晋 中 学 院

本科毕业论文（设计）

**题 目** 基于Java在线考试系统

设计与实现

**院 系** 信息技术与工程学院

**专 业** 网络工程

**姓 名** 邓晨阳

**学 号** 1709122108

**学习年限**2017年9月至2021年7月

**指导教师** 杜涛

**申请学位** 理学 学士学位

2021年4月22日

**基于Java在线考试系统设计与实现**

学生姓名：邓晨阳 指导教师：杜涛

摘 要：随着Web技术的发展，传统的考试方式必将面临变革，Web网络考试已经成为一个非常重要的方向。而且今年来，由于新冠疫情的影响，线上教学在全国迅速普及，因此，智能高效的在线考试系统的研发，势在必行。

基于以上需求，本课题采用前端分离架构模式设计和实现在线考试系统，前端采用Vuejs 技术和Element\_UI组件库配合开发，后端使用 Spring Boot 技术实现，并使用MySQL数据做数据持久化，基于REST风格完成接口设计。经现实使用该系统可为老师和学生提供一个方便的在线考试系统，且具有较好的适用性。

关键词**：**在线考试；Vue；Spring Boot；MySQL

**Design and implementation of online examination system based on Java**

**Author’s Name: DENG Chen-yang Tutor: DU Tao**

**ABSTRACT：**With the development of Web technology, the traditional examination method is bound to face changes, and the Web network examination has become a very important direction. In recent years, due to the influence of COVID-19, online teaching has been rapidly popularized throughout the country. Therefore, it is imperative to develop an intelligent and efficient online examination system.

Based on the above requirements, this topic adopts the front-end separation architecture pattern to design and implement the online examination system. The front-end uses Vue.js technology and Element\_UI component library to coordinate the development, the back-end uses Spring Boot technology, and uses MySQL data to do data persistence, and completes the interface design based on the REST style. In practice, the system can provide a convenient online examination system for teachers and students, and has good applicability...

**KEYWORDS:** online examination；Vue；Spring Boot；MySQL

目录

**[1 引言](#_Toc1575371083_WPSOffice_Level1)** **[1](#_Toc1575371083_WPSOffice_Level1)**

[1.1 课题研究背景和意义](#_Toc972642096_WPSOffice_Level2) [1](#_Toc972642096_WPSOffice_Level2)

[1.2 本系统要实现的基本目标和研究内容](#_Toc550186508_WPSOffice_Level2) [1](#_Toc550186508_WPSOffice_Level2)

[1.3 论文组织结构](#_Toc2067539621_WPSOffice_Level2) [2](#_Toc2067539621_WPSOffice_Level2)

**[2 基本知识和基本原理](#_Toc972642096_WPSOffice_Level1)** **[2](#_Toc972642096_WPSOffice_Level1)**

[2.1 Java语言的简介](#_Toc705518040_WPSOffice_Level2) [2](#_Toc705518040_WPSOffice_Level2)

[2.2 IntelliJ IDEA工具简介](#_Toc1384483193_WPSOffice_Level2) [2](#_Toc1384483193_WPSOffice_Level2)

[2.3 Vue技术](#_Toc1023709506_WPSOffice_Level2) [2](#_Toc1023709506_WPSOffice_Level2)

[2.4 Spring Boot技术](#_Toc1994171225_WPSOffice_Level2) [4](#_Toc1994171225_WPSOffice_Level2)

[2.5 MySQL数据库介绍](#_Toc258499846_WPSOffice_Level2) [4](#_Toc258499846_WPSOffice_Level2)

**[3 系统分析](#_Toc550186508_WPSOffice_Level1)** **[5](#_Toc550186508_WPSOffice_Level1)**

[3.1 可行性分析](#_Toc247493841_WPSOffice_Level2) [5](#_Toc247493841_WPSOffice_Level2)

[3.2 需求分析](#_Toc2100645095_WPSOffice_Level2) [5](#_Toc2100645095_WPSOffice_Level2)

[3.3 非功能性需求](#_Toc910954985_WPSOffice_Level2) [6](#_Toc910954985_WPSOffice_Level2)

**[4 系统详细设计](#_Toc2067539621_WPSOffice_Level1)** **[7](#_Toc2067539621_WPSOffice_Level1)**

[4.1 设计原则](#_Toc1009513432_WPSOffice_Level2) [7](#_Toc1009513432_WPSOffice_Level2)

[4.2 设计模式](#_Toc1771440324_WPSOffice_Level2) [7](#_Toc1771440324_WPSOffice_Level2)

[4.3 数据库设计](#_Toc2031727107_WPSOffice_Level2) [8](#_Toc2031727107_WPSOffice_Level2)

**[5 系统实现](#_Toc705518040_WPSOffice_Level1)** **[13](#_Toc705518040_WPSOffice_Level1)**

[5.1 用户登录模块](#_Toc100016402_WPSOffice_Level2) [13](#_Toc100016402_WPSOffice_Level2)

[5.2 前台考试子系统](#_Toc1643456460_WPSOffice_Level2) [13](#_Toc1643456460_WPSOffice_Level2)

[5.3 后台管理子系统](#_Toc638055506_WPSOffice_Level2) [17](#_Toc638055506_WPSOffice_Level2)

**[6 系统的测试](#_Toc1384483193_WPSOffice_Level1)** **[19](#_Toc1384483193_WPSOffice_Level1)**

[6.1 测试方法](#_Toc1413039871_WPSOffice_Level2) [19](#_Toc1413039871_WPSOffice_Level2)

[6.2 黑盒测试用例](#_Toc2086943371_WPSOffice_Level2) [20](#_Toc2086943371_WPSOffice_Level2)

[表6-1 用户登录操作测试](#_Toc406829946_WPSOffice_Level2) [20](#_Toc406829946_WPSOffice_Level2)

[表6-2 主要功能操作测试](#_Toc2970374_WPSOffice_Level2) [20](#_Toc2970374_WPSOffice_Level2)

**[7 结束语](#_Toc1023709506_WPSOffice_Level1)** **[21](#_Toc1023709506_WPSOffice_Level1)**

[7.1 总结](#_Toc530951937_WPSOffice_Level2) [21](#_Toc530951937_WPSOffice_Level2)

[7.2 展望](#_Toc914651874_WPSOffice_Level2) [21](#_Toc914651874_WPSOffice_Level2)

**[参考文献](#_Toc1994171225_WPSOffice_Level1)** **[1](#_Toc1994171225_WPSOffice_Level1)**

1 引言

1.1 课题研究背景和意义

现在网络技术已经逐渐渗透到社会生活的各个层面，传统的考试方式将面临变革，而网络考试是一个非常重要的方向。基于Web技术的网络考试系统可以借助Internet在全球范围内进行，既可以在本地进行，也可以在不同的地方进行，大大扩展了考试的灵活性。可以根据题库的内容立即生成试卷，避免考试前的护送。而且，大量的标准化试题可以通过计算机进行评判，大大提高了阅卷的效率。还可以直接将结果发送到数据库、统计、排序等操作。所以现在网络更好的测试方法测试,测试内容在服务器上,考生通过姓名、准考证号码和口令进行登录,考试答案也存放在服务器中,这样考试的公平性、答案的安全性可以得到有效的保证。因此，使用网络考试将是未来考试的发展趋势。

1.2 本系统所要实现的基本目标和研究内容

本系统分为前端考生系统和后端信息管理系统两个子系统，需要给用户良好的体验，尽可能为用户提供相关需求。具体设计目标如下:

（1）系统管理员和教师可以使用的功能包括: 考试管理、题库管理、成绩查询、学生管理和教师管理。

（2）学生可以使用的功能有：管理中心、我的试卷、答题详情页、我的练习、我的分数、给我留言

（3）处理用户需求时高效快捷；如果待响应，会给用户相应提示。

（4）用户界面美观整齐，给用户良好的视觉体验。

1.3 论文组织结构

本论文由下面几个部分组成：

第一章 引言，阐述选题意义。

第二章 介绍系统开发过程中使用到的开发技术、开发工具以及系统的基本原理

第三章 系统分析，首先从技术可行性、经济可行性来分析系统是否可行，并对系统进行了需求分析。

第四章是系统的详细设计，详细阐述了系统具体模块的算法和设计模式。

第五章 系统实现，主要阐述系统中各个模块的具体实现。

第六章通过一定的测试方法对程序的功能部分进行测试。

第七章 结束语，对本次系统的设计开发做出工作总结，并对将来工作的一些展望。

# 2 基本知识和基本原理

2.1 Java语言的简介

Java编程语言具有简单易学、面向对象、平台无关性、可靠性、安全性、编译与解释的特点。并且Java的生态完善，有丰富的组件和框架可供使用，测试起来也比较方便，基于这些优势，本系统选择Java作为开发的编程语言

2.2 IntelliJ IDEA工具简介

IntellJ IDEA是由JetBrains软件公司开发的商业销售的Java集成开发环境工具软件。与市场上主流的Java集成开发环境相比，IDEA是最受欢迎的IDE，具有更多的智能和便利性。

## 2.3 Vue技术

vue是用于构建用户界面vue的渐进框架。js是MVVM的实现者，核心就是实现了DOM监听和数据绑定。vue且拥有以下特点：

·轻量级

·易上手，学习曲线平稳，文档齐全

·开源，社区活跃度高

·......

基于这些优势，本系统选择使用vue作为前端开发的技术，并且使用到以下vue插件：

(1) vue-cli

vue-cli是官方提供的一个脚手架，用于快速生成一个vue的项目模板。主要的功能有：

·统一的目录结构

·本地调试

·热部署

·单元测试

·集成打包上线

(2) vue-router

vue router 是 vue.js 官方的路由管理器。它和 vue.js 的核心深度集成，让构建单页面应用变得易如反掌。包含的功能有：

·嵌套的路由/视图表

·模块化的、基于组件的路由配置

·路由参数、查询、通配符

·基于 vue.js 过渡系统的视图过渡效果

·细粒度的导航控制

·带有自动激活的 CSS class 的链接

·自定义的滚动条行为

(3) vue-cookies

在vue中如果想要操作cookie，除了可以使用自己封装好的操作cookie的方法之外，还可以使用vue-cookies插件，这是一个简单的vue.js插件，专门用于在vue中处理浏览器的cookie操作，vue-cookies没有依赖关系，它可以独立存在，对vuejs友好。

(4) Vuex

vuex是适用于在vue项目开发时使用的状态管理工具。如果在一个项目开发中频繁的使用组件传参的方式来同步data中的值，一旦项目变得很庞大，管理和维护这些值将是相当棘手的工作。为此，vue为这些被多个组件频繁使用的值提供了一个统一管理的工具——vuex。在带有vuex的vue项目中，您只需要在vuex中定义这些值即可在整个vue项目的组件中使用它们。

(5) Axios

Axios是近年来备受推崇的网络请求库。它以基于承诺的方式封装了浏览器的XMLHttpRequest和服务器端节点http请求，这使我们能够按照es6的建议异步处理网络请求。

(6) Element-UI

VUE的核心思想是组件和数据驱动，但是每个组件要编写自己的模板样式，添加事件，数据等非常麻烦，因此，饿了么团队推出了基于VUE2.0的组件库，名称为element-ui，其中提供了丰富的PC端组件，可以帮助我们快速完成网站的样式设计。

## 2.4 Spring Boot技术

Spring Boot是一种从Spring framework 4.0衍生而来的框架，用于快速构建独立的、基于生产级别的Spring应用程序，可以以最小的依赖性引入该框架来构建Spring应用程序。并且它还具有以下特点：

（1）具有嵌入式Tomcat，Jetty，Undertow或Reactor Netty (不需要部署war文件)

（2）尽可能的自动配置（@AutoConfiguration）Spring和第三方库

（3）提供特定于生产环境的功能，如运行指标、运行状态的检查和外部化的配置

（4）无需麻烦而冗余的XML配置，一切都可以使用Java配置

2.5 MySQL数据库介绍

MySQL 是⼀种关系型数据库，在Java企业级开发中⾮常常⽤，因为 MySQL 是开源免费的，并且⽅便扩展。Alibaba数据库系统也⼤量⽤到了MySQL，因此它的稳定性是有保障的。MySQL的默认端⼝号是3306。MySQL有着诸多特性：

1. 性能卓越服务稳定，很少出现异常宕机。

（2）开放源代码且无版权制约，自主性强、使用成本低。

（3）历史悠久、社区及用户非常活跃，遇到问题，可以很快获取到帮助。

（4）软件体积小，安装使用简单，并且易于维护，安装及维护成本低。

（5）跨平台特性，支持多种操作系统，同时MySQL对很多编程语言的支持也相当好。

（6）开发成本低，经济性高，查询速度快，运行效率高。

这些只是MySQL优越性的一部分，正是因为MySQL这些优点，使得它成为最受用户喜爱的数据库之一。

# 3 系统分析

3.1 可行性分析

**（1）技术可行性**

本小程序采用的开发软件为IntellJ IDEA。以MySQL作为数据库，利用Vue技术，可以高效快捷的开发本程序。从开发环境上看，环境已经搭建完毕，从开发人员上看，开发者都具有丰富的开发经验和技术储备。因此该系统在技术上讲是完全可行的。

**（2）经济可行性**

由于本工程是作为毕业设计课题，基本无需经济投入。而且本系统对硬件要求也并不太高，一台普通的笔记本电脑就可以满足开发需要。而且项目设计完成后既可以投入使用，完全不需要其他的经济消耗。而且系统部署到云端上线之后，还可以带来可观的经济收益，完全可以填补前期开发的成本费用。因此该系统在经济上是完全可行的。

3.2 需求分析

1. 系统管理员功能

在本系统中，系统管理员登录后进入到后台管理子系统中，模块功能有：题库管理、考试管理、成绩查询、教师管理、学生管理这五大功能模块。系统管理员用例图如图3-1所示。

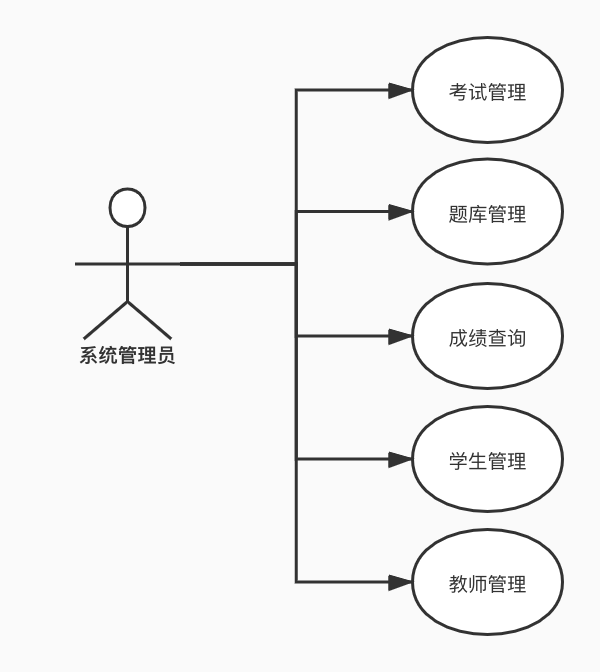


图3-1 管理员用例图

1. 教师功能

在该系统中，教师用户登录后进入后台管理子系统。模块功能包括: 题库管理、考试管理、成绩查询和学生管理。。教师用例图如图3-2所示。

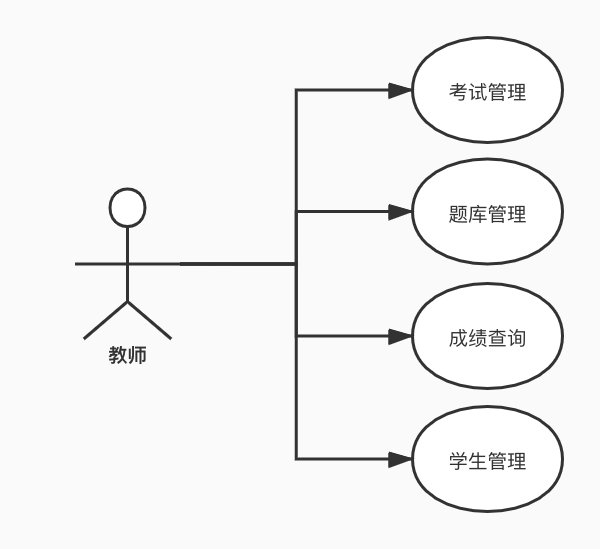


图3-2 教师用例图

1. 学生功能

在该系统中，学生用户登录后进入前台考试子系统。模块功能包括: 我的试卷，我的练习，我的分数，我的信息，这四个功能模块。学生用例图如图3-2所示。

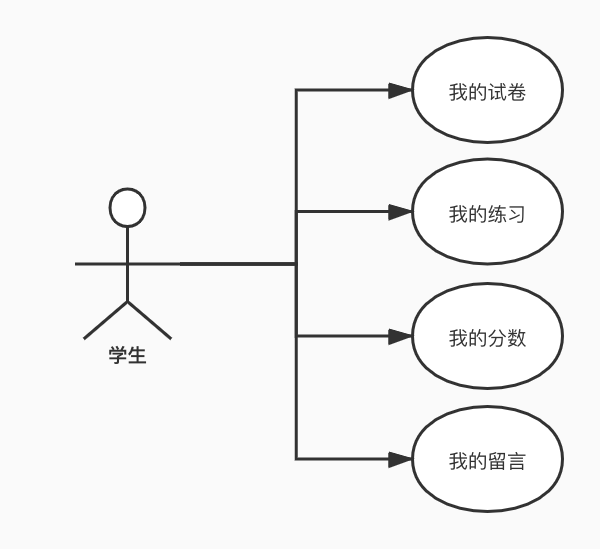


图3-3 学生用例图

3.3 非功能性需求

基于B/S架构的在线考试系统通过浏览器（Browser）和服务器（Server）交互的方式实现，在登录、管理、考试这些整体功能的界面应当遵循界面简洁、操作简单的原则，避免繁琐问题给用户增加工作量等，具体的要求包括：

(1) 界面要求: 界面是人加交互的界面，根据大多数Web应用的操作模式进行UI布局设计，拥有良好的UI界面是吸引用户最直接的因素。

(2) 性能需求：若在本地浏览器中对静态资源做好缓存会大大减少用户在加载页面时的等待时间，对网站页面进行优化也是本系统需要做的。

(3) 操作简单：本小程序目标群体为广大家长学生，所以小程序应当尽可能简单，直观并且易于学习。

(4) 多用户考核：本系统是目标用户之一是考生，系统要尽可能满足考生同时进行在线考试的并发量要求。

(5) 运行稳定：在线考试系统是教育考核的一种新方式，在考核的过程中必须保证应用的稳定性，避免发生崩溃导致考生需要重复考等故障。

# 4 系统详细设计

4.1 设计原则

（1）设计研究计划，利用现有资源完成开发。

（2）遵守软件工程开发的基本原则和方法。

（3）系统的设计应便于后续管理员的维护。

（4）系统的设计应当方便对数据库后续的维护，进而能够实现不同用户对数据库请求的处理。

（5）系统应当利于系统功能方面进行维护和更改。

(6）系统在进行设计时应该同在线考试行业发展相适应。

（7）系统的设计中关于数据库设计应满足设计的相关规范。

4.2 设计模式

**4.2.2总体结构设计**

本在线考试系统是基于B/S架构实现的，整体结构有三部分组成Web服务器（Nginx）、系统管理功能、后台数据库三大结构。总体架构如图4-1所示

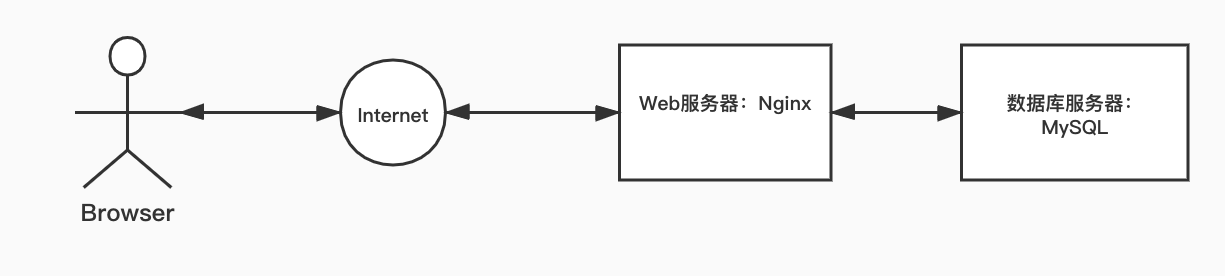


图4-1 系统架构图

**4.2.3总体功能模块图**

系统的使用是通过浏览器访问Web服务器的，系统整体设计分为三个角色，分别为系统管理员、教师、学生。系统管理员用户和教师用户共有题库管理、考试管理、成绩管理、学生管理这四大功能模块。系统管理员用户还具有教师管理功能。学生用户功能模块有：我的试卷、我的练习、我的分数、我的留言。系统结构图如图4-2所示。

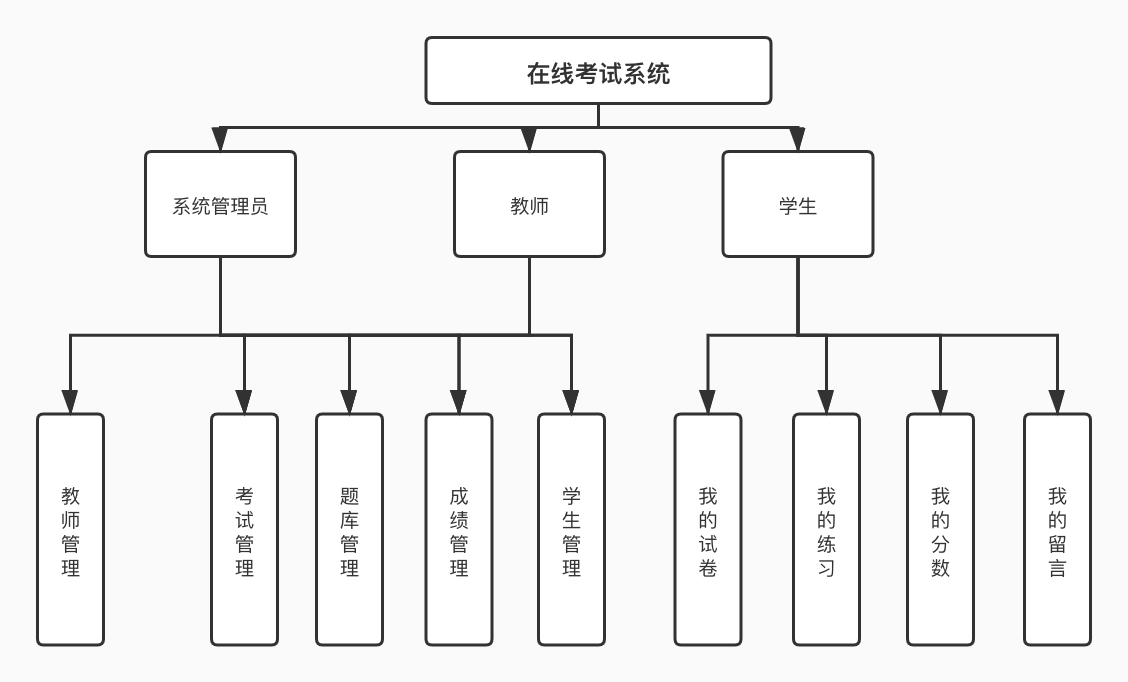


图4-2 系统功能结构图

4.3 数据库设计

**4.3.1数据库表设计**

根据对E-R图分析和系统的数据处理方面，为了存储各种数据，数据库里建立了以下几张表。

（1）管理员信息表

表4-3 admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | adminId | int |  | ID号 |
| 2 | adminName | varchar | 20 | 姓名 |
| 3 | sex | varchar | 2 | 性别 |
| 4 | tel | varchar | 11 | 电话号码 |
| 5 | email | varchar | 20 | 电子邮箱 |
| 6 | pwd | varchar | 16 | 密码 |
| 7 | cardId | varchar | 18 | 身份证号码 |
| 8 | role | varchar | 1 | 角色：0管理员1教师2学生 |

（2）考试管理表

表4-4 exam\_manage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | examCode | int |  | 考试编号 |
| 2 | description | varchar | 50 | 该次考试介绍 |
| 3 | source | varchar | 20 | 课程名称 |
| 4 | paperId | int |  | 试卷编号 |
| 5 | examDate | varchar | 10 | 考试日期 |
| 6 | totalTime | Int |  | 持续时间 |
| 7 | grade | varchar | 10 | 年级 |
| 8 | term | varchar | 10 | 学期 |
| 9 | major | varchar | 20 | 专业 |
| 10 | institute | varchar | 20 | 学院 |
| 11 | totalScore | int |  | 总分 |
| 12 | type | varchar | 255 | 考试类型 |
| 13 | tips | varchar | 255 | 考生须知 |

（3）填空题题库表

表4-5 fill\_question

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | questionId | int |  | 试题编号 |
| 2 | subject | varchar | 20 | 考试科目 |
| 3 | question | varchar | 255 | 试卷内容 |
| 4 | answer | varchar | 255 | 正确答案 |
| 5 | analysis | varchar | 255 | 题目解析 |
| 6 | score | int default 2 | 2 | 分数 |
| 7 | level | varchar | 5 | 难度等级 |
| 8 | section | varchar | 20 | 所属章节 |

（4）判断题题库表

表4-6 judge\_question

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | questionId | int |  | 试题编号 |
| 2 | subject | varchar | 20 | 考试科目 |
| 3 | question | varchar | 255 | 试卷内容 |
| 4 | answer | varchar | 255 | 正确答案 |
| 5 | analysis | varchar | 255 | 题目解析 |
| 6 | score | int default 2 | 2 | 分数 |
| 7 | level | varchar | 1 | 难度等级 |
| 8 | section | varchar | 20 | 所属章节 |

（5）留言表

表4-7 message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | id | int |  | 留言编号 |
| 2 | title | varchar | 20 | 标题 |
| 3 | content | varchar | 255 | 留言内容 |
| 4 | time | varchar | date | 留言时间 |

（6）选择题题库表

表4-8 multi\_question

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | questionId | int |  | 试题编号 |
| 2 | subject | varchar | 20 | 考试科目 |
| 3 | question | varchar | 255 | 问题题目 |
| 4 | answerA | varchar | 255 | 选项A |
| 5 | answerB | varchar | 255 | 选项B |
| 6 | answerC | varchar | 255 | 选项C |
| 7 | answerD | varchar | 255 | 选项D |
| 8 | rightAnswer | varchar | 10 | 正确答案 |
| 9 | analysis | varchar | 255 | 题目解析 |
| 10 | score | int default 2 | 2 | 分数 |
| 11 | section | varchar | 20 | 所属章节 |
| 12 | level | varchar | 1 | 难度等级 |

（7）试卷管理表

表4-9 paper\_manage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | paperId | int |  | 试卷编号 |
| 2 | questionType | int |  | 题目类型 |
| 3 | questionId | int |  | 题目编号 |

（8）回复表

表4-10 replay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | messageId | int |  | 留言编号 |
| 2 | replayId | int |  | 回复编号 |
| 3 | replay | varchar | 255 | 内容 |
| 4 | replayTime | date |  | 回复时间 |

（9）成绩管理表

表4-11 score

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | scoreId | int |  | 分数编号 |
| 2 | examCode | int |  | 考试编号 |
| 3 | studentId | int |  | 学号 |
| 4 | subject | varchar | 20 | 课程名称 |
| 5 | ptScore | int |  | 平时成绩 |
| 6 | etscore | int |  | 期末成绩 |
| 7 | score | int |  | 总成绩 |
| 8 | answerDate | varchar | 10 | 答题成绩 |

（10）学生信息表

表4-12 student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | studentId | int |  | ID |
| 2 | studentName | varchar | 20 | 姓名 |
| 3 | grade | varchar | 4 | 年级 |
| 4 | major | varchar | 20 | 专业 |
| 5 | clazz | varchar | 10 | 班级 |
| 6 | institute | varchar | 30 | 学院 |
| 7 | tel | varchar | 11 | 电话号码 |
| 8 | email | varchar | 30 | 电子邮件 |
| 9 | pwd | varchar | 16 | 密码 |
| 10 | cardId | varchar | 18 | 身份证号码 |
| 11 | sex | varchar | 2 | 性别 |
| 12 | role | varchar | 1 | 角色0管理员1教师2学生 |

（11）教师信息表

表4-13 teacher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 说明 |
| 1 | teacherId | int |  | ID |
| 2 | teacherName | varchar | 20 | 姓名 |
| 3 | institute | varchar | 20 | 学院 |
| 4 | sex | varchar | 2 | 性别 |
| 5 | tel | varchar | 11 | 电话号码 |
| 6 | email | varchar | 20 | 邮箱 |
| 7 | pwd | varchar | 16 | 密码 |
| 8 | cardId | varchar | 18 | 身份证号码 |
| 9 | type | varchar | 20 | 职称 |
| 10 | role | varchar | 1 | 角色0管理员1教师2学生 |

# 5 系统实现

**5.1 用户登录模块**

在进入系统前，需要用户通过后台已注册好的账号进行登录。如果没有账号可联系系统管理员或教师在后台管理子系统进行注册。

前台使用Axios实现AJAX异步通信，发起异步请求，核心代码如下：

this.$axios({

url: `/api/login`,

method: 'post',

data: {

...this.formLabelAlign

}

}).then(res=>{

let resData = res.data.data

//根据resData.role 来决定跳转的逻辑

})

后台使用SpringBoot、Spring Data JPA等技术实现，返回给前端JSON格式的数据。核心代码如下：

@PostMapping("/login")

public ApiResult login(@RequestBody Login login) {

// service业务层查询结果

return ApiResultHandler.buildApiResult(‘’,”msg”,res);//封装查询结果

}

**5.2 前台考试子系统**

1. 用户登录

如5.1所述，当学生用户登录后，进入到前台考试子系统中，如图5-2-1所示：

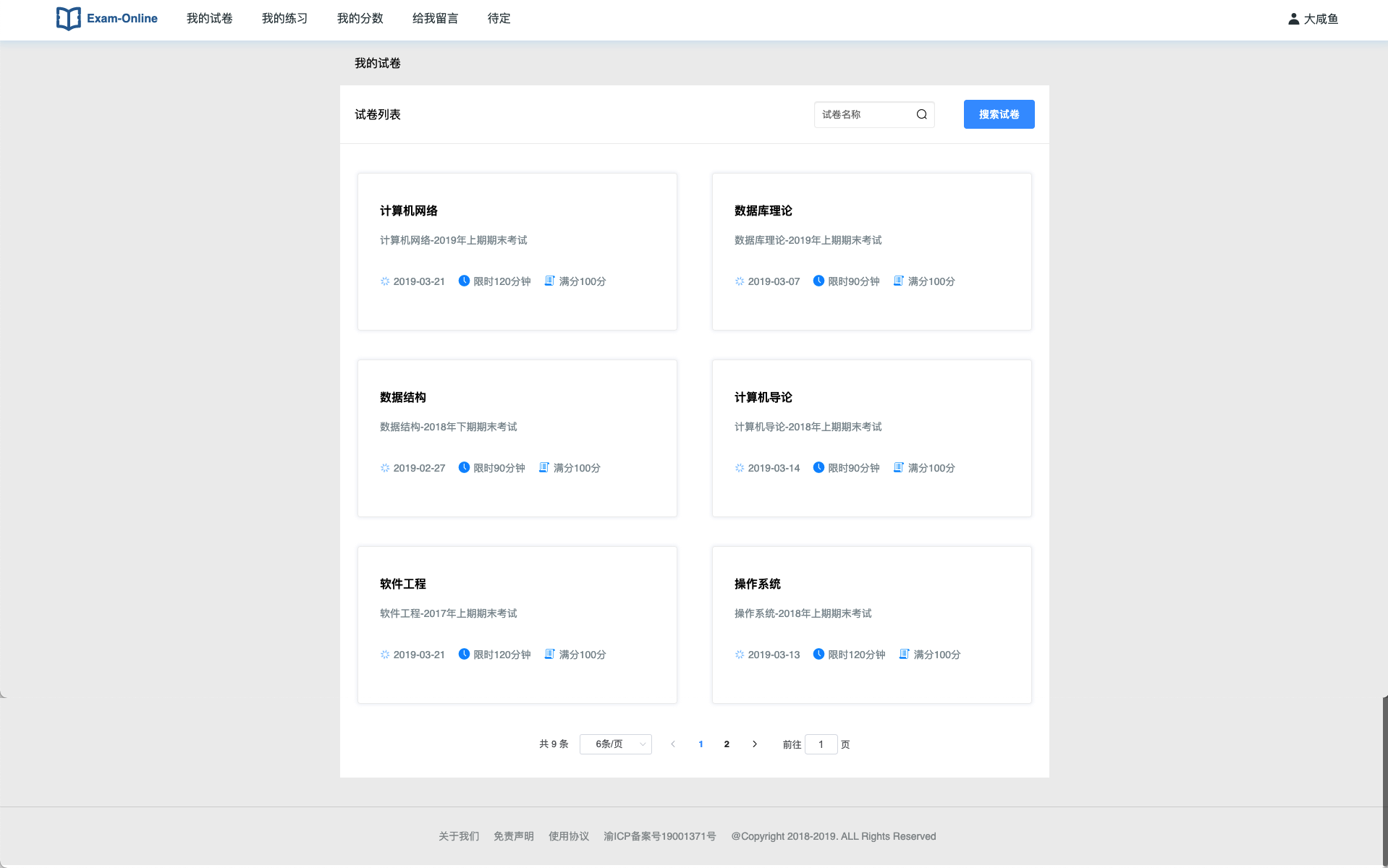


图5-2-1 前台考试子系统

可以看到考试前台相关功能：试卷列表、我的练习、我的分数、给我留言。点击右上角的“退出”，即可退出登录

1. 试卷列表模块

刚进入系统或点击“试卷列表”可以查看到当前所有的考试信息，同时支持分页查询和按试卷名称模糊查询考试信息。在点击某个试卷后跳转到“试卷详情页”。如图5-2-2所示：



图5-2-2 试卷列表页面

核心实现代码：

//获取当前所有考试信息

getExamInfo() {

this.$axios(`/api/exams/${this.pageNameber}/${this.pageSize}`).then(res => {

//...

}).catch(error => {

console.log(error)

})

},

1. 答题模块

在我的试卷页面中，点击任意一个试卷跳转到“试卷详情列表”，能够看到考试题型、分值、考生须知、总分等试卷详情。

当点击“开始答题”跳转到“答题页面”，如图5-2-3-1所示：



图5-2-3-1 考试答题页面

当考试时间结束或考生主动点击结束考试按钮后，跳转到“成绩页面”，如图5-2-3-2所示。由于试题答案信息直接存放到数据库中，所以结束考试后可以实现即时判卷、即时出成绩。

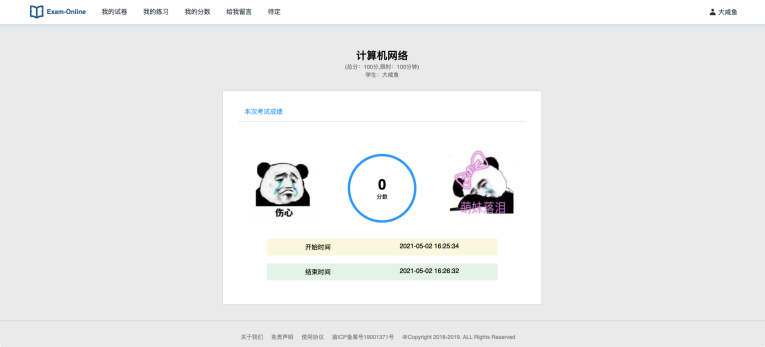


图5-2-3-2 考试成绩页面

核心实现代码：

//答案提交计算分数

commit() {

/\* 计算选择题总分 \*/

/\* 计算判断题总分 \*/

/\* 计算判断题总分 \*/

if(this.time != 0) {

this.$confirm("考试结束时间未到,是否提前交卷","友情提示",{

confirmButtonText: '立即交卷',

cancelButtonText: '再检查一下',

type: 'warning'

}).then(() => {

/\* 交卷 \*/

}).then(res => {

/\* 交卷成功后将成绩写入成绩表 \*/

})

}).catch(() => {

console.log("继续答题")

})

}

},

1. 练习模块

在前台考试子系统中，点击我的练习跳转到“试卷列表页面”。点击任意一个试卷跳转到“试卷详情列表”，能够看到此试卷的详细信息。

当点击“开始答题”，跳转到“答题页面”。与考试时的答题有所区别，练习模式答题中能看到正确答案提示、题目解析，如图5-2-4所示。练习模式交卷后的成绩不会写入到成绩表中。



图5-2-4 练习答题页面

1. 我的分数模块

点击 “我的成绩”，跳转到 “成绩查看页面”，显示当前考生的考试成绩信息: 考试日期、课程名称、考试成绩、是否通过领域。如图5-2-5所示：



图5-2-5 分数查看页面

核心实现代码：

// 获取当前学生的成绩表

getScore() {

let studentId = this.$cookies.get("cid")

this.$axios(`/api/score/${studentId}`).then(res => {

//...

}

})

}

1. 我的留言模块

点击“我的留言”跳转到“留言查看页面”，显示系统中的所有留言信息，用户可以提交留言、回复留言。如图5-2-6所示：



图5-2-6 留言页面

核心实现代码：

// 获取所有留言信息

getMsg() {

this.$axios(`/api/messages/${this.pageNumber}/${this.pageSize}`).then(res => {

//...

})

},

**5.3 后台管理子系统**

1. 用户登录

如5.1所述，当教师或管理员用户登录后，进入到后台管理子系统中，如图5-3-1所示：



图5-3-1 后台考试子系统

可以看到后台管理子系统的相关功能：考试管理、题库管理、成绩查询、学生管理、教师管理点击右上角的“退出”，即可退出登录。

1. 考试管理模块

点击“考试管理”进入到考试管理模块。在此模块中系统管理员或教师用户可以进行考试查询、添加考试操作。在点击考试查询后跳转到“考试列表页面”，在该页面中具有分页功能，还能对具体考试进行编辑、删除操作。在点击添加考试后跳转到“考试添加页面”，在该页面中填写考试相关信息，继续点击“创建考试”按钮保存考试信息。

老师发布了考试,学生才可以在主页面看到相应的考试信息。 有考试安排表以后,才能给该次考试添加题目,对应数据表是exam\_manage。 该表保存该次考试,课程名称,考试时间,所属专业,学院等等信息。

核心实现代码：

//分页查询所有试卷信息

getExamInfo() {

this.$axios(`/api/exams/${this.pageNumber}/${this.pageSize}`).then(res => {

//...

}).catch(error => {

})

},

1. 题库管理模块

点击“题库管理”进入到题库管理模块。在此模块中系统管理员或教师用户可以进行题库查询、增加题库操作。在点击题库查询后跳转到“题库列表页面”，在该页面中分页显示所有题目信息。在点击增加题库后跳转到“试卷列表页面”，在该页面中分页显示所有试卷信息，同时具有对具体试卷增加试题的功能。

题库表设计和普通数据表设计有所区别。 分为了三张表,分别是选择题题库表,填空题题库表,判断题题库表, 每个表保存相应类型的题库,通过一张中间表,将题库和试题关联起来。 这样就组成了一张完整的试卷。

核心实现代码：

//分页查询所有试卷信息

getAnswerInfo() {

this.$axios(`/api/answers/${this.pageNumber}/${this.pageSize}`

).then(res => {

//...

}).catch(error => {});

},

1. 成绩查询模块

点击“成绩查询”即可进入成绩查询模块。在此模块中可以进行学生成绩查询、考试成绩分段查询。点击学生成绩查询后分页显示学生用户，再点击查看成绩按钮，显示出该学生用户成绩的折线图。点击成绩分段查询后分页显示考试信息，再点击查看分段，显示出该考试的分数段饼图。

核心实现代码：

score() {

let studentId = this.$route.query.studentId

this.$axios(`/api/score/${studentId}`).then(res => {

//根据学生Id查询成绩

//...

})

}

1. 学生管理模块

点击“学生管理”，进入学生管理模块。在此模块中可以对学生用户进行增删改差的操作，对应数据库中的student表。

核心实现代码：

//分页查询所有学生信息

getTeacherInfo() {

this.$axios(`/api/students/${this.pageNumber}/${this.pageSize}`).then(res => {

//...

}).catch(error => {});

}

1. 教师管理模块

点击“教师管理”，进入教师管理模块。在此模块中可以对教师用户进行增删改差的操作，对应数据库中的teacher表。

核心实现代码：

//分页查询所有教师信息

getTeacherInfo() {

this.$axios(`/api/teachers/${this.pageNumber}/${this.pageSize}`).then(res => {

//...

}).catch(error => {});

}

# **6 系统的测试**

**6.1 测试方法**

(1) 本系统采用黑盒法进行系统测试。

(2) 黑盒法就是将该改项目看成一个黑盒子，完全不考虑其内在构造。只是考虑功能是否与用户需求相适应，程序的输入，输出数据是否完全正确，并且整个系统必须保证外部结构完整。

**6.2 黑盒测试用例**

本次系统进行了很多黑盒案例测试，对于这些测试用例，使用了特别严格的测试标准，以保证系统可以最大程度上满足用户的需求。由于篇幅限制，在此仅展示部分测试用例。

（1）系统登录模块的测试用例和测试过程如表6-1所示

表6-1 用户登录操作测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试步骤 | 操作或数据输入 | 显示或数据输出 | 实际输出结果 |
| ST-1 | 登录名为空，点击“确定” | 显示“登录名不能为空”警告  框 | 显示“登录名不能为空”警告框 |
| ST-2 | 输入“登陆名称”，并输入错误的“密码” | 显示“密码错误”警告框 | 显示“密码错误”警告框 |
| ST-3 | 输入正确的“登陆名称及密码” | 进入系统主界面 | 进入系统主界面 |

（2）系统其他主要功能模块的测试用例和测试程序如表6-2所示。

表6-2 主要功能操作测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 操作或数据输入 | 显示或数据输出 | 实际输出结果 |
| ST-1 | 教师发布考试信息 | 考试新添加成功 | 显示添加成功的信息 |
| ST-2 | 学生进入答题模块 | 显示题目 | 显示题目 |
| ST-3 | 结束考试 | 考试成绩 | 提交成功，刷新当前页面显示当前成绩信息 |

经过多次测试和改进，原本拟定的各个系统的功能基本实现，基本完成在线考试系统的各项功能需求。由于该系统仍处于测试阶段，仍有许多不足之处。后期投入使用后，提出的修改建议结合用户试用后的反馈进行改进。

# **7 结束语**

**7.1 总结**

本文从在线考试系统行业的背景和发展现状的研究中揭示了开发该系统的意义。接着介绍了本次系统设计与实现中运用到的相关变成软件和架构，第三章对系统的可行性、需求进行了详细的分析。第四章起系统的详细设计，第五章编码实现到最后第六章系统测试，详细地描述了本系统的实现过程。

本人通过对考试系统的设计与实现，掌握了运用Vue.js开发网站的具体流程与设计技巧，对开发工具有了进一步认识，并在设计与实现过程中学习到了如何使用云服务器与调用微信接口等一系列新技术新知识。在开发中对网站开发中使用到了各种web技术，如vue，Spring Boot，MySQL等，加深了我对web开发的熟练程度，提升了我对系统机制的认识高度，积累了大量开发经验，提高了对程序Bug的修复能力。同时，针对代码优化与实现特殊效果，本系统采用了目前了比较流行的web开发框架与开源插件来实现，通过对这些插件的使用与了解，加深了我对web开发的认识，提高了开发效率，精简了代码，提高了软件可维护性。

本系统使用简单，页面美观大方，但由于本人能力尚且不足，目前该程序还存在着一些缺陷，主要表现在以下三个方面：

1. 在数据库设计方面还存在着一定问题，很多的字段属性定义不清，并且没有考虑每个字段的具体范围，因此在后期数据库处理上产生了一定影响，并且数据库的优化上做的还不是很好。
2. 系统设计虽然满足了自己的预期目标，但是还有很多没有考虑到的地方，一些功能仍然不是很完善，有时会无法满足用户的相关需求。

（3）在线考试用户数受系统架构及服务器性能限制，无法支撑大规模的同时在线人数。但目前本人能力尚且不能解决这个问题，等到以后技术更加全面成熟一定妥善解决这个问题。

**7.2 展望**

鉴于目前在线考试行业发展的迅猛姿态，就着先入为主的理念，我想尽快将本在线考试系统平台上线部署并开始运营，之后进行更好的宣传和推广。假以时日，相信可以取得客观的经济效益。

**参考文献**

[1]龚华堂.实现一种基于SSM框架的考试系统研究[J].电脑知识与技术,2021,17(04):59-61.

[2]熊志勇.基于在线考试系统的PHP课程教学探索和应用[J].电脑知识与技术,2021,17(04):150-151.

[3]董晓睿,邸文华,崔健,商凯.基于Spring Boot的在线考试系统的设计与开发[J].电脑编程技巧与维护,2020(10):36-38.

[4]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.

[5]张春雨. 大规模在线考试考务综合管理系统的设计与实现[D].哈尔滨工业大学,2020.

[6]张杰. 基于React+Spring的教学系统设计与实现[D].山东师范大学,2019.

[7]叶惠仙,沈文杰.基于Bootstrap+spring boot框架的在线考试系统开发[J].网络安全技术与应用,2019(12):54-57.

[8]赵雪辉,王金峰,于善彬,董普祥,张雪玉.基于SSM框架的在线考试系统的开发与设计[J].现代信息科技,2019,3(07):82-84.