



西南大學

应用程序设计报告

课程名称: 《数据库系统原理》

项目名称: 找朋友活动发布系统

项目组成员: 罗江涛

李峰

葛广付

指导老师: 李娅

2014 年 6 月 3 日



找朋友活动发布系统目录

一、系统概述	1
二、需求分析	2
2.1 系统功能说明	2
2.1.1 用户注册登录	2
2.1.2 个人首页	2
2.1.3 活动发布	2
2.1.4 活动报名	3
2.1.5 活动录取和站内私信	3
2.1.6 综合搜索	3
2.1.7 联系我们	3
2.2 项目执行流程图	3
2.3 系统目录结构	4
2.4 数据字典	5
三、系统设计	9
3.1 概念结构设计	9
3.2 逻辑结构设计	10
3.3 系统功能模块图	11
四、详细设计	12
4.1 数据库设计主要代码	12
4.2 程序设计部分代码	16



4.2.1 算法举例.....	16
4.2.2 数据库安全性控制.....	18
五、系统实现与测试.....	21
5.1 开发平台及工具.....	21
5.2 系统实现.....	22
六、团队总结.....	29
6.1 开发心得体会.....	29
6.2 团队合作开发分工情况.....	29
七、参考文献.....	30



找朋友活动发布系统

一、系统概述

人脉资源是人生一笔宝贵的财富，拥有更多朋友，不断提升沟通交流能力对于大学生的成长大有裨益。依据匈牙利作家 *Frigyas Karinthy* 提出的六度人脉理论，我们可以认识朋友的朋友为基础，通过网络扩展自己的人脉。并且能够无限扩张自己的人脉，在彼此需要的时候，可以随时得到该人脉的帮助。

本系统发布线下的活动，实现了大学生户线下交友互动，促进大学生之间的交流。基于六度人脉理论，可以结交更多的好友，大学生能够通过走出宅的宿舍，通向更加广阔的天地，得到更多的历练，与此同时也可以一并解决在虚拟世界里人们之间的信任问题。

全站中的查询操作采用无回送技术，提高了访问数据库的速度，对数据库操作实现存储过程化，实现了预编译，避免了 SQL 语句在网络中传输然后再解释的低效率，方便了修改，同时也提高了安全性，不会有 SQL 注入的问题。除此之外，系统后台管理系统对用户密码采用 MD5 算法加密，提高了系统的安全性；另一方面，还通过系统日志记录在任何页面对数据库进行的任何操作，能够有效保护数据库。

本系统是基于 B/S 模式下的应用软件，B/S 模式的优点在于利用当今最为普及的 Internet 网络，使任何用户都能够使用浏览器进行各自所需的操作，发布活动，结交有共同兴趣的好友，实现线下的交友。目前还没有网站涉及这个方面，这也是本系统的创新点所在。

下图为本系统 B/S 模型的系统结构图：

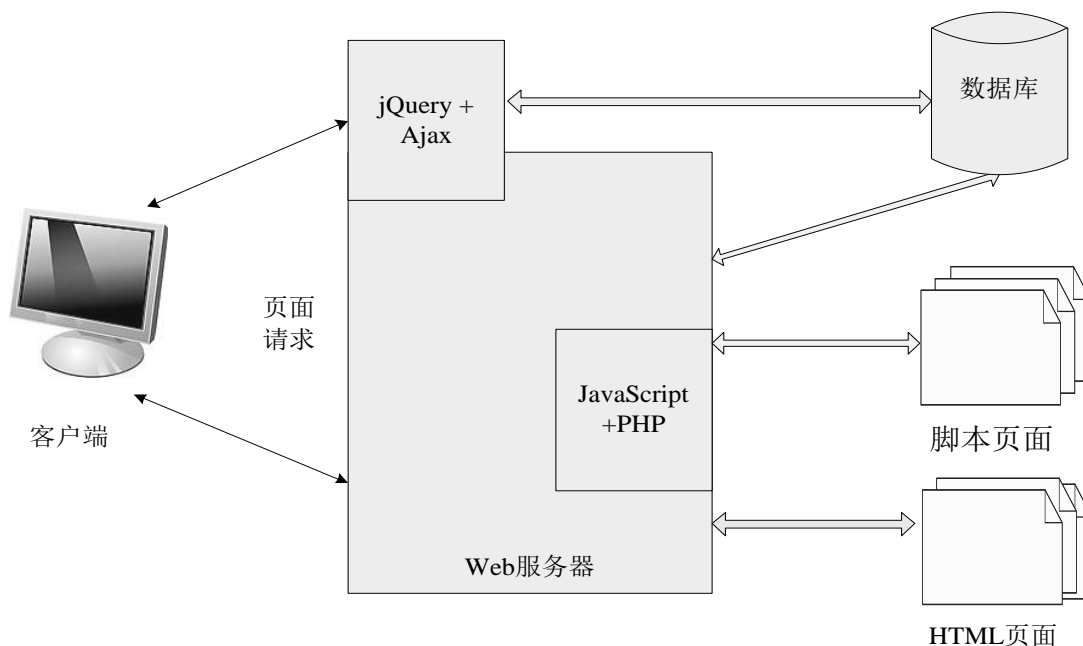


图 1 系统结构图



二、需求分析

在互联网高度发达的今天，社交网络也越来越普及，然而，大多数网站只限于线上的交流，并把人们束缚在了网络上，这严重影响了人类生产生活中的正常交往。人们越来越倾向于线上社交，很多朋友亲人等也都被搬到了社交网络上，导致了人们越来越依赖于社交网络。而人们面对面沟通交流的机会越来越少了。

现在大学生之间或多或少地存在交流障碍，有些甚至导致了人格或心灵上的缺失，以至于类似于复旦投毒案的事情发生，“宅男腐女”也成为大学生的代名词。为了减少宅现象，能够使更多的大学生走出寝室，走向户外，一起运动交流，结识更多的朋友来扩大自己的社交范围。我们开发了找朋友活动发布系统，该系统依据六度人脉理论，通过发布线下的活动，满足人们线下交友的需求。

2.1 系统功能说明

通过此网站可以发布活动，别的同学可以看到发布的活动，并且可以报名参加。发布活动的同学可以看到报名者的信息并选择是否录取。在报名成功后，即可以通过站私信进行认识，也可以通过留下的联系方式，进行线下社交活动。

项目网址（可直接登录进行查看）：<http://www.zyzpy.com>

在本系统中，定义了两个主要角色：管理员、注册用户。因此，本网站设立了管理员后台。在网站前台，有多个交互平台，每一个平台设置不同的模块，相互之间相对独立，但是又相互影响。

2.1.1 用户注册登录

网站提供用户的注册和登录，将注册提交的信息保存到服务器端文件中，用户在登录页面 `index.html` 正确输入用户名和密码登录，即可成功。若登录不成功，则提示“用户名或密码错误”，其中注册信息包括用户名、密码、性别、昵称等信息。

注册时会对用户名和密码进行简单的判断如：当用户输入注册信息时，如果用户名或密码没有填写时，系统出现会提示信息“用户名或密码不能为空！”；如果用户名在数据库中已经存在通过判断语句系统会有弹出提示信息“用户名已经存在，请重新注册！”。

网站浏览者可以通过本功能成为注册会员，注册完成后能以会员身份浏览网站，并具备更多的权限和访问更多资源。而管理员可以在后台对注册用户进行添加、删除、管理（例如用户附属信息修改、密码修改等）。同时也可以通过腾讯 QQ 账号来进行第三方的登录，我们现在使用的是西南大学内测版本的首页。

2.1.2 个人首页

个人首页为用户修改个人基本信息、上传用户图像、查看用户自己已发布的活动信息的功能。该页面还可以显示自己已经关注过的好友的列表，并且对自己的关注的好友、粉丝人数和已经发表过的活动进行统计。

2.1.3 活动发布



本网站主要是为活动发布和先下交友服务的，为此我们设计了活动发布和相应的活动录取页面，这也是我们网站的特色所在。

本系统主要功能进行了总体的描述，页面有四套模板相结合，为用户提供了舒适可靠的颜色选择。在系统左右页面的接口，方便访问者访问，主要有关关注好友、只看男生、只看女生等。在页面上提供查询本页面将为用户提供最新动态的好友活动发布通知，并且能够查看用户活动的具体内容。

本页面力求简单实用，页面清爽，容易操作，对活动发布、评论者提供头像和表情的服务，增加活动发布与评论，朋友之间交流的乐趣性。同时管理员可在后台对信息条目进行添加、删除及修改等操作。

2.1.4 活动报名

通过查找用户自己感兴趣的活动的，点击查看已经发布活动的具体内容，了解活动发布的时间、地点、活动组织形式和活动组织人员，决定是否报名。

2.1.5 活动录取和站内私信

由活动发布者来确认是否进行录取，并且通过站内私信的方式加以询问和通知，或者由留下的联系方式直接进行联系。此模块功能包括用户之间的短消息管理和系统短消息管理，用户之间可在站内进行有效沟通，以及能及时掌握网站发布的最新动态等信息。

2.1.6 综合搜索

注册用户通过关键词检索网站信息，搜索完成后返回包含关键词信息的页面列表，只要包含了关键字的页面都会根据相关度出现在搜索结果中。只要搜索好友或者活动，系统将会返回的结果。点击相应的查询结果即可进入查看活动的详情。

2.1.7 联系我们

一个好的网站应该是一个时时与访问者进行交流，让访问者提出改进的意见，因此我们设立了联系方式板块，建立与访问者互动交流的桥梁。

同时我们也在寻找志同道合的战友，在本页面介绍我们开发团队及成员的信息，提供了联系方式和交流 QQ 群，与此同时我们也希望能够有更多的人来加入我们，和我们一起努力一起奋斗。

2.2 项目执行流程图

在找朋友动态发布系统中，该项目的执行流程图如下：

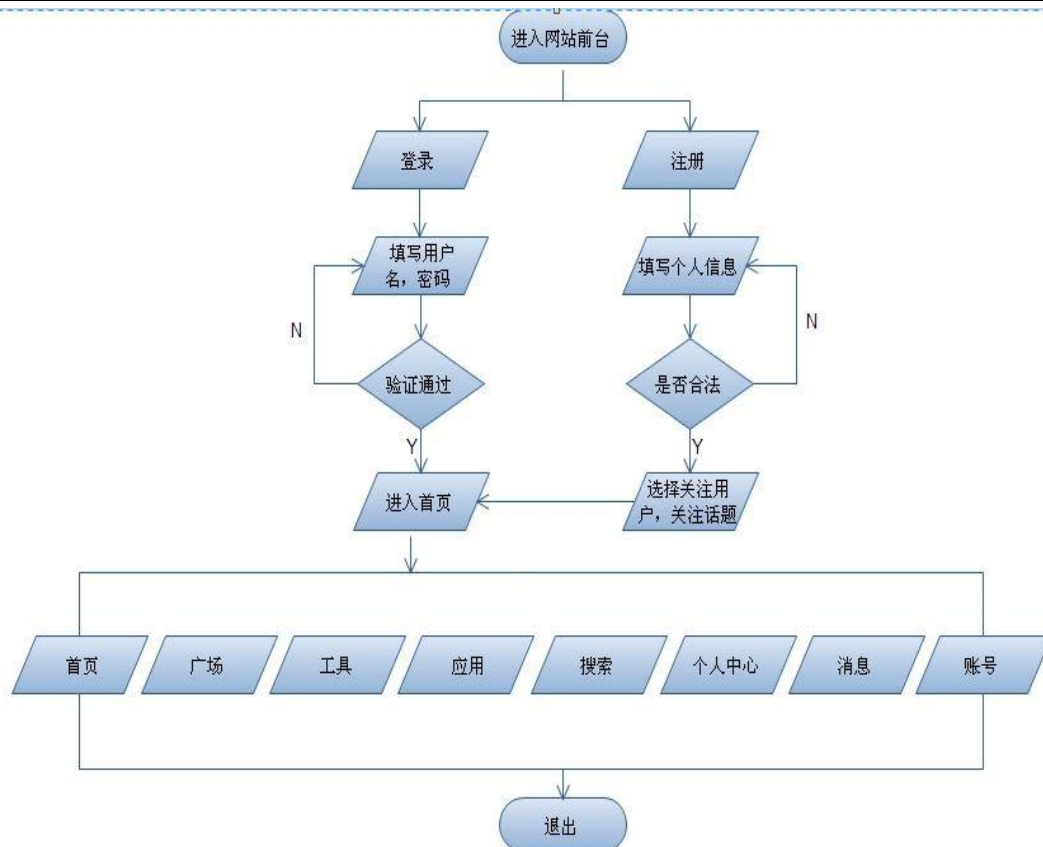


图 2.2 执行流程图

2.3 系统目录结构

在需求阶段，我们将网站程序分为了前台和后台两个大的应用。由于是通过 ThinkPHP 框架进行开发，所以只需要为每个应用单独声明一个入口文件，分别命名为 index.php(前台入口)和 admin.php(后台应用入口)，并存放到项目目录 zyzpy（该目录存放在 Web 服务器的项目目录下）下面。在前台应用的入口文件 index.php 中指定前台的应用目录为“index”目录，在后台应用的入口文件 admin.php 中指定后台的应用目录为“admin”目录。分别访问两个入口文件 ThinkPHP 自动生成项目结构目录，如下所示：

-- zyzpy 目录	#项目根目录
-- ThinkPHP 目录	#ThinkPHP 框架目录
-- index.php 文件	#前台主入口文件（可用其他名称）
-- index 目录	#自定义的前台项目应用目录
-- Lib 目录	#声明控制器类的目录（前台）
-- Conf 目录	#声明配置文件目录（前台）
-- Tpl 目录	#声明视图的目录（前台）
-- admin.php 文件	#后台主入口文件（可用其他名称）
-- admin 目录	#自定义的后台项目应用目录
-- Lib 目录	#声明控制器类的目录（后台）



- Conf 目录 #声明配置文件目录（后台）
- Tpl 目录 #声明视图的目录（后台）
- public 目录 #项目的所有应用公用的资源目录

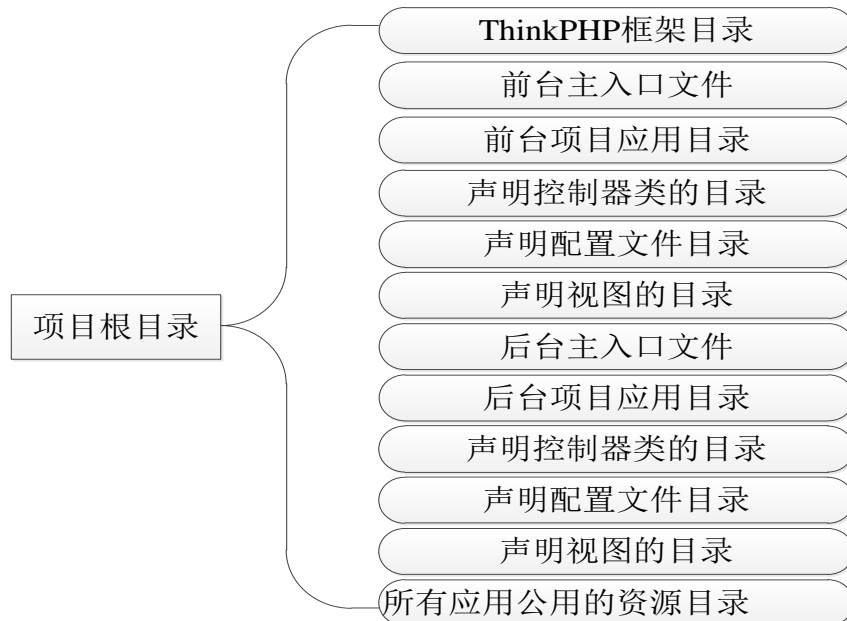


图 2.3 系统目录结构图

2.4 数据字典

字段名	字段类型	是否 为空/ 默认 值	是否自增	主键/ 唯一 索引	代表意思
hd_user 用户表					
Id	Int	not null	auto_incremen t	PRI	Id
Account	varchar(60)	not null	NO		账号
Password	varchar(60)	not null	NO		密码
Registime	Int	not null	NO		注册时间
Lock	tinyint(1)	0	NO		是否锁定（0： 否，1：是）
hd_userinfo 用户信息表					



Id	Int	not null	auto_increment	PRI	Id
Username	varchar (33)	not null	NO		用户昵称
Truename	char (14)	not null	NO		真实名称
Sex	Int	not null	NO		性别
Location	varchar (30)	(NULL)	NO		所在地
Constellation	varchar (30)	(NULL)	NO		星座
Intro	varchar (30)	(NULL)	NO		一句话介绍自己
face50	varchar (120)	Null	NO		50*50 头像
face80	varchar (33)	Null	NO		80*80 头像
face180	varchar (120)	Null	NO		180*180 头像
Style	varchar (30)	Default	NO		个性模版
Follow	Int	0	NO		关注数
Weibo	Int	0	NO		活动数
Uid	Int	0			所属用户 ID
hd_group 分组表					
Id	Int	not null	auto_increment	PRI	Id
Name	varchar (60)	not null	NO		分组名称
Uid	Int	not null	NO		所属用户的 ID
hd_follow 关注表					
Follow	Int	not null	auto_increment	PRI	Id
Fans	Int	not null	NO		粉丝用户 ID
Gid	Int	not null	NO		所属关注分组 ID
hd_weibo 活动表					
Id	Int	not null	auto_increment	PRI	Id
Content	varchar (33)	not	NO		活动内容



)	null			
Isturn	Int	not null	NO		是否转发 (0: 原创, 如果是转发的则保存该转发活动的 ID)
Time	Int	not null	NO		发布时间
Turn	Int	not null	NO		转发次数
Keep	Int	not null	NO		收藏次数
Comment	Int	not null	NO		评论次数
Uid	Int	not null	NO		所属用户的 ID
hd_picture 图片表					
Id	Int	not null	auto_increment	PRI	Id
Mini	varchar(50)	not null	NO		小图
Medium	varchar(50)	not null	NO		中图
Max	varchar(50)	not null	NO		大图
Wid	Int	(NULL)	NO		所属活动 ID
hd_atme 艾特表					
Id	Int	not null	auto_increment	PRI	Id
Wid	Int	not null	NO		提到我的活动 ID
Uid	Int	not null	NO		所属用户 ID
hd_letter 私信表					
Id	Int	not null	NO	PRI	Id
From	Int	not null	NO		发私用户 ID
Content	varchar(50)	not null	NO		私信内容
Time	Int	not null	NO		私信发送时间



Uid	Int	not null	NO		所属用户 ID (收信人)
hd_comment 评论表					
Id	Int	(NULL)	auto_increment	PRI	Id
Content	varchar(50)	(NULL)	NO		评论内容
Time	Int	(NULL)	NO		评论时间
Uid	Int	(NULL)	NO		评论用户的 ID
Wid	Int	(NULL)	NO		所属活动 ID
hd_keep 收藏表					
Id	Int	not null	auto_increment	PRI	Id
Uid	Int	not null	NO		收藏用户的 ID
Wid	Int	not null	NO		收藏活动的 ID
Time	Int	not null	NO		收藏时间
hd_activity 活动详细信息表					
Id	Int	not null	NO	PRI	Id
Location	varchar(60)	not null	NO		活动地点
activity_time	Int	not null	NO		活动时间
Qq	varchar(30)	not null	NO		联系 QQ
Telephone	varchar(30)	not null	NO		联系电话
Wid	Int	not null	NO		活动 ID
hd_baoming 报名信息表					
Id	Int	not null	auto_increment	PRI	Id
Uid	Int	not null	NO		报名人 ID
is_luqu	Int	not null	NO		是否录取: 0 没 1 是
Wid	Int	not null	NO		所属活动 ID
hd_admin 管理员表					



Id	int(10)	not null	auto_increment	prI	Id
Username	varchar(32)	not null	No		管理员登录名
Password	char(32)	not null	No		登录密码
Logintime	INT	not null	No		登录时间
Loginip	varchar()	not null	No		登录 IP
Lock	tinyint(1)	not null	No		管理员的登录状态
Admin	Int	not null	No		0: 超级管理员; 1: 普通管理员

三、系统设计

3.1 概念结构设计

本系统基于六度人脉理论，是通过活动将不认识的陌生人联系起来的，同时，活动的形式也多种多样。通过分析，我们得出用户在现实生活中的联系图如下：

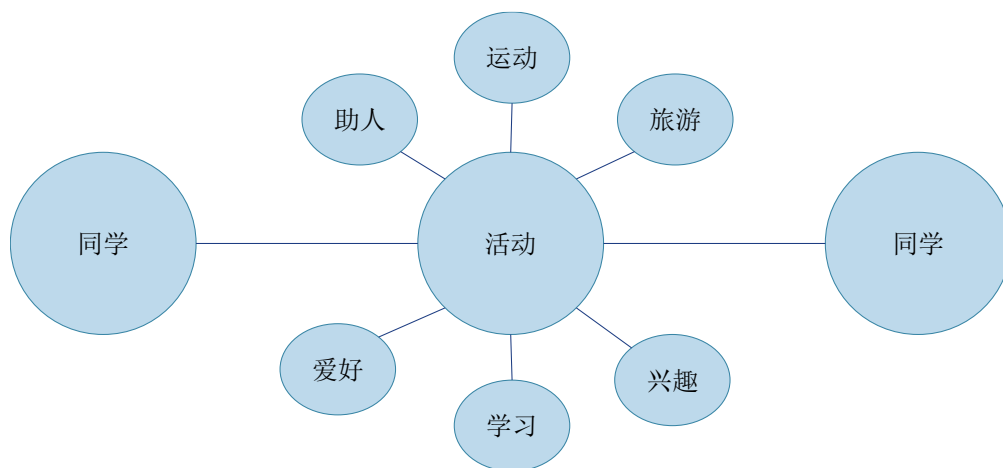


图 3.1.1 现实中用户之间的联系

对以上系统进行分析，依次可以抽象出系统的 E-R 图，系统的关系模式图，系统功能模块图等。

其中，找朋友动态发布系统 E-R 图，如下面 3.1.2 图所示：

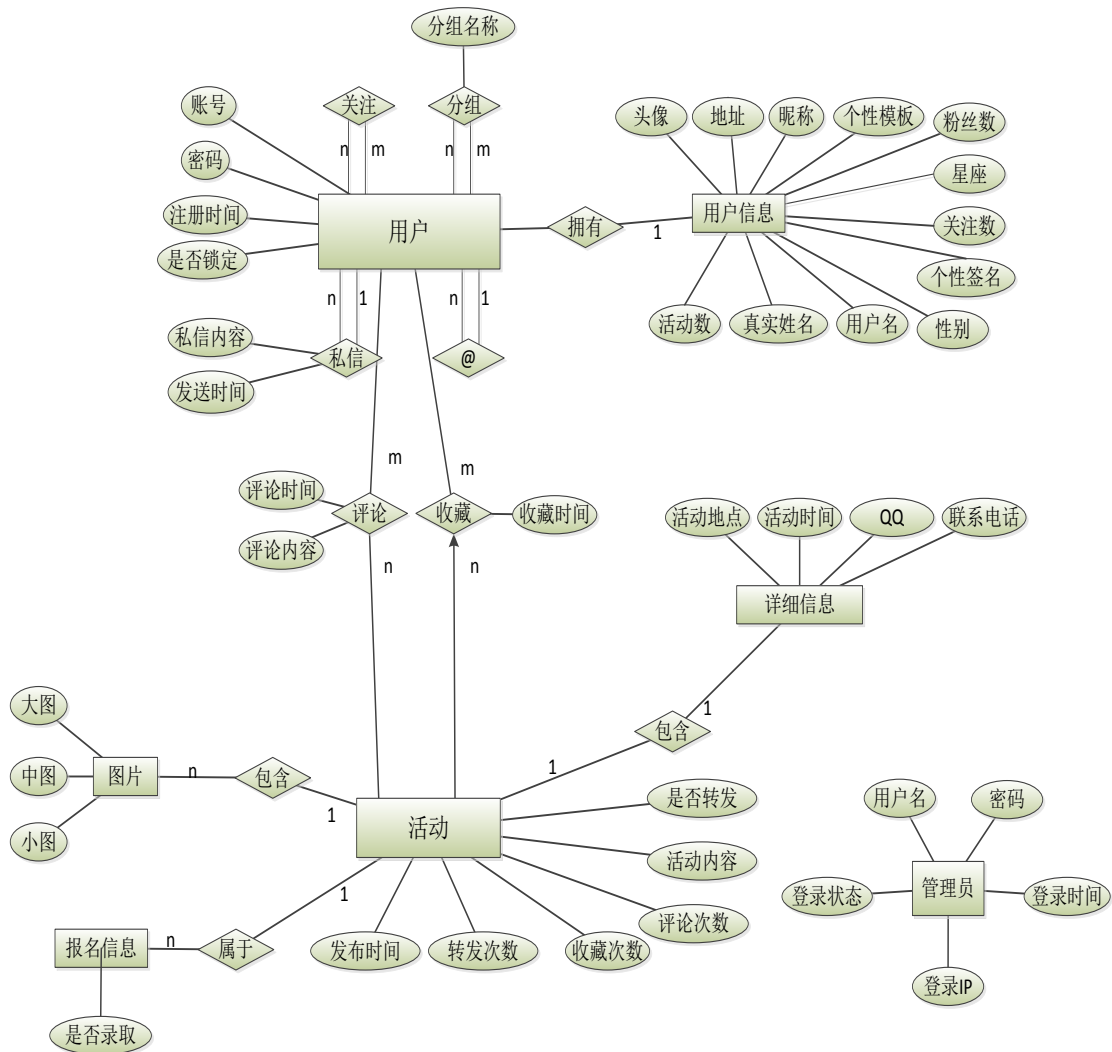


图 3.1.2 系统 E-R 图

3.2 逻辑结构设计

通过分析出的 E-R 图，可以得到如下图所示的系统关系模式图：

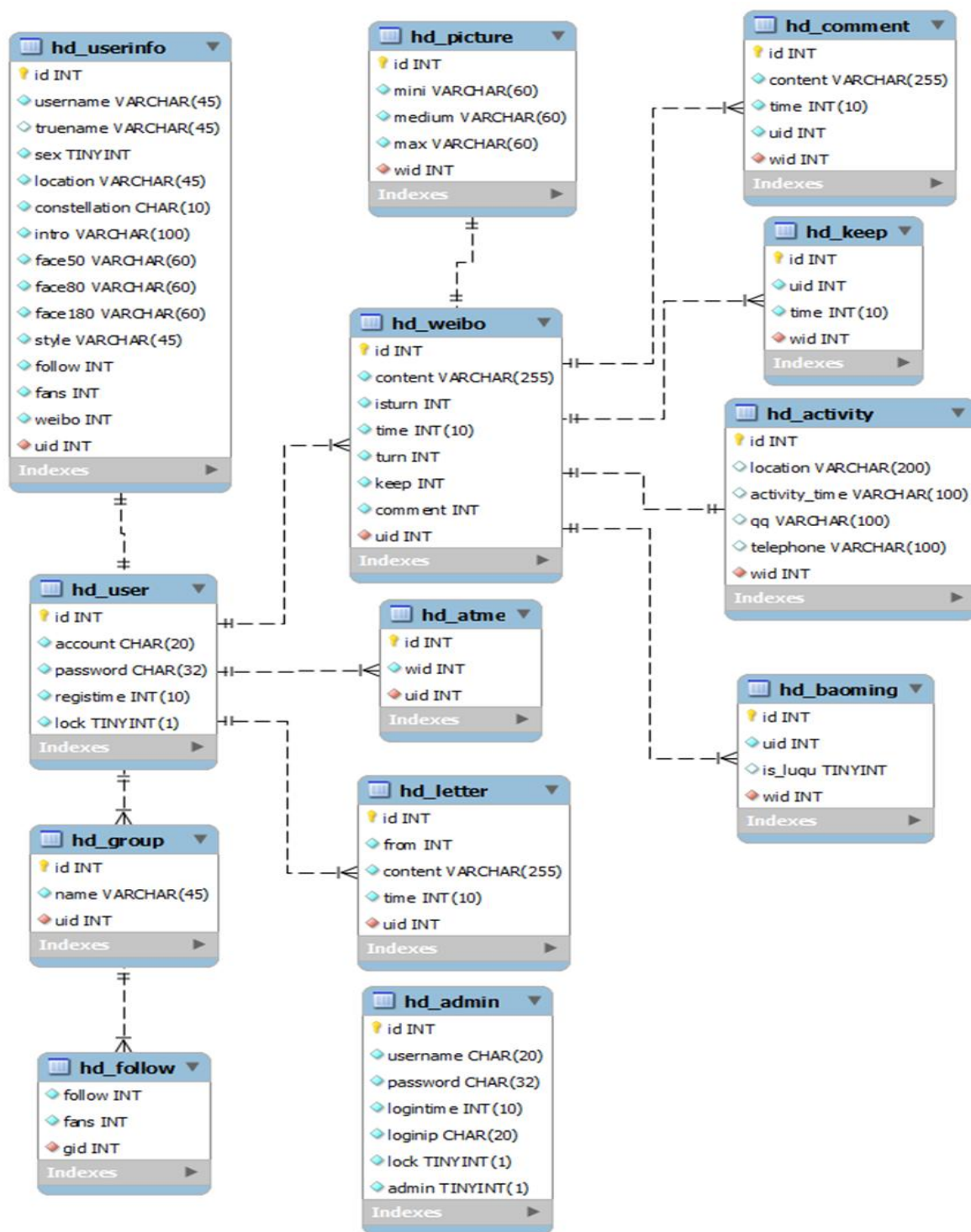
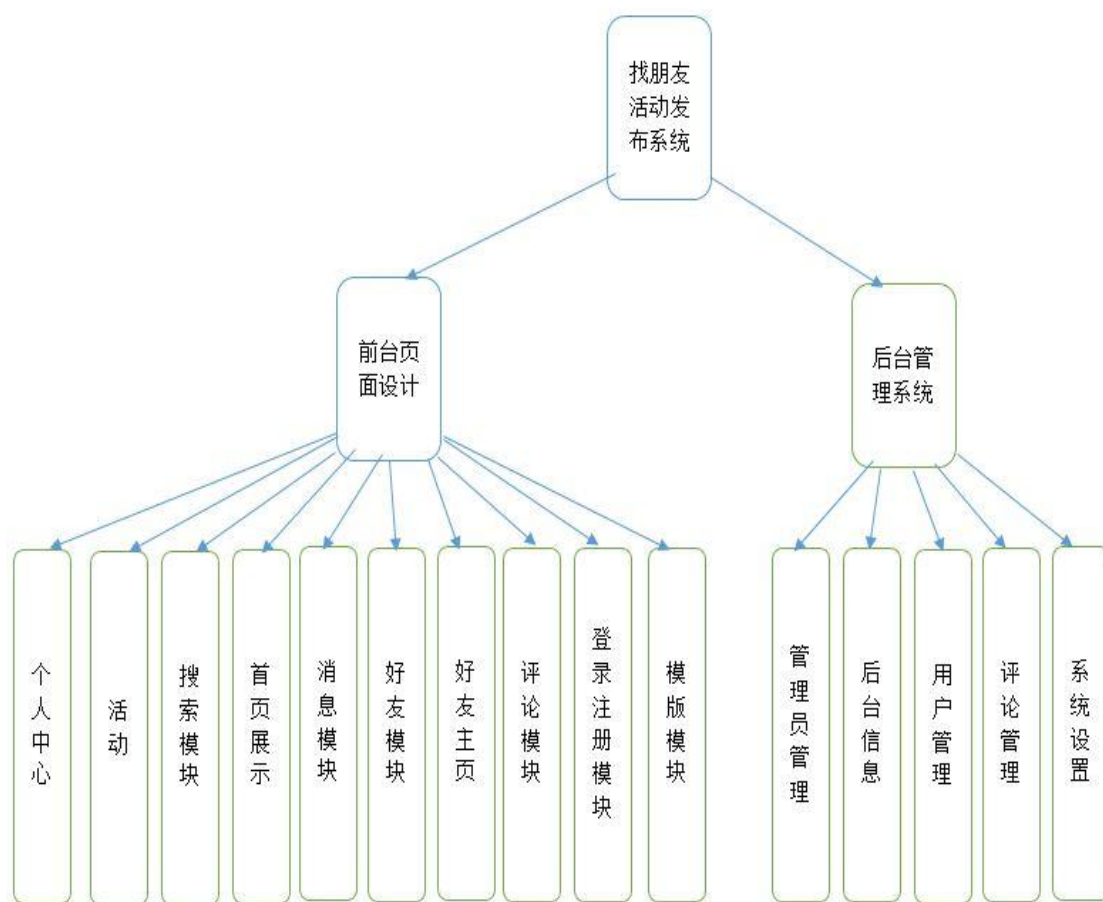


图 3.2.1 系统 UML 图

3.3 系统功能模块图

根据我们设计的系统和实现的功能，我们可以得到如下功能图：



四、详细设计

4.1 数据库设计主要代码

```
DROP DATABASE IF EXISTS weibo;
create database weibo;
use weibo;
DROP TABLE IF EXISTS `hd_user` ;
CREATE TABLE IF NOT EXISTS `hd_user` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `account` CHAR(20) NOT NULL DEFAULT '' COMMENT '账号' ,
  `password` CHAR(32) NOT NULL DEFAULT '' COMMENT '密码' ,
  `registime` INT(10) UNSIGNED NOT NULL COMMENT '注册时间' ,
  `lock` TINYINT(1) UNSIGNED NOT NULL DEFAULT 0 COMMENT '是否锁定 (0: 否, 1: 是)' ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `account` (`account` ASC) )
DEFAULT CHARSET=utf8,
```



COMMENT = '用户表';

DROP TABLE IF EXISTS `hd_userinfo` ;

```
CREATE TABLE IF NOT EXISTS `hd_userinfo` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `username` VARCHAR(45) NOT NULL DEFAULT '' COMMENT '用户昵称' ,  
  `truename` VARCHAR(45) NULL COMMENT '真实名称' ,  
  `sex` tinyint NOT NULL DEFAULT '1' COMMENT '性别' ,  
  `location` VARCHAR(45) NOT NULL DEFAULT '' COMMENT '所在地' ,  
  `constellation` CHAR(10) NOT NULL DEFAULT '' COMMENT '星座' ,  
  `intro` VARCHAR(100) NOT NULL DEFAULT '' COMMENT '一句话介绍自己' ,  
  `face50` VARCHAR(60) NOT NULL DEFAULT '' COMMENT '50*50 头像' ,  
  `face80` VARCHAR(60) NOT NULL DEFAULT '' COMMENT '80*80 头像' ,  
  `face180` VARCHAR(60) NOT NULL DEFAULT '' COMMENT '180*180 头像' ,  
  `style` VARCHAR(45) NOT NULL DEFAULT 'default' COMMENT '个性模版' ,  
  `follow` INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '关注数' ,  
  `fans` INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '粉丝数' ,  
  `weibo` INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '微博数' ,  
  `uid` INT NOT NULL COMMENT '所属用户 ID' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `uid` (`uid` ASC) )  
DEFAULT CHARSET=utf8,  
COMMENT = '用户信息表';  
DROP TABLE IF EXISTS `hd_group` ;
```

```
CREATE TABLE IF NOT EXISTS `hd_group` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(45) NOT NULL DEFAULT '' COMMENT '分组名称' ,  
  `uid` INT NOT NULL COMMENT '所属用户的 ID' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `uid` (`uid` ASC) )  
DEFAULT CHARSET=utf8,  
COMMENT = '关注分组表';  
DROP TABLE IF EXISTS `hd_follow` ;
```

```
CREATE TABLE IF NOT EXISTS `hd_follow` (  
  `follow` INT UNSIGNED NOT NULL COMMENT '关注用户的 ID' ,  
  `fans` INT UNSIGNED NOT NULL COMMENT '粉丝用户 ID' ,  
  `gid` INT NOT NULL COMMENT '所属关注分组 ID' ,  
  INDEX `follow` (`follow` ASC) ,  
  INDEX `fans` (`fans` ASC) ,  
  INDEX `gid` (`gid` ASC) )  
DEFAULT CHARSET=utf8,
```




COMMENT = '关注与粉丝表';

DROP TABLE IF EXISTS `hd_letter` ;

```
CREATE TABLE IF NOT EXISTS `hd_letter` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `from` INT NOT NULL COMMENT '发私用户 ID' ,  
  `content` VARCHAR(255) NOT NULL DEFAULT '' COMMENT '私信内容' ,  
  `time` INT(10) UNSIGNED NOT NULL COMMENT '私信发送时间' ,  
  `uid` INT NOT NULL COMMENT '所属用户 ID (收信人)' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `uid` (`uid` ASC) )  
DEFAULT CHARSET=utf8,  
COMMENT = '私信表';  
DROP TABLE IF EXISTS `hd_weibo` ;
```

```
CREATE TABLE IF NOT EXISTS `hd_weibo` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `content` VARCHAR(255) NOT NULL DEFAULT '' COMMENT '微博内容' ,  
  `isturn` INT NOT NULL DEFAULT 0 COMMENT '是否转发 (0: 原创, 如果是转发的则保存该转发微博的 ID)' ,  
  `time` INT(10) UNSIGNED NOT NULL COMMENT '发布时间' ,  
  `turn` INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '转发次数' ,  
  `keep` INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '收藏次数' ,  
  `comment` INT UNSIGNED NOT NULL DEFAULT 0 COMMENT '收藏次数' ,  
  `uid` INT NOT NULL COMMENT '所属用户的 ID' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `uid` (`uid` ASC) )  
DEFAULT CHARSET=utf8,  
COMMENT = '微博表';  
DROP TABLE IF EXISTS `hd_picture` ;
```

```
CREATE TABLE IF NOT EXISTS `hd_picture` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `mini` VARCHAR(60) NOT NULL DEFAULT '' COMMENT '小图' ,  
  `medium` VARCHAR(60) NOT NULL DEFAULT '' COMMENT '中图' ,  
  `max` VARCHAR(60) NOT NULL DEFAULT '' COMMENT '大图' ,  
  `wid` INT NOT NULL COMMENT '所属微博 ID' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `wid` (`wid` ASC) )  
DEFAULT CHARSET=utf8,  
COMMENT = '微博配图表';  
DROP TABLE IF EXISTS `hd_comment` ;
```



```
CREATE TABLE IF NOT EXISTS `hd_comment` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `content` VARCHAR(255) NOT NULL DEFAULT '' COMMENT '评论内容' ,  
  `time` INT(10) UNSIGNED NOT NULL COMMENT '评论时间' ,  
  `uid` INT NOT NULL COMMENT '评论用户的 ID' ,  
  `wid` INT NOT NULL COMMENT '所属微博 ID' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `uid` (`uid` ASC) ,  
  INDEX `wid` (`wid` ASC) )  
  DEFAULT CHARSET=utf8,  
COMMENT = '微博评论表';  
DROP TABLE IF EXISTS `hd_keep` ;
```

```
CREATE TABLE IF NOT EXISTS `hd_keep` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `uid` INT NOT NULL COMMENT '收藏用户的 ID' ,  
  `time` INT(10) UNSIGNED NOT NULL COMMENT '收藏时间' ,  
  `wid` INT NOT NULL COMMENT '收藏微博的 ID' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `wid` (`wid` ASC) ,  
  INDEX `uid` (`uid` ASC) )  
  DEFAULT CHARSET=utf8,  
COMMENT = '收藏表';  
DROP TABLE IF EXISTS `hd_atme` ;
```

```
CREATE TABLE IF NOT EXISTS `hd_atme` (  
  `id` INT NOT NULL AUTO_INCREMENT ,  
  `wid` INT NOT NULL COMMENT '提到我的微博 ID' ,  
  `uid` INT NOT NULL COMMENT '所属用户 ID' ,  
  PRIMARY KEY (`id`) ,  
  INDEX `uid` (`uid` ASC) ,  
  INDEX `wid` (`wid` ASC) )  
  DEFAULT CHARSET=utf8,  
COMMENT = '@提到我的微博';
```

```
DROP TABLE IF EXISTS `hd_admin` ;  
create table hd_admin(  
  id int unsigned not null primary key auto_increment,  
  username char(20) not null default '',  
  password char(32) not null default '',  
  logintime int(10) unsigned not null default 0,  
  loginip char(20) not null default '',  
  `lock` tinyint(1) unsigned not null default 0,
```



```
`admin` tinyint(1) unsigned not null default 1
)DEFAULT CHARSET=utf8;

insert into hd_admin set
username='admin',password=md5('admin'),logintime=unix_timestamp(now
()),loginip='127.0.0.1',admin=0;
```

4.2 程序设计部分代码

4.2.1 算法举例



图 4.2.1 复杂算法举例

首先，什么是可能感兴趣的人？依据六度人脉理论，朋友的朋友就是朋友，这是我们所用算法的基本思想。

➤ 算法步骤

1. 找出我所关注的人
2. 找出我所关注的人所关注的人
3. 统计共同好友，并按统计数排序
4. 关联 userinfo 表找出用户相关信息，如姓名，头像等。

➤ 算法代码

```
<!--可能感兴趣的人-->
<?php
$db=M('follow');
$where=array('fans'=>session('uid'));
$follow=$db->where($where)->field('follow')->select();
foreach($follow as $k=>$v){
    $follow[$k] = $v['follow'];
}
```



//这条 sql 语句是算法的关键

```
$sql='SELECT `uid`,`username`,`face50` AS `face`,COUNT(f.`follow`)
AS `count` FROM `hd_follow` f LEFT JOIN `hd_userinfo` u ON
f.`follow`=u.`uid` WHERE f.`fans` IN(' .implode(',',$follow).') AND
f.`follow` NOT IN(' . implode(',',$follow) .') AND f.`follow` <> '.
session('uid') .' GROUP BY f.`follow` ORDER BY `count` DESC LIMIT 4';
```

//发送 sql

```
$friend=$db->query($sql);
if(!$friend) {
$where=array('fans'=>session('uid'));
$follow=$db->where($where)->field('follow')->select();
if($follow) {
foreach($follow as $k=>$v) {
$follow[$k]=$v['follow'];
}
}
$follow[]=session('uid');
$where=array('uid'=>array('NOT IN',$follow));
$friend=M('userinfo')->limit(4)->where($where)->field(array('face
50'=>'face','username','uid'))->select();
}
```

?>

<hr/>

<h6 class='center-block text-center text-muted'>可能感兴趣的人</h6>

//循环出查询结果

```
<foreach name='friend' item='v'>
```

```
<p>
```

```
<div class="row">
```

```
<div class="col-md-3">
```

```
<a href="{:U('/'.$v['uid'])}" target="_blank">
```

```
__ROOT__/Uploads/Face/{ $v.face}
```

```
<else/>
```

```
__PUBLIC__/Them/{ $style}/images/1.jpg
```

```
</if>
```

```
" width="30px" height="30px"/></a>
```

```
</div>
```

```
<div class="col-md-5">
```

```
<b><a href="{:U('/'.$v['uid'])}"
target="_blank">{ $v.username}</a></b><br/>
```

```
<small><if condition=" $v['count']">{ $v.count}<else/>0</if> 个好友
</small>
```



```

</div>
<div class="col-md-3">
<!--<button type="button" class="btn btn-default btn-xs pull-left">
<span class="glyphicon glyphicon-plus"></span>关注
</button>-->

<button class="btn btn-warning guanzhu btn-xs" uid='{ $v.uid}'
data-toggle="modal" data-target="#guanzhu">
<span class="glyphicon glyphicon-plus"></span>关注
</button>
</div>
</div>
</p>
</foreach>
//循环结束

```

4.2.2 数据库安全性控制

◆ 防止 SQL 注入

对于 WEB 应用来说，SQL 注入攻击无疑是首要防范的安全问题，系统底层对于数据安全方面本身进行了很多的处理和相应的防范机制，例如：

1. `$User = M("User");` // 实例化 User 对象
2. `$User->find($_GET["id"]);`

即使用户输入了一些恶意的 id 参数，系统也会强制转换成整型，避免恶意注入。这是因为，系统会对数据进行强制的数据类型检测，并且对数据来源进行数据格式转换。

然而在实际应用中，通常的安全隐患在于你的查询条件使用了字符串参数，然后其中一些变量又依赖于客户端的用户输入，要有效的防止 SQL 注入问题，应该注意以下几点：我们采取了以下几种方式：

1. 查询条件尽量使用数组方式，这是更为安全的方式；
2. 如果不得已必须使用字符串查询条件，使用预处理机制；
3. 开启数据字段类型验证，可以对数值数据类型做强制转换；
4. 使用自动验证和自动完成机制进行针对应用的自定义过滤；
5. 字段类型检查、自动验证和自动完成机制。

◆ 查询条件预处理

在安全性的控制之上，我们除了采取了上述防止 SQL 注入的方式之外，还对查询条件进行了预处理处理。在程序中 where 方法使用字符串条件的时候，支持预处理（安全过滤），并支持两种方式传入预处理参数，例如：

```
$Model->where("id=%d and username='%s' and xx='%f'", array($id, $username, $xx))->select();
```

或者

```
$Model->where("id=%d and username='%s' and xx='%f'", $id, $username, $xx)->select();
```

模型的 query 和 execute 方法 同样支持预处理机制，例如：



```
$model->query('select * from user where id=%d and status=%d', $id, $status);
```

或者

```
$model->query('select * from user where id=%d and status=%d', array($id, $status));
```

● 数据库安全性控制举例

由于数据库安全性问题在程序设计和实现中十分突出，而在实际操作过程中必须处处注意这个问题。在此，我们将给出，找朋友动态发布系统中用户注册之后修改个人资料时的源代码，主要利用上面两种方法，从程序设计的角度具体来看我们在实际应用过程中采取的措施。

```
<?php
//用户信息控制器
class UserSettingAction extends Action{
//用户基本信息视图
public function index() {
    $where = array('uid' => session('uid'));
    $field = array('username', 'truename', 'sex', 'location',
'constellation', 'intro', 'face180');
    $user=M('userinfo')->field($field)->where($where)->find();
    $this->user=$user;
    //p($user);die;
    $this->display();
}
//修改用户基本信息
public function editBasic() {
    $data=array(
        'username'=>$this->_post('username'),
        'truename'=>$this->_post('truename'),
        'sex'=>$this->_post('sex'),
        'location'=>$this->_post('location'),
        'constellation'=>$this->_post('constellation'),
        'intro'=>$this->_post('intro'),
    );

    $sex=$_POST['sex'];
    if($data['sex'] == 0){
        $style=array(
            'style'=>'style2'
        );
        M('userinfo')->where(array('uid'=>session('uid')))->save($style);
    }
    $where=array('uid'=>session('uid'));
    $db=M('userinfo');
```



```
if($db->where($where)->data($data)->save()){
    $this->success('修改成功 ^_^',U('index'));
}else{
    $this->error('修改失败 ~~!');
}
}

//修改用户头像
public function editFace() {
    if(!$this->isPost()){
        halt('页面不存在');
    }
    $db=M('userinfo');
    $where=array('uid'=>session('uid'));
    $filed=array('face50','face80','face180');
    $old=$db->where($where)->field($filed)->find();
    if($db->where($where)->save($_POST)){
        if(!empty($old['face180'])){
            @unlink('./Uploads/Face/'.$old['face180']);
            @unlink('./Uploads/Face/'.$old['face80']);
            @unlink('./Uploads/Face/'.$old['face50']);
        }
        $this->success('修改成功',U('index'));
    }else{
        $this->error('修改失败，请重试。。。');
    }
}

//修改密码
public function editPwd() {
    if(!$this->isPost()){
        halt('页面不存在');
    }
    $db=M('user');
    $where=array('id'=>session('uid'));
    $old=$db->where($where)->getField('password');
    if($this->_post('oldPassword','md5')!=$old){
        $this->error('旧密码错误');
    }

    if($this->_post('newPassword')!=$this->_post('rePassword')){
        $this->error('两次密码不一致');
    }
    $newPwd=$this->_post('newPassword','md5');
```



```
$data=array(  
    'id'=>session('uid'),  
    'password'=>$newPwd  
);  
if($db->save($data)){  
    $this->success('修改成功',U('index'));  
}else{  
    $this->error('修改失败,请重试');  
}  
}  
?>
```

五、系统实现与测试

5.1 开发平台及工具

CPU: Intel 2.5GHz

开发环境: Microsoft Windows 7。

开发平台: Zend Studio 10

数据库系统: Mysql,

系统架构: 系统采用了 MVC 框架开发

所用技术: PHP、ThinkPHP、HTML、CSS、JavaScript、Jquery、

BootStrap、 Ajax 异步等

运行要求: Windows 下 IE9 以下不支持、Firefox 小于 3.5 不支持。Chrome 小于 3.0 不支持, Opera 小于 10.5 不支持。

名称	说明
MySQL	MySQL 是最流行的开放源码 SQL 数据库管理系统。在实际使用中 MySQL 数据库服务器具有快速、可靠和易于使用的特点。MySQL 服务器工作在客户端/服务器模式下, 或嵌入式系统中, 能够工作在众多不同的平台上。
MVC	MVC 是一个框架模式, 它强制性的使应用程序的输入、处理和输出分开。使用 MVC 应用程序被分成三个核心部件: 模型、视图、控制器。它们各自处理自己的任务, 由于运用 MVC 的应用程序的三个部件是相互独立, 改变其中一个不会影响其它两个, 所以依据这种设计思想能构造良好的松耦合和重用性高的构件。
AJAX	AJAX 并不是一门新的语言, 而也不是一个全新的技术, 它只是将现的技术结合起来, 构建更高效的 WEB 的一个技术。AJAX 在浏览器与 Web 服务器之间使用异步数据传输, 这样就可使网页从服务器请求少量的信息, 而不是整个页面。采用 AJAX 技术应用程序更小、更快, 更友好。



Bootstrap	<p>Bootstrap 是 Twitter 推出的一个开源的用于前端开发的工具包。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它即是由动态 CSS 语言 Less 写成。</p> <p>Bootstrap 是基于 jQuery 框架开发的，它在 jQuery 框架的基础上进行了更为个性化和人性化的完善，形成一套自己独有的网站风格，并兼容大部分 jQuery 插件。Bootstrap 把现有框架进行了清晰的功能划分，主要分为框架（Scaffolding），基础 CSS，构件库和 jQuery 插件库。这些在实际操作过程中都十分有利于程序的开发。</p>
ThinkPHP	<p>ThinkPHP 是一个遵循 Apache2 开源协议发布的，免费开源、快速、简单的面向对象的轻量级 PHP 开发框架。ThinkPHP 提供了灵活的全局配置功能，采用最有效率的 PHP 返回数组方式定义，支持惯例配置、项目配置、分组配置、调试配置和动态配置，并且会自动生成配置缓存文件，无需重复解析的开销。</p> <p>此外，ThinkPHP 在项目配置上面创造了自己独有的分层配置模式，其配置层次体现在：惯例配置->项目配置->调试配置->分组配置->扩展配置->动态配置上面。</p>
jQuery	<p>jQuery 是一个兼容多浏览器的 javascript 框架。jQuery 的语法设计可以使开发者更加便捷，例如操作文档对象、选择 DOM 元素、制作动画效果、事件处理、使用 Ajax 以及其他功能。除此以外，jQuery 提供 API 让开发者编写插件。其模块化的使用方式使开发者可以很轻松的开发出功能强大的静态或动态网页。</p>

表 5.1 开发工具选择说明

5.2 系统实现

以下为系统实现后的主要界面截图：

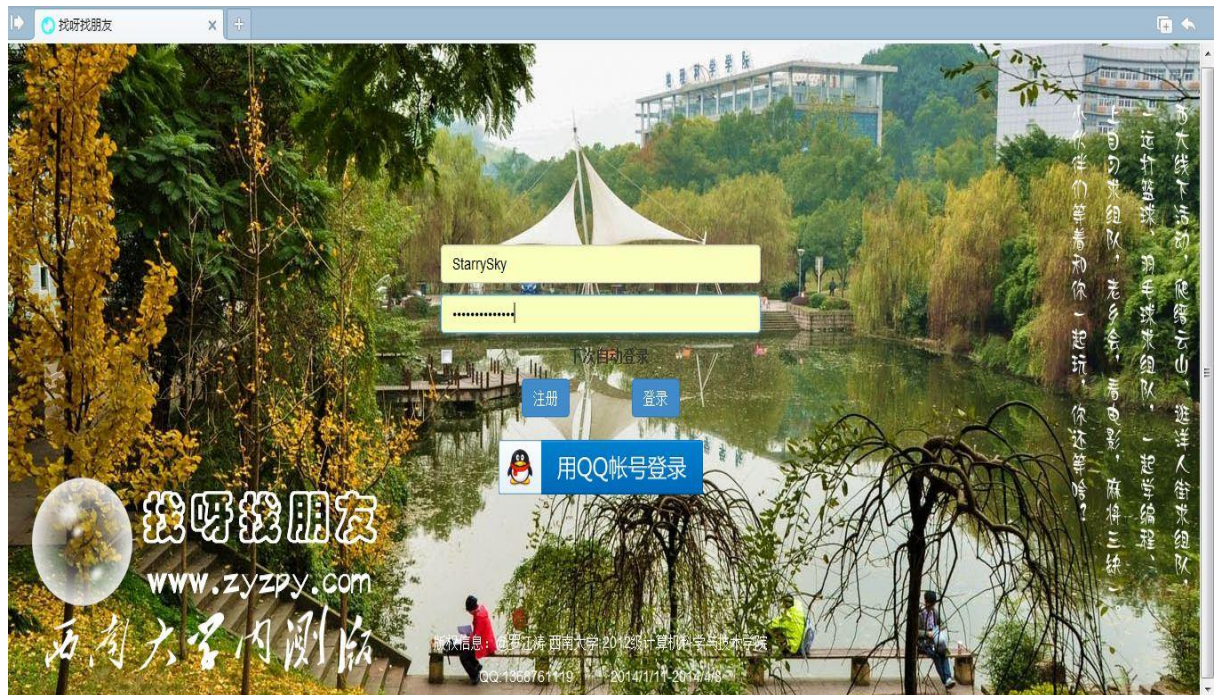


图 5.2.1 系统登录界面

登录 注册

用户名

☐男 ☐女

密码

确认密码

昵称

验证码

5878

注册

版权信息: @罗江涛 西南大学 2012级计算机科学与技术学院

QQ:1368761119 2014/1/11-2014/4/8

图 5.2.2 用户注册界面

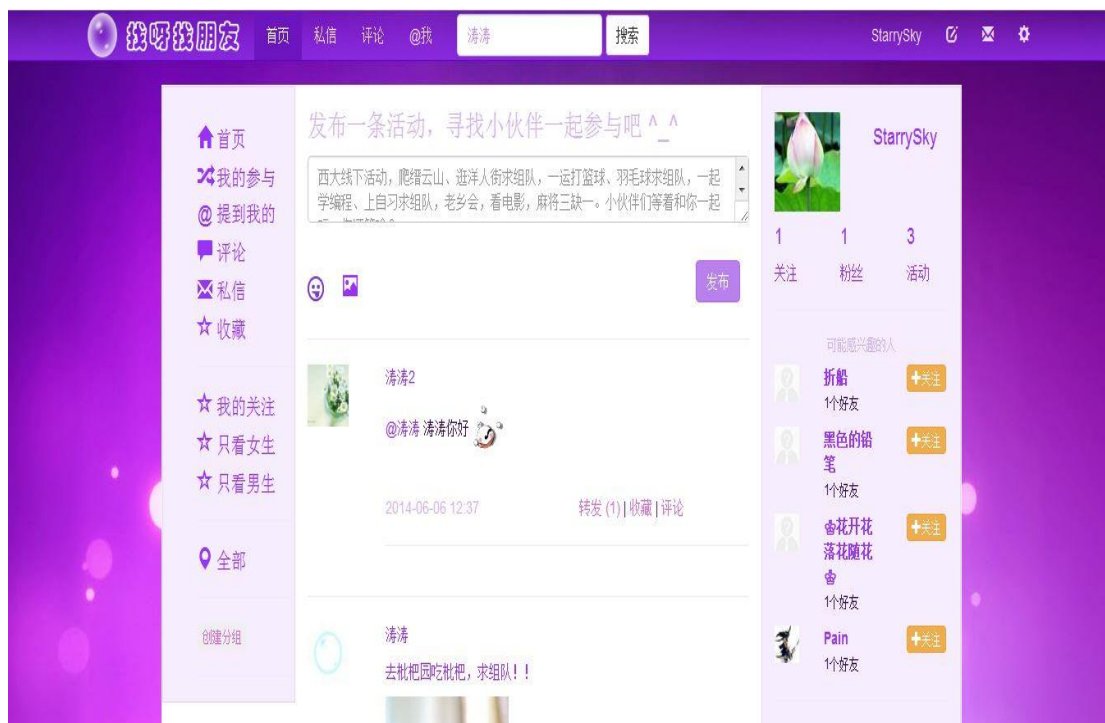


图 5.2.3 系统登录后界面

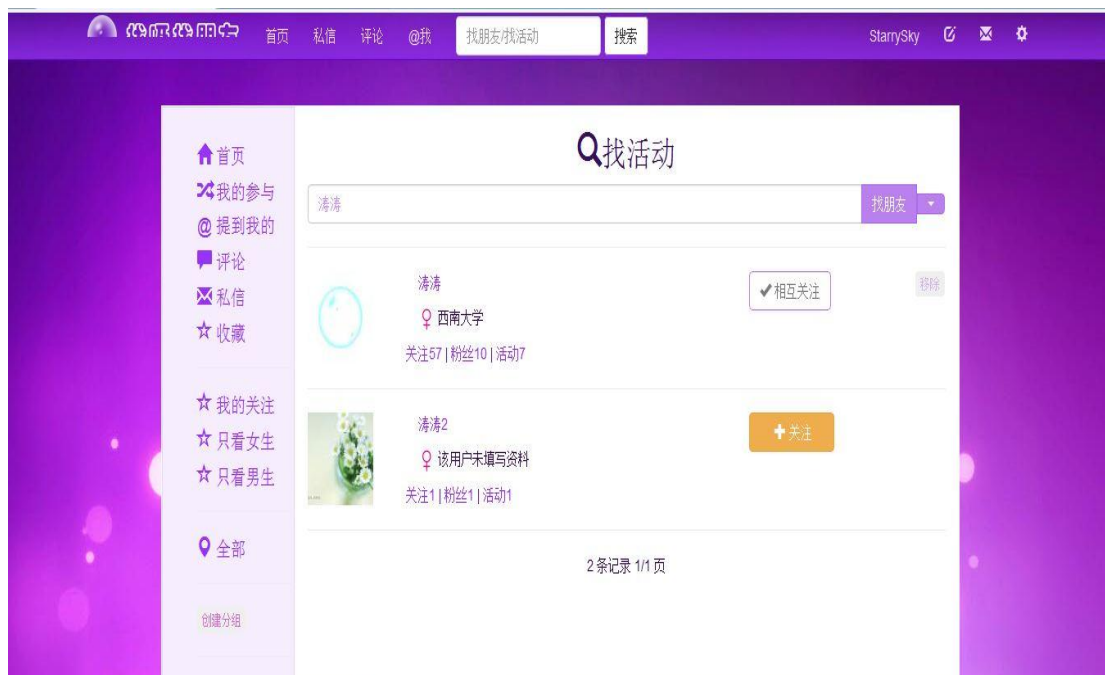


图 5.2.4 系统搜索功能



图 5.2.5 用户个人主页

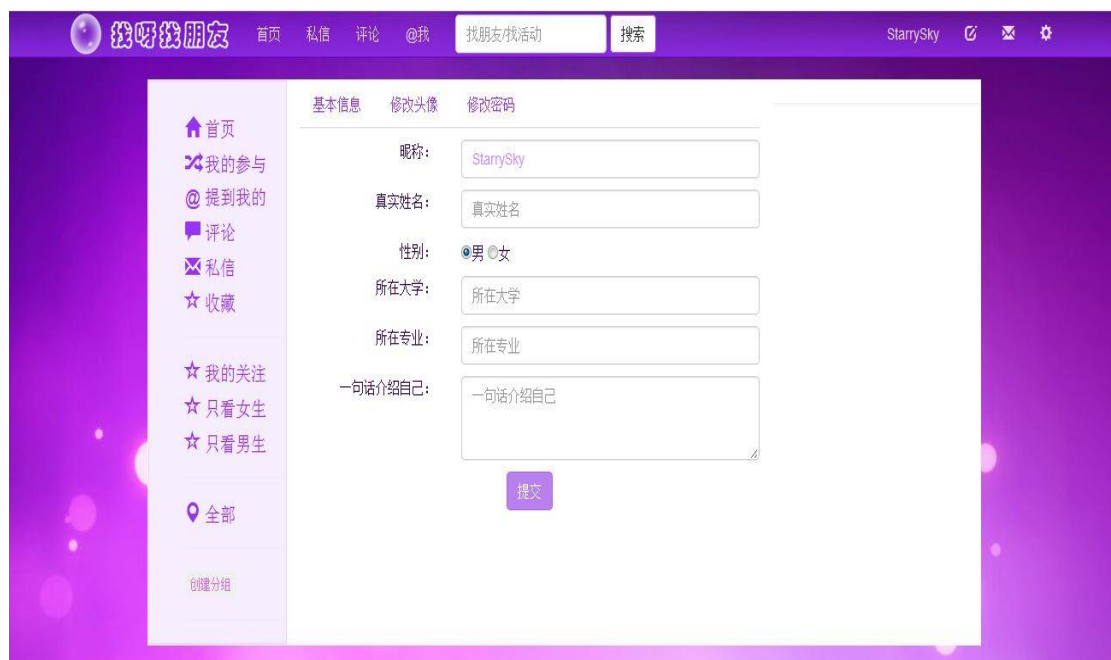


图 5.2.6 修改用户资料



图 5.2.7 发送私信界面



图 5.2.8 更换模板



西南大学

《找朋友动态发布系统》设计报告——罗江涛、李峰、葛广付



图 5.2.9 活动报名



图 5.2.10 后台登陆界面



图 5.2.11 后台登陆信息



图 5.2.12 后台管理界面



图 5.2.13 后台管理功能如：设置过滤关键词

六、团队总结

6.1 开发心得体会

由于现有知识水平的欠缺以及设计时间较短，又追求一定的技术实现，所以该系统还有一些不完善的地方，比如功能实现不够完善、用户界面不够美观、出错处理不够等。这些都有待进一步提高。但是，通过此次数据库应用项目的开发，我们团队通过边学习边讨论，一步一步将整个系统逐步实现，积累了不少技术基础和经验。

值得一提的是，为了让自整个系统的设计过程更具有专业性和可行性，我们自学了许多新的技术和框架，如 Ajax、bootstrap，以求达到预期的效果。并且，在完成整个项目的过程中，我们团队成员更是学到了许多出技术方面之外的东西，如团队协作、相互体谅、相互尊重等，我们也相信这将对我们的以后的学习、生活乃至工作都有着非常积极的指导意义。

6.2 团队合作开发分工情况

罗江涛： 对整体系统的把关，对数据库的分析与设计，前台页面逻辑功能的实现，各种查询功能、前台与后台交互功能的完成等；

李峰： 对数据库的分析与设计，负责后天设计的完成等；

葛广付： 对数据库的分析与设计，负责整个系统的全部页面的设计，美工，留言板的完成，文档的撰写等；

	数据库分析	数据库设计	需求分析及功能模块	代码实现	文档撰写	界面设计
罗江涛	50%	30%	%20	%40	%20	%20
李峰	30%	50%	%30	%40	%20	%20



葛广付	20%	20%	%50	%20	%60	%60
-----	-----	-----	-----	-----	-----	-----

七、参考文献

- 【1】 萨师煊，数据库系统概述，北京：高等教育出版社，2006；
- 【2】 Abraham Silberschatz，Database System Concepts，机械工业出版社，2008；
- 【3】 Ryan Asleson Nathaniel T.Schutta，Ajax 基础教程，人民邮电出版社，2006；
- 【4】 单东林 张晓菲 魏然，锋利的 jQuery，人民邮电出版社，2009；