兔子大学北京校区

**课程设计报告**

**基于SpringBoot + Vue的仓库管理系统的设计与实现**

学 院： 计算机学院

专 业 名 称： 软件工程

学 生 姓 名： 忧伤大白兔

学 号： 00000001

完成日期： xxxx年 xx月xx日

**摘 要**

该系统的设计初衷是解决传统仓库管理中存在的一些痛点和问题。传统的手工管理往往容易出现库存错误、信息查找困难、操作不便等情况。本系统的目标是通过引入计算机技术和自动化管理，优化仓库管理流程，提升管理的准确性和效率。

系统分为超级管理员、仓库管理员和普通用户三种角色，为不同角色的用户提供不同的功能和权限。超级管理员具备全面管理权限，能够管理管理员和用户信息、仓库和物品分类等。仓库管理员拥有对仓库和物品的管理权限，可以进行入库和出库操作。普通用户则可以查询和管理个人信息、查看物品信息和操作日志。

在系统设计方面，采用了前后端分离的架构，确保系统具备良好的可维护性和扩展性。前端使用Vue框架实现用户界面，后端采用Spring Boot框架处理业务逻辑和数据存储。数据存储方面，使用MySQL数据库进行持久化存储，以确保数据安全性和可靠性。

系统的最终目的是提供一个用户友好的界面和简化的操作流程，帮助用户快速准确地完成仓库管理任务。通过系统的自动化管理和信息化处理，能够减少人为错误和操作失误，提高工作效率和精确度。同时，系统还提供操作日志查询功能，便于管理员进行审计和追踪操作记录，增强仓库管理的透明度和安全性。

总而言之，该仓库管理系统的设计初衷是为了提升仓库管理的效率和精确度，解决传统仓库管理中的问题。通过引入计算机技术和自动化管理，该系统能够帮助企业实现更高效、便捷和安全的仓库管理。

**关键字：仓库管理 Java Vue.js 互联网技术**

**Abstract**

The original intention of this system is to address the pain points and issues in traditional warehouse management. Manual management in traditional approaches often leads to inventory errors, difficulties in information retrieval, and inconvenient operations. The goal of this system is to optimize the warehouse management process, enhance accuracy and efficiency by introducing computer technology and automation.

The system is divided into three roles: super administrator, warehouse administrator, and ordinary user, each providing different functions and permissions. The super administrator has comprehensive management authority, including managing administrator and user information, warehouses, and item categories. Warehouse administrators have the privilege to manage warehouses and items, including the ability to carry out inbound and outbound operations. Ordinary users can access and manage their personal information, view item details, and access operation logs.

In terms of system design, a front-end and back-end separation architecture is adopted to ensure good maintainability and scalability. The user interface is implemented using the Vue framework, while the Spring Boot framework manages business logic and data storage on the back-end. The MySQL database is used for persistent storage to ensure data security and reliability.

The ultimate goal of the system is to provide a user-friendly interface and simplified operation process, helping users to complete warehouse management tasks quickly and accurately. By automating management and processing information, the system reduces human errors and operational mistakes, and improves work efficiency and accuracy. Additionally, the system offers an operation log query function, facilitating auditing and tracking of operation records, enhancing transparency and security in warehouse management.

In conclusion, the design of this warehouse management system aims to improve efficiency and accuracy in warehouse management and address the issues present in traditional approaches. By introducing computer technology and automation, the system assists companies in achieving more efficient, convenient, and secure warehouse management.

**Keywords: Warehouse management, Java, Vue.js, Internet technology**

目 录

[第1章 绪论 4](#_Toc150617681)

[1.1 项目背景 4](#_Toc150617682)

[1.2 任务分解 4](#_Toc150617683)

[1.3 设计报告章节分布 5](#_Toc150617684)

[第2章 相关技术和开发环境 6](#_Toc150617685)

[2.1 相关技术 6](#_Toc150617686)

[2.1.1 Java语言 6](#_Toc150617687)

[2.1.2 HTML、CSS、JavaScript 6](#_Toc150617688)

[2.1.3 MySQL 7](#_Toc150617689)

[2.1.4 SSM框架 7](#_Toc150617690)

[2.1.5 Vue.js 8](#_Toc150617691)

[2.1.6 SpringBoot 8](#_Toc150617692)

[2.2 开发环境 8](#_Toc150617693)

[第3章 系统分析 9](#_Toc150617694)

[3.1 可行性分析 9](#_Toc150617695)

[3.1.1 经济可行性 9](#_Toc150617696)

[3.1.2 技术可行性 9](#_Toc150617697)

[3.1.3 运行可行性 9](#_Toc150617698)

[3.2 业务流程分析 9](#_Toc150617699)

[3.2.1 业务流程分析 9](#_Toc150617700)

[3.2.2流程图 11](#_Toc150617701)

[3.2.2.1 数据流程图 11](#_Toc150617702)

[3.2.2.2 业务流程图 12](#_Toc150617703)

[3.3 需求分析 12](#_Toc150617704)

[3.3.1 需求概述 13](#_Toc150617705)

[3.3.1.1 系统管理员角色需求概述 13](#_Toc150617706)

[3.3.1.2 仓库管理员角色需求概述 13](#_Toc150617707)

[3.3.1.3 普通用户角色需求概述 13](#_Toc150617708)

[3.3.2 用例图 14](#_Toc150617709)

[3.3.3 系统类图 18](#_Toc150617710)

[3.3.4 数据需求分析 19](#_Toc150617711)

[3.3.4.1 实体集分析 19](#_Toc150617712)

[3.3.4.2 实体集数据项分析 19](#_Toc150617713)

[3.3.2.3 数据操作分析 19](#_Toc150617714)

[第4章 概要设计 20](#_Toc150617715)

[4.1 系统设计 20](#_Toc150617716)

[4.1.1 系统架构设计 20](#_Toc150617717)

[4.1.2 系统功能模块图 20](#_Toc150617718)

[4.2 数据库概要设计 21](#_Toc150617719)

[4.2.1 ER图 21](#_Toc150617720)

[4.2.2 数据库表设计 21](#_Toc150617721)

[第5章 详细设计 24](#_Toc150617722)

[5.1 详细功能模块 24](#_Toc150617723)

[5.1.1 主要模块 - 用户登录 25](#_Toc150617724)

[5.1.2 主要模块 - 商品出入库 26](#_Toc150617725)

[第6章 系统实现 27](#_Toc150617726)

[6.1 后端接口代码实现 27](#_Toc150617727)

[6.1.1 物品管理接口 27](#_Toc150617728)

[6.1.2 物品分类管理接口 28](#_Toc150617729)

[6.1.3 菜单管理接口 30](#_Toc150617730)

[6.1.4 操作日志接口 30](#_Toc150617731)

[6.1.5 仓库管理接口 32](#_Toc150617732)

[6.1.6 用户管理接口 33](#_Toc150617733)

[6.2 前端页面代码实现 37](#_Toc150617734)

[6.2.1 登录模块 38](#_Toc150617735)

[6.2.2 个人中心模块 40](#_Toc150617736)

[6.2.3 管理员管理模块 42](#_Toc150617737)

[6.2.4 用户信息管理模块 47](#_Toc150617738)

[6.2.5 仓库信息管理模块 52](#_Toc150617739)

[6.2.6 物品分类管理模块 55](#_Toc150617740)

[6.2.7 物品信息管理模块 58](#_Toc150617741)

[6.2.8 操作日志模块 66](#_Toc150617742)

[第7章 系统测试 69](#_Toc150617743)

[7.1 软硬件环境 69](#_Toc150617744)

[7.2 测试方法 69](#_Toc150617745)

[7.3 测试用例 69](#_Toc150617746)

[7.4 结果及分析 70](#_Toc150617747)

[第8章 总结与展望 72](#_Toc150617748)

[8.1 总结 72](#_Toc150617749)

[8.2 展望 72](#_Toc150617750)

[参考文献 74](#_Toc150617751)

[致谢 75](#_Toc150617752)

# 第1章 绪论

# 1.1 项目背景

随着全球经济的不断发展和市场竞争的加剧，企业对于仓储和物流管理的要求日益提高。传统的仓储管理方式往往存在一些问题，如库存管理不精确、货物追踪困难、物流流程繁琐等，严重影响了企业的运营效率和客户满意度。

此外电子商务的兴起和全球贸易的增长，仓储操作的规模和复杂性不断增加。准确管理库存水平、确保准时履行订单、优化存储能力等成为了满足客户期望和保持竞争力的关键。传统的手工仓储管理方法不仅耗时，而且容易出错，导致运营效率低下和客户不满。

我们的项目旨在利用计算机技术的能力，减少人为干预，降低错误率，提高仓储管理的整体效率。通过自动化和数字化，实现实时追踪库存、高效处理订单以及与销售和采购等其他系统的无缝集成。

此外，系统的功能设计广泛考虑了仓储环境中的不同角色需求。超级管理员、仓库管理员和普通用户等角色的定义，保证了适当的访问控制和权限管理，实现仓库人员之间的高效协作、安全数据处理和流程优化。

考虑到数据完整性和安全性的重要性，我们选择采用MySQL数据库进行持久化存储。这样可以确保关键信息如库存记录、客户详细信息和交易历史的安全可靠存储。同时，系统实施了用户身份验证和访问控制等安全措施，保障敏感数据的安全，防止未经授权的访问。

总而言之，该项目旨在开发一个现代化的仓储管理系统，解决传统仓储管理方法中存在的问题。通过利用技术、自动化流程和提升数据安全性，该系统将帮助企业高效管理库存、优化仓储运营，以适应竞争激烈的市场需求。

# 1.2 任务分解

本课题根据前期的问卷调查以及实地考察，作出了适合实际的需求分析，采用了前后端分离的思想，后端使用了SpringBoot和SpringMVC框架，前端使用了Vue.js和ElementUI进行页面的搭建，并且使用了MySQL数据库进行数据交互。整个系统的使用者有3种角色，分别是系统管理员、仓库管理员和普通用户。

以下是三种不同的角色所分别拥有的功能：

1. 系统管理员：拥有全面的管理权限，包括管理员和用户信息的管理，仓库、物品和物品分类的管理，以及所有用户的操作日志查询。
2. 仓库管理员：负责仓库的管理，包括仓库、物品和物品分类的管理，以及所有用户的操作日志查询。
3. 普通用户：可以查询和管理个人信息，查询物品信息，以及查看自己的操作日志。

# 1.3 设计报告章节分布

本文主要设计和实现一个基于JavaEE的仓库管理系统。

以下罗列出了本论文的章节安排:

第1章 绪论：简明扼要地介绍了本课题的研究背景和意义，阐明了课题的主要任务，最后陈列出了本论文的章节安排。

第2章 相关技术和开发环境：本章介绍了系统用到的相关技术和开发环境。

第3章 系统分析：从经济、技术、运行三个方面介绍了系统可行性，包括业务流程分析和需求分析。

第4章 概要设计：介绍了本系统的设计，包括架构、功能、数据库以及接口的设计。

第5章 详细设计：对系统的主要功能以及数据库的设计进行了详细介绍。

第6章 系统实现：展示了系统的主要功能代码和主要界面截图。

第7章 系统测试：简要介绍了系统的测试方法和测试环境，编写了主要测试用例，并分析记录了测试结果。

第8章 总结与展望：对本次的课题设计进行了总结，展望该系统仍需完善的功能。

# 第2章 相关技术和开发环境

# 2.1 相关技术

本课题在Windows系统下使用了JavaEE开发技术、Spring Boot技术、Vue.js技术、Tomcat服务器、MySQL关系型数据库以及ElementUI前端框架技术进行开发。以下是关键技术及开发工具介绍。

## 2.1.1 Java语言

Java是一种面向对象的编程语言，由Sun Microsystems（现在是Oracle公司）于1995年首次发布。它具有可移植性、安全性和跨平台等特性，因此被广泛应用于各个领域的软件开发中。

Java的平台无关性是其最重要的特性之一。通过Java虚拟机（JVM），Java程序可以在不同的操作系统和硬件上运行，而不需要针对特定平台进行修改。这意味着一次编写的Java代码可以在Windows、Linux、Mac等多个平台上运行。

Java是一种面向对象的语言，它支持封装、继承和多态等面向对象的编程概念。这使得Java程序更易于理解、扩展和维护。它还提供了强大的异常处理机制，使开发者能够更好地应对错误和异常情况。

Java还具有丰富的标准库和第三方库，以及一个庞大的开发者社区。这些库提供了许多常用的功能和工具，使得Java的开发变得更加高效和便捷。

总之，Java是一种功能强大、可靠性高、安全性好的编程语言，非常适合开发各种类型的应用程序，包括桌面应用程序、Web应用程序、移动应用程序等。它的广泛应用和稳定性使得Java成为了当今世界上最流行的编程语言之一。

## 2.1.2 HTML、CSS、JavaScript

HTML（超文本标记语言）是一种用于创建和组织网页内容的标记语言。它由一系列标签组成，标签用于标记和描述文档中的不同部分，如标题、段落、链接和图像等。HTML提供了结构化的标记方式，使得浏览器能够正确地解析并显示网页内容。

CSS（层叠样式表）是一种用于描述网页外观和布局的样式语言。通过CSS，可以对HTML文档中的元素进行样式设置，包括字体、颜色、大小、间距、边框和背景等。CSS的设计目标是将样式与内容分离，使得网页的样式可以独立于内容进行修改和调整，从而提高网页的可维护性和灵活性。

JavaScript是一种动态的、基于对象和事件驱动的脚本语言。它广泛用于前端开发，用于为网页增加交互性和动态性。通过JavaScript，可以操作网页的元素、处理用户的输入、响应事件、进行表单验证和数据处理等。它还可以通过AJAX技术与服务器进行数据交互，实现动态加载内容和实时更新。

HTML、CSS和JavaScript三者通常一起使用，被称为前端开发技术。HTML负责网页的内容和结构，CSS负责网页的样式和布局，JavaScript负责网页的交互和动态效果。它们共同作用于网页的不同方面，使网页成为一个功能丰富、外观吸引人且易于使用的用户界面。

总结起来，HTML提供了网页内容的结构和标记，CSS负责网页的样式和布局，JavaScript为网页增加了交互性和动态功能。三者的协同工作使得构建现代化、交互式的网页成为可能。

## 2.1.3 MySQL

MySQL是一种关系型数据库管理系统，是一个多用户，多线程的SQL数据库。MySQL的优点是体积小、速度快、总体拥有成本低，开放源代码等，使得它成为了目前中小企业的最爱。

## 2.1.4 SSM框架

SSM框架是由Spring，SpringMVC和MyBatis三个开源框架整合而成，是当前较为流行稳定的Web项目框架。

Spring是一个开源框架，它是为了解决企业应用开发的复杂性而创建的。Spring的核心是控制反转和面向切面编程。

SpringMVC是Spring框架的组成部分之一，MVC框架提供了模型-视图-控制的体系结构和可以用来开发灵活、松散耦合的Web应用程序的组件。

MyBatis是一个基于Java的持久层框架,它支持定制化SQL、存储过程以及高级映射[[1]](#footnote-1)[3]。该框架解除了SQL与程序代码的耦合，提高了代码的可维护性。同时，MyBatis较为灵活，简单易学以及支持动态SQL语句等优点，让它深受广大编程人员的喜爱。

## 2.1.5 Vue.js

Vue.js是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue被设计为可以自底向上逐层应用。Vue的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。

## 2.1.6 SpringBoot

SpringBoot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。使用SpringBoot的最大好处就是简化配置，它实现了自动化配置。

# 2.2 开发环境

以下展示了系统所需的开发环境。

系统所需开发环境表

|  |  |
| --- | --- |
| 操作系统 | Windows 10 |
| 开发工具 | IntelliJ IDEA 2021.03 |
| Java环境 | JDK1.8，Maven 3.8.1 |
| 数据库 | MySQL 8.0.19、Redis 3.0.10 |
| 服务器 | Tomcat 9.0.30 |

# 第3章 系统分析

# 3.1 可行性分析

## 3.1.1 经济可行性

本系统为本科毕业设计，未真正投入到实际生活中，所需要的硬件设备目前只需用到一部电脑，软件方面大部分为官方提高免费使用的，所需资料也可在网上或者查询相关书籍即可找到。因此，本系统开发在经济方面是可行的。

## 3.1.2 技术可行性

本系统的主要开发语言为Java，基于目前较为流行的SpringBoot以及Vue.js框架进行编写。本系统采用了MySQL数据库作为数据存储基础阵地，保证了一个安全、稳定的数据使用环境。本系统使用的技术，基本上都是目前较为流行的并且是开源的技术，文档和资料在互联网上都非常健全，且有许多开发成功的项目。因此，本系统开发在技术方面是可行的。

## 3.1.3 运行可行性

本系统基于SpringBoot框架，需要搭建Java，Maven， MySQL等软件环境。在搭建的过程中，有遇到过种种的问题，但通过查询资料和请教导师，都能一一解决，并能熟练掌握。因此，本系统开发在运行方面是可行的。

# 3.2 业务流程分析

## 3.2.1 业务流程分析

本系统的使用者一共有系统管理员、仓库管理员和普通用户这3种角色:

1. 系统管理员：通过登录系统后，可以进行管理员和用户信息的管理、仓库和物品分类的管理，以及操作日志的查询，具有全面的系统管理权限。
2. 仓库管理员：登录系统后，可以进行仓库和物品分类的管理，同时也能够查询所有用户的操作日志，以便了解仓库的出入库情况和操作记录。
3. 普通用户：登录系统后，可以查询和管理个人信息，同时也可以查询物品的详细信息，以及查看自己的操作日志，帮助用户了解自己的出入库操作记录和历史纪录。

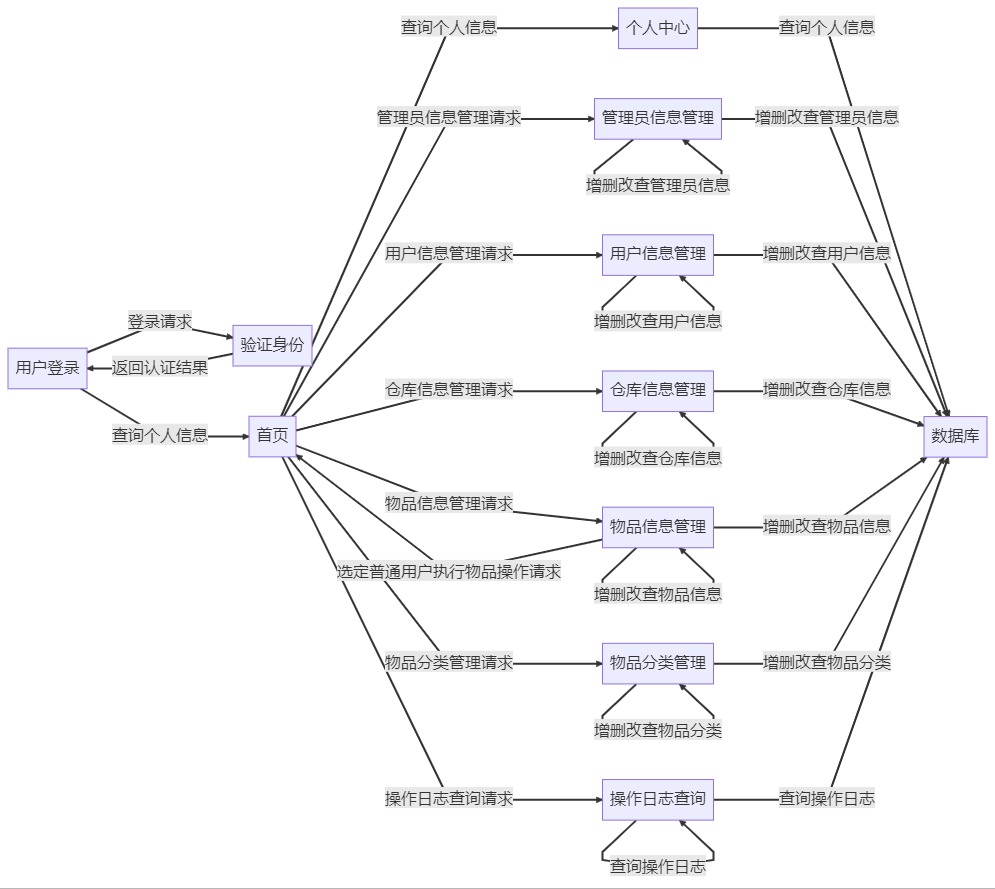
本系统的业务需求主要是用户登录与权限管理、个人信息管理、仓库信息管理、物品分类管理、物品信息查询、操作日志查询等。

业务需求表

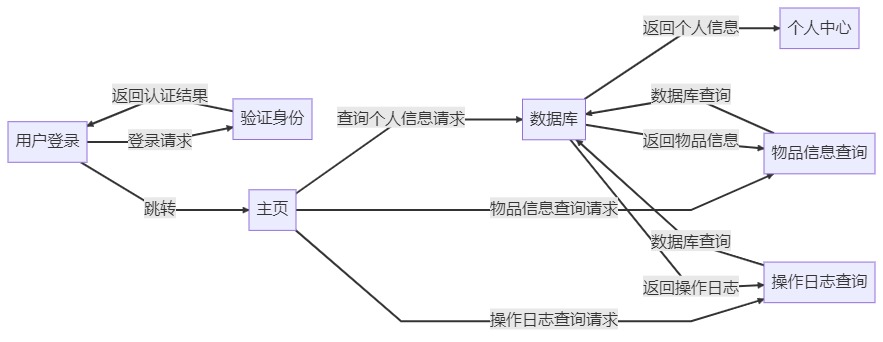
| **编号** | **业务名称** | **操作者** |
| --- | --- | --- |
| 1 | 登录功能 | 所有角色 |
| 2 | 个人中心 | 超级管理员、仓库管理员、普通用户 |
| 3 | 管理员信息管理 | 超级管理员 |
| 4 | 用户信息管理 | 超级管理员 |
| 5 | 仓库信息管理 | 超级管理员、仓库管理员 |
| 6 | 物品信息管理 | 超级管理员、仓库管理员 |
| 7 | 物品出入库操作 | 超级管理员、仓库管理员 |
| 8 | 物品分类管理 | 超级管理员 |
| 9 | 操作日志查询 | 超级管理员、仓库管理员、普通用户 |

## 3.2.2流程图

### 3.2.2.1 数据流程图

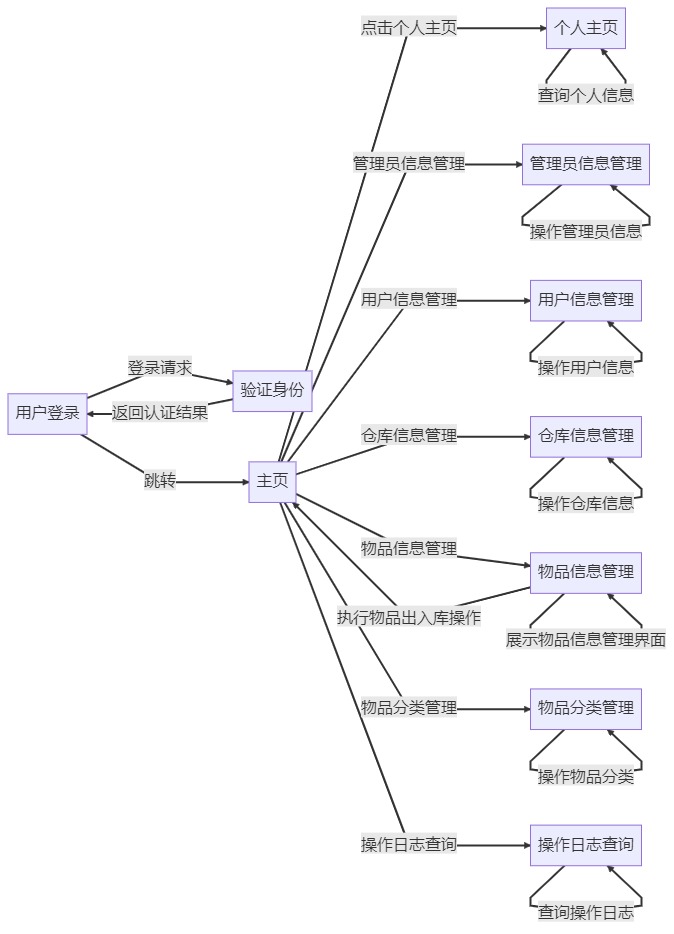


管理员数据流程图

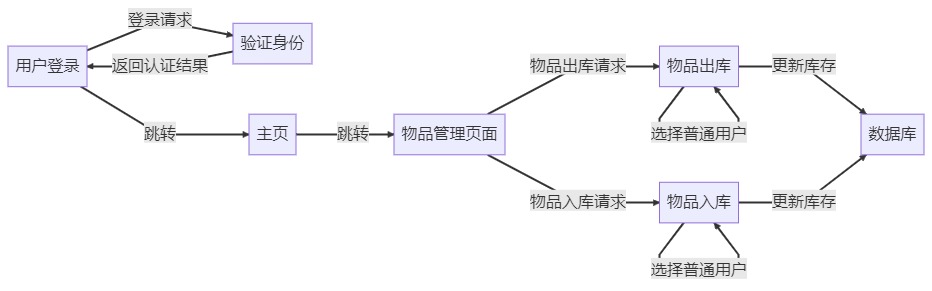


普通用户数据流程图

### 3.2.2.2 业务流程图



管理员业务流程图



物品出入库业务流程图

# 3.3 需求分析

## 3.3.1 需求概述

### 3.3.1.1 系统管理员角色需求概述

系统管理员是仓库管理系统中至关重要的角色。作为系统管理员，他们拥有全面的管理权限，负责管理管理员和用户信息、仓库信息、物品信息及分类，并查看所有用户的出入库操作日志。系统管理员需要具备登录功能、个人中心（查询个人信息）、管理员信息管理、用户信息管理、仓库信息管理、物品信息管理、物品分类管理和操作日志查询等能力。他们的主要任务是确保系统的安全性和正常运行，并对系统进行必要的维护和管理。通过系统管理员的操作，保证了仓库管理系统的高效运作与稳定性。

### 3.3.1.2 仓库管理员角色需求概述

仓库管理员是仓库管理系统中的关键角色之一。作为仓库管理员，他们负责具体的仓库管理任务，包括对仓库和物品进行管理、执行出入库操作，并查看所有用户的出入库操作日志。与系统管理员相比，仓库管理员无需进行管理员信息管理，但其他功能与系统管理员完全一致。

仓库管理员需要具备登录功能、个人中心（查询个人信息）、用户信息管理、仓库信息管理、物品信息管理、物品分类管理和操作日志查询等能力。他们需要熟悉系统的操作流程，准确执行出入库操作，并对所管理的仓库和物品进行有效的管理和调度。仓库管理员的主要任务是保证仓库的日常运营，确保物品的准确入库和出库，并维护良好的仓库记录和操作日志。通过仓库管理员的工作，提高了仓库管理的效率与精确度，进而推动整个仓库管理系统的顺利运行。

### 3.3.1.3 普通用户角色需求概述

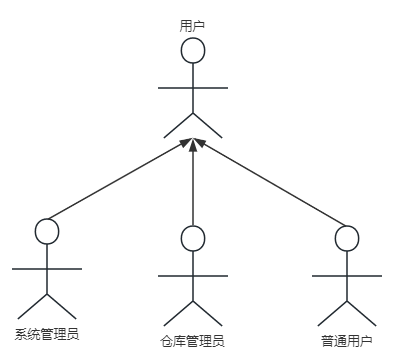
普通用户在仓库管理系统中扮演重要的角色之一。作为普通用户，他们拥有基本的功能和权限，主要关注个人信息查询、物品信息查询以及操作日志查询。

普通用户需要具备登录功能，通过用户名和密码进行系统登录。一旦登录成功，普通用户可以访问个人中心，查询和管理个人信息。对于物品，普通用户可以进行物品信息的查询，以便了解物品的详细信息和可用性等。此外，普通用户还可以查询与自己相关的出入库操作日志，以便跟踪个人的操作记录和历史。

普通用户的主要任务是利用系统提供的功能，方便地查询和管理个人信息、了解物品信息，并跟踪自己的操作日志。通过这些功能，普通用户可以更好地了解自己的仓库情况，并在需要时进行相应的操作。普通用户的参与和合作，为仓库管理系统的使用和数据记录提供了重要的补充。

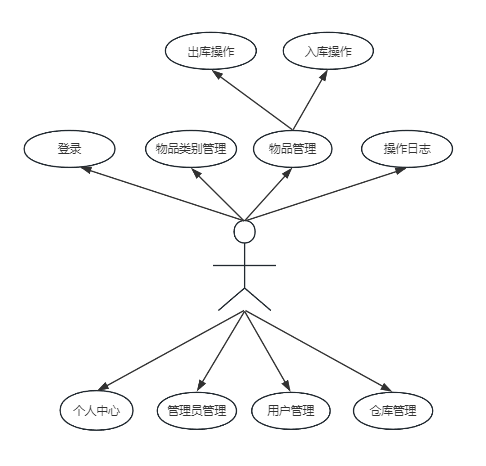
## 3.3.2 用例图

根据本系统的使用者将用户划分为三种角色，分别是系统管理员、仓库管理员和普通用户。进入系统前需要通过账号以及密码进行登录，登录成功后则根据他们的各自的角色进入相应的页面。



用户关系用例图

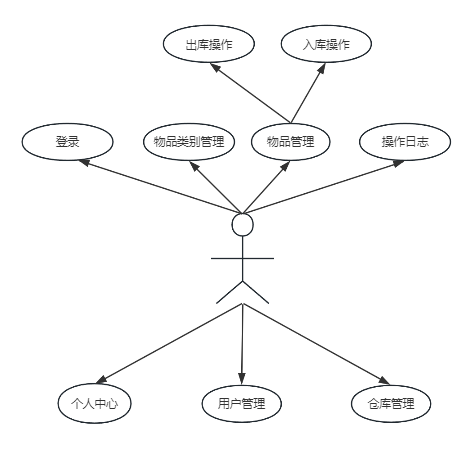
系统管理员进入网站的页面后，可以进行的功能用例如图所示:



系统管理员用例图

1. 管理员信息管理：管理员可以通过关键字搜索、分页展示管理员信息，也可以进行添加、删除和修改管理员信息。
2. 用户信息管理：管理员可以通过关键字搜索、分页展示用户信息，也可以进行添加、删除和修改用户信息。
3. 仓库信息管理：管理员可以通过关键字搜索、分页展示仓库的信息，也可以进行添加、删除和修改仓库的信息。
4. 物品信息管理：管理员可以通过关键字搜索、分页展示物品的信息，也可以进行添加、删除和修改物品的信息。
5. 物品出库操作：管理员可以选择对应的操作用户执行出库操作。
6. 物品入库操作：管理员可以选择对应的操作用户执行入库操作。
7. 物品分类管理：管理员可以通过关键字搜索、分页展示物品分类的信息，也可以进行添加、删除和修改物品分类。
8. 操作日志查询：管理员可以根据时间范围、操作类型等关键字进行操作日志的查询，以便审计和追踪操作记录。

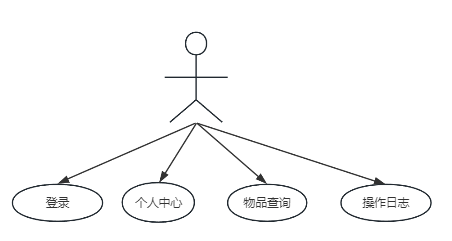
仓库管理员进入网站的页面后，可以进行的功能用例如图所示:



仓库管理员用例图

1. 用户信息管理：管理员可以通过关键字搜索、分页展示用户信息，也可以进行添加、删除和修改用户信息。
2. 仓库信息管理：管理员可以通过关键字搜索、分页展示仓库的信息，也可以进行添加、删除和修改仓库的信息。
3. 物品信息管理：管理员可以通过关键字搜索、分页展示物品的信息，也可以进行添加、删除和修改物品的信息。
4. 物品出库操作：管理员可以选择对应的操作用户执行出库操作。
5. 物品入库操作：管理员可以选择对应的操作用户执行入库操作。
6. 物品分类管理：管理员可以通过关键字搜索、分页展示物品分类的信息，也可以进行添加、删除和修改物品分类。
7. 操作日志查询：管理员可以根据时间范围、操作类型等关键字进行操作日志的查询，以便审计和追踪操作记录。

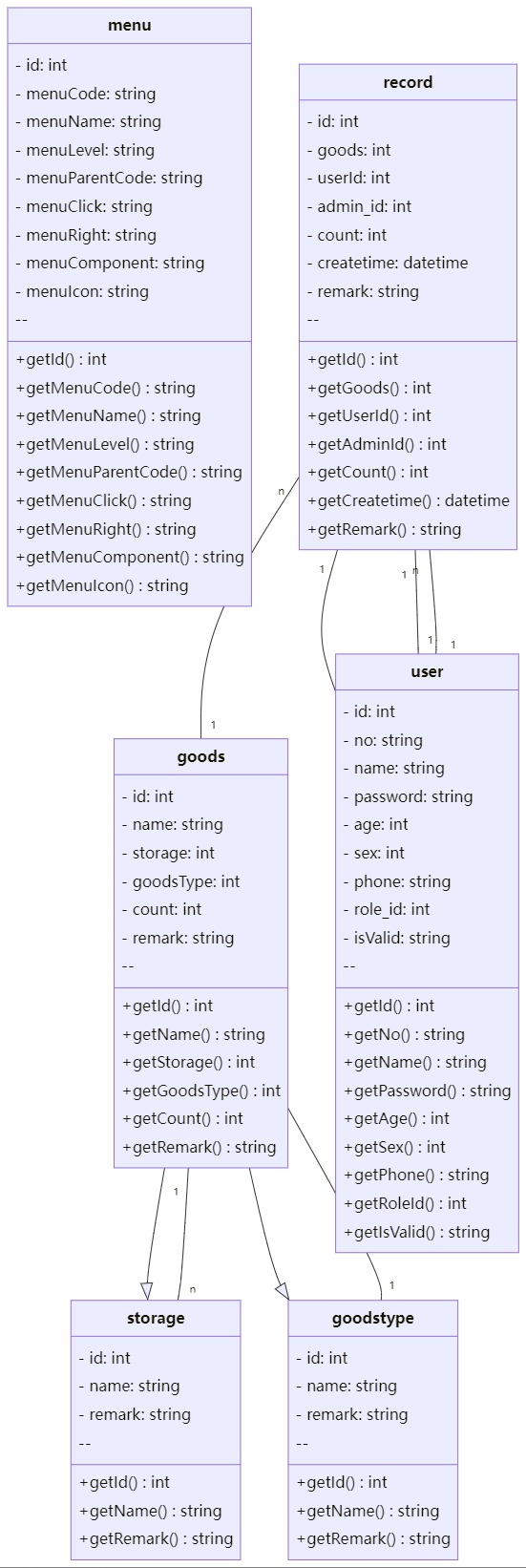
普通用户进入网站的页面后，可以进行的功能用例如图所示:



普通用户用例图

1. 查询个人信息：用户可以查看和管理自己的个人信息，包括姓名、联系方式等。
2. 物品信息查询：用户可以通过关键字搜索、分页展示物品的信息，以便了解物品的名称、价格、库存等。
3. 操作日志查询：用户可以查询和查看自己的出入库操作日志，了解自己的操作记录和历史。

## 3.3.3 系统类图



## 3.3.4 数据需求分析

### 3.3.4.1 实体集分析

实体集是具有相同类型及相同属性的实体的集合。本系统主要的实体集主要有：管理员信息，用户信息，仓库信息，物品信息，物品类别信息，操作日志信息等。

### 3.3.4.2 实体集数据项分析

管理员信息：管理员ID，账号，名字，密码，角色，是否有效；

用户信息：用户ID，账号，名字，密码，年龄，性别，电话，角色，是否有效；

仓库信息：仓库ID，仓库名，备注；

物品信息：物品ID，货名，仓库，分类，数量，备注；

物品类别信息：类别ID，分类名，备注；

操作日志信息：日志ID，物品ID，取货人/补货人，操作人ID，数量，操作时间，备注。

### 3.3.2.3 数据操作分析

超级管理员角色数据操作分析： 超级管理员可以进行管理员信息管理、用户信息管理、仓库信息管理、物品信息管理、物品分类管理和操作日志查询等操作，包括增删改查各个实体集的数据，以及查询所有用户的出入库操作日志。

仓库管理员角色数据操作分析： 仓库管理员拥有和超级管理员相似的权限，可以进行用户信息管理、仓库信息管理、物品信息管理、物品分类管理和操作日志查询等操作，但不具备管理员信息管理的功能，只能查询和操作与其所管理的仓库相关的数据。

普通用户角色数据操作分析： 普通用户具有较少的功能，可以查询个人信息、物品信息和操作日志，但只能查询自己的出入库操作日志，无法对其他实体集进行数据操作。

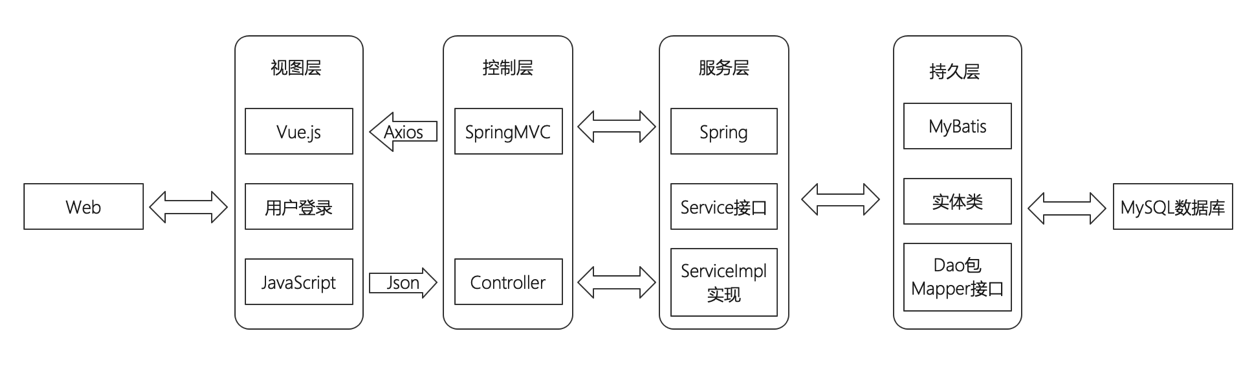
根据角色的不同权限和功能需求，每个角色在仓库管理系统中有不同的数据操作能力和限制。

# 第4章 概要设计

# 4.1 系统设计

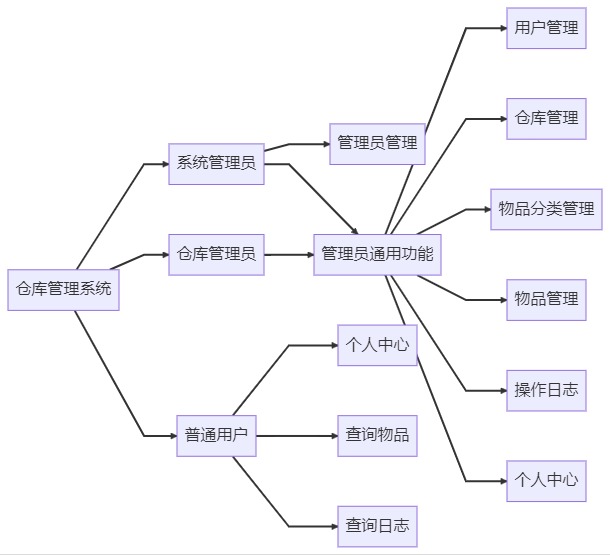
## 4.1.1 系统架构设计

本系统使用SpringBoot、Vue.js等主流开发框架进行开发，遵守MVC模式，同时使用到了Element-UI进行页面的设计。为方便前后端进行整合和日后系统的维护，本系统分为视图层、控制层、业务层和持久层。各层之间相互独立，通过之间的接口进行通信，高内聚，低耦合[[2]](#footnote-2)[6]。



系统架构图

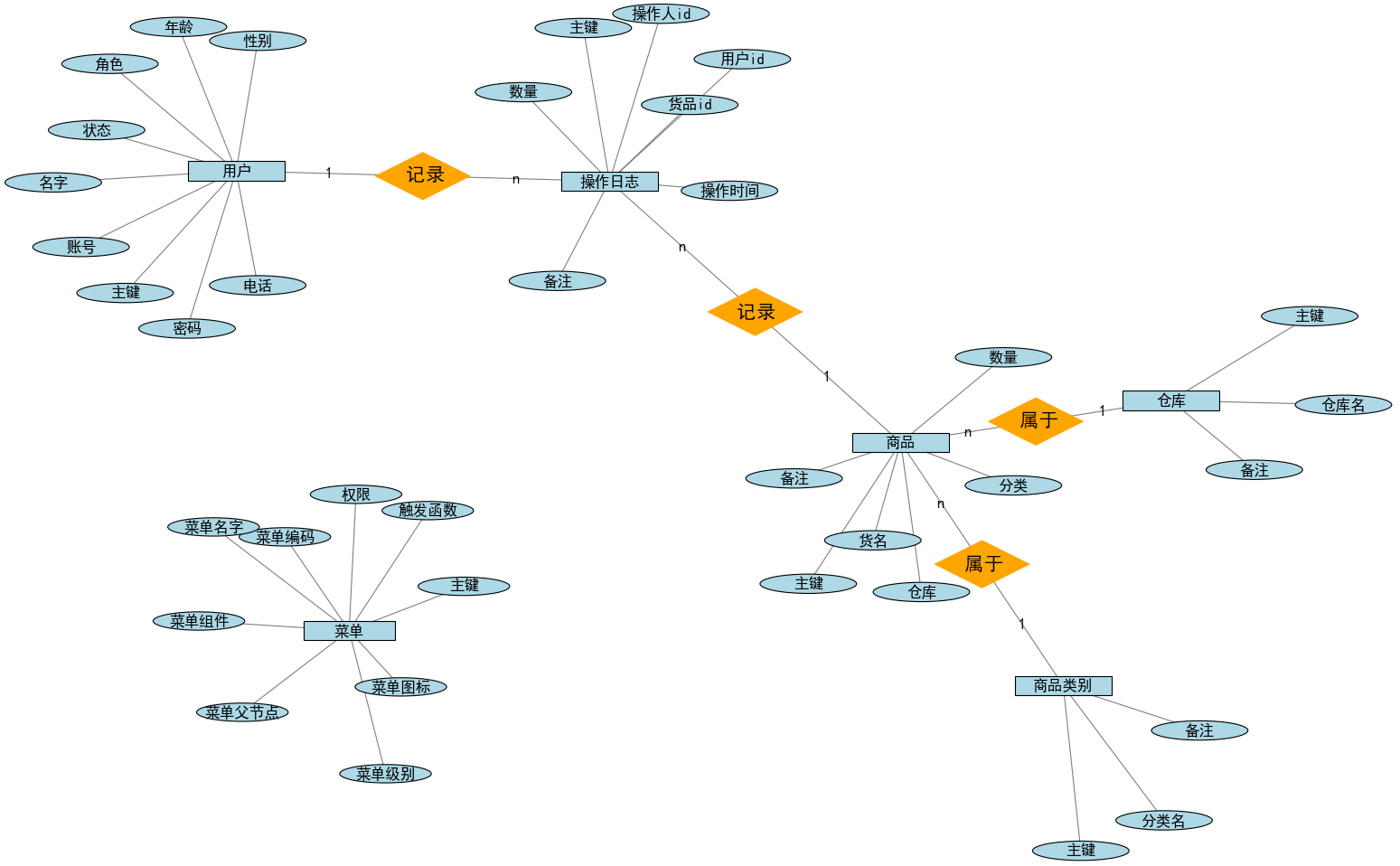
## 4.1.2 系统功能模块图



系统功能模块图

# 4.2 数据库概要设计

## 4.2.1 ER图



## 4.2.2 数据库表设计

1、表goods

| **字段名称** | **字段类型** | **中文名称** |
| --- | --- | --- |
| id | int | 主键 |
| name | varchar | 货名 |
| storage | int | 仓库 |
| goodsType | int | 分类 |
| count | int | 数量 |
| remark | varchar | 备注 |

2、表goodstype

| **字段名称** | **字段类型** | **中文名称** |
| --- | --- | --- |
| id | int | 主键 |
| name | varchar | 分类名 |
| remark | varchar | 备注 |

3、表menu

| **字段名称** | **字段类型** | **中文名称** |
| --- | --- | --- |
| id | int | 主键 |
| menuCode | varchar | 菜单编码 |
| menuName | varchar | 菜单名字 |
| menuLevel | varchar | 菜单级别 |
| menuParentCode | varchar | 菜单父节点 |
| menuClick | varchar | 触发函数 |
| menuRight | varchar | 权限 |
| menuComponent | varchar | 菜单组件 |
| menuIcon | varchar | 菜单图标 |

4、表record

| **字段名称** | **字段类型** | **中文名称** |
| --- | --- | --- |
| id | int | 主键 |
| goods | int | 货品id |
| userId | int | 用户id |
| admin\_id | int | 操作人id |
| count | int | 数量 |
| createtime | datetime | 操作时间 |
| remark | varchar | 备注 |

5、表storage

| **字段名称** | **字段类型** | **中文名称** |
| --- | --- | --- |
| id | int | 主键 |
| name | varchar | 仓库名 |
| remark | varchar | 备注 |

6、表user

| **字段名称** | **字段类型** | **中文名称** |
| --- | --- | --- |
| id | int | 主键 |
| no | varchar | 账号 |
| name | varchar | 名字 |
| password | varchar | 密码 |
| age | int | 年龄 |
| sex | int | 性别 |
| phone | varchar | 电话 |
| role\_id | int | 角色 |
| isValid | varchar | 状态 |

# 第5章 详细设计

# 5.1 详细功能模块

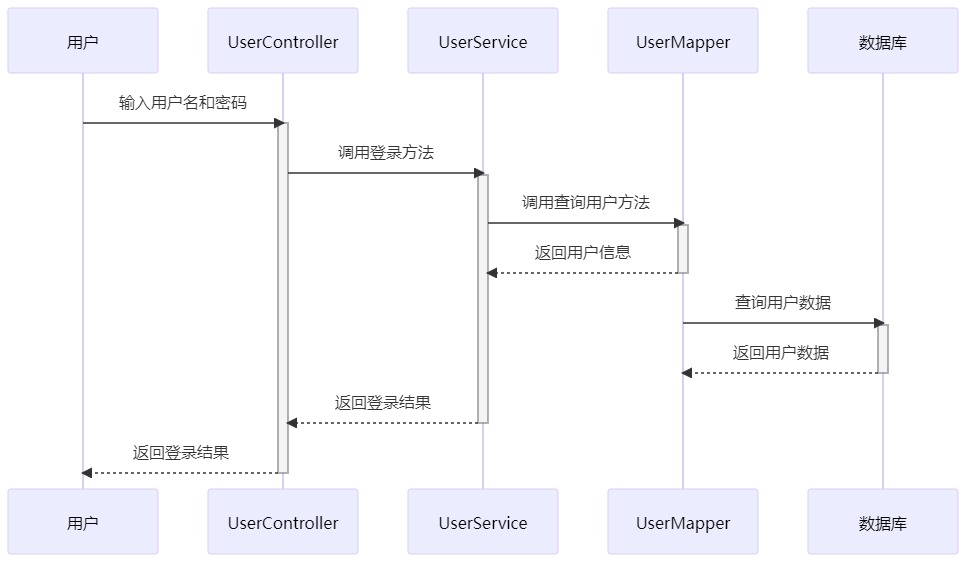
系统功能模块如表所示:

| **功能名称** | **输入** | **处理** | **输出** |
| --- | --- | --- | --- |
| 用户登录 | 用户名、密码 | 验证用户名和密码，校验成功则登录，并记录登录日志 | 登录成功，跳转到管理员首页 |
| 个人中心查询 | 无 | 查询当前用户的个人信息 | 显示当前用户的个人信息 |
| 用户信息管理 | 输入关键字查询用户信息，提供增加、删除、修改、查询等操作 | 1. 增加用户信息：将新增的用户信息插入到数据库 2. 删除用户信息：从数据库中删除对应的用户信息 3. 修改用户信息：更新数据库中对应的用户信息 | 保存操作结果，提示操作成功或失败 |
| 仓库信息管理 | 输入关键字查询仓库信息，提供增加、删除、修改、查询等操作 | 1. 增加仓库信息：将新增的仓库信息插入到数据库 2. 删除仓库信息：从数据库中删除对应的仓库信息 3. 修改仓库信息：更新数据库中对应的仓库信息 | 保存操作结果，提示操作成功或失败 |
| 物品信息管理 | 输入关键字查询物品信息，提供增加、删除、修改、查询等操作 | 1. 增加物品信息：将新增的物品信息插入到数据库 2. 删除物品信息：从数据库中删除对应的物品信息 3. 修改物品信息：更新数据库中对应的物品信息 | 保存操作结果，提示操作成功或失败 |
| 物品分类管理 | 输入关键字查询物品分类，提供增加、删除、修改、查询等操作 | 1. 增加物品分类：将新增的物品分类插入到数据库 2. 删除物品分类：从数据库中删除对应的物品分类 3. 修改物品分类：更新数据库中对应的物品分类 | 保存操作结果，提示操作成功或失败 |
| 物品出入库操作 | 选择物品、选择操作类型、输入数量 | 1. 根据选择的物品和操作类型进行相应的数据库插入或更新操作 2. 根据输入的数量更新物品的库存数量 | 保存操作结果，提示操作成功或失败 |
| 操作日志查询 | 选择查询条件，如时间范围、操作类型等 | 根据选择的查询条件从数据库中查询相应的操作日志 | 显示查询到的操作日志列表 |
| 管理员信息管理 | 输入关键字查询管理员信息，提供增加、删除、修改、查询等操作 | 1. 增加管理员信息：将新增的管理员信息插入到数据库 2. 删除管理员信息：从数据库中删除对应的管理员信息 3. 修改管理员信息：更新数据库中对应的管理员信息 | 保存操作结果，提示操作成功或失败 |
| 物品信息查询 | 输入关键字进行搜索 | 根据关键字从数据库中查询匹配的物品信息 | 显示查询到的物品信息 |
| 操作日志查询 | 选择查询条件，如用户、时间范围、操作类型等 | 根据选择的查询条件从数据库中查询匹配的操作日志 | 显示查询到的操作日志列表 |

系统功能模块表

## 5.1.1 主要模块 - 用户登录

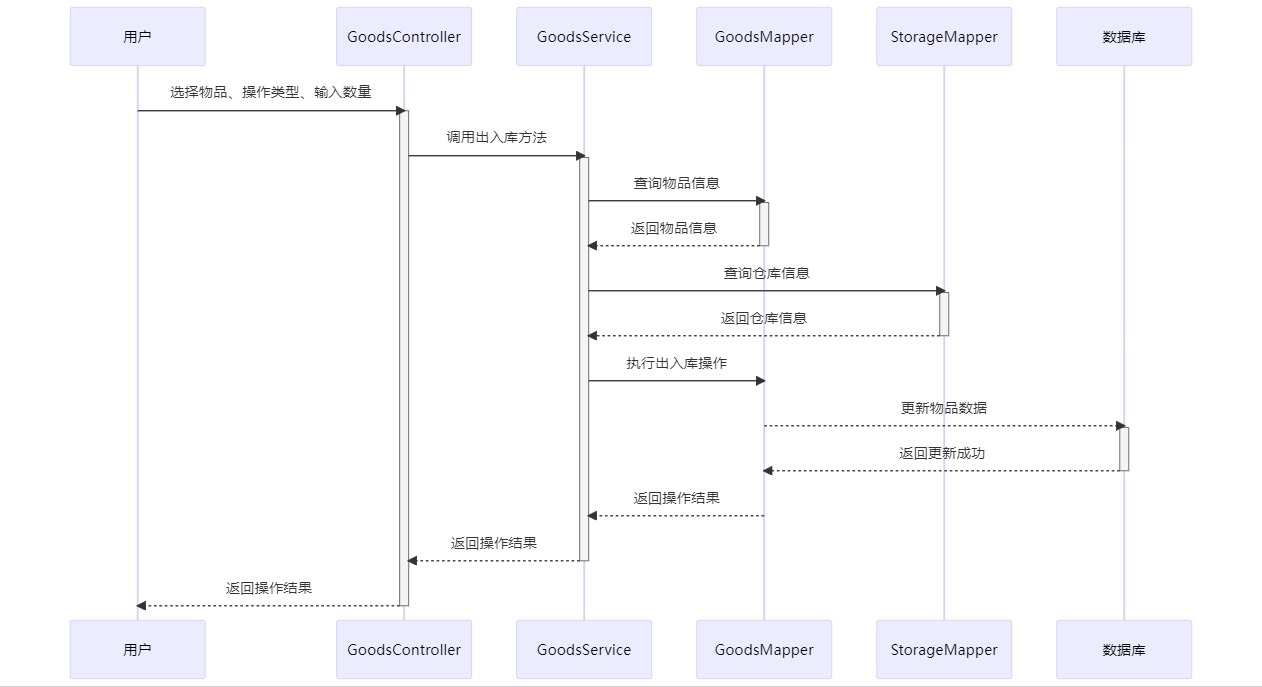
用户输入帐号和密码后，传至系统后台进行校验。如果帐号密码不匹配，跳回登录页面。账号密码都匹配的话，返回给前端，如图所示。



登录时序图

## 5.1.2 主要模块 - 商品出入库

管理员可以针对物品进行出入库操作，具体流程如图所示。



预约挂号时序图

# 第6章 系统实现

# 6.1 后端接口代码实现

## 6.1.1 物品管理接口

*/\*  
 \* 新增物品  
 \* @author rabbiter  
 \* @date 2023/1/6 12:12  
 \*/*@PostMapping("/save")  
public Result save(@RequestBody Goods goods){  
 return goodsService.save(goods)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 更新物品  
 \* @author rabbiter  
 \* @date 2023/1/6 13:22  
 \*/*@PostMapping("/update")  
public Result update(@RequestBody Goods goods){  
 return goodsService.updateById(goods)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 删除物品  
 \* @author rabbiter  
 \* @date 2023/1/6 13:24  
 \*/*@GetMapping("/del")  
public Result del(@RequestParam String id){  
 return goodsService.removeById(id)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 模糊查询：根据输入查询物品并以分页的形式展示  
 \* @author rabbiter  
 \* @date 2023/1/6 13:31  
 \*/*@PostMapping("/listPage")  
public Result listPage(@RequestBody QueryPageParam query){  
 HashMap param = query.getParam();  
 String name = (String)param.get("name");  
 String goodstype = (String)param.get("goodstype");  
 String storage = (String)param.get("storage");  
  
 Page<Goods> page = new Page();  
 page.setCurrent(query.getPageNum());  
 page.setSize(query.getPageSize());  
  
 LambdaQueryWrapper<Goods> queryWrapper = new LambdaQueryWrapper<>();  
 if(StringUtils.*isNotBlank*(name) && !"null".equals(name)){  
 queryWrapper.like(Goods::getName,name);  
 }  
 if(StringUtils.*isNotBlank*(goodstype) && !"null".equals(goodstype)){  
 queryWrapper.like(Goods::getGoodstype,goodstype);  
 }  
 if(StringUtils.*isNotBlank*(storage) && !"null".equals(storage)){  
 queryWrapper.like(Goods::getStorage,storage);  
 }  
  
 IPage result = goodsService.pageCC(page,queryWrapper);  
 return Result.*success*(result.getRecords(),result.getTotal());  
}

## 6.1.2 物品分类管理接口

*/\*  
 \* 新增物品分类  
 \* @author rabbiter  
 \* @date 2023/1/5 20:39  
 \*/*@PostMapping("/save")  
public Result save(@RequestBody Goodstype goodstype){  
 return goodstypeService.save(goodstype)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 更新物品分类  
 \* @author rabbiter  
 \* @date 2023/1/5 20:41  
 \*/*@PostMapping("/update")  
public Result update(@RequestBody Goodstype goodstype){  
 return goodstypeService.updateById(goodstype)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 删除物品分类  
 \* @author rabbiter  
 \* @date 2023/1/5 20:43  
 \*/*@GetMapping("/del")  
public Result del(@RequestParam String id){  
 return goodstypeService.removeById(id)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 查询物品分类列表  
 \* @author rabbiter  
 \* @date 2023/1/5 21:06  
 \*/*@GetMapping("/list")  
public Result list(){  
 List list = goodstypeService.list();  
 return Result.*success*(list);  
}  
  
*/\*  
 \* 模糊查询：根据输入查询物品分类并以分页的形式展示  
 \* @author rabbiter  
 \* @date 2023/1/5 21:13  
 \*/*@PostMapping("/listPage")  
public Result listPage(@RequestBody QueryPageParam query){  
 HashMap param = query.getParam();  
 String name = (String)param.get("name");  
  
 Page<Goodstype> page = new Page();  
 page.setCurrent(query.getPageNum());  
 page.setSize(query.getPageSize());  
  
 LambdaQueryWrapper<Goodstype> queryWrapper = new LambdaQueryWrapper();  
 if(StringUtils.*isNotBlank*(name) && !"null".equals(name)){  
 queryWrapper.like(Goodstype::getName,name);  
 }  
  
 IPage result = goodstypeService.pageCC(page,queryWrapper);  
 return Result.*success*(result.getRecords(),result.getTotal());  
}

## 6.1.3 菜单管理接口

*/\*  
 \* 根据用户身份获取菜单列表  
 \* @author rabbiter  
 \* @date 2023/1/3 20:48  
 \*/*@GetMapping("/list")  
public Result list(@RequestParam String roleId){  
 List list = menuService.lambdaQuery()  
 .like(Menu::getMenuright,roleId)  
 .list();  
 return Result.*success*(list);  
}

## 6.1.4 操作日志接口

*/\*  
 \*   
 \* @author rabbiter  
 \* @date 2023/1/6 20:58  
 \*/*@PostMapping("/listPage")  
public Result listPage(@RequestBody QueryPageParam query){  
 HashMap param = query.getParam();  
 String name = (String)param.get("name");  
 String goodstype = (String)param.get("goodstype");  
 String storage = (String)param.get("storage");  
 String roleId = (String)param.get("roleId");  
 String userId = (String)param.get("userId");  
  
 Page page = new Page();  
 page.setCurrent(query.getPageNum());  
 page.setSize(query.getPageSize());  
  
 QueryWrapper<Record> queryWrapper = new QueryWrapper();  
 queryWrapper.apply("a.goods=b.id and b.storage=c.id and b.goodsType=d.id ");  
  
 if("2".equals(roleId)){  
 *// queryWrapper.eq(Record::getUserid,userId);* queryWrapper.apply(" a.userId= "+userId);  
 }  
  
 if(StringUtils.*isNotBlank*(name) && !"null".equals(name)){  
 queryWrapper.like("b.name",name);  
 }  
 if(StringUtils.*isNotBlank*(goodstype) && !"null".equals(goodstype)){  
 queryWrapper.eq("d.id",goodstype);  
 }  
 if(StringUtils.*isNotBlank*(storage) && !"null".equals(storage)){  
 queryWrapper.eq("c.id",storage);  
 }  
  
 IPage result = recordService.pageCC(page,queryWrapper);  
 return Result.*success*(result.getRecords(),result.getTotal());  
}  
  
*/\*  
 \* 新增记录  
 \* @author rabbiter  
 \* @date 2023/1/6 21:21  
 \*/*@PostMapping("/save")  
public Result save(@RequestBody Record record){  
 Goods goods = goodsService.getById(record.getGoods());  
 int n = record.getCount();  
 *// 出库* if("2".equals(record.getAction())){  
 n = -n;  
 record.setCount(n);  
 }  
  
 int num = goods.getCount()+n;  
 if(num < 0) {  
 return Result.*fail*();  
 }  
 goods.setCount(num);  
 goodsService.updateById(goods);  
 record.setCreatetime(LocalDateTime.*now*());  
 return recordService.save(record)?Result.*success*():Result.*fail*();  
}

## 6.1.5 仓库管理接口

*/\*  
 \* 新增仓库  
 \* @author rabbiter  
 \* @date 2023/1/5 19:36  
 \*/*@PostMapping("/save")  
public Result save(@RequestBody Storage storage){  
 return storageService.save(storage)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 更新仓库  
 \* @author rabbiter  
 \* @date 2023/1/5 19:38  
 \*/*@PostMapping("/update")  
public Result update(@RequestBody Storage storage){  
 return storageService.updateById(storage)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 删除仓库  
 \* @author rabbiter  
 \* @date 2023/1/5 19:40  
 \*/*@GetMapping("/del")  
public Result del(@RequestParam String id){  
 return storageService.removeById(id)?Result.*success*():Result.*fail*();  
}  
  
*/\*  
 \* 查询仓库列表  
 \* @author rabbiter  
 \* @date 2023/1/5 19:42  
 \*/*@GetMapping("/list")  
public Result list(){  
 List list = storageService.list();  
 return Result.*success*(list);  
}  
  
*/\*  
 \* 模糊查询：根据输入查询仓库并以分页的形式展示  
 \* @author rabbiter  
 \* @date 2023/1/5 19:43  
 \*/*@PostMapping("/listPage")  
public Result listPage(@RequestBody QueryPageParam query){  
 HashMap param = query.getParam();  
 String name = (String)param.get("name");  
  
 Page<Storage> page = new Page();  
 page.setCurrent(query.getPageNum());  
 page.setSize(query.getPageSize());  
  
 LambdaQueryWrapper<Storage> queryWrapper = new LambdaQueryWrapper<>();  
 if(StringUtils.*isNotBlank*(name) && !"null".equals(name)){  
 queryWrapper.like(Storage::getName,name);  
 }  
  
 IPage result = storageService.pageCC(page,queryWrapper);  
 return Result.*success*(result.getRecords(),result.getTotal());  
}

## 6.1.6 用户管理接口

*/\*  
 \* 查询全部用户  
 \* @author rabbiter  
 \* @date 2023/1/2 19:26  
 \*/* @GetMapping("/list")  
 public List<User> list(){  
 return userService.list();  
 }  
  
 */\*  
 \* 根据账号查找用户  
 \* @author rabbiter  
 \* @date 2023/1/4 14:53  
 \*/* @GetMapping("/findByNo")  
 public Result findByNo(@RequestParam String no){  
 List list = userService.lambdaQuery()  
 .eq(User::getNo,no)  
 .list();  
 return list.size()>0?Result.*success*(list):Result.*fail*();  
 }  
  
 */\*  
 \* 新增用户  
 \* @author rabbiter  
 \* @date 2023/1/2 19:11  
 \*/* @PostMapping("/save")  
 public Result save(@RequestBody User user){  
 return userService.save(user)?Result.*success*():Result.*fail*();  
 }  
  
 */\*  
 \* 更新用户  
 \* @author rabbiter  
 \* @date 2023/1/2 19:11  
 \*/* @PostMapping("/update")  
 public Result update(@RequestBody User user){  
 return userService.updateById(user)?Result.*success*():Result.*fail*();  
 }  
  
 */\*  
 \* 用户登录：登录的时候一并将菜单信息也查询出来  
 \* @author rabbiter  
 \* @date 2023/1/3 14:08  
 \*/* @PostMapping("/login")  
 public Result login(@RequestBody User user){  
 *//匹配账号和密码* List list = userService.lambdaQuery()  
 .eq(User::getNo,user.getNo())  
 .eq(User::getPassword,user.getPassword())  
 .list();  
  
 if(list.size()>0){  
 User user1 = (User)list.get(0);  
 List<Menu> menuList = menuService.lambdaQuery()  
 .like(Menu::getMenuright,user1.getRoleId())  
 .list();  
 HashMap res = new HashMap();  
 res.put("user",user1);  
 res.put("menu",menuList);  
 return Result.*success*(res);  
 }  
 return Result.*fail*();  
 }  
  
 */\*  
 \* 修改用户  
 \* @author rabbiter  
 \* @date 2023/1/4 15:02  
 \*/* @PostMapping("/mod")  
 public boolean mod(@RequestBody User user){  
 return userService.updateById(user);  
 }  
   
 */\*  
 \* 新增或修改：存在用户则修改，否则新增用户  
 \* @author rabbiter  
 \* @date 2023/1/2 19:12  
 \*/* @PostMapping("/saveOrUpdate")  
 public Result saveOrUpdate(@RequestBody User user){  
 return userService.saveOrUpdate(user)?Result.*success*():Result.*fail*();  
 }  
  
 */\*  
 \* 删除用户  
 \* @author rabbiter  
 \* @date 2023/1/2 19:15  
 \*/* @GetMapping("/del")  
 public Result delete(Integer id){  
 return userService.removeById(id)?Result.*success*():Result.*fail*();  
 }  
  
 */\*  
 \* 模糊查询  
 \* @author rabbiter  
 \* @date 2023/1/2 19:36  
 \*/* @PostMapping("/listP")  
 public Result query(@RequestBody User user){  
 LambdaQueryWrapper<User> wrapper = new LambdaQueryWrapper<>();  
 if(StringUtils.*isNotBlank*(user.getName())){  
 wrapper.like(User::getName,user.getName());  
 }  
 return Result.*success*(userService.list(wrapper));  
 }@PostMapping("/listPage")  
 public List<User> listPage(@RequestBody QueryPageParam query){  
 HashMap param = query.getParam();  
 String name = (String)param.get("name");  
 System.*out*.println("name=>"+(String)param.get("name"));  
  
 Page<User> page = new Page();  
 page.setCurrent(query.getPageNum());  
 page.setSize(query.getPageSize());  
  
 LambdaQueryWrapper<User> lambdaQueryWrapper = new LambdaQueryWrapper();  
 lambdaQueryWrapper.like(User::getName,name);  
  
  
 IPage result = userService.page(page,lambdaQueryWrapper);  
  
 System.*out*.println("total=>"+result.getTotal());  
  
 return result.getRecords();  
 }  
  
 */\*  
 \* 查询功能：根据前端表单输入的信息或者下拉框选择查询用户，并以分页的形式返回前端  
 \* @author rabbiter  
 \* @date 2023/1/4 20:28  
 \*/* @PostMapping("/listPageC1")  
 public Result listPageC1(@RequestBody QueryPageParam query){  
 HashMap param = query.getParam();  
 String name = (String)param.get("name");  
 String sex = (String)param.get("sex");  
 String roleId = (String)param.get("roleId");  
  
 Page<User> page = new Page();  
 page.setCurrent(query.getPageNum());  
 page.setSize(query.getPageSize());  
  
 LambdaQueryWrapper<User> lambdaQueryWrapper = new LambdaQueryWrapper();  
 if(StringUtils.*isNotBlank*(name) && !"null".equals(name)){  
 lambdaQueryWrapper.like(User::getName,name);  
 }  
 if(StringUtils.*isNotBlank*(sex)){  
 lambdaQueryWrapper.eq(User::getSex,sex);  
 }  
 if(StringUtils.*isNotBlank*(roleId)){  
 lambdaQueryWrapper.eq(User::getRoleId,roleId);  
 }  
  
 IPage result = userService.pageCC(page,lambdaQueryWrapper);  
  
 System.*out*.println("total=>"+result.getTotal());  
  
 return Result.*success*(result.getRecords(),result.getTotal());  
 }

# 6.2 前端页面代码实现

## 6.2.1 登录模块

<template>

    <div class="loginBody">

        <div

            style="

                width: 98%;

                text-align: center;

                color: white;

                padding: 8vh 100px 0 0;

                -webkit-text-stroke: 1px black;

                font-size: 3vh;

            "

        >

            <h1>欢迎访问仓库管理系统</h1>

        </div>

        <div class="loginDiv">

            <div class="login-content">

                <el-row>

                    <el-col style="text-align: center; padding: 25px 0 25px 0">

                        <i

                            class="iconfont icon-r-building"

                            style="font-size: 36px"

                        ></i>

                        <b style="font-size: 36px"> 仓库管理系统</b>

                    </el-col>

                </el-row>

                <el-form

                    :model="loginForm"

                    label-width="100px"

                    :rules="rules"

                    ref="loginForm"

                >

                    <el-form-item label="账号" prop="no">

                        <el-input

                            style="width: 200px"

                            type="text"

                            v-model="loginForm.no"

                            autocomplete="off"

                        ></el-input>

                    </el-form-item>

                    <el-form-item label="密码" prop="password">

                        <el-input

                            style="width: 200px"

                            type="password"

                            v-model="loginForm.password"

                            show-password

                            autocomplete="off"

                            @keyup.enter.native="confirm"

                        ></el-input>

                    </el-form-item>

                    <el-form-item style="padding-left: 45px">

                        <el-button

                            type="primary"

                            @click="confirm"

                            :disabled="confirm\_disabled"

                            style="font-size: 22px"

                        >

                            登 录</el-button

                        >

                    </el-form-item>

                </el-form>

            </div>

        </div>

    </div>

</template>



## 6.2.2 个人中心模块

<template>

    <div

        style="

            text-align: center;

            height: 100%;

            padding: 20px 0 0 0;

            margin: 0px;

        "

    >

    <div class="userInfo">

        <b> 当前登录信息</b>

    </div>

        <el-descriptions

            title=""

            :column="1"

            size="40"

            border

            style="padding: 20px 0 20px 0"

        >

            <el-descriptions-item>

                <template slot="label">

                    <i class="el-icon-s-custom"></i>

                    账号

                </template>

                <el-tag type="info">{{ user.no }}</el-tag>

            </el-descriptions-item>

            <el-descriptions-item>

                <template slot="label">

                    <i class="el-icon-user"></i>

                    姓名

                </template>

                <el-tag type="info">{{ user.name }}</el-tag>

            </el-descriptions-item>

            <el-descriptions-item>

                <template slot="label">

                    <i class="el-icon-mobile-phone"></i>

                    电话

                </template>

                <el-tag type="info">{{ user.phone }}</el-tag>

            </el-descriptions-item>

            <el-descriptions-item>

                <template slot="label">

                    <i class="el-icon-location-outline"></i>

                    性别

                </template>

                <el-tag

                    :type="user.sex == 1 ? 'primary' : 'danger'"

                    disable-transitions

                    ><i

                        :class="

                            user.sex == 1 ? 'el-icon-male' : 'el-icon-female'

                        "

                    ></i

                    >{{ user.sex == 1 ? "男" : "女" }}</el-tag

                >

            </el-descriptions-item>

            <el-descriptions-item>

                <template slot="label">

                    <i class="iconfont icon-r-mark1"></i>

                    角色

                </template>

                <el-tag type="success" disable-transitions>{{

                    user.roleId == 0

                        ? "超级管理员"

                        : user.roleId == 1

                        ? "管理员"

                        : "用户"

                }}</el-tag>

            </el-descriptions-item>

        </el-descriptions>

        <hr />

        <el-row style="text-align: left">

            <el-button

                type="danger"

                @click="logout"

                style="margin: 25px 0 0 90px; font-size: 22px;"

                >

                 退出登录</el-button

            >

        </el-row>

    </div>

</template>



## 6.2.3 管理员管理模块

<template>

    <div>

        <div style="margin-bottom: 5px">

            <el-input

                v-model="name"

                placeholder="请输入名字"

                suffix-icon="el-icon-a-042"

                style="width: 200px"

                @keyup.enter.native="loadPost"

            ></el-input>

            <el-select

                v-model="sex"

                filterable

                placeholder="请选择性别"

                style="margin-left: 5px"

            >

                <el-option

                    v-for="item in sexs"

                    :key="item.value"

                    :label="item.label"

                    :value="item.value"

                >

                </el-option>

            </el-select>

            <el-button

                type="primary"

                style="margin-left: 30px; font-size: 20px"

                @click="loadPost"

            >

                查询</el-button

            >

            <el-button

                type="warning"

                @click="resetParam"

                style="font-size: 20px"

            >

                重置</el-button

            >

            <el-button

                type="primary"

                style="margin-left: 10px; font-size: 20px"

                @click="add"

            >

                新增</el-button

            >

        </div>

        <el-table

            :data="tableData"

            :header-cell-style="{ background: '#f2f5fc', color: '#555555' }"

            border

        >

            <el-table-column prop="id" label="ID"> </el-table-column>

            <el-table-column prop="no" label="账号">

            </el-table-column>

            <el-table-column prop="name" label="姓名">

            </el-table-column>

            <el-table-column prop="age" label="年龄" width="80">

            </el-table-column>

            <el-table-column prop="sex" label="性别" width="80">

                <template slot-scope="scope">

                    <el-tag

                        :type="scope.row.sex === 1 ? 'primary' : 'success'"

                        disable-transitions

                        >{{ scope.row.sex === 1 ? "男" : "女" }}</el-tag

                    >

                </template>

            </el-table-column>

            <el-table-column prop="roleId" label="角色" width="120">

                <template slot-scope="scope">

                    <el-tag

                        :type="

                            scope.row.roleId === 0

                                ? 'danger'

                                : scope.row.roleId === 1

                                ? 'primary'

                                : 'success'

                        "

                        disable-transitions

                        >{{

                            scope.row.roleId === 0

                                ? "超级管理员"

                                : scope.row.roleId === 1

                                ? "管理员"

                                : "用户"

                        }}</el-tag

                    >

                </template>

            </el-table-column>

            <el-table-column prop="phone" label="电话">

            </el-table-column>

            <el-table-column

                prop="operate"

                label="操作"

                width="260"

                fixed="right"

            >

                <template slot-scope="scope">

                    <el-button

                        style="font-size: 20px"

                        type="success"

                        @click="mod(scope.row)"

                    >

                        编辑</el-button

                    >

                    <el-popconfirm

                        title="确定删除吗？"

                        @confirm="del(scope.row.id)"

                        style="margin-left: 5px"

                    >

                        <el-button

                            slot="reference"

                            type="danger"

                            style="font-size: 20px"

                        >

                            删除</el-button

                        >

                    </el-popconfirm>

                </template>

            </el-table-column>

        </el-table>

        <el-pagination

            @size-change="handleSizeChange"

            @current-change="handleCurrentChange"

            :current-page="pageNum"

            :page-sizes="[5, 10, 20, 30]"

            :page-size="pageSize"

            layout="total, sizes, prev, pager, next, jumper"

            :total="total"

        >

        </el-pagination>

        <el-dialog

            title="提示"

            :visible.sync="centerDialogVisible"

            width="30%"

            center

        >

            <el-form ref="form" :rules="rules" :model="form" label-width="80px">

                <el-form-item label="账号" prop="no">

                    <el-col :span="20">

                        <el-input v-model="form.no"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="名字" prop="name">

                    <el-col :span="20">

                        <el-input v-model="form.name"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="密码" prop="password">

                    <el-col :span="20">

                        <el-input v-model="form.password"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="年龄" prop="age">

                    <el-col :span="20">

                        <el-input v-model="form.age"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="性别">

                    <el-radio-group v-model="form.sex">

                        <el-radio label="1">男</el-radio>

                        <el-radio label="0">女</el-radio>

                    </el-radio-group>

                </el-form-item>

                <el-form-item label="电话" prop="phone">

                    <el-col :span="20">

                        <el-input v-model="form.phone"></el-input>

                    </el-col>

                </el-form-item>

            </el-form>

            <span slot="footer" class="dialog-footer">

                <el-button

                    @click="centerDialogVisible = false"

                    style="font-size: 18px"

                    >

                    取 消</el-button

                >

                <el-button type="primary" @click="save" style="font-size: 18px"

                    >

                    确 定</el-button

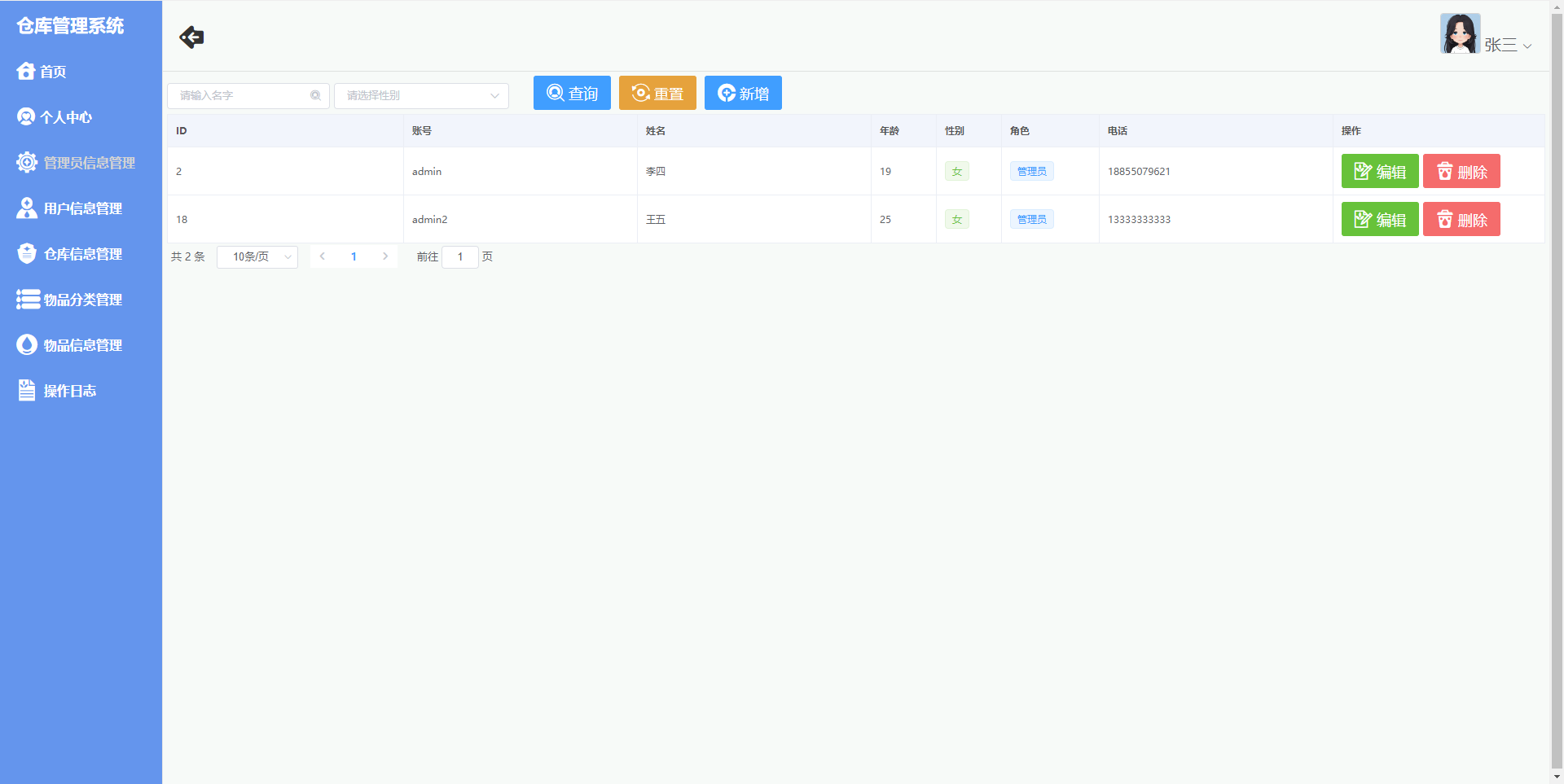
                >

            </span>

        </el-dialog>

    </div>

</template>



## 6.2.4 用户信息管理模块

<template>

    <div>

        <div style="margin-bottom: 5px">

            <el-input

                v-model="name"

                placeholder="请输入名字"

                suffix-icon="el-icon-a-042"

                style="width: 200px"

                @keyup.enter.native="loadPost"

            ></el-input>

            <el-select

                v-model="sex"

                filterable

                placeholder="请选择性别"

                style="margin-left: 5px"

            >

                <el-option

                    v-for="item in sexs"

                    :key="item.value"

                    :label="item.label"

                    :value="item.value"

                >

                </el-option>

            </el-select>

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="loadPost"

            >

                查询</el-button

            >

            <el-button

                type="warning"

                @click="resetParam"

                style="font-size: 18px"

            >

                重置</el-button

            >

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="add"

            >

                新增</el-button

            >

        </div>

        <el-table

            :data="tableData"

            :header-cell-style="{ background: '#f2f5fc', color: '#555555' }"

            border

        >

            <el-table-column prop="id" label="ID" width="60"> </el-table-column>

            <el-table-column prop="no" label="账号" width="180">

            </el-table-column>

            <el-table-column prop="name" label="姓名" width="180">

            </el-table-column>

            <el-table-column prop="age" label="年龄" width="80">

            </el-table-column>

            <el-table-column prop="sex" label="性别" width="80">

                <template slot-scope="scope">

                    <el-tag

                        :type="scope.row.sex === 1 ? 'primary' : 'success'"

                        disable-transitions

                        >{{ scope.row.sex === 1 ? "男" : "女" }}</el-tag

                    >

                </template>

            </el-table-column>

            <el-table-column prop="roleId" label="角色" width="120">

                <template slot-scope="scope">

                    <el-tag

                        :type="

                            scope.row.roleId === 0

                                ? 'danger'

                                : scope.row.roleId === 1

                                ? 'primary'

                                : 'success'

                        "

                        disable-transitions

                        >{{

                            scope.row.roleId === 0

                                ? "超级管理员"

                                : scope.row.roleId === 1

                                ? "管理员"

                                : "用户"

                        }}</el-tag

                    >

                </template>

            </el-table-column>

            <el-table-column prop="phone" label="电话">

            </el-table-column>

            <el-table-column

                prop="operate"

                label="操作"

                width="240"

                fixed="right"

            >

                <template slot-scope="scope">

                    <el-button

                        style="font-size: 18px"

                        type="success"

                        @click="mod(scope.row)"

                    >

                        编辑</el-button

                    >

                    <el-popconfirm

                        title="确定删除吗？"

                        @confirm="del(scope.row.id)"

                        style="margin-left: 5px"

                    >

                        <el-button

                            slot="reference"

                            style="font-size: 18px"

                            type="danger"

                        >

                            删除</el-button

                        >

                    </el-popconfirm>

                </template>

            </el-table-column>

        </el-table>

        <el-pagination

            @size-change="handleSizeChange"

            @current-change="handleCurrentChange"

            :current-page="pageNum"

            :page-sizes="[5, 10, 20, 30]"

            :page-size="pageSize"

            layout="total, sizes, prev, pager, next, jumper"

            :total="total"

        >

        </el-pagination>

        <el-dialog

            title="提示"

            :visible.sync="centerDialogVisible"

            width="30%"

            center

        >

            <el-form ref="form" :rules="rules" :model="form" label-width="80px">

                <el-form-item label="账号" prop="no">

                    <el-col :span="20">

                        <el-input v-model="form.no"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="名字" prop="name">

                    <el-col :span="20">

                        <el-input v-model="form.name"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="密码" prop="password">

                    <el-col :span="20">

                        <el-input v-model="form.password"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="年龄" prop="age">

                    <el-col :span="20">

                        <el-input v-model="form.age"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="性别">

                    <el-radio-group v-model="form.sex">

                        <el-radio label="1">男</el-radio>

                        <el-radio label="0">女</el-radio>

                    </el-radio-group>

                </el-form-item>

                <el-form-item label="电话" prop="phone">

                    <el-col :span="20">

                        <el-input v-model="form.phone"></el-input>

                    </el-col>

                </el-form-item>

            </el-form>

            <span slot="footer" class="dialog-footer">

                <el-button

                    @click="centerDialogVisible = false"

                    style="font-size: 18px"

                    >

                    取 消</el-button

                >

                <el-button type="primary" @click="save" style="font-size: 18px"

                    >

                    确 定</el-button

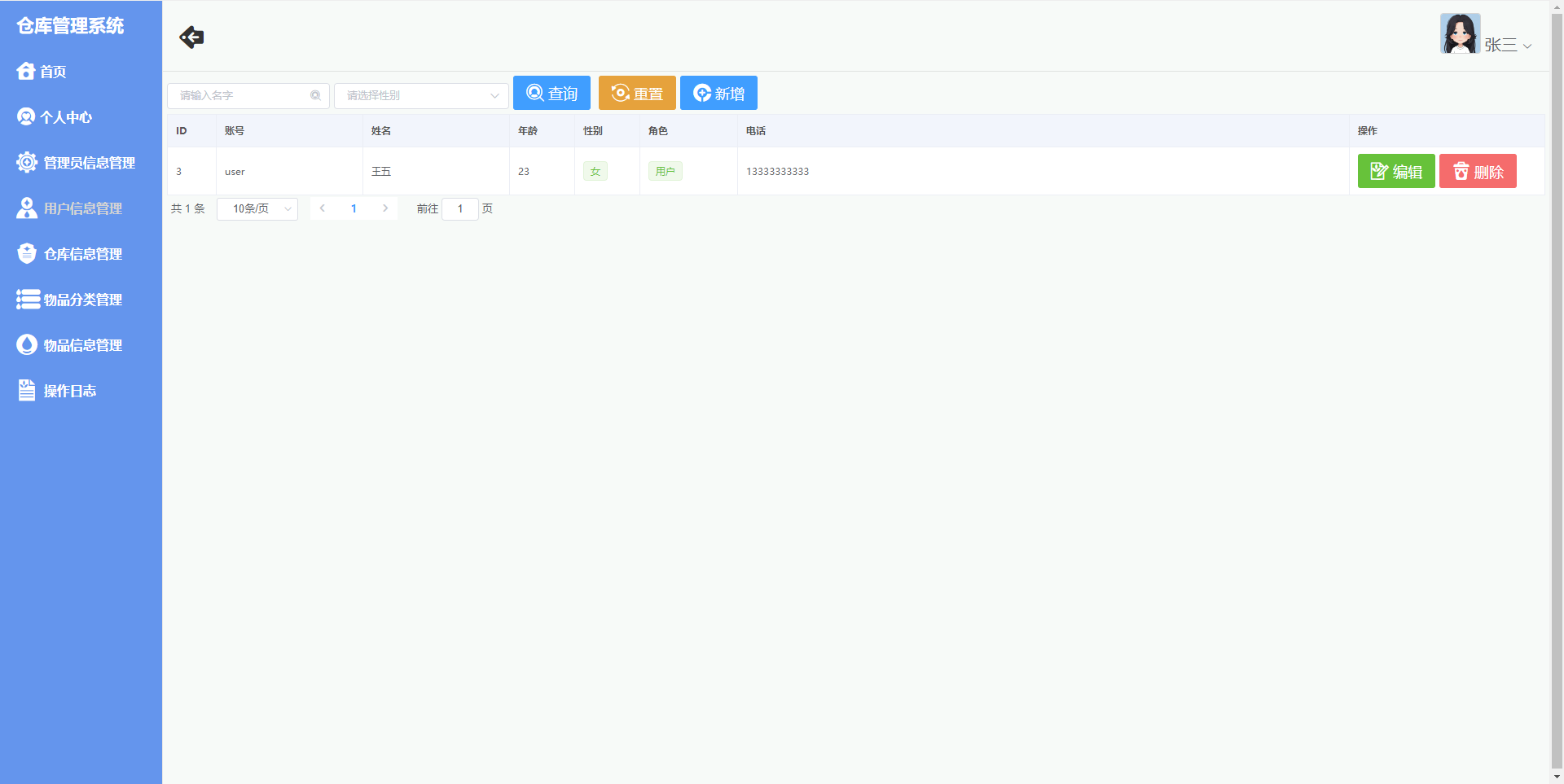
                >

            </span>

        </el-dialog>

    </div>

</template>



## 6.2.5 仓库信息管理模块

<template>

    <div>

        <div style="margin-bottom: 5px">

            <el-input

                v-model="name"

                placeholder="请输入仓库名"

                suffix-icon="el-icon-a-042"

                style="font-size: 18px; width: 200px"

                @keyup.enter.native="loadPost"

            ></el-input>

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="loadPost"

            >

                查询</el-button

            >

            <el-button

                type="warning"

                style="font-size: 18px"

                @click="resetParam"

            >

                重置</el-button

            >

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="add"

            >

                新增</el-button

            >

        </div>

        <el-table

            :data="tableData"

            :header-cell-style="{ background: '#f2f5fc', color: '#555555' }"

            border

        >

            <el-table-column prop="id" label="ID" width="60"> </el-table-column>

            <el-table-column prop="name" label="仓库名" width="180">

            </el-table-column>

            <el-table-column prop="remark" label="备注"> </el-table-column>

            <el-table-column

                prop="operate"

                label="操作"

                width="240"

                fixed="right"

            >

                <template slot-scope="scope">

                    <el-button

                        style="font-size: 18px"

                        type="success"

                        @click="mod(scope.row)"

                    >

                        编辑</el-button

                    >

                    <el-popconfirm

                        title="确定删除吗？"

                        @confirm="del(scope.row.id)"

                        style="margin-left: 5px"

                    >

                        <el-button

                            slot="reference"

                            type="danger"

                            style="font-size: 18px"

                        >

                            删除</el-button

                        >

                    </el-popconfirm>

                </template>

            </el-table-column>

        </el-table>

        <el-pagination

            @size-change="handleSizeChange"

            @current-change="handleCurrentChange"

            :current-page="pageNum"

            :page-sizes="[5, 10, 20, 30]"

            :page-size="pageSize"

            layout="total, sizes, prev, pager, next, jumper"

            :total="total"

        >

        </el-pagination>

        <el-dialog

            title="提示"

            :visible.sync="centerDialogVisible"

            width="30%"

            center

        >

            <el-form ref="form" :rules="rules" :model="form" label-width="80px">

                <el-form-item label="仓库名" prop="name">

                    <el-col :span="20">

                        <el-input v-model="form.name"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="备注" prop="remark">

                    <el-col :span="20">

                        <el-input

                            type="textarea"

                            v-model="form.remark"

                        ></el-input>

                    </el-col>

                </el-form-item>

            </el-form>

            <span slot="footer" class="dialog-footer">

                <el-button

                    @click="centerDialogVisible = false"

                    style="font-size: 18px"

                    >

                    取 消</el-button

                >

                <el-button type="primary" @click="save" style="font-size: 18px"

                    >

                    确 定</el-button

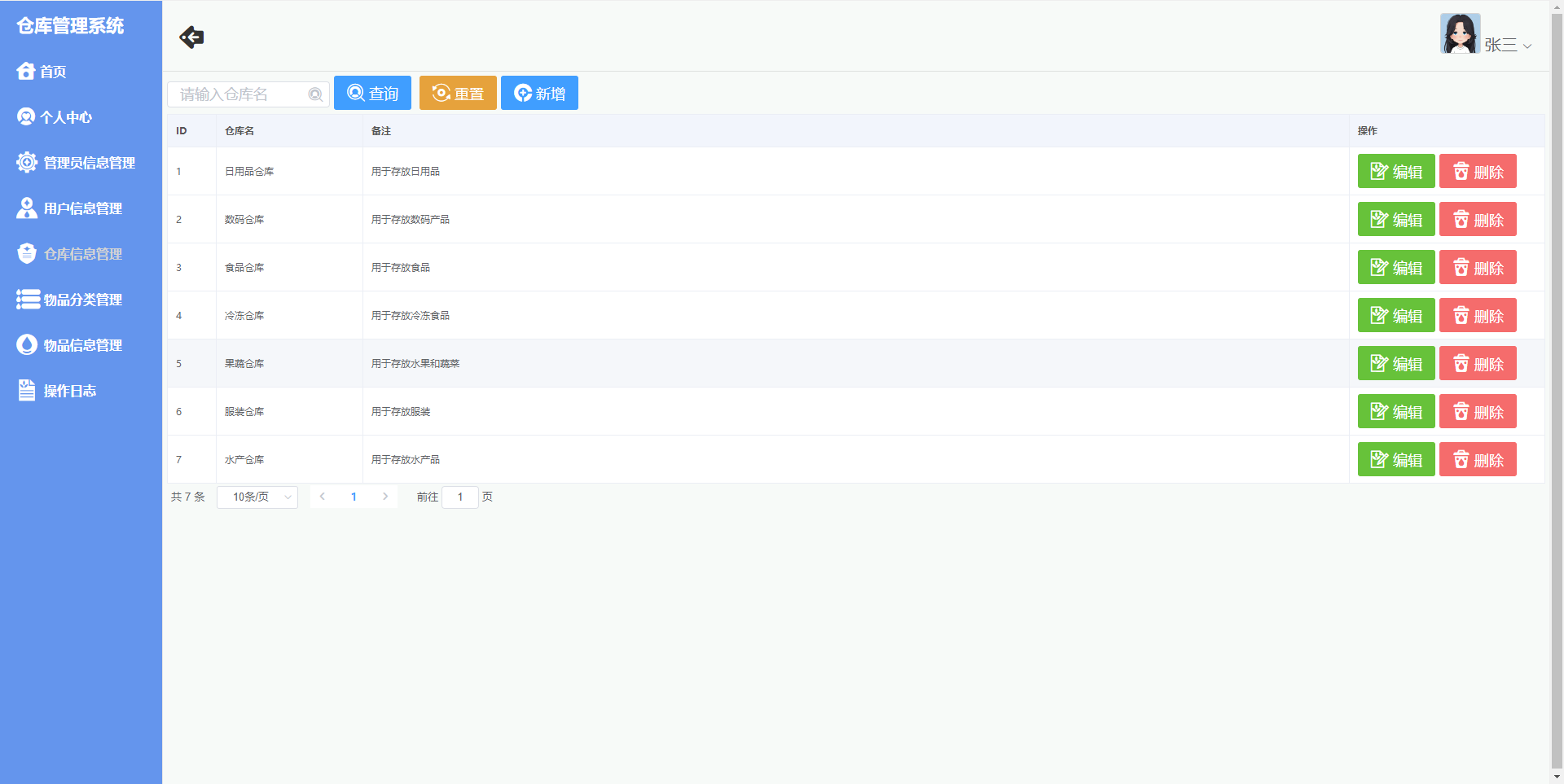
                >

            </span>

        </el-dialog>

    </div>

</template>



## 6.2.6 物品分类管理模块

<template>

    <div>

        <div style="margin-bottom: 5px">

            <el-input

                v-model="name"

                placeholder="请输入分类名"

                suffix-icon="el-icon-a-042"

                style="width: 200px"

                @keyup.enter.native="loadPost"

            ></el-input>

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="loadPost"

            >

                查询</el-button

            >

            <el-button

                type="warning"

                @click="resetParam"

                style="font-size: 18px"

            >

                重置</el-button

            >

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="add"

            >

                新增</el-button

            >

        </div>

        <el-table

            :data="tableData"

            :header-cell-style="{ background: '#f2f5fc', color: '#555555' }"

            border

        >

            <el-table-column prop="id" label="ID" width="60"> </el-table-column>

            <el-table-column prop="name" label="分类名" width="180">

            </el-table-column>

            <el-table-column prop="remark" label="备注"> </el-table-column>

            <el-table-column

                prop="operate"

                label="操作"

                width="240"

                fixed="right"

            >

                <template slot-scope="scope">

                    <el-button

                        style="font-size: 18px"

                        type="success"

                        @click="mod(scope.row)"

                    >

                        编辑</el-button

                    >

                    <el-popconfirm

                        title="确定删除吗？"

                        @confirm="del(scope.row.id)"

                        style="margin-left: 5px"

                    >

                        <el-button

                            slot="reference"

                            style="font-size: 18px"

                            type="danger"

                        >

                            删除</el-button

                        >

                    </el-popconfirm>

                </template>

            </el-table-column>

        </el-table>

        <el-pagination

            @size-change="handleSizeChange"

            @current-change="handleCurrentChange"

            :current-page="pageNum"

            :page-sizes="[5, 10, 20, 30]"

            :page-size="pageSize"

            layout="total, sizes, prev, pager, next, jumper"

            :total="total"

        >

        </el-pagination>

        <el-dialog

            title="提示"

            :visible.sync="centerDialogVisible"

            width="30%"

            center

        >

            <el-form ref="form" :rules="rules" :model="form" label-width="80px">

                <el-form-item label="分类名" prop="name">

                    <el-col :span="20">

                        <el-input v-model="form.name"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="备注" prop="remark">

                    <el-col :span="20">

                        <el-input

                            type="textarea"

                            v-model="form.remark"

                        ></el-input>

                    </el-col>

                </el-form-item>

            </el-form>

            <span slot="footer" class="dialog-footer">

                <el-button

                    @click="centerDialogVisible = false"

                    style="font-size: 18px"

                    >

                    取 消</el-button

                >

                <el-button type="primary" @click="save" style="font-size: 18px"

                    >

                    确 定</el-button

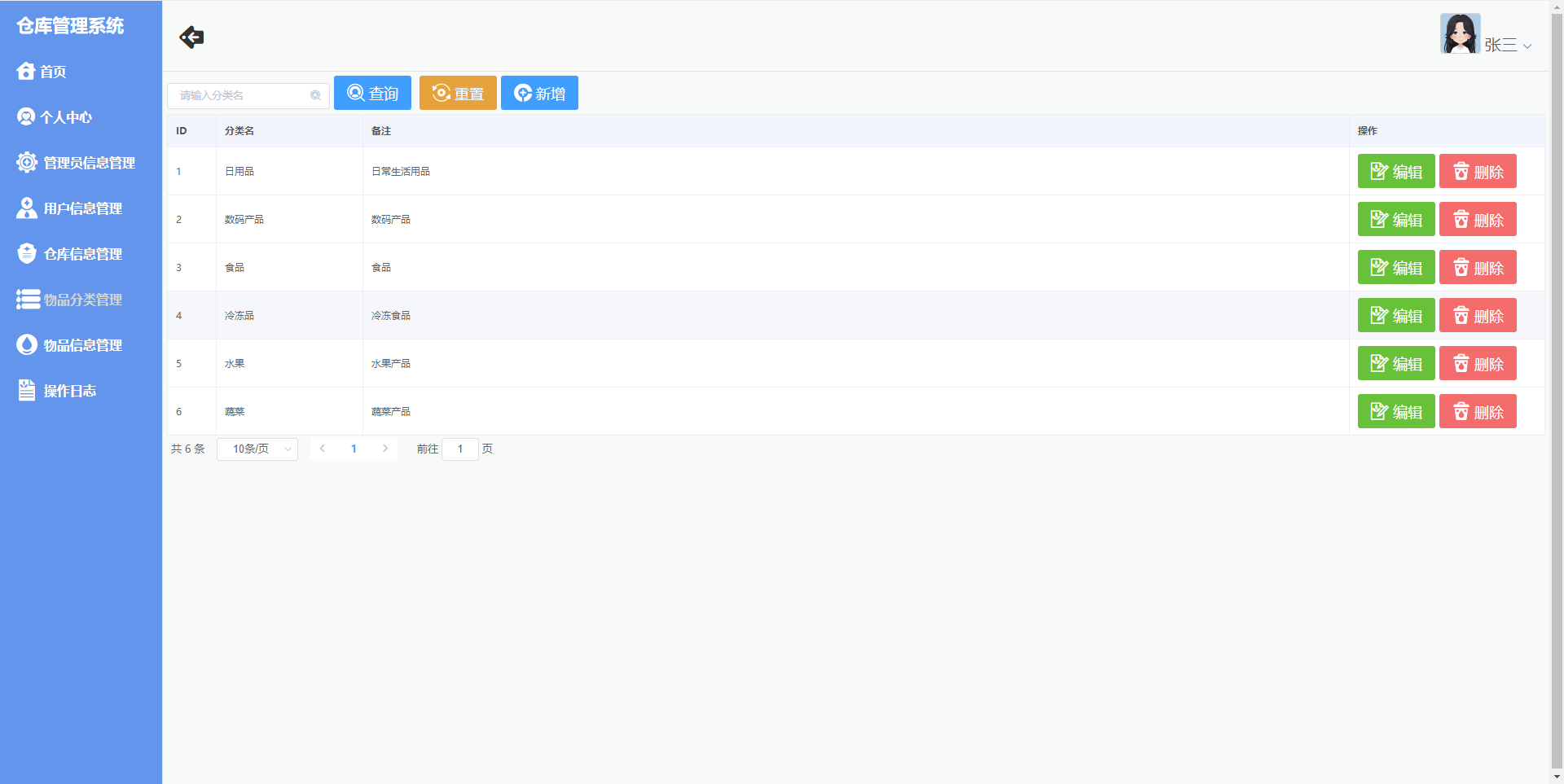
                >

            </span>

        </el-dialog>

    </div>

</template>



## 6.2.7 物品信息管理模块

<template>

    <div>

        <div style="margin-bottom: 5px">

            <el-input

                v-model="name"

                placeholder="请输入物品名"

                suffix-icon="el-icon-a-042"

                style="width: 200px"

                @keyup.enter.native="loadPost"

            ></el-input>

            <el-select

                v-model="storage"

                placeholder="请选择仓库"

                style="margin-left: 5px"

            >

                <el-option

                    v-for="item in storageData"

                    :key="item.id"

                    :label="item.name"

                    :value="item.id"

                >

                </el-option>

            </el-select>

            <el-select

                v-model="goodstype"

                placeholder="请选择分类"

                style="margin-left: 5px"

            >

                <el-option

                    v-for="item in goodstypeData"

                    :key="item.id"

                    :label="item.name"

                    :value="item.id"

                >

                </el-option>

            </el-select>

            <br /><br />

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="loadPost"

            >

                查询</el-button

            >

            <el-button

                type="warning"

                @click="resetParam"

                style="font-size: 18px"

            >

                重置</el-button

            >

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="add"

                v-if="user.roleId != 2"

            >

                新增</el-button

            >

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="inGoods"

                v-if="user.roleId != 2"

            >

                入库</el-button

            >

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="outGoods"

                v-if="user.roleId != 2"

            >

                出库</el-button

            >

        </div>

        <el-table

            :data="tableData"

            :header-cell-style="{ background: '#f2f5fc', color: '#555555' }"

            border

            highlight-current-row

            @current-change="selectCurrentChange"

        >

            <el-table-column prop="id" label="ID" width="60"> </el-table-column>

            <el-table-column prop="name" label="物品名" width="160">

            </el-table-column>

            <el-table-column

                prop="storage"

                label="仓库"

                width="160"

                :formatter="formatStorage"

            >

            </el-table-column>

            <el-table-column

                prop="goodstype"

                label="分类"

                width="140"

                :formatter="formatGoodstype"

            >

            </el-table-column>

            <el-table-column prop="count" label="数量" width="150">

            </el-table-column>

            <el-table-column prop="remark" label="备注"> </el-table-column>

            <el-table-column

                prop="operate"

                label="操作"

                v-if="user.roleId != 2"

                fixed="right"

                width="240"

            >

                <template slot-scope="scope">

                    <el-button

                        type="success"

                        @click="mod(scope.row)"

                        style="font-size: 18px"

                    >

                        编辑</el-button

                    >

                    <el-popconfirm

                        title="确定删除吗？"

                        @confirm="del(scope.row.id)"

                        style="margin-left: 5px"

                    >

                        <el-button

                            slot="reference"

                            type="danger"

                            style="font-size: 18px"

                        >

                            删除</el-button

                        >

                    </el-popconfirm>

                </template>

            </el-table-column>

        </el-table>

        <el-pagination

            @size-change="handleSizeChange"

            @current-change="handleCurrentChange"

            :current-page="pageNum"

            :page-sizes="[5, 10, 20, 30]"

            :page-size="pageSize"

            layout="total, sizes, prev, pager, next, jumper"

            :total="total"

        >

        </el-pagination>

        <el-dialog

            title="物品维护"

            :visible.sync="centerDialogVisible"

            width="30%"

            center

        >

            <el-form ref="form" :rules="rules" :model="form" label-width="80px">

                <el-form-item label="物品名" prop="name">

                    <el-col :span="20">

                        <el-input v-model="form.name"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="仓库" prop="storage">

                    <el-col :span="20">

                        <el-select

                            v-model="form.storage"

                            placeholder="请选择仓库"

                            style="margin-left: 5px"

                        >

                            <el-option

                                v-for="item in storageData"

                                :key="item.id"

                                :label="item.name"

                                :value="item.id"

                            >

                            </el-option>

                        </el-select>

                    </el-col>

                </el-form-item>

                <el-form-item label="分类" prop="goodstype">

                    <el-col :span="20">

                        <el-select

                            v-model="form.goodstype"

                            placeholder="请选择分类"

                            style="margin-left: 5px"

                        >

                            <el-option

                                v-for="item in goodstypeData"

                                :key="item.id"

                                :label="item.name"

                                :value="item.id"

                            >

                            </el-option>

                        </el-select>

                    </el-col>

                </el-form-item>

                <el-form-item label="数量" prop="count">

                    <el-col :span="20">

                        <el-input v-model="form.count"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="备注" prop="remark">

                    <el-col :span="20">

                        <el-input

                            type="textarea"

                            v-model="form.remark"

                        ></el-input>

                    </el-col>

                </el-form-item>

            </el-form>

            <span slot="footer" class="dialog-footer">

                <el-button

                    @click="centerDialogVisible = false"

                    style="font-size: 18px"

                    >

                    取 消</el-button

                >

                <el-button type="primary" @click="save" style="font-size: 18px"

                    >

                    确 定</el-button

                >

            </span>

        </el-dialog>

        <el-dialog

            title="出入库"

            :visible.sync="inDialogVisible"

            width="30%"

            center

        >

            <el-dialog

                width="75%"

                title="用户选择"

                :visible.sync="innerVisible"

                append-to-body

            >

                <SelectUser @doSelectUser="doSelectUser"></SelectUser>

                <span slot="footer" class="dialog-footer">

                    <el-button

                        @click="innerVisible = false"

                        style="font-size: 18px"

                        >

                         取 消</el-button

                    >

                    <el-button

                        type="primary"

                        @click="confirmUser"

                        style="font-size: 18px"

                        >

                         确 定</el-button

                    >

                </span>

            </el-dialog>

            <el-form

                ref="form1"

                :rules="rules1"

                :model="form1"

                label-width="80px"

            >

                <el-form-item label="物品名">

                    <el-col :span="20">

                        <el-input v-model="form1.goodsname" readonly></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="申请人">

                    <el-col :span="20">

                        <el-input

                            v-model="form1.username"

                            readonly

                            @click.native="selectUser"

                        ></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="数量" prop="count">

                    <el-col :span="20">

                        <el-input v-model="form1.count"></el-input>

                    </el-col>

                </el-form-item>

                <el-form-item label="备注" prop="remark">

                    <el-col :span="20">

                        <el-input

                            type="textarea"

                            v-model="form1.remark"

                        ></el-input>

                    </el-col>

                </el-form-item>

            </el-form>

            <span slot="footer" class="dialog-footer">

                <el-button

                    @click="inDialogVisible = false"

                    style="font-size: 18px"

                    >

                    取 消</el-button

                >

                <el-button

                    type="primary"

                    @click="doInGoods"

                    style="font-size: 18px"

                    >

                    确 定</el-button

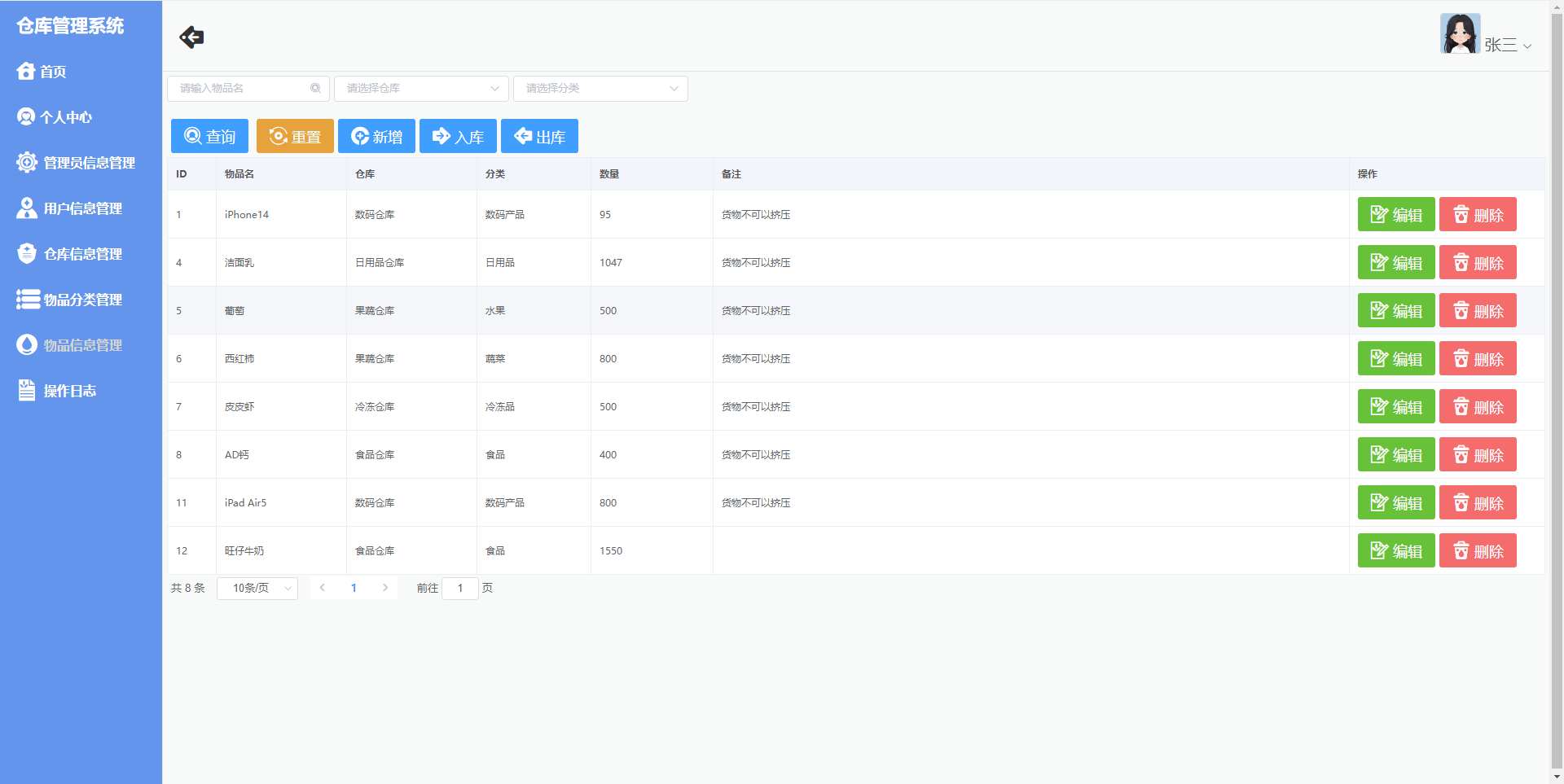
                >

            </span>

        </el-dialog>

    </div>

</template>



## 6.2.8 操作日志模块

<template>

    <div>

        <div style="margin-bottom: 5px">

            <el-input

                v-model="name"

                placeholder="请输入物品名"

                suffix-icon="el-icon-a-042"

                style="width: 200px"

                @keyup.enter.native="loadPost"

            ></el-input>

            <el-select

                v-model="storage"

                placeholder="请选择仓库"

                style="margin-left: 5px"

            >

                <el-option

                    v-for="item in storageData"

                    :key="item.id"

                    :label="item.name"

                    :value="item.id"

                >

                </el-option>

            </el-select>

            <el-select

                v-model="goodstype"

                placeholder="请选择分类"

                style="margin-left: 5px"

            >

                <el-option

                    v-for="item in goodstypeData"

                    :key="item.id"

                    :label="item.name"

                    :value="item.id"

                >

                </el-option>

            </el-select>

            <el-button

                type="primary"

                style="margin-left: 5px; font-size: 18px"

                @click="loadPost"

            >

                查询</el-button

            >

            <el-button

                type="warning"

                style="font-size: 18px"

                @click="resetParam"

            >

                重置</el-button

            >

        </div>

        <el-table

            :data="tableData"

            :header-cell-style="{ background: '#f2f5fc', color: '#555555' }"

            border

        >

            <el-table-column prop="id" label="ID" width="60"> </el-table-column>

            <el-table-column prop="goodsname" label="物品名" width="150">

            </el-table-column>

            <el-table-column prop="storagename" label="仓库" width="150">

            </el-table-column>

            <el-table-column prop="goodstypename" label="分类" width="150">

            </el-table-column>

            <el-table-column prop="adminname" label="操作人" width="150">

            </el-table-column>

            <el-table-column prop="username" label="申请人" width="150">

            </el-table-column>

            <el-table-column prop="count" label="数量" width="150">

            </el-table-column>

            <el-table-column prop="createtime" label="操作时间" width="180">

            </el-table-column>

            <el-table-column prop="remark" label="备注">

            </el-table-column>

        </el-table>

        <el-pagination

            @size-change="handleSizeChange"

            @current-change="handleCurrentChange"

            :current-page="pageNum"

            :page-sizes="[5, 10, 20, 30]"

            :page-size="pageSize"

            layout="total, sizes, prev, pager, next, jumper"

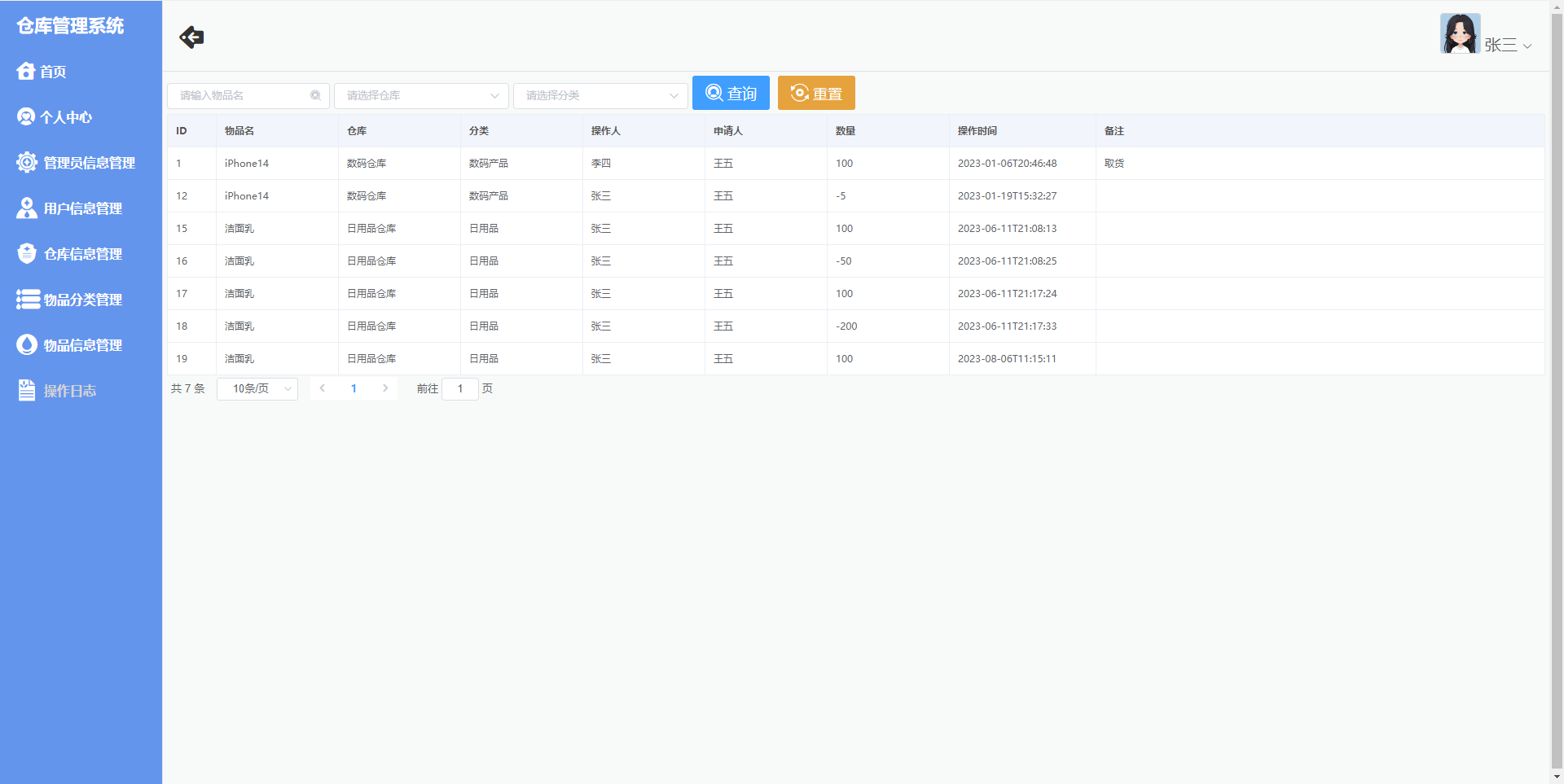
            :total="total"

        >

        </el-pagination>

    </div>

</template>



# 第7章 系统测试

# 7.1 软硬件环境

操作系统：Windows 10

数据库：MySQL 8.0.19

应用软件：Idea 2021.03

浏览器：Chrome 90.0.4430.93

软硬件环境：10台电脑

# 7.2 测试方法

系统测试是指在完成项目编码后，系统正式上线前，根据系统前期的需求分析以及规格说明进行最终审查。系统测试可以验证系统是否达到需求，每个功能模块是否存在着一些缺陷或错误引发系统不正常的运行[[3]](#footnote-3)[8]，从而保证了系统的准确性与可靠性。

本系统在完成每一个小功能之后，都会进行单元测试，以此来找到存在的BUG并解决，同时可以检验功能实现是否达到了预期的效果。在系统完成后使用黑盒方法对本系统进行测试。

# 7.3 测试用例

| **编号** | **用例名** | **预期结果** | **实际结果** |
| --- | --- | --- | --- |
| cs0101 | 系统管理员登录成功 | 显示系统管理员首页 | 登录成功，显示系统管理员首页 |
| cs0102 | 系统管理员登录失败 | 显示登录失败提示消息 | 登录失败，提示用户名或密码错误 |
| cs0103 | 系统管理员个人中心查询 | 显示系统管理员的个人信息 | 显示系统管理员的个人信息 |
| cs0104 | 添加管理员信息 | 成功添加管理员信息，保存到数据库 | 添加管理员信息成功，数据保存到数据库中 |
| cs0105 | 管理员信息管理-删除管理员 | 成功删除指定管理员信息，数据库中相应记录被删除 | 删除管理员信息成功，数据库中相应记录已被删除 |
| cs0106 | 仓库信息管理-添加仓库 | 成功添加仓库信息，保存到数据库 | 添加仓库信息成功，数据保存到了数据库中 |
| cs0107 | 仓库信息管理-修改仓库信息 | 成功更新仓库信息，数据库中相应记录被更新 | 修改仓库信息成功，数据库中相应记录已被更新 |
| cs0201 | 仓库管理员登录成功 | 显示仓库管理员首页 | 登录成功，显示仓库管理员首页 |
| cs0202 | 仓库管理员登录失败 | 显示登录失败提示消息 | 登录失败，提示用户名或密码错误 |
| cs0203 | 仓库管理员个人中心查询 | 显示仓库管理员的个人信息 | 显示仓库管理员的个人信息 |
| cs0204 | 物品信息管理-查询物品信息 | 显示物品信息列表 | 显示物品信息列表 |
| cs0205 | 物品信息管理-修改物品信息 | 成功更新物品信息，数据库中相应记录被更新 | 修改物品信息成功，数据库中相应记录已被更新 |
| cs0301 | 普通用户登录成功 | 显示普通用户首页 | 登录成功，显示普通用户首页 |
| cs0302 | 普通用户登录失败 | 显示登录失败提示消息 | 登录失败，提示用户名或密码错误 |
| cs0303 | 普通用户个人中心查询 | 显示普通用户的个人信息 | 显示普通用户的个人信息 |
| cs0304 | 操作日志查询 | 显示相应操作日志列表 | 显示相应操作日志列表 |

# 7.4 结果及分析

在进行系统测试时，我们按照测试用例的要求，逐个执行了各个功能的测试，并记录了实际测试结果。

通过对测试结果的分析，我们可以认为系统在这些功能上表现良好，与预期结果一致。这些测试涵盖了系统管理员登录、个人中心查询、管理员信息管理等关键功能，通过了测试，符合预期结果。系统具有稳定性和准确性，在提供用户界面和功能方面表现良好。

然而，为了保证系统的质量和稳定性，我们需要进行更全面的测试和评估。这些包括对其他角色功能的测试、性能和压力测试、安全性和可靠性的评估等。通过全面测试和评估，我们可以更好地了解系统的性能、稳定性和用户体验，确保系统对于用户而言是可靠且高效的。

在测试中，我们也发现了一些问题和改进的空间。例如，在管理员信息管理功能中，我们可以进一步测试修改管理员信息和删除管理员信息的功能，以确保这些功能也能正常工作并与预期结果一致。此外，我们还可以加强输入边界测试，确保系统能够处理各种不同的输入情况，并提供相应的错误处理和提示信息。

总之，通过以上的测试结果及分析，我们对系统的功能进行了验证和评估，系统在关键功能上表现良好，具有稳定性和准确性。然而，为了确保系统的质量，我们还需要进一步完善和加强更全面的测试和评估。

# 第8章 总结与展望

# 8.1 总结

经过几个月的艰苦奋斗，终于完成了本系统。因为之前对医院的业务需求不太熟悉，所以在开始编写本系统之前，我做了问卷调查和去了实地进行需求调研，最后确定了本系统的基本技术框架和基本业务功能。使用MySQL数据库建立完善的关系型数据库，用了SpringBoot作为后端基本框架，前端选用了Vue.js加上ElementUI进行搭建。因为对相关技术不够熟悉，在完成本系统的过程中，遇到了各种大大小小的问题。但是通过查询资料和请教我的导师，都能一一解决。

本系统的主要技术难点如下：

前后端分离架构的实现： 该系统采用了前后端分离的架构，前端使用Vue框架实现用户界面，后端使用Spring Boot框架提供数据处理和业务逻辑。在前后端分离架构中，前端和后端需要通过API进行通信。实现前后端分离涉及到跨域问题、API设计、数据传输格式等方面的技术难点。需要确保前端和后端的协作顺畅，保证数据的准确传递和处理。

以上技术难点是在该仓库管理系统中可能会遇到的挑战。前后端分离架构的实现需要关注前后端的协作和数据传输，而角色权限管理需要设计灵活、可扩展的权限控制机制。解决这些技术难点将确保系统的运行和安全性，并为用户提供良好的使用体验。

# 8.2 展望

随着科技的不断进步和仓库管理需求的不断增长，仓库管理系统将继续发展和改进，以满足日益复杂和多样化的业务需求。展望未来，仓库管理系统有以下几个方面的发展趋势和展望：

智能化与自动化：随着物联网（IoT）、人工智能（AI）等技术的迅速发展，仓库管理系统将越来越注重智能化和自动化。智能仓库系统可以通过传感器和智能设备来实时监测和管理库存情况，自动化货物的装卸和移动，提供智能化的调度和优化方案。这将提高仓库管理的效率和准确性，降低人力成本和人为错误的风险。

数据分析和预测：数据分析和预测将成为仓库管理系统的重要组成部分。通过对大量的仓库数据进行分析和挖掘，系统可以帮助管理人员更好地了解库存趋势、需求预测和供应链优化。利用机器学习和数据模型，仓库管理系统可以提供更准确的库存管理和物流调度预测，以及合理的物品配置和补货策略。

移动化：随着移动设备的普及和移动互联网的发展，仓库管理系统将更加注重移动化。通过手机、平板电脑等移动设备，仓库管理人员可以随时随地访问系统，进行库存查询、订单处理、出入库操作等。移动化的仓库管理系统将提供更高的灵活性和响应速度，使仓库管理更加便捷高效。

云计算与共享经济：云计算技术的应用将为仓库管理系统带来更大的灵活性和可扩展性。通过将系统部署在云平台上，可以实现资源的共享和按需分配，降低系统的硬件和维护成本。同时，云计算还可以实现多地点、多用户的协同工作，提高仓库管理的全局效率。

综上所述，未来仓库管理系统将朝着智能化、自动化、数据分析与预测、移动化以及云计算与共享经济的方向发展。这些趋势将进一步优化仓库管理流程，提高效率和准确性，为企业创造更大的价值。

**参考文献**

[1]王越. JAVA编程语言在计算机软件开发中的应用[J]. 电子技术与软件工程, 2019.

[2]程亚维, 王东霞. 基于Java的农机作业调度管理Web平台架构技术研究[J]. 农机化研究, 2019.

[3]葛萌, 黄素萍, 欧阳宏基. 基于Spring MVC框架的Java Web应用[J]. 计算机与现代化, 2018.

[4]窦丹宁。仓储管理和现代物流发展的分析[J].科技创新与应用，2017,02:274.  
[5]王锡琳。如何通过优化仓储管理降低企业成本[J].知识经济，2017,02:91-92.

[6]刘潇潇。基于品类管理的电商仓储企业数据化运营管理模式探讨[J].商业经济研究，2017,01:95-96.  
[7]赵增辉，冯旭。基于NHibernate的仓储管理系统设计与实现[J].物流工程与管理，2017,02:61-62.

[8]刘忠印。仓储管理中“流”元素探析及管理选择[J].物流工程与管理，2017,01:27-28.  
[9]曹孟可，杨伶俐。RFID技术在跨境电商企业仓储中的应用[J].物流工程与管理，2017,02:59-60.

[10]任天舒。基于物流技能大赛的高职类现代仓储管理课程构建[J].物流工程与管理，2017,03:179-180.  
[11]项道东。射频技术下数字化仓储物流管理系统的设计研究[J].中国商论，2017,05:46-47.

**致谢**

四年的大学生活就快走入尾声，我的校园生活就要划上句号，心中是无尽的难舍与眷恋。从这里走出，对我的人生来说，将是踏上一个新的征程，要把所学的知识应用到实际工作中去。回首四年，取得了些许成绩，生活中有快乐也有艰辛。感谢老师们四年来对我孜孜不倦的教诲，对我成长的关心和爱护。

1. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)