|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 单位代码 | 10642 | 密 级 | 公开 | 学 号 | 201558274039 |

重庆文理学院

**学士学位论文**

**基于微服务架构的在线试题系统设计与实现**

论文作者： 韩金明

指导教师： 罗代忠

专业班级： 2015级计算机科学与技术1班

提交论文日期：2019年4月22日

论文答辩日期：2019年4月27日

学位授予单位：重庆文理学院

中 国 · 重 庆

2019年4月

目录

**中文摘要**................................................................................................................................................................I

**英文摘要**...............................................................................................................................................................II

**[1 绪论](#_Toc29659)** [1](#_Toc29659)

**[1.1研究背景](#_Toc2677)** [1](#_Toc2677)

**[1.2系统开发的意义](#_Toc517)** [1](#_Toc517)

**[1.2.1现状分析](#_Toc16818)** [1](#_Toc16818)

[1.2.2可行性分析 2](#_Toc17985)

[1.2.3功能简介 2](#_Toc2497)

**[2 系统简介](#_Toc14312)** [3](#_Toc14312)

**[2.1系统技术支持](#_Toc16107)** [3](#_Toc16107)

**[2.2需求分析](#_Toc24896)** [4](#_Toc24896)

[2.2.1功能需求 4](#_Toc29588)

[2.2.2性能需求 4](#_Toc8603)

[2.2.3高可用 4](#_Toc1392)

[2.2.4 高可维护性和可扩展性 4](#_Toc31078)

**[3系统设计](#_Toc17296)** [5](#_Toc17296)

**[3.1题库的定义](#_Toc9628)** [5](#_Toc9628)

**[3.2系统功能图](#_Toc26060)** [5](#_Toc26060)

**[3.3数据库设计](#_Toc14959)** [5](#_Toc14959)

**[3.4开发和发布环境](#_Toc25693)** [11](#_Toc25693)

**[4功能实现](#_Toc11290)** [12](#_Toc11290)

**[4.1用户登陆](#_Toc9358)** [12](#_Toc9358)

**[4.2试题添加](#_Toc15835)** [12](#_Toc15835)

**[4.3试题管理](#_Toc30694)** [15](#_Toc30694)

**[4.4 试卷生成](#_Toc20005)** [16](#_Toc20005)

**[4.5 考试答题](#_Toc15285)** [17](#_Toc15285)

**[4.6 成绩管理](#_Toc27083)** [18](#_Toc27083)

**[5系统测试](#_Toc25639)** [19](#_Toc25639)

[5.1用户测试 19](#_Toc7718)

[5.2试题测试 19](#_Toc22562)

[5.3数据测试 19](#_Toc28775)

[5.4试卷生成测试 20](#_Toc30999)

[5.5考试功能测试....................................................................................................................................20](#_Toc30999)

[5.6抗压测试............................................................................................................................................ 20](#_Toc30999)

**[6总结与展望](#_Toc15869)** [21](#_Toc15869)

**[致 谢](#_Toc6171)** [22](#_Toc6171)

**[参 考 文 献](#_Toc32323)** [23](#_Toc32323)

**基于微服务架构的在线试题系统设计与实现**

计算机科学与技术 韩金明

指导教师 罗代忠

摘要：

随着互联网的急速发展，考试的方式和介质也发生了很大的改变，传统的考试形式正面迎着时代的冲击。教育行业也应该和网联网相结合，考试的形式从传统的纸质考试阅卷到计算机辅助阅卷，再到实现了完全的计算机考试，实现了时代化的改变，在线考试系统具有巨大的优势，可以减少纸张的使用，组卷者的工作量和阅卷的时间，考试的时间和空间限制也不再是难以协调的问题。

本系统基于SpringBoot微服务的应用系统，利用当前相对成熟的技术进行系统的架构和数据的处理，系统分为管理员和用户，而用户同时拥有leader(领导者)和flower(跟随者)两种角色，leader可以出题生成试卷，并制定考试安排给他的团队，flower则可以出题或参加其所加入团队的考试。用户自己添加的试题和生成的试卷可以选择账户公开、账户私有或者完全公开共所有人访问，进而分享自己资源。

本系统使用分布式高可用的部署方式，可以有效的减少系统出现由硬件事故导师的数据丢失，还可以提高系统的响应速度，再加上系统丰富的功能，能极大方便学习者是教学者。

关键词：试题库; 在线考试系统； 在线试题生成系统

**Online Test System Based On Micro Service**

Major: Computer Science and Technology

Class: One

Author: Han Jinming Supervisor: Luo Daizhong

**Abstract：**With the rapid development of the Internet, the methods and media of the examination have also undergone great changes. The traditional form of examination is facing the impact of the times. The education industry should also be combined with the network of the Internet. The form of the examination is from the traditional paper examination to the computer-aided examination, and then to the complete computer examination, realizing the changes of the times. The online examination system has great advantages. Reducing the use of paper, the workload of the tester and the time of scoring, the time and space constraints of the exam are no longer difficult to coordinate.

The system is based on the application system of SpringBoot microservices. It uses the current relatively mature technology to process the architecture and data of the system. The system is divided into administrators and users, and the user has both leader (leader) and flower (follower). Roles, leaders can generate questions and develop test schedules for his team, and flowers can take questions or take exams for the teams they join. The questions and generated test papers added by the user can be shared by the account, the account is private, or the account is completely public, so that they can share their resources.

The system uses distributed high-availability deployment mode, which can effectively reduce the data loss caused by hardware accident instructors in the system, and can also improve the response speed of the system. Together with the rich functions of the system, it can greatly facilitate the learner to be a teacher.

**Keywords**：Test question bank; online test system; online test question generation system.

1绪论

1.1研究背景

据某调查小组统计，国内大部分的学校的命题还是任课老师出题制卷的方式，为保证考试的公平性，防止漏题和其他机构押题，往往需要同时出多套难度和题量相同的不同试卷，并且考试前试卷的保密工作，考试中的公正参考，考试后的公正阅卷都是一个比较难解决的问题。

1.2系统开发的意义

随着互联网的发展和办公自动化需求的提高，数据化，共享和智能已经成为行业追求的标准。而学校作为孩子学习的地方，跟上时代的步伐是刻不容缓的。而拥有一个智能，科学化的教学系统不仅可以大幅度提高教学的效率，还能体现出教学成果。相对于传统的考试形式，采用纸、笔的考试形式，用到大量的纸张，对自然资源造成大量的浪费，再加上后期试卷批改的工作量，不仅成本较高，而且效率也不是很高。而在线考试系统不仅拥有大量的题库，能生产质量较高的试卷，还避免了对纸张的浪费，保护了大自然，更是拥有自动审题系统，较少了教师的工作量。

而建立题库，使用算法可快速生成符合要求的不同试卷，试卷创建者可以快速的通知参考人员在对应的时间范围内参加考试，利用学校的网络硬件资源，还能保证考试的公平，再加上系统的自动阅卷功能，更能大幅度减少老师的工作量。当题库使用者达到一定数量后，试题分享更能使得命题更加灵活和多变。

1.2.1现状分析

传统考试方试是要求考试组织者提前进行试卷命题，然后寻找一个共同时间，让考试参与者到考场地点进行统一测试。

这种模式存在一些难以忽略的问题：

命题：由于考试需要同时进行，试卷就得提前命题，命题过程中的保证命题的难易程度、是否超出考试范围，命题后试卷的如何保证其保密性也是一个较大的难题，毕竟试题一旦泄露，考试的意义页不复存在。

试卷印刷：传统测验的试卷是纸质的，碰到大规模考试时，需要印刷大量的试卷，其耗时，成本都是不容忽视的。当遇到异地命题时，还得考虑传输中的试卷丢失，泄密和损坏的问题。

阅卷：当试卷数量过多时，阅卷者阅卷的准确度难以得到保证，工作量也很大。

1.2.2可行性分析

经济可行性：开发系统需要一定的人力成本，再加上系统运行环境和后期维护成，但与之带来的便利性相比，就显得不那么重要了。

技术可行性：如今web技术的高速发展和硬件不断的更新，可靠性越来越高，其上千万级的并行访问都能完美驾驭，还能保证其用户数据的安全，简单的一个系统更是绰绰有余。

1.2.3功能简介

其功能为：

（1）用户模块，提供用户注册登陆和加入组织的功能；

（2）试题模块，提供试题管理和分享功能；

（3）试卷模块，提供试卷生成功能。

（4）考试模块，提供参加考试功能。

（4）论坛模块，提供参考者交流评论功能。

2 系统简介

随着大数据、微服务、分布式等技术的出现和成熟,性能和效率也变得越来越重要,本系统使用nginx+spring zuul作反向代理、负载均衡和高可用配置模式, 使用前后段分离提高开发效率和减少维护成本,以mysql作为数据库减少开发成本, redis作为缓存服务器提高性能,尽可能提高系统的性能。

2.1系统技术支持

2.1.1 JAVA

Java作为1995年推出存活至今的编程语言[1]，也是现如今最流行的编程语言之一，其安全性也经得住时间的考验。Java语言不仅简单易学，有很高的安全性，支持跨平台，多线程和网络编程；而且社区非常活跃，遇到问题也相对容易解决；更重要的是它拥有很广泛的工具包，能够较快速的达到预期的效果。

2.1.2 SpringBoot

Spring自2004年3月24日发行以来，深受java开发者的青睐，这是一个力图突破JavaWeb传统开发的困境，从实际的需求出发，注重于灵巧、轻便、易于开发、测试和部署的轻量级框架，拥有丰富的子项目，比如Spring mvc, spring cloud, spring data, spring aop等

而SpringBoot则是spring团队推出的全新框架[2]，其目的是进一步简化项目的搭建和开发过程，并且可以做到模块化和微服务化，进一步保障系统的健壮性。

2.1.3 Vue

vue是一套构建用户界面的渐进式、数据驱动的前端框架[3]。它能简化和模块化前段页面，使得开发和维护更加容易

2.1.4 Restful

Resultfu是一组架构约束条件和准则，让系统分成化[4]。而本系统采用该架构模式，使前后端分离，提高系统的开发速度和减少后期的维护成本

2.1.5 MySql

Mysql作为最流行的关系型数据库之一[5]，具有体积小，稳定性高、成本低和速度快的优势，是中小型企业和个人的最佳选择。

2.1.6 Intellij IDEA

Intellij IDEA作为java界最好用的编辑器[6]，选择它，意味着选择了简单，高效。

2.2需求分析

2.2.1功能需求

1.试题管理

允许用户对试题进行编辑和管理，比如添加，修改和删除试题，对试题的可访问权限进行灵活控制，并且增加分享机制，加大试题库的规模。

2.试卷管理

允许用户生成和管理试卷，或者参加被人的试卷测试。

3.团队管理

允许用户创建团队并进行管理，或者参加别人的团队。

2.2.2性能需求

尽可能的提升用户访问的流畅度，优化用户的操作友好度，支持较大数量并行访问。

2.2.3高可用

高可用的服务器部署环境。在以前，部署的web服务使用时间一长，就很容易访问速度延迟，或者由于其他意外原因，导致web服务环境断电重启，使服务中断，这是非常不友好的，甚至还会造成数据丢失。

本系统使用两套隔离的系统环境部署两个web服务，采用zookeeper+kafka消息同步机制同步系统间的数据，用nginx进行负载均衡和代理转发，redis作为缓存数据共享，尽可能的使得系统更加稳定，即使出现什么不可预知的问题，也能保障系统的正常运行。

2.2.4 高可维护性和可扩展性

一个系统在发布之后，优化、消缺和改善是免不了的。开发系统时能够考虑到后期维护的话，利用微服务将功能模块合理分化，每个模块都建立可熔断转发机制，避免后期更新时影响项目运行。

在增加新功能时，能够快速稳定的在系统运行期间发布新功能。数据库也需要做好良好的可维护性，可回退性，用户删除的数据进行延缓删除，修改做好记录，并定期清除历史遗留的过期信息，保证数据库的响应速度。

3系统设计

3.1题库的定义

题库作为整个系统的核心，是最为主要的，业务逻辑也是最为繁杂的。建立题库不仅要考虑试题的多样性，科学性和有效性，还要对试题的使用频率，正确率做出统计。这就需要建立一个严谨的数据模型和算法。

3.2系统功能图

一个系统是怎样的就要看它的数据流图，数据流图能够较清晰的向用户介绍系统的功能，有利用用户的使用和开发者的维护和扩展，如图3-1。

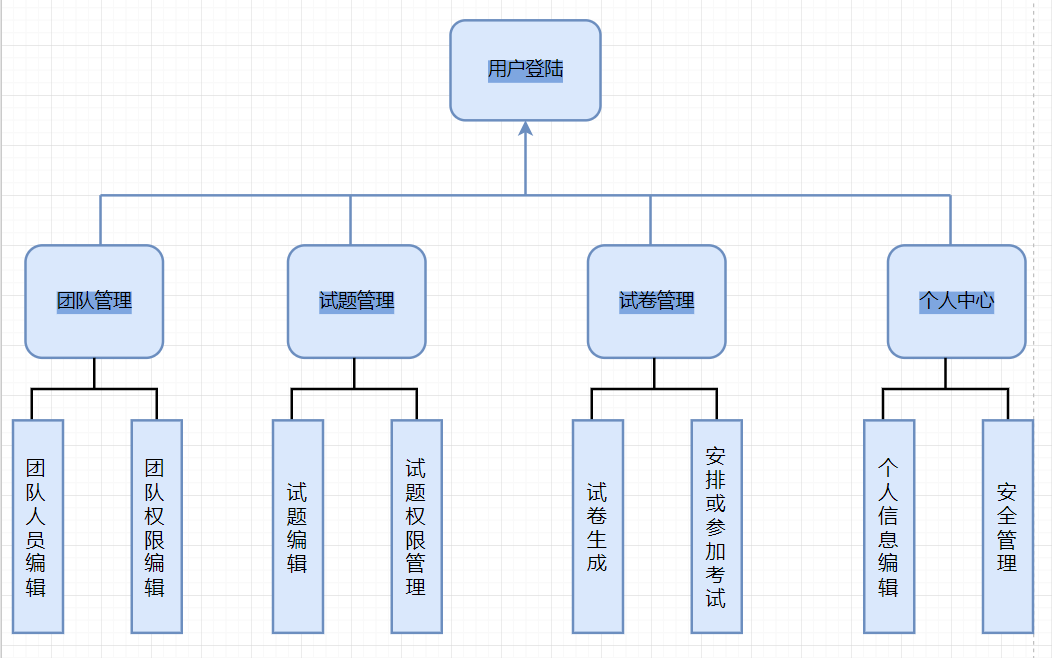


图3-1 系统功能图

3.3数据库设计

本系统使用的是开源数据库mysql,数据库中的表数量较多，其中有：

用户作为交互的载体，其表的设计就显得极为重要，用户保存者用户的id，用户名，昵称等信息。具体设计如表3-1所示：

表3-1 user表

|  |  |  |  |
| --- | --- | --- | --- |
| Id | int | 表逻辑ID | 自增 |
| username | varchar(32) | 用户名 | 无 |
| password | varchar(32) | 密码 | 无 |
| nickname | varchar(1) | 昵称 | 无 |
| birthday | date | 出生年月 | 无 |
| avatar | varchar(256) | 头像 | 无 |
| tel | varchar(16) | 电话号码 | 无 |
| email | varchar(64) | 邮箱 | 无 |
| create\_time | datatime | 创建时间 | 当前时间 |
| term | varchar(200) | 团队信息（json） | 无 |
| Id | int | 表逻辑ID | 自增 |
| roles | varchar(200) | 用户所拥有的角色 | 无 |
| region | varchar(32) | 区域 | 无 |
| autograph | varchar(64) | 个性签名 | 无 |
| synopsis | varchar(512) | 个人介绍 | 无 |

续表3-1

登录记录表，用于保存用户的登录信息，其作用为检查用户的账号安全，和系统的使用频率。具体设计如表3-2所示：

表3-2 loginRecord表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| userid | int(32) | 用户id | 无 |
| username | varchar(32) | 用户名 | 无 |
| login\_time | datetime | 登陆时间 | 当前时间 |
| login\_region | varchar(32) | 登陆地点 | 无 |
| login\_ip | varchar(128) | 登陆ip | 无 |
| logout\_time | datetime | 退出登陆时间 | 无 |
| logout\_way | varchar(20) | 退出登陆方式 | 无 |

团队信息表, 用于保存用户所加入的班级和组织,团队的管理者可以管理其他成员的权限和发布团队任务。具体设计如表3-3所示：

表3-3 term表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| avatar | varchar(512) | 团队头像 | 默认头像 |
| level | int | 团队级别 | 1 |
| memberRank | varchar(256) | 成员等级划分信息 | 默认划分 |
| description | varchar(512) | 团队简介 | 无 |
| createDate | datetime | 创建时间 | 当前时间 |
| createUserId | int(11) | 团队创建人Id | 无 |
| createUsername | varchar(32) | 团队创建人 | 无 |
| lastModifyTime | datetime | 最后编辑时间 | 保存时间 |
| lastActiveTime | datatime | 最后活动时间 | 无 |
| leader | varchar(32) | 团队领导者 | 无 |
| leaderId | int(11) | 团队领导者Id | 无 |
| memberNumber | int | 团队成员数 | 1 |
| dueTime | int | 解散时间 | 无 |

菜单表, 用于保存系统所定义的基本菜单和用户自定义的菜单,给予用户灵活的配置空间。具体设计如表3-4所示：

表3-4 menu表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| name | varchar(32) | 菜单名称 | 无 |
| parentId | int(11) | 父级菜单id | 0 |
| url | varchar(64) | 菜单链接地址 | 无 |
| icon | varchar(64) | 菜单图标 | 默认图标 |
| createUserId | int(11) | 创建用户id | 无 |
| createUser | varchar(32) | 创建用户 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

科目表, 是保存试题类型的载体，用户可以自定义自己的科目，分享科目，并未自己的科目添加试题和生成试卷。具体设计如表3-5所示：

表3-5 subject表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| name | varchar(128) | 科目名称 | 无 |
| description | varchar(256) | 科目介绍 | 无 |
| classifyId | varchar(11) | 所属科目id | 0 |
| classify | varchar(128) | 所属科目名称 | 无 |
| createUser | varchar(32) | 创建人 | 无 |
| createUserId | int(11) | 创建人id | 无 |
| createDate | datetime | 创建时间 | 当前时间 |
| permission | varchar(200) | 科目权限 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

选择题作为最常见的题，它与其他题比较较为精简，是最受欢迎的题型。选择题分为多选和单选，而答案是相对主观唯一的。具体设计如表3-6所示：

表3-6 question\_choose表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| remarks | varchar(512) | 备注 | 无 |
| topic | varchar(512) | 题目 | 无 |
| choices | varchar(512) | 选项信息 | 无 |
| answer | varchar(20) | 答案 | 无 |
| type | tinyint | 类型：｛0单选，1多选｝ | 无 |
| subjectId | int(11) | 所属科目id | 无 |
| subject | varchar(128) | 所属科目名称 | 无 |
| createUser | varchar(32) | 创建人 | 无 |
| 列名 | 类型 | 注释 | 默认值 |
| createUserId | int(11) | 创建人id | 无 |
| permission | varchar(200) | 访问权限 | 无 |
| createDate | datetime | 创建时间 | 无 |
| keyword | varchar(48) | 关键字 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

续表3-6

简答题是答案很灵活的表，其检阅也是相对困难的，自动检阅一直是计算机未完善的技术难点之一。具体设计如表3-7所示：

表3-7 question\_jquiz表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| topic | varchar(512) | 题目 | 无 |
| answer | varchar(512) | 答案 | 无 |
| remarks | varchar(512) | 备注 | 无 |
| subjectId | int(11) | 所属科目id | 无 |
| subject | varchar(128) | 所属科目名称 | 无 |
| createUser | varchar(32) | 创建人 | 无 |
| createUserId | int(11) | 创建人id | 无 |
| createDate | datetime | 创建时间 | 无 |
| permission | varchar(200) | 访问权限 | 无 |
| keyword | varchar(48) | 关键字 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

填空题作为每一门学科的试卷上必有的考题，分为定量填空和定性填空，答案可唯一可多变，其特点为题小、覆盖面广、形式灵活、跨度大。具体设计如表3-8所示：

表3-8 question题

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| topic | varchar(512) | 题目 | 无 |
| answer | varchar(256) | 答案 | 无 |
| remarks | varchar(512) | 备注 | 无 |
| subjectId | int(11) | 所属科目id | 无 |
| subject | varchar(128) | 所属科目名称 | 无 |
| createUser | varchar(32) | 创建人 | 无 |
| createUserId | int(11) | 创建人id | 无 |
| createDate | datetime | 创建时间 | 无 |
| permission | varchar(200) | 访问权限 | 无 |
| keyword | varchar(48) | 关键字 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

判断题是一种以对或错来选择的答案，其数据结构是最为简单的。具体设计如表3-9所示：

表3-9 判断题

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| topic | varchar(512) | 题目 | 无 |
| answer | tinyint | 答案 | 无 |
| remarks | varchar(512) | 备注 | 无 |
| subjectId | int(11) | 所属科目id | 无 |
| subject | varchar(128) | 所属科目名称 | 无 |
| createUser | varchar(32) | 创建人 | 无 |
| createUserId | int(11) | 创建人id | 无 |
| createDate | datetime | 创建时间 | 无 |
| permission | varchar(200) | 访问权限 | 无 |
| keyword | varchar(48) | 关键字 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

试卷表作为系统的核心之一，其生成算法和检阅算法是最为重要的。本系统的试卷可以机阅（客观题支持较好，主观题建议人工阅卷）和人工阅卷，具有较高的灵活性。具体设计如表3-10所示：

表3-10 testPaper表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| name | varchar(128) | 试卷标题 | 无 |
| subjectId | integer(11) | 科目id | 无 |
| subject | varchar(128) | 科目名字 | 无 |
| createUser | varchar(32) | 创建人 | 无 |
| createUserId | int(11) | 创建人id | 无 |
| createDate | datetime | 创建时间 | 无 |
| totalScore | varchar(500) | 总分 | 无 |
| totalTime | varchar(3) | 总时长 | 无 |
| topics | varchar(500) | 题目id集 | 无 |
| description | varchar(200) | 介绍 | 无 |
| permission | varchar(200) | 访问权限 | 无 |
| keyword | varchar(48) | 关键字 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

考试安排表是对生成试卷后功能的一种扩展，用户生成试卷后可以邀请他人一起做试卷或者分享试卷。具体设计如表3-11所示：

表3-11 testPlan表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| name | varchar(128) | 名称 | 无 |
| testId | integer(11) | 试卷id | 无 |
| testName | varchar(128) | 试卷名称 | 无 |
| remarks | varchar(200) | 备注 | 无 |
| subjectId | integer(11) | 科目id | 无 |
| subject | varchar(128) | 科目名字 | 无 |
| createUser | varchar(32) | 创建人 | 无 |
| createUserId | int(11) | 创建人id | 无 |
| createDate | datetime | 创建时间 | 无 |
| startTime | datetime | 开始时间 | 无 |
| endTime | datetime | 结束时间 | 无 |
| terms | varchar(500) | 面向人员 | 无 |
| joinNumber | int | 参加人数 | 无 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

考试成绩表是对用户参加考试安排的效果的一种衡量，用户可模拟教师生成试卷给团队的其他成员(学生)测试，并批阅试卷，从而获取学生水平。具体设计如表3-12所示：

表3-12 testScore表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| testPlanId | int(11) | 考试安排id | 无 |
| testPlanName | varchar(128) | 考试安排信息 | 无 |
| participantId | int(11) | 参与人id（user表userid） | 无 |
| participantName | varchar(32) | 参与人姓名 | 无 |
| createDateTime | datetime | 创建时间 | 无 |
| score | int | 分数 | 无 |

评论表是用于存储用户对试题、试卷和分享的评论，用于加强用户之间的交互性和系统社区的灵活性。具体设计如表3-13所示：

表3-13 review表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 注释 | 默认值 |
| id | int(11) | 表逻辑ID | 自增 |
| context | varchar(500) | 评论内容 | 无 |
| reviewUserId | int(11) | 评论人Id | 无 |
| reviewUser | varchar(32) | 评论人名字 | 无 |
| reviewUserAvatar | varchar(32) | 评论人头像 | 无 |
| reviewTime | datetime | 评论时间 | 当前时间 |
| patternId | varchar(32) | 评论所在区域Id | 无 |
| 列名 | 类型 | 注释 | 默认值 |
| parentReviewId | int(11) | 回复评论id | 无 |
| good | int | 点赞人数 | 0 |
| bad | int | 鄙视人数 | 0 |
| dataStatus | varchar(32) | 数据状态 | active （使用中) |

**3.4开发和发布环境**

3.4.1开发环境

为了开发速度，本系统的开发环境为：

（1）Windows 10系统 --操作系统，最新的windows操作系统

（2）Intellij IDEA 2018 --开发平台，最好用的java开发工具

（3）jdk 1.8 --开发核心环境，当前最稳定的java版本

（4）gradle --代码包管理工具

（5）git --代码管理平台

（6）mysql --数据存储中心，免费的，节约成本的不二之选

（7）redis --缓存中心，用于记录临时信息

（8）apache tomcat --web部署工具

（9）springboot --开发框架

（10）nginx --静态资源服务器，用于代理转发

3.4.2发布环境

性能优先，一切为性能服务：

（1）CentOS 7 系统 --操作系统

（2）jdk1.8 --核心环境

（3）mysql --数据存储中心

（4）redis --缓存中心

（5）apache tomcat --web部署工具

（6）nginx --静态资源服务器

（7）kafka --消息同步工具，用于同步数据

（8）zookeeper -- 数据同步工具

4功能实现

4.1用户登陆

该系统的大部分功能灯饰需要登录的，所以登陆功能的实现是十分重要的。实现思路：前台获取用户的输入，对密码进行加密后用ajax传输给后台[7]，后台判断登录成功，如果成功登陆的话，系统会生成一个token,并存入redis缓存数据库[8],再将token和登录信息放会给前端存储起来。

实现页面如图4.1所示：



图4.1登录界面效果图

4.2试题添加

系统的关键功能，试题添加的方式有多种：逐条添加、批量添加、和excel导入，试题的类型也有5种：单选题、多选题、判断题、填空题和简答题，支持用户自定义内容，还可录入答案，进行自动批阅。

页面效果如图4.2所示:

逐条添加－单选题（其有题目，选项，答案，和关键字信息）

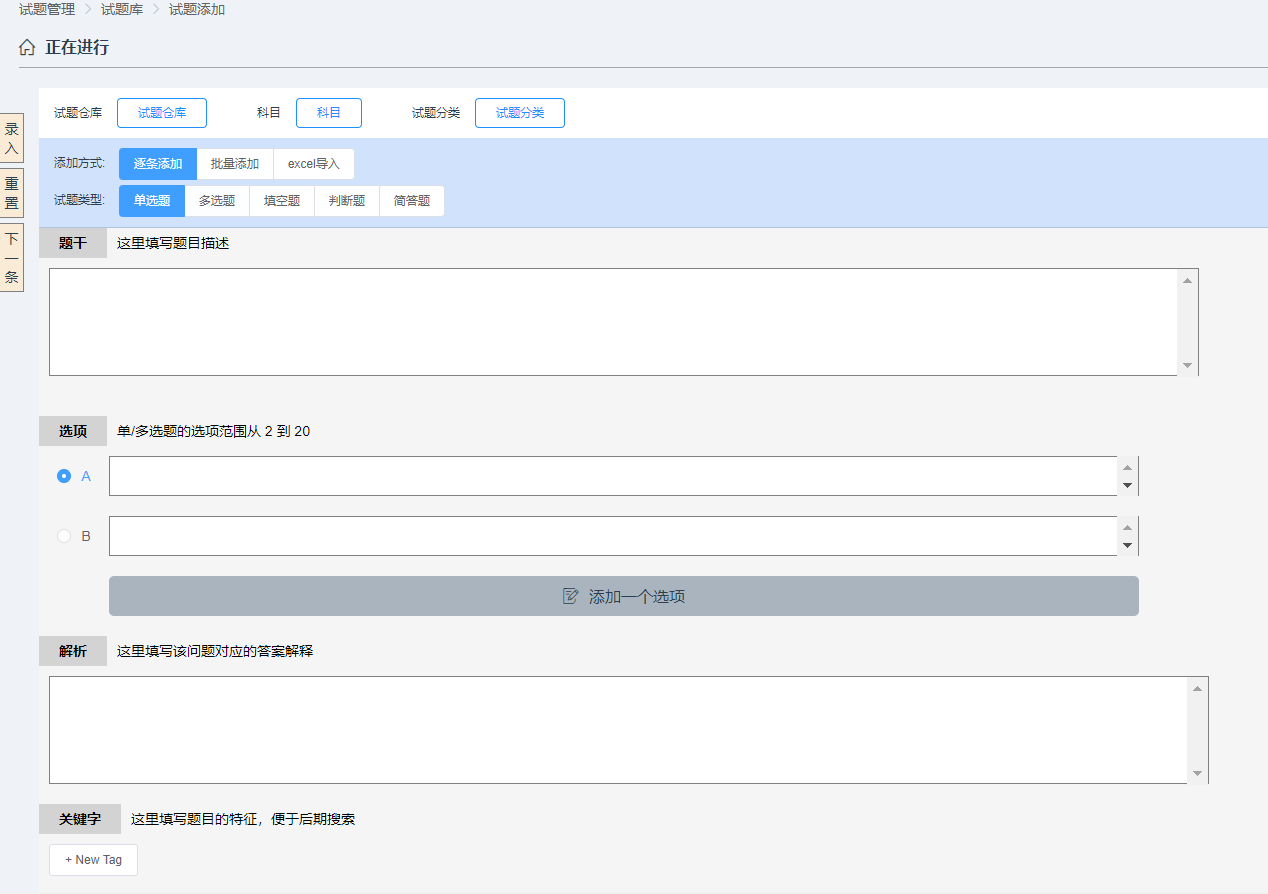


图4.2单选题添加

逐条添加－多选题（与选择题一致）如图4.3所示

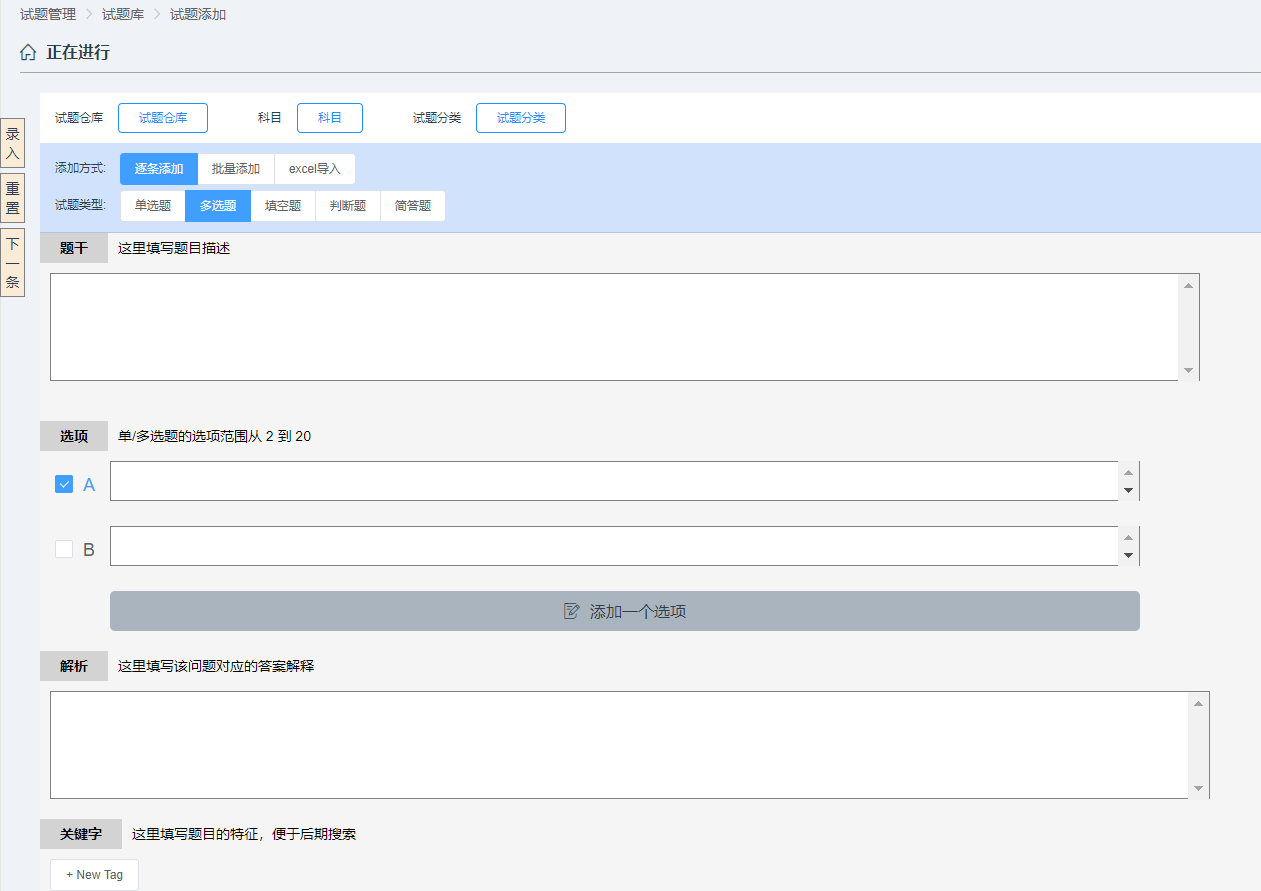


图4.3多选题添加

逐条添加－填空题（拥有题目、答案、解析和关键字信息)如图4.4所示

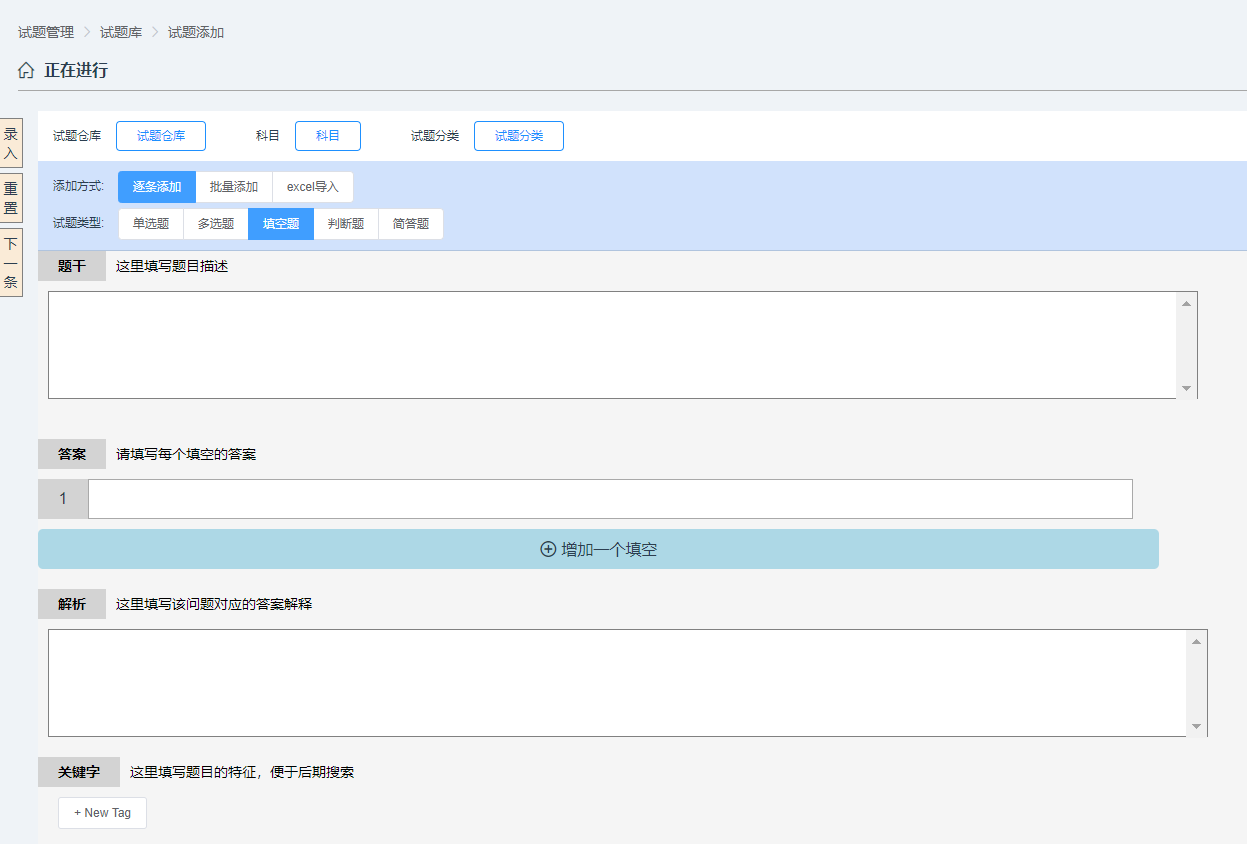


图4.4填空题添加

逐条添加－判断题如图4.5所示

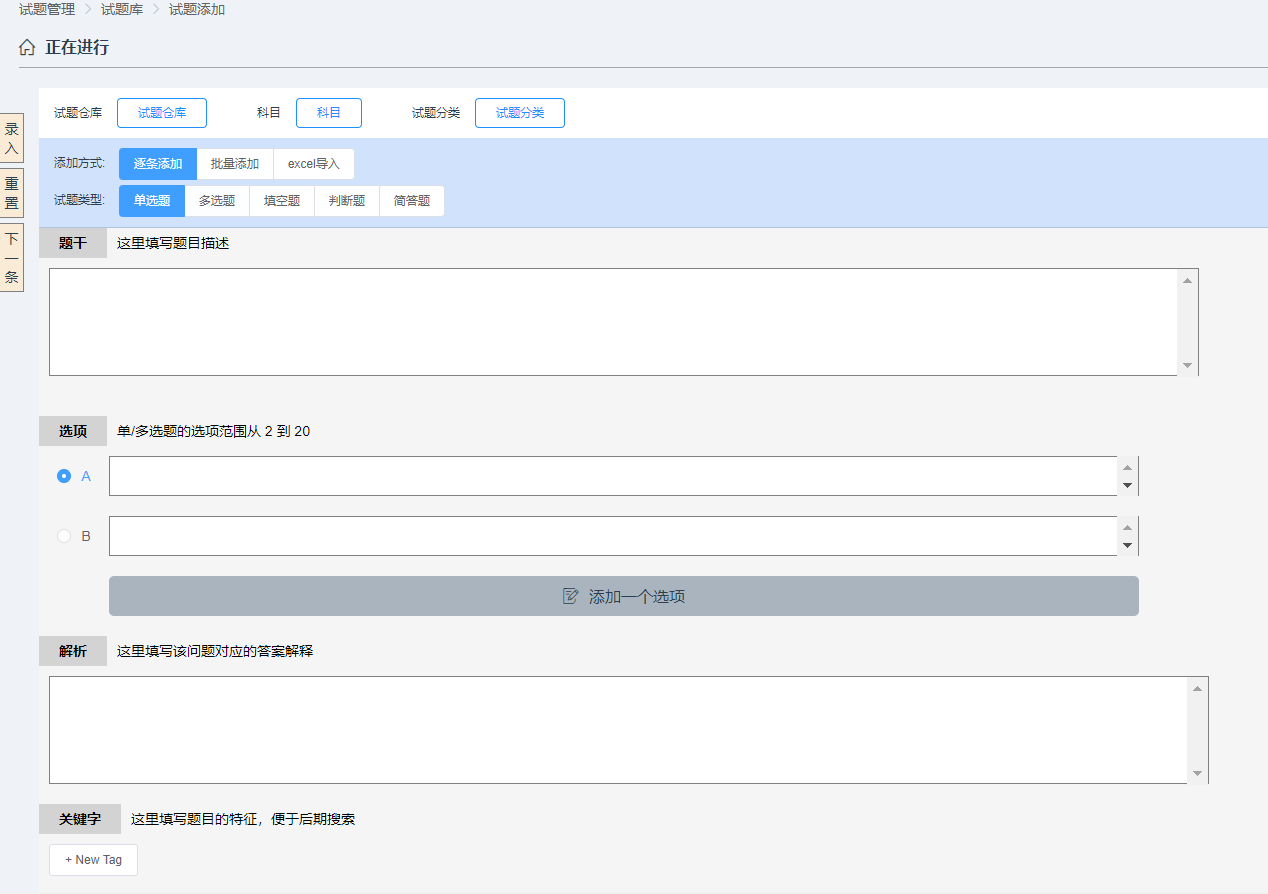


图4.5判断题添加

逐条添加－简答题如图4.6所示

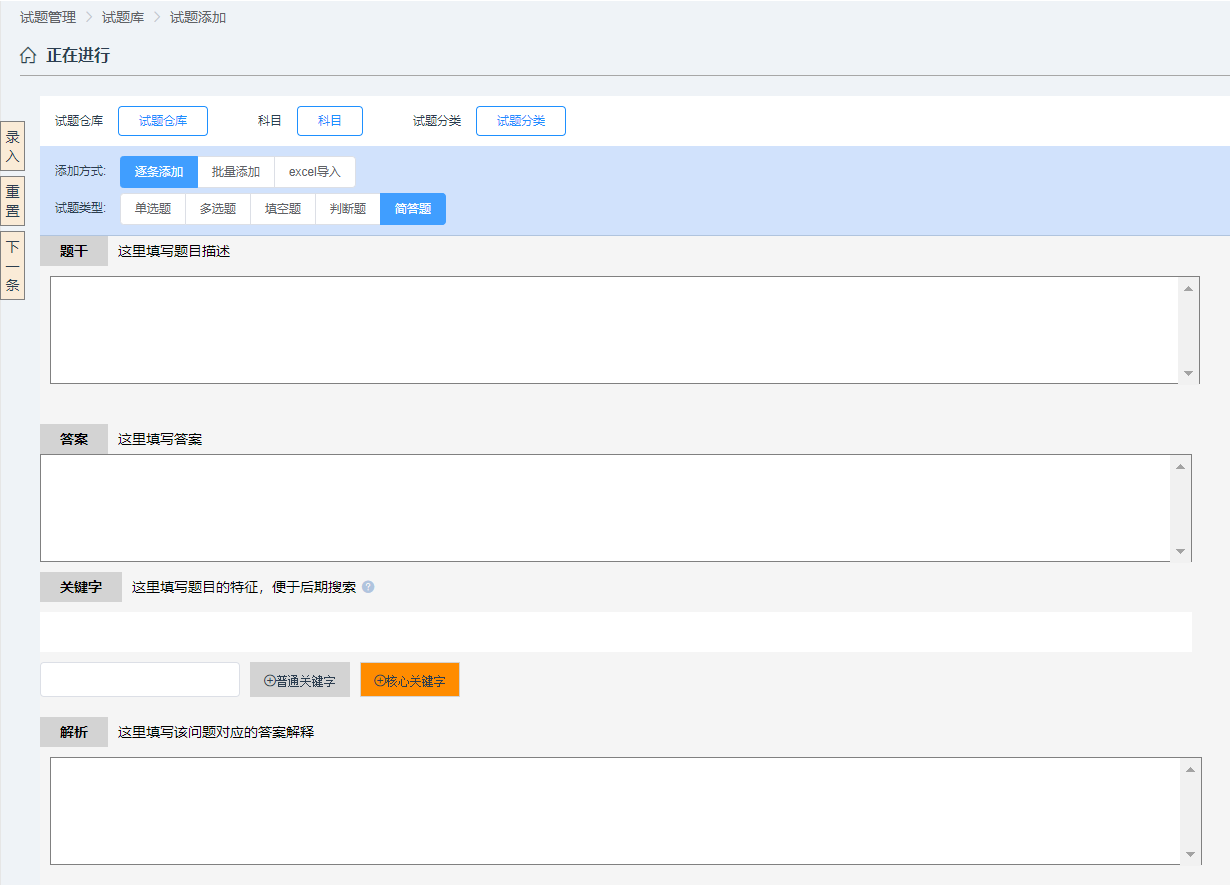


图4.6简答题添加

4.3试题管理

系统的核心功能，用于展示试题添加信息，用户可对私有试题进行管理，还可从公共试题补充试题到私有库如图4.7所示。



图4.7试题列表

4.4 试卷生成

系统的核心功能之一，用户可以根据试题库的试题，灵活的规定每道题的类型、分数，试卷的总分数，生成一张符合要求的试卷。

生成方法：

(1)手动选择试题法

用户可手动调整试题的试题类型，选择不同的到试卷，生成自定义的试卷

(2)根据关键字及试题模版生成试题

系统和根据用户提供的关键字，比如输入六年级、语文、扩展训练等关键字，系统会根据用户题库里的试题生成一张六年级难度偏高的语文试卷，并且可以灵活设置每道题的分值。

生成试卷步骤如图4.8所示：



图4.8试卷生成编辑

生成试卷浏览如图4.9所示



图4.9

4.5 考试答题

考试答题是作为组织者用户的一种权限，组织者可以邀请其他用户进行答题，并获取他人的测试效果如图4.10所示



图4.10试卷浏览

4.6 成绩管理

成绩管理是对组织者发布考试的参考人员测验成果的管理模块， 用户可以在该模块查看和批改测验人员的考试成绩，以便了解参考人员的知识水平。

参考学生信息列表查看如图4.11所示:



图4.11考试信息浏览

批阅学生考试信息如图4.12所示:



图4.12S批阅试卷效果图

# 5系统测试

5.1用户测试

测试目的: 验证用户功能的完整性与用户信息的安全新问题。

测试数据：用户名hjm0928s,密码123456

测试过程：创建用户，模拟用户登录系统。对用户进行修改密码，信息和团队管理等操作，模拟第三方尝试获取用户个人信息。

预期结果： 创建成功

测试结果： 创建成功

5.2试题测试

测试目的: 验证试题添加功能，测试是否能够成功添加试题。

测试数据：用户模拟数据添加

测试过程: 模拟用户添加不同试题，设置不同的关键字，查看试题是否添加成功。

预期结果：成功添加试题数据

测试结果：成功添加试题数据

5.3数据测试

测试目的： 验证用户操作数据和流程进行是否正常

测试数据： 模拟数据

测试过程：模拟用户对各个功能进行测试，尝试以未登录状态获取其他用户隐私信息，人为破环请求结果，测试系统的安全性，模拟系统所在主机异常宕机，测试系统是否具有高可用特性，能否在一套系统宕机之后，正常访问系统和当前正在访问系统的用户是否造成数据丢失。

预期结果：功能正常，数据流程一切正常

测试结果：功能数据正常，隐私数据加密性不够强，一台主机宕机后，用户任可以正常访问，不过某些用户的数据会有一些出入。

5.4 试卷生成测试

测试目的： 验证试卷生成功能

测试数据：试题库数据

测试过程：模拟用户，根据用户的试题库，设计试卷生成规则，题型的类型数量和分值，并生成试卷。

预期结果：试卷生成成功，并且符合要求

测试结果：试卷生成成功，并且符合要求

5.5 考试功能测试

测试目的：测试并验证考试功能，能否供用户进行考试操作。

测试数据：用户生成的试卷

测试过程：打开用户发布的试卷进行模拟答题并提交答题结果

预期结果：试卷发布者接受到答题者的答题数据，并且能进行批阅

测试结果：能够成功接受到答题者的答题数据，进行批阅，给予成绩

5.6抗压测试

测试目的：了解系统的对多用户操作是的抗压能力

测试数据：无

测试过程：利用python爬虫，模拟500个用户对系统进行压力测试，测试系统的在多用户下数据的一致性问题和访问延迟性处理性能。

预期结果: 能够支持超过500个用户的访问。

测试结果：部署环境配置比较低，500个用户访问系统响应延迟较高，系统数据个别出现不一致问题，还需后期维护，不过nginx负载均衡测试正常，kafka日志同步一致。

# 6总结与展望

本系统使用jdk8+springboot+jpa+zuul作为后台开发，mysql作为数据中心，vue作前端显示，经过几个月的奋斗，成功的完成了该系统。该系统预先设计为试题库，用户中心，考试管理，论坛管理等多个子系统，用户之间能够灵活的拥有分享和管理自己试题与试卷的权限，能极大的加大和活跃用户之间的交流，能够使得考试的组织者从大量选题命题制卷的工作中解放出来，从个人和小组织的组织方式变为众多参与者的集体方式，再加上论坛，试卷分享的功能，能够极大的方便众多用户。

由于时间有限，系统的很多功能并没有开发出来，论坛系统和考试管理还需要地段时间来完善，用户权限功能也没有做到很严谨，用户界面的友好型还不够高，还有很大的提升空间，系统数据表的设计也没有做到规范，再加上系统发布的时间不长，运维方便还没怎么进行，还需要很长时间去完成。

通过编写这次系统，我也收获了很多，已经灵活的将以前学的不成系统的知识成功的连接到了一起，加上在编写工程中学到的一些新的知识，极大的丰富了我的知识储备，再加上系统编写过程中的心理变化由开始时的不知所措到项目完成时的有紧有条，我感觉到我自己的变化很大，成熟了很多。

# 致 谢

时间总是转瞬即逝，毕业论文的完成，象征者大学生活的结束。这次毕业设计的成功完成，少不了指导老师的帮助。在选题时，感谢指导老师给予宝贵的参考意见，开题报告的完成也是在导师的指导下完成的。系统设计中，他给我了很多想法；论文编写时，在他的帮助和指导下，了解了论文的编写规范和格式，以及应该着重注意的步骤。

在论文的撰写过程中，我学会了冷静和从容，知道了做事要一步一个脚印，遇到棘手的问题，冷静思考，从多方便寻找解决问题的途径，从容的面对各种可能会出现的难题，并且还要拥有一定的耐心，坚持自己的选择，自信也是很重要的，毕竟连自己都不相信自己，还指望别人相信你？

在大学生活即将结束之际，我想借此机会对出现在我大学生活里的每一个人，感谢你们为我的大学生活描绘了不同的色彩，积累了很多的社会经验。更要感谢同学们的陪伴，让大学生活拥有了更多的精彩，还有老师们的栽培，是你们告诉了我，如何成为一个有价值的人，面对困难时如何选择和面对， 是你们让我拥有了更宽阔的视野，能够从更多角度看待每一个问题，从而做出一个正确的决定。

在此，我要向所有曾帮助和关心过我的所有人送出最真诚的谢意！

# 参 考 文 献

[1] 明日科技.Java从入门到精通.第4版.北京:清华大学出版社.2016

[2]汪云飞. JavaEE开发的颠覆者：Spring Boot实战[M].北京：电子工业出版社.2016

[3]梁灏.Vue.js实战[M].北京:清华大学出版社.2017.10

[4]Leonard Richardson/Mike Amundsen(编写)，赵震/李哲(译).RESTful Web APIs[M].北京:电子工业出版社.2014

[5]Ben Forta(英),刘晓霞/钟鸣(译).MySQL Crash Course[M].北京:人民邮电出版社.2009

[6]LucindaDykes(美),ChrisUllman(美).Ajax入门经典[M].第二版.北京:清华大学出版社.2012

[7]Josiah L. Carlson(美)，黄健宏(译).Redis实战.北京:人民邮电出版社.2015.10

[8]陈雄华，林开雄,文建国.精通Spring4.x 企业应用开发实战[M]. 北京：电子工业出版社.2016

[9]汪云飞. JavaEE开发的颠覆者：Spring Boot实战[M].北京：电子工业出版社.2016

[10]姜承尧. MySQL技术内幕:InnoDB存储引擎[M].第三版.美 2016

[11]李宏誓,杨莫(译). MySQL 排错指南[M].第三版.北京: 人民邮电出版社.2015

[12]约书亚·布洛赫. 有效Java[M].第二版. sun公司. 2015

[13]周志明.深入理解Java虚拟机[M].第三版.北京：机械工业出版社. 2016

[14]Martin Fowler(美) 熊节(译). 重构改善既有代码的设计[M].第五版.北京：中国电力出版社.2015

[15]阳小兰, 罗明. 基于Spring+SpringMVC+MyBatis网上论坛的设计与实现[J]. 黑龙江科技信息, 2016(36):279-280.

**附1： 论文独创性声明及论文使用授权声明**

**论文独创性声明**

我的论文《\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_》是我个人在导师指导下进行的研究工作及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其他机构已经发表或撰写过的研究成果。其他同志对本研究的启发和所做的贡献均已在论文中作了明确的声明并表示了谢意。

作者签名：\_\_\_\_\_\_\_\_\_\_ 日期：\_\_\_\_\_\_\_\_\_\_\_

**论文使用授权声明**

本人完全了解重庆文理软件工程学院有关保留、使用毕业论文的规定，即：软件工程学院有权保留送交论文的复印件，允许论文被查阅和借阅；软件工程学院可以公布论文的全部或部分内容，可以采取影印、缩印或其他复制手段保存论文。保密的论文在解密后遵守此规定。

作者签名：\_\_\_\_\_\_\_\_\_\_ 日期：\_\_\_\_\_\_\_\_\_\_\_