

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

Integration of Flash user interface with PHP and SQL database

Yin Yiming

Thesis of Business Information Technology Degree Program

Bachelor of Business Administration

TORNIO 2012

ABSTRACT

Yin, Yiming 2012. Integration of Flash user interface with PHP and SQL database. Bachelor's thesis. Kemi-Tornio University of Applied sciences. Business and Culture. Pages 40.

This research aims to help Kemi-Tornio University of Applied Sciences in educating and anyone who is interested in integration of Flash user interface with PHP and SQL database.

The first objective of this research is to integrate Flash user interface with PHP and SQL databases. In addition, the second objective is to find out the practicable solution for a PHP web application based on SQL database that presents data via Flash user interface. To accomplish the work, the knowledge of PHP, Flash Action Script 3.0 and SQL database were studied.

As the primary methodology, constructive research is chosen for the whole research process. The findings are implemented practically to come out with an online gallery template for media students at KTUAS to exhibit their works. Interviews were performed among students and staffs in order to increase understanding of their requirements. The literature and documentation reviewed were used to gather information about Flash Action Script 3.0 and PHP.

The outcome of my work is the implemented online gallery template for KTUAS media students and the thesis, as a theoretical report for future research.

Keywords: PHP, SQL, Flash Action Script 3.0, Online Gallery, Flash AS3 and PHP communication

FIGURES

Figure 1. Database structure.....	10
Figure 2. Use-case diagram of Gallery.....	12
Figure 3. Activity diagram of Log-in.....	14
Figure 4. Data communication process between Flash and Database.....	15
Figure 5. Picture table in Database.....	16
Figure 6. PHP Function of loading data and creating XML output.....	16
Figure 7. Action Script 3 code for loading data.....	17
Figure 8. Displaying picture in Flash file.....	18
Figure 9. User log-in function.....	19
Figure 10. Page effect of Log-in function.....	20
Figure 11. Log-out function.....	20
Figure 12. User profile function.....	21
Figure 13. Displaying Album.....	21
Figure 14. Code of displaying picture.....	22
Figure 15. Displaying picture.....	23
Figure 16. Displaying comments.....	23
Figure 17. Code of display comment.....	24
Figure 18. Code of adding comment.....	24
Figure 19. Front-end effect of uploading picture.....	25
Figure 20. Code of uploading picture function.....	25

CONTENT

ABSTRACT

FIGURES

1	INTRODUCTION	1
1.1	Motivation	1
1.2	Structure of thesis.....	2
2	BACKGROUND INFORMATION	3
2.1	PHP	3
2.2	HTML	4
2.3	Flash	4
2.4	Action Script	4
2.5	Action Script 3	5
2.6	SQL	5
2.7	Xampp.....	5
3	RESEARCH DESIGN	6
3.1	Research Topic and Research Questions	6
3.2	Research Methodology.....	7
3.2.1	Research Method.....	7
3.2.2	Data Collection and Analyze	8
4	SYSTEM DESIGN	10
4.1	Database Structure	10
4.2	UML Diagrams	11
4.2.1	Use-case Diagram	11
4.2.2	Activity Diagram.....	14
5	SYSTEM IMPLEMENTATION	15
5.1	Interacting between Flash and SQL	15
5.2	Implemented Functions.....	18
5.2.1	User Log-in Function	19
5.2.2	Log-out Function.....	20
5.2.3	Displaying User Profile Function.....	21
5.2.4	Display Picture	22
5.1.1	Display Comment.....	23
5.2.5	Adding Comment.....	24
5.2.6	Uploading Picture	25

6	CONCLUSIONS.....	27
	REFERENCES.....	29
	APPENDICES	31
	Use-case description.....	31
	Activity diagrams	38

1 INTRODUCTION

In this chapter, the selection of the topic, beneficiaries of this thesis, and motivation are introduced. The structure of the thesis is presented as well.

The objective of my research is to find a solution for integrate PHP with Flash user interface. As a practicable result, I will implement an online gallery template for media students studying at Kemi-Tornio University of Applied Sciences. There will be three beneficiaries in terms of my research work. Firstly, the theoretical report of PHP and Flash integration will support the future education of my university. In addition, PHP and Flash integration will also benefit all who are interested in web-programming. Secondly, students who are from media department of my university will get a platform from my research to present their works. In addition, companies or organizations that need a web image gallery will be beneficiaries by using this developed template. Thirdly, for me, it is a good opportunity to familiarize myself with the whole working process from design to implementation of a complete system. It allows me to get knowledge of Flash Action Script of which I have no experience before. It improves my skills on the field of web programming as well as the field of writing.

1.1 Motivation

Designing and implementing an online gallery by using Flash Action Script 3.0 and PHP is an interesting and challenging topic since Flash has become an increasingly popular graphic user interface designing tool among web developers. The motivation of the research is to find out a proper solution to communicate and connect back and forth between PHP and Flash.

Along with the development of the Internet technology, increasing personalized activity can be done online. For instance, an online gallery is a web application that provides flexible and convenient performance for users to share their photos and art works with others. Comparing with a traditional static online gallery, the most famous feature of Flash gallery is “dynamic”. Pictures are presented lively instead of a steady image. It provides a

large number of transition effects through different template designs, such as desktop effect, star effect, blurs, gradients. Meanwhile, flash gallery provides function to keep real-time interactions with users based on multi-presentation effect. In addition, Flash gallery combines impressive experiences to the user via the features of videos, animations, texts, and music. On the other hand, Flash gallery produces a variety of functions and materials. By this way, picture presenting is not only a simple uploading and displaying process but also a personalized creation. Furthermore, Flash gallery provides a safety solution to protect the copyright for the users. Pictures inside a Flash gallery cannot be saved to a visitor's local computer.

1.2 Structure of thesis

This research is divided into 6 chapters. In chapter 2, some useful background information relating to the tools, programming language, and techniques is introduced. Chapter 3 presents the research phases which include research topic, research questions and research methodology. The technique about collecting and analyzing data is described as well. In chapter 4, design process is presented for database structure design and function development. Chapter 5 is written to introduce the implementation phase. Critical codes for the implemented functions are presented and explained in this chapter. As a conclusion phase, chapter 6 introduces the result of thesis and suggestions for further work.

2 BACKGROUND INFORMATION

In this chapter, all the background information of my research which is obtained from the literature review is introduced as follows.

2.1 PHP

“PHP is a widely used open source, general-purpose scripting language” (Valade 2004, 9). It was originally created by Rasmus Lerdorf in 1995 (PHP Documentation Group 2000), and has been in continuous development ever since.

Nowadays, PHP become a full-featured language (Valade 2004, 9), acquiring the recursive acronym of “Hypertext Preprocessor” (PHP Documentation Group 2000). Much of its syntax is borrowed from C, Java and Perl with a selection of unique PHP-specific features included in. The goal of the language is to allow web developers to write dynamically generated pages quickly. (Php.net 2010.) “The main implementation of PHP is now produced by The PHP Group and serves as the de facto standard for PHP as there is no formal specification.” PHP is free software released under the PHP License. (Lerdorf & Tatroe & MacIntyre 2006.)

When a PHP web page is requested by visitor, PHP codes are processed by web server. After this phase, all of necessary objects such as content and pictures are determined to show and the other parts are hidden such as file operations, math calculations. Elements that confirming by last phase are translating from PHP into HTML. After the translation, the final effect of web page is sent to visitors’ web browser. “The current version of PHP is PHP 5.3.2, and comparing with the earlier, it has improved LCG entropy” (Lerdorf & Tatroe & MacIntyre 2006) and “fixed safe_mode validation inside tempnam() when the directory path does not end with a /” (Jansen, 2010).

2.2 HTML

Hyper Text Markup Language “HTML” is a language which is used to describe web page. According to Raggett’s (2005) definition, HTML is not a programming language but a markup language since it uses markup tags to describe a web page. Because of historical reasons, the original HTML was developed and popularized by CERN and NCSA for military use. The latest version, HTML 5 was accepted in 2007 by W3C—World Wide Web Consortium and published in 2008. Comparing with earlier version, HTML 5 has the following features. Firstly, HTML 5 provides more APIs— Application Programming Interface for multiple applications and hardware. Secondly, it supports functions such as audio, video and image and creates new tags for web page, such as "<header>", "<section>" and "<footer>". (Meyer 2010, 1-3.)

2.3 Flash

As a standard of interactive vector graph and web animation, Flash was published by Macromedia on 1996 (Gassner 2010, 19). According to Mook (2007, xxiv), the features of Flash are including the following: “graphics and video display; a hierarchical event architecture; text display and input; mouse and keyboard control; network operations for loading external data and communicating with server-side applications; audio playback; printing; communicating with external local applications; programming utilities”. The first eight versions of Flash were developed by Macromedia and the newest version was published by Adobe after merging (W3CSchools 2010).

2.4 Action Script

Action Script is a scripting language for creating applications for the Web, desktop, and personal devices. It is unique compared to many languages in that it is not strictly a data exchange language. It is capable of generating interactive, rich media-driven applications, complete with design, animations, sounds, and video. (Ypenburg 2008, 1-3.)

2.5 Action Script 3

Action Script 3 is a “well-organized, mature language that shares much of its syntax and methodologies with other object oriented, strongly typed languages” (Braunstein 2010, 3). According to Ypenburg’s (2008, xiii) definition, Action Script 3 offers improved event handling, a new display list, and a drawing API.

2.6 SQL

SQL, an “abbreviation of Structured Query Language”, is not a computer programming language (W3CSchools 2010). “Instead SQL is a language used exclusively to create, manipulate, and interrogate databases” (Boczko, 324). As the predecessor of SQUARE language, the first version of SQL was developed by IBM on 1981 to manage SYSTEM R (a kind of relationship database management system). The features of SQL are as follows: (a) simple structure of language; (b) powerful functions for creating, managing and querying; and (c) easy to understand and use. (Instructional Software Research and Development Group 2006, 102-103.)

2.7 Xampp

Xampp is a distribution which can provide an easy solution to install PHP and Apache programming environment on the computer (Fakhroudinov 2011). The letter "x" means that the distribution is a cross-platform. The letter "a" refers to Apache. Letter "m" is the first letter of MySQL and letter "p" means PHP and Perl. As a powerful distribution, Xampp can be used in multiple operating systems such as Microsoft Windows, Linux, and Mac OS. (Keynote Support 2012.)

3 RESEARCH DESIGN

The aim of this chapter is to present two main contents. Firstly, the research topic, objectives and related research questions are described. Secondly, the research methodology which is used for gathering answer of above questions is provided and explained.

3.1 Research Topic and Research Questions

The topic of this research is an online gallery template based on Flash, PHP, and SQL database. The general objective of this work is to study and determine a measure of integration the communication process between Flash user-interface with SQL database.

The first concrete objective of this work is to develop a Flash user interface for displaying the gallery's functions using Adobe Flash. The second objective is to find an appropriate solution for integrating a Flash-designed user interface into web pages that is developed by PHP code. The last concrete objective is to identify what the most necessary functions are for an online gallery according to the aimed users.

In this work the research questions that will be addressed are listed below. The motivation for them and the techniques for finding answers to them are also discussed.

1 How can data be loaded from a SQL database and displayed on the flash user interface via PHP codes?

This question is concerned about the core technology of the study, and by answering this question a clear result is received which will benefit my future programming work. Documentary analysis is an effective methodology to be used to find answers to this question. The reason for this choice is that PHP and SQL have already been well developed and proven. Therefore, there are numerous books, articles and tutorials which are related to this field. Documentary analysis is a convenient way to find out a suitable solution for this question.

2 How can the performance of compatibility in multiple web browsers be improved?

The objective of my research work is to find a practicable solution for a PHP web application based on SQL database that presents data with a Flash user interface. A successful study result should include a developed online gallery which is suitable to any kind of web browsers, such as IE, Mozilla Firefox, and Safari. For answering this question, I use a simulation tool to test the template with different simulative environments; also, there are some useful resources about how to increase the compatibility. Therefore, documentary analysis is used here as well.

3 What are the most necessary functions for a personal album according to the aimed users?

A good online gallery must satisfy most of the aimed users, and the functions it provides should always follow the user requirements. In order to achieve this goal, the methodology I chose is interview.

3.2 Research Methodology

This chapter describes the research methods applied for the study. In addition, it also presents the reasons for choosing them.

“Methodology is a detailed account of the research design and the way the project is implemented. Research work and comparable development work refer to systematic activity to increase the level of knowledge to find new application. The essential criterion is whether the activity generates fundamental new knowledge.” (Hair et al. 2007, 398.)

3.2.1 Research Method

This research applies constructive research method. The reason for choosing this methodology is that the research objective is to design and develop a new solution for a practical problem which was built based on previous example research. The aim of my research is to find a practicable solution of PHP web application based on SQL database

that present data with a Flash user interface. However, the final result of my research work does not only focus on theoretical parts but also on implementing an application based on Flash and PHP.

According to Lindholm (2008, 345), constructive research “was originally developed in the field of management accounting in the 1980s in order to assist management accounting academics in taking more active role in improving the existing practices and gaining deeper understanding of the actual practices of the organizations”. In addition, it is widely used in software engineering and computer science. The artifacts such as models, diagrams, plans, organization charts, system designs, algorithms and artificial languages and software development methods are typical constructs used in research (Dodig-Crnkovic 2010, 2). In terms of my research work, the study of finding the solution of integrates PHP and Flash provides the theoretical foundation. As the techniques of constructive research method, interviews and documentary analysis can be used to gather and analyze data.

3.2.2 Data Collection and Analyze

As I presented above, interviews and documentary analysis are two research techniques included in constructive research. An interview is defined to be a discussion between a researcher and an interviewee (Maykut & Morehouse 1994, 121), or “an interchange of views between two or more people on a topic of mutual interest, sees the centrality of human interaction for knowledge production, and emphasizes the social situations of research data”(Kvale 1996, 14). For my research work the interviewees were determined by my supervisor. In order to understand and satisfy the user requirements, I interviewed one teacher who is from the media department of our university. The requested questions mainly focus on the architecture and functions of the personal online gallery template. To improve the quality of the collected information, face to face interviews were taken. The two individual interviews were hold for the teacher since she is the most experienced and with comprehensive knowledge on the field of media. The contents of interviews are not present in the research because they were held privately.

Documentary analysis is another technique that was used to collect, review, and analyze information related with this research. The literature to be reviewed includes information about PHP programming, SQL database, and Flash.

For collecting data from interview, the scope of interviewee should be determined at the beginning. According to my supervisors' (Koskenniemi 2011) suggestion, one interviewee was chosen. As I present in the previous chapter, for the two individuals interview, I created detailed and specialized questions. After the interviews, I sorted and analyze the collected data for finding the expected result.

For the documentary analysis, I asked my supervisor to suggest which books should be read. From previous study at KTUAS, I learned PHP programming and SQL database design and maintaining. For instance I took the courses of PHP programming, Database, and Programming project which is based on the knowledge of PHP and SQL. Therefore, I focused on reading books and instructions of Flash.

4 SYSTEM DESIGN

This chapter concentrates on the design process of the online gallery system. Two mainly measures are taken during this process. One is database structure and the other is UML diagram.

4.1 Database Structure

Database structure is a critical part for any system design process. In this case the database is named as gallery. There are the following five tables designed for the study: student table, creator table, album table, picture table, and comment table. The figure below presents the structure of the database design, and each table is introduced shortly.

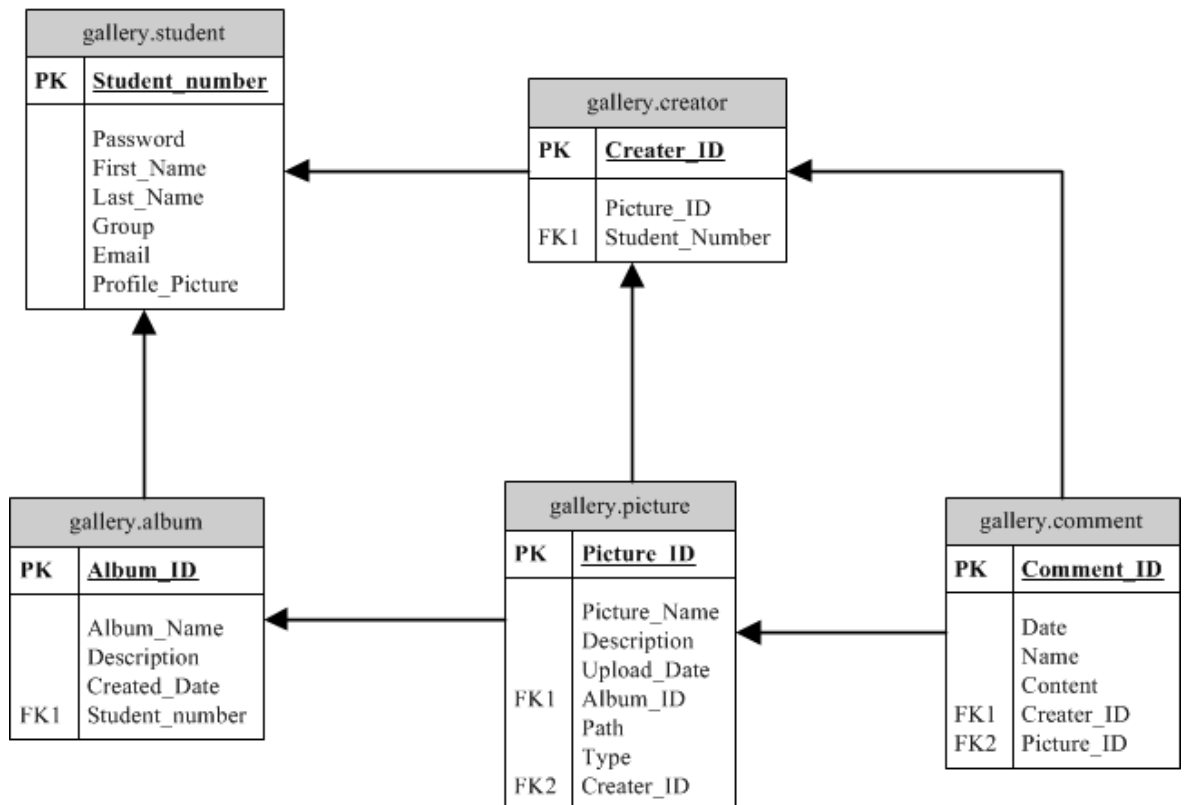


Figure 1. Database structure

The table of student is used to store all the personal information relating to the students who have an account from the system, such as student number, password and group. In addition, the column of Student_Number is used as a user name for students when they log-in to the system. In the album table, there is detailed information about the album. In order to store the content of these albums, the picture table is created. Detailed picture information, such as picture name, path and picture classification should be found in this table. The table of creator is an important part of this database considering that in some situations one picture has more than one author in this system, and therefore this table provides a possibility to store more authors for one picture. Furthermore, it connects information between the table of student and the table of picture as well. The last table is the table of comment. It allows the database to store the content of comment, the published date of the comment and other relative information for a specific picture.

Moreover, the relationship between the tables can be seen from the figure clearly. For instance, the column of Album_ID in the picture table is saved as foreigner key which is used to connect with the album table.

4.2 UML Diagrams

The Unified Modeling Language “UML” is a standard visual modeling language. It “is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems” (Kobryn 2001). In this case, the system is designed based on UML diagrams, such as class diagram, use-case diagram and activity diagram.

4.2.1 Use-case Diagram

Use case diagram, which was published by Doctor Iva Jackboson, is a method for describing system requirements. “Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors and use cases.” (Engineering Notebook 1998.)

Figure 2 describes all the required functions and actors in the system. There are 15 functions working between 3 types of actors, i.e. student, gallery, and database. One point should be noticed that “gallery” as an actor in this diagram means the system itself.

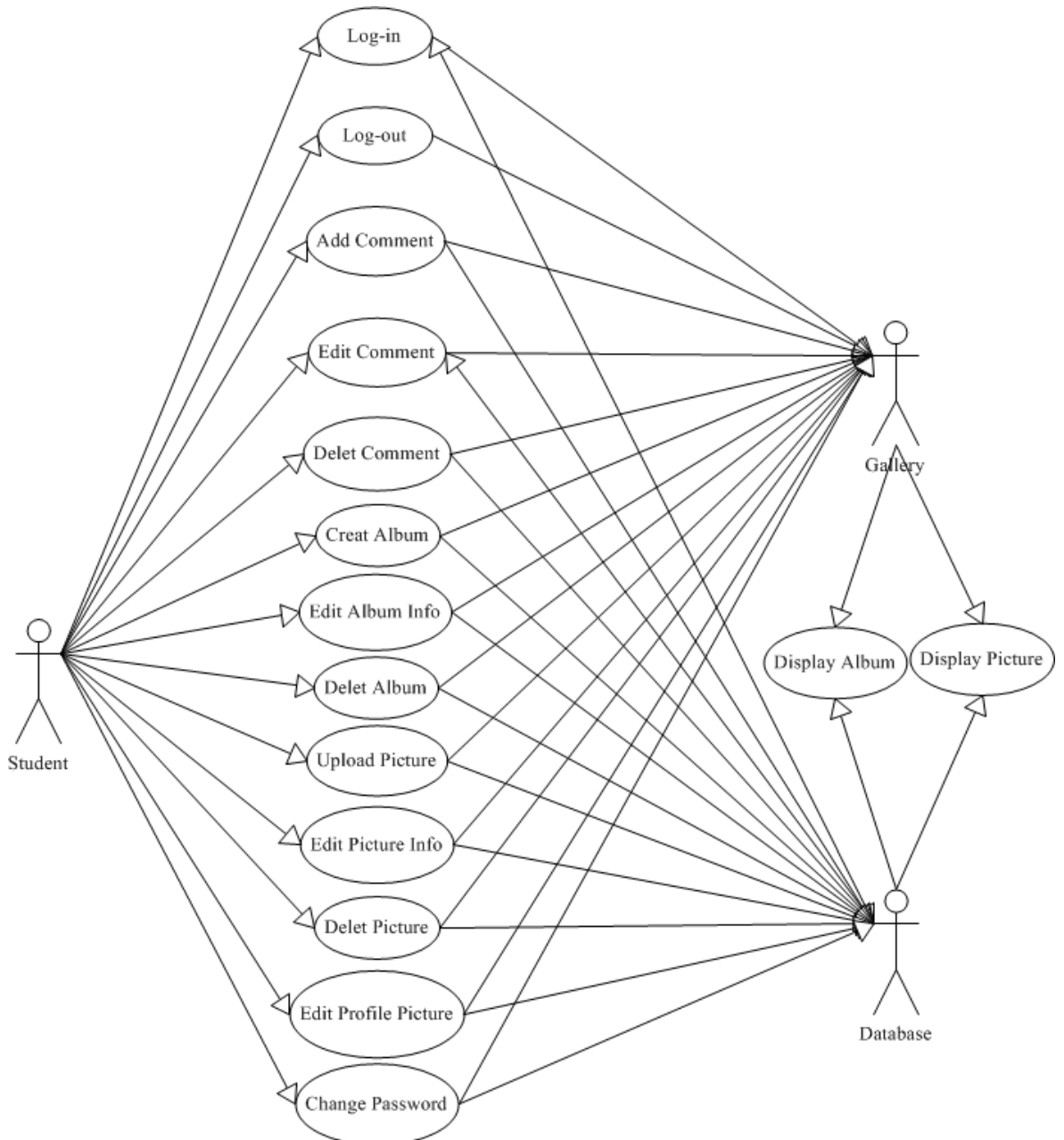


Figure 2. Use-case diagram of Gallery

In order to achieve all the user requirements, a better understanding about the step-by-step work flow among each use-case is necessary. Therefore, a detailed use-case document description was written. Table 1 below illustrates the work process and requirements for

function Log-in. The other 14 tables of each use-case description are shown in Appendix 1 on page 31.

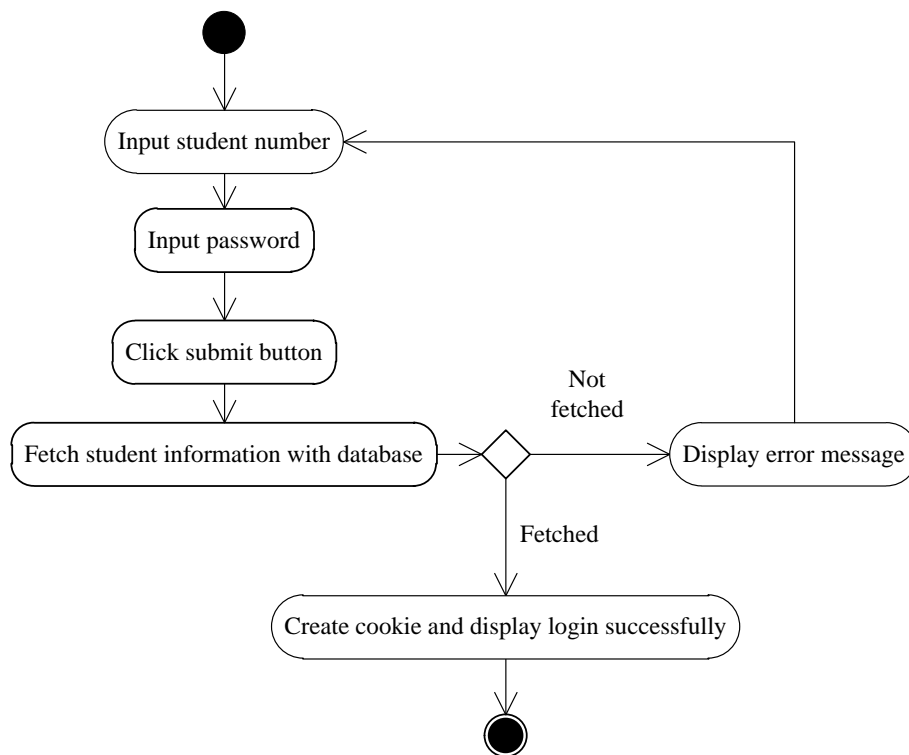
Table 1. Use-case description of Log-in

Use Case ID:	1		
Use Case Name:	Log-in		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student log-in to the online gallery.
Trigger:	Student starts to input information in log-in area.
Preconditions:	<ol style="list-style-type: none"> 1. Student wants to log-in to the Gallery. 2. Student wants to make a comment to a picture.
Postconditions:	<ol style="list-style-type: none"> 1. Student has an account in the gallery.
Normal Flow:	<ol style="list-style-type: none"> 1. Student inputs his/her student number. 2. Student input his/her password 3. Student clicks the button of submit. 4. Fetching student information with database. 5.1 If all the information is fetched, gallery create cookie for student and display log-in successfully. 5.2 If the information is not fetched, gallery displays an error message, and return to the log-in page.
Alternative Flows:	None
Exceptions:	None
Priority:	High
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

4.2.2 Activity Diagram

Lano (2009, 22) suggests that activity diagram is provided to describe behavior composed of collections of tasks in a graphical manner, such as the algorithms of operations, or the work flows of business processes. In terms of this research, there are 15 specified activities, and each activity is represented by several actions. The Figure 3 represents the activity flow of log-in. The other activity diagrams can be found in the Appendix 8 on page 38.



Figur 3. Activity diagram of Log-in

In terms of the Log-in activity, there are six actions performed by the system between the users. The activity is start from the user input his/her student number into the log-in field. After this step, the user input his/her password and click the button of submit on the right side of the log-in area. The system receives the inputted data from the user and processes by fetching them with the specific data that stored in the database. If the data is fetched, the system will create cookie for the user and the activity is finished successfully with a welcome message displayed to the user. If the data is not fetched, an error message is shown to the user and the user is asked by the system to re-do this process from the beginning.

5 SYSTEM IMPLEMENTATION

The aim of this chapter is to present the development and programming work for the system. As the key point and primary result of this research, the solution and workflow of the communicating function between Flash and SQL is described and explained below. In addition, the required functions of the system are presented and code explanations are showed as well.

5.1 Interacting between Flash and SQL

In this case, it can be concluded that Flash itself cannot be used to gather and display any data from a database. After the phase of research, a measure of communicating from SQL to Flash based on additional tools is provided and distributed as a function of Gallery. In this solution, PHP plays a role of a bridge between Flash and SQL to transfer data which is requested from database. Figure 4 below shows the data communication process between Flash and database.

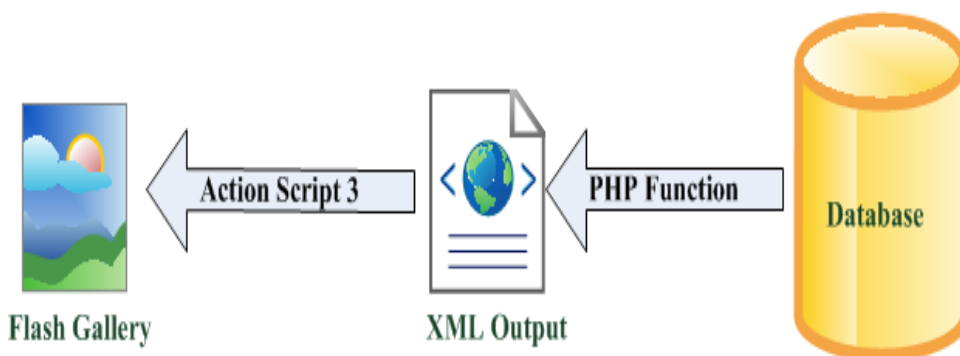


Figure 4. Data communication process between Flash and Database

When data is loaded by the PHP function, a tank is needed to keep those data and it should be understood and read by Flash Action Script 3 language. To transfer data for Action Script 3, XML is needed since it can be used to store data based on different tags and these tags can be complied by action script 3. Therefore, the clear process of communicating process between Flash and SQL can be determined. Firstly, PHP loads data from database and creates a dynamic XML output with the loaded data. After that, the Flash Action Script

function imports the dynamic XML output and converts data based tags. Finally the loaded data is displayed. In conclusion, this transportation process can be regarded as SQL passes a data to Flash via the bridge—PHP ,and the data is stored in a container—XML.

Figure 5 presents the necessary table which is used to load picture information. The columns of the table are listed below.

	Field	Type
<input type="checkbox"/>	Picture_ID	int (10)
<input type="checkbox"/>	Picture_Name	varchar (20)
<input type="checkbox"/>	Description	text
<input type="checkbox"/>	Upload_Date	timestamp
<input type="checkbox"/>	Album_ID	int (10)
<input type="checkbox"/>	Path	varchar (50)
<input type="checkbox"/>	type	varchar (20)

Figure 5. Picture table in Database

In this case, not all of columns are needed. In practice, the following four columns should be requested from this table: (1) Picture_Name; (2) Description; (3) Path; (4) Picture_ID. The Picture_ID column is used to confirm which picture's information should be loaded. The value is determined from the parameter that is added after the URL request. Comparing with the three others, the column of Path is quite important because it is linked to the place where the picture is stored on server.

Figure 6 below shows PHP function for loading picture information and outputting as a XML formatting.

```

1  <?php
2  session_start();
3  ob_start();
4  require("connect.php");
5  $id=$_SESSION['Album_ID'];
6  //echo $id;;
7  unset($_SESSION['Album_ID']);
8  echo "<?xml version='1.0'?">";
9  echo "<NODETOP>";
10 $SQL="SELECT * FROM album.picture WHERE Album_ID='$id'";
11 $query=mysql_query($SQL) or die ("failed to connect");
12 while ($row=mysql_fetch_array($query))
13 {
14     //echo $row[Picture_Name];
15     echo "<NODE><TITLE>". $row[Picture_Name]. "</TITLE>";
16     echo "<COPY>". $row[Description]. "</COPY>";
17     echo "<LINK>". $row[Path]. "</LINK>";
18     echo "</NODE>";
19 }
20 echo "</NODETOP>";
21 ?>

```

Figure 6. PHP Function of loading data and creating XML output

The example case is “preview album” function which is published in the system. The SQL statement is typed to select the picture data based on the Album_ID column. When data is fetched, PHP echoes a dynamic XML output and loaded data such as the Picture_Name, the Description are arranged into different tags.

Figure 7 displays the Action Script 3 function which is created in Flash file (.swf) to read and transfer data from XML output.

```

1  var loader:URLLoader;
2  loader = new URLLoader();
3  var increment = 0;
4  var total = 0;
5  var i = 0;
6  var TITLES = [];
7  var COPYS = [];
8  var LINKS = [];
9  loader.addEventListener(Event.COMPLETE, xmlLoaded);
10 var request:URLRequest = new URLRequest("loadpic.php");
11 loader.load(request);
12 function xmlLoaded(event:Event):void {
13     var myXML:XML = new XML(loader.data);
14     total = myXML.children().length();
15     trace(total);
16     imageNUMB.text = "1 of " + total;
17     textFieldA.text = myXML.children().TITLE.children()[0];
18     textFieldB.text = myXML.children().COPY.children()[0];
19     ImageLoad(myXML.children().LINK[0],holderClip,0,0);
20     for (i=0; i<total; i++) {
21         TITLES[i] = myXML.children().TITLE.children()[i];
22         COPYS[i] = myXML.children().COPY.children()[i];
23         LINKS[i] = myXML.children().LINK.children()[i];
24     }
25 }

```

Figure 7. Action Script 3 code for loading data

In this function, three main variables are declared to sync and store information which are published in XML output as same tags. In addition, the variable “total” is used for counting number of the loaded picture information. The function of “ImageLoad” is loaded for displaying the picture on Flash file and the variable of LINKS is imported as the parameter. Finally, the picture is presented as Figure 8 below.

As presented above, Figure 8 is the output that displayed in user's browser of the "preview album" function.

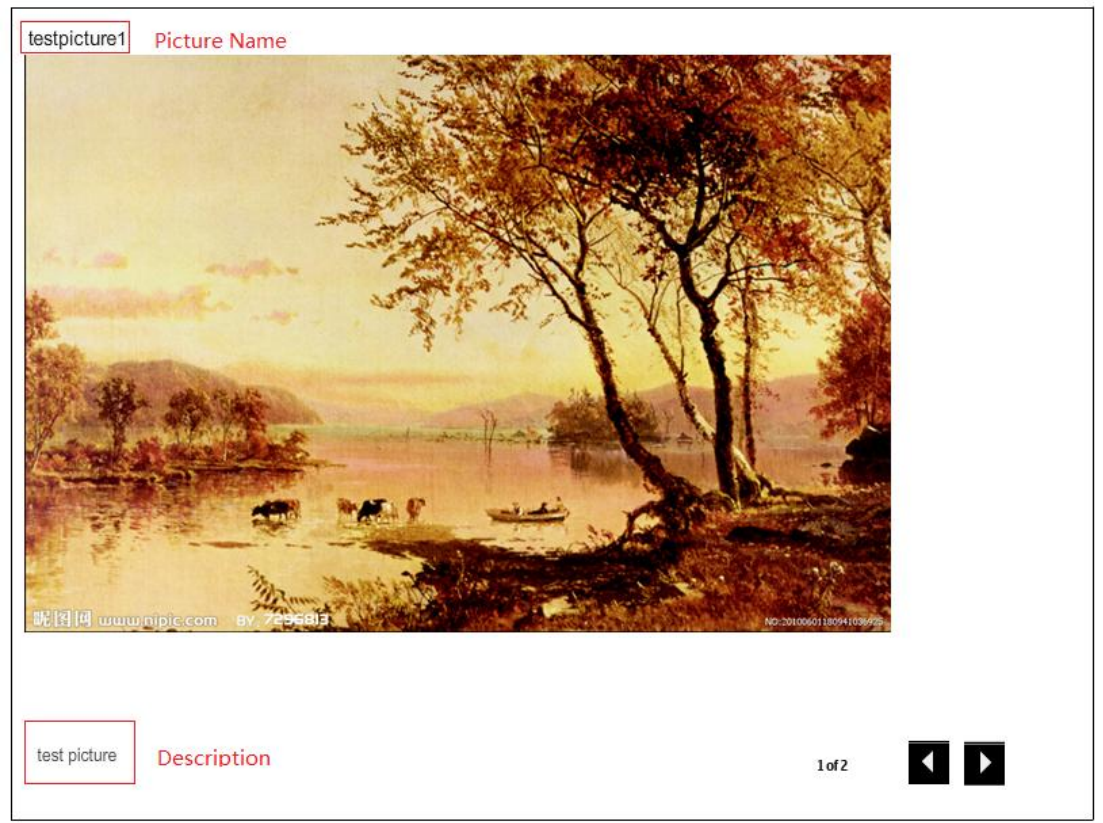


Figure 8. Displaying picture in Flash file

The pictures that stored in the selected album are performed on the middle of the frame. The names of each picture are presented on the top of the frame. In addition, on the bottom of the frame the description relating to each picture as well as the picture number is presented.

5.2 Implemented Functions

In this sub-chapter, all of the implemented functions which are used in Gallery, except for function which is presented previously, are presented and necessary code explanations are given. To make a clear impression for readers, every function explanation is divided to two parts. The code analysis is given firstly and output is added for an intuitive imaging.

5.2.1 User Log-in Function

Based on previous discussion with the customer (the teacher who provides topic and requests output for this thesis), user group is created within system database for testing in current version. A further work for separating user data and interacting with university's existing user database will be done in the future. Figure 9 depicts the user log-in function.

```

1  <?php
2  ob_start();
3  session_start();
4  require("connect.php");
5  $username=$_REQUEST['username'];
6  $password=$_REQUEST['password'];
7  $SQL="SELECT * FROM album.student WHERE Student_Number='$username' AND Password='$password'";
8  $query=mysql_query($SQL) or die ("failed to connect");
9  $row=mysql_fetch_array($query);
10 if($row > 0){
11     $fname=$row[First_Name];
12     $lname=$row[Last_Name];
13     setcookie("username",$username,time()+3600);
14     setcookie("password",$password,time()+3600);
15     setcookie("fname",$fname,time()+3600);
16     setcookie("lname",$lname,time()+3600);
17     echo "<script language='javascript'>location.href='index.php';</script>";
18 }
19 else{
20     echo "<script language='javascript'>location.href='index.php?type=1';</script>";
21 }
22 ?>

```

Figure 9. User log-in function

As is presented in figure 9, firstly another PHP file is imported for connecting the database. When user fills necessary username and password in the index page, those data are submitted by post to the “login.php” file. Two variables are declared to store the submitted data and search the student table in database based on them. If all information is fetched with database, the system cookie is created to save the student information and it will be kept for 1 hour. If inputted information is not fetched, the location will be transferred to the index page and an error message will be displayed.

The effect of “log-in” function is presented below in Figure 10. The figure displays the effect of page for before “log-in” and after “log-in”.



Figure 10. Page effect of Log-in function

In the effect of after “log-in”, a student’s first name is loaded from the database and displayed with the welcome message. There are two links providing on this page. The “My Profile” link allows user to check the student information and the “Log Out” link is used to call the “log-out” function.

5.2.2 Log-out Function

The Log-out function is called when user leaves the system to destroy cookie from browser. Otherwise user’s cookie is destroyed after 1hour. Figure 11 displays the function of Log out from system.

```

1  <?php
2  setcookie("username","");
3  setcookie("password","");
4  setcookie("lname","");
5  setcookie("fname","");
6
7  echo "<script language='javascript'>location.href='index.php';</script>";
8  ?>

```

Figure 11. Log-out function

The system calls this function to re-set user cookies such as username and password with empty value. After the cookies are destroyed, the page is re-located to the index page.

5.2.4 Display Picture

The display picture function is developed for displaying all pictures which belong to one album. There are two situations for calling this function. Firstly, the user can view his/her own album and pictures after logging in. Another way is displaying other user's album and pictures by clicking the button of "KÄYTTÄJÄT". Figure 14 shows the code of this function.

```
<?php
if($type!="")
{
    $SQL="SELECT * FROM album.picture WHERE Type='$type' Order by Upload_Date DESC";
}
if($aid!="")
{
    $SQL="SELECT * FROM album.picture WHERE Album_ID='$aid' Order by Upload_Date DESC";
}
$query=mysql_query($SQL) or die ("failed to connect");
// $row=mysql_fetch_array($query);
$count=0;
?>
<div style="margin-top:2%; margin:0 auto; width:auto; height:auto;">
<table class="xzd2" align="center" style="border:1px solid #d2d5b3" width="600px">
<tr>
<?php
while ($row=mysql_fetch_array($query))
{
?>
    <td align="center">
    <a href="picture.php?type=<?php echo $type;?>&pid=<?=$row[Picture_ID]?>&aid=<?=$row[Album_ID]?>">
    <br />
    <?=$row[Picture_Name]?></a></td>
<?php
$count++;
if($count==2)
{echo "</tr><tr>"; $count=0;}
}
?>
</tr>
</table>
```

Figure 14. Code of displaying picture

In terms of this function, when the user select displaying picture for his/her own album or the user select checking the other user's album, the variable of \$aid is used for fetching the target data. This variable is assigned and transferred to the "picture.php" or the "albumcontent.php" which is created for user's own album or other's album.

When the target data is fetched, as in the function of displaying album, a table is created and all the loaded data are filled into it as is presented in figure 15.



Figure 15. Displaying picture

5.1.1 Display Comment

The function of displaying comment is designed and developed for the user relating to the displaying picture function. When a picture is displayed, comments relating with this picture are presented on the right part of the page as figure 16 shows.

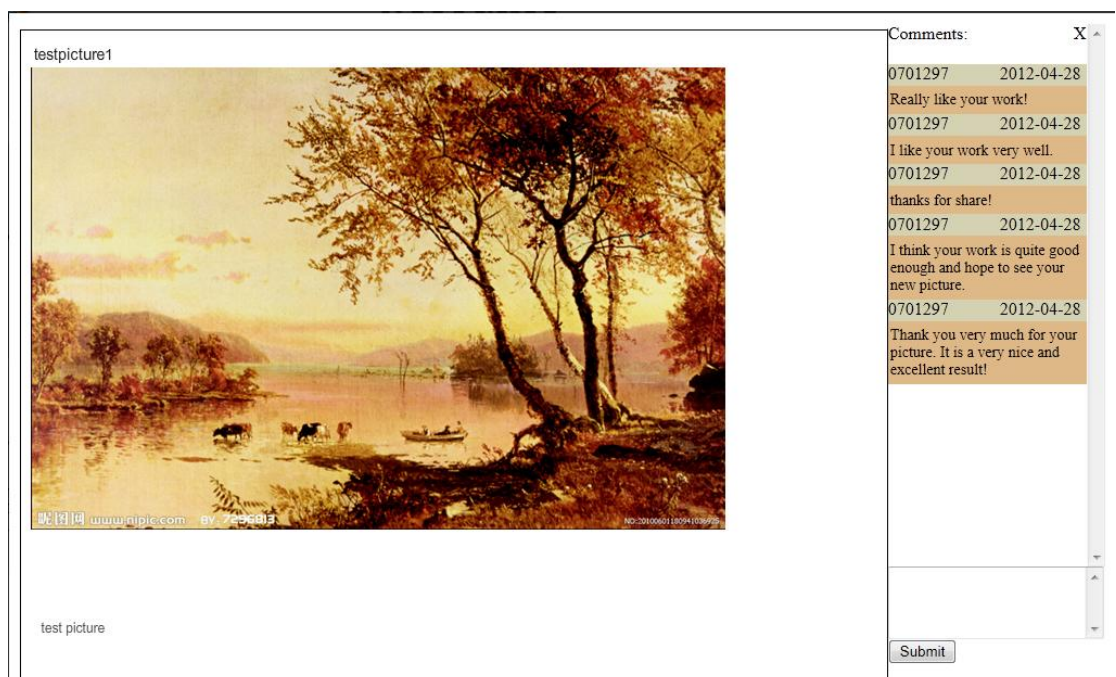


Figure 16. Displaying comments

In this function, the variable of “\$pid” is used for fetching the target picture comments from database. When data are fetched and loaded, a function which named “substr” is called to transfer the date of the comment to correct formatting. Example code of this function is presented in Figure 17 below.

```
<div style="width:200px; height:600px; overflow:hidden; float:right;">
<div style="width:200px; height:500px; border:0px; overflow-y:scroll; overflow-x:hidden; clear:both;">
<div style="width:180px; height:auto; font-size:20px; font-weight:bold;">
<div style="float:left; width:170px;">Comments:</div>
<div style="float:right; width:9px;">X</div></div>
<div style="clear:both;"></div>
<br />
<?php
    $SQL="SELECT * FROM album.comment WHERE Picture_ID='$pid' Order by Date DESC";
    $query=mysql_query($SQL) or die ("failed to connect");
    while ($row=mysql_fetch_array($query))
    {
        $date=substr($row[Date],0,10);
    }
    <div style="width:103px; float:left; height:20px; font-size:16px; font-color:white; background-color:#D5D2B3;"><?=$row[Name]></div>
    <div style="width:80px; float:right; height:20px; font-size:16px; font-color:white; background-color:#D5D2B3;"><? echo $date:></div>
    <div style="width:183px; clear:both; font-size:14px; background-color:#deb887; padding-top:5px; padding-bottom:5px; padding-right:2px; padding-left:2px;">
    <?=$row[Content]></div>
<?php
    }
<?
</div>
```

Figure 17. Code of display comment

5.2.5 Adding Comment

The function of adding comment is developed for the use that need to write comment for a picture. This function is divided into three parts. Firstly, a user interface is provided for typing the content at the front end page. Secondly, when the user finishes writing comment and clicking the “submit” button, a JAVASCRIPT function named “writecomment” is called for transferring the data to a PHP file which is responsible for adding them into database. The detailed code example is presented in figure 18 below.

```
if($type=="comment")
{
    $Picture_ID=$_REQUEST['pid'];
    $Content=$_REQUEST['comment'];
    $Name=$_COOKIE['username'];
    $SQL="INSERT INTO comment (Picture_ID, Name, Content) VALUES ('$Picture_ID', '$Name', '$Content')";
    $query=mysql_query($SQL) or die ("failed to connect");
    $response="Add comment Successfully!";
    echo $response;
}
```

Figure 18. Code of adding comment

5.2.6 Uploading Picture

The function of uploading picture is a critical function for the system. This function allows the user to upload the picture to the target album with necessary parameters such as the description of picture and the target category. The Figure below presents the effect of this function in the front-end page.

Figure 19. Front-end effect of uploading picture

One fact should be noticed in above figure. The text within red rectangle is presented basing on the user's browser language because of the feature of input tag. Therefore, it can be understood by different language users.

The figure below shows the PHP part of uploading picture function. The effect of this function is introduced as well.

```

$name=$_FILES["file_img"]["name"];
$type=$_FILES["file_img"]["type"];
$size=$_FILES["file_img"]["size"];
$album_id=$_POST["aid"];
$updir = "picture/".$_COOKIE["username"]."/";
$path = $updir.$name;
$type=$_POST["ptype"];
$description=$_POST["Description"];
if($description==""){$description="No Description.";}
if($name=="") {echo"<script>alert('');window.history.back();</script>";}
$tmp_name=$_FILES["file_img"]["tmp_name"];
if(move_uploaded_file($tmp_name,$updir.$name))
{
    $sql="INSERT INTO album.picture (Picture_Name, Description, type, Album_ID, Path) VALUES ('$name', '$description', '$type', '$album_id', '$path')";
    $query=mysqli_query($sql) or die ("failed to connect");
    $sql="UPDATE album.album SET Preview = '$path' WHERE Album_ID = '$album_id' LIMIT 1";
    $query=mysqli_query($sql) or die ("failed to connect");
    echo"<script>alert('Upload photo successfully!');
    location.href='profile.php?type=1';</script>";
}

```

Figure 20. Code of uploading picture function

When the form is submitted from front-end page, all data is transferred to this PHP function by POST. The process of this function is done by three phases. Firstly, the selected picture is uploaded to the server in the target folder. After that, the picture information is added into database at the table of picture. Finally, the “Preview” column which is in table of album is updated for storing the newest album thumbnail path. When all the previous phases are finished, the page is located back to the “profile.php” and a message for the user is displayed.

6 CONCLUSIONS

In the conclusion chapter, I sum up my findings based on this research and draw conclusions from the research.

In conclusion, this thesis work took almost six months to complete from getting thesis topic to implementing the expected output. The whole working process can be divided to three phases. First, the research phase was critical part of this process and it provided the necessary pre-study and the documentary analysis for solution confirmation. Secondly, the system design phase provided a solid foundation and specified a right way to the expected outcome. Finally, the implementation phase provided enough benefits from the other two phases and lead to the final result.

As was discussed in the previous, the overall objective of this thesis work was to design and implement an online gallery template solution based on PHP, Flash and SQL. To finish and get an excellent outcome, three research questions are asked to find out answers during the research process. As a result, all of research questions were answered.

As the primary and critical result during the research phase, the measure of communicating between Flash and SQL was determined and implemented. In terms of this research, Flash itself cannot be used to gather and display any data from a database. To solve this question, PHP plays a role as a bridge between Flash and SQL to transfer data which is requested from the database. In addition, after the data is loaded by PHP function, a tank is needed to keep those data and it should be understood and read by Flash Action Script 3 language. To archive this task, XML provides good performance since it can store data based on different tags and these tags can be complied by the Action Script 3. Therefore, Flash and SQL communicating process can be determined clearly. Firstly, PHP loads data from the database and creates a dynamic XML output with the loaded data. Secondly, the Flash Action Script function imports the dynamic XML output and converts the data based on the specific tags. Finally, the browser loads the data and displays it to the user.

Furthermore, as a practical outcome, an online gallery framework is developed and implemented. All of the designed functions such as log-in, displaying picture via Flash and adding comment are implemented as well.

REFERENCES

- Boczko, Tony 2007, Corporate accounting information systems. Prentice Hall. New Jersey, USA. 323-324.
- Braunstein, Roger 2010. Action Script 3.0 Bible (2nd Edition). Wiley. Hoboken, NJ, USA. 2010. 3-8.
- Dodig-Crnkovic, G. 2010. Constructive Research and Info-Computational Knowledge Generation. Mälardalen University. Sweden. Downloaded 9 May, 2012.
< <http://www.mrtc.mdh.se/~gdc/work/MBR09ConstructiveResearch.pdf> >
- Frantzell Nils-Erik 2004, Install XAMPP for easy, integrated development, Read 02.2012
< <http://www.ibm.com/developerworks/linux/library/l-xampp/> >
- Fakhroutdinov, Kirill 2011. UML Notation and Examples. Read 04. 2012.
< <http://www.uml-diagrams.org/> >
- Gassner, David 2010. Flash Builder 4 and Flex 4 Bible. John Wiley & Sons. Hoboken, NJ, USA.
- Hair, Joseph F. & Money, Arthur H. & Samouel, Phillip & Page, Mike 2007. Research method for business. John Wiley & Sons Ltd, West Sussex, England.
- Instructional Software Research and Development Group 2006. Introduction to database management systems. Tata McGraw-Hill. New Delhi.
- Keynote Support 2012. Turn Your PC Into a Web Server: Installing XAMPP Lite (Apache HTTP Server) Freeware. Read 01.2012.
< <http://www.keynotesupport.com/xampp-apache-tutorial.shtml> >
- Kobryn, C 1999. UML 2001: A Standardization Odyssey, Communications of the ACM. Vol. 42 No. 10, 29-37.
- Koskenniemi Yrjö September, 2011, Discuss with the supervisor.
- Lano Kevin 2009. UML 2 Semantics and Applications. Wiley. Hoboken, NJ, USA.
- Lerdorf, Rasmus & Tatroe, Kevin & MacIntyre, Peter 2006. A Brief History of PHP (Programming PHP) 2nd edition. O'Reilly. Media. Read 01. 2012.
< <http://scholar.lib.vt.edu/manuals/php/intro-history.html> >
- Lindholm, Anna-Liisa 2008. A constructive study on creating core business relevant CREM strategy and performance measures. Facilities. Vol. 26 Issue 7, 343 – 358.

- Maykut, P. & Morehouse, R. 1994. Beginning qualitative research: a philosophic and practical guide. Falmer Press. London, UK.
- Meyer, Jeanine 2010. The essential guide to HTML5: using games to learn HTML5 and JavaScript. Apress. New York, USA.
- Moock, Colin 2007. Essential ActionScript 3. O'Reilly. Beijing.
- PHP Documentation Group 2000. PHP Manual. Read 02. 2012.
<<http://scholar.lib.vt.edu/manuals/php/introduction.html>>
- PHP.net 2010. General Information. The PHP Group. Read 01. 2012.
<<http://php.net/manual/en/faq.general.php>>
- Raggett, Dave, 2005. Getting started with HTML. Revised 24 May 2005. Read 01.2012.
<<http://www.w3.org/MarkUp/Guide/>>
- Valade, Janet 2004. PHP 5 for Dummies. Wiley. Hoboken, NJ, USA. 9-10.
- W3Schools 2010a, Introduction to Flash, Read 12.2011.
<http://www.w3schools.com/flash/flash_intro.asp>
- W3School 2010b, Introduction to SQL, Read 12.2011.
<http://www.w3schools.com/sql/sql_intro.asp>
- Ypenburg, Derrick 2008a, Action Script 3.0: learn Action Script the quick and easy way. Peachpit Press, cop. Berkeley, Calif. 1-3.
- Ypenburg, Derrick 2008b, Action Script 3.0: learn Action Script the quick and easy way. Peachpit Press, cop. Berkeley, Calif. xiii.

APPENDICES

APPENDIX 1

Use-case description

Use Case ID:	2		
Use Case Name:	Log-out		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student log-out from the online gallery.
Trigger:	Student clicks the button of log-out.
Preconditions:	1. Student wants to log-out to the Gallery.
Postconditions:	Student has logged-in to the gallery.
Normal Flow:	1. Student clicks the button of log-out. 2. Gallery deletes cookie for student. 3. Return to the main page.
Alternative Flows:	None
Exceptions:	E 1. Cookie is expired before log-out.
Priority:	High
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

Use Case ID:	3		
Use Case Name:	Add comment		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student adds a comment to the picture in the gallery.
Trigger:	Student starts to input information in comment area.
Preconditions:	1. Student has logged-in to the Gallery.
Postconditions:	None
Normal Flow:	1. Student inputs his/her comment to the comment area. 2. Student clicks the button of submit. 3. Comment is saved to the database. 4. The new comment is displayed on the webpage.
Alternative Flows:	1. If student input comment without log-in, then an message will be displayed to ask him/her log-in first.
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

APPENDIX 2

Use Case ID:	4		
Use Case Name:	Edit comment		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student edits a comment which was created by him/her about a picture.
Trigger:	Student clicks the button of edit from the comment area.
Preconditions:	1. Student has logged-in to the Gallery. 2. This comment is created by him/her.
Postconditions:	None
Normal Flow:	1. Student selects a comment which is written by him/her. 2. Student clicks the button of edit. 3. Student input new comment. 4. Comment is saved to database. 5. New comment is displayed on the website.
Alternative Flows:	1. If student input comment without log-in, then a message will be displayed to ask him/her to log-in first.
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

Use Case ID:	5		
Use Case Name:	Delete Comment		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student deletes the comment that him/her has created.
Trigger:	Student clicks the button of delete from the comment area.
Preconditions:	1. Student has logged-in to the Gallery.
Postconditions:	None
Normal Flow:	1. Student chooses a comment which is created by him/her. 2. Student clicks the button of delete. 3. Comment is deleted from the database.
Alternative Flows:	None
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

APPENDIX 3

Use Case ID:	6		
Use Case Name:	Create Album		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student creates a new album.
Trigger:	Student clicks the button of create an album.
Preconditions:	1. Student has logged-in to the Gallery.
Postconditions:	None
Normal Flow:	1. Student clicks the button of create an album. 2. Student input the album name. 3. Student clicks the submit button. 4. The information of this album is saved to database.
Alternative Flows:	None
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

Use Case ID:	7		
Use Case Name:	Edit Album Information		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student edits the name and text description for his/her album.
Trigger:	Student clicks the button of edit for an album.
Preconditions:	1. Student has logged-in to the Gallery. 2. Album existing. 3. Album has been created by him/her.
Postconditions:	None
Normal Flow:	1. Student clicks the button of edit album. 2. Student changes the information of the album name and/or the text description. 3. Student clicks the submit button. 4. The information of this album is updated to database.
Alternative Flows:	None
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

Use Case ID:	8		
Use Case Name:	Delete Album		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
---------	----------------------------

Description:	Student deletes an album which is created by him/her.
Trigger:	Student clicks the button of delete for a selected album.
Preconditions:	1. Student has logged-in to the Gallery. 2. The album existing.
Postconditions:	None
Normal Flow:	1. Student chooses an album which is created by him/her. 2. Student clicks the button of delete. 3. Gallery displays a message to ask user to confirm “Do you want to delete the album?” 4.1.1 If student chooses “Yes”, all the pictures belong to this album are deleted from the database. 4.1.2 The information of this album is deleted from the database. 4.2.1 If student choose “No”, function stop.
Alternative Flows:	None
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student’s action.
Special Requirements:	Internet connection, computer

Use Case ID:	9		
Use Case Name:	Upload Picture		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student upload picture to a selected album.
Trigger:	Student clicks the button of upload a picture to a selected album.
Preconditions:	1. Student has logged-in to the Gallery. 2. Student selects an album.
Postconditions:	None
Normal Flow:	1. Student clicks the button of upload a picture. 2. Gallery displays a message to ask student to choose the path of the picture. 3. Student chooses a picture from local computer. 4. Student clicks the upload button. 5. The picture is uploaded to the server, and the information of the picture is saved to the database.
Alternative Flows:	None
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student’s action.
Special Requirements:	Internet connection, computer

Use Case ID:	10		
Use Case Name:	Edit Picture Information		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
---------	----------------------------

Description:	Student edits the name and text description for picture.
Trigger:	1. Student clicks the button of edit for a picture and/or student starts to input text description for a picture.
Preconditions:	1. Student has logged-in to the Gallery. 2. Picture existing. 3. Picture has been uploaded by him/her.
Postconditions:	None
Normal Flow:	1. Student selects the picture from an album. 2. Student clicks edit button. 3. Student changes the information of the picture name and/or the text description. 4. Student clicks the save button. 5. The information of this picture is updated to database.
Alternative Flows:	None
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

Use Case ID:	11		
Use Case Name:	Delete Picture		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student deletes a picture which is uploaded by him/her.
Trigger:	Student clicks the button of delete for a selected picture.
Preconditions:	1. Student has logged-in to the Gallery. 2. The picture existing.
Postconditions:	None
Normal Flow:	1. Student chooses a picture which is created by him/her. 2. Student clicks the button of delete. 3. Gallery displays a message to ask user to confirm "Do you want to delete the picture?" 4.1.1 If student chooses "Yes", the picture is deleted from the server. 4.1.2 The information of this picture is deleted from the database. 4.2.1 If student choose "No", function stop.
Alternative Flows:	None
Exceptions:	None
Priority:	Medium
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

Use Case ID:	12		
Use Case Name:	Edit Profile Picture		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student edits his/her profile picture.

Trigger:	Student clicks the button of edit profile picture.
Preconditions:	1. Student has logged-in to the Gallery.
Postconditions:	None
Normal Flow:	<ol style="list-style-type: none"> 1. Student selects a picture from local computer 2. Student clicks the button of upload. 3. The picture is uploaded to the server. 4. The new avatar information is updated to the database.
Alternative Flows:	None
Exceptions:	None
Priority:	Low
Frequency of Use:	Depends on student's action.
Special Requirements:	Internet connection, computer

Use Case ID:	13		
Use Case Name:	Change Password		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Student, Gallery, Database
Description:	Student change his/her password to the Gallery.
Trigger:	1. Student clicks the button of change password.
Preconditions:	1. Student has logged-in to the Gallery.
Postconditions:	None
Normal Flow:	<ol style="list-style-type: none"> 1. Student input his/her new password. 2. Student re-types his/her new password. 3. Student clicks the button of submit. 4. The Gallery checks the consistency of the new password. <ol style="list-style-type: none"> 4.1 If the password is the same with the re-typed one, then the new password is updated to the database. 4.2 If the password is not same with the re-typed one, then the Gallery displays an error message to the student.
Alternative Flows:	None
Exceptions:	None
Priority:	Low
Frequency of Use:	Low
Special Requirements:	Internet connection, computer

Use Case ID:	14		
Use Case Name:	Display Album		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Gallery, Database
Description:	Gallery displays album information from database.
Trigger:	1. User visits the Gallery.

APPENDIX 7

Alternative Flows:	None
Exceptions:	E 1 There is no album information existing in database.
Priority:	High
Frequency of Use:	High
Special Requirements:	Internet connection, computer

Use Case ID:	15		
Use Case Name:	Display Picture		
Created By:	Yin Yiming	Last Updated By:	Yin Yiming

Actors:	Gallery, Database
Description:	Gallery displays picture information from database.
Trigger:	1. User visits an album.
Preconditions:	Album existing.
Postconditions:	None
Normal Flow:	1. Gallery requests picture information from database. 2. Database sends picture information to the Gallery. 3. Gallery displays the picture and it's information..
Alternative Flows:	None
Exceptions:	None
Priority:	High
Frequency of Use:	High
Special Requirements:	Internet connection, computer

Activity diagrams

