www.frwr.net/minishowcase>
- [8] ImageGallery. [Online]. Gabriel. Available: <http://www.geocities.com/SiliconValley/Way/8870/image.htm>
- [9] Image Browser. [Online]. Crugnola, A. Available: http://www.sephiroth.it/test/PHP_remoting/imagebrowser
- [10] Macromedia Flash. [Online]. Macromedia, Inc. Available: <http://www.macromedia.com/software/flash/flashpro>
- [11] Macromedia Flex. [Online]. Macromedia, Inc. Available: <http://www.macromedia.com/software/flex>
- [12] OpenLaszlo, [Online]. Laszlo Systems, Inc. Available: <http://www.openlaszlo.org>
- [13] Neovision. [Online]. Amanagroup. Available: <http://neovision.jp>
- [14] Flex Store. [Online]. Macromedia, Inc. Available: <http://flexapps.macromedia.com/flex15/flexstore/flexstore.mxml>
- [15] Laszlo Calendar. [Online]. Laszlo Systems, Inc. Available: <http://www.laszlo.com/lps/sample-apps/calendar/calendar.lzo?lzt=html>
- [16] Weblog. [Online]. Wikipedia. Available: <http://en.wikipedia.org/wiki/Weblog>
- [17] Blogger. [Online]. Google. Available: <http://www.blogger.com>
- [18] Xanga. [Online]. Xanga. Available: <http://www.xanga.com>
- [19] VIC@FYP. [Online]. VIC@FYP. Available: <http://www.flysforum.net/vicafyp>

- [20] Webopedia. [Online]. Jupitermedia Corporation. Available: http://www.webopedia.com/TERM/W/Web_authoring.html
- [21] Yahoo! Geocities. [Online]. Yahoo! Inc. Available: <http://geocities.yahoo.com>
- [22] Google Page Creator. [Online]. Google. Available: <http://pages.google.com>
- [23] b82CMS. [Online]. Beaver82. Available: <http://cms.beaver82.com>
- [24] fCMS. [Online]. FFF Web Media Inc. Available: <http://www.flashloaded.com/flashcomponents/fcms>
- [25] Flashblocks. [Online]. Flashblocks. Available: <http://www.flashblocks.com>
- [26] Embedded font trouble. [Online]. Flashblocks forum. Available: http://www.flashblocks.com/index.php?option=com_simpleboard&Itemid=41&func=view&id=439&catid=4
- [27] Flash Decompiler 2.9. [Online]. Eltima Software GmbH. Available: <http://www.eltima.com/products/flashdecompiler>
- [28] ASO Lite. [Online]. Genable Labs. Available: <http://www.genable.com/asolite.html>
- [29] Simple design patterns in Macromedia Flash MX 2004. [Online]. Parente, P. Available: <http://www.cs.unc.edu/~parente/tech/tr03.shtml>
- [30] Moock, C. (2004). Essential ActionScript 2.0. O'Reilly. 352 - 380.
- [31] Shneiderman, B. (1998). Designing the User Interface. Addison Wesley. 124 - 150.
- [32] FLABER. [Online]. FLABER 1.0 RC2. Available: <http://www.flysforum.net/vicker/flaber>
- [33] Web Page Analyzer. [Online]. Web Site Optimization, LLC. Available: <http://www.websiteoptimization.com/services/analyze>
- [34] City University of Hong Kong Department of Computer Science. [Online]. City University of Hong Kong Department of Computer Science. Available: <http://www.cs.cityu.edu.hk>
- [35] CityU CS in FLABER. [Online]. FLABER 1.0 RC2. Available: http://www.flysforum.net/vicker/flaber_cityu
- [36] Warenix in FLABER. [Online]. FLABER 1.0 RC2. Available: http://www.flysforum.net/vicker/flaber_warenix
- [37] Soki in FLABER. [Online]. FLABER 1.0 RC2. Available: http://www.flysforum.net/vicker/flaber_soki
- [38] FLABER. [Online]. SourceForge. Available: <http://www.sourceforge.net/projects/flaber>
- [39] Flash XTR. [Online]. FLABER 1.0 RC2. Available: <http://shirtprinting4u.com/flaber/index.html>

Additional References (Books)

Franklin, D. and Makar, J. (2003). Macromedia Flash MX 2004 ActionScript: Training from the Source. Macromedia Press.

Jacobson, D. and Jacobson, J. (2001). Flash and XML: A Developer's Guide. Addison Wesley.

Lee, J., Ware, B. (2002). Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP. Addison Wesley.

Moock, C. (2004). Essential ActionScript 2.0. O'Reilly.

Ray, E.T. (2003). Learning XML. 2nd ed. O'Reilly.

Additional References (Webs)

Actionscript.org. [Online]. Actionscript.org. Available: <http://www.actionscript.org>

Actionscript optimizations tutorial. [Online]. Lapi, M. Available: http://www.gotoandplay.it/articles/2004/01/as_optimizations.php

Dr. Andy Chun – Student Projects. [Online]. Chun, H.W. Available: <http://www.cs.cityu.edu.hk/~hwchun/FYP>

Experts Exchange. [Online]. Experts Exchange LLC. Available: <http://www.experts-exchange.com>

Flash MX and the Bigger Picture: Lightweight Internet Application. [Online]. Appnel, T. Available: http://www.oreillynet.com/pub/a/javascript/2002/09/05/flash_runtime.html

ONLamp.com. [Online]. O'Reilly Media, Inc. Available: <http://www.onlamp.com>

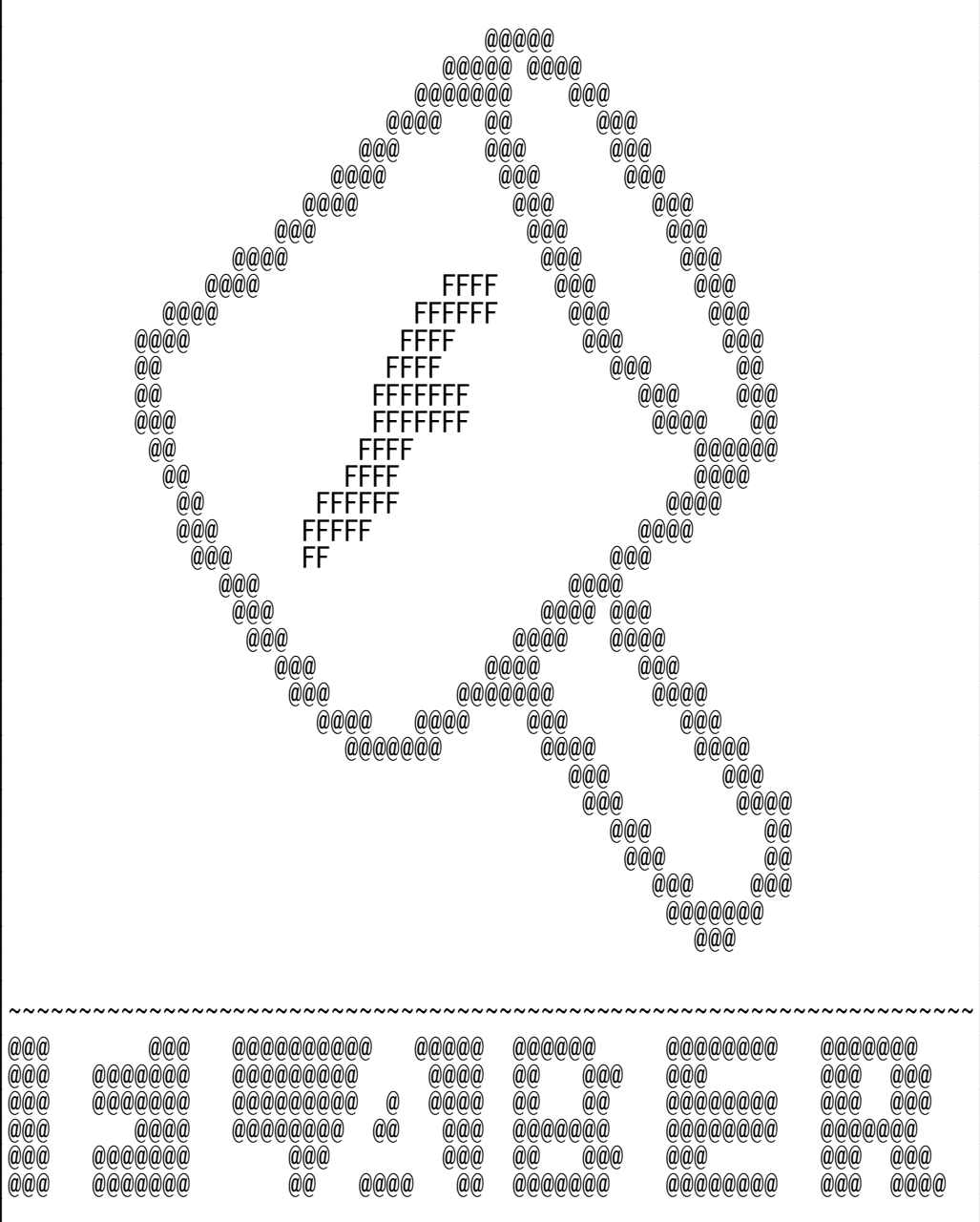
W3Schools Online Web Tutorials. [Online]. Refsnes Data. Available: <http://www.w3schools.com>

World Wide Web Consortium. [Online]. W3C. Available: <http://www.w3c.org>

APPENDICES



Appendices A. User Documentation of FLABER 1.0 RC2



000 000 0000000000 00000 000000 00000000 00000000
000 00000000 0000000000 0000 00 000 000 000 000 000
000 00000000 0000000000 0 0000 00 00 00000000 00000000
000 00000 00000000 00 000 00000000 0000000000 00000000
000 00000000 000 000 00 00 000 000 000 000 000
000 00000000 00 0000 00 00000000 0000000000 000 0000

```

~~~~~
FLABER (FLAsh-based web BuildER)
  Flash it! Drag it! Build it!

SourceForge Project Page
  http://sourceforge.net/projects/flaber

SourceForge Demo Page
  http://flaber.sourceforge.net/

Flysforum Demo Page (Faster connection)
  http://www.flysforum.net/vicker/flaber

```

```

~~~~~
FLABER version 1.0 RC2 User Documentation

```

* Just use the find function using the keys, e.g. "C." to jump to a particular section.

- ```

~~~~~
A.  Author's Voice
B.  Why FLABER?
C.  System Requirements
D.  Installation
E.  Using FLABER
    E. a.  Action Mode
    E. b.  Edit Mode
    E. c.  Admin Login
    E. d.  Add and Edit Items
    E. e.  Page Properties
    E. f.  Web Properties
    E. g.  New / Open and Saving
    E. h.  XML editing
F.  Contact
G.  Acknowledgement

```

```

~~~~~
A. Author's Voice

```

Is been a long time since I started this project. Finally I have a chance to present the very first release. Counting my fingers, this project is running for nearly a year already. I wont say that this product is great, however I will say that you will have fun with it!

Vicker LEUNG @ 2006 Mar 16  
Student of City University of Hong Kong, SAR

## B. Why FLABER?

FLABER is a pure ActionScripts 2 driven Flash Rich Internet Application (RIA), while the server side scriptings relies on PHP, data storage mainly using XML.

This project is first started as my personal final year project. (University homework... in other words =.=) And when I developed it further, I want to make it an open source product. So now this project is hosted in SourceForge under General Public License (GPL).

The main usage of FLABER is to allow any people with different skillsets, even you know nothing about Flash can still build up your own personal Flash-based website.

## C. System Requirements

Server side:  
IIS / Apache with PHP 4.0 or above

Client side:  
Flash Player 8 is a must

## D. Installation

- Go to download the latest package of FLABER from <http://sourceforge.net/projects/flaber>
- Unzip the package
- Copying the whole FLABER folder to your personal web space
- Change the file and folder permission as follows

```
flaber/
 function/
 img/ 777 with folder contents 666
 page/ 777 with folder contents 666
 style/
 index.html
 flaber.swf
 Flaber.xml 666
 MenuBar.xml
 NavigationMenu.xml 666
```

- Enjoy using FLABER by browsing to the [http://installation\\_path/flaber/index.html](http://installation_path/flaber/index.html)

~~~~~  
E. Using FLABER

- Apolgy first I dont have much time to prepare a good documentation. I will improve in the forthcoming releases

~~~~~  
E. a. Action Mode

There are actually two main viewpoint of FLABER, "Action Mode" and "Edit Mode".

Action mode is the mode that is in default when anyone browse FLABER. So this can be also called the browsing mode. In this mode, you can browse the web just like any normal web page. Text, Images, Shapes, Links and Navigation Menus are all available.

~~~~~  
E. b. Edit Mode

When you are in edit mode, all the elements is editable. You can try to move your mouse over any element and a small panel will be placed near the element which is called the "Edit Panel".

In order to change the mode, you need admin login which will be mentioned in next paragraph.

When you are logged in as an admin, you can enter edit mode by the menu item "Mode"

Mode > Edit Mode

~~~~~  
E. c. Admin login

You can enter the admin login by:

Pressing "CTRL + E"

\*\*\*\*\*  
\* Internet Explorer user may experience problem that "CTRL + E" \*  
\* is mapped to other shortcut function. Try "SHIFT + CTRL + E" \*  
\*\*\*\*\*

The default password is "flaber" without the double quotes.

\*\*\*\*\*  
\* You can change the password by editing flaber/function/admin\_login.php \*  
\*\*\*\*\*

After login, you will notice that a new menu bar come out, here you can access numerous functions including mode change.

```

~~~~~
E. d. Add and Edit Items
~~~~~

```

Currently FLABER supports the following page items:

- TextField
- Image
- Links
- Shapes (Rectangles only)

Adding new element is simple, you can use the "Insert" menu bars  
A window will bring up asking for the item details

e.g. Insert > TextField

Editing existing element will require "Edit Mode". (mentioned in chapter E.b.)

After changing to edit mode, you can left click on any elements to bring up a little panel called edit panel which have numerous editing functions:  
(from left to right)

```

~~~~~
Move | Resize | Rotate | Properties | Delete
~~~~~

```

|            |                                                  |                         |
|------------|--------------------------------------------------|-------------------------|
| Move       | Left click and drag to move the item             |                         |
| Resize     | Left click and drag to resize the item           | (not all items support) |
| Rotate     | Left click and drag to rotate the item           | (not all items support) |
| Properties | Left click will bring up an advance dialogue box |                         |
| Delete     | Left click will remove the item                  |                         |

```

~~~~~
E. e. Page Properties
~~~~~

```

Page properties contain all the settings according to the current opening page, including

- Background Color
- Background Image

Page Properties are accessible through the menu item "Modify"

Modify > Page Properties

~~~~~  
E. f. Web Properties  
~~~~~

Web properties are similar as page properties, however the target is on the whole web site. Functions includes

- Index Page (that is the first page that FLABER will load when a visitor comes)
- Navigation Menu (whether a navigation menu will be used throughout the web)
- Page Transitions (special animation effects when there are page changes)

Web Properties are accessible through the menu item "Modify"

Modify > Web Properties

~~~~~  
E. g. New / Open and Saving  
~~~~~

All of these functions can be found through the menu item "File".

!!!!!!!!!! ALWAYS REMEMBER TO SAVE THE CONTENT BEFORE EXITING THE WEB !!!!!!!!!!!

```

~~~~~
E.h. XML Editing
~~~~~

```

Actually FLABER is still in development, some of the functions are still not editable through the GUI. For example

- Depth changing
- Image uploading

And some expert users may like editing codes rather than using the GUI. In this case, you are always welcome to edit the XML files directly.

```

!!!!!!!!!! For non-expert users, try not to edit the XML directly !!!!!!!!!!!
!!!!!!!!!! which may results unbelievable damage to the FLABER data !!!!!!!!!!!

```

The XML files are placed in the following manner:

```

flaber/
 function/
 img/
 page/ all the seperate pages, you can see it like a single html file
 index.xml
 page2.xml
 style/
 index.html
 flaber.swf
 Flaber.xml the so called global config file, controlling the page transitions etc...
 MenuBar.xml the menu bar. no fun in here so I dont guess you will need to edit this
 NavigationMenu.xml the navigation menu which will appear on EVERY page of the web

```

```

~~~~~
F.  Contacts
~~~~~

```

For general issues, such as suggestions making, bug reporting, commenting, donation etc. Please kindly move one more step to the SourceForge project forum.

[http://sourceforge.net/forum/?group\\_id=152518](http://sourceforge.net/forum/?group_id=152518)

Your support is always welcome and valuable for me to develop even more products.

For other issues, such as joining me as the development team. Please dont hesitate to contact me directly through email.

vicker[at]gmail.com

ICQ also welcomes at 25264936.

```
~~~~~
G. Acknowledgement
~~~~~
```

Here, I grab the chance to thank some of the people who helped me a lot in the succeed in this project.

<< Dr. Andy CHUN, Hon Wai >>

Professor in City University of Hong Kong.  
Experts in Artificial Intelligence and Arts.  
He is a great professor who provided lots of knowledge and suggestions to me.

<< Mr. Tim SHIU, Ka Kei >>

A friend of mine and a great programmer in many different languages.  
He have a very rigid logic and thinking which really helps me a lot during development, especially those OO concepts.

<< Ms. Eva SHI, Yi Fan >>

My girlfriend and also my first-line tester.  
I already forgot how many times she sitting besides me watching me typing all those crazy Flash codes.  
Sorry about that and I love you :)

<< Ms. Soki HO >>

College student and one of my user tester.  
Responsible to test whether the product is usable for user who have no Flash experience.

<< Special thanks to >>

OOHK                      Project hosting  
SourceForge              Project hosting  
Macromedia Flash  
SEPY ActionScript Editor

Carmen  
Kay CHEUNG  
Michael STOCK  
Olivia  
Sam LAM

### Appendices B. Statistical Data of FLABER at SourceForge

(Live data available at [http://sourceforge.net/project/stats/?group\\_id=152518&ugn=flaber](http://sourceforge.net/project/stats/?group_id=152518&ugn=flaber))

**Daily Ranking of FLABER in SourceForge (latest 2 months)**



| Date (UTC)         | Rank    | Total Pages | Downloads | Project Web Hits |
|--------------------|---------|-------------|-----------|------------------|
| Apr 2006           | 2,372   | 914         | 222       | 1,132            |
| Mar 2006           | 54,009  | 1,853       | 204       | 363              |
| Feb 2006           | 76,002  | 65          | 0         | 118              |
| Jan 2006           | 55,194  | 135         | 0         | 145              |
| Dec 2005           | 52,806  | 114         | 0         | 142              |
| Nov 2005           | 144,916 | 131         | 0         | 102              |
| <b>Grand Total</b> | 1,555   | 3,212       | 424       | 2,002            |

**Table A.1 - Monthly statistics of FLABER at SourceForge**

Definition and calculations of the statistics can be found in SourceForge documentation [http://sourceforge.net/docman/display\\_doc.php?docid=14040&group\\_id=1](http://sourceforge.net/docman/display_doc.php?docid=14040&group_id=1)

**Appendices C. Copy of Email by Vesselin Drangajov dated Apr 4<sup>th</sup>, 2006**

Hello ,

My name is Vesselin and I am the manager of a web hosting company called Host Color. I was browsing Source Forge projects recently and I came across you flabber project.

I must say it is really impressive. In my view as a web developer it stands somewhere in the middle between tinyMCE (a online wysyg editor) and a fully fledged CMS. Using Flash for it however is a genuinely new idea.

I also very much like the philosophy of keeping it presentetion media independent. In this sense I wanted to ask whether you plan to produce a html visualization mode for it?

Also flabber can indeed turn into a CMS (content management system) of a new type. We offer all kinds of GPL software as a free extra to our plans and flabber is something we'd genuinely like to see and use (with your permission of course).

Thank you for your time and good luck.

Vesselin (vesselin@hostcolor.com)

## Appendices D. Project Progress Log

### 2005 October

- ◆ Research on the different developing technologies and combinations, including LAMP, ActionScript, Macromedia Flex and Laszlo.
- ◆ Practicing Object Oriented approach in developing Flash Application.
- ◆ Developed a simple bookmark application to practice XML data fetching.
- ◆ Developed a simple color picker application with well defined window class to act as the future interface template.
- ◆ Built a Blog to let my supervisor know my progress and give comments easily.

### 2005 November

- ◆ Researched on the workflow and user interface of some Flash Content Management System (Flash CMS) including Flash Blocks and b82CMS
- ◆ Designed and sketched the roughwork of the user interface
- ◆ Defined the application name as FLABER (FLAsh-based web BuILDER)
- ◆ Wrote the interim report

### 2005 December

- ◆ Finished the prototype version 0.1  
(Focusing on XML reading and Flash element building referring to XML)
- ◆ Finished the prototype version 0.2  
(Focusing on the Flash element interaction and XML write back)
- ◆ Prototype version 0.3 in progress

## 2006 January

- ◆ Finished the prototype version 0.3  
(Focusing on the message window showing system message / error message)
- ◆ Finished the prototype version 0.4  
(Focusing on the page elements)
- ◆ Release 1 in progress

## 2006 February

- ◆ Finished development of Release 1
- ◆ Testing phase for Release 1
- ◆ Preparing final report

## 2006 March

- ◆ Public release of Release 1 in SourceForge
- ◆ Development of Release 2
- ◆ Preparing final report

## 2006 April

- ◆ Public release of Release 2 in SourceForge
- ◆ User evaluations
- ◆ Performance evaluations
- ◆ Finished final report
- ◆ Development of Release 3