



廣州城市理工學院

Guangzhou City University of Technology

(原华南理工大学广州学院)

本科毕业设计（论文）

基于 Spring Boot + Vue 的口腔医院管理系统的设计 与实现

学 院 计算机工程学院

专业班级 计算机科学与技术 4 班

学生姓名 温添淀

学生学号 201910089218

指导教师 吕绪洋 讲师

提交日期 2023 年 4 月 18 日

广州城市理工学院

(原华南理工大学广州学院)

学位论文原创性声明

本人郑重声明：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。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：温瑶淀 日期：2023 年 4 月 12 日

学位论文版权使用授权书

本人完全了解广州城市理工学院（原华南理工大学广州学院）关于收集、保存、使用学位论文的规定，即：按照有关要求提交学位论文的印刷本和电子版本；广州城市理工学院（原华南理工大学广州学院）图书馆有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务；可以采用复印、数字化或其它复制手段保存论文；在不以赢利为目的的前提下，可以公布论文的部分或全部内容。

学位论文作者签名：温瑶淀 日期：2023 年 4 月 12 日

指导教师签名： 日期：2023 年 4 月 17 日

作者联系电话：18219135363 电子邮箱：2357996956@qq.com

摘要

随着医院信息化的发展，口腔医院管理系统的设计与实现变得越来越重要。本文着重讨论了如何设计一个高效的口腔医院管理系统。本文介绍了这个系统的整体设计和架构。系统主要由前端界面和后端数据库组成，前端界面通过 PC 端 Web 页面呈现给医院工作人员和移动端 HTML5 页面给患者。后端数据库存储着系统所需的各种信息，例如病历、医生和患者的信息等。阐述了系统所需的主要功能，移动端包括预约、挂号、查询病历等；后台管理 PC 端包括预约挂号管理、科室管理、电子病历管理、药物管理、人员管理等。这些功能的实现需要基础数据的支持，例如患者信息、医生信息、诊断信息等。为此，设计了数据模型，以便管理和存储相关数据。介绍了系统的具体实现。该系统采用了 Java 作为开发语言，使用 Vue2.0、Spring Boot 框架来实现前后端分离的系统架构，并利用 MySQL 关系型数据库来存储各种数据。系统的实现过程中，使用了多种技术手段，例如 MybatisPlus、Element-UI、Ajax 等。

经过不断地开发和优化，该口腔医院管理系统完成了基本功能的实现。该系统通过其高效、灵活和可扩展的优势，帮助医院提升了患者就医的效率、减少了工作人员工作量，实现了医院信息化管理的目标。

关键词：管理系统；后台管理；前台移动端；spring boot

Abstract

With the development of hospital informationization, the design and implementation of oral hospital management system has become increasingly important. This paper focuses on the discussion of how to design an efficient oral hospital management system. This paper introduces the overall design and architecture of the system. The system mainly consists of a front-end interface and a back-end database, where the frontend interface is presented to hospital staff through PC-side web pages and to patients through mobile-side HTML5 pages. The backend database stores various information required by the system, such as medical records, doctor and patient information, etc. The main functions required by the system are elaborated, where mobile-side functions include reservation, registration, and inquiry, while PC-side functions include appointment and registration management, department management, electronic medical record management, drug management, personnel management, etc. The implementation of these functions requires the support of basic data, such as patient information, doctor information, diagnosis information, etc. Therefore, designed a data model for managing and storing relevant data. Finally, the specific implementation of the system is introduced, where the system uses Java as the development language, and employs Vue2.0, Spring Boot framework to realize a frontend and backend separation system architecture, and uses MySQL relational database to store various data. Multiple technical means, such as Mybatis-Plus, Element-UI, Ajax, etc., were utilized in the implementation process.

After continuous development and optimization, the oral hospital management system has achieved the implementation of basic functions. With its efficient, flexible and scalable advantages, the system has helped hospitals improve the efficiency of patient treatment, reduce the workload of hospital staff, and realize the goal of hospital information management.

Key words: Manage the system; Back-office management; Front mobile; spring boot

目录

摘 要	I
Abstract.....	II
第一章 绪论	1
1.1 研究背景.....	1
1.2 国内外现状.....	1
1.3 研究目的及意义.....	2
1.4 主要研究内容.....	2
1.5 本文结构安排.....	3
第二章 相关技术介绍	4
2.1 开发环境.....	4
2.2 JAVA 语言	4
2.3 Vue	5
2.4 Spring Boot	6
2.5 MyBatis-Plus.....	7
2.6 MySQL 数据库.....	8
2.7 本章小结.....	9
第三章 系统分析	10
3.1 可行性分析.....	10
3.1.1 操作可行性分析.....	10
3.1.2 技术可行性分析.....	10
3.1.3 经济可行性分析.....	10
3.2 需求分析.....	10
3.2.1 系统需求.....	10
3.2.2 功能需求.....	11
3.3 业务流程.....	13
3.4 本章小结.....	13
第四章 系统设计	14
4.1 数据库设计.....	14

4.1.1 E-R 模型	14
4.1.2 数据库表设计.....	16
4.2 数据流设计.....	20
4.3 系统功能模块设计.....	20
4.3.1 设计思想.....	20
4.3.2 系统结构设计.....	21
4.3.3 系统设计流程.....	22
4.4 本章小结.....	23
第五章 系统实现	24
5.1 移动端.....	24
5.2.1 用户登录模块.....	24
5.2.2 用户挂号预约模块.....	25
5.2 PC 管理端.....	26
5.1.1 挂号预约模块.....	26
5.1.2 科室管理模块.....	26
5.1.3 电子病历模块.....	27
5.1.4 药物管理模块.....	28
5.1.5 缴费订单模块.....	30
5.1.6 人员管理模块.....	31
5.1.7 系统管理模块.....	32
5.3 本章小结.....	34
第六章 系统测试	35
6.1 功能测试.....	35
6.1.1 移动端用户业务功能测试用例.....	35
6.1.2 PC 端用户业务功能测试用例.....	36
6.2 本章小结.....	36
结 论	37
参考文献	38
致谢	39

第一章 绪论

1.1 研究背景

随着社会经济的发展和医疗技术的进步，口腔医院承担着越来越多的医疗任务。然而，口腔医疗工作的复杂性和工作量也随之增加。这不仅对医院管理带来了很大的压力，同时也给患者就医带来了困难。在这种背景下，口腔医院管理系统日益成为医院发展的必需品。

提高医疗服务质量。口腔医院管理系统将医生、护士、医院管理层、患者等多个方面的信息有机地联系在一起，从而为医疗服务提供了全面、细致的支撑。

1.2 国内外现状

目前在中国国内的口腔医院管理系统在目前尚属于发展初期，但随着社会信息化水平的提高，越来越多的口腔医院开始尝试使用信息化手段，以达到更高的工作效率和更好的医疗效果。目前，对中小医院来说，其网络信息管理系统的选择就应该以很高的通用性为主，因为当前的中小型的医院的管理模式和内部机构的组成差异都很大，所以，就需要当前的信息管理系统有很强的通用性。^[1]国内口腔医院管理系统的应用主要处于两方面：一是预约挂号功能，二是电子病历管理。对于患者来说，预约挂号可以大大缩短患者等候的时间，为患者提供快捷的就诊服务。目前市面上有许多医院预约挂号的平台，例如“好医生”、“挂号网”等，这些平台通过网络技术让患者能够方便地完成预约挂号、在线支付等操作，提高了患者就医的便捷性和舒适度。另一方面，电子病历管理在国内口腔医疗行业也逐渐得到应用，并被纳入国家医疗信息化建设中。通过建立科学合理的口腔病历模板，医生可以更加准确地记录诊断，对治疗效果进行监控和分析，为医生提供更准确的医疗信息和诊疗方案。综上所述，医院信息管理系统的实施，把药品的管理完全通过计算机来完成，完全改变了以前手工的资料统计、查询。更提高了院长做决策的效率和准确度；减少药品因管理不当造成的损失；缩短了病人看病的等待时间。^[2]

在国外对于口腔医院管理系统的研究和应用已经处于比较成熟的发展阶段。欧美国家一直把医疗信息化工作作为医疗卫生事业的重要组成部分，致力于提高医疗服务效率和质量、降低医疗费用、保护患者隐私等方面。现阶段，国外口腔医院管理系统主要体现在系统优化功能更加全面，国外的口腔医院管

理系统不仅包括基本的预约挂号、病历管理等功能，还包含诊疗过程中所需的所有工具，如 X 光片、视频通话、诊断系统等；数据化程度高，国外许多口腔医院管理系统提供数据分析功能；医疗设备智能化，在国外口腔医疗系统中，医疗设备智能化技术得到了广泛的应用。系统可以接收来自医疗设备的数据，并将其整合到电子病历中，以提供更准确的诊断和治疗方案。总体而言，国外口腔医院管理系统更加完善和智能化，与传统的口腔医疗模式相比，这些系统确实具有更高的效率和更好的医疗结果，并为医生和患者提供了更好的服务。

总之，只要医院能在信息化进程最大限度地把握主动医院必能在信息化的带动下以最快的速度进入真正的医院现代化。^[3]

1.3 研究目的及意义

其研究的目的在于提高医院管理效率。口腔医院管理系统的建立，将大量的信息录入和查询等工作自动化，大大提高了医院工作效率。

提高患者就医体验。传统的口腔医疗模式存在一些问题，例如必要时需等待时间长、诊断不够及时等等。而系统化的口腔医院管理，可以通过设立预约功能、提示患者取药时间等方法，提高患者的就医体验。此外，口腔医院管理系统还可以通过移动端提供远程咨询服务，方便患者获取医疗服务。

综上所述，口腔医院信息系统建设完成不仅为医生和患者带来便捷,整个口腔医院日常活动产生的信息在系统中进行统计分析后能够为医院管理层决策提供数据支撑,如医院财务管理、物资管理、人力资源管理等系统能够为管理层提供医院日常运营情况统计分析。^[4]口腔医院管理系统的设计及实现对于口腔医疗行业的高效、规范和顺畅发展具有重要意义。其能够为医生提供全方位的工作支持，提高了工作效率和医疗服务质量；对于医院来说，可以提高管理效率和精细化水平；对于患者来说，可以有效地提高治疗效果和医疗体验。只有对医学信息进行科学、有效地管理，才能保证医疗工作的正常进行，从而促进医院各项业务工作的快速发展，而信息管理体系是医院信息管理工作正常开展的基础。^[5]

1.4 主要研究内容

本篇口腔医院管理系统论文的主要研究内容，围绕着如何设计和实现一个高效的口腔医院管理系统，旨在解决口腔医院管理面临的问题，提高医疗服务质量和管理效率。主要的研究口腔医院管理系统的设计与架构，针对口腔医院管理系统的特点和需求，本论文提出了一种面向对象的设计方法，采用前后端

分离的架构，将系统分为前端展示和后端数据处理两部分。前台移动端实现了预约挂号、在线病历查询等方便患者及前端后台管理实现预约挂号、在线病历查询、缴费等方便工作人员的服务接口；后端包括医护信息管理、科室管理、挂号信息管理、病历管理等任务区域，并采用 Java 语言实现开发。本论文所设计的口腔医院管理系统，采用了 Spring Boot 和 Vue2.0 等高效的开源框架，强调了系统的高可用性和可扩展性，使得系统更加灵活和方便进行修改和维护。

1.5 本文结构安排

论文共由六个章节和总结组成，主要结构安排如下：

第一章，绪论，介绍课题研究背景、研究意义和主要研究内容，并对国内外技术现状做简要分析。

第二章，相关技术介绍，介绍了整个系统设计中所使用的相关技术及环境的介绍，包括开发环境、开发语言、开发使用的框架。

第三章，系统分析，再开发系统前，对系统进行可行性分析和系统功能需求分析。

第四章，系统设计，根据系统分析的功能需求分析对系统进行设计，包括了数据库的设计和系统功能模块的设计。

第五章，系统实现，对整个系统进行编码实现，包括实现各个功能，以及对整个业务流程的描述。

最后是总结部分，得出结论，总结设计的完成度，同时指出整个系统设计开发工作的不足之处和需要改进的地方。

第二章 相关技术介绍

2.1 开发环境

JAVA 版本: Java 8

前端开发框架: Vue-cli

前端开发工具: IntelliJ IDEA 2021.3.2

前端服务器: Node.js 16.14.2

后端开发框架: spring boot

后端开发工具: IntelliJ IDEA 2021.3.2

后端服务器: Tomcat 8.5.72

数据库: MySQL 8.0.27

数据库可视化工具: Navicat Premium 12

2.2 JAVA 语言

Java 是一种高级编程语言，由 Sun Microsystems 公司的 James Gosling 与他的团队在 20 世纪 90 年代初开发的。它是一种面向对象的编程语言，其设计目的是用于在跨平台计算机网络环境中编写应用程序。相比于其他编程语言，Java 的最大优势在于其“一次编写，到处运行”的特性，因此能够大大减少开发难度和成本。

Java 的语法相对简单，易于学习。它有类、接口、继承、多态等面向对象的特性，并且支持异常处理和垃圾回收机制，能够帮助程序员更好地掌控程序的运行状态和资源利用。此外，Java 也提供了大量的 API（应用程序接口）和库，许多常见的编程问题都能够通过直接调用 API 来解决，如文件输入输出、网络通信、图形化操作等等。

Java 还有着广泛的应用领域。它可以用于开发 Web 应用程序、桌面应用程序、游戏、移动应用等等。很多大型的企业级应用也都是基于 Java 开发的。因此，Java 是一种非常有前途和实用价值的编程语言，尤其适合那些追求跨平台、可靠性和高效性的开发者。

在 Java 的发展历程中，还衍生了一系列技术和框架，如 JSP、Servlet、Struts、Hibernate、Spring 等等。这些技术和框架能够大幅度地提高 Java 开发的效率和质量，也极大地推动了 Java 在 Web 开发和企业级应用开发领域的普及和推广。

当然，Java 语言的不足之处也是存在的。由于其程序需要运行在 Java 虚拟机（JVM）上，因此程序的性能无法和一些底层语言相比，并且开发的过程可能需要比其他语言更多的工作量。此外，Java 的安全性也备受争议，因为其容易受到黑客攻击。

总之，Java 作为一种高级编程语言，具有良好的跨平台性、可读性、可靠性和扩展性。它在自身基础上研发了许多技术和框架，使得 Java 开发逐渐成为一项非常成熟和广泛应用的技术。如今，Java 的应用场景和领域还在不断拓展和扩大，相信它将在未来继续成为开发者首选的语言之一。

2.3 Vue

Vue.js 是一个 JavaScript 的渐进式框架，用于构建用户界面。它的理念是将视图（View）层抽象成组件化的方式，通过数据（Data）层的绑定，简化了前端开发的流程，并提高了代码的可维护性和可重用性。

Vue.js 框架的特点：

- （1）双向数据绑定：Vue.js 通过指令（Directive）的方式实现双向数据绑定，无需手动操作 DOM 元素，能够很方便地更新视图。
- （2）组件化：Vue.js 将一个页面分成多个组件，每个组件都有自己独立的作用域，能够增加代码的可读性和可维护性。
- （3）模板语法：Vue.js 使用简洁的模板语法，能够更容易地理解和维护代码。
- （4）轻量级：Vue.js 的文件大小只有 20KB，能够快速加载并执行。
- （5）生命周期：Vue.js 拥有丰富的生命周期函数，能够更好地控制组件的渲染过程。
- （6）插件化：Vue.js 支持插件机制，开发者能够将自己的插件添加到 Vue.js 中，增加功能并拓展框架。

除了上述特点，Vue.js 也具有完善的社区生态系统，例如 Vue-CLI、Vue Router、Vuex 等插件，能够快速构建完整的应用程序。

Vue.js 2.0 然后对比 1.x 版本进行了以下改进：

- （1）更加快速和轻量级，Vue.js 2.0 通过对虚拟 DOM 的优化，能够更快地响应数据变化，同时减小了底层代码的复杂程度。
- （2）更加简化的 API，Vue.js 2.0 将多个 API 进行了简化和重构，使得开发者能够更加方便地使用和理解。
- （3）更好的开发体验，Vue.js 2.0 改进了一些调试工具，能够更好地协助开发者排查问题。

（4）更好的组件化能力，Vue.js 2.0 优化了组件之间的通讯方式，增强了组件的复用性和可维护性。

总体来说，Vue.js 是一个非常优秀的 JavaScript 框架，无论从体积、性能、易用性还是扩展性上，都具有很高的价值和前景。在未来的 Web 开发中，Vue.js 将会有更广泛的应用场景。

2.4 Spring Boot

Spring Boot 框架是一个用来构建基于 Java 的企业级应用的框架。它是 Spring 框架的一部分，但比 Spring 更加轻量级、便利和快速。Spring Boot 基于约定优于配置的开发原则，使用默认配置而不是 XML 文件或其他方式，从而减少了开发人员的配置工作。Spring Boot 支持很多应用程序开发方面，包括 web、rest、安全性、事务、JPA、集成测试和许多其他领域。

Spring Boot 框架的主要特点包括：

（1）快速启动：Spring Boot 可以快速启动并运行程序而无需像其他应用程序一样重新构建应用程序。这使得开发人员可以节省时间和精力。

（2）简化配置：Spring Boot 使用默认配置并提供一组工具来轻松配置应用程序。这减轻了开发人员的负担，并允许他们更快地构建应用程序。

（3）嵌入式服务器：Spring Boot 框架包含多个嵌入式服务器，例如 Tomcat、Jetty、Undertow，并且支持无缝切换。这意味着开发人员不再需要为应用程序安装和配置 Web 服务器，可以直接使用内嵌服务器。

（4）自动配置：Spring Boot 框架通过自动配置过程，会根据应用程序的依赖关系和配置文件来自动配置应用程序。这消除了开发人员手动配置应用程序的必要性。

（5）生产就绪功能：Spring Boot 提供包括指标、健康检查、远程 Shell、日志记录等在内的许多生产准备好的功能，并可以轻松地添加到应用程序中。

（6）独立性：Spring Boot 不依赖于其他框架或库，并包括必要的依赖关系，因此应用程序在任何环境中都可以很好地运行。

（7）可扩展：Spring Boot 支持许多插件，例如 Maven 插件和 Gradle 插件，这些插件可以使开发人员更好地工作并增强应用程序的功能。

Spring Boot 是一种可靠、高效、轻量级和快速的框架，用于构建 Java 应用程序。它提供了各种功能和插件，使得开发人员可以更加轻松地开发应用程序。Spring Boot 也支持多种技术栈，包括 web、rest、安全性、事务、JPA 等，在应用程序开发中非常有用。

2.5 MyBatis-Plus

MyBatis-Plus 是 MyBatis 框架的一个增强工具包，它在 MyBatis 的基础上增加了许多常用的功能模块，使得我们在进行数据操作时更加高效、方便。

MyBatis-Plus 所实现的功能包括：通用 Mapper、分页插件、性能分析插件、自动填充插件、数据范围插件和多租户插件等。下面就分别对这些功能进行介绍。

通用 Mapper 是 MyBatis-Plus 中的核心功能之一，它实现了对 Mapper 类自动生成单表的 CRUD SQL 语句。为我们节约了编写 SQL 语句的时间和精力，同时也保证了 SQL 语句的正确性。在使用通用 Mapper 时，我们只需创建一个继承了 BaseMapper 接口的接口类，就可以直接使用该方法来操作数据表，而无需自己手写 SQL 语句。

分页插件是 MyBatis-Plus 中的又一个工具，它提供了两种常用的分页方式：常规分页和物理分页。常规分页就是在查询语句中使用 LIMIT 关键字限制返回的数据行数，而物理分页则是在查询语句中加入 OFFSET 和 FETCH NEXT 关键字。使用 MyBatis-Plus 实现分页功能非常地方便，只需在调用方法时传入页数和每页行数即可。

性能分析插件是 MyBatis-Plus 中的又一个提升开发效率的工具。它会打印出 SQL 语句的执行时间、执行次数、执行结果等信息，帮助我们快速发现代码中的性能问题。使用该插件非常简单，只需在 MyBatis 配置文件中加入插件的配置即可。

自动填充插件也是 MyBatis-Plus 中的一个非常有用的功能，它可以在插入或更新数据时，自动填充一些列的值，比如创建时间、更新时间、创建人、更新人等等。使用自动填充插件需要实现 MetaObjectHandler 接口，然后在配置文件中加入该插件的配置即可。

数据范围插件是 MyBatis-Plus 中的又一个非常实用的功能模块。它可以帮助开发者快速实现数据范围的过滤，比如只查询当前用户可以访问的数据。使用该插件需要实现 DataScopeInterceptor 接口，然后在配置文件中加入该插件的配置即可。

多租户插件则是 MyBatis-Plus 中的一种特殊功能模块，它可以帮助开发者实现多租户的数据隔离。使用该插件需要实现多租户的处理器，然后在配置文件中加入该插件的配置即可。

总之，MyBatis-Plus 是 MyBatis 框架的一个增强工具包，它大大提高了我们进行数据操作的开发效率，同时也降低了代码的复杂度。

2.6 MySQL 数据库

MySQL 是一种流行的关系型数据库管理系统，它被广泛应用于 Web 应用程序的开发和管理中。MySQL 是免费开源的，使用简单、高效、稳定，被广泛使用，是因为它优秀的性能、灵活的结构以及广泛的支持。它支持多个操作系统平台，如 Windows、Linux、Mac OS 等，并且容易安装和配置。MySQL 可以帮助用户存储、更新、管理和查询数据。在本文中，我们将探讨 MySQL 数据库的主要特点和其优势。

MySQL 数据库的主要特性：

（1）高性能：MySQL 数据库系统具有高效的存储和查询功能，允许数据的快速插入、更新和查询，从而提高用户对数据的处理效率。

（2）安全性：MySQL 数据库可以为数据提供严格的安全措施来保护数据的完整性和安全性。主要的安全特性包括强密码、SSL 加密和访问控制等等。

（3）可扩展性：MySQL 数据库是一个高度可扩展的系统，可以通过添加额外的存储容量、集群配置和负载平衡等功能，以处理不断增长的数据，满足用户不断变化的需求。

（4）标准化：MySQL 数据库符合 ANSI SQL 标准，这意味着它可以为不同平台和编程语言提供固定且稳定的数据格式，并且能够快速协作和整合。

（5）开源性：MySQL 数据库是一个开源产品，可以大大降低用户的成本。它还拥有庞大的社区支持和开发者社区，可以提供最新的解决方案和应对问题。

MySQL 数据库的优势：

（1）简单易用：MySQL 存储和查询数据的语句简单明了，易于操作和管理。它还提供了图形用户界面和工具，以使用户轻松地管理和维护数据库。

（2）高可靠性：MySQL 数据库系统具有良好的稳定性和健壮性，可以长时间运行，减少断电或其他问题带来的数据丢失风险。

（3）性能优良：MySQL 数据库可以快速处理大量的数据，并且可以针对复杂的查询，提高查询效率。它还支持带有特殊字符和语言的查询。

（4）可移植性：MySQL 数据库系统可以在不同的系统和平台上运行，这使得它成为一种高度便携的数据库解决方案。

（5）可扩展性：MySQL 数据库可以轻松地进行扩展，可以增加多个服务器和容量以满足用户不断增长的数据量 and 处理需求。

总的来说，MySQL 数据库可以帮助用户存储、更新、管理和查询数据，具有高性能、安全、可扩展、简单易用等优秀的特性。

2.7 本章小结

本章节主要介绍设计与实现口腔医院管理系统所涉及的开发环境、开发语言、开发框架和数据库，分析它们的特点及优势，为接下来的系统设计开发做好技术准备。

第三章 系统分析

3.1 可行性分析

3.1.1 操作可行性分析

近年来，随着医疗服务的提高与人们对口腔健康的重视，全国范围内口腔诊所与口腔医院相继出现，但是由于其大量的日常业务，医院管理难度大、人力物力成本高，也面临着一系列的挑战。为了解决以上的问题和提高口腔医院的管理效率，智能化的口腔医院管理系统应运而生。

口腔医院管理系统是一款复杂的软件系统，需要考虑到多个方面的因素，包括用户需求、系统稳定性、易用性及数据管理。如果这几个方面都能得到有效且合理地运用，则系统的实施会更加简单方便，为医务工作者和患者提供更好的服务，将提高患者的就医体验，提高医院的服务质量和效率，为口腔医院开展工作提供了保障和支撑。

3.1.2 技术可行性分析

目前，随着信息技术的不断发展，许多先进的技术可以应用于口腔医院的管理中。例如，网络技术、数据存储技术、自然语言处理和机器学习等技术。这些技术可以实现医院信息的高效传输、存储和分析，从而实现更高效、更精确的管理。此外，硬件方面也能得到充分的保障，因为随着计算机技术的不断改进，硬件设备的性能和精度将大大提高。

3.1.3 经济可行性分析

口腔医院管理系统的成本可能与传统的手工记录方式相当，但其效率更高，操作更简单，可以节省更多的时间和资源。因此，这种新型的管理方式可以为医院节省大量的人力物力。使用管理系统可以大大提高医院的服务质量，减少人为因素的干扰，减少医疗差错，采用这种管理方式更有利于提高医院的整体效益。

3.2 需求分析

3.2.1 系统需求

口腔医院管理系统的系统需求分析旨在更好的为口腔医院的管理提供技术支持，提高医院的运营效率和质量，为患者提供更好的医疗服务。系统需求分析以了解口腔医院管理系统的整体需求，确定需求的准确性、可行性和实现场景等。其中功能范围包括挂号预约管理：患者在移动端挂号预约录入信息、生

成挂号号码，后台可以查看挂号预约的信息；科室管理：科室信息的录入和维护；病历管理：病历信息的录入和浏览，以及病人病历的查询。收费管理：对诊治、药品等医疗服务的收费。药品管理：系统可以进行药品的查询、出入库等；人员管理：对患者和医院的医护工作者的信息进行管理；系统管理：系统的数据人员信息管理。系统环境的需求：硬件环境：服务器采用 Linux 系统，要求具备较好的处理器和内存，硬盘存储空间要求不低于 500GB。软件环境：数据库采用 MySQL 数据库，Web 服务器采用 Apache Tomcat，操作系统采用 Linux。前端采用 Vue.js 2.0 实现，后端采用 Spring Boot 框架。

3.2.2 功能需求

本系统需要将患者、医护人员和患者的挂号预约病历信息进行管理。其中患者需要在移动端进行挂号预约，后台医生才能看到其的挂号预约信息，医生给患者进行诊疗，开完药后会生成药物单给计价员进行医疗服务的总体计价，然后生成缴费订单。本系统还对每个科室、人员、权限分配进行管理。本系统的主要用户是患者、医护人员、系统管理员，所以需要对用户角色处理。同时需要获取和保存用户的相关信息，需要用户登录、检验用户角色，保证数据访问的安全性。综合上述业务需求描述，得出如图 3-1 所示的需求功能模块图。

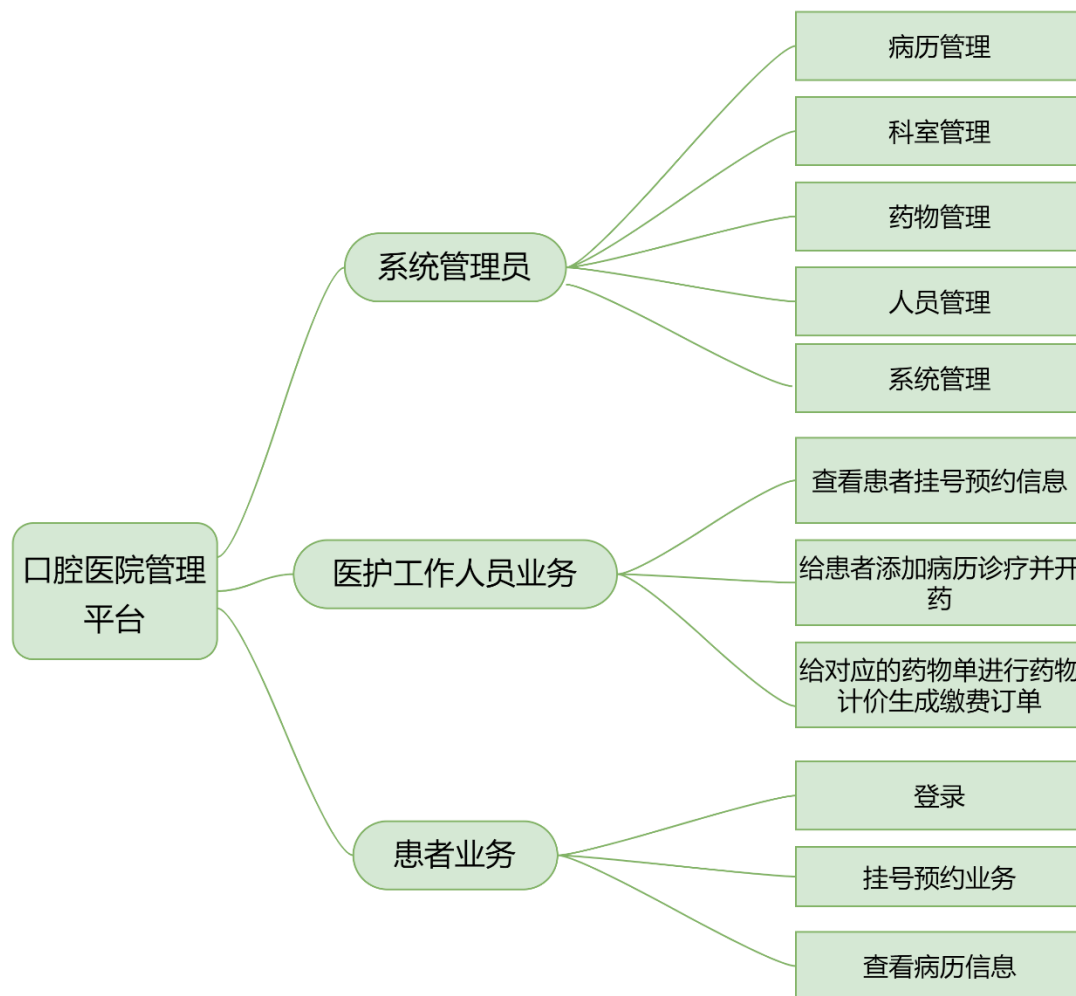


图 3-1 需求功能模块图

（1）病患操作：

病患进行看诊操作，病人在挂号台进行登录之后挂号预约，包括挂号预约信息的录入、生成对应编号。可以查看自己的历史病历信息。

（2）医生看诊的管理：根据挂号预约信息，对病患进行看诊操作，并对患者开药生成药物单（处方）。

（3）财务员的操作：对生成的药物单进行对该患者的医疗服务计价，生成缴费订单。

（4）药物管理人员：对药品信息的维护和药品的出入库进行管理。

（5）系统管理员：对系统的挂号预约模块、科室管理模块、药物管理模块、缴费订单模块、人员管理模块和系统管理模块进行管理，其中系统管理模块包括对每个角色的权限管理。

3.3 业务流程

整个口腔医院管理系统主要的业务流程是首先患者进行挂号预约，系统会把患者的挂号信息或者预约信息进行保存，然后在后台管理展示患者的挂号或者预约信息，医生对挂号预约的患者进行就诊，并新增患者的电子病历和对应的药物单，财务人员再把药物单进行计价后生成缴费订单，最后患者给对应订单进行缴费。根据以上描述画出业务流程图，如图 3-2 所示。

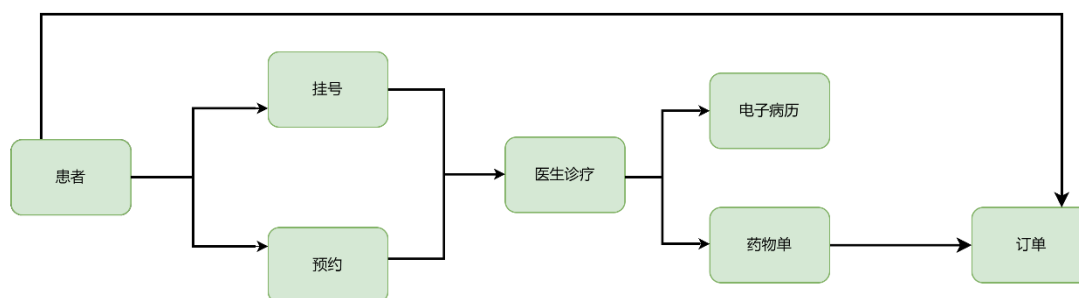


图 3-2 业务流程图

3.4 本章小结

本章结的需求分析导出了口腔医院管理系统的基础逻辑模型，为口腔医院管理系统提出了更加具体、准确、清晰的需求，为接下来的系统开发指明了开发方向，同时为后阶段的系统测试工作提供了依据。

第四章 系统设计

4.1 数据库设计

4.1.1 E-R 模型

（1）系统的整个业务关系 E-R 图

患者是需要接受治疗的人，每个患者都有唯一的患者 ID、姓名、性别、联系方式等个人信息；医护、财务员、药物管理人员分别是负责给患者提供治疗建议和治疗方案、对药物单进行计价以及对药物库进行管理的人，每个医护人员、财务员、都有唯一的用户 ID、姓名、职称等信息；科室是诊断和治疗患者的场所，每个科室都有唯一的科室 ID、名称和科室医生人数等信息；电子病历和药物单是医生给患者进行诊疗后得出的诊断结果，每个病历和药物单都有唯一病历 ID 和药物单 ID 以及对应记录诊断的描述和用药情况；缴费订单是患者看完病后生成的缴费订单，有唯一的订单 ID 和用户的订单相关信息。

其中医护与患者之间存在诊疗关系，即一个医生可以为多个患者提供治疗服务，而一个患者只能被一个医生接诊；医生和科室之间存在分配关系，即一个医生只能分配到一个科室工作，而一个科室可以分配多名医生工作；患者与科室之间存在预约关系，即一个患者可以在一个科室预约看病，而一个科室可以拥有多个患者；医生和病历、药物单存在新增病历和开药的关系，即一个病历或者药物单只能被一个医生产生，而一个医生可以产生多个病历或者药物单；患者和病历、药物单存在拥有关系，即一个病历或药物单只属于一个患者，而一个患者可以拥有多个病历和药物单；电子病历和药物单存在对应关系，即一个病历对应一个药物单，一个药物单对应一个病历；财务人员和药物单存在计价关系，即一个药物只能被一个财务员计价，而一个财务员可以对多个药物单进行计价；财务员和缴费订单存在管理关系，即一个订单只能被一个财务员负责生成，而一个财务员可以生成多个订单；药物管理员与药物库存在管理关系，即一个药物可以被多个药物管理员管理，一个药物管理员也可以管理多个药物。综上所述可以得知系统的主要业务关系分别有患者、医护人员、财务员、药物管理员、科室、单子病历、药物单、药物库和缴费订单这九个实体，画出口腔医院管理系统的整体业务关系数据库 E-R 图，如下图 4-1 所示。

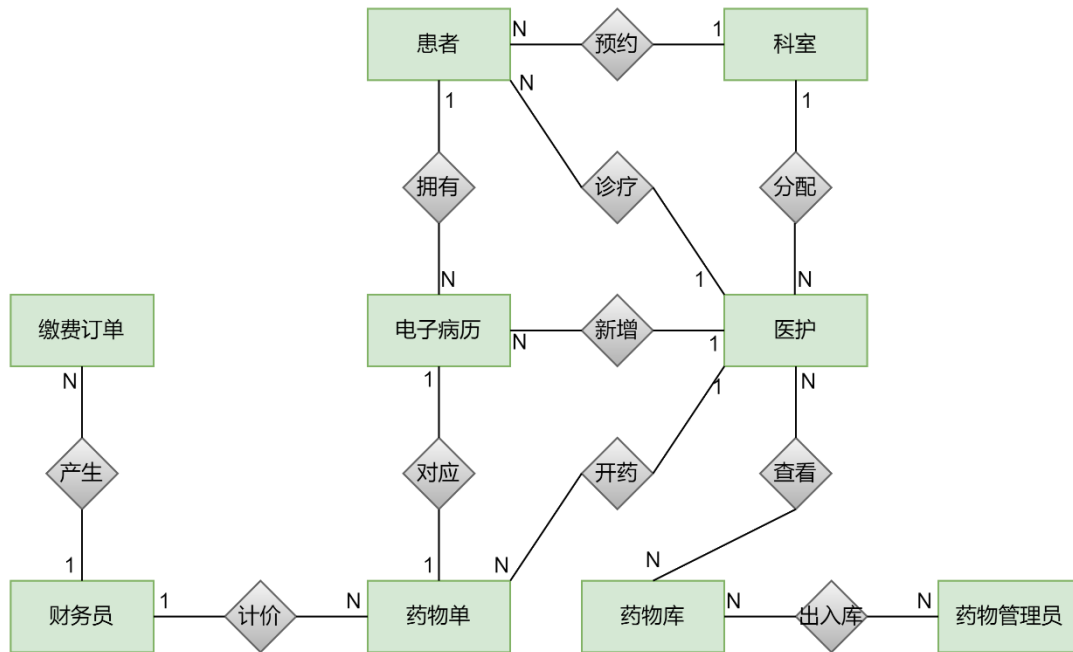


图 4-1 口腔医院管理系统的整体业务关系数据库 E-R 图

(2) 再者，其中在整个后台 PC 端管理系统里每个用户都有拥有不同的角色，不同的角色拥有不同的访问权限。一个用户的角色数目不设限制，用户与角色的关系为多对多关系。而每个角色有各自的系统菜单模块的不同，每个用户与菜单形成多对多的关系，每个角色和菜单也有多对多的关系。所有根据这些关系画出系统用户角色和菜单的权限关系 E-R 图，如图 4-2 所示。

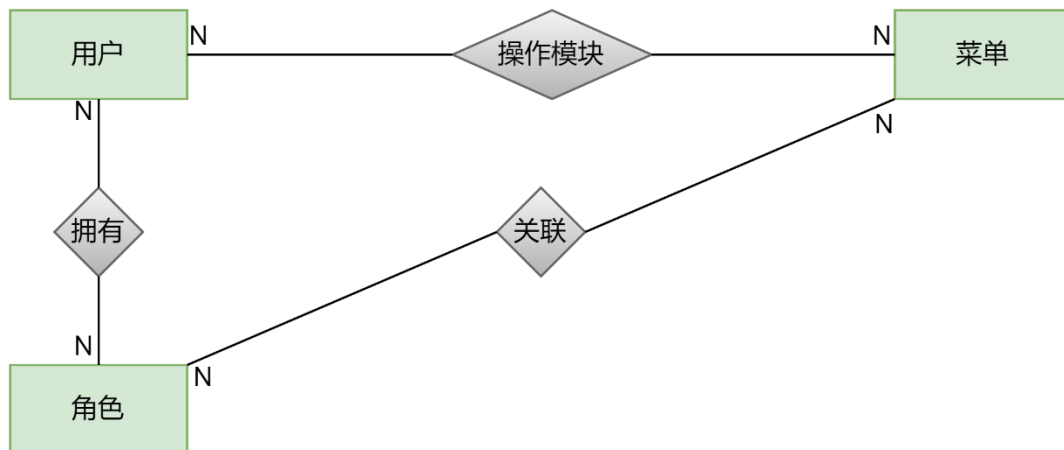


图 4-2 系统用户角色和菜单的权限关系 E-R 图

4.1.2 数据库表设计

（1）医护和工作人员信息表

在用户人员信息表中应该包含个人基本信息、登录信息，具体字段有：用户 ID（主键）、用户名、密码、用户真实姓名、邮箱、手机号、性别、使用状态等字段。设计出字典数据如表 4-1 所示。

表 4-1 用户信息表

字段	数据类型	长度	主/外键	允许空	备注
user_id	int	0	主键	否	用户 ID
username	varchar	50	——	否	用户名
password	varchar	100	——	是	密码
email	varchar	100	——	是	邮箱
mobile	varchar	100	——	是	手机号
status	tinyint	0	——	是	状态
create_user_id	int	0	——	是	创建者 ID
create_time	datetime	0	——	是	创建时间
name	varchar	255	——	是	真实姓名
sex	int	0	——	是	1:男 0: 女
modify_user_id	varchar	255	——	是	修改者 ID
modify_time	datetime	0	——	是	修改时间

（2）患者信息表

患者信息表包含患者的基本信息：患者 ID、患者名称、患者性别、患者手机号、患者年龄、用户状态和患者邮箱，由于患者是在移动端通过手机号码发送验证码登录，所以不需设置密码。设计出数据字典如表 4-2 所示。

表 4-2 患者信息表

字段	数据类型	长度	主/外键	允许空	备注
patient_id	varchar	0	主键	否	患者 ID
patient_name	varchar	255	——	否	患者姓名
sex	int	0	——	是	性别
age	varchar	255	——	是	年龄
mobile	varchar	255	——	否	手机号
email	varchar	255	——	是	邮箱
status	int	0	——	是	状态

(3) 科室信息表

科室信息表包含有科室的基本信息：科室 ID、科室名称、科室主任、医生人数、剩余可预约人数、科室位置、科室使用状态和科室描述，具体设计的数据字典如 4-3 表所示。

表 4-3 科室信息表

字段	数据类型	长度	主/外键	允许空	备注
dept_id	int	0	主键	否	科室 ID
dept_name	varchar	255	——	否	科室名称
doc_num	int	0	——	否	医生人数
dept_room	varchar	255	——	是	科室房间
status	varchar	255	——	是	状态
director	varchar	255	——	是	科室主任
residual_num	int	0	——	是	预约人数
description	varchar	5000	——	是	描述

(4) 电子病历信息表

电子病历信息表字段包括病历 ID、患者 ID、医生名、护士名、是否复诊、第几次就诊、诊断描述、治疗方案、治疗过程医嘱、就诊时间、药物单 ID、患者名称和治疗科室 ID。具体数据字典的设计如表 4-4 所示。

表 4-4 电子病历信息表

字段	数据类型	长度	主/外键	允许空	备注
ele_records_id	int	0	主键	否	病历 ID
patient_id	varchar	500	外键	否	病人 ID
doc_name	varchar	255	——	是	医生
nurse_name	varchar	255	——	是	护士
is_referral	int	0	——	是	是否复诊
treatment_num	int	0	——	是	就诊次数
treatment_time	datetime	0	——	是	就诊时间
treatment_description	varchar	2000	——	是	诊断描述
treatment_method	varchar	2000	——	是	治疗方案
treatment_during	varchar	2000	——	是	治疗过程
doc_orders	varchar	2000	——	是	医嘱
prescription_id	int	0	外键	是	药物单 id
patient_name	varchar	255	——	是	患者名称

表 4-4 电子病历信息表（续）

字段	数据类型	长度	主/外键	允许空	备注
dept_id	int	0	外键	否	科室 ID

（5）药物单信息表

药物单信息表字段包含药物单号、患者 ID、科室 ID、开药时间、使用方法、开药医生、药剂师、财务员、状态和医院名称，具体数据字典设计如表 4-5 所示。

表 4-5 药物单信息表

字段	数据类型	长度	主/外键	允许空	备注
prescription_id	int	0	主键	否	药物单 ID
hospital	varchar	255	——	是	医院名称
patient_id	varchar	500	外键	否	病人 ID
dept_id	int	0	外键	否	科室 ID
prescription_time	datetime	0	——	是	开药时间
use_method	varchar	2000	——	是	使用方法
doc_name	varchar	255	——	是	医生
pharmacist	varchar	255	——	是	药剂师
estimator	varchar	255	——	是	财务员
status	int	0	——	否	状态

（6）药品信息表

药品信息表包含药品的基本信息以及出入库信息，字段有药品 ID、药品名称、药品剂型、药品规格库存量、单价、药品描述、药品图片、用法用量、存放地址、保质期、供应商、入库时间、入库批次、出库时间、出库批次和状态。具体数据字典设计如表 4-6 所示。

表 4-6 药品信息表

字段	数据类型	长度	主/外键	允许空	备注
drug_id	int	0	主键	否	药品 ID
drug_name	varchar	255	——	否	名称
drug_dosage_form	varchar	255	——	是	剂型
drug_size	varchar	255	——	是	规格
inventory	int	0	——	是	库存
price	varchar	255	——	是	单价
description	varchar	2000	——	是	药品描述

表 4-6 药品信息表（续）

字段	数据类型	长度	主/外键	允许空	备注
image_name	varchar	255	——	是	图片名称
usage_dosage	varchar	1000	——	是	用法用量
address	varchar	255	——	否	存放地址
life_time	varchar	50	——	是	保质期
vendor	varchar	255	——	是	供应商
inbound_time	datetime	0	——	是	入库日期
outbound_time	datetime	0	——	是	出库日期
inbound_batch	varchar	255	——	是	入库批次
outbound_batch	varchar	255	——	是	出库批次
status	int	0	——	是	状态

(7) 订单信息表

订单信息主要包含了订单 ID、患者 ID、药物单 ID、总价、生成订单时间和状态。具体数据字典的设计如表 4-7 所示。

表 4-7 订单信息表

字段	数据类型	长度	主/外键	允许空	备注
order_id	int	0	主键	否	订单 ID
patient_id	int	500	外键	否	患者 ID
prescription_id	int	0	外键	否	药物单 ID
sum	varchar	255	——	是	总价
order_time	datetime	0	——	是	订单时间
status	int	0	——	是	订单状态

(8) 医院医护和工作人员的角色信息表

角色表主要包括角色 ID、角色名称和角色描述。设计出数据字典如表 4-8 所示。

表 4-8 角色信息表

字段	数据类型	长度	主/外键	允许空	备注
role_id	int	0	主键	否	角色 ID
role_name	varchar	255	——	否	角色名称
remark	varchar	2000	——	否	备注

(9) 用户角色信息表

用户角色信息表是通过角色 ID 和用户 ID 把用户和该用户拥有的角色信息绑定起来，使用户和角色两张表分开管理，易维护。主要字段包含字段 ID、用户 ID

和角色 ID。如表 4-9 所示。

表 4-9 用户角色信息表

字段	数据类型	长度	主/外键	允许空	备注
id	bigint	0	主键	否	字段 ID
user_id	bigint	0	外键	否	用户 ID
role_id	bigint	0	外键	否	角色 ID

4.2 数据流设计

根据以上系统分析和业务流程可以知道一共有五个实体类，根据不同实体的业务画出数据流并把数据存入数据库对应的信息表。根据以上信息画出数据流程图如图 4-3 所示。

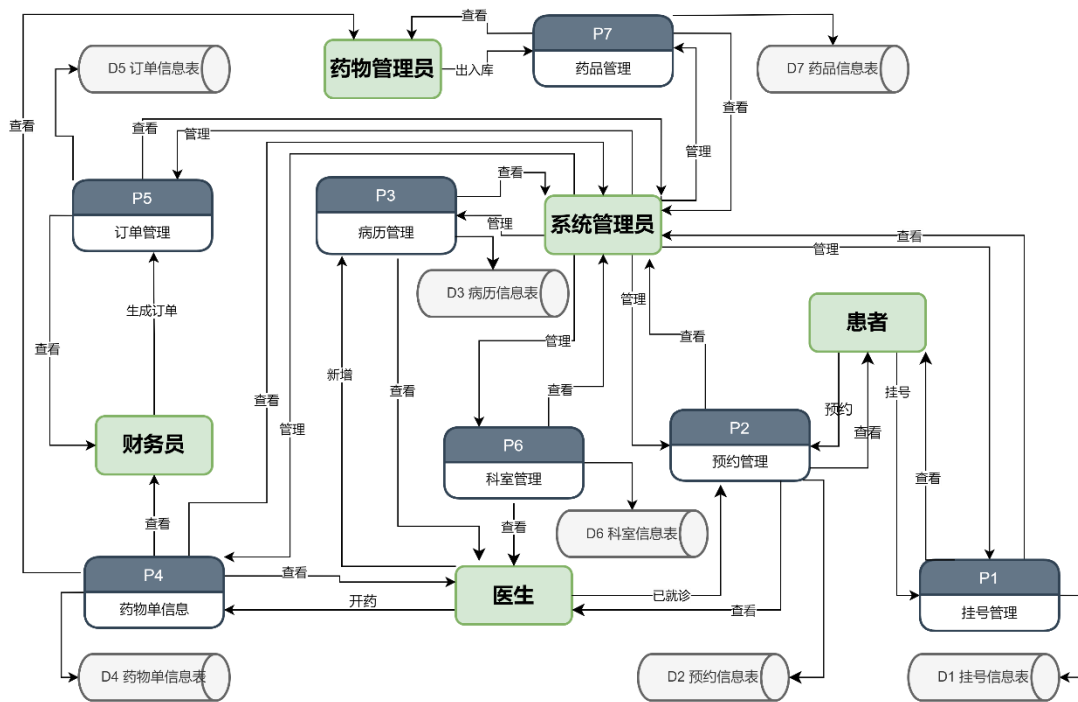


图 4-3 数据流图

4.3 系统功能模块设计

4.3.1 设计思想

根据需求分析的描述，建立起对应的实体结构，设计出的数据库表关系比较清晰，故采用开源的关系型数据库--MySQL 数据库。目前较为主流的程序设计语言，Java 语言具有安全性、稳健性等特征。目前 Java 处于成熟阶段、有较多 Java 语言开发工具和以 Java 语言为基础的大量实用便捷的框架，符合系统的

开发需求。对比目前其它主流的 Web 开发框架，如 Spring、Struts 2 等，Spring Boot 框架有快速打包部署的特点，符合课题系统开发需求。Java 是面向对象编程语言，系统涉及到对关系数据库的操作。基于便捷开发，本课题将使用持久化框架，采用前后端分离的结构，用面向对象的方式操作关系数据库。根据系统的功能需求分析以及系统流程最终确定系统的功能层次，并得出系统功能模块层次图如图 4-4 所示。

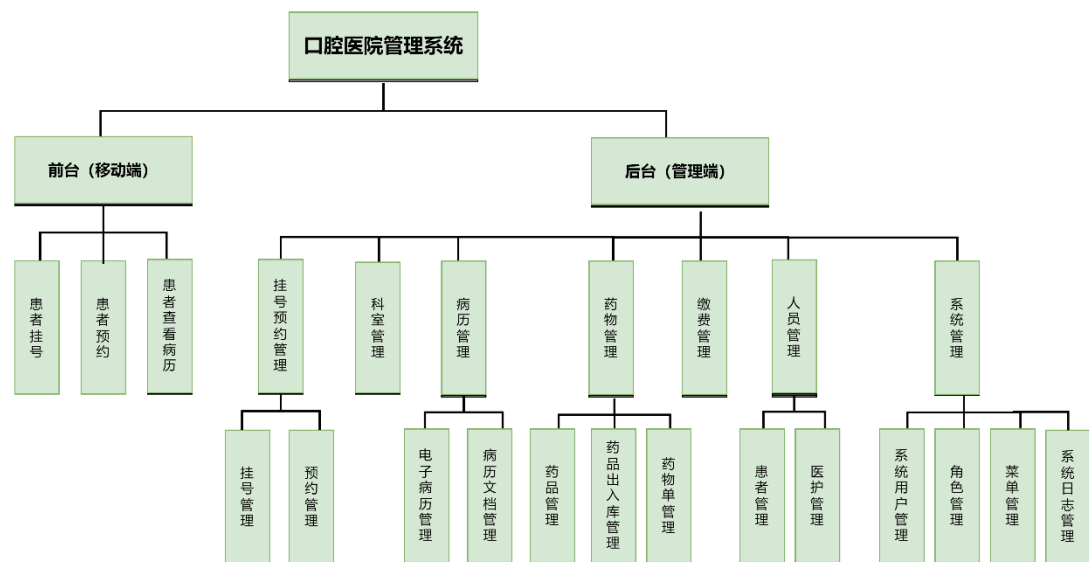


图 4-4 系统功能模块层次图

4.3.2 系统结构设计

根据项目需求分析，系统采用前后端分离的方式进行设计，前端包括口腔医院后台管理端和患者移动端；后端采用 MVC 的设计模式，将系统的数据、业务逻辑和用户操作界面分离开来。采用这样设计的优势是在日常维护过程中，例如当需要修改后端某个层的业务逻辑时，只需对该层的包进行修改，不会影响到其他层的数据，将代码进行模块化的管理，使其更具有扩展性和可维护性。整个口腔医院管理系统的架构设计图如图 4-5 所示。

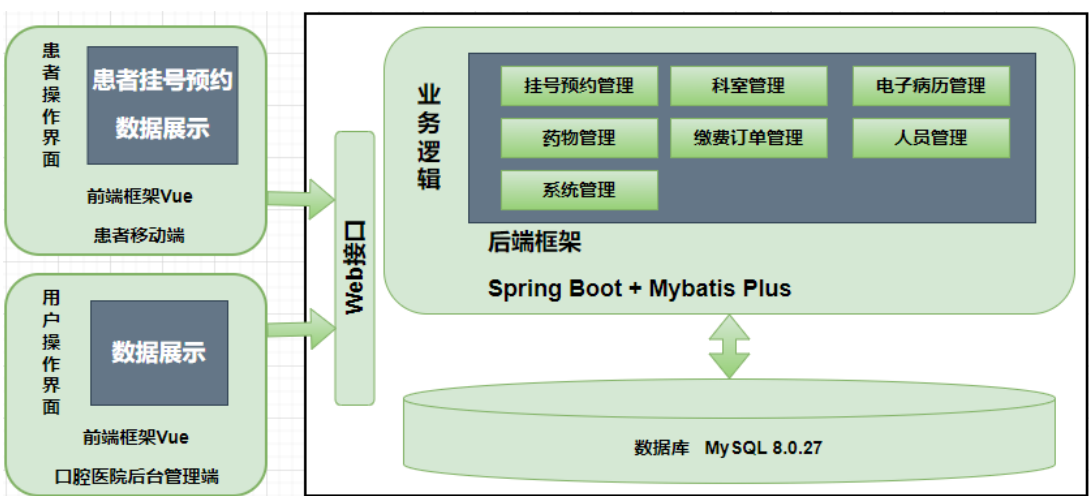


图 4-5 口腔医院管理系统的架构设计图

在上图 4-5 中，展示了口腔医院管理系统的整体设计架构，其中前端有患者使用的移动端前台，患者可以在这里进行挂号预约和查看相关信息；口腔医院的后台管理 PC 端，在这里医院的医护人员和其他工作人员以及管理员可以进行相对应的管理操作。

本系统的后端对前端提供 Restful 风格的接口，方便应用层的 PC 端浏览器、手机端方便调用接口，获取资源信息，同时使得接口更加规范。

后端业务逻辑这模块采用了 Spring Boot 和 MyBatis-Plus 框架。Spring Boot 简化了 Spring 应用的搭建和 Maven 的配置，便于快速开发。Spring Boot 内嵌了 Tomcat 服务器，符合珍稀项目系统的发开需求，同时简化了项目的部署步骤。项目后期可以将系统打包成 JAR 格式，直接在 Java 环境下运行。通过 MyBatis-Plus 的持久层框架，可以方便实现对数据的增删改查，有助于减少 SQL 语句的出错率和灵活操作数据库。

4.3.3 系统设计流程

整个系统采用 MVC 的设计结构，将系统的数据层、业务逻辑层、控制层和视图层分开，使得各个层之间的耦合度降低，更有利于系统的开发和维护。其中 Model、View、Controller 之间的关系如图 4-6 所示。

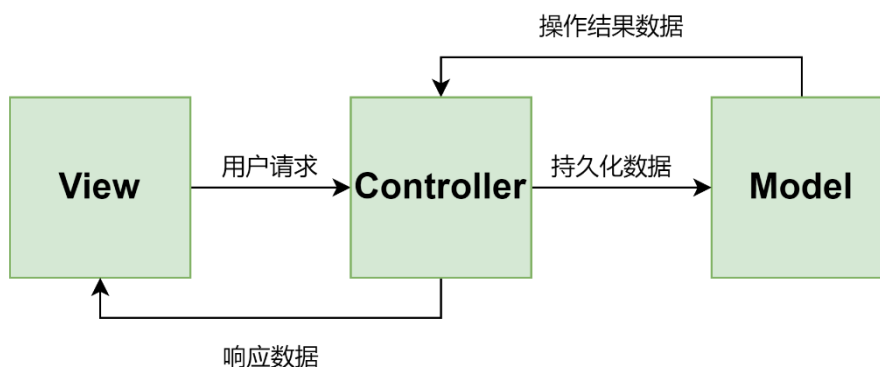


图 4-6 Model、View、Controller 关系图

系统的数据流程：其中，视图层是与用户交互的层，这一层将用户填写的表单信息，通过发送 HTTP 请求把数据传给后端服务，而控制层首先接收请求并分析数据，再把数据传给业务逻辑层，业务逻辑层在调用数据层将数据存入数据库；而向前端响应数据的流程也是一样，视图层需要显示的数据，通过发送 HTTP 请求给后端，后端的控制层接收请求，再调用业务逻辑层去操作数据层，通过数据层向数据库查询数据将其返回给控制层，然后控制层响应给视图层进行数据的展示。

系统的设计流程：在设计口腔医院管理系统时，首先依照数据的设计建立模型层（Model），模型层包括了数据层（DAO）和业务逻辑层（Service），确定需要的实体，建立对应的实体类并建立实体之间的关系类。后端使用 MyBatis-Plus 框架对数据表进行操作，然后编写控制层（Controller）请求的接收。最后，前端使用 Vue 编写视图页面，实现用户和系统的交互。

4.4 本章小结

在系统设计中，根据系统的需求分析，设计并建立起数据库表，再根据数据库表的关系进行对系统的整体模块结构设计，确定了系统的模块特征，为接下来的具体编码详细系统实现打下基础。

第五章 系统实现

5.1 移动端

5.2.1 用户登录模块

该模块主要是患者在手机移动端进行登录，且登录只要用户输入手机号码，点击发送验证码，通过验证码来实现登录操作，并且使用 `session` 保存用户 ID，当其退出登录后销毁 `session`。

验证码生成详细实现：当前端用户点击发送验证码后，后端接收到请求并执行生成验证码工具类随机生成 4 位数的验证码。这里并未实现通过第三方短信接口进行发送验证码，只是在后台生成后在控制台里打印出来。前端输入验证码后，点击登录页面跳转到 `index.html` 主页面，如果未登录，而直接访问 `index` 页面，页面首先会验证 `session` 是否存在，若不存在则自动跳转到登录页面。前端的关键代码实现：`sessionStorage.setItem("id",res.user.patientId)`通过 `sessionStorage.setItem` 在前端保存患者的 ID。前端实现如图 5-1 所示。



图 5-1 移动端登录界面图

5.2.2 用户挂号预约模块

患者登录后即可进行挂号和预约操作，患者在挂号或者预约中输入相关的信息后提交表单，后台接收患者请求的表单后进行分析，通过获取 session 中的用户 ID 对该患者进行 CRUD 操作。其中，患者可以挂号预约信息的录入和查看挂号预约信息、管理员和医护人员可以查看患者的挂号信息，得出用例图如图 5-1 所示。

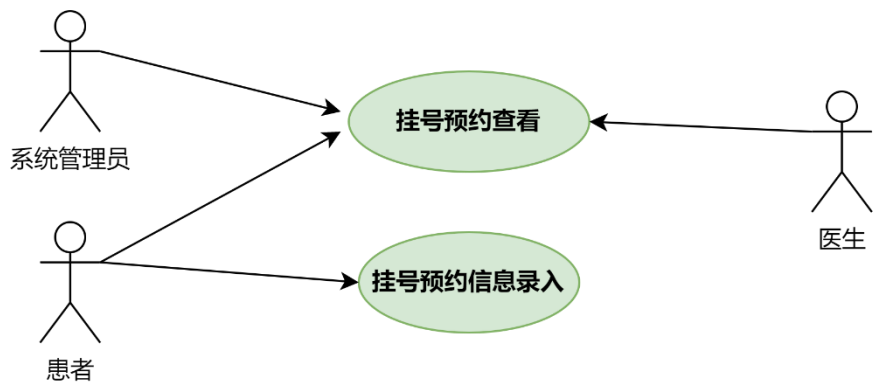


图 5-1 用户挂号预约功能用例图

患者挂号前端实现效果如图 5-2 所示。



图 5-2 用户挂号成功界面图

5.2 PC 管理端

5.1.1 挂号预约模块

当有患者进行挂号预约操作时，数据库有响应的数据后，后台管理端展示患者的挂号预约信息。在项目启动时通过`@EnableScheduling` 注释启动定时器，进行对挂号和预约的信息进行周期性的检查是否哪些挂号或者预约信息已过期。通过方法注释`@Scheduled(cron = "0 30 * * * ?")` 每 30 分钟执行一次检查，挂号预约信息是否已过期，截取当前系统时间与用户的挂号时间相减，如果大于 24 小时则把该挂号信息状态改为已过期，比较预约信息，如果当前时间在用户预约时间之后，则把该预约信息的状态置为已过期。关键代码如下：

```
//获取当前时间和挂号时间的差
long cha = nowTime.getTime() - ghTime.getTime();
//计算该差值的 24 小时制
double result = cha * 1.0 / (1000 * 60 * 60);
if(cha > 0 && result > 24){ //比较时间差大小
sysGhEntityList.get(i).setStatus(2);
//更新挂号信息状态为 2:已过期
sysGhService.updateById(sysGhEntityList.get(i));
```

根据系统功能需求，前端实现的效果如图 5-3 所示。

挂号ID	患者名	性别	年龄	手机号	挂号时间	状态	操作
21	www	男	23	18219135363	2023-03-27 21:30:55	未就诊	已就诊 删除
19	www	男	23	18219135363	2023-03-27 21:24:21	已就诊	删除
18	www	男	23	18219135363	2023-03-20 20:43:44	已过期	删除
17	www	男	23	18219135363	2023-03-20 20:40:35	已过期	删除
13	www	男	23	18219135363	2023-03-18 10:52:08	已过期	删除
12	kkk	女	23	19254802938	2023-03-16 00:40:43	已过期	删除
11	www	男	23	18219135363	2023-03-16 00:26:06	已过期	删除
2	bbb	女	13	1954652865	2023-02-28 10:01:15	已过期	删除

图 5-3 后台挂号管理界面图

5.1.2 科室管理模块

管理员可对该模块进行管理，对科室的查询新增修改删除的操作，可以对科室启用和禁用。后台通过 `MyBatis-Plus` 对数据库进行增删改查。通过分页查询，查询所有的科室信息并返回给前端展示。后端关键代码分析：


```
//构造分页构造器
IPage<SysDept> page=new Page<>(current,pageSize);
//条件构造器
LambdaQueryWrapper<SysDept>queryWrapper=new LambdaQueryWrapper<>()
//添加条件过滤
queryWrapper.like(Strings.isEmpty(deptId),SysDept::getDeptId,deptId)
page = sysDeptDao.selectPage(page,queryWrapper);//向数据库查询数据
```

5.1.3 电子病历模块

电子病历模块是医生对患者进行诊疗后新增一个病历，医生通过对患者的诊断结果进行填写相关信息，例如对该患者的诊疗描述，治疗过程和遗嘱进行填写还有上传多媒体病历，例如诊疗图片或者相关文件。值得注意的是这里医生需要对患者进行开药，则需要同时生成一个药物单信息，在给患者新增病历的同时新增一个唯一对应的药物单，该药物单记录了药品的信息。所以该模块后端详细实现过程中需要操作到四张信息表，一是电子病历信息表，二是药物单信息表，三是药品信息表，四是药物单对应的药品信息的关系表。

其中，根据病历 ID 查询病历详细信息，在操作数据库时，首先根据病历 ID 查出该条病历信息，然后通过药物单 ID 查询与之对应的药物单信息，再根据药物单 ID 查出所有药物单对应的药品信息的关系，最后逐个获取药品信息，再把所有信息整合封装后返回给前端。在医生对患者新增病历时首先需要保存药物单信息关键代码实现 `prescriptionService.save(prescriptionEntity)`，`prescriptionEntity` 为药物单实体类;然后根据患者 ID 查询刚刚保存的药物单 ID，把药物单 ID set 进病历实体然后保存患者病历和一些多媒体病历，例如图片。关键代码实现:`sysFileService.saveBatch(fileList)`，`fileList` 为图片文件列表;最后在把药物单 ID 和药品 ID 关联起来新增保存。

医生给患者进行诊疗并填写诊疗结果，上传多媒体病历，开出药方，选择药品信息以及填写药品数量和使用方法。前端实现如图 5-4 和图 5-5 所示。

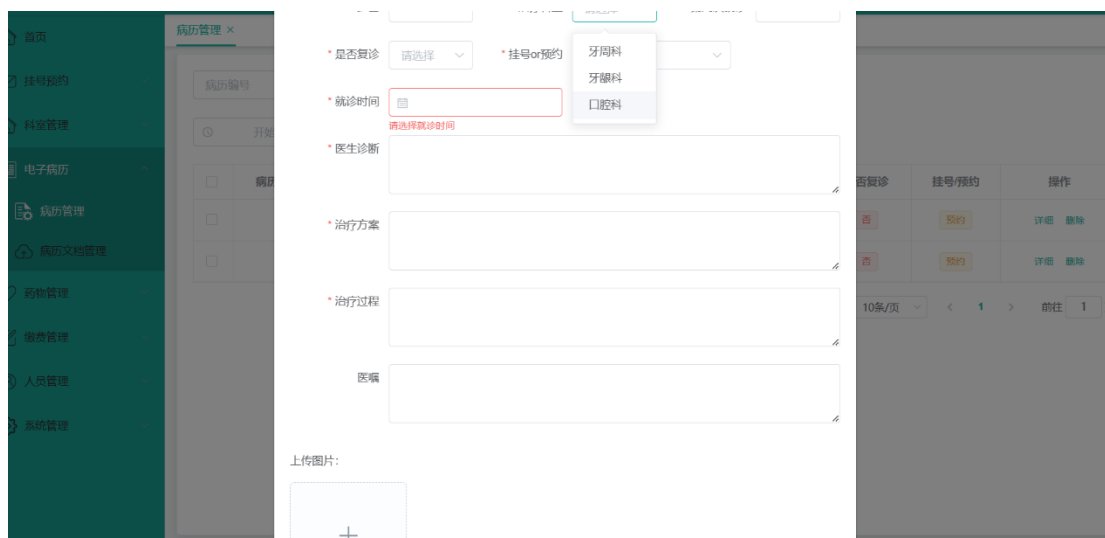


图 5-4 新增病历界面图



图 5-5 开药界面图

5.1.4 药物管理模块

药物管理模块包含了药品的管理和药品的出入库操作以及药物单的管理，医院管理人员可以对药品管理对药品进行 CRUD 的操作，对药品进行入库和出库操作。其中关键的是对药物单的管理，根据药物但 ID 查询该药物下对应的药物信息，则需要通过查询药物单对应药品信息的关系来确定该药物单对应的药品和药品数量信息。

后端首先需要根据药物单 ID 查询药物单的信息，然后再根据药物但查询药物单与药品关系，查出所有对应药品 ID，再通过 stream 流的方式根据药品 ID 逐个查询出该药品的信息并收集该实体信息，返回给前端一个列表对象信息。关键代码实现：

```
//使用 stream 流的方式
List<SysDrugEntity>sysDrugEntityList=drugPreEntityList.stream().map((item)->
{
SysDrugEntity sysDrugEntity = new SysDrugEntity();
...
return sysDrugEntity;
}).collect(Collectors.toList());//收集实体
```

药物管理分为药品管理、药品出入库和药物单的管理，药品分类展示所有药品信息，可以通过药品编号、选择药品、药品剂型和创建时间进行筛选查询药品信息。前端效果展示如图 5-6 所示。

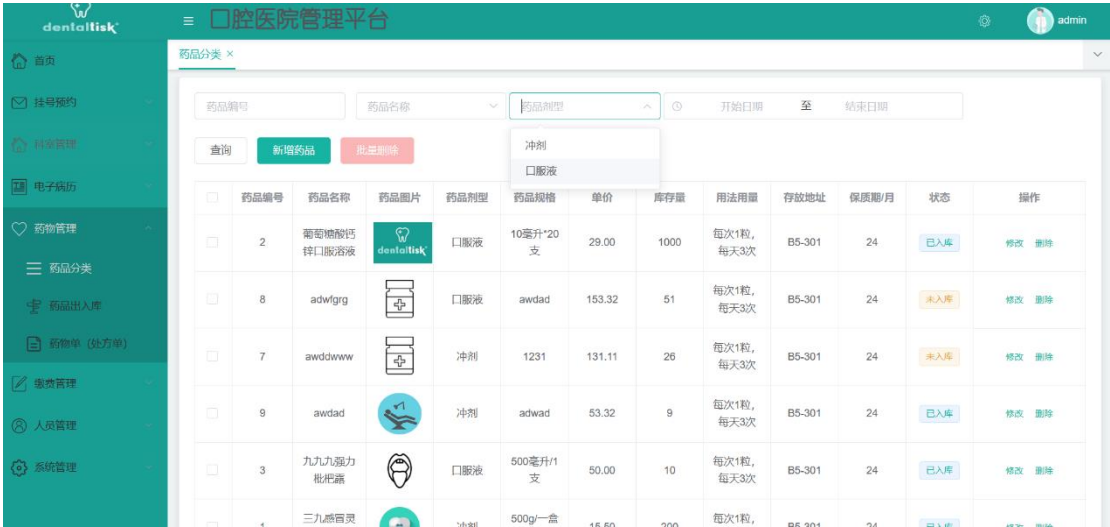


图 5-6 药品管理界面图

药品的出入库，主要对药品库存量的增减操作，如图 5-7 所示。

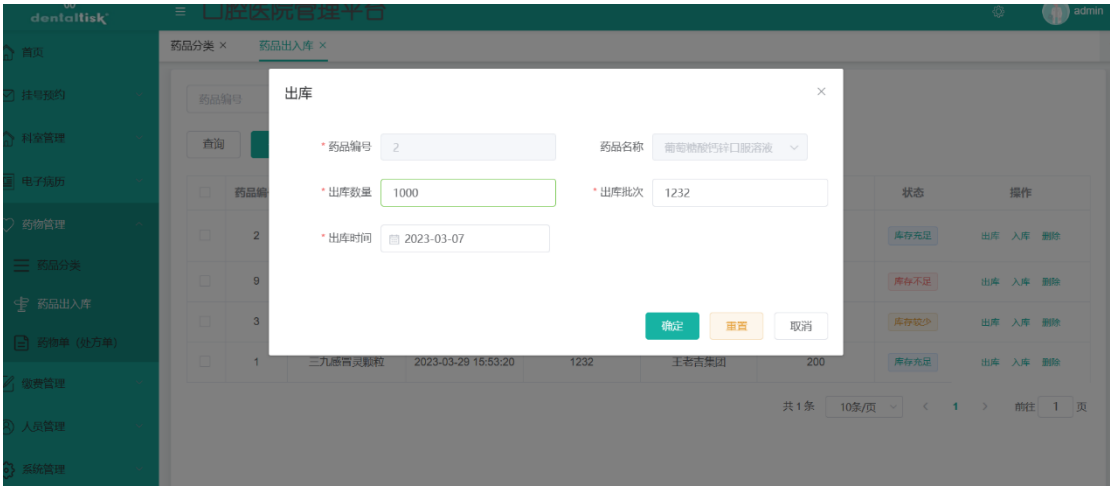


图 5-7 出库操作界面图

药物单管理主要是财务员给对应的药物单进行计价操作并生成缴费订单。如图 5-8 所示。



图 5-8 药物单计价操作界面图

5.1.5 缴费订单模块

缴费订单模块是整个系统业务流程的最终生成数据的模块，该模块主要是展现订单信息，不可进行修改。以下为该业务流程图和医生、患者、财务人员、药物管理员的用例图，如图 5-9 系统业务流程图和图 5-10 所示。

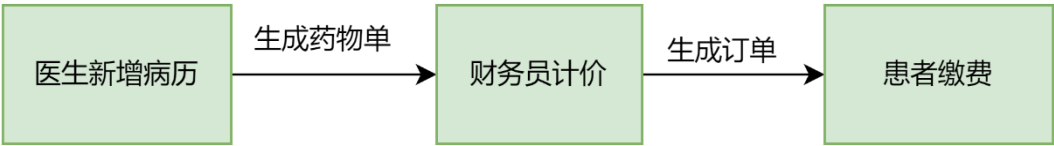


图 5-9 系统业务流程图

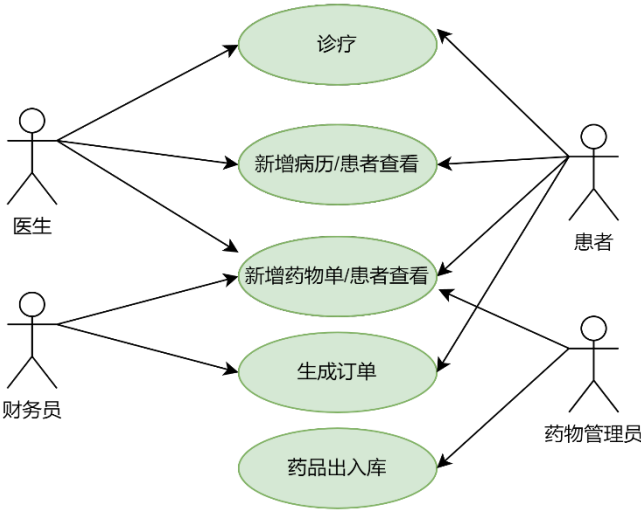


图 5-10 后台医护工作人员业务用例图

订单管理时最后生成的模块信息，不可对其进行修改操作，只能查看详细。前端实现如图 5-11 所示。

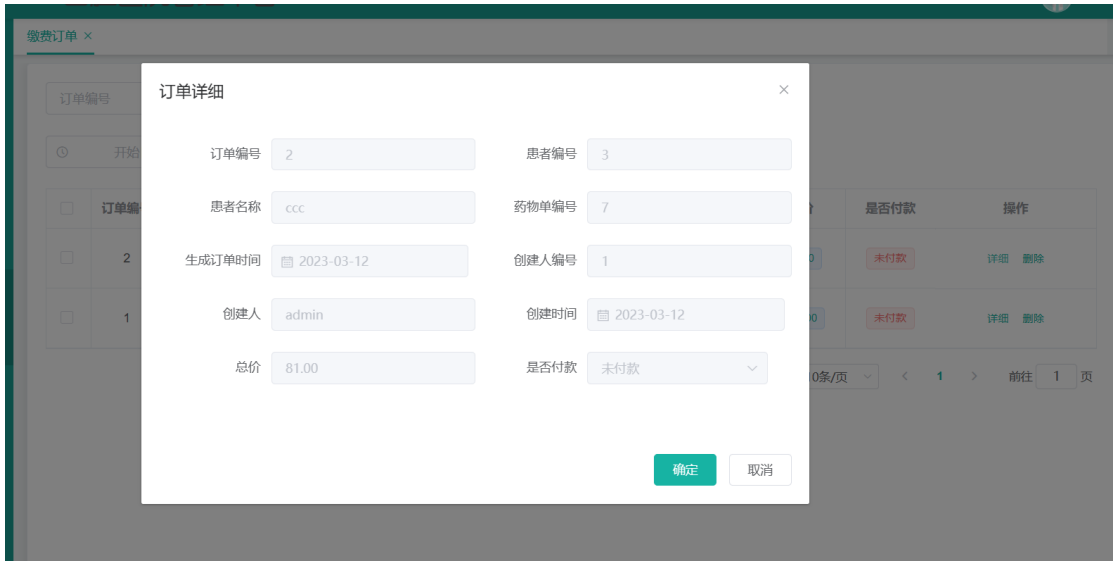


图 5-11 订单详细界面图

5.1.6 人员管理模块

人员管理模块包括了患者和医院所有工作人员的管理，其中医院的只有管理员才有权限对人员管理模块进行 CRUD 操作。医护人员可以在这里查询有国就诊记录的患者信息，还能查询患者的历史病历以及用药情况。方便医护人员对患者病情的把控，作进一步的诊疗。后端使用嵌套 stream 流响应数据。关键代码实现如下：

```
eleRecordsEntityList.stream().map((item)->{ //使用 stream 流的方式
...
    drugPreEntityList.stream().map((j)->{//内嵌 stream 流
...
        return j;//返回实体

    }).collect(Collectors.toList());//收集实体
//把该列表数据集存入主 stream 流
item.setDrugPreEntityList(drugPreEntityList);
return item;
}).collect(Collectors.toList());//收集实体
```

人员管理模块包括患者管理和医院所有工作人员的信息管理。工作人员普通用户没有权限对其进行修改操作。其中在患者管理不仅可查看患者的基本信息还可以查看患者的历史病历。前端效果如图 5-12 和图 5-13 所示。

图 5-12 患者信息界面图

图 5-13 患者历史病历信息界面图

5.1.7 系统管理模块

系统管理模块包括管理员列表管理、角色管理、菜单管理和系统日志管理。该模块只对超级管理员开放。超级管理员可以新增一个管理员并赋予其角色，而菜单管理则是对每个角色使用的菜单模块进行权限分配。使得系统的权限得以有效管理。其中查询所有菜单列表时需要将 list 进行排序，关键代码实现如下：

```

Collections.sort(menuList); //查询完成 对此 list 直接排序
//new 一个 HashMap 用来存方菜单的 ID
HashMap<Long, SysMenuEntity> menuMap = new HashMap<>(12);
...
for (SysMenuEntity s : menuList) { //循环该列表进行判断
    SysMenuEntity parent = menuMap.get(s.getParentId());
    if (Objects.nonNull(parent)) { //判断该菜单的父节点 ID 是否为空
        s.setParentName(parent.getName()); //若不为空则获取父节点菜单名
    }
}

```

系统管理只有超级管理员才有权限操作，包含了管理员信息的管理、角色管理、菜单管理和系统日志管理。其中管理员列表可对其进行新增管理员，并对其赋予与角色。前端实现如图 5-14 所示。

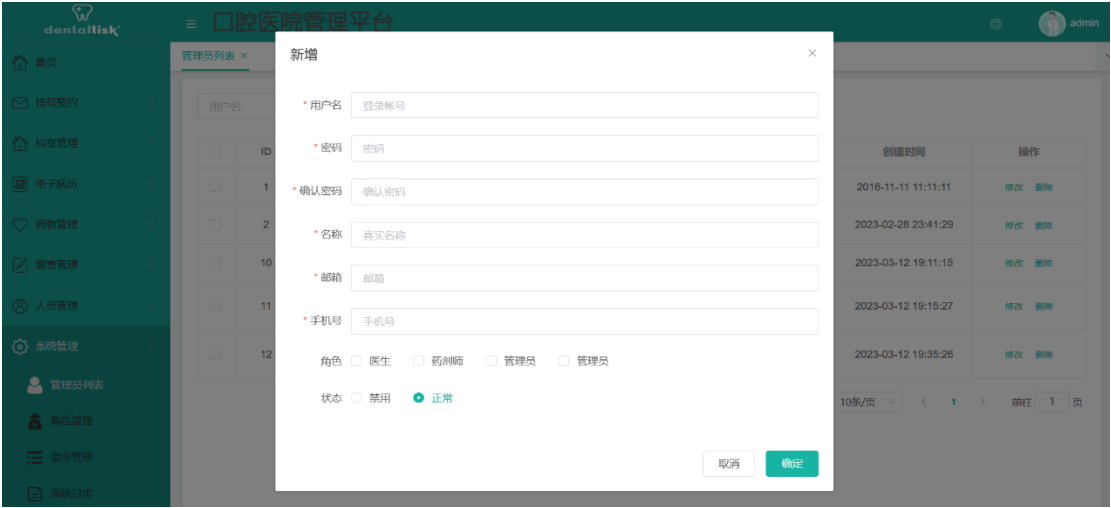


图 5-14 管理员新增人员操作界面图

角色管理对管理员可以对该角色进行菜单的权限分配，前端如图 5-15 医生权限分配所示。

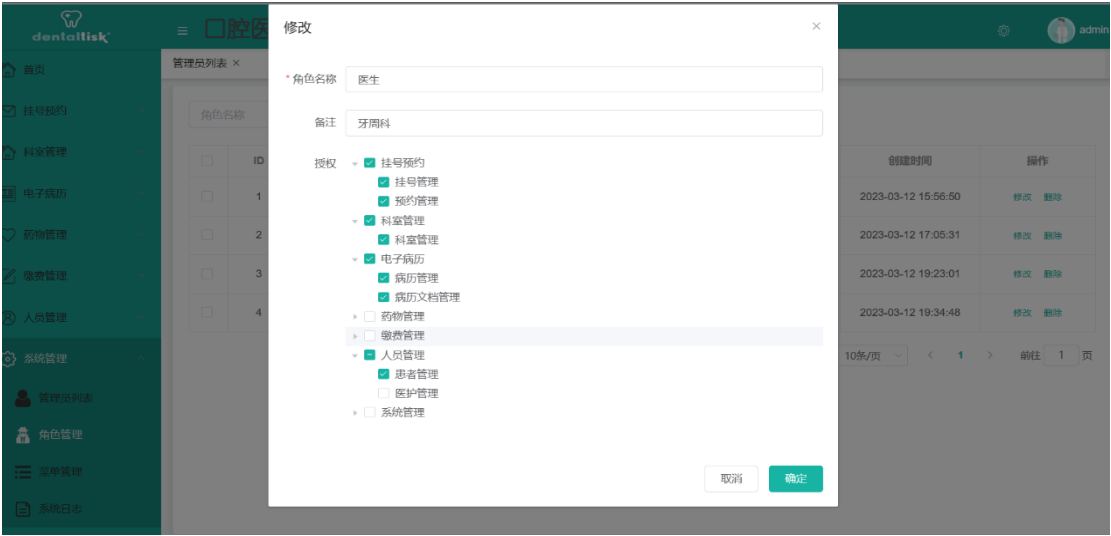


图 5-15 管理员修改角色权限界面图

在菜单管理中管理员可新增模块和修改模块的排序等功能。如图 5-16 所示。

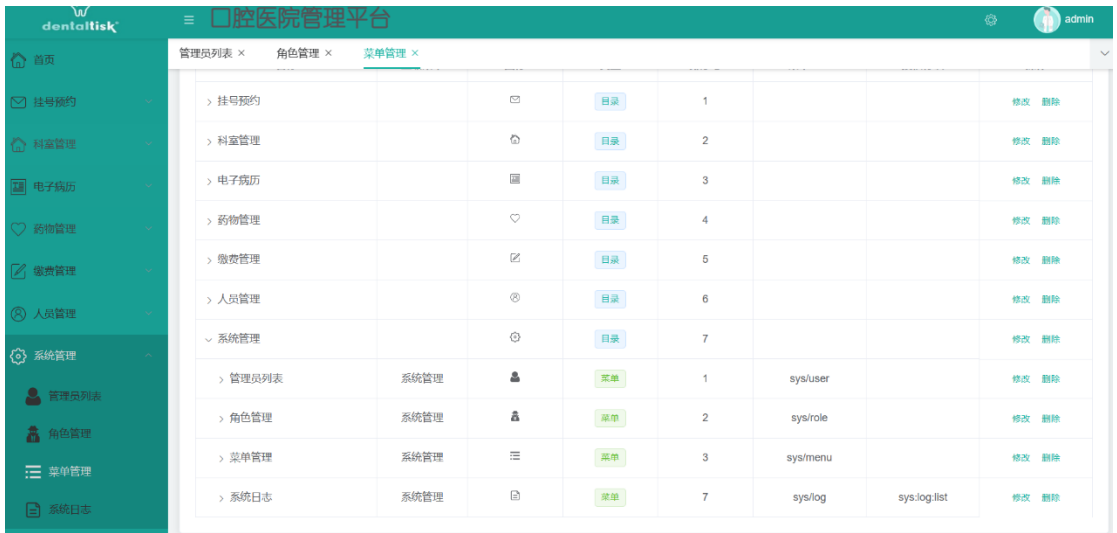


图 5-16 菜单列表界面图

最后是系统日志管理，记录所有对系统的增删改查操作人和操作时间、操作内容等信息，如图 5-17 所示。



图 5-17 系统日志信息界面图

5.3 本章小结

在详细设计中，通过简单的清晰的自然语言、流程图和主要攻克的关键代码方式说明了系统每个功能模块，确保提高编码的质量。

第六章 系统测试

6.1 功能测试

6.1.1 移动端用户业务功能测试用例

本次口腔医院管理系统移动端测试主要使用了黑盒测试，在本次测试案例中，通过设计合理清晰的测试用例来检查口腔医院管理系统的移动端的功能是否达标，是否符合需求，有没有报错等问题。用户以患者为例，其中移动端部分测试如下表 6-1 所示。

表 6-1 移动端用户业务功能测试用例表

测试编号	测试项	预期效果	输入	输出	测试结果
Test1	手机号非空	提示 请输入手机号码	空	系统提示	成功
Test2	验证码	提示 请输入验证码	空	系统提示	成功
Test3	非法手机号	提示 请输入正确手机号	13232	系统提示	成功
Test4	挂号	挂号成功 名字不能为空	挂号 www	挂号成功	成功
Test5	显示挂号信息	展示患者挂号信息	空	系统展示	成功
Test6	预约	预约成功 预约时间、科室、医生 不能为空	预约 2023-4-1 口腔科 ys	预约成功	成功
Test7	显示预约信息	展示患者预约信息	空	系统展示	成功
Test8	查看病历	展示患者所有病历信息	空	系统展示 出所有病 历信息	成功
Test9	退出登录	页面回到登录界面	空	登录界面	成功

6.1.2 PC 端用户业务功能测试用例

口腔医院管理系统后台管理端测试也使用了黑盒测试，在本次测试案例中，通过设计合理清晰的测试用例来检查口腔医院管理系统的后台管理端的功能是否达标，是否符合需求，有没有报错等问题。本次测试用户以医生为例，其中医生后台管理端部分测试如下表 6-2 所示。

表 6-2 PC 端用户业务功能测试用例

测试编号	测试项	预期效果	输入	输出	测试结果
Test1	查看患者挂号信息	展示所有患者的挂号信息	空	系统列表展示	成功
Test2	把未就诊的患者置为已就诊	列表数据显示已就诊	空	系统列表展示	成功
Test3	新增病历	新增患者病历	患者的基本信息、诊断结果和开药信息	新增成功	成功
Test4	修改患者病历	更新患者病历信息	需要更新的患者信息、诊断结果和开药信息	更新成功	成功
Test5	上传病历文档文件	上传与对应病历相关文件	PDF 文件	上传成功	成功
Test6	查看患者信息	系统展示患者基本信息及历史病历信息	空	系统列表展示	成功

6.2 本章小结

本章主要测试了移动端和 PC 端的页面功能实现情况，其基本的功能都已基本完成，并对各个功能模块进行了测试。经过测试，系统基本通过了测试的所有内容。

结 论

从整体上观看本系统，基本达到预期的目的和效果。口腔医院管理系统实现了挂号预约管理、科室管理、电子病历管理、药物管理、订单管理、人员管理和系统管理七大功能模块，基本完成了需求分析中所涉及到功能。本系统目前所用到的技术均是比较流行前沿的技术，前端的设计使用了 Vue 框架和 Element-UI 组件等技术，使开发过程降低了编码难度，后期维护也容易。后端的设计用到了 Spring Boot、MyBatis-Plus 等框架，具有可扩展性和易维护性。项目不足之处是开发时重心放在了后台 PC 端的设计和实现上，虽大致的功能具备了，但系统的界面还是简单粗糙，并且还缺乏有许多细节的功能。如果往后的用户需求对界面有进一步的要求，当继续不断改进和完善。

如今系统已经基本完成，往后可以迭代式开发，不断迭代，完善已有的功能，并根据现实需求，不断更新和增添新的功能，如设计成微服务、分布式、实现系统数据安全性等等。如今的科学技术迅速发展的时代，例如人工智能的出现，它的出现将会迎来更多的人工智能的崛起。未来肯定会有更好的技术，去开发和完善口腔医院管理系统，使得系统更加智能，更便于使用，可以提高医疗机构的数字化竞争力，为医疗机构带来更多的患者资源和商业价值。只有拥有了先进的信息技术，才能够更好地提供医疗服务，满足患者的需求。

参考文献

- [1]陈木生.浅谈中小医院信息管理[J].计算机光盘软件与应用,2014,17(01):98+100
- [2]贺敬.高校医院信息管理系统设计[J].电脑编程技巧与维护,2015(10):84-85.
- [3]梁力中,吴进军.医院信息管理系统构建模式的探讨[J].现代医院,2009,9(05):1-3.
- [4]蒋冰.口腔医院信息系统的设计与实现[D].山东大学,2017.
- [5]刘志国.医院信息管理体系的构建[D].天津师范大学,2004.
- [6]鲁秀明.医院信息管理系统的架构设计[J].计算机时代,2008(07):44-45.
- [7]叶伟峰.现代医院信息系统[J].硅谷,2013,5(01):237.
- [8]白海金.医院 HIS 系统设计与实现[D].北京工业大学,2015.
- [9]徐宁,张磊,李新艳,周培莹.基于计算机技术的医院信息化系统建设[J].电子技术与软件工程,2021(23):176-178.
- [10]张文学,郑丽萍,张利,任建玲.浅谈医院信息管理[J].中国信息界,2004(06):52.
- [11]陈云涛,韩竑,李秀娥.口腔专科医院物资信息化管理体会[J].护士进修杂志,2013,28(05):414-416.
- [12]黄文娟,包宇.试论医院信息管理系统模式的构建[J].中西医结合心血管病电子杂志,2017,5(29):28.
- [13]张志彬.基于计算机技术在医院信息化管理系统设计[J].计算机测量与控制,2021,29(07):185-189.
- [14]姜云翔.现代化医院管理中的计算机应用浅谈[J].办公自动化,2008(24):18-19.
- [15]辛小霞,吴汝明.现代医院信息系统(HIS)的设计[J].现代计算机,2006(11):56-58.
- [16]赵洋.计算机通信技术在信息管理系统中的应用[J].中国新通信,2020,22(01):24.
- [17]朱冬旭.计算机技术在医院信息管理系统中的应用[J].电子世界,2019(11):136-138.
- [18]彭宇.计算机信息管理系统在医院管理工作中的应用[J].电子技术与软件工程,2022(15):240-243.
- [19]曾凡启.计算机网络技术在医院信息管理中的应用[J].无线互联科技,2021,18(22):98-99.
- [20]陈兆明.计算机网络化在医院管理信息系统中的作用[J].吉林医学,2005(10):1090.

致谢

在我完成本次毕业设计的过程中，我收获了许多帮助和支持。在此，我想要表达我的感激之情，感谢所有曾经给予过我帮助的人。

首先，我要感谢我的导师吕绪洋讲师和我的指导小组成员们，他们为我提供了充分的指导和帮助，包括解决研究过程中的问题等。他们耐心细致的教诲和指导，让我在毕业设计的过程中茁壮成长，我将永远铭记他们的教诲和恩情。

其次，我要感谢我的同学们、朋友们和家人们，他们在我长达数月的毕业设计过程中一直给予我无尽的鼓励和支持，他们的陪伴和关注让我始终保持着斗志和信心。

此外，我要感谢广州城市理工学院，在完成毕业设计的过程中，为我们提供了充足的实验设备和相关资源，让我们的研究有了坚实的基础。

最后，我要感谢参与了本次研究的调查协助研究的人，没有他们的支持和配合，我们的调研无法顺利完成。

此外，我还要感谢我的家人、亲戚和朋友们，在我学习和生活中始终给我无微不至的关爱、帮助和支持，他们是我不断进步和学有所成的动力源泉。

在此，再次对所有帮助过我的人表达感激之情，我将不断努力进取，不负个人和母校的期望。