

学校代码：10255

学 号：2171804

基于 SpringBoot 和 Vue 框架的第三方医疗器械供应链平台
的设计与实现

THE DESIGN AND IMPLEMENTATION OF THIRD PARTY
MEDICAL DEVICE SUPPLY CHAIN PLATFORM BASED ON
SPRINGBOOT AND VUE FRAMEWORK

学科专业：计算机技术

学生姓名：贺紫珺

校内导师：石秀金

校外导师：贾彦君

答辩日期：2019 年 5 月 22 日

东华大学 计算机科学与技术学院

School of Computer Science and Technology

Donghua University

东华大学

工程硕士学位论文答辩委员会成员名单

| 姓名 | 职称 | 职务 | 工作单位 | 备注 |
|-----|------|---------|-----------|----|
| 李康安 | 主任医师 | 答辩委员会主席 | 上海市第一人民医院 | |
| 李继云 | 教授 | 答辩委员会委员 | 东华大学 | |
| 乐嘉锦 | 教授 | 答辩委员会委员 | 东华大学 | |
| 孙莉 | 副教授 | 答辩委员会委员 | 东华大学 | |
| 于守健 | 副教授 | 答辩委员会委员 | 东华大学 | |
| 王绍宇 | 副教授 | 答辩委员会秘书 | 东华大学 | |

东华大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，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。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

东华大学学位论文版权使用授权书

学位论文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，允许论文被查阅或借阅。本人授权东华大学可以将本学位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学位论文。

保密 ☐，在 ____ 年解密后适用本版权书。

本学位论文属于

不保密 ☐。

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

基于 SpringBoot 和 Vue 框架的第三方医疗器械供应链平台的设计与实现

摘 要

随着国家对医疗器械服务企业的监管要求越来越高,医疗器械供应链管理朝着信息化和透明化的趋势发展,本文针对 A 企业第三方医疗器械供应链管理业务的实际需求,研究了满足监管要求的第三方医疗器械供应链管理平台(Third-party medical device supply chain platform, TMDSCP)的设计与实现。

本文所做的主要工作及贡献如下:

1. 首先分析了目前市面上现有的医疗器械供应链平台,然后对 A 企业实际业务需求进行了充分的调研,最终确定了符合国家监管部门要求的第三方医疗器械供应链平台的需求。

2. 针对上述分析的需求,分析并比较目前流行的前端技术框架 Vue、React 及 Angular,以及流行的后端技术框架 Spring 与 SpringBoot。最后决定采用 B/S 架构,运用 MVC 思想,并使用前后端分离技术,前端使用 Vue.js 框架,后端使用流行框架 SpringBoot 并结合 Mysql 数据库的系统开发方案。

3. 本系统使用上述具体技术实现开发,能够管理企业人员和参数等企业基本信息,能够管理合作公司以及首营资质和品种的相关信息,并通过设计 PC 端与移动端相结合的方式,以便于管理企业实际订单业务,如创建、修改订单和订单审核等,以及通过信息化技术设置了扫码出入库功能,从而匹配企业实际仓库管理模式,还能够利用上述技术为企业产生各业务的报表分析。

4.为了提高仓库配货效率,本系统分析了业务订单中客户长期搭配购买的医疗器械,通过使用 Apriori 算法和泊松分布计算商品之间和商品数量之间的关联关系,从而设计了多套预配货模板,这些模板使仓库人员可以提前准备与模板相匹配的预配货箱子,从而提高仓库

的配货效率。系统还通过医生对每次手术订单所用预配货模板的评分为不同手术部位计算出效果最佳和性价比最高的两款模板，并在每次创建订单时推荐相应手术部位的两款模板以便进行多种选择。

本文所开发的平台为企业委托项目，现已投入使用，经实践表明其能够满足业务需求，通过了国家监管部门要求的审核，并为其他类似系统的研发提供了借鉴。

关键词：医疗器械供应链；Vue；SpringBoot；Apriori 算法；预配套

THE DESIGN AND IMPLEMENTATION OF THIRD PARTY MEDICAL DEVICE SUPPLY CHAIN PLATFORM

ABSTRACT

With the increasing regulatory requirements of medical device service enterprises in the country and the development of medical device supply chain management towards informationization and transparency, this paper studies the actual needs of A company's third-party medical device supply chain management business. Design and implementation of a third-party medical device supply chain platform (TMDSCP) that meets regulatory requirements.

The main work and contributions of this paper are as follows:

1. Firstly, it analyzes the existing medical device supply chain platform on the market, and then conducts sufficient research on the actual business needs of A-company, and finally determines the requirements of the third-party medical device supply chain platform that meets the requirements of the national regulatory authorities.
2. Analyze and compare the current front-end technology frameworks Vue, React, and Angular, as well as the popular back-end technology frameworks Spring and SpringBoot for the needs of the above analysis. Finally decided to adopt the B/S architecture, use MVC ideas, and use of the front-end separation technology, the front-end uses the Vue.js framework, the back-end uses the popular framework SpringBoot and combined with the Mysql database system development program.
3. The system uses the above specific technologies to achieve development, can manage basic information of enterprise personnel and parameters, and can manage related information of cooperative companies and first-class qualifications and varieties, and design a combination of PC and mobile, so that in order to manage the actual order business of the enterprise, such as creating and modifying orders and order audits, and setting up the scanning code warehousing function through information technology to match the actual warehouse management mode of the enterprise, it can also use the above technology to generate reports analysis for each business.

4. In order to improve the efficiency of warehouse distribution, the system analyzes the medical equipment purchased by customers in the business order for a long time, and uses the Apriori algorithm and Poisson distribution to calculate the relationship between the goods and the quantity of the goods, thus designing multiple sets of pre-plans. Distribution templates that allow warehouse personnel to prepare pre-shipment boxes that match the template in advance, thereby increasing the efficiency of the warehouse. The system also calculates the best and most cost-effective templates for different surgical sites by the doctor's score for the pre-allocation template used for each surgical order, and recommends two templates for the corresponding surgical site each time an order is created for multiple selections.

The platform developed in this paper is an enterprise commissioned project, which has been put into use. It has been shown by practice that it can meet the business needs, has passed the audit required by the national regulatory authorities, and provides reference for the research and development of other similar systems.

He Zijun(Computer Technology)

Supervised by Shi Xiujin

KEY WORDS: Medical Device Supply Chain; Vue; SpringBoot; Apriori Algorithm; Pre-match

目录

| | |
|----------------------------|------------|
| 摘 要..... | I |
| ABSTRACT..... | III |
| 第一章 绪论..... | 1 |
| 1.1 研究背景..... | 1 |
| 1.2 研究现状..... | 1 |
| 1.2.1 医疗器械系统研究现状..... | 1 |
| 1.2.2 前端框架现状..... | 2 |
| 1.2.3 后端框架现状..... | 3 |
| 1.3 研究内容..... | 3 |
| 1.4 论文章节安排..... | 4 |
| 1.5 本章小结..... | 5 |
| 第二章 相关技术..... | 6 |
| 2.1 Vue 框架..... | 6 |
| 2.1.1 MVX 框架..... | 6 |
| 2.1.2 Vue 原理解析..... | 7 |
| 2.1.3 Vue 与其他框架对比..... | 8 |
| 2.2 数据库技术..... | 8 |
| 2.2.1 关系型数据库..... | 8 |
| 2.2.2 非关系型数据库..... | 9 |
| 2.3 SpringBoot 框架..... | 9 |
| 2.3.1 Spring 框架介绍..... | 9 |
| 2.3.2 SpringBoot 框架介绍..... | 10 |
| 2.4 关联规则与 Apriori 算法..... | 10 |
| 2.4.1 关联规则相关定义..... | 10 |
| 2.4.2 Apriori 算法..... | 11 |
| 2.5 本章小结..... | 12 |
| 第三章 TMDSCP 需求分析..... | 13 |
| 3.1 总体需求分析..... | 13 |
| 3.2 功能性需求分析..... | 14 |
| 3.2.1 业务逻辑需求分析..... | 14 |
| 3.2.2 企业管理模块..... | 15 |
| 3.2.3 合作伙伴管理模块..... | 16 |

| | |
|--------------------------------|----|
| 3.2.4 订单管理模块..... | 17 |
| 3.2.5 仓库管理模块..... | 20 |
| 3.2.6 报表管理模块..... | 22 |
| 3.2.7 首营管理模块..... | 23 |
| 3.2.8 预配货模板功能模块..... | 23 |
| 3.2.9 推荐预配货模板功能模块..... | 24 |
| 3.3 非功能性需求分析..... | 24 |
| 3.4 本章小结..... | 25 |
| 第四章 TMDSCP 详细设计..... | 26 |
| 4.1 系统总体设计与模块设计..... | 26 |
| 4.1.1 系统总体设计..... | 26 |
| 4.1.2 企业管理模块设计..... | 28 |
| 4.1.3 合作伙伴管理模块设计..... | 29 |
| 4.1.4 订单管理模块设计..... | 30 |
| 4.1.5 仓库管理模块设计..... | 31 |
| 4.1.6 首营管理模块设计..... | 33 |
| 4.1.7 报表管理模块设计..... | 33 |
| 4.1.8 基于 Apriori 算法的预配货模板..... | 34 |
| 4.1.9 推荐预配货模板功能模块设计..... | 36 |
| 4.2 系统架构设计..... | 37 |
| 4.3 数据库设计..... | 38 |
| 4.3.1 数据库概念设计..... | 38 |
| 4.3.2 数据库表设计..... | 39 |
| 4.4 本章小结..... | 43 |
| 第五章 TMDSCP 实现与测试..... | 44 |
| 5.1 系统主页面的实现..... | 44 |
| 5.2 企业管理模块的实现..... | 45 |
| 5.2.1 部门/员工管理 | 46 |
| 5.2.2 角色权限管理..... | 47 |
| 5.2.3 系统参数配置..... | 48 |
| 5.3 合作伙伴管理模块的实现..... | 49 |
| 5.3.1 合作科室与医生..... | 49 |
| 5.3.2 商品分类..... | 50 |
| 5.4 订单管理模块的实现..... | 52 |
| 5.4.1 供货商进货单..... | 52 |

| | |
|------------------------|----|
| 5.4.2 手术寄售单..... | 54 |
| 5.4.3 手术寄售回收单..... | 55 |
| 5.4.4 手术寄售单审核..... | 56 |
| 5.5 仓库管理模块的实现..... | 58 |
| 5.5.1 出库管理-手术寄售单 | 58 |
| 5.6 首营管理模块的实现..... | 63 |
| 5.6.1 首营资质登记..... | 63 |
| 5.6.2 库存效期预警..... | 64 |
| 5.7 报表管理模块的实现..... | 66 |
| 5.7.1 库存汇总表功能模块..... | 66 |
| 5.7.2 统计分析..... | 67 |
| 5.8 预配货模板的实现..... | 69 |
| 5.9 推荐预配货模板的实现..... | 70 |
| 5.10 系统测试..... | 72 |
| 5.10.1 系统测试环境..... | 72 |
| 5.10.2 系统测试内容..... | 72 |
| 5.10.3 系统测试用例和结果..... | 73 |
| 5.10.4 系统测试结果分析..... | 74 |
| 5.11 系统应用情况与效果..... | 74 |
| 5.12 本章小结..... | 74 |
| 第六章 结论与展望..... | 75 |
| 6.1 结论..... | 75 |
| 6.2 展望..... | 75 |
| 参考文献..... | 76 |
| 攻读学位期间的研究成果..... | 78 |
| 致 谢..... | 79 |

第一章 绪论

1.1 研究背景

我国医疗器械产业整体发展势头迅猛，为医疗器械专业物流带来了广阔的发展机遇^[1]，但由于医疗器械产业的特殊性，例如器械本身及其双向物流渠道等，导致专业的第三方医疗器械物流企业屈指可数。由于国家对医疗器械领域的监管相对松散以及该领域的技术应用相对落后，导致医疗器械物流一直存在着许多不够规范的现象，如无法保障运输过程中的全程运输监控及产品追溯^[2]，导致出现质量失控的问题。随着国家监管越来越严格，需要企业利用信息化技术解决上述问题。

通过实际调研 A 企业的运营方式，本文设计了一款结合供应链技术以及公司内部人员安排管理的第三方医疗器械供应链平台，该平台能够提高企业业务和内部管理的效率。前端部分使用 Vue.js 技术，它是一套构建用户界面的渐进式框架，采用自底向上增量开发的设计方式^[3]，架构简单，适合开发人员快速掌握，还便于与第三方库或者已有的项目进行整合，后端部分使用 SpringBoot 技术，该技术可以简化 Spring 框架的搭建、配置、开发、调试以及部署过程^[4]，通过前后端分离的技术，实现了第三方医疗器械供应链平台的开发，从而通过信息化技术辅助企业管理。

随着互联网的进步，电子商务也在经历飞速的发展，医疗器械电商配送渐成体系^[5]，订单数量急剧增长。企业面对海量的数据，希望能从中明确各大医院对医疗器械的不同需求，本文通过分析大量订单数据，为重复搭配使用的医疗器械设计了预配货模板，使得仓库管理员面对相似商品搭配时可以提前按照预配货模板装配好相应的预配货箱子，减少了创建订单后再配货的时间，从而提高配货效率，不仅如此，还通过分析客户医院对预配货模板的评分，在每一次创建订单时通过不同部位为用户推荐效果最佳及性价比最高两款模板。

1.2 研究现状

1.2.1 医疗器械系统研究现状

如今我国医疗器械产业整体发展势头迅猛，中国是全球医疗器械十大新兴市场之一，已成为除了日本之外的亚洲最大市场^[6]，医疗器械技术的发展日益强大，为医疗器械供应链系统带来了广阔的发展机遇。

目前市面上有一款医捷云系统，这一系统的功能主要包括订单管理、仓库管理和合作伙伴基本信息管理功能。然而，通过调研 A 企业实际操作流程和分析有关部门的监管要求发现医捷云缺少了权限管理、运输信息、司机信息、医疗器械回收管理等功能。而本文设计的平台对于这些缺失的功能进行了完善，在合作伙伴模块中添加了首营管理以保存对方合作伙伴的运营资质、合同和医疗产品许可证等文件的复印件，规范产品的生产及销售过程；本平台还添加了司机的产品运输信息，以确保产品能够安全、准时的送达目的地；本平台中的权限管理也能够满足员工在实际业务中的权限控制。

在市面上还有另一家企业——新海医疗器械管理系统。但该系统的功能只有商品的进销存管理和权限管理，缺少了许多其他关于企业管理的功能，而这些功能都是在医疗器械管理中所必需的。由于医疗器械产品的特殊性，在国内关于医疗器械管理的系统只有医捷云、新海、智邦等几家，而这些系统的功能很不完善，不能够满足对相关医疗器械企业的所有管理，所以此类的系统十分紧缺。但由于国家对于医疗器械企业的监管越来越严格，本文介绍的系统不仅满足《医疗器械监督管理条例》的相关规定，与其他现有的系统相比本文系统的管理更加的符合实际流程，很大程度上减少了员工的操作，极大的提高了工作效率。

1.2.2 前端框架现状

The State Of JavaScript 发布了《2018 年的 JavaScript 现状调查报告》，调查了超过 20000 名 JavaScript 开发者，以确定目前前端框架的趋势。图 1-1 是对前端框架的调查结果，结果显示 React 和 Vue 排行靠前，Angular 有垮台的趋势。虽然 React 仍然拥有更大的市场份额，但 Vue 的崛起也没有停止的迹象^[7]。



图 1-1 StateOfJS 2018 年的 JavaScript 前端框架调查结果图

Vue.js 是一套构建用户界面的渐进式框架。是用于构建交互式的 Web 界面的库。它提供了 MVVM 数据绑定和一个可组合的组件系统，具有简单、灵活

的 API。简单来说,就是前端的一个框架,专注于 Web 端,其实实质还是 DOM 操作^[8]。它如此受欢迎的原因是 Vue.js 兼具 Angular.js 和 React.js 的优点,并剔除了它们的缺点。不仅如此,它也是开源世界华人的骄傲,因为它的作者是位中国人—尤雨溪(Evan You)。

1.2.3 后端框架现状

Spring 开源框架是现在市场占有率最高的基于 Java EE 框架的研发型软件产品,其推出的 Spring MVC 框架被广泛的应用于各类 Web 产品的研发当中。Spring 先后发布了多个版本,但各个版本之间最大的不同在于繁琐的配置工作随着版本的更新在逐步的减少,也就是说, Spring 框架一直致力于解决繁琐的配置工作问题。

SpringBoot 项目是 Pivotal 团队开发的一个全新 Spring 框架,其开发的目的在于解决 Spring 开发过程中的复杂配置操作,最终实现零配置为目标, SpringBoot 作为 Spring 项目组的微服务产品^[9],正在大规模的占领软件市场,其简化的应用开发搭建过程使得开发人员从繁琐的配置文件中解脱出来,大量的精力投入到项目逻辑研发当中,其致力于成为快速应用开发领域的领导者。

1.3 研究内容

本文所研究的第三方医疗器械供应链平台,是一个符合《医疗器械监督管理条例》相关规定的系统,本系统包括企业管理、合作伙伴管理、订单管理、仓库管理、报表管理、首营管理等 6 个模块,除此之外,本文为系统设计了预配货模板功能模块和推荐预配货模板功能模块。

本系统中企业管理模块主要负责管理企业内部人员分工、企业基本信息及企业相关信息的系统配置等;合作伙伴管理模块主要负责管理与公司有合作的其他公司的相关资料及信息等;订单管理模块主要实现系统“进-销”环节相关功能,如订单创建、修改、查看和删除等;仓库管理模块主要实现系统“进-销-存”环节中进货入库以及售货出库的处理;报表管理模块负责为企业大量信息生成一些有效的报表分析;首营管理模块实现系统有关首营的管理功能。

预配货模板功能模块通过分析所有订单中商品信息,利用 Apriori 算法和泊松分布为企业设计了预配货模板,并通过系统进行添加、查看及修改;推荐预配货模板功能模块通过分析不同手术部位的所有历史手术回收单中医生对模板的评分及模板的售价等信息,在创建订单选择手术部位后,为医生推荐效果最佳及性价比最高两款模板,以便进行多种选择。

基于上述需求,本文采用 Java EE 体系结构,运用 MVC 模式的设计思想,使用了当今流行的 SpringBoot 和 Vue 框架,结合了 Mysql 数据库,实现了第三

方医疗器械供应链平台的基本功能和推荐预配货模板功能模块，并通过 Python 实现 Apriori 算法计算出预配货模板，最后通过系统管理预配货模板相关信息。

在基于 MVC 模式的基础上，将整个系统分为 3 层：

(1) 视图层：基于 Vue 框架，采用单页面开发，通过 Vue 路由和 Ajax 进行请求，运用 Element UI 提升用户体验；

(2) 数据层：主要用来控制数据库的访问，使用了 Mysql 数据库；

(3) 控制层：基于 SpringBoot 框架，让前后端交互更加方便，也让前端的作用发挥的更大。

本文对系统中使用的前后端框架、Apriori 算法及关联规则做了深入的研究，并对各个模块和两大功能模块的相关详细实现做了阐述。

1.4 论文章节安排

根据对第三方医疗器械供应链平台的需求分析、系统详细设计、实现和测试，本文共分为六个章节来阐述，主要内容如下：

第一章：绪论

本章介绍了本文的研究背景，对医疗器械供应链系统，前端框架及后端框架的研究现状做了阐述，并简单介绍了本文的研究内容和文章结构。

第二章：相关技术

本章是对第三方医疗器械供应链平台所用到的技术和涉及的相关研究进行阐述，包括对 Vue.js 框架、SpringBoot 框架、Mysql 数据库和关联规则及 Apriori 算法的研究及相关知识的介绍。

第三章：需求分析

本章对第三方医疗器械供应链平台的需求进行分析，详细描述了本系统的设计目标、功能性需求和非功能性需求。

第四章：详细设计

本章是在前一章的基础上，对系统的详细设计，主要从技术架构和功能架构两方面对本系统进行总体设计，详细设计了系统各模块的工作流程以及系统的数据库。

第五章：实现与测试

本章主要结合了前两章的分析与设计，对本平台进行实现和测试，主要描述本平台各模块的实现过程和结果，并按照软件测试的流程进行测试。

第六章：总结与展望

本章对全文的研究内容进行了概括，通过分析实现的成果，提出本文贡献与不足，以及对未来的展望。

1.5 本章小结

本章对本文研究课题的背景做了介绍，延伸出本课题的研究意义，并基于对文献的阅读总结得出本课题的研究方向，简单叙述了本课题的研究内容，最后介绍了本文的章节安排。

第二章 相关技术

本章是对第三方医疗器械供应链平台所用到的技术和涉及的相关研究进行阐述,包括对 Vue.js 框架、SpringBoot 框架、Mysql 数据库以及关联规则和 Apriori 算法等相关知识的介绍。

2.1 Vue 框架

随着 Web2.0 的发展和移动互联网时代的到来,前端开发在整个 Web 应用软件开发中的地位越来越重要。现在的 Web 系统中越来越多的数据处理和业务逻辑开始偏向于前端,逐渐形成了“大前端”的局面,前端对性能和开发效率的要求也越来越高^[10]。这时如果不转变思路,仍然采用传统的方式,无疑会影响前端开发速度并且后期的维护也变得困难。目前市面上出现了许多优秀的前端框架已解决了上述问题,Vue 就是这样一款优秀的框架,本节将详细的介绍基于 MVVM 模式的轻量级响应式框架—Vue。

2.1.1 MVX 框架

在前端发展历程中先后出现了多种不同的架构,这里对 MVX 形式的三种架构进行介绍:

(1)MVC 架构: MVC 架构是由 Model(模型)层、View(视图)层、Controller(控制器)层构成,它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^[11],通过分层的方式,让彼此的职责分开,使得系统代码清晰,易于维护。

(2) MVP 架构: MVP 架构是从 MVC 演变而来的,相同点在于都是通过 Controller/Presenter 负责逻辑的处理,Model 提供数据,View 负责页面显示。不同点在于 MVP 架构中的 View 并不是直接地使用 Model^[12],而是通过 Presenter (MVC 架构中的 Controller)来进行的,但在 MVC 架构中 View 可以直接从 Model 中读取数据不需要通过 Controller 间接得到数据。

(3) MVVM 架构: MVVM 架构是把 MVC 里的 Controller 和 MVP 里的 Presenter 换成了 ViewModel。在 MVVM 架构中,View(视图)和 Model(数据)与 MVP 架构一样是不可以直接进行信息交互的,ViewModel 是两者信息交互的桥梁^[13]。所以此时的信息交互方式为:当 View 发生改变,ViewModel 感知到了变化,然后通知 Model 发生相应改变。反之当 Model 发生了改变,

ViewModel 能够感知到相应变化，然后通知 View 更新视图。这就是著名的双向绑定的概念。

2.1.2 Vue 原理解析

Vue 是一套强大的用于构建用户界面的渐进式框架，它是基于 MVVM 框架开发的。Vue 与其他重量级框架的区别是：Vue 采用自底向上增量开发的设计，其核心库的关注点都放在了视图层上，它不仅容易上手学习成本较低，而且 Vue 还可以结合 Vuex、Vue-router 等插件能够使得复杂的应用程序变得简单^[14]。另外需要强调的是，传统 DOM 操作可能需要使用多层复杂的选择器来选中 DOM 元素，而 Vue 通过响应式地双向绑定数据，实时地反映数据的变化并将这种变化映射到相应的数据模型上，从而避免了页面开发中 DOM 选择器繁杂的操作，简化了前端开发，提升了开发效率，降低开发时间和开发成本^[8]。

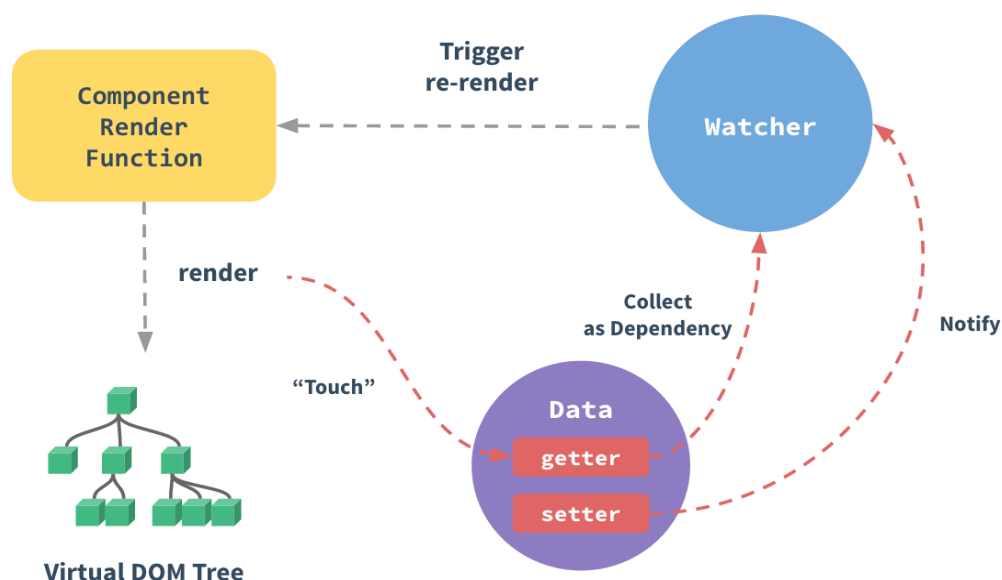


图 2-1 Vue 框架原理

Vue 框架原理如图 2-1 所示，Vue 的独特之处是响应式原理，该特性的亮点在于它把一个 JS 对象（数据模型）传给 Vue 实例的 Data 的同时，每个 Vue 实例都会有相应的 Watcher 实例对象。如果 Data 的属性发生改变，会通知 Watcher 重新计算，从而使得与其关联的组件得以更新。另外 Vue 对更新还做了一定的优化，Vue 会异步地执行相应的 DOM 更新操作。一旦观察到数据发生了变化，Vue 将自动开启一个队列结构，并且会缓冲在同一事件循环中所有的数据变化。如果存在同一个 Watcher 被多次地触发，那么只会一次推入到队列中，并且在缓冲中除去重复的数据，从而避免不必要的重复计算和频繁的 DOM 操作^[3]，以提高系统反应速度。

2.1.3 Vue 与其他框架对比

上文已经对 Vue 做了尽可能详尽的介绍，事实上，目前前端比较流行的框架还有很多，如 React、Angular 框架。下面文章将 Vue 与 React、Angular 两大主流的流行框架进行对比^[15]。

Vue 与 Angular 的相同点：

- (1) 均支持指令，如内置指令和自定义指令；
- (2) 均支持过滤器，如内置过滤器和自定义过滤器；
- (3) 均支持双向数据绑定；
- (4) 均不支持低端浏览器。

Vue 与 Angular 不同点：

(1) Angular 的学习成本高，比如增加了 Dependency Injection 特性，而 Vue 本身提供的 API 都比较简单、直观。

(2) 在性能上，Angular 依赖对数据做脏检查，所以 Watcher 越多会越慢。Vue 基于观察并且使用异步队列更新。所有的数据都是独立触发的。对于庞大的应用，这个优化差异会异常明显。

Vue 与 React 相同点：

- (1) 中心思想相同，即一切都是组件，组件实例之间可以嵌套；
- (2) 均提供了合理的钩子函数，可以让开发者定制化地去处理需求；
- (3) 均不内置列数 Ajax，Route 等功能到核心包，而是以插件的方式加载。

Vue 与 React 不同点：

- (1) React 依赖 Virtual DOM,而 Vue 使用的是 DOM 模板。
- (2) Vue 在模板中提供了指令,过滤器等,可以非常方便,快捷地操作 DOM。

2.2 数据库技术

数据库的发展已经持续了很多年，当前，主流的数据库技术分类主要分为两种，分别为关系型数据库和非关系型数据库，下面分别介绍这两个类别的数据库技术。

2.2.1 关系型数据库

关系型数据库在当前各种应用系统中得到广泛的应用，只要有数据存储的地方就有它的身影。实际上关系型数据库管理系统非常多，经过时间的检验，目前比较流行的数据库系统主要有 MySQL、Oracle、SQL Server 等。MySQL 数据库是一个开源免费的数据库，因而得到了大量中小企业的用户群体，Oracle 数据库是一个收费的高效数据库，由甲骨文公司提供相应的优质的数据库服务，因而一般大型公司会使用。SQL Server 数据库是微软公司开发和推广使用的一个优秀的

数据库，但除了利用微软体系开发的应用系统，其他体系使用的并不广泛。所以通过综合对比，本文最终选取 MySQL 作为数据存储工具，下面对 MySQL 进行介绍：

MySQL 数据库是瑞典一家公司开发的完全网络化的跨平台关系型数据库系统，是一个真正的多用户、多线程 SQL 数据库服务器，同时是具有客户机/服务器体系结构的分布式数据库管理系统。它具有功能强、使用简便、管理方便、运行速度快、安全、可靠性强等优点，用户可利用许多语言编写访问 MySQL 数据库的程序^[16]。

2.2.2 非关系型数据库

与关系型数据库相对应的就是非关系型数据库，非关系数据库比起关系型数据库，其特点也是非常明确的，它可以存储各种类型的数据，并没有特定的关系约束，比较常见的非关系型数据库主要有：Redis、Hbase、MongoDB 等^[17]。Redis 数据库因其速度快，经常被作为缓存来存储一些被频繁访问的数据，Hbase 的主要应用场景是大数据，由于其自身特点，在大数据领域中得到广泛的应用。MongoDB 数据库是一个介于关系数据库和非关系数据库之间的产品^[18]，是非关系型数据库当中功能最丰富，最像关系型数据库的。通常被用来存储关系不紧密的数据，如大量复杂的事务逻辑操作。

2.3 SpringBoot 框架

2.3.1 Spring 框架介绍

Spring 框架是一款优秀的 Java 框架，使得程序员从繁重的 EJB(Enterprise JavaBean)中解脱出来，其核心功能是 AOP(面向切面编程)和 IOC(控制反转)，它的完整架构图如图 2-2 所示^[19]：

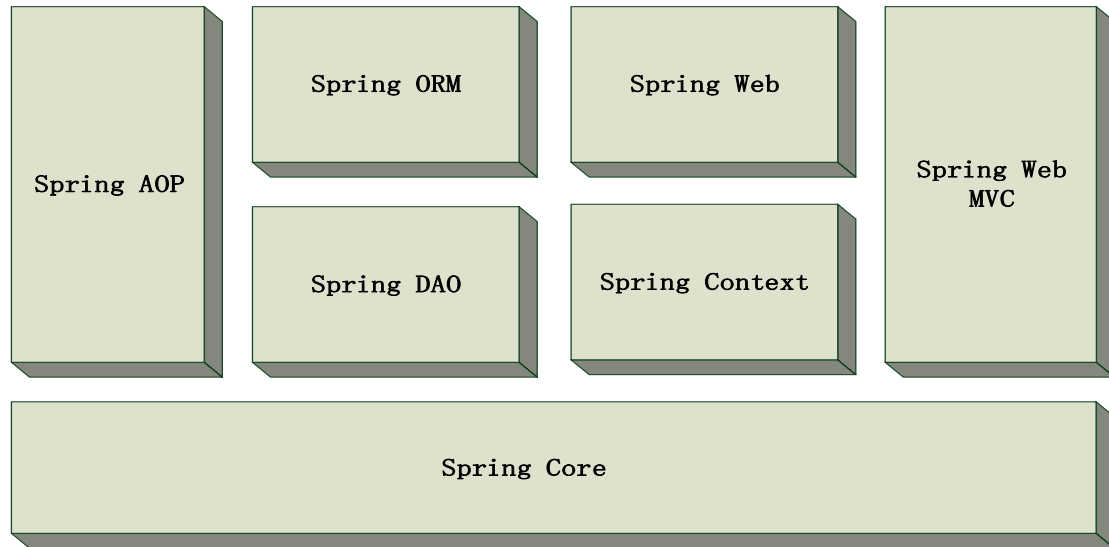


图 2-2 Spring 框架结构

2.3.2 SpringBoot 框架介绍

本文第 2.3.1 节提及 Spring 的出现提高了开发效率，但也存在一些问题，如 Spring 框架的配置过于繁琐，而且有些配置相对固定，但若没有配置，系统会报错。为了简化 Spring 开发，避免程序员将过多的时间浪费在配置文件上，SpringBoot 应运而生。SpringBoot 的出现使得程序员从繁重的配置文件中解放出来，专心进行业务的开发，极大的提高了开发效率，因而受到广泛的欢迎和应用。

SpringBoot 的特点主要有以下几点：

- (1) 创建独立的 Spring 应用程序；
- (2) 嵌入的 Tomcat，无需部署 WAR 文件；
- (3) 简化 Maven 配置；
- (4) 自动配置 Spring；
- (5) 提供生产就绪型功能，如指标，健康检查和外部配置；
- (6) 绝对没有代码生成并且对 XML 也没有配置要求^[20]。

2.4 关联规则与 Apriori 算法

如果两个或多个变量的取值之间存在某种规律性，则称为关联。关联规则是寻找在同一个事件中出现的不同项的相关性，在机器学习研究领域，学者已经提出多种关联规则的挖掘算法，如 Apriori，FP-growth，Eclat 算法等^[21]，本文主要使用的是 Apriori 算法。

2.4.1 关联规则相关定义

定义 1：项，项集和 k-项集，事务和事务集。假设 $I = \{i_1, i_2, \dots, i_m\}$ 表示所有项的集合，I 是 m 个不同项的集合，每个 i_k ($k=1,2,\dots,m$) 称为项，I 为项的集合，即项集；包含 k 个项的项集称为 k-项集。假设 $D = \{T_1, T_2, \dots, T_n\}$ 表示所有事务的

集合，每个事务 T_i 由一个或多个项构成，即 $T_i \subseteq I$ 。假设以 0 和 1 表示项是否在事务中出现，具体表示见表 2-1^[22]：

表 2-1 事务数据库 D 中的二元 0/1 表示

| TID | i_1 | i_2 | | i_m |
|-------|-------|-------|-------|-------|
| T_1 | 1 | 0 | | 1 |
| T_2 | 1 | 1 | | 0 |
| | | | | |
| T_n | 0 | 1 | | 0 |

定义 2：关联规则。关联规则可以表示成 $X \Rightarrow Y$ 的蕴含式，其中 $X \in I$ ， $Y \in I$ ，且 $X \cap Y = \emptyset$ ^[23]。

定义 3：关联规则的支持度。关联规则的支持度是指事务数据库 D 中，蕴含有 X 以及 Y 事务占据全部事务的百分比，即：

$$\text{support}(X \Rightarrow Y) = \frac{D \text{中包含项集 } X \cup Y \text{ 的事务数}}{D \text{中事务总数}} = P(X \cup Y) \quad (2-1)$$

一般将其看成是最小支持度，记为 \min_sup 。

定义 4：置信度。关联规则 $X \Rightarrow Y$ 置信度是指 D 中包含 X 的事务中同时也包含 Y 事务的百分比，即为：

$$\text{confidence}(X \Rightarrow Y) = \frac{\text{support}(X \Rightarrow Y)}{\text{support}(X)} = \frac{P(X \cup Y)}{P(X)} = P(Y|X) \quad (2-2)$$

一般将其看成是最小置信度，记为 \min_conf 。

定义 5：频繁项集。满足最小支持度的项集称为频繁项集，记为 L_k 。

定义 6：强关联规则。强关联规则指的是满足最小支持度和最小置信度的规则。关联规则挖掘过程分为 2 个步骤：找出事务数据库中所有频繁项集；由频繁项集产生强关联规则。

2.4.2 Apriori 算法

Apriori 算法^[24]是关联规则中经典算法之一。要想获得频繁 k 项集，必须通过频繁 k-1 项集迭代得到，以此类推，得到全部项集。Apriori 算法主要步骤如下：

(1) 连接

要得到频繁项集 L_k ，首先需要计算出候选项集 $C_k = L_{k-1} \times L_{k-1}$ 。假设 l_1 和 l_2 是 L_{k-1} 内的两个项集，且 l_1 和 l_2 中有(k-2)个项相同，假设相同项为 $\{l[1], l[2], \dots, l[k-2]\}$ ，不同项分别为 $l_1[k-1]$ 和 $l_2[k-1]$ ，则表示 l_1 和 l_2 可连接，连接结果为 $C_k = \{l[1], l[2], \dots, l[k-2], l_1[k-1], l_2[k-1]\}$ 。

(2) 剪枝

由于 C_k 是 L_k 的超集，故 C_k 中可能包含由 L_k 自连接产生的非频繁项集，需将这些非频繁项集删除，才可得到 L_k 。主要是利用 Apriori 算法的先验性质：若有

某一候选项集 $t \in C_k$ ，且 t 的某个子集 $s \notin L_k$ ，即 s 不是频繁项集，故将 t 从 C_k 中删除。

(3) 产生强关联规则

由 (1) 和 (2) 步骤找出事务数据库 D 中的频繁项集 L_k 之后，通过以下步骤得到满足最小支持度和最小置信度的关联规则：1. 计算出每个频繁项集 l 的所有非空子集；2. 对于每个 l 的非空子集 p ，若 $\frac{\text{support}(l)}{\text{support}(p)} \geq \text{min_conf}$ ，则称 $p \Rightarrow (l - p)$ 为强关联规则。

2.5 本章小结

本章介绍了本文研究所需要的相关技术。首先文章从系统的实现上介绍了系统所用的前端技术 `Vue.js` 并将它与其他框架进行了比较，然后介绍了后端技术 `SpringBoot`，以及数据库技术 `Mysql`，并分析了关系型数据库及非关系型数据库的特点，最后介绍了关联规则和 `Apriori` 算法的相关概念。

第三章 TMDSCP 需求分析

本章节主要从第三方医疗器械供应链平台的总体需求、系统功能性需求和非功能性需求三个方面进行分析，确定系统设计的目的以及功能。

为了便于文章的理解，现将文中出现较多的名词解释如下：

合作伙伴：与本企业签订了相关合同，有商业往来的单位或个人。

首营：与本企业首次发生供需关系的生产企业或经营企业或者本企业与生产企业或经营企业首次发生供需关系所涉及的产品。

供货方：直接向本企业提供医疗器械产品的合作伙伴（生产企业或委托方）。

生产企业：本企业所储运的产品的生产企业。

经销商：获得企业授权的，在某一区域和领域只拥有销售该企业授权产品或提供服务权利的单位或个人。

委托方：将产品交与本企业代为储运的单位或个人。

3.1 总体需求分析

本系统是由订单管理模块、仓库管理模块、合作伙伴管理模块、企业管理模块、首营管理模块以及报表管理模块等六个模块组成。

其中订单管理模块和仓库管理模块主要是为了本平台供应链的整个流程的运转而设置，报表管理模块和首营管理模块主要是为了适应监管部门要求，给监管部门进行信息查看而设置，而其他两个模块主要是为了使整个供应链流程的操作更便捷而设置的。订单管理模块负责企业所有订单的增加、删除、修改、查看及审核，是系统供应链流程最重要的环节，仓库管理模块负责管理企业所有订单中商品的“进-出”及企业的预配货模板，合作伙伴管理模块负责管理与企业有合作的其他企业的信息以及企业的商品信息，企业管理模块负责管理企业人员职责、系统参数信息及企业基本信息，报表管理模块负责管理企业各种业务的统计报表，该模块是提供给监管部门查看的模块，首营管理模块负责管理与首营业务相关信息，该模块也是提供给监管部门查看的模块。为了使系统得到更好的运转，本文为系统设计了预配货模板功能模块和推荐预配货模板功能模块，从而提高仓库配货效率，并增加了用户的选择。

本平台最终是以满足目前医疗器械行业的业务需求及监管要求为前提，通过相关技术的实现，从而使业务操作更简便，易上手，减少人为出错的概率。本系

统是一个集第三方医疗器械企业的业务流程和企业管理为一体的综合应用系统，完全符合食药监有关 GSP 监管要求，满足医疗器械行业经营与现代化医疗器械第三方物流服务系统的实际需求，实现了操作流程化、规范化和标准化。

3.2 功能性需求分析

本文分析了企业的实际情况后，将系统的用户分为以下几种不同角色^[25]：系统管理员，药监人员，销售人员，订单审核员，仓库管理员，理配员和司机。

(1) 系统管理员

系统管理员拥有最高权限，包括订单管理模块、仓库管理模块、企业管理模块、合作伙伴管理模块、报表管理模块及首营管理模块这 6 大模块的所有功能，即拥有销售人员、药监人员、订单审核员和仓库管理员等角色的所有权限。

(2) 药监人员

药监人员是负责监管企业的人员，该角色只能够查看报表管理模块及首营管理模块的信息，但不能做任何修改。

(3) 销售人员（跟单员）

销售人员是专门负责订单模块里创建/修改订单的业务员，也就是俗称的“跟单员”，只允许操作订单模块中创建/修改订单的操作。

(4) 订单审核员

订单审核员是负责审核销售人员所添加/修改订单信息的准确性以及本次订单所需商品与库存量是否匹配，并为该订单找到一个满足需求的仓库。

(5) 仓库管理员

仓库管理员负责指派理配员和司机，并审核理配员生成的配货单是否与将要发给客户的货物信息相符。

(6) 理配员

理配员负责入库和配货，并生成相应的入库单或配货单。

(7) 司机

司机负责确认订单发货的情况，并且该角色只存在于与出库相关的业务。

3.2.1 业务逻辑需求分析

本系统主要业务是第三方企业的医疗器械供应链，即“进-销-存”，首先上游企业将医疗器械出售给企业，此为公司的“进”货操作；企业将所购入的货物放至自己的仓库中，此为公司的“存”货操作；最后，企业将仓库中的器械出售给下游企业，此为公司的“销”货操作。如此，一套供应链的流程就形成了。

根据总体需求分析和以上业务逻辑分析，本系统主要完成以下功能，系统用例图如图 3-1 所示，由于系统管理员拥有所有角色的权限，故下面分模块叙述时拥有重复功能的用例图不表示出来。

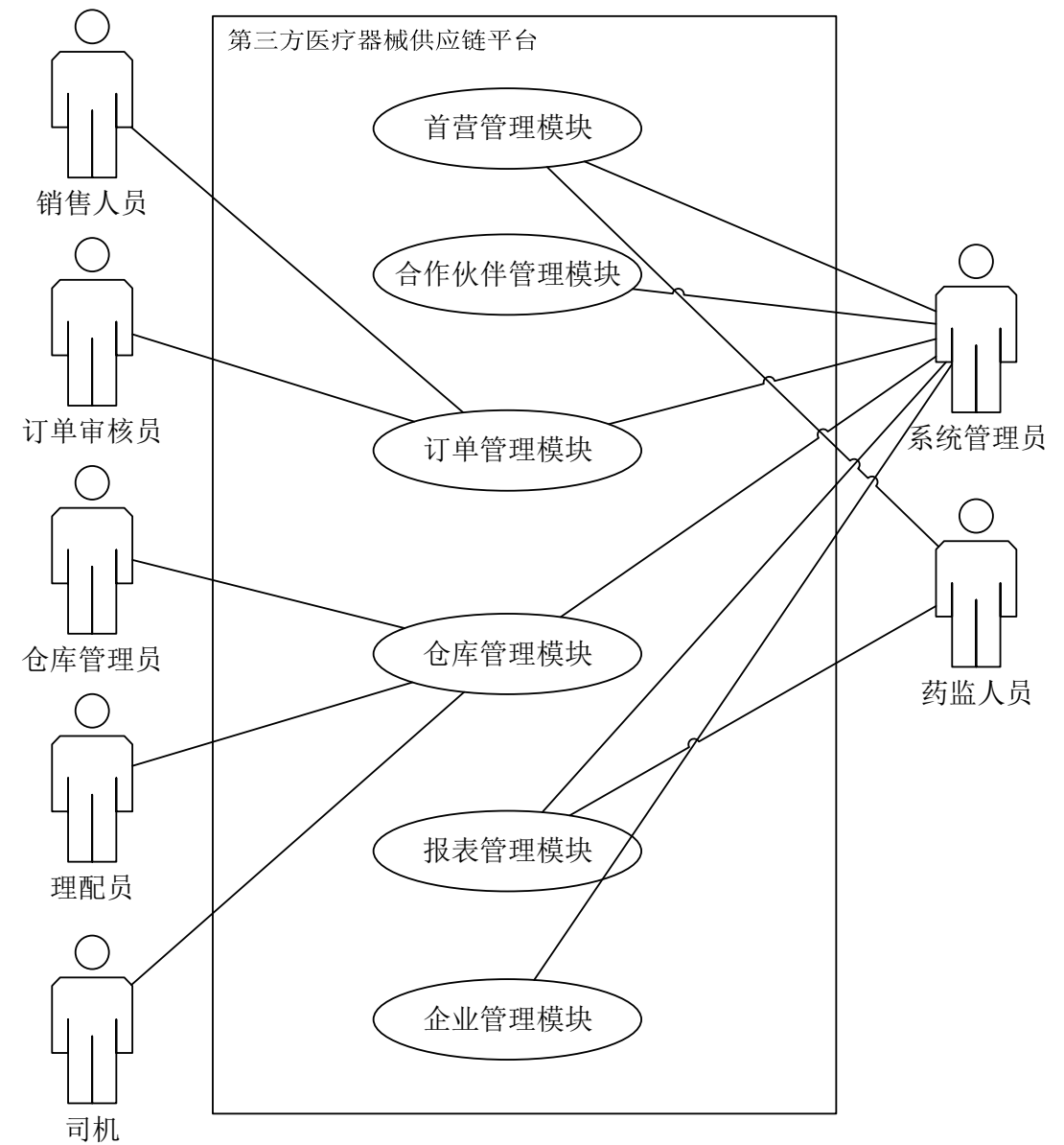


图 3-1 系统用例图

3.2.2 企业管理模块

企业管理模块主要实现企业内部管理的功能。该模块包含部门/员工管理、角色权限管理、销售人员管理、企业基本信息管理及系统参数配置等 5 个子模块，这些子模块大部分是为了便于其他模块的操作而设计的，如设置各个角色权限及系统的相关参数。企业管理模块用例图如图 3-2 所示：

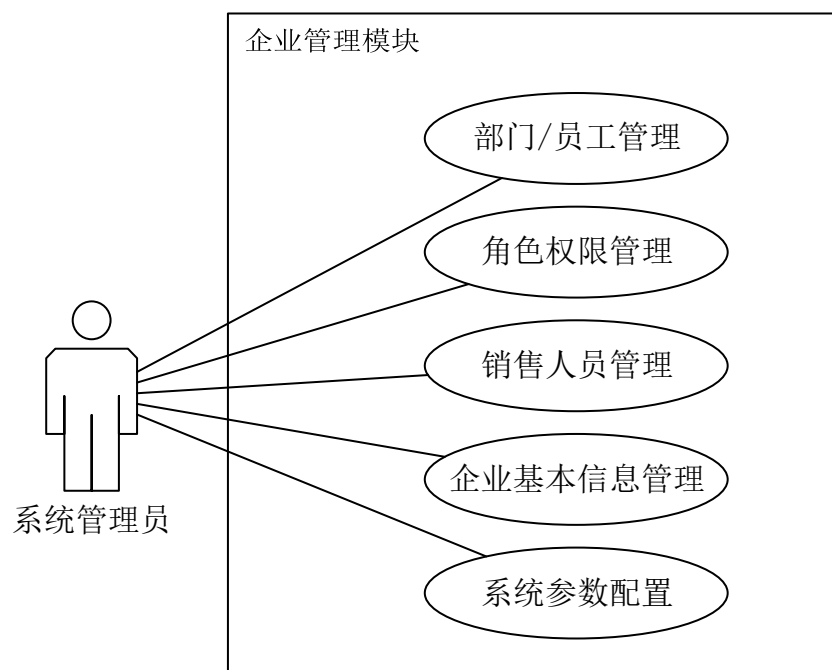


图 3-2 企业管理模块用例图

3.2.2.1 部门/员工管理

该子模块主要是为了管理企业的部门以及对应部门里员工的相关信息，可以增加、删除及修改部门/员工的信息。

3.2.2.2 角色权限管理

该子模块主要是为了管理系统的角色，从而可以限制一些角色的权限。

3.2.2.3 销售人员管理

该子模块主要是为了给客户医院设置对应的销售人员。

3.2.2.4 企业基本信息管理

该子模块主要是添加企业的相关合法信息。信息用图片和文字表示，如企业的一些合格证等。

3.2.2.5 系统参数配置

该子模块主要是配置系统的一些常用参数，比如器械的过期天数、产品品牌、产品类型、销售类型、产品线和客户医院手术的就诊方式等。

3.2.3 合作伙伴管理模块

该模块分为合作伙伴和商品信息 2 个子模块，合作伙伴子模块主要是存储一些与企业合作的公司或个人的信息，商品信息子模块主要是存储企业里商品的一些信息。合作伙伴管理模块用例图如图 3-3 所示：

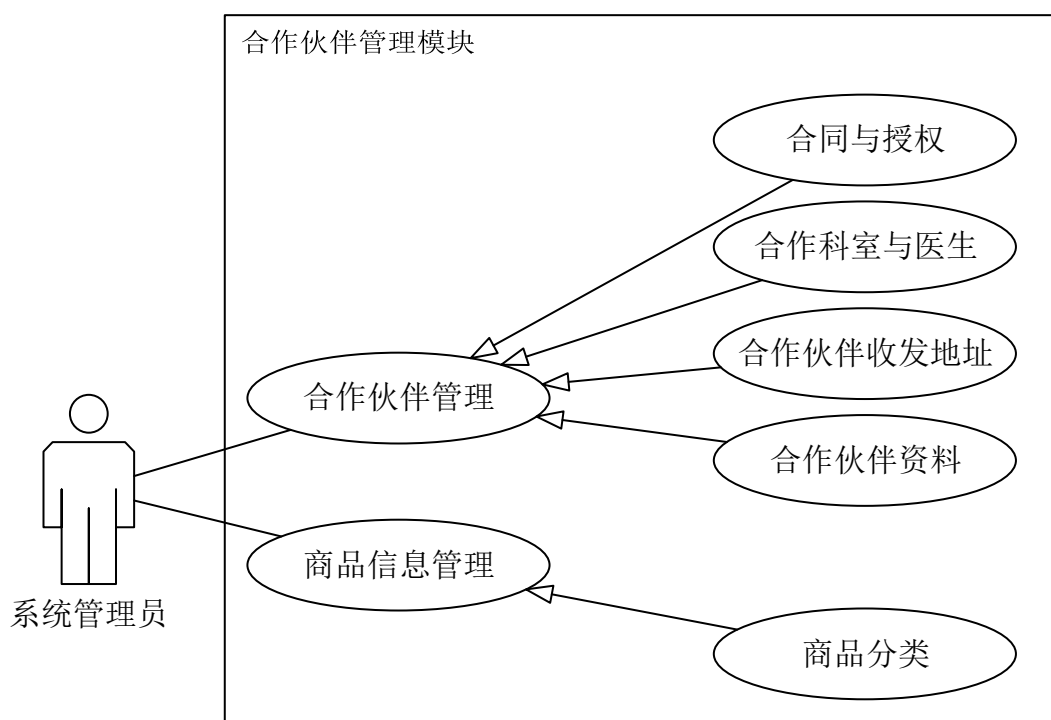


图 3-3 企业管理模块用例图

3.2.3.1 合作伙伴

该子模块包括 4 个小部分：合作与授权，合作科室与医生，合作伙伴收发地址，合作伙伴资料。

(1) 合作与授权

主要为了存储与企业签约的公司或个人的合作信息，比如合作类型等。

(2) 合作科室与医生

主要为了存储与企业合作的客户医院及该医院对应的科室和该科室对应的医生的信息等。

(3) 合作伙伴收发地址

主要是存储所有合作伙伴的收/发货地址。

(4) 合作伙伴资料

主要是为了存储合作伙伴的公司信息。

3.2.3.2 商品信息

该子模块包括 1 个小部分：商品分类。

主要是通过商品的品牌、类型和分类名称对商品进行分类，便于搜索企业中已有的一些商品的信息。

3.2.4 订单管理模块

在订单管理模块主要实现系统“进-销”环节相关订单的处理，包括采购订单、经销商订单、手术订单和借调单以及对上述订单的审核等。

订单模块的用例图如图 3-4 所示，销售人员负责创建或修改所有订单，订单审核员负责对销售人员创建或修改的订单进行审核：

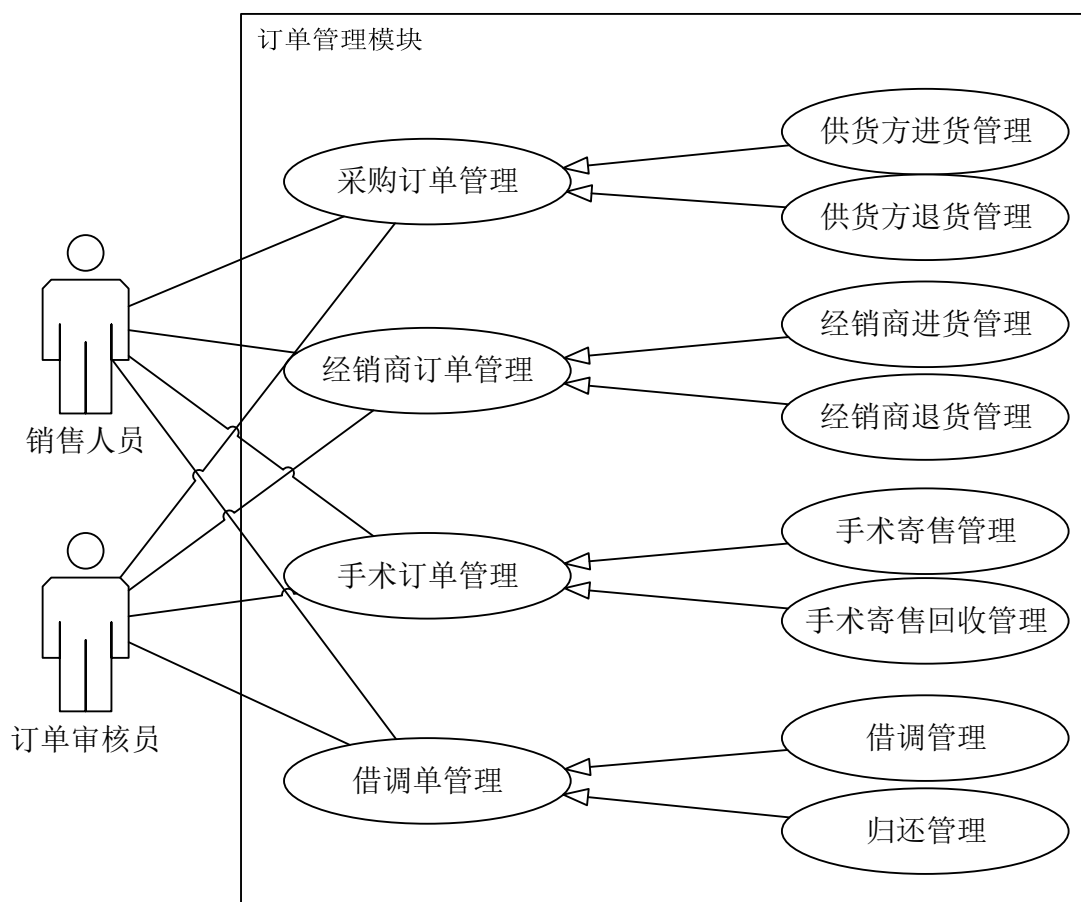


图 3-4 订单管理模块用例

(1) 采购订单管理

包括供货方进货单和供货方退货单，该子模块主要功能是管理企业每次进货时向上游供货方进货的订单信息以及向供货方提出退货的订单信息。

(2) 经销商订单管理

包括经销商进货单和经销商退货单，该子模块主要功能是管理企业每次销售给下游经销商的订单信息以及下游经销商退货的订单信息。

(3) 手术订单管理

包括手术寄售单和手术寄售回收单，该子模块主要功能是管理企业的手术寄售单和手术结束后产生的手术寄售回收单的相关订单信息。

(4) 借调单管理

包括借调单和借调归还单，该子模块主要功能是管理企业每次与兄弟公司的借入和借出相关操作的订单信息。

3.2.4.1 采购订单管理

1. 供货方进货管理

(1) 能够显示所有供货方进货单的基本信息，以及查看每个订单的订单详情，并根据订单状态对该类型所有订单进行筛选，还可以通过订单编号对某个订单进行搜索；

(2) 能够创建新的或修改已有的进货单信息。

该订单包含的基本信息有：供货方订单编号（供货方提供）、单据类型、产品品牌、供货方、客户代码、采购员、联系方式、付款方式、收货地址及备注；

该订单包含的订单详情信息有：商品编号、商品数量、商品名称、商品单位、商品单价、商品总价及备注。

2. 供货方退货管理

(1) 能够显示所有供货方退货单的基本信息，以及查看每个订单的订单详情，并根据订单状态对该类型所有订单进行筛选，还可以通过订单编号对某个订单进行搜索；

(2) 能够创建新的或修改已有的退货单信息；

该订单包含的基本信息有：供货方订单编号（供货方提供）、单据类型、供货方、客户代码、退货员、联系方式、退货原因、预计出库时间及备注；

该订单包含的订单详情信息有：商品编号、商品数量、商品名称、商品单位、商品单价、商品总价、效期、注册证及备注。

3.2.4.2 经销商订单管理

1. 经销商进货管理

(1) 能够显示所有经销商进货单的基本信息，以及查看每个订单的订单详情，并根据订单状态对该类型所有订单进行筛选，还可以通过订单编号对某个订单进行搜索；

(2) 能够创建新的或修改已有的进货单信息；

该订单包含的基本信息有：订单编号、产品品牌、单据类型、经销商名称、联系人、联系方式、销售经理、销售经理联系方式及备注；

该订单包含的订单详情信息有：商品数量、商品名称、商品编号、商品单价、商品总价。

2. 经销商退货管理

(1) 能够显示所有经销商退货单的基本信息，以及查看每个订单的订单详情，并根据订单状态对该类型所有订单进行筛选，还可以通过订单编号对某个订单进行搜索；

(2) 能够创建新的或修改已有的退货单信息；

该订单包含的基本信息有：订单编号、产品品牌、单据类型、经销商名称、联系人、联系方式、销售经理、销售经理联系方式、退货原因及备注；

该订单包含的订单详情信息有：商品编号、商品数量、商品名称、商品单位、商品单价、商品总价、效期、注册证及备注。

3.2.4.3 手术订单管理

1. 手术寄售管理

功能与采购订单和经销商订单类似，只是订单详细信息中的基本信息和商品信息不同，手术寄售单的基本信息和商品信息如下：

基本信息：销售类型、产品线、品牌、产品类型、到货时间、手术模板、医院名称、手术名称、手术部位、医生、手术时间、病人姓名、性别、年龄、床位号、住院号、是否加急、预配货模板；

商品信息：商品编号、商品数量、商品名称及备注。

2. 手术寄售回收管理

每一笔手术寄售单，医生手术使用完后，企业需进行回收，故在回收时自动生成手术寄售回收单，并与相应的手术寄售单进行对应。该部分需要填写相应的手术实际使用的器械信息，方便财务进行费用的评估。

3.2.4.4 借调单管理

1. 借调管理

(1) 与上述类型的订单管理类似，能够对所有借调单（包括借入和借出单）进行管理。

(2) 能够创建新的或修改已有的借入/出订单。订单基本信息包括：借入/出公司名称、公司联系人、单据类型、联系方式、销售经理、联系方式、预计归还时间、备注。订单商品信息包括：商品名称、商品数量、商品编号及备注等。

2. 归还管理

(1) 能够查看所有待归还的订单及已归还订单的基本信息和详细信息；

(2) 若实际器械已归还，能够将未归还的订单状态变为已归还。

3.2.4.5 订单审核

上述提到的供货方进货单、供货方退货单、经销商进货单、经销商退货单、借入单、借出单、手术寄售单及手术寄售回收单等创建或修改完成后，订单状态变为“未审核”。

该功能模块主要是审核订单信息有无错误及商品数量是否与仓库库存匹配，若订单有误或不匹配，则进行驳回操作并提交驳回意见，将订单状态变为已驳回，然后订单返回至跟单员处，跟单员重新修改订单信息；若订单无误，订单审核员根据订单信息中的货物及各仓库的库存进行分配仓库操作，并将订单状态变为已审核，提交至相应的仓库管理员处。

3.2.5 仓库管理模块

在仓库管理模块主要实现系统“销-存”环节相关订单的处理，主要会涉及到入库和出库两种操作，该模块包括 3 个子模块：入库管理、出库管理和预配货管理。该模块用例图如图 3-5 所示：

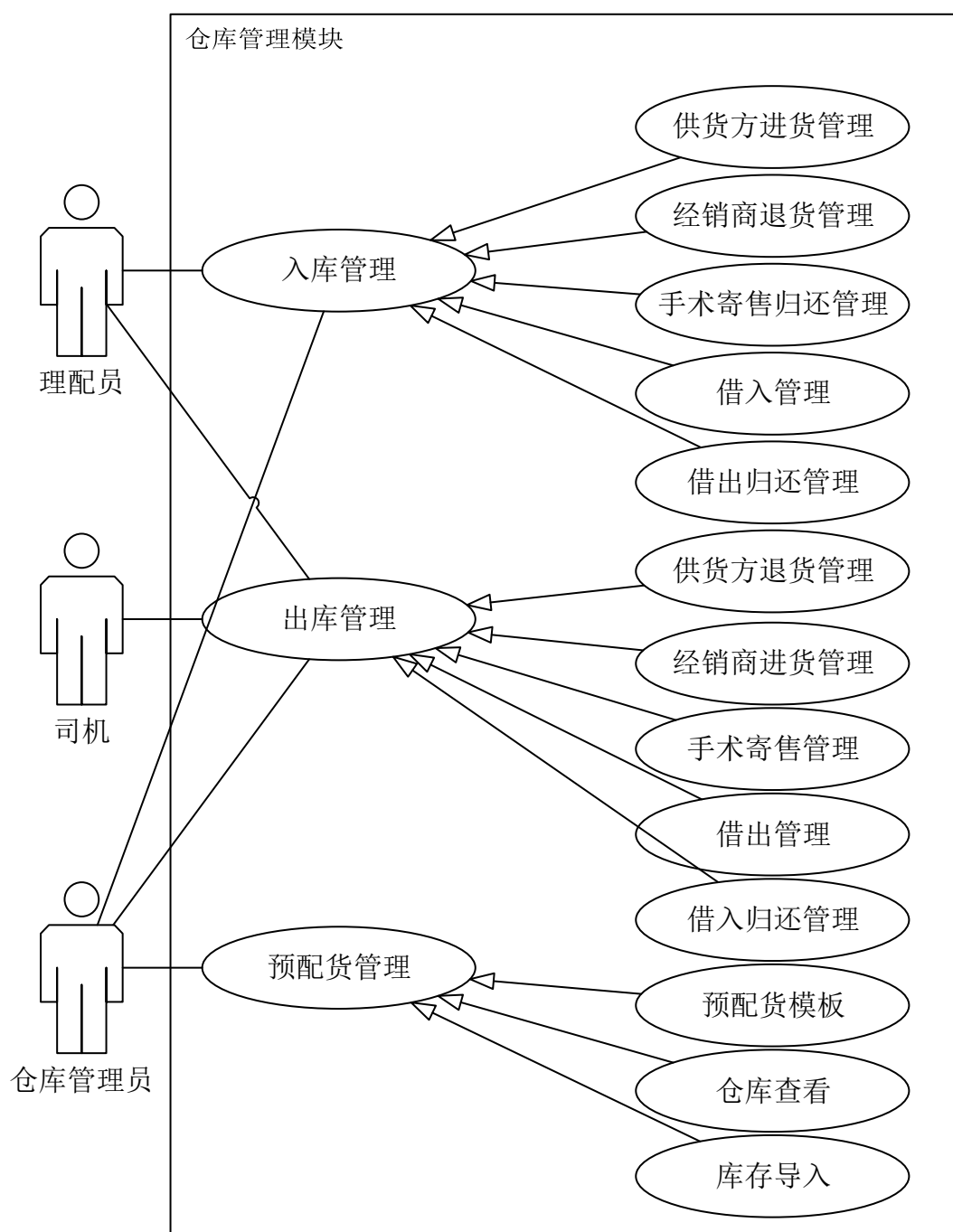


图 3-5 仓库管理模块用例图

仓库管理模块用例图如图 3-5 所示，仓库管理员负责仓库所有的管理，包括仓库货物的管理以及出入库管理，理配员负责整理仓库里货物，如入库时对货物进行扫码入库的操作，出库时对货物进行扫码出库操作，司机负责运送货物，并在系统里提交运输相关信息等。

3.2.5.1 出库管理

只有企业“销”货时，才会出现出库的情况。所以，该功能只涉及 5 种订单：供货商退货单、经销商进货单、手术寄售单、借出单和借入归还单。出库操作的步骤如下：

(1) 选择理配员

以上 5 种订单通过订单审核并分配好仓库后，会将订单权限移至相应仓库的仓库管理员处，由仓库管理员指定对该订单进行出库操作的理配员。

(2) 配货

理配员收到配货通知后，通过订单里的商品需求，扫描仓库里相应货物的条形码进行配货操作，并打印配货单。

(3) 审核配货结果并选择司机

仓库管理员审核理配员的配货单是否与订单里的商品信息一致，若不一致返回 (2) 重新配货，若一致，仓库审核员选择司机，安排发货。

(4) 司机发货；

(5) 跟单员收到货物，订单结束。

3.2.5.2 入库管理

只有企业“存”货时，才会出现入库的情况。所以，该功能只涉及 5 种订单：供货商进货单、经销商退货单、手术寄售回收单、借入单和借出归还单。入库操作的步骤如下：

(1) 选择理配员

以上 5 种订单通过订单审核并分配好仓库后，会将订单权限移至相应仓库的仓库管理员处，由仓库管理员指定对该订单进行入库操作的理配员。

(2) 入库

理配员收到入库通知后，扫描订单里随行货物的条形码，以实际货物情况入库并打印配货单。

(3) 审核入库结果

仓库管理员审核理配员的入库单是否与实际入库的商品数量及信息一致，若不一致返回 (2) 重新入库或与订单公司协调。

(4) 审核通过后入库成功，订单结束；

3.2.5.3 预配货管理

该子模块分为 3 个小部分：预配货模板、仓库查看和库存导入。

(1) 预配货模板

该部分主要需求分析见本文第 3.2.8 小节。

(2) 仓库查看

该部分主要是为了查看各大仓库的商品剩余库存。

(3) 库存导入

该部分主要是为了导入库存信息。

3.2.6 报表管理模块

该模块的功能是查看公司的各个业务的报表，主要是供监管部门查看。分为进货管理、销售管理、库存管理和统计分析 4 个子模块，前 3 个模块里都是

存放着企业的信息，进货管理主要存储进货信息，销售管理主要是存储企业的销售信息，库存管理主要是存储企业的库存信息，统计分析子模块主要是对订单信息进行相应的需求数据分析。该模块用例图如图 3-6 所示：

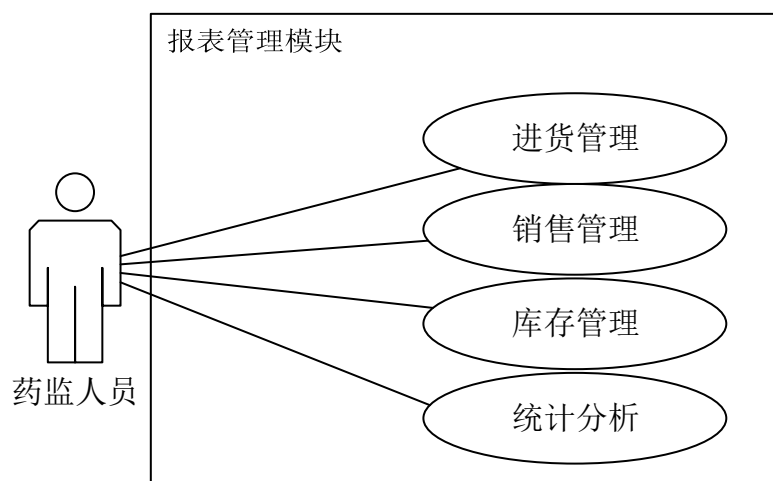


图 3-6 报表管理模块用例图

3.2.7 首营管理模块

该模块是为了实现系统有关首营的管理功能，也是食药监部门检查的模块，分为首营资质维护、首营品种维护、产品注册证维护、效期管理等子模块。该模块用例图如图 3-7 所示：

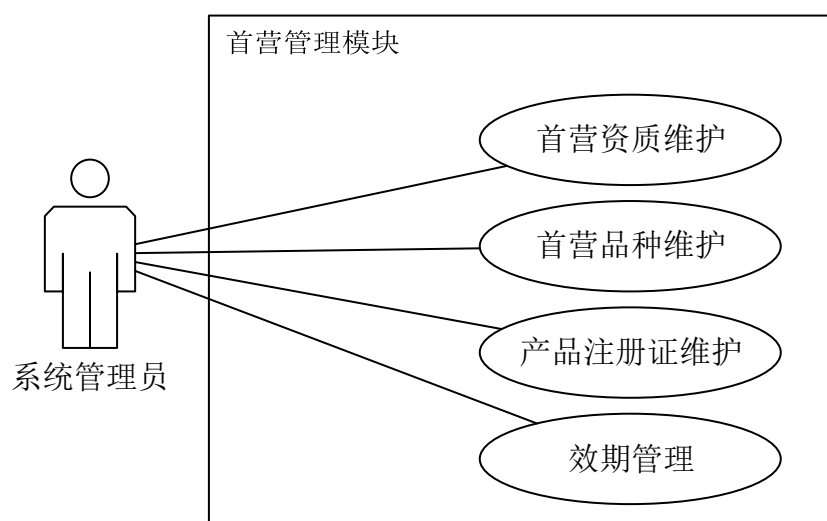


图 3-7 首营管理模块用例图

3.2.8 预配货模板功能模块

本功能模块属于仓库管理模块中的预配货管理子模块。随着企业订单数量越来越多，通过历史手术订单中的商品信息，分析出所有订单中长期搭配使用的医疗器械，设计成相应的预配货模板。系统管理员通过设计的预配货模板信息，在系统中对预配货模板进行添加、删除、修改等操作。

仓库理配员通过预配货模板信息，对仓库的货物进行装箱配货，形成大量匹配预配货模板的预配货箱子。每一个预配货箱子有一个条形码，扫描该条形

码可以得到箱子里所有货物的条形码，从而减少配货时对每个货物进行扫码的时间，提高仓库配货效率。除去提高配货效率，为了减少预配货箱子的占用率，系统通过算法计算出每个预配货模板里对应器械的数量，使数量尽可能满足大部分订单需求，从而达到提高效率并减少占用的效果。

3.2.9 推荐预配货模板功能模块

本功能模块属于订单模块中手术寄售单子模块。在创建手术寄售单选择医生的预配货模板时，选择完手术部位后，通过分析该部位的历史订单中医生对预配货模板的评分结果，给客户医院的医生推荐效果最佳和性价比最高的两种预配货模板，从而给客户进行多种选择。

每一次手术寄售回收单会让医生为本次使用器械进行打分，通过分析所有回收单的打分，根据手术部位不同，每个手术部位所用的医疗器械模板评出一款大部分医生认为效果最佳的模板，在每一次销售人员下单时进行推荐。除此之外，还希望能通过分析模板的性价比，从而根据不同手术部位推荐一款性价比最高的模板。将这两款模板推荐至选择医生模板的页面中，供客户医院的医生进行选择。

3.3 非功能性需求分析

（1） 规范性

整个系统的各种软件、硬件均应符合相关的国际国内标准，以保证系统与其它业务系统能实现有效的连接。

（2） 开放性

提供标准的接口，便于其他系统与本系统的对接；便于与其他系统的数据交换；包括：身份认证接口、界面集成接口及信息数据接口。

（3） 先进性

采用先进的基于计算机网络的软件、硬件产品及模块化的软硬件设计，保证系统在技术上的领先。

（4） 实用性

充分考虑服务平台系统的建设方向和特点，在满足当前业务需求的情况下，依据目前的用户规模、业务运营情况和应急的服务需求，在保证系统运行可靠的前提下，设计与规模、软件功能和业务功能相适应的系统。

（5） 可靠性

整个系统应采用多种手段，保证系统的高可靠性和稳定性，保证系统的正常运行。

（6） 扩展性^[26]

软件、硬件平台在系统容量、处理能力和业务应用等方面应具有良好的可扩展能力，能够方便地进行系统升级和更新，以适应业务的不断发展。

3.4 本章小结

本章从软件工程的标准流程出发，简单介绍了系统的所有模块以及所有模块对应的业务逻辑需求，并阐述了系统的功能性需求和非功能性需求。

第四章 TMDSCP 详细设计

本章在第三章需求分析的基础上，对第三方医疗器械供应链平台进行详细的设计，包括系统功能模块设计、系统架构的设计及数据库设计等。本章中的设计充分考虑了软件的易理解性、可拓展性、可重用性与可维护性，坚持面向对象的设计原则^[27]，详细地描述了整个平台的设计过程。

4.1 系统总体设计与模块设计

4.1.1 系统总体设计

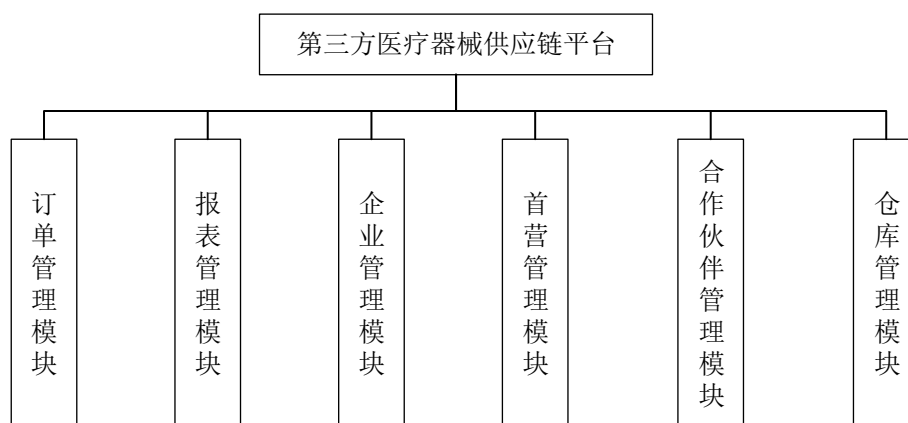


图 4-1 系统总体结构

如图 4-1 所示，第三方医疗器械供应链平台主要由订单管理模块、报表管理模块、企业管理模块、首营管理模块、合作伙伴管理模块和仓库管理模块组成。

系统整体业务主要是对于企业订单的管理，由于订单分为供货商订单、经销商订单、手术寄售单及借调单这 4 个大类，而这 4 个大类涉及入库和出库两种类型，入库类型的订单主要有：供货商进货订单、经销商退货订单、手术寄售回收单、借入单和借出归还单，出库类型的订单主要有：供货商退货订单、经销商进货订单、手术寄售单、借出单和借入归还单。

由于所有需要入库的订单以及所有需要出库的订单大致都相同，故入库以供货商进货订单为例，出库以手术寄售单为例。供货商进货单的流程图如图 4-2 所示，手术寄售单的流程图如图 4-3 所示：

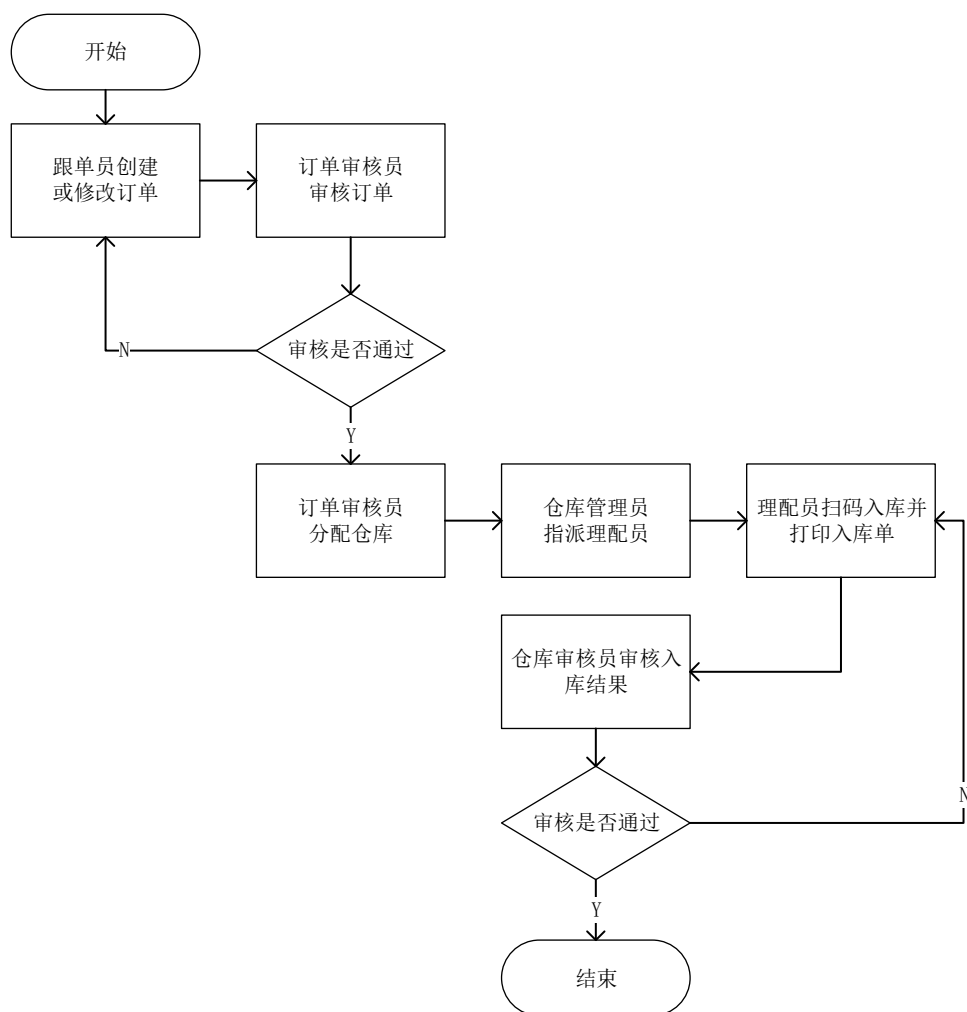


图 4-2 系统整体业务流程图（供货商进货）

由图 4-2 可知，系统供货商进货单是整个业务流程首先是由跟单员创建订单，然后交由订单审核员对本订单进行审核，若订单有疑问则驳回此订单，返至跟单员处，由跟单员根据驳回原因重新修改本订单的相关信息，若订单审核成功则分配仓库至相应仓库管理员处，然后相应仓库的仓库管理员指派理配员进行扫码入库操作，理配员入库完成后打印入库单交由仓库管理员审核入库结果，若审核不通过，理配员重新进行入库操作，若审核通过，入库操作结束。

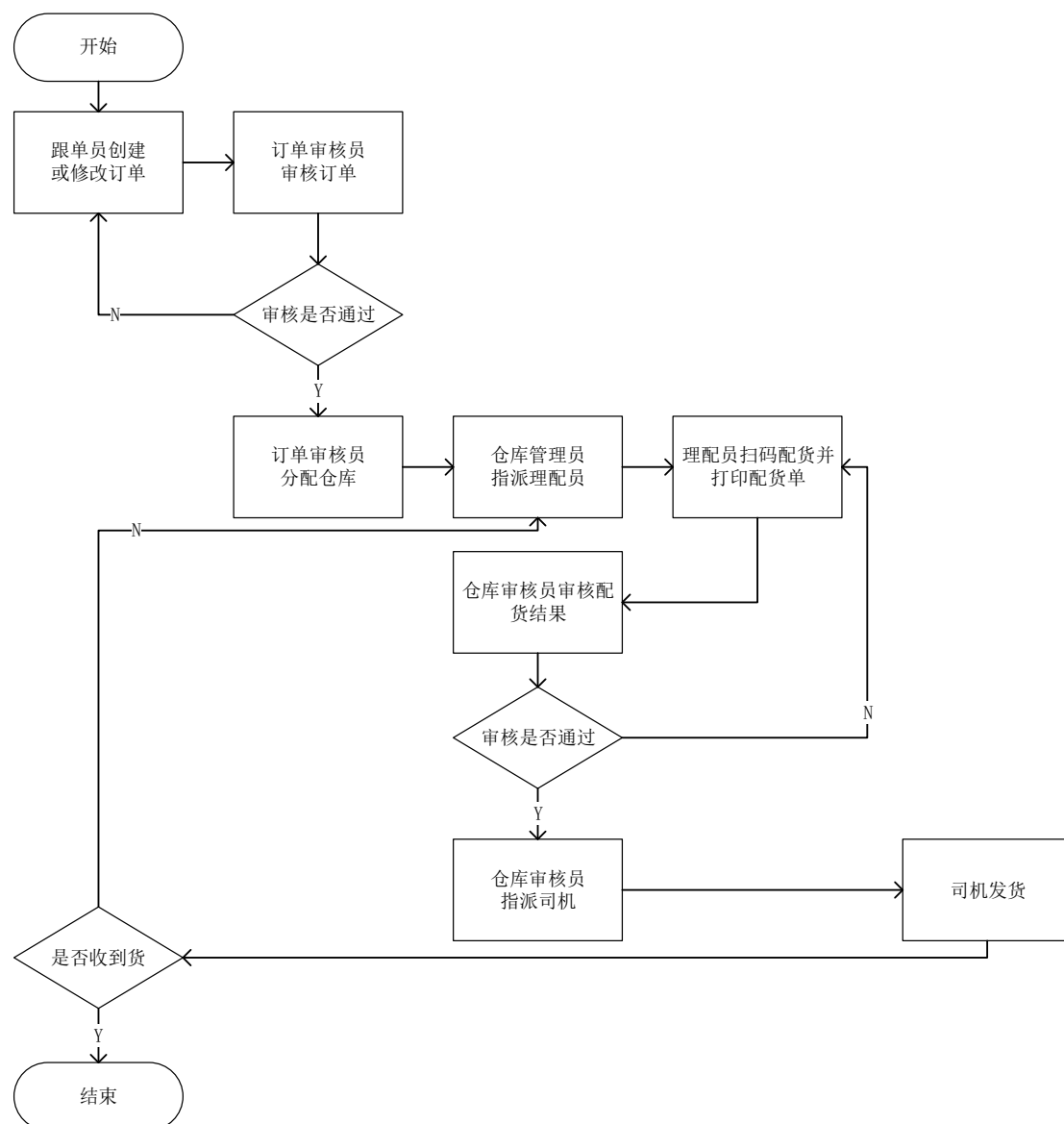


图 4-3 系统整体业务流程图（手术寄售）

由图 4-3 可知，系统手术寄售单的整体业务流程首先是由跟单员创建订单，然后交由订单审核员对本订单进行审核，若订单有疑问则驳回此订单，返至跟单员处，由跟单员重新修改本订单相关信息，若订单审核成功则分配仓库至相应仓库管理员处，然后相应仓库的仓库管理员指派理配员进行扫码配货操作，理配员配货完成后打印配货单交由仓库管理员审核配货结果，若审核不通过理配员重新进行配货，若审核通过仓库管理员指派司机进行发货，若未收到货再重新返回至仓库管理员处进行操作，若收到货本订单结束。

4.1.2 企业管理模块设计

该模块主要是在系统刚创建时，由系统管理员来操作的。分为部门/员工管理子模块、角色权限管理子模块、销售人员管理子模块、企业基本信息管理子模块、系统参数配置子模块。其框架图如图 4-4 所示：

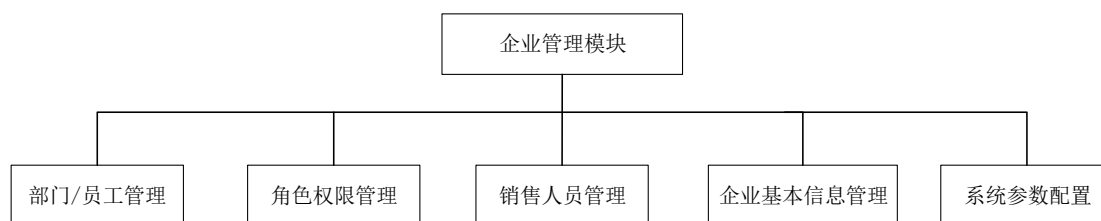


图 4-4 企业管理模块架构图

该模块中大部分功能均是本系统创建后的第一步，由系统管理员添加或修改的。

(1) 部门/员工管理

该子模块是由系统管理员负责，可进行创建、查看或删除部门及相应部门的员工及其相关信息，从而创建本系统所有的用户；

(2) 角色权限管理

该子模块的功能是由系统管理员创建本系统中的所有角色，主要是为了区分本平台的所有角色，并在创建员工信息时便于添加员工角色，从而达到分工明显，各司其职的效果；

(3) 销售人员管理

该子模块是由系统管理员将每一个客户公司分配一些销售代表，这些销售代表负责某个或某些客户的销售业务；

(4) 企业基本信息管理

该子模块功能是让系统管理员将企业的基本信息如企业营业执照、企业机构名称及经营范围等录入系统；

(5) 系统参数配置

该子模块是为了配置本系统中常见的一些参数，比如企业里所有的医疗器械的产品品牌和产品类型，以及医疗器械的过期天数等，从而可以通过过期天数产生效期预警。

4.1.3 合作伙伴管理模块设计

该模块也主要是由系统管理员创建，其他角色成员进行查看的模块，分为合作伙伴子模块和商品信息子模块。其框架图如图 4-5 所示：

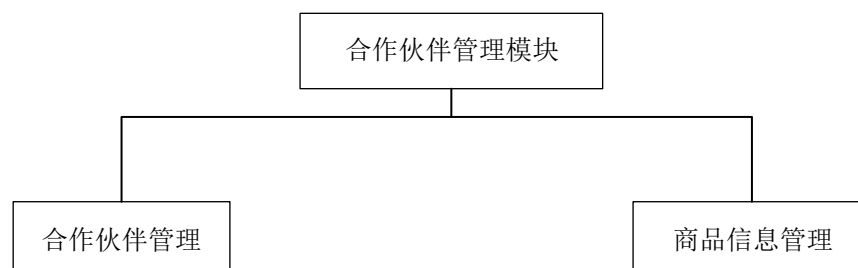


图 4-5 合作伙伴管理模块架构图

其中合作伙伴子模块包括与企业已签约公司的合同信息、与企业合作的医院的地址和医生的科室及医生的其他信息等；商品信息子模块是将仓库里所有的商品分门别类，从而方便进行查看某一类商品的详细信息。

4.1.4 订单管理模块设计

订单管理模块主要包括供货商订单管理、经销商订单管理、手术寄售单管理、借调单管理和订单审核五个功能模块。其框架图如图 4-6 所示：

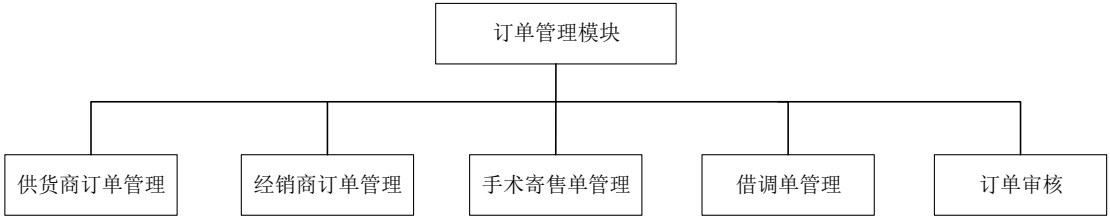


图 4-6 订单管理模块架构图

由图 4-2 和图 4-3 可知，订单模块所有子模块的步骤都是销售人员先填写订单信息，然后交给订单审核员审核，若审核通过，由订单审核员来分配仓库，并将相应的信息传递给所分配仓库的仓库管理员，若审核不通过，订单审核员将驳回此单并填写驳回意见返回至创建该订单的销售人员处。

(1) 创建订单或修改订单

首先，销售人员创建/修改订单，填写订单信息及商品信息（手术寄售单还需填写手术信息），填写完成后保存订单，系统返回保存结果并将此订单状态设置为“未审核”。

(2) 审核订单

完成创建/修改订单后，订单状态变为“未审核”，交由订单审核员审核，若审核通过，审核员会在提交审核结果时分配仓库，将订单交由分配的仓库管理员处，且订单状态变为“已审核”，若审核驳回，提交驳回意见，该订单返回至原销售人员处，订单状态变为“已驳回”。

结合图 4-2 和图 4-3 可知，所有类型订单的流程均一致，故订单模块的活动图如图 4-7 所示：

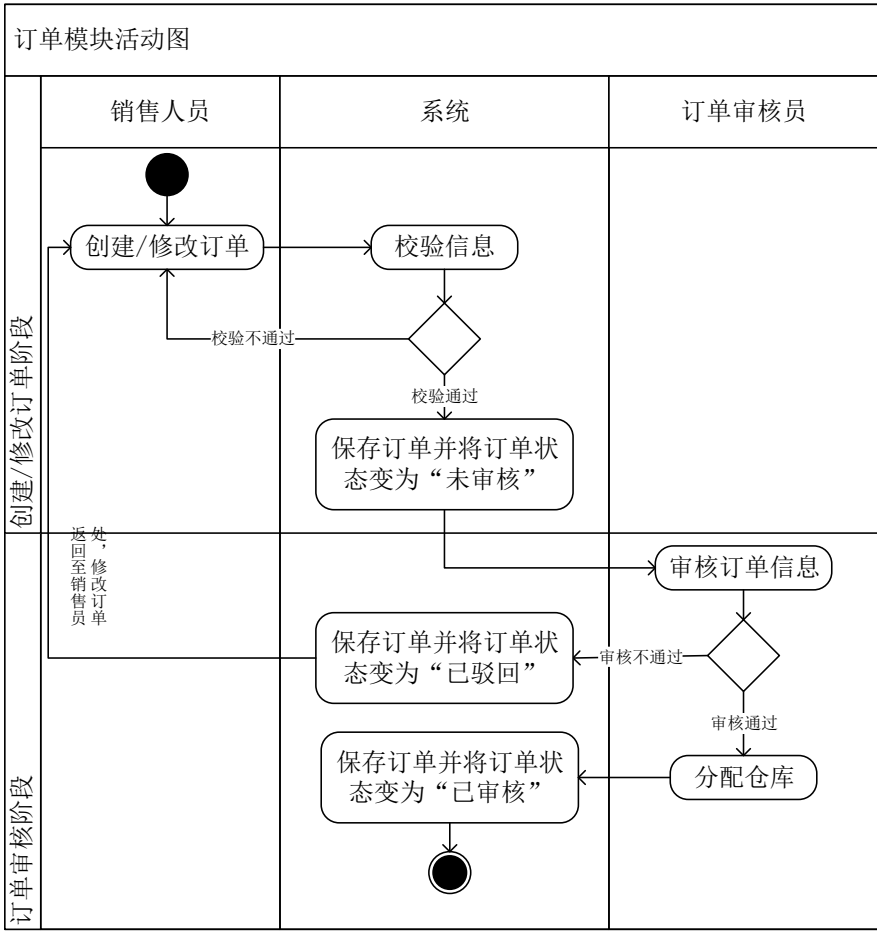


图 4-7 订单模块活动图

4.1.5 仓库管理模块设计

本小节将对仓库管理模块进行设计，因为“进-销-存”是一体的，所以仓库管理模块会与订单模块有联系。仓库管理模块如图 4-8 所示，分为入库子模块、出库子模块和预配货管理子模块，本文第 3.2.5 小节已介绍与入库相关的订单和与出库相关订单是哪些，这里便不作赘述，下面将分开描述仓库管理的三个模块。如图 4-8 所示：



图 4-8 仓库管理模块架构图

(1) 入库模块

由图 4-2 可知，仓库管理员收到订单审核员发送过来的订单信息后，将指派理配员进行入库的操作，理配员通过扫描货物条码入库并打印入库单，仓库管理员通过入库单以及入库的货物进行入库审核，审核通过入库结束，若审核不通过，订单再返回至理配员处，重新进行入库操作。

(2) 出库模块

由图 4-3 可知，当订单审核员分配好相应仓库后，对应的仓库管理员开始指派理配员，理配员开始扫描货物进行出库操作，并打印随行的出库单，然后仓库管理员审核出库单的信息，若审核不通过，直接将订单返回至理配员处重新理配货物并打印新的出库单，若审核通过，仓库管理员指派司机配送货物至客户处，最后负责该订单的销售人员确认收货，订单结束。

由于出入库流程除去出库需分配司机外，大致都相似，故出/入库模块的活动图如图 4-9 所示：

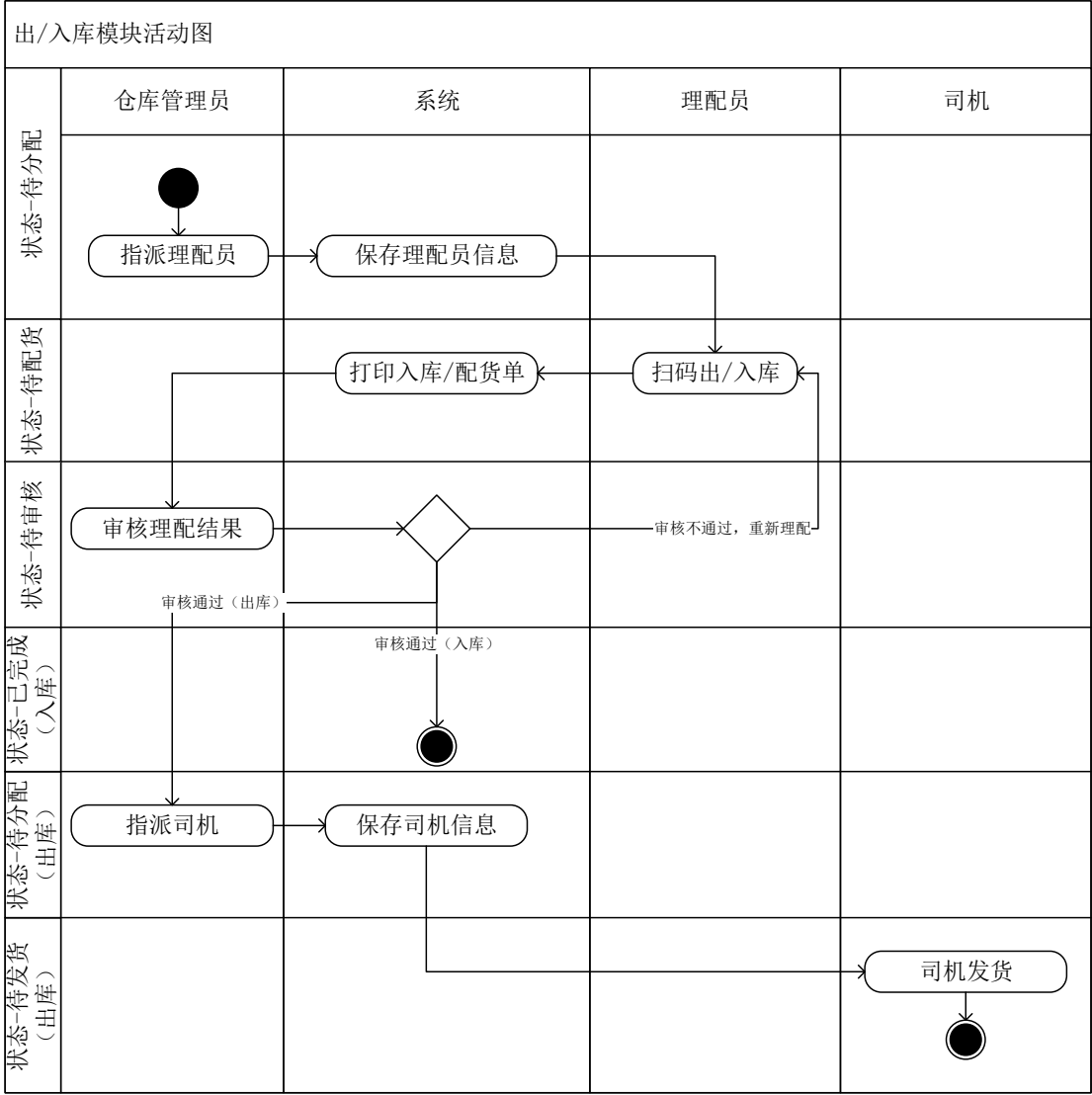


图 4-9 出入库模块活动图

(3) 预配货管理子模块

此模块是由仓库管理员或系统管理员控制的，首先通过分析历史订单，计算出经常搭配使用的医疗器械，并为其设计相应的预配货模板，在此子模块中可以添加和查看所设计的预配货模板，还可以通过预配货管理子模块中的仓库查看模

块来查看企业及其合作伙伴仓储模式，选择仓库，可以进入不同合作伙伴的仓库进行查看仓库库存等信息。

4.1.6 首营管理模块设计

该模块主要是由系统管理员操作的，分为首营资质维护子模块、首营品种维护子模块、产品注册证维护子模块及效期管理子模块。其框架图如图 4-10 所示：

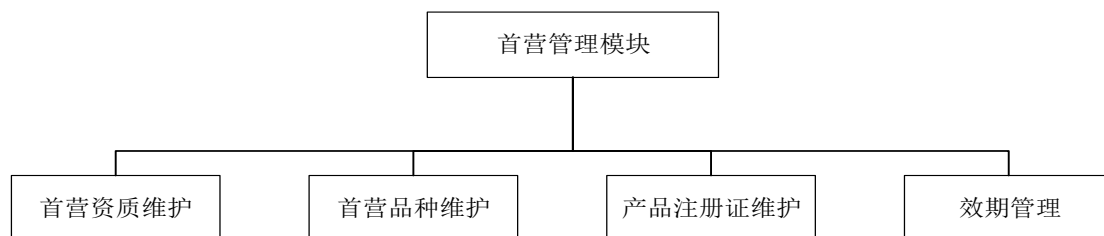


图 4-10 首营管理模块框架图

这些子模块记录了与企业有关的公司及企业自身的合法性证明，以及企业内部品种和产品注册证等合法性证明，并通过效期管理子模块来设置企业仓库中医疗器械的有效期为该器械设置预警。

4.1.7 报表管理模块设计

该模块主要是方便企业进行信息记录及监管部门的查看，包括进货管理子模块、销售管理子模块、库存管理及统计分析子模块。前 3 个子模块的功能都是为了将相对应的数据进行整理，从而形成一系列报表，并显示该类型报表；统计分析子模块主要是对大量的订单信息进行分析，从而获取相应的信息，如分析月/年销量最高、月/年利润最高的预配货模板等。以月销量最高为例，流程图如图 4-11 所示：

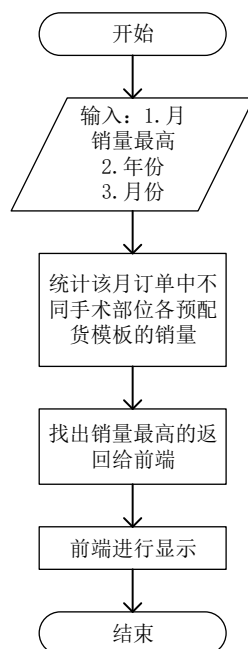


图 4-11 统计分析-月销量最高功能模块流程图

4.1.8 基于 Apriori 算法的预配货模板

本功能模块通过分析大量相同手术部位的订单信息，发现大部分订单会搭配使用一些固定的医疗器械。经过分析与整理每一笔订单里经常搭配使用的医疗器械，将这些医疗器械组合起来，形成一个预配货模板。本文主要通过 3 个步骤设计企业预配货模板：（1）给每种器械进行编号，通过关联规则计算出搭配使用的医疗器械的编号；（2）通过泊松分布计算出相应器械的数量在所有订单中出现的大概率区间；（3）通过关联规则分析各器械数量的大概率区间是否相关，若相关，则各器械的数量取能够满足大部分订单的数量；若无关，则取其中一个器械的大概率区间，然后分析出与它相关器械的取值区间，最后取这些区间里的中位数为预配货模板里对应器械的数量。

4.1.8.1 利用 Apriori 算法找出搭配使用的器械编号

（1）本文中的项指的是 m 种器械的编号，即 $1 \sim m$ ，故项集 $I = \{1, 2, \dots, m\}$ ；每笔交易称为一个事务 T 。比如若第 i 笔交易中使用了 1 号和 3 号器械，则事务数据库 D_1 中 $T_i = \{1, 3\}$ ，设置最小支持度阈值（ \min_sup ）为 0.8。事务数据库 D_1 相应的二元 0/1 表示如表 4-1 所示：

表 4-1 事务数据库 D_1 中的二元 0/1 表示

| TID | 1 | 2 | 3 | | m |
|-------|-------|-------|-------|-------|-------|
| T_1 | 1 | 0 | 0 | | 1 |
| T_2 | 1 | 1 | 1 | | 0 |
| | | | | | |
| T_i | 1 | 0 | 1 | | 0 |
| | | | | | |
| T_n | 0 | 1 | 0 | | 0 |

（2）由本文 2.4 节可知，通过计算使得 $\text{support}(A) = \frac{D \text{中包含项集} A \text{的事务数}}{D \text{中事务总数}} \geq \min_sup$ ，即 A 为频繁项集，使用 Apriori 算法逐层的迭代计算项集 I 在事务数据库 D_1 中的频繁 1 项集，频繁 2 项集，...，频繁 k 项集，每次迭代计算结束后通过先验性质剔除部分项集，得到最终频繁 k 项集 L_k 。

（3）计算 $L_k = \{l_1, l_2, \dots, l_k\}$ 里的每一个频繁项 $l_i (i \in [1, k], \text{且} i \text{为整数})$ 是否与其他项满足强关联规则，删除不满足强关联规则的项，最终得到的 L_k 中的每一项即表示一个预配货模板中所包含器械的编号。

4.1.8.2 利用泊松分布和 Apriori 算法得出每个模板中所有器械的数量

假设通过本文第 4.1.8.1 小节已找出所有订单中搭配使用的器械组合，以某个预配货组合 $l_i (i \in [1, k], \text{且} i \text{为整数}) = \{x, y, z\}$ 为例。

1. 利用泊松分布^[28]计算出模板中器械数量的大概率分布区间

分别计算出当 x, y 和 z 三种类型器械进行组合时, 每种器械数量的概率分布:

(1) 泊松分布介绍

设随机变量 X 所有可能取值为 0,1,2,...,而取各个值的概率是:

$$P\{X = k\} = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, 2, \dots \quad (4-1)$$

其中 $\lambda > 0$ 是常数。则称 X 是服从参数 λ 的泊松分布, 记为 $X \sim \pi(\lambda)$;

(2) 统计包含该种组合的所有订单中 x 的数量, 设为数组 $C = [x_1, x_2, \dots, x_n]$, 取 C 中所有数的平均值 $\bar{X} = \frac{x_1 + x_2 + \dots + x_n}{n}$;

(3) 将 $\bar{X} \approx \lambda$, 则关于 x 编号的器械的泊松分布为:

$$P_x\{X = k\} = \frac{\bar{X}^k e^{-\bar{X}}}{k!}, \quad k = x_1, x_2, \dots, x_n \quad (4-2)$$

(4) 求出当 $P_x\{X = k\} \geq 0.8$ 时的所有 x 的取值, 该范围为器械 x 在所有订单中的取值范围, 记为 $[x_i, x_j]$;

(5) 同理计算可得器械 y 和 z 在所有订单中的出现概率大于等于 80% 时的取值范围, 记为 $[y_i, y_j]$ 和 $[z_i, z_j]$;

2. 利用 Apriori 算法计算出其余器械数量的取值

(1) 将所有同时出现 x, y, z 的订单设成新的事务数据库 D_2 , 器械 x, y 和 z 的个数为每个事务 $T_i = \{a, b, c\}$, i 为某个订单, a、b 和 c 分别对应器械 x, y 和 z 在每个订单的数量;

(2) 设置最小支持度阈值 (min_sup) 为 80%。若事务 T_i 能够满足 $a \in [x_i, x_j]$, $b \in [y_i, y_j]$, $c \in [z_i, z_j]$ 则 $T_i = \{a, b, c\}$, 事务数据库 D_2 相应的二元 0/1 表示如表 4-2 所示, :

表 4-2 事务数据库 D_2 中的二元 0/1 表示

| TID | a | b | c |
|-------|-------|-------|-------|
| T_1 | 1 | 0 | 0 |
| T_2 | 1 | 1 | 1 |
| | | | |
| T_i | 1 | 1 | 1 |
| | | | |
| T_n | 0 | 1 | 0 |

(3) 设新的项集 $I_1 = \{a, b, c\}$, 通过 Apriori 算法确认该项集 I_1 是否能够满足 $\text{support}(a \Rightarrow b) \geq 0.8$ 以及是否满足 $\text{support}(a \Rightarrow c) \geq 0.8$, 若满足则通过上述泊

松分布求出 $P_x\{X = k\} \geq 0.9$ 时，将每次计算得出 x 的最小值分别赋给 a, b, c 成为模板中器械 x, y, z 分别对应的数量。若不满足进行第（5）步；

（4）通过 Apriori 算法求出满足 $\text{support}(a \Rightarrow b) \geq 0.8$ 和 $\text{support}(a \Rightarrow c) \geq 0.8$ 的 b 和 c 的区间。将该区间的中位数作为器械 y 和 z 对应的数量。

4.1.9 推荐预配货模板功能模块设计

推荐预配货模板功能模块主要是用于订单管理模块的手术寄售单中，在每一次创建订单选择医生模板时进行推荐。

通过分析手术寄售回收单的历史信息，根据手术部位不同，通过分析手术回收寄售单中客户医院的医生对本次预配货医生模板的评分（总分 10 分），找出大部分医生认为效果最好的模板即评分最高的模板，以及在效果不差的情况下（评分不低于 6 分）价格最低的预配货模板。从而在每一次创建新的手术寄售单时对医生进行模板推荐：1.大部分医生认为效果最好的模板；2.性价比最高的模板。

本功能模块的流程图如图 4-12 所示，首先销售人员创建了新的手术寄售单，输入本次订单的手术部位后，系统通过分析历史记录中相同手术部位的订单，如果是第一次创建该手术部位的订单，则推荐空，若不是第一次，则找到历史数据中的回收订单里的所有用户评分和售价，将该手术部位评分最高的即效果最佳的模板和性价比最高的模板推荐给用户。

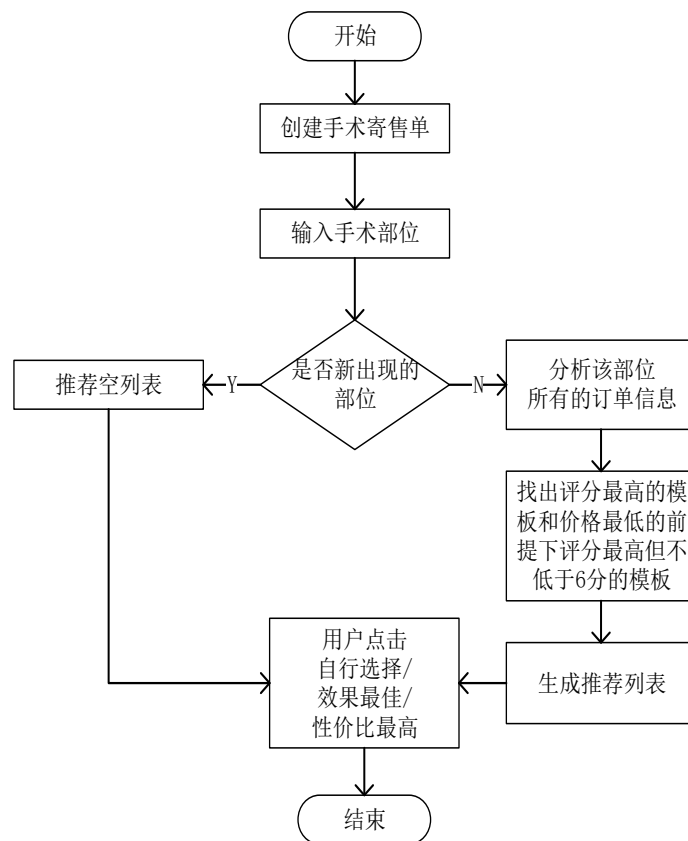


图 4-12 推荐预配货模板功能模块流程图

4.2 系统架构设计

本文第二章已详细介绍了目前前端及后端框架的现状以及一些框架的对比，考虑到系统的开发成本、可维护性以及可拓展性，本项目的移动端部分和 PC 端部分均采用 B/S 架构，遵循 MVC 设计思想，并对系统做了详细划分。因此本系统主要包括表示层，业务逻辑层和数据层。所设计的系统技术架构如图 4-13 所示：

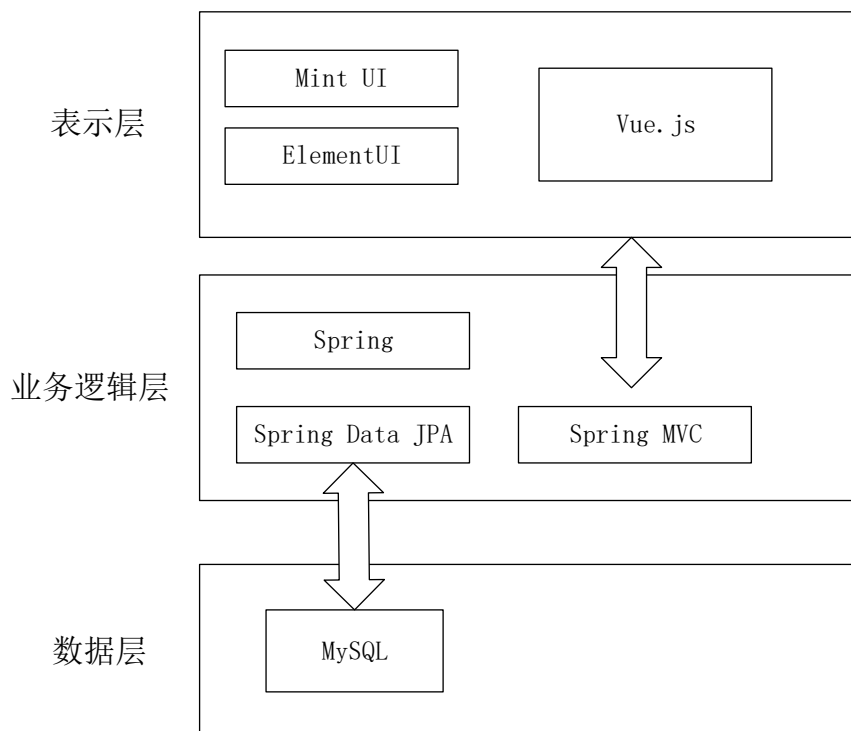


图 4-13 系统架构设计图

表示层主要是为了与用户进行交互，为了简化项目开发，PC 端部分与移动端部分均使用 B/S 架构，无论移动端部分还是 PC 端部分，用户都是主要通过浏览器与平台进行交互，移动端部分为了适应手机屏幕大小使用 Mint UI 使界面更加友好，PC 端部分则使用 Element UI 改善用户体验，并搭配流行框架 Vue 加速开发和扩展。

表示层通过 AJAX 异步地将请求数据传到业务逻辑层，业务逻辑层负责处理表示层中用户行为产生的请求，并将响应结果返回至表示层进行前端页面渲染^[29]。业务逻辑层主要通过 Spring、Spring MVC 与 Spring Data JPA 技术实现，其中 Spring MVC 负责控制前端请求的分配，将从前端传过来的数据进行 JSON 解析后，利用 Spring 的依赖注入技术@Autowired 进行业务逻辑处理，最后通过 Spring Data JPA 调用 Repository 层接口与数据层 Mysql 数据库进行交互。

4.3 数据库设计

4.3.1 数据库概念设计

分析了需求里各模块后，设计了本系统涉及的实体、属性以及实体之间的关系，这些数据模型结构如下：

(1) 企业管理 E-R 模型

企业管理 E-R 模型如图 4-14 所示：

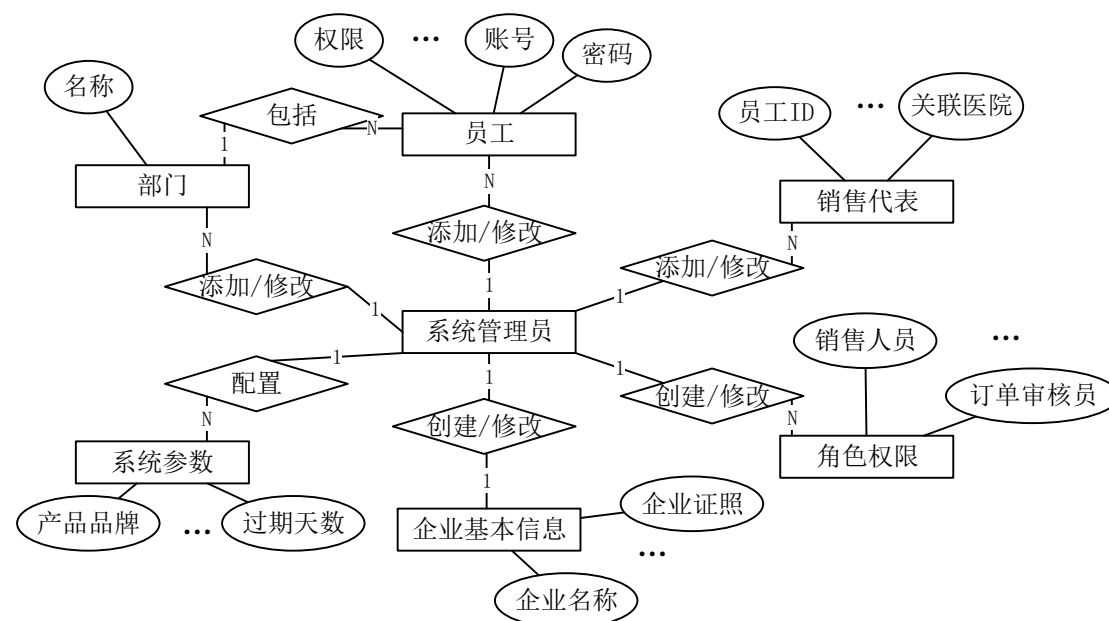


图 4-14 企业管理 E-R 模型

(2) 合作伙伴管理 E-R 模型

合作伙伴管理 E-R 模型如图 4-15 所示：

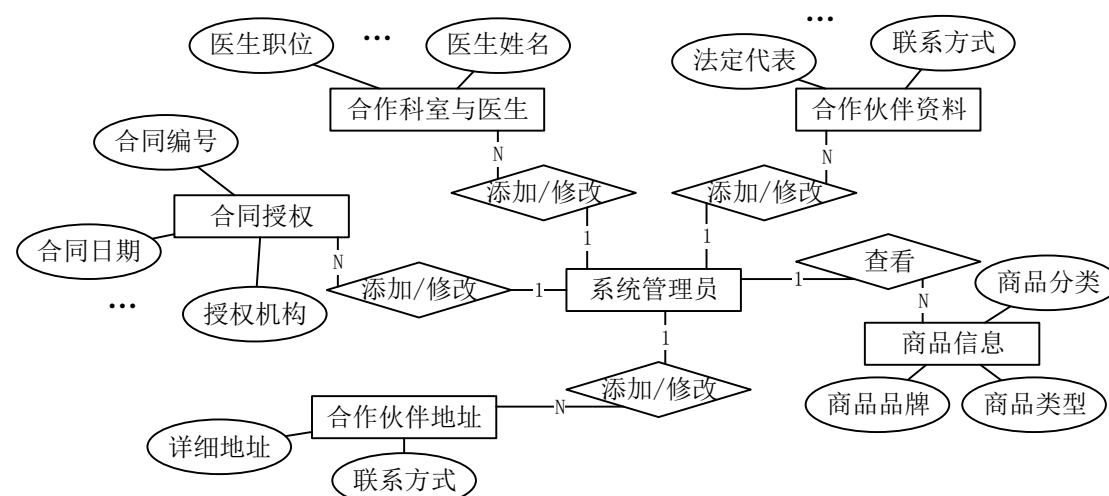


图 4-15 合作伙伴管理 E-R 模型

(3) 订单管理 E-R 模型

订单管理 E-R 模型如图 4-16 所示:

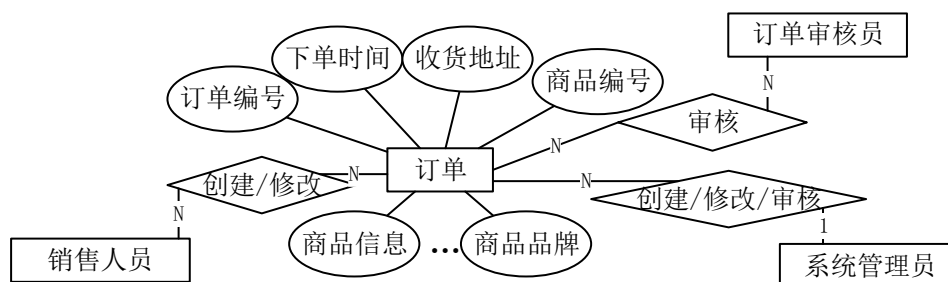


图 4-16 订单管理 E-R 模型

(4) 仓库管理 E-R 模型

仓库管理 E-R 模型如图 4-17 所示:

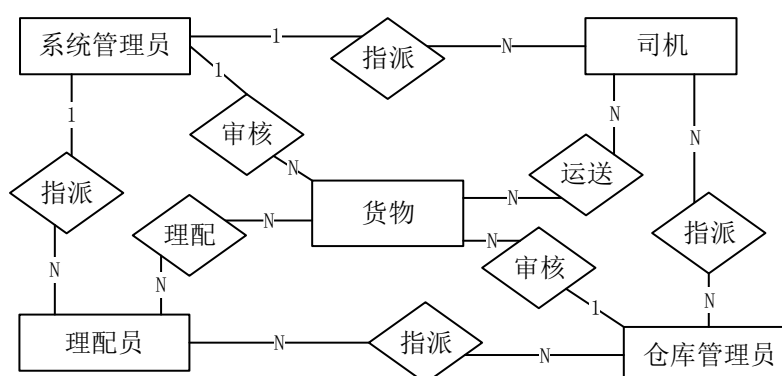


图 4-17 仓库管理 E-R 模型

(5) 首营管理 E-R 模型

首营管理 E-R 模型如图 4-18 所示:

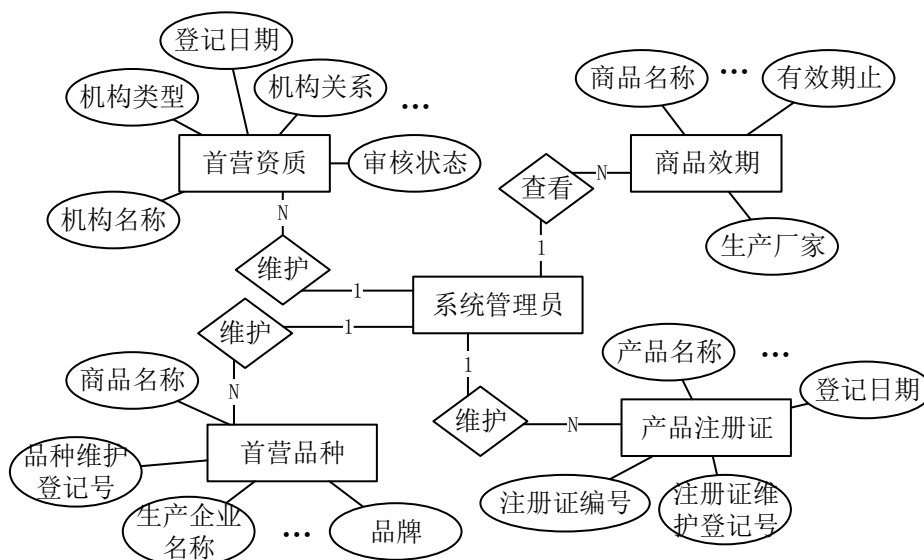


图 4-18 首营管理 E-R 模型

4.3.2 数据库表设计

由于平台中需要建立多张数据库表, 根据第 4.3.1 小节所设计的模块数据库 E-R 模型, 本系统主要包含用户信息表, 手术寄售单基本信息表, 预配货模板基

本信息表，合作伙伴收发地址表，首营品种登记表，仓库库存数量表等 80 张表，以上述 6 张表为例，信息如表 4-3 所示：

(1) 用户信息表

表 4-3 用户信息表 user_info

| No. | 字段名 | 字段含义 | 类型（长度） | 非空 | 主键 |
|-----|------------|-------|--------------|----|----|
| 1 | id | 编号 | int(11) | √ | √ |
| 2 | remark | 备注 | varchar(255) | | |
| 3 | account | 账户 | varchar(30) | | |
| 4 | address | 地址 | varchar(255) | | |
| 5 | age | 年龄 | int(11) | | |
| 6 | department | 部门 | varchar(30) | | |
| 7 | job | 职责 | varchar(30) | | |
| 8 | loginip | 上线 IP | varchar(30) | | |
| 9 | login_time | 上线时间 | datetime | | |
| 10 | mail | 邮箱 | varchar(255) | | |
| 11 | name | 姓名 | varchar(20) | √ | |
| 12 | password | 密码 | varchar(255) | √ | |
| 13 | sex | 性别 | int(11) | | |
| 14 | tele | 电话 | varchar(20) | | |
| 15 | company | 公司 | varchar(255) | | |

(2) 手术寄售单基本信息表

由于订单模块中所有订单基本信息类似，故将最复杂的手术寄售单为例，如表 4-4 所示：

表 4-4 手术订单表 surgery_order_info

| No. | 字段名 | 字段含义 | 类型（长度） | 非空 | 主键 |
|-----|----------------|------|--------------|----|----|
| 1 | id | 编号 | int(11) | √ | √ |
| 2 | admission_no | 住院号 | varchar(255) | | |
| 3 | arrive_time | 到货时间 | datetime | | |
| 4 | bed_no | 床位号 | varchar(255) | | |
| 5 | create_time | 开单时间 | datetime | | |
| 6 | current_status | 当前状态 | int(11) | | |
| 7 | doctor | 主刀医生 | varchar(255) | | |
| 8 | hospital_name | 医院名称 | varchar(255) | | |
| 9 | is_delete | 是否删除 | int(11) | | |

| | | | | | |
|----|------------------------|------|--------------|---|--|
| 10 | is_returned | 是否回收 | int(11) | | |
| 11 | order_applicant | 下单员 | varchar(255) | | |
| 12 | order_id | 订单编号 | varchar(20) | √ | |
| 13 | order_type | 订单类型 | int(11) | | |
| 14 | patient_name | 病人姓名 | varchar(255) | | |
| 15 | patient_sex | 病人性别 | int(11) | | |
| 16 | product_brand | 产品品牌 | varchar(255) | | |
| 17 | product_line | 产品线 | varchar(11) | | |
| 18 | product_type | 产品类型 | varchar(255) | | |
| 19 | remark | 备注 | varchar(255) | | |
| 20 | sale_type | 销售类型 | varchar(11) | | |
| 21 | sales_delegate | 销售代表 | varchar(255) | | |
| 22 | sales_delegate_address | 联系方式 | varchar(255) | | |
| 23 | surgery_name | 手术名称 | varchar(255) | | |
| 24 | surgery_part | 手术部位 | varchar(255) | | |
| 25 | surgery_template_id | 医生模板 | varchar(255) | | |
| 26 | surgery_time | 手术时间 | datetime | | |

(3) 预配货模板基本信息如表 4-5 所示。

表 4-5 预配货模板基本信息表 prematch_template_info

| No. | 字段名 | 字段含义 | 类型（长度） | 非空 | 主键 |
|-----|--------------------|-----------|--------------|----|----|
| 1 | id | 编号 | int(11) | √ | √ |
| 2 | productline | 商品产品线 | varchar(20) | √ | |
| 3 | productbrand | 商品品牌 | varchar(255) | √ | |
| 4 | producttype | 商品类型 | varchar(255) | √ | |
| 5 | templatecreator | 预配货模板创建人 | varchar(255) | | |
| 6 | templatecreatetime | 预配货模板创建时间 | datetime | | |
| 7 | remark | 备注 | varchar(255) | | |
| 8 | templateid | 预配货模板编号 | varchar(255) | √ | |
| 9 | templatename | 预配货模板名称 | varchar(255) | √ | |
| 10 | totalprice | 销售总价 | double | | |
| 11 | costprice | 进价 | double | | |

(4) 合作伙伴收发地址信息如表 4-6 所示。

表 4-6 合作伙伴收发地址表 address_info

| No. | 字段名 | 字段含义 | 类型（长度） | 非空 | 主键 |
|-----|-----|------|--------|----|----|
|-----|-----|------|--------|----|----|

| | | | | | |
|---|-------------|-------|--------------|---|---|
| 1 | id | 编号 | int(11) | √ | √ |
| 2 | add_type | 收发货类型 | varchar(20) | | |
| 3 | area | 地区 | varchar(255) | | |
| 4 | detail_add | 地址详情 | varchar(255) | | |
| 5 | linkman | 联系人 | varchar(255) | | |
| 6 | linkman_tel | 联系人电话 | varchar(255) | | |
| 7 | authnum | 机构编号 | varchar(255) | | |
| 8 | authbody | 机构名称 | varchar(255) | | |

(5) 首营品种登记表如表 4-7 所示。

表 4-7 首营品种登记表 product_main_info

| No. | 字段名 | 字段含义 | 类型（长度） | 非空 | 主键 |
|-----|---------------------|-----------------------------|--------------|----|----|
| 1 | id | 编号 | int(11) | √ | √ |
| 2 | leaderauthorization | 法人授权委托书 | int(11) | | |
| 3 | checkbook | 检验报告书 | int(11) | | |
| 4 | datefrom | 有效期始 | varchar(255) | | |
| 5 | dateto | 有效期止 | varchar(255) | | |
| 6 | instruction | 说明书(0:有,1:无) | int(11) | | |
| 7 | mainfunction | 主要功能 | varchar(255) | | |
| 8 | manufact | 生产厂家 | varchar(255) | | |
| 9 | pictureurl | 上传图片地址 | text | | |
| 10 | productbrand | 商品品牌 | varchar(255) | | |
| 11 | productcharacter | 产品性质(0:有,1:无) | int(11) | | |
| 12 | productlicense | 生产企业器械许可证 | varchar(255) | | |
| 13 | productline | 业务线 | varchar(255) | | |
| 14 | productmaintainno | 品种维护登记号 | varchar(20) | √ | |
| 15 | productname | 商品名称 | varchar(255) | | |
| 16 | productpass | 产品合格证(0:有,1:无) | int(11) | | |
| 17 | productreg | 产品注册证号 | varchar(255) | | |
| 18 | productstandard | 产品执行标准 | varchar(255) | | |
| 19 | regtime | 登记日期 | datetime | | |
| 20 | remark | 备注 | varchar(255) | | |
| 21 | status | 状态(1:未提交,2:已提交,3:审核通过,4:审核驳 | int(11) | | |

| | | | | | |
|----|----------------|------------------|--------------|--|--|
| | | 回,5:审批通过,6:审批驳回) | | | |
| 22 | store | 储存要求 | varchar(255) | | |
| 23 | suppliercom | 供货企业名称 | varchar(255) | | |
| 24 | supplierid | 供货人身份证号 | varchar(255) | | |
| 25 | suppliername | 供货人姓名 | varchar(255) | | |
| 26 | supplierpermit | 供货企业器械许可证号 | varchar(255) | | |

(6) 仓库库存数量表如表 4-8 所示。

表 4-8 仓库库存数量表 warehouse_sto_number

| No. | 字段名 | 字段含义 | 类型（长度） | 非空 | 主键 |
|-----|----------------|-------------------|--------------|----|----|
| 1 | id | 编号 | int(11) | √ | √ |
| 2 | productid | 商品编号 | varchar(255) | √ | |
| 3 | batchnumber | 批号 | varchar(255) | √ | |
| 4 | effecttime | 有效期至 | datetime | | |
| 5 | productmodel | 商品规格（型号） | varchar(255) | √ | |
| 6 | stocknumber | 库存数量 | int(11) | √ | |
| 7 | warningnumber | 预警数量 | int(11) | √ | |
| 8 | productbrand | 商品品牌 | varchar(255) | | |
| 9 | productname | 商品名称 | varchar(255) | | |
| 10 | producttime | 生产日期 | datetime | | |
| 11 | productreg | 注册证号 | varchar(255) | | |
| 12 | suppliercom | 生产厂商 | varchar(255) | | |
| 13 | enterprisename | 公司名称 | varchar(255) | | |
| 14 | warehousename | 仓库名称 | varchar(255) | | |
| 15 | status | 状态(0:普通仓库,1:过期仓库) | int(11) | | |

4.4 本章小结

本章在第 3 章需求分析的基础上，对平台在技术架构和功能架构上对系统进行总的模块设计，并对这些模块进行更加详细的划分和设计，设计了每个模块 Mysql 的数据表，完成了对该平台的详细设计。

第五章 TMDSCP 实现与测试

本章节主要是以第三章需求分析和第四章详细设计为基础，实现本系统所有的功能，并对部分功能进行了测试。本文第 4.2 小节系统架构设计中的业务逻辑层的实现包括 4 个主要类：控制层类、业务逻辑类、持久对象类和数据持久类。该部分所有的具体实现的步骤均为：首先 Controller 接收前端的 RESTful 请求，然后 Controller 层去调用相应的 Service 服务接口（这里实际上是 Service 对应的 ServiceImpl 类来处理业务逻辑），接着 ServiceImpl 会调用数据持久化类 Repository，若是进行保存操作，则将数据持久化到数据库中即结束，若是获取数据的情况，那么 Repository 拿到数据库中数据后逐层返回给调用类。本章将详细介绍每个模块的实现：

5.1 系统主页面的实现

本系统为匹配企业业务流程设计了 PC 端与移动端，PC 端包含上述设计的所有模块，但移动端仅包含 PC 端部分业务。由于登陆页面较为简单，在此便不做实现部分的详细介绍，主要实现的控制层类为 UserInfoController.java，PC 端和移动端登陆的界面分别如图 5-1 和图 5-2 所示：

海王运和医疗器械管理系统

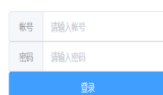


图 5-1 PC 端登陆界面



图 5-2 移动端登陆界面

本系统的系统管理员登录之后主页面的界面图在 PC 端和移动端分别如图 5-3 和 5-4 所示：

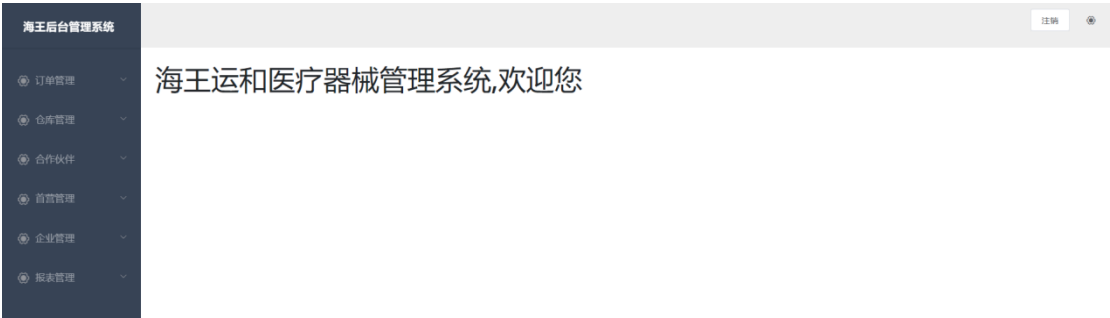


图 5-3 PC 端主页面界面



图 5-4 移动端主页面界面

由以上信息可知，本系统的移动端未包含企业管理模块、首营管理模块、报表管理模块及合作伙伴管理模块。

5.2 企业管理模块的实现

企业管理模块包括部门/员工管理、角色权限管理、销售人员管理、企业基本信息管理和系统参数配置 5 个子模块。该模块主要是为了便于管理企业内部信息及企业各种参数的配置，系统管理员在此处主要为企业内部管理添加各种信息，在此只介绍部门/员工管理子模块、角色权限管理子模块和系统参数配置子模块。

5.2.1 部门/员工管理

系统管理员通过部门/员工管理功能模块给企业添加部门及相应部门的员工账号，以便企业员工可以登录本系统。

本功能模块主要包含的类如表 5-1 所示：

表 5-1 部门/员工管理功能模块主要类

| | |
|-------|--|
| 控制层类 | UserInfoController.java DepartmentController.java MenuController.java |
| 业务逻辑类 | UserInfoService.java DepartmentService.java MenuService.java UserInfoServiceImpl.java DepartmentServiceImpl.java MenuServiceImpl.java |
| 持久对象类 | UserInfo.java Department.java Menu.java |
| 数据持久类 | UserInfoRepository.java DepartmentRepository.java MenuRepository.java |

在第一次登录系统时，系统管理员首先通过前端页面新建部门并输入部门名称，确认后由控制层的 DepartmentController 类中的 createDepartment 方法经过相应的 ServiceImpl 类处理，并将刚输入的部门数据经 DepartmentRepository 中同名方法保存至数据库中，随后通过前端页面新建该部门的员工，保存后由 UserInfoController 类中的 newUser 方法经 UserInfoServiceImpl 类处理，最终将新增员工的数据通过 UserInfoRepository 保存至数据库中。

若已添加了部分部门及员工，进入该功能模块时，会首先通过前端页面调用 DepartmentController 类中的 getDepartment 方法显示目前已有的部门，或是通过该类中的 deleteDepartment 方法删除已有的部门。点击进入相应部门后，将部门名称传给后台并调用 UserInfoController 类中的 getUserByName 方法，显示该部门所有的员工，若希望看到企业所有员工，则前端进行相应操作后，调用 UserInfoController 类中的 getUser 方法，若此时用户还需要添加部门或某部门的员工，系统依然是通过 createDepartment 方法和 newUser 方法处理。Menu 类在此处主要是为了在创建新员工设置该员工角色时调用该类中的 getMenuList 方法从而调用数据库中管理员已设置好的几种角色（即第 5.2.2 小节角色权限管理设置的角色）。

对创建新部门进行保存的部分代码如图 5-5 所示，创建新员工的界面如图 5-6 所示：

```
@GetMapping("/createDepartment/{name}")
public void createDepartment(@PathVariable String name){
    //将前端输入的部门名称传给name, 然后进行添加操作
    DepartmentInfo departmentInfo = new DepartmentInfo(name);
    departmentInfo.setNumber(0);
    departmentInfoService.save(departmentInfo);
}
```

图 5-5 创建新部门功能部分代码图



图 5-6 创建新员工功能界面图

5.2.2 角色权限管理

系统管理员通过角色权限管理子模块给本企业添加角色及角色的权限，以便管理员在新建员工时可以添加员工的角色，从而可以通过系统角色权限实现企业员工各司其职的功能。

本功能模块主要包含的类如表 5-2 所示：

表 5-2 角色权限管理功能模块主要类

| | |
|-------|--|
| 控制层类 | MenuController.java |
| 业务逻辑类 | MenuService.java MenuServiceImpl.java |
| 持久对象类 | Menu.java |
| 数据持久类 | MenuRepository.java |

本功能模块主要是系统管理员通过前端创建系统角色并为之分配权限，保存后调用后台 MenuController 类中的 createMenu 方法，最终将保存的角色及其所拥有的权限通过 MenuRepository 保存至数据库中。

对创建新角色进行保存的部分代码如图 5-7 所示，界面如图 5-8 所示：

```
@PostMapping("/createMenu")
public void createMenu(@RequestParam String id, @RequestParam String name, @RequestParam String detail) {
    //从前端获取了角色的id, 名称, 和其他细节信息
    Menu menu = new Menu();
    detail = detail.substring(0, id.length());
    menu.setMenuDetail(detail);
    id = id.substring(1, id.length() - 1);
    String array[] = id.split(regex: ";");
    menu.setName(name);
    for (String a : array) { //设置相关权限: 如 "setMenu1" 表示订单模块相关权限
        switch (a) { case "1":
            menu.setMenu1(true);
            break;
            case "2":
            menu.setMenu11(true);
            break;
            case "3":
            menu.setMenu111(true);
        }
    }
}
```

图 5-7 创建新角色权限功能界面图

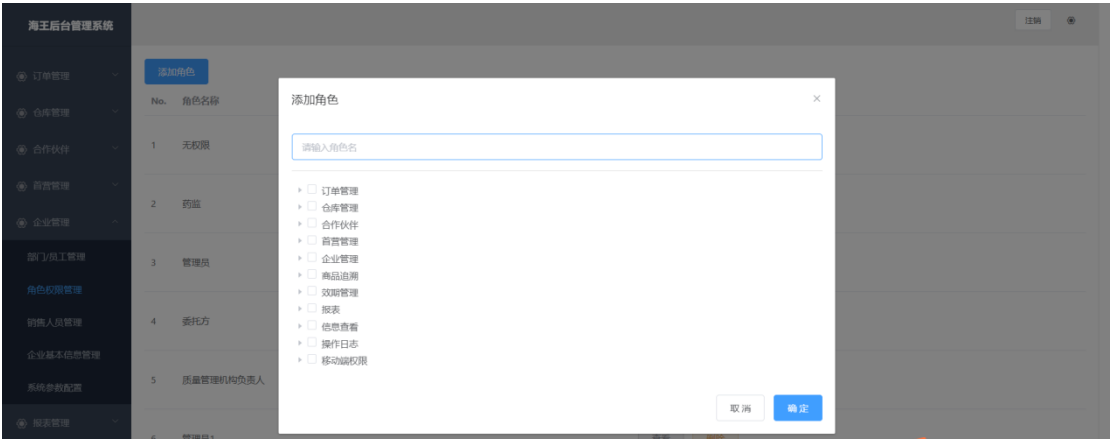


图 5-8 创建新角色权限功能界面图

5.2.3 系统参数配置

系统管理员通过系统参数配置为企业配置一些常见的系统参数，使其他模块的功能实现更加简便。

本功能模块主要包含的类如表 5-3 所示：

表 5-3 系统参数配置功能模块主要类

| | |
|-------|--|
| 控制层类 | ConfigDetailController.java |
| 业务逻辑类 | ConfigDetailService.java ConfigDetailServiceImpl.java |
| 持久对象类 | ConfigDetail.java |
| 数据持久类 | ConfigDetailRepository.java |

系统参数配置功能主要为系统配置 6 种参数：过期天数、产品品牌、产品类型、销售类型、就诊方式和产品线。由于功能实现都较简单，以过期天数的实现为例，界面图如图 5-9 所示：

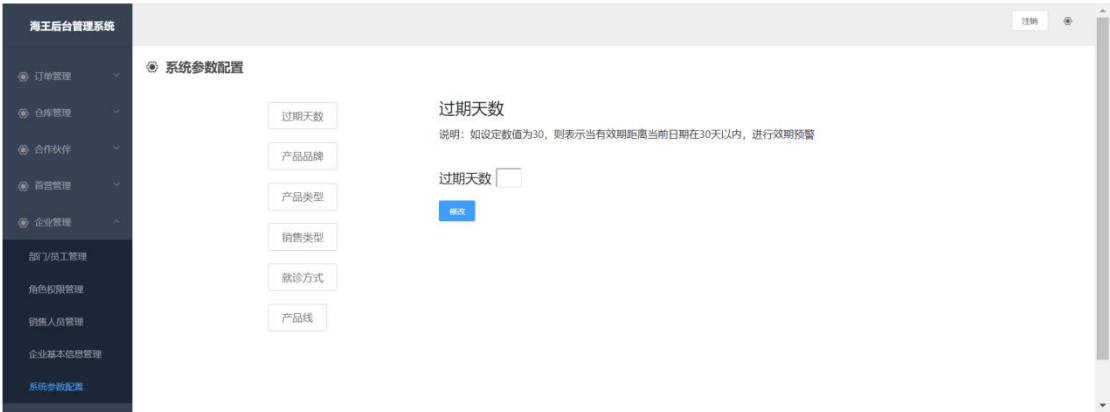


图 5-9 系统参数配置功能界面图

5.3 合作伙伴管理模块的实现

合作伙伴管理模块包括合作伙伴和商品信息两个子模块，合作伙伴子模块中分为合同与授权、合作科室与医生、合作伙伴首发地址和合作伙伴资料等子页面，商品信息中分为商品分类子页面。合作伙伴相关主要是将与企业合作的公司的信息记录在系统中；商品信息中商品分类也主要是记录企业所使用的商品和模板。下面主要介绍合作伙伴子模块中的合作科室与医生功能和商品信息子模块中的商品分类功能：

5.3.1 合作科室与医生

该部分主要是系统管理员添加与企业合作的相关医院里的科室及相应医生的信息。

本功能模块中包含的主要类如表 5-4 所示：

表 5-4 合作科室与医生功能模块主要类

| | |
|-------|--|
| 控制层类 | DoctorInfoControllor.java |
| 业务逻辑类 | DoctorInfoService.java DoctorInfoServiceImpl.java |
| 持久对象类 | DoctorInfo.java |
| 数据持久类 | DoctorInfoRepository.java |

本功能模块主要是系统管理员通过 DoctorInfoControllor 类添加与企业合作的相关医院，以及与企业合作的相关医院的科室及其医生的信息，实现的部分代码图如图 5-10 和 5-11 所示，界面图如下 5-12 所示：

```
finddoctor(item){
    this.isShow=1;
    this.tableData=[];
    this.currentdepart=item.name;
    //通过AJAX将医院和科室的信息传给后台，并接受后台返回的某医院某科室的所有医生信息
    this.$ajax.get('http://localhost:8080/doctorinfo/findByAuthAndDept',{
        params:{
            authnum:this.$router.currentRoute.params.tableid,
            departmentname:item.name
        }
    }).then(response=>{
        for(let i=0;i<response.data.length;i++){
            let mid={
                name:'',
                sex:'',
                account:'',
                tel:'',
                mail:'',
                position:''
            };
            mid.name=response.data[i].doctorname;
```

图 5-10 查看某医院某科室相关医生信息前端部分代码图

```
@GetMapping("/findByAuthbodyAndDept")
public List<DoctorInfo> findByAuthbodyAndDept(@RequestParam String authbody,@RequestParam String departmentname){
    //接受前端传过来的医院和科室信息，并调用以下方法找到数据库中相关信息
    List<DoctorInfo> test=doctorInfoService.findByAuthbodyAndDepartmentname(authbody, departmentname);
    return test;
}
```

图 5-11 查看某医院某科室相关医生信息后端部分代码图

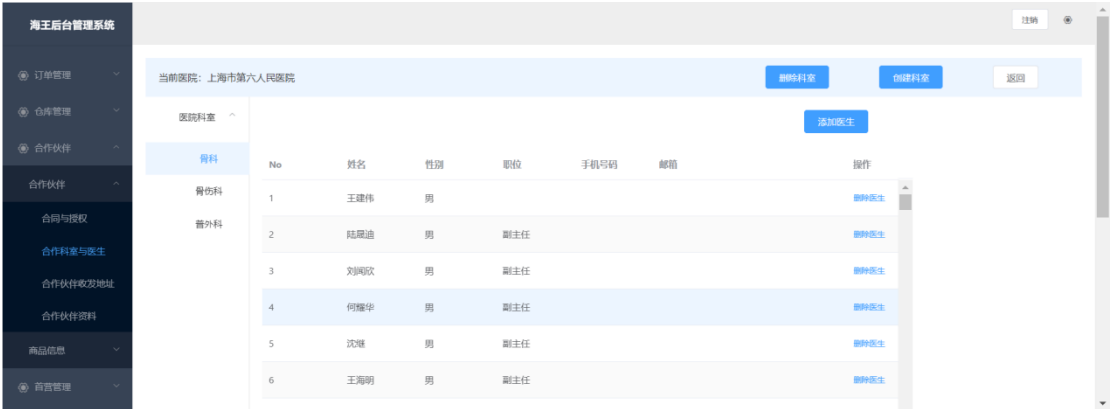


图 5-12 查看某医院某科室相关医生信息界面图

5.3.2 商品分类

商品分类主要是存储了所有商品的基本信息，为了便于企业根据商品品牌、商品类型或商品分类筛选并查找某个商品的信息。

本功能模块中包含的主要类如表 5-5 所示：

表 5-5 商品分类功能模块主要类

| | |
|-------|--|
| 控制层类 | CheckProductIController.java |
| 业务逻辑类 | CheckProductInfoService.java CheckProductInfoServiceImpl.java |
| 持久对象类 | CheckProductInfo.java |
| 数据持久类 | CheckProductRepository.java |

此处首先是将数据库中所有的商品信息通过 CheckProductIController 类中 findProduct 方法显示出来，并通过前端选择商品品牌来进行筛选显示商品信息及该品牌的商品类型，若用户希望二级筛选可以选择相应品牌的商品类型筛选显示相应商品信息，还可以通过商品分类进行三次筛选显示，或是通过输入商品编号进行搜索商品信息，下面将该功能部分代码图显示如下 5-13 所示，界面如下 5-14 所示：

```
@Override
public HashMap<String, Object> findProduct(String productbrand, String producttype, String productdesc, int currentpage, int pagesize){
    //接受前端传过来的商品品牌，商品类型，商品分类名称等信息
    HashMap<String, Object> map=new HashMap<>();
    //对传过来的3个数据进行判断，查询相应的数据库信息，分别进行显示
    if(productbrand.equals("")){
        Pageable pageable = new PageRequest( page: currentpage-1,pagesize);
        Page<CheckProductInfo> pages=checkProductRepository.findAllProduct(new PageRequest( page: currentpage-1,pagesize));
        map.put("tableData",pages);
        List<String> brandlist=checkProductRepository.findDistinctProductbrandByStatus(status);
        map.put("brandoptions",brandlist);
        List<String> prodcuttypelist=new ArrayList<>();
        map.put("typeoptions",prodcuttypelist);
        List<String> categorylist=new ArrayList<>();
        map.put("categoryoptions",categorylist);
    }else {
        if(producttype.equals("")){
            Pageable pageable = new PageRequest( page: currentpage-1,pagesize);
            Page<CheckProductInfo> pages=checkProductRepository.findAllByProductbrand(productbrand,pageable);
```

图 5-13 商品分类功能模块部分代码图



图 5-14 商品分类功能模块部分经三次筛选后界面图

5.4 订单管理模块的实现

订单管理模块主要包括采购订单、手术订单、经销商订单、借调单及订单审核 5 个子模块。本模块除手术订单既有移动端又有 PC 端，其他订单只有 PC 端，但在移动端和 PC 端两处的实现只是前端不同，后端及数据库均相同。

本小节订单相关的主要以采购订单中的供货商进货单功能模块、手术订单中的手术寄售单功能模块和手术寄售回收单功能模块相关的实现为例，并详细介绍了手术寄售单中给客户医生推荐模板功能的实现，订单审核模块因实现功能类似主要以手术寄售单审核为例。

5.4.1 供货商进货单

销售人员通过 PC 端创建供货商进货单，并填写相关信息。本功能模块主要包含的类如表 5-6 所示：

表 5-6 订单管理模块主要类

| 分类 | 主要包含的类 |
|-------|--|
| 控制层类 | SupplierOrderController.java |
| 业务逻辑类 | SupplierOrderService.java SupplierOrderServiceImpl.java |
| 持久对象类 | SupplierOrder.java |
| 数据持久类 | SupplierOrderRepository.java |

首先在前端的每个类型订单的汇总页面，通过访问 SupplierOrderController 类里的 findAll 接口，最后经过 Repository 与数据库交互后，将数据库中所有该类型订单返回给前端，使前端在页面显示该类型的所有订单基本信息；然后在此页面可以添加创建新的订单，用户点击添加后，在新的页面中填写新订单所有相关信息，通过访问相应订单类型的 SupplierOrderController 类里 saveOrder 接口，随后 SupplierOrderServiceImpl 实现类的方法 saveOne 将提交的新订单基本信息通过 Repository 在数据库中进行更新，若订单已创建，修改订单内容时调用 SupplierOrderController 类里 updateOrder 接口。

供货方进货单汇总页面的 PC 端界面图、创建供货方进货单页面的 PC 端界面图、保存新订单信息的 saveOne 方法分别如图 5-15，图 5-16，图 5-17 所示：

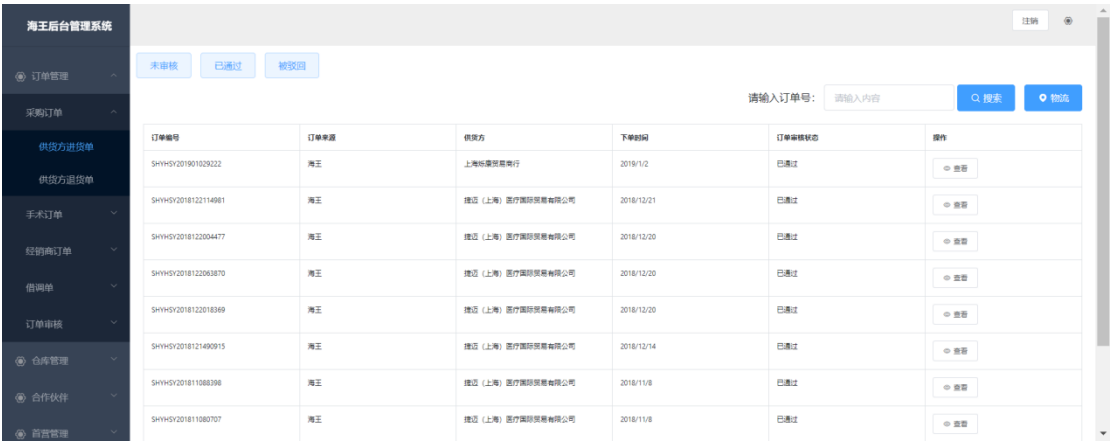


图 5-15 供货方进货单汇总页面



图 5-16 供货方进货单创建订单界面图

```
/**
 *保存供货商订单
 * @param supplierOrderDto
 * @return
 */
@Override
public String saveOne(SupplierOrderDto supplierOrderDto) { //supplierOrderDto是前端传过来的订单信息
    boolean flag = false;
    SupplierOrder supplierOrder = supplierOrderDto.getSupplierOrder();
    String orderId = IdUtil.getId( prefix: "SHYHSY");//自动获取orderId
    supplierOrder.setOrderId(orderId);
    supplierOrder.setCreateTime(new Date());
    //将信息进行转换，并保存至数据库中
    ArrayList<SupplierOrderDetail> supplierOrderDetails = (ArrayList<SupplierOrderDetail>) supplierOrderDto.getSupplierOrderDetails();
    for (SupplierOrderDetail cur: supplierOrderDetails) {
        cur.setOrderId(orderId);
        cur.setProductReg(checkProductRepository.findProductreg(cur.getProductNo()));
    }
    try{
        supplierOrderRepository.save(supplierOrder);
        supplierOrderDetailRepository.save(supplierOrderDetails);
        flag = true;
        logger.info("保存成功");
    }catch (Exception e){
    }
```

图 5-17 供货方进货单保存新订单 saveOne 方法部分代码图

5.4.2 手术寄售单

销售人员通过 PC 端或移动端创建手术寄售单，本功能模块的主要类如表 5-7 所示：

表 5-7 手术寄售单功能模块主要类

| 分类 | 主要包含的类 |
|-------|--|
| 控制层类 | SurgeryOrderController.java |
| 业务逻辑类 | SurgeryOrderService.java SurgeryOrderServiceImpl.java |
| 持久对象类 | SurgeryOrder.java |
| 数据持久类 | SurgeryOrderRepository.java |

本部分实现的方式和供货商进货单类似，在此不做赘述。手术寄售单汇总页面的 PC 端和移动端的界面图分别如图 5-18 和图 5-19 所示，创建手术订单页面的 PC 端和移动端的界面图分别如图 5-21 和图 5-20 所示，保存新订单信息的 saveOne 方法如图 5-22 所示：

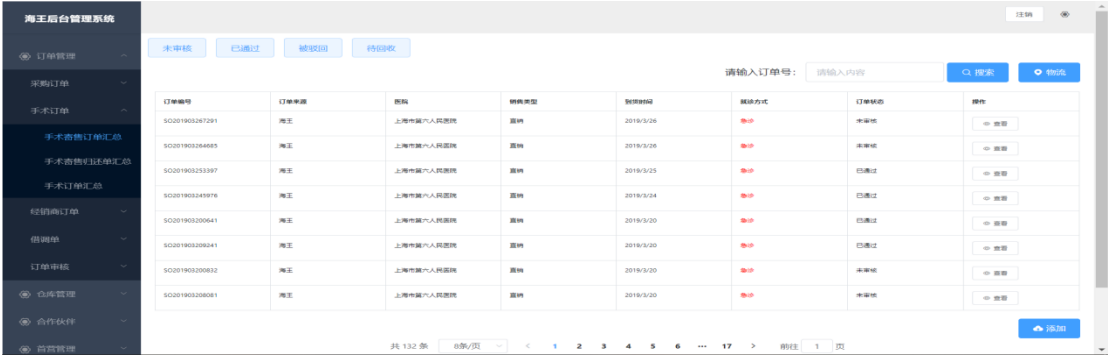


图 5-18 手术寄售单汇总页面-PC 端



图 5-19 手术寄售单汇总页面-移动端



图 5-20 手术寄售单创建订单界面-移动端

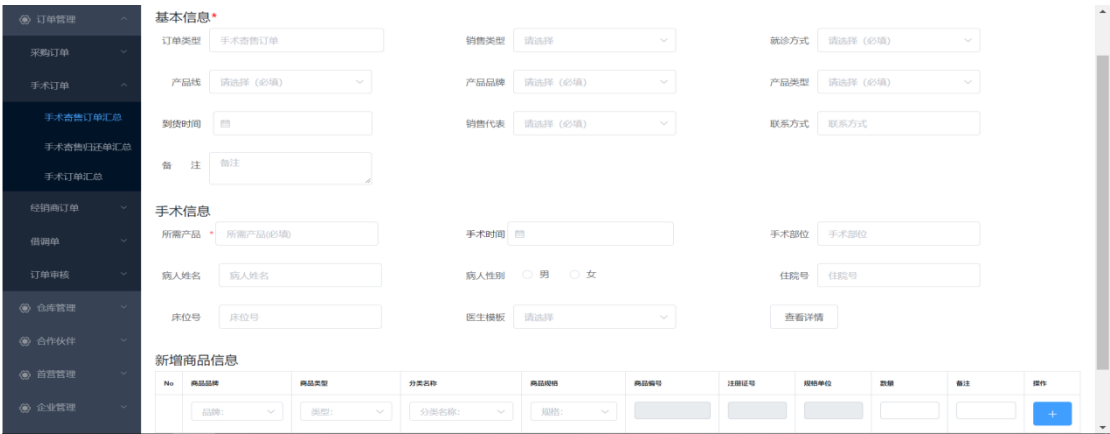


图 5-21 手术寄售单创建订单界面-PC 端

```
/**
 *保存手术订单
 * @param surgeryOrderDto
 * @return
 */
@Override
public String saveOne(SurgeryOrderDto surgeryOrderDto) {
    //将前端通过ajax传过来的信息进行相应处理并保存至数据库中
    boolean flag = false;
    String orderId = IdUtil.getId( prefix: "SO");
    SurgeryOrder surgeryOrder = surgeryOrderDto.getSurgeryOrder();
    surgeryOrder.setOrderId(orderId);
    surgeryOrder.setCreateTime(new Date());
    List<SurgeryOrderDetail> surgeryOrderDetails = surgeryOrderDto.getSurgeryOrderDetails();
    for (SurgeryOrderDetail cur:
        surgeryOrderDetails) {
        cur.setOrderId(orderId);
        cur.setDescription(cur.getCategoryName() + "," + cur.getProductModel());
    }
    String templateId = surgeryOrder.getSurgeryTemplateId();
    SurgeryOrderDetail surgeryOrderDetail;
    if (templateId != null){
        List<TemplateDetail> templateDetails = templateDetailRepository.findAllNoPreMatched(templateId);
        if (templateDetails != null){
            for (TemplateDetail cur:
                templateDetails){

```

图 5-22 手术寄售单保存新订单信息 saveOne 接口部分代码

5.4.3 手术寄售回收单

客户医生手术完，企业需要将使用的部分器械进行回收。本功能模块是销售人员通过 PC 端或移动端创建手术寄售回收单，主要类如表 5-8 所示：

表 5-8 手术寄售回收单功能模块主要类

| | |
|-------|--|
| 分类 | 主要包含的类 |
| 控制层类 | SurgeryOrderReturnController.java |
| 业务逻辑类 | SurgeryOrderReturnService.java SurgeryOrderReturnServiceImpl.java |

| | |
|-------|-----------------------------------|
| 持久对象类 | SurgeryOrderReturn.java |
| 数据持久类 | SurgeryOrderReturnRepository.java |

为了匹配企业实际业务流程，手术回收寄售单的创建仅在移动端实现，PC端只可以进行订单的查看、更新和删除。保存手术寄售回收单的后端接口代码与其他保存功能类似，在此便不多介绍，创建手术寄售回收单的移动端页面如图 5-23 所示：

< 返回

订单详情

备注:

手术信息

更新

手术名称:

是

手术时间:

手术部位:

213

病人名字:

123

病人性别:

☐ 男

☒ 女

住院号:

123

床位号:

213

医院名称:

医院

主刀医生:

王文杰

医生模板:

testLea

回收商品汇总

商品品牌:

邦美

商品类型:

螺钉

分类名称:

多向螺钉

商品规格:

10mm

商品编号:

816312010

注册证号:

22222

规格单位:

个

数量:

1

备注:

回收数量:

选择司机:

提交

图 5-23 创建手术寄售回收单截面图

5.4.4 手术寄售单审核

销售人员创建完新的订单后，订单审核员对该订单信息及库存进行审核。本功能模块主要类如表 5-9 所示：

表 5-9 手术寄售单审核功能模块主要类

| | |
|-------|--|
| 分类 | 主要包含的类 |
| 控制层类 | OrderAuditController.java |
| 业务逻辑类 | OrderAuditService.java OrderAuditServiceImpl.java |
| 持久对象类 | OrderAudit.java |
| 数据持久类 | OrderAuditRepository.java |

订单审核员通过前端将审核好的结果通过 AJAX 传递给后台，后台经 OrderAuditController 类的 saveSurgeryAudit 接口，经实现类和 Repository 最终与数据库交互，保存订单审核结果。订单审核页面部分代码及审核页面的移动端和 PC 端的界面测试分别如图 5-24、图 5-25 和图 5-26 所示：

```
@PostMapping("/saveSurgeryAudit")
public String saveSurgeryAudit(@RequestBody OrderAudit orderAudit){
    //接受前端传递过来的信息
    OrderAuditDto orderAuditDto = new OrderAuditDto();
    String orderId = orderAudit.getOrderId();
    boolean flag = false; //通过flag判断此订单是否第一次被审核
    flag = orderAuditService.isOrderIdExist(orderId);
    if (flag){ //订单如果不是第一次被审核，更新该订单审核信息
        OrderAudit orderAudit1 = new OrderAudit();
        orderAudit1 = orderAuditService.findOneByOrderId(orderId);
        orderAudit.setId(orderAudit1.getId());
    }
    int status = orderAudit.getOrderAuditStatus();
    try {
        surgeryOrderService.updateStatus(status, orderId);
        orderAuditDto.setSurgeryOrderDetails(surgeryOrderService.findAllExtra(orderId));
        orderAuditService.saveOrderVirSto(orderAuditDto);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return orderAuditService.saveOne(orderAudit);
}
```

图 5-24 订单审核页面部分代码-saveSurgeryAudit 接口



图 5-25 订单审核 PC 端界面图

< 返回

订单详情

商品品牌: 邦美

商品类型: 钉板

分类名称:

商品规格: 邦美

商品编号: 824071002

注册证号: 243341

规格单位: 个

数量: 2

备注:

商品品牌: 捷迈

商品类型: 钢板

分类名称:

商品规格: 捷迈

商品编号: 471673846

注册证号: 国食药监械(进)字2018第2161642号

规格单位: 袋

数量: 3

备注:

商品品牌: 邦美

商品类型: 棒

分类名称:

商品规格: 邦美

商品编号: 76324532524

注册证号: 312312

规格单位: 个

数量: 4

备注:

仓库信息

仓库:

请选择仓库名称

审核通过

审核驳回

图 5-26 订单审核移动端界面图

5.5 仓库管理模块的实现

仓库管理模块包括入库管理、出库管理和预配货管理 3 个子模块。入库主要涉及 5 种订单类型，出库主要涉及另外 5 种类型订单，预配货管理包括预配货模板、仓库查看和库存导入 3 个功能模块。

由于入库模块与出库模块类似，但出库模块多一个司机配货的实现，且其他功能模块功能实现较简单，故在此介绍出库模块中的手术寄售单功能模块。

5.5.1 出库管理-手术寄售单

出库功能模块涉及 3 个角色：仓库管理员、理配员及司机。首先仓库管理员收到出库提醒，为本次出库指派一个理配员，随后该理配员进行扫码出库操作并打印出库单，接着仓库管理员审核理配员配货操作，审核通过后指派司机进行发货，司机发货客户确认收货后订单结束。

本功能模块主要类如表 5-10 所示：

表 5-10 手术寄售单出库管理功能模块主要类

| | |
|------|-----------------------------|
| 控制层类 | SurgeryOrderController.java |
| | OutAuditController.java |
| | OutPickController.java |

| | |
|-------|--|
| 业务逻辑类 | SurgeryOrderService.java OutAuditService.java OutPickService.java SurgeryOrderServiceImpl.java OutAuditServiceImpl.java OutPickServiceImpl.java |
| 持久对象类 | SurgeryOrder.java OutAudit.java OutPick.java |
| 数据持久类 | SurgeryOrderRepository.java OutAuditRepository.java OutPickRepository.java |

下面主要从 5 个方面的实现进行介绍：

(1) 仓库管理员指派理配员

仓库管理员在订单汇总页面选择好待分配理配员的订单后，进入指派理配员页面，将选择好的理配员通过 AJAX 传给后台，后台调用 OutAuditService 类的 saveSurgeryAudit 经实现类和数据持久类与数据库交互，保存提交的信息。仓库管理员指派理配员界面如图 5-27 和图 5-28 所示，移动端出库审核汇总界面如图 5-29 所示：

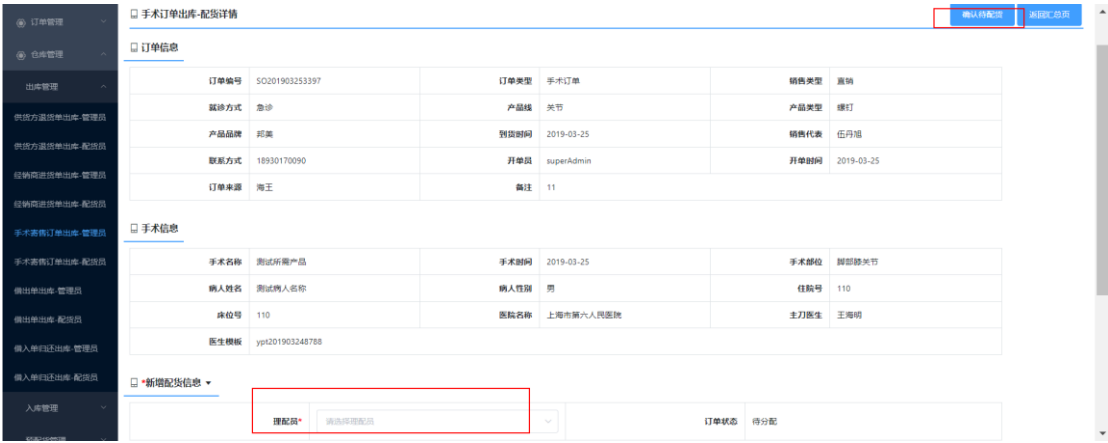


图 5-27 仓库管理员指派理配员界面图



图 5-28 出库审核汇总界面图



图 5-29 指派理配员移动端界面图

(2) 理配员配货

理配员收到通知后，进入该订单理配详情页面，点击配货扫码弹出模态框，如图 5-30 所示在模态框里点击开始扫码，点击完开始扫码后如图 5-31 显示使用扫码枪扫描货物的条形码（本处调试时直接输入，实际操作可以借助扫码枪操作），生产批号自动匹配副条形码第 11 位开始后的数值，点击确定配货完成，回到页面点击打印配货单，如图 5-32 所示弹出配货单模态框点击打印。本功能模块大部分逻辑实现均由前端操作，最后保存信息主要是通过后台 OutPickController 类的 saveOutPicks 方法。



图 5-30 配货扫码界面图

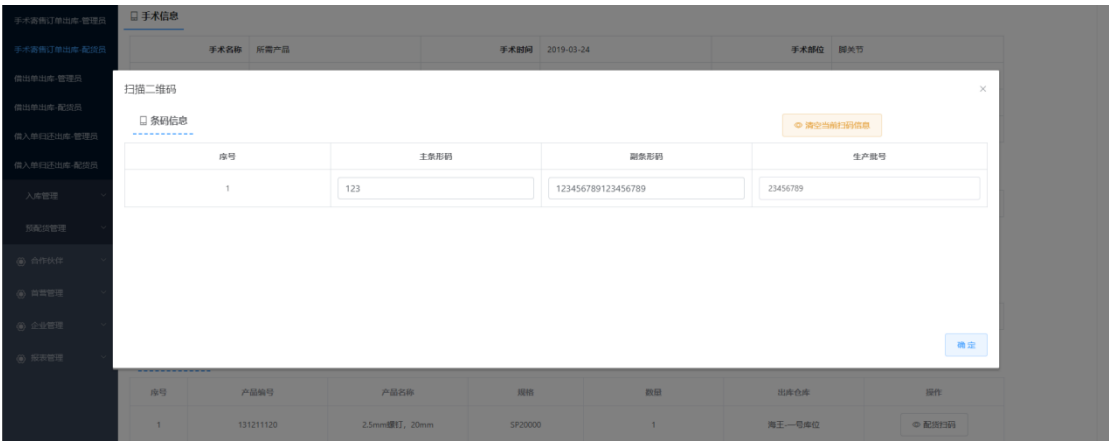


图 5-31 开始扫码界面图



图 5-32 打印配货单界面图

(3) 仓库管理员审核

理配员打印完配货单后交给仓库管理员，仓库管理员通过配货单及所配货物进行核对。本功能模块主要是经 OutAuditController 类的 saveSurgeryAudit 方法保存审核信息，PC 端和移动端审核界面分别如图 5-33 和图 5-34 所示。



图 5-33 仓库审核 PC 端界面图

< 返回

订单出库详情

床位号: 112

医院名称: 上海市第六人民医院

主刀医生: 何耀华

医生模板: ypt201903271656

配货信息

订单状态: 待审核

审核员:

出库仓库:

模板清单

新增商品

产品编号: 131211112

产品名称: 2.5mm螺钉, 12mm

规格: SP12000

数量: 1

出库仓库: 海王一号库位

产品编号: 131211120

产品名称: 2.5mm螺钉, 20mm

规格: SP20000

数量: 1

出库仓库: 海王一号库位

仓库审核通过

仓库审核驳回

图 5-34 仓库审核移动端界面图

< 返回

订单出库详情

病人注册: 男

住院号: 3213

床位号: 123

医院名称: 上海市第六人民医院

主刀医生: 何耀华

医生模板: ypt201903248722

配货信息

订单状态: 待发货

审核员:

审核意见:

送货司机: 请选择司机!

发货时间:

出库仓库:

模板清单

新增商品

产品编号: SP14000

产品名称: 2.5mm螺钉, 14mm

规格: SP14000

数量: 3

出库仓库: 海王-null

确认待发货

图 5-35 指派司机移动端界面图

(4) 仓库管理员指派司机

仓库管理员审核好后，指派该订单的司机进行发货，本功能模块主要调用后台 OutAuditController 类的 updateDriver 方法。实现后移动端和 PC 端的界面图分别为图 5-35 和 5-36 所示：

海王系统管理后台

订单管理

订单出库-配货详情

订单信息

订单编号: SC201901177216

订单类型: 手术订单

销售类型: 直销

配送方式: 急诊

产品品牌: 邦瀚

联系电话: 15802121034

发货时间: 2019-01-16

开单员: admin

开单时间: 2019-01-17

订单来源: 海王

备注: 222

手术信息

手术名称: F3

病人姓名: 张小五

手术时间: 2019-01-17

手术部位: 肘关节

病人性别: 男

住院号: 3213

医生模板: ypt201903248722

医院名称: 上海市第六人民医院

配货信息

配货员: 付永亮

审核员: admin

订单状态: 待发货

审核意见: 出库审核通过!

发货时间: 2019-04-09

送货司机: 请选择司机!

图 5-36 指派司机 PC 端界面图

(5) 司机发货

司机收到发货通知后，在手机进行确认发货操作。该功能模块主要是由 SurgeryOrderController 类的 updateStatus 方法实现，司机送货界面图如下 5-37 所示，点击确认接单后如图 5-38 所示：



图 5-37 司机送货界面图



图 5-38 司机点击确认接单后界面图

5.6 首营管理模块的实现

首营管理模块主要包括首营资质维护、首营品种维护、产品注册证维护及效期管理 4 个子模块，首营资质维护中包括资质登记、资质质检审核和资质法人审批 3 个功能模块，首营品种维护中包括品种登记、品种质检审核和品种法人审批 3 个功能模块，产品注册证维护包括产品注册证登记、产品注册证质检审核和产品注册证法人审批 3 个功能模块，效期管理包括库存效期预警和产品注册证效期预警 2 个功能模块。由于部分功能实现代码相似，本小节主要介绍首营资质维护子模块中的资质登记功能模块以及效期管理子模块中的库存效期预警功能模块。

5.6.1 首营资质登记

该部分主要是添加新的首营资质信息和查看已有的资质信息，在这里主要介绍添加新的首营资质信息相关的实现。

本功能模块的主要类如表 5-11 所示：

表 5-11 合作科室与医生功能模块主要类

| | |
|-------|--|
| 控制层类 | AuthorizedBodyControllor.java |
| 业务逻辑类 | AuthorizedBodyService.java AuthorizedBodyServiceImpl.java |
| 持久对象类 | AuthorizedBody.java |

| | |
|-------|-------------------------------|
| 数据持久类 | AuthorizedBodyRepository.java |
|-------|-------------------------------|

系统管理员输入相关信息后，通过 AuthorizedBodyControllor 类中的 createBasicInfo 方法保存，最终通过 AuthorizedBodyRepository 保存至相应数据库中，实现的部分代码如下图 5-39 所示，界面如图 5-40 所示：

```
@Transactional
@PostMapping("/createBasicInfo")
public void createBasicInfo(@RequestBody QualificationDto resultJson) {
    //从前端传入信息给后台，后台进行保存
    BasicInfo basicInfo = resultJson.getBasicInfo();
    basicInfoRepository.save(basicInfo);
    AuthorizedBody authorizedBody = authorizedBodyService.findOne(resultJson.getAuthId());
    authorizedBody.setBasicInfo(basicInfo);
    authorizedBody.setAuthbody(basicInfo.getEnterpriseName());
    authorizedBodyService.save(authorizedBody);
}
```

图 5-39 保存资质登记结果的部分代码图

图 5-40 保存资质登记结果的界面图

5.6.2 库存效期预警

该部分主要显示库存里已过期及临近过期的商品信息，本功能模块的主要类如表 5-12 所示：

表 5-12 库存效期预警功能模块主要类

| | |
|-------|--|
| 控制层类 | WarehouseStoNumberController.java |
| 业务逻辑类 | WarehouseStoNumberService.java WarehouseStoNumberServiceImpl.java |
| 持久对象类 | WarehouseStoNumber.java |
| 数据持久类 | WarehouseStoNumberRepository.java |

此处主要只是通过前端给后台传一个企业名称，随后调用后台 WarehouseStoNumberController 类的 findAllExpire 接口，通过 ServiceImpl 最终调用 WarehouseStoNumberRepository 与数据库进行交互，然后将得到的数据显示到前端页面。库存效期预警前后端实现代码图及界面图分别如图 5-41,图 5-42,图 5-43 所示：

```
view(enterprisenam) { //通过输入框传入值
    if (enterprisenam != "") {
        this.$ajax.get('http://localhost:8080/WarehouseStoNumber/findAllExpire',
        {
            params: {
                enterprisenam: enterprisenam //将该值传给后台
            }
        })
    }.then(response => { //后台返回结果，并将相应的结果存放至前端的数据中，最后显示出来
        //显示过期
        this.overbatchnum = response.data.expiry.length;
        this.expiry = response.data.expiry;
        for (let i = 0; i < response.data.expiry.length; i++) {
            this.producttime = new Date(response.data.expiry[i].producttime).toLocaleDateString();
            this.effecttime = new Date(response.data.expiry[i].effecttime).toLocaleDateString();
            this.overdays = (Date.parse(new Date()) - Date.parse(new Date(response.data.expiry[i].effecttime))) / (1000 * 60 * 60 * 24);
            this.overnum += response.data.expiry[i].stocknumber;
            this.expiry[i].overdays = Math.floor(this.overdays);
        }
        // 显示临近过期
        this.nearbatchnum = response.data.nearexpiry.length;
        this.nearexpiry = response.data.nearexpiry;
        for (let i = 0; i < response.data.nearexpiry.length; i++) {
            this.producttime = new Date(response.data.nearexpiry[i].producttime).toLocaleDateString();
            this.effecttime = new Date(response.data.nearexpiry[i].effecttime).toLocaleDateString();
            this.neardays = (Date.parse(new Date(response.data.nearexpiry[i].effecttime)) - Date.parse(new Date())) / (1000 * 60 * 60 * 24);
            this.nearexpiry[i].neardays = Math.floor(this.neardays);
        }
    })
}
```

图 5-41 库存效期预警的前端实现部分代码图

```
@Override
public HashMap<String, Object> findAllExpire(String enterprisenam) { //接受前端传过来的值
    if(enterprisenam.equals(Constant.HW_CONSTANT)) //判断前端传过来的公司是否是“海王”
    { //如果是，直接访问海王数据库中信息，并返回结果
        HashMap<String, Object> map=new HashMap<>();
        List<WarehouseStoNumber> list1= warehouseStoNumberRepository.findExpiry();
        map.put("expiry",list1);
        ConfigDetail configDetail=configDetailRepository.findById(Config.Expirydate.getIndex()).get(0);
        List<WarehouseStoNumber> list2=warehouseStoNumberRepository.findNearexpiry(Integer.parseInt(configDetail.getDetailName()));
        map.put("nearexpiry",list2);
        return map;
    } else { //否则，访问相对应公司的数据库中信息，并返回结果
        HashMap<String, Object> map=new HashMap<>();
        List<WarehouseStoNumber> list1= warehouseStoNumberRepository.findExpiryByEnterprisenam(enterprisenam);
        map.put("expiry",list1);
        ConfigInfo configInfo=configRepository.findById(Configname("临近过期日期"));
        List<WarehouseStoNumber> list2=warehouseStoNumberRepository.findNearexpiryByEnterprisenam(enterprisenam,configInfo.getConfignumber());
        map.put("nearexpiry",list2);
        return map;
    }
}
```

图 5-42 库存效期预警的后端实现部分代码图

海王

订单管理

仓库管理

合作伙伴

运营管理

运营管理维护

运营品种维护

产品注册证维护

效期管理

库存效期预警

合作伙伴证照效期预警

产品注册证效期预警

员工培训记录登记

员工健康记录登记

仓库存储记录登记

售后投诉及处理登记

库存效期预警

已过期! 共15批, 969件

| No. | 商品名称 | 商品编号 | 商品批号 | 过期数量(件) | 生产日期 | 有效期至 | 过期天数 | 注册证号(备案号) | 操作 |
|-----|-------------|-------------|---------------|---------|------------|------------|------|-----------------|----|
| 1 | 双向螺钉 | 816312010 | 6901939651209 | 97 | 2014-06-24 | 2018-06-07 | 67 | 国食药监械(备)第32233C | - |
| 2 | 3.5mm皮套螺钉 | 816357685 | 9787121270116 | 70 | 2015-07-24 | 2018-06-01 | 73 | | - |
| 3 | 双向螺钉 | 816367855 | 6901939651209 | 165 | 2018-06-24 | 2018-07-13 | 31 | | - |
| 4 | 5.5棒 | 76324532518 | 9787111421900 | 4 | 2018-06-30 | 2018-07-27 | 17 | | - |
| 5 | 第一跖趾关节融合术骨板 | 76324532514 | 1111 | 89 | | 2018-06-27 | 47 | | - |
| 6 | 双向螺钉 | 2341414 | 412431431 | 0 | 2018-06-28 | 2018-06-28 | 46 | 1 | - |
| 7 | 钉子 | 816333324 | 6939501807501 | 30 | 2018-06-18 | 2018-06-18 | 56 | HJ456 | - |
| 8 | 钉子 | 342543 | 9787111547426 | 30 | 2018-06-18 | 2018-06-18 | 56 | HJ456 | - |
| 9 | 钉子 | 33113488 | 969696 | 1 | 2018-06-18 | 2018-06-18 | 56 | HJ456 | - |
| 10 | 钉子 | 33113487 | 969696 | 1 | 2018-06-18 | 2018-06-18 | 56 | HJ456 | - |
| 11 | 钉子 | 33113488 | 969696 | 1 | 2018-06-18 | 2018-06-18 | 56 | HJ456 | - |

图 5-43 库存效期预警界面图

5.7 报表管理模块的实现

报表管理模块分为进货管理、销售管理、库存管理和统计分析等 4 个子模块。其中进货管理包括进货订单明细表、进货入库明细表、进货入库汇总表、进货订单未到货表、退货出库明细表和退货出库汇总表等功能模块，销售管理包括销售明细表、销售物料明细表、销售物料汇总表、年度销售汇总表和销售汇总表等功能模块，库存管理包括库存汇总表、库存明细表、出入库汇总表和出入库明细表等功能模块。由于功能类似，本小节以库存管理的库存汇总表和仓库报表分析为例。

5.7.1 库存汇总表功能模块

库存汇总表功能模块的主要类如表 5-13 所示：

表 5-13 库存汇总表功能模块主要类

| | |
|-------|--|
| 控制层类 | WarehouseStoNumberController.java |
| 业务逻辑类 | WarehouseStoNumberService.java WarehouseStoNumberServiceImpl.java |
| 持久对象类 | WarehouseStoNumber.java |
| 数据持久类 | WarehouseStoNumberRepository.java |

用户在此页面输入相关信息后通过前端调用 WarehouseStoNumberController 类中的 findAllByAllCondition 方法，最终通过 Repository 与数据库进行交互，将相应的数据传给前端，随后前端进行显示。本功能的部分代码如图 5-44 所示，界面如图 5-45 所示：

```
@Override
public List<WarehouseDetail> findAllByCondition(WarehouseDetailVo warehouseDetailVo) {
    Specification specification=new Specification() { //从前端传入所有筛选信息
        @Override
        public Predicate toPredicate(Root root, CriteriaQuery query, CriteriaBuilder cb) {
            List<Predicate>predicates=new ArrayList<>();
            if(warehouseDetailVo.getWarehousename()!=""){
                predicates.add(cb.like(root.get("warehousename"), S: "%"+warehouseDetailVo.getWarehousename()+"%"));
            }
            if(warehouseDetailVo.getProductid()!=""){
                predicates.add(cb.like(root.get("productid"), S: "%"+warehouseDetailVo.getProductid()+"%"));
            }
            if(warehouseDetailVo.getEnterprisename()!=""){
                predicates.add(cb.like(root.get("enterprisename"), S: "%"+warehouseDetailVo.getEnterprisename()+"%"));
            }
            if(warehouseDetailVo.getUpdatetime() != null){
                predicates.add(cb.equal(root.get("udpatedate"), warehouseDetailVo.getUpdatetime()));
            }
            return cb.and(predicates.toArray(new Predicate[predicates.size()]));
        }
    };
    //筛选结束后，调用findAll接口，显示所有筛选出来的数据
    List<WarehouseStoNumber> warehouseStoNumbers=warehouseStoNumberRepository.findAll(specification);
    List<WarehouseDetail> warehouseDetails=new ArrayList<>();
}
```

图 5-44 库存汇总表查询功能部分代码图

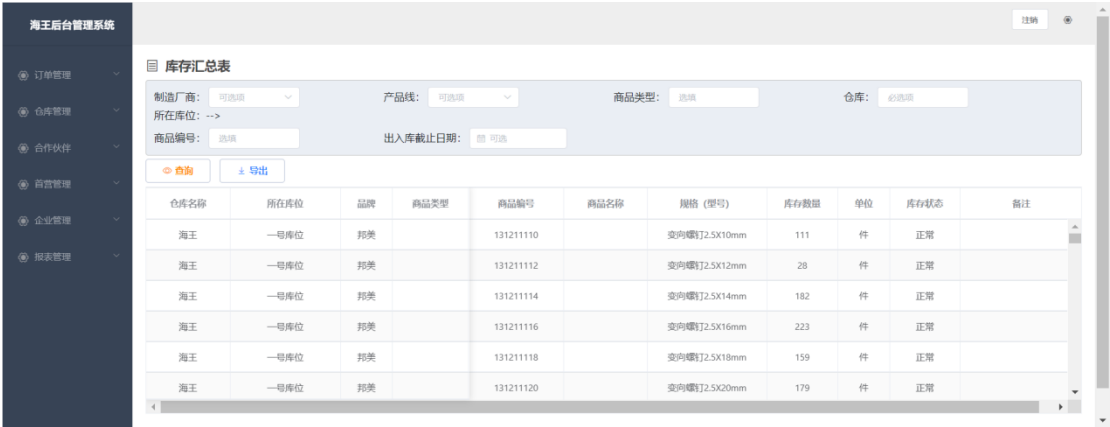


图 5-45 库存汇总表界面图

5.7.2 统计分析

本功能模块主要是系统管理员在统计分析页面进行相关设置，对已有的报表进行分析，本小节主要介绍月销量、年利润及评分最高但小于 6 的统计分析，主要涉及的类如表 5-14 所示：

表 5-14 仓库报表分析功能模块主要类

| | |
|-------|--|
| 控制层类 | PrematchBoxInfoController.java |
| 业务逻辑类 | PrematchBoxInfoService.java PrematchBoxInfoServiceImpl.java |
| 持久对象类 | PrematchBoxInfo.java |
| 数据持久类 | PrematchBoxInfoRepository.java |

(1) 所有手术部位月销量最高的模板

前端通过 AJAX 将所数据传给后台，后台通过 PrematchBoxInfoController 类的 getEverySurgeryPartOrderMostTemplateIdByMonth 接口，最后将数据传给前端，后端 Repository 部分代码如图 5-46 所示，该功能模块实现界面如图 5-47 所示：

```
@Query(value = "select surgery_template_id,templatename,productbrand,productline,producttype, surgery_part,templatecreatetime\n" +
"FROM(SELECT surgery_part,surgery_template_id\n" +
"FROM (\n" +
"SELECT a.create_time,a.surgery_part,a.surgery_template_id, date_format(a.create_time, '%Y%m') month,count(a.surgery_template_id) count\n" +
"FROM surgery_order_info AS a\n" +
"WHERE date_format(a.create_time, '%Y%m') = ?1 GROUP BY a.surgery_part,a.surgery_template_id ORDER BY count desc\n" +
"tmp GROUP BY surgery_part ORDER BY count desc )\n" +
"tmp2,prematch_template_info WHERE tmp2.surgery_template_id = prematch_template_info.templateid", nativeQuery = true)
List<Object[]> getEverySurgeryPartOrderMostTemplateIdByMonth(String month);
```

图 5-46 月销量最高 Repository 部分 SQL 代码

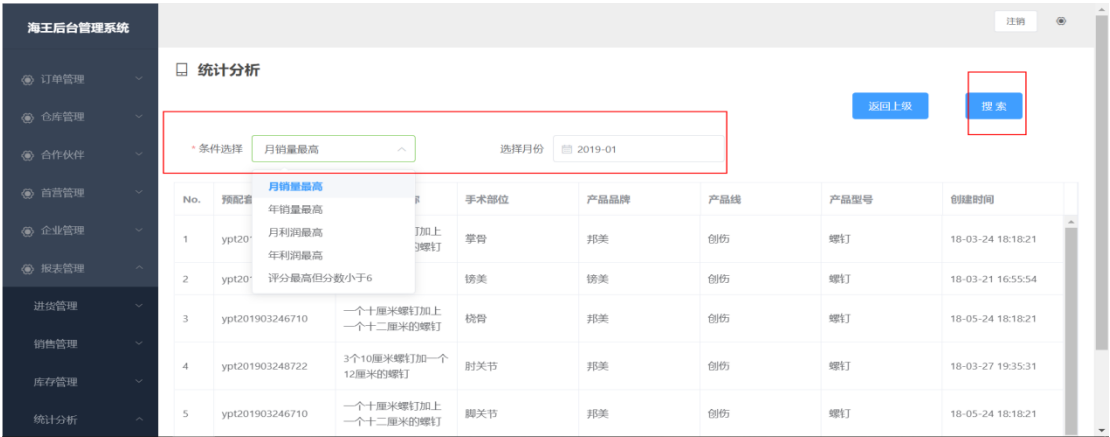


图 5-47 月销量最高界面图

(2) 所有手术部位年利润最高的模板

该功能模块的实现与（1）类似，但后端是调用与（1）相同的类的 `getEverySurgeryPartProfitMostTemplateIdByYear` 方法。该方法的 Repository 如图 5-48 所示，该功能模块的实现如图 5-49 所示：

```
@Query(value = "select surgery_template_id,templatename,productbrand,productline,producttype, surgery_part,templatecreatetime\n" +
    "FROM(\n" +
    "select * FROM(\n" +
    "SELECT tmp.create_time, tmp.surgery_part, surgery_template_id,templateid, tmp.month,pre.totalprice,pre.costprice,count, " +
    "(totalprice - pre.costprice) benefit,count*(totalprice - pre.costprice) totalbenefit FROM (\n" +
    "SELECT a.create_time,a.surgery_part,a.surgery_template_id, date_format(a.create_time, '%Y') month,count(a.surgery_template_id) count \n" +
    "FROM surgery_order_info AS a\n" +
    "WHERE date_format(a.create_time, '%Y') = ?1 GROUP BY a.surgery_part,a.surgery_template_id\n" +
    ") tmp,prematch_template_info pre\n" +
    "WHERE tmp.surgery_template_id = pre.templateid ORDER BY totalbenefit desc LIMIT 1000\n" +
    ") sectmp GROUP BY surgery_part\n" +
    ") tmp2,prematch_template_info WHERE tmp2.surgery_template_id = prematch_template_info.templateid", nativeQuery = true)
List<Object[]> getEverySurgeryPartProfitMostTemplateIdByYear(String year);
```

图 5-48 年利润最高 Repository 部分 SQL 代码



图 5-49 年利润最高界面图

(3) 所有手术部位评分最高但分数小于 6 的模板

该功能模块的实现与（1）类似，但后端是调用与（1）相同的类的 `getEverySurgeryPartMostScoreLess6TemplateId` 方法。该方法的 Repository 如图 5-50 所示，该功能模块的实现如图 5-51 所示：


```
@Query(value = "select surgery_template_id,templatename,productbrand,productline,producttype, surgery_part,templatecreatetime\n" +
"FROM(\n" +
"SELECT surgery_order_info.order_id,surgery_order_info.surgery_part,surgery_order_info.surgery_template_id, surgery_order_return_info.score\n" +
"FROM surgery_order_info , surgery_order_return_info \n" +
"WHERE surgery_order_info.order_id = surgery_order_return_info.order_id and (surgery_part,score) in \n" +
"(\n" +
"\tSELECT surgery_part,max(score)\n" +
"\tFROM surgery_order_info , surgery_order_return_info\n" +
"\tWHERE surgery_order_info.order_id = surgery_order_return_info.order_id GROUP BY surgery_part\n" +
") and score < 6) tmp2,prematch_template_info WHERE tmp2.surgery_template_id = prematch_template_info.templateid", nativeQuery = true)
List<Object[]> getEverySurgeryPartMostScoreLess6TemplateId();
```

图 5-50 评分最高但小于 6 的 Repository 部分 SQL 代码



图 5-51 评分最高但小于 6 的界面图

5.8 预配货模板的实现

本文通过 python 实现 Apriori 算法计算预配货模板的，主要代码图如图 5-52 所示：

```
def apriori(dataSet, minSupport = 0.8):
    C1 = createC1(dataSet)
    D = list(map(set, dataSet))
    L1, supportData = scanD(D, C1, minSupport)#单项最小支持度判断 0.8，生成L1
    L = [L1]
    k = 2
    while (len(L[k-2]) > 0):#创建包含更大项集的更大列表,直到下一个大的项集为空
        Ck = aprioriGen(L[k-2], k)#生成Ck
        Lk, supK = scanD(D, Ck, minSupport)#得到Lk
        supportData.update(supK)
        L.append(Lk)
        k += 1
    return L, supportData
```

图 5-52 Python 实现 Apriori 算法主要代码图

得到预配货模板信息后，系统管理员将相应的信息录入系统，本部分功能主要是对预配货模板的增删改查，主要涉及的类如表 5-15 所示：

表 5-15 预配货模板功能模块主要类

| | |
|-------|--|
| 控制层类 | PrematchBoxInfoController.java PrematchTemplateInfoController.java |
| 业务逻辑类 | PrematchBoxInfoService.java PrematchTemplateInfoService.java PrematchBoxInfoServiceImpl.java PrematchTemplateInfoServiceImpl.java |
| 持久对象类 | PrematchBoxInfo.java PrematchTemplateInfo.java |
| 数据持久类 | PrematchBoxInfoRepository.java PrematchTemplateInfoRepository.java |

首先预配货模板汇总页面显示所有的预配货模板，该功能主要是通过 PrematchTemplateInfoController 类的 findAllByPage 方法实现的，使用 PC 端创建新模板界面图如图 5-53 所示，该页面保存模板的功能主要通过 PrematchTemplateInfoController 类的 saveTemplate 方法实现，查看模板信息是通过 PrematchTemplateInfoController 类的 finddetail 方法实现。

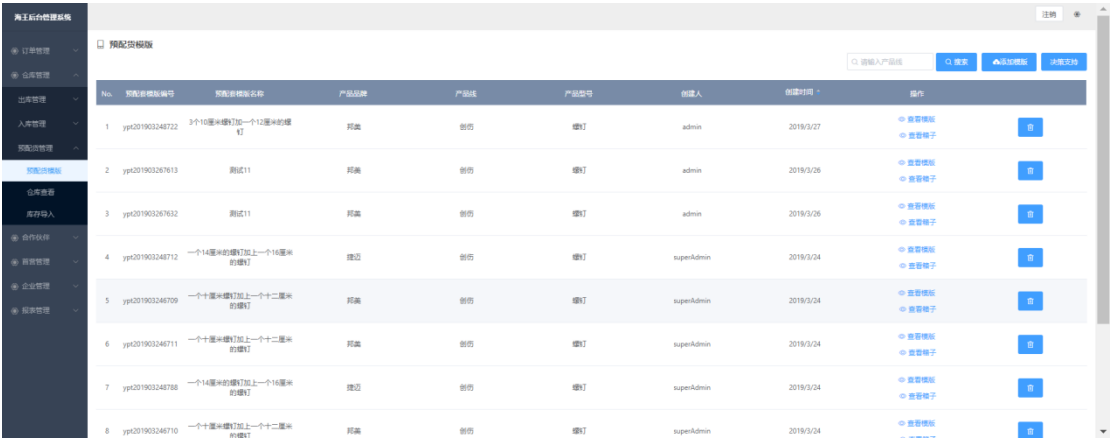


图 5-53 预配货模板汇总页界面图

5.9 推荐预配货模板的实现

预配货模板主要是在手术寄售单创建的状态下，销售人员选择了手术部位后，系统自动匹配出该部位经手术寄售回收单评分最高的医生模板和评分大于 6 的前提下价格最低的医生模板，前者表示医生认为该模板效果最佳，后者表示该模板性价比最高。将这两种模板显示出来，推荐给医生，供医生进行选择，第一次合作的医生可以选择这两种之一，长期合作的医生可以考虑是否更换自己原始的配套模板。

本功能模块的主要类如表 5-16 所示：

表 5-16 推荐预配货模板功能模块主要类

| 分类 | 主要包含的类 |
|-------|--|
| 控制层类 | SurgeryOrderController.java |
| 业务逻辑类 | SurgeryOrderService.java SurgeryOrderServiceImpl.java |
| 持久对象类 | SurgeryOrder.java |
| 数据持久类 | SurgeryOrderRepository.java |

销售人员在下单填写好手术部位后，通过 AJAX 将手术部位传给后台的 SurgeryOrderController 类的 findMaxScoreSurgeryTemplateId 方法和 findCostLeastPerformanceBestSurgeryTemplateIdBySurgeryPart 方法，随后两种方法通过 ServiceImpl 执行类最后经 Repository 与数据库交互，通过数据分析得到效果最佳的模板及性价比最高的模板传给前端，最后显示在前端页面上，该功能的部分代码如图 5-54，图 5-55，图 5-56 所示，例如前端将手术部位“肩部”传给后台后，页面显示如图 5-57 所示：

```
/**
 *根据surgeryPart查询所有已评分的订单中分数最高模板Id
 * @param surgeryPart
 * @return String
 */
@Override
public String findMaxScoreSurgeryTemplateId(String surgeryPart) {
    String surgeryTemplateId = "";
    try{
        surgeryTemplateId = surgeryOrderRepository.findMaxScoreSurgeryTemplateIdBySurgeryPart(surgeryPart);
        logger.info("查询成功...");
    }catch (Exception e){
        logger.info("查询失败...");
    }
    return surgeryTemplateId;
}
```

图 5-54 推荐预配货模板的部分代码图

```
@Query(value = "SELECT " +
    "surgery_order_info.surgery_template_id " +
    "FROM " +
    "surgery_order_info , " +
    "surgery_order_return_info " +
    "WHERE " +
    "surgery_order_info.order_id = surgery_order_return_info.order_id and surgery_part = ?1 " +
    "ORDER BY surgery_order_return_info.score desc LIMIT 1",nativeQuery = true)
String findMaxScoreSurgeryTemplateIdBySurgeryPart(String surgeryPart);
```

图 5-55 推荐效果最佳的预配货模板的 Repository 部分代码图

```
@Query(value = "SELECT tmp.surgery_template_id +  
FROM ( +  
SELECT +  
surgery_order_info.surgery_part, +  
surgery_order_info.surgery_template_id, +  
surgery_order_return_info.score +  
FROM +  
surgery_order_info , +  
surgery_order_return_info +  
WHERE +  
surgery_order_info.order_id = surgery_order_return_info.order_id and surgery_order_info.surgery_part=?1 and score>6 ) +  
tmp,prematch_template_info +  
WHERE tmp.surgery_template_id = prematch_template_info.templateid ORDER BY totalPrice LIMIT 1",nativeQuery = true)  
String findCostLeastPerformanceBestSurgeryTemplateIdBySurgeryPart(String surgeryPart);
```

图 5-56 推荐性价比最高的预配货模板的 Repository 部分代码图

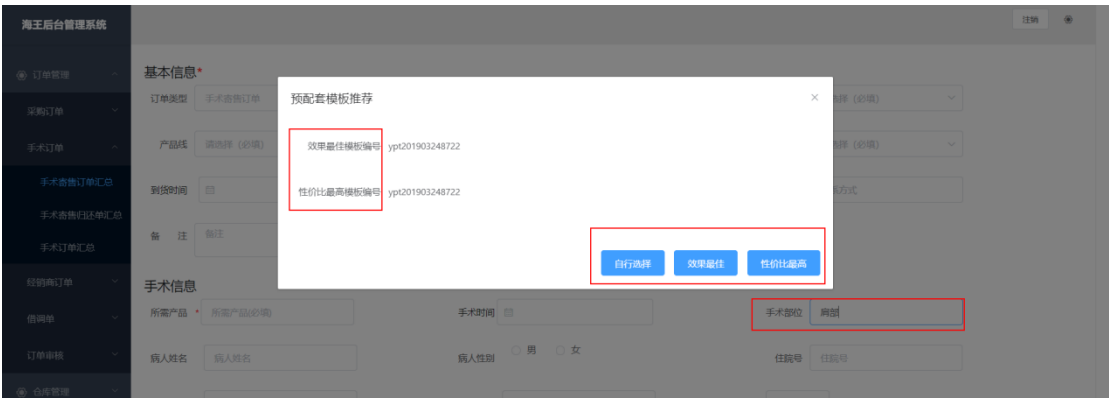


图 5-57 输入“肩部”后推荐预配货模板的界面图

5.10 系统测试

系统测试在系统设计的过程中必不可少，它能检验系统是否能够保质保量交付用户。软件测试可以降低软件发生故障的几率，完善软件的各个功能，争取让软件功能的实现能满足客户的实际需求。

5.10.1 系统测试环境

Web 端系统测试环境：

- (1) 操作系统：Windows
- (2) 服务器：Tomcat
- (3) 数据库：Mysql
- (4) 开发环境：JetBrains WebStorm 2018、IntelliJ IDEA 2018
- (5) 客户端浏览器：Google Chrome、Internet Explorer 8.0、Mozilla Firefox、Internet Explorer 10.0

移动端测试环境：

- (1) 操作系统：IOS / Android
- (2) 开发环境：JetBrains WebStorm 2018、IntelliJ IDEA 2018

5.10.2 系统测试内容

针对本系统的设计与实现，将对系统进行如下的测试：

(1) 界面测试

本部分的测试主要是为了提高用户体验，检查界面的展示是否正常，页面的样式和整体布局是否符合用户使用习惯，前端页面的色彩搭配是否合理，色调是否一致，整体是否美观等。

(2) 功能测试

本部分主要是为了测试系统的功能是否正确且满足需求。

(3) 兼容性测试

本部分主要是测试系统在三种或以上内核浏览器上运行时页面是否正常，功能是否正确等。

5.10.3 系统测试用例和结果

依据上节介绍的系统测试环境和测试内容，对本系统展开测试结果如下所示：

(1) 界面测试及兼容性测试，如表 5-17 所示。

表 5-17 系统界面测试及兼容性测试用例和测试结果

| | |
|------|---|
| 测试项目 | 界面测试、兼容性测试 |
| 测试目的 | 1.验证界面是否满足用户体验 2.验证系统是否能在三种不同内核的浏览器上运行 |
| 测试条件 | IE 浏览器（Trident 内核）、Firefox 浏览器（Gecko 内核）和 Chrome 浏览器（Webkit 内核） |
| 测试步骤 | 1.分别用各浏览器登陆第三方医疗器械供应链系统 2.进入手术寄售单汇总页面 3.进入添加手术寄售单页面 |
| 预期结果 | 1.登陆页面及登陆成功页面显示正常 2.手术寄售单汇总页面显示正常 3.添加手术寄售单页面显示正常 |
| 测试结果 | 与预期结果一致 |

(2) 订单管理模块功能测试，如表 5-18 所示。

表 5-18 系统订单管理模块功能测试用例和测试结果

| | |
|------|--|
| 测试项目 | 功能测试 |
| 测试目的 | 是否满足订单管理模块的功能需求 |
| 测试步骤 | 1.创建手术寄售单 2.在移动端进行手术回收 |
| 预期结果 | 1.手术寄售单必要信息若填写不完整，系统进行提示补充完整，否则不予保存 2.使用移动端可以正常进行手术回收操作 |

| | |
|------|---------|
| 测试结果 | 与预期结果一致 |
|------|---------|

(3) 仓库管理模块功能测试，如表 5-19 所示。

表 5-19 系统仓库管理模块功能测试用例和测试结果

| | |
|------|---|
| 测试项目 | 功能测试 |
| 测试目的 | 是否满足仓库管理模块的功能需求 |
| 测试步骤 | 1.使用仓库管理员角色登陆进行指派相应配货员 2.使用上述配货员角色登陆进行配货操作 |
| 预期结果 | 1.指派成功 2.收到指派消息，并配货成功 |
| 测试结果 | 与预期结果一致 |

5.10.4 系统测试结果分析

通过上述对系统的界面、兼容性、功能及速度等方面编写测试用例并按照此用例进行测试得出的结果，验证了系统的基本供应链流程，页面展示得到较好的用户体验，各功能实现情况良好并达到兼容性要求，基本实现了系统的设计，满足了用户需求。

5.11 系统应用情况与效果

系统通过 SpringBoot 和 Vue 框架实现后已于 2018 年 9 月在 A 企业服务器运行系统的第一个版本，经该企业人员运用后，可以满足 A 企业业务流程的需求，也通过了国家食药监监管部门的检查，解决了 A 企业从前业务流程困难的问题，同时也提高了该企业销售量以及企业内部相关人员管理的效率。

5.12 本章小结

本章在第三章需求分析和第四章详细设计的基础上，对系统各个模块具体实现进行了描述，对这些模块中部分代码做了简单介绍和实现描述，并对关键功能做了界面展示。然后，对本系统进行了相关软件测试。最后，对本平台的应用情况与效果做了介绍。

第六章 结论与展望

6.1 结论

本文主要叙述了第三方医疗器械供应链平台的设计与实现,介绍了平台的研究背景、研究现状、项目需求分析、详细设计、数据库设计和模块详细实现等内容。

本文所设计的第三方医疗器械供应链平台,是一个符合食药监要求的自动化系统。包括企业管理、合作伙伴管理、订单管理、仓库管理、首营管理和报表管理等模块。其中企业管理主要是负责企业基本管理,如企业部门和员工、企业基本信息和企业参数配置等;合作伙伴管理主要负责管理企业的合作伙伴相关信息,如合作伙伴基本信息和合同信息等;订单管理主要负责管理企业的各种类型订单;仓库管理主要负责管理企业仓库的进出业务;首营管理主要负责第一次与企业发生供需关系的企业的资质等信息;报表管理主要负责管理企业的各种报表。本文还通过 Apriori 算法计算订单中商品之间的关联性,并设计了预配货模板功能模块;以及通过手术回收时医生对预配货模板的评分,在创建订单时对客户医生进行推荐效果最佳及性价比最高的两款模板。

在本平台中,大部分模块之间相互协作,实现模块之间的数据传递,系统运用了目前流行且先进的框架 SpringBoot 和 Vue,保证了平台按照预期效果正常运行。

6.2 展望

本文的第三方医疗器械供应链平台是由 Vue 和 SpringBoot 框架开发实现的易于维护和扩展的第三方医疗器械双向物流供应链系统,为了更好地匹配企业需求,以及更好地提高企业工作人员办事效率,系统还需完善以下部分:

1. 增加灵活的辅助企业决策支持功能;
2. 利用推荐算法匹配邻居医生,推荐邻居医生模板,供客户医生进行多重选择;
3. 报表的分析可以考虑添加一些数据可视化功能。

参考文献

- [1] 崔忠付.我国医疗器械物流发展问题与趋势分析[J].物流技术与应用, 2015, 20 (11):50-51.
- [2] 袁渊. 应大力发展医疗器械第三方物流[N]. 中国医药报,2014-12-18(003).
- [3] 朱二华.基于 Vue.js 的 Web 前端应用研究[J].科技与创新.2017(20):119-121.
- [4] Shufan Liu. Design and Implementation of Communication Base Station Survey System based on Springboot[A]. 香港新世纪文化出版社 (Hongkong New Century Cultural Publishing House). 2018 年智慧教育与人工智能发展学术会议论文集 (第一部分) [C].香港新世纪文化出版社 (Hongkong New Century Cultural Publishing House):香港新世纪文化出版社有限公司,2018:3.
- [5] 叶楠. 医疗器械电商转型的全渠道战略[D].北京外国语大学,2017.
- [6] 孙德刚.国内外数字化医疗设备发展现状与趋势[J].机器人技术与应用,2003(01):10-13.
- [7] StateOfJS. The State Of JavaScript 2018[EB/OL]. <https://2018.stateofjs.com/introduction/>, 2018-11-30/2019-04-01.
- [8] 麦冬,陈涛,梁宗湾.轻量级响应式框架 Vue.js 应用分析[J].信息与电脑(理论版),2017(07):58-59.
- [9] 王振宇. 基于 SpringBoot 的整车出库管理系统设计与开发[D].南京邮电大学,2018.
- [10]易剑波.基于 MVVM 模式的 WEB 前端框架的研究[J].信息与电脑(理论版),2016(19):76-77+84.
- [11]田媛. 基于 ASP.NET 的鲜花销售系统[D].天津大学,2016.
- [12]张正龙,陈永政. 浅谈 MVP 设计模式[J].科学咨询(科技·管理),2014(09):71.
- [13]陈涛. MVVM 设计模式及其应用研究[J].计算机与数字工程,2014,42(10):1982-1985.
- [14]单智峰. 论《招生录取管理系统》开发设计与实现[J].通讯世界,2017(04):203-206.
- [15]李洁. 主流 JavaScript 框架——AngularJS、React 和 Vue 使用体会[J].电脑迷,2019(01):77.
- [16]郑彦孚. MySQL 的 JDBC 编程实例[J].电脑开发与应用,2007(01):58+63.
- [17]辛晓越. 文档型数据库的存储模型设计和研究[D].中山大学,2015.
- [18]张海勇, 冯海波, 李洪奇. 基于 MongoDB 的特殊测井分布式数据库系统[J].中国管理信息化,2014,17(23):45-46.

- [19]翟剑锟. Spring 框架技术分析及应用研究[D].中国科学院大学(工程管理与信息技术学院),2013.
- [20]叶云鹏,毕津源.基于 Spring Boot 的家政服务平台设计[J].科技广场,2017(03):182-185.
- [21]刘军煜,贾修一.一种利用关联规则挖掘的多标记分类算法[J].软件学报,2017,28(11):2865-2878.
- [22]Pang-Ning Tan, Michael Steinbach, Vipin Kumar. 数据挖掘导论[M]. 范明, 范宏建, 译. 北京: 人民邮电出版社,2010: 328-330. Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to data mining[M]. Translated by Fan Ming, Fan Hongjian. Beijing: Posts and Telecom Press, 2010: 328-330(in Chinese).
- [23]王春玲,李川,李想.基于 Apriori 算法的高校 Web 日志挖掘系统构建[J].中国林业教育,2019,37(02):22-26.
- [24]吴小东,曾玉珠.基于 Apriori 算法的高校学生成绩数据挖掘[J].廊坊师范学院学报(自然科学版),2019,19(01):31-36.
- [25]徐倩雯. 基于 J2EE 的供应链管理系统的设计与实现[D].吉林大学,2016.
- [26]焦鹏琿. 基于 SpringBoot 和 Vue 框架的电子招投标系统的设计与实现[D].南京大学,2018.
- [27]黎炳权. 基于 Web 和移动应用的高校学生信息管理平台[D].东华大学,2017.
- [28]赵瑛. 关于泊松分布及其应用[J]. 辽宁省交通高等专科学校学报,2009,11(02):77-78.
- [29]丁东辉. 基于 Hadoop 和 D3.JS 的互联网+博物馆可视化平台的研究与实现[D].东华大学,2017.

攻读学位期间的研究成果

[1] 贺紫珺. 基于关联规则的医疗器械预配货模板的设计与实现. 智能计算机与应用. 已录用

致 谢

发自内心的说，本论文的完成离不开我的导师石秀金老师的耐心指导。在论文撰写期间，石老师给了我很多实用性的建议，经常让我的思路一下子变得豁然开朗，石老师并不像其他的老师那么严格，于我来说可能更像一个“慈母”，让我感觉到很温暖，我不是个聪明的学生，老师却虚心的指导我，从没有说过一句狠话，研究生生涯能遇到石老师真的觉得很幸福，老师平时还有很多教学上的事情，但他却还是一直很认真指导我的论文，在此向石老师表达我最真诚最真挚的感谢。还有感谢百忙之中抽空对我的论文进行评审的院内老师们，你们的每一个意见都非常宝贵。

还要感谢我实验室的张天同学、牛鑫同学、黄科科同学和张梦娜同学，他们的乐观积极的态度，让我每天压力再大，都有一种阳光照耀着的感觉，以及他们在科研路上的上进心，督促着我前进的脚步。

最后，感谢今天在百忙之中抽空参加答辩的老师们，你们辛苦了！