

Database Project Report

CS 6083 INET, Spring 2016

Group Member:

Sijie Liu (N11120596)
Chenglong Sun (N17840926)

Date: May 8th 2016
20 Pages

Contents

1. Introduction & Design method

2. Main function

- 2.1 Sign up & log in
- 2.2 My welcome page
- 2.3 My index
- 2.4 Post
- 2.5 Social features
- 2.6 Personal index
- 2.7 Search function

3. Friends relationship

- 3.1 Friend request
- 3.2 Accept/Decline friend request
- 3.3 My friends & FOF

4. Post diary & activity

- 4.1 Post a diary or activity
- 4.2 Add a tag
- 4.3 Add a location
- 4.4 Privilege of posts
- 4.5 Like a post

5. Some features

- 5.1 People you might know
- 5.2 Posts you might like
- 5.3 Connection with Google maps

1. Introduction

Our database project is based on one of the most popular social software, Instagram, it offers a platform where people could post photos, videos to share happiness with each other. Users could attach not only the location to their posts, but also the tags to their posts, every photo could have a special tag of itself. “Like” is also a special way to express one’s emotion to some post, it makes our social website more colorful. We fetch these special features and used our knowledge of database and php, html to create our project --- the dark forest (inspired by the novel “Three Body”).

Design method

Our social website mainly concentrates on management of users, friendship and post. A user starts from sign up, then log in system to operate his personal profile, post a diary to record his daily life, see his friends post. He may make friends with others, socialize with others through some features in the post. He could explore his friendship and his interest with the search function inserted in system. System will recommend some new friends or post match the user’s interest as well.

2. Main function

2.1 Sign up & log in

Sign up in system:

Sign up to share your life with the world.

Harry

.....

.....

Harry@gmail.com

Sign up

After new user typed in their information, we will check the legality of their information:

```
//Judge the register information legality
if(!preg_match('/^[\w\x80-\xFF]{3,15}$/, $username)){//Check the username
    exit('Wrong!: username is illegal<a href="javascript:history.back(-1);">Back</a>');
}
if(strlen($password) < 6){//Check the password
    exit('Wrong!: the password length is too short<a href="javascript:history.back(-1);">Back</a>');
}
if(!preg_match('/^([a-zA-Z0-9][_-].*)*[a-zA-Z0-9]@([a-zA-Z0-9][_-].*)*[a-zA-Z0-9].[a-zA-Z]{2,3}$/, $email)){//Check the email address
    exit('Wrong: the format of email is wrong<a href="javascript:history.back(-1);">Back</a>');
}
include('conn.php');//Connection with database
//Check whether this username has existed
$ccheck_query = mysql_query("select uid from user where username='$username' limit 1");
if(mysql_fetch_array($ccheck_query)){
    echo 'Wrong: the username of '$username' has existed.<a href="javascript:history.back(-1);">Back</a>';
    exit;
}
```

If the register information is legal, we will insert new user's information into our database system:

```
$sql = "INSERT INTO user(username,password, email)VALUES('$username','$password', '$email')";
if(mysql_query($sql,$conn)){
    //Insert the information into the database
    exit('Register Success ! Click here to <a href="login.html">log in</a>');
} else {
    //Register failed and retry
    echo 'Sorry! Register failed: ',mysql_error(),'<br />';
    echo 'Click here to <a href="javascript:history.back(-1);">Back</a> Retry';}
```

username	password	email	register_time
Harry	000000	harry@gi	2016-05-04 17:25:22

Figure 2.1 New user information appears in database system

Log in the system:



The image shows a login screen for a game called "THE DARK FOREST". The title is displayed prominently in large, orange, blocky letters against a dark, futuristic background featuring a circular structure and glowing particles. Below the title, a subtext reads "Log in with your Dark Forest account." A login form is present, containing two input fields labeled "Sijie" and "....." (representing a password), and a "Log in" button. At the bottom of the form, there is a link "Don't have an account? [Sign up](#)".

Figure 2.2 Log in screen shot, login.html

We get the log in information through function InputCheck

```
function InputCheck(LoginForm){
if (LoginForm.username.value == ""){
    alert("Type in the username!");
    LoginForm.username.focus();
    return (false);
}
if (LoginForm.password.value == ""){
    alert("Type in the password!");
    LoginForm.password.focus();
    return (false);}}
```

Connect the system with the database in localhost

```
*****Connect with database*****
$conn =
@mysql_connect("localhost","root","");
if (!$conn){
die("connection failed: " . mysql_error());}
```

The login program

```
date_default_timezone_set('US/Eastern'); //Set the time zone at American Eastern Area
header("Content-type: text/html; charset=utf-8");
session_start();
//Log in
if(!isset($_POST['submit'])){
exit('Illegal log in!');} //Log in failed
$username = htmlspecialchars($_POST['username']);
$password = $_POST['password'];
//Make sure the system is connected with my localhost database
include('conn.php');
//Check the username and password in system
$check_query = mysql_query("select uid from user where username='$username' and
password='$password'
limit 1");
if($result = mysql_fetch_array($check_query)){
//Log in successfully!
$_SESSION['username'] = $username;
$_SESSION['userid'] = $result['uid'];
echo $username, ' Welcome ! ';
...
} else { //Log in failed
exit('Permission denied! Click <a href="javascript:history.back(-1);">Back</a> ');}

```

Sijie Welcome! Enter [My homepage](#)
Click here [logout](#)
Your new friend request:

Figure 2.3 Log in successfully

That is not a match! [Back](#) Try again

Figure 2.4 Log in failed

2.2 My welcome page

This is the page appears after the user successfully logged in, we will show the user sever options, he could check the friends request, go to the News Feed and log out. Here is several main program:

```

$check_query = mysql_query("select uid from user where username='".$username' and password='".$password' limit 1");
if($result = mysql_fetch_array($check_query)){
    //Log in successfully!
    $_SESSION['username'] = $username;
    $_SESSION['userid'] = $result['uid'];
    //Go to the My News Feed
    echo $username,' Welcome ! Go to <a href="my.php">My index</a><br />';
    //Log out
    echo 'Click here <a href="logout.php?action=logout">Log out</a><br />';
    $uid = $_SESSION['userid'];
    $lastlogin = date("Y-m-d H:i:s");
    $sqllastlogin = mysql_query("UPDATE `user` SET `last_login` = '$lastlogin' WHERE `user`.`uid` = '$uid'");
    //update last login time
    echo 'You have these friend requests:<br />';} else { //Log in failed
        exit('Permission denied ! Click <a href="javascript:history.back(-1);">Back</a> ');}
    //Check friends request and select friend request sender
    $userid = $result['uid'];
    $my_friends = mysql_query("SELECT uid, username FROM `friends` JOIN `user` WHERE sender_id = uid AND receiver_id = $userid AND status = 'pending'");
    while($row = mysql_fetch_array($my_friends))
    {//Click name to go to the personal index
        echo '<a href="myindex.php?username='.$row[1].'">'.$row[1].'</a>';
        echo "requests to be your friend<br />";
    //Accept/Decline the request
        echo '<a href="fr_accept.php?receiver_id='.$userid.'&sender_id='.$row[0].'">'.'Accept'.'</a>';
        echo '/';
        echo '<a href="fr_decline.php?receiver_id='.$userid.'&sender_id='.$row[0].'">'.'Decline'.'</a>';
        echo '<br />';}

```

Chenglong Welcome! Enter [My homepage](#)
 Click here [logout](#)
 Your new friend request:
 Sijie sent you a friend request.
[Accept](#)/[Decline](#)

Figure 2.5 my welcome page

2.3 My index

On this page, user could edit his personal profile, post a new diary, search some his interested things and see his friends post in these days.

Figure 2.6 Edit personal profile, post a new diary

```
//Edit my profile edit.php
if ($realname != "") {
    $sqlrealname = "UPDATE `user` SET `real_name` = '$realname' WHERE `user`.`uid` = '$uid'";
    if(mysql_query($sqlrealname,$conn)){
        echo 'realname edit successfully!','<br />';
    } else {
        echo 'Sorry! Add my information failed: ',mysql_error(),'<br />'; }
}
if ($age != "") {
    $sqlage = "UPDATE `user` SET `age` = '$age' WHERE `user`.`uid` = '$uid'";
    if(mysql_query($sqlage,$conn)){
        echo 'age edit successfully!','<br />';
    } else {
        echo 'Sorry! Add my information failed: ',mysql_error(),'<br />'; }
}
if ($gender != "") {
    $sqlgender = "UPDATE `user` SET `gender` = '$gender' WHERE `user`.`uid` = '$uid'";
    if(mysql_query($sqlgender,$conn)){
        echo 'gender edit successfully!','<br />';
    } else {
        echo 'Sorry! Add my information failed: ',mysql_error(),'<br />'; }
}
if ($city != "") {
    $sqlcity = "UPDATE `user` SET `city` = '$city' WHERE `user`.`uid` = '$uid'";
    if(mysql_query($sqlcity,$conn)){
        echo 'city edit successfully!','<br />';
    } else {
        echo 'Sorry! Add my information failed: ',mysql_error(),'<br />'; }
}
```

My Friend List
[He](#)
[Sijie](#)

Other people you may want to know
[Yi](#)
[Yali](#)
[Back to my homepage](#)

Figure 2.7 show my friend list and people I might know

```
//Show my friends list
echo "My Friend List","<br />"; //show my friends
while($row = mysql_fetch_array($my_friends))
{
echo '<a href="myindex.php?userid='.$row[0].'&myid='.$userid.'">'.$row[1].'</a>';
echo "<br />";
}
?>
</div>
<div class="_fqv9h">
<?php
echo "Other people you may want to know","<br />"; //show my friends of friends
while($row = mysql_fetch_array($my_FOF))
{
echo '<a href="myindex.php?userid='.$row[0].'&myid='.$userid.'">'.$row[1].'</a>';
echo "<br />";
}
```

[Sijie](#)

Title: My school

NYU is great!!!

Upload time: 2016-05-08 17:19:06

location: NYU lat: 31 lng: -106

This post is about: NYU



NYU

[He](#)

Title: Bobst

Let's study together in bobst library this Sunday

Upload time: 2016-04-10 14:54:07

location: nyc lat: 678 lng: 345

This post is about: study



Figure 2.8 My friends' posts (Diary and activity)


```

echo 'location: '.$row['location_name'],' lat: '.$row['latitude'],' lng: '.$row['longitude'],'<br />'; }
if($check_tag = mysql_query("SELECT * FROM tag WHERE post_id = $postid")){
    if ($row = mysql_fetch_array($check_tag)) {
        echo 'This post is about: '.$row['tag'].'';
        while ($row = mysql_fetch_array($check_tag)) {
            echo ''.$row['tag'].'';
        }
        echo "<br />";?
    }
<li><p><a href="#" title="like" class="img_on" rel="<?php echo $postid;?>"><?php echo $likes;?></a></p></li>
<?php
} else {
$postid = $row['post_id'];
$likes = $row['likes'];
$title = $row['title'];
echo 'Title: '.$row['title'],'<br />';
echo $row['text'],'<br />';
echo 'Upload time: '.$row['upload_time'],'<br />';
$url = $row['url'];
$locationid = $row['location_id'];
if ($check_location = mysql_query("SELECT * FROM location WHERE location_id = $locationid")) {
    $row = mysql_fetch_array($check_location);
    echo 'location: '.$row['location_name'],' lat: '.$row['latitude'],' lng: '.$row['longitude'],'<br />'; }
if($check_tag = mysql_query("SELECT * FROM tag WHERE post_id = $postid")){
    if ($row = mysql_fetch_array($check_tag)) {
        echo 'This post is about: '.$row['tag'].'';
        while ($row = mysql_fetch_array($check_tag)) {
            echo ''.$row['tag'].'';
        }
        echo "<br />";?
    }
<li>"><p><a href="#" title="like" class="img_on" rel="<?php echo $postid;?>"><?php echo $likes;?></a></p></li>

```

2.4 Post a diary or activity

New Post

Post Type: Activity

Title:

Text:

Privilege: All

Multimedia: 选择文件 未选择任何文件

Figure 2.9 Post a new diary or activity

User could post a diary with photo or video or an activity as a new post, he may add a title text, privilege and a multimedia in this new post. As to the privilege, he could choose post this diary to all (everyone logged in this website), his friends, his friends of friends, private(only seen by himself). Multimedia could be a url of a photo or video from other website, it also could be a file on the poster's computer. After the user posted a diary or activity, he will be noticed to attach a location and tags with his new post, a location could tell readers where it happened and a tag could add a special mark of the poster's own, such like #nyc, #myfirstcar, etc.

2.5 Social features

In the user's post list, there is a "like" button, we record the "like"s if the user click this button, and let the number of "like"s appear under the posted photo, the more "like"s the post got, the more popular this post will be.



We fetch the userid and his IP address of the user to record his like to one post, so that one post could be liked by one user by only once, if he click the "like" button twice, the system will notice him "You have liked this post".

2.6 Personal interface

Every user has a personal interface, which show the profile and posts of the user. It provides a "add friend" button to send a friend request to the user. And the posts appear in different user will also have some difference, depending on the privilege of the posts. If a post is a sent to ALL diary, everyone could see it; otherwise, if it's a sent to FRIENDS one, only the user's friends could see it; if it's a FOF one, and so on.

2.7 Search function

In our website we offer a search function, user typed in some keyword, system will result in some user, location, tag, post which contains this keyword, click their names, you could get the information in detail.

```
$search_friends = mysql_query("select * from `user` where  
(`username` like '%$keyword%'");  
echo "These are users contain \"";  
echo $keyword;// Out put the users whose name contains the  
keyword  
echo " " :;  
echo "<br />";  
while($row1 = mysql_fetch_array($search_friends))  
//Show the tables, then save the selected table name as  
"tablename"  
echo '<a  
href="myindex.php?userid='.$row1['uid'].'&myid='.$myid.'">'.$row  
1['username'].':</a>';  
echo "<br />"; }
```

We just introduced the keyword search in users, so as to the location, tags, post.

User information:
ID: 12
Email: 0
Register time: 2016-04-10 13:43:38
Realname: Sijie Liu
Gender: male
City: bklyn
Age: 24
Last log in: 2016-05-08 20:13:39
Last log out: 2016-05-08 20:28:26

[Add Friend](#)

Sijie
Title: Campus
NYU
2016-05-08 20:26:42

location: NYU lat: 31 lng: -106
This post is about: NYU



Figure 2.10 Personal interface

Users about C:
[Chenglong](#)
Tags about C:
Locations about C:
[nyc](#)
Posts about C:
[Back to my homepage](#)

Figure 2.11 Search keyword “C” result

3. Friends relationship

3.1 Friend request

When one user wants to add another user as a friend, he should send a friend request. After the request was created, the system will insert a pending statue in friends table, meanwhile, this request information will appear at the receiver’s welcome page.

User information:
ID: 15
Email: 0
Register time: 2016-04-10 13:45:43
Realname: He Huang
Gender: male
City: manhattan
Age: 23
Last log in: 2016-02-02 11:00:00
Last log out: 2016-02-13 11:00:00

[Add Friend](#)

He
Title: Bobst
Let's study together in bobst library this Sunday
2016-04-10 14:54:07

location: nyc lat: 678 lng: 345
This post is about: study



Figure 3.1 Add friend in personal interface

Request sent! Click here [Back](#)

Figure 3.2 Sent a friend request

```
$sql = "INSERT INTO friends(sender_id,receiver_id)VALUES('$myid','$userid');"
```

		sender_id	receiver_id	status
<input type="checkbox"/>	编辑 复制 删除	12	13	accepted
<input type="checkbox"/>	编辑 复制 删除	12	14	accepted
<input type="checkbox"/>	编辑 复制 删除	12	15	pending

Figure 3.3 Friend status changed in friends database(receiver_id = 15)

3.2 Accept/Decline friend request

After a user logged in the system, the welcome will show friends requests after the user last logout, he could see the personal index of the request sender by click the sender's name. Accept or decline is very easy to go, we will update the friends status into "accepted" or "declined", to establish the relationship between these two people.

```
//Accept the request
$accept = mysql_query("UPDATE `friends` SET `status`='accepted' WHERE sender_id =
$sender AND receiver_id = $receiver");
//Decline the request
$accept = mysql_query("UPDATE `friends` SET `status`='declined' WHERE sender_id =
$sender AND receiver_id = $receiver");
```

He Welcome! Enter [My homepage](#)
 Click here [logout](#)
 Your new friend request:
Sijiesent you a friend request.
[Accept/Decline](#)

Figure 3.4 Friend request appear in my welcome index

Add friend success!
 Click here [Back](#)

Figure 3.5 Accept the request

		sender_id	receiver_id	status
<input type="checkbox"/>	编辑 复制 删除	12	13	accepted
<input type="checkbox"/>	编辑 复制 删除	12	14	accepted
<input type="checkbox"/>	编辑 复制 删除	12	15	accepted

Figure 3.6 Change in the friends database(receiver_id = 15)

3.3 My friends & FOF

In this system we have a further definition of friends, because there are two ways to become friends, one is request someone to be a friend of mine and he accepted, the other way is accept the request from someone.

```
//Select my friends' userid
(SELECT sender_id FROM `friends` WHERE receiver_id = $userid AND status = 'accepted')
UNION
(SELECT receiver_id FROM `friends` WHERE sender_id = $userid AND status = 'accepted')
```

As to friends of friends(FOF for short), we define them as the friends of my friends, but this group of people does not contain those people already have been my friends.

```
//Select my FOFs' userid
$my_FOF = mysql_query("SELECT uid, username FROM `friends` JOIN `user` WHERE receiver_id =
uid AND STATUS = 'accepted' AND receiver_id <>
$userid AND sender_id IN (
SELECT sender_id FROM `friends` WHERE receiver_id = $userid AND STATUS = 'accepted')
UNION
SELECT uid, username FROM `friends` JOIN `user` WHERE receiver_id = uid AND STATUS =
'accepted' AND receiver_id <>
$userid AND sender_id IN
(SELECT receiver_id FROM `friends` WHERE sender_id = $userid AND STATUS = 'accepted')
UNION
SELECT uid, username FROM `friends` JOIN `user` WHERE sender_id = uid AND STATUS =
'accepted' AND sender_id <> $userid
AND receiver_id IN (
SELECT sender_id FROM `friends` WHERE receiver_id = $userid AND STATUS = 'accepted')
UNION
SELECT uid, username FROM `friends` JOIN `user` WHERE sender_id = uid AND STATUS =
'accepted' AND sender_id <> $userid
AND receiver_id IN
(SELECT receiver_id FROM `friends` WHERE sender_id = $userid AND STATUS = 'accepted')");
```

Both Friends and FOF will play an important role in the post section, it will decide whether this post will be seen by everyone, my friends, my FOF or only by myself. We will talk about this in details in section 4[Post diary & activity].

4. Post diary & activity

New Post
Post Type: Activity
Title:
Text:
Privilege: All
Multimedia: 选择文件 未选择任何文件
Post

4.1 Post a diary or activity

User could post a diary with photo or video or an activity as a new post, he may add a title text, privilege and a multimedia in this new post. As to the privilege, he could choose post this diary to all (everyone logged in this website), his friends, his friends of friends, private(only seen by himself). Multimedia could be a url of a photo or video from other website, it also could be a file on the poster's computer.

Figure 4.1 Post a new diary or activity interface

```
<div>
<fieldset>
<legend>New Post</legend>
<form name="PostForm" method="post" action="newpost.php" onSubmit="return
PostInputCheck(this)">
<p>
<label for="PostType" class="label">Post Type:</label> //Post type
<select name="PostType" class="input"/>
<option value="Activity">Activity</option>
```

```

<option value="Diary">Diary</option>
</select>
<p/>
<p>
<label for="Title" class="label">Title:</label> //Title
<input id="Title" name="Title" type="text" class="input" />
<p/>
<p>
<label for="Text" class="label">Text:</label> //Text
<textarea id="Text" name="Text" rows="10" cols="30" class="input"/></textarea>
<p/>
<p>
<label for="Privileges" class="label">Privilege:</label> //Set privileges
<select name="Privileges" class="input">
<option value="All">All</option>
<option value="Friends">Friends</option>
<option value="Friends and FOFs">Friends and FOFs</option>
<option value="Private">Private</option>
</select>
<p/>
<p>
<label for="Multimedia" class="label">Multimedia:</label> //Multimedia
<input type='text' name='textfield' id='textfield' class='txt' />
<!--<input type='button' class='btn' value='浏览...' /-->
<input type="file" name="fileField" class="file" id="fileField" size="28"
onchange="document.getElementById('textfield').value=this.value" />
<span>Required for diary post</span>
<p/>
<p>
<input type="submit" name="submit" value=" Post " class="left" />
</p>

```

4.2 Add a tag

After the user posted a diary or activity, he will be noticed to attach tags with his new post, a tag could add a special mark of the poster's own, such like #nyc, #myfirstcar, etc.

```

//Add a tag to a post
<form action="addtagtopost.php" method="post" name="tag" id="tag">
<input type="text" name="postid" value="<?=$postid?>">
    <input name="tag" type="text" id="tag" />
    <input name="Submit" type="submit" value="Add" />
</form>

```

postid: 19	nba	Add
------------	-----	-----

Figure 4.2 Add a tag to post (show in php)

	<input type="button" value="←"/>	<input type="button" value="→"/>	▼	post_id	1	tag
<input type="checkbox"/>		编辑		复制		删除
				19	nba	
<input type="checkbox"/>		编辑		复制		删除
				7	study	
<input type="checkbox"/>		编辑		复制		删除
				6	vacation	

Figure 4.3 Add a tag to post (show in database)

```
$tag_post = mysql_query("SELECT * FROM `post` WHERE `post_id` = $postid ");
while($row = mysql_fetch_array($tag_post)) {
    //Show the tags in post
    echo '<a href="post.php?post_id='.$row['post_id'].'&uid='.$row['uid'].'"">'.$row['title'].'</a>';
    echo "<br />"; }
```

Title: NBA

funny

Upload time: 2016-05-08 19:58:48

This post is about: NBA

[Add Tag](#)



Figure 4.4 A new post with tag “NBA”

4.3 Add a location

After the user posted a diary or activity, he will be noticed to attach a location with his new post, a location could tell readers where it happened. The user may not only search a location already exists in the database, he might but also create a new location in our database with his created name.

[nyc](#)

Can not find the location [Create new location](#)

Figure 4.5 Search a location named “nyc”

We offer the search function of the latitude and longitude information for a location on the Google map, so that every new location created by users might be a real existed place in the world. We fetch the latitude and longitude from Google map API, so that we could give the users accurate latitude and longitude number of the special place for their use.

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false&language=en"></script>
<script type="text/javascript">
var geocoder;
var map;
function initialize() {
geocoder = new google.maps.Geocoder();
var myOptions = {
```

```

zoom : 17,//adjust the proportion of zoom
//•MapTypeId.HYBRID •MapTypeId.TERRAIN
mapTypeId : google.maps.MapTypeId.ROADMAP }
map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
codeAddress(); }

function codeAddress() {
var address = document.getElementById("address").value;;
//analysis of the location
geocoder.geocode( { 'address' : address }, function(results, status) {
if (status == google.maps.GeocoderStatus.OK) {
//keep the location coordinate in the center of screen
map.setCenter(results[0].geometry.location);
var marker = new google.maps.Marker({
map : map,
position : results[0].geometry.location,
title : address,
// Animation realization of the location coordinate
animation : google.maps.Animation.DROP });
var display = "<b>address: </b>" + results[0].formatted_address;
var position = results[0].geometry.location;
var infowindow = new google.maps.InfoWindow({
content : "<span style='font-size:11px'><b>name: </b>" +
+ address + "<br>" + display + "<br/><b>location:</b>" + position + "</span>",
leftpixelOffset : 0,
position : results[0].geometry.location });
//output the location coordinate and the real location in the Google Map
infowindow.open(map, marker);
google.maps.event.addListener(marker, 'click', function() {
infowindow.open(map, marker); });
} else {
alert("Geocode was not successful for the following reason: " + status); });
}); }

</script>

```

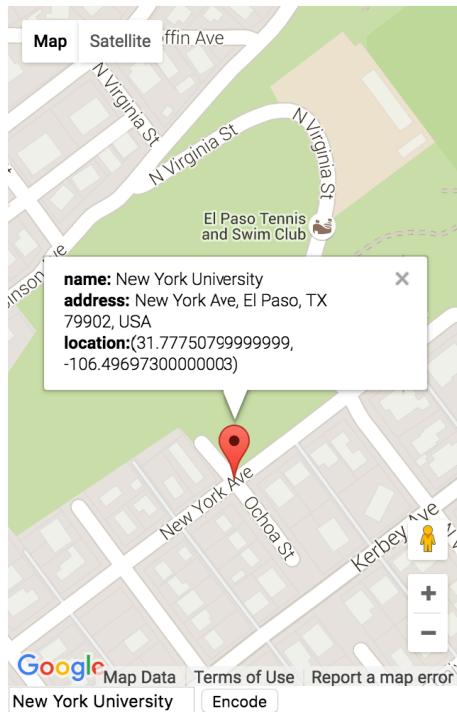


Figure 4.6 Show latitude and longitude in Google map(New York University)

After we got the location coordinate of the place we want to create, we just type in the latitude, longitude and location name of our own, then we created a new location data in the database.

```
$newlocation = mysql_query("INSERT INTO `location` (`latitude`, `longitude`, `location_name`)
VALUES ($lat, $lng, '$name')");
```

new location id:49
post id:32
Create location success!
[Back](#)

Figure 4.7 Create location success!

longitude	latitude	location_id	location_name
-106	31	49	NYU

Figure 4.8 A new location named “NYU” was added into location database

4.4 Privilege of posts

There are four privileges of posts: All, Friends, FOF, Private

post_id	uid	title	text	upload_time	location_id	type	url	privileges	likes
32	15	NYC	Empire States	2016-05-08 20:01:34	49	Diary	[BLOB - 69 字节]	All	0
31	15	NBA	funny	2016-05-08 19:58:48	NULL	Diary	[BLOB - 66 字节]	Friends	0

Figure 4.9 Different privileges of posts from same poster

He
Title: NYC
 Empire States
 Upload time: 2016-05-08 20:01:34
 location: NYU lat: 31 lng: -106



Figure 4.10 He's posts seen by a stranger

He
Title: NYC
 Empire States
 Upload time: 2016-05-08 20:01:34
 location: NYU lat: 31 lng: -106

He
Title: NBA
 funny
 Upload time: 2016-05-08 19:58:48
 This post is about: NBA



Figure 4.11 He's posts seen by his friend

4.5 Like a post

We fetch the userid and his IP address of the user to record his like to one post, so that one post could be liked by one user by only once, if he click the “like” button twice, the system will notice him “You have liked this post”.

```
$ip = get_client_ip();
$pid = $_POST['id'];
if(!isset($pid) || empty($pid)) exit;
$sql=mysql_query("select ip from post_ip where pid='$pid' and ip='$ip'");
$count=mysql_num_rows($sql);
if($count==0){
    $sql = "update post set likes=likes+1 where post_id='$pid'";
    mysql_query( $sql);
    $sql_in = "insert into post_ip (pid,ip) values ('$pid','$ip')";
    mysql_query( $sql_in);
    $result = mysql_query("select likes from post where post_id='$pid'");
    $row = mysql_fetch_array($result);
    $love = $row['likes'];
    echo $love;
} else{
    echo " Liked";
}
```

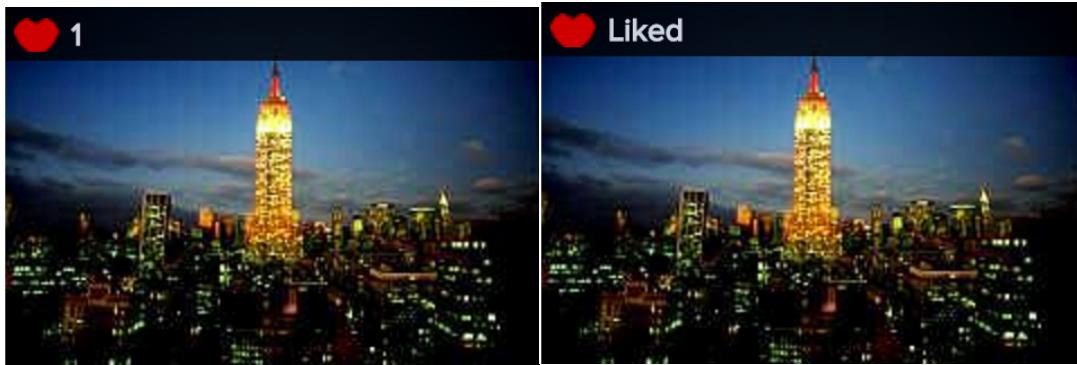


Figure 4.12 Like a photo and like it twice

5. Some features

5.1 People you might know

It is a social website we designed, so the social function is essential. In our website, we will recommend people that user might know. We will check the friend list, find all friends of the user, and then get a list of FOF (friends of friends), the more mutual friends they have, the more recommended the person will be.

```
//List people you might know
$my_FOF = mysql_query("SELECT uid, username FROM `friends` JOIN `user` WHERE receiver_id =
uid AND STATUS = 'accepted' AND receiver_id <>
$userid AND sender_id IN (
SELECT sender_id FROM `friends` WHERE receiver_id = $userid AND STATUS = 'accepted')
UNION
SELECT uid, username FROM `friends` JOIN `user` WHERE receiver_id = uid AND STATUS =
'accepted' AND receiver_id <>
$userid AND sender_id IN
(SELECT receiver_id FROM `friends` WHERE sender_id = $userid AND STATUS = 'accepted')
UNION
SELECT uid, username FROM `friends` JOIN `user` WHERE sender_id = uid AND STATUS =
'accepted' AND sender_id <> $userid
AND receiver_id IN (
SELECT sender_id FROM `friends` WHERE receiver_id = $userid AND STATUS = 'accepted')
UNION
SELECT uid, username FROM `friends` JOIN `user` WHERE sender_id = uid AND STATUS =
'accepted' AND sender_id <> $userid
AND receiver_id IN
(SELECT receiver_id FROM `friends` WHERE sender_id = $userid AND STATUS = 'accepted'));
```

My Friend List
[Chenglong](#)
[Yi](#)
[He](#)

Other people you may want to know
[He](#)
[Yali](#)
[Chenglong](#)
[Back to my homepage](#)

Figure 5.1 People you might know

5.2 Posts you might like

When user search some his interested topic of posts, he might search them through our search function as we provided. In the result page of the searched out post, we do a furthermore search by fetching the tag of the post and output the posts with same tag or similar tags in the system. So that we could make sure more interesting posts will be reachable for the user by search once.

```
//Other post you might like
$tag_query = mysql_query("SELECT DISTINCT `post_id`, `uid`, `title`, `text`, `upload_time`, `privileges`, `location_id`, `type`, `url`, `likes` FROM tag NATURAL JOIN post WHERE tag IN (SELECT tag FROM `tag` WHERE `post_id` = $postid) AND post_id <> $postid");
while($row = mysql_fetch_array($tag_query)){
    if ($row['privileges'] == 'All') {
        $postuid = $row['uid'];
        $userinfo_query = mysql_query("SELECT username FROM user WHERE uid = $postuid");
        $userrow = mysql_fetch_array($userinfo_query);
        $postusername = $userrow[0];
        echo $postusername,'<br />';
        if ($row['type'] == 'Activity') {
            // $postid = $row['post_id'];
            echo $row['title'];
            $title = $row['title'];
            echo $row['text'];
            $likes = $row['likes'];
            echo "<br />";
            $locationid = $row['location_id'];
            if ($check_location = mysql_query("SELECT * FROM location WHERE location_id = $locationid")) {
                $row = mysql_fetch_array($check_location);
                echo 'location: ', $row['location_name'], ' lat: ', $row['latitude'], ' lng: ', $row['longitude'], '<br />';
            }
            if ($check_tag = mysql_query("SELECT * FROM tag WHERE post_id = $postid")){
                if ($row = mysql_fetch_array($check_tag)) {
                    echo 'This post is about: ', $row['tag'];
                    while ($row = mysql_fetch_array($check_tag)) {
                        echo ', ', $row['tag'];
                    }
                }
            }
            echo "<br />";?>
```

Sjie
Title: Campus
NYU
2016-05-08 20:26:42

location: NYU lat: 31 lng: -106
This post is about: NYU



0

Other post you may interested in:

He
Title: NYU life
2016-05-08 20:30:23

This post is about: NYU



0

Figure 5.2 Recommend similar post with search result

5.3 Connection with Google maps

We connect our system with Google Map API, we could request the latitude and longitude coordinate in the real world, meanwhile we could let the location we searched show in the Google Map. It's convenient to use, help user find the place they want efficiently.

With the help of Google map, we could customize a place belong to ourselves, it is a real existed place with a customized name, it will give users a new taste of fun when they are using our system.

