



**本科毕业设计（论文）**

基于SpringBoot框架的毕业论文管理系统

|  |  |
| --- | --- |
| 学院 | 计算机学院 |
| 专业 | 计算机科学与技术 |
| 年级班别 | 计算机科学与技术（2）班 |
| 学号 | 3118004130 |
| 学生姓名 | 吴海韬 |
| 指导教师 | 陈平华 |

2022年4月

摘要

近年来，随着高校的逐年扩招，本科毕业人员日益增多，学校处理毕业论文的压力也不断增大。为了解决的高等院校毕业论文管理流程繁杂，交流困难，文件信息繁多的问题，本文综合了当前后端框架技术，设计了基于SpringBoot框架的毕业论文管理系统，使用JWT、Vue路由技术实现用户认证、权限管理，Axios实现http请求，实现前后端分离，为管理员提供对整个系统以及其他使用者信息的统筹管理功能，为学生提供选择指导老师发布的毕业课题和上传过程文档的功能，为指导老师提供根据学生的需求及任务书，上传毕业课题，审核过程文档功能，审核课题是否通过功能，论文成绩管理。同时使用Simhash算法，通过分词：提取文本所有关键词；计算Hash值：通过hash函数计算成hash值；加权：对Hash值根据出现频率赋权；合并：合加所有加权完的序列值；降维：降维成只为0，1的二进制数，再通过计算机汉明距离计算得文本相似度，最终实现查重功能。该系统有效减少了管理员的工作量，提高高校对论文管理与论文质量把控的工作效率，便利学生与导师之间交流，查重功能有利于再毕业论文初期，不借助校外查重网站的情况下，对毕业论文质量控制由初步的把握，这对推动高等院校全面电子化、信息化具有重要意义。

关键词：SpringBoot，simhash，汉明距离，毕业论文管理

Abstract

In recent years, with the expansion of the operating scale of colleges and universities, the total number of undergraduates and college graduates has continued to increase and the pressure on schools to deal with graduation thesis has also increased. In order to solve the problems of complicated graduation thesis management process, communication difficulties,and large amounts of file information in colleges and universities, this paper syntheszes the current back-end framework technology, designs a graduation thesis management system based on SpringBoot framework, uses JWT, Vue routing technology to achieve user authentication, permission management, Axios implements http requests, realizes front-end and back-end separation, provides user information management, student management, instructor management, topic management, graduation topic selection, process document information management, Dissertation score management and other functions. At the same time, simhash algorithm is used to extract all keywords of the text by participle: extract all keywords of the text; calculate the hash value: calculate the word obtained by the participle as the hash value; weighting: assign the hash value according to the frequency of occurrence; merge: add all the weighted sequence values; reduce the dimension: reduce the dimension into a binary number of only 0,1, and then calculate the text similarity through the computer Hamming distance, and finally realize the double check function. The system effectively reduces the workload of administrators, improves the work efficiency of colleges and universities on thesis management and thesis quality control, facilitates communication between students and tutors, and the check function is conducive to the early stage of the graduation thesis, without the help of the off-campus check website, the quality control of the graduation thesis is initially grasped, which is of great significance for promoting the comprehensive electronic and informatization of colleges and universities.

Key words: SpringBoot ,simhash , Hamming distance ,graduation thesis management

**目录**

[1 绪论 1](#_Toc102171218)

[1.1 研究背景和意义 1](#_Toc102171219)

[1.2 研究现状 1](#_Toc102171220)

[2 相关技术介绍 3](#_Toc102171221)

[2.1 SpringBoot框架 3](#_Toc102171222)

[2.2 Vue框架 3](#_Toc102171223)

[2.3 MySQL 数据库 3](#_Toc102171224)

[2.4 JWT(Json Web Token) 3](#_Toc102171225)

[2.4.1 JWT概念 3](#_Toc102171226)

[2.4.2 JWT认证流程 4](#_Toc102171227)

[3 系统分析 5](#_Toc102171228)

[3.1 可行性分析 5](#_Toc102171229)

[3.1.1 技术可行性 5](#_Toc102171230)

[3.1.2 操作可行性 5](#_Toc102171231)

[3.2 系统需求分析 5](#_Toc102171232)

[3.2.1 功能需求 5](#_Toc102171233)

[3.2.2 安全性需求 7](#_Toc102171234)

[3.2.3 用例图 7](#_Toc102171235)

[3.2.4 时序图 8](#_Toc102171236)

[4 数据库设计 12](#_Toc102171237)

[4.1 E-R图设计 12](#_Toc102171238)

[4.2 数据字典 15](#_Toc102171239)

[5 论文查重算法设计 19](#_Toc102171240)

[5.1 Simhash算法 19](#_Toc102171241)

[5.1.1 计算simhash值 19](#_Toc102171242)

[5.1.2 计算汉明距离 20](#_Toc102171243)

[6 系统设计 21](#_Toc102171244)

[6.1 系统概要设计 21](#_Toc102171245)

[6.1.1 系统框架图 21](#_Toc102171246)

[6.1.2 系统功能模块图 22](#_Toc102171247)

[6.1.3 类模块设计 23](#_Toc102171248)

[6.2 系统详细设计 24](#_Toc102171249)

[6.2.1 前端设计与实现 24](#_Toc102171250)

[6.2.2 后端设计与实现 24](#_Toc102171251)

[7 系统测试 29](#_Toc102171252)

[7.1 测试用例 29](#_Toc102171253)

[7.1.1 登录、注册模块测试 29](#_Toc102171254)

[7.1.2 用户管理模块测试 29](#_Toc102171255)

[7.1.3 学生管理模块测试 30](#_Toc102171256)

[7.1.4 导师课题管理模块测试 31](#_Toc102171257)

[7.1.5 导师审核模块测试 31](#_Toc102171258)

[7.1.6 导师成绩管理模块测试 31](#_Toc102171259)

[7.1.7 学生毕业选题模块测试 32](#_Toc102171260)

[7.1.8 学生提交文档模块测试 32](#_Toc102171261)

[7.2 测试结果 32](#_Toc102171262)

[7.2.1 登录、注册模块测试结果 32](#_Toc102171263)

[7.2.2 用户管理模块测试结果 33](#_Toc102171264)

[7.2.3 学生管理模块测试结果 34](#_Toc102171265)

[7.2.4 导师课题管理模块测试结果 35](#_Toc102171266)

[7.2.5 导师审核模块测试结果 36](#_Toc102171267)

[7.2.6 导师成绩管理模块测试结果 37](#_Toc102171268)

[7.2.7 学生毕业选题模块测试结果 37](#_Toc102171269)

[7.2.8 学生提交文档模块测试结果 38](#_Toc102171270)

[结论 40](#_Toc102171271)

[参考文献 41](#_Toc102171272)

[致谢 42](#_Toc102171273)

# 绪论

* 1. 研究背景和意义

毕业论文的写作是检验高校毕业生大学四年学习结果的精华，也是检验高等院系对于本科生的教育成果的主要环节。一位本科毕业生要经历三年的理论学习，才能在最后一学年进行毕业论文的写作，在写作期间，需经历论文开题报告撰写、论文正文撰写、毕业论文查重、毕业论文答辩，才能取得学位证书和毕业证书。但是本科的毕业论文撰写时间短暂，从论文的社会效应和学术效应，很难说有较大的影响。但本科毕业论文的撰写是对本科生三年理论研究的综合实践，同时也是对毕业生学术水平的初步训练，是保障毕业生成才的重要手段。流程的把控和对质量的检测是保证学生顺利撰写毕业论文的重要举措，也是高等院校保证论文质量的重要前提

高等院校毕业论文管理流程复杂，交流困难，文件信息庞杂。一般要经历繁多的步骤，需要导师、学生、教务管理员三方协作交流。随着中国高等院校的进一步扩招，以往的微信交流传送文件、人工线下搜集资料的方法已经难以满足日渐庞大论文管理工作。而如近高等院校信息化程度不断提高，互联网软件发展迅猛。结合当今互联网软件技术，急需设计出一个供学生、导师、管理员使用的毕业论文管理系统

* 1. 研究现状

从查阅文献的来看，如今已开发的毕业论文管理系统主要是以浏览器/服务器框架为主，其特点在于只需在浏览器键入网址即可使用。但是如近大多数的毕业论文管理系统使用的是 Spring+SpringMVC 为后端服务技术，但是当前 Spring 技术已经不再流行，更多的人更想使用 SpringBoot 为后端技术，相较于 Spring 框架SpringBoot 更简单，减少了很多的样板代码[6]

于此同时，当前毕业论文设计管理系统大多和传统的管理系统类似，只有简单的增加、删除、修改、查询、上传、下载文档功能，而因为有第三方查重网站，如维普、知网的存在，缺乏系统本身对毕业论文的质量控制手段，缺少学校端对毕业论文的初步审核，没有重复利用学校以往毕业论文的数据库。因此，本系统将在基础的管理系统的基础上，以学校的论文库为，基础增加查重功能。

# 相关技术介绍

* 1. SpringBoot框架

SpringBoot在一个全新的轻量级框架，是所有基于 Spring 开发的项目的起点，其设计目的是为了尽可能快的运行Spring 应用程序并且尽可能减少配置文件，同时默认配置了很多框架的使用方式。其拥有四个主要特性：

1、SpringBoot Starter

将常用的依赖分组进行了整合，将其合并到一个依赖中，这样就可以一次性添加到项目的Maven或Gradle构建中；

2、自动配置

SpringBoot的自动配置特性利用了Spring4对条件化配置的支持，合理地推测应用所需的bean并自动化配置他们；

3、命令行接口

SpringBoot的CLI发挥了Groovy编程语言的优势，并结合自动配置进一步简化Spring应用的开发；

4、Actuatir

为SpringBoot应用的所有特性构建一个小型的应用程序。

* 1. MySQL 数据库

MySQL数据库是关系型数据库管理系统，其性能卓越，服务稳定，极少出去宕机，同时历史悠久，代码开源且无版权限制，学习成本低，为多种开发语言提供API接口。许多项目中将博客、用户、管理员等信息都保存在 MySQL 数据库的表中[10]。

* 1. JWT(Json Web Token)

JWT是JSON Web Token的简称，从内容上看，只是一个包含某种意义数据的JSON串，其最重要的特性是确认其自身是否有效。因为JWT携带了信息验证码MAC(Message Authentication Code)，这意味着不需要借助于第三方服务或者在多个请求之间将其保存在内存中,只需看JWT本身的内容。从组成上看，JWT包含3个部分：头部Header，数据Payload，签名Signature

1、Payload

Payload只是一个普通的Javascript 对象。对于payload的内容，JWT是没有任何限制的，但由于JWT是无加密的特性，因此不能在Payload存放用户信息

2、Header

Payload的内容在接收者端是通过签名(Signature)来校验的。不过存在多种类型的签名，因此，接收者需要知道使用的是哪种类型的签名。而携带token本身元数据的信息便存放一个独立的JSON对象，即JWT Header。

3、Signatures

JWT的最后一部分是签名，它也叫信息验证码MAC。签名只能由拥有Payload、Header和密钥的角色生成。签名生成的过程又称JWT认证流程，该流程是服务端通过浏览器上传的用户信息生成生成签名，再将其发送给浏览器。浏览器保存到本地，再以后每次发送请求时携带上token，而服务器只需验证token即可

# 系统分析

* 1. 可行性分析

可行性分析指，在工厂项目开发前，对项目本身的从技术、操作、经济等各个方面情况进行调研和评估，内容包括：必须性、合理性、风险程度。从而对项目的可行性有明确的认识。

* + 1. 技术可行性

在开发技术方面，本项目使用的主流的Java编程语言，java解决了c和c++的内存管理问题，同时拥有强大的第三方生态支持，在开发网页方面有无与伦比的优势，同时Java虚拟机技术保证了可移植性和安全性，支持继承，重载，多态等面向对象的特性便于网站维护。idea程序开发工具，便于系统开发

在数据库技术方面，本系统使用主流的MySql，由于 MySQL 数据库运行速率较快、代码开源、占据空间小，其有着广泛的应用在中小心型网站的开发中占据很大的市场

* + 1. 操作可行性

在使用人群方面，管理员一般要求掌握计算机技术，会使用基础的管理软件，熟悉IT产品，有丰富的管理系统使用经验。在学生和导师方面，现代互联网普及程度较高，高等院校人员对互联网软件操作已十分熟悉，只需要提供操作文档，即可快速上手使用

* 1. 系统需求分析

为了让系统能充分满足用户需求，在制作系统之前必须了解用户的需求，因此系统的需求作完整充足的分析，是制作成功的软件系统的必要前提。有利于把握软件开发的质量，也方便流程所设计人员之间的合作。

* + 1. 功能需求

为了更好的进行系统功能分析，应从系统用户的角度出发，分析用户的需求，了解用户使用方式和思维习惯，来帮助我们更好的分析系统设计。毕业论文管理系统有三个用户主体，分别为：管理员、学生、导师

1、管理员

(1)通知管理：为了让使用的学生和导师可以第一时间了解到毕业论文相关的通知信息，设计出通知管理模块来对通知进行统筹管理。首先只有管理员有权限发布通知，在通知发布通知页面，有通知标题和通知内容，输入后即可发表。并在查看通知页面查看已发表的通知。同时管理员还需要对已发表的通知的进行综合管理，所有需设计通知管理页面，对通知及删除、更改和模糊查询操作

(2)菜单管理：显示目前系统中各菜单的属性，方便管理员进行后期维护的初步修改

(3)权限管理：本系统中设计了三个不同的权限角色：管理员、学生、导师，每位角色都用不同的访问权限。为了实现管理员对角色的管理，故设计权限管理功能，在页面会显示三位角色所拥有的功能模块，管理员可以根据需求对角色的功能模块进行增减操作

(4)人员管理：管理员需要对使用该系统的人员信息进行统筹管理，这些信息包括学生、导师的个人简介中的信息，用户本身在系统中的信息。为了降低系统的耦合度，本系统将人员信息和用户信息作分别管理。分为人员信息模块和用户信息模块，在人员信息的保存了用户本人的信息：如姓名、学号、所在院系、职称等，而用户信息保存用户在系统的信息：如用户名和密码。管理员对这两个模块都能进行删除、更改、增加和模糊查询操作

(5)文件管理：本系统会保存学生发来的过程文档信息，因此管理员需要过程文档进行管理。过程文档管理包括对开题报告和论文正文的管理，页面中可对文件删除、模糊查询，还可下载文件

2、导师

(1)修改个人信息：显示个人信息，并且可以修改提交到后台

(2)课题管理:课题管理分为两个模块：上传课题和审核课题，指导老师在上传课题页面可以根据学生发送的任务书以及前期组会的信息选择适当的课题信息，输入课题信息后发布在本系统的学生页面选择课题中，学生可以根据自身情况选择合适的课题，这些信息会出现在审核课题模块中，指导老师课决定是否通过选题，若选题未通过，学生不能进行下一流程

(3)查看学生信息：显示导师指导的学生各项信息

(4)过程文档管理：过程文档管理分为两个模块:审核开题报告和审核毕业论文，学生提交的文档的会在该页面显示，导师需下载文档对文档进行评阅并填写评阅意见 ，再决定是否通过审核，毕论论文需对文档进行院内查重

(5)成绩管理：学生的毕业论文文档通过审核，方可进行成绩评定，该部分分为两个模块，添加成绩部分可以选择学生姓名、输入对应学生成绩进行成绩提交，在查看成绩模块的页面，指导老师可以看到学生的成绩信息。

3、学生：

(1)毕业选题：学生在需要看到自己导师发布的课题，因此需设计选择课题模块，在该 模块会显示导师发布的课题，学生可以对课题进行挑选。选择后的课题会出现在课题结果模块中，课题信息会显示该课题是否被导师通过。学生也可以根据需求在课题结果模块对课题进行退选操作

(2)过程文档管理：过程文档管理分为提交开题报告和提交毕业论文两个模块，两个模块页面相似，学生可以提交报告到系统，同时显示最新已提交报告的的信息和审核情况信息。

(3)修改个人信息：显示个人信息，并且可以修改提交到后台

* + 1. 安全性需求

建立合理的用户登录注册体系，完善的权限管理，对访问后台各个页面数据的请求药进行权限认证，保护数据库数据安全

* + 1. 用例图

用例图，是用户和与之相关的用例的来表示的用户与系统交互的最简明的心态，开发者可以与其他图表文档配合，以更高的层次观察整体系统的功能与细节，也是开发者相互交流的好形式，用例图如图3.1

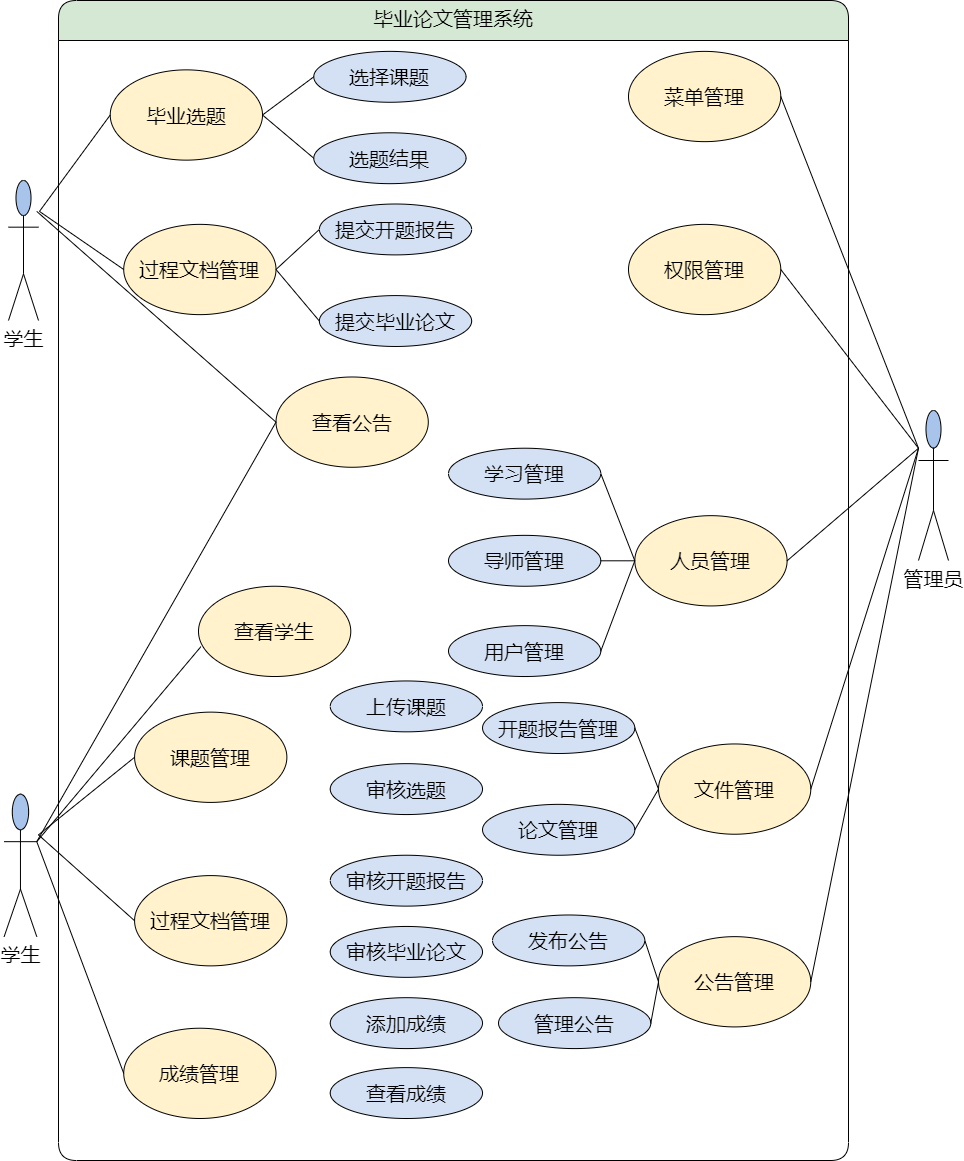


图3.1 用例图

* + 1. 时序图

1、用户、学生、导师管理操作

此项为管理员对用户、学生、导师管理操作，管理员首先进入学生管理、导师管理、用户管理页面，在页面中会显示学生、导师、用户的各项信息，页面上会有对应的增删改查管理按钮，通过这些按钮对信息进行交互会生成操作信息,然后将操作信息提交到数据库。用户、学生、导师管理时序图如图3.2所示

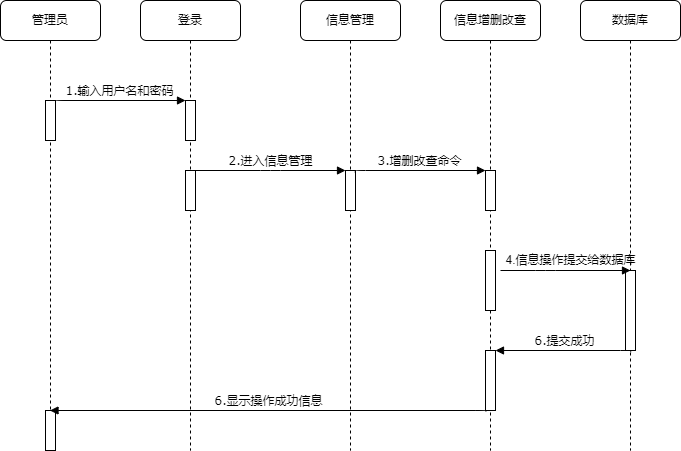


图3.2 用户、学生、导师管理时序图

2、学生选择课题流程操作

学生首先进入毕业选题页面，在毕业选题页面会显示当前导师发布的课题内容，点击选题操作后，系统会将选题信息提交到数据库。学生选择课题时序图如图3.3所示

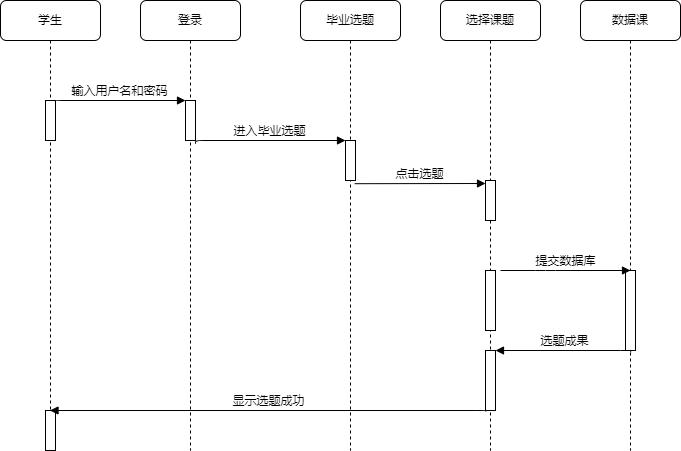


图3.3学生选择课题时序图

3、导师查重论文

导师首先进入论文审核页面，在论文审核页面会显示当前学生已提交的论文的信息，点击所需查重的论文，提交到数据库进行比较。导师查重时序图如图3.4所示

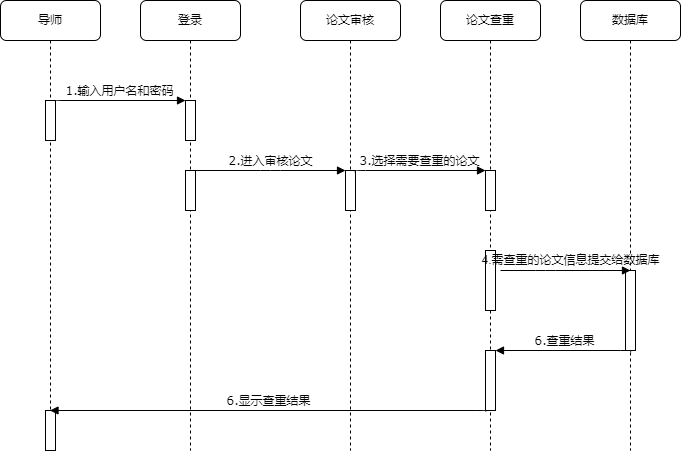


图3.4导师查重时序图

4、导师审核论文

导师首先进入论文审核页面，在论文审核页面会显示当前学生已提交的论文的信息，对页面上审核按钮进行交互会生成审核操作信息，这些审核操作信息会提交到数据库，并返回提交信息给前端，导师审核时序图如图3.5所示

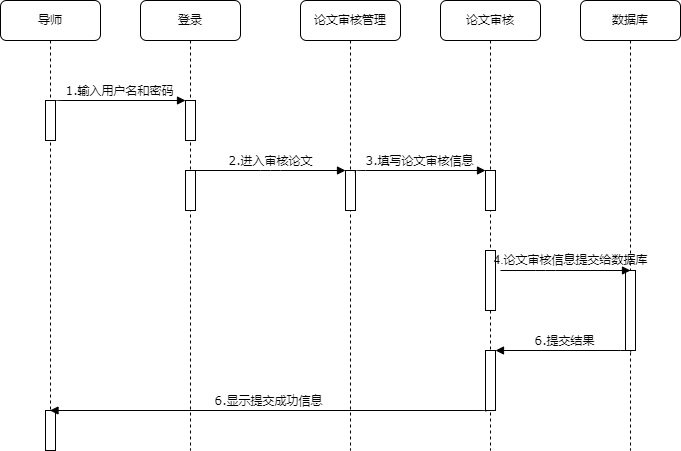


图3.5导师审核时序图

5、学生提交论文

学生首先进入过程文档管理页面，选择将要上传的论文，然后回提交到数据库，数据库返回提交成功，并且将当前论文的信息显示在过程文档管理页面上，学生提交论文时序图如图3.5所示

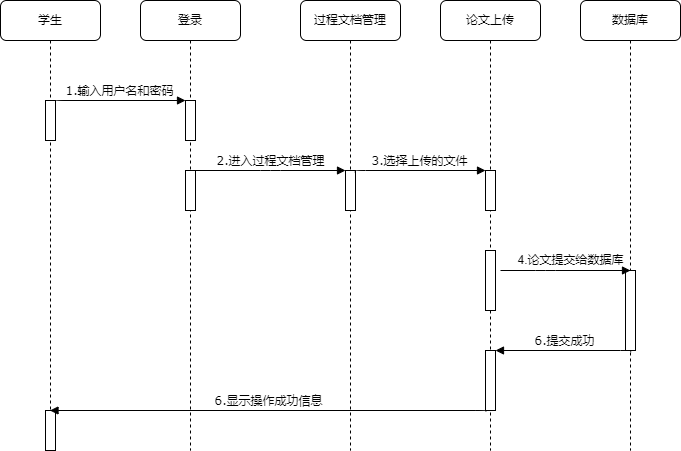


图3.5学生提交论文时序图

6、导师添加成绩

在成绩管理中的添加成绩模块，导师可以填写相应的论文成绩，通过与提交按钮进行页面交互将提交成绩信息发送数据库中，导师添加成绩时序图如图3.6所示

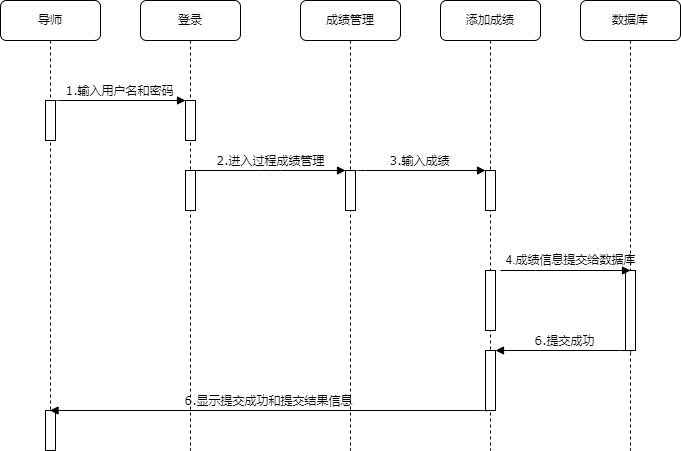


图3.6导师添加成绩时序图

# 数据库设计

* 1. E-R图设计

要对真实的概念模式进行建模，就必须对所设计的表进行识别成实体图，并对其进行分析，而表示这些实体图之间关系的图即为E-R图，总体E-R图如图5.11所示

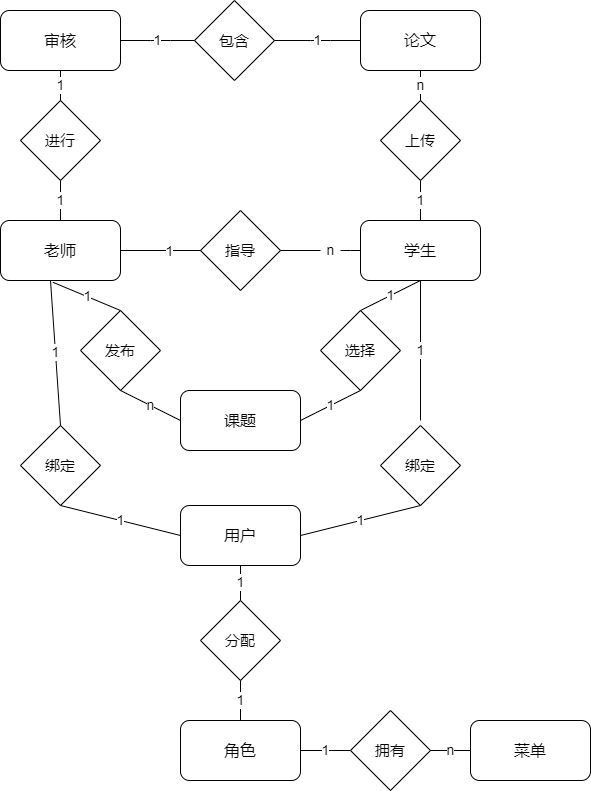


图5.11 总体E-R图

1. 用户E-R图如图5.12所示。

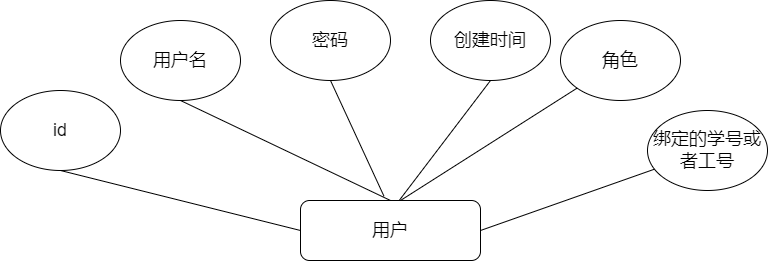


图5.12 用户E-R图

1. 学生E-R图如图5.13所示。

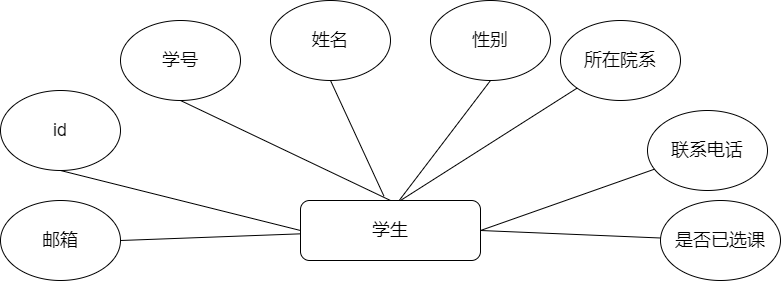


图5.13学生E-R图

1. 导师E-R图如图5.14所示。

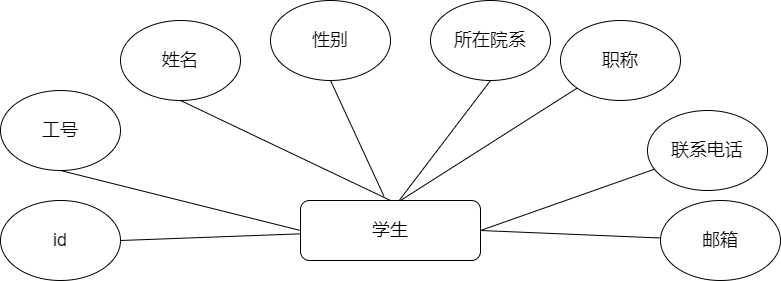


图5.14 导师E-R图

1. 课题E-R图如图5.15所示。

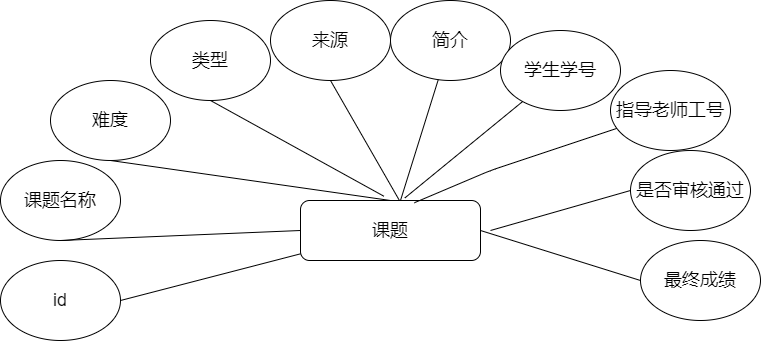


图5.15 课题E-R图

1. 审核E-R图如图5.16所示。

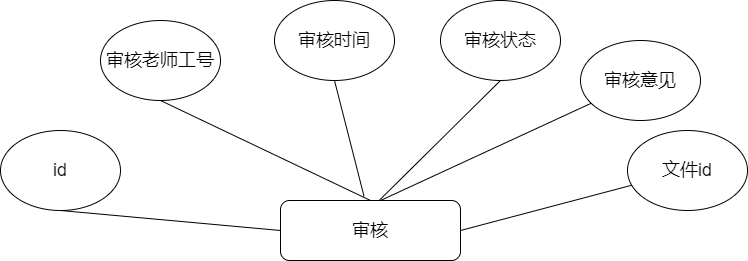


图5.16 审核E-R图

1. 角色E-R图如图5.17所示。

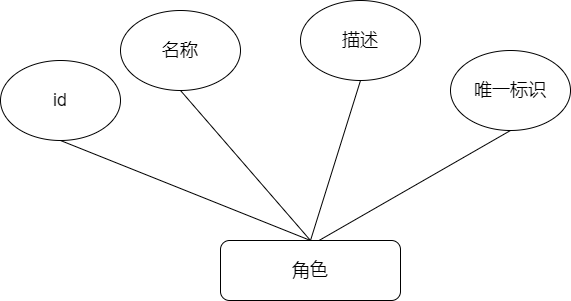


图5.17 角色E-R图

1. 论文E-R图如图5.17所示。

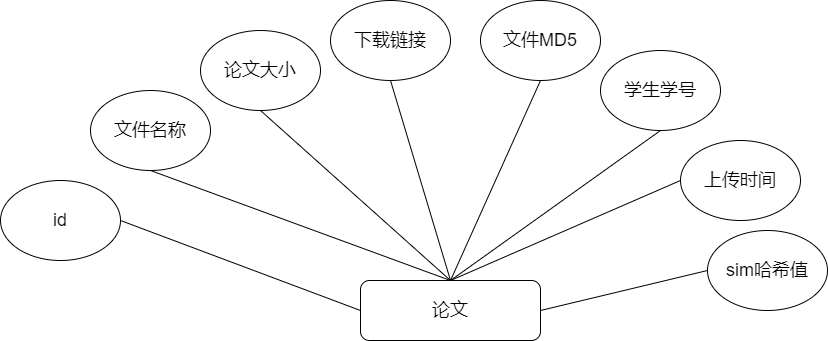


图5.18 论文E-R图

1. 公告E-R图如图5.17所示

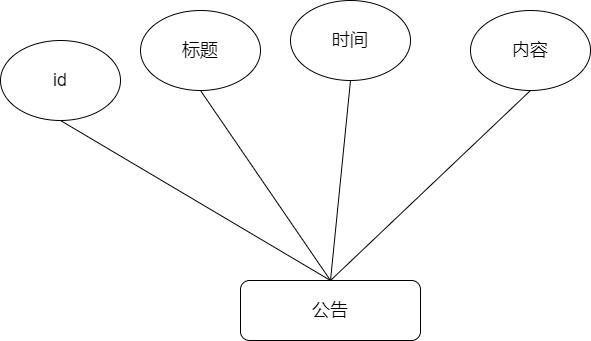


图5.19 公告E-R图

1. 菜单E-R图如图5.17所示。

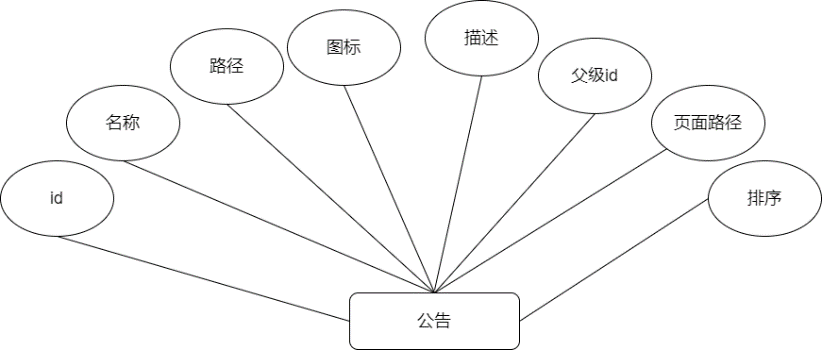


图5.20 菜单E-R图

* 1. 数据字典

1、用户表（hai\_sys\_user\_tao）

**表5.1 用户表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项目名称 | 数据类型 | 长度(kb) | 描述 |
| Hai0id\_tao | int | 0 | id |
| Hai\_username\_tao | varchar | 50 | 用户名 |
| Hai\_password\_tao | varchar | 50 | 密码 |
| Hai\_create\_time\_tao | timestamp | 0 | 创建时间 |
| Hai\_role\_tao | varchar | 50 | 角色 |
| Hai\_st\_num\_tao | varchar | 10 | 绑定的学号或者工号 |

1. 导师表（hai\_sys\_teacher\_tao）

**表5.2 导师表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项目名称 | 数据类型 | 长度(kb) | 描述 |
| hai\_id\_tao | int | 0 | 导师id |
| hai\_tno\_tao | varchar | 30 | 导师工号 |
| hai\_name\_tao | varchar | 30 | 导师姓名 |
| hai\_sex\_tao | varchar | 2 | 导师性别 |
| hai\_dept\_tao | varchar | 30 | 导所在院系 |
| hai\_title\_tao | varchar | 10 | 导师职称 |
| hai\_phone\_tao | varchar | 30 | 导师联系电话 |
| hai\_email\_tao | varchar | 30 | 导师邮箱 |

1. 学生表（hai\_sys\_student\_tao）

**表5.3 学生表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项目名称 | 数据类型 | 长度(kb) | 描述 |
| hai\_id\_tao | int | 0 | 学生id |
| hai\_sno\_tao | varchar | 20 | 学生学号 |
| hai\_name\_tao | varchar | 30 | 学生姓名 |
| hai\_sex\_tao | varchar | 3 | 学生性别 |
| hai\_dept\_tao | varchar | 20 | 学生所在院系 |
| hai\_phone\_tao | varchar | 20 | 学生联系电话 |
| hai\_email\_tao | varchar | 30 | 学生邮箱 |
| hai\_is\_select\_tao | tinyint | 1 | 是否已选课 |

1. 课题表（sys\_titile\_tao）

**表5.4 课题表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项目名称 | 数据类型 | 长度(kb) | 描述 |
| hai\_id\_tao | int | 0 | 课题id |
| hai\_name\_tao | varchar | 30 | 课题名称 |
| hai\_difficulty\_tao | varchar | 10 | 课题难度 |
| hai\_type\_tao | varchar | 10 | 课题类型 |
| hai\_source\_tao | varchar | 20 | 课题来源 |
| hai\_brief\_tao | varchar | 1000 | 课题简介 |
| hai\_stu\_num\_tao | varchar | 20 | 课题学生id |
| hai\_tea\_num\_tao | varchar | 20 | 课题指导老师id |
| hai\_is\_verify\_tao | tinyint | 1 | 是否审核通过 |
| hai\_score\_tao | float | 0 | 最终成绩 |

5、审核表（hai\_sys\_verify\_tao）

**表5.5 审核表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项目名称 | 数据类型 | 长度(kb) | 描述 |
| hai\_id\_tao | int | 0 | 审核表id |
| hai\_tea\_num\_tao | varchar | 30 | 审核老师工号 |
| hai\_time\_tao | datetime | 0 | 审核时间 |
| hai\_state\_tao | tinyint | 1 | 审核状态 |
| hai\_comment\_tao | varchar | 500 | 审核意见 |
| hai\_file\_id\_tao | int | 0 | 文件id |

1. 论文表（hai\_sys\_paper\_tao）

**表5.6 论文表**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项目名称 | 数据类型 | 长度(kb) | 描述 |
| hai\_id\_tao | int | 0 | 文件id |
| hai\_name\_tao | varchar | 30 | 文件名称 |
| hai\_size\_tao | bigint | 0 | 论文大小 |
| hai\_url\_tao | varchar | 255 | 下载链接 |
| hai\_md5\_tao | varchar | 255 | 文件MD5 |
| hai\_stu\_num\_tao | varchar | 20 | 学生学号 |
| hai\_create\_time\_tao | datetime | 0 | 上传时间 |
| hai\_sim\_hash\_tao | varchar | 255 | sim哈希值 |

# 论文查重算法设计

为了提高毕业论文质量水平，增强高等院校对论文质量的把控能力。本系统设计了论文查重功能，通过比对学校自由的数据库，来检查学生是否有抄袭往届学生的论文，这项功能是利用现有的查重算法，对两篇文本进行相似度检测，最终通过数字化的数据，判断两个文本的内容是否雷同[3]

* 1. Simhash算法

根据文献[1]文本相似度主要采用simhas算法、分词算法并结合海明距离计算进行实现。simhas算法是由谷歌的Charikar在其论文“detecting near-duplicates for web crawling”提出的，其的主要思想是降维，将高维的特征向量映射成低维向量，通过计算比较两个低维向量的海明距离来判断两个文本是否相似，在判断的时候，因为已对文章进行降维，所在在比较的时候无需处理大量文本数据，同时较为算法本身简易，对于初步的文本比较行为较为友好，因此simhash算法在文本相似度检测领域有着广泛的应用[2]

* + 1. 计算simhash值

1、分词

对于一端文本，首先要进行分词，即将文本分成数个有实际意义的词语作为特征向量，然后根据该词语在文本中出现的次数为其赋权，一般将权重分为5个级别

2、计算hash值

将分词得到的特征向量通过hash函数计算成hash值，这些hash值的表达形式为二进制，本系统中为64位的二进制数

3、加权

在hash值的基础上，为文本生成的所有特征向量进行加权操作，操作为：若某一位数为1则该位数置为正权值，若某一位数为0则改位数置为负权值。

4、合并

累加所有经过加权的特征向量，最终变成单独的序列串

5、降维

对于累加后的序列传的进行如下操作，若某位数比0大，则其值为1，若某位数比0小，则其值为0，最终遍历完所有的位数，实现降维操作

* + 1. 计算汉明距离

在信息论中，汉明距离是描述两个n长码字 = (…)，…)之间的距离

其中，表示模2加法，，.

D{x,y}表示两码字在相同位置上不同码符号的数目的总和，它能够反映两码字之间的差异，进而提供码字之间的相似程度的客观依据[12] 。根据可查询的资料显示，对于64位的 simhash值，汉明距离在3以内的可认为相似度比较高。

# 系统设计

* 1. 系统概要设计
     1. 系统框架图

MVC 的缩写是模型(model)、视图(view)、控制器(controller)。作为交互模式，其主要的特点为：数据可视化、UI处理逻辑和业务逻辑

1、Model（模型）： 模型是前后端定义的数据传输单元，通常前端只是需要后端提供一些数据来展示而已，比如用户名、处理结果来，需要HTML输出的东西；

2、View（视图）：视图代表模型包含的数据的可视化，在视图层显示管理员、用户、导师的前端页面

3、Controller（控制器）：其实就是业务逻辑，类似于之前的Servlet开发，区别在于现在将处理结果都放在Model里，然后发给View；

系统框架图如图4.1所示

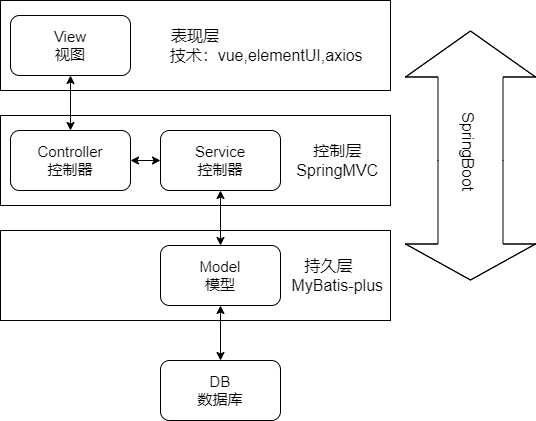


图4.1系统框架图

* + 1. 系统功能模块图

根据需求分析，毕业论文管理系统总体分为管理员、学生、导师三个用户，三个用户分别对应了三个功能模块

1. 管理员模块

管理员模块主要对系统总体进行管理，包括权限管理：对分配用户角色可使用的菜单信息进行分配；人员管理：显示学生、导师信息，并对其进行增删改查操作，用户管理：显示用户信息，并对其进行增删改查操作；文件管理：对开题报告、论文进行删改查操作；公告管理包括发布公告和修改公告操作，管理员模块图如图4.2

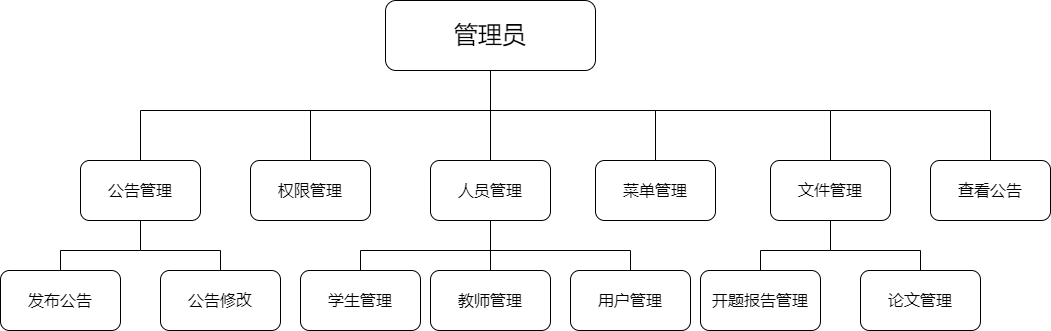


图4.2管理员模块图

1. 导师模块

导师模块包括课题管理：上传课题、审核课题；过程文档管理：审核开题报告、审核毕业论文，其中审核毕业论文要进行校内查重；成绩管理：通过学生的姓名可选择性的进行成绩的添加操作，在查看成绩页面，显示已添加的成绩信息，根据需要导师可对成绩信息进行更改；导师模块图如图4.3

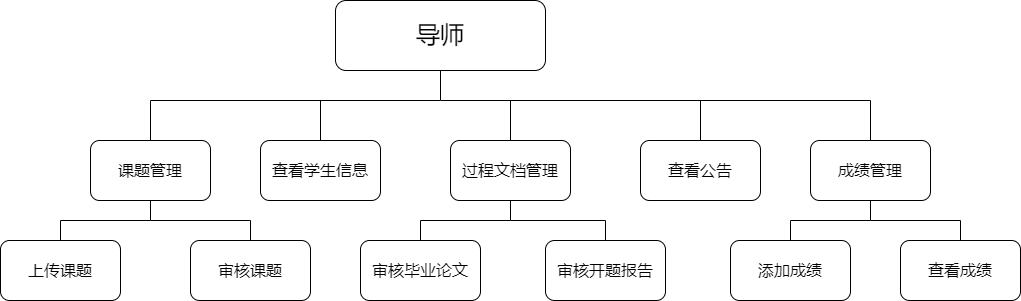


图4.3导师模块图

1. 学生模块

学生模块包括毕业选题：选择课题、课题结果，如果已选择课题，则不能再选择，再课题结果模块可以退选课题；过程文档管理：提交毕业论文、提交开题报告，在提交文档页面显示当前最新提交文档的信息和文档审核的信息；查看公告，学生模块图如图4.4

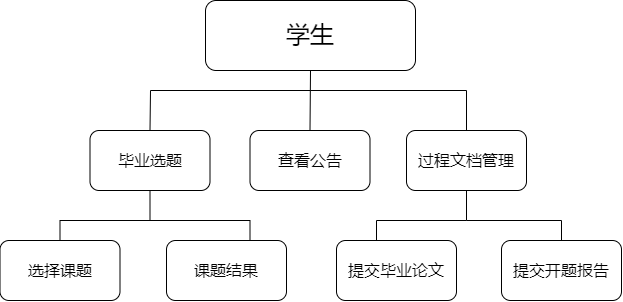


图4.4学生模块图

* + 1. 类模块设计

1. 用户实现类

用户服务执行类继承用户服务接口，其中login方法实现登录功能；register方法实现注册功能；updatePassword方法实现更新密码功能；setStudentTeacher方法实现设置学生老师表；getUserInfo方法实现获取当前用户功能，getRoleMenus方法获取当前角色的菜单列表。

1. 导师实现类

老师服务执行类继承老师接口，其中getTeaPage方法实现分页获取导师信息，同时可以实现模糊查询， checkPaper方法实现导师论文查重功能

1. 学生实现类

学生服务执行类继承学生服务接口，其中getStuPage实现分页获取学生信息，同时可以实现模糊查询，GetStuFileInfo实现根据导师获取指导的学生信息，

1. 课题实现类

课题服务执行类继承课题服务接口，其中pageTitle方法实现分页获取课题信息，getResult方法获取学生课题成绩，Withdrawal方法实现改选课题

1. 角色实现类

角色服务执行类继承角色服务接口，其中setRoleMenu方法实现设置角色菜单功能；getRoleMenu方法获取当前角色菜单

1. 菜单实现类

菜单服务执行类继承菜单服务接口，其中findMenus方法获取当前数据库所以菜单，并按照一级二级菜单形式进行存储。

1. 论文实现类

论文服务执行类继承论文服务接口，其中showPaperInfo方法获取学生上传的论文信息；uploadPaper方法实现论文上传

* 1. 系统详细设计
     1. 前端设计与实现

1、Vue路由功能实现

为了将跳转路径和页面中的组件进行一一对应，使用Vue路由来对访问的路径进行配置以manage.vue（如图）为管理系统主体框架，实现路由组件切换。根据用户的角色权限获取的菜单数据，对组件Mange.vue实现路由动态拼接，以实现用户权限管理控制

2、axios实现ajax请求

使用axios将request请求封装到request.js文件中，并且设置好request 拦截器,对请求发送前进行统一的数据处理，比如统一加上token。设置好response 拦截器，对服务器返回的数据作统一处理。

* + 1. 后端设计与实现

1、管理员登录功能实现

用户在浏览器输入用户登录信息传递到服务器，用户服务执行类将浏览器传递来的信息进行合法性验证，只有验证通过，才能确认用户合法性，之后通过TokenUtils类中的getToken方法生成token签名，同时将该用户的菜单信息传递到浏览器。在request.js文件中将token设置到请求头中，这样每次访问的服务端接口的时候都会发送token，在服务器端设只JWT拦截器，在除登录、注册等页面请求访问时，需要解析token 中的用户信息用来确认该用户的身份。

2、管理员注册功能实现

前端发起两次post请求，第一次调用用户服务执行中register方法，将用户名和密码存储到数据库中，第二次调用导师控制类或者学生控制类中save方法，将个人信息保存在数据库相应的表中

3、管理员人员管理功能实现

(1)学生管理功能

管理员删除学生信息，在后端学生控制类中调用delete方法;修改学生信息，在后端学生控制类中调用save方法;模糊查询和分页查询查找，需要经学生控制类，学生服务执行类调用学生Mapper中getStuPage方法查询数据; 设置学生的指导老师，调用的是用户服务执行类中setStudentTeacher方法向学生设置指导老师

(2)导师管理功能

管理员删除导师信息，在后端导师控制类中调用delete方法;修改导师信息，在后端导师控制类中调用save方法;模糊查询和分页查询查找，需要经导师控制类，导师服务执行类调用学生Mapper中getTeaPage方法，最终通过导师Mapper中的sql语句向数据库查询数据;

(3)用户管理功能

管理员增加用户可分别按照三个角色添加：管理员、学生、导师，代码逻辑和注册功能一样，删除用户信息，在后端用户控制类中调用delete方法;修改导师信息，在后端用户控制类中调用save方法;模糊查询和分页查询查找，在后端用户控制类中调用findPage方法

4、管理员权限功能实现

权限管理页面的增删改查与上述管理页面逻辑类似。管理员对角色菜单重新设置，在前端，将角色Id，与菜单Id传递到后端角色服务执行类中，然后先删除当前角色id所有的绑定关系，再把前端传过来的菜单id数组绑定到当前的这个角色id上去，由于菜单中存在父级菜单和二级菜单，则需要判断菜单是否为二级菜单，若传过来菜单Id数组里面没有它的父级id，则需要补充这个父级id。

5、学生选择课题功能实现

(1)选择课题功能

学生选择课题提交后，将学生的学号和所选课题的id提交到后端课题服务执行类中，先由KeithMapper中updateById方法更新数据库信息，再由学生Mapper中setIsSelect方法更新学生表中is\_select字段信息

(2)课题结果功能

课题结果页面首先将后台的课题信息显示再前端，需要在课题Mapper中对导师表和学生表进行联表查询。课题退选功能是将要退选的课题id和学生学号传递到课题服务类，然后在课题服务类类中由学生Mapper设置学生表中中is\_select字段信息，再由课题Mapper中WithdrawalTitle更改课题表学生学号字段信息。

6、学生过程文档管理功能实现

(1)文件上传功能

学生上传文件要将文档信息和当前学生学号传递到后端的论文服务执行类中首先先判断学生是否选择课题，若有则再判断课题是否通过审核，若有则进入上传文档逻辑:首先初始化文件唯一的标识码，然后判断配置的文件目录是否存在，若不存在则创建一个新的文件目录，生成文件的md5，查询数据库是否存在相同的记录，若无将文件上传到磁盘，并生成新的url作为下载的链接，最后将上述的信息存储至数据库。

(2)显示最新论文信息和审核信息

论文信息是通过论文服务执行类中showPaperInfo方法查询，先根据前端传来的学生学号查询数据库中所有的论文，再根据上传时间排序，提取时间最新的一条新整理发送到前端，前端再根据最新的论文id为参数发送get请求请求审核信息

7、导师过程文档管理功能实现

(1)显示文档信息功能

显示文档信息功能先将改导师的导师工号作为参数,进行get请求，后端在导师服务执行类，对请求文档类型进行判断，最后在mapper中对数据库进行查询操作。

(2)论文查重功能

在论文上传时使用getSimHash方法计算每个文档的simHash值，并存储到数据库，导师发送查重请求，使用getHammingDistance方法计算海明距离，将数据处理传递到前端,simhash算法主要代码如下：

1. 分词：

List<String>keywordList = HanLP.extractKeyword(str, str.length());

1. 获取hash值

String keywordHash = getHash(keyword);

if (keywordHash.length() < 128) {

nt dif = 128 - keywordHash.length();

for (int j = 0; j < dif; j++) {

keywordHash += "0";

}

}

1. 加权、合并

for (int j = 0; j < v.length; j++) {

if (keywordHash.charAt(j) == '1'){

v[j] += (10 - (i / (size / 10)));

} else {

v[j] -= (10 - (i / (size / 10)));

}

}

1. 降维

String simHash = "";

for (int j = 0; j < v.length; j++) {

if (v[j] <= 0) {

simHash += "0";

} else {

simHash += "1";

}

}

1. 计算汉明距离

int distance = 0;

if (simHash1.length() != simHash2.length()) {

distance = -1;

} else {

for (int i = 0; i < simHash1.length(); i++) {

if (simHash1.charAt(i) != simHash2.charAt(i)) {

distance++;

}

# 系统测试

* 1. 测试用例
     1. 登录、注册模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 登录测试 | 1. 入用户名和密码  2. 登录按钮 | 用户名：11111  密码：123 | 登录成功 |
| 学生注册测试 | 1. 点击学生注册按钮 2. 输入注册信息 3. 点击注册按钮 | 用户名：88888  密码：123  姓名：李四  学号: 311808 | 学生注册成功 |
| 导师注册测试 | 1. 点击导师注册按钮  2. 输入注册信息  3. 点击注册按钮 | 用户名：77777  密码：123  姓名：吴三  学号: 107 | 导师注册成功 |

* + 1. 用户管理模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 新增管理员用户 | 1. 点击新增用户按钮 2. 输入管理员用户信息 3. 点击确定按钮 | 用户名：admin2  密码：admin2 | 添加管理员用户成功 |
| 新增学生用户 | 1. 增学生按钮  2. 输入学生用户信息 | 用户名：xuesheng  密码：123  姓名：黄八  学号：311810 | 添加学生用户成功 |
| 新增导师用户 | 1. 点击新增学生按钮  2. 输入导师用户信息  3. 确定按钮 | 用户名：daoshi  密码：123  姓名：陈五  工号：113 | 添加导师用户成功 |
| 删除用户 | 1. 点击删除按钮 2. 点击确定 | — | 删除用户成功 |
| 编辑用户 | 1. 点击编辑用户按钮 2. 修改用户信息   点击确定 | 密码：  admin2 ->admin3 | 编辑用户成功 |

* + 1. 学生管理模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 编辑学生信息 | 1. 点击编辑学生按钮 2. 修改学生信息，点击确定 | 联系电话：  2222 -> 1780000 | 修改成功 |
| 删除学生信息 | 1. 点击删除按钮 2. 点击确定 | — | 删除成功 |
| 选择指导老师 | 1. 点击选择指导老师按钮 2. 选指导老师 3. 点击确定 | — | 设置成功 |

* + 1. 导师课题管理模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 上传课题 | 1. 输入课题信息 2. 点击发布课题 | 课题名称：基于卷积层的人脸识别系统  课题难度:偏难  课题类型：计算机软件研发类  课题来源：自拟项目  课题摘要：这是…. | 添加成功 |
| 审核选题 | 1. 点击审核通过按钮 2. 确认审核 | — | 审核成功 |

* + 1. 导师审核模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 查重论文 | 点击查重按钮 | — | 1. 显示查重通过 2. 显示查重结果 |
| 审核论文 | 1. 点击审核按钮 2. 输入审核信息 | 部通过  审核意见：查重率过高 | 审核通过 |

* + 1. 导师成绩管理模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 添加成绩 | 1. 选择学生 2. 输入成绩信息 3. 点击添加按钮 | 成绩：80 | 添加成功 |
| 修改成绩 | 1. 进入查看成页面 2. 点击修改按钮 3. 输入修改的数据 | 成绩：  80->70 | 修改成功 |

* + 1. 学生毕业选题模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 选择课题 | 1. 点击选择按钮 2. 确认选题 | — | 选题成功 |
| 退选课题 | 1. 点击退选 2. 确认退选 | — | 退选成功 |

* + 1. 学生提交文档模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试标题 | 测试步骤 | 测试数据 | 预期结果 |
| 上传论文 | 1. 点击上传按钮 2. 选择上传文件 | 文件：本科毕业论文.txt | 上传成功 |

* 1. 测试结果
     1. 登录、注册模块测试结果

1. 登录测试结果如图8.1、图8.2所示

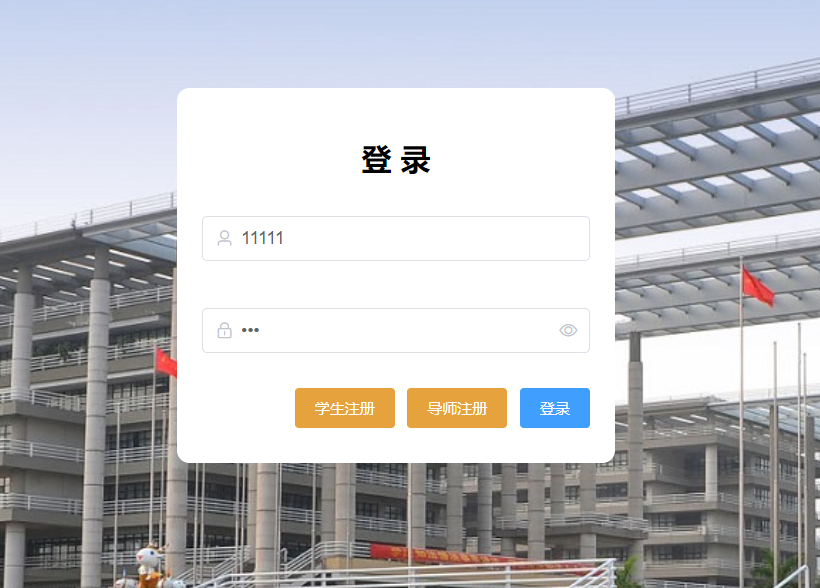


图8.1 输入登录数据

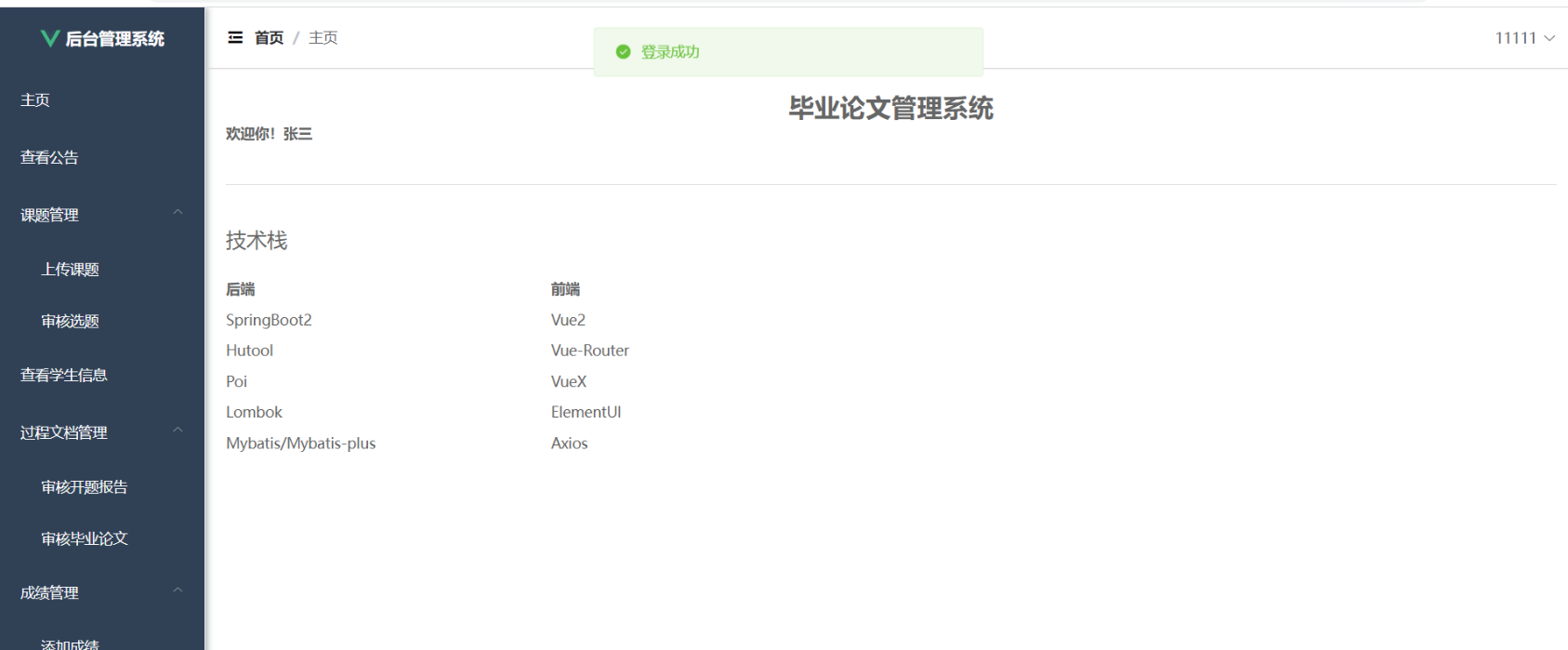


图8.2 显示登录成功

1. 注册测试结果如图8.3、图8.4所示



图8.3 输入注册数据



图8.4 显示注册成功

* + 1. 用户管理模块测试结果
  1. 新增学生用户如图8.7、图8.8所示

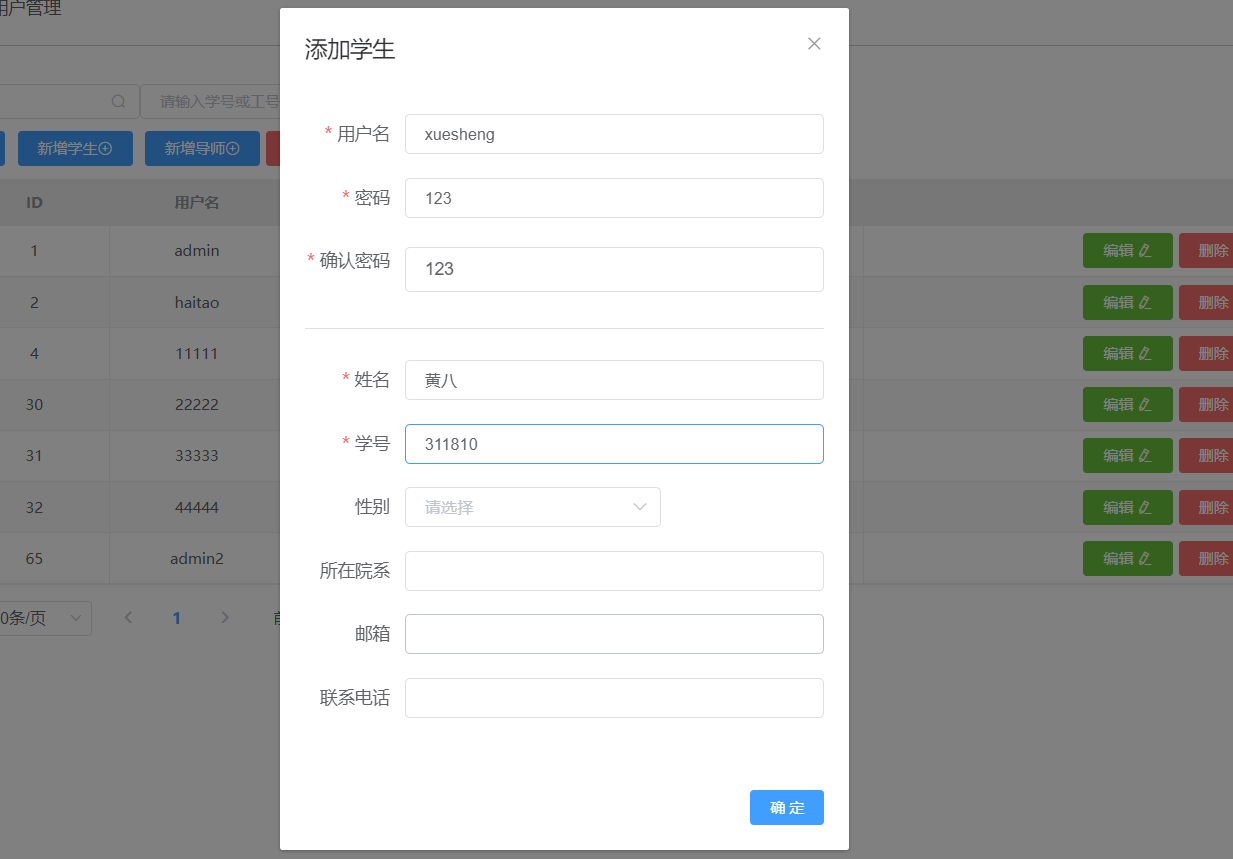


图8.7输入添加学生用户数据



图8.8显示添加结果

* 1. 删除用户如图8.11、图8.12所示



图8.11点击删除

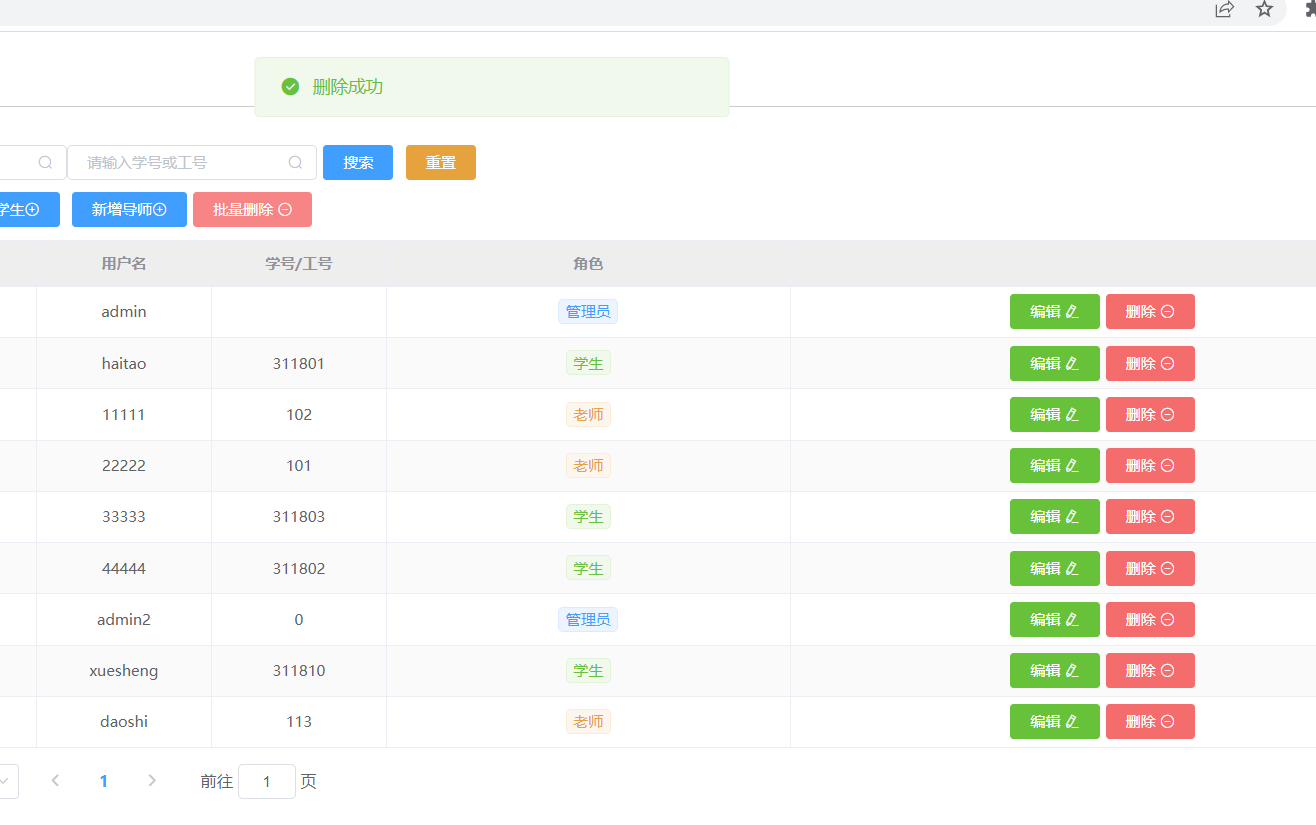


图8.12显示删除结果

* 1. 用户如图8.13、图8.14所示



图8.13输入编辑用户信息

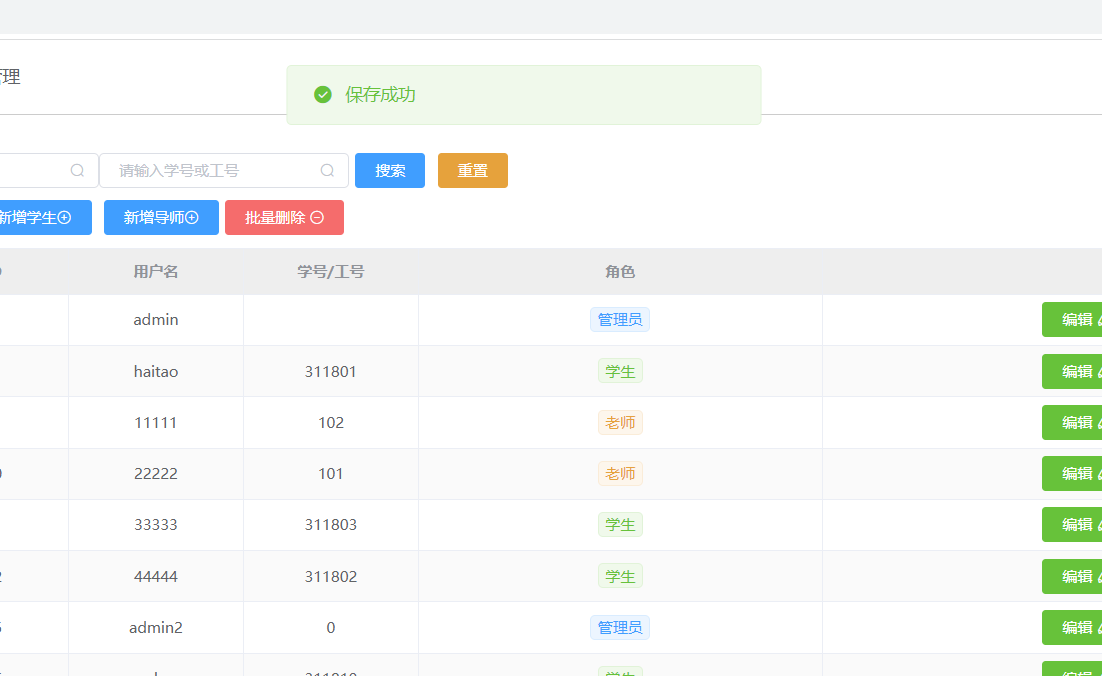


图8.14编辑保存成功

* + 1. 学生管理模块测试结果
  1. 编辑学生信息如图8.15、图8.16所示



图8.15输入编辑学生信息



图8.16显示编辑成功

* 1. 删除学生信息如图8.17、图8.18所示

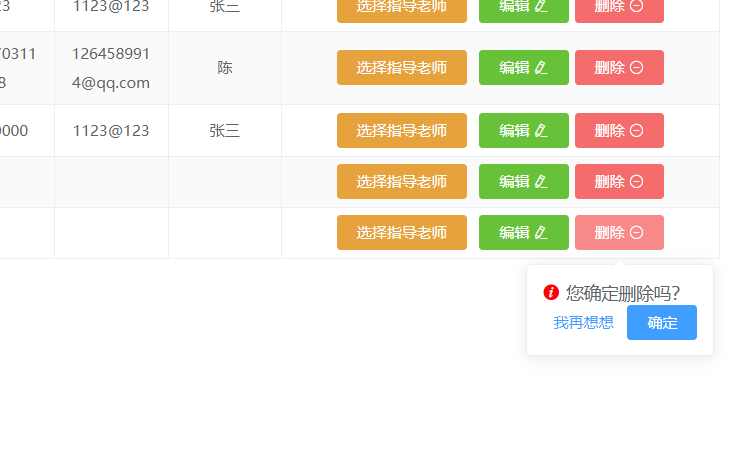


图8.17点击删除学生



图8.18删除成功

3、选择指导老师如图8.19、图8.20所示

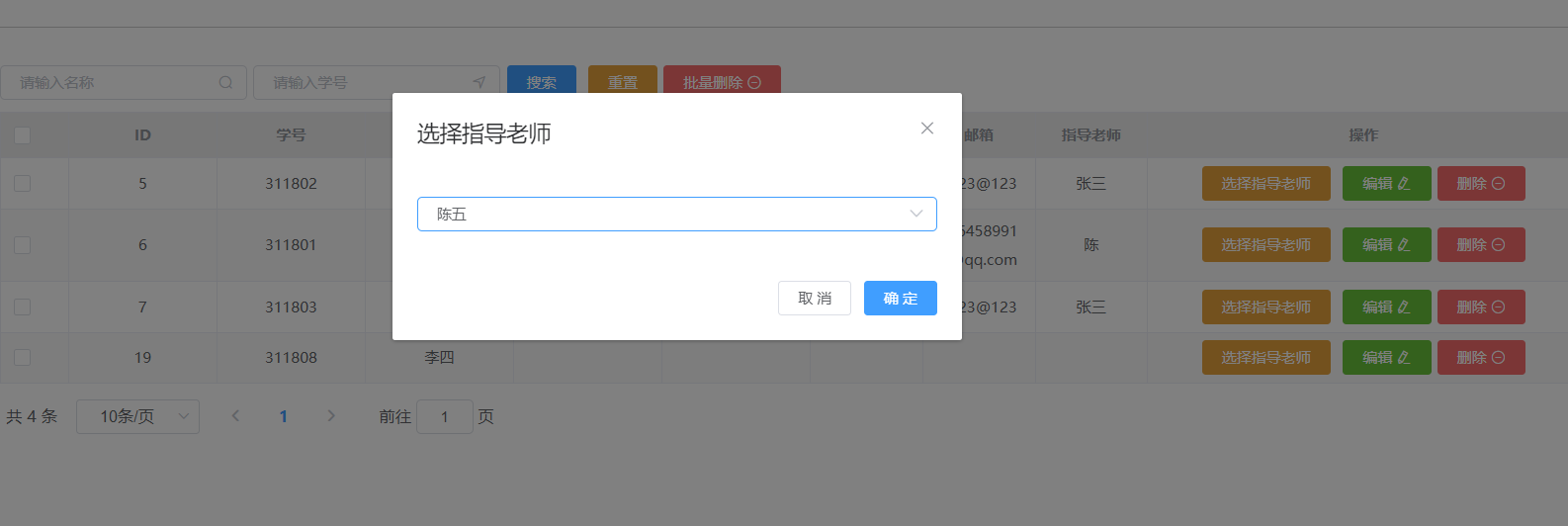


图8.19设置指导老师页面



图8.20显示设置成功

* + 1. 导师课题管理模块测试结果

1. 上传课题如图8.21所示



图8.21输入课题信息

1. 审核选题如图8.23、图8.24所示



图8.23点击审核中通过



图8.24显示审核结果

* + 1. 导师审核模块测试结果
  1. 查重论文如图8.25所示



图8.25查重结果页面

* 1. 审核论文如图8.26、图8.27所示



图8.26输入查重信息

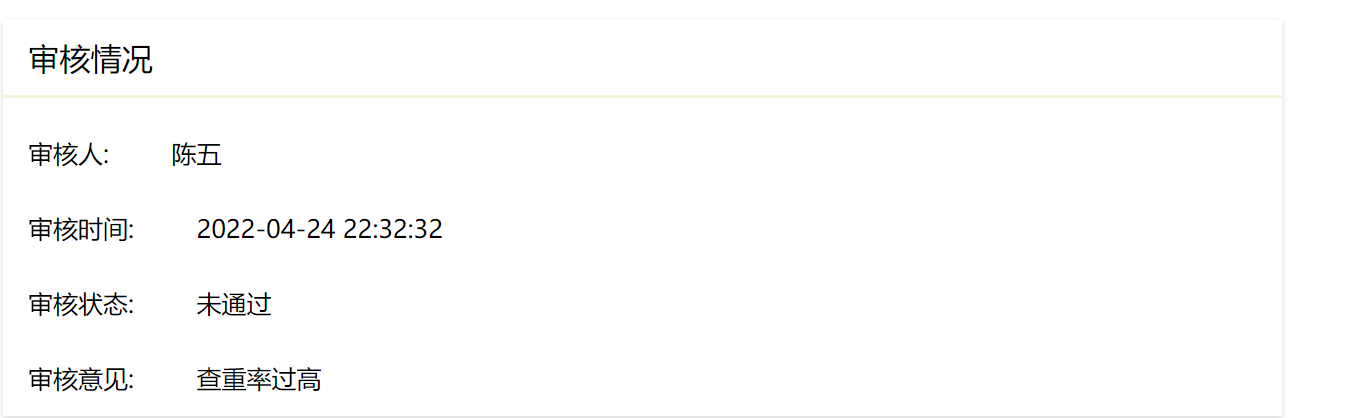


图8.27查重结果

* + 1. 导师成绩管理模块测试结果
  1. 添加成绩如图8.28、图8.29所示



图8.28输入成绩

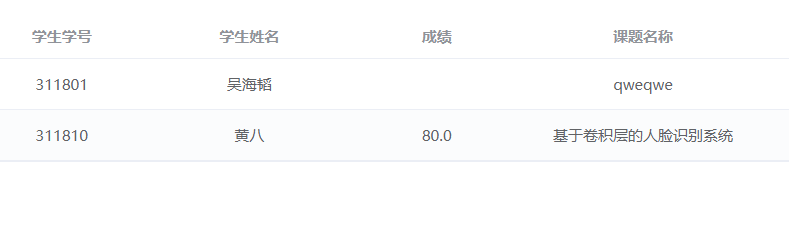


图8.29添加结果

* 1. 修改成绩如图8.30、图8.31所示



图8.30输入修改成绩



图8.31显示修改成功

* + 1. 学生毕业选题模块测试结果

1. 选择课题如图8.32、图8.33所示

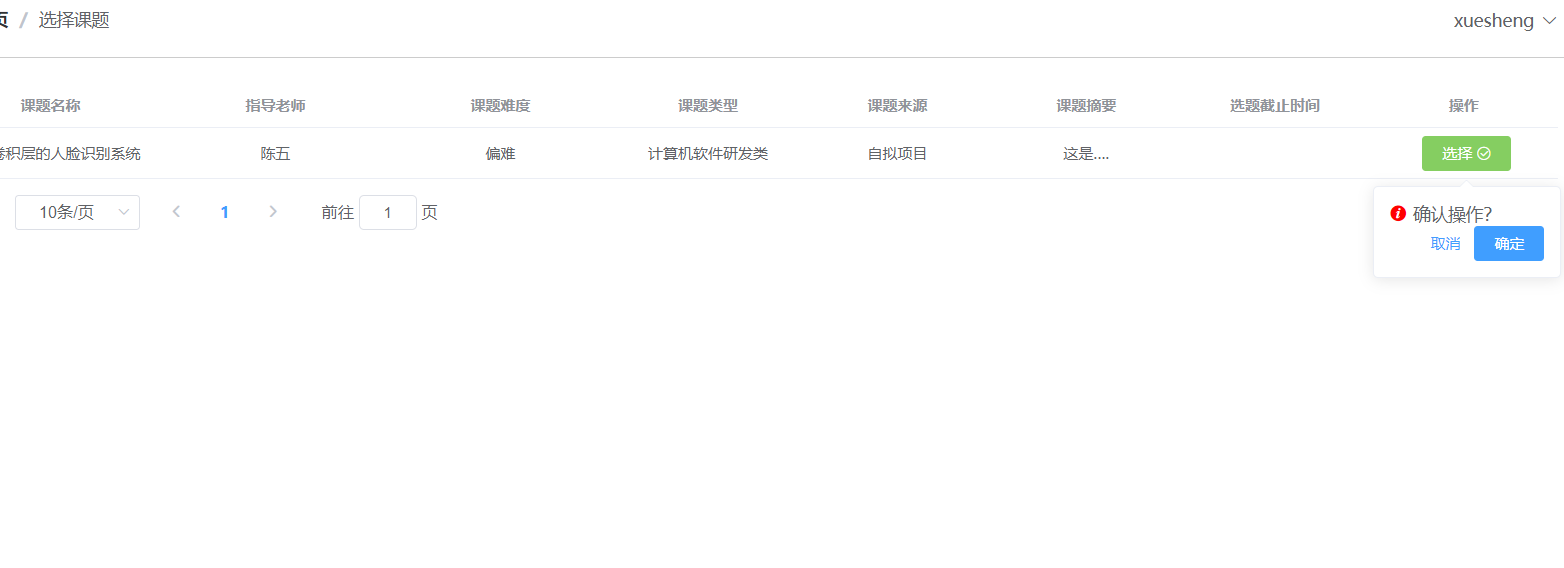


图8.32点击选题



图8.33显示选题成功

1. 退选课题如图8.34、图8.35所示



图8.34点击退选 图8.35显示退选成功

* + 1. 学生提交文档模块测试结果

学生上传文档如图8.36、图8.37所示

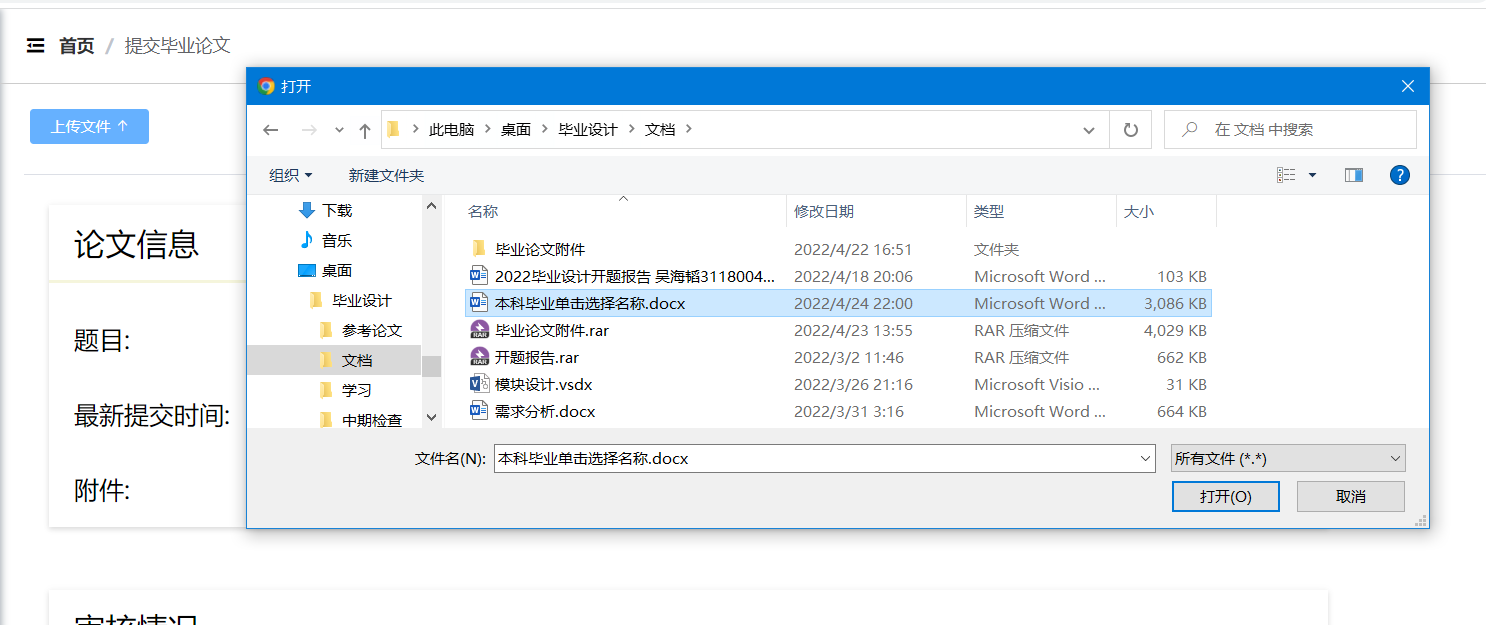


图8.36选择文件上传

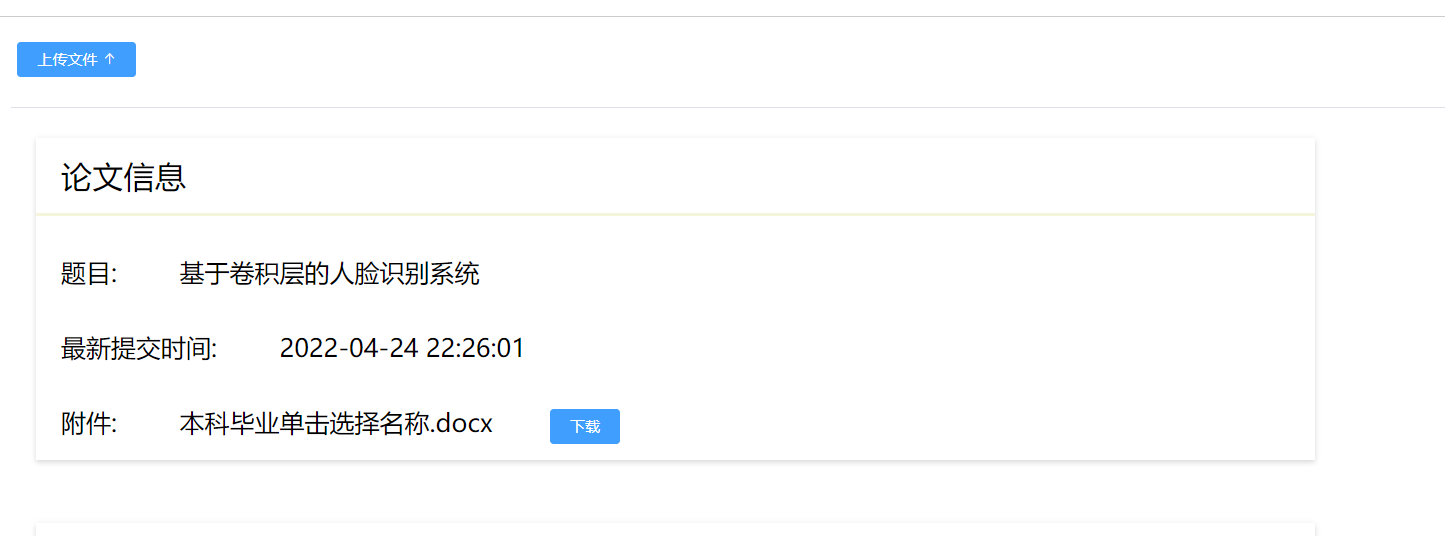


图8.37文件上传结果

# 结论

本文论述了使用SpringBoot和Vue框架结合simhash算法制作出具有查重功能的毕业论文管理系统。本系统具有较强的可维护性和可移植性，提供较为简洁和优美的用户页面，缩短了高校管理员处理毕业论文的时长，减轻了教师指导毕业论文的压力，也为学生上传毕业论文，查看自身论文情况带来了便利。查重功能也能为高校做论文的初步审查，对提高毕业论文质量具有重要作用。

通过对基于SpringBoot的毕业论文管理系统的设计和开发工作，我完成了以下的具体工作：

1. 对SpirngBoot、mybatis、vue框架进行了的基本的学习
2. 掌握了Simhash算法，并使用该算法实现了文本相似度比较的查重功能
3. 对软件开发的理论知识进行了有效实践

同时系统在非需求功能，如性能需求、数据库安全性还有待改进，在查重算法方面存在计算准确率无法保证、运算效率低的问题。

参考文献

* + - 1. Sadowski C, Levin G. Simhash: Hash-based similarity detection[J]. 2007.

[2] Sood S, Loguinov D. Probabilistic near-duplicate detection using simhash[C]//Proceedings of the 20th ACM international conference on Information and knowledge management. 2011: 1117-1126.

[3]王宇,夏丽丽,何颖刚.基于SSM框架的高校毕业论文管理系统的设计与实现[J].电子技术与软件工程,2021(22):26-29

[4]张晨阳,段国云,文春生.基于Simhash算法的文本查重系统的设计与实现[J].湖南科技学院学报,2021,42(5):51-54

[5]彭双和,图尔贡.麦提萨比尔,周巧凤.基于Simhash的中文文本去重技术研究[J].计算机技术与发展,2017,27(11):137-140145

[6]胡明,文翠,黄川.基于SpringBoot的毕业论文管理系统设计和开发[J].电脑编程技巧与维护,2020(7):92-93104

[7]吕宇琛.SpringBoot框架在web应用开发中的探讨[J].科技创新导报,2018,15(8):168-168173

[8]焦鹏珲. 基于SpringBoot和Vue框架的电子招投标系统的设计与实现[D].南京大学,2018.

[9]边超.基于B/S架构高校毕业论文管理系统的设计——以中国矿业大学银川学院为例[J].企业科技与发展,2020(12):64-67.

[10]钟怡旸,郭昱君.基于Springboot的博客管理系统设计与实现[J].现代信息科技,2021,5(07):18-20+24.DOI:10.19850/j.cnki.2096-4706.2021.07.005.

[11]徐济惠.基于Simhash算法的海量文档反作弊技术研究[J].计算机技术与发展,2014,24(9):103-107

[12]张焕炯,王国胜,钟义信.基于汉明距离的文本相似度计算[J].计算机工程与应用,2001,37(19):21-22

[13]胡维华,鲍乾,李柯.结合汉明距离及语义的文本相似度量方法研究[J].杭州电子科技大学学报：自然科学版,2016,36(3):36-41

[14]周虎.一种基于JWT认证token刷新机制研究[J].软件工程,2019,22(12):18-20

致谢

一晃四年过去，又来到一个人生阶段的结束的结点。四年的本科生涯经历了转专业、职业志向的转变，经历了很多困难、茫然和挫折。但幸运的遇见了很多朋友，在无助的时候，拉了我一把，在这里我想对他们表示我的感谢

感谢我的导师——陈平华教授，作为导师尽心尽责为了答疑解惑。正是导师的耐心指导，我才能顺利完成毕业论文

感谢我的舍友，与我度过了一个短暂又精彩的大学生活，也是有了他们的帮助，让我对计算机领域有了更深的认识

感谢我的父母的养育之恩，在我困难的时候给与支持，对我的任何选择都无条件支持。

感谢我的好朋友——曾彦祺，作为好朋友，给了我精神上极大的支持，为我加油打气、为了排忧解难，我很幸运能在生命中遇到这样一位挚友。