

本科毕业设计（论文）



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **题目：** | | |  | | --- | | **基于Vue+SpringBoot 教考分离系统** | | |
|  | | **的设计与实现** | |

|  |  |  |  |
| --- | --- | --- | --- |
| **学生姓名** | **汪思朋** | **学　号** | 201831082129 |
| **教学院系** | **计算机科学学院** | | |
| **专业年级** | **计算机科学与技术2018级** | | |
| **指导教师** | **谯英** | **职　称** | **副教授** |
| **单　　位** | **计算机科学学院** | | |
| **完成日期** | **2022-05-20** | | |

Southwest Petroleum University

Graduation Thesis



Design and implementation teaching test separation system based on Vue and SpringBoot

**Grade: 2018**

**Name: Wang Sipeng**

**Speciality: Computer Science and Technology**

**Instructor: Qiao Ying**

**School of Computer Science**

**2021-5**

**摘要**

所谓教考分离,就是把教学和考试分开，根据培养目标、教学目的、教学大纲,制定考核大纲，建立一套包括试题库、命题、阅卷、考试、成绩管理等各种规章制度完备的考核管理系统。相对于教考合一而言，教考分离有利于教学考核的公平、公正，有利于教学评价的科学、有效,有利于教风、学风的根本好转，从而形成重教重学的良好氛围,不断提高教学质量。

本文从选题背景出发，综合分析了教考分离在国内外的实践情况以及实践成果，然后，在对超星、慕课网、中国大学MOOC等学习平台进行参考分析之后，确定了页面的大体样式以及交互逻辑。同时参考这些系统的系统架构、设计，再结合自己的技术栈，确定了本系统的开发结构与路线。

接下来，本文对教考分离系统的各个模块做出详细的介绍，包括流程、用例、功能模块的使用。分析了在开发的过程中，遇到了哪些困难，自己是怎么去思考、解决这些困难的，以及采取什么样的设计优化此系统。

然后对整个系统的使用流程以及功能模块进行了详细的测试，保证系统的可靠性，然后对测试结果进行分析和讨论。

**关键词: 教考分离；考试系统；Vue；Spring Boot；B/S系统设计；**

Abstract

The so-called separation of teaching and examination is to separate teaching and examination, according to training objectives, teaching objectives, teaching syllabus, formulate assessment outline, establish a set of including test question bank, proposition, reading papers, examination, achievement management and other rules and regulations complete assessment management system. Compared with the integration of teaching and examination, the separation of teaching and examination is conducive to the fairness and justice of teaching assessment, scientific and effective teaching evaluation, and the fundamental improvement of teaching style and learning style, thus forming a good atmosphere of stressing teaching and learning, and constantly improving the quality of teaching.

The design of the system and the integration of undergraduate computer science and technology of all kinds of knowledge, the integration of theoretical knowledge and popular technology, but also the integration of theoretical knowledge and business applications and other technical applications. Can reflect the professional design, through this design, to understand the application of professional in the industry and the system to build business process has great significance and role.

First, this paper analyzes the status quo of the system at home and abroad and the significance of the topic. According to the analysis, the separation of teaching and examination is mainly practiced in the field of higher education at home and abroad and has certain positive results, so this topic is based on the evidence, there are scenes to be implemented.

Then, after reference analysis of superstar, MOOC, Chinese university MOOC and other learning platforms, the general style and interaction logic of the page are determined. At the same time reference these systems system architecture, design, combined with their own technology stack, determine the development structure and route of the system.

Next, this paper makes a detailed introduction to each module of the separation system, including the process, use cases, the use of functional modules. Analyzed in the process of development, encountered what difficulties, how to think, to solve these difficulties, and what kind of design to optimize the system.

Finally, the use process and functional modules of the whole system were tested in detail to ensure the reliability of the system, and then the test results were analyzed and discussed.

**Key words: separation of teaching and examination; Examination system; Vue. Spring the Boot; B/S system design；**

目录

[1 绪论 1](#_Toc103695129)

[1.1 选题的目的和研究意义 1](#_Toc103695130)

[1.2 相关现状研究 1](#_Toc103695131)

[1.2.1 国外现状研究 1](#_Toc103695132)

[1.2.2 国内现状研究 1](#_Toc103695133)

[1.3 本文工作 1](#_Toc103695134)

[1.4 本文的章节安排 1](#_Toc103695135)

[2 技术选型的分析与研究 2](#_Toc103695136)

[2.1 架构选型分析 2](#_Toc103695137)

[2.1.1 C/S架构介绍 2](#_Toc103695138)

[2.1.2 B/S架构介绍 2](#_Toc103695139)

[2.1.3 为什么采用B/S架构 2](#_Toc103695140)

[2.2 前端技术选型分析 3](#_Toc103695141)

[2.2.1 HTML、CSS、JavaScript 介绍[17] 3](#_Toc103695142)

[2.2.2 Vue 介绍 4](#_Toc103695143)

[2.3 后端技术选型分析 4](#_Toc103695144)

[2.3.1 Java 介绍 4](#_Toc103695145)

[2.3.2 Java Web介绍 5](#_Toc103695146)

[2.3.3 Spring Boot 介绍 6](#_Toc103695147)

[2.3.4 MySql 介绍 6](#_Toc103695148)

[2.4 本章总结 6](#_Toc103695149)

[3 总体设计 7](#_Toc103695150)

[3.1 需求分析 7](#_Toc103695151)

[3.2 数据库设计 8](#_Toc103695152)

[3.2.1 用户表 8](#_Toc103695153)

[3.2.2 菜单表 9](#_Toc103695154)

[3.2.3 组织表 9](#_Toc103695155)

[3.2.4 题库表 10](#_Toc103695156)

[3.2.5 大纲表 10](#_Toc103695157)

[3.2.6 问题表 10](#_Toc103695158)

[3.2.7 试卷表 11](#_Toc103695159)

[3.2.8 考生表 11](#_Toc103695160)

[3.2.9 考场表 12](#_Toc103695161)

[3.2.10 答案表 12](#_Toc103695162)

[3.3 各功能模块流程分析 13](#_Toc103695163)

[3.3.1 用户注册模块 13](#_Toc103695164)

[3.3.2 用户登录模块 13](#_Toc103695165)

[3.3.3 系统管理功能 14](#_Toc103695166)

[3.3.4 题目相关功能 14](#_Toc103695167)

[3.3.5 试卷的组织功能 15](#_Toc103695168)

[3.3.6 考试功能 16](#_Toc103695169)

[3.3.7 判卷及统计功能 16](#_Toc103695170)

[4 系统详细设计 17](#_Toc103695171)

[4.1 登录模块的实现分析 17](#_Toc103695172)

[4.1.1 邮箱验证登录是如何实现的 17](#_Toc103695173)

[4.1.2 JWT验证的实现 18](#_Toc103695174)

[4.2 不同权限显示不同菜单功能的分析 20](#_Toc103695175)

[4.3 通过excel添加用户的实现 21](#_Toc103695176)

[4.4 题目管理的实现分析 22](#_Toc103695177)

[4.5 自动组卷的实现分析 24](#_Toc103695178)

[4.6 考试功能的实现分析 28](#_Toc103695179)

[4.6.1 如何去收集试题的答案数据 29](#_Toc103695180)

[4.6.2 如何处理多人同时提交 29](#_Toc103695181)

[4.7 阅卷功能的实现分析 30](#_Toc103695182)

[4.8 系统的优化设计 30](#_Toc103695183)

[4.8.1 提交场景的防抖实现 30](#_Toc103695184)

[4.8.2 前端缓存http请求的实现 31](#_Toc103695185)

[5 系统的测试 31](#_Toc103695186)

[5.1 注册登录相关测试 31](#_Toc103695187)

[5.2 函数管理、菜单相关测试 33](#_Toc103695188)

[5.3 组织管理相关测试 34](#_Toc103695189)

[5.4 大纲管理相关测试 35](#_Toc103695190)

[5.5 题库管理相关测试 36](#_Toc103695191)

[5.6 题目管理相关测试 37](#_Toc103695192)

[5.7 考试管理相关测试 38](#_Toc103695193)

[5.8 阅卷相关测试 39](#_Toc103695194)

[5.9 成绩统计相关测试 40](#_Toc103695195)

[5.10 组卷相关测试 41](#_Toc103695196)

[总结 42](#_Toc103695197)

[致谢 43](#_Toc103695198)

[参考文献 44](#_Toc103695199)

# 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[13]，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[9]。教考分离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[8]，所以“教考分离”是适应当代教育发展的理想的考试模式[10]。

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[7]。

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。

这个系统的设计和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

### 国外现状研究

在Google Scholar以” separate of teach and test”为关键字搜到国外以教考分离的结果有520000条，使用西南石油大学图书馆平台搜索到的外文结果为9484条。从论文、期刊等paper的出现时间来看，国外很早就已经开始着手于教考分离相关的研究了，且研究的领域主要在高等教学领域。Donghong Peng分析道：教考分离有助于培养学生的自学能力。同时，题库和组题的试题数量要不断完善，这是保证学生学习能力客观反映的前提。[11]同时De-rong指明，其教授的交通运输专业表演课教学与考试分离的实践证明是可行的，它有助于提高本科生的教学质量[16]。高等教育实行教考分离，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[12]。

### 国内现状研究

我国是考试的发祥地之一，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[14]，现在"教考分离"的考试制度正在逐步推进[1] 。

教考分离在我国有着众多的实践以及成果，上海农林职业技术学院的陈晶等人则结合上海农林职业技术学院双证融通试点工作的经验，将教考分离的教学评价方式进行了推广，为相关院校课程教学评价改革提供参考，共同实现教学质量提升。[4] 盐城工学院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。[6] 丽水学院实行教考分离，提高了教学质量。[15]

赵欣认为建设高质量题库是实现教考分离和提高教学质量的重要途径,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。[2]

柯小玲说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,是目前和今后高校教学改革的一个重要方向。[5]

笔者在中国知网对国内相关文献资料进行检索时分别以 “教考分离” “教学质量”为关键词进行高级检索，截至2022年4 月12 日，获取到的相关文献信息见图1.1。



图1.1‘教考分离’相关文献统计

可以看出教学质量始终是教育界研究的热点课题，从教考分离的角度看，目前国内外关于教考分离研究的文献非常少，且多是高等教育领域。

## 本文工作

本文说明了基于Vue + SpringBoot 等技术实现的教考分离系统的原理。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、教考分离的国内外现状进行了相关研究、分析。明确了当前大部分的学校的教考制度都是教考分离。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： 本文的绪论，主要介绍了教考合一在中国的发展历史、以及其中的优缺点，继而演变出教考分离。然后分析了教考分离系统的产生背景，其国内外现状以及此系统的好处。还有就是此系统的大致开发工作。

第二章：教考分离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：教考分离系统的总体设计。按照实际的工程化思路，构建完整的应用流程。

第四章：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。

第五章：教考分离系统的测试。

# 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。C/S架构是通过提供查询响应而不是总文件传输来减少了网络流量。它允许多用户通过GUI前端更新到共享数据库，在客户端和服务器之间通信一般采用远程调用（RPC）或标准查询语言（SQL）语句。

### 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)架构模式，是随着Internet技术的兴起，对[C/S架构](https://baike.baidu.com/item/C%2FS%E6%9E%B6%E6%9E%84/1050735)的一种变化或者改进的架构。在这种架构下，用户工作界面是通过WWW浏览器来实现，极少部分[事务](https://baike.baidu.com/item/%E4%BA%8B%E5%8A%A1)逻辑在前端（Browser）实现，但是主要事务逻辑在服务器端([Server](https://baike.baidu.com/item/Server))实现，形成所谓三层3-tier结构。B/S架构是WEB兴起后的一种网络架构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser），如[Netscape Navigator](https://baike.baidu.com/item/Netscape%20Navigator)或[Internet Explorer](https://baike.baidu.com/item/Internet%20Explorer)，服务器安装[Oracle](https://baike.baidu.com/item/Oracle)、[Sybase](https://baike.baidu.com/item/Sybase)、[Informix](https://baike.baidu.com/item/Informix)或 [SQL Server](https://baike.baidu.com/item/SQL%20Server)等数据库。浏览器通过[Web Server](https://baike.baidu.com/item/Web%20Server)同数据库进行数据交互。 这样就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 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务。B/S 建立在广域网之上的, 不必是专门的网络硬件环境,例如电话上网, 租用设备，信息管理，有比C/S更强的适应范围, 一般只要有操作系统和浏览器就行。

B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上. 比C/S有更高的要求 B/S结构的程序架构是发展的趋势, 从MS的.Net系列的BizTalk 2000 Exchange 2000等, 全面支持网络的构件搭建的系统. SUN 和IBM推的JavaBean 构件技术等,使 B/S更加成熟.

C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统

B/S 构件组成,方面构件个别的更换,实现系统的无缝升级. 系统维护开销减到最小.用户从网上自己下载安装就可以实现升级.

C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统

B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是C/S无法作到的. 与操作系统平台关系最小.

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[17]

1. HTML

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。HTML 可以看作一个程序版的 Word，我们在Word中进行文字的编写、图片的排版以及一些超链接的引入等，都是使用office直接提供的GUI进行操作，而HTML 就是命令式操作。

比如向Word中插入一个图片，需要从本地文件中复制或者引入，批量插入的时候就比较麻烦，而HTML用一个 <img src=’’> 标签就可以完成这些事情，非常的方便。

2. CSS

上文说到HTML 可以像word那样插入各种各样的内容，有的时候 word还需要排版、整理文字或者其他内容的样式，在word中可以通过开始菜单中的控制面板进行控制，而在HTML 中，CSS就是起了一个这样的作用，比如我给HTML某段内容设置文字颜色为红色，那么可以用CSS { color: red }, 进行控制。

3. JavaScript

上述的HTML和CSS已经能够覆盖 word 的大部分使用场景了，为什么还要使用 JavaScript呢？

其实大部分时候，我们在使用word的时候，都是使用的一个静态的页面，也就是只有只读功能，但是我们的系统还需要进行一个复杂的操作，是HTML 和 CSS 不能实现的，这时候就需要JavaScript来实现了。

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

上文说到 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。比如我想要获取一个输入框中的内容，我需要先获取这个输入框的Dom，然后取得其中的值，一个注册页面，有很多的输入框，挨个取获取内容太麻烦了。再比如我要去渲染一个列表，我首先要通过网络请求拿到后端的数据，然后将数据与HTML进行拼接，如果是一个比较复杂的列表，拼接的过程十分麻烦，而且极易出错，不方便修改也不易维护。

这个时候，就需要去寻找一个JavaScript框架来解决上述问题，那么我选用的是Vue框架。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)可以理解为MVC框架中的 controller, view就是HTML模板，model就是JavaScript中需要渲染到HTML上的变量, VM相当于一个中间人，帮助我们把 view和model绑定起来，将model的数据渲染到view上，同时也让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，不仅吸收了C++ 语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念。因此Java语言具有功能强大和简单易用两个特征。因为在实习工作的时候，接触到的后端同学都是以写Java为主，对Java也就比较熟悉了，同时，Java生态比较成熟，有众多的工具包和解决方案，在我遇到问题的时候，能够快速去社区通过问问题的方式解决。

### Java Web介绍

1. JavaEE

JavaEE 全称 Java Platform, Enterprise Edition，它是对 JavaSE(Java Platform, Standard Edition) 的扩展，加入了面向企业开发（实际上就是网络和 Web 有关开发）的支持，包括 Servlet，WebSocket，EL，EJB 等。简单理解，JavaEE 就是 JavaSE + 更多的 jar 包，这些 jar 包命名以 javax 开头，例如 javax.servlet, javax.websocket 等。

2. Servlet 和 Servlet Container

在 JavaEE 的诸多组件中，做 Web 开发一定躲不开的是 Servlet。Servlet 是一套用于处理 HTTP 请求的 API 标准。我们可以基于 Servlet 实现 HTTP 请求的处理。但是 JavaEE 当中只提供了 Servlet 的标准，要真正运行 Servlet，需要使用 Servlet Container。

如果 Servlet 是电器，Servlet Container 就是电源插座。这层抽象让 Servlet 可以跑在任何一个 Container 当中，隔绝了对 Runtime 环境的依赖。

JavaEE 本身没有提供 Servlet Container，比较常用支持 Servlet Container 的 Server 软件有 Apache Tomcat，Glassfish，JBoss，Jetty 等等。

JavaEE 开发完全可以不使用任何第三方框架，仅仅通过 Servlet 和 JSP 技术，在 JSP 之上构建 View，数据库，业务逻辑等，辅以 EL 表达式等技术简化开发，

3. Java Bean

Java Bean 是一个Java的规范,规定了

所有属性为private

(2)提供默认构造方法

(3)提供getter和setter

(4)实现serializable接口

4. Spring IoC

控制反转 (下文统一简称为 IoC) 把传统模式中需要自己通过 new 实例化构造函数，或者通过工厂模式实例化的任务交给容器。通俗的来理解，就是本来当需要某个类（构造函数）的某个方法时，自己需要主动实例化变为被动，不需要再考虑如何实例化其他依赖的类，只需要依赖注入 (Dependency Injection, 下文统一简称为 DI), DI 是 IoC 的一种实现方式。所谓依赖注入就是由 IoC 容器在运行期间，动态地将某种依赖关系注入到对象之中。所以 IoC 和 DI 是从不同的角度的描述的同一件事情，就是通过引入 IoC 容器，利用依赖注入的方式，实现对象之间的解耦。

### Spring Boot 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 还会帮我们做许多配置工作。通过 @SpringBootApplication 这个 Annotation，它会帮我们打开 @EnableWebMvc，以及 @ComponentScan 功能，这意味着 SpringBoot 会和 SpringMvc 找到同一个 package 下的 @Controller，@Configuration 等类，进行自动配置。

简单来说，Spring Boot 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。

整个系统的后台部分，采用传统的Java Web 技术路线进行开发，Java、MySQL、springboot、maven等等都需要我去从零开始学习，其实自己在后端方面也有着广泛的涉猎，用Node写过后台，大概的工程化的东西与Java Web相差不大，所需要学习的就是语言范式、工程结构、代码组织等配置类的东西。

前端部分，因为我有着一段相关的实习经历，所以实现起来不会太难。

# 总体设计

## 需求分析

教考分离系统最核心的功能莫过于试卷的组织以及考试。因为教考分离与教考合一的一个重要的区别就是，组织试卷的老师是否为教授此门课程的老师，因此在此系统中设计了一个题库来收集不同教师出的试题以及分发不同教师的试题。大概的流程就是，用户登录之后，系统会根据用户的身份来显示不同的菜单，学生的主要功能有接收考试通知，参加考试和成绩查看功能。教师的主要功能就是建立题库、组织试卷、批阅试卷和统计考试结果。管理员主要负责系统管理和用户管理模块，包括组织机构的管理，功能的开放管理，角色权限管理，用户注册等。具体功能如图3.1所示。

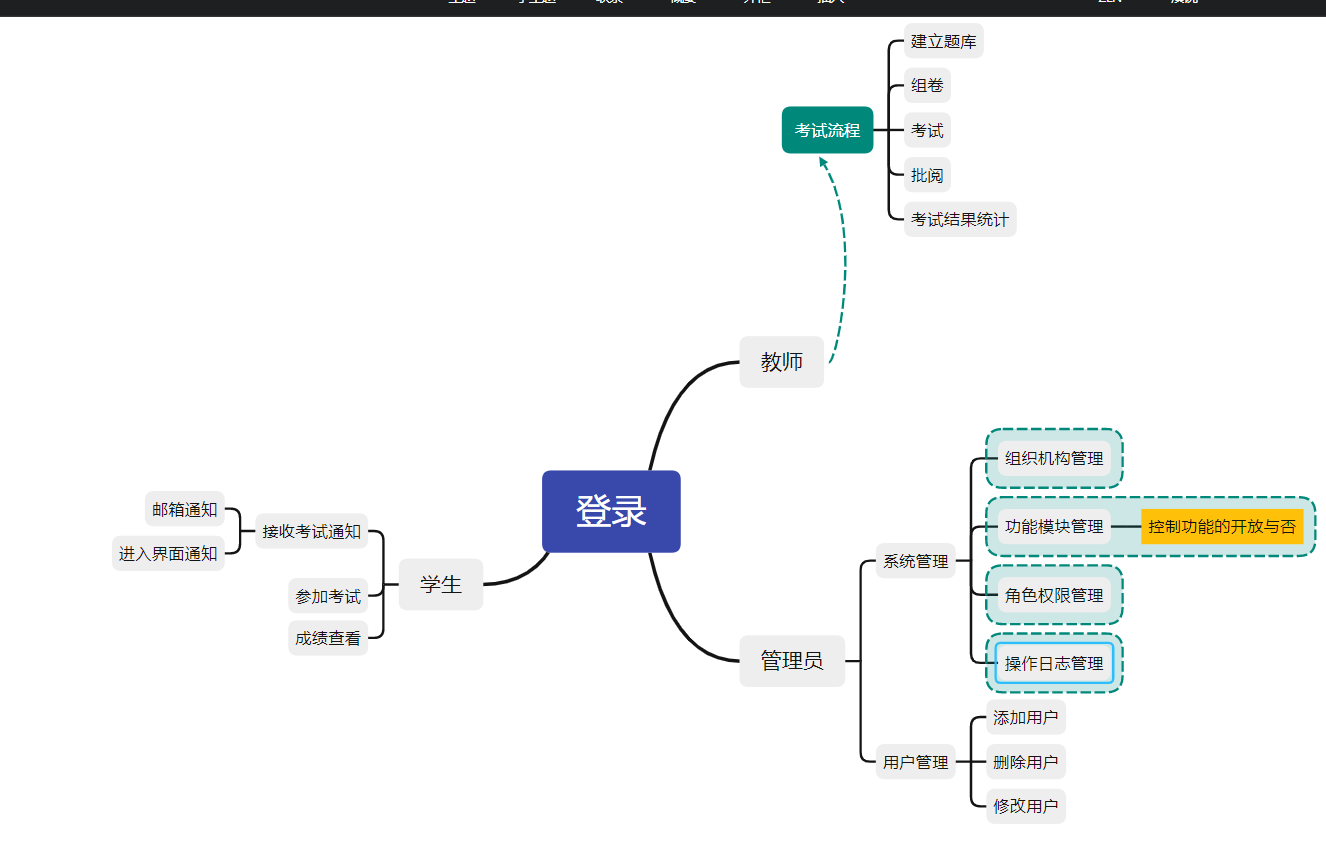


图3.1 教考分离功能图

## 数据库设计

总图设计如图3.2所示

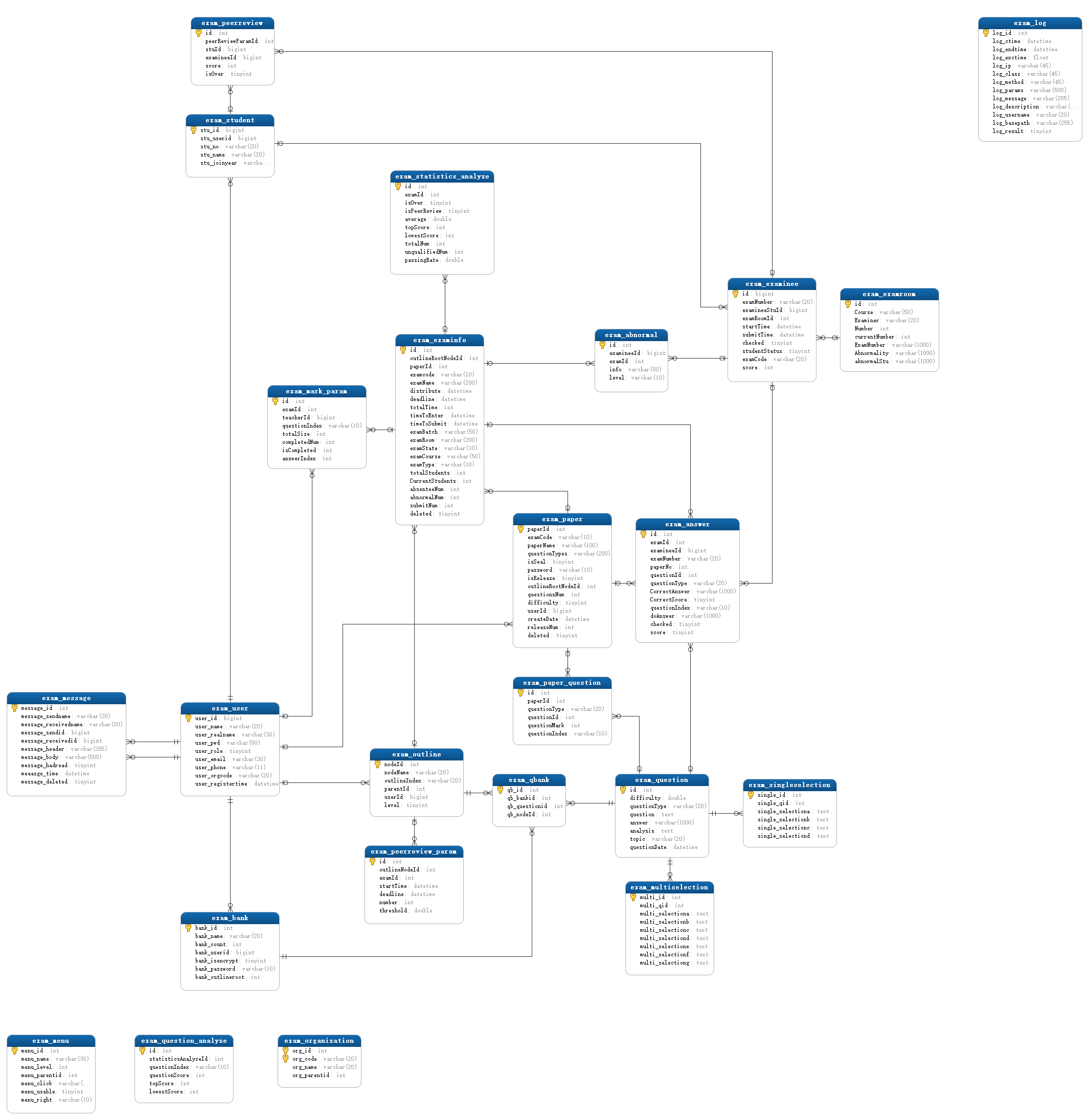


图3.2 数据库总体E-R图

### 用户表

用户表如表3.1所示

表3.1 用户表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| user\_id | bigint |
| user\_name | varchar(20) |
| user\_realname | varchar(30) |
| user\_pwd | varchar(50) |
| user\_role | tinyint |
| user\_email | varchar(30) |
| user\_phone | varchar(11) |
| user\_orgcode | varchar(20) |
| user\_registertime | datetime |

### 菜单表

菜单如表3.2所示

表3.2 菜单表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| menu\_id | int |
| menu\_name | varchar(30) |
| menu\_level | int |
| menu\_parentid | int |
| menu\_click | varchar(100) |
| menu\_usable | tinyint |
| menu\_right | varchar(10) |

### 组织表

组织如表3.3所示

表3.3 组织表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| org\_id | int |
| org\_code | varchar(30) |
| org\_name | varchar(30) |
| org\_parentid | int |

### 题库表

题库如表3.4所示

表3.4 题库表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| bank\_id | int |
| bank\_name | varchar(20) |
| bank\_count | int |
| bank\_userid | bigint |
| bank\_isencrypt | tinyint |
| bank\_password | varchar(10) |
| bank\_outlineroot | int |

### 大纲表

大纲如表3.5所示

表3.5 大纲表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| nodeiId | int |
| nodeName | varchar(20) |
| outlineIndex | varchar(20) |
| parentId | int |
| userId | bitint |
| level | tinyint |

### 问题表

问题如表3.6所示

表3.6 问题表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| id | int |
| difficulty | double |
| questionType | varchar(20) |
| question | text |
| answer | varchar(1000) |
| analysis | text |
| topic | varchar(20) |
| questionDate | datetime |

### 试卷表

试卷如表3.7所示

表3.7 试卷表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| paperId | int |
| examCode | varchar(10) |
| paperName | varchar(100) |
| questionTypes | varchar(200) |
| isSeal | tinyint |
| password | varchar(10) |
| isRelease | tinyint |
| outlineRootNodeId | int |
| questionsNum | int |
| difficulty | tinyint |
| userId | bigint |
| createDate | datetime |
| releaseNum | int |
| deleted | tinyint |

### 考生表

考生如表3.8所示

表3.8 考生表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| id | bigint |
| examNumber | varchar(10) |
| examnieeStuId | bigint |
| examRoomId | int |
| startTime | datetime |
| submitTime | datetime |
| checked | tinyint |
| studentStatus | tinyint |
| examCode | varchar(20) |
| score | int |

### 考场表

考场如表3.9所示

表3.9 考场表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| id | bigint |
| Course | varchar(50) |
| Examiner | varchar(20) |
| Number | int |
| currentNumber | int |
| ExamNumber | varchar(1000) |

### 答案表

答案如表3.10所示

表3.10 答案表

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| id | int |
| examId | int |
| examineeId | bigint |
| examNumber | varchar(20) |
| paperNo | int |
| questionId | int |
| questionType | varchar(20) |
| correctAnswer | varchar(1000) |
| correctScore | tinyint |
| questionIndex | varchar(10) |
| doAnswer | varchar(1000) |
| checked | tinyint |
| score | tinyint |

## 各功能模块流程分析

### 用户注册模块

用户注册模块功能如图3.3所示

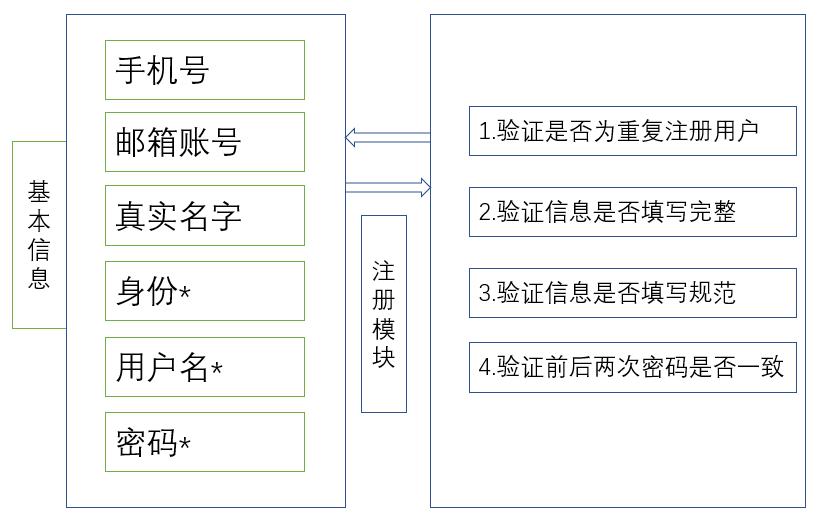


图3.3 用户注册模块

### 用户登录模块

用户登录模块如图3.4所示

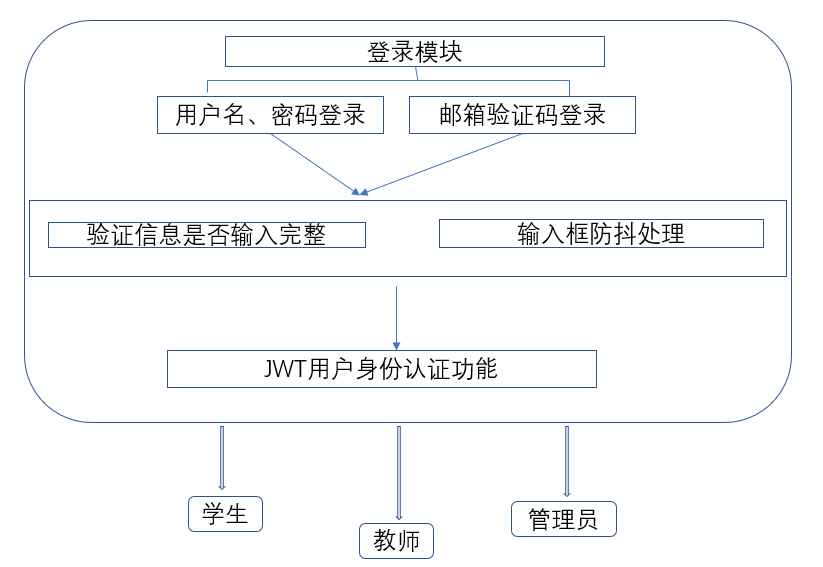


图3.4 用户登录模块

### 系统管理功能

系统功能如图3.5所示

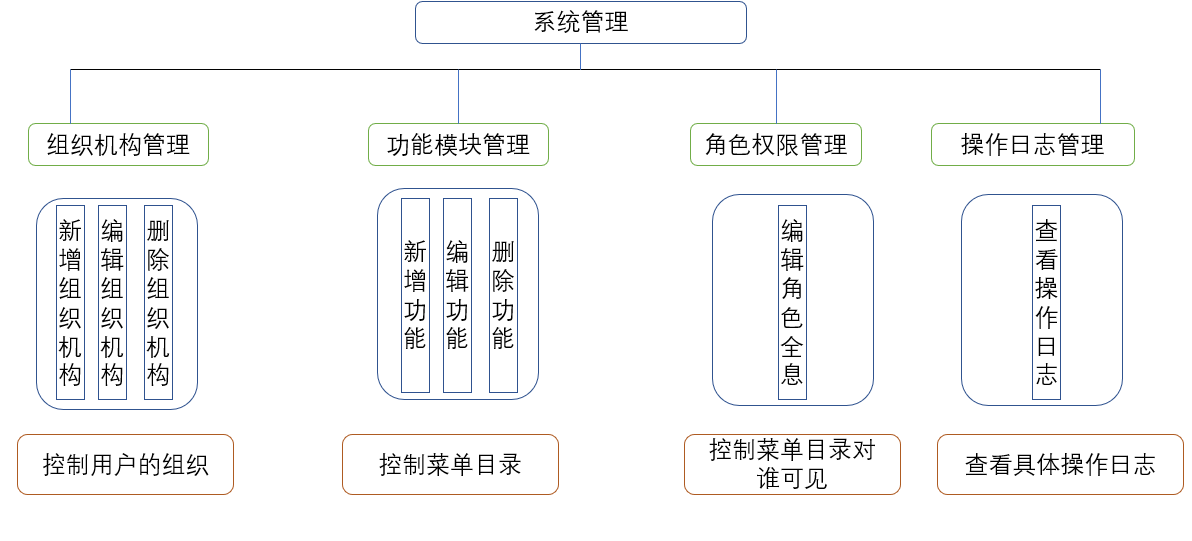


图3.5 系统功能示意图

### 题目相关功能

题目相关功能如图3.6所示

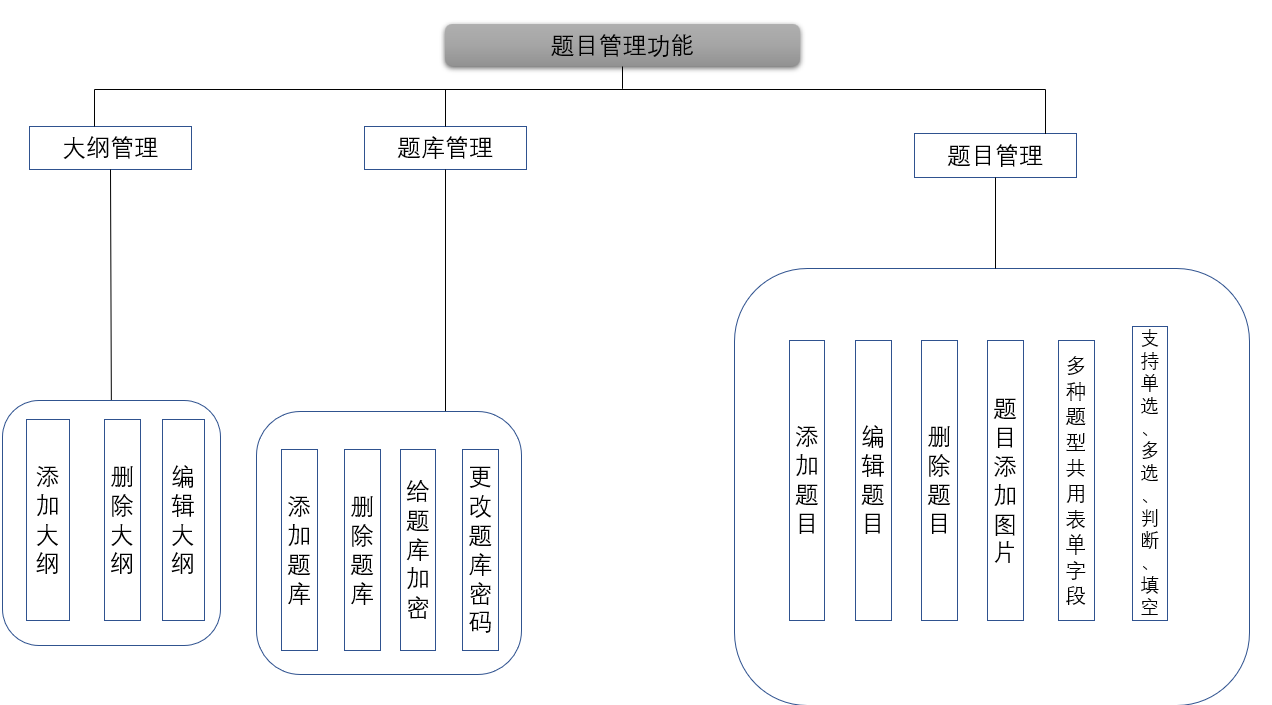


图3.6 题目相关功能

### 试卷的组织功能

试卷的组织功能如图3.7所示

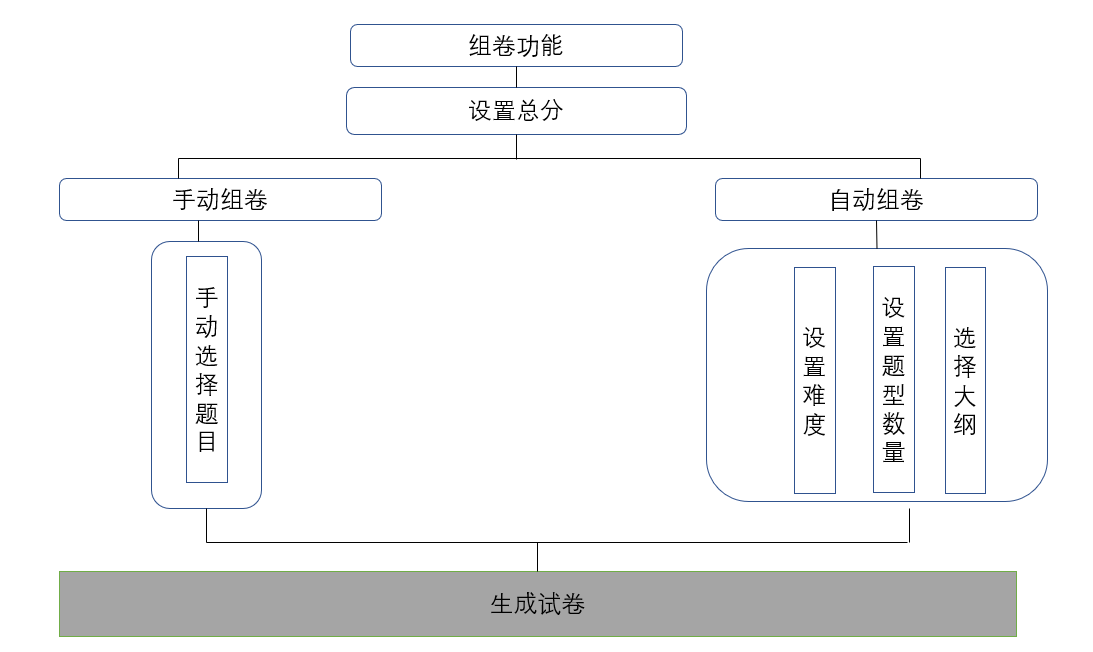


图3.7 试卷组织功能

### 考试功能

考试功能如图3.8所示

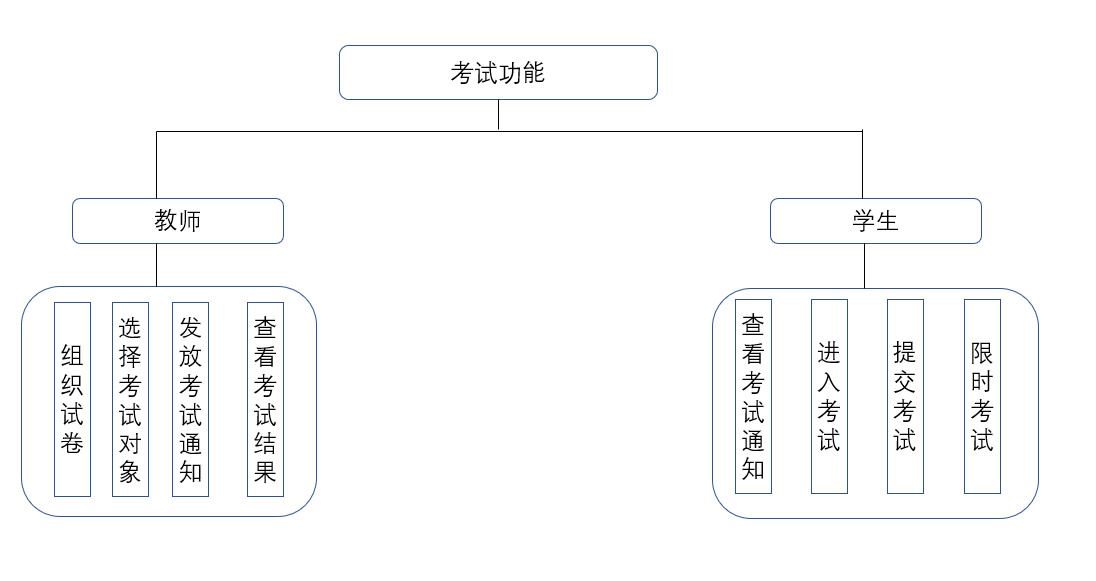


图3.8 考试功能

### 判卷及统计功能

判卷及统计功能如图3.9所示

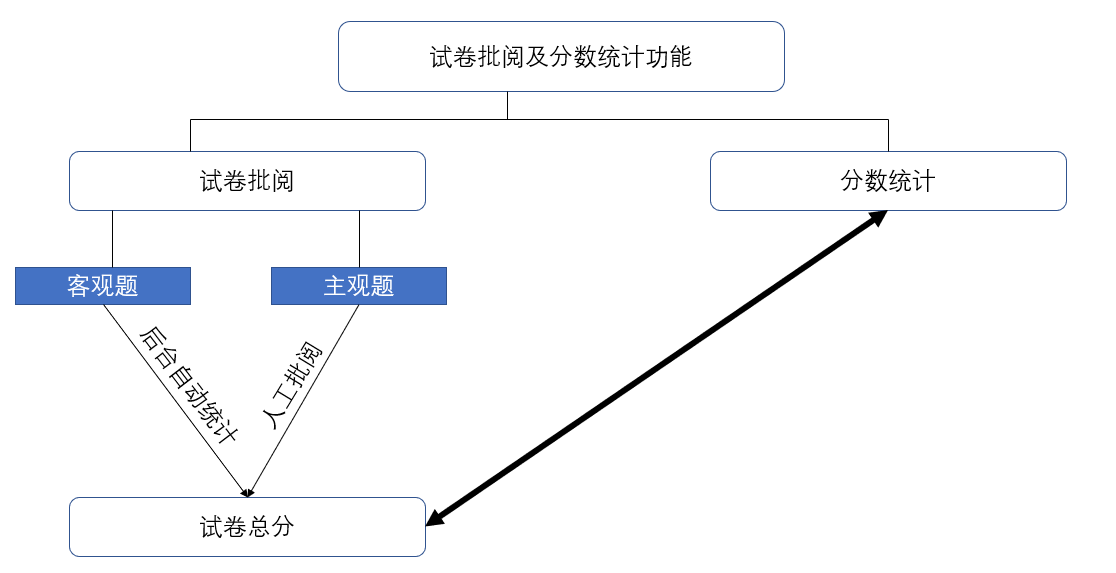


图3.9 判卷及统计功能

# 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。

### 邮箱验证登录是如何实现的

**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，验证成功之后就可以向指定邮箱发送验证消息了。

|  |
| --- |
| /\*\*  \* 发送验证码  \* @param email  \* @return  \*/  @Override  public int sendAuthCodeEmail(String email) {  try {  HtmlEmail mail = new HtmlEmail();  /\*发送邮件的服务器 126邮箱为smtp.126.com,163邮箱为163.smtp.com，QQ为smtp.qq.com\*/  mail.setHostName("smtp.qq.com");  /\*不设置发送的消息有可能是乱码\*/  mail.setCharset("UTF-8");  /\*IMAP/SMTP服务的密码\*/  mail.setAuthentication("596015571@qq.com ","xxa ");  /\*发送邮件的邮箱和发件人\*/  mail.setFrom("发件邮箱", "发件人");  /\*使用安全链接\*/  mail.setSSLOnConnect(true);  /\*接收的邮箱\*/  mail.addTo("18616165@qq.com");  /\*验证码\*/  String code = this.generateVerifyCode(6);  /\*设置邮件的主题\*/  mail.setSubject("登录验证码");  /\*设置邮件的内容\*/  mail.setMsg("尊敬的用户:你好! 登录验证码为:" + code + "(有效期为一分钟)");  mail.send();//发送  } catch (Exception e) {  return 0;  }  return 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。

### 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。为了防止别人伪造，我们可以把数据加上一个只有自己才知道的密钥，做一个签名，把数据和这个签名一起作为 token 发送过去。这样我们就不用保存 token 了，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我用同样的算法和密钥对这个 token 中的数据进行加密，如果加密后的结果和 token 中的签名一致，那我们就可以进行鉴权，并且也能从中取得用户信息。具体流程如图4.1所示。

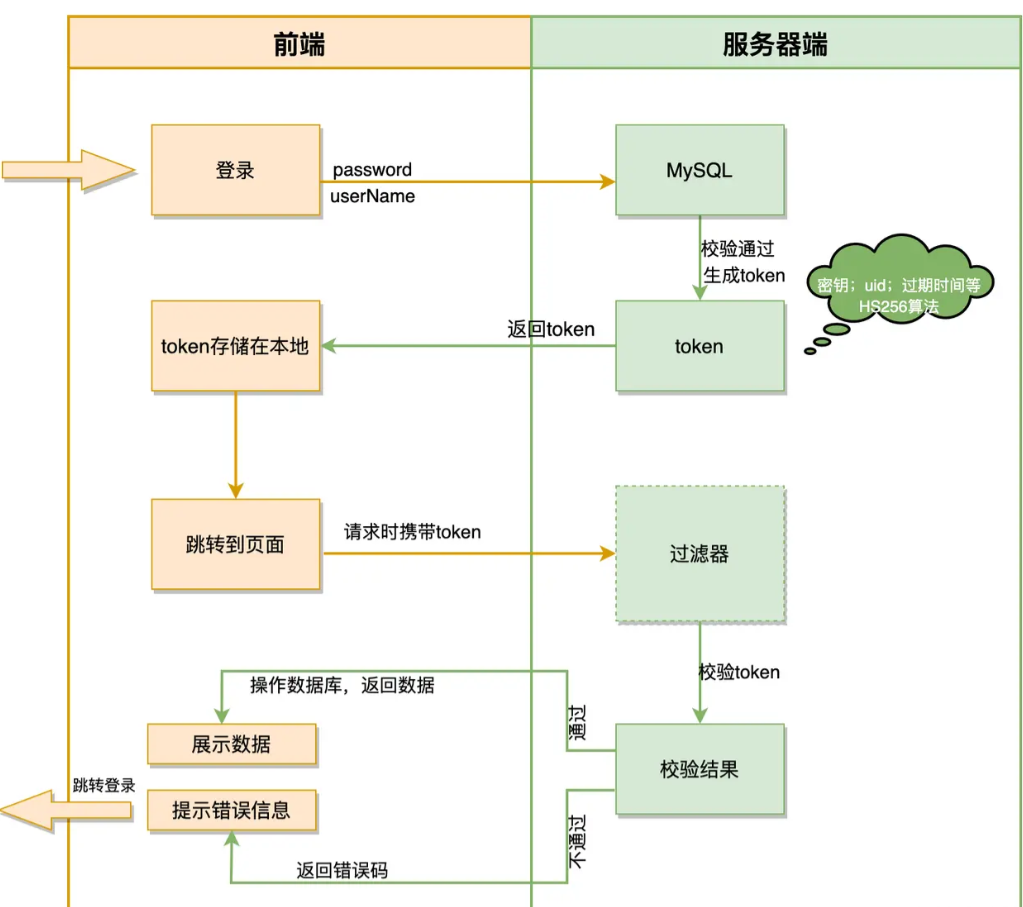


图4.1 登录流程图

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，那么会自动地在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。

## 不同权限显示不同菜单功能的分析

在需求分析中可以看到，不同的角色会有着不同的功能，也就是不同的菜单。菜单界面如图4.2所示。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，就是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。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，如图4.2所示。



图4.2 菜单界面

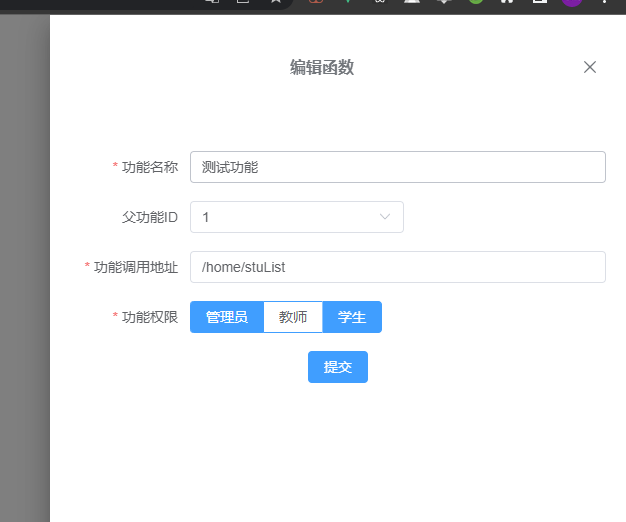


图4.3 编辑菜单

## 通过excel添加用户的实现

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，势必会非常耗时耗力，而且极有可能出错，所以我设计了一个通过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。具体上传界面如图4.4所示。



图4.4 通过excel添加用户

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，索性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。

## 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。题目编辑如图4.5所示。

**如何存储图片与题目(选项)描述**

一开始，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。

**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。

**使用文件+数据库操作**

上文说到使用文件操作的一个典型特点就是慢，不仅需要前台对图片文件进行读取，后台也需要对图片文件进行读取。考虑到计算机的内存没有那么大，同时，上传的图片也没有特别打的，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。

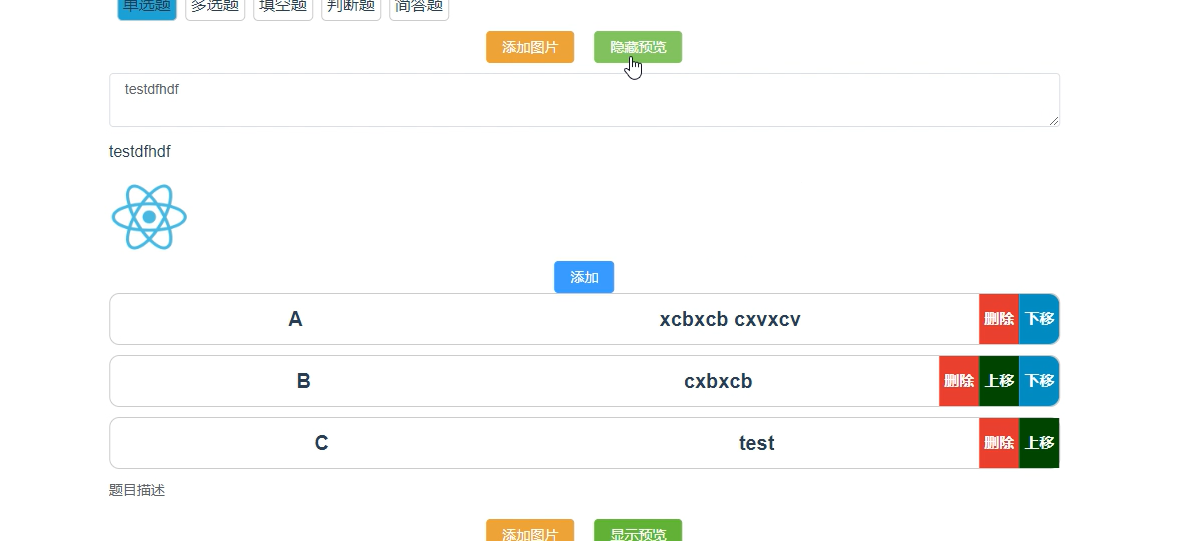


图4.5 题目编辑

**表单如何最大限度地复用**

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，也是一个值得考量地问题。表单复用流程如图4.6所示。

首先，每一中类型地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。

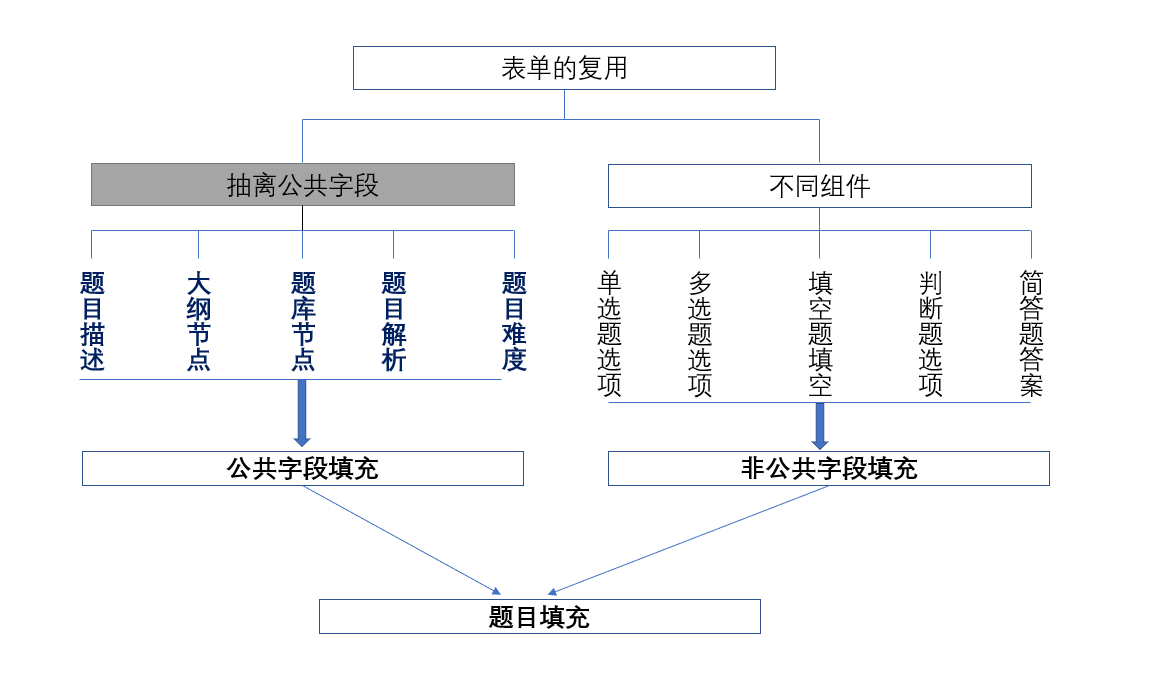


图4.6 表单复用流程图

## 自动组卷的实现分析

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，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。遗传算法的五个阶段如图4.7所示。

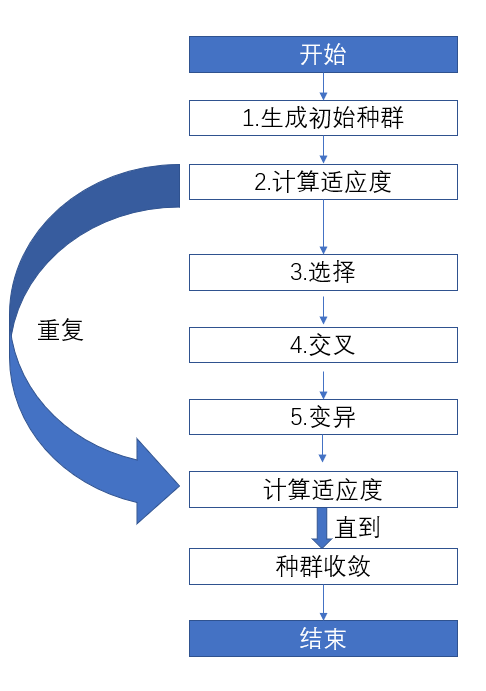


图4.7 遗传算法的五个阶段

**1.初始种群**

该过程从一组称为Population的个体开始。每个人都是您要解决的问题的解决方案。

一个人的特征是一组称为基因的参数（变量） 。基因连接成一个字符串以形成染色体（溶液）。

在遗传算法中，个体的基因集用字符串表示，用字母表示。通常使用二进制值（1 和 0 的字符串）。我们说我们在染色体中编码基因。如图4.8所示

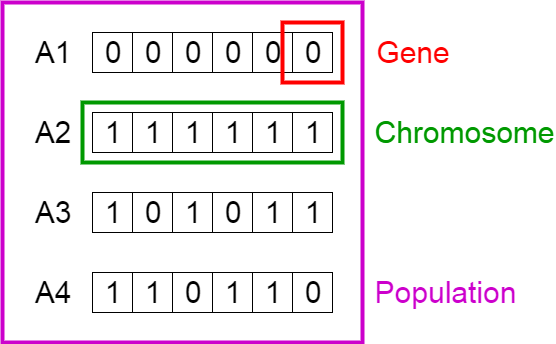


图4.8 初始种群

**2.适应函数**

**适应度函数**决定了个体的适应程度（个体与其他个体竞争的能力）。它给每个人一个**适应分数。**个体被选择进行繁殖的概率取决于其适应度得分。

**3.选择**

选择阶段的想法是选择最适合的个体，让他们将基因传给下一代。

根据他们的健康分数选择两对个体（父母）。适应度高的个体有更多的机会被选择进行繁殖。

**4.交叉**

**交叉**是遗传算法中最重要的阶段。对于每对要交配的父母，从基因中随机选择一个**交叉点。**

例如，假设交叉点为 3，如下图4.9所示。

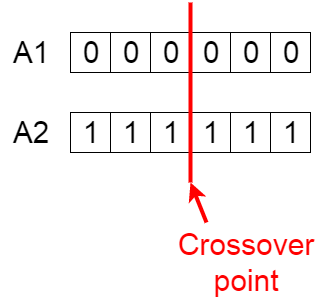


图4.9 交叉点

后代是通过在他们之间交换父母的基因直到达到交叉点而产生的。交叉产生后代如图4.10所示。

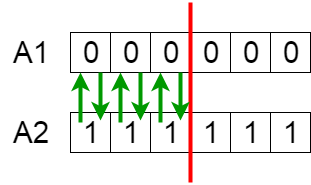


图4.10 交叉产生后代

新的后代被添加到人口中。如图4.11所示。

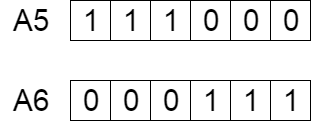


图4.11 新的后代被添加到人口中

**5.突变**

在某些新形成的后代中，它们的某些基因可能会以较低的随机概率发生突变。这意味着位串中的一些位可以翻转。突变前后对比如图4.12所示。

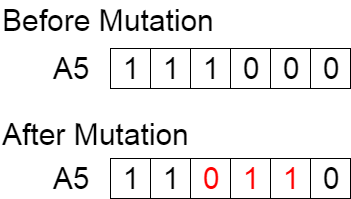


图4.12 突变前后对比

**突变**

如果种群已经收敛（不会产生与上一代有显着差异的后代），则算法终止。那么遗传算法已经为我们的问题提供了一套解决方案。

**遗传算法在自动组卷中的应用**

1. **染色体的编码以及初始群体的设计**

在传统的遗传算法中，用二进制编码时，一份试卷可能有很多题目，用1来表示选中该题目，0表示未选中题目，这样的二进制位串比较长。且在进行变异和交叉遗传算子操作时，各种题型的题目数量不好控制。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**

适应度函数是用来评判试卷群体中个体的优劣程度的指标，遗传算法利用适应度值这一信息来指导搜索方向，而不需要适应度函数连续或可导以及其他辅助信息。因为题数、总分等要求在初始化种群时已经考虑，这里只剩下知识点分布和难度系数要考虑得了。所以适应度函数跟试卷难度系数和知识点分布有关。

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。

知识点分布用一个个体知识点的覆盖率来衡量，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-(1-M/N)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。

1. **遗传因子的设计**

**选择算子**：选择算子的作用在于根据个体的优劣程度决定他在下一代是被淘汰还是被复制。通过选择，将使适应度高的染色体有较大的生存机会。本系统采用随机选择方法。

**交叉算子**：随机交叉点。

**变异算子**：在遗传算法中，变异概率一般较小。这里不分段进行变异，而是对某段上的某个基因进行变异。变异的操作如下：在[1,n]范围内随机生成一个变异位置P，以一定的原则从题库中选择一个变异基因，变异基因的选择原则为：与原基因题型相同的，分数相同，与至少包含原题目一个有效知识点(期望试卷中也有此知识点)。

## 考试功能的实现分析

考试界面如图4.13所示。学生进入考试界面后，系统会设置一个倒计时来题型学生注意时间，如果倒计时时间到了，系统会自动交卷。

### 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，不用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。

因此我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。

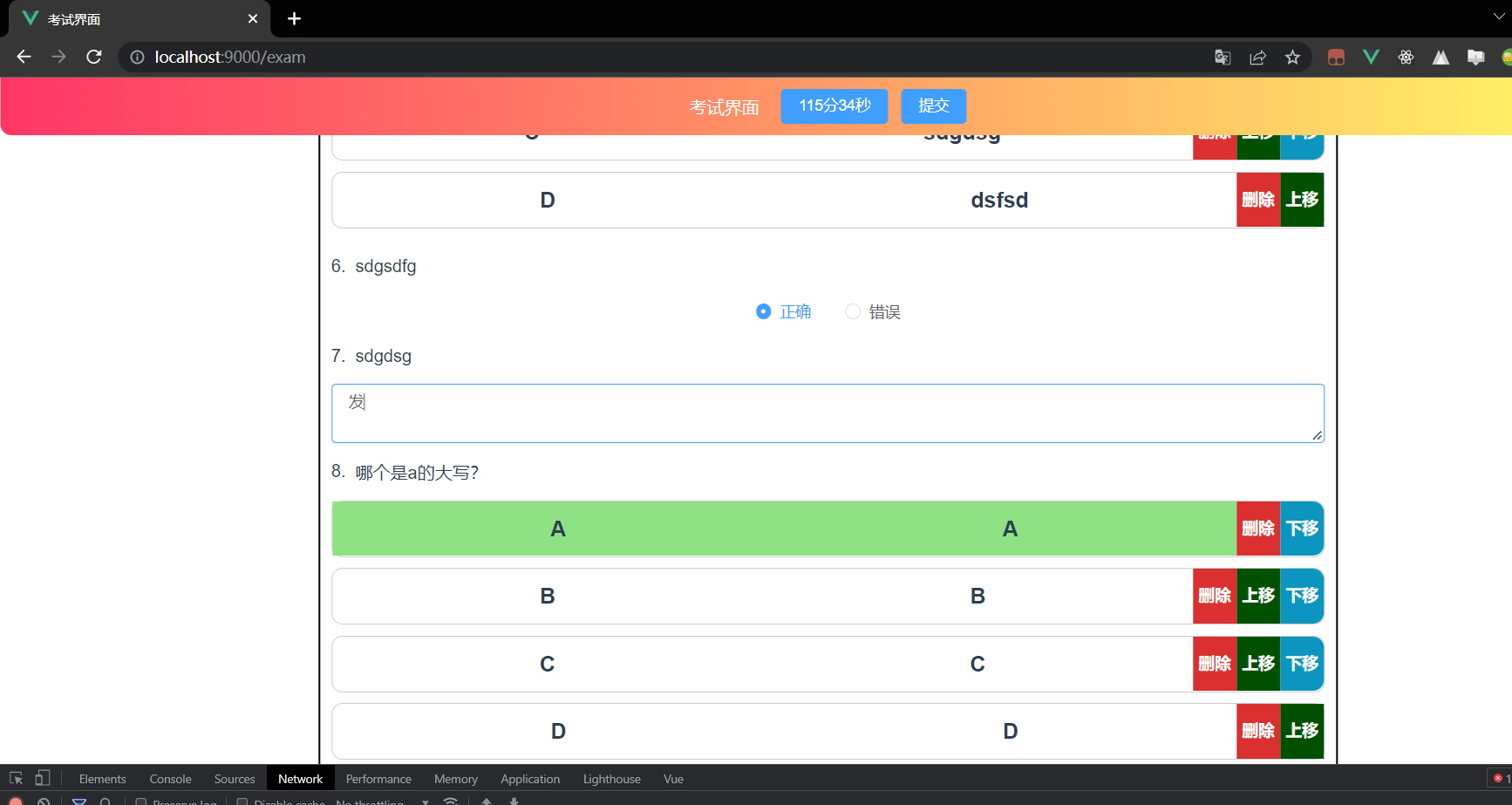


图4.13 考试界面

### 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4.14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。

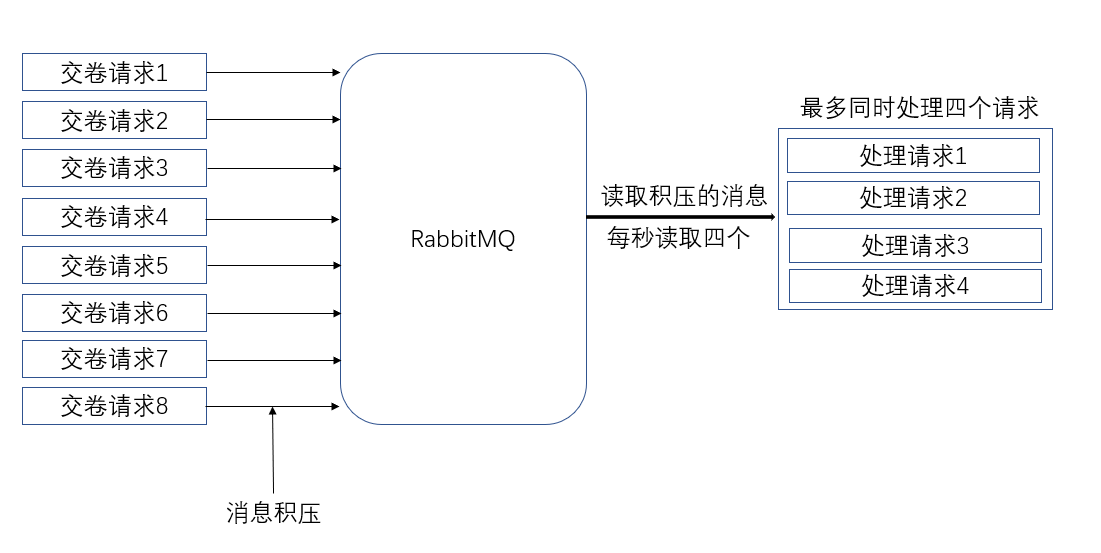


图4.14 RaddbitMQ处理消息流程

## 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4.15所示。

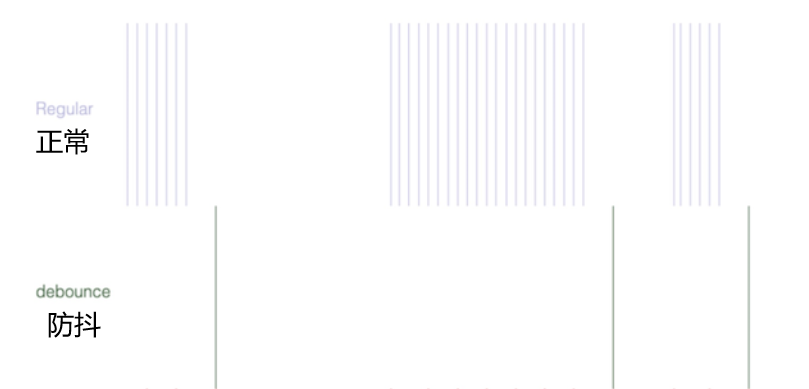


图4.15 防抖示意图

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。

# 系统的测试

## 注册登录相关测试

注册登录测试截图如图5.1所示，测试结果如表5.1所示。

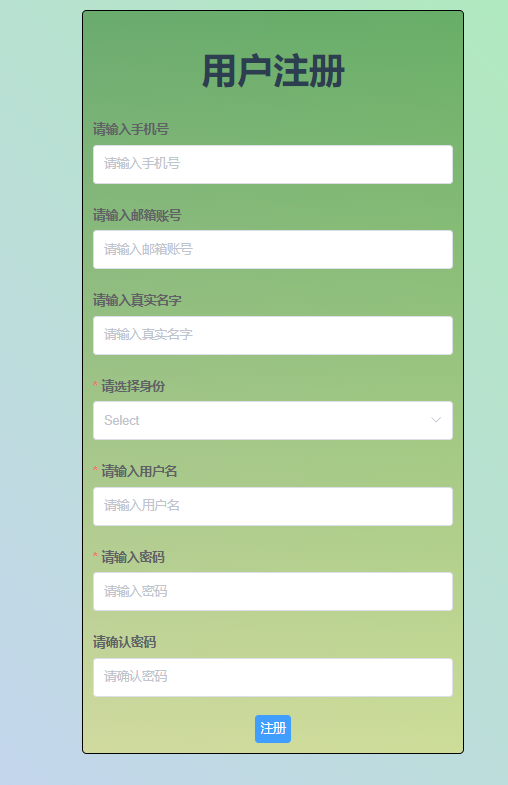


图5.1 注册登录测试截图

表5.1 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5.2、5.3所示，测试结果如表5.2所示。



图5.2 菜单测试截图

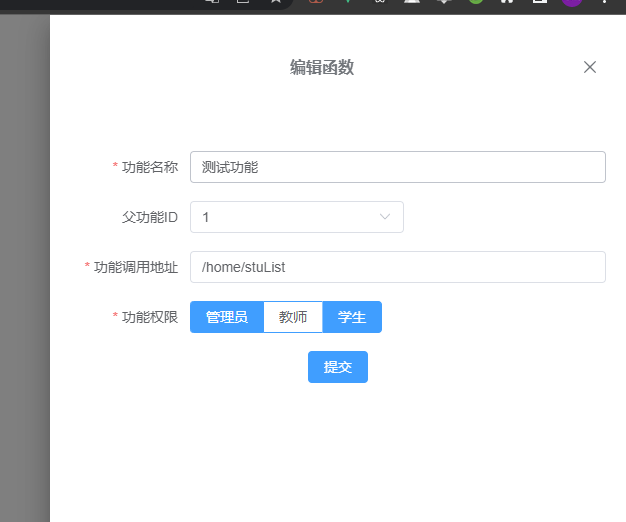


图5.3 函数管理测试截图

表5.2 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5.4、5.4所示，测试结果如表5.3所示。

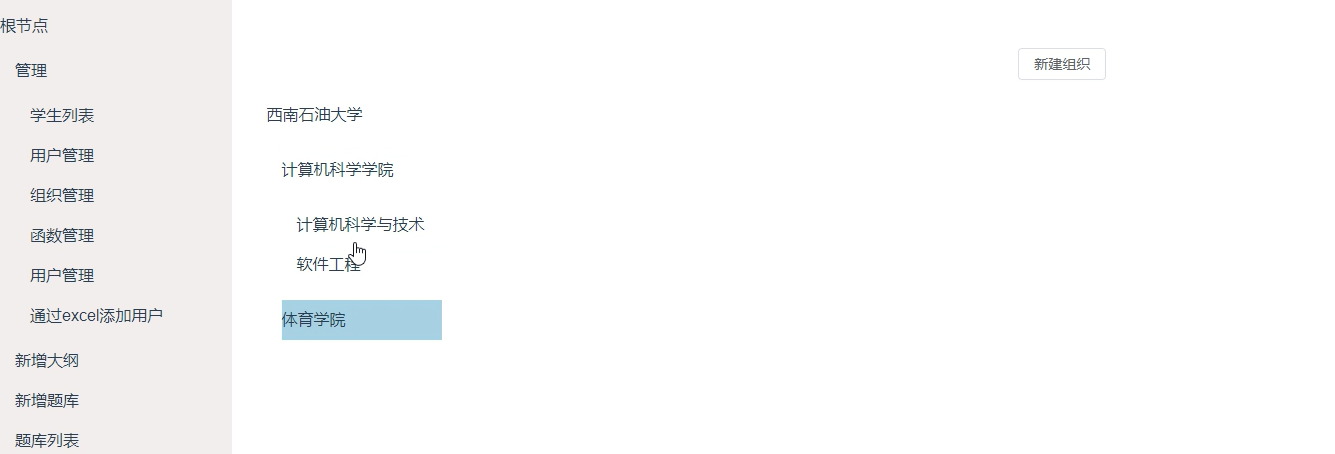


图5.4 组织管理测试截图

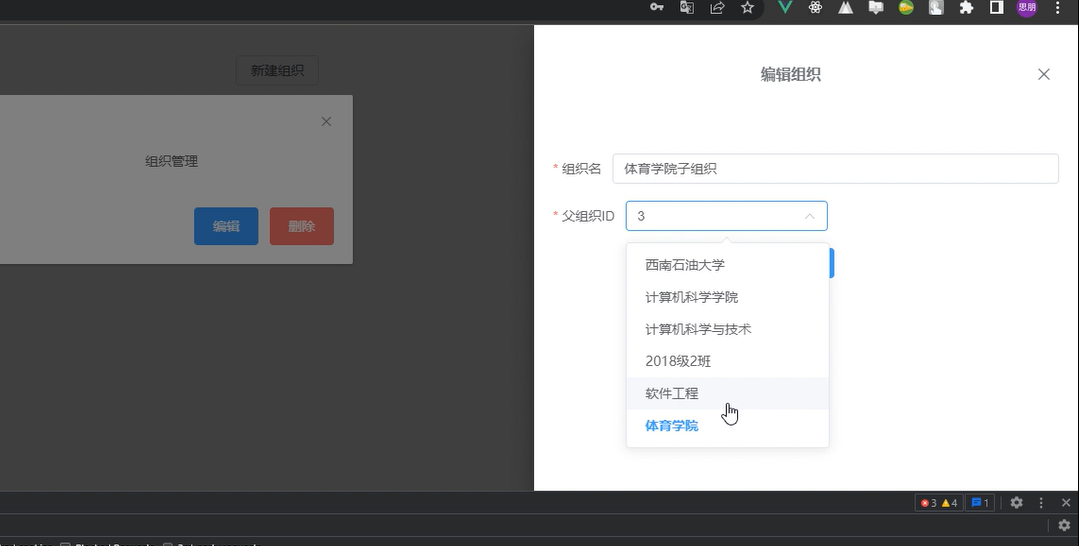


图5.5 组织编辑测试截图

表5.3 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5.6、5.7所示，测试结果如表5.4所示

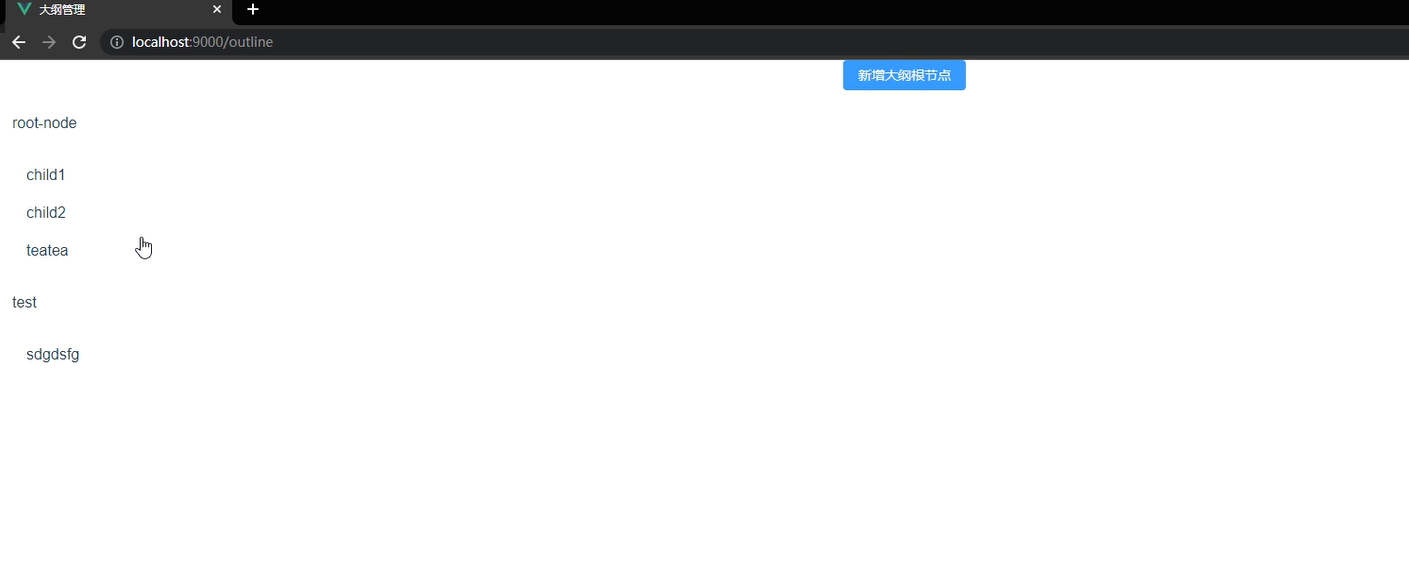


图5.6 大纲管理截图

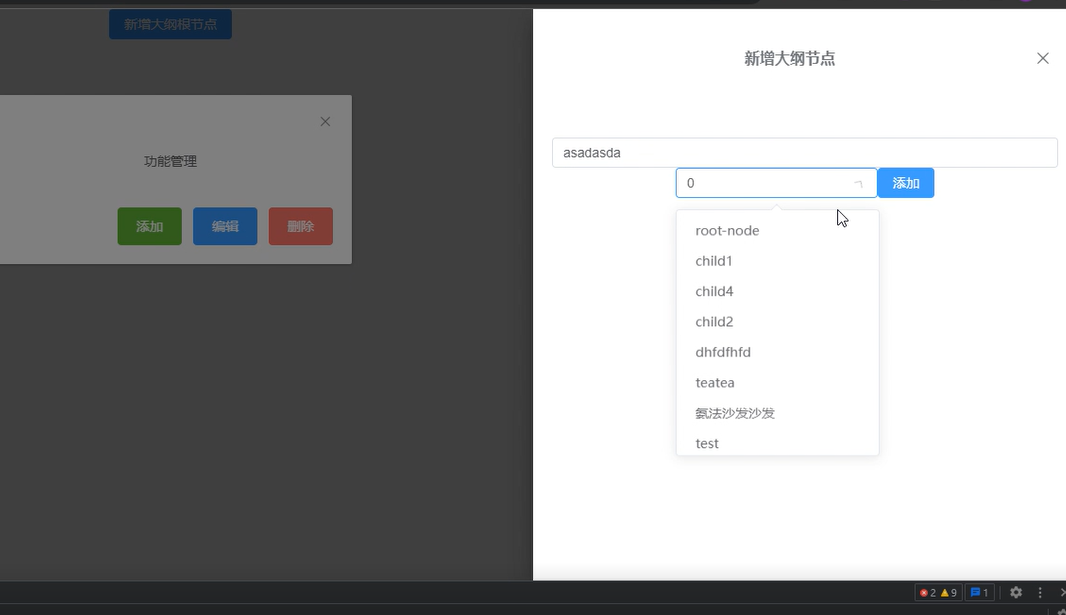


图5.7 编辑大纲截图

表5.4 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5.8、5.9所示，测试结果如表5.5所示

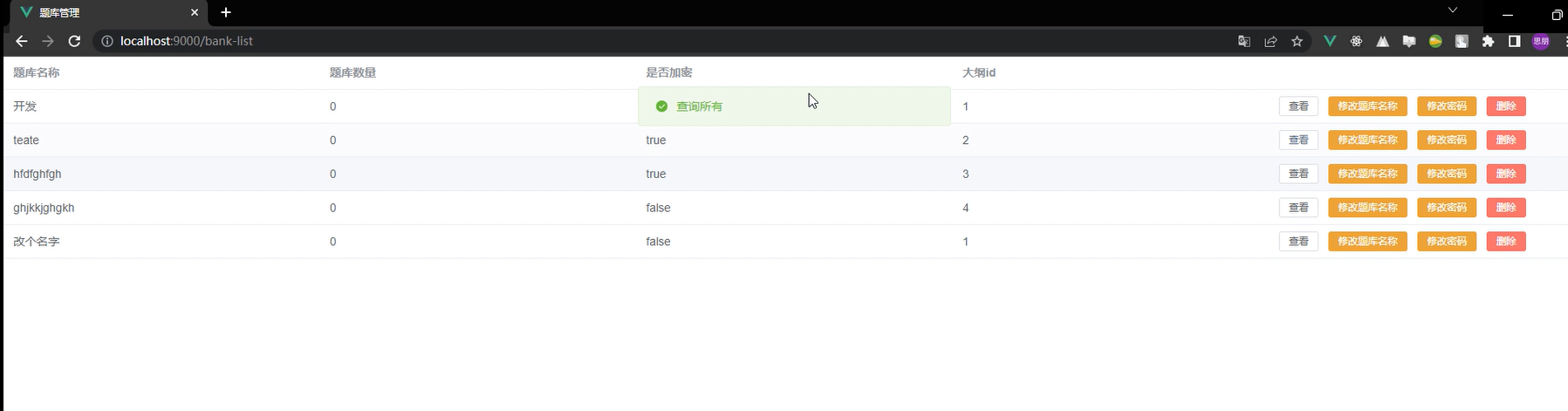


图5.8 题库管理截图

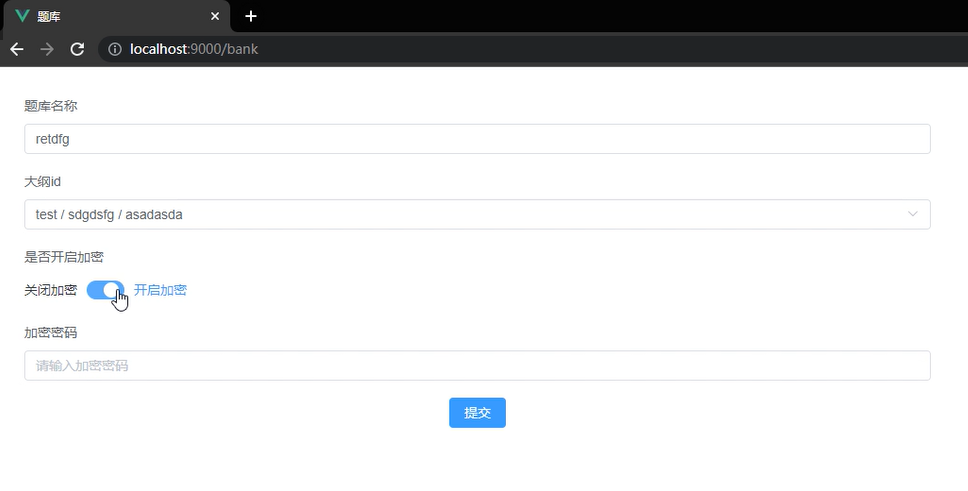


图5.9 编辑题库截图

表5.5 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5.10、5.11所示，测试结果如表5.6所示。

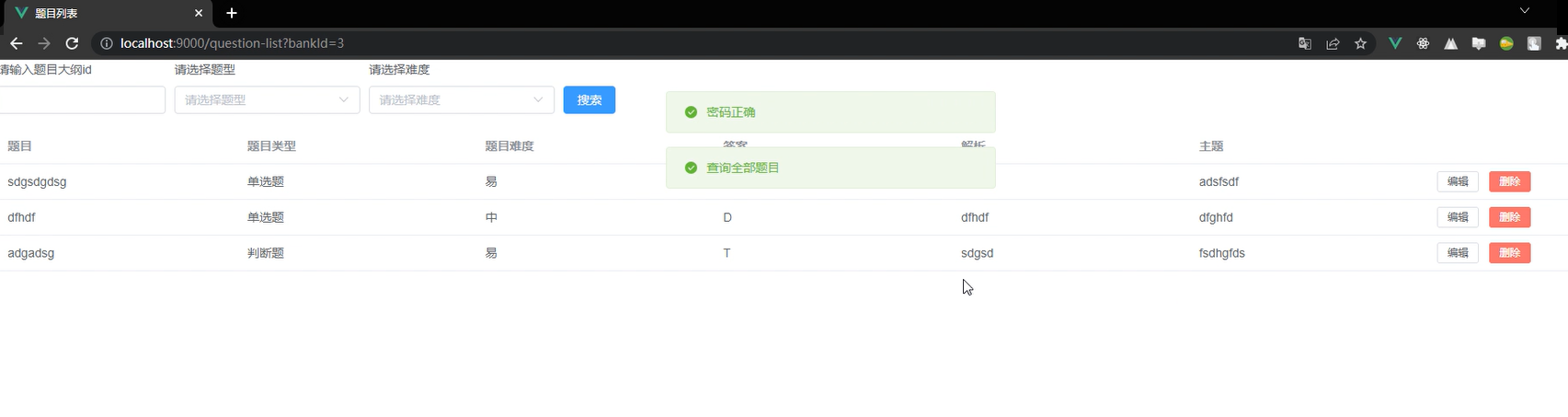


图5.10 题目列表

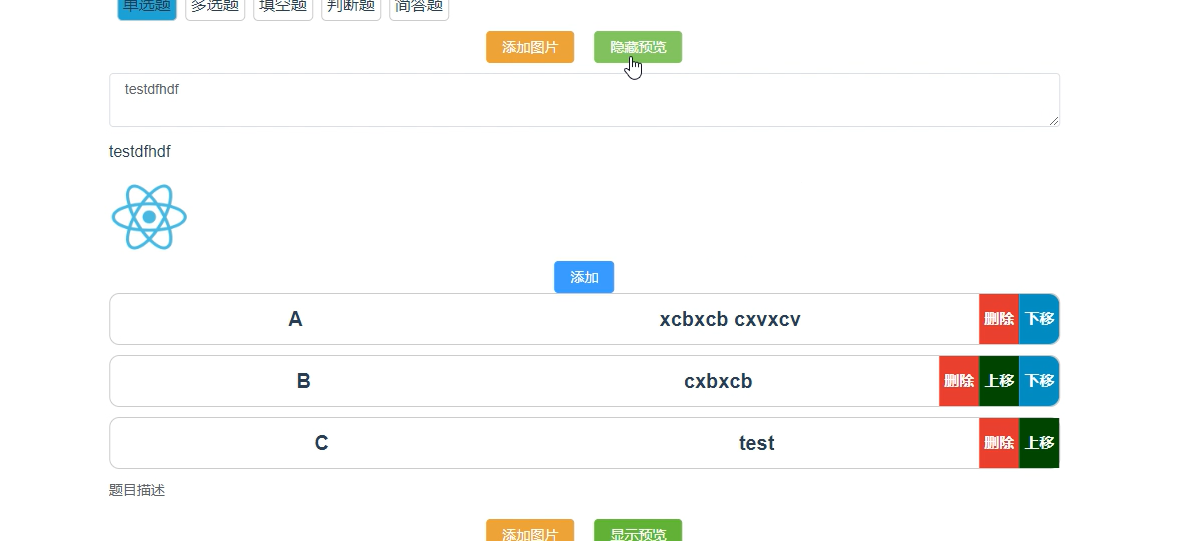


图5.11 编辑题目

表5.6 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5.12、5.13所示，测试结果如表5.7所示

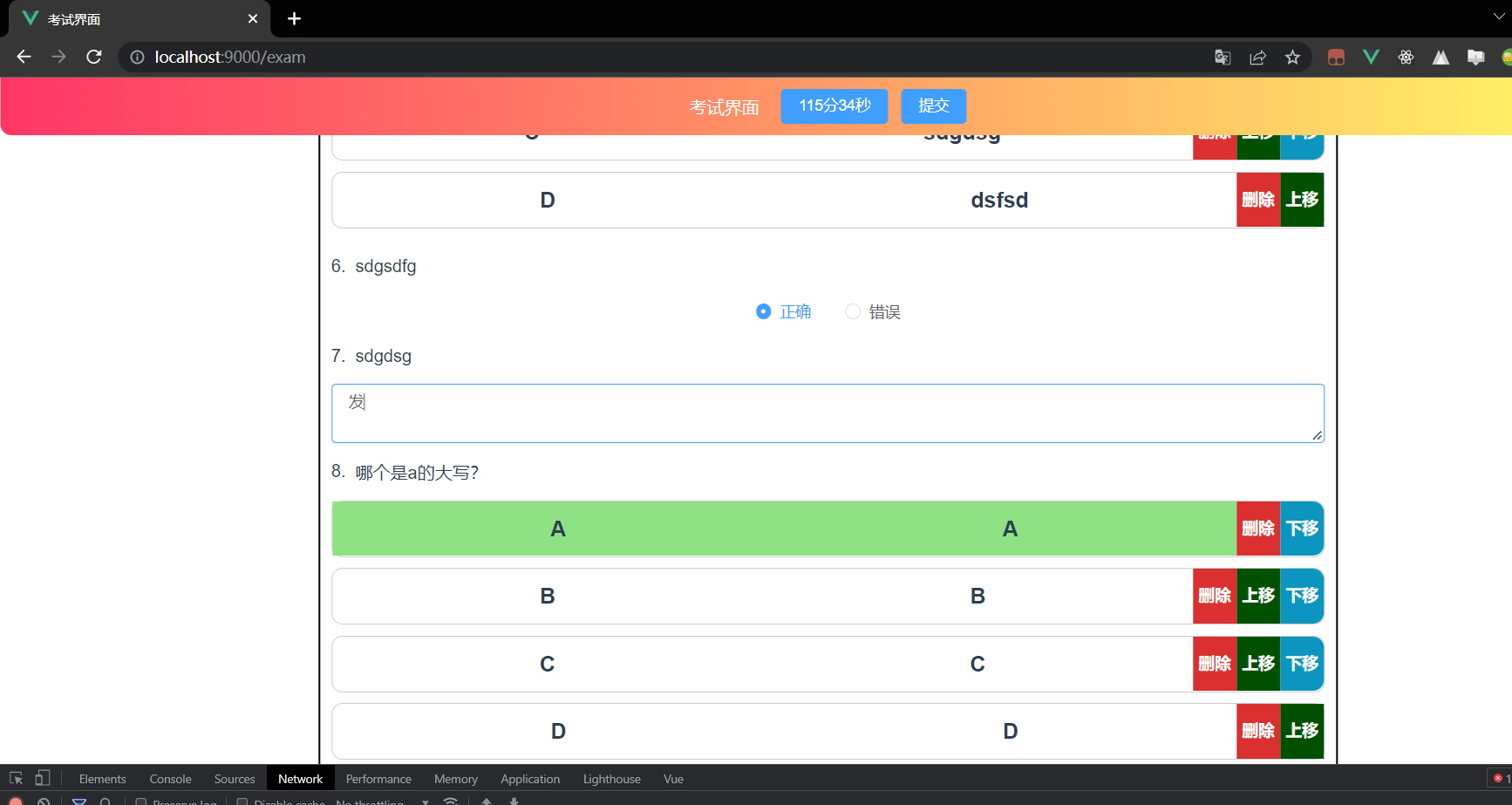


图5.12 参加考试

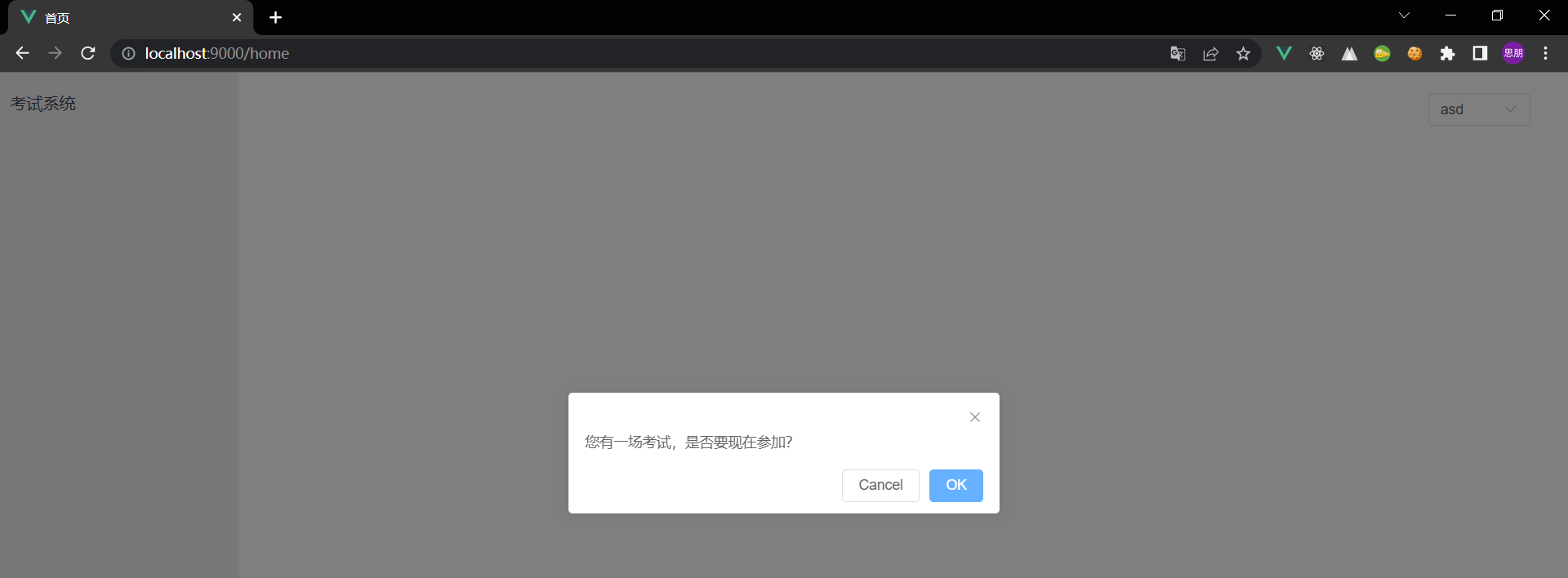


图5.13 考试通知

表5.7 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5.14所示，阅卷结果如表5.8所示

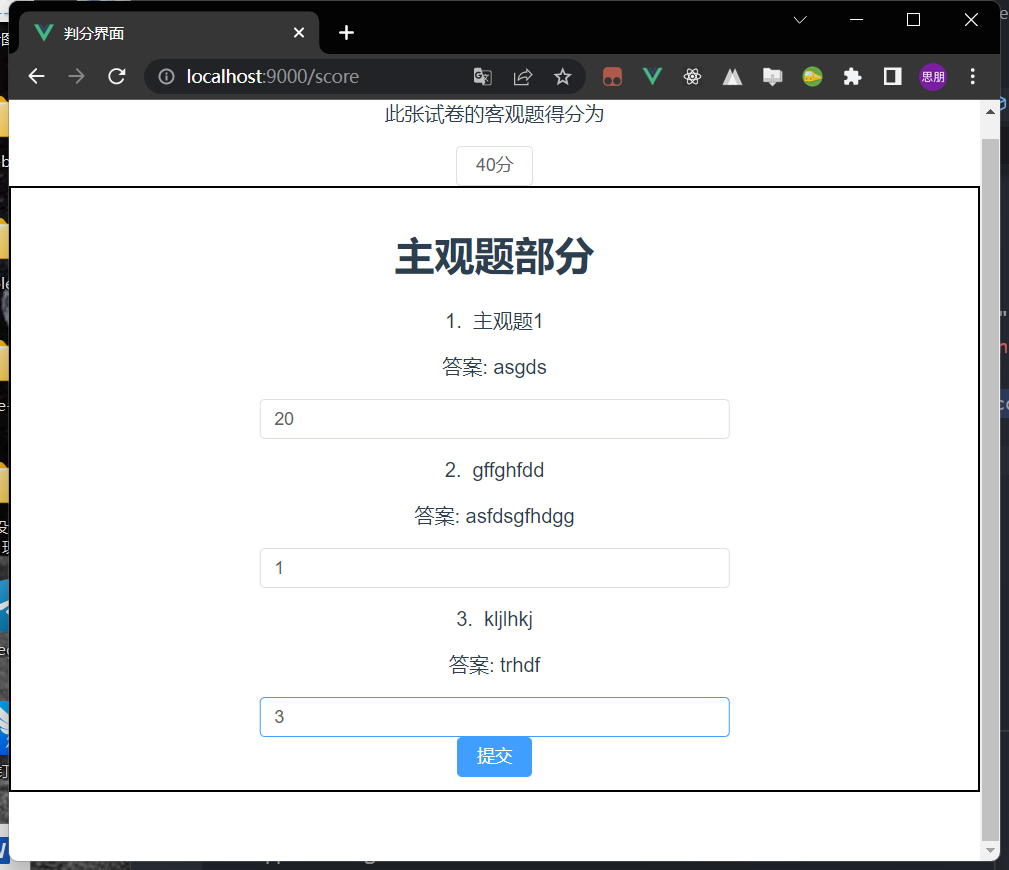


图5.14 阅卷截图

表5.8 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

|  |  |  |
| --- | --- | --- |
| 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 |
| 成绩统计图 | 查看成绩统计图 | 通过 |

## 组卷相关测试

组卷相关测试如图5.15所示，测试结果如表5.10所示

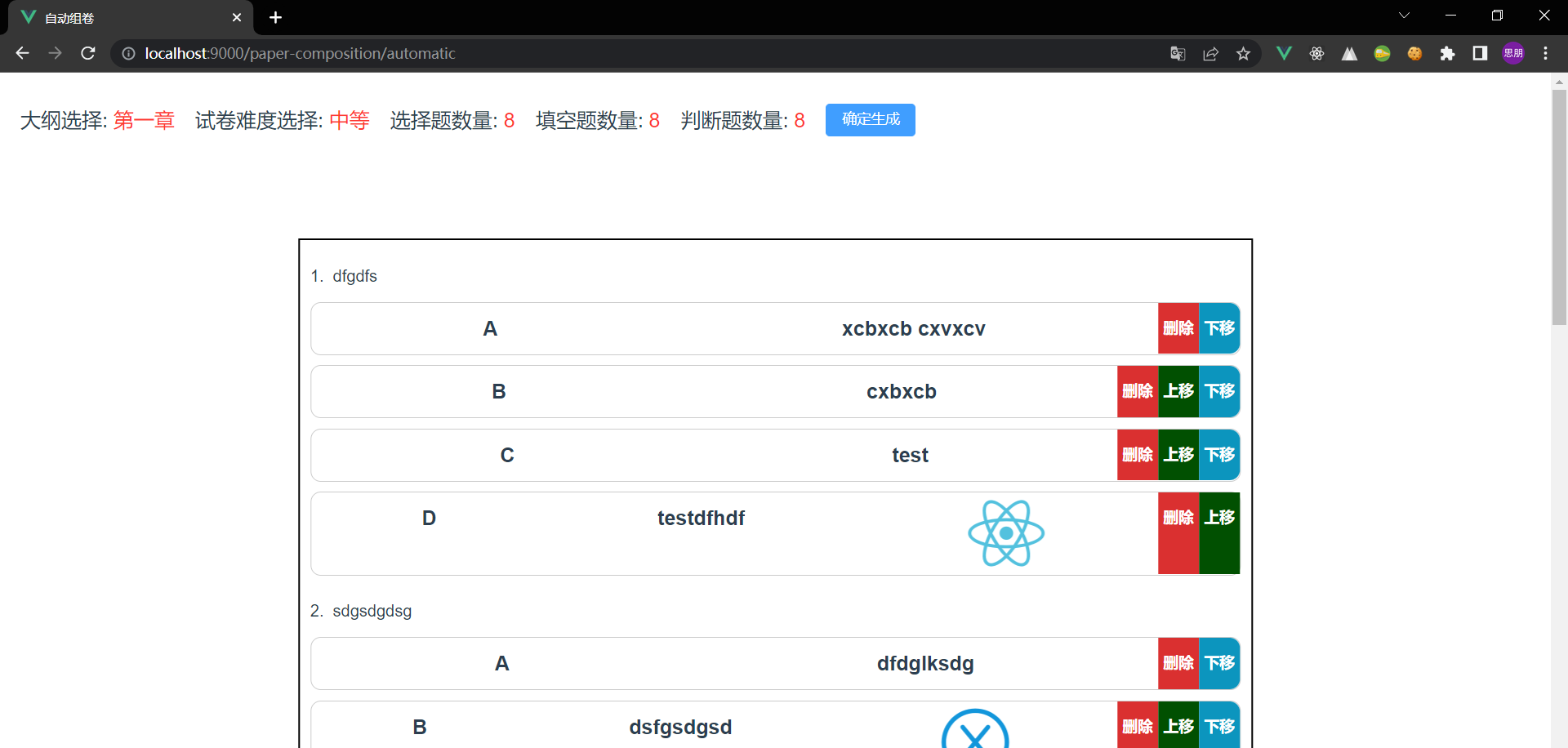


图5.15 组卷

表5.10 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

本文对基于Vue+SprintBoot的教考分离系统的设计与实现进行了详细地阐述。参考各种教考分离系统，利用软件开发的技术，实现了一个集用户管理，功能管理，题库管理，试卷管理的一个教考分离的系统。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. 姜雨薇， 马碧书， 张芸娇.“教考分离”初探[J].科学咨询, 2021(10):22-23。
2. 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。
3. 黄镜流.充分发挥考试的功能[J].职教论坛,1997(2):29。
4. 陈晶， 范丽平， 代书玲， 卢洪秀.高职食品类专业核心课程教考分离的探索与实践.[J]科技视界, 2022(3):148-149。
5. 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。
6. 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。
7. 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。
8. 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。
9. 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。
10. 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。

1. [Donghong PENG](https://pesquisa.bvsalud.org/portal/?lang=pt&q=au:%22Donghong%20PENG%22); [Hong TAN](https://pesquisa.bvsalud.org/portal/?lang=pt&q=au:%22Hong%20TAN%22); [Ziyu HUA](https://pesquisa.bvsalud.org/portal/?lang=pt&q=au:%22Ziyu%20HUA%22). Analysis of the final theory examinatiJon results of pediatrics undergraduates in clinical medicine based on separation of teaching and testing[J].中华医学教育探索杂志，2018：（12）：601-605。

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)。

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. 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. Yonghe DENG.Improve Quality of Teaching and Studying Based on Separation of Examination and Teaching.[C]。
4. Tan, De-rong;Sun, Liang.Method and Practice of Separation of Teaching and Examination on University Talents Training.[C].EDUCATION AND EDUCATION MANAGEMENT。
5. 前端开发技术介绍.Mozilla官网.https://developer.mozilla.org/zh-CN/docs/Learn.
6. Vue.js官方文档.https://cn.vuejs.org/
7. SpringBoot官方文档.https://spring.io/projects/spring-boot
8. MySQL官方文档.https://www.mysql.com/