西 南 交 通 大 学

本科毕业设计（论文）

智慧库房Web端前后台设计与实现

年 级： 2016 级

学 号： 2016112385

姓 名： 李 松

专 业： 计算机科学与技术

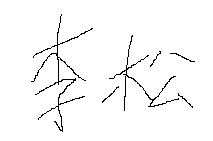
指导教师： 陈 剑 波

二零二零年五月

西南交通大学

本科毕业设计（论文）学术诚信声明

本人郑重声明：所呈交的毕业设计（论文），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成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。



作者签名：

日期： 2020 年 05 月 30 日

西南交通大学

本科毕业设计（论文）版权使用授权书

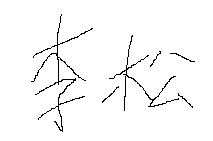
本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南交通大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

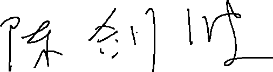
**保密**□，在 年解密后适用本授权书。

本论文属于

**不保密**☑。

（请在以上方框内打“🗸”）



作者签名： 指导教师签名：

日期： 2020 年 05 月 30 日 日期：2020 年 05 月 30 日

**指导教师评语**

院 系 信息科学与技术学院 专 业 计算机科学与技术

年 级 2016级 姓 名 李松

题 目 智慧库房Web端前后台设计与实现

指导教师

评 语 该同学在毕业设计过程中，基本能够保持与导师沟通，并在导师的指导下学习了相关的系统研发技术。

论文选题结合导师实际项目，基于B/S模式，使用Spring Boot后台框架进行开发，前端采用当前流行的MVVM模式结合WebSocket通信技术设计并实现了一个库房工器具出入库系统。主要实现了工器具类型管理、工单管理、工器具入库推荐、工器具出库等功能模块。

论文选题与专业相关，格式规范，有一定工作量，已达到西南交通大学本科毕业设计要求，同意提交答辩。

指导教师  (签章)

**评阅人评语**

院 系 信息科学与技术学院 专 业 计算机科学与技术

年 级 2016级 姓 名 李松

题 目 智慧库房Web端前后台设计与实现

评 阅 人

评 语



评阅人 (签章)

**答辩成绩**

院 系 信息科学与技术学院 专 业 计算机科学与技术

年 级 2016级 姓 名 李松

题 目 智慧库房Web端前后台设计与实现

成 绩

答辩委员会主任 (签章)

2020年 05 月 30 日

**毕业设计（论文）任务书**

班 级 计算机2016-01班 学生姓名 李 松 学 号 2016112385

发题日期：2019年11月22日 完成日期：2020年5月30日

题 目 智慧库房Web端前后台设计与实现

1、本论文的目的、意义

本系统与国家电网配网不停电作业管控平台（系统）相关，是在基于PMS2.0系统应用的基础上实现功能延伸补充，达到软件系统对配网不停电作业的全过程管控。管理内容主要包含：计划管理、培训管理、计划作业全过程管理、抢修作业全过程管理、智能库房应用、全年工作大数据分析。配网不停电作业管控平台基于公司内网进行通信，实现对各个驻点（各基层单位带电作业专业）的有效管控。

作为计算机专业的本科生，选此题目作为毕业设计，能够了解电力行业10kv不停电作业系统流程及相关工器具的使用和维护，并在系统实现中学会Java Web基本开发技能，并通过WebSocket与基于C++语言编写的客户端通信，具有相当重要的学习意义。

2、学生应完成的任务

（l）阅读相关文献，调查、理解分析针对智慧库房Web端前后台的业务需求、使用环境、用户分类、系统组成结构、功能结构、数据组织结构等，撰写完成“智慧库房Web端前后台”需求分析报告。（支撑毕业要求达成指标点2.3，3.4，6.2）

（2）掌握B/S服务结构以及Java、WebSocket等相关系统开发设计语言、调试工具等，搭建系统开发环境。（支撑毕业要求达成指标点5.3，12.1）

（3）根据需求分析报告和掌握的系统开发技术，完成系统结构设计、业务功能和流程设计、数据库模型设计和数据库设计，撰写完成系统设计报告。（支撑毕业要求达成指标点3.3，7.2，11.1）

（4）进行程序设计、调试，并进行必要的验证测试，撰写应用系统安装、系统用户使用说明文档。（支撑毕业要求达成指标点4.3，10.2）

（5）完成本科毕业设计（论文）和翻译文献，完成毕业答辩PPT、答辩演示准备工作。（支撑毕业要求达成指标点10.1，12.2）

3、本论文与本专业的培养目标达成度如何？（如在知识结构、能力结构、素质结构等方面有哪些有效的训练。）

本论文支撑本专业以下毕业要求的达成：**（1）**能够通过查阅和分析文献，为计算机系统及工程的问题求解寻找方案，并认识到所求解的问题具有多种可能的解决途径（指标点2.3）；**（2）**能够针对特定需求确定目标，设计计算机系统框架、组成模块，合理组织/存储数据，基于适当的模型进行系统设计与实现，并体现一定的创新意识（指标点3.3）；**（3）**能够在解决方案中从技术、非技术（如经济、社会、健康、安全、法律、文化以及环境等）角度，对设计方案的可行性进行评价和分析（指标点3.4）；**（4）**能够采用科学方法对计算机系统及工程问题进行研究，通过实验对比、文献综合、归纳整理得到合理有效结论，并对其进行规范表述（指标点4.3）；**（5）**能够利用开发环境和工具，对计算机系统及工程问题进行模拟仿真和数据分析（指标点5.3）；**（6）**能识别、分析、评价特定需求的计算机系统在设计和实现中对社会、健康、安全、法律以及文化的影响，并明确自己应承担的责任（指标点6.2）；**（7）**能够评价计算机系统设计、开发、运行和维护对环境保护和社会持续发展的影响（指标点7.2）；**（8）**能够通过口头、文稿、图表等方式、陈述和表达自己的观点，能够就计算机系统及工程问题与同行和相关人员进行交流（指标点10.1）；**（9）**能够根据对工作内容和过程的记录与整理，撰写技术报告和设计文稿、陈述发言或回应质询（指标点10.2）；**（10）**了解计算机系统工程管理原理与经济决策方法，理解计算机系统项目的组织模式和实施过程，掌握项目管理原理和内容（指标点11.1）；**（11）**正确认识自主学习的必要性和重要性，认识到本专业是一个发展迅速的学科，具有自主学习和终身学习的意识（指标点12.1）；**（12）**具备自主学习新技术和新方法的能力，能够通过学习不断提高、适应信息技术和职业的发展（指标点12.2）。

4、论文各部分内容及时间分配：（共18周）

第一部分 查阅相关文献资料，功能需求分析 (3 周)

第二部分 学习开发工具、开发语言，组建开发环境 (2 周)

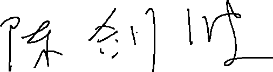
第三部分 系统结构分析、功能模块设计、程序设计 (4 周)

第四部分 程序调试、测试、完善、优化 (5 周)

第五部分 完成论文写作 (3 周)

评阅及答辩 论文撰写、修改、打印与装订 (1 周)

备 注



指导教师： 2019年11月22日

摘 要

信息时代的到来，使得人们的生活，工作，娱乐，社交等都有了极大地变化。计算机给人们带来便利的同时，也逐步进入到社会的各个行业领域中，将计算机这一信息处理利器应用于仓库的日常管理已是势必所然。采用计算机管理信息系统已成为仓库管理科学化和现代化的重要标志，它能给企业管理来了明显的经济效益和社会效益。主要体现在：极大提高了仓库工作人员的工作效率，大大减少了以往入出存流程繁琐，杂乱，周期长的弊端。基于仓库管理的全面自动化，可以减少入库管理、出库管理及库存管度理中的漏洞，可以节约不少管理开支，增加企业收入。提高管理水平对生产加工质量和效率以及降低人工管理成本有着重要的意义和价值。

本课题以Spring Boot技术为基础，设计并实现了一个库房管理后台系统。主要包括工器具管理，库房管理，车辆管理，节点管理，工单管理以及用户权限管理等模块；本文论述了该系统的设计思路、系统分析、功能结构、实现过程和关键技术。

在系统的实现过程中，采用B/S模式结构，项目前后端分离开发，前端使用Vue官方脚手架创建，后端使用Spring Boot技术框架，在持久层使用MyBatis框架，权限管理方面使用了Apache Shiro框架对用户登录以及权限进行控制。项目的亮点主要有两个，其一是项目可以通过WebSocket与基于C++语言编写的客户端通信，实现工器具的信息反馈，其二在于可以根据工器具的使用情况进行推荐。

关键词：库房管理系统；B/S架构；Spring Boot技术

Abstract

The advent of the information age has greatly changed people's lives, work, entertainment, and social interactions. At the same time that computers bring convenience to people, they have gradually entered into various fields of society. It is inevitable to apply computers as an information processing tool to the daily management of warehouses. The use of computer management information system has become an important symbol of the scientific and modern warehouse management, which can bring significant economic and social benefits to enterprise management. Mainly reflected in: greatly improved the efficiency of warehouse staff, greatly reducing the previous cumbersome, messy and long cycle of the deposit and withdrawal process. Full automation based on warehouse management can reduce loopholes in warehouse management, warehouse management and inventory management, and can save a lot of management expenses and increase corporate income. Improving the management level is of great significance and value to the quality and efficiency of production and processing and reducing the cost of manual management.

This subject is based on Spring Boot technology, and designed and implemented a warehouse management back-end system. It mainly includes modules such as tool management, warehouse management, vehicle management, node management, work order management and user authority management. This article discusses the design idea, system analysis, functional structure, implementation process and key technologies of the system.

In the implementation of the system, the B/S mode structure is adopted. The front and back ends of the project are developed separately. The front end is created using the official Vue scaffolding. The back end uses the Spring Boot technology framework. The persistence layer uses the MyBatis framework. The authority management uses the Apache Shiro framework. Control user login and permissions. There are two main highlights of the project. The first is that the project can communicate with the client written in C++ language through WebSocket to realize the information feedback of the tools. The second is that it can be recommended according to the use of the tools.

**Keywords:** Treasury management system; B/S architecture; Spring Boot technology

目 录

[第1章 绪 论 1](#_Toc41337510)

[1.1 背景与意义 1](#_Toc41337511)

[1.2 国内外发展（应用）现状 1](#_Toc41337512)

[1.3 论文所做工作及思路 3](#_Toc41337513)

[1.4 论文章节安排 3](#_Toc41337514)

[第2章 系统基础理论与技术介绍 4](#_Toc41337515)

[2.1 Vue.js前端框架简介 4](#_Toc41337516)

[2.2 MVC设计模式 5](#_Toc41337517)

[2.3 Spring与Spring Boot技术介绍 6](#_Toc41337518)

[2.3.1 Spring框架简介 6](#_Toc41337519)

[2.3.2 Spring Boot框架简介 7](#_Toc41337520)

[2.4 MyBatis持久层框架介绍 8](#_Toc41337521)

[2.5 本章小结 8](#_Toc41337522)

[第3章 系统需求分析 9](#_Toc41337523)

[3.1 系统需求描述 9](#_Toc41337524)

[3.1.1 总体目标 9](#_Toc41337525)

[3.1.2 可行性分析 9](#_Toc41337526)

[3.2 系统功能需求 10](#_Toc41337527)

[3.2.1 系统管理模块需求 10](#_Toc41337528)

[3.2.2 系统功能需求 11](#_Toc41337529)

[3.3 本章小结 12](#_Toc41337530)

[第4章 系统详细设计 13](#_Toc41337531)

[4.1 系统架构设计 13](#_Toc41337532)

[4.1.1 系统整体架构 13](#_Toc41337533)

[4.1.2 系统逻辑架构 14](#_Toc41337534)

[4.2 系统功能模块设计 15](#_Toc41337535)

[4.2.1 库房管理模块设计 15](#_Toc41337536)

[4.2.2 工器具管理模块设计 15](#_Toc41337537)

[4.2.3 车辆管理模块设计 15](#_Toc41337538)

[4.2.4 工单管理模块设计 15](#_Toc41337539)

[4.2.5 WebSocket通信模块设计 15](#_Toc41337540)

[4.2.6 节点管理模块设计 15](#_Toc41337541)

[4.3 系统管理模块设计 16](#_Toc41337542)

[4.3.1 用户管理功能设计 16](#_Toc41337543)

[4.3.2 角色管理功能设计 18](#_Toc41337544)

[4.3.3 日志管理功能设计 19](#_Toc41337545)

[4.4 系统数据库设计 20](#_Toc41337546)

[4.4.1 数据库概念设计 20](#_Toc41337547)

[4.4.2 数据库物理设计 22](#_Toc41337548)

[4.5 本章小结 25](#_Toc41337549)

[第5章 系统实现与测试 26](#_Toc41337550)

[5.1 系统开发环境 26](#_Toc41337551)

[5.1.1 硬件环境 26](#_Toc41337552)

[5.1.2 软件环境 26](#_Toc41337553)

[5.2 系统模块功能实现 26](#_Toc41337554)

[5.2.1 功能模块实现 26](#_Toc41337555)

[5.2.2 系统管理模块实现 30](#_Toc41337556)

[5.3 系统测试 34](#_Toc41337557)

[5.3.1 测试任务及目标 34](#_Toc41337558)

[5.3.2 系统功能测试 34](#_Toc41337559)

[5.4 本章小结 37](#_Toc41337560)

[结 论 38](#_Toc41337561)

[致 谢 39](#_Toc41337562)

[参考文献 40](#_Toc41337563)

第1章 绪 论

1.1 背景与意义

20世纪以来，计算机快速发展，信息技术已经进入到每个人的日常生活中，与此同时，信息技术的迅猛发展也成为社会发展的重要动力。它可以提高人们的工作效率，同时提高社会生产率。并且，信息网络技术的发展使经济全球化成为可能，这为全球化的经济发展提供了强大的动力。计算机和网络信息技术具有开放性，同步性，创新性和效率性的优势。这些优势支配着世界经济的发展，并影响着世界经济结构[1]。

管理仓库是一项繁琐而复杂的任务[2]。管理人员每天需要处理大量数据，并且需要生成各种数据报告并及时更新它们，例如进/出货物的时间，进/出的单位数，诸如人员管理，仓库维护，检查管理和仓库环境信息之类的数据。随着公司和机构的发展日益壮大，仓库的数量不断增加，产品的种类越来越复杂，使得仓库中的数据量呈爆炸式增长，并且仓库的操作也越来越复杂。使用依赖于人工记录和信息验证的传统仓库管理方法，已经使公司无法有效利用各种资源。它不仅效率低下，问题多，敏感度低，更难更新和维护，而且出错率更高 [3]。

仓库信息的爆炸式增长使数字仓库管理系统的建设尤为重要。基于数据库技术的仓库信息管理系统记录简单方便，查询快速可靠，并且大容量存储，信息安全，成本低[4]。

1.2 国内外发展（应用）现状

传统的库房管理方法使用手动记录来记录和跟踪进出货物和其他仓库数据。 这不仅效率低下，难以维护和更新，而且也很容易出错[5]。

就目前来看，仓储管理在美国、日本、欧洲等一些发达国家发展较为顺畅，在上个世纪七十年代，这些国家在仓储行业中普遍采用了计算机技术。最近一些年来，发达国家不断尝试和探索研发仓储管理信息系统，并且获得了较多的经验，现代物流技术的飞速发展，也为其提供了有利的条件，从而形成了一定的发展模式，不断运用新的管理方法以及思想，是设计仓储管理信息系统的关键所在。

以沃尔玛集团举例，沃尔玛是由美国创立的，其范围遍布全世界，是一家连锁企业，其效率更是令人惊叹，在技术方面的敏感以及投入是这家零售巨头有如此高的成就的原因：二十世纪六十年代运用计算机进行库存管理，二十世纪八十年代采用条码和电子数据交换技术（EDI），2003年广泛运用RFID技术，为了可以更好的将信息处理效率进行提高，在1987年，沃尔玛竟然花费数亿美元拥有了自己的专属卫星，这一举措尤为惊人，在总部和连锁店之间能够实时双向语音，可以方便快捷的传输数据，统一了条形码、库存管理、EDI、会员管理、补货以及收银等所有的系统，其效率之快据说仅次于五角大楼。沃尔玛的仓储系统的如此之快速发展，首先是他们超前的管理思想意识，尖端科技和物流系统巧妙的搭配，以及明确的规模效益目的。

与发达国家相比，我国的仓储管理信息系统仍处于初级阶段，还存在较大的差距，大部分的企业仓储管理系统服务水平不达标，并且功能比较单一，不能很好的针对自身行业特点需求进行有效的管理，实现信息对接和数据资源共享；大多数的仓库设备较为落后，甚至基本靠人力工作，业务量较大，达不到较高的效率，准确度不高，长久滞留损失也随之增大；仓库利用率低，基本小于40%，更有甚者长期处于闲置状态，还有仓库资源紧缺而盲目建立新库，造成成本增加；仓储管理水平低，仓储管理方面的法制法规不够健全。甚至还有些企业至今还没有对仓储成本重视起来，仍没有计划建立仓储管理系统，根据相关的研究可以得知，在货物生产总成本中，仓储成本占到了15%-70%，因此减少此方面的成本可以有效的降低生产成本，从而取得可观的经济效益[6]。

1.3 论文所做工作及思路

本次设计通过分析国内外库房管理系统研究现状，结合国家电网配网不停电作业管控系统对库房管理的需求，分析智能库房系统功能结构，研究和实现智能库房管理系统。

首先，先阅读国内外相关文献，总结仓库管理系统现状，分析库房管理系统特点，对比国内外库房管理系统存在的问题和弊端，结合国家电网配网不停电作业管控系统需求，提出智能库房系统需求分析。本系统主要由功能模块、系统管理模块组成。其次，提出库房管理系统需求分析，之后确定系统使用的开发路线和系统架构，根据系统需求分析详细设计系统的各个模块并设计数据库概念模型和物理模型，最后，进行系统的开发。具体是先将系统整个框架搭建起来，使得系统有个整体结构，然后设计数据库表，插入数据，完成系统开发任务。在开发完一个功能模块的的同时进行测试，保证功能完整正确。

在进行系统开发的时候，由于自身的能力有限经验不足，可能会遇到各种各样的问题，那么需要自己耐下心来，慢慢学习，从网上书上寻找解决办法。要相信，任何事情都有解决的办法的。

1.4 论文章节安排

本论文共分为五个章节。

第1章 绪论主要介绍本项目的研究背景与开发意义，国内外研究现状以及本文的主要思路和文章组织结构。

第2章 介绍了系统开发技术路线中所用到的关键技术，对其中的重要理论知识做了较为详细的介绍。

第3章 对系统做需求分析，分析规划系统的功能模块和系统的管理模块。首先对系统做需求描述，然后对每一个模块做需求描述。

第4章 系统的详细设计部分先介绍系统的整体架构，对系统有一个宏观把握，然后介绍系统逻辑结构，最后分模块详细介绍以及系统的数据库设计。

第5章 系统的实现与测试部分根据各个模块依次简要阐述模块实现的功能和模块的运行截图，最后进行系统测试保证各个模块的功能正确无误。

第2章 系统基础理论与技术介绍

2.1 Vue.js前端框架简介

Vue.js是一个用来开发web界面的前端库，也是一个轻量级开发工具。Vue.js不但属于响应式编程，还具有组件化的特点，恰如官方网站所言：通过尽可能简单的API实现响应的数据绑定和组合的视图组件。在Vue.js、AngularJS和ReactJS等前端MVVM框架都实现了使视图与状态保持一致性[7]（这种数据的同步更新即为数据绑定），MVVM可以拆分成：Model---ViewModel---View三部分，如图2-1 MVVM框架图所示。



图2-1 MVVM框架图

在图2-1中，左侧的Model相当于数据对象，右侧的View相当于页面DOM内容，中间用ViewModel进行连接，相当于页面的展示逻辑[8]。使用中间的监控者ViewModel对两侧的数据和操作进行监控，一旦一方发生改变，就通知相应的另一方完成数据的修改。Vue.js使得前端开发人员不必再编写复杂的DOM操作通过this来回寻找相关节点，大大提高了开发的效率。通过MVVM框架[9]，自动完成了数据和视图的同步更新。

Vue.js的“一切皆为组件[10]”理念，是指任意完成封装的代码块都可以被看作组件，例如：Vue.component(‘test’,test)，能够以<test></test>的形式在模板中调用。进行合理的抽象，就可以将网页划分成一个个区块，区块由大大小小的组件组成[11]。

2.2 MVC设计模式

MVC的全名是Model View Controller，是Model-View-Controller的缩写。 它是软件设计的模型，具有业务逻辑，数据和界面显示分离的三层模型。组织代码时将业务逻辑收集到组件中，同时改进个性化自定义界面和用户交互，无需重写业务逻辑[12]。

下图为MVC设计模式，如图2-2所示。



图2-2 MVC设计模式

模型（Model）是应用程序的一部分，用于处理应用程序数据逻辑。通常，模型对象负责访问数据库中的数据。视图（View）是应用程序中处理数据显示的部分。视图通常是基于模型数据创建的。Controller是应用程序中处理用户交互的部分。通常，控制器负责从视图中读取数据，控制用户输入并将数据发送到模型。

MVC设计模式是独特的，可以在逻辑图形用户界面结构中映射传统的输入，处理和输出功能。MVC模式最早由Trygve Reenskaug提出，成为施乐帕罗奥多研究中心(Xerox PARC)的Smalltalk语言发明的一种软件设计模式。如图2-3展示了MVC工作模式。



图2-3 MVC工作模式

MVC可以为以后的程序维护和扩展提供便利，并使重用程序的某些部分变得容易。 而且MVC还简化了程序，更加直观。MVC设计模式并不是java语言独有的设计模式，几乎所有的B/S结构的项目都在使用这种设计模式[13]。

2.3 Spring与Spring Boot技术介绍

2.3.1 Spring框架简介

Spring是一个开源框架，它由[Rod Johnson](https://baike.baidu.com/item/Rod%20Johnson" \t "_blank)创建。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。Spring的大致组成结构如图2-4所示：

图2-4 Spring框架组成结构

Spring框架的核心是面向切面编程[14]（AOP）思想，它通过依赖注入[15]（dependency injection）来减少耦合性。通过支持数据库和与Web技术，使得从开发到使用到维护的Web应用程序的难度都降低了。

2.3.2 Spring Boot框架简介

Spring Boot是Pivotal团队提供的全新框架。 它的设计目的是简化新的Spring应用程序的初始构建和开发过程。 该框架使用一种特定的方式进行配置，因此开发人员不再需要定义示例配置。 这样，Spring Boot成为快速应用程序开发的领导者。

Spring Boot框架中有两个思想：开箱即用[16]和约定优于配置[17]。开箱即用，意思是在开发过程中，将相关的依赖程序包添加到MAVEN项目的pom文件中，然后使用相应的注释替换繁琐的XML配置文件来管理对象的生命周期。 此功能使开发人员可以摆脱复杂的配置工作和相关的管理工作，而将更多精力放在业务逻辑上。约定胜于配置是一种软件设计范例，由 Spring Boot本身配置目标结构，开发人员向该结构添加信息。 尽管此功能降低了部分灵活性，并增加了BUG定位的复杂性，但它减少了开发人员需要做出的决策数量，同时减少了大量XML配置，并且可以自动执行代码编译，测试和打包。

本文采用Spring Boot框架开发项目，使用Maven [18]进行项目管理，使得开发人员更容易管理项目，减少配置。



图2-5 Spring Boot配置流程

2.4 MyBatis持久层框架介绍

ORM框架MyBatis是当前主流的持久层框架 [19]。它与Hibernate框架相比，可以配置动态SQL并可以通过配置决定SQL的规则。使用MyBatis可以很好处理一些复杂而且需要性能优化的项目。MyBatis的目标明确，就是使得数据持久层的开发变得方便简洁[20]。它的执行流程图如图2-6所示：

图2-6 MyBatis执行流程图

2.5 本章小结

本章确定了系统开发中使用的技术与框架，并简单介绍了这些内容。这将为系统的后续设计提供技术支撑。

第3章 系统需求分析

3.1 系统需求描述

3.1.1 总体目标

传统的仓库管理工作繁琐又复杂，严重影响企业的办事效率。为了提高企业的运行效率，充分利用资源，使用基于数据库技术的仓库信息管理系统刻不容缓。系统的总体目标就是开发一个基于B/S架构的仓库管理系统，具体功能包括用户登录，用户权限控制修改，用户个人信息修改，节点管理，库房管理，车辆管理，工器具管理，工单管理等。

为确保开发的一个新系统满足预期的期望，在确定系统的确切需求之前，需要与经理和管理者进行沟通。 系统需求的目的是调查和研究系统，并且开发人员对系统的功能和目标有清晰而深入的了解，因此对系统有一系列限制。 只有完全了解了客户需求，才能开始设计和开发。

3.1.2 可行性分析

在一个待开发的系统中，需要使用到各种技术或者框架，这些技术或者框架的稳定性和成熟度是系统开发成功与否的关键。并且，对于这些技术，开发人员是否可以使用，如何获得受保护的技术或者框架的使用权，这些都亟待解决。在本次库房管理系统的开发过程中，整体采用B/S体系架构，前后端分离开发，前端使用Vue.js前端框架并且使用丰富的element-ui组件库来提高开发效率，后台使用Spring Boot框架处理后台逻辑，包括数据库的数据操作和网络请求的处理。目前，这些技术框架是当前主流，有不少的公司以及开发人员在使用这些技术，有很多参考示例和问题解决办法，并且这些技术易于学习。

在大学期间，使用过这些技术，因此进一步降低了系统开发失败的可能。在遇到问题时，也可以向有经验的老师和同学询问。

由于是仓库管理系统，没有用户注册的功能，所有账号均事先分配好，每个用户拥有若干角色，每个角色对应一些系统功能菜单，即每个角色拥有不同的操作权限；这样，在某个用户登录到自己的账号后，系统只会加载此用户拥有的权限菜单。超级权限可以查看所有内容，即拥有所有权限。对于每一个功能模块，可以查看的用户均可以进行增删改查操作（某些模块由于需求可能没有某个操作）。

3.2 系统功能需求

智能库房管理系统总体功能分为两个模块：系统管理模块和系统功能模块。如图3-1所示。



图3- 1智能库房管理系统总体功能

3.2.1 系统管理模块需求

在系统管理模块中，主要包含用户登录，用户个人信息修改，用户删除，用户退出，角色添加，角色修改，角色删除等功能，如图3-2所示。



图3-2 系统管理模块

1. 用户登录

每个用户都有自己的账号和密码，用户登录是必需品，既然有用户登录，就要对用户登录状态进行维持，采用Apache Shiro安全框架即可解决问题。

1. 信息修改

用户登录系统之后，要可以对自己的个人信息进行修改，包括电话号码，用户签名等。其中，由于密码修改比较敏感以及用户头像设置的特殊性，对这两种修改单独进行。

1. 用户删除

用户登录后，若有系统管理的权限，才可以删除其他用户，自己的账号不能删除。此处涉及到下面介绍的角色权限问题。

1. 用户退出

这是管理系统的必须要素，此处与安全框架结合很好实现。

1. 角色添加

每个用户都有多个角色，每个角色拥有若干权限（菜单），添加角色的时候除了要设置角色名字外，还要设置角色权限。

1. 角色修改

角色修改时同角色添加类似，对权限进行添加或减少。

1. 角色删除

拥有系统管理权限的用户可以删除某个角色。这样综合在一起就实现了不同用户登录系统时菜单的动态加载，即实现了权限控制。

1. 日志管理

用户在进行系统操作的时候，如添加库房，登录系统，修改个人信息等，均会将操作存储下来，具有日志管理权限的人员可以查看。

3.2.2 系统功能需求

在系统功能模块中，包含节点管理，库房管理，工器具信息管理，工器具出入库管理，车辆信息管理，车辆出入库管理，工单管理等功能，如图3-3所示。

图3-3 系统功能模块



1. 节点管理

一个节点代表一个地方，一个节点可能只有一个库房，也可能有多个库房。节点管理要完成节点信息的查询，编辑，添加和删除。

1. 库房管理

库房从总体上来分有两大类，一类是工具库房，一类是车库。节点与库房是一对多的关系，这个部分要完成库房信息的查询，编辑，添加和删除。

1. 工器具管理

工器具与工单相关联，工单会使用工具，一个正在进行的工单会改变工具的状态；当工单结束时，工具归还，工具状态改变的同时工具使用度+1。这个部分要完成工器具信息的查询，编辑，添加和删除。

1. 工器具出入库

这里展示工器具的出入库状态，与工单状态相关联。这个部分只需完成信息的查询与筛选。

1. 车辆管理

车辆与工单相关联，工单可能会使用车辆，一个正在进行的工单会改变车辆的状态；当工单结束时，车辆归还。这个部分要完成车辆信息的查询，编辑，添加和删除。

1. 车辆出入库

这里展示车辆的出入库状态，与工单状态相关联。这个部分只需完成信息的查询与筛选。

1. 工单管理

工单中会使用一些工具和车辆，在新建一个工单的时候，优先推荐使用次数少的工具，做到资源充分循环利用。对于一个正在执行的工单，可以新增工具和车辆或者结束工单。由于工单的特殊性，不予删除。

3.3 本章小结

本章阐述了系统的需求并对系统的模块和模块具体功能进行了一定的说明，之后会更进一步，对这些内容做详细设计，为下一章系统总体设计提供必要的参考。

1. 系统详细设计

系统设计同样是项目开发过程中一个重要的阶段，此章节将阐述系统的详细设计，先介绍系统的整体架构，对系统有一个宏观把握，然后介绍系统逻辑结构，最后分模块详细介绍以及系统的数据库设计。

4.1 系统架构设计

4.1.1 系统整体架构

智能库房管理系统经设计划分可以分为四个部分，它们分别是用户计算机（浏览器），前端服务器，后端服务器以及数据库。

对智能库房管理系统的系统架构图进行展示，如图4-1所示。



图4-1 系统架构图

管理员在计算机通过浏览器访问前端服务器，前端服务器返回网页，在访问前端服务器的同时，前端服务器会发送网络请求到后端服务器请求数据来渲染页面，后端服务器对数据库进行连接和数据的增删改查。在开发过程中，浏览器选择的是火狐，前端服务器是基于node.js搭建的，后端服务器即Spring Boot中内嵌的Tomcat服务器，数据库使用MySQL。

在项目中还有后端服务器与C++客户端的WebSocket通信，通信的主要内容是库房中工具的提取与归还信息。本项目主要涉及的内容是图中网页应用的开发。

4.1.2 系统逻辑架构

大部分的Web系统都是基于下面这个逻辑架构，这个架构主要包括四层，从上到下分别是视图层，控制层，业务处理逻辑层和数据访问层。

视图层可以看做系统的网页部分，主要处理用户界面和系统信息的展示。

控制层主要处理视图层发送过来的HTTP请求并返回响应的数据，与业务逻辑层进行交流，可以看做系统后台的Controller层次。

业务逻辑层主要处理系统中功能模块的具体实现，其中，业务逻辑层又可以分为接口层和实现层，接口层定义操作，实现层实现接口层定义的操作，业务逻辑层可以看做系统后台的Service层。

数据访问层主要处理与数据库的的连接与对数据库的操作，操作包括增加，删除，修改，查询，可以看做系统后台的Dao层。本文使用MyBatis作为数据持久化的框架。

下面对库房管理系统的逻辑架构图进行展示，如图4-2所示。



图4-2 逻辑架构图

4.2 系统功能模块设计

所有模块的大致开发流程是先完成数据库表的建立与数据的插入，然后设计后台实体类，编写前端页面并设计好与后台数据交互的接口，最后完成各个接口的后台代码编写，运行测试这个模块，无误后即可开发下一个模块。

4.2.1 库房管理模块设计

库房从总体上来分有两大类，一类是工具库房，一类是车库。节点与库房是一对多的关系，这个部分要完成库房信息的查询，编辑，添加和删除。

4.2.2 工器具管理模块设计

工器具与工单相关联，工单会使用工具，一个正在进行的工单会改变工具的状态；当工单结束时，工具归还，工具状态改变的同时工具使用度+1，工具使用度会作为工具推荐的依据。这个部分要完成工器具信息的查询，编辑，添加和删除。

4.2.3 车辆管理模块设计

车辆与工单相关联，工单可能会使用车辆，一个正在进行的工单会改变车辆的状态；当工单结束时，车辆归还。这个部分要完成车辆信息的查询，编辑，添加和删除。车辆出入库展示车辆的出入库状态，与工单状态相关联。这个部分只需完成信息的查询与筛选。

4.2.4 工单管理模块设计

工单中会使用一些工具和车辆，在新建一个工单的时候，优先推荐使用次数少的工具，做到资源充分循环利用。对于一个正在执行的工单，可以新增工具和车辆或者结束工单。由于工单的特殊性，不予删除。

4.2.5 WebSocket通信模块设计

这个模块是智慧库房的体现，它会接收来自C++客户端的信息，然后后台进行信息的记录与业务处理。首先打开WebSocket连接，然后输入信息，这是后台会返回信息作为回应。同时根据客户端ID来区分多个客户端，可以与多个客户端同时进行通信，如图4-3所示。

4.2.6 节点管理模块设计

一个节点代表一个地方，一个节点可能只有一个库房，也可能有多个库房，每个节点有一个联系人进行维护。节点管理要完成节点信息的查询，编辑，添加和删除。



图4-3 WebSocket通信模块

4.3 系统管理模块设计

4.3.1 用户管理功能设计

1. 登录

用户在输入用户名，密码与验证码后，点击确定后，后台进行登录校验返回信息。若校验失败则前端提示用户名不存在，密码错误或者验证码错误。实现的具体过程是用户点击登录按钮后，前端发送post请求到后台，后台首先对存储在Session里的正确验证码与传输来的验证码进行比对，若正确则进行账户的校验，否则返回验证码错误信息；接着从数据库查询此用户名的密码，若查不到此用户，则返回用户名不存在的错误信息；最后将传输来的密码进行加密并与数据库中的加密密码进行比较，若正确则返回成功信息，前端进行页面跳转到系统管理页，否则返回密码错误信息。若成功登陆，则后台会对此登陆用户的登陆状态进行管理。流程图如4-4 所示。

1. 个人信息修改

将个人信息修改分为三个部分，一个是基本信息展示与修改，一个是密码修改，一个是头像修改。基本信息的修改主要是用户身份信息的修改，如电话号码，用户签名等，用户的密码修改会让用户输入旧密码，新密码与确认密码，输入完成后首先在前端进行新密码与确认密码的校验，然后将旧密码与新密码传至后端，在后端进行旧密码与数据库存储密码的对比，若正确，进行新密码的存储，反之则返回密码错误的提示信息。流程图如4-5所示

1. 其他用户管理

对于其他用户的管理，可以修改用户基本信息与删除用户。在修改用户时，可以对其的角色进行设置，如添加角色或者删除角色。



图4-4 用户登录流程图



图4-5 密码修改流程图



图4-6 用户管理过程流程图

4.3.2 角色管理功能设计

系统先查出所有角色使用表格列出，对每一条数据，即每一个角色可以进行修改删除操作，在表格的上方有添加角色按钮和搜索框。修改角色时根据菜单的树形控件，勾选或者去除某一个菜单，点击确定后即可完成此角色权限的修改。流程图如4-7所示。



图4-7 角色管理过程流程图

4.3.3 日志管理功能设计

当用户在进行某些操作的时候，如用户登录，修改个人信息，添加工具等，后台会使用基于注解的方式将该操作信息保存到数据库中，当用户有日志管理的权限时，打开日志管理界面就可以看到详细的信息（某用户在什么时候做了什么事）。

4.4 系统数据库设计

4.4.1 数据库概念设计

数据库管理系统需要采用某种数据结构来定义和存储需要管理的数据，数据结构的生成就需要数据模型，数据模型实现现实系统到计算机管理的转变。数据库概念设计是以需求分析的结果为依据，最后生成数据模型。最常用到的数据模型是实体联系模型（entity-relationship model, ERM），又叫E-R模型，是一种高级数据模型，广泛应用于现实世界的建模中。

在明确了系统的需求分析后，可以开始对整个数据库系统进行建模。先确定各个实体并分析每个实体需要的属性，再根据需求确定各个实体之间的关系，最后整个数据库概念设计完成。本文在绘制实体联系图的时候，使用了Visio 绘图工具。

在完成数据库概念设计后，就可以设计数据库表并插入数据，这就是下面将要介绍的数据库物理设计。



图4-8 用户角色权限E-R图

图4-9 节点库房E-R图



图4-10 工单工具车辆E-R图

其中车辆与库房是有多对一的关系，由所在库房ID标识，工具与库房从实际情况上来说，也有多对一的关系，简单考虑，就用工具位置属性标识了。

4.4.2 数据库物理设计

数据库物理设计的过程就是直接操作数据库，设计数据库的数据库表和表中的字段，设计时要结合数据库概念设计的成果E-R图，尽量减少数据库表之间的关系。还有重要的一点是，字段的数据类型要考虑好，因为这将影响到后台实体类的与数据库表的映射。本文数据库使用MySQL数据库，版本5.7，数据库可视化工具Navicat Premium 12。数据库名称是gradient\_project，数据库中主要包含以下几张数据库表。

表4-1 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| user\_id | Int | 是 | 否 | 主键，自动递增，用户ID |
| user\_sex | tinyint |  | 是 | 用户性别 |
| user\_name | varchar |  | 是 | 用户名 |
| user\_signature | varchar |  | 是 | 用户签名 |
| name | varchar |  | 是 | 姓名 |
| user\_password | varchar |  | 是 | 密码 |
| salt | varchar |  | 是 | 密码盐值 |
| phone | varchar |  | 是 | 电话号码 |
| enabled | varchar |  | 是 | 用户状态 |
| user\_avatar | varchar |  | 是 | 用户头像 |
| birthday | datetime |  | 是 | 生日 |

这个表记录系统的所有管理员信息，其中密码是加密处理的，用户是否可以登录根据enabled字段进行判断。

表4-2 角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| id | Int | 是 | 否 | 主键，自动递增，角色ID |
| name | varchar |  | 是 | 角色英文名 |
| name\_ch | varchar |  | 是 | 角色中文名 |
| create\_time | datetime |  | 是 | 创建时间 |
| modify\_time | datetime |  | 是 | 修改时间 |

角色表用来为不同的角色分配权限，利用角色id来记录可以为该角色显示的菜单，从而使得不同身份的账号所看到的管理页面是不同的。

表4-3 菜单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| id | Int | 是 | 否 | 主键，自动递增，菜单ID |
| path | varchar |  | 否 | 路径 |
| name | varchar |  | 否 | 菜单英文名 |
| component | varchar |  | 否 | 组件 |
| icon | varchar |  | 是 | 图标样式 |
| name\_ch | varchar |  | 否 | 菜单中文名 |
| parent\_id | Int |  | 否 | 菜单父ID |
| permission | varchar |  | 否 | 权限字符串 |

菜单表用来记录菜单的信息，每个菜单都有一个id方便进行权限管理时不同的身份进行选择是否显示该菜单。

表4-4 节点表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| id | Int | 是 | 否 | 主键，自动递增，节点ID |
| code | varchar |  | 是 | 节点编码 |
| name | varchar |  | 是 | 节点名 |
| principal | Int |  | 是 | 联系人ID |
| address | varchar |  | 是 | 节点地址 |
| note | varchar |  | 是 | 备注 |
| create\_time | datetime |  | 是 | 创建时间 |
| modify\_time | datetime |  | 是 | 修改时间 |

节点表用来存储系统中的所有节点，每个节点由一个联系人，即管理员进行管理。

表4-5 库房表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| id | Int | 是 | 否 | 主键，自动递增，库房ID |
| treasury\_code | varchar |  | 是 | 库房编码 |
| warehouse\_name | varchar |  | 是 | 库房名称 |
| warehouse\_type | Int |  | 是 | 库房类型 |
| node | varchar |  | 是 | 所属节点名 |
| note | varchar |  | 是 | 备注 |

库房表存储所有库房的信息，每个库房属于某个节点。

表4-6 车辆表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| id | Int | 是 | 否 | 主键，自动递增，车辆ID |
| vehicle\_name | varchar |  | 是 | 车辆名字 |
| vehicle\_code | varchar |  | 是 | 车辆编码 |
| vehicle\_location | Int |  | 是 | 所属库房ID |
| note | varchar |  | 是 | 备注 |
| create\_time | datetime |  | 是 | 创建时间 |
| modify\_time | datetime |  | 是 | 修改时间 |
| vehicle\_status | varchar |  | 是 | 车辆状态 |

车辆表存储所有库房中的车辆并管理它们的状态，每个车辆有库房所在地信息。

表4-7 工具表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| id | Int | 是 | 否 | 主键，自动递增，工具ID |
| tool\_code | varchar |  | 是 | 工具编码 |
| tool\_name | varchar |  | 是 | 工具名称 |
| tool\_type | Int |  | 是 | 工具类型 |
| tool\_location | varchar |  | 是 | 工具位置 |
| **续表** | | | | |
| note | varchar |  | 是 | 备注 |
| unit | varchar |  | 是 | 工具单位 |
| create\_time | datetime |  | 是 | 创建时间 |
| modify\_time | datetime |  | 是 | 修改时间 |
| use\_time | int |  | 是 | 使用次数 |
| tool\_status | varchar |  | 是 | 工具状态 |

工具表存储所有库房的工器具并管理工具的状态，使用次数是工具推荐的依据。

表4-8 工单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **是否允许空** | **备注** |
| id | Int | 是 | 否 | 主键，自动递增，工单ID |
| creator | Int |  | 是 | 创建者 |
| worksheet\_name | varchar |  | 是 | 工单名称 |
| worksheet\_value | varchar |  | 是 | 工单产值 |
| worksheet\_type | Int |  | 是 | 工单类型 |
| note | varchar |  | 是 | 备注 |
| begin\_time | datetime |  | 是 | 开始时间 |
| end\_time | datetime |  | 是 | 结束时间 |
| worksheet\_status | varchar |  | 是 | 工单状态 |

工单表存储所有工单的信息，包括已完成和未完成的，工单表会与工具表和车辆表有所关联。以上表格只是罗列了整个数据库表中的重要数据表，其他数据表是一些关系表，虽然字段比较少，但是表仍然是整个数据库中不可或缺的一部分。

4.5 本章小结

本章阐述了智能库房前后端管理系统的详细设计，对系统中的系统管理模块，系统功能等模块进行了详细的设计说明，并且结合数据库设计对核心模块进行了详细的讨论和说明。

第5章 系统实现与测试

5.1 系统开发环境

5.1.1 硬件环境

处理器：Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz

内存：12.00GB

硬盘：500GB

显卡：NVIDIA GeForce GTX 950M

5.1.2 软件环境

操作系统：Microsoft Windows 10 家庭中文版 (64位)

开发平台：IntelliJ IDEA 2019.3.3、JDK1.8；

浏览器：Firefox Browser 76.0.1；

数据库：MySql 5.7.28、Navicat Premium；

其他：Visio2019

5.2 系统模块功能实现

5.2.1 功能模块实现

1. 节点管理

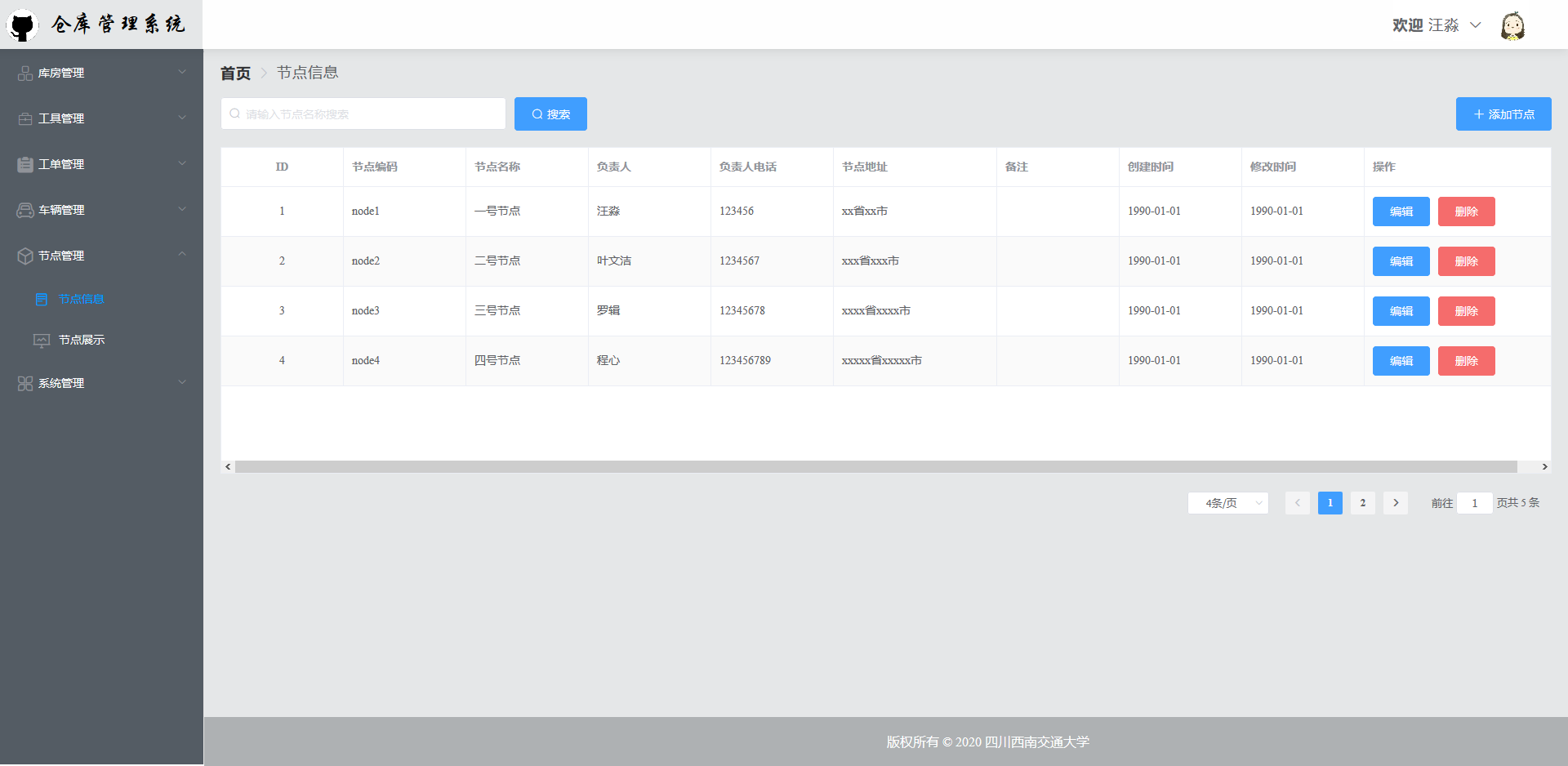
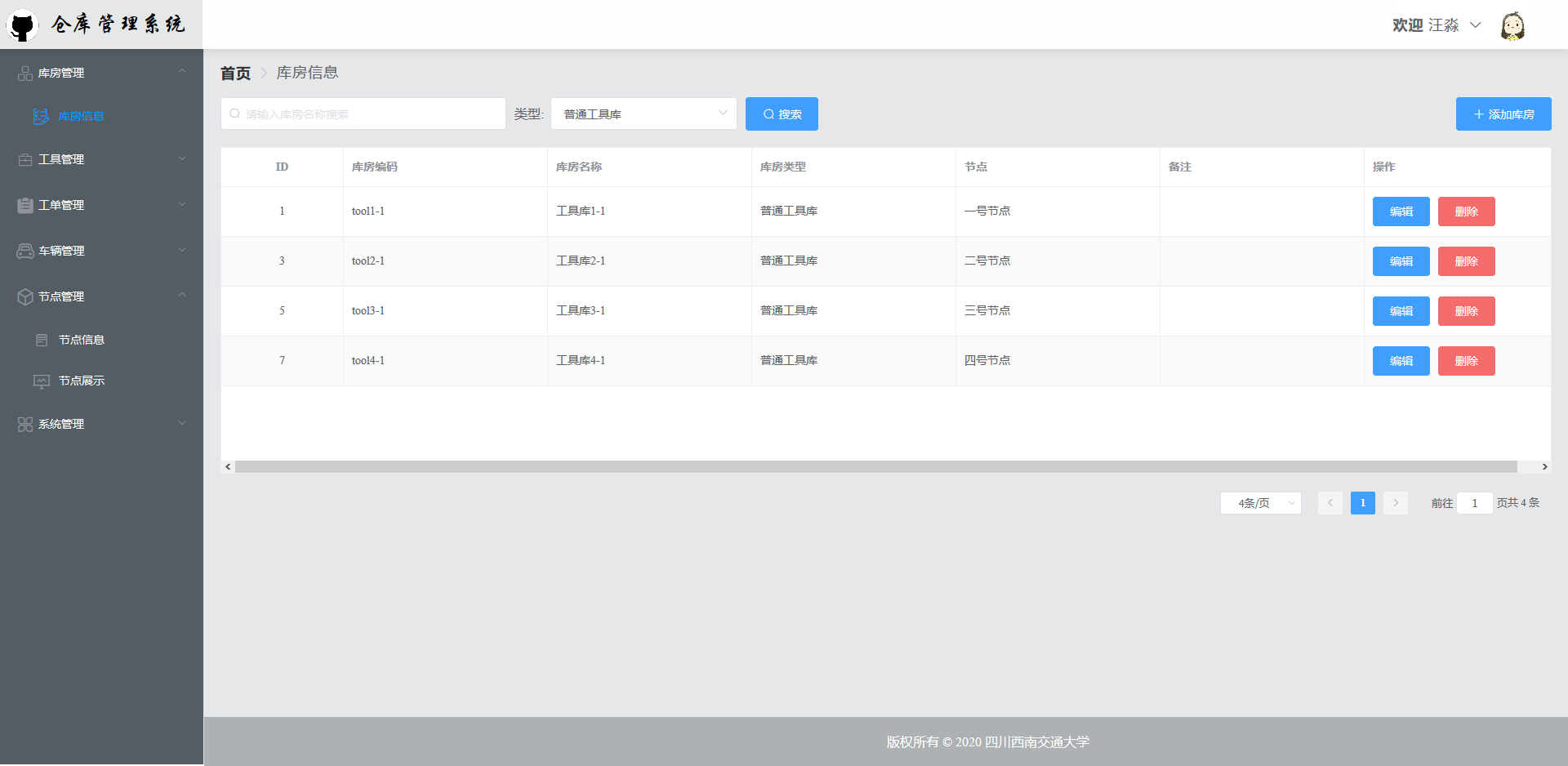
 这里展示节点信息，可以对节点进行增删改查。

图5-1 节点管理页

1. 库房管理

这里展示库房信息，同样可以进行增删改查操作，可以通过库房名或库房类型进行筛选。

图5-2 库房管理页

1. 工具管理

这里展示所有库房的工具信息以及工具的出入库信息，可以从中看出哪些工具正在使用中，同样可以进行增删改查操作，可以通过工具名进行筛选。工具信息是所有库房中存储的工具，包括工具购买时间，工具单位，工具位置以及工具信息的修改时间等，拥有此权限的用户可以对此进行工具信息的修改或者向库房里添加工具或者删除某个工具。工具出入库是对当前工具是否在库的管理，这些不在库的工具都会与当前正在进行的工单相关。

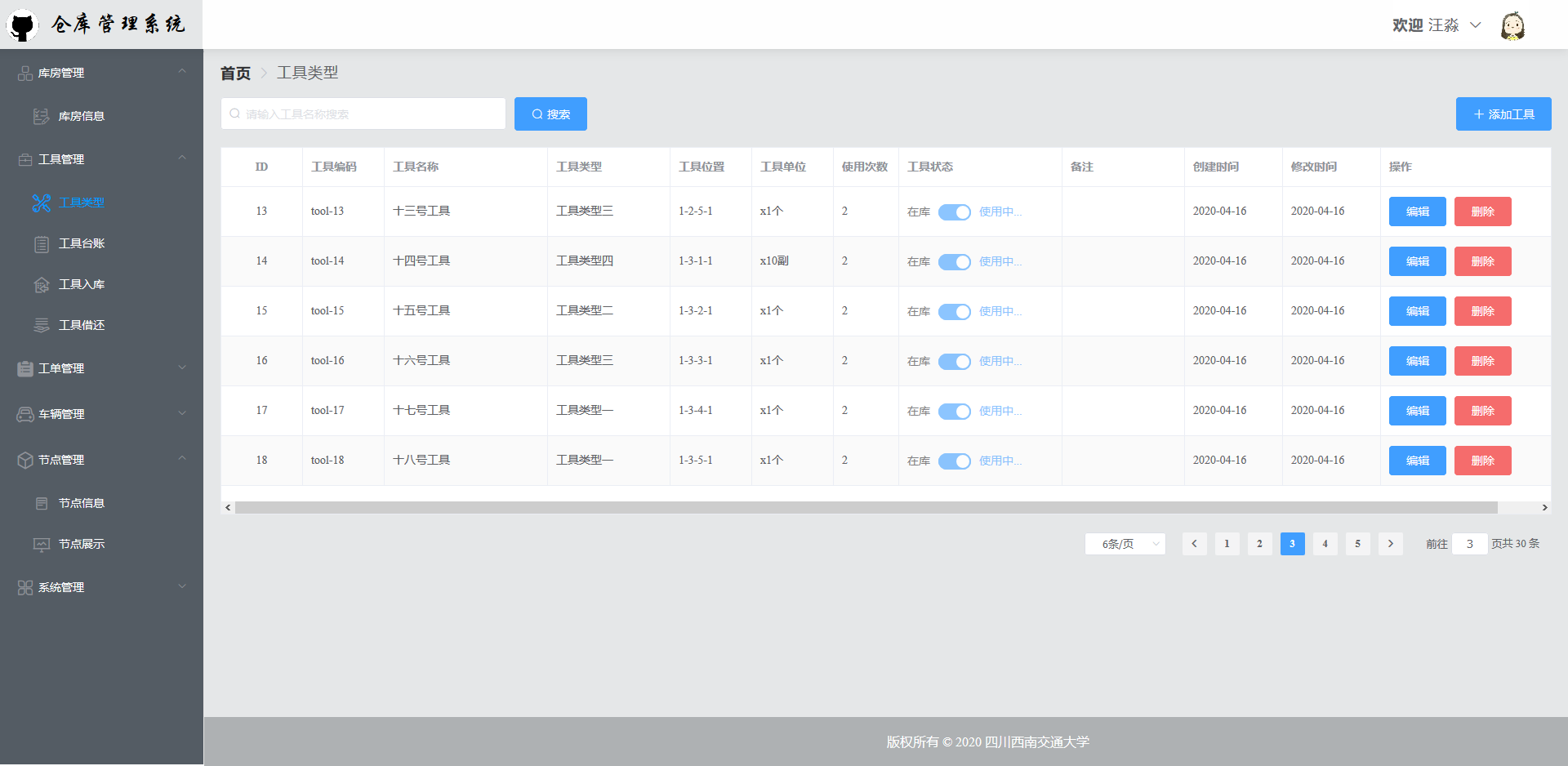
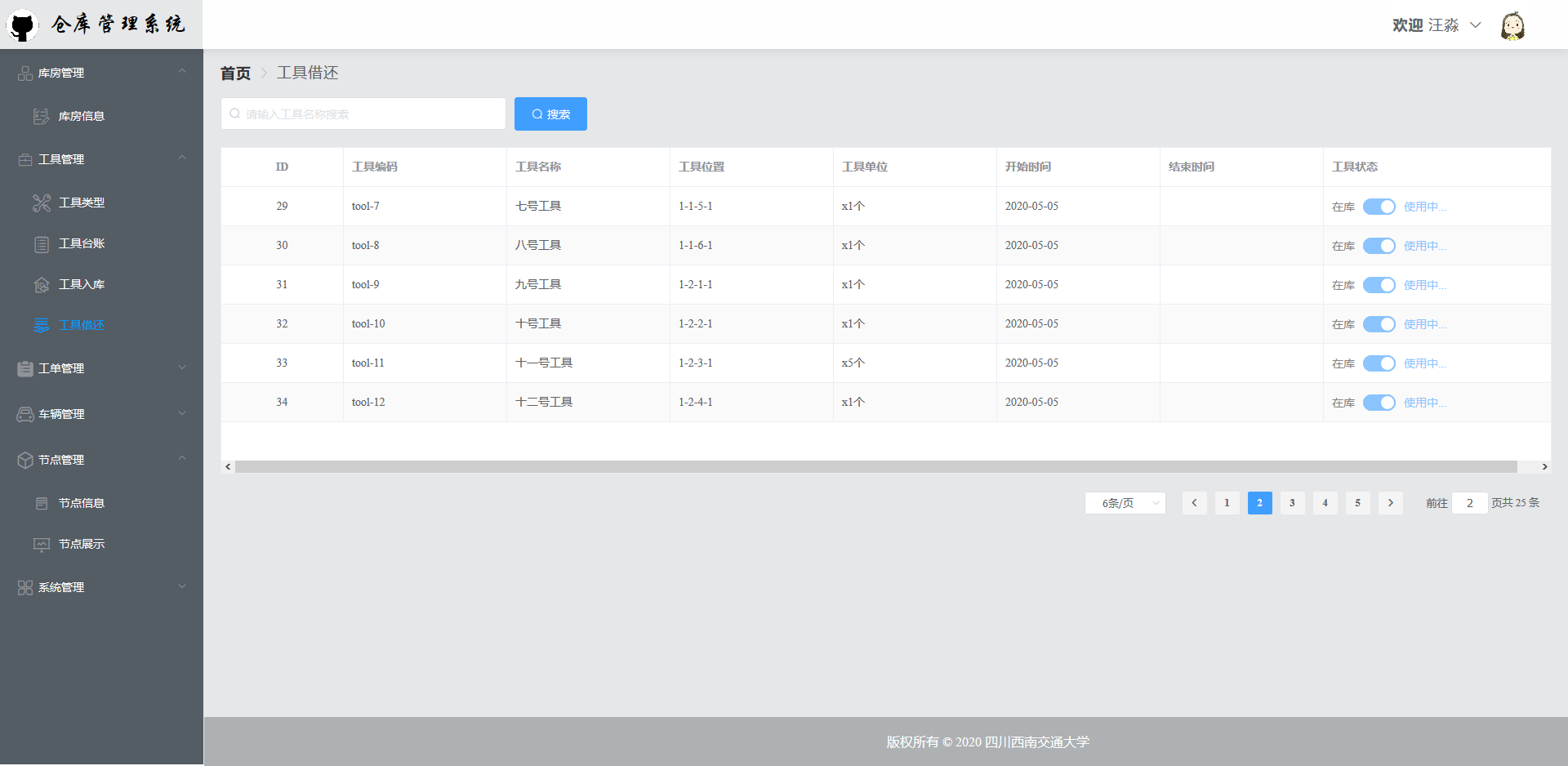
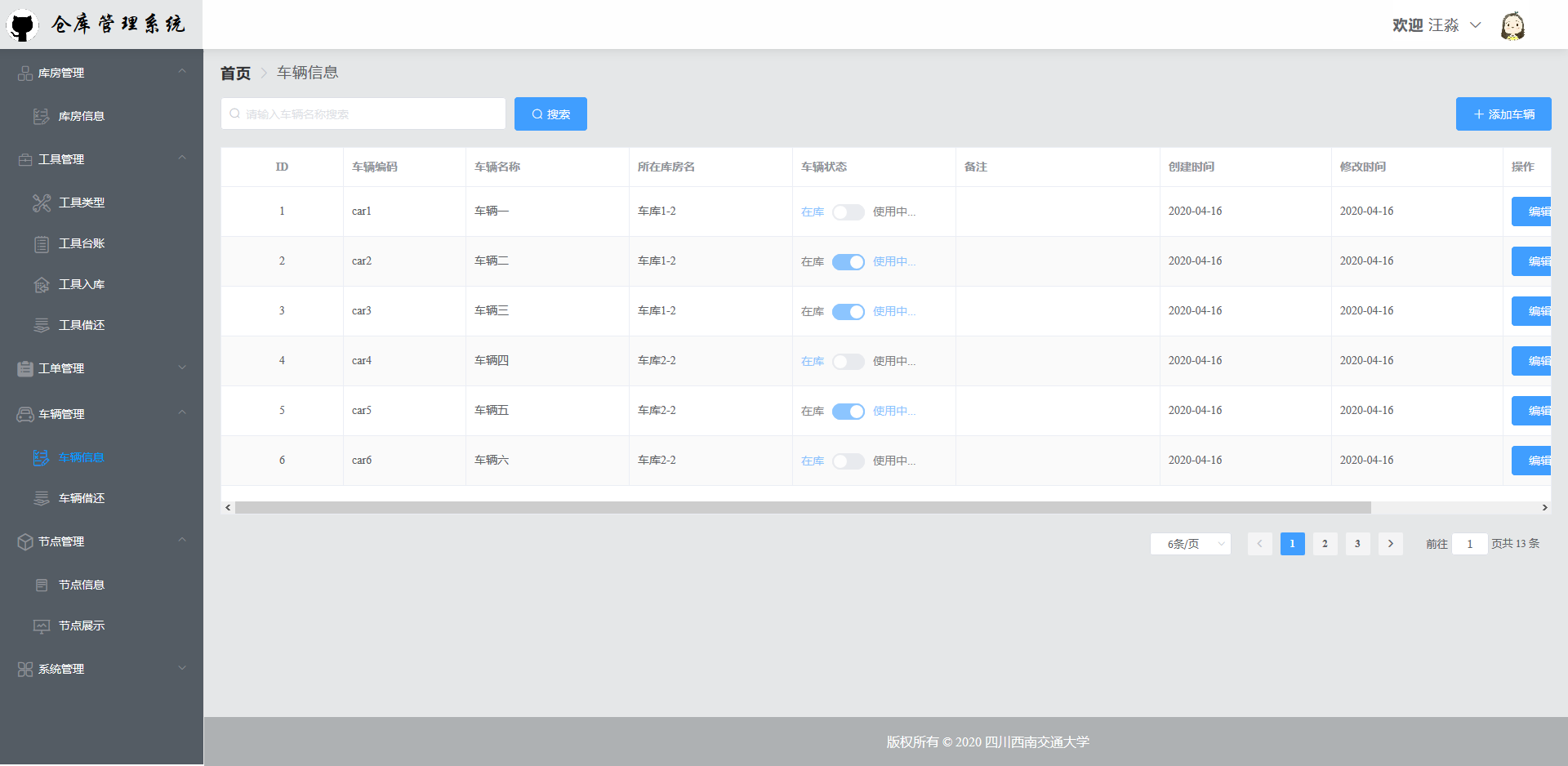
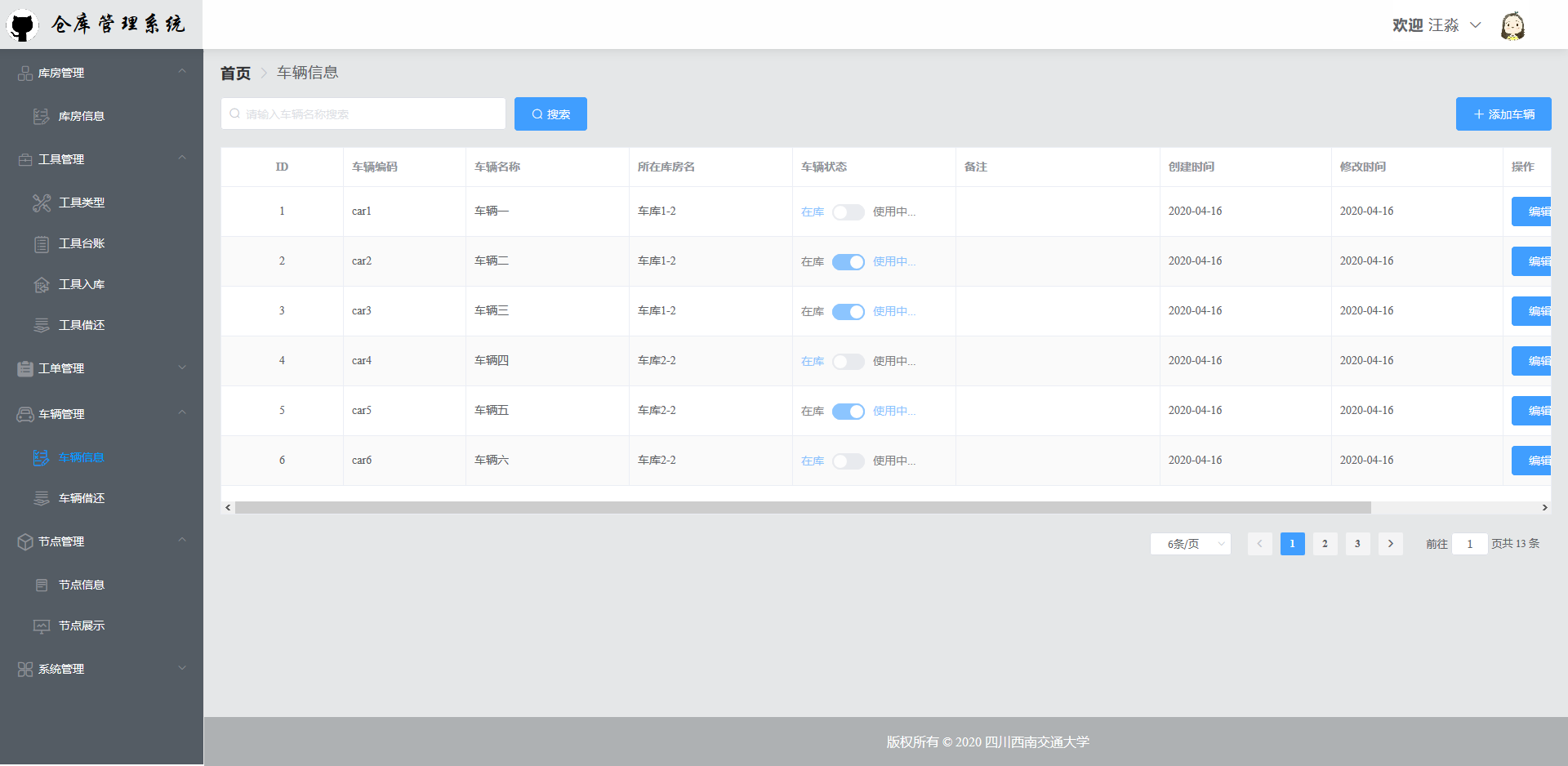
图5-3 工器具类型页面

图5-4 工器具出入库页面

1. 车辆管理

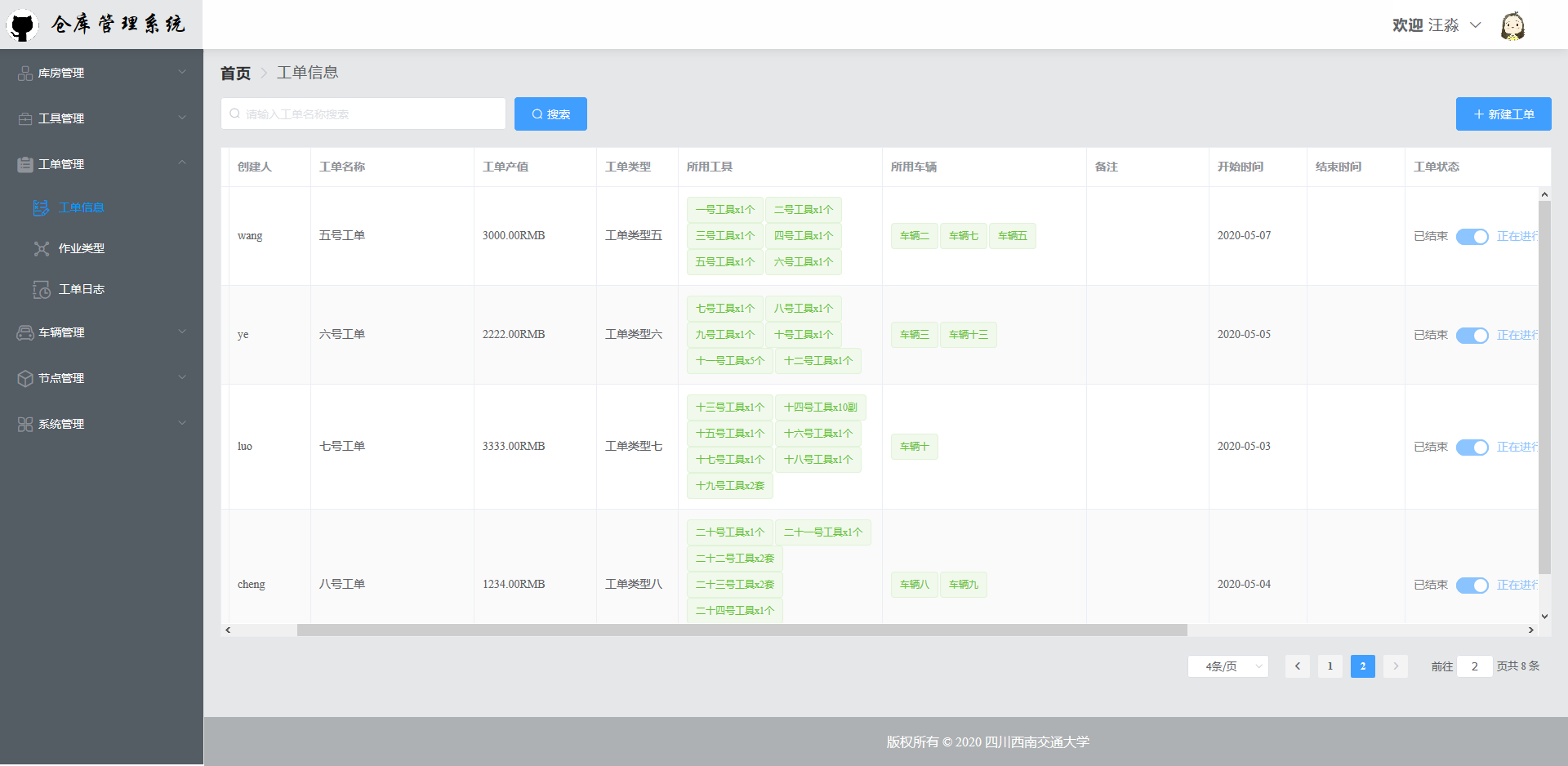
这里展示所有库房的车辆信息以及车辆的出入库信息，可以从中看出哪些车辆正在使用中，同样可以进行增删改查操作，可以通过车辆名进行筛选。车辆信息是所有库房中存储的车辆，包括车辆购买时间，车辆存储在哪个库房以及车辆信息的修改时间等，拥有此权限的用户可以对此进行车辆信息的修改或者向库房里添加车辆或者删除某个车辆。车辆出入库是对当前所有车辆是否在库的管理，这些不在库的车辆都会与当前正在进行的工单相关。

图5-5 车辆信息页面

图5-6 车辆出入库页面

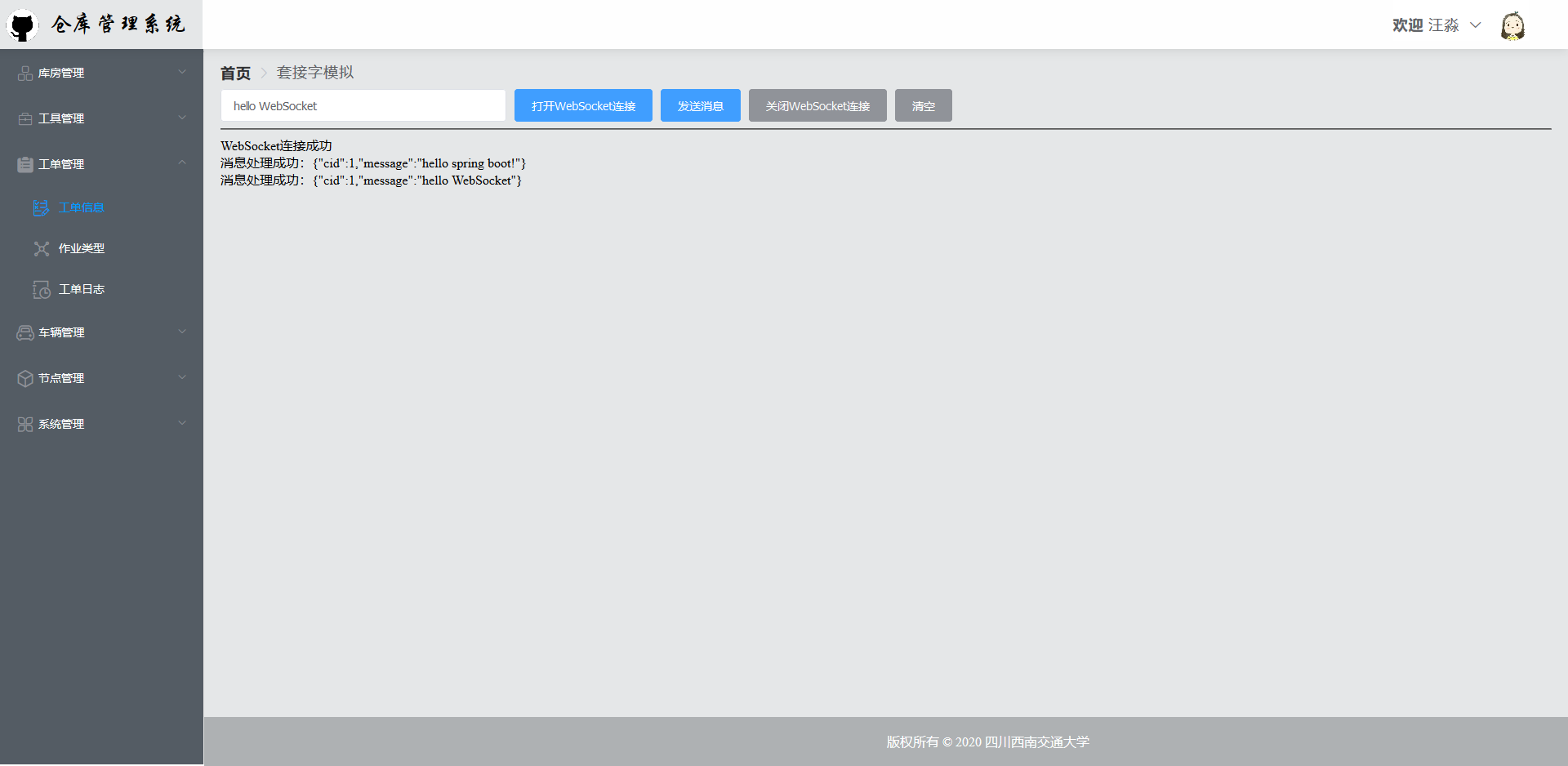
1. 工单管理

这个部分是整个系统中最为复杂的地方，在这里可以查看所有工单，包括已完成的和未完成的，对于未完成的工单，可以添加工具或者车辆，最后可以结束工单。同样，新建工单的时候，会推荐使用工具，根据工具的使用次数，使用次数少的优先推荐。

 图5-7 工单信息页

1. WebSocket模块

这个模块是模拟C++客户端的信息发送过程。模块负责与C++客户端进行通信，当工具拿出或归还，车辆驶离库房等情况发生时，C++客户端会发送信息到后端，后端进行信息记录或处理。这也是系统智慧的部分体现。

图5-8 WebSocket模块

5.2.2 系统管理模块实现

系统管理模块包括用户登录，用户个人信息的设置，用户管理以及角色权限的管理等几个部分组成。下面就这几个部分进行系统实现展示并简要说明，由于页面过多，仅进行部分展示。

1. 用户登录

系统首页就是用户登录页，用户必须输入正确的用户名及密码，还有正确的图片验证码才能登录成功，当然这些输入框均有非空校验。若有一个不正确，则返回提示信息，如用户名错误。登录成功后即进入管理页面，此时菜单已经根据此用户的权限动态加载，即此用户无未加载菜单的任何权限。

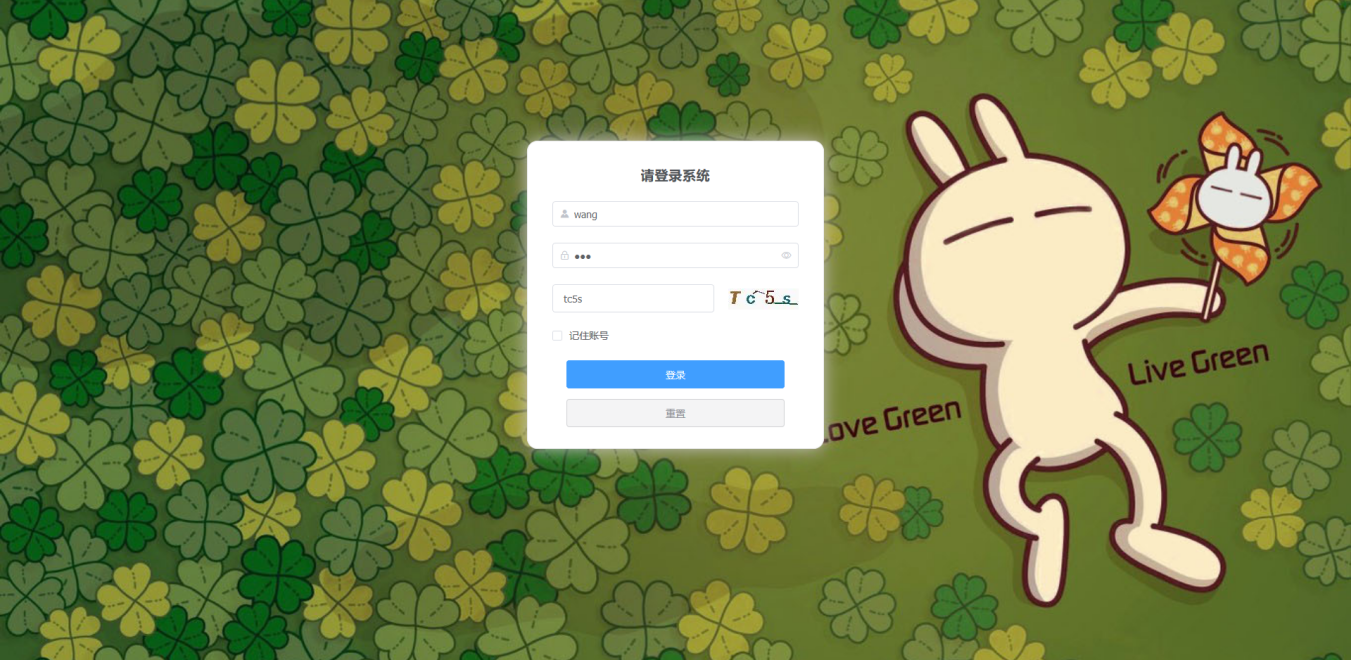
图5-9 用户登录页

图5-10 系统管理页

1. 个人信息展示与设置

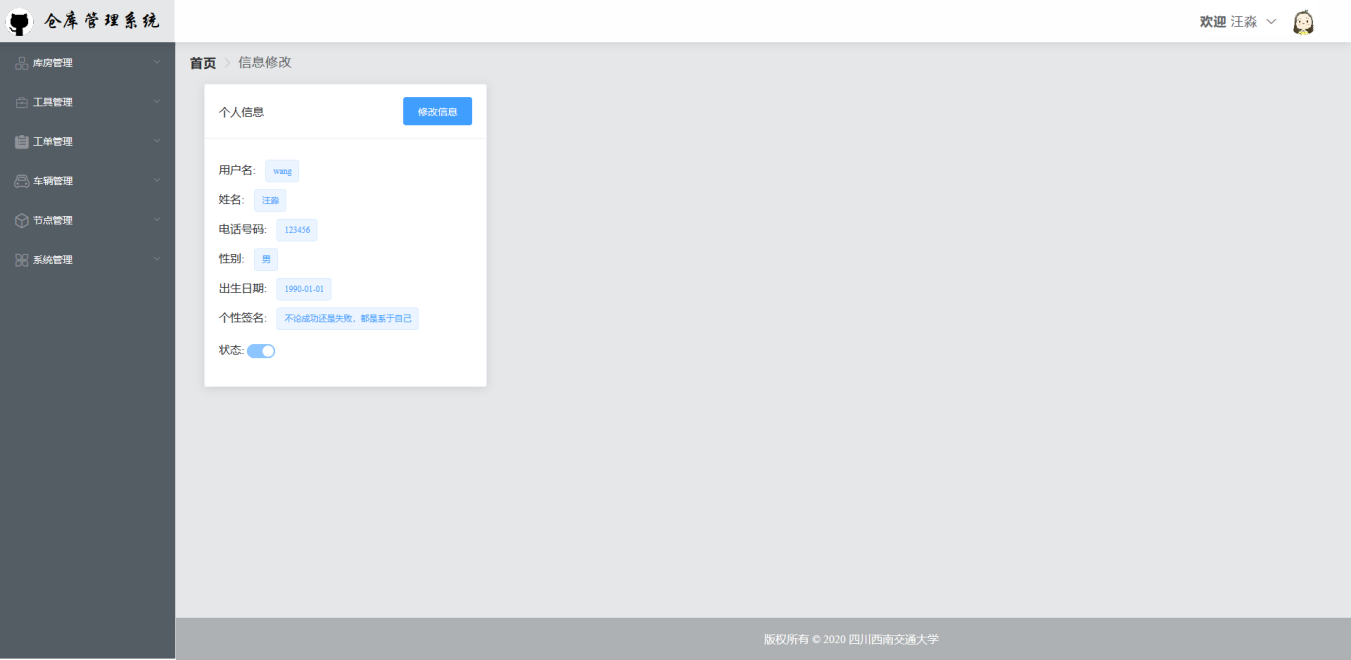
 这里会展示用户的基本信息，用户可以修改个人信息，修改头像，更改密码或者选择退出登录。在用户更改密码后，会强制让用户退出登录并重新登录，这样是为了加深用户的记忆以及为了系统的安全性着想。

图5-11 个人信息展示页

1. 用户管理

在这里会展示所有用户的信息，有权限的用户可以编辑其他用户的非重要信息或者删除此账号（登录用户无法删除自己的账号）。如果用户过多，会有分页效果并且可以通过名字进行模糊搜索。用户管理中可以设置用户的角色，这将影响到此用户的权限。

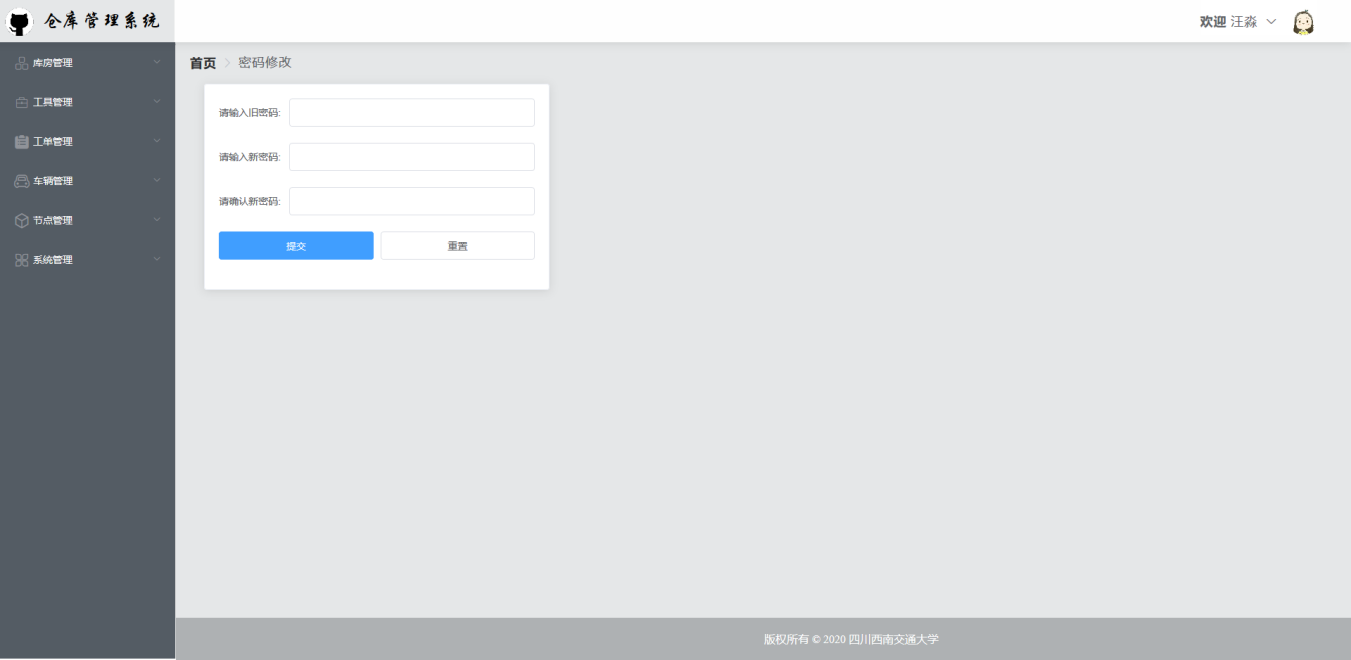
图5-12 密码修改页

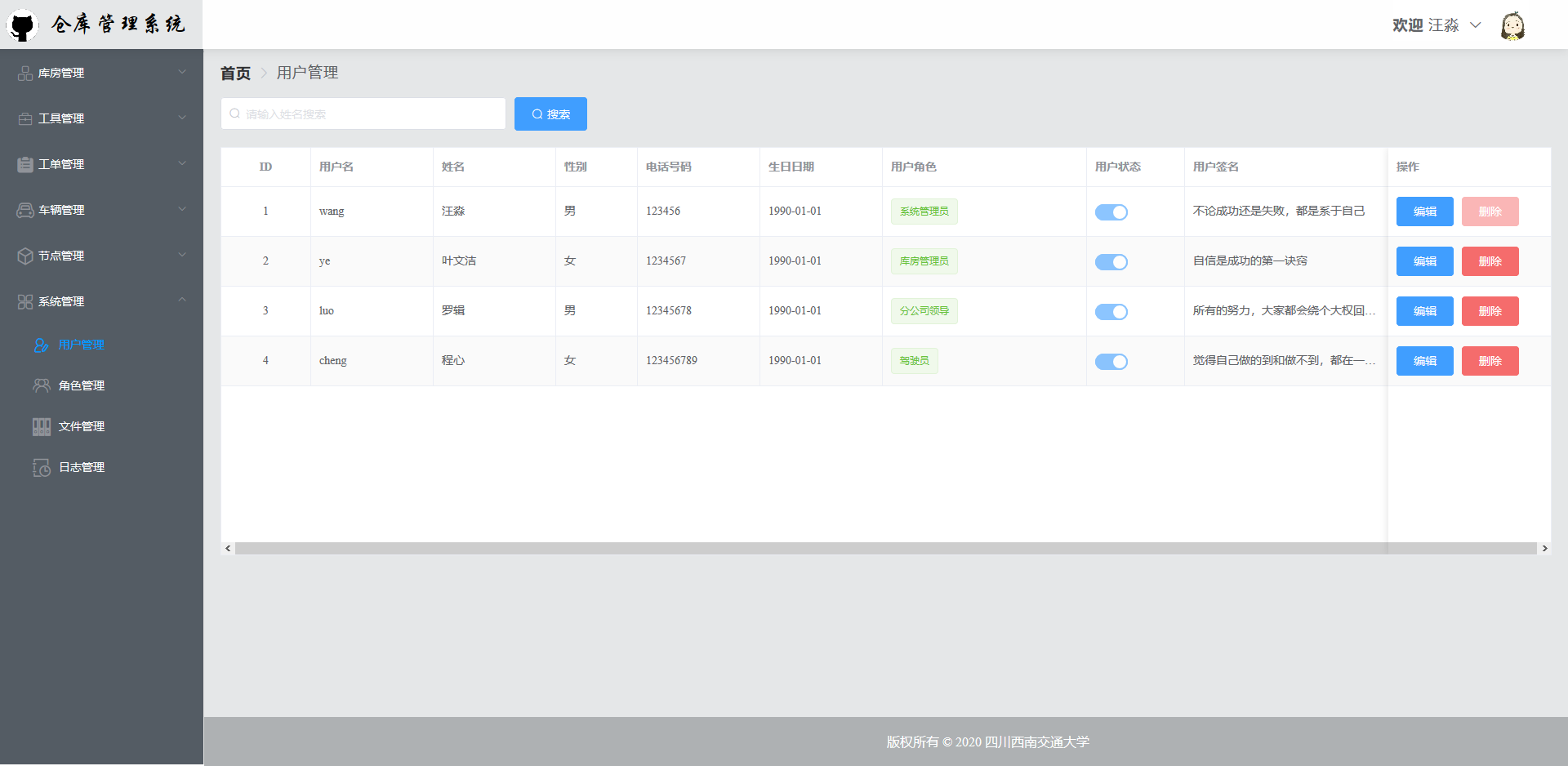
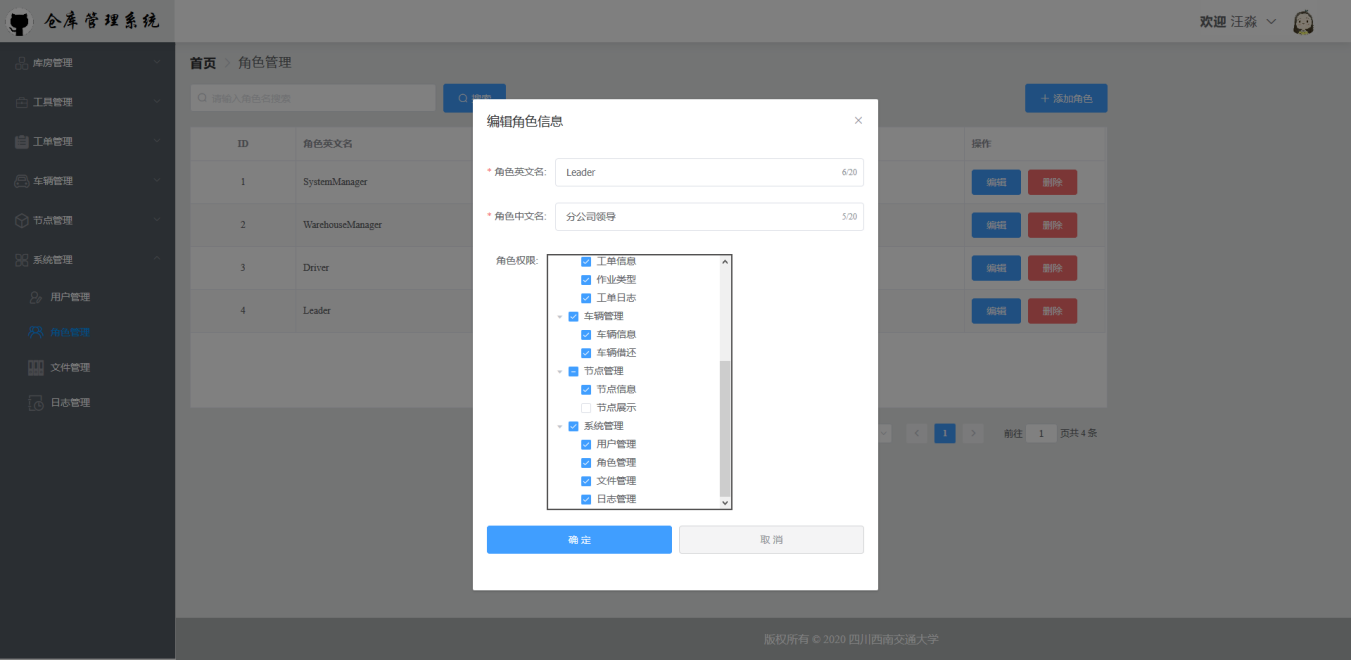
图5-13 头像修改页面

1. 角色管理

这里显示系统中的所有角色并分页显示，同样可以编辑角色信息。如图5-14所示。在编辑的时候，会有一个树形结构的控件，勾选或者取消即可修改对应的权限。如图5-15所示。

1. 日志管理

这里显示系统中用户的所有操作信息，即用户对系统做了什么事都会记录下来，具体会显示某人在什么时间做了什么事，如登录系统，修改个人信息，增删改系统功能模块的内容。如图5-16所示。

图5-14 用户管理页

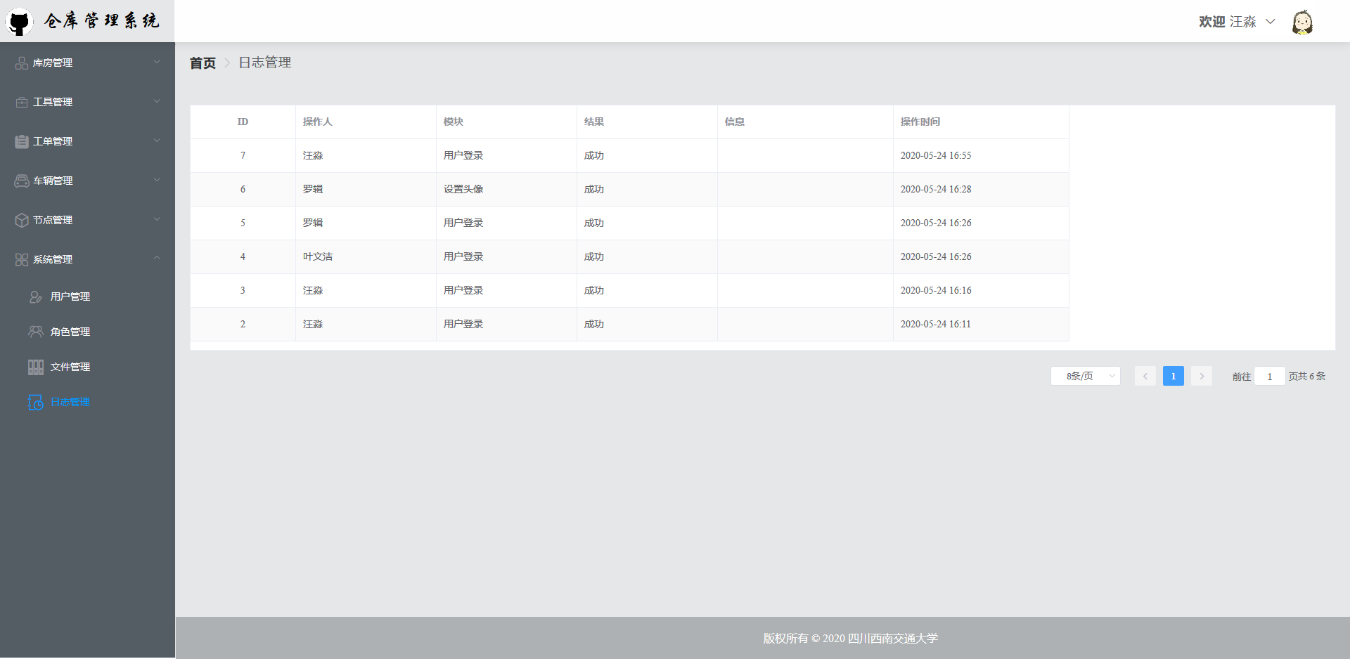
图5-15 角色管理页

图5-16 日志管理页

5.3 系统测试

5.3.1 测试任务及目标

软件系统在研发上线之前都需要进行测试工作，执行软件系统的标准化测试。在软件规模提升的过程中。随着软件系统的结构逐渐复杂，这使得软件开发出现错误的概率会不断的提升。

测试基于来自设计目标的输入信息，并验证系统的输出是否符合预期。 在不同的开发阶段，测试的目的是不同的。子模块的设计是检查子模块的功能是否实现，子模块的测试是否异常，以找出错误的原因。纠正错误后，测试将检查纠正是否正确。

通过系统功能测试，确保软件系统的操作结果符合预期。尽可能多的找出软件开发过程中的错误或者不足之处，如果在测试过程中，能够发现自己没有意识到的软件错误，系统的测试便是成功的测试。开发者结合系统测试的反馈，对系统的不足之处进行完善，再次进行测试，来确定软件系统是否可以上线。

5.3.2 系统功能测试

系统功能测试包括输入数据、文本、一系列操作步骤的测试。以下表格是对系统部分功能的测试，均与预期一致。

表5-1 管理员登录测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **测试描述** | **操作过程** | **预期结果** | **类型** |
| 管理员登录 | 登录系统，根据提示输入正确的用户名，密码和验证码 | 登录系统，在输入框中输入正确的用户名ye和对应的密码123正确验证码 | 系统提示登录成功，进入系统管理页面 | 判断数据是否正确 |
| 登录系统，按照提示输入不正确的用户名或错误密码或错误验证码 | 登录系统，在输入框中输入不正确的用户名，如xxx或者错误密码或者错误验证码 | 系统提示用户名或密码或验证码错误，登录失败 | 判断数据是否正确 |

表5-2 节点管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **测试描述** | **操作过程** | **预期结果** | **类型** |
| 节点  管理 | 进入节点管理页面，新增一个节点 | 进入节点管理页面，打开新增节点对话框，输入信息，点击确定 | 系统提示添加成功，并在表格中展示出来 | 功能点 |
| 进入节点管理页面，修改一个节点信息 | 进入节点管理页面，打开修改节点对话框，修改信息，点击确定 | 系统提示数据更新成功，并在表格中展示出来 | 功能点 |
| 进入节点管理页面，删除一个节点信息 | 进入节点管理页面，在一条数据后面点击删除按钮 | 系统提示数据删除成功，表格中没有此条数据 | 功能点 |

表5-3 库房管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **测试描述** | **操作过程** | **预期结果** | **类型** |
| 库房  管理 | 进入库房管理页面，新增一个库房 | 进入库房管理页面，打开新增库房对话框，输入信息，点击确定 | 系统提示添加成功，并在表格中展示出来 | 功能点 |
| 进入库房管理页面，修改一个库房信息 | 进入库房管理页面，打开修改库房对话框，修改信息，点击确定 | 系统提示数据更新成功，并在表格中展示出来 | 功能点 |
| 进入库房管理页面，删除一个库房信息 | 进入库房管理页面，在一条数据后面点击删除按钮 | 系统提示数据删除成功，表格中没有此条数据 | 功能点 |

表5-4 工单管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **测试描述** | **操作过程** | **预期结果** | **类型** |
| 工单  管理 | 进入工单管理页面，新增一个工单 | 进入工单管理页面，打开新建工单对话框，输入信息，点击确定 | 系统提示添加成功，并在表格中展示出来 | 功能点 |
| 进入工单管理页面，修改一个工单信息 | 进入工单管理页面，打开修改工单对话框，修改信息，点击确定 | 系统提示数据更新成功，并在表格中展示出来 | 功能点 |
| 进入工单管理页面，删除一个工单信息 | 进入工单管理页面，在一条数据后面点击删除按钮 | 系统提示数据删除成功，表格中没有此条数据 | 功能点 |

表5-5 工具管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **测试描述** | **操作过程** | **预期结果** | **类型** |
| 工具  管理 | 进入工具管理页面，新增一个工具 | 进入工具管理页面，打开新建工具对话框，输入信息，点击确定 | 系统提示添加成功，并在表格中展示出来 | 功能点 |
| 进入工具管理页面，修改一个工具信息 | 进入工具管理页面，打开修改工具对话框，修改信息，点击确定 | 系统提示数据更新成功，并在表格中展示出来 | 功能点 |
| 进入工具管理页面，删除一个工具信息 | 进入工具管理页面，在一条数据后面点击删除按钮 | 系统提示数据删除成功，表格中没有此条数据 | 功能点 |

表5-6 管理员信息修改用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **测试描述** | **操作过程** | **预期结果** | **类型** |
| 管理员信 息  设 置 | 进入管理页面，修改本人信息，如电话号码，用户签名，密码等 | 进入信息设置页面，修改电话号码为123 | 系统提示修改成功，并在个人信息展示页正确显示 | 功能点 |
| 进入头像修改页面，上传设置新头像 | 进入头像修改页面，点击上传框，从文件中选择一个图片 | 系统提示头像修改成功，并在管理页面正确显示 | 功能点 |

5.4 本章小结

本章对智能仓库管理信息系统的具体实现与测试部分根据各个模块依次简要阐述，并给出模块实现的功能和模块的运行截图，最后进行系统测试保证各个模块的功能正确无误，系统通过测试。

结 论

本课题从国家电网配网不停电作业管控系统的开发背景出发，结合自己对库房管理系统的理解，对整个系统做了需求分析、详细设计以及最终效果的呈现。智慧库房Web端前后台由工器具管理，库房管理，车辆管理，节点管理，工单管理以及用户权限管理等模块组成。

本文先对一般库房管理系统进行需求分析，确定了系统总体目标和一些必须的功能需求。之后在需求分析的基础上提出系统总体架构，确定详细的技术路线，分块设计模块功能。然后介绍系统的数据库设计，包括数据库概念设计和物理设计，还介绍了系统部分重要数据表。最后，使用IntelliJ IDEA作为系统开发平台，对管理系统进行了编码实现，通过功能页面的展示来说明系统实现过程，并且给出了核心模块的界面实现效果。系统测试是一个检测软件质量的过程，为了找到运行过程中不合理的因素，我们需要在一定规则约束下操作程序检验错误。设计测试用例，测试各个模块的功能，最终测试结果均符合预期。

由于本次毕业设计一些条件的限制，本系统功能还存在需要扩展、完善的地方，需要进一步研究与设计：

（1）在工作量上，因为时间问题以及个人做事效率问题，可能显得有点少，因此系统待添加模块，升级系统。

（2）系统的性能部分可以提升，如使用Redis缓存等。

致 谢

光阴似箭，日月如梭，异地求学的四年大学时光转瞬即逝。经过四个多月的研究学习与设计，本次毕业设计也顺利走向尾声。本次毕业设计的顺利完成，离不开亲人，老师以及同学对我的支持和帮助。

感谢我的指导老师陈剑波老师，他在本次毕业设计的选题、问题解决、技术方案、文档编写等方面上给了我很大帮助。感谢他在整个毕业设计过程中对于论文与报告的指导与帮助。

感谢母校提供优质的学习环境、生活环境，我们在这里可以享受学习、享受生活。尤其此次学校为做毕业设计的同学提供VPN服务，方便毕业班的同学查阅文献资料。希望母校可以发展的越来越好。

感谢我的父母和亲人，他们对我的大学生活提供了强有力的物质生活保障，让我能够无忧无虑的完成我的学业。

感谢所有教育我的老师们，教授了我理论知识，提高我的文化知识水平，希望所有老师工作顺利，身体健康。

感谢网络上的博主们以及开源框架的开发设计者们，是你们的工作为后来者提供了很大的便利，希望你们工作顺利。

最后，再次感谢所有对本次毕业设计提供帮助的人们，祝愿大家幸福安康！

参考文献

1. 罗立勤，焦曙光. 医院药品管理系统的开发与应用[J]. 宜昌：医药导报，2004：130.
2. 刘现明. 部队仓库管理系统的设计与实现[D]. 山东大学, 2013.
3. 潘恒标. 广西财经学院后勤仓库管理系统的设计与实现[D]. 电子科技大学, 2013.
4. 张莉英. 百货公司仓库管理系统的设计与实现[D]. 电子科技大学, 2013.
5. 张天一. 仓库管理系统的设计与实现[D]. 电子科技大学, 2013.
6. 杨雪菲. 基于B/S结构的仓储管理系统的设计与实现[D]. 天津大学, 2018.
7. 张涛, 王海鹏, 胡正国. 基于UML用例图的软件产品线需求建模方法[D]. 2004.
8. Warmer J B, Kleppe A G. The object constraint language: Precise modeling with uml (addison-wesley object technology series)[J]. 1998.
9. Larman C. Applying UML and patterns: an introduction to object oriented analysis and design and interative development[M]. Pearson Education India, 2012.
10. 孙娉娉, 李新, 史广军. 基于前后端分离的内容管理系统[J]. 科研信息化技术与 应用, 2016, 4.
11. 张培妍. 基于Vue.js的智能匹配系统的设计与实现[D]. 西安电子科技大学, 2019.
12. 肖平, 基于Spring Boot的出版社信息管理系统设计与实现[D]. 硕士论文. 华中 科技大学, 2009.
13. 于子贻, 基于Spring MVC的教务信息管理系统设计与实现[D]. 硕士论文. 电子 科技大学, 2013.
14. 陈涛. 基于工作流的监狱事务审批系统的分析与实现[D]. 中山大学, 2014.
15. 朱少敏. 基于工作流技术的政府电子采购计划管理系统的研究与实现[D]. 暨南 大学, 2006.
16. 朱利民, 岑俊杰. 基于ASPNET技术的新闻发布系统的设计与实现[D]. 河南机电 高等专科学校学报, 2011.
17. 王波. 基于BS模式的网络在线考试系统的设计与实现[D]. 电子科技大学, 2008.
18. 胡丽娜. 基于J2EE的增值业务自动化管理平台的设计与实现[D]. 北京邮电大学, 2007.
19. 王伟丽. 智慧社区电子商务系统的设计与实现[D]. 北京邮电大学, 2018.
20. 熊凯. 基于Spring Cloud的培训商务系统的设计与实现[D]. 北京邮电大学, 2019.