

重庆交通大学信息科学与工程学院

课程设计报告

(成员设计报告)

题 目 交通运输人员就业与再就业教育培训平台设计与实现

课程名称 企业级软件系统开发实践

专业班级 计算机科学与技术 20 级 班

团队名称 银航小组

学 号 632007060519

姓 名 张峰银

指导教师 鲁云平

2023 年 2 月

《企业级软件系统开发实践》答辩评分标准

序号	项目	评分标准与等级	评分等级
1	设计解决方案及实现60%	A、根据建立的系统逻辑模型，从多个角度合理的设计了架构，功能设计合理、数据模型设计合理；并根据设计的架构，实现的功能完善，测试充分，系统运行正确，人机接口界面友好。	
		B、根据建立的系统逻辑模型，进行了架构设计，功能设计合理、数据模型设计合理；并根据设计的架构，实现功能较为完善，测试合理，系统运行较正确，人机接口界面友好。	
		C、根据建立的系统逻辑模型，有架构设计，功能设计基本合理、数据模型设计基本合理；能实现分工的功能，系统能运行基本正确，能进行单元测试和集成测试，有基本的人机接口界面。	
		D、能设计架构，功能分解基本合理，数据模型设计基本合理，能完成基本数据的增、删、改、查功能；有基本的测试，系统能运行。	
		E、未进行架构设计，功能分解不合理，数据模型设计不合理，没有实现分工的功能，或整个设计抄袭。	
2	使用现代工具20%	A、在可行性分析和需求分析阶段，根据需要选择了合适的企业常用分析工具；在设计阶段选择了合适的、稳定的、目前企业常用的开发框架。	
		B、在可行性分析和需求分析阶段，根据需要能选择较合适的企业常用分析工具；在设计阶段选择了满足本项目的开发要求的开发框架。	
		C、在可行性分析和需求分析阶段，根据需要能选择除 OFFICE 外的分析工具；在设计阶段能选择开发框架进行开发。	
		D、在可行性分析和需求分析阶段，主要采用 OFFICE 工具记性分析；在设计阶段没有选择使用开发框架进行。	
		E、未完成可行性分析、需求分析和设计，或者抄袭。	
3	团队协作（20%）	A、独立完成了分工任务的可行性和需求分析、架构设计和实现，分析设计完善，并能有机的集成到整个系统；个人工作量饱满，个人设计能很好地与其它功能协作工作。	
		B、独立完成了分工任务的可行性和需求分析、架构设计和实现，分析设计合理，并能有机的集成到整个系统；个人工作量较饱满，个人设计能很好地与其它功能协作工作。	
		C、独立完成了分工任务的可行性和需求分析、架构设计和实现，并能集成到整个系统；个人工作量适中，个人设计能与其它功能协作工作。	
		D、能独立完成分工任务的可行性、需求分析、架构设计和实现，在协助下能集成到整个系统；个人工作量偏少，个人设计基本能与其它功能协作工作。	
		E、未完成分工任务的需求分析、架构设计和实现或抄袭。	
答辩总分：			

成员设计报告评分标准

项目	分值	评分标准与等级	等级	得分
个人设计报告	100	A、个人设计文档内容详细、功能模块设计合理，单元测试和集成测试详细，格式规范。		
		B、个人设计文档内容较为详细、功能模块设计较为合理，单元测试和集成测试较为详细，格式规范。		
		C、个人设计文档内容较为详细、功能模块设计基本合理，能实现个人功能模块的测试，格式较规范。		
		D、个人设计文档内容较为详细、功能模块设计基本合理，但存在 30%以内条目书写不完全符合要求。有个人功能模块的测试，格式基本规范。		
		E、报告不规范或存在 30%以上条目书写不完全符合要求，或抄袭、复制别人程序与文档。		
教师签字			得分	

注：1) 答辩和成员设计报告质量分别按百分制打分；

2) 综合成绩=过程表现*0.1 + 团队设计报告*0.2 + 成员设计报告*0.2+项目答辩*0.5

3) 单边和评阅报告时在相应等级画“√”，分数换算为 A\B\C\D\E 等级分别计为 95、85、75、65、30 分

目 录

根据文档自动生成目录

第一章 功能设计列表

序号	模块编号	模块名称	功能模块简述
1	1	用户管理模块	负责用户的增删改查
2	2	角色管理模块	不同的用户拥有不同的角色,不同的角色拥有不同的功能
3	3	考试管理模块	管理员可以发布考试,为考试布置试卷
4	4	试题管理模块	可以为题库进行增删改查,也可以选择试题来组成试卷
5	5	机构管理模块	负责机构资料的增删改查
6	6	资料管理模块	管理员上传文件,视频等资料

第二章 功能模块设计

2.1 用户管理模块

2.1.1 用户管理模块功能描述

可以对用户的信息进行增删改查

2.1.2 用户管理模块增删改查

管理员通过新增按钮来新增加一个用户,填写用户信息,并为用户分配角色。

角色

密码

用户信息

用户名

猪猪侠

角色

普通用户

密码

123

取消

确定

新增 导入 导出					
id	姓名	角色	密码	创建时间	操作
125	ss	ROLE_ADMIN	ss	2023-01-26T16:30:57.000+00:00	编辑 删除
126	24k纯帅	ROLE_USER	24k555	2023-10-26T17:15:28.000+00:00	编辑 删除
132	猪猪侠	ROLE_USER	123	2023-02-09T09:37:33.000+00:00	编辑 删除

点击删除按钮会提示是否删除用户，再次确认则可以删除该用户

新增

导入

导出

id	姓名	角色	密码	创建时间	操作
125	ss	ROLE_ADMIN	ss	2023-01-26T16:30:57.000+00:00	<div>编辑</div> <div>删除</div>
126	24k纯帅	ROLE_USER	24k555	2023-10-26T17:15:28.000+00:00	<div>编辑</div> <div>删除</div>
132	猪猪侠	ROLE_USER	123	2023-02-09T09:37:33.000+00:00	<div>编辑</div> <div>删除</div>

共 8 条

5条/页

< 1 2 >

前往 2 页

这是一段内容确定删除吗?

不删了

好的

2.1.3 XXX 模块设计流程

从业务角度详细描述根据输入数据产生输出数据的业务算法和流程。一般采用程序流程图描述实现的算法，并对实现算法进行描述

2.1.4 用户管理模块用户界面设计

新增

导入

导出

```
<el-button type="primary" @click="handleAdd">新增</el-button>
<el-upload action="http://localhost:9090/user/import"
  style="..."
  :show-file-list="false"
  :on-success="handleExcelImportSuccess"
>
  <el-button type="primary" >导入</el-button>
</el-upload>
<el-button type="primary" @click="exp">导出</el-button>
```

该用户界面中的按钮设计

新增 导入 导出

id	姓名	角色	密码	创建时间	操作
1	root	ROLE_ROOT	root	2022-01-26T16:29:00.000+00:00	<div>编辑 删除</div>
5	陶八纯	ROLE_USER	bai	2022-04-26T16:29:00.000+00:00	<div>编辑 删除</div>
11	11	ROLE_USER	11	2022-06-26T16:29:00.000+00:00	<div>编辑 删除</div>
17	孙德空	ROLE_USER	bai	2022-09-26T16:29:00.000+00:00	<div>编辑 删除</div>
124	沙德	ROLE_USER	22	2022-11-26T16:29:00.000+00:00	<div>编辑 删除</div>

共 7 条 5条/页 < 1 2 > 前往 1 页

```
<el-table :data="tableData" border style="width: 100%">
  <el-table-column prop="id" label="id"></el-table-column>
  <el-table-column prop="username" label="姓名"></el-table-column>
  <el-table-column prop="role" label="角色"></el-table-column>
  <el-table-column prop="password" label="密码"></el-table-column>
  <el-table-column prop="createTime" label="创建时间"></el-table-column>

  <el-table-column label="操作" width="200" align="center">
```

该用户界面的表单设计



```

<el-popconfirm
  confirm-button-text='好的'
  cancel-button-text='不用了'
  icon="el-icon-info"
  icon-color="red"
  title="这是一段内容确定删除吗？"
  @confirm="del(scope.row.id)"
>

```

删除按钮提示删除确认设计

2.1.5 接口说明

增加或更新通过 save() 函数将创建的用户数据传给后台并保存到数据库之中，保存成功或失败都会在页面中显示对应的消息。

```

save(){
  this.request.post("/user",this.form).then(res=>{
    if (res.code === '200') {
      this.$message.success( text: "保存成功")
      this.dialogFormVisible=false
    }else{
      this.$message.error("保存失败")
    }
    this.load()
  })
},

```

删除通过 del() 接口传入将要删除的用户 id, 在数据库中会找到该用户的 id 并将其删除


```

del(id){
    this.request.delete("/user/"+id).then(res =>{
        if (res.code === '200') {
            this.$message.success( text: "删除成功")
            const totalPages = Math.ceil( x: (this.total - 1) / this.pageSize)

            //记录当前页码
            //记住删除最后一条数据时当前码是最后一页
            const pagenum = this.pageNum > totalPages ? totalPages : this.pageNum

            //实现跳转
            this.pageNum = pagenum < 1 ? 1 : pagenum
            this.load()
        }else{
            this.$message.error("删除失败")
        }
    })
}

```

2.1.6 调用方式

根据前端请求的接口路径，调用对应的方法

```
private IUserService userService;
```

//新增或更新

```
@PostMapping
```

```
public Result save(@RequestBody User user){
```

```
    return Result.success(userService.saveOrUpdate(user));
```

```
}
```

```
@DeleteMapping("/{id}")
```

```
public Result delete(@PathVariable Integer id) {
```

```
    return Result.success(userService.removeById(id));
```

```
}
```

//查询

```
@GetMapping
```

```
public Result findAll() {
```

```
    return Result.success(userService.list());
```

```
}
```

```
@GetMapping("/username/{username}")
```

```
public Result findOne(@PathVariable String username) {
```

```
    QueryWrapper<User> queryWrapper=new QueryWrapper<>();
```

```
    queryWrapper.eq("username",username);
```

```
    return Result.success(userService.getOne(queryWrapper));
```

```
}
```

```
@GetMapping("/page")
```

```
public Result findPage(@RequestParam Integer pageNum,
```

```
                        @RequestParam Integer pageSize,
```

```
                        @RequestParam(defaultValue = "") String name) {
```

```
    QueryWrapper<User> queryWrapper=new QueryWrapper<>();
```

```
    queryWrapper.like("username",name);
```

```
    return Result.success(userService.page(new Page<>(pageNum,
    pageSize),queryWrapper));
```

```
}
```

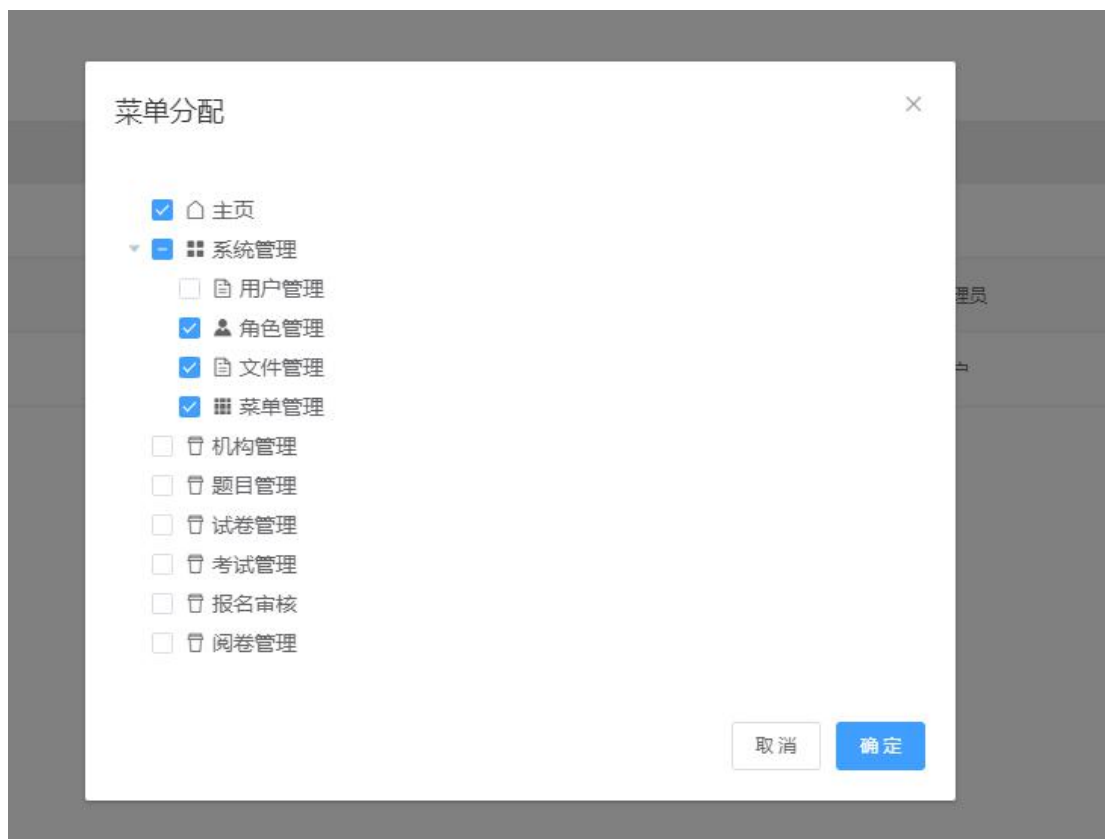
2.2 角色管理模块

2.2.1 角色管理模块功能描述

为系统创建不同的角色，分配不同的权限，管理不同的功能



ID	名称	唯一标识	描述	操作
4	管理员	ROLE_ADMIN	管理员	分配权限 编辑 删除
3	超级管理员	ROLE_ROOT	超级管理员	分配权限 编辑 删除
2	普通用户	ROLE_USER	普通用户	分配权限 编辑 删除



2.2.2 角色管理模块界面设计

请输入名称

搜索

新增

新增

批量删除

<input type="checkbox"/>	ID	名称	唯一标识	描述	操作
<input type="checkbox"/>	4	管理员	ROLE_ADMIN	管理员	<div><div>分配权限</div><div>编辑</div><div>删除</div></div>
<input type="checkbox"/>	3	超级管理员	ROLE_ROOT	超级管理员	<div><div>分配权限</div><div>编辑</div><div>删除</div></div>
<input type="checkbox"/>	2	普通用户	ROLE_USER	普通用户	<div><div>分配权限</div><div>编辑</div><div>删除</div></div>

共 3 条

5条/页

< 1 >

前往 1 页

2.2.3 接口设计

将角色信息传入后端，在后端获取所有的菜单，根据角色信息提供他权限以内的功能菜单

```
//
async selectMenu(role) {

    this.roleId = role.id
    this.roleFlag = role.flag

    // 请求菜单数据
    this.request.get("/menu").then(res => {
        this.menuData = res.data

        // 把后台返回的菜单数据处理成 id数组
        this.expends = this.menuData.map(v => v.id)
    })

    this.request.get("/role/roleMenu/" + this.roleId).then(res => {
        this.checks = res.data

        this.request.get("/menu/ids").then(r=>{
            const ids=r.data
            ids.forEach(id=>{
                if(!this.checks.includes(id)){
                    this.$refs.tree.setChecked(id, checked: false)
                }
            })
        })
    })
}
```

2.2.4 调用分析

将角色 id 和菜单 id 一一对应，传入后台数据库中储存。调用时从数据库中取出角色对应的菜单 id 并显示。

```
@PostMapping("/roleMenu/{roleId}")

public Result roleMenu(@PathVariable Integer roleId, @RequestBody
List<Integer> menuIds) {

    roleService.setRoleMenu(roleId, menuIds);

    return Result.success();

}
```

```
@GetMapping("/roleMenu/{roleId}")

public Result getRoleMenu(@PathVariable Integer roleId) {
```

```

return Result.success( roleService.getRoleMenu(roleId));
}

```

2.3 考试管理模块

2.3.1 考试管理功能功能描述

管理员可以发布考试，学员在前端就会看到发布的考试信息，学生报名参加后，需要管理员审核通过后可以参加考试。

2.3.2 考试管理功能演示

发布考试，填写考试相关信息

为考试配置对应的考卷

2.3.4 考试管理功能界面设计

<input type="text" value="请输入名称"/> <input type="button" value="搜索"/> <input type="button" value="重置"/>							
<input type="button" value="新增"/> <input type="button" value="批量删除"/>							
ID	考试名称	考试时间	持续时间(分)	考试状态	设置考卷	操作	
2	入职培训考试	2023-03-30 01:00	180	未开始	<input type="button" value="设置"/>	<input type="button" value="编辑"/>	<input type="button" value="删除"/>
1	在就业培训考试	2023-02-09 19:00	120	进行中	<input type="button" value="设置"/>	<input type="button" value="编辑"/>	<input type="button" value="删除"/>

共 2 条 5条/页 < 1 > 前往 1 页

2.3.5 接口设计

将考试 id 传入后台，获取数据库中对对应机构的试卷，将其分配给此次考试

```
setPaper(examId){
    this.form1={examId:examId}
    this.request.get("/examPaper/exam/"+examId).then(res=>{
        if(res.data){
            this.form1.paperId = res.data.paperId
            this.dialogFormVisible1=true
        }
    })
},
```

2.3.6 调用分析

根据前端传入的考试 id，在数据库中查询过后，将其结果返回

@GetMapping("/exam/{examId}")

```
public Result findByExam(@PathVariable Integer examId) {
    return Result.success(examPaperService.getOne(new
QueryWrapper<ExamPaper>().eq("exam_id",examId)));
}
```

2.4 试题管理模块

2.4.1 试题管理功能功能描述

可以为题库进行增删改查，也可以选择试题来组成试卷

2.4.2 试题管理功能演示

可以对试题进行基础的增删改查

题目信息

题目类型

选择题

课程

在就业培训机构

题目

AAAa

选项A

1

选项B

2

选项C

3

选项D

4

分数

5

答案

☒ A ☐ B ☐ C ☐ D

解析

取消

确定

<input type="checkbox"/>	ID	题目	课程名称	题目类型	选项A	选项B	选项C	选项D	分数	答案	解析	作者id	创建时间	操作
<input type="checkbox"/>	40	AAAa	在就业培训机构	选择题	1	2	3	4	5	A		1	2023-02-09 20:15:14	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	36	青青姐好不好18	在就业培训机构	判断题					5	是		1	2023-01-29 23:39:06	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	35	青青姐好不好17	在就业培训机构	判断题					5	是		1	2023-01-29 23:39:06	<div>编辑</div> <div>删除</div>

也可以为题库增加，删除试卷，有自动组卷和手动组卷两个模式可供选择

自动组卷

选择题

4

判断题

4

问答题

2

取消

确定

查看试卷

(选择题)1.青哥哥好不好1

A.好 B.很好 C.差 D.很差

答案: A

解析:

(选择题)2.青哥哥好不好2

A.好 B.很好 C.差 D.很差

答案: B

解析:

(选择题)3.青哥哥好不好3

A.好 B.很好 C.差 D.很差

答案: C

解析:

(选择题)4.青哥哥好不好4

A.好 B.很好 C.差 D.很差

答案: D

解析:

(判断题)5.青哥哥好不好10

答案: 是

解析:

(判断题)6.青哥哥好不好11

答案: 是

解析:

(判断题)7.青哥哥好不好12

手动组卷

列表 1 0/17

请选择类型: 1选择, 29

☐ 青哥哥好不好3

☐ 青哥哥好不好4

☐ 青哥哥好不好5

☐ 青哥哥好不好6

☐ 青哥哥好不好7

☐ 青哥哥好不好8

☐ 青哥哥好不好9

列表 2 0/2

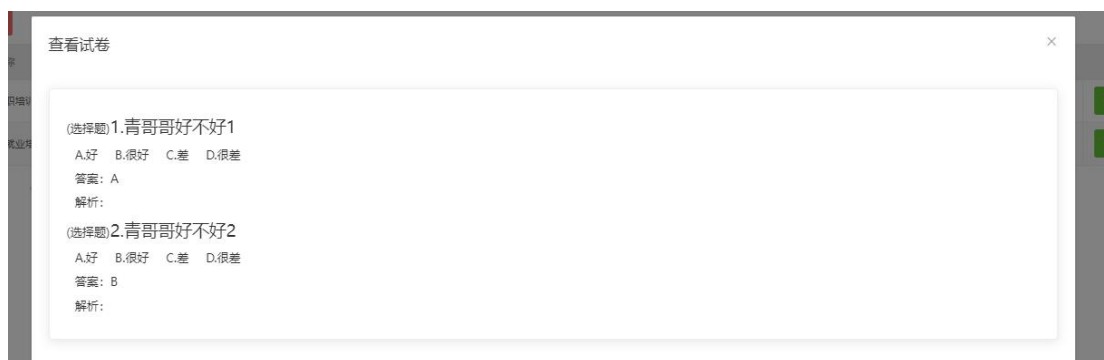
请选择类型: 1选择, 29

☐ 青哥哥好不好1

☐ 青哥哥好不好2

取消

确定



2.4.4 试题管理功能界面设计

请输入名称 <input type="text"/> <input type="button" value="搜索"/> <input type="button" value="重置"/>													
新增 <input type="button" value="新增"/> 批量删除 <input type="button" value="批量删除"/>													
ID	题目	课程名称	题目类型	选项A	选项B	选项C	选项D	分数	答案	解析	作题数	创建时间	操作
40	AAAa	在就业培训机构	选择题	1	2	3	4	5	A		1	2023-02-09 20:15:14	<input type="button" value="编辑"/> <input type="button" value="删除"/>
36	青哥哥好不好18	在就业培训机构	判断题					5	是		1	2023-01-29 23:39:06	<input type="button" value="编辑"/> <input type="button" value="删除"/>
35	青哥哥好不好17	在就业培训机构	判断题					5	是		1	2023-01-29 23:39:06	<input type="button" value="编辑"/> <input type="button" value="删除"/>
34	青哥哥好不好16	在就业培训机构	问答题					10	错		1	2023-01-29 23:39:06	<input type="button" value="编辑"/> <input type="button" value="删除"/>
33	青哥哥好不好15	在就业培训机构	问答题					10	错		1	2023-01-29 23:39:06	<input type="button" value="编辑"/> <input type="button" value="删除"/>

共 19 条 页

root

请输入名称 <input type="text"/> <input type="button" value="搜索"/> <input type="button" value="重置"/>													
新增 <input type="button" value="新增"/> 批量删除 <input type="button" value="批量删除"/>													
ID	名称	总分	课程名称	试题	操作								
2	入职培训试题		文鼎就业培训机构	<input type="button" value="查看试题"/>	<input type="button" value="编辑"/>	<input type="button" value="自动删除"/>	<input type="button" value="手动删除"/>	<input type="button" value="删除"/>					
1	在就业培训试题		在就业培训机构	<input type="button" value="查看试题"/>	<input type="button" value="编辑"/>	<input type="button" value="自动删除"/>	<input type="button" value="手动删除"/>	<input type="button" value="删除"/>					

共 2 条 页

2.4.5 接口设计

查询数据库中的题目信息，并将其呈现在界面上

```

load() {
  this.request.get("/question/page", {
    params: {
      pageNum: this.pageNum,
      pageSize: this.pageSize,
      name: this.name,
    }
  }).then(res => {
    this.tableData = res.data.records
    this.total = res.data.total
  })

  this.request.get("/question/ids").then(r => {
    this.ids = r.data
  })
},

```

自动组卷将前端选好的各种类型的题目数量传到后台，由后台产生随机数，取出题目组成试卷。

手动组卷将所有题目获取到前端，选择题目后将题目 id 记录下，存储到试卷中，组成试卷

```
generatePaper(){
    this.request.post("/paper/takePaper", this.form1).then(res =>{
        if (res.code === '200') {
            this.$message.success( text: "自动组卷成功")
            this.dialogFormVisible1 = false
            // this.load()
        } else {
            this.$message.error("保存失败")
        }
    })
},
generateHandPaper(){
    this.request.post("/paper/handPaper", this.form2).then(res =>{
        if (res.code === '200') {
            this.$message.success( text: "手动组卷成功")
            this.dialogFormVisible3 = false
            // this.load()
        } else {
            this.$message.error("保存失败")
        }
    })
},
}
```

2.4.6 调用分析

@PostMapping("/takePaper")

public Result takePaper(@RequestBody PaperDto paperDto) {

//删除老的试卷

UpdateWrapper<PaperQuestion> updateWrapper=new

UpdateWrapper<>();

updateWrapper.eq("paper_id",paperDto.getPaperId());

paperQuestionService.remove(updateWrapper);

```

    QueryWrapper<Question> queryWrapper = new QueryWrapper<>();
    queryWrapper.eq("course_id",paperDto.getCourseId());
    //根据课程 id 查出所有该课程的题目，再根据 type 区分
    List<Question> questionList = questionService.list(queryWrapper);
    List<Question> type1List = questionList.stream().filter(question ->
question.getType()==1).collect(Collectors.toList());    //选择
    List<Question> type2List = questionList.stream().filter(question ->
question.getType()==2).collect(Collectors.toList());    //判断
    List<Question> type3List = questionList.stream().filter(question ->
question.getType()==3).collect(Collectors.toList());    //问答
    if(type1List.size()<paperDto.getType1()){
        throw new ServiceException("-1","选择题数量不足");
    }
    if(type2List.size()<paperDto.getType2()){
        throw new ServiceException("-1","判断题数量不足");
    }
    if(type3List.size()<paperDto.getType3()){
        throw new ServiceException("-1","问答题数量不足");
    }

    //开始随机组卷
    List<PaperQuestion> paperQuestion
=getPaperQuestion(type1List.size(),paperDto.getType1(),type1List,paperDto.getPape
rId());

    paperQuestion.addAll(getPaperQuestion(type2List.size(),paperDto.getType2(),type2L
ist,paperDto.getPaperId()));

    paperQuestion.addAll(getPaperQuestion(type3List.size(),paperDto.getType3(),type3L

```

```
ist,paperDto.getPaperId()));

        paperQuestionService.saveBatch(paperQuestion);

//        paperService.saveOrUpdate(paperDto);

        return Result.success();

    }
}
```

2.5 机构管理模块

2.5.1 机构管理功能描述

负责机构的增删改查

2.5.2 机构管理功能演示

点击新增按钮来填写机构信息，确认后将机构信息存储到数据库里

题目信息

名称

1111

备注

11

取消

确定

<input type="checkbox"/>	ID	名称	备注	操作
<input type="checkbox"/>	5	1111	11	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	2	交通职业培训机构	无	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	1	在就业培训机构	无	<div>编辑</div> <div>删除</div>

共 3 条5条/页<1>前往1页

同样也可以对机构进行基础的增删改查

新增 删除

<input type="checkbox"/>	ID	名称	备注	操作
<input type="checkbox"/>	5	1111	11	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	2	交通职业培训机构	无	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	1	在就业培训机构	无	<div>编辑</div> <div>删除</div>

共 3 条5条/页<1>前往1页

删除成功

请输入名称

搜索

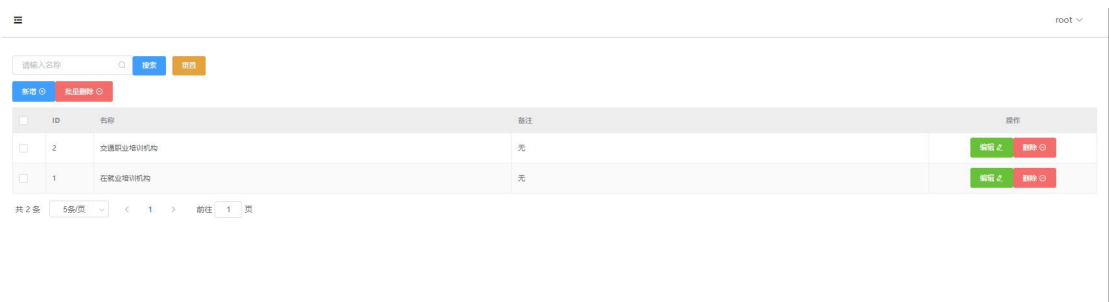
清除

新增 删除

<input type="checkbox"/>	ID	名称	备注	操作
<input type="checkbox"/>	2	交通职业培训机构	无	<div>编辑</div> <div>删除</div>
<input type="checkbox"/>	1	在就业培训机构	无	<div>编辑</div> <div>删除</div>

共 2 条5条/页<1>前往1页

2.5.4 机构管理功能界面设计



2.5.5 接口设计

```
save() {
    this.request.post("/course", this.form).then(res => {
        if (res.code === '200') {
            this.$message.success( text: "保存成功")
            this.dialogFormVisible = false
            this.load()
        } else {
            this.$message.error("保存失败")
        }
    })
},
```

```
del(id) {
    this.request.delete("/course/" + id).then(res => {
        if (res.code === '200') {
            this.$message.success( text: "删除成功")
            this.load()
        } else {
            this.$message.error("删除失败")
        }
    })
},
```

2.5.6 调用分析

将前端表单数据传到后台，后台将数据存入数据库中

// 新增或者更新

```
@PostMapping
public Result save(@RequestBody Course course) {
    courseService.saveOrUpdate(course);
}
```

```
        return Result.success();
    }

    @DeleteMapping("/{id}")
    public Result delete(@PathVariable Integer id) {
        courseService.removeById(id);
        return Result.success();
    }

    @PostMapping("/del/batch")
    public Result deleteBatch(@RequestBody List<Integer> ids) {
        courseService.removeByIds(ids);
        return Result.success();
    }
```

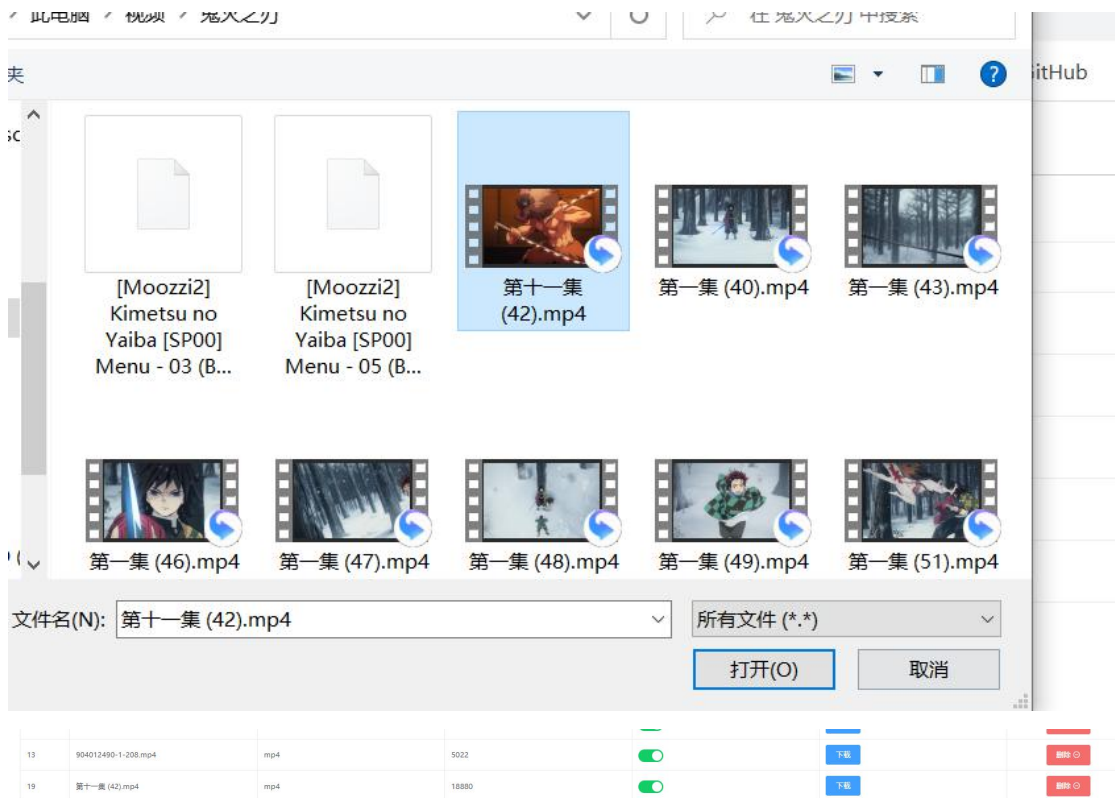
2.6 资料管理模块

2.6.1 资料管理功能描述

管理员上传文件，视频等资料

2.6.2 资料管理功能演示

点击上传文件，选择文件后可以上传



也可以通过下载按钮来在线获取文件，可以对文件进行删除



2.6.4 资料管理功能界面设计

root						
上传文件						
ID	文件名	文件类型	文件大小(kb)	启用	下载	操作
8	1537162709627.jpg	jpg	100	<input checked="" type="checkbox"/>	下载	删除
9	1111.jpg	jpg	100	<input checked="" type="checkbox"/>	下载	删除
11	QQ图片20220425203510.jpg	jpg	139	<input type="checkbox"/>	下载	删除
12	用户信息 (1).xlsx	xlsx	3	<input checked="" type="checkbox"/>	下载	删除
13	904012490-1-208.mp4	mp4	5022	<input checked="" type="checkbox"/>	下载	删除
19	第十一集 (42).mp4	mp4	18880	<input checked="" type="checkbox"/>	下载	删除

2.6.5 接口设计

从数据库中请求存储的文件信息数据，将其展现给管理员，方便管理员查看

```

load(){
    this.request.get("/file/page",{
        params:{
            pageNum:this.pageNum,
            pageSize:this.pageSize
        }
    }).then(res=>{
        console.log(res)
        this.tableData=res.data.records
        this.total=res.data.total
    })
},

```

2.6.6 调用分析

选择文件上传后，先将文件存到磁盘，然后判断配置的文件目录是否存在，如果不存在则为其分配唯一的 md5 标识，将其信息存入数据库保存

@PostMapping("/upload")

public String upload(@RequestParam MultipartFile file) throws IOException

{

String originalFilename =file.getOriginalFilename();

String type= FileUtil.extName(originalFilename);

long size =file.getSize();

//先存到磁盘

File uploadParentFile = new File(fileUploadPath);

//判断配置的文件目录是否存在

if(!uploadParentFile.exists()){

uploadParentFile.mkdir();

}

String uuid =IdUtil.fastSimpleUUID();

String fileUuid=uuid+ StrUtil.DOT+type;

String md5,url;

File uploadFile=new File(fileUploadPath+fileUuid);


```

if(uploadFile.exists()){
    //获取文件的 md5
    md5= SecureUtil.md5(uploadFile);

    //从数据库查询是否存在相同的记录
    Files dbFiles=getFileByMd5(md5);

    if(dbFiles!=null){
        url=dbFiles.getUrl();
    }else {
        //获取的文件存储到磁盘文件
        file.transferTo(uploadFile);
        url="http://localhost:9090/file/"+fileUuid;
    }

}

}

//存储数据库
Files saveFile=new Files();
saveFile.setName(originalFilename);
saveFile.setType(type);

```

```

        saveFile.setSize(size/1024);

        saveFile.setUrl(url);

        saveFile.setMd5(md5);

        fileMapper.insert(saveFile);

        return url;
    }

    @GetMapping("/{fileUuid}")
    public void download(@PathVariable String fileUuid, HttpServletResponse
response) throws IOException {

        File uploadFile =new File(fileUploadPath + fileUuid);

        ServletOutputStream os = response.getOutputStream();

        URLEncoder urlEncoder=new URLEncoder();

        response.addHeader("Content-Disposition","attachment;filename=" +
urlEncoder.encode(fileUuid, StandardCharsets.UTF_8));

        response.setContentType("application/octet-stream");

        os.write(FileUtil.readBytes(uploadFile));

        os.flush();

        os.close();

    }

```

第三章 测试

3.1 测试环境

3.2 功能模块测试

采用路径测试法，对实现的每个功能模块的所有执行路径进行测试。

3.3 模块功能测试

测试对象的功能测试应该侧重于可以被直接追踪到用例或业务功能和业务规则的所有测试需求。这些测试的目标在于核实能否正确地接受、处理和检索数

据以及业务规则是否正确实施。这种类型的测试基于黑盒方法，即通过图形用户界面（GUI）与应用程序交互并分析输出结果来验证应用程序及其内部进程。以下列出的是每个应用程序推荐的测试方法概要：

测试目标：	确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等。
方法：	利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容： <ul style="list-style-type: none">• 在使用有效数据时得到预期的结果。• 在使用无效数据时显示相应的错误消息或警告消息。• 各业务规则都得到了正确的应用。
完成标准：	<ul style="list-style-type: none">• 所计划的测试已全部执行。• 所发现的缺陷已全部解决。
需考虑的特殊事项：	确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）

3.4 用户管理模块的集成测试

点击新增用户



新增用户成功

保存成功

root

新增	导入	导出			
id	姓名	角色	昵称	创建时间	操作
1	root	ROLE_ROOT	root	2022-01-26T16:29:00.000+00:00	编辑 删除
5	猪八戒	ROLE_USER	bai	2022-04-26T16:29:00.000+00:00	编辑 删除
11	11	ROLE_USER	11	2022-06-26T16:29:00.000+00:00	编辑 删除
17	孙悟空	ROLE_USER	bai	2022-09-26T16:29:00.000+00:00	编辑 删除
124	沙僧	ROLE_USER	22	2022-11-26T16:29:00.000+00:00	编辑 删除
125	ss	ROLE_ADMIN	ss	2023-01-26T16:30:57.000+00:00	编辑 删除
126	24k纯纯	ROLE_USER	24k555	2023-10-26T17:15:28.000+00:00	编辑 删除
134	777	ROLE_USER	77	2023-02-10T04:44:24.000+00:00	编辑 删除

共 8 条10条/页< 1 >前往 1 页

编辑用户信息成功

保存成功

root

新增	导入	导出			
id	姓名	角色	昵称	创建时间	操作
1	root	ROLE_ROOT	root	2022-01-26T16:29:00.000+00:00	编辑 删除
5	猪八戒	ROLE_USER	bai	2022-04-26T16:29:00.000+00:00	编辑 删除
11	11	ROLE_USER	11	2022-06-26T16:29:00.000+00:00	编辑 删除
17	孙悟空	ROLE_USER	bai	2022-09-26T16:29:00.000+00:00	编辑 删除
124	沙僧	ROLE_USER	22	2022-11-26T16:29:00.000+00:00	编辑 删除
125	ss	ROLE_ADMIN	ss	2023-01-26T16:30:57.000+00:00	编辑 删除
126	24k纯纯	ROLE_USER	24k555	2023-10-26T17:15:28.000+00:00	编辑 删除
134	7778888	ROLE_USER	77	2023-02-10T04:44:24.000+00:00	编辑 删除

共 8 条10条/页< 1 >前往 1 页

删除用户成功

删除成功

root

新增	导入	导出			
id	姓名	角色	昵称	创建时间	操作
1	root	ROLE_ROOT	root	2022-01-26T16:29:00.000+00:00	编辑 删除
5	猪八戒	ROLE_USER	bai	2022-04-26T16:29:00.000+00:00	编辑 删除
11	11	ROLE_USER	11	2022-06-26T16:29:00.000+00:00	编辑 删除
17	孙悟空	ROLE_USER	bai	2022-09-26T16:29:00.000+00:00	编辑 删除
124	沙僧	ROLE_USER	22	2022-11-26T16:29:00.000+00:00	编辑 删除
125	ss	ROLE_ADMIN	ss	2023-01-26T16:30:57.000+00:00	编辑 删除
126	24k纯纯	ROLE_USER	24k555	2023-10-26T17:15:28.000+00:00	编辑 删除

共 7 条10条/页< 1 >前往 1 页

3.5 角色管理模块的集成测试

测试角色的菜单分配功能

root

系统管理

用户管理

角色管理

文件管理

菜单管理

机构管理

岗位管理

试卷管理

考试管理

报名审核

网络管理

请输入姓名

新增

删除

id	名称	唯一标识	描述	操作
<input type="checkbox"/>	4	管理员	ROLE_ADMIN	分配菜单 编辑 删除
<input type="checkbox"/>	3	超级管理员	ROLE_ROOT	分配菜单 编辑 删除
<input type="checkbox"/>	2	普通用户	ROLE_USER	分配菜单 编辑 删除

共 3 条5条/页< 1 >前往 1 页



3.6 考试管理模块的集成测试

发布考试信息

The form for publishing exam information includes the following fields:

- 考试名称: 在线考试test
- 考试时间: 2023-02-11 00:00
- 持续时间: 120
- 课程: 在就业培训机构

Buttons: 取消, 确定

Table with 8 columns: ID, 考试名称, 考试时间, 持续时间(分), 考试状态, 设置专家, 操作. The table contains 3 rows of exam data.

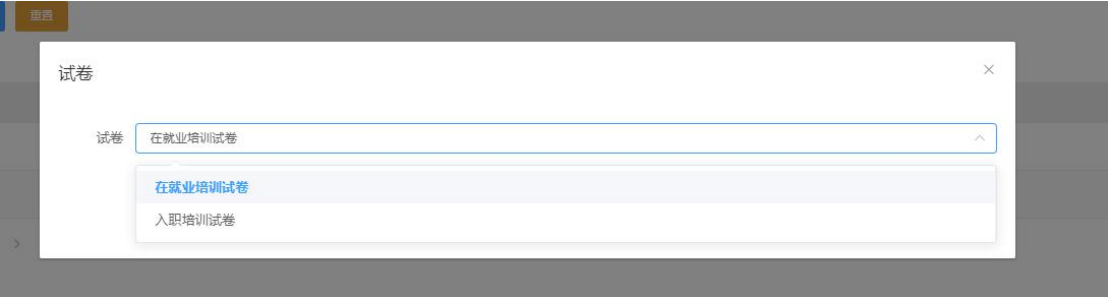
ID	考试名称	考试时间	持续时间(分)	考试状态	设置专家	操作
4	在线考试test	2023-02-11 00:00	120	未开始	设置	编辑 删除
2	入职培训考试	2023-03-30 01:00	180	未开始	设置	编辑 删除
1	在就业培训机构	2023-02-09 19:00	120	已结束	设置	编辑 删除

删除考试信息

Table with 8 columns: ID, 考试名称, 考试时间, 持续时间(分), 考试状态, 设置专家, 操作. The table contains 2 rows of exam data after deleting the first row.

ID	考试名称	考试时间	持续时间(分)	考试状态	设置专家	操作
2	入职培训考试	2023-03-30 01:00	180	未开始	设置	编辑 删除
1	在就业培训机构	2023-02-09 19:00	120	已结束	设置	编辑 删除

为考试设置试卷



3.7 试题管理模块的集成测试

新增加一道题目



保存成功

ID	题目	课程名称	题目类型	选项A	选项B	选项C	选项D	分数	答案	解析	作者Id	创建时间	操作
41	这是一道题	交通职业培训机构	判断题					5	是		1	2023-02-10 13:11:09	编辑 删除
36	哥哥好不好18	在就业培训机构	判断题					5	是		1	2023-01-29 23:39:06	编辑 删除
35	哥哥好不好17	在就业培训机构	判断题					5	是		1	2023-01-29 23:39:06	编辑 删除
34	哥哥好不好16	在就业培训机构	问答题					10	是		1	2023-01-29 23:39:06	编辑 删除

删除题目

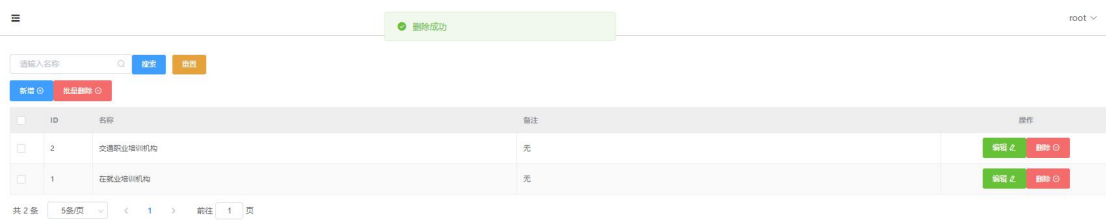


3.8 机构管理模块的集成测试

新增加一个机构

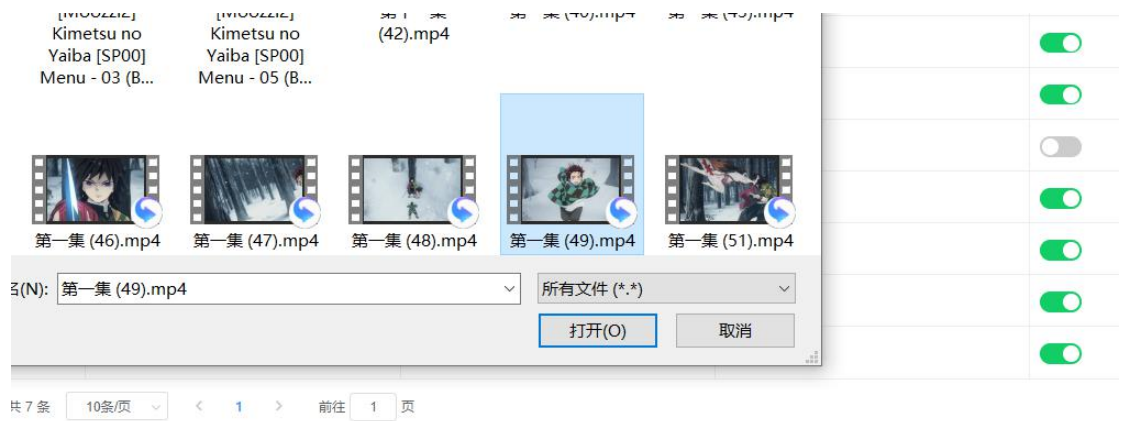


删除机构



3.9 资料管理模块的集成测试

上传视频资料



11	QQ图片20220425203510.jpg	jpg	139	<input type="checkbox"/>	下载	删除
12	用户信息 (1).xlsx	xlsx	3	<input checked="" type="checkbox"/>	下载	删除
13	904012490-1-208.mp4	mp4	5022	<input checked="" type="checkbox"/>	下载	删除
19	第十一集 (42).mp4	mp4	18880	<input checked="" type="checkbox"/>	下载	删除
20	第一集 (43).mp4	mp4	9107	<input checked="" type="checkbox"/>	下载	删除
21	第一集 (49).mp4	mp4	7435	<input checked="" type="checkbox"/>	下载	删除

共 8 条 10条/页 < 1 > 前往 1 页

删除资料

删除资料						
上传文件						
ID	文件名	文件类型	文件大小(kb)	启用	下载	操作
1	1537162703627.jpg	jpg	100	<input checked="" type="checkbox"/>	下载	删除
1	1111.jpg	jpg	100	<input checked="" type="checkbox"/>	下载	删除
1	QQ图片20220425203510.jpg	jpg	139	<input type="checkbox"/>	下载	删除
2	用户信息 (1).xlsx	xlsx	3	<input checked="" type="checkbox"/>	下载	删除
3	904012490-1-208.mp4	mp4	5022	<input checked="" type="checkbox"/>	下载	删除
9	第十一集 (42).mp4	mp4	18880	<input checked="" type="checkbox"/>	下载	删除
0	第一集 (43).mp4	mp4	9107	<input checked="" type="checkbox"/>	下载	删除

共 7 条 10条/页 < 1 > 前往 1 页

第四章 谈谈你对项目管理的理解