



Build and design an event calendar website

Master of Science Thesis

Master Degree Program, Dependable Computer System

FENG SI

Department of Computer Science and Engineering
Division of Distributed Computing and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2009

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

To build and design an event calendar website

Si Feng

Si Feng, March 2009.

Examiner: Philippas Tsigas
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover: How the basic environment, WAMP, works in a computer.

Department of Computer Science and Engineering
Göteborg, Sweden September 2008

Acknowledgment

I would like to thank everyone who supported me in my thesis work especially my supervisor Philippas Tsigas.

Abstract

In this master thesis work an event website was designed to add and search events. A user can register as a member and subscribe any type of events. When an new event is added, the system would automatically send members who subscribe this kind of event an E-mail with an attachment of iCal event file. When users search an event, he/she could find the location of the event on a google map and also find events of the same day and nearby events.

Keywords: add events, search events, google map, iCal file, send E-mail

Contents

1 Objective.....	1
2 Background	1
2.1 PHP.....	1
2.2 MySQL	1
2.3 PHP+MySQL.....	2
2.4 Apache HTTP Server.....	2
2.5 WAMP.....	2
2.6 Adobe Dreamweaver	3
3 Build a website “eventsearch”	3
3.1 Install WAMP	3
4 User management.....	4
4.1 Create a database	4
4.2 Create a user table.....	5
4.3 User registration.....	6
4.4 User Login	8
4.5 User Logout	10
5 Add events	10
5.1 Create tables in the database.....	11
5.1.1 Concert table	11
5.1.2 Musical, theatre and opera table.....	11
5.1.3 Conference table	11
5.1.4 Other table.....	12
5.2 Add events	13

6 Search events	15
6.1 General search.....	15
6.1.2 Circulation search	15
6.1.2 Show results by page.....	18
6.2 Keyword Search.....	19
7 Search results	20
7.1 Websites of events	20
7.2 Event details.....	21
7.3 Modify events	22
7.4 Find nearby events	22
7.5 Find same date events	23
7.6 Location of events.....	23
7.6.1 Geocode address	23
7.6.2 Show location on Google map	25
7.7 Create iCal file	28
7.7.1 iCalCreator.....	28
7.7.2 Download iCal files	29
8 Send E-mail	30
8.1 Class phpmailer for sending E-mail.....	30
8.2 Class SMTP for communicating with an SMTP server	32
8.3 Send E-mail.....	35
9 Conclusions.....	35
10 References.....	36
11 Appendix.....	36
11.1 Register as a member	36
11.2 User log in and out	39

11.3 Add events.....	40
11.4 Search events	48
11.5 Event details.....	59
11.6 Modify events	62
11.7 Find nearby events	67
11.8 Same day events.....	71
11.9 Location of events on Google map	74
11.10 Download iCal file.....	75
11.11 Send E-mail.....	77

1 Objective

The objective of this master's thesis project is to develop an event website to add and search events. This project's work involves how to build a dynamic website by PHP+MySQL+Apache, use SQL insert and search statements, send E-mail by free smtp service, connect PHP+MySQL with google map and create iCal files.

2 Background

2.1 PHP

PHP was originally created by Rasmus Lerdorf in 1995, 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. Released under the PHP License, the Free Software Foundation considers it to be free software.

PHP is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input and creating web pages as output. It can also be used for command-line scripting and client-side GUI applications. It can be deployed on most web servers and on almost every operating system and can be used with many relational database management systems [1]. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

2.2 MySQL

The MySQL software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software [2].

MySQL is popular for web applications and acts as the database component of the LAMP, BAMP, MAMP, and WAMP platforms. Its popularity for use with web applications is closely tied to the popularity of PHP and Ruby on Rails, which are often combined with MySQL. MySQL works on many different system platforms.

To administer MySQL databases one can use the included command-line tool (commands: mysql and mysqladmin). Also downloadable from the MySQL site are GUI administration tools: MySQL Administrator and MySQL Query Browser. And

there are several other commercial and non-commercial tools available. Examples include phpMyAdmin, a free Web-based administration interface implemented in PHP, Navicat Lite Edition, a free desktop based GUI tool.

2.3 PHP+MySQL

PHP and MySQL when used together become powerful tools that are used to create dynamic and interactive websites. Both are opensource tools and can be easily acquired for free.

2.4 Apache HTTP Server

The Apache HTTP Server is a web server notable for playing a key role in the initial growth of the World Wide Web. Since April 1996 Apache has been the most popular HTTP server on the World Wide Web. As of December 2008 Apache served 51% of all websites [3].

Apache is primarily used to serve both static content and dynamic Web pages on the World Wide Web. Many web applications are designed expecting the environment and features that Apache provides.

Apache is the web server component of the popular LAMP web server application stack, alongside MySQL, and the PHP/Perl/Python (and now also Ruby) programming languages.

Apache is used for many other tasks where content needs to be made available in a secure and reliable way. Programmers developing web applications often use a locally installed version of Apache in order to preview and test code as it is being developed.

2.5 WAMP

WAMPs are packages of independently-created programs installed on computers that use a Microsoft Windows operating system. The interaction of these programs enables dynamic web pages to be served over a computer network, such as the internet or a private network.

"WAMP" is an acronym formed from the initials of the operating system (Windows) and the package's principal components: Apache, MySQL and PHP (or Perl or Python). Apache is a web server, which allows people with web browsers like Internet Explorer or Firefox to connect to a computer and see information there as web pages. MySQL is a database manager, that is, it keeps track of data in a highly organized way. PHP is a scripting language which can manipulate information held in a database

and generate web pages afresh each time an element of content is requested from a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

Generally, only a user with administrator privileges can install a WAMP package. This means that these packages cannot be installed to a hosted service but only to a computer to which the user has complete access.

2.6 Adobe Dreamweaver

Adobe Dreamweaver is a web development application originally created by Allaire Systems, which was acquired approximately 1998 by Macromedia and is now owned by Adobe Systems, which acquired Macromedia in 2005.

Dreamweaver is available for both Mac and Windows operating systems. Recent versions have incorporated support for web technologies such as CSS, JavaScript, and various server-side scripting languages and frameworks including ASP.NET, ColdFusion, JavaServer Pages, and PHP.

Dreamweaver can hide the HTML code details of pages from the user, making it possible for non-coders to create web pages and sites. Dreamweaver allows users to preview websites in many browsers, provided that they are installed on their computer. It also has some site management tools, such as the ability to find and replace lines of text or code by whatever parameters specified across the entire site, and a templatisation feature for creating multiple pages with similar structures. The behaviours panel also enables use of basic JavaScript without any coding knowledge.

3 Build a website “eventsearch”

3.1 Install WAMP

First,download the latest version of WAMP at <http://www.wampserver.com/>. Install it at D:\\wamp.When finishing the installation, start it. And it is shown at the right bottom of the screen like this:



Figure 1: *Installed WAMP*

We can see the environment Apache+ PHP+ MySQL is successfully equipped in my computer. Specific version for each is: Apache 2.2.8 + PHP 5.2.5 + MySQL 5.0.51a.

Create a new folder eventsearch under the directory D:\\wamp\\www and create and save html or php files in the folder. Then run <http://localhost/eventsearch/>, we can see webpages as we design.

4 User management

A user can register to be a member by inputting his/her information like username, password and E-mail address. There are several types of events, like concert, musical, opera, theatre, and other. The user can subscribe his/her interesting types. Then a member log in if he/she wants to add an event in the database.

4.1 Create a database

Create a new database “eventsearch”, we can save users’ information and event data in it.

A mysql command to create a new database:
mysql> CREATE DATABASE eventsearch;

Creating a database does not select it for use; we must do that explicitly. To make eventsearch the current database, use this command [4]:

```
mysql> USE eventsearch;
```

Database changed

Our database needs to be created only once, but we must select it for use each time we begin a mysql session.

4.2 Create a user table

We have created a database eventsearch, but at this point it's empty. We can create a table "user" which stores users' information in the database. We need to record user's username, password and E-mail address. Each user has a unique username. And we need to know which type the user subscribe.

Field Name	Value Type	Comments
username	VARCHAR(12)	Intitiled by users,different from others'
password	VARCHAR(12)	Typed by users
email	VARCHAR(40)	Users' E-mail address
concert	VARCHAR(10)	If a user subscribe it, save as "concert", otherwise save null
musical	VARCHAR(10)	Similar as concert
opera	VARCHAR(10)	Similar as concert
theatre	VARCHAR(10)	Similar as concert
conference	VARCHAR(10)	Similar as concert
other	VARCHAR(10)	Similar as concert

Table 1: *User table*

Create table user:

```
mysql> CREATE TABLE user (username VARCHAR(12),  
->password VARCHAR(12),  
->email VARCHAR(40),  
->concert VARCHAR(10),  
->musical VARCHAR(10),  
->opera VARCHAR(10),  
->theatre VARCHAR(10),  
->conference VARCHAR(10),  
->other VARCHAR(10));
```

VARCHAR is a good choice for the username, password and email columns because the column values vary in length.

To verify that our table was created the way we expected, use a DESCRIBE

statement:

```
mysql> DESCRIBE user;
```

Field	Type	Null	Key	Default	Extra
username	varchar(12)	YES		NULL	
password	varchar(12)	YES		NULL	
email	varchar(40)	YES		NULL	
concert	varchar(10)	YES		NULL	
musical	varchar(10)	YES		NULL	
opera	varchar(10)	YES		NULL	
theatre	varchar(10)	YES		NULL	
conference	varchar(10)	YES		NULL	
other	varchar(10)	YES		NULL	

Table 2: *User table*

4.3 User registration

Create a file register.html by dreamweaver. A user inputs his/her username, password, repassword, e-mail address and subscribing information. Make a html form `<form id="form1" name="form1" method="post" action="register.php"></form>`. It is used to transfer parameters. When the user presses submit, the values he/she fills are transferred as variables to another file register.php.

Create check boxes to show which type a user chooses to subscribe.

concert

musical

opera

theatre

conference

other

Figure 2: *Subscribe events*

Create a file register.php. It receives variables \$username, \$password, \$repassword

and \$email and subscribing values such as \$concert from register.html by function \$_POST['username'], \$_POST['password'], \$_POST['repassword'], \$_POST['email'] and \$_POST['concert'].

First check \$username and make sure it's not empty.

```
$username=$_POST['username'];
if($username=="")
{
    echo 'username cant be empty,please back <a
href="http://localhost/eventsearch/register.html">register</a>';
    exit;
}
```

If empty, back to the register webpage and the user input again. Use the same way to check password, repassword and e-mail address.

It also checks subscribing values such as \$concert, if it's not empty, evaluate "concert" to \$concert, else keep it null.

Then make sure that users type their right e-mail addresses.

```
if (!preg_match("/^([a-zA-Z0-9])+@([a-zA-Z0-9_-])+\.([a-zA-Z0-9_-])+/",$_POST['email']))
{
    echo 'Invalid E-mail address,please back <a href=
"http://localhost/eventsearch/register.html">register</a>';
    exit;
}
```

The e-mail address should match the usual e-mail address pattern ([a-zA-Z0-9])+@([a-zA-Z0-9_-])+([a-zA-Z0-9_-]), otherwise back to the register webpage and the user try again.

If the variables are all right, connect the database eventsearch:

```
$link=mysql_connect("localhost","root");
mysql_select_db("eventsearch");
```

"mysql_connect" is a function to open a connection to a MySQL server. Syntax is mysql_connect(string [hostname] [:port], string [username], string [password]) [5]. In this case, hostname and username should be the default ones localhost and root .And password can be ignored since I didn't set one.

“mysql_select_db” is used to select a MySQL database. In this case, it’s certainly eventsearch.

We have built a connection with the database eventsearch. Since every username should be different, we search current usernames of members from user table to check if there is already a existing username as \$username.

```
$query="select username from user where username='$username' ";  
$result=mysql_query($query);  
if($answer=mysql_fetch_array($result)) {  
echo 'there is a same username, please back <a  
href="http://localhost/eventsearch/register.html">register</a>';  
mysql_close($link);  
}
```

The select statement is used to select data from a database. Syntax is SELECT column_name(s) FROM table_name WHERE conditions. Here we search username from table user where username equals to \$username.

To get PHP to execute the select statement above we must use the mysql_query() function. This function is used to send a query or command to a MySQL connection.

“mysql_fetch_array” is to fetch a result row as an associative array, a numeric array, or both. It returns an array of strings that corresponds to the fetched row, or FALSE if there are no more rows. If a same username exists, back to register webpage and use mysql_close closes the connection to the MySQL server.

On the other hand, if no same username exists, stores \$username, \$password and \$email and subscribing values into user table. The user registers successfully.

```
$query="insert into user (username,password,email,concert,musical,opera,theatre,  
conference,other)  
values  
(' $username', '$password', '$email', '$concert', '$musical', '$opera', '$theatre', '$conferenc  
e', '$other')";  
mysql_query($query);
```

The insert statement is INSERT INTO table_name (field1, field2,...fieldN) VALUES (value1, value2,...valueN). To insert string data types it is required to keep all the values into double or single quote, for example:"\$username".

4.4 User Login

If a user has registered and his/her information has stored in the database, he/her

needs to log in now. How we define a member log in or not? We can use a cookie.

Cookies are a small computer-generated text file (no larger than 4K). It can record some user information. And a cookie may have a period of validity, so we can set a username as a cookie, if the program finds there is a username cookie, we assume the user has log in.

PHP offers a function `setcookie()` to set a cookie. We must send a cookie before any other head information and it's a restriction of a cookie. When one file sets a cookie, it's not invisible, unless another file use it. When programming, we should notice the two things. Once a cookie is set up, we can use it as a PHP variable [6].

Create `login.html`. Members type their usernames and passwords and press log in. Their input parameters are transferred by a html form to another file `login.php`.

User Login

username

password

Figure 3: `login.html`

The file `login.php` receives variables `$username` and `$password`.

If `$username` or `$password` is empty, back to `login.html` and the user types again.

```
if($username==""){$password=""})
{
    echo 'wrong information,please back
        <a href="http://localhost/eventsearch/login.html">login</a>';
    exit;
}
```

Otherwise connect the database.


```
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch");
```

To make sure the user inputs right information, search in the database to see if there is an existing \$username, if does, compare \$password the user inputs with the one saved in the database, if matches, the user is a registered member and login successfully. Set a cookie by the member's username and the period of validity is 1 hour.

```
$query="select password from user where username='$username'";
$result=mysql_query($query);
if($answer=mysql_fetch_array($result));
{
  if($password==$answer["password"])
  {
    setcookie("username",$username,time()+3600);
    mysql_close($link);
    header("Location:add.php");
    exit;
  }
}
```

Otherwise, input is wrong, back to login.html and try again.

```
mysql_close($link);
echo 'not exist user or wrong password,please try again
<a href="http://localhost/eventsearch/login.html">login</a>
,or
<a href="http://localhost/eventsearch/register.html">register</a> first';
```

4.5 User Logout

We just need to set a cookie, making the username a empty string. So the username cookie generated when log in is cleaned and non-existent.

```
setcookie("username", "", time() - 3600);
```

5 Add events

Once a member log in, he/she can add an event in the database. Events are divided into several types, including concert, musical, opera, theatre, conference and other. All types of events have several same attributes, like displayname, fullname and topkeywords. Meanwhile, each type has some specific attributes.

5.1 Create tables in the database

Four types of events, concert, musical, opera, theatre, have the same attributes. So we build four tables with the same architecture in the database eventsearch.

5.1.1 Concert table

Field Name	Value Type	Comments
id	int	Index of concerts and auto_increment
code	Char(2)	One code represents one type
Displayname	Varchar(50)	A short name of the concert
Fullname	Varchar(100)	A whole name of the concert
Topkeywords	Varchar(100)	Any related keyword for the concert
Year	Char(4)	Year of the concert
location	Varchar(100)	Specific location of the concert
website	Varchar(100)	If there is a website for the concert
lat	Float(10,6)	Latitude of the location
lng	Float(10,6)	Longitude of the location
tickets	Varchar(3)	Tickets for the concert are available or not
startdate	Datetime	Beginning date of the concert
enddate	Datetime	End date of the concert

Table 3: *Basic architecture of table concert*

SQL commands to create a table concert, id is the primary key of the table:

```
mysql> CREATE TABLE concert (id int unsigned not null auto_increment primary key, code char(2), Displayname varchar(50), Fullname varchar(100), Topkeywords varchar(100), Year char(4), location varchar(100), website varchar(100), lat float(10,6), lng float(10,6), tickets varchar(3), startdate datetime, enddate datetime);
```

5.1.2 Musical, theatre and opera table

Tables musical, theatre and opera have similar architecture as table concert.

Create three tables musical, theatre and opera similar as table concert. Id is the primary key of the table:

5.1.3 Conference table

Conference has some different attributes from above types.

Field Name	Value Type	Comments
------------	------------	----------

id	int	Index of conference and auto_increment
code	Char(2)	One code represents one type
Displayname	Varchar(50)	A short name of the conference
Fullname	Varchar(100)	A whole name of the conference
Topkeywords	Varchar(100)	Any related keyword for the conference
Year	Char(4)	Year of the conference
location	Varchar(100)	Specific location of the conference
website	Varchar(100)	If there is a website for the conference
lat	Float(10,6)	Latitude of the location
lng	Float(10,6)	Longitude of the location
Registration	Date	Registration day of the conference
Submission	Date	Submission day of the conference
Notification	Date	Notification day of the conference
FinalVersion	Date	The date of confirming final version
startdate	Datetime	Beginning date of the conference
enddate	Datetime	End date of the conference

Table 4: *Basic architecture of table conference*

SQL commands to create a table conference, id is the primary key of the table:

```
mysql> CREATE TABLE conference (id int unsigned not null auto_increment
primary key, code char(2), Displayname varchar(50), Fullname
varchar(100),Topkeywords varchar(100), Year char(4), location varchar(100),website
varchar(100), lat float(10,6), lng float(10,6), Registration date, Submission date,
Notification date, FinalVersion date, startdate datetime, enddate datetime);
```

5.1.4 Other table

The type other include events that don't fit any type above.

Field Name	Value Type	Comments
id	int	Index of other and auto_increment
code	Char(2)	One code represents one type
Displayname	Varchar(50)	A short name of the event
Fullname	Varchar(100)	A whole name of the event
Topkeywords	Varchar(100)	Any related keyword for the event
Year	Char(4)	Year of the event
location	Varchar(100)	Specific location of the event
website	Varchar(100)	If there is a website for the event
lat	Float(10,6)	Latitude of the location
lng	Float(10,6)	Longitude of the location
dates	Varchar(300)	Any important date for the event
startdate	datetime	Start date of the event

enddate	datetime	End date of the event
---------	----------	-----------------------

Table 5: *Basic architecture of table other*

SQL commands to create a table other, id is the primary key of the table:

```
mysql> CREATE TABLE other (id int unsigned not null auto_increment primary key,
code char(2), Displayname varchar(50), Fullname varchar(100),Topkeywords
varchar(100), Year char(4), location varchar(100),website varchar(100), lat float(10,6),
lng float(10,6),dates varchar(300),startdate datetime, enddate datetime);
```

5.2 Add events

We can make a webpage as below. Members can fill in event information and submit. Then the data would be disposed and saved in the database.

Display name:

Full name:

Top Keywords:

Figure 4: *add.php*

Since we have several types, and each type has some special attributes, we can make a choice menu, like this:

Figure 5: *Choice list*

Members can choose the type he/she wants. Each type would lead to a targeted webpage and members can fill in corresponding event information. And we can use a url to transfer the parameter like `add.php?action=concert` then we can tell which type the member choose. All data is post to `addevent.php`.

```
<form action="addevent.php?type=<?php echo $_GET['action'] ?>"
```

```

method="post">
<select name="jumpMenu" id="jumpMenu"
onchange="MM_jumpMenu('parent',this,0)">
  <option value="">choose a type</option>
  <option value="add.php?action=concert">concert</option>
  <option value="add.php?action=musical">musical</option>
  <option value="add.php?action=opera">opera</option>
  <option value="add.php?action=theatre">theatre</option>
  <option value="addcon.php?action=conference">conference</option>
  <option value="addother.php?action=other">other</option>
</select>
</form>

```

The file addevent.php receives data including the type of the event and event information. It checks if a member has login first by a username cookie. If not, back to login.html.

```

if(!$_COOKIE["username"])
{
  echo 'please back <a href="login.html">login</a>';
}

```

If a member has login, it collects all data and inserts into the database. Data with different type is saved in different table. We use function switch to make a judge.

```

switch ($_GET['type'])
{
  case "concert":
    $member_addc="insert into concert
(code,Displayname,Fullname,Topkeywords,Year,location,website,lat,lng,tickets,startdate,enddate) values
('e1','$displayname','$fullname','$stopkeywords','$year','$location','$website','$lat','$lng','$tickets','$startdate','$enddate')";
    $member_add_savec=mysql_query($member_addc,$link) or die
(mysql_error());
    break;}

```

Concert is assigned to the code e1. And musical, opera, theatre, conference and other are assigned to e2,e3,e4,e5,e6 separately. Lat and lng for latitude and longitude of the location are calculated by google maps before inserting into the database. We will talk about it later.

6 Search events

The search webpage lists any type of events that best match a given query. The following two query types are available:

General Search

Keyword Search

6.1 General search

6.1.2 Circulation search

A query can consist of single keywords, important dates or any information of the events.

A user can input one or several keywords for a query, and separate them by a space. So we have to check what a user inputs and split it if it isn't a single word.

```
$search=$_POST['search'];  
$search=rtrim(ltrim($search));  
$keyword=split(" ",$search);  
$max=count($keyword);
```

The function “ltrim” is to throw the space at the left of the string, while “rtrim” is to throw the right one. “split” is to split keywords by a space. Then count the number of the keywords.

If there is only one keyword, search the database in a normal way. We have 6 tables, concert, musical, theatre, opera, conference, other, saving event information and they have different architecture, so we use different query statements. We search most columns of the tables, except the unrelated ones like code or id.

```
static $query1="";  
static $query2="";  
static $query3="";  
if ($max==1){  
$i=0;  
$query1=$query1."Displayname like '%$keyword[$i]%' or Fullname like  
 '%$keyword[$i]%' or Topkeywords like '%$keyword[$i]%' or Year like  
 '%$keyword[$i]%' or location like '%$keyword[$i]%' or website like  
 '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or enddate like  
 '%$keyword[$i]%'";  
//for table concert, musical, theatre and opera
```

```

$query2=$query2."Displayname like '%$keyword[$i]%' or Fullname like
 '%$keyword[$i]%' or Topkeywords like '%$keyword[$i]%' or Year like
 '%$keyword[$i]%' or location like '%$keyword[$i]%' or website like
 '%$keyword[$i]%' or Registration like '%$keyword[$i]%' or Submission like
 '%$keyword[$i]%' or Notification like '%$keyword[$i]%' or FinalVersion like
 '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or enddate like
 '%$keyword[$i]%'";
//for table conference
$query3=$query3."Displayname like '%$keyword[$i]%' or Fullname like
 '%$keyword[$i]%' or Topkeywords like '%$keyword[$i]%' or Year like
 '%$keyword[$i]%' or location like '%$keyword[$i]%' or website like
 '%$keyword[$i]%' or dates like '%$keyword[$i]%'";
//for table other
}

```

The LIKE condition allows us to use wildcards in the where clause of an SQL statement. This allows us to perform pattern matching. The LIKE condition can be used in any valid SQL statement - select, insert, update, or delete. % allows us to match any string of any length (including zero length). Here, we can find any information which contains the characters of the keyword.

If there are several keywords, we search each keyword one by one through a circulation.

It's the start of a sql statement and search with the first keyword. Beginning with "(" and end with "or".

```

else{
for ($i = 0; $i < $max; $i++){
if ($i==0){
$query1=$query1 ."(Displayname like '%$keyword[$i]%' or Fullname like
 '%$keyword[$i]%' or Topkeywords like '%$keyword[$i]%' or Year like
 '%$keyword[$i]%' or location like '%$keyword[$i]%' or website like
 '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or enddate like
 '%$keyword[$i]%' or ";
$query2=""; // search every field name in table conference, similar above, ignore
here
$query3=""; // search every field name in table other, similar above, ignore here
}
}

```

It's the end of the sql statement, when the circulation has processed with the last keyword.

```

else{

```

```

if ($i==($max-1)){
$query1=$query1 ."Displayname like '%$keyword[$i]%' or Fullname like
 '%$keyword[$i]%' or Topkeywords like '%$keyword[$i]%'or Year like
 '%$keyword[$i]%' or location like '%$keyword[$i]%' or website like
 '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or enddate like
 '%$keyword[$i]%'";
$query2=""; // search every field name in table conference, similar above, ignore
here
$query3=""; // search every field name in table other, similar above, ignore here
}

```

This is the middle part of the sql statement.

```

else{
$query1=$query1 ."Displayname like '%$keyword[$i]%' or Fullname like
 '%$keyword[$i]%' or Topkeywords like '%$keyword[$i]%'or Year like
 '%$keyword[$i]%' or location like '%$keyword[$i]%' or website like
 '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or enddate like
 '%$keyword[$i]%' or ";
$query2=""; // search every field name in table conference, similar above, ignore
here
$query3=""; // search every field name in table other, similar above, ignore here
}

```

Now it would return values where matching the query in the database. To collect all data from 6 different tables, we use “union all” between select statement, like:

```

select field from table1 where condition
union all
select field from table2 where condition;

```

The above code is for tables with same field names, since we want to get values from 6 tables with different field names, we need some change:

```

Select 'field1' as field, 'tbl1' as tb from taA where condition
Union all
Select 'field2' as field, 'tbl2' as tb from taB where condition;

```

According to the method, we write concrete codes like:

```

$query="select id, code, Displayname, Fullname, Topkeywords, Year, location,
website, " as Registration, " as Submission, " as Notification, " as FinalVersion,
startdate, enddate, " as dates, 'tbl1' as tbl from concert where ".$query1
."union all
//select from table musical, theatre, opera, similar as concert, ignore here

```



```

"union all
select id, code, Displayname, Fullname, Topkeywords, Year, location, website,
Registration, Submission, Notification, FinalVersion, startdate, enddate, " as dates,
'tbl5' as tbl from conference where " .$query2
"union all
select id, code, Displayname, Fullname, Topkeywords, Year, location, website, " as
Registration," as Submission, " as Notification, " as FinalVersion, startdate, enddate,
dates, 'tbl6' as tbl from other where " .$query3
"limit $offset, $pagesize";
$result=mysql_query($query) or die("query failed:".mysql_error());

```

6.1.2 Show results by page

After we get the results, we want to order them and make them shown by pages. Here's are two important parameters:

\$pagesize: how many records are shown on one page;

\$page: the current page number;

Notice the above sql statement, ended with "limit \$offset, \$pagesize", \$offset is to record offset by calculation of $offset = pagesize * (page - 1)$.

First, calculate how many records fitting the query. Add numbers from 6 tables.

```

$rowc=mysql_query("select count(*) from concert where " .$query1);
$rowm=mysql_query("select count(*) from musical where " .$query1);
$rowt=mysql_query("select count(*) from theatre where " .$query1);
$rowo=mysql_query("select count(*) from opera where " .$query1);
$rowcon=mysql_query("select count(*) from conference where " .$query2);
$rowoth=mysql_query("select count(*) from other where " .$query3);
$myrowc=mysql_fetch_array($rowc);
$myrowm=mysql_fetch_array($rowm);
$myrowt=mysql_fetch_array($rowt);
$myrowo=mysql_fetch_array($rowo);
$myrowcon=mysql_fetch_array($rowcon);
$myrowoth=mysql_fetch_array($rowoth);
$numrows=$myrowc[0]+$myrowm[0]+$myrowt[0]+$myrowo[0]+$myrowcon[0]+$
myrowoth[0];

```

How to set the page number:

```

$pagesize=10;
$pages=intval($numrows/$pagesize);
if ($numrows%$pagesize)
$pages++;
if (isset($_GET['page'])) {

```


and the regular expression engine takes the grammar and compares it with the string. The engine returns an indication of whether the string matches the grammar; that is, if the string contains substrings that can be derived from the grammar. And the marks `[:<:]`, `[:>:]` represents the boundary of a word. They match the beginning and the end of a word separately. A word here means a string of characters, no characters before and after it.

For example, we search a keyword in table concert by keyword search:

```
$query="select * from concert where
    Displayname regexp '[:<:]*$keyword[$i][:>:]' or
    Fullname regexp '[:<:]*$keyword[$i][:>:]' or
    Topkeywords regexp '[:<:]*$keyword[$i][:>:]' or
    Year regexp '[:<:]*$keyword[$i][:>:]' or
    location regexp '[:<:]*$keyword[$i][:>:]' or
    website regexp '[:<:]*$keyword[$i][:>:]' or
    startdate regexp '[:<:]*$keyword[$i][:>:]' or
    enddate regexp '[:<:]*$keyword[$i][:>:]' ";
```

By this method, we accomplish keyword search and show results by pages.

7 Search results

7.1 Websites of events

Assume we input “gothenborg” and press search, it will give us any information about gothenborg and show the results like this:

WWW	Name	Location	Modify	Nearby	SameDateEvents	iCalFile
WWW	cth	gibraltargaten, gothenborg, sweden	Modify	nearby	date	Download
WWW	S	avenue, gothenborg, sweden	Modify	nearby	date	Download
WWW	a	gothenborg, sweden	Modify	nearby	date	Download
WWW	try	avenue, gothenborg, sweden	Modify	nearby	date	Download
WWW	office	gothenborg	Modify	nearby	date	Download

1 pages(1/1)[1]

Figure 6: Search results

When we click [www](#), it would go to the website of the event directly by retrieving the website of the search answer.

```
$url=&$answer["website"];
<a href="http://<?php echo $url?>">WWW</a>
```

7.2 Event details

In the name row, it gives the displayname of an event. If we click it, it will show us every detail information of the event like.

```
Display name: cth
Full name:    chalmers university
Top Keyword:  education
```

Figure 7: Event details

Because each event has a unique id in one table of the database, we can use id and code to identify an event. Thus it transfers id and code of the search result by url to the eventdetail file.

```
<a href="eventdetails.php?id=<?php echo $answer["id"]?>&code=<?php echo $answer["code"]?>"><?php echo $answer["Displayname"];?></a>
```

The file eventdetails.php would get the id and code. If the code is “e5” or “e6”, that is the type of conference or other, the parameters would be transferred to other files again by a header function.

```
if($code==e5){
header("Location:condetails.php?id=$id&code=$code");}
```

Otherwise, we search the answer by id and code from 4 tables, concert, musical, theatre, opera.

```
$query="select * from concert where id='$id' and code='$code'
union all
select * from musical where id='$id' and code='$code'
union all
select * from theatre where id='$id' and code='$code'
union all
select * from opera where id='$id' and code='$code'";
$result=mysql_query($query) or die("query failed:".mysql_error());
$answer=mysql_fetch_array($result);
```

We get the result and display every detail by a function echo. For example, top keyword is shown by <?php echo \$answer["Topkeywords"];?>.

Similarly, the file condetails.php or otherdetails.php would do the same search and

show the results when it gets id and code.

7.3 Modify events

Modify also shows everything of the result, further more, users can edit the information and update.



Display name:

Full name:

Top Keywords:

Figure 8: *Modify events*

Similarly, the file eventmodify.php also get id and code. If a user change information of an event, it judges which type it is, and update context in the corresponding table.

```
if ($code=='e1'){  
$query="update concert set Displayname='$displayname', Fullname='$fullname',  
Topkeywords='$stopkeywords', Year='$year', location='$location', website='$website',  
lat='$lat', lng='$lng', tickets='$tickets', startdate='$start', enddate='$end'  
where id=$id";  
mysql_query($query) or die("query failed:".mysql_error());  
}
```

7.4 Find nearby events

Nearby checks that if there is any event around the result and gives several choices for users, like 100km, 200km, 300km, 400km, 500km away from the result event. Google maps group offers a function to find nearby events [7].

```
$query = "select id,code,location,Displayname,lat,lng,( 6371 *  
acos( cos( radians('$lat') ) * cos( radians( lat ) ) * cos( radians( lng ) -  
radians('$lng') ) + sin( radians('$lat') ) * sin( radians( lat ) ) ) ) AS distance FROM  
concert HAVING distance < 100
```

Here's the SQL statement that will find the closest locations that are within a radius of 100 kilometers to the \$lat, \$lng (latitude and longitude of the result event) coordinate. It calculates the distance based on the latitude/longitude of that row and the target latitude/longitude, and then asks for only rows where the distance value is less than 100km. 6371 is a code for kilometers. Then union all search results from 6 event

tables, we would get all events around 100km or 200km to 500km away.

7.5 Find same date events

SameDateEvents would find any event with the same start date or end date of the result, but different times. Since startdate and enddate are saved as type datetime, first we retrieve date by getting first 10 date characters from the datetime string.

```
$start=substr($startdate,0,10);  
$end=substr($enddate,0,10);
```

Then complete a simple SQL search in 6 event tables:

```
$query = "select id,code,Displayname,startdate,enddate from concert where startdate  
like '$start%' or enddate like '$end%'  
union all  
//similar search in other 5 tables, ignore here;
```

7.6 Location of events

When users click “location”, the location of the result and nearby events(100km around) would be shown on a google map.

7.6.1 Geocode address

First we need geocode addresses of events. 'Geocoding' is the process of converting an address into a set of latitude/longitude coordinates, making it possible to indicate addresses on a map.

When a user adds an event, we geocode the address and save latitude and longitude with other information the user fills in an event table of the database. If the user modify the location later, we geocode the new address and update in the database.

The HTTP geocoder can respond with either XML or CSV output through changing the "output" query parameter. We can retrieve CSV output and process those results. Use the PHP function `file_get_contents` to retrieve the CSV response, and then split to separate out the coordinate values and status code of the returned CSV [8].

We have specified geocoder output, our requests will be subject to the geocoder's maximum query rate and 15,000 queries per day based on my IP. Additionally, a status code of 620 will be returned by the geocoder if we query it faster than it can handle. To ensure we don't send queries too rapidly to the geocoder, we can specify a delay between each geocode request. We can increase this delay each time we receive

a 620 status, and use a while loop to ensure we've successfully geocoded an address before iterating to the next one.

Sign up for the Google maps API at <http://code.google.com/apis/maps/signup.html> and get a map API key. Each website url needs a unique key. The Google Maps API lets us geocode addresses and embed Google Maps in our own web pages.

Once we send off requests to the geocoder and receive responses back, we save them in the database.

```
define("MAPS_HOST", "maps.google.com");
define("KEY",
"ABQIAAAAAn_yWEGNyLH_GB43JkuJxhTE5397KEOkSQkq_59pmCk-UQrEkRST
Hh2CrrMAPbD8ahwaXhHj9rp-BQ");
// Initialize delay in geocode speed
$delay = 0;
$base_url = "http://" . MAPS_HOST . "/maps/geo?output=csv&key=" . KEY;
// Iterate through the rows, geocoding each address
if ($location=$_POST['location']) {
    $geocode_pending = true;
    if ($geocode_pending) {
        $location=$_POST['location'];
        $request_url = $base_url . "&q=" . urlencode($location);
        $csv = file_get_contents($request_url) or die("url not loading");
        $csvSplit = split(",", $csv);
        $status = $csvSplit[0];
        $lat = $csvSplit[2];
        $lng = $csvSplit[3];
        if (strcmp($status, "200") == 0) {
            // successful geocode
            $geocode_pending = false;
            $lat = $csvSplit[2];
            $lng = $csvSplit[3];
        }
        else if (strcmp($status, "620") == 0) {
            // sent geocodes too fast
            $delay += 100000;
        } else {
            // failure to geocode
            $geocode_pending = false;
            echo "Address " . $address . " failed to geocoded. ";
            echo "Received status " . $status . "\n";
        }
    }
    usleep($delay);
}
```

```
}}
```

We have completed geocoding and stored latitude and longitude in the database. We now need to write some PHP statements to export the table data into an XML format that our map can retrieve through asynchronous JavaScript calls. We can simply output the XML with the echo function. When using just the echo function, we'll need to use a helper function (e.g. parseToXML) that will correctly encode a few special entities (<, >, ", ') to be XML friendly.

7.6.2 Show location on Google map

In the PHP, connect to the database and execute the SELECT * (select all) query on the event tables. First retrieve latitude and longitude of the result event. And find nearby (100km around) events by the function we mentioned in finding nearby events chapter. Then echo out the parent markers node, and iterate through the query results. For each row in the table (each location), we need to echo out the XML node for that marker, sending the name and address fields through the parseToXML function first in case there are any special entities in them [8]. Finish the script by echoing out the closing markers tag.

```
echo '<markers>';
// Iterate through the rows, printing XML nodes for each
while ($row = @mysql_fetch_assoc($result1)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ' .
    echo 'name="'. parseToXML($row['Displayname']). "' ";
    echo 'address="'. parseToXML($row['location']). "' ";
    echo 'lat="'. $row['lat']. "' ";
    echo 'lng="'. $row['lng']. "' ";
    echo 'type="'. $row['code']. "' ";
    echo 'id="'. $row['id']. "' ";
    echo '/>';
}
// End XML file
echo '</markers>';
```

Input keyword “gothenborg” and call this PHP script from the browser to make sure it's producing valid XML.

```
- <markers>
  <marker name="cth" address="gibraltargaten,gothenborg,sweden" lat="57.6874" lng="11.9824" type="e1" id="2" />
  <marker name="s" address="avenue,gothenborg,sweden" lat="57.7234" lng="11.846" type="e1" id="3" />
  <marker name="a" address="gothenborg,sweden" lat="57.697" lng="11.9864" type="e1" id="5" />
  <marker name="foll" address="avenue,gothenborg,sweden" lat="57.7234" lng="11.846" type="e4" id="1" />
</markers>
```

Figure 8: *Output XML*

Once the XML is working in the browser, it's time to move on to actually creating the map with JavaScript. To load the XML file into our page, we can take advantage of the API function `GDownloadURL`. `GDownloadURL` is a wrapper for the `XMLHttpRequest` that's used to request an XML file from the server where the HTML page resides. The first parameter to `GDownloadURL` is the path to my file—it's usually easiest to have the XML file in the same directory as the HTML so that we can just refer to it by filename. The second parameter to `GDownloadURL` is the function that's called when the XML is returned to the JavaScript.

In the callback function, we need to find all the "marker" elements in the XML, and iterate through them. For each marker element we find, retrieve the name, address, type, and lat/lng attributes and pass them to `createMarker`, which returns a marker that we can add to the map.

```
function load() {
  if (GBrowserIsCompatible()) {
    var searchUrl = 'phpsqlajax_genxml2.php?id=' + id + '&code=' + code ;
    GDownloadUrl(searchUrl, function(data) {
      var xml = GXml.parse(data);
      var markers = xml.documentElement.getElementsByTagName("marker");
      var i=0;
      var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
        parseFloat(markers[i].getAttribute("lng")));
      var map = new GMap2(document.getElementById("map"));
      map.addControl(new GSmallMapControl());
      map.addControl(new GMapTypeControl());
      map.setCenter(point, 6);
      for (var i = 0; i < markers.length; i++) {
        var name = markers[i].getAttribute("name");
        var address = markers[i].getAttribute("address");
        var type = markers[i].getAttribute("type");
        var id = parseInt(markers[i].getAttribute("id"));
        var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
          parseFloat(markers[i].getAttribute("lng")));
        var marker = createMarker(point, name, address, type,id);
        map.addOverlay(marker);
      }
    });
  }
}
```

We can use the `GIcon` class to define custom icons which can later be assigned to the markers. Start by declaring a `GIcon` object—`iconRed`—and define their properties.

```
var iconRed = new GIcon();
iconRed.image = 'http://labs.google.com/ridefinder/images/mm_20_red.png';
iconRed.shadow = 'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
```

```

iconRed.iconSize = new GSize(12, 20);
iconRed.shadowSize = new GSize(22, 20);
iconRed.iconAnchor = new GPoint(6, 20);
iconRed.infoWindowAnchor = new GPoint(5, 1);
var customIcons = [];
customIcons["e1", "e2", "e3", "e4", "e5", "e6"] = iconRed;

```

We should have all our marker creation code in a createMarker function. We can retrieve the appropriate GIcon by using the type as the key for the associative array that was globally defined, and pass that into the GMarker constructor. Then, construct the HTML that we want to show up in the info window by concatenating the address. After constructing the HTML string, add an event listener to the marker so that when clicked, an info window is displayed.

```

function createMarker(point, name, address, type, id) {
  var marker = new GMarker(point, customIcons[type]);
  var html = address;
  GEvent.addListener(marker, 'click', function() {
    marker.openInfoWindowHtml(html);
  });
  return marker;
}

```

Finally tie the markers, icons, and XML together. When the page loads, the load function is called. This function sets up the map and then calls GDownloadUrl. Make sure our GDownloadUrl is passing in the file that outputs the XML and that we can preview that XML in the browser.



Figure 9: location on Google map

7.7 Create iCal file

7.7.1 iCalCreator

iCal is a personal calendar application made by Apple Inc, that runs on the Mac OS X operating system. iCal was the first calendar application for Mac OS X to offer support for multiple calendars and the ability to publish/subscribe calendars to WebDAV server. It tracks events and appointments, allows multiple calendar views (such as calendars for "home", "work", and other calendars you create) to quickly identify conflicts and free time.

iCalendar is a standard (RFC 2445) for calendar data exchange. The standard is sometimes referred to as "iCal", which also is the name of the Apple, Inc. calendar program that provides one of the implementations of the standard.

The top-level object in iCalendar is the Calendaring and Scheduling Core Object. This is a collection of calendaring and scheduling information. Typically, this information will consist of a single iCalendar object. However, multiple iCalendar objects can be sequentially grouped together. The first line must be "BEGIN:VCALENDAR", and the last line must be "END:VCALENDAR"; the contents between these lines is called the "icalbody". The body of the iCalendar object (the icalbody) consists of a sequence of calendar properties and one or more calendar components. The calendar properties are attributes that apply to the calendar as a whole. The calendar components are collections of properties that express a particular calendar semantic. For example, the calendar component can specify an event, a to-do, a journal entry, time zone information, or free/busy time information, or an alarm.

PHP offers a class: iCalCreator. It is a PHP implementation to create, parse and edit iCal (rfc2445) files for non-calendar systems like CMS, project management systems, and other applications able to support calendar information like agendas, tasks, reports, todo lists, journaling data and communicate with calendar applications. Download it from website as a file iCalcreator.class.php.

When a user add a event and click insert, the system would collect all the information and create a iCal file by using iCalCreator class.

First create a new calendar instance. Set some iCal properties, for instance, set unique_id as events, set method as Publish and set timezone as Europe/Stockholm.

```
require_once( 'iCalcreator.class.php' );  
$v = new vcalendar();  
$v->setConfig( 'unique_id', 'events' );  
$v->setProperty( 'method', 'PUBLISH' );
```

```

$V->setProperty( "x-wr-calname", " Calendar event" );
$V->setProperty( "X-WR-CALDESC", "Calendar Description" );
$V->setProperty( "X-WR-TIMEZONE", "Europe/Stockholm" );

```

Then create an event calendar component. Add event information to the iCal calendar. The title of the calendar is the displayname of the event which a user inputs, and description of the calendar file is topkeywords of the event, and location, startdate and enddate correspond with location, startdate and enddate of the event.

```

$vevent = new vevent();
$vevent->setProperty( 'title', "$displayname" );
$vevent->setProperty( 'description', "$topkeywords $dates" );
$vevent->setProperty( 'LOCATION', "$location" );
$vevent->setProperty( 'dtstart', "$startdate");
$vevent->setProperty( 'dtend', "$enddate");
$V->setComponent ( $vevent );

```

Finally we create a iCal file in a stated directory as a stated file name.

```

$V->setConfig( 'directory', 'd:/upload/' );
$V->setConfig( 'filename', "$type$displayname.ics" );
$V->saveCalendar();cs

```

7.7.2 Download iCal files

When a user searches an event and results come out, the user can download an iCal file of the event.

First check whether the target iCal file exists in the iCal file directory, if does, open the file by a function fopen.

```

$file_name = "$type$displayname.ics";
$file_dir = "d:/upload/";
if (!file_exists($file_dir . $file_name)) {
echo "Cann't find the file";
exit;
} else {
$file = fopen($file_dir . $file_name,"r");

```

Then use Header information to pass related file information like size of the file to apache server and IE browser.

```

Header("Content-type: application/octet-stream");
Header("Accept-Ranges: bytes");

```

```
Header("Accept-Length: ".filesize($file_dir . $file_name));
Header("Content-Disposition: attachment; filename=" . $file_name);
```

Finally output the iCal file to IE browser by the use of function fread. Browser would remind users to download the file.

```
echo fread($file,filesize($file_dir . $file_name));
fclose($file);
}
```

8 Send E-mail

When a user adds an event, the system would automatically send a email to users who subscribe the event with a attachment of iCal file.

First, we need an smtp server. Now some websites provide free smtp service. The Simple Mail Transfer Protocol (SMTP) service, which transfers e-mail, is installed as part of E-mail Services, along with the POP3 service. SMTP controls how e-mail is transported and then delivered across the Internet to the destination server. The SMTP service sends and receives e-mail between servers, whereas the POP3 service retrieves e-mail from the mail server to the user's computer.

I choose a Chinese website www.sina.com.cn and open an email account on it, then open the POP3 service of the email account. Now we can connect its smtp server smtp.sina.com.cn. And we use the username and password of the email account as smtp authentication.

8.1 Class phpmailer for sending E-mail

Create a class phpmailer for sending Email. Define necessary variables, like \$From is the From email address for the message, \$Sender is the sender email of the message, \$SMTPAuth is to set smtp authentication, and \$Username and \$Password are for setting SMTP username and password.

Define necessary methods. AddAddress is to add a "To" address.

```
function AddAddress($address, $name = "") {
    $cur = count($this->to);
    $this->to[$cur][0] = trim($address);
    $this->to[$cur][1] = $name;
}
```

Send is to create message and assigns Mailer. If the message is not successfully sent then it returns false. Use the `ErrorInfo` variable to view description of the error.

```
function Send() {
    $header = "";
    $body = "";
    if((count($this->to) + count($this->cc) + count($this->bcc)) < 1)
    {
        $this->error_handler("You must provide at least one recipient email address");
        return false;
    }
    // Set whether the message is multipart/alternative
    if(!empty($this->AltBody))
        $this->ContentType = "multipart/alternative";
    // Attach sender information & date
    $header = $this->received();
    $header .= sprintf("Date: %s%s", $this->rfc_date(), $this->LE);
    $header .= $this->create_header();
    if(!$body = $this->create_body())
        return false;
    //echo "<pre>".$header . $body . "</pre>"; // debugging
    // Choose the mailer
    if($this->Mailer == "sendmail")
    {
        if(!$this->sendmail_send($header, $body))
            return false;
    }
    elseif($this->Mailer == "mail")
    {
        if(!$this->mail_send($header, $body))
            return false;
    }
    elseif($this->Mailer == "smtp")
    {
        if(!$this->smtp_send($header, $body))
            return false;
    }
    else
    {
        $this->error_handler(sprintf("%s mailer is not supported", $this->Mailer));
        return false;
    }
}
```

```

        return true;
    }

```

And AddAttachment is to add a attachment from a path on the file system. Checks if attachment is valid and then adds the attachment to the list.

```

function AddAttachment($path, $name = "", $encoding = "base64", $type =
"application/octet-stream") {
    if(!@is_file($path))
    {
        $this->error_handler(sprintf("Could not access [%s] file", $path));
        return false;
    }
    $filename = basename($path);
    if($name == "")
        $name = $filename;
    // Append to $attachment array
    $cur = count($this->attachment);
    $this->attachment[$cur][0] = $path;
    $this->attachment[$cur][1] = $filename;
    $this->attachment[$cur][2] = $name;
    $this->attachment[$cur][3] = $encoding;
    $this->attachment[$cur][4] = $type;
    $this->attachment[$cur][5] = false; // isStringAttachment
    $this->attachment[$cur][6] = "attachment";
    $this->attachment[$cur][7] = 0;
    return true;
}

```

8.2 Class SMTP for communicating with an SMTP server

Create an SMTP class that can be used to connect and communicate with any SMTP server.

Define necessary methods. Function connect is to connect the server specified on the port specified. If the port is not specified use the default SMTP_PORT. If tval is specified then a connection will try and be established with the server for that number of seconds. If tval is not specified the default is 30 seconds to try on the connection. Tval is the time trying to connect the server.

```

function Connect($host,$port=0,$tval=30) {
    $this->error = null;
}

```

```

if($this->connected()) {
    $this->error =
        array("error" => "Already connected to a server");
    return false;
}
if(empty($port)) {
    $port = $this->SMTP_PORT;
}
#connect to the smtp server
$this->smtp_conn = fsockopen($host, # the host of the server
                            $port, # the port to use
                            $errno, # error number if any
                            $errstr, # error message if any
                            $tval); # give up after ? secs
# verify we connected properly
if(empty($this->smtp_conn)) {
    $this->error = array("error" => "Failed to connect to server",
                        "errno" => $errno,
                        "errstr" => $errstr);
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": $errstr ($errno) " . $this->CRLF;
    }
    return false;
}
# sometimes the SMTP server takes a little longer to respond
# so we will give it a longer timeout for the first read
if(substr(PHP_OS, 0, 3) != "WIN")
    socket_set_timeout($this->smtp_conn, 1, 0);
$announce = $this->get_lines();
if($this->do_debug >= 2) {
    echo "SMTP -> FROM SERVER:" . $this->CRLF . $announce;
}
return true;
}

```

Authenticate is to perform authentication by sending encoded username and password. Returns true if successfully authenticated.

```

function Authenticate($username, $password) {
    // Start authentication
    fputs($this->smtp_conn, "AUTH LOGIN" . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply, 0, 3);
}

```



```

if($code != 334) {
    $this->error =
        array("error" => "AUTH not accepted from server",
            "smtp_code" => $code,
            "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": " . $rply . $this->CRLF;
    }
    return false;
}
// Send encoded username
fputs($this->smtp_conn, base64_encode($username) . $this->CRLF);
$rply = $this->get_lines();
$code = substr($rply,0,3);
if($code != 334) {
    $this->error =
        array("error" => "Username not accepted from server",
            "smtp_code" => $code,
            "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": " . $rply . $this->CRLF;
    }
    return false;
}
// Send encoded password
fputs($this->smtp_conn, base64_encode($password) . $this->CRLF);
$rply = $this->get_lines();
$code = substr($rply,0,3);
if($code != 235) {
    $this->error =
        array("error" => "Password not accepted from server",
            "smtp_code" => $code,
            "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": " . $rply . $this->CRLF;
    }
    return false;
}
return true;
}

```

8.3 Send E-mail

Use the two defined classes to send Email by filling our email information.

```
require("class.phpmailer.php");
$mail = new PHPMailer(); //Build a new Mailer class
$address = $_POST['address'];
$mail->IsSMTP(); // Sending Email by smtp
$mail->Host = "smtp.sina.com"; // smtp server name
$mail->SMTPAuth = true; // Use smtp authentication
$mail->Username = "sif99@sina.com.cn"; // Username of Email account
$mail->Password = "*****"; // Password of Email account
$mail->From = "user@xxx.com"; //Sender Email address
$mail->FromName = ""; //Sender name
$mail->AddAddress("$address", ""); // Recipient Email address and name
$mail->AddAttachment("/var/tmp/file.tar.gz"); // Add a attachment
$mail->Subject = "$displayname"; //Subject of Email
$mail->Body="There is a new event,
http://localhost/eventsearch/eventdetails.php?id=$id&code=$code";
//Context of Email, the url of the event
if(!$mail->Send())
{
    echo "Failure send . <p>";
    exit;
}
echo "Send successfully";
```

9 Conclusions

The objective of this thesis work is to design a website to add and search events. It also connects with some popular web applications, like Google map and iCal file. The main methods I used to achieve this project are the following:

- Connect PHP with MySQL. Create a database and use SQL insert statement to store information;
- Use SQL select statement to achieve search event work in 7 event database tables; do a circulation search if there are multiple search keywords;
- Set a cookie to judge whether a member logs in or not;
- To show location on a Google map, geocode addresses first, then output latitude and longitude by XML;
- Use iCalCreator class to create an iCal file;
- To send E-mail, create two classes, one for sending issues, another for connect and authenticate with an SMTP server.

10 References

- [1] O'Reilly, "Embedding PHP in HTML", (2008).
- [2] MySQL AB, "What is MySQL?, MySQL 5.0 Reference Manual", (2007).
- [3] Netcraft, "December 2008 Web Server Survey", (2009).
- [4] MySQL AB, "create and choose a database, MySQL 5.0 Reference Manual", (2007).
- [5] PHP Manual, "mysql_connect", (2007).
- [6] Zhou Hao, "PHP, develop E-business", (2001).
- [7] Pamela Fox, "Creating a Store Locator with PHP, MySQL & Google Maps", (2008).
- [8] Pamela Fox, "Using PHP/MySQL with Google Maps", (2007).

11 Appendix

In this chapter, I'll show all needed php codes and files for this project work..

11.1 Register as a member

The file register.html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>register</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="register.php">
  <p>your username (no longer than 12)
    <input name="username" type="text" size="12" maxlength="12" />
  </p>
  <p>&nbsp;</p>
  <p>your password (no longer than 12)
    <input name="password" type="password" size="12" maxlength="12" />
  </p>
  <p>&nbsp;</p>
  <p>confirm your password (no longer than 12)
    <input name="repassword" type="password" size="12" maxlength="12" />
  </p>
```

```

<p>&nbsp;</p>
<p>E-mail address <input name="email" type="text" size="20" maxlength="40" /></p>
<p>Subscribe Events</p>
<p><input name="concert" type="checkbox" value="concert" />
concert&nbsp;</p>
<p><input name="musical" type="checkbox" value="musical" />
musical&nbsp;</p>
<p><input name="opera" type="checkbox" value="opera" />
opera&nbsp;</p>
<p><input name="theatre" type="checkbox" value="theatre" />
theatre&nbsp;</p>
<p><input name="conference" type="checkbox" value="conference" />
conference&nbsp;</p>
<p><input name="other" type="checkbox" value="other" />
other&nbsp;</p>
<p>
  <input name="submit" type="submit" value="submit" />
</p>
</form>
</body>
</html>

```

Register.php:

```

<?php
$username=$_POST['username'];
$password=$_POST['password'];
$repassword=$_POST['repassword'];
$email=$_POST['email'];
if($username=="")
{
  echo      'username      cant      be      empty,please      back      <a
href="http://localhost/eventsearch/register.html">register</a>';
  exit;
}
if($password=="")
{
  echo      'password      cant      be      empty,please      back      <a
href="http://localhost/eventsearch/register.html">register</a>';
  exit;
}
if($email=="")
{
  echo      'E-mail      address      cant      be      empty,please      back      <a

```

```

href="http://localhost/eventsearch/register.html">register</a>;
    exit;
}
if($password!=$repassword)
{
    echo 'repassword doesnt match password,please back <a href=
        "http://localhost/eventsearch/register.html">register</a>;
    exit;
}
if (!preg_match("/^([a-zA-Z0-9])+@([a-zA-Z0-9_-]+)(.[a-zA-Z0-9_-]+)"/, $_POST['email']))
{

    echo 'Invalid E-mail address,please back <a href=
        "http://localhost/eventsearch/register.html">register</a>;
    exit;
}
if(!empty($_POST["concert"])){
$concert=$_POST["concert"];}
if(!empty($_POST["musical"])){
$musical=$_POST["musical"];}
if(!empty($_POST["opera"])){
$opera=$_POST["opera"];}
if(!empty($_POST["theatre"])){
$theatre=$_POST["theatre"];}
if(!empty($_POST["conference"])){
$conference=$_POST["conference"];}
if(!empty($_POST["other"])){
$other=$_POST["other"];}
$link=mysql_connect("localhost","root");
mysql_select_db("eventsearch");
$query="select username from user where username='$username' ";
$result=mysql_query($query);
if(!($answer=mysql_fetch_array($result)))
{
    $query="insert
                into
                user
(username,password,email,concert,musical,opera,theatre,conference,other)
('$username','$password','$email','$concert','$musical','$opera','$theatre','$conference','$other)";
    if(mysql_query($query))
    {
        mysql_close($link);
        setcookie('username',$username,time()+3600);
        header("Location:search.html");
        exit;
    }
}

```

```

        mysql_close($link);
        echo 'registration failed,please back <a
href="http://localhost/eventsearch/register.html">register</a>';
        exit;
    }
    echo 'there is a same username,please back <a
href="http://localhost/eventsearch/register.html">register</a>';
    mysql_close($link);
?>

```

11.2 User log in and out

Login.html:

```

<form id="form1" name="form1" method="post" action="login.php">
  <h1>User Login</h1>
  <h2>username
    <input name="uname" type="text" size="12" maxlength="12" /></h2>
  <p>&nbsp;</p>
  <h2>password
    <input name="passwd" type="password" size="12" maxlength="12" />
  </h2>
  <p>&nbsp;</p>
  <p><input name="login" type="submit" value="login" />&nbsp;</p>
  <p><a href="register.html">I'm a new user</a></p>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
</form>

```

Login.php:

```

<?php
$username=$_POST['uname'];
$password=$_POST['passwd'];
if($username=="||$password=="")
{
    echo 'wrong information,please back <a
href="http://localhost/eventsearch/login.html">login</a>';
    exit;
}
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch");

```

```

$query="select password from user where username='$username'";
$result=mysql_query($query);
if($answer=mysql_fetch_array($result));
{
if($password==$answer["password"])
{
setcookie("username",$username,time()+3600);
mysql_close($link);
header("Location:add.php");
exit;
} }
mysql_close($link);
echo 'not exist user or wrong password,please try again <a
href="http://localhost/eventsearch/login.html">login</a>
,or <a href="http://localhost/eventsearch/register.html">register</a> first';
?>

```

Logout.php:

```

<?php
setcookie("username", "", time() - 3600);
?>

```

11.3 Add events

Add.php is to add four type of events, concert, musical, opera, and theatre:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<script type="text/javascript">
<!--
function MM_jumpMenu(targ,selObj,restore){ //v3.0
eval(targ+".location='"+selObj.options[selObj.selectedIndex].value+"'");
if (restore) selObj.selectedIndex=0;
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>add</title>
</head>
<body>
<form action="addevent.php?type=<?php echo $_GET['action'] ?>" method="post">
<h1>Add Event </h1>

```

```

<h2>*General Attributes</h2>
<p><select name="jumpMenu" id="jumpMenu" onchange="MM_jumpMenu('parent',this,0)">
  <option value="">choose a type</option>
  <option value="add.php?action=concert">concert</option>
  <option value="add.php?action=musical">musical</option>
  <option value="add.php?action=opera">opera</option>
  <option value="add.php?action=theatre">theatre</option>
  <option value="addcon.php?action=conference">conference</option>
  <option value="addother.php?action=other">other</option>
</select>
</p>
<p>
  <label for="addConferenceForm:dispName">Display name:</label><input
  name="displayname" type="text" size="25" maxlength="50" />
</p>
<p>
  <label for="addConferenceForm:fullName">Full name: </label>
  <input name="fullname" type="text" size="50" maxlength="100" />
</p>
<p>
  <label for="addConferenceForm:keywords">Top Keywords: </label>
  <label>
  <textarea name="topkeywords" cols="" rows="3"></textarea>
  </label>
</p>
<p>&nbsp;</p>
<h2>Locations and Websites </h2>
<table width="704" height="91" border="1">
  <tr>
    <td width="88">Year</td>
    <td width="203">Location(City,Country)</td>
    <td width="391">Website</td>
  </tr>
  <tr>
    <td><input name="year" type="text" size="4" maxlength="4" />&nbsp;</td>
    <td><input name="location" type="text" size="30" maxlength="100" />&nbsp;</td>
    <td><input name="website" type="text" size="25" maxlength="100" /></td>
  </tr>
</table>
<p>&nbsp;</p>
<h2>Important Dates</h2>
<table width="736" height="92" border="1">
  <tr>

```



```

        <td width="204">Tickets still available? </td>
        <td width="264">*Start Date(YYYY-MM-DD hh:mm:ss) </td>
        <td width="246">*End Date(YYYY-MM-DD hh:mm:ss)</td>
    </tr>
    <tr>
        <td><label>
            <select name="tickets">
                <option>Yes</option>
                <option>No</option>
            </select>
        </label></td>
        <td><input name="startdate" type="text" size="15" maxlength="25" /></td>
        <td><input name="enddate" type="text" size="15" maxlength="25" /></td>
    </tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p><input name="Insert" type="submit" value="Insert" />&nbsp;<
    <input name="Cancel" type="reset" value="Cancel" />
</p>
<p>
    <label></label>
    <label></label>
    <p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>
    <label for="addConferenceForm:fullName"></label>
</p>
<p>&nbsp;</p>
</form>
</body>
</html>

```

Addcon.php and addother.php is to add events of conference and other respectively, similarly as add.php, ignoring here.

When a member fills information of an event, addevent.php would collect the information, save in the database, geocode the address, create an iCal file and send E-mail to subscribing users:

```

<?php
    if(!$_COOKIE["username"])
    {
        echo 'please back <a href="login.html">login</a>';
    }

```

```

}
else
{
    define("MAPS_HOST", "maps.google.com");
    define("KEY",
        "ABQIAAAAan_yWEGNyLH_GB43JkuJxhTE5397KEOkSQkq_59pmCk-UQrEkRSTHh2
CrrMAPbD8ahwaXhHj9rp-BQ");
    // Initialize delay in geocode speed
    $delay = 0;
    $base_url = "http://" . MAPS_HOST . "/maps/geo?output=csv&key=" . KEY;
    // Iterate through the rows, geocoding each address
    if ($location=$_POST['location']) {
        $geocode_pending = true;
        if ($geocode_pending) {
            $location=$_POST['location'];
            $request_url = $base_url . "&q=" . urlencode($location);
            $csv = file_get_contents($request_url) or die("url not loading");
            $csvSplit = split(",", $csv);
            $status = $csvSplit[0];
            $lat = $csvSplit[2];
            $lng = $csvSplit[3];
            if (strcmp($status, "200") == 0) {
                // successful geocode
                $geocode_pending = false;
                $lat = $csvSplit[2];
                $lng = $csvSplit[3];
            }
            else if (strcmp($status, "620") == 0) {
                // sent geocodes too fast
                $delay += 100000;
            } else {
                // failure to geocode
                $geocode_pending = false;
                echo "Address " . $address . " failed to geocoded. ";
                echo "Received status " . $status . "\n";
            }
        }
        usleep($delay);
    }
}
$displayname=$_POST['displayname'];
$fullname=$_POST['fullname'];
$topkeywords=$_POST['topkeywords'];
$year=$_POST['year'];
$location=$_POST['location'];
$website=$_POST['website'];

```

```

$registration=$_POST['registration'];
$submission=$_POST['submission'];
$notification=$_POST['notification'];
$finalversion=$_POST['finalversion'];
$startdate=$_POST['startdate'];
$enddate=$_POST['enddate'];
$type=$_GET['type'];
$dates=$_POST['dates'];
$tickets=$_POST['tickets'];
if ($stopkeywords==true && $displayname==true && $fullname==true && $type==true &&
$startdate==true && $enddate==true)
{
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch");
switch ($_GET['type'])
{
case "concert":
$query1="select Displayname from concert where Displayname='$displayname' ";
$result1=mysql_query($query1);
if(!($answer1=mysql_fetch_array($result1))){
$member_addc="insert into concert
(code,Displayname,Fullname,Topkeywords,Year,location,website,lat,lng,tickets,startdate,enddate)
values
('e1','$displayname','$fullname','$stopkeywords','$year','$location','$website','$lat','$lng','$tickets','$
startdate','$enddate)";
$member_add_savec=mysql_query($member_addc,$link) or die (mysql_error()); }
else{
echo 'there is a same displayname,please back <a
href="http://localhost/eventsearch/add.php">addevent</a>;'
break;
case "musical":
$query2="select Displayname from musical where Displayname='$displayname' ";
$result2=mysql_query($query2);
if(!($answer2=mysql_fetch_array($result2))){
$member_addm="insert into musical
(code,Displayname,Fullname,Topkeywords,Year,location,website,lat,lng,tickets,startdate,enddate)
values
('e2','$displayname','$fullname','$stopkeywords','$year','$location','$website','$lat','$lng','$tickets','$
startdate','$enddate)";
$member_add_savem=mysql_query($member_addm,$link) or die (mysql_error());
}
else{
echo 'there is a same displayname,please back <a
href="http://localhost/eventsearch/add.php">addevent</a>;'

```

```

break;
case "opera":
$query3="select Displayname from opera where Displayname='$displayname' ";
$result3=mysql_query($query3);
if(!($answer3=mysql_fetch_array($result3))){
$member_addo="insert                into                opera
(code,Displayname,Fullname,Topkeywords,Year,location,website,lat,lng,tickets,startdate,enddate)
values
('e3','$displayname','$fullname','$stopkeywords','$year','$location','$website','$lat','$lng','$tickets','$
startdate','$enddate)";
$member_add_saveo=mysql_query($member_addo,$link) or die (mysql_error()); }
else{
echo 'there is a same displayname,please back <a
href="http://localhost/eventsearch/add.php">addevent</a>';}
break;
case "theatre":
$query4="select Displayname from theatre where Displayname='$displayname' ";
$result4=mysql_query($query4);
if(!($answer4=mysql_fetch_array($result4))){
$member_addt="insert into theatre
(code,Displayname,Fullname,Topkeywords,Year,location,website,lat,lng,tickets,startdate,enddate)
values
('e4','$displayname','$fullname','$stopkeywords','$year','$location','$website','$lat','$lng','$tickets','$
startdate','$enddate)";
$member_add_savet=mysql_query($member_addt,$link) or die (mysql_error()); }
else{
echo 'there is a same displayname,please back <a
href="http://localhost/eventsearch/add.php">addevent</a>';}
break;
case "conference":
$query5="select Displayname from conference where Displayname='$displayname' ";
$result5=mysql_query($query5);
if(!($answer5=mysql_fetch_array($result5))){
$member_addc="insert into conference
(code,Displayname,Fullname,Topkeywords,Year,location,lat,lng,website,Registration,Submission,
Notification,FinalVersion,startdate,enddate) values
('e5','$displayname','$fullname','$stopkeywords','$year','$location','$lat','$lng',
'website','$registration','$submission','$notification','$finalversion','$startdate','$enddate)";
$member_add_savec=mysql_query($member_addc,$link) or die (mysql_error()); }
else{
echo 'there is a same displayname,please back <a
href="http://localhost/eventsearch/add.php">addevent</a>';}
break;
case "other":

```

```

$query6="select Displayname from other where Displayname='$displayname' ";
$result6=mysql_query($query6);
if(!$answer6=mysql_fetch_array($result6)){
$member_addc="insert into other
(code,Displayname,Fullname,Topkeywords,Year,location,lat,lng,website,dates,startdate,enddate)
values      ('e6','$displayname','$fullname','$topkeywords','$year','$location','$lat','$lng',
'$website','$dates','$startdate','$enddate)";
$member_add_savec=mysql_query($member_addc,$link) or die (mysql_error());  }
else{
echo 'there is a same displayname,please back <a
href="http://localhost/eventsearch/add.php">addevent</a>';}
break;
}
require_once( 'iCalcreator.class.php' );
$v = new vcalendar();
// create a new calendar instance
$v->setConfig( 'unique_id', 'events' );
// set Your unique id
$v->setProperty( 'method', 'PUBLISH' );
// required of some calendar software
$v->setProperty( "x-wr-calname", " Calendar event" );
// required of some calendar software
$v->setProperty( "X-WR-CALDESC", "Calendar Description" );
// required of some calendar software
$v->setProperty( "X-WR-TIMEZONE", "Europe/Stockholm" );
// required of some calendar software
$vevent = new vevent();
// create an event calendar component
$vevent->setProperty( 'title', "$displayname" );
$vevent->setProperty( 'summary', "$fullname" );
$vevent->setProperty( 'description', "$topkeywords $dates" );
$vevent->setProperty( 'LOCATION', "$location" );
$vevent->setProperty( 'attachments', "$website" );
$vevent->setProperty( 'dtstart', "$startdate");
$vevent->setProperty( 'dtend', "$enddate");
$v->setComponent ( $vevent );
// add event to calendar
$v->setConfig( 'directory', 'd:/upload/' );
// set directory
$v->setConfig( 'filename', "$type$displayname.ics" );
// set file name
$v->saveCalendar();
// save calendar to file
if($type=="concert"){

```

```

$code="e1";}
if($type=="musical"){
$code="e2";}
if($type=="opera"){
$code="e3";}
if($type=="theatre"){
$code="e4";}
if($type=="conference"){
$code="e5";}
if($type=="other"){
$code="e6";}
$query7="select id from concert where Displayname='$displayname' and code='$code'
union all
select id from musical where Displayname='$displayname' and code='$code'
union all
select id from opera where Displayname='$displayname' and code='$code'
union all
select id from theatre where Displayname='$displayname' and code='$code'
union all
select id from conference where Displayname='$displayname' and code='$code'
union all
select id from other where Displayname='$displayname' and code='$code'";
$result7=mysql_query($query7);
$answer7=mysql_fetch_array($result7);
$id=$answer7["id"];
$query="select username,email from user where concert='$type' or musical='$type' or
opera='$type' or theatre='$type' or conference='$type' or other='$type'";
$result=mysql_query($query);
require("class.phpmailer.php");
while ($row = @mysql_fetch_assoc($result)){
$mail = new PHPMailer();
$mail->IsSMTP();
$mail->Host = "smtp.sina.com";
$mail->SMTPAuth = true;
$mail->Username = "sif99";
$mail->Password = "*****";
$mail->From = "sif99@sina.com";
$mail->FromName = "ss";
$mail->AddAddress($row['email'], $row['username']);
$mail->AddAttachment("d:/upload/$type$displayname.ics");
$mail->Subject = "$displayname";
$mail->Body = "There is a new
event,http://localhost/eventsearch/eventdetails.php?id=$id&code=$code";
$mail->AltBody = "";

```

```

if(!$mail->Send())
{
    echo "";
    exit;
}
echo "";
}}
else
    {
        echo 'please fill displayname,fullname,topkeywords,startdate and enddate or choose a
type,back <a href="add.php">addevent</a>';}}
?>

```

11.4 Search events

Search.html for general search:

```

<form id="form1" name="form1" method="post" action="searchgeneral.php">
    <p><a href="register.html">Register a new member</a></p>
    <p><a href="login.html">Login</a></p>
    <p><a href="add.php">Add events</a></p>
    <p><a href="#"
        onclick="clearFormHiddenParams__id0('__id0');document.forms['__id0']['_id0:_idcl'].value='_id
0:_id35'; document.forms['__id0'].submit(); return false;">General</a> &nbsp;&nbsp;&nbsp;<a
href="searchkeyword.html"
onclick="clearFormHiddenParams__id0('__id0');document.forms['__id0']['_id0:_idcl'].value='_id0:_
id35'; document.forms['__id0'].submit(); return false;">Keyword</a> </p>
    <tr></tr>
    </p>
    <input name="search" type="text" size="20" maxlength="100" />
<input name="search2" type="submit" value="search" />
</form>

```

Searchgeneral.php for general search and shows the results:

```

<?php
$search=$_POST['search'];
if ($search==""){
header("Location:search.html");
}
$search=rtrim(ltrim($search));
$keyword=split(" ",$search);
$max=count($keyword);
static $query1="";

```

```

static $query2="";
static $query3="";
if ($max==1){
$i=0;
$query1=$query1."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
      Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
      location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or startdate
      like '%$keyword[$i]%' or enddate like '%$keyword[$i]%'";
$query2=$query2."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
      Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
      location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or
      Registration like '%$keyword[$i]%' or Submission like '%$keyword[$i]%' or
Notification like '%$keyword[$i]%' or FinalVersion like '%$keyword[$i]%' or
startdate like '%$keyword[$i]%' or enddate like '%$keyword[$i]%'";
$query3=$query3."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
      Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
      location like '%$keyword[$i]%' or website like '%$keyword[$i]%'
      or dates like '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or
enddate like '%$keyword[$i]%'";
}
else{
for ($i = 0; $i < $max; $i++){
if ($i==0){
$query1=$query1 ."(Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
      Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
      location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or
      startdate like '%$keyword[$i]%' or enddate like '%$keyword[$i]%' or ";
$query2=$query2 ."(Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
      Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
      location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or
      Registration like '%$keyword[$i]%' or Submission like
      '%$keyword[$i]%' or Notification like '%$keyword[$i]%' or FinalVersion
      like '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or
      enddate like '%$keyword[$i]%' or ";
$query3=$query3."(Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
      Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
      location like '%$keyword[$i]%' or website like '%$keyword[$i]%'
      or dates like '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or
      enddate like '%$keyword[$i]%' or ";
}
else{
if ($i==( $max-1)){
$query1=$query1 . "Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
      Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or

```



```

        location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or
        like '%$keyword[$i]%' or enddate like '%$keyword[$i]%'");
$query2=$query2 ."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
        Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
        location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or
        Registration like '%$keyword[$i]%' or Submission like '%$keyword[$i]%' or
        Notification like '%$keyword[$i]%' or FinalVersion like '%$keyword[$i]%'
        or startdate like '%$keyword[$i]%' or enddate like '%$keyword[$i]%'");
$query3=$query3."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
        Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
        location like '%$keyword[$i]%' or website like '%$keyword[$i]%'
        or dates like '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or
        enddate like '%$keyword[$i]%'");
    }
    else{
        $query1=$query1 ."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
        Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
        location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or
        like '%$keyword[$i]%' or enddate like '%$keyword[$i]%' or ";
        $query2=$query2 ."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
        Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
        location like '%$keyword[$i]%' or website like '%$keyword[$i]%' or
        Registration like '%$keyword[$i]%' or Submission like '%$keyword[$i]%' or
        FinalVersion like '%$keyword[$i]%' or
        startdate like '%$keyword[$i]%' or enddate like '%$keyword[$i]%' or ";
        $query3=$query3."Displayname like '%$keyword[$i]%' or Fullname like '%$keyword[$i]%' or
        Topkeywords like '%$keyword[$i]%' or Year like '%$keyword[$i]%' or
        location like '%$keyword[$i]%' or website like '%$keyword[$i]%'
        or dates like '%$keyword[$i]%' or startdate like '%$keyword[$i]%' or
        enddate like '%$keyword[$i]%' or ";
    }
}
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch") or die("could not select database");
$rowc=mysql_query("select count(*) from concert where " . $query1);
$rowm=mysql_query("select count(*) from musical where " . $query1);
$rowt=mysql_query("select count(*) from theatre where " . $query1);
$rowo=mysql_query("select count(*) from opera where " . $query1);
$rowcon=mysql_query("select count(*) from conference where " . $query2);
$rowoth=mysql_query("select count(*) from other where " . $query3);
$myrowc=mysql_fetch_array($rowc);
$myrowm=mysql_fetch_array($rowm);
$myrowt=mysql_fetch_array($rowt);
$myrowo=mysql_fetch_array($rowo);
$myrowcon=mysql_fetch_array($rowcon);

```

```

$myrowoth=mysql_fetch_array($rowoth);
$numrows=$myrowc[0]+$myrowm[0]+$myrowt[0]+$myrowo[0]+$myrowcon[0]+$myrowoth[0]
;
$pageSize=10;
$pages=intval($numrows/$pageSize);
if ($numrows%$pageSize)
$page++;
if (isset($_GET['page'])){
$page=intval($_GET['page']);
}
else{
$page=1;
}
$offset=$pageSize*($page-1);
$query="select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
        " as Notification," as FinalVersion,startdate,enddate," as dates,'tbl1' as tbl from concert where
" . $query1
        ."union all
        select id,code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
        " as Notification," as FinalVersion,startdate,enddate," as dates, 'tbl2' as tbl from musical
where " . $query1
        ."union all
        select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
        " as Notification," as FinalVersion,startdate,enddate," as dates, 'tbl3' as tbl from theatre where
" . $query1
        ."union all
        select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
        " as Notification," as FinalVersion,startdate,enddate," as dates, 'tbl4' as tbl from opera where
" . $query1
        ."union all
        select id, code,Displayname,Fullname,Topkeywords,Year,location,website,
Registration,Submission,Notification,FinalVersion,startdate,enddate," as dates, 'tbl5' as tbl from
conference where " . $query2
        ."union all
        select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
        " as Notification," as FinalVersion,startdate,enddate,dates, 'tbl6' as tbl from other where
" . $query3
        ."limit $offset,$pageSize";
$result=mysql_query($query) or die("query failed:".mysql_error());

```

```

if ($answer=mysql_fetch_array($result))
{
$url=&$answer["website"];
$i=0;
?>
<table border="0" width="80%">
<tr>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">WWW</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Name</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Location</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Modify</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Nearby</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">SameDateEvents</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">iCalFile</td>
</tr>
<?php
do {
$i++;
?>
<tr>
<td width="20%"><a href="http://<?php echo $answer["website"]?>">WWW</a></td>
<td width="20%"><a href="eventdetails.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>"><?php echo $answer["Displayname"];?></a></td>
<td width="20%"><a href="google.html?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>"><?php echo $answer["location"];?></a></td>
<td width="20%"><a href="modify.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">Modify</a></td>
<td width="20%"><a href="nearby.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">nearby</a></td>
<td width="20%"><a href="date.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">date</a></td>
<td width="20%"><a href="download.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">Download</a></td>
</tr>
<?php
}
while ($answer=mysql_fetch_array($result));

```

```

echo "</table>";
}
echo "<div align='center'>".$pages." pages(".$page."/".$pages.")";
for ($i=1;$i<=$page;$i++)
echo "<a href='resultsgeneral.php?page=".$i."&search=".$search."'>[".$i."]</a>";
echo "[".$page."]";
for ($i=$page+1;$i<=$pages;$i++)
echo "<a href='resultsgeneral.php?page=".$i."&search=".$search."'>[".$i."]</a>";
echo "</div>";
?>

```

Resultgeneral.php is basically same as searchgeneral.php but the way it receives a variable \$search. It receives the search keyword from searchgeneral.php through the link:

```

<?php
$search=$_GET['search'];
?>

```

Searchkeyword.html is for keyword search:

```

<form id="form1" name="form1" method="post" action="searchkeyword.php">
  <p>&nbsp;</p>
  <p><a href="register.html">Register a new member</a></p>
  <p><a href="login.html">Login</a></p>
  <p><a href="add.php">Add events</a></p>
  <p><a href="searchgeneral.html"
onclick="clearFormHiddenParams__id0('__id0');document.forms['__id0']['_id0:_idcl'].value='_id0:_id35'; document.forms['__id0'].submit(); return false;">General</a> &nbsp;&nbsp;&nbsp;<a
href="searchkeyword.html"
onclick="clearFormHiddenParams__id0('__id0');document.forms['__id0']['_id0:_idcl'].value='_id0:_id35'; document.forms['__id0'].submit(); return false;">Keyword</a> </p>
  <tr></tr>
  </p>
  <input name="search" type="text" size="20" maxlength="100" />
  <input name="search2" type="submit" value="search" />
</form>

```

Searchkeyword.php is for keyword search and shows the results:

```

<?php
$search=$_POST['search'];
if ($search=="") {
header("Location:searchkeyword.html");
}

```

```

function searchReplace(&$search){
$search = str_replace(" ", "$search");
$search
=
str_replace(array('~','`','#','@','!','$','%','^','&','*','(',')','-', '_', '=', '+', '[', ']', '{', '}', '|', chr(92), chr(47), ':', ';', '"',
''', '?', '>', '<', ',', '.', ''), $search);
return $search;
}
searchReplace($search);
$search=rtrim(ltrim($search));
$keyword=split(" ", $search);
$max=count($keyword);
static $query1="";
static $query2="";
static $query3="";
$link=mysql_connect("localhost", "root") or die(mysql_error());
mysql_select_db("eventsearch") or die("could not select database");
if ($max==1){
$i=0;
$query1=$query1."Displayname regexp '[:<:]]$keyword[$i][:>:]' or
Fullname regexp '[:<:]]$keyword[$i][:>:]' or
Topkeywords regexp '[:<:]]$keyword[$i][:>:]' or
Year regexp '[:<:]]$keyword[$i][:>:]' or
location regexp '[:<:]]$keyword[$i][:>:]' or
website regexp '[:<:]]$keyword[$i][:>:]' or
startdate regexp '[:<:]]$keyword[$i][:>:]' or
enddate regexp '[:<:]]$keyword[$i][:>:]'";
$query2=$query2."Displayname regexp '[:<:]]$keyword[$i][:>:]' or
Fullname regexp '[:<:]]$keyword[$i][:>:]' or
Topkeywords regexp '[:<:]]$keyword[$i][:>:]' or
Year regexp '[:<:]]$keyword[$i][:>:]' or
location regexp '[:<:]]$keyword[$i][:>:]' or
website regexp '[:<:]]$keyword[$i][:>:]' or
Registration regexp '[:<:]]$keyword[$i][:>:]' or
Notification regexp '[:<:]]$keyword[$i][:>:]' or
Submission regexp '[:<:]]$keyword[$i][:>:]' or
FinalVersion regexp '[:<:]]$keyword[$i][:>:]' or
startdate regexp '[:<:]]$keyword[$i][:>:]' or
enddate regexp '[:<:]]$keyword[$i][:>:]'";
$query3=$query3."Displayname regexp '[:<:]]$keyword[$i][:>:]' or
Fullname regexp '[:<:]]$keyword[$i][:>:]' or
Topkeywords regexp '[:<:]]$keyword[$i][:>:]' or
Year regexp '[:<:]]$keyword[$i][:>:]' or
location regexp '[:<:]]$keyword[$i][:>:]' or
website regexp '[:<:]]$keyword[$i][:>:]' or

```

```

        dates regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]";
    }
else{
for ($i = 0; $i < $max; $i++){
if ($i==0){
$query1=$query1 . "(Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]' or ";
$query2=$query2 . "(Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        Registration regexp '[:<:)]$keyword[$i][:>:)]' or
        Notification regexp '[:<:)]$keyword[$i][:>:)]' or
        Submission regexp '[:<:)]$keyword[$i][:>:)]' or
        FinalVersion regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]' or ";
$query3=$query3 . "(Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        dates regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]' or ";
    }
else{
    if ($i==($max-1)){
        $query1=$query1 . "Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or

```

```

        website regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]";
$query2=$query2."Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        Registration regexp '[:<:)]$keyword[$i][:>:)]' or
        Notification regexp '[:<:)]$keyword[$i][:>:)]' or
        Submission regexp '[:<:)]$keyword[$i][:>:)]' or
        FinalVersion regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]";
$query3=$query3."Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        dates regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]";
    }
    else{
        $query1=$query1 . "Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]' or ";
        $query2=$query2."Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        Registration regexp '[:<:)]$keyword[$i][:>:)]' or
        Notification regexp '[:<:)]$keyword[$i][:>:)]' or
        Submission regexp '[:<:)]$keyword[$i][:>:)]' or
        FinalVersion regexp '[:<:)]$keyword[$i][:>:)]' or

```

```

        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]' or ";
$query3=$query3."Displayname regexp '[:<:)]$keyword[$i][:>:)]' or
        Fullname regexp '[:<:)]$keyword[$i][:>:)]' or
        Topkeywords regexp '[:<:)]$keyword[$i][:>:)]' or
        Year regexp '[:<:)]$keyword[$i][:>:)]' or
        location regexp '[:<:)]$keyword[$i][:>:)]' or
        website regexp '[:<:)]$keyword[$i][:>:)]' or
        dates regexp '[:<:)]$keyword[$i][:>:)]' or
        startdate regexp '[:<:)]$keyword[$i][:>:)]' or
        enddate regexp '[:<:)]$keyword[$i][:>:)]' or ";
    }}}
$rowc=mysql_query("select count(*) from concert where " . $query1);
$rowm=mysql_query("select count(*) from musical where " . $query1);
$rowt=mysql_query("select count(*) from theatre where " . $query1);
$rowo=mysql_query("select count(*) from opera where " . $query1);
$rowcon=mysql_query("select count(*) from conference where " . $query2);
$rowoth=mysql_query("select count(*) from other where " . $query3);
$myrowc=mysql_fetch_array($rowc);
$myrowm=mysql_fetch_array($rowm);
$myrowt=mysql_fetch_array($rowt);
$myrowo=mysql_fetch_array($rowo);
$myrowcon=mysql_fetch_array($rowcon);
$myrowoth=mysql_fetch_array($rowoth);
$numrows=$myrowc[0]+$myrowm[0]+$myrowt[0]+$myrowo[0]+$myrowcon[0]+$myrowoth[0]
;
$pageSize=10;
$pages=intval($numrows/$pageSize);
if ($numrows%$pageSize)
    $pages++;
if (isset($_GET['page'])) {
    $page=intval($_GET['page']);
}
else {
    $page=1;
}
$offset=$pageSize*($page-1);
$query="select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
        " as Notification," as FinalVersion,startdate,enddate," as dates, 'tbl1' as tbl from concert where
" . $query1
        ."union all
        select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,

```



```

" as Notification," as FinalVersion,startdate,enddate," as dates, 'tbl2' as tbl from musical
where " . $query1
    ."union all
    select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
" as Notification," as FinalVersion,startdate,enddate," as dates, 'tbl3' as tbl from theatre where
" . $query1
    ."union all
    select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
" as Notification," as FinalVersion,startdate,enddate, " as dates,'tbl4' as tbl from opera where
" . $query1
    ."union all
    select
                                                    id,
code,Displayname,Fullname,Topkeywords,Year,location,website,Registration,Submission,
Notification,FinalVersion,startdate,enddate, " as dates,'tbl5' as tbl from conference where
" . $query2
    ."union all
    select id, code,Displayname,Fullname,Topkeywords,Year,location,website,"as
Registration," as Submission,
" as Notification," as FinalVersion,startdate,enddate,dates, 'tbl6' as tbl from other where
" . $query3
    ."limit $offset,$pagesize";
$result=mysql_query($query) or die("query failed:".mysql_error());
if ($answer=mysql_fetch_array($result))
{
$url=&$answer["website"];
$i=0;
?>
<table border="0" width="80%">
<tr>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">WWW</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Name</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Location</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Modify</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">Nearby</td>
<td width="20%" bgcolor="#E0E0E0">
<p align="center">SameDateEvents</td>
<p align="center">iCalFile</td>

```

```

</tr>
<?php
do {
$i++;
?>
<tr>
<td width="20%"><a href="http://<?php echo $answer["website"]?>">WWW</a></td>
<td width="20%"><a href="eventdetails.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>"><?php echo $answer["Displayname"];?></a></td>
<td width="20%"><a href="google.html?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>"><?php echo $answer["location"];?></a></td>
<td width="20%"><a href="modify.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">Modify</a></td>
<td width="20%"><a href="nearby.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">nearby</a></td>
<td width="20%"><a href="date.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">date</a></td>
<td width="20%"><a href="download.php?id=<?php echo $answer["id"]?>&code=<?php echo
$answer["code"]?>">Download</a></td>
</tr>
<?php
}
while ($answer=mysql_fetch_array($result));
echo "</table>";
}
echo "<div align='center'>".$pages." pages('".$page."//".$pages.)";
for ($i=1;$i<$page;$i++)
echo "<a href='resultskeyword.php?page=".$i."&search=".$search.">[".$i.]</a>";
echo "[".$page.]";
for ($i=$page+1;$i<=$pages;$i++)
echo "<a href='resultskeyword.php?page=".$i."&search=".$search.">[".$i.]</a>";
echo "</div>";
?>

```

Resultkeyword.php is basically same as searchgeneral.php but the way it receives a variable \$search:

```

<?php
$search=$_GET['search'];
?>

```

11.5 Event details

Eventdetails.php shows four type event details, concert, musical, opera and theatre:

```

<?php
$id=$_GET['id'];
$code=$_GET['code'];
if($code==e5){
header("Location:condetails.php?id=$id&code=$code");}
else{
if($code==e6){
header("Location:otherdetails.php?id=$id&code=$code");}
else{
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch") or die("could not select database");
$query="select * from concert where id='$id' and code='$code'
        union all
        select * from musical where id='$id' and code='$code'
        union all
        select * from theatre where id='$id' and code='$code'
        union all
        select * from opera where id='$id' and code='$code'";
$result=mysql_query($query) or die("query failed:".mysql_error());
$answer=mysql_fetch_array($result);
}}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title> </title>
</head>
<body>
<table width="100%">
<tbody>
<tr>
<td class="titleLeft"><table>
<tbody>
<tr>
<td><h2><span class="h2"><?php echo $answer["Displayname"];?></span></h2></td>
</tr>
<tr>
<td><h3><span class="h3">Conference Details</span></h3></td>
</tr>
</tbody>
</table></td>

```

```

<h3>General Attributes</h3>
<table>
<tbody>
<tr class="heightEqualButton">
<td class="tdNoWrap"><label for="_id0:dispName" style="font-style:italic;">
Display name: </label></td>
<td class="td"><span id="_id0:dispName"><?php echo $answer["Displayname"];?></span></td>
</tr>
<tr class="heightEqualButton">
<td class="tdNoWrap"><label for="_id0:fullname" style="font-style:italic;">
Full name: </label></td>
<td class="td"><span id="_id0:fullname"><?php echo $answer["Fullname"];?></span></td>
</tr>
<tr class="heightEqualButton">
<td class="tdNoWrap"><label for="_id0:keyword" style="font-style:italic;">
Top Keyword: </label></td>
<td class="td"><span id="_id0:keyword"><?php echo $answer["Topkeywords"];?></span></td>
</tr>
</tbody>
</table>
<div style="height:15px;"></div>
<h3>Locations and Websites</h3>
<table cellpadding="0" cellspacing="0" width="100%">
<thead>
<tr><th width="21%" class="bottom">Year
</th>
<th width="42%" class="bottom">Location</th><th width="37%"
class="bottom">Website</th></tr></thead>
<tbody id="_id0:_id17:tbody_element">
<tr class="heightEqualButton">
<td class="yearColumn"><?php echo $answer["Year"];?></td>
<td class="nameColum"><?php echo
$answer["location"];?></td>
<td class="nameColumn"><a href="http://<?php echo $answer["website"];?>"><?php echo
$answer["website"];?></a></td>
</tr>
</tbody>
</table>
<div style="height:15px;"></div>
<h4>Important Dates</h4>
<table width="101%" cellpadding="0" cellspacing="0">
<tr>
<th width="48%" class="bottom">Tickets still available
</th>
<th width="27%" class="bottom">
Start Date </th>
</tr>

```

```

                <th width="25%" class="bottom">
                    End Date                </th>
            </tr>
            <tr class="heightEqualButton">
                <td>
                    <?php echo $answer["tickets"];?>                </td>
                <td>
                    <?php echo $answer["startdate"];?>                </td>
                <td>
                    <?php echo $answer["enddate"];?>                </td>
            </tr>
        </table>
        <div style="height:15px"></div>
        <div class="navigationDiv">
            <table>
            <p><form id="form1" name="form1" method="post" action="javascript:history.go(-1)"><input
            name="Back" type="submit" value="Back" /></form>
            <form id="form2" name="form2" method="post" action="modify.php?id=<?php echo
            $_GET['id'];?>&code=<?php echo $_GET['code'];?>">
                <input name="Edit" type="submit" value="Edit" />
            </form>
            </p>
        </body>
    </html>

```

Condetails.php and otherdetails.php show conference and other event details, similarly as eventdetails.php, ignoring here.

11.6 Modify events

Modify.php gives every detail of four type event, concert, musical, opera and theatre:

```

<?php
$id=$_GET['id'];
$code=$_GET['code'];
if($code==e5){
header("Location:modifycon.php?id=$id&code=$code");}
else{
if($code==e6){
header("Location:modifyother.php?id=$id&code=$code");}
else{
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch") or die("could not select database");
$query="select * from concert where id='$id' and code='$code'

```

```

        union all
        select * from musical where id='$id' and code='$code'
        union all
        select * from theatre where id='$id' and code='$code'
        union all
        select * from opera where id='$id' and code='$code';
$result=mysql_query($query) or die("query failed:".mysql_error());
$answer=mysql_fetch_array($result);
}}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title> </title>
</head>
<body>
<form action="eventmodify.php?id=<?php echo $_GET['id'];?>&code=<?php echo
$_GET['code'];?>" method="post">
<table width="100%">
<tbody>
<tr>
<td class="titleLeft"><table>
<tbody>
<tr>
<td><h2><span class="h1"><?php echo $answer["Displayname"];?></span></h2></td>
</tr>
<tr>
<td><h2><span class="h3">Edit </span></h2></td>
</tr>
</tbody>
</table>
</td>
</tr>
</tbody>
</table>
<h3>General Attributes</h3>
        <table>
        <tbody>
        <tr class="heightEqualButton">
        <td class="tdNoWrap"><label for="editForm:dispName" style="font-style:italic;">
Display name: </label></td>
        <td class="td"><input id="editForm:dispName" type="text" name="displayname" value="<?php

```



```

        <td>
            <input id="editForm:lastSubmission" type="text" name="tickets"
value="<?php echo $answer["tickets"];?>" />
        </td>
        <td>
            <input id="editForm:lastStart" type="text" name="Start" value="<?php
echo $answer["startdate"];?>" />
        </td>
        <td>
            <input id="editForm:lastEnd" type="text" name="End" value="<?php
echo $answer["enddate"];?>" />
        </td>
    </tr>
</table>
<p></p>
<input name="submit" type="submit" value="Commit" >
</form>
</body>
</html>

```

Eventmodify.php updates the database after event details are modified:

```

<?php
if(!$_COOKIE["username"])
{
    echo 'please back <a href="login.html">login</a>';
}
$id=$_GET['id'];
$code=$_GET['code'];
$displayname=$_POST["displayname"];
$fullname=$_POST["fullname"];
$topkeywords=$_POST["topkeywords"];
$year=$_POST["year"];
$location=$_POST["location"];
$website=$_POST["website"];
$tickets=$_POST["tickets"];
$start=$_POST["Start"];
$end=$_POST["End"];
if ($location=$_POST['location']) {
define("MAPS_HOST", "maps.google.com");
define("KEY",
"ABQIAAAAAn_yWEGNyLH_GB43JkuJxhT9M5QyglirRTSToIpEURSgIoaUpBScgy2Uh6Eo-c
6roa4o_CJZzfRoWg");
$delay = 0;
$base_url = "http://" . MAPS_HOST . "/maps/geo?output=csv&key=" . KEY;
$geocode_pending = true;
    if ($geocode_pending) {

```



```

$request_url = $base_url . "&q=" . urlencode($location);
$csv = file_get_contents($request_url) or die("url not loading");
$csvSplit = split(",", $csv);
$status = $csvSplit[0];
$lat = $csvSplit[2];
$lng = $csvSplit[3];
if (strcmp($status, "200") == 0) {
    // successful geocode
    $geocode_pending = false;
    $lat = $csvSplit[2];
    $lng = $csvSplit[3];
}
else if (strcmp($status, "620") == 0) {
    // sent geocodes too fast
    $delay += 100000;
} else {
    // failure to geocode
    $geocode_pending = false;
    echo "Address " . $address . " failed to geocoded. ";
    echo "Received status " . $status . "\n";
}
usleep($delay);
}}
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch") or die("could not select database");
if ($code=='e1'){
$query="update concert set
Displayname='$displayname',Fullname='$fullname',Topkeywords='$topkeywords',Year='$year',lo
cation='$location',website='$website',lat='$lat',lng='$lng',tickets='$tickets',startdate='$start',enddat
e='$end'
where id=$id";
mysql_query($query) or die("query failed:".mysql_error());
}
if ($code=='e2'){
$query="update musical set
Displayname='$displayname',Fullname='$fullname',Topkeywords='$topkeywords',Year='$year',lo
cation='$location',website='$website',lat='$lat',lng='$lng' ,tickets='$tickets',startdate='$start',endda
te='$end'
where id=$id";
mysql_query($query) or die("query failed:".mysql_error());
}
if ($code=='e3'){
$query="update opera set
Displayname='$displayname',Fullname='$fullname',Topkeywords='$topkeywords',Year='$year',lo

```

```

cation='$location',website='$website',lat='$lat',lng='$lng',tickets='$tickets',startdate='$start',enddate='Send'
where id=$id";
mysql_query($query) or die("query failed:".mysql_error());
}
if ($code=='e4'){
$query="update theatre set
Displayname='$displayname',Fullname='$fullname',Topkeywords='$topkeywords',Year='$year',location='$location',website='$website' ,lat='$lat',lng='$lng',tickets='$tickets',startdate='$start',enddate='Send'
where id=$id";
mysql_query($query) or die("query failed:".mysql_error());
}
?>

```

Similarly, there are 4 other files to modify conference and other events, ignoring here.

11.7 Find nearby events

Nearby.php gives choices for users, to choose events about 100km, 200km, 300km, 400km, or 500km away around:

```

<form id="form1" name="form1" method="post" action="near1.php?id=<?php echo
$_GET['id'] ?>&code=<?php echo $_GET['code'] ?>">
  <h1>Nearby Events </h1>
  <p>Find nearby events about
    <select name="near">
      <option>100km</option>
      <option>200km</option>
      <option>300km</option>
      <option>400km</option>
      <option>500km</option>
    </select>
    around </p>
  <p><input name="Submit" type="submit" value="Submit" />&nbsp;</p>
</form>
<?php
$id=$_GET['id'];
$code=$_GET['code'];
?>

```

Near1.php shows events about 100km away:

```

<?php
$id=$_GET['id'];

```

```

$code=$_GET['code'];
if($_POST["near"]==200){
header("Location:near2.php?id=$id&code=$code");}
if($_POST["near"]==300){
header("Location:near3.php?id=$id&code=$code");}
if($_POST["near"]==400){
header("Location:near4.php?id=$id&code=$code");}
if($_POST["near"]==500){
header("Location:near5.php?id=$id&code=$code");}
$connection=mysql_connect ("localhost", "root");
if (!$connection) {
die('Not connected : ' . mysql_error());
}
$db_selected = mysql_select_db("eventsearch", $connection);
if (!$db_selected) {
die ('Can\'t use db : ' . mysql_error());
}
$query = "select lat,lng from concert where id='$id' and code='$code'
union all
select lat,lng from musical where id='$id' and code='$code'
union all
select lat,lng from theatre where id='$id' and code='$code'
union all
select lat,lng from opera where id='$id' and code='$code'
union all
select lat,lng from conference where id='$id' and code='$code'
union all
select lat,lng from other where id='$id' and code='$code'";
$result=mysql_query($query) or die("query failed:".mysql_error());
$answer1=mysql_fetch_array($result);
$lat=$answer1['lat'];
$lng=$answer1['lng'];
$myrowc=0;
$myrowm=0;
$myrowt=0;
$myrowo=0;
$myrowcon=0;
$myrowoth=0;
$rowc=mysql_query("SELECT id, ( 6371 * acos( cos( radians('$lat') ) * cos( radians( lat ) ) *
cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) * sin( radians( lat ) ) ) ) AS distance
FROM concert HAVING distance < 100 ");
while ($row1 = @mysql_fetch_assoc($rowc)){
$myrowc++;}
$rowm=mysql_query("SELECT id, ( 6371 * acos( cos( radians('$lat') ) * cos( radians( lat ) ) *

```

```

cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) * sin( radians( lat ) ) ) AS distance
FROM musical HAVING distance < 100 ");
while ($row2 = @mysql_fetch_assoc($rowm)){
$myrowm++;}
$rowt=mysql_query("SELECT id, ( 6371 * acos( cos( radians('$lat') ) * cos( radians( lat ) ) *
cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) * sin( radians( lat ) ) ) ) AS distance
FROM theatre HAVING distance < 100 ");
while ($row3 = @mysql_fetch_assoc($rowt)){
$myrowt++;}
$rowo=mysql_query("SELECT id, ( 6371 * acos( cos( radians('$lat') ) * cos( radians( lat ) ) *
cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) * sin( radians( lat ) ) ) ) AS distance
FROM opera HAVING distance < 100 ");
while ($row4 = @mysql_fetch_assoc($rowo)){
$myrowo++;}
$rowcon=mysql_query("SELECT id, ( 6371 * acos( cos( radians('$lat') ) * cos( radians( lat ) ) *
cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) * sin( radians( lat ) ) ) ) AS distance
FROM conference HAVING distance < 100 ");
while ($row5 = @mysql_fetch_assoc($rowcon)){
$myrowcon++;}
$rowoth=mysql_query("SELECT id, ( 6371 * acos( cos( radians('$lat') ) * cos( radians( lat ) ) *
cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) * sin( radians( lat ) ) ) ) AS distance
FROM other HAVING distance < 100 ");
while ($row6 = @mysql_fetch_assoc($rowoth)){
$myrowoth++;}
$numrows=$myrowc+$myrowm+$myrowt+$myrowo+$myrowcon+$myrowoth;
if ($numrows==1){
echo "Not Found";}
else{
$page=10;
$pages=intval($numrows/$pagesize);
if ($numrows%$pagesize)
$pages++;
if (isset($_GET['page'])){
$page=intval($_GET['page']);
}
else{
$page=1;
}
$offset=$pagesize*($page-1);
$query2 = "select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM concert HAVING distance < 100
union all
select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *

```

```

cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) AS distance FROM musical HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM theatre HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM opera HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM conference HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM other HAVING distance < 100 "
    ."limit $offset,$pagesize";
$result1=mysql_query($query2) or die("query failed:".mysql_error());
if ($answer=mysql_fetch_array($result1))
{
$i=0;
?>
<table border="0" width="80%">
<tr>
<td width="50%" bgcolor="#E0E0E0">
<p align="Left">Name</td>
<td width="50%" bgcolor="#E0E0E0">
<p align="Left">Location</td>
</tr>
<?php
do {
$i++;
?>
<tr>
<td width="50%"><a href="eventdetails.php?id=?php echo $answer["id"]?>&code=?php echo
$answer["code"]?>"><?php echo $answer["Displayname"];?></a></td>
<td width="50%"><a href="google.html?id=?php echo $answer["id"]?>&code=?php echo
$answer["code"]?>"><?php echo $answer["location"];?></a></td>
</tr>
<?php
}
while ($answer=mysql_fetch_array($result1));

```

```

echo "</table>";
}
echo "<div align='center'>".$pages." pages(".$page."/".$pages.")";
for ($i=1;$i<$page;$i++)
echo "<a href='nearby.php?page=".$i."&id=".$id."&code=".$code."'>[".$i.]</a>";
echo "[".$page.]";
for ($i=$page+1;$i<=$pages;$i++)
echo "<a href='nearby.php?page=".$i."&id=".$id."&code=".$code."'>[".$i.]</a>";
echo "</div>";
}
?>

```

We can use the same way in near1.php to find 200km—500km events.

11.8 Same day events

Date.php:

```

<?php
$id=$_GET['id'];
$code=$_GET['code'];
$connection=mysql_connect ("localhost", "root");
if (!$connection) {
    die('Not connected : ' . mysql_error());
}
$db_selected = mysql_select_db("eventsearch", $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}
$query = "select startdate,enddate from concert where id='$id' and code='$code'
        union all
        select startdate,enddate from musical where id='$id' and code='$code'
        union all
        select startdate,enddate from theatre where id='$id' and code='$code'
        union all
        select startdate,enddate from opera where id='$id' and code='$code'
        union all
        select startdate,enddate from conference where id='$id' and code='$code'
        union all
        select startdate,enddate from other where id='$id' and code='$code'";
$result=mysql_query($query) or die("query failed:".mysql_error());
$answer1=mysql_fetch_array($result);
$start=substr($answer1['startdate'],0,10);
$end=substr($answer1['enddate'],0,10);

```

```

$query1="startdate like '$start%' or enddate like '$end%'";
$rowc=mysql_query("select count(*) from concert where " . $query1);
$rowm=mysql_query("select count(*) from musical where " . $query1);
$rowt=mysql_query("select count(*) from theatre where " . $query1);
$rowo=mysql_query("select count(*) from opera where " . $query1);
$rowcon=mysql_query("select count(*) from conference where " . $query1);
$rowoth=mysql_query("select count(*) from other where " . $query1);
$myrowc=mysql_fetch_array($rowc);
$myrowm=mysql_fetch_array($rowm);
$myrowt=mysql_fetch_array($rowt);
$myrowo=mysql_fetch_array($rowo);
$myrowcon=mysql_fetch_array($rowcon);
$myrowoth=mysql_fetch_array($rowoth);
$numrows=$myrowc[0]+$myrowm[0]+$myrowt[0]+$myrowo[0]+$myrowcon[0]+$myrowoth[0]
;
if ($numrows<=1){
echo "Not Found";}
else{
$page=1;
$pagesize=10;
$pages=intval($numrows/$pagesize);
if ($numrows%$pagesize)
$page++;
if (isset($_GET['page'])){
$page=intval($_GET['page']);
}
else{
$page=1;
}
$offset=$pagesize*($page-1);
$query2 = "select id,code,Displayname,startdate,enddate from concert where startdate like '$start%' or enddate like '$end%'
union all
select id,code,Displayname,startdate,enddate from musical where startdate like '$start%'
or enddate like '$end%'
union all
select id,code,Displayname,startdate,enddate from theatre where startdate like '$start%'
or enddate like '$end%'
union all
select id,code,Displayname,startdate,enddate from opera where startdate like '$start%' or
enddate like '$end%'
union all
select id,code,Displayname,startdate,enddate from conference where startdate like '$start%' or enddate like '$end%'
union all

```

```

        select id,code,Displayname,startdate,enddate from other where startdate like '$start%' or
enddate like '$end%'
        ."limit $offset,$pagesize";
$result1=mysql_query($query2) or die("query failed:".mysql_error());
if ($answer=mysql_fetch_array($result1))
{
    $i=0;
    ?>
    <table border="0" width="80%">
    <tr>
    <td width="30%" bgcolor="#E0E0E0">
    <p align="Left">Name</td>
    <td width="30%" bgcolor="#E0E0E0">
    <p align="Left">StartDate</td>
    <td width="30%" bgcolor="#E0E0E0">
    <p align="Left">EndDate</td>
    </tr>
    <?php
    do {
    $i++;
    ?>
    <tr>
    <td width="30%"><a href="eventdetails.php?id=<?php echo $answer["id"]?>&code=<?php echo
    $answer["code"]?>"><?php echo $answer["Displayname"];?></a></td>
    <td width="30%"><?php echo $answer["startdate"];?></td>
    <td width="30%"><?php echo $answer["enddate"];?></td>
    </tr>
    <?php
    }
    while ($answer=mysql_fetch_array($result1));
    echo "</table>";
    }
    echo "<div align='center'>".$pages." pages( ".$page."/ ".$pages." )";
    for ($i=1;$i<=$page;$i++)
    echo "<a href='date.php?page=".$i."&id=".$id."&code=".$code."'>[".$i.]</a>";
    echo "[".$page."]";
    for ($i=$page+1;$i<=$pages;$i++)
    echo "<a href='date.php?page=".$i."&id=".$id."&code=".$code."'>[".$i.]</a>";
    echo "</div>";
    }
    ?>

```


11.9 Location of events on Google map

phpsqlajax_genxml2.php outputs XML:

```
<?php
$id=$_GET['id'];
$code=$_GET['code'];
function parseToXML($htmlStr)
{
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('"','&quot;',$xmlStr);
$xmlStr=str_replace("'",'&apos;',$xmlStr);
$xmlStr=str_replace("&","&amp;",$xmlStr);
return $xmlStr;
}
$connection=mysql_connect ("localhost", "root");
if (!$connection) {
    die('Not connected : ' . mysql_error());
}
$db_selected = mysql_select_db("eventsearch", $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}
$query = "select lat,lng from concert where id='$id' and code='$code'
        union all
        select lat,lng from musical where id='$id' and code='$code'
        union all
        select lat,lng from theatre where id='$id' and code='$code'
        union all
        select lat,lng from opera where id='$id' and code='$code'
        union all
        select lat,lng from conference where id='$id' and code='$code'
        union all
        select lat,lng from other where id='$id' and code='$code'";
$result=mysql_query($query) or die("query failed:".mysql_error());
$answer=mysql_fetch_array($result);
$lat=$answer['lat'];
$lng=$answer['lng'];
$query1 = "select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM concert HAVING distance < 100
        union all
        select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
```

```

cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) AS distance FROM musical HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM theatre HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM opera HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM conference HAVING distance < 100
    union all
    select id,code,location,Displayname,lat,lng,( 6371 * acos( cos( radians('$lat') ) *
cos( radians( lat ) ) * cos( radians( lng ) - radians('$lng') ) + sin( radians('$lat') ) *
sin( radians( lat ) ) ) ) AS distance FROM other HAVING distance < 100 ";
$result1=mysql_query($query1) or die("query failed:".mysql_error());
header("Content-type: text/xml");
// Start XML file, echo parent node
echo '<markers>';
// Iterate through the rows, printing XML nodes for each
while ($row = @mysql_fetch_assoc($result1)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ';
    echo 'name="'. parseToXML($row['Displayname']) . "' ";
    echo 'address="'. parseToXML($row['location']) . "' ";
    echo 'lat="'. $row['lat'] . "' ";
    echo 'lng="'. $row['lng'] . "' ";
    echo 'type="'. $row['code'] . "' ";
    echo 'id="'. $row['id'] . "' ";
    echo '>';
}
echo '</markers>';
?>

```

11.10 Download iCal file

Download.php:

```

<?php
$id=$_GET['id'];
$code=$_GET['code'];

```

```

if($code=="e1"){
$type="concert";}
if($code=="e2"){
$type="musical";}
if($code=="e3"){
$type="opera";}
if($code=="e4"){
$type="theatre";}
if($code=="e5"){
$type="conference";}
if($code=="e6"){
$type="other";}
$link=mysql_connect("localhost","root") or die(mysql_error());
mysql_select_db("eventsearch");
$query="select Displayname from concert where id='$id' and code='$code'
union all
select Displayname from musical where id='$id' and code='$code'
union all
select Displayname from opera where id='$id' and code='$code'
union all
select Displayname from theatre where id='$id' and code='$code'
union all
select Displayname from conference where id='$id' and code='$code'
union all
select Displayname from other where id='$id' and code='$code'";
$result=mysql_query($query);
$answer=mysql_fetch_array($result);
$displayname=$answer["Displayname"];
$file_name = "$type$displayname.ics";
$file_dir = "d:/upload/";
if (!file_exists($file_dir . $file_name)) {
echo "Can't find the file";
exit;
} else {
$file = fopen($file_dir . $file_name,"r");
Header("Content-type: application/octet-stream");
Header("Accept-Ranges: bytes");
Header("Accept-Length: ".filesize($file_dir . $file_name));
Header("Content-Disposition: attachment; filename=" . $file_name);
echo fread($file,filesize($file_dir . $file_name));
fclose($file);
}
?>

```

11.11 Send E-mail

Class.phpmailer.php:

```
<?php
class phpmailer
{
    var $Priority          = 3;
    var $CharSet          = "iso-8859-1";
    var $ContentType     = "text/plain";
    var $Encoding         = "8bit";
    var $ErrorInfo       = "";
    var $From            = "root@localhost";
    var $FromName        = "Root User";
    var $Sender          = "";
    var $Subject         = "";
    var $Body            = "";
    var $AltBody         = "";
    var $WordWrap        = 0;
    var $Mailer          = "mail";
    var $Sendmail        = "/usr/sbin/sendmail";
    var $UseMSMailHeaders = false;
    var $PluginDir       = "";
    var $Version         = "1.54";
    var $ConfirmReadingTo = "";
    var $LE              = "\n";
    var $Host            = "localhost";
    var $Port            = 25;
    var $Helo           = "localhost.localdomain";
    var $SMTPAuth        = false;
    var $Password        = "";
    var $Timeout         = 10;
    var $SMTPDebug       = false;
    var $to              = array();
    var $cc              = array();
    var $bcc             = array();
    var $ReplyTo        = array();
    var $attachment     = array();
    var $CustomHeader    = array();
    var $message_type    = "";
    var $boundary        = array();
    function IsHTML($bool) {
        if($bool == true)
            $this->ContentType = "text/html";
    }
}
```

```

        else
            $this->ContentType = "text/plain";
    }
    function IsSMTP() {
        $this->Mailer = "smtp";
    }
    function IsMail() {
        $this->Mailer = "mail";
    }
    function IsSendmail() {
        $this->Mailer = "sendmail";
    }
    function IsQmail() {
        //$this->Sendmail = "/var/qmail/bin/qmail-inject";
        $this->Sendmail = "/var/qmail/bin/sendmail";
        $this->Mailer = "sendmail";
    }
    function AddAddress($address, $name = "") {
        $cur = count($this->to);
        $this->to[$cur][0] = trim($address);
        $this->to[$cur][1] = $name;
    }
    function AddCC($address, $name = "") {
        $cur = count($this->cc);
        $this->cc[$cur][0] = trim($address);
        $this->cc[$cur][1] = $name;
    }
    function AddBCC($address, $name = "") {
        $cur = count($this->bcc);
        $this->bcc[$cur][0] = trim($address);
        $this->bcc[$cur][1] = $name;
    }
    function AddReplyTo($address, $name = "") {
        $cur = count($this->ReplyTo);
        $this->ReplyTo[$cur][0] = trim($address);
        $this->ReplyTo[$cur][1] = $name;
    }
    function Send() {
        $header = "";
        $body = "";
        if((count($this->to) + count($this->cc) + count($this->bcc)) < 1)
        {
            $this->error_handler("You must provide at least one recipient email address");
            return false;
        }
    }

```

```

}
if(!empty($this->AltBody))
    $this->ContentType = "multipart/alternative";
$header = $this->received();
$header .= sprintf("Date: %s%s", $this->rfc_date(), $this->LE);
$header .= $this->create_header();
if(!$body = $this->create_body())
    return false;
if($this->Mailer == "sendmail")
{
    if(!$this->sendmail_send($header, $body))
        return false;
}
elseif($this->Mailer == "mail")
{
    if(!$this->mail_send($header, $body))
        return false;
}
elseif($this->Mailer == "smtp")
{
    if(!$this->smtp_send($header, $body))
        return false;
}
else
{
    $this->error_handler(sprintf("%s mailer is not supported", $this->Mailer));
    return false;
}
return true;
}
function SendToQueue($queue_path, $send_time = 0) {
    $message = array();
    $header = "";
    $body = "";

    if($send_time == 0)
        $send_time = time();
    if(!is_dir($queue_path))
    {
        $this->error_handler("The supplied queue directory does not exist");
        return false;
    }
    if((count($this->to) + count($this->cc) + count($this->bcc)) < 1)
    {

```

```

        $this->error_handler("You must provide at least one recipient email address");
        return false;
    }
    if(!empty($this->AltBody))
        $this->ContentType = "multipart/alternative";
    $header = $this->create_header();
    if(!$body = $this->create_body())
        return false;
    mt_srand(time());
    $msg_id = md5(uniqid(mt_rand()));

    $fp = fopen($queue_path . $msg_id . ".pqm", "wb");
    if(!$fp)
    {
        $this->error_handler(sprintf("Could not write to %s directory", $queue_path));
        return false;
    }
    $message[] = sprintf("----START PQM HEADER----%s", $this->LE);
    $message[] = sprintf("SendTime: %s%s", $send_time, $this->LE);
    $message[] = sprintf("Mailer: %s%s", $this->Mailer, $this->LE);
    if($this->Mailer == "sendmail")
    {
        $message[] = sprintf("Sendmail: %s%s", $this->Sendmail, $this->LE);
        $message[] = sprintf("Sender: %s%s", $this->Sender, $this->LE);
    }
    elseif($this->Mailer == "mail")
    {
        $message[] = sprintf("Sender: %s%s", $this->Sender, $this->LE);
        $message[] = sprintf("Subject: %s%s", $this->Subject, $this->LE);
        $message[] = sprintf("to: %s%s", $this->addr_list($this->to), $this->LE);
    }
    elseif($this->Mailer == "smtp")
    {
        $message[] = sprintf("Host: %s%s", $this->Host, $this->LE);
        $message[] = sprintf("Port: %d%s", $this->Port, $this->LE);
        $message[] = sprintf("Helo: %s%s", $this->Helo, $this->LE);
        $message[] = sprintf("Timeout: %d%s", $this->Timeout, $this->LE);
        if($this->SMTPAuth)
            $auth_no = 1;
        else
            $auth_no = 0;
        $message[] = sprintf("SMTPAuth: %d%s", $auth_no, $this->LE);
        $message[] = sprintf("Username: %s%s", $this->Username, $this->LE);
        $message[] = sprintf("Password: %s%s", $this->Password, $this->LE);
    }

```

```

$message[] = sprintf("From: %s%s", $this->From, $this->LE);

$message[] = sprintf("to: %s%s", $this->addr_list($this->to), $this->LE);
$message[] = sprintf("cc: %s%s", $this->addr_list($this->cc), $this->LE);
$message[] = sprintf("bcc: %s%s", $this->addr_list($this->bcc), $this->LE);
}
else
{
    $this->error_handler(sprintf("%s mailer is not supported", $this->Mailer));
    return false;
}
$message[] = sprintf("----END PQM HEADER----%s", $this->LE); // end of pqm
header
$message[] = $header;
$message[] = $body;
fwrite($fp, join("", $message));

return ($msg_id . ".pqm");
}
function sendmail_send($header, $body) {
    if ($this->Sender != "")
        $sendmail = sprintf("%s -oi -f %s -t", $this->Sendmail, $this->Sender);
    else
        $sendmail = sprintf("%s -oi -t", $this->Sendmail);
    if(!@$mail = popen($sendmail, "w"))
    {
        $this->error_handler(sprintf("Could not execute %s", $this->Sendmail));
        return false;
    }
    fputs($mail, $header);
    fputs($mail, $body);

    $result = pclose($mail) >> 8 & 0xFF;
    if($result != 0)
    {
        $this->error_handler(sprintf("Could not execute %s", $this->Sendmail));
        return false;
    }
    return true;
}
function mail_send($header, $body) {
    //$to = substr($this->addr_append("To", $this->to), 4, -2);
    $to = $this->to[0][0]; // no extra comma
    for($i = 1; $i < count($this->to); $i++)

```



```

        $to .= sprintf(",%s", $this->to[$i][0]);
    if ($this->Sender != "" && PHP_VERSION >= "4.0")
    {
        $old_from = ini_get("sendmail_from");
        ini_set("sendmail_from", $this->Sender);
    }
    if ($this->Sender != "" && PHP_VERSION >= "4.0.5")
    {
        $params = sprintf("-oi -f %s", $this->Sender);
        $rt = @mail($to, $this->Subject, $body, $header, $params);
    }
    else
    {
        $rt = @mail($to, $this->Subject, $body, $header);
    }
    if (isset($old_from))
        ini_set("sendmail_from", $old_from);
    if (!$rt)
    {
        $this->error_handler("Could not instantiate mail()");
        return false;
    }
    return true;
}
function smtp_send($header, $body) {
    include_once($this->PluginDir . "class.smtp.php");
    $smtp = new SMTP;
    $smtp->do_debug = $this->SMTPDebug;
    $hosts = explode(";", $this->Host);
    $index = 0;
    $connection = false;
    $smtp_from = "";
    $bad_rcpt = array();
    $e = "";
    while($index < count($hosts) && $connection == false)
    {
        if(strpos($hosts[$index], ":"))
            list($host, $port) = explode(":", $hosts[$index]);
        else
        {
            $host = $hosts[$index];
            $port = $this->Port;
        }
        if($smtp->Connect($host, $port, $this->Timeout))

```

```

        $connection = true;
        //printf("%s host could not connect<br>", $hosts[$index]); //debug only
        $index++;
    }
    if(!$connection)
    {
        $this->error_handler("SMTP Error: could not connect to SMTP host server(s)");
        return false;
    }
    $smtp->Hello($this->Helo);
    if($this->SMTPAuth)
    {
        if(!$smtp->Authenticate($this->Username, $this->Password))
        {
            $this->error_handler("SMTP Error: Could not authenticate");
            return false;
        }
    }
    if ($this->Sender == "")
        $smtp_from = $this->From;
    else
        $smtp_from = $this->Sender;
    if(!$smtp->Mail(sprintf("<%s>", $smtp_from)))
    {
        $e = sprintf("SMTP Error: From address [%s] failed", $smtp_from);
        $this->error_handler($e);
        return false;
    }
    for($i = 0; $i < count($this->to); $i++)
    {
        if(!$smtp->Recipient(sprintf("<%s>", $this->to[$i][0])))
            $bad_rcpt[] = $this->to[$i][0];
    }
    for($i = 0; $i < count($this->cc); $i++)
    {
        if(!$smtp->Recipient(sprintf("<%s>", $this->cc[$i][0])))
            $bad_rcpt[] = $this->cc[$i][0];
    }
    for($i = 0; $i < count($this->bcc); $i++)
    {
        if(!$smtp->Recipient(sprintf("<%s>", $this->bcc[$i][0])))
            $bad_rcpt[] = $this->bcc[$i][0];
    }
    if(count($bad_rcpt) > 0)

```

```

    {
        for($i = 0; $i < count($bad_rcpt); $i++)
        {
            if($i != 0)
                $e .= ", ";
            $e .= $bad_rcpt[$i];
        }
        $e = sprintf("SMTP Error: The following recipients failed [%s]", $e);
        $this->error_handler($e);
        return false;
    }
    if(!$smtp->Data(sprintf("%s%s", $header, $body)))
    {
        $this->error_handler("SMTP Error: Data not accepted");
        return false;
    }
    $smtp->Quit();

    return true;
}

function addr_append($type, $addr) {
    $addr_str = $type . ": ";
    $addr_str .= $this->addr_format($addr[0]);
    if(count($addr) > 1)
    {
        for($i = 1; $i < count($addr); $i++)
        {
            $addr_str .= sprintf(", %s", $this->addr_format($addr[$i]));
        }
        $addr_str .= $this->LE;
    }
    else
        $addr_str .= $this->LE;
    return($addr_str);
}

function addr_list($list_array) {
    $addr_list = "";
    for($i = 0; $i < count($list_array); $i++)
    {
        if($i > 0)
            $addr_list .= ";";
        $addr_list .= $list_array[$i][0];
    }
    return $addr_list;
}

```

```

}
function addr_format($addr) {
    if(empty($addr[1]))
        $formatted = $addr[0];
    else
        $formatted = sprintf("%s" <%s>', addslashes($addr[1]), $addr[0]);

    return $formatted;
}
function word_wrap($message, $length, $qp_mode = false) {
    if ($qp_mode)
        $soft_break = sprintf(" =%s", $this->LE);
    else
        $soft_break = $this->LE;
    $message = $this->fix_eol($message);
    if (substr($message, -1) == $this->LE)
        $message = substr($message, 0, -1);
    $line = explode($this->LE, $message);
    $message = "";
    for ($i=0 ;$i < count($line); $i++)
    {
        $line_part = explode(" ", $line[$i]);
        $buf = "";
        for ($e = 0; $e < count($line_part); $e++)
        {
            $word = $line_part[$e];
            if ($qp_mode and (strlen($word) > $length))
            {
                $space_left = $length - strlen($buf) - 1;
                if ($e != 0)
                {
                    if ($space_left > 20)
                    {
                        $len = $space_left;
                        if (substr($word, $len - 1, 1) == "=")
                            $len--;
                        elseif (substr($word, $len - 2, 1) == "=")
                            $len -= 2;
                        $part = substr($word, 0, $len);
                        $word = substr($word, $len);
                        $buf .= " " . $part;
                        $message .= $buf . sprintf(" =%s", $this->LE);
                    }
                }
            }
            else

```

```

        {
            $message .= $buf . $soft_break;
        }
        $buf = "";
    }
    while (strlen($word) > 0)
    {
        $len = $length;
        if (substr($word, $len - 1, 1) == "=")
            $len--;
        elseif (substr($word, $len - 2, 1) == "=")
            $len -= 2;
        $part = substr($word, 0, $len);
        $word = substr($word, $len);

        if (strlen($word) > 0)
            $message .= $part . sprintf("=%s", $this->LE);
        else
            $buf = $part;
    }
}
else
{
    $buf_o = $buf;
    if ($e == 0)
        $buf .= $word;
    else
        $buf .= " " . $word;
    if (strlen($buf) > $length and $buf_o != "")
    {
        $message .= $buf_o . $soft_break;
        $buf = $word;
    }
}
}
$message .= $buf . $this->LE;
}
return ($message);
}
function create_header() {
    $header = array();
    $uniq_id = md5(uniqid(time()));
    $this->boundary[1] = "b1_" . $uniq_id;
    $this->boundary[2] = "b2_" . $uniq_id;
}

```

```

if(($this->Mailer != "mail") && (count($this->to) > 0))
    $header[] = $this->addr_append("To", $this->to);
$header[] = sprintf("From: \"%s\" <%s>%s", addslashes($this->FromName),
                    trim($this->From), $this->LE);
if(count($this->cc) > 0)
    $header[] = $this->addr_append("Cc", $this->cc);
if(((($this->Mailer == "sendmail") || ($this->Mailer == "mail"))) && (count($this->bcc) >
0))
    $header[] = $this->addr_append("Bcc", $this->bcc);
if(count($this->ReplyTo) > 0)
    $header[] = $this->addr_append("Reply-to", $this->ReplyTo);
if($this->Mailer != "mail")
    $header[] = sprintf("Subject: %s%s", trim($this->Subject), $this->LE);
$header[] = sprintf("X-Priority: %d%s", $this->Priority, $this->LE);
$header[] = sprintf("X-Mailer: phpmailer [version %s]%s", $this->Version, $this->LE);
$header[] = sprintf("Return-Path: %s%s", trim($this->From), $this->LE);
if($this->ConfirmReadingTo != "")
    $header[] = sprintf("Disposition-Notification-To: <%s>%s",
                        trim($this->ConfirmReadingTo), $this->LE);
for($index = 0; $index < count($this->CustomHeader); $index++)
    $header[] = sprintf("%s%s", $this->CustomHeader[$index], $this->LE);
if($this->UseMSMailHeaders)
    $header[] = $this->AddMSMailHeaders();
$header[] = sprintf("MIME-Version: 1.0%s", $this->LE);
if(count($this->attachment) < 1 && strlen($this->AltBody) < 1)
    $this->message_type = "plain";
else
{
    if(count($this->attachment) > 0)
        $this->message_type = "attachments";
    if(strlen($this->AltBody) > 0 && count($this->attachment) < 1)
        $this->message_type = "alt";
    if(strlen($this->AltBody) > 0 && count($this->attachment) > 0)
        $this->message_type = "alt_attachments";
}
switch($this->message_type)
{
    case "plain":
        $header[] = sprintf("Content-Transfer-Encoding: %s%s",
                            $this->Encoding, $this->LE);
        $header[] = sprintf("Content-Type: %s; charset = \"%s\"",
                            $this->ContentType, $this->CharSet);
        break;
    case "attachments":

```

```

        case "alt_attachments":
            if($this->EmbeddedImageCount() > 0)
            {
                $header[] = sprintf("Content-Type:
%s;%s\ttype=\"text/html\";%s\tboundary=\"%s\"%s",
                                "multipart/related", $this->LE, $this->LE,
                                $this->boundary[1], $this->LE);
            }
            else
            {
                $header[] = sprintf("Content-Type: %s;%s",
                                "multipart/mixed", $this->LE);
                $header[] = sprintf("\tboundary=\"%s\"%s", $this->boundary[1],
$this->LE);
            }
            break;
        case "alt":
            $header[] = sprintf("Content-Type: %s;%s",
                                "multipart/alternative", $this->LE);
            $header[] = sprintf("\tboundary=\"%s\"%s", $this->boundary[1], $this->LE);
            break;
    }
    if($this->Mailer != "mail")
        $header[] = $this->LE.$this->LE;
    return(join("", $header));
}
function create_body() {
    $body = array();
    if($this->WordWrap > 0)
        $this->Body = $this->word_wrap($this->Body, $this->WordWrap);
    switch($this->message_type)
    {
        case "alt":
            $bdry = new Boundary($this->boundary[1]);
            $bdry->CharSet = $this->CharSet;
            $bdry->Encoding = $this->Encoding;
            $body[] = $bdry->GetSource();
            $body[] = sprintf("%s%s", $this->AltBody, $this->LE.$this->LE);
            $bdry = new Boundary($this->boundary[1]);
            $bdry->CharSet = $this->CharSet;
            $bdry->ContentType = "text/html";
            $bdry->Encoding = $this->Encoding;
            $body[] = $bdry->GetSource();
            $body[] = sprintf("%s%s", $this->Body, $this->LE.$this->LE);

```

```

$body[] = sprintf("%s--%s--%s", $this->LE,
                  $this->boundary[1], $this->LE.$this->LE);

break;
case "plain":
    $body[] = $this->Body;
    break;
case "attachments":
    $bdry = new Boundary($this->boundary[1]);
    $bdry->CharSet = $this->CharSet;
    $bdry->ContentType = $this->ContentType;
    $bdry->Encoding = $this->Encoding;
    $body[] = sprintf("%s%s%s%s", $bdry->GetSource(false), $this->LE,
                      $this->Body, $this->LE);
    if(!$body[] = $this->attach_all())
        return false;
    break;
case "alt_attachments":
    $body[] = sprintf("--%s%s", $this->boundary[1], $this->LE);
    $body[] = sprintf("Content-Type: %s;%s" .
                      "\tboundary=\"%s\"%s",
                      "multipart/alternative", $this->LE,
                      $this->boundary[2], $this->LE.$this->LE);

    $bdry = new Boundary($this->boundary[2]);
    $bdry->CharSet = $this->CharSet;
    $bdry->ContentType = "text/plain";
    $bdry->Encoding = $this->Encoding;
    $body[] = $bdry->GetSource() . $this->LE;
    $body[] = sprintf("%s%s", $this->AltBody, $this->LE.$this->LE);
    $bdry = new Boundary($this->boundary[2]);
    $bdry->CharSet = $this->CharSet;
    $bdry->ContentType = "text/html";
    $bdry->Encoding = $this->Encoding;
    $body[] = $bdry->GetSource() . $this->LE;

    $body[] = sprintf("%s%s", $this->Body, $this->LE.$this->LE);

    $body[] = sprintf("%s--%s--%s", $this->LE,
                      $this->boundary[2], $this->LE.$this->LE);
    if(!$body[] = $this->attach_all())
        return false;
    break;
}
$sBody = join("", $body);
$sBody = $this->encode_string($sBody, $this->Encoding);

```



```

        return $sBody;
    }
    function AddAttachment($path, $name = "", $encoding = "base64", $type =
"application/octet-stream") {
        if(!@is_file($path))
        {
            $this->error_handler(sprintf("Could not access [%s] file", $path));
            return false;
        }
        $filename = basename($path);
        if($name == "")
            $name = $filename;
        $cur = count($this->attachment);
        $this->attachment[$cur][0] = $path;
        $this->attachment[$cur][1] = $filename;
        $this->attachment[$cur][2] = $name;
        $this->attachment[$cur][3] = $encoding;
        $this->attachment[$cur][4] = $type;
        $this->attachment[$cur][5] = false; // isStringAttachment
        $this->attachment[$cur][6] = "attachment";
        $this->attachment[$cur][7] = 0;
        return true;
    }
    function attach_all() {
        $mime = array();
        for($i = 0; $i < count($this->attachment); $i++)
        {
            $isString = $this->attachment[$i][5];
            if ($isString)
            {
                $string = $this->attachment[$i][0];
            }
            else
            {
                $path = $this->attachment[$i][0];
            }
            $filename = $this->attachment[$i][1];
            $name = $this->attachment[$i][2];
            $encoding = $this->attachment[$i][3];
            $type = $this->attachment[$i][4];
            $disposition = $this->attachment[$i][6];
            $cid = $this->attachment[$i][7];
            $mime[] = sprintf("--%s%s", $this->boundary[1], $this->LE);
            $mime[] = sprintf("Content-Type: %s; name=\"%s\"%s", $type, $name,

```

```

$this->LE);
    $mime[] = sprintf("Content-Transfer-Encoding: %s%s", $encoding, $this->LE);
    if($disposition == "inline")
        $mime[] = sprintf("Content-ID: <%s>%s", $cid, $this->LE);
    else
        $mime[] = sprintf("Content-ID: <%s>%s", $name, $this->LE);
    $mime[] = sprintf("Content-Disposition: %s; filename=\"%s\"%s",
        $disposition, $name, $this->LE.$this->LE);
    if($isString)
    {
        if(!$mime[] = sprintf("%s%s", $this->encode_string($string, $encoding),
            $this->LE.$this->LE))
            return false;
    }
    else
    {
        if(!$mime[] = sprintf("%s%s", $this->encode_file($path, $encoding),
            $this->LE.$this->LE))
            return false;
        $mime[] = sprintf("--%s--%s", $this->boundary[1], $this->LE);
    }
}
return(join("", $mime));
}
function encode_file ($path, $encoding = "base64") {
    if(!@$fd = fopen($path, "rb"))
    {
        $this->error_handler(sprintf("File Error: Could not open file %s", $path));
        return false;
    }
    $file = fread($fd, filesize($path));
    $encoded = $this->encode_string($file, $encoding);
    fclose($fd);
    return($encoded);
}
function encode_string ($str, $encoding = "base64") {
    switch(strtolower($encoding)) {
        case "base64":
            $encoded = chunk_split(base64_encode($str));
            break;
        case "7bit":
        case "8bit":
            $encoded = $this->fix_eol($str);
            if (substr($encoded, -2) != $this->LE)

```

```

        $encoded .= $this->LE;
        break;
    case "binary":
        $encoded = $str;
        break;
    case "quoted-printable":
        $encoded = $this->encode_qp($str);
        break;
    default:
        $this->error_handler(sprintf("Unknown encoding: %s", $encoding));
        return false;
    }
    return($encoded);
}
function encode_qp ($str) {
    $encoded = $this->fix_eol($str);
    if (substr($encoded, -2) != $this->LE)
        $encoded .= $this->LE;
    $encoded = preg_replace("/([\001-\010\013\014\016-\037\075\177-\377])/e",
        "'='.sprintf('%02X', ord('\1'))", $encoded);
    $encoded = preg_replace("/([\011\040]).$this->LE."/e",
        "'='.sprintf('%02X', ord('\1')).".$this->LE.'"', $encoded);
    $encoded = $this->word_wrap($encoded, 74, true);
    return $encoded;
}
function AddStringAttachment($string, $filename, $encoding = "base64", $type =
"application/octet-stream") {
    $cur = count($this->attachment);
    $this->attachment[$cur][0] = $string;
    $this->attachment[$cur][1] = $filename;
    $this->attachment[$cur][2] = $filename;
    $this->attachment[$cur][3] = $encoding;
    $this->attachment[$cur][4] = $type;
    $this->attachment[$cur][5] = true; // isString
    $this->attachment[$cur][6] = "attachment";
    $this->attachment[$cur][7] = 0;
}
function AddEmbeddedImage($path, $cid, $name = "", $encoding = "base64", $type =
"application/octet-stream") {
    if(!@is_file($path))
    {
        $this->error_handler(sprintf("Could not access [%s] file", $path));
        return false;
    }
}

```

```

$filename = basename($path);
if($name == "")
    $name = $filename;
$scur = count($this->attachment);
$this->attachment[$scur][0] = $path;
$this->attachment[$scur][1] = $filename;
$this->attachment[$scur][2] = $name;
$this->attachment[$scur][3] = $encoding;
$this->attachment[$scur][4] = $type;
$this->attachment[$scur][5] = false; // isStringAttachment
$this->attachment[$scur][6] = "inline";
$this->attachment[$scur][7] = $cid;
return true;
}
function EmbeddedImageCount() {
    $ret = 0;
    for($i = 0; $i < count($this->attachment); $i++)
    {
        if($this->attachment[$i][6] == "inline")
            $ret++;
    }
    return $ret;
}
function ClearAddresses() {
    $this->to = array();
}
function ClearCCs() {
    $this->cc = array();
}
function ClearBCCs() {
    $this->bcc = array();
}
function ClearReplyTos() {
    $this->ReplyTo = array();
}
function ClearAllRecipients() {
    $this->to = array();
    $this->cc = array();
    $this->bcc = array();
}
function ClearAttachments() {
    $this->attachment = array();
}
function ClearCustomHeaders() {

```

```

        $this->CustomHeader = array();
    }
    function error_handler($msg) {
        $this->ErrorInfo = $msg;
    }
    function rfc_date() {
        $tz = date("Z");
        $tzs = ($tz < 0) ? "-" : "+";
        $tz = abs($tz);
        $tz = ($tz/3600)*100 + ($tz%3600)/60;
        $date = sprintf("%s %s%04d", date("D, j M Y H:i:s"), $tzs, $tz);
        return $date;
    }
    function received() {
        $str = sprintf("Received: from phpmailer ([%s]) by %s " .
            "with HTTP;%s\t %s%s",
            $this->get_server_var("REMOTE_ADDR"),
            $this->get_server_var("SERVER_NAME"),
            $this->LE,
            $this->rfc_date(),
            $this->LE);

        return $str;
    }
    function get_server_var($varName) {
        global $HTTP_SERVER_VARS;
        global $HTTP_ENV_VARS;
        if(!isset($_SERVER))
        {
            $_SERVER = $HTTP_SERVER_VARS;
            if(!isset($_SERVER["REMOTE_ADDR"]))
                $_SERVER = $HTTP_ENV_VARS; // must be Apache
        }
        if(isset($_SERVER[$varName]))
            return $_SERVER[$varName];
        else
            return "";
    }
    function fix_eol($str) {
        $str = str_replace("\r\n", "\n", $str);
        $str = str_replace("\r", "\n", $str);
        $str = str_replace("\n", $this->LE, $str);
        return $str;
    }
}

```

```

function AddCustomHeader($custom_header) {
    $this->CustomHeader[] = $custom_header;
}
function AddMSMailHeaders() {
    $MSHeader = "";
    if($this->Priority == 1)
        $MSPriority = "High";
    elseif($this->Priority == 5)
        $MSPriority = "Low";
    else
        $MSPriority = "Medium";
    $MSHeader .= sprintf("X-MSMail-Priority: %s%s", $MSPriority, $this->LE);
    $MSHeader .= sprintf("Importance: %s%s", $MSPriority, $this->LE);
    return($MSHeader);
}
}
class Boundary
{
    var $ID = 0;
    var $ContentType = "text/plain";
    var $Encoding = "";
    var $Disposition = "";
    var $FileName = "";
    var $CharSet = "";
    var $LE          = "\n";
    function Boundary($boundary_id) {
        $this->ID = $boundary_id;
    }
    function GetSource($bLineEnding = true) {
        $ret = array();
        $mime[] = sprintf("--%s%s", $this->ID, $this->LE);
        $mime[] = sprintf("Content-Type: %s; charset = \"%s\"%s",
            $this->ContentType, $this->CharSet, $this->LE);
        if(strlen($this->Disposition) > 0)
        {
            $mime[] = sprintf("Content-Disposition: %s;",
                if(strlen($this->FileName) > 0)
                    $mime[] = sprintf("filename=\"%s\"", $this->$this->FileName);
        }
        if($bLineEnding)
            $mime[] = $this->LE;
        return join("", $mime);
    }
}
}

```

?>

Class.smtp.php:

```
<?php
class SMTP {
    var $SMTP_PORT = 25;
    var $CRLF = "\r\n";
    var $smtp_conn;
    var $error;
    var $helo_rply;
    var $do_debug;
    function SMTP() {
        $this->smtp_conn = 0;
        $this->error = null;
        $this->helo_rply = null;

        $this->do_debug = 0;
    }
    function Connect($host,$port=0,$stval=30) {
        $this->error = null;
        if($this->connected()) {
            $this->error =
                array("error" => "Already connected to a server");
            return false;
        }
        if(empty($port)) {
            $port = $this->SMTP_PORT;
        }
        $this->smtp_conn = fsockopen($host,
                                    $port,
                                    $errno,
                                    $errstr,
                                    $stval);
        if(empty($this->smtp_conn)) {
            $this->error = array("error" => "Failed to connect to server",
                                "errno" => $errno,
                                "errstr" => $errstr);
            if($this->do_debug >= 1) {
                echo "SMTP -> ERROR: " . $this->error["error"] .
                    ": $errstr ($errno)" . $this->CRLF;
            }
            return false;
        }
    }
}
```

```

if(substr(PHP_OS, 0, 3) != "WIN")
    socket_set_timeout($this->smtp_conn, 1, 0);
$announce = $this->get_lines();
if($this->do_debug >= 2) {
    echo "SMTP -> FROM SERVER:" . $this->CRLF . $announce;
}
return true;
}
function Authenticate($username, $password) {
    fputs($this->smtp_conn, "AUTH LOGIN" . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply, 0, 3);
    if($code != 334) {
        $this->error =
            array("error" => "AUTH not accepted from server",
                "smtp_code" => $code,
                "smtp_msg" => substr($rply, 4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
    fputs($this->smtp_conn, base64_encode($username) . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply, 0, 3);
    if($code != 334) {
        $this->error =
            array("error" => "Username not accepted from server",
                "smtp_code" => $code,
                "smtp_msg" => substr($rply, 4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
    fputs($this->smtp_conn, base64_encode($password) . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply, 0, 3);
    if($code != 235) {
        $this->error =
            array("error" => "Password not accepted from server",
                "smtp_code" => $code,

```



```

        "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": " . $rply . $this->CRLF;
    }
    return false;
}
return true;
}
function Connected() {
    if(!empty($this->smtp_conn)) {
        $sock_status = socket_get_status($this->smtp_conn);
        if($sock_status["eof"]) {
            if($this->do_debug >= 1) {
                echo "SMTP -> NOTICE:" . $this->CRLF .
                    "EOF caught while checking if connected";
            }
            $this->Close();
            return false;
        }
        return true;
    }
    return false;
}
function Close() {
    $this->error = null; # so there is no confusion
    $this->helo_rply = null;
    if(!empty($this->smtp_conn)) {
        # close the connection and cleanup
        fclose($this->smtp_conn);
        $this->smtp_conn = 0;
    }
}
function Data($msg_data) {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Data() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"DATA" . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
}

```

```

}
if($code != 354) {
    $this->error =
        array("error" => "DATA command not accepted from server",
            "smtp_code" => $code,
            "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": " . $rply . $this->CRLF;
    }
    return false;
}
$field = substr($lines[0],0,strpos($lines[0],":"));
$in_headers = false;
if(!empty($field) && !strstr($field," ")) {
    $in_headers = true;
}
$max_line_length = 998;
while(list($line) = @each($lines)) {
    $lines_out = null;
    if($line == "" && $in_headers) {
        $in_headers = false;
    }
    while(strlen($line) > $max_line_length) {
        $pos = strrpos(substr($line,0,$max_line_length)," ");
        $lines_out[] = substr($line,0,$pos);
        $line = substr($line,$pos + 1);
        if($in_headers) {
            $line = "\t" . $line;
        }
    }
    $lines_out[] = $line;
    while(list($line_out) = @each($lines_out)) {
        if(strlen($line_out) > 0)
        {
            if(substr($line_out, 0, 1) == ".") {
                $line_out = "." . $line_out;
            }
        }
        fputs($this->smtp_conn,$line_out . $this->CRLF);
    }
}
fputs($this->smtp_conn, $this->CRLF . "." . $this->CRLF);
$rply = $this->get_lines();

```

```

$code = substr($rply,0,3);
if($this->do_debug >= 2) {
    echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
}
if($code != 250) {
    $this->error =
        array("error" => "DATA not accepted from server",
              "smtp_code" => $code,
              "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": " . $rply . $this->CRLF;
    }
    return false;
}
return true;
}
function Expand($name) {
    $this->error = null;

    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Expand() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"EXPN " . $name . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250) {
        $this->error =
            array("error" => "EXPN not accepted from server",
                  "smtp_code" => $code,
                  "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
    $entries = explode($this->CRLF,$rply);
    while(list($i) = @each($entries)) {

```

```

        $list[] = substr($l,4);
    }
    return $rval;
}
function Hello($host="") {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Hello() without being connected");
        return false;
    }
    if(empty($host)) {
        $host = "localhost";
    }
    fputs($this->smtp_conn,"HELO " . $host . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER: " . $this->CRLF . $rply;
    }
    if($code != 250) {
        $this->error =
            array("error" => "HELO not accepted from server",
                "smtp_code" => $code,
                "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
    $this->helo_rply = $rply;
    return true;
}
function Help($keyword="") {
    $this->error = null; # to avoid confusion
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Help() without being connected");
        return false;
    }
    $extra = "";
    if(!empty($keyword)) {
        $extra = " " . $keyword;
    }
}

```

```

}
fputs($this->smtp_conn,"HELP" . $extra . $this->CRLF);
$rply = $this->get_lines();
$code = substr($rply,0,3);
if($this->do_debug >= 2) {
    echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
}
if($code != 211 && $code != 214) {
    $this->error =
        array("error" => "HELP not accepted from server",
              "smtp_code" => $code,
              "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            ": " . $rply . $this->CRLF;
    }
    return false;
}
return $rply;
}
function Mail($from) {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Mail() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"MAIL FROM:" . $from . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250) {
        $this->error =
            array("error" => "MAIL not accepted from server",
                  "smtp_code" => $code,
                  "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
}

```

```

        return true;
    }
function Noop() {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Noop() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"NOOP" . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250) {
        $this->error =
            array("error" => "NOOP not accepted from server",
                "smtp_code" => $code,
                "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
    return true;
}
function Quit($close_on_error=true) {
    $this->error = null; # so there is no confusion
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Quit() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"quit" . $this->CRLF);
    $byemsg = $this->get_lines();
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $byemsg;
    }
    $rval = true;
    $e = null;
    $code = substr($byemsg,0,3);
    if($code != 221) {

```

```

        $e = array("error" => "SMTP server rejected quit command",
                  "smtp_code" => $code,
                  "smtp_rply" => substr($byemsg,4));
        $rval = false;
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $e["error"] . ": " .
                $byemsg . $this->CRLF;
        }
    }
    if(empty($e) || $close_on_error) {
        $this->Close();
    }
    return $rval;
}

function Recipient($to) {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Recipient() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"RCPT TO:" . $to . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250 && $code != 251) {
        $this->error =
            array("error" => "RCPT not accepted from server",
                  "smtp_code" => $code,
                  "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
    return true;
}

function Reset() {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(

```

```

        "error" => "Called Reset() without being connected");
    return false;
}
fputs($this->smtp_conn,"RSET" . $this->CRLF);
$rply = $this->get_lines();
$code = substr($rply,0,3);
if($this->do_debug >= 2) {
    echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
}
if($code != 250) {
    $this->error =
        array("error" => "RSET failed",
            "smtp_code" => $code,
            "smtp_msg" => substr($rply,4));
    if($this->do_debug >= 1) {
        echo "SMTP -> ERROR: " . $this->error["error"] .
            " : " . $rply . $this->CRLF;
    }
    return false;
}
return true;
}

function Send($from) {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Send() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"SEND FROM:" . $from . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250) {
        $this->error =
            array("error" => "SEND not accepted from server",
                "smtp_code" => $code,
                "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                " : " . $rply . $this->CRLF;
        }
    }
}

```



```

        return false;
    }
    return true;
}
function SendAndMail($from) {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called SendAndMail() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"SAML FROM:" . $from . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250) {
        $this->error =
            array("error" => "SAML not accepted from server",
                "smtp_code" => $code,
                "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
    return true;
}
function SendOrMail($from) {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called SendOrMail() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"SOML FROM:" . $from . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250) {

```

```

$this->error =
    array("error" => "SOML not accepted from server",
          "smtp_code" => $code,
          "smtp_msg" => substr($rply,4));
if($this->do_debug >= 1) {
    echo "SMTP -> ERROR: " . $this->error["error"] .
        ": " . $rply . $this->CRLF;
}
return false;
}
return true;
}
function Turn() {
    $this->error = array("error" => "This method, TURN, of the SMTP " .
                        "is not implemented");
    if($this->do_debug >= 1) {
        echo "SMTP -> NOTICE: " . $this->error["error"] . $this->CRLF;
    }
    return false;
}
function Verify($name) {
    $this->error = null; # so no confusion is caused
    if(!$this->connected()) {
        $this->error = array(
            "error" => "Called Verify() without being connected");
        return false;
    }
    fputs($this->smtp_conn,"VRFY " . $name . $this->CRLF);
    $rply = $this->get_lines();
    $code = substr($rply,0,3);
    if($this->do_debug >= 2) {
        echo "SMTP -> FROM SERVER:" . $this->CRLF . $rply;
    }
    if($code != 250 && $code != 251) {
        $this->error =
            array("error" => "VRFY failed on name '$name'",
                  "smtp_code" => $code,
                  "smtp_msg" => substr($rply,4));
        if($this->do_debug >= 1) {
            echo "SMTP -> ERROR: " . $this->error["error"] .
                ": " . $rply . $this->CRLF;
        }
        return false;
    }
}

```

```

        return $rply;
    }
function get_lines() {
    $data = "";
    while($str = fgets($this->smtp_conn,515)) {
        if($this->do_debug >= 4) {
            echo "SMTP -> get_lines(): \"$data was \"$data\" .
                $this->CRLF;
            echo "SMTP -> get_lines(): \"$str is \"$str\" .
                $this->CRLF;
        }
        $data .= $str;
        if($this->do_debug >= 4) {
            echo "SMTP -> get_lines(): \"$data is \"$data\" . $this->CRLF;
        }
        if(substr($str,3,1) == " ") { break; }
    }
    return $data;
}
}
?>

```