

《万卷书成长网后台会员管理》检测报告

PaperOK

总相似率：33.43%

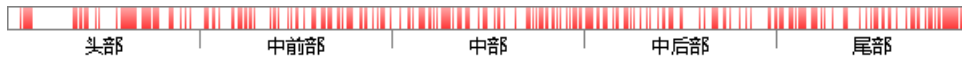
基本信息

文档名称	万卷书成长网后台会员管理
报告编号	7b4146ea-b55f-4aa6-a59c-94ff1ccf3bcf
文档字数	24972
提交人姓名	王岁洋
提交方式	粘贴文本检测
检测范围	学位论文库（含硕博）、学术期刊库、会议论文库、法律法规库、互联网资源库、自建比对库
提交时间	2018-03-26 16:46:09

检测报告指标详情

原创率	抄袭率	引用率	字数统计	参考文献字数
66.57%	33.43%	0%	24972	-

相似片段位置图



注：红色部分为论文相似部分，蓝色部分为论文引用部分

相似片段详情（仅显示前10条）

序号	篇名	来源	相似率
1	tp5的模型必须放在common目录下吗_百度知道	互联网资源库	1.7%
2	2017新手学习框架必备:Thinkphp5框架教程之命名空间详解-PHP头条-...	互联网资源库	0.94%
3	输入变量·ThinkPHP5.0完全开发手册·看云	互联网资源库	0.89%
4	基于WebGIS的卫星观测计划编制系统	学位论文库	0.74%
5	短信服务 REST API 详解 - LeanCloud 文档	互联网资源库	0.68%
6	关于session存储用户信息和验证码(短信,邮箱之类的)的细节问题	互联网资源库	0.68%
7	流媒体服务器Red5的扩展设计、测试与优化	学位论文库	0.61%
8	时间戳·ThinkPHP5.0完全开发手册·看云	互联网资源库	0.58%
9	请求信息·ThinkPHP5.0完全开发手册·看云	互联网资源库	0.54%
10	一种新型的组态软件Web发布系统的设计	学术期刊库	0.43%

文档原文标注

毕业设计（论文）

万卷书成长网后台会员管理

学 院： 电子与物联网学院

专 业： 网络工程

学生姓名： 王岁洋 学号： 149100327

指导教师： 万川梅 职称： 副教授

（ 任务起止日期 2017 年 11 月 1 日至 2018 年 5 月 30 日 ）

系（ 教研室 ）主 任： 万川梅 2017 年 11 月 1 日

摘要

信息处理已成为当今世界上一项重要的社会活动。随着21世纪国家改革开放的不断深入，市场经济建设的不断完善以及社会的飞速发展，用户管理系统模式正迅猛发展并深入到各行各业中。

??在这一新的形势下，要求程序设计人员能根据不同行业、不同需求的特点来快速、简捷并且准确按照需求分析来编写软件。该系统改变了传统管理的办法，顺应了办公自动化、现代化的潮流。

该系统极大丰富了管理功能，集用户管理、权限管理等功能于一体。实现了数据录入、数据查询、数据汇总以及报表输出的功能。一方面方便了用户掌握自己的基本信息、历史纪录以及变动情况；另一方面也使网络管理者更加准确、方便和全面的掌握用户的信息。

网络管理员可以随时添加、删除和修改用户信息，及时对用户的注册情况进行掌握，从而可以及时统计客户数量，做到灵活机动的管理。

关键字：thinkphp5、管理、函数、请求

目录

摘要.....2

第一章绪论

1.1项目背景.....4

1.2需求分析.....4

1.3 主要模块.....5

第二章 登录模块

2.1普通登录.....6

2.2短信验证登录.....6

第三章公共模块

3.1权限控制.....7

3.2操作日志.....8

第四章用户模块

4.1用户管理.....8

4.1角色管理.....11

4.1权限管理.....12

第五章资讯模块

5.1咨询管理.....14

5.1栏目管理.....16

第六章基础数据模块

6.1日志管理.....18

6.1节点管理.....19

第七章 总结.....20

参考文献.....21

致谢.....21

第一章 绪论

一、项目背景

（一）基本概念、理论或应用背景：

当代社会的各行各业已经走向了信息化的时代，当然教育行业自然也不例外。随着学生人数的逐渐增长，传统教育的改变已经变得刻不容缓。所以我们的这个万卷书成长网教育平台针对当前学生的学习要求而量身订制。最主要的特色有：亲子结伴（让学生与家长一起学习）、话题互动（调动学生们的积极热情）、家教之窗（让学生们找到好老师，让家长省去时间）、升学资讯（时刻了解最新的学校资讯、学业资讯）、冬夏令营（在努力学习后的寒暑假有意义的玩耍一翻）、海外研学。

“万卷书成长网”就是为家长对孩子的素质教育的以一个信息平台。其中为孩子们提供了各种各样的户外活动，以培养孩子的动手能力和团队协作能力。让孩子在活动中学习到在校园学习不到的知识。而一个网站的建立就缺少不了对会员的管理，本次论文的核心就是围绕“万卷书成长网”的会员的管理而展开。

（二）发展趋势

当今社会随着人们生活条件，经济的日益增长，对青少年的教育，培养也越加重视。“万卷书成长网”的发展空间良好。随着青少年的素质教育的重视，“万卷书成长网”将会迎来越来越多的客户，而“万卷书成长网”的会员管理模块将成为该网站重中之重。

二、需求分析

“用户管理”模块中的用户管理，角色管理，权限管理；‘资讯管理’模块中完成：资讯管理，栏目管理；‘基础数据管理’模块中日志管理和节点管理等功能的实现。

1.概要设计分析

?总述：该项目针对当前的教育形势给予学生、家长们一个好的教育沟通桥梁，让学生、家长能通过这个平台更方便的学习、交流。

?总体设计：从全局上来看该平台的作用与天猫、淘宝等购物网站相似，只不过它只提供教育活动及教育相关信息。

?模块设计：主要有用户管理、资讯管理、基础数据管理

?总体设计方案：

1、会员管理需求分析

2、逻辑结构设计流程草图

3、设计用到的变量名（采用驼峰命名法）

4、设计具有扩张性的接口，为二次开发预留接口

2.环境搭建

?后台：采用了thinkphp5+mysql数据库结合的方式编写。运用的主要编程软件有phpstudy（Windows下的Apache+MySQL+PHP5.6集成环境）、sublime。

?前端：采用layer弹窗，hui-admin前端框架

三、主要模块

（一）登陆模块

1、普通登录（通过用户名和密码登录到后台）

2、短信验证登录（通过手机号和验证码登录到后台）

（二）公共模块

1、权限控制函数（控制用户权限的方法）

2、操作日志函数（记录用户操作的方法）

3、是否已登录的函数（判断用户是否已在线，防止重复登录和跳过登录通过url访问后台主页）

（三）用户管理模块

1、用户管理（实现对用户增，删，改，查）

2、角色管理（实现对用户进行角色的分类，对角色增，删，改，查）

3、权限管理（实现对角色及用户的操作权限的分配）

（四）资讯管理模块

1、资讯管理（实现对资讯信息增，删，改，查，审核，发布，下架）

2、栏目管理（实现对资讯的栏目名管理，对栏目的增，删，改，查）

（五）基础数据管理模块

1、日志管理（实现对操作方法的记录）

2、节点管理（实现对权限节点的管理）

第二章 登录模块

主要是用户的登录，Session方法存储用户的信息，用于当前用户信息展示。

1.普通登录

（1）模型（user.php）

使用的用户模块中的user.php模型。

（2）控制器(login.php)

namespace app\login\controller语句声明根命名空间，use think\Session声明thinkphp官方Session类命名空间，use think\Validate声明验证类命名空间，use think\Request声明thinkphp官方Request类命名空间，use app\user\model\User申明用户模型命名空间，use app\user\model\Role;申明角色模型命名空间。

Index函数首先检查是否已有用户登录，已登录跳转到后台，否渲染普通登录的视图界面。

Loginvalidate函数，获取前端界面传递的数据赋值给\$data.设置状态值\$status为0，使用thinkphp5.0自带验证方法验证用户数据是否符合规则。不符合返回对应的消息和状态。符合就查询用户名和密码对应的用户并且该用户必须在启用的状态下才能登陆成功。登陆成功后Session函数存储用户的信息，更新用户的登陆时间，设置\$status的值为1，并返回消息登陆成功和状态值。

（3）视图(index.html)

普通登陆的视图，有跳转短信登录的链接。

2.短信验证登录

（1）模型

使用的用户模块中的user.php模型。

（2）控制器（login.php）

namespace app\login\controller语句声明根命名空间，use think\Session声明thinkphp官方Session类命名空间，use think\Validate声明验证类命名空间，use think\Request声明

thinkphp官方Request类命名空间，use app\user\model\User申明用户模型命名空间，use app\user\model\Role;申明角色模型命名空间。use app\login\controller\SmsSingleSender;声明腾讯云的SDK类的命名空间。

sms_login函数用于渲染短信验证登录界面和验证短信验证码和匹配手机号，首先渲染短信验证登录界面，当检测到有post请求时获取发送的验证码于提交的验证码比较，相同时查询提交的手机号码的用户信息，当验证码匹配且数据库有该手机号，Session函数存储用户的信息，更新用户的登陆时间，设置\$status的值为1，并返回消息登陆成功和状态值。

sms_send函数用于腾讯云短信的发送，开通腾讯云服务短信业务，获得短信应用SDK AppID和短信应用SDK AppKey。Rand函数随机生成验证码，调用SDK类中的 sendWithParam函数将SDK AppID，SDK AppKey，验证码，手机号传递到短信云平台上，然后接收发送的返回值判断成功与否，返回对应的消息和状态。

(3) 视图 (sms_login.html)

短信验证登录视图。

第三章 公共模块

公共模块中主要有权限控制，用户登录状态的检查，操作日志的生成。位于thinkphp5框架自定义应用目录admin中的com模块目录。

1 . 权限的控制

controller目录中的Accesscontrol.php控制器中实现对权限的控制。首先namespace app\user\controller语句声明根命名空间,use think\Controller声明thinkphp5的官方控制器命名空间，用于后期继承thinkphp5控制器类。use think\Session声明thinkphp5的官方Session命名空间，用于后期调用Session类中的函数过去用户的信息use think\Request声明thinkphp5的官方Request类的命名空间，用于后期获取请求。

class Accesscontrol extends Controller语句定义Accesscontrol 类并继承thinkphp5的控制器类。public \$powerid定义权限id;protected function _initialize()定义Accesscontrol 控制器初始化方法，\$powerid=Session::get('role_info.powerid')获取用户所属角色的所有权限id。因为role数据表的powerid储存为权限id的字符串的集合，所以使用explode函数将其拆分为数组并赋值给Accesscontrol 类中的公有变量：\$powerid。

public function control(\$control_id)定义控制函数并有一个形参\$control_id，\$request=Request::instance()初始化请求类并赋值给\$request。if(!in_array(\$control_id,\$this->powerid))判断传递的实参是否在Accesscontrol 类中的公有变量：\$powerid中，存在那么将拥有权限，否则没有权限。当不存在时，调用前面初始化请求类中的isajax()函数判断是否为ajax请求，是则\$message="无权访问";return \$message;返回数据信息无权访问。否则 echo "无权访问" 直接输出无权访问。

2 . 用户登录状态的检查

controller目录中的base.php控制器中实现对用户登录状态的检查。首先namespace app\user\controller语句声明根命名空间,use think\Controller声明thinkphp5的官方控制器命名空间。use think\Session声明thinkphp5的官方Session类命名空间。class Base extends Controller定义Base控制器并继承thinkphp5的控制器类。protected function _initialize()定义Base控制器初始化方法，define('USER_ID', session::get('user_info.id'));获取用户信息中的id值设为USER_ID常量。

定义公有函数unlogin，通过if(empty(USER_ID))语句判断常量USER_ID是否存在，从而判断用户是否登录，没有登陆则\$this->error('用户未登录，无权访问',url('login/login/index'))跳转登录界面。定义公有函数islogin，通过if(empty(USER_ID))语句判断常量USER_ID是否存在，从而判断用户是否登录，当进入登录界面调用该函数如果session存储的用户信息没有过期，则直接跳转\$this->success('用户已登陆,请勿重复登陆。',url('com/index/index'));后台主页。

3. 操作日志的生成

生成日志方法，在Accesscontrol控制器中，声明空间及与权限控制相同，定义公共addlog函数用于操作日志的生成并带有一个形参\$operation，表示操作内容。首先判断\$operation是否存在，不存在结束该函数，存在则将用户名，姓名，操作内容添加到日志数据表中。

第四章 用户管理模块

用户管理模块分为三大功能模块：用户管理，角色管理，权限管理。分别对应用户表，角色表，权限表。用户管理模块对应thinkphp5框架自定义应用目录admin中的user模块目录。用户管理中主要是实现对用户的添加，修改，查询等。角色管理主要作用适用于用户的分类，如该系统中的会员用户与管理用户分别为不同类别的角色。不同的角色将赋予不同的权限，角色中的用户将拥有该角色对应的所有权限。权限管理就是赋予不同角色不同的权限。

1. 用户管理

用户管理对应的是user模块目录下controller目录中的User.php(控制器);model目录下的user.php(模型);view目录下的user目录(用户视图集合)。

(1) 模型 (user.php)

namespace app\user\model语句声明根命名空间，use think\Model声明thinkphp官方model类命名空间。

class Role extends Model定义Role类并继承Model类。protected \$updateTime = false定义受保护的变量updateTime并赋值false，表示关闭自动写入'update_time'字段；protected \$createTime = 'create_roletime'定义受保护的变量'createTime'并赋值create_roletime，表示设置自动写入时间戳字段名。

(2) 控制器 (user.php)

namespace app\user\controller语句声明根命名空间,首先namespace app\user\controller语句声明根命名空间,use think\Controller声明thinkphp5的官方控制器命名空间，用于后期继承thinkphp5控制器类。use think\Request声明thinkphp5的官方Request类的命名空间，用于后期获取请求。use app\com\controller\Accesscontrol申明公共模块的权限控制类的命名空间。use app\user\model\User as UserModel声明user模型的命名空间；use app\user\model\Role声明role模型的命名空间。use think\Db声明thinkphp5官方Db类，用于后面对数据库的操作。

class User extends Accesscontrol定义user类并继承控制器。定义公有函数index，调用公共函数control，传入一个实参id（对应power表中的用户查询的id），判断是否拥有权限查看user信息列表。\$list=Db::table('user')->alias('a')->join('role w','a.roleid = w.role_id')->paginate(5)使用Db类的链式操作方法链接查询user，role数据表。paginate(5)表示每5条数据进行分页。\$count=UserModel::count()调用模型的count方法，查询user记录数。使用assign方法将查询的值：\$list,\$count模板赋值给视图。最后调用fetch函数渲染视图界面。

公共status方法中首先进行权限控制，调用公共函数control，传入一个实参id（对应power表中的用户启用/停用的id），返回值赋给\$message。判断\$message是否空，而判断是否拥有权限启用/停用用户。拥有则\$request = Request::instance()初始化请求类并赋值给\$request。\$id = \$request->param('user_id')获取前端ajax传递的user_id，\$status=UserModel::where('user_id',\$id)->value('status')获得id中的状态值（1或0）。\$status=(\$status==1)?0:1改变状态值，\$rule=UserModel::where(['user_id'=>\$id])->update(['status'=>\$status])更改user数据表中的状态值，返回结果赋值给\$rule。查询对应\$id的用户名赋值给\$user，判断\$status的值0为禁用。1为启用赋值给\$info，调用公共日志函数addlog，传入实参:\$info加\$user。\$message=(\$rule===null)?"操作失败":"操作成功"，判断\$rule的值进行判断更新数据表结果是否成功。return ['message'=>\$message];返回执行结果。没有权限则返回\$message。

公共adduser方法中查询数据表role的所有数据，首先进行权限控制，调用公共函数control，传入一个实参id（对应power表中的用户添加的id），判断是否拥有权限添加用户，拥有权限则assign模板赋值给视图adduser.html。fetch函数渲染视图。

公共usersave方法用于接收adduser传递的表单数据。初始化状态值为空，\$request = Request::instance()初始化请求类并赋值给\$request。\$data = \$request->param()获取前端ajax传递的表单数据。设置验证规则：username为必填项，长度为2到17，并且只能为字母和数字的集合；name必填项，长度为2到12；password为必填项，长度为3到17，且只能是数字，字母，_@~!的集合；角色类型必选，赋值给变量\$rule。定义错误信息赋值给\$msg。\$message=\$this->validate(\$data,\$rules,\$msg)调用validate函数执行验证。

当验证结果\$message为true时，调用静态模型中的get方法查询user表中是否已存在当前添加的用户名，存在则赋值变量\$message="用户名已存在。"，调用公共日志函数addlog，传入实参:'添加用户:'.\$data['usercode'],'结果:'.\$message。然后返回错误结果和状态。不存在则通过验证将表单数据中的面进行加密md5(\$data['username'].\$data['password'].'~!@)。然后create函数将数据添加到user表中，返回结果赋值给\$user。通过判断\$user是否存在，而确定是否添加成功返回对应的消息与状态值。最后调用公共日志函数addlog，传入实参:'添加用户:'.\$data['usercode'],'结果:'.\$message，然后返回消息和状态值。

公共useredit方法用于用户的修改，首先进行权限控制，调用公共函数control，传入一个实参id（对应power表中的用户修改的id），判断是否拥有权限修改用户信息。\$request = Request::instance()初始化请求类并赋值给\$request。if(\$request->ispost())判断前端是否有post请求，是则执行同用户添加除密码以外的验证，验证通过后调用静态模型中的get方法查询user表中是否已存在当前修改的用户名，存在则赋值变量\$message="用户名已存在。"，调用公共日志函数addlog，传入实参:'修改用户:'.\$data['usercode'],'结果:'.\$message，然后返回错误结果和状态。不存在将更新数据使用update方法写入user数据表中，然后判断更新语句是否成功，赋值给添加成功返回对应的消息与状态值变量。最后调用公共日志函数addlog，传入实参:'添加用户:'.\$data['usercode'],'结果:'.\$message，然后返回消息和状态值。不存在post请求则使用\$id = \$request->param('user_id')获取前端ajax传递用户id赋值给\$id，\$list=UserModel::get(\$id)根据id值查询该用户的信息并连同查询用户角色列表使用ssgin模板赋值给视图useredit.html。fetch函数渲染视图。

公共userdel方法用于用户的删除，但用户关联性太广，所以该操作通常只有超级管理员才拥有。首先进行权限控制，调用公共函数control，传入一个实参id（对应power表中的用户删除的id），并把返回值付给\$message。判断\$message是否为空，空则拥有权限删除用户信息。根据前端ajax传递的id值赋值给\$id，使用模型destroy函数删除\$id的用户。判断删除结果，调用日志函数\$this->addlog('删除用户:'.\$user,'结果:'.\$message);添加日志，最后从而返回对应的消息。否则返回\$message。

公共deleteuser函数，用户的批量删除，根据前端ajax传递的id值集合的数组，使用for语句循环执行模型的destroy函数，当删除成功则执行下一个循环，失败则中断循环返回该出错id值，循环

结束后调用 `$this->addlog('批量删除用户，结果：'.$message);`添加日志，最后全部成功则返回删除成功等信息。

公共searchuser函数用于接收前端get请求传输的关键字执行搜索操作。首先进行权限控制，调用公共函数control，传入一个实参id（对应power表中的用户搜索的id），判断是否拥有权限搜索用户信息。使用`$retuls=Db::table('user')->alias('a')->join('role w','a.roleid = w.role_id')->where("usercode like '%$value%' or username like '%$value%')->paginate(10);`语句模糊查找关键字信息。`$count = Db::table('user')->where("usercode like '%$value%' or username like '%$value%')->count();`查询记录数。调用日志函数`$this->addlog('搜索用户关键信息：'.$value);`添加日志记录。`$retuls`和`$count`一起ssgin模板赋值给视图index.html，fetch函数渲染视图index.html。

公共changepassword方法用于用户密码的更改。调用公共函数control，传入一个实参id（对应power表中的用户密码的更改的id），判断是否拥有权限修改密码。`$request = Request::instance()`初始化请求类并赋值给`$request`。`if($request->ispost())`判断前端是否有post请求，是则设置状态值为0，param方法获取请求参数，判断输入的旧密码是否正确，错误则`$this->addlog('修改用户：'.$data['usercode'].'密码，'.'结果：'.$message);`添加错误日志，返回消息旧密码错误。正确则将新密码与用户名进行md5加密后更新至数据表。更新成功后则将状态值改为1，消息则返回密码更新成功。将成功的操作添加到日志表`$this->addlog('修改用户：'.$data['usercode'].'密码，'.'结果：'.$message);`没有post请求获取前端传递id值，查询该id的用户信息，将用户信息使用assgin模板赋值给视图changepassword.html。fetch函数渲染视图。

（3）视图（user目录）

?Index.html

用于显示用户列表，及增.删.改等操作按钮的显示。Layer弹窗插件进行弹层。

?Adduser.html

用于添加用户界面的显示，Ajax传递表单信息到后台，回调函数输出执行结果

?Useredit.html

用于修改用户界面的显示，并当前修改用户的信息为文本框初始值。Ajax传递表单信息到后台，回调函数输出执行结果

?Changepassword.html

用于密码的修改，并显当前修改用户名。Ajax传递表单信息到后台，回调函数输出执行结果。

2.角色管理

（1）模型(role.php)

`namespace app\user\model;`语句声明根命名空间use think\Model;声明thinkphp5的官方模型命名空间，用于后期继承model类。`class Role extends Model`定义role类并继承model。定义受保护的`protected $updateTime = false;`和`protected $createTime = 'create_roletime';`分表代表关闭自动写入update_time字段和更改时间戳字段名。

（2）控制器(role.php)

首先namespace app\user\controller语句声明根命名空间。use think\Request声明thinkphp5的官方Request类的命名空间，用于后期获取请求。use app\com\controller\Accesscontrol申明公共模块的权限控制类的命名空间。use app\user\model\User as UserModels声明user模型的命名空间取别名为UserModels；use app\user\model\Role as RoleModel;声明role模型的命名空间取别名为RoleModels。

class Role extends Accesscontrol定义Role控制器并继承Accesscontrol类。创建公共index函数，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中的角色查看的id），判断是否拥有权限查看角色信息。用模型的所有和count函数完成角色表的查询和记录数的统计。在使用assign函数模板赋值给视图index.html。fetch函数渲染视图。

公共roleadd方法先进行权限控制，调用公共函数control，传入一个实参id（对应power表中的角色添加的id），判断是否拥有权限添加信息，有权限则使用fetch方法对角色添加界面roleadd.html的渲染，公共rolesave方法用于对角色添加信息的保存。\$request = Request::instance()初始化请求类并赋值给\$request，设置状态值为0，调用请求中的param方法获取所有表单数据赋值给\$data。查找前端的传递角色名在数据库中的id值赋值给\$id。判断是否存在\$data['id']存在则是角色更新的保存，否则使角色添加的保存。更新角色需\$id存在并且不等于\$data['id']，则将\$data中的数据更新到\$data['id']的记录中，在使用日志函数\$this->addlog('修改角色：'.\$data['roleName'].'，'.'结果：'.\$message)添加日志。并返回更新数据成功或失败的对应消息和状态值。否则返回消息角色名重复。添加角色需判断\$id是否存在，存在则返回返回消息角色名重复，不存在则将\$data中的数据添加到角色表中，调用公共日志函数将该操作日志添加进数据表，返回添加结果对应的消息和状态值。

公共status函数用于角色的停用/启用，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中角色的停用/启用的id）返回值赋给\$message，判断\$message是否为空从而判断是否拥有权限修改用户状态。为空则\$request = Request::instance()初始化请求类并赋值给\$request。\$id = \$request->param('id')获取前端ajax传递的角色id。查询用户表中是否存在该角色的用户，存在则返回消息：请清空角色中的用户后，才能停用。不存在则获取该\$id对应的角色信息，然后将其状态值取反，然后更新至数据库。获取\$id对应的角色状态，如果为0则赋值\$info为启用，反之为禁用。使用日志函数\$this->addlog(\$info.'角色：'.\$role.'，'.'结果：'.\$message)添加操作日志，返回更新结果对应的消息。

公共roledel方法用于角色名的删除，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中角色删除的id）返回值赋给\$message，判断\$message是否为空从而判断是否拥有权限删除角色。为空则根据前端ajax传递的id值赋值给\$id，使用模型destroy函数删除\$id的角色。查询\$id对应的角色名，执行日志函数\$this->addlog('删除角色：'.\$role.'，'.'结果：'.\$message);添加日志。最后判断删除结果，从而返回对应的消息。

公共deleteuser函数，角色的批量删除，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中角色的批量删除的id）返回值赋给\$message，判断\$message是否为空从而判断是否拥有权限批量删除角色。为空则根据前端ajax传递的id值集合的数组，使用for语句循环执行查询语句查询当前id的角色名并连接到\$role和模型的destroy函数，当删除成功则执行下一个循环，失败则中断循环返回该出错id值，日志函数\$this->addlog('批量删除角色：'.\$role.'，'.'结果：'.\$message)添加日志;最后全部成功则返回删除成功等信息。

（3）视图(role目录)

?Index.html

用于显示用户列表，及增.删.改.等操作按钮的显示。Layer弹窗插件进行弹层。

?roleadd.html

用于添加角色界面的显示，Ajax传递表单信息到后台，回调函数输出执行结果

?Useredit.html

用于修改用户角色的显示，并当前修改用户的信息为文本框初始值。Ajax传递表单信息到后台，回调函数输出执行结果

3.权限管理

(1) 模型 (Power.php\Module.php\Controller.php)

namespace app\system\model语句声明根命名空间，use think\Model声明thinkphp官方model类命名空间。

class Power extends Model定义Power类并继承Model类。class Module extends Model定义Module类并继承Model类。class Controller extends Model定义Controller类并继承Model类。protected \$autoWriteTimestamp = false;定义受保护的变量autoWriteTimestamp 并赋值false，表示关闭自动写入时间戳。

(2) 控制器 (power.php)

首先namespace app\user\controller语句声明根命名空间,use app\com\controller\Accesscontrol;申明公共模块的权限控制类的命名空间,

use think\Request;声明thinkphp5的官方Request类的命名空间，用于后期获取请求。use app\user\model\Role;声明role模型的命名空间;use app\system\model\Power as PowerModel;声明Power模型的命名空间区别名为PowerModel；use app\system\model\Module;声明Module模型的命名空间。use app\system\model\Controller as ControllerModel;声明Controller模型的命名空间区别名为ControllerModel。

class Power extends Accesscontrol定义Power类并继承Accesscontrol。

公共index函数，用于角色信息的展示。模型all函数查询角色信息，assign函数模板赋值给视图index.html。fetch函数渲染视图。

公共poweredit方法用与角色权限的树形展示和修改。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中权限的展示id），通过则\$request = Request::instance()初始化请求类并赋值给\$request。\$id = \$request->param('id')获取前端ajax传递的角色id。根据id值获取当前角色对应的权限id值的字符串。使用explode函数将其拆分为数组赋值给\$arr_power。定义一个空数组\$html,查询Module，ControllerModel，PowerModel的所有信息，使用如图1所示代码三层嵌套foreach循环和array_push函数，形成一个三维数组。json_encode函数将该数组转为json格式后使用assign将其模板赋值给视图poweredit.html。fetch函数渲染视图。

公共addpower方法权限的保存操作。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中权限添加的id）返回值赋给\$message，判断\$message是否为空从而判断是否拥有权限修改权限。为空则\$request=Request::instance()初始化请求类并赋值给\$request，定义一个空变量\$id,请求的param函数获取角色id值赋值给\$role_id，input将获取所有末端节点值并转为数组。Foreach和explode相结合将所有权限id值集合到一起为字符串赋值给\$id,再将\$id更新到\$role_id这条记录中。查询\$id对应的角色名，使用日志函数\$this->addlog('修改角色：'. \$role.'的权限')添加日志，最后返回对应的消息与状态值。

图1:

(3) 视图

?index.html

用于显示角色信息，权限管理等操作。

?poweredit.html

用于对角色权限的管理。

第五章 升学资讯

升学资讯模块分为二大功能模块：栏目管理，资讯管理。分别对应用栏目表，资讯表。升学资讯模块对应thinkphp5框架自定义应用目录admin中的info模块目录。栏目管理中主要是实现对栏目的添加，修改，查询等。栏目主要作用使用于资讯的分类。资讯管理实现对资讯的管理，主要是升学资讯，如小学升初中，初中升高中等资讯。

1.资讯管理

(1) 模型 (P_info.php)

namespace app\info\model语句声明根命名空间，use think\Model声明thinkphp官方model类命名空间。

class Info_type extends Model定义Power类并继承Model类。class P_info extends Model定义P_info 类并继承Model类。protected \$updateTime= false;定义受保护的变量updateTime并赋值false，表示关闭自动写入更新时间戳。

(2) 控制器(info.php)

首先namespace app\info\controller语句声明根命名空间,use app\com\controller\Accesscontrol;申明公共模块的权限控制类的命名空间,

use think\Request;声明thinkphp5的官方Request类的命名空间，用于后期获取请求。use app\info\model\info_type;声明info_type模型的命名空间;use app\info\model\P_info as InfoModel;声明P_info模型的命名空间取别名为InfoModel。

class Info extends Accesscontrol定义Info类并继承Accesscontrol。公共index函数，用于资讯信息的展示。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中资讯查询的id），判断是否拥有权限查看资讯信息，有权限则模型all函数查询所有的资讯信息和启用的栏目，count函数查询记录数，assign函数模板赋值给视图index.html。fetch函数渲染视图。

公共infoadd函数，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中资讯添加的id），判断是否拥有权限添加资讯信息，有权限则\$request = Request::instance()初始化请求类并赋值给\$request，判断是否有post请求，如果没有则渲染infoadd.html视图。有则设置状态值为空，调用请求中的param方法获取所有表单数据赋值给\$data。查询传递的资讯题目是否存在，存在则返回资讯名已存在，不存在则将\$data中的数据添加到对应的字段到栏目数据表。判断添加是否成功，返回对一个的状态值与消息。

公共info函数查看资讯详细信息，先进行权限控制，调用公共函数control，传入一个实参

id (对应power表中资讯详查的id) , 判断是否拥有权限查看资讯详情, 有权限则\$request = Request::instance()初始化请求类并赋值给\$request, 请求param函数获取请求的id值, 查询该id的资讯内容。sign函数模板赋值给视图info.html。fetch函数渲染视图。

公共infodel方法用于资讯的删除, 首先进行权限控制调用公共函数control, 传入一个实参id (对应power表中资讯删除的id) 返回值赋给\$message, 判断\$message是否为空从而判断是否拥有权限删除资讯。为空则根据id值使用模型destroy函数删除对应的资讯。判断删除结果, 从而返回对应的消息和状态。

公共status方法用于资讯的审核, 先进行权限控制, 调用公共函数control, 传入一个实参id (对应power表中资讯审核的id) 返回值赋给\$message, 判断\$message是否为空从而判断是否拥有权限审核资讯。为空则权限通过, \$request = Request::instance()初始化请求类并赋值给\$request请求param函数获取请求的状态值, 获取当前用户为审核人, 当前时间为审核时间, 并一起更新至数据库, 然后调用日志函数, 将该操作写入日志, 然后return返回结果。

公共article_status方法用于资讯状态的修改, 先进行权限控制, 调用公共函数control, 传入一个实参id (对应power表中资讯状态的修改的id) 返回值赋给\$message, 判断\$message是否为空从而判断是否拥有权限修改资讯的状态。为空则权限通过\$request = Request::instance()初始化请求类并赋值给\$request请求param函数获取请求的数据, 获得id值和状态值并一起更新至数据库, switch语句判断id值, 0-下架; 1-申请上线; 2-发布。然后调用日志函数, 将该操作 (下架, 申请上线, 发布) 写入日志, 然后return返回结果。

公共方法infoedit用于资讯的修改, 先进行权限控制, 调用公共函数control, 传入一个实参id (对应power表中资讯状态修改的id) , 有权限则\$request = Request::instance()初始化请求类并赋值给\$request, 调用请求的ispost方法判断是否有post请求, 有则param获取所有请求数据赋值给\$data, \$id=InfoModel::where('title',\$data['title'])->value('id');查询标题为data['title']的资讯id, 如果低存在且不等于data['id'], 那么提示标题重复, 并写入日志修改资讯失败, 结果: 标题重复。如果不在id那么即没有重复标题, 将\$data中的所有数据更新到data['id']这条资讯中。添加操作日志, 返回更新数据结果和状态值。不存在post请求那么获取id值, 查询该id的所有资讯信息, 查询所有的资讯栏目, 一起用assign函数模板赋值给视图infoedit.html。fetch函数渲染视图。

(3) 视图(info目录)

?Index.html

资讯列表的显示, 对栏目操作等按钮显示。

?infoadd.html

添加资讯的视图。

?info.html

查看资讯内容的视图。

2. 栏目管理

(1) 模型 (info_type.php)

namespace app\info\model语句声明根命名空间, use think\Model声明thinkphp官方model类命名空间。

class Info_type extends Model定义Power类并继承Model类。class Info_type extends Model定义Info_type类并继承Model类。protected \$autoWriteTimestamp = false;定义受保护的变量autoWriteTimestamp 并赋值false，表示关闭自动写入时间戳。

(2) 控制(Infotype.php)

首先namespace app\info\controller语句声明根命名空间,use app\com\controller\Accesscontrol;申明公共模块的权限控制类的命名空间,

use think\Request;声明thinkphp5的官方Request类的命名空间，用于后期获取请求。use app\info\model\info_type as info_typeModel;声明info_type模型的命名空间取别名为info_typeModel;use app\info\model\P_info;声明P_info模型的命名空间。

class Infotype extends Accesscontrol定义Infotype类并继承Accesscontrol。公共index函数，用于栏目信息的展示。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中栏目查看的id），判断是否拥有权限，拥有则模型all函数查询栏目信息，count函数查询记录数，assign函数模板赋值给视图index.html。fetch函数渲染视图。

公共typeadd函数，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中资讯状添加的id），判断是否有权限添加资讯，拥有则\$request = Request::instance()初始化请求类并赋值给\$request，判断是否有post请求，如果没有则渲染typeadd.html视图。有则设置状态值为空，调用请求中的param方法获取所有表单数据赋值给\$data。查询传递的栏目是否已存在，存在则返回栏目名已存在，且添加操作日志添加栏目失败，不存在则将\$data中的数据添加到对应的字段到栏目数据表。判断添加是否成功，添加操作日志,返回对应的状态值与消息。

公共status函数，启用/禁用栏目,先进行权限控制，调用公共函数control，传入一个实参id（对应power表中栏目状态的修改的id）返回值赋给\$message，判断\$message是否为空从而判断是否拥有权限修改栏目的状态。为空则\$request = Request::instance()初始化请求类并赋值给\$request，请求param函数获取请求的id值，判断是否有该栏目中存在资讯，存在则返回栏目中存在资讯，操作失败。不存在则获取当前状态值，取反后根据id值更新到数据表中，返回对应的状值和消息。

公共typedel方法用于栏目名的删除，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中栏目删除的id）返回值赋给\$message，判断\$message是否为空从而判断是否拥有权限删除栏目。为空则根据前端ajax传递的id值赋值给\$id,根据id值获取栏目中的资讯，判断是否有该栏目中是否存在资讯，存在则返回栏目中存在资讯，操作失败。不存在则使用模型destroy函数删除\$id的栏目。判断删除结果，从而返回对应的消息和状态。

公共typedit方法用于栏目的修改。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中栏目修改的id）判断是否有用权限，没有权限则结束后面的语句，有则\$request = Request::instance()初始化请求类并赋值给\$request。\$data = \$request->param()获取前端ajax传递的数据赋值给\$data。判断是否存在post请求，不存在则查询id为\$data['id']的栏目信息，ssign函数模板赋值给视图typedit.html。fetch函数渲染视图。存在post请求，设置状态值为空，查询修改后的栏目名是否已存在，存在返回消息栏目名重复，不存在将修改后的数据更新到id为\$data['id']的记录中，判断更新是否成功，返回对应的状态值与消息。

(3) 视图 (infotype目录)

?Index.html

栏目列表的显示，对栏目操作等按钮显示。

?Typeadd.html

添加栏目的视图。

?Typeedit.html

更新栏目的视图，显示更新的初始数据。

第六章 基础数据管理

基础数据管理主要是一些对基础数据的管理，如：节点管理，操作日志管理。节点管理主要是用于权限管理的树形复选框的节点进行管理。由三个数据表组成，模块表，控制器表，权限表（操作）。日志管理记录用户操作日志。

1.节点管理

（1）模型（Power.php\Module.php\Controller.php）

namespace app\system\model语句声明根命名空间，use think\Model声明thinkphp官方model类命名空间。

class Power extends Model定义Power类并继承Model类。class Module extends Model定义Module类并继承Model类。class Controller extends Model定义Controller类并继承Model类。protected \$autoWriteTimestamp = false;定义受保护的变量autoWriteTimestamp 并赋值false，表示关闭自动写入时间戳。

（2）控制器（Node.php）

首先namespace app\system\controller语句声明根命名空间,use app\com\controller\Accesscontrol;申明公共模块的权限控制类的命名空间,

use think\Request;声明thinkphp5的官方Request类的命名空间，用于后期获取请求。use app\system\model\Power;声明Power模型的命名空间；use app\system\model\Module;声明Module模型的命名空间。use app\system\model\Controller as ControllerModel;声明Controller模型的命名空间取别名为ControllerModel。

class Node extends Accesscontrol定义Node类并继承Accesscontrol。

公共index函数，用于节点的展示。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中节点查看的id）判断是否有权限，有权限则定义一个\$html为一个空数组，模型all函数查询Power,Module,Controller信息，如下图1所示，将三层嵌套foreach循环和array_push函数相结合，形成一个三维数组赋值给\$html。json_encode函数将该数组转为json格式,然后assign函数模板赋值给视图index.html。fetch函数渲染视图。

公共nodeadd函数用于节点添加，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中节点的id）判断是否有权添加节点，有则\$request=Request::instance()初始化请求类并赋值给\$request，判断是否有post请求，没有则渲染nodeadd.html视图。有则使用请求类中的param函数获取所有请求数据赋值给\$data。判断其中的\$data['type']中的值，如果为1则添加的节点为一级节点，将数据写入module数据表中；如果为2则添加节点为二级节点，将数据写入controller数据表中；如果为3则添加节点为三级节点，将数据写入power数据表中。调用日志函数将结果写入日志中，最后返回添加结果。

公共getparentnode用于获取父节点，当添加节点为二级和三级节点时，需要确定父节点。
\$request=Request::instance()初始化请求类并赋值给\$request，获取请求中的type值赋值给\$type，以确定当前添加的节点级数。当\$type为2时，及添加二级节点，则查询module数据表中的所有数据，返回给视图。当\$type为3时，及添加三级节点，则查询controller数据表中的所有数据，返回给视图。

公共deletenode函数用于节点的删除，先进行权限控制，调用公共函数control，传入一个实参id（对应power表中节点删除的id）返回值赋给\$message，判断\$message是否为空从而判断是否拥有权限删除节点。为空则\$request=Request::instance()初始化请求类并赋值给\$request，\$data=input('post.id/a');获取请求中的数组数据，foreach(\$data as \$value)循环删除多个节点，explode函数分离\$value得到节点级数和id值，模型destroy函数删除对应的节点。调用日志函数将结果写入日志中，最后返回添加结果。

图1：

（3）视图(Node目录)

?Index.html

节点信息的展示和节点操作等按钮的显示。

?Nodeadd.html

添加节点的视图。

2.日志管理

（1）模型（Log.php）

namespace app\system\model语句声明根命名空间，use think\Model声明thinkphp官方model类命名空间。

class Log extends Model定义Log类并继承Model类。protected \$updateTime = false;定义受保护的变量\$updateTime并赋值false，表示关闭自动写入更新时间戳。

（2）控制器(Log.php)

首先namespace app\system\controller语句声明根命名空间,use app\com\controller\Accesscontrol;申明公共模块的权限控制类的命名空间,

use think\Request;声明thinkphp5的官方Request类的命名空间，用于后期获取请求。use app\system\model\Log as LogModel;声明Log模型的命名空间取别名为LogModel。

class Log extends Accesscontrol定义Log类并继承Accesscontrol。

公共index函数，用于操作日志信息的展示和管理。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中日志查看的id）判断是否拥有权限查看日志。拥有权限则模型all和count函数查询log数据表的信息和记录数。然后结果用assign函数模板赋值给视图index.html。再fetch函数渲染视图。

公共deletelog方法用于操作日志的删除。先进行权限控制，调用公共函数control，传入一个实参id（对应power表中日志删除的id）返回值赋给\$message，判断\$message是否为空从而判断是

否拥有权限删除日志。为空则\$request=Request::instance()初始化请求类并赋值给\$request，定义一个空变量\$message，请求的param函数获取操作日志id值的集合赋值给\$data。for语句循环执行destroy函数删除操作日志，当删除失败时跳出循环，并返回删除中断。添加操作日志，并把删除的日志数写入到日志中，最后Return返回结果

(3) 视图(log目录)

?Index.html

操作日志信息的展示和删除操作等按钮的显示。

第七章 总结

通过这几个月时间的努力，本次毕业项目设计的预期效果基本实现了。在撰写论文和设计项目过程中，我的专业综合能力明显比以前增强了许多，我对未来的专业能力更加有信心了。

在设计项目遇到阻碍时，先自己努力使用各种方式来查阅资料，实在搞不明白，我就虚心向老师和同学请教。我觉得无论如何都要端正自己的态度，不要有那种敷衍心理，既然做一件事情，那就要把它做到最好。

此外，老师每次下达任务时，我都会保质保量地完成。与此同时，我自身也存在很多问题。譬如我在撰写过程中会遇到很多阻碍，但是缺乏一定的耐心，急于求成，不能定下心来去冷静地思考进而想出对策来解决问题。

除此之外，虽然本次毕业项目整体效果感觉还不错，但实际许多功能都不是很完善，主要是权限控制那一块做得不怎么完美，权限分配不是那么精准，如：用户本身即使没有权限，但也能查看自己相关的信息。

这次论文让我明白一个道理，之前为了尽早完成毕业设计，便四处寻求老师、学长、同事，寻求他人的指引。但是我后来才发现，其实没有必要向他们请教太多问题，应该自己去查找资料，培养自学的能力。这个方面，我感觉自己做得不够好，在今后的工作生活中，我会努力完善自己。

主要参考文献

[1]齐剂. PHP技术应用于中小企业网站开发[D].吉林大学,2012.

[2]戴一平. MVC设计模式在PHP开发中的应用[J]. 计算机与现代化,2011,(03):33-37+43.

[3]余肖生,易恩. 基于PHP的开发环境搭建与网站设计实现[J]. 重庆理工大学学报(自然科学版),2011,25(03):74-77+101.

[4]易竞. PHP访问MySQL数据库的实现[J]. 保险职业学院学报,2008,(03):79-81.

[5]刘智莹,朱程荣. 基于PHP实现数据安全性的方法及比较[J]. 计算机工程与设计,2009,30(19):4387-4389+4392.

[6]崔伟. 基于MVC设计模式的PHP快速开发框架研究与实现[D].西北农林科技大学,2010.

[7]李小遐. 商务网站搭建技术初探[J/OL]. 现代交际,(2017-10-09).

[8]楼丹娜. 基于PHP和AJAX的企业内部网站系统设计与实现[J]. 电脑迷,2017,(01):192.

[9]Shashank Gupta,B.B. Gupta. Automated Discovery of JavaScript Code Injection Attacks in PHP Web Applications[J]. Procedia Computer Science,2016,78:.

[10]Mario Heiderich. PHP[M].Elsevier Inc.:2011.

致谢

我首先要感谢我的论文指导老师：电子信息学院的万川梅老师。万老师对我论文的研究方向做出了指导性的意见和推荐，在论文撰写过程中及时对我遇到的困难和疑惑给予悉心指点，提出了许多有益的改善性意见，投入了超多的心血和精力。万老师对我的帮忙和关怀表示诚挚的谢意!同时，还要感谢授课的老师是他们教会我我专业知识能力。

此外，还要感谢朋友以及同学们在论文编写中带给的大力支持和帮忙，给我带来极大的启发。也要感谢参考文献中的作者们，透过他们的研究文章，使我对研究课题有了很好的出发点。

最后，谢谢论文评阅老师们的辛苦工作。衷心感谢我的家人、朋友，以及同学们，真是在他们的鼓励和支持下我才得以顺利完成此论文。