

大学生论文检测系统

文本复制检测报告单(全文标明引文)

№: ADBD2023R_20230523120832473572654641

检测时间: 2023-05-23 12:08:32

篇名: 基于Web的竞赛管理系统的设计与实现

作者: 孙慧心

指导教师: 赵海龙

检测机构: 北京建筑大学

文件名: 基于Web的竞赛管理系统的设计与实现.docx

检测系统: 大学生论文检测系统

检测类型: 大学生论文

检测范围: 中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
大学生论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
源代码库
CNKI大成编客-原创作品库
机构自建比对库

时间范围: 1900-01-01至2023-05-23

检测结果

去除本人文献复制比: 10.1%

跨语言检测结果: -

去除引用文献复制比: 5.6%

总文字复制比: 10.1%

单篇最大文字复制比: 1.9% (基于Vue.js的科研管理系统实现策略研究)

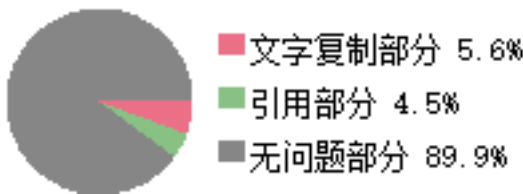
重复字数: [3979] 总段落数: [4]

总字数: [39372] 疑似段落数: [4]

单篇最大重复字数: [762] 前部重合字数: [2004]

疑似段落最大重合字数: [2114] 后部重合字数: [1975]

疑似段落最小重合字数: [344]



指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似整体剽窃 ☐ 过度引用

相似表格: 0 相似公式: 没有公式 疑似文字的图片: 0

21.8%(2114)	21.8%(2114)	基于Web的竞赛管理系统的设计与实现_第1部分 (总9707字)
4%(344)	4%(344)	基于Web的竞赛管理系统的设计与实现_第2部分 (总8651字)
4.3%(511)	4.3%(511)	基于Web的竞赛管理系统的设计与实现_第3部分 (总11877字)
11.1%(1010)	11.1%(1010)	基于Web的竞赛管理系统的设计与实现_第4部分 (总9137字)

指导教师审查结果

指导教师: 赵海龙

审阅结果:

审阅意见: 指导老师未填写审阅意见

1. 基于Web的竞赛管理系统的设计与实现_第1部分

总字数: 9707

相似文献列表

去除本人文献复制比: 21.8%(2114) 文字复制比: 21.8%(2114) 疑似剽窃观点: (0)

1	<u>基于Vue.js的科研管理系统实现策略研究</u> 程萍; - 《电脑知识与技术》 - 2023-02-25	7.9% (762) 是否引证: 是
2	<u>基于Spring Cloud的微服务架构设计</u> 曾佳; - 《电子技术》 - 2023-01-20	4.7% (459) 是否引证: 是
3	<u>大学生综合竞赛管理系统</u> 吕哲心 - 《大学生论文联合比对库》 - 2021-05-24	2.3% (220) 是否引证: 否
4	<u>基于Java EE学科竞赛管理系统的设计与实现</u> 彭杨丽;徐家喜;陈培培; - 《电脑知识与技术》 - 2018-07-05	2.2% (217) 是否引证: 是
5	16113320_李鉴_基于Java的学科竞赛信息查询网的后端设计与实现 李鉴 - 《大学生论文联合比对库》 - 2020-05-15	2.1% (204) 是否引证: 否
6	16113320_李鉴_基于Java的学科竞赛信息查询网的后端设计与实现 李鉴 - 《大学生论文联合比对库》 - 2020-05-13	2.1% (203) 是否引证: 否
7	<u>基于Java+EE学科竞赛管理系统的设计与实现</u> - 道客巴巴 - 《互联网文档资源 (https://www.doc88.co)》 - 2020	2.0% (197) 是否引证: 否
8	<u>对高校学科竞赛管理系统改进的研究</u> 任嘉;徐卓农;邓飞; - 《科技视界》 - 2020-07-15	1.8% (174) 是否引证: 否
9	<u>高校学生学科竞赛管理平台的系统设计与实现</u> 简国明;蔡润哲;林兴;刘金梅;彭建高; - 《高师理科学刊》 - 2018-05-30	1.4% (138) 是否引证: 是
10	<u>杨海泽_基于JavaWeb大学生学科竞赛发布系统的设计与实现</u> 杨海泽 - 《大学生论文联合比对库》 - 2019-05-19	1.4% (136) 是否引证: 否
11	<u>大学生科技竞赛管理方式探讨</u> 赵小蕾;谢庆辉; - 《计算机时代》 - 2015-06-15	1.0% (96) 是否引证: 是
12	<u>大学生学科竞赛管理系统的设计与实现</u> 刘伟;付元礼;黄辛迪;李智;胡为; - 《电脑知识与技术》 - 2020-06-15	0.8% (74) 是否引证: 否
13	<u>“智愿点津”高考志愿填报辅助系统设计与实现</u> 李志远;姚明菊;魏强; - 《电脑编程技巧与维护》 - 2023-01-18	0.6% (61) 是否引证: 是
14	<u>基于深度学习的超分辨率图像重建方法研究</u> 左元勋(导师: 李莉) - 《北京邮电大学硕士论文》 - 2021-06-02	0.4% (36) 是否引证: 否
15	<u>基于PDM的产品图档管理系统的研制与开发</u> 解学科;于长伟;陈东; - 《大众科技》 - 2011-03-10	0.4% (35) 是否引证: 否
16	<u>基于ASP.NET技术的小区物业管理系统的设计与实现</u> 米国佼(导师: 王连平) - 《吉林大学硕士论文》 - 2015-06-01	0.3% (33) 是否引证: 否
17	<u>王曼华计算机科学与技术20111303012</u> 王曼华 - 《大学生论文联合比对库》 - 2015-05-28	0.3% (33) 是否引证: 否
18	<u>面向移动运营商的业务投诉智能分析系统设计与实现</u>	0.3% (32)

	李文超(导师:倪巍伟;邱文君) - 《东南大学硕士论文》- 2021-06-10	是否引证: 否
19	移动终端下五层十五级遥感瓦片的多维显示及缓存优化方法研究 范心仪(导师:刘仁义) - 《浙江大学硕士论文》- 2019-06-10	0.3% (32) 是否引证: 否
20	基于Silverlight技术的社区物业管理系统的分析与设计 万念斌(导师:陈海山) - 《厦门大学硕士论文》- 2013-10-01	0.3% (31) 是否引证: 否
21	基于Web的程序设计类课程在线学习系统的设计与实现 许武贤 - 《大学生论文联合比对库》- 2016-04-02	0.3% (31) 是否引证: 否

原文内容

本科生毕业设计(论文)
基于Web的竞赛管理系统的设计与实现
姓名: 孙慧心
学号: 201903030113
学院: 电气与信息工程学院
专业(方向): 计算机科学与技术
班级: 计191
指导教师: 赵海龙
创见性声明

本人声明: 所提交的毕业设计(论文)是本人在指导教师的指导下进行的工作和取得的成果, 符合学校及学院的毕业设计(论文)管理规定, 论文中所引用的他人已经发表或撰写过的研究成果, 均加以特别标注并在此表示致谢。与我一同工作的同志对本论文所做的任何贡献也已在论文中作了明确的说明并表示谢意。

毕业论文作者签名: 签字日期: 年月日
本科毕业设计(论文)授权使用授权书

本毕业设计(论文)作者完全了解北京建筑大学有关保留、使用毕业设计(论文)的规定。特授权北京建筑大学可以将毕业设计(论文)的全部或部分内容编入有关数据库进行检索, 并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交毕业设计(论文)的复印件和磁盘。

(保密的毕业论文在解密后适用本授权说明)

毕业论文作者签名: 指导教师签名:
签字日期: 年月日 签字日期: 年月日

摘要
随着高等教育领域的快速发展和竞争压力的日益加剧, 越

姓名:	孙慧心
学号:	201903030113
学院:	电气与信息工程学院
专业(方向):	计算机科学与技术
班级:	计191
指导教师:	赵海龙

来越多的大学生选择参与各种学科竞赛以提升自身的综合素质和专业能力。然而, 由于大学学科竞赛涉及的专业范围广泛、参赛人数众多以及赛程和赛制各异等特点, 高等教育机构在组织和信息监控过程中面临诸多管理挑战。传统的人工管理方式由于效率低下、难以监控、容易出错以及信息共享困难等问题, 已经无法满足高校日益增长的学科竞赛管理需求。现有的一些学科竞赛管理系统, 功能过于简单, 无法覆盖学科竞赛的全过程, 缺乏全流程管理。

因此, 本文在对当前研究状况和相关技术进行深入调研后, 以满足大学生参与竞赛的需求为目标, 利用当下流行的微服务架构, 结合Spring Cloud、Spring Boot、Vue等先进技术, 设计并开发了一个基于Web的竞赛管理系统。本文的主要研究内容包括以下几方面:

首先, 为解决单体架构在扩展和灵活性方面的问题, 本竞赛管理系统采用了先进的微服务架构, 将系统功能拆分为九大功能模块, 分别对用户、院系、角色、通知公告、竞赛信息、竞赛队伍、竞赛文件、荣誉信息和赛事宣传进行全面管理。

其次, 针对赛事流程无法全面覆盖的问题, 我们进行了详尽的动态模型规划, 确保全程覆盖从赛事宣传到颁发奖项的每一个环节。

最后, 我们对已开发完成的系统进行了全面的系统测试, 检验各项功能的实现情况。经过一系列的测试, 该系统的运行状况良好, 各项功能均已实现, 完全符合预期的开发目标。

关键词: 竞赛管理, 微服务架构, 管理系统
ABSTRACT

With higher education's rapid progression and growing competition, more university students are participating in diverse academic contests to bolster their competencies and skills. However, due to the contests' diverse nature, numerous participants, and varying rules and schedules, higher education institutions face significant management challenges. Traditional manual management, characterized by low efficiency, difficulties in monitoring, susceptibility to errors, and sharing information, can no longer suffice the rising demand for managing academic contests in universities. Existing contest management systems are simplistic and fail to manage the whole contest process comprehensively.

Therefore, after in-depth research on the current research situation and related technologies, this paper aims to meet the needs of college students participating in competitions, using the currently popular microservices architecture, combined with advanced technologies such as Spring Cloud, Spring Boot, and Vue, to design and develop a Web-based competition management system. The main research contents of this paper include the following aspects:

Firstly, to solve the problem of the lack of scalability and flexibility in monolithic architecture, this competition management system adopts an advanced microservices architecture and splits the system functions into nine major functional modules, each fully managing users, departments, roles, notifications and announcements, competition information, competition teams, competition files, honor information, and competition promotion.

Secondly, in response to the problem of the competition process not being fully covered, we carried out detailed dynamic model planning to ensure that every aspect from competition promotion to award presentation is fully covered.

Finally, we conducted a comprehensive system test on the system that has been developed, testing the implementation of each function. After a series of tests, the system runs well, all functions have been implemented, and it fully meets the expected development goals.

Keywords: Competition Management, microservices architecture, Management System

目录

摘要.....I

ABSTRACT.....II

1.....绪言.....4

1.1.....课题背景.....4

1.2.....课题研究的目的和意义.....4

1.3.....国内外概况.....5

1.4.....论文的组织结构.....6

2.....相关技术环境介绍.....7

2.1.....微服务架构.....7

2.2.....Vue.....7

2.3.....本章小结.....8

3.....需求分析.....9

3.1.....业务分析.....9

3.2.....功能需求分析.....11

3.3.....非功能性需求分析.....21

3.4.....本章小结.....22

4.....系统设计.....23

4.1.....系统总体架构设计.....23

4.2.....功能总体设计.....24

4.3.....类和对象的设计.....35

4.4.....动态模型设计.....35

4.5.....数据库设计.....38

4.6.....本章小结.....41

5.....系统实现.....43

5.1.....系统开发环境配置.....43

5.2.....系统整体结构.....43

5.3.....系统功能的实现.....45

5.4.....系统测试.....61

5.5.....本章小结.....68

6.....总结与展望.....69

6.1.....总结.....69

6.2.....展望.....69

参考文献.....70

附录.....72

致谢.....73

1 绪言

1.1 课题背景

随着高等教育的发展，大学生参与各类学科竞赛已成为提高学生综合素质和能力的重要途径，也是对传统课堂教育的有效补充。学科竞赛可以激发学生的学习兴趣，培养创新思维和团队合作精神，加深对所学知识的理解和应用，锻炼解决实际问题的能力，提高其专业水平和实践经验[1]。

但是，由于高校学科竞赛涉及专业广泛、参赛人数众多和赛程赛制区别较大等特点，导致高校在组织开展和信息监控的过程中面临许多管理方面的困难，低效率、难监控、易出错、难共享的传统人工管理模式已无法满足高校日益增长的学科竞赛管

理需求[2]。

同时，随着21世纪科技的快速发展和数字化的推进，教育部已经明确把加强学科竞赛工作纳入实践教学与人才培养模式改革创新的质量工程中[3]。推动数字化转型，数字化管理系统的需求日益增长。尤其在大学环境中，各类竞赛活动如科研、创新创业和艺术等为学生提供了宝贵的实践机会。

竞赛管理系统作为一种能够帮助竞赛组织方高效地完成竞赛组织等相关工作的系统。它能够让学生更快地获得本校内或者高校间的学科竞赛信息，同时还可以在系统上进行竞赛的报名和成绩查询。对于竞赛的组织方而言，他们可以方便地申请举办一个新的竞赛，发布竞赛相关信息，在平台上查看竞赛的报名情况以及报名学生的基本信息[4]。

虽然目前已研发了一些学科竞赛管理系统，但部分系统功能较为简单，无法覆盖学科竞赛全过程，缺乏全流程管理[5]。所以开发一个功能全面、可以覆盖整个竞赛参与流程的竞赛管理系统，具有非常重要的实用价值。

1.2 课题研究的目的是意义

本研究的目标是设计并实现一个大学竞赛管理系统，以提高竞赛的管理效率和参与体验。该系统将解决以下问题：

1、提升大学竞赛的管理效率：

通过采用数字化管理系统，可以将管理报名、审核、资源分配和成果展示等过程自动化和集中化，大大提高了竞赛的组织效率、管理效率，减少人力资源和时间成本，使得竞赛能够更加顺利地进行 [1, 6]。

2、提高竞赛参与体验：

通过提供清晰的竞赛进度、便捷的报名方式等功能，该系统可以提高学生的参与体验，进而提高参与度。参赛者可以方便地了解活动信息、报名参赛和提交作品，从而提升参与的便利性和满意度[7]。

3、促进教育的数字化：

该研究有助于推进教育的数字化进程，为未来的教育管理提供参考。通过引入数字化管理系统，大学竞赛活动可以更好地适应科技发展的需求，实现信息化、网络化和智能化的管理。这将有助于提高教育管理效率、优化资源配置以及培养学生的数字素养和创新能力[8]。

4、推动资源的优化配置：

通过竞赛管理系统组织者和管理者可以收集竞赛数据，通过对数据的分析，可以获得对竞赛进展和结果的实时了解，从而制定更科学的资源分配策略和管理决策，提升竞赛活动的质量[9]。

综上所述，本研究对于推进大学竞赛的管理水平，提高大学生的参与体验，以及促进教育的数字化都具有重要的意义。通过设计和实现一个高效的大学竞赛管理系统，可以有效应对竞赛管理中的挑战，提升竞赛效率和质量，从而为大学竞赛活动的发展和提升做出积极的贡献。

1.3 国内外概况

近年来国内对竞赛管理系统的研究逐渐增多。一些研究聚焦于特定类型的竞赛，如科研竞赛、创新创业竞赛和学科竞赛等。这些研究往往侧重于竞赛管理系统的功能设计和应用实践，旨在提高管理效率和参与体验[6]。此外，一些研究关注竞赛管理系统的技术实现和架构设计。这些研究探讨了不同技术平台和开发框架，如Web应用、云计算和大数据分析等，以支持竞赛管理系统的开发和运行[7]。

国外的竞赛管理系统研究也有较多的成果。一些研究聚焦于教育领域，探讨如何利用竞赛管理系统来提高学生参与和学习成效。这些研究强调系统的用户友好性、数据分析功能以及学习资源的整合和分享。

另一些研究侧重于竞赛组织和评审的自动化和智能化。这些研究通过引入自然语言处理、机器学习和人工智能等技术，提供智能化的评审和结果分析，从而提高竞赛的公平性和准确性。

竞赛管理系统的研究在中国国内外都有一定的发展。国内的研究主要集中在功能设计、应用实践和技术实现方面，旨在提高管理效率和参与体验。国外的研究则更加注重教育和智能化方面的应用，致力于提供个性化的学习支持和智能化的评审过程。

然而，目前仍存在一些挑战和研究方向。例如，如何充分利用大数据和人工智能等新技术，进一步提升竞赛管理系统的效能和创新性；如何设计系统以满足不同类型竞赛的需求，包括科研竞赛、创新创业竞赛和学科竞赛等；如何提高竞赛管理系统的安全性和隐私保护等。

综上所述，竞赛管理系统的研究在中国国内外都取得了一定的进展，但仍有待进一步深入研究和应用，以促进竞赛活动的有效管理和提升学生的参与体验。

1.4 论文的组织结构

第1章，绪言。介绍了竞赛管理的研究背景、意义、国内外研究现状、所要解决的问题。

第2章，相关技术环境介绍。在构建竞赛管理系统中所使用的Vue、Spring Cloud、微服务主要技术。

第3章，需求分析。对竞赛管理系统进行面向对象的需求分析，并辅以用例图等图示进行更加标准的需求分析。

第4章，系统设计。阐述系统实现的主要重点，并辅以实体类图、功能模块图等图示进行详细明确的阐述。

第5章，系统实现。使用相关开发工具完成角色管理系统微服务的搭建、前端和后端的开发，并展示部分关键代码和界面。

通过展示重点功能的测试用例，确保系统的实现如预期设计。

第6章，总结及展望。对竞赛管理系统的设计与实现工作做了简要的总结，并对后续工作提出了设想。

2 相关技术环境介绍

2.1 微服务架构

2.1.1 微服务架构的特点

微服务架构是从 SOA 发展而来。微服务据业务范围划分多个小的工作单位，每个单位具有高内聚性、低协同性，能以独立实体形式存在系统进程中。在技术维度方面来看，可将微服务视作一个个组件，运行便捷，占用空间也少。不像单体架构，不同服务之间会有耦合关系，如若在系统运行过程之中对某一系统改变，就得重新架构服务。在代码实现方面来看，几个功能固定通用的情况下，就可将应用分解，模块化的解决重复编码的问题，提高应用效果，不管开发还是运维都更加快捷，所以微服务有着分解化服务处理、复杂度低、技术多元化的特点[10]。这样的设计让微服务架构实现了解耦，减少了开发大型集成程序的必要性，让公司能够快速、可靠地对程序进行迭代[11]。

近些年来，微服务架构越来越受欢迎，它模块化的风格特别适合云环境。很多互联网公司的应用程序都已经从单一架构发展到微服务架构，比如谷歌和亚马逊[12]。尤其是在繁琐的业务情境下，无法采用单体架构去解决问题，而微服务架构就能很好地解决这个问题，实现不同功能的独立部署。

因此，采用微服务架构作为框架可以更好的进行功能拓展，降低运维成本。

2.1.2 Spring Cloud微服务架构

首先，完整的微服务需要注册中心用来集中管理，实现服务的注册，发现，检查等功能。其次，架构中每个微服务一般会部署为单个独立的进程，无服务状态时，会有多个独立进程提供服务。同时，架构中要内嵌一种负载均衡机制。最后，一个好的微服务平台应该方便扩展和部署。Spring Cloud就是一套完整的微服务解决方案，包括服务发现、服务注册、配置中心、链路追踪、异常处理等模块。Spring Cloud提供了标准化的、全栈式的技术方案，可以认为Spring Cloud是构建微服务的最佳方案[13]。

2.2 Vue

Vue.js是轻量级前端框架，其具有良好的性能和较强灵活性，它基于标准HTML、CSS和JavaScript构建，并提供了一套声明式的、组件化的编程模型，帮助高效地开发用户界面[14]。

其采用的是虚拟DOM机制，具体操作内存中对象运行速度较快，待更新后，可将最终js对象映射为真实DOM，并交由浏览器进行绘制，进而解决浏览器性能问题[15-16]。Vue能够通过各变量添加多个setter以及getter，这样实现双向数据绑定，能够简化Web前端的开发流程，帮助研发者降低开发难度，提升前端开发效率，使高校能够缩短研发周期和成本。

Vue-router以及vuex。Vue-router是Vue.js所发布的路由插件，其中Vue.js为深度集成。传统页面运用，是采用超链接的方式实现页面跳转、切换，而Vue-router路由插件单页面应用过程中，是通过路径之间切换的。而Vue.js实现无刷新访问页面、改变页面，实现页面路径动态调整，便于页面的切换，不断提升用户的体验感。vuex是Vue.js所提出的全局状态管理方案。能够在全球各实例中放置vuex，使各组件同步响应，进而减少系统运行复杂性。在管理全局状态过程中，需要区分组建本地状态以及应用层级状态，以防组建未来状态，被放置应用及状态中进行管理[16]。

Axios。Axios是基于promise的HTTP库，其能够用于node.js以及浏览器中。Axios可从浏览器创建XMLHttpRequests，能够支持API、拦截响应、请求转换，请求数据、响应数据等[17]。

云数据存储技术。该系统数据存储采用租赁第三方数据中心方式完成，即云存储技术，该技术将数据存储在云端多个服务器上，用户可方便通过系统程序接口API访问存储资源池中的数据。使用云存储技术不但可以提高系统的安全可靠性，而且实现了系统数据的海量存储[18]。

2.3 本章小结

本章主要介绍了在构建竞赛管理系统中所使用的微服务框架、Spring Cloud框架Vue这几种相关技术的技术介绍和特点分析。为之后的实现奠定了基础。

3 需求分析

3.1 业务分析

系统预计设定三个用户角色：管理员、学生用户和教师用户。

1. 学生用户：学生需要在系统中进行个人信息管理，如：修改个人信息。他们还需要在系统中了解最新的竞赛信息，选择想参与的竞赛，与同学及指导教师组队并报名参与竞赛，在竞赛进行期间，为了信息交流、成果共享，学生还还需要上传或下载与竞赛相关的文件。最后，在竞赛结束后，如果他们取得了奖项或产生了一些成果，他们需要上传自己的竞赛奖项和成果信息。当系统中发布了一些通知公告或赛事宣传信息，学生需要及时的查看赛事宣传和通知公告。

2. 教师用户的权限与学生用户相似，他们也可以管理自己的个人信息，查阅最新的竞赛信息，查看系统中的通知公告、赛事宣传信息。在竞赛进行的过程中，教师可以在所指导的小队中上传一些指导材料，也可以下载学生在竞赛中产生的一些竞赛交流文件。比赛结束后，他们还可以查看小队的竞赛奖项和成果信息。

3. 管理员在系统中拥有最高权限，他们可以进行用户管理，包括管理用户个人信息，用户角色信息和用户所在的院系信息。在竞赛开始前，管理员需要在系统中发布一些赛事宣传信息，同时还要对竞赛信息进行维护。此外，当学生报名竞赛后，管理员需要负责审批参赛队伍的报名申请，管理竞赛过程的文件，审批小队获得的奖项和成果，以及发布通知公告和赛事宣传。

根据以上描述，并参考面向对象软件工程的方法，可以将该系统初步划分为以下模块：

1. 管理员端：用户管理模块（包括个人信息管理、角色管理、院系管理）、通知公告模块、竞赛信息管理模块、竞赛队伍管理模块（包括报名审批、文件管理）、荣誉信息管理模块（包括奖项和成果审批）、赛事宣传模块。

2. 学生用户端：个人信息管理模块、通知公告查阅模块、竞赛信息查阅和申请模块（包括组队、报名、文件上传下载）、获奖成果上传模块。

3. 教师用户端：个人信息管理模块、通知公告查阅模块、竞赛信息查阅和指导模块（包括上传下载竞赛通知、查看小队获奖成果）。

这个系统设计涵盖了竞赛管理的全流程，可以使管理员、学生和教师更高效地参与和管理竞赛活动，具有较高的应用价值。

根据业务分析，图31中给出了包含系统中所有功能的用例图。

图31系统总用例图

3.2 功能需求分析

3.2.1 用户管理功能需求分析

本竞赛管理系统需要对使用者进行身份鉴定，必须是符合条件的使用者才能使用本系统。由于本系统属于内部使用，因此并没有开放注册功能。新用户的账户创建是由管理员在用户管理模块中完成的。管理员将新用户的账户名和密码创建完成后直接下发给新用户，由此新用户才能开始使用该系统。

图32中展示了用户管理模块管理员可以操作的功能

图32用户管理用例图

表格31给出了管理员添加用户的用例描述。

表格 31添加用户信息用例描述

用例编号	1
用例描述	管理员添加用户信息
参与者	管理员用户
前置条件	管理员已登录系统并进入用户管理模块
后置条件	系统保存用户信息并显示添加成功提示
基本流程	管理员点击添加按钮，系统显示添加页面 管理员输入用户姓名、性别、年龄、联系方式、角色、院系等信息 点击查询按钮 系统保存用户信息并显示添加成功提示，返回列表页面

用例编号 1
用例描述管理员添加用户信息
参与者管理员用户
前置条件管理员已登录系统并进入用户管理模块
后置条件系统保存用户信息并显示添加成功提示
基本流程管理员点击添加按钮，系统显示添加页面
管理员输入用户姓名、性别、年龄、联系方式、角色、院系等信息
点击查询按钮
系统保存用户信息并显示添加成功提示，返回列表页面

3.2.2 院系管理功能需求分析

院系管理模块旨在向管理员提供一个用户管理途径，帮助管理员更精准的、快捷的管理用户所属的院系信息。

此外，只有在完成院系的设置之后，管理员才能够对用户进行院系分配。通过这种方式，管理员能够更清晰、更直观地了解和统计院系内的用户分布情况。这不仅便于内部管理，同时也有助于在赛事宣传方面，根据不同院系的特点和需求，发布更加有针对性的赛事宣传内容。这种细分的宣传策略可能会提高赛事的参与度和影响力。

图 33中给出了院系管理模块管理员可以进行的操作。

图 33院系管理用例图

表格 32中给出了查询用户信息的用例描述。

表格 32查询院系信息用例描述

用例编号	2
用例描述	管理员根据条件查询院系信息
参与者	管理员用户
前置条件	管理员已登录系统并进入院系管理模块
后置条件	系统查询到院系信息并显示
基本流程	管理员根据院系名称、负责人等条件输入查询关键字 管理员点击查询按钮 系统根据输入的查询条件搜索匹配的院系信息 系统显示符合查询条件的院系信息列表

用例编号 2
用例描述管理员根据条件查询院系信息
参与者管理员用户
前置条件管理员已登录系统并进入院系管理模块
后置条件系统查询到院系信息并显示
基本流程管理员根据院系名称、负责人等条件输入查询关键字
管理员点击查询按钮
系统根据输入的查询条件搜索匹配的院系信息
系统显示符合查询条件的院系信息列表

3.2.3 角色管理功能分析

角色管理模块的主要目的在于定义并分配用户在系统中的数据访问权限。

在系统中，不同的用户担任着不同的角色，每一种角色在系统中享有各自特定的数据权限。例如，管理员角色拥有对系统中所有数据的全权管理，学生角色在用户管理模块中仅能访问和修改自己的个人信息，而教师角色在竞赛队伍模块中则只能访问和管理所指导小队的相关文件数据。

角色管理在整个系统中占据了至关重要的地位，它是系统安全性和有效性的关键保障，有力地维护了不同角色用户的权益，也有助于管理员高效管理系统。

图 34中给出了角色管理模块管理员可以进行的操作。

图 34角色管理用例图

表格 33中展示了角色管理模块角色分配数据权限功能的用例描述。

表格 33为角色分配数据权限用例描述

用例编号	3
用例描述	管理员为角色分配数据权限
参与者	管理员用户
前置条件	管理员已登录系统并进入院系角色模块，选择要分配数据权限的角色，点击数据权限按钮
后置条件	系统保存角色数据权限信息并显示修改成功提示

用例编号 3
用例描述管理员为角色分配数据权限
参与者管理员用户

前置条件管理员已登录系统并进入院系角色模块，选择要分配数据权限的角色， 点击数据权限按钮
后置条件系统保存角色数据权限信息并显示修改成功提示
(续)

基本流程	系统显示当前角色已有的数据权限和可选的数据列表 管理员在可选数据列表中 选择要分配的数据 管理员点击添加按钮，将选中的数据分配给角色 系统验证管理员的权限分配操作 系统更新角色的数据权限
------	---

基本流程系统显示当前角色已有的数据权限和可选的数据列表
管理员在可选数据列表中选择要分配的数据
管理员点击添加按钮，将选中的数据分配给角色
系统验证管理员的权限分配操作

系统更新角色的数据权限

3. 2. 4 通知公告模块

通知公告模块的主要功能在于发布及管理 与竞赛相关的各类通知和公告。此模块提供了一个桥梁，使得关于竞赛的活动、政策和服务等重要信息能够有效地传递给学生和教师。不同于赛事宣传模块针对特定竞赛的推广，通知公告模块发布的信息面向全体学生，涵盖通知公告的全局性内容。这一模块的存在极大地提高了信息传递的效率，保证了信息的准确性和及时性。

图 35展示了用户在通知公告模块可进行的操作。

图 35通知公告模块用例图

表格 34中给出了管理员查看通知公告详细信息的用例描述。

表格 34管理员查看通知公告内容用例描述

用例编号	4
用例描述	管理员查看通知公告的详细信息。
参与者	管理员用户
前置条件	管理员已登录到系统中，并进入到通知公告模块，模块中至少存在一个需要查看的通知公告
后置条件	系统显示选中通知公告的详细信息。
基本流程	管理员选择执行通知公告查看功能。 系统显示通知公告查看界面。 管理员选择要查看的通知公告。

用例编号 4

用例描述管理员查看通知公告的详细信息。

参与者管理员用户

前置条件管理员已登录到系统中，并进入到通知公告模块，模块中至少存在一个需要查看的通知公告

后置条件系统显示选中通知公告的详细信息。

基本流程管理员选择执行通知公告查看功能。

系统显示通知公告查看界面。

管理员选择要查看的通知公告。

3. 2. 5 竞赛信息管理功能分析

竞赛信息管理模块的任务在于处理和管理所有竞赛的信息。学生和教师用户可以根据详细竞赛信息，作出参赛的决策。他们可以根据竞赛类型、时间安排，以及其他相关信息选择参加哪一项竞赛，从而实现高效的赛事管理和参与。

指 标

疑似剽窃文字表述

1. 背景

随着高等教育的发展，大学生参与各类学科竞赛已成为提高学生综合素质和能力的重要途径，也是对传统课堂教育的有效补充。

2. 基于Web的竞赛管理系统的设计与实现_第2部分	总字数：8651
----------------------------	----------

相似文献列表

去除本人文献复制比：4%(344) 文字复制比：4%(344) 疑似剽窃观点：(0)
--

1	4_杨远林_基于SSH的OA办公系统的设计与实现	2.0% (172)
	杨远林 - 《高职高专院校联合比对库》 - 2018-04-30	是否引证：否
2	基于PMI的电子政务访问权限控制的研究与实现	1.4% (122)
	马欣(导师：武维善) - 《西安建筑科技大学硕士论文》 - 2008-05-01	是否引证：否
3	14-1401430120-李乐鸿	0.4% (32)
	李乐鸿 - 《大学生论文联合比对库》 - 2018-06-07	是否引证：否
4	地铁供电系统可靠性和安全性分析方法研究	0.3% (30)
	莫景泉； - 《中国高新技术企业》 - 2013-01-01	是否引证：否

原文内容

总体而言，通过这一模块，各类用户可以轻松管理和访问竞赛信息，从而增强了竞赛管理系统的实用性和用户体验。图 36中展示了竞赛信息管理模块用户可以进行的操作。

图 36竞赛信息管理用例图

表格 35中给出了管理员导出竞赛信息的功能描述。

表格 35导出竞赛信息用例描述

用例编号	5
用例描述	管理员导出竞赛信息
参与者	管理员用户
前置条件	管理员已登录系统并进入院系竞赛信息管理模块，模块中至少存在一个可以导出的竞赛
后置条件	竞赛信息导出成功
基本流程	系统显示选中竞赛的当前信息 管理员选择要导出的竞赛信息 点击导出按钮。 系统导出并下载竞赛的信息文件

用例编号 5

用例描述管理员导出竞赛信息

参与者管理员用户

前置条件管理员已登录系统并进入院系竞赛信息管理模块，模块中至少存在一个可以导出的竞赛

后置条件竞赛信息导出成功

基本流程系统显示选中竞赛的当前信息

管理员选择要导出的竞赛信息

点击导出按钮。

系统导出并下载竞赛的信息文件

3.2.6 竞赛队伍管理模块

竞赛队伍管理模块的主要目标在于促使学生用户自主组建小队并报名参加竞赛。总体来说，竞赛队伍管理模块在整个系统中占据着核心地位，它实现了学生用户的主动参与和自我管理，有助于提高竞赛的活跃度，同时也大大提升了管理的效率和便利性。图 37中展示了用户在竞赛队伍管理模块中可以进行的操作。

图 37竞赛队伍管理模块用例图

表格 36中给出了学生报名参加竞赛的功能描述。

表格 36学生报名参加竞赛用例描述

用例编号	6
用例描述	学生用户报名参加竞赛
参与者	学生用户
前置条件	学生用户已登录到系统中并进入竞赛队伍管理模块，在竞赛信息管理模块中至少存在一个竞赛可以报名参加。
后置条件	学生用户成功报名参加竞赛
基本流程	学生在报名竞赛页面填写小队信息 系统显示可报名参加的竞赛列表。 用户选择要参加的竞赛 系统验证用户满足报名参加的条件。 用户确认报名操作，并点击报名按钮 系统将用户加入竞赛的参赛队伍中

用例编号 6

用例描述学生用户报名参加竞赛

参与者学生用户

前置条件学生用户已登录到系统中并进入竞赛队伍管理模块，在竞赛信息管理模块中至少存在一个竞赛可以报名参加。

后置条件学生用户成功报名参加竞赛

基本流程学生在报名竞赛页面填写小队信息

系统显示可报名参加的竞赛列表。

用户选择要参加的竞赛

系统验证用户满足报名参加的条件。

用户确认报名操作，并点击报名按钮

系统将用户加入竞赛的参赛队伍中

3.2.7 竞赛文件管理模块

当竞赛队伍参与竞赛的过程中，组内会有分工合作，指导教师也会队小组成员进行指导，这时后就需要竞赛文件管理模块。学生用户可以在这个模块上传/下载所需的竞赛文件，队内的组员，可以及时的接收到最新竞赛进度，教师也可以及时的向小队下发任务。这个功能的需求主要是提供一个队内成员信息交流的平台。

竞赛文件管理模块的主要目标在于整合管理竞赛过程中所产生的各类文件内容，涵盖了竞赛通知、竞赛过程文件、竞赛报名表等。

此模块特别强调了文件管理的权限划分和信息安全。每个竞赛文件都与特定的小队相对应，因此，小队成员只能查看和操作属于自己小队的竞赛文件。这样的设计旨在保护信息的隐私性和完整性，防止非相关队伍的成员查看或修改文件内容。

通过有效地组织和管理竞赛文件，此模块极大地提高了信息查找的效率，同时也为竞赛过程中的沟通和协作提供了便利。总体来说，竞赛文件管理模块在整个系统中起着至关重要的作用，它确保了竞赛过程中各类文件的安全存储和高效管理。

图 38中给出了用户在竞赛文件管理模块可以进行的操作。

图 38竞赛文件管理模块用例图

表格 37展示了学生用户下载竞赛文件的用例描述。

表格 37学生下在竞赛文件用例描述

用例编号	7
用例描述	教师用户下载竞赛文件
参与者	教师用户
前置条件	教师用户已登录到系统中，并进入竞赛文件管理模块，用户报名参加了竞赛，且用户所在竞赛小队中至少存在一个可下载的竞赛文件。
后置条件	系统将选中的文件下载到本地设备，显示成功下载文件的提示
基本流程	系统显示可下载的竞赛文件列表 教师用户选择要下载的文件 教师用户选择执行文件下载功能 教师用户点击下载按钮

用例编号 7

用例描述教师用户下载竞赛文件

参与者教师用户

前置条件教师用户已登录到系统中，并进入竞赛文件管理模块，用户报名参加了竞赛，且用户所在竞赛小队中至少存在一个可下载的竞赛文件。

后置条件系统将选中的文件下载到本地设备，显示成功下载文件的提示

基本流程系统显示可下载的竞赛文件列表

教师用户选择要下载的文件

教师用户选择执行文件下载功能

教师用户点击下载按钮

3.2.8 荣誉信息管理模块

荣誉信息管理模块的主要目标在于集中管理竞赛结束后，各竞赛队伍所获得的奖项和成果信息。此模块收录的荣誉信息涵盖了荣誉名称、荣誉类型、获奖的参赛队伍、所参加的竞赛等信息。

此模块在权限管理上特别注意了信息的对应性和隐私性。每项竞赛荣誉都以每个小队为单位，进行荣誉信息的管理和展示。因此，小队成员只能查看和操作属于自己小队的荣誉信息。这种设计旨在确保信息的隐私性，防止非相关小队的成员查看或修改荣誉信息，同时也鼓励团队内部的自主管理和信息分享。

荣誉信息管理模块主要职责在于对竞赛结束后产生的荣誉信息进行统一的管理和存档，以供将来的学生参考和学习。通过提供详细且全面的历史荣誉记录，此模块为学生提供了丰富的参考资料，有助于他们了解过往的竞赛历程，并从中学习和借鉴。

总体来说，荣誉信息管理模块在整个系统中起着关键的作用。它既为当前的小队成员提供了方便快捷的荣誉信息管理服务，也为未来的学生提供了学习和参考的宝贵资源。此模块对于激发学生的学习兴趣，增强学生的学习动力，以及提高整个竞赛的影响力和公信力都起到了至关重要的作用。

图 39中展示了用户在荣誉信息管理模块可以进行的操作。

图 39荣誉信息管理模块用例图

表格 38中给出了学生用户添加荣誉信息功能的用例描述。

表格 38学生添加荣誉信息功能用例描述

用例编号	8
用例描述	学生用户添加荣誉信息
参与者	学生用户，管理员用户
前置条件	学生用户已登录到系统中，并进入荣誉信息管理模块
后置条件	新的荣誉信息成功添加到系统中，并提示新增成功

用例编号 8

用例描述学生用户添加荣誉信息

参与者学生用户，管理员用户

前置条件学生用户已登录到系统中，并进入荣誉信息管理模块

后置条件新的荣誉信息成功添加到系统中，并提示新增成功

(续)

基本流程	学生用户用户点击荣誉信息添加 系统显示荣誉信息添加界面 管理员输入新荣誉的名称、描述、获得日期等基本信息 学生用户提交荣誉信息 管理员确认输入的信息无误后，审核完成 系统将新的荣誉信息保存存到系统中
------	---

基本流程学生用户用户点击荣誉信息添加

系统显示荣誉信息添加界面

管理员输入新荣誉的名称、描述、获得日期等基本信息

学生用户提交荣誉信息

管理员确认输入的信息无误后，审核完成

系统将新的荣誉信息保存存到系统中

3.2.9 赛事宣传模块

为了优化竞赛的宣传效果，更有效地吸引学生关注和参与，该模块在系统内设立了首页展示状态。一旦某项赛事的宣传状态被设定为首页宣传图，该赛事的宣传图便会在首页处以滚动播放的形式进行展示。这样的设计旨在通过高效的信息传播和视觉冲击力，最大化地提升竞赛活动的公众知晓度。它是链接竞赛活动与潜在参赛者的重要桥梁。通过有效的信息推广和展示，此模块有助于激发学生的竞赛热情，从而推动整个竞赛活动的顺利进行。图 310给出了用户在赛事宣传模块可进行的操作。

图 310赛事宣传管理模块用例图
表格 39展示了管理员用户设置首页宣传图的用例描述。
表格 39管理员设置首页宣传图用例描述

用例编号	9
用例描述	管理员用户设置首页宣传图
参与者	管理员用户
前置条件	管理员已登录到系统中，并进入赛事宣传模块，模块中存在需要设置为首页宣传图图片
后置条件	首页宣传图成功设置
基本流程	系统显示可选的宣传图片列表 管理员选择要设置为首页宣传图图片 管理员确认选择的图片无误后，点击保存按钮 系统验证管理员的设置权限 系统将选中的图片设置为首页宣传图

用例编号 9

用例描述管理员用户设置首页宣传图

参与者管理员用户

前置条件管理员已登录到系统中，并进入赛事宣传模块，模块中存在需要设置为首页宣传图图片

后置条件首页宣传图成功设置

基本流程系统显示可选的宣传图片列表

管理员选择要设置为首页宣传图图片

管理员确认选择的图片无误后，点击保存按钮

系统验证管理员的设置权限

系统将选中的图片设置为首页宣传图

3.3 非功能性需求分析

3.3.1 可靠性分析

基于以上的功能性分析，目标是构建一个基于Web的竞赛管理系统。这类系统典型地属于传统的后端管理系统，所以在其开发过程中所面临的风险较低，开发效率高。

为提高系统的可靠性，本系统采用的技术栈包括Vue.js框架、Spring Cloud框架、MyBatis框架以及MySQL数据库。这些都是经过实践检验且被广泛接受的成熟技术，适合于快速且高效地构建Web应用程序。

另外，系统还采用了微服务架构，这一设计将使得各个模块间的耦合度降低，更加利于维护和迭代。同时，本系统采用的前后端分离的开发方式，使得前端和后端的开发能够并行进行，从而进一步提高开发效率。

总体而言，从可靠性角度看，该系统的开发并无明显的技术难题。各项所需技术已在市场上得到了广泛应用和验证，可确保项目的顺利进行。同时，微服务和前后端分离的架构设计，也将有助于提高开发效率，减少后期的维护成本。

3.3.2 安全性分析

在开发竞赛管理系统的过程中，安全性的问题尤为重要，主要涉及到私密性和可靠性两个方面。私密性意味着必须确保系统中的资源不会被未经授权的个体窃取或盗用，而可靠性则保证资源不会遭受丢失、损坏或篡改。

首先，对于前端安全性，本系统开发采用Vue.js框架，这种设计只向用户展示HTML页面元素，而业务逻辑则隐藏在前端JS代码中，这一点有效地防止了用户从前端找到并攻击系统的接口。

对于后端，系统将采用了MyBatis框架进行数据库交互，而避免了直接生成SQL语句的做法。这样的设计能够在很大程度上预防SQL注入攻击，从而避免了因此类攻击导致的系统数据丢失或篡改的风险。

同时，系统中将加入合适的认证和授权机制，确保只有授权的用户才能访问到相应的资源和服务，进一步提高了系统的私密性和可靠性。

综上所述，本系统在设计之初就充分考虑了安全性问题，采取了一系列策略和技术来确保系统的私密性和可靠性，从而达到高效且安全的管理竞赛信息的目标。

3.4 本章小结

在本章节中，首先，确定了系统的用户角色划分；其次，划分并分析了各个关键模块，并通过用例图对各模块功能进行了描绘；此外，还对系统的可靠性和安全性进行了分析，满足了面向对象软件工程需求分析部分的基本要求。

4 系统设计

4.1 系统总体架构设计

构建竞赛管理系统的总体设计遵循了GRASP的原则。具体设计如下：

首先，系统将相同的对象封装到各自的模块中，清晰地定义了各个模块的职责，从而提升了系统的内聚性。同时，通过采取前后端分离的开发方式，系统的耦合性得以降低，使得开发和维护工作更加灵活和便捷。

后端微服务采用了经典的三层架构，由控制层处理前端的请求，服务层负责业务逻辑的处理，而数据访问层则进行具体的数据操作。这种设计使得各层职责明确，结构清晰，同时保障了系统的扩展性和可维护性。

系统采用了MVC的设计模式加上微服务架构，使每个微服务都可以被看作是一个独立的MVC应用。同时，每个模块负责各自的功能，有利于新功能的添加和系统的扩展。模块与模块间的隔离，也可以有效减少系统变化对其他模块的影响，提高了系统的稳定性和鲁棒性。

4.1.1 MVC模式

本系统设计采用MVC设计模式，其核心思想是将应用程序的内部数据、用户界面和控制器逻辑分离开来，使系统设计更加清晰，程序的可维护性和可扩展性也得到了提高。

MVC模式中，用户与View进行交互，View将用户的操作转发给Controller。Controller接收到用户的请求后，根据请求的内容，Controller会决定调用哪个Model来处理这个请求。Model接收到请求后，进行相应的数据处理，然后将处理结果返回给Controller。Controller接收到Model的处理结果后，将这个结果传递给相应的View进行展示。View接收到数据后，将其渲染到

用户界面上。这个过程形成了一个完整的请求处理流程，实现了数据、界面和逻辑的分离。

4.1.2 微服务总体架构设计

微服务架构通过将单体系统分解成许多独立的模块，允许这些模块在不同的服务器上运行，因此需要进行远程调用以实现模块间通信的问题。Spring Cloud Alibaba引入了注册中心来对微服务模块进行统一管理和调用。所有的微服务模块都需在注册中心注册自己的地址和名称，同时也需要通过注册中心获取其他模块的地址和名称，以此实现模块间的远程调用。

然而，多模块的分开部署意味着不同服务有不同的IP地址。在某些情况下，前端可能需要调用不同的微服务模块以完成相应的操作，这无疑带来了一定的不便。为了解决这个问题，Spring Cloud Alibaba引入了一个网关来对外部访问进行集中管理。外部应用只需直接访问网关地址，网关则从注册中心读取不同服务的地址，并将对应的请求进行转发。这种方法实现了一致的路由方式，减少了因模块地址不同导致的繁琐调用过程，显著提升了系统的便捷性和效率。

4.2 功能总体设计

在第三章的需求分析中已经对各个模块的基础功能和作用进行了详细的阐述。因此，本章节将通过展示管理员用户端、学生用户端和教师用户端的各自模块划分图，更直观的阐述模块与模块之间的关系。同时，通过展示各用户端主要功能的功能流程图更清晰的展现功能设计。

图 41中展示了系统总体功能模块图。

图 41系统功能模块图

4.2.1 管理员端功能设计

管理员端端功能模块如图 42所示。

管理员端主要负责对各类数据的管理，是整个系统中最常用且最主要的功能。它被划分为了以下9个模块：用户信息管理，通知公告管理，角色管理，院系管理，竞赛信息管理、竞赛小队管理、竞赛文件管理、荣誉信息管理和赛事宣传。

图 42管理员端功能模块图

用户管理的设计的核心目标是处理竞赛管理系统内各类用户的基本信息。

这个模块主要包含四个核心功能：用户信息添加、修改、查询、删除功能。

在用户信息添加功能中，管理员需要输入新用户的基础信息，包含姓名、性别、年龄、联系方式、角色和所属院系等数据

。用户信息修改功能使管理员能够更新已有用户的基本信息，这包括但不限于姓名、性别、年龄、联系方式、角色以及所属院系。系统用户如因毕业、离职等原因不再使用本系统，管理员可以通过将用户状态设置为禁用来阻止其再次登录系统。这种策略有助于维护系统的安全性和有效性，保证只有当前活跃的用户才能登录和使用系统。这项功能确保了用户信息的时效性和准确性。

用户信息查询功能赋予了管理员根据特定条件检索用户信息的能力，这些条件可能包括用户名、联系方式、所属院系等。这对于管理员筛选特定用户群体非常有用。

在用户信息删除功能中，管理员有权移除用户的基础信息，实现对不再需要的用户数据进行清理。

管理员新增用户的流程，如图 43所示。

图 43管理员新增用户功能流程图

院系管理功能的设计考虑了其应用环境为大学校园，管理员需要根据学校的最新政策或实际情况，对院系信息进行维护、更新。这个模块的主要功能分为四个部分：院系信息添加、信息修改、信息查询以及信息删除。

在院系信息添加功能中，管理员可以创建新的院系，并为其填入相关的基础信息。这样的设定确保了随着院系结构的调整或新院系的设立，系统能够实时更新，与实际情况保持一致。

通过院系信息修改功能，管理员有权调整已有院系的信息，从而保证信息的时效性和准确性，以适应可能发生的任何院系变动。

在院系信息查询功能中，管理员可以根据特定的条件筛选院系，例如，通过院系名称、院系负责人等关键字进行搜索。这有助于管理员快速定位特定的院系信息。

使用院系信息删除功能，管理员可以删除不再存在或者不再需要的院系信息，保持系统内的信息简洁准确。

图 44管理员修改院系信息功能流程图

角色管理功能设计的原因是为系统内不同用户分配数据权限，以确保数据的安全性、私密性。这个模块除了基本的增删改查功能，还专门设计了三个核心功能：菜单权限分配、数据权限分配、以及用户角色分配。

菜单权限分配功能允许管理员为不同的角色分配不同的菜单权限，这确保了用户只能访问到自己角色范围内的功能模块，提升了系统的安全性和易用性。

数据权限分配功能则关注于具体的数据访问和操作权限。根据角色的不同，管理员可以灵活设置各种数据的查看、修改、删除、创建等权限。

用户角色分配功能允许管理员为每个用户分配具体的角色，从而确定他们在系统中的数据访问和操作权限。这一功能为用户管理提供了灵活性和便捷性，使得权限管理能够精确到个体用户。例如，管理员可以为“教师”角色分配仅个人数据权限，那么该角色用户就只能看到本人的个人信息及所在小队的数据。管理员为角色分配用的流程如图 45所示。

图 45管理员为角色分配用户功能流程图

赛事宣传功能的设计目标是提升竞赛活动在学生和教师群体中的传播范围。在赛事宣传模块中，主要功能包括赛事宣传的添加、修改、查询和删除。

管理员用户具有发布新的赛事宣传信息的权利。在发布新的赛事宣传信息时，管理员需要填写相关的信息，如赛事的名称、时间、地点、规则、参赛条件等，以及上传相应的宣传图。管理员还可以选择将某个特定的赛事宣传图设为首页宣传图，这样，当用户登录系统时，首页上会滚动播放的就是该赛事的宣传图，提升了赛事的曝光度。

管理员更改赛事宣传功能流程如图 46所示。

图 46管理员更改赛事宣传状态功能流程图

竞赛信息管理模块的设计目标主要是能更清晰的向学生和教师展示出，具体竞赛的相关信息如：竞赛类型、竞赛开始时间

和结束时间等，方便学生及教师根据赛事的重要程度、个人时间安排上更精确的选择想要参与的竞赛。此模块的核心功能包括增加、删除、修改、查看竞赛信息，以及查看并统一管理报名参加各项竞赛的队伍。

管理员负责在系统中录入各项竞赛的信息，包括但不限于竞赛名称、竞赛类型、赞助单位、官方链接、竞赛的开始与结束时间，以及竞赛状态。这种详尽的信息录入确保了系统的内容完整性和准确性。

管理员在可以利用竞赛信息管理功能对竞赛信息进行实时更新，确保学生了解竞赛的性质及时间规划，可以顺利的报名竞赛、参与竞赛。当学校需要向学生下发所有的竞赛信息列表文件时，管理员可以在系统中导出需要展示的竞赛，以文件的形式下发给线下所有学生或教师。如图 47所示。

图 47管理员导出竞赛信息功能时序图

4.2.2 学生用户端功能设计

学生用户端功能模块如图 48学生端功能模块图所示。

图 48学生端功能模块图

学生用户端的功能结构设计与管理端有所区别，其包括用户信息管理、通知公告、竞赛信息、竞赛小队管理、竞赛文件管理、荣誉信息管理和赛事宣传共计7个模块。相对于管理端，学生用户仅拥有自己的数据权限，对于竞赛信息管理模块仅有查看的权限，而在竞赛小队管理、竞赛文件管理、荣誉信息管理模块中，学生用户只能对自己所在小队内的数据进行操作。另外，在竞赛小队管理模块中，学生用户拥有报名竞赛的功能；而在赛事宣传和通知公告模块中，学生用户仅有查看权限。

通知公告功能的设计目标是确保学生和教师能快速了解到所参与竞赛的最新动向，管理员会不定期发布最新的竞赛政策、活动信息等重要通知。管理员具有权限执行添加、修改、搜索、删除及查看等操作。这使得管理员有能力发布新的通知和公告，同时也能根据实际需求修改或删除已发布的信息。此外，管理员亦可利用搜索或查看功能，审查已发布或已关闭的通知和公告的相关信息。

学生用户和教师用户都可以在首页和通知公告页面上查看最新的通知公告，如图 49所示。这些公告以列表形式展示，每条公告的标题、发布时间以及简要内容都清晰可见，用户只需点击标题就能查看公告的全文内容。此外，用户还可以根据关键词搜索特定的公告，方便用户找到自己感兴趣或需要了解的信息。

图 49学生查看通知公告功能流程图

竞赛队伍管理的设计是为了能够使学生更便捷的报名参加比赛。在这个功能模块中设计了报名竞赛、管理竞赛队伍等功能，同时提供了竞赛队伍相关信息如竞赛文件管理、荣誉信息管理的功能，方便对竞赛队伍信息进行全面管理。

学生可以创建新的竞赛小队，并录入小队的基本信息，包括小队名称、成员构成等，如图 410所示。

在小队创建之后，学生用户可以进一步通过修改功能来完善小队信息，例如指定小队的队长、分配指导教师等。完成这些设置后，小队的报名信息将提交给管理员进行审批。只有经过管理员的批准后，小队才能成功报名参赛。

图 410学生报名竞赛功能流程图

荣誉信息管理的主要设计目的是在竞赛结束后，学生可以及时的将参与竞赛所获得的奖项/成果进行存档，如图 411所示，以便学生在后续的学习生活如：奖学金申请、留学申请等情况下可以及时的从系统中调取信息。同时也方便了管理员对于荣誉信息的管理，以便在日后的竞赛事宣传中能够获取到往届学生的获奖信息，宣传给之后报名参赛的学生，是竞赛可持续化进行。

其核心功能主要包括荣誉信息的添加、修改、查询和删除。一旦荣誉信息被添加进系统，管理员将进行审核，以确保信息的准确性和完整性。

图 411学生添加荣誉信息流程图

4.2.3 教师用户端功能设计

教师用户端的功能架构设计在很大程度上与学生端相似，然而在竞赛小队管理中，教师用户无权报名竞赛、修改竞赛队伍，或删除竞赛队伍。这种设计出于对教师用户身份特性和权限定位的考量，即，教师的主要职责在于指导学生参赛，而不是直接参与竞赛。所有这些设定的目标均为确保系统的权限管理顺畅、有序，并与实际需求相吻合，从而提高竞赛管理系统的效率与实用性。

教师端端功能模块图如图 412所示。

图 412教师端功能模块图

竞赛文件管理的设计目的是为了给参与竞赛的队伍提供一个信息交流、分享的平台。这个模块的核心功能包括文件的上传、下载、修改、查询以及删除。竞赛队伍中的用户可以随时上传、下载参与的竞赛相关文件，进行数据共享。同时也方便了管理员对于系统中文件信息的管理，及时的清理存储空间。

图 413教师下载竞赛文件功能流程图

4.3 类和对象的设计

经过设计，绘制系统内的整体功能模块的实体类图如图 414所示。

图 414系统实体类图

4.4 动态模型设计

竞赛管理系统的核心流程主要体现在一个完整的竞赛参与流程中，以下对这个过程进行详细描述：
首先，在竞赛启动阶段，管理员在赛事宣传模块中发布最新即将开始或已经开启报名的竞赛信息。

指 标
疑似剽窃文字表述

1. Controller接收到用户的请求后，根据请求的内容，Controller会决定调用哪个Model来处理这个请求。Model接收到请求后，进行相应的

相似文献列表

去除本人文献复制比：4.3%(511) 文字复制比：4.3%(511) 疑似剽窃观点：(0)

1	20162153_毕成铭_教室管理系统的设计与实现	2.9% (344)
	毕成铭 - 《大学生论文联合比对库》 - 2020-05-20	是否引证：否
2	校园车辆管理系统	1.0% (121)
	马娟娟 - 《大学生论文联合比对库》 - 2020-05-29	是否引证：否
3	美食浏览系统的设计与实现	0.3% (30)
	段合 - 《大学生论文联合比对库》 - 2014-05-29	是否引证：否

原文内容

这些信息将在系统中进行展示，使得学生用户可以了解最近的竞赛活动，从而自主决定是否参与其中。

接下来，如果学生用户决定参与某项竞赛，他们可以在竞赛队伍管理模块中，根据预定的竞赛组建自己的竞赛小队，并进行报名操作，如图 415所示。学生在报名过程中，需要输入竞赛队伍的基本信息，并上传竞赛报名申请表。

图 415学生报名参加竞赛时序图

提交报名申请后，管理员用户会收到报名信息，进行审核。一旦审核通过，该竞赛队伍就成功报名了该项竞赛，如图 416所示。

图 416管理员审批竞赛队伍时序图

在参赛过程中，竞赛队伍可能会产生各种竞赛过程文件。在这种情况下，竞赛队伍的成员可以在竞赛文件管理模块中上传和下载相关文件，并进行管理操作。如图 417所示。

图 417教师用户上传竞赛文件时序图

最终，竞赛结束后，竞赛小队的成员可以在荣誉信息管理模块中，根据他们在竞赛中取得的奖项和成果进行信息上传。完成上传后，管理员会对这些荣誉信息进行审核。一旦审核通过，这些荣誉信息就会被存档到系统中。若审核失败，则需重新上传或删除相关荣誉信息。如图 418所示。

动态模型设计的优势主要在于其能够以流程视角捕捉系统运行时的行为，从而有效地反映出系统实体间的相互作用和协同工作的过程。这样的设计方式能使系统设计者和用户更加清晰地理解系统运行的过程，以及各功能模块如何协同工作来实现系统的总体目标。通过对动态行为的细致描绘，动态模型设计有助于发现系统设计中可能存在的问题，并提供改进的机会。

总体而言，这个动态模型为理解和参与竞赛的全过程提供了一种有序且详细的视角，充分体现了各功能模块如何协作以支持竞赛管理的整体目标。

图 418学生添加荣誉信息管理员审批时序图

4.5 数据库设计

根据需求分析中所提取的需求，展示与主要功能相关的9张数据表，各数据库表的名称、属性、类型、默认值以及主键的设计情况展示如表格 41至表格 49。

表格 41用户信息表sys_user

中文名称	英文名称	类型长度	主键	空值	默认值	备注
用户id	user_id	bigint	Y	N	无	自动递增
用户账号	user_name	varchar	N	N	无	无
用户昵称	nick_name	varchar(30)	N	Y	无	无
用户类型	user_type	varchar(2)	N	Y	无	无
用户邮箱	email	varchar(50)	N	Y	无	无
手机号码	phonenumber	varchar(11)	N	Y	无	无
用户性别	sex	char	N	Y	0	无

中文名称英文名称类型长度主键空值默认值备注

用户id user_id bigint Y N 无自动递增

用户账号 user_name varchar N N 无无

用户昵称 nick_name varchar(30) N Y 无无

用户类型 user_type varchar(2) N Y 无无

用户邮箱 email varchar(50) N Y 无无

手机号码 phonenumber varchar(11) N Y 无无

用户性别 sex char N Y 0 无

(续)

中文名称	英文名称	类型长度	主键	空值	默认值	备注
密码	password	varchar(100)	N	Y	123456	无
帐号状态	status	char	N	Y	无	无
部门id	dept_id	bigint	N	Y	无	无

中文名称英文名称类型长度主键空值默认值备注

密码 password varchar(100) N Y 123456 无

帐号状态 status char N Y 无无
部门id dept_id bigint N Y 无无
表格 42院系信息表sys_dept

中文名称	英文名称	类型长度	主键	空值	默认值	备注
院系id	dept_id	bigint	Y	N	无	自动递增
父院系id	parent_id	bigint	N	Y	0	无
祖级列表	ancestors	varchar(50)	N	Y	无	无
院系名称	dept_name	varchar(30)	N	Y	无	无
负责人	leader	varchar	N	Y	无	无
联系电话	phone	varchar(11)	N	Y	无	无
邮箱	email	varchar(50)	N	Y	无	无
院系状态	status	char	N	Y	0	无

中文名称英文名称类型长度主键空值默认值备注
院系id dept_id bigint Y N 无自动递增
父院系id parent_id bigint N Y 0 无
祖级列表 ancestors varchar(50) N Y 无无
院系名称 dept_name varchar(30) N Y 无无
负责人 leader varchar N Y 无无
联系电话 phone varchar(11) N Y 无无
邮箱 email varchar(50) N Y 无无
院系状态 status char N Y 0 无
表格 43角色信息表sys_role

中文名称	英文名称	类型长度	主键	空值	默认值	备注
角色ID	role_id	bigint	Y	N	无	自动递增
角色名称	role_name	varchar(30)	N	N	无	无
角色权限字符串	role_key	varchar(100)	N	N	无	无
数据范围	data_scope	char	N	Y	1	无
菜单树选择项是否关联显示	menu_check_strictly	tinyint	N	Y	1	无
部门树选择项是否关联显示	dept_check_strictly	tinyint	N	Y	1	无
角色状态	status	char	N	N	无	无

中文名称英文名称类型长度主键空值默认值备注
角色ID role_id bigint Y N 无自动递增
角色名称 role_name varchar(30) N N 无无
角色权限字符串 role_key varchar(100) N N 无无
数据范围 data_scope char N Y 1 无
菜单树选择项是否关联显示 menu_check_strictly tinyint N Y 1 无
部门树选择项是否关联显示 dept_check_strictly tinyint N Y 1 无
角色状态 status char N N 无无
表格 44通知公告表sys_notice

中文名称	英文名称	类型长度	主键	空值	默认值	备注
公告ID	notice_id	bigint	Y	N	无	自动递增
公告标题	notice_title	varchar(50)	N	N	无	无
公告类型	notice_type	char	N	N	0	无
公告内容	notice_content	longblob	N	Y	无	无
公告状态	status	char	N	Y	无	无

中文名称英文名称类型长度主键空值默认值备注
公告ID notice_id bigint Y N 无自动递增
公告标题 notice_title varchar(50) N N 无无
公告类型 notice_type char N N 0 无
公告内容 notice_content longblob N Y 无无
公告状态 status char N Y 无无
表格 45竞赛信息表sys_comp

中文名称	英文名称	类型长度	主键	空值	默认值	备注
竞赛id	comp_id	bigint	Y	N	无	自动递增
竞赛名称	comp_name	varchar	N	N	无	无
竞赛类型	comp_type	varchar	N	Y	无	无
主办方	comp_sponser	varchar	N	Y	无	无
竞赛链接	comp_link	varchar	N	Y	无	无
开始时间	comp_start_time	datetime	N	Y	无	无
结束时间	comp_end_time	datetime	N	Y	0	无
竞赛状态	status	char	N	Y	无	无

中文名称英文名称类型长度主键空值默认值备注

竞赛id comp_id bigint Y N 无自动递增
竞赛名称 comp_name varchar N N 无无
竞赛类型 comp_type varchar N Y 无无
主办方 comp_sponser varchar N Y 无无
竞赛链接 comp_link varchar N Y 无无
开始时间 comp_start_time datetime N Y 无无
结束时间 comp_end_time datetime N Y 0 无
竞赛状态 status char N Y 无无
表格 46竞赛队伍表sys_squad

中文名称	英文名称	类型长度	主键	空值	默认值	备注
队伍id	squad_id	bigint	Y	N	无	自动递增
队伍名称	squad_name	varchar	N	N	无	无
队伍组长	squad_leader	varchar	N	Y	创建人	无
队伍人数	member_num	varchar	N	Y	无	无
队伍成员	squad_member	bigint	N	Y	无	无
指导教师	squad_teacher	varchar	N	Y	无	无
联系电话	phone	varchar(11)	N	Y	0	无
邮箱	email	varchar(50)	N	Y	无	无
队伍状态	status	char	N	Y	无	无

中文名称英文名称类型长度主键空值默认值备注
队伍id squad_id bigint Y N 无自动递增
队伍名称 squad_name varchar N N 无无
队伍组长 squad_leader varchar N Y 创建人无
队伍人数 member_num varchar N Y 无无
队伍成员 squad_member bigint N Y 无无
指导教师 squad_teacher varchar N Y 无无
联系电话 phone varchar(11) N Y 0 无
邮箱 email varchar(50) N Y 无无
队伍状态 status char N Y 无无
表格 47竞赛文件表sys_file_info

中文名称	英文名称	类型长度	主键	空值	默认值	备注
文件id	file_id	bigint	Y	N	无	自动递增
文件名	file_name	varchar	N	Y	无	无
文件路径	file_path	varchar	N	Y	无	无
文件类型	file_type	varchar	N	Y	无	无
竞赛队伍id	squad_id	bigint	N	Y	无	无

中文名称英文名称类型长度主键空值默认值备注
文件id file_id bigint Y N 无自动递增
文件名 file_name varchar N Y 无无
文件路径 file_path varchar N Y 无无
文件类型 file_type varchar N Y 无无
竞赛队伍id squad_id bigint N Y 无无
表格 48荣誉信息表sys_honor

中文名称	英文名称	类型长度	主键	空值	默认值	备注
荣誉id	honor_id	bigint	Y	N	无	自动递增
荣誉名	honor_name	varchar(50)	N	N	无	无
荣誉类型	honor_type	varchar	N	N	无	无
荣誉内容	honor_content	longblob	N	Y	无	无
荣誉获得时间	honor_date	datetime	N	Y	无	无
荣誉状态	status	char	N	Y	0	无
竞赛队伍id	squad_id	bigint	N	Y	无	无

中文名称英文名称类型长度主键空值默认值备注
荣誉id honor_id bigint Y N 无自动递增
荣誉名 honor_name varchar(50) N N 无无
荣誉类型 honor_type varchar N N 无无
荣誉内容 honor_content longblob N Y 无无
荣誉获得时间 honor_date datetime N Y 无无
荣誉状态 status char N Y 0 无
竞赛队伍id squad_id bigint N Y 无无
表格 49赛事宣传表sys_public

中文名称	英文名称	类型长度	主键	空值	默认值	备注
宣传id	public_id	bigint	Y	N	无	自动递增
宣传标题	public_title	varchar	N	N	无	无

宣传图	public_pic	varchar	N	N	无	无
宣传内容	public_content	longblob	N	Y	无	无
宣传状态	status	char	N	Y	无	无

中文名称 英文名称 类型 长度 主键 空值 默认值 备注

宣传id public_id bigint Y N 无 自动递增

宣传标题 public_title varchar N N 无 无

宣传图 public_pic varchar N N 无 无

宣传内容 public_content longblob N Y 无 无

宣传状态 status char N Y 无 无

4.6 本章小结

在本章中，我们将需求分析阶段的成果进行了深化，并将其转化为可以理解和实现的具体设计方案。我们利用时序图对系统的动态行为进行了细致描绘，此举不仅明晰了各模块之间的交互顺序，同时也为后续的系统实现阶段构建了坚实的基础。

5 系统实现

5.1 系统开发环境配置

在开始本项目的开发之前，首先需要进行以下系统环境配置：

1. Java开发环境：安装JDK >= 1.8。因为后端服务主要是基于Java进行开发的，而JDK是运行Java程序的基本环境。
2. 数据库环境：安装MySQL >= 5.7.0作为数据库系统。同时，安装Navicat作为数据库管理工具，以便更方便地查看和管理数据库。

3. ORM框架：安装MyBatis 作为自定义SQL、存储过程和高级映射的持久层框架，能够方便地操作数据库。

4. 缓存环境：安装Redis >= 3.0作为缓存服务器，提高系统的性能。

5. 项目管理工具：安装Maven >= 3.0以方便依赖管理和项目构建。

6. 前端开发环境：安装Node.js >= 12的Vue CLI。

7. 微服务管理和配置：安装Nacos >= 2.x.x进行服务的注册和发现，以及配置管理。

8. 流量控制和服务降级：安装Sentinel >= 1.6.0进行流量控制和服务降级。

9. Spring Boot 和 Spring Cloud：系统采用Spring Boot进行微服务的开发，并使用Spring Cloud作为微服务框架。

5.2 系统整体结构

本文所述的竞赛管理系统采用了先进的前后端分离设计架构进行实现。在这种架构中，前端使用了Vue框架进行开发，后端则使用了Spring Cloud、Spring Boot以及MyBatis的组合模式进行构建。

项目前端结构如图 51所示。后端代码结构如图 52所示。

这一项目实行前后端分离的架构，前端主要通过axios.js来调用后端接口进行数据通信。在微服务的框架下，每个服务都具有各自独立的后端接口，这一特性在图 53中得到了具体展示。同时，为了更加高效地管理和调度这些服务，我们选用了Nacos作为统一的服务管理平台。

通过此种设计架构，可以更好地增强系统的稳定性和扩展性，同时也便于进行分布式开发和管理，有助于提高开发效率和系统运行性能。

图 51前端代码结构

图 52后端代码结构

图 53系统服务端口

5.3 系统功能的实现

5.3.1 用户管理模块——添加用户信息

添加用户信息首先需要用户点击系统管理菜单，当社管理员在前端页面点击用户管理按钮时，会在vue中跳转到显示所有用户信息的user页面，点击页面中的添加按钮el-button组件，配合el-dialog组件会弹出新增用户的弹窗，新增界面如图 54所示。在弹窗内使用el-form-item组件配合el-input组件方便输入用户信息并点击提交按钮，触发一个submit方法，这个方法会调用/api/system/user中的addUser方法向后端发送一个请求。

图 54用户管理新增界面

在填写用户信息时，为确保用户信息录入正确，会对填写内容进行校验，前端代码实现如下：

// 表单校验

```
rules: {
  iletype: [
    { required: true, message: "用户名称不能为空", trigger: "blur" },
    { min: 2, max: 20, message: '用户名称长度必须介于 2 和 20 之间', trigger: 'blur' }
  ],
  iletype: [
    { required: true, message: "用户昵称不能为空", trigger: "blur" }
  ],
  email: [
    {
      type: "email",
      message: "请输入正确的邮箱地址",
      trigger: [ "blur", "change" ]
    }
  ]
}
```



```

],
phoneNumber: [
{
pattern: /^1[3|4|5|6|7|8|9][0-9]\d{8}$/,
message: “请输入正确的手机号码”,
trigger: “blur”
}
]
}
};
},

```

然后，后端接收到请求后，就调用SysUserController类的add方法，这个方法会使用SysUserServiceImpl类的insertUser方法来处理业务逻辑，并使用SysUserMapper类的insertUser方法来操作数据库。代码如下。

```

@RequiresPermissions(“system:user:add”)
@Log(title = “用户管理”, businessType = BusinessType.INSERT)
@PostMapping
public AjaxResult add(@Validated @RequestBody SysUser user)
{
    . . .
    return toAjax(userService.insertUser(user));
}

```

在系统实现的过程中，我们通过在XML文件中执行一条SQL语句，实现了新用户信息的插入到sys_user表的功能。完成数据库插入操作后，系统会返回一个整数值，这个值代表了数据库受影响的行数。如果返回值为1，那么就意味着用户信息成功地插入到了数据库中。

在后端，SysUserController类的add方法会将这个整数值封装为一个Response对象，并返回给前端。当前端接收到这个响应之后，它会首先判断新用户信息是否成功插入。如果成功，前端则会弹出一个message组件，展示“新增成功”的提示信息，并随后重定向回user页面。代码如下：

```

submitForm: function() {
    this.$refs[“form”].validate(valid => {
    if (valid) {
        . . .
    } else {
        addUser(this.form).then(response => {
            this.$modal.msgSuccess(“新增成功”);
            this.open = false;
            this.getList();
        });
    }
    });
},

```

在user页面上，管理员能够看到刚刚添加的新用户信息。这样，我们就成功实现了新用户信息的添加和展示功能。

5.3.2 院系管理模块——查询院系信息

在管理员进入系统的院系管理页面后，他们能够利用查询功能，准确地定位到所需要操作的院系。在院系列表的上方，我们配置了一个院系信息搜索框，如图 55所示，这个搜索框通过使用el-form组件，并配合el-input和el-select组件，使得管理员可以输入院系名称或者从下拉菜单中选择院系状态，以便快速搜索到特定院系。被查询到的院系信息将会以列表形式展示在页面下方，方便管理员进行后续操作。

图 55查询院系信息界面

查询院系信息的前端代码如下：

```

<el-form :model=“queryParams” ref=“queryForm” size=“small” :inline=“true” v-show=“showSearch” >
<el-form-item label=“院系名称” prop=“deptName” >
<el-input
v-model=“queryParams.deptName”
placeholder=“请输入院系名称”
clearable
@keyup.enter.native=“handleQuery” />
</el-form-item>
<el-form-item label=“状态” prop=“status” >
<el-select v-model=“queryParams.status” placeholder=“院系状态” clearable>
<el-option
v-for=“dict in dict.type.sys_normal_disable”

```

```

:key=" dict.value"
:label=" dict.label"
:value=" dict.value" >
</el-select>
</el-form-item>
<el-form-item>
<el-button type=" primary" icon=" el-icon-search" size=" mini" @click=" handleQuery" >搜索</el-button>
<el-button icon=" el-icon-refresh" size=" mini" @click=" resetQuery" >重置</el-button>
</el-form-item>
</el-form>

```

要实现查询院系的功能，首先需要设置查询参数，代码如下：

```

queryParams: {
  deptName: undefined,
  status: undefined
},

```

当管理员输入或选择了所需查询的院系信息后，只需点击搜索按钮，系统就会根据查询参数重新调用后端的 SysDeptController 类的 list 方法，这个方法会调用 SysDeptServiceImpl 类的 selectDeptTreeList 方法来处理业务逻辑，再使用 SysDeptMapper 类的 selectDeptTreeList 方法在数据库中查询相关数据，返回查询结果。

需要特别提及的是，院系管理模块中，院系信息在数据库中的组织方式是树状结构。在构建院系列表时，我们优先构建父节点，然后构建其子节点。因此，在进行院系搜索时，系统会同时展示该院系下所有的子节点院系信息。这样的设计，可以提供更全面的院系结构信息，有助于管理员进行准确的管理操作。代码如下：

```

public List<SysDept> buildDeptTree(List<SysDept> depts)
{
  List<SysDept> returnList = new ArrayList<SysDept>();
  List<Long> tempList = depts.stream().map(SysDept::getDeptId).collect(Collectors.toList());
  for (SysDept dept : depts)
  {
    // 如果是顶级节点，遍历该父节点的所有子节点
    if (!tempList.contains(dept.getParentId()))
    {
      recursionFn(depts, dept);
      returnList.add(dept);
    }
  }
  if (returnList.isEmpty())
  {
    returnList = depts;
  }
  return returnList;
}

```

5.3.3 角色管理模块——分配用户

管理员进入角色管理模块，后选择可以对每一个角色进行分配用户的操作。分配用户界面如图 56 所示。

图 56 分配用户界面

管理员选择想要分配用户的角色后点击更多下拉按钮，点击分配用户按钮。这里的更多下拉列表采用了 el-dropdown 组件，并利用 el-dropdown-menu 设置下拉菜单。代码如下：

```

<el-dropdown size=" mini" @command=" (command) => handleCommand(command, scope.row)" v-
hasPermi=" [ 'system:role:edit' ]" >
  <el-button size=" mini" type=" text" icon=" el-icon-d-arrow-right" >更多</el-button>
  <el-dropdown-menu slot=" dropdown" >
    <el-dropdown-item command=" handleDataScope" icon=" el-icon-circle-check"
v-hasPermi=" [ 'system:role:edit' ]" >数据权限</el-dropdown-item>
    <el-dropdown-item command=" handleAuthUser" icon=" el-icon-user"
v-hasPermi=" [ 'system:role:edit' ]" >分配用户</el-dropdown-item>
  </el-dropdown-menu>
</el-dropdown>

```

下拉列表会监听点击菜单项触发的事件回调，并根据被点击的菜单项触发对应的功能。代码如下：

```

handleCommand(command, row) {
  switch (command) {
    case "handleDataScope":
      this.handleDataScope(row);
      break;
  }
}

```

```

case "handleAuthUser" :
this.handleAuthUser(row);
break;
default:
break;
}
},

```

当点击到分配用户按钮后，handleAuthUser方法会根据当前行的roleId，采用动态路由的方式，导向对应角色的用户分配界面，下列带码是动态路由的实现：

```

{
path: '/system/role-auth',
component: Layout,
hidden: true,
permissions: [ 'system:role:edit' ],
children: [
{
path: 'user/:roleId(\\d+)',
component: () => import( '@views/system/role/authUser' ),
name: 'AuthUser',
meta: { title: '角色用户管理', activeMenu: '/system/role' }
}
]
},

```

进入到分配用户界面后，页面会调用SysRoleController类的allocatedList方法构建已分配该角色的用户列表。allocatedList方法会调用SysUserServiceImpl类的selectAllocatedList方法，并通过SysUserMapper在数据库中查询已经分配该角色的用户列表。同样的逻辑也适用于角色添加用户功能，使得系统可以展示尚未分配到该角色的用户列表。

5.3.4 通知公告模块——查看通知公告

系统中的通知公告会放置在首页利用el-card配合el-collapse折叠面板进行展示，，用户点击感兴趣的通知公告标题后，折叠面板会展开，完整展示通知公告的内容。如图 57所示。

图 57查看通知公告界面展示

在首页中首先我们需要引用通知公告接口，同时，会定义一个created函数，在组件创建时调用一个listNotice方法，该方法会发送一个请求到后端的/api/system/notice/list接口。后端的SysNoticeController类接收到前端传来的请求，再通过服务层和数据层查询到前台需要的数据后给前端返回结果。才能将所有通知公告文件以列表的形式展现出来。

5.3.5 竞赛信息管理模块——导出竞赛信息

管理员在进入竞赛信息管理界面后，界面如图 58所示。

图 58竞赛信息管理界面

管理员可以通过勾选需要导出的竞赛数据，然后点击导出按钮。在Vue中，这一操作会调用一个名为handleExport的方法。该方法会向后端的 /api/system/comp/export 接口发送一个请求，同时传递查询条件。代码如下：

```

handleExport() {
this.download( 'system/management/export' , {
...this.queryParams
}, `management_${new Date().getTime()}.xlsx`)
}
}
};

```

在后端，SysCompController类接收到该请求后，将调用其export方法。在调用这个方法之前，我们使用@PreAuthorize注解对接口进行拦截，并通过PermissionService的hasPermi方法检查请求者的权限。

接下来，export方法会调用SysCompServiceImpl类的selectCompList方法，这个方法负责封装业务逻辑。然后，再调用SysCompMapper类的selectSysCompList方法，该方法在XML文件中执行相应的SQL语句，查询数据库中的竞赛信息。

查询完成后，selectSysCompList方法返回一个List<SysComp>对象，这代表查询到的竞赛信息列表，包括竞赛编号、竞赛类型、竞赛时间、竞赛赞助商等内容。SysCompServiceImpl类的selectSysCompList方法将竞赛数据存储为List格式。

接着，SysCompController类的export方法利用ExcelUtil工具类的exportExcel方法，将这个List对象转换为一个Excel对象。该方法设置文件名、表头、列名等属性，再将Excel对象写入到输出流中，并设置响应头和响应类型，然后返回给前端。代码如下：

```

public void exportExcel(HttpServletResponse response) {
try {
writeSheet();
wb.write(response.getOutputStream());
}
catch (Exception e)
log.error( "导出Excel异常{}" , e.getMessage());
}

```



```
finally
IOUtils.closeQuietly(wb);
}
```

前端在接收到响应后，会弹出一个下载窗口，此时管理员可以看到导出的Excel文件。导出后的文件如图 59所示。

图 59竞赛信息导出文件

5.3.6 竞赛队伍管理模块——报名竞赛

学生在进入竞赛队伍管理界面后，点击报名竞赛按钮，页面里通过使用el-dialog配合visible.sync="openCreate" 实现弹出添加竞赛队伍页面，如图 510所示。

图 510报名竞赛功能界面

报名竞赛按钮通过v-hasRole实现了角色的权限管理，只有管理员和学生用户可以进行操作。代码如下：

```
import store from '@store'
export default {
  inserted(el, binding, vnode) {
    const { value } = binding
    const super_admin = "admin";
    const roles = store.getters && store.getters.roles
    if (value && value instanceof Array && value.length > 0) {
      const roleFlag = value
      const hasRole = roles.some(role => {
        return super_admin === role || roleFlag.includes(role)
      })
      if (!hasRole) {
        el.parentNode && el.parentNode.removeChild(el)
      }
    } else {
      throw new Error(` 请设置角色权限标签值`)
    }
  }
}
```

在弹出页面，学生需要填写竞赛队伍基本信息、添加竞赛队伍成员、上传竞赛报名表。这个页面使用el-form组件来收集用户的输入。在用户输入信息完成后，前端会调用/api/system/squad中的addSquad方法向后端发送一个请求，后端接收到请求后，调用SysSquadController类的add方法，这个方法会调用SysSquadServiceImpl类的insertSysSquad、insertSysFileInfo、addAuthUser;方法来新增竞赛队伍、队伍成员、竞赛报名表等信息，

在报名竞赛的过程中，由于竞赛队伍需要管理员的审批，所以新组建的竞赛队伍的默认状态为组队中，当管理员审批后，竞赛队伍状态才会改变。

4. 基于Web的竞赛管理系统的设计与实现_第4部分			总字数：9137
相似文献列表			
去除本人文献复制比：11.1%(1010) 文字复制比：11.1%(1010) 疑似剽窃观点：(0)			
1	12177405205_邓佳阳_老年公寓管理系统的设计与实现	5.8% (531)	
	邓佳阳 - 《大学生论文联合比对库》 - 2021-05-10		是否引证：否
2	矿石品位检测系统设计与开发	1.8% (162)	
	杨雨柔 - 《大学生论文联合比对库》 - 2021-06-03		是否引证：否
3	基于CS架构和PB技术的人大建议督办系统的设计与实现	1.5% (137)	
	施欣好 - 《大学生论文联合比对库》 - 2018-05-30		是否引证：否
4	LX公司远程培训管理系统设计	1.2% (112)	
	张锦雷(导师：冯秀珍) - 《北京工业大学硕士论文》 - 2014-04-01		是否引证：否
5	05_111111401026_李儒华	0.4% (32)	
	李儒华 - 《大学生论文联合比对库》 - 2015-05-20		是否引证：否
6	Linux应用程序进程pmu性能事件检测分析系统	0.3% (31)	
	巩学超(导师：马军) - 《山东大学硕士论文》 - 2015-04-20		是否引证：否
7	小蜜蜂旧书城网上书店系统的设计与实现	0.3% (31)	
	齐思远 - 《大学生论文联合比对库》 - 2014-05-06		是否引证：否
原文内容			

5.3.7 竞赛文件管理模块——上传竞赛文件

随着竞赛的进行，小队成员们由于分工不同等其他原因会产生不同的阶段性成果文件，这时用户就可以通过竞赛文件管理模块中的上传竞赛文件功能将自己的材料等文件上传到系统中与队内成员共享。

在用户进入竞赛文件管理模块后，可以看到本竞赛小队内的竞赛文件列表，用户点击上传按钮后会弹出添加竞赛文件窗口，如图 511所示。

图 511上传竞赛文件功能界面

在此页面中利用了file-upload组件来完成文件的上传功能。在此组件中设置了文件大小、文件类型等参数来控制用户上传的文件，保证系统的安全性与稳定性。下面是file-upload组件的实现方法：

```
<el-upload
multiple
:action=" uploadFileUrl"
:before-upload=" handleBeforeUpload"
:file-list=" fileList"
:limit=" limit"
:on-error=" handleUploadError"
:on-exceed=" handleExceed"
:on-success=" handleUploadSuccess"
:show-file-list=" false"
:headers=" headers"
class=" upload-file-uploader"
ref=" fileUpload" >
<!-- 上传按钮
<el-button size=" mini" type=" primary">选取文件</el-button>
<!-- 上传提示
<div class=" el-upload__tip" slot=" tip" v-if=" showTip" >
请上传
<template v-if=" fileSize" > 大小不超过 <b style=" color: #f56c6c" >{{ fileSize }}MB</b> </template>
<template v-if=" iletype" > 格式为 <b style=" color: #f56c6c" >{{ iletype.join( "/" ) }}</b> </template>
的文件
</div>
</el-upload>
```

用户在选择上传文件后，系统首先会检查文件的大小和类型是否符合要求。如果满足条件，系统会向后端的SysFileController发起请求。SysFileController接收到请求后，会调用IsysFileServiceImpl的uploadFile方法，将文件上传并返回文件路径给前端。

当用户点击提交按钮，前端会将返回的文件路径和所选的文件类型数据一并发送到后端的SysFileInfoController。SysFileInfoController接收到请求后，会调用FileUtils类中的方法对文件路径进行切片，以此获取文件名。具体获取文件名的方法如下。

```
public static String getName(String filePath)
{
    if (null == filePath)
    {
        return null;
    }
    int len = filePath.length();
    if (0 == len)
    {
        return filePath;
    }
    if (isFileSeparator(filePath.charAt(len - 1)))
    {
        len--;
    }
    int begin = 0;
    char c;
    for (int I = len - 1; I > -1; i--)
    {
        c = filePath.charAt(i);
        if (isFileSeparatorI)
        {
            begin = I + 1;
        }
    }
}
```

```

break;
}
}
return filePath.substring(begin, len);
}

```

最后，IsysFileInfoServiceImpl会调用SysFileInfoMapper，将文件信息写入数据库。需要注意的是，文件信息以文件路径的方式存储在数据库中。这样，我们就成功实现了文件的上传与共享，提升了团队之间的合作效率。

5.3.8 荣誉信息管理模块——添加荣誉信息

在竞赛结束后，一些竞赛队伍会获得奖项或者产生一些成果。这时竞赛队伍中的成员就可以在荣誉信息管理列表中添加相关的荣誉信息。添加界面如图 512所示。

图 512荣誉信息添加界面

学生在进入荣誉信息管理界面后点击添加荣誉信息，输入他们在竞赛中所取得的奖项或成果名称，选择所在的竞赛队伍、参与的竞赛后就可以对其获得的荣誉进行撰写、上传荣誉证书等。本模块运用了editor富文本组件，它不仅满足了文字输入的需求，也支持上传荣誉证书等成果展示图片。富文本组件的工具栏的实现方法如下：

```

modules: {
// 工具栏配置
toolbar: [
[ "bold", "italic", "underline", "strike" ], // 加粗斜体下划线删除线
[ "blockquote", "code-block" ], // 引用代码块
[ { list: "ordered" }, { list: "bullet" } ], // 有序、无序列表
[ { indent: "-1" }, { indent: "+1" } ], // 缩进
[ { size: [ "small", false, "large", "huge" ] } ], // 字体大小
[ { header: [ 1, 2, 3, 4, 5, 6, false ] } ], // 标题
[ { color: [] }, { background: [] } ], // 字体颜色、字体背景颜色
[ { align: [] } ], // 对齐方式
[ "clean" ], // 清除文本格式
[ "link", "image", "video" ] // 链接、图片、视频
],
},
placeholder: "请输入内容",
readOnly: this.readOnly,
},

```

本模块的荣誉信息新增方法与其他模块的新增功能实现方法类似，不过有一个明显的区别，那就是在富文本编辑器中输入的内容以及上传图片的文件地址会以String类型存储到longblob容器中。这种实现方式可以有效地存储大量的文本和图片信息，满足了荣誉信息管理的需求。

5.3.9 赛事宣传模块——设置首页展示图

赛事宣传的展示功能放置在首页上，首页赛事宣传图采用了el-carousel走马灯组件进行动态展示，首页界面如图 513所示。

图 513竞赛宣传界面

当管理员发布赛事宣传后，他们可以根据宣传活动的重要性和紧急程度，决定是否将该赛事宣传置于首页宣传图的位置进行展示。若需要设置为首页宣传图，管理员可通过点击对应赛事宣传的修改按钮以修改赛事宣传状态。

管理员点击前端页面的修改按钮后，会在vue中跳转至赛事宣传详情页面。该页面利用了el-dialog组件，并结合visible.sync="open"实现弹出式的修改页面，同时传递赛事宣传编号publicId作为参数。代码如下。

```

/** 修改按钮操作 */
handleUpdate(row) {
this.reset();
const publicId = row.publicId || this.ids
getPublic(publicId).then(response => {
this.form = response.data;
this.open = true;
this.title = "修改赛事宣传";
});
},

```

一旦管理员完成修改并点击提交按钮，前端将携带publicId调用/system/api/public中的updatePublic方法，向后端发起请求。后端的SysPublicController类收到请求后，将调用edit方法进行数据修改，服务层的IsysPublicServiceImpl类进一步调用数据层的SysPublicMapper进行数据库更新。

数据库更新后，首页会重新调用system/api/public中的list方法，从数据库中检索出赛事宣传状态为首页宣传图的项目，并获取其图片路径进行展示。这样，系统就能够动态地更新和展示首页的赛事宣传图，提升了宣传活动的效率和可见度。

5.4 系统测试

5.4.1 测试目的及测试方法

系统测试是在开发过程中的一个关键步骤，它的主要目的是确保软件系统在各种预期的使用环境和条件下均能正常工作

，以及能满足所有的业务需求和技术规格。系统测试的主要目标包括但不限于：

- 1. 验证软件系统的功能完整性：确保系统实现了所有预定的功能，并且每个功能都能正常工作。
- 2. 验证系统的安全性：确保系统能够有效地防止未授权的访问和操作，保护用户的数据安全。

常见的测试方法有两种，一种是黑盒测试，一种是白盒测试。黑盒测试也叫做规格说明测试，它的特点是不考虑代码，根据规格说明来选择测试用例。白盒测试也叫做逻辑驱动测试，它的特点是不考虑规格说明，根据代码来选择测试用例。

5.4.2 测试方案及测试用例

由于本系统涵盖的功能模块众多，受限于篇幅，无法逐一展示所有功能的测试用例。因此，将在本节中挑选在系统设计和实现过程中涉及的主要功能模块，为这些模块设计具体的测试用例并进行详述。

5.4.2.1 用户管理模块

方案：主要提供用户信息管理模块中，添加用户信息功能的测试用例。管理员登录后，点击上方菜单栏的系统管理管理下的用户管理功能，点击“添加”按钮，页面弹出“新增用户”界面，输入用户信息。点击“提交”按钮即可完成添加工作。

测试用例如表格 51所示。

表格 51添加用户功能测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	添加用户信息	点击“添加”，不填写任何信息，点击“提交”按钮。	弹出添加失败信息	同预期结果
2		点击“添加”按钮，填写所有信息，点击“提交”按钮。	添加框关闭，列表刷新，显示添加用户	同预期结果
3		点击“添加”按钮，填写所有信息，点击“取消”按钮。	添加框关闭，列表信息未变化	同预期结果
4		点击“关闭”按钮。	弹出框关闭，列表信息未变化	同预期结果

测试序号测试项测试数据预期结果测试结果

- 1 添加用户信息点击“添加”，不填写任何信息，点击“提交”按钮。 弹出添加失败信息同预期结果
- 2 点击“添加”按钮，填写所有信息，点击“提交”按钮。 添加框关闭，列表刷新，显示添加用户同预期结果
- 3 点击“添加”按钮，填写所有信息，点击“取消”按钮。 添加框关闭，列表信息未变化同预期结果
- 4 点击“关闭”按钮。 弹出框关闭，列表信息未变化同预期结果

5.4.2.2 院系管理模块

方案：主要提供院系信息管理模块中，院系信息查询功能的测试用例。管理员点击系统管理菜单中的院系管理功能，在搜索框中输入院系名称或选择院系状态，点击“搜索”按钮，页面中显示查询到的院系列表。点击“重置”按钮后，显示全部院系列表即查询成功。测试用例如表格 52所示。

表格 52查询院系信息测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	查询院系信息	输入正确的院系名，点击搜索	查询到院系及其子院系列表	同预期结果

测试序号测试项测试数据预期结果测试结果

- 1 查询院系信息输入正确的院系名，点击搜索查询到院系及其子院系列表同预期结果
- (续)

测试序号	测试项	测试数据	预期结果	测试结果
2	查询院系信息	输入错误的院系名，点击搜索	显示空白列表	同预期结果
3		点击“重置”按钮	搜索框清空，显示所有院系列表	同预期结果

测试序号测试项测试数据预期结果测试结果

- 2 查询院系信息输入错误的院系名，点击搜索显示空白列表同预期结果
- 3 点击“重置”按钮搜索框清空，显示所有院系列表同预期结果

5.4.2.3 角色管理模块

方案：主要提供角色管理模块中，分配用户功能的测试用例。管理员点击系统管理菜单中的角色管理功能，选择要分配用户的角色行，点击更多下拉菜单中的“分配用户”按钮，进入分配用户界面，点击“添加用户”按钮，页面弹出选择用户弹窗，选择要分配的用户后，点击“确定”按钮则用户分配成功。

测试用例如表格 53所示。

表格 53分配用户测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	分配用户	点击“分配用户”按钮	跳转到“分配用户”界面，同时查询到已分配用户列表	同预期结果
2		点击添加按钮，选择用户后，点击“提交”按钮	“选择用户”框关闭，显示添加的用户信息	同预期结果
3		点击添加按钮，不选择用户后，点击“提交”按钮	“选择用户”框关闭，列表信息未变化	同预期结果
4		点击添加按钮，选择用户后，点击“取消”按钮	“选择用户”框关闭，列表信息未变化	同预期结果

测试序号测试项测试数据预期结果测试结果

1 分配用户

点击“分配用户”按钮跳转到“分配用户”界面，同时查询到已分配用户列表同预期结果

- 2 点击添加按钮，选择用户后，点击“提交”按钮 “选择用户”框关闭，显示添加的用户信息同预期结果
- 3 点击添加按钮，不选择用户后，点击“提交”按钮 “选择用户”框关闭，列表信息未变化同预期结果
- 4 点击添加按钮，选择用户后，点击“取消”按钮 “选择用户”框关闭，列表信息未变化同预期结果

5.4.2.4 通知公告模块

方案：主要提供通知公告模块中，查看通知公告功能的测试用例。用户查看到首页的通知公告折叠面板，点击其中一条通知公告，折叠面板展开显示通知公告内容则查看成功。

测试用例如表格 54所示。

表格 54查看通知公告测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	查看通知公告	不点击任何按钮	显示通知公告列表	同预期结果
2		点击“通知公告1”	这一条通知公告折叠面板打开，显示”通知公告1内容	同预期结果
3		点击“通知公告2”	通知面板2面板打开，显示通知公告2内容	同预期结果
4		再次点击“通知公告2 “	通知公告2面板合起，显示通知公告列表	同预期结果

测试序号测试项测试数据预期结果测试结果

- 1 查看通知公告不点击任何按钮显示通知公告列表同预期结果
- 2 点击“通知公告1” 这一条通知公告折叠面板打开，显示”通知公告1内容同预期结果
- 3 点击“通知公告2” 通知面板2面板打开，显示通知公告2内容同预期结果
- 4 再次点击“通知公告2 “ 通知公告2面板合起，显示通知公告列表同预期结果

5.4.2.5 竞赛信息管理模块

方案：主要提供竞赛信息管理模块中，删除竞赛信息功能的测试用例。管理员点击竞赛管理菜单下的竞赛信息管理功能，选择其中一条竞赛信息点击“删除”按钮，在确认删除弹窗中点击“确定”按钮，即删除竞赛信息完成。

测试用例如表格 55所示。

表格 55删除竞赛信息测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	删除竞赛信息	不选择竞赛信息，点击“删除”按钮	无法点击“删除”按钮	同预期结果

测试序号测试项测试数据预期结果测试结果

- 1 删除竞赛信息不选择竞赛信息，点击“删除”按钮无法点击“删除”按钮同预期结果

(续)

测试序号	测试项	测试数据	预期结果	测试结果
2	删除竞赛信息	点击竞赛信息，点击“删除”按钮，点击“确定”按钮	显示删除竞赛信息后的竞赛信息列表	同预期结果
3		点击竞赛信息，点击“删除”按钮，点击“取消”按钮	竞赛信息未被删除，显示所有的竞赛列表	同预期结果

测试序号测试项测试数据预期结果测试结果

- 2 删除竞赛信息点击竞赛信息，点击“删除”按钮，点击“确定”按钮显示删除竞赛信息后的竞赛信息列表同预期结果
- 3 点击竞赛信息，点击“删除”按钮，点击“取消”按钮竞赛信息未被删除，显示所有的竞赛列表同预期结果

5.4.2.6 竞赛队伍管理模块

方案：主要提供竞赛队伍管理模块中，报名竞赛功能的测试用例。学生点击竞赛管理菜单下的竞赛队伍管理功能，点击“报名竞赛”按钮，输入竞赛队伍信息，如：队伍名称、参与竞赛等信息。然后点击添加小队成员的“添加”按钮，输入小队成员的用户id，点击上传竞赛报名表的“添加”按钮，上传竞赛报名表。最后点击“提交”按钮，竞赛报名成功。

测试用例如表格 56所示。

表格 56报名竞赛功能测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	报名竞赛	点击“报名竞赛”按钮，不输入任何信息，点击“提交”按钮	弹出添加失败信息，需要输入必填信息	同预期结果
2		点击“报名竞赛”按钮，输入所有信息，点击“提交”按钮	添加框关闭，列表刷新，显示新添加的小队	同预期结果
3		点击“报名竞赛”按钮，填写所有信息，点击“取消”按钮。	添加框关闭，列表信息未变化	同预期结果
4		点击“关闭”按钮。	弹出框关闭，列表信息未变化	同预期结果

测试序号测试项测试数据预期结果测试结果

- 1 报名竞赛点击“报名竞赛”按钮，不输入任何信息，点击“提交”按钮弹出添加失败信息，需要输入必填信息同预期结果
- 2 点击“报名竞赛”按钮，输入所有信息，点击“提交”按钮添加框关闭，列表刷新，显示新添加的小队同预期结果
- 3 点击“报名竞赛”按钮，填写所有信息，点击“取消”按钮。 添加框关闭，列表信息未变化同预期结果
- 4 点击“关闭”按钮。 弹出框关闭，列表信息未变化同预期结果

5.4.2.7 竞赛文件管理模块

方案：主要提供竞赛文件管理模块中，添加竞赛文件功能的测试用例。学生点击竞赛管理菜单下的竞赛文件管理功能，点击“添加”按钮，点击“选择文件”上传竞赛文件、输入其他文件信息，如：文件类型、参与竞赛等信息。最后点击提交按钮

，竞赛文件添加成功。
测试用例如表格 57所示。
表格 57添加竞赛文件功能测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	添加竞赛文件	点击“添加”按钮，不上传文件，点击“提交”按钮	弹出添加失败信息，需要输入必填信息	同预期结果
2		点击“添加”按钮，输入所有信息，点击“提交”按钮	添加框关闭，列表刷新，显示新添加的竞赛文件	同预期结果
3		点击“添加”按钮，填写所有信息，点击“取消”按钮	添加框关闭，列表信息未变化	同预期结果
4		点击“关闭”按钮。	弹出框关闭，列表信息未变化	同预期结果

测试序号测试项测试数据预期结果测试结果
1 添加竞赛文件点击“添加”按钮，不上传文件，点击“提交”按钮弹出添加失败信息，需要输入必填信息同预期结果
2 点击“添加”按钮，输入所有信息，点击“提交”按钮添加框关闭，列表刷新，显示新添加的竞赛文件同预期结果
3 点击“添加”按钮，填写所有信息，点击“取消”按钮。添加框关闭，列表信息未变化同预期结果
4 点击“关闭”按钮。弹出框关闭，列表信息未变化同预期结果
5.4.2.8 荣誉信息管理模块

方案：主要提供荣誉信息管理模块中，添加荣誉信息功能的测试用例。管理员登录后，点击上方菜单栏的竞赛管理管理下的荣誉信息管理功能，点击“添加”按钮，页面弹出“新增荣誉信息”界面，输入荣誉信息。点击“提交”按钮即可完成添加工作。
测试用例如表格 58所示。
表格 58添加荣誉信息功能测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	添加荣誉信息	点击“添加”按钮，不填写任何信息，点击“提交”按钮。	弹出添加失败信息，需要输入必填信息	同预期结果
2		点击“添加”按钮，填写所有信息，点击“提交”按钮。	添加框关闭，列表刷新，显示添加荣誉信息	同预期结果
3		点击“添加”按钮，填写所有信息，点击“取消”按钮。	添加框关闭，列表信息未变化	同预期结果
4		点击“关闭”按钮。	弹出框关闭，列表信息未变化	同预期结果

测试序号测试项测试数据预期结果测试结果
1 添加荣誉信息点击“添加”按钮，不填写任何信息，点击“提交”按钮。弹出添加失败信息，需要输入必填信息同预期结果
2 点击“添加”按钮，填写所有信息，点击“提交”按钮。添加框关闭，列表刷新，显示添加荣誉信息同预期结果
3 点击“添加”按钮，填写所有信息，点击“取消”按钮。添加框关闭，列表信息未变化同预期结果
4 点击“关闭”按钮。弹出框关闭，列表信息未变化同预期结果
5.4.2.9 赛事宣传模块

方案：主要提供赛事宣传管理模块中，修改赛事宣传状态功能的测试用例。管理员登录后，点击上方菜单栏的竞赛管理管理下的赛事宣传功能，选择要修改的赛事宣传，点击“修改”按钮，页面弹出“修改赛事宣传”界面，修改赛事宣传状态。点击“提交”按钮即可完成添加工作。测试用例如表格 59所示。
表格 59修改赛事宣传状态测试用例

测试序号	测试项	测试数据	预期结果	测试结果
1	修改赛事宣传状态	默认赛事宣传状态为“首页宣传图”，点击“修改”按钮，不选择赛事宣传状态，点击“确定”按钮	修改框关闭，赛事宣传状态为“首页宣传图”，宣传图在首页滚动播放	同预期结果
2		点击“修改”按钮，选择赛事宣传状态为“非首页宣传图”，点击“确定”按钮	修改框关闭，赛事宣传状态变为“非首页宣传图”首页不显示此宣传图	同预期结果

测试序号测试项测试数据预期结果测试结果
1 修改赛事宣传状态默认赛事宣传状态为“首页宣传图”，点击“修改”按钮，不选择赛事宣传状态，点击“确定”按钮修改框关闭，赛事宣传状态为“首页宣传图”，宣传图在首页滚动播放同预期结果
2 点击“修改”按钮，选择赛事宣传状态为“非首页宣传图”，点击“确定”按钮修改框关闭，赛事宣传状态变为“非首页宣传图”首页不显示此宣传图同预期结果
(续)

测试序号	测试项	测试数据	预期结果	测试结果
3		点击“修改”按钮，选择赛事宣传状态为“非首页宣传图”，点击“取消”按钮。	修改框关闭，赛事宣传状态为“首页宣传图”，宣传图在首页滚动播放	同预期结果
4		点击“关闭”按钮。	修改框关闭，列表信息未变化	同预期结果

测试序号测试项测试数据预期结果测试结果
3 点击“修改”按钮，选择赛事宣传状态为“非首页宣传图”，点击“取消”按钮。修改框关闭，赛事宣传状态为“首页宣传图”，宣传图在首页滚动播放同预期结果

4 点击“关闭”按钮。 修改框关闭，列表信息未变化同预期结果

5.5 本章小结

由于篇幅的限制，我们并未能对所有功能的实现进行详细解说，而是选择了几个具有代表性的功能进行深入阐述。本章侧重于详细介绍这些选定功能的实现机制，通过配合前端代码、后端代码、用户界面图像以及文字描述，尽可能详尽地阐明了系统中部分主要功能的实现方案，深入理解系统的工作原理。最后，介绍了软件测试相关的知识，并以竞赛管理系统的几个功能为例，展示了对系统中各个功能的测试情况。通过测试阶段后，确保系统能够正常工作，切实实现目标功能，保质保量完成目标任务。

6 总结与展望

6.1 总结

通过对论文的研读和对竞赛管理发展现状的研究，设计并完成了基于微服务架构的竞赛管理系统系统。竞赛管理系统的前端主要采用了市面上比较流行的 Vue框架及其相关生态。竞赛管理系统的后端主要采用了 SpringBoot 框架并整合 Spring Cloud 作为项目的微服务框架，在数据存储方面，系统使用 Mysql数据库对系统中的数据进行管理。竞赛管理系统旨在为广大学生及教师用户提供一个便捷、稳定的竞赛管理平台。本文主要做了以下的工作：

首先，对竞赛管理系统的课题背景进行深入研究，查找相关资料，确定研究背景及研究目的。其次，根据研究的结果，从业务以及功能的角度对系统进行了需求分析，大致确定了系统功能模块的设计方向。然后，对系统进行了详细的设计，确定了本系统设有学生用户端、教师用户端和管理员用户端。学生用户可以在系统中进行报名竞赛、上传荣誉信息等操作，教师可以通过竞赛文件上传的功能指导学生参与竞赛，管理员则可以在系统中进行用户管理、发布赛事宣传等操作。最后依次实现设计的功能流程，并对功能进行测试，确保系统实现了覆盖竞赛参与的全过程。

6.2 展望

近年来，由于高等教育的飞速发展以及国家对高等教育的重视，各种学科类竞赛应运而生。竞赛种类以及数量的增多也为当前大学生的提供了许多宝贵的学习、实践的机会。同时，随着科技、互联网的发展以及相关政策的支持，数字化教育平台也得到了飞速的发展。在这种潮流的驱使下，大学生竞赛管理系统也成为了一个新的研究方向。

本文实现了竞赛管理系统的设计与开发。但是碍于时间紧迫，且本人的能力限制，竞赛管理系统还有很多可以完善和改进的地方，例如可以再添加一些系统功能，亦或者针对使用者的特点进行特定方向的优化。此外，虽然现在本系统的设计初衷是帮助大学生参与竞赛更加便捷，但是如果将该系统未来进行界面上的优化、功能扩充、流程的完善，并且与竞赛相关的机构进行对接，该系统将为更多的大学生提供更加便利、高效的竞赛服务。

参考文献

[1] 罗玲玲. 高校学科竞赛信息管理系统的研究与分析[J]. 高师理科学刊, 2018, 38 (05): 40-43.

[2] 彭杨丽, 徐家喜, 陈培培. 基于Java EE学科竞赛管理系统的设计与实现[J]. 电脑知识与技术, 2018, 14(19): 93-95. DOI:10.14004/j.cnki.ckt.2018.2438.

[3] 赵小蕾, 谢庆辉. 大学生科技竞赛管理方式探讨[J]. 计算机时代, 2015, No. 276(06): 76-78. DOI:10.16644/j.cnki.cn33-1094/tp.2015.06.012. 冯子辉, 欧阳勇. 通用学科竞赛管理系统的设计与实现[J]. 信息与电脑(理论版), 2017(19): 74-77.

[4] 简国明, 蔡润哲, 林兴, 刘金梅, 彭建高. 高校学生学科竞赛管理平台的系统设计与实现[J]. 高师理科学刊, 2018, 38 (05): 40-43.

[5] 李婷婷, 张建农. 基于Web技术的大学生创新创业项目管理系统设计与实现[J]. 计算机工程与应用, 2018, 54(16): 156-161.

[6] 赵娜, 李宏伟. 基于云计算平台下大学生创新创业项目管理系统设计[J]. 计算机产品与流通, 2019(10): 132-133.

[7] Chen C H, Lee H M, Chen Y C. NTU online judge: A web-based programming contest management system[J]. IEEE Transactions on Education, 2005, 48(2): 325-333.

[8] Li J, Liu Y, Chen Y, et al. A novel competition management system for enterprise innovation contests[J]. Computers in Industry, 2018, 101: 1-12.

[9] 曾佳. 基于Spring Cloud的微服务架构设计[J]. 电子技术, 2023, 52(01): 54-55.

[10] Alshuqayran N, Ali N, Evans R. A systematic mapping study in microservice architecture[C]//2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA). IEEE, 2016: 44-51.

[11] Cosmina I. Spring microservices with spring cloud[M]//Pivotal Certified Professional Spring Developer Exam. Apress, Berkeley, CA, 2017: 435-459.

[12] 曾佳. 基于Spring Cloud的微服务架构设计[J]. 电子技术, 2023, 52(01): 54-55.

[13] 张培妍. 基于Vue.js的智能匹配系统的设计与实现[D]. 西安电子科技大学, 2019. DOI:10.27389/d.cnki.gxadu.2019.001932.

[14] 李志远, 姚明菊, 魏强. “智愿点津” 高考志愿填报辅助系统设计与实现[J]. 电脑编程技巧与维护, 2023, No. 451(01): 123-126. DOI:10.16184/j.cnki.comprg.2023.01.003.

[15] 王志任. 基于Vue.js的开发平台的设计与实现[D]. 广东工业大学, 2018.

[16] 刘金羽. 前后端分离的在线考试系统设计与实现[J]. 电脑编程技巧与维护, 2020(04): 44-46. DOI:10.16184/j.cnki.comprg.2020.04.017.

[17] 程萍. 基于Vue.js的科研管理系统实现策略研究[J]. 电脑知识与技术, 2023, 19(06): 31-32+50. DOI:10.14004/j.cnki.ckt.2023.0311.

[18] 冯龙. 云存储技术在高校信息化服务中的应用与研究[D]. 天津科技大学, 2013.

附录

英文缩写	英文全称	中文译名
GRASP	General Responsibility Assignment Software Patterns	通用职责分配软件模式
API	Application Programming Interface	应用程序接口

MVC	Model View Controller	模型，视图，控制器
SOA	SOAframework	面向服务框架
DOM	Document Object Model	文档对象模型
CSS	Cascading Style Sheets	层叠样式
HTML	Hyper Text Markup	超文本标记语言

英文缩写英文全称中文译名

GRASP General Responsibility Assignment Software Patterns 通用职责分配软件模式

API Application Programming Interface 应用程序接口

MVC Model View Controller 模型，视图，控制器

SOA SOAframework 面向服务框架

DOM Document Object Model 文档对象模型

CSS Cascading Style Sheets 层叠样式

HTML Hyper Text Markup 超文本标记语言

致谢

此番执笔颇为忐忑竟不知如何言语。

感谢我的导师赵海龙老师，感谢老师从毕设选题到论文的撰写的过程中一直给予我学术上的充分指导，以及精神上的支持，一步步引领我完成了我大学生涯的最后一份答卷。同时，我要感谢在这四年中每一位教导过我、帮助过我的老师们，是您们孜孜不倦的教诲，使我积累了前进的动力，点亮了我一盏盏知识的灯，灯光将为我指引未来前行的方向。

感谢我的父母，给了我一个开放自由的成长环境，无条件支持我做的每一个决定。在我的人生路上他们一直是我最坚强的后盾。

感谢我身边的每一位朋友，青春的路上有你们相伴，何其有幸。

最后，我要感谢我自己，感谢自己的勇敢，感谢自己的从不放弃，感谢自己在过去做出的所有努力。

时光总是在人最不经意时悄然滑过，当我在键盘上敲击着这段文字时，说明我那漫长而又短暂的大学四年已经从我的指缝间偷偷的溜走了。回望来时的路，纵有千般不舍，也只剩下一句“再见！”。

指 标

疑似剽窃文字表述

1. 信息，需要输入必填信息同预期结果
- 2 点击“添加”按钮，输入所有信息，点击“提交”按钮添加框关闭，

说明：1. 总文字复制比：被检测论文总重合字数在总字数中所占的比例

2. 去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

3. 去除本人文献复制比：去除作者本人文献后，计算出来的重合字数在总字数中所占的比例

4. 单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比

5. 复制比：按照“四舍五入”规则，保留1位小数

6. 指标是由系统根据《学术论文不端行为的界定标准》自动生成的

7. **红色文字**表示文字复制部分；**绿色文字**表示引用部分（包括系统自动识别为引用的部分）；**棕灰色文字**表示系统依据作者姓名识别的本人其他文献部分

8. 本报告单仅对您所选择的比对时间范围、资源范围内的检测结果负责



✉ amlc@cnki.net

🌐 <https://check.cnki.net/>