**大连理工大学本科毕业设计（论文）**

**基于微信公众平台的学籍管理系统设计与实现**

**Design and implementation of student status management system based on wechat public platform**

学 院（系）： 软件学院

专 业： 软件工程

学 生 姓 名： 吴楠

学 号： 201393223

指 导 教 师： 张政凯

评 阅 教 师：

完 成 日 期： 2017

大连理工大学

Dalian University of Technology

# 摘 要

随着互联网时代的迅猛发展，各个领域行业都参与到了其中，由此微信不仅仅作为人们聊天通信的一款软件，而其中的微信公众号更成为了一个可以查询各个领域信息的便民平台。近十亿的使用人群，每天巨大的流量，生活中的每个人都因为自己所需的东西而关注不同的公众号。

对于在校的大学生，微信更是不可或缺的一部分。从每日的信息推送，到通过关注的公众号进行信息的查询。随时随地，用手机点一点，便可获得相关信息，微信公众号正是将获取信息的途径变得更加方便，快捷。

本文将设计实现一个B/S架构的连通微信平台的学籍信息系统，其展现形式为三部分，查询学籍成绩信息的微信查询公告号，展示学院相关信息的学院微主页，用与统计和分析的后台系统。

本文网站的搭建采用前后端分离的开发模式，后端在Node的环境下，采用express框架。前端则是采用Vue （MVVM框架）和 Webpack（基于node的打包合并压缩工具）的开发方案，配合bootstrap，font-awsome等第三方开源样式库，进行整体网站页面的开发。开发完成后，进行测试上线，保证了网站的前端页面的具有一定的兼容性，后台有一定的健壮性和容错能力。

**关键词：**微信公众平台；公众号服务；统计分析；Node；

**Design and implementation of student status management system based on wechat public platform**

# Abstract

With the rapid development of the Internet era, the industry in various fields are involved, the WeChat not only as a people chat communication software, and WeChat, which became a public one can query the various fields of information platform. Nearly one billion people use the crowd, which creates a huge flow every day, and everyone in the life pays attention to different public numbers to get what they need.

For college students, WeChat is a necessary part of life. By WeChat public platform, they can get daily information and inquiries information which are concerned. Anytime, anywhere, use the phone to click a bit, you can get the relevant information, WeChat public platform makes obtaining information more convenient and fast.

This article will design and implement student management system based on a B/S architecture, to achieve connectivity of WeChat. The main includes three part, querying student achievement information thourgh Wechat platform, displaying college infotmation thourgh micro home page, statistics and analysis on the management background system.

The construction of this website adopts the development model of separation from front to back, and the back end adopts express framework in the environment of Node. The front end is used Vue (MVVM framework) and Webpack (packaged with node based compression tool) development scheme, with bootstrap, font-awsome and three party open source style library, to integrate the development of the web page. After the completion of the development, on-line testing, to ensure that the front page of the site has a certain compatibility, the background has a certain robustness and fault tolerance.

**Key Words：**Wechat public platform; Public number service; statistics and analysis; Node

目 录

[摘 要 I](#_Toc453449125)

[Abstract II](#_Toc453449126)

[引 言 1](#_Toc453449127)

[1 背景及技术介绍 2](#_Toc453449128)

[1.1 项目背景及意义 2](#_Toc453449129)

[1.1.1 项目背景 2](#_Toc453449130)

[1.1.2 项目意义 2](#_Toc453449131)

[1.2 相关技术 2](#_Toc453449132)

[1.2.1 MySQL数据库 2](#_Toc453449133)

[1.2.2 Laravel框架 3](#_Toc453449134)

[1.2.3 Nginx服务器 3](#_Toc453449135)

[1.2.4 B/S架构 3](#_Toc453449136)

[1.2.5 Ajax技术简介 4](#_Toc453449137)

[1.3 本章小结 4](#_Toc453449138)

[2 系统可行性与需求分析 5](#_Toc453449139)

[2.1 可行性分析 5](#_Toc453449140)

[2.1.1 技术可行性 5](#_Toc453449141)

[2.1.2 功能可行性 5](#_Toc453449142)

[2.1.3 管理可行性 5](#_Toc453449143)

[2.1.4 经济可行性 5](#_Toc453449144)

[2.2 需求分析 6](#_Toc453449145)

[2.2.1 功能性需求分析 6](#_Toc453449146)

[2.2.2 系统的非功能性需求 12](#_Toc453449147)

[3 系统设计 13](#_Toc453449148)

[3.1 系统整体架构设计 13](#_Toc453449149)

[3.2 系统功能模块设计 13](#_Toc453449150)

[3.2.1 学生子系统模块 13](#_Toc453449151)

[3.2.2 系统管理员子系统模块 14](#_Toc453449152)

[3.2.3 教务老师子系统模块 15](#_Toc453449153)

[3.2.4 任课老师子系统模块 15](#_Toc453449154)

[3.3 数据库设计 16](#_Toc453449155)

[3.3.1 数据库需求分析 16](#_Toc453449156)

[3.3.2 E-R模型设计 19](#_Toc453449157)

[3.3.3 数据库表设计 20](#_Toc453449158)

[3.4 前端界面设计 20](#_Toc453449159)

[4 系统实现 21](#_Toc453449160)

[4.1 系统实现说明 21](#_Toc453449161)

[4.2 用户端系统实现 21](#_Toc453449162)

[4.2.1 登录功能 21](#_Toc453449163)

[4.2.2 课程浏览功能 24](#_Toc453449164)

[4.2.3 在线测试功能 25](#_Toc453449165)

[4.3 后台管理系统的实现 26](#_Toc453449166)

[4.3.1 登录和权限验证功能 26](#_Toc453449167)

[4.3.2 课程管理功能 27](#_Toc453449168)

[4.3.3 题库管理功能 32](#_Toc453449169)

[4.3.4 任务管理功能 34](#_Toc453449170)

[4.3.5 消息通知功能 36](#_Toc453449171)

[4.3.6 完成任务功能 36](#_Toc453449172)

[4.3.7 个人中心功能 38](#_Toc453449173)

[4.3.8 在线学习功能 39](#_Toc453449174)

[5 系统测试 40](#_Toc453449175)

[5.1 功能测试 40](#_Toc453449176)

[5.2 健壮性测试 41](#_Toc453449177)

[5.3 测试总结 42](#_Toc453449178)

[结 论 43](#_Toc453449179)

[参 考 文 献 44](#_Toc453449180)

[附录A 数据库表设计 45](#_Toc453449181)

[致 谢 50](#_Toc453449182)

# 引 言

在互联网盛行的今天，几乎所有高校大学生都会用微信，以及微信公众号上的订阅号来进行信息的获取。在微信上获取信息，方便省时，这也渐渐成为了大家的一种行为习惯。而通过订阅号，输入自己教务平台的账号密码，进行网上教务平台与微信订阅号的绑定，那么以后便可通过微信订阅号来随时随地，免登录来查询成绩了。这种方式，方便快捷，许多高校的学院都有此类的公众号。

正是这种微信订阅号，将信息查询的灵活性大大提高，而我校的远程教育学院，虽然有众多的学员用户，却未能提供此类的公众订阅号。同时远程教育学院目前有PC端的网站，但是不能很好的兼容手机端的页面，为了学院更好的宣传，为了使学员更便捷的查询到自己的学籍信息，一个带有学籍信息查询功能，并且可以跳转到微主页的微信公众号，是一个很好的选择。

本文实现的内容包括两端，包含手机端和PC端，手机端则是微信公众号和以公众号为入口的微主页，以来对学院的信息进行展示，及学员信息的查询，同时配备PC端的后台管理平台，来统计公众号和微主页的流量，以及远程教育学院近些年的学院信息统计分析。

订阅公众号，点击公众号一级菜单栏的第一个按钮，便可跳转到微主页，一级菜单栏的第二个按钮组包含了学生的基本查询信息，而不论是在微主页上的浏览还是微信菜单上的点击，所有的操作都会被PC端的后台管理系统所记录，以便管理人员对流量的统计和分析。

# 

# 1 背景及技术介绍

## 1.1 项目背景及意义

### 1.1.1 项目背景

随着微信公众号的普及，人们使用微信公众号查询信息的习惯已经养成，对于微信公众的使用方法也有了一定的了解。在这样的前提下，人们对于可以更加便捷的查询到自己所需信息的微信公众号，认可度很高，愿意使用和推广这类工具。同时微信公众号，不仅可以提供查询等相关功能，也可以作为其它手机页面的入口，从微信跳转到手机页面也十分自然。

### 1.1.2 项目意义

微信公众平台的出现，让学生查询相关信息更加便利，节省了必须登录PC网站，然后登录查询的流程，一次绑定，便可在手机上实时查询。

从微信跳转到手机端的微主页，简要的展示了学院的相关信息，提供了网上报名，学习中心查询等功能，为学院的招生以及宣传起到了一定作用。

对于后台管理系统，监控流量以及数据统计分析，让学院更好的看到了，自己学员的信息，可以有针对性地调整改善学院发展策略。

## 1.2 相关技术

基于微信平台的学籍管理网站采用B/S架构，主要用到的技术有Webpack, Vue, Node, Express, MongoDb。

### 1.2.1 B/S架构

B/S是Brower/Server的缩写，客户机上只要安装一个浏览器（Browser），如Netscape Navigator或Internet Explorer，服务器安装Oracle、Sybase、Informix或 SQL Server等数据库。浏览器通过Web Server 同数据库进行数据交互。

相对于C/S架构，B/S架构的性能要求和处理事务更多是在后台，而对于前端浏览器的性能要求就降低了不少，这样无论PC端或是手机端的浏览器都可以更好的运行。这在当前移动端用户的大大增加的背景下，B/S架构有着更有利的优势。

### 1.2.2 Webpack

Webpack是一种常用的前端打包工具，它是一个模块打包机，它做的事情是，分析你的项目结构，找到JavaScript模块以及其它的一些浏览器不能直接运行的拓展语言（Scss，TypeScript等），并将其打包为合适的格式以供浏览器使用。

Webpack的工作方式是：把要打包项目当做一个整体，通过一个给定的主文件（如：index.js），Webpack将从这个文件开始找到你的项目的所有依赖文件，使用loaders处理它们，最后打包为一个浏览器可识别的JavaScript文件。在实际开发当中，Html文件和Css文件都会被打包到这个JavaScript文件当中。同时它还可以在打包的过程中帮助我们把一些预编译的Saas等文件，处理为浏览器可以识别的Css文件。

### 1.2.3 Vue

Vue是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，它不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与单文件组件和 Vue 生态系统支持的库结合使用时，Vue 也完全能够为复杂的单页应用程序提供驱动。

总的来说，Vue是一种目前公司常用的MVVM框架，它体积小，操作灵活，上手快，可以单独引入来开发小型网站，也可以配合相关的vue插件来开发复杂的网站应用。

### 1.2.4 Node

Node.js是一个Javascript运行环境。实际上它是对Google V8引擎进行了封装。V8引 擎执行Javascript的速度非常快，性能非常好。Node.js对一些特殊用例进行了优化，提供了替代的API，使得V8在非浏览器环境下运行得更好。

### 1.2.5 Express

Express 是一个基于 Node.js 平台的极简、灵活的 web 应用开发框架，它提供一系列强大的特性，帮助你创建各种 Web 和移动设备应用。

本文中的网站主要用此框架中的一些路由，http接口，让底层封装的更优雅。

### 1.2.6 MongoDB

MongoDB 是一个基于分布式文件存储的数据库。由 C++ 语言编写。旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。

MongoDB虽然使用方便，可以插入各种格式，但是如果前期的梳理不当，会导致数据库中存储的数据十分混乱，一些关系型数据库拥有的例如插入之前的类型检测等，在MongoDB中都不存在了，任何数据都可以被插入，所以一些逻辑的验证和主键依赖等，应该在数据插入数据库之前，在后台处理好。

## 1.3 本章小结

本章主要介绍了基于微信的学籍管理网站的创建背景及意义，同时介绍了开发网站的相关开发技术。简要的介绍了本系统采用的数据库管理系统MongoDB的特点和优势，Node服务器的特性，B/S架构的优势，以及前端框架及打包工具的特点。

# 2 系统可行性与需求分析

## 2.1 可行性分析

可行性分析是分析开发的软件是否有价值开发、估算软件成本、分析收入、描绘出流程图、与用户分析讨论征求意见、最后生成可行性研究报告。对于软件项目而言，可行性分析是必须要结合当前实际情况的。

本系统的可行性分析包括以下几方面的内容：

### 2.1.1 技术可行性

技术可行性是根据当前技术是否能够实现系统平台的功能，以及是否能满足用户的需要。

首先本系统是是基于B/S架构的，这种架构已经相当成熟，开发流程也十分清晰。

其次无论在前端与后台上的技术类型选择，都选择的是目前流行并且有稳定人员维护的第三方开源框架，选择第三方开源框架有着诸多好处，对于框架使用的疑问，会有众多热心开发人员的解答，并且即使出现因为框架本身带来的bug也可以很快的解决（例如在github上对vue框架提issue，如果是框架本身问题，那么隔天就会有人提出解决方案）。

对于数据库的选择，在对MySQL与MongoDB进行比较之后，选择了后者，因为将数据存储在物理内存中，拥有适量级内存的MongoDB性能是十分高效的，并且插入读取的格式都为JSON型，与Node.js有更好的衔接。

### 2.1.2 功能可行性

为了使学生查询信息更便利，所提供的微信公众号绑定学院平台账号功能，让同学查询学籍信息更加便利。

微主页里面包含的学院介绍，学院动态，热门专业，网上报名，学习中心查询，通知公告，学习帮助，联系大工等一系列咨询内容，让学院信息可以得到更好推广。

后台管理平台包含的流量统计和学员信息统计分析，可以帮助学院的工作人员更好的了解到学院近些年的发展情况和学员信息。

### 2.1.3 经济可行性

对于开发过程中，认证微信公众号需要300元，租用阿里云的ECS服务，每月近100元，这些都是可以负担的。本系统开发完成后，虽无直接的利益收益，单可以给学院带来更好的宣传和用户体验，有助于学院的长久发展。

### 2.1.4 维护可行性

考虑到后期维护的工作，本系统开发文件设计严格遵从规范，代码风格严格遵循代码规范，并有必要的说明文档。

## 2.2 需求分析

需求分析是在开发之前，对所有要开发功能的一个梳理。本章将从功能性和非功能性需求分析两方分别进行展开。

### 2.2.1 功能性需求分析

功能性需求分析是开发者为用户提供一定的使用功能，以来满足用户的需求。

从功能性需求分析上来看，网站分为2个子系统：学生功能模块，管理员功能模块。

（一）学生功能

学生功能模块的实现将在微信公众号里和微主页上。

对于学生查询个人信息，以及了解学院近期动态的需求。

a 查询个人信息

学生可以登录微信，关注公众号。将微信号绑定自己注册过的网上教务平台账号，进行相关学籍信息的概要查询。绑定以后，以后查询不需要重复的输入账号密码，并且账户有所变动，可以更改自己绑定的账号密码。

b 了解近期学院动态

通过微信公众号菜单栏的按钮链接，跳转到学院相关的微主页，进行学院相关信息的浏览。

同时，由于微主页是独立的手机端网站，除了微信公众号菜单栏中的按钮链接作为入口，以后还可以开放其它的入口。学院微主页提供的网上学习中心查询，在线报名等内容，对于学院招生提供了更加便利的途径。

学生功能用例图如 图2.2

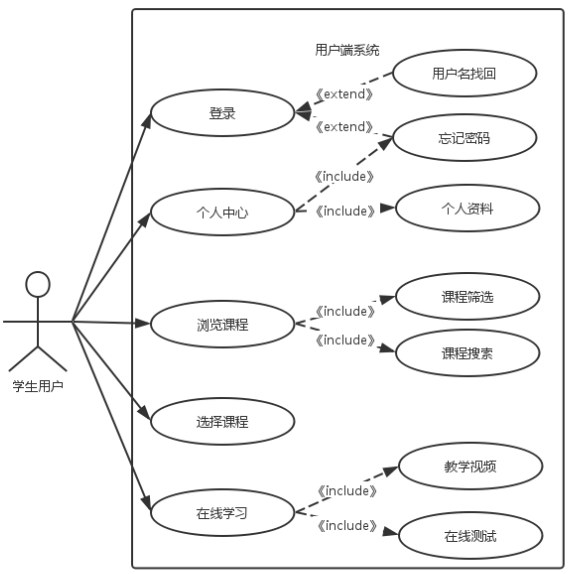


图2.2 学生功能用例图

1. 管理员功能

网站的管理员需要对网站的运营情况有基本的了解，并且学院需要对近几年的学院信息有一个汇总统计，以来了解学院的整体情况。

根据以上两点需求，管理员模块可以分为三部分：

网站的流量的统计：对网站的运营情况有基本的反馈，显示最近一周的网站访问量和访问人数。

学员信息的汇总：学院的招生来源有基本的显示，包括显示学员的年龄段，报考类型，学籍状况。

年度汇总：明确反馈出近年的热点信息，包括热门专业，热门报考中心，近年的招生情况，帮助学院更好的规划来年的招生计划。

管理员功能用例图 如图2.3

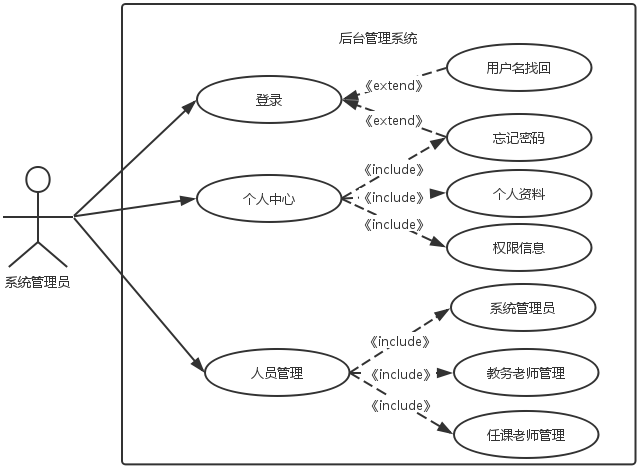


图2.3 系统管理员功能用例图

### 2.2.2 系统的非功能性需求

软件的需求可以分为功能性需求和非功能性需求，当功能性需求开发完成后，非功能性需求往往可以决定软件产品的最终质量，进而影响一个软件的成败。

非功能性需求往往来自用户体验，而用户体验的好坏，受到网站稳定性和速度，UI界面的影响。为了网站的稳定，本站的服务器选择了阿里云的ECS服务，并且近可能的用后端模板进行html的解析，同时配和AJAX进行异步加载，保证了网站基本访问速度。对于UI界面，采用色调简约的bootstrap样式库和简洁的font-awsome的字体图标库。同时对于手机页面在遵循一定的设计规范的同时，采用自适应的布局，以此来适配各种不同尺寸的机型。

# 3 系统设计

本节将按照实际的开发流程，从功能设计，界面设计，技术选择，代码管理，数据库设计这几方面分别展开。

## 3.1 系统整体架构设计

本网站分为2按用户需求分为两大模块：学生用户模块，管理员模块。其中学生模块主要提供给学院的注册学员进行信息查询和学院周知的浏览。管理员模块用于微主页流量和学院情况的统计分析。

多角色多媒体在线教育系统

后台管理系统

用户端系统

题库管理

题库审核

人员管理

课程管理

课程预览

在线学习

在线测试

任务管理

图3.1 系统架构设计图

## 3.2 系统功能设计

下面分别介绍各模块功能的具体设计：

（一）学生模块

学生模块的实现，在微信公众号和微主页上。

a 微信公众号

在微信公众号里的有一个菜单栏，包含3个按钮：

点击第一个按钮‘微主页’可以跳转到微主页。

点击第二个按钮‘个人信息’中的子按钮（我的信息，我的成绩等），会先进行微信号验证，判断是否绑定了该微信号。如果没有绑定，则会回复一个包含跳转链接的话术，点击链接跳转到绑定页面，进行微信号和教务号的绑定。如果进行了绑定，则回复相关查询信息。

点击第三个按钮‘联系我们’，可以跳转到微主页的联系界面（包含联系电话和在线qq咨询）。

b 微主页

微主页包含一个头部导航栏，可以快速跳转到主页和个人信息页，在未登录的时候会显示登录按钮，登录以后则显示退出登录按钮。

微主页主体由学院介绍，学院动态，热门专业，网上报名，学习中心查询，通知公告，学习帮助，联系大工，个人信息（我的信息，我的学习进度，我的考场，我的成绩）这12部分组成。其中个人信息部分需要先进行登录才可以查看。

补充：微信公众号和微主页都提供个人信息的查询，只不过登录时效和显示内容不同。

登录失效方面：公众号为一次绑定，永不过期。微主页则是设置了时效为一天的过期时间，过期要重新登录。

显示内容方面：由于微信公众号回复内容长度和种类的限制，在微信公众号查到的是个人的重点概要信息（例如：我的成绩只显示学生的平均分，gpa）。而在微主页上则会显示更多更详细的信息包括每科的具体成绩和已经通过的考试科目等。

（二）管理员模块

通过PC端的后台系统，管理员可以看到3类数据指标。

第一类：为学院微主页的流量，包含PV和UV两部分。

第二类：远程教育学院自网上平台开始录入数据到2017年5月所有注册学院的年龄，层级，学籍状态汇总。

第三类：年度汇总，汇总了近两年最热门的专业，学习中心和招生中心。

## 3.3 界面设计

PC端和手机端的页面均采用bootstrap风格，自适应屏幕大小。页面中的图标使用的是font-awsome字体库图标，与整体风格很好搭配的同时，省去了加载图片所带来的延迟，加快了页面的渲染速度。

手机端从上到下的整体布局为：导航栏，轮播图， 主体内容，页脚。其中导航栏中包含主页和个人信息的跳转，帮助用户在浏览网站时候的快捷操作。

PC端的布局格式由上到下为：导航栏，主体内容，页脚。

以上的布局格式，是采用现在网站开发的常用布局，设计简单，风格清新，用户学习成本低，更易接受。

## 3.3 技术选型

整体开发模式为前后分离的开发模式，前后端通过ajax进行交流。

### 3.3.1 前端选型

a MVVM架构选择原因和基本介绍

MVVM架构选择原因：

1. 首先，MVVM框架可以更好的将事件绑定从html中分开。在用原生js或jq做传统的事件绑定时候，往往根据html中的一个dom节点的某个属性（如id，class等），进行事件的绑定，这样造成了绑定事件对dom有着强依赖，从而造成html和js之间的耦合。MVVM架构基于虚拟节点，不会根据dom节点的属性进行绑定，只要在需要绑定的位置，加上事件的声明即可，解耦了html和js之间的耦合。
2. 其次，对前端来说，相较于MVC架构，MVVM架构更适合前端。因为对于C控制器这一部分，包括路由和对数据的处理，往往是后台在做。而前端更关注的是页面的渲染，以及数据与视图之间的联系，所以MVVM架构应运而生，去掉C这一概念，引入了VM，即指代一个视图与模型的相互绑定：视图的改变可以影响到数据的改变，反过来，数据的改变也会影响视图的改变。

MVVM基本原理：



在js中定义了Vue实例后，其内部会有一张数据表，以来记录实际dom节点和数据的映射关系，在通过Vue对表中数据与视图的检测，做到双向绑定。VM在框架中的具体实现就是，一个页面中的dom属性和一个Vue实例的数据进行绑定，在html页面中，当dom属性被一些事件改变的时候（如输入字符会更改dom节点input的value值），与之绑定的Vue实例的数据也会随之改变。

b MVVM架构的框架选择-Vue

确定了前端使用MVVM架构开发后，比较了多种MVVM框架（包括Ractive, React等）。从运行效率，扩展性，学习成本，框架关注度这几方面考虑后，选择了vue框架。Vue本身体积很小，页面的渲染是基于虚拟dom树进行的，所以渲染效率高。并且vue配有丰富的第三方插件，方便以后做扩展。同时vue拥有完善的api和文档，方便学习，更易上手。最后vue的热度在国内外都很高，并且有专门的人员维护，框架本身如果有bug，也会有人来解决。

c 打包工具的选择-webpack

比较了gulp等打包工具后，选择了配置更加便捷，更注重前端代码合并压缩的webpack，并且webpack可以和vue框架更好的组合（是目前公司里十分流行的一种组合方式）。Webpack会根据入口的js文件自动找到依赖的文件，从而打包到一起。

前端技术小结：前端所处理的主要逻辑使用MVVM框架vue编写，编写完成后使用webpack将逻辑中所引用到的html，css进行文件合并，打包压缩处理成一个js文件。样式库选择bootstrap进行整体布局，自适应屏幕，选用font-awsome字体图标代替图片。同时引入jquery库来简化代码并兼容各个版本的浏览器，移入moment库来进行对时间的规范处理。引入第三库echarts，进行对图表信息的绘制。

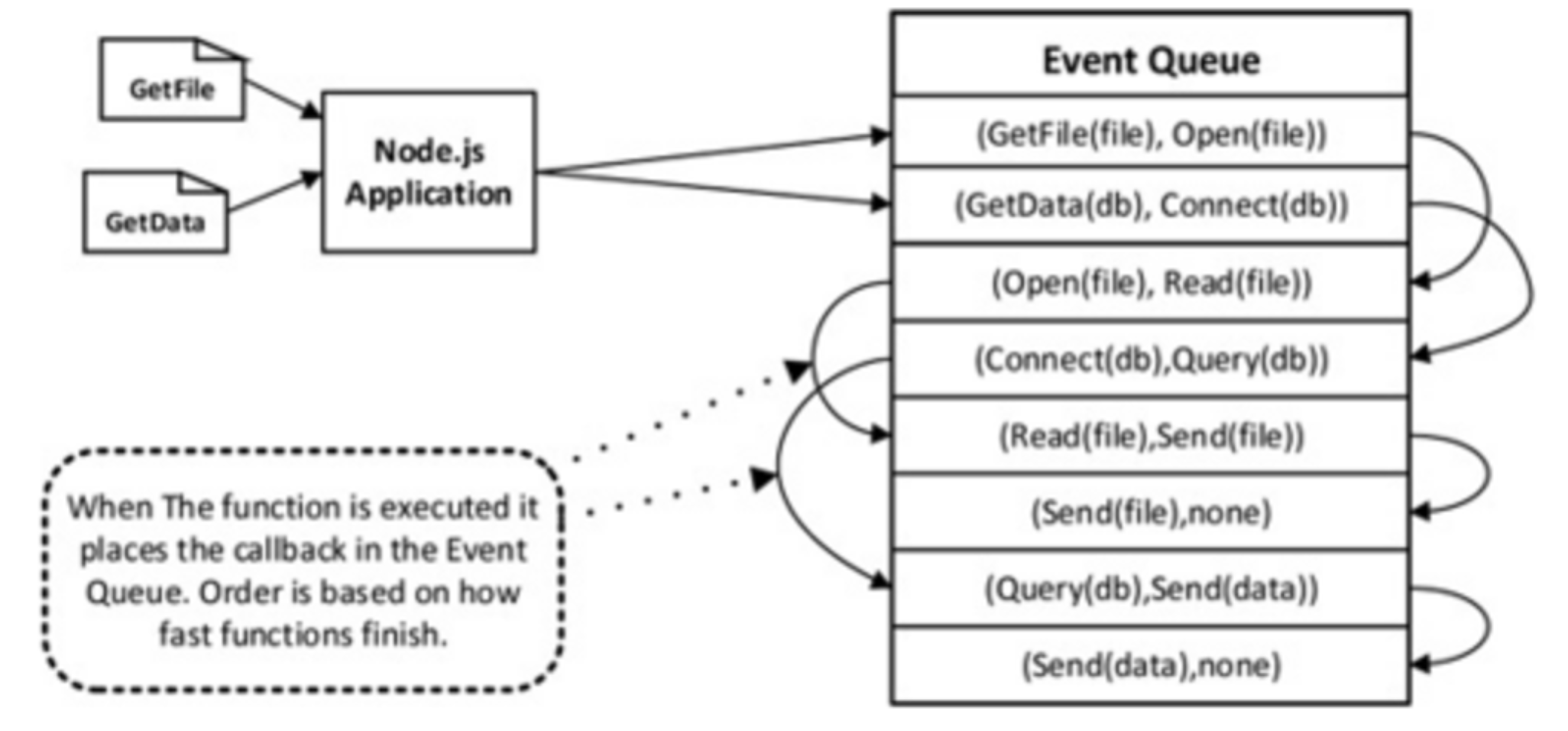
### 3.3.2 后端选型

选择node为服务器环境，搭配express框架进行开发。

a Node基本介绍和特点

Node是一种运行后台服务的环境，它可以解析用Js语言编写的程序，它也提供了丰富的系统接口（如读取文件等），同时在github上有众多基于node编写的第三方中间件，以来方便开发者调用一些接口更加的便捷和舒适。

相对于其它后端环境，node.js有着非阻塞，事件驱动I/O等特性，让高并发的访问不在卡死。由于本身是事件驱动的特性，node.js在面对众多用户的时候，不会像php一样分配多个进程，同时服务，它会让后来的用户排队，等前面的用户服务完，再回调后来的用户，来开始后面用户的调度。这样的好处是，面对超高并发访问的时候，同时分配进程执行任务，很可能导致服务器的崩溃，而node.js的这种模式，虽然让用户等待时间有些长，但是不会导致服务器直接卡死或者崩溃。淘宝等一些有着巨大的并发量的公司已经将公司的部分服务的后台转为node.js了。



b Expres框架

Express是一个基于Node.js平台的灵活的web应用开发框架，有着快捷的http接口，和丰富的中间件，它是对node.js的二次抽象，并进行了相应的web扩展。

Express提供restful api。这正是后台经常处理http请求所需要的，而且express本身提供的中间件，省去了很多重复的逻辑处理，可以更关注于业务逻辑。

### 3.3.3 数据库选型

本网站需要将微主页的访问情况（pv,uv），微信用户的相关信息，微信用户的绑定信息等存入数据库。需要一个轻便快捷，处理效率高的数据库。所以选择数据库的时候，主要对Mongodb和Mysql进行了比较:

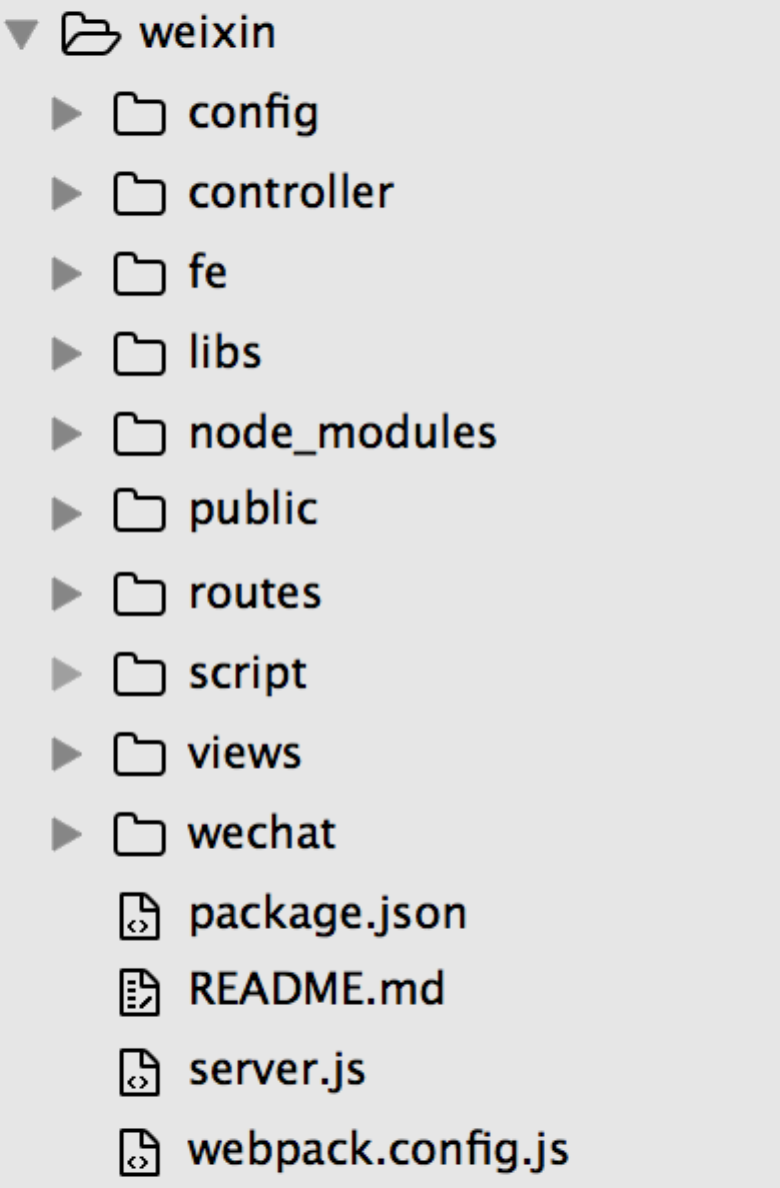
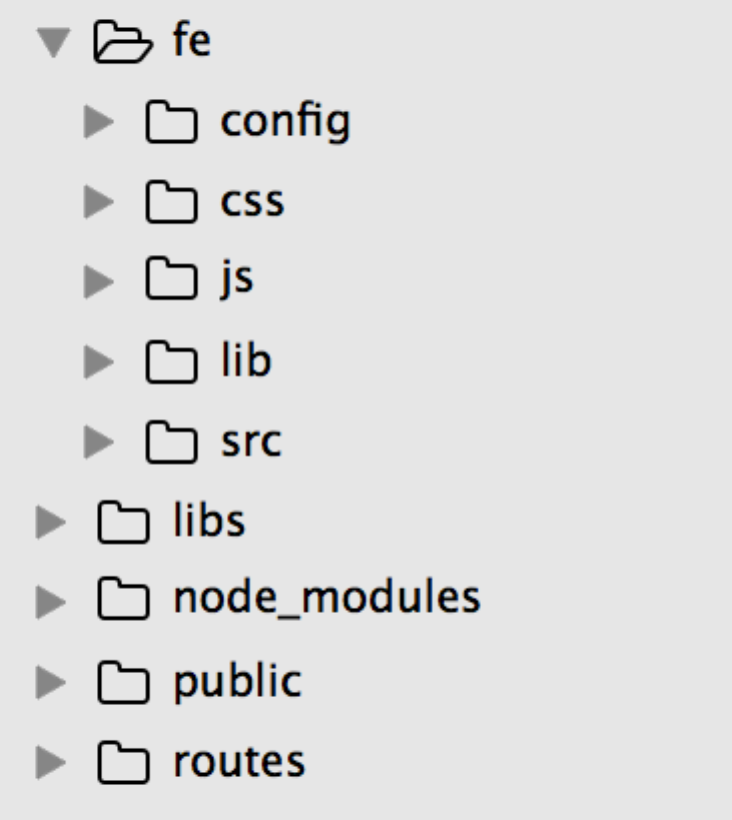
Mongodb 与 Mysql对照表

|  |  |  |
| --- | --- | --- |
|  | Mongodb | Mysql |
| 数据库模型 | 非关系型 | 关系型 |
| 存储方式 | 虚拟内存 + 持久化 | 不同引擎方式不同 |
| 查询语句 | 独特Mongodb查询方式 | 传统Sql语句 |
| 数据处理方式 | 基于内存，将热数据  存在物理内存中，从而达到高速读写 | 不同引擎存储方式不同 |
| 广泛度 | Nosql数据库中，最为完善的数据库之一，  使用人群在不断增加 | 随着开源数据库的份额不断增加，  mysql的份额也在不断增加 |

经过一些初步比较和网上参考，Mongodb有着非关系型数据库的一些优势，读写速度快（经测试，三万条数据读写的时间都可以保持在几百毫秒内），读写方式采用json格式，可以和node.js有着很好组合。对于本网站来说，一个轻便快捷并且读写速度快的数据库是很不错的选择，还能与后台良好的搭接（node + express + mongodb是很多全栈开发人员衷爱的选择），所以最终选择了Mongodb作为数据库。

### 3.3.4 代码相关设计

a 代码目录结构

1) 文件

server.js：服务器的启动文件，每次要用node启动该文件，以来开启后台http等相关服务。

package.json：node环境所需的第三方包的依赖表。

webpack.config.js：webpack进行打包时，所执行的脚本文件。

README.md：含本站的一些注释性说明。

2) 文件夹

fe：所有前端的内容都放在这里。fe里js文件夹存放打包后生成的js文件，src为打包之前的文件，按功能划分文件夹，每个文件夹里包括html,css,js。

routes：存放后端路由文件。

controls：存放后端控制器文件。

view：存放后端模板文件。

public：存放后端静态文件。

node\_modules：存放node依赖的第三方库。

wechat：存放开发时，自定义的common规范的模块。

## 3.3 数据库设计

数据库设计[13](Database Design)是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）。本文建立的数据库表大致符合第三范式要求，已达到精简数据存储的目的，部分表结构会重复存储，这是为了优化查询性能造成的数据冗余。

### 3.3.1 数据库需求分析

设计一个性能良好的数据库系统，明确应用环境对系统的要求是首要的和基本的。因此，应该把对用户需求的收集和分析作为数据库设计的第一步。需求分析的重点是调查、收集和分析用户数据管理中的信息需求、处理需求、安全性与完整性要求。信息需求是指用户需要从数据库中获得的信息的内容和性质。

根据系统的设计要求，多角色多媒体在线教育首先需要用户表，这里分为学生表和管理员表，来存储用户相关信息。学生可以进行选课，并进行相关学习，所以这里还需要课程表供学生筛选，为了区分课程属于那块分类，还需建立课程分类表。管理员是分为多个角色的，因此我们需要区分管理员角色信息，由此可以引入角色表，与之相对应的角色权限表也需要建立。从管理员功能来看，管理员需要发布任务，需将任务信息存储下来，并通知给相应的接收人，由此可以建立任务表和工作表。在任课教师完成各部分任务工作时，需要添加题库等信息，因此我们需要记录上传的题目，建立问题表。还有一些详细的数据表信息将在下面接着论述，在此不做展开。

针对多角色多媒体在线教育系统的功能要求，该系统所要完成功能的数据流图如下图所示：

多角色多媒体在线教育系统的顶层数据流图如图3.6：

图3.6 系统顶层数据流图



由于该图较为复杂，分层画出数据流图是最好的方式，但是由于篇幅过长，此处不列出各个分层数据流图，总的数据流图如图3.6：

图3.7 系统总的数据流图



### 3.3.2 E-R模型设计

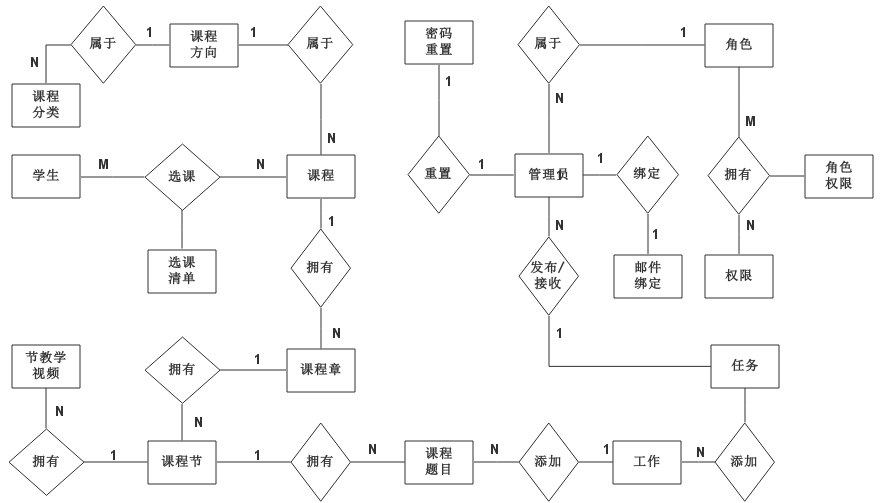


图3.8 系统ER图分析

学生实体是用来记录学生的基本信息，课程实体是用来记录课程的基本信息，从图3.8中，可以看出学生实体和课程实体是多对多的关系，根据数据库设计的第三范式，需要实体联系集选课清单来记录学生与课程之间的关系。课程分类与课程是1对多的关系，应将课程分类主键放在多的一端课程中。工作实体与课程题目之间也是1对多关系，一个工作完成后可以添加多个题目。任务实体与工作实体之间也是1对多关系，一个任务可以发布多个教师的工作，因此将任务主键放在工作实体中。管理员实体与任务实体是1对多关系，一个管理员可以发布多条任务。角色实体与管理员实体是1对多关系，1个角色对应多个管理员。角色与权限之间是多对多关系，因此衍生出实体联系集角色权限。密码重置和邮件绑定实体与管理员都是1对1关系。

### 3.3.3 数据库表设计

数据库表的建立是非常重要的环节，它涉及数据是否冗余，逻辑代码是否复杂，系统性能等各个方面的问题。因此本次数据库表的建立主要是根据上述数据库需求分析和ER图，在大致满足第三范式的要求下设计数据库表，部分容易是考虑到系统性能和逻辑复杂度而作的优化，由于表数据较多，该部分表设计放在附录A数据库表设计中，详情请查看附录A。

## 3.4 前端界面设计

系统管理界面采用AdminLTE主题，它是基于Bootstrap的AdminLTE免费高级管理控制面板主题。AdminLTE 是一个完全响应式管理模板。基于Bootstrap3框架。高度可定制的，易于使用。适合从小型移动设备到大型台式机很多的屏幕分辨率。浏览器支持如IE9+、Firefox 5+、Chrome 14+、Safari 5+、Opera 11+等多种。JavaScript插件包和CSS包主要用到Bootstrap、DataTables、Jquery、Select2等。

前端界面设计主要是采用AdminLTE前端开源模板，进行定制型开发，筛选合适的框架组件进行合理的设计，形成良好的用户界面，详细的用户界面请参考系统实现中的界面设计。

# 4 系统实现

## 4.1 系统实现说明

本系统的开发环境是搭建在Linux系统上，采用开源的Nginx服务器，数据库采用开源的MySQL，PHP后端采用Laravel开源框架，前端使用AdminLTE模板和Ajax技术实现异步操作。本章节根据系统架构，将分别介绍系统的管理后台和用户端系统的实现。

## 4.2 用户端系统实现

### 4.2.1 登录功能

登录功能是后台管理的最基本的功能之一，用户必须登陆后才可以进入系统，进行下一步操作。同时登录界面提供忘记密码操作，提供用户进行密码找回操作，登录的系统流程图和密码找回流程图如图4.1：

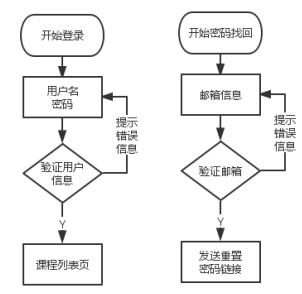


图4.1 登录系统和密码找回流程图

用户登录时，一般的错误信息都会是用户名或密码错误，用户填写正确密码时即可成功登录。用户在进行密码找回时需事先绑定邮箱，如果为绑定邮箱，提示的错误信息是请联系教务老师重置密码，而后进行密码修改操作。如果已绑定邮箱，则登录邮箱后打开相应链接即可进行密码重置和修改的操作。

用户的登录界面如图4.2：

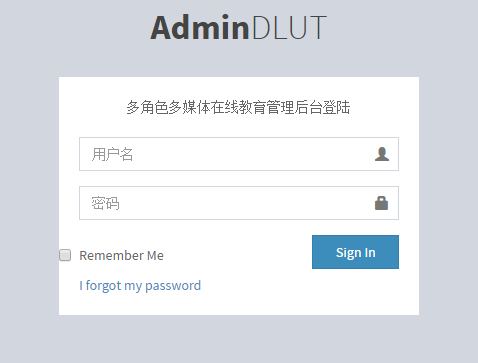


图4.2 系统登录界面

进行登录操作时，采用对用户名+IP+用户类型进行Base64编码，密码进行md5算法加密操作，存入相应的Cookies。

在判断用户是否登录时，我们检验相应的Cookies，对用户名的Cookies进行Base64解密操作，获取用户的用户名，再去数据库相应表获取相应密码，进行md5加密，最后进行IP地址比较，密码的比较，如果检验通过则说明该用户登录验证通过。

下面是登录模块的简要登录代码：

$this->user\_id = $result[0]->id;

$this->username = $username;

$this->ok = true;

$username = $username.$this->depr.$this->get\_ip().$this->depr.$type;

$user\_name = $this->encrypt($username,'E',$this->salt);

$\_SESSION[$this->pre\_username]=$user\_name;

$\_SESSION[$this->pre\_password]=md5(md5($password,$this->salt)); setcookie('user\_id',$this->user\_id ,time()+$this->cookie\_time);

setcookie($this->pre\_username,$user\_name,time()+$this->cookie\_time); setcookie($this->pre\_password,md5(md5($password,$this->salt)),time()+$this->cookie\_time);

return true;

}

在各个页面我们需要验证该用户是否登录和具有相应的权限，如果在代码中每个Controller中重复该部分检验冗余，因此我们Laravel框架中的HTTP中间件的机制来过滤进入应用程序的 HTTP 请求，因此我们将验证用户身份的登录模块注册为中间件，如果用户为通过登录验证，中间则会将用户导向登陆界面，反之，当用户通过登录验证，中间件会通过此请求并继续该次请求。注册中间件代码如下：

public function handle($request, Closure $next)

{

$AuthSvc = new AuthSvc();

$check\_cookie = $AuthSvc->check\_cookie();

if($check\_cookie){

return $next($request);// 通过验证，继续该请求

}else{

return redirect('admin/login')->with("errors","没有权限！");

}

}

在登入界面可提供了用户名和密码找回的功能，当用户忘记用户名时，可以使用已绑定的邮箱来找回用户名，系统会向绑定的邮箱发送遗忘的用户名信息。邮件服务是采用Laravel中的Mailgun驱动，每月将有1万封免费邮件可以发送。

当用户忘记密码时，可以使用密码重置功能，该功能也是通过邮件发送相关验证链接至用户绑定邮箱，系统会根据用户信息生成加密串，作为相应的链接发送至用户邮箱，用户登录邮件后点击验证链接后跳转至密码修改界面。密码重置流程图如图4.3：

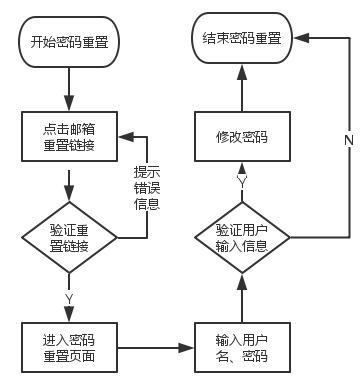


图4.3 密码重置流程图

该页面会获取加密、用户名、新密码、用户Id信息，去后台做相关检验，会判断加密串是否跟密码重置数据表中存储的是否一致，用户填写的用户名是否跟该条记录的请求发送者是否一致，如果相同则更改用户的相应密码，否则提示错误信息。

### 4.2.2 课程浏览功能

课程浏览是学生用户的基本功能，用户登录系统后即可以看见课程列表界面，课程列表界面展示的是现有并已对用户放的课程，用户可以筛选课程分类来检索相关的课程如前端开发、后端开发等类型，用户可以根据自己的爱好和方向选择合适自己的课程进行报名和学习。同时提供搜索功能，供用户精确检索课程信息。课程浏览首页界面如图4.4所示：

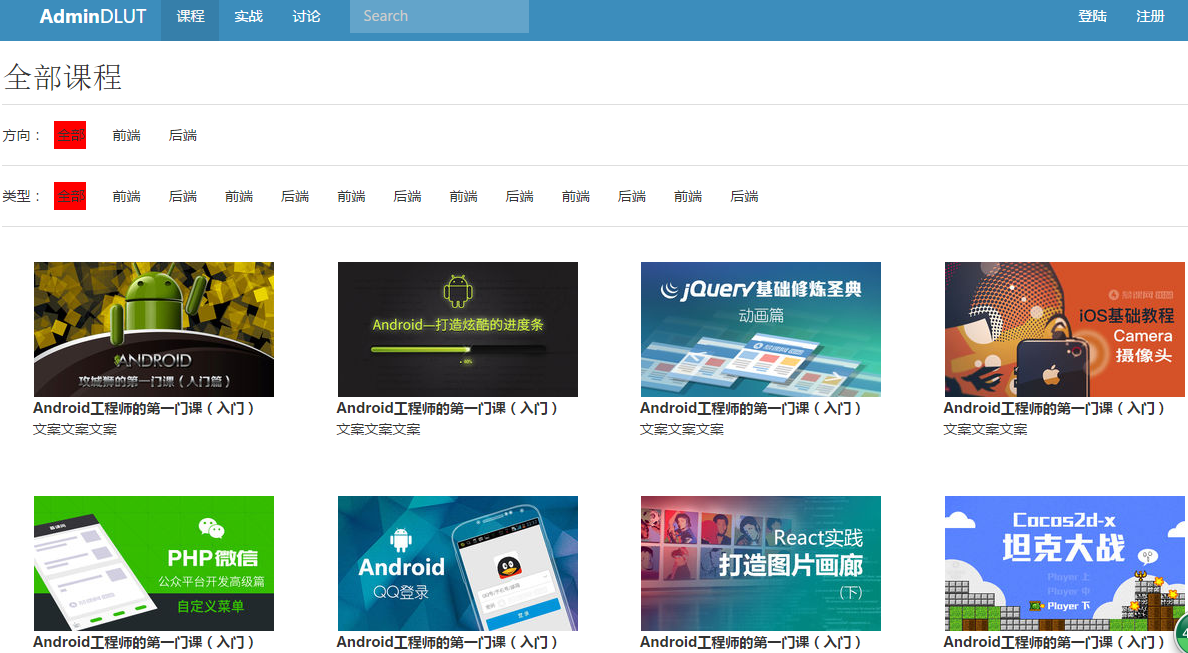


图4.4 课程列表界面

用户点击相关课程，可以进去查看该门课程的预览信息，包括任课教师信息，课程简介，课程的章节信息，课程的学习人数等信息。

当学生用户点击开始学习，即选择了该门课程，将跳转至该课程的学习页面，用户可以进行课程的学习和简单测试等功能。



图4.5 课程预览界面

### 4.2.3 在线测试功能

当用户学习完成课程的某一章节时，可以选择章节测试，在视频播放页面中点击章节测试按钮，即可跳转至课程章节测试界面。该界面的数据为后台管理系统的任课老师上传的章节测试题库。在用户作答题目后，点击提交按钮后，答题数据会进行解析，对比是否正确，并将各题目的正确与否存于题目错误记录表中，提供给后台管理员进行题目错误率的参考。章节在线测试界面如图4.6：

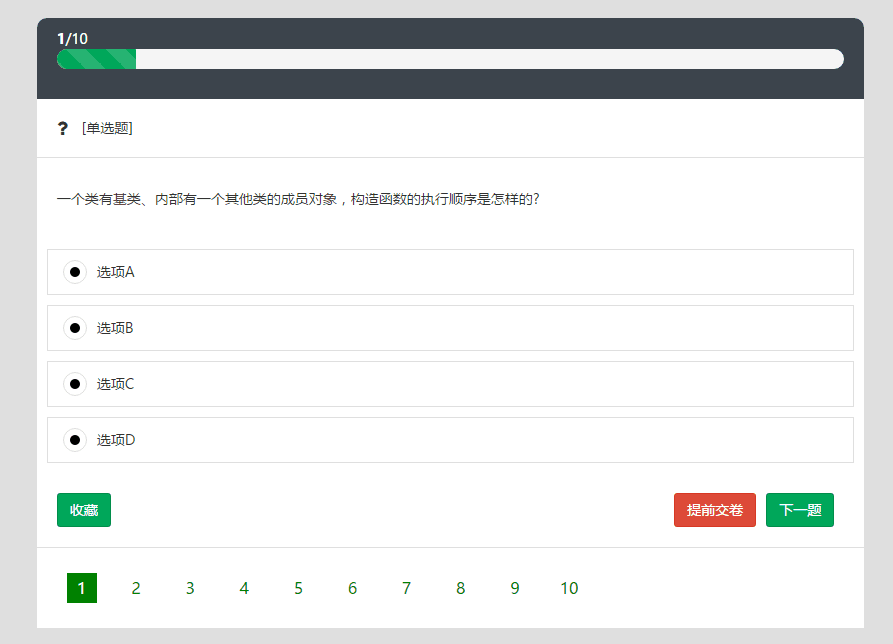


图4.6 在线测试界面

## 4.3 后台管理系统的实现

多角色多媒体在线教育平台是多角色的，相互协作的Web系统，其后台管理人员分为三类：系统管理员、教务老师、任课老师，由于其中功能有较多重复，所以不分开来说，在每个功能中会指出那类管理员会有权限进行操作。

### 4.3.1 登录和权限验证功能

登录功能跟用户端的登录相似，在此不做重复赘述。此处重点做权限验证的功能的说明，权限验证是后台管理系统的每项操作都必须进行的部分，为了简化代码重复，我使用类继承方式，在后台管理的所有Controller继承于AdminBaseController，在父类AdminBaseController中进行权限的获取，在子类Controller中的每项Action中检验是否有相应的权限，如有则继续执行，否则提示用户无权限操作。

权限获取和验证的简易代码如下：

class AdminBaseController extends Controller

{

public static $permission;

public function \_\_construct(){

//判断用户是否登录

$id = isset($\_COOKIE['user\_id']) ? $\_COOKIE['user\_id'] : 0;

if(empty($id) || !is\_numeric($id) || $id < 0)

{

return redirect('admin/login');

}

if(empty(self::$permission)){

$PermissionDao = new PermissionDao();

$permission = $PermissionDao->getByTeacherId($id);

self::$permission = $this->fmtPermission($permission);

}

}

当用户登录后，跳转首页时就会去查询用户的相关权限列表，进行存储，为了提升性能，考虑到用户的权限变化的不连续性，此处只查询一次用户权限表，当用户权限变化时，需要登出系统重新登录才会有变化。

获取用户的权限列表后，只需要在对应controller里获取相应的权限判断用户是否有权限访问。

### 4.3.2 课程管理功能

课程管理模块是教务老师所拥有的权限管理，课程管理模块分为课程方向管理、课程分类管理、上线课程管理、未上线课程管理以及删除课程恢复等模块。

其中课程方向管理和课程分类管理是提供给用户进行课程检索使用，能够让用户分方向和类别来筛选自己感兴趣的模块。其中是一个课程方向可以包含多个课程分类，如前端开发方向包括HTML、CSS、Javascript等多个课程分类，在添加课程时需要确定课程所属的方向和类别。

教务老师可以查看相关的课程列表并进行相关操作，如增删改查等基本功能，课程管理界面如图4.7：

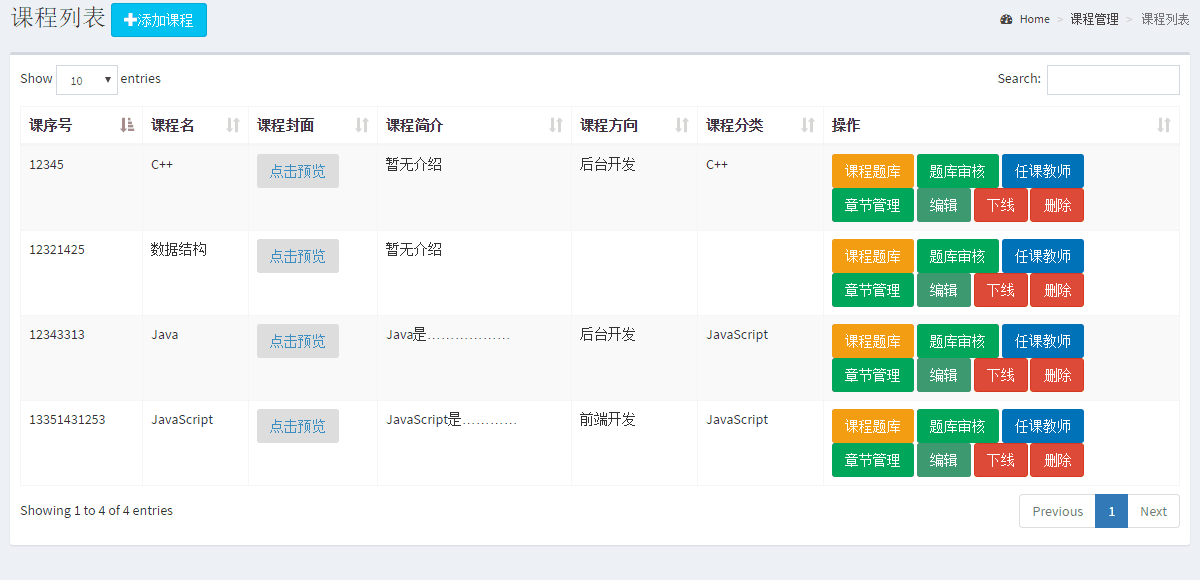


图4.7 课程管理界面

课程列表界面是刷选出当前有效的课程，并进行相应的管理，剔除了一些过期无效的课程。

1. 课程添加

点击上图中的添加课程按钮，添加课程使用Bootstrap中的模拟框Modal插件，会弹出添加课程的弹窗，需要填写课程信息，如课程封面、课序号、课程名、课程简介等信息。如图4.8所示。

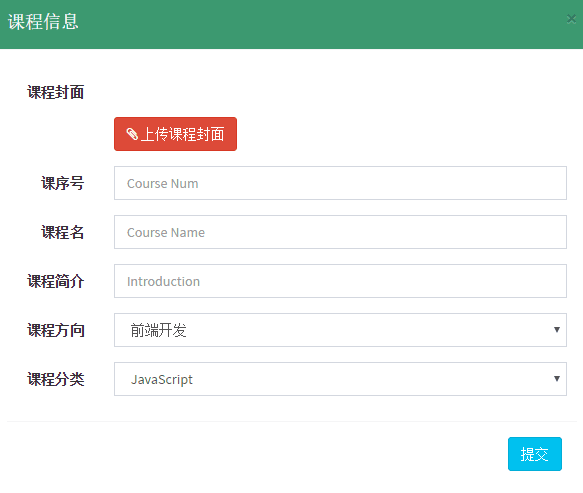


图4.8 添加课程界面

其中点击上传课程封面按钮，JS采用图片的预加载功能，实时显示上传的图片。其中课序号，课程名，课程简介都不能为空。

点击上图提交按钮时，会触发Ajax请求，根据服务器端返回的消息来提示用户的操作结果，有着良好的交互效果。下列是Ajax请求的部分代码：

$("#form").submit(function(e){

e.preventDefault();

e.stopPropagation();

var form = new FormData($("#form")[0]);

$.ajax({

type: "POST",

url: '/admin/savecourse',//提交的URL

data: form,

processData:false,

contentType:false,

success:function(data){

if(data.error == "0"){

alert(data.msg);

location.reload();

}

else{

alert(data.msg);

}

},

error:function(){

alert(' 更改失败,内部服务出错，请联系管理员！');

}

});

});

课程添加完成后需要进一步完善该课程信息，因此课程添加成功为状态为未上线，并且将跳转至课程章节信息，管理员需要填写课程共有多少章，由Js代码实现动态生成需要填写的各章节名称，当用户提交时，将各章名称相应的存入数据库中，添加章节界面如图4.9：

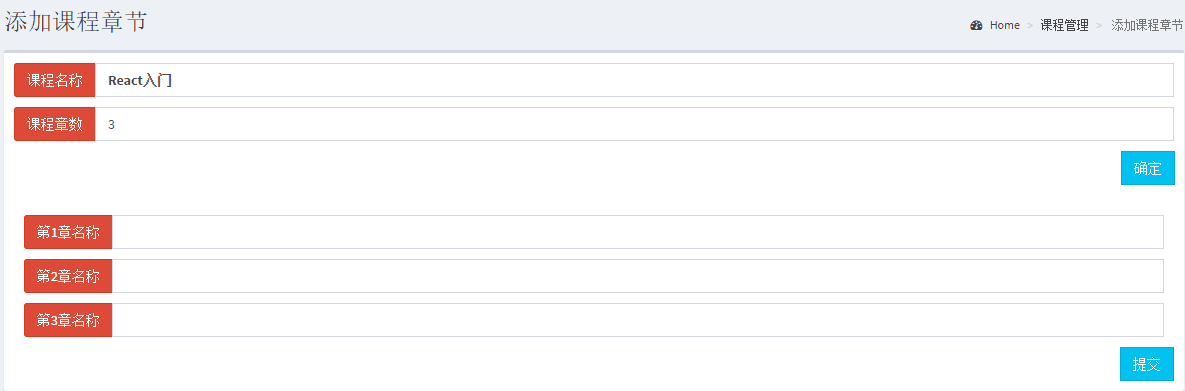


图4.9 编辑课程界面

其中用户填写课程章数后，点击确定，生成下列的章节名称的Js代码如下：

$("#sureChapterCount").click(function(){  
  var chapterCount = $("#chapterCount").val();  
  if(chapterCount == '' || chapterCount <= 0){  
    alert('请填写正确的章数');  
  }  
  else{  
    var html = "";  
    for(i=0;i<chapterCount;i++){  
      html = html +   
          '<div class="input-group margin">'+  
          '<div class="input-group-btn">'+  
         '<button type="button" class="btn btn-danger ">'+  
          '第'+(i+1)+'章名称</button>'+  
          '</div>'+  
          '<input type="text" class="form-control" id=""

 name=chapter'+(i+1)+' required>'+  
          '</div>';  
     }  
     html = html +

'<input type="text" hidden="hidden" name="count"

value='+chapterCount+' />'+  
  ' <div class="input-group-btn">'+  
  '<button type="submit" id="sureChapterCount"

class="btn btn-info btn-flat pull-right">提交</button></div>';

$('#chapterContent').html(html);  
  }  
});

当课程章节完成后，课程的整体设计就大致完成，任课教师可以根据章来对应的添加节，用户进行课程的学习，视频的观看，都是通过章中的节来确定的。总的来说，只有当课程的章节都确定时，并有相关任课老师添加相应的教学视频和题目，用户才能进行正常的学习。

1. 课程编辑

同理，课程编辑也是才有Bootstrap的模拟框Modal插件和Ajax异步提交数据，因此在此不重复赘述该模块，编辑课程界面如图4.10：

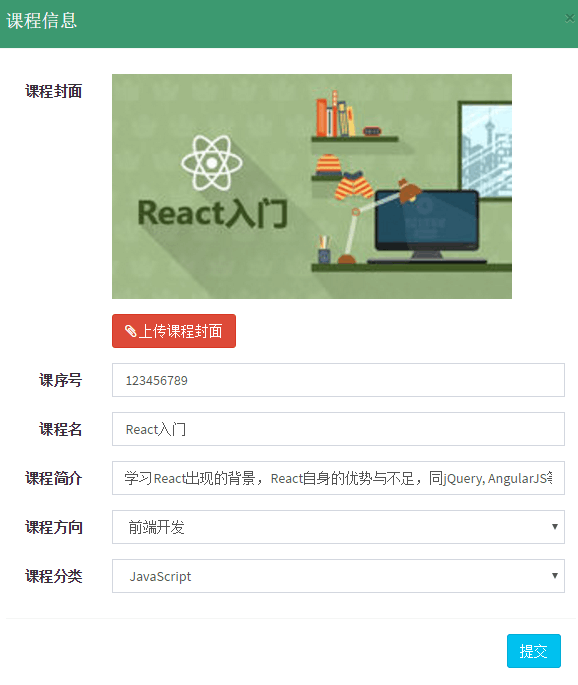


图4.10 编辑课程界面

1. 课程删除

课程删除的真实含义是将课程的状态置为无效，并不会真实的去删除该门课程和相关的数据库等信息。进行该操作会通过Ajax发送请求给后台，后台处理后会提示相关信息，删除失败或者删除成功等信息，提示用户，提升用户体验感。

1. 课程上下线

课程上下线是设置用户是否能看到该课程信息并进行学习的关键设置，用户只能查看到已上线的课程，当课程为准备好或者课程过期时，可以设置课程状态为下线，这样用户即不可见该课程。

1. 课程方向及分类

课程方向和分类是将用户筛选课程的依据，管理员可以管理课程方向及分类，在添加课程时选择相应的方向和分类即可。其中课程分类是属于课程方向的，即课程方向包含多个课程分类，如前端设计，有HTML、CSS、JavaScript等分类。课程方向及分类界面设计如图4.11：

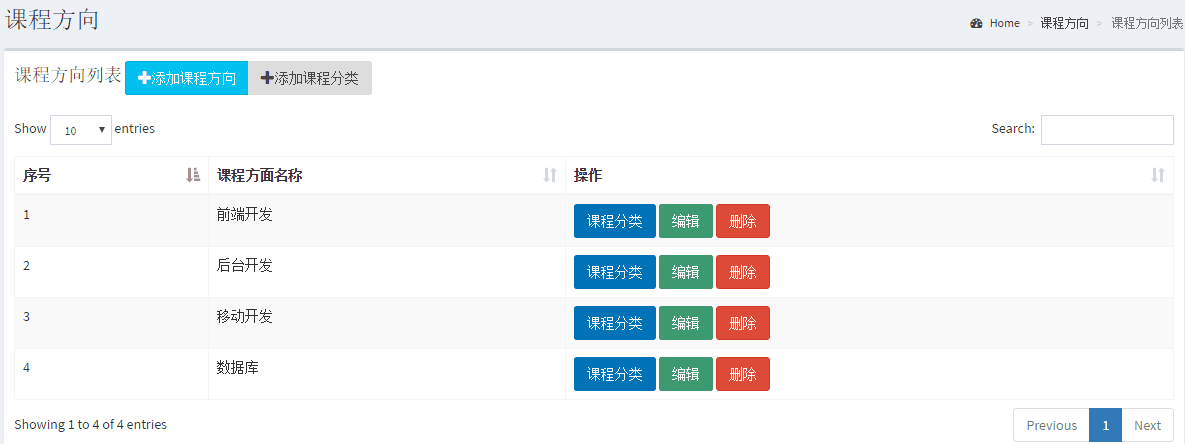


图4.11 课程方向及分类界面

其中添加课程分类采用实时异步方式获取某一课程方向的课程分类，点击课程方向时，下列的课程分类会实时同步，添加课程方向分类界面如图4.12：

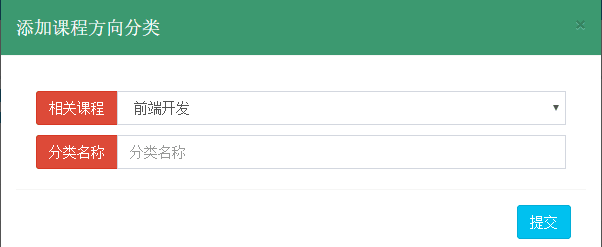


图4.12 添加课程分类界面

### 4.3.3 题库管理功能

（1）已审核题库模块

题库管理模块也是是教务老师所拥有的权限管理，教务老师可以查看相关的课程的题库并进行相关操作，但是只有删和查等操作，题库管理界面如图4.13：



图4.13 题库管理界面

其中包括三大模块，分别为章节目录、章节题目、章节教学视频。其中章节目录是根据题库中已有题目动态生成的章节树形结构，可以让管理员选择相应的章节进行管理。

章节题目列表是该章节的部分题目的展示，包括题目的添加人、添加时间、题目题干、选项、正确答案、解析、题目正确率和删除操作，其中题目正确率是通过记录用户端系统记录学生的测试记录的正确与错误次数计算而来。由于题目所占篇幅较大，因此章节列表下方提供了分页查询的功能，可以查询相应的页数的题目列表。

章节教学视频是该章节所有相关视频的展示，其中包括视频的发布人、发布时间、视频标题、视频简介等信息。管理员可以观看教学视频，删除教学视频等功能。视频题库的界面如图4.14：

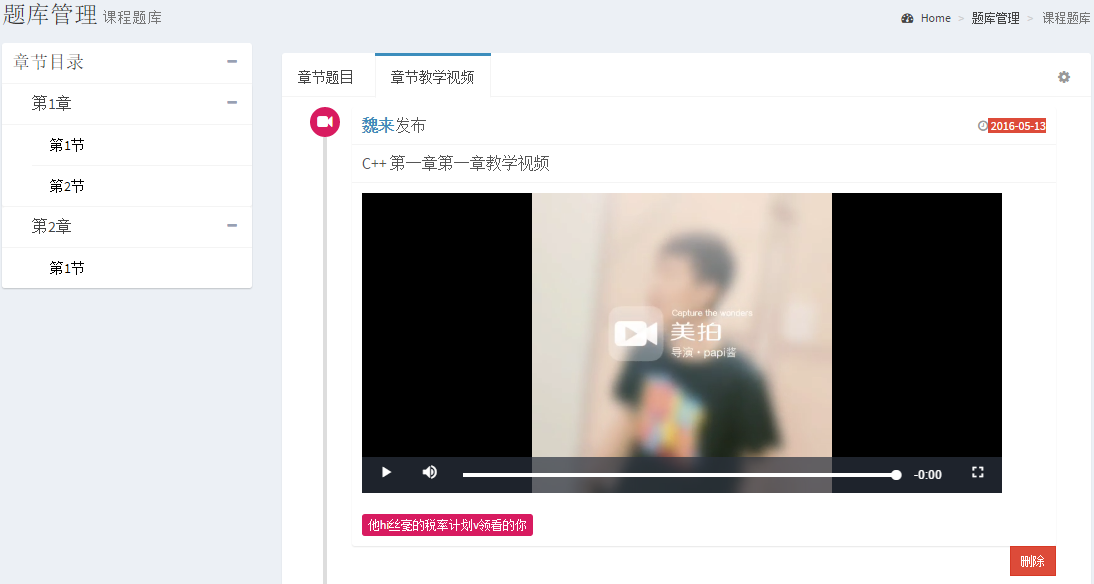


图4.14 课程视频题库管理界面

教学视频的删除也是将该视频置为无效状态，该视频的对外就是不可见的。

（2）审核题库模块

教务老师也可以审核题库，这里的题库是指任课老师提交的相关课程的任务所发布的教学视频和相关的题目，这类题目还未经教务审核，所以不能直接进入题库，对外开发，所以在此加入教师对题目的人工审核状态，需要教务确认该类型题目适合该类课程的当前教学任务，才能正式进入题库系统。题库审核跟题库管理相类似，只是删除操作对应成入库的功能，请参见图4.15：



图4.15 题库审核入库界面

点击入库后会提示用户相应的信息，成功或失败，入库成功后，你可以在意审核题库中找到该题目。

### 4.3.4 任务管理功能

任务管理模块也是是教务老师所拥有的权限管理，教务老师可以查看相关任务并进行相关操作，包括添加、删出和查看内容等操作，任务列表的界面如图4.16：



图4.16 任务管理界面

任务列表中包括任务标题、相关课程、创建时间、截止时间、任务状态、发布人和操作等信息，其中任务状态表示该任务是否完成、未完成、和已过期三种状态。

1. 查看任务内容

点击查看内容即跳转至任务的详情页，其中会显示任务的标题、发布人、创建时间、课程名、接收人信息、任务内容等。任务详情页界面如图4.17：

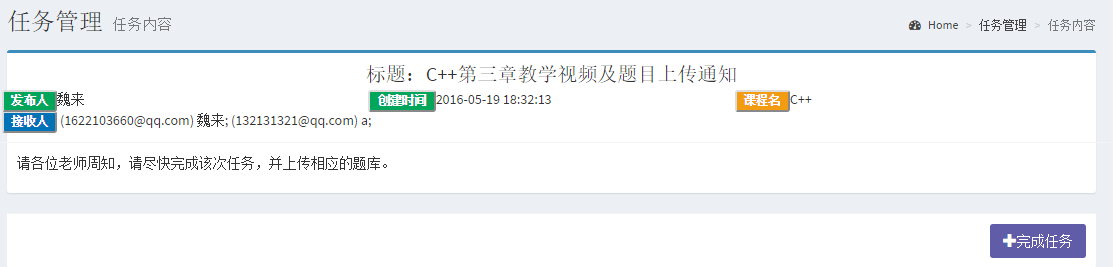


图4.17 任务详情页界面

1. 添加任务

添加任务功能是教务管理员任课老师之间进行工作布置的重要功能，它涉及任课老师的下一步的工作计划和所授课程，任课老师会根据该任务去准备相关的教学视频和相关题库。添加任务界面如图4.18：

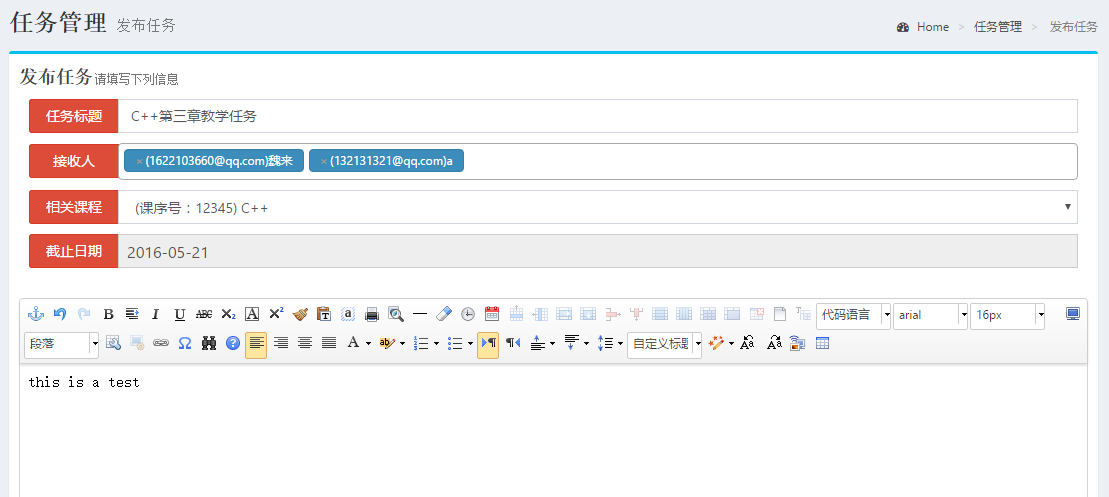


图4.18 发布任务界面

任务标题是该次通知标题，任务接受人是任课老师，设计的是可以选择多个任课老师进行任务的接收，前端是采用开源的JS库Select2实现。相关课程是下拉框组件，可以选择该次任务是哪门课程相关的，截止日前是各个任课老师完成该次任务的最晚日期。最下方的文本框是输入该次任务的详细介绍信息。各项都不能为空，点击发布后会通过Ajax请求发送至后台，返回的信息会提示用户是否操作成功。

点击提交按钮，该任务将会通过Ajax异步提交，在服务端将会将该任务存入数据库，并将通知接收人该条任务，任课教师能够及时阅读和完成任务。

1. 删除任务

删除任务也是将该次任务置为无效状态，各个接收该任务的任课教师会看到该此任务已无效，而不会去完成该任务。

### 4.3.5 消息通知功能

消息通知功能包括两个方面，分别为任务消息通知和通知消息。下面将分别介绍这两个模块。

任务消息是指教务老师发布某个任务时，当前登录用户在该任务的接收人列表中，即该用户需要完成这个任务时，该项任务即会存在任务消息中，任务消息类似于邮件信息一样，分类有四种：未读消息、已读消息、已完成消息和已过期消息。

任务消息页面详解如图4.19：

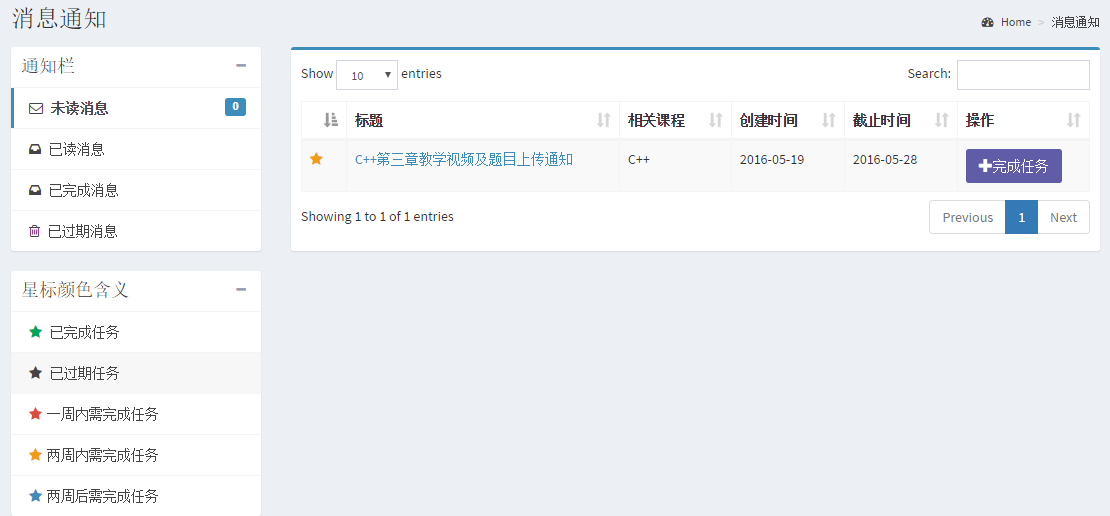


图4.19 任务消息界面

通知栏中为四类分类，下面将说名四类消息的含义：未读消息是指教务已发布该项任务，但是当前用户并没有阅读该项任务的详细内容，未读消息会提示用户及时阅读该项任务并及时完成该项任务。已读消息是指当前用户已读相关的任务列表，然而这些任务并没有完成。已完成消息是当前用户已完成的任务列表，用户可以查看以往的记录。已过期消息是指用户并没有完成任务，但是任务已经超过了截止日期。

上图的的星标颜色也做了很好的提示用户关注各项任务消息的时限，其中绿色星标表示任务已完成，灰色星标白表示任务已过期，红色星标表示任务需在一周内完成，黄色星标表示任务需在两周内完成，蓝色星标表示任务可在两周后完成。

### 4.3.6 完成任务功能

（1）功能简述和界面设计

完成任务功能是任课老师的权限，任课老师接收到教务老师所发布的任务后，可以在消息通知的任务消息中查看相关任务，并在截止日期前完成相应课程的章节教学任务，这一模块就是任课老师的完成任务功能模块。在消息管理模块中，任课老师可以查看该任务消息，并且点击图4.19中的完成任务，则可以跳转至完成任务界面。如图4.24所示。

由图4.20，我们需要相关课程信息、视频名称、上传题目的execl文件、授课视频、该章节的简介信息等，下面将详细说明各个信息的来源和设计。

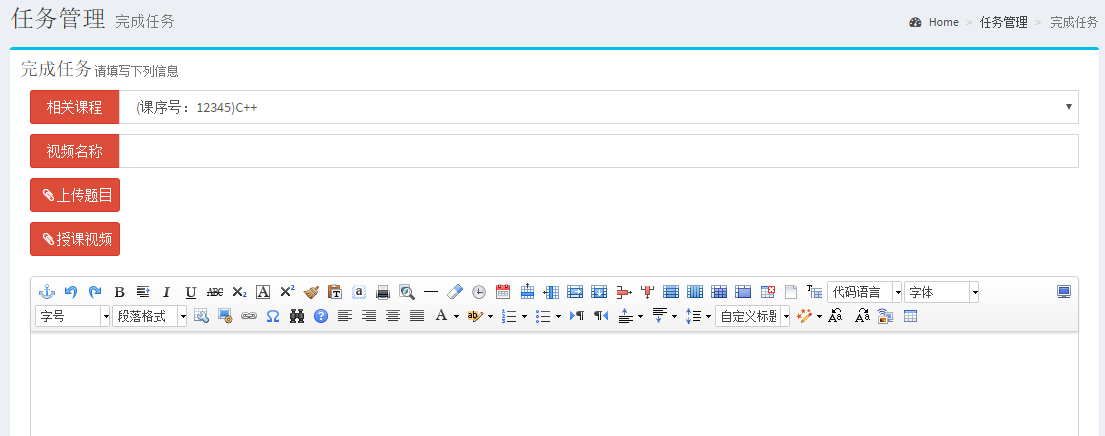


图4.20 完成任务界面

相关课程是根据任务发布的相关课程来定，是教务老师发布任务时所确定的，在这里任课教师是不可修改的。

视频名称是任课老师给当前上传的授课视频的名称，由当前授课老师自己来顶。

上传的题目是规定好格式的excel文件，上传excel格式如图4.21：

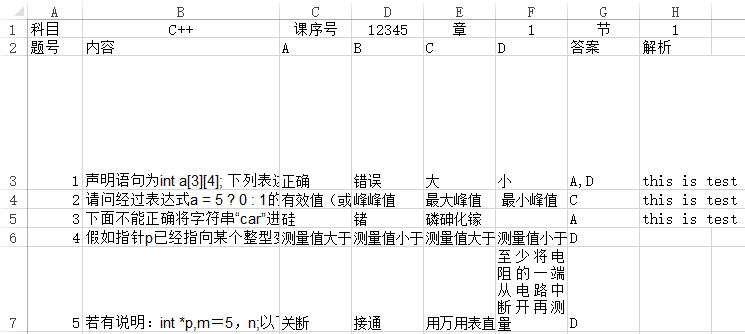


图4.21 题目上传的Excel格式

其中填写课程名称、课序号需要与当前相关课程信息一致，其中章节信息不可为空，它们确定当前题目是属于该课程的第几章第几节。

授课视频的大小不能超过1G，类型需要为MP4，否则无法正确上传和播放。

（2）相关技术和代码

其中上传excel技术是采用Laravel中的Laravel-Excel插件用作excel的文件解析，将excel转换成相应的数组。相应的代码如下：

Excel::load($\_FILES["question"]["tmp\_name"],function($reader) use(&$rows){

$reader = $reader->getSheet(0);//获取excel第一张表

$rows = $reader->toArray();

});

接下来会检验excel中的数据，分别检验课程名，课序号，章节信息等是否正确，当课程所有验证通过时，才可以正确上传该组题目。

### 4.3.7 个人中心功能

个人中心功能是所有后台管理员都具有的功能，其中涉及个人信息的编辑，相关邮件的绑定和密码重置。个人信息的编辑很简单，跟以前的Ajax异步提交一样，修改后台数据库并提示相关的信息。

密码重置跟用户端系统的密码重置相类似，在此不再说明。

绑定邮件是当用户还没有填写邮件信息进行的操作，用户填写完成邮件信息，点击提交后，该请求通过Ajax请求发送至邮件服务，当绑定成功，会提示你已向该邮件发送绑定链接，登录邮箱后，点击该链接即会将该邮件绑定至用户的邮箱信息，否则提示错误信息。下列是绑定邮件服务的相关代码：

try {

Mail::send('admin.sendbindemail', ['data' => $data], function($message) use($data)

{

$message->to($data['email'])->subject('Admin\_DLUT 密码修改');

});

$result['msg'] = '邮箱确认连接已发送至邮件,请登录邮箱进行操作';

$result['error'] = 0;

return $result;

} catch (Exception $e) {

$result['msg'] = '内部服务出错，请联系管理员';

return $result;

}

### 4.3.8 在线学习功能

用户端系统主要提供的功能就是在线学习的功能，当用户选择某门课程点击后进入课程预览页，点击开始学习既可以跳转至课程学习页面，此处可以查看章节的详细信息，点击相关章节及可以观看该章节的教学视频，并给教师留言提问、以及简单的小测试等功能。课程列表界面如图4.22：

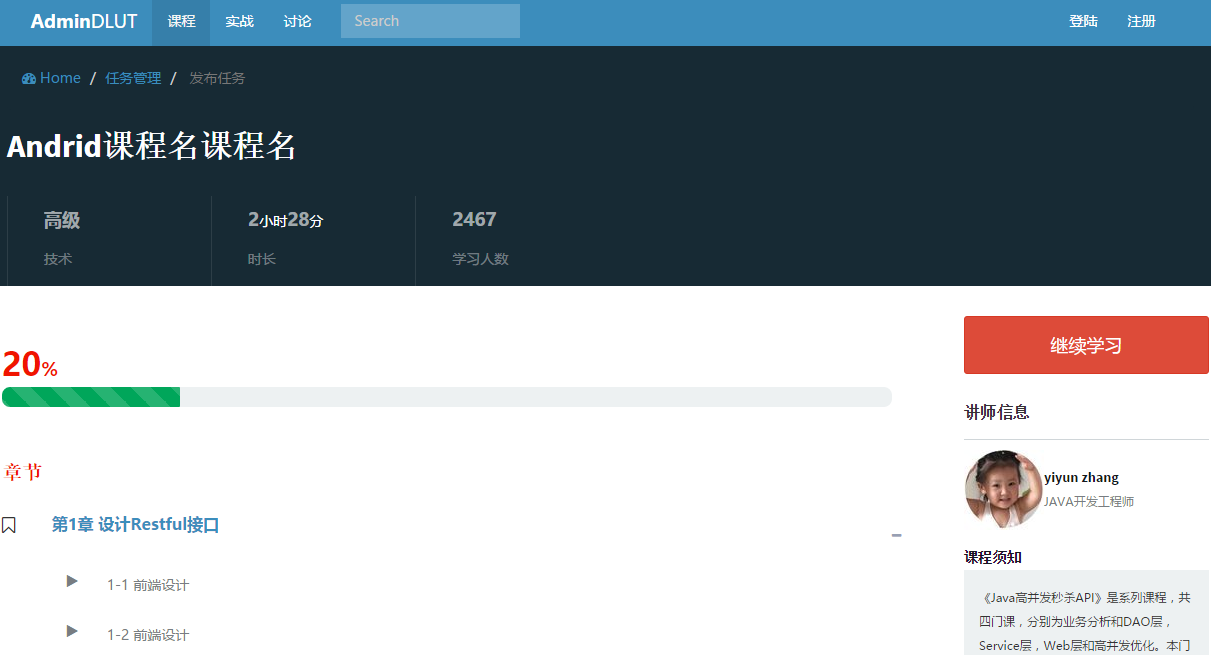


图4.22 课程列表界面

点击章节播放链接，即会跳转至课程的播放页，学生可以在当前界面观看当前章节的教学视频。课程的在线播放页如图4.23：

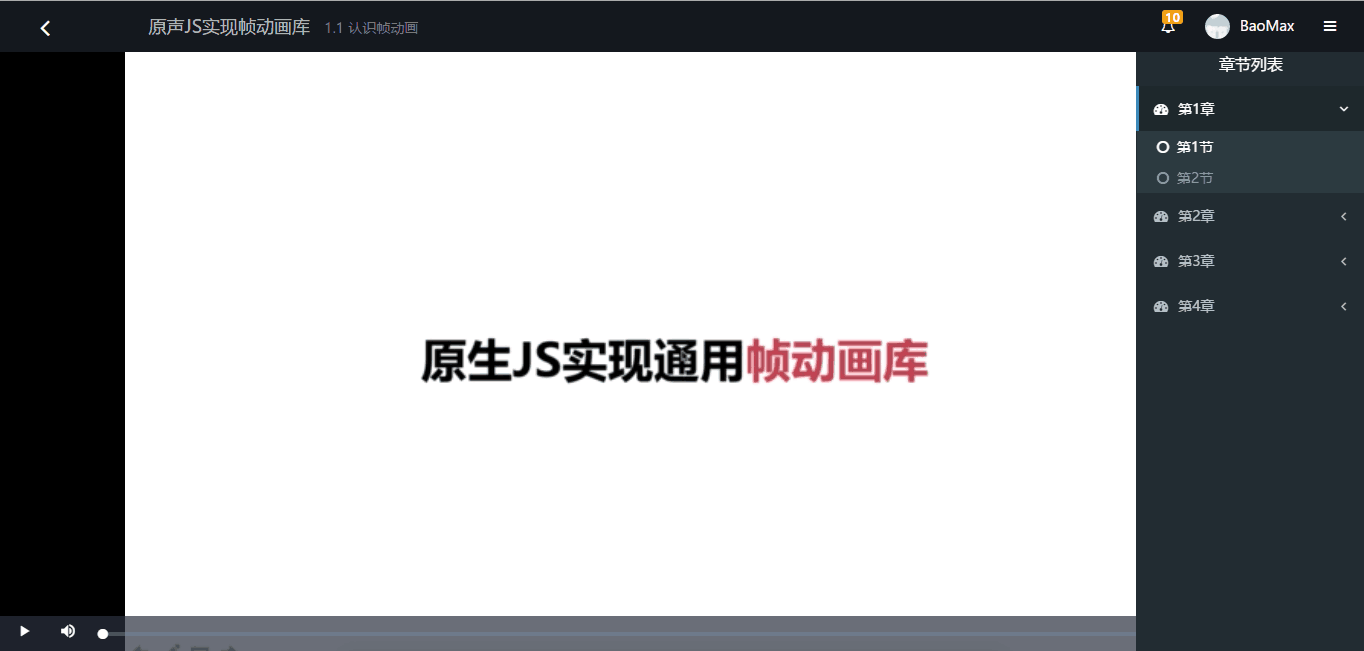


图4.23 课程播放界面

5 系统测试

系统测试[14]是将经过集成测试的软件，作为计算机系统的一个部分，与系统中其他部分结合起来，在实际运行环境下对计算机系统进行的一系列严格有效地测试，以发现软件潜在的问题，保证系统的正常运行。

## 5.1 功能测试

功能测试也叫黑盒测试或数据驱动测试，测试软件系统的功能是否正确，其依据是需求文档，不需要考虑整个软件的内部结构及代码.一般从软件产品的界面、架构出发，按照需求编写出来的测试用例，输入数据在预期结果和实际结果之间进行评测，进而提出更加使产品达到用户使用的要求。

本文功能测试主要才采用等价类划分和边界值分析法进行测试。

等价类划分[15]是把所有可能的输入数据，即程序的输入域划分成若干部分（子集），然后从每一个子集中选取少数具有代表性的数据作为测试用例.该方法是一种重要的，常用的黑盒测试用例设计方法。

边界值分析[16]方法是对等价类划分方法的补充，主要是考虑系统错误大多是发生在输入或输出范围的边界上，而不是发生在输入输出范围的内部。因此针对各种边界情况设计测试用例，可以查出更多的错误。

本次功能测试的环境如表5.1：

表5.1 系统测试的环境

|  |  |
| --- | --- |
| 服务器端操作系统 | CentOS 7.2.1511 |
| 搭建环境 | 阿里云 ESC |
| 数据库客户端 | MySQL 5.6.29 |
| Web服务器代理 | Nginx 1.9.11 |
| 编程语言 | PHP Laravel框架 |
| 版本控制 | Git |
| 编码格式 | UTF-8 |

由于系统的功能测试较多，在此只列出用户端系统模块的功能测试。

用户端系统模块的部分功能测试用例见表5.2：

表5.2 用户端系统模块的功能测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 系统  模块 | 功能点 | 用例  编号 | 前置  条件 | 输入 | 预期结果 | 测试结果 |
| 1：  用  户  端  模  块 | 1.1  登录  密码  找回 | 1.1.1  用户  登录 | 无 | 用户名  密码 | 用户名和密码正确，正常进入系统，不正确提示相关错误信息 | 结果  正确 |
| 1.1.2  密码  找回 | 已绑定  邮箱信息 | 邮件 | 将发送重置链接给绑定邮箱 | 结果  正确 |
| 1.2 课程  浏览 | 1.2.1  检索课程 | 用户已登录 | 检索输入框输入课程名 | 返回正确数据给前端界面展示 | 检索结果正常 |
| 1.2.2  分类筛选 | 进入课程列表 | 点击相关分类按钮 | 返回该分类的课程列表 | 分类正常 |
| 1.2.3  分页 | 进入课程列表 | 点击页码按钮 | 正确执行分页操作 | 分页正常 |
| 1.3  预览课程 | 1.3.1查看课程详情 | 用户已登录 | 点击某门课程 | 进入课程预览界面 | 展示正常 |
| 1.4  开始  学习 | 1.4.1 进入学习课程页 | 进入课程预览界面 | 点击开始学习 | 进入课程学习界面 | 展示正常 |
| 1.4.2 选择课程章节 | 进入学习界面 | 点击相关章节 | 跳转至章节教学视频页 | 播放正常 |
| 1.5  在线测试 | 1.5.1  课程的简单测试 | 进入课程学习页面 | 观看教学视频后可作答相关题目 | 做完提交后显示答案和解析 | 操作正常 |

## 5.2 健壮性测试

健壮性测试。即测试软件系统在异常情况下能否正常运行的能力。健壮性有两层含义：一是容错能力，二是恢复能力。

下面将在不同浏览器中测试各项功能是否正常，来保证系统的兼容性良好。浏览器测试结果见表5.3：

表5.3 浏览器的兼容性测试

|  |  |  |
| --- | --- | --- |
| 浏览器 | 版本 | 测试结果 |
| Google Chrome | 49.0.2623.112 | 正常 |
| Firefox | 46.0.1 | 正常 |
| Safari | 5.1,7 | 正常 |
| Opera | 37.0.2178.43 | 正常 |
| IE | 9+ | 正常 |
| IE | 8- | 界面显示错位，不支持 |

表5.3可以说明该系统具有良好的兼容性，可以满足大部分浏览器正常显示，不支持IE低于9的浏览器访问该系统。

## 5.3 测试总结

通过对系统进行相应的功能测试和健壮性测试，确实发现了系统存在的问题，并及时进行了修正，修正后，系统的功能性更加稳定，出错率明显降低，使系统更加稳定的运行。

在所有测试结束后，该系统基本能保持稳定运行，能确保一定的兼容性，确保功能的稳定运行，满足了用户的需求。

# 结 论

本文是基于B/S架构的多角色、多媒体的教育网站系统的设计与实现。多角色代表的是多用户，多权限的管理，不同的用户有不同的职责和权限，负责系统功能的一部分，限制其没有权限的功能。多媒体代表是音频，视频的在线可视化教育，可以让相应学生随时随地的学习。在系统中需要各个角色的相互配合和督导，使系统正常稳定的运行下去。基于Web的网络学习与测试大大扩展了考试的灵活性，任何人可以在任何时间和任何地方都可以登录系统进行学习，具有极大的优势。

本文阐述了多角色、多媒体在线教育网站系统的背景、设计和实现方案，主要包括下列几个方面：

1. 说明系统开发的项目背景、项目意义以及相关技术的简介。
2. 通过系统的可行性分析明确项目是否可行，通过相关的功能划分，用例图分析，明确相关的需求说明。
3. 项目的总体和概要设计，说明了系统各模块的功能和情况，完成各功能的设计与分析。其中包括系统的架构设计、系统功能模块设计和数据库设计，详细的阐述了系统的总体设计。
4. 项目的系统实现是本系统的实现过程，该部分说明了各模块是如何实现与设计，包含具体界面和逻辑代码。该部分分为两部分：用户端系统功能和后台管理系统，从两个不同功能模块切入系统，进行开发。
5. 系统测试是对系统的验收测试，包括功能性测试和健壮性测试，在此过程中，不断修复BUG，完善系统功能和稳定性。

该系统基本能保持稳定运行，能确保一定的兼容性，确保功能的稳定运行，满足了用户的需求。

参 考 文 献

[1] Tamariz F, Merrell R, Popescu I, et al. Design and implementation of a web-based system for intraoperative consultation[J]. World journal of surgery, 2009, 33(3): 448-454.

[2] Desir N. Design and implementation of genre specific multimedia educational supplements for non-major undergraduate students[D]. MORGAN STATE UNIVERSITY, 2012.

[3] ZHOU X, BAI Z. Design and Implementation of Multimedia Remote Educational System [J]. Modern Electronic Technique, 2004, 18: 006.

[4] 张莉萍, 余璆. 多媒体网络教学系统的设计与实现[J]. 计算机与现代化, 2004 (5): 58-59.

[5] 蒋东兴, 罗念龙, 戚丽, 等. 校园网络教学系统及其关键技术研究[J]. 计算机应用, 2001, 21(9): 23-25.

[6] 胡蕾. 在线教育背景下的新型师生互动探究[J]. 教学月刊: 中学版 (教学管理), 2015 (7): 59-62.

[7] Schwartz,B.,Zaitsev,P.,Tkachenko,V. 高性能MySQL[M]. 北京:电子工业出版社,2013.

[8] Laravel社区. Laravel开发文档. http://www.golaravel.com/laravel/docs/5.1/.

[9] 苗泽. Nginx高性能Web服务器详解[M]. 北京:电子工业出版社,2013.

[10] 赖尔, 苏金国. Head First Ajax (中文版)[M]. 北京: 中国电力出版社,2010,07.

[11] 戈马，彭鑫. 软件建模与设计:UML、用例、模式和软件体系结构[M]. 北京:机械工业出版社,2014.

[12] 汤鹏翔，陈梦东，刘连忠等. 一种实用的基于角色访问控制系统设计与实现[J].计算机应用,2002,22(12):41-43.

[13] 西尔伯沙茨　等著,杨冬青 等译. 数据库系统概念[M]. 北京:机械工业出版社,2012.

[14] 田仲. 系统测试性设计分析与验证[M]. 北京:北京航天航空大学出版社,2003.

[15] 范明红, 浦云明, 汪志华. 等价类测试与划分研究[J]. 计算机技术与发展, 2009, 19(7): 62-65.

[16] 赵瑞莲. 软件测试方法研究[D]. 中国科学院研究生院 (计算技术研究所), 2001.

# 附录A 数据库表设计

表A.1 学生表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | username | 用户名 | varchar(255) | NOT NULL |
| 3 | name | 姓名 | varchar(255) | NOT NULL |
| 4 | email | 邮箱 | varchar(255) | NULL |
| 5 | clazz | 班级 | varchar(255) | NULL |
| 6 | phone | 电话 | varchar(255) | NULL |
| 7 | password | 密码 | varchar(255) | NOT NULL |

表A.2 课程表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | course\_num | 课序号 | varchar(20) | NOT NULL |
| 3 | name | 课程名 | varchar(255) | NOT NULL |
| 4 | introduction | 课程简介 | text | NOT NULL |
| 5 | pic | 课程封面 | varchar(255) | NOT NULL |
| 6 | type | 课程分类 | varchar(255) | NOT NULL |
| 7 | status | 课程状态，结束或正开课 | int(2) | NOT NULL  默认值：1 |

表A.3 课程方向表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | name | 课程方向名 | varchar(255) | NOT NULL |

表A.4 课程方向分类表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | name | 课程方向名 | varchar(255) | NOT NULL |
| 3 | direction\_id | 课程方向id | bigint(20) | NOT NULL |

表A.5 学生选课表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | course\_id | 课程id | bigint(20) | NOT NULL |
| 3 | student\_id | 学生id | bigint(20) | NOT NULL |

表A.6 管理员表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | username | 用户名 | varchar(255) | NOT NULL |
| 3 | name | 姓名 | varchar(255) | NOT NULL |
| 4 | email | 邮箱 | varchar(255) | NULL |
| 5 | master\_id | 上级主管 | bigint(20) | NULL |
| 6 | phone | 电话 | varchar(255) | NULL |
| 7 | password | 密码 | varchar(255) | NOT NULL |

表A.7 角色表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | display\_name | 显示名称 | varchar(255) | NOT NULL |
| 3 | name | 角色名称 | varchar(255) | NOT NULL |
| 4 | description | 角色说明 | varchar(255) | NOT NULL |
| 5 | create\_at | 创建时间 | timestamp | NOT NULL  默认值：当前时间 |
| 6 | update\_at | 更新时间 | timestamp | NULL |

表A.8 权限表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | display\_name | 显示名称 | varchar(255) | NOT NULL |
| 3 | name | 权限名称 | varchar(255) | NOT NULL |
| 4 | create\_at | 创建时间 | timestamp | NOT NULL默认值：当前时间 |
| 5 | update\_at | 更新时间 | timestamp | NULL |

表A.9 用户权限表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | permission\_id | 权限id | bigint(20) | NOT NULL |
| 3 | teacher\_id | 管理员id | bigint(20) | NOT NULL |
| 4 | operation | 操作类型 | enum(‘a’,’d’,’u’,’s’) | NULL |

表A.10 任务表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | course\_id | 课程id | bigint(20) | NOT NULL |
| 3 | publish\_id | 发布教师id | bigint(20) | NOT NULL |
| 4 | title | 任务标题 | varchar(255) | NOT NULL |
| 5 | content | 题干 | text | NOT NULL |
| 6 | create\_at | 创建时间 | timestamp | NOT NULL默认值：当前时间 |
| 7 | update\_at | 更新时间 | timestamp | NULL |
| 8 | status | 状态 | int(2) | -1 是该题作废，0是发布未审核，  1是通过审核 |
| 9 | deadline | 截止日期 | timestamp | NOT NULL |

表A.11 题目错误记录表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | question\_id | 显示名称 | bigint(20) | NOT NULL |
| 3 | correct | 课程id | int(11) | NOT NULL |
| 4 | wrong | 发布人id | int(11) | NOT NULL |

表A.12 密码重置表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | teacher\_id | 管理员id | bigint(20) | NULL |
| 3 | student\_id | 用户id | bigint(20) | NULL |
| 3 | \_token | 验证字符串 | varchar(255) | NOT NULL |
| 4 | create\_at | 创建时间 | timestamp | NOT NULL |

表A.13 题库表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | work\_id | 显示名称 | bigint(20) | NOT NULL |
| 3 | course\_id | 课程id | bigint(20) | NOT NULL |
| 4 | teacher\_id | 发布人id | bigint(20) | NOT NULL |
| 5 | content | 题干 | text | NOT NULL |
| 6 | options | 选项集合 | text | NOT NULL |
| 7 | answer | 答案 | text | NOT NULL |
| 8 | analysis | 解析 | text | NOT NULL |
| 9 | chapter | 章 | int(2) | NOT NULL |
| 10 | section | 节 | int(2) | NOT NULL |
| 11 | course\_name | 所属课程名 | varchar(255) | NULL |
| 12 | course\_num | 课序号 | varchar(255) | NULL |
| 13 | create\_at | 创建时间 | timestamp | NOT NULL默认值：当前时间 |
| 14 | update\_at | 更新时间 | timestamp | NULL |
| 15 | status | 题状态 | int(2) | -1 是该题作废，0是发布未审核，  1是通过审核 |

表A.14 工作表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | course\_id | 课程id | bigint(20) | NOT NULL |
| 3 | teacher\_id | 发布人id | bigint(20) | NOT NULL |
| 4 | task\_id | 任务id | bigint(20) | NOT NULL |
| 5 | create\_at | 创建时间 | timestamp | NOT NULL默认值：当前时间 |
| 6 | update\_at | 更新时间 | timestamp | NULL |
| 7 | status | 状态 | int(2) | -1 是该工作未读，0是已读，  1是完成该工作。 |

表A.15 邮件绑定表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | teacher\_id | 显示名称 | bigint(20) | NOT NULL |
| 3 | \_token | 验证字符串 | varchar(255) | NOT NULL |
| 4 | email | 绑定邮件 | int(11) | NOT NULL |
| 5 | create\_at | 创建时间 | timestamp | NOT NULL |

表A.16 课程章表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | chapter\_index | 章序号 | int(11) | NOT NULL |
| 3 | chapter\_name | 章名称 | varchar(255) | NOT NULL |
| 4 | course\_id | 所属课程id | bigint(20) | NOT NULL |

表A.17 课程节表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 字段名 | 字段描述 | 数据类型 | 说明 |
| 1 | id | 主键 | bigint(20) | 主键，自增 |
| 2 | chapter\_index | 章序号 | int(11) | NOT NULL |
| 3 | section\_name | 节名称 | varchar(255) | NOT NULL |
| 4 | section\_index | 节序号 | int(11) | NOT NULL |
| 5 | course\_id | 所属课程id | bigint(20) | NOT NULL |
| 6 | title | 视频标题 | varchar(255) | NOT NULL |
| 7 | content | 视频内容简介 | varchar(255) | NOT NULL |
| 8 | videopath | 视频路径 | varchar(255) | NOT NULL |
| 9 | course\_id | 所属课程id | bigint(20) | NOT NULL |
| 10 | status | 状态 | int(2) | 0是提交为审核，  1是审核通过。 |

# 致 谢

转眼间大学四年就快步入尾声，这段岁月既漫长又短暂，时而充满坎坷，充满艰辛，让人疲惫，时而充满欢笑，充满温馨，让人陶醉。感谢陪我一起度过这段岁月的老师和同学们，正是有你们的陪伴，你们的支持，我才能克服重重苦难，让我不断坚持，正是有你们的指引，你们的解惑，让我不断成长，不断的收获。

本学位论文是在导师的殷切关怀和耐心指导下完成的，在此，我衷心的感谢导师对我的关心和帮助。导师待人友善、科学严谨、品德高尚、有渊博的学识，这些优良的品质都深深感染和熏陶着我，让我不断学习。从课题的选择明确到项目的完成，再到学位论文的完成，导师都始终在旁给予我细心的指导和关怀，并提出相关的建议，让我不断修正，收获颇丰。在生活上，导师也是我的益友，我们就像朋友般交谈，一起讨论相关的问题和技术。在此，我谨向导师致以衷心的感谢和崇高的敬意。

感谢大连理工大学和软件学院提供的优秀的教学资源和广阔的学习平台，正是有你们的支持，我才能在不断的学习，不断的成长，你们是我行进路上的明灯，指引我前行。

感谢师兄师姐们给予的帮助，师兄师姐不仅在学术和技术给我指引，而且也在生活上给予关怀，从他们身上我学到了很多优秀的品质。

感谢同班同学的关怀和帮助，在生活中，我们互帮互助，一起欢笑，一起流泪；在科技竞赛中，我们一起努力，共同研究，共同收获荣获与喜悦；在学习中，我们共同分享，一起进步。

感谢我的父母和其他亲朋好友对我的关心、支持和理解，没有他们对我的关心、鼓励和支持，我就没有现在的成就。

最后，感谢曾经教育和帮助过我的所有老师。衷心地感谢为评阅本论文而付出宝贵时间和辛勤劳动的专家和教授们！