

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： 李松

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：李松

指导教师签名：陈剑明

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

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

指导教师 陈剑时 (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

（1）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 and 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： 陈剑时 2019 年 11 月 22 日

审 批 人： \_\_\_\_\_ 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，大大减少了以往出入存流程繁琐，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
1.1 背景与意义.....	1
1.2 国内外发展（应用）现状.....	1
1.3 论文所做工作及思路.....	3
1.4 论文章节安排.....	3
第 2 章 系统基础理论与技术介绍.....	4
2.1Vue.js 前端框架简介.....	4
2.2MVC 设计模式.....	5
2.3Spring 与 Spring Boot 技术介绍.....	6
2.3.1Spring 框架简介.....	6
2.3.2Spring Boot 框架简介.....	7
2.4MyBatis 持久层框架介绍.....	6
2.5 本章小结.....	6
第 3 章 系统需求分析.....	9
3.1 系统需求描述.....	9
3.1.1 总体目标.....	9
3.1.2 可行性分析.....	9
3.2 系统功能需求.....	10
3.2.1 系统管理模块需求.....	10
3.2.2 系统功能需求.....	10
3.3 本章小结.....	12
第 4 章 系统详细设计.....	13
4.1 系统架构设计.....	13
4.1.1 系统整体架构.....	13
4.1.2 系统逻辑架构.....	13
4.2 系统功能模块设计.....	14
4.2.1 库房管理模块设计.....	15
4.2.2 工器具管理模块设计.....	15
4.2.3 车辆管理模块设计.....	15
4.2.4 工单管理模块设计.....	15

4.2.5 WebSocket 通信模块设计.....	15
4.2.6 节点管理模块设计.....	15
4.3 系统管理模块设计.....	14
4.3.1 用户管理功能设计.....	15
4.3.2 角色管理功能设计.....	15
4.4 系统数据库设计.....	14
4.4.1 数据库概念设计.....	16
4.4.2 数据库物理设计.....	18
4.5 本章小结.....	22
<b>第 5 章 系统实现与测试.....</b>	<b>27</b>
5.1 系统开发环境.....	27
5.1.1 硬件环境.....	27
5.1.2 软件环境.....	27
5.2 系统模块功能实现.....	27
5.2.1 功能模块实现.....	27
5.2.2 系统管理模块实现.....	30
5.3 系统测试.....	29
5.3.1 测试任务及目标.....	27
5.3.2 系统功能测试.....	30
5.4 本章小结.....	35
<b>结 论.....</b>	<b>38</b>
<b>致 谢.....</b>	<b>39</b>
<b>参考文献.....</b>	<b>40</b>

## 第 1 章 绪 论

### 1.1 背景与意义

自二十世纪以来,计算机和信息化发展速度一日千里,信息技术已经渗透到社会各行各业以及人们的日常生活中,同时,信息化技术的迅猛发展也为社会的发展提供了重要的推动力,提高了工作效率的同时解放了生产力。同时,信息网络技术的发展使得经济全球化得以实现,对全球化的经济进展提供了巨大的动力。计算机和网络的信息技术具有公开性、同步性、创新性以及高效性等优点。正是这些优势,支配着全球经济的发展,同时影响了世界经济格局<sup>[1]</sup>。

仓库管理工作是一项既繁琐又复杂的工作<sup>[2]</sup>,管理人员每天都要处理大量的数据,各种数据报表都要及时的生成和更新,包括货物的出入时间、出入量、进出库房的人员管理、库房维修、检查管理、库房环境信息等数据。随着企事业单位的发展需求,库房数量在不断增加,货物种类亦是越发繁杂,库房数据信息量在爆发式的增长,库房作业则越来越复杂,越来越多样化,在这样的情况下,传统的,靠人工抄录、核对信息的库房管理方法已经没有办法保证让企业高效地利用各种资源了,不仅效率低下、查询麻烦、保密性差、更新维护困难,还会导致出错率高,给企业带来重大的损失<sup>[3]</sup>。

库房管理的内容与企事业单位的工作息息相关,甚至会直接影响到单位的决策者制定战略与决策,是企事业单位不可或缺的一部分。库房信息的爆炸式增长,使得数字式的仓库管理系统显得尤为重要。基于数据库技术的仓库信息管理系统具有记录简单方便,查询快捷可靠,存储量大,信息安全,成本低等诸多优势<sup>[4]</sup>,成为仓库管理工作的最佳工具。不同的企事业单位,其库房作业内容不尽相同,仓库管理系统主要提供一个仓库业务及其作业管理的信息存储和检索工作。

### 1.2 国内外发展（应用）现状

传统的仓库管理方法使用人工记录的方式来记录跟踪进出货物及其他仓库数据,不仅效率低,维护和更新困难,而且极易出错<sup>[5]</sup>。

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。海尔在这方面做出了良好的表率。

海尔 1984 年创立，也曾经过国内很多企业经过的仓储阶段，随着发展外租仓库，这些仓库也存在管理落后、效率低下、损耗严重等问题。但经过长久的分析后，海尔集团发现这些仓储问题严重影响着海尔的发展，所以选择了仓储作为突破口，以空调事业部作为试点，对物流进行相应的改革。1998 年，张瑞敏提出，将立体仓库建立在海尔园内建。在这之前，海尔工厂并未设立单独的仓库，而是工厂和仓库共用，在生产规模逐渐扩大的形势下，需要不断增加生产线，使得能够存放产品的面积逐渐缩小，不得不采取外租仓库的形式存放产品，这样一来，大大的增加了其成本。当海尔制定建立仓库计划时，张瑞敏提出要以现代物流为基础，建设立体仓库，坚决不能使用传统仓库。因此位于青岛的海尔开发区工业园于 2000 年 5 月建成了一座全自动智能化的国际物流中心，占地面积为 1.92 万平方米，同时可以容纳 1.8 万多个货位，能够存放开发区内所有冰箱、小家电、商用空调、冷柜以及电热器等的原材料以及成品。在功能方面，该中心可以连接海尔集团的 ERP 系统，能够较好的适应电子商务的需求，实现与商流、物流、信息流、资金流之间的数据传输。

我们从海尔的发展历程可以看出，该集团摒弃传统仓储物流，建立具有先进水平的现代化物流，搭建了相应的物流信息系统平台，发展成统一的集团物流，从而可以完成仓储自动化、机械化、仓储智能化、仓储信息网络化，大大提高了仓储利用率和工作效率、降低了成本和伤害。但相对发达国家的尖端科技我国显现的有些逊色，还

有他们超前的思想也是值得我们去学习借鉴的<sup>[6]</sup>。

### 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，通过阅读国内外相关文献，分析研究库房管理系统现状，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，提出库房管理系统总体业务流程设计，确定系统开发技术和路线，设计各个模块的功能并设计数据库模型方便后续开发。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

论文共分五章。

第1章 绪论主要介绍了系统的开发背景与开发意义，以及国内外的研究现状，说明了库房管理系统需要解决的主要问题和本文的主要工作及组织结构。

第2章 系统基础理论与技术介绍，概述了与系统开发相关的开发技术与开发方法，并对部分重要理论做出详细介绍。

第3章 系统需求分析。主要围绕着需求统一化过程的几个关键步骤，描述了需求分析阶段的主要工作。首先，对系统需求进行了概述。然后，对于该系统的功能需求进行描述。

第4章 系统详细设计。通过对系统的技术构架、功能构架的分析与设计，确定了系统的整体架构和核心功能。然后，对于系统的整体结构和数据库结构进行了详细的设计。

第5章 系统的实现与测试。该章节是需求分析和系统设计的后续环节，在本章中将重点描述主要功能模块实现的情况，并给出部分模块的运行效果图，并简要介绍系统测试。

## 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<sup>[7]</sup>（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<sup>[8]</sup>。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<sup>[9]</sup>，自动完成了数据和视图的同步更新，并且与其他前端框架相比，Vue.js 非常简单易上手，没有太多新概念，学习成本较低，在对实例 `new Vue({data:data})` 进行声明后 data 中的数据自动与视图进行绑定，一旦 data 中的数据发生变化，视图中相应数据会随之更改。

Vue.js 的“一切皆为组件”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。

除此之外，Vue.js 可以与其他工具搭配使用，例如 `vue-resource` 和 `vue-router`，对路由和异步请求提供支持，这样就可以完成单页面的开发工作。Vue.js 以易上手、轻量级的优点为人称道，MVVM 的开发模式为前端人员免去了原先冗余繁重的 DOM 操作，代码更加易于维护，Vue.js 2.0 版本已经发布，借助轻量级的 virtual-DOM 实现页面渲染<sup>[10]</sup>，使得在大部分使用场景中初始化页面渲染速度可以提升 2~4 倍。作为非常轻量级的前端框架，在前后端分离的项目中能够最大程度上发挥 Vue.js 的优势和特性，本项目既有移动端也有 PC 端，Vue.js 在移动浏览器上压缩后只有 18KB 且没有其他



的依赖，相较于擅长构建 APP 的 AngularJS 和需要本地开发人员的 ReactJS，Vue.js 结合了它们的优势—灵活、简单和轻量，更适合 Web 开发，所以选择 Vue.js 最为合适 [11]。

## 2.2 MVC 设计模式

MVC 全称是 Model View Controller，即模型视图控制器，是模型(model)-视图(view)-控制器(controller)的缩写，是一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑[12]。

下图为 MVC 设计模式，如图 2-2 所示。

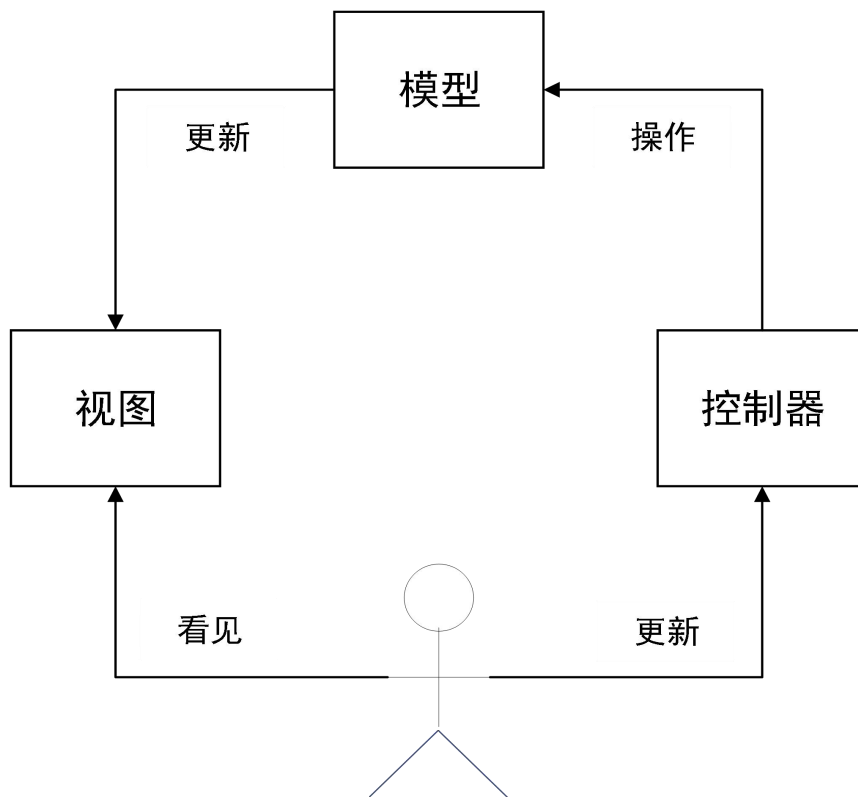


图 2-2 MVC 设计模式

**Model（模型）**是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。

**View（视图）**是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。

**Controller（控制器）**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读取数据，控制用户输入，并向模型发送数据。

MVC 设计模式独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑

的图形化用户界面的结构中。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。

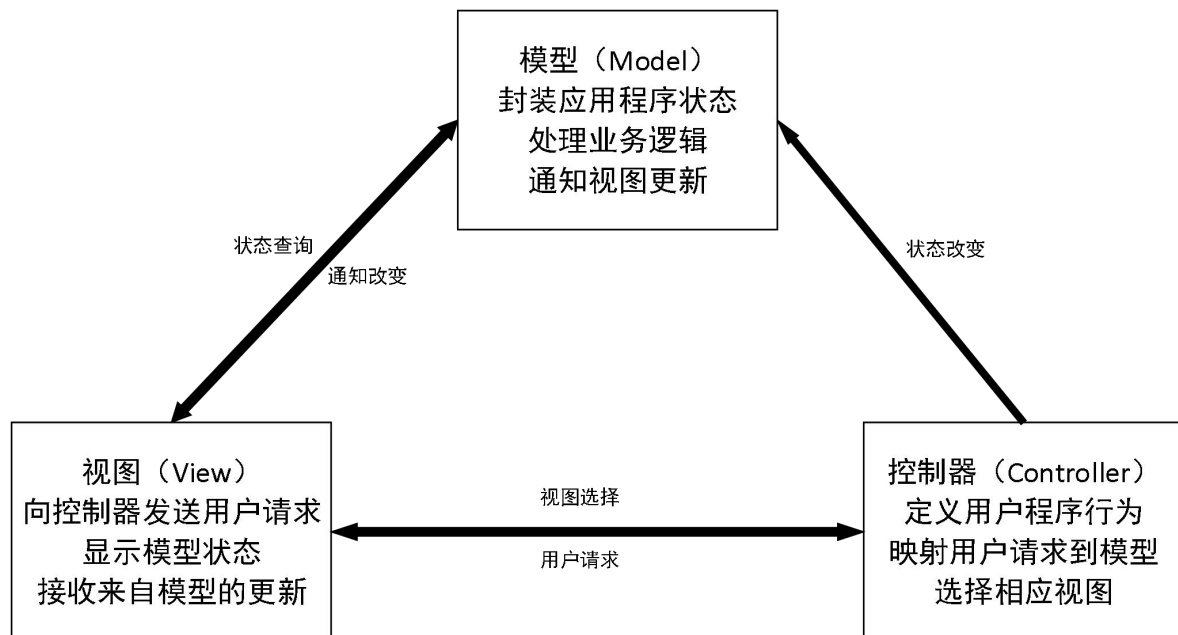


图 2-3 MVC 工作模式

MVC 可对程序的后期维护和扩展提供了方便，并且使程序某些部分的重用提供了方便。而且 MVC 也使程序简化，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<sup>[13]</sup>。

## 2.3 Spring 与 Spring Boot 技术介绍

### 2.3.1 Spring 框架简介

Spring 框架是由 Rod Johnson 首次编写并发表的一个开源框架，应用于构建 web 应用。是 JAVA EE 平台上的一种流行框架。由 2003 年发布的 1.0 版本至今已经发布到 5.0。其大致组成如图 2-4 所示：

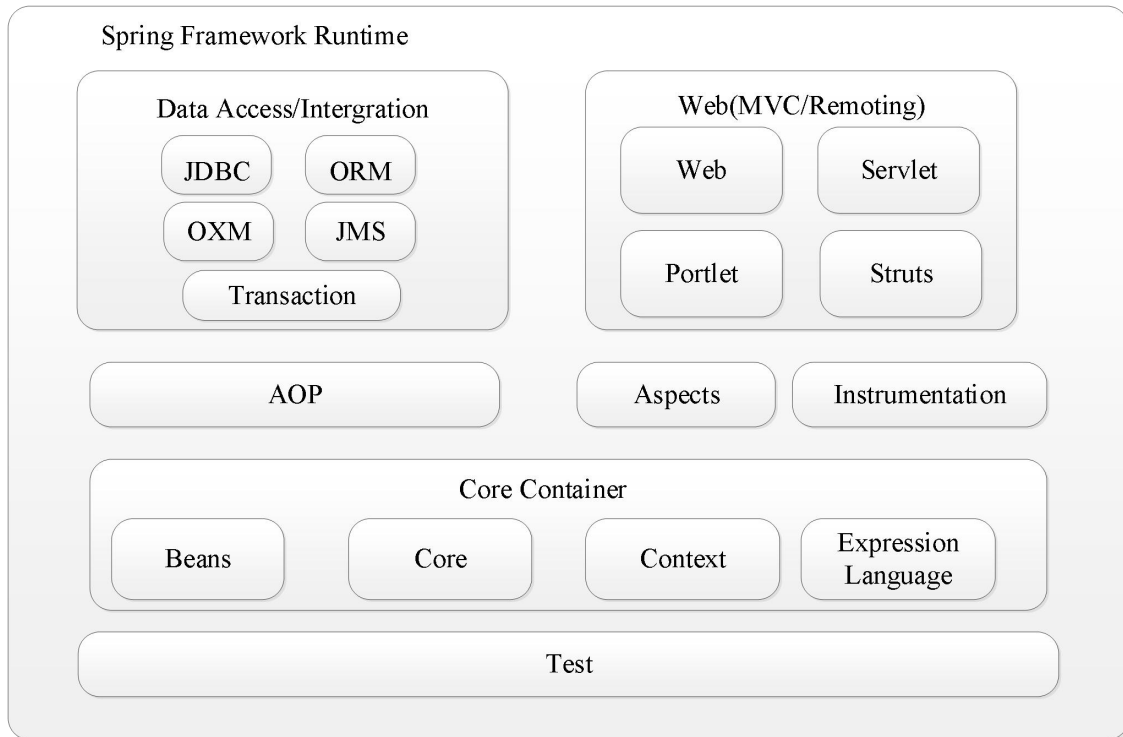


图 2-4 Spring 框架

Spring 框架的核心在于面向切面编程<sup>[14]</sup>（AOP）思想与通过依赖注入<sup>[15]</sup>（dependency injection）降低耦合性。辅以对于数据库的支持和 Web 相关技术，使得一个 Web 应用程序从开发到使用到维护的难度降低。

### 2.3.2 Spring Boot 框架简介

为了解决在 Spring 框架中出现的大量 xml 配置文件，书写困难，调试困难。以及开发环境搭建时，各个框架直接的各个版本的兼容性问题。基于上述原因 Pivotal 团队提供了一种全新的开源框架 Spring Boot，其目的是简化使用 Spring 开发项目。

Spring Boot 的核心思想为开箱即用<sup>[16]</sup>（Out of box），通过“约定优先配置<sup>[17]</sup>（Convention over configuration）”的一种思想。提前在项目中提供一些缺省配置（AutoConfiguration），当用户需要改变配置时，通过配置文件声明需要改变的内容其余内容使用缺省配置。同时使用 properties 或 yaml 文件保存配置信息，在读写的便利性上也优于使用 xml 文件保存配置信息。

本次采用的 Maven 项目<sup>[18]</sup>与 Spring Boot 框架开发项目，使得项目对于环境的依赖与管理更加容易。

## 2.4 MyBatis 持久层框架介绍

MyBatis 是当前的主流的持久层框架，是一种 ORM（Object/Rational Mapping）框架<sup>[19]</sup>。与 Hibernate 框架相比，MyBatis 可以配置动态 SQL 并优化 SQL，可以通过配置决定 SQL 规则。对于一些复杂的和需要性能优化的项目，使用 MyBatis 无疑是更好的选择。MyBatis 入门非常简单，其主要设计目的就是使数据持久层的开发变得更加方便<sup>[20]</sup>。其执行流程如图 2-5 所示：

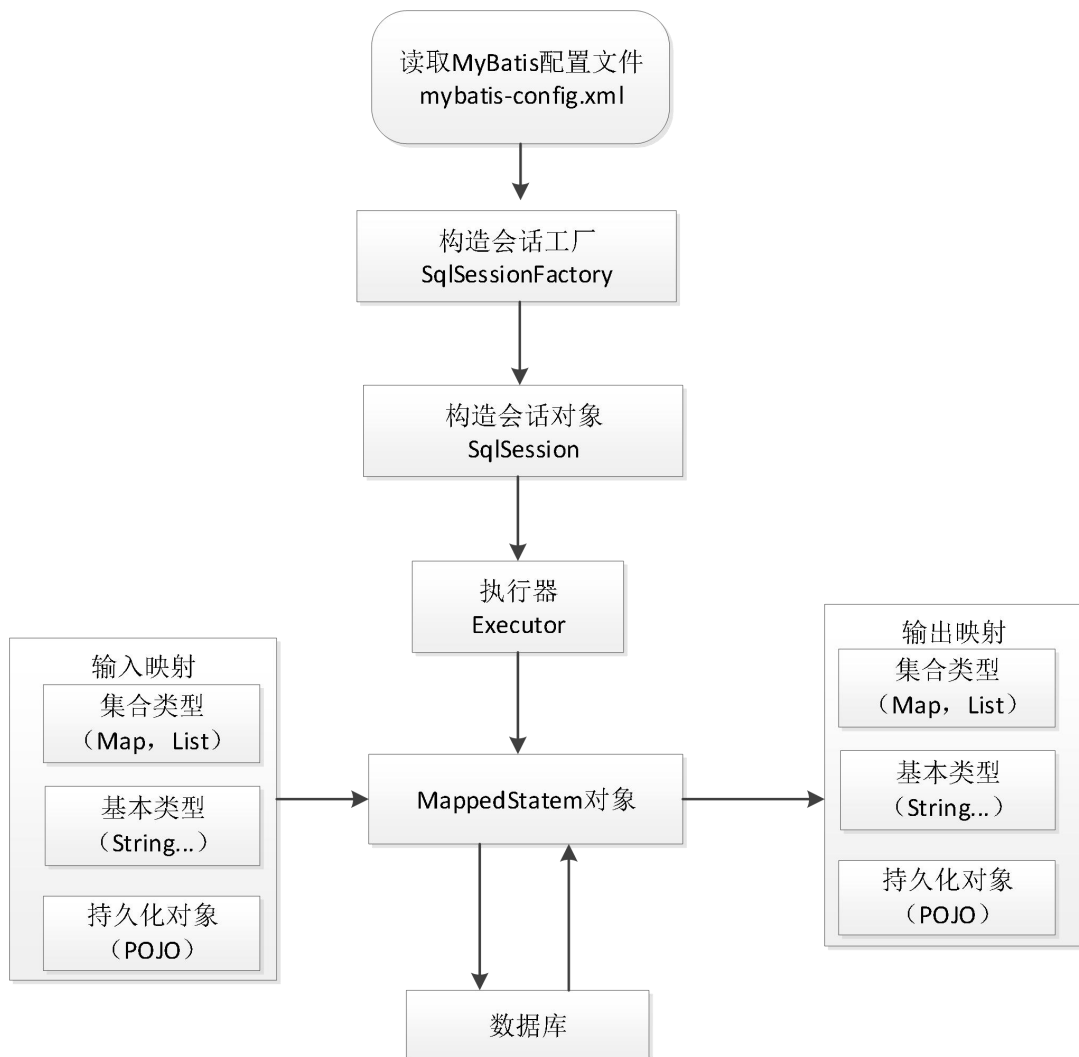


图 2-5 MyBatis 执行流程图

## 2.4 本章小结

本章确定了开发环境所需要的框架与技术，并对这些内容进行了简要的介绍。为系统的设计提供了技术方面的支撑。了解这些框架与技术，对于在系统的分析设计与开发过程中有着重大的意义。

## 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。

为使开发的新系统更加符合用户的预期，在设计开发前要与管理者、操作者多次交流，以确定对系统的确切需求。客户对系统功能上的要求是指系统可以做什么，对性能上的要求是指系统相关功能做的怎么样。只有完全理解了客户的需求才可以着手设计开发，否则可能背道而驰，最终以失败告终。

#### 3.1.2 可行性分析

一个系统的开发需要涉及到很多种技术，这些技术是否成熟稳定是系统开发成败的关键。所涉及的技术是否有知识产权保护，如何取得使用权，开发团队是否有能力运用这些技术。仓库管理系统的开发主要使用 Vue.js 作为前端框架，JAVA 语言编写后台处理逻辑，使用 MYSQL 数据库管理系统管理数据。目前 B/S 架构的应用系统技术非常成熟，使用 JAVA 语言开发 B/S 架构系统成为主流。IntelliJ IDEA 是一个集成的开发环境，内部集成了多种工具 WEB 服务器，支持多种编程语言。基于 Vue.js 的 B/S 架构系统技术已经很成熟，现在主流系统开发都在使用，会此技术的开发人员很多，可供查阅的资料和案例也很丰富。界面操作简单，易学易用，面向对象的开发技术，封装了很多常用的组件，提高了开发效率。

以上开发工具本人都使用过，即便遇到一些问题通过向老师请教和查阅资料都可以解决，所以在技术上是可以实现。

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。

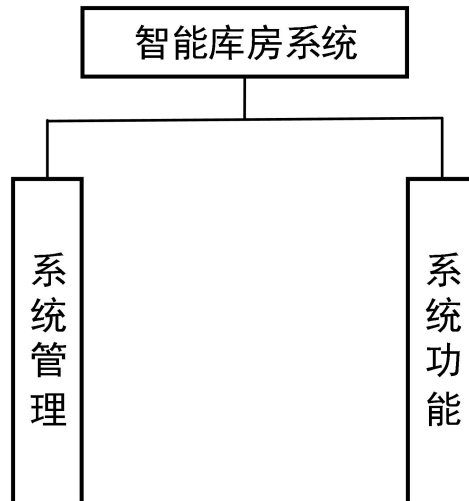


图 3-1 智能库房管理系统总体功能

### 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。

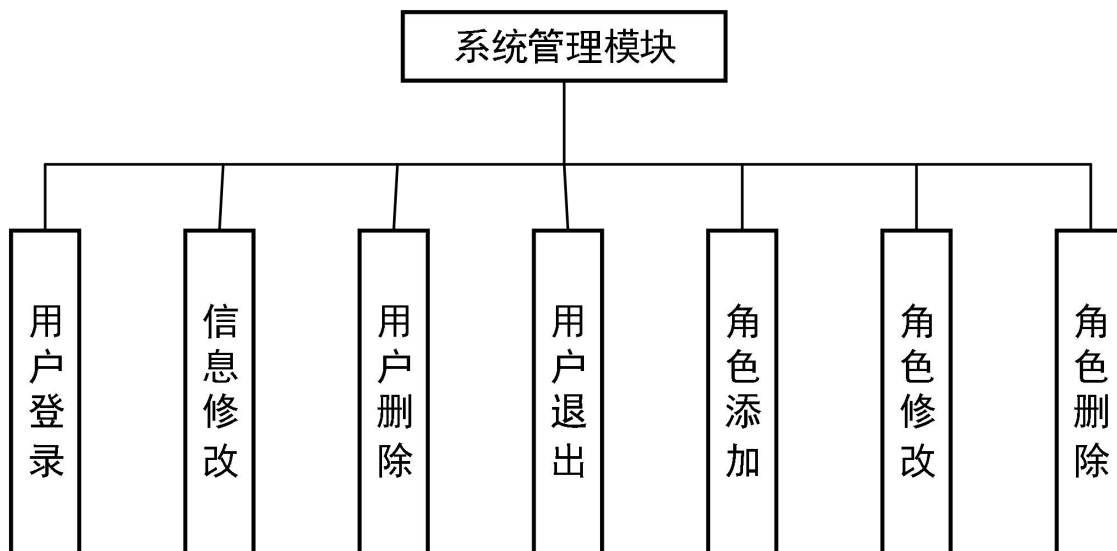


图 3-2 系统管理模块

#### （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。

#### （2）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。

#### （3）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。

#### （4）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。

#### （5）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。

#### （6）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。

#### （7）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。

### 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。

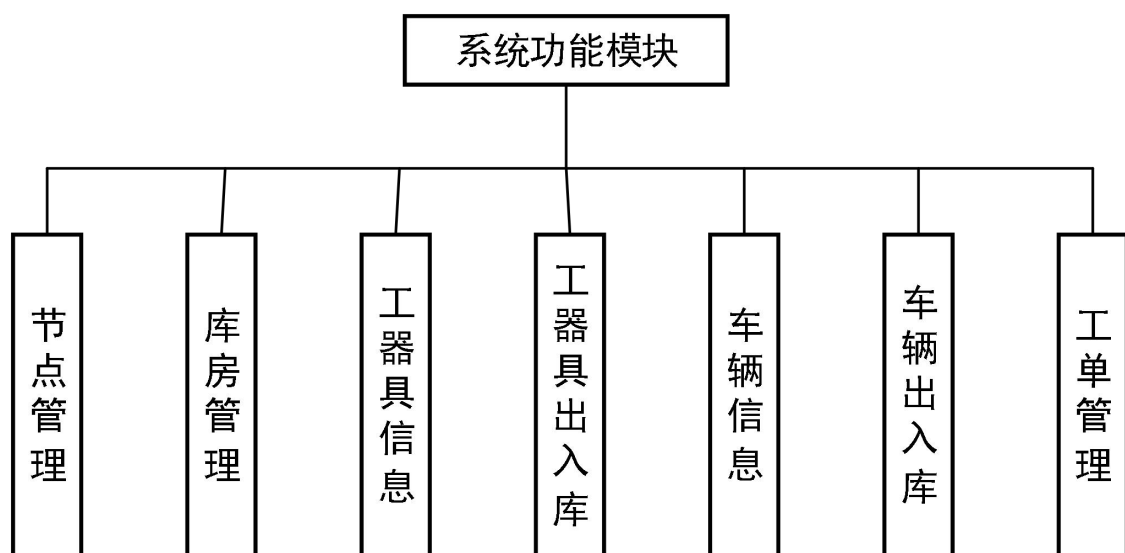


图 3-3 系统功能模块

#### （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。

#### （2）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。

#### （3）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。

#### （4）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。

#### （5）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。

#### （6）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。

#### （7）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。



## 第 4 章 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，此章节将阐述仓库管理系统的设计要求，对于系统的详细设计做出介绍，并重点介绍系统主要功能模块的设计情况。最后，详细阐述系统数据库系统的设计情况。

### 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。

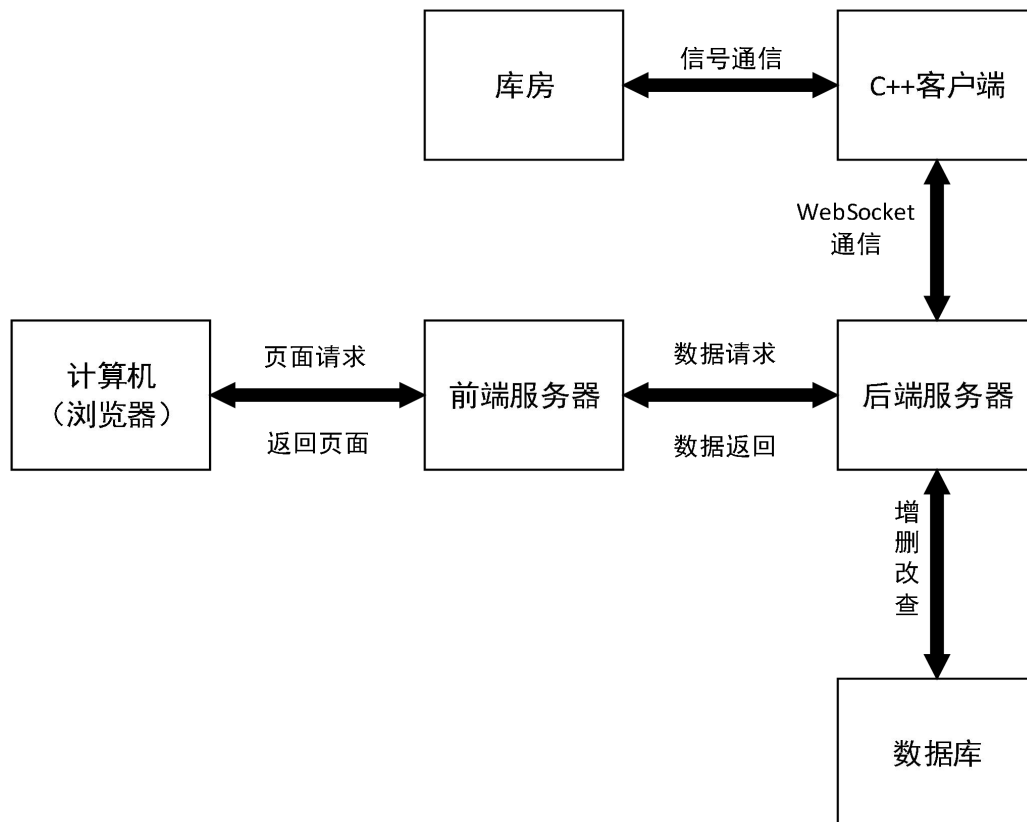


图 4-1 系统架构图

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虚线框中的内容。

#### 4.1.2 系统逻辑架构

库房管理系统的逻辑架构包括控制层，业务逻辑层，数据访问层和视图层。

控制层：对系统功能模块流程的控制。它的具体功能包括了，解析 HTTP 请求，调用相应的业务逻辑层，实现具体的功能，以及返回系统所相应的数据。

业务逻辑层：实现具体功能模块的逻辑应用设计，具体功能是实现系统功能模块的业务逻辑。业务逻辑层又具体分为业务逻辑接口和业务逻辑实现。业务逻辑接口用于定义需要的业务逻辑操作。业务逻辑实现用于实现业务逻辑接口所定义的业务逻辑操作。

数据访问层：负责数据持久化的相关功能。具体功能是连接后台数据库和对数据实现增删改查的功能。本文的系统使用了开源框架 MyBatis 框架实现数据访问层，MyBatis 框架是一种支持定制化 SQL，存储过程和数据映射模型的数据持久化框架，它避免传统的 JDBC 代码和手动设置参数的麻烦。

视图层：负责前端界面的显示工作。系统通过视图层收集用户的请求信息。

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。

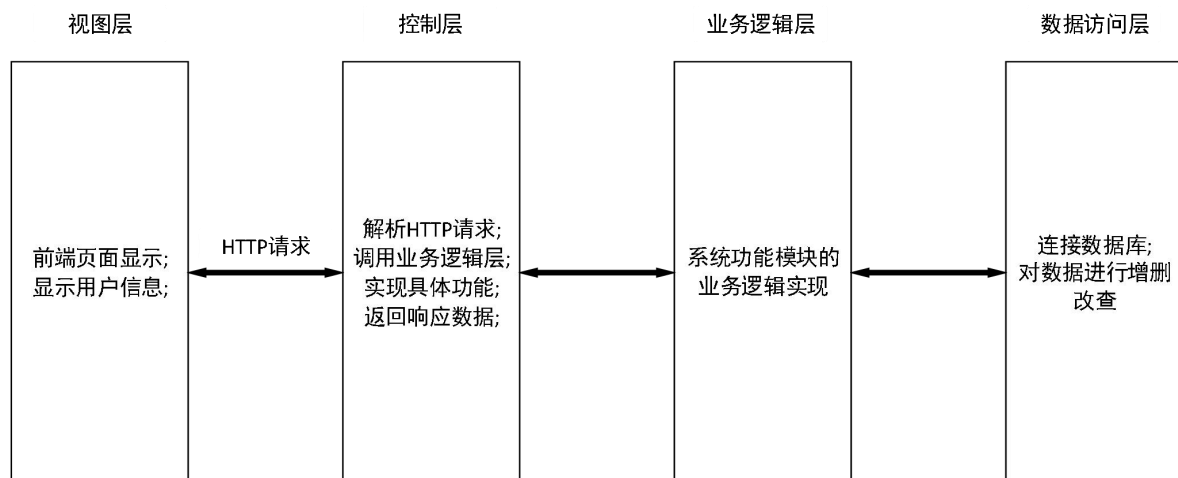


图 4-2 逻辑架构图

## 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。流程图如 4-3 所示。

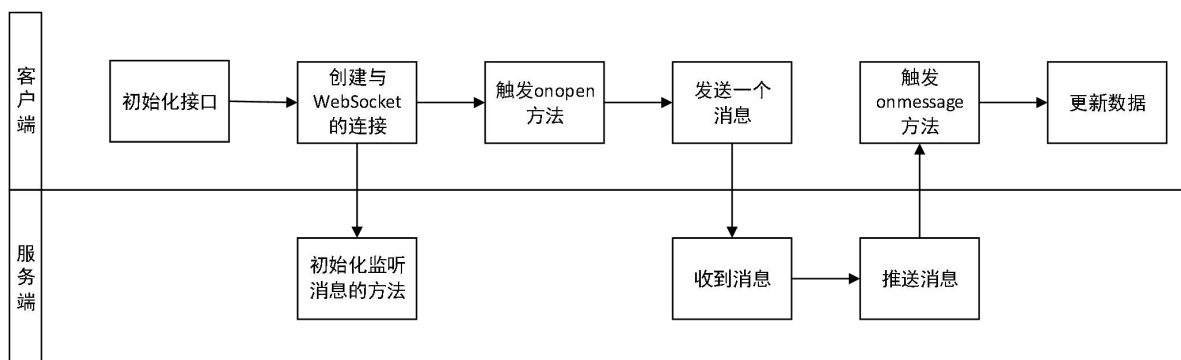


图 4-3 通信流程图

#### 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。节点管理要完成节点信息的查询，编辑，添加和删除。

### 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。流程图如 4-4 所示。

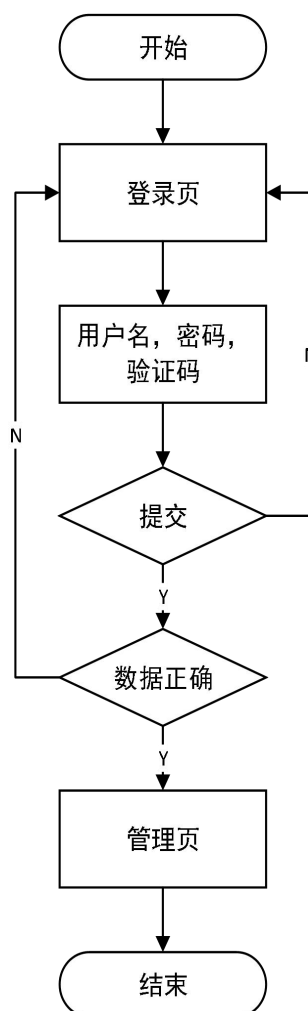


图 4-4 用户登录流程图

### （2）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。密码修改流程图如 4-5 所示

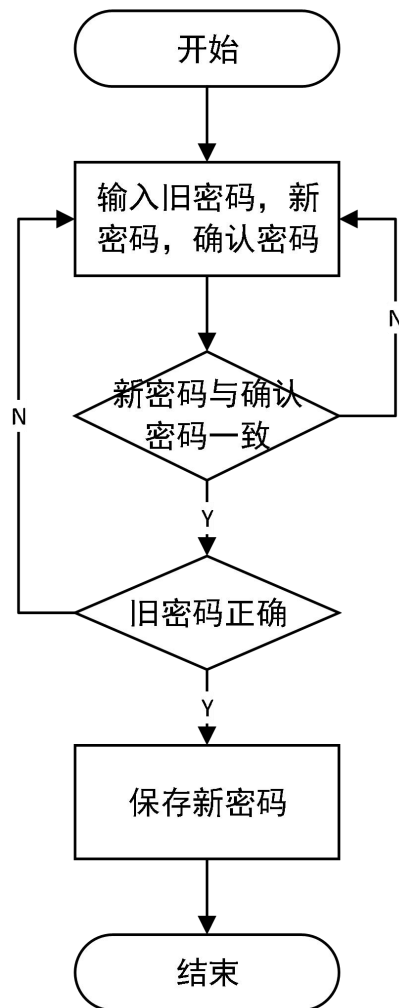


图 4-5 密码修改流程图

### （3）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。

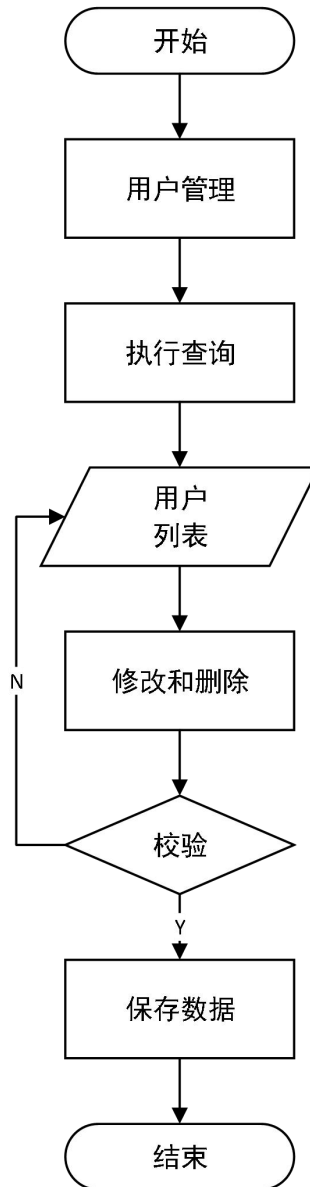


图 4-6 用户管理过程流程图

### 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进行权限的增删。流程图如 4-7 所示。

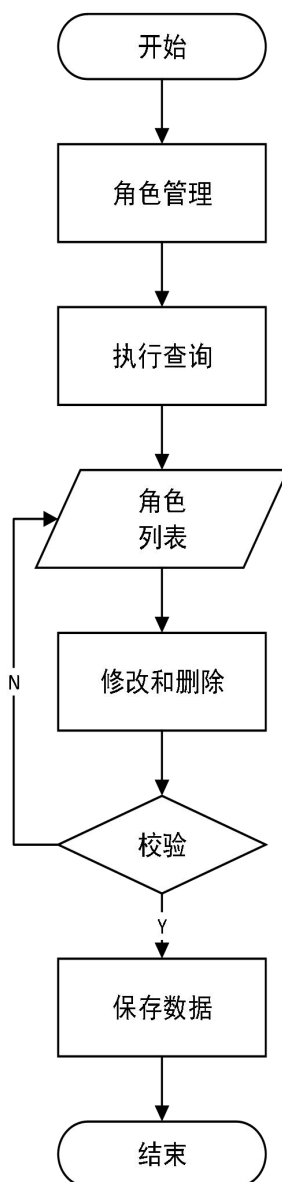


图 4-7 角色管理过程流程图

## 4.4 系统数据库设计

### 4.4.1 数据库概念设计

在多种类型的数据库中只有关系型数据库最适合管理信息系统，在管理信息系统中的事物都可以抽象成具有一定格式的记录，事物之间都有一定联系。关系型数据库中通过记录的主键来标识唯一一条记录，一个关系通过外键与另一个关系记录保持了联系。利用索引机制可以快速查询需要的信息。

在明确本系统的各项架构后，可以开始针对整个系统进行数据库建模。首先采用 E-R 模型的方式创建实体关系图，然后由实体关系图动态的生成对应的物理表结构。

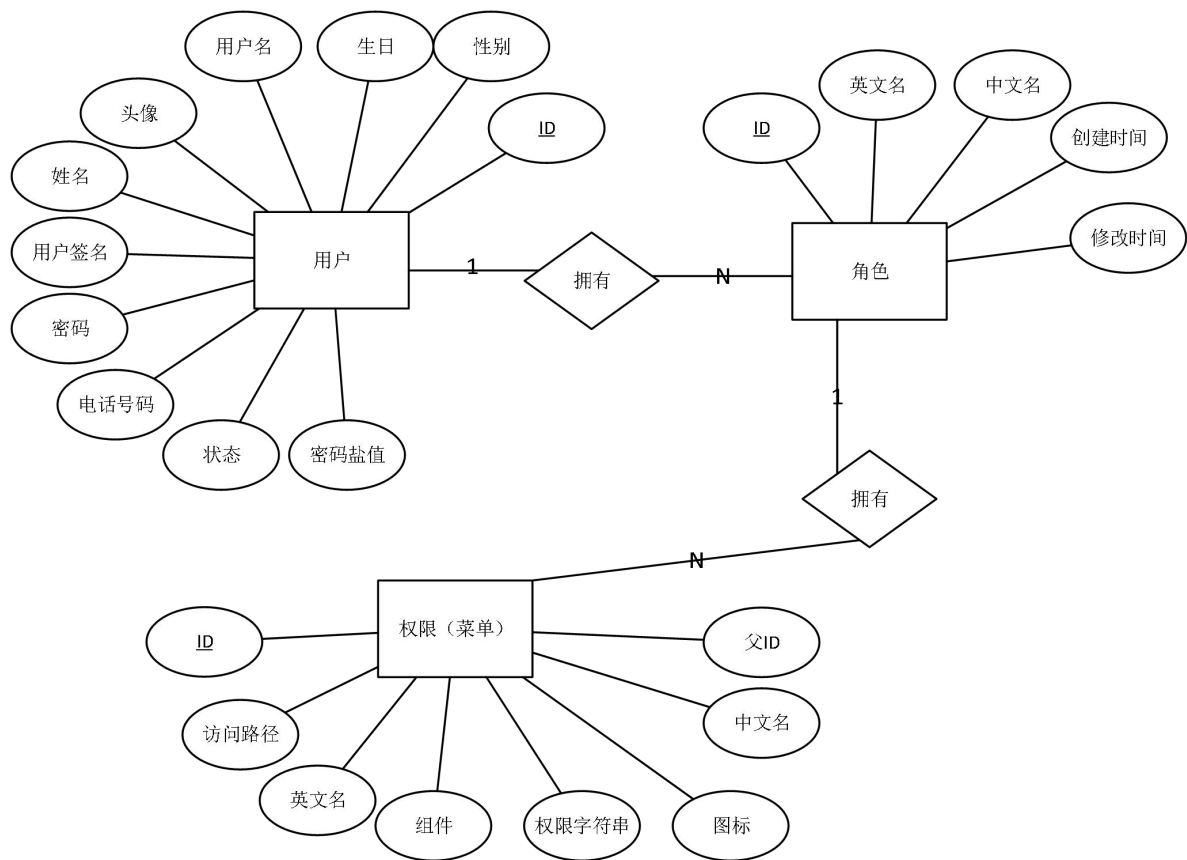


图 4-8 用户角色权限 E-R 图



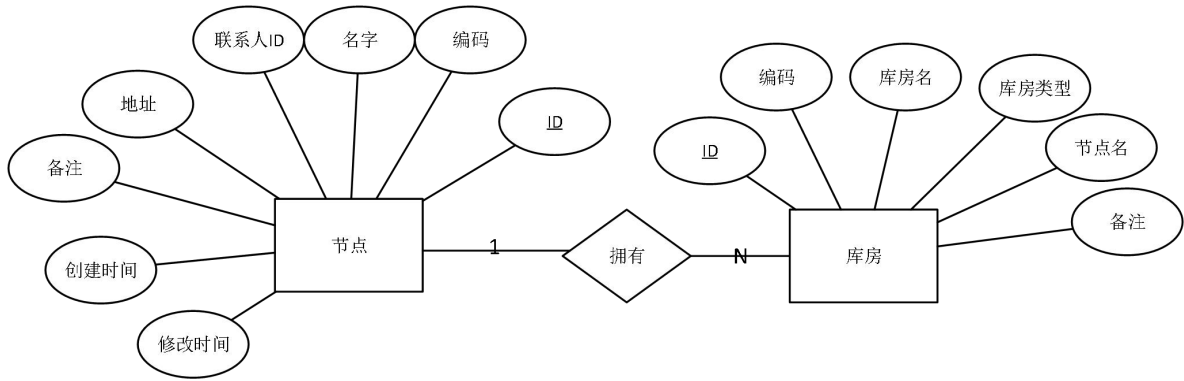


图 4-9 节点库房 E-R 图

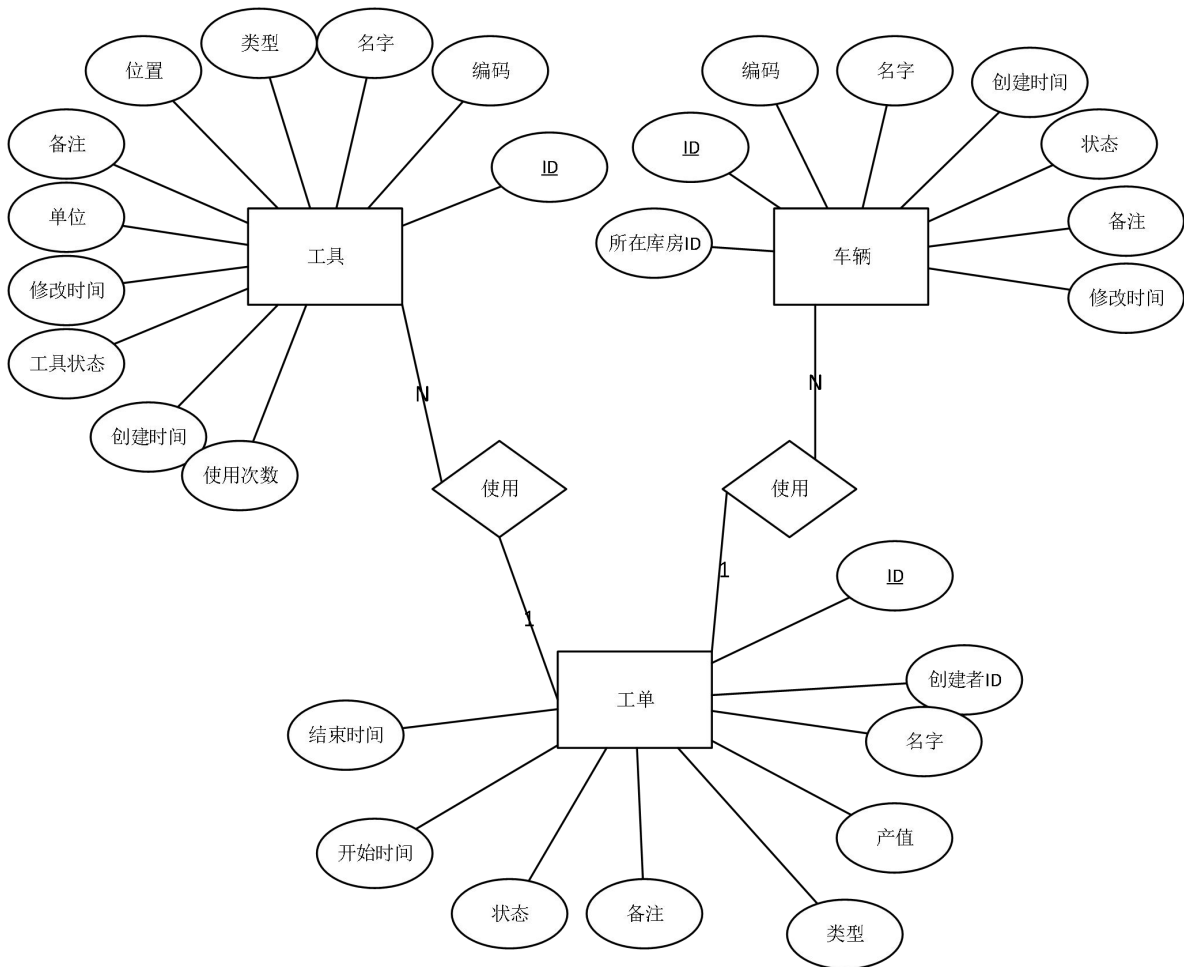


图 4-10 工单工具车辆 E-R 图

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

数据库的物理设计就是设计数据库中的表和表内各字段的属性。一个成功的管理信息系统一定是一个设计科学合理的数据库为基础，管理系统就是以对数据信息的管理为核心。数据库的物理设计决定应用系统设计成败。本系统采用了 MySQL 管理系统，它使用图形化的管理界面，操作简单方便，功能实用。非常适合中小型应用系统的开发。

库房管理系统在 MySQL 中创建数据库，数据库名为 graduation\_project。主要包含如下几个表。

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
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

本章对于智能库房管理系统的详细设计进行了阐述，并且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。

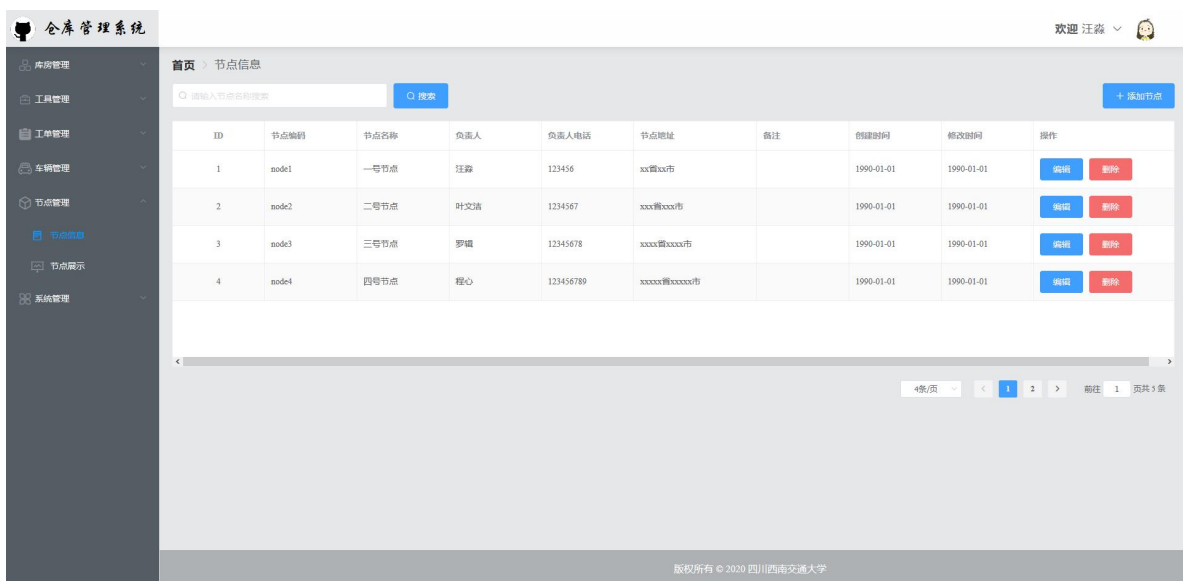


图 5-1 节点管理页

### （2）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。

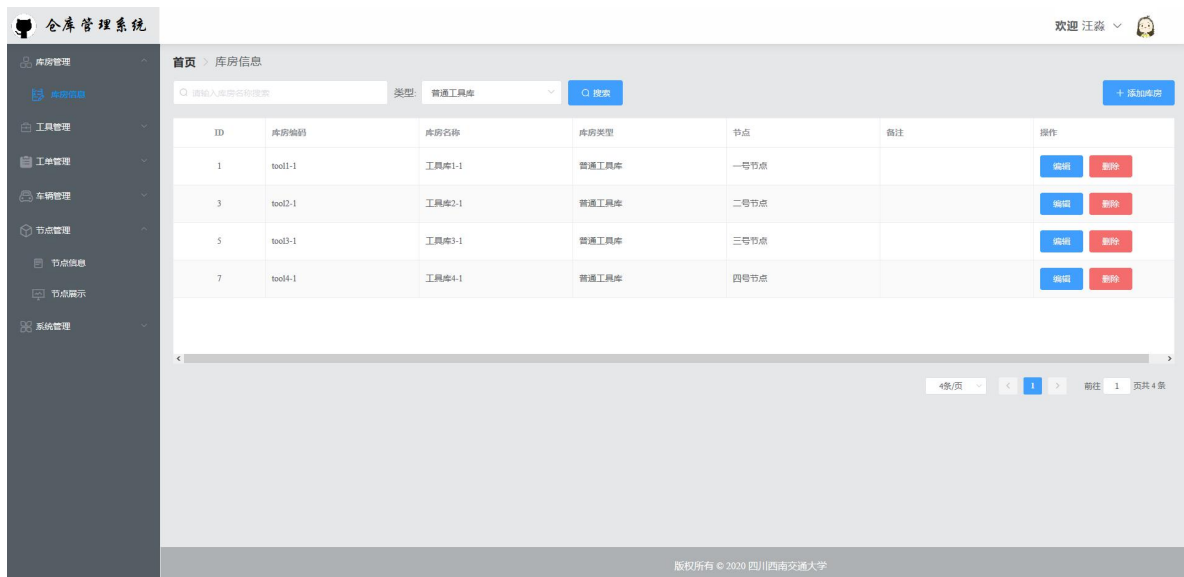


图 5-2 库房管理页

### （3）工具管理

这里展示所有库房的工具信息以及工具的使用状况，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。

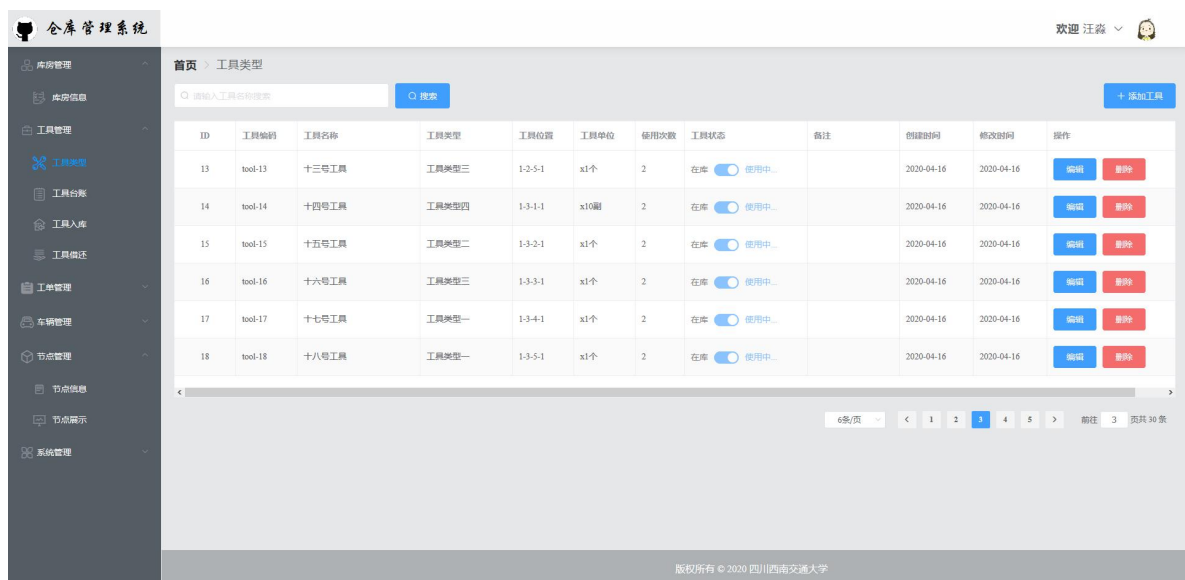


图 5-3 工具信息页面



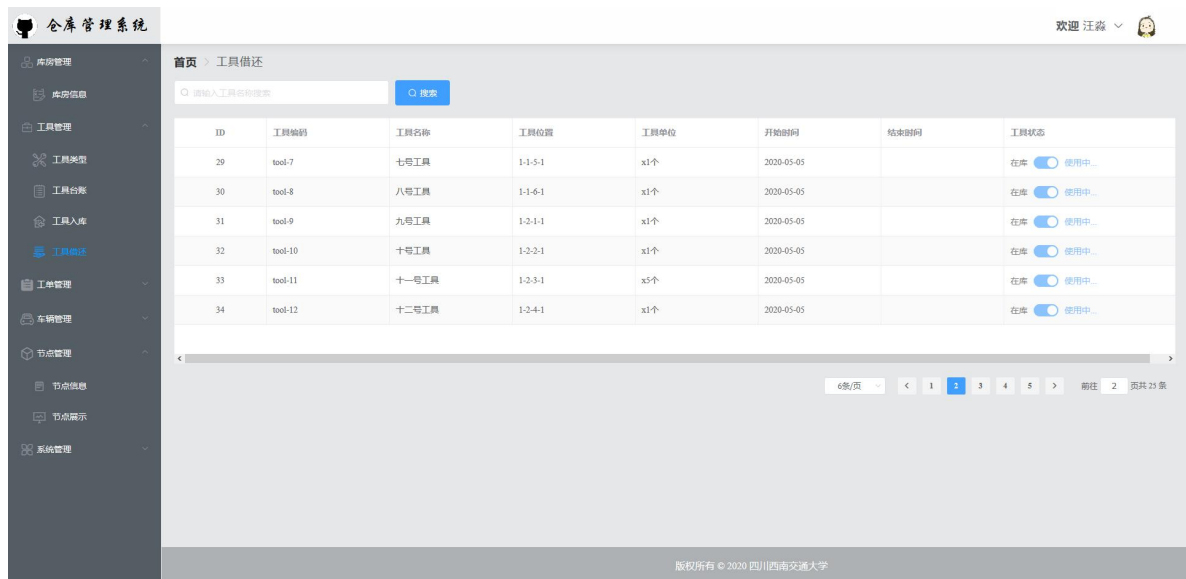


图 5-4 工具状态页

#### (4) 车辆管理

这里展示所有库房的车辆信息以及车辆的使用状况，可以从中看到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。

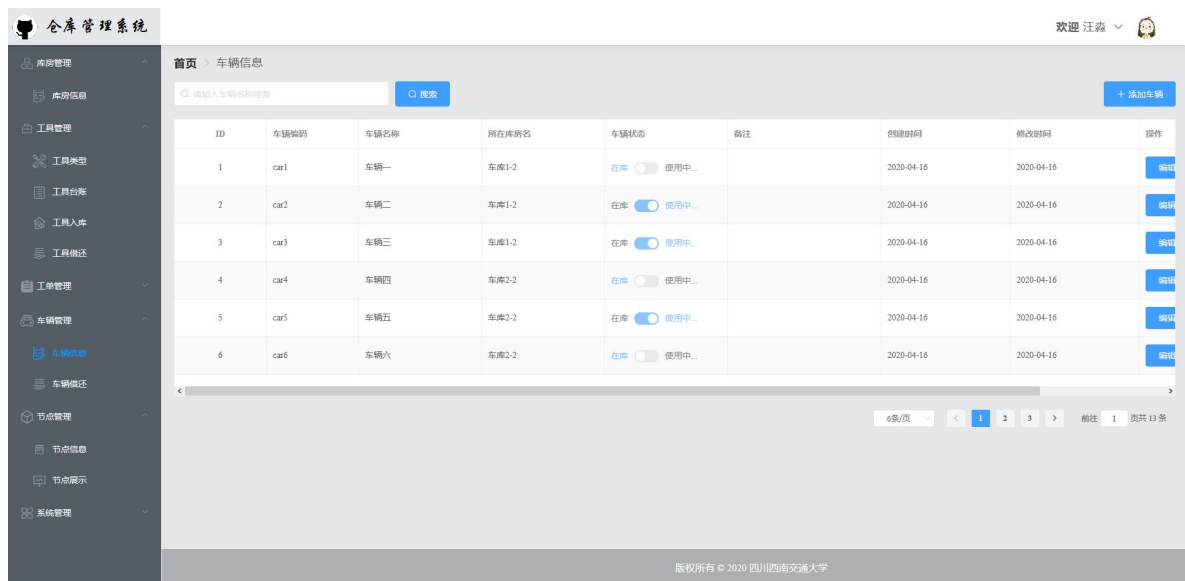


图 5-5 车辆状态页面

#### (5) 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。

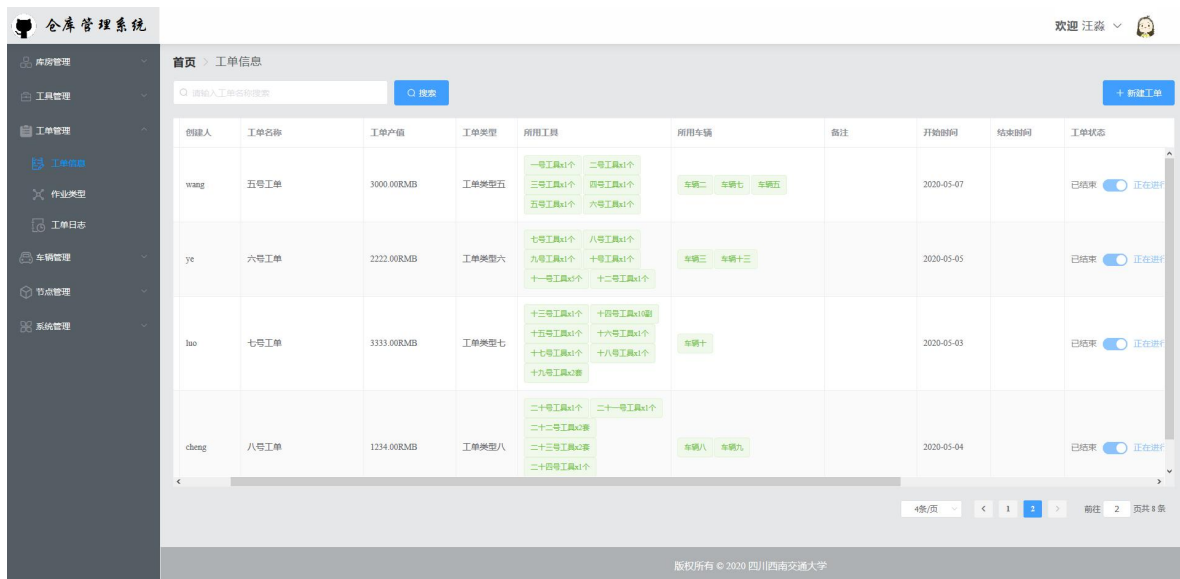


图 5-6 工单信息页

### （6）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。

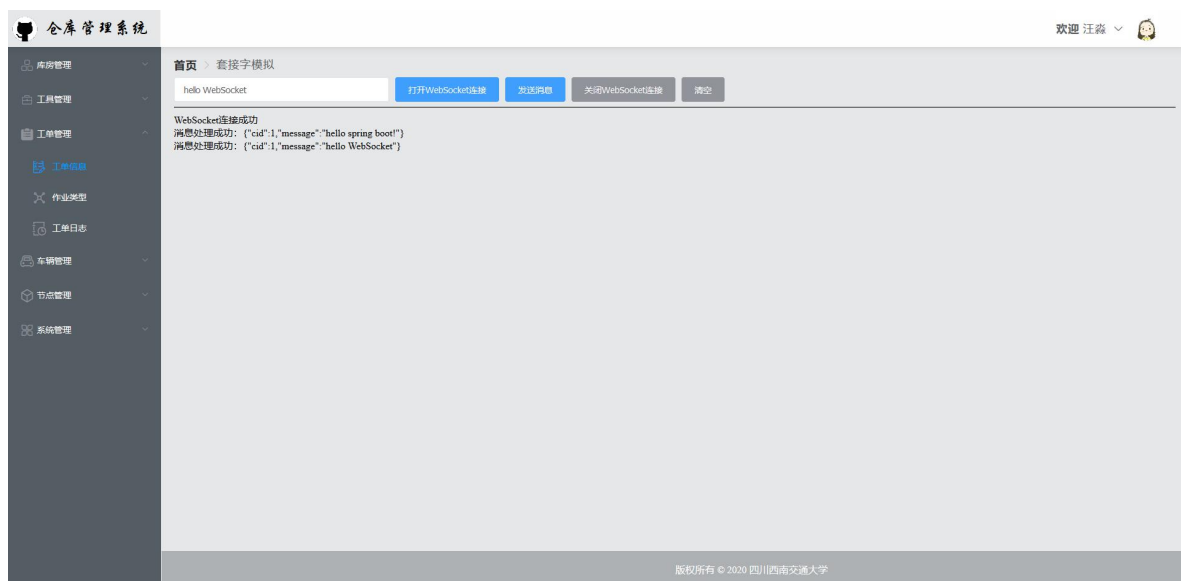


图 5-7 WebSocket 模块

## 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。

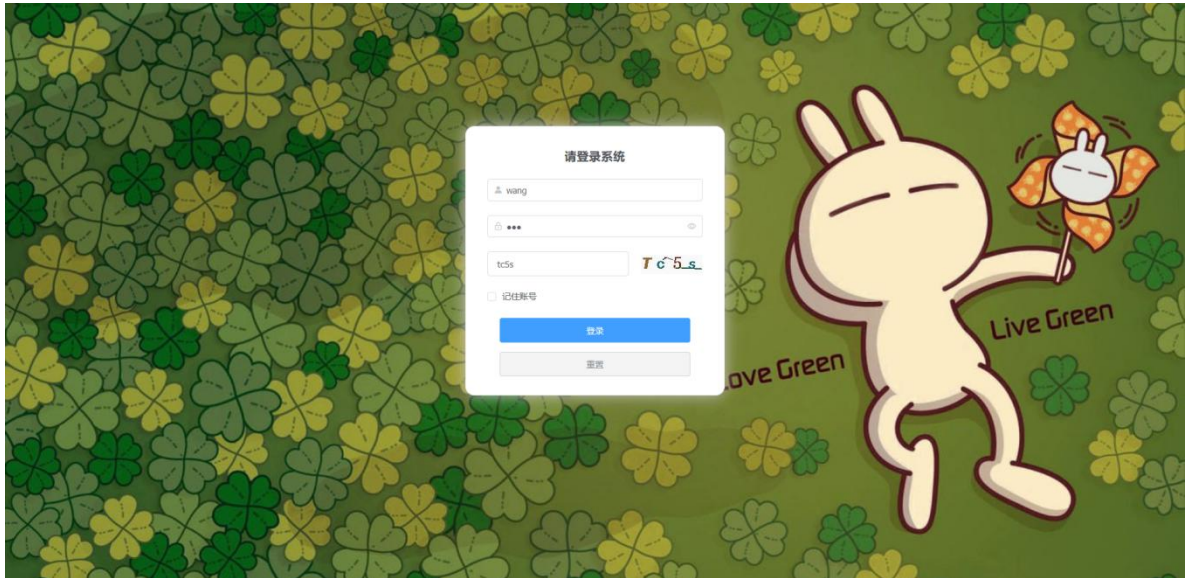


图 5-8 用户登录页



图 5-9 系统管理页

## （2）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。

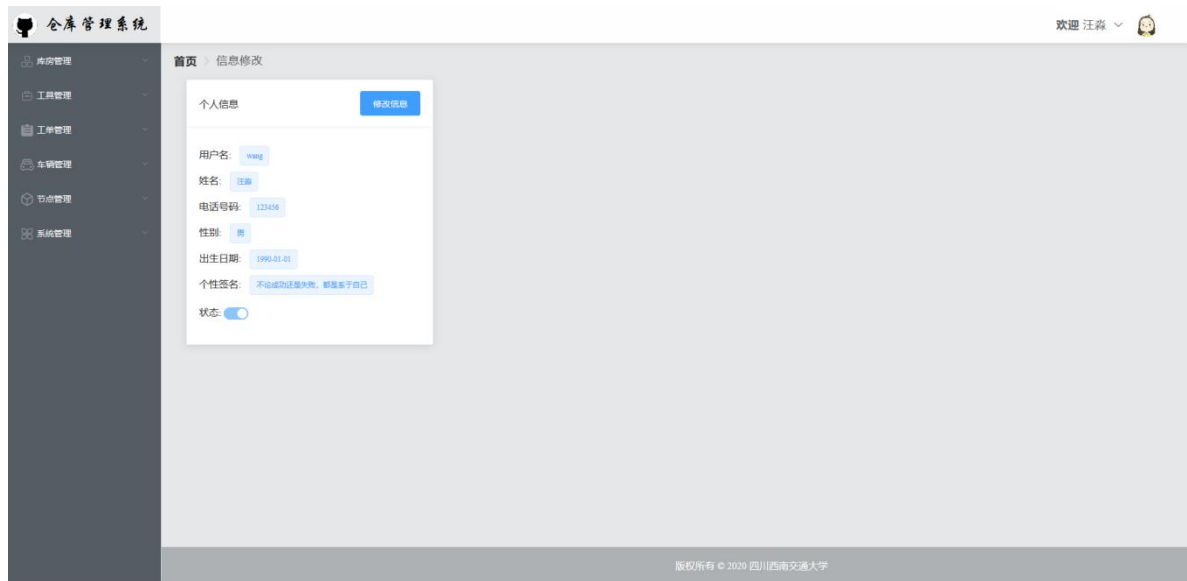


图 5-10 个人信息展示页



图 5-11 密码修改页

### （3）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。

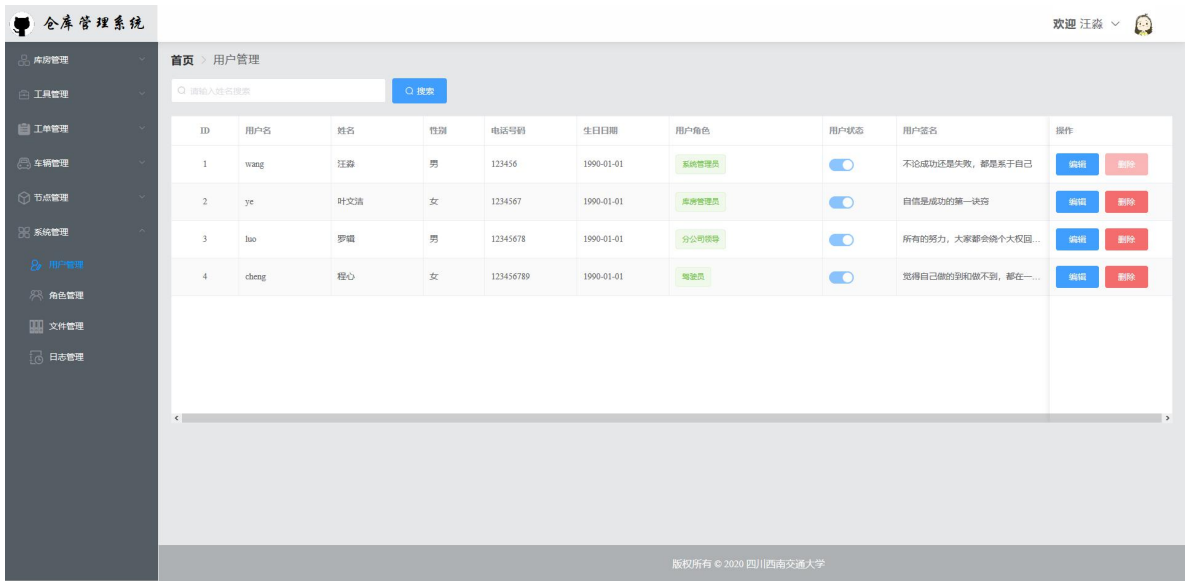


图 5-12 用户管理页

(4) 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。在编辑的时候，会有一个树形结构的控件，勾选即可添加权限。

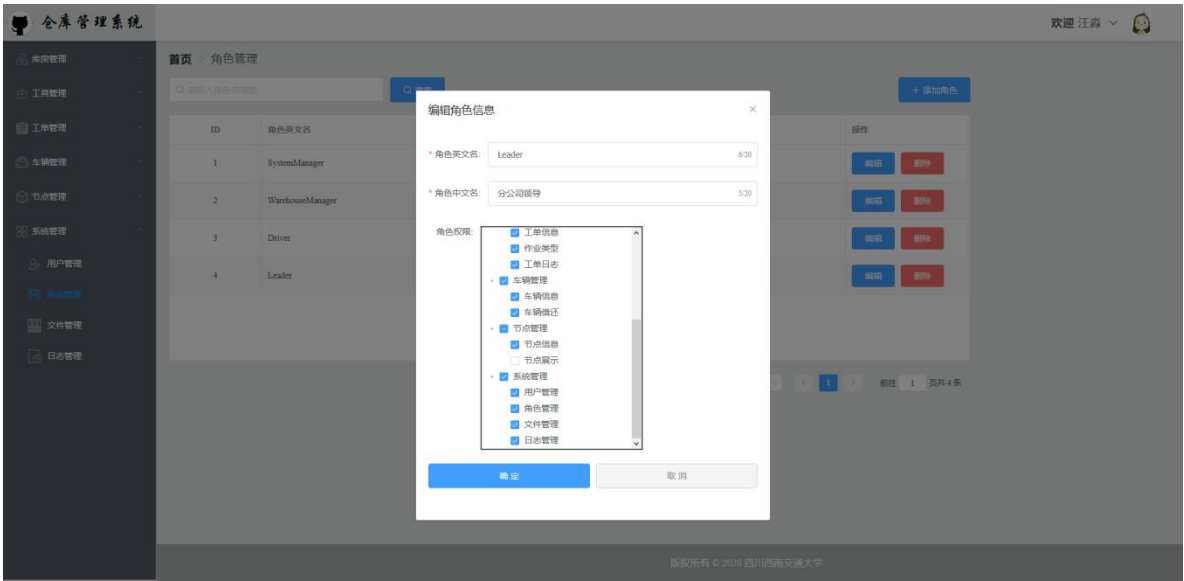


图 5-13 角色管理页

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。

所谓测试就是根据设计目标输入信息，检验系统的输出结果是否符合预期。在不同的开发阶段，测试的目的也不一样。在子模块设计时是为了检验子模块的功能是否实现，子模块测试出现异常时，测试是为了找到错误原因。错误被修改后的测试是为了检验修改后是否正确。

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	测试用例描述	操作过程及数据	预期结果	用例类型
管理员 登录模 块	进入首页，按照提示输入系统中存在的用户名和密码，图片验证码	进入登录页面，在文本框中输入系统中存在的用户名 wang，密码 123 和正确验证码	系统提示登录成功，进入管理员管理页面	判断数据正确与否
	进入管理员子系统登录窗口，按照提示输入系统中不存在用户名或错误密码或错误验证码	进入管理员子系统登录窗口，在文本框中输入系统中不存在的用户名 wan，或错误密码 12	系统提示用户名或密码或验证码错误，登录失败，返回登录页面	判断数据正确与否

表 5-2 管理员信息修改用例表

模块	测试用例描述	操作过程及数据	预期结果	用例类型
管理员 信息设 置模块	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

表 5-3 节点管理测试用例表

模块	测试用例描述	操作过程及数据	预期结果	用例类型
节点管 理模块	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 5-4 库房管理测试用例表

模块	测试用例描述	操作过程及数据	预期结果	用例类型
库房管理模块	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 5-5 工单管理测试用例表

模块	测试用例描述	操作过程及数据	预期结果	用例类型
工单管理模块	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 5-6 工具管理测试用例表

模块	测试用例描述	操作过程及数据	预期结果	用例类型
工具管理模块	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

## 5.4 本章小结

本章对智能仓库管理信息系统的具体实现进行了详细的描述，之后对于系统功能模块进行了详细的说明，给出了系统实现的界面图。最后对软件系统进行了多项的测试工作，软件通过了测试工作的认定。

## 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，进行了整个系统的需求分析、概要设计、详细设计以及最终代码的实现。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。之后介绍本课题数据库设计，包括数据库概念设计和物理设计，并说明部分重要数据表的结构。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。根据测试用例对系统进行功能测试与性能测试。编码完成后，经过测试检验，符合预期的各种需求。

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，因为个人能力的不足使得系统工作量看起来偏少，可以考虑增加更多的功能模块，升华系统的实用性。

（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 workflow 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 ASP.NET 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.