**西安邮电大学**

**毕业设计（论文）**

题目： 基于Springboot框架技术的

住院管理系统设计与实现

学院： 计算机学院

专业： 软件工程

班级： 软件1702

学生姓名： 靳飞虎

学号： 04173052

导师姓名： 拓守恒 职称： 副教授

起止时间：2020年12月7日 至 2021年6月8日

毕业设计（论文）承诺书

本人所提交的毕业论文《基于Springboot框架技术的住院管理系统设计与实现》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；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注明并表示感谢。

本人深知本承诺书的法律责任，违规后果由本人承担。

签名：

日期：2021年6月1日

西安邮电大学本科毕业设计(论文)选题审批表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 申报人 | 靳飞虎 | | 职称 | | 学生 | | 学院 | | | | 计算机学院 | |
| 题目  名称 | 基于Springboot框架技术的住院管理系统设计与实现 | | | | | | | | | | | |
| 题目  来源 | 科研 | | |  | 教学 | | |  | | 其它 | | √ |
| 题目  类型 | 软件系统研发 | | | √ | 软件产品设计 | | |  | | 软件技术研究 | |  |
| 题目  简述 | 1. 课题背景   随着科学技术的不断提高，计算机科学日渐成熟，其强大的功能已为人们深刻认识，在各行各业中都离不开信息处理，这正是计算机被广泛应用于信息管理系统的环境。计算机最大的好处在于利用它能够进行信息管理，使用计算机进行信息控制，不仅提高了工作效率，而且大大提高了安全性。医院作为一种服务型的机构，需要统计和保存的病人和医生资料繁多，包含很多的信息数据的管理，基于这种目的，有必要建立一个医院管理系统，使医院管理的工作规范化，系统化，程序化，避免医院管理的随意性，提高信息处理的速度和准确性，及时准确的查询和录入病人的入院、出院情况。   1. 意义   对于医院而言，医院住院管理系统是每一个医疗机构管理病人及医师收入和支出来说不可缺少的一个管理系统，它的内容对于医疗机构的管理者来说是至关重要要的，所以它应该能够为每一个医疗机构的管理者提供充足的信息和快捷的查询手段，这在很大程度上方便医疗机构的管理者的合理管理。   1. 研究内容   由于早期的医院信息系统都是由JSP/Servlet实现，因此在维护和性能上已经跟不上时代的发展，因此本项目采用Spring Boot、Vue的经典前后端分离的框架对医院信息系统进行了重构。其主要功能为能够实现辅助医院医护人员的工作，提高医院医护人员的工作效率。本系统需要实现；   1. 入院管理，包括入院登记和住院查询； 2. 出院管理，包括出院登记和查询及转病房等； 3. 病房管理；新添加病房，病房价格的调整等； 4. 结算中心；包括费用预交，实际缴费，费用查询等； 5. 护理中心；包括住院历史，用药历史等； 6. 统计中心；涵盖病人统计和病房统计； 7. 药品管理；药品入库，药品发放，退药，入库等； 8. 用户管理；系统用户的注册登录等； 9. 医生管理；为医院录入医生的相关信息； | | | | | | | | | | | |
| 对学  生知  识与  能力  要求 | 要求学生：  （1）熟悉一门高级语言编程：Java、JavaScript （2）熟悉J2EE技术，SSM、Spring Boot等  （3）了解前端Vue技术 （4）熟悉MYSQL、REDIS | | | | | | | | | | | |
| 预期  目标 | 1. 应完成的工作：完成出入院管理和病人及病房管理等模块的设计，能够帮助医护人员录入管理、帮助病人检索相关的医护信息、入院。 2. 预期目标：加快办理入院、出院的速度，实现病人资料的全信息化管理；准确记录病人和病床的信息，提供检索功能，即时掌握病人的各种信息，提高工作效率等。 3. 成果形式：   （1）论文 （2）软件 | | | | | | | | | | | |
| 时间  进度 | 2020-12-08——2020-12-19：准备工作：阅读相关文献和资料，熟悉所做课题内容，准备论文的绪论部分。  2020-12-20——2021-01-11：需求分析和设计，完成论文的绪论部分。  2021-01-12——2021-03-20：完成程序的主要界面和后台代码的实现，同时进行论文正文相应部分的撰写。  2021-03-21——2021-04-30：完善程序的所有功能，并继续撰写完成论文。  2021-05-01——2021-06-07：进行论文内容、格式等的检查、完善与修正。 | | | | | | | | | | | |
| 系（教研室）主任  签字 | | 年 月 日 | | | | 主管院长  签字 | | | 年 月 日 | | | |

西安邮电大学本科毕业设计（论文）开题报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 靳飞虎 | 学号 | 04173052 | 专业班级 | 软件1702 |
| 指导教师 | 拓守恒 | 题目 | 基于Springboot框架技术的住院管理系统设计与实现 | | |
| 选题目的  当今,我国的医疗改革逐步深化,医院分类管理已开始实施,对医院管理提出了新的更高的要求。迫切需要医院管理创新理论的指导,必须深入开展医院管理创新的理论研究和实践探索。在Spring Boot出现之前，医院入院住院系统主要是通过Spring、Spring MVC 、Mybatis 实现，这时候项目依赖多，配置文件随着需求的增长随之增多，不利于项目的维护升级。使用Spring Boot实现，采取starter来简化依赖的获取，它使用了大量的自动配置，可直接使用模板类；另外，再此基础上又提供了单配置文件可以修改默认实现的自动配置，非常易于维护和读取。再者由于早期的医院信息系统都是由JSP/Servlet实现，因此在维护和性能上已经跟不上时代的发展，因此本项目采用Spring Boot、Vue的前后端分离的框架对医院信息系统进行了重构，在功能上实现了一般性的需求。 | | | | | |
| 前期基础  已学课程：数据结构、面向对象程序设计Java、软件工程、操作系统、计算机网络。  掌握的工具：版本控制工具Git、IntelliJ IDEA、VScode、StarUml、PowerDesigner  资料累积：《Springboot开发实战》、《深入浅出Mybatis技术原理与解析》、《Spring技术内幕》、《Redis实战》、《Vue开发实战》。  软硬件环境：Win10系统一台（已安装JRE），CentOS云服务器一台。 | | | | | |
| 要研究和解决的问题  为满足住院管理的需要，需要实现如下功能：  图1 系统功能结构图   1. 登陆模块，医生或医生助理登入该系统，帮助病人办理相关住院和联络医生的一些必要要求，为病人的查询需要，提供帮助。 2. 入院管理，在医生诊断后，较为严重的病人如果需要进行住院治疗，需要该模块进行病房的分配和相关的医疗人员的护理。 3. 出院管理，病人在进行一段时间的住院治疗观察后，在医生的观察后，情况良好批准出院后，回收相应的病房，并派遣相关的护理人员前去处理消毒，留待需要。 4. 医生管理，将医院的所有医生的相关信息录入系统，一是在病人前来看病时，为病人分配相应的医生，二是便于管理医生，在新医生入行或者有医生离职时，能更有效，更即时的反应 5. 病房管理，当病人需要住院时，分配相应的病房。 6. 统计中心，在每日和每月的报表时，由此功能实现。 7. 用户管理，系统用户管理，包括患者、护士、医生、系统管理员等角色，用于注册和添加新患者或者医生。 8. 结算中心，患者看完病后，在结算中心结算相应的花费，可以查询以往的缴费记录，提供更为精确的费用查询。 9. 系统配置，提供系统运行日志的查询和系统参数的设置。 | | | | | |
| 工作思路和方案  1.工作思路  课题可以分为以下几个阶段，分阶段完成：  (1)需求分析，根据要完成的项目需求，分析要完成课题所需的数据库表。  (2)界面设计。  (3)具体业务代码编写。  (4)文档编写，对课题进行总结，编写每一个部分的文档。  2.技术方案  项目整体结构如图2所示：    图2 组织架构图  项目整体采用Spring Boot做为项目框架选型，前端使用Vue和Axios实现前端请求和页面的动态渲染，后端采用Spring MVC做请求分发控制器，Spring做业务处理，嵌入Spring Security作为认证和授权处理，数据访问使用Mybatis技术框架作为支撑，数据库选型为Mysql和Redis，以Mysql作为主要的存储，Redis作为缓存共同作用。  前端使用Vue作为页面的渲染，利用Axios异步请求库发送Ajax请求，后端整体采用MVC设计模式，通过SpringMVC接受请求，然后交由Spring进行具体的业务处理和事务控制，然后由底层的Mybatis实现具体的数据库访问控制，处理完成后，发送响应给Axios的请求接收，然后反映到Vue 的实例属性中，通过Vue的生命周期对页面进行动态的渲染，以达到较好的用户体验。  3.进度计划  2020-12-08——2020-12-19：准备工作：阅读相关文献和资料，熟悉所做课题内容，准备论文的绪论部分。  2020-12-20——2021-01-11：需求分析和设计，完成论文的绪论部分。  2021-01-12——2021-03-20：完成程序的主要界面和后台代码的实现，同时进行论文正文相应部分的撰写。  2021-03-21——2021-04-30：完善程序的所有功能，并继续撰写完成论文。  2021-05-01——2021-06-07：进行论文内容、格式等的检查、完善与修正。 | | | | | |
| 指导教师意见  签字： 年 月 日 | | | | | |

西安邮电大学毕业设计(论文)成绩评定表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 靳飞虎 | | | 性别 | 男 | | 学号 | | 04173052 | | 专业  班级 | | 软件1702 | |
| 课题  名称 | 基于Springboot框架技术的住院管理系统设计与实现 | | | | | | | | | | | | | |
| 前期  成绩 | 背景与目标  (目标1,30) | 参考文献  (目标7,20) | | | | 设计方案  (目标2,30) | | | | 撰写质量  (目标6,20) | | | | 总分 |
|  |  | | | |  | | | |  | | | |  |
| 中期  成绩 | 完成情况  (目标4,20) | 关键问题  (目标2,30) | | | | 前期问题改进  (目标7,20) | | | | 方案创新性与合理性(目标3,30) | | | | 总分 |
|  |  | | | |  | | | |  | | | |  |
| 指导  教师  意见 | （从项目实现情况、创新性、毕设过程中学生的学习能力、翻译的质量等方面进行考核）  指导教师(签字)**：\_\_\_\_\_\_\_\_\_**  2021年5月28日 | | | | | | | | | | | | | |
| 项目论证  (目标1,50) | 创新意识  (目标3,10) | | | | 自学能力  (目标7,30) | | | | 译文  (目标7,10) | | | | 总分（百分制） |
|  |  | | | |  | | | |  | | | |  |
| 评阅  教师  意见 | （从设计方案的合理性、测试设计、论文质量和对社会的影响等方面进行考核）  评阅教师(签字)：\_\_\_\_\_\_\_\_\_\_\_ 2021年5月31日 | | | | | | | | | | | | | |
| 设计方案  (目标2,20) | 测试方案  (目标4,20) | | | | 社会影响  (目标5,10) | | | | 撰写质量  (目标6,50) | | | | 总分（百分制） |
|  |  | | | |  | | | |  | | | |  |
| 验收  小组  意见 | （从设计方案的实现程度、创新性、项目代码完成情况等方面进行考核）  验收教师(签字)：\_\_\_\_\_\_\_\_\_\_\_\_ 2021年 月 日 | | | | | | | | | | | | | |
| 设计方案  (目标2,40) | | 创新意识  (目标3,20) | | | | | 完成情况  (目标4,40) | | | | 总分(百分制) | | |
|  | |  | | | | |  | | | |  | | |
| 答辩  小组  意见 | （从答辩过程体现出的创新意识，项目对社会影响的论述、阐述的项目实现过程、回答问题等方面进行考核）  答辩小组组长(签字)：\_\_\_\_\_\_\_\_\_\_\_\_ 2021年6月7日 | | | | | | | | | | | | | |
| 创新意识  (目标3,20) | | 社会影响  (目标5,20) | | | | | 答辩质量  (目标6,60) | | | | 总分(百分制) | | |
|  | |  | | | | |  | | | |  | | |
| 评分比例 | 前期情况总分(10％) 中期情况总分 (10％) 指导教师评分 (20％)  评阅教师评分(25％) 验收小组评分 (25％) 答辩小组评分 (10％) | | | | | | | | | | | | | |
| 学生总评  成绩 | 百分制成绩 | |  | | | | 等级制成绩 | | | | |  | | |
| 答辩委员会意见 | 毕业论文(设计)最终成绩(等级)**：\_\_\_\_\_\_\_\_\_\_\_\_**  学院答辩委员会主任(签字、学院盖章)**：**  2021年6月11日 | | | | | | | | | | | | | |

摘 要

传统手工处理方式在进行住院相关的数据管理过程中存在工作量大、易出错、检索不便等问题。为解决这些问题，本文设计并实现了一个住院管理系统，其不仅能提高医院的信息化方面的管理，提升医院住院部的工作效率及医疗方面的质量，还能规范医院的业务流程，最大程度的避免人为处理的一些失误，提高医疗保障处理的速度和准确性。

系统采用了B/S模式，基于MVC的三层架构，主要由Spring Boot框架和Vue框架作为技术支撑实现，系统的权限拦截凭借Spring Security控制，至于数据存储方面以MySQL作为主要存储、以Redis为辅助存储。在系统功能上划分为登陆模块、收费模块、药品管理模块、住院管理模块、统计中心和系统管理模块。项目能有效的对医院信息进行处理，改变了传统医院的信息管理方式，提高了医院信息处理的效率，方便了医务工作者对医院的信息化管理。

关键词：Spring Boot框架；Vue；住院信息系统；效率

ABSTRACT

　　The traditional manual processing method has problems such as high workload, error-prone and inconvenient retrieval in the process of managing inpatient-related data. In order to solve these problems, this paper designs and implements an inpatient management system, which not only improves the information management of the hospital, enhances the efficiency and medical quality of the hospital inpatient department, but also standardizes the business process of the hospital, minimizes some mistakes of human processing, and improves the speed and accuracy of medical processing.

　　The system adopts the B/S model, based on MVC three-tier architecture, mainly by the Spring Boot framework and Vue framework as the technical support to achieve, the system's permission interception by virtue of Spring Security control, as for data storage to MySQL as the main storage, to Redis as secondary storage. The system is divided into login module, billing module, drug management module, inpatient management module, statistics center and system management module. The project can effectively process hospital information, change the traditional hospital information management method, improve the efficiency of hospital information processing, and facilitate the information management of hospitals by medical workers.

**Key words：**Spring Boot Framework; Vue; Hospital Information System; Efficiency

目 录

[第一章 概述 1](#_Toc21700)

[1.1课题背景 1](#_Toc11215)

[1.2国内外研究现状 2](#_Toc14677)

[1.3课题研究目标 2](#_Toc4802)

[1.4本文内容安排 3](#_Toc4783)

[第二章 系统需求 5](#_Toc30653)

[2.1业务需求 5](#_Toc15811)

[2.1.1 办理入院手续 5](#_Toc24848)

[2.1.2 缴费模块 6](#_Toc22589)

[2.1.3医嘱流程 7](#_Toc19433)

[2.1.4出院登记 7](#_Toc12284)

[2.2系统功能需求 7](#_Toc9837)

[2.2.1用户登录 8](#_Toc31564)

[2.2.2入院登记 9](#_Toc13196)

[2.2.3病房管理 9](#_Toc2021)

[2.2.4结算中心 10](#_Toc28283)

[2.2.5统计中心 11](#_Toc16132)

[2.2.6药品管理 11](#_Toc29434)

[2.2.7开立医嘱 12](#_Toc3858)

[2.2.8出院登记 12](#_Toc22737)

[2.2.9系统管理 13](#_Toc4016)

[2.3系统非功能需求 13](#_Toc9256)

[2.3.1性能需求 13](#_Toc6966)

[2.3.2适应性需求 13](#_Toc14753)

[2.3.3扩展性需求 14](#_Toc12493)

[第三章 系统设计 15](#_Toc31851)

[3.1设计决策 15](#_Toc19677)

[3.1.1技术方案 15](#_Toc11672)

[3.1.2工具及环境平台 16](#_Toc4449)

[3.2体系结构设计 16](#_Toc8066)

[3.2.1逻辑架构 16](#_Toc9821)

[3.2.2开发架构 17](#_Toc9478)

[3.2.3物理架构 18](#_Toc14991)

[3.3界面设计 18](#_Toc9175)

[3.4数据存储设计 20](#_Toc3849)

[3.4.1数据库模型设计 20](#_Toc10815)

[3.4.2数据库表设计 21](#_Toc3175)

[3.5业务模块设计 28](#_Toc11266)

[3.5.1登录模块 29](#_Toc5801)

[3.5.2 入院登记模块 30](#_Toc26433)

[3.5.3 立行医嘱模块 31](#_Toc15619)

[第四章 系统开发 34](#_Toc14735)

[4.1开发环境 34](#_Toc3367)

[4.2关键技术 34](#_Toc11927)

[4.2.1 Spring Boot 34](#_Toc19919)

[4.2.2 Spring Security、JWT 35](#_Toc7425)

[4.2.3 Vue 36](#_Toc17035)

[4.3开发成果 37](#_Toc2941)

[4.3.1登陆模块 37](#_Toc27432)

[4.3.2入院登记 37](#_Toc10006)

[4.3.3病房管理 38](#_Toc809)

[4.3.4结算中心 39](#_Toc7492)

[4.3.5统计中心 40](#_Toc27177)

[4.3.6药品管理 42](#_Toc32589)

[4.3.7医嘱模块 43](#_Toc9241)

[4.3.8出院登记 44](#_Toc16888)

[4.3.9系统管理 44](#_Toc22721)

[第五章 系统测试 46](#_Toc17318)

[5.1测试设计 46](#_Toc13288)

[5.2测试用例及测试记录 46](#_Toc21569)

[5.2.1登录模块 46](#_Toc21513)

[5.2.2结算中心 46](#_Toc21929)

[5.2.3入库模块 47](#_Toc9101)

[5.2.4入院模块 47](#_Toc135)

[5.2.5医嘱模块 47](#_Toc6357)

[5.2.6医生模块 48](#_Toc30889)

[5.2.7用户模块 48](#_Toc31647)

[5.3测试结果及结论 49](#_Toc30041)

[第六章 展望与总结 50](#_Toc650)

[6.1本文工作总结 50](#_Toc9546)

[6.2未来工作展望 50](#_Toc29125)

[结束语 51](#_Toc29456)

[致 谢 52](#_Toc27695)

[参考文献 53](#_Toc13889)

# 第一章 概述

## 1.1课题背景

近年来，我国信息产业发展迅速。随着计算机技术的迅速发展，价格正在下降。计算机已广泛应用于医疗、管理、教学、科研等各个方面。医院也同样面临着信息时代的挑战，医院信息管理的网络化与计算机化已经成为现代医院运营过程中必不可少的基础设施与支持环境。医院面临着激烈的市场竞争。要想在竞争中取胜，就必须采用合理并且先进的管理方法和手段[1]。计算机最大的优点是可以用于信息管理，利用计算机进行信息控制，不仅提高了效率，而且大大提高了安全性。医院作为一种服务型的机构，需要统计和保存的病人和医生资料繁多，包含很多的信息数据的管理，基于这种目的，有必要建立一个医院管理系统，使医院管理的工作规范化，系统化，程序化，避免医院管理的随意性，提高信息处理的速度和准确性[2]，及时准确的查询和录入病人的入院、出院情况。

医院信息化建设既是医院管理的需要，也是医院高效运行的需要。随着医改的深入，医疗市场的竞争日趋激烈。因此，转变医院管理模式迫在眉睫，进入信息化管理的时代。从长远来看，信息化模式能够带来以下好处：

1. 在一定程度上能够缩短患者的就诊时间，提高患者的看病提速的最佳效果。患者可以通过登陆系统结算中心查看缴费记录和缴费金额。
2. 规范了医院管理的流程，使用随时了解医院的医疗情况、等各种医疗用具情况，为医院的可持续发展，提供快速准确的支持，提高医院的运行效率。
3. 增加收支，避免漏记漏收，人情看病现象的发生，通过报表统计等可视化工具，为医院的发展，提高管理水平、辅助决策，使医院经济朝着高效益、高质量的方向发展。

## 1.2国内外研究现状

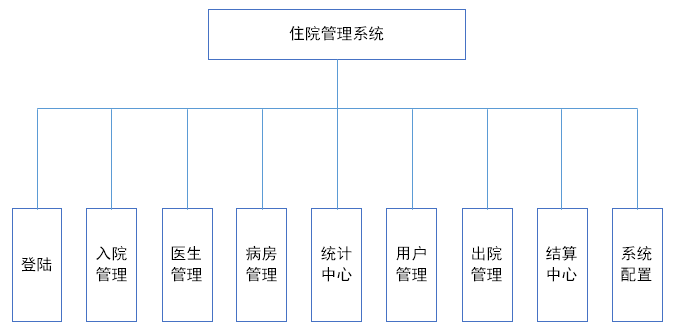
在60年代国外的医院管理系统开始起步，在80年代进入一个发展的上升期，在90年代则拥有了城市的一套系统。以美国为例，美国的住院管理系统软件是以病房管理为出发点，紧接着由于需求的增加，逐步向财务收费系统、辅助检查系统等等系统进行扩展，电子病历系统则在90年代的时候已经基本成熟。2009—2019年间，中美两国政府均出台了不同的政策来推进以电子病历为核心内容的HIT行业的发展，极大地促进了本国电子病历的推广使用[3]。

我国的医院信息化相对于国际晚了20年，但是我国的医疗管理系统的速度却大大超过了国际水平。从最初的病案统计起步，又以财务管理为重心进行发展，由于中国医院的国情、民情的特殊性，国际上的公司的系统进入国内的成功案例比较少，再这样的环境下我国自主设计拥有了发展空间。二十世纪至今，从无到有，从功能单一到多功能一体化，我国自产的医院管理系统的发展速度是日新月异的。

在早期，医院信息系统主要是基于Java EE，主要通过Servlet/JSP实现，实现比较困难，需要人工从请求中取请求参数，视图和数据仍然纠缠在后端，必须由后端人员进行整合。随着Java Web技术的迅猛发展,特别是在JavaScript脚本语言得到广泛应用后,涌现出了众多基于Java Web应用的前端开发框架[4]。在Spring Boot出现之前，医院入院住院系统主要是通过Spring实现，这时候项目依赖多，配置文件随着需求的增长随之增多，不利于项目的维护升级。随着Web应用的进一步演化出现了前后端分离开发模式，这时候新兴了一些强大的JavaScript框架如Vue，采用了MVVM设计模式[5]的优势，提高了Web开发效率。Vue的组件化开发思想非常适合信息化的管理系统，构建简洁明了，层次分明，使管理者能快速的一目了然掌握情况和规律,并且从减少HTTP请求次数、组件复用、安全防御等方面对系统进行优化,实现简洁美观、操作界面的亲和力强以及便于后期扩展的信息管理系统[6][7]。而后台采用使用Spring Boot、Redis实现。Redis作为一个高性能的缓存数据库，在性能上面表现很好[8]，很适合查询请求较多的场景；Spring Boot采取starter进行依赖的递归获取和版本控制，它使用了大量的自动配置[9][10]，可直接使用，另外，其内置了模板、工厂、观察者等多种设计模式，对于编写企业级应用的实现和维护降低了门槛[11]。

## 1.3课题研究目标

由于早期的医院信息系统都是由JSP/Servlet实现，因此在维护和性能上已经跟不上时代的发展，因此本项目采用Spring Boot、Vue的前后端分离的框架对医院信息系统进行了重构,采用Redis作为辅助存储，提高访问效率。为满足一般性医院住院系统管理的需要，主要核心功能如下图1.1所示：

图1.1 系统功能结构图

1. 登陆模块，医生或医生助理登入该系统，帮助病人办理相关住院和联络医生的一些必要要求，为病人的查询需要，提供帮助。
2. 入院管理，在医生诊断后，较为严重的病人如果需要进行住院治疗，需要该模块进行病房的分配和相关的医疗人员的护理。
3. 出院管理，病人在进行一段时间的住院治疗观察后，在医生的观察后，情况良好批准出院后，回收相应的病房，并派遣相关的护理人员前去处理消毒，留待需要。
4. 医生管理，将医院的所有医生的相关信息录入系统，一是在病人前来看病时，为病人分配相应的医生，二是便于管理医生，在新医生入行或者有医生离职时，能更有效，更即时的反应
5. 病房管理，当病人需要住院时，分配相应的病房。
6. 统计中心，在每日和每月的报表时，由此功能实现。
7. 用户管理，系统用户管理，包括患者、护士、医生、系统管理员等角色，用于注册和添加新患者或者医生。
8. 结算中心，患者看完病后，在结算中心结算相应的花费，可以查询以往的缴费记录，提供更为精确的费用查询。
9. 系统配置，提供系统运行日志的查询和系统参数的设置。

## 1.4本文内容安排

本文从住院信息系统的需求分析出发，一步一步的介绍具体的功能要点，从业务需求深入到功能需求，细化每个需求的要点；然后讨论了系统的非功能需求。最后通过详细的系统设计方案将任务目标具现化，并在其中简单介绍了项目开发的关键技术要点。最后通过基准测试，测试结果良好，最后在文章末尾进行了项目总结。

# 第二章 系统需求

## 2.1业务需求

系统总体目标为完成供医院住院部使用的一套提供信息化的管理方式，其核心内容就是协助医生、护士完成日常医疗工作，主要为图2.1所示业务流程；

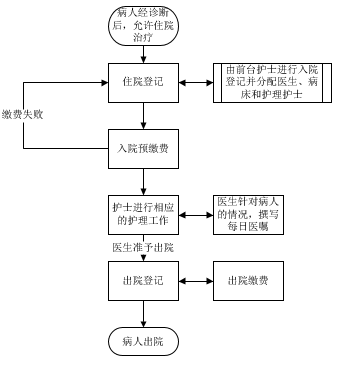


图2.1 住院流程图

从参与者的角度看，总共可以分为四类角色，病人、护理护士、前台护士、医生。病人经过门诊部系统的诊断后，判断可以进行住院治疗，经过患者同意后，由前台护士办理入院登记手续，包括分配一个主治医师，一个专业的护理护士和一个病床，并预缴相关的住院手续费用。然后由护士带领病人到给他分配的病床，并做一些相应的入院检查工作，然后并将其交给主治医生。之后，医生查验后，判断给该病人分配哪些药品，或者要安排手术等等，将这些医嘱发给护理护士，由护士进行确认并安排流程。

系统流程上主要由入院登记、缴费手续、立行医嘱、出院登记这几步核心业务构成。

### 2.1.1 办理入院手续

病人经由门诊部医生诊断后，确认可以进行住院治疗，需要到门诊部前台办理入院手续，由前台护士进行关于病人信息和住院相关信息的录入，其流程如下图2.2所示；

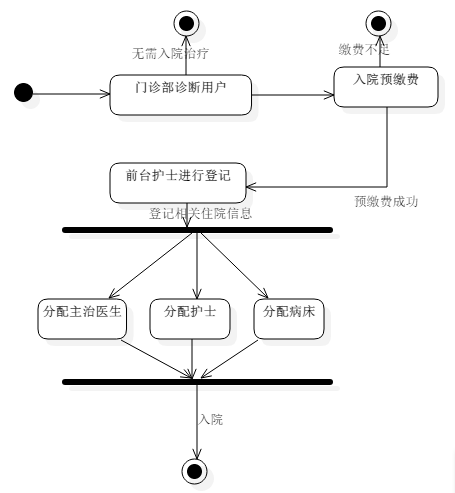


图2.2 住院活动图

### 2.1.2 缴费模块

病人从一般情况来说，在整个住院周期中，需要上缴两次费用。初次是在住院登记前，必须预缴至少一定数量的金额，以便进行安排床位、护士、医生进行住院治疗介入；最后一次则是在办理出院手续时，需要额外进行结算一次，这时候病人可能要补交或者由住院部返回预缴费时剩下数额。当然，在住院期间，也可能会出现紧急手术或者申请药品等情况，如果病人预缴余额不足时，需要由负责护士提醒病人进行再次缴费。具体流程如图2.3所示；

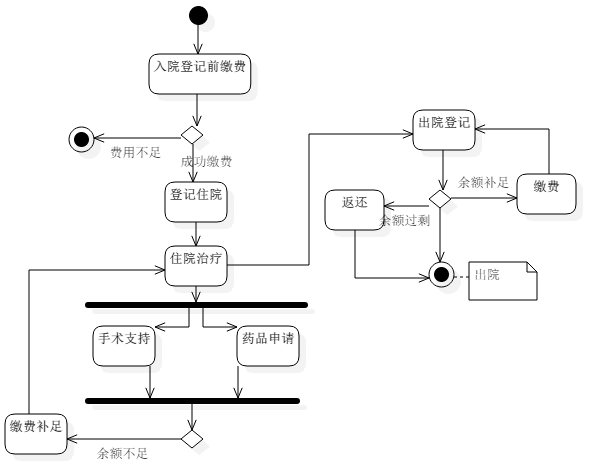


图2.3 缴费活动图

### 2.1.3医嘱流程

医嘱模块，其主要发生于在病人住院期间，负责该病人的护士和医生之间。

由于最贴近病人的是护士，因此护士比医生能更加清楚医生真正的生理状态，而医生主要业务为根据护士为病人护理的一些医疗特征进行相应的一些方案诊治，也就是说，病人很多时候是根据护士提供的医疗单去思虑病人应该接受的诊疗方案，而具体的实行则又需要护士去亲身力行，因此二者之间是互相协同工作的。其流程可以用图2.4来表示；

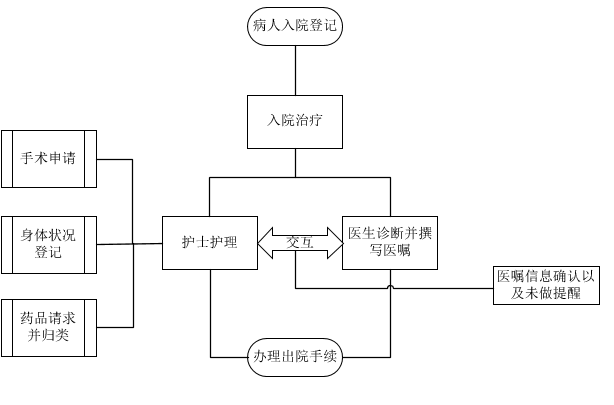


图2.4 医嘱模块流程图

### 2.1.4出院登记

出院登记业务，和住院手续类似，但又有些不同，具体内容可以细分为：病人费用结算、明细费用查询、病历信息归档，如果条件支持的话，最好在打印一个出院单，以便患者能够进行核对和查看，出院登记比较简单，就不需要流程图来演示，其主要内容如下；

1. 查询病人的所有住院治疗费用，比如床铺费、医药费、护理费等等，如果有手术需要的话还需要相应的费用叠加。
2. 进行结算。主要分为两种情况，一是患者余额足够，这时候需要住院部将其余额以某种形式返还给用户，如今一般为电子货币。二是患者所预缴的费用不足，前台需要提醒用户要补交额外的费用，之后病人登记出院。
3. 善后工作。主要是病床的回收和病人信息的归档，保存相应的治疗信息，医嘱信息和缴费信息等。

## 2.2系统功能需求

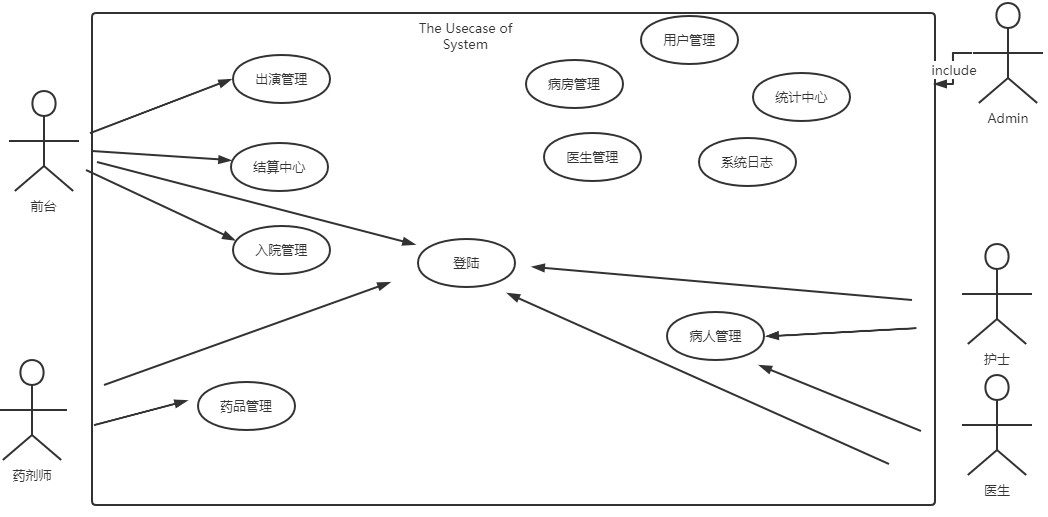


图2.5 系统总用例图

表2.1 系统用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **用例名称** | **用例标识符** | **用例描述(功能说明)** |
| 1 | 用户登录 | HIS\_01\_LOGIN | 根据角色的不同，登录后显示不同的登录页面 |
| 2 | 入院登记 | HIS\_01\_IN | 登记病人的个人信息、住院信息 |
| 3 | 病房管理 | HIS\_01\_WARD | 新增病房、或者对病房价格进行调整 |
| 4 | 结算中心 | HIS\_01\_PAY | 缴费、缴费记录查询、结算 |
| 5 | 统计中心 | HIS\_01\_STATISTIC | 病人统计、病房统计 |
| 6 | 药品管理 | HIS\_01\_MEDICINE | 添加、分发、回收 |
| 7 | 开立医嘱 | HIS\_01\_NOTE | 书写医嘱、执行医嘱 |
| 8 | 出院登记 | HIS\_01\_OUT | 登记出院信息 |
| 9 | 系统管理 | HIS\_01\_SYSTEM | 日志查看、用户注册、医生管理、护士注册 |

### 2.2.1用户登录

本节详细描述帖子管理功能的需求。

* 用例描述符：HIS\_01\_LOGIN
* 用例名称：用户登录
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：该系统所有角色
* 涉众

管理员、医生、护士：角色登录后，显示不同的需求界面，

以供工作使用。

* 前置条件：该角色账号已经由管理员进行了注册。
* 后置条件：角色登陆成功，显示出符合自己角色预期的界面。
* 主成功场景：

不同的用户角色登录后，显示不同的界面菜单。

比如管理员，在登陆后，菜单界面会显示出包括入院、出院、结算、用户注册、医生登记、系统日志等所有的菜单。

* 特殊需求：无

### 2.2.2入院登记

本节详细描述入院登记功能的需求。

* 用例描述符：HIS\_01\_IN
* 用例名称：入院登记
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：前台护士、病人
* 涉众

前台护士：帮助病人登记病人的个人信息，并选择相关的主治医师和护理护士和病床，以便后续的进一步观察治疗。

病人：主要为前台护士提供自己的个人信息和病历单，由此交由前台护士判断如何分配。

* 前置条件：该角色成功登入系统。
* 后置条件：生成入院信息、即分配病房、护士和主治医师。
* 主成功场景：

1. 病人提供个人信息和病历，前台护士选择相应的病床和主旨医生、护士，登入生成入院登记单。
2. 登记成功后，在查询页面可以正常查询到该病人的入院信息。

* 特殊需求：无

### 2.2.3病房管理

本节详细描述病房管理功能的需求。

* 用例描述符：HIS\_01\_WARD
* 用例名称：病房管理
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：管理员
* 涉众

管理员：可以添加一个病房，并指定这个病房的类型，比如ICU（重症急救室）、普通病房等；不同的类型病房生成的病房个数是不同的；可以动态的调整不同类型病房的价格。

* 前置条件：该账号必须是管理员角色，并且已经成功登陆。
* 后置条件：新增的病房会永久的存入数据库，可以在前台病人登记时显示。
* 主成功场景：

1. 用户输入病房的基本信息，勾选类型后，会在数据库生成该病房的信息，及对应的病床；
2. 可修改病房的病床价格。

* 特殊需求：无

### 2.2.4结算中心

本节详细描述结算中心功能的需求。

* 用例描述符：HIS\_01\_PAY
* 用例名称：结算中心
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：前台护士、病人
* 涉众

前台护士：用户住院登记后，需要再进行办理住院预缴费，以便进行之后的治疗工作；在用户住院期间，可能会出现余额不足的情况，也需要办理缴费；最后在出院登记时，需要对总的缴费和扣费信息进行结算。

病人：主要就是去前台缴费。

* 前置条件：该角色账号必须是前台护士，并且登陆成功。
* 后置条件：缴费成功，数据库生成缴费记录。
* 主成功场景：

不同的用户角色登录后，显示不同的界面菜单。

比如管理员，在登陆后，菜单界面会显示出包括入院、出院、结算、用户注册、医生登记、系统日志等所有的菜单。

* 特殊需求：无

### 2.2.5统计中心

本节详细描述统计中心功能的需求。

* 用例描述符：HIS\_01\_ADMIN
* 用例名称：统计中心
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：管理员
* 涉众

管理员：提供了根据病人、病房两种统计方式。主要是按照不同科室进行区别。

* 前置条件：该账号必须是管理员角色，并且已经成功登陆。
* 后置条件：无
* 主成功场景：

1. 病人统计，主要是科室对住院的病人进行分类，分别统计入院人数和出院人数，以表格和条形统计图和饼图方式显示到界面上，留待进一步的决策。
2. 病房统计，和病人统计类似，不过这一项主要是统计病床的使用率。

* 特殊需求：无

### 2.2.6药品管理

本节详细描述药品管理功能的需求。

* 用例描述符：HIS\_01\_MEDICINE
* 用例名称：用户登录
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：管理员
* 涉众

管理员：管理员登陆后，可以安排进行药品入库、药品发放、退药处理、入库查询、库存查询功能。

* 前置条件：管理员账号登陆成功。
* 后置条件：药品信息成功入库或成功发放或回收。
* 主成功场景：

1. 药品入库成功，在库存查询可以显示出来。
2. 药品发放，由管理员进行药品申请的确认，并进行药品的发放，一般而言是递交给护士。
3. 退药处理，与药品发放类似，回收护士多申请的药品，并将其重新入库。

* 特殊需求：无

### 2.2.7开立医嘱

本节详细描述开立医嘱功能的需求。

* 用例描述符：HIS\_01\_NOTE
* 用例名称：开立医嘱
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：医生、护士
* 涉众

医生：医生登入系统后，在首页会显示负责的病人信息，可以查看对于该病人历史写入的医嘱信息并可以进行新增医嘱。

护士：可以接收到医生写入的新的医嘱信息，需要对此进行确认并在此基础上予以执行，包括检测病人的体征信息和药品或手术申请。

* 前置条件：医生或护士成功登入该系统
* 后置条件：主治医生撰写医嘱，护士对此进行确认和执行。
* 主成功场景：

1.医生根据病人的相关身体特征表现，撰写医嘱，并签字去人；对应的护理护士收到该医嘱时，对此进行确认。如果有疑问，可以和相关的医生进行商谈。

* 特殊需求：无

### 2.2.8出院登记

本节详细描述出院登记功能的需求。

* 用例描述符：HIS\_01\_OUT
* 用例名称：出院登记
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：前台护士
* 涉众

前台护士：病人提供信息，然后由前台护士进行信息登记；之后需要进行结算。

* 前置条件：该角色账号已经由管理员进行了注册。
* 后置条件：角色登陆成功，显示出符合自己角色预期的界面。
* 主成功场景：

1. 病人信息登出成功，会安排人员进行床位的回收、人员的回收。
2. 住院状态会被设置为出院状态，并可以进行下一步的出院结算。

* 特殊需求：无

### 2.2.9系统管理

本节详细描述系统管理功能的需求。

* 用例描述符：HIS\_01\_SYSTEM
* 用例名称：系统管理
* 范围：业务用例
* 级别：用户目标级别
* 主要角色：管理员
* 涉众

管理员：进行包括用户注册、(医生、护士)信息登记、日志系统。

* 前置条件：管理员成功登入系统
* 后置条件：用户、医生、护士等信息成功登记
* 主成功场景：

1. 系统配置，可以根据日志的不同级别，查看系统运行的日志。
2. 用户注册，包括护士账号、服务前台账号、医生、管理员账号的注册功能。
3. 医生管理，主要是对该住院部医生信息的录入和查看，并能动态修改入职和离职状态。

* 特殊需求：无

## 2.3系统非功能需求

……….

……..

……..

……..

……..

……..

……..

……..

……..

### 2.3.1性能需求

一般性的，从医院的规模上考虑，中小型的住院系统内部最多只有几百名医生和护士，因此该系统的并发量只需要支撑上千的QPS（Queries Per Second）即可。因此，采用的方案为利用Nginx做Tomcat集群，通过Nginx将请求进行分流，分别负载到不同的服务节点上进行处理。

### 2.3.2适应性需求

医院对住院系统是适应性方面主要包含两个方面，一是在性能上达到5s内响应即可，二是能够保证基本的运行状态并且在使用方面能够达到简洁方便，易于使用。

### 2.3.3扩展性需求

根据目前的需求分析，项目已经基本实现，但也难保后期可能会出现一些业务性的其他需求，因此系统要能够支持更快的迭代开发新的适应系统的需求。因此，针对这一方面，将原本几乎串行的编写方式（先写前端页面，之后后台再进行进一步业务实现）进行了修改，纳入了前后端分离开发的模式，这样就保证了并行开发，提高了效率，能更快的适应新需求的发展。

# 第三章 系统设计

## 3.1设计决策

### 3.1.1技术方案

总的来说，采用前后端分离的模式开发，这主要应用了Ajax异步请求技术，会有大量的请求通过Ajax进行发送。

首先，针对后台方面，应用的主要技术栈为Spring Boot、Spring、Spring MVC、Mybatis、Spring Security。简单解释一下，Spring是一个轻量级的Inversion Of Control（IOC）和Aspect Oriented Programming（AOP）容器框架，IOC寓意着控制反转，即互相依赖的对象之间并不是通过自己创建，而是容器在初始化的时候不等依赖对象请求，由容器主动传递给它，这样就实现了松耦合关系；采用AOP实现了面向切面编程，使得业务代码内就是业务逻辑，不掺杂其他的关注点比如事务、日志处理等。Spring MVC是控制层的框架主要就是根据不同URL接收不同请求的。Mybatis是数据访问层（Data Access Control）的框架，采用ORM映射关系，将原本和代码耦合的字段通过映射文件进行了解耦，在一定程度上减少了和数据库字段之间的耦合关系。Spring Security是针对系统权限控制的，通过一系列过滤器实现，主要和JWT进行结合使用。

前台主要技术为Vue、Vuex、Axios、LocalStorage，通过彼此结合实现。Vue是一种组件化的前端开发技术，将页面拆分为一个个组件提高了复用性。Vuex用来作为应用的状态管理，由于采用了组件化的思想，因此每个组件之间的数据是隔离的，在某些情况下，需要共享一个外部的状态，也即应用的状态，比如登录状态等，这时候Vuex是最佳实现。Axios主要封装了不同请求方法的异步请求库，LocalStorage是浏览器的缓存技术，会保存每个站点的非隐私的数据。

整体模型如图3.1所示，用户通过浏览器发送请求，前台是以Node为底层，利用Webpack构建的使用Vue技术的前台服务，通过Axios和后台服务器进行交互，后台主要由Java语言实现，通过Servlet提供服务，运行在Tomcat容器中，凭借Spring Security实现的各种过滤器实现权限校验，校验通过后会传递给Spring MVC的中转站，即DispatcherSevlet，通过其进行请求分发，并通过业务层Spring，数据访问层Mybatis实现业务需求，在返回给前台。如果需要保存系统的状态，需要由Vuex或者LocalStorage实现。

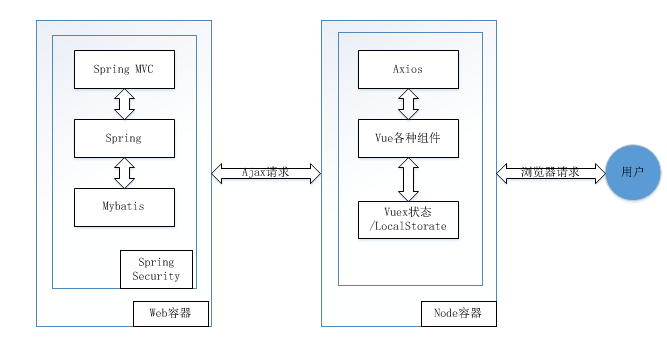


图3.1 技术方案图

### 3.1.2工具及环境平台

工具：IntelliJ IDEA、StarUML、PowerDesigner、Vscode、Sqlyog

平台：Windows10 JDK8

## 3.2体系结构设计

### 3.2.1逻辑架构

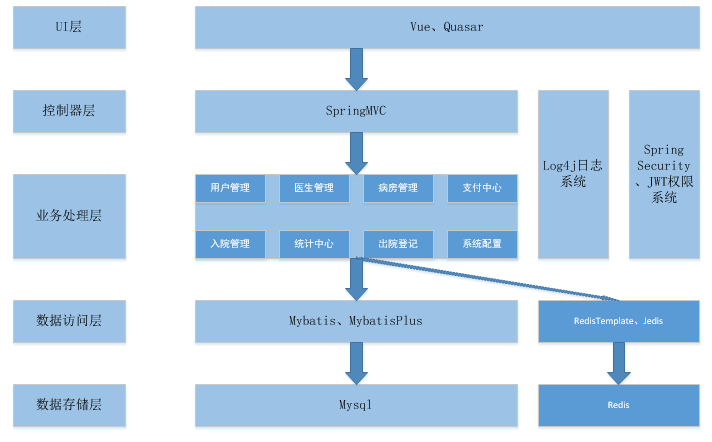


图3.2 逻辑架构图

整体采用Spring Boot做为项目框架选型，前端使用Vue和Axios实现前端请求和页面的动态渲染，后端采用Spring MVC做请求分发控制器，Spring做业务处理，嵌入Spring Security作为认证和授权处理，数据访问使用Mybatis技术框架作为支撑，数据库选型为Mysql和Redis，以Mysql作为主要的存储，Redis作为缓存共同作用。

### 3.2.2开发架构

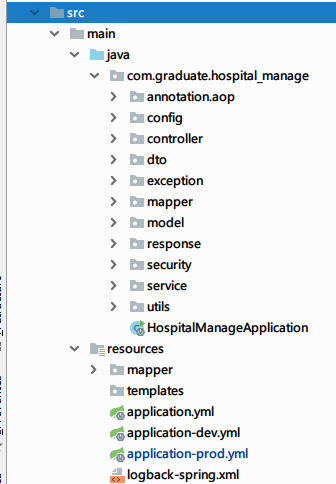


图3.3 开发目录结构图

整体分为两部分，java目录为Java类源代码代码目录；resources目录下主要放置着配置文件，包括系统配置文件，和框架配置文件。

类目录中，annotation.aop包是关于缓存实现；config主要是系统配置；controller为后台请求接受的入口，主要通过dto来接受前台发送的参数；exception是系统全局异常处理的类；response主要是对返回值进行统一，并对返回状态进行了约束；security是Spring Security的核心，主要和JSON Web Token（JWT）结合使用；utils是项目使用的一些工具类；HospitalManageApplication是项目的入口，即启动类。

### 3.2.3物理架构

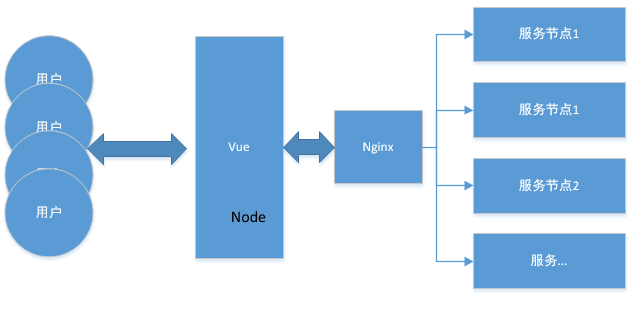


图3.4 物理架构图

简单解释一下，处于性能上和准确性上的考究，并未参考微服务的设计实现。项目整体是通过服务节点的集群实现，这样在一定程度上实现了水平扩展，服务节点数量在理想条件下可以无限叠加下去。另一方面，由于Nginx服务器的高性能，已经能够实现上万连接的请求处理，并不需要做额外的请求限制，只需要对数据库连接池的配置和Java虚拟机的配置作进一步优化即可。

## 3.3界面设计

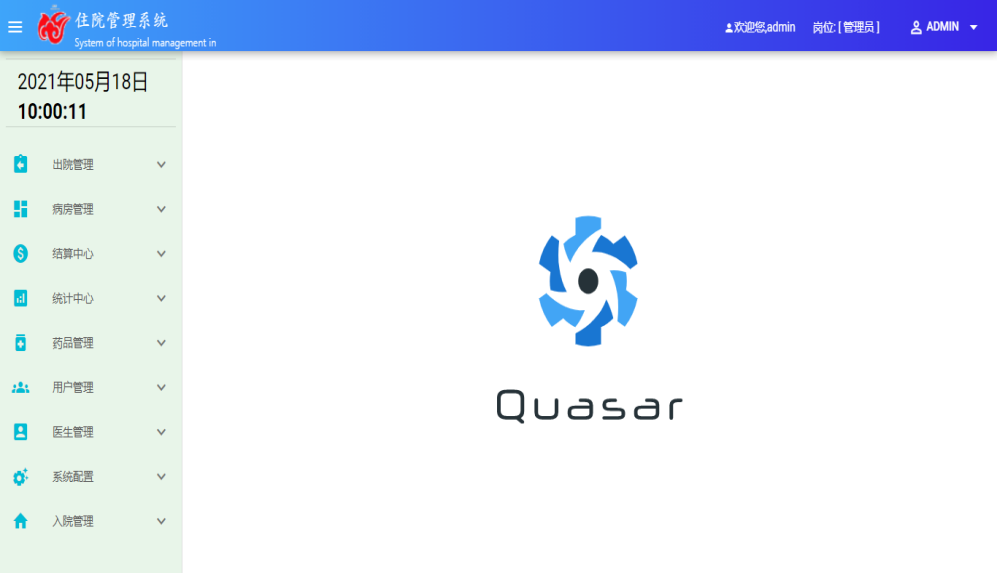


图3.5 界面原型图

如上3.5界面原型所示，主要由三部分构成，顶部为logo和副菜单功能，左侧提供项目的核心业务菜单，右侧是主要的内容显示区域。用户点击左侧按钮，视图出口都会显示到右侧的容器中，如下图3.6所示的情况；

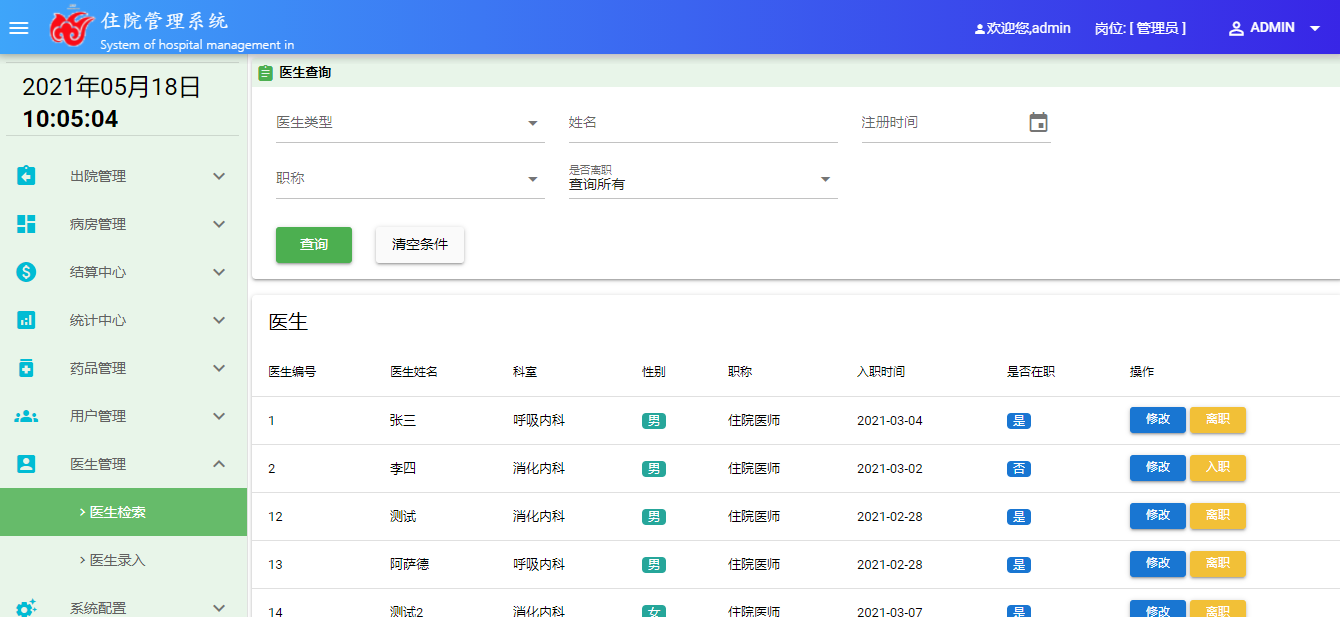


图3.6 菜单样例图

为了适应不同种类大小的屏幕，采用了适应性的布局，在界面小到一定程度时，左侧菜单会退化为弹出按钮，如图3.7所示；



图3.7 菜单状态变化图

上侧导航栏主要提供个人职位信息的显示和登入登出、修改密码的功能，图3.8所示；

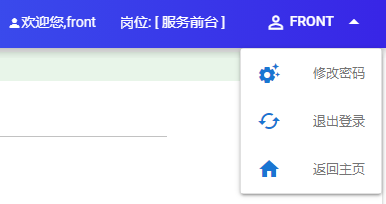


图3.8 上侧副菜单图

## 3.4数据存储设计

### 3.4.1数据库模型设计

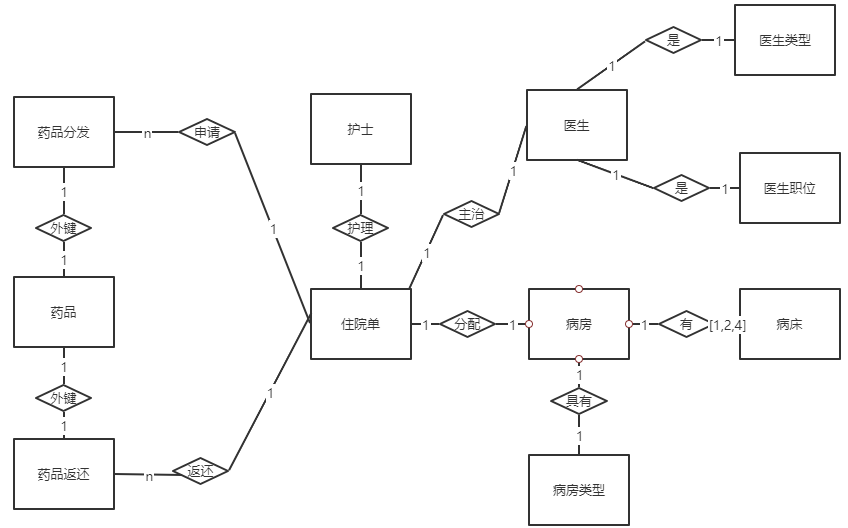


图3.9 ER-1图

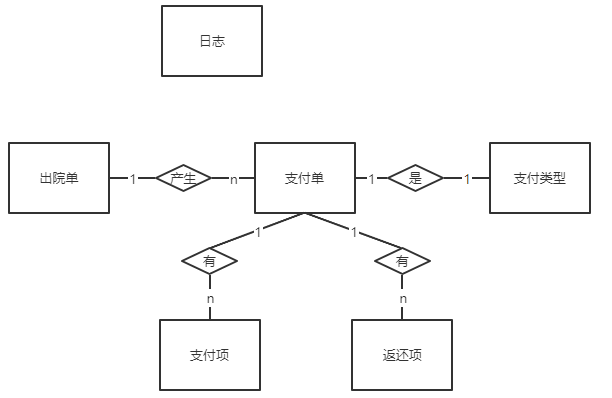


图3.10 ER-2图

实体关系核心围绕两个方面进行展开，如图3.9和3.10所示。图3.9主要根据住院单作为中心进行联系，住院单依赖于医生和护士以及病房实体；在住院期间护士可能会根据医生的医嘱为该病人开药，这就需要药品实体以及相应的分发表和返还表；病房实体又根据病床相联系，于是便有了图3.9的关系。图3.10主要围绕支付单展开，支付单通过出院单进行结算，出院时会生成该病人的支付单，支付单与支付项和返还项联系；支付单有三种状态，包括未缴费、预缴费、已经缴费，在入院的时候，需要进行预缴费，另外在住院期间余额不足时，也需要再次进行预缴费，最后出院时进行最后的阶段，补缴费或者返还相应的费用。

### 3.4.2数据库表设计

1. 字典表设计

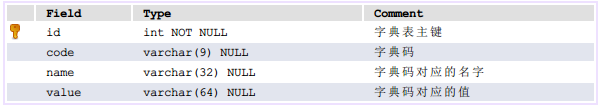


图3.11 字典表图

表设计为约束行为，比如民族的选项是一定的或者日志的级别也是一定的，id为主键，code为字典码，name为名字，比如民族或日志，value是具体内容比如汉族或者INFO级别。

1. 医生表设计

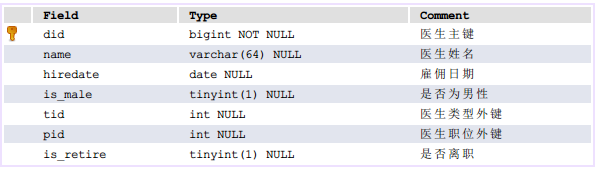


图3.12 医生表图

did为主键，name为医生姓名，hiredate为雇佣日期，is\_male有两种值：1为男，0为女，tid为医生类型外键，比如呼吸科等，pid为职位外键，比如住院医师等。

1. 医生职位表设计

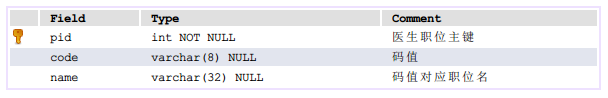


图3.13 职位表设计图

pid为主键，code为01串的码值，比如001某个职位，name是其码值对应的职位名。

1. 医生类型表设计

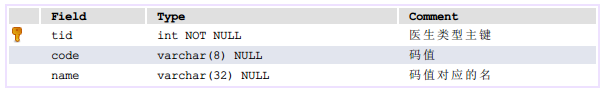


图3.14 医生所属科室表

tid为主键，和图3.12所示类似，但是业务不一样，name主要指医生所属的科室。

1. 住院单设计

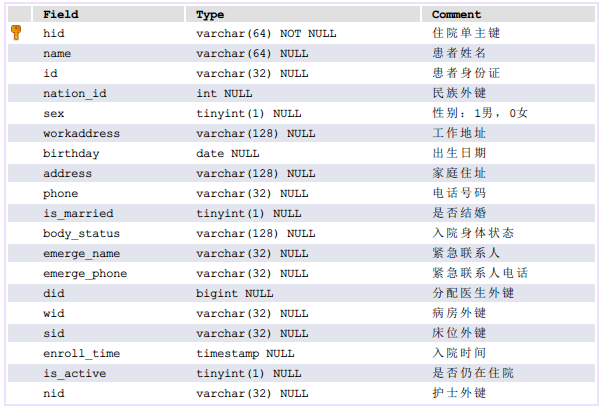


图3.15 住院单表

hid为主键列，通过雪花算法生成，name为姓名，id为身份证列，nation\_id为外键列，表示该病人的民族，sex为性别列：1为男，0位女，workaddress表示工作地址，没有可不填，birthday为出生日期 ，address为家庭住址可不填，phone为联系电话必填，is\_married是否结婚：1表示已婚，0表示未婚，body\_status表示身体状态，emerge\_name表示紧急联系人，emerge\_phone紧急联系人电话可填亲属。did为外键列，表示医生，wid为外键，表示病房，sid为床位外键，enroll\_time为入院时间，is\_active表示是否仍在住院状态：1表示住院，0表示已出院，nid为护士外键，这里指的是用户表主键。

1. 日志表设计

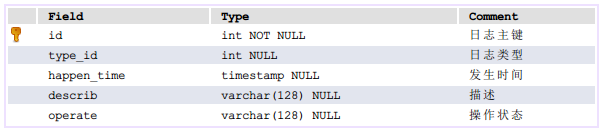


图3.16 日志表图

id为主键，数据库自增，type\_id为外键，指向日志类型表，happen\_time表示事件发生时间，describ表示事件内容，operate表示操作状态。

1. 医嘱表设计

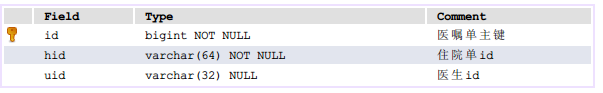


图3.17 医嘱表图

id为主键列，数据库自增，hid为外键，指向住院单，uid为外键，指向用户表，表示医生账户。

1. 医嘱项设计

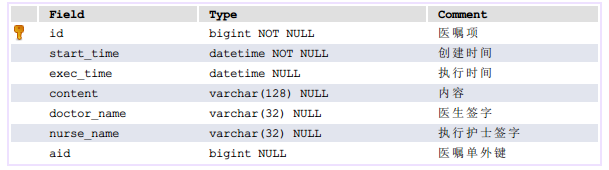


图3.18 医嘱项设计图

id为医嘱表主键，数据库自增，start\_time表示该医嘱项创建时间，exec\_time表示执行时间，content表示医嘱项内容，doctor\_name表示医生姓名，医生发布医嘱项时需要进行签名确认，nurse\_name表示护士姓名，同样在执行时需要进行签名确认，sid表示引用的医嘱单外键。

1. 药品表设计

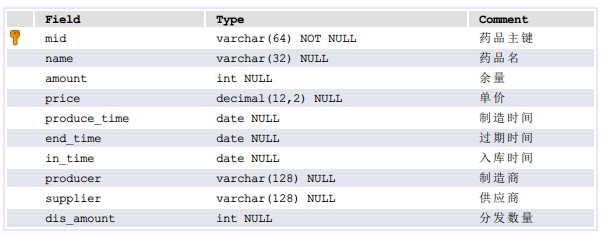


图3.19 药品表设计图

mid表示药品表主键，同样也是药品唯一标识，name药品名，amount表示药品的入库总数量不会下降，price表示药品的单价，produce\_time表示制造时间，end\_time指药品过期时间，in\_time为入库时间，producer为制造商，supplier指供应商，dis\_amount表示分发数量。

1. 药品分发表设计

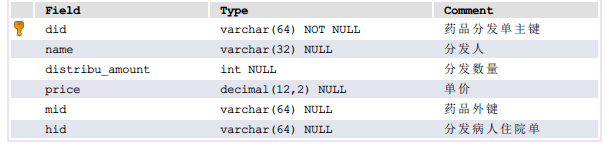


图3.20 药品分发表设计图

did表示药品分发单的主键，由雪花算法生成，name表示申请人，一般指病人，distribu\_amount表示分发数量，price表示项目单价，mid表示药品的外键，hid指病人的住院号。

1. 药品返还表设计

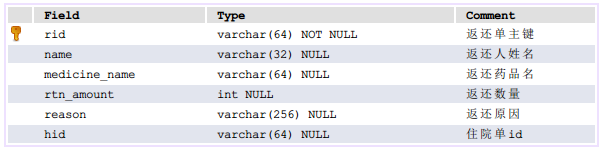


图3.21 药品返还表设计图

rid表示药品返还表的主键，name指的是病人姓名，medicine\_name为药品名，rtn\_amount为返还数量，reason表示返还原因，hid为住院单主键值。

1. 菜单表设计

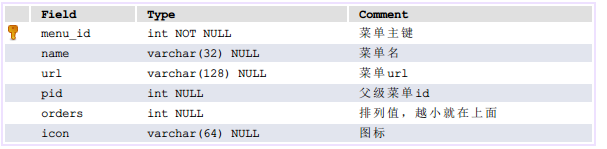


图3.22 菜单表设计图

menu\_id为资源表主键，主要用于权限管理，name为菜单的显示名，url为资源的路径，pid为父菜单的主键，orders为菜单的排序值，越小越大，icon为菜单显示时的图标。

1. 角色表设计

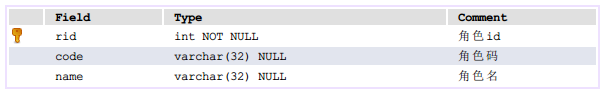


图3.23 角色表设计图

rid为主键，code为01串码值，name表示码值对应的角色名，值的是ROLE\_USER等。

1. 角色菜单中间表设计

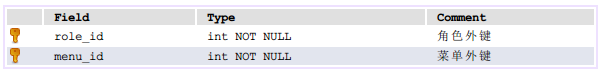


图3.24 角色菜单中间表设计图

该表为角色表和资源表的中间表，表示两表之间的多对多关系。

1. 支付单设计

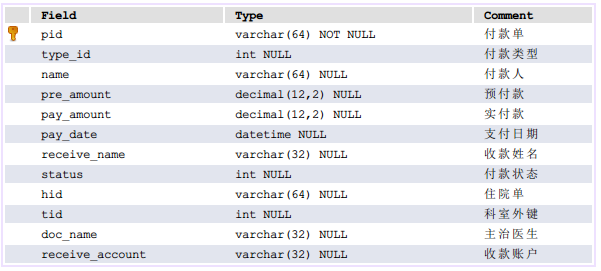


图3.25 支付单设计图

支付单表，pid为主键，由雪花算法生成，唯一，type\_id为支付类型外键，name值付款人，pre\_amount为预付款指预缴的费用，pay\_amount为结算时的费用，pay\_date表示支付日期，receive\_name指收款人姓名，可以不是该系统的用户，status为支付状态：0为未支付，1为已预缴费，9表示支付成功。hid指住院单，tid指的是入院的科室外键，doc\_name指主治医生，receive\_account指的是系统用户。

1. 支付项设计

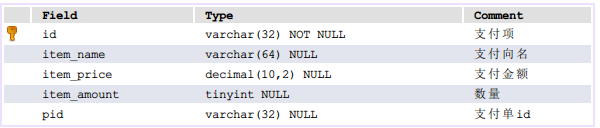


图3.26 支付项设计图

id为支付项的主键，雪花算法生成，唯一，item\_name指支付的项目，item\_price指价格一般为单价，item\_amount支付数量，pid为外键，引用支付单。

1. 返还项设计

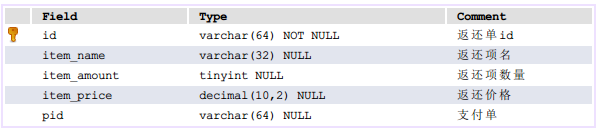


图3.27 返还项设计图

该表主要用于支付表使用。

1. 支付类型设计

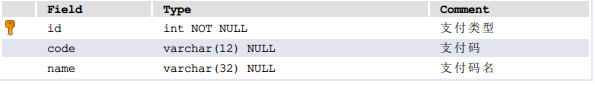


图3.28 支付类型设计图

id为支付类型主键，code为01串表示码值，name为支付名。

1. 病床表设计

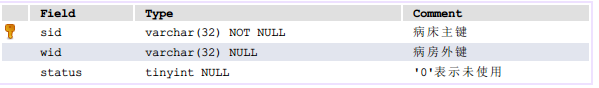


图3.29 病床表设计图

sid表示病床主键，由系统生成，唯一。wid为病房外键，status为病床使用状态：1为正在使用，0表示空。

1. 出院单设计

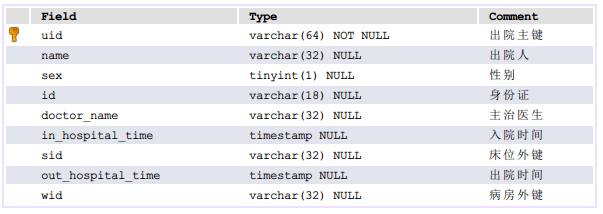


图3.30 出院单设计图

登记出院信息，uid为主键，雪花算法生成，唯一，name为出院人姓名，sex为性别：1为男，0位女，id为身份证号，doctor\_name为主治医生姓名，in\_hospital\_time为入院时间，out\_hospital\_time为出院时间，sid为住院时所填，wid为所在病房。

1. 用户表设计

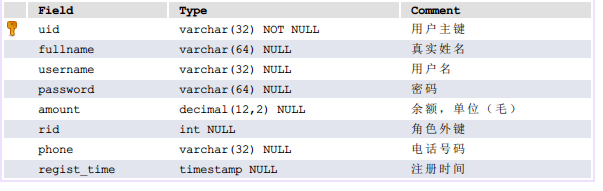


图3.31 用户表设计图

用户表，uid表示主键，由雪花算法生成，唯一，fullname为真实姓名，username为账户名，password为密码，通过BCrypt算法加密，amount账户余额，rid为角色外键，phone为电话号码，regist\_time为注册时间。

1. 病房设计

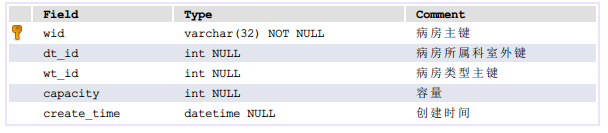


图3.32 病房设计图

wid为病房表外键，由用户输入和楼层有关比如100，101等，dt\_id表示给该病房的主要所属科室，wt\_id为类型外键，如ICU等，capacity为容量，create\_time为创建时间。

1. 病房类型表设计

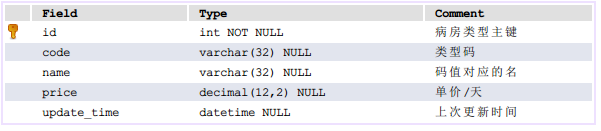


图3.33 病房类型图

id为主键，code为01串码值，name为病房类型，比如普通病房，price为病房单价，单位是元/天，update\_time为上次更新该表的时间。

## 3.5业务模块设计

### 3.5.1登录模块

登陆模块的核心内容是权限拦截，主要实现为根据登陆账号的不同角色，通过后台查询不同的菜单接口，然后交由前端显示不同的左侧功能菜单，以此来实现权限拦截功能，包括管理员、护士、医生、前台等角色。

（1）执行单元构成

登录模块的类图如下图3.34所示；

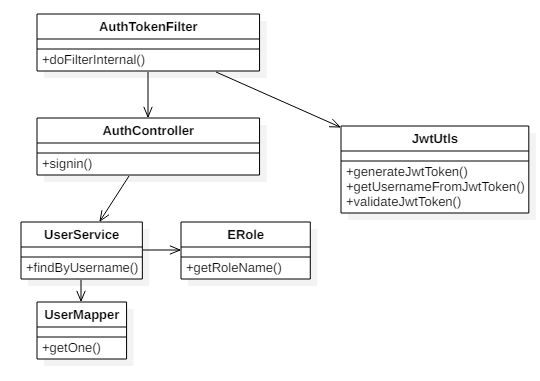


图3.34 登录模块类图

在控制器层之前，请求会被AuthTokenFilter拦截，主要内容是校验请求是否携带了本系统应有的认证头Authorization。

控制器层，主要是将登陆请求的信息封装为一个认证信息，然后交由Spring Security去认证，认证成功会返回有效的认证信息。

业务逻辑层，核心主要是根据用户名查询真实存在的用户其他信息，尤其是加密后的密码和角色信息。

数据访问层，从数据库获取用户信息，由业务逻辑层调用。

（2）执行流程设计

在控制器层之前，会对请求进行拦截，校验请求头字段是否存在自定义的token头，如果该请求头不存在该请求会被截断，并不会继续校验用户名密码。

到了控制器层，AuthController根据传入的用户名、密码创建一个认证和数据库中的字段进行比对，这时候会通过业务逻辑层的UserService，通过其再次调用UserMapper拿出对应的真实有效的信息进行校验。校验成功，向Spring Security管理器中设置一个认证信息，AuthController拿出该认证信息，取出该角色对应的非私密的权限信息返回给前端；若校验失败，会截断该请求，并不会调用AuthController接下来的代码，直接向前端返回认证失败的信息即可。整个访问过程如图3.35所示；

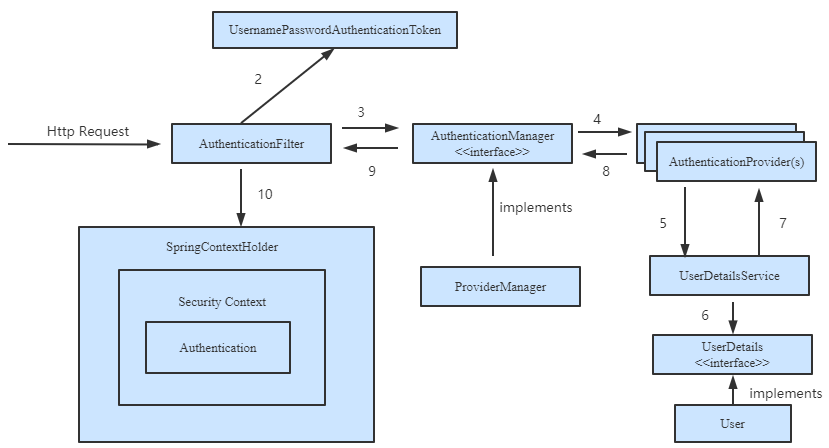


图3.35 登陆流程图

### 3.5.2 入院登记模块

入院登记，主要内容就是前台护士帮助病人选择主治医生和相关护理护士和病床信息，留待进一步的医疗诊治。

（1）执行单元构成

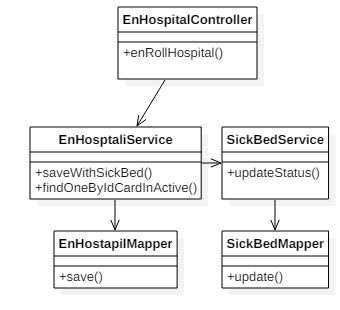


图3.36 入院登记模块类图

控制器层，EnHospitalController主要是接受前端传入的参数，并对其有效性进行校验，然后封装为一个实体类对象，将其传递给业务层。

业务逻辑层，核心是保存病人的入院单，和相关的信息，修改入院单中指定的病床状态。在本功能中主要由两个业务类构成，EnHospitalService用来对住院单信息进行修改，SickBedService主要是更新登记时传入的病床的状态，将其置位已使用。数据访问层，真正和MYSQL数据库进行信息交换的层，通过EnHospitalMapper和SockBedMapper进行调用。

（2）执行流程设计

流程上，如图3.37所示;

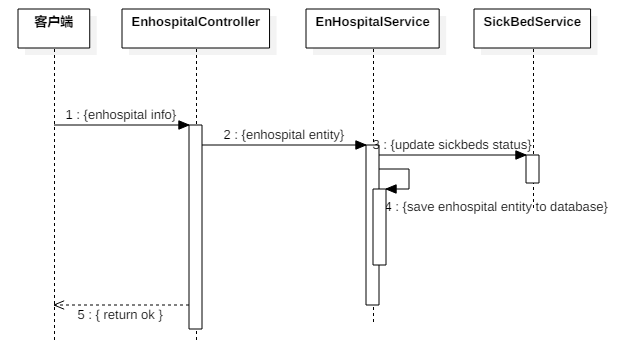


图3.37 入院登记时序图

前台护士根据病人或病人家属说明的信息，然后根据门诊部的诊断说明，在Vue写的浏览器客户端选择为该病人分配主治医生、护士、病床等，选择后发送提交会作为请求数据传递给后台EnHospitalController，在方法中会将有效信息摘取出来。首先会对该病人的部分重要信息进行校验，若校验不通过直接返回失败，否则作为业务属性传递给EnHospitalService，由它去调用SickBedService修改对应座位的状态，然后自身调用save方法将其入院信息录入到系统，为前台返回业务成功的状态码。

### 3.5.3 立行医嘱模块

（1）执行单元构成

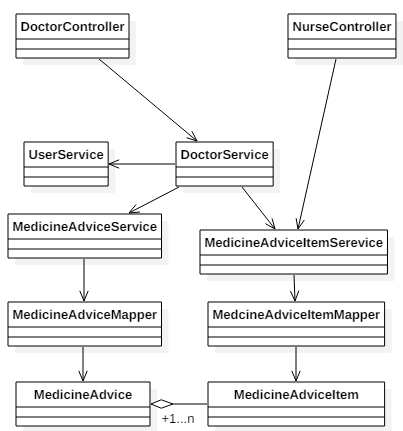


图3.38 医嘱模块类图

从上到下分别为控制层、业务层、数据访问层和实体类对象。

医嘱模块主要牵涉两个实体类，分别是医嘱单（MedicineAdvice）和医嘱项（MedicineAdviceItem），他们彼此之间是1对多的组合关系。

控制层DoctorController将请求数据进行解构，将其转交给业务层DoctorService，由他再进行业务层之间的调用，去校验、序列化等；NurseController比较简单，直接将数据传递给业务层，主要是对医生发送的医嘱进行核对和予以执行。

（2）执行流程设计

从医生的角度看，主要负责编写医嘱，其流程如下图3.39所示；

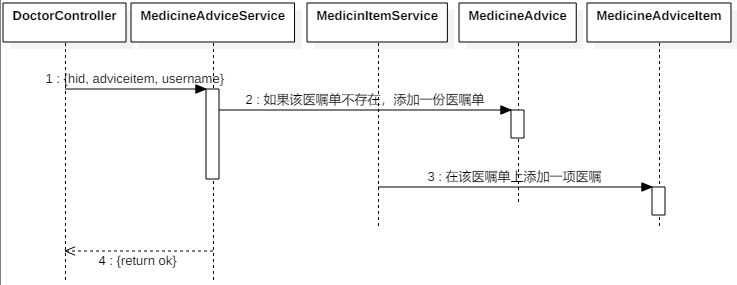


图3.39 医嘱执行顺序图

医生在接受到病人后，根据病人的情况开出适当的医嘱信息，将其提交到系统中，由DoctorController接受，之后取出关键信息，先由住院号查询该医嘱单是否存在，如果不存在则创建，之后根据新增的医嘱单作为医嘱项的外键，将其序列化到数据库中。

从护士的角度看，主要内容就是查看所负责病人的医嘱单、医嘱项，进行签字确认并执行，将该确认信息存储到硬盘上，如图3.40所示；

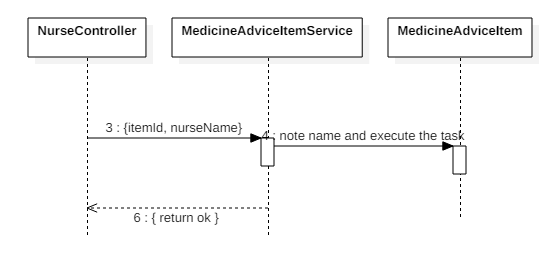


图3.40 护士查看并执行医嘱时序图

# 第四章 系统开发

## 4.1开发环境

本系统采用的开发环境如表4.1所示；

表4.1 开发环境配置表

|  |  |
| --- | --- |
| 开发语言 | Java、JavaScript |
| 开发工具 | IntelliJ IDEA、Vscode |
| 开发框架 | 前端框架：  Vue、Vuex、Axios、Quasar |
| 后端框架：  Spring Boot、Spring、Spring Mvc、Mybatis、MybatisPlus |
| 存储方案 | 1. MYSQL作为主要存储 2. Redis作为中间件缓存。 |
| 配置 | 1. 需要Redis、MYSQL数据库 2. 项目由Maven构建而成，因此依赖于Maven 3. 主配置文件在resouces下，主要通过application.yml进行版本控制，application-dev.yml是开发环境,application-prod.yml是生产环境 |

## 4.2关键技术

### 4.2.1 Spring Boot

由于J2EE 笨重的开发、繁多的配置、低下的开发效率、复杂的部署流程、第三方技术集成难度大，都成为了阻碍开发者效率的难关。Spring Boot 的任务就是用来简化Spring应用开发，约定大于配置，去繁从简，Just run 就能创建一个独立的，产品级别的应用。Spring Boot 通过整合整个 spring 系列的技术栈，从而简化了企业级开发。Spring Boot 帮助我们自动配置好spring的相关环境，用户只需从一个微小的入口就能打开整个企业级开发的大门。

从内容上来说，Spring Boot仍然应用了Spring、Spring MVC、Mybatis（SSM）这样的三层框架；但其优越性在于，首先，由于其内置的Servlet容器，因此无需通过Maven打包成war包，再部署到外部的Tomcat容器上；二则是通过大量的自动配置类，使用模板设计模式，提供了大量随时可注入的模板对象，简化开发，摒弃了原有的大量堆砌XML文件的方式；三则是依赖版本控制，通过starters的自动依赖，简化了进行版本更迭时，更细化的检查maven依赖的每一个配置项问题；最优越的一点则是其与微服务的无缝衔接，只需引入依赖，开箱即用。

从结构上描述的话，如下图4.1所示；

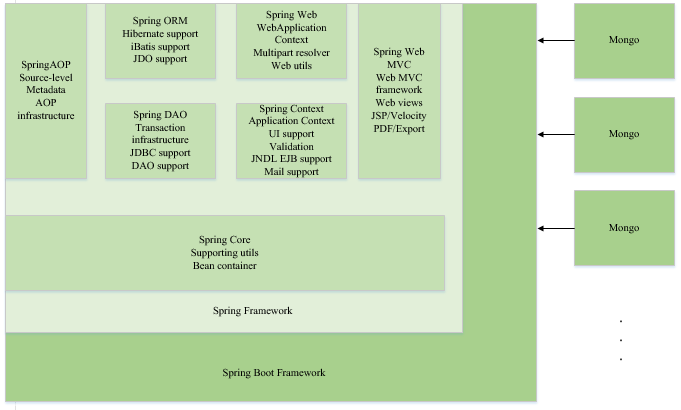


图4.1 Spring Boot结构图

其通过大量的自定义Java注解的方式，将配置、生命周期、组件化等等的思想纳入其中，但其内部流程仍是凭借以往SSM的流程进行业务实现，因此开发人员可以很快的从原本非Spring Boot的技术栈迁移过来。

### 4.2.2 Spring Security、JWT

1.JWT（JSON WEB TOKEN）技术

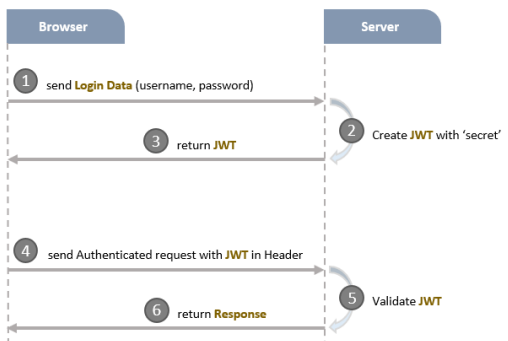


图4.2 含有JWT的登陆数据流图

JWT是用于认证和信息交换的，用于替代以前的基于Session的认证。客户端携带用户名密码登录，服务器认证成功后，生成一个JSON Web Token，发送给客户端，客户端之后的请求都会携带这个Token在请求头，服务端会校验若校验成功，才会进行接口调用，并返回相应；否则，请求会被切断。

JWT一般由三部分构成，header、payload、signature；header作为头部信息，主要承载内容所使用的类型和加密算法；payload是存放有效信息的部分；signature是一些用算法生成的能够认证身份的字符串，一般是主要是把头部的base64UrlEncode与负载的base64UrlEncode拼接起来，再进行HMACSHA256加密，加密结果再进行base64url加密，最终得到的结果作为签名部分。JWT的大致结构如下所示；

header.payload.signature

2.Spring Security

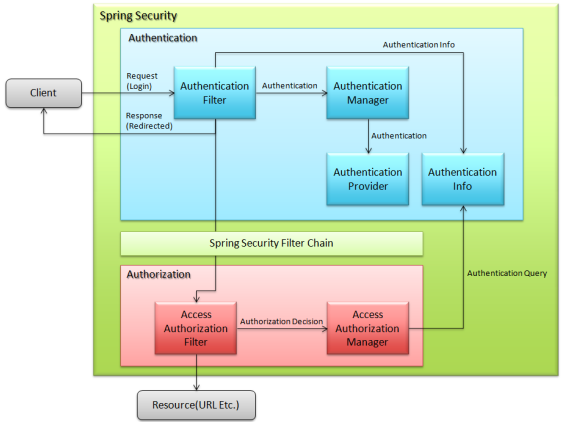


图4.3 处理流程图

Spring Security通过一系列过滤器实现，由“身份验证”和“授权”组成，负责应用程序的安全性。它作为主要功能提供。认证功能是用于识别用户以便防止通过欺骗的未授权访问的功能。授权功能取决于经过身份验证（登录）的用户的权限。此功能控制对系统资源的访问。

### 4.2.3 Vue

Vue被设计成从下到上逐层应用。Vue的核心库只关注可视层，它不仅易于使用，而且易于与第三方库或现有项目集成。另一方面，当与现代工具链和各种支持类库相结合时，Vue还可以为复杂的单页应用程序提供驱动程序。其主要特点如下[12]。

1. 轻量级框架，提供了双向的数据绑定，Vue.js可以自动跟踪依赖模板表达式和计算属性，提供MVVM数据绑定和可组合组件系统，API简单灵活，便于读者理解和快速入门[12]。
2. 组件化，组件(Component)是Vue.js最主要的特性之一。组件可以进一步的扩展 HTML 元素,封装可重用的代码。在 Vue 中,父子组件通过 props 传递通信,从父向子单向传递。子组件与父组件通信,通过emit触发事件通知父组件改变数据。这样就形成了一个基本的父子通信模式。在开发中组件和HTML、JavaScript等有非常紧密的关系时,可以根据实际的需要自定义组件,使开发变得更加便利,可大量减少代码编写量。组件还支持热重载(hot reload)[12]。
3. 客户端路由，Vue router是Vue.js官方路由插件,与Vue.js深度集成,以构建单页面应用。Vue单页面应用是基于路由和组件的,路由用于设定访问路径,并将路径和组件映射起来,传统的路由页面跳转是通过超链接实现页面的切换和跳转的[12]。

## 4.3开发成果

### 4.3.1登陆模块

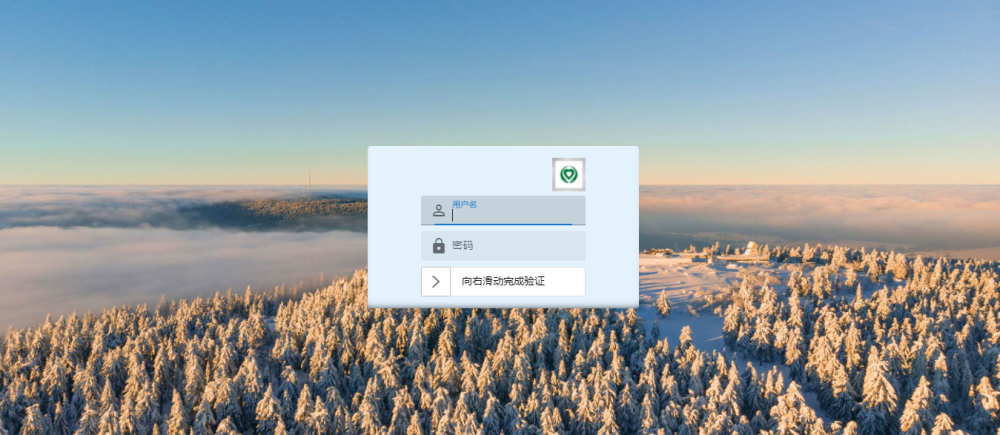


图4.4 登陆界面图

用户输入用户名、密码进行登录后，右划下侧的滑块进行登录，登陆成功跳转到首页；登陆失败时，会显示登陆失败的信息在屏幕上方。

### 4.3.2入院登记



图4.5 入院登记个人信息图

前台护士帮助录入病人的基本个人信息，然后点击下一步。



图4.6 入院登记住院信息图

前台护士根据病人的门诊的相关信息，选择对应的主治医师、护理护士、病房，并点击登记，登记成功会有消息提示。

### 4.3.3病房管理



图4.7 新增病房图

管理员新增病房时，输入这些基本信息，会向系统中添加一个病房，在病人入院登记时会显示出来。



图4.8 病房价位调整图

管理员点击调整后的价格，会弹出一个输入新房价的输入框，输入并点击确认即可实现病房价格的设置。

### 4.3.4结算中心



图4.9 费用预缴图

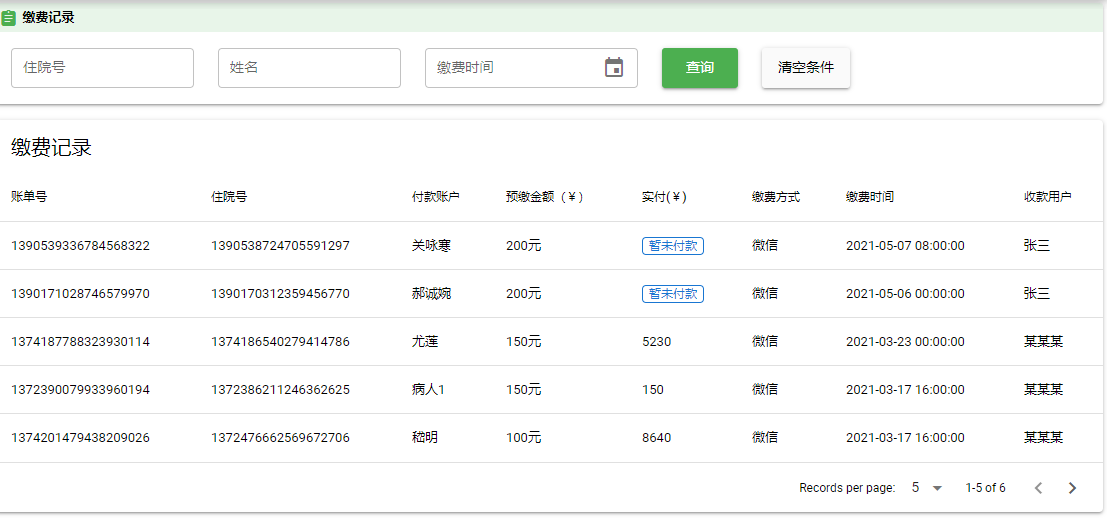
用户在前台进行缴费，前台输入住院单号，住院人等必要信息，进行记录。

图4.10 缴费记录查询图



图4.11 结算页面

病人在进行出院登记后，即可进行结算，输入其住院号，点击查询，会出现相关的费用项和返还项；点击结算按钮，会自动计算出需要补交或者要返还的余额，由护士点击按钮进行确认支付。

### 4.3.5统计中心



图4.12 病人统计表格图

该模块是管理员的功能，管理员输入开始日期、结束日期，点击查询会统计出在该时间段内的各科室的入院、出院人数统计情况，主要表现为表格形式如图4.12或条形统计图如4.13。

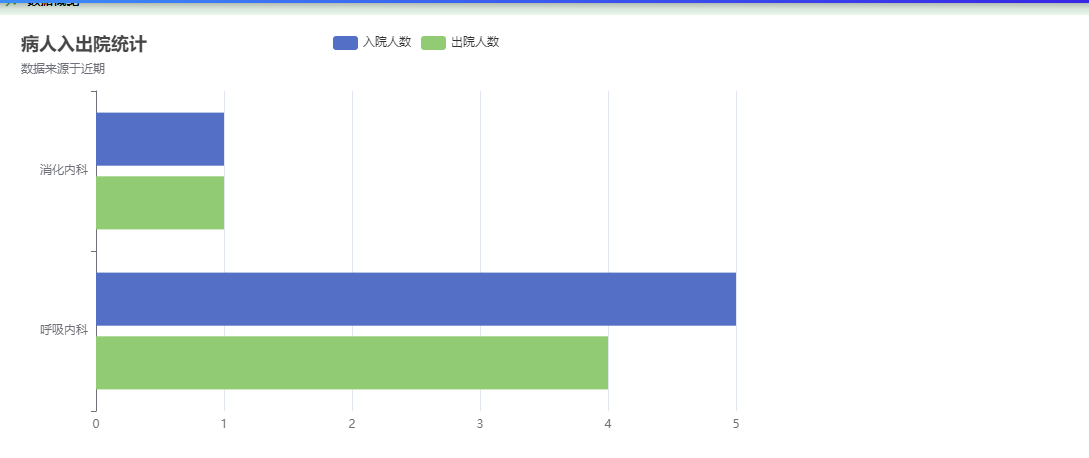


图4.13 病人统计条形图

图4.14以及如4.15也是管理员角色的功能模块。根据选择病房类型去统计该类型病房当前的使用情况；若不进行选择，则默认是统计各个病房类型的。



图4.14 病房统计表格图

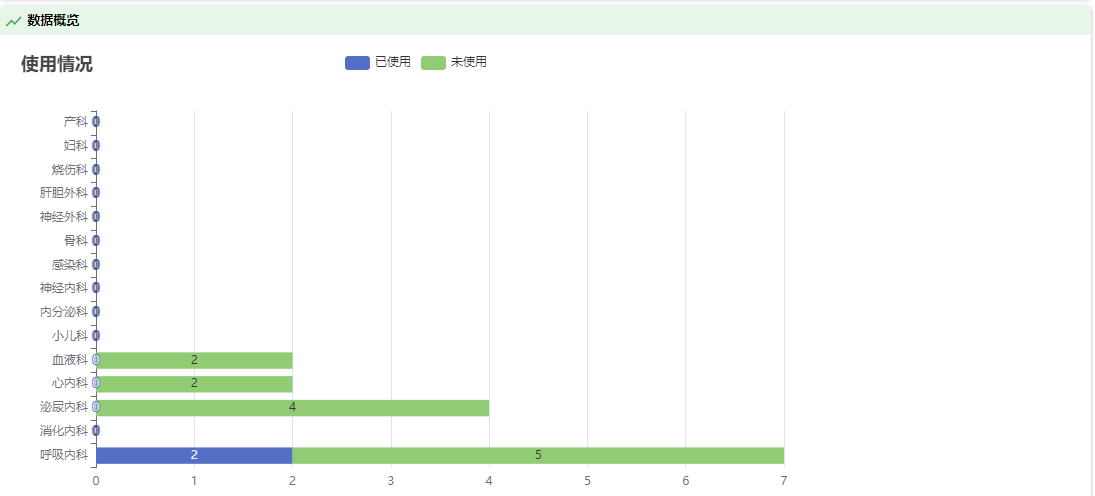


图4.15 病房统计条形图

### 4.3.6药品管理



图4.16 药品入库

管理员角色输入药品的相关信息，选择保存，将药品入库到系统中，供给住院病人发放。



图4.17 入库记录查询

管理员角色登入该系统，输入药品编号或者名称或者入库时间即可查询药品的相关信息，包括药品名称、单价、药品数量等。



图4.18 库存查询

该功能也是提供的查询界面，与图4.17的不同在于该模块会显示库存数量即该药品是否需要进货，而并不是入库时的数量。

### 4.3.7医嘱模块

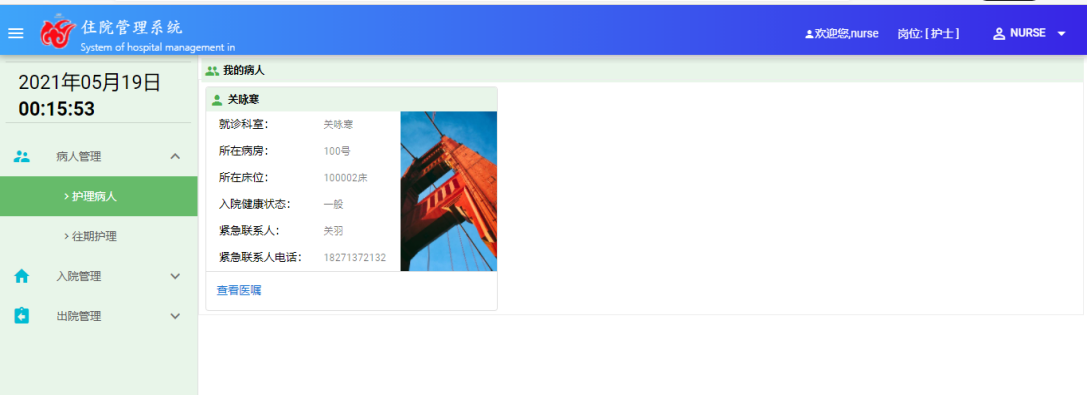


图4.19 医嘱界面图

医生或者护士登入该系统后，点击护理病人，即可看到自己所负责的病人的基本信息，点击查看医嘱就会进入图4.20所示的界面，医生可以在该界面撰写对该病人如何进行治疗的相关举措，由护理的护士前去施行。



图4.20 写医嘱界面图



图4.21 护士查验图

医生对该病人撰写医嘱后，护士相应的医嘱界面会显示待执行的项目，护士可以查验该项，如果不符合可以和医生商量，护士需要对此项目执行，完毕后需要进行签字确认。

### 4.3.8出院登记



图4.22 出院登记图

前台护士帮助病人输入其相关的住院信息，登记办理出院手续，在登记过后，才可进行最终的结算出演，为了方便检索提供了如图4.23的查询功能。

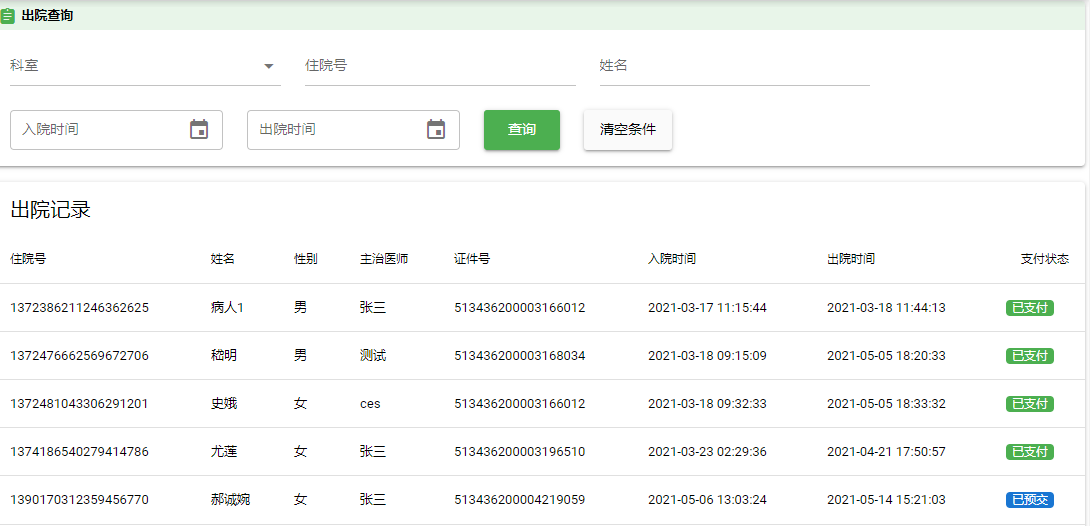


图4.23 出院查询图

### 4.3.9系统管理

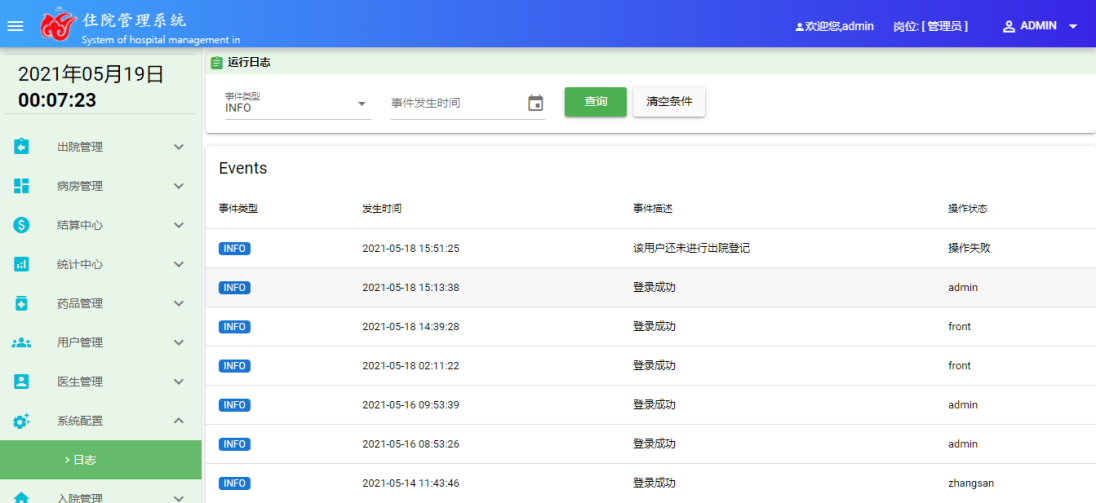


图4.24 日志查询图

该模块主要起监控行为，主要记录系统的核心错误和一些登陆请求信息以及审批药品等的记录信息。

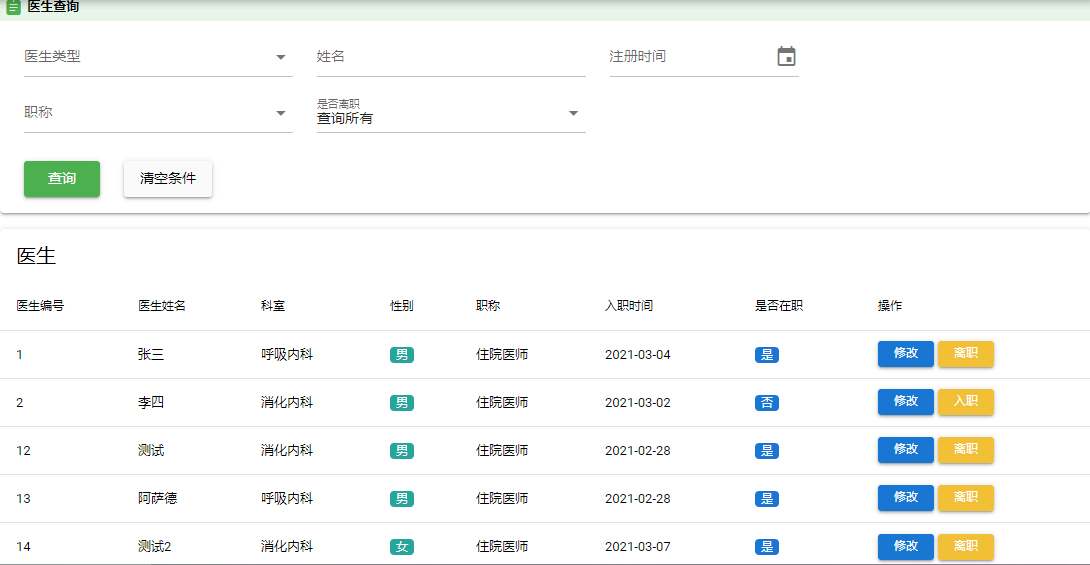


图4.25 医生信息检索图

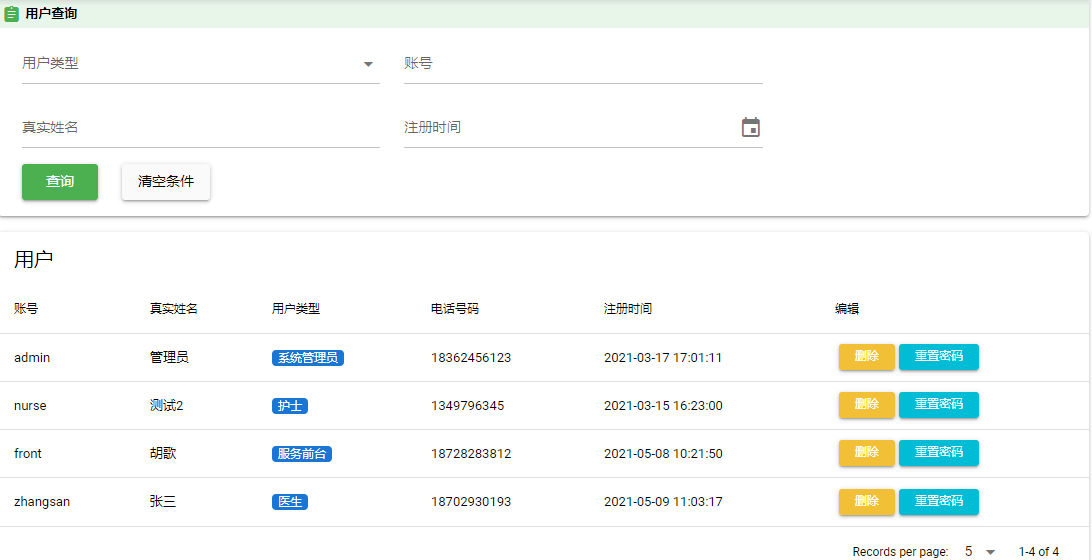


图4.26 用户检索图

该系统提供了注册功能，都必须为管理员角色，包括用户注册和医生信息录入，并提供了查询页面，如图4.25,4.26所示；管理员可以对用户进行密码重置或信息修改，对于医生来说额外提供了提供了离职功能。

# 第五章 系统测试

## 5.1测试设计

为测试所设计的系统是否满足要求，对系统进行了测试。测试方法主要采用黑盒测试方法。通过使用两种不同的系统，从登录模块开始，对系统的各个功能模块逐一测试，主要包括登陆模块、结算模块、入库模块、入院模块、医嘱模块、医生模块、用户模块。

## 5.2测试用例及测试记录

### 5.2.1登录模块

表5.1 登录模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入条件** | **预期输出** | **确认输出** | **备注** |
| 1 | 用户名：admin  密码：amdin | 正常登入系统，菜单显示正常 | 成功 | 管理员账户 |
| 2 | 用户名：123(不存在)  密码：xxx | 登入失败，提示信息，清空 | 成功 | 无 |
| 3 | 用户名: front  密码：front | 正常登入，菜单显示正常 | 成功 | 前台账户 |
| 4 | 用户名：zhangsan  密码：111(错误) | 登入失败，提示信息，清空 | 成功 | 无 |

从测试结果可以看出，登陆模块测试良好，能够完成基本的登陆需求以及不同用户的之间权限拦截的需求，能支持系统运行。

### 5.2.2结算中心

表5.2 结算中心测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入条件** | **预期输出** | **确认输出** | **备注** |
| 1 | 住院号：-1 | 提示住院号不存在消息通知 | 成功 | 无 |
| 2 | 住院号：13721378391（存在）  未进行出院登记 | 提示住院号未登记出院手续消息 | 成功 | 无 |
| 3 | 缴费查询  住院id：-1 | 未查出记录 | 成功 | 无 |

根据表5.2的测试结果，结算中心的结算功能反应良好，能判断出住院号不存在和存在但未出院的情况，缴费查询能够判断错误的输入情况，运行情况良好。

### 5.2.3入库模块

表5.3 医嘱测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入条件** | **预期输出** | **确认输出** | **备注** |
| 1 | 单价：-1 | 单价不能为负数 | 成功 | 无 |
| 2 | 单价：1 | 入库成功 | 成功 | 无 |
| 3 | 在已经入库的药品，再次入库 | 将数量添加到原有的药品数额上 | 成功 | 无 |

入库模块的测试根据需求上来说，应满足基本的输入校验，并且能够在已经入库的药品时，能动态的将数量添加上去，而不是新建一条记录。从测试结果来看，反应良好，可以投入使用。

### 5.2.4入院模块

表5.4 入院模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入条件** | **预期输出** | **确认输出** | **备注** |
| 1 | 身份证号: 123456 | 提示输入合理的长度 | 成功 | 无 |
| 2 | 身份证号：610115200002038814 | 入院成功 | 成功 | 无 |
| 3 | 科室：null | 提示入院失败消息 | 失败 | 无 |
| 4 | 医生：null | 提示入院失败消息 | 失败 | 无 |

根据表5.4的测试样例，在进行输入时，只对id进行了比较完善的校验，在对其他字段的校验上并不是很充裕，需要用户在输入时注意字段的填写是否完备。

### 5.2.5医嘱模块

表5.5 医嘱模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入条件** | **预期输出** | **确认输出** | **备注** |
| 1 | 医师签名：jxx | 提示该医师不存在 | 失败 | 无 |
| 2 | 医嘱内容：mnz | 提示内容不合理 | 失败 | 无 |
| 3 | 医师签名：张三 | 添加成功 | 成功 | 无 |

根据测试结果，从总体上，可以通过测试，但是在医生输入的时候没有限制，这就需要医生个人去进行输入约束。

### 5.2.6医生模块

表5.6 医生模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入条件** | **预期输出** | **确认输出** | **备注** |
| 1 | 姓名：xxx | 添加成功 | 成功 | 录入 |
| 2 | 科室：null | 提示错误消息 | 成功 | 录入 |
| 3 | 职位：null | 提示错误消息 | 成功 | 录入 |
| 4 | 医生类型：呼吸内科 | 检索成功 | 成功 | 检索 |

由表5.6的4条测试记录来看，录入子模块的校验还是比较完善，能够通过几乎所有的输入校验；其次对于检索子模块，由于检索条件没有限制，因此该模块更易于通过测试，医生模块测试结果良好。

### 5.2.7用户模块

表5.7 用户模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入条件** | **预期输出** | **确认输出** | **备注** |
| 1 | 姓名：三 | 张三 | 成功 | 检索 |
| 2 | 姓名：三 | 添加成功 | 成功 | 录入 |
| 3 | 权限设置：前台 | 添加成功 | 成功 | 录入 |

用户模块分为录入和检索两个功能，录入主要是为使用该系统的用户，提供一个访问账号，另外将每个用户的权限区分开，录入功能良好；检索可以通过姓名、账号、注册时间等进行搜索，可以满足需求。

## 5.3测试结果及结论

根据测试用例的执行结果，在一般情况下，系统能够完成大部分的测试结果，需要对部分输入进行人为控制，但不影响整体流程，可以交付使用。

# 第六章 展望与总结

## 6.1本文工作总结

该系统采用了B/S基于MVC三层架构的模式，使用了当前比较受欢迎的Spring Boot和MyBatis-Plus框架搭建Web项目，使用了MySQL数据库来存储数据，以此实现了一个医院住院管理系统，

系统中已经实现的功能包括用户的登录注册，医生管理、病房管理、入院登记、出院登记，统计中心，检索病历信息，医嘱模块，药品的申请发放以及最后的出院结算功能。

在已经完成的功能中，尚且还有很多不足之处，比如在病人在入院后，医生和护士对其进行护理的任务是不同的。类似较多细节问题还需不断改进，使其更加的贴近实际生活并能应用到医院住院系统中。

## 6.2未来工作展望

为了增加用户的美好体验，使用户能更加实惠又方便的使用此系统，该系统应有一个扫码的功能，用户进行扫码实名后可以进行住院查询历史病历，支付，实现扫码就医。

# 结束语

针对住院管理系统的需求，设计了一种基于Spring Boot的住院管理系统。系统使用Spring Boot、Vue框架开发，具有结构清晰、便于扩展和维护等特点。整个系统包含登陆模块、入院模块、出院模块、药品管理模块、注册模块、统计模块、支付模块、医嘱模块、系统管理模块、病房管理模块共十个模块，满足住院系统的基本需求。详细介绍了系统的总体设计、数据库设计和关键技术，并对系统进行了基本测试。测试结果说明该系统能够完成基本的住院需求。

# 致 谢

感谢培养教育我的母校，浓厚的学术氛围，和谐的人文氛围，在论文的写作上给予了我很大的帮助。

感谢我的导师拓守恒副教授，他严谨细致、一丝不苟的作风一直是我工作、学习中的榜样；在我的开题报告和最后毕业论文编写的时候，给予了很大的帮助，他循循善诱的教导和认真负责的态度让我很是敬重。

感谢我的117室友们，从遥远的家来到这个陌生的城市里，是你们和我共同维系着彼此之间兄弟般的感情，维系着寝室那份家的融洽。四年了，仿佛就在昨天。四年里，我们没有红过脸，没有吵过嘴，没有发生上大学前所担心的任何不开心的事情。只是今后大家就难得再聚在一起吃每年元旦那顿饭了吧，没关系，各奔前程，大家珍重。

4

# 参考文献

[1]王家芝. 成都大学校医院管理系统的设计与实现[D].电子科技大学,2013.

[2]刘丽波. 医院管理理论与系统研究[D].天津大学,2010.

[3]舒婷,刘海一,赵韡.中美电子病历系统十年发展启示[J].中国数字医学,2019,14(11):10-13.

[4]田洋. Web应用系统的通用开发架构研究与应用[D].山东农业大学,2016.

[5]李嘉,赵凯强,李长云.Web前端开发技术的演化与MVVM设计模式研究[J].电脑知识与技术,2018,14(02):221-222+251.

[6]蔡泽铭,王文华.基于Vue.js的信息管理系统前端架构[J].电子技术与软件工程,2020(18):142-144.

[7]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.

[8]曾超宇,李金香.Redis在高速缓存系统中的应用[J].微型机与应用,2013,32(12):11-13.

[9]Omar S. Gómez,Raúl H. Rosero,Karen Cortés-Verdín. CRUDyLeaf: A DSL for Generating Spring Boot REST APIs from Entity CRUD Operations[J]. Cybernetics and Information Technologies,2020,20(3):3-14..

[10]Yao Zhang Li,Sheng Gao,Jing Pan,Bi Feng Guo,Pei Feng Xie. Research and Application of Template Engine for Web Back-end Based on MyBatis-Plus[J]. Procedia Computer Science,2020,166:1-7.

[11]Sudha Rajesh,A. Chandrasekar. Esteemed software patterns for banking system[J]. Cluster Computing,2019,22(5).

[12]陈晓,李杰,邱祺.基于Vue.js框架的智慧交通程序方案[J].信息技术与信息化,2020(11):240-243.