湖南科技大学计算机科学与工程学院

计算机科学与技术生产实习

课程设计报告

专业班级： 21 级计算机科学与技术 7 班

学号姓名： 2102010629 陈琪琪 成绩 等级

学号姓名： 2101050325 邓子豪 成绩 等级

学号姓名： 2106040128 章丽红 成绩 等级

指导教师： 肖宇锋

时　　间： 2024.09.02-2024.09.15

地　　点： 逸夫楼 433

指导教师评语：

签名：

年　月　日

目录

第一章 系统需求分析

1.1 系统开发目的

目标设计开发一套用于管理高校中存在的重要资产设备（特种设备）管理系统，高校重要资产设备设备主要包括电梯、起重机、压力容器三类。该系统全称为“重要资产设备维保预警管理系统”，简称“特种设备管理系统”。根据高校重要资产设备管理需求分析，线上管理系统具有以下功能需求：

（1）用户管理功能，包括用户注册和登录的功能；

（2）新购特种设备注册管理功能；

（3）特种设备使用管理功能；

（4）特种设备维护保养管理功能；

（5）特种设备维修管理功能；

（6）特种设备定期检测检验管理功能；

（7）特种设备报废管理功能。

我校特种设备主要包括电梯、起重机、压力容器三类，针对其管理系统涉及到特种设备管理员、特种设备详细信息、维护保养信息、维护保养管理、检测检验信息、检测检验管理、特种设备操作人员、报废管理等。

管理员负责对我校中所有设备进行统一管理，包括新购设备注册管理、设备日常使用情况、维护保养情况、维修情况、检测检验情况、报废情况等。特种设备使用必须由取得特种设备操作人员证的人员才可进行。特种设备管理系统的业务工作流程如图 1 所示。特种设备管理系统业务工作流程图如图 1 所示。

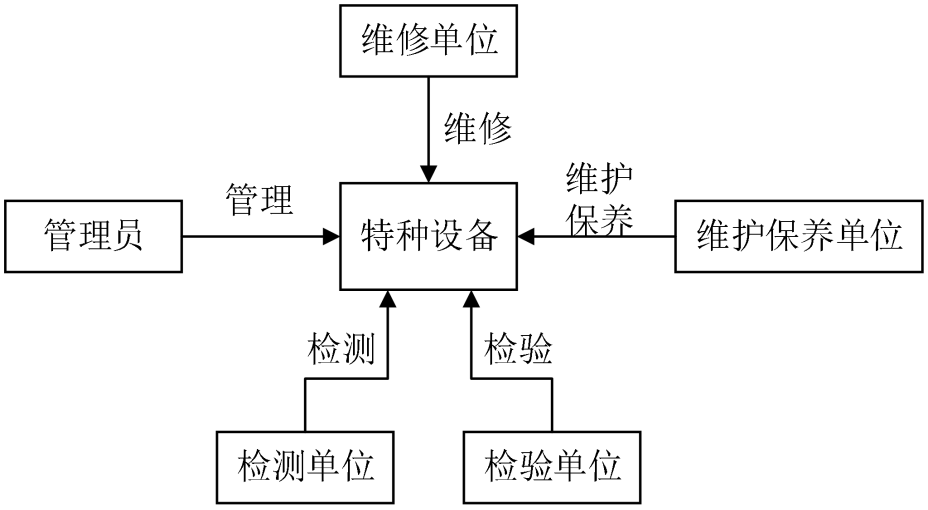

图 1 特种设备管理系统业务工作流程图

1.2 系统需求概述

现如今，各大高校内的特种设备呈现设备样式多元化、检修时间不统一、设备分布离散化的特点。因此，传统人工维护检修重点资产设备一直是安全运维的难点、痛点。为了解决重点资产设备规范化管理，及时通知检修人员进行安全检修。本项目旨在设计开发一套高校重要资产设备预警管理系统。在设备保修期限、定期检测、强制报废等时间节点前，通过多种途径，自动提醒管理人员和维修人员，有效提高相关资产的运营服务质量、减轻运维工作量。

1.3 系统架构图

本项目的设计基于 RuoYi 框架，RuoYi 是一个 Java EE 企业级快速开发平台，基于经典技术组合 Spring Boot、Apache Shiro、MyBatis、Thymeleaf、Bootstrap。框架支持主流的部署方式，可根据自身需求选择合适的部署方式，并提供了一些实用的开发工具和技术文档，降低了维护难度，能够有效为高校特种设备提供规范、便利的管理。

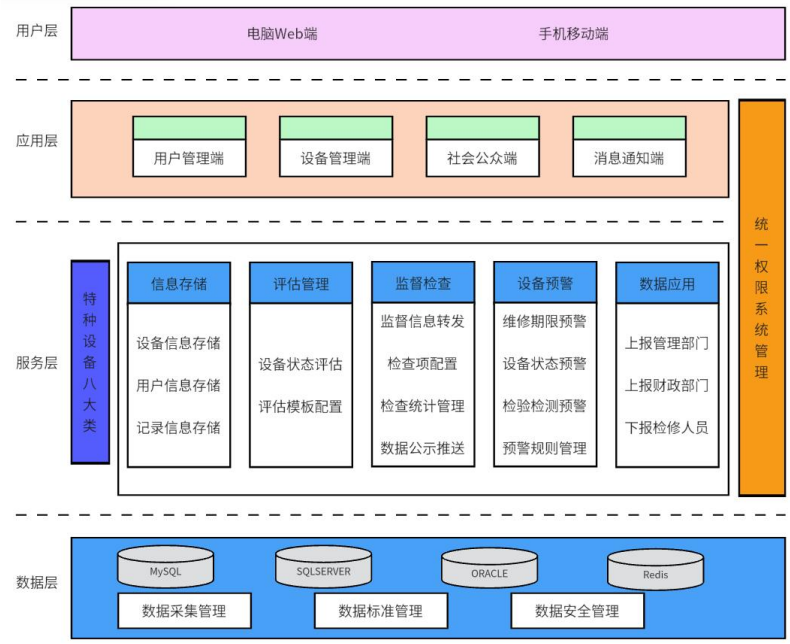

图 2 特种设备管理系统系统架构图

1.4 系统功能需求分析

该管理系统涵盖设备详细信息、维护保养管理、检测检验管理、设备操作人员管理及报废管理等功能。系统运行流程包括管理员录入新购设备信息、系统可视化呈现设备信息、以及短信通知维护人员进行检修。系统功能模块分为基础信息管理、特种设备信息管理和日志管理三大模块。系统具体功能如下图 3 所示。

图 3 特种设备管理系统功能模块图

1.5 系统模块用例图

1.5.1 角色关系用例图

如图 4 所示为角色关系用例图，项目系统由四种角色组成：用户、检修员、企业负责人和管理员。

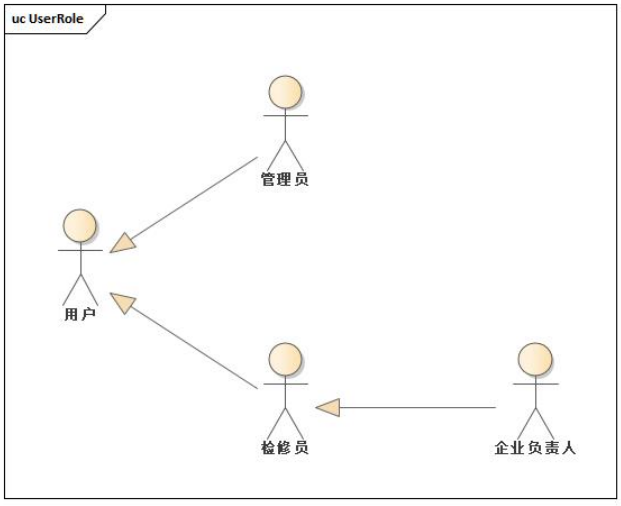

图 4 角色关系用例图

1.5.2 用户管理模块用例

（1）目的 管理特种设备管理系统用户账号信息的系统管理，用户登录修改信息等功能操作。管理员负责的用户权限等级的分配。

（2）前置条件 数据库中含有相应的权限等级和用户信息数据

（3）触发器 管理员需要添加、修改、删除或查找用户账号，需要分配权限。用户需要登录账号或修改账号信息。企业负责人需要为本企业相关的维修人员添加、删除或查看账号信息。

（4）备选流程 如果填写的账号信息不符合规范，系统会提示相关人员进行修正。

（5）后置条件 使用者成功完成账号信息的添加、修改、删除或查找操作后，系统更新并保存相应的单位信息，使用者可以继续进行其他操作。

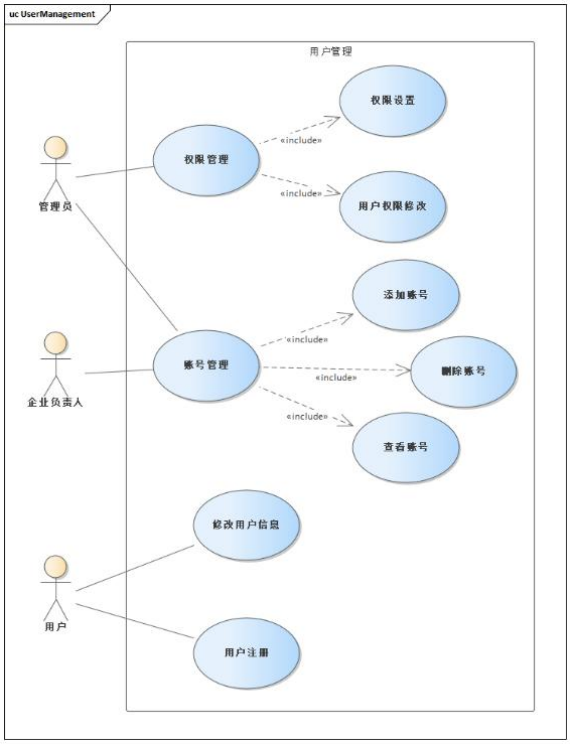

图 5 用户管理模块用例图

1.5.3 角色管理模块用例

（1）目的 管理特种设备管理系统角色管理子模块，管理员可以创建、修改、查看和删除系统存在的角色。

（2）前置条件 用户已经通过身份验证登录到系统中；用户拥有管理员权限。

（3）备选流程 如果填写的角色不符合规范，系统会提示管理员进行修正。

（4）后置条件 管理员成功完成角色的添加、修改、删除或查找操作后，系统更新并保存相应的角色信息，管理员可以继续进行其他操作。

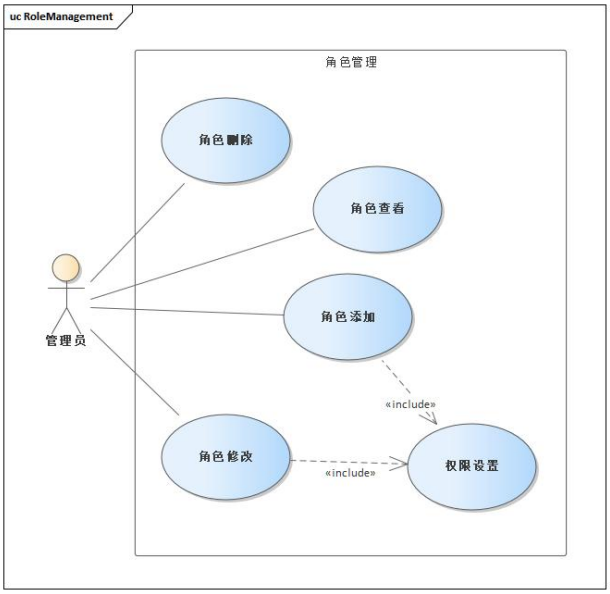

图 6 角色管理模块用例图

1.5.4 单位管理模块用例图

（1）目的 管理特种设备管理系统中的单位信息，包括校内单位和校外单位。

（2）参与者 管理员（负责管理单位信息，包括增加、删除、修改和查找单位信息）

（3）前置条件 用户已经通过身份验证登录到系统中；用户拥有管理员权限。

（4）触发器 管理员需要添加、修改、删除或查找单位信息。

（5）备选流程 如果填写的单位信息不符合规范，系统会提示管理员进行修正。

（6）后置条件 管理员成功完成单位信息的添加、修改、删除或查找操作后，系统更新并保存相应的单位信息，管理员可以继续进行其他操作。

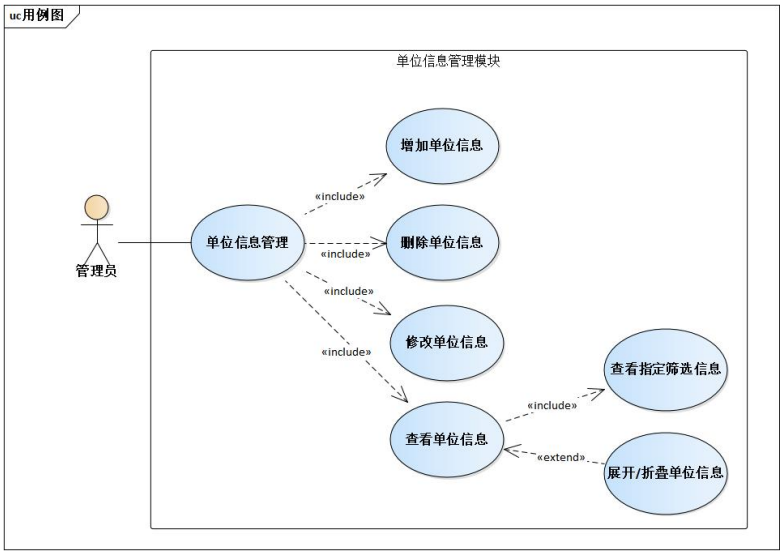

图 7 单位管理模块用例图

1.5.5 岗位管理模块用例图

（1）目的 管理特种设备管理系统中的岗位信息，包括开发人员、设备总负责人、单位负责人和维保人员。

（2）参与者 管理员（负责管理岗位信息，包括增加、删除、修改和查找岗位信息）

（3）前置条件 用户已经通过身份验证登录到系统中且用户拥有管理员权限。

（4）触发器 管理员需要添加、修改、删除或查找岗位信息。

（5）备选流程 如果填写的岗位信息不符合规范，系统会提示管理员进行修正。

（6）后置条件 管理员成功完成岗位信息的添加、修改、删除或查找操作后，系统更新并保存相应的岗位信息，管理员可以继续进行其他操作。

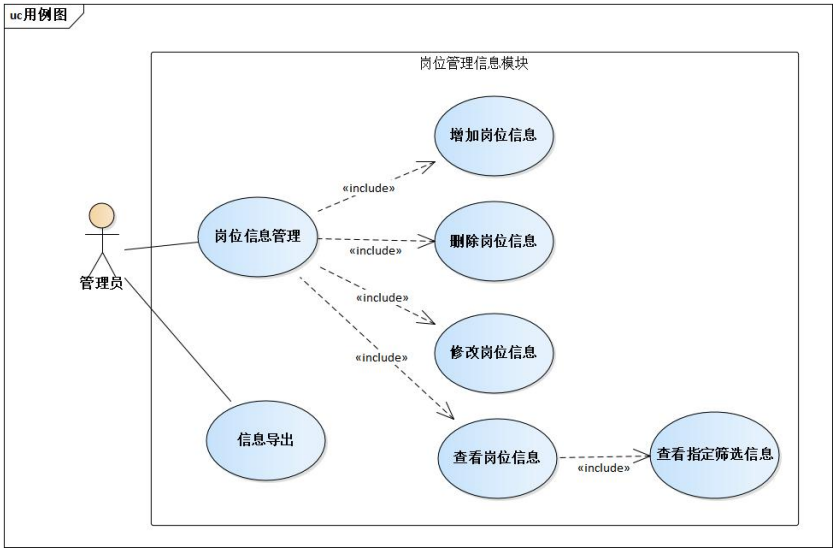

图 8 岗位管理模块用例图

1.5.6 设备管理模块用例图

（1）目的 管理特种设备管理系统中的电梯、起重机和压力容器等设备信息。

（2）参与者 管理员负责管理所有设备信息，包括增加、删除、修改和查找设备息，并控制界面字段的显示和隐藏；单位负责人负责查看本单位的设备信息，并控制界面字段的显示和隐藏，导出本单位设备信息。设备维保人员负责查看自己管理的设备信息，并控制界面字段的显示和隐藏，同时在维保日期收到短信通知去维保设备，导出自己负责维保的设备信息。

（3）前置条件 用户已经通过身份验证登录到系统中。

（4）触发器 用户需要查看、修改、删除、导出设备信息，或控制界面字段的显示和隐藏。

（5）备选流程 如果填写的设备信息不符合规范，系统会提示用户进行修正；如果用户权限不足，系统会提示用户无权进行相应操作。

（6）后置条件 用户成功完成设备信息的增加、删除、修改、查找、导出或界面字段的显示和隐藏操作后，系统更新并保存相应的设备信息或设置，用户可以继续进行其他操作

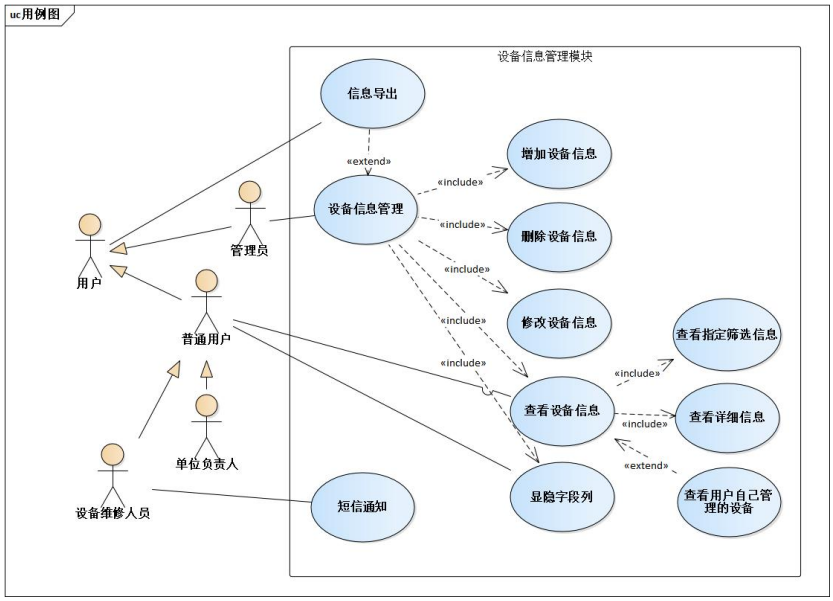

图 9 设备管理模块用例图

1.5.7 年检类型管理模块用例图

（1）目的 管理特种设备管理系统中的年检类型信息，包括年检周期中对应的年份、年检类型和备注等信息。

（2）参与者 管理员（负责管理年检类型信息，包括增加、删除、修改和查找检验类型信息。

（3）前置条件 用户已经通过身份验证登录到系统中且用户拥有管理员权限。

（4）触发器 管理员需要添加、修改、删除或查找检验类型信息。

（5）备选流程 如果填写的检验类型信息不符合规范，系统会提示管理员进行修正。

（6）后置条件 管理员成功完成检验类型信息的添加、修改、删除或查找操作后，系统更新并保存相应的检验类型信息，管理员可以继续进行其他操作。

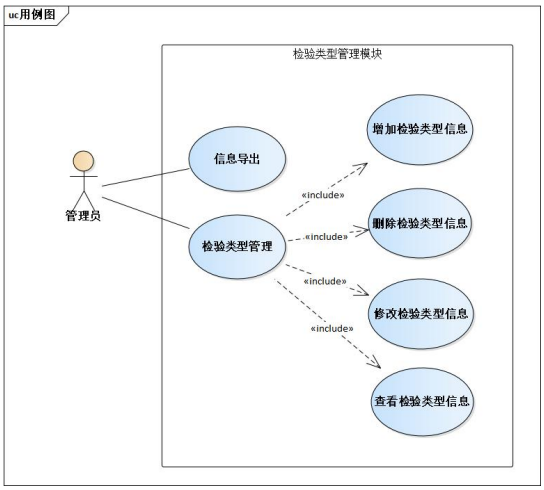

图 10 年检类型管理模块用例图

1.5.8 日志管理模块用例图

（1）目的 管理特种设备管理系统中的用户所填写的检修日志信息

（2）参与者 管理员和检修员（负责管理年检类型信息，包括增加、删除、修改和查找检验类型信息。）

（3）前置条件 用户已经通过身份验证登录到系统中且用户拥有管理员或检修员权限。

（4）触发器 管理员或检修员需要添加、修改、删除或查找检验检修日志。

（5）备选流程 如果填写的日志信息不符合规范，系统会提示管理员进行修正。

（6）后置条件 管理员成功完成检验类型信息的添加、修改、删除或查找操作后，系统更新并保存相应的检验类型信息，管理员可以继续进行其他操作。

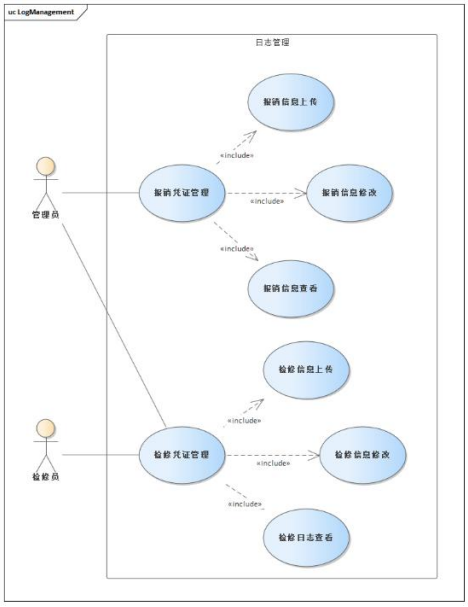

图 11 日志管理模块用例图

第二章 系统概要设计

2.1 技术体系

重要资产设备维保预警管理系统其整个开发技术栈可以划分为四部分，分别是前端、后端、数据库以及安全开发框架。其中前端所采用的技术有：Vue + Thymeleaf + Bootstrap；后端所采用的技术有 Spring + SpringBoot + MyBatis；数据库采用的相关技术有 MySQL 和 Redis；安全开发相关的框架有 ApacheShiro。

2.1.1 总体结构设计

I3CRM 系统采用 B/S 模式开发，支持 PC 机和手机的客户端调用。采用前后端完全分离的三层结构。

2.1.2 前端表示层

（1）Vue.js 是一个用于构建用户界面的渐进式 JavaScript 框架。它设计简单易用，核心库专注于视图层，并且可以与其他库或现有项目进行集成。Vue.js 的主要特性包括响应式数据绑定和组件化开发，使得开发者可以更轻松地构建动态和互动丰富的前端应用。通过使用 Vue 的生态系统工具，如 Vue Router 和 Vuex，开发者可以构建复杂的单页面应用（SPA）。

（2）Thymeleaf 是一个现代的 Java 模板引擎，用于在服务器端生成 HTML、XML、JavaScript 和文本内容。它设计用于与 Spring 框架无缝集成，支持 HTML5 和自然模板，这意味着模板可以在浏览器中作为静态内容进行查看和调试。Thymeleaf 提供了强大的表达式语言和逻辑控制功能，使得动态内容的生成变得简单高效。

（3）Bootstrap 是一个开源的前端框架，用于开发响应式和移动优先的网站。它提供了一组预定义的 CSS 和 JavaScript 组件，包括排版、表单、按钮、导航和网格系统，帮助开发者快速构建美观且一致的用户界面。Bootstrap 的设计理念是简化开发过程，提高生产效率，同时确保应用在不同设备和浏览器上具有良好的兼容性。

2.1.3 业务逻辑处理层

（1）Spring 是一个开源的企业级 Java 应用程序开发框架。它提供了全面的基础设施支持，包括依赖注入（DI）、面向切面编程（AOP）、事务管理、持久化、Web 应用程序开发等。Spring 的核心理念是通过简化配置和促进代码的松耦合，提高开发效率和应用的可维护性。Spring 生态系统还包括 Spring Boot（简化了 Spring 应用的配置和部署）和 Spring Cloud（用于构建分布式系统）。

（2）IoC 是 Spring 框架的核心概念之一，通过 IoC 容器实现了对象之间的解耦和

依赖关系的管理。在传统的 Java 开发中，对象之间的创建和依赖关系通常由程序员手动管理，而在 Spring 中，这些工作交由 IoC 容器负责，使得应用程序更加灵活、可维护和可测试。

（3）AOP Spring 框架的另一个重要特性，通过 AOP 可以将应用程序的横切关注点（如日志记录、事务管理等）与业务逻辑分离开来，使得代码更加模块化、清晰和可重用。Spring 提供了丰富的 AOP 支持，包括切点、通知、切面等，可以轻松实现复杂的横切逻辑。

（4）SpringBoot 是基于 Spring 框架的开源项目，旨在简化 Spring 应用程序的创建和配置。它通过提供默认配置和自动化配置机制，使得开发者能够快速构建独立运行的、生产级别的 Spring 应用程序，而无需手动编写大量的配置代码。Spring Boot 提供了内嵌的服务器（如 Tomcat 和 Jetty），无需外部服务器即可运行应用。它还包含丰富的生态系统支持和开箱即用的功能，使得开发微服务架构和 RESTful 应用程序变得更加高效。

（5）MyBatis 是一个流行的 Java 持久层框架，用于简化数据库访问和操作。它通过 XML 或注解的方式，将 SQL 查询、存储过程和 Java 对象进行映射，从而简化了数据库交互的代码编写。与全自动的 ORM 框架不同，MyBatis 允许开发者对 SQL 进行细粒度的控制，更灵活地执行复杂查询。MyBatis 支持动态 SQL、缓存以及事务管理，非常适合需要灵活操作 SQL 的项目。

2.1.4 数据持久层

（1）MySQL 是一个广泛使用的开源关系型数据库管理系统（RDBMS），以其性能高、可靠性强和使用简单著称。它使用结构化查询语言（SQL）进行数据管理和操作，适合各种应用场景，从小型项目到大型企业应用。MySQL 支持多种存储引擎、事务处理、主从复制以及强大的安全机制，广泛应用于互联网应用、数据仓库和嵌入式系统中。

（2）Redis 是一个开源的、基于内存的数据结构存储系统，通常用作数据库、缓存和消息代理。它支持多种数据结构，如字符串、列表、集合、有序集合、哈希和位图等。由于其所有数据都存储在内存中，Redis 具有极高的读写性能，非常适合需要快速响应的应用场景。同时，Redis 支持数据持久化，可以将内存中的数据定期保存到磁盘上。Redis 还具备丰富的功能，如发布/订阅、Lua 脚本、事务和自动故障转移。

2.1.5 安全开发层

（1）ApacheShiro 是一个强大且灵活的 Java 安全框架，旨在简化身份验证、授权、会话管理和加密等安全功能的实现。Shiro 提供了一种易于使用的 API，让开发者可以轻松地为 Java 应用程序添加安全控制，包括用户登录/登出、访问权限控制和数据保护。

它支持多种认证方式和授权机制，适用于从小型应用到大型企业级应用的各种场景。

2.2 开发环境

2.2.1 集成开发环境（IDE）

团队采用 IntelliJ IDEA、Eclipse 和 Visual Studio Code 开发工具。这些 IDE 提供了代码编辑、调试、版本控制等功能，能够极大地提高开发效率。

2.2.2 编程语言

根据所选择的技术栈，前端开发使用 HTML、CSS、JavaScript 以及 Vue.js 等；后端开发可能会使用 Java 语言，并结合 Spring 框架进行开发；安全开发涉及到 Java 语言和 ApacheShiro 框架。

2.2.3 版本控制管理

使用版本控制管理如 Git 进行代码管理是开发中的常见做法。开发团队使用 GitHub、GitLab 或 Bitbucket 等平台来托管代码，并进行版本控制和团队协作。

2.2.4 数据库

根据技术栈中提到的数据库技术，开发者使用 MySQL 和 Redis 作为数据库。使用 MySQL Workbench 或者 Navicat 等数据库管理工具可用于数据库的设计、管理和查询。

2.2.5 Web 服务器

开发过程中，需要使用 Tomcat、Jetty 或者 Spring Boot 内置的服务器来运行和调试 Web 应用程序。

2.2.6 API 开发工具

系统需要提供 RESTful API 接口，使用 Swagger 或 Postman 等工具来设计、测试和文档化 API 接口。

2.2.7 前端工具和框架

前端开发使用 Node.js、npm 或者 Yarn 来管理项目依赖和构建工具。此外，前端框架如 Vue.js 和 Bootstrap 用于快速构建用户界面。npm 是 Node.js 的包管理器，用于安装、管理和发布 JavaScript 包。开发者可以使用 npm 来安装各种前端工具、框架和库，以及管理项目的依赖关系。

2.2.8 测试工具

开发过程中可能会使用单元测试和集成测试来确保代码质量和功能的正确性。常见的 Java 测试框架包括 JUnit 和 Mockito 等。

2.3 接口设计

2.3.1 基础信息管理模块

（1）登录和注册

表 1 登录和注册接口设计表

接口名	接口描述
Public AjaxResult login(@RequestBody LoginBody loginBody)	获得用户输入的账户密码验证码信息，生成登录令牌
public AjaxResult register(@RequestBody RegisterBody user)	根据用户输入的注册信息判断是否可以合法注册

（2）用户管理模块

表 2 用户管理接口设计表

接口名	接口描述
public TableDataInfo list (SysUser user)	获取用户列表
public void export (HttpServletResponse response, SysUser user)	以 excel 文件格式导出用户数据
public AjaxResult getInfo (@PathVariable(value = "userId", required = false) Long userId)	根据用户编号获取详细信息
public AjaxResult add (@Validated @RequestBody SysUser user)	新增用户
public AjaxResult edit (@Validated @RequestBody SysUser user)	修改用户
public AjaxResult remove (@PathVariable Long[] userIds)	删除用户
public AjaxResult resetPwd (@RequestBody SysUser user)	重置密码
public AjaxResult changeStatus (@RequestBody SysUser user)	用户状态修改

public AjaxResult authRole (@PathVariable("userId") Long userId)	根据用户编号获取授权角色
public AjaxResult insertAuthRole (Long userId, Long[] roleIds)	用户授权角色
public AjaxResult deptTree (SysDept dept)	获取部门树列表

（3）角色管理模块

表 3 角色管理接口设计表

接口名	接口描述
public TableDataInfo list(SysRole role)	获取角色列表
public void export (HttpServletResponse response, SysRole role)	以 excel 文件格式导出角色数据
public AjaxResult getInfo (@PathVariable Long roleId)	根据角色编号获取详细信息
public AjaxResult add (@Validated @RequestBody SysRole role)	新增角色
public AjaxResult edit (@Validated @RequestBody SysRole role)	修改保存角色
public AjaxResult dataScope (@RequestBody SysRole role)	修改保存数据权限
public AjaxResult changeStatus (@RequestBody SysRole role)	状态修改
public AjaxResult remove (@PathVariable Long[] roleIds)	删除角色
public TableDataInfo allocatedList (SysUser user)	查询已分配用户角色列表
public TableDataInfo unallocatedList (SysUser user)	查询未分配用户角色列表
public AjaxResult cancelAuthUserAll (Long roleId, Long[] userIds)	批量取消授权用户
public AjaxResult selectAuthUserAll (Long roleId, Long[] userIds)	批量选择用户授权
public AjaxResult deptTree (@PathVariable("roleId") Long roleId)	获取对应角色部门树列表

（4）单位管理模块

表 4 单位管理接口设计表

接口名	接口描述
public AjaxResult list(SysDept dept)	获取单位列表
public AjaxResult excludeChild (@PathVariable(value = "deptId", required = false) Long deptId)	查询单位列表（排除节点）
public AjaxResult getInfo (@PathVariable Long deptId)	根据单位编号获取详细信息
public AjaxResult add (@Validated @RequestBody SysDept dept)	新增单位
public AjaxResult edit (@Validated @RequestBody SysDept dept)	修改单位
public AjaxResult remove (@PathVariable Long deptId)	删除单位

（5）岗位管理模块

表 5 岗位管理接口设计表

接口名	接口描述
public TableDataInfo list (SysPost post)	获取岗位列表
public void export (HttpServletResponse response, SysPost post)	以 excel 文件格式导出岗位数据
public AjaxResult getInfo (@PathVariable Long postId)	根据岗位编号获取详细信息
public AjaxResult add (@Validated @RequestBody SysPost post)	新增岗位
public AjaxResult edit (@Validated @RequestBody SysPost post)	修改岗位
public AjaxResult remove (@PathVariable Long[] postIds)	删除岗位
public AjaxResult optionselect()	获取岗位选择框列表

2.3.2 设备信息管理模块

（1）设备管理模块

表 6 设备管理接口设计表

接口名	接口描述
public TableDataInfo list (EquipmentLift equipmentLift)	查询电梯管理列表
public void export (HttpServletResponse response, EquipmentLift equipmentLift)	导出电梯管理列表
public AjaxResult getInfo (@PathVariable("liftId") Long liftId)	获取电梯管理详细信息
public AjaxResult add (@RequestBody EquipmentLift equipmentLift)	新增电梯管理
public AjaxResult edit (@RequestBody EquipmentLift equipmentLift)	修改电梯管理
public AjaxResult remove (@PathVariable Long[] liftIds)	删除电梯管理
public TableDataInfo list (EquipmentCrane equipmentCrane)	查询起重机管理列表
public void export (HttpServletResponse response, EquipmentCrane equipmentCrane)	导出起重机管理列表
public AjaxResult getInfo (@PathVariable("craneId") Long craneId)	获取起重机管理详细信息
public AjaxResult add (@RequestBody EquipmentCrane equipmentCrane)	新增起重机管理
public AjaxResult edit (@RequestBody EquipmentCrane equipmentCrane)	修改起重机管理
public AjaxResult remove (@PathVariable Long[] craneIds)	删除起重机管理
public TableDataInfo list (EquipmentVessel equipmentVessel)	查询压力容器管理列表
public void export (HttpServletResponse response, EquipmentVessel equipmentVessel)	导出压力容器管理列表
public AjaxResult getInfo (@PathVariable("vesselId") Long vesselId)	获取压力容器管理详细信息
public AjaxResult add (@RequestBody	新增压力容器管理

EquipmentVessel equipmentVessel)	
public AjaxResult edit (@RequestBody EquipmentVessel equipmentVessel)	修改压力容器管理
public AjaxResult remove (@PathVariable Long[] vesselIds)	删除压力容器管理

（2）年检类型管理模块

表 7 年检类型接口设计表

接口名	接口描述
public TableDataInfo list (EquipmentCheck equipmentCheck)	查询检验类型管理列表
public void export (HttpServletResponse response, EquipmentCheck equipmentCheck)	导出检验类型管理列表
public AjaxResult getInfo (@PathVariable("checkId") Integer checkId)	获取检验类型管理详细信息
public AjaxResult add (@RequestBody EquipmentCheck equipmentCheck)	新增检验类型管理
public AjaxResult edit (@RequestBody EquipmentCheck equipmentCheck)	修改检验类型管理
public AjaxResult remove (@PathVariable Integer[] checkIds)	删除检验类型管理

2.3.3 日志信息管理模块

（1）日志信息管理模块

表 8 日志信息管理接口设计表

接口名	接口描述
public Image selectImage ByLogId (Long logId);	查询检修日志登记
public List<Image> selectImageList (Image image);	查询检修日志登记列表
public int insertImage (Image image);	新增检修日志登记
public int updateImage (Image image);	修改检修日志登记
public int deleteImageByLogId (Long logId);	删除检修日志登记

public int deleteImageByLogIds (Long[] logIds);	批量删除检修日志登记
public TableDataInfo list (Image image)	查询检修日志登记列表
public void export (HttpServletResponse response, Image image)	导出检修日志登记列表
public AjaxResult getInfo (@PathVariable("logId") Long logId)	获取检修日志登记详细信息
public AjaxResult add (@RequestBody Image image)	新增检修日志登记
public AjaxResult edit (@RequestBody Image image)	修改检修日志登记
public AjaxResult remove (@PathVariable Long[] logIds)	删除检修日志登记
public Image selectImageByLogId (Long logId);	查询检修日志登记
public List<Image> selectImageList (Image image);	查询检修日志登记列表
public int insertImage (Image image);	新增检修日志登记
public int updateImage (Image image);	修改检修日志登记
public int deleteImageByLogIds (Long[] logIds);	批量删除检修日志登记
public int deleteImageByLogId (Long logId);	删除检修日志登记信息

2.4 模块介绍

2.4.1 系统总体数据流

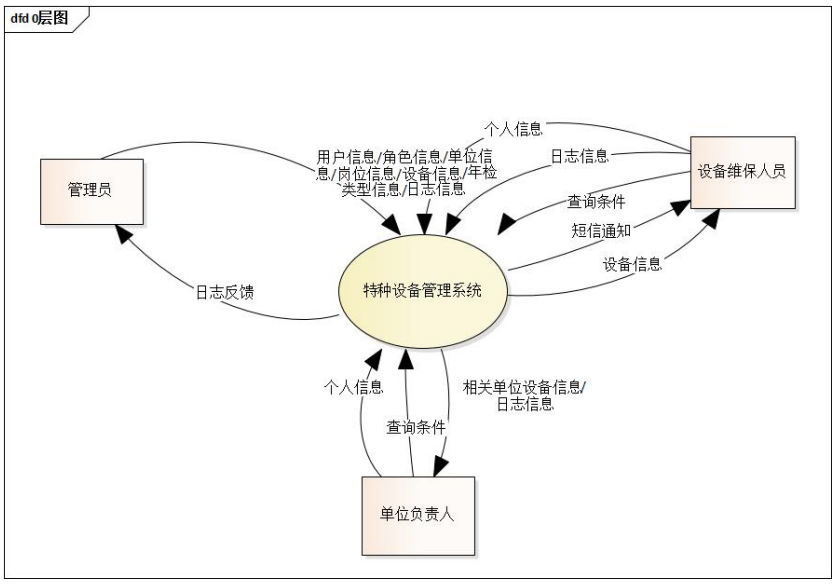

图 12 系统总体数据流图

图 12 展示了特种设备管理系统中的各个模块之间的数据流动情况。以下是对图中系统的介绍：

（1）特种设备管理系统（中心处理器） 系统的核心部分，负责处理和管理与特种设备相关的信息。该部分与多个外部实体进行数据交互，包括管理员、设备保管人员、单位负责人等。

（2）外部实体 管理员负责维护系统中的用户信息、角色信息和设备信息，同时接收系统生成的日志反馈信息；设备保管人员负责接收和处理设备的相关通知，如设备信息、报警事件和短信通知；单位负责人负责管理本单位的设备和人员信息，并与系统互动，查看报警事件、设备信息和日志信息。

（3）主要数据流动 用户信息/角色信息/单位信息/设备信息等数据由管理员输入到系统中，用于系统的基础数据维护；当发生报警事件的时候，系统会根据设备状态生成报警事件，并通知相关的设备保管人员和单位负责人；系统可以根据设备报警情况发送短信通知给设备保管人员；各外部实体可以向系统发送查询请求，以获取相关的设备信息和日志信息；系统会向管理员和单位负责人提供操作日志和系统运行情况的反馈。

（4）数据存储 系统会记录和存储设备相关的操作日志和报警事件，供管理员和单位负责人查询和分析。这个数据流图展示了一个集中的设备管理系统如何与不同角色的用户交互，如何处理与设备相关的关键数据，以及如何通过日志和报警机制来确保设备的安全运行。

2.4.2 基础信息管理模块数据流

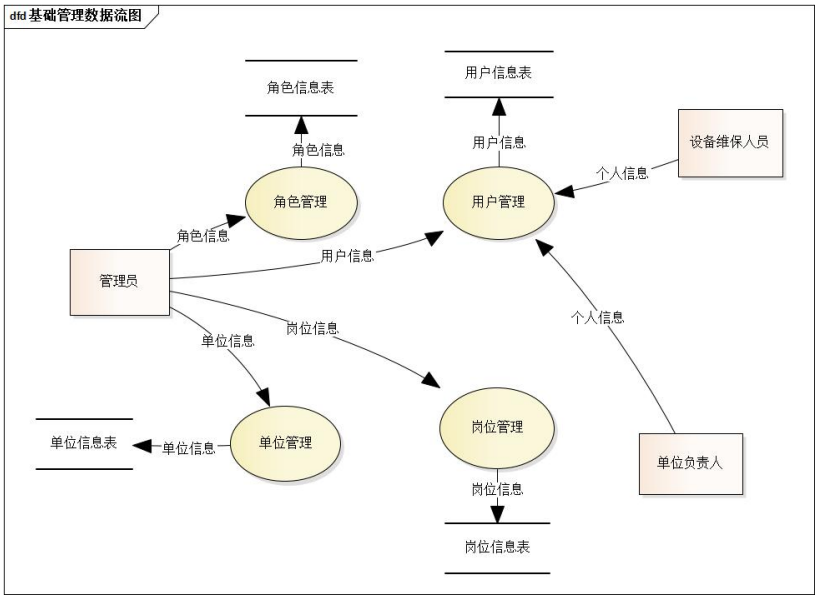

图 13 信息管理模块数据流图

信息管理模块可以分为角色管理、用户管理、单位管理以及外部实体四部分。

（1）角色管理 管理员可以维护和管理系统中的角色信息。角色信息包括各个用户在系统中的权限和职责，管理员更新角色信息后，会影响用户的权限。角色管理模块会把角色信息传递给用户管理和单位管理模块，以确保这些模块根据用户的角色来限制和管理权限。

（2）用户管理 这个模块负责维护系统中的用户信息，如用户的基本信息、角色、所属单位等。管理员能够添加、修改或删除用户信息，并将这些信息传递给相关模块，如设备保管人员和单位负责人。用户信息也与角色管理和单位管理模块相互关联，确保用户的角色和单位信息是同步的。

（3）单位管理 责管理系统中的单位信息，如单位的基本信息、关联的人员和设备等。单位管理模块会与用户管理和角色管理模块互动，以确保单位内的人员信息和其对应的角色权限是准确的。单位负责人能够查看和管理本单位的相关信息，并通过此模块获取设备和人员的最新信息。

（4）外部实体 设备保管人员接收和查看与其相关的用户信息和个人信息；单位负责人通过单位管理模块来管理和查看本单位的用户信息和设备信息。

（5）数据流动 用户信息从用户管理模块流向相关的外部实体（设备保管人员、单位负责人）；角色管理模块生成角色信息表，并将其传递到相关模块进行用户权限控制；单位管理模块生成单位信息表，传递给管理员和单位负责人。

这个信息管理模块的数据流图描述了如何通过角色管理、用户管理和单位管理来综合管理系统中的用户、角色和单位信息，以确保各个实体能够根据其权限访问和操作系统资源。

2.4.3 设备信息管理模块数据流

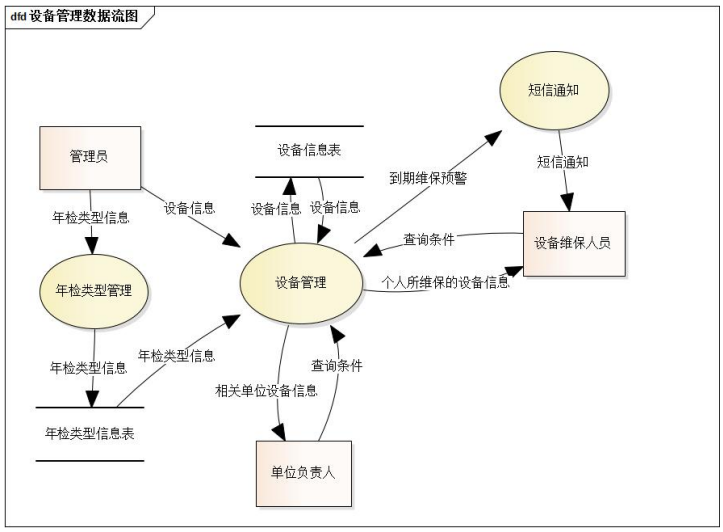

图 14 设备信息管理模块数据流图

这个图展示了设备信息管理模块的数据流情况。以下是对该模块的详细介绍：

（1）设备管理模块（核心） 设备管理模块是系统的核心部分，负责管理所有与设备相关的信息。它接收来自其他模块的输入，如设备信息表、年检类型信息，并与外部实体进行数据交互。

（2）外部实体 管理员负责录入和维护设备信息，并将这些信息提交到设备管理模块；设备保管人员通过设备管理模块获取其管理的设备信息，并在设备即将到期时收到短信通知；单位负责人通过设备管理模块查询和管理本单位的设备信息。

（3）辅助模块 年检类型管理模块负责管理设备的年检信息，并将年检类型信息传递给设备管理模块。年检类型信息包括设备需要定期检验的项目和频率。

（4）数据流动 设备信息表由管理员输入并传递给设备管理模块，设备管理模块基于这些信息进行设备管理和维护；年检类型信息:从年检类型管理模块传递到设备管理模块，用于指导设备的定期检验；设备管理模块根据设备到期信息生成短信通知，发送给相关的设备保管人员，以提醒设备即将到期需要检验；设备保管人员和单位负责人可以通过查询事件获取设备的最新信息；当设备接近年检或有其他问题时，设备管理模块会生成到期提示或报警信息。

这个设备信息管理模块确保了设备信息的及时更新和管理，并通过短信通知等机制，确保设备保管人员能够及时获得设备状态信息，从而保证设备的正常运作和安全管理。

2.4.4 日志信息管理模块数据流

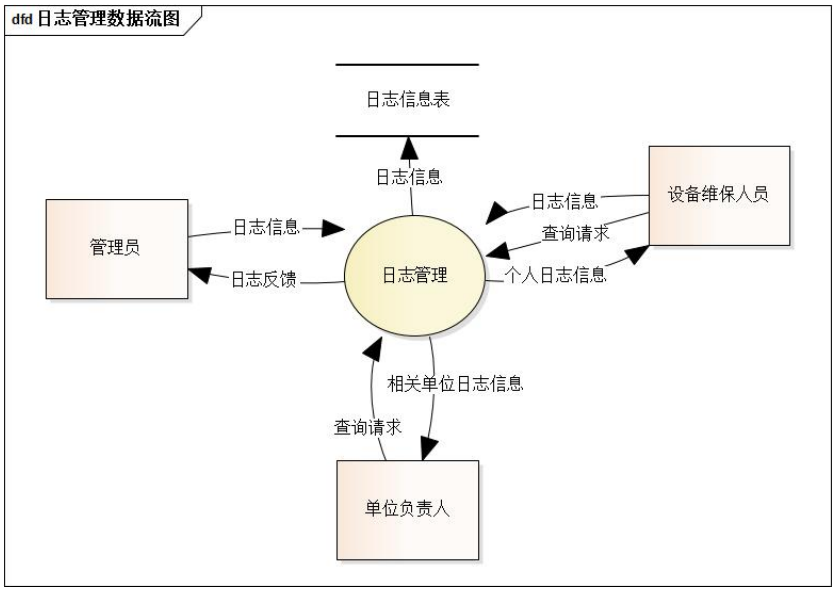

图 15 日志信息管理模块数据流图

这个图展示了日志模块的数据流情况。以下是对该模块的详细介绍：

（1）设备管理模块（核心） 日志管理模块是系统的核心部分，负责管理所有与日志相关的信息。它接收来自其他模块的输入，如日志信息表、年检类型信息，并与外部实体进行数据交互。

（2）外部实体 管理员负责录入和维护日志信息，并将这些信息提交到日志管理模块；日志保管人员:通过日志管理模块获取其管理的日志信息，并在日志即将到期时收到短信通知；单位负责人通过日志管理模块查询和管理本单位的日志信息。

（3）辅助模块 年检类型管理模块负责管理日志的年检信息，并将年检类型信息传递给日志管理模块。年检类型信息包括日志需要定期检验的项目和频率。

（4）数据流动 日志信息表由管理员输入并传递给日志管理模块，日志管理模块基于这些信息进行日志管理和维护；年检类型信息从年检类型管理模块传递到日志管理模块，用于指导日志的定期检验；日志管理模块根据日志到期信息生成短信通知，发送给相关的日志保管人员，以提醒日志即将到期需要检验；日志保管人员和单位负责人可以通过查询事件获取日志的最新信息；当日志接近年检或有其他问题时，日志管理模块会生成到期提示或报警信息。

这个日志信息管理模块确保了日志信息的及时更新和管理，并通过短信通知等机制，确保日志保管人员能够及时获得日志状态信息，从而保证日志的正常运作和安全管理。

2.5 系统整体设计

2.5.1 系统整体用例图

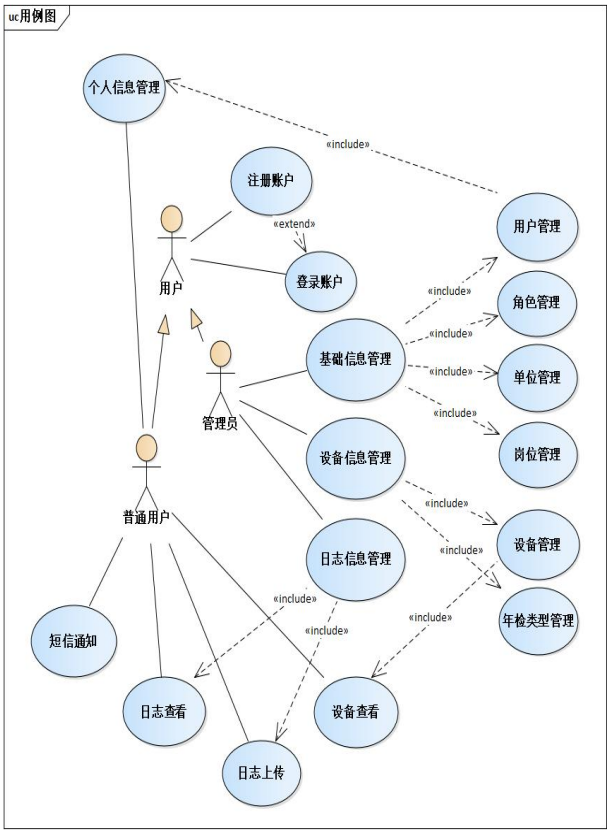

图 16 系统总体用例图

2.5.2 系统整体包图

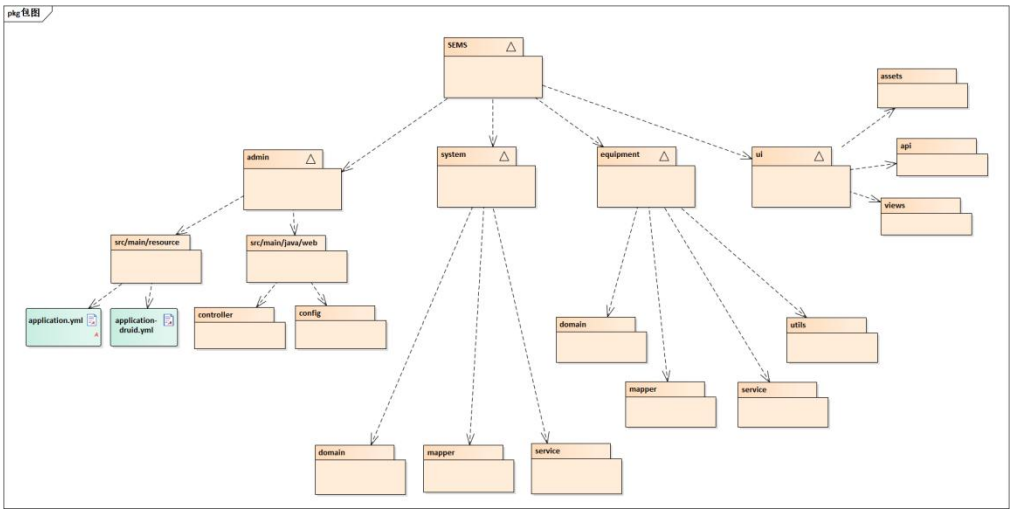

图 17 系统整体包图

2.5.3 系统整体类图

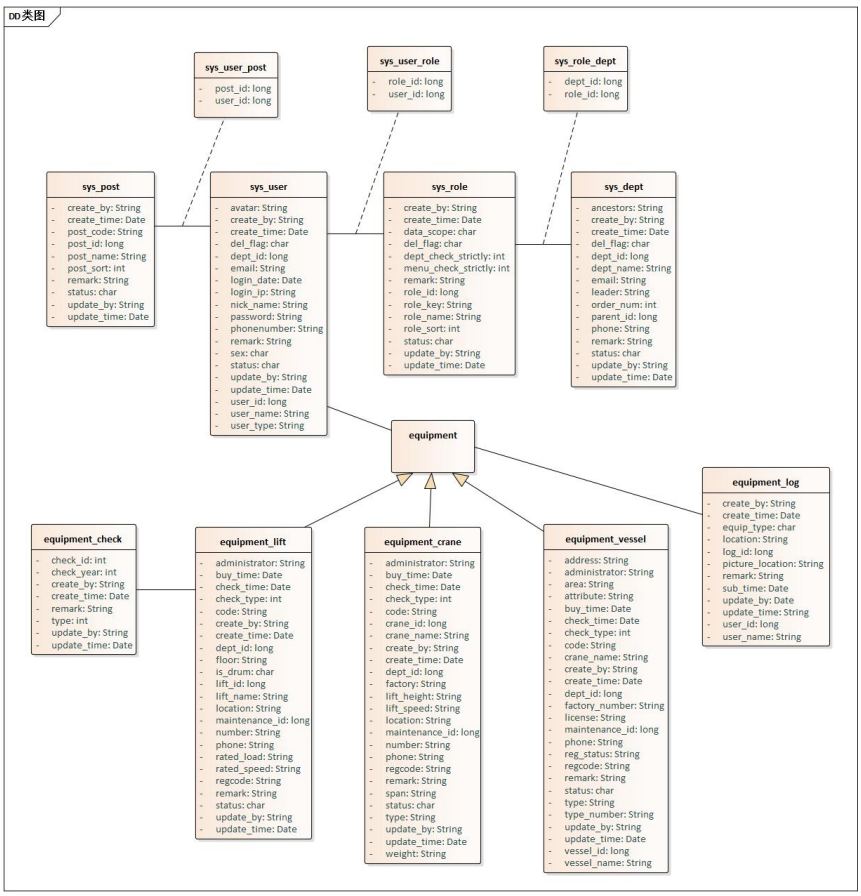

图 18 系统整体类图

2.6 模块详细设计

2.6.1 基础信息管理模块

（1）用户管理模块

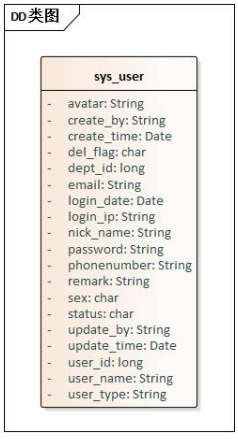

图 19 用户管理模块类图

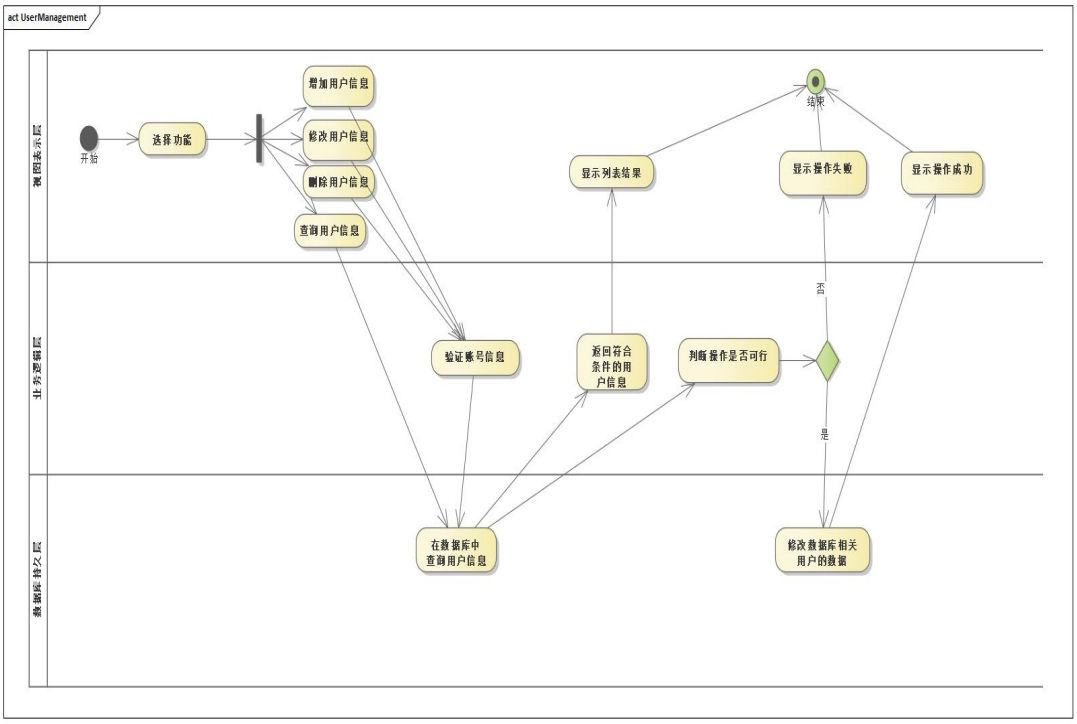

图 20 用户管理模块活动图

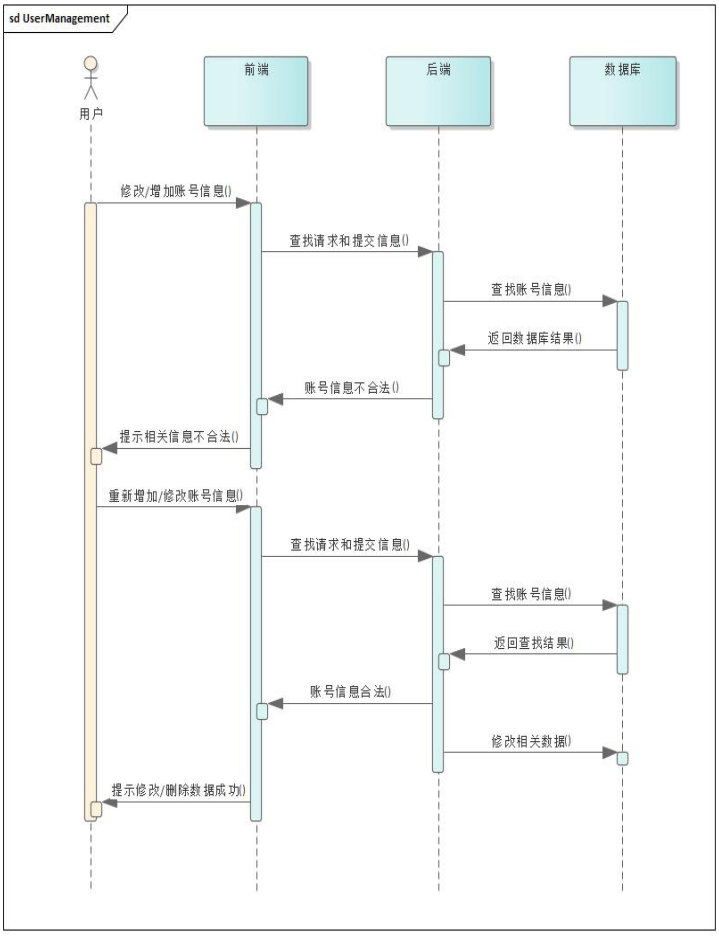

图 21 用户管理模块数据流图

（2）角色管理模块

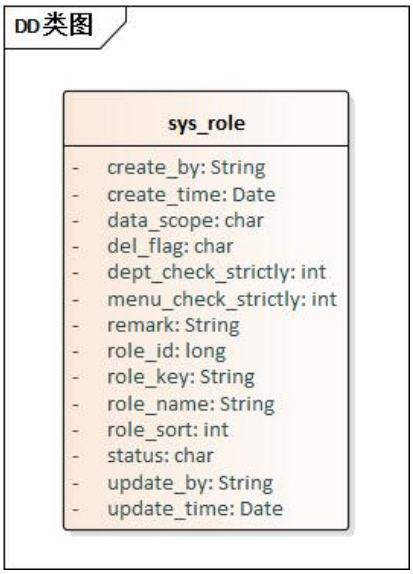

图 22 角色管理模块类图

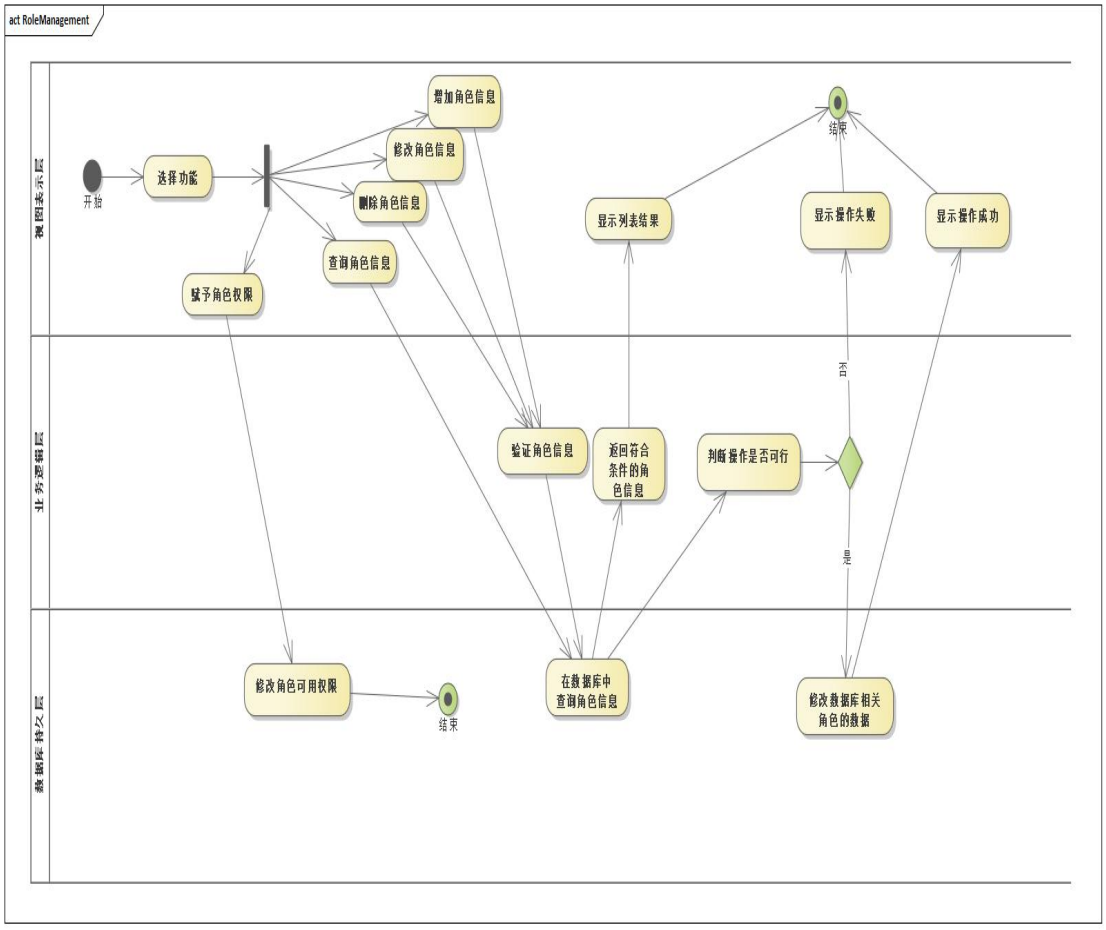

图 23 角色管理模块活动图

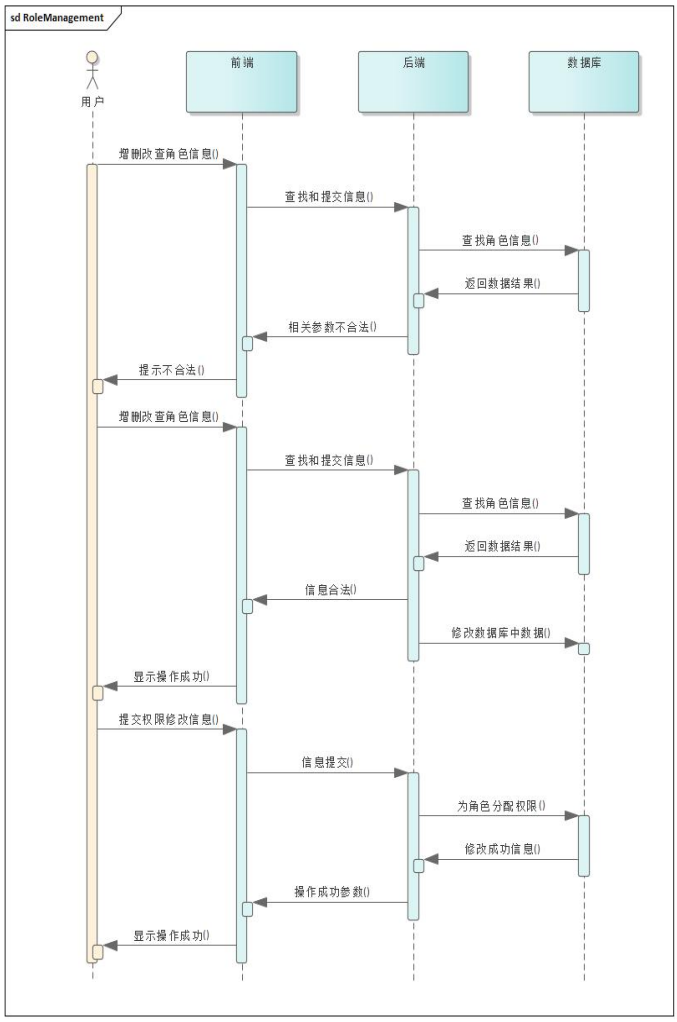

图 24 角色管理模块顺序图

（3）单位管理模块

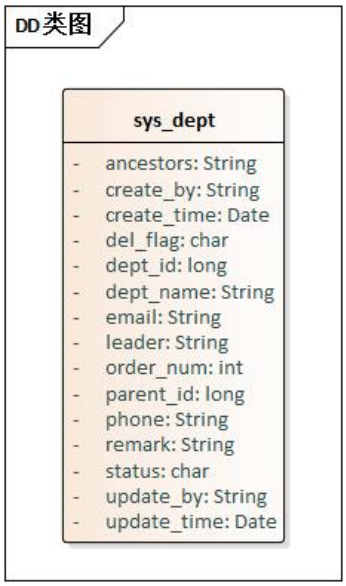

图 25 单位管理模块类图

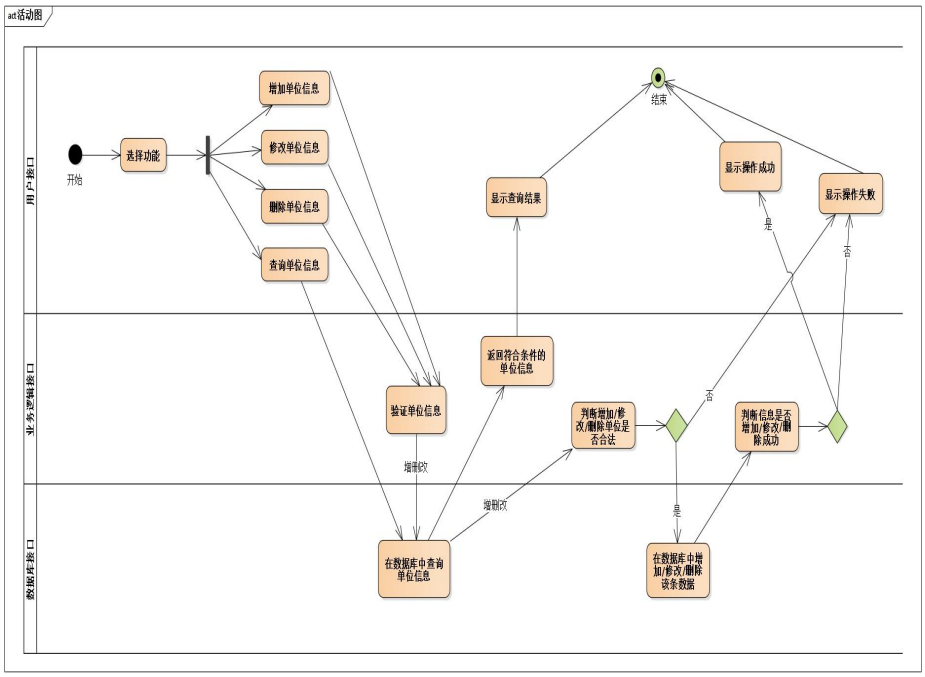

图 26 单位管理模块活动图

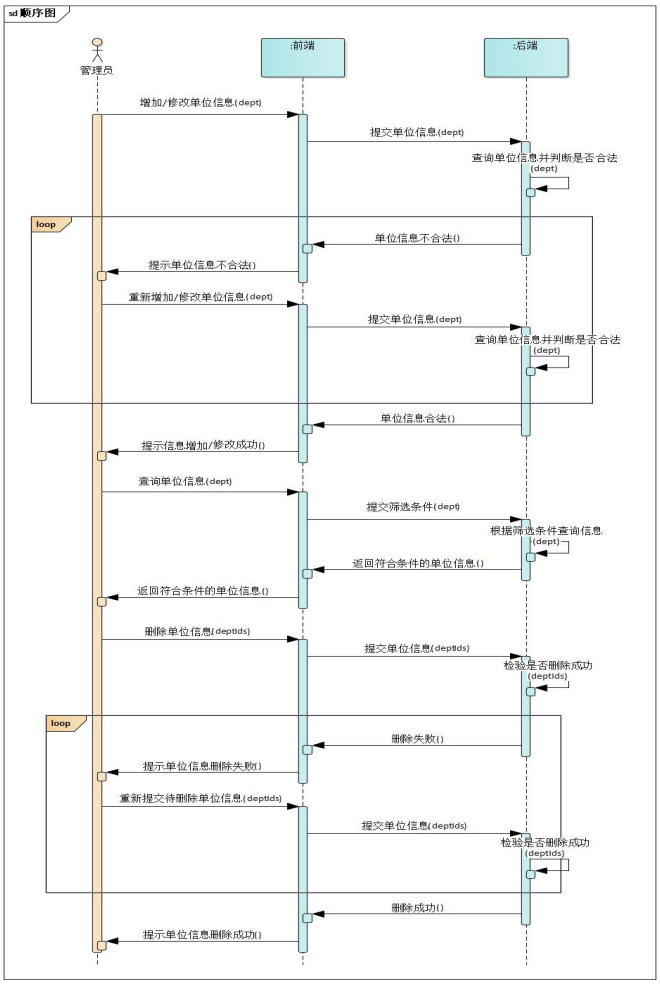

图 27 单位管理模块顺序图

（4）岗位管理模块

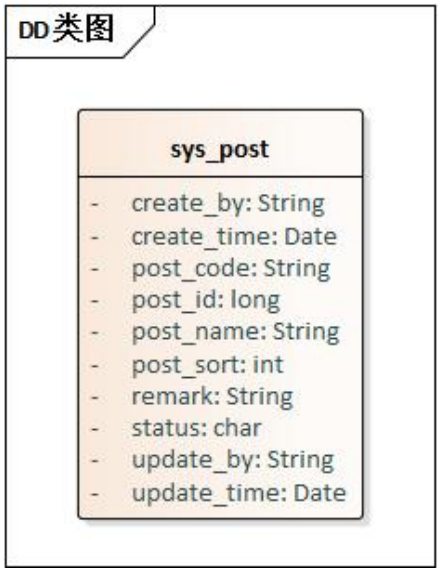

图 28 岗位管理模块类图

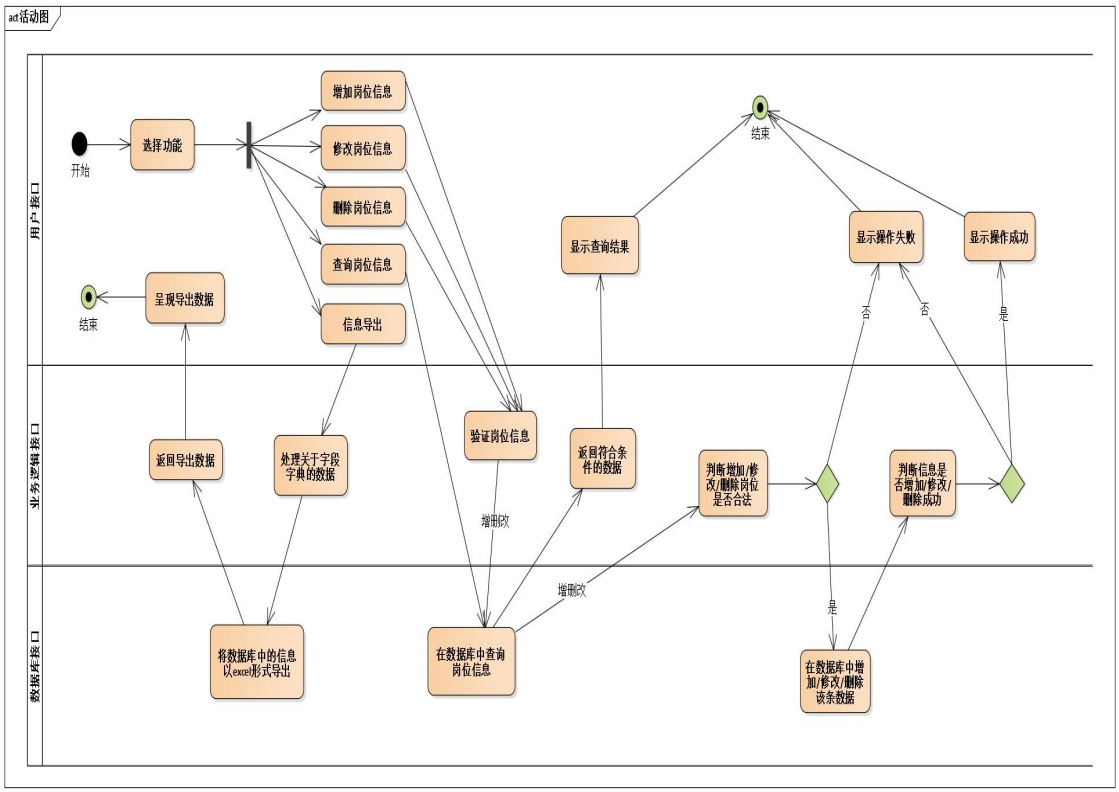

图 29 岗位管理模块活动图

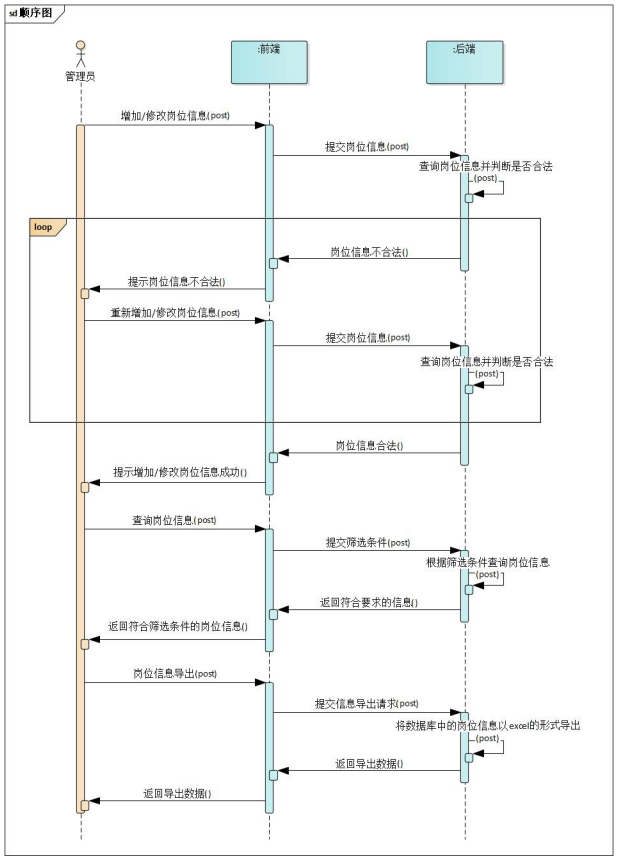

图 30 岗位管理模块顺序图

2.6.2 设备信息管理模块

（1）设备管理模块

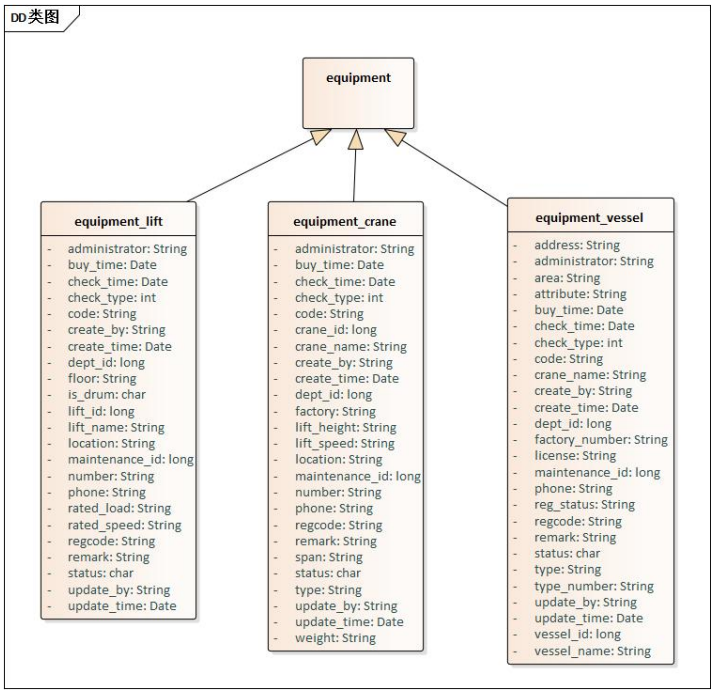

图 31 设备管理子模块类图

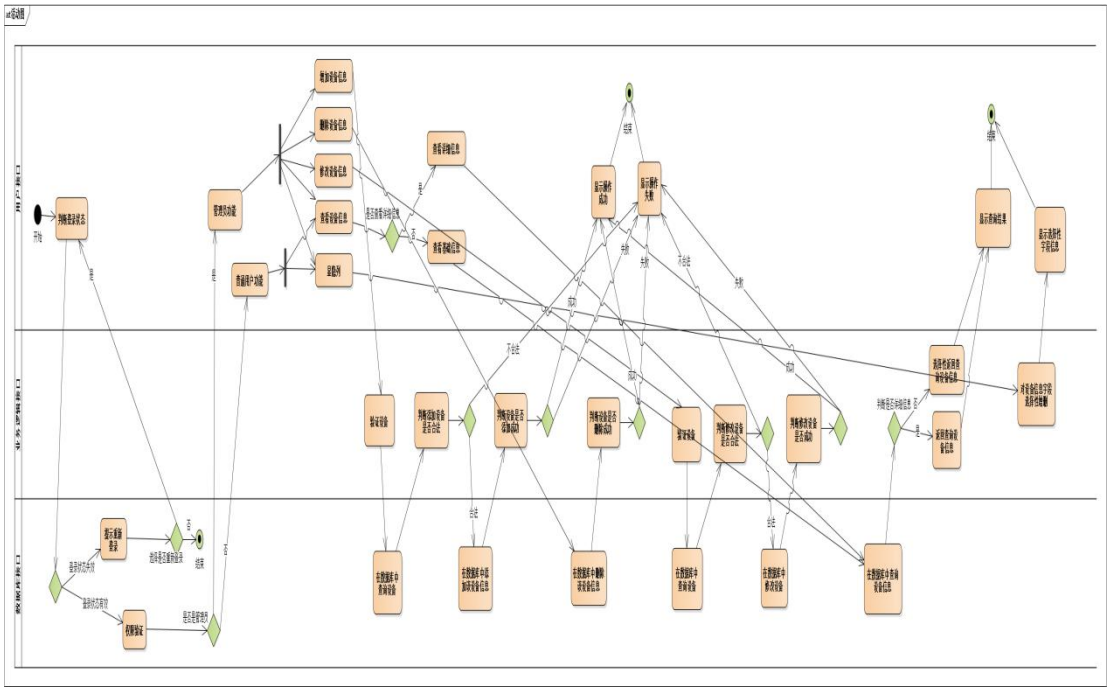

图 32 设备管理模块活动图

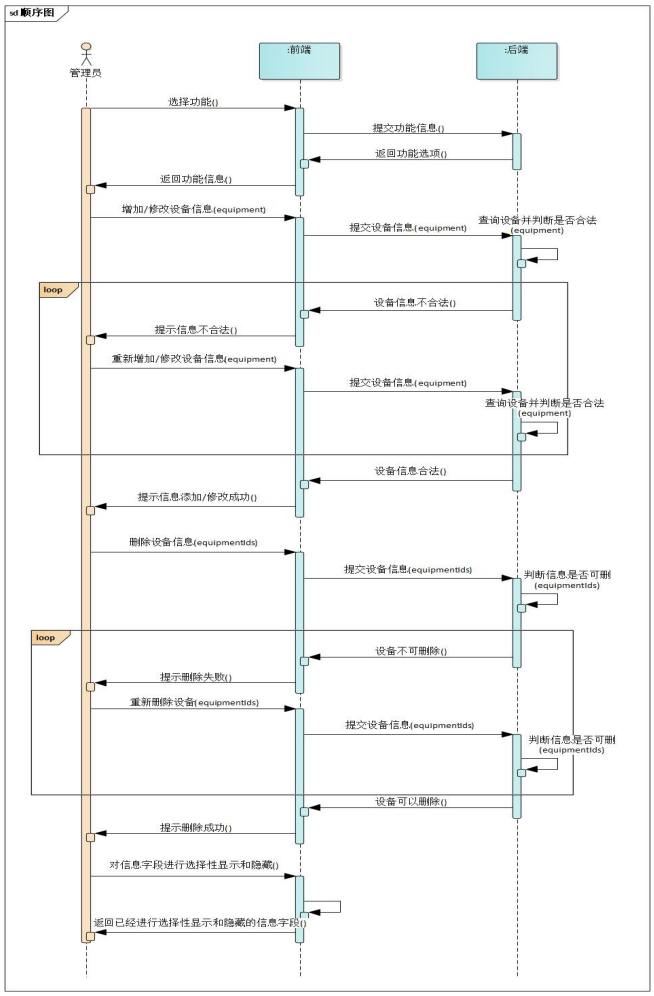

图 33 设备管理模块顺序图

（2）年检类型管理模块

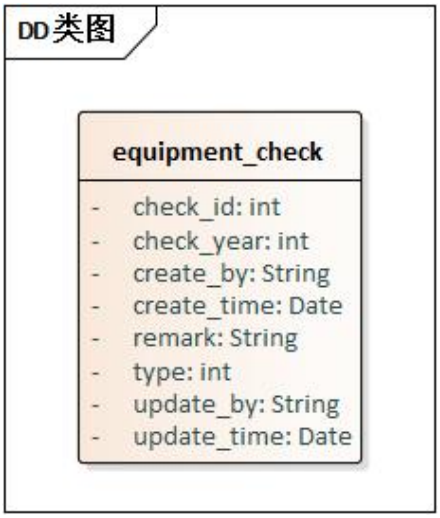

图 34 年检类型管理子模块类图

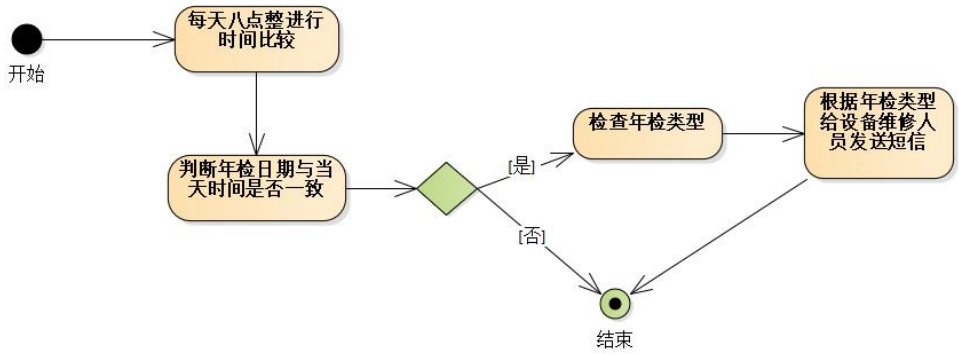

图 35 年检类型管理模块状态图

2.6.3 日志信息管理模块

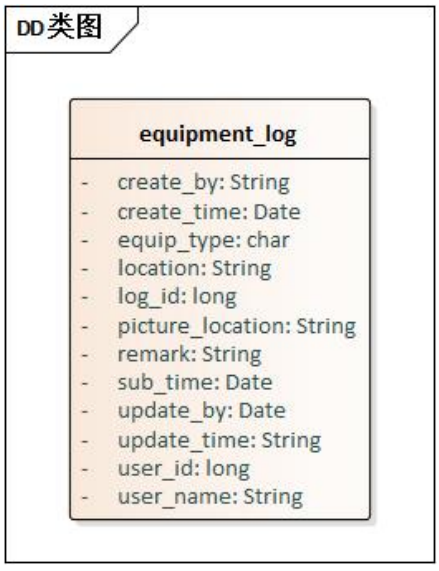

图 36 日志信息管理模块类图

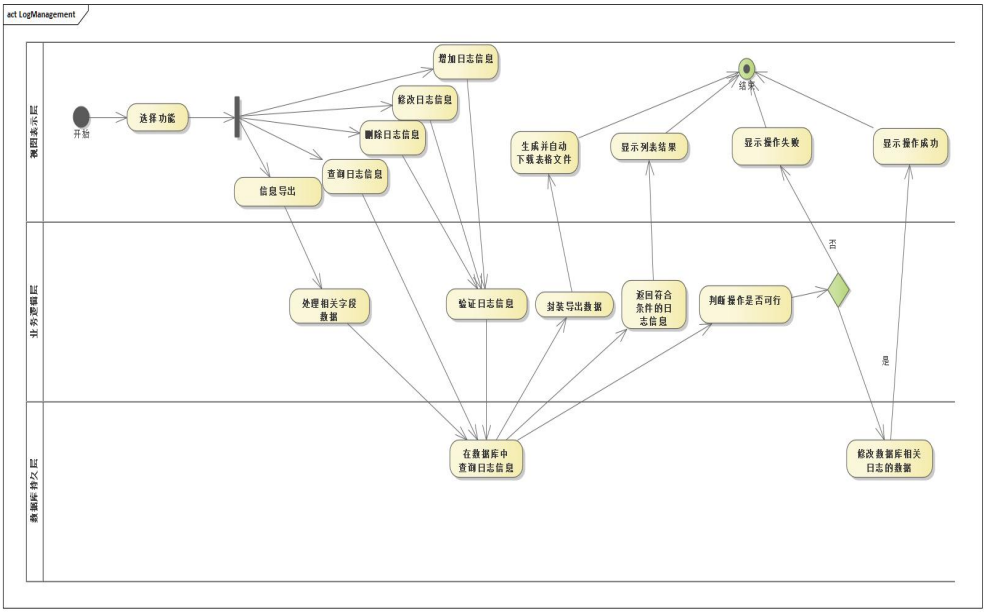

图 37 日志信息管理模块活动图

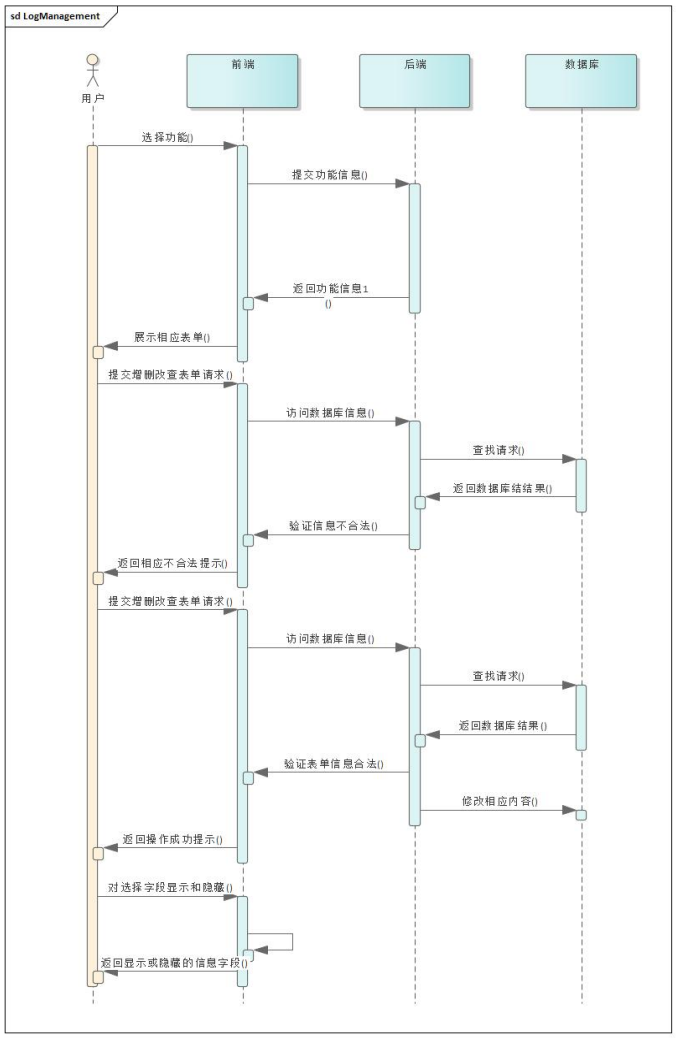

图 38 日志信息管理模块顺序图

第三章 系统详细设计

3.1 系统运行流程

本系统前端使用 JSP，使用 JS 向后端将数据包装成 JSON 格式发送 AJAX 请求，后端使用 SSM 框架，DAO 层向数据库 MySQL 发送查询语句，最后结果返回给到前端也页面展示。对于用户的每一次请求而言，该系统的运行流程中，数据传输到后端处理，再返回的过程可以简要描述如下且由图 39 表示

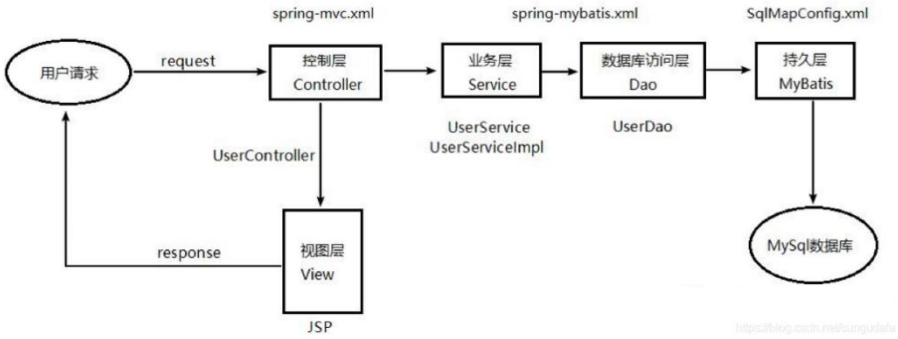

图 39 系统运行流程图

3.1.1 前端页面展示

前端使用 JSP 来构建页面，通过 HTML、CSS 和 JavaScript 等前端技术来实现页面的布局、样式和交互。前端页面中的表单或者其他交互组件，例如按钮、输入框等，可以触发相应的事件，如点击事件、提交表单等。

3.1.2 前端向后端发送请求

当用户在前端页面进行操作或者提交表单时，前端使用 JavaScript 通过 AJAX（Asynchronous JavaScript and XML）技术向后端发送请求。通过构建合适的 URL 和请求参数，可以将用户输入的数据或者其他需要传递给后端的信息包装成 JSON 格式，并通过 AJAX 发送给后端。

3.1.3 后端接收请求

后端接收到前端发送的请求后，SSM 框架中的 Spring MVC 组件会根据请求的 URL 和请求参数来匹配相应的 Controller 方法。Controller 方法使用注解和参数映射，将接收到的请求参数解析到对应的 Java 对象中。

3.1.4 后端处理请求

在 Controller 方法中，后端根据接收到的请求参数进行相应的业务逻辑处理。这可

能涉及到调用 Service 层的方法来处理数据，进行数据库的增删改查操作，或者其他的业务逻辑。

3.1.5 数据库交互

后端在处理请求的过程中，如果涉及到对数据库的操作，会使用 MyBatis 框架来执行相应的数据库操作。通过配置 MyBatis 的映射文件，将 Java 对象与数据库表进行映射，执行相应的 SQL 语句来实现数据的持久化操作。

3.1.6 后端返回响应

在后端处理完请求后，可以将处理结果封装成 JSON 格式的数据，并通过响应对象返回给前端。后端可以使用 Spring MVC 提供的工具类或者注解来实现 JSON 格式数据的封装和返回。前端接收后通过 JavaScript 解析响应数据，并根据解析的结果来进行相应的页面更新和交互操作。

3.2 数据结构设计

3.2.1 基础信息管理模块

（1）用户信息表（sys_user 表) 用户信息表用于系统中保存和管理用户的详细数据。这些数据包括用户的基本身份信息（如用户 ID、用户名、昵称、邮箱和手机号码），用户的账户状态（如账号是否正常、是否被删除），以及用户的登录记录（如最后登录时间和 IP 地址）。表中还记录了账户的创建和更新时间，更新者信息，以及其他相关备注。这些信息帮助系统有效地管理用户账户、跟踪账户活动并维护系统安全。

表 9 用户信息表（sys_user 表)

字段名	字段类型	字段描述
user_id	bigint	用户 ID
dept_id	bigint	部门 ID
user_name	varchar	用户账号
nick_name	varchar	用户昵称
user_type	varchar	用户类型（00 系统用户）
email	varchar	用户邮箱
phonenumber	varchar	手机号码
sex	char	用户性别（0 男 1 女 2 未知）
avatar	varchar	头像地址
password	varchar	密码

status	char	帐号状态（0 正常 1 停用）
del_flag	char	删除标志（0 代表存在 2 代表删除）
login_ip	varchar	最后登录 IP
login_date	datetime	最后登录时间
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间
remark	varchar	备注

（2）角色信息表（sys_role 表) 角色信息表用于存储系统中角色的详细信息，包括角色的唯一标识、名称、权限设置、数据访问范围、菜单和部门选择项的显示关联，以及角色的状态和操作记录。这些数据帮助系统管理和配置用户角色的权限和访问控制。

表 10 角色信息表（sys_role 表)

字段名	字段类型	字段描述
role_id	bigint	角色 ID
role_name	varchar	角色名称
role_sort	varchar	角色权限字符串
role_key	int	显示顺序
data_scope	char	数据范围（1：全部数据权限 2：自定数据权限 3：本部门数据权限 4：本部门及以下数据权限）
menu_check_strictly	tinyint	菜单树选择项是否关联显示
dept_check_strictly	tinyint	部门树选择项是否关联显示
status	char	角色状态（0 正常 1 停用）
del_flag	char	删除标志（0 代表存在 2 代表删除）
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间
remark	varchar	备注

（3）单位信息表（sys_dept 表) 单位信息表用于存储和管理组织结构中的部门数据，包括部门的唯一标识、名称、层级关系、负责人及联系信息，以及部门的状态和操作记录。这些信息帮助系统有效地组织和管理公司或机构的部门结构。

表 11 单位信息表（sys_dept 表)

字段名	字段类型	字段描述
dept_id	bigint	部门 id
parent_id	bigint	父部门 id
ancestors	varchar	祖级列表
dept_name	varchar	部门名称
order_num	Int	显示顺序
leader	varchar	负责人
phone	varchar	联系电话
email	varchar	邮箱
status	char	部门状态（0 正常 1 停用）
del_flag	char	删除标志（0 代表存在 2 代表删除）
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间

（4）岗位信息表（sys_post 表) 岗位信息表用于记录和管理组织中的岗位信息，包括岗位的唯一标识、编码、名称及显示顺序。它还包括岗位的状态（正常或停用）、创建和更新的详细信息，以及备注。该表有助于系统化地管理和配置组织内的岗位信息，以便于员工分配和岗位调整。

表 12 岗位信息表（sys_post 表)

字段名	字段类型	字段描述
post_id	bigint	岗位 ID
post_code	varchar	岗位编码
post_name	varchar	岗位名称
post_sort	int	显示顺序
status	char	状态（0 正常 1 停用）
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间
remark	varchar	备注

3.2.2 设备信息管理模块

（1）电梯信息表（equipment_lift 表) 电梯信息表用于管理和记录电梯设备的详细数据，包括电梯的唯一标识、使用地点、注册代码、设备编码及名称。它还涵盖了电梯的购置日期、额定载程、额定速度和层站信息，以及年检类型和下次年检时间。表中还记录了维保单位信息、是否为鼓式制动机的标志和设备状态（正常、停用或其他状态）。这些信息有助于系统有效管理电梯的使用、维护和检修计划，确保设备安全和正常运行。

表 13 电梯信息表（equipment_lift 表)

字段名	字段类型	字段描述
lift_id	bigint	序号 ID
dept_id	bigint	分布单位
location	varchar	使用地点
regcode	varchar	注册代码
code	varchar	设备代码
administrator	varchar	管理人员
phone	varchar	联系方式
number	varchar	设备编码
lift_name	varchar	设备名称
buy_time	datetime	购置日期
rated_load	varchar	额定载程
rated_speed	varchar	额定速度
floor	varchar	层站
check_type	int	年检类型（1 检验，2 检测，3 检验与检测）
check_time	datetime	下次年检时间
maintenance_id	bigint	维保单位
is_drum	char	是否鼓式制动机（0 不是，1 是）
status	char	状态（0 正常，1 停用，2 其他状态）
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间
remark	varchar	备注

（2）起重机信息表（equipment_crane 表) 起重机信息表用于记录和管理起重机设备的详细数据，包括起重机的唯一标识、使用地点、注册代码、设备编码及名称。表中包含

了起重机的购置日期、规格、生产厂家、型号规格、起升高度、跨度、起升速度等技术参数。此外，它还记录了年检类型和下次年检时间、维保单位信息，以及设备的状态（正常、停用或其他）。这些信息帮助系统全面管理起重机的使用、维护和检修计划，确保设备的安全性和有效运行。

表 14 起重机信息表（equipment_crane 表)

字段名	字段类型	字段描述
crane_id	bigint	序号 ID
dept_id	bigint	分布单位
location	varchar	使用地点
regcode	varchar	注册代码
code	varchar	设备代码
administrator	varchar	管理人员
phone	varchar	联系方式
number	varchar	设备编码
crane_name	varchar	设备名称
buy_time	datetime	购置日期
weight	varchar	规格
factory	varchar	生产厂家
type	varchar	型号规格
lift_height	varchar	起升高度
span	varchar	跨度
lift_speed	varchar	起升速度
check_type	int	年检类型（1 检验，2 检测，3 检验与检测）
check_time	datetime	下次年检时间
maintenance_id	bigint	维保单位
status	char	状态（0 正常，1 停用，2 其他状态）
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间
remark	varchar	备注

（3）压力容器信息表（equipment_vessel 表) 压力容器表用于管理和记录设备的详细信息，包括设备的唯一标识、所在地区、详细地址、设备代码、名称及类型。表中包含了

设备的购置日期、出厂编码、注册编码、注册状态以及使用证编号等信息。还记录了设备的型号、属性、年检类型和下次年检时间、维保单位信息，以及设备的状态（如正常、停用或其他状态）。这些信息有助于系统全面跟踪设备的使用、维护和检修情况，确保设备的正常运行和安全。

表 15 压力容器信息表（equipment_vessel 表)

字段名	字段类型	字段描述
vessel_id	bigint	序号 ID
dept_id	bigint	分布单位
area	varchar	设备所在地区
address	varchar	设备详细地址
code	varchar	设备代码
administrator	varchar	管理人员
phone	varchar	联系方式
factory_number	varchar	出厂编码
regcode	varchar	注册编码
reg_status	varchar	注册状态
vessel_name	varchar	设备名称
buy_time	datetime	购置日期
license	varchar	使用证编号
type	varchar	设备类型
type_number	varchar	设备型号
attribute	varchar	设备属性
check_type	int	年检类型（1 检验，2 检测，3 检验与检测）
check_time	datetime	下次年检时间
maintenance_id	bigint	维保单位
status	char	状态（0 正常，1 停用，2 其他状态，例如：安装中）
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间
remark	varchar	备注

（4）年检类型信息表（equipment_check 表) 年检类型信息表用于存储设备检验的详细信息，包括每个检验记录的唯一标识、对应年份、检验类型（如检验、检测、检测 1.25 或监督）。表中还记录了创建者和更新者的相关信息，以及检验记录的创建和更新时间。备注字段用于添加额外的信息或说明。这些数据有助于系统跟踪和管理设备的检验情况，确保设备符合相关的安全和质量标准。

表 16 年检类型信息表（equipment_check 表)

字段名	字段类型	字段描述
check_id	int	序列 ID
check_year	int	对应年份
type	int	检验类型（1 为检验，2 为检测，3 为检测 1.25，4 为监督）
create_by	varchar	创建者
create_time	datetime	创建时间
update_by	varchar	更新者
update_time	datetime	更新时间
remark	varchar	备注

3.2.3 日志信息管理模块

（1）日志信息管理表（equipment_log 表) 日志信息管理表用于记录用户上传设备图片的详细信息，包括每条记录的唯一标识、上传人的 ID 和姓名、上传时间、设备类型、设备地点以及图片的存储位置。表中还包含记录的创建者、创建时间、更新者和更新时间。这些信息有助于系统跟踪设备图片的上传和管理，确保设备数据的完整性和可追溯性。

表 17 日志信息管理表（equipment_log 表)

字段名	字段类型	字段描述
log_id	bigint	序列 ID
user_id	bigint	上传人 ID
user_name	varchar	上传人姓名
sub_time	datetime	上传时间
equip_type	int	设备类型
location	varchar	设备地点
picture_location	varchar	图片地点
create_by	varchar	创建人
create_time	datetime	创建时间

update_by	varchar	更新人
update_time	datetime	更新时间
remark	varchar	备注

3.3 项目结构设计

为了使得项目在开发过程中更加的高效，一个好的项目结构是必不可少的。在开发过程中设计了如图的项目结构，图示的仅是部分模块的项目结构设计图。这样做的好处有以下几点：

（1）结构清晰可维护 该项目结构按照 MVC 模式进行组织，将不同的组件和功能分层放置，使得代码的逻辑和职责清晰可见。这有助于团队成员更好地理解和维护代码，提高项目的可维护性。

（2）分离关注点 通过将控制器、服务、数据访问等不同层次的代码分开放置，实现了关注点的分离。这样的结构使得代码更加可读、可测试，降低了不同模块之间的耦合性。

（3）可扩展性强 项目结构的模块化设计使得可以轻松地进行功能的扩展和修改。例如，可以通过添加新的控制器、服务类来实现新的功能模块，并且不会对已有的代码产生太大的影响。

（4）提高开发效率 良好的项目结构设计有助于提高开发效率。通过按照约定的目录结构组织代码，开发人员可以更快地定位和编辑相关的文件，减少开发过程中的查找和配置时间。

（5）便于版本控制和部署 清晰的项目结构使得版本控制和部署过程更加简单。不同模块的代码和配置文件都有明确的位置，可以方便地进行版本管理和部署操作。

接下来讲解各个目录主要存放的文件代码作用。其中：

src/main/java：主要存放项目的 Java 源代码文件。

controller 用于存放控制器(Controller)类，负责处理请求和返回响应。

dto 下存放数据传输对象(DTO)类，用于在不同层之间传递数据。

entity：存放实体类(Entity)或领域模型类，用于表示数据库中的表或领域对象。

enums：存放枚举类，用于定义一组相关的常量。

interceptor：存放拦截器(Interceptor)类，用于拦截和处理请求。

mapper：存放 MyBatis 的映射器(Mapper)接口，定义数据库操作的方法。

service：存放服务(Service)类，负责业务逻辑的处理。

hnust.util：存放工具类，提供一些通用的功能方法。

src/main/resources：主要存放项目的配置文件和资源文件。

mapper：存放 MyBatis 的映射文件(XML)，定义数据库操作的 SQL 语句。

mybatis：存放 MyBatis 的配置文件，包括 mybatis-config.xml 等。

spring：存放 Spring 框架的配置文件，包括 spring-mvc.xml、spring-mybatis.xml

db.properties：存放数据库相关的配置信息，数据库连接 URL、用户名、密码。

logback.xml：存放日志配置文件，用于配置日志的输出格式和级别等。

src/main/webapp：存放 Web 应用的静态资源和 WEB-INF 目录。

resource：存放 Web 应用的静态资源文件，如图片、CSS 样式表、JS 文件等。

WEB-INF：存放 Web 应用的受保护资源和配置文件。

view：存放视图(View)文件，是 JSP 文件或模板文件，用于呈现页面内容。

Admin：存放管理员相关的视图文件。

Home：存放用户/前台相关的视图文件。

src/test：存放项目的测试代码文件，用于编写单元测试和集成测试。

这样的项目结构按照 MVC（Model-View-Controller）的设计模式进行组织，将不同的组件和功能分层放置，使得项目结构清晰、可维护性高，并提供了一种良好的代码组织方式。

```
src
├── main
│   ├── java
│   │   └── com
│   │       └── hnust
│   │           ├── controller
│   │           ├── dto
│   │           ├── entity
│   │           ├── enums
│   │           ├── interceptor
│   │           ├── mapper
│   │           ├── service
│   │           └── util
│   ├── resources
│   │   ├── mapper
│   │   ├── mybatis
│   │   ├── spring
│   │   ├── db.properties
│   │   └── logback.xml
│   └── webapp
│       ├── resource
│       │   └── assets
│       │       ├── css
```

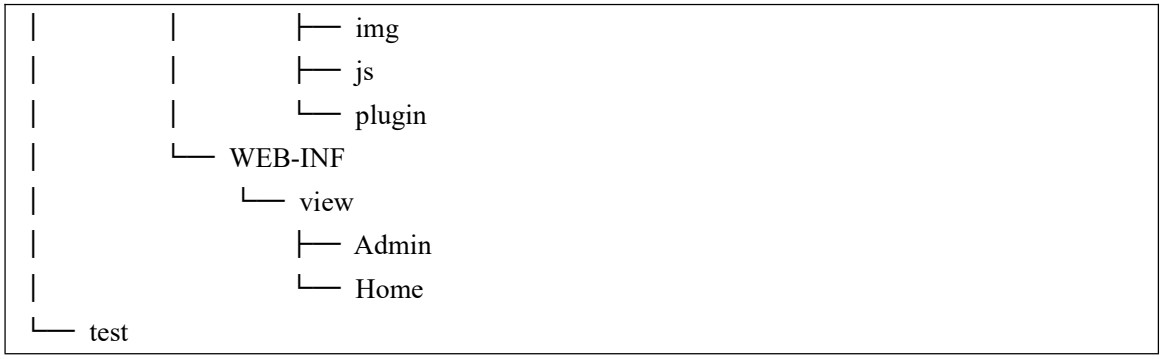

图 40 项目结构设计图

第四章 系统部署与测试

4.1 系统部署

4.1.1 系统部署地址

http://hnustsems.cn

4.1.2 部署说明

本系统部署在阿里云 ESC 提供的云服务器上，采用前后端分离的架构。为简化部署过程，使用了可视化工具宝塔面板，并结合 Xshell 和 Xftp 进行操作。宝塔面板是一款基于 Web 的服务器管理软件，旨在帮助用户在 Linux 系统上高效地部署和管理服务器环境。通过宝塔面板，用户可以轻松管理网站、数据库、FTP 等服务，并支持一键安装常见的 Web 应用程序及其环境配置。接下来，将提供云服务器的详细配置信息及其软件配置。

表 18 云服务器配置信息

CPU&内存	2 核(vCPU） 4GiB
公网 IP	8.134.141.232
主私网 IP	172.20.78.25
操作系统	CentOS 7.8 64 位
网络类型	专有网络
公网带宽	100 Mbps（峰值）

表 19 云服务器上软件配置信息

软件名	版本
Nginx	1.20.2
MySQL	8.0.24
Redis	7.2.4
Jdk	1.8
宝塔 SSH 终端	1.0

4.1.3 部署步骤

（1）前端打包成 dist 文件 进入前端项目目录，使用 npm 打包，在终端输入“npm run build”命令，命令会根据配置将前端代码打包成 dist 文件夹，通常位于项目的根目录中。

（2）后端打包成 jar 包 进入后端项目目录，使用 Maven 打包，在终端输入“mvn clean package”命令，会在 target 目录中生成一个 jar 文件。

（3）在 Linux 系统中部署并运行 通过 Xftp 上传 dist 文件夹和 jar 包到服务器的目

标目录。使用 Nginx 部署前端项目，编辑 Nginx 配置文件来指向 dist 目录，保存配置并重启 Nginx。运行 jar 包，通 Xshell 在终端输入后台运行指令“nohup java jar semsadmin.jar &”，输入指令“tail f nohup.out”查看 nohup.out。

（4）补充说明 环境变量配置，确保服务器上配置了所需的环境变量，例如 JAVA_HOME，PATH 中包含 node 和 npm，并且在配置文件中指定了正确的 BASE_URL 等变量。防火墙配置，如果后端服务需要暴露在互联网，确保服务器防火墙配置开放了相关端口（例如 80, 443, 8080 等）。

4.2 系统测试

4.2.1 测试环境

表 20 手机端测试环境

手机型号	Honor 50
处理器	高通骁龙 778G
型号	NTHAN00
RAM 容量	8GB
ROM 容量	256GB
屏幕分辨率	2340×1080 像素
手机操作系统	Android 12.0

表 21 PC 端测试环境

电脑型号	华硕无畏 15pro
处理器	12th Gen Intel(R) Core(TM) i512500H 2.50 GHz
RAM 容量	16GB
ROM 容量	512GB
屏幕分辨率	2880×1620 像素
操作系统	Windows 11 64bit

4.2.2 系统测试过程

为确保重要资产设备维保预警管理系统的所有功能正常工作，对系统进行了全面且系统的功能测试。此次测试主要覆盖以下关键模块：用户管理、角色管理、单位管理、岗位管理、设备管理、年检类型管理、日志管理。通过这些模块的功能测试，确保系统的稳定性和用户体验，及时发现并修复潜在缺陷，从而提升系统的质量与可靠性。撰写这些模块的测试用例，能够全面覆盖系统的各项核心功能，保障用户在使用过程中顺畅且稳定的体验。

表 22 用户注册测试用例

用例名称	用户管理测试用例
目的	管理重要资产设备维保预警管理系统用户账号信息的系统管理，用户登录修

	改信息等功能操作，管理员负责的用户权限等级的分配
前提	数据库中含有相应的权限等级和用户信息数据
测试流程	使用者打开用户信息管理界面，进行相应的功能操作
预期结果	用户相应操作得以成功执行
实际结果	与预期结果一致

表 23 角色管理测试用例

用例名称	角色管理测试用例
目的	管理重要资产设备维保预警管理系统模块，管理员可以创建、修改、查看和删除系统存在的角色
前提	用户已经通过身份验证登录到系统中，用户拥有管理员权限
测试流程	使用者打开角色信息管理界面，进行相应的功能操作
预期结果	用户相应操作得以成功执行
实际结果	与预期结果一致

表 24 单位管理测试用例

用例名称	单位管理测试用例
目的	管理重要资产设备维保预警管理系统单位信息，包括校内单位和校外单位
前提	用户已经通过身份验证登录到系统中，用户拥有管理员权限
测试流程	使用者打开单位信息管理界面，进行相应的功能操作
预期结果	用户相应操作得以成功执行
实际结果	与预期结果一致

表 25 岗位管理测试用例

用例名称	岗位管理测试用例
目的	管理重要资产设备维保预警管理系统岗位信息，包括开发人员、设备总负责人、单位负责人和维保人员
前提	用户已经通过身份验证登录到系统中，用户拥有管理员权限
测试流程	使用者打开岗位信息管理界面，进行相应的功能操作
预期结果	用户相应操作得以成功执行
实际结果	与预期结果一致

表 26 设备管理测试用例

用例名称	设备管理测试用例
目的	管理重要资产设备维保预警管理系统中的电梯、起重机和压力容器等设备信息，并在指定时间自动短信通知维保人员进行维修或维保特种设备
前提	用户已经通过身份验证登录到系统中
测试流程	使用者打开设备信息管理界面，进行相应的功能操作
预期结果	用户相应操作得以成功执行
实际结果	与预期结果一致

表 27 年检类型管理测试用例

用例名称	年检类型管理测试用例
目的	管理重要资产设备维保预警管理系统中的年检类型信息，包括年检周期中对应的年份、年检类型和备注等信息

前提	用户已经通过身份验证登录到系统中，用户拥有管理员权限
测试流程	使用者打开年检类型信息管理界面，进行相应的功能操作
预期结果	用户相应操作得以成功执行
实际结果	与预期结果一致

表 28 日志管理测试用例

用例名称	日志管理测试用例
目的	管理重要资产设备维保预警管理系统中的用户所填写的检修日志信息
前提	用户已经通过身份验证登录到系统中，用户拥有管理员或检修员权限
测试流程	使用者打开日志信息管理界面，进行相应的功能操作
预期结果	用户相应操作得以成功执行
实际结果	与预期结果一致

4.2.3 测试结果分析

在测试结果的详细分析中，对重要资产设备维保预警管理系统进行了深入的功能性检验。结果显示，系统在各关键功能模块的性能和稳定性方面表现出色，无严重缺陷。用户管理、角色管理、单位管理、岗位管理、设备管理、年检类型管理及日志管理等核心功能均通过了全面且严格的测试，实际结果与预期高度一致。这些测试结果证明了系统的操作流畅性和用户友好性，为系统的正式上线提供了坚实的保障，确保用户能够获得稳定、可靠的服务体验。

第五章 小结与心得体会

5.1 项目创新点与不足

5.1.1 系统的创新点

（1）多方位管理特种设备的信息 本系统具有细致化的信息管理功能，包括设备的“单位”、“名称”、“型号”及“制造日期”等信息，通过将这些信息传递到系统中，并在系统内进行数据逻辑规划，提供了全面、详尽的管理信息并实现了设备相关信息的集中管理，使得特种设备检测检验、维保等有记录可供查询、报修与维修更便捷规范。

（2）短信提醒特种设备维保 本系统通过发送提醒短信方式帮助用户记住特种设备的半月检、季度检、半年检及年检等重要节点，提前提示用户进行维保工作。这项提醒功能与系统的其他功能相结合，形成闭环管理系统，大大降低了由于维护不到位或漏检等导致的安全风险，提升了特种设备的安全运行能力，也为提高特种设备运维效率提供了有力支持。

5.1.2 系统存在以下不足之处

（1）在设备信息管理中，目前存储了“设备负责人”的姓名，而非其系统账号 ID。这种做法主要是考虑到部分负责人可能尚未在系统中注册账号。然而，这种处理方式在权限分配方面引发了一些复杂的问题，使得问题处理变得相当繁琐。因此，我们认识到此部分仍有较大优化空间，需要进一步探索更为合理和高效的解决方案。

（2）在系统的短信发送功能中，我们发现与之前小组成员所熟悉的业务存在差异。开发过程中，团队成员对新业务的操作还不够熟练，这反映了我们在该功能相关经验上的不足。为此，我们需要加强学习和实践，以提升对新业务的掌握能力。

5.2 系统优化思路与解决方案

5.2.1 数据表优化与持久化存储

为了确保系统的长期稳定运行，可对各数据表中的字段进行标准化的数据库存储管理。这将允许用户对数据字段进行增删改查等基础操作，实现更加自动化、可操作化的管理方式。通过这种优化，不仅可以提高系统的持久性和灵活性，还能显著减少开发和维护的工作量。

5.2.2 设备定位功能的增强

为了提升维保人员的工作效率，我们计划引入设备定位功能。通过集成先进的定位插件，系统将能够直观地显示设备的实际位置。这一功能将使维保人员能够更加便捷、快速地定位设备位置，从而加快维修和保养工作的进程。

5.2.3 实时监控与可视化分析

借助 ECharts 等可视化工具，系统可以提供直观、可交互的图表显示，实时监控设备状态和系统运行情况。这不仅有助于管理者及时发现潜在问题，还可以通过图形化展示来增强数据的可读性和决策的精准性。

5.3 项目分工情况

表 29 小组成员分工情况

姓名	学号	开发系统分工	报告撰写分工
陈琪琪	2102010629	电梯设备管理模块、起重机设备管理模块、压力容器管理模块、登录注册页面、首页	系统部署与测试
邓子豪	2101050325	维修日志管理模块、报销凭证管理模块	系统概要设计、系统详细设计
章丽红	2106040128	用户管理模块、角色管理模块、单位管理模块、岗位管理模块	系统需求分析

5.4 成员个人总结

表 30 小组成员个人总结

姓名	个人总结
陈琪琪	在这个项目中，我不仅深入理解了系统开发的全过程，还在多个方面提升了自己的能力。首先，通过参与项目的各个阶段，从需求分析到最终部署，我的动手能力得到了显著提高，特别是在前后端分离的开发和部署方面，积累了丰富的实战经验。其次，在解决实际问题时，我学会了如何高效地调研、分析和应用新技术，加深了对技术应用的理解。此外，团队合作过程中，我的沟通和协作能力也得到了加强，能够更好地与他人协调工作，共同解决复杂问题。通过这个项目，我不仅提高了技术技能，还增强了面对挑战时的应变能力和问题解决能力，为未来的项目开发打下了坚实的基

	础。
邓子豪	在这次生产实习中，我对使用 SSM 框架构建特种设备管理系统的过程有了一些心得体会。首先，学习和掌握 SSM 框架是一个关键的前提。SSM 框架整合了 Spring、Spring MVC 和 MyBatis，为开发 Java Web 应用提供了强大的支持。通过学习和实践，我深刻理解了框架的原理和使用方式，能够高效地开发出符合需求的系统。对于特种设备关系系统的实际应用，需求分析和系统设计是非常重要的。我需要考虑到不同角色的用户需求，并设计相应的功能和界面。合理的数据库设计和业务逻辑的实现对系统的可用性和性能至关重要。此外，前后端的分离和合作也是关键。在该系统中，前台和后台具有不同的功能和权限需求，因此需要明确的接口设计和数据传输方式。通过与前端开发人员的沟通和配合，确保系统的功能和用户体验得到有效实现。最后，系统的性能和稳定性测试是不可忽视的一部分。通过使用工具如 Apache JMeter 进行压力测试，可以评估系统在不同负载下的表现，并发现潜在的性能问题和瓶颈。在测试的基础上，可以针对性地进行优化和调整，以提供更好的用户体验。
章丽红	在参与开发用户管理模块、角色管理模块、单位管理模块和岗位管理模块的过程中，我对系统开发的全过程有了更深的理解，特别是在需求分析、系统设计和模块实现方面积累了宝贵的经验。在开发用户管理模块时，我学习到如何高效地进行需求分析和系统设计。用户管理模块是整个系统的基础，涉及用户的注册、登录、权限管理等功能。岗位管理模块的开发让我更深入地理解了系统中的职位和职责分配。通过需求分析，我设计了岗位与用户、角色之间的关系模型，并在实现过程中考虑了如何确保数据的一致性和系统的高效性。岗位管理模块不仅需要处理复杂的数据关系，还需要与其他模块进行有效的协作，确保系统的整体性和可维护性。总的来说，在需求分析阶段，我深入了解了不同用户角色的需求，设计了合理的数据结构和业务逻辑。在整个开发过程中，系统需求分析是关键的一环。通过与用户和团队的紧密沟通，我能够准确捕捉到系统的核心需求，并在设计阶段将这些需求转化为具体的功能和模块。合理的需求分析不仅提高了开发效率，还确保了系统的易用性和用户体验。