Li Yicheng

Development of a blog system using CodeIgniter framework

Development of a blog system using CodeIgniter framework

Li Yicheng

Bachelor’s thesis

Spring 2011

Degree Programme in Business Information Technology

Oulu University of Applied Sciences

Abstract

School: Oulu University of Applied Sciences

Programme: Degree Programme in Business Information Technology

Author: Li Yicheng

Supervisor: Matti Viitala

The title of thesis: Development of a blog system using CodeIgniter framework

Year: 2011

Number of pages + number of appendices: 59

This thesis is project based thesis. It is a simple blog system. In the thesis will introduces the development blog background and discuss the design and development platform targets, set out the structure of the website and the function modules and introduce how to build this system step by step. The whole system can be divided into three parts: Information services, Album service, Message service.

However, due to the large size of the project, It cannot be explained how to make it step and step. So in this thesis will be focus more on the main part. In the theory part, will be discussed the exactly advantages of those programming language which will be used in the project. In the tutorial part, will be introduced how to build this blog system. Use the programming codes and pictures which from the project to describe clearly.

This purpose of this bachelor's thesis is to introduce the advantages and techniques of PHP framework (CodeIgniter) and jQuery. Another purpose is to build an example system which is using PHP framework and jQuery to student who want to learn more about programming. The codes of HTML and PHP are

very detailed. That help students who want learn PHP framework easier.

As a result of the thesis, Oulu University of Applied Sciences will have a good introduction of blog system which are using PHP framework. The website and codes can be used later for the course material.

Keywords: PHP, JQuery, PHP framework, CodeIgniter

Content

Development of a blog system using CodeIgniter framework

1 INTRODUCTION

"Blog" is an abbreviated version of "weblog," which is a term used to describe web sites that maintain an ongoing chronicle of information. A blog features diary-type commentary and links to articles on other Web sites, usually presented as a list of entries in reverse chronological order (wordpress blog, referred 10.02.2011).

Normally it can be understood that blog is a type of website, with regular entries of commentary, descriptions of events, or other material such as graphics or video (wordpress blog, referred 10.02.2011). It is a place to express your thoughts and feelings. It is a place to introduce yourself to the world and let more other people know you.

In recent years, blog has become very popular. More and more people have created their own blogs and view others' blogs. Today blogs are being used for almost all sorts of purposes. For business blog that company can use it to gain insight into your customers by allowing your customers to leave comment on the blog. You can put advertise on the blog, it is an opportunity to mention your company name and products. For personal blog that user can use it connect your family and friends easier. User can post anything on the blog to tell friends how you fell and what are you doing now. It is a great opportunity to know some new friends.

1.1 System development background

Before blogging became popular, people usually use Bulletin Board Systems (BBS) or some other chat tools like MSN, Gtalk to express their own ideas. By 2000, blog is developing with a high speed. Due to Blogger, Pita, Manila, Big blog Tool, Diaryland and much other related software became free software, and they also provide free server. It makes blog easier to update, publish and maintain. With these, blog is not just a place for user to post something or leave some comments. User can also watch video, listen music and so on.

According to incomplete statistics, it already has 40 million of blogs until now.
It has become a way to reflect on life. Some sites, such Twitter or Facebook, allow users to post anything on the blog, share thoughts and feelings, contact friends or family easier.

If the blog site is a stage, then all of the blogger is one of the dancers, they will own the real side show in the online world. If they just want to write blog, then the blog is a mirror, can show yourself in the mirror. If the blog is a media, you can use it to publish some valuable Information and share with friends.

1.2 The present situation of development

It can be clearly observed from the picture below, the trend of blog going up these years. According to the chart, the number of blogs increased from 4.3 million in 2006 to 28 million in 2009. November 3.2009, there are already have

28.3 million blog sites and 16.87 million bloggers. That means average 1.67 blogs for each user. Nearly a million bloggers update their own blogs at least once a week. New blog creation continues to grow. More than 75,000 new blogs created every day, which means that on average, a new blog is created every second of every day.

"The blogosphere continues to grow at a quickening pace." wrote, Technorati CEO Dave Sifry (masternewmedia blog, referred 12.02.2011).

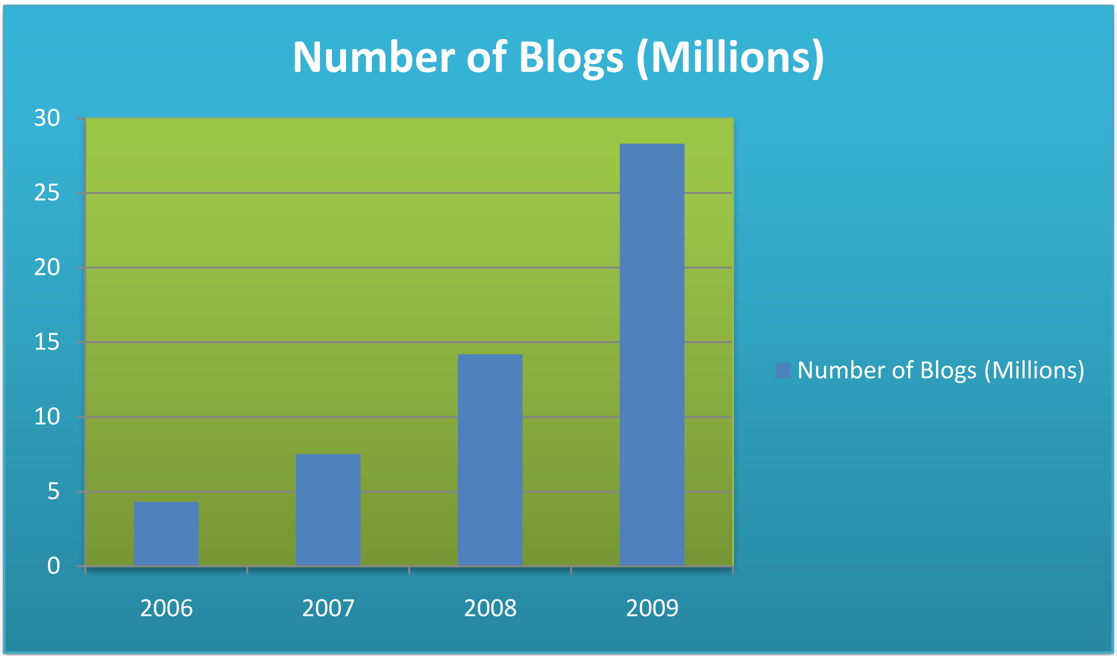


Figure 1 – chart for number of blog

1.3 Development trend

Early blog website were simple, web style rigid, but with the technology, have variety of programming skills come out, including the CGI, ASP, PHP, Java, script language such as: JavaScript, jQuery, VBScript and so forth. JavaScript and VBScript can be embedded in HTML to implement some of the dynamic capabilities. PHP, ASP make system are easy to maintenance and code, also it offers many levels of security to prevent malicious attacks.

Early blog were primarily textual, but with the multimedia development, blog have become more and more rich media. Blog provide commentary or news on a particular subject; others function as more personal online diaries, and now many blogs have provided videos (video blogging – e.g. www.brightcove.com), music (MP3 blog – e.g. prettymuchamazing.com), and audio (podcasting - e.g. www.rhythmcreation.com).

Now, blog has become an online Reputation mechanism. The user account will be a personal virtual identity on the internet like an ID card in the real life, it show the real name and personal info. People used to use anonymity before and we do not know each other. But now we can use blog to find someone and follow him. It can also avoid bad thing happened like dissemination some false news.

With the development of the blog, more and more people are start to using it. Blog as an Information media, it has very large information per day and some junk info was inevitable in it. Like someone just registered a lot of user accounts but never use it or post some porn ads on the blog. In the future, it will have a system to filter information and use personal virtual identity to find who publish it.

2 TECHNOLOGY ANALYSIS

Several different programming techniques are using in this blog system like PHP framework (CodeIgniter), jQuery, Ajax and so on. In this chapter, will be discussed the advantages of the techniques and introduce how to use it with a short description.

2.1 CodeIgniter

CodeIgniter is a powerful PHP framework with a very small footprint, built for PHP coders who need a simple and elegant toolkit to create full-featured web applications (CodeIgniter, referred 15.02.2011).

There are many other advantages of using CodeIgniter:

- Small, Fast, Simple, Easy to learn.
- It is easy to migration from one server to another. Just need to change the URL.
- CodeIgniter is easy to install, just need few minutes. All you need to do is download the CodeIgniter folder from www.Codeigniter.com. Extract it and put it into the server.
- Simple to debug
- Active Record Implementation is simply superb and easy to remember.
- One of the most important features of the CodeIgniter is that it has a most well - document.
- The collection of libraries that it posses is also good enough.
- And as previously said awesome documentation of the user guide, which makes any coder easy to use the whole framework (macronimous, referred 15.02.2011).

The following example is display data from database. Through this example it clear to see the difference between normal PHP and CodeIgniter. For normal PHP, sometimes the codes were mixed with HTML, and hard to read and modify. Using CodeIgniter those problems won't happen. In this example, the controller folder is to connect the back end and front end. In this example, the book function is to select data from database and load a view file. The view folder is to display what users in face see.

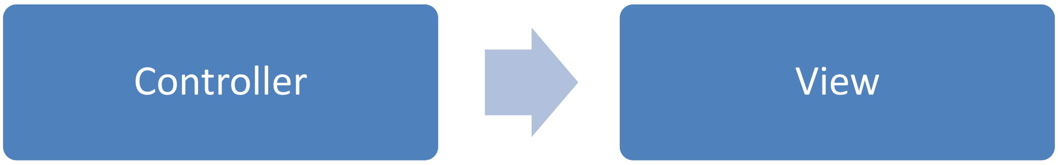


Figure 2 – CodeIgniter process

Create a file named book.php in views folder.

View (views/book.php)

```
<Html>
         <Body>
            <p><?= $a;?></p>
         </Body>
</Html>
```

Create a file named my_controller.php in Controllers folder.

Controllers (controllers/my_controller.php)

```
Function book () {
            $this->db->where (”type”, “book”);
            $this->db->from (”library”);
            $result = $this->db->get ();
            $count = $result->num_rows ();
            $data [“a”] = $count;
            $this->load->view (“book”, $data);
}
```

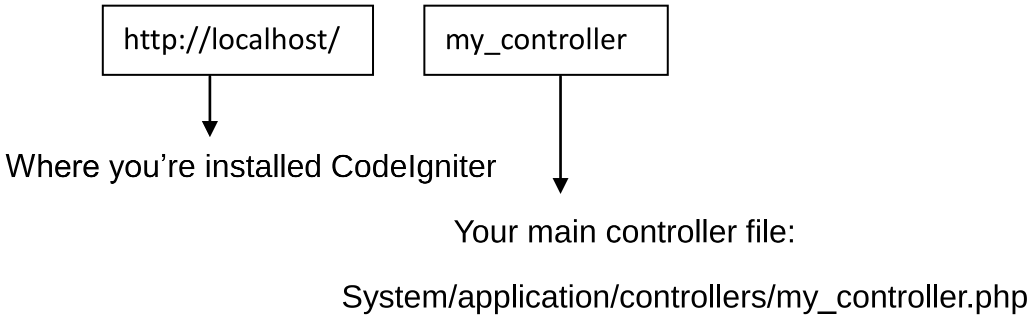

http://localhost/
my_controller
Where you’re installed CodeIgniter
Your main controller file:
System/application/controllers/my_controller.php

Figure 3 – CodeIgnier URL

2.2 Ajax

Ajax (shorthand for Asynchronous JavaScript and XML) is a group of interrelated web development methods used on the client-side to create interactive web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data is usually retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not needed (JSON is often used instead), and the requests need not be asynchronous. (Wiki Ajax, Referred 15.02.2011)

One of the main reasons uses Ajax. With Ajax, the web page can be refreshed dynamically. The web page does not have to be reloaded each time after the user requests a change. For example a login system, with normal login page, user need enter the data into the login form and then click on the submit button to submit the request to the server. Page will jump to PHP page and use PHP function to check the account. Then return to the login page to show the message like login success or wrong password. This process is inefficient and time consuming. However, using Ajax it can use the application to check user account without refreshing the whole page. Ajax will help to increase the web page's speed and usability.

2.3 JQuery

JQuery is a new kind of JavaScript library. It is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. JQuery is designed to change the way that you write JavaScript. (JQuery, referred 16.02.2011)

Advantages:

- Easy to use - This is the main advantage of using JQuery; it is easier to use compared to standard JavaScript and other JavaScript libraries. With the simple syntax, it can save coding a lot. Use a small amount of codes to achieve the same feature in comparison.
- Large library - Various number of plug-in are free to download and use it.
- Great documentation and tutorials - The JQuery website has the most well - document and tutorials with some useful examples.
- Simplifies Ajax
- jQuery is open source software

Here is two Basic Examples of using jQuery:

```
$(document).ready(function(){
   $('#hide').click(function(){
      $('a').hide('fast');
   });
});
```

For this example, after a element with id ‘hide’ is clicked, all the contents element (‘a’) will be hidden immediately with slight animation for the whole page.

```
$(document).ready(function(){
   $(’.load’).load(‘example.html’);
});
```

The above example is using Ajax on jQuery. It is simple to get data from other page. The response get from the example.html is fill inside the element with class is ‘load’

3 SYSTEM DEVELOPMENT

3.1 System Analysis

This blog is similar like WordPress, but it was a totally different system. It will have much more powerful functions. It is not just a blog for user writes something or leaves some comments. You can create your own album, Initiate votes, and meet friends.

The purpose of the project is to provide a better place to promote friendship for friends. The users can be used for notice, announcement and event. Therefore, in the project there are three main parts. Through this you can share the idea and connect your friends easier.

- ➢ Information services:

- User can write your own blog, view other user’s blog.
- Add Profile infomation
- Upload Profile image

➢ Album service:

- Allows user to send pictures to blog’s album and make tag on the pictures.

➢ Message service:

- Allows visitor to leave comment or message each other on the blogs

3.2 System Flow Chart

Blog system base on B / S (Browser / Server) structure (browser, web server, and database). The following map shows the process of the system.

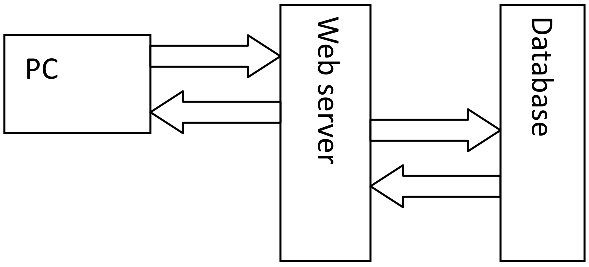


Figure 4 - System main structures

Major functions for this blog:

Login: user can use a username and password to log into the system to obtain access and can then log out or log off when the access is no longer needed.

User info: Introduce yourself from what you're doing now to where you grew up and upload your profile picture.

BLOG: user can write your own blog, view other user's blogs and leave comments for it.

Album: allows user to send pictures to his/her own blog picture album.

Wall: The Wall is user's profile page. Visitors can see his/her full profile and free to post messages on blog**.** Similar like Facebook, but it's a simple one.

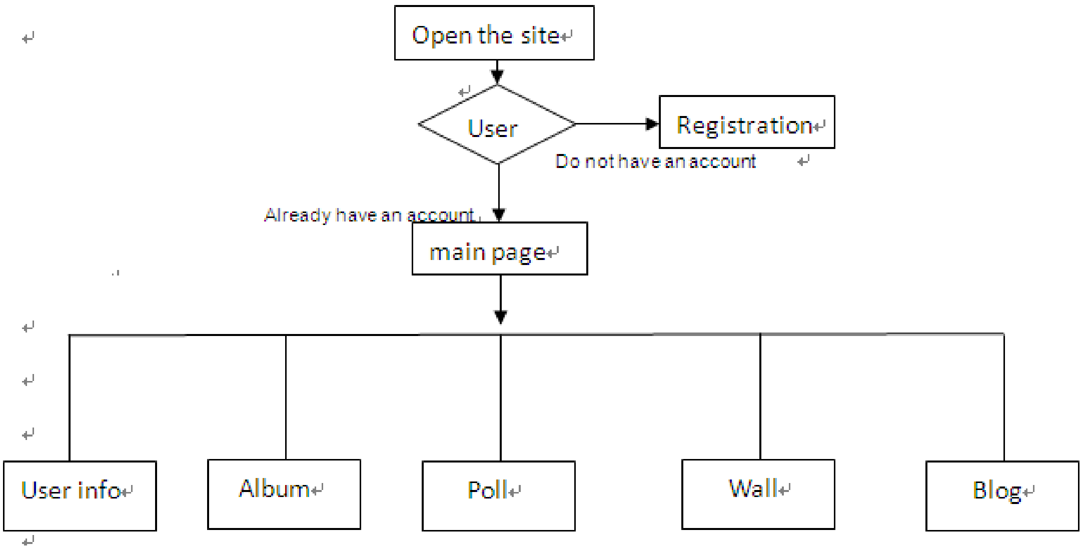
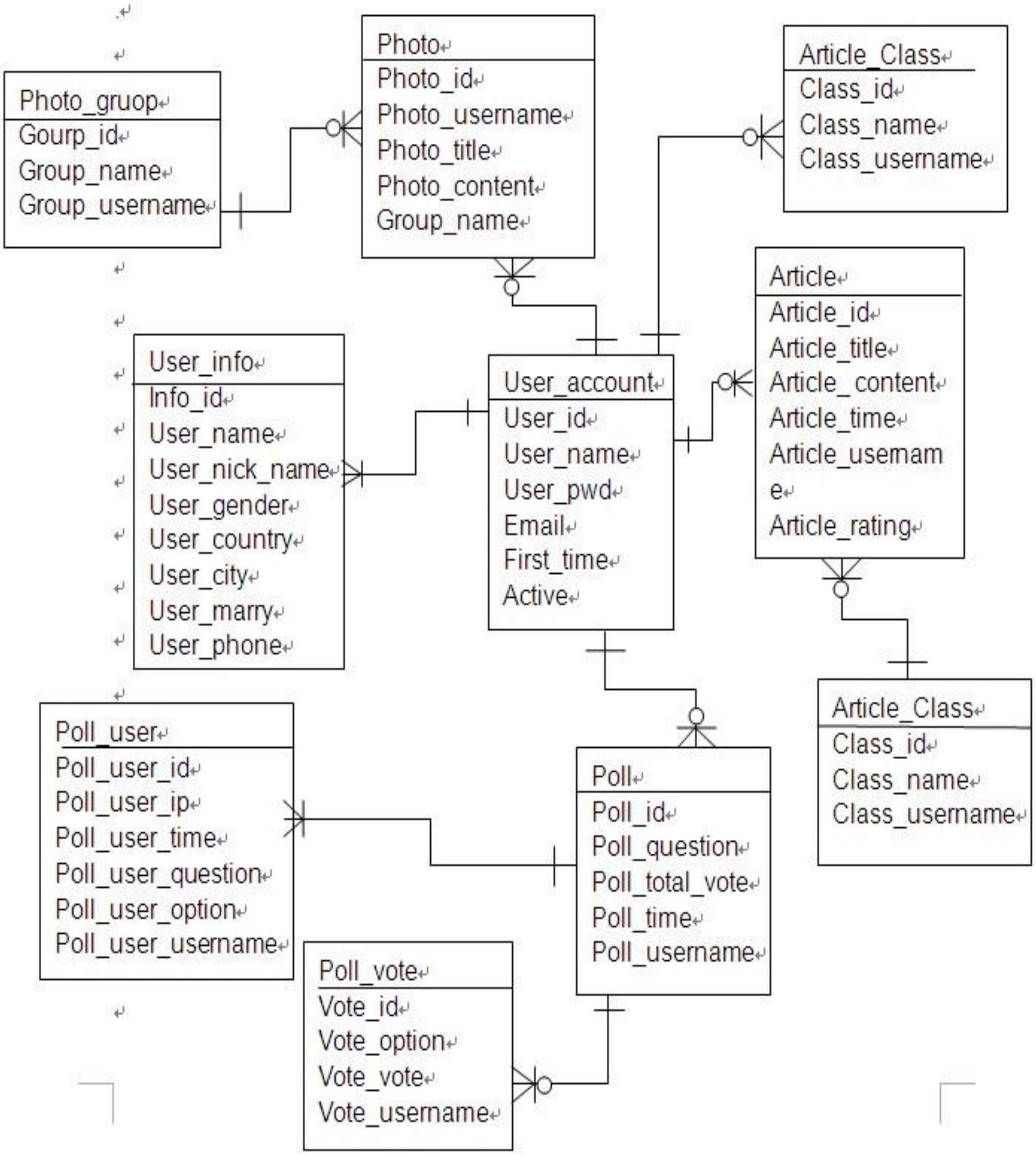


Figure 5 - System process

3.3 Database design

A pivotal setup in designing this blog is making up the database. Although this might appear to be a simple task, but sometimes a well-designed database that provides you with an easy way to get the data you want and minimizes design mistakes. That's can save your time for coding because we could not have to always update or tweak system structure. The database will be easy to modify and maintain. If you want changes a field or table, it would not negatively affect any other field or table in the database. Conversely, a bad-designed database will destroy your system. Make your system structure complicated. If you want to modify something or update some functions, is hard to find the right place and easy to make a mistake.

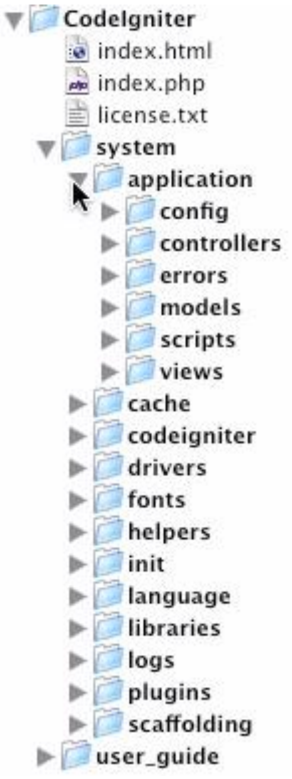
Here is the ER model for blog's database and the codes for database will be show on the end of thesis.



. Figure 6 - ER model for blog's database

3.4 Installation

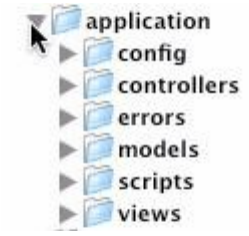
After database is really, we can start to install CodeIgniter. It is easy to install. First, need to download CodeIgniter from www.Codeigniter.com. Unlike other PHP framework, you don’t need to install anything (e.g. django framework can’t run without python environment), only need an apache server with PHP and MySQL. There is also does not have to type anything at the command line (e.g. to start django server need to run the command: python manage.py runserver), which some might be a little more comfortable with. After already installed the CodeIgniter to your server, the directory structure should be looks like this:



```
▼ CodeIgniter
    index.html
    index.php
    license.txt
  ▼ system
    ▼ application
      ▶ config
      ▶ controllers
      ▶ errors
      ▶ models
      ▶ scripts
      ▶ views
    ▶ cache
    ▶ codeigniter
    ▶ drivers
    ▶ fonts
    ▶ helpers
    ▶ init
    ▶ language
    ▶ libraries
    ▶ logs
    ▶ plugins
    ▶ scaffolding
  ▶ user_guide
```

Figure 7 - CodeIgniter system directory structure

Then we should focus on the application folder, is this where you will be working most of time and building your applications. Controllers folder is the most important part of your application, as they determine how HTTP requests should be handled. All the html web pages will be building in views folder.



application
config
controllers
errors
models
scripts
views

Figure 8 - application folder

Within that application folder:

Config – deals with the setting up of your database connection, base URL, auto loader (which loads other classes and helper functions as well as plug-in) and routes.

Controllers – are like traffic cops, directing requests to the proper models and views.

Errors – deals with your 404 errors.

Models – contains database functions such as inserting, deleting, updating and retrieving information from your database.

Scripts – a place to store your own script includes or classes.

Views – which are the pages your visitors will see. (godbit.CodeIgniter, referred 20.02.2011).

Before start to coding, there are some default config files in CodeIgniter need to be done. First is you index page of this website. Open config.php which is in the config folder (system/application/config/config.php). At line 14:

```
$config['base_url']  = "http://www.your-site.com/";
```

Edit this use your URL to replace http://www.your-site.com/. In this case, the

URL will look like this:

```
$config[‘base_url’]                                          =
“http:/www.cylyc.com/liyicheng/zone/”;
```

The next is connecting database; need open file autoload.php which is in the same folder with config.php. Add database in the libraries array that means we need to use database in this project. If you doing a project do not need database, you can skip this step.

```
$aotuload[‘libraries’] = array (‘database’);
```

Then open database.php. This file will contain the settings needed to access your database. ['hostname'] is the hostname of your database server. ['username']   ['password'] are the username and password which are used to connect to the database. ['database'] The name of the database you want to connect to

```
$db['default']['hostname'] = "hostname";
$db['default']['username'] = "username";
$db['default']['password'] = "password";
$db['default']['database'] = "database";
$db['default']['dbdriver'] = "mysql";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['char_set'] = "utf8";
$db['default']['dbcollat'] = "utf8_general_ci";
```

The final step is creating folders for icons, CSS files and jQuery files under application folder. All the jQuery files can be downloading from the link below. http://code.jQuery.com/jQuery-1.4.2min.js

3.5 Login

After finish those settings, we can start the system development. The first part is login system. Create a new PHP file on controllers and named it zone, when you name your file is must be the same name as class of controller with PHP extension. The class and file name must be match. A controller is similar like a PHP class. Function zone is to load helpers which will be used on the project. Helper is some groups of functions which can help you with coding. Each helper file is simply a collection of functions in a particular category (CodeIgniter helper, referred 22.02.2011). In the project, these helpers will be used: URL Helpers that assist in creating links, Form Helpers that help you create form elements, file helper that assist in working with files, date helper that help you work with dates. Use index function to load index.php from views folder.

```
Class zone extends Controller {
      function zone(){
         parent::Controller();
         $this->load->helper("url");
         $this->load->helper("form");
         $this->load->helper('file');
         $this->load->database();
         $this->load->helper('date');
      }
      function index(){
         $this->load->view("index");
      }
}
```

Login

a) Username min. 5 characters

b) Password min. 6 characters

A jQuery plug-in (Validation) will be used in this part to check format for each input. It is fast and efficient and provides quick replies to your visitor in the event of any errors. Use Inline-error-message display (no more alerts!) the error. It is easy to implement validation plug-in for all your live validation needs. Can be downloaded in: http://docs.jQuery.com/Plugins/validation

The next step is HTML part. After downloaded these files, create a new HTML file on views folder and named it index.html. This HTML layout will show the login form. The codes are below and the login page will be looks like Figure 9:

```
<form action="index.php/zone/main/" method="post"
class="cmxform" id="commentForm" >
      <p>
         <label for="user_name">Username</label>
         <input id="user_name" name="user_name"
class="required" minlength="5" />
      </p>
      <p>
```

```
         <label for="user_pwd">Password</label>
      <input id="user_pwd" name="user_pwd" type="password"
 class="required" minlength="6" />
      </p>
      <input class="button" type="submit" value="Login" />
</form>
```

JQuery Plugin can be activated like this:

```
$(document).ready(function() {
   $("#commentForm").validate({
   });
```

The above code is quite simple, does not need much explanation. However, there are a few important points that you need to pay attention.

- Do not forget to load the jQuery file and validation plug-in. When loading the files, the link of jQuery file must be in the 'head' section.
- Do not forget to assign the 'id' attribute of the login form and the 'name' attribute of the fields which you want to validate.

```
<script                  src="<?php                  echo
base_url();?>system/application/views/js/jQuery.js"
type="text/javascript"></script>
<script                  src="<?php                  echo
base_url();?>system/application/views/js/validate/jQuery.
validate.js" type="text/javascript"></script>
```

Then, we need to add validation rules and the respective error message. Codes will be show on the end of thesis

In this case, there are two parameters we used are:

- Rules: allow you to specify which fields you want to validate. In this case, we are validating name, password. For each field, we specify if the field is required (required: true)
- Message: allow showing the message on a particular field.

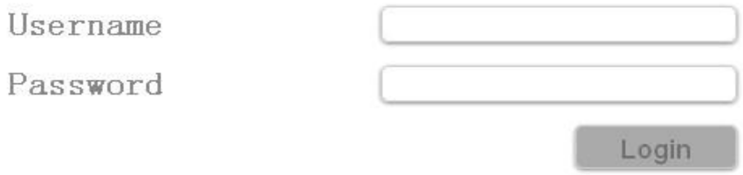


Figure 9 - Login part

If the format for each input is current, data will post to url: "zone/login_pwd" to check the user account exists which means the login_pwd function in zone.php. If the username and password are current, function will return true then you can enter your blog, otherwise will return false and show the error like username and password were incorrect or Your account has been disabled.

```
if(isset($_POST["user_name"])){
          $user_name=$_POST["user_name"];
          $valid='TRUE';
       $data["check_name"]=$this->db->query("SELECT*FROM
       user_account WHERE user_name='$user_name'");
          if($data["check_name"]->num_rows() > 0){
             echo "true";
          }
```

```
            else{
               echo "false";
            }
       }
```

3.6 User info

Basic Information

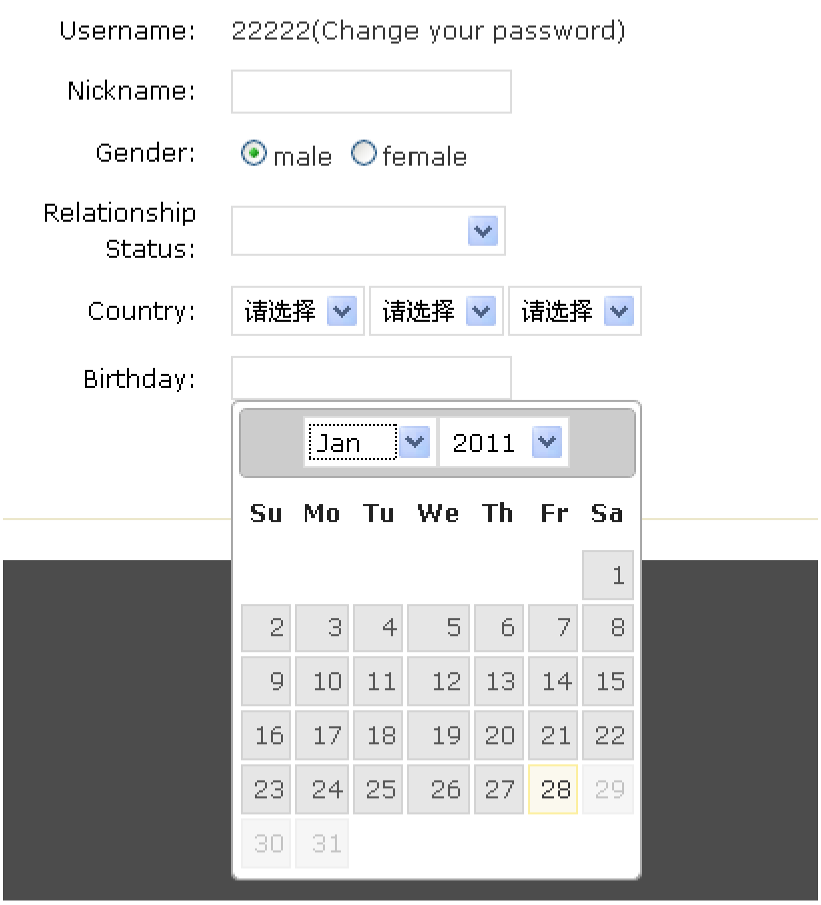
Something about personal information, you can introduce yourself from what you're doing now to where you grew up.

a) Birthday format 28.11.1983
b) Gender i.e. Male / Female
c) Country
d) City
e) Relationship Status (single, in a relationship)

In CodeIgniter, form_open() instead of the HTML <form> tag. If you need a form in your HTML code, you have to use in this way <?php echo form_open("action"); ?>, action is the way for server-side form handler and method is post. Now, we need to create a folder named user in the directory views and create a file named user_info.php and fill the codes below to make the table. Due to the large size of code, so I put it on the end of thesis.

When user is adding birthday, calendar will opens in a small overlay on Focus and closes automatically on Blur or when a date is selected. It's easier for users of the website to enter dates. For this function date format will customize to the format like day-month-year and restrict the selectable date ranges the max date is today because person cannot be born in the future.

```
$(function() {
      $("#datepicker").datepicker({
            dateFormat: 'dd-mm-yy',
            changeMonth: true,
            changeYear: true,
            maxDate: '+0M +0D'
      });
});
```



Username: 22222(Change your password)

Nickname:

Gender: ● male ○ female

Relationship Status:

Country: 请选择 请选择 请选择

Birthday:

Jan 2011

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Figure 10 - calendar

The following codes are for background system to update database after user submit the form. In CodeIgniter, SQL languages are different from PHP. You can pass an array to the function like this.

Normal PHP SQL

```
UPDATE user SET password = '{$passowrd}', name = '{$name}'
WHERE id = $id
```

CodeIgniter, SQL

```
$data = array(
            'password' => $password,
            'name' => $name,
         );
$this->db->where('id', $id);
$this->db->update('user', $data);
```

Here is the code for update data in database:

```
function user_info_insert(){
   $data=array('user_nick_name'=>$_POST["nick_name"],
      'user_sex'=>$_POST["sex"],'user_marry'=>$_POST["mar
      ry"],
      'user_country'=>$_POST["province"],'user_city'=>$_P
      OST["city"],
      'user_brithday'=>$_POST["datepicker"]);
      $user_name=$_POST["user_name"];
      $this->db->where('user_name',$user_name);
      $this->db->update('user_info', $data);
   }
```

Change password

Allow user to change password. After user type the username and old password, script will use jQuery validation plugin to check it same like login system. But it has a new validation method will be used.

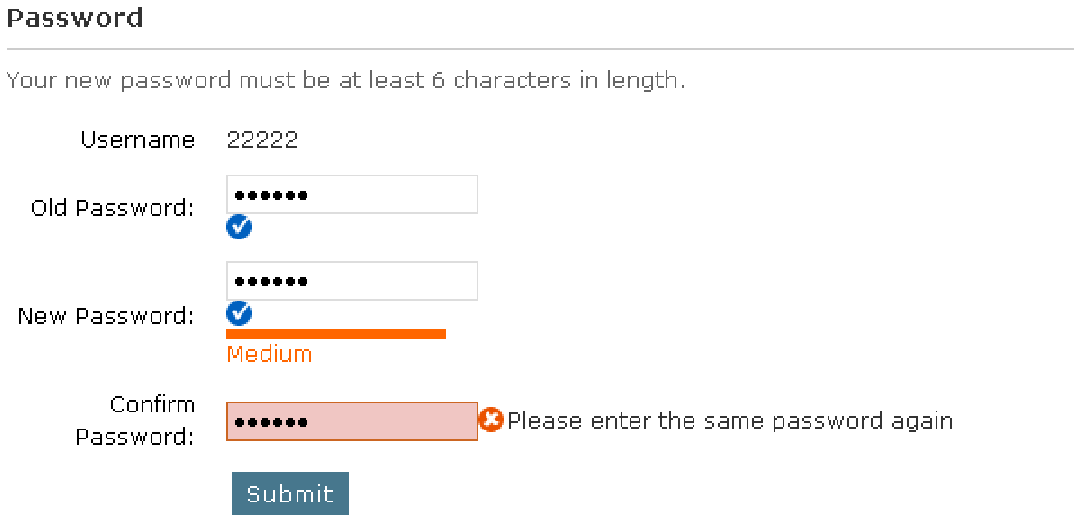
equalTo: check the element whether the another one is the same. Makes "field" required to be the same as “#password”

```
password2: {
               required: true,
               equalTo: "#password"
            }
```

When user typing password, script will get the value of the password field and checking for the character types: numbers, letters, lowercase and uppercase and special characters to evaluate the strength of the password.

To use, call the function on a password field:

```
$('.password').pstrength();
```


Password

Your new password must be at least 6 characters in length.

Username 22222

Old Password: •••••• ✓

New Password: •••••• ✓ Medium

Confirm Password: •••••• ☒ Please enter the same password again

Figure 11 - password strength

Profile picture

Upload images can be broken down into four following steps:

Create a folder named upload.

Create a user_pic.php under views/user folder. An HTML div with a browse button to allow the user to choose which file to upload.

```
<div id="left_col">
     <fieldset>
        <legend>Standard Use</legend>
           <div id="upload" >
              <span>Upload File<span>
           </div>
     </fieldset>
     <br/>
<small style="font-weight: bold; font-style:italic; font-size:13;">
Supported File Types: gif, jpg, png
</small>
```

```
    </div>
```

Use jQuery to process upload, check the file size, file type, and file extension etc. Support File Types: gif, jpg, png, jepg. max_size:2000kb.

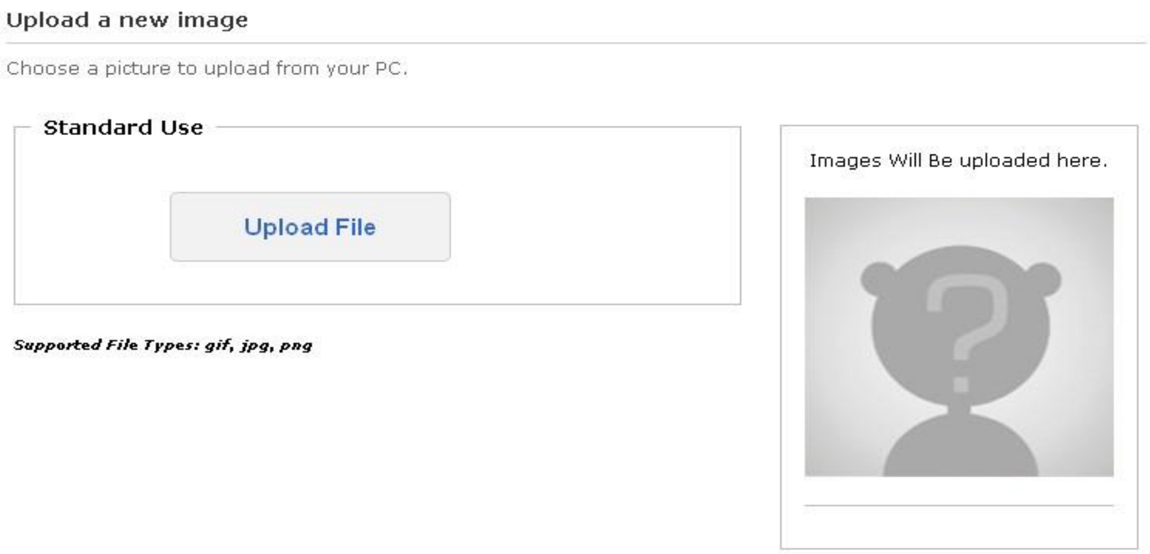
Post data to background system, at the same time show a loading image to inform user process is in progress please wait for a second.

Rename and resize the file in the background then make a copy for this image into the upload folder that you just created and insert image name into database.

Hide loading image and advice the user the upload was success.

(Otherwise, If you do not want upload any image, the system will choose a default image for you.)

In order to distinguish between all images, so we should rename it. Use a method substr to return part of name from the first letter of the image’s name to the fifteenth: $image_name = substr($image_name 0, 15). Use function date to get the time now: $time = date ("Y-m-d-H-i"). Use the time and ‘substr(name)’ to form a new name: $image_name."_".$time." Then make a copy for this image into the upload folder with the new name in order to unified image standards system will resize it and insert the new name into database. Codes for this part will be show on the end of thesis.



Upload a new image

Choose a picture to upload from your PC.

Standard Use

Supported File Types: gif, jpg, png

Figure 12 - upload image

3.7 Album

This part also needs to upload image so the code are similar like “Profile picture”. But it will use an amazing jQuery image plug-in (zoomimage) to present images. Present in stylish way. The link are unobtrusively highjacked to open the images in an inpage popup with drop shadow and border (jQuery zoomimage referred 02.03.2011).

Create the function named photo, use to get the image name from database for this user.

```
function photo(){
        $data["user_name"]=$this->uri->segment(3);
        $gruop_username=$this->uri->segment(3);
         $data["gruop"]=$this->db->query("SELECT    *    FROM
photo_gruop WHERE gruop_username='$gruop_username' order by
gruop_id desc");
      $this->load->view("file/photo/photo",$data);
   }
```

In CodeIgniter data is passed by way of an array. Need to use foreach to print it. It is different for normal PHP. When using foreach in CodeIgniter that there are no braces, need to use endforeach to replace the end brace. The control structures listed below also should follow the similar closing syntax: endif, end for, endforeach and endwhile. Also notice that instead of using a semicolon after each structure (except the last one),

There is a colon.

`foreach($db->result() as $row)` (Codeigniter ,referred 02.03.2011).

Firstly, create a photo folder under views folder and create a show_photo.php file. Codes for this part will be show on the end of thesis.

Secondly, we need to use jQuery zoomimage plugin. Download this plugin from the link below. Be sure to include jQuery files and css files in this document.

http://www.eyecon.ro/zoomimage/#implement

The final step is use jQuery present images. All the things need to do is to select the elements in a jQuery way and call the plugin.

There are 3 parameters will be used:

- Border: image’s border
- Centered: whatever if the image should be centered in the viewport or relative to the trigger.

- Hidesource: Whatever to hide source when the image is opened.

You can find an exhaustive list of parameter at http://www.eyecon.ro/zoomimage/#implement

```
$(function(){
$('a.bwGal').zoomimage({
            border: 20,
            centered: true,
            hideSource: true
      });
});
```

If user want delete an image or use an image for the cover image of the album, can go to image management page, select the image which you want it to be cover image or you want delete. Then click set cover button or delete button (below the image). Codes for management page will be show on the end of thesis.

In the code, unlink ("path") function is to delete the file and path is the path of the image, in order to migration to another server easier so need to use `$_SERVER['DOCUMENT_ROOT']` (document root) for file path. `$_SERVER['DOCUMENT_ROOT']` is very useful especially when you are working in the development environment. It make including a large number of files into the page and migration from one server to another easier.

3.8 Blog

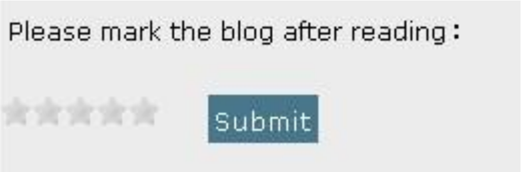
This part is much easier than others. Only need to post data and insert it like a message system. User can write your personal blog and edit or delete it. Only one different is after other user visit your blog. They can mark the blog after reading that is using a Star Rating Plug to mark it. For normal star rating plug-in just 5 stars and let users choose one of it. This one the function is much more powerful. It also has 5 stars but scoring criteria for the selection that allows the user to select from 1 to 10.

So let’s focus on the rating system. This plug-in is special but is easy to use. After install it in HTML page, and then just add the star class to the radio boxes.

Can be downloaded in:

(http://www.fyneworks.com/jQuery/star-rating/#tab-Overview)

```
<input name="rating" type="radio" class="star"/>
<input name="rating" type="radio" class="star"/>
<input name="rating" type="radio" class="star"/>
<input name="rating" type="radio" class="star"/>
<input name="rating" type="radio" class="star"/>
```



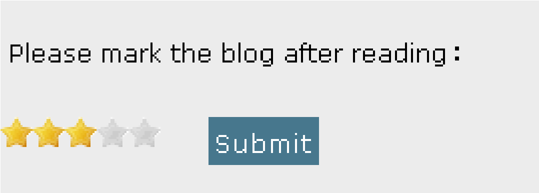
Please mark the blog after reading：

Figure 13 - rating system

In this way, we can mark it but we can’t see the result. So we need to calculate average score and insert it into database and show it on the page. Need to use the checked property to specify the initial/default value of the control

HTML code should be like this: always show the average score, if nobody marks this blog, the score is 0.

```
<input class="star {split:4}" type="radio" name="rating"
value="1" <?php if($row->article_rating==1){echo
"checked='checked'";}?> />
<input class="star {split:4}" type="radio" name="rating"
value="2" <?php if($row->article_rating==2){echo
"checked='checked'";}?> />
<input class="star {split:4}" type="radio" name="rating"
value="3" <?php if($row->article_rating==3){echo
"checked='checked'";}?> />
<input class="star {split:4}" type="radio" name="rating"
value="4" <?php if($row->article_rating==4){echo
"checked='checked'";}?> />
<input class="star {split:4}" type="radio" name="rating"
value="5" <?php if($row->article_rating==5){echo
"checked='checked'";}?> />
```



Please mark the blog after reading :

Figure 14 - rating system

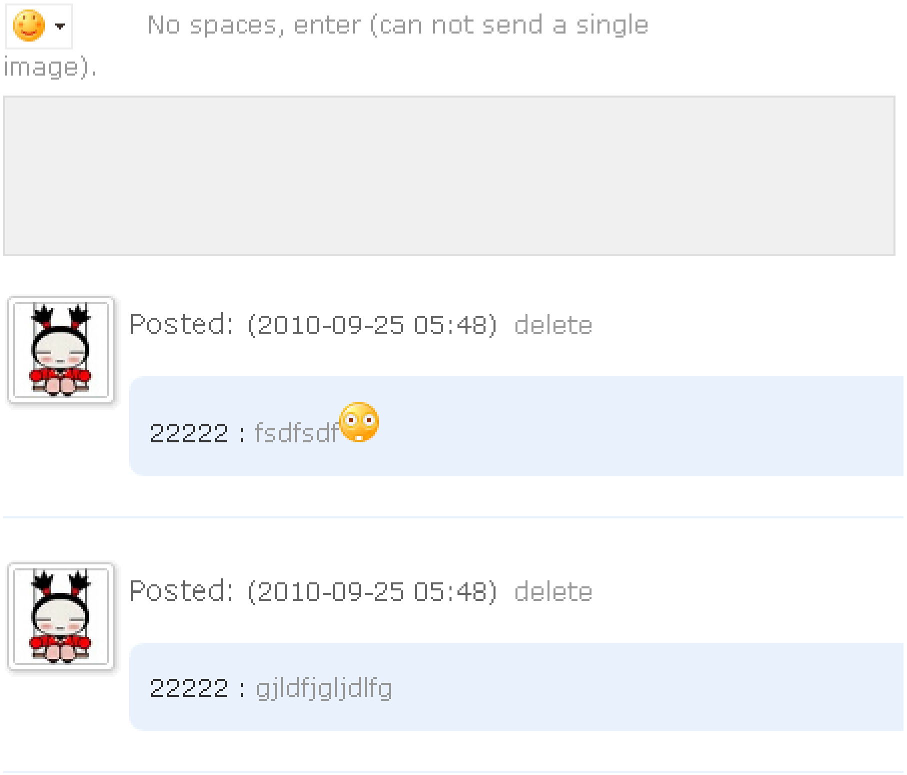
3.9 Wall

This function is for user to post and share content with his/her friends. In order to lets user display a set of smiley images on the content form, need to use facebox plug-in. It is for a Facebook style pop up window. Before begin, we need download this plug-in. Then set up an html document making sure include the necessary links to jQuery and the css needed for facebox. Now, below your JavaScript for facebox.js let's add the jQuery that makes this all work. Can be downloaded in: http://defunkt.io/facebox/. Codes for this part will be show on the end of thesis

In HTML code need a DIV link to open facebox window and hide it. After submit, if the content includes some images, script will turn the image representation (like☺) into plain text image (like *:-))*. Then use Ajax post it to background system and insert it.

Here is the code for Ajax:

```
$.ajax({
 type: "POST",
 url: "<?php echo base_url();?>index.php/zone/sign_insert",
            data: "user_name="+user_name+"&sign="+sign,
             success: function(data){
               $("#show").html(data);
            }
 });
```



No spaces, enter (can not send a single image).

Posted: (2010-09-25 05:48) delete

22222 : fsdfsdf

Posted: (2010-09-25 05:48) delete

22222 : gjldfjgljdlfg

Figure 15 - wall system

4 Conclusion

Currently, after the Email, BBS, ICQ network, blog has become another way to communicate on the internet. With the development of the network, Blog are getting more and more popular especially for teens and adults. According to incomplete statistics, it already has 40 million of blogs until now. It change the way of people’s life. When we meet, we used to ask ‘what is the weather like today’. However, now people usually ask ‘do you update your blog or do you see my comments that leave it last night ’. It represents a new way of life and new ways of working, but also represents a new approach to learning.

Consequently, blog is a place to express their thoughts, fells and a great opportunity to know some new friends. You can use it to connect with your family, friends and relatives. You can post anything on your blog. Through this, you can announce an outing or event. You can also tell your friend how you feel or your new idea.

In addition, to the development part, PHP framework is written in PHP that provides a simplified way to build the build application. In other words, PHP framework makes application development easier, which help you write code in less time because you do not have to always modify or debug some parts, it can improve code quality and reduces a lot of repetitive coding for developers. PHP framework also helps to build more stable applications especially for PHP beginner who do not have much experience in this area.

5 Discussion

I have been interested in programming since I did the practice job in an IT company. During my work practice I learn a lot of new things about programming, such as JavaScript, HTML, CSS, and PHP.

PHP is kind of a server scripting language; it is very easy to learn and use. Many functions are included without needing any sort of namespace importing. And you do not even have to write OO code if you do not want to. But PHP's ease of use is also its downfall. Because there are less restrictions on the structure of the code you write, it's much easier to write bad code. There is a solution: use a framework. Compare with other PHP framework, CodeIgniter has the most well-documented and approachable PHP MVC framework out there. Unlike other PHP frameworks, CodeIgniter works with both PHP 4 and 5. That makes the lives of someone like me who has to be able to work seamlessly between the two environments much easier. So I choose CodeIgniter as the PHP framework which will be used in the project.

Before start the project, I collected a great deal of information and material about PHP framework (CodeIgniter) from internet and books. After reading these I have clear framework concept in my mind and made this website by PHP framework. In the project of my thesis, it can divide into three parts: Information services, Album service, Message service.

Because I need to make everything by myself even the user interface and the project is quite huge. Therefore, in the thesis I cannot explain every step of making the system and also the PHP frameworks is not easy to use for the PHP beginners, it requires medium level with PHP. So in this thesis I focused more on the main part.

During the process of establish the project, I have encountered some problems. JQuery as the main tool in this project. Sadly jQuery has dropped full support for IE lower version 6. Sometime I have to use JavaScript instead of jQuery. Another problem was I am not familiar with how to use Codeigniter, always make mistakes in the project.

Made a website is not that simple as could be expected, but experience can be achieved during the process.

Reference

Printed Sources:

Cody lindley, 2009, jQuery Cookbook, Solutions & Examples for jQuery Developers: O'Reilly Media, inc

Jon A. phillips, 2007, Learning PHP & MySQL: O'Reilly Media, inc

Rebecca M. riordan, 2008, Head First Ajax: O'Reilly Media, inc

Web Design, 2008, Web Design: O'Reilly Media, inc

Electronic Sources:

Worrdpress. Blog [Referred 10.02.2011]
Available: http://codex.wordpress.org/Introduction_to_Blogging

Wiki. Blog [Referred 10.02.2011]
Available: http://en.wikipedia.org/wiki/Blog

Masternewmedia. Blog [Referred 12.02.2011]
Available:http://www.masternewmedia.org/news/2006/04/27/blog_usage_statistics_and_trends.htm

Wiki. Wall [Referred 13.02.2011]
Available: http://en.wikipedia.org/wiki/Facebook

Codeigniter [Referred 15.02.2011]
Available: http://Codeigniter.com/

Macronimous [Referred 15.02.2011]

Available:http://www.macronimous.com/resources/using_Codeigniter_for_PHP_application_development.asp

Wiki. Ajax [Referred 15.02.2011]

Available: http://en.wikipedia.org/wiki/Ajax_%28programming%29

JQuery [Referred 16.02.2011]

Available: http://jQuery.com/

Godbit.CodeIgniter [Referred 20.02.2011]

Available: http://godbit.com/article/introduction-to-code-igniter

Codeigniter. Helper [Referred 22.02.2011]

Available: http://Codeigniter.com/user_guide/general/helpers.html

JQuery. Zoomimage [Referred 02.03.2011]

Available: http://www.eyecon.ro/zoomimage/#about

Here is the code for user_info.php:

```
<?php echo form_open("zone/user_info_insert"); ?>
   <table cellspacing="0"cellpadding="0"class="formtable">
         <tbody><tr>
            <th style="width: 10em;">Username:</th>
            <td>
               <?php echo $user_name;?>
            </td>
         </tr>
         <tr>
            <th style="width: 10em;">Nickname:</th>
            <td>
               <input type="text" class="t_input" value=""
name="nick_name" id="nick_name">
            </td></tr>
         <tr>
            <th style="width: 10em;">Gender:</th>
            <td>
               <label     for="sex"><input     type="radio"
name="sex" value="male" checked>male</label>
               <label     for="sex"><input     type="radio"
name="sex" value="female">female</label>
            </td>
         </tr><tr>
            <th      style="width:      10em;">Relationship
Status:</th>
```

```
            <td>
               <select name="marry" id="marry">
                  <option value="0">&nbsp;</option>
                  <option value="Single">Single</option>
                  <option value="In a Relationship">In a
Relationship</option>
               </select>
            </td>
         </tr>
         <tr>
            <th>Country:</th>
            <td id="birthcitybox">
            <div id="test"></div>
            </td>
         </tr>
         <tr>
            <th>Birthday:</th>
            <td>
               <input      type="text"      id="datepicker"
name="datepicker" value="" class="t_input">
            </td>
         </tr>
         <tr>
            <th style="width: 10em;">&nbsp;</th>
            <td>
               <?php                                  echo
form_hidden("user_name",$this->uri->segment(3)); ?>
```

```
               <input type="submit" class="submit"
value="Submit" name="profilesubmit">
            </td>
         </tr></tbody>
      </table>
   </form>
```

Here is the code for upload image function:

```
function upload_pic() {
       $config['upload_path'] = "./uploads/pic/";
       $config['allowed_types']= 'gif|jpg|png|jpeg';
       $config['max_size']     = '2000';
       $config['max_width']   = '2000';
       $config['max_height']     = '2000';
       $this->load->library('upload', $config);
       if ($this->upload->do_upload("photo")) {
           $data=$this->upload->data();
           $this->load->library('image_lib') ;
           $img['width']   = 200;
           $img['height']  = 200;
           $img['image_library'] = 'GD2';
           $img['quality']     = '100%' ;
           $img['source_image']                          =
"./uploads/pic/".$data['file_name'] ;
          $img['maintain_ratio'] = FALSE;
           $this->image_lib->initialize($img);
           $this->image_lib->resize();
           $this->image_lib->clear() ;
         $image=$data['file_name'];
         $user_name=$this->uri->segment(3);
         $data["pic"]=$this->db->query("SELECT    *    FROM
user_account WHERE user_name='$user_name'");
         foreach($data["pic"]->result_array() as $row){
   unlink($_SERVER['DOCUMENT_ROOT'].'/liyicheng/zone/uplo
ads/pic/'.$row["image"]); }
```

```
          $data["update_pic"]=$this->db->query("UPDATE
user_account SET image='$image' WHERE
user_name='$user_name'");

          echo "success";
       }
       else {
           echo "error";
       }
}
```

Here is the code for show_photo.php:

```
<table cellspacing="0" cellpadding="0" width="100%" class="photo_list" >
      <tbody><tr>
      <?php $i=0; foreach($gruop_photo->result() as $row):?>
            <?php if ($i % 5 == 0): ?>
              </tr><tr>
            <?php endif; ?>
             <ul class="navigationTabs">
              <li><a></a></li>
             </ul>
           <div class="tab">
              <P>
              <td style="padding-bottom: 20px;">
                 <a href="<?php echo base_url();?>uploads/photo/show_photo/<?php echo $row->photo_name;?>" title="<?php echo $row->photo_title;?>" class="lightsGal">
                    <img src="<?php echo base_url();?>uploads/photo/photo/<?php echo $row->photo_name;?>"  class="mark"></a>
              </td>
              </P>
           </div>
         <?php $i++; endforeach; ?>
      </tr></tbody>
   </table>
```

Here is the code for delete photo:

```
function photo_delete(){
      $photo_name=$this->uri->segment(4);
      $gruop_name=$this->uri->segment(5);
      $data["check_pic"]=$this->db->query("select  *  FROM
photo_gruop     WHERE     gruop_name='$gruop_name'     and
gruop_username='$user_name' and gruop_pic='$photo_name'");
      $data["upload"]=$this->db->query("select * FROM photo
WHERE             gruop_name='$gruop_name'             and
photo_username='$user_name'                            and
photo_name='$photo_name'");
      $data["delete"]=$this->db->query("DELETE  FROM  photo
WHERE             gruop_name='$gruop_name'             and
photo_username='$user_name'                            and
photo_name='$photo_name'");
      $data["photo_total"]=$this->db->query("SELECT * FROM
photo      WHERE      photo_username='$user_name'      and
gruop_name='$gruop_name'");
      if($data["check_pic"]->num_rows() > 0){
         $data["check_pic"]=$this->db->query("UPDATE
photo_gruop     SET     gruop_pic='no_photo.png'     WHERE
gruop_username='$user_name'                            and
gruop_name='$gruop_name'");
      }
      if($data["upload"]->num_rows() > 0){

   unlink($_SERVER['DOCUMENT_ROOT'].'/liyicheng/zone/uplo
ads/photo/real/'.$photo_name);
```

```
   unlink($_SERVER['DOCUMENT_ROOT'].'/liyicheng/zone/uplo
ads/photo/show_photo/'.$photo_name);
         $data["gruop_photo"]=$this->db->query("SELECT   *
FROM   photo   WHERE   photo_username='$user_name'   and
gruop_name='$gruop_name'");
         $data["gruop_name"]=$this->uri->segment(5);
         $data["user_name"]=$this->uri->segment(3);

   $this->load->view("file/photo/photo_management",$data)
;
      }
      else{
         $data["gruop_photo"]=$this->db->query("SELECT   *
FROM   photo   WHERE   photo_username='$user_name'   and
gruop_name='$gruop_name'");
         $data["gruop_name"]=$this->uri->segment(5);
         $data["user_name"]=$this->uri->segment(3);

   $this->load->view("file/photo/photo_management",$data)
;
      }
   }
```

Here is the code for facebox in wall page:

```
$(document).ready(function() {
   $('.face').toggle(function(){
         $('.facebox').show();
    },function(){
         $('.facebox').hide();
   });
   $('.click_face').click(function(){
         var image = $(this).attr("src");
         var old_message = $('#sign').html();
     $('#sign').html(old_message+'<img src="'+image+'">');
     $('.facebox').hide();
   });
     $('#add').click(function(){
         var sign = $('#sign').html();
         var user_name = $('#user_name').val();
         $('#signs').val(sign);
         var sign = $('#signs').html();
     });
});
```

Here is the code for upload image with Ajax:

```
$(function(){
      var btnUpload=$('#upload');
      var status=$('#status');
      new AjaxUpload(btnUpload, {
      action:                 '<?php                 echo
base_url();?>index.php/zone/upload_pic/<?php          echo
$this->uri->segment(3);?>',
         name: 'photo',
         onSubmit: function(file, ext){
          if (! (ext && /^(jpg|png|jpeg|gif)$/.test(ext))){
               status.html('<img      src="<?php      echo
base_url();?>uploads/error.gif"></img>&nbsp&nbsp;&nbsp;On
ly JPG, PNG or GIF files are allowed');
               return false;}
            status.html('<img       src="<?php       echo
  base_url();?>system/application/views/image/loading1.gif
  "/>');},
         onComplete: function(file, response){
            status.text('');
            if(response==="success"){
               status.html('<img      src="<?php      echo
base_url();?>uploads/success.gif"></img>&nbsp&nbsp;&nbsp;
Success<br>');
               $("#imgs").attr("src", function() {
                    return           "<?php            echo
base_url();?>uploads/pic/" + file; });} else{
    $('<li></li>').appendTo('#files').text(file).addClass(
'error');
                }
            }
        });
    });
```

Here is the code for login page:

```
$(document).ready(function() {
   $("#commentForm").validate({
      success: "valid",
      rules: {
            user_name: {
               required: true,
               minlength: 5,
             },
            user_pwd: {
               required: true,
               minlength: 6,
               number: true,
               remote: {
                  url: "zone/login_pwd",
                  type: "post",
                  data: {
                     user_pwd: function(){
                        return $("#user_pwd").val();
                     },
                     user_name: function(){
                        return $("#user_name").val();
                     }
                  }
               }
              },
      },
      messages: {
            user_name: {
               required: "Please enter the username",
               remote: "Username does not exist",
               minlength: "Please enter a value greater than
or equal to 5"
            },
            user_pwd: {
               required: "Please enter the password",
               remote: "Password error or the account has not
been activated.",
               minlength: "Please enter a value greater than
or equal to 6",
               number: "Please enter only digits"
            }
         }
    });
});
```

Database creation codes for this project:

```
CREATE TABLE IF NOT EXISTS `article` (
  `article_id` int(255) NOT NULL AUTO_INCREMENT,
  `article_title` varchar(200) NOT NULL,
  `article_body` varchar(500) NOT NULL,
  `article_time` varchar(200) CHARACTER SET ucs2 NOT NULL,
  `article_username` varchar(100) NOT NULL,
  `article_rating` varchar(100) NOT NULL,
  `class_name` varchar(100) NOT NULL,
  PRIMARY KEY (`article_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=20 ;

CREATE TABLE IF NOT EXISTS `article_class` (
  `class_id` int(255) NOT NULL AUTO_INCREMENT,
  `class_name` varchar(100) NOT NULL,
  `class_username` varchar(100) NOT NULL,
  PRIMARY KEY (`class_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=11 ;

CREATE TABLE IF NOT EXISTS `photo` (
  `photo_id` int(255) NOT NULL AUTO_INCREMENT,
  `photo_username` varchar(50) NOT NULL,
  `photo_title` varchar(200) NOT NULL,
  `photo_name` varchar(200) NOT NULL,
  `gruop_name` varchar(100) NOT NULL,
  PRIMARY KEY (`photo_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=85 ;
```

```
CREATE TABLE IF NOT EXISTS `photo_gruop` (
  `gruop_id` int(255) NOT NULL AUTO_INCREMENT,
  `gruop_name` varchar(200) NOT NULL,
  `gruop_username` varchar(100) NOT NULL,
  `gruop_pic` varchar(200) NOT NULL,
  PRIMARY KEY (`gruop_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=29 ;

CREATE TABLE IF NOT EXISTS `poll` (
  `poll_id` int(255) NOT NULL AUTO_INCREMENT,
  `poll_question` varchar(200) NOT NULL,
  `poll_total_vote` varchar(900) NOT NULL,
  `poll_time` varchar(100) CHARACTER SET ucs2 NOT NULL,
  `poll_username` varchar(100) NOT NULL,
  PRIMARY KEY (`poll_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=49 ;

CREATE TABLE IF NOT EXISTS `poll_user` (
  `poll_user_id` int(255) NOT NULL AUTO_INCREMENT,
  `poll_user_ip` varchar(200) NOT NULL,
  `poll_user_time` varchar(200) NOT NULL,
  `poll_user_question` varchar(200) CHARACTER SET ucs2 NOT
NULL,
  `poll_user_option` varchar(200) NOT NULL,
  `poll_user_username` varchar(100) NOT NULL,
  PRIMARY KEY (`poll_user_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=32
```

```
CREATE TABLE IF NOT EXISTS `poll_vote` (
  `vote_id` int(255) NOT NULL AUTO_INCREMENT,
  `vote_question` varchar(200) NOT NULL,
  `vote_option` varchar(200) NOT NULL,
  `vote_vote` varchar(900) NOT NULL,
  `vote_username` varchar(100) NOT NULL,
  PRIMARY KEY (`vote_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=107 ;

CREATE TABLE IF NOT EXISTS `sign` (
  `sign_id` int(255) NOT NULL AUTO_INCREMENT,
  `sign_username` varchar(200) NOT NULL,
  `sign_body` varchar(500) NOT NULL,
  `sign_time` varchar(100) NOT NULL,
  PRIMARY KEY (`sign_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=175 ;

CREATE TABLE IF NOT EXISTS `user_account` (
  `user_id` int(255) NOT NULL AUTO_INCREMENT,
  `user_name` varchar(50) NOT NULL,
  `user_pwd` varchar(50) NOT NULL,
  `first_time` varchar(50) NOT NULL,
  `email` varchar(100) NOT NULL,
  `nameMD` varchar(50) CHARACTER SET ucs2 NOT NULL,
  `captcha` varchar(50) CHARACTER SET ucs2 NOT NULL,
  `active` varchar(50) NOT NULL,
  `image` varchar(200) NOT NULL
  PRIMARY KEY (`user_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;

CREATE TABLE IF NOT EXISTS `user_info` (
  `info_id` int(255) NOT NULL AUTO_INCREMENT,
  `user_name` varchar(50) NOT NULL,
  `user_nick_name` varchar(50) NOT NULL,
  `user_sex` varchar(50) NOT NULL,
  `user_country` varchar(50) NOT NULL,
  `user_city` varchar(50) NOT NULL,
  `user_city2` varchar(50) NOT NULL,
  `user_brithday` varchar(50) NOT NULL,
  `user_marry` varchar(50) NOT NULL,
  `user_phone` varchar(50) NOT NULL,
  `user_qq` varchar(50) NOT NULL,
  PRIMARY KEY (`info_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;

CREATE TABLE IF NOT EXISTS `zone_game` (
  `game_id` int(255) NOT NULL AUTO_INCREMENT,
  `game_title` varchar(50) NOT NULL,
  `game_body` varchar(500) NOT NULL,
  PRIMARY KEY (`game_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=7 ;

CREATE TABLE IF NOT EXISTS `zone_gift` (
  `gift_id` int(255) NOT NULL AUTO_INCREMENT,
  `gift_title` varchar(50) NOT NULL,
  `gift_body` varchar(500) NOT NULL,
  PRIMARY KEY (`gift_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=7 ;


CREATE TABLE IF NOT EXISTS `zone_notice` (
  `notice_id` int(255) NOT NULL AUTO_INCREMENT,
  `notice_title` varchar(50) NOT NULL,
  `notice_body` varchar(500) NOT NULL,
  PRIMARY KEY (`notice_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```