目录

[**1绪论 1**](#_Toc104541254)

[1.1 选题的目的和研究意义 1](#_Toc104541255)

[1.2 相关现状研究 1](#_Toc104541256)

[1.2.1国外现状研究 1](#_Toc104541257)

[1.2.2国内现状研究 2](#_Toc104541258)

[1.3 本文工作 2](#_Toc104541259)

[1.4 本文的章节安排 3](#_Toc104541260)

[**2技术选型的分析与研究 4**](#_Toc104541261)

[2.1 架构选型分析 4](#_Toc104541262)

[2.1.1 C/S架构介绍 4](#_Toc104541263)

[2.1.2 B/S架构介绍 4](#_Toc104541264)

[2.1.3 为什么采用B/S架构 4](#_Toc104541265)

[2.2 前端技术选型分析 4](#_Toc104541266)

[2.2.1 HTML、CSS、JavaScript 介绍 4](#_Toc104541267)

[2.2.2 VUE 介绍 5](#_Toc104541268)

[2.3 后端技术选型分析 5](#_Toc104541269)

[2.3.1 Java 介绍 5](#_Toc104541270)

[2.3.2 Java Web介绍 5](#_Toc104541271)

[2.3.3 Springboot 介绍 5](#_Toc104541272)

[2.3.4 MySql 介绍 6](#_Toc104541273)

[2.4 本章总结 6](#_Toc104541274)

[**3需求分析与数据库设计 7**](#_Toc104541275)

[3.1 需求分析 7](#_Toc104541276)

[3.2 数据库设计 8](#_Toc104541277)

[3.2.1 用户表 8](#_Toc104541278)

[3.2.2 菜单表 9](#_Toc104541279)

[3.2.3 组织表 9](#_Toc104541280)

[3.2.4 题库表 9](#_Toc104541281)

[3.2.5 大纲表 10](#_Toc104541282)

[3.2.6 问题表 10](#_Toc104541283)

[3.2.7 试卷表 11](#_Toc104541284)

[3.2.8 考生表 11](#_Toc104541285)

[3.2.9 考场表 12](#_Toc104541286)

[3.2.10 考试信息表 12](#_Toc104541287)

[3.2.11 答案表 12](#_Toc104541288)

[**4系统总体设计 14**](#_Toc104541289)

[4.1 用户注册功能 14](#_Toc104541290)

[4.2 用户登录功能 14](#_Toc104541291)

[4.3 系统管理功能 14](#_Toc104541292)

[4.4 题库相关功能 15](#_Toc104541293)

[4.5 试卷的组织功能 16](#_Toc104541294)

[4.6 考试功能 16](#_Toc104541295)

[4.7 判卷及统计功能 17](#_Toc104541296)

[**5系统详细设计 18**](#_Toc104541297)

[5.1 登录模块的实现分析 18](#_Toc104541298)

[5.1.1 JWT验证的实现 18](#_Toc104541299)

[5.1.2 邮箱验证登录是如何实现的 19](#_Toc104541300)

[5.2 功能(菜单)模块的实现分析 19](#_Toc104541301)

[5.3 组织机构管理的实现分析 21](#_Toc104541302)

[5.4 操作日志管理的实现分析 21](#_Toc104541303)

[5.5 用户管理的实现分析 22](#_Toc104541304)

[5.5.1 上传excel注册用户 22](#_Toc104541305)

[5.5.2 通过接口注册用户 23](#_Toc104541306)

[5.6 大纲管理的实现分析 24](#_Toc104541307)

[5.7 题库管理的实现分析 25](#_Toc104541308)

[5.8 题目管理的实现分析 25](#_Toc104541309)

[5.8.1 图片上传实现 25](#_Toc104541310)

[5.8.2 表单复用实现 26](#_Toc104541311)

[5.9 考试管理的实现分析 27](#_Toc104541312)

[5.9.1 自动组卷的实现 27](#_Toc104541313)

[5.9.2 考试的实现分析 28](#_Toc104541314)

[5.9.3 如何去收集试题的答案数据 29](#_Toc104541315)

[5.9.4 如何处理多人同时提交 29](#_Toc104541316)

[5.10 阅卷功能的实现分析 30](#_Toc104541317)

[5.11 系统的优化设计 31](#_Toc104541318)

[5.11.1 提交场景的防抖实现 31](#_Toc104541319)

[5.11.2 前端缓存http请求的实现 31](#_Toc104541320)

[**6系统的测试 32**](#_Toc104541321)

[6.1 注册登录相关测试 32](#_Toc104541322)

[6.2 函数管理、菜单相关测试 33](#_Toc104541323)

[6.3 组织管理相关测试 34](#_Toc104541324)

[6.4 大纲管理相关测试 35](#_Toc104541325)

[6.5 题库管理相关测试 37](#_Toc104541326)

[6.6 题目管理相关测试 37](#_Toc104541327)

[6.7 考试管理相关测试 38](#_Toc104541328)

[6.8 阅卷相关测试 39](#_Toc104541329)

[6.9 成绩统计相关测试 40](#_Toc104541330)

[6.10 组卷相关测试 41](#_Toc104541331)

[**总结 42**](#_Toc104541332)

[**致谢 43**](#_Toc104541333)

[**参考文献 44**](#_Toc104541334)

# 1绪论

## 选**题的目的和研究意义**

考试是教学过程中的重要组成部分[1]，传统的考试方式组织一次考试至少要经过人工出卷、考生考试、人工阅卷、成绩统计和试卷分析五个步骤。随着考试规模的不断扩大，工作量将会越来越大而且容易出错。因此开发一套在线的教考分离的考试系统具有重要的现实意义。

教考分离工作是高效考试工作改革的必然趋势[2]。教考分离的考试方式是指命题考试小组负责抽考和统考命题及组织评卷，任课教师负责出卷，并尝试建立试题库，使考试逐步走向规范化和制度化[3]，将过去某一课程由任课教师自己命题、自己评分的做法，改为由教学管理部门组织教学经验较为丰富的非任课老师依据教学大纲命题，或从规范、标准的试题库中筛选、组合出符合要求的试卷。并统一组织考试，统一评阅试卷。教考分离考试制度避免了随意性，减少了主观性，增大了考试成绩的可比性，增强了贯彻教学大纲的自觉性[4]，所以“教考分离”是适应当代教育发展的理想的考试模式[5]。

设计教考分离考试系统，把教学和考试分开，任课老师不参与所任课程的期末考试，依照教学大纲出题、建立题库、自动组卷和评卷。通过教考分离来检查教师教的效果[6]。

为了实现上述的功能，我对超星系统进行了仔细地分析，着重挑选了其中的用户管理功能、试题录入功能、组卷功能进行重点实现。

这个系统的设计综合了本科计算机科学与技术的各类知识，综合了理论知识和流行技术，还综合了理论知识和业务应用等一些技术的应用。能够体现出设计的专业性，通过此设计，对认识专业在行业内的应用情况和对系统搭建业务流程具有重大意义和作用。

## 相关现状研究

### 1.2.1国外现状研究

教考分离考试在国外主要集中在高等教育领域，且已经经历了一段时间的发展，有明显的正向效果。Zhiling Wu分析道：教考分离考试使学生在学习课堂内容的同时，更能主动地拓展学术知识和英语思维，从而明显减少了仓促取得高分的情况。对学生课程成绩的评估也会变得更加有效和公平[7]。同时De-rong指明，其教授的交通运输专业表演课的考试进行了教考分离，实践证明是可行的，它有助于提高本科生的教学质量[8]。高等教育考试实行教考分离，可以调动学生的学习积极性，使学生更好地掌握课程内容，提高教师的教学水平[9]。

### 1.2.2国内现状研究

我国是考试的发源地之一，在历经了“教考合一”这一考试制度的长期洗礼后， 各大高校在教学和考核方面也进行了相应的改革[10]，现在"教考分离"的考试制度正在逐步推进[11]。

教考分离的考试在我国有着众多的实践以及成果，武汉生物工程学院艺术系的吕道远指出考试命题、出卷、阅卷等各环节分开,是实现教学考试规范化及公平公正的目标所在,有利科学地开展教学质量评价和深化高等教育改革[12]。 盐城工学院将考试实施践行教考分离得出结论，对良好教学学风的形成具有较为明显的效果[13]。丽水学院推广教考分离的考试，提高了教学质量[14]。

赵欣认为建设高质量题库是实现教考分离考试系统的重要途径，能够提高考试的信度和效度，为教学单位改进教学提供正确反馈，优化教学质量管理[15]。

柯小玲指出教考分离是针对传统的教考合一存在的种种弊端而产生的一种创新性教学质量评价方式，是目前和今后高校考试改革的一个重要方向[16]。

国内的超星网，中国大学慕课等平台也相继推出了教考分离考试系统，已经应用到各个高校的教学工作中。

## 本文工作

本文说明了基于VUE+ SpringBoot 等技术实现的教考分离考试系统的原理。分析了为什么要实施教考分离以及教考合一的弊端，对教考合一、教考分离的考试系统国内外现状进行了相关研究、分析。明确了当前大部分学校的考试制度都是教考分离。也是本文的选题目的以及意义。详细说明了此项工作的技术选型考虑、重点实现功能分析、可行性分析、重难点解决过程分析以及自己对此项工作的一些思考。

## 本文的章节安排

本文对各个章节安排如下:

第一章： 分析教考合一考试的缺点，继而演变出教考分离考试。然后分析了教考分离考试系统的产生背景，其国内外现状以及此系统的好处。继而说明了此系统的大致开发工作。

第二章：教考分离考试系统的技术选型研究，使用什么技术能够实现此功能。为什么要采用B/S架构而不是C/S架构。在采用B/S架构的时候会遇到哪些问题，browser端和server端为什么要用VUE和 SpringBoot 框架，web端和后端的相关开发技术介绍。

第三章：教考分离考试系统的需求分析与数据库设计。

第四章：考试系统中的总体设计，大致的功能图。

第五章：考试系统中的详细设计，分析了其中的设计、实现原理。以及自己是通过什么方式、渠道去解决的。

第六章：教考分离考试系统的测试。

# 2技术选型的分析与研究

## **架构选型分析**

### C/S架构介绍

C/S架构全称为客户端/服务器体系结构，它是一种网络体系结构，其中客户端是用户运行应用程序的PC端或者工作站，客户端要依靠服务器来获取资源。

### B/S架构介绍

B/S架构即浏览器和[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)架构模式，B/S架构浏览器通过[Web Server](https://baike.baidu.com/item/Web%20Server)同数据库进行数据交互。大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本(TCO)。

### 为什么采用B/S架构

C/S 一般建立在专用网络上，小范围里的网络环境。

B/S 建立在广域网上， 面向不同的用户群，分散地域，且不用考虑操作系统的兼容性。

因为Chrome浏览器已经实现了对Windows、Linux、MacOs的跨平台访问，为了尽可能减少用户的使用负担，方便用户随时随地使用不同设备、不同操作系统对此系统进行访问，本文的架构选型选择了B/S架构。

## 前端技术选型分析

### HTML、CSS、JavaScript 介绍

1. HTML [17]

超文本标记语言（英语：HyperText Markup Language，简称：HTML）是一种用于创建网页的标准标记语言。HTML 运行在浏览器上，由浏览器来解析。

1. CSS

CSS(Cascading Style Sheets) 主要用来渲染和修饰HTML。

1. JavaScript

JavaScript是一个运行在浏览器运行时的一个动态解释型语言。这也就决定了JavaScript只能运行在浏览器之上，各个厂家的浏览器内核不同，为了统一JavaScript的运行时标准，欧洲计算机协会指定了ECMAScript 规范。保证各个厂家的浏览器都实行同一套标准。此次系统中JavaScript主要用来进行 IO操作，网络操作，渲染操作。

### VUE 介绍

HTML、CSS、JavaScript已经能够实现此系统的功能。但是实现起来相当地繁琐，并且也不易于更新与维护。为了实现此系统，选用了一款渐进式MVVM前端框架VUE。VUE是一个数据驱动的MVVM框架[18]。对比于传统的 MVC框架，其中的VM(view model)等同于MVC框架中的 controller，view是HTML模板，model是JavaScript中需要渲染到HTML上的变量， VM是一个中间件，将 view和model进行双向绑定，便于将model的数据渲染到view上，同时让view的改变映射到model中，利用设计模式中的【发布/订阅模式】实现了双向绑定，极大地方便了前端页面的开发工作。

## 后端技术选型分析

### Java 介绍

Java是一门面向对象的编程语言，具有功能强大和简单易用两个特征。同时，Java生态比较成熟，有众多的工具包和解决方案，在遇到问题的时，能够快速去社区通找到解决方案。

### Java Web介绍

Java Web 应用程序是动态资源（例如 Servlet、JavaServer Pages、Java 类和 jar）和静态资源（HTML 页面和图片）的集合。Java Web 应用程序可以部署为 WAR （Web ARchive）文件。

### Springboot 介绍

SpringBoot 内置了一个 Tomcat 服务器，也就是内嵌了一个 Servlet Container，因此直接运行工程就可以看到效果，不需要再进行额外的部署，也无需进行 Servlet 的有关配置[19]。同时 Springboot 还会为用户预设许多配置 。简单来说，Springboot 就是帮我在Java Web 开发中简化了很多配置。

### MySql 介绍

MySQL是一个关系型数据库[20]，并与 PHP、Java 等主流编程语言紧密结合。它简单易用并且有免费的社区版供我使用，体积小，安装方便，易于在本机和服务器进行调试。

## 本章总结

本章节详细介绍了从系统架构到前后台、数据库的技术选型工作。讨论了过时技术与新兴技术的利弊权衡，确定了考试系统的开发技术路线。

# 3需求分析与数据库设计

## 需求分析

系统主要的功能说明如下。

1. **注册登录功能**

管理员可以通过接口和文件注册新的用户。登录功能保持用户的登录状态达到认证的效果。

1. **组织机构管理功能**

每个用户的组织可以修改，同时管理员可以编辑、删除组织。在发卷的时候是以一个组织为单位进行考试的发放。

1. **功能模块管理功能**

功能管理模块实际上是菜单管理模块，根据用户的权限来展示不同的菜单。管理员可以对菜单进行增删改。

1. **用户管理功能**

管理员可以对用用户进行增删改。

1. **操作日志管理功能**

涉及到高权限的操作会进行日志的上报。

1. **大纲功能**

大纲相当于知识点，自动组卷是以大纲和题目难度为准设计的适应度函数。

1. **题库功能**

题库是具有相同大纲的一类题目的集合，可以加密，加密的题库需要密码才能访问。

1. **组卷功能**

采用了遗传算法的自动组卷功能能更好地让试卷贴合大纲，具有最好的适应度，同时减轻老师的工作量。

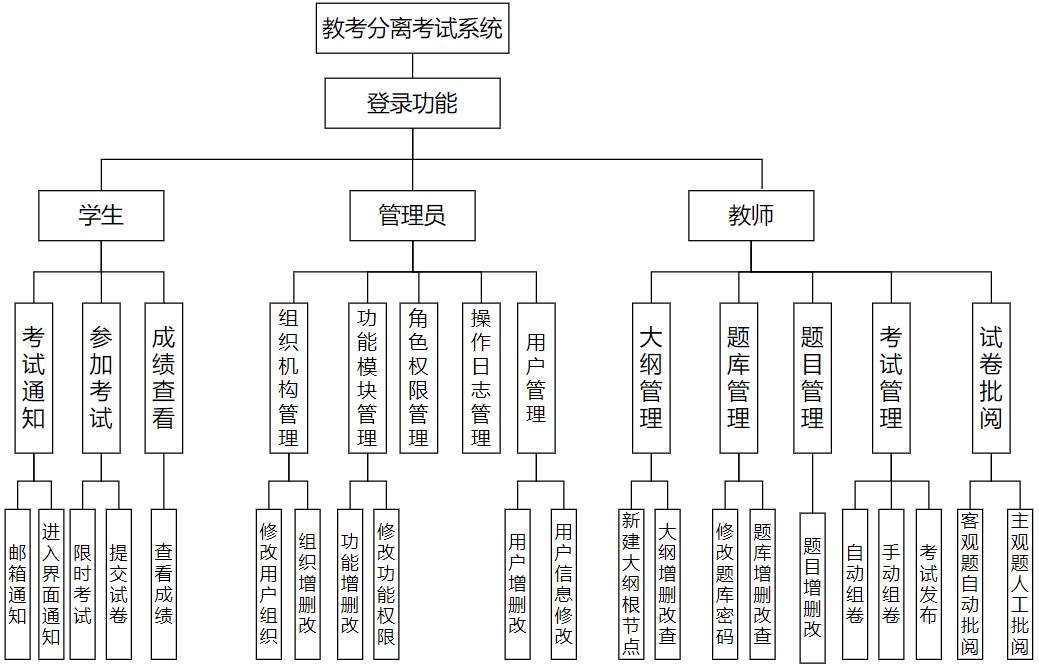
1. **考试功能**

教师根据班级来发放试卷，学生限时考试，后端处理学生提交的答案。

1. **批阅功能**

实现了客观题主动批阅，主观题手动批阅的效果。

具体功能如图3.1所示。



**图3.1 教考分离功能图**

## 数据库设计

根据前面的需求分析，确定实体，实体属性，实体与实体之间的联系得到系统的总体设计。具体的表的设计如下文所示。

### 用户表

用户表如表3.1所示。

**表3.1 用户表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| user\_id | 用户id | bigint |
| user\_name | 用户名 | varchar(20) |
| user\_realname | 用户真名 | varchar(30) |
| user\_pwd | 用户密码 | varchar(50) |
| user\_role | 用户身份 | tinyint |
| user\_email | 用户邮箱 | varchar(30) |
| user\_phone | 用户手机号 | varchar(11) |
| user\_orgcode | 用户组织代码 | varchar(20) |
| user\_registertime | 用户注册时间 | datetime |

### 菜单表

菜单如表3.2所示。

**表3.2 菜单表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| menu\_id | 菜单id | int |
| menu\_name | 菜单名 | varchar(30) |
| menu\_level | 菜单级别 | int |
| menu\_parentid | 菜单父级id | int |
| menu\_click | 菜单路径 | varchar(100) |
| menu\_usable | 菜单是否可用 | tinyint |
| menu\_right | 菜单对应的权限 | varchar(10) |

### 组织表

组织如表3.3所示。

**表3.3 组织表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| org\_id | 组织id | int |
| org\_code | 组织代码 | varchar(30) |
| org\_name | 组织名 | varchar(30) |
| org\_parentid | 组织父级id | int |

### 题库表

题库如表3.4所示。

**表3.4 题库表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| bank\_id | 题库id | int |
| bank\_name | 题库名 | varchar(20) |
| bank\_count | 题库数量 | int |
| bank\_userid | 题库创建者id | bigint |
| bank\_isencrypt | 是否加密 | tinyint |
| bank\_password | 加密密码 | varchar(10) |
| bank\_outlineroot | 题库对应的大纲 | int |

### 大纲表

大纲如表3.5所示。

**表3.5 大纲表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| nodeId | 大纲节点id | int |
| nodeName | 大纲名字 | varchar(20) |
| outlineIndex | 大纲序号 | varchar(20) |
| parentId | 大纲父节点id | int |
| userId | 大纲创建者id | bitint |
| level | 大纲级别 | tinyint |

### 问题表

问题如表3.6所示。

**表3.6 问题表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| id | 问题id | int |
| difficulty | 问题难度 | double |
| questionType | 问题类型 | varchar(20) |
| question | 问题 | text |
| answer | 问题答案 | varchar(1000) |
| analysis | 问题分析 | text |
| topic | 问题主题 | varchar(20) |
| questionDate | 问题时间 | datetime |

### 试卷表

试卷如表3.7所示。

**表3.7 试卷表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| paperId | 试卷id | int |
| examCode | 考试代码 | varchar(10) |
| paperName | 试卷名称 | varchar(100) |
| questionTypes | 问题类型 | varchar(200) |
| isRelease | 是否发放 | tinyint |
| outlineRootNodeId | 试卷对应大纲节点 | int |
| questionsNum | 题目id | int |
| difficulty | 试卷难度 | tinyint |
| userId | 试卷创建者id | bigint |
| createDate | 试卷创建时间 | datetime |
| releaseNum | 发放数量 | int |
| deleted | 是否已经删除 | tinyint |

### 考生表

考生如表3.8所示。

**表3.8 考生表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| id | 考生表id | bigint |
| examNumber | 考试id | varchar(10) |
| examnieeStuId | 考生id | bigint |
| examRoomId | 考场id | int |
| startTime | 开始时间 | datetime |
| submitTime | 结束时间 | datetime |
| checked | 是否进入 | tinyint |
| studentStatus | 考生状态 | tinyint |
| score | 分数 | int |

### 考场表

考场如表3.9所示。

**表3.9 考场表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| id | 考场id | bigint |
| Course | 课程 | varchar(50) |
| Examiner | 考生 | varchar(20) |
| Number | 考生数量 | int |
| currentNumber | 当前进入的考生数量 | int |
| eaxmNumber | 考试id | int |

### 考试信息表

考试信息如表3.10所示。

**表3.10 考试信息表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| id | 表id | int |
| outlineRootNodeId | 大纲id | int |
| paperId | 试卷id | int |
| examcode | 考试id | int |
| eaxmName | 考试名称 | varchar(200) |
| examRoom | 考场号 | int |

### 答案表

答案如表3.11所示。

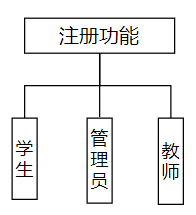
**表3.11 答案表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 含义 | 字段类型 |
| id | 表id | int |
| examId | 考试id | int |
| examineeId | 考生id | bigint |
| examNumber | 考试序号 | varchar(20) |
| paperNo | 试卷id | int |
| questionId | 问题id | int |
| questionType | 问题类型 | varchar(20) |
| correctAnswer | 正确答案 | varchar(1000) |
| correctScore | 正确分数 | tinyint |
| questionIndex | 问题序号 | varchar(10) |
| doAnswer | 考生答案 | varchar(1000) |
| checked | 是否作答 | tinyint |
| score | 分数 | tinyint |

# 4系统总体设计

## 用户注册功能

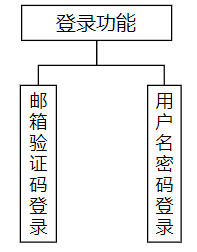
用户注册功能如图4.1所示。



**图4.1 用户注册模块**

## 用户登录功能

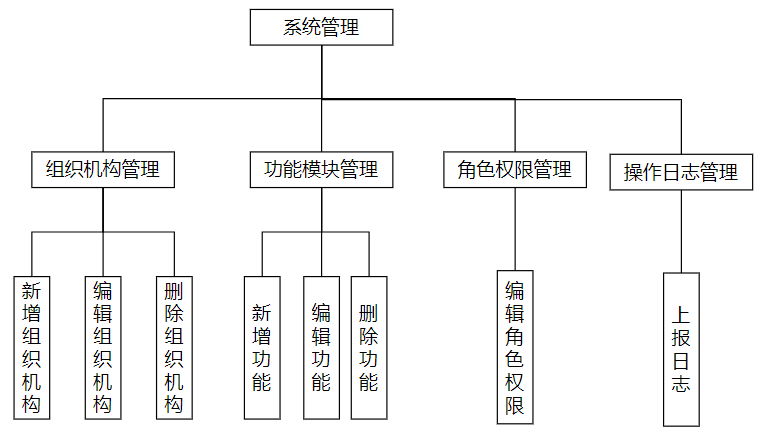
用户登录功能如图4.2所示。



**图4.2 用户登录模块**

## 系统管理功能

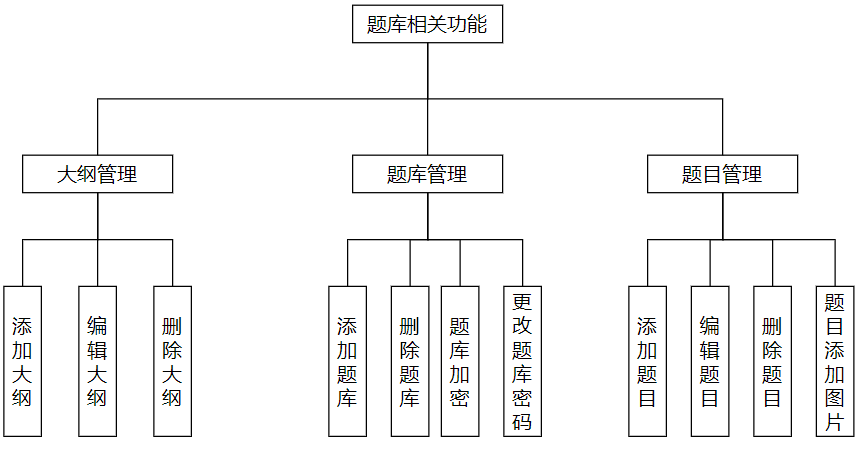
系统功能如图4.3所示。



**图4.3 系统功能示意图**

## 题库相关功能

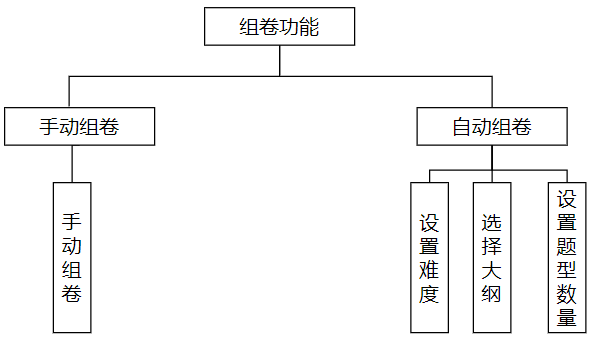
题库相关功能如图4.4所示。



**图4.4 题目相关功能**

## 试卷的组织功能

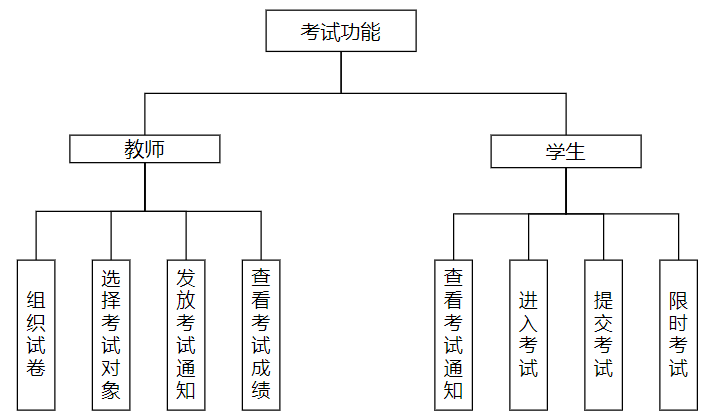
试卷的组织功能如图4.5所示。



**图4.5 试卷组织功能**

## 考试功能

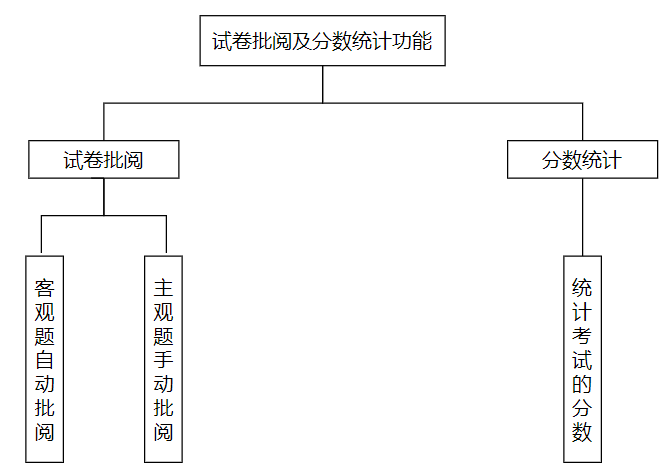
考试功能如图4.6所示。



**图4.6 考试功能**

## 判卷及统计功能

判卷及统计功能如图4.7所示。

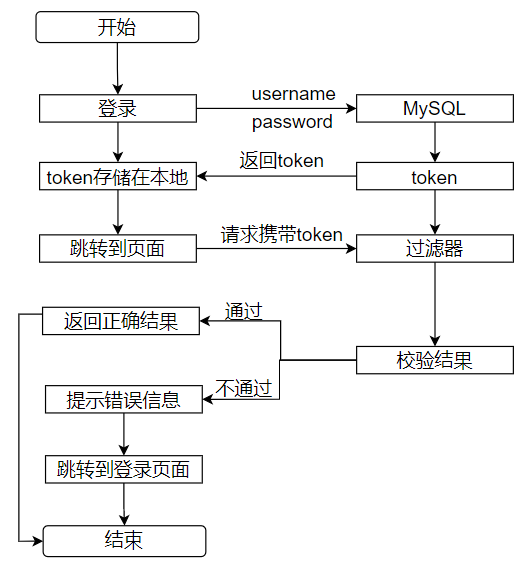


**图4.7 判卷及统计功能**

# 5系统详细设计

## **登录模块的实现分析**

前后端访问控制都对用户登录做出了要求，实现了对登录状态的全局拦截。这里采用的方法是传统的JWT(Json Web Token)验证模式。主要的难点有邮箱验证登录的实现和JWT的验证，整体流程图如图5.1所示。



**图5.1 登录流程图**

### JWT验证的实现

用户通过账号密码或者邮箱验证码登录系统后，服务器给他下发一个令牌(token)，下一次用户再次通过 Http 请求访问服务器的时候，把这个 token 通过 Http header 或者 url 带过来进行校验。为了防止伪造，数据加上一个只有自己才知道的密钥，做签名，将数据和签名一起作为 token 发送过去。因此可以不用保存 token ，因为发送给用户的令牌里，已经包含了用户信息。当用户再次请求过来的时候使用同样的算法和密钥对这个 token 中的数据进行解密，如果加解密后的结果和 token 中的签名一致，那就可以进行鉴权，并且也能从中取得用户信息。

根据上述流程，分别在前后端设置了全局的拦截器，对token进行全局拦截，如果用户已经登录，会自动地在http请求头中携带token信息传递给后端，后端验证信息通过就会返回对应地response，而如果用户未登录的时候访问需要登录的接口，后端拦截到没有token信息，后端就会在http response返回一个未登录的信息，前端设置的全局http拦截器就会拦截到这个未登录的response，然后重定向到登录页面。

### 邮箱验证登录是如何实现的

1. 发送验证码

在需要发验证码的邮箱设置中，开启IMAP/STMP服务，然后会获得一个第三方授权码，在Java的mail包中，有一个验证授权码的工具函数，验证成功之后即可向指定邮箱发送验证消息。

1. 缓存验证码

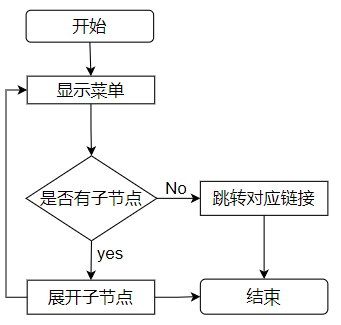
为了防止用户多次发起发送验证码的请求，设置了一个验证码的有效时间，在有效时间内前端上限制用户重新发送请求。同时后端将邮箱和验证码以key-value的形式存放到Redis(Remote Dictionary Server)中，并设置一个过期的时间。在过期时间之前这个验证码都是有效的，过期之后这个验证码就会被删除，然后用户重新发起发送验证码的请求即可。

## 功能(菜单)模块的实现分析

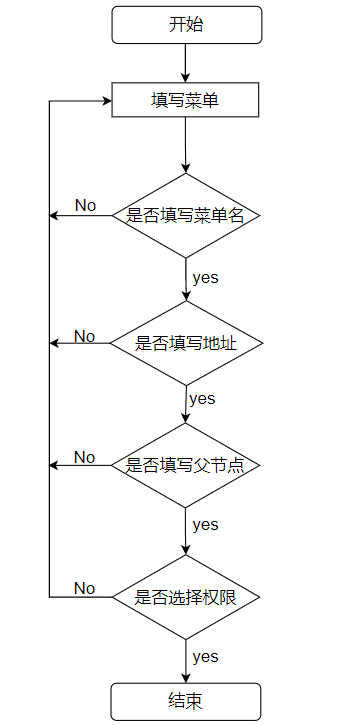
不同的角色会有着不同的功能，也就是不同的菜单。在菜单的添加页面，可以在功能权限上选择是否对某种身份的用户开放此功能，所以在后端的数据库中，会存放一个字段来表明用户的身份标识，因此设计了一个接口，根据用户的身份信息从数据库中返回不同的菜单信息给前端。

但是，这么做有一个弊端，每次查看菜单都需要对用户的身份信息进行解析，比较麻烦。所以就将菜单信息和用户身份信息存储在浏览器中，在进行用户身份切换的时候，再从浏览器中取出菜单根据用户的身份信息进行过滤即可。这个菜单是一个树状的结构，所以用到了递归算法将其进行转换。为没有子节点的菜单项绑定点击事件为路由跳转，为有子节点的菜单项绑定的事件则为展开子节点。菜单显示流程如图5.2所示。

在对菜单进行编辑的时候，右键某一项，会获取当前项的菜单名，父共id，功能调用地址，功能权限等，回填到编辑信息中。 菜单填写流程如图5.3所示。



**图5.2 菜单显示流程**



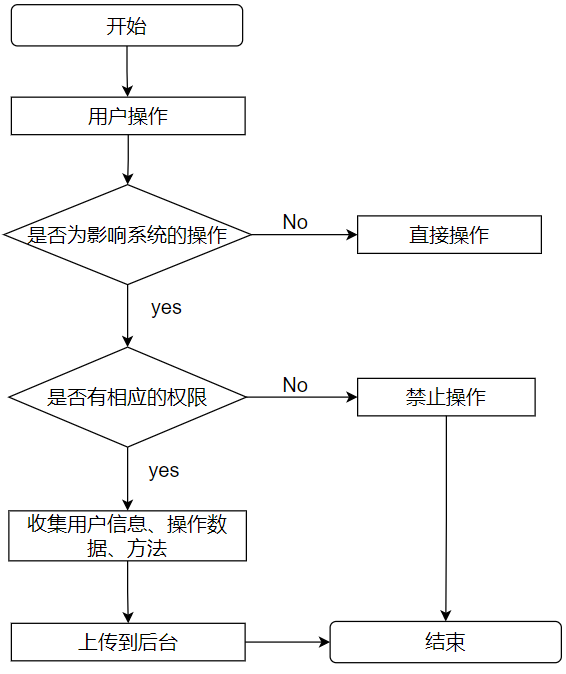
**图5.3 菜单填写流程**

## 组织机构管理的实现分析

组织机构是一个树状结构，除root组织外，每一个组织都有一个parentId指向其父级组织的id。如计算机科学学院的父组织为西南石油大学，因此在组织表的设计中，将parentId作为外键指向了组织表的id，这样在删除某个组织时，可以级联将其子组织删除，而不需要设置额外的触发器。

## 操作日志管理的实现分析

因为系统设置了不同的用户，在涉及到某些对系统安全性造成影响的操作如删除组织的时候，第一是会做权限限制，第二是会收集操作人、操作数据、操作方法的数据作为日志上报到后台，以便后期的溯源查询。具体流程如图5.4所示。



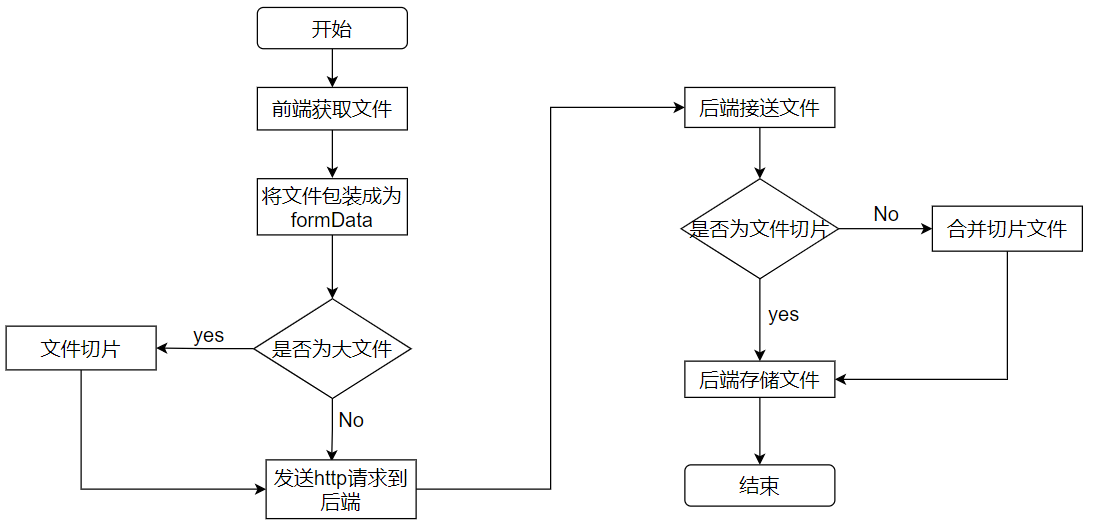
**图5.4 日志上报流程图**

## 用户管理的实现分析

用户管理的重点主要是用户注册，以下着重对用户注册的实现进行分析。

### 上传excel注册用户

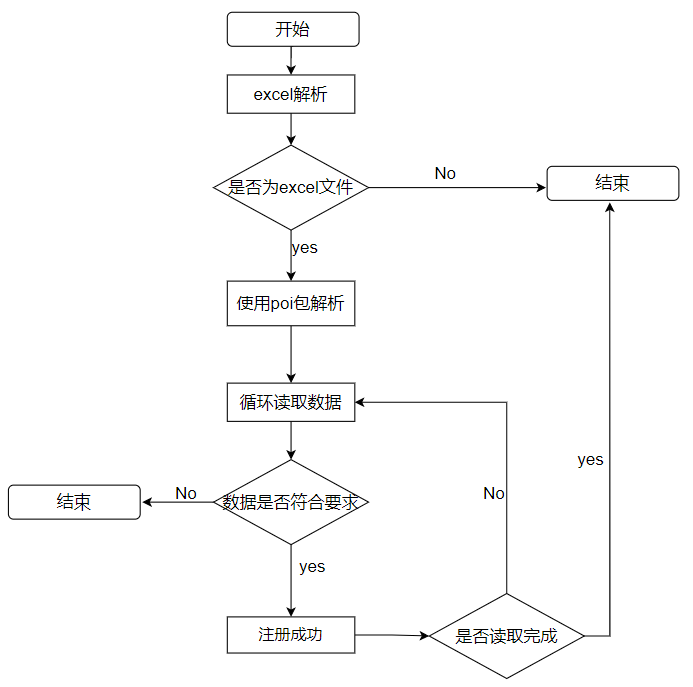
在实际的应用当中，学生和老师都是成百上千个，如果管理员一个个手动去添加用户的信息，会非常耗时耗力，且极有可能出错，因此设计了一个通过excel添加用户的功能。实际的应用场景就是管理员下发一个在线协同的excel表，让用户按照规范填写自己的信息，然后将这张表上传到这个系统中就可以添加新的用户了。具体上传流程如图5.5所示。



**图5.5 文件上传流程**

首先前端的浏览器会调用系统的文件API，去获取拿到的文件，因为后面还有上传图片的需求，将上传做成了一个组件。拿到文件之后，前端以formData的形式将文件放入http请求的请求体中，后端拿到文件之后，使用apache 的 poi包对excel进行解析，将excel中的 workbook 对象转换为 csv格式的对象，然后再转换为普通的java 对象就可以对数据进行读取了，拿到数据后，就是对数据进行批量处理的过程，批量注册用户。Excel注册流程图如图5.6所示。

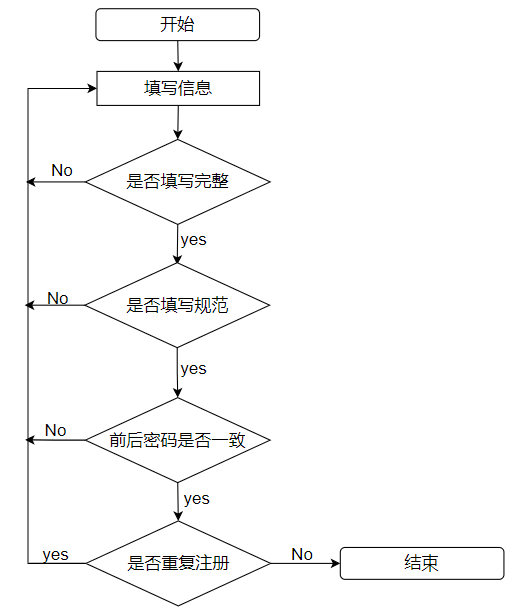
因为用户的数量庞大，excel文件可能会比较大，所以在前后端实现了大文件上传的功能。具体做法是在前端对大文件进行切片，将切片通过http串行传送到后端的服务，后端接收到切片会为每一个切片生成一个文件，在最后一个切片到达的时候，后端对整个临时文件夹的切片进行合并就完成了大文件的上传。



**图5.6 excel注册流程图**

### 通过接口注册用户

除了上传excel注册用户外，还可以使用常规的界面填写表单的形式注册用户。在使用表单注册的时候，为了防止网络攻击和用户填写不规范的数据对数据库造成影响，会使用正则表达式对用户输入的数据进行验证和过滤。若用户填写的信息未通过验证，前端界面会对不规范的字段进行标红，且不会发起HTTP请求。最大程度上保证了数据库的安全。其注册流程图如图5.7所示。



**图5.7 接口注册流程**

## 大纲管理的实现分析

大纲等同于一本书的目录章节，也是一个树状的结构，其实现方式与前面的组织管理类似，这里的重点在于，在新建试题和组卷的时候都会选择大纲，需要做一个级联选择器来选择大纲或者大纲中的某一节。

这里采用的是递归组件的形式来制作级联选择器，首先会通过DFS算法得到某个节点及其所有子节点，将树状结构转换成为对象的形式，这个对象的结构为{ rootName: ‘a’, children: [] }，可以看到，每个节点都会有children字段，如果节点的children字段不为空，就递归渲染此组件。

每个组件有select和expand方法，在expand的时候，会记录上一级的rootName，在select的时候，会将之前记录的rootName的值进行拼接，得到一个选择的路径，这样就做好了一个级联选择器。

## 题库管理的实现分析

题库是属于同一类大纲的题目的一个集合，需要将大纲表和题目表连接起来，因此单独设计了一个表存储大纲id、题库id和题目的id，用于三个表之间的连接。题库可以加密，因此在请求某个加密题库的题目时，需要通过验证之后才能返回题目，否则禁止访问此题库。

## 题目管理的实现分析

整个题库有着不同的题目，包括选择题、填空题、简答题等。这些文本的存储整体来说不是很难，但是，有的选项或者题目需要上传一些图片上去，如何上传与保存图片是一个难题。选择题与填空、简答题如何复用表单也是一个难题。

### 图片上传实现

起始时，题目、选项、答案部分的数据库字段仅有文本部分，并未添加图片部分，后来在开发的过程中，参考到超星、慕课、腾讯课堂等平台的组卷系统，发现大多数平台的试题组织(题目)在添加题目的时候，都做了富文本的处理。包括文本的加粗、插入表情、插入图片等等。考虑到富文本的存储复杂以及前端实现的难度，因此放弃了添加富文本的计划，仅保留一个插入图片的功能。上传图片的方法分为base64和文件传输。

1. 使用base64进行传输

一张图片是由一个个像素点组成的一组像素集合，也就是说在计算机中，图片的存储方式是一个个像素点，也就是成千上万的点阵信息。在前端拿到图片的点阵信息之后，可以通过base64编码将其转换为base64的字符串，方便在HTTP中进行传输，并且不用通过文件的形式在后台对图片进行读取操作，响应的时间及其迅速。但是，使用base64有一个弊端，就是如果图片所占用的内存太大，经过转换后的base64字符串长度有可能超过数据库那个字段的限制，会导致读取的时候出现错误，同时如果用户上传了多张相同的图片，数据库不会对此进行去重操作，所以会导致数据库的数据冗余。

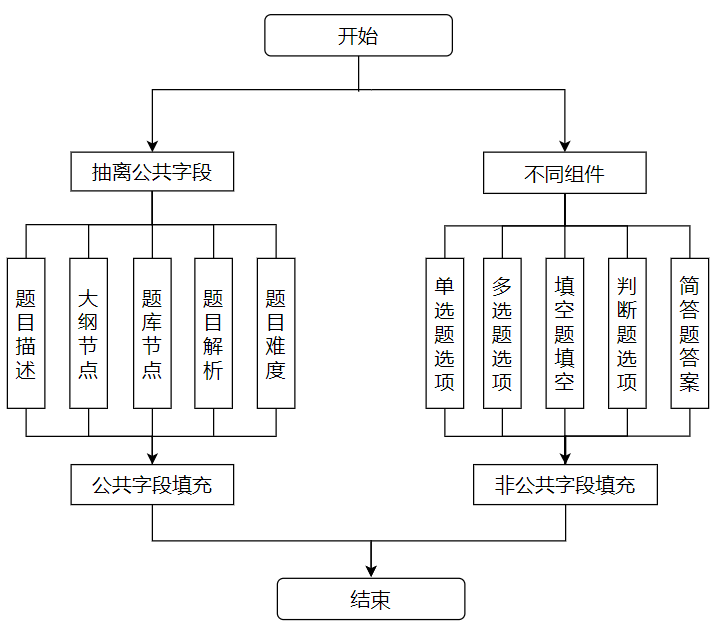
1. 使用文件+数据库操作

Base64传输的过程可能会失去一些图片信息，所以没有采纳base64的方式对图片进行传输，而是采用了文件操作的方式，在前端发出http请求，将二进制文件传递到后端的时候。后端将其存储起来，并依次遍历已有的文件，获取其md5的值，依次与此次上传文件的md5值进行比较，如果有重复的就放弃生成图片文件，依然沿用之前的文件。确定了文件的位置之后，生成http格式的文件链接并存储此文件的位置链接至数据库中，下发此文件的链接返回给前端，链接也是一个字符串，但是这个链接的长度远远小于base64编码后的字符串的长度，因此在数据库中的存取不会出现任何问题。

综上所述，系统采用文件加数据库的方式进行图片的传输。

### 表单复用实现

目前题库中的题目类型多种多样，如果为每一种题目开启一个新的页面进行题目的添加的话，会做很多重复性的工作，如果需要复用一些表单字段，如何最大限度地复用，也是一个值得考量的问题。表单复用流程如图5.8所示。



**图5.8 表单复用流程图**

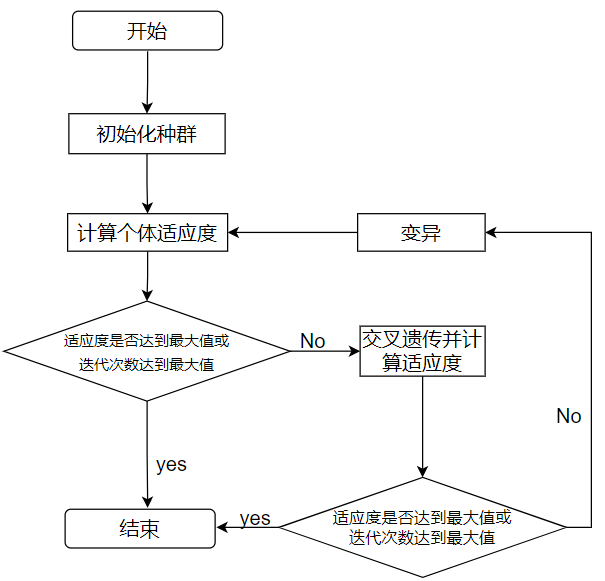
首先，每一种类型的题目都有一些共同的字段如题目描述、大纲节点、题库id、题解答案等，将这些通用的表单做成一个组件，嵌入在各个类型的题目中，作为公共字段进行提交。各个不同类型的题目做成不同类型的组件，既方便对其进行修改，也能保证表单最大限度的复用。题目的信息回填也是分不同的组件进行回填，保证不同字段之间的解耦，增强程序的健壮性。

## 考试管理的实现分析

### 自动组卷的实现

自动组卷就是用户自己选择想出的题型及其数量，设置难度，知识点分布(大纲)，系统自动生成试卷。这里主要用到了遗传算法的相关知识。

遗传算法是一种启发式搜索，其灵感来自查尔斯·达尔文的自然进化论。该算法反映了自然选择的过程，其中选择最适合的个体进行繁殖以产生下一代的后代。遗传算法流程图如图5.9所示。



**图5.9 遗传算法流程图**

在应用中，遗传算法的难点主要有三个，染色体的编码的设计、适应度函数的设计，接下来，分析这两个难点的实现。

1. 染色体的编码的设计

采用实数编码方案，将一份试卷映射为一个染色体，组成该试卷的每道题的题号作为基因，基因的值直接用试题号表示，每种题型的题号放在一起，按题型分段，在随后的遗传算子操作时也按段进行，保证了每种题型的题目总数不变。

1. 适应度函数的设计

适应度函数跟试卷难度系数和知识点覆盖率有关。

试卷难度系数公式：P=∑Di×Si/∑Si；其中i=1，2，…N，N是试卷所含的题目数，Ｄi，Ｓi分别是第i题的难度系数和分数。

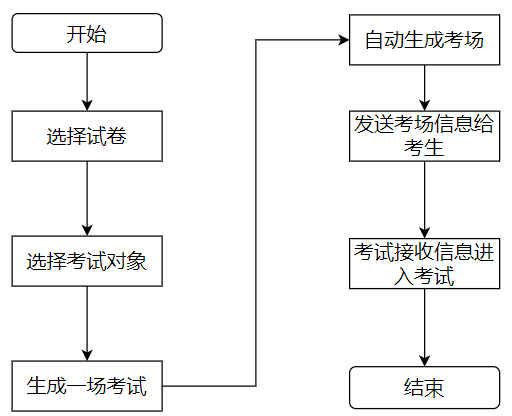
知识点覆盖率系数公式：D = (1 – M/N)，例如期望本试卷包含N个知识点，而一个个体中所有题目知识点的并集中包含M个（M<=N），则知识点的覆盖率为M/N。用户的期望难度系数EP与试卷难度系数P之差越小越好，知识点覆盖率越大越好，因此适应度函数为：f=1- D \*f1-|EP-P|\*f2。其中f1为知识点分布的权重，f2为难度系数所占权重。当f1=0时退化为只限制试题难度系数，当f2=0时退化为只限制知识点分布。这里(1- M/N)是因为覆盖率越大，M/N这个值就越大，因为适应度是一个正向的反馈值，所以要用(1-M/N)。

所以仅需要输入好试卷标题、试卷难度、组卷数量、各类型题目的题型和数量之后，系统将会根据输入的试卷难度和课程设置的大纲下的各个章节的掌握度，自动计算出整张试卷每个章节的题目占所有题目的权重，在最后得到符合输入的试卷整体难度的试卷。

通过遗传算法，将多张试卷进行交叉遗传并变异，直到难度系数与用户难度系数相当，知识点覆盖率最大的试卷为止。

### 考试的实现分析

考试流程如图5.10所示。教师选择好试卷、学生之后形成一场考试，会自动分配一个考场，然后系统会发送通知给分配到考试的学生。学生接收到通知进入考试界面后，系统会设置一个倒计时来题型学生注意时间，倒计时时间到，系统会自动交卷并退出考试页面。其中的难点在于收集答案数据和处理多人同时提交。



**图5.10 考试流程**

### 如何去收集试题的答案数据

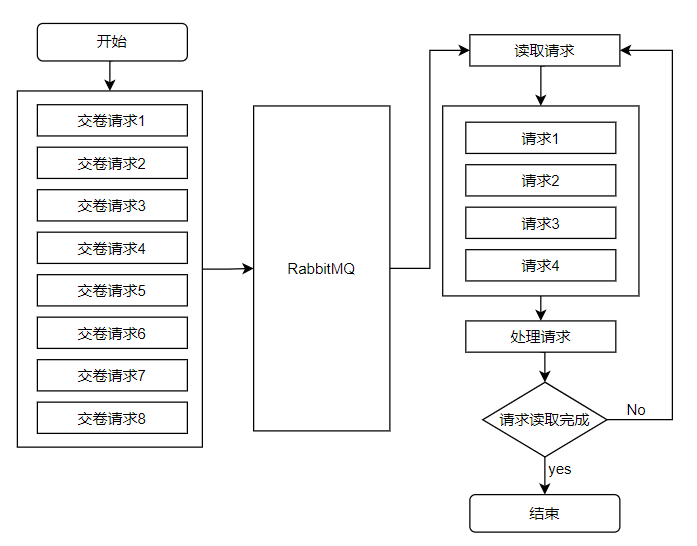
首先在显示试卷上，复用了之前题目管理的试题组件，不再为每一道题目单独写template模板，给每一个题目类型的组件添加一个 returnAnswer 的方法，返回该题目的答案。

考试的试卷不同于常规的表单，数据是固定的，因为VUE是一个MVVM框架，数据与HTML模板进行了双向绑定，可以通过双向绑定获取输入框地内容，所以一开始可以将formData与表单进行强绑定。试卷的试题不是固定的，且试题类型不统一，给收集试题答案的数据带来了极大的困难。

因此设计了一种思路去动态地收集试卷的数据。首先进入考试界面通过HTTP请求拿到试题的数据之后，将数据转换为一个表单数组，这个数组的长度就是此张试卷试题的数量，然后通过表单数组去动态地自定义渲染模板中的数据，重新进行数据的绑定。数组中的每一个元素都绑定到了对应的组件上，就可调用组卷的returnAnswer方法获取数据，这样试卷的表单就和普通的表单一样，有着固定的字段绑定，无论是修改还是提交都非常地方便。

### 如何处理多人同时提交

在考试的时候，会遇到很多同学同时提交试卷的情况，这个时候后端是无法承受住如此高的并发量的，所以此时用到了RabbitMQ(Message Queue)高级消息队列协议。具体流程如图5.11所示。RabbitMQ相当于一个缓存的中间件，在多名同学同时发出交卷请求的时候，会将这个时刻所有的请求体的内容存到RabbitMQ中，而不是直接到后台去进行数据库的添加工作，因为数据库每秒执行的SQL数量有限。然后后台系统就从RabbitMQ中一条一条地读取这些数据，再进行操作，既没有影响用户体验，也很好地保护了服务器资源。



**图5.11 RaddbitMQ处理消息流程**

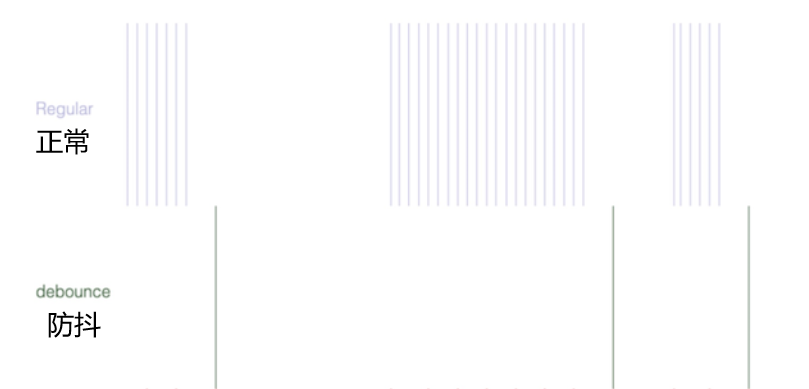
## 阅卷功能的实现分析

在学生提交完试卷之后，就会进行到阅卷的工作了。系统会自动将试卷中的客观题部分答案与题库中的答案进行对比，完成客观题的自动阅卷。主观题部分，会将每一道题分配给不同的用户(教师、学生)去进行批阅，因此在题目表中设计了一个totalSize字段来设置批阅的人数，同时设置completedNum来统计已经完成批阅的人数，在completedNum 等于totalSize 的时候，会将多人的批阅的分数总和求平均值，就是当前题目的分数了。

## 系统的优化设计

### 提交场景的防抖实现

大部分的提交按钮都是携带的表单数据进行HTTP请求，因为后端的处理能力有限，如果用户重复提交的话，可能会导致后端的服务宕机，所以这个时候就需要前端对提交做限制，防止用户在某一个时间内重复提交，也就是所谓的防抖。其示意图如图5.12所示。



**图5.12 防抖示意图**

可以看到debounce减少了很多的http请求，其原理就是在http请求发出后的n秒内，如果再次发出同样的请求，则重新计算时间（就是在触发某个事件后，在下一次触发之前，中间的间隔时间如果超过设置的时间才会发送请求，一直触发就不会发送请求）。

### 前端缓存http请求的实现

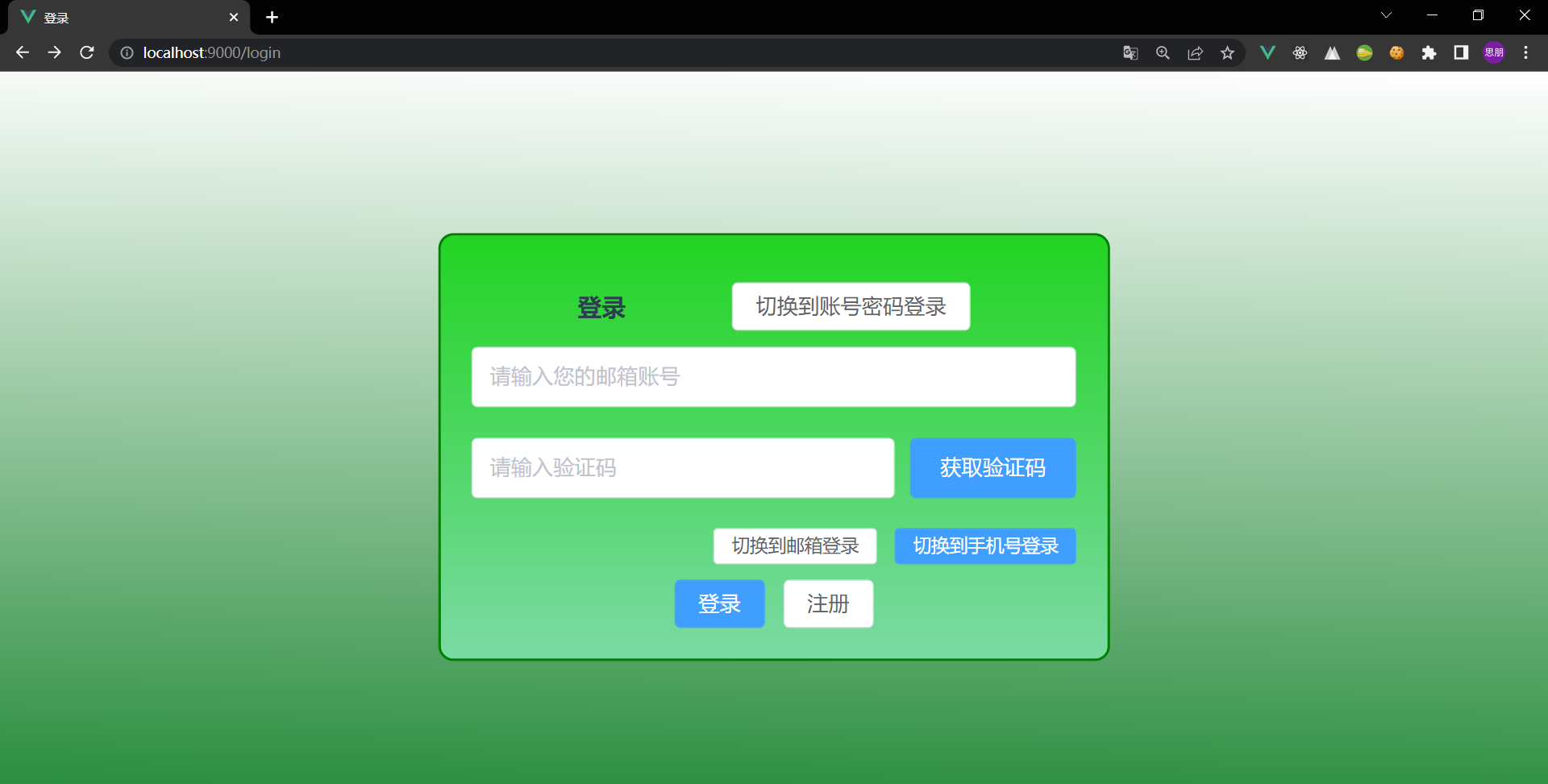
在开发的时候，后端的idea和spring boot启动速度非常慢，而且非常占用电脑的内存，一旦后端服务跑起来，遇到接口问题再重启的时候是一件非常耗时的事情，前端的大多数请求都是简单的查数据库表，不涉及到后端进行处理。所以可以将这类请求用service worker缓存起来，在下一次发起这类请求得时候，service worker会直接将缓存的请求拦截，返回上一次请求的数据。这样既方便前端进行测试，也节省了内存资源和时间。

# 6系统的测试

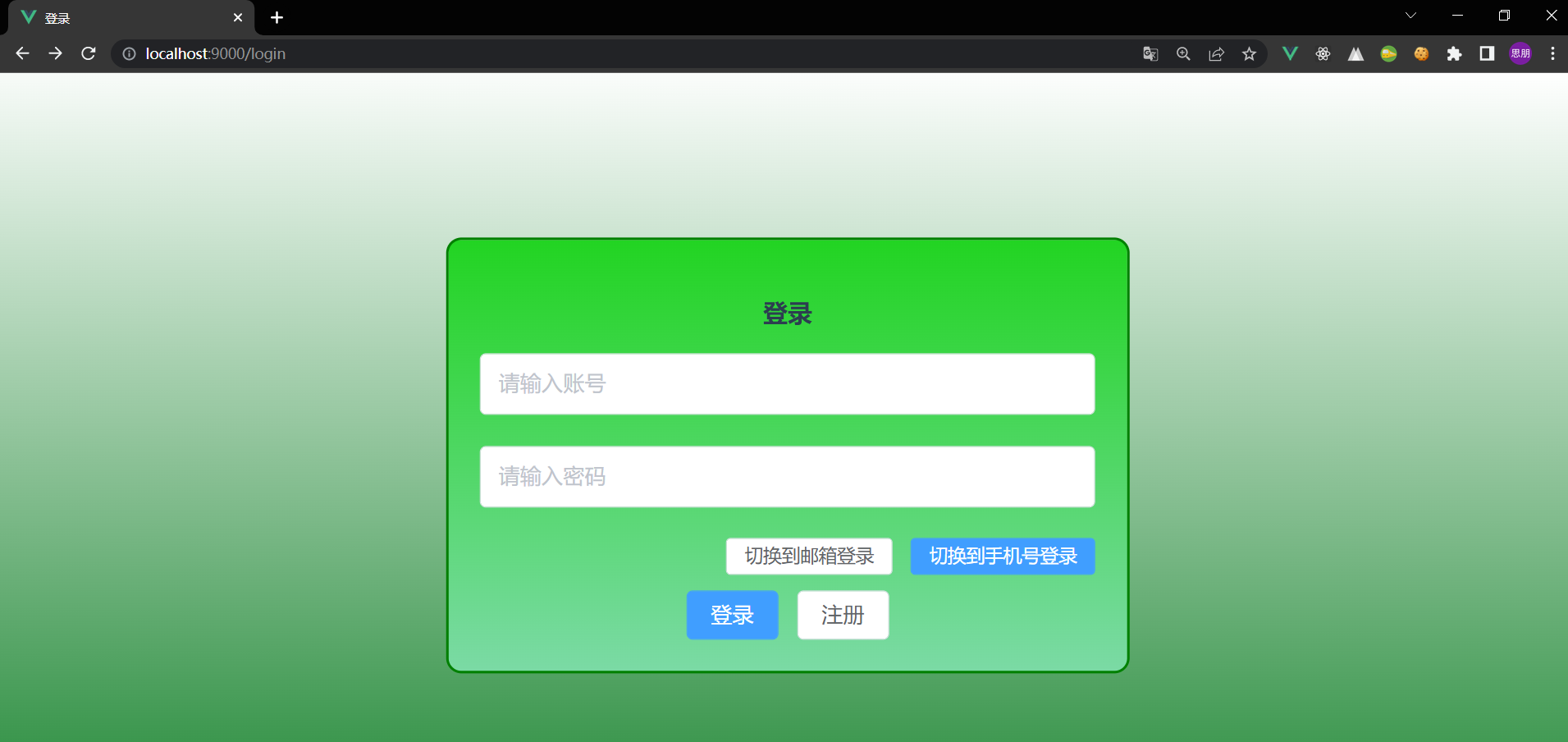
## 注册登录相关测试

注册登录测试截图如图6.1，6.2，6.3所示，测试结果如表6.1所示。



**图6.1 注册截图**

**图6.2 邮箱登录截图**



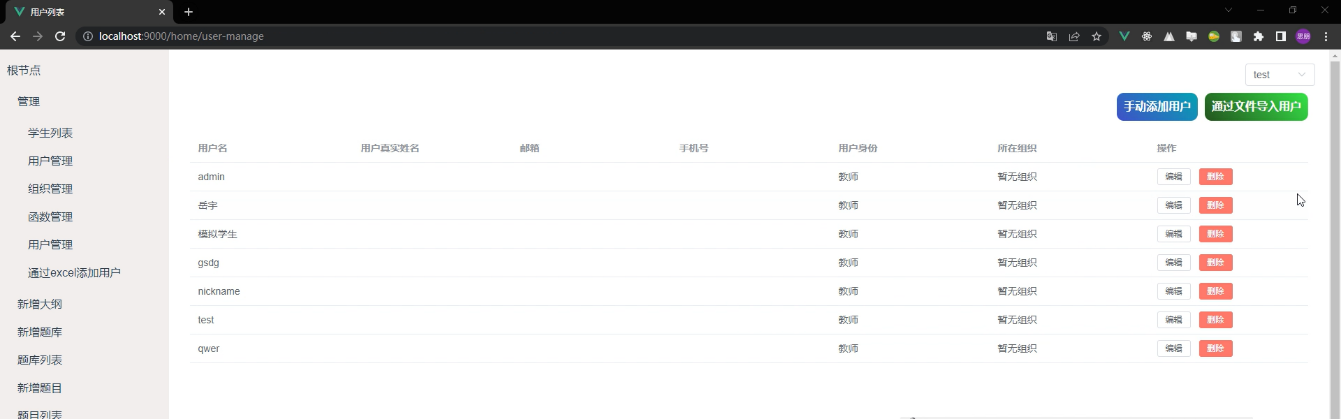
**图6.3 账号密码登录截图**

**表6.1 登录注册测试表**

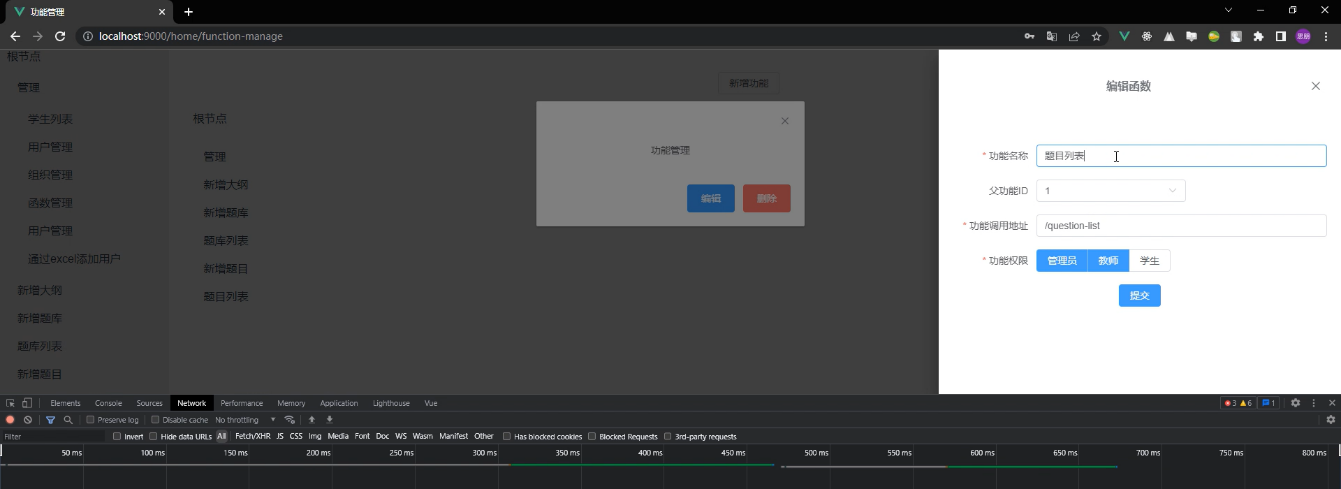
|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 错误注册表单提示 | 错误填写注册表单 | 通过 |
| 正确注册表单 | 正确填写注册表单 | 通过 |
| 用户列表 | 查看用户列表 | 通过 |
| 删除用户 | 删除用户 | 通过 |
| 通过用户名、密码登录 | 输入用户名、密码 | 通过 |
| 通过邮箱登录 | 输入邮箱及验证码 | 通过 |
| 未登录是否有全局拦截 | 未登录进入列表页 | 通过 |
| 已登录是否会携带token | 查看http请求头 | 通过 |
| 通过文件导入用过 | 通过文件导入用户 | 通过 |

## 函数管理、菜单相关测试

函数管理、菜单相关测试截图如图6.4，6.5所示，测试结果如表6.2所示。



**图6.4 菜单测试截图**



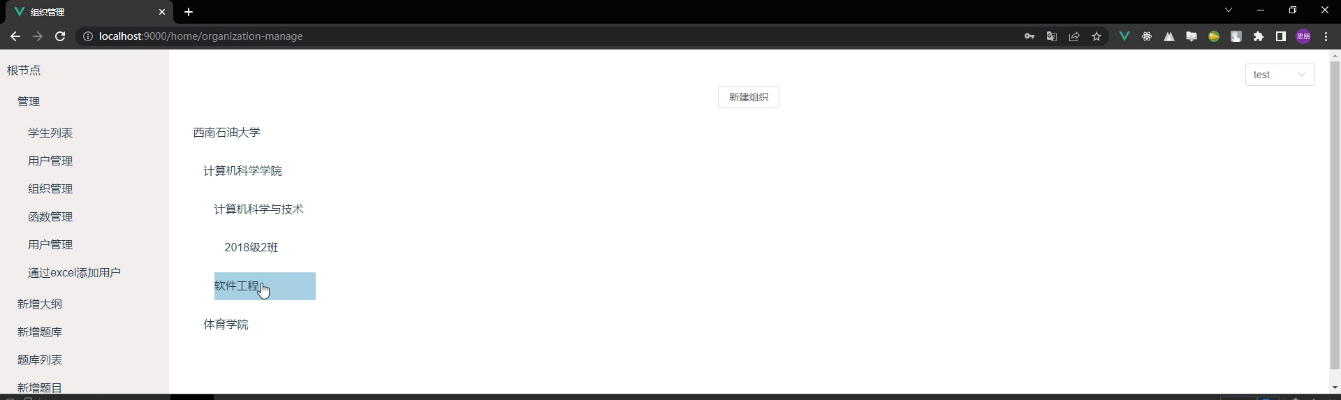
**图6.5 函数管理测试截图**

**表6.2 函数管理测试表**

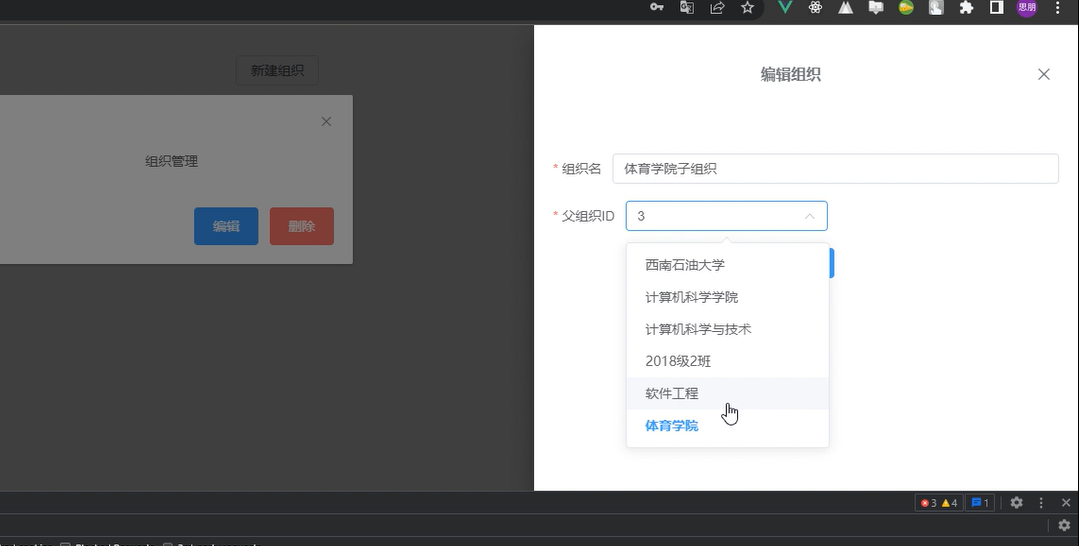
|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 新增函数 | 新增函数 | 通过 |
| 删除函数 | 删除函数 | 通过 |
| 编辑函数 | 编辑函数 | 通过 |
| 查看函数列表 | 查看函数列表 | 通过 |
| 模拟学生登录 | 模拟学生登录 | 通过 |
| 模拟老师登录 | 模拟老师登录 | 通过 |
| 模拟管理员登录 | 模拟管理员登录 | 通过 |

## 组织管理相关测试

组织管理测试截图如图6.6，6.7所示，测试结果如表6.3所示。



**图6.6 组织管理测试截图**



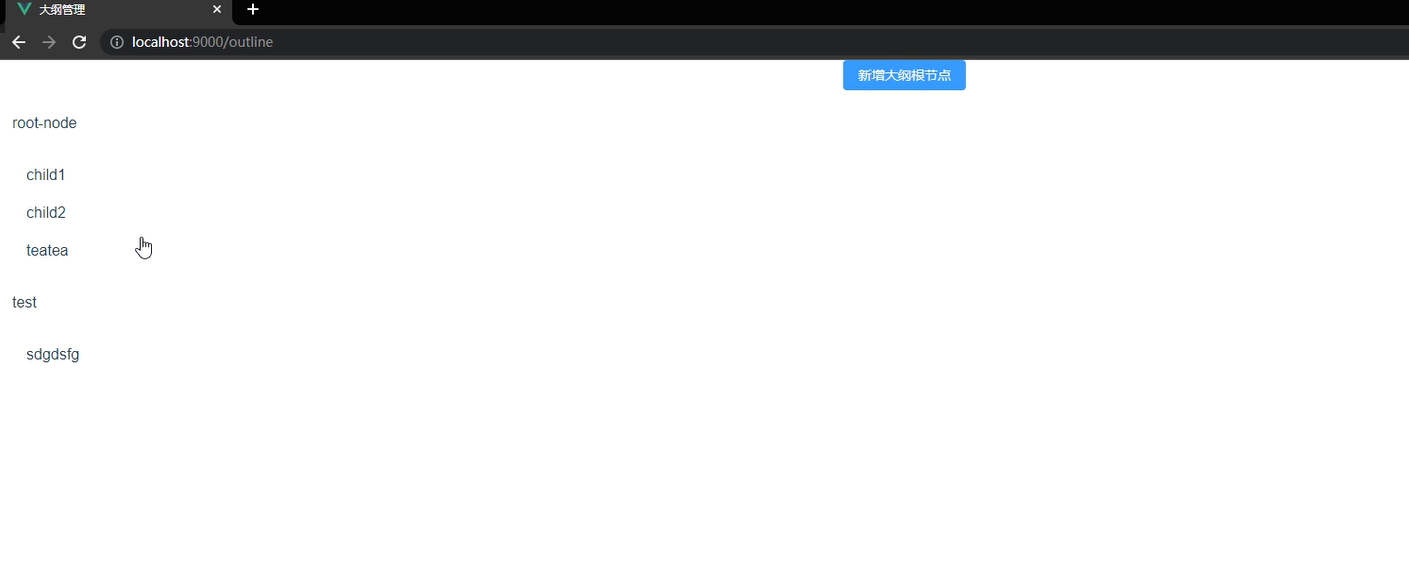
**图6.7 组织编辑测试截图**

**表6.3 组织管理测试表**

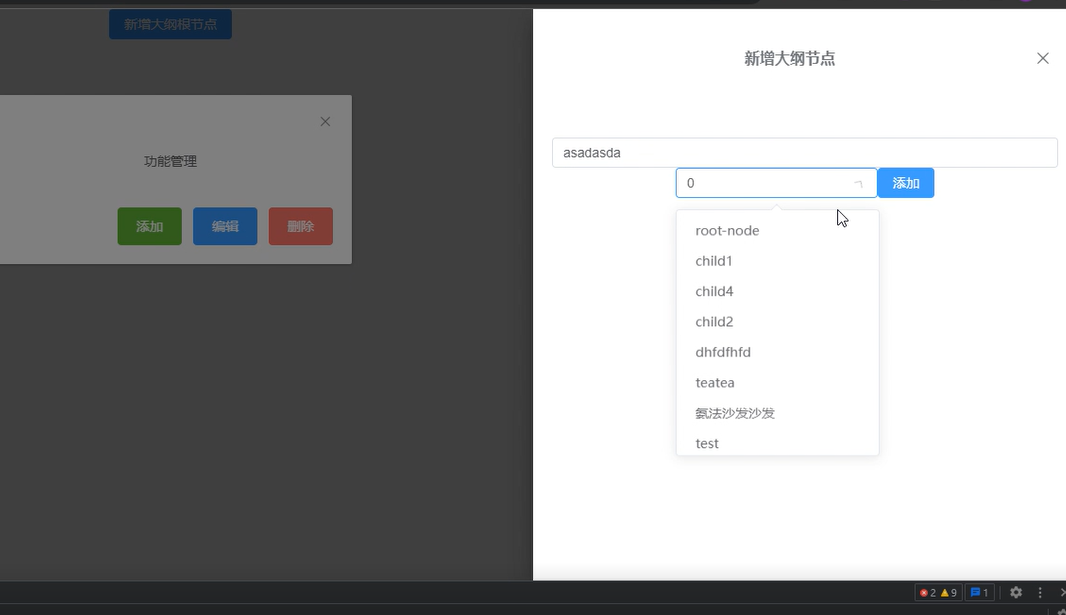
|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 新增组织 | 新增组织 | 通过 |
| 删除组织 | 删除组织 | 通过 |
| 编辑组织 | 编辑组织 | 通过 |
| 查看组织列表 | 查看组织列表 | 通过 |

## 大纲管理相关测试

大纲管理测试截图如图6.8，6.9所示，测试结果如表6.4所示



**图6.8 大纲管理截图**



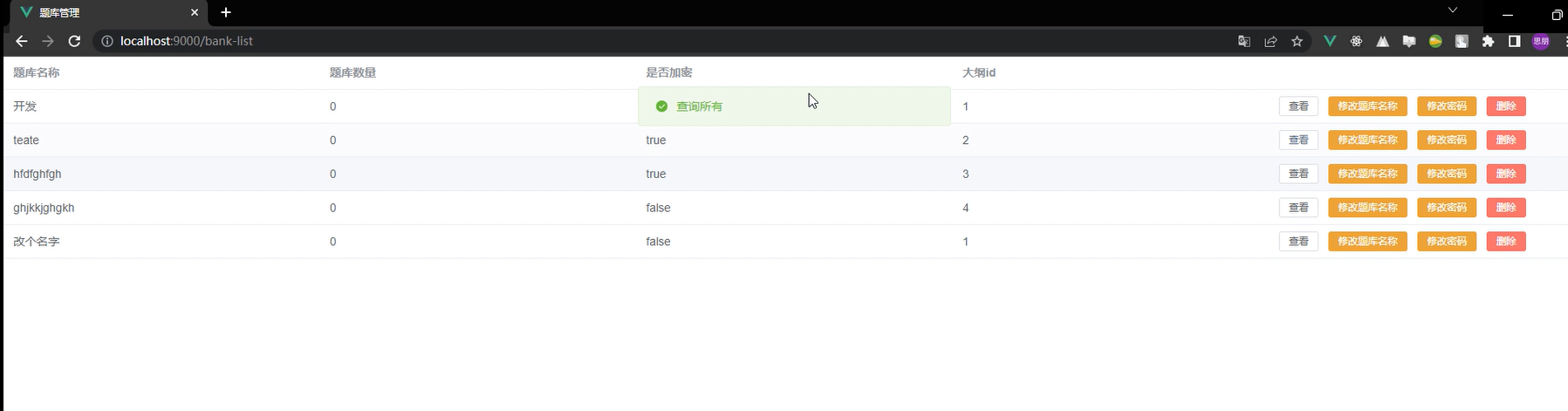
**图6.9 编辑大纲截图**

**表6.4 大纲功能测试表**

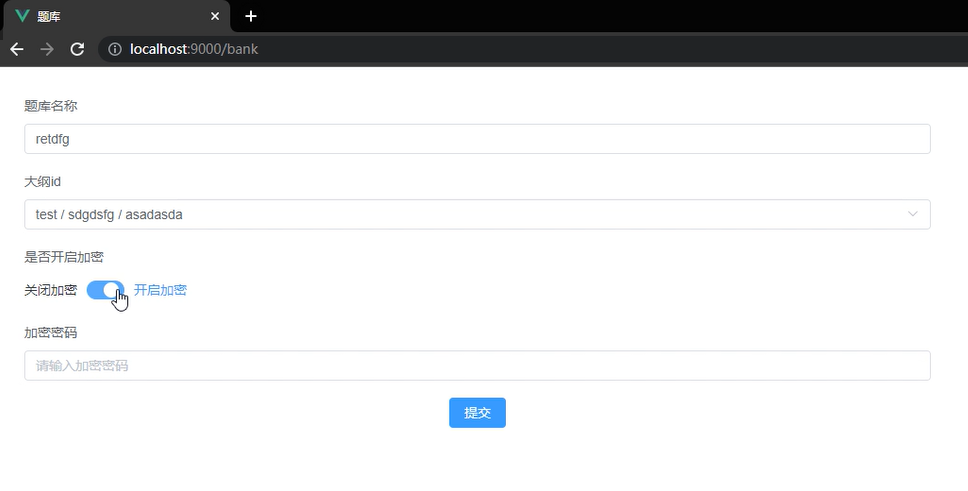
|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 新增大纲 | 新增大纲 | 通过 |
| 删除大纲 | 删除大纲 | 通过 |
| 编辑大纲 | 编辑大纲 | 通过 |
| 查看大纲列表 | 查看大纲列表 | 通过 |

## 题库管理相关测试

题库管理相关截图如图6.10，6.11所示，测试结果如表6.5所示



**图6.10 题库管理截图**



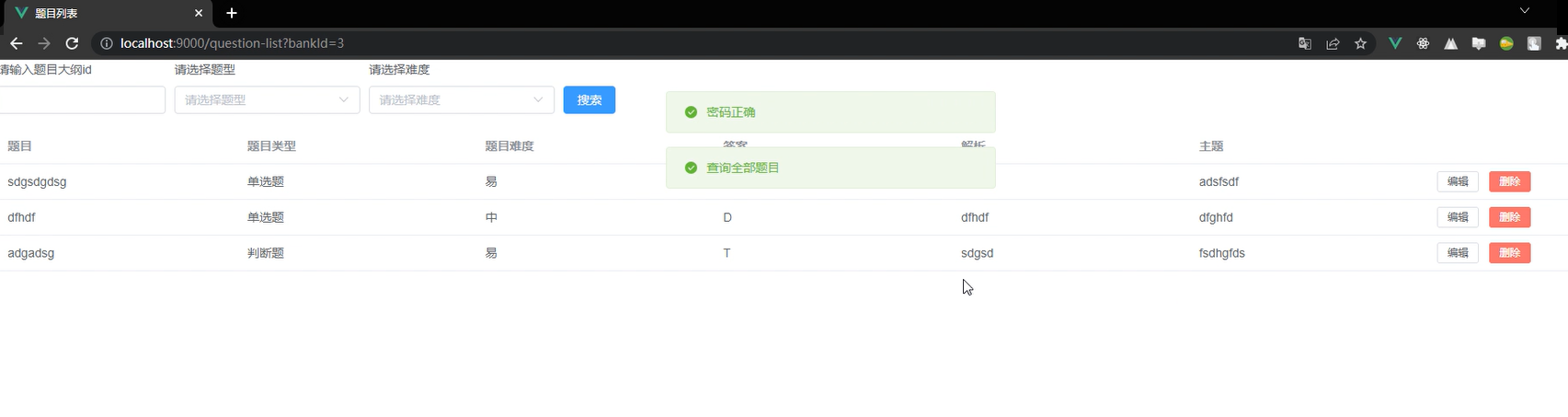
**图6.11 编辑题库截图**

**表6.5 题库测试结果表**

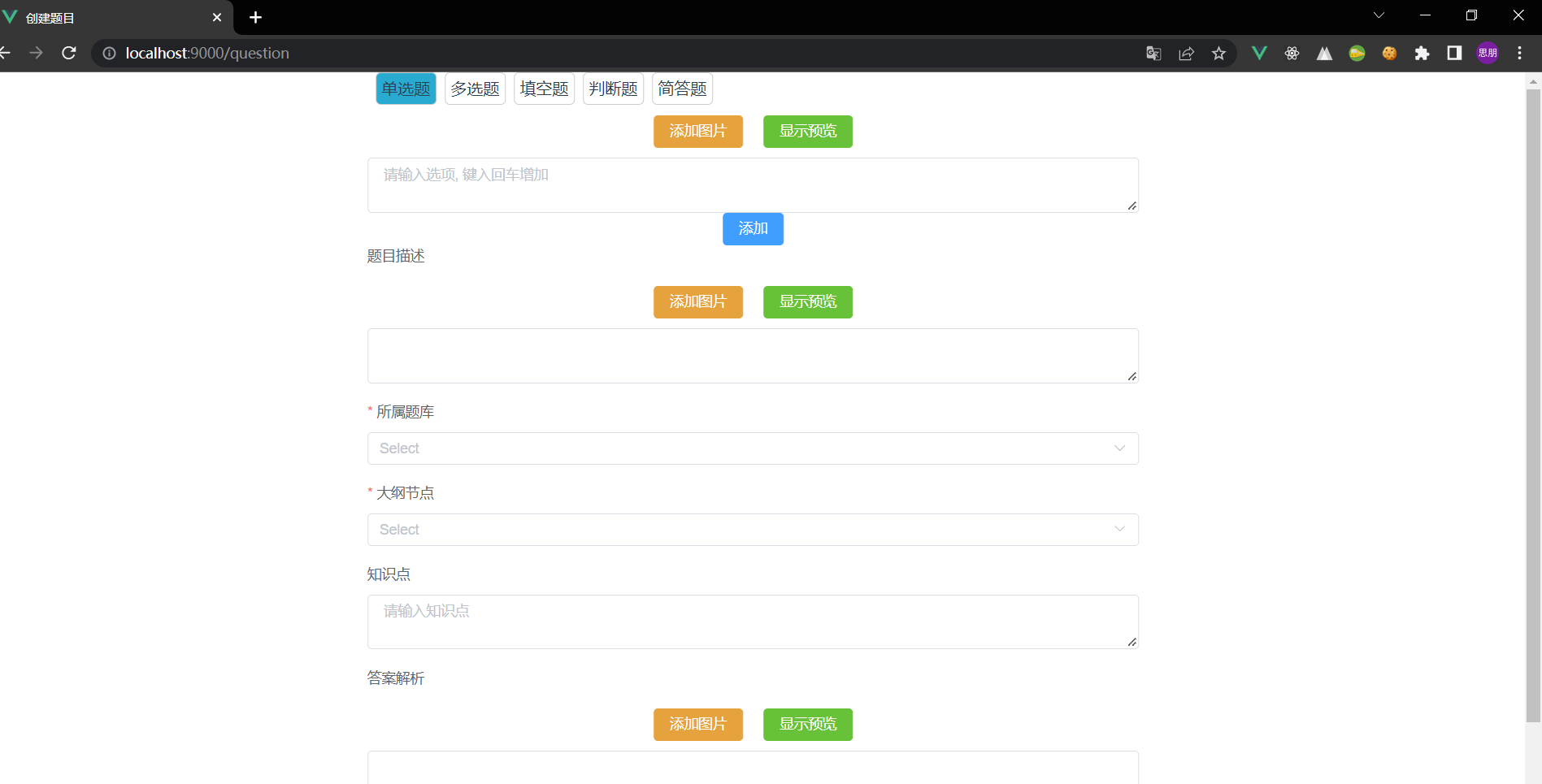
|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 新增题库 | 新增题库 | 通过 |
| 删除题库 | 删除题库 | 通过 |
| 编辑题库 | 编辑题库 | 通过 |
| 查看题库列表 | 查看题库列表 | 通过 |

## 题目管理相关测试

题目管理测试相关截图如图6.12，6.13所示，测试结果如表6.6所示。



**图6.12 题目列表**



**图6.13 编辑题目**

**表6.6 题目功能测试表**

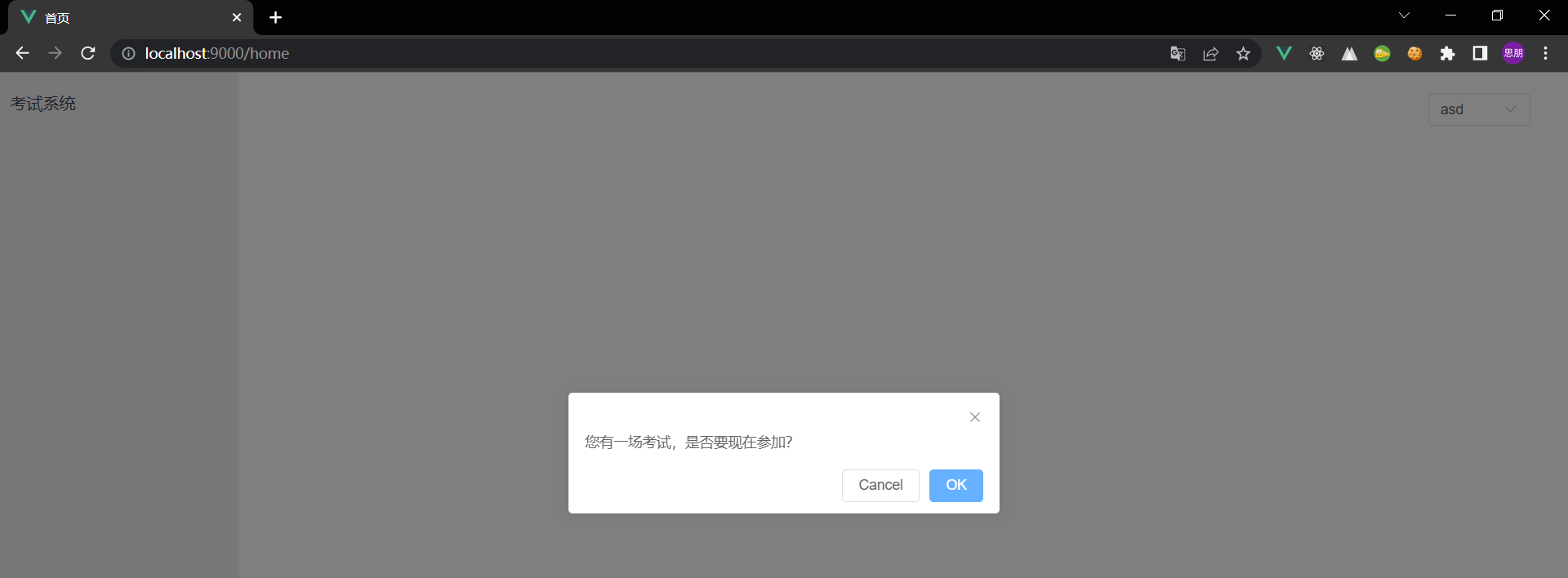
|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 新增、编辑单选题 | 新增、编辑单选题 | 通过 |
| 新增、编辑多选题 | 新增、编辑多选题 | 通过 |
| 新增、编辑判断题 | 新增、编辑判断题 | 通过 |
| 新增、编辑填空题 | 新增、编辑填空题 | 通过 |
| 新增、编辑简答题 | 新增、编辑简答题 | 通过 |
| 删除题目 | 删除题目 | 通过 |
| 上传图片 | 上传图片 | 通过 |

## 考试管理相关测试

考试管理测试截图如图6.14，6.15所示，测试结果如表6.7所示



**图6.14 参加考试**



**图6.15 考试通知**

**表6.7 考试测试结果**

|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 考试通知 | 考试通知 | 通过 |
| 参加考试 | 参加考试 | 通过 |
| 提交考试 | 提交考试 | 通过 |

## 阅卷相关测试

阅卷功能测试截图如图6.16所示，阅卷结果如表6.8所示



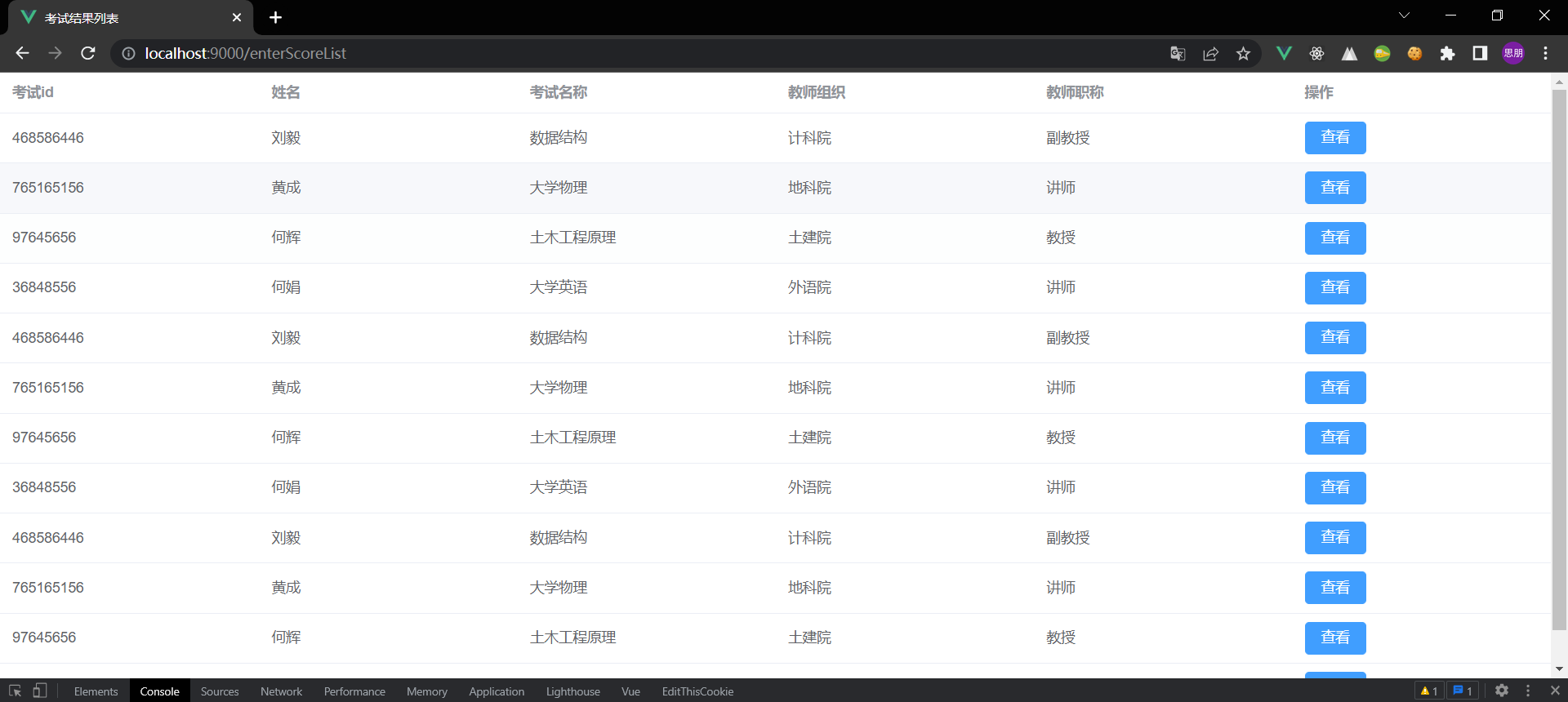
**图6.16 阅卷截图**

**表6.8 阅卷测试表**

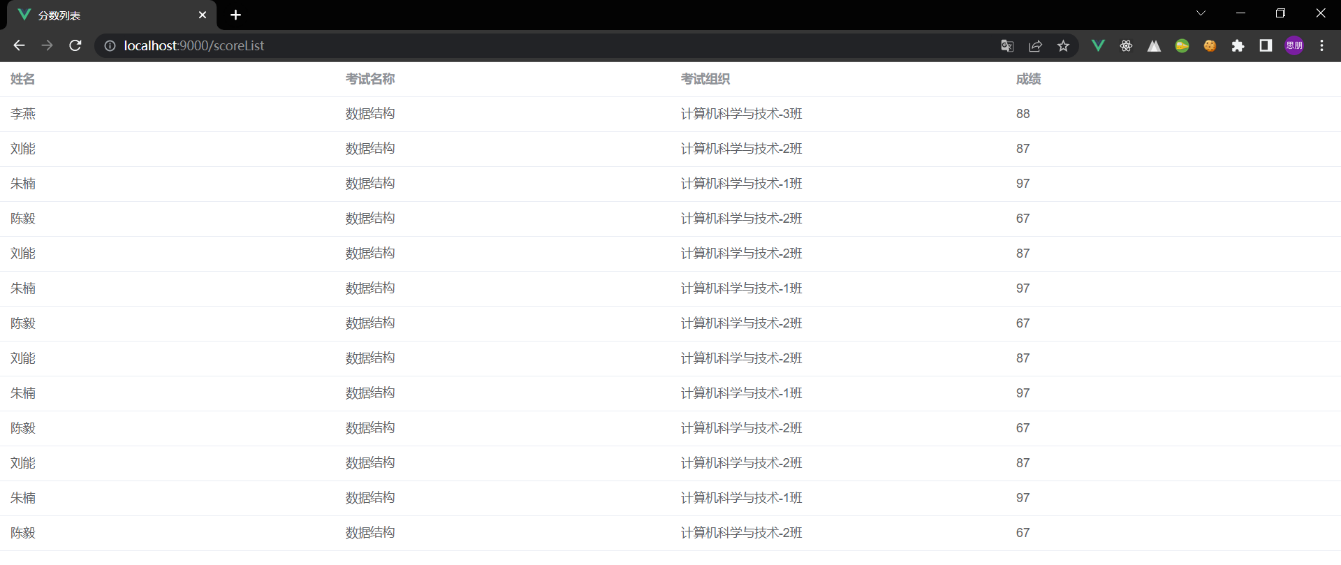
|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 客观题自动阅卷 | 客观题自动阅卷 | 通过 |
| 主观题手动阅卷 | 客观题自动阅卷 | 通过 |

## 成绩统计相关测试

成绩统计相关截图如图6.17，6.18所示，结果如表6.9所示。



**图6.17 试卷结果列表**



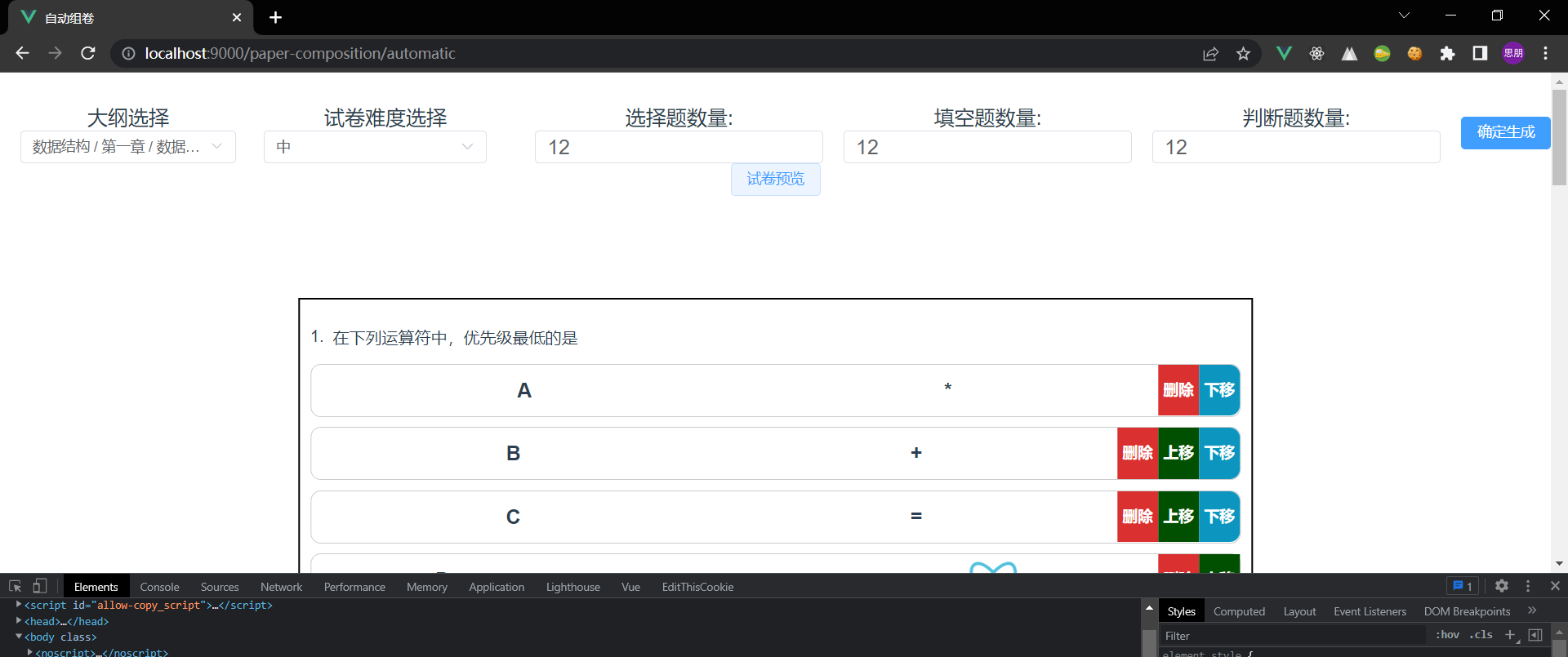
**图6.18 学生结果列表**

**表6.9 成绩统计相关测试**

|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 成绩统计列表 | 查看成绩统计列表 | 通过 |
| 学生结果列表 | 查看学生结果列表 | 通过 |

## 组卷相关测试

组卷相关测试如图6.19所示，测试结果如表6.10所示



**图6.19 组卷**

**表6.10 组卷测试表**

|  |  |  |
| --- | --- | --- |
| 测试功能 | 测试方法 | 测试结果 |
| 自动组卷 | 自动组卷 | 通过 |
| 手动组卷 | 手动组卷 | 通过 |

总结

本文参考各种教考分离考试系统，利用VUE+SprintBoot技术进行开发，综合各种软件开发的技术，实现了一个具有试题录入、用户注册登录、菜单管理、大纲建立、题库建立、自动组卷、考试发放、试卷批阅功能的教考分离的考试系统。本课题主要完成的工作如下：

1. 背景调研。通过学校图书馆平台、百度学术、Google Scholar查询国内外相关论文、研究，确定了开发的目的和意义。
2. 技术调研。对比Broswer、Server端的开发工具，最新技术的优缺点，技术学习成本以及实现的复杂度，确定了开发的技术路线。
3. 需求分析。分析要实现的功能点，根据功能之间的联系设计E-R图，再对数据库进行设计。搭建整体的框架。
4. 功能实现。本系统采用前后端分离的开发方式，便于管理和维护。分别使用了MvvM、MVC框架VUE和SpringBoot。数据库采用Mysql数据库。借鉴经典的设计模式，不断对代码进行简洁优化，保证代码易于维护。
5. 系统的测试。在开发的过程中每一个用例都会去分配单元测试，开发完成后，使用Istanbul对整个系统进行全方位的测试，保证了系统的安全、稳定。

该系统仅仅是一个简易版的考试系统，相比于市面上的系统，还有许多未完善的功能，如在线监考功能。了解到在线监考类似于在线聊天，采用轮询的方式向后台不断地发送当前地界面数据，或者使用websocket与后台保持长连接进行数据传输，之后会对这一功能进行实现并不断完善系统。

致谢

论文至此，我的毕业设计也画上一个句号，大学四年也即将结束，回顾这四年，自己不算风雨兼程，但也算勤勤恳恳吧。在四年的时光里，尝试了许多东西，经历了挫败和成长。这其中，要特别感谢我的父母，无论是在学业还是生活、情感上，都是我最坚强的后盾。同时我的女朋友左玲，陪伴我走过大学最美好的时刻，谢谢你，为我的人生画上精彩的一笔。其次要感谢我的室友们，吴昊阳、赵哲一、蒲涛在生活、学习上互帮互助，谭磊、姚志昆在找工作的时候携手并进，共同努力，没有你们就没有今天的我，谢谢你们。

最后，要特别感谢在毕业设计以及论文上对我悉心指导的谯英教授，从论文的选题到确定思路，从资料的搜集、提纲的拟定到内容的写作与修改，都离不开您对我的帮助，感谢您！

参考文献

1. [Xuefei Wang](https://ss.zhizhen.com/s?sw=author%28Xuefei+Wang%29" \t "_blank)， [Ping Wang](https://ss.zhizhen.com/s?sw=author%28Ping+Wang%29). A Tripartite Analysis of the Separation of Teaching and Examination and the Integration of Teaching and Examination[C]. 2019 9th International Conference on Management，Education and Information(MEICI 2019).
2. 张东旭，滕玉石.高校“教考分离”工作的探讨.[J].辽宁工学院学报：社会科学版， 2005(2):115-118.
3. 黄镜流.充分发挥考试的功能[J].职教论坛，1997(2):29.
4. 丁延平，王益平.教考分离考试制度的利弊思考.[J].南京中医药大学学报.2002(3):1.
5. 蒋秋艳，张东.对教考分离模式的再思考.[J].南通工学院学报， 2003(2):61-62.
6. 王鹏.对教考分离考试制度的认识与思考.[J].辽宁工业大学学报（社会科学版）， 2008，10(4):92-94.

1. [Zhiling Wu](https://ss.zhizhen.com/s?sw=author%28Zhiling+Wu%29" \t "_blank)， [Li Wei](https://ss.zhizhen.com/s?sw=author%28+Li+Wei%29). An Exploratory Reform in Separation of Teaching and Examination for Postgraduate Academic English Course Based on POA[J]. Open Journal of Social Sciences，2022：（3）.
2. Tan， De-rong;Sun， Liang.Method and Practice of Separation of Teaching and Examination on University Talents Training.[C].EDUCATION AND EDUCATION MANAGEMENT.

1. [Yan Liu](https://ss.zhizhen.com/s?sw=author%28Yan+Liu%29" \t "_blank)， [Yunqing Zhang](https://ss.zhizhen.com/s?sw=author%28Yunqing+Zhang%29). Exploration and Practice of Separation of Teaching and Examination in Higher Vocational Education[C]. 2019 7th ICASS International Conference on Management，Business and Technology Education(ICMBT 2019).
2. Lijun Pei. Research on the Application of the Improved Assessment Model Based on the Separation of Teaching and Examination in C Programming.[C]. 2019 14th International Conference on Computer Science & Education (ICCSE).
3. 姜雨薇， 马碧书， 张芸娇.“教考分离”初探[J].科学咨询，2021(10):22-23.
4. 吕道远.基于教考分离的考试制度改革与实践探索—以W学院艺术史论课程为例.[J].才智，2016(2):101-102.
5. 葛友华，吴乃领.教考分离模式的研究[J]盐城工学院学报（社会科学版），2003(3): 67-68.
6. Yonghe DENG.Improve Quality of Teaching and Studying Based on Separation of Examination and Teaching.[C].
7. 王萱， 刘时乔， 刘宇， 赵欣.基于教考分离的医学高校题库建设探索与研究——以《临床中药学》为例[J].时珍国医国药，2017，2(31):445-47。
8. 魏首柳， 柯小玲.基于教考分离的课程试题库建设的探讨与实践[J].高教学刊，2021(3):100-103.
9. 前端开发技术介绍.Mozilla官网.https://developer.mozilla.org/zh-CN/docs/Learn.
10. Vue.js官方文档.https://cn.vuejs.org.
11. SpringBoot官方文档.https://spring.io/projects/spring-boot/
12. MySQL官方文档.https://www.mysql.com.