



电子元器件管理系统设计

Design of Electronic Component Management System

学 院： 信息科学与技术学院
专 业： 自动化
姓 名： 潘韵学
学 号： 200910301
指 导 教 师： 邓开连
提 交 日 期： 2024年6月

东华大学

毕业论文（设计）学术诚信声明

本人郑重声明：所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。本论文（设计）为本人亲自撰写，对所写的内容负责。本人完全意识到本声明的法律结果由本人承担。

作者签名：潘韵学

日期：2024 年 6 月 17 日

东华大学

毕业论文（设计）版权使用授权书

本毕业论文（设计）作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权东华大学可以将本毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业论文（设计）。

保密☐，在____年解密后适用本授权书。

本论文属于

不保密☒。

（请在以上方框内打“√”）

指导教师签名：邓开连

作者签名：潘韵宇

日期：2024 年 6 月 17 日

日期：2024 年 6 月 17 日

电子元器件管理系统设计

摘 要

电子元器件是高校实验室必备物品，由于其种类多、信息量庞大、体积小等特性，管理电子元器件难度很大。国内很多高校仍在采用较为原始的人工记录方式管理电子元器件库，缺乏系统科学的管理方式，给管理员带来了巨大的工作量，也使学生取用不便，易造成浪费。

随着信息技术的发展，数据库管理系统的应用逐渐广泛，市场上陆续出现大型企业级 ERP 管理系统，但此类系统一般操作流程复杂，学习成本较高，对于校级实验室的体量来说并不合适。本次电子元器件管理系统的设计，针对目标用户群体展开需求分析，旨在提高校内电子实验室管理元器件的效率，分析业务流程后，系统功能包括元器件的分类管理，元器件采购和申领，及权限管理、用户管理等信息管理。

本次系统开发的工作包括需求分析、数据表设计、功能的设计与实现。其中功能的设计与实现为重点工作。系统开发采用前后端分离技术，前端采用 Vue.js 技术完成页面设计，后端采用 Springboot 集成 Maven 实现前后端数据接口，集成 MybatisPlus 实现对 MySQL 数据库的读写。能实现包括采购、入库、申领等基本功能的元器件管理，应用场景广泛。

关键词： 元器件，Vue.js 技术，Springboot 框架，MySQL，Java

DESIGN OF ELECTRONIC COMPONENT MANAGEMENT SYSTEM

ABSTRACT

Electronic components are essential items in college laboratories, and it is very difficult to manage electronic components due to their characteristics such as multiple types, huge amount of information, and small size. Many domestic colleges and universities are still using the more primitive manual records to manage the electronic components library, the lack of systematic and scientific management, to the administrator has brought a huge workload, but also to make the students inconvenient to access, easy to waste.

With the development of information technology, the application of database management system is gradually widespread, the market has appeared one after another large-scale enterprise-level erp management system, but such systems are generally complex operation process, high learning costs, for the volume of school-level laboratories is not suitable. The design of the electronic components management system, for the target user groups to carry out demand analysis, aims to improve the efficiency of the school electronic laboratory management components, analysis of the business process, the system functions include classification management of components, components procurement and requisition, and rights management, user management and other information management.

The work of this system development includes requirement analysis, data table design, design and implementation of functions. Among them, the design and implementation of the function is the key work. The system development adopts front-end and back-end separation technology, the front-end uses Vue.js technology to complete the page design, the back-end uses Springboot to integrate Maven to achieve

the front and back-end data interfaces, and integrates MybatisPlus to achieve the reading and writing of the MySQL database. Can achieve the basic functions of component management including procurement, warehousing, requisitioning, etc., with a wide range of application scenarios.

Key words: Components, Vue.js Technology, Springboot Framework, MySQL, Java

目 录

1 绪论.....	7
1.1 研究背景.....	7
1.2 研究现状.....	8
1.3 研究内容.....	9
1.4 论文主要内容和组织架构.....	9
2 开发环境和关键技术.....	11
2.1 开发环境.....	11
2.2 开发工具 IDEA.....	11
2.3 MySQL 数据库.....	11
2.4 Vue.js 技术.....	12
2.5 Springboot 框架.....	12
2.6 MyBatis-Plus.....	13
2.7 Redis.....	13
3 需求分析.....	14
3.1 业务流程分析.....	14
3.2 功能需求分析.....	14
3.2.1 登录与注册.....	15
3.2.2 主页功能.....	15
3.2.3 权限管理.....	15
3.2.4 部门管理.....	15
3.2.5 用户管理.....	15
3.2.6 元器件分类管理.....	16
3.2.7 元器件采购管理.....	16
3.2.8 元器件领用管理.....	16
3.2.9 数据统计.....	16

3.2.10 日志管理	16
3.3 其他需求分析	17
3.3.1 系统安全性	17
3.3.2 系统可维护性	17
3.3.3 系统易用性	17
4 系统详细设计	18
4.1 软件设计模式	18
4.2 系统体系结构	18
4.3 前端界面设计	20
4.4 数据表设计	21
4.4.1 用户表	22
4.4.2 部门表	23
4.4.3 角色与权限表	24
4.4.4 数据字典表	25
4.4.5 元器件档案表	27
4.4.6 元器件领用表	28
4.4.7 元器件采购表	30
4.4.8 供应商表	31
4.4.9 仓库表	32
4.4.10 日志管理表	33
4.5 功能模块的接口设计	34
4.5.1 登录与注册功能的接口设计	34
4.5.2 权限管理功能的接口设计	35
4.5.3 部门管理功能的接口设计	36
4.5.4 用户管理功能的接口设计	37
4.5.5 元器件分类管理功能的接口设计	37
4.5.6 元器件采购管理功能的接口设计	38
4.5.7 元器件领用管理功能的接口设计	39
4.5.8 日志管理功能的接口设计	40

4.5.9 个人门户管理功能的接口设计	41
4.6 其他功能设计	41
5 电子元器件管理系统实现与测试	43
5.1 平台部署	43
5.2 系统页面展示	46
5.3 系统测试	61
6 结论	66
6.1 总结	66
6.2 展望	66
参考文献	67
致谢	69

1 绪论

1.1 研究背景

为了深化高校学生的实践与创新能力培养，教育部积极推动实验教学改革和实验室建设，特别重视国家级实验教学示范中心的建设，其中理工科领域的示范中心占比高达 69.72%^[1]。随着实验室的全面开放，越来越多的学生表现出对自主设计实验和竞赛准备的浓厚兴趣。在这个过程中，学生们需要频繁地申领和使用电子元器件，以满足他们的实验需求。

电子元器件，作为电子元件与小型仪器的核心组成部分^[2]，涵盖了广泛的种类，并且随着技术的不断进步，新产品层出不穷。按照常见的分类，它们可以被归为电阻类、电容类、磁性元件类、电位器类、继电器类、开关类、接插件、滤波元件、PCB 板等多个类别。然而，这些类别下又细分出众多型号与规格，导致学生在申领使用时耗费大量时间，同时也给管理员带来了繁重的统计和采购工作。

电子元器件的管理挑战不仅源于其种类繁多、数量庞大的特性，更因其每个元件都包含型号、封装、引脚定义、工作电压范围、最大功率、供应商等详尽信息，数据量庞大。当前，许多高校实验室依旧依赖传统的管理方式，即手工记录学生取用情况，通过人工手段追踪元件使用，并依赖人工判断库存情况进行补充^[3]。这种方法既耗时又易出错，且存在非实名制取用导致的过量浪费和未使用元件无法及时回收等问题，从而阻碍了电子元器件的高效管理^[4]。因此，如何设计一套有效的电子元器件管理系统，以提升高校学生和管理人员的使用和管理效率，成为了一个亟待解决的课题。

随着我国数据库技术的持续发展和普及，其在互联网领域的应用愈发广泛，已广泛渗透到政府及企业的管理之中。数据库，作为一种经过精心设计的数据结构集合，旨在有序地存储、管理和维护数据，这些信息长期保留在计算机系统中，具有显著的组织性特征，便于数据间的共享和一致性的维护。在现有的众多数据库技术中，基于 Web 的数据库系统因其广泛的适用性和重要性而备受青睐^[5]。因

此，开发一款针对电子元器件的基于 Web 的数据库管理系统，将极大地促进实验室电子元器件的精细化管理，提高管理效率。

1.2 研究现状

DBMS，即数据库管理系统，是专门用于建立、操作和维护数据库的大型软件。随着技术的不断发展，数据库管理已经历了从人工管理到文件系统，再到数据库管理系统的三个重要阶段。这些阶段反映了数据库技术不断成熟和完善的过程。

(1) 数据的产生可追溯到 60 多年前，初始阶段的数据形态相对简单。早期，人们通过处理大量穿孔卡片上的数据，利用专门的机器进行分类、对比和表格绘制。处理后的数据通常以打印或新卡片的形式呈现。这一阶段的数据管理主要围绕穿孔卡片的保存与处理进行。

(2) 在 50 年代至 60 年代中期，数据管理主要通过文件系统实现。文件系统作为一种机械手段，能够快速检索打卡信息。随着第二代编程语言（汇编语言）的兴起，程序员开始使用字母表中的字母进行编程，提高了程序的可读性，减少了繁琐且易出错的计算过程。因此，汇编语言成为早期用于组织和管理数据的方法。在这一时期，数据主要存储在磁盘上，初步支持了联机实时处理和批处理，并且数据可以长期保存。然而，数据冗余较多，共享性和独立性较差，尚未实现系统性的数据管理。

(3) 在 20 世纪 60 年代，数据库系统开始崭露头角，随着计算机技术的广泛应用，数据共享的需求愈发迫切。传统数据管理手段已难以应对这种不断增长的需求，于是，数据库管理系统应运而生，它实现了数据的统一管理和高效共享。其中，数据库模型作为该系统的核心要素，为多样化的数据库软件奠定了统一的基础框架。根据这些模型的不同特点，数据库通常被划分为网状、层次、关系和面向对象这四大类别^[6]。这些分类体现了数据库在结构和管理方式上的多样性，满足了不同领域对数据管理的特定需求。

1961 年，通用电气公司首创了网状数据库，但早期版本受限于只能在自家主机上运行，且基于人工编码。后来，古立德公司对其进行了改进和重新命名。网状数据库因其对层次和非层次结构的自然模拟而具有重要意义。而层次数据库随后出现，由 IBM 公司于 1968 年开发，专为主机环境设计。经过四十多年的发展，

这类数据库如 IMS V6，持续提供数据群集、共享和消息队列等功能，在商务智能和网络应用中持续发挥作用。

尽管网状数据库和层次数据库在数据集中和共享方面表现出色，但在处理数据独立性和抽象性时仍显不足^[7]。而关系数据库则有效弥补了这一缺陷。1970 年，IBM 的科德博士首次提出了关系模型的概念，为关系数据库的发展奠定了坚实基础。随后，在 1976 年，霍尼韦尔公司成功研发了首个关系数据库。这种数据库基于关系代数理论，经过长时间的应用和不断优化，技术逐渐成熟，成为数据管理的重要工具^[8]。

自 20 世纪 80 年代起，随着数据管理需求的演变，数据库技术再次取得了显著进展，特别是在面向对象数据库系统 OODBS，即 Object Oriented Database System 的研究与发展上取得了重要突破^[9]。

当前市场上，数据库管理系统产品种类繁多，其中 Oracle、IBM、Microsoft、Sybase 和 MySQL 等公司的产品因其卓越的性能和广泛的应用而备受欢迎^[10]。

1.3 研究内容

研究内容主要为电子元器件管理系统的设计、实现与部署。电子元器件管理系统的开发旨在服务于各大高校实验室，达到科学管理、合理采购、便于追踪、减少浪费的目的。系统一般用户角色包括管理员、仓管员、普通用户。其中管理员可以添加新角色，并给该角色分配任意菜单权限，再给用户分配角色，使其获得角色相应权限。系统应具有元器件分类管理、元器件采购管理、元器件领用管理、日志管理、数据统计等基本功能。管理员拥有所有功能权限。包括维护元器件类型，对新增的采购申请和领用申请进行审批，查看元器件去向，管理用户信息；仓管员可以维护元器件信息，审批元器件的领用申请和采购申请，但没有管理用户的权限；普通用户则只有申请领用和申请采购的权限，不能进行审批或管理用户。

1.4 论文主要内容和组织架构

本文具体内容有以下章节阐述：

第一章，绪论。从电子元器件管理的背景和现状出发，介绍在当前背景下开发电子元器件管理系统的意义，简述了该论文的主要研究内容。

第二章，开发环境与关键技术。陈述了系统开发的环境和搭建系统所使用的主要工具与框架技术。

第三章，需求分析。针对电子元器件管理系统的一般业务流程进行需求分析，包括功能分析和安全性、易用性、可维护性考量。

第四章，系统详细设计。陈述系统的设计模式和体系结构，以及前端页面设计。并详细介绍了数据表的设计，包括 E-R 图的绘制。

第五章，功能模块的设计与实现。详细陈述了系统各个功能模块的前后端接口的开发、后端与数据库的交互方式。

第六章，界面展示与平台部署。展示了系统运行时的前端页面与功能，并介绍了如何配置运行环境，进行本地部署。

第七章，结论。对此次毕业设计进行总结，说明该系统仍存在的缺陷与不足，并提出改进方向。

2 开发环境和关键技术

2.1 开发环境

硬件环境：Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz + 16G 内存 + 500G 硬盘。

开发环境：jdk17.0 + node.js20.11.0 + redis5.0.14.1 + apache-maven-3.8.4

开发工具：IDEA2022.1.4 + MySQL+Visual Studio2022。

2.2 开发工具 IDEA

IntelliJ IDEA，由 JetBrains 公司精心打造，是一款功能强大的集成开发环境（IDE），特别适用于 Java 开发，同时亦支持多语言与技术栈的项目^[11]。在软件开发界，这款 IDE 广受好评，其用途广泛，能够支持从桌面应用、Web 应用到移动应用，甚至大数据和云计算应用等多类型应用的开发。以下是 IntelliJ IDEA 的几个显著特点和其带来的优势：

(1) 智能编码助手：IntelliJ IDEA 的智能编码助手功能能够基于开发者的代码上下文和项目需求，提供实用的编码建议，帮助开发者编写出更加高效、精确的代码。

(2) 灵活的插件体系：该 IDE 拥有强大的插件体系，允许开发者根据项目需要自由扩展和定制开发环境。无论是集成新的框架、库还是工具，都能通过插件轻松实现。

(3) 构建与版本管理集成：IntelliJ IDEA 无缝集成了领先的构建工具，如 Maven 和 Gradle，大大简化了项目的构建过程。此外，它还支持如 Git 等版本控制系统，为开发者提供了便捷的代码库管理方式^[12]。

(4) 调试与性能优化工具：这款 IDE 提供了强大的调试功能，允许开发者逐行跟踪代码并查看变量值。此外，其内置的性能分析工具也能帮助开发者识别和修复应用程序中的性能瓶颈。

2.3 MySQL 数据库

MySQL, 作为一款开源的关系型数据库管理系统, 起源于瑞典的 MySQL AB 公司, 如今在 Oracle 公司的维护下继续发展。它支持跨平台使用, 包括 Linux、Windows、macOS 等操作系统, 采用客户端/服务器架构, 为用户提供高效、可靠且稳定的数据存储与管理服务^{[13][14]}。MySQL 在开源数据库市场中占据显著地位, 尤其在 Web 应用、企业应用和移动应用等多个领域有着广泛的应用。其主要优势如下:

- (1) 高性能表现: MySQL 的运行速度非常快, 为用户提供了高效的数据处理能力。
- (2) 成本效益: 对于大多数个人用户而言, MySQL 是免费的, 降低了使用成本。
- (3) 简易学习与管理: 相比其他大型数据库, MySQL 的设置与管理流程较为简化, 学习门槛较低, 易于上手。
- (4) 跨平台兼容性: MySQL 能够在多种操作系统上稳定运行, 如 Windows、Linux、Unix、macOS 等, 为用户提供了灵活的部署选择^[15]。
- (5) 丰富的接口支持: MySQL 提供了多种编程语言的 API 接口, 如 C、C++、Java、Python 等, 方便开发者进行集成开发。
- (6) 标准 SQL 支持: MySQL 遵循标准 SQL 语法, 并支持 ODBC (开放式数据库连接), 使得应用程序能够轻松与其交互。

2.4 Vue.js 技术

Vue.js 是一个构建数据驱动的 web 界面的渐进式框架^[16]。其核心理念在于简化数据响应式绑定和视图组件的集成过程, 核心机制依赖于一个响应式数据绑定系统。Vue.js 以其双向数据同步、模块化组件构建和优化的虚拟 DOM 处理机制著称^[17], 能够显著提高 Web 应用开发速度, 尤其是单页面应用(SPA)的开发效率。

2.5 Springboot 框架

SpringBoot 作为一种创新的框架规范, 其核心价值在于简化了 Spring 生态系统中繁杂的配置流程。通过减少大量配置文件的编写, SpringBoot 为开发者提供了更加便捷的框架使用体验, 堪称一种“框架的助手”^[18]。其特点包括快速启动项目、内置 Web 服务器、广泛的适用性、自动化的配置选项以及强效的应用监

控能力。SpringBoot 的推出显著提升了后端开发的效率，成为开发者不可或缺的工具之一。

2.6 MyBatis-Plus

MyBatis-Plus，简称 MP，是对 MyBatis 的一个强化工具，旨在通过增强而非改变 MyBatis 的基础功能，来简化开发流程并提升开发效率。后端框架集成 MyBatis-Plus 具备以下主要优势：

(1) 简易的 CRUD 功能：集成通用的 Mapper 和 Service，仅需少量设置即可执行单表的大部分 CRUD 操作，并提供灵活的条件构造器以满足各种需求^[19]。

(2) Lambda 表达式支持：采用 Lambda 表达式，简化查询条件的编写，确保字段准确性。

(3) 多样化的主键生成：支持四种主键生成策略（包括分布式唯一 ID 生成器），灵活配置，解决主键难题。

(4) ActiveRecord 风格：提供 ActiveRecord 调用方式，通过继承 Model 类，实体类即可实现强大的 CRUD 功能。

(5) 自定义全局操作：支持全局通用方法的灵活注入。

2.7 Redis

Redis，全称 REmote DIctionary Server。是一个由 Salvatore Sanfilippo 开发的 key-value 存储系统，作为一个跨平台的非关系型数据库，在本次系统开发中扮演了关键的数据缓存角色，特别是用于验证码功能。Redis 具备以下显著优势：

(1) 高效性：作为一款内存数据库，Redis 具备极高的读写速度和并发处理能力，特别适用于处理如验证码这样的实时数据需求^[20]。

(2) 分布式能力：Redis 支持分布式架构，允许在不同服务器间轻松实现数据共享，为验证码的分布式管理提供了可能。

(3) 数据持久性：通过提供多种持久化选项，Redis 确保数据即使在系统重启后也能保持完整性和可靠性，这对于验证码等关键数据尤为重要。

(4) 灵活数据结构：Redis 支持多种数据结构，如字符串、哈希、有序集合等，使得开发者能更灵活地处理验证码相关的数据需求。

3 需求分析

3.1 业务流程分析

本系统的应用场景为高校实验室的电子元器件管理。目标用户包括管理员、仓管员、普通用户等，不同用户应当具备不同权限。基本业务流程包括：

(1) 用户的注册与登录。注册后用户信息自动录入系统，管理员和用户本人可维护个人信息。新用户注册时会获得部分默认权限。

(2) 采购电子元器件需提交采购申请单，由管理员审批，通过后元器件信息回写元器件档案。

(3) 所有用户可查看元器件库存，提交领用申请，管理员、仓管员审批申请。

(4) 管理员维护元器件档案数据，实现各型号元器件分类管理。

业务流程图如图 3-1 所示：

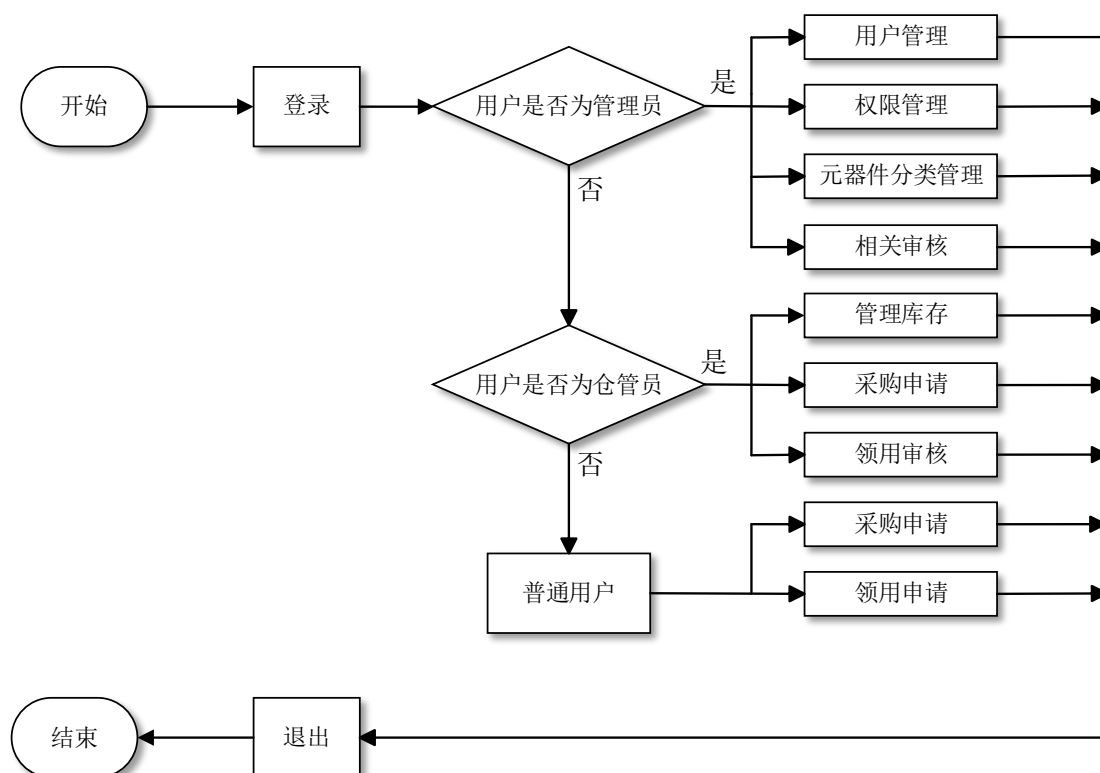


图 3-1 业务流程图

3.2 功能需求分析

电子元器件基本功能应包括电子元器件的采购、申领、分类管理、审批。基本操作包括添加、删除、修改等。除外，系统还应具备日志管理、数据统计功能，便于追踪平台用户使用历史和可视化分析采购支出。

3.2.1 登录与注册

系统应存在原始管理员用户，拥有系统所有功能权限。新用户可在登陆页面点击注册创建账户，管理员也可通过用户管理功能创建新用户。创建新用户时要求输入用户手机号码作为该用户账号，手机号格式应进行校验，当不符合格式会提示用户“输入格式错误”。为加强系统安全性，登录页面应设置验证码区分在页面上进行输入操作的是人还是机械化的代码编程。用户密码不能在数据库中直接显示，这直接威胁到数据库安全，应对密码加密后再存储。

已存在的用户输入账号、密码、验证码均正确后点击登录跳转系统主页，若输入的账户信息不匹配，系统会提示“输入账号或密码错误”。若用户忘记密码，无法自行找回，需求助管理员重置用户密码。

3.2.2 主页功能

系统主页设置快捷功能模块，用户可自定义主页的常用功能，点击即可跳转该功能页面，无需在菜单栏查找，便于用户使用。

3.2.3 权限管理

系统初始用户应拥有系统所有功能权限，并可以管理其他用户权限。新用户注册时应自动获得部分权限。当用户登陆成功进入系统，菜单栏仅显示该用户拥有权限的功能模块。管理员可创建不同角色，并给角色自由分配菜单权限。

3.2.4 部门管理

高校的实验室会提供给不同学院专业使用，用户包括本科生、研究生、教师等，身份复杂。为便于统计，按部门进行用户管理。基本功能应包括部门的新增、修改、删除。当部门下存在用户时，部门无法删除。可查询各部门下属用户。

3.2.5 用户管理

系统应有用户管理功能。管理员可使用该功能模块启用或禁用现存用户；可重置用户密码；更改用户角色身份、邮箱、性别、部门等信息。用户信息可以批量导出为 Excel 表格或直接被打印。

3.2.6 元器件分类管理

电子元器件种类繁多，以电阻为例，下属又有贴片电阻、插件电阻、排阻等众多类别，每一类别又包含无数型号。系统应能实现元器件的分类清晰直观，便于用户查找。同时每一具体型号的元器件应包含规格、封装等基本信息，便于用户在领用时检索到自己所需的元器件。元器件分类可实现添加、删除、修改。在添加时给予管理员提示，若元器件分类已存在系统提示“该元器件分类已存在，请勿重复添加！”。若某一型号元器件库存大于 0，则无法删除该项分类，系统会提示“仓库中存在此元器件，无法删除！”。要求支持电子元器件图片的上传和预览，可查询元器件存储位置。仓管员可维护元器件仓库信息。

3.2.7 元器件采购管理

当用户查询到元器件库存不足时可以新增元器件采购单，实现从已有元器件列表导入需采购的型号，选择数量、价格、对应供应商后可提交申请。供应商信息由管理员维护。管理员界面会收到提交的申请，可审批通过或驳回。用户可查看自己的申请进度，显示“未提交”、“已提交”、“申请通过”、“申请驳回”。若要采购现存元器件档案还没有的型号，需先在元器件分类功能中添加该型号。采购申请通过后，元器件库存自动更新。所有采购申请可导出为 Excel 表格，也可直接打印网页。

3.2.8 元器件领用管理

用户可按所需元器件名称或型号检索元器件，点击申请领用即可申请，用户可查询申请的审核进度，管理员可审批所有的领用申请。申请通过后，元器件库存会自动更新。所有元器件的去向应被记录，包括是否被申请、是否被领用及领用人、领用时间、审核人的信息等。

3.2.9 数据统计

系统应能统计采购支出和数量并可视化，以图表展示，便于比较各类元器件采购支出，及不同型号元器件受欢迎程度。

3.2.10 日志管理

针对元器件模块的用户操作数据，系统需实现自动化记录功能。管理员能够在后台便捷地检索这些记录，日志应详尽包含执行操作的人员、具体的时间点以

及操作的具体内容。

3.3 其他需求分析

3.3.1 系统安全性

该数据库管理系统应遵循一般软件设计的安全性原则。具体需求包括：

(1) 用户身份验证

(a) 系统中每一位用户都应有唯一标识，不允许同一用户的多端登录。

(b) 限制用户对资源的访问，例如用户信息、系统文件等信息不对普通用户暴露。

(2) 信息安全性

(a) 避免在代码中存储机密信息，即 Web 服务器上应完全杜绝源代码的暴露。

(b) 捕捉异常，并给出合理、正确的提示信息，免于系统被攻击。

(c) 不在网络上以纯文本形式发送密码，密码存储到数据库应经过算法加密。

(d) 对所有输入的文本进行校验，确保其符合该信息的格式要求，防御恶意的输入和攻击。

3.3.2 系统可维护性

系统的代码结构应清晰、可读性强。添加适量的注释。便于后续系统若投入使用，可进行二次开发新增功能。当系统运行出现错误，应对开发者做出提示，便于开发者修复 bug。

3.3.3 系统易用性

本系统着重于提供出色的用户体验，确保系统易于理解、操作简便且高效。为实现这一目标，系统的模块设计力求合理，旨在缩短用户熟悉系统所需的时间，并提升操作与运行效率。此外，浏览内容的设计旨在使用户能够快速获取所需信息，避免在查找过程中消耗不必要的时间。

4 系统详细设计

4.1 软件设计模式

软件的前端开发使用 vue.js 框架，该框架的核心架构为 MVVM 设计模式，MVVM 是 Model-View-ViewModel 的简写。它本质上就是 MVC，即 Model-View-Controller 的改进版。M、V、C 分别表示数据、视图、控制器，工作模式如图 4-1 所示：

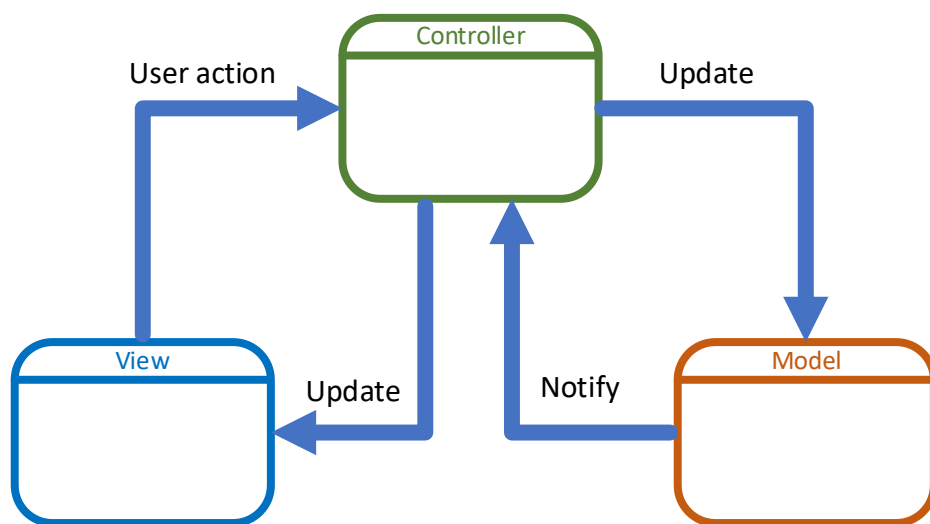


图 4-1 MVC 工作模式

而 MVVM 相比 MVC，多一个 ViewModel，其工作模式如图 4-2 所示：

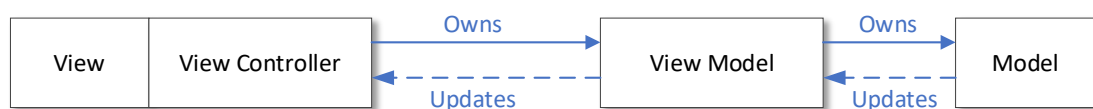


图 4-2 MVVM 工作模式

在 MVVM 架构中，View Model 的角色是将 MVC 架构中 Controller 所承担的数据和逻辑管理功能分离出来，作为一个单独的组件来运作。这种做法显著降低了 Controller 的代码复杂度，通过数据绑定机制将 Controller 与 View Model 连接，不仅提高了系统的可测试性，也使得系统维护更为便捷^[21]。

4.2 系统体系结构

依据系统功能需求，系统总体设计体系如图 4-3 所示。

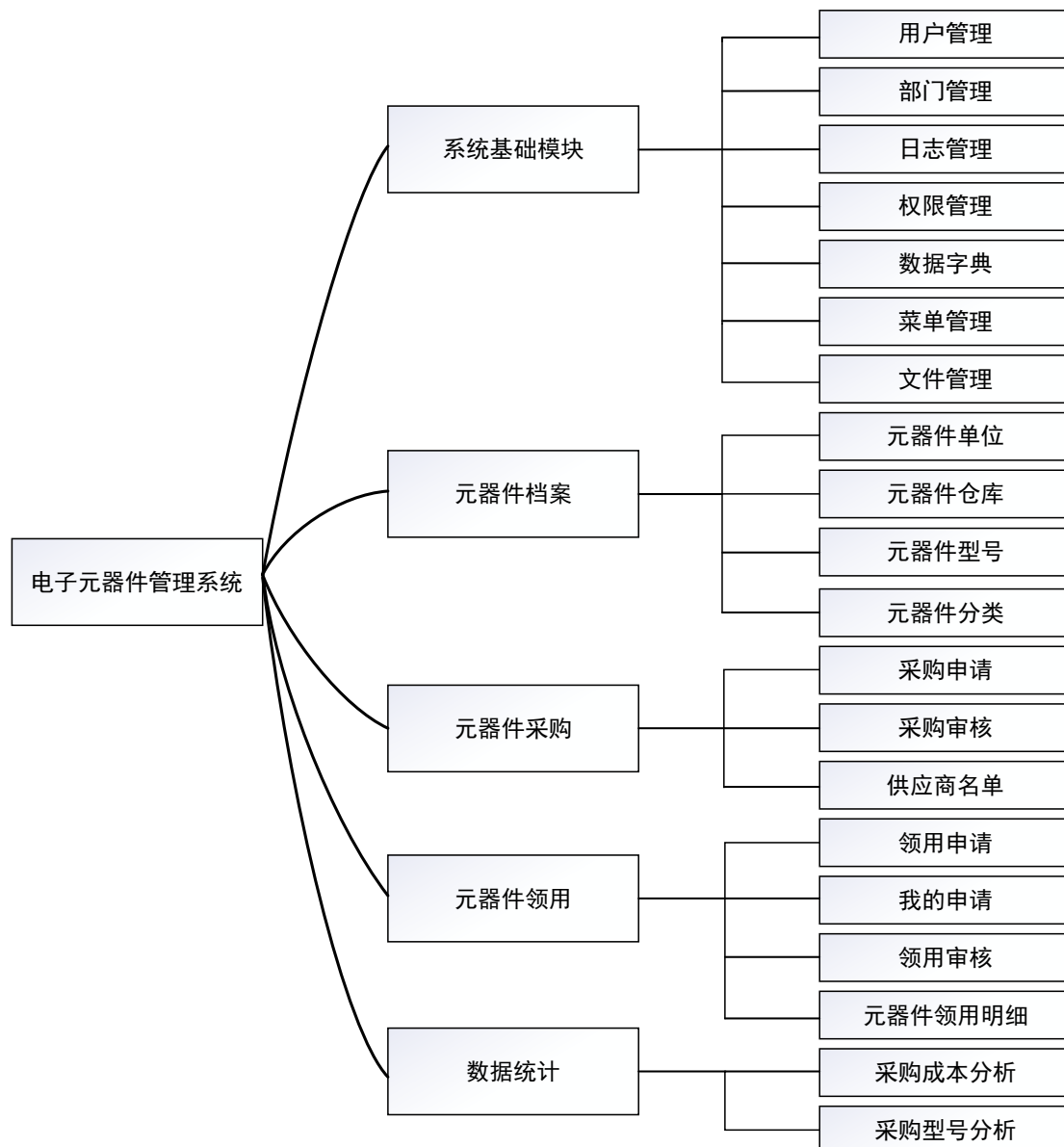


图 4-3 系统总体设计体系

(1) 将与电子元器件管理无关的功能模块全部放置在系统基础模块，涵盖了用户管理、权限管理、部门管理、日志管理等。还加入菜单管理，便于二次开发添加新页面；文件管理，存储系统用户上传的所有文件，如用户头像、元器件图片等；数据字典通过为每个数据元素提供详细的定义、名称、含义和用途等信息，帮助用户理解和使用数据。它可以减少数据解释和沟通上的误解，保持数据的一致性和标准化。元器件大类提前写入数据字典，在元器件管理时可以通过数据字典中的提示进行分类。

(2) 元器件档案模块实现元器件从类别到型号的分类。元器件单位用于新增元器件型号时提供单位数据支持。元器件仓库用于采购元器件入库和查询元器件库存时提供数据支持，包含仓库名称和位置信息。

(3) 元器件采购模块中，用户可在采购申请页面新增和提交采购申请、查询审核进度。管理员可在采购审核页面进行审批。供应商名单为提交采购申请时选择供应商功能提供数据支持。

(4) 元器件领用模块中，用户可在领用申请页面查看库存的所有元器件，并可使用名称或具体型号查询到所需元器件。我的申请页面可查看申请的审批进度。管理员在领用审核页面对所有申请进行审批。元器件领用明细记录了所有元器件的去向，包括申请、领用状态及时间、审批人等，页面可直接被打印。当库存元器件意外损毁，管理员可在此页面更新元器件状态。

4.3 前端界面设计

在考察市场上常用的 erp 系统后，考虑使用的便捷性，前端页面采用常见的顶端标题、菜单栏、内容式设计，如图 4-4 所示。头部标题左侧显示系统名称，右侧显示锁屏、全屏功能按钮。个人账户处显示用户头像及身份。菜单栏显示五大功能模块，点击具体功能模块时展开显示该模块的所有功能，选择功能后内容变换为该功能页面。导航栏显示所有已打开的功能页面，主页面始终置顶，可点击导航栏页面名称跳转至该页面。头部标题、个人账户、菜单栏颜色均设计为深灰色，内容处为白色。



图 4-4 前端界面设计

4.4 数据表设计

根据系统需求分析，依照系统数据流和一般流程特点，设计系统数据库 E-R 图和数据表。系统总体 E-R 图如图 4-5 所示。每个实体属性众多，故在此 E-R 中省略，由后续具体实例的 E-R 图展示。

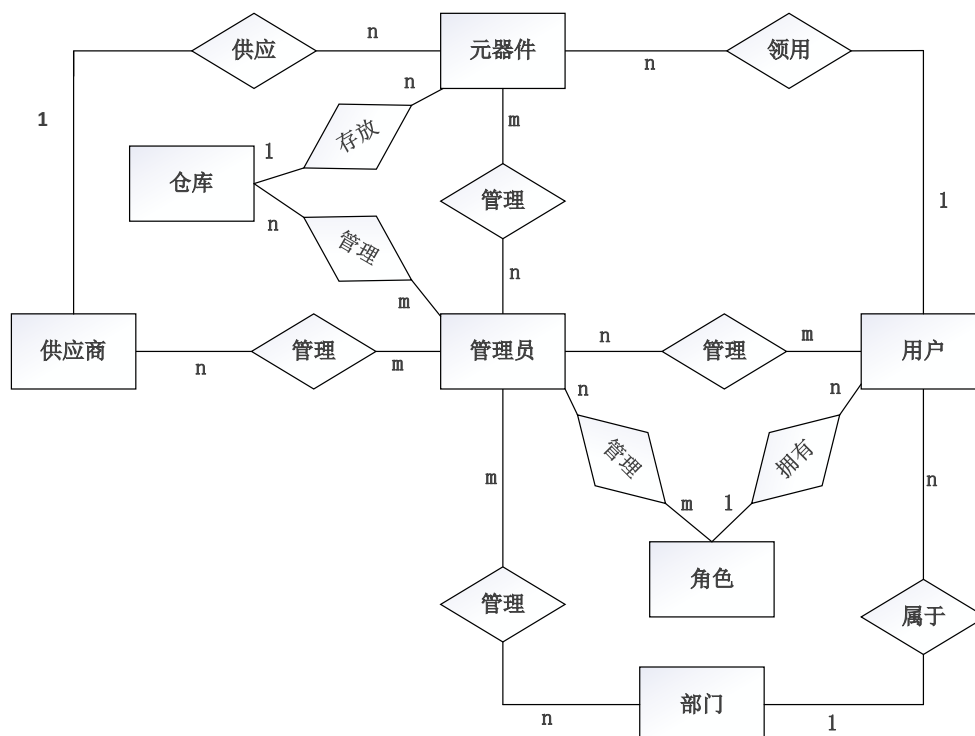


图 4-5 电子元器件管理系统总体 E-R 图

4.4.1 用户表

系统用户表的设计应包括用户基本信息，如性别、昵称、账号、密码、联系方式、所属部门等。为确保用户与使用者一一对应，创建用户时账号默认为手机号码，所有用户手机号码、邮箱、账号均唯一，不可重复。用户 id 为用户表主键，分配用户角色时使用用户 id 与角色 id 一一对应。

用户 E-R 图如图 4-6 所示。

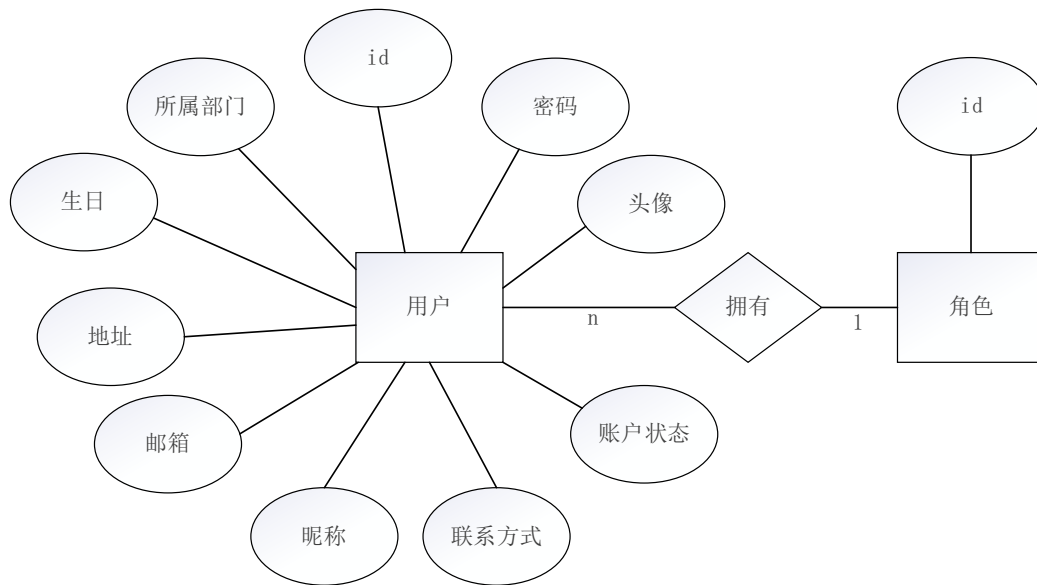


图 4-6 用户 E-R 图

用户数据表的设计采取双表，用户个人基本信息均囊括在用户表 a_user 内，具体字段与类型如表 4-1 所示。

表 4-1 用户表 a_user

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
address	varchar(255)	地址
avatar	varchar(255)	头像
description	varchar(255)	描述
email	varchar(255)	邮箱，唯一
mobile	varchar(255)	联系方式，唯一
nickname	varchar(20)	昵称
password	varchar(255)	密码
sex	varchar(255)	性别

续表 4-1

字段	类型	备注
status	int	是否启用
type	int	用户类型(管理员/普通用户)
username	varchar(255)	账号, 唯一
department_id	varchar(255)	所属部门 id
department_title	varchar(255)	所属部门名称
birth	datetime(6)	生日

用户信息中涉及另一实体“角色”的数据由用户角色表 a_user_role 管理, 具体字段与类型如表格 4-2 所示。

表 4-2 用户角色表 a_user_role

字段	类型	备注
id	varchar(255)	主键, 自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
role_id	varchar(255)	角色 id
user_id	varchar(255)	用户 id

4.4.2 部门表

部门表采取级联设计, 每个部门可有下属部门, 属性包含父部门 id 和是否有子部门标识。部门 id 为主键, 自增。部门名称唯一, 不可重复。

部门 E-R 图如图 4-7 所示。

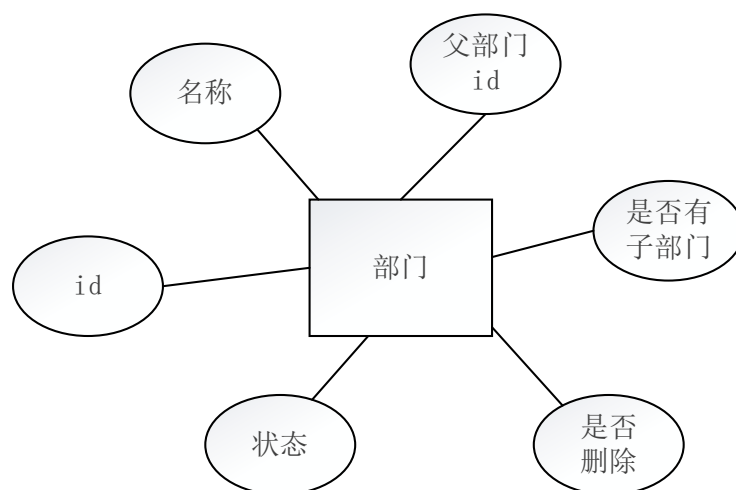


图 4-7 部门 E-R 图

部门表具体字段、类型设计如表 4-3 所示。

表 4-3 部门表 a_department

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
parent_id	varchar(255)	父部门 id
status	int	是否启用
title	varchar(255)	部门名称
is_parent	bit	是否有下属部门

4.4.3 角色与权限表

权限的管理通过设计角色表和权限表实现。权限表数据包括主键自增 id、名称、对应路径、父权限 id、级别、图标等信息；角色表数据包括主键自增 id、角色名、描述、是否默认。角色 id 和权限 id 均为两笔中唯一标识，通过 n 个权限 id 与角色 id 对应形成角色权限表，使角色获得权限。再将角色 id 赋予用户 id，使用户获得该角色身份，拥有相应权限。

角色-权限 E-R 图清晰的展示了角色与权限的关系及属性，如图 4-8 所示。

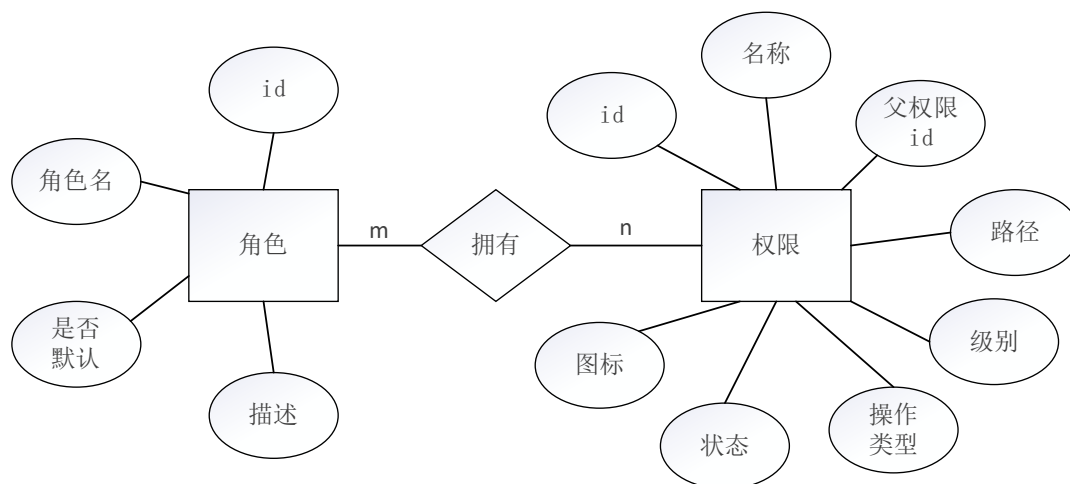


图 4-8 角色-权限 E-R 图

角色表具体字段、类型设计如表 4-4 所示。

表 4-4 角色表 a_role

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间

续表 4-4

字段	类型	备注
update_time	datetime(6)	更新时间
name	varchar(255)	角色名
update_by	varchar(255)	更新人
default_role	bit	是否为默认角色
description	varchar(255)	描述

角色权限表具体字段、类型设计如表 4-5 所示。

表 4-5 角色权限表 a_role_permission

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
permission_id	varchar(255)	权限 id
role_id	varchar(255)	角色 id

权限表具体字段、类型设计如表 4-6 所示。

表 4-6 权限表 a_permission

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
name	varchar(255)	英文名称
parent_id	varchar(255)	父权限 id
sort_order	decimal(10, 2)	排序值
component	varchar(255)	对应位置
path	varchar(255)	路径
title	varchar(255)	权限名称
icon	varchar(255)	图标
level	int	级别
button_type	varchar(255)	操作类型
status	int	状态

4.4.4 数据字典表

由于元器件种类繁多，不同管理员在分类命名时极易出现混乱，难以统一。

数据字典表的设计是为了减少数据解释和沟通上的误解，保持数据的一致性和标

准化。将部分基本不变的数据类别建立数据字典，如性别仅含男、女、保密，元器件种类大体可分为电阻/电容/电感、连接器/开关等若干类。

数据字典 E-R 图如图 4-9 所示。

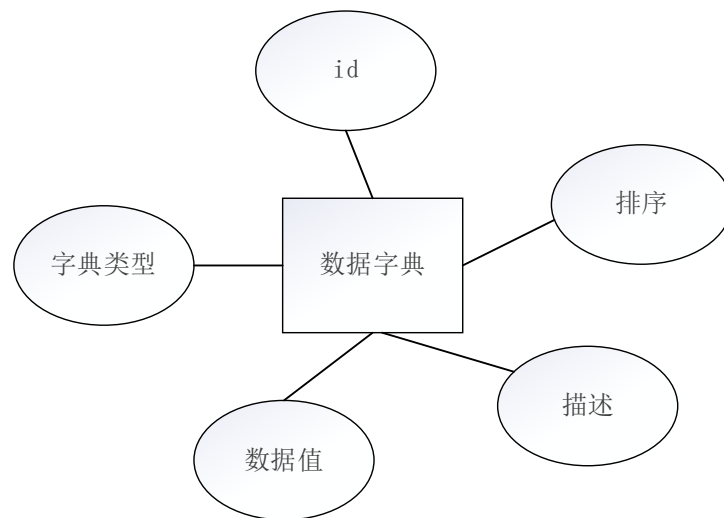


图 4-9 数据字典 E-R 图

数据字典主体表的具体字段、类型设计如表 4-7 所示，此表维护了数据字典的类别，如性别、元器件大类、分权限等。管理员可自行添加新的字典类型，该表主要数据包括字典名称、主键自增 id、类型等。

表 4-7 数据字典主体表 a_dict

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
description	varchar(255)	描述
title	varchar(255)	名称
type	varchar(255)	类型
sort_order	decimal(10,2)	排序值

数据字典数据表的具体字段、类型设计如表 4-8 所示，每个类型的数据字典的具体数据由此表维护，管理员可对数据进行添加、删除、修改操作。

表 4-8 数据字典数据表 a_dict_data

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间

续表 4-8

字段	类型	备注
dict_id	varchar(255)	数据字典 id
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
description	varchar(255)	描述
sort_order	decimal(10,2)	排序值
status	int	状态
title	varchar(255)	名称
value	varchar(255)	数据值

4.4.5 元器件档案表

元器件分类采取级联设计，实现大类—类别—名称—型号四级分类。元器件类型表完成大类—类别的分类，主要数据包括主键自增 id、类型编码、备注类型名称、大类等。其中大类的数据值由数据字典提供支持。元器件型号表完成类别—名称—型号的分类，主要数据包括主键自增 id、元器件编码、规格、图片、名称、类别、大类、型号等。

元器件档案表的 E-R 图如图 4-10 所示。

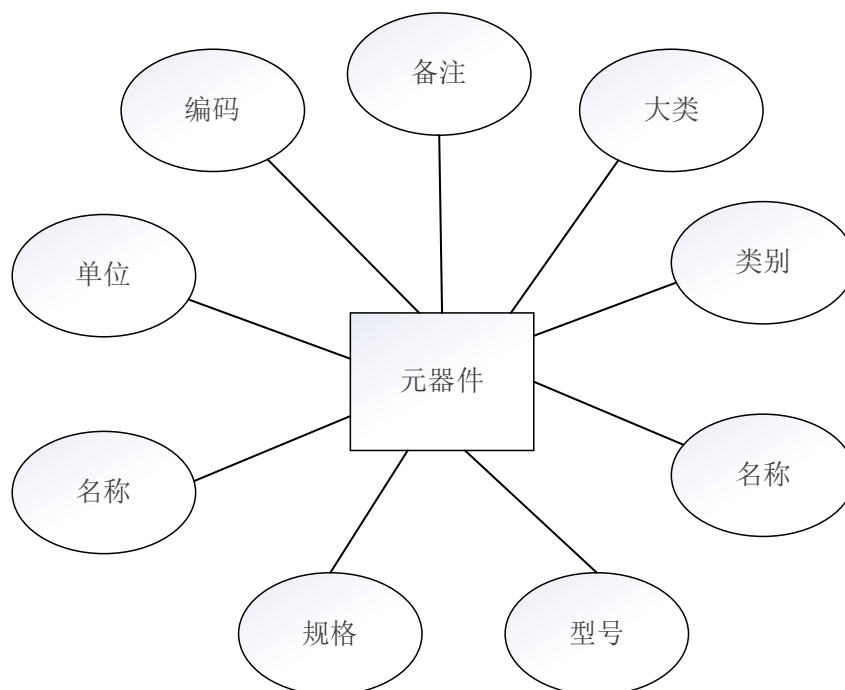


图 4-10 元器件档案 E-R 图

元器件类型数据表的具体字段、类型设计如表 4-9 所示。

表 4-9 元器件类型表 a_admin_asset_type

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
code	varchar(255)	类型编码
remark	varchar(255)	备注
sort_order	decimal(38,2)	排序值
title	varchar(255)	类别
type	varchar(255)	大类

元器件型号数据表的具体字段、类型设计如表 4-10 所示。

表 4-10 元器件型号表 a_admin_asset

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
code	varchar(255)	元器件编码
gg	varchar(255)	规格
image_url	varchar(255)	元器件图片
jldw	varchar(255)	计量单位
name	varchar(255)	元器件名称
remark	varchar(255)	备注
status1	varchar(255)	采购状态
status2	varchar(255)	物料状态
type	varchar(255)	类别
type2	varchar(255)	大类
xh	varchar(255)	型号

4.4.6 元器件领用表

元器件的领用数据表的设计用于实现领用申请单的生成、审批状态的记录和领用去向的记录。主要数据包括主键自增 id、申请时间、申请状态、申请人、型号 id、采购信息、领用人信息、元器件状态、存放仓库信息等。

元器件领用表的 E-R 图如图 4-11 所示。

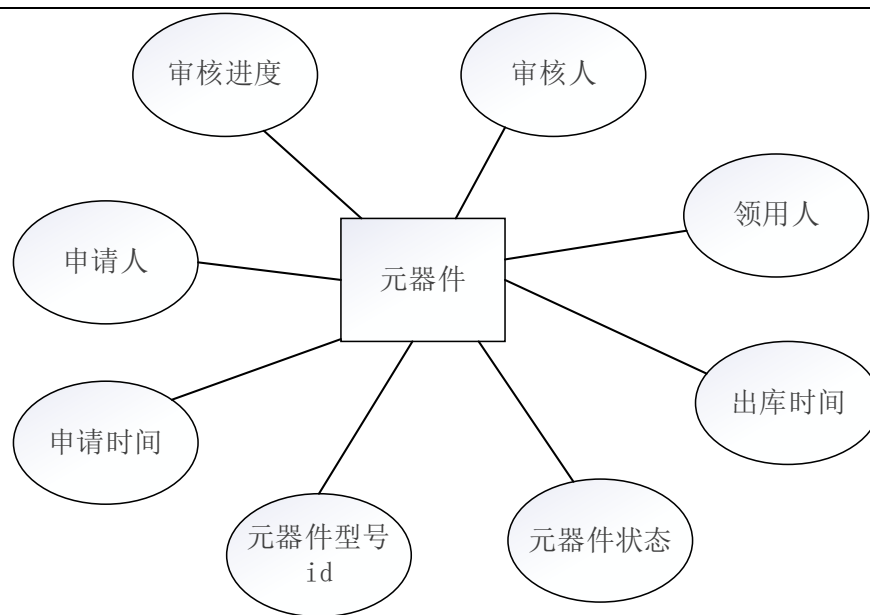


图 4-11 元器件领用表 E-R 图

元器件领用表的具体字段、类型设计如表 4-11 所示。

表 4-11 元器件领用表 a_admin_assets

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
apply_flag	int	是否被申请
apply_time	varchar(255)	申请时间
apply_user	varchar(255)	申请人
asset_id	varchar(255)	型号 id
buy_date	varchar(255)	采购日期
buy_user	varchar(255)	采购人
code	varchar(255)	元器件编码
destroy_name	varchar(255)	销毁人
destroy_time	varchar(255)	销毁时间
give_id	varchar(255)	领用用户 id
give_job	varchar(255)	领用人账号
give_name	varchar(255)	领用人昵称
give_type	int	领用人身份类型
model	varchar(255)	元器件型号
name	varchar(255)	元器件名称
out_flag	int	是否出库
out_time	varchar(255)	出库时间
out_work	varchar(255)	出库审核人

续表 4-11

字段	类型	备注
status	int	状态
unit	varchar(255)	单位
warehouse	varchar(255)	仓库 id
warehouse_name	varchar(255)	仓库名

4.4.7 元器件采购表

元器件采购管理的数据表设计应包含采购发起人、发起时间、采购具体型号、采购数量、采购金额、供应商信息、付款账户、审核进度、审核人、审核时间、入库情况等主要数据。要求该表包含的数据能支持系统完成新增采购单、申请审核、采购数据统计的功能。

采购表 E-R 图设计如图 4-12 所示。

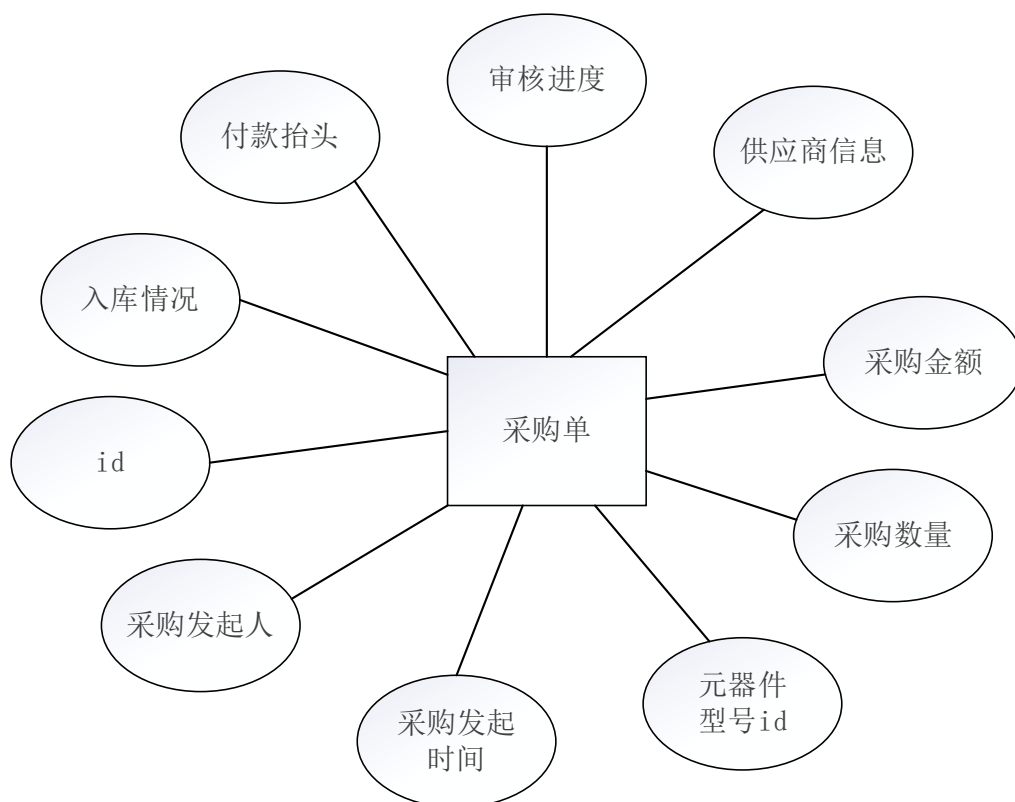


图 4-12 采购表 E-R 图

元器件采购表的具体字段、类型设计如表 4-12 所示。

表 4-12 元器件采购表 a_admin_assets_buy

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人

续表 4-12

字段	类型	备注
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
asset_code	varchar(255)	元器件型号编码
asset_id	varchar(255)	元器件型号 id
audit_flag	int	审核状态
model	varchar(255)	规格
money	decimal(38, 2)	采购总金额
money1	decimal(38, 2)	采购预付款
name	varchar(255)	元器件名称
open	varchar(255)	供应商开户行
push_name	varchar(255)	收款人姓名
quantity	decimal(38, 2)	采购总数量
quantity2	decimal(38, 2)	可入库数量
reason	varchar(255)	采购原因
rise	varchar(255)	付款抬头
submit_flag	int	是否提交审核
supplier_id	varchar(255)	供应商 id
supplier_name	varchar(255)	供应商名称
unit	varchar(255)	单位
user_code	varchar(255)	申请人账号
user_id	varchar(255)	申请人 id
user_name	varchar(255)	申请人昵称
ware_flag	int	是否入库

4.4.8 供应商表

供应商数据表为采购功能提供数据支持，由管理员维护，用户在发起采购时需要选择元器件供应商。根据一般采购流程，供应商信息应包含供应商公司名称、地址、付款银行信息、付款抬头、供应商官网链接等。

供应商实体 E-R 图如图 4-13 所示。

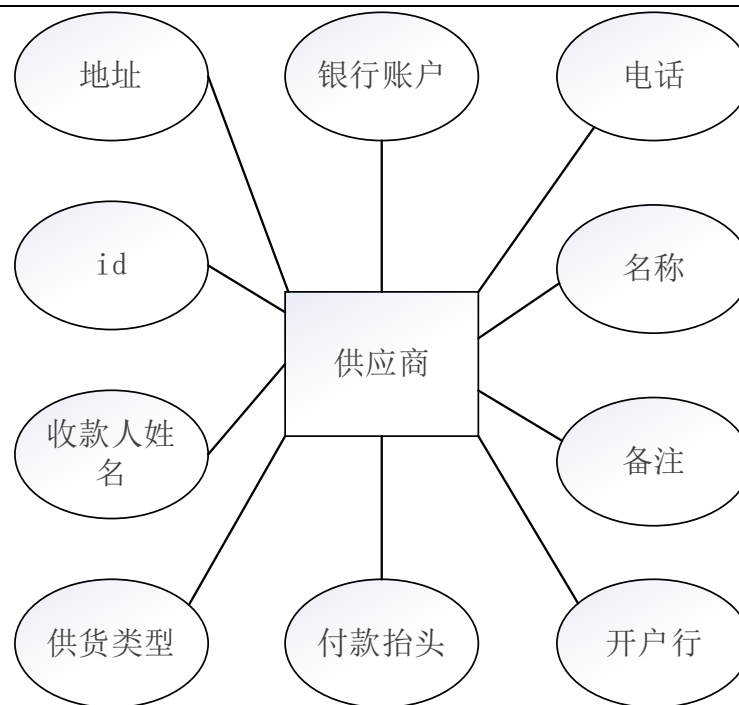


图 4-13 供应商表 E-R 图

供应商数据表具体字段、类型设计如表 4-13 所示。

表 4-13 供应商数据表 a_admin_asset_supplier

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
address	varchar(255)	地址
code	varchar(255)	银行账户
mobile	varchar(255)	电话
name	varchar(255)	名称
open	varchar(255)	开户行
remark	varchar(255)	备注
rise	varchar(255)	付款抬头
status	bit	状态
type	varchar(255)	供货类型
url	varchar(255)	链接
user_name	varchar(255)	收款人姓名

4.4.9 仓库表

仓库表的为入库功能提供数据支持。仓库表的设计应包含的数据有主键自增 id、仓库名称、地址、仓管员、仓库状态。管理员可新增、修改、删除仓库并给

仓库分配仓管员。当有电子元器件存放在仓库中，该仓库不得被删除。用户申请领用元器件时，只有元器件所存放的仓库仓管员有权限审批。

仓库实体的 E-R 图设计如图 4-14 所示。

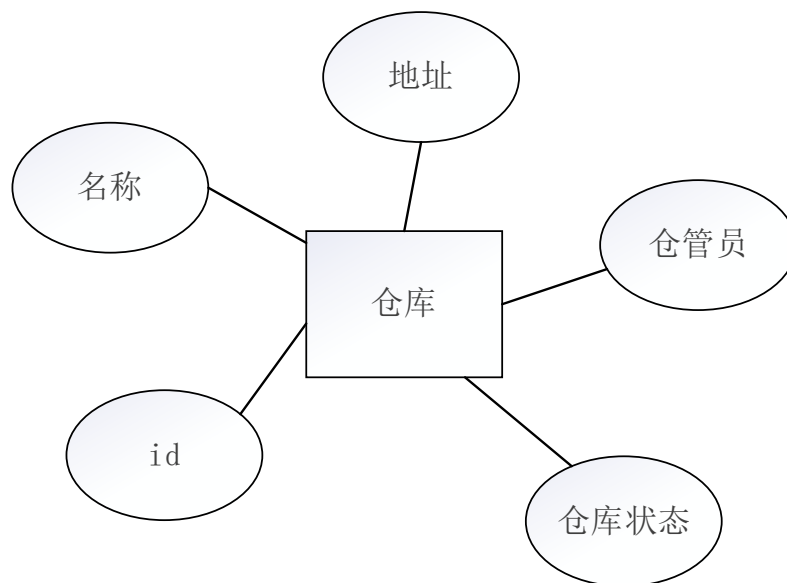


图 4-14 仓库实体 E-R 图

仓库表的具体字段、类型设计如表 4-14 所示。

表 4-14 仓库表 a_admin_asset_ware

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
address	varchar(255)	地址
admin_name	varchar(255)	仓管员
job_number	varchar(255)	仓管员账户
name	varchar(255)	仓库名称
remark	varchar(255)	备注
status	bit	状态

4.4.10 日志管理表

日志管理表的设计应记录用户登录系统的 ip、登陆方式、操作名称、请求参数、请求类型等数据。便于追踪系统使用记录。

日志管理的 E-R 图设计如图 4-15 所示。

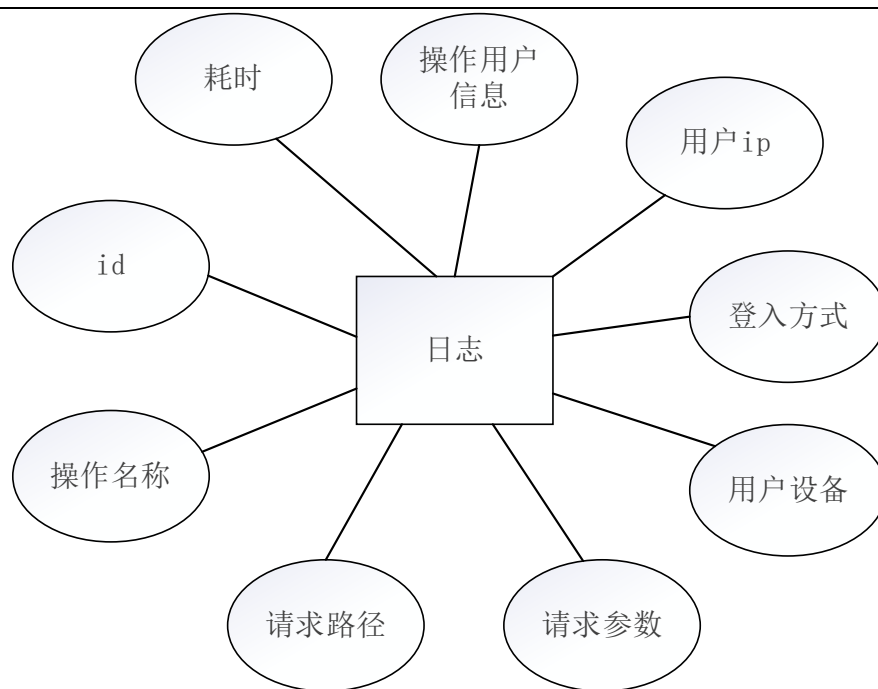


图 4-15 日志管理 E-R 图

日志管理表的具体字段、类型设计如表 4-15 所示。

表 4-15 日志管理表 a_log

字段	类型	备注
id	varchar(255)	主键，自增 id
create_by	varchar(255)	创建人
create_time	datetime(6)	创建时间
update_by	varchar(255)	更新人
update_time	datetime(6)	更新时间
cost_time	int	耗时
ip	varchar(255)	操作者 ip
ip_info	varchar(255)	ip 定位
name	varchar(255)	操作名称
request_param	text	请求参数
request_type	varchar(255)	请求类型
request_url	varchar(255)	请求路径
username	varchar(255)	用户昵称
log_type	int	登陆方式
device	varchar(255)	操作设备

4.5 功能模块的接口设计

4.5.1 登录与注册功能的接口设计

登陆功能是系统其他功能的前置条件。系统已存在的用户正确输入账号密码、验证码后可登入平台。若用户不存在，可点击“没有账号？点我注册”跳转注册页面创建新用户。

(1) 登陆功能页面包含账号输入、密码输入、验证码输入、登录按钮、注册按钮。账号输入框提示用户账号为默认注册手机号。账号、密码、验证码均有提示不能为空。验证码功能由 Springboot 集成 Redis 缓存实现。注册功能页面包括手机号输入组件、密码输入组件、姓名输入组件、验证码输入组件、注册按钮、返回登陆按钮。创建成功的用户账号会默认为手机号，且创建成功后系统会自动跳转登陆页面。当输入的手机号与现存用户账号重复时，系统会提示“登陆账号已存在”。

(2) 前后端数据交互通过 Axios 请求实现，后端进行账号、密码、验证码校验。校验成功实现登录成功回调，校验失败实现登陆失败回调。

登录与注册功能的接口如表 4-16 所示。

表 4-16 登录与注册功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
登录请求	/login	post	账号、密码、验证码	状态码，信息，token
注册请求	/user/regist	post	账号、验证码	状态码，信息

4.5.2 权限管理功能的接口设计

(1) 权限管理功能展示现存角色及其菜单权限。管理员可添加、删除、编辑角色，页面上方搜索框可按角色名称或描述搜索。操作框内的“菜单权限”可分配角色权限。当存在用户拥有某一角色身份时，该角色不能被删除，系统提示“不能删除正在使用的角色”。可将某一角色设为默认角色，新用户注册时会自动获取该身份。

(2) 前后端数据交互通过 Axios 请求实现，后端会根据请求返回角色列表、权限，实现对角色权限的更改。

权限管理功能的接口如表 4-17 所示。

表 4-17 权限管理功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
获取角色列表	/role/getAllByPage	get	无	状态码, 信息, 角色列表
获取权限列表	/permission/getAllList	get	无	状态码, 信息, 权限列表
添加角色	/role/save	post	角色名称、角色描述	状态码, 信息
编辑角色	/role/edit	post	角色名称、角色描述	状态码, 信息
删除角色	/role/delByIds	post	角色名称、角色描述、角色id	状态码, 信息
设置默认角色	/role/setDefault	post	无	状态码, 信息
更改角色权限	/role/editRolePerm	post	权限列表	状态码, 信息

4.5.3 部门管理功能的接口设计

(1) 部门管理可实现级联管理, 页面展示部门及其下属部门。可在部门用户中选择用户设置为部门领导, 点击部门可查询部门下属用户信息。搜索框可模糊查询部门。

(2) 部门管理功能的接口如表 4-18 所示。

表 4-18 部门管理功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
获取部门列表	/department/getByParentId	get	无	状态码, 信息, 部门列表
添加部门	/department/add	post	部门名称、排序值、是否启用	状态码, 信息
编辑部门	/department/edit	post	部门名称、排序值、是否启用	状态码, 信息
删除部门	/department/delByIds	post	部门名称、排序值、是否启用	状态码, 信息

续表 4-18

接口功能	接口路径	请求类型	请求参数	返回数据
获取部门下属用户	/user/getByDepartmentId	post	部门 id	状态码, 信息, 用户列表

4.5.4 用户管理功能的接口设计

(1) 用户管理页展示所有用户信息, 包括用户名、账号、性别、用户类型等信息。可编辑、删除、禁用用户, 还可重置用户密码。编辑用户中包括给用户分配角色使其在登陆时获得角色相应权限。

(2) 用户管理功能的接口如表 4-19 所示。

表 4-19 用户管理功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
获取用户信息	/user/getUserList	get	无	状态码, 信息, 用户列表
编辑用户	/user/admin/edit	post	用户信息	状态码, 信息
删除用户	/user/delByIds	post	用户 id	状态码, 信息
启用用户	/user/enable	post	无	状态码, 信息
禁用用户	/user/disable	post	无	状态码, 信息
重置密码	/user/resetPass	post	无	状态码, 信息

4.5.5 元器件分类管理功能的接口设计

(1) 元器件分类管理通过“元器件分类”和“元器件型号”两页面实现。“元器件分类”页面展示元器件大类及类别, 可实现对元器件类别的管理, 包括添加、删除、编辑元器件类别。可按类名搜索。“元器件型号”页面显示元器件具体信息, 包括元器件类别、名称、型号、规格、库存个数、图片等信息。可实现对元器件具体型号的添加、编辑、删除、搜索。

(2) 元器件分类管理功能的接口如表 4-20 所示。

表 4-20 元器件分类管理功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
获取元器件类别	/adminAssetType/getByPage	get	无	状态码, 信息, 元器件类别列表

续表 4-20

接口功能	接口路径	请求类型	请求参数	返回数据
新增元器件类别	/adminAssetType/insert	post	大类, 类别, 类型编码	状态码, 信息
编辑元器件类别	/adminAssetType/update	post	大类, 类别, 类型编码	状态码, 信息
删除元器件类别	/adminAssetType/delByIds	post	待删除元器件类别 id	状态码, 信息
获取元器件型号	/adminAsset/getByPage	get	无	状态码, 信息, 元器件型号列表
新增元器件型号	/adminAsset/insert	post	分类, 名称, 型号, 规格, 计量单位, 用途, 图片, 备注	状态码, 信息
编辑元器件型号	/adminAsset/update	post	分类, 名称, 型号, 规格, 计量单位, 用途, 图片, 备注	状态码, 信息
删除元器件型号	/adminAsset/delByIds	post	待删除元器件型号 id	状态码, 信息
查询元器件类型级联数据	/adminAssetType/getCascader	get	无	元器件类型级联数据

4.5.6 元器件采购管理功能的接口设计

(1) 元器件采购功能模块设计为三个页面, 分别为“采购申请”、“采购审核”、“供应商名单”。采购申请页面显示当前登录用户所有创建的采购单状态, 包括是否提交状态、审核状态、入库状态。用户可在采购申请页面新增申请, 可从档案中导入需采购的元器件, 填写数量金额后提交审核。供应商信息由供应商名单页面维护。通过的审核可一件入库。仓库信息由元器件仓库页面提供支持。“采购审核”页面可基于元器件名称查找采购单, 管理员在此页面进行审批, 有批量审批功能。

(2) 元器件采购管理功能的接口如表 4-21 所示。

表 4-21 元器件采购管理功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
获取我的采购申请	/adminAssetsBuy/getByPage1	get	无	状态码, 信息, 采购单列表
新增采购单	/adminAssetsBuy/insert	post	元器件型号, 元器件型号编码, 采购数量, 预付款, 总付款	状态码, 信息
编辑采购单	/adminAssetsBuy/update	post	元器件型号, 元器件型号编码, 采购数量, 预付款, 总付款	状态码, 信息
删除采购单	/adminAssetsBuy/delByIds	post	采购单 id	状态码, 信息
提交审批	/adminAssetsBuy/submitData	post	供应商, 采购原因	状态码, 信息
入库	/adminAssetsBuy/inWare	post	仓库名称, 入库数量	状态码, 信息
获取仓库列表	/adminAssetWare/getAll	get	无	状态码, 信息, 仓库列表
获取供应商列表	/adminAssetSupplier/getAll	get	无	状态码, 信息, 供应商列表
获取已提交的采购单	/adminAssetsBuy/getByPage2	get	无	状态码, 信息, 采购单列表
审核	/adminAssetsBuy/auditData	post	无	状态码, 信息, 采购单列表

4.5.7 元器件领用管理功能的接口设计

(1) 元器件领用功能设计为“领用申请”、“我的申请”、“领用审核”、“元器件领用明细”四个页面。“领用申请”页面展示所有元器件信息，包括基本档案信息，如类别、型号、规格、所存放的仓库、领用情况等。用户可以使用元器件名称或型号检索所需元器件，点击申请领用按钮可提交申请。“我的申请”页面展示当前登录用户的所有申请进度，包括申请状态、时间等。“领用审核”页面展示所有未审核的申请，可导出为 Excel 表格。“元器件领用明细”页面展示所有元器件状态、去向。可按名称、采购人、领用人、所存放仓库、采购日期检索。元器件若损坏可将其销毁。

(2) 元器件领用功能的接口如表 4-22 所示。

表 4-22 元器件领用功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
申请领用	/adminAssets/apply	post	申请的元器件 id	状态码，信息，申请人，申请时间
获取当前用户申领记录	/adminAssets/getByMyApplyPage	get	当前登录用户 id	状态码，信息，申领单，审核状态
获取所有未审核的申请	/adminAssets/getByNotAuditPage	get	审核状态	状态码，信息，申领单
审核通过	/adminAssets/applySuccess	post	无	状态码，信息
申请驳回	/adminAssets/applyFail	post	无	状态码，信息

4.5.8 日志管理功能的接口设计

(1) 日志管理功能页面展示了操作人、操作时间、操作类型等信息。可使用 IP 地址、用户名、起始时间等检索日志。

(2) 日志管理功能的接口如表 4-23 所示。

表 4-23 日志管理功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
获取日志信息	/log/getAllByPage	get	无	状态码, 信息, 日志列表

4.5.9 个人门户管理功能的接口设计

(1) 个人门户页面可编辑个人信息, 包括性别、住址、头像等个人信息, 可设置主页常用模块。可修改密码, 需输入原密码验证, 并输入两遍新密码进行确认, 修改密码成功后跳转登陆页面, 提示需重新登录。

(2) 个人门户管理功能的接口如表 4-24 所示。

表 4-24 个人门户管理功能的接口

接口功能	接口路径	请求类型	请求参数	返回数据
获取个人常用模块	/myDoor/getMyDoorList	get	无	状态码, 信息, 常用模块
修改个人常用模块	/myDoor/setMyDoorList	post	常用模块名称	状态码, 信息
修改个人信息	/user/edit	post	登陆账号, 姓名, 性别, 头像, 地址, 生日等	状态码, 信息
修改密码	/user/modifyPass	post	用户 id, 原密码, 修改后密码	状态码, 信息

4.6 其他功能设计

(1) Excel 导出功能

在领用记录、元器件档案等页面均设有此功能, 该功能实现是基于 Vue 框架, 使用 Vue 的表格组件提供的 exportCsv 方法实现, 点击“导出”按钮时, 系统会生成一个名为“导出结果.csv”的 CSV 格式文件。关键代码如图 4-16 所示。

```
excelData() {
  this.$refs.table.exportCsv( params: {
    filename: "导出结果",
  });
},
```

图 4-16 Excel 导出功能的关键代码

(2) 打印功能

打印功能可实时打印当前网页信息，关键代码如图 4-17 所示。

```
printAll(){  
  window.print();  
},
```

图 4-17 打印功能代码

(3) 锁屏功能

通过动态改变页面元素的样式和位置，结合 JavaScript 计时器和 Cookie 存储，实现了 UI 的锁屏切换，并保留了用户导航历史以便后续恢复。关键代码如图 4-18 所示。

```
lockScreen() {  
  let lockScreenBack = document.getElementById( elementId: "lock_screen_back");  
  lockScreenBack.style.transition = "all 3s";  
  lockScreenBack.style.zIndex = 10000;  
  lockScreenBack.style.boxShadow =  
    "0 0 0 " + this.lockScreenSize + "px #667aa6 inset";  
  this.showUnlock = true;  
  Cookies.set("last_page_name", this.$route.name);  
  setTimeout( callback: () => {  
    lockScreenBack.style.transition = "all 0s";  
    this.$router.push({  
      name: "locking"  
    });  
  }, ms: 800);  
  Cookies.set("locking", "1");  
}
```

图 4-18 锁屏功能关键代码

5 电子元器件管理系统实现与测试

5.1 平台部署

本节将依次从数据库、后端、前端的部署展开，展示如何启动电子元器件管理系统。

(1) 下载数据库 MySQL8.0，数据库可视化工具 HeidiSQL。打开 HeidiSQL，输入用户名、密码、端口与 MySQL 配置一致。如图 5-1 所示。

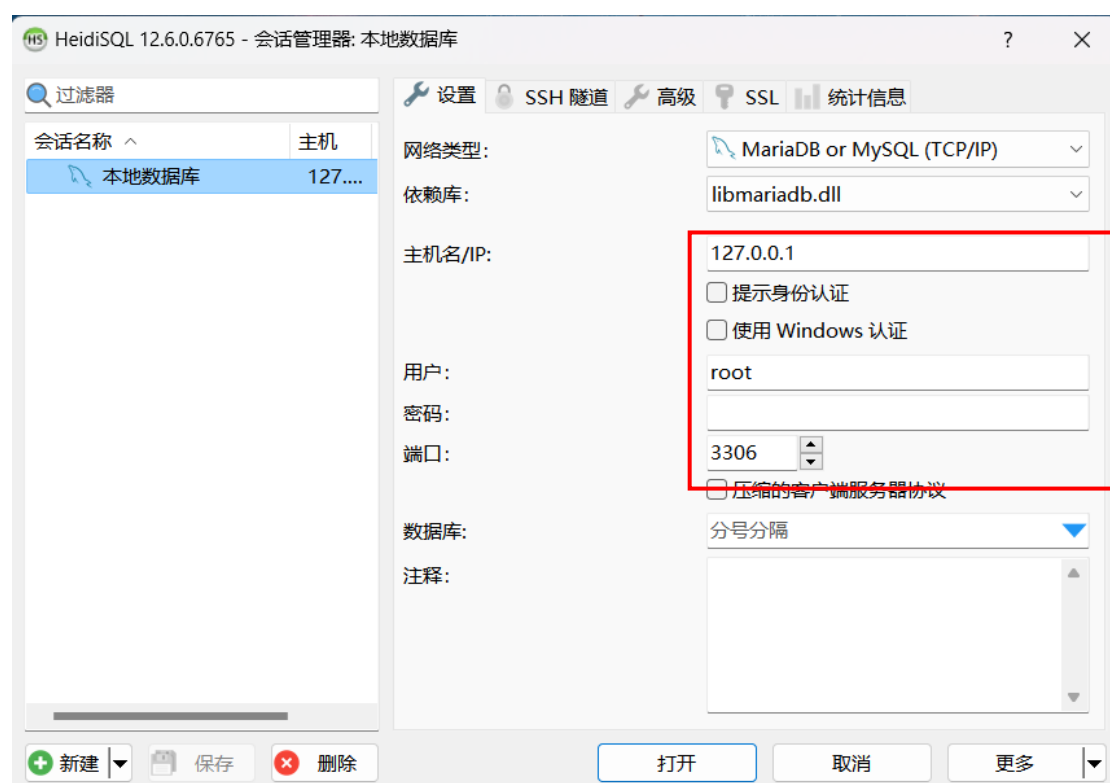


图 5-1 启动 HeidiSQL

成功启动 HeidiSQL，在文件选项中点击“运行 SQL 文件”，选择项目文件中的数据库文件“ele.sql”，如图 5-2 所示。运行完成，成功配置数据库。

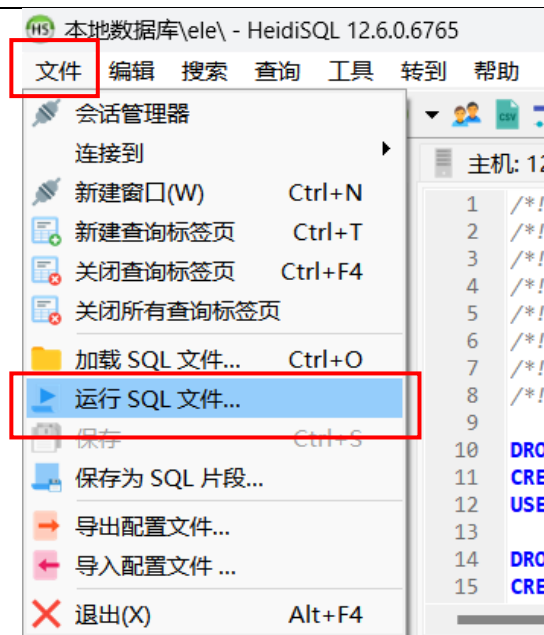


图 5-2 运行 SQL 文件

(2) 下载用于缓存的数据库 Redis5.0.14, JDK17, 开发工具 IDEA2022.1.4, 依赖管理工具 apache-maven-3.8.4。使用 IDEA 打开项目文件中的 back 文件夹, 如图 5-3 所示。

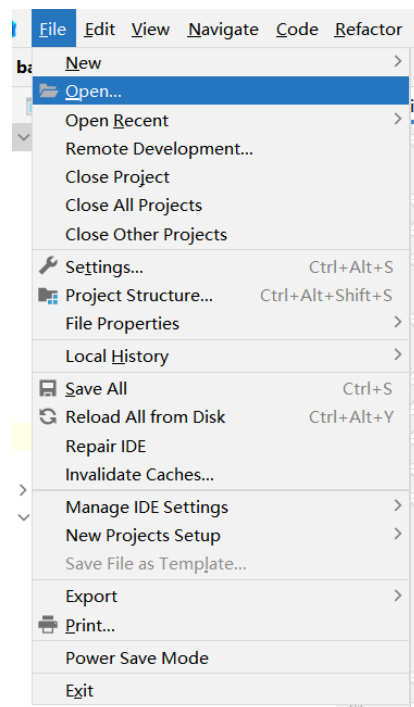


图 5-3 IDEA 打开后端文件夹

打开设置, 如图 5-4 所示。

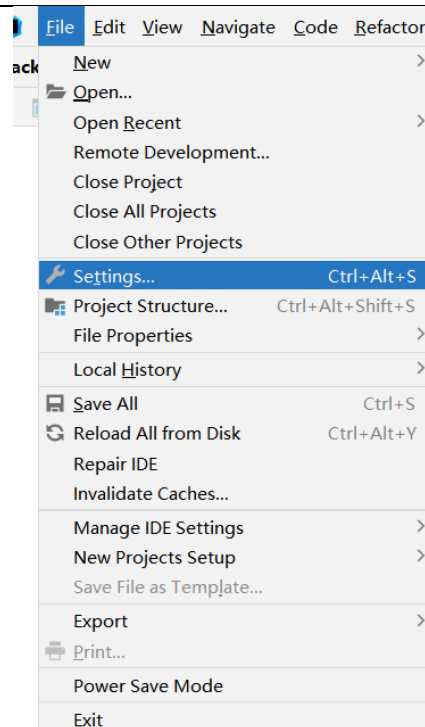


图 5-4 打开 IDEA 设置

根据 maven 下载路径完成配置，如图 5-5 所示。

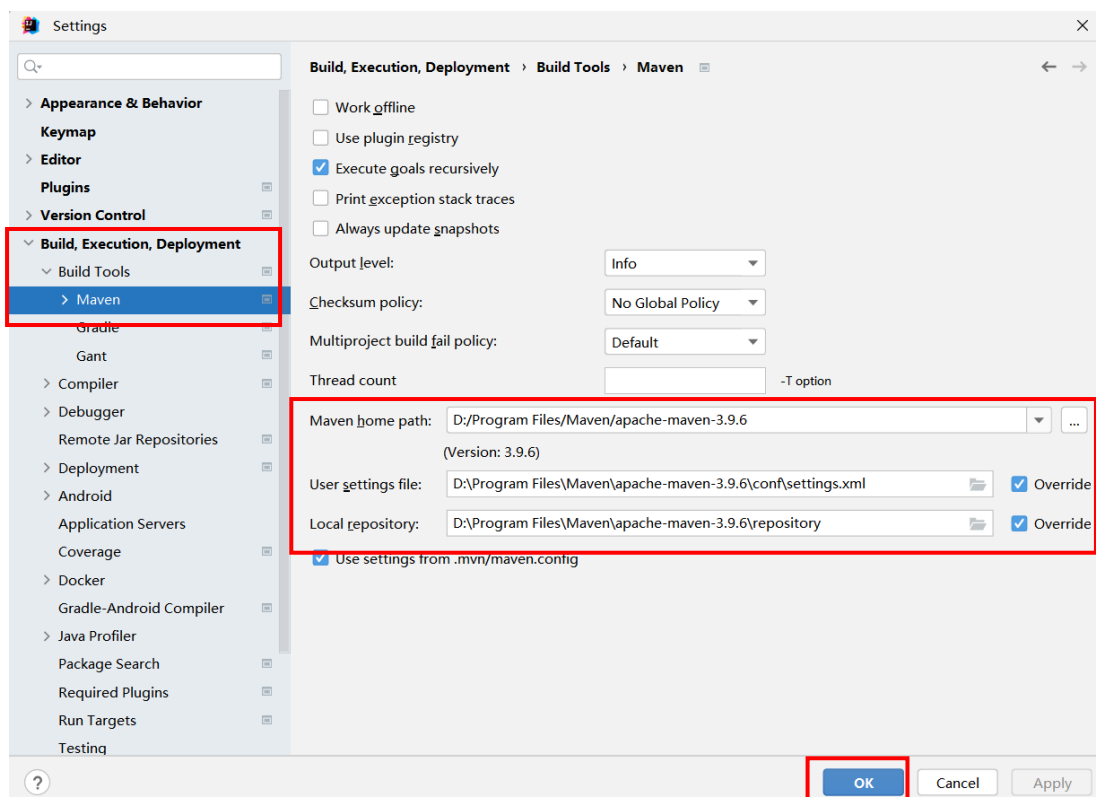


图 5-5 配置 maven 路径

在项目的 src 路径下打开 application.yml，根据数据库文件“ele.sql”的路径配置 url，根据 MySQL 数据库密码配置密码，如图 5-6 所示。



图 5-6 后端与数据库连接的配置

(3) 安装 node.js 20.11.0，安装成功后使用 cmd 输入“npm i -g @vue/cli”安装 vue 脚手架，安装 Visual Studio2022。使用 Visual Studio2022 打开项目的 front 文件夹，在 Terminal 输入“npm i”拉取前端依赖。安装完成后在 Terminal 输入“npm run dev”即可成功启动前端程序。

打开浏览器，访问“http://localhost:8080/”，成功进入系统，如图 5-7 所示。

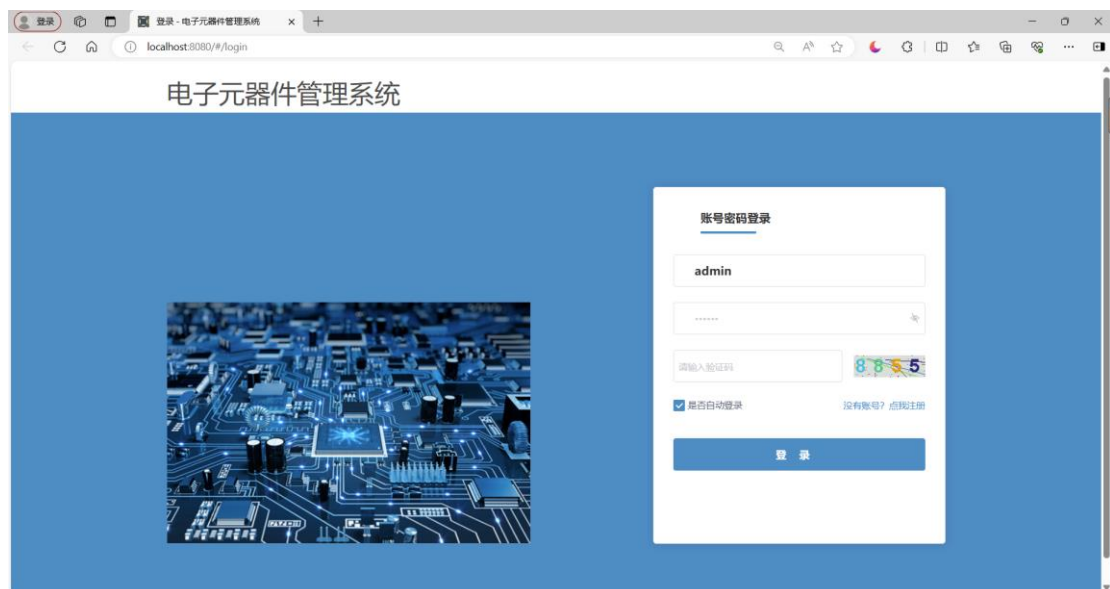


图 5-7 系统初始页面

5.2 系统页面展示

前后端程序成功运行后，在浏览器中访问端口“http://localhost:8080/”，运行结果如图 5-8 所示。进入系统登陆页面，点击验证码图片可更新验证码。并且验证码有时效性。

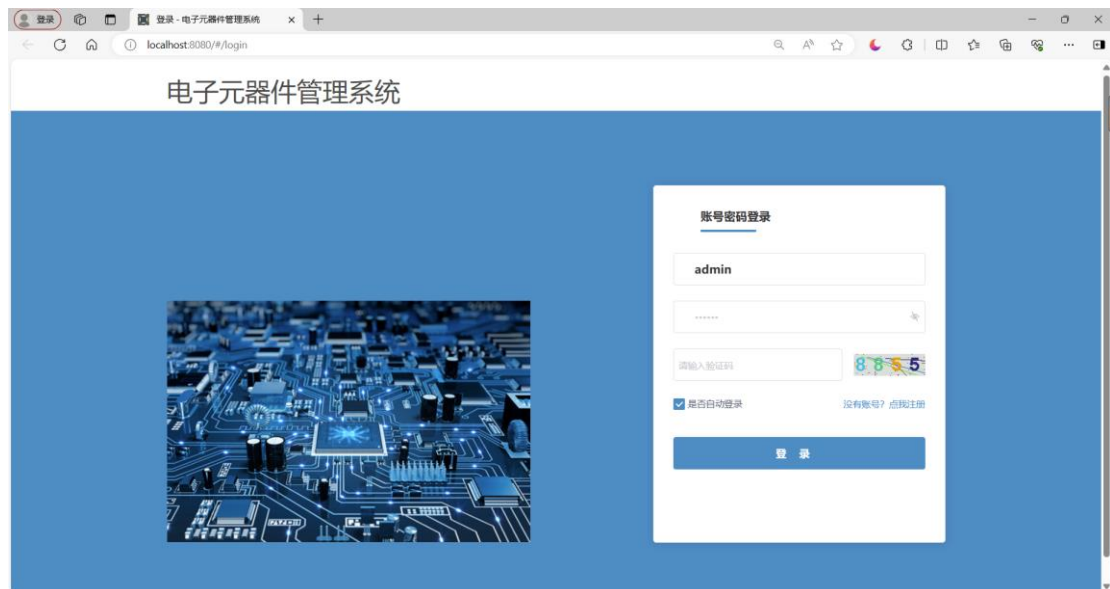


图 5-8 登录页面

没有账号可以点击注册按钮，跳转注册页面，如图 5-9 所示。

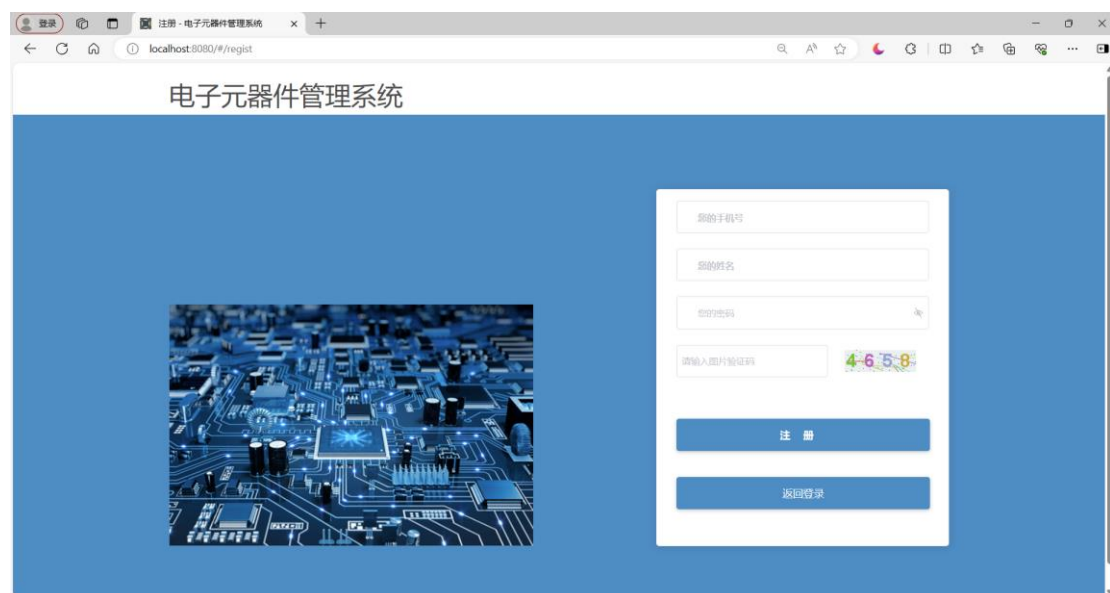


图 5-9 注册页面

登陆成功，进入系统主页面，如图 5-10 所示，可设置常用模块，点击常用模块可直接跳转该页面。

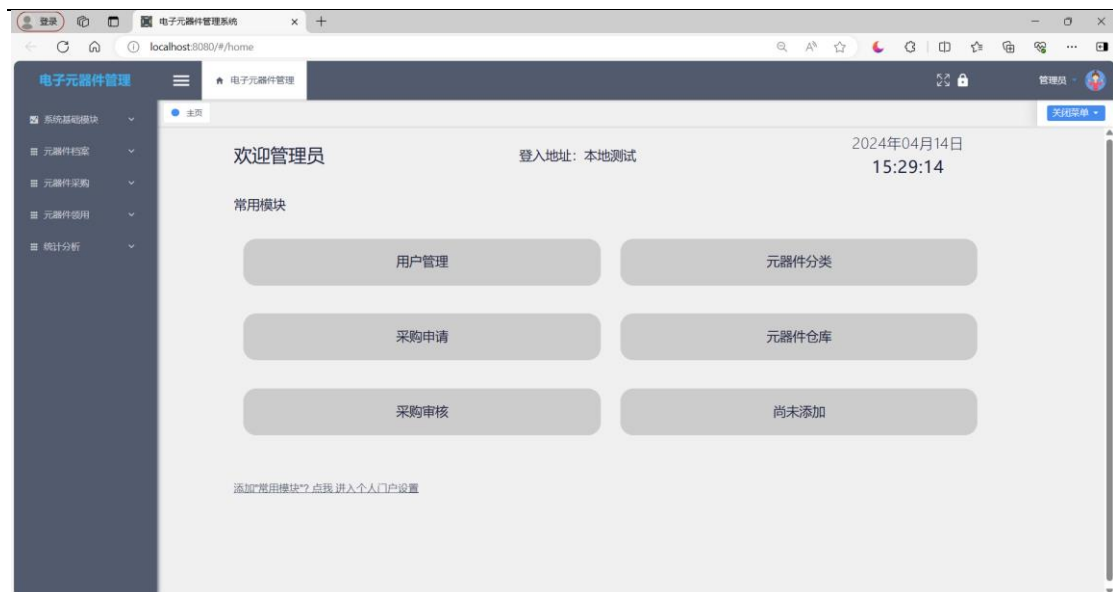


图 5-10 主页

用户可在个人门户自定义常用模板，在用户中心页面除了添加常用模块，用户还可编辑个人信息，包括头像、家庭住址、个性签名等。个人中心页面如图 5-11 所示。

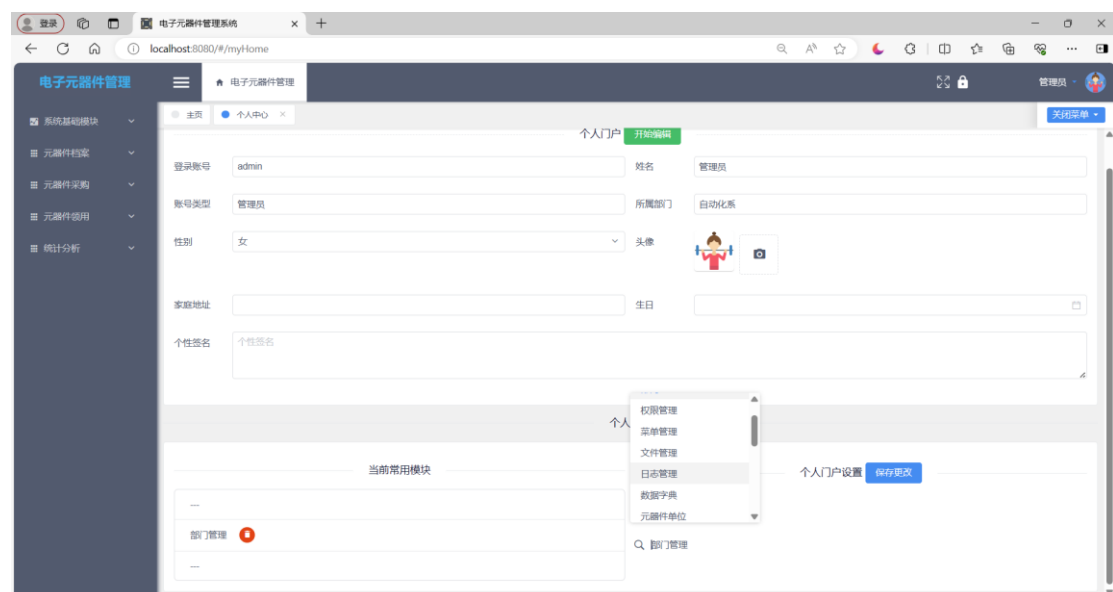


图 5-11 个人中心

用户管理界面可实现对已有用户的管理。包括搜索用户、编辑用户信息、重置密码、禁用、启用、删除用户。还可实现 Excel 批量导出用户信息和直接打印网页信息。用户管理页面如图 5-12 所示。

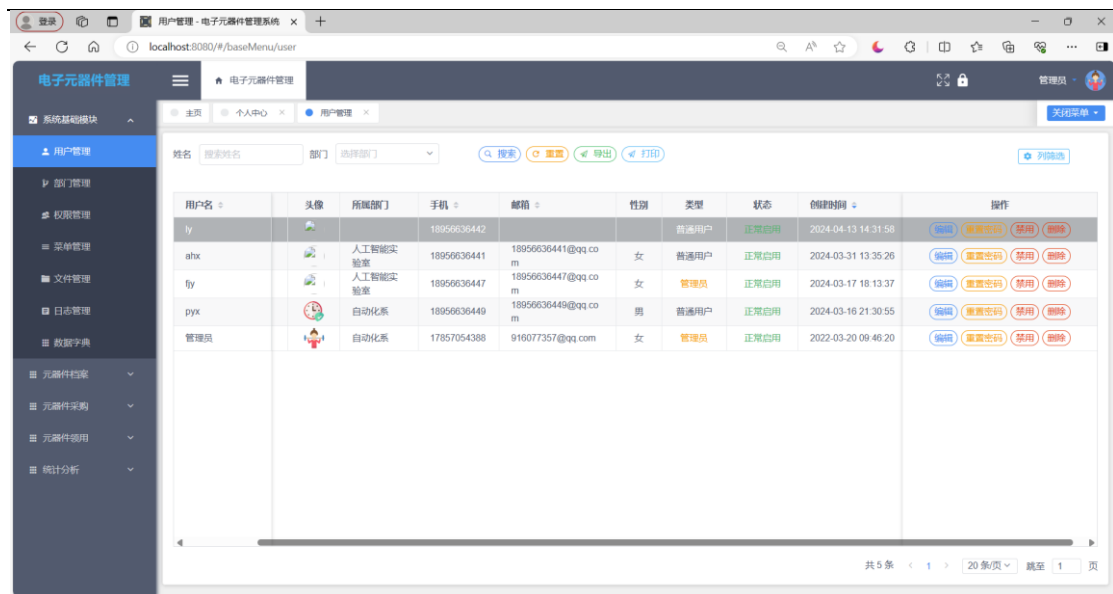


图 5-12 用户管理

编辑用户的重要功能之一是通过给用户分配角色，使用户获得该角色相应权限。编辑用户页面如图 5-13 所示。

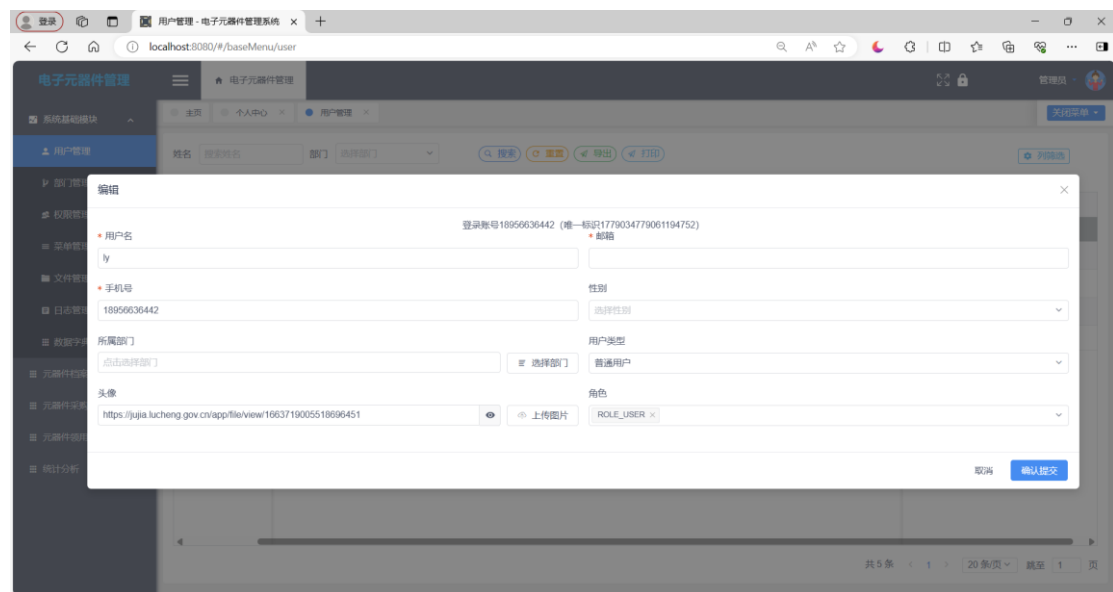


图 5-13 编辑用户信息

部门管理界面可实现添加、删除部门、按部门查找用户和导出用户 Excel 表格功能。部门管理页面如图 5-14 所示。

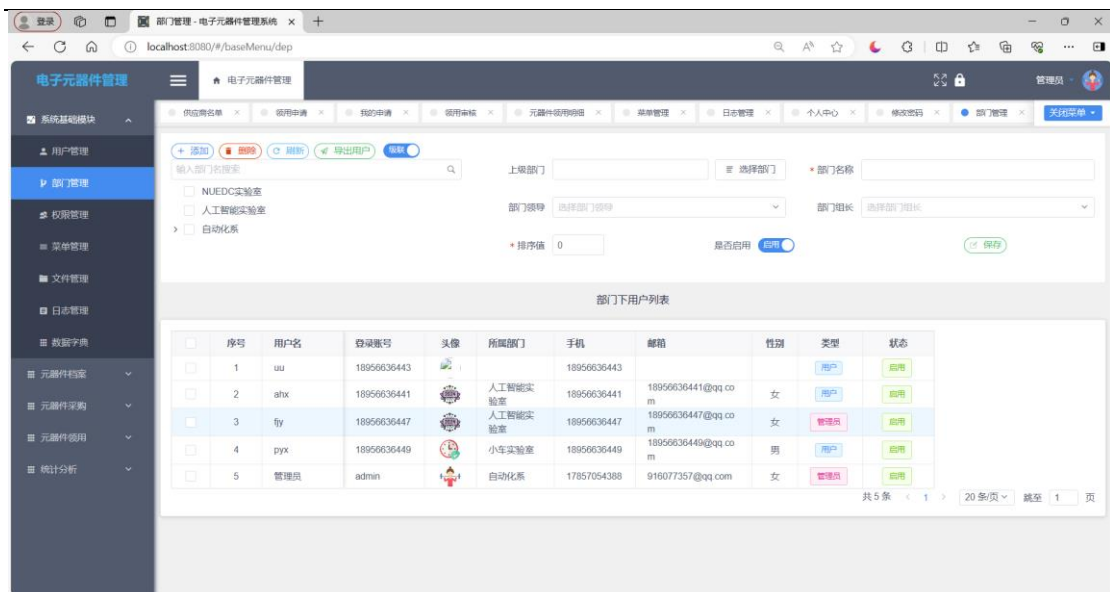


图 5-14 部门管理

权限管理页面可以新增、编辑、删除角色，例如可设定超级管理员、普通管理员、仓管员、普通用户等。还可给角色分配权限，设置默认角色。权限管理页面如图 5-15 所示。

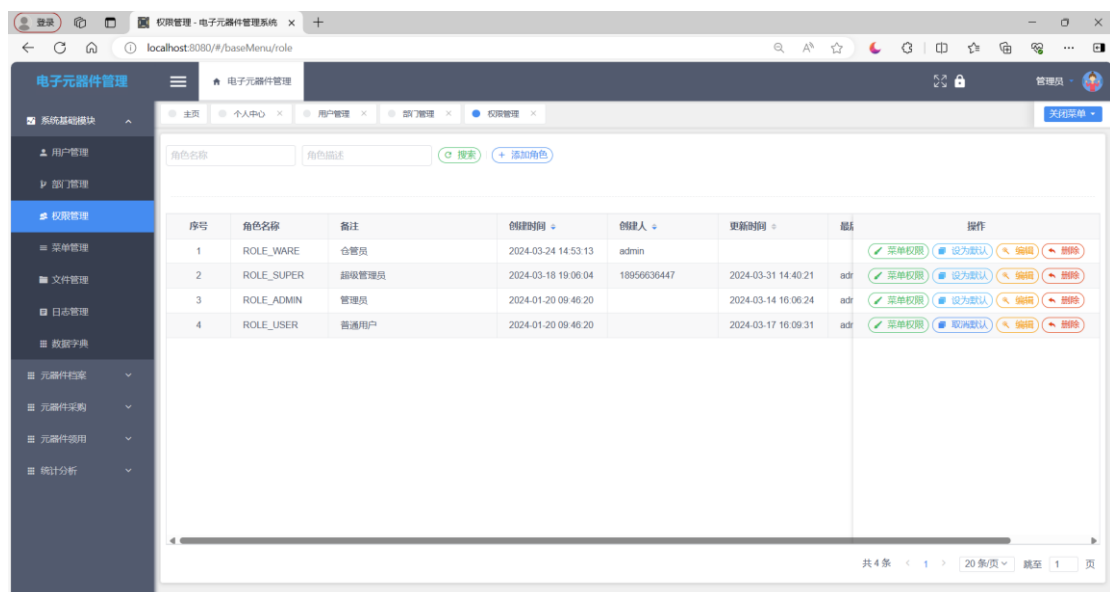


图 5-15 权限管理

点击“菜单权限”按钮，给角色分配权限的页面如图 5-16 所示。

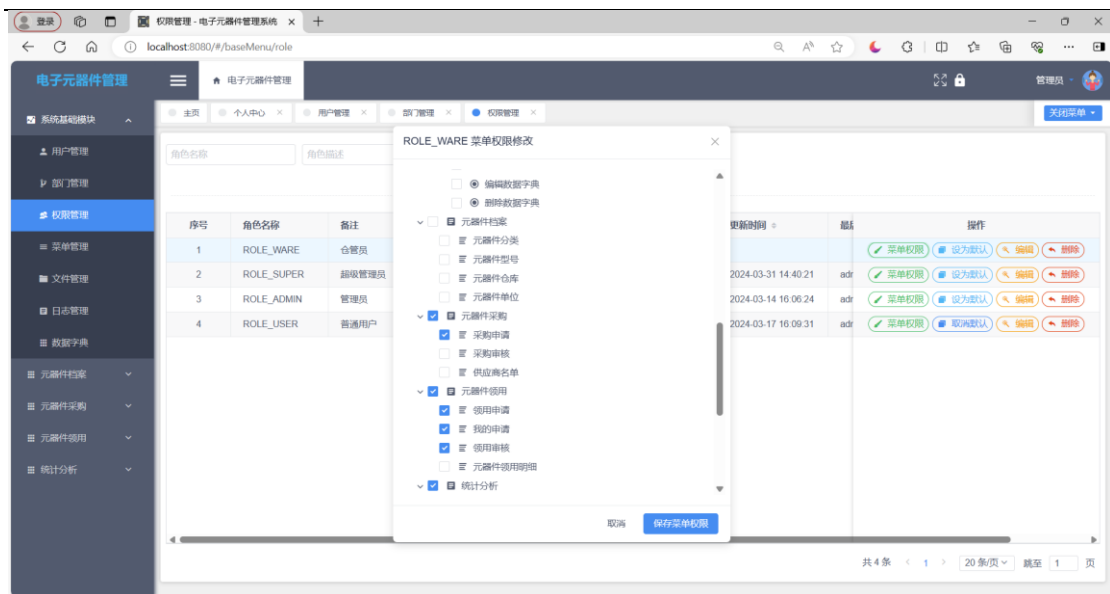


图 5-16 分配角色权限

元器件采取四级分类，实现大类->分类->名称->具体型号的分类。通过数据字典维护元器件大类，如图 5-17 所示。

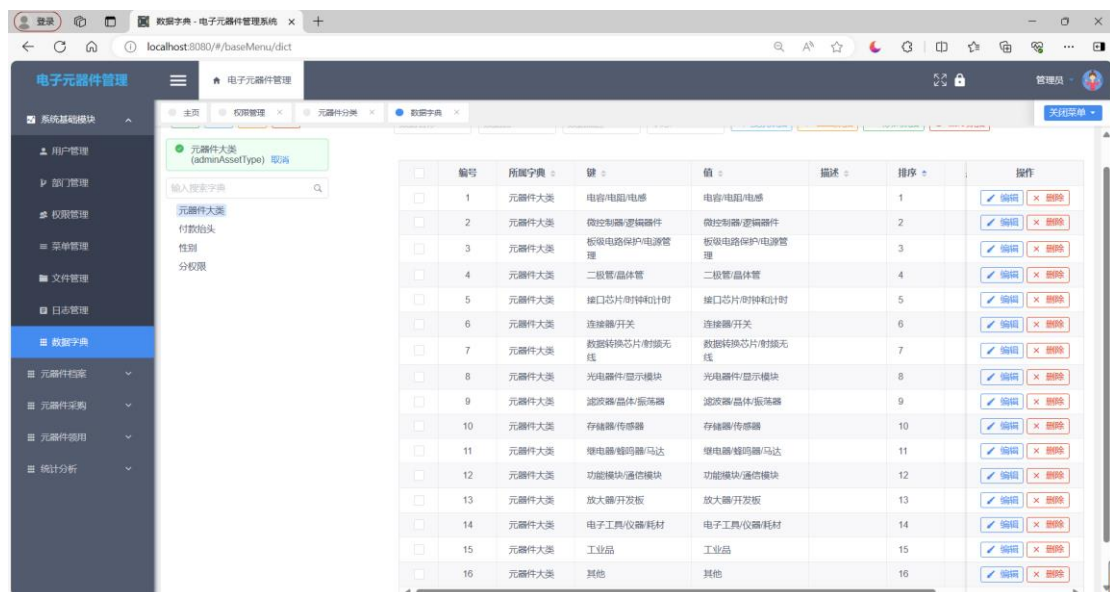


图 5-17 数据字典

数据字典页面可以管理字典类型和数据，如要添加元器件大类的数据，先选中“元器件大类”字典，再点击“添加数据”按钮，出现添加字典数据窗口，如图 5-18 所示。

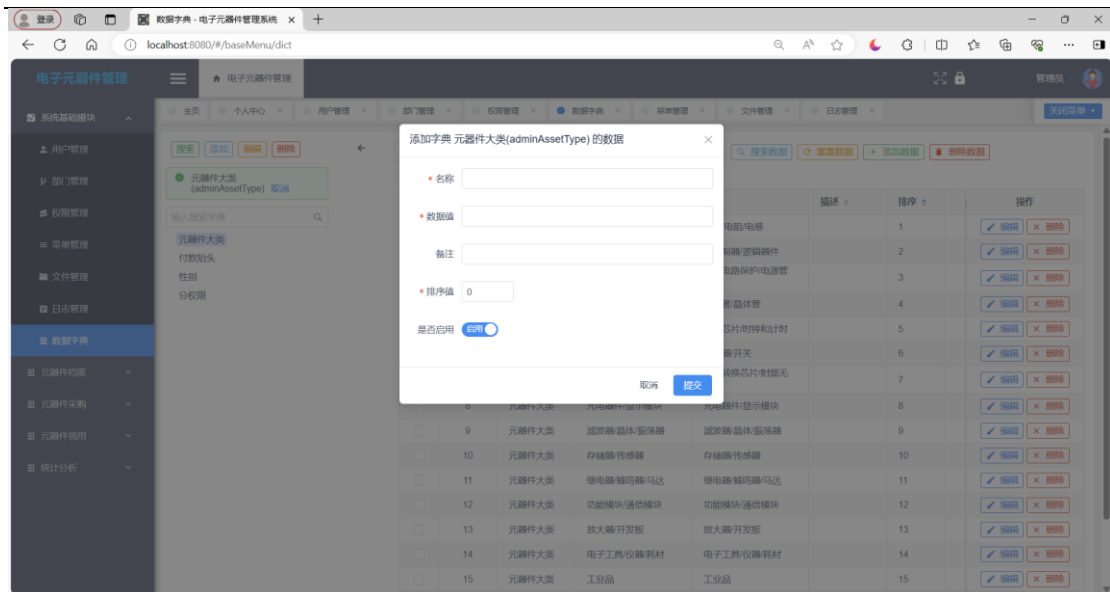


图 5-18 数据字典添加数据

在元器件分类页面管理具体分类，可实现 Excel 导出功能。如图 5-19 所示。

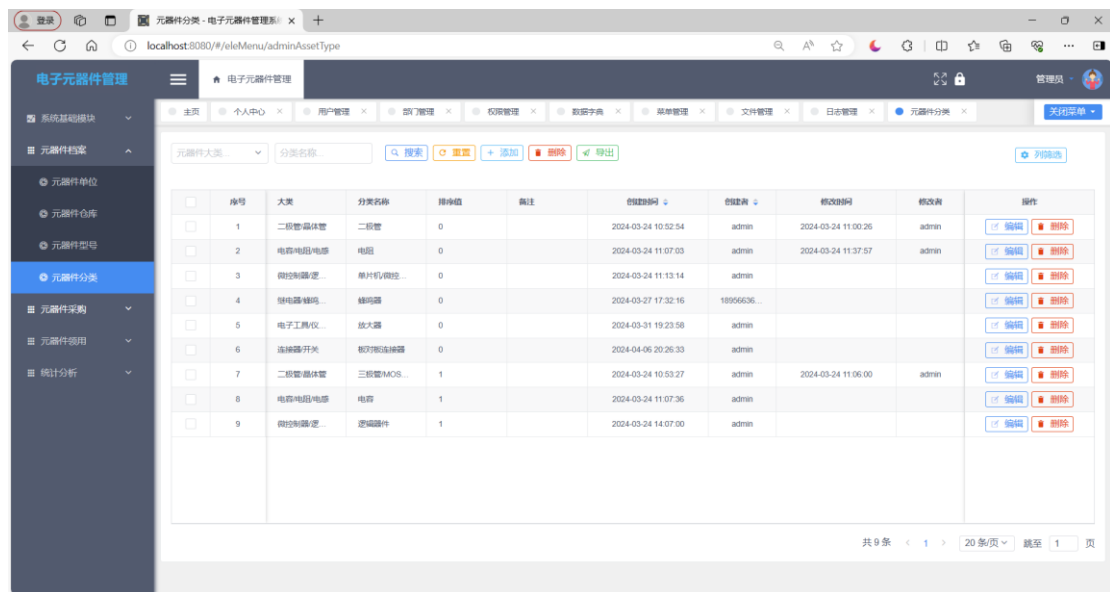


图 5-19 元器件分类

点击新增或编辑按钮，弹出窗口如图 5-20 所示。

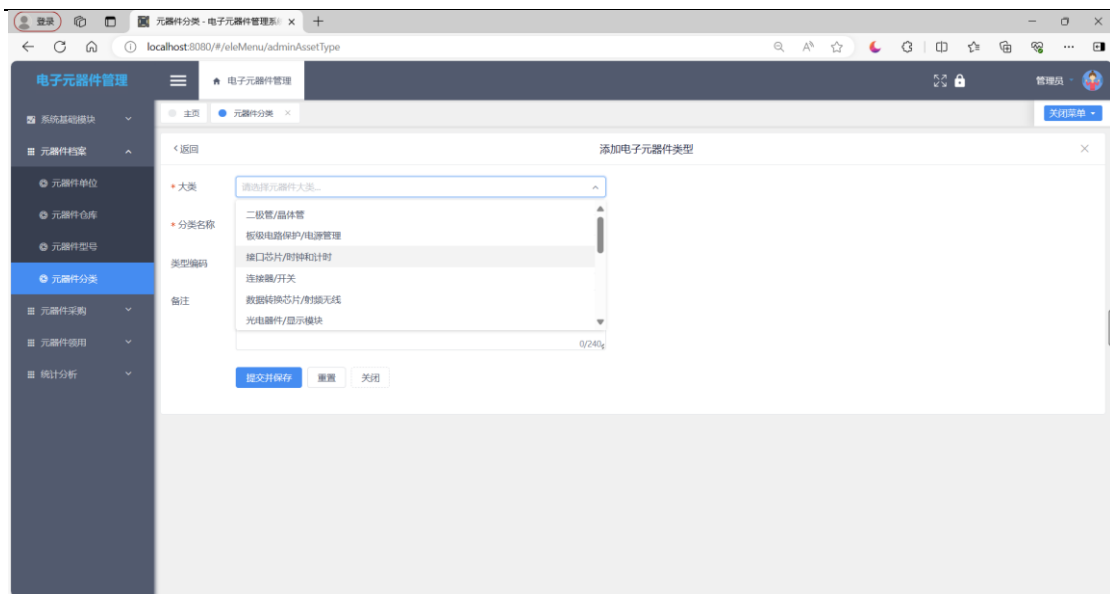


图 5-20 添加元器件分类

元器件型号界面展示具体名称及型号，如图 5-21 所示。可实现 Excel 导出。

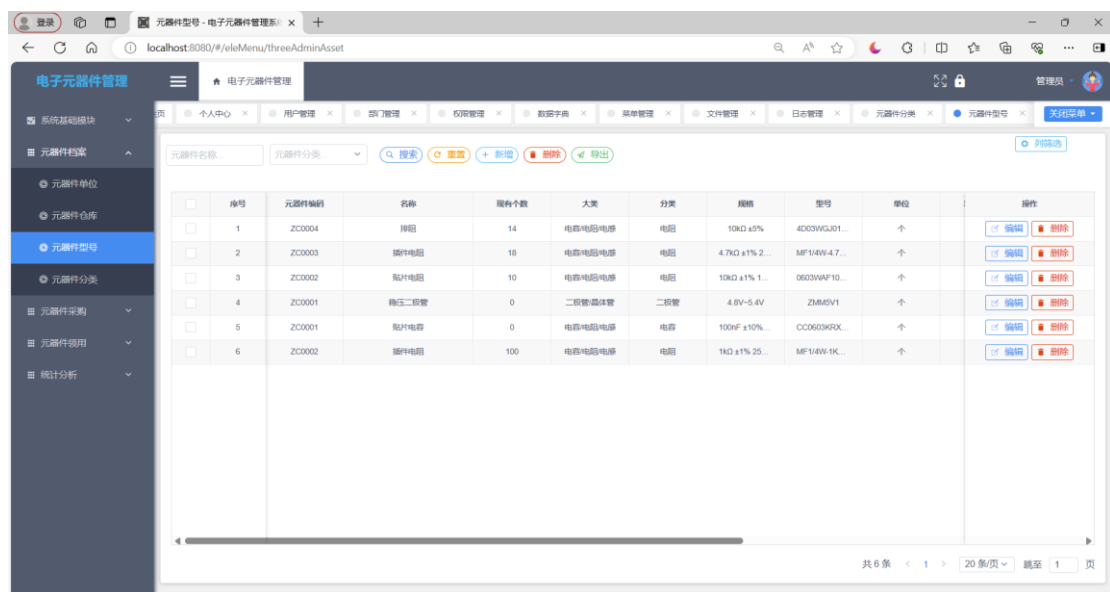


图 5-21 元器件型号

点击“新增”按钮添加元器件型号，在分类下拉框可以选择元器件分类中已有大类、分类级联选项，添加元器件型号的页面如图 5-22 所示。

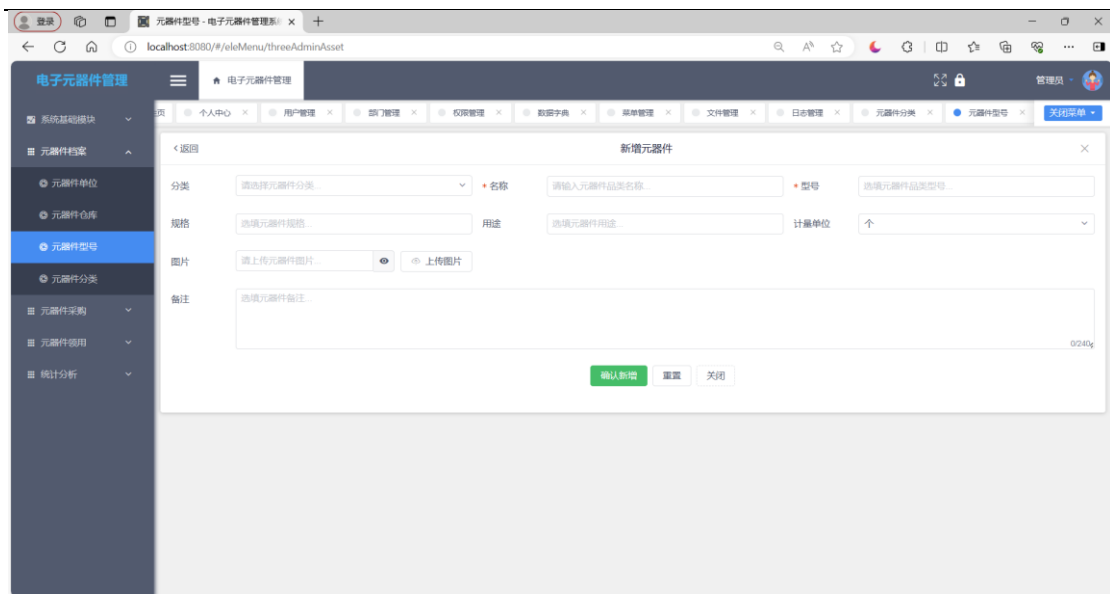


图 5-22 添加元器件型号

采购申请页面如图 5-23 所示。如没有所需元器件则需要先维护元器件档案信息。提交审批后需选择供应商。

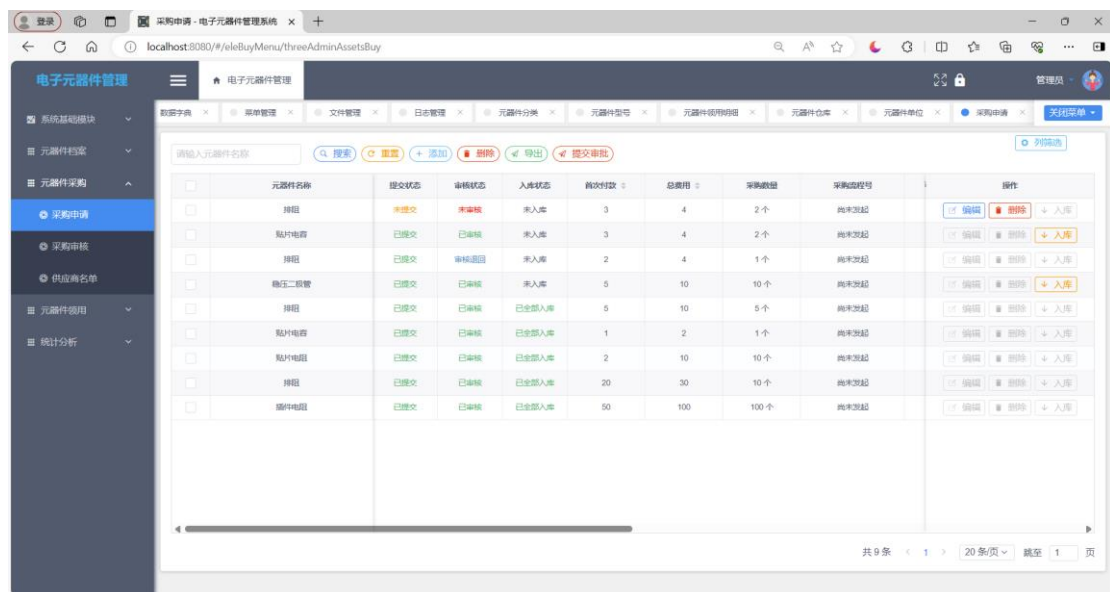


图 5-23 采购申请

点击“添加”可新增申请，如图 5-24 所示。新增采购单点击图标可从已有型号页面快速导入需要采购的元器件，输入采购数量、金额即可完成采购单的填写。

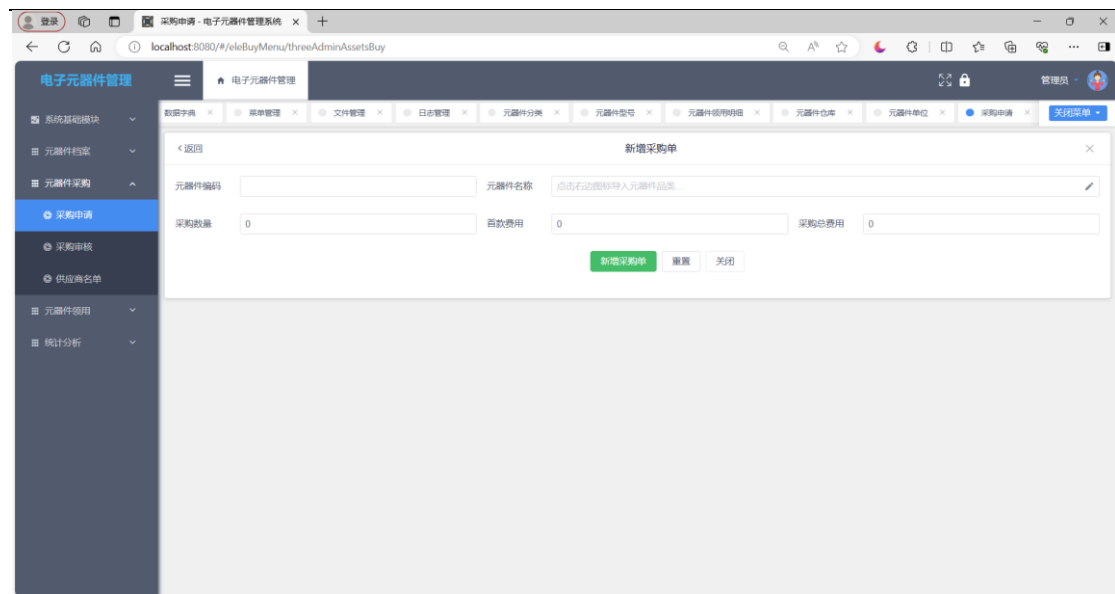


图 5-24 新增采购单

导入所需型号的页面如图 5-25 所示。

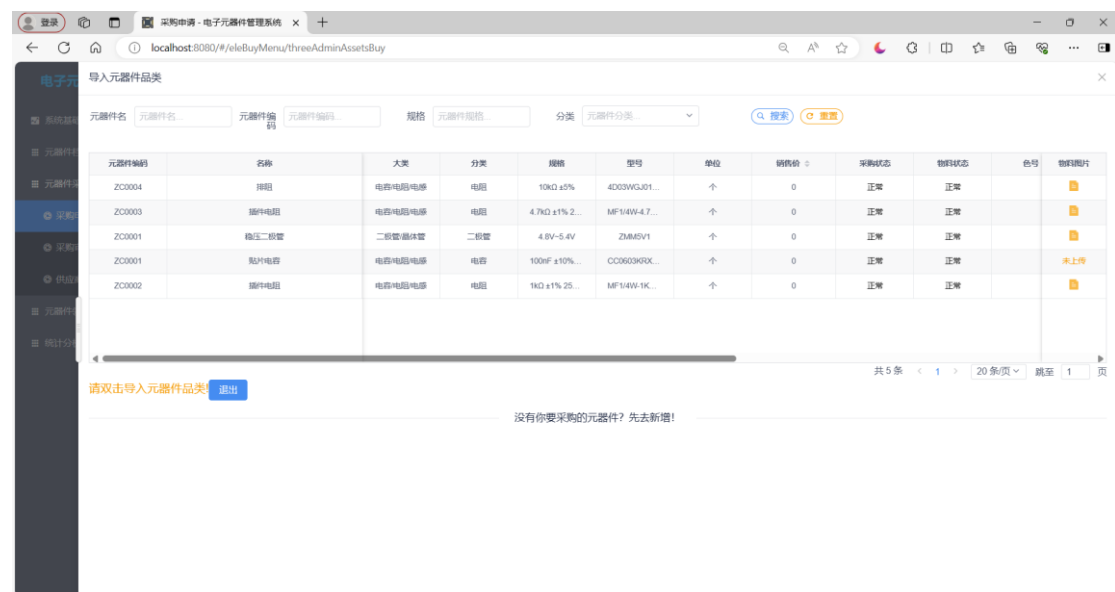


图 5-25 导入元器件型号

采购审核界面展示待审核的采购申请，拥有该界面权限的用户可进行审批。页面如图 5-26 所示。

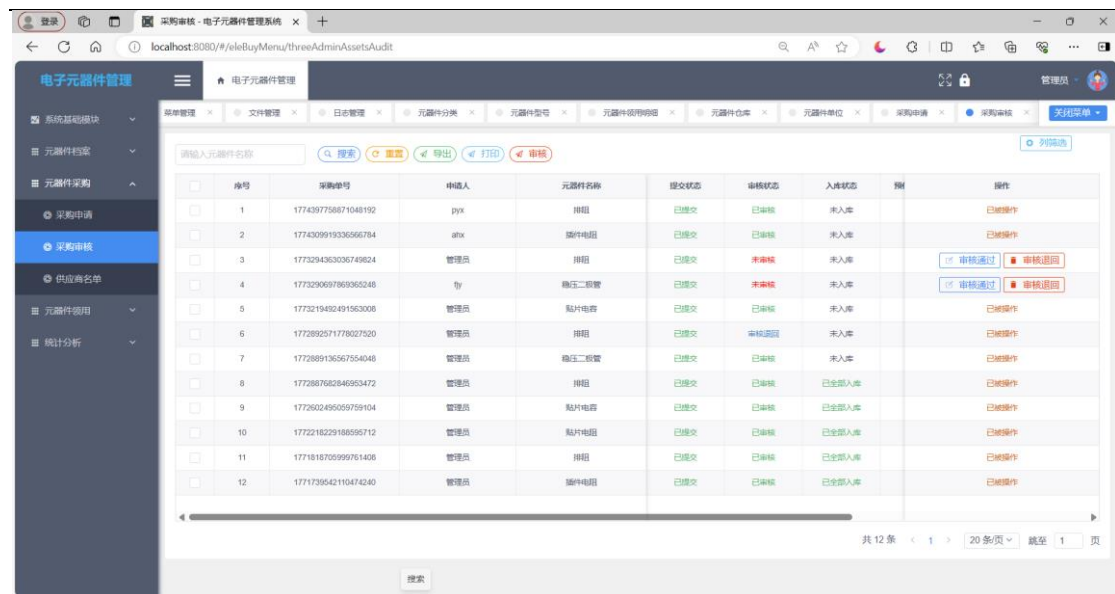


图 5-26 采购审核

供应商名单页面用于维护供应商数据，可实现添加、删除、编辑供应商信息。供应商名单页面如图 5-27 所示。

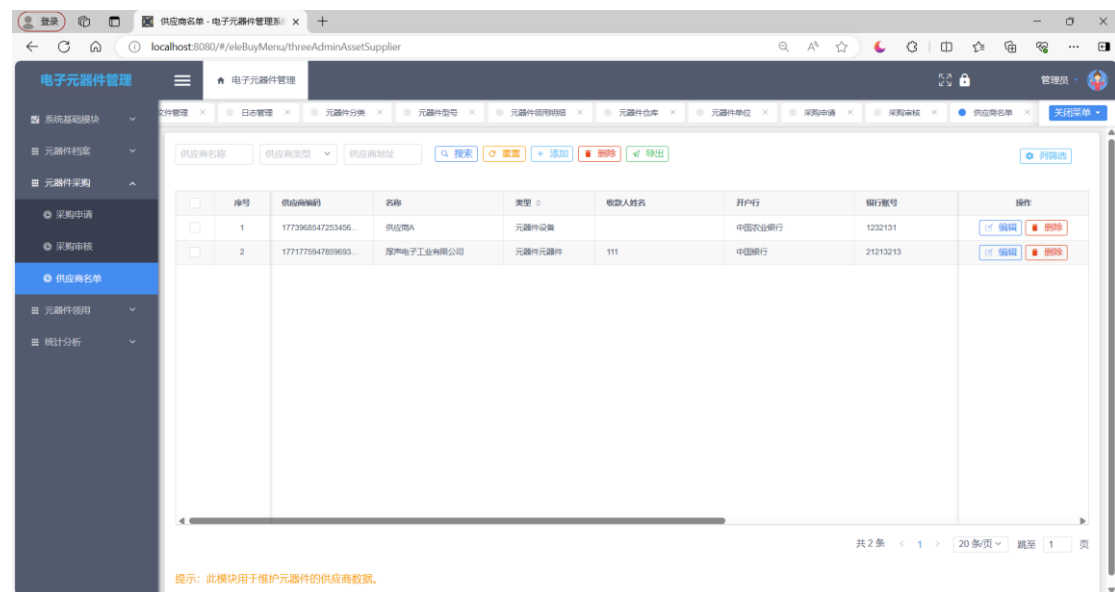


图 5-27 供应商名单

用户可在领用申请界面输入元器件名称或型号查找所需元器件，点击“申请领用”即可提交申请。领用申请页面如图 5-28 所示。

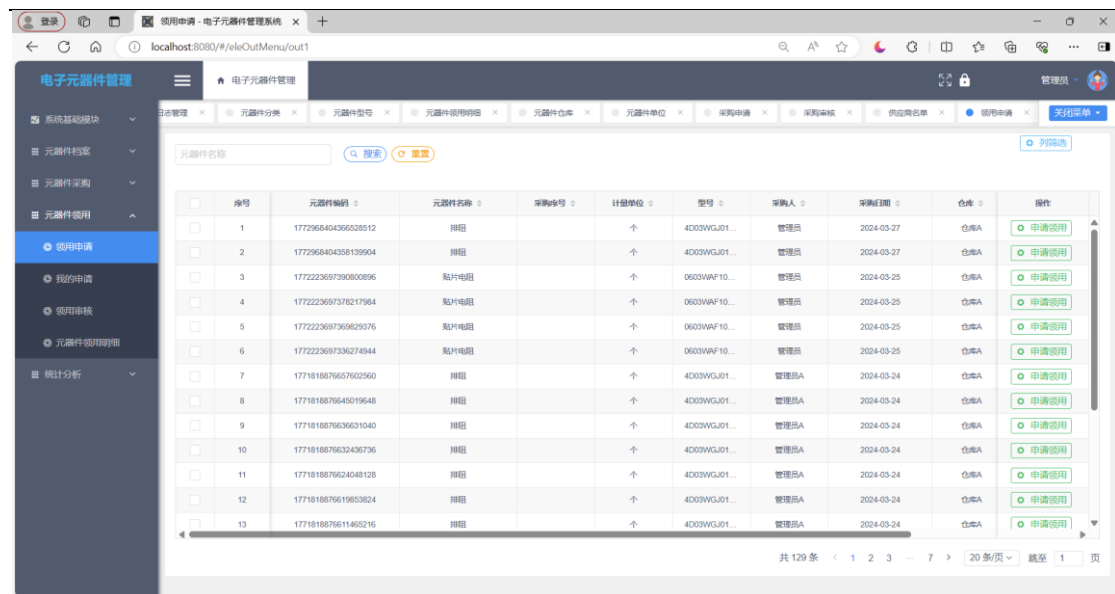


图 5-28 领用申请

用户可在我的申请页面查看领用的审核进度，根据管理员操作，申请状态会分别显示为“等待审批”、“已通过”或“申请驳回”，如图 5-29 所示。

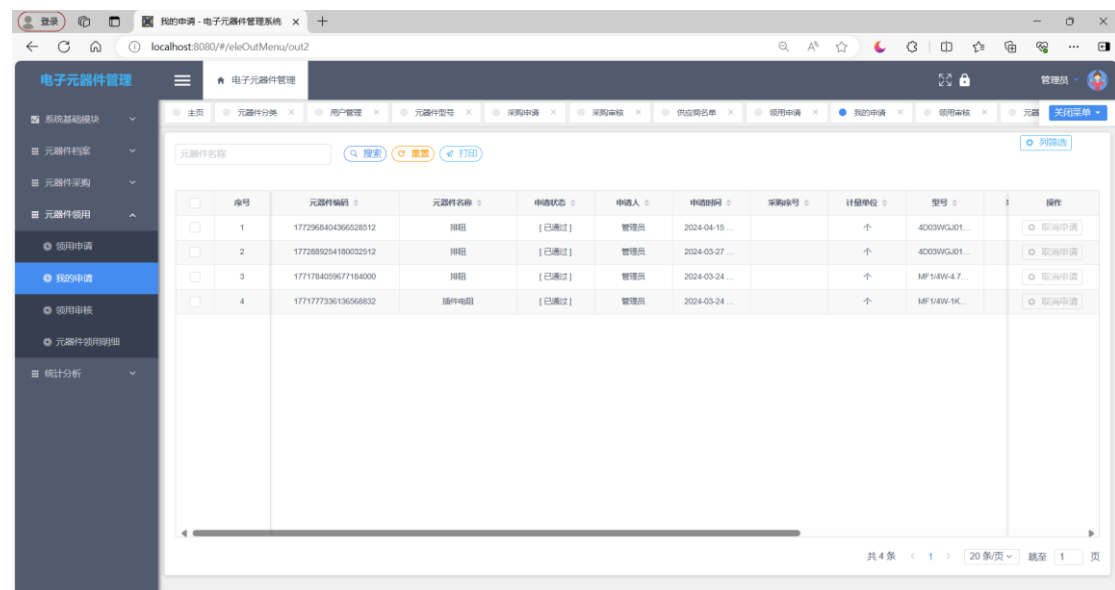


图 5-29 我的申请

图 5-30 为领用审核界面，显示待审核的领用申请，一般只对管理员开放权限。

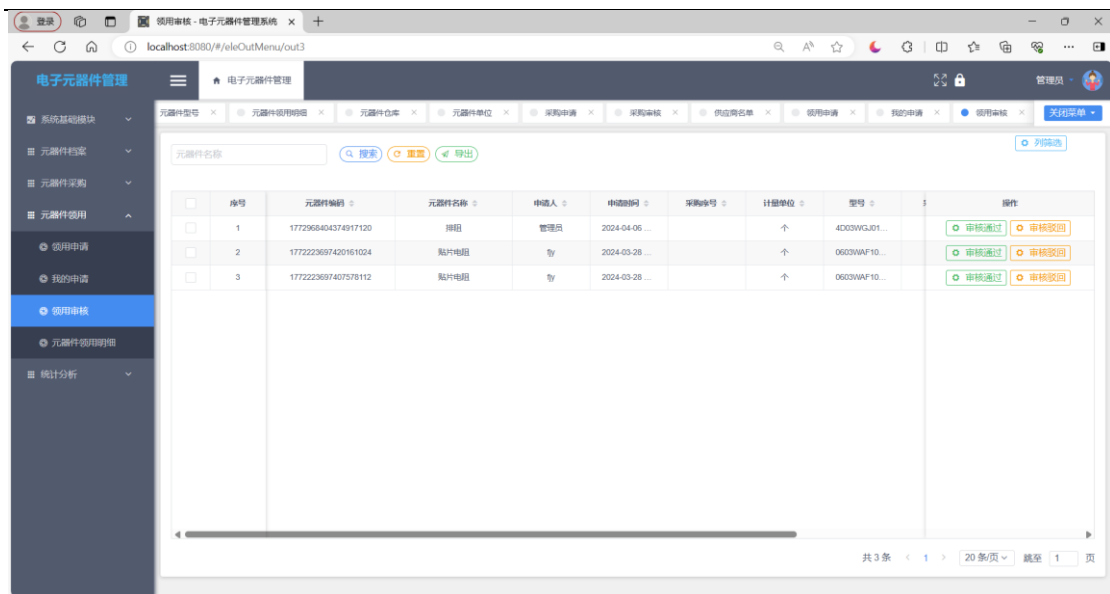


图 5-30 领用审核

图 5-31 为元器件领用明细页面，记录了所有元器件的去向，包括元器件状态，是否被申请、是否领用、或已损坏被销毁。管理员可在此页面查看所有元器件使用情况。该页面也设置了导出 Excel 表格和打印功能。

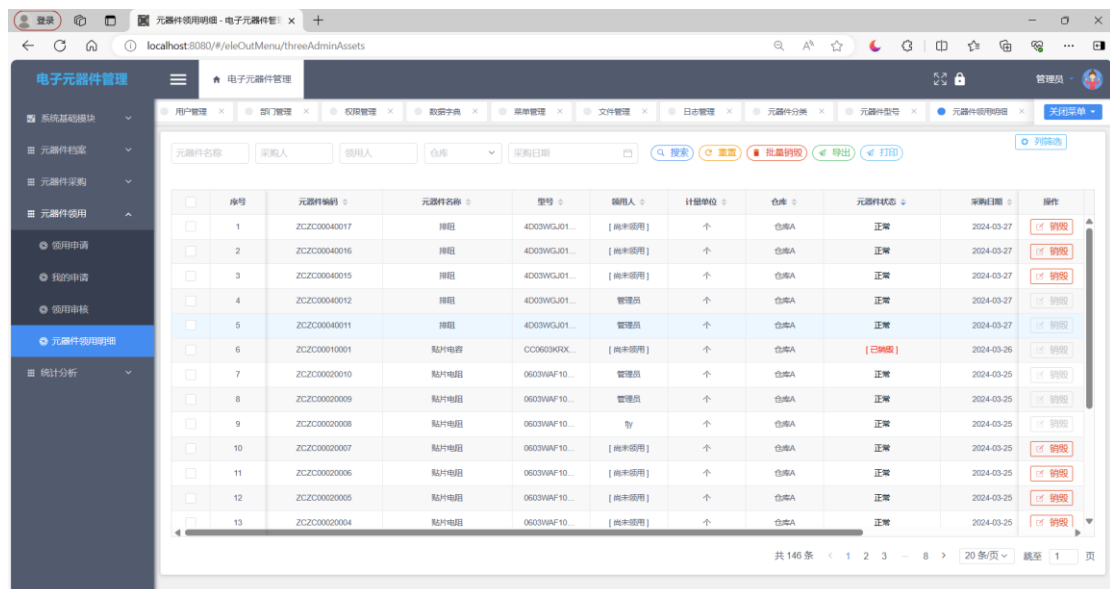


图 5-31 元器件领用明细

数据统计模块分为采购成本分析和采购型号分析。采购成本分析按照元器件二级分类统计了采购总额，便于使用者查看各类采购花费高低。例如图中可清晰地看到采购电阻支出最多，采购电容支出最少。采购型号分析页面统计了各型号采购总数，便于使用者统计哪种型号采购总数最多。

采购成本分析如图 5-32 所示。

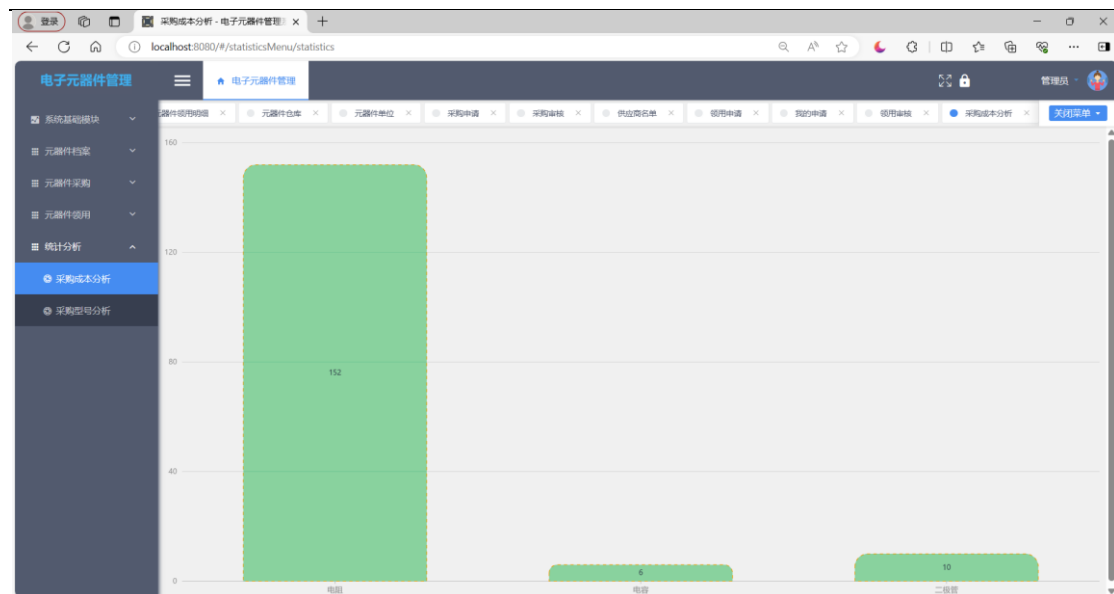


图 5-32 采购成本分析

采购型号分析如图 5-33 所示。

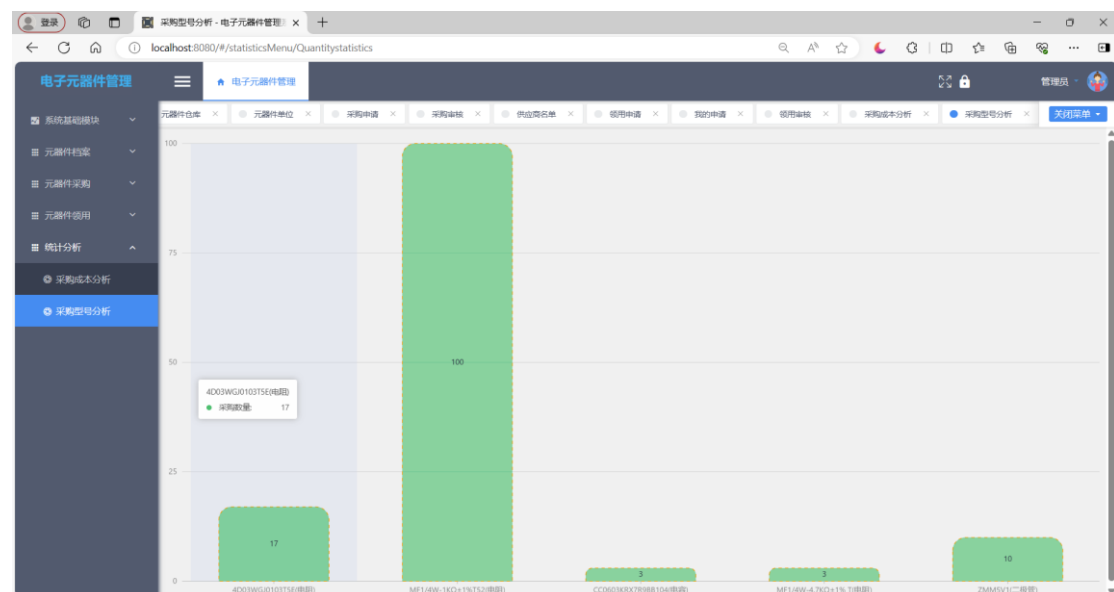


图 5-33 采购型号分析

菜单管理页面可在对应位置填写 router 路径，给系统添加新页面，便于二次开发增加新功能。菜单管理页面如图 5-34 所示。

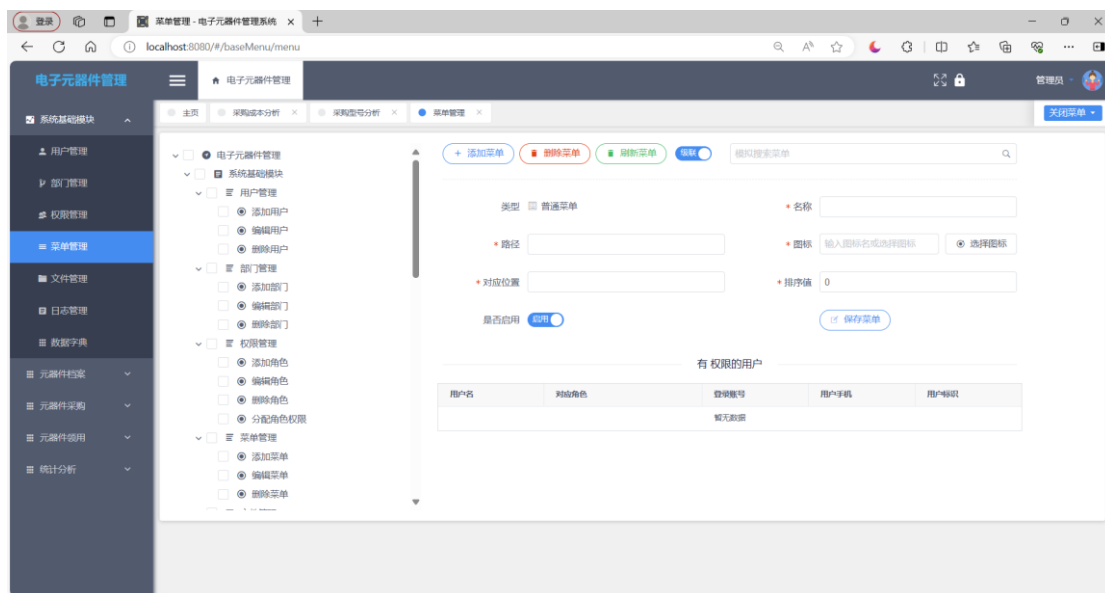


图 5-34 菜单管理

文件管理页面保存了所有上传到系统的文件，包括各用户头像、元器件图像等。文件管理页面如图 5-35 所示。

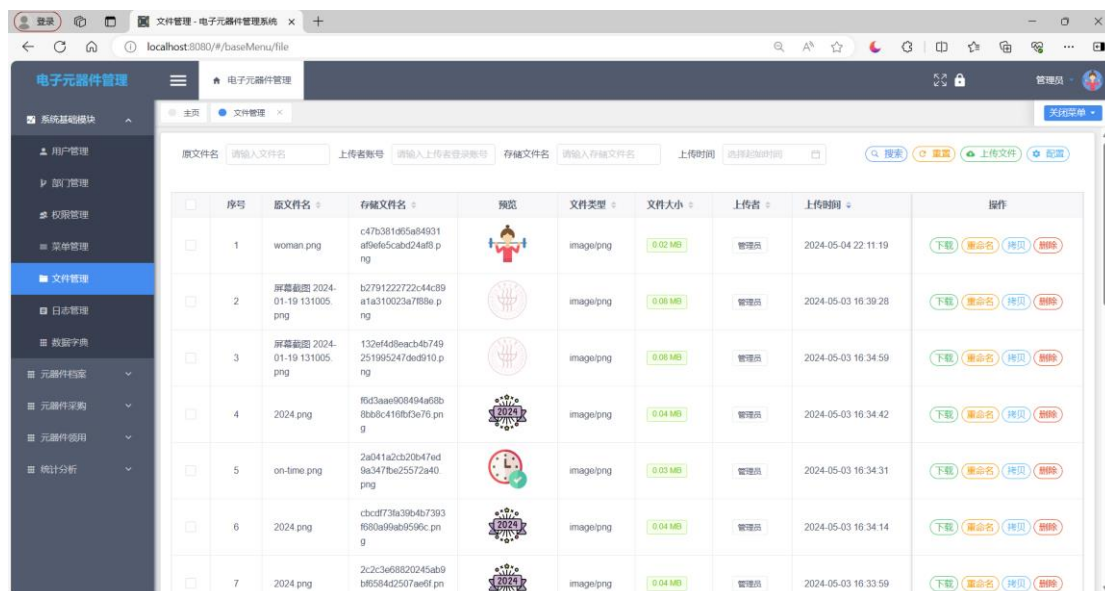


图 5-35 文件管理

日志管理页面可查看登录日志、数据中心日志等。便于追踪系统使用情况。日志管理页面如图 5-36 所示。

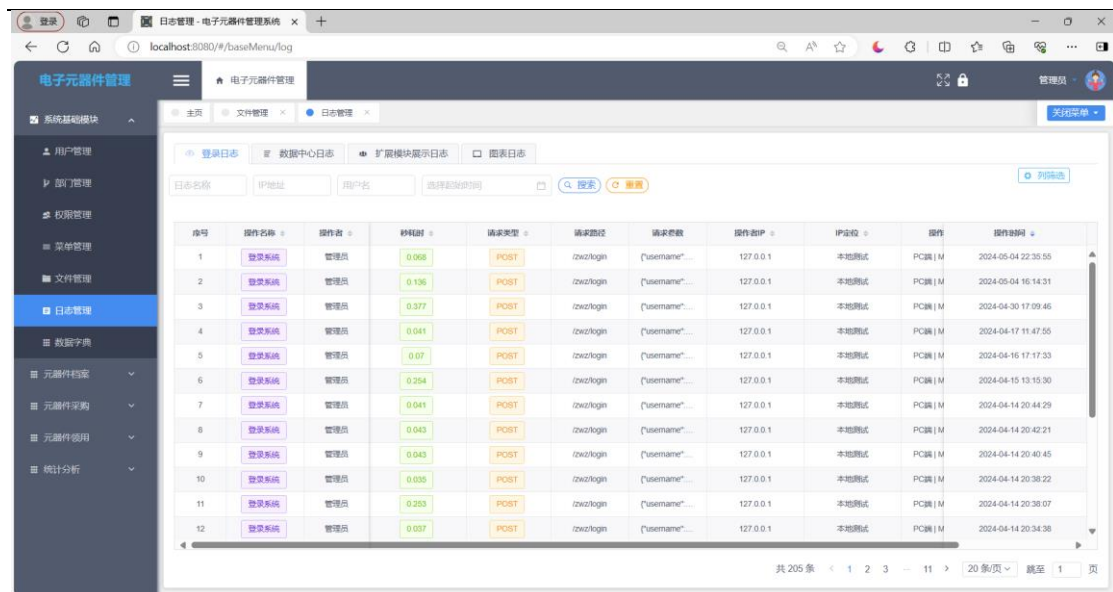


图 5-36 日志管理

5.3 系统测试

系统测试是一项关键的集成测试阶段，旨在将软件、计算机硬件、外部设备、网络等各个元素结合，进行全面的信息系统组装与验证。这一测试环节专注于整个产品系统的表现，旨在核实系统是否完全满足预设的需求规格，同时揭示任何与这些规格不符或相冲突的部分，进而提出优化和改进建议。

在系统测试过程中，一旦发现问题，将进行深入的分析与调试，以确定错误的根源和位置，并据此进行必要的修正。这一过程通常基于系统的整体需求文档，采用黑盒测试方法，确保所有关联组件都被纳入测试范围。测试记录如下。

(1) 登陆与注册功能测试如表 5-1 所示。

表 5-1 登陆与注册功能测试

序号	功能描述	测试方法	预期结果	是否通过测试
1	已存在用户登录系统	在登录页面输入已存在用户的账号和对于密码，输入图形验证码，点击登录按钮	成功进入主页	通过测试
2	新用户注册	在登陆页点击注册按钮，系统跳转注册页面。输入账号、姓名、密码、验证码。点击新增。系统跳转回登陆页面，在登录页面重新登陆	注册时若账号已存在会提醒用户。注册成功，可实现登陆功能	通过测试

(2) 权限管理功能测试如表 5-2 所示。

表 5-2 权限管理功能测试

序号	功能描述	测试方法	预期结果	是否通过测试
1	添加角色	点击“添加角色”，填写角色名称和角色描述	角色列表出现新角色	通过测试
2	编辑角色	点击“编辑”，填写角色名称和角色描述	该角色变更为填写的新角色	通过测试
3	删除角色	点击“删除”，并在弹窗选择“确定”	若该角色无用户使用，可成功删除，若该角色有用户使用，系统提示无法删除	通过测试
4	设置默认角色	点击某一角色“设为默认”	新用户注册时自动获得默认角色身份	通过测试
5	分配角色权限	点击“菜单权限”，勾选角色拥有权限菜单功能	身份为该角色的用户进入系统，菜单列只显示拥有权限的功能模块	通过测试

(3) 部门、用户管理功能测试如表 5-3 所示。

表 5-3 部门、用户管理功能测试

序号	功能描述	测试方法	预期结果	是否通过测试
1	添加部门	不选中已有部门，点击“添加”，输入名称；选中已有部门，点击“添加”，输入名称，添加子部门	成功新增部门，部门间呈级联关系	通过测试
2	编辑部门	选中部门，编辑信息，点击“保存”	部门信息成功变更	通过测试
3	删除部门	选中部门，点击“删除”按钮	部门下属无用户时，成功删除；部门下有用户存在，系统提示无法删除	通过测试
4	查看部门下属用户	选中部门	下方用户列表仅展示该部门用户	通过测试
5	编辑用户	点击用户右侧“编辑”按钮，输入信息	用户信息成功变更	通过测试

续表 5-3

序号	功能描述	测试方法	预期结果	是否通过测试
6	删除用户	点击用户右侧“删除”按钮，在弹窗点击“确认”	用户从用户列表消失，无法登陆系统	通过测试
7	启用用户	点击用户右侧“启用”按钮	用户恢复正常状态，可以正常登陆系统，进行操作	通过测试
8	禁用用户	点击用户右侧“禁用”按钮	用户无法登陆系统	通过测试
9	重置密码	点击用户右侧的“重置密码”，在弹窗点击“确认”	该用户密码被重置为“123456”	通过测试

(4) 元器件分类管理功能测试如表 5-4 所示。

表 5-4 元器件分类管理功能测试

序号	功能描述	测试方法	预期结果	是否通过测试
1	新增元器件类别	在元器件分类页面点击“新增”，在下拉框选择大类，输入其他信息	该分类不存在时，元器件分类表成功添加此分类；若分类已存在，系统提示无法重复添加	通过测试
2	编辑元器件类别	在元器件分类页面点击“编辑”，输入信息	该元器件分类信息成功变更	通过测试
3	删除元器件类别	在元器件分类页面点击“删除”	该分类下有具体型号时，系统提示无法删除；若下无具体型号，成功删除	通过测试
4	新增元器件型号	在元器件型号页面点击“新增”，在分类下拉框选择分类，并填写其他信息	该型号不存在时，元器件型号表成功添加此分类；若型号已存在，系统提示无法重复添加	通过测试
5	编辑元器件型号	在元器件分类页面点击“编辑”，输入信息	该元器件型号信息成功变更	通过测试

续表 5-4

序号	功能描述	测试方法	预期结果	是否通过测试
6	删除元器件型号	在元器件型号页面点击“删除”	该型号库存数量不为 0 时，系统提示无法删除；若无库存，成功删除	通过测试

(5) 元器件采购管理功能测试如表 5-5 所示。

表 5-5 元器件采购管理功能测试

序号	功能描述	测试方法	预期结果	是否通过测试
1	新增采购单	在采购申请页面点击“添加”，从已存在的型号导入所需型号，填写数量、金额信息。点击“新增采购单”	成功新增该采购单	通过测试
2	编辑采购单	点击采购单右侧“编辑”按钮，填写采购信息	未提交的采购单可以成功更改信息，已提交的采购单无法编辑	通过测试
3	删除采购单	点击采购单右侧“删除”按钮	未提及的采购单可以删除，其他状态采购单无法删除	通过测试
4	提交审批	选择采购单，点击“提交审批”，在弹窗选择供应商，点击“确定”	采购单状态变更为“已提交”，管理员可以在审核页面看到提交的采购单	通过测试
5	入库	点击已审批通过的采购单右侧“入库”按钮，填写数量、仓库	仓库库存更新	通过测试
6	审核	管理员可审批已提交的采购单，点击“审核通过”，或“审核驳回”	该采购单审核状态更新	通过测试

(6) 元器件领用功能测试如表 5-6 所示。

表 5-6 元器件领用管理功能测试

序号	功能描述	测试方法	预期结果	是否通过测试
1	申请领用	点击领用申请页面，使用上方搜索框搜索需要的元器件按名称或型号，点击元器件右侧的“申请领用”按钮	点击搜索后页面展示所有该元器件库存及信息，申请领用后该申请出现在我的申请页面，管理员可以看到该申请	通过测试
2	取消申请	在我的页面，还未审批的领用申请右侧点击“取消申请”按钮	该申请被撤回，从我的申请和管理员审核页面清除	通过测试
3	审核通过/驳回	在申请审核页面点击“审核通过”或“审核驳回”	元器件状态及用户申请页面状态同步更新	通过测试
4	销毁元器件	在元器件领用明细页面点击“销毁”	该元器件状态变为“已销毁”，无法再被领用	通过测试

6 结论

6.1 总结

本次电子元器件管理系统的设计历经需求分析、数据表设计、代码编写等一系列工作。在比较分析现有技术后,选择使用 Vue 与 Springboot 结合的前后端分离框架编写程序,最终完成该系统开发。系统开发选取 Vue.js 技术完成前端开发,其组件式开发的特点使得系统的可维护性很强。后端采用 Springboot 框架配置 Maven 依赖,大大化简了开发流程。MybatisPlus 使得对数据库的读写变得简易,可以直接使用 java 语言而非 sql 语句。

技术难点集中在数据表的设计,由于元器件种类繁多,信息量庞大,所设计的系统的多个功能模块会从不同的数据表抓取数据。功能模块的设计与实现则是核心工作,实现了元器件的级联分类、元器件的采购与审核、元器件的申请与领用等功能,完整覆盖所有业务需求。与市场上现存的 erp 系统相比,体量较轻,操作方式简单友好,达到预期目标。

6.2 展望

此次电子元器件管理系统的设计,尽管已经覆盖使用的大部分需求,但由于本人时间紧张,能力有限等多方面因素限制,系统仍存在很多不足之处,希望能在未来继续研究,完善系统功能。其中主要缺陷记录如下:

- (1) 前端获取数据信息时响应较慢,查询方式可进一步优化。
- (2) 为便于二次开发,将菜单栏数据也存储在数据库中,如果出现数据库被攻击导致信息丢失的情况,系统易崩溃故障。
- (3) 没有搭建备份服务器,难以提供不间断的系统服务。

参考文献

- [1] 张晓宁. 国家级实验教学示范中心建设状况[J]. 实验室研究与探索, 2009, 28(10): 85-88+161.
- [2] 鄢京. 基于机器视觉的电子元件检测系统设计[J]. 自动化应用, 2022, (04): 79-80+83.
- [3] 何丽娟, 赖路燕. 电子元件精细化管理系统构建研究[J]. 电子世界, 2013, (12): 154-155.
- [4] 池晓宇, 彭泽顺, 沙毅等. 高校实验室电子元件智慧管理系统设计与实现[J]. 科技视界, 2018, (36): 55-57.
- [5] 王钊. 数据库技术在信息管理中的应用[J]. 现代工业经济和信息化, 2023, 13(01): 142-144.
- [6] 杨晓君. 数据库技术发展概述[J]. 科技情报开发与经济, 2011, 21(03): 152-153+156.
- [7] 李昕航, 李超, 张桂刚, 邢春晓. 区块链与数据库技术融合综述[J]. 计算机科学与探索, 2023, 17(04): 761-770.
- [8] 张华强. 关系型数据库与 NoSQL 数据库[J]. 电脑知识与技术, 2011, 7(20): 4802-4804.
- [9] 程华, 高挺, 潘冲. 面向对象数据库在电信网络管理中的应用[J]. 信息与电脑(理论版), 2019, (11): 151-152.
- [10] 李业田. 常用数据库类型介绍与解析[J]. 电子世界, 2018, (20): 101.
- [11] 钱宝健, 汪鹏, 曹如权. 基于 Spring Boot 的物流寄查系统设计与实现[J]. 电脑编程技巧与维护, 2023, (09): 63-65+69.
- [12] 黄强文, 曾丹. 基于 Spring Cloud 和 Docker 的分布式微服务架构设计[J]. 微型电脑应用, 2019, 35(06): 98-101.
- [13] Tummalapalli S, Machavarapu VR. Managing Mysql Cluster Data Using Cloudera Impala[J]. Procedia Computer Science, 85: 463-474.
- [14] Schwartz B, Zaitsev P, Tkachenko V, High Performance MySQL: Optimization, Backups, and Replication[M]. O'Reilly Media, Inc. 2012.
- [15] 万志强. 基于 Java 语言的地铁数据库网站设计与实现[J]. 现代信息技术, 2024, 8(07): 25-29.
- [16] Wohlgethan E. Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue. js[D]. Hochschule für Angewandte Wissenschaften Hamburg, 2018.
- [17] 张汉昌. 基于 MVVM 模式的企业人事管理系统设计与实现[D]. 西南民族大学, 2023.
- [18] 王东升. 基于 SpringBoot 的闲置资产盘活系统研究与实现[D]. 烟台大学, 2024.
- [19] 陈小燕, 朱映辉, 余晓春. 基于 SpringBoot+Vue 的好农作物商城的设计与实现[J]. 电脑知识与技术, 2022, 18(22): 37-39.
- [20] 万建民. 基于 Netty 和 Redis 应对高并发场景的研究和实现[D]. 南京邮电大学, 2023.

-
- [21]Z.Q. Z ,F. W ,B.J. X , et al.A new remote web-based MDSplus data visualization system for EAST[J].Fusion Engineering and Design,2023,186

致谢

行文至此，百感交集，本科时光即将落幕，新的征程仍在继续。首先我想感谢自己。这半年时光里，毕设、实习、留学多管齐下，致使我时常被焦虑淹没，又一步步梳理思路，带着信念重振旗鼓。感谢自己的独立与坚韧助我跨越重重挑战。

感恩我的家人，父母不善表达，但永远给予我安全感和尊重，支持我的每一个决定，鼓励我大胆随心去做。弟弟总会无条件的、真挚热情地表达爱，让我在数个心情低落时刻感受到温暖。

感谢我的密友妍、瑾、缘，虽身处异地，言语的关怀与深切的情谊仍能将我们连接。我们衷心为对方取得的成就而欣喜，更不吝在低谷时托举对方，朋友们是我的避风港。感谢我的舍友，四年相逢，数次交谈至深夜，我们剖析自己、赞美对方，我们越来越贴近、理解彼此，即将分离，诸多不舍。

感谢台剧《俗女养成记》，带给我太多感动，感谢陈嘉玲的勇敢，她乌托邦式的破局触发我诸多思考。

最后，感谢我的论文指导老师邓开连老师，以及答辩组的李晓丽、刘浩、徐海芹、高世萍老师，感谢老师们提出的建议，让我能顺利完成本次毕业设计。