

实训报告

计算机 161 王泽

目录

| | |
|--------------------------|---|
| 一、 项目背景..... | 1 |
| 1、 项目链接..... | 1 |
| 2、题目的来源..... | 1 |
| 3、后端框架简介..... | 1 |
| 4、前端框架简介..... | 1 |
| 5、开发过程中的参考资料..... | 2 |
| 二、 功能介绍..... | 2 |
| 1、 用户功能..... | 2 |
| 2、 公告功能..... | 3 |
| 三、 数据库设计..... | 3 |
| 1、 think_user 用户表..... | 3 |
| 2、 think_notice 公告表..... | 3 |
| 3、 think_file 文件表..... | 4 |
| 四、 开发前准备..... | 4 |
| 1、 数据库设置..... | 4 |
| 2、 视图输出字符串内容替换..... | 4 |

| | |
|--|----|
| 3、 创建模型..... | 5 |
| 4、 创建验证器..... | 5 |
| 5、 导入 bootstrap..... | 6 |
| 6、 创建前端模板..... | 6 |
| 五、网站后端功能与实现..... | 7 |
| 1、 用户 | 7 |
| 1、 注册功能..... | 7 |
| 2、 登录功能..... | 9 |
| 3、 更改密码功能..... | 10 |
| 4、 退出登录功能..... | 11 |
| 2、公告..... | 12 |
| 1、 发布公告..... | 12 |
| 2、 公告列表..... | 14 |
| 3、 公告详情..... | 14 |
| 4、 删除公告..... | 15 |
| 5、修改公告..... | 16 |
| 如果用户选择重新上传文件则调用 delete_file 删除现有的重新上传新的..... | 17 |
| 六、 网站前端功能与实现..... | 17 |
| 1、用户..... | 17 |
| 1、注册页面..... | 17 |
| 2、登录页面..... | 17 |

| | |
|------------------------------------|----|
| 3、修改密码..... | 18 |
| 4、退出功能..... | 18 |
| 2、公告..... | 19 |
| 1、发布公告..... | 19 |
| 2、公告列表..... | 19 |
| 3、管理公告..... | 20 |
| 七、一些细节..... | 21 |
| 1、替换 success 等跳转参数中的空格..... | 21 |
| 2、文件过大..... | 21 |
| 3、跳转页面优化..... | 21 |
| 4、判断用户登录和权限..... | 22 |
| 5、表单输入体验优化..... | 22 |

一、项目背景

1、项目链接

该网站部署在校内网才能访问的服务器上。

<http://ubuntu.huangdf.com/~jsj16140/notice/index.php>

2、题目的来源

由于 QQ 公告群信息繁杂，班委所发的公告容易被一些没用的消息刷掉，其他群员可能无法及时得知通知内容，导致错过一些重要通知，于是开发此网站解决该问题。

3、后端框架简介

后端框架用到 php 的 ThinkPHP5 框架，ThinkPHP 是一个快速、简单的基于 MVC 和面向对象的轻量级 PHP 开发框架，遵循 Apache2 开源协议发布，从诞生以来一直秉承简洁实用的设计原则，在保持出色的性能和至简的代码同时，尤其注重开发体验和易用性，并且拥有众多的原创功能和特性，为 WEB 应用和 API 开发提供了强有力的支持。

4、前端框架简介

前端框架用到 Web 框架 Bootstrap，Bootstrap 是美国 Twitter 公司的设计师 Mark Otto 和 Jacob Thornton 合作基于 HTML、CSS、

JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它即是由动态 CSS 语言 Less 写成。

5、开发过程中的参考资料

ThinkPHP5.0 完全开发手册

ThinkPHP5 快速入门

[bootstrap 菜鸟教程](#)

[bootstrap 中文网](#)

一些比较杂的问题都是百度

二、功能介绍

1、用户功能

注册功能：用户输入用户名、姓名、密码、职位后注册一个新用户。

登录功能：用户输入用户名、密码和验证码登录，可选择是否记住用户名。

修改密码功能：输入旧密码和新密码进行更改。

2、公告功能

发布公告：输入标题、内容和附加文件，标题不能跟已有的冲突，内容和附加文件都可为空。

修改公告：可更改内容以及附加文件。

公告列表：列表展示所有公告。

公告详情：显示公告的所有信息，图片预览和文件下载。

删除公告：彻底删除公告所有信息，包括上传到服务端的文件。

三、数据库设计

1、think_user 用户表

```
username    # jsj16140，主键
fullname    # 王泽
password    # sha1(12345678)
Job          # '' or '学委'
```

2、think_notice 公告表

```
fullname    # 王泽
job          # '' or '学委'
title        # 领书啦
```

```
content      # balabala

extra        # true

create_at    # 2018-07-04/23:15  公告发布时间
```

3、think_file 文件表

```
title        # 领书啦

file_name     # images.jpg

file_type     # image/word/excel
```

四、开发前准备

1、数据库设置

app/database.php

```
14     'type'                => 'mysql',
15     // 服务器地址
16     'hostname'            => '127.0.0.1',
17     // 数据库名
18     'database'            => 'jsj16140',
19     // 用户名
20     'username'            => 'jsj16140',
21     // 密码
22     'password'            => '12345678',
31     // 数据库表前缀
32     'prefix'              => 'think_',
```

2、视图输出字符串内容替换

设置之后在模板输入路径将变得简单且简洁

app/config.php

```
'view_replace_str'        => [
```

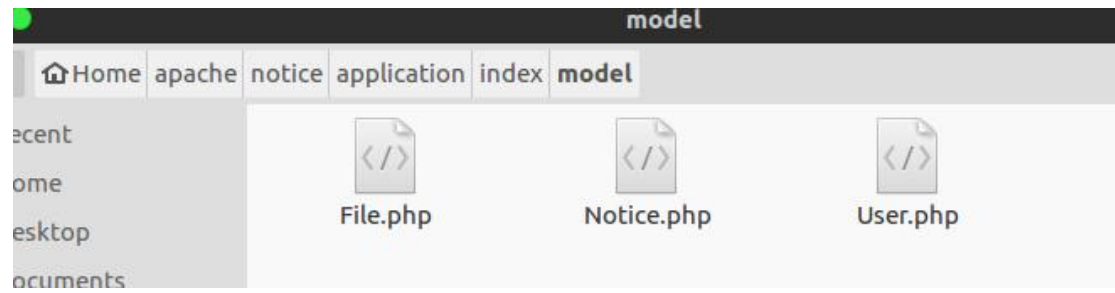


```

143         '__COMMON__' => 'public/static/common',
144         '__BCSS__'    => 'public/static/common/bootstrap-3/css',
145         '__BJS__'     => 'public/static/common/bootstrap-3/js',
146         '__IMAGES__'  => 'public/static/common/images',
147         '__UPLOADS__' => 'public/uploads',
148     ],

```

3、创建模型



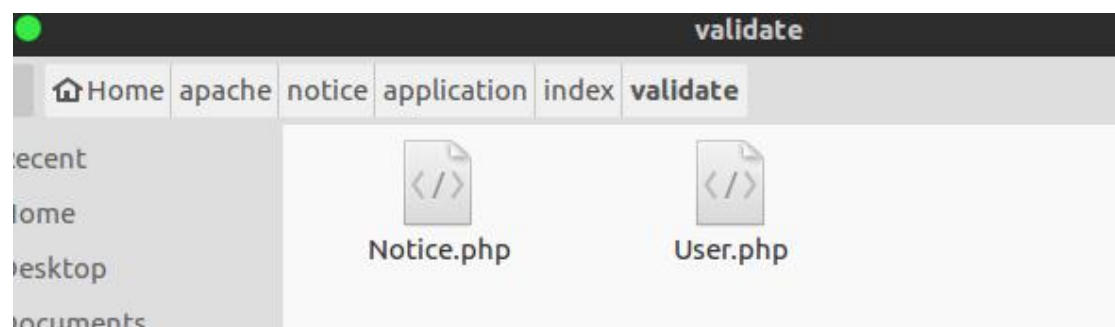
```

3 namespace app\index\model;
4
5 use think\Model;
6
7 class User extends Model
8 {
9     // 设置数据表，包含前缀 think_
10     protected $name = 'user';
11 }

```

通过模型直接操作模型绑定的数据库

4、创建验证器



对于表单数据的验证都通过验证器进行验证

5、导入 bootstrap

```
notice/header.html (view) - Brackets

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta name="description" content="自己做的第一个web网站，使用tp5和bootstrap框架，用于发布 查看和管理班级公告" />
5      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6      <link rel="shortcut icon" href="__ROOT__/__IMAGES__/logo.ico">
7      <title>{$title|default='宝贝请注意'}</title>
8      <script src="__ROOT__/__COMMON__/jquery-3.3.1.min.js"></script>
9      <script src="__ROOT__/__BJS__/bootstrap.js"></script>
10     <link rel="stylesheet" href="__ROOT__/__BCSS__/bootstrap.css">
11
```

6、创建前端模板

将不变的代码分成两部分放在 header.html 和 footer.html 中

宝贝请注意 >> 公告列表

[公告列表](#) [发布公告](#) [管理公告](#) [登录](#) [注册](#)

班级信息

班级：计算机161
人数：58
班主任：
联系方式：

今天适宜看公告

```
notice/deliver.html (view) - Brackets

1  {include file='notice/header' /}
2
3  |
4  {include file='notice/footer' /}
```

编写页面代码时只需修改中间的部分即可

五、网站后端功能与实现

1、用户

在控制器 User 中实现关于用户管理的所有功能

1、注册功能

获取表单输入后验证数据并通过模型写入到数据库中

大部分页面都通过同样的方式跳转

app/index/controller/User.php

```
61     public function register()  
62     {  
63         // 用户注册界面  
64         $this->assign('page', '用户注册');  
65         $this->assign('act_register', 'active');  
66         return $this->fetch();  
67     }
```

```
69     public function do_register(Request $request)  
70     {  
71         // 注册用户  
72         try{  
73             // 实例化模型  
74             $user = new UserModel;  
75             // 获取表单输入并赋值给模型属性  
76             $user->username = $request->param('username');  
77             $user->fullname = $request->param('fullname');  
78             $user->job      = $request->param('job');  
79             $passwd1 = $request->param('passwd');  
80             $passwd2 = $request->param('passwd2');  
81             $passwords = array($passwd1, $passwd2);  
82             $user->password = sha1($passwd1); // 使用 sha1 对密码加密  
83  
84             // 保存用户信息报错前的表单输入  
85             Session::set('username', $user->username);  
86             Session::set('name', $user->fullname);  
87             Session::set('job', $user->job);
```

```

88
89         // 需要验证的数据
90         $data = [
91             'username' => $user->username,
92             'fullname' => $user->fullname,
93             'passwd'    => $passwd1,
94             'passwds'   => $passwds,
95             'user'      => $user->username,
96         ];
97
98         // 注册用户
99
100                                     $result =
101         $user->allowField(true)->validate('User.register')->save($data);
102         if(false === $result){
103             throw new Exception($user->getError());
104         }else{
105             session(null);
106             $this->success('注册成功', 'user/login');
107         }
108     }catch(Exception $e){
109         $this->error($e->getMessage(), 'user/register', '', 1);
110     }
111 }

```

检查密码一致性代码如下 app/index/validate/User.php

```

33     protected function checkPasswd($passwds)
34     {
35         // 检查密码是否一致
36         if($passwds[0] !== $passwds[1]){
37             return '两次输入的密码不一致';
38         }else{
39             return true;
40         }
41     }

```

设置验证规则

```

11     protected $rule = [
12         'username|用户名' => 'require|max:25',
13         'fullname|姓名'   => 'require|max:25',
14         'email'            => 'email',
15         'passwds'          => 'checkPasswd',
16         'passwd|密码'      => 'require|min:6|max:16',
17         'user'             => 'uni_user',
18         'captcha'          => 'require|check_captcha:null',
19         'legal_login'      => 'check_login'

```

```
20 ];
```

设置情景模式

```
27     protected $scene = [  
28         'register' => ['username', 'fullname', 'passwd', 'passwds', 'user'],  
29         'login'    => ['captcha', 'username', 'passwd', 'legal_login'],  
30         'update'   => ['passwd', 'passwds', 'legal_login'],  
31     ];
```

2、登录功能

获取表单输入后验证是否正确，正确则登录成功

app/index/controller/User.php

```
21     public function do_login(Request $request)  
22     {  
23         // 用户登录  
24         $username = $request->param('username');  
25         $passwd    = $request->param('password');  
26         $captcha   = $request->param('captcha');  
27         $record    = $request->param('record');  
28         $user_info = array($username, $passwd);  
29         // 保存用户信息报错前的表单输入  
30         Session::set('login_name', $username);  
31  
32         $data = [  
33             'username' => $username,  
34             'passwd'   => $passwd,  
35             'captcha'  => $captcha,  
36             'legal_login' => $user_info,  
37         ];  
38         // 要在验证数据后面进行加密，否则密码加密后的长度过长  
39         $passwd = sha1($passwd);  
40         try{  
41             $result = $this->validate($data, 'User.login');  
42             if(true !== $result){  
43                 throw new Exception($result);  
44             }  
45  
46             Session::delete('login_name');  
47             $user = UserModel::get($username);  
48             Session::set('username', $username);  
49             Session::set('fullname', $user->fullname);
```

```

50         Session::set('job', $user->job);
51         if($record != ''){
52             cookie('username', $username, 3600*24*7);
53         }
54         $this->success('登录成功', 'notice/index', '', 1);
55     }catch(Exception $e){
56         $this->error($e->getMessage(), 'user/login', '', 1);
57     }
58
59 }

```

检查用户名是否已存在代码如下 `app/index/validate/User.php`

```

43     protected function uni_user($username)
44     {
45         // 检查用户名是否已存在
46         $result = UserModel::get(['username'=>$username]);
47         if($result == ''){
48             return true;
49         }else{
50             return '用户名被用了!';
51         }
52     }

```

验证码输入验证

```

54     protected function check_captcha($value)
55     {
56         $captcha = new Captcha();
57         if($captcha->check($value)){
58             return true;
59         }else{
60             return '验证码错了!';
61         }
62     }

```

3、更改密码功能

用户输入正确的旧密码以及新密码即可更改密码，通过旧密码和当前登录用户名与数据库表格进行比较

`app/index/controller/User.php`

```

129         // 验证数据，表单是否填写正确
130         $data = [
131             'passwd' => $old_passwd,

```

```

132         'passwd' => $new_passwd1,
133         'passwd' => $new_passwd2,
134         'passwds' => $passwds,
135         'legal_login' => $user_info,
136     ];
137     try{
138         $user = UserModel::get($username);
139         $user->password = sha1($new_passwd1);
140         // 验证, 更新密码
141
142         $result =
143         $user->allowField(true)->validate('User.update')->save($data);
144         if(false === $result){
145             throw new Exception($user->getError());
146         }else{
147             session(null);
148             $this->success('修改成功, 请重新登录~', 'user/login', '',
149             1);
150         }
151     }

```

这当然得在登录的状态下进行, 验证代码如下

app/index/validate/User.php

```

64     protected function check_login($user_info)
65     {
66         // 检查用户名和密码是否正确
67         $username = $user_info[0];
68         $passwd = $user_info[1];
69         $passwd = sha1($passwd);
70         $result = UserModel::where('username', $username)
71             ->where('password', $passwd)
72             ->select();
73         if(count($result) <= 0){
74             return '用户名或密码错误!';
75         }
76         return true;
77     }

```

4、退出登录功能

删除所有 session 即可

```

154     public function logout()
155     {
156         // 退出登录
157         session(null);
158         $this->success('退出成功', 'user/login', "", 1);

```

2、公告

在控制器 Notice 中实现关于用户管理的所有功能

1、发布公告

获取表单输入验证通过直接写入数据库，文件表直接存入文件的文件名，在保存和读取时才加上路径

app/index/controller/Notice.php

```

40         $notice = new NoticeModel;
41         $title = trim($request->param('title'));
42         $notice->fullname = session('fullname');
43         $notice->title     = $title;
44         $notice->content  = $request->param('content');
45         $notice->job      = session('job');
46         $files           = $request->file("input_file");
47         // 判断是否有上传文件
48         if(count($files) > 0){
49             $notice->extra = 1;
50         }else{
51             $notice->extra = 0;
52         }
53         // 验证标题
54         $data = [
55             'title'      => $title,
56             'uni_title'  => $title,
57         ];
58         // 验证公告信息并写入数据库
59                                     $result =
$notice->allowField(true)->validate('Notice.deliver')->save($data);
60         // 将文件信息存储到数据库
61         $this->add_file($files, $notice->title);
62         if(false === $result){
63             throw new Exception($notice->getError());
64         }else{
65             $this->success('发布成功', 'notice/index', '', 1);
66         }

```


检查标题是否已存在

app/index/validate/Notice.php

```
26     protected function check_title($title)
27     {
28         // 检查标题是否已存在
29         $result = NoticeModel::get(['title'=>$title]);
30         if($result == ''){
31             return true;
32         }else{
33             return '重复了，换个标题试试';
34         }
35     }
```

上传文件到服务端及存入数据表中

```
200     protected function add_file($files, $title)
201     {
202         // 文件信息列表
203         $file_list = array();
204         foreach($files as $file){
205             $info = $file->move(ROOT_PATH.'public' . DS . 'uploads', '');
206             // 判断文件是否图片，如果是赋值1，如果不是赋值0
207             $type = $this->validate(
208                 ['image' => $info],
209                 ['image' => 'require|image']
210             );
211             if(true !== $type){
212                 $file_type = 0;
213             }else{
214                 $file_type = 1;
215             }
216             $file_name = $info->getSaveName();
217             $file_info = [
218                 'title'      => $title,
219                 'file_name' => $file_name,
220                 'file_type' => $file_type,
221             ];
222             $file_list[] = $file_info;
223         }
224         $file = new FileModel;
225         // 将文件信息存储到数据库
226         $file->saveAll($file_list);
227     }
```

2、公告列表

读取 think_notice 和 think_file 表格所有数据并分页显示
app/index/controller/Notice.php

```
74     $this->check_login_identity();
75     // 管理公告页面
76     $fullname = session('fullname');
77     $job      = session('job');
78     $this->assign('page', '管理公告');
79     $this->assign('act_manage', 'active');
80     // 读取登录用户的所有公告
81     $self_notice = Db::name('notice')->where('fullname', $fullname)
82         ->where('job', $job)
83         ->order('create_at desc')
84         ->paginate(5);
85     $this->assign('list', $self_notice);
86     return $this->fetch();
```

3、公告详情

因为在前端页面中是通过表单三个提交按钮决定对公告执行的操作,为了跳转采用一下方法

app/index/controller/Notice.php

```
89     public function operate(Request $request)
90     {
91         // 判断对公告进行哪种操作
92         $message = $request->param('message');
93         $title = explode('&', $message)[0];
94         // 替换标题中的空格, 跳转传参会将空格换成加号
95         $title = str_replace(' ', '%20', $title);
96         $operate = explode('&', $message)[1];
97
98         if($operate == '查看'){
99             $this->redirect('notice/detail', ['title'=>$title]);
100        }
101        if($operate == '编辑'){
102            $this->redirect('notice/edit', ['title'=>$title]);
103        }
104        if($operate == '删除'){
105            $this->redirect('notice/delete', ['title'=>$title]);
106        }
107    }
```

公告详情，提交按钮绑定了公告标题和操作，通过标题查询对应的公告信息并通过绑定到模板变量进行显示

```
109     public function detail($title)
110     {
111         // 公告详情页面
112         $title = str_replace('%20', ' ', $title);
113         //$result = Db::name('notice')->where('title', $title)->find();
114         $result = NoticeModel::get(['title'=>$title]);
115         // 判断是否有文件
116         if($result['extra']){
117             $files = FileModel::where('title',$title)->order('file_type
desc')->select();
118             $this->assign('extra_file', $files);
119         }
120         $this->assign('msg', $result);
121         $this->assign('page', '公告详情');
122         return $this->fetch();
123     }
```

4、删除公告

删除 notice 和 file 表中相关的所有记录以及上传到服务端的文件

```
170     public function delete($title)
171     {
172         $this->check_login_identity();
173         // 删除公告及公告附件
174         $title = str_replace('%20', ' ', $title);
175         // 读取该标题所有信息
176         $result = NoticeModel::get(['title'=>$title]);
177         // 判断是否有文件
178         if($result['extra']){
179             $is_delete = $this->delete_file($result['title']);
180         }
181         $result = NoticeModel::where('title',$title)->delete();
182         $this->success('公告删除成功', 'notice/manage', '', 1);
183     }
```

删除文件代码

```
185     protected function delete_file($title)
186     {
187         // 删除公告附件
188         $files = FileModel::where('title',$title)->select();
189         foreach($files as $file){
```

```

190         $filename = $file['file_name'];
191         $file = ROOT_PATH . 'public' . DS . 'uploads' . DS . $filename;
192         if(file_exists($file)){
193             unlink($file);
194         }
195     }
196     FileModel::where('title',$title)->delete();
197     clearstatcache();
198 }

```

5、修改公告

跳转到修改页面时将数据库对应的公告信息显示到页面中

```

125     public function edit($title)
126     {
127         $this->check_login_identity();
128         // 编辑公告页面
129         $title = str_replace('%20', ' ', $title);
130         $result = NoticeModel::get(['title'=>$title]);
131         $this->assign('content', $result->content);
132         $this->assign('title', $title);
133         $this->assign('page', '编辑公告');
134         return $this->fetch();
135     }

```

点击确定修改按钮后

```

142         $title = $request->param('title');
143         // 替换标题中的空格，跳转传参会将空格换成加号
144         $str_title = str_replace(' ', '%20', $title);
145         $files = $request->file('input_file');
146         $notice = NoticeModel::get(['title'=>$title]);
147         $notice->content = $request->param('content');
148         // 判断是否要重新上传文件
149         if(count($files) > 0){
150             // 删除原来的文件信息及服务端的文件并写入新的文件
151             if($notice->extra){
152                 $is_delete = $this->delete_file($title);
153             }
154             $this->add_file($files, $title);
155             $notice->extra = 1;
156         }
157
158         $result = $notice->save();

```

```

159         if(false === $result){
160             throw new Exception($notice->getError());
161         }else{
162             $this->success('修改成功', 'notice/manage', '', 1);
163         }

```

如果用户选择重新上传文件则调用 delete_file 删除现有的重新上传新的

六、网站前端功能与实现

1、用户

1、注册页面

使用表单提交数据，没什么特殊的

2、登录页面

也是使用表单，加入了验证吗和记住功能

宝贝请注意 >> 用户登录

公告列表 发布公告 管理公告

登录 注册

用户名: 请输入用户名

密码: 请输入密码

验证码:  请输入验证码

点击图片刷新

☐ 记住我

登录

班级信息

班级: 计算机161

人数: 58

班主任:

联系方式:

今天适宜看公告

验证码需要使用插件 think-captcha

```

22 <div class="form-group">
23   <label for="captcha" class="col-sm-2 control-label">验证码</label>
24   <div class="col-sm-4">
25     
27     <p class="help-block">
28       点击图片刷新
29     </p>
30   </div>
31   <div class="col-sm-6">
32     <input class="form-control" type="text" name="captcha"
33       placeholder="请输入验证码">
34   </div>
35 </div>

```

记住功能是在控制器 User 中设置 cookie

3、修改密码

同样使用表单提交数据修改密码

4、退出功能



通过一个按钮调用

2、公告

1、发布公告

通过表单调用接口将数据存入数据库中

添加更多按钮，使用 js 实现

```
29  { /*  
30      添加更多文件的按钮点击事件函数  
31      这里使用innerHTML  
32      如果使用appendChild会获取不到上传的文件  
33  */  
34  <script>  
35      window.onload=function(){  
36          var add_file = document.getElementById("add_file");  
37          add_file.onclick=function(){  
38              var file = "<input name='input_file[]' type='file' />";  
39              file_form.innerHTML += file;  
40          }  
41      }  
42  </script>
```

2、公告列表

分页显示所有公告

```

1  {include file='notice/header' /}
2
3  {//公告列表展示}
4  <form method="post" action="{:url('notice/operate')}">
5  {//循环输出，name为绑定的变量名，id为别名，不能和name冲突}
6  {volist name='list' id='notice_list'}
7  <div style="margin: 10px">
8      <button class="btn btn-default btn-link btn-block" style="padding-left: 0px"
9      name="message" value="{ $notice_list.title }^&查看" type='submit'>
10         <h4 style="float: left">
11             { $notice_list.title }</h4>
12             <small style="float: right; color: dimgray"><br/> { $notice_list.job }
13             { $notice_list.fullname }</small>
14         </button>
15         <p style="overflow: hidden; white-space: nowrap; text-overflow: ellipsis">
16             &nbsp;&nbsp;&nbsp;& { $notice_list.content }
17         </p>
18         <em>&nbsp;&nbsp;& { $notice_list.create_at } </em>
19         <hr style="height: 1px; border: none; border-top: 1px solid #ADADAD;" />
20     </div>
21 {/volist}
22 <div style="text-align: center">{ $list->render() }</div>
23 </form>
24 {include file='notice/footer' /}

```

3、管理公告

通过表格管理公告

宝贝请注意 >> 管理公告

公告列表

发布公告

管理公告

王泽

| 标题 | 发布时间 | 操作 |
|--------|---------------------|--|
| ubuntu | 2018-07-10 23:09:48 | 查看 编辑 删除 |
| 测试权限 | 2018-07-10 22:47:42 | 查看 编辑 删除 |
| 增加字段了 | 2018-07-10 22:25:25 | 查看 编辑 删除 |
| 有图有字 | 2018-07-10 13:18:21 | 查看 编辑 删除 |
| 两个 | 2018-07-10 00:12:39 | 查看 编辑 删除 |

1

2

39

班级信息

班级：计算机161
人数：58
班主任：
联系方式：

今天适宜看公告

每个按钮都绑定了标题和操作


```

</td>
<td>
    <button class="btn btn-default btn-link" name="message" value="
    {$self_notice.title}^&查看" type='submit'><span class="glyphicon
    glyphicon-check" aria-hidden="true"></span> 查看</button>
</td>
<td>
    <button class="btn btn-default btn-link" name="message" value="
    {$self_notice.title}^&编辑" type='submit'><span class="glyphicon
    glyphicon-edit" aria-hidden="true"></span> 编辑</button>
</td>
<td>
    <button class="btn btn-default btn-link" name="message" value="
    {$self_notice.title}^&删除" type='submit'><span class="glyphicon
    glyphicon-trash" aria-hidden="true"></span> 删除</button>
</td>
</tr>

```

七、一些细节

1、替换 **success** 等跳转参数中的空格

跳转操作会将参数中的空格替换成加号，即使再将加号替换成空格也会出错，采用一下方式解决

| | |
|-----|--|
| 95 | \$title = str_replace(' ', '%20', \$title); // 跳转前 |
| 112 | \$title = str_replace('%20', ' ', \$title); // 跳转后取出 |

2、文件过大

表单提交的超过 2M 的文件在 php 文件获取不到，需要修改 php.ini

3、跳转页面优化

修改 app/thinkphp/tp1/dispatch_jump.tpl 即可修改跳转页面样式

4、判断用户登录和权限

在发布公告、修改公告等操作函数最前面加上验证函数

```
229     protected function check_login_identity()  
230     {  
231         // 检查用户是否已登录以及有无权限进行操作  
232         if(!session('?fullname')){  
233             $this->error('请先登录', 'user/login', '', 1);  
234         }  
235         if(session('job') == ''){  
236             $this->error('对不起，你没有权限', 'notice/index', '', 1);  
237         }  
238     }
```

5、表单输入体验优化

如果表单填写错误则记录除密码意外的输入框，避免用户重复操作