****

**湖南工程职业技术学院**

**毕业设计**

**题 目**  宠物领养系统的设计与实现

**毕业设计类型** ☑产品设计 □工艺设计 □ 方案设计

**专 业 名 称**  软件技术

**所 属 班 级**  2021031102

**学 生 姓 名**  朱 偲 壕

**学 生 学 号**  202003110247

**指 导 老 师 （学校）**  陈鑫亚

**指 导 老 师 （企业）**  李怡彤

**湖南工程职业技术学院教务处 制**

**目 录**

[1 项目概况 1](#_Toc136091498)

[1.1 项目背景 1](#_Toc136091499)

[1.2 系统开发的意义 1](#_Toc136091500)

[1.3 系统运行环境 1](#_Toc136091501)

[2 系统需求分析 2](#_Toc136091502)

[2.1 可行性分析 2](#_Toc136091503)

[2.2 功能需求 2](#_Toc136091504)

[2.3 用例描述 6](#_Toc136091505)

[2.4 性能需求 9](#_Toc136091506)

[3 系统设计 11](#_Toc136091507)

[3.1 设计思想 11](#_Toc136091508)

[3.2 系统架构设计 11](#_Toc136091509)

[3.3 功能模块设计 12](#_Toc136091510)

[3.4 数据库设计 13](#_Toc136091511)

[3.4.1 概念结构设计 13](#_Toc136091512)

[3.4.2 逻辑结构设计 14](#_Toc136091513)

[3.4.2 数据库连接 25](#_Toc136091514)

[3.5 详细设计 25](#_Toc136091515)

[3.5.1 用户信息类图 25](#_Toc136091516)

[3.5.2 用户信息管理时序图 26](#_Toc136091517)

[3.5.3 资产借还类图 28](#_Toc136091518)

[3.5.4 资产借还信息管理时序图 29](#_Toc136091519)

[4 系统实现 30](#_Toc136091520)

[4.1 用户登录 30](#_Toc136091521)

[4.2 个人信息 32](#_Toc136091522)

[4.3 基本数据管理 32](#_Toc136091523)

[4.3.1 部门管理 32](#_Toc136091524)

[4.3.2 人员管理 38](#_Toc136091525)

[4.3.3 资产类别管理 42](#_Toc136091526)

[4.4日常管理 46](#_Toc136091527)

[4.4.1 宠物领养 46](#_Toc136091528)

[4.4.2 资产审批管理 50](#_Toc136091529)

[4.5 统计分析 52](#_Toc136091530)

[4.5.1 功能实现 52](#_Toc136091531)

[4.5.2 关键代码 54](#_Toc136091532)

[5 系统测试 55](#_Toc136091533)

[5.1 测试方法与策略 55](#_Toc136091534)

[5.2 登录模块测试 55](#_Toc136091535)

[5.3 部门管理模块测试 56](#_Toc136091536)

[5.4 资产入库模块测试 59](#_Toc136091537)

[5.5 退出系统测试 62](#_Toc136091538)

# 1 项目概况

## 1.1 项目背景

随着城市化进程的加快，宠物流浪和遗弃问题日益凸显。传统的宠物领养流程常常存在信息不对称、流程繁琐等问题，导致许多可爱的宠物难以找到新家。本毕业设计致力于开发一个宠物领养系统，目标是为流浪或被遗弃的宠物寻找温暖的家庭，同时也为希望领养宠物的人提供一个方便、可靠的平台。

该系统旨在简化领养流程，提高透明度，并提供一个互动的社区，促进宠物爱好者之间的交流。通过详细的宠物资料展示、用户评价机制和领养后跟踪服务，系统将改善领养体验，提高宠物的领养率，同时降低因不合适领养而可能产生的再次遗弃风险。

本系统整体采用B/S的系统结构、MVC设计模式、MySQL关系型数据库，将其以一个网页的形式展示在网络平台，用户可以通过网页实现宠物的浏览、领养等功能。

## 1.2 系统开发的意义

本项目不仅具有技术挑战性，更带有强烈的社会责任感。作为一个热爱动物并关心动物福利的软件开发者，我希望通过我的技术力量，为解决社会问题贡献一份力量，让更多宠物和人类建立起长久而稳固的友谊。此外，通过实现这一平台，我能够将所学知识与实际应用相结合，为我的软件开发职业生涯打下坚实的基础。

## 1.3 系统运行环境

（1）运行环境。为满足该系统的运行需要，对操作系统、Web服务器等提出了要求，具体见表1-1所示。

表1-1 运行环境一览表

|  |  |
| --- | --- |
| CPU | 要求在2GHZ以上 |
| 内存 | 4GB以上 |
| 硬盘 | 40GB以上 |
| 操作系统 | Windows10或Window11 |
| Web服务器 | Tomcat 8.5.47 |
| 数据库服务器 | MySQL 5.0以上 |

（2）开发环境：操作系统Windows 11，Web服务器Tomcat8.5.47，开发工具 Idea旗舰版 Maven，版本控制 Git，数据库MySQL5.7。

# 2 系统需求分析

## 2.1 可行性分析

基于对宠物领养问题的系统调查，本分析旨在评估宠物领养系统的开发和实施是否切实可行，确保项目的成功实施，避免投资失误。通过深入研究，本项目的可行性从以下四个主要方面进行了详尽分析：

经济可行性： 经济可行性分析主要关注项目潜在的经济效益和成本效益比。经预算评估，本系统的开发成本包括初期的市场调研、软硬件投资、人力资源、培训与支持以及长期的维护费用。通过比较现有市场上同类系统的成本和收费模式，结合预期的广告收入、合作伙伴的资助和用户付费内容，我们预计系统在两年内可收回成本，之后将实现经济效益的稳定增长。此外，系统预计将间接为用户节省时间和资源，提高领养效率，具有良好的市场推广前景。

技术可行性： 技术可行性评估重点关注在现有技术资源下系统开发的可能性。本系统选用成熟的开发工具与技术框架，如 IntelliJ IDEA 作为开发环境，前端设计采用流行的 Vue.js 框架和 Element UI 库，后端数据库选用广泛支持的 MySQL 8.0。这些技术的稳定性和成熟度为系统开发提供了坚实基础。结合开发团队对这些技术的熟悉度，以及持续的技术支持和更新，确保了系统的技术实施是可行的。

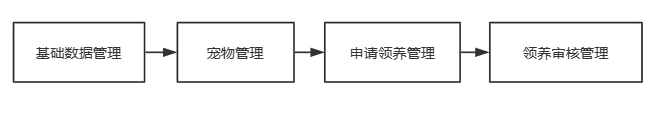
管理可行性： 管理可行性主要从组织和管理的角度评估项目的实施可能性。本系统的开发得到了企业管理层的充分支持，并结合现有的管理制度进行设计，旨在提高日常工作效率和准确性。系统将通过自动化工具和数据库管理来改善现有的手工和纸质记录方式，减少错误和提升宠物领养效率。此外，已制定详细的实施计划和培训方案，确保各级管理人员和普通用户能够快速适应新系统。

综上所述，本项目在经济、技术和管理各方面均表现出积极的可行性，具备了成功开发和实施的所有必要条件。我们相信，宠物领养系统将有效地解决当前宠物领养中存在的问题，并为社会带来积极的价值。

## 2.2 功能需求

宠物领养系统主要流程包括基础数据管理、宠物管理、申请领养管理、领养审核管理等。

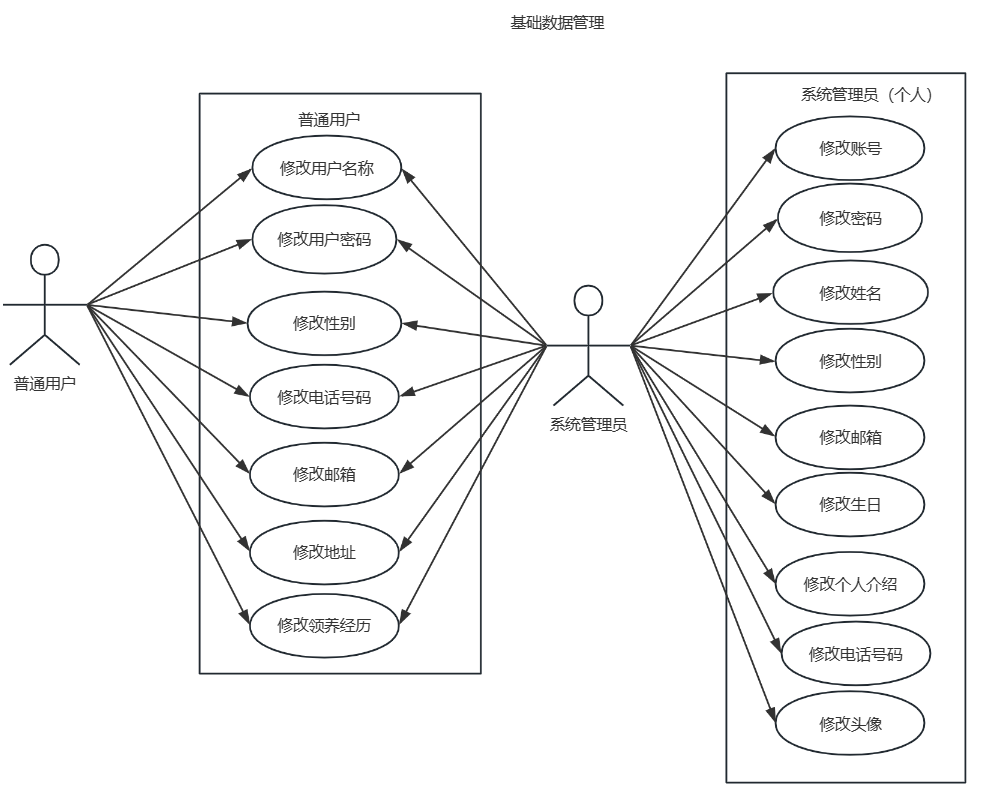
宠物领养系统中基础数据管理、宠物管理、申请领养管理、领养审核管理等模块之间相互关系如图2-1所示。



**图2-1系统各个模块之间的联系**

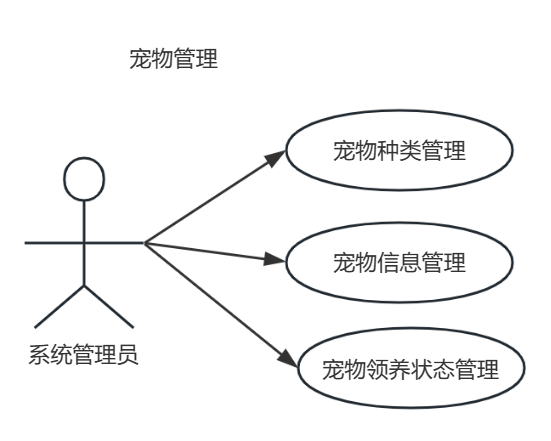
该系统分为三个角色：系统管理员、普通用户。系统管理员和普通用户都只能对自己的账号信息进行修改，系统管理员无法修改其他系统管理员信息，系统管理员可以管理普通用户。从图中可以看出，普通用户可以对自己的基础数据进行管理，并且对宠物进行领养申请，系统管理员可以对自己的基础数据及普通用户进行管理以及管理普通用户领养宠物的申请进行审核。

(1)基础数据管理。基础数据管理功能是对系统管理员和普通用户的基础信息进行管理,用例如图2-2所示。



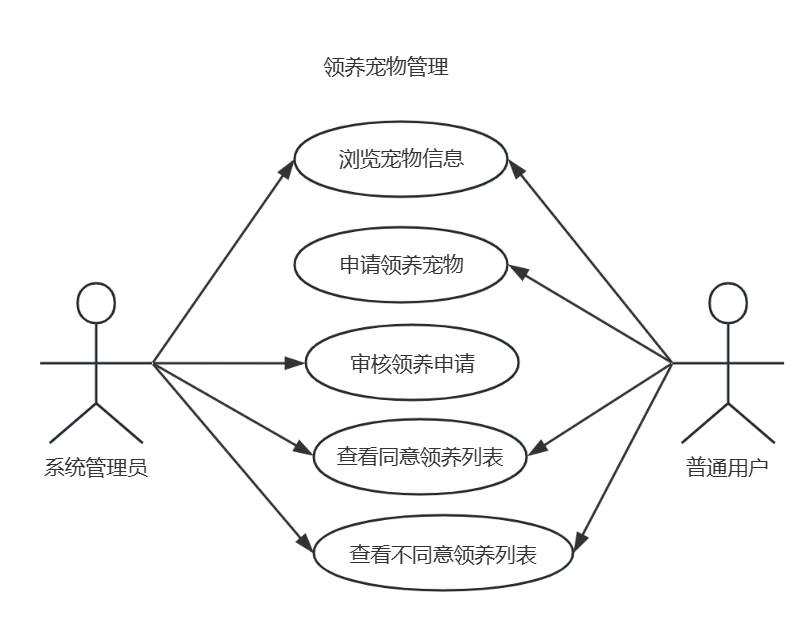
**图2-2基础数据管理用例图**

(2) 宠物管理。宠物管理是本系统的基础模块之一，主要包括宠物种类管理、宠物信息管理、宠物领养状态管理，用例如图2-3所示。



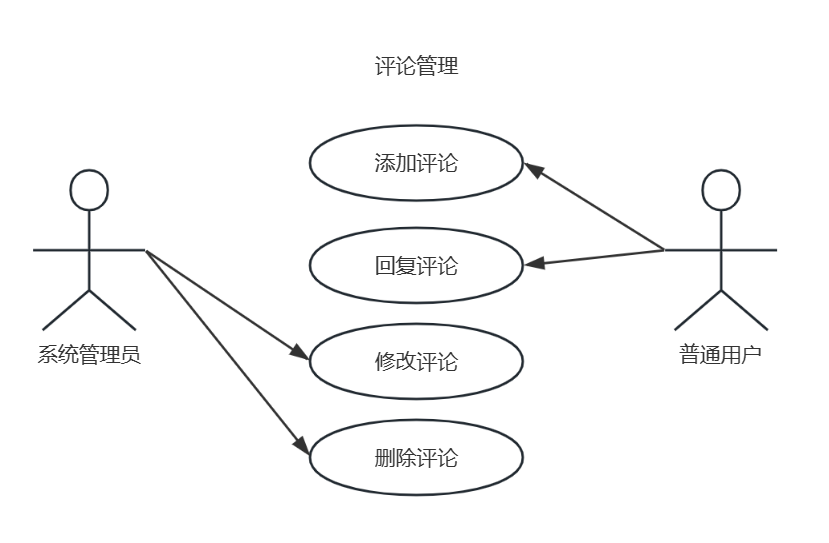
**图2-3宠物管理用例图**

(3) 领养宠物管理。领养宠物管理是宠物领养系统的核心模块，该模块主要包括的功能有浏览宠物信息、申请领养宠物、审核领养申请、查看同意领养列表、查看不同意领养列表，用例如图2-4所示。



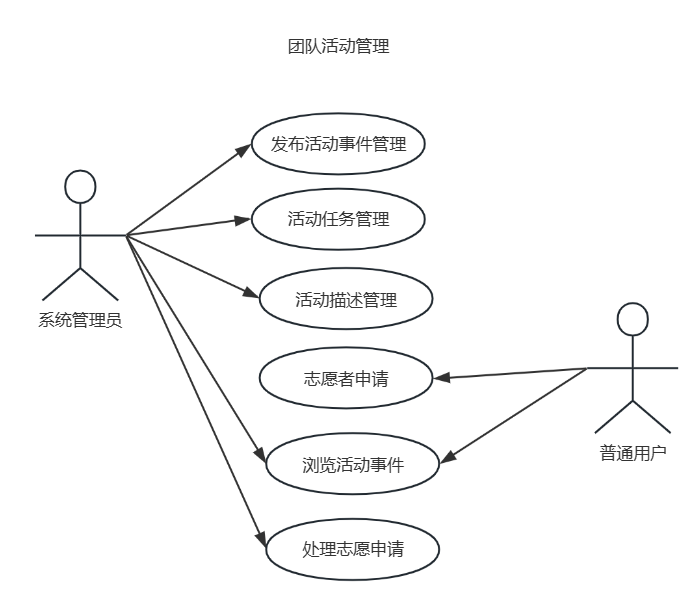
**图2-4领养宠物用例图**

(4)评论管理。该模块包括对展示的宠物信息添加评论、回复评论、删除评论等功能，用例如图2-5所示。



**图2-5评论管理用例图**

(5)团队活动管理。该模块包括发布活动事件管理、活动人物管理、活动描述管理、志愿者申请、处理志愿申请等功能，用例如图2-6所示。



**图2-6团队活动用例图**

## 2.3 用例描述

1.登录功能用例描述

**表2-1 登录功能用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE01 |
| **用例名称** | 前端登录功能 |
| **用例描述** | 普通用户输入账号和密码进行登录 |
| **前置条件** | 普通用户的账号和密码存在 |
| **后置条件** | 普通用户的账号和密码必须正确 |
| **活动步骤** | 1. 进入系统首页。 2. 点击【登录/注册】 按钮。 3. 输入账号、密码。 4. 点击【登录】按钮。 5. 登录成功，页面右上出现头像信息，按钮变为【退出】。 |
| **扩展点** | 如果账号、密码输入错误，提示登录失败。 |
| **数据格式** | 用户信息（用户名、密码） |

2.修改手机号码用例描述

**表2-2 修改手机号码用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE02 |
| **用例名称** | 修改手机号码 |
| **用例描述** | 输入手机号码点击保存 |
| **前置条件** | 手机号码格式正确 |
| **后置条件** | 确认手机号码是否保存成功 |
| **活动步骤** | 1. 点击个人信息——进入个人信息页面 2. 手机号码——正确手机号码 3. 保存完点击【确认】 4. 跳转页面，个人信息 |
| **扩展点** | 手机号码修改错误提示请重新修改手机号码 |
| **数据格式** | 手机号码（手机号码） |

3.新增资产类别用例描述

**表2-3新增资产类别用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE03 |
| **用例名称** | 新增资产类别 |
| **用例描述** | 点击新增资产类别进行新增 |
| **前置条件** | 新增资产类别页面正确显示 |
| **后置条件** | 确认资产类别信息是否新增正确 |
| **活动步骤** | 1. 点击资产类别——新增 2. 进入新增资产类别页面后 3. 新增完点击【确认】 4. 跳转页面，资产类别信息列表 |
| **扩展点** | 是否已经新增 |
| **数据格式** | 资产类别信息（类别编码、类别名称） |

4.新增品牌用例描述

**表2-4新增品牌用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE04 |
| **用例名称** | 新增品牌 |
| **用例描述** | 点击新增品牌进行新增 |
| **前置条件** | 新增品牌页面正确显示 |
| **后置条件** | 确认品牌信息是否新增正确 |
| **活动步骤** | 1. 点击品牌——新增 2. 进入新增品牌页面后 3. 新增完点击【确认】 4. 跳转页面，品牌信息列表 |
| **扩展点** | 是否已经新增 |
| **数据格式** | 品牌信息（品牌名称、品牌说明） |

5.资产入库登记用例描述

**表2-5 资产入库登记用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE05 |
| **用例名称** | 资产入库登记 |
| **用例描述** | 新增资产 |
| **前置条件** | 确认是否资产各项属性是否已添加 |
| **后置条件** | 该资产信息是否在管理列表中展示 |
| **活动步骤** | 1. 点击资产入库功能菜单——进入资产入库管理列表页面 2. 点击资产入库登记按钮 3. 进行资产入库登记页面 4. 保存完点击【确认】 5. 跳转页面，资产入库信息列表 |
| **扩展点** | 资产新增是否成功 |
| **数据格式** | 资产（资产名称、资产编码、资产类别、供应商、取得方式、入库时间、存放地点、设备用途、资产图片） |

6.资产报废登记用例描述

**表2-6 资产报废登记用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE06 |
| **用例名称** | 资产报废登记 |
| **用例描述** | 资产报废登记 |
| **前置条件** | 确认资产是否添加成功 |
| **后置条件** | 所提交的报废登记是否在列表中显示 |
| **活动步骤** | 1. 点击资产报废功能菜单——进入资产报废管理列表页面 2. 点击资产报废登记按钮 3. 进行资产报废登记页面 4. 提交完点击【确认】 5. 跳转页面，资产报废信息列表 |
| **扩展点** | 资产报废是否登记成功 |
| **数据格式** | 资产报废登记（资产名称、申请人、报废方式、报废日期、报废原因） |

7.资产报废审批用例描述

**表2-7 资产报废管理用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE07 |
| **用例名称** | 资产报废审批 |
| **用例描述** | 资产报废 |
| **前置条件** | 确认资产报废登记是否提交成功 |
| **后置条件** | 所报废的资产是否报废成功 |
| **活动步骤** | 1. 点击资产报废审批功能菜单——进入资产报废审批管理列表页面 2. 点击资产报废审批 3. 弹出资产报废审批弹窗 4. 点击通过/不通过 5. 报废完点击【确认】 6. 跳转页面，资产报废审批信息列表 |
| **扩展点** | 资产报废登记是否审批成功 |
| **数据格式** | 资产报废登记审批（审批结果、审批不通过原因） |

8.新增盘点单用例描述

**表2-8 新增盘点单用例描述**

|  |  |
| --- | --- |
| **用例编号** | CASE08 |
| **用例名称** | 新增盘点单 |
| **用例描述** | 新增盘点单 |
| **前置条件** | 资产盘点页面是否正确显示 |
| **后置条件** | 新增盘点单是否在列表中显示 |
| **活动步骤** | 1. 点击资产盘点——进入资产盘点管理页面 2. 点击新增盘点单按钮——进入新增盘点单页面 3. 输入相应内容选择盘点资产点击【保存】 4. 跳转页面，角色信息列表 |
| **扩展点** | 资产盘点单是否新增成功 |
| **数据格式** | 资产盘点单（盘点单名称、盘点说明、盘点资产） |

## 2.4 性能需求

非功能性需求包括最大并发数，页面响应时间，安全性、稳定性、可扩展性等，结合系统的建设目标，科研成果的性能要求如下。

（1）用户并发数。一般公司现有在职员工300左右，系统访问量较大时间应该星期一上班时系统访问频繁，服务器负载重。平时主要是管理员访问，经过统计用户并发数为50左右，其中查询统计操作是比较耗时的操作，并且使用频率较高，可以通过负载均衡技术和优化数据库结构来提高系统运行效率。

（2）响应时间。响应时间是指访问系统时相关页面的平均响应时间，页面响应时间越短，用户体验越好，一般来说页面内响应时间在0.5秒以下用户操作体验非常好，在0.5至2秒用户操作体检较好，在2至5秒之间用户操作体验一般，在5至8秒之间用户操作体验较差，在8秒以上以上用户操作体验非常差。为了提高系统的用户体验，宠物领养系统的平均页面响应时间应在2秒以内，查询统计页面的页面响应时间应该在6秒以内。

（3）安全性。安全性对宠物领养系统很重要，为提高系统的安全性，所有访问本系统的用户都要使用自身账号登录，包括具体的功能操作。系统访问用户只能是在职人员，其他用户不能访问宠物领养系统。同时，系统应设置自动数据备份功能，及时对系统数据库进行备份，防止因软、硬件引起的错误而丢失数据。

# 3 系统设计

## 3.1 设计思想

结合公司整体数字化企业建设要求，建设企业宠物领养系统的主要目标包括：

（1）根据企业信息化建设规范，要求宠物领养系统开发使用B/S结构，JAVA语言开发，能实现与其他应用系统的数据共享和信息交换，访问页面友好。

（2）新开发的宠物领养系统要有良好的可扩展性，能够适应管理模式的变革，适应相关部门的调整及公司结构的变化。

（3）新开发的宠物领养系统要有良好的安全性，宠物领养系统的设计引入了基于角色的权限管理模型，实现了用户和权限的分离，能有效保障系统的安全性；同时由于采用B/S结构，实现了应用服务器与数据库服务器的有效分类，在某种程度上也提高了系统的安全性。

## 3.2 系统架构设计

宠物领养系统使用 MVC 设计模式进行设计，设置较为灵活的用户页面，提高用户体验效果。

本网站采用的是 SSM 框架，第一个 S 代表 Spring，Spring 是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架，第二个 S 代表 Spring MVC，Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色。M 代表 MyBatis， MyBatis 使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的 POJO（Plain Old Java Objects，普通的 Java 对象）映射成数据库中的记录。

MVC 设计模式包括业务模型、用户界面、控制器三层，MVC 是一种使用 MVC(Model View Controller 模型-视图-控制器)设计创建 Web 应用程序的模式:

vue实现视图，显示数据；

com.hngc.eam.domain存放实体类；

com.hngc.eam.controller 实现控制器，处理写入数据库记录；

com.hngc.eam.dao存放数据操作接口；

com.hngc.eam.service 存放业务逻辑接口；

com.hngc.eam.bean 和 com.hngc.eam.controller 和 com.hngc.eam.service 为实现全部数据操作为模型层。

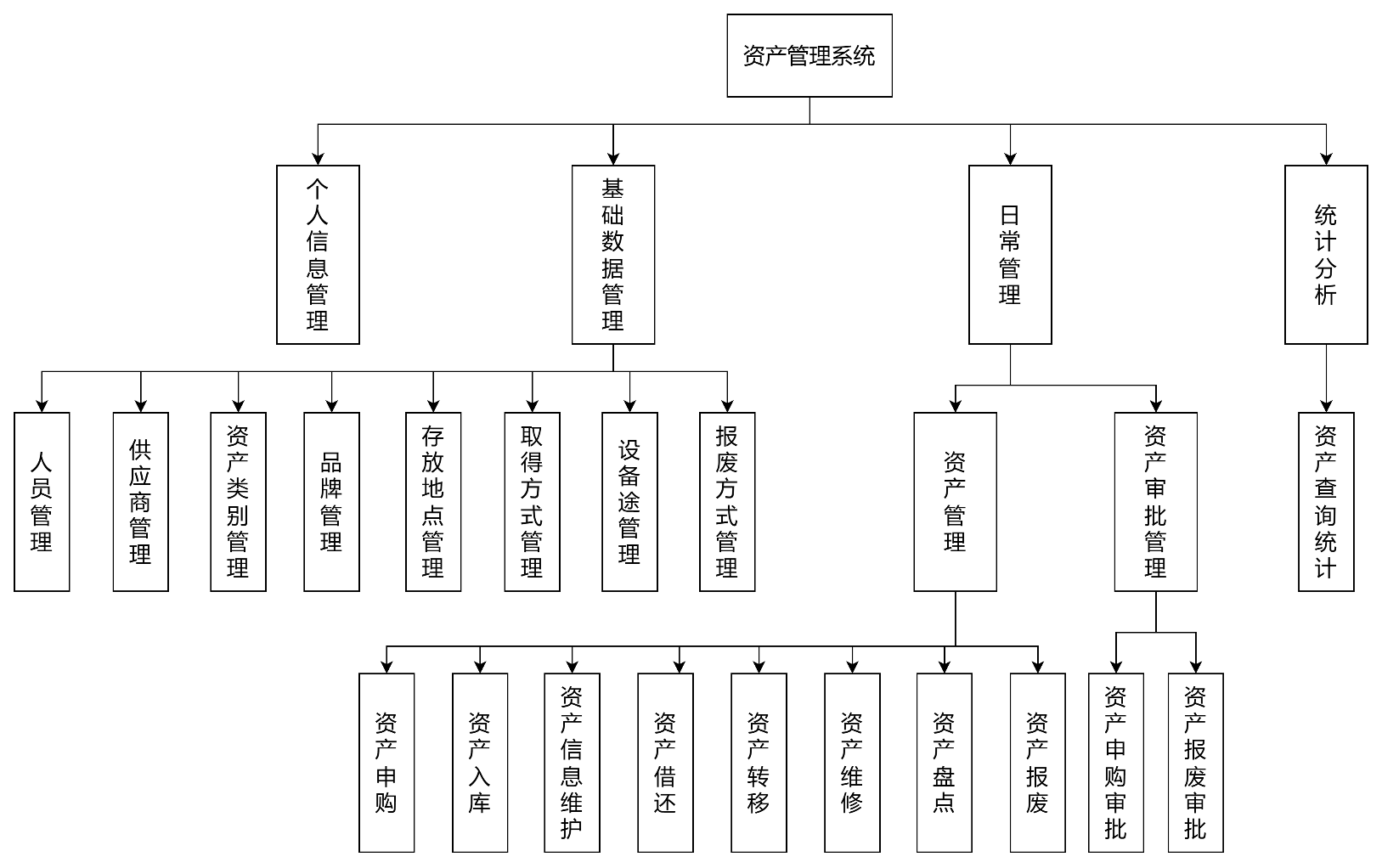
**图片包含 图形用户界面

描述已自动生成**

**图 3-1 宠物领养系统架构图**

## 3.3 功能模块设计

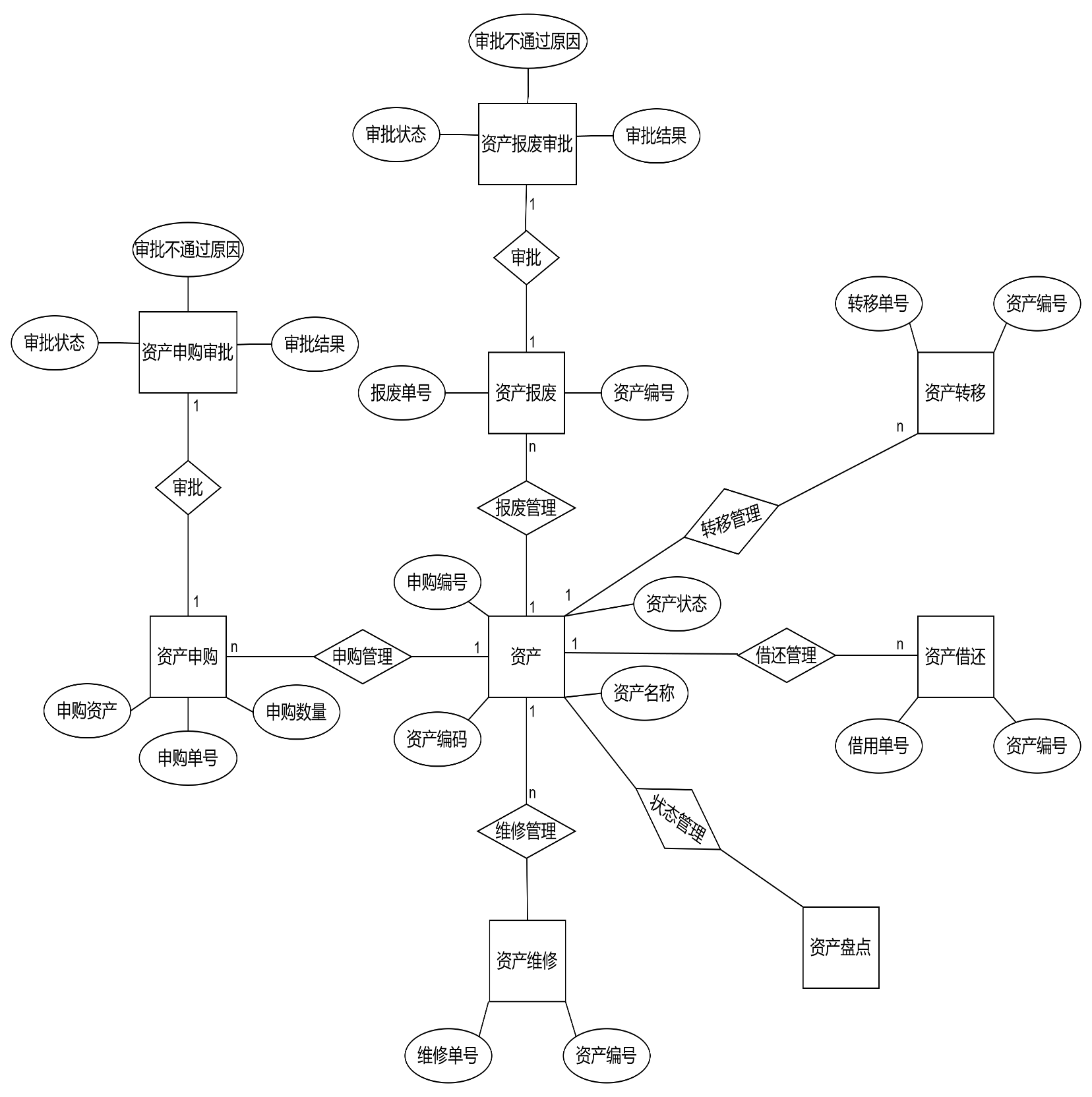
本系统根据功能划分，主要包含系统管理员、宠物领养员、资产领导等3个功能模块，系统功能模块划分如图3-2所示。

**图3-2 系统功能模块结构图**

## 3.4 数据库设计

### 3.4.1 概念结构设计

宠物领养系统包含取得方式、设备用途、申购审批、报废审批、资产借还、品牌表、资产盘点、盘点资产、资产盘点中间、资产类别、部门、员工、菜单、存放地点、资产入库、资产申购、资产维修、权限、角色、角色权限、资产报废、报废方式、资产转移、供应商、用户等实体类，每个实体包含各自属性，实体间具有不同联系，可相互间进行操作，系统主要E-R图如图3-3所示。



**图3-3 E-R图**

### 3.4.2 逻辑结构设计

宠物领养系统主要包括取得方式表、设备用途表、申购审批表、报废审批表、资产借还表、品牌表表、资产盘点表、盘点资产表、资产盘点中间表、资产类别表、部门表、员工表、菜单表、存放地点表、资产入库表、资产申购表、资产维修表、权限表、角色表、角色权限表、资产报废表、报废方式表、资产转移表、供应商表、用户等等。

（1）取得方式表（acquired表）

取得方式表主要用于存放资产的取得方式信息，它的主要信息项有取得方式编号、取得方式编码、取得方式名称、状态、创建时间，该表结构如表3-1所示。

**表3-1取得方式表**

| 表名 | acquired | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| acquiredId | int (10) | 非空 | PRIMARY KEY | 取得方式编号 |
| acquiredCoding | varchar(30) | 非空 |  | 取得方式编码 |
| acquiredName | varchar(30) | 非空 |  | 取得方式名称 |
| acquiredStatus | int(10) | 非空 |  | 状态 |
| acquiredCreateDate | datetime | 非空 |  | 创建时间 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（2）设备用途表（application表）

设备用途表主要用于存放资产的设备用途信息，它的主要信息项有设备用途编号、设备用途编码、设备用途名称、状态、创建时间、备注，该表结构如表3-2所示。

**表3-2设备用途表**

| 表名 | application | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| applicationId | int (10) | 非空 | PRIMARY KEY | 设备用途编号 |
| applicationCoding | varchar(30) | 非空 |  | 设备用途编码 |
| applicationName | varchar(30) | 非空 |  | 设备用途名称 |
| applicationStatus | int(10) | 非空 |  | 状态 |
| applicationCreateDate | datetime | 非空 |  | 创建时间 |
| remarks | varchar(255) | 非空 |  | 备注 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（3）申购审批表（approvepurchase表）

申购审批表主要用于存放资产申购的审批信息，它的主要信息项有申购审批编号、审批状态、审批时间、审批结果、审批不通过原因、用来获取主键的标志，该表结构如表3-3所示。

**表3-3申购审批表**

| 表名 | approvePurchase | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| approvePurchaseId | int (10) | 非空 | PRIMARY KEY | 申购审批编号 |
| approvePurchaseStatus | int (10) | 非空 |  | 审批状态 |
| approvePurchaseDate | datetime(19) | 非空 |  | 审批时间 |
| approvePurchaseResult | int(10) | 非空 |  | 审批结果 |
| approvePurchaseReason | varchar(200) | 非空 |  | 审批不通过原因 |
| approvePurchaseFlag | varchar(255) | 非空 |  | 用来获取主键的标志 |
| 补充说明 | 审批状态，0为未审批，1为已审批；审批结果，1为不通过，2为已通过 | | | |

（4）报废审批表（approvepurchase表）

报废审批表主要用于存放资产报废的审批信息，它的主要信息项有报废审批编号、审批状态、审批时间、审批结果、审批不通过原因、用来获取主键的标志，该表结构如表3-4所示。

**表3-4报废审批表**

| 表名 | approveScrap | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| approveScrapId | int (10) | 非空 | PRIMARY KEY | 报废审批编号 |
| approveScrapStatus | int (10) | 非空 |  | 审批状态 |
| approveScrapDate | datetime(19) | 非空 |  | 审批时间 |
| approveScrapResult | int(10) | 非空 |  | 审批结果 |
| approveScrapReason | varchar(200) | 非空 |  | 审批不通过原因 |
| approveScrapFlag | varchar(255) | 非空 |  | 用来获取主键的标志 |
| 补充说明 | 审批状态，0为未审批，1为已审批；审批结果，1为不通过，2为已通过 | | | |

（5）资产借还表（borrow表）

资产借还表主要用于存放资产借还信息，它的主要信息项有借用编号、借用单号、资产编号、借用日期、登记时间、人员编号、借用原因、归还日期、归还状态、备注，该表结构如表3-5所示。

**表3-5资产借还表**

| 表名 | borrow | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| borrowId | int (10) | 非空 | PRIMARY KEY | 借用编号 |
| borrowingNumber | varchar (50) | 非空 |  | 借用单号 |
| propertyId | int(10) | 非空 |  | 资产编号 |
| borrowingDate | datetime(19) | 非空 |  | 借用日期 |
| borrowingTime | datetime(19) | 非空 |  | 登记时间 |
| emoId | int(10) | 非空 |  | 人员编号 |
| borrowingReason | varchar(255) | 非空 |  | 借用原因 |
| returnDate | datetime(19) | 非空 |  | 归还日期 |
| returnStatus | int(10) | 非空 |  | 归还状态 |
| remarks | varchar(255) | 非空 |  | 备注 |
| 补充说明 | 归还状态，0为未归还，1.为已归还 | | | |

（6）品牌表（brand表）

品牌表主要用于存放资产的品牌信息，它的主要信息项有品牌编号、品牌编码、品牌名称、状态、创建时间、备注，该表结构如表3-6所示。

**表3-6品牌表**

| 表名 | brand | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| applicationId | int (10) | 非空 | PRIMARY KEY | 品牌编号 |
| applicationCoding | varchar(30) | 非空 |  | 品牌编码 |
| applicationName | varchar(30) | 非空 |  | 品牌名称 |
| applicationStatus | int(10) | 非空 |  | 状态 |
| applicationCreateDate | datetime | 非空 |  | 创建时间 |
| remarks | varchar(255) | 非空 |  | 备注 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（7）盘点表（check表）

盘点表主要用于存放资产盘点单信息，它的主要信息项有盘点编号、盘点单号、盘点单名称、盘点说明、资产数量、盘点单状态、盘点开始日期、盘点结束日期、创建日期，该表结构如表3-6所示。

**表3-7盘点表**

| 表名 | check | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| checkId | int (10) | 非空 | PRIMARY KEY | 盘点编号 |
| checkNumber | varchar(50) | 非空 |  | 盘点单号 |
| checkName | varchar(50) | 非空 |  | 盘点单名称 |
| checkExplanation | varchar(50) | 非空 |  | 盘点说明 |
| proNum | int (10) | 非空 |  | 资产数量 |
| checkStatus | int (10)) | 非空 |  | 盘点单状态 |
| checkStartDate | datetime(19) | 非空 |  | 盘点开始日期 |
| checkEndDate | datetime(19) | 非空 |  | 盘点结束日期 |
| checkCreateDate | datetime(19) | 非空 |  | 创建日期 |
| 补充说明 | 盘点单状态，0为未开始，1为进行中，2为已结束 | | | |

（8）盘点资产表（checkIn表）

盘点资产表主要用于存放资产盘点单信息，它的主要信息项有编号、资产编号、盘点结果、盘点备注、盘点资产状态、盘点单号，该表结构如表3-8所示。

**表3-8盘点资产表**

| 表名 | check | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| checkInId | int (10) | 非空 | PRIMARY KEY | 编号 |
| propertyId | int (10) | 非空 |  | 资产编号 |
| checkInResult | int (10) | 非空 |  | 盘点结果 |
| checkInMemo | varchar(50) | 非空 |  | 盘点备注 |
| checkInStatus | int (10) | 非空 |  | 盘点资产状态 |
| checkId | int (10)) | 非空 |  | 盘点单号 |
| 补充说明 | 盘点结果，1为正常，2为已报废；盘点资产状态，0为未盘点，1为已盘点 | | | |

（9）盘点中间表（checkMiddle表）

盘点中间表主要用于存放资产盘点单信息，它的主要信息项有编号、资产编号，该表结构如表3-8所示。

**表3-9盘点资产表**

| 表名 | checkMiddle | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| checkMiddleId | int (10) | 非空 | PRIMARY KEY | 编号 |
| propertyId | int (10) | 非空 |  | 资产编号 |
| 补充说明 |  | | | |

（10）资产类别表（classes表）

资产类别表主要用于存放资产的资产类别信息，它的主要信息项有资产类别编号、资产类别编码、资产类别名称、状态、创建时间、备注，该表结构如表3-10所示。

**表3-10资产类别表**

| 表名 | classes | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| classId | int (10) | 非空 | PRIMARY KEY | 资产类别编号 |
| classCoding | varchar(30) | 非空 |  | 资产类别编码 |
| className | varchar(30) | 非空 |  | 资产类别名称 |
| classStatus | int(10) | 非空 |  | 状态 |
| classCreateDate | datetime | 非空 |  | 创建时间 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（10）部门表（department表）

部门表主要用于存放人员的部门信息，它的主要信息项有部门编号、部门编码、部门名称、状态、创建时间、备注，该表结构如表3-10所示。

**表3-10部门表**

| 表名 | department | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| departmentId | int (10) | 非空 | PRIMARY KEY | 部门编号 |
| departmentCoding | varchar(30) | 非空 |  | 部门编码 |
| departmentName | varchar(30) | 非空 |  | 部门名称 |
| departmentInfo | int(10) | 非空 |  | 状态 |
| departmentCreateDate | datetime | 非空 |  | 创建时间 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（10）员工表（employee表）

员工表主要用于存放员工信息，它的主要信息项有编号、姓名、工号、性别、生日、所属部门编号、状态、照片，该表结构如表3-10所示。

**表3-10部门表**

| 表名 | employee | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| emoId | int (10) | 非空 | PRIMARY KEY | 编号 |
| name | varchar(20) | 非空 |  | 姓名 |
| emoNo | varchar(50) | 非空 |  | 工号 |
| sex | int(10) | 非空 |  | 性别 |
| birthday | datetime | 非空 |  | 生日 |
| departmentId | int (10) | 非空 |  | 所属部门编号 |
| emoTypeId | int (10) | 非空 |  | 状态 |
| emoImage | varchar(255) | 非空 |  | 照片 |
| 补充说明 | 状态，0为在编、1为合同、2为临时、3为删除 | | | |

（11）菜单表（menus表）

该表存放前端页面的菜单信息，与权限表相辅相成，该表结构如表3-3所示。

**表3-11菜单信息表 menu表**

| 表名 | Menu | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| menu\_id | bigint(20) | 非空 | Primary key | 菜单编号 |
| menu\_name | varchar(50) | 非空 |  | 菜单名称 |
| menu\_url | varchar(200) | 空 |  | 菜单URL |
| menu\_icon | varchar(50) | 空 |  | 图标 |
| rid | int(10) | 空 |  | 对应权限id |
| 补充说明 |  | | | |

（12）存放地点表（place表）

存放地点表主要用于存放资产的存放地点信息，它的主要信息项有存放地点编号、存放地点名称、存放地点类型、状态、备注、创建时间，该表结构如表3-12所示。

**表3-12存放地点表**

| 表名 | place | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| placeId | int (10) | 非空 | PRIMARY KEY | 存放地点编号 |
| placeName | varchar(60) | 非空 |  | 存放地点名称 |
| placeType | varchar(50) | 非空 |  | 存放地点类型 |
| placeStatus | int(10) | 非空 |  | 状态 |
| placeRemarks | varchar(30) | 非空 |  | 备注 |
| placeCreateDate | datetime(19) | 非空 |  | 创建时间 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（13）资产入库表（property表）

资产入库表主要用于存放资产信息，它的主要信息项有资产编号、资产编码、资产名称、资产类别、供应商编号、品牌编号、取得方式编号、申购编号、入库时间、登记时间、资产入库编号、设备用途编号、资产图片、资产状态，该表结构如表3-13所示。

**表3-13资产入库表**

| 表名 | property | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| propertyId | int(10) | 非空 | PRIMARY KEY | 资产编号 |
| propertyCoding | varchar(50) | 非空 |  | 资产编码 |
| propertyName | varchar(50) | 非空 |  | 资产名称 |
| classId | int(10) | 非空 |  | 资产类别 |
| supplierId | int(10) | 非空 |  | 供应商编号 |
| brandId | int(10) | 非空 |  | 品牌编号 |
| acquiredId | int(10) | 非空 |  | 取得方式编号 |
| purchaseId | int(10) | 非空 |  | 申购编号 |
| stockingDate | datetime(19) | 非空 |  | 入库时间 |
| propertyTime | datetime(19) | 非空 |  | 登记时间 |
| placeId | int(10) | 非空 |  | 资产入库编号 |
| applicationId | int(10) | 非空 |  | 设备用途编号 |
| propertyImage | varchar(50) | 非空 |  | 资产图片 |
| propertyStatus | int(10) | 非空 |  | 资产状态 |
| 补充说明 | 资产状态，固定为“正常”，0为正常，1为已报废，2为借用中; 申购编号，如果是取得方式为购买的话，需填写的申购编号 | | | |

（14）资产申购表（purchase表）

资产申购表主要用于存放资产信息，它的主要信息项有申购编号、申购单号、申购资产、资产类别、设备用途编码、预计价格、申购数量、申请人编号、申购建议、申请时间、登记时间、申购理由、提交状态、申购审批编号，该表结构如表3-14所示。

**表3-14资产申购表**

| 表名 | purchase | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| purchaseId | int(10) | 非空 | PRIMARY KEY | 申购编号 |
| purchaseNumber | varchar(50) | 非空 |  | 申购单号 |
| propertyName | varchar(50) | 非空 |  | 申购资产 |
| classId | int(10) | 非空 |  | 资产类别 |
| applicationId | int(10) | 非空 |  | 设备用途编码 |
| rmb | double(23) | 非空 |  | 预计价格 |
| purchaseCount | int(10) | 非空 |  | 申购数量 |
| emoId | int(10) | 非空 |  | 申请人编号 |
| purchaseAdvice | varchar(50) | 非空 |  | 申购建议 |
| applicationDate | datetime(19) | 非空 |  | 申请时间 |
| purchaseTime | datetime(19) | 非空 |  | 登记时间 |
| purchaseReason | varchar(200) | 非空 |  | 申购理由 |
| commitStatus | int(10) | 非空 |  | 提交状态 |
| approvePurchaseId | int(10) | 非空 |  | 申购审批编号 |
| 补充说明 | 提交状态，0为未提交，1为已提交 | | | |

（15）资产维修表（purchase表）

资产维修表主要用于存放资产信息，它的主要信息项有维修编号、维修单号、资产编号、报修日期、登记时间，系统当前时间、报修人、报修原因、维修单位、修复日期、维修费用、故障说明，该表结构如表3-15所示。

**表3-15资产维修表**

| 表名 | purchase | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| repairId | int(10) | 非空 | PRIMARY KEY | 维修编号 |
| repairNumber | varchar(50) | 非空 |  | 维修单号 |
| propertyId | int(10) | 非空 |  | 资产编号 |
| repairDate | datetime(19) | 非空 |  | 报修日期 |
| repairTime | datetime(19) | 非空 |  | 登记时间 |
| emoId | int(10) | 非空 |  | 报修人 |
| repairReason | varchar(200) | 非空 |  | 报修原因 |
| mro | varchar(200) | 非空 |  | 维修单位 |
| restorationDate | datetime(19) | 非空 |  | 修复日期 |
| repairCost | float(13) | 非空 |  | 维修费用 |
| explanation | varchar(255) | 非空 |  | 故障说明 |
| 补充说明 | 登记时间，系统当前时间 | | | |

（16）权限表（right表）

该表存放权限信息，与角色表相辅相成，该表结构如表3-16所示。

**表3-16权限表**

| 表名 | right | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| right\_id | int(10) | 非空 | PRIMARY KEY | 权限编号 |
| right\_parent\_id | int(10) | 非空 |  | 模块父编号 |
| right\_type | int(10) | 空 |  | 权限类型 |
| right\_text | varchar(100) | 空 |  | 权限名称 |
| right\_url | varchar(50) | 空 |  | 接口网址 |
| 补充说明 | 权限类型，0.菜单级别 1.按钮级别; 接口网址,权限对应接口的访问url | | | |

（17）角色表（role表）

该表存放角色信息，与权限表相辅相成，该表结构如表3-17所示。

**表3-17角色表**

| 表名 | role | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| role\_id | int(10) | 非空 | PRIMARY KEY | 角色编号 |
| role\_name | varchar(50) | 非空 |  | 角色名称 |
| role\_desc | varchar(50) | 空 |  | 角色备注 |
| 补充说明 |  | | | |

（18）角色权限表（roleRight表）

角色权限表，该表的作用是在菜单和角色间起一个承接的作用，该表结构如表3-4所示。

**表3-4角色权限表**

| 表名 | roleRight | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| rl\_id | int(10) | 非空 | PRIMARY KEY | 编号 |
| rl\_role\_id | int(10) | 非空 |  | 角色编号 |
| rl\_right\_id | int(10) | 非空 |  | 权限编号 |
| 补充说明 |  | | | |

（19）资产报废表（scrap表）

资产报废表主要用于存放资产信息，它的主要信息项有报废编号、报废单号、资产编号、申请人、申请时间、登记时间、报废方式编码、报废原因、提交状态、报废审批编号，该表结构如表3-19所示。

**表3-19资产报废表**

| 表名 | scrap | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| scrapId | int(10) | 非空 | PRIMARY KEY | 报废编号 |
| scrapNumber | varchar(50) | 非空 |  | 报废单号 |
| propertyId | int(10) | 非空 |  | 资产编号 |
| emoId | int(10) | 非空 |  | 申请人 |
| scrapDate | datetime(19) | 非空 |  | 申请时间 |
| scrapTime | datetime(19) | 非空 |  | 登记时间 |
| scrapWayId | int(10) | 非空 |  | 报废方式编码 |
| scrapReason | varchar(200) | 非空 |  | 报废原因 |
| commitScrapStatus | int(10) | 非空 |  | 提交状态 |
| approveScrapId | int(10) | 非空 |  | 报废审批编号 |
| 补充说明 | 提交状态，0为未提交，1为已提交 | | | |

（20）报废方式表（scrapWay表）

报废方式表主要用于存放资产的报废方式信息，它的主要信息项有报废方式编号、报废方式编码、报废方式名称、状态、创建时间、备注，该表结构如表3-20所示。

**表3-20报废方式表**

| 表名 | scrapWay | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| scrapWayId | int(10) | 非空 | PRIMARY KEY | 报废方式编号 |
| scrapWayCoding | varchar(30) | 非空 |  | 报废方式编码 |
| scrapWayName | varchar(30) | 非空 |  | 报废方式名称 |
| scrapWayStatus | int(10) | 非空 |  | 状态 |
| scrapWayCreateDate | datetime(19) | 非空 |  | 创建时间 |
| remarks | varchar(255) | 非空 |  | 备注 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（21）资产转移表（shift表）

资产转移表主要用于存放资产转移信息，它的主要信息项有转移编号、转移单号、资产编号、原使用人员编号、新使用人员编号、转移日期、登记时间、转移原因，该表结构如表3-21所示。

**表3-21资产转移表**

| 表名 | shift | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| shiftId | int(10) | 非空 | PRIMARY KEY | 转移编号 |
| shiftNumber | varchar(100) | 非空 |  | 转移单号 |
| propertyId | int(10) | 非空 |  | 资产编号 |
| oldId | int(10) | 非空 |  | 原使用人员编号 |
| newId | int(10) | 非空 |  | 新使用人员编号 |
| shiftDate | datetime(19) | 非空 |  | 转移日期 |
| shiftTime | datetime(19) | 非空 |  | 登记时间 |
| shiftReason | varchar(255) | 非空 |  | 转移原因 |
| 补充说明 |  | | | |

（22）供应商表（supplier表）

供应商表主要用于存放供应商信息，它的主要信息项有供应商编号、供应商名称、供应商类别、状态、联系人、移动电话、地址、创建时间、备注，该表结构如表3-22所示。

**表3-22供应商表**

| 表名 | supplier | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| supplierId | int(10) | 非空 | PRIMARY KEY | 供应商编号 |
| supplierName | varchar(50) | 非空 |  | 供应商名称 |
| supplierType | varchar(50) | 非空 |  | 供应商类别 |
| supplierStatus | int(10) | 非空 |  | 状态 |
| supplierlinkMan | varchar(50) | 非空 |  | 联系人 |
| supplierPhone | varchar(11) | 非空 |  | 移动电话 |
| supplierAddress | varchar(50) | 非空 |  | 地址 |
| supplierCreateDate | datetime(19) | 非空 |  | 创建时间 |
| remarks | varchar(255) | 非空 |  | 备注 |
| 补充说明 | 状态，0.已启用 1.已禁用 | | | |

（23）用户表（user表）

用户表主要用于存放用户信息，它的主要信息项有用户编号、姓名、用户名、密码、性别、手机号码、最后登录时间、登录次数、部门编号、角色编号，该表结构如表3-22所示。

**表3-23用户表**

| 表名 | user | | | |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 | 备注 |
| userId | int(10) | 非空 | PRIMARY KEY | 用户编号 |
| name | varchar(10) | 非空 |  | 姓名 |
| userName | varchar(20) | 非空 |  | 用户名 |
| password | varchar(255) | 非空 |  | 密码 |
| sex | int(10) | 非空 |  | 性别 |
| phone | varchar(11) | 非空 |  | 手机号码 |
| lastTime | datetime(19) | 非空 |  | 最后登录时间 |
| loginCount | int(10) | 非空 |  | 登录次数 |
| departmentId | int(10) | 非空 |  | 部门编号 |
| roleId | int(10) | 非空 |  | 角色编号 |
| 补充说明 | 性别，1为男，0为女 | | | |

### 3.4.2 数据库连接

|  |
| --- |
| spring:  datasource:  druid:  driver-class-name: com.mysql.cj.jdbc.Driver  url: jdbc:mysql://localhost:3306/eam?serverTimezone=Asia/Shanghai&useUnicode=true&characterEncoding=utf8&useSSL=false&allowPublicKeyRetrieval=true  username: root  password: 123456 |

## 3.5 详细设计

### 3.5.1 用户信息类图

用户信息的类图如下图3-4所示。各类具体说明如下：

(1) UserController类：UserController类主要完成用户信息的增删改查操作，位于该项目目录com.hngc.eam.controller处。

(2) User类：User类是用户基本信息类，位于该项目目录com.hngc.eam.domain处。

(3) IUserService接口：IUserService用户接口，位于该项目目录com.hngc eam.service处。

(4) UserServiceImpl类：UserServiceImpl类，位于该项目目录com.hngc eam.service.impl处

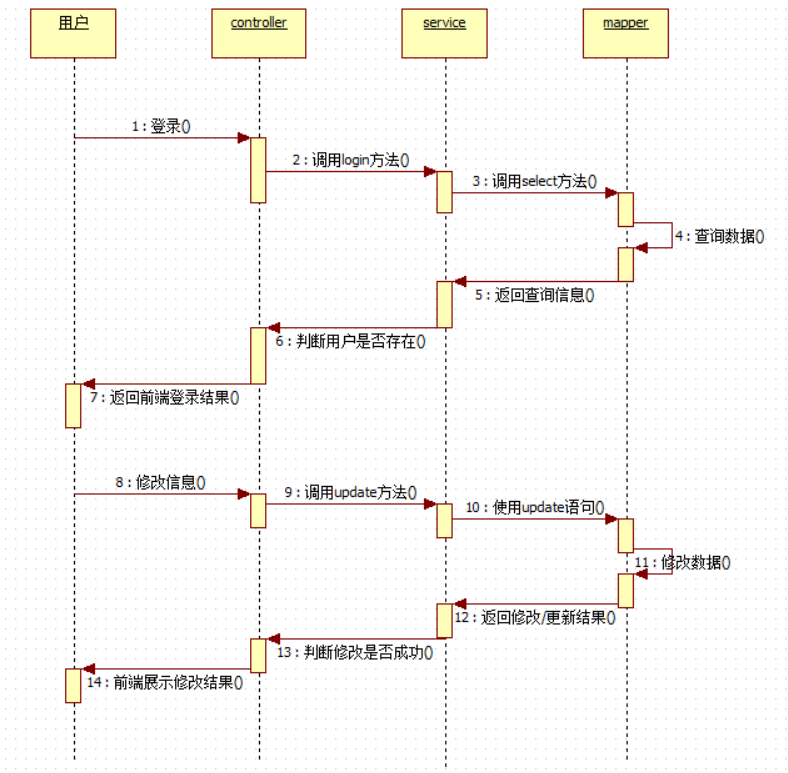
图片包含 图形用户界面

描述已自动生成(5) UserMapper文件：实现UserServiceImpl类中的一些方面

**图 3-****4 用户信息管理类图**

### 3.5.2 用户信息管理时序图

用户信息管理模块主要完成对登录修改用户信息功能,其操作流程如图3-5用户信息管理时序图所示:

**图 3-5用户信息管理时序图**

### 3.5.3 资产借还类图

资产借还的类图如下图3-6所示。各类具体说明如下：

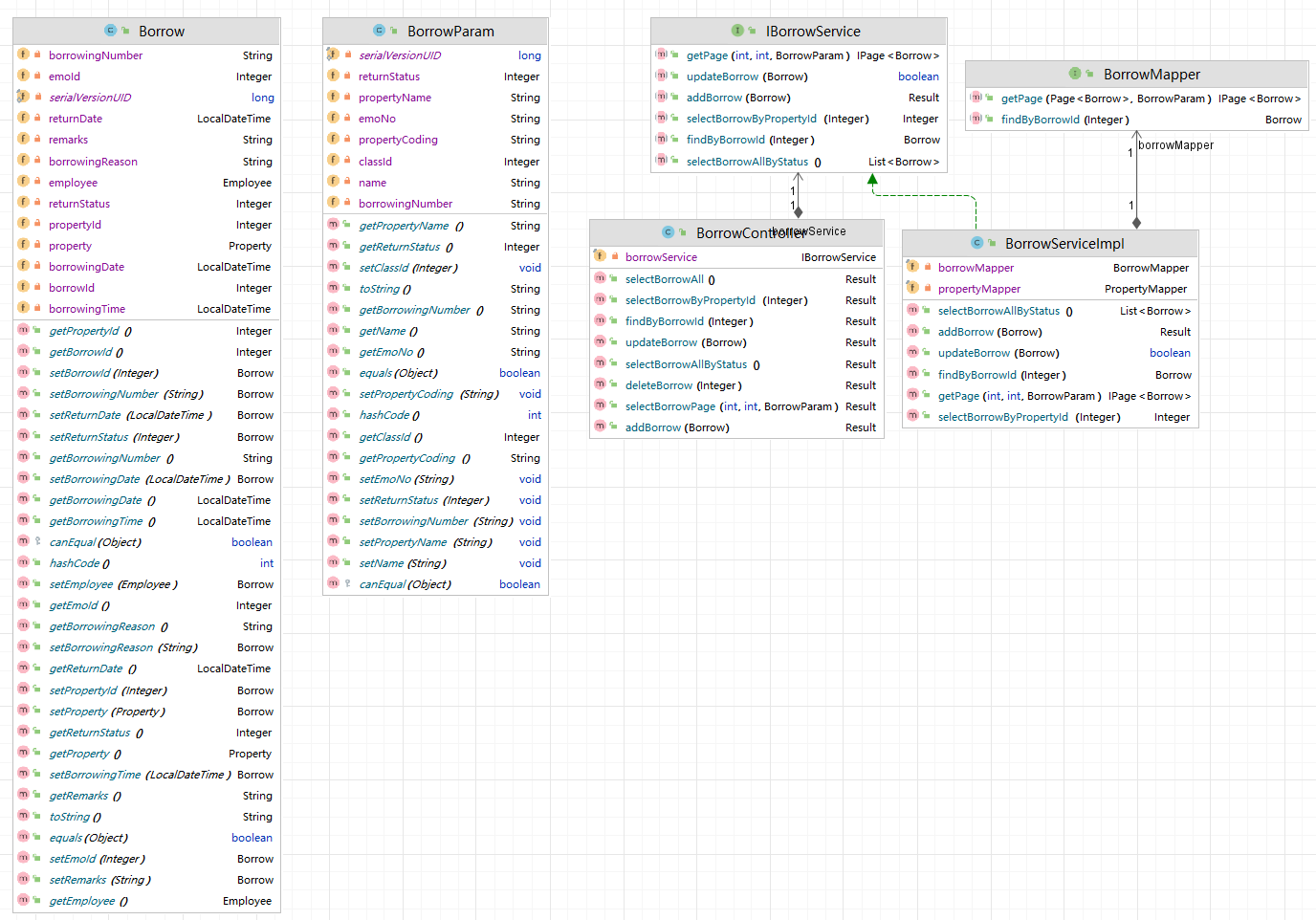
(1) BorrowController类：BorrowController类主要完成资产借还信息的增删改查操作，位于该项目目录com.hngc.eam.controller处。

(2) Borrow类：Borrow类是用户基本信息类，位于该项目目录com.hngc.eam.domain处。

(3) IBorrowService接口：IBorrowService用户接口，位于该项目目录com.hngc eam.service处。

(4) BorrowServiceImpl类：BorrowServiceImpl类，位于该项目目录com.hngc eam.service.impl处

(5) BorrowMapper文件：实现BorrowServiceImpl类中的一些方面

(6) BorrowParam类：接受前端的参数

**图 3-6 资产借还信息管理类图**

### 3.5.4 资产借还信息管理时序图

图示

描述已自动生成资产借还信息管理模块主要完成对资产的借用和归还,其操作流程如图3-7资产借还信息管理时序图所示:

**图 3-7资产借还信息管理时序图**

# 4 系统实现

## 4.1 用户登录

用户打开浏览器，在地址栏内输入宠物领养系统URL地址，按回车键，显示用户登录页面，在登录页面中输入账号、密码及实现登录操作，具体登录页面如图4-1所示。



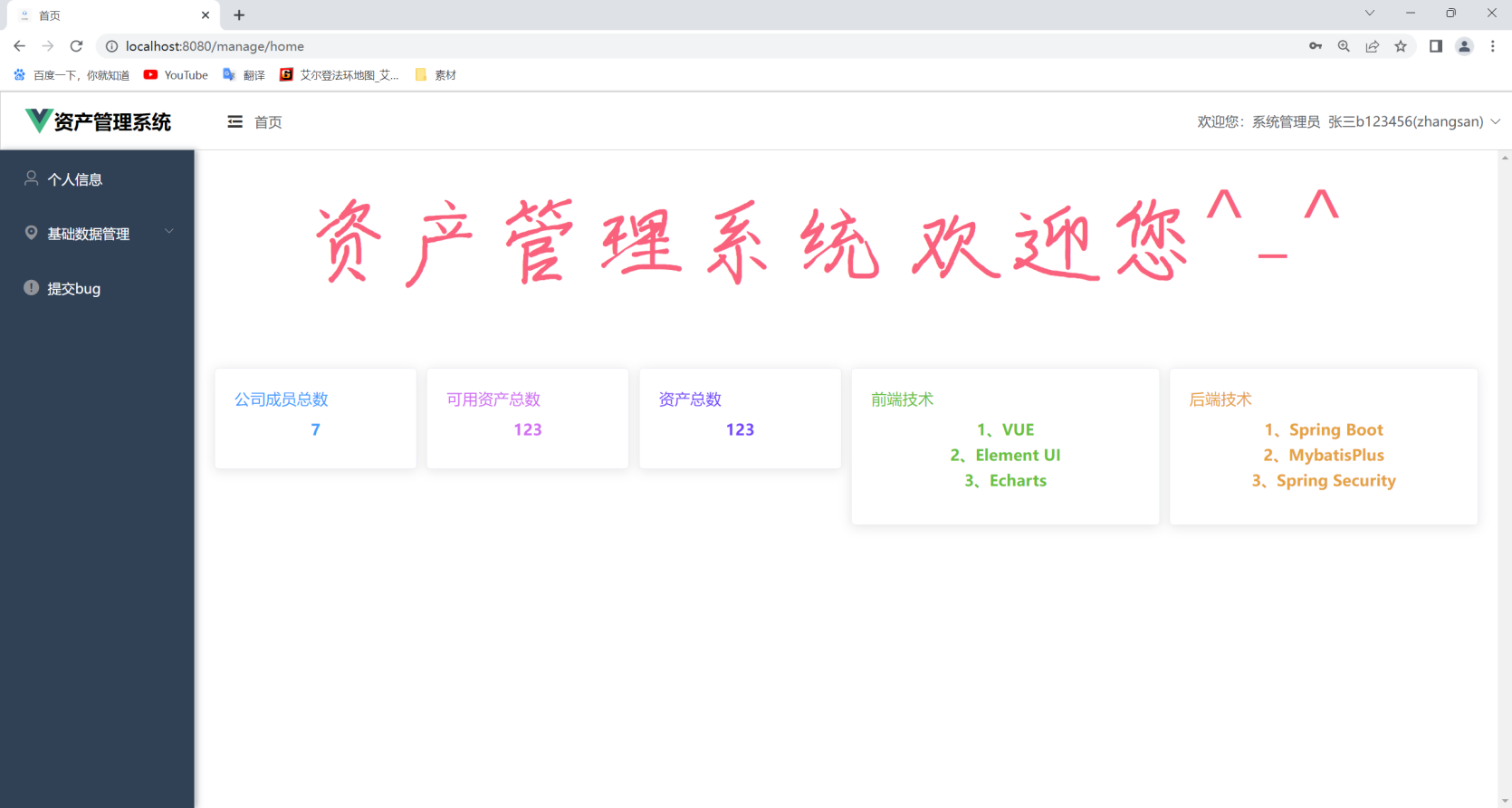
**图4-1 系统登录界面**

当输入的用户名、密码为空或错误时，系统会给出相应的错误提示，当输入正确的用户名、密码点登录按钮时，系统会进入管理主界面，如图4-2、图4-3所示。

图形用户界面, 网站

描述已自动生成

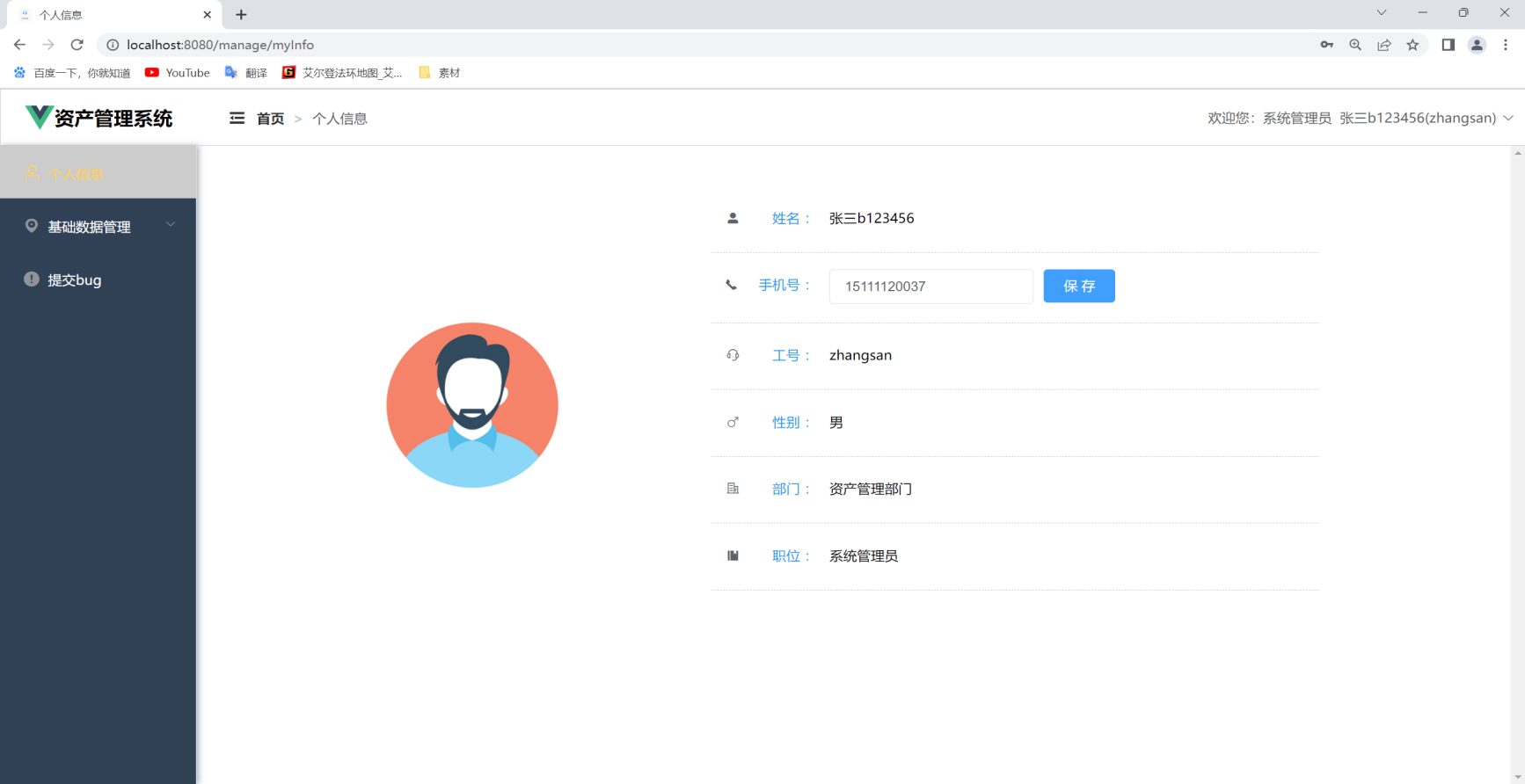
**图4-2登录失败**



**图4-3登录成功,进入主界面**

## 4.2 个人信息

此界面是对用户信息的初步展示，涉及用户的姓名、手机号、工号、姓名、部门、职位，如图4-4所示。

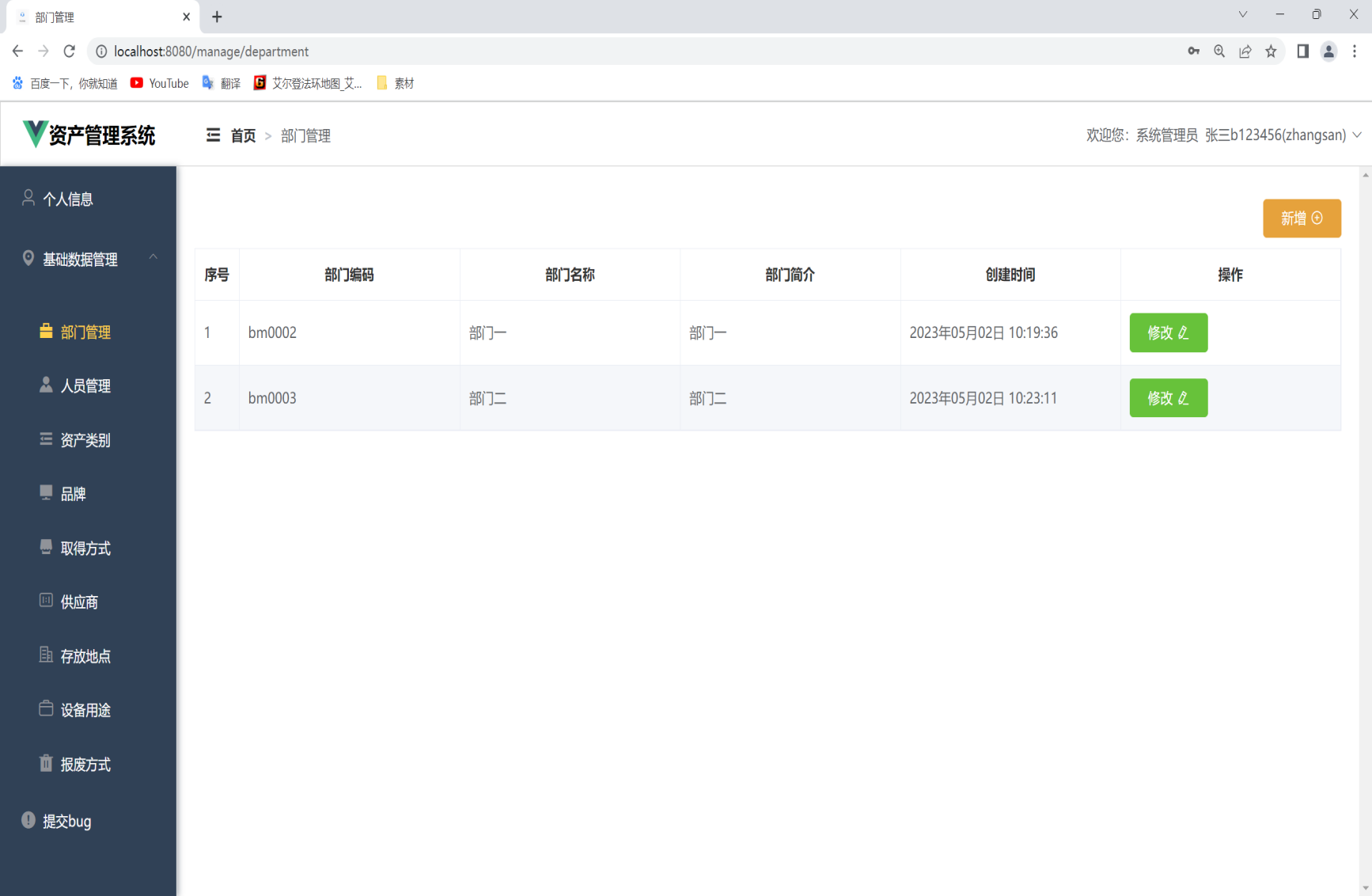


**图4-4 首页**

## 4.3 基本数据管理

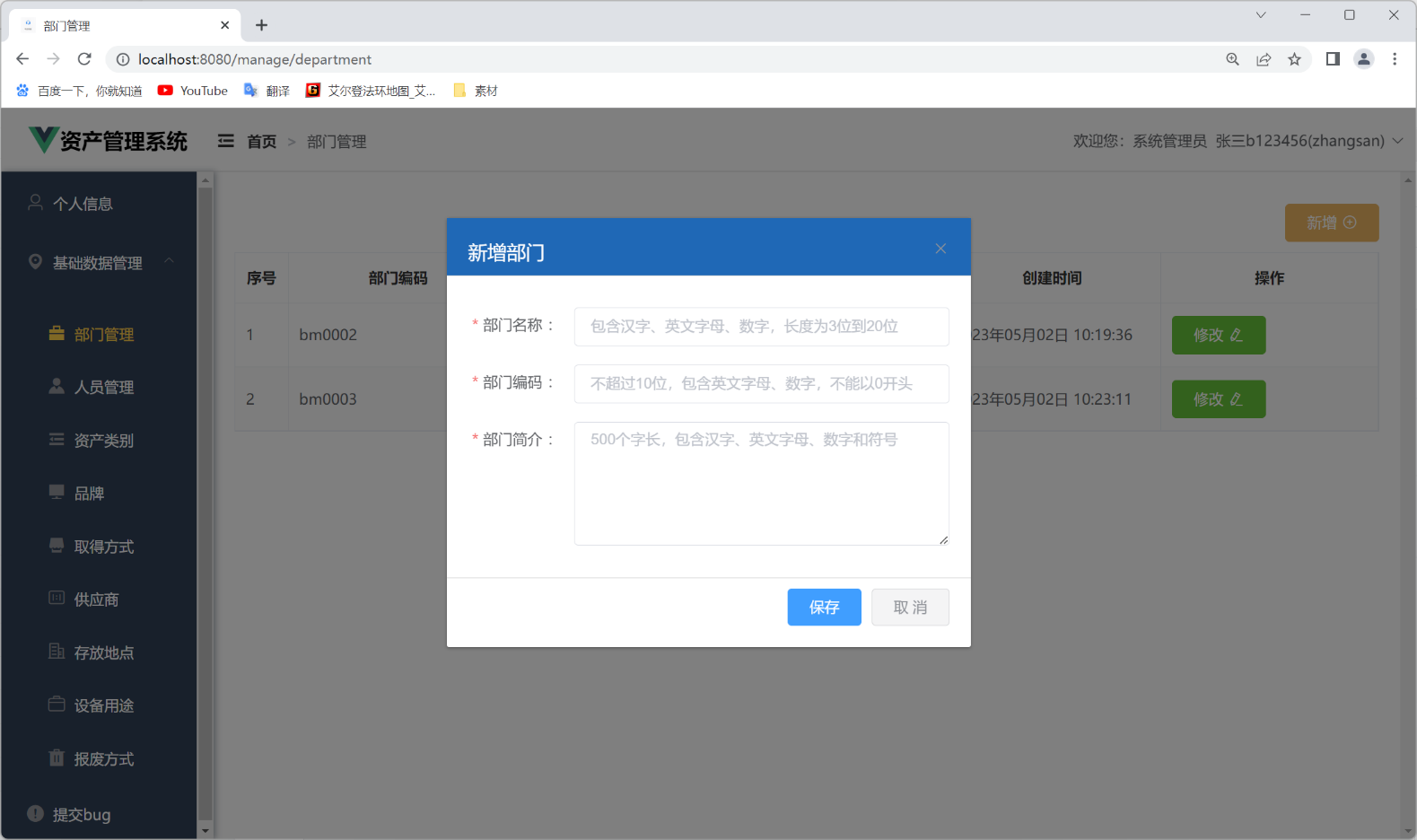
### 4.3.1 部门管理

点击部门管理可进入相应界面，如图4-5所示。



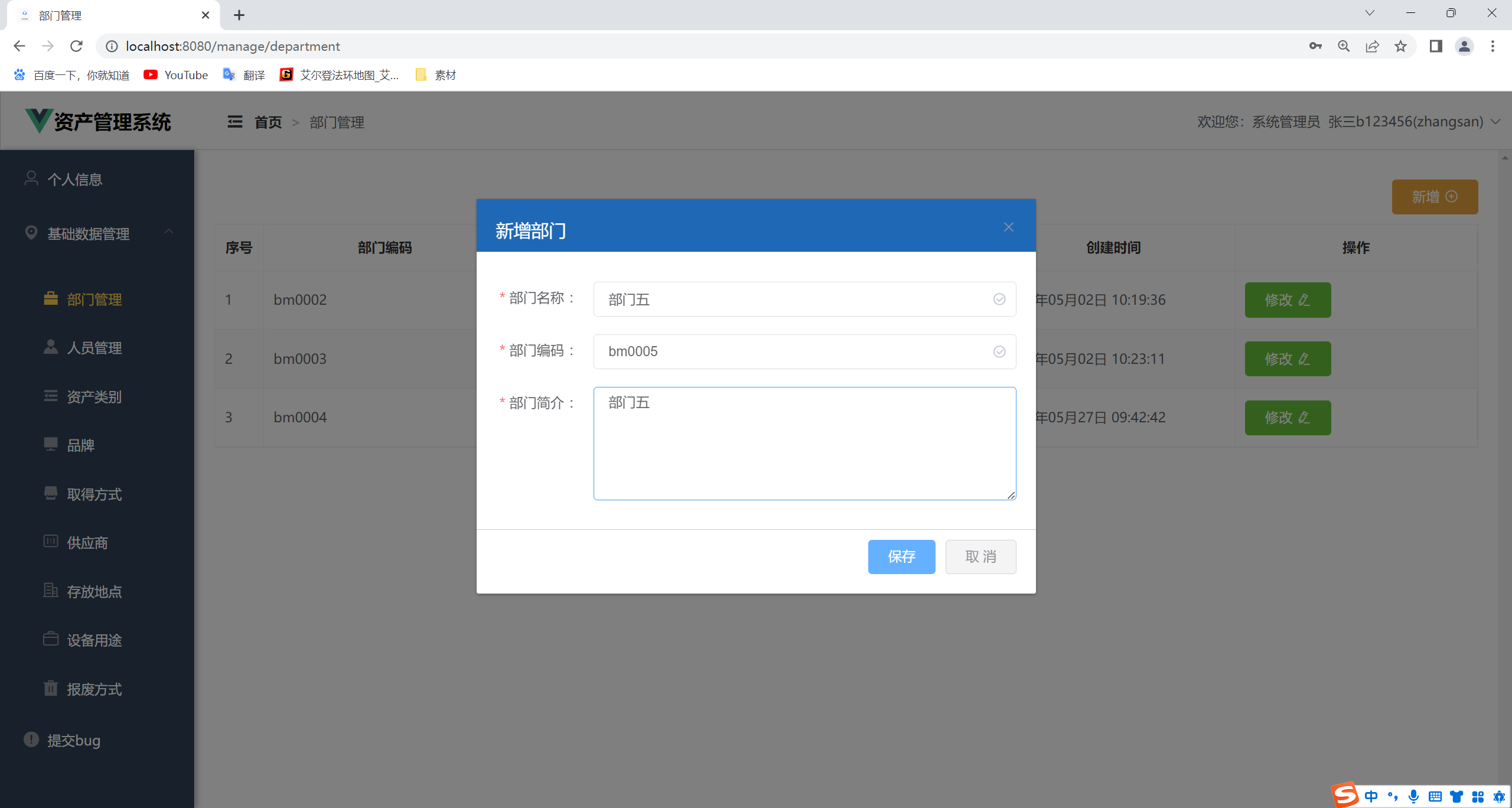
**图4-5 部门管理图**

点击新增，会弹出相应的表单让我们输入内容，如图4-6所示。

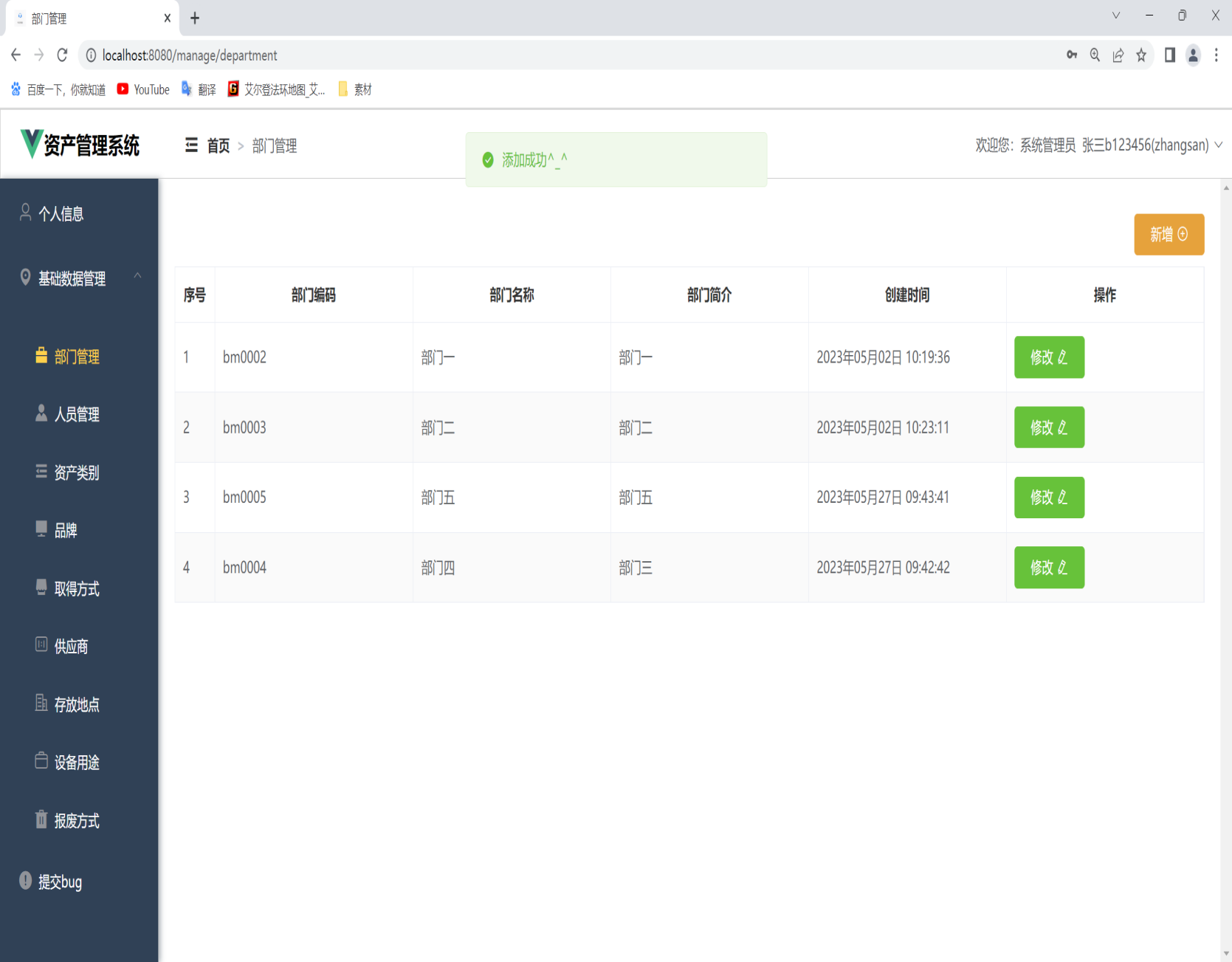


**图4-6 新增部门**

输入完相应内容，点击确认，如数据无误，则会相应的显示出新增的部门在列表中，如图4-7、图4-8所示。



**图4-7新增部门弹框**

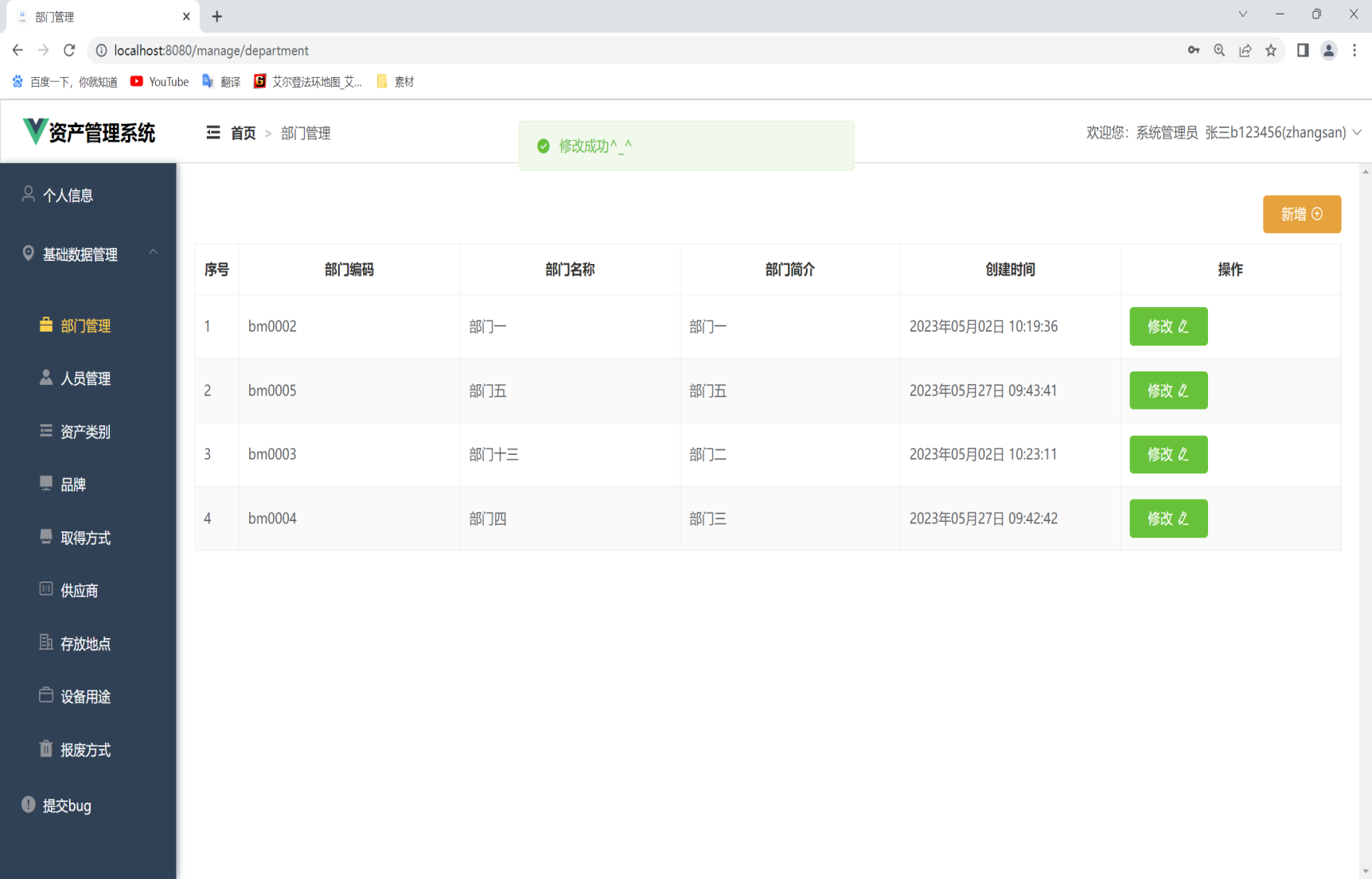


**图4-8新增部门成功**

修改部门信息，点击编辑即可，单击后会出现如下弹窗，如图4-9所示，点击确定，即可修改成功，如图4-10所示。



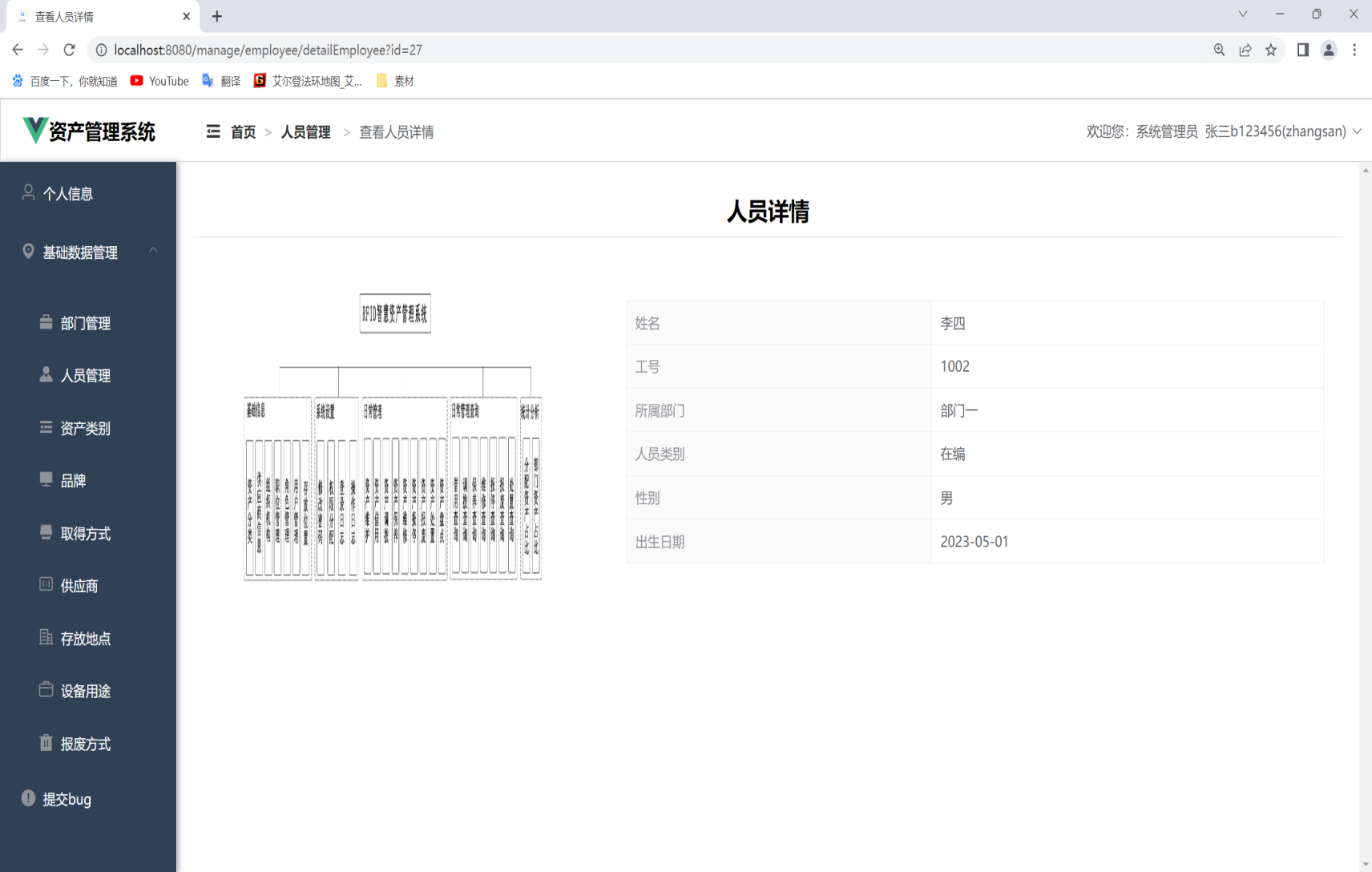
**图4-9修改部门弹框**



**图4-10修改部门成功**

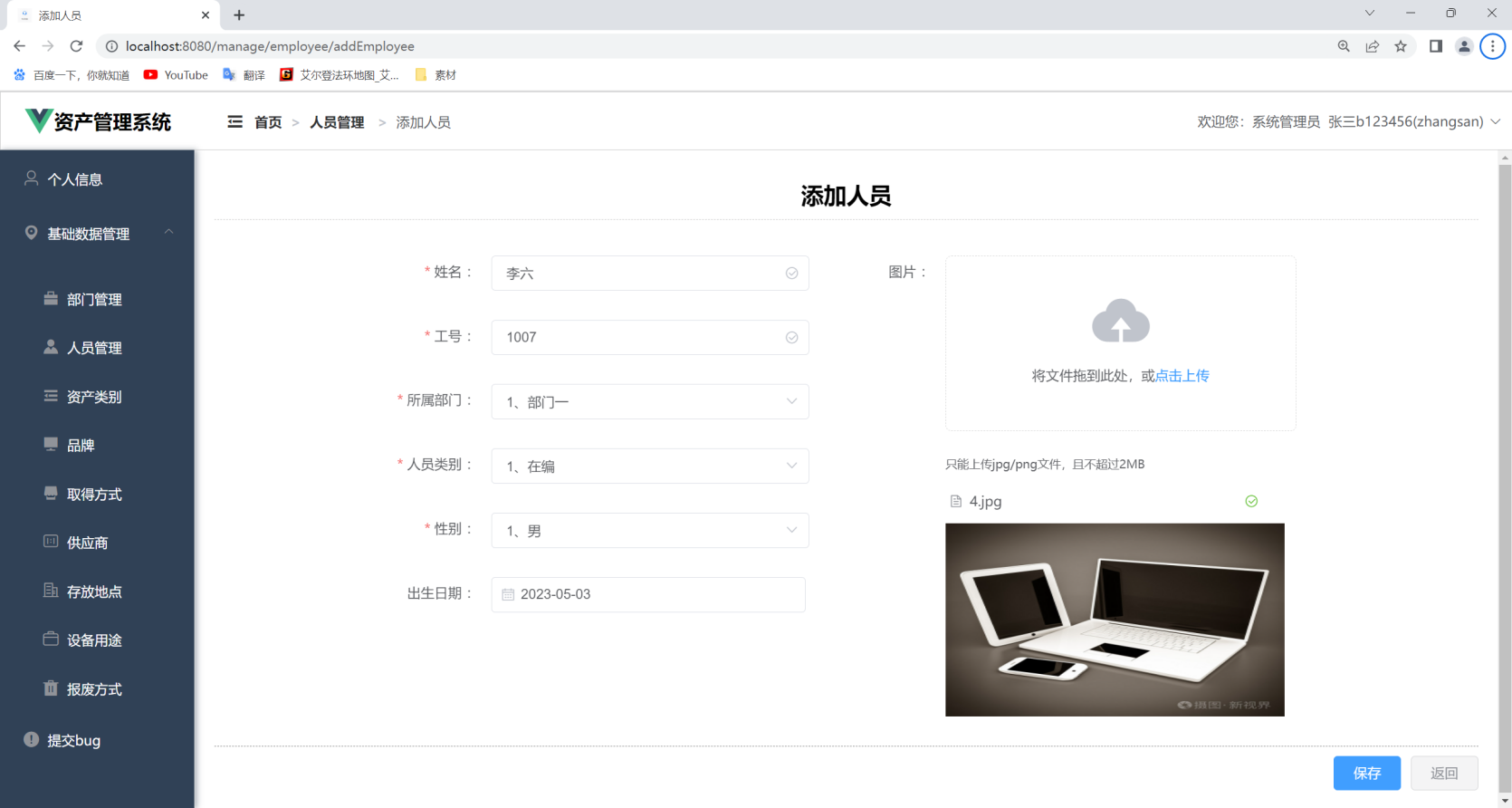
### 4.3.2 人员管理

点击任意人员即可查看，单击后展示界面如图4-11所示。

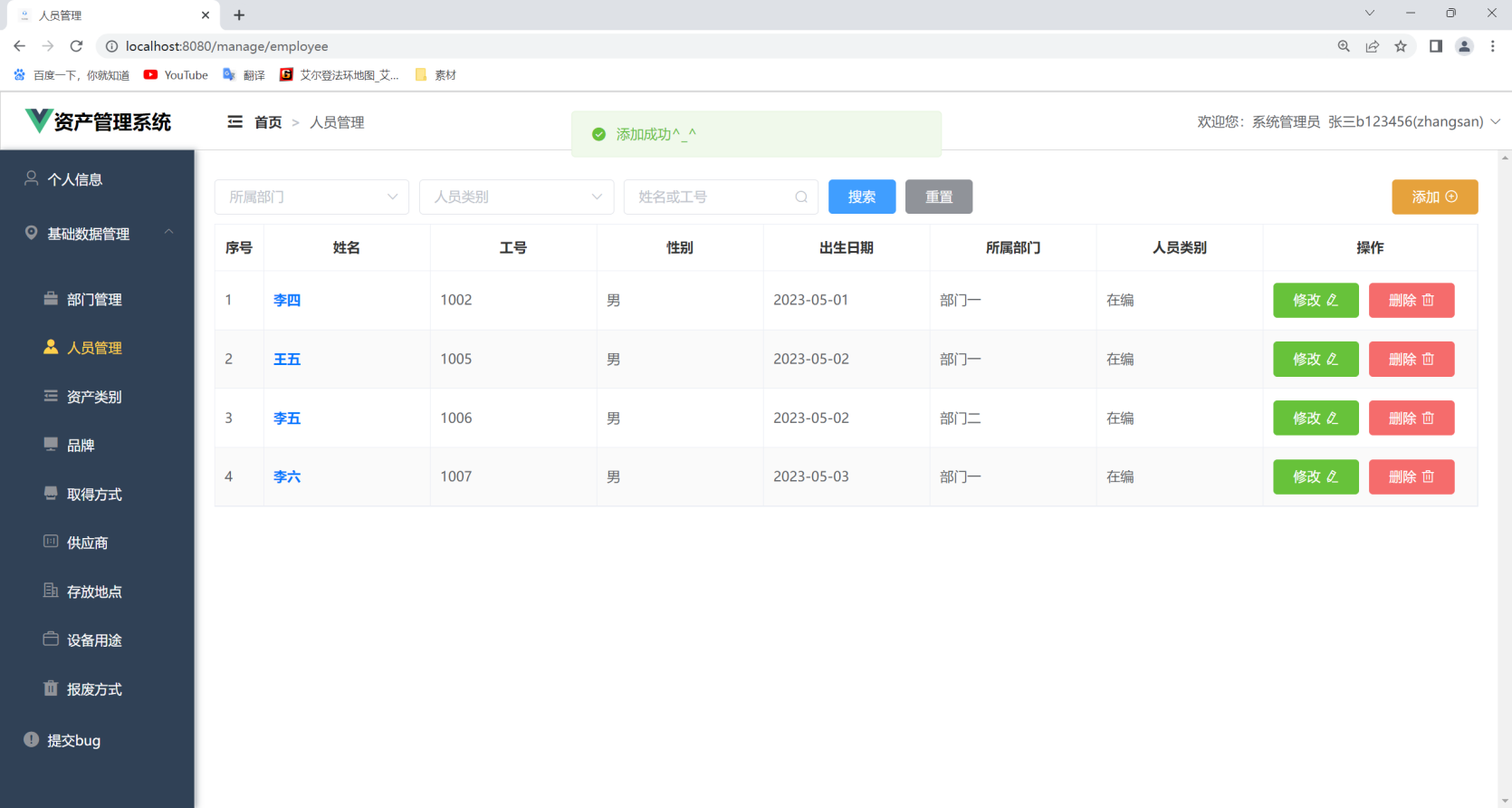


**图4-11 人员信息**

点击添加人员，可进行人员添加操作，视选择不同可进行不同的部门添加方案，如图4-12、图4-13所示。

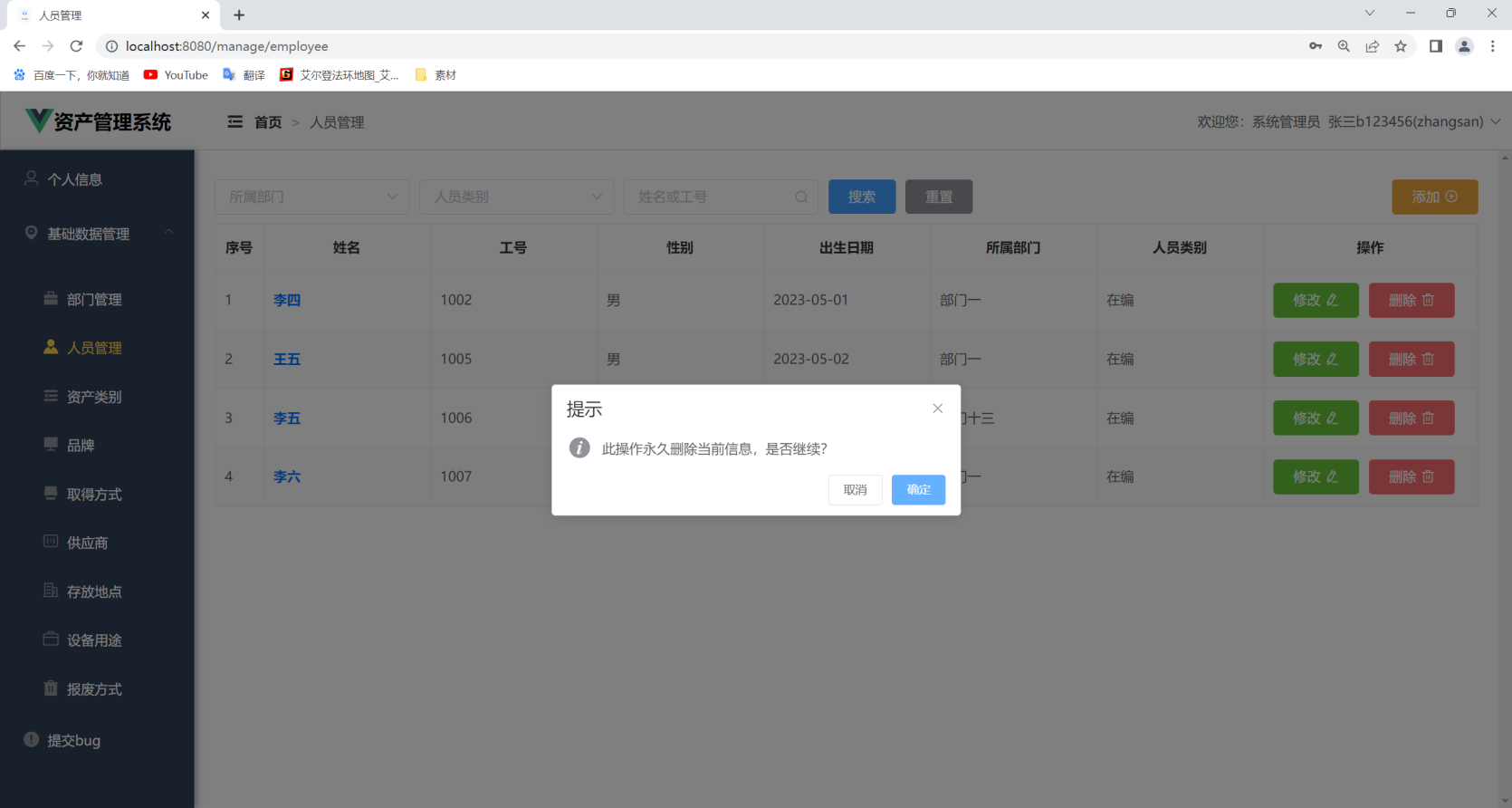


**图4-12 新增人员**

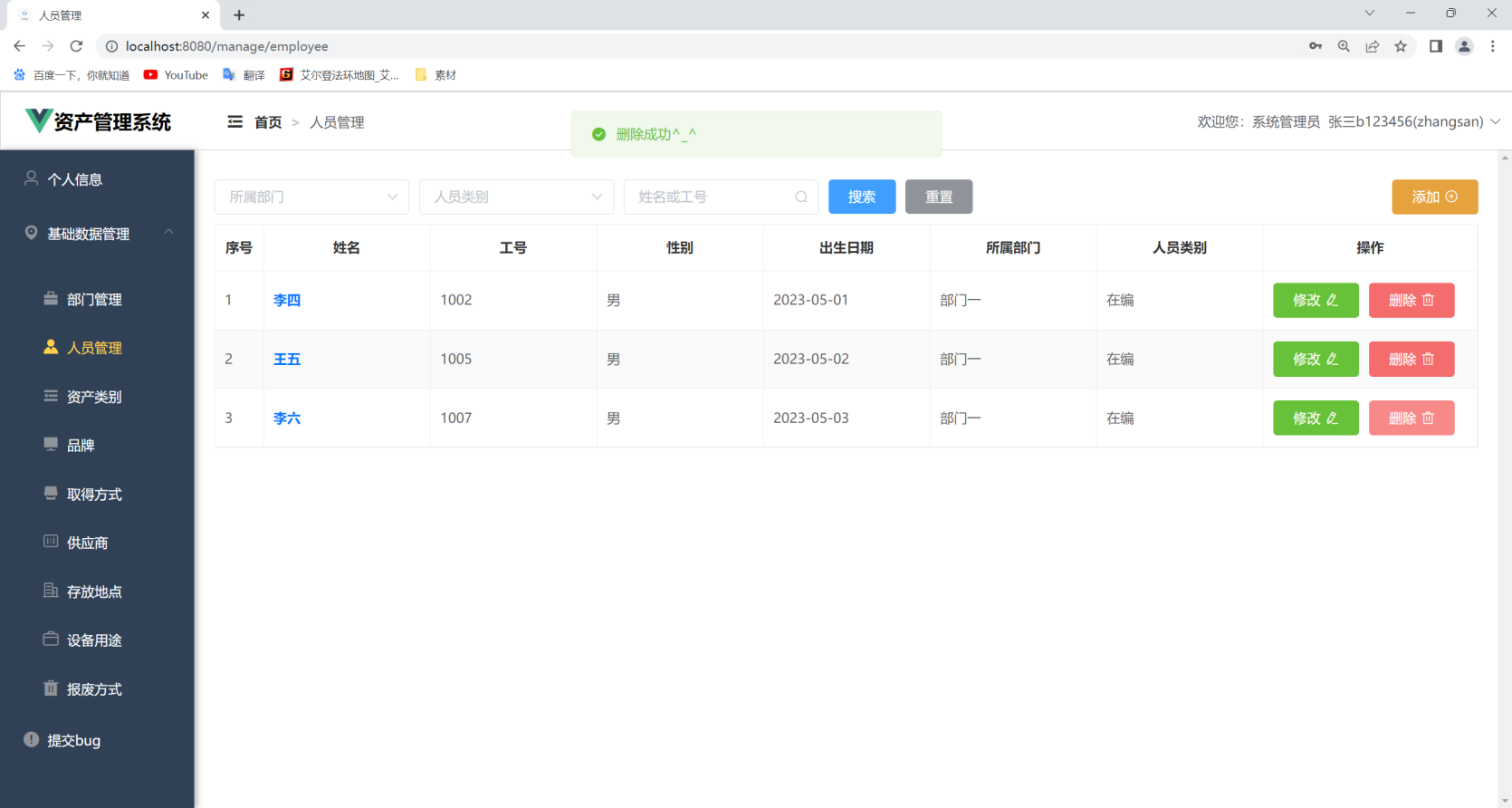


**图4-13 添加人员成功**

单击删除可将本人员从此部门中移除，具体如图4-14、图4-15所示。



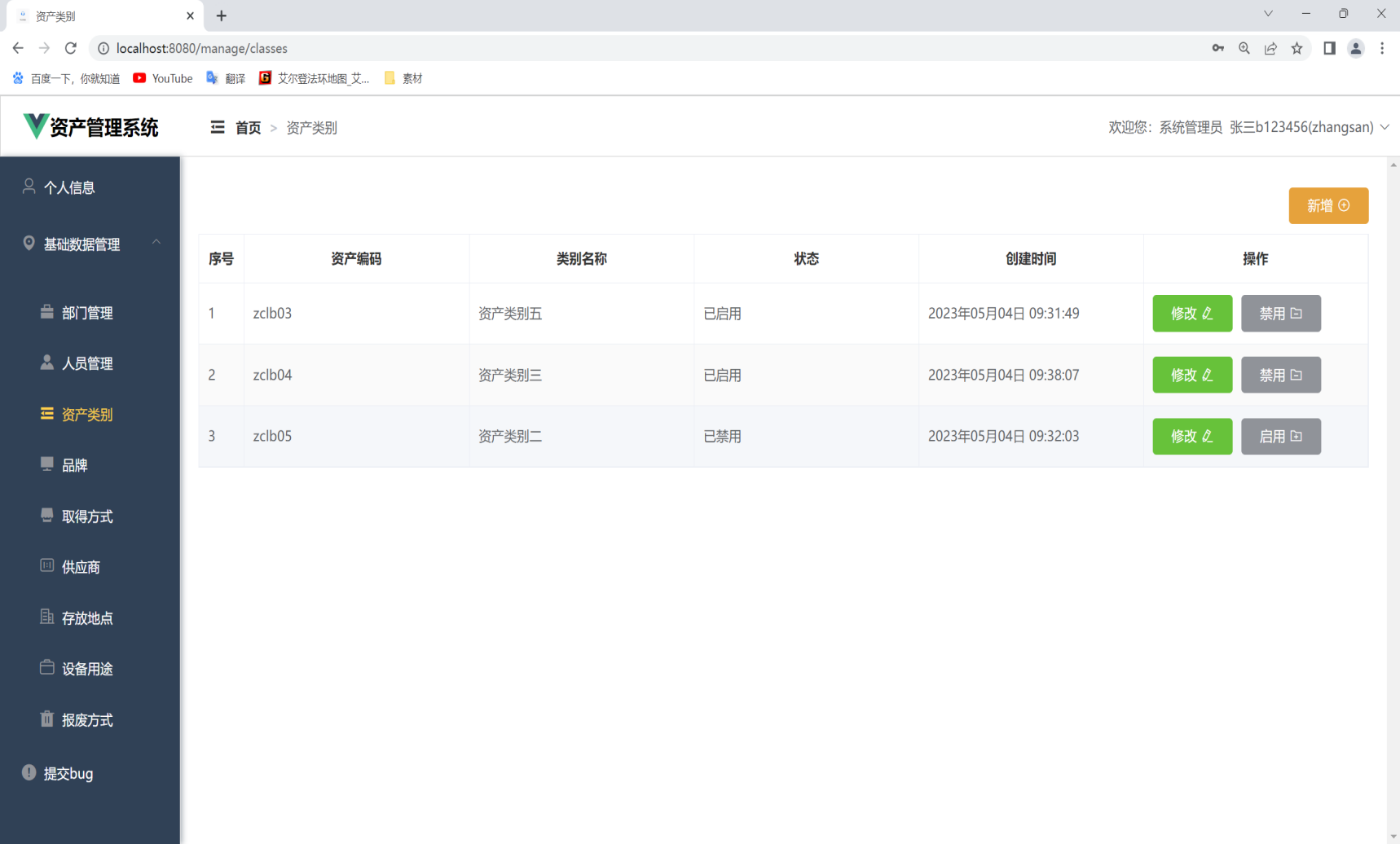
**图4-14删除人员**



**图 4-15删除成功**

### 4.3.3 资产类别管理

单击资产类别可进入相应界面，如图4-16所示，该界面主要作用是列出所有资产类别信息，以一页15条展示，在此可对相应数据进行增删改查的操作。



**图 4-16 资产类别管理**

单击新增可新增资产类别相关，具体如图4-17所示，单击保存新增成功，如图4-18所示。

图形用户界面, 应用程序

描述已自动生成

**图 4-17 新增资产类别**

图形用户界面, 应用程序, 电子邮件

描述已自动生成

**图 4-18 新增成功**

单击编辑可修改资产类别相关信息，具体如图4-19所示，单击确定修改成功，如图4-20所示。

图形用户界面, 应用程序

描述已自动生成

**图4-19 修改资产类别**

图形用户界面, 应用程序

描述已自动生成

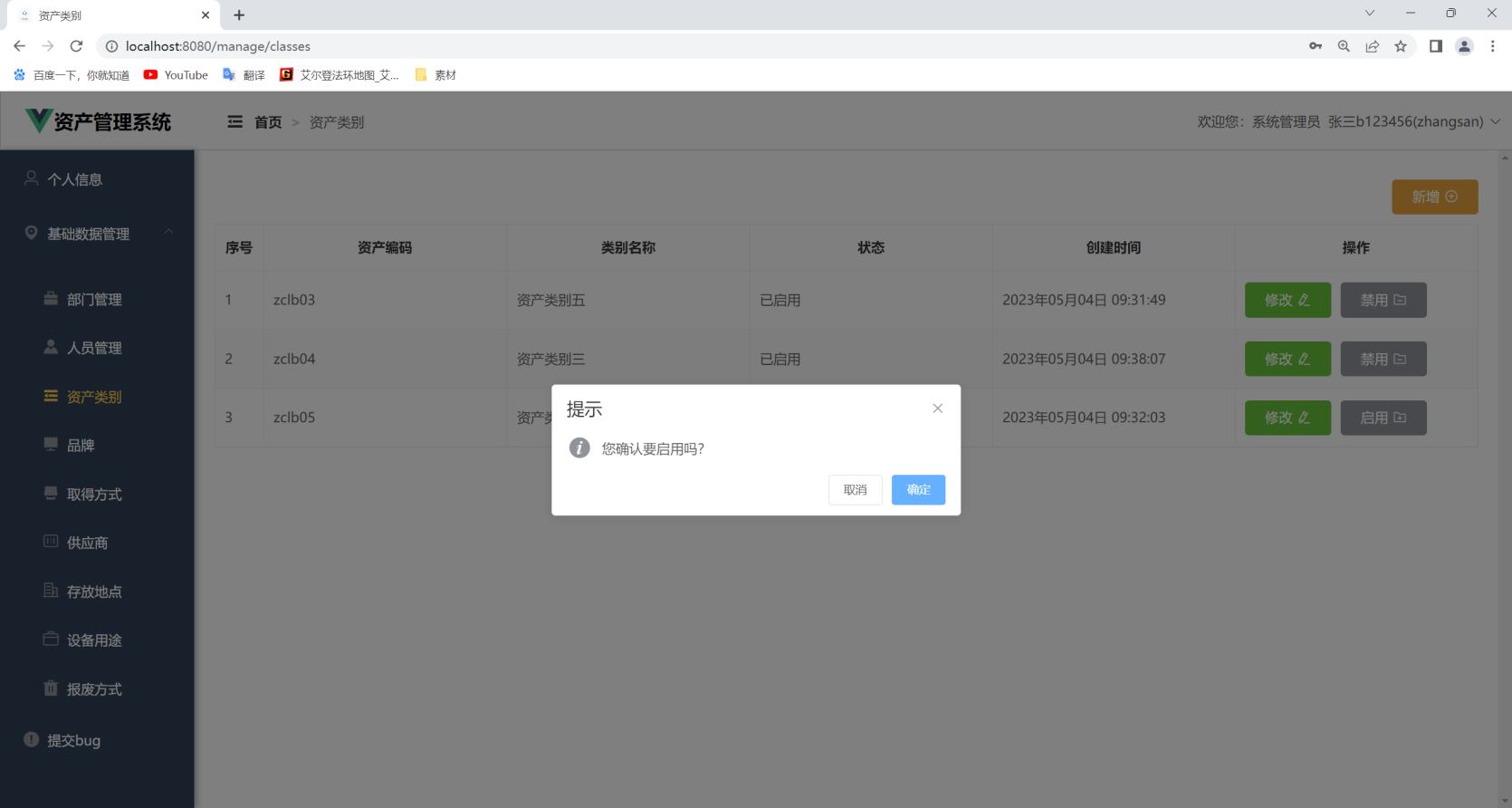
**图4-20 修改成功**

单击禁用可禁用资产类别，当然，事先会有确认弹窗，如图4-21所示。



**图4-21 禁用确认**

单击启用可启用资产类别，当然，事先会有确认弹窗，如图4-22所示。



**图4-22 启用确认**

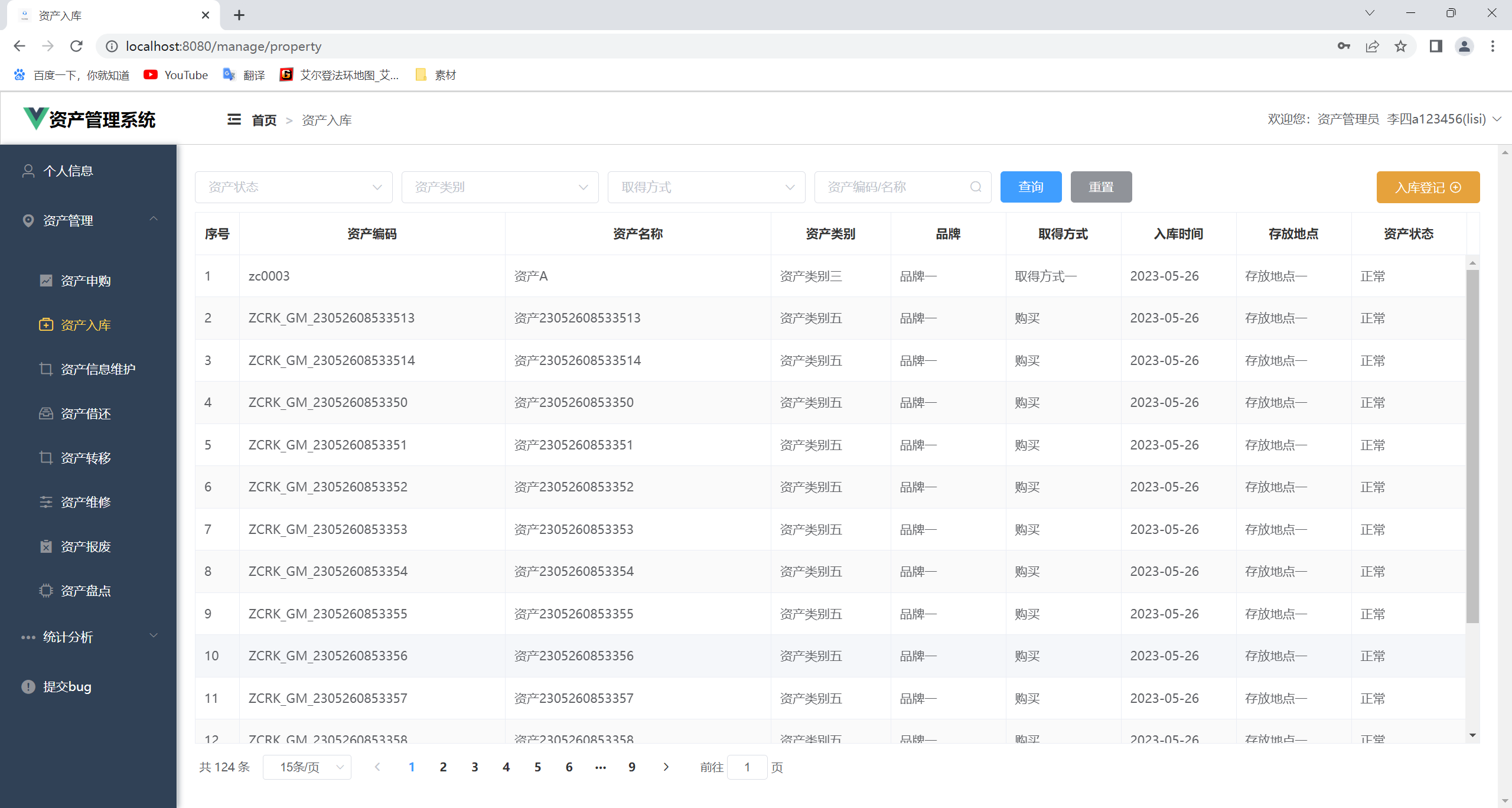
## 4.4日常管理

### 4.4.1 宠物领养

该部分是针对资产的信息进行相应管理操作

#### 4.4.1.1 资产入库

该部分是对获得资产进行入库，通过单击宠物领养——资产入库，即可进入相应界面，如图4-25所示。



**图4-25 资产入库列表**

在此还是先说资产入库，可以看见在上图（4-25）中，我们看见有入库登记按钮，单击按钮后我们可以看见如图4-26所示界面。

图形用户界面, 应用程序, 电子邮件

描述已自动生成

**图4-26 入库登记页面**

在输入完相应数据且无错误后我们就可以单击保存，当列表中出现刚刚输入的资产信息那就说明入库成功了，出现如图4-27所示界面。

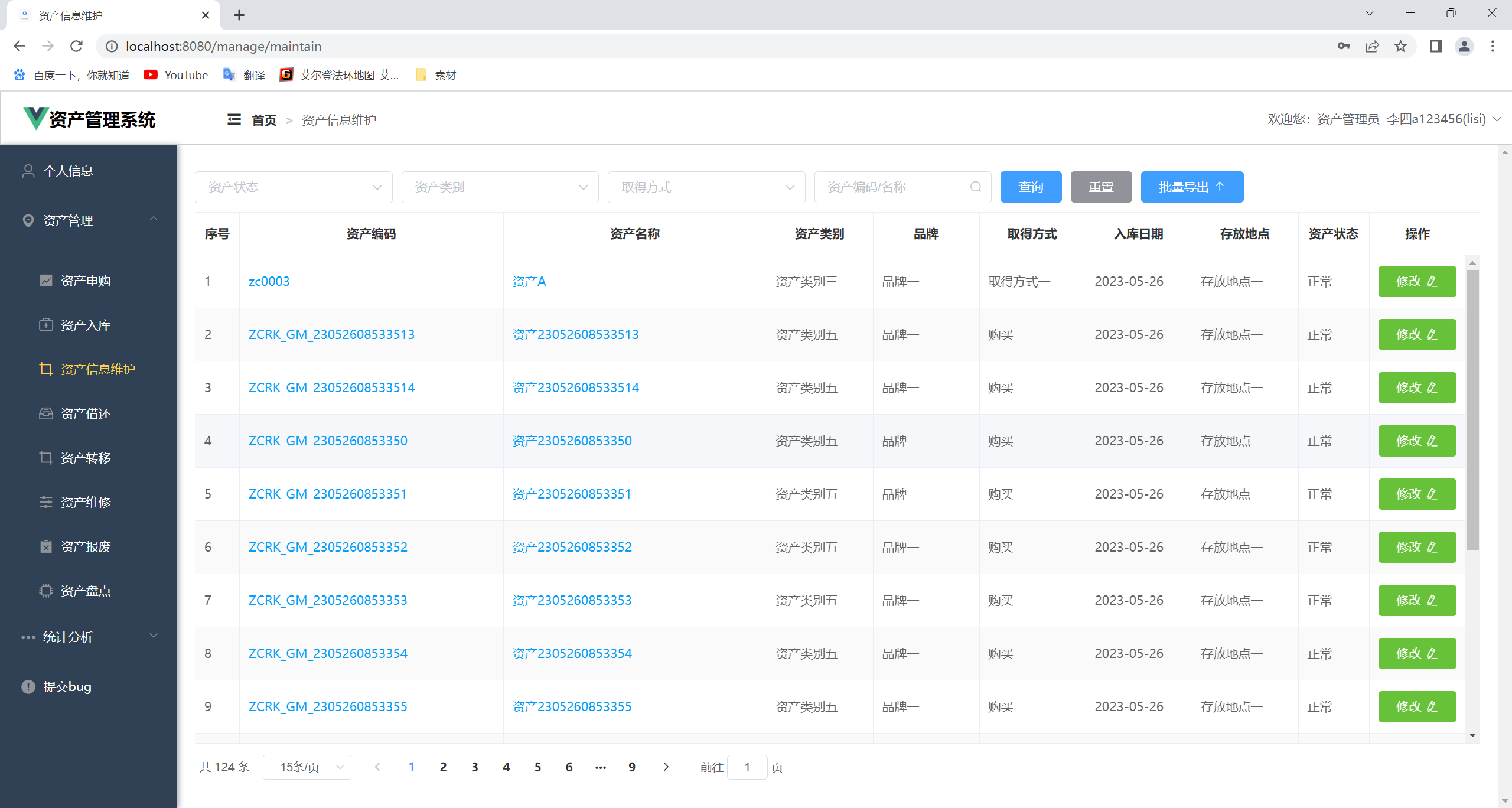
图形用户界面, 应用程序

描述已自动生成

**图4-27 入库登记成功**

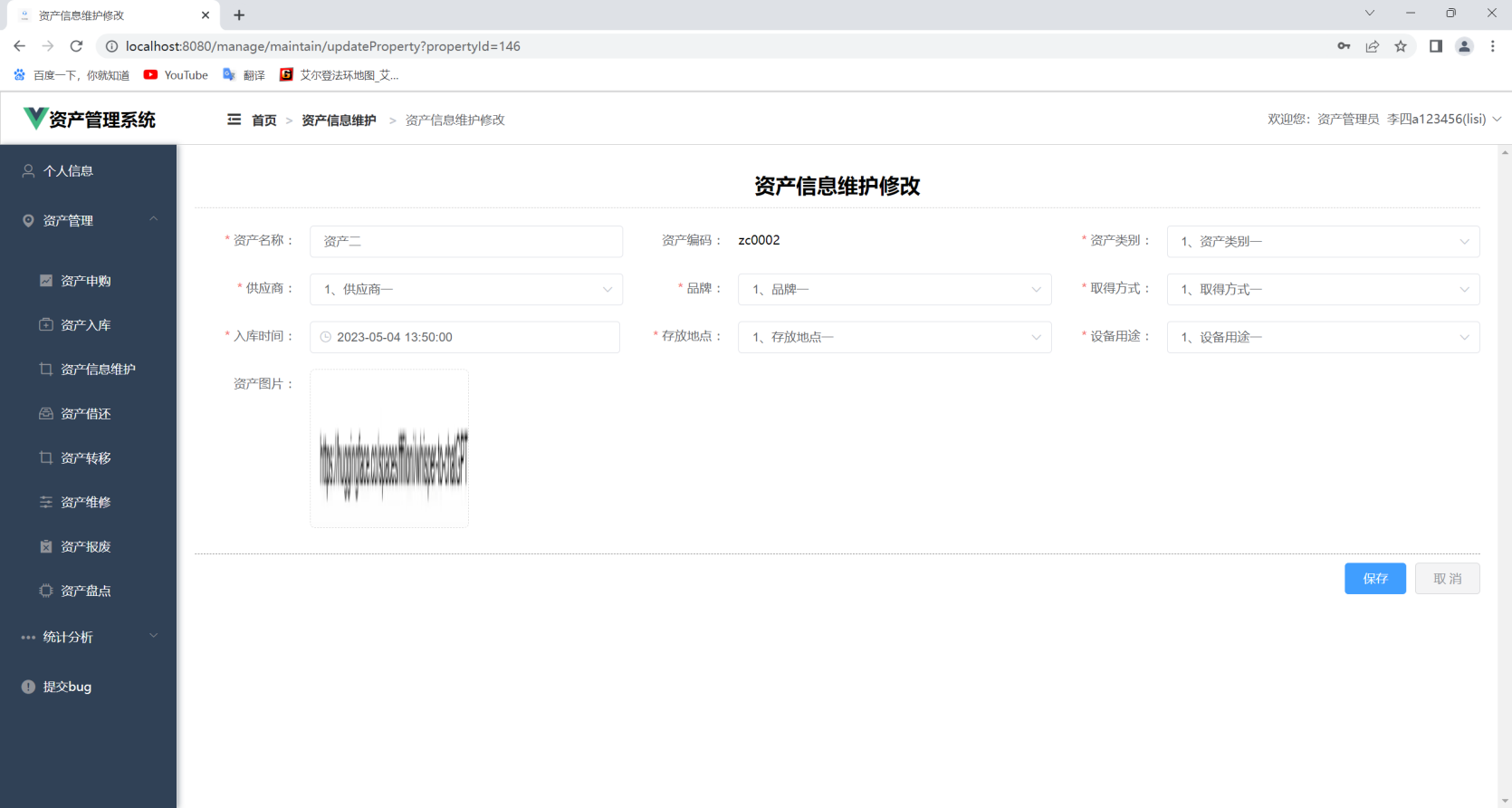
#### 4.4.1.2资产信息维护

该部分是对入库资产进行信息维护，通过单击宠物领养——资产信息维护，即可进入相应界面，如图4-28所示。



**图4-28 资产信息维护列表**

在每条资产信息操作列有修改按钮，单击按钮后我们可以看见如图4-29所示界面。



**图4-29 资产信息维护修改页面**

在输入完相应数据且无错误后我们就可以单击保存，当列表中出现刚刚输入的资产信息那就说明修改成功了，出现如图4-30所示界面。

图形用户界面, 应用程序, 电子邮件

描述已自动生成

**图4-30 资产信息修改成功**

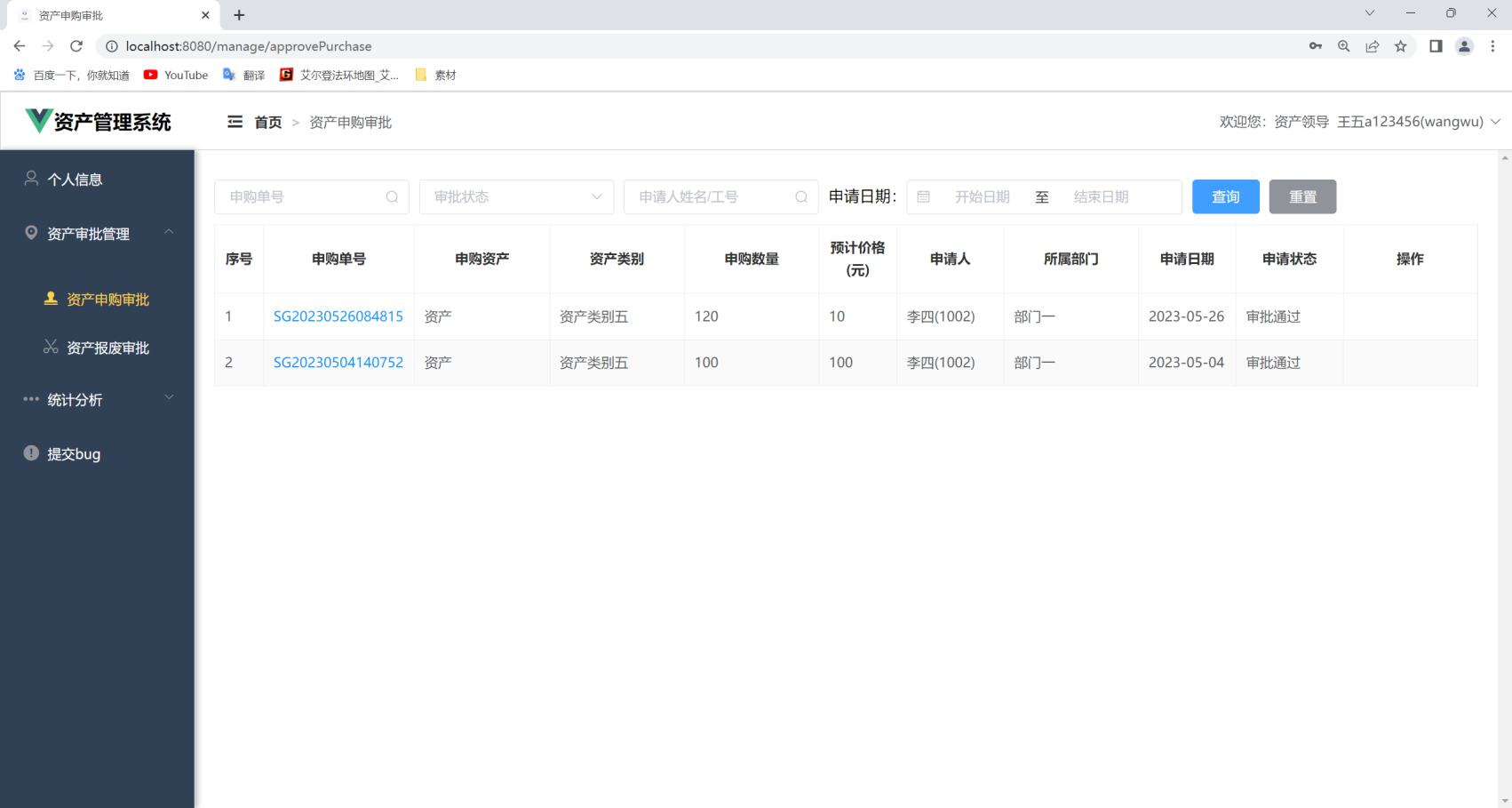
此上便是宠物领养的主要内容。

### 4.4.2 资产审批管理

该部分是针对资产审批信息进行相应管理操作

#### 4.4.2.1 资产申购审批

单击资产审批管理——资产申购审批即可进入该项，详细界面如图4-31所示。



**图4-31 资产申购审批列表**

在这个界面中我们可以进行资产申购审批，点击审批按钮，弹出资产申购登记审批弹窗。选择审批结果，单击保存按钮可对数据进行相应操作。出现如图4-32、4-33所示的界面。

图形用户界面, 应用程序

描述已自动生成

**图4-32 资产申购登记审批**

图形用户界面, 应用程序

描述已自动生成

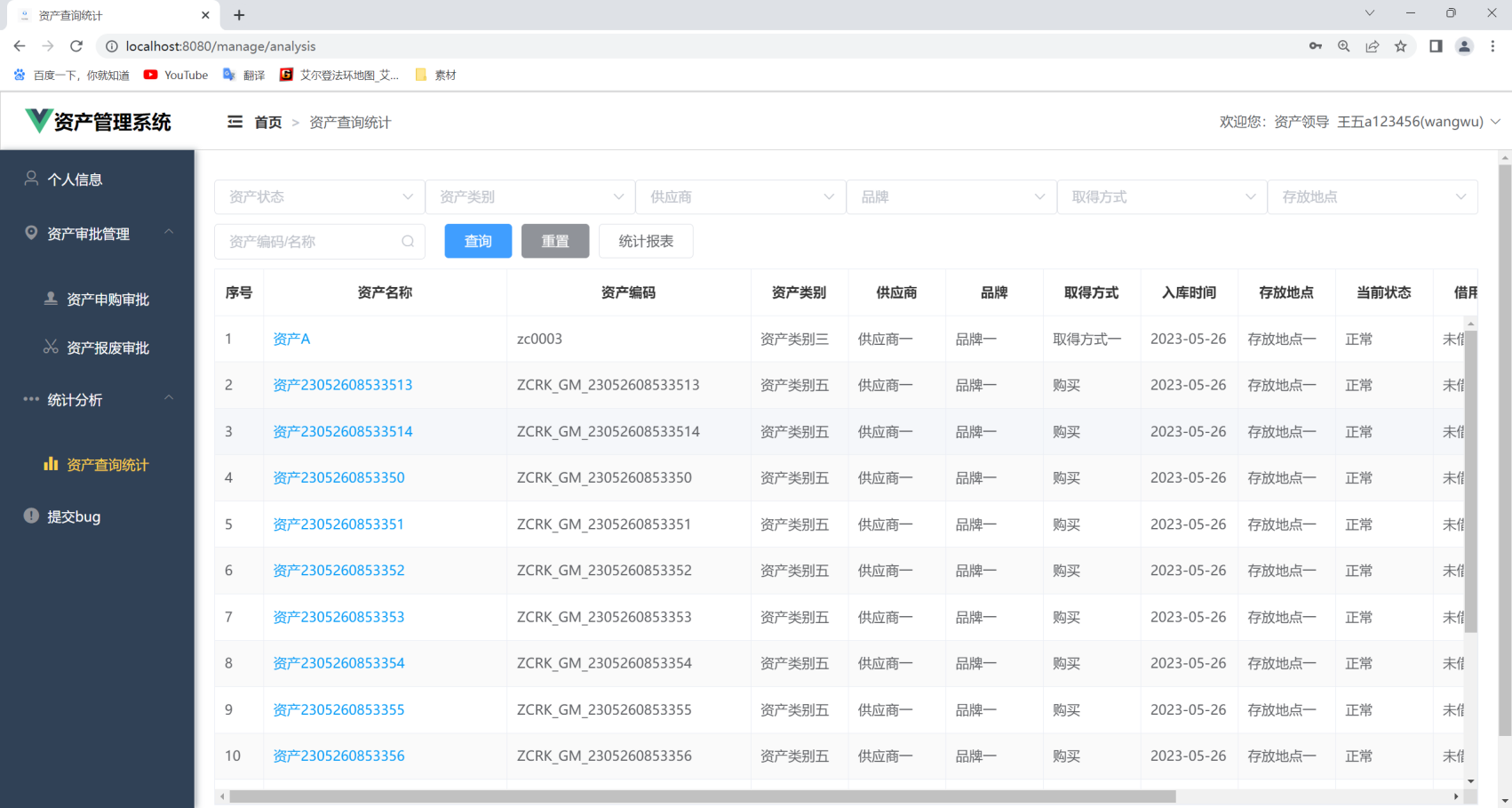
**图4-33 资产申购登记审批成功**

此上便是资产审批管理的主要内容。

## 4.5 统计分析

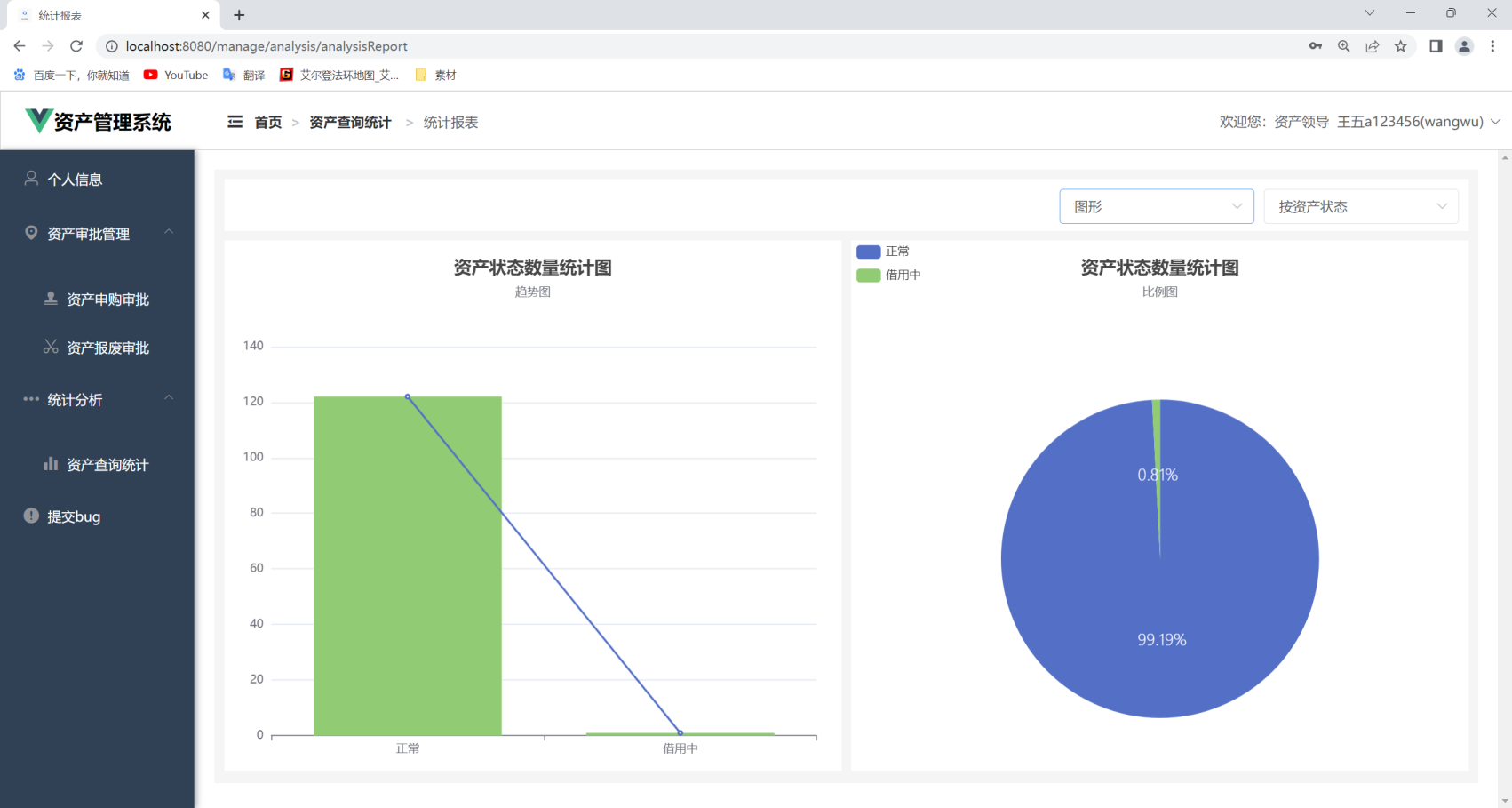
### 4.5.1 功能实现

该功能项仅有一个模块，如图4-34所示。



**图4-34 资产查询统计**

这个界面负责展示所有的资产信息以及进行相关信息的统计查询，另外点击统计报表按钮可以进入统计报表页面，如图4-35所示。



**图4-35统计报表页面**

### 4.5.2 关键代码

Long total = propertyMapper.selectCount(null);List<PropertyStatusVO> propertyStatusVOList = propertyMapper.getStatusCount();  
 *//数组* List<Integer> statuses = new ArrayList<>();  
 String[] statusNames = new String[propertyStatusVOList.size()];  
 for (int i = 0; i < propertyStatusVOList.size(); i++) {  
 PropertyStatusVO propertyStatusVO = propertyStatusVOList.get(i);  
 if (propertyStatusVO.getPropertyStatus() == 0) {  
 statusNames[i] = "正常";  
 } else if (propertyStatusVO.getPropertyStatus() == 1) {  
 statusNames[i] = "已报废";  
 } else if (propertyStatusVO.getPropertyStatus() == 2) {  
 statusNames[i] = "借用中";  
 }  
 statuses.add(Math.*toIntExact*(propertyStatusVO.getNum()));  
 }  
 Map<String, Object> mapStatus = new HashMap<>();  
 mapStatus.put("x", CollUtil.*newArrayList*(statusNames));  
 mapStatus.put("y", statuses);  
 return mapStatus;

}

}

此上便是我系统功能的简要描述，基本功能已简单概述完。

# 5 系统测试

## 5.1 测试方法与策略

软件设计完成后，要对其使用进行测试，本系统采用的开发工具是 IntelliJ IDEA2020、MySQL、Maven、Git，首先打开 MySQL 数据库，将建立的数据库载入到此数据库开发工具里， 然后在浏览器的地址栏里输入系统的起始页的地址。

本系统主要采用人工测试的方式进行黑盒测试对系统功能进行验证，使用到的测试用例设计方法包括：边界值分析、等价类划分、错误猜测法、因果图法等。

本项目的测试策略包括：

1.功能测试

功能测试验证系统按照设计要求实现模块功能，数据完整、界面美观、操作方便。

2.数据项测试

输入数据项是否能够正确显示，并添加到系统中；

是否能够识别非法数据；

提示消息是否可理解。

3.业务测试

功能测试完成后进行业务测试，业务测试关注的要点是业务流程，及数据流从软件中的一个模块流到另一个模块的过程中的正确性。

## 5.2 登录模块测试

登录模块测试主要验证能否正确实现用户登录，对用户输入的数据进行合法性的验证，对输入的非法数据给出提示，并验证提示信息的可理解性，该模块的测试用例及测试执行结果如下表所示。

**表 5-1** **登录模块测试用例表**

| **用例**  **编号** | **测试项目** | **测试标题** | **重要**  **级别** | **预置条件** | **输入** | **执行步骤** | **预期输出** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 用户  登录 | 【登录】按钮功能测试 | 高 | 打开登  录界面 | 无 | 点击【登录】  按钮 | 账号密码  不能为空 |
| 2 | 用户  登录 | 【登录】按钮功能测试 | 高 | 打开登  录界面 | 账号 | 输入账号，  不输入密码，  点击【登录】  按钮 | 密码不能  为空 |
| 3 | 用户  登录 | 【登录】按钮功能测试 | 高 | 打开登  录界面 | 密码 | 输入密码，  不输入账号，  点击【登录】  按钮 | 账号不能  为空 |
| 4 | 用户  登录 | 【登录】按钮功能测试 | 高 | 打开登  录界面 | 账号  密码 | 输入账号，  错误密码，  点击  【登录】按钮 | 密码错误 |
| 5 | 用户  登录 | 【登录】按钮功能测试 | 高 | 打开登  录界面 | 账号  密码 | 输入密码，  错误账号，  点击【登录】  按钮 | 账号错误 |
| 6 | 用户  登录 | 【登录】按钮功能测试 | 高 | 打开登  录界面 | 账号  密码 | 输入有效账号  密码，  点击【登录】  按钮 | 登录成功 |

## 5.3 部门管理模块测试

部门管理模块的测试主要看部门的新增、修改操作是否能正常进行，对用户输入的数据进行合法性的验证，对输入的非法数据给出提示，并验证提示信息的可理解性，该模块的测试用例及测试执行结果如下表所示。

**表 5-2 部门管理模块测试用例表**

| **用例编号** | **测试项目** | **测试标题** | **重要级别** | **预置条件** | **输入** | **执行步骤** | **预期输出** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |
| 1 | 部门管理列表 | 列表表头文字正确性验证 | 中 | 角色为：系统管理员  登录成功  点击部门管理菜单 | 无 | 查看界面导航、  列表表头名称、  表格样式显示 | 1、面包屑：首页>部门管理2、左侧菜单：部门管理 高亮显示3、表头名称显示：序号、部门编码、部门名称、创建日期、部门简介、操作，样式正确4、页面title显示“部门管理”5、页面显示：新增按钮 |  |
| 2 | 部门管理列表 | 面包屑 | 低 | 角色为：系统管理员  登录成功 | 无 | 点击导航：首页 | 跳转到首页 |  |
| 3 | 部门管理列表 | 列表排序 | 低 | 角色为：系统管理员  登录成功  点击部门管理菜单 | 无 | 部门管理列表有多条部门数据 | 列表部门按照部门名称升序，列表默认显示所有部门信息 |  |
| 4 | 部门管理新增 | 新增部门管理 | 高 | 角色为：系统管理员  登录成功  点击部门管理菜单 | 无 | 点击“新增”按钮 | 1、弹出新增部门弹框，弹框title：新增部门，弹框有X按钮  2、显示：部门名称、部门编码、部门简介，【保存】【取消】按钮  3、部门名称、部门编码前均显示红色\* |  |
| 5 | 部门管理新增 | 输入全部正确信息，进行新增操作 | 高 | 角色为：系统管理员  登录成功，进入部门管理模块  用户打开“新增部门管理”弹框 | 1、部门名称：20个字长；包含汉字、英文字母、数字，同已有部门名称不重复  2、部门编码：10个字长，包含英文和数字  3、部门简介：500个字长，包含汉字、英文字母、数字和符号 | 输入以上数据  点击保存按钮 | 1、保存成功，新增部门管理弹框关闭  2、添加部门管理成功  3、新添加的部门管理显示在列表中，列表数据显示信息同新增一致，创建日期显示创建具体日期，显示格式：2017-03-03  4、列表操作显示：修改 |  |
| 6 | 部门管理新增 | 部门名称不正确（未输入），进行新增 | 高 | 角色为：系统管理员  登录成功，进入部门管理模块  用户打开“新增部门管理”弹框 | 1、部门名称为空  2、部门编码：12345678ab  3、部门简介：67fjg你好!@#$ | 输入以上数据  点击保存按钮 | 提示：部门名称必填，请重新输入。 |  |
| 7 | 部门管理新增 | 部门名称不正确（字数不正确，小于3个字），进行新增 | 高 | 角色为：系统管理员  登录成功，进入部门管理模块  用户打开“新增部门管理”弹框 | 1、部门名称：基础  2、部门编码：12345678ab  3、部门简介：67fjg你好!@#$ | 输入以上数据  点击保存按钮 | 提示：部门名称输入有误，请重新输入。 |  |

## 5.4 资产入库模块测试

资产入库模块测试为入库登记测试，对用户输入的数据进行合法性的验证，对输入的非法数据给出提示，并验证提示信息的可理解性，该模块的测试用例及测试执行结果如下表所示。

**表 5-3资产入库模块测试用例表**

| **用例编号** | **测试项目** | **测试标题** | **重要级别** | **预置条件** | **输入** | **执行步骤** | **预期输出** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |
| 1 | 资产入库列表 | 界面导航、页面信息验证 | 中 | 用户登录成功  打开资产入库页面 | 无 | 查看界面显示 | 1、页面title显示“资产入库”、面包屑：首页>资产入库  2、左侧菜单：资产入库 高亮显示  3、页面显示查询条件、【查询】、【入库登记】按钮 |  |
| 2 | 资产入库列表 | 面包屑 | 低 | 角色为：宠物领养员登录成功 | 无 | 点击导航：首页 | 跳转到首页 |  |
| 3 | 资产入库列表 | 初始列表 | 中 | 宠物领养员登录成功，打开资产入库界面 | 无 | 查看列表 | 默认不显示列表 |  |
| 4 | 资产入库列表 | 列表表头文字正确性验证 | 中 | 宠物领养员登录成功，打开资产入库界面 | 无 | 点击“查询”按钮 | 1、表头名称显示：序号、资产编码、资产名称、资产类别、供应商、品牌、取得方式、入库日期、存放地点、资产状态，样式正确  2、列表显示资产申购数据 |  |
| 5 | 资产入库登记 | 资产入库登记界面显示 | 高 | 资产入库弹框有“入库登记”按钮 | 无 | 点击“入库登记”按钮 | 1、打开资产入库登记页面，页面title显示“资产入库登记”，面包屑：首页>资产入库>资产入库登记  2、显示：资产名称、资产编码、资产类别、供应商、品牌、取得方式、入库日期、存放地点、资产图片、【保存】【取消】按钮。  3、资产名称、资产编码、资产类别、供应商、品牌、取得方式、入库日期、存放地点字段前均显示红色\*  4、资产编码下方红色提示文字：“注意：资产编码保存后不允许修改，请认真填写。”  5、入库日期输入框默认当天 |  |
| 6 | 资产入库登记 | 面包屑 | 低 | 进入“资产入库登记”页面 | 无 | 点击导航：首页 | 跳转到首页 |  |
| 7 | 资产入库登记 | 面包屑 | 低 | 进入“资产入库登记”页面 | 无 | 点击导航：资产入库 | 跳转到资产入库列表 |  |

## 5.5 退出系统测试

该部分测试，是对系统基本功能的测试，该部分的测试用例及测试执行结果如下表所示。

**表 5-6 退出功能测试用例表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例编号** | **测试项目** | **测试标题** | **重要**  **级别** | **预置条件** | **输入** | **执行步骤** | **预期输出** |
| 1 | 退出系统 | 退出系统 | 高 | 单击系统右上角头像 | 无 | 点击【退出】按钮 | 退出系统，回到登录界面 |
| 2 | 退出系统 | 回退按钮，回归系统 | 高 | 退出系统 | 无 | 单击回退按钮，或者输入系统主界面网址 | 进入失败，停留登录界面 |