

本科毕业论文（设计）

题 目 基于web的作业批改提交系统

设计与实现

学 院 计算机与信息科学学院

专 业 计算机科学与技术

年 级 2014

学 号 222014321210031

姓 名 贺卫龙

指 导 教 师 陈怀东

成 绩

2018年 4 月 8 日

目录

[摘要： 1](#_Toc511856873)

[Abstract: 1](#_Toc511856874)

[第1章 绪论 2](#_Toc511856875)

[1.1建立作业批改提交系统的必要性 2](#_Toc511856876)

[1.2系统开发的背景 2](#_Toc511856877)

[1.3 作业批改提交系统的提出 3](#_Toc511856878)

[第2章 相关技术及开发工具介绍 4](#_Toc511856879)

[2.1 相关技术 4](#_Toc511856880)

[2.1.1 laravel框架 4](#_Toc511856881)

[2.1.2 Ajax异步传输 5](#_Toc511856882)

[2.1.3uploadify文件上传插件 5](#_Toc511856883)

[2.1.4 redis缓存技术 6](#_Toc511856884)

[2.2 使用工具 6](#_Toc511856885)

[2.2.1 PhpStorm 6](#_Toc511856886)

[2.2.2 navicat & mysql 7](#_Toc511856887)

[2.2.3 apache 8](#_Toc511856888)

[2.2.4 Bootstrap工具 8](#_Toc511856889)

[2.2.5 composer 管理 9](#_Toc511856890)

[第3章 系统总体设计 10](#_Toc511856891)

[3.1 用户需求分析 10](#_Toc511856892)

[3.1.1 用户业务需求 10](#_Toc511856893)

[3.2 系统设计原则 11](#_Toc511856894)

[3.2.1 设计基本原则 11](#_Toc511856895)

[3.2.2 系统总体设计需满足的要求 12](#_Toc511856896)

[3.3 系统功能模块设计 12](#_Toc511856897)

[3.3.1 总体功能模块设计 12](#_Toc511856898)

[3.3.2 个人设计功能模块详述 13](#_Toc511856899)

[3.4 数据库设计 14](#_Toc511856900)

[3.4.1 数据库概念设计 14](#_Toc511856901)

[3.4.2 数据库逻辑 15](#_Toc511856902)

[3.4.3数据库物理设计 16](#_Toc511856903)

[第4章 系统具体实现 19](#_Toc511856904)

[4.1系统运行环境 19](#_Toc511856905)

[4.1.1硬件环境 19](#_Toc511856906)

[4.1.2软件环境 19](#_Toc511856907)

[4.2程序主要页面及主要代码 19](#_Toc511856908)

[4.2.1 后台首页 19](#_Toc511856909)

[4.2.2 管理员管理 23](#_Toc511856910)

[4.2.3学校组织机构 26](#_Toc511856911)

[4.2.4 课程相关操作 28](#_Toc511856912)

[4.2.5 教师管理页面 31](#_Toc511856913)

[4.2.6 注册登录 35](#_Toc511856914)

[第5章 展望与总结 38](#_Toc511856915)

[5.1总结 38](#_Toc511856916)

[5.2进一步工作 38](#_Toc511856917)

[参考文献 38](#_Toc511856918)

[致谢 39](#_Toc511856919)

**基于web的作业批改提交系统设计与实现**

贺卫龙

西南大学计算机与信息科学学院、软件学院

摘要：随着互联网的高速发展，我们的日常学习生活已经离不开网络，而作业是我们经常要面对的事情，从最早的提交纸质作业到现在的电子提交，互联网已经成了当前社会的潮流。本系统运用laravel框架开发，主要功能是为全国院校提供一个快速的作业提交批改平台，另外还有一整套教务管理系统，老师可以添加作业和附件，并为学生在线答疑，学生可以在实践规定内提交作业、评论和回复，既方便了学生与老师交作业的麻烦，还可以通过评论完成答疑过程，后续还准备添加一些分类的学术文章等供老师学生阅读与学习。

本文主要对以下功能模块进行设计与实现：学校管理系统、作业上传平台、系统设置、作业实时评论与回复功能。

**关键词**：laravel、无极限分类、redis缓存、orm查询构造器

**Design and implementation of job correcting and submitting system based on Web**

HE Weilong

College of Computer and Information Science, Southwest University, Chongqing 400715, PR China

Abstract: With the rapid development of the Internet, our daily learning and life can not be separated from the network, and the homework is the thing we often have to face. From the earliest paperwork to the present electronic submission, the Internet has become the current trend of the society. The system uses the laravel framework to develop the system. The main function is to provide a rapid operation and correction platform for the national colleges and universities. In addition, there is a complete set of educational administration system. The teacher can add homework and accessories and answer questions for students online. Students can provide homework, comment and reply in practice, which is convenient for learning. The trouble of making the homework with the teacher can also complete the process of answering the question through the comments, and the follow-up is ready to add some classified academic articles for the teachers and students to read and study.

This paper mainly designs and implements the following functional modules: school management system, job upload platform, system setup, real-time reviews and reply functions.

**Key words:** Laravel, no limit classification, redis cache, ORM query builder

# 绪论

## 1.1建立作业批改提交系统的必要性

科学技术日新月异，计算机的发展越来越迅速，我们的生活已经和计算机不可分割。计算机的高效性使得人们的工作效率也有了很大的进步空间，与此同时，人的生活节奏也在朝着更快的方向发展，此时就需要充分的利用互联网的特性来方便我们的生活。

本校作业提交系统主要任务是为教师和学生提供一定的便利服务，只要能完成文件的上传下载大致就能满足需求，在此基础上我发现可以提供一个实时交流的功能，在每次作业下面老师和学生都可以评论、回复与问答，另外老师可以在创建作业的同时自动创建文件目录，且可以添加附件，还可以对学生提交的作业进行审阅批改，整体上方便了很多，因此，充分发挥强大的校园网络服务功能，为老师学生提供更多更好的帮助，建立一个作业提交批改系统成为了当前重要的一个任务。

## 1.2系统开发的背景

伴随着人们生活节奏的加速化，互联网的使用越来越频繁，随处可见，尽管现在也有一些网上作业上传系统，但其都有一大诟病，使用起来都不是很简单，操作都有一定的复杂度，另外未提供实时交流功能，而且作业提交这件事几乎是每天必须要做的事情，如果浪费上课时间来完成这件事，那么上课的节奏就会变慢，如果使用以前的作业上传系统，经常会出现系统崩溃的现象，各种问题层出不穷。

基于以上情况，我准备写出一个使用起来方便快捷，界面优美的web作业提交系统为更多需要的学校提供相关服务。

## 1.3 作业批改提交系统的提出

伴随着社会的进步，科技的快速发展，计算机与我们的生活已经密不可分，成为了我们生活的一部分，作业提交是每个学生必面临的问题，网上作业提交系统就可以满足当前社会对作业提交的需求，不仅可以及时的完成作业的提交，还可以与老师在网上进行问题的讨论，而且还可以同学间互相探讨，在一定的意义上能够节约不少的时间，还可以及时的达到解惑的效果。广大的学生通过使用这个作业上传平台还可以获取到当前最新潮的一些技术，还可以与志同道合的人进行交流，学生也可以自己创建相关主题进行更深的交流，使得自己能更好的理解知识。同时老师可以添加一些学术文章放到网站供学生阅览

作业批改系统的出现不仅可以解决当前网上作业提交的大部分弊端，还可以有更多的功能提供给学生老师使用，且界面友好，整洁，使用起来心旷神怡，大大提高了工作效率。

# 相关技术及开发工具介绍

## 2.1 相关技术

## 2.1.1 laravel框架

Laravel是一个免费的、开源的PHP的Web框架，由Taylor Otwell创造的、用于Web应用程序的模型视图控制器下的发展（MVC）架构模式，基于symfony。Laravel是用一个专门的依赖管理的模块化包装系统，用于访问关系数据库的方式不同，公用事业。源代码在GitHub托管和特许的MIT许可证的条款。功能强大的ORM（对象关系映射）是活动记录模式的高级PHP实现，同时提供内部方法来强制对数据库对象之间的关系进行约束。它不仅能构建一个完美的网络APP，并且使每行代码都简洁并富于表达力。另外，还提供了验证（authentication）、路由（routing）、session和缓存（caching）等开发工具和功能。

Laravel具有以下特点：

(1)功能强大的ORM将数据库表作为类，将它们的对象实例绑定到单个表行。应用程序逻辑是开发的应用程序的一个组成部分，通过使用控制器或作为路由声明的一部分来实现。用于定义应用程序逻辑的语法类似于西纳特拉框架使用的语法。

(2) 类自动加载提供PHP类的自动加载，而不需要手动维护包含路径。按需加载防止包含不必要的组件，因此只有实际使用的组件被加载。

(3) IOC容器，使得新的对象将产生以下控制反转（IOC）的原则，其中框架调用到应用程序或特定任务的代码，可选的实例化和引用新对象的单身。

(4) 自动分页的简化实现分页的任务自动化的方法集成到Laravel代替传统的手工实现的方法。

(5) 形式要求的一个特征是laravel作为形式输入验证由内部绑定事件侦听器的基础，导致自动调用的形式验证方法和实际形态的生成。

(6) 文档详尽，并且laravel社会强力支持。Laravel代码本身的表现力和良好的文档使PHP程序编写令人愉快。[1]

本系统使用的是laravel5.3版本，版本更高，功能更加齐全。

### 2.1.2 Ajax异步传输

AJAX即“Asynchronous JavaScript and XML”，无需重新加载整个网页的情况下，能够更新部分网页，完成数据传输与校验。是一套综合了多项技术的[浏览器](https://zh.wikipedia.org/wiki/%E7%80%8F%E8%A6%BD%E5%99%A8)端[网页](https://zh.wikipedia.org/wiki/%E7%B6%B2%E9%A0%81)开发技术。该概念由[杰西·詹姆士·贾瑞特](https://zh.wikipedia.org/wiki/%E5%82%91%E8%A5%BF%C2%B7%E8%A9%B9%E5%A7%86%E5%A3%AB%C2%B7%E8%B3%88%E7%91%9E%E7%89%B9)所提出。[2]

传统的Web应用需要用户在用户端填写数据表单，提交表单时就向[后端服务器](https://zh.wikipedia.org/wiki/%E7%B6%B2%E9%A0%81%E4%BC%BA%E6%9C%8D%E5%99%A8)发送一个数据请求，服务器接收并处理表单传来的数据，然后跳转到一个新的网页，但这个做法浪费很多带宽资源，因为在前后两个网页的大部分前端代码往往是相同的。由于每次应用的沟通都需要向服务器发送请求，应用的回应时间依赖于服务器的回应时间，这导致了用户界面的响应速度比本机应用慢得多。与此不同的是AJAX应用可以仅向服务器发送并取回必须的数据，并在客户端采用JavaScript及时处理来自服务器的回应。因为在服务器和浏览器之间交换的数据大量减少，服务器反应速度更快了。同时，很多的处理工作可以在发出请求的[客户端](https://zh.wikipedia.org/wiki/%E5%AE%A2%E6%88%B7%E7%AB%AF)机器上完成，因此Web服务器的压力也减少了。

Ajax不需要任何浏览器插件，但需要用户允许[JavaScript](https://zh.wikipedia.org/wiki/JavaScript)在浏览器上执行。随着Ajax的成熟，一些简化Ajax使用方法的程序库也相继问世。同样，也出现了另一种辅助程序设计的技术，为那些不支持[JavaScript](https://zh.wikipedia.org/wiki/JavaScript)的用户提供替代功能。[3]

### 2.1.3uploadify文件上传插件

Uploadify是一款来自国外的jQuery插件，主要功能是批量上传文件，此插件现在已经被很多项目运用，并有一整套的说明文档，但是也发现不少问题，必须得花点时间才能搞清这个插件的使用，只有通过不断的采坑与填坑才能真真正正的学会这一插件的使用，并完美的运用到自己的项目中。[4]

一般情况下上传文件，都会使用input标签中的file类型。如果上传的文件过大，或者文件过多，input标签将达不到系统的要求，正是由于这个自带标签的局限性，有了Uploadify这款优秀的插件诞生。与ajax和flash相结合，实现多线程上传的功能，不仅界面样式优美且上传速度也高效，使用起来也能很快上手，在第一次使用之后第二次就能很迅速的完成。

### 2.1.4 redis缓存技术

Redis是一种非关系型数据库，NoSQL是对不同于传统的[关系数据库](https://zh.wikipedia.org/wiki/%E9%97%9C%E8%81%AF%E5%BC%8F%E8%B3%87%E6%96%99%E5%BA%AB)的[数据库管理系统](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)的统称。两者存在许多显著的不同点，其中最重要的是NoSQL不使用SQL作为查询语言。其数据存储可以不需要固定的表格模式，也经常会避免使用SQL的[JOIN](https://zh.wikipedia.org/wiki/%E8%BF%9E%E6%8E%A5_(SQL))操作，一般有[水平可扩展性](https://zh.wikipedia.org/w/index.php?title=%E6%B0%B4%E5%B9%B3%E5%8F%AF%E6%89%A9%E5%B1%95%E6%80%A7&action=edit&redlink=1)的特征。根据月度排行网站DB-Engines.com的数据显示，Redis是最流行的键值对存储数据库。[4]

具备以下优势：

(1)异常快速：Redis的速度非常快，每秒能执行约11万集合，每秒约81000+条记录。

(2)支持丰富的数据类型：Redis支持最大多数开发人员已经知道像列表，集合，有序集合，散列数据类型。这使得它非常容易解决各种各样的问题，因为我们知道哪些问题是可以处理通过它的数据类型更好。

(3)操作都是原子性：所有Redis操作是原子的，这保证了如果两个客户端同时访问的Redis服务器将获得更新后的值。

(4)多功能实用工具：Redis是一个多实用的工具，可以在多个用例如缓存，消息，队列使用(Redis原生支持发布/订阅)，任何短暂的数据，应用程序，如Web应用程序会话，网页命中计数等。

## 2.2 使用工具

### 2.2.1 PhpStorm

phpStorm是PHP编辑器，PhpStorm的支持PHP代码5.3、5.4、5.5、5.6和7，包括命名空间、列表和数组语法。它包含一个完整的SQL编辑器，带有可编辑的查询结果。phpStorm是基于IntelliJ IDEA，这是用java写的。开发人员可以通过安装插件扩展IDE的IntelliJ平台创建或编写自己的插件。

PhpStorm是一个轻量级且便捷的PHP IDE，其旨在提高用户工作效率，可深刻理解用户的编码，提供智能代码补全，快速导航以及语法错误和拼写错误检查。相对其他编辑器来说，phpstorm支持php代码补全、支持Smarty模板和PHPDoc，支持多编程语言混合，属于轻量级IDE，安装过程简单易懂、在Windows，Mac OS X，Linux上都可以运行，项目配置简单，可以再任意地方打开代码，另外环境智能化，支持SVN、Git、Mercurial，支持FTP和远程文件同步、可以记录本地修改，无需任何配置，支持设置断点、批量代码分析。[12]

### 2.2.2 navicat & mysql

Navicat premium是一款数据库管理工具，使用起来相当方便，之前也用过一些其他的可视化工具，像phpadmin等，操作起来都相对复杂一点，navicat支持远程数据库连接和本地数据库连接，相对命令行来说方便的多。

Navicat的设计合符合各种开发人员要求，小到数据库管理员和程序员，大到各种为客户服务并与合作伙伴共享信息的不同企业或公司。Navicat 是一个跨平台工具，支持 [Microsoft Windows](https://zh.wikipedia.org/wiki/Microsoft_Windows)、[Mac OS X](https://zh.wikipedia.org/wiki/Mac_OS_X) 及 [Linux](https://zh.wikipedia.org/wiki/Linux) 平台。购买软件时，使用者可以从 11 个可用的语言中选择一种语言：英文，法文，德文，西班牙文，日文，波兰文，俄文，葡萄牙文，韩文，简体中文和繁体中文。不同的资料库的批次处理任务也可以设定排程并在指定的时间执行。其他功能包括汇入、汇出精灵、查询建立工具、报表产生器、资料同步、备份、工作排程及更多。

本程序使用的是mysql数据库，MySQL是一种关系型数据库管理系统，和非关系型数据不同的是关系型数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL性能高、成本低、可靠性好，已经成为最流行的开源数据库，因此被广泛地应用在[Internet](https://zh.wikipedia.org/wiki/Internet)上的中小型[网站](https://zh.wikipedia.org/wiki/%E7%B6%B2%E7%AB%99)中。本程序在本地使用的是wamp，服务端使用的是lnmp。

有以下特性：

(1)源代码具有可移植性，使用[C](https://zh.wikipedia.org/wiki/C%E8%AA%9E%E8%A8%80)和[C++](https://zh.wikipedia.org/wiki/C%2B%2B)编写，并使用多种[编译器](https://zh.wikipedia.org/wiki/%E7%B7%A8%E8%AD%AF%E5%99%A8)测试基于开源许可发布，无需付费即可使用，

(2)支持在多种[操作系统](https://zh.wikipedia.org/wiki/%E4%BD%9C%E6%A5%AD%E7%B3%BB%E7%B5%B1)上运行，如[Linux](https://zh.wikipedia.org/wiki/Linux)、[Mac OS](https://zh.wikipedia.org/wiki/Mac_OS)、[Windows](https://zh.wikipedia.org/wiki/Windows)等

(3)可为多种編程语言提供了[API](https://zh.wikipedia.org/wiki/API)。这些編程语言包括[C](https://zh.wikipedia.org/wiki/C%E8%AA%9E%E8%A8%80)、[C++](https://zh.wikipedia.org/wiki/C%2B%2B)、[C#](https://zh.wikipedia.org/wiki/C%E2%99%AF)、[Java](https://zh.wikipedia.org/wiki/Java)、[PHP](https://zh.wikipedia.org/wiki/PHP)、[Python](https://zh.wikipedia.org/wiki/Python)、等语言。适用于 PHP 这种 Web 开发者最喜欢使用的语言。

(4)支持[多线程](https://zh.wikipedia.org/wiki/%E5%A4%9A%E7%BA%BF%E7%A8%8B)，充分利用[CPU](https://zh.wikipedia.org/wiki/CPU)资源，支持多用户。

(5) 使用业内所熟悉的标准SQL数据库语言，有效地提高查询速度。

(6)提供多语言支持，常见的编码如中文的[GB 2312](https://zh.wikipedia.org/wiki/GB_2312)、[BIG5](https://zh.wikipedia.org/wiki/%E5%A4%A7%E4%BA%94%E7%A2%BC)等都可以用作数据表名和数据列名。

(7) 非常迅速，即使面对大型数据集也毫无滞涩,可以处理拥有上千万条记录的大型数据库。[6][7]

### 2.2.3 apache

随着科技的不断发展，服务器种类也越来越多，在众多的服务器种类中，我一般会使用apache和nginx，由于本机是windows系统，故使用apache。一般会在本地搭建虚拟域名，修改vhost相关配置、hosts本地配置等可以直接通过虚拟域名访问到服务器地址，操作起来相当方便。

  Apache是世界使用排名第一的Web[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)软件。它可以运行在几乎所有广泛使用的[计算机平台](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E5%B9%B3%E5%8F%B0)上，由于其[跨平台](https://baike.baidu.com/item/%E8%B7%A8%E5%B9%B3%E5%8F%B0)和安全性被广泛使用，是最流行的Web服务器端软件之一。它快速、可靠并且可通过简单的API扩充，将[Perl](https://baike.baidu.com/item/Perl)/[Python](https://baike.baidu.com/item/Python)等[解释器](https://baike.baidu.com/item/%E8%A7%A3%E9%87%8A%E5%99%A8)编译到服务器中。[10]

### 2.2.4 Bootstrap工具

Bootstrap 是一个用于快速开发 Web 应用程序和网站的前端框架。它为开发人员创建接口提供了一个简洁统一的解决方案，包含了功能强大的内置组件，易于定制，还提供了基于 Web 的定制。[8]

Bootstrap采用模块化设计，并且用[LESS样式表语言](https://zh.wikipedia.org/wiki/LESS_(%E5%B1%82%E5%8F%A0%E6%A0%B7%E5%BC%8F%E8%A1%A8))来实现各种组件和工具。一个名为bootstrap.less的文件包括了这些组件和工具，开发者可以修改这个文件，自行决定项目需要哪些组件。

从Bootstrap 2.0开始，Bootstrap文档包括一个叫做“自定义”的特别选项，开发者可以根据自己的实际需要来选择包含的组件和效果，然后生成和下载已经编译好的包。

### 2.2.5 composer 管理

Composer 是 PHP5.3以上 的一个依赖管理工具，它的使命就是帮你为项目自动安装所依赖的开发包，它允许你声明项目所依赖的代码库，它会在你的项目中为你安装他们。作为一个开发人员，你所要做的就是在 composer.json 文件中声明当前项目所依赖的开发包，然后运行 composer.phar install 就行了。composer.json 文件定义了当前项目所依赖的开发包和 composer 的配置信息。

  Composer允许你申明项目所依赖的代码库，它会在你的项目中为你安装他们。主要解决的问题：有非常多优秀的开源库，直接使用可以大大节约开发的 时间，但是在标准的库里面，又没有这些功能，于是在运行他们对应的系统的时候，需要去下载相关的库。因为你需要跨平台的移植性，并对所引用的库 进行说明与管理，由此就composer出现在开发人员的圈子里。[9][11]

# 系统总体设计

## 3.1 用户需求分析

### 3.1.1 用户业务需求

本系统由后台管理系统、教师管理系统、学生作业提交系统三大模块组成，主要用于学生网上作业的提交与沟通、教师网上作业批改。通过对本系统的分析，首先要创建一个学校管理系统，在学校下设立学院、学院下面设立专业、专业下边设立班级和课程，学院下也可以添加课程。学生可以选课，也可以系统的分配课程。系统主要有以下角色：教师、学生、管理员。管理员可以管理文件、学生、教师以及配置信息，教师可以添加作业和附件到某课程，还可以答疑，另外学生可以上传作业和评论等，以下为系统用户功能与权限介绍，用户需求模型如

图1.1：



图1.1 系统需求模型图

Fig.1.1 System requirement model diagram

(1)未注册用户

未注册用户只能访问前端展示页面，这里会有一些学术信息展示，另外可以完成注册与登录功能

(2)教师

添加作业与附件、批改作业、添加课程、班级学生管理

(3)学生

提交作业与评论、查看个人信息、查看班级课程信息、查看作业信息

(4)管理员

管理学生与老师，管理学校组织机构的设立与修改，前台配置、课程分配与课程添加等

## 3.2 系统设计原则

### 3.2.1 设计基本原则

(1) 系统定位

明确当前系统的具体服务对象以及要实现的相关功能点，通过不断斟酌找到合适的方法来实现

(2) 技术的实时性与效率性

在系统设计时，根据系统定位以及系统规模的大致估计，不仅要考虑到技术是否已经落伍，还要考虑技术是否具有其他技术所没有的重要特性，使得在系统设计的时候能达到很高的性价比。

(3) 系统的友好性、可扩展性与可维护性

首先在设计系统的时候必须要考虑到系统设计的友好性，以及当前系统是否能够快速完成功能的扩展、后期维护是否要容易一些，充分考虑这些问题之后，可以充分发挥出系统的功效以及实时跟进社会潮流。

(4) 系统安全性

系统安全性是最为重要的一点，需要对来访人员权限进行限制，用户登录使用token进行规束，如果是放到服务器上边的话需要进行各项安全设置、数据要及时备份，以防病毒入侵导致数据丢失。

### 3.2.2 系统总体设计需满足的要求

(1) 功能可扩展、可裁剪

后台逻辑的设计和数据库表的建立要考虑之后的扩展需求，通过使用mysql数据库和php的优秀框架laravel开发本平台系统，使系统适应性更强，然后在性能上laravel框架也是很高的，开发起来也相当方便，集成了各种类库。

(2) 较高安全性

Laravel的csrf功能确保表单的安全性，另外有各种密码加密算法确保密码不会外泄，将数据库连接的各项配置进行封装，为系统提供了较高的安全性

(3) 用户界面友好性

用户界面友好是每个系统设计之初必须要考虑的问题，主要采用了bootstrap的一些样式完成网页基础界面样式和特效，另外使用laravel的模板继承功能完成页面架构的设计，设计风格想对正规，符合一般的网页设计。

## 3.3 系统功能模块设计

### 3.3.1 总体功能模块设计

根据上边的需求分析以及对系统大致流程的确定，分不同的角色对功能进行全面的分析，并结合实际情况列出以下功能。 图2.1为系统的整体功能模块图：



图2.1：系统总体功能模块图

Fig.2.1 System overall function module diagram

(1)学校系统建立

学校系统的建立需要用到无极限分类，通过继承id完成对应的数据结构，每个类有一个父id与path路径，路径是用来判断当前机构所属类别，比如说学校、学院、专业等，此外还有一个类别是班级，班级有两种情况，一种是专业下边添加的班级，另外一种是学校性选修课创建的班级，这两种班级分类一样，但对应的path路径不同。

(2)课程创建与分配

管理员和教师都拥有创建课程的权限，课程同时也分为两种，一种为专业课，另外一种为通选课，都是建立在学校系统的基础上产生的，专业课是通过分配相应班级，并分配一名授课教师以完成课程分配，若果该班级已被分配教师，则会有已分配提示；通选课开通首先要创建相应班级，通选课则是全校性选修课，学生通过选课系统进行选课后，学生名单会记录在该班级记录中，后期可通过教师确认得以选到该课程。

(3)作业添加与上传

教师具有作业添加的功能，在其相关课程相关班级下边可以添加作业详情及附件，文件会自动保存到相关的文件夹中，学生登陆后会显示所有相关课程，在相关课程中选择相关的作业批次，完成作业的提交。

(5)作业评论及回复

同以上功能，在相关作业下边提供了评论框，学生可以将自己的疑惑都书写下来，教师在看到该学生评论后可以给予相应的解答，另外学生也可以回复学生，实时性的交流与沟通，以加强学生的学习效果。

(6)用户管理模块

后台管理员可以进行所有用户的管理、添加、修改与删除，学生、教师账号可以自行进行注册登录，在注册的时候选择到自己对应的组织中。另外教师可以看到自己所在的班级和课程，也可以通过班级查看到对应的学生信息。所有用户在登陆之后都可以在第一栏或者右上角处点击查看自己的个人信息与修改，所有用户在未登录时不能访问到系统的关键性界面，在登陆之后系统会根据你的用户类别自动跳转到相应的用户页面以便后续操作。

### 3.3.2 个人设计功能模块详述

(1) 课程信息管理

后台管理员和相关教师可以对课程进行配置与修改，可以开启课程、设置课程开设时间、设置课程选课状态、设置课程的其他基础性功能。

(2) 图片管理

后台管理员可以通过轮播图的配置来对前台首页的轮播图进行全方位的配置，此处使用的是通过数据库存储的轮播图信息，轮播图片则是通过本地缓存复制到相应的文件夹下边，除了轮播图，另外还有前端logo、icon等图标，都是通过本地缓存，然后根据当前时间和该图片的分类上传到不同的文件夹中，整体规划完整统一。

(3) 用户信息反馈

每个用户都可以提相关意见到本系统，在首页会有一个弹框来完成相应的功能，在收到一定数量的反馈信息后，本系统会根据广大用户的意见进行相应的修改，确保完整的用户体验效果，游客也可以提自己的意见，当然，属于匿名提交。

(4) 作业管理

教师可以为课程添加相关作业和附件，在作业产生后服务器会自动常见相关目录文件来存储，并将其文件名存储进数据库中，学生可以再系统中查看到相关已修课程作业，并上传作业到该作业下边，后台管理员可以查看所有文件信息。另外教师可以在学生上传作业后预览学生作业并完成批改，学生在此查看时可以看到自己当前作业的分数。

## 3.4 数据库设计

### 3.4.1 数据库概念设计

通过对业务流程的整体分析与规划，形成反映信息需求的概念数据模型。从而得出作业上传系统的E-R图，如图3.1所示



图3.1：E-R图

Fig.3.1 E-R diagram

### 3.4.2 数据库逻辑

在概念设计的基础上进行逻辑设计，将概念模型转化为关系数据模型， 从而给出数据库中的关系模式，并对关系数据模型进行优化和消除依赖处理，下面是作业上传系统的关系模式，其中带下画线的属性组为关系的主码。

机构类别表：

(ID、机构名称、父id值、路径、权重、类别、描述、创建时间)

课程表：

(ID、课程名称、课程简介、所继承机构id、是否开启、选课状态、类型、开始时间、结束时间)

班级表

(ID、班级名称、班级父类机构id、路径、描述、创建时间)

作业表：

(ID、作业标题、作业详情、作业附件、教师id值、班级id值、课程id值、创建时间)

教师表/学生表

(ID、用户名、密码、手机号、邮箱账号、所属机构id值、学号、开始时间、结束时间、上次登录时间、状态、照片)

管理员表：

(ID、名称、密码、开始时间、结束时间、类型、状态)

评论表：

(ID、作业id值、回复评论id值、回复内容、发布时间、教师id值、学生di值)

分数表：

(ID、学生id值、教师id值、问题id值、分数)

### 3.4.3数据库物理设计

通过前边的概念设计和逻辑设计，可以轻松的搭建好物理设计、物理设计可以通过navicat来快速搭建、也可以通过命令行完成数据库一步一步搭建，数据库设计的好坏决定了系统的好坏。下面为我的物理设计表结构

表3.1 admin信息表

Table3.1 admin information table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否为空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | 递增 | 管理员id |
| name | varchar | 否 |  | Null | 管理员名称 |
| password | varchar | 否 |  | Null | 管理员密码 |
| starttime | timestamp | 否 |  | Null | 创建时间 |
| lastlogin | timestamp | 否 |  | Null | 上次登录时间 |
| typeid | int | 是 |  | Null | 所属分类id |
| kind | Enum('3','2','1','0') | 是 |  | Null | 种类 |
| status | enum('0','1') | 否 |  | 1 | 状态 |

表3.2 用户基础表

Table3.2 user base table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | 递增 | 用户id值 |
| username | varchar | 否 |  | Null | 用户名 |
| password | varchar | 否 |  | Null | 密码 |
| phone | varchar | 否 |  | Null | 电话 |
| gender | enum('0','1') | 否 |  | Null | 性别 |
| typeid | int | 否 |  | Null | 所属组织id |

表3.3 用户详情表

Table3.3 user detail table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否为空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 否 | 同上表id | 用户id值 |
| name | varchar | 否 |  | Null | 姓名 |
| number | varchar | 否 |  | Null | 学号 |
| email | varchar | 是 |  | Null | 邮箱 |
| cid | int | 是 |  | Null | 所属班级 |
| starttime | date | 否 |  | Null | 入学时间' |
| endtime | date | 是 |  | Null | 毕业时间 |
| lastlogin | datetime | 否 |  | 1970-01-01 00:00:00 | 最近登录时间 |
| state | enum('1','0') | 是 |  | Null | 状态 |
| photo | varchar | 否 |  | male.jpg | 照片地址 |

表3.4 组织分类表

Table3.4 orignazition type table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | 自增 | id值 |
| name | varchar | 否 |  | Null | 组织名称 |
| pid | int | 是 |  | Null | 父id值 |
| path | varchar | 否 |  | Null | 路径 |
| sort | int | 否 |  | Null | 权重值 |
| kind | int | 否 |  | Null | 种类 |
| description | varchar | 否 |  | Null | 描述 |
| createtime | date | 是 |  | Null | 创建时间 |

表3.5 课程信息表

Table3.5 course information table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | 自增 | id值 |
| name | varchar | 否 |  | Null | 课程名称 |
| info | varchar | 否 |  | Null | 简介 |
| typeid | int | 否 |  | Null | 组织id |
| isable | enum('0','1') | 否 |  | 0 | 是否可用 |
| isselect | enum('0','2','1') | 否 |  | 0 | 是否可选择 |
| type | enum('1','0') | 否 |  | 0 | 课程类型 |
| starttime | date | 否 |  | Null | 开始时间 |
| endtime | date | 否 |  | Null | 结束时间 |

表3.6 用户关系表

Table3.6 user relation table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | 自增 | id值 |
| studentid | int | 否 |  | Null | 用户id |
| classid | int | 否 |  | Null | 班级id |

表3.7 作业表

Table3.7 homework table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | 自增 | id值 |
| name | varchar | 否 |  | Null | 问题名称 |
| info | varchar | 否 |  | Null | 问题介绍 |
| tid | int | 否 |  | Null | 教师id |
| cid | int | 否 |  | Null | 班级id |

表3.8关系表

Table3.8 information table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | 自增 | id值 |
| studentid | int | 否 |  | Null | 用户id |
| classid | int | 否 |  | Null | 班级id |

表3.9 评论表

Table3.8 review table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | Null | 评论id |
| qid | int | 否 |  | Null | 作业id |
| toid | int | 是 |  | Null | 回复id |
| content | varchar | 否 |  | Null | 回复内容 |
| publishtime | datetime | 否 |  | Null | 发表时间 |
| tid | int | 是 |  | Null | 教师id |
| sid | int | 是 |  | Null | 学生id |

表3.10分数表

Table3.10 score table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 是否允许空 | 主键 | 默认值 | 描述 |
| id | int | 否 | 是 | Null | 分数id |
| sid | int | 否 |  | Null | 学生id |
| tid | int | 否 |  | Null | 教师id |
| cid | int | 否 |  | Null | 课程id |
| qid | int | 否 |  | Null | 作业id |
| score | int | 否 |  | Null | 分数 |

# 系统具体实现

## 4.1系统运行环境

### 4.1.1硬件环境

(1) 服务器端：服务器当前用户量小，采用了腾讯云主机，centos7.0系统

配置：1核 1GB 1Mbps，系统盘：普通云硬盘，网络：基础网络

后续用户量增加会加大服务器配置

(2) 用户端：客户端主要用于浏览器和操作数据，客户端想对要求较低。随着科学技术的进步，性价比想对高了很多，一般配置的电脑足以达到配置要求。

### 4.1.2软件环境

(1) 服务器：

操作系统：windows7+

网络协议：TCP/IP

Web服务器：tomcat/apache

数据库：mysql

缓存数据库：redis

(2) 用户端：

操作系统：windows7+

网络协议：TCP/IP

浏览器：chrome、firefox、ie8+

## 4.2程序主要页面及主要代码

### 4.2.1 后台首页

主要截图如图4.1所示



图4.1 后台首页

Fig.4.1 admin first page

主要代码如下：

<**div class="container"**><**nav class="navbar navbar-default navbar-fixed-top navbar-blue" role="navigation"**>  
 <**div class="container-fluid"**><**div class="navbar-header"**>  
 <**button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"**>  
 <**span class="sr-only"**>Toggle navigation</**span**>  
 <**span class="icon-bar"**></**span**>  
 <**span class="icon-bar"**></**span**>  
 <**span class="icon-bar"**></**span**>  
 </**button**>  
 <**a class="navbar-brand" href="/admin"**><**img style="display**:**inline" width="30px" src="/style/admin/img/1.png" alt=""**> 后台管理系统</**a**>  
 </**div**><**div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1"**>  
 <**ul class="nav navbar-nav navbar-right"**>  
 <**li**><**a href="/admin/flush"**><**span class="glyphicon glyphicon-refresh"**></**span**>清除缓存</**a**></**li**>  
 <**li class="dropdown"**>  
 <**a href="#" class="dropdown-toggle" data-toggle="dropdown"**>后台管理<**span class="caret"**></**span**></**a**>  
 <**ul class="dropdown-menu" role="menu"**>  
 <**li**><**a href="/admin/info/**{{session(**'adminUserInfo.id'**)}}**"**>**@if(**session(**'adminUserInfo.name'**)**)** {{session(**'adminUserInfo.name'**)}}  
 **@else** admin  
 **@endif** </**a**>  
 </**li**>  
 <**li**><**a href="#" data-toggle="modal" data-target="#editPass"**>修改密码</**a**></**li**>  
 <**li**><**a href="/" target="\_blank"**>前台首页</**a**></**li**>  
 <**li**><**a href="/admin/logout"**>退出</**a**></**li**>  
 </**ul**>  
 </**li**>  
 </**ul**>  
 </**div**></**div**></**nav**>  
 *<!-- 内容区域 -->* <**div class="row body"**>  
 *<!-- 菜单 -->* <**div class="col-md-2"**>  
 *<!-- 管理员管理-->* <**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h2 class="panel-title" id="admin"**><**span class="glyphicon glyphicon-user"**></**span**> 管理员管理</**h2**>  
 </**div**>  
 <**ul class="list-group"**>  
 <**li class="list-group-item"**><**a href="/admin/info/**{{session(**'adminUserInfo.id'**)}}**"**>个人信息</**a**></**li**>  
 <**li class="list-group-item"**><**a href="/admin/admin"**>管理员列表</**a**></**li**>  
   
 </**ul**>  
 </**div**>  
 *<!—用户管理 -->* <**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h2 class="panel-title" id="user"**><**span class="glyphicon glyphicon-user"**></**span**> 用户管理</**h2**>  
 </**div**>  
 <**ul class="list-group"**>  
 <**li class="list-group-item"**><**a href="/admin/user/0"**>教师列表</**a**></**li**>  
 <**li class="list-group-item"**><**a href="/admin/user/1"**>学生列表</**a**></**li**>  
 </**ul**>  
 </**div**>  
 *<!-- 分类管理 -->* <**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h2 class="panel-title" id="type"**><**span class="glyphicon glyphicon-tasks"**></**span**> 分类管理</**h2**>  
 </**div**>  
 <**ul class="list-group"**>  
 <**li class="list-group-item"**><**a href="/admin/type"**>分类列表</**a**></**li**>  
 <**li class="list-group-item"**><**a href="/admin/manager"**>班级管理</**a**></**li**>  
 </**ul**>  
 </**div**>  
 *<!—课程管理 -->* <**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h2 class="panel-title"**><**span class="glyphicon glyphicon-gift"**></**span**> 课程管理</**h2**>  
 </**div**>  
 <**ul class="list-group"**>  
 <**li class="list-group-item"**><**a href="/admin/course"**>课程列表</**a**></**li**>  
 *{{--<li class="list-group-item"><a href="">选课列表</a></li>--}}* <**li class="list-group-item"**><**a href="/admin/assign"**>课程分配</**a**></**li**>  
  
 </**ul**>  
 </**div**>  
 *<!—文件管理 -->* <**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h2 class="panel-title"**><**span class="glyphicon glyphicon-list-alt"**></**span**> 文件管理</**h2**>  
 </**div**>  
 <**ul class="list-group"**>  
 <**li class="list-group-item"**><**a href=""**>文件列表</**a**></**li**>  
 </**ul**>  
 </**div**>  
 *<!-- 系统管理 -->* <**div class="panel panel-primary"**>  
 <**div class="panel-heading"**>  
 <**h2 class="panel-title"**><**span class="glyphicon glyphicon-certificate"**></**span**> 系统管理</**h2**>  
 </**div**>  
 <**ul class="list-group"**>  
 <**li class="list-group-item"**><**a href="/admin/config"**>系统配置</**a**></**li**>  
 <**li class="list-group-item"**><**a href="/admin/slider"**>轮播图管理</**a**></**li**>  
 <**li class="list-group-item"**><**a href="/admin/article"**>学术论文管理</**a**></**li**>  
 <**li class="list-group-item"**><**a href="/admin/types"**>论文分类管理</**a**></**li**>  
 </**ul**>  
 </**div**>  
 </**div**>**@yield('main')** *<!-- 占位 -->*</**div**>  
</**div**>

### 4.2.2 管理员管理

(1)管理员展示页面 如图4.2

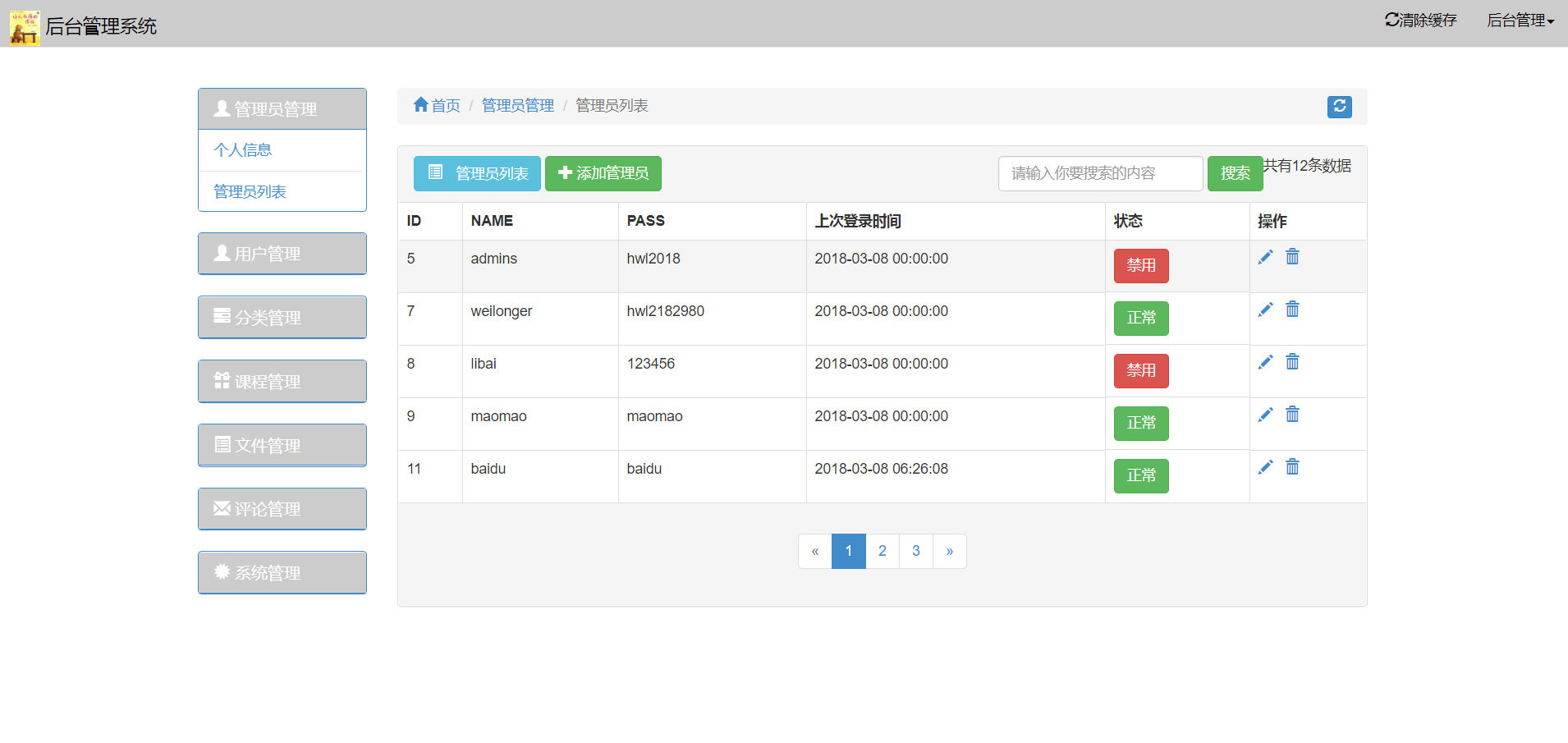


图4.2 管理员展示页面

Fig.4.2 admin show page

主要代码如下：

**public function** index(Request $request){  
*// echo "管理员控制器";* $tot=DB::*table*(**'admin'**)->count();  
 $users = DB::*table*(**'admin'**)->paginate(5);  
 $type=\DB::*table*(**'type'**)->select(\DB::*raw*(**'\*,concat(path,id) as p'**))->where(**'kind'**,**'<='**,**'2'**)->orderBy(**'p'**,**'asc'**)->get();  
 **return** view(**'admin.admin.index'**)->with(**'data'**,$users)->with(**'tot'**,$tot)->with(**'type'**,$type);  
 }

(2)管理员修改页面 如图4.3

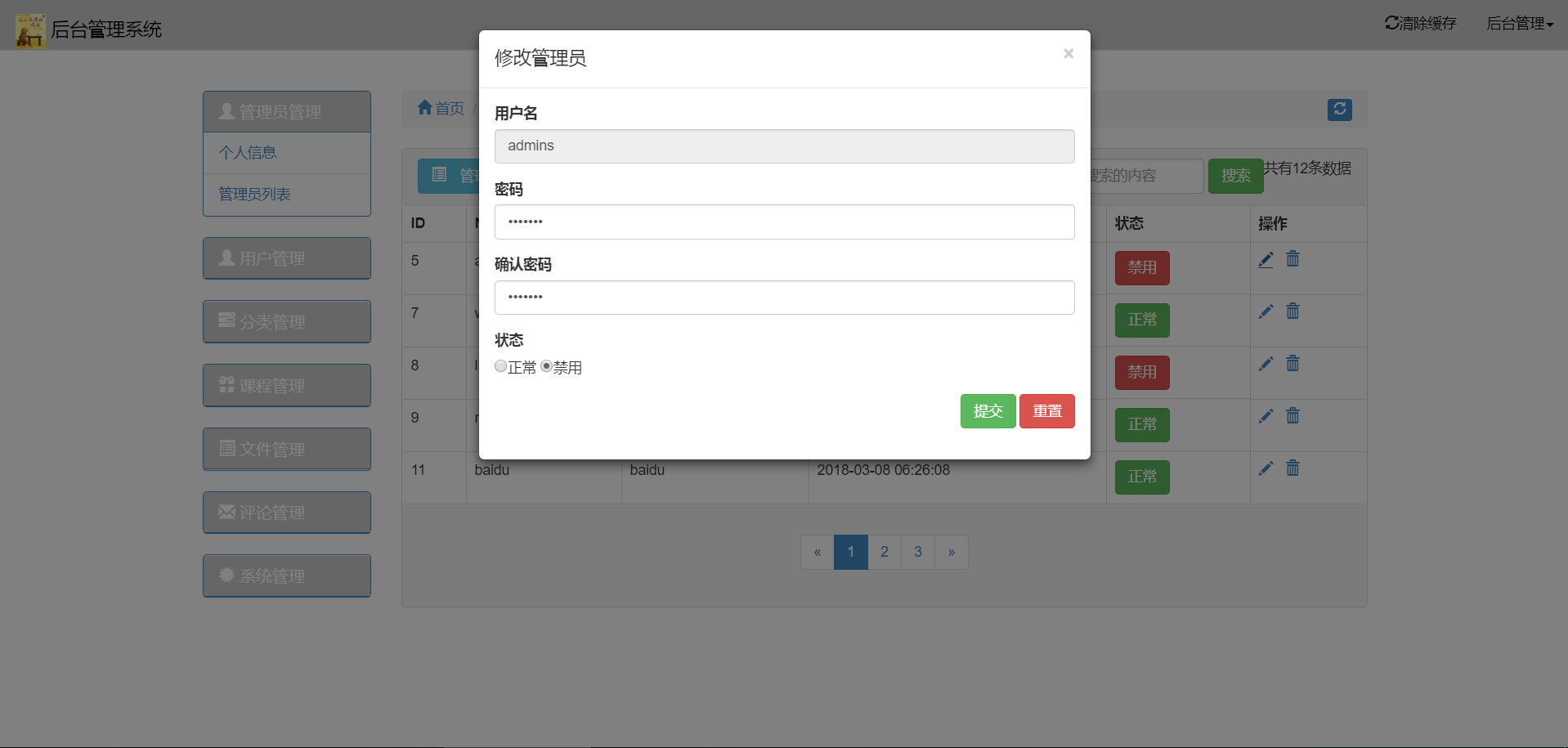


图4.3 管理员修改弹框

Fig.4.3 admin modify modal

主要代码如下：

**public function** store(){ *parse\_str*($\_POST[**'str'**],$arr);  
 *// 表单验证的规则* $rules=[  
 **'name'** => **'required|unique:admin|between:5,12'**,  
 **'password'** => **'required|same:repass|between:5,12'**,  
 ];  
 *// 表单验证的提示信息* $message=[  
 **"name.required"**=>**"请输入用户名"**,  
 **"password.required"**=>**"请输入密码"**,  
 **"name.unique"**=>**"用户名已存在"**,  
 **"password.same"**=>**"两次密码不一致"**,  
 **"password.between"**=>**"密码长度不在5-12位之间"**,  
 **"name.between"**=>**"用户名长度不在5-12位之间"**,  
 ];$validator = Validator::*make*($arr,$rules,$message);**if** ($validator->passes()) {  
 *// 验证通过添加数据库* **unset**($arr[**'repass'**]);  
 $arr[**'password'**]=\Crypt::*encrypt*($arr[**'password'**]);  
 $arr[**'scid'**] = 1;  
 $arr[**'starttime'**]=*date*(**'Y-m-d H:i:s'**,*time*());  
 $arr[**'lastlogin'**]=*date*(**'Y-m-d H:i:s'**,*time*());**if** (DB::*table*(**"admin"**)->insert($arr)) {  
 **return** 1;  
 }**else**{  
 **return** 0;  
 }  
 }**else**{**return** $validator->getMessageBag()->getMessages();  
 }  
}

(3)管理员添加页面 如图4.4

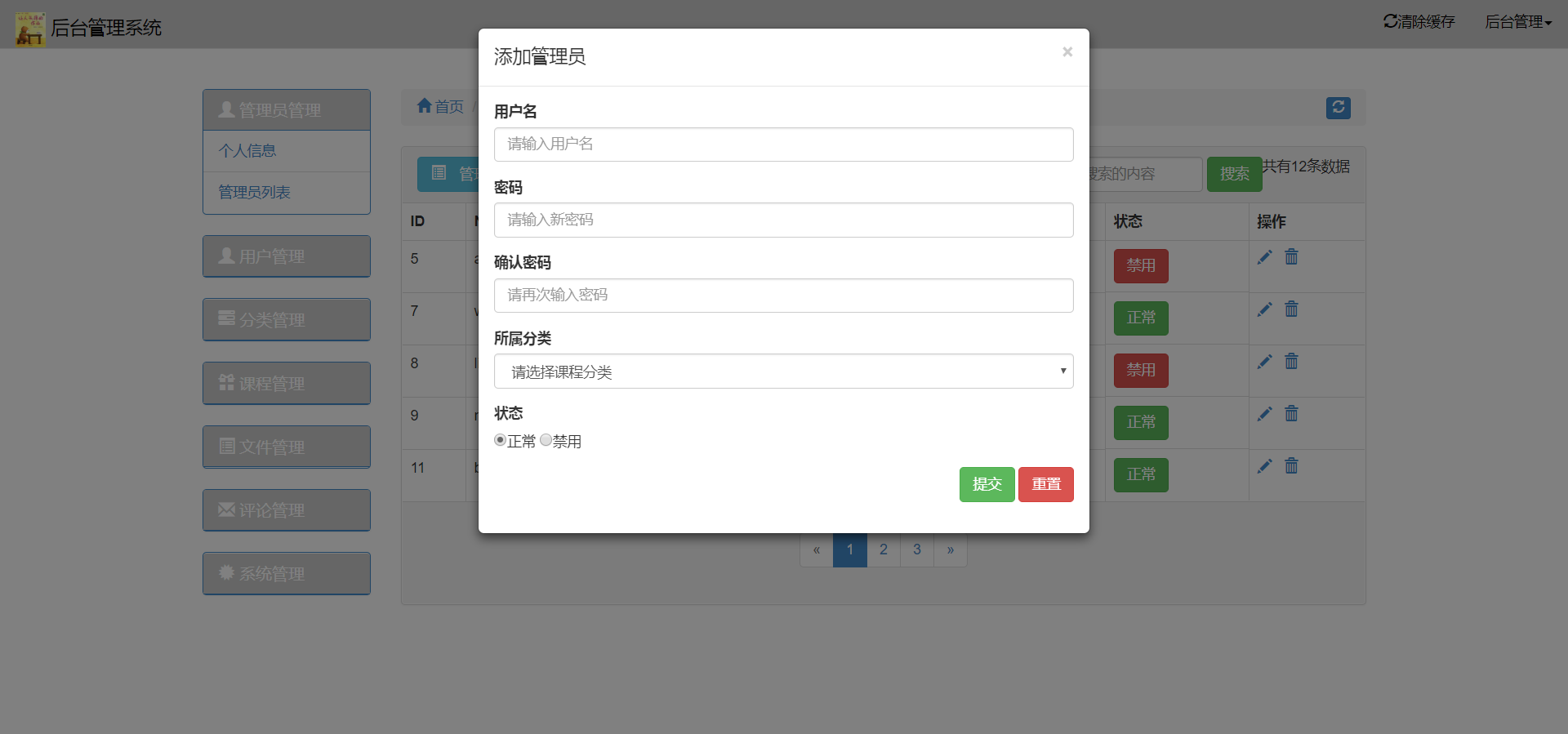


图4.4 管理员添加弹框

Fig.4.4 admin add modal

主要代码如下：

**public function** update(){  
 *parse\_str*($\_POST[**'str'**],$arr);$rules=[  
 **'password'** => **'required|same:repass|between:5,12'**,  
 ];$message=[  
 **"password.required"**=>**"请输入密码"**,  
 **"password.same"**=>**"两次密码不一致"**,  
 **"password.between"**=>**"密码长度不在5-12位之间"**,  
 ];$validator = Validator::*make*($arr,$rules,$message);**if** ($validator->passes()) {**unset**($arr[**'repass'**]);  
 $arr[**'password'**]=Crypt::*encrypt*($arr[**'password'**]);**if** (\DB::*update*(**"update admin set status= '**$arr[status]**' ,password='**$arr[password]**' where id=**$arr[id]**"**)) {**return** 1;  
 }**else**{  
 **return** 0;  
 }  
 }**else**{**return** $validator->getMessageBag()->getMessages();  
 }  
 }

### 4.2.3学校组织机构

(1) 组织机构展示页面 如图4.5

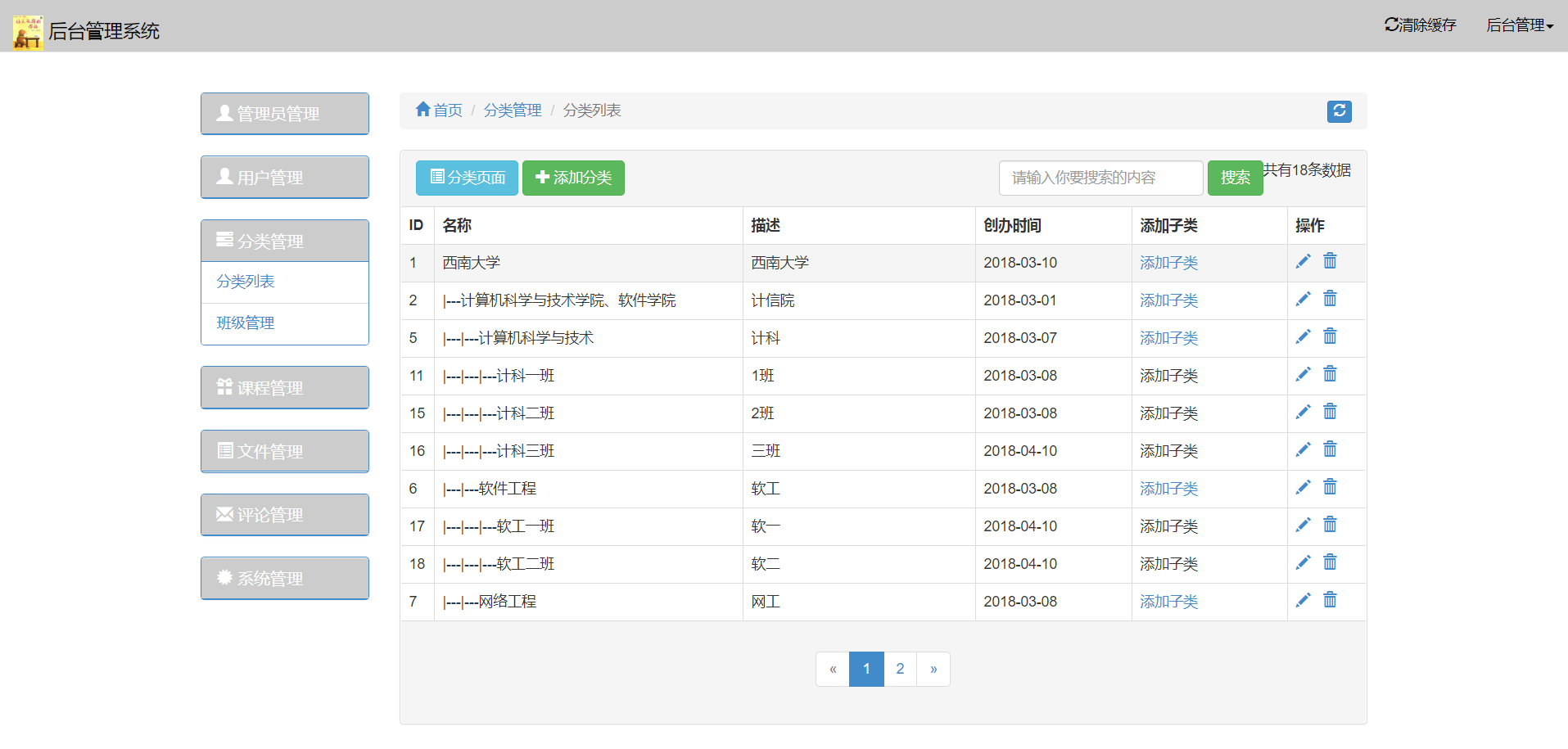


图4.5 组织机构展示页面

Fig.4.5 orgination show page

主要代码如下：

后台逻辑代码：

**public function** index(Request $request){$data=\DB::*table*(**'type'**)->select(DB::*raw*(**'\*,concat(path,id) as p'**))->orderBy(**'p'**,**'asc'**)->paginate(10);  
 $tot= \DB::*table*(**'type'**)->count();**return** view(**"admin.type.index"**)->with(**"data"**,$data)->with(**"tot"**,$tot);  
 }

前台显示代码：

<**table class="table-bordered table table-hover"**>  
 <**th**>ID</**th**>  
 <**th class="col-sm-4"**>名称</**th**>  
 <**th class="col-sm-3"**>描述</**th**>  
 <**th class="col-sm-2"**>创办时间</**th**>  
 <**th class="col-sm-2"**>添加子类</**th**>  
 <**th class="col-sm-2"**>操作</**th**>  
 **@foreach(**$data **as** $value**)** <**tr**>  
 <**td**>{{$value->**id**}}</**td**>  
 **<?php** $arr=*explode*(**'-'**,$value->**p**);  
 $kind = *count*($arr)-2;  
 **?>** <**td**>{{*str\_repeat*(**"|---"**,$kind)}}{{$value->**name**}}</**td**>  
 <**td**>{{$value->**description**}}</**td**>  
 <**td**>{{$value->**createtime**}}</**td**>  
 **@if(**$kind>=3**)** <**td**>添加子类</**td**>  
 **@else** <**td**><**a href="/admin/type/create?pid=**{{$value->**id**}}**&path=**{{$value->**path**}}{{$value->**id**}}**-"**>添加子类</**a**></**td**>  
 **@endif**<**td**><**a href="/admin/user/admin/1/edit" class="glyphicon glyphicon-pencil"**></**a**>**&nbsp;&nbsp;&nbsp;**<**a href="javascript:;" onclick="***del*({{$value->**id**}})**" class="glyphicon glyphicon-trash"**></**a**></**td**>  
 </**tr**>  
 **@endforeach**</**table**>

(2) 添加机构页面 如图4.6

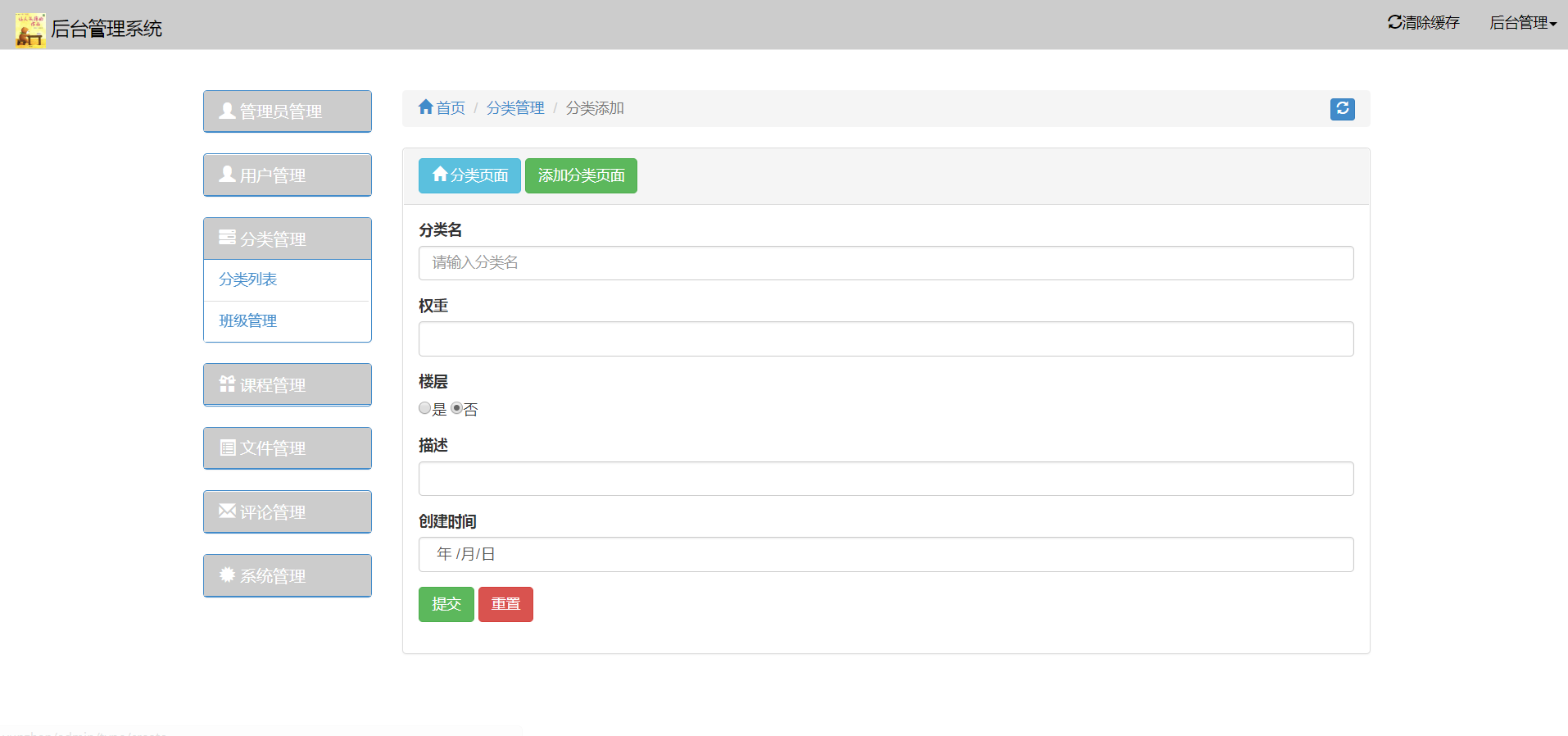


图4.6 机构添加页面

Fig.4.6 orgination add page

主要代码如下：

**public function** store(Request $request){  
 $data = $request->except(**"\_token"**);  
 $data[**'kind'**]=*substr\_count*($data[**'path'**],**'-'**)-1;**if**(DB::*table*(**'type'**)->insert($data)){  
 **return** redirect(**'admin/type'**);  
 }**else**{  
 **return** back()->with(**"error"**,**"存储失败"**);  
 }  
 }

### 4.2.4 课程相关操作

(1) 课程显示页面 如图4.7

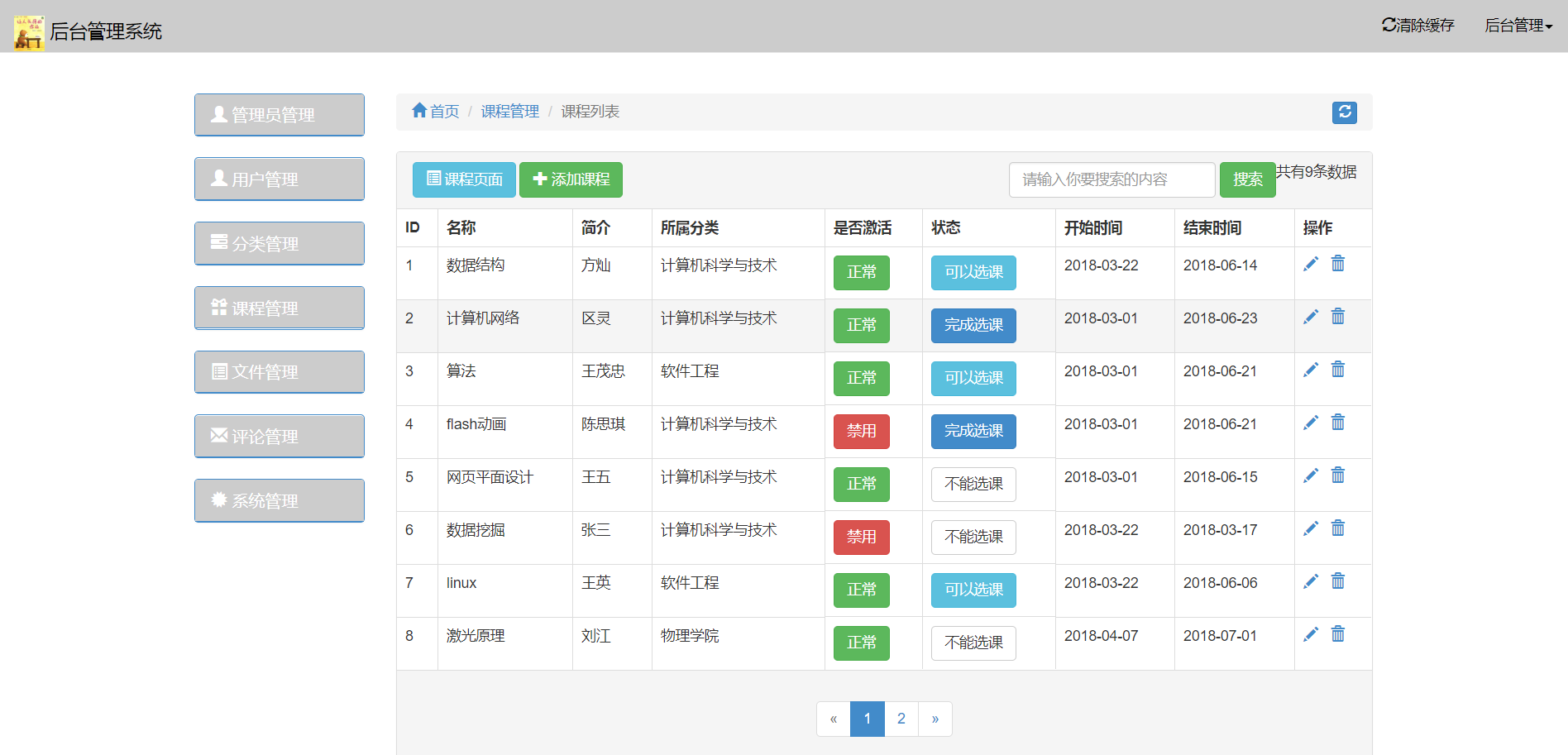


图4.7 课程显示页面

Fig.4.7 course show page

后台逻辑代码：

**public function** index(Request $request){  
 $tot = \DB::*table*(**'course'**)->count();  
 $data = \DB::*table*(**'course'**)->paginate(8);  
 **foreach** ($data **as** $vv){  
 $vv->**type** = $this->getName($vv->**typeid**);  
 }**return** view(**'admin.course.index'**)->with(**'tot'**,$tot)->with(**'data'**,$data);  
 }

前台显示页面：

<**table class="table-bordered table table-hover"**>  
 <**th**>ID</**th**>  
 <**th**>名称</**th**>  
 <**th**>简介</**th**>  
 <**th**>所属分类</**th**>  
 <**th**>是否激活</**th**>  
 <**th**>状态</**th**>  
 <**th**>开始时间</**th**>  
 <**th**>结束时间</**th**>  
 <**th**>操作</**th**>  
 **@foreach(**$data **as** $value**)** <**tr**>  
 <**td**>{{$value->**id**}}</**td**>  
 <**td**>{{$value->**name**}}</**td**>  
 <**td**>{{$value->**info**}}</**td**>  
 <**td**>{{$value->**type**}}</**td**>  
 **@if(**$value->**isable)** <**td**><**span class="btn btn-success" onclick="***status*(**this**,{{$value->**id**}},1)**"**>正常</**span**></**td**>  
 **@else** <**td**><**span class="btn btn-danger" onclick="***status*(**this**,{{$value->**id**}},0)**"**>禁用</**span**></**td**>  
 **@endif  
 @if(**$value->**isselect** == **'0')** <**td**><**span class="btn btn-default"**>不能选课</**span**></**td**>  
 **@elseif(**$value->**isselect** == **'1')** <**td**><**span class="btn btn-info"**>可以选课</**span**></**td**>  
 **@else** <**td**><**span class="btn btn-primary"**>完成选课</**span**></**td**>  
 **@endif** <**td**>{{$value->**starttime**}}</**td**>  
 <**td**>{{$value->**endtime**}}</**td**>  
 <**td**><**a href="javascript:;" onclick="***edit*({{$value->**id**}})**" data-toggle="modal" data-target="#edit" class="glyphicon glyphicon-pencil"**></**a**>**&nbsp;&nbsp;&nbsp;**<**a href="javascript:;" onclick="***del*(**this**,{{$value->**id**}})**" class="glyphicon glyphicon-trash"**></**a**></**td**>  
 </**tr**>  
 **@endforeach**</**table**>  
*<!-- 分页效果 -->*<**div class="panel-footer"**>  
 <**nav style="text-align**:**center**;**"**>  
 {{ $data->links() }}  
 </**nav**>  
</**div**>

(2) 课程分配页面 如图4.8



图4.8：课程份分配页面

Fig.4.8 course assign page

主要代码如下：

后台逻辑代码：

**public function** assign(){  
 $data = \DB::*table*(**'course'**)->where(**'isable'**,**'1'**)->paginate(8);  
 **foreach** ($data **as** $vv){  
 $vv->**tpname** = $this->getName($vv->**typeid**);  
 }  
 $tot =\DB::*table*(**'course'**)->where(**'isable'**,**'1'**)->count() ;**return** view(**'admin.course.assign'**)->with(**'data'**,$data)->with(**'tot'**,$tot);  
 }

前端页面代码：

<**table class="table-bordered table table-hover"**>  
 <**th**>ID</**th**>  
 <**th**>名称</**th**>  
 <**th**>简介</**th**>  
 <**th**>所属分类</**th**>  
 <**th**>选课性质</**th**>  
 <**th**>开始时间</**th**>  
 <**th**>结束时间</**th**>  
 <**th**>操作</**th**>  
 **@foreach(**$data **as** $value**)** <**tr**>  
 <**td**>{{$value->**id**}}</**td**>  
 <**td**>{{$value->**name**}}</**td**>  
 <**td**>{{$value->**info**}}</**td**>  
 <**td**>{{$value->**tpname**}}</**td**>  
 **@if(**$value->**type)** <**td**>通选课</**td**>  
 **@else** <**td**>专业课</**td**>  
 **@endif** <**td**>{{$value->**starttime**}}</**td**>  
 <**td**>{{$value->**endtime**}}</**td**>  
 <**td**>  
 **@if(**$value->**type)** <**a href="javascript:;" onclick="***establish*({{$value->**id**}},{{$value->**typeid**}})**" data-toggle="modal" data-target="#establish" class="glyphicon glyphicon-plus"**></**a**>  
 **@else** <**a href="javascript:;" onclick="***allocate*({{$value->**id**}},{{$value->**typeid**}})**" data-toggle="modal" data-target="#allocate" class="glyphicon glyphicon-arrow-right"**></**a**>  
 **@endif** </**td**>  
 </**tr**>  
 **@endforeach**</**table**>

### 4.2.5 教师管理页面

(1) 教师班级管理 如图4.9



图4.9 教师班级管理

Fig.4.9 teacher class manager

主要代码如下：

查看代课班级：

**public function** banji(){  
 $id=session(**'userInfo.id'**);  
 $info1 = \DB::*table*(**'teacher\_relation'**)  
 ->join(**'type'**,**'type.id'**,**'teacher\_relation.classid'**)  
 ->where([  
 [**'teacher\_relation.teacherid'**,$id],  
 [**'type.kind'**,**'3'**]  
 ])  
 ->get();$info2 = \DB::*table*(**'relation'**)  
 ->join(**'type'**,**'type.id'**,**'relation.classid'**)  
 ->where([  
 [**'relation.teacherid'**,$id],  
 [**'type.kind'**,**'3'**]  
 ])  
 ->get();$class = \DB::*table*(**'type'**)->select(\DB::*raw*(**'\*,concat(path,id) as p'**))->where(**'kind'**,**'3'**)->orderby(**'p'**)->get();  
 **return** view(**'home.teacher.type.banji'**)->with(**'data1'**,$info1)->with(**'data2'**,$info2)->with(**'class'**,$class);  
 }  
 查看代课班级学生：  
 **public function** check($id){  
 $info = \DB::*table*(**'student\_relation'**)  
 ->join(**'student'**,**'student.id'**,**'student\_relation.studentid'**)  
 ->join(**'student\_info'**,**'student\_info.id'**,**'student\_relation.studentid'**)  
 ->where(**'student\_relation.classid'**,$id)  
 ->paginate(6);  
 $tot = *count*(\DB::*table*(**'student\_relation'**)->where(**'student\_relation.classid'**,$id)->get());**return** view(**'home.teacher.type.check'**)->with(**'info'**,$info)->with(**'tot'**,$tot);  
 }

(2) 教师课程管理 如图4.10

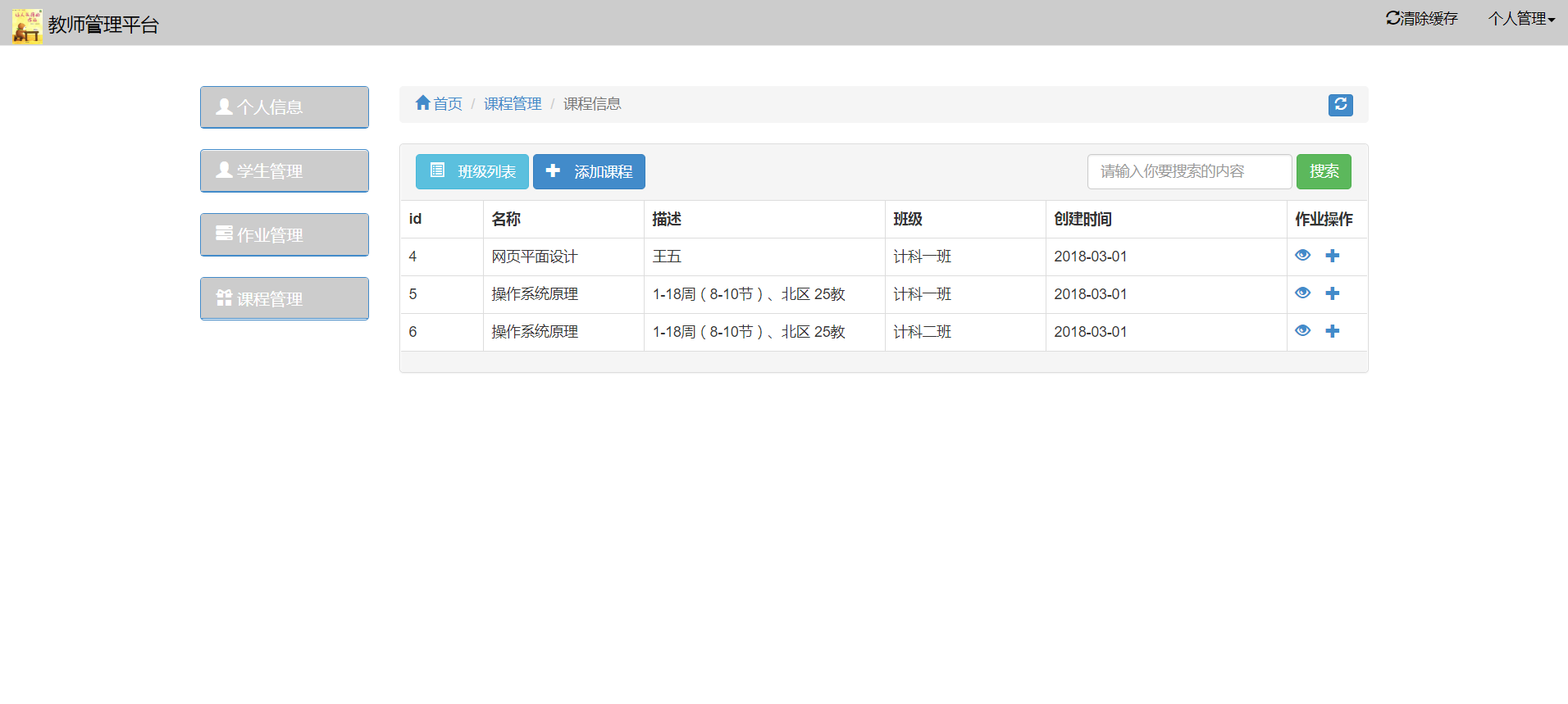


图4.10 教师课程管理

Fig.4.10 teacher course manager

主要代码如下：

显示页面逻辑

**public function** index(){  
 $id = session(**'userInfo.id'**);  
 $info = \DB::*table*(**'relation'**)  
 ->select(**'course.\*'**,**'relation.\*'**,**'type.name as name1'**)  
 ->join(**'course'**,**'course.id'**,**'relation.courseid'**)  
 ->join(**'type'**,**'type.id'**,**'relation.classid'**)  
 ->where(**'relation.teacherid'**,$id)  
 ->paginate(6);**return** view(**'home.teacher.course.index'**)->with(**'data'**,$info);  
 }

添加操作

**public function** charu(){  
 *parse\_str*($\_POST[**'str'**],$arr);$rules=[  
 **'name'** => **'required'**,  
 **'info'** => **'required'**,  
 **'typeid'** => **'required'**,  
 **'isable'**=> **'required'**,  
 **'starttime'** => **'required'**,  
 **'endtime'** => **'required'**,  
 ];$message=[  
 **"name.required"**=>**"请输入名称"**,  
 **"info.required"**=>**"请输入简介"**,  
 **"typeid.required"**=>**"请选择分类"**,  
 **'isable.required'**=>**'请选择是否开启'**,  
 **"starttime.required"**=>**"请输入课程开启时间"**,  
 **"endtime.required"**=>**"请输入课程关闭时间"**,  
 ];$validator = \Validator::*make*($arr,$rules,$message);  
 $typeid = $arr[**'typeid'**];  
 $data = \DB::*table*(**'type'**)->select(**'\*'**)->where(**'id'**,$typeid)->first();  
 $kind = $data->**kind**;**if** ($kind<2){  
 $arr[**'type'**] = **'1'**;  
 }**else**{  
 $arr[**'type'**] = **'0'**;  
 }  
 **if** ($validator->passes()) {  
 **if** (\DB::*table*(**"course"**)->insert($arr)) {**return** 1;  
 }**else**{  
 **return** 0;  
 }  
 }**else**{**return** $validator->getMessageBag()->getMessages();  
 }  
 }

(3) 教师作业管理 如图4.11



图4.11教师作业添加

Fig.4.11 E-R teacher homework add

主要代码如下：

**public function** addquestion(){  
 *parse\_str*($\_POST[**'str'**],$arr);  
 $arr[**'classid'**] = $\_POST[**'classid'**];  
 $arr[**'courseid'**] = $\_POST[**'courseid'**];  
 $arr[**'teacherid'**] = $\_POST[**'teacherid'**];  
 $time = *date*(**'Y-m-d H:i:s'**,*time*());  
 $arr[**'createtime'**] = $time;  
 $rules=[  
 **'name'** => **'required'**,  
 **'info'** => **'required'**,  
 ];  
 $message=[  
 **"name.required"**=>**"请输入作业题目"**,  
 **"info.required"**=>**"请输入作业详情"**,  
 ];  
 $validator = \Validator::*make*($arr,$rules,$message);**if** ($validator->passes()) {**if** (DB::*table*(**"question"**)->insert($arr)) {  
 **return** 1;  
 }**else**{  
 **return** 0;  
 }  
 }**else**{**return** $validator->getMessageBag()->getMessages();  
 }  
}

**public function** question($id){  
 $question = \DB::*table*(**'question'**)->where(**'courseid'**,$id)->get();  
 **return** view(**'home.teacher.course.question'**)->with(**'question'**,$question);  
}

### 4.2.6 注册登录

(1) 注册 如图：4.12



图4.12 注册页面

Fig.4.12 register page

主要代码如下：

public function add(Request $request){

$raw = file\_get\_contents('php://input');

$json = json\_decode($raw);

$type = $json->type;

// print\_r($json);

// exit();

if($type == '1'){

$table = 'student';

$table\_info = 'student\_info';

$table\_relation = 'student\_relation';

$pre = '11';

}elseif($type == '0'){

$table = 'teacher';

$table\_info = 'teacher\_info';

$table\_relation = 'teacher\_relation';

$pre = '00';

}

$name = $json->name;

$json->password = \Crypt::encrypt($json->password);

$data = [

'username'=>$json->username,

'gender'=>$json->gender,

'typeid'=>$json->typeid,

'password'=>$json->password,

'phone'=>$json->phone];

$id = \DB::table($table)->insertGetId($data);

$var=sprintf("%04d", $id);

$number = $pre.date('Y').$var;

$starttime = date('Y-m-d H:i:s',time());

$data\_info = [

'id'=>$id,

'number'=>$number,

'name'=>$name,

'email'=>$json->mail,

'starttime'=>$starttime, ];

$info = \DB::table($table\_info)->insert($data\_info);

if($type == '1') {

$data\_relation = [

'studentid' => $id,

'classid' => $json->typeid,

];

}elseif($type == '0'){

$data\_relation = [

'teacherid' => $id,

'classid' => $json->typeid,

];

}

$relation = \DB::table($table\_relation)->insert($data\_relation);

if($info){

echo('{"code":200, "message":"注册成功"}');

}else{

echo('{"code":1, "message":"注册失败"}');

}

}

(2) 登录 如图：4.13



图4.13 登录页面

Fig.4.13 login page

主要代码如下：

public function checklogin(){

$raw = file\_get\_contents('php://input');

$json = json\_decode($raw);

$type = $json->type;

if($type == '1'){

$table = 'student';

$table\_info = 'student\_info';

}elseif($type == '0'){

$table = 'teacher';

$table\_info = 'teacher\_info';

}

$username = $json->username;

$password = $json->password;

$data = \DB::table($table)->where('username',$username)->first();

$id = $data->id;

$real\_password =\Crypt::decrypt($data->password);

if($password == $real\_password){

$arr=[];

$time = date('Y-m-d H:i:s',time());

$arr['lastlogin']=$time;

\DB::table($table\_info)->where('id',$id)->update($arr);

\Session::put('userInfo.username', $username);

\Session::put('userInfo.id', $id);

\Session::put('userInfo.type',$type);

if($type == '1'){

echo('{"code":200, "message":"登录成功" ,"href":"student"}');

}else{

echo('{"code":200, "message":"登录成功" ,"href":"teacher"}');

}

}else{

echo('{"code":1, "message":"登陆失败"}');

}

}

# 展望与总结

## 5.1总结

本系统主要实现了集学校管理、作业上传、课后交流为一体的学校作业管理平台，能够为所有的同学、教师提供服务。

总的来说，系统的功能比较完善，通过首页的轮播图展示页面，可以让学生和教师了解到当前学术的最新动态，学生教师可以通过课后作业进行交流沟通，教师可以管理所代理班级的学生和课程，另外管理员可以管理学校所有组织信息，学生可以上传作业与问问题。系统用户权限的设定实现了系统的访问控制，并使用了数据加密和csrf报表验证保证了本系统使用的安全性，不会被盗用账号。

大概用了三个月左右的时间进行系统的实现，刚开始的时候打算使用yaf框架来完成本系统的工作，在小半个月时间的研究后感觉时间有些来不及，临时学了laravel框架，另外laravel作为当前市场上最受欢迎的php框架，跟着文档慢慢的开始了系统的开发。在开发中总是会碰到这样那样的问题，刚开始的时候文件上传使用的是fileinput插件，但调试几波都没能成功，又使用了uploadify插件；还有就是laravel的中间件这个工具、laravel的session与php的session时相互独立的，处于惯性思维，踩了几次的坑，但经过不断地调试与博客的查看最终解决。

通过本次的系统设计，是我对自己的未来方向更加明确，当然也巩固了之前学到的知识，最终完成一个完整的系统出来。

## 5.2进一步工作

本系统虽然整体上比较完整，但我还有几个想法由于时间的问题没能实现，比如说作业成绩的excel导出、学生教师信息excel导入；添加首页学术文章的添加与上传；文件预览功能，另外还有一些小的用户友好性处理没能处理完好，会在之后的时间内尽快完善。

就目前网站发展趋势，用户对前端的要求也越来越高，我现在主要是研究的后端逻辑，对前端也只是一知半解，css和js都采用了集成的bootstrap，虽然说bootstrap很强大，但很难满足更多的人的需求，另外就是信息的实时处理。注册的时候短信发送功能和邮件验证码发送功能都还没实现，这些都是目前系统存在的最大的问题。

# 参考文献

1. 陈昊. laravel框架关键技术解析[M]. 北京电子工业出版社，2016
2. 传智播客高教产品研发部 PHP+Ajax+jQuery网站开发项目式教程[M] 人民邮电出版社 2016
3. 传智播客高教产品研发部 PHP程序设计高级教程[M] 中国铁道出版社 2016
4. 陈小龙 php7实践指南 清华大学出版社[M] 2017
5. 卡尔森 (美) (Carlson, Josiah L.) Redis实战[M] 人民邮电出版社 2015
6. 福塔 (英) (Forta Ben) Mysql必知必会人民邮电出版社[M] 2009
7. 姜承尧 MySQL技术内幕 机械工业出版社[M] 2012
8. 肖睿 Bootstrap与jQuery UI框架设计[M] 中国水利水电出版社 2017
9. Matt Zandstra(美) 深入PHP 面向对象、模式与实践[M] 人民邮电出版社 2016
10. 上野 宣(日) 图解HTTP[M] 人民邮电出版社 2014
11. Michael Morrison(美) Head First PHP&MySQL[M] 中国电力出版社 2011
12. 李开涌 PHP MVC开发实践[M] 机械工业出版社 2013

# 致谢

通过这一阶段的努力，我的毕业论文《基于web的作业批改上传系统设计与实现》终于完成了，这意味着大学生活即将结束。回首既往，自己一生最宝贵的时光能于这样美丽和谐的校园之中。在这四年的时间里，我在学习上和思想上都受益非浅。这除了自身努力外，与各位老师、同学和朋友的关心、支持是分不开的。

　　首先特别感谢陈怀东老师的热切指导，让我在设计和完成毕业设计的时候得心应手。同时我还要感谢在我学习期间给我极大关心和支持的各位老师以及关心我的同学和朋友。正是由于他们，我才能在各方面取得显著的进步，在此向他们表示我由衷的谢意，并祝所有的老师培养出越来越多的优秀人才，桃李满天下!也祝愿我的同学的在以后的道路上越走越宽，前程似锦!

　　大学生活即将结束，我要即将步入社会，毕业设计中展示了我大学四年的学习成果，也能体现出学校的整体教育水平。