四川农业大学

本科毕业论文（设计）

（2020 届）

|  |  |
| --- | --- |
| 题 目： | 基于web的论文审核系统设计与实现 |
| 学 院： | 信息工程学院 |
| 专 业： | 物联网工程 |
| 学生姓名： | 钟诚 学号： 201603968 |
| 导师： | 李志勇 职称： 讲师 |
|  |  |
|  |  |
| 完成日期： | 2020 年 4 月 13 日 |

目 录

1 绪论 5

1.1 开发背景 5

1.2 设计目的 5

1.3 研究现状 5

1.4 开发工具简介 5

2 系统功能设计 6

2.1 系统结构图 6

2.2 web端介绍 7

2.2.1基础功能 7

2.2.2老师模块 7

2.2.3学生模块 7

2.3 技术实现 8

3 数据库设计 8

3.1 student表 8

3.2 tutor表 9

3.3 task表 9

3.4 thesis表 10

3.5 stu\_tutor\_relation表 11

3.6 appraise表 12

4 系统功能的实现 12

4.1 主页面 13

4.1.1 整体布局的实现 13

4.2 登录模块 14

4.2.1 保持登录状态 14

4.3 学生名单模块 15

4.4 任务模块 15

4.4.1 发布任务 15

4.4.2 查看任务 16

4.5 论文模块 16

4.6 信息修改模块 18

4.7 文件模块 19

4.4.1 文件预览模块 19

4.4.1 文件打包模块 19

4.8 全局模块 19

4.4.1 后端封装模块 19

4.4.1 前端封装模块 19

5 结论 20

参考文献 21

致谢 22

附录 23

基于web的论文审核系统

物联网工程 钟诚

导师： 李志勇

**摘要**：毕业设计管理是高校管理工作中的一项重要内容。目前,很多高校的毕业设计管理还存在一些问题。在技术和需求的推动下,毕业生以原始手工报送的选题方式,暴露出了很多缺点和不足。毕业设计管理系统将服务于高校毕业生设计工作的各个方面,能够为学生和指导教师提供很多方便,也为教务处开展毕业生设计管理工作提供了信息支持。实现了web应用的快速开发，极大地便捷了学生和指导教师，显著提高了工作效率。

**关键词**：毕业论文；Vue；springboot；审核系统

**The thesis review system based on web**

Internet of Things Engineering Zhong Cheng

Supervisor: LI Zhi-yong

**Abstract**: Graduation project management is an important part of university management.At present, there are still some problems in the management of graduation design in many universities.Under the promotion of technology and demand, the original manual method of subject selection submitted by graduates as a professional unit has exposed many shortcomings and deficiencies.Graduation design management system will serve all aspects of college graduates design work, can provide a lot of convenience for students and instructors, but also for the office of academic affairs to carry out graduate design management work to provide information support.The back-end of the system is developed by springboot+mybatis+shiro framework, and the front-end is developed by vue+element- UI.The rapid development of web application is realized, which greatly facilitates students and instructors and significantly improves work efficiency.

**Key words**: thesis;Vue; springboot;Audit system

1 绪论

1.1 开发背景

随着计算机技术的不断发展, 越来越多的办公方式由线下纸质档转为线上电子档。在大学教育中毕业论文是高等教育检查学生，对专业知识掌握情况的一个重要手段，尤其是在大学阶段，普通的考试已经很难满足对于专业性知识人才的检验。所以，一个高等学校能建立一个相关论文的管理系统,才能确保高校毕业论文教学过程的顺利进行。但是国内目前的高校论文管理现状,并不是很好,尤其是在许多高校中,论文还是以人工的方式,对于论文内容加以处理和整理,不但增加了相关工作人员的工作量,同时管理的效率也相当低下。难以使高效的管理,要用现代化代替所谓的人工管理,运用网络技术,以便使论文更优等。

为了适应这一发展的需要，有些高校教师要求学生的论文以电子文档的形式予以提交。 但是大多数学生对于电子文档论文的提交方法不易理解掌握，教师对于学生提交的大量电子文档论文的管理都存在很多不便[1-2]。因此, 有必要专门设计一个论文提交及管理系统, 使学生可以方便快捷的提交论文及任务, 教师轻松的完成对这些论文及任务的管理。

1.2 研究现状

毕业设计（论文）管理系统的发展大致经历了两个阶段。一是受到计算机硬件和软件技术限制，以单机和独立模块为主题的毕业设计（论文）管理系统。这一阶段大致处于 20 世纪 90 年代末，此时的管理系统功能相对独立，大量数据只能以独享方式存放在一台计算机上，无法整合信息；二是以局域网和数据库为基础的毕业论文管理系统。在这一阶段计算机的硬件性能提升价格降低，软件技术发展迅速。这一阶段大致为 2002 年以后，此时的毕业论文管理以先进的教育思想和基于 WEB 技术的理念为指导，配合其他的技术形成了信息安全可靠、处理能力强、信息和资源共享的毕业论文管理环境。

随着信息化的迅速发展，信息化技术的发展在带动教学工作的现代化、促进优质资源共享和创新高校人才培养模式等方面正发挥前所未有的作用。各个高校对于教务工作的信息化一直非常重视，各种管理系统的开发使用提高了教学工作的效率，减轻了教务人员的负担。

目前网络化教学管理已经受到大多数国内高校的欢迎，例如教务管理系统、课程管理系统、选课管理系统等，大部分高校也有自己的论文审核系统。有的是依赖原有的教务系统的架构再添加功能，即使有一些高校使用了论文管理软件，也只是用于存储和检索的已成论文管理系统，功能十分简单，一般只包含论文的上传、下载、检索等，[3-5]。但是大部分系统主要存在两个问题：（1）导师和学生之间没有即时的沟通，审核的结果存在滞后性，老师点评后没有直接通知学生做出修改，导致整个流程的效率不高。（2）老师在评审论文的时候，给出的建议全部都由老师手动输入，使得审核过程效率低，以及审核的准确性也不高。为了提高论文完成的效率和质量，提高高校管理水平，把论文完成及管理集合到一个专业的平台中，设计和实现基于 Web 的高校毕业论文管理系统是十分必要的。

1.3 设计目的

为了方便老师对学生论文的管理，并且能及时的对学生提交的论文做出评价以及准确的提出修改意见，进行对论文审核系统的设计。论文系统可以借助日常使用的及时通信工具来进行沟通。将通讯工具的入口挂载到系统上，老师可以随时查看自己所指导的学生名单以及学生的联系方式，点击该入口后，可以快速的与学生行沟通和交流，同学就可以及时的收到老师的评价并做出相应的修改。

本文设计与实现的毕业论文管理系统的最终目的是以网络管理论文，减轻在毕业论文完成工作中的所有人员的工作分量，降低他们的工作压力，提高论文的质量，并且简化审核流程，提升效率与准确性。

1.4 开发工具介绍

IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、J2EE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。 加之有很多扩展性插件，因此具有很大优势，在编写代码、检查代码等各方面具有很大优势，减少开发成本。

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle公司。 MySQL 使用的是 SQL 语言，是用于访问数据库的最常用的标准化语言。它将数据保存在不同的表中，增加了速度并提高了灵活性，加上其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般的个人使用者和中小型企业都选择 MySQL 作为网站数据库。本系统只在校园使用，加之在毕业季时使用，平时使用率不高，因此不必要使用大型数据库，MySQL是很好的选择。

2 系统功能设计

2.1 系统结构图

本软件由客户端、服务端和数据库三大部分组成。客户端只要是用户使用以及管理员使用，用户不需要明白后端操作原理，数据库的增删改查对用户来说具体操作是透明的，用户只需要通过可视化的按钮、选项等直观操作就行，因此客户端的界面交互尤为重要。服务端是部署在服务器上的Java web程序，监听客户端传过来的http/https请求，来执行相应的具体逻辑操作，最终实现数据的增删改查。数据的持久化在数据里，统一管理数据。系统结构如图 1 所示。

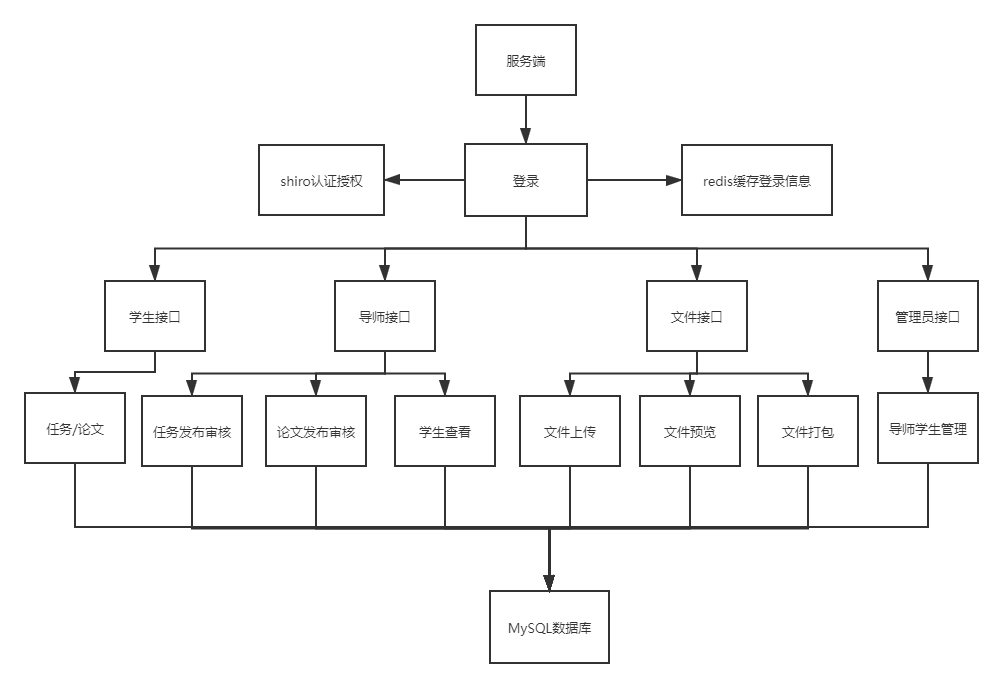


图1系统结构示意图

2.2 页面模块介绍

2.2.1 基础模块

（1）登录

登录模块用于用户登录及退出登录，进入页面后输入正确的账号和密码，并且选择对应的身份，学生或者老师。输入正确则进入对应的功能页面，输入错误或者身份不正确则会弹出登录失败的消息。退出登录则在页面的右上角，点击则会清除登录状态，删除本地登录消息，并且重新返回登录界面。

（2）信息修改

信息修改模块用于同学或老师修改添加个人信息资料，如：电话号，办公地点，email，QQ号等便于师生间的沟通交流，该模块也可对账户密码进行修改，需要先输入旧密码，再输入新密码以及确认密码。

2.2.2 老师模块

（1）任务模块

任务模块用于任务的发布及查看，老师在任务发布页面输入任务名称，介绍，选择发布对象，提交指导文件，点击创建按钮即可发布任务。查看任务，可查看总体的任务完成度，点击对应的任务，可查看对应任务的每位同学的提交情况。

（2）论文模块

论文模块是用于下载学生提交的论文并且给予评价及修改意见。每当学生修改并重新提交后，老师即可再次点评。

（3）学生名单

学生名单模块是查看老师所指导的学生的全部名单，该模块支持按年级选择，输入学号或者姓名查找对应的学生。还可以查看对应学生的具体联系方式，点击QQ，或者email后则会直接弹出聊天窗口，便于交流。

2.2.3 学生模块

（1）任务提交

学生任务模块用于学生查看老师所发布的任务或者公告，左上方可选择任务或者公告，并可以下载老师提交的指导文件，如果是任务，学生需要按照任务的介绍和要求，提交对应的作业。

（2）论文提交

论文提交模块用于学生提交论文资料，待老师评价后可查看具体的修改建议，然后可重新提交

2.3 技术实现

本系统的后端采用springboot+mybatis框架开发，Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程，MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。

使用maven管理项目，同时采用redis缓存用户登录信息，服务端与客户端通信令牌采用token形式，这种无状态认证减少了服务端的内存压力，同时增强了前后端分离开发的便利性。

使用Shiro来进行安全性的验证。Shiro 是 Java 的一个安全框架。目前，使用 Apache Shiro 的人越来越多，因为它相 当简单，对比 Spring Security，可能没有 Spring Security 做的功能强大，但是在实际工作时 可能并不需要那么复杂的东西，所以使用小而简单的Shiro 就足够了。该项目中需要对导师、学生、管理员进行认证授权，因此选用安全框架是必不可少的，加之shiro框架使用起来很方便，能加快代码开发效率，本项目中就选用了Shiro安全框架。

前端采用 vue[6]开发。Vue是MVVM模式的前端开发框架，是一套构建用户界面的框架，只关注视图层，易上手，有配套的第三方类库。让用户只关心数据即可，提高开发效率，帮助减少不必要的dom操作；双向数据绑定，通过框架提供的指令，前端只需要关注业务逻辑，不再关心dom如何渲染。，可以快速开发前端页面。

3 数据库设计

数据库端主要设计了8个表，分别是student（学生表）、tutor（导师表）、task（任务/公告表）、thesis（论文表）、file（文件表）、appraise（点评表）、student\_task\_relation（学生任务关系表）、stu\_tutor\_relation（学生导师关系表）。下面对几个相对重要的数据表进行介绍。

3.1 student表

该student表存储了学生的基本信息，包括学号、姓名、密码等等，学号和姓名由学校数据库导入，电话、邮箱、qq等信息采用学生自己修改，如果有权限也可从学校数据库里导入。数据表结构如表 1 所示：

表1 student表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 注释 | 类型 | 默认值 |
| stu\_id | 学号 | int | 非空 |
| stu\_name | 名字 | varchar | 非空 |
| password | 密码 | varchar | 统一默认密码 |
| phone | 联系电话 | varchar | 学生填写修改 |
| email | 邮箱 | varchar | 学生填写修改 |
| qq  address | 学生qq  联系地址 | Varchar  Varchar | 学生填写修改  学生填写修改 |

3.2 tutor表

该tutor表存储了导师基本信息表，导师职工号和姓名以及密码由学校数据库导入，在权限允许的情况下可以导入电话、邮箱、qq等基本信息，并可自己修改。结构如表2所示：

表2 tutor表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 注释 | 类型 | 默认值 |
| tutor\_id | 导师职工号 | int | 非空 |
| tutor\_name | 导师名 | varchar | 非空 |
| password | 密码 | varchar | 统一默认密码 |
| phone | 联系电话 | varchar | 导师填写修改 |
| email | 邮箱 | varchar | 导师填写修改 |
| qq | 导师qq | varchar | 导师填写修改 |

3.3 task表

该task表主要存储了导师发布的任务信息，当type为“任务”时，该记录是任务相关数据，含有截至时间、导师评价关系信息，当type为“公告”时，该记录是公告相关数据，不包含截至时间以及评价等信息，学生不需要提交相关资料，因此也不需要文件和完成度显示。表结构如下：

表3 task表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 注释 | 类型 | 默认值 |
| id | 任务唯一id | Long | MySQL生成 |
| name | 任务名 | varchar | 任务/公告名 |
| type | 任务类型 | varchar | 任务/公告 |
| end\_time | 任务截至时间 | timestamp | 为任务时填写 |
| description | 描述 | varchar | 用户填写 |
| stu\_ids | 学号数组 | text | 导师发布对象的 |
| file\_id | 文件id | bigint | null |
| status\_flag | 删除标识 | tinyint | 1：启用 |
| create\_time | 创建时间 | timestamp | 当前时间戳生成 |
| creator\_id | 创建者id | int | 发布任务导师id |
| finished | 任务完成度 | varchar | 0/0 |

3.4 thesis表

该thesis表存储的是学生提交的论文相关数据，学生在提交论文时需要填写论文title、type，上传论文文件后将返回的文件id一起传到后台存储。老师点评论文后才会生成appraise\_id。通过status\_flag来实现软删除，只是将该字段置为2便删除，1则是启用，这样能找回误操作的数据。结构如表4：

表4 task表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 注释 | 类型 | 默认值 |
| id | 论文id | bigint | MySQL生成 |
| title | 论文标题 | varchar | 学生提交填写 |
| type | 论文类型 | varchar | 学生提交填写 |
| file\_id | 文件id | bigint | 学生提交文件id |
| appraise\_id | 点评id | bigint | 自动生成 |
| create\_time | 创建时间 | timestamp | 自动生成 |
| update\_time | 修改时间 | timestamp | 更新时间戳 |
| stu\_id | 学生学号 | int | 从服务系统获取 |
| status\_flag | 删除标志 | tinyint | 默认1 |

3.5 stu\_tutor\_relation表

该表为学生导师关系表，存储的是学生和导师双向选择的数据，一个学生对应了一个导师，一个导师对应多个学生。表结构如下：

表5 stu\_tutor\_relation表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 注释 | 类型 | 默认值 |
| id | 学生导师关系id | bigint | Mysql自动生成 |
| college | 学院 | varchar | 非空 |
| major | 专业 | varchar | 非空 |
| grade | 年级 | int | 非空 |
| classes | 班级 | varchar | 非空 |
| stu\_id | 学号 | int | 非空 |
| stu\_name | 学生姓名 | char | 非空 |
| tutor\_name | 导师姓名 | char | 非空 |
| tutor\_id | 导师职工号 | int | 非空 |

3.6 appraise表

该appraise表存储的是导师点评的数据，type为1时表示点评的是任务，为2时表示点评的是论文，学生根据导师的点评内容进行修改相应的文件。表结构如下：

表6 appraise表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 注释 | 类型 | 默认值 |
| 字段名 | 注释 | 类型 | 默认值 |
| id | 点评id | bigint | Mysql自动生成 |
| score | 评分 | tinyint | 导师自己提交 |
| format | 格式问题 | varchar | null |
| format\_labels | 格式问题分类 | varchar | null |
| content | 内容问题 | varchar | null |
| content\_labels | 内容问题分类 | varchar | null |
| type | 评论类型 | tinyint | 默认1 |
| task\_thesis\_id | 任务/论文id | bigint | 非空 |

4系统功能的实现

4.1 主页面

结合论文的研究背景及意，论文系统的主页面效果图如图2所示，主页面分为首部（header），导航栏（navigation bar），主要内容（container），尾部（footer）。首部展示了当前用户信息，链接到教务处和课程平台的链接，退出登录的按钮。导航栏使用vue-router进行路由跳转。主要内容部分包含当前路由所对应的页面。尾部则显示网站基本信息。

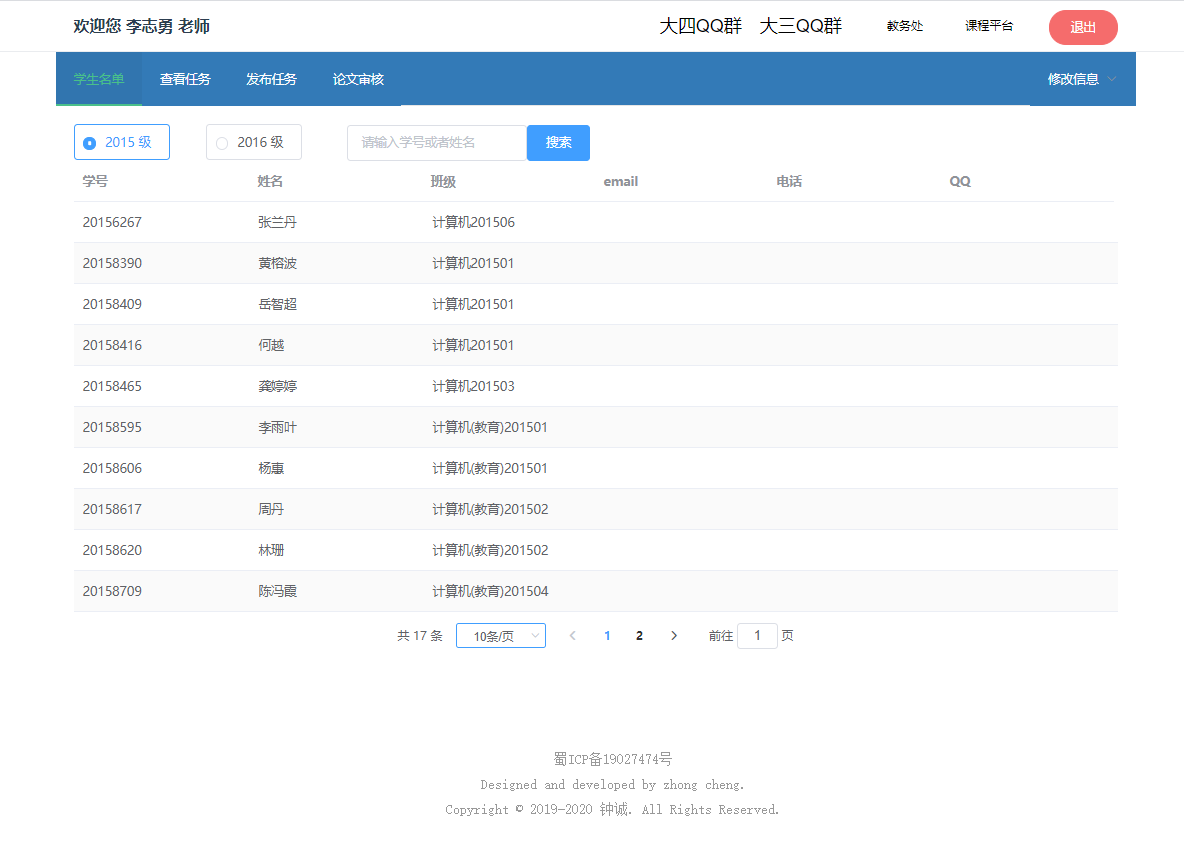


图2系统主页面

4.1.1 整体布局的实现

基本页面布局复杂度不高，大部分都可以按照文档流进行基本的布局[7]，采用组件化思想进行页面进行细分[8]。

4.2 登录模块

“登录”界面如图 4 所示，在其中输入正确的账号密码后选择对应的身份，点击登录按钮即可进入系统



图3登录页面

4.2.1 保存登录状态

在用户登录成功后，会返回对应的姓名，id，和一串用于身份验证的token，Token采用JWT加密，将用户基本信息进行base64编码后，再用密钥将编码后的数据进行对称加密，使之成为用户每次请求的令牌[9]。只有服务端对令牌解密通过后才能进行具体的业务处理，使数据交互与认证授权更为方便[10]。前端将token存入localStorage里面，并使用axios.headers让请求头携带token。如何服务器收到请求后验证token的合法性以及时效性，如果身份不正确或者太长时间为操作导致token过期或者失效，则返回登录界面。

4.3 学生名单模块

学生模块可选择年级，搜索学生学号或者姓名显示对应的学生。点击email或者QQ号即可调用本地对应的应用程序与学生交流。页面如图4：



图4学生名单页面

上方为选择年级的按钮，数据的划分的实现代码如下，搜索功能的代码附录（3）

4.4 任务模块

4.4.1 发布任务

由老师填写好名称，类型，介绍，发布对象后点击创建按键即可创建任务。界面如图5所示：

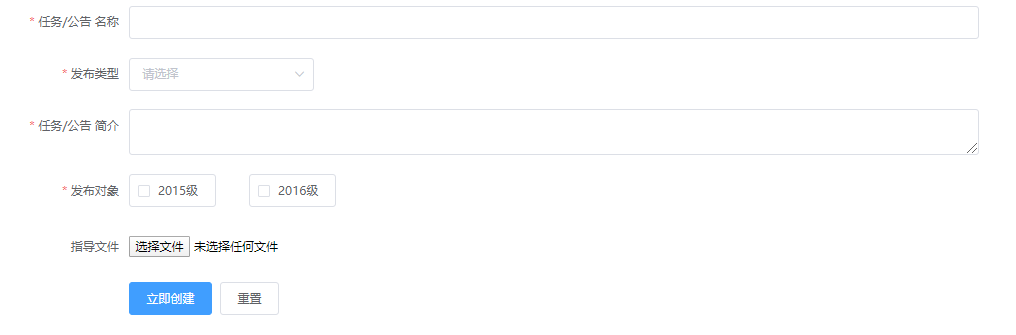


图5发布任务页面

如果需要提交文件则需要先把文件提交至服务器，服务器返回文件的file\_id，前端再将返回的file\_id添加进表单再向服务器发送创建新任务的请求[11]，整个过程采用Priomise链式调用，用catch方法捕获异常，部分代码如附录（4）

4.4.2 查看任务

老师可查看或者删除发布的任务，以及学生的提交情况，点击任务完成的按钮后会出现每位同学完成的情况。页面如图6所示



图6系统主页面

4.5 论文模块

老师可以查看同学们提交的论文类型，题目等，评价栏中如果蓝色有点评按钮则说明学生在导师点评后重新提交过论文，点击下载按钮则会下载学生提交的文件。可在输入框中输入年级，姓名或者学号搜索学生，点击右上方的复制按钮可将已经处理论文的学生名单复制到粘贴板上，再点击对应的QQ群，可将复制的内容发至群聊里，提醒学生做出修改。页面如图7所示。



图7查看论文页面

点击评论则会弹出一个模态框，可以点评学生的论文。老师选择或者填写需要修改的部分后，给学生的论文打分。点击提交后即可点评成功。学生查看点评内容并且重新提交后，教师可再次点评。点评界面如图8所示

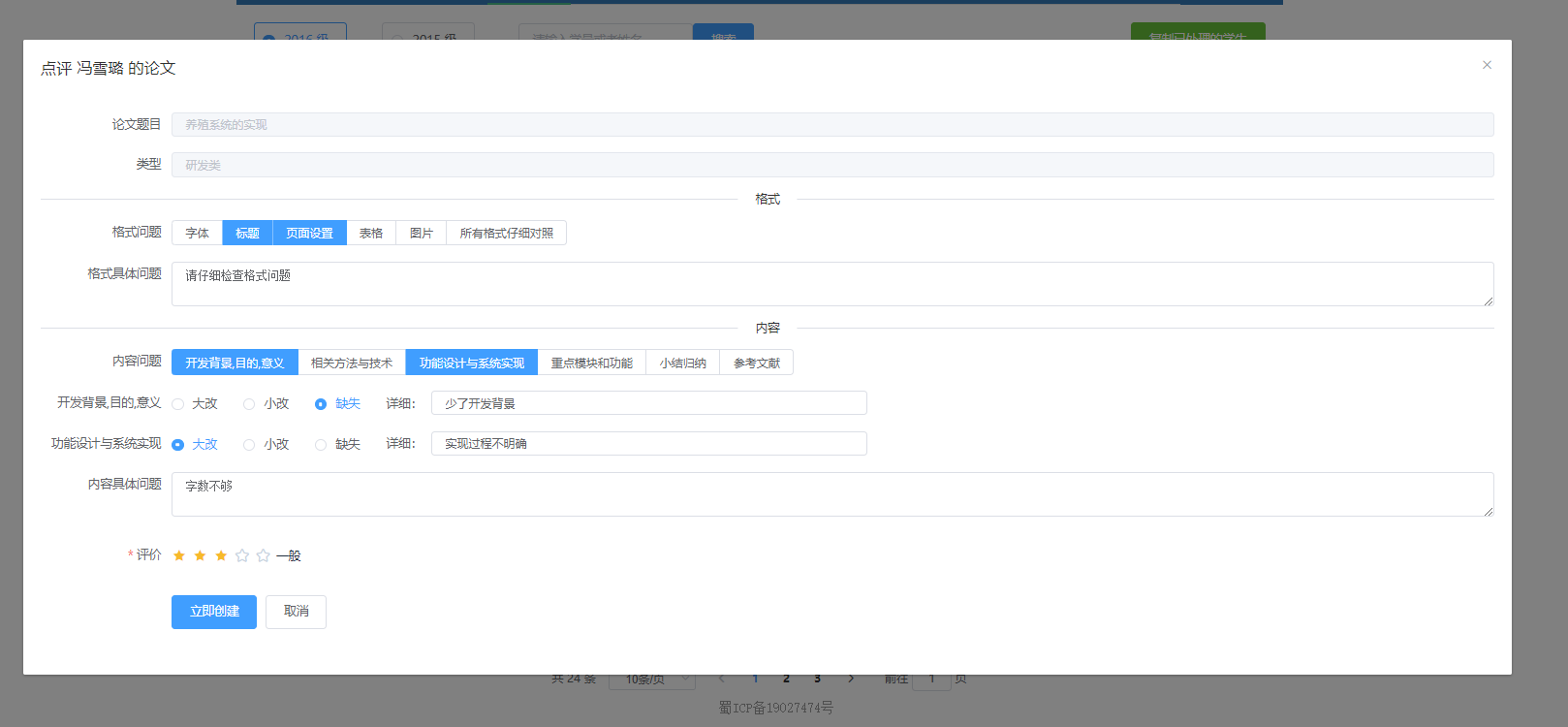


图8点评页面

评价栏的查看点评按钮可以显示这次或者上一次的点评内容，页面如图9

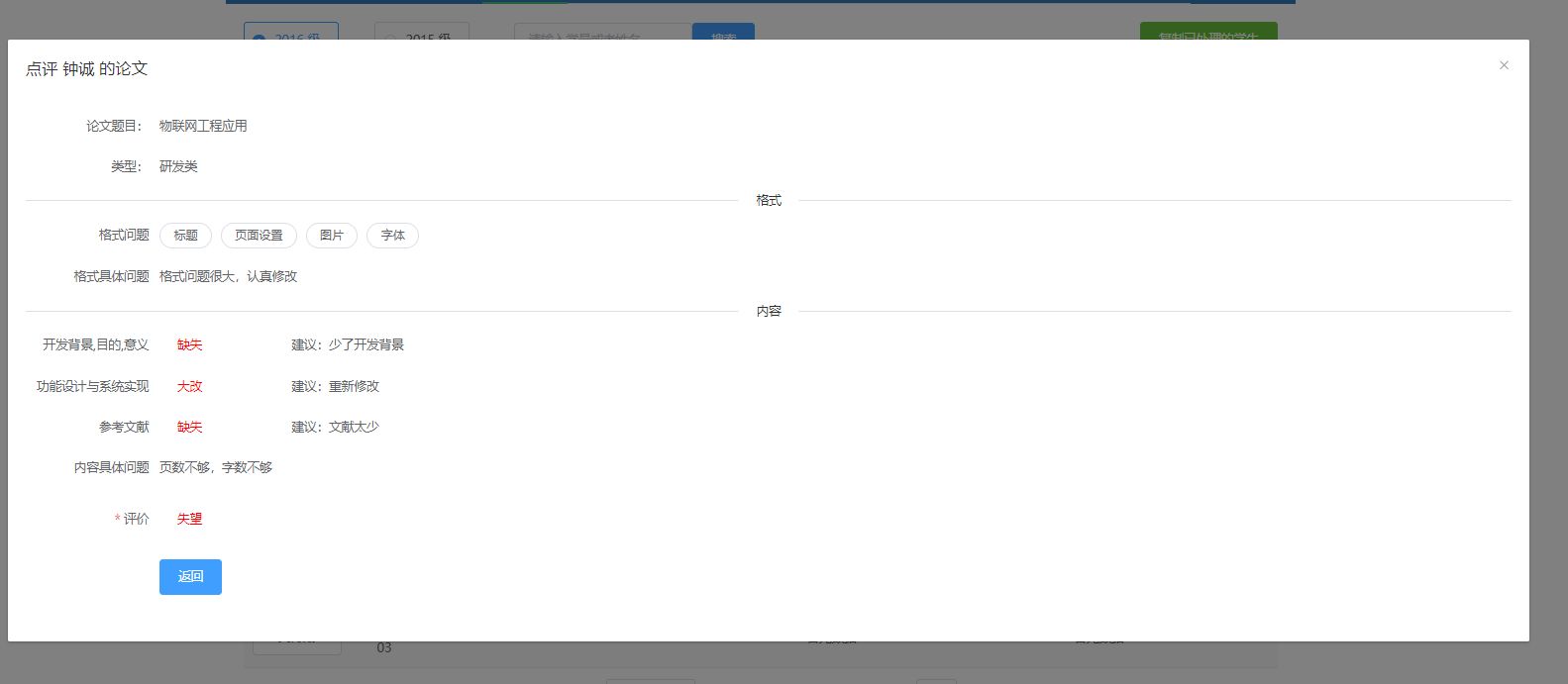


图9查看点评页面

在此模块中，点击学生的姓名则会弹出学生的所有信息包括，学号，电话号，QQ号等，如果点击页面中的QQ号则会弹出与该学生的聊天界面，可快速与学生沟通。页面如图10



图10学生信息页面

4.6信息修改模块

用户可修改密码以及身份信息，修改密码需要输入旧密码，和新密码以及确认密码。修改信息需要填入邮箱，电话号，办公地址，QQ。如果格式不正确则不能提交。页面如图9所示



图9信息修改页面

验证代码如附录(5)。

4.7文件模块

4.7.1 文件预览模块

用户点击文件预览，传递文件id给后端，后端在文件服务器找到该文件，判断该文件是否需要转码（因为pdf文件和txt文件是可以直接预览），将需要转码的文件转成pdf进行预览。文件预览代码如附录(6)

4.7.2 文件打包模块

用户前端将需要打包的文件id数组传到后端，后端进行验证文件id是否可以获取，没有异常则通过文件id在文件服务器中找到具体文件，将对应的文件打包成zip压缩包，再写入响应体中，供前端下载。文件打包代码如附录(7)，接口调用zip打包工具类，将对应文件进行打包成zip文件，再返回给前端。

4.8全局模块

4.8.1 后端封装模块

后端统一封装了返回的json对象，返回对象包括三个字段：status（状态）、msg（消息）、data（数据）。status表示后端操作成功与否，以及各种状态码，msg则是当操作出错时，返回的错误信息，data则是操作成功后返回的具体数据。统一封装json对象的好处则是有个规范的json对象，在前后端交互时，能更方便。通过封装静态类方法，使接口返回更为方便的调用返回对象。关键代码如附录(8)

4.8.2 前端封装模块

前端使用的请求方式为axios，Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。本系统对axios做了封装，利用axios. Interceptors在请求头中添加身份验证的token,再使用axios.interceptors.response.use()方法来设置请求，响应拦截，使用Element Ui的给请求，响应处理添加动画效果，并且对状态码403,401等状态码进行相应的处理，对所有的请求错误进行捕获与处理。最后将封装好的模块利用Vue.prototype挂载到全局。

5 结论

通过测试，本应用实现了以下主要功能：修改个人信息，任务发布与提交，并且可以查看所选学生的提交内容，还有论文的点评，学生名单的查看以及搜索等功能。毕业设计管理系统依据高校项目化教学、人才培养的要求及毕业设计采用原始手工报送的现状为研究的出发点，通过深入研究和探索，设计出适用于高校的毕业设计管理系统，脱离了以前纸质化、杂乱无序的管理过程，系统具有使用方便、操作简单，效率较高，界面友好，简洁，等特色实现了网络化全面管理，有效提高了论文管理的效率。

通过这一次的毕业设计，体会到了今后的开发过程中，一定要先对项目有一个整体的设计，比如如何进行项目的框架的划分，理清模块之间的关联，做好统一的命名规则和代码规范等，相信这次的经验对今后的项目开发会有很大的帮助。

参考文献

[1]李莉“互联网+”背景下毕业设计管理系统的研究与设计[J]. 信息与电脑(理论版), 2019: 101-103

[2]曾婷.高校毕业论文(设计)管理系统的设计与实现[J].软件导刊(教育技术),2013(3):81-82.

[3]基于MVC模式的毕业论文(设计)管理系统设计与实现[J]. 樊伟宏,杨文婷,王昊,张仕霆,甘晓炫. 软件导刊. 2017(04)

[4]毕业设计管理系统[J]. 孟显英. 中国科技信息. 2010(13)

[5]基于Web的高校毕业论文管理系统的设计与实现[D]. 刘飞.江西财经大学 2017

[6]Vue.js:a lightweight and efficient front-end componentization scheme[J]. Evan Y. Programmer . 2015

[7]DIV+CSS技术在网页布局中的应用[J]. 王海榕. 电子技术与软件工程. 2017(07)

[8]网站前端模块化开发策略研究[J]. 肖慧明. 数字技术与应用. 2018(03)

[9]Improving Data Transmission in Web Applications via the Translation between XML and JSON. Guanhua Wang. Communications and Mobile Computing (CMC) 2011 Third International Conference on . 2011

[10]前后端数据交互的安全机制[J]. 巫万坚,吴小勇. 计算机与现代化. 2019(07)

[11]网络文件传输机制探析[J]. 曾高峰. 现代商贸工业. 2013(01)

致谢

在毕业之际，我真诚地感谢给我无私帮助的信息工程学院的全体老师，他们的教诲为本文的论述提供了理论基础，并创造了许多必要条件与学习机会。还有朝夕相处的班的同学们，与我度过了四年的大学美好时光，在此我也一并深深的感谢你们！最后，我要感谢我的父母及家人，没有人比你们更爱我，你们对我的关爱让我深深感受到了生活的美好，谢谢你们一直以来给予我的理解、鼓励和支持，你们是我不断取得进步的永恒动力。

附录

（1）搜索功能

select (e) {

let result = []

for (let i of this.allData) {

if (String(i.stuId).indexOf(e) !== -1 || i.stuName.indexOf(e) !== -1) {

result.push(i)

}

}

this.tableData = result

this.setPaginations()

},

（2）发布任务功能的请求过程

submitForm (formName) {

let from = this.ruleForm

this.$refs[formName].validate((valid) => {

if (valid) {

from.studentIds = []

from.endTime = from.date1 + " " + from.date2

for (let i of from.arr) {

from.studentIds.push(...i)

}

let formData = new FormData()

let file = this.fileList

formData.append('file', file)

if (this.fileList) {

// --------------------------------------有文件 上传文件

sendFile(formData).then((res) => {

if (res.status == 200) {

from.fileId = res.data.id

//----------------------------------再上传任务

sendTask(from).then((res) => {

if (res.status == 200) {

this.$router.push('/teacher/taskList')

}})}})

} else {

sendTask(from).then((res) => {

if (res.status == 200) {

this.$router.push('/teacher/taskList')

} else {

}})}

} else {

return false;

}});},

（3）处理数据

import { getStudent } from '@/Api/teacher.js' //获取封装好的axios接口

getStudent({ pageSize, pageNum }).then((res) => {

this.allData = res.data.list

let map = new Map() // 采用map的数据结构

for (let i of this.allData) {

if (map.has(i.grade)) {

map.set(i.grade, [...map.get(i.grade), i])

} else {

map.set(i.grade, [i])

}

}

this.grade = map

this.tableData = this.allData //双向绑定改变页面

this.setPaginations() //分页

})

（4）验证功能

var validatePass2 = (rule, value, callback) => {

if (value === '' && this.ruleForm.pass != '') {

callback(new Error('请再次输入密码'));

} else if (value !== this.ruleForm.pass) {

callback(new Error('两次输入密码不一致!'));

} else {

callback();

}};

submitForm (formName) {

this.$refs[formName].validate((valid) => {

if (valid) {

let newPassword = this.ruleForm.pass

let oldPassword = this.ruleForm.oldPass

changePass({

oldPassword, newPassword

}).then((res) => {

window.alert(res.msg)

})}});},

（5）文件预览功能

public ResponseResult viewFile(HttpServletResponse httpServletResponse, @RequestBody FileViewReq fileViewReq){

FileDto fileDto = fileService.getFileById(fileViewReq.getFileId());

String suffix = fileDto.getFileSuffix();

if (suffix.equals(Const.PDF\_SUFFIX)){

ResponseResult.createBySuccess(Const.DOWNLOAD\_PATH+fileDto.getFileName());

}else if(Const.DOC\_SUFFIX.equals(suffix)||Const.DOCX\_SUFFIX.equals(suffix)||

Const.XLS\_SUFFIX.equals(suffix)||Const.XLSX\_SUFFIX.equals(suffix)){ File file = new File(Const.UPLOAD\_PATH+fileDto.getFileName());

if (!file.exists()){

BusinessAssert.throwException("文件不存在");

}

File targetFile = new File(Const.UPLOAD\_PATH+fileDto.getFileName()+"."+Const.PDF\_SUFFIX); try {

documentConverter.convert(file).to(targetFile).execute();  
 return ResponseResult.createBySuccess

(Const.DOWNLOAD\_PATH+fileDto.getFileName()+"."+Const.PDF\_SUFFIX);  
 } catch (OfficeException e) {  
 e.printStackTrace();  
 BusinessAssert.throwException("服务器异常");  
 }  
 }else {

BusinessAssert.throwException("该文件不支持预览");

}

return ResponseResult.createBySuccess();

}

（6）文件打包功能

public static String zip(List<String> filePath, Boolean enableCustomName, List<String> fileName) throws Exception {  
 //要生成的压缩文件  
 ZipOutputStream out = new ZipOutputStream(new FileOutputStream(Const.UPLOAD\_PATH+System.currentTimeMillis()+Const.ZIP\_NAME));  
 File[] files = new File[filePath.size()];  
 byte[] buffer = new byte[1024];  
 for (int i = 0; i < filePath.size(); i++) {  
 files[i] = new File(filePath.get(i));  
 }  
 for (int i = 0; i < files.length; i++) {  
 FileInputStream fis = new FileInputStream(files[i]);  
 if (enableCustomName) {  
 out.putNextEntry(new ZipEntry(System.currentTimeMillis()+fileName.get(i)));  
 } else {  
 out.putNextEntry(new ZipEntry(files[i].getName()));  
 }  
 int len;  
 while ((len = fis.read(buffer)) > 0) {  
 out.write(buffer, 0, len);  
 }  
 out.closeEntry();  
 fis.close();  
 }  
 out.close();  
 return Const.UPLOAD\_PATH+System.currentTimeMillis()+Const.ZIP\_NAME;  
}

（7）后台封装代码

public class ResponseResult<T> implements Serializable {  
 private int status;  
 private String msg;  
 private T data;  
 private ResponseResult(int status, String msg) {  
 this.status = status;  
 this.msg = msg;  
 }  
  
 private ResponseResult(int status, String msg, T data) {  
 this.status = status;  
 this.msg = msg;  
 this.data = data;  
 }  
 private ResponseResult(int status, T data) {  
 this.status = status;  
 this.data = data;  
 }  
 public int getStatus(){  
 return status;  
 }  
 public T getData(){  
 return data;  
 }  
 public String getMsg(){  
 return msg;  
 }  
 public static <T> ResponseResult<T> createBySuccess(){  
 return new ResponseResult<T>(ResponseCode.SUCCESS.getCode(),"操作成功");  
 }  
 public static <T> ResponseResult<T> createBySuccess(T data){  
 return new ResponseResult<T>(ResponseCode.SUCCESS.getCode(),data);  
 }  
 public static <T> ResponseResult<T> createByError(){  
 return new ResponseResult<T>(ResponseCode.ERROR.getCode(),  
 ResponseCode.ERROR.getMsg());  
 }  
 public static <T> ResponseResult<T> createByErrorCodeMessage(int errorCode, String errorMessage){  
 return new ResponseResult<T>(errorCode,errorMessage);  
 }  
}