

计算机科学与技术学院课程设计成绩单

课程名称：数据库系统课程设计

| | | | | | | | | |
|------------------------------|-------------|----|------|----|--------------|--|------|-----------------|
| 姓名 | 卢俊朴 | 性别 | 男 | 学号 | 202113407120 | | 班级 | 软 件 工 程 2101 |
| 电话 | 15527752855 | | 综合成绩 | | | | 成绩等级 | |
| 程序运行情况 （占总成绩 20%） | | | | | | | | |
| 程序功能完善程度 （占总成绩 10%） | | | | | | | | |
| 数据库逻辑结构的合理性 （占总成绩 10%） | | | | | | | | |
| 对问题的答辩情况 （占总成绩 40%） | | | | | | | | |
| 学生的工作态度与独立工作能力 （占总成绩 10%） | | | | | | | | |
| 设计报告的规范性 （占总成绩 10%） | | | | | | | | |

A: 90~100 分 A-: 85~89 分 B+: 82~84 分 B: 78~81 分 B-: 75~77 分

C+: 72~74 分 C: 68~71 分 C-: 64~67 分 D: 60~63 分 F: <60 分



武汉科技大学

Wuhan University of Science & Technology

计算机科学与技术学院

课 程 设 计 报 告

课程名称： 数据库系统原理

专 业： 软件工程

班 级： 21 级 2101 班

学 号： 202113407120

姓 名： 卢俊朴

指导老师： 张智

目录

| | |
|---------------------------------------|----|
| 校园选课系统 | 7 |
| 一、 需求分析 | 7 |
| 设计用途: | 7 |
| 功能模块: | 7 |
| 整体功能: | 7 |
| 细节功能（方便使用的功能）: | 8 |
| 二、 概念结构设计 | 8 |
| E-R 图: | 8 |
| 三、 逻辑结构设计 | 9 |
| 表结构: | 9 |
| 项目内文件结构: | 12 |
| 四、 开发工具 | 14 |
| 五、 整体运行展示 | 14 |
| 管理员 | 14 |
| 教师 | 18 |
| 学生 | 20 |
| 六、 具体实现 | 21 |
| 说明: | 21 |
| （一） 注册模块 | 21 |
| 运行展示 | 21 |
| 实现步骤 | 22 |
| 涉及主要代码 | 22 |
| （二） 登录模块（以管理员 Admininfo 的的登录为例） | 24 |
| 运行展示 | 25 |
| 实现步骤 | 25 |

| | |
|-------------------------------------|----|
| 涉及主要代码 | 26 |
| (三) 修改密码模块 | 30 |
| 运行展示 | 30 |
| 实现步骤 | 30 |
| 涉及主要代码 | 31 |
| (四) 登出模块 | 33 |
| 实现步骤 | 33 |
| 涉及主要代码 | 33 |
| (五) 个人主页模块实现 (以 AdminInfo 为例) | 33 |
| 运行展示 | 34 |
| 实现步骤 | 34 |
| 实现步骤涉及主要代码 | 35 |
| (六) 信息的增删改查 (以 AdminInfo 为例) | 39 |
| 1. 增加 | 39 |
| 运行展示 | 39 |
| 增加实现步骤 | 39 |
| 增加涉及主要代码 | 40 |
| 3, exception 包内的自定义异常类和异常处理类 | 43 |
| 2. 查询 | 45 |
| 运行展示 | 45 |
| 查询实现步骤 | 45 |
| 查询涉及主要代码 | 45 |
| 3. 编辑 | 47 |
| 编辑实现步骤 | 48 |
| 编辑涉及主要代码 | 48 |
| 4. 删除 | 49 |
| 运行展示 | 50 |

| | |
|----------------------------------|----|
| 删除实现步骤 | 50 |
| 删除涉及主要代码 | 50 |
| (七) 信息分页查询模块 | 52 |
| 运行展示 | 52 |
| 实现步骤 | 53 |
| 涉及主要代码 | 53 |
| (八) 根据名字模糊查询模块 | 55 |
| 实现步骤 | 56 |
| 涉及主要代码 | 56 |
| (九) 双表联动 | 58 |
| 1. 学生数据库获取学院数据库信息 | 59 |
| 运行展示 | 59 |
| 实现步骤 | 59 |
| 涉及主要代码 | 59 |
| 2. 学生模块根据与学院模块共有属性展示学院特有信息 | 60 |
| 运行展示 | 61 |
| 实现步骤 | 61 |
| 涉及主要代码 | 61 |
| 步骤 2 实现的两种方式 | 62 |
| (十) 三表联动 | 63 |
| 运行展示 | 64 |
| 实现步骤 | 64 |
| 涉及主要代码 | 64 |
| (十一) 四表联动 | 65 |
| 运行展示 | 65 |
| 实现步骤 | 66 |
| 涉及主要代码 | 66 |

| | |
|------------------------|----|
| (十二) 用户角色权限的控制模块 | 66 |
| 实现步骤 | 67 |
| 部分使用演示 | 67 |
| (十三) 学生选课和退课模块 | 68 |
| 运行展示 | 68 |
| 实现步骤 | 69 |
| 涉及主要代码 | 69 |
| (十四) 老师开课和结课模块 | 75 |
| 实现步骤 | 76 |
| 涉及主要代码 | 76 |
| 六、总结 | 79 |

校园选课系统

一、需求分析

设计用途：

本系统设计可以提供一个学校中不同身份用户都可以注册登录，更改个人信息，方便管理员统一管理。并可以让不同身份用户显示不同信息、实现不同功能。方便管理员可以增删改查各种信息(管理员、教师、学生、学院、专业、课程)，来达到对学校注册此网站人员的管理和学院、专业，课程的管理。让每个学生都能查看各个课程的信息，并进行选课退课，查看自己所选的所有课程的信息。并让教师可以选择何时开课结课，查看自己所任职班级的学生情况。

功能模块：

整体功能：

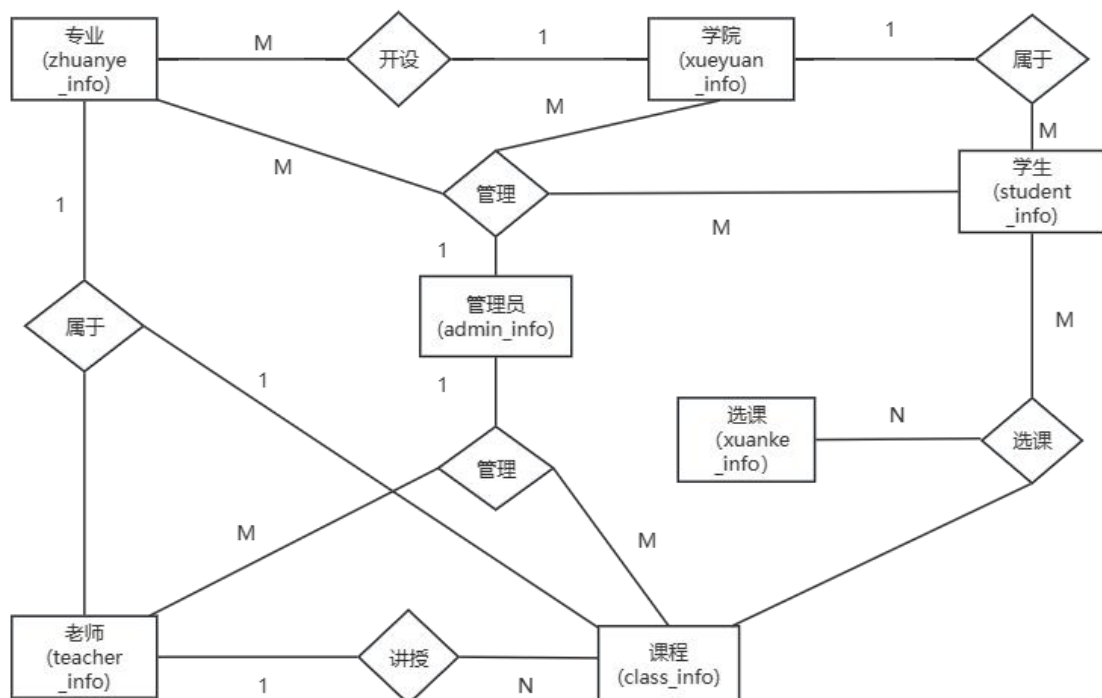
1. 可以实现三种不同类型用户(管理员、教师、学生)的注册、登录、修改密码、登出。
2. 每钟类型的每个用户都有自己的个人信息界面，可以更新，设置，查看自己的个人信息。
3. 当作为管理员登录时，可以查看并增加、删除，修改管理员、教师、学生、学院、专业和课程信息。
4. 当作为教师登陆时，能查看学院，专业，课程和选课信息。而且能看到所有自己所教课的选课情况，决定何时开课，开课后学生无法选课。并可以选择何时结课，结课后会根据课程的学分增加选课学生的总学分。
5. 当作为学生登陆时，只能查看学院，专业，课程和选课信息，并能在老师开课前选课和退课。

细节功能（方便使用的功能）：

1. 在管理员、教师，学生信息模块，进行了分页，游览信息更清晰有序。
2. 所有信息显示模块都可以进行模糊查询（输入部分用户名/学院名/专业名/课程名查询所有）。
3. 各个用户可以在个人信息模块，下拉框选中外码所在数据库中数据选择自身数据。（因为外键一般为其他表的id，在更新设置自身属性时根据其他表 name 自动填入 id 更加方便实用）
4. 在各个信息显示的模块，显示的信息还有外键所在从表的信息。（因为外键一般是 id，只显示 id 不方便查看信息）

二、概念结构设计

E-R 图：



三、逻辑结构设计

表结构：

表一：admin_info 管理员信息表

| 属性名 | 含义 | 类型 | 说明 |
|----------|------|---------|-----------------|
| id | id | Long | 主键，自增 |
| name | 用户名 | String | |
| password | 密码 | String | |
| sex | 性别 | String | |
| age | 年龄 | string | |
| phone | 电话号码 | String | |
| level | 权限等级 | Integer | 1 管理员 2 教师 3 学生 |

表二：student_info 学生信息表

| 属性名 | 含义 | 类型 | 说明 |
|----------|------|---------|-----------------|
| id | id | Long | 主键，自增 |
| name | 用户名 | String | |
| password | 密码 | String | |
| sex | 性别 | String | |
| age | 年龄 | string | |
| code | 学号 | String | 学号 |
| level | 权限等级 | Integer | 1 管理员 2 教师 3 学生 |

| | | | |
|-----------|-------|---------|-------------|
| xueyuanId | 学院 id | String | 外键，引用学院表 id |
| score | 总学分 | Integer | 完成课程所获得的总学分 |

表三：teacher_info 教师信息表

| 属性名 | 含义 | 类型 | 说明 |
|-----------|-------|---------|-----------------|
| id | id | Long | 主键，自增 |
| name | 用户名 | String | |
| password | 密码 | String | |
| sex | 性别 | String | |
| age | 年龄 | string | |
| zhicheng | 职称 | String | |
| level | 权限等级 | Integer | 1 管理员 2 教师 3 学生 |
| zhuanyeld | 专业 id | Long | 外键，引用专业 id |

表四：xueyuan_info 学院信息表

| 属性名 | 含义 | 类型 | 说明 |
|-------------|------|---------|---------------|
| id | id | Long | 主键，自增 |
| name | 学院名 | String | |
| description | 学院描述 | String | |
| score | 学分限制 | Integer | 这个学院最多可以修多少学分 |

表五：zhuanye_info 专业信息表

| 属性名 | 含义 | 类型 | 说明 |
|------------|------|--------|------------|
| id | id | Long | 主键，自增 |
| name | 学院名 | String | |
| department | 所属系 | String | 本专业属于哪个系 |
| xueyuanId | 所属学院 | Long | 外键，引用学院 id |

表六：class_info 课程信息表

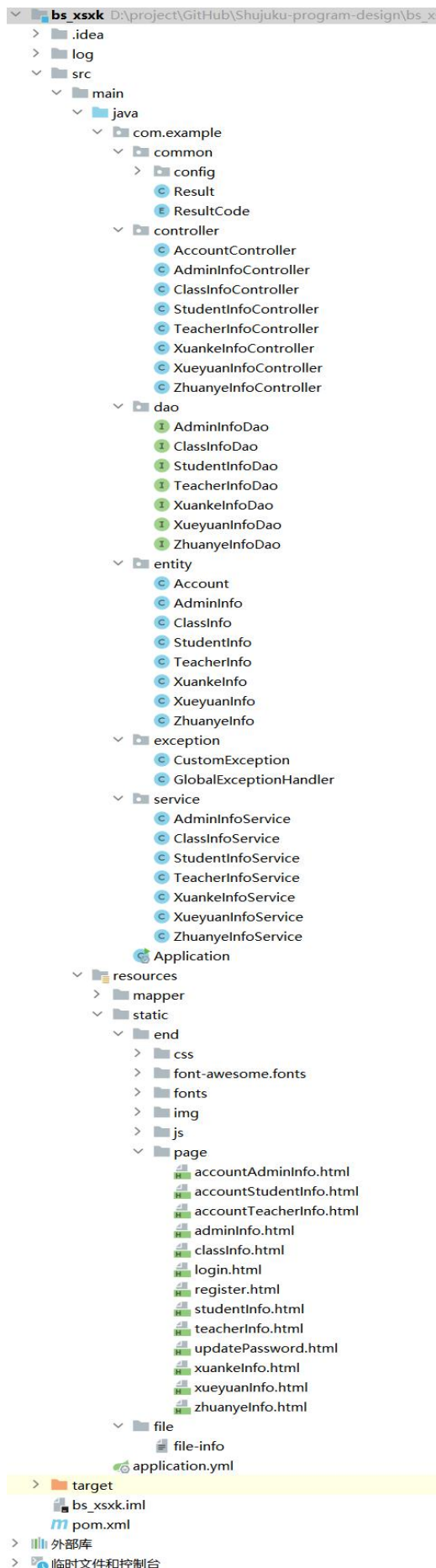
| 属性名 | 含义 | 类型 | 说明 |
|-------------|---------|---------|--------------|
| id | id | Long | 主键，自增 |
| name | 课程名 | String | |
| description | 课程描述 | String | |
| zhuanyId | 所属专业 | Long | 外键，引用专业 id |
| score | 课程学分 | Integer | 学生学完增加多少学分 |
| teacherId | 课程教师 id | Long | 外键，引用教师 id |
| kaiban | 开班人数 | Integer | 最多能有多少学生选择此课 |
| time | 开课时间 | String | |
| location | 开课位置 | String | |
| yixuan | 已选人数 | Integer | |

表七：xuanke_info 选课信息表

| 属性名 | 含义 | 类型 | 说明 |
|-----|----|----|----|
|-----|----|----|----|

| 属性名 | 含义 | 类型 | 说明 |
|-------------|---------|---------|-------------------|
| id | id | Long | 主键，自增 |
| name | 课程名 | String | |
| description | 课程描述 | String | |
| zhuanYeId | 所属专业 | Long | 外键，引用专业 id |
| score | 课程学分 | Integer | 学生学完增加多少学分 |
| teacherId | 课程教师 id | Long | 外键，引用教师 id |
| kaiban | 开班人数 | Integer | 最多能有多少学生选择此课 |
| time | 开课时间 | String | |
| location | 开课位置 | String | |
| studentId | 选课学生 id | Long | 外键，引用学生 id |
| status | 课程状态 | String | 1 待开课 2 已开课 3 已结课 |

项目内文件结构：



使用 REST 软件架构设计风格

MVC 开发模式

Config 包中的 Result 泛型类为返回统一结果的类

ResultCode 枚举类为各种结果

Controller 包中的 AccountController 为处理各种用户问题的控制器(登录，注册，获取当前用户，登出)

entity 包中的每个类对应数据库各个表

其中 Account 为三种用户类型的共同父类，有他们的

共同特征，方便 AccountController 的用户相关操作

exception 包内为自定义捕获异常和处理异常的类

四、开发工具

开发环境：Win11, IDEA 2023.1.1, MySQL Workbench 8.0 CE, GitHub Desktop

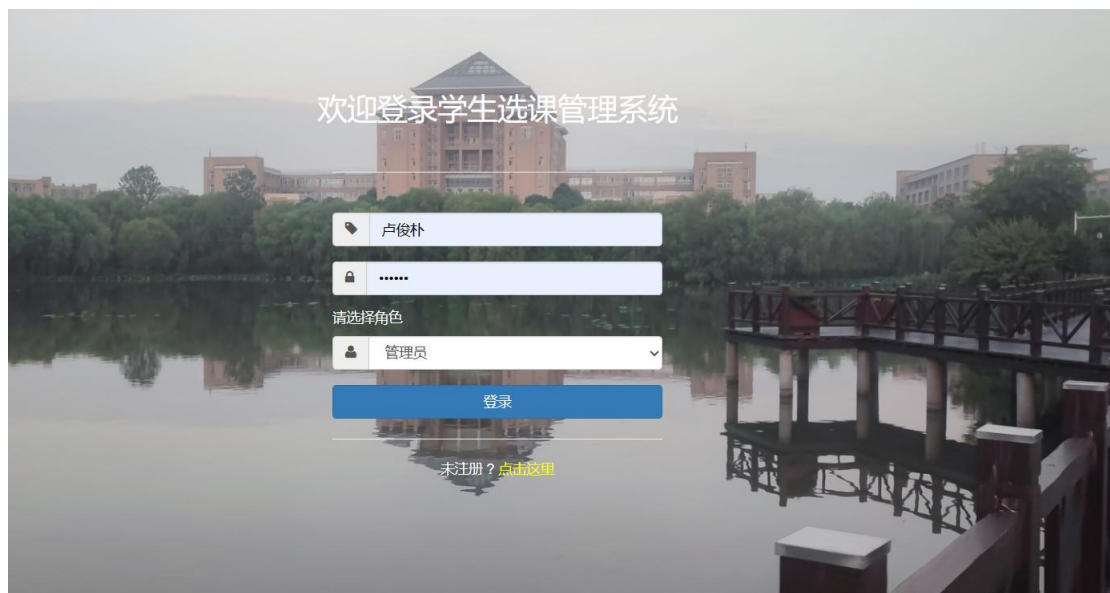
开发工具：Git, Spring MVC, Vue2, Mybatis4.1.5, JPA, Maven, Gradle, Bootstrap

编程语言：java1.8

其他语言：HTML, SQL, JavaScript

五、整体运行展示

管理员



点击登录：

学生选课管理系统

欢迎您, 卢俊朴

信息管理

管理员信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

个人信息

姓名

卢俊朴

性别

☒男☐女

年龄

20

电话

520520520520

保存

学生选课管理系统

欢迎您, 卢俊朴

信息管理

管理员信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

管理员信息

新增

请输入搜索内容

| 姓名 | 性别 | 年龄 | 电话 | 操作 |
|--------|----|----|---------------|-----------------|
| 胡婷 | 女 | 20 | 520520520520 | <div>编辑删除</div> |
| 卢俊朴 | 男 | 20 | 520520520520 | <div>编辑删除</div> |
| admin | 男 | 21 | 18877778888 | <div>编辑删除</div> |
| admin1 | 男 | 12 | 2131241241278 | <div>编辑删除</div> |
| 卢俊辰 | 男 | 10 | 24445568778 | <div>编辑删除</div> |

上一页

1

2

下一页

点击增加/编辑：

管理员信息

新增

姓名

胡婷

卢俊朴

admin

admin1

卢俊辰

性别

男

男

男

男

男

年龄

20

20

21

12

10

电话

520520520520

520520520520

18877778888

2131241241278

24445568778

操作

编辑删除

编辑删除

编辑删除

编辑删除

编辑删除

请填写信息

姓名

卢俊朴

性别

☒男☐女

年龄

20

电话

520520520520

关闭

保存

搜索：

管理员信息

新增

卢

| 姓名 | 性别 | 年龄 | 电话 | 操作 |
|-----|----|----|--------------|-----------------|
| 卢俊朴 | 男 | 20 | 520520520520 | <div>编辑删除</div> |
| 卢俊辰 | 男 | 10 | 24445568778 | <div>编辑删除</div> |

上一页

1

下一页

各个模块：

学生选课管理系统

欢迎您，卢俊朴

信息管理

管理员信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

教师信息

新增

请输入搜索内容

| 用户名 | 性别 | 年龄 | 职称 | 负责专业 | 操作 |
|-----|----|----|-----|--------|-----------------|
| 卢老师 | 男 | 33 | 教授 | 软件工程 | <div>编辑删除</div> |
| 古月婷 | 女 | 34 | 副教授 | 德语 | <div>编辑删除</div> |
| 杨松 | 男 | 31 | 副教授 | 软件工程 | <div>编辑删除</div> |
| 王小明 | 男 | 55 | 院长 | 电子信息工程 | <div>编辑删除</div> |
| 晓黄仁 | 女 | 42 | 教授 | 古汉语信息 | <div>编辑删除</div> |

上一页

1

2

下一页

学生选课管理系统

欢迎您，卢俊朴

信息管理

管理员信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

学生信息

新增

请输入搜索内容

| 用户名 | 性别 | 年龄 | 学号 | 学分 | 所属学院 | 操作 |
|-----|----|----|--------------|----|--------|-----------------|
| 张三 | 男 | 24 | 20000 | 0 | 信息工程学院 | <div>编辑删除</div> |
| 李四 | 男 | 23 | 20001 | 0 | 计算机学院 | <div>编辑删除</div> |
| 王五 | 男 | 23 | 20002 | 0 | 软件学院 | <div>编辑删除</div> |
| 赵六 | 男 | 23 | 20003 | 0 | 外国语学院 | <div>编辑删除</div> |
| 卢同学 | 男 | 20 | 202113407120 | 3 | 计算机学院 | <div>编辑删除</div> |

上一页

1

2

下一页

学生选课管理系统

欢迎您, 卢俊朴

信息管理

管理员工信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

学院信息

新增

请输入学院名称

| 学院id | 学院名称 | 学院介绍 | 学分限制 | 操作 |
|------|--------|--------------|------|-----------------------------|
| 1 | 信息工程学院 | 这是一条描述这是一条描述 | 100 | <div>编辑</div> <div>删除</div> |
| 2 | 计算机学院 | 这是一条描述这是一条描述 | 120 | <div>编辑</div> <div>删除</div> |
| 3 | 软件学院 | 这是一条描述这是一条描述 | 120 | <div>编辑</div> <div>删除</div> |
| 4 | 外国语学院 | 这是一条描述这是一条描述 | 60 | <div>编辑</div> <div>删除</div> |
| 7 | 文学院 | 这是一条描述这是一条描述 | 60 | <div>编辑</div> <div>删除</div> |
| 8 | 体育学院 | 这是一条描述这是一条描述 | 50 | <div>编辑</div> <div>删除</div> |

学生选课管理系统

欢迎您, 卢俊朴

信息管理

管理员工信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

专业信息

新增

请输入专业名称

| 专业id | 专业名称 | 系名 | 所属学院 | 操作 |
|------|--------|-----|--------|-----------------------------|
| 8 | 电子信息工程 | 电子系 | 信息工程学院 | <div>编辑</div> <div>删除</div> |
| 9 | 软件工程 | 工科系 | 计算机学院 | <div>编辑</div> <div>删除</div> |
| 12 | 古汉语信息 | 文学系 | 文学院 | <div>编辑</div> <div>删除</div> |
| 13 | 网络安全 | 工科系 | 计算机学院 | <div>编辑</div> <div>删除</div> |
| 14 | 德语 | 语言系 | 外国语学院 | <div>编辑</div> <div>删除</div> |
| 15 | 英语 | 语言系 | 外国语学院 | <div>编辑</div> <div>删除</div> |

学生选课管理系统

欢迎您, 卢俊朴

信息管理

管理员工信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

课程信息

新增

请输入搜索内容

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所学专业 | 开班人数 | 上课时段 | 上课地点 | 已选人数 | 操作 |
|----------|--------------|------|------|-------|------|--------|-------|------|-----------------------------|
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 2 | <div>编辑</div> <div>删除</div> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 3 | <div>编辑</div> <div>删除</div> |
| Python基础 | 学会Python基础 | 3 | 杨松 | 软件工程 | 15 | 周二第一大节 | 11213 | 2 | <div>编辑</div> <div>删除</div> |
| 德语商务 | 学会跟德国人谈生意 | 3 | 古月婷 | 德语 | 5 | 周三第二大节 | 30402 | 1 | <div>编辑</div> <div>删除</div> |
| 古代晋语 | 教你元谋人怎么书画 | 2 | 晓黄仁 | 古汉语信息 | 5 | 周四第三大节 | 33401 | 2 | <div>编辑</div> <div>删除</div> |
| 商务英语 | 各种商务场景英语交流技巧 | 3 | 杜坤 | 英语 | 10 | 周五第六大节 | 22301 | 3 | <div>编辑</div> <div>删除</div> |
| 网络安全概论 | 这是一条课程介绍 | 1 | 卢老师 | 网络安全 | 10 | 周五第三大节 | 11202 | 1 | <div>编辑</div> <div>删除</div> |

学生选课管理系统

欢迎您, 卢俊朴

信息管理

管理员信息

教师信息

学生信息

学院信息

专业信息

课程信息

选课信息

个人信息

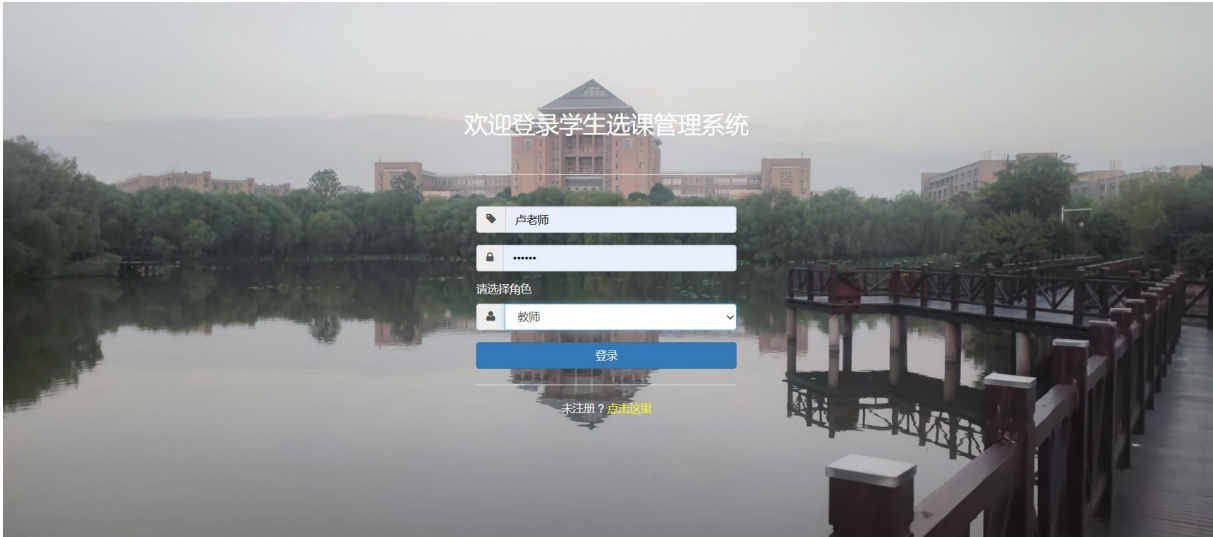
修改密码

退出登录

选课信息

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 选课学生 | 课程状态 |
|----------|--------------|------|------|-------|------|--------|-------|------|------|
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 李四 | 待开课 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 20 | 周一第一大节 | 11201 | 卢同学 | 已结课 |
| 德语商务 | 学会跟德国人谈生意 | 3 | 古月婷 | 德语 | 5 | 周三第二大节 | 30402 | 胡婷 | 待开课 |
| 商务英语 | 各种商务场景英语交流技巧 | 3 | 杜啤 | 英语 | 10 | 周五第六大节 | 22301 | 卢同学 | 待开课 |
| Python基础 | 学会Python基础 | 3 | 杨松 | 软件工程 | 15 | 周二第一大节 | 11213 | 卢同学 | 待开课 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 小明 | 待开课 |
| Python基础 | 学会Python基础 | 3 | 杨松 | 软件工程 | 15 | 周二第一大节 | 11213 | 小明 | 待开课 |
| 商务英语 | 各种商务场景英语交流技巧 | 3 | 杜啤 | 英语 | 10 | 周五第六大节 | 22301 | 卢同学 | 待开课 |
| 古代汉语 | 教你元谋人怎么书画 | 2 | 晓黄仁 | 古汉语信息 | 5 | 周四第三大节 | 33401 | 卢同学 | 待开课 |
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 卢同学 | 待开课 |
| 古代汉语 | 教你元谋人怎么书画 | 2 | 晓黄仁 | 古汉语信息 | 5 | 周四第三大节 | 33401 | 胡婷 | 待开课 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 胡婷 | 已开课 |
| 网络安全概论 | 这是一条课程介绍 | 1 | 卢老师 | 网络安全 | 10 | 周五第三大节 | 11202 | 卢同学 | 待开课 |

教师



点击登录

学生选课管理系统

欢迎您, 卢老师

信息管理

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

个人信息

姓名

卢老师

性别

☒ 男 ☐ 女

年龄

33

专业id

9

职称

教授

保存

各个模块：

学生选课管理系统

欢迎您, 卢老师

信息管理

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

学院信息

请输入学院名称

| 学院id | 学院名称 | 学院介绍 | 学分限制 |
|------|--------|--------------|------|
| 1 | 信息工程学院 | 这是一条描述这是一条描述 | 100 |
| 2 | 计算机学院 | 这是一条描述这是一条描述 | 120 |
| 3 | 软件学院 | 这是一条描述这是一条描述 | 120 |
| 4 | 外国语学院 | 这是一条描述这是一条描述 | 60 |
| 7 | 文学院 | 这是一条描述这是一条描述 | 60 |
| 8 | 体育学院 | 这是一条描述这是一条描述 | 50 |

学生选课管理系统

欢迎您, 卢老师

信息管理

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

专业信息

请输入专业名称

| 专业id | 专业名称 | 系名 | 所属学院 |
|------|--------|-----|--------|
| 8 | 电子信息工程 | 电子系 | 信息工程学院 |
| 9 | 软件工程 | 工科系 | 计算机学院 |
| 12 | 古汉语信息 | 文学系 | 文学院 |
| 13 | 网络安全 | 工科系 | 计算机学院 |
| 14 | 德语 | 语言系 | 外国语学院 |
| 15 | 英语 | 语言系 | 外国语学院 |

学生选课管理系统

欢迎您, 卢老师

信息管理

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

课程信息

请输入搜索内容

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所学专业 | 开班人数 | 上课时段 | 上课地点 | 已选人数 |
|----------|--------------|------|------|-------|------|--------|-------|------|
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 2 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 3 |
| Python基础 | 学会Python基础 | 3 | 杨松 | 软件工程 | 15 | 周二第一大节 | 11213 | 2 |
| 德语商务 | 学会跟德国人谈生意 | 3 | 古月婷 | 德语 | 5 | 周三第二大节 | 30402 | 1 |
| 古代鲁迅 | 教你元谋人怎么书画 | 2 | 晓黄仁 | 古汉语信息 | 5 | 周四第三大节 | 33401 | 2 |
| 商务英语 | 各种商务场景英语交流技巧 | 3 | 杜啤 | 英语 | 10 | 周五第六大节 | 22301 | 3 |
| 网络安全概论 | 这是一条课程介绍 | 1 | 卢老师 | 网络安全 | 10 | 周五第三大节 | 11202 | 1 |

（只显示自己任课的选课信息）点击开课，学生不可退课，点击结课，学生学分增加

学生选课管理系统

欢迎您, 卢老师

信息管理

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

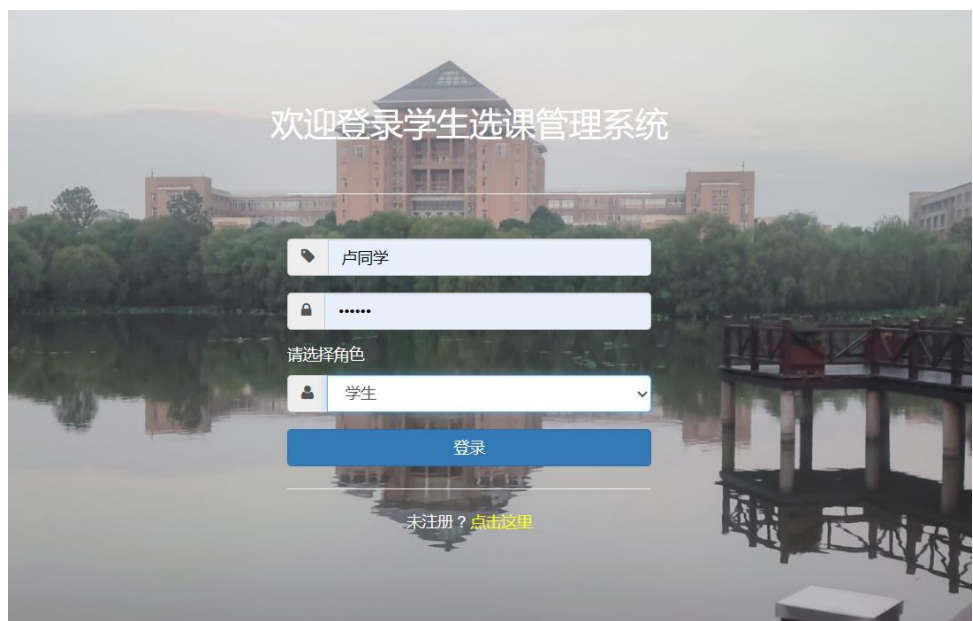
选课信息

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所学专业 | 开班人数 | 上课时段 | 上课地点 | 选课学生 | 课程状态 | 操作 |
|--------|------------|------|------|------|------|--------|-------|------|------|---|
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 20 | 周一第一大节 | 11201 | 卢同学 | 已结课 | <button>开课</button> <button>结课</button> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 小明 | 待开课 | <button>开课</button> <button>结课</button> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 胡婷 | 已开课 | <button>开课</button> <button>结课</button> |
| 网络安全概论 | 这是一条课程介绍 | 1 | 卢老师 | 网络安全 | 10 | 周五第二大节 | 11202 | 卢同学 | 待开课 | <button>开课</button> <button>结课</button> |

选课信息

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 选课学生 | 课程状态 | 操作 | |
|--------|------------|-----------------|------|------|------|--------|--------|-------|------|---|---|
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 20 | 周一第一大节 | 11201 | 卢同学 | 已结课 | <button>开课</button> <button>结课</button> | |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 小明 | 待开课 | <button>开课</button> <button>结课</button> | |
| Java基础 | 学会Java语言基础 | <div>系统提示</div> | | | | | 周一第一大节 | 11201 | 胡婷 | 已开课 | <button>开课</button> <button>结课</button> |
| 网络安全概论 | 这是一条课程介绍 | | | | | | 周三三大节 | 11202 | 卢同学 | 待开课 | <button>开课</button> <button>结课</button> |

学生



点击登录

学生选课管理系统

信息管理

学院信息

专业信息

课程信息

选课信息

个人信息

修改密码

退出登录

欢迎您, 卢同学

个人信息

姓名

卢同学

性别

☒男
 ☐女

年龄

20

学号

202113407120

学院Id

2

总学分

3

保存

学院信息和专业信息的显示和教师一样，重复选课会提示



| 课程名称 | 课程介绍 | ● 你已经选过了 | | | 班人数 | 上课时段 | 上课地点 | 已选人数 | 操作 | |
|----------|--------------|----------|-----|------|-----|--------|-------|------|----|----|
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 2 | 选课 | 退课 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 3 | 选课 | 退课 |
| Python基础 | 学会Python基础 | 3 | 杨松 | 软件工程 | 15 | 周二第一大节 | 11213 | 2 | 选课 | 退课 |

六、具体实现

说明：

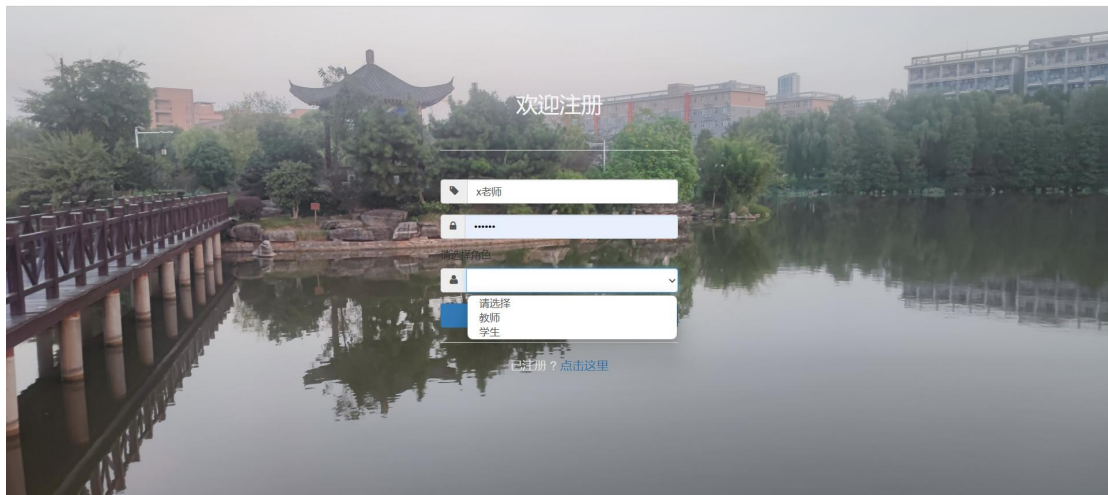
- 以下各个模块代码所展示的运行图片是系统完成后的，展示的代码都是实现当前模块的代码（全部展示字数太多了），各个模块的代码会在之后描述的模块进行扩充和完善。
- 此系统有很多相近重复的操作，比如登录模块有三个类型用户的登录，类似这种重复或变化很小的模块，只列举一个。

（一）注册模块

以教师注册为例

运行展示

只能注册老师和学生，管理员无法注册。点击注册后会跳转登录页面(之后模块有展示)



实现步骤

1. 前端注册提交按钮绑定使用 `axios.post("/register", this.user)` 的方法，传入与注册输入信息双向绑定的数据到控制器
2. 控制器用通用父类接收注册信息，用 `BeanUtils.copyProperties` 将其转换成子类对象，调用对应注册用户的 Service 方法 `register`
3. `register` 获得传入实参，先判断是否有同名，再调用 dao 层 Mybatis 提供的插入数据库方法完成注册

涉及主要代码

以下主要代码前序号为对应的实现步骤

1,register.html

```
1. <form role="form">
2.   <hr/>
3.   <br/>
4.   <div class="form-group input-group">
5.     <span class="input-group-addon"><i class="fa fa-tag"></i></span>
6.     <input type="text" class="form-control" v-model="user.name" placeholder
       ="用户名"/>
7.   </div>
8.   <div class="form-group input-group">
9.     <span class="input-group-addon"><i class="fa fa-lock"></i></span>
10.    <input type="password" class="form-control" v-model="user.password" placeholder="密码"/>
11.  </div>
```

```

12. <h5>请选择角色</h5>
13. <div class="form-group input-group">
14.   <span class="input-group-addon"><i class="fa fa-user"></i></span>
15.   <select class="form-control" v-model="user.level">
16.     <option value="" selected="">请选择</option>
17.     <option value="2">教师</option>
18.     <option value="3">学生</option>
19.   </select>
20. </div>
21.
22. <div style="text-align: center">
23.   <a href="javascript:void(0)" class="btn btn-block btn-primary" @click="re
    gister()">注册</a>
24. </div>
25. <hr/>
26. <div style="text-align: center; color:whitesmoke; font-size: larger">
27.   <span>已注册 ?</span> <a href="login.html">点击这里 </a>
28. </div>
29. </form>
30. new Vue({
31.   el: "#app",
32.   data: {
33.     user: {},
34.   },
35.   created: function () {
36.   },
37.   methods: {
38.     register() {
39.       axios.post("/register", this.user).then(res => {
40.         if (res.data.code === '0') {
41.           window.location = "/end/page/login.html";
42.         } else {
43.           alert(res.data.msg);
44.         }
45.       })
46.     },
47.   }
48. });

```

2,AccountController()

```

1. @PostMapping("/register")

```



```

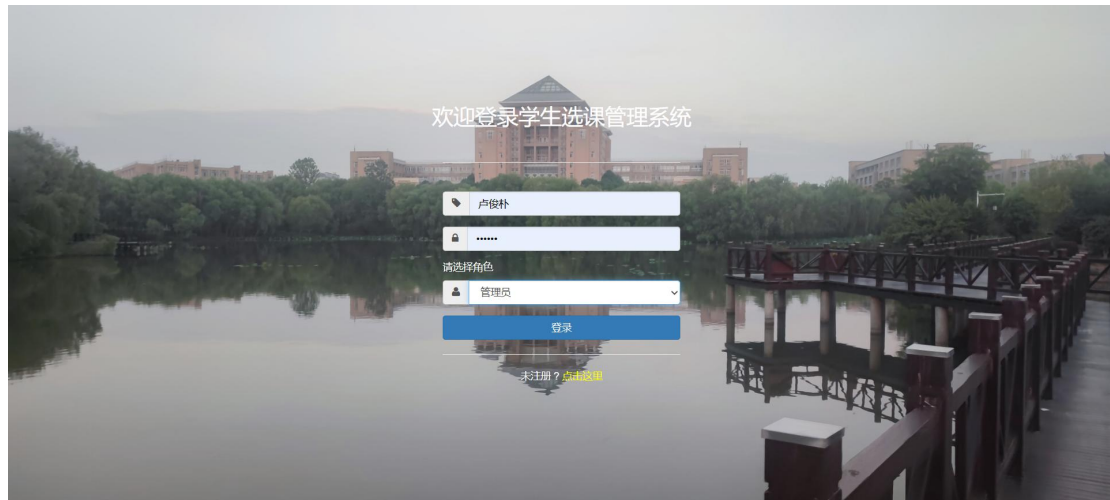
2.     public Result register(@RequestBody Account user, HttpServletRequest request) {
3.         if (ObjectUtil.isEmpty(user.getName()) || ObjectUtil.isEmpty(user.getPassword()) || ObjectUtil.isEmpty(user.getLevel())) {
4.             return Result.error("-1", "输入的信息不完善");
5.         }
6.         Integer level = user.getLevel();
7.         if (2 == level) {
8.             //教师注册
9.             TeacherInfo teacherInfo = new TeacherInfo();
10.            //      teacherInfo = (TeacherInfo) user;
11.            BeanUtils.copyProperties(user, teacherInfo); //BeanUtils.copyProperties("转换前的类", "转换后的类");
12.            teacherInfoService.resigter(teacherInfo);
13.        }
14.        if (3 == level) {
15.            //学生注册
16.            .....
17.        }
18.        TeacherInfoService
19.        @Service
20.        public class TeacherInfoService {
21.            @Resource
22.            private TeacherInfoDao teacherInfoDao;
23.            public void resigter(TeacherInfo teacherInfo) {
24.                //1.检查数据库中有没有同名的教师，如果有需要提示前台用户重新输入
25.                TeacherInfo info = teacherInfoDao.findByName(teacherInfo.getName());
26.                if (ObjectUtil.isNotEmpty(info)) {
27.                    //已经有教师用户了
28.                    throw new CustomException(ResultCode.USER_EXIST_ERROR); //如果用户名一样，抛出异常，阻断登录，阻止数据库新增，提示用户名存在
29.                }
30.                //2.如果没有，说明可以注册，向数据库里新增一条数据
31.                teacherInfoDao.insertSelective(teacherInfo);
32.            }
33.        }

```

（二）登录模块（以管理员 Admininfo 的的登录为例）

登录系统，根据数据库数据，进入不同类型用户所属界面，

运行展示



实现步骤

1. 前端输入用户名密码登录，前端把输入的信息 axios.post 给 /login (AccountController 内)
2. AccountController 接收信息，检查登录位置（看 level）如果是 1-> 登录 AdminInfo->
3. 调用 AdminInfoService 的 login 方法->
4. AdminInfoService 的 login 通过使用 AdminInfoDao 对象的方法，查看是否是在数据库中的数据->
5. 是则通过 dao 中 SQL 语句从数据库里找出所有数据传输到 AccountController，通过 request.getSession().setA.....存一份信息给后序模块使用->
6. 经过 Result<>返回符合的 account 信息给前端->
7. 前端 vue 接收后，通过 Result 中的 Code 判断是否成功->
8. 通过 level 判断是哪类用户登陆的就进入哪个用户类型的 html 页面

涉及主要代码

解释

以下主要代码前序号为对应的实现步骤

1, 7, 8, login.xml

```
1.    <form role="form">
2.        <hr/>
3.        <br/>
4.        <div class="form-group input-group">
5.            <span class="input-group-addon"><i class="fa fa-tag"></i></span>
6.            <input type="text" class="form-control" v-model="user.name" placeholder="用户名"/>
7.        </div>
8.        <div class="form-group input-group">
9.            <span class="input-group-addon"><i class="fa fa-lock"></i></span>
10.           <input type="password" class="form-control" v-model="user.password" placeholder="密码"/>
11.        </div>
12.        <h5 style="color: white">请选择角色</h5>
13.        <div class="form-group input-group">
14.            <span class="input-group-addon"><i class="fa fa-user"></i></span>
15.            <select class="form-control" v-model="user.level">
16.                <option value="" selected="">请选择</option>
17.                <option value="1">管理员</option>
18.                <option value="2">教师</option>
19.                <option value="3">学生</option>
20.            </select>
21.        </div>
22.        <div style="text-align: center">
23.            <a href="javascript:void(0)" class="btn btn-block btn-primary" @click="login">登录</a>
24.        </div>
25.        <script>
26.            new Vue({
27.                el: "#app",
28.                data: {
29.                    user: {} //接收前端输入的值
30.                },
31.                created: function () {
```

```

32.      .....
33.    },
34.    methods: {
35.      login() {
36.        axios.post("/login", this.user).then(res => {
37.          if (res.data.code === '0') {
38.            let user = res.data.data;
39.            localStorage.setItem('user', JSON.stringify(user));
40.            if (user && user.level && 1 === user.level) {
41.              window.location = "/end/page/accountAdminInfo.html";
42.            }
43.            if (user && user.level && 2 === user.level) {
44.              window.location = "/end/page/accountTeacherInfo.html";
45.            }
46.            if (user && user.level && 3 === user.level) {
47.              window.location = "/end/page/accountStudentInfo.html";
48.            }
49.          } else {
50.            alert(res.data.msg);
51.          }
52.        })
53.      }
54.    }
55.  });
56. </script><input type="text">

```

2, 6, AccountController

```

1.  @Resource
2.  private AdmininfoService admininfoService;
3.  @PostMapping("/login")
4.  public Result login(@RequestBody Account user,HttpServletRequest request){
5.    //校验数据是否填
6.    if(ObjectUtil.isEmpty(user.getName())||ObjectUtil.isEmpty(user.getPassword())||
7.      ObjectUtil.isEmpty(user.getLevel())){
8.      return Result.error("-1","输入的信息不完善");
9.    }
10.   //都有值
11.   Integer level = user.getLevel();

```

```

12.     Account loginUser = new Account();//用来统一接收登录信息
13.     if(1 == level){//管理员登陆,去管理员信息表查询,调用管理员 Service
        的业务代码
14.         loginUser=adminInfoService.login(user.getName(),user.getPassword())
        ;
15.     }
16.     if(2 == level){//教师登陆,去教师信息表查询
17.
18.     }
19.     if(3 == level){//学生登陆,去学生信息表查询
20.
21.     }
22.     //在 session 里面把用户信息存一份
23.     request.getSession().setAttribute("user",loginUser);
24.
25.     return Result.success(loginUser);
26.     //这里把登陆的用户信息返回到前端
27. }

```

3, 4, AdminInfoService

```

1.     @Resource
2.     private AdminInfoDao adminInfoDao;
3.
4.     public Account login(String name,String password) {
5.         //通过用户名和密码去数据库里查一条记录出来
6.         AdminInfo adminInfo = adminInfoDao.findByNameAndPass(name,password);
7.
8.         if(ObjectUtil.isEmpty(adminInfo)) {
9.             throw new CustomException("-1","用户名, 密码或角色错误");
10.        }
11.        return adminInfo;
12.    }

```

5, AdminInfoDao

```

1.     @Repository
2.     public interface AdminInfoDao extends Mapper<AdminInfo> {
3.
4.         @Select("select * from admin_info where name = #{name} and password =
        #{password}")

```

```

5.     AdminInfo findByNameAndPass(@Param("name") String name,@Param("
    password") String password);
6.
7.     }

```

6, Result

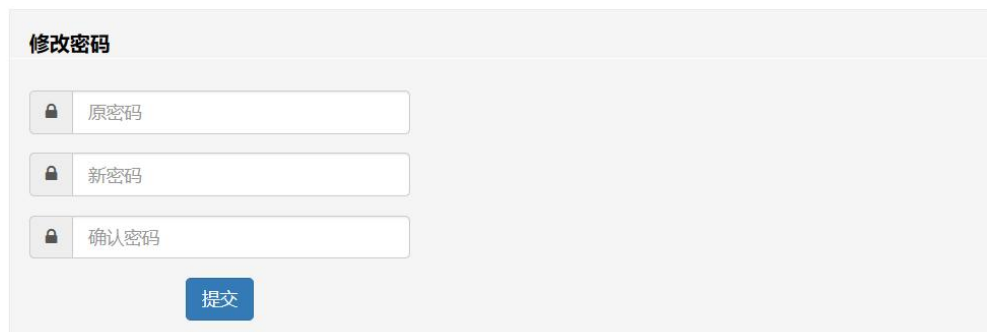
```

1.     public class Result<T> { //Result 是统一返回前端的实体类，用到了统一返
    回编程，统一返回到前端，前端通过 code 判断是否接受请求
2.         private String code;
3.         private String msg;
4.         private T data;
5.
6.         private Result(T data) {
7.             this.data = data;
8.         }
9.
10.        public Result() {
11.        }
12.        .....
13.        public static Result success() {
14.            Result tResult = new Result<>();
15.            tResult.setCode(ResultCode.SUCCESS.code);
16.            tResult.setMsg(ResultCode.SUCCESS.msg);
17.            return tResult;
18.        }
19.        public static <T> Result<T> success(T data) {
20.            Result<T> tResult = new Result<>(data);
21.            tResult.setCode(ResultCode.SUCCESS.code);
22.            tResult.setMsg(ResultCode.SUCCESS.msg);
23.            return tResult;
24.        }
25.        public static Result error() {
26.            Result tResult = new Result<>();
27.            tResult.setCode(ResultCode.ERROR.code);
28.            tResult.setMsg(ResultCode.ERROR.msg);
29.            return tResult;
30.        }
31.        .....get set 方法
32.    }

```

（三）修改密码模块

运行展示



The screenshot shows a web form titled "修改密码" (Modify Password). It contains three input fields, each with a lock icon on the left: "原密码" (Original Password), "新密码" (New Password), and "确认密码" (Confirm Password). Below the fields is a blue button labeled "提交" (Submit).



修改成功后会跳转至

实现步骤

前端先获取信息，并双向绑定输入,当原始密码正确以及两次新密码输入一致时，进行 `axios.put` 更新信息（调用的更新方法都在之前模块用过）

涉及主要代码

updatePassword.html

```
1. <form role="form" style="width: 300px">
2.   <input type="hidden" id="id" name="id" v-model="entity.id">
3.   <div class="form-group input-group">
4.     <span class="input-group-addon"><i class="fa fa-lock"></i></span>
5.     <input type="password" class="form-control" v-model="password"
6.       placeholder="原密码"/>
7.   </div>
8.
9.   <div class="form-group input-group">
10.    <span class="input-group-addon"><i class="fa fa-lock"></i></span>
11.    <input type="password" class="form-control" v-model="newPassword"
12.      placeholder="新密码"/>
13.   </div>
14.
15.   <div class="form-group input-group">
16.    <span class="input-group-addon"><i class="fa fa-lock"></i></span>
17.    <input type="password" class="form-control" v-model="new2Password"
18.      placeholder="确认密码"/>
19.   </div>
20.   <div style="text-align: center">
21.     <a href="javascript:void(0)" @click="updatePassword()"
22.       class="btn btn-primary">提交</a>
23.   </div>
24. </form>
25. new Vue({
26.   .....
27.   el: '#wrapper',
28.   data: {
29.     user: {},
30.     entity: {},
31.     password: "",
32.     newPassword: "",
33.     new2Password: "",
34.   },
35.   .....
36.   methods: {
37.     loadData() { //显示登录信息
38.       axios.get("/getUser").then(res => {
39.         if (res.data.code === "0") {
```

```

40.         this.entity = res.data.data;
41.     } else {
42.         msg("error", res.data.msg);
43.     }
44. })
45. },
46. updatePassword() {
47.     entity = this.entity;
48.     if (entity.password !== this.password) {
49.         alert('原始密码不正确。');
50.         return;
51.     }
52.     else if(this.newPassword !== this.new2Password) {
53.         alert('新密码和确认密码不匹配，请重新输入。');
54.         return;
55.     }
56.     entity.password = this.newPassword;
57.     if(entity.level === 1){
58.         axios.put("/adminInfo",entity).then(res => {
59.             if (res.data.code === '0') {
60.                 msg("success", "密码修改成功，请重新登录");
61.                 window.location = "/end/page/login.html";
62.             } else {
63.                 msg("error", res.data.msg);
64.             }
65.         })
66.     }else if(entity.level === 2){
67.         axios.put("/teacherInfo",entity).then(res => {
68.             if (res.data.code === '0') {
69.                 msg("success", "密码修改成功，请重新登录");
70.                 window.location = "/end/page/login.html";
71.             } else {
72.                 msg("error", res.data.msg);
73.             }
74.         })
75.     }else if(entity.level === 3){
76.         axios.put("/studentInfo",entity).then(res => {
77.             if (res.data.code === '0') {
78.                 msg("success", "密码修改成功，请重新登录");
79.                 window.location = "/end/page/login.html";
80.             } else {
81.                 msg("error", res.data.msg);
82.             }
83.         })

```



```

84.     }
85.     }
86.   },
87. })

```

（四）登出模块

实现步骤

前端登出按钮绑定登出方法，在 my.js 实现登出方法，axios.get 到控制器控制器清除登陆后传入的 HttpServletRequest 对象数据，然后跳转登入页面。再在各个有登出按钮的前端引入 my.js，直接调用登出方法。

涉及主要代码

my.js

```

1.  function logout() {
2.    axios.get("/logout").then(res => {
3.      if (res.data.code === '0') {
4.        window.location = "/end/page/login.html";
5.      } else {
6.        msg("error", res.data.msg);
7.      }
8.    })
9.  }

```

其他各个需要用到登出方法的 vue

```

1.  methods: {
2.    .....
3.    logout() {
4.      logout();//在 my.js 内
5.    }
6.  }

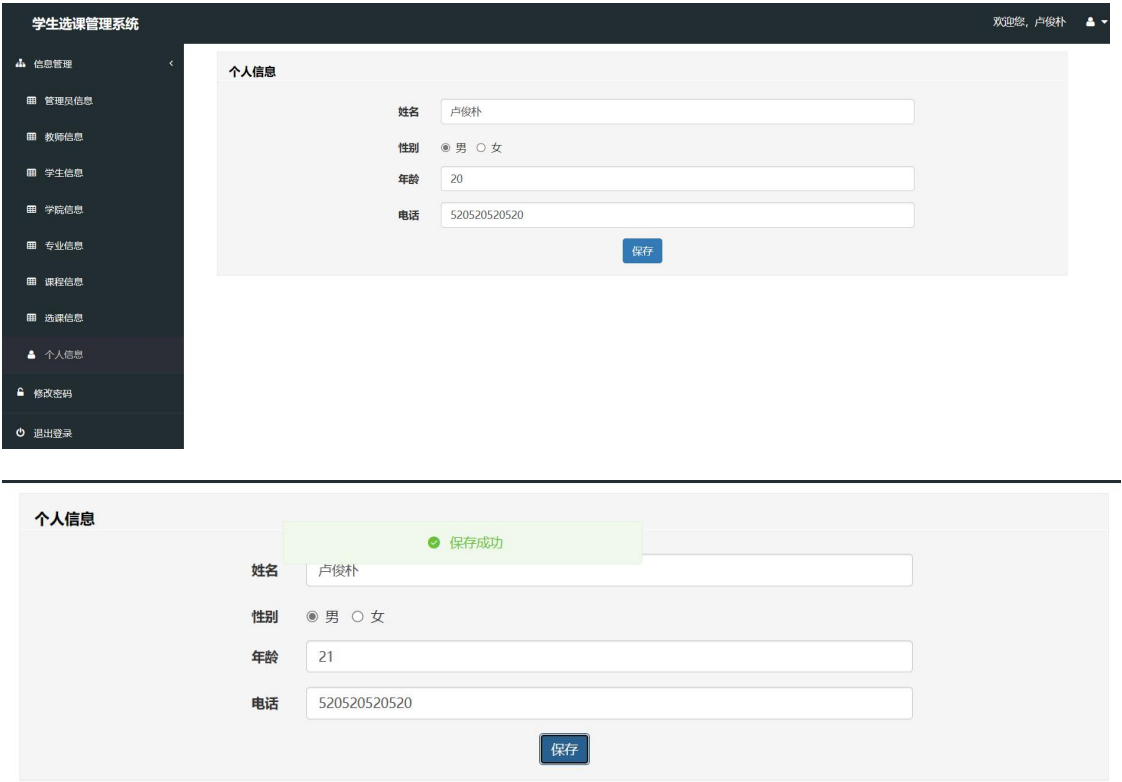
```

（五）个人主页模块实现（以 AdminInfo 为例）

显示当前登录用户的可见数据，并可以对自己的账户信息进行更改。

每种类型的用户都有这个模块，差别很小，这里只展示 AdminInfo 的个人主页模块。

运行展示



实现步骤

显示当前用户数据

1. 先把主页实现前端写好（accountAdminInfo.html）
2. 在 vue 实现 loadData 方法（用 axios.get（“/getUser”）.then），在 AccountController 中获取之前登录模块预留的登录信息（是 Account 父实体类）
3. 相似 AccountController 的 GetMapping（“/getUser”）获得 Account 的 level，判断登录用户类型
4. 通过 id，调用对应登陆类型的 Service 方法，对应类型 Service 方法使用对应类型的 dao 内 SQL
5. 最后返回值给控制器的 getUser

6. 然后由于前台的 `axios.get`，返回给前端，前端根据 `get` 的值，进行显示

在个人列表更新保存数据

7. 在保存按钮增加 `@ click` 点击事件，创建 `update ()`，在其中增加保存的 `axios.put ("/admininfo")` 语句，把所有前端写入的值传给 `AdminInfoController` 控制器的 `RequestMapping ("/admininfo")`

8. 在控制器中的 `@PutMapping` 注释的 `update` 方法调用 `admininfoService` 方法

9. 由于是基本的数据库更改，直接使用 `@ Resource` 获取 `bean` 后的 `AdminInfoDao` 对象的内置方法

实现步骤涉及主要代码

以下主要代码前序号为对应的实现步骤

1, 2, 6, accountAdminInfo.html 主要代码

```
1. <div class="panel-body">
2.   <form class="form-horizontal">
3.     <input type="hidden" v-model="entity.id">
4.
5.     <div class="form-group">
6.       <label class="col-sm-3 control-label">姓名</label>
7.       <div class="col-sm-7">
8.         <input type="text" class="form-control" v-model="entity.name"
9.           placeholder="请输入姓名">
10.      </div>
11.    </div>
12.
13.    <div class="form-group">
14.      <label class="col-sm-3 control-label">性别</label>
15.      <div class="col-sm-7">
16.        <label class="radio-inline">
17.          <input type="radio" value="男" v-model="entity.sex">男
18.        </label>
19.        <label class="radio-inline">
20.          <input type="radio" value="女" v-model="entity.sex">女
21.        </label>
22.      </div>
23.    </div>
```

```

24.
25.   <div class="form-group">
26.     <label class="col-sm-3 control-label">年龄</label>
27.     <div class="col-sm-7">
28.       <input type="text" class="form-control" v-model="entity.age"
29.         placeholder="请输入年龄">
30.     </div>
31.   </div>
32.
33.   <div class="form-group">
34.     <label class="col-sm-3 control-label">电话</label>
35.     <div class="col-sm-7">
36.       <input type="text" class="form-control" v-model="entity.phone"
37.         placeholder="请输入电话">
38.     </div>
39.   </div>
40. </form>
41. <div style="text-align: center">
42.   <button class="btn btn-primary">保存</button>
43. </div>
44. </div>
45. <script>
46.   function msg(type, msg) {
47.     Vue.prototype.$message({
48.       type: type, // success (成功)、warning (警告)、error(错误)
49.       message: msg,
50.       duration: 2000,
51.       offset: 100,
52.       center: true
53.     })
54.   }
55.
56.   new Vue({
57.     el: '#wrapper',
58.     data: {
59.       user: {},
60.       entity: {},
61.     },
62.     created: function () {
63.       .....
64.       this.user = JSON.parse(localStorage.getItem('user'));
65.       this.loadData();//页面加载时调用 获取 登录对象的方法
66.     },
67.     methods: {

```

```

68.         loadData() {
69.             axios.get("/getUser").then(res => {
70.                 if(res.data.code === "0"){
71.                     this.entity = res.data.data;
72.                 }else{
73.                     msg("error",res.data.msg);
74.                 }
75.             })
76.         },
77.         logout() {
78.         }
79.     }
80. })
81. </script>

```

3, 4, 5, 6 AccountController 主要代码

```

1.  @GetMapping("/getUser")
2.  public Result getUser(HttpServletRequest request){
3.      //先从 session 里面获取当前存的用户登录的信息
4.      Account user = (Account) request.getSession().getAttribute("user");
5.      //判断当前登录的用户是什么角色
6.      Integer level = user.getLevel();
7.      if(1 == level){//管理员登陆,去管理员信息表查询,调用管理员 Service 的业
        务代码
8.          //获取管理员信息
9.          AdminInfo adminInfo = adminInfoService.findById(user.getId());//通过登
        录模块保存的主键,在 AdminInfo 服务和 Dao 获取表中 SQL 管理员信息
10.         return Result.success(adminInfo);
11.     }
12.     if(2 == level){//教师登陆,去教师信息表查询
13.         //从教师表里面获取教师信息
14.     }
15.     if(3 == level){//学生登陆,去学生信息表查询
16.         //从学生表里获取学生信息
17.     }
18.     return Result.success(new Account());
19. }

```

4, AdminInfoService 主要方法

```

1.  @Resource
2.  private AdminInfoDao adminInfoDao;

```

```

3. ....
4. public AdminInfo findById(Long id){
5.     return adminInfoDao.selectByPrimaryKey(id);
6. }

```

解释

由于 AdminInfoDao 接口继承了 Mapper<AdminInfo>, 所以简单的增删改查以及根据主键查询会自动生成, 通过 @Resource 或者 @Autowired 已经获取了 Bean, 可以直接使用这个 Dao 简单的 SQL 语句方法。

7, accountAdminInfo vue 主要代码

```

1. methods: {
2.     .....
3.     update(){//用 put 方法把整个更改的表格中的信息返回给控制器
4.         axios.put("/adminInfo",this.entity).then(res=>{
5.             if(res.data.code === "0"){
6.                 msg("success","保存成功");
7.                 this.loadData();
8.             }else{
9.                 msg("error",res.data.msg);
10.            }
11.        })
12.    },
13.    .....
14. }

```

8, AdminInfoController 主要代码

```

1. @RestController
2. @RequestMapping("/admininfo")
3. public class AdminInfoController {
4.     @Resource
5.     private AdmininfoService admininfoService;
6.     @PutMapping
7.     public Result update(@RequestBody AdminInfo adminInfo){
8.         admininfoService.update(adminInfo);
9.         return Result.success();
10.    }
11. }

```

9, AdmininfoService 主要代码

```

1. @Resource

```

```

2. private AdminInfoDao adminInfoDao;
3. ....
4. public void update(AdminInfo adminInfo){
5.     adminInfoDao.updateByPrimaryKeySelective(adminInfo);
6. }

```

（六）信息的增删改查(以 AdminInfo 为例)

管理员使用，用来增删改查，统一管理各个模块的信息。用到这个模块的地方非常多，这里只列举 AdminInfo 的增删改查（不然字数太多了）

1. 增加

在 xxx 信息的模块，点击增加，弹出模态框，在里面填入数据后保存，可以增加数据到数据库。此处以管理员信息为例（AdminInfo）。

运行展示

The screenshot displays the 'AdminInfo' management interface. On the left, a sidebar contains a '新增' (Add) button and a list of admin names: '胡婷', '卢俊朴', and 'admin'. The main area shows a modal form titled '请填写信息' (Please fill in the information) with fields for '姓名' (Name), '性别' (Gender), '年龄' (Age), and '电话' (Phone). The '姓名' field contains '增加测试' (Add test), '性别' has radio buttons for '男' (Male) and '女' (Female), '年龄' contains '123', and '电话' contains '123'. At the bottom of the modal are '关闭' (Close) and '保存' (Save) buttons. Below the modal is a table of existing admins with columns for '姓名', '性别', '年龄', '电话', and '操作' (Operations). The table contains two rows: one for 'admin2' (male, age 44, phone 21312412412) and one for '增加测试' (female, age 123, phone 123). Each row has '编辑' (Edit) and '删除' (Delete) buttons. At the bottom of the table are pagination controls: '上一页' (Previous page), '1', '2', and '下一页' (Next page).

增加实现步骤

1. 前端添加模态框，模态框内为输入数据，v-model 和 Vue 内定义的对象双向绑定，增加按钮加函数 add 调用模态框（用于输入添加信息），在模态框的前端提交 @ click 一个 submit() 函数，用于 axios, post 数据给控制器。
2. 控制器调用 AdminInfoService 服务。并且解决初始密码的设置（新增管理员不帮他设置密码），重名问题。

3. 重名问题需要自定义 dao 的 SQL 语句,再自定义一个异常类 CustomException 和 ResultCode 联动使用, 一个 GlobalExceptionHandler 作为异常捕获类, 处理信息。如果重名抛出异常无法添加
4. 服务直接用 Mybatis 框架实现简单的增加功能。

增加涉及主要代码

以下主要代码前序号为对应的实现步骤

1, adminInfo.html

```
1. <div class="modal" tabindex="-1" role="dialog" id="mod">
2.   <div class="modal-dialog" role="document">
3.     <div class="modal-content">
4.       <div class="modal-header">
5.         <span class="modal-title">请填写信息</span>
6.         <button type="button" class="close" data-dismiss="modal" aria-label="Close" @click="entity={}">
7.           <span aria-hidden="true">&times;</span>
8.         </button>
9.       </div>
10.      <div class="modal-body">
11.        <form class="form-horizontal">
12.          <input type="hidden" id="id" name="id" v-model="entity.id">
13.          <div class="form-group">
14.            <label class="col-sm-3 control-label">姓名</label>
15.            <div class="col-sm-7 form-inline">
16.              <input type="text" class="form-control" v-model="entity.name" placeholder="请输入姓名"
17.                style="width: 100%">
18.            </div>
19.          </div>
20.          <div class="form-group">
21.            <label class="col-sm-3 control-label">性别</label>
22.            <div class="col-sm-7">
23.              <label class="radio-inline">
24.                <input type="radio" value="男" v-model="entity.sex">男
25.              </label>
26.              <label class="radio-inline">
27.                <input type="radio" value="女" v-model="entity.sex">女
28.              </label>
29.            </div>
30.          </div>
```



```

31.     <div class="form-group">
32.         <label class="col-sm-3 control-label">年龄</label>
33.         <div class="col-sm-7 form-inline">
34.             <input type="text" class="form-control" v-model="entity.age" placeholder="请输入年龄"
35.                 style="width: 100%">
36.         </div>
37.     </div>
38.     <div class="form-group">
39.         <label class="col-sm-3 control-label">电话</label>
40.         <div class="col-sm-7 form-inline">
41.             <input type="text" class="form-control" v-model="entity.phone" placeholder="请输入电话"
42.                 style="width: 100%">
43.         </div>
44.     </div>
45. </form>
46. </div>
47. <div class="modal-footer">
48.     <button type="button" class="btn btn-secondary" data-dismiss="modal"
49.         @click="entity={} ">关闭
50.     <button type="button" class="btn btn-primary" @click="submit()">保存
51. </button>
52. </div>
53. </div>
54. </div>
55. data: {
56.     user: {}, // 一个对象，很多字段
57.     entity: {}, // 只有一个对象，用于传入前端输入的增加或更新数据的
58.         值给控制器
59.     tableData: [], // 有很多对象，用于接收查询所有
60.     .....
61. },
62. methods: {
63.     add() { // 点击增加跳出模板框
64.         this.entity = {};
65.         $('#mod').modal("show"); // 调用前端模板框
66.     },
67.     submit() { // 提交模板框输入数据到数据库
68.         axios.post("/adminInfo", this.entity).then(res => {
69.             if (res.data.code === "0") {
70.                 $('#mod').modal("hide"); // 隐藏前端模板框

```

```

70.         msg("success","插入成功");
71.     }else{
72.         msg("error",res.data.msg);
73.     }
74. })
75. },
76.     logout() {
77.     }
78. }
79. 2, AdminInfoController
80. @RestController
81. @RequestMapping("/adminInfo")
82. public class AdminInfoController {
83.     @Resource
84.     private AdmininfoService admininfoService;
85.     .....
86.     @PostMapping
87.     public Result add(@RequestBody AdminInfo adminInfo){
88.         admininfoService.add(adminInfo);
89.         return Result.success();
90.     }
91. }

```

2, AdminInfoController

```

1.  @RestController
2.  @RequestMapping("/adminInfo")
3.  public class AdminInfoController {
4.      @Resource
5.      private AdmininfoService admininfoService;
6.      .....
7.      @PostMapping
8.      public Result add(@RequestBody AdminInfo adminInfo){
9.          admininfoService.add(adminInfo);
10.         return Result.success();
11.     }
12.
13. }

```

2,4, AdminInfoService

```
1.  public void add(AdminInfo adminInfo) {
2.      //判断是否重名
3.      AdminInfo info =adminInfoDao.findByName(adminInfo.getName());
4.      if(ObjectUtil.isNotEmpty(info)){//不空就是有重名
5.          throw new CustomException(ResultCode.USER_EXIST_ERROR);//
        如果用户名一样，抛出异常，阻断登录，阻止数据库新增，提示用户名存
        在
6.      }
7.      //由于新增管理员没有密码字段，需要初始化一个密码
8.      if(ObjectUtil.isEmpty(adminInfo.getPassword())){
9.          adminInfo.setPassword("123456");
10.     }
11.     adminInfoDao.insertSelective(adminInfo);
12. }
```

3,4, AdminInfoDao

```
1.  @Repository
2.  public interface AdminInfoDao extends Mapper<AdminInfo> {
3.      .....
4.      @Select("select * from admin_info where name = #{name}")
5.      AdminInfo findByName(@Param("name") String name);
6.
7.  }
```

3, exception 包内的自定义异常类和异常处理类

CustomException 自定义异常类

```
1.  public class CustomException extends RuntimeException { //捕获异常
2.      private String code;
3.      private String msg;
4.
5.      public CustomException(ResultCode resultCode) {
6.          this.code = resultCode.code;
7.          this.msg = resultCode.msg;
8.      }
9.      public CustomException(String code, String msg) {
10.         this.code = code;
11.         this.msg = msg;
```

```

12.     }
13.
14.     public String getCode() {
15.         return code;
16.     }
17.     public void setCode(String code) {
18.         this.code = code;
19.     }
20.     public String getMsg() {
21.         return msg;
22.     }
23.     public void setMsg(String msg) {
24.         this.msg = msg;
25.     }
26. }

```

GlobalExceptionHandler 异常捕获类

```

1.     @ControllerAdvice(basePackages="com.example.controller")
2.     // @ControllerAdvice 来声明一些全局性的东西，最常见的是结合
   // @ExceptionHandler 注解用于全局异常的处理
3.     public class GlobalExceptionHandler { // 处理捕获的异常
4.
5.         private static final Log log = LogFactory.get();
6.
7.
8.         // 统一异常处理 @ExceptionHandler, 主要用于 Exception
9.         @ExceptionHandler(Exception.class)
10.        // Spring 的 @ExceptionHandler 可以用来统一处理方法抛出的异常
11.        @ResponseBody // 返回 json 串
12.        public Result error(HttpServletRequest request, Exception e) {
13.            log.error("异常信息: ", e);
14.            return Result.error();
15.        }
16.
17.        @ExceptionHandler(CustomException.class)
18.        @ResponseBody // 返回 json 串
19.        public Result customError(HttpServletRequest request, CustomException e)
20.        {
21.            return Result.error(e.getCode(), e.getMsg());
22.        }
23.    }

```

2. 查询

刚点击 XXX 信息，都会从数据库根据 SQL 语句获取信息（有些会获取额外信息）并显示在屏幕上，并在完成一些增删改操作后都要重新查询展示。这里以管理员（AdminInfo）为例

运行展示

The screenshot displays a database management tool interface. On the left, a tree view shows the database schema, including tables like `admin_info`, `class_info`, `student_info`, `teacher_info`, `xuanke_info`, and `Administration`. The main area shows a SQL query: `SELECT * FROM bs_xk.admin_info;`. Below the query, a table of results is shown with columns: `id`, `name`, `password`, `sex`, `age`, `phone`, and `level`. The results include rows for users like 胡婷, 卢俊朴, admin, admin1, 卢俊辰, and admin2.

Below the database tool, a web application interface is shown. It features a sidebar menu with options like 信息管理, 管理员信息, 教师信息, 学生信息, 学院信息, 专业信息, 课程信息, 选课信息, and 个人信息. The main content area is titled '管理员信息' (Admin Information) and contains a table with columns: 姓名 (Name), 性别 (Gender), 年龄 (Age), 电话 (Phone), and 操作 (Action). The table lists the same admin users as the database tool. Each row has '编辑' (Edit) and '删除' (Delete) buttons. A search bar and pagination controls are also present.

查询实现步骤

1. `adminInfo.html` 构建显示前端，数据和 Vue 内新建数组对象双向绑定，Vue 方法新增函数 `tableData()`，用 `axios.get("/adminInfo")` 让 Controller 处理前端事件
2. Controller 直接使用 Service 方法
3. Service 使用 Mybatis 自动生成的 Dao 查询所有方法

查询涉及主要代码

以下主要代码前序号为对应的实现步骤

1, adminInfo.html

```
1. <table class="table table-striped table-bordered table-hover">
2.   <thead>
3.     <tr>
4.       <th>姓名</th>
5.       <th>性别</th>
6.       <th>年龄</th>
7.       <th>电话</th>
8.       <th>操作</th>
9.     </tr>
10.  </thead>
11.  <tbody v-for="obj in tableData">
12.    <tr>
13.
14.      <td>{{obj.name}}</td>
15.      <td>{{obj.sex}}</td>
16.      <td>{{obj.age}}</td>
17.      <td>{{obj.phone}}</td>
18.      <td>
19.        <button class="btn-primary btn-sm">编辑</button>
20.        <button class="btn-danger btn-sm">删除</button>
21.      </td>
22.
23.    </tr>
24.  </tbody>
25. </table>
26. new Vue({
27.   el: '#wrapper',
28.   data: {
29.     user: {},//一个对象，很多字段
30.     entity: {},//只有一个对象，用于传入前端输入的增加或更新数据的
      值给控制器
31.     tableData: [],//有很多对象，用于接收查询所有
32.     .....
33.   },
34.   created: function () {
35.     .....
36.     this.loadTable();
37.   },
38.
39.   methods: {
40.     loadTable(){//点击管理员信息时，查询所有人物信息并显示
41.       axios.get("/adminInfo").then(res => {
42.         if(res.data.code === '0'){
```

```

43.         this.tableData = res.data.data;
44.     }else{
45.         msg("error",res.data.msg);
46.     }
47. })
48. },
49. ....
50. },
51. }
52. })

```

2, AdminInfoController

```

1. @GetMapping
2. public Result findAll(){
3.     List<AdminInfo> list = admininfoService.findAll();
4.     return Result.success(list);
5. }

```

3, AdminInfoService

```

1. public List<AdminInfo> findAll(){
2.     return adminInfoDao.selectAll();
3. }

```

3. 编辑

运行展示

| 姓名 | 性别 | 年龄 | 电话 | 操作 |
|-----|----|----|--------------|---|
| 胡婷 | 女 | 20 | 520520520520 | <button>编辑</button> <button>删除</button> |
| 卢俊朴 | 男 | 21 | 520520520520 | <button>编辑</button> <button>删除</button> |

请填写信息

姓名

卢俊朴

性别

☒ 男 ☐ 女

年龄

21

电话

1234567891234

关闭

保存

● 编辑成功

新增

请输入搜索

| 姓名 | 性别 | 年龄 | 电话 | 操作 |
|-----|----|----|---------------|-----------------|
| 胡婷 | 女 | 20 | 520520520520 | <div>编辑删除</div> |
| 卢俊朴 | 男 | 21 | 1234567891234 | <div>编辑删除</div> |

编辑实现步骤

前端给每个编辑设置点击函数, `init(obj)` `obj` 为 `list` 内 `adminInfo` 对象, 用 `this.entity = JSON.parse(JSON.stringify(obj))`; 获取当前对象数据库的所有数据存放在 `entity`。再调用模态框, 此时, 由于模态框有隐藏的 `id`, 尽管 `add` 和 `init` 调用的是同一个模态框, 但 `add` 的 `entity` 没有隐藏的 `id` 数据。所以可以在模态框中根据是否有 `id` 数据, 来判断是应该新增数据库内数据还是编辑数据库数据。

编辑涉及主要代码

adminInfo.html 主要代码（后端在之前模块已经实现了，这里不展示）

模态框隐藏的 `id`

```

1. <div class="modal-body">
2.   <form class="form-horizontal">
3.     <input type="hidden" id="id" name="id" v-model="entity.id">

```

`submit` 的修改, 和 `init` 方法的实现

```

1. data: {
2.   user: {}, // 一个对象, 很多字段
3.   entity: {}, // 只有一个对象, 用于传入前端输入的增加或更新数据的
   值给控制器
4.   tableData: [], // 有很多对象, 用于接收查询所有
5.   .....
6. },
7. submit() { // 提交模板框输入数据到数据库
8.   // 模态框 id 属性隐藏了, 但是 add 情空了, init 没有, 所以 init 有 id, 可
   以根据 id 来判断模态框用的是 add 还是 init
9.   if(this.entity.id) { // 编辑操作
10.    axios.put("/adminInfo", this.entity).then(res => { // 在个人主页模块已经实现
11.      if(res.data.code === '0') {
12.        $('#mod').modal("hide"); // 关掉模态框
13.        msg("success", "编辑成功");
14.        this.loadTable();
15.      } else {
16.        msg("error", res.data.msg);

```



```

17.     }
18. })
19. } else{//新增操作
20.   axios.post("/adminInfo",this.entity).then(res=>{
21.     if(res.data.code === '0'){
22.       $('#mod').modal("hide");//隐藏前端模板框
23.       msg("success","插入成功");
24.       this.loadTable();
25.     }else{
26.       msg("error",res.data.msg);
27.     }
28.   })
29. }
30. },
31.
32.
33. init(obj){
34.   //JSON.stringify() 方法将一个 JavaScript 对象或值转换为 JSON 字符串，
35.   // 如果指定了一个 replacer 函数，则可以选择性地替换值，或者指定的
   // replacer 是数组，
36.   //则可选择性地仅包含数组指定的属性。
37.   // JSON.parse() 方法用来解析 JSON 字符串，构造由字符串描述
   // 的 JavaScript 值或对象。
38.   this.entity = JSON.parse(JSON.stringify(obj));
39.   $('#mod').modal("show");//出现问题，和增加用的一个模态框，但一个是
   // 新增，一个是更改
40.   //解决：模态框 id 属性隐藏了，但是 add 情空了，init 没有，所以 init 有
   // id，可以根据 id 来判断
41.   //模态框用的是 add 还是 init
42. },

```

4. 删除

在对应人员类型的管理员信息模块，点击删除，弹出框确定删除后，在主页和数据库实时删除信息。

运行展示



删除实现步骤

前端删除 button 添加 del () 函数，del 函数弹出框确认后，axios.delete 调用控制器方法，用 @PathVariable 获取路径信息，调用服务层函数，直接用 dao 层 Mybatis 提供的 SQL，deldeteByPrimaryKey

删除涉及主要代码

以下主要代码前序号为对应的实现步骤

adminInfo.html

```
1. <tbody v-for="obj in tableData">
2.   <tr>
3.
4.     <td>{{obj.name}}</td>
5.     <td>{{obj.sex}}</td>
6.     <td>{{obj.age}}</td>
7.     <td>{{obj.phone}}</td>
8.     <td>
9.       <button class="btn-primary btn-sm" @click="init(obj)">编辑</button>
10.      <button class="btn-danger btn-sm" @click="del(obj.id)">删除</button>
11.    </td>
12.
```

```

13.   </tr>
14. </tbody>
15. del(id) {
16.   let _this = this; //为了把弹出框的 this 和删除后的 loadTable 的 this 分开
17.   this.$confirm('确认删除?', '系统提示', {
18.     confirmButtonText: '确定',
19.     cancelButtonText: '取消',
20.     cancelButtonClass: 'btn-custom-cancel',
21.     type: 'warning'
22.   }).then(function() {
23.     // 删除
24.     axios.delete("/adminInfo/" + id).then(res => {
25.       if (res.data.code === '0') {
26.         msg("success", "删除成功");
27.         _this.loadTable();
28.       } else {
29.         msg("error", res.data.msg);
30.       }
31.     })
32.   }).catch(function() {
33.     console.log("取消成功!");
34.   });
35. },

```

AdminInfoController

```

1.   <tbody v-for="obj in tableData">
2.     <tr>
3.
4.       <td>{{obj.name}}</td>
5.       <td>{{obj.sex}}</td>
6.       <td>{{obj.age}}</td>
7.       <td>{{obj.phone}}</td>
8.       <td>
9.         <button class="btn-primary btn-sm" @click="init(obj)">编辑</button>
10.        <button class="btn-danger btn-sm" @click="del(obj.id)">删除</button>
11.      </td>
12.
13.    </tr>
14.  </tbody>
15.  del(id) {
16.    let _this = this; //为了把弹出框的 this 和删除后的 loadTable 的 this 分开
17.    this.$confirm('确认删除?', '系统提示', {
18.      confirmButtonText: '确定',
19.      cancelButtonText: '取消',

```

```

20.     cancelButtonClass: 'btn-custom-cancel',
21.     type: 'warning'
22. }).then(function() {
23.     // 删除
24.     axios.delete("/adminInfo/" + id).then(res => {
25.         if (res.data.code === '0') {
26.             msg("success", "删除成功");
27.             _this.loadTable();
28.         } else {
29.             msg("error", res.data.msg);
30.         }
31.     })
32. }).catch(function() {
33.     console.log("取消成功！");
34. });
35. },

```

AdmininfoService

```

1.  @Service
2.  public class AdminInfoService {
3.      @Resource
4.      private AdminInfoDao adminInfoDao;
5.      .....
6.
7.      public void deleteById(Long id){
8.          adminInfoDao.deleteByPrimaryKey(id);
9.      }
10.
11. }

```

（七）信息分页查询模块

让获取的数据分页显示，在管理员信息，学生信息，教师信息都用到了这个模块功能，下方展示是以 AdminInfo 为例。

运行展示

| | | | | | |
|-----|---|---|---|----|----|
| aaa | 男 | 1 | 1 | 编辑 | 删除 |
|-----|---|---|---|----|----|

上一页
 1
 2
 下一页

| 新增 | | | | |
|---|----|----|----|---------------------------------------|
| 姓名 | 性别 | 年龄 | 电话 | 操作 |
| bbb | 男 | 11 | 11 | 编辑 删除 |
| 上一页 1 2 下一页 | | | | |

实现步骤

1. 应用 PageHelper，用 PageHelper 相关数据实现前端，Vue 函数主要为扩展上述模块的 loadTable()方法，增加一个形参，添加传入控制器两个数据，用于进行 PageHelper 的运用
2. 控制器返回 PageInfo<AdminInfo>对象，里面包含各种页面信息和一个管理员 List 集合
3. 服务器在 findAll()的基础上增加了 PageHelper 的使用框架，具体看代码注释

涉及主要代码

以下主要代码前序号为对应的实现步骤

1, adminInfo.html

```

1. <nav aria-label="Page navigation example">
2.   <ul class="pagination">
3.     <!-- preActive 是 false 执行 -->
4.     <li class="page-item" v-bind:class="{ disabled: preActive }">
5.       <a class="page-link" href="javascript:void(0)"
6.         v-on:click="loadTable(pageInfo.hasPreviousPage? (pageInfo.pageNum -
7.           1) : pageInfo.pageNum)">上一页</a>
8.     </li>
9.     <li class="page-item">
10.      <a class="page-link" href="javascript:void(0)" v-if="pageInfo.pageNum >
11.        1" v-on:click="loadTable(pageInfo.pageNum - 1)">{{pageInfo.pageNum - 1}}
12.      </a>
13.    </li>
14.    <li class="page-item disabled">
15.      <a class="page-link" aria-disabled="true" href="javascript:void(0)">{{page
16.        Info.pageNum}}</a>
17.    </li>
18.    <li class="page-item">
19.      <a class="page-link" href="javascript:void(0)" v-on:click="loadTable(pageInfo.pageNum + 1)">
20.        下一页</a>
21.    </li>
22.  </ul>
23. </nav>

```

```

15.   <a class="page-link" href="javascript:void(0)" v-if="pageInfo.hasNextPage
    " v-on:click="loadTable(pageInfo.pageNum + 1)">{{pageInfo.pageNum + 1}}<
    /a>
16.   </li>
17.   <li class="page-item" v-bind:class="{ disabled: nextActive }">
18.     <a class="page-link" href="javascript:void(0)" v-on:click="loadTable(pageI
        nfo.hasNextPage? (pageInfo.pageNum + 1) : pageInfo.pageNum)">下一页</a>
19.   </li>
20. </ul>
21. </nav>
22. new Vue({
23.   el: '#wrapper',
24.   data: {
25.     user: {}, // 一个对象，很多字段
26.     entity: {}, // 只有一个对象，用于传入前端输入的增加或更新数据的值给
        控制器
27.     tableData: [], // 有很多对象，用于接收查询所有
28.     pageInfo: {},
29.     preActive: false,
30.     nextActive: false,
31.     search: ""
32.   },
33.   created: function () {
34.     .....
35.     this.loadTable(1);
36.   },
37.   methods: {
38.     add() { // 点击增加跳出模板框
39.       this.entity = {};
40.       $('#mod').modal("show"); // 调用前端模板框
41.     },
42.     .....
43.     loadTable(pageNum) { // 查询数据库所有人物信息并显示出来
44.       axios.get("/adminInfo/page?pageNum=" + pageNum + "&pageSize=5").th
        en(res => {
45.         if(res.data.code === '0'){
46.           this.tableData = res.data.data.list;
47.           this.pageInfo = res.data.data;
48.           this.preActive = !(res.pageInfo.hasPreviousPage);
49.           this.nextActive = !(res.pageInfo.hasNextPage);
50.         } else {
51.           msg("error", res.data.msg);
52.         }
53.       })

```

```

54.     },
55.     ..... } })

```

2, AdminInfoController

```

1.  @GetMapping("/page")//@RequestParam 获取传进来的值
2.  public Result findPage(@RequestParam Integer pageNum,@RequestParam
    Integer pageSize){
3.      //PageInfo 里面包含 list 和很多页面信息
4.      PageInfo<AdminInfo> info = adminInfoService.findPage(pageNum, pageSi
        ze);//返回的是 PageInfo.of(列表);
5.      //MyBatis 分页插件
6.      return Result.success(info);
7.  }

```

3, AdminInfoService

```

1.  @Service
2.  public class AdminInfoService {
3.      @Resource
4.      private AdminInfoDao adminInfoDao;
5.
6.      .....
7.
8.      public PageInfo<AdminInfo> findPage(Integer pageNum, Integer pageSize)
        {
9.          //1.开启分页
10.         PageHelper.startPage(pageNum,pageSize);
11.         //2.之后的查询会自动根据 pageNum 和 pageSize 来查询对应的数据
12.         List<AdminInfo> infos = adminInfoDao.selectAll();
13.         return PageInfo.of(infos);
14.     }
15. }

```

(八) 根据名字模糊查询模块

可以通过输入用户姓名（可以是一部分），快速锁定用户信息并显示

以 AdminInfo 管理员为例

运行展示

管理员信息

| <div>新增</div> <div>卢</div> | | | | |
|--|----|----|---------------|-----------------------------|
| 姓名 | 性别 | 年龄 | 电话 | 操作 |
| 卢俊朴 | 男 | 21 | 1234567891234 | <div>编辑</div> <div>删除</div> |
| 卢俊辰 | 男 | 10 | 24445566778 | <div>编辑</div> <div>删除</div> |
| <div>上一页</div> <div>1</div> <div>下一页</div> | | | | |

实现步骤

1. 前端搜索增加 v-model 双向绑定 search，并用@Keyup.enter 来让回车后调用显示模块。search 用来区分显示某一用户类信息时是否使用模糊查询，并改进 loadTable()，增添如果有 search 传入，则用 axios.put 到新增可传输部分名字的控制器
2. 此控制器类主要扩展分页查询所有为可以根据部分名字进行模糊查询
3. Dao 层实现需要用到 like concat 自定义查询语句

涉及主要代码

以下主要代码前序号为对应的实现步骤

1, adminInfo.html

```
1. <input type="text" placeholder="请输入搜索内容" v-model="search"
2. @Keyup.enter="loadTable(1)"><!--只有在 key 是 Enter 时调用,并传入输入
   数据 -->
3. data: {
4.     user: {},//一个对象，很多字段
5.     entity: {},//只有一个对象，用于传入前端输入的增加或更新数据的
      值给控制器
6.     tableData: [],//有很多对象，用于接收查询所有
7.     pageInfo: {},
8.     preActive: false,
9.     nextActive: false,
10.    search: ""
11. },
12. loadTable(pageNum) {//查询数据库人物信息并显示出来
13.     if(this.search){
14.         axios.get("/adminInfo/page/"+this.search+"?pageNum="+pageNum+"&
           pageSize=5").then(res => {
15.             if(res.data.code === '0') {
```



```

16.     this.tableData = res.data.data.list;
17.     this.pageInfo = res.data.data;
18.     this.preActive = !(res.pageInfo.hasPreviousPage);
19.     this.nextActive = !(res.pageInfo.hasNextPage);
20.   } else {
21.     msg("error", res.data.msg);
22.   }
23. })
24. } else {
25.   axios.get("/adminInfo/page?pageNum=" + pageNum + "&pageSize=5").then(
     res => {
26.     if (res.data.code === '0') {
27.       this.tableData = res.data.data.list;
28.       this.pageInfo = res.data.data;
29.       this.preActive = !(res.pageInfo.hasPreviousPage);
30.       this.nextActive = !(res.pageInfo.hasNextPage);
31.     } else {
32.       msg("error", res.data.msg);
33.     }
34.   })
35. }
36. },

```

2, AdminInfoController

```

1.  @RestController
2.  @RequestMapping("/adminInfo")
3.  public class AdminInfoController {
4.    @Resource
5.    private AdminInfoService adminInfoService;
6.    .....
7.    @GetMapping("/page")//用于一般分页显示
8.    public Result findPage(@RequestParam Integer pageNum, @RequestParam
      Integer pageSize) {
9.      PageInfo<AdminInfo> info = adminInfoService.findPage(pageNum, pag
      eSize);
10.     return Result.success(info);
11.   }
12.   @GetMapping("/page/{name}")//用于模糊查询
13.   // @RequestParam 获取传进来的值
14.   public Result findPage(@RequestParam Integer pageNum, @RequestParam
      Integer pageSize, @PathVariable String name){
15.     //PageInfo 里面包含 list 和很多页面信息
16.     PageInfo<AdminInfo> info = adminInfoService.findPageName(pageNu
      m, pageSize, name);//返回的是 PageInfo.of(列表);

```

```

17.    //MyBatis 分页插件
18.    return Result.success(info);
19. }
20. }

```

3, AdminInfoService

```

1.  public class AdminInfoService {
2.      @Resource
3.      private AdminInfoDao adminInfoDao;
4.      .....
5.      public PageInfo<AdminInfo> findPage(Integer pageNum, Integer pageSize)
        {
6.          //1.开启分页
7.          PageHelper.startPage(pageNum,pageSize);
8.          //2.之后的查询会自动根据 pageNum 和 pageSize 来查询对应的数据
9.          List<AdminInfo> infos = adminInfoDao.selectAll();
10.         return PageInfo.of(infos);
11.     } //直接查询并分页显示
12.     //根据名字模糊查询
13.     public PageInfo<AdminInfo> findPageName(Integer pageNum, Integer pageSize, String name) {
14.         PageHelper.startPage(pageNum,pageSize);
15.         List<AdminInfo> infos = adminInfoDao.findByNamePage(name);
16.         return PageInfo.of(infos);
17.     }
18. }

```

4, AdminInfoDao

```

1.  @Repository
2.  public interface AdminInfoDao extends Mapper<AdminInfo> {
3.
4.      @Select("select * from admin_info where name = #{name}")
5.      AdminInfo findByName(@Param("name") String name);
6.      //放在一起用来区分不同
7.      @Select("select * from admin_info where name like concat('%',#{name},'%')") //模糊查询之 LIKE CONCAT
8.      List<AdminInfo> findByNamePage(String name);
9.  }

```

(九) 双表联动

本例以学生模块与学院模块的双表联动为例，学生模块含有学院模块的主键(id)

1. 学生数据库获取学院数据库信息

在学生信息的管理与显示时，利用外键，获取外键所在表其他信息。

运行展示

学生信息

| 新增 | | | | | |
|-----|----|----|--------------|----|--------|
| 用户名 | 性别 | 年龄 | 学号 | 学分 | 所属学院 |
| 张三 | 男 | 24 | 20000 | 0 | 信息工程学院 |
| 李四 | 男 | 23 | 20001 | 0 | 计算机学院 |
| 王五 | 男 | 23 | 20002 | 0 | 软件学院 |
| 赵六 | 男 | 23 | 20003 | 0 | 外国语学院 |
| 卢同学 | 男 | 20 | 202113407120 | 3 | 计算机学院 |

学生数据库中没有学院名，是通过 `xueyuanId` 这个外键获取的

实现步骤

在 `studentInfo.html` 中增加在获取信息时，同步获取学员信息，再在编辑信息和增加信息的模态框中，用 `v-for` 和单选模块，根据获取到的学院的信息，下拉框显示所有学院名字，根据学院名字选择的值为学院 `Id` 同时，用 `v-model` 把学院 `id` 给学生的学院 `id`。

以实现在学生的信息修改模块，通过点击学院名称来更改自身的学院 `id` 值

涉及主要代码

`accountStudentInfo.html`

```
1. <div class="form-group">
2.   <label class="col-sm-3 control-label">所属学院</label>
3.   <div class="col-sm-7">
4.     <input disabled type="text" class="form-control" v-model="entity.xueyuanId"
5.       placeholder="">
6.   </div>
7. </div>
```

studentInfo.html

```
1. <!--此代码为模态框中增加的代码，模态框具体代码上述模块有记录 -->
2. <!-- 模态框 (Modal) -->
3. <div class="form-group">
4.   <label class="col-sm-3 control-label">所属学院</label>
5.   <div class="col-sm-7 form-inline">
6.     <select class="form-control" v-model="entity.xueyuanId" style="width: 100
      %">
7.       <option v-for="item in xueyuanObjs" :value="item.id">{{item.name}}</o
      ption>
8.     </select>
9.   </div>
10. </div>
11. data: {
12.   .....
13.   entity: {},//只有一个对象，用于传入前端输入的增加或更新数据的值给
      控制器
14.   xueyuanObjs: [],//多个对象，用于接收学院信息，与学院联动
15.   .....
16. },
17. methods:{
18.   loadXueyuan() { //学生每次加载页面都加载学院信息
19.     axios.get("/xueyuanInfo").then(res => {
20.       if (res.data.code === '0') {
21.         this.xueyuanObjs = res.data.data;
22.       } else {
23.         msg("error,res.data.msg");
24.       }
25.     })
26.   },
27. }
```

2. 学生模块根据与学院模块共有属性展示学院特有信息

由于学生数据库只有学院 id 与学院数据库相关，但在学生数据展示要展示学院名称

运行展示

实现步骤

1. 先在学生信息实体类中定义一个 `xueyuanName` 变量（用 `@Transient` 注解，代表这个变量不是数据库中的属性，不会映射到数据库中），用于存储根据学生数据库的学院 `id` 在学院数据库中查找到的学院名称。
2. 然后在学生模块增加查询返回的数据（在原本 `List` 中增加定义的非自身 SQL 中的数据），并显示在前端的 `Service` 方法中（`findPage` 和 `findPageSearch`）。有两种方式。

方式 1：在包含学生对象的 `List` 列表中，根据 `studentInfo` 里面的 `xueyuanId`，查询到学院信息，然后通过学院的 `Dao` 方法，查到名称赋值给 `xueyuanName` 变量。

最后更改学生信息中的显示内容

方式 2：用 SQL 关联查询语句，在 `dao` 中编写。通过两者共同 `Id`，`LEFT JOIN`（连接）两表，模糊查询有输入搜索字段的语句。（注意：）

涉及主要代码

1, StudentInfo

```
1. ....
2. @Transient//代表这个变量不是数据库中的属性
3. private String xueyuanName;
4.
5. public String getXueyuanName() {
6.     return xueyuanName;
7. }
8.
```

```

9.  public void setXueyuanName(String xueyuanName) {
10.     this.xueyuanName = xueyuanName;
11.     .....
12. }

```

步骤 2 实现的两种方式

方式一

StudentInfoService

有问题：在涉及到多表联合以及模糊查询时，无法使用，必须使用第二种方法编写 SQL 语句实现

```

1.  .....
2.  @Resource
3.  private StudentInfoDao studentInfoDao;
4.  @Resource
5.  private XueyuanInfoDao xueyuanInfoDao;
6.  .....
7.  public PageInfo<StudentInfo> findPage(Integer pageNum, Integer pageSize)
    {
8.      PageHelper.startPage(pageNum, pageSize);
9.      List<StudentInfo> list = studentInfoDao.selectAll();
10.     //根据学生数据库的学院 id 在 学院中找到 id 对应的学院名称
11.     //方式一：
12.     for (StudentInfo studentInfo : list) {
13.         if (ObjectUtil.isEmpty(studentInfo.getXueyuanId())) {
14.             //根据 studentInfo 里面的 xueyuanId，查询到学员信息，然后通过
             学院的 Dao 方法，查到名称赋值给 不在学生数据库中的 xueyuanName
15.             XueyuanInfo xueyuanInfo = xueyuanInfoDao.selectByPrimaryKey(
                studentInfo.getXueyuanId());
16.             studentInfo.setXueyuanName(xueyuanInfo.getName());
17.         }
18.     }
19.     return PageInfo.of(list);
20. }
21. public PageInfo<StudentInfo> findPageSearch(Integer pageNum, Integer pa
    geSize, String search) {
22.     PageHelper.startPage(pageNum, pageSize, search);
23.     List<StudentInfo> list = studentInfoDao.findByName(search);
24.     return PageInfo.of(list);
25. }
26. .....

```

方式二：

StudentInfoService

```
1. ....
2. @Resource
3. private StudentInfoDao studentInfoDao;
4. @Resource
5. private XueyuanInfoDao xueyuanInfoDao;
6. ....
7. public PageInfo<StudentInfo> findPage(Integer pageNum, Integer pageSize)
   {
8.     PageHelper.startPage(pageNum, pageSize);
9.     //方式二，使用sql 关联语句
10.    List<StudentInfo> list = studentInfoDao.findAllJoinXueyuan();
11.    return PageInfo.of(list);
12. }
13. public PageInfo<StudentInfo> findPageSearch(Integer pageNum, Integer pageSize, String search) {
14.     PageHelper.startPage(pageNum, pageSize, search);
15.     List<StudentInfo> list = studentInfoDao.findByLikeName(search);
16.     return PageInfo.of(list);
17. }
18. ....
```

StudentDao

```
1. @Select("SELECT a.*,b.name AS xueyuanName FROM student_info AS a LEFT JOIN xueyuan_info AS b ON a.xueyuanId = b.Id")
2. List<StudentInfo> findAllJoinXueyuan();
3. @Select("select a.*,b.name AS xueyuanName " +
4.     "from student_info AS a " +
5.     "LEFT JOIN xueyuan_info AS b " +
6.     "ON a.xueyuanId = b.id " +
7.     "where a.name like concat('%',{search},%')")
8. List<StudentInfo> findByLikeName(@Param("name") String search);
```

（十）三表联动

本例以选课模块和教师模块与专业模块的联动为例

选课信息（classInfo）包含了教师的主键（id）和专业的主键（id）

运行展示

| 新增 | | | | | | | | | 请 | |
|----------|--------------|------|------|-------|------|--------|-------|------|---|--|
| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 已选人数 | | |
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 2 | | |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 3 | | |
| Python基础 | 学会Python基础 | 3 | 杨松 | 软件工程 | 15 | 周二第一大节 | 11213 | 2 | | |
| 德语商务 | 学会跟德国人谈生意 | 3 | 古月婷 | 德语 | 5 | 周三第二大节 | 30402 | 1 | | |
| 古代曾语 | 教你元谋人怎么书画 | 2 | 晓黄仁 | 古汉语信息 | 5 | 周四第三大节 | 33401 | 2 | | |

课程信息中没有教师名和专业名称，在查询时通过外键在查询中获取的。

实现步骤

与双表联动基本相同，但在进行根据共有信息查询另外两表特有信息时，由于涉及三表连接容易发生异常，所以 ClassInfoDao 的 SQL 语句十分重要

涉及主要代码

ClassInfo

```
1. @Transient
2. private String teacherName;
3. @Transient
4. private String zhuanYeName;
```

ClassInfoDao

```
1. @Repository
2. public interface ClassInfoDao extends Mapper<ClassInfo> {
3.
4.     @Select("select a.*, b.name AS teacherName , c.name AS zhuanYeName\n"
5.         +
6.         "from class_info AS a \n" +
7.         "INNER JOIN teacher_info AS b \n" +
8.         "RIGHT JOIN zhuanYe_info AS c\n" +
9.         "ON a.teacherId = b.id AND a.zhuanYeId = c.id\n" +
10.        "where a.name like concat('%',{search},%')")
```



```

10.    List<ClassInfo> findSearch(@Param("search") String search);
11.
12.    @Select("select a.*, b.name AS teacherName , c.name AS zhuanYeName\n"
    +
13.        "from class_info AS a \n" +
14.        "INNER JOIN teacher_info AS b \n" +
15.        "INNER JOIN zhuanYe_info AS c\n" +
16.        "ON a.teacherId = b.id AND a.zhuanYeId = c.id")
17.    List<ClassInfo> findAll();
18.
19. }

```

其他涉及代码和上述模块相似，不做过多说明

重要涉及知识：

left join 和 inner join 的区别

1. left join 是返回右边全部记录,右表不满足匹配条件的记录返回对应行返回 null,那么单纯的对比逻辑运算量的话,inner join 是需要返回 2 个表的交集部分,left join 多返回一部分左表没有返回的数据.
2. mysql 文档来看 inner join 在连接的时候 mysql 回自动选择较小的表来作为驱动表,从而达到减少循环次数的目的.
3. 在使用 left join 的时候它是默认左表作为驱动表的,那么左表的大小是我们来控制的,如果控制不当,左表比较大,那么循环次数自然就会比较多,效率就会下降.

（十一）四标联动

应用场景：选课信息的 SQL 表中，存储着，老师 id，专业 id，学生 id，在查看选课信息时，为了提高程序的可读性，需要根据选课信息内存储的三个 id，查找另外三个数据库中对应 id 的名字

运行展示

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 选课学生 | 课程状态 |
|--------|--------------|------|------|------|------|--------|-------|------|------|
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 李四 | 待开课 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 20 | 周一第一大节 | 11201 | 卢同学 | 已结课 |
| 德语商务 | 学会跟德国人谈生意 | 3 | 古月婷 | 德语 | 5 | 周三第二大节 | 30402 | 胡婷 | 待开课 |
| 商务英语 | 各种商务场景英语交流技巧 | 3 | 杜碑 | 英语 | 10 | 周五第六大节 | 22301 | 胡婷 | 待开课 |

实现步骤

与双表，三表联动基本相同，这里只说明 SQL 语句的编写

涉及主要代码

XuankeInfo

```
1. ....
2. @Transient
3. private String teacherName;
4. @Transient
5. private String zhuanyeName;
6. @Transient
7. private String studentName;
```

XuankeInfoDao（选课模块还有改进）

```
1. @Repository
2. public interface XuankeInfoDao extends Mapper<XuankeInfo> {
3.
4.
5. @Select("SELECT a.*,b.name AS teacherName,c.name AS studentName,d.name AS zhuanyeName\n" +
6. "FROM xuanke_info AS a\n" +
7. "INNER JOIN teacher_info AS b\n" +
8. "INNER JOIN student_info AS c\n" +
9. "INNER JOIN zhuanye_info AS d\n" +
10. "ON a.teacherId=b.id AND a.studentId=c.id AND a.zhuanyeId = d.id")
11. List<XuankeInfo> findAllJoinTeacherAndZhuanye();
12. }
```

（十二）用户角色权限的控制模块

本系统是根据 SQL 中的 level 属性，来判断是哪种类型的学生登录，所以用 level 来进行权限控制。来实现对应类型用户访问不同类型模块。

运行展示在整体运行展示部分。

实现步骤

如果向直接把 `v-if="user.level === 1"` 加在对应前端标签里, 表示只有管理员能看到此标签。

部分使用演示

xueyuanInfo.html

以下代码让除了管理员外的用户类型, 无法看到全体管理员信息, 全体教师信息, 全体老师信息。并且只能看到, 而无法编辑学院信息和专业信息

```
1. ....
2. <li v-if="user.level === 1">
3.   <a href="adminInfo.html"><i class="fa fa-table sub-icon-mg" aria-hidden="true"></i><span class="mini-sub-pro">管理员信息</span></a>
4. </li>
5. <li v-if="user.level === 1">
6.   <a href="teacherInfo.html"><i class="fa fa-table sub-icon-mg" aria-hidden="true"></i><span class="mini-sub-pro">教师信息</span></a>
7. </li>
8. <li v-if="user.level === 1">
9.   <a href="studentInfo.html"><i class="fa fa-table sub-icon-mg" aria-hidden="true"></i><span class="mini-sub-pro">学生信息</span></a>
10. </li>
11. ....
12. <thead>
13.   <tr>
14.     <th>学院 Id</th>
15.     <th>学院名称</th>
16.     <th>学院介绍</th>
17.     <th>学分限制</th>
18.     <th v-if="user.level === 1">操作</th>
19.   </tr>
20. </thead>
21. <tbody v-for="obj in tableData">
22.   <tr>
23.     <td>{{obj.id}}</td>
24.     <td>{{obj.name}}</td>
25.     <td>{{obj.description}}</td>
26.     <td>{{obj.score}}</td>
27.     <td v-if="user.level === 1">
```

```

28.         <button v-if="user.level === 1" class="btn btn-sm btn-primary" @click="init(obj)">编辑</button>
29.         <button v-if="user.level === 1" class="btn btn-sm btn-danger" @click="del(obj.id)">删除</button>
30.     </td>
31. </tr>
32. </tbody>

```

(十三) 学生选课和退课模块

需求：

学生用户登录，在 classInfo 模块点击选课按钮，把 StudentInfo 和 ClassInfo 的 SQL 数据，传入到 XuankelInfo 的 SQL 数据，之后 ClassInfo 的已选人数+1。

每个学生只能看到自己的选课信息。每个老师只能看到和自己有关的选课信息。

只有学生能够点击退课，删除所选课程。

运行展示

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 已选人数 | 操作 |
|------|--------------|------|------|------|------|--------|-------|------|---|
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 1 | <button>选课</button> <button>退课</button> |

| | | | | | | | | | 选课成功 |
|---------|--------------|------|------|------|------|--------|-------|------|---|
| 请输入搜索内容 | | | | | | | | | |
| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 已选人数 | 操作 |
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 2 | <button>选课</button> <button>退课</button> |

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 已选人数 | 操作 |
|----------|--------------|------|------|------|------|--------|-------|------|---|
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 2 | <button>选课</button> <button>退课</button> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 3 | <button>选课</button> <button>退课</button> |
| Python基础 | 学会Python基础 | 3 | 卢老师 | 软件工程 | 15 | 周二第一大节 | 11213 | 2 | <button>选课</button> <button>退课</button> |
| 德语商务 | 学会跟德国人谈 | 3 | 卢老师 | 软件工程 | 15 | 周三第二大节 | 30402 | 1 | <button>选课</button> <button>退课</button> |
| 古代鲁语 | 教你元谋人怎么 | 3 | 卢老师 | 软件工程 | 15 | 周四第三大节 | 33401 | 2 | <button>选课</button> <button>退课</button> |

| | | | | | | | | | 请输入搜索内容 |
|------|--------------|------|------|------|------|--------|-------|------|---|
| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 已选人数 | 操作 |
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 1 | <button>选课</button> <button>退课</button> |

| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 选课学生 | 课程状态 |
|----------|--------------|------|------|-------|------|--------|-------|------|------|
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 20 | 周一第一大节 | 11201 | 卢同学 | 已结课 |
| Python基础 | 学会Python基础 | 3 | 杨松 | 软件工程 | 15 | 周二第一大节 | 11213 | 卢同学 | 待开课 |
| 商务英语 | 各种商务场景英语交流技巧 | 3 | 杜啤 | 英语 | 10 | 周五第六大节 | 22301 | 卢同学 | 待开课 |
| 古代汉语 | 教你死读人怎么书画 | 2 | 晓黄仁 | 古汉语信息 | 5 | 周四第三大节 | 33401 | 卢同学 | 待开课 |
| 网络安全概论 | 这是一条课程介绍 | 1 | 卢老师 | 网络安全 | 10 | 周五第三大节 | 11202 | 卢同学 | 待开课 |
| 网络攻防 | 教授黑客入侵的方法与防治 | 4 | 叶凡 | 网络安全 | 15 | 周二第四大节 | 31302 | 卢同学 | 待开课 |

实现步骤

1. 在 ClassInfo.html 中设置只有学生才能显示的选课按钮，设置点击函数 xuanke()，如果课程已选人数没有达到开班人数，点击后 axios.post 点击行的 CassInfo 对象到 ClassInfoController，进行操作
2. 在 CalssInfoController 先获取当前登录的用户，然后进行三个步骤：
 - 把课程信息塞进选课信息表里。
 - 把其他信息塞进选课信息表里。
 - 课程信息表里已选人数加一。
3. 对 Xuanke 加载用户后的获取所有然后显示在前端改进，在 XuankelInfoController 根据登录用户 level 等级，使用不同的 Service 来调用不同的 Dao 语句（都是四表联动），来达到只能查看和自己有关的选课信息的目的。
4. 退课模块是选课模块的简化版，实现更简单，不多做说明

涉及主要代码

解释

以下主要代码前序号为对应的实现步骤

1, 4,classInfo.html

```

1. <tbody v-for="obj in tableData">
2.   <tr>
3.     <td>{{obj.name}}</td>
4.     <td>{{obj.description}}</td>
5.     <td>{{obj.score}}</td>
6.     <td>{{obj.teacherName}}</td>
7.     <td>{{obj.zhuanyeName}}</td>
8.     <td>{{obj.kaiban}}</td>

```

```

9.      <td>{{obj.time}}</td>
10.     <td>{{obj.location}}</td>
11.     <td>{{obj.yixuan}}</td>
12.     <td v-if="user.level !== 2">
13.       <button v-if="user.level === 3" :disabled="obj.kaiban === obj.yixuan"
14.         class="btn btn-sm btn-primary" @click="xuanke(obj)">选课
15.     </button>
16.     <button v-if="user.level === 3" :disabled="'0' === obj.yixuan"
17.       class="btn btn-sm btn-danger" @click="tuike(obj.name)">退课
18.     </button>
19.     <button v-if="user.level === 1" class="btn btn-sm btn-primary"
20.       @click="init(obj)">编辑
21.     </button>
22.     <button v-if="user.level === 1" class="btn btn-sm btn-danger"
23.       @click="del(obj.id)">删除
24.     </button>
25.   </td>
26.
27. </tr>
28. </tbody>
29. xuanke(obj) {
30.   axios.post("/classInfo/xuanke", JSON.parse(JSON.stringify(obj))).then(res => {
31.     if (res.data.code === '0') {
32.       msg("success", "选课成功");
33.       this.loadTable();
34.     } else {
35.       msg("error", res.data.msg)
36.     }
37.   })
38. },
39. tuike(name){
40.   let _this = this; //为了把弹出框的 this 和删除后的 loadTable 的 this 分开
41.   this.$confirm('确认删除?', '系统提示', {
42.     confirmButtonText: '确定',
43.     cancelButtonText: '取消',
44.     cancelButtonClass: 'btn-custom-cancel',
45.     type: 'warning'
46.   }).then(function () {
47.     axios.delete("/classInfo/tuike/" + name).then(res => {
48.       if (res.data.code === '0') {
49.         msg("success", "退课成功");
50.         this.loadTable();
51.       } else {

```

```

52.         msg("error", res.data.msg)
53.     }
54. })
55. }).catch(function () {
56.     console.log("取消成功！");
57. });
58. },

```

2,ClassInfoController

```

1.  @RestController
2.  @RequestMapping("/classInfo")
3.  public class ClassInfoController {
4.
5.      @Resource
6.      private ClassInfoService classInfoService;
7.      @Resource
8.      private XuankeInfoService xuankeInfoService;
9.
10.     @PostMapping("/xuanke")
11.     public Result xuanke(@RequestBody ClassInfo classInfo, HttpServletRequest
request){
12.         //如果课程已选人数是否达到开班人数（前端实现）
13.         //  后端实现方
14.         法： if (classInfo.getKaiban().equals(classInfo.getYixuan())) {
15.             //          throw new CustomException("-1", "选课人数已满");
16.             //      }
17.
18.         Account user = (Account) request.getSession().getAttribute("user");//拿到
19.         当前登录用户
20.
21.         //防止登录失效
22.         if(ObjectUtil.isEmpty(user)){
23.             throw new CustomException("-1","登陆已失效，请重新登录");
24.         }
25.
26.         //防止重复选一门课
27.         if(ObjectUtil.isNotEmpty(xuankeInfoService.selectByThreeInfo(classInfo.g
etName(),user.getId()))){
28.             throw new CustomException("-1","你已经选过了");
29.         }
30.
31.         //1,把课程信息塞进选课信息表里
32.         XuankeInfo xuankeInfo=new XuankeInfo();
33.         BeanUtils.copyProperties(classInfo,xuankeInfo);

```

```

29.    // BeanUtils.copyProperties, BeanUtils 它提供了对java 反射和自省
    API 的包装如果有两个具有很多相同属性的JavaBean, 用copyProperties
30.    xuankeInfo.setId(null);//清空 classInfo 传来的Id
31.
32.    //2,把其他信息塞进选课信息表里
33.    xuankeInfo.setStudentId(user.getId());
34.    xuankeInfo.setStatus("待开课");
35.    xuankeInfoService.add(xuankeInfo);//调用选课信息服务的增加, 存入选
    课数据库中
36.
37.    //3, 课程信息表里已选人数加一
38.    classInfo.setYixuan(classInfo.getYixuan()+1);
39.    classInfoService.update(classInfo);
40.
41.    return Result.success();
42. }
43.
44. @DeleteMapping("/tuike/{name}")
45. public Result tuike(@PathVariable String name,HttpServletRequest request){
46.    //选课人数为0 无法退课 (前端实现)
47.    Account user = (Account) request.getSession().getAttribute("user");//拿到
    当前登录用户
48.    //防止登录失效
49.    if(ObjectUtil.isEmpty(user)){
50.        throw new CustomException("-1","登陆已失效, 请重新登录");
51.    }
52.    //退课
53.    Long studentId = user.getId();
54.    xuankeInfoService.tuike(name,studentId);
1.    ClassInfo classInfo = classInfoService.selectByName(name);
2.        classInfo.setYixuan(classInfo.getYixuan()-1);
3.        classInfoService.update(classInfo);
55.    return Result.success();
56. }
57. }

```

2, XuankeInfoController

```

1.    @RestController
2.    @RequestMapping("xuankeInfo")
3.    public class XuankeInfoController {
4.        @Resource
5.        XuankeInfoService xuankeInfoService;
6.
7.        @GetMapping

```



```

8.    public Result findAll(HttpServletRequest request){//根据登录的用户是谁，
    来显示只能和自己相关的选课内容，不能侵犯他人隐私
9.        Account user = (Account) request.getSession().getAttribute("user");
10.        if(ObjectUtil.isEmpty(user)){
11.            throw new CustomException("-1","登录失效，请重新登陆");
12.        }
13.        if(1 == user.getLevel()) {
14.            //管理员
15.            //管理员可以看到所有的算可信息
16.            List<XuankeInfo> list = xuankeInfoService.findAllJoinTeacherAndZhua
            nye();
17.            return Result.success(list);
18.        }
19.        else if(2 == user.getLevel()){
20.            //教师
21.            //教师只能看到自己代课的的选课信息
22.            List<XuankeInfo> list = xuankeInfoService.findAllJoinTeacherAndZhua
            nyeThroughTeacherId(user.getId());
23.            return Result.success(list);
24.        }
25.        else if(3 == user.getLevel()){
26.            //学生
27.            //学生只能看到自己的选课信息
28.            List<XuankeInfo> list = xuankeInfoService.findAllJoinTeacherAndZhua
            nyeThroughStudentId(user.getId());
29.            return Result.success(list);
30.        }
31.        else
32.            return Result.error();
33.    };
34.    }

```

3, 4, XuankeInfoService

```

1.    @Service
2.    public class XuankeInfoService {
3.        @Resource
4.        private XuankeInfoDao xuankeInfoDao;
5.        public XuankeInfo selectByThreeInfo(String name, Long studentId) {
6.            return xuankeInfoDao.selectByThreeInfo(name, studentId);
7.        }
8.
9.        public void tuike(String classname, Long studentId) {
10.            xuankeInfoDao.tuike(classname,studentId);
11.        }

```

```

12.
13. public List<XuankeInfo> findAllJoinTeacherAndZhuanyeThroughTeacherId(
    Long teacherId) {
14.     return xuankeInfoDao.findAllJoinTeacherAndZhuanyeThroughTeacherId(te
        acherId);
15. }
16.
17. public List<XuankeInfo> findAllJoinTeacherAndZhuanyeThroughStudentId(
    Long studentId) {
18.     return xuankeInfoDao.findAllJoinTeacherAndZhuanyeThroughStudentId(st
        udentId);
19. }
20. }

```

3, 4, XuankeInfoDao

```

1. @Repository
2. public interface XuankeInfoDao extends Mapper<XuankeInfo> {
3.     @Select("SELECT a.*,b.name AS teacherName,c.name AS studentName,d.n
        ame AS zhuanyeName\n" +
4.         "FROM xuanke_info AS a\n" +
5.         "INNER JOIN teacher_info AS b\n" +
6.         "INNER JOIN student_info AS c\n" +
7.         "INNER JOIN zhuanye_info AS d\n" +
8.         "ON a.teacherId=b.id AND a.studentId=c.id AND a.zhuanyeId = d.id")
9.     List<XuankeInfo> findAllJoinTeacherAndZhuanye();
10.
11. // @Select("SELECT * FROM xuanke_info WHERE studentId = #{ID}")
12. // XuankeInfo selectByStudentId(@Param("ID") Long ID); 这个会导致一
    个人只能选一门课
13.
14. @Select("SELECT * FROM xuanke_info WHERE studentId=#{studentId} A
    ND name=#{name}")
15. XuankeInfo selectByThreeInfo(@Param("name") String name, @Param("stud
    entId") Long studentId);
16.
17. @Delete("DELETE FROM xuanke_info WHERE name = #{classname} AN
    D studentId = #{studentId}")
18. void tuike(@Param("classname") String classname, @Param("studentId") Lon
    g studentId);
19.
20. @Select("SELECT a.*,b.name AS teacherName,c.name AS studentName,d.na
    me AS zhuanyeName\n" +
21.     "FROM xuanke_info AS a\n" +
22.     "INNER JOIN teacher_info AS b\n" +

```

```

23.         "INNER JOIN student_info AS c\n" +
24.         "INNER JOIN zhuanye_info AS d\n" +
25.         "ON a.teacherId=b.id AND a.studentId=c.id AND a.zhuanyeId = d.id\n"
26.         +
27.         "WHERE a.teacherId=#{teacherId}" )
28. List<XuankeInfo> findAllJoinTeacherAndZhuanyeThroughTeacherId(@Param("teacherId") Long teacherId);
29. @Select("SELECT a.*,b.name AS teacherName,c.name AS studentName,d.name AS zhuanyeName\n" +
30.         "FROM xuanke_info AS a\n" +
31.         "INNER JOIN teacher_info AS b\n" +
32.         "INNER JOIN student_info AS c\n" +
33.         "INNER JOIN zhuanye_info AS d\n" +
34.         "ON a.teacherId=b.id AND a.studentId=c.id AND a.zhuanyeId = d.id\n"
35.         +
36.         "WHERE a.studentId=#{studentId}")
37. List<XuankeInfo> findAllJoinTeacherAndZhuanyeThroughStudentId(@Param("studentId") Long studentId);
38. }

```

(十四) 老师开课和结课模块

| ● 开课成功 | | | | | | | | | | |
|--------|------------|------|------|------|------|--------|-------|------|------|---|
| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 选课学生 | 课程状态 | 操作 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 20 | 周一第一大节 | 11201 | 卢同学 | 已结课 | <button>开课</button> <button>结课</button> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 小明 | 已开课 | <button>开课</button> <button>结课</button> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 胡婷 | 已开课 | <button>开课</button> <button>结课</button> |
| 网络安全概论 | 这是一条课程介绍 | 1 | 卢老师 | 网络安全 | 10 | 周五第三大节 | 11202 | 卢同学 | 待开课 | <button>开课</button> <button>结课</button> |

| | | | | | | | | |
|----|----|--------|---|----|-----------|---|---|---|
| 33 | 胡婷 | 123456 | 女 | 20 | 202110000 | 3 | 4 | 3 |
| 34 | 小明 | 123456 | 男 | 21 | 20223474 | 3 | 2 | 0 |

| ● 结课成功 | | | | | | | | | | |
|--------|------------|------|------|------|------|--------|-------|------|------|---|
| 课程名称 | 课程介绍 | 课程学分 | 授课教师 | 所属专业 | 开班人数 | 上课时段 | 上课地点 | 选课学生 | 课程状态 | 操作 |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 20 | 周一第一大节 | 11201 | 卢同学 | 已结课 | <button>开课</button> <button>结课</button> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 小明 | 已结课 | <button>开课</button> <button>结课</button> |
| Java基础 | 学会Java语言基础 | 3 | 卢老师 | 软件工程 | 15 | 周一第一大节 | 11201 | 胡婷 | 已结课 | <button>开课</button> <button>结课</button> |
| 网络安全概论 | 这是一条课程介绍 | 1 | 卢老师 | 网络安全 | 10 | 周五第三大节 | 11202 | 卢同学 | 待开课 | <button>开课</button> <button>结课</button> |

| | | | | | | | | |
|----|----|--------|---|----|-----------|---|---|---|
| 33 | 胡婷 | 123456 | 女 | 20 | 202110000 | 3 | 4 | 6 |
| 34 | 小明 | 123456 | 男 | 21 | 20223474 | 3 | 2 | 3 |

实现步骤

在前端增加只有老师有权限的，开课和结课按钮，未开课才能开课，开课了才能结课。并添加相对应的点击函数。通过 `axios.put` 调用数据到控制器进行更改操作。

在控制器，开课只需要改变选课信息的状态，并且阻止学生选课。结课则要先给选课学生加学分，再更新选课信息的状态

涉及主要代码

XuankeInfo.html

```

1.  <td v-if="user.level === 2">
2.    <button v-if="user.level === 2" :disabled="obj.status === '已开课'" class="btn btn-sm btn-primary" @click="kaike(obj)">开课</button>
3.    <button v-if="user.level === 2" :disabled="obj.status !== '已开课'" class="btn btn-danger btn-primary btn-sm" @click="jieke(obj)">结课</button>
4.  </td>
5.  kaike(obj){
6.    obj.status = "已开课";
7.    axios.put("/xuankeInfo/kaike",obj).then(res=> {
8.      if (res.data.code === '0') {
9.        msg('success', '开课成功');
10.       this.loadTable();
11.      } else {
12.        msg('error', res.data.msg);
13.      }
14.    })
15.  },
16.  jieke(obj){
17.    this.$confirm('确认结课?', '系统提示', {
18.      confirmButtonText: '确定',
19.      cancelButtonText: '取消',
20.      cancelButtonClass: 'btn-custom-cancel',
21.      type: 'warning'
22.    }).then(function () {
23.      obj.status = "已结课";
24.      axios.put("/xuankeInfo/jieke",obj).then(res=> {

```

```

25.     if (res.data.code === '0') {
26.         msg('success', '结课成功');
27.         this.loadTable();
28.     } else {
29.         msg('error', res.data.msg);
30.     }
31. })
32. }).catch(function () {
33.     console.log("取消成功！");
34. });
35. },

```

XuankeInfoController

```

1.  @RestController
2.  @RequestMapping("/xuankeInfo")
3.  public class XuankeInfoController {
4.      @Resource
5.      XuankeInfoService xuankeInfoService;
6.      @Resource
7.      StudentInfoService studentInfoService;
8.      .....
9.      @PutMapping("/kaike")
10.     public Result kaikeupdate(@RequestBody XuankeInfo xuankeInfo) {
11.         xuankeInfoService.update(xuankeInfo);
12.         return Result.success();
13.     }
14.
15.     @PutMapping("/jieke")
16.     public Result jiekeupdate(@RequestBody XuankeInfo xuankeInfo) {
17.         //给选课学生增加学分
18.         Long studentId = xuankeInfo.getStudentId();
19.         StudentInfo student = studentInfoService.findById(studentId);
20.         student.setScore(student.getScore()+ xuankeInfo.getScore());
21.         studentInfoService.update(student);
22.         //更新课程状态
23.         xuankeInfoService.update(xuankeInfo);
24.         return Result.success();
25.     }
26. }

```

XuankeInfoService

```

1.  @Service
2.  public class XuankeInfoService {

```

```
3.  @Resource
4.  private XuankeInfoDao xuankeInfoDao;
5.  public void update(XuankeInfo xuankeInfo) {
6.      xuankeInfoDao.updateByPrimaryKeySelective(xuankeInfo);
7.  }
8.  }
```

六、总结

这是我第一次搞前后端的全栈开发，感觉很有意思，学数据库和 javaEE 的时候一直没时间把学到用到实战中。

这次课设，我从头开始复习了 java 和 javaEE 的基础知识，学习了一些前端框架和项目脚手架的应用。开始把学习的内容用到实战中去，比如数据库的 SQL 语句，如何通过把需求转化为 ER 图再构造实体，以及 java 的集合，多态，泛型，各种包的实战运用。

通过这次课设，我大概了解了开发的流程（产品分析，ER 图构建，分析需求，搭前端框架，填充后端内容，测试等等），也算完成了一个最最基础的开发，我也明白了自学是多么的重要。我在课设开始前复习了一下语言知识，才让我在课设的开始阶段没那么困难。

之前做什么设计和课堂练习，我都是万一改错就不同 ctrl+z，找 IDEA 恢复历史记录，这次暑假我在学 Linux 的时候，刷到了 Git 和 GitHubDesktop 的教学视频，我就跟着学了一下，最后发现这个刚好可以解决我的需求。

同时，在进行数据库操作的时候我遇到了很多很多困难，之前上课学的都是理论知识，上机课并没有用到数据库和 IDEA 等软件联合使用，真正自己使用并根据 SQL 语句解决问题的时候，一开始就非常的蒙圈，直到做完登录和个人信息显示模块才开始熟练（所以个人信息显示模块步骤写的非常细致）。

到后来，大体框架成型后，在增加更多功能的时候发现，基础无比重要，java 的基础，vue 的基础，SQL 语句编写的基础，都决定着我要改多久的 BUG。

最后，我也充分体会到了 AI 的冲击，我在课设快写完的时候，有一个模块不会写，就去网上冲浪，看到了有个叫 CodeGeeX 的插件。我就尝试用了一下，让我非常震惊，虽然有时候有些不是很好用，但很多时候我觉得它甚至会读心，能猜到我接下来想写什么，这让我对未来感到迷茫。

总之，这次课设对我的作用十分巨大，让我认识到了自己是多么渺小，一个简单的功能有时候要想一两个小时，一个 bug 能让我卡两三个小时，这给了我学习动力。同时我也学会了如何在项目中运用书本上的部分知识。