南 京 审 计 大 学

**研究生课程报告**

题目： 学生选课系统

组长学号姓名： 罗伟力MP2309018 模块： 用户管理、课程信息管理

成员学号姓名： 徐俊华MP2309021 模块： 学生选课

成员学号姓名： 杨瑞MP2309033 模块： 导入导出、数据库管理

课程名： 高级程序设计语言 编号： XX20090

任课教师： 唐明伟 职称： 副教授

二○二三年 一 月

**1.开发目标**

本项目旨在设计和实现一套学生选课系统，以提高学校管理效率和学生选课体验。该系统将涵盖学生、教师和管理员三个角色，为其提供相应的功能和权限。

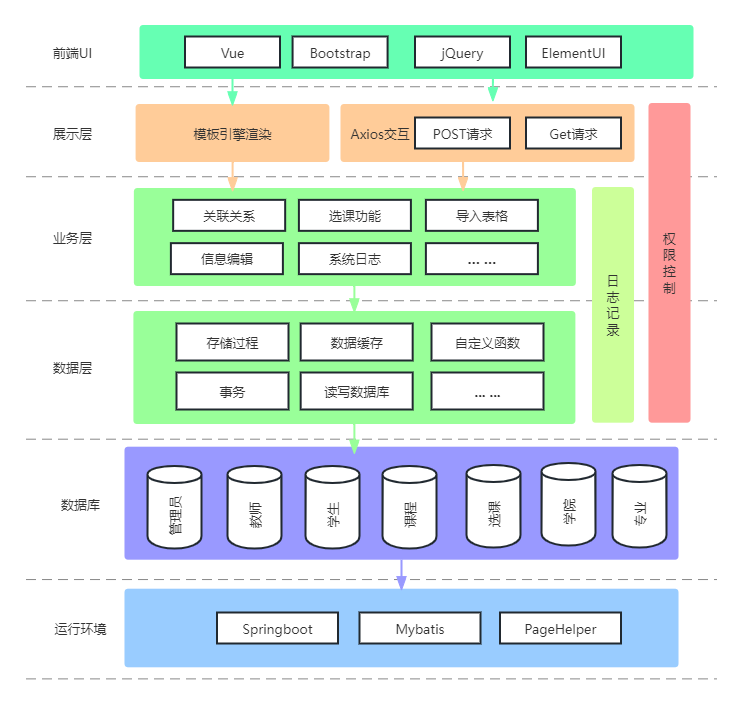
1、提高选课效率：简化选课流程，使学生能够快速、方便地选择所需课程。

2、优化排课管理：为教师和管理员提供有效的排课工具，确保课程安排合理，避免时间冲突。

3、提升系统安全性：采用合适的身份验证机制，保障学生和教师信息的安全性。

4、实现数据统计功能：为管理员提供选课数据统计和分析，帮助优化学校资源分配。

**2. 系统架构**

****

前端UI 部分：Vue是一款流行的响应式数据框架，它是一款轻量级的JavaScript库，用于构建交互式Web界面。

Element UI是一款基于Vue的第三方组件库，它提供了丰富的UI组件和示例，如导航栏、表格、弹窗、日期选择等，可以帮助开发者快速构建出美观、交互式的Web界面。Element UI还提供了可自定义主题和可拖放排序等高级功能。

Bootstrap是Twitter开源的一款前端样式库，它提供了丰富的CSS和JavaScript组件，如网格系统、表单、按钮、导航、弹窗等，可以帮助开发者快速构建出优雅、响应式的Web页面。jQuery是一个快速、简洁的JavaScript库，它被设计用于简化HTML文档遍历、事件处理、动画效果和AJAX等常见操作。它提供了一系列易于使用的API，可以大大简化JavaScript代码的编写。

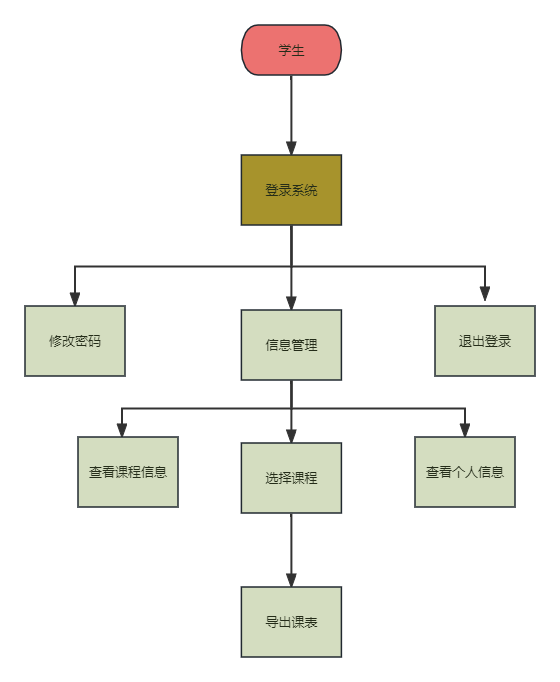
后端：Spring Boot是一款用于快速构建独立的、生产级别的Spring应用程序的开源框架。它简化了Spring应用程序的配置和部署过程，并提供了一系列的默认配置，使得开发者可以更专注于业务逻辑的实现。Spring Boot基于Spring框架，通过自动化配置和约定优于配置的原则，使得开发者可以更快速地搭建起可独立运行的Spring应用程序。

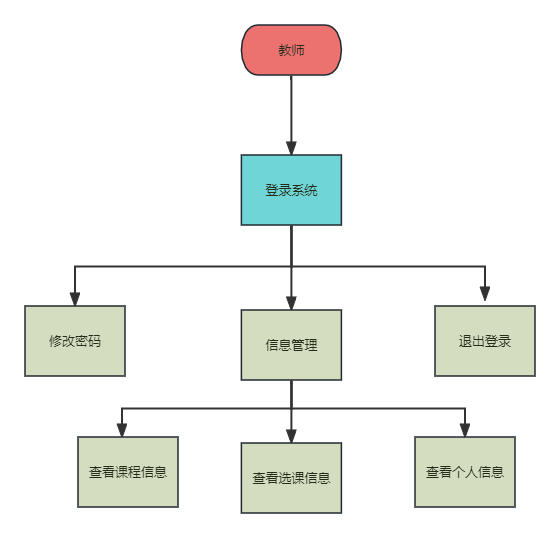
MyBatis是一个开源的Java持久层框架，它简化了数据库访问的过程。与传统的ORM框架不同，MyBatis采用了半自动化的方式，通过XML文件或注解来描述SQL语句和数据库映射关系。开发者可以直接编写原生的SQL语句，同时也支持使用动态SQL来构建复杂的查询。

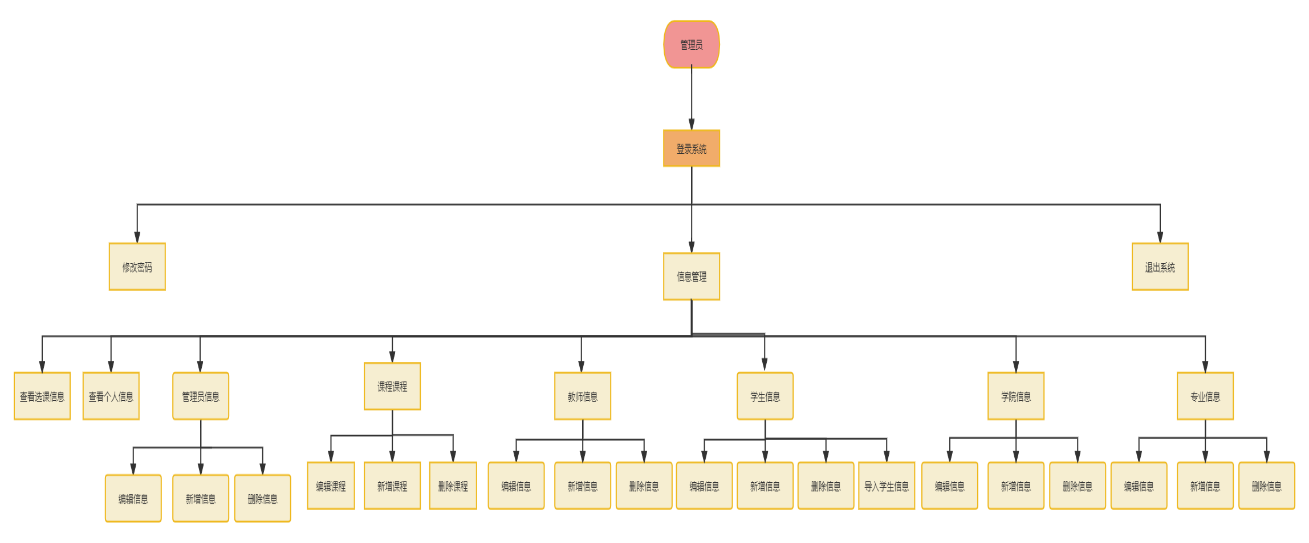
PageHelper是一个开源的MyBatis分页插件，它可以帮助开发者简化在数据库查询中的分页操作。通过引入PageHelper插件，开发者可以使用类似于MySQL的LIMIT语句来实现分页查询，无需手动编写复杂的分页逻辑。

**3. 系统设计**

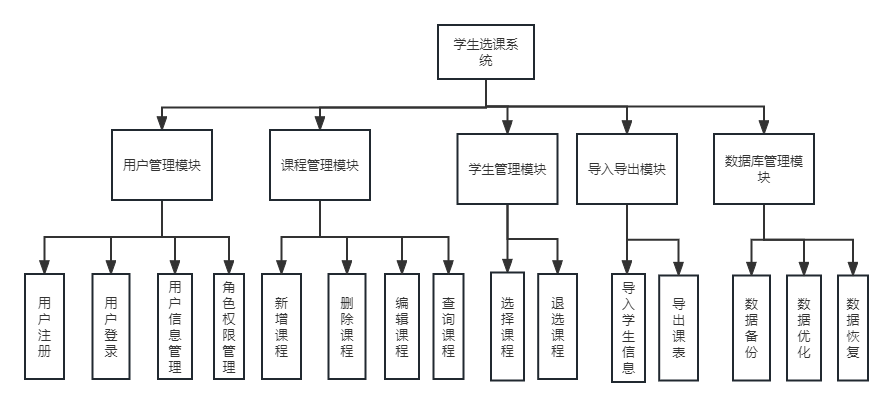
**3.1. 主要业务流程图**

****

****



**3.2 系统功能设计**

****

（1）用户管理模块

用户注册：学生或教师可以通过该功能创建自己的账号，输入基本信息并选择角色。

用户登录：注册过的用户可以通过账号和密码登录到系统中。

用户信息管理：学生和教师可以查看和修改个人信息，如姓名、性别、联系方式等。

角色权限管理：管理员可以设定学生和教师的系统权限级别，以控制其操作范围。

（2）课程管理模块

该模块负责管理所有的课程信息，包括添加、编辑、删除和查询等功能。学生可以查看可选课程列表，包括课程详情、授课教师和时间安排。

（3）学生选课模块

该模块是学生选课系统的核心模块，学生可以浏览系统中的课程列表，了解每门课程的基本信息和开课时间等，通过该模块选择自己喜欢的课程。系统能够根据学生选择的课程和时间进行排课。学生也可以在选课期间或选课后退选已选择的课程。同时，系统还提供了选课结果查询功能，让学生了解自己的选课情况。

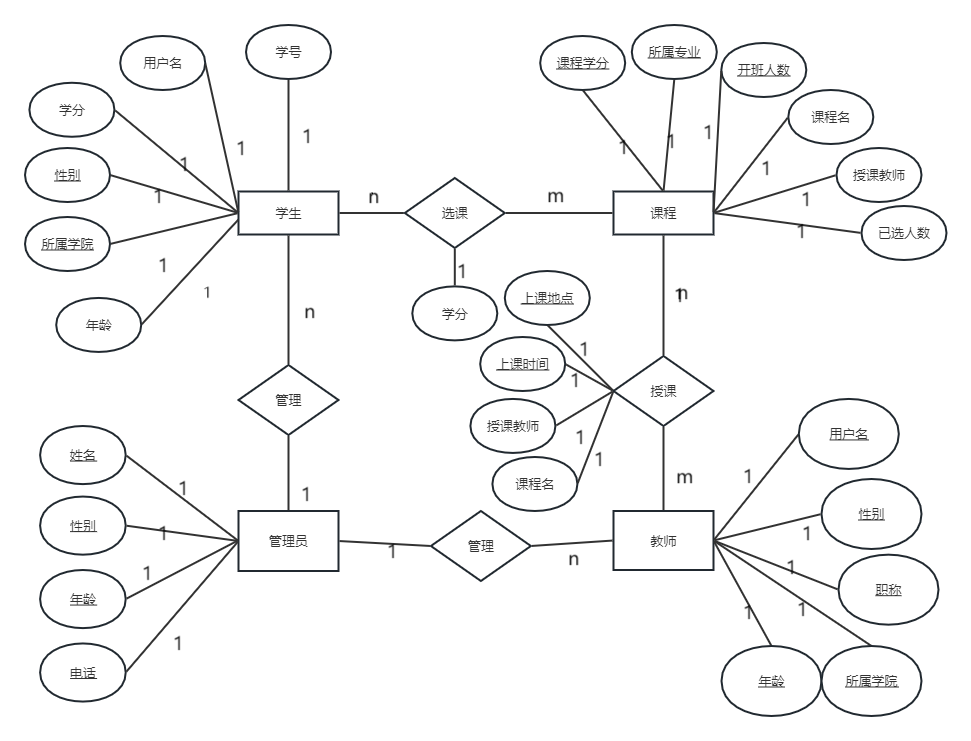
（4）导入导出模块

该模块负责课表的导出和学籍信息的导入功能，可以根据学生的选课信息生成课表，批量导入学籍学期信息。

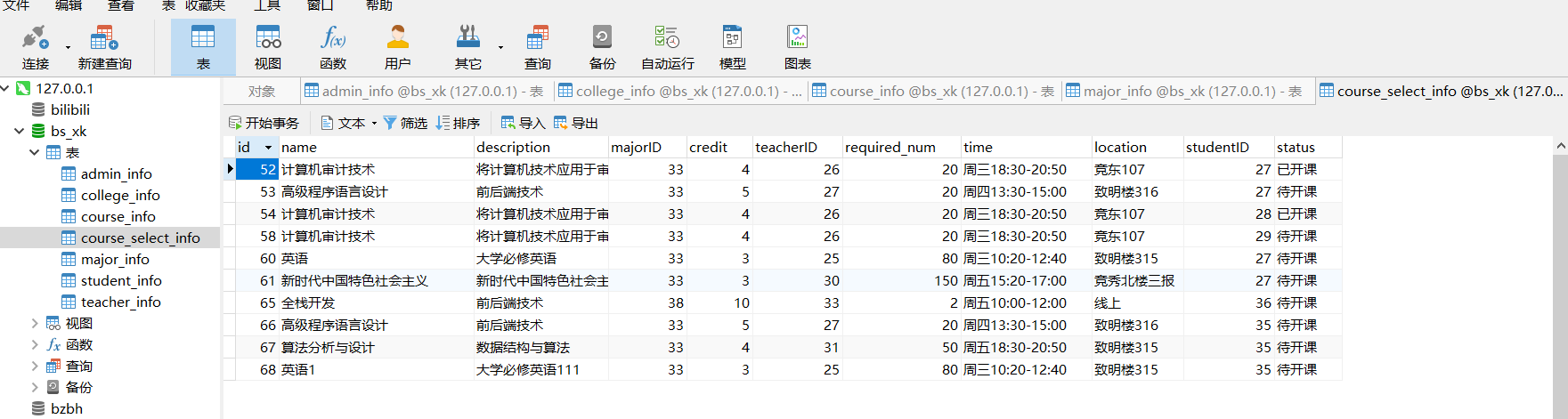
（5）数据库管理模块

该模块负责管理系统的数据库，包括数据的备份、恢复和优化等功能。系统应该采用合适的数据库管理系统，以保证数据的安全性和可靠性。

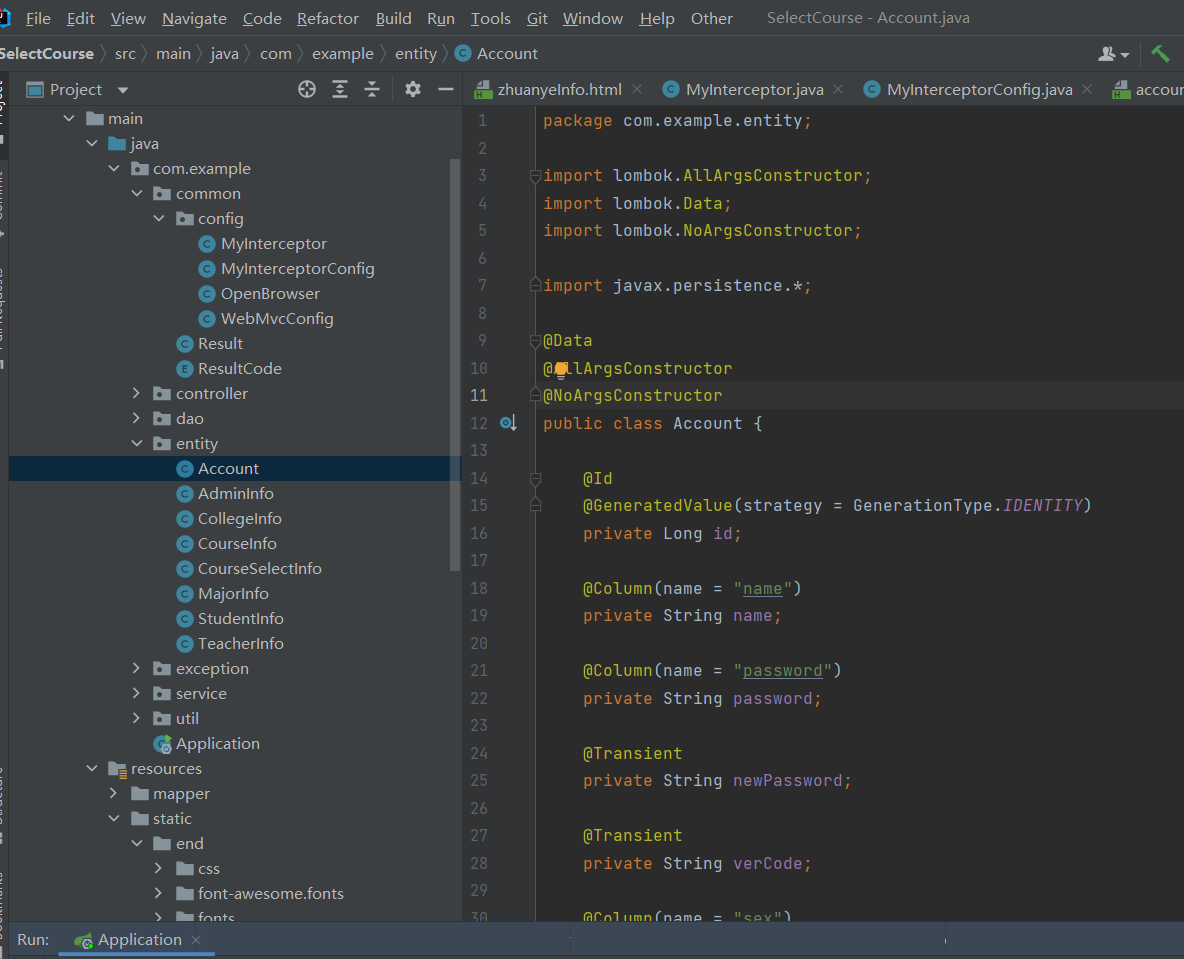
**3.3. 数据库设计**

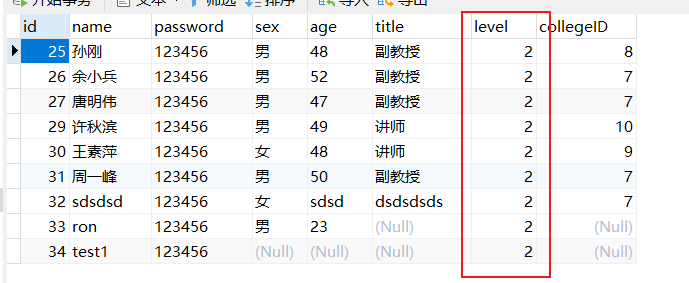
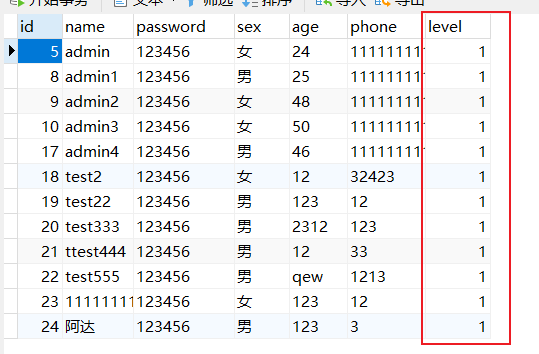
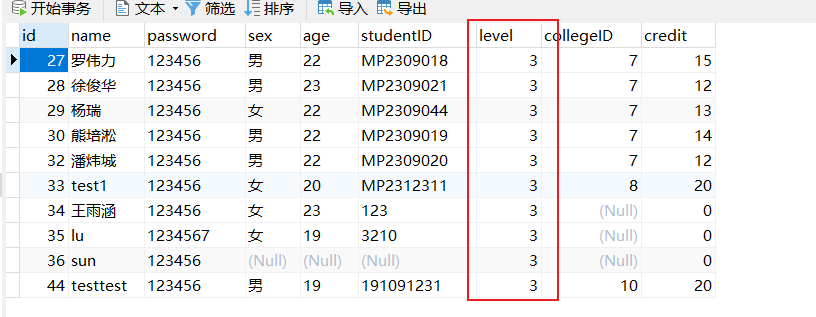
****

数据库一共涉及七张表：

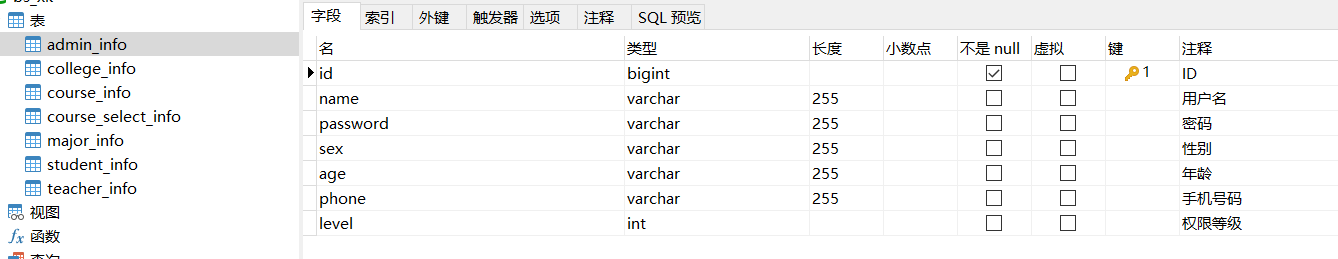


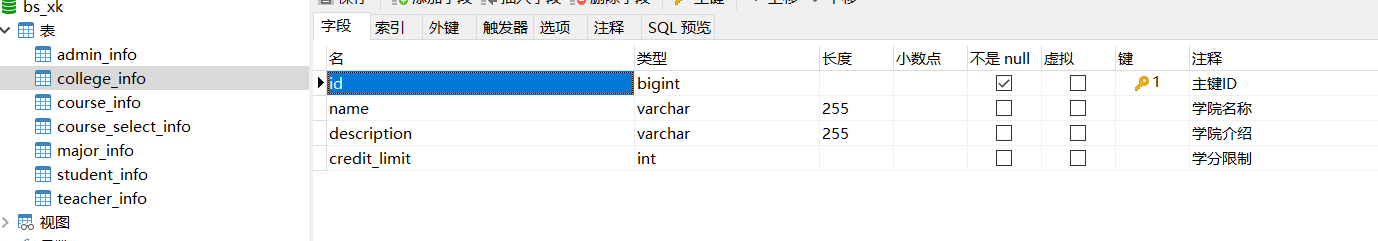
一共涉及到八个实体：

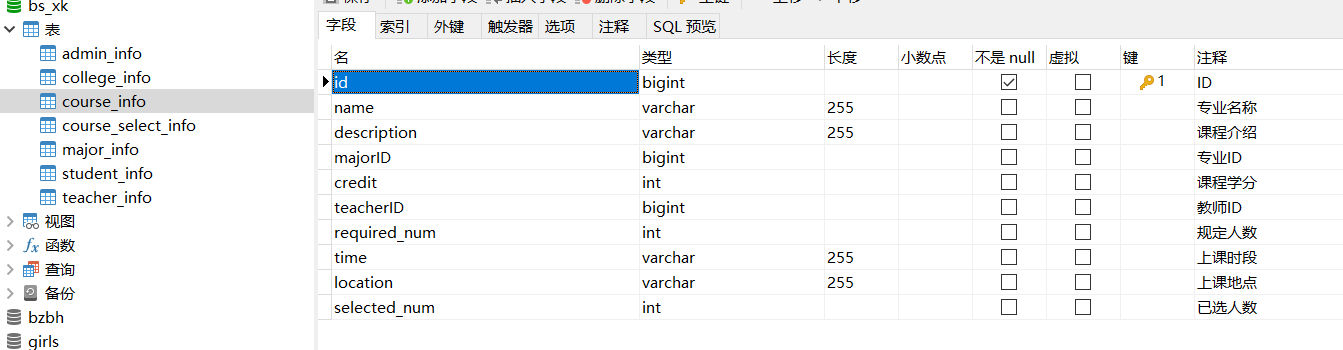
* Accout，是管理员、教师、学生的基类，代码截图：
* AdminInfo，对应管理员表
* CollegeInfo，对应学院表
* CourseInfo，对应课程表
* CourseSelectInfo，对应选课表
* MajorInfo，对应专业表
* StudentInfo，对应学生表
* TeacherInfo，对应教师表

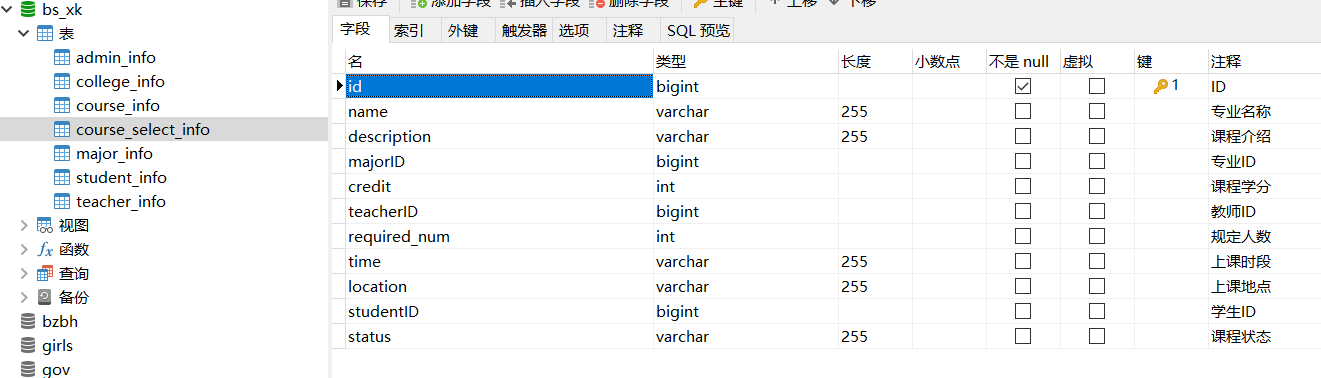
整个系统的核心是管理员表、学生表、教师表都维护了一个level字段。其中level==1代表是管理员，level==2代表教师，level==3代表学生。  


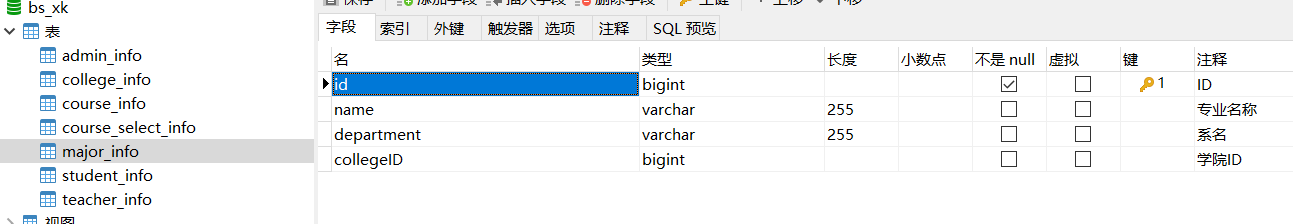
各个表的设计：

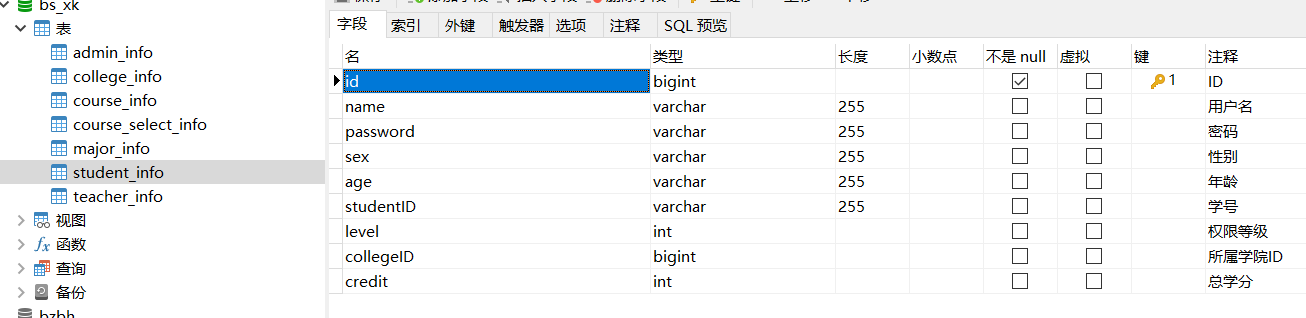


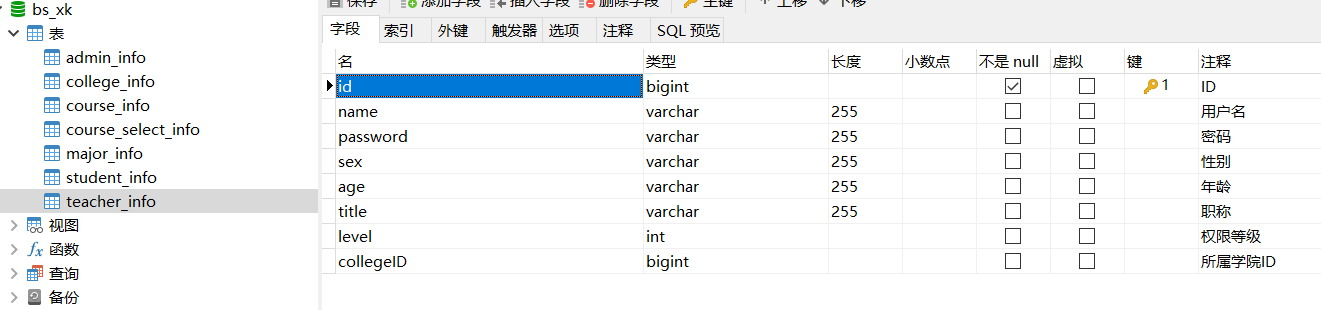






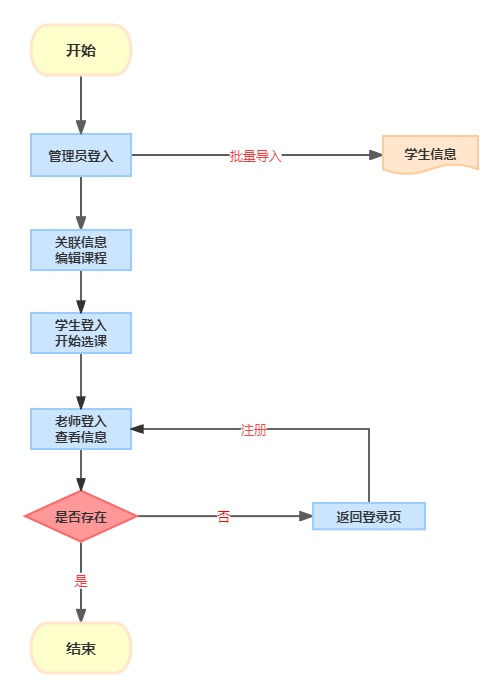


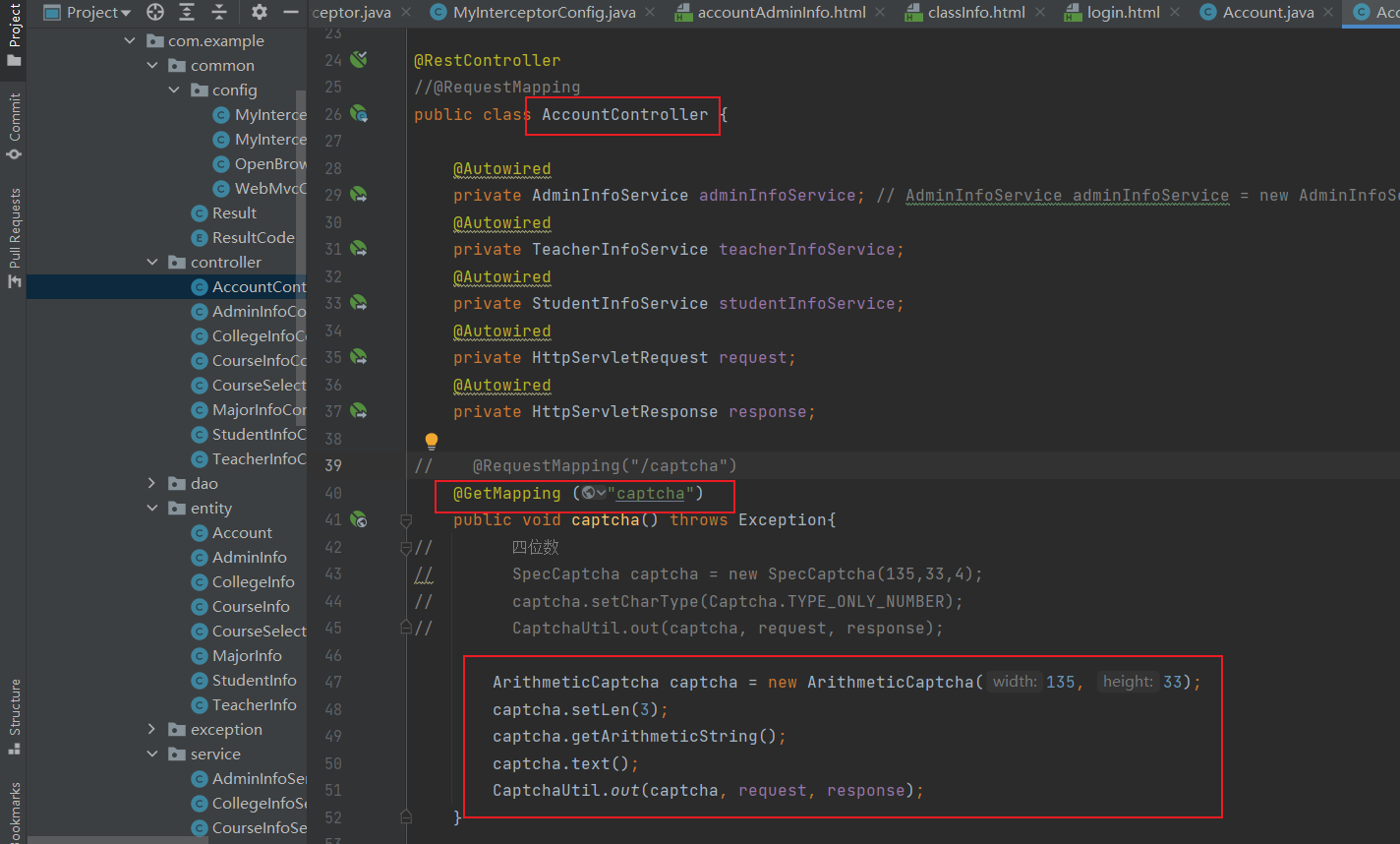
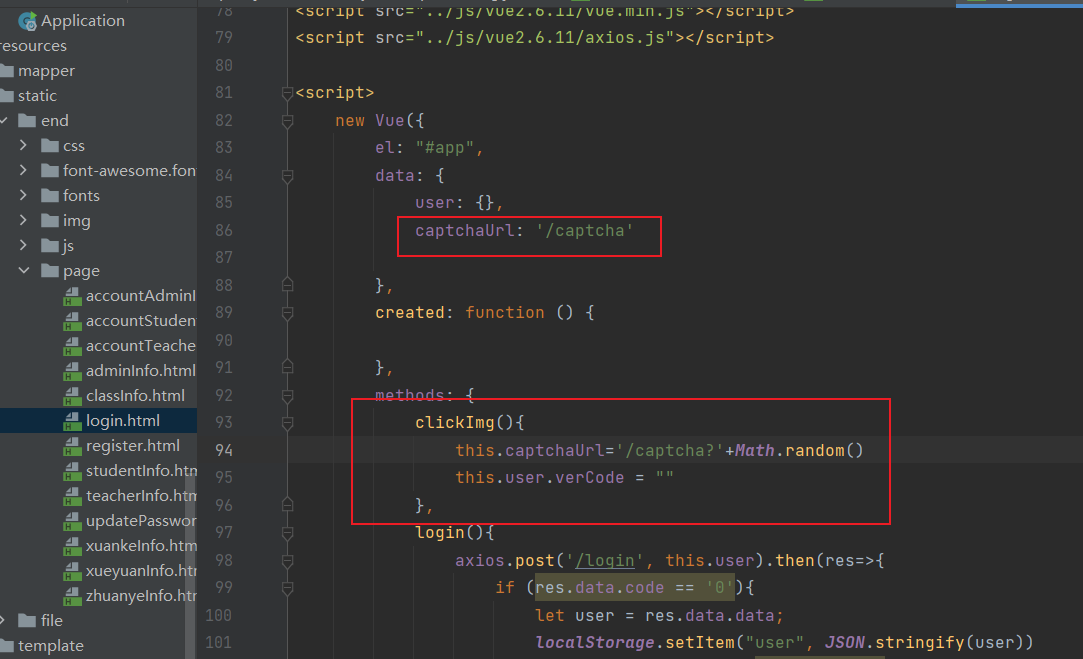
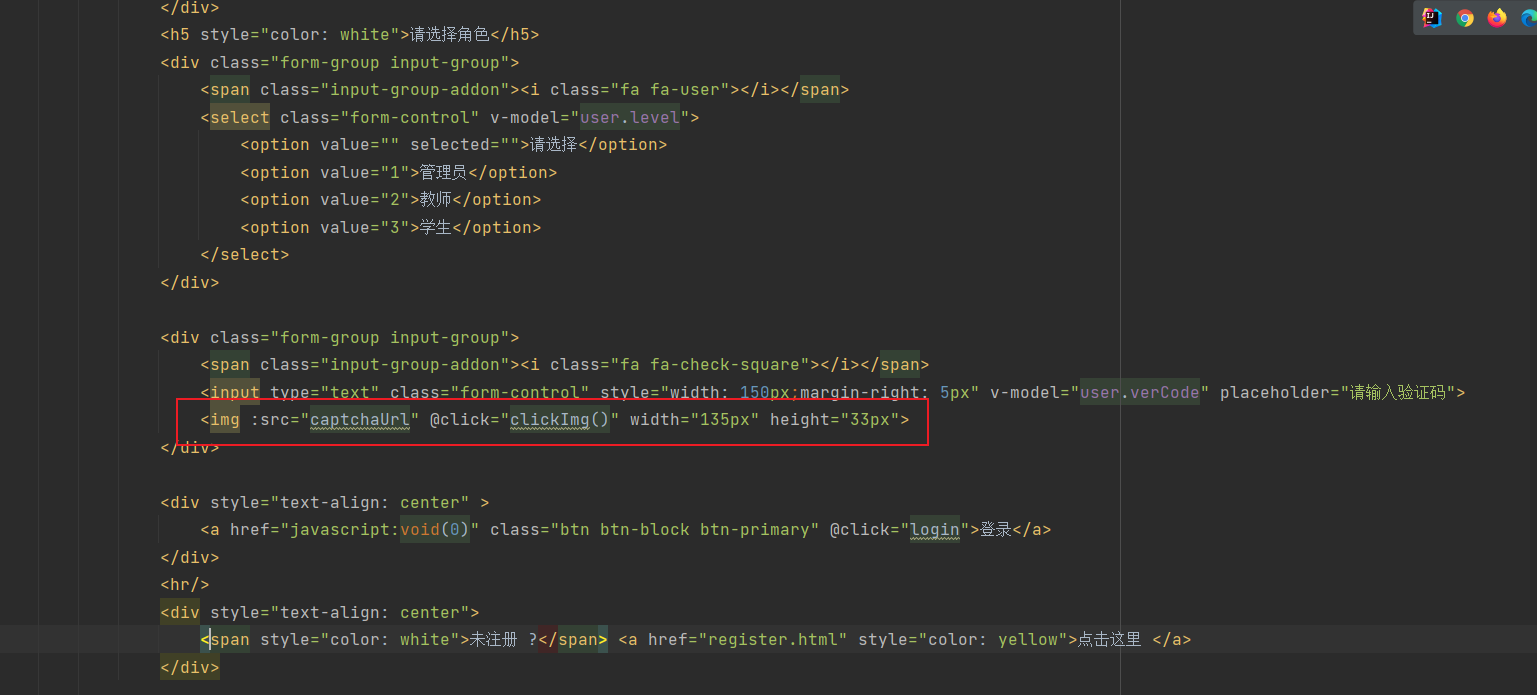
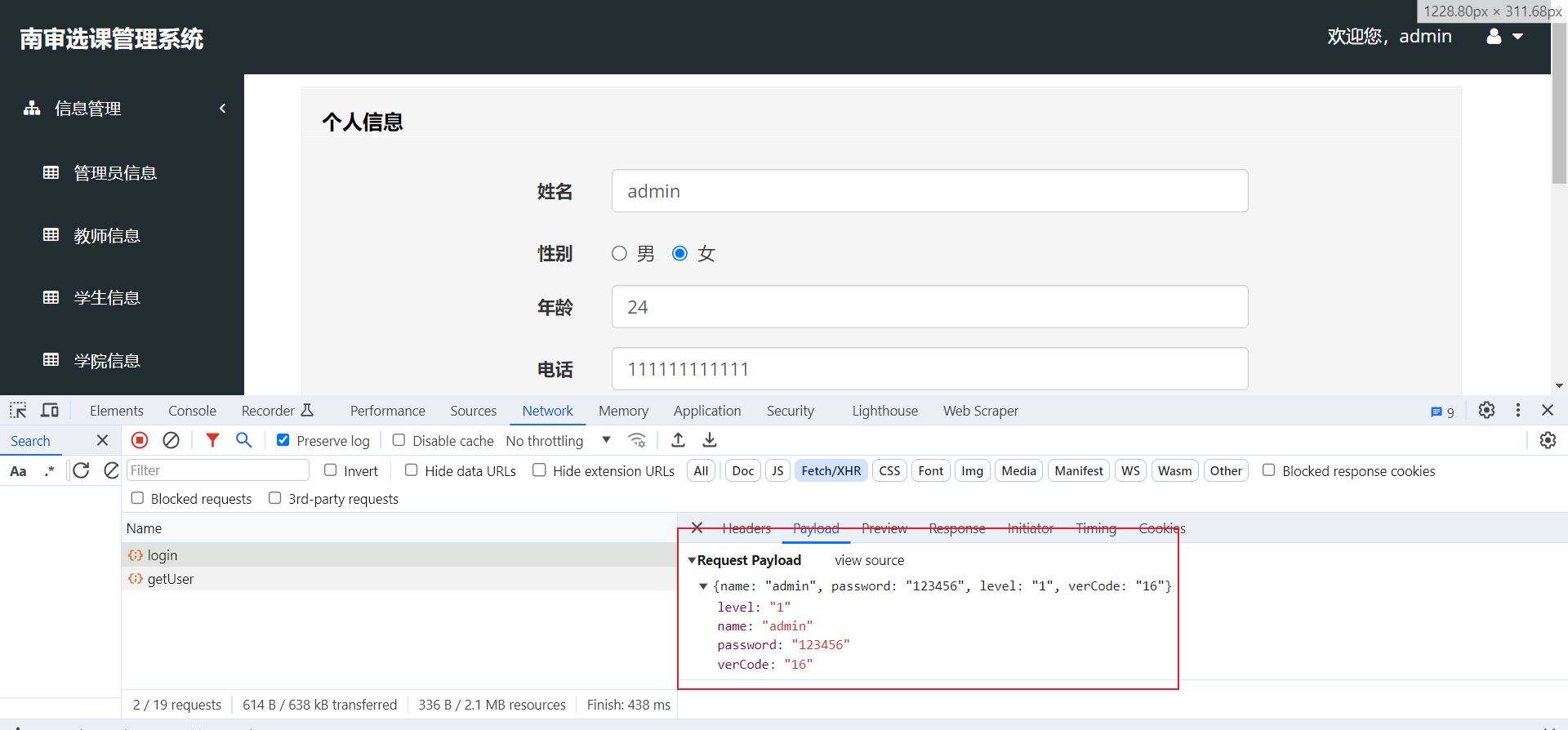
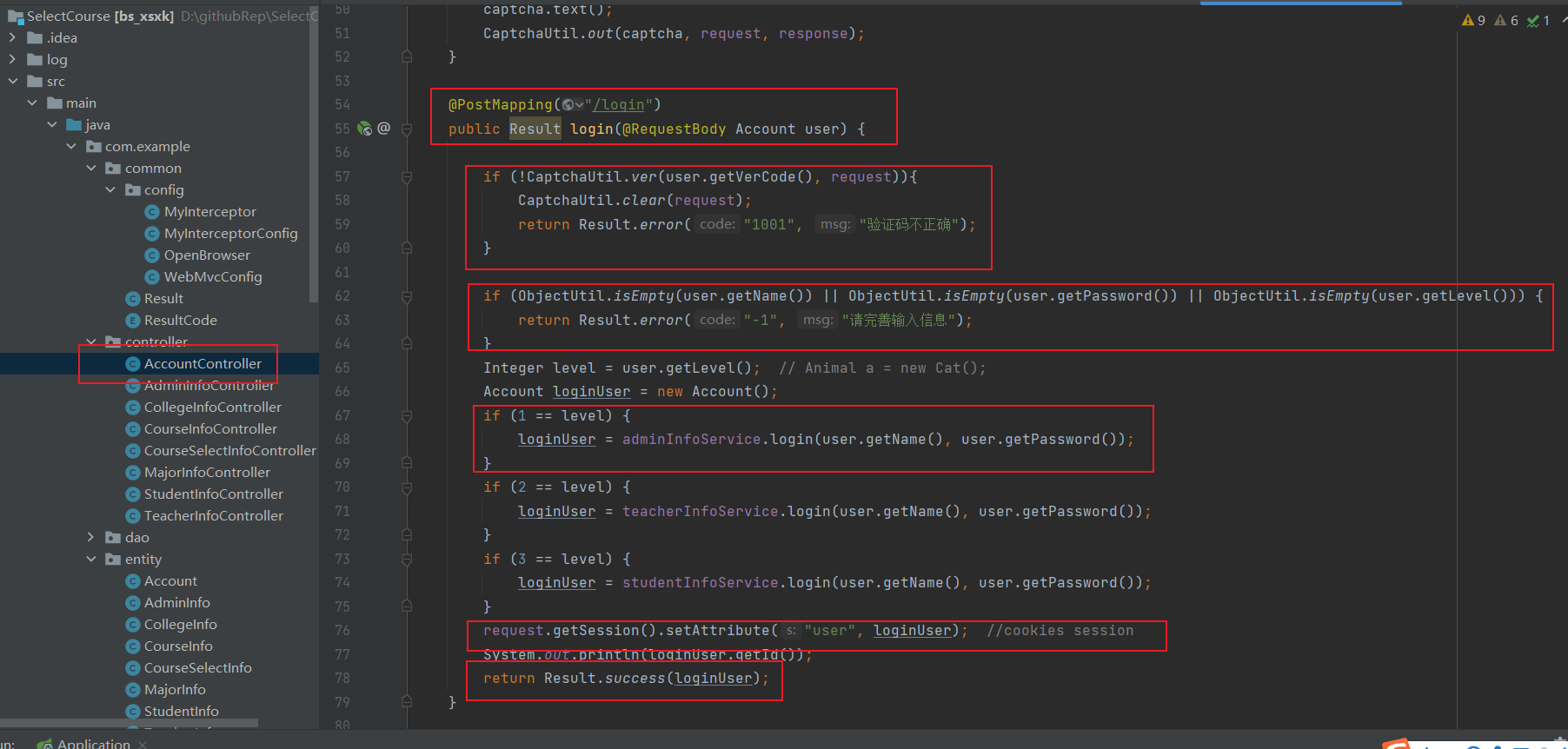


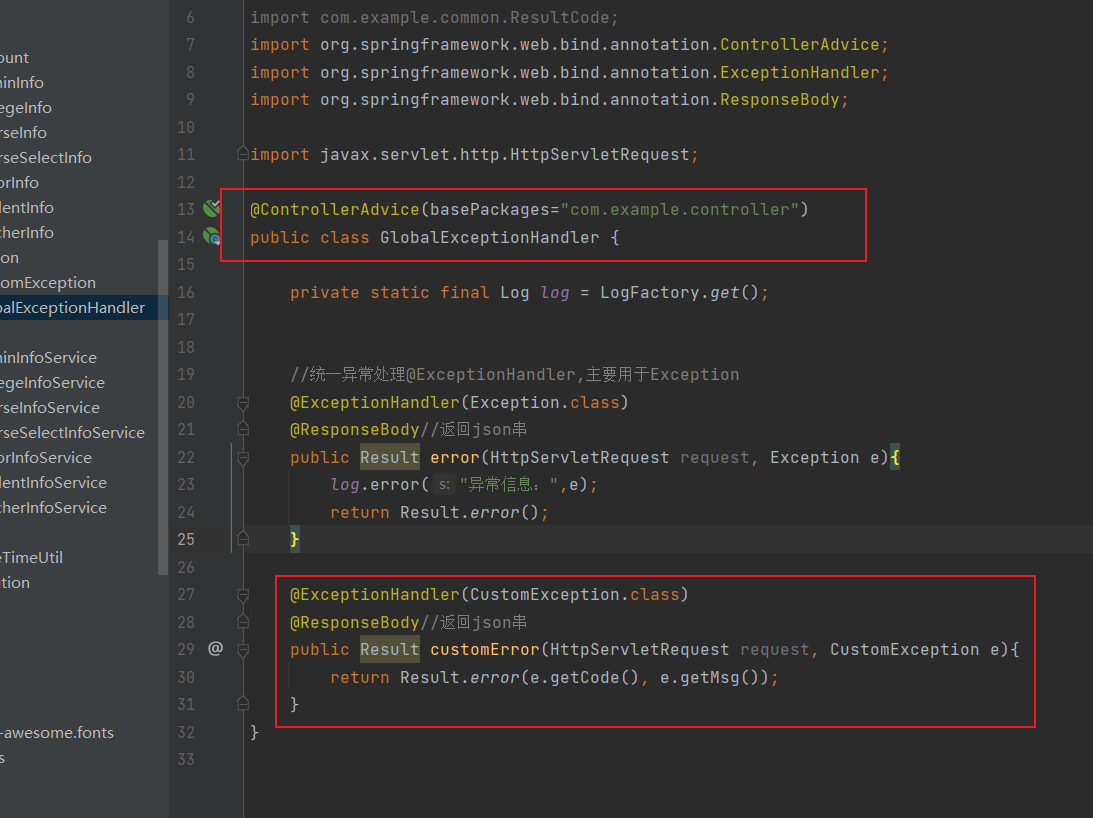
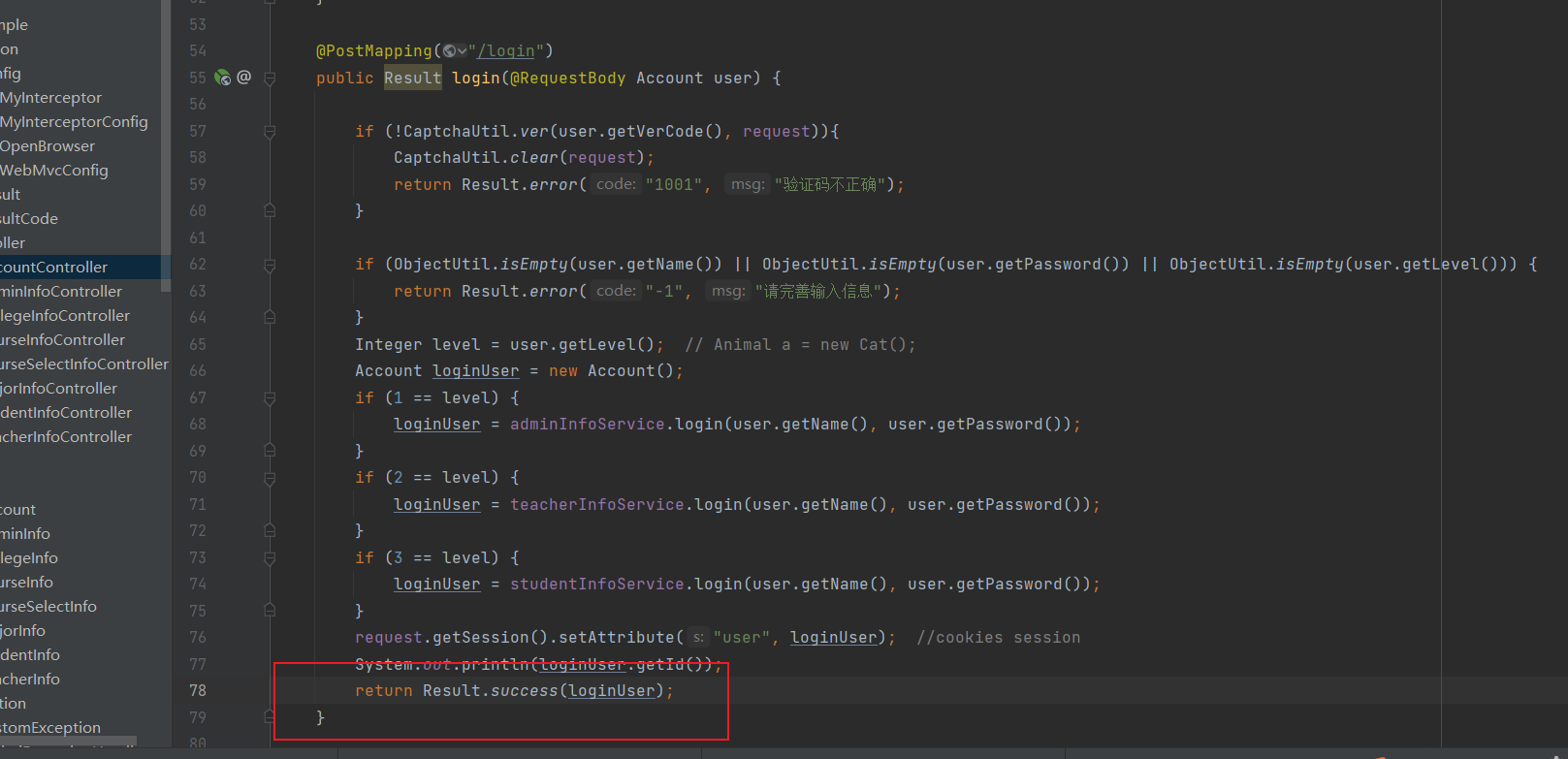


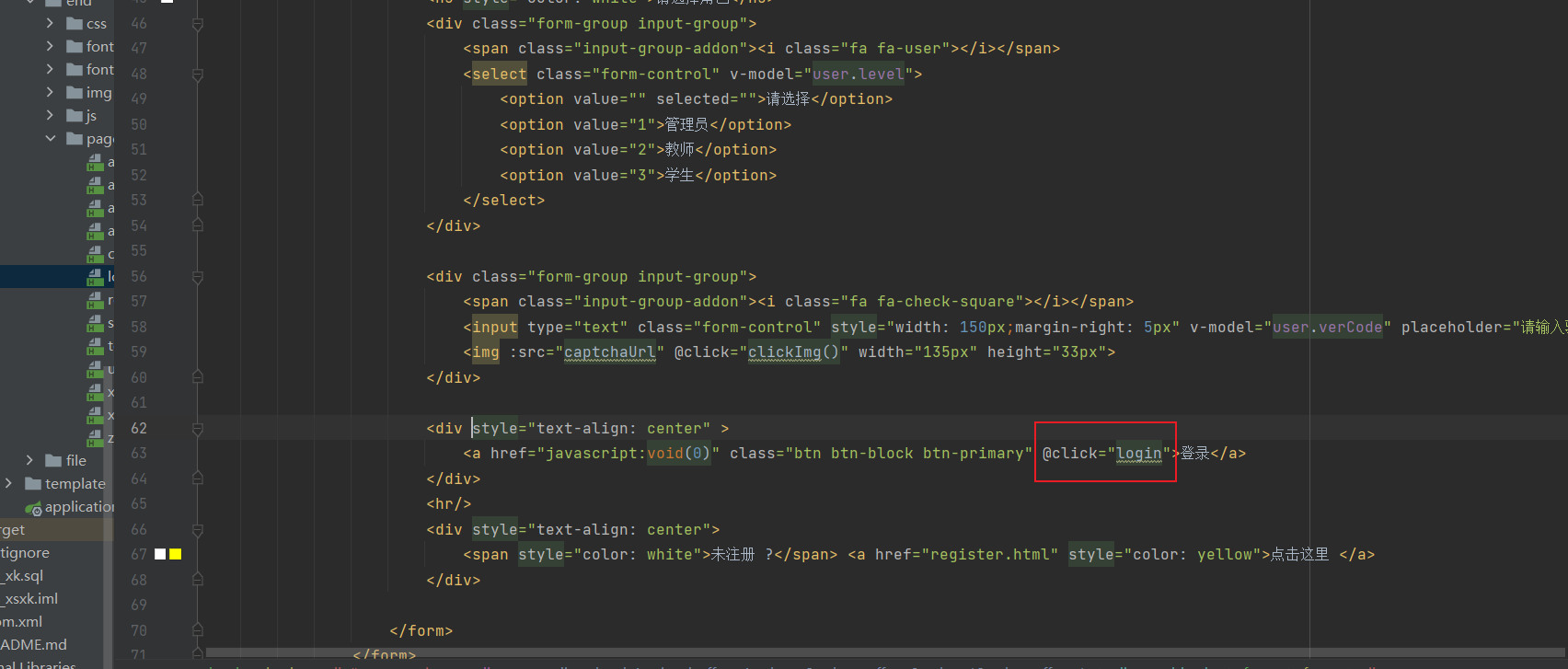
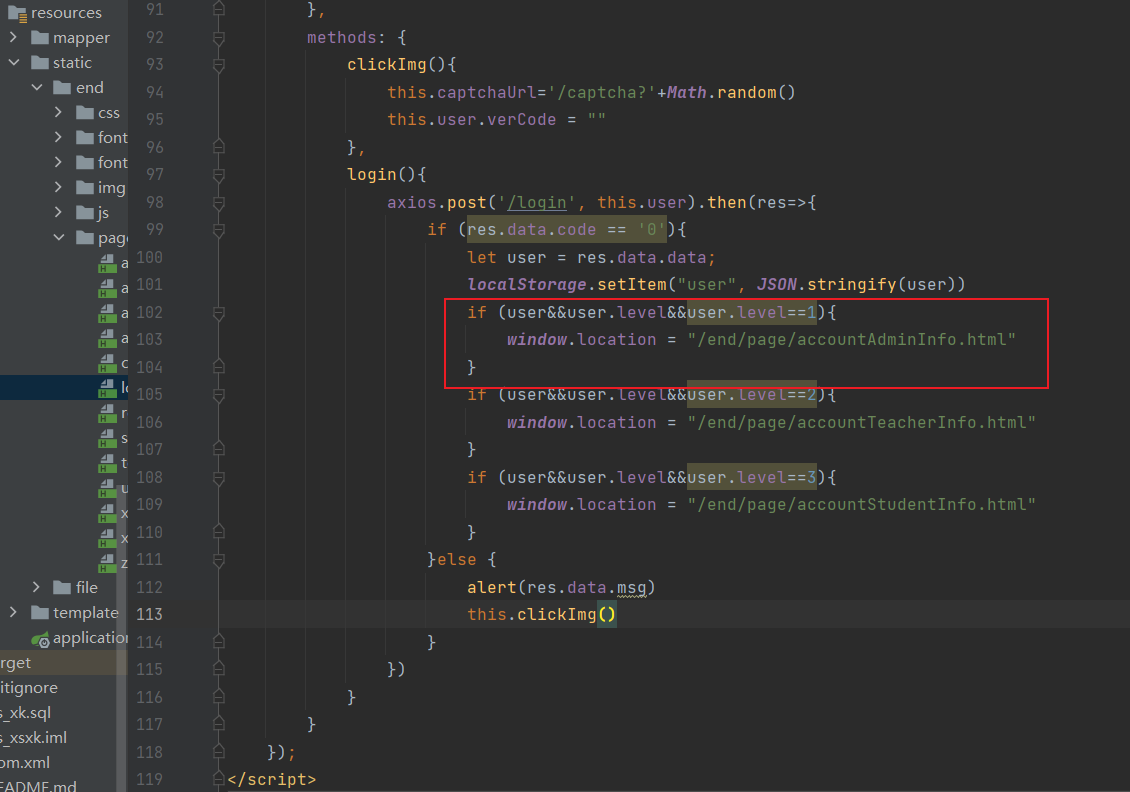
1. **系统测试**

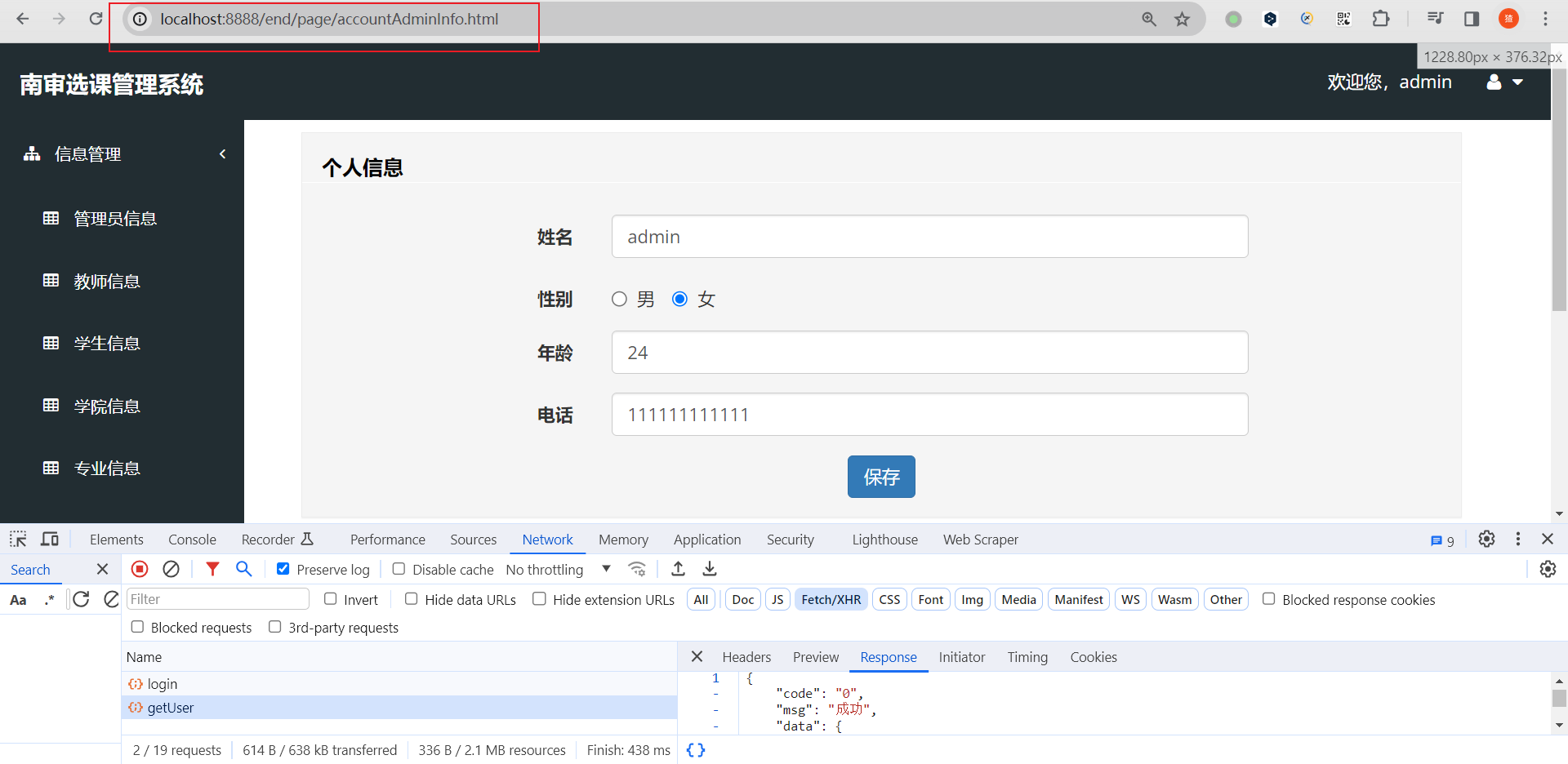
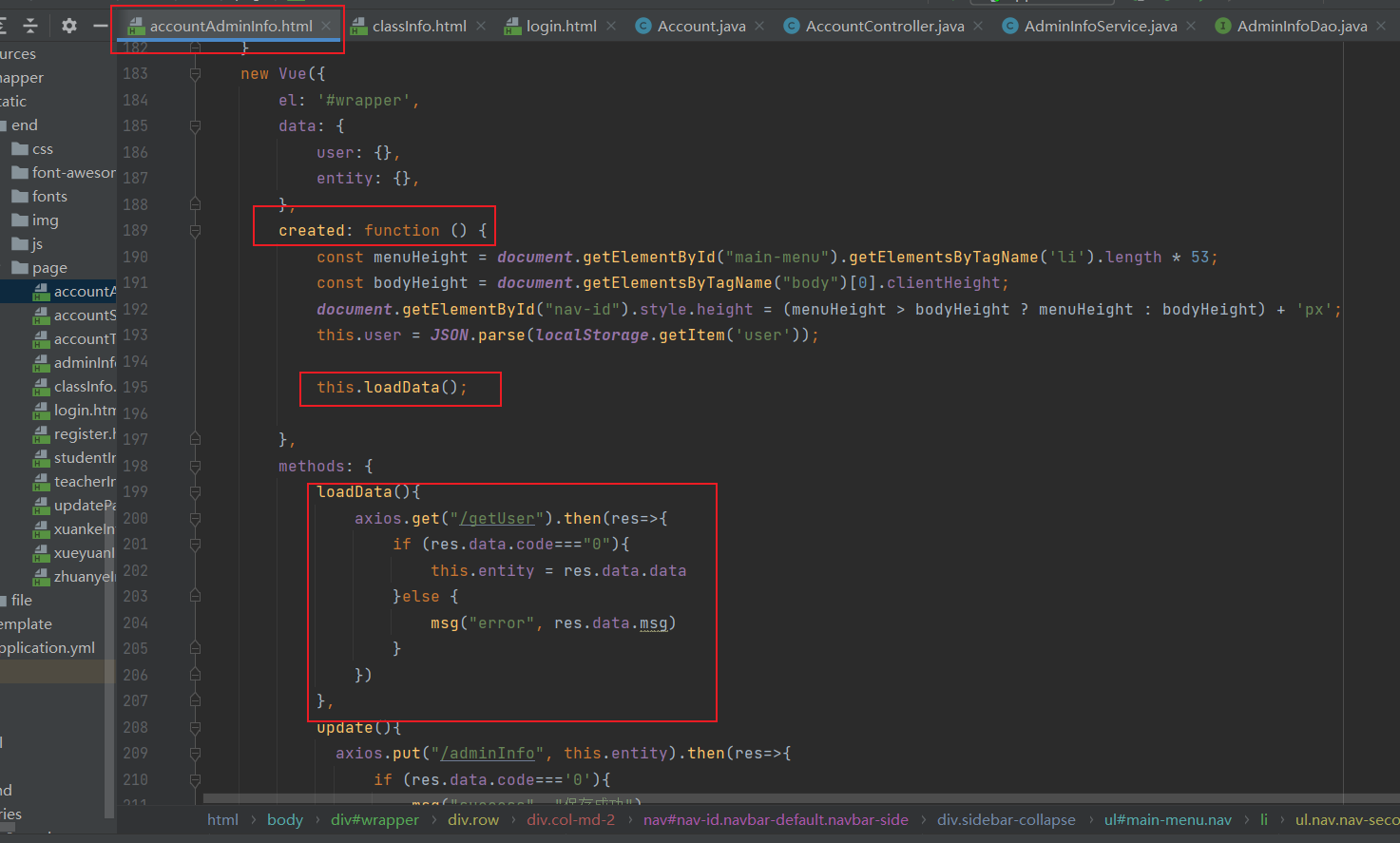
系统测试流程图

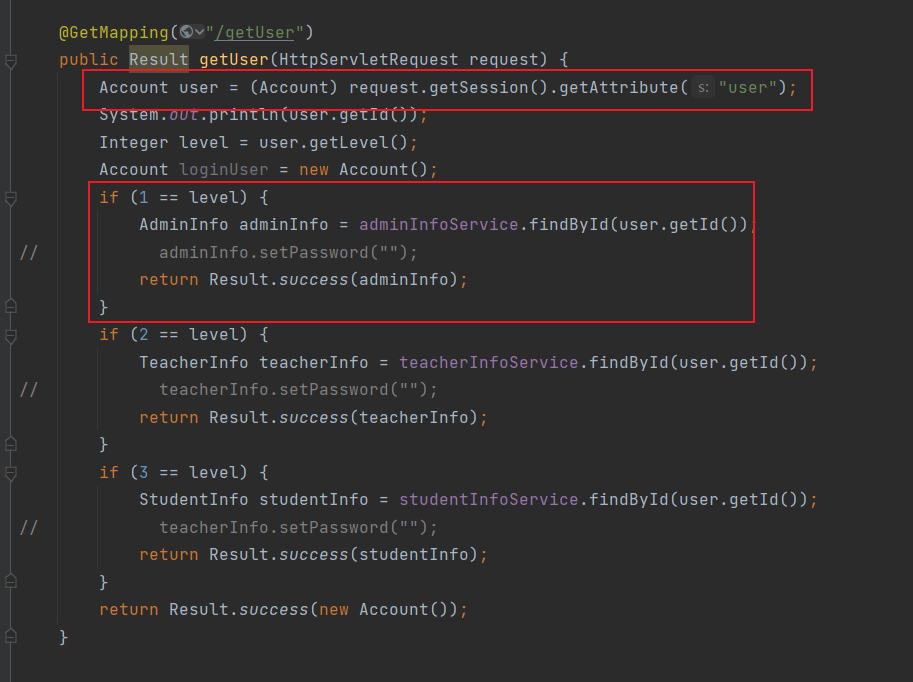
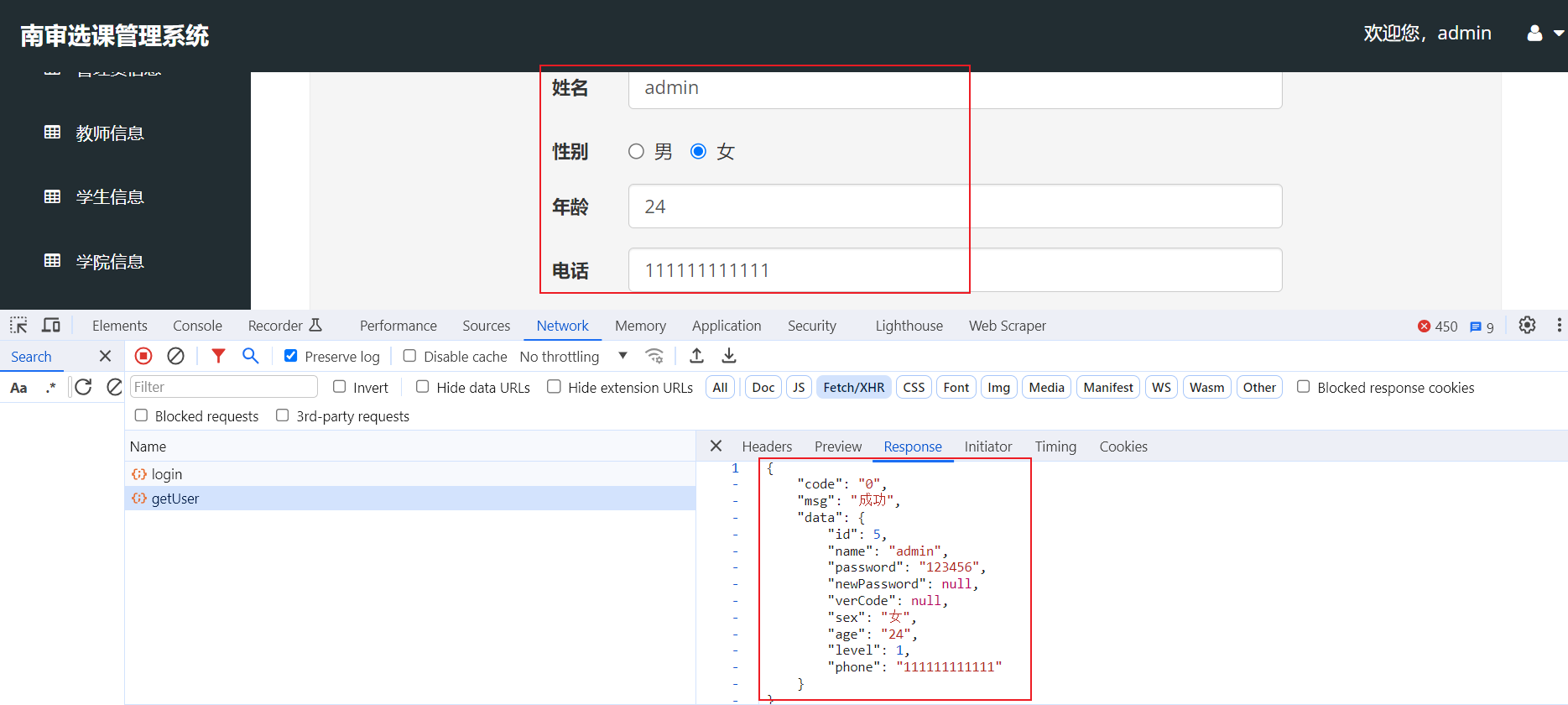
****

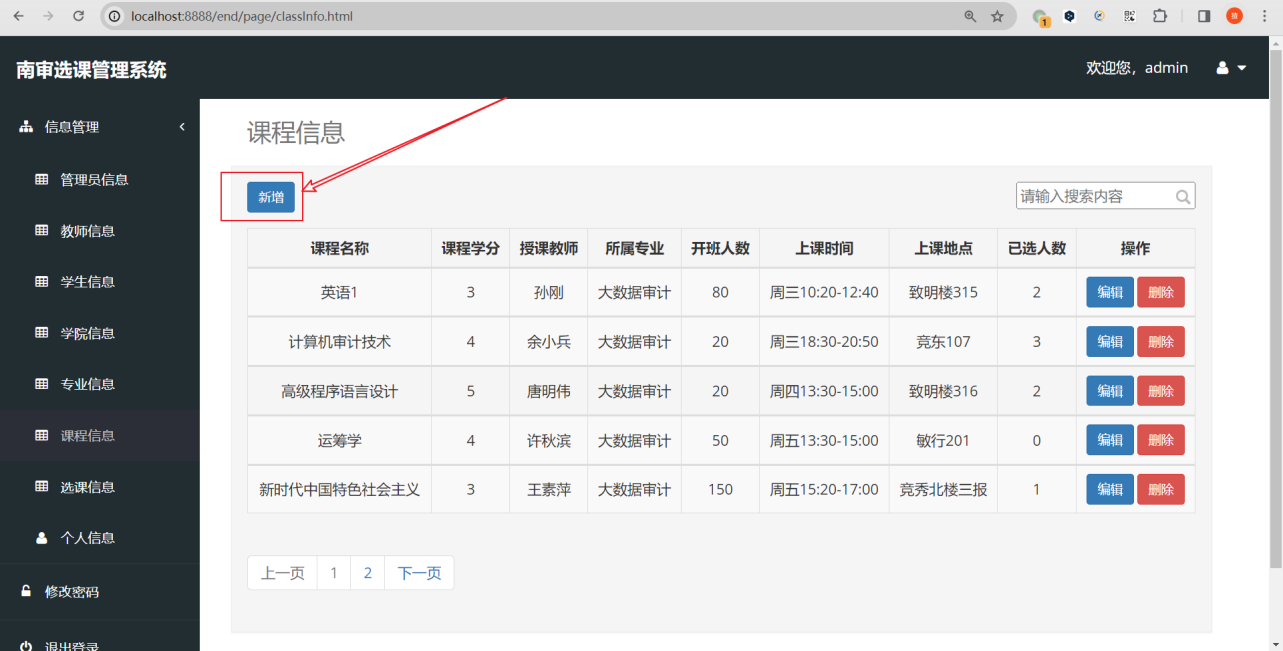
1. 启动系统，进入首页： 在后端的AccountController中有一个/captcha的get接口，这个方法就是实现的验证码功能。而在前端的登录页中，有一个变量captchaUrl为字符串'/captcha'img标签通过src属性向后端请求验证码图片并展示在此处
2. 输入相关信息，点击登录。发出两条接口：。本次请求对应的payload，即在登录页输入的相关信息。而在后端的AccountController中提供了一个/login的post接口：。这个方法主要实现了以下流程：
   1. 首先校验验证码是否正确；
   2. 接着校验名字、密码、身份是否完整
   3. 其次按照前端给定的level请求对应的service
   4. 返回统一结果类

* 这里我们登陆的是管理员身份，那么进入68行adminInfoService.login。首先进入dao层中查询是否有符合的数据，dao层代码如下：。通过Mybatis提供的Select标签拼写一个SQL语句。如果在数据库中没有检索到对应的管理员数据，那么向外抛出一个自定义异常CustomException("-1","用户名、密码和角色错误");，这个异常会被全局异常处理区捕获，统一返回给前端。这里给出的信息都是正确无误的，因此在controller层最后会执行return语句：

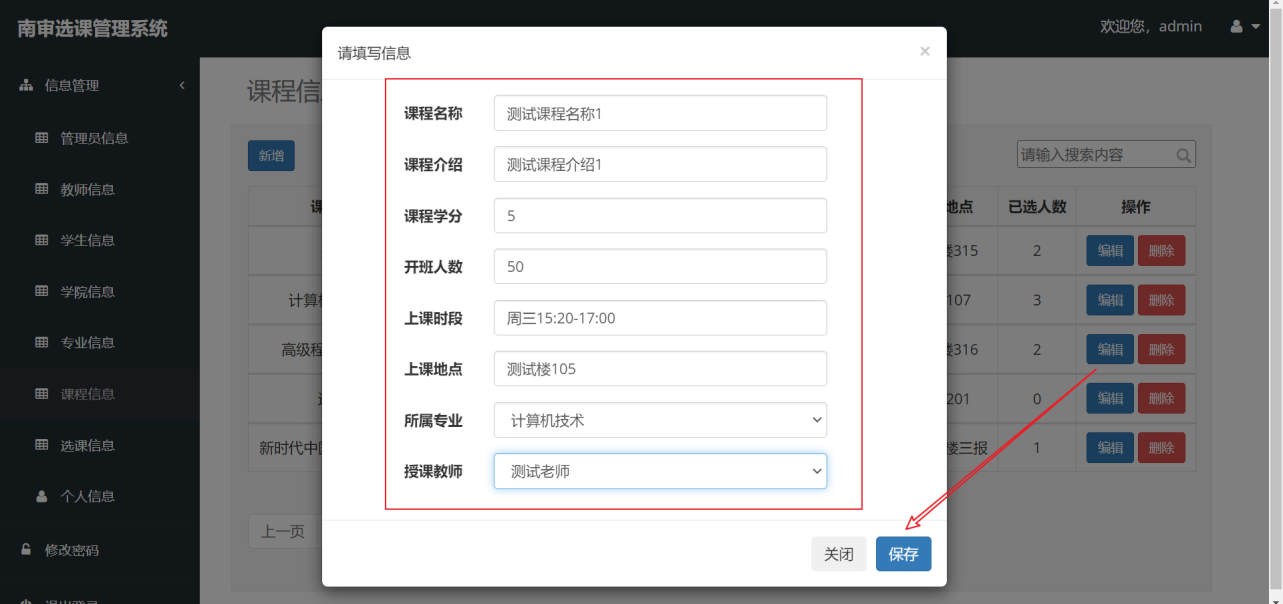
登陆按钮绑定了login函数：Login的实现逻辑是：

因为在登录的时候，我们是以管理员身份登陆的，并且后端统一结果类的泛型也是Account实体类，那么现在就会执行第103行。因而进入管理员页面：。又因为在accountAdminInfo.html中有生命周期函数created，里面调用函数loadData()。即此页面初始化成功后就向后端'/getUser'发起get请求，并且成功后把后端返回的数据复制给entity对象：

entity是vue的响应式数据，当获得数据后就展示，做到了页面一初始化成功就把管理员信息回显到表单上。在后端的/getUser接口中，会先从session中取出已经登录校验成功的user对象，然后根据level字段去service中查询数据。因此做到了回显

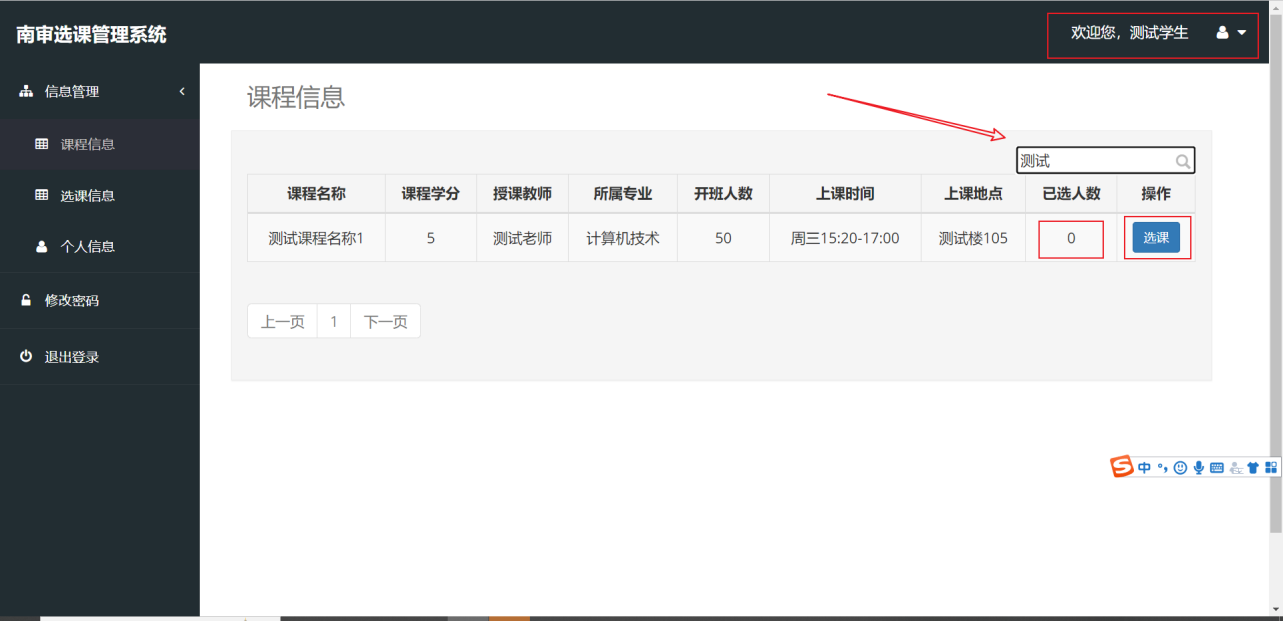
以上便是一个完整的登陆流程，进入课程信息页面，并点击新增

以admin用户新增一门测试课程的相关信息，并点击保存



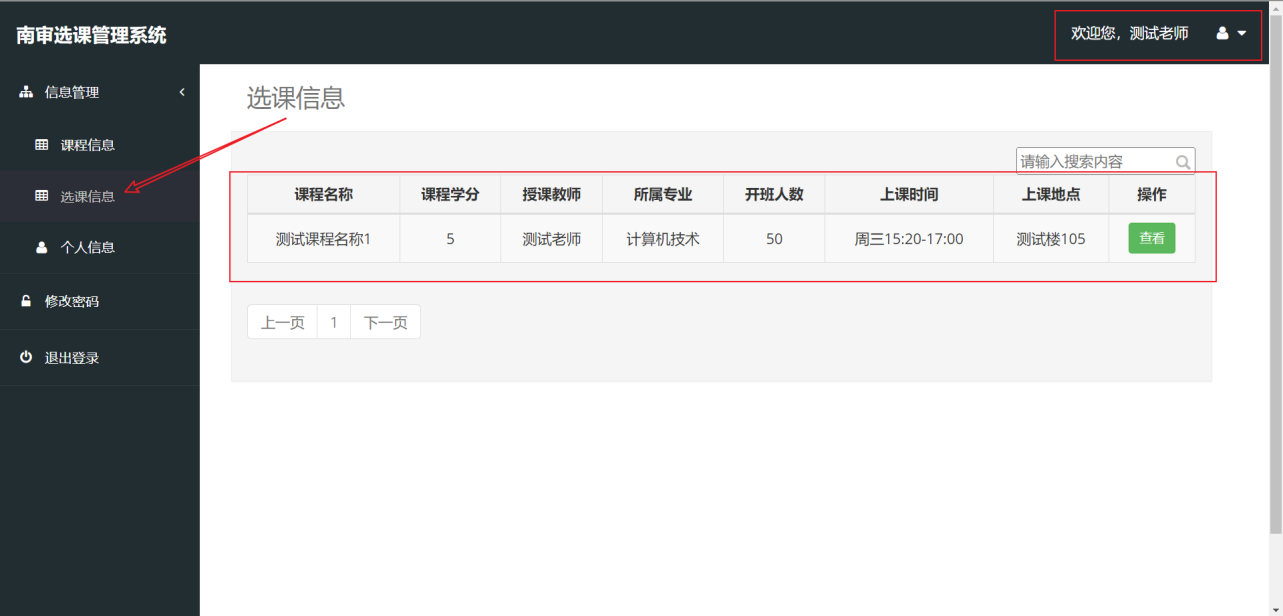
点击下一页可以看见新增的测试课程已经在分页的最后一条

以测试学生身份登陆系统，进入课程信息页面，在搜索框输入测试，单击回车，出现测试课程



点击选课，弹出选课成功提示框，并且已选人数+1

再次点击选课会弹出警告提示框

以测试老师身份登陆系统，进入选课信息页面，结果如下

回到admin人员，查看课程信息页面如下