

专业综合实训

web

考试报名系统

学院：信息与控制工程学院

姓名：赵帅

学 号：201722240

专业班级：软件 173

任务书

1.设计目的

线上考试报名系统是社会人员以及学生参加大型考试的主要报名途径。通过设计此系统可以简化用户操作，提高报名效率并且增加系统的稳健型，便利用户操作，避免系统运行高峰时出现崩溃的情况。

2.设计内容及要求

2.1 设计内容

设计内容包含三部分：系统前端界面设计、系统后端数据操作、数据库设计与数据存储。

2.1.1 系统前端界面设计

系统前端界面包括三个部分：用户登陆界面，考生界面，管理员界面。

用户登陆界面。用户输入账号和密码进行登陆，若为新用户，则点击链接“注册新用户”，注册新账号，再进行登陆。由单选框选择为考生登陆或者管理员登陆，从而登录进入不同界面。

考生界面。采用“上-左右”的页面布局。上侧为系统介绍，左侧为操作按钮，通过点击不同的按钮，进行不同的操作，对应的操作界面会显示再右侧页面。左侧为侧栏垂直菜单并嵌套子菜单，分别为“个人信息”，“考试信息”，“财务信息”。在个人信息部分，可以查看个人信息以及修改登录密码。在考试信息部分，可以注册考试，并查看已经注册考试以及考试成绩。在财务信息部分，可以进行支付和退款操作。

管理员界面。同样采用“上-左右”的页面布局。通过点击左侧按钮，对应右侧的操作界面。左侧为侧栏垂直菜单并嵌套子菜单，分别为“管理学生信息”，“管理考试信息”，“管理考生考试信息”，“申请消息”，“密码修改”。可以依次对学生，考试，考生考试关联关系，申请消息进行增删改查操作，并可以修改登陆密码。

2.1.2 系统后端数据操作

本系统采用 Spring Boot 框架进行构建，从而使系统可以简化配置，进一步简化了 Spring 应用的整个搭建和开发过程。采用 Spring Data JPA 技术，进行数据库相关操作，从而大部分简单的 CRUD 操作可以直接生成，简化了开发过程。其也保留了一些灵活性，从而开发者可以在遵循一定规则的情况下，按照需求编写查询方法。

2.1.3 数据库设计与数据存储

本系统采用 MySQL 数据库设计。并在其中设置五张数据表。admin 表用来存储管理员信息，apply 表用来存储申请信息，exam 表用来存储考试相关信息，student 表用来存储考试信息，student_exam 表用来关联考试和考生信息。

2.2 设计要求

(1)准确性。要确保考试信息与用户信息的相关准确性，以及对用户申请信息的操作准确性。

(2)易操作性。本系统要以方便用户使用为准，制作操作简单，易使用的系统。并且在系统中给定足够的操作提示。

(3)安全性。在用户使用此系统的同时，要确保用户的数据安全以及数据库中数据的安全性。

(4)低成本。本系统在保证系统的安全性和易操作性的前提下，尽量使系统的开发、使用和后续维护成本降到最低。

3.语言与技术

本系统采用 Java 语言编写。采用 SpringBoot 框架以及 Spring Data JPA 技术设计后端系统。采用 Vue 框架和 elementUI 技术设计前端系统。

4.项目分工

图 4 分工表示图

1	系统前端界面设计	赵帅
2	系统后端设计	赵帅
3	数据库设计	赵帅

目录

1 可行性分析-----	4
2 需求分析-----	4
2.1 项目背景-----	4
2.2 系统整体描述-----	4
2.3 功能需求-----	5
2.4 性能需求-----	6
3 系统设计-----	6
3.1 系统设计目标-----	6
3.2 系统功能模块划分-----	7
3.3 系统的实体联系关系-----	8
3.4 数据流图-----	9
3.5 数据字典-----	9
3.6 对系统进行面向对象的程序设计-----	11
4 实现与测试-----	11
5 结论与展望-----	17
6 主要参考文献-----	17
7 附录-----	18

1.可行性分析

(1) 技术可行性:

本系统采用 Java 语言编写。采用 Spring Boot 框架以及 Spring Data JPA 技术设计后端系统。采用 Vue 框架和 ElementUI 技术设计前端系统。使用 MySQL 数据库进行信息存储。到目前为止，这些技术已经十分成熟，系统开发技术可行。

(2) 经济可行性:

在对系统的开发、使用和后续维护的过程中，只需要在购买一些正版软件和教材的费用，用于对软件的开发和维护，并不需要额外费用。

(3) 操作可行性:

在软件开发时，可使用 IntelliJ IDEA、WebStorm 和 MySQL 软件进行技术开发，并设计简单的前端界面提示以进行功能测试。具有较强的操作可行性。

2.需求分析

2.1 项目背景

线上考试报名系统是社会人员以及学生参加大型考试的主要报名途径。通过设计此系统可以简化用户操作，提高报名效率并且增加系统的稳健型，便利用户操作，避免系统运行高峰时出现崩溃的情况。

2.2 系统整体概述

考试报名系统，可以为用户提供有效且快捷的报名方法。由此可以简化公共大型考试的组织过程，从而提高组织过程的效率与准确性。

此系统采用 Spring Boot 框架以及 Spring Data JPA 技术设计后端系统，由此完成后端数据的增删查改操作。采用 Vue 框架和 ElementUI 技术设计前端系统，通过与后端系统的交互可分别完成考生对“个人信息”、“考试信息”、“财务信息”的相关操作。同时也设置了管理员界面，完成对学生，考试，考生考试关联关系，申请消息进行增删改查操作，并可以修改登陆密码。设计 MySQL 数据库，分别存储考试、考生、申请消息、关联信息和管理员信息的信息。

2.3 功能需求

2.3.1 登录功能

- (1) 为用户提供考生、管理员的登录选择，并通过登录进入操作界面。
- (2) 为新用户提供注册方法，新用户首先需要注册，进而登录系统。

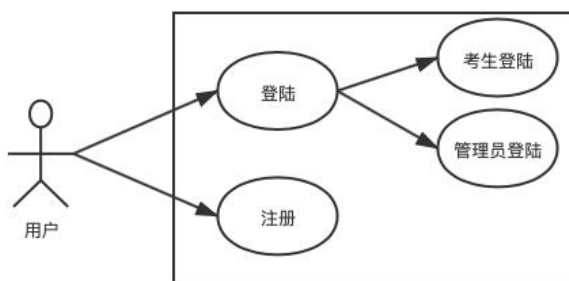


图 1 用户用例图

2.3.2 考生功能

- (1) 个人信息。为考生提供查看个人信息，修改登陆密码的功能。
- (2) 财务信息。为考生提供充值和退款申请的功能。
- (3) 考试信息。为考试提供注册考试的功能。考生可以同时查看已注册考试，并且取消已注册考试。可以查看已进行完的考试，申请成绩复议。

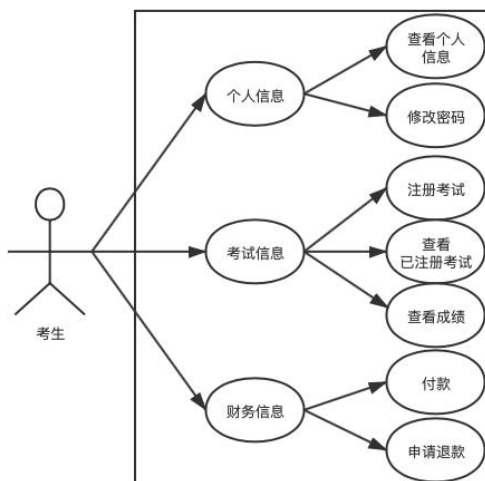


图 2 考生用例图

2.3.3 管理员功能

- (1) 为管理员提供对学生信息的 CRUD 操作的功能。
- (2) 为管理员提供对考试信息的 CRUD 操作的功能。
- (3) 为管理员提供对申请信息的查看和执行操作的功能。
- (4) 为管理员提供对考试和考生关联信息的 CRUD 操作的功能。
- (5) 为管理员提供修改登陆密码的功能。

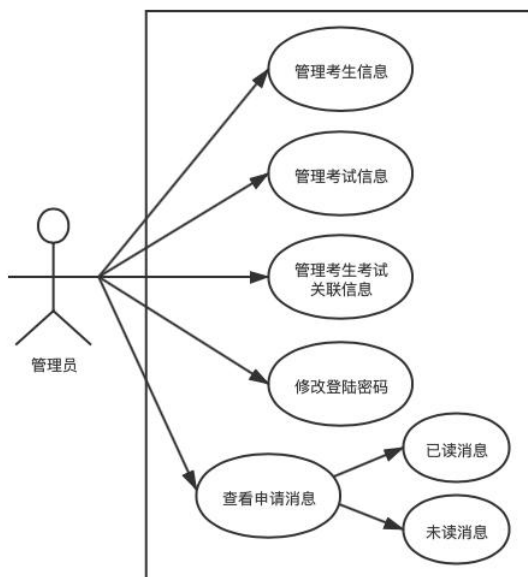


图 3 管理员用例图

2.4 性能需求

考试报名系统对于性能需求较高。主要性能需求在于能够在大量用户同时操作时，保持系统的稳定快速。在确保稳定的同时，也要保证操作信息的准确性，使用户可以快速，准确地预定考试，并查询考试结果。

3.系统设计

3.1 系统设计目标

基于通用的 SpringBoot+Vue 框架开发此项目，开发成本低，稳定的考试报名系统。主要目标为能够方便、准确地使用户注册考试，并及时查看考试结果。考试用户可以向系统发送申请消息，并由系统管理员具体完成申请要求。

3.2 系统功能模块划分

系统中功能划分为三部分。第一部分，为用户提供登陆操作。第二部分，考生登陆后进行个人信息处理，考试注册等操作，同时也可以进行充值或者申请退款。第三部分，系统管理员针对数据库内存储的考试、考生、关联信息、申请信息进行 CRUD 操作。

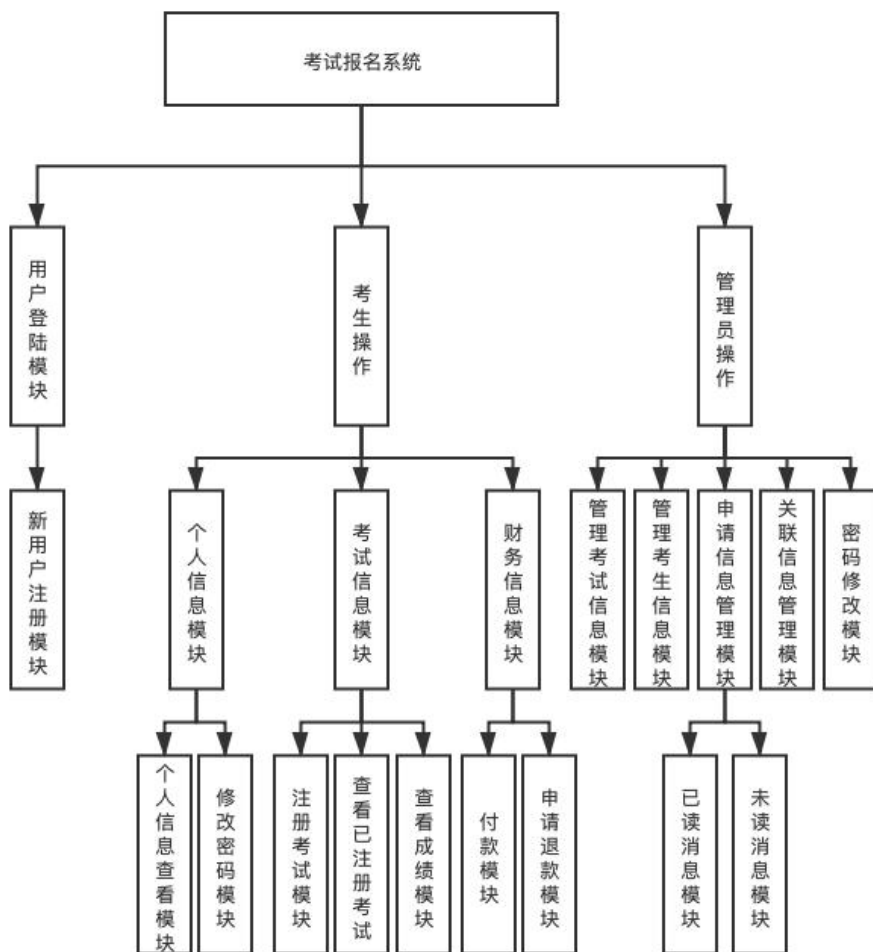


图 4 功能结构图

3.3 系统的实体联系关系

实体联系中，一个考生可以注册多个考试，同时一个考试也可以对应多个考生，从而形成多对多关系，多对多关系由 student_exam 表进行关联表示。一个考生可以发送多个申请信息，从而形成一对多关系。一个管理员可以对信息操作，所有管理员对考生、考试、关联表、申请信息四个实体构成一对多关系。

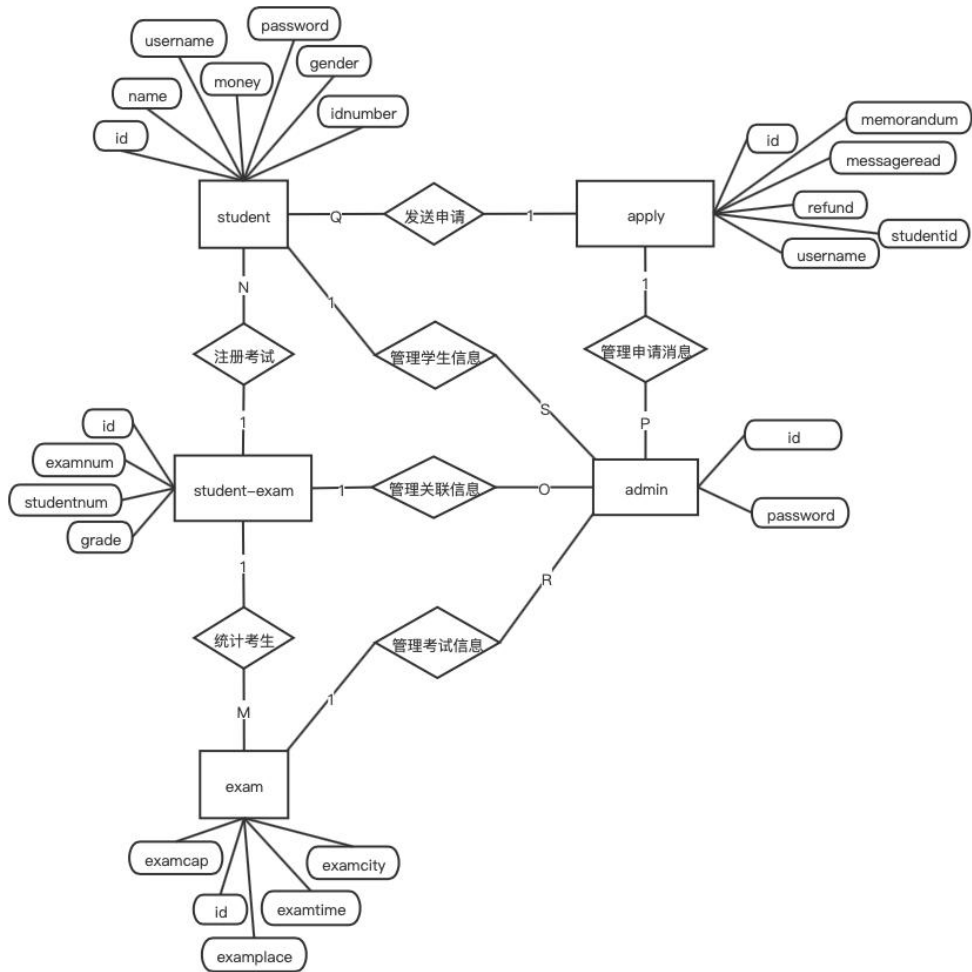


图 5 实体联系图

3.4 数据流图

考试报名系统中数据的流动为考生通过注册账号，充值，注册考试等操作，在表格中生成相关记录。系统管理员可以操作生成的表中的记录，并且可以对考试的相关信息数据进行增删改查操作。

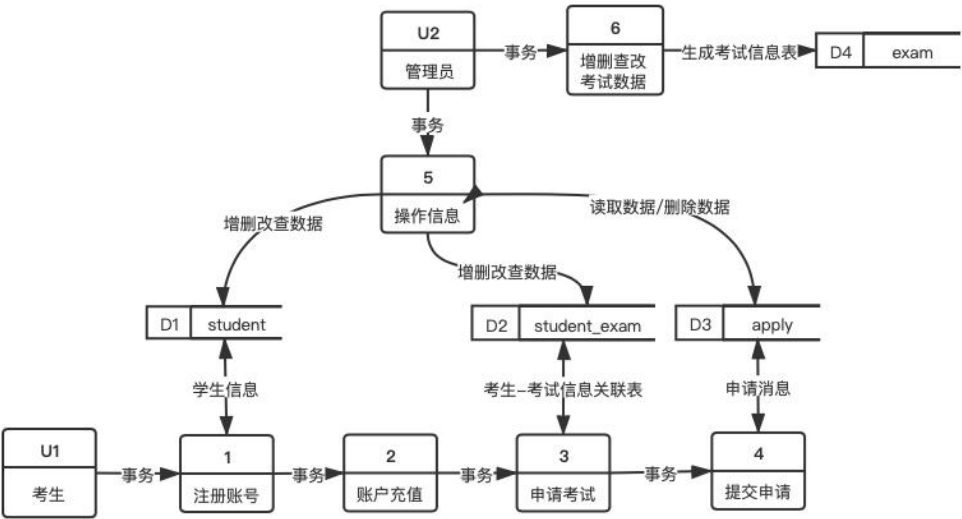


图 6 数据流图

3.5 数据字典

数据字典包含五个数据表，分别用于存储考生、考试、申请消息、关联关系和管理员信息。

表 1 student 表

名称	字符类型	主键	能否为空	备注
id	INT	是	否	考生编号
gender	VARCHAR	否	能	考生性别
idnumber	VARCHAR	否	能	考生身份证号
money	DOUBLE	否	能	考生账户余额
name	VARCHAR	否	能	考生姓名
password	VARCHAR	否	能	考生账户密码
username	VARCHAR	否	能	考生账号用户名

表 2 exam 表

名称	字符类型	主键	能否为空	备注
id	INT	是	否	考试编号
examcap	INT	否	能	考位容量
examplace	VARCHAR	否	能	考试地点
examtime	VARCHAR	否	能	考试时间
examcity	VARCHAR	否	能	考试城市

表 3 student_exam 表

名称	字符类型	主键	能否为空	备注
id	INT	是	否	关联信息编号
studentnum	INT	否	能	考生编号
examnum	INT	否	能	考试编号
grade	INT	否	能	考试成绩

表 4 apply 表

名称	字符类型	主键	能否为空	备注
id	INT	是	否	申请消息编号
messageread	INT	否	能	判断是否已读
memorandum	VARCHAR	否	能	申请消息编号备注
refund	DOUBLE	否	能	操作退款/考试编号
username	VARCHAR	否	能	考生用户名
studentid	INT	否	能	考生编号

表 5 admin 表

名称	字符类型	主键	能否为空	备注
id	INT	是	否	管理员账号用户名
password	VARCHAR	否	能	管理员账号密码

3.6 对系统进行面向对象的程序设计

以下为后端项目文件结构：

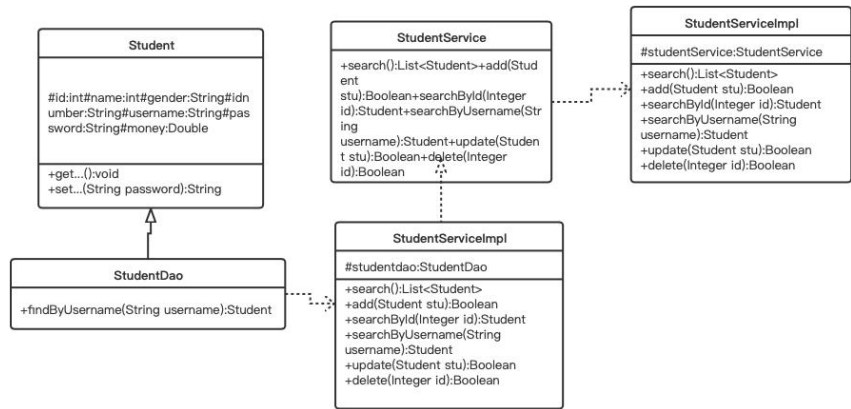


图 7 学生 UML 类图

4.实现与测试

4.1 登陆界面

(1) 登陆

输入账号和密码，通过点击单选框确定登陆界面为考试界面或者管理员操作界面。

账号

密码

请选择: ☐ 学生登陆 ☐ 管理员登陆 [注册新用户](#)

图 8 登陆界面图

(2) 注册新用户

姓名

性别 ☐ 男 ☐ 女

身份证号码

用户名

密码

图 9 注册新用户界面图

4.2 考生界面

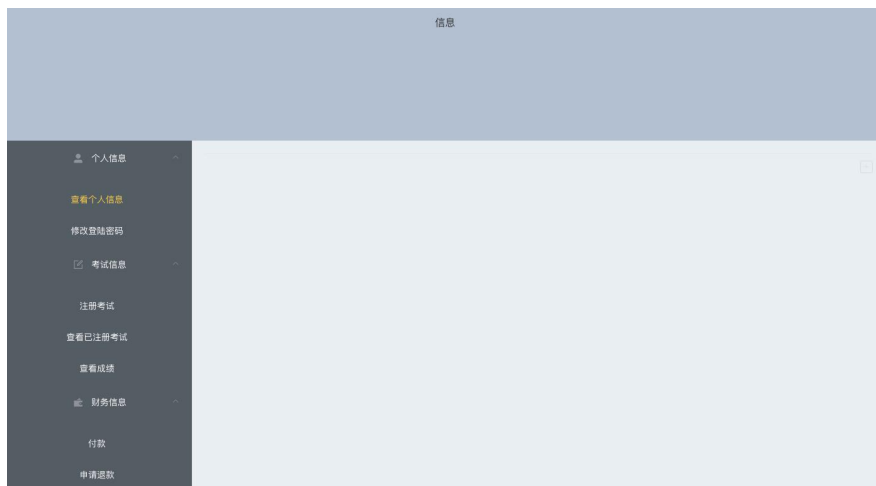


图 10 考生操作界面图

(1) 个人信息

学生信息 ×					
姓名	性别	身份证号码	账号余额	用户名	密码
赵帅	男	1231433242	600	123456	123

图 11 查看个人信息界面图

密码修改 ×	
新密码	<input type="password"/>
确认密码	<input type="password"/>

图 12 修改考生账户密码界面图

(2) 考试信息

考生通过选择地点，时间从而可以查询特定考试安排。点击表格中考试项目，即可选中，再点击申请考位，会弹出确认框。

注册考试 ×		
请选择：地点	北京	
时间	2021-01-16	
<input type="button" value="确定"/>		
请选择考位：		
考试时间	考试地点	考位
2021-01-16	北京王府学校	32
2021-01-16	北京外国语学院	47
<input type="button" value="申请考位"/>		

图 13 注册考试界面图

确认信息后，点击确认即可注册考试。

提示

姓名

赵帅

性别

男

身份证号

1231433242

考试地点

北京王府学校

考试时间

2021-01-16

考试金额

280

确定

图 14 申请考试确认界面图

查看注册考位时，可以点击取消考试从而取消预定。下图中的考试为当前日期之后的考位信息。

已注册考试

已注册考位:		
考试时间	考试地点	操作
2021-03-12	四川大学	取消考试
2021-03-12	成都理工大学	取消考试
2021-03-12	上海大学	取消考试

图 15 查看已注册考试信息界面图

查询成绩信息为当前日期之前的考试信息。成绩在考试日之后即可统计到系统中。若异议，则可申请成绩复议，等待管理员处理。

已完成考试:

考试时间	考试地点	考试成绩	操作
2020-10-20	大连理工大学	91	成绩复议

若申请成绩复议，将收取120CNY复议费用*

图 16 查看考试成绩界面图

(3) 财务信息

考生在考试前需先进行充值。在此界面中，考生可以根据参考款项为自己的账号进行充值，提供微信/支付宝付款。

支付的款项类别参考：		
款项类别	单价(人民币元)	数量
报名费	RMB¥280	1
退考费	RMB¥50	1
成绩复议费	RMB¥120	1

请输入支付金额：

请选择：

☐ 微信支付

☐ 支付宝付款

确定

图 17 账号充值界面图

考生可以在此界面中填写信息，向管理员发送退款请求，退款操作则由管理员在后台完成操作。

您当前余额为：

600

请输入您的用户名：

请输入退款金额：

备注：

提交

图 18 申请退款界面图

4.3 管理员界面

Header

管理学生信息

管理考试信息

管理考生考试信息

申请消息

密码修改

图 19 管理员操作界面图

(1)考生信息管理

在此界面管理员可以查看所有考生信息，并且通过下方的按钮进行添加、修改、删除考生信息的操作。

学生管理

学号	姓名	性别	身份证号码	余额	用户名	密码
1	赵帅	男	1231433242	600	123456	123
2	张三	男	342525234	0	1234567	123
3	找sss	男	123423545245	200	12345678	12345678
5	胡之来	男	23424234223	0	123456789	123456789
8	zs2s	女	2131414	0	12345	123

添加

修改

删除

图 20 考生信息管理界面图

填入新增考生信息，点击确认即可增加新考生。

提示

姓名

性别

☐男

☐女

身份证号

用户名

密码

确定

图 21 考生信息新增界面图

自动填入考生原始信息，在此基础上修改即可。

提示

姓名

zs2s

性别

☐男

☒女

身份证号

2131414

余额

0

用户名

12345

密码

123

确定

图 22 考生信息修改界面图

(2)考试信息管理

同学生信息管理，通过下侧三个按钮对考试信息进行增删查改管理。

考试管理				
考试编号	考试地点	考试时间	城市	考点容量
1	北京王府学校	2021-01-16	beijing	32
2	北京外国语学院	2021-01-16	beijing	47
3	河北师范大学	2021-01-16	shijiazhuang	0
4	河北外国语学院	2021-02-20	shijiazhuang	47
5	青岛理工大学	2021-02-20	qingdao	29
6	中国海洋大学	2021-02-20	qingdao	60
7	四川大学	2021-03-12	chengdu	59
8	成都理工大学	2021-03-12	chengdu	49
9	上海大学	2021-03-12	shanghai	39
10	复旦大学	2021-04-10	shanghai	50
11	深圳大学	2021-04-10	shenzhen	30
12	香港中文大学	2021-04-10	xianggang	30
添加 修改 删除				

图 23 考试信息修改界面图

(3)考生考试关联信息管理

同考生/考试信息管理操作。

编号	考生编号	考试编号	成绩
4	1	7	0
5	1	8	90
6	1	9	100
7	1	13	91
添加 修改 删除			

图 24 考试-考生关联信息修改界面图

(4)申请信息管理

申请信息分为已读消息和未读消息。

未读消息可以被删除，可以设置为已读消息。

未读消息				
信息编号	消息备注	退款金额/复议成绩编号	申请学生编号	申请学生用户名
18	成绩复议	13	1	123456
19	成绩复议	13	1	123456
设为已读 删除				

图 25 未读消息界面图

已读消息界面中，可以删除消息。

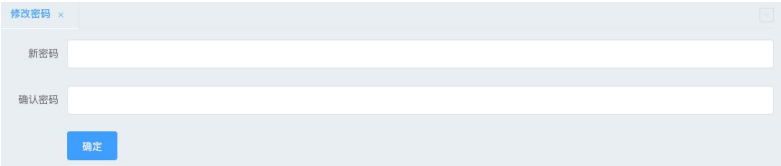


未读消息 已读消息 x				
信息编号	消息备注	退款金额/复议成绩编号	申请学生编号	申请学生用户名
19	成绩复议	13	1	123456
20	余额退款	200	1	123456
21	成绩复议	13	1	123456
22	成绩复议	13	1	123456
26	余额退款	200	1	123456
27	成绩复议	13	1	123456

删除

图 26 已读消息界面图

(5) 密码修改



修改密码 x

新密码

确认密码

确定

图 27 管理员修改密码界面图

5.结论与展望

本次开发的考试报名系统，可以较为基本实现用户登陆，考生注册、取消考试，以及充值、退款等考生相关操作，也可以完成管理员对大部分信息的管理与消息的处理等操作。基本满足线上考试报名的基本要求。系统较为稳定，操作简单、准确，用户方便使用。

通过本次对考试报名系统的开发，让我进一步了解和熟练了对 Spring Boot 框架和 Vue 框架的使用。从而快捷搭建项目，提高开发效率。

在后端系统中，采用 Spring Data JPA 技术进行数据的相关操作，从而可以自动生成部分简单 SQL 语句，并且保留了一定的灵活性，可以通过标注，自己编写所需要的 SQL 语句。前端采用 Vue+elementUI，进一步熟悉了各个界面控件的使用，熟练对前端界面设计的操作。

对于本次开发的考试报名系统，其未来的前景还是十分广阔的。现如今，各种办理过程都转移到线上，这样有助于相关活动的举办。并且，由于一般的公共活动有较多人参加，因此，需要构建稳定的系统，可供大量人群同时操作。因此，考试报名系统未来应用十分广阔，可以提高活动或考试的举办效率。

6.主要参考文献

[1]克雷格 沃斯.Spring Boot 实战.人民邮电出版社,2016-9
[2]张耀春.Vue.js 权威指南.电子工业出版社,2016-8-1

7.附录

1、后端项目

以学生操作为例进行展示

(1) **Student 实体类**

@Entity

@Table

```
public class Student {
```

```
    @Id
```

```
    @GeneratedValue(strategy= GenerationType.IDENTITY)
```

```
    private int id;
```

```
    @Column
```

```
    private String username;
```

```
    @Column
```

```
    private String password;
```

```
    @Column
```

```
    private String name;
```

```
    @Column
```

```
    private String gender;
```

```
    @Column
```

```
    private String idnumber;
```

```
    @Column
```

```
    private double money;
```

```
    .....
```

```
}
```

(2) **StudentDao**

@Repository

```
public interface StudentDao extends JpaRepository<Student, Integer> {
```

```
    Student findByUsername(String username);
```

```
}
```

(3) **StudentServiceImpl**

@Service

```
public class StudentServiceImpl implements StudentService {
```

```
    @Autowired
```

```
    StudentDao stuDao;
```

```
    @Override
```

```
    public List<Student> search() {
```

```

        return stuDao.findAll();
    }
    @Override
    public boolean add(Student student) {
        Student newStu=stuDao.save(student);
        return newStu!=null;
    }
    @Override
    public Student searchById(Integer id) {
        return stuDao.findById(id).get();
    }
    @Override
    public Student searchByUsername(String username) {
        return stuDao.findByUsername(username);
    }
    @Override
    public boolean update(Student student) {
        Student newStu=stuDao.save(student);
        return newStu!=null;
    }
    @Override
    public boolean delete(Integer id) {
        stuDao.deleteById(id);
        return true;
    }
}

```

(4) **StudentController**

```

@RestController
@RequestMapping("stu")
public class Studentcontroller {
    @Autowired
    StudentService studentService;
    @Autowired
    ExamService examService;
    //对所有学生信息的查询
    @GetMapping
    public List<Student> search() {

```

```

        List<Student> studentList = studentService.search();
        return studentList;
    }
    //以查询学号的方式查询具体学生
    @GetMapping("/studentsingle/{id}")
    public Student searchById(@PathVariable Integer id) {
        Student stu = studentService.searchById(id);
        return stu;
    }
    //以查询用户名的方式查询具体学生
    @GetMapping("/student/{username}")
    public Student searchByUsername(@PathVariable String username) {
        Student stu = studentService.searchByUsername(username);
        System.out.println(stu);
        return stu;
    }
}

//增加新生信息
@PostMapping
public String add(@RequestBody Student student) {
    boolean flag = studentService.add(student);
    if (flag) {
        return "redirect:search";
    }
    return "";
}

//修改新生信息
@PutMapping
public String update(@RequestBody Student stu) {
    if (studentService.update(stu)) {
        return "redirect:search";
    } else {
        return "update";
    }
}

//删除某个新生信息
@DeleteMapping("/{id}")
public String delete(@PathVariable Integer id) {

```

```

        if (studentService.delete(id)) {
            return "redirect:search";
        } else {
            return "";
        }
    }
}

```

2、前端项目

(1)beginInterface

```

<template>
  <div id="container">
    <el-form ref="form" label-width="120px">
      <el-form-item label="账号">
        <el-input v-model="username"></el-input>
      </el-form-item>
      <el-form-item label="密码">
        <el-input v-model="password" type="password"></el-input>
      </el-form-item>
      <el-form-item label="请选择: ">
        <el-radio-group v-model="way">
          <el-radio label="学生登陆"></el-radio>
          <el-radio label="管理员登陆"></el-radio>
          <el-link type="primary" @click="register">注册新用户</el-link>
        </el-radio-group>
      </el-form-item>
      <el-form-item>
        <el-button type="primary" @click="admin">登陆</el-button>
      </el-form-item>
    </el-form>
  </div>
</template>
<script>
  export default {
    name: "admin",
    data(){
      return {
        objects:"",

```

```

        username:"",
        password:"",
        way:""
    }
},
methods:{
    register(){
        this.$router.push({name:"registerinterface"})
    },
    admin(){
        if(this.way==="学生登陆"){
            this.axios.get('http://localhost:8090/stu')
            .then(function (res) {
                this.objects=res.data;
                for(var i in this.objects){
                    if((this.username==this.objects[i].username)
(this.objects[i].password==this.password)){
                        this.$router.push({name:"stuInterface",
requiredAuth:true});
                        params:{id:this.username},
                    }
                }
                this.username=""
                this.password=""
            }).bind(this))

            .catch(function (err) {
                if (err.response){
                    //失败
                    console.log(err.response)
                }
            }).bind(this))
        }else if(this.way==="管理员登陆"){
            this.axios.get('http://localhost:8090/admin')
            .then(function (res) {
                this.objects=res.data;
                console.log(this.objects)
                for(var i in this.objects){

```

```

        if((this.username==this.objects[i].id)                                &&
        (this.objects[i].password==this.password)){
            this.$router.push({name:"adminInterface",                      params:{id:this.username},
            requiredAuth:true});
        }
    }
    this.username=""
    this.password=""
    }.bind(this))
    .catch(function (err) {
        if (err.response){
            //失败
            console.log(err.response)
        }
    }.bind(this))
    }else {
        alert("请填写完整信息! ")
    }
}
}
}
</script>
(2)stuInterface
<el-main height="800px">
    <el-tabs    v-model="selectedTab"    type="card"    editable    @edit="handleTabsEdit"
    @tab-remove="removeTab">
        <el-tab-pane
            :key="modules[key].name"
            v-for="key in tabs"
            :label="modules[key].title"
            :name="modules[key].name"
        >
            <component :is="modules[key].component"></component>
        </el-tab-pane>
    </el-tabs>
</el-main>
return {

```



```

selectedTab: 'null',
//id:this.$route.params.data,
tabs:[, //元素和 modules 下元素的 name 相同
modules:{
  personalInfoShow:{
    title:'学生信息',
    name:'personalInfoShow',
    component:PersonalInfo
  },
},
(3)ChooseExam
addexam() {
  if (this.formpeople.money < 280) {
    alert("余额不足，请充值！")
  }else{
    this.searchExamInfo()
    //判断考试
    this.axios.get('http://localhost:8090/student_exam/single/' + this.formpeople.id + "/" +
this.selectedId)
      .then(function (res) {
        //成功
        if (res.data.id != null) {
          alert("已经注册此考试！")
        } else {
          this.dialogVisible = true;
        }
      }).bind(this))
      .catch(function (err) {
        if (err.response) {
          //失败
          console.log(err.response)
        }
      }).bind(this))
  }
}
(4)RegisteredExam
handleDelete(index, row) {
  this.formexam.id=row.id
  this.formexam.examtime=row.examtime

```

```

this.formexam.examplace=row.examplace
this.formexam.examcap=row.examcap
this.formexam.examcity=row.examcity
this.axios.delete('http://localhost:8090/student_exam/deleteexam/'+this.peopleid+'/'+row.id)
    .then(function (res) {
        //将考试费用退换，但收取 50 元退考费
        this.returnExamfee();
        //将考试考位恢复
        this.recoverexamcap();
        //成功
        alert("取消成功! ")
        this.searchPersonalInfo();
    })
}

```

(5)studentCRUD

```

<el-dialog
  title="提示"
  :visible.sync="dialogVisible"
  width="50%"
>
  <el-form ref="formstudent" :model="formstudent" label-width="80px" style="text-align: left">
    <el-form-item label="姓名">
      <el-input v-model="formstudent.name"></el-input>
    </el-form-item>
    <el-form-item label="性别">
      <el-radio-group v-model="formstudent.gender">
        <el-radio label="男"></el-radio>
        <el-radio label="女"></el-radio>
      </el-radio-group>
    </el-form-item>
    <el-form-item label="身份证号">
      <el-input v-model="formstudent.idnumber"></el-input>
    </el-form-item>
    <el-form-item label="用户名">
      <el-input v-model="formstudent.username"></el-input>
    </el-form-item>
    <el-form-item label="密码">
      <el-input v-model="formstudent.password"></el-input>
    </el-form-item>
  </el-form>

```

```
</el-form-item>
<el-form-item>
  <el-button type="primary" @click="addStudent">确定</el-button>
</el-form-item>
</el-form>
</el-dialog>
```

成绩评分表

序号	评价内容	评价等级				
1	设计思路阐述清晰，格式符合要求	优	良	中	及格	不及格
2	结构严谨，逻辑性强，语言表达准确	优	良	中	及格	不及格
3	文字通顺，技术用语准确，基本概念清楚	优	良	中	及格	不及格
4	基础理论知识扎实，回答问题有理论根据	优	良	中	及格	不及格
5	有较强的计算机应用能力	优	良	中	及格	不及格
6	功能齐全完善，能正确处理实验数据	优	良	中	及格	不及格
7	系统设计与实现方法有技巧性、新有创新意识	优	良	中	及格	不及格
8	有一定的理论或应用价值，设计思路新颖，对问题有较深刻的认识	优	良	中	及格	不及格
9	学习态度端正，主动参与性强	优	良	中	及格	不及格
10	与指导教师及时沟通，学风严谨务实，按期圆满完成规定的任务	优	良	中	及格	不及格
备注						
总分		教师 签字				