1. 运行环境
   1. 软件

本系统主要分为vue前台、java后台、mysql数据库三部分。

Vue前台的运行环境为node.js18。

Java后台的运行环境为jdk1.8。

Mysql数据库的运行环境为mysql5.7。

以上三部分均需要运行于服务器操作系统Ubuntu。

* 1. 硬件

本系统的硬件环境为2核cpu，4gneic，40g硬盘的华为云服务器。

1. 部署说明
   1. 前台部署

本系统前台基于vue开发，使用vue脚手架vite打包。以管理员身份运行控制台，cd到前台项目目录下，输入指令npm run build，指令执行完毕后，会在项目目录下生成dist文件夹，该文件夹中的文件为打包后的前台代码。将代码上传至服务器/home/java/static目录下，即可完成前台代码部署。

* 1. 后台部署

本系统后台为基于java开发的ssm项目，使用java脚手架springboot，项目管理工具maven。在项目目录下运行maven build指令，指令执行完毕后会在项目目录下的target目录下生成后台项目的jar包，将该jar包上传至服务器/home/java目录下，运行指令nohup java -jar xxx.jar即可完成后台代码部署。

* 1. 总结

在java运行环境，vue运行环境，前台代码部署，后台代码部署，服务器端口开放等操作均完成后，项目即可正常运行访问。

1. 使用步骤

先打开浏览器，然后在浏览器输入栏中输入<http://123.249.31.57:82/index.html>，即可访问系统登录页面。在系统登录页面中输入用户名、密码后，点击登录，在验证用户名、密码通过后，即可跳转到系统主页面使用系统。

1. 编码规范

本系统统一使用utf8mb4作为数据传输编码。命名均使用驼峰命名法。以每张表作为一个单独的模块进行抽象，按照高内聚低耦合的设计原则进行模块的设计划分。

1. 模块设计
   1. 登录
      1. 页面文件



* + 1. 核心技术点

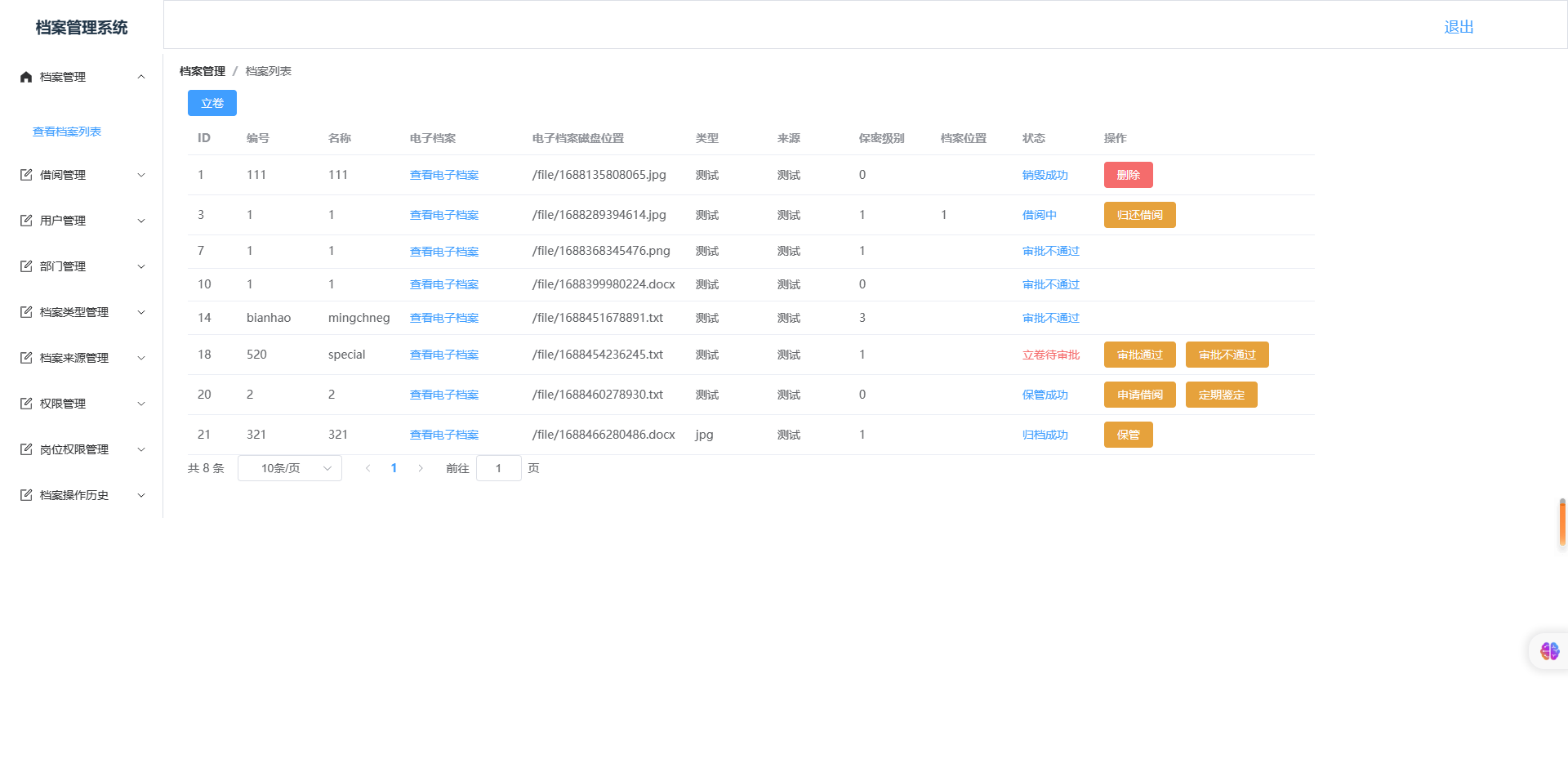
前台：Axios请求、vue-router路由。

后台：jwt生成Token，ssm接收并验证数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/login") public Message login(String username, String password) {  User user = this.userService.selUserByUsernameAndPassword(username, password);  if (user == null) {  return new Message("登录失败", 0, null);  } else {  String token = JWT.*create*().setPayload("username", username).setKey(key.getBytes(StandardCharsets.*UTF\_8*)).sign();  return new Message("登录成功", 1, token);  } } |

* 1. 档案管理
     1. 页面文件



* + 1. 核心技术点

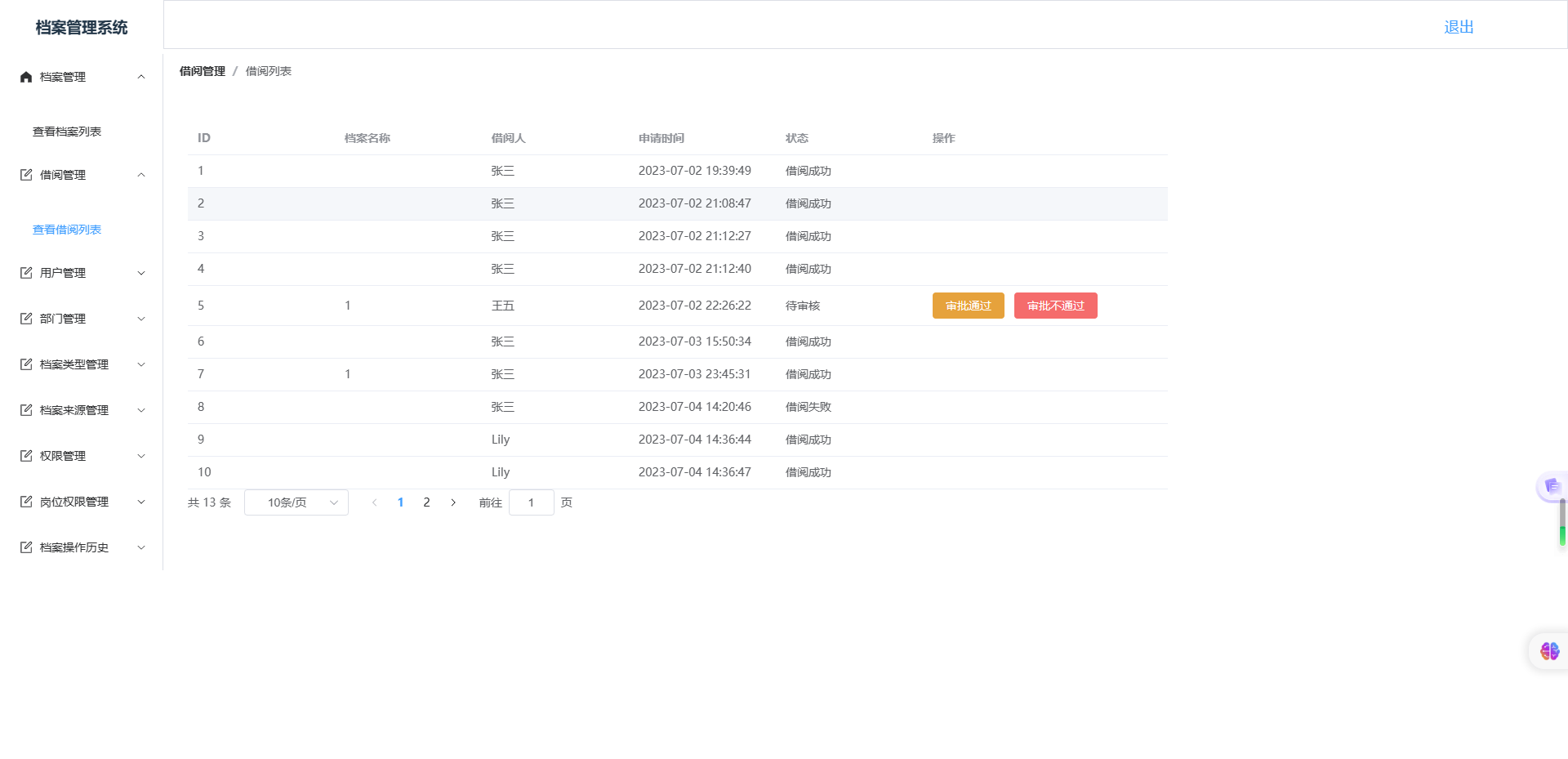
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selDanganPage") public Page selDanganPage(int page, int limit) {  Page page1 = this.danganService.selArchivePage(page, limit);  return page1; } |

* 1. 借阅管理
     1. 页面文件



* + 1. 核心技术点

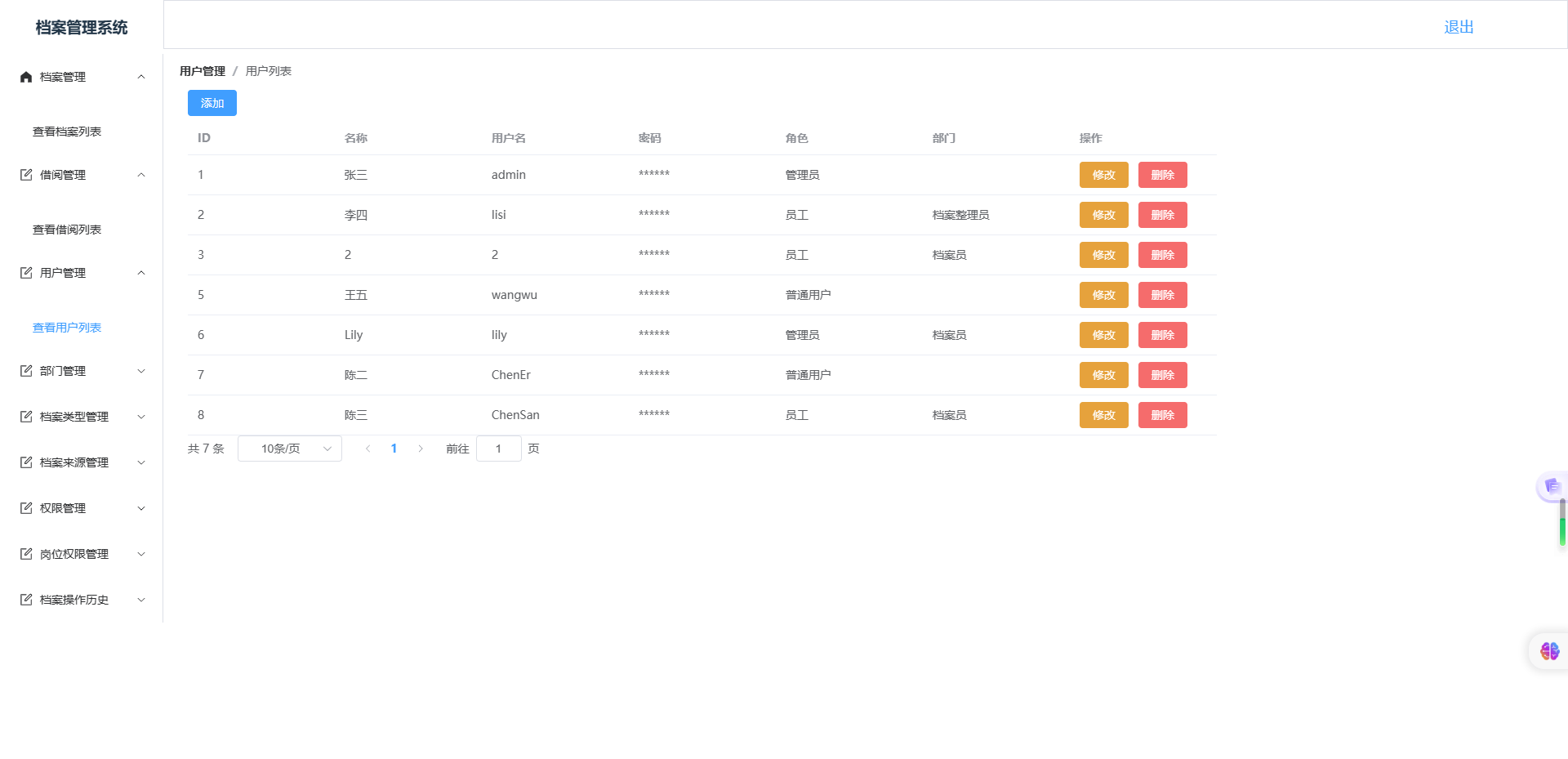
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selJieyuePage") public Page selJieyuePage(int page, int limit) {  Page page1 = this.jieyueService.selJieyuePage(page, limit);  return page1; } |

* 1. 用户管理
     1. 页面文件



* + 1. 核心技术点

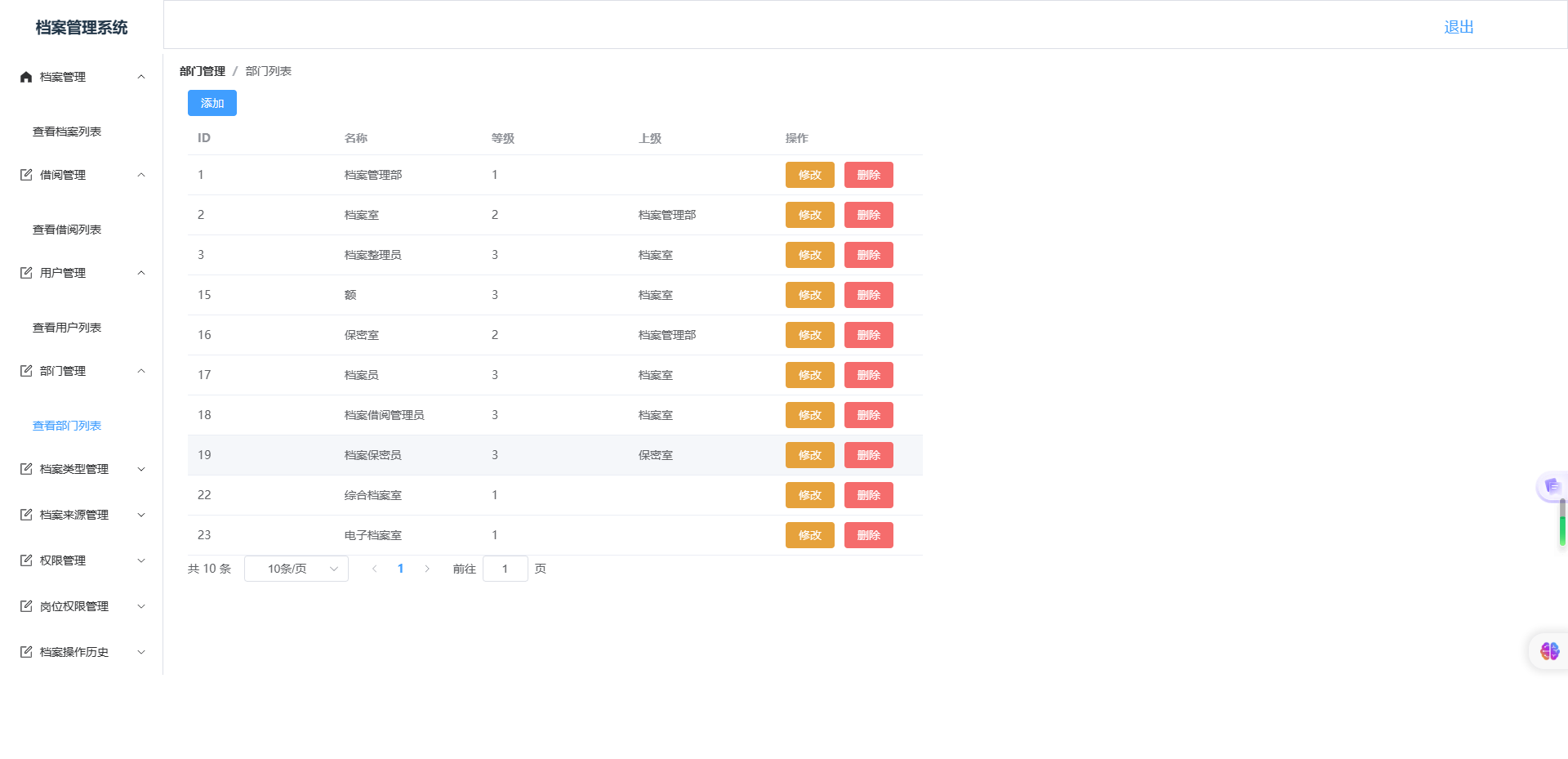
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selUserPage") public Page selUserPage(int page, int limit) {  Page page1 = this.userService.selUserPage(page, limit);  return page1; } |

* 1. 部门管理
     1. 页面文件



* + 1. 核心技术点

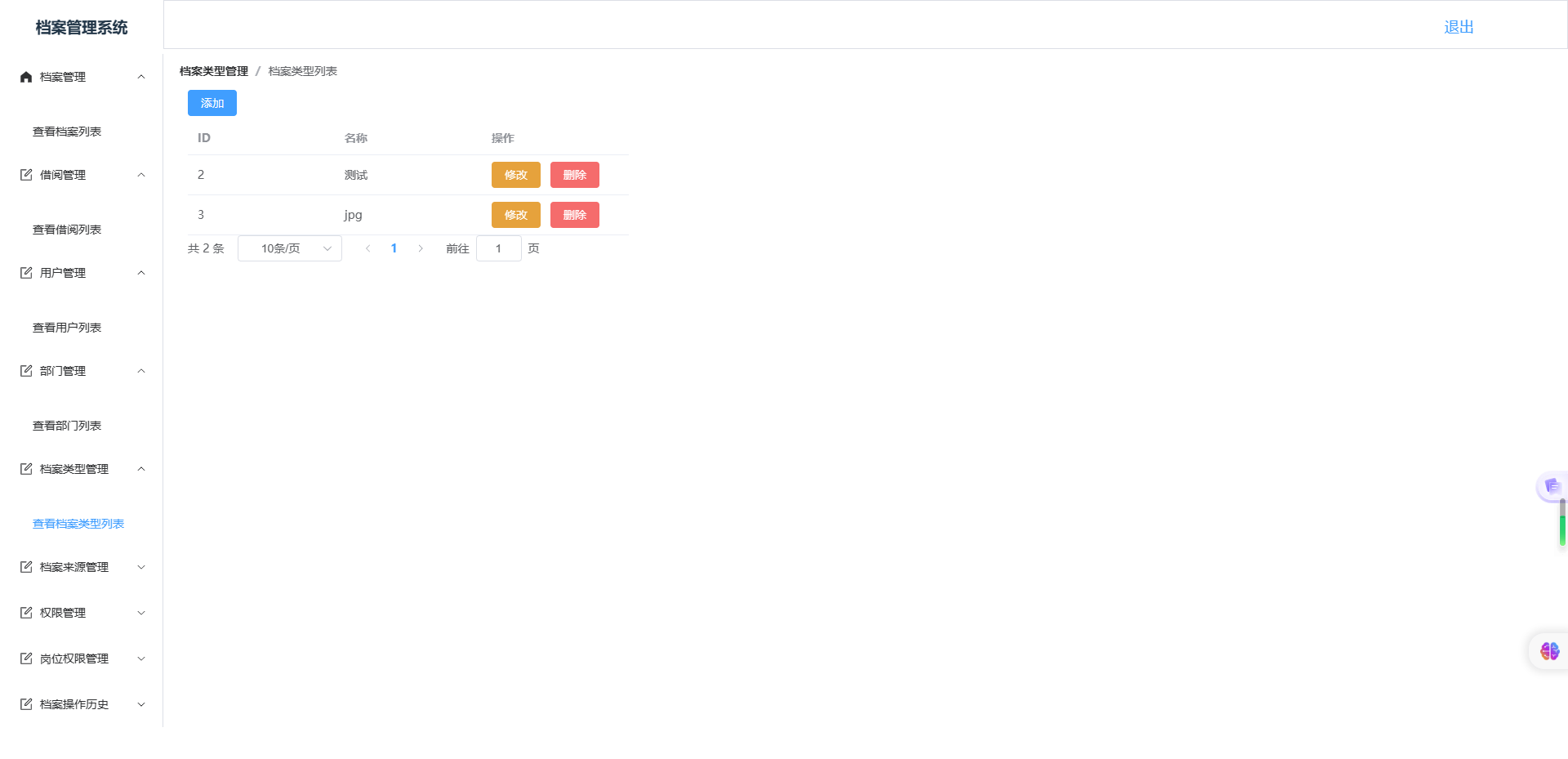
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selBumenPage") public Page selBumenPage(int page, int limit, int dengji, int shangji\_id) {  Page page1 = this.bumenService.selBumenPage(page, limit, dengji, shangji\_id);  return page1; } |

* 1. 档案类型管理
     1. 页面文件



* + 1. 核心技术点

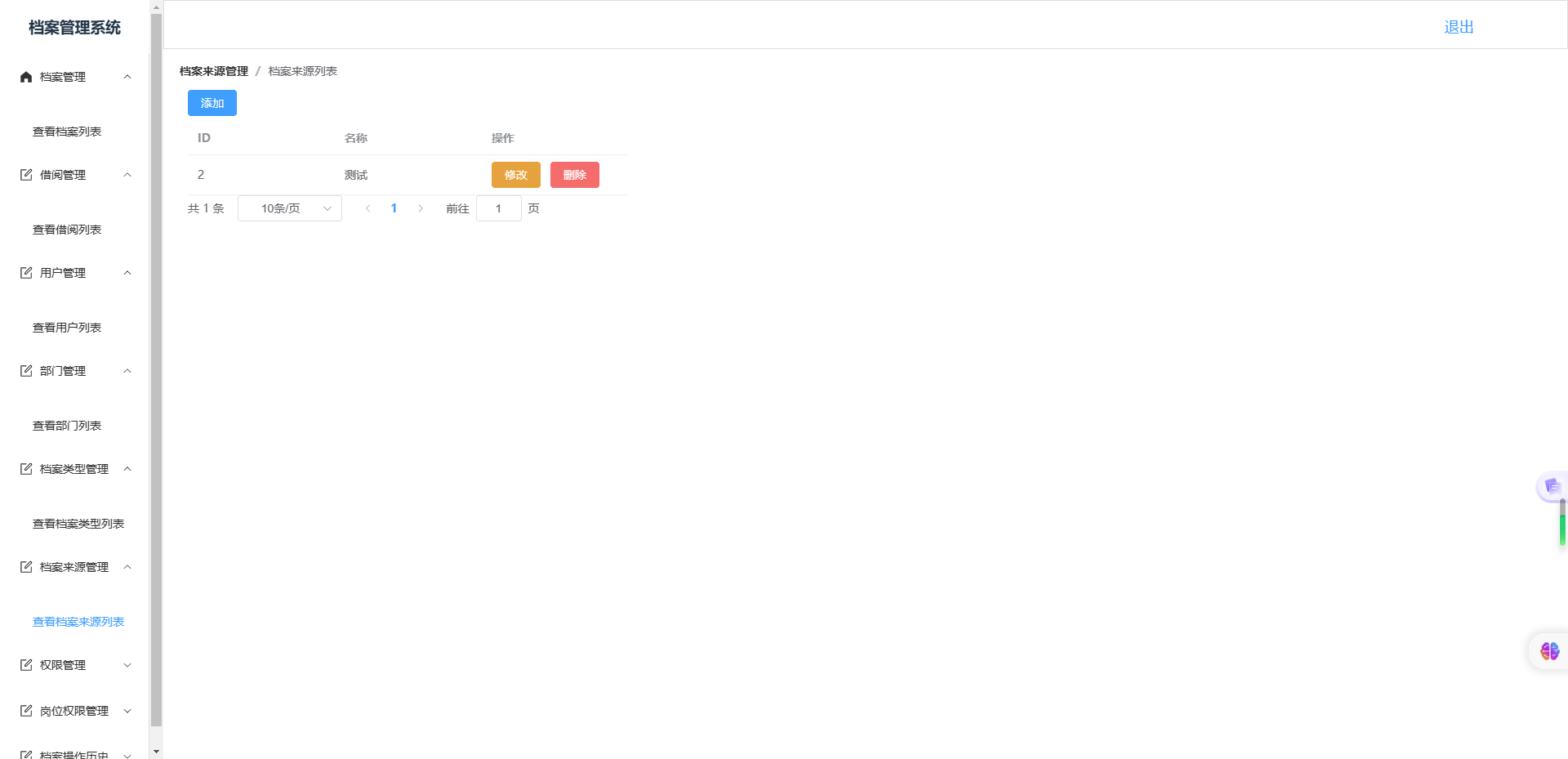
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selDalxPage") public Page selDalxPage(int page, int limit) {  Page page1 = this.dalxService.selDalxPage(page, limit);  return page1; } |

* 1. 档案来源管理
     1. 页面文件



* + 1. 核心技术点

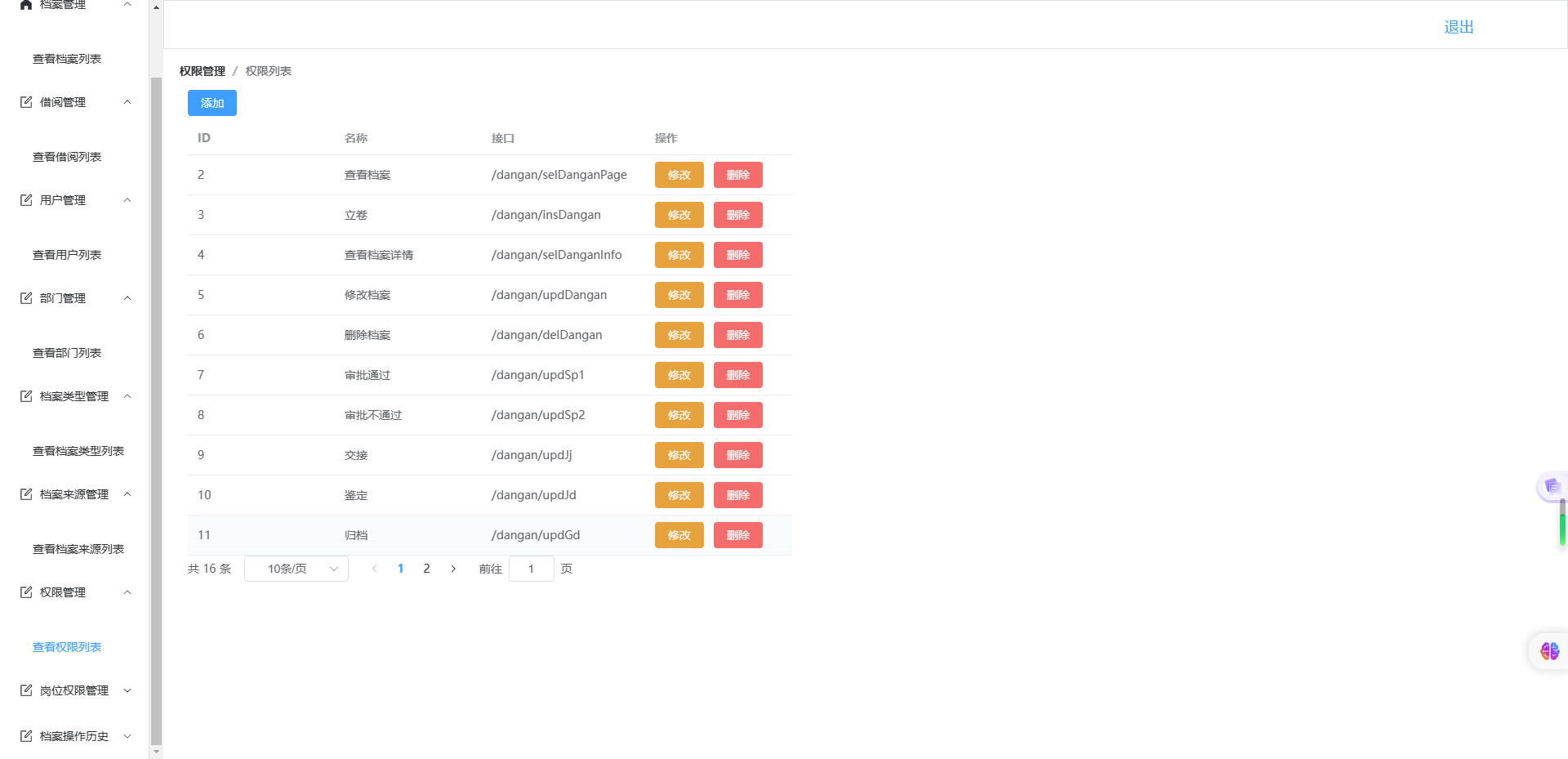
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selDalyPage") public Page selDalyPage(int page, int limit) {  Page page1 = this.dalyService.selDalyPage(page, limit);  return page1; } |

* 1. 权限管理
     1. 页面文件



* + 1. 核心技术点

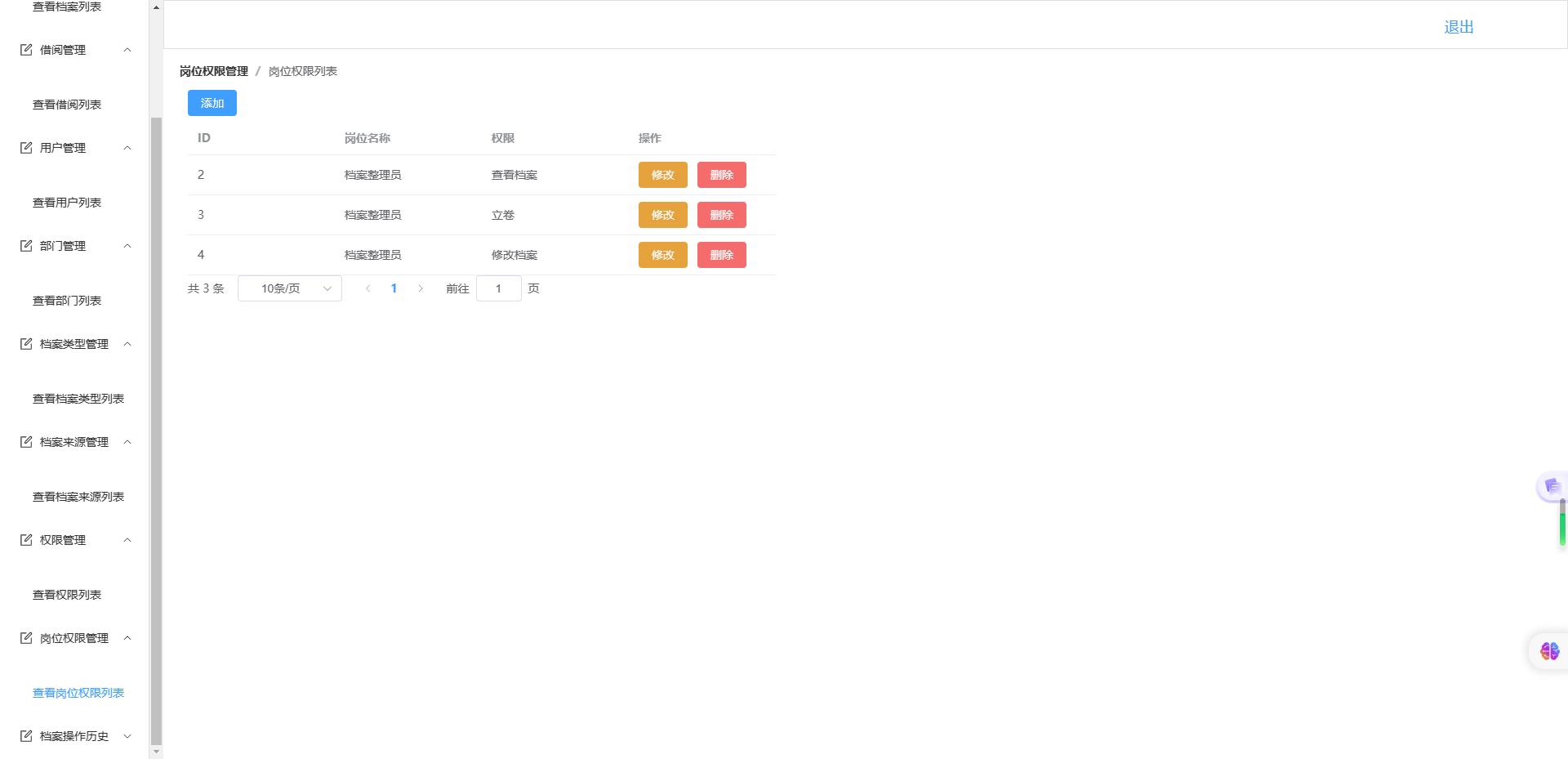
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selQuanxianPage") public Page selQuanxianPage(int page, int limit) {  Page page1 = this.quanxianService.selQuanxianPage(page, limit);  return page1; } |

* 1. 岗位权限管理
     1. 页面文件



* + 1. 核心技术点

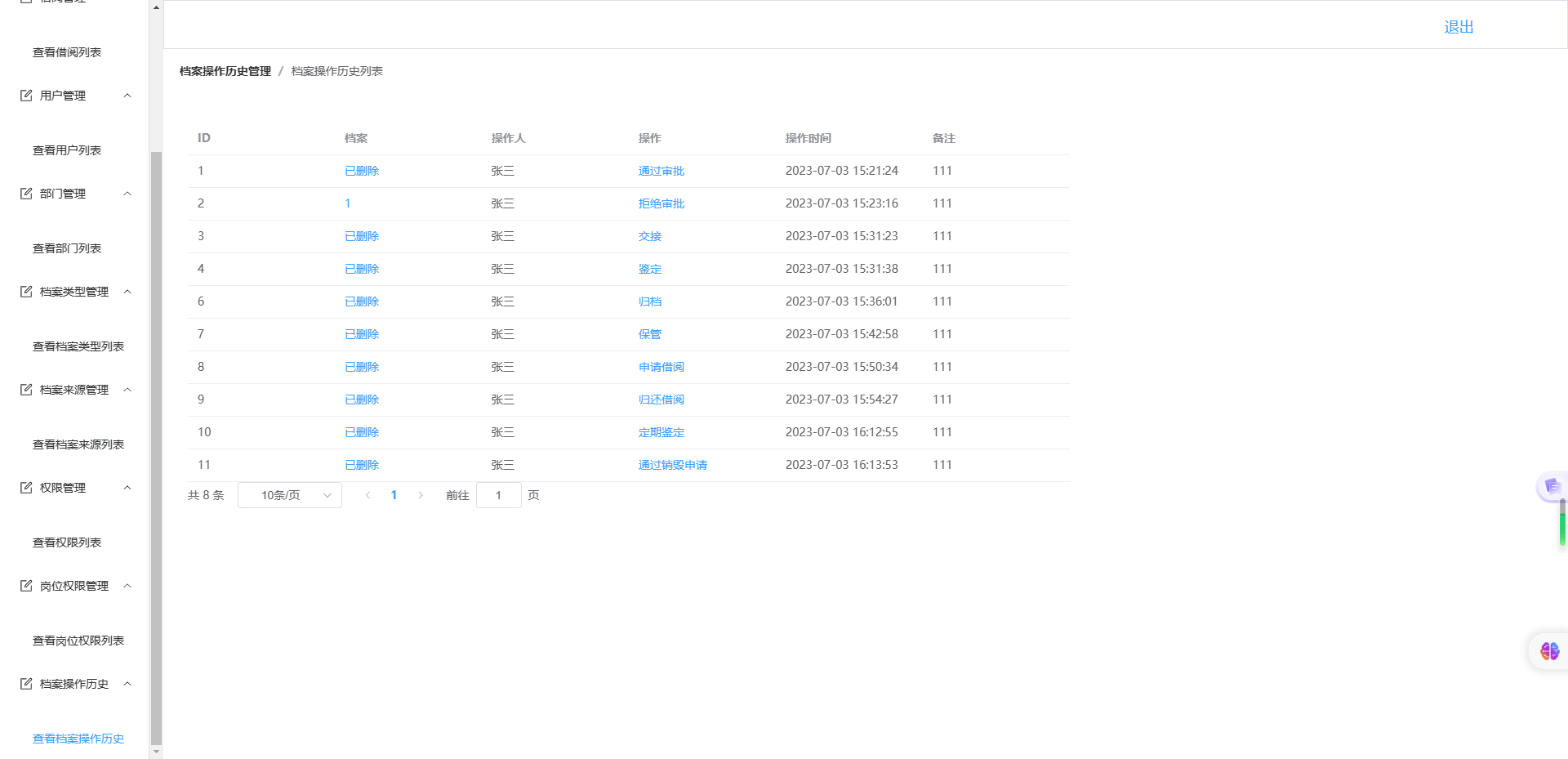
前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selBmqxPage") public Page selBmqxPage(int page, int limit) {  Page page1 = this.bmqxService.selBmqxPage(page, limit);  return page1; } |

* 1. 档案操作历史
     1. 页面文件



* + 1. 核心技术点

前台：Axios请求、vue-router路由。

后台：ssm接收并处理、返回数据。

* + 1. 核心代码

|  |
| --- |
| @GetMapping("/selDalsPage") public Page selDalsPage(int page, int limit) {  Page page1 = this.dalsService.selDalsPage(page, limit);  return page1; } |