**杭州电子科技大学信息工程学院**

**课程设计报告**

|  |  |
| --- | --- |
| **课 程** | 企业级JavaEE案例分析及实战开发 |
| **系** | 计算机系 |
| **专 业** | 软件工程 |
| **班 级** | 软件工程三班 |
| **学 号** | 16905710 |
| **学生姓名** | 陈其淇 |
| **任课教师** | 周梦熊 |
| **学 期** | 2017-2018学年第二学期 |

目录

[1 系统需求分析 3](#_Toc516650509)

[1.1 用户需求分析 3](#_Toc516650510)

[1.2 功能需求分析 4](#_Toc516650511)

[1.3 业务层分析 4](#_Toc516650512)

[2 系统设计 5](#_Toc516650513)

[2.1 系统架构及工作原理 5](#_Toc516650514)

[2.1.1 系统架构 5](#_Toc516650515)

[2.1.2 系统工作原理 6](#_Toc516650516)

[2.2 系统总体设计 6](#_Toc516650517)

[2.2.1 系统功能模块设计 6](#_Toc516650518)

[2.2.2 系统流程设计 7](#_Toc516650519)

[2.2.3 学生课堂系统系统建模 8](#_Toc516650520)

[2.2.4 安全性需求 9](#_Toc516650521)

[2.2.5 业务逻辑层设计 9](#_Toc516650522)

[2.3 系统数据模型设计 10](#_Toc516650523)

[2.3.1 数据概念结构设计—E-R图 10](#_Toc516650524)

[2.3.2 数据逻辑结构设计—关系表 10](#_Toc516650525)

[3 系统实现 15](#_Toc516650526)

[3.1 数据库连接的实现 15](#_Toc516650527)

[3.2 登录模块的实现 17](#_Toc516650528)

[3.3 首页的显示 18](#_Toc516650529)

[3.3.1 学生首页的显示 18](#_Toc516650530)

[3.3.2 教师首页的显示 19](#_Toc516650531)

[3.4 课程公告模块 20](#_Toc516650532)

[3.4.1 学生课程公告的显示 20](#_Toc516650533)

[3.4.2 教师课程公告的显示 22](#_Toc516650534)

[3.5 课件下载模块 22](#_Toc516650535)

[3.5.1 学生的课程下载 22](#_Toc516650536)

[3.5.2 教师的课件上传与管理 24](#_Toc516650537)

[3.6 作业上交模块 25](#_Toc516650538)

[3.6.1 学生的作业上交模块 25](#_Toc516650539)

[3.6.2 教师的作业管理模块 27](#_Toc516650540)

[3.7 在线考试模块 28](#_Toc516650541)

[3.7.1 学生的在线考试模块 28](#_Toc516650542)

[3.8 签到模块 29](#_Toc516650543)

[3.8.1 教师的签到模块 29](#_Toc516650544)

[3.9 教师课堂模块 31](#_Toc516650545)

[4 总结 32](#_Toc516650546)

# 1 系统需求分析

## 1.1 用户需求分析

需求分析是系统开发过程中必不可少的一步，一个系统要想具有良好的用户体验，必须要先进行系统需求分析，一个优秀的学生课堂系统，无论是对学生，还是对老师，都要有良好的功能体验，如此才能够提高学生的学习效率，以及老师的工作效率。

(1) 学生课堂系统应该具有强大的信息交互功能

由于教师所指导的班级可能不止有一个，而且每个班级的学生又非常之多，难以进行管理，无论是发布作业，发布平时练习，以及课堂签到都要花费大量的时间进行。这无疑加大了教师的工作量，造成了一些可以避免的时间被浪费掉。

而对于学生来说，学生每学期的课程量非常之多，学生作业可能也会因此难以记忆，如果学生可以有选择地查看自己的课程作业，这样可以很好的解决问题。

(2) 学生课堂系统应该具有丰富的功能以及良好的实用性

学生课堂系统优秀与否的标准应该在于能否最大程度的增加老师和学生的工作效率，因此把大量的事件委托给课堂系统去做，可以很好地达到这个目的，对于老师来说，一个系统如果能够帮自己计算一场考试地平均分，找出考试中错的最多的题目，可以让教师更有针对性的制定教学计划。一个学生如果能够快速知道哪些作业需要更早上交，哪周要先进行考试，就会更有秩序地开展学习计划，进行课堂学习，这些无疑成为学生课堂系统重中之重的问题。

(3)学生课堂系统的资源管理应用

由于学生与教师的交互在一般课堂进行，传统意义上的老师通过课代表收发作业。在本系统，增加了资源管理模块，学生可以把作业直接上传给对应班级，把资源信息存储在以班级为单位的课堂之中，简化了传统意义上的流程。一个教师可能有多个班级，他可以查看自己所有管理班级的学生作业，有利于老师浏览学生的作业。同样，课件下载模块也是课堂为单位，教师可以把课件上传到班级中供学生下载。

(4)学生课堂系统解决学生和教师的直接联系冲突。

一个学生可以加入多个课堂，一个老师可以开展多个课堂。学生与教师并不发生直接联系，而是以课堂，或者班级作为纽带，因此，就算一个学生参加同一个老师的不同课程，也不会发生逻辑冲突。

## 1.2 功能需求分析

理解需求是在问题及其最终解决方案之间架设桥梁的第一步。开发者只有和客户充分理解了需求之后才能开始设计系统，否则，对需求定义的任何改进，设计上都必须大量的返工。根据前面的用户需求分析，整理出了学生课堂系统的功能需求。

(1) 根据所属身份的不同，分为教师、学生，他们的权限如下：

教师：管理课堂班级，发布课堂公告，发布课堂作业，发起课堂签到，发布课堂考试，上传资源文件。

学生：加入课堂班级，浏览所在班级公告，查看及上传作业，进行签到，参加课堂考试，下载资源文件。

(2) 教师可以在次考试模块中给每个学生的每道题打分，对于选择题，判断题可以由系统进行自动评测，提高老师的工作效率，系统也可以分析出一场考试的平均分，成绩的分布情况，选择题，判断题错误较多的题目。

(3) 教师也可以在签到模块，查看没有完成签到的人，进而判断一次课堂的缺席情况，把这些工作交给课堂系统去完成，可以大幅节约时间，增加课堂的可支配时间，使得课堂授业更加从容。

## 1.3 业务层分析



图1-1业务层分析

通过对本系统的需求分析，系统的基本功能已经大体确定，需要把课堂作为载体，使得学生与教师产生联系，这样复杂的逻辑开始被简化。学生课堂作为一个独立的逻辑单元而存在，中间的业务逻辑围绕着其进行展开。

# 2 系统设计

工欲善其事，必先利其器。在进行软件开发的时候，如果能够使用一些技术非常成熟的框架，那么对于项目的快速开发，安全性的提升，系统架构等，都有非常大的帮助。只有运用一些先进的技术，引入一些好用的框架，规定一些开发的准则，那么，才能保证开发工作顺利进行及保证系统的稳定性灵活性，而且可以进行拓展。在设计之前有需求开发工作，在设计之后还有编码，测试和维护工作，无论哪个环节出了差错，都会让软件无法正常运行。因此，本系统在设计系统时具体来讲应该遵循以下原则：

(1) 先进性原则。

(2) 资源共享性原则。

(3) 安全可靠性原则。

(4) 操作、维护的简便性原则。

(5) 可扩充性、兼容性原则。

## 2.1 系统架构及工作原理

### 2.1.1 系统架构

本系统采用SpringBoot作为系统架构的后台框架，采用了分层架构的理念，

把后台代码结构分为控制器层，服务层，数据库层。系统前端则采用纯html页面，与后台的代码交互完全通过ajax技术，这样的好处在于使得前端与后台的代码实现完全分离。

前端　页面显示：前端的数据由ajax向后台的控制器层发出请求获取，利用vue.js这个前端框架在页面上进行显示。这样非常有利于前端编写。前端编写不需要考虑后台的种种复杂逻辑，对于表格显示，也不需要用复杂的jquery语句操作页面元素，js代码不会过于臃肿。

第一层　控制器层：控制器的主要工作，即接受前端的数据，向服务层传递数据，调用服务层的服务，返回需要的数据，送到前端。

第二层 服务层：接受控制层的数据，进行复杂的逻辑判断，调用数据库层，访问数据库数据，把数据返回控制层，或者其他服务层。

第三层 数据库层，数据库层采用mybaits数据库访问框架，数据库的sql语句写在xml文件中， 或者接口注解的方式进行。这样充分利用java面向对象的思想，让服务层完全采用面向对象的思想进行操作。

### 2.1.2 系统工作原理

在SpringBoot框架下，手机客户端运行App程序，App程序以ajax请求向Web服务器提出访问数据库的请求，Web服务器接受客户端请求后，将这个请求转化为SQL语法，并提交给数据库服务器，数据库服务器得到请求后，验证其合法性，并进行数据处理，然后将处理后的结果返回给Web服务器，Web服务器再一次将得到的结果进行转化，变成json数据形式，转发给手机客户端以进行数据的显示，如图2-1所示。



图2-1 系统架构图

## 2.2 系统总体设计

### 2.2.1 系统功能模块设计

通过对系统的需求分析和可行性分析，本系统设计了以下几个模块，如图2-2所示。



图2-2 系统模块图

### 2.2.2 系统流程设计

本系统的主要流程：首先用户打开前端网站，用户需要注册登录以后才能进入首页。当用户进行登录以后，系统就会对用户进行判断，根据用户身份的不同，会跳转到不同的网页，进行操作。用户登录以后，服务器会返回一个token给前端。用户的每个后台访问请求，都要携带这个token作为数据，一起传递到后台，并且重新对token进行检查，如果token过期，那么就会退回到登录界面，否则就进行逻辑判断和正常的流程操作，进一步响应用户的请求。如图2-3所示。



图2-3 系统流程图

### 2.2.3 学生课堂系统系统建模

以下有两个用例，一个为学生用户，另一个为教师用户。从图2-4中，我们很容易地看出，学生用户与教师用户没有产生直接联系，教师创建班级以后，学生可以加入该班级。而教师在这个课堂中进行各项操作，比如发布考试，发布签到，上传课件，发布作业等，都可以由在这个教师中的学生进行响应完成，对于加入课堂的所有学生来说，他们都能够获取老师的这些操作的信息，并且可以响应由教师发布的课堂请求。教师与学生以课堂作为纽带进行紧密联系，逻辑因此变得不再复杂，如图2-4所示。



图2-4 系统建模图

部署图，如图2-5所示。



图2-5 部署图

### 2.2.4 安全性需求

1. Token是用户身份唯一判断标识，用户的一切操作需要建立在token合法的基础之上，所以后台会对前端的token进行数据校验，确保系统的安全运行。

### 2.2.5 业务逻辑层设计

“学生课堂系统”主要包括了公告发布模块，作业上交模块，课件下载模块，学生签到模块，在线考试模块等。这些模块从用户的角度出发，满足了用户了基本需求，使得学生和老师的产生了互动。因此，可以将系统按以下模块进行划分。

(1) 公告模块设计

公告查看：学生可以在班级课堂的班级公告里查看该班级的所有通知。

公告操作：某个班级的老师，具有发布公告的权限，并且可以显示或隐藏历史公告，还可以删除公告。一般公告发布后，在首页也会显示，这样学生用户不需要进入具体的某个课堂，就可以查看所有新发布的公告，这样增加了学生的用户体验。

(2) 作业上传模块：教师发布作业以后，学生可以上交作业。对于教师来说，每份作业都存着所有学生的上交作业情况，并且可以随时下载这些作业。当然，教师也可以在学生列表页面，查看某个学生的所有上交作业情况，这样更加具有针对性，也提高了教师的工作效率。

(3) 课件下载模块

学生可以在课件下载模块，下载由老师发布与教学内容有关的课件资源，那可以是一张图片，也可以txt文件，也可以是pdf文本，或者是压缩包类型。相对的，教师可以在这个模块上传这些类型的文件。为了增强用户体验，该模块采用了层级目录结构，教师可以创建文件夹，分多级文件，这样分门别类的管理更加清晰，也更加像操作本地文件。

(4) 学生签到模块

学生签到模块中，并不是学生可以随时随地每天进行签到的，只有当老师发起签到的时候，学生在规定时间内，才能进行这种签到，这样签到的好处在于，可以节约教师地上课，系统会自动生成课程上没有进行签到地学生信息，方便教师的管理，而其中预防代签与远距离签到，是一大难点。

(5) 在线考试模块

在在线考试模块中，教师可以创建自定试卷，这些试卷可以作为在线考试的试卷让学生进行，教师通过这些考试，可以获取学生掌握课程的情况，然后展开针对性的教学活动。在教学考试模块中，学生在规定时间内可进行答题活动，但是超出时间的答题是被制止的。教师可以查看这场考试的所有细节，包括每个学生的具体答题情况。

## 2.3 系统数据模型设计

### 2.3.1 数据概念结构设计—E-R图

E-R图(实体-关系图)提供了表示实体、属性和联系的方法。实体用矩形表示，矩形框内写明实体名。 属性用椭圆型表示，并用无向边与其相应实体连接起来。联系用菱形表示，菱形框内写明联系名，并用无向边与其有关实体连接。本系统采用E-R图的方法进行数据库概念结构设计，系统部分E-R图，如图2-5所示。



图2-5 部分E-R图

### 2.3.2 数据逻辑结构设计—关系表

在系统的数据库设计中，首先要创建系统数据库，然后在数据库中创建需要的表和字段。在这个数据库管理系统中要建立5张数据表。这1张数据表的字段说明。

表2-1 用户表(user\_info)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| user\_id | int (11) | 否 | 账号 | 主键 |
| account | varchar(255) | 否 | 任务名 |  |
| password | varchar (255) | 否 | 任务名 |  |
| username | varchar (255) | 否 | 用户昵称 |  |
| school | varchar (255) | 否 | 所在学校 |  |
| professional | varchar (255) | 否 | 所属专业 |  |
| classes | varchar (255) | 否 | 感兴趣的课程 |  |
| achievement\_value | int (11) | 否 | 成就值 |  |
| token | varchar (255) | 否 | 用户身份验证码 |  |
| token\_time | varchar (255) | 否 | 到期时间 |  |
| head\_url | varchar (255) | 否 | 头像相对路径 |  |

表2-2 班级表(class\_room)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| class\_number | int | 否 | 课堂编号 | 主键 |
| class\_name | varchar | 否 | 课堂名称 |  |
| teacher\_id | varchar | 否 | 任课老师id |  |
| begin\_time | varchar | 否 | 开班时间 |  |
| end\_time | varchar | 是 | 结束时间 |  |
| request\_code | varchar | 否 | 班级邀请码 |  |
| class\_head\_url | varchar | 否 | 课程图像 |  |
| class\_intro | varchar | 否 | 课程介绍 |  |
| class\_easy | int | 否 | 课程难易程度 |  |
| class\_time | int | 否 | 课时数 |  |
| class\_goal | int | 否 | 课堂学分 |  |
| remarks | varchar | 是 | 老师给课程增加的备注 |  |

表2-3 通知表(class\_notice)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 自增id | 主键 |
| notice\_title | varchar | 否 | 通知标题 |  |
| notice\_content | varchar | 否 | 通知内容 |  |
| class\_number | int | 否 | 班级号 |  |
| time | varchar | 否 | 通知的时间 |  |
| is\_show | int | 否 | 是否显示通知 |  |

表2-4 课件表(class\_resource)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 资源id | 主键 |
| class\_number | int | 否 | 资源所属于的班级号 |  |
| url | varchar | 是 | 资源在服务器的url，如果是文件夹则为- |  |
| file\_name | varchar | 是 | 资源名 |  |
| father\_id | int | 是 | 它所属于的文件夹 |  |
| sort\_id | int | 是 | 排序id |  |
| is\_show | int | 否 | 是否显示 |  |
| types | varchar | 是 | 文件类型 dir txt png jpg zip |  |

表2-5 考试表(class\_test)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 考试id | 主键 |
| class\_number | int | 否 | 考试所在班级 |  |
| test\_name | varchar | 否 | 考试名 |  |
| begin\_time | varchar | 否 | 开始时间 |  |
| end\_time | varchar | 是 | 结束时间 |  |

表2-6 数据字典表(dictionary\_data)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 数据字典id | 主键 |
| name | varchar | 否 | 数据字典名 |  |
| pid | int | 否 | 父id |  |
| is\_show | int | 否 | 是否显示 |  |
| sort\_id | int | 否 | 排序值 |  |

表2-7 作业-学生对应表表(homework\_student)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 自增id | 主键 |
| homework\_id | int | 否 | 作业id |  |
| student\_id | int | 否 | 学生id |  |
| file\_url | varchar | 否 | 文件路径 |  |
| last\_upload\_time | varchar | 是 | 最后上传时间 |  |

表2-8 班级-学生对应表(student\_class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 自增id | 主键 |
| student\_id | int | 否 | 学生id |  |
| class\_number | int | 否 | 班级id |  |
| begin\_time | varchar | 是 | 学生加入班级时间 |  |

表2-9 手机验证码表(student\_class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 自增id | 主键 |
| telephone | varchar | 否 | 手机号 |  |
| code | varchar | 否 | 验证码 |  |
| time | varchar | 是 | 验证码到期时间 |  |

表2-10 签到表(sign)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 签到id | 主键 |
| class\_number | int | 否 | 对应班级 |  |
| longitude | varchar | 否 | 签到地经度 |  |
| latitude | varchar | 否 | 签到地纬度 |  |
| begin\_time | varchar | 否 | 签到开始时间 |  |
| end\_time | varchar | 否 | 签到结束时间 |  |

表2-11 问答专区表(sign)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 问题或回答id | 主键 |
| question\_title | varchar | 否 | 标题 |  |
| question\_desc | varchar | 否 | 描述 |  |
| user\_id | int | 否 | 发布人id |  |
| publish\_time | varchar | 否 | 发布时间 |  |
| last\_response\_time | varchar | 否 | 最后响应时间 |  |
| hot | int | 否 | 热度 |  |
| good | int | 否 | 点赞数 |  |
| is\_show | int | 否 | 是否显示 |  |
| tag | varchar | 否 | 标签 |  |
| father\_id | int | 否 | 父id |  |

表2-12 考试题目表(test\_question)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 题目id | 主键 |
| test\_id | int | 否 | 题目的试卷id |  |
| value | int | 否 | 分值 |  |
| sort\_id | int | 否 | 排序值 |  |
| types | int | 否 | 题目类型 |  |
| desc | varchar | 否 | 题目描述 |  |
| content | varchar | 否 | 题目内容 |  |
| answers | varchar | 否 | 正确答案 |  |

表2-13 学生-签到对应表(student\_sign)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 无意义自增 | 主键 |
| sign\_id | int | 否 | 签到id |  |
| student\_id | int | 否 | 学生id |  |
| ip | varchar | 否 | 学生签到ip |  |

表2-14 学生-答题表(test\_question\_student\_answer)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 可否为空 | 说明 | 备注 |
| id | int | 否 | 无意义自增 | 主键 |
| test\_question\_id | int | 否 | 考试id |  |
| student\_id | int | 否 | 学生id |  |
| answers | varchar | 否 | 学生答案 |  |

# 3 系统实现

## 3.1 数据库连接的实现

该系统使用的数据库管理系统是MySQL，Web容器是SpingBoot，使用的MySQL版本是MySQL 5.1。jdk版本是1.7。将两个软件分别进行安装、配置及测试。

(1) 配置POM文件

配置mybaits进行数据库连接，添加与mybaits有关的依赖，使得后台数据库采用mybaits框架,并且添加mysql的有关依赖，如表3-1。

|  |
| --- |
| <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>1.3.1</version>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  </dependency> |

表3-1 mybaits框架的依赖

并且增加SpringBoot依赖，使用SpringBoot框架

|  |
| --- |
| <dependency>  <groupId> org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>1.3.1</version>  </dependency>      <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency> |

表3-2 SpringBoot框架的依赖

以下是application.properties配置文件

|  |
| --- |
| server.port=8888  spring.mvc.view.suffix=.jsp  spring.mvc.view.prefix=/WEB-INF/views/  FILE\_PATH\_PATTERN=/upload  user\_head\_img\_url=C:/Users/lenovo/Desktop/headimg/  filePath = /school\_help/img |

表3-3 application.properties

以下是application.yml配置文件

|  |
| --- |
| **spring**:  profiles:  active: dev  #公共配置与profiles选择无关  **mybatis**:  typeAliasesPackage: cn.edu.hziee.model  mapperLocations: classpath:mapper/\*.xml  ---  #开发配置  **spring**:  profiles: dev  datasource:  url: jdbc:mysql://localhost:3306/class\_manage\_system  username: root  password: root  driver-class-name: com.mysql.jdbc.Driver |

表3-4 application.yml

至此，后台的运行环境已经搭建完成，可以部署使用。

在application.yml配置文件中，含有数据库的连接信息，用户名，密码，连接号，驱动信息。Mybaits框架部署已经完成。

## 3.2 登录模块的实现

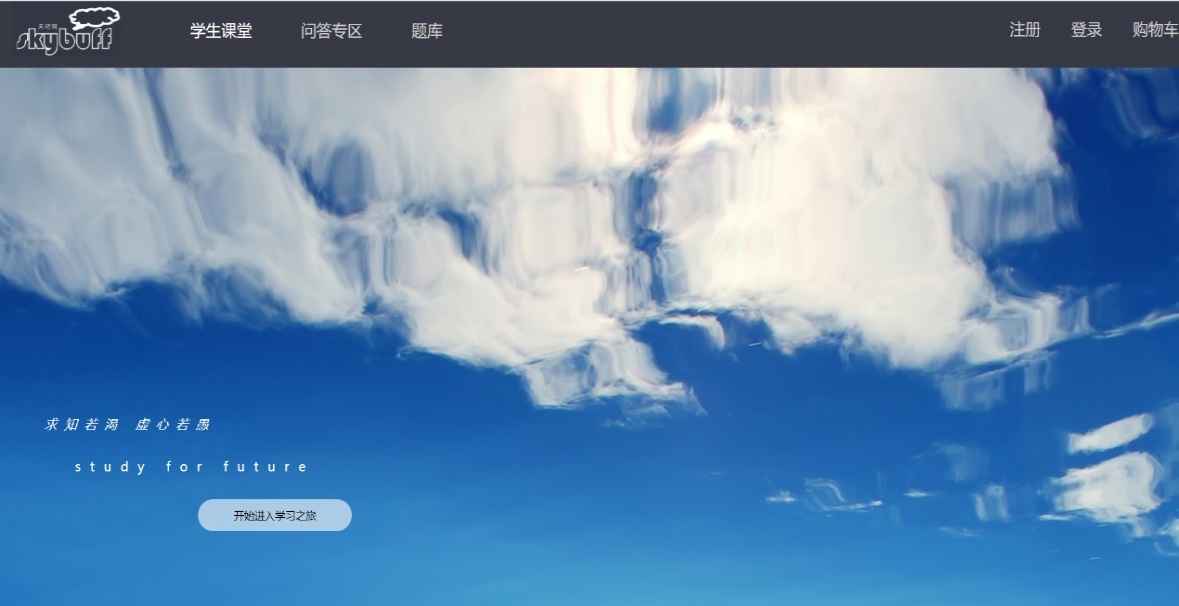


图3-5 登录界面

用户要进入该系统，就需要有一个自己的账号进行登录，用户可以通过立即注册的页面，输入手机号发送验证码到手机，输入密码与刚刚获取的验证码来完成注册操作，用户注册以后，直接可以登录主页面，当然，用户token到期后，要输入手机号和密码，也可进入主页面。登录界面，如图3-5显示。

点击“开始进入学习之旅”，就会弹出登录框，如图3-6所示。



图3-6 登录弹出框

点击按钮以后，左下角的字会渐变得隐藏，而中央得登录框会渐变得显示。如表3-7

|  |
| --- |
| $(".studyButton").click(function(){  $(".show").fadeIn("slow");  $(".bodyer").fadeOut("slow");  }); |

表 3-7 登录框的渐变显示效果

如果没有输入账号或密码，那么就会阻止ajax提交操作，而是先聚焦到没有填写的输入框，要求用户进入账号或密码的输入。如表3-8

|  |
| --- |
| if($("#account").val()==""){  $("#account").focus();  return;  }  if($("#password").val()==""){  $("#password").focus();  return;  } |

文件3-8 聚焦不提交的代码

用户登录成功了以后，会根据用户身份的不同，定位到学生或老师的首页。此外用户登录以后，后台会随机生成一个新的token返回给前端，并且把该token存储到数据库中，并且记录token的失效时间，以后用户的每一次操作，都要携带这个token到后台，进行检验。

## 3.3 首页的显示

### 3.3.1 学生首页的显示

学生的首页的显示分为了五个模块，包括个人用户信息,班级公告信息，已选课程信息，问答专区的热门信息，以及还有课程推荐的信息。

这五块内容分5次分别用ajax请求不同的控制器，然后获取数据再显示在前端。列表类型的数据显示，都采用了vue框架的for循环输出。这样使得前端代码的编程不会拥有过于臃肿的js操作，也非常便于维护。

在学生的首页中，学生还可以完善自己的学校信息，专业信息等。

学生模块的首页内容足够丰富，可以很好地提高用户体验，用户在已选课程中可以快速进入到自己想要前往地页面，并且可以在课程选择中选择新地课程，首页基本上可以进入到本系统地任何角落。如图3-9所示。

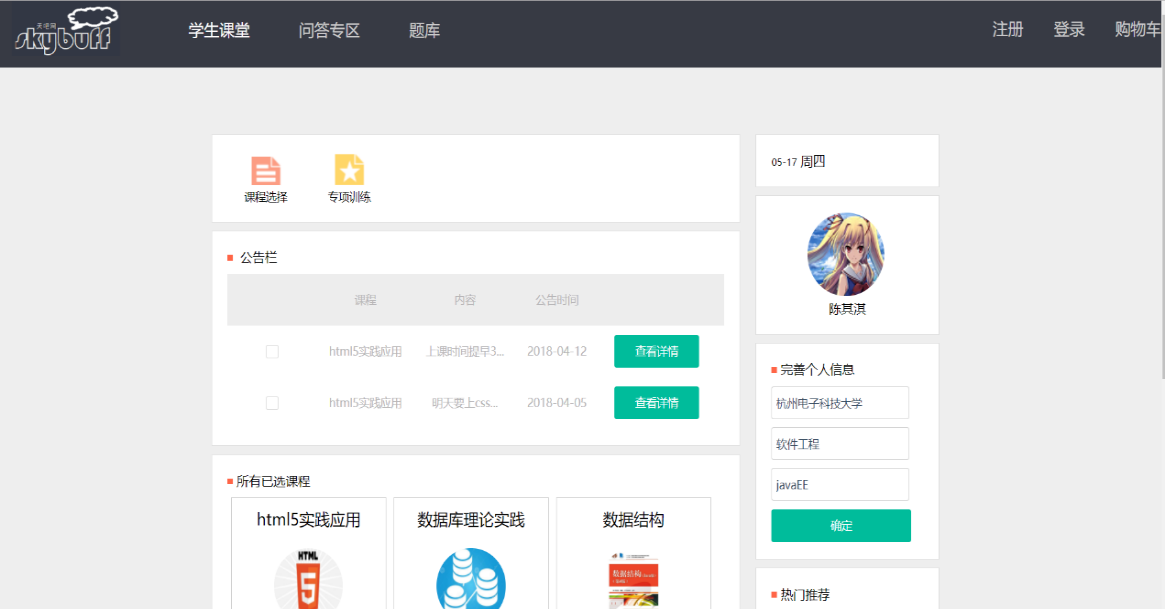


图 3-9 学生的首页

### 3.3.2 教师首页的显示

教师的首页的显示主要有三块内容，分别是班级列表，个人信息，以及备忘录。在这个首页，教师可以创建新的班级，或者进入某个具体的班级查看该班级具体的信息，或者增加一条备忘录。首页的信息量丰富而实用。点击头像，教师可以更改自己的头像。教师的首页如图3-10。

选中一门课程，教师可以查看某个班级内部的具体信息。而班级是有班级邀请码的，班级邀请码显示在首页，方便教师查看。教师可以把邀请码公布给学生，学生找到班级以后，可以凭借邀请码进入班级。

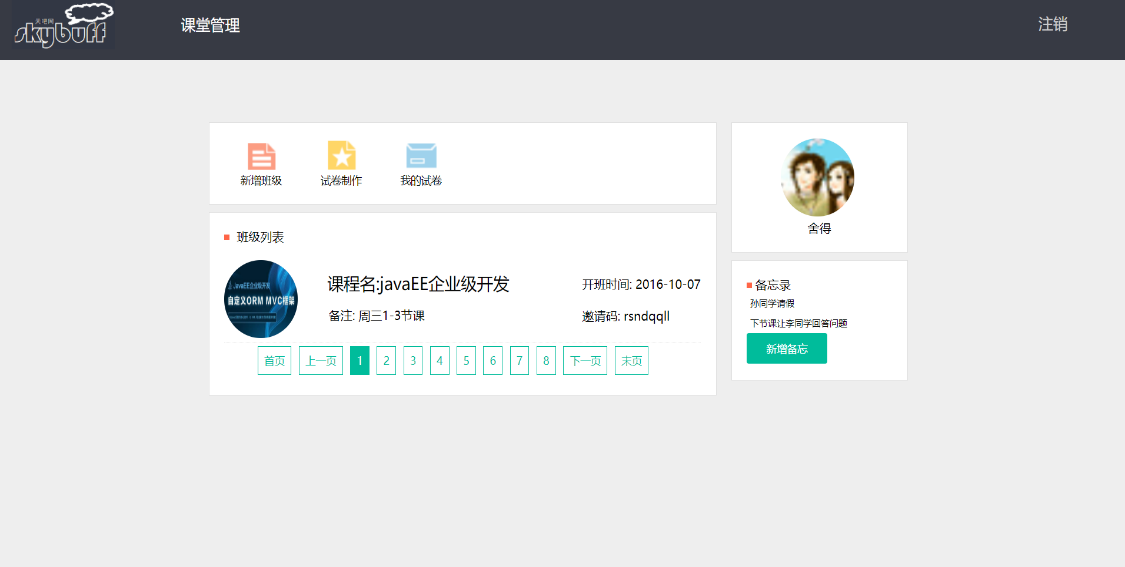


图3-10 教师的首页

## 3.4 课程公告模块

### 3.4.1 学生课程公告的显示

学生在首页的已选课程中，选择其中一门课程可以进入课程公告模块，与首页的课程公告显示有所不同的是首页的课程公告显示的是所有课程最新的课程公告，而已选课程的公告可以显示专门一门课的所有公告，这样可以方便查看某门课程的历史公告，即使首页的公告不显示了，也可以在课程公告模块查看。如图3-11。

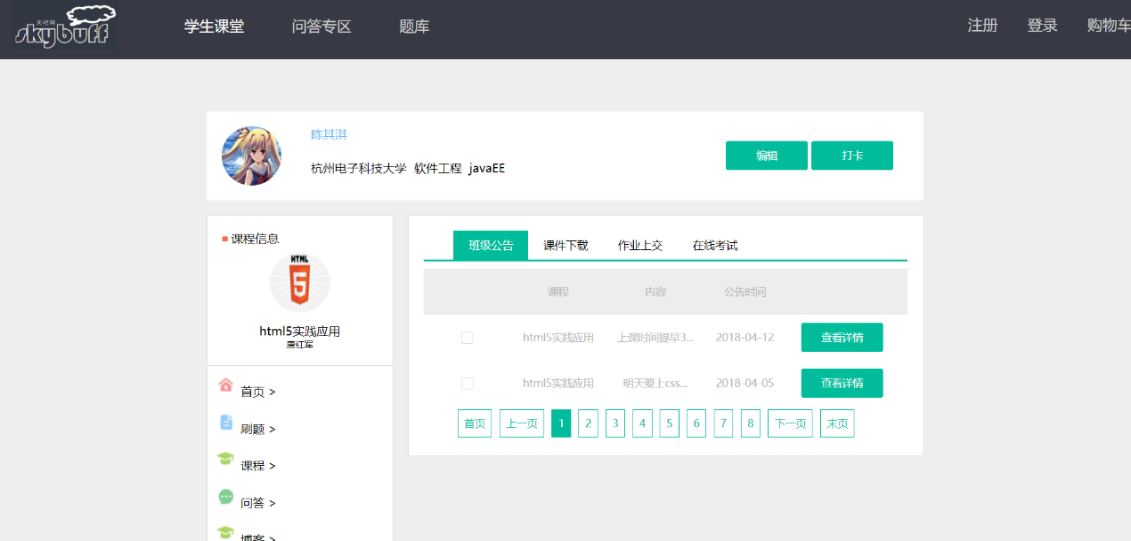


图3-11 课程公告模块

在课程公告模块，只要点击查看详情，就可以查看具体的公告。如图3-12。

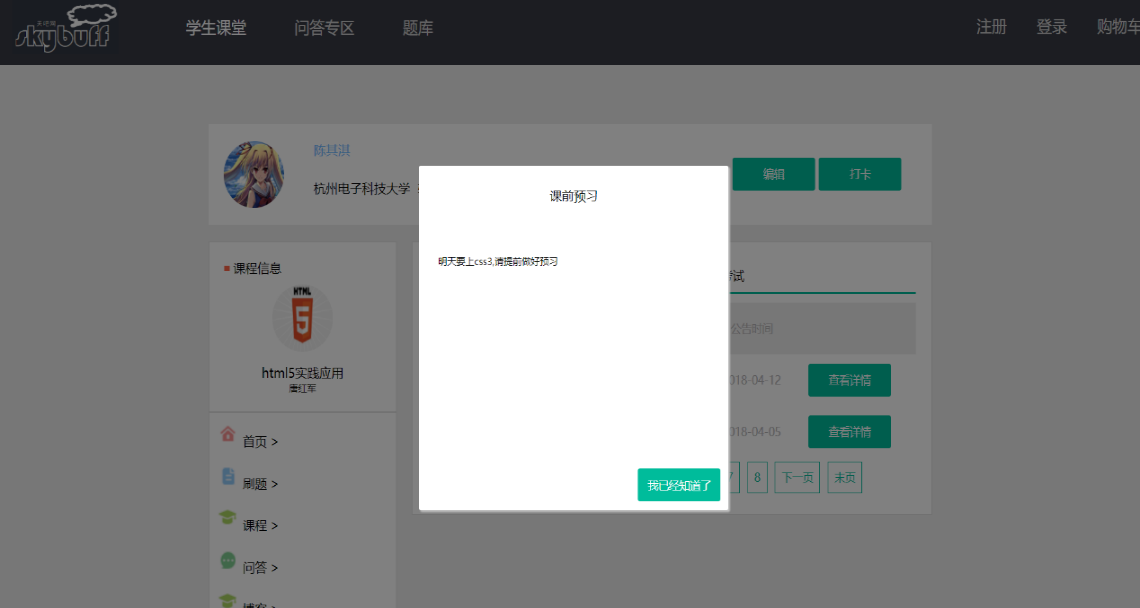


图3-12 点击详情以后

这里使用了js的遮盖层技术，点击以后，其他地方变成灰色且不可点击，并且在页面的正中央显示一个弹出框。这个遮盖层与弹窗在系统的许多地方用到，因此被封装成一个组件，供其他页面使用。如表 3-13。

|  |
| --- |
| //对话框的显示和隐藏  function showDialog(data){  showMask();  if(data!=null){  var dataId = data.id+"";    if(dataId[0]!='#')dataId = '#'+dataId;  $(dataId).css("width",data.width==undefined?400:data.width);  $(dataId).css("height",data.height==undefined?400:data.height);  var left = ($(document).width()-data.width)/2;  $(dataId).css("left",left);  if(data.id==undefined){  $(".dialog").show();  }else{  $(dataId).show();  }  }  }  function hideDialog(id){  hideMask();  if(id==undefined||id==null)$(".dialog").hide();  else $("#"+id).hide();  }  function showMask(){  $("#mask").css("height",$(document).height());  $("#mask").css("width",$(document).width());  $("#mask").show();  }  //隐藏遮罩层  function hideMask(){  $(".mask").hide();    } |

表 3-13 点击详情以后

### 3.4.2 教师课程公告的显示

教师的课堂公告模块，教师可以查看在某门课发布的具体信息，与学生的个人公告模块类似，但不同的是，教师在课堂公告模块中，可以发布新的公告，教师发布的新的公告，会被直接推送到学生的首页，方便信息的交流。而且，教师可以管理历史发布的公告，包括公告的显示和隐藏，公告的修改以及公告的删除。如图3-12。

与课堂公告模块相连的模块，有课件下载，作业上交，以及在线考试这三个模块。这也是最重要的四个模块，在这个页面可以快速跳转到其他模块，增强了用户体验。

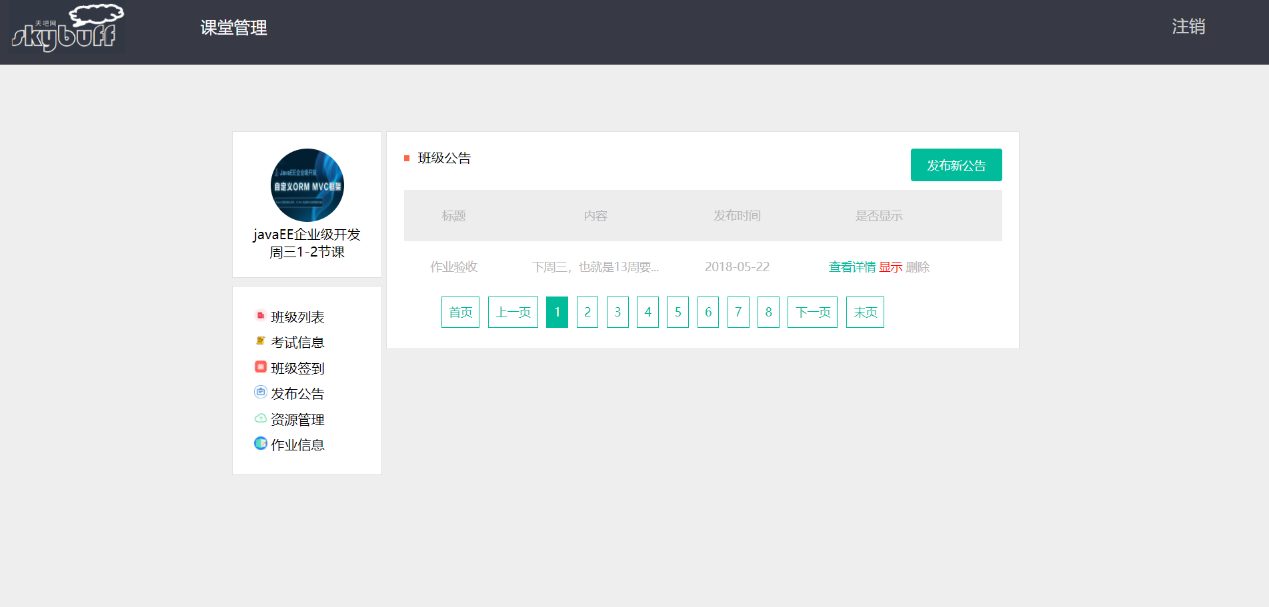


图3-14 教师的发布公告页面

## 3.5 课件下载模块

### 3.5.1 学生的课程下载

在课件下载模块，可以下载该课堂的教师上传的课堂资源。在该模块中，文件采取多级目录的方式进行显示，学生可以点击文件夹进入下一级目录。也可以点击除文件夹以外的课件资源进行下载了。多级目录的方式，使得文件存放的逻辑感更强，教师可以根据自己的命名习惯创建目录，便于学生的学习与下载，进一步提高了用户的体验。如图3-15所示。



图 3-15 课件下载页面

课件的下载，主要的逻辑代码在一个资源控制器中，这样a标签只要携带被下载文件在服务器的文件名即可。以下是文件下载的资源控制器，如表3-16。

|  |
| --- |
| @RequestMapping(value="download")  **public** **void** download(HttpServletRequest request,HttpServletResponse response)**throws** Exception {  response.setCharacterEncoding("UTF-8");  //设置ContentType字段值  response.setContentType("text/html;charset=utf-8");  //获取所要下载的文件名称  String filename = request.getParameter("url");    System.***out***.println("fileName="+filename);  //下载文件所在目录    //通知浏览器以下载的方式打开  response.addHeader("Content-type", "appllication/octet-stream");  response.addHeader("Content-Disposition", "attachment;filename="+filename);  //通知文件流读取文件  System.***out***.println("G:\\download\\"+filename);  InputStream in = **new** FileInputStream("G:\\download\\"+filename);  //request.getServletContext().getResourceAsStream(folder+filename);  //获取response对象的输出流  OutputStream out = response.getOutputStream();  **byte**[] buffer = **new** **byte**[1024];  **int** len;  //循环取出流中的数据  **while**((len = in.read(buffer)) != -1){  out.write(buffer,0,len);  }  in.close();    } |

表 3-16 ResourceController.java部分代码

### 3.5.2 教师的课件上传与管理



图3-17 教师的上传课件的页面

教师打开课件上传模块的时候，教师可以对课件上传模块进行管理，包括上传文件，创建文件夹，以及删除文件，还有重命名文件，这样极大的便利了教师对上传的课件资源进行管理。从某种程度上，提高了用户的体验。

其中，实现右键弹出菜单，还有右键弹出菜单的js代码，如文件

|  |
| --- |
| function showMenuForOneFile(e){  e.preventDefault();  var menu=document.querySelector("#hideMenu2");  menu.style.left=e.clientX+'px';  menu.style.top=e.clientY+'px';  $("#hideMenu2").show();  }  function showMenu(e){  e.preventDefault();  var menu=document.querySelector("#hideMenu");  menu.style.left=e.clientX+'px';  menu.style.top=e.clientY+'px';  $("#hideMenu").show();    }  //关闭右键菜单，很简单  $("#hideMenu2 .hideMenuOption").on("click",function(){    $("#hideMenu").hide();  $("#hideMenu2").hide();  if($(this).html()=="删除文件"){  deleteFile();  }else if($(this).html()=="重命名"){  showDialog({  id:"dialog",  width:450,  height:120    })  }  })  $("#hideMenu .hideMenuOption").on("click",function(){  $("#hideMenu").hide();  $("#hideMenu2").hide();  if($(this).html()=="创建文件夹"){  mkdir();  }else if($(this).html()=="上传文件"){  $("#file").click();  }  })  window.onclick=function(e){  $("#hideMenu").hide();  $("#hideMenu2").hide();  } |

表3-17 右键菜单的实现代码

这样，课件下载的模块就介绍完成了。

## 3.6 作业上交模块

### 3.6.1 学生的作业上交模块

学生选择某门课程，进入作业上交模块后，学生可以查看该门课程的所有发布的作业，并且可以上传作业文件给该门课程。如图



图 3-18 作业上交页面

选择某一个作业，可以查看该作业详细的作业描述，并且可以进行作业的上交。如图3-18。学生在该弹出框提交作业以后，那么还会显示最后一次提交时间，学生还可以将其重新下载回本地。同一份作业，学生多次上传只保留最后一份，其他都会被覆盖，这样有利于教师的作业审查。当然，作业是有截至时间的，如果在截至时间之后再上交作业，那么那次作业的上交是失败的。

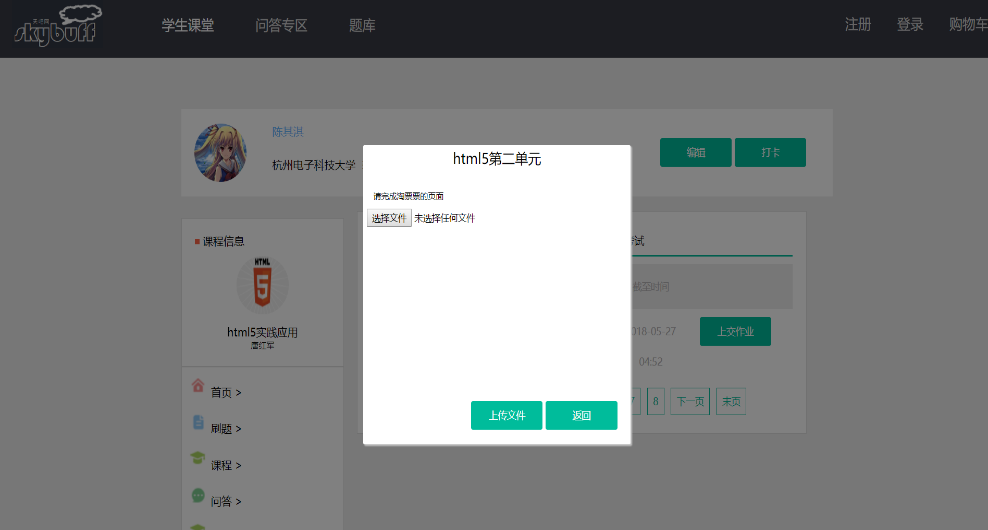


图3-19 弹出作业上交的窗口

作业的上交，在后台最重要的是文件上传控制器，该控制器提供文件上传的接口。如表3-20。

|  |
| --- |
| @RequestMapping(value = "/upload", method = RequestMethod.***POST***)  @ApiOperation(value="单个文件的上传操作", notes="主要用于资源的上传")  **public** String upload(@RequestParam("file") MultipartFile file,HttpServletResponse response) {  String name = "null";  **if** (!file.isEmpty()) {  **try** {    String fileName= file.getOriginalFilename();  String suffix = fileName.substring(fileName.lastIndexOf('.'));    name = *getUUID*()+suffix;  System.***out***.println(fatherDir);  BufferedOutputStream out = **new** BufferedOutputStream(    **new** FileOutputStream(**new** File(fatherDir,name)));  out.write(file.getBytes());  out.flush();  out.close();  } **catch** (FileNotFoundException e) {  e.printStackTrace();  **return** "上传失败," + e.getMessage();  } **catch** (IOException e) {  e.printStackTrace();  **return** "上传失败," + e.getMessage();  }  **return** ajaxReturn(response, 1, name, **null**);  } **else** {  **return** ajaxReturn(response, 1, "上传失败，因为文件是空的.", **null**);  }  } |

表3-20 FileUploadController.java部分代码

一份作业被上交了以后，用户可以查看自己在这份作业上的历史提交，并且可以再次下载。

### 3.6.2 教师的作业管理模块



图3-21教师的作业管理模块

在这个模块中，教师可以发布作业，也可以修改已经发布的作业的内容。对于学生上交的作业，可以在这个页面进行查看。点击上交下载按钮，即会出现如图3-22所示的学生作业下载列表。



图3-22点击提交下载以后的弹出窗

在这个页面上，教师就可以下载由学生提交的所有作业，方便老师对作业的下载。

## 3.7 在线考试模块

### 3.7.1 学生的在线考试模块

然后是在线考试模块，在这个模块中，老师发布完毕考试以后，学生可以在规定时间内参加这场考试，学生可以点击如图3-23的考试列表的进入考试来参加考试，或者查看考试的考试情况。



图3-23 考试列表界面

在考试列表选择一场考试即可进入考试，如图3-24是进入考试的页面。

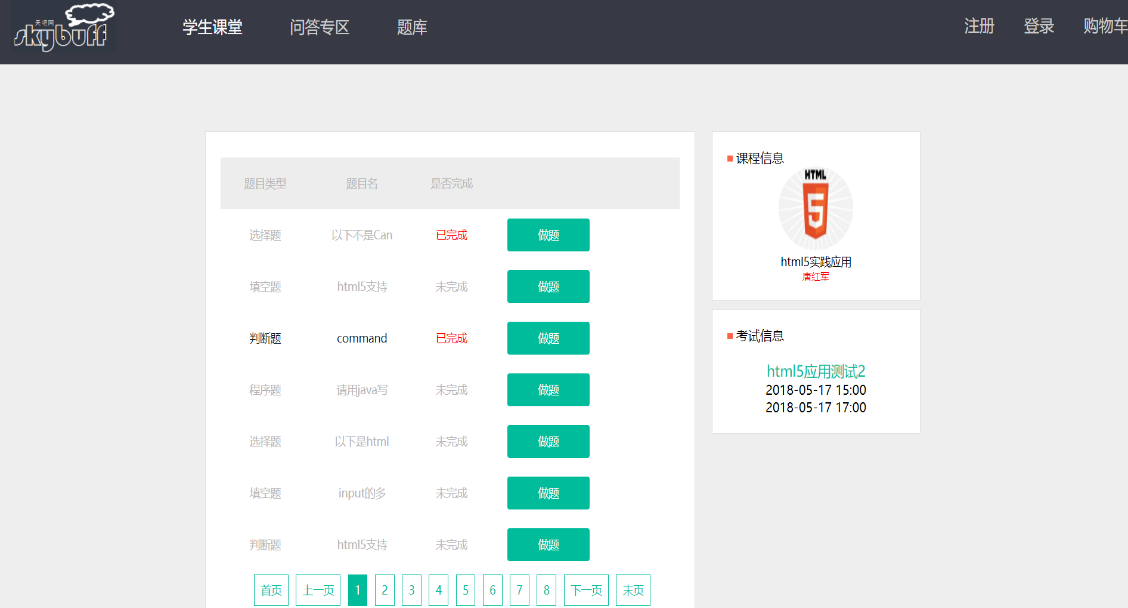


图3-24 进入考试的页面

点击做题按钮，即可弹出做题的窗口。如图3-25所示。

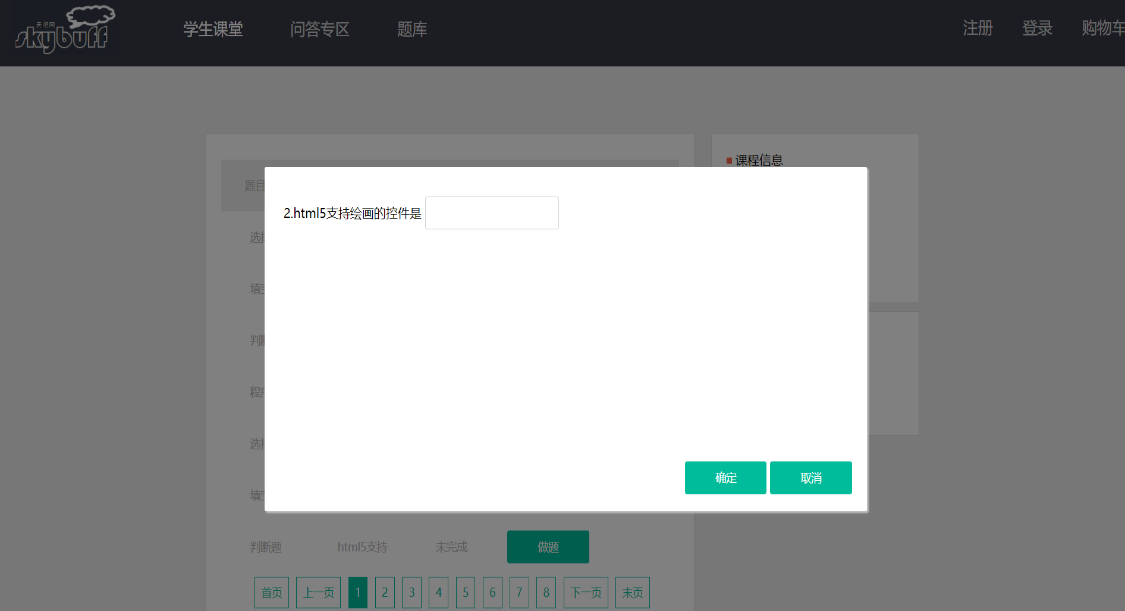


图3-25 题目的弹出框

完成题目，点击确定以后，那么学生的作答情况就会保存下来，然后题目表格中就由未完成变成已完成，这样学生可以清楚的知道自己完成了哪些题目。与作业上交模块相同的是，如果在考试期限外再做题目，题目是无法提交的。

## 3.8 签到模块

### 3.8.1 教师的签到模块



图3-26 教师的签到模块页面

在如图3-26的页面，教师可以发起签到，在同一个班级的同一个时间，教师是不能发起两次签到的。教师点击发起签到，即生效一次发布签到操作。此时，学生可以完成签到，来响应此处操作，对于没有签到的学生，教师可以查看某此签到的未到学生，方便教师的管理。如图3-27所示。



图3-17 点击“查看未到学生”弹出的窗口。

当点击查看未到学生的时候，实际上，通过ajax请求去访问UserInfoController的一个查看未到学生的接口，如表 3-27

|  |
| --- |
| @RequestMapping("selectNoSignStudentListBySignId")  **public** String selectNoSignStudentListBySignId(HttpServletRequest request,HttpServletResponse response){  **int** signId =Integer.*valueOf*( request.getParameter("signId"));  **int** classNumber =Integer.*valueOf*( request.getParameter("classNumber"));  **int** first =Integer.*valueOf*( request.getParameter("first"));  **int** each =Integer.*valueOf*( request.getParameter("each"));  List<?>list = UserInfoService.selectNoSignStudentListBySignId(signId,classNumber,first,each);  **return** ajaxReturn(response,1,"success",list);  } |

表3-28 UserInfoController.java部分代码

其中UserControler的这个接口是访问了UserService的一个方法，如表3-29所示

|  |
| --- |
| **public** List<?> selectNoSignStudentListBySignId(Integer signId,Integer classNumber,Integer first,Integer each) {  **return** mapper.selectNoSignStudentListBySignId(signId,classNumber, (first-1)\*each,each);  } |

表3-29 UserInfoService.java部分代码

而它是调用了mapper层的代码。如表3-30

|  |
| --- |
| @Select("select \* FROM user\_info AS u where u.user\_id not in (SELECT student\_id from student\_sign where sign\_id = #{signId}) and u.user\_id in (select student\_id from student\_class WHERE student\_class.class\_number= #{classNumber}) limit #{first},#{each} ")  List<HashMap<String,String>> selectNoSignStudentListBySignId(@Param("signId")Integer signId,@Param("classNumber")Integer classNumber,@Param("first")Integer first,@Param("each")Integer each); |

表3-30 UserInfoService.java部分代码

## 3.9 教师课堂模块

教师课堂模块包括教师对课程的基本管理，包括查看自己的课堂列表，创建新的课堂等。在首页点击新增班级即可进入创建班级的页面，如图3-18所示。



图3-18 创建班级的页面

# 4 总结

本学期我主要学习通过SpringBoot进行javaEE企业应用开发，SpringBoot作为非常优秀的微服务架构框架，能够让前端与后台实现完全分离，极大的方便了我的开发过程。通过本学期的学习，我java代码的编程能力得到了一些提高。一个人通过不懈地奋斗，完成了学生课堂系统的功能。在这个过程中，我学习到了许多新的知识，并且把它运用到自己的项目中去。而且，我更加熟练地掌握与运用了mybaits框架，初步学会了使用前端vue框架，并且能够获得了独立自主的完成数据库的设计，培养了逻辑设计能力，能够对需求进行比较准确分析。并且，我的debug能力已经有了不少进步。这些宝贵的编程经验以及不懈努力的精神，将会伴随我进入我以后的工作。

在这个过程中，我经常利用百度，查询许多自己以前不曾了解过的知识，实现了许多前端效果，也解决了许多的后台bug，我的个人学习能力得到了很大的提高。

在项目制作的过程中，也有许多不足的地方。比如，因为我对需求实现评估能力不足，导致之前需求文档写的东西过多，而实际过程中却没有全部实现，这也是比较遗憾的地方。

但是，当我把如此多的页面写出来的时候，我发现，有些时候，只要坚持不懈的努力，和相当大的毅力，才能完成看似非常难以完成的任务。

而且，我明白了，只有不断地学习新的知识，才能让自己在以后的工作道路上越走越宽，也越走越远。