

毕 业 论 文

题 目 [基于Springboot+Vue的智慧](javascript:showViewModel(0,'1449682527517552641'))

[医疗系统的设计与实现](javascript:showViewModel(0,'1449682527517552641'))

学 院 计算机科学与软件学院

专 业 软件工程

年 级 2018级

学 号 201824113327

学生姓名 郑隆杰

指导教师 李芳、纪孟延（企业）

完成时间 2022 年 4 月

肇庆学院教务处制

**学术诚信声明**

本人所呈交的毕业论文，是在指导教师的指导下独立完成。研究工作所取得的成果、数据、图片资料均真实可靠。除文中已注明引用的内容外，不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究做出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名（手写）： 日期： 年 月 日

**目 录**

[摘要与关键词 1](#_Toc18116)

[1 绪论 1](#_Toc32590)

[1.1 研究目的与意义 1](#_Toc8254)

[1.2 国内外研究现状 1](#_Toc8246)

[1.3 研究目标 2](#_Toc28841)

[1.4 开发思路 2](#_Toc25642)

[2 开发技术和工具 3](#_Toc6474)

[2.1 开发技术 3](#_Toc20595)

[2.1.1 B/S模式 3](#_Toc22288)

[2.1.2 SpringBoot后端开发 3](#_Toc28642)

[2.1.3 Vue+ElenmentUi前端开发 3](#_Toc19615)

[2.1.4 Mybatis框架 3](#_Toc31038)

[2.2 开发语言 3](#_Toc16783)

[2.3 开发工具 4](#_Toc31232)

[3 需求分析 4](#_Toc26799)

[3.1 可行性需求分析 4](#_Toc12306)

[3.1.1 经济可行性 4](#_Toc19786)

[3.1.2 社会可行性 4](#_Toc21103)

[3.1.3 技术可行性 5](#_Toc18375)

[3.2 功能性需求分析 5](#_Toc18388)

[3.2.1 注册登录 5](#_Toc7198)

[3.2.2 系统首页 5](#_Toc992)

[3.2.3 医护管理 5](#_Toc14567)

[3.2.4 患者管理 5](#_Toc18009)

[3.2.5 药品管理 5](#_Toc1055)

[3.2.6 病历管理 5](#_Toc5485)

[3.2.7 就诊会话 6](#_Toc19154)

[3.2.8 错误处理 6](#_Toc21289)

[3.3 非功能性需求分析 6](#_Toc21317)

[3.3.1 安全性 6](#_Toc3964)

[3.3.2 易用性 6](#_Toc19809)

[3.3.3 可维护性与可扩展性 6](#_Toc898)

[4 系统设计 6](#_Toc31586)

[4.1 系统总体结构 6](#_Toc18407)

[4.2 系统详细设计 7](#_Toc32028)

[4.2.1 注册登录 7](#_Toc10144)

[4.2.2 系统首页 7](#_Toc14935)

[4.2.3 医护管理 7](#_Toc14682)

[4.2.4 患者管理 8](#_Toc3339)

[4.2.5 药品管理 8](#_Toc32060)

[4.2.6 病历管理 8](#_Toc4744)

[4.2.7 就诊会话 8](#_Toc27132)

[4.2.8 错误处理 8](#_Toc714)

[4.3 数据库设计 8](#_Toc27475)

[4.3.1 数据库E-R图 8](#_Toc10125)

[4.3.2 数据库表设计 10](#_Toc9553)

[5 系统实现 12](#_Toc6364)

[5.1 系统开发环境 12](#_Toc11048)

[5.1.1 硬件环境 12](#_Toc21291)

[5.1.2 软件环境 12](#_Toc3423)

[5.2 系统登录与注册 13](#_Toc6977)

[5.3 系统首页 16](#_Toc652)

[5.3.1 患者首页 16](#_Toc1666)

[5.3.2 医生首页 17](#_Toc18191)

[5.3.3 管理员首页 18](#_Toc29259)

[5.4 后台管理 19](#_Toc29533)

[5.4.1 医护管理 19](#_Toc19335)

[5.4.2 患者管理 21](#_Toc17879)

[5.4.3 药品管理 23](#_Toc20397)

[5.4.4 病历管理 24](#_Toc1101)

[5.4.5 就诊会话 25](#_Toc31413)

[6 系统测试 27](#_Toc31382)

[6.1 登录注册功能测试 27](#_Toc12757)

[6.2 后端管理功能测试 27](#_Toc1764)

[6.3 就诊会话功能测试 28](#_Toc30976)

[7 结论 28](#_Toc27071)

[参考文献 28](#_Toc12668)

[Abstract and Keywords 30](#_Toc32457)

[致 谢 31](#_Toc4619)

[基于Springboot+Vue的智慧医疗系统的设计与实现](javascript:showViewModel(0,'1449682527517552641'))

摘 要: 我国是一个发展不均衡的国家，对于地处偏远欠发达地区的人们而言，医疗资源相对匮乏，基于此，本论文设计并实现了基于Spingboot后端技术，以MySQL作为数据库，以Vue作为前端框架，以ElementUi技术完成了数据可视化，设计并实现了智慧医疗系统。该医疗系统借助互联网实现了患者与医生进行远程就诊，医疗部门实时存储掌握患者及医疗就诊信息，实现对医疗信息的便捷管理，能够帮助最大程度利用医疗资源，初步实现了“智慧医疗”的理念。

关键词:智慧医疗; 前后端分离; 数据库

# 1 绪论

## 1.1 研究目的与意义

随着社会经济的发展，人民的生活水平不断提高，人们更加注重自己的身体健康。然而，对于医疗资源的渴望，不仅是欠发达地区老百姓的愿望，也是每个人对于自己身体健康的最大诉求，也是建立在追求实现伟大的中国梦的又一历史前提。因此，开发一套更为便捷，更为智能化的医疗系统非常有利于促进社会的和谐稳定，也满足了人民群众的根本利益所在。

近些年来，随着医疗技术的广泛应用，医院无纸化服务的水平不断加强，不过，由于侧重点的不同，患者仍需到达医院进行就诊，而无法实现真正的远程会诊。智慧医疗就是当前依托信息技术而言的，试图去摆脱传统的排队额方式，直接与医生进行交流，通过线上的平台技术，将自己的病情直达医生，从而减少了不必要的排队，挂号，从而实现高效率的患者就诊。就目前而言，每个人都已经拥有自己的移动终端，在该医疗系统中，通过登录相应的平台，从而与医生实现会话就诊，构建医生与患者的信息管理，将处方信息，病历信息等信息管理做到统一存储，以便于实现就诊的愿望。

## 1.2 国内外研究现状

近十年以来，国内医疗的发展较为迅速，许多省、市级医院已经逐步建立起智慧医疗体系，实施了线上预约，线上接收检查报告等基础操作，实现患者与医疗人员、医疗结构之间的信息共享机制。然而，我国是一个资源分布不均衡的国家，主要表现为经济发达的东部沿海地区与中西部内陆地区之间的差距，以及中大型城市医疗服务体系与乡村基层医疗服务能力之间的差距。使得我国的智慧医疗体系的普及存在着较大的困难，因此完善智慧医疗体系，加速互联网技术与医疗行业的深度融合，构建智慧医疗平台，已成为亟待解决的难题[1]。

目前，国外随着医疗技术水平的不断提升，以互联网技术为前提的远程处理与远程服务已经较为成熟，医生能通过互联网技术，进而逐步改变在过去仅通过面对面解决的望闻问切的方式，从而极大的提高了医疗资源的配置与资源浪费，减少了不必要的就诊手续，从而真正实现了智慧医疗。近年来，西方传统医疗行业也已经逐步转型，进一步促进智慧医疗的发展。

总而言之，智慧医疗体系的前景已日渐明朗，随着社会的发展以及信息化技术的日渐普及，智慧医疗已成为不可阻挡的潮流，也是社会健康体系的迫切需求。人民对于健康的渴求，也是促进智慧医疗发展的重要原因，因此，开发出完善的智慧医疗平台，已成为亟待解决的问题。

## 1.3 研究目标

基于上述分析可以看出，适应当前情况下的医疗体系，首先应提出一种基于“智慧医疗”理念的解决方案。研究关于如何将就诊从线下转化为线上，将无纸化的服务彻底实现的就诊平台。该平台针对医疗体系的短板，完成了患者从就诊到个人病历信息的填写，通过可视化技术，将数据存储与显示的功能进行设计与实现，分别为不同的用户采用不同的端口，完成对系统的使用。最后基于Springboot、vue框架技术[2]，采用Java编程语言搭建一套运行相对稳定，且基本实现上述功能的智慧医疗系统。

## 1.4 开发思路

智慧医疗系统以SpringBoot+Vue框架，开发功能模块与接口的编写，实现对整个系统的功能管理，包括人员信息的管理实现，不同用户身份的端口处理，以及整个系统的运行。后端以Vue框架为基础，结合脚手架的使用，根据该框架中提供的element-ui来开发页面，实现数据的可视化编程。并且利用Mybatis-plus-generator和velocity结合自定义生成模板实现对应数据库表基本增删查改的controller、service、serviceImpl、mapper、entity的生成，避免的代码的重复编写。该系统采用MySQL作为数据库存储系统信息，采用Druid数据源实现对数据库信息的访问。

# 2 开发技术和工具

## 2.1 开发技术

### 2.1.1 B/S模式

Browser/Server开发模式使用的是浏览器和服务器的模式[3]，是在互联网发展的进程中，随着Internet技术的流行，对于以前经常使用的Client/Server模式的一种优化。这种浏览器与服务器的模式，通过对项目配置的端口信息，即可通过地址访问相应的服务器，在最大程度上减少了用户端的电脑负荷。以目前的情况来看，Browser/Server的最大好处方便了用户的使用，基本上不需要安装其他任何东西就可以进行远程访问与使用，一般只需要针对于对系统的身份认证，即可实现。开发者需要尽可能多的考虑到用户使用的交互性与实时更新需求。Browser/Server结构主要利用了不断成熟的Web浏览器技术，加之框架开发的搭建，从而更好的推进了该模式的广泛应用。

### 2.1.2 SpringBoot后端开发

SpringBoot是在Spring4.0的基础之上进行重新改良的版本，在这种改良的前提之下，不仅将Spring的配置信息进行了最大程度的简单与优化，而且在进行搭建Spring环境时，还做到了简单与优化配置的过程，可以对用户创建项目得到了最便捷的使用方式。SpringBoot创建的初始化程序中还包含着两个重要的策略：开箱即用和约定优于配置[4]。

### 2.1.3 Vue+ElenmentUi前端开发

Vue是用于创建用户使用界面的框架，这个框架直接负责用户的使用界面的设计，关乎用户的交互性需求与整体体验效果，负责三层架构中的视图层。Vue中设计了很多组件，很容易与其他库或者项目使用，不仅做到了数据同步，而且大大增强了页面的渲染能力，其组件化的形式，增强了页面的交互能力，极大的提高了代码的重用率。

### 2.1.4 Mybatis框架

Mybatis框架时针对数据库而存在的框架，主要负责对数据库的数据的处理操作。在创建项目并引入相应的jar包以及依赖之后，便可实现对数据的基本处理等操作，在底层源码的支撑下，可以通过源码分析等函数，直接处理数据库，完善后端的处理功能，本系统正是Mybatis框架实现对数据库的管理操作。

## 2.2 开发语言

Java是一门面向对象的编程语言，具有功能是强大和简单易用两个特征[5]。开发者更多的只是需要基于其面向对象的优点，从而基于编程语言的开发过程进行项目开发。Java语言经历过长时间的发展，目前已经被许多公司接受并为其开发出成熟的软件，可用于编译并且运行Java程序，最终实现了所开发系统的运行效率与运行目的。就其稳健性而言，消除了某些编程错误，使得它写可靠软件相当容易。

## 2.3 开发工具

数据库：MySql5.5.62

数据库管理工具：Navicat Premium12

项目管理：Spring Initializr

开发工具：JDK1.8

编程语言：Java

后端编程工具：IntelliJ IDEA 2019.3.3

前端编程工具：Visual Studio Code

# 3 需求分析

## 3.1 可行性需求分析

### 3.1.1 经济可行性

目前，进行可行性的目标，就是要在控制成本尽可能低的前提下为系统的额开发带来最佳的经济效益。在前后端分离技术日益成熟的今天，以SpringBoot作为系统核心的后端开发框架，在这种改良的前提之下，不仅将Spring的配置信息进行了最大程度的简单与优化，而且在进行搭建Spring环境时，还做到了简单与优化配置的过程，可以对用户创建项目得到了最便捷的使用方式。SpringBoot创建的初始化程序中还包含着两个重要的策略：开箱即用和约定优于配置[6]，而不要XML配置文件，尤其是做到了开放源码，并且支持多种引擎开发，因此不需要支付额外的费用。

### 3.1.2 社会可行性

就目前的社会环境变化以及人口的数量变化，在本就不充裕的医疗资源的背景下，医疗系统的开发，是医疗体系发展不可或缺的一个大的方向，也是解决民生问题的最大导向。人民对于医疗技术的渴求，对于更智能化，更便捷的医疗技术平台，使其可以更好的了解自己的身体状况，更好的呵护自身健康。目前互联网技术的发展，大范围的网络覆盖，实现了信息交流的可能，使得该系统的使用前提得到了保证，不仅能获得医疗资源，对于自身的疾病隐患，也能够在提前获取医疗资源并进行一定程度上的自我判断。

### 3.1.3 技术可行性

基于该系统开发的智慧医疗平台，以目前相对成熟的Java编程语言开发，使得系统实现成为了可能。Java的强大功能以及简单易用的两大特征，在经历了几十年的运用，已经被设计成写高可靠和文件软件的首选，在这种前提下，极大的简化了该系统的开发过程，并且为该系统的实现目标做好了很好的铺垫。目前而言，基于在校学习经历以及实习阶段掌握的相关技术，对轻量型框架的使用已掌握基本技能，因此，目前的技术水平已具备开发智慧医疗系统。

## 3.2 功能性需求分析

### 3.2.1 注册登录

注册登录模块显示患者医生两种不同身份的信息注册及登录，注册功能中包括了对用户输入信息的一般限制以及规范；当输入信息错误时需进行提示。

### 3.2.2 系统首页

系统首页显示登录者身份的头像信息和相关信息，在头像下方显示跳转到设置界面或者退出；在系统首页下方展示对应的医生列表信息或者患者信息，在导航栏要显示对医生或者科室的信息查询；后端管理者界面显示导航栏信息，一些统计图表以及系统的相关信息。

### 3.2.3 医护管理

医护管理模块显示对医护人员信息的增加，删除，查询，修改等功能，在删除模块包括了对医护信息进行单项删除与多项删除，查询模块包括了对医护信息进行模糊查询，还包括了对某一位医护人员接诊的患者信息的管理。

### 3.2.4 患者管理

患者管理模块显示对患者信息的增加，删除，查询，修改等功能，在删除模块包括了对患者信息进行单项删除与多项删除，在查询模块包括了对患者信息进行模糊查询。

### 3.2.5 药品管理

药品管理模块显示对药品信息的增加，删除，查询，修改等功能，在删除模块包括了对药品信息进行单项删除与多项删除，在查询模块包括了对药品信息进行模糊查询。

### 3.2.6 病历管理

病历管理模块显示对病历信息的增加，删除，查询，修改等功能，在删除模块包括了对病历信息进行单项删除与多项删除，在查询模块包括了对病历信息进行模糊查询。

### 3.2.7 就诊会话

就诊会话功能需展示会话者之间的所有会话记录，并对会话信息的发送做到实时响应与页面刷新功能，不用患者与医生之间的会话记录做到屏蔽与隐私保护。

### 3.2.8 错误处理

使用该系统时，需进行系统的错误信息反馈，对添加、删除、发送信息等操作时，需提示操作的成功与否等。

## 3.3 非功能性需求分析

### 3.3.1 安全性

系统的数据在获取之后，需存放于数据库中，且应对数据库进行设置密码保护；访问数据库时需要获取权限，如系统后端连接数据库时需配置账号密码等信息；根据用户身份的不同，授予相应的医生和患者具有的注册、登录以及会话的权限，根据权限控制访问数据。

### 3.3.2 易用性

本系统基于患者与医生的用户对象，需保证进行使用时的操作简易性，能够在界面直观的展示系统的使用，防止用户产生误解从而无法实现易操作性，界面应该尽可能做到美观，简洁，布局合理。

### 3.3.3 可维护性与可扩展性

系统功能应做到模块化，当业务流程产生并更后，能够进行灵活配置，尽可能减少项目后期进行开发的重复性工作；可复用的组件应该加以分类，分开封装以便于进行调用。

# 4 系统设计

## 4.1 系统总体结构

基于上述分析，系统设计主要分为身份注册登录，医护管理，患者管理，病历管理，系统首页管理，就诊会话管理。具体结构大致如下图4-1所示。

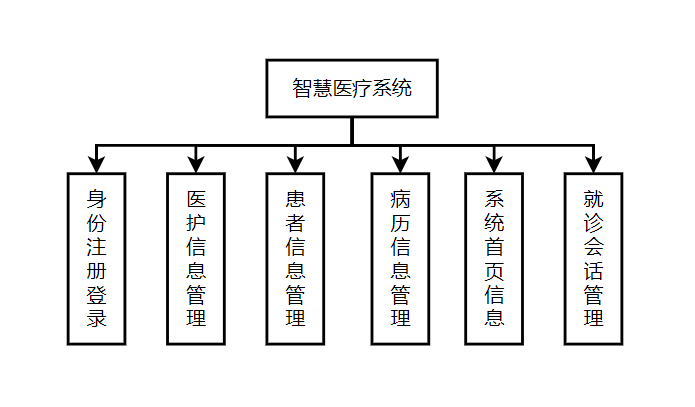


图4-1系统结构图

## 4.2 系统详细设计

### 4.2.1 注册登录

系统的登录用户为医生和患者，具有不同身份的人员信息需要配置不同的登录端口，并且为医生与患者提供注册功能，登录信息需与数据库信息进行匹配，在未注册信息时，显示用户登录失败，在登录成功后需进行相应页面的跳转。

### 4.2.2 系统首页

(1)患者首页

患者登录系统之后，展示该患者的姓名，头像等基本信息，医院的医生列表，对医院的医生和科室分别进行查询，单击头像下拉框可以选择退出登录；在医生列表中，可以选择自己所要就诊的医生，进行会话就诊。

(2)医生首页

医生登录系统之后，显示该医生接诊下的患者的列表信息；选择不同的患者后，界面显示该患者的基本信息，之后医生开始与患者进行就诊会话，同时保证不同患者之间的会话信息屏蔽。

(3)管理平台首页

管理员登录管理平台后，系统页面展示管理员的登录信息，头像等，在头像右侧设置下拉框，可以进行系统设置与退出系统登录，最右侧设置全屏功能，可将整个系统界面全屏；在左侧导航栏中，分别显示医护管理、患者管理、药品管理、病历管理等，并能够跳转至相应的界面进行管理。

### 4.2.3 医护管理

医护管理界面展示所有医护人员的信息，其中，包括了对医护人员信息的批量删除，模糊查询，添加、修改、单项删除等功能；在展示医护人员信息时，需做到分页处理；在每条医护信息中，显示对应的就诊人管理，即可以做到对每一位医生对应的患者进行管理，并显示出对应的患者列表；管理界面可以实现对每一位医护人员的照片进行更新与上传，并且能够加载到项目中保存。

### 4.2.4 患者管理

患者管理界面展示所有患者信息，其中，包括了对患者人员信息的批量删除，模糊查询，添加、修改、单项删除等功能；在展示患者人员信息时，需做到分页处理；管理界面可以实现对每一位患者的照片进行更新与上传，并且能够加载到项目中保存。

### 4.2.5 药品管理

药品管理界面展示所有药品信息，其中，包括了对药品信息的批量删除，模糊查询，添加、修改、单项删除等功能；在展示药品信息时，需做到分页处理。

### 4.2.6 病历管理

病历管理界面展示所有病历信息，其中，包括了对病历信息的批量删除，模糊查询，添加、修改、单项删除等功能；在展示病历信息时，需做到分页处理。

### 4.2.7 就诊会话

就诊界面功能为医生或患者进行会话时，能够实时发送消息，在消息发送后，页面提示消息状态，之后在会话窗口下方展示该患者与医生的所有会话记录；并且并且保证会话信息只在该患者与该接诊医生是可见的，对其他患者或者医生屏蔽。

### 4.2.8 错误处理

当对信息的操作出现错误，如添加失败，删除失败，消息发送失败等情况，系统需提示相应的错误信息，并阻止操作的进一步进行。

## 4.3 数据库设计

### 4.3.1 数据库E-R图

(1)患者E-R图

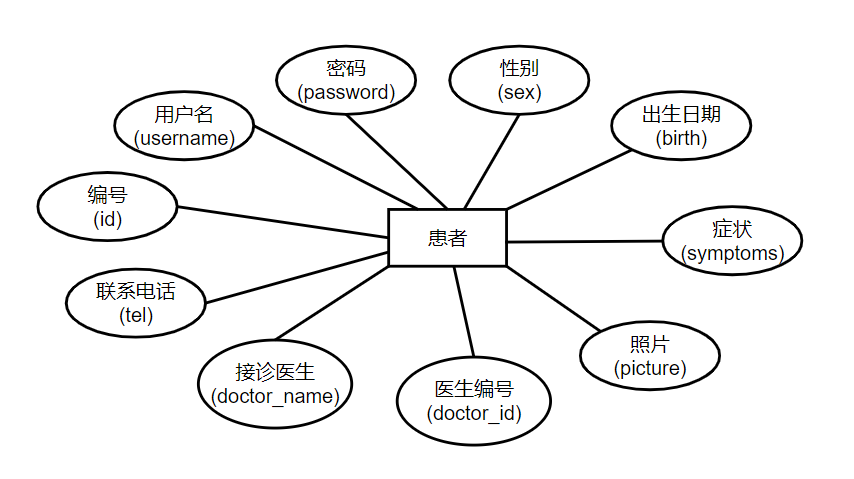


图4-2患者E-R图

(2)医生E-R图

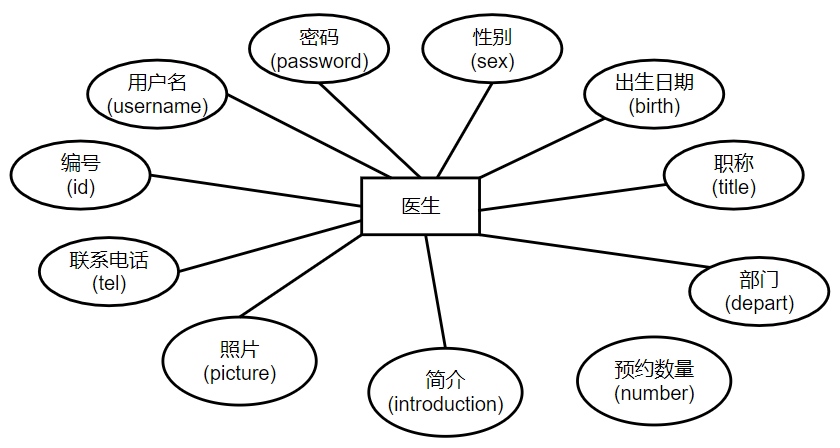


图4-3医生E-R图

(3)病历E-R图

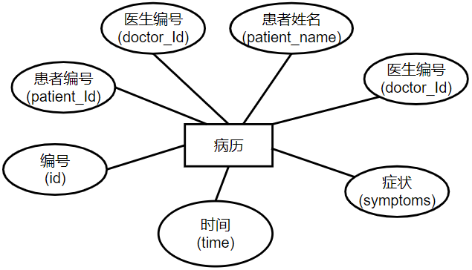


图4-4病历E-R图

(4)就诊会话E-R图

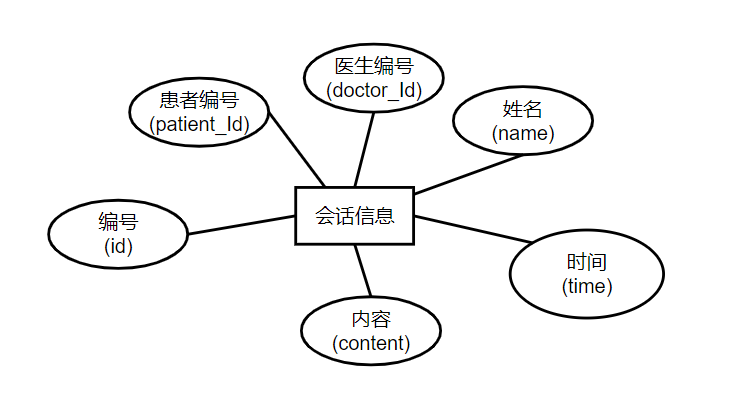


图4-5会话信息E-R图

(5)药品E-R图

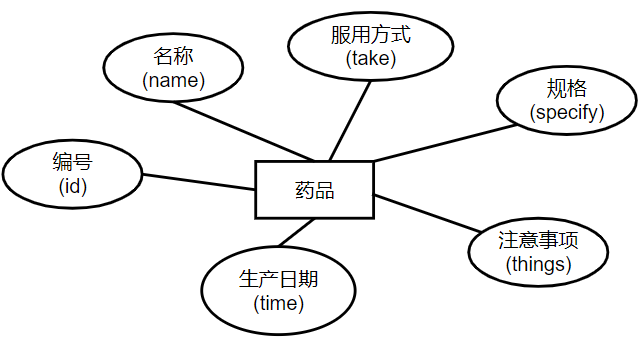


图4-6药品E-R图

(6)关系E-R图

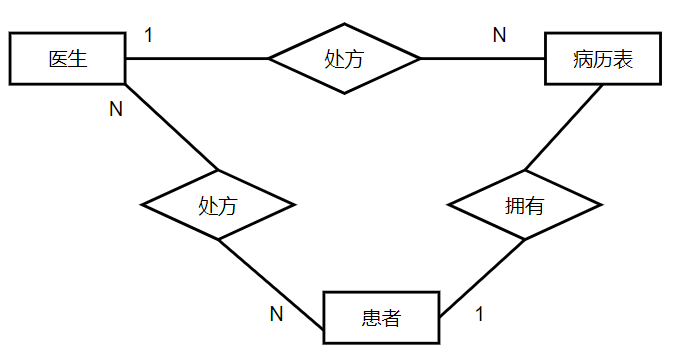


图4-7关系E-R图

### 4.3.2 数据库表设计

（1）医生信息表（medical\_doctor），表结构如表4-1所示。

表描述：记录医生的基本信息。

表4-1 医生信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 小数点 | 允许为null | 主键 | 注释 |
| id | int | 11 | 0 | 否 | 是 | 编号 |
| name | varchar | 11 | 0 | 否 | 否 | 姓名 |
| password | varchar | 100 | 0 | 否 | 否 | 密码 |
| sex | int | 1 | 0 | 否 | 否 | 0-男,1-女 |
| tel | varchar | 11 | 0 | 是 | 否 | 注册手机号 |
| birth | datetime | 0 | 0 | 否 | 否 | 出生日期 |
| depart | varchar | 100 | 0 | 是 | 否 | 部门 |
| introduction | varchar | 100 | 0 | 是 | 否 | 简介 |

（2）患者信息表（t\_test\_authority），表结构如表4-2所示。

表描述：记录患者的基本信息。

表4-2 患者信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 小数点 | 允许为null | 主键 | 注释 |
| id | int | 11 | 0 | 否 | 是 | 编号 |
| name | varchar | 11 | 0 | 否 | 否 | 姓名 |
| password | varchar | 100 | 0 | 否 | 否 | 密码 |
| sex | int | 1 | 0 | 否 | 否 | 0-男,1-女 |
| tel | varchar | 11 | 0 | 否 | 否 | 注册手机号 |
| birth | datetime | 0 | 0 | 否 | 否 | 出生日期 |
| doctor\_name | varchar | 100 | 0 | 是 | 否 | 接诊医生姓名 |
| doctor\_Id | varchar | 100 | 0 | 是 | 否 | 接诊医生编号 |
| picture | varchar | 100 | 0 | 是 | 否 | 照片 |

（3）病历信息表（t\_test\_environment），表结构如表4-3所示。

表描述：记录患者的病历信息。

表4-3 病历信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 小数点 | 允许为null | 主键 | 注释 |
| id | int | 11 | 0 | 否 | 是 | 主键 |
| patient\_id | int | 11 | 0 | 否 | 否 | 患者编号 |
| patient\_name | varchar | 11 | 0 | 否 | 否 | 患者姓名 |
| doctor\_id | int | 11 | 0 | 否 | 否 | 接诊医生编号 |
| doctor\_name | varchar | 11 | 0 | 否 | 否 | 接诊医生姓名 |
| create\_time | datetime | 0 | 0 | 是 | 否 | 创建时间 |
| symptoms | varchar | 255 | 0 | 是 | 否 | 症状 |

（4）会话信息表（t\_test\_environment），表结构如表4-4所示。

表描述：记录患者与医生的会话信息。

表4-4 会话信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 小数点 | 允许为null | 主键 | 注释 |
| id | int | 11 | 0 | 否 | 是 | 主键 |
| patient\_id | int | 11 | 0 | 否 | 否 | 患者编号 |
| doctor\_id | int | 11 | 0 | 否 | 否 | 接诊医生编号 |
| name | varchar | 11 | 0 | 否 | 否 | 发送人名字 |
| create\_time | datetime | 0 | 0 | 是 | 否 | 创建时间 |
| content | varchar | 255 | 0 | 是 | 否 | 会话内容 |
| picture | varchar | 255 | 0 | 是 | 否 | 发送者照片 |

（5）药品信息表（t\_test\_environment），表结构如表4-5所示。

表描述：记录药品信息。

表4-5 药品信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 小数点 | 允许为null | 主键 | 注释 |
| id | int | 11 | 0 | 否 | 是 | 主键 |
| name | varchar | 11 | 0 | 否 | 否 | 药品名称 |
| create\_time | datetime | 0 | 0 | 是 | 否 | 生产日期 |
| take | varchar | 11 | 0 | 否 | 否 | 服用方式 |
| specify | varchar | 11 | 0 | 否 | 否 | 规格 |
| number | int | 11 | 0 | 是 | 否 | 库存数量 |

# 5 系统实现

## 5.1 系统开发环境

### 5.1.1 硬件环境

表5-1 硬件环境

|  |  |
| --- | --- |
| 硬件设备 | 硬件参数 |
| 处理器 | Intel(R) Core (TM) i7-7700HQ CPU @ 2.80GHz |
| 内存 | 8 GB |
| 主硬盘 | Lenovo SSD SL700 M.2 128G ( 128 GB / 固态硬盘 ) |
| 主显卡 | 英特尔 未知(5916h) ( 128 MB ) |
| 显示器 | Intel(R) HD Graphics 630 |
| 网卡 | RTL8168/8111/8112 Gigabit Ethernet Controller |
| 声卡 | Speakers/Headphones |

### 

### 5.1.2 软件环境

表5-2 软件环境

|  |  |
| --- | --- |
| 软件类型 | 软件名称 |
| 数据库 | MySql 5.5.62 |
| 开发工具 | JDK（Java） |
| 项目管理工具 | Maven |
| 编程工具 | IntelliJ IDEA+Visual Studio Code |
| 浏览器 | Google |
| 操作系统 | Windows 10 专业版 64位 |

## 5.2 系统登录与注册

用户输入用户名与密码信息或者进行注册后进行登录，登录成功后才能进行该系统并使用其功能。

登录与注册类相关方法说明如下表5-3、表5-4、表5-5、表5-6所示。

表5-3 医生登录相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| DoctorController | 医生登录控制器，即医生登录的后端接口。 |
| Doctor | 医生实体类，与数据库中医生信息表相互对应。 |
| DoctorService | 医生登录服务类，负责对医生登录相关逻辑的封装。 |

表5-4 医生登录方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| DoctorController. login() | 接收传入的医生的账号密码等信息。检验后返回医生的登录信息后，传递令牌并将令牌存入cookie。 |
| DoctorController. logout() | 清除医生的登录令牌信息，撤销医生的登录信息与状态。 |

表5-5 患者登录相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| PatientController | 患者登录控制器，即患者登录的后端接口。 |
| Patient | 患者实体类，与数据库中患者信息表相互对应。 |
| PatientService | 患者登录服务类，负责对患者登录相关逻辑的封装。 |

表5-6 患者方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| PatientController. login() | 接收传入的患者的账号密码等信息。检验后返回患者的登录信息后，传递令牌并将令牌存入cookie。 |
| Patient Controller. logout() | 清除患者的登录令牌信息，撤销患者的登录信息与状态。 |

表5-7 管理员登录相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| ManagerController | 管理员登录控制器，即管理员登录的后端接口。 |
| Manager | 管理员实体类，与数据库中管理员信息表相互对应。 |
| ManagerService | 管理员登录服务类，负责对管理员登录相关逻辑的封装。 |

表5-8 管理员登录方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| ManagerController. login() | 接收传入的管理员的账号密码等信息。检验后返回管理员的登录信息后，传递令牌并将令牌存入cookie。 |
| ManagerController. logout() | 清除管理员的登录令牌信息，撤销管理员的登录信息与状态。 |

医生和患者实现登录流程如下图5-1所示。

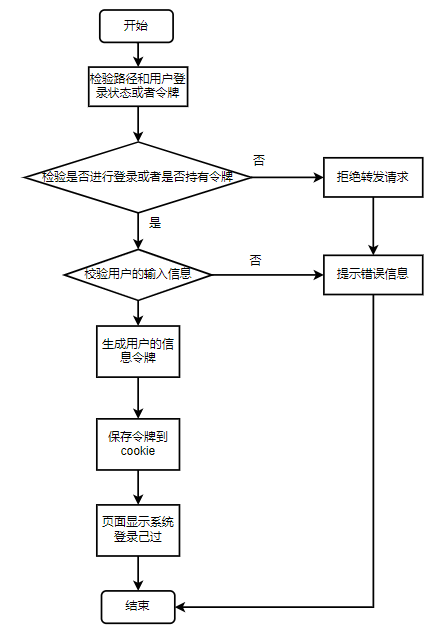


图5-1 登录流程图

智慧医疗系统医生登录界面如图5-2、注册界面如图5-3所示。



图5-2 登录界面图

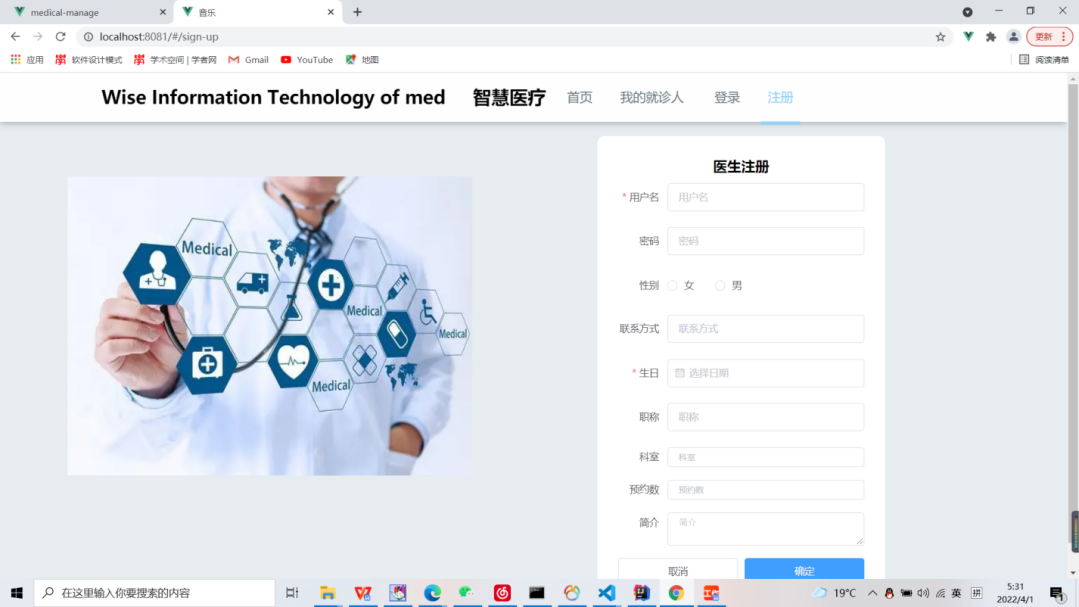


图5-3 注册界面图

智慧医疗系统患者登录界面如图5-4、注册界面如图5-5所示。

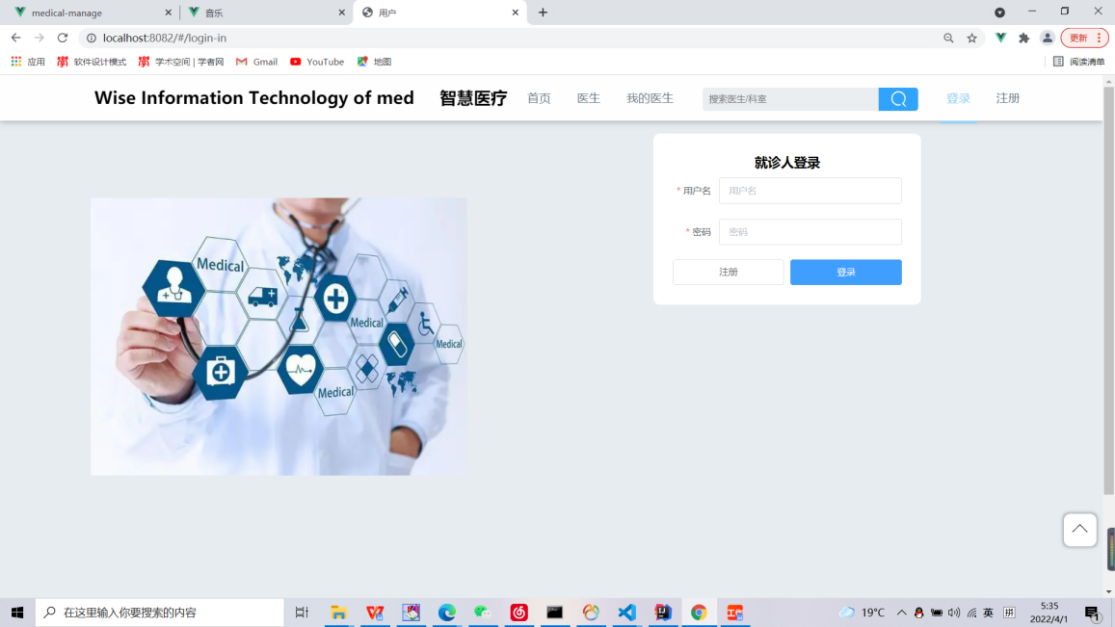


图5-4 注册界面图

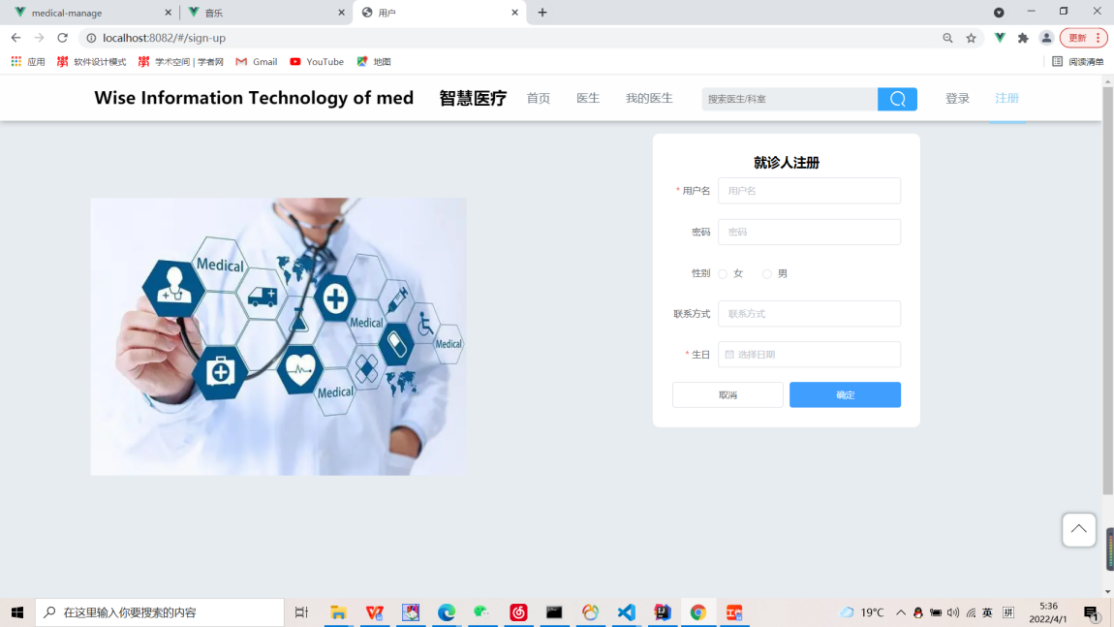


图5-5 登录界面图

智慧医疗系统管理员登录界面如图5-6所示。

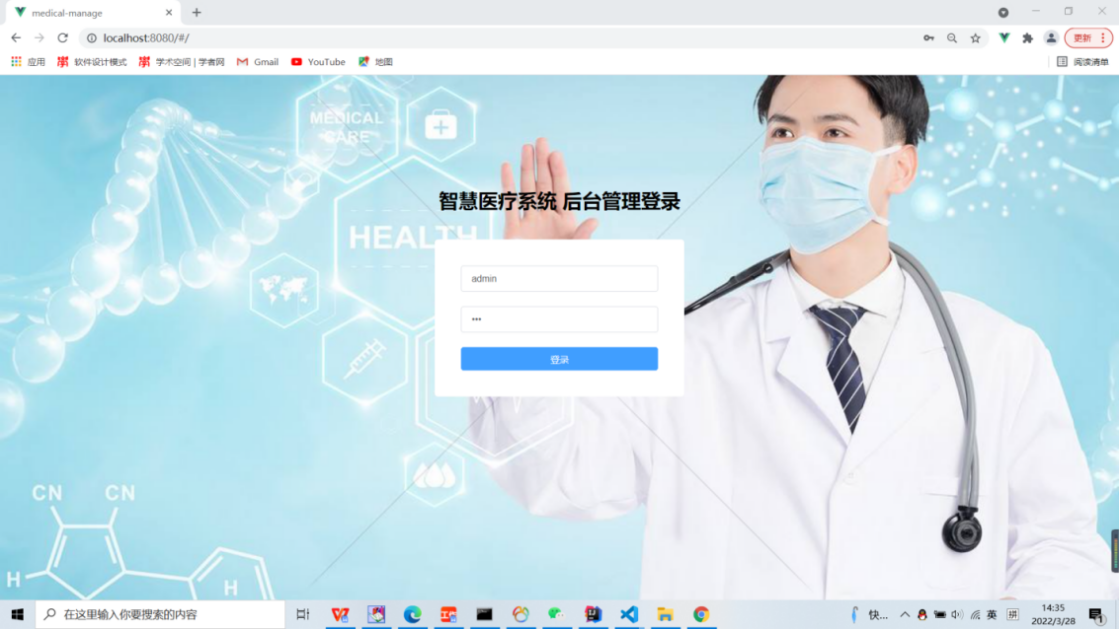


图5-6 登录界面图

## 5.3 系统首页

### 5.3.1 患者首页

患者进入登录界面后，首页信息显示该患者的头像，下方显示所有的医生列表信息，顶部导航栏显示对医生或者医院科室的查询，点击相应的医生可进行会话就诊。

患者首页前端方法说明如表5-9所示。

表5-9 患者首页前端方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| SelectBykey() | 查询所有的医生信息，供患者选择就诊。 |
| SetMessage() | 发送会话信息。 |
| GetAllMessage() | 查询对应医生与患者之间的信息。 |

患者首页、就诊会话界面分别如下图5-7、图5-8所示。

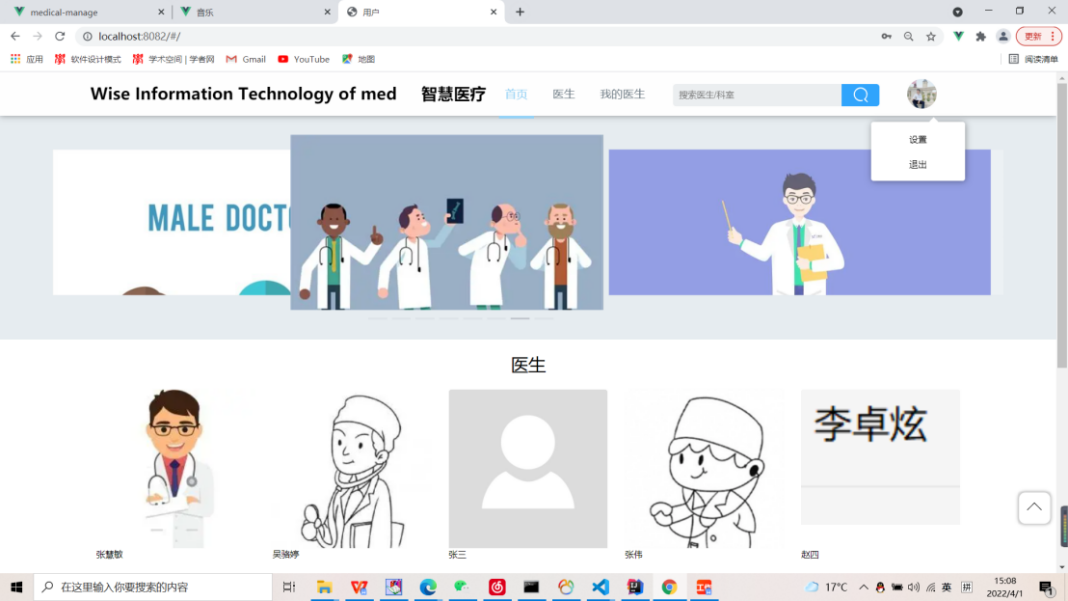


图5-7 患者首页图

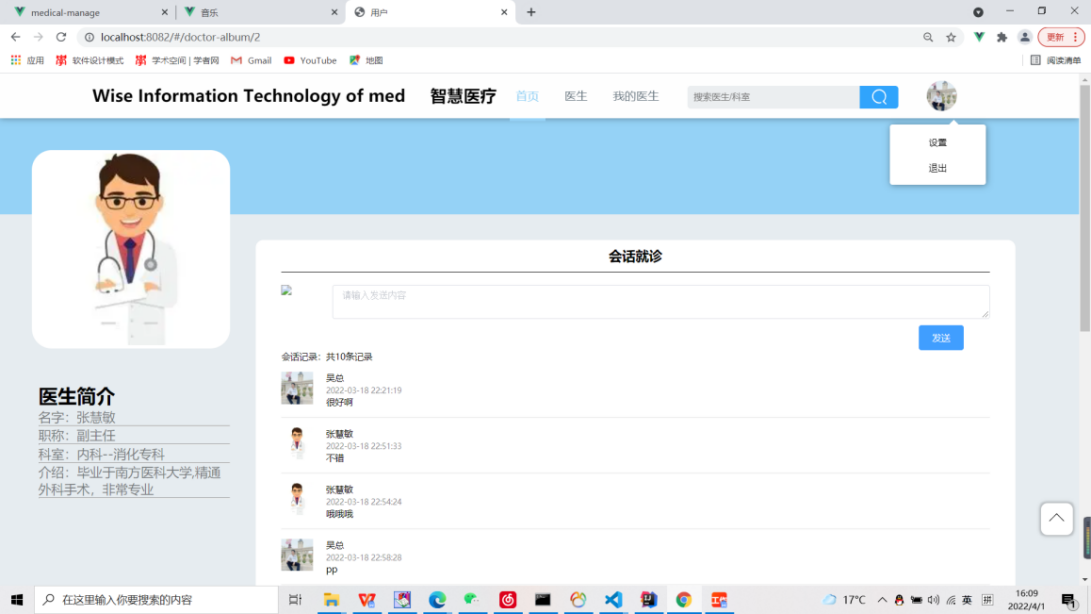


图5-8 患者会话就诊图

### 5.3.2 医生首页

医生登录进入首页后，将显示该医生对应的所有患者的信息列表，医生单击列表中的患者信息，即可开始就诊会话，就诊结束后，医生可为患者填写病历信息。

医生首页前端方法说明如表5-10所示。

表5-10 医生首页前端方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| SelectBykey() | 查询所有的患者信息，供医生就诊。 |
| SetMessage() | 发送会话信息。 |
| GetAllMessage() | 查询对应医生与患者之间的信息。 |

医生首页、就诊会话界面分别如下图5-9、图5-10所示。

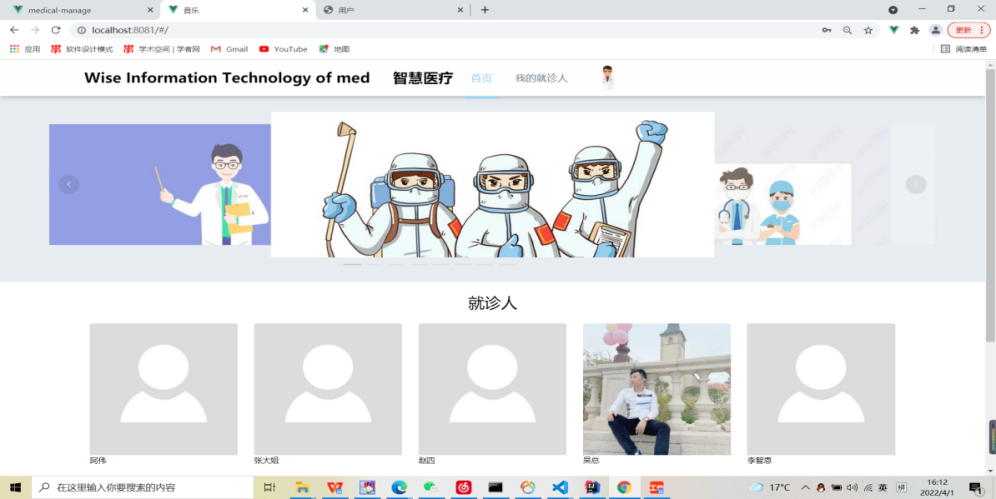


图5-9 医生首页图

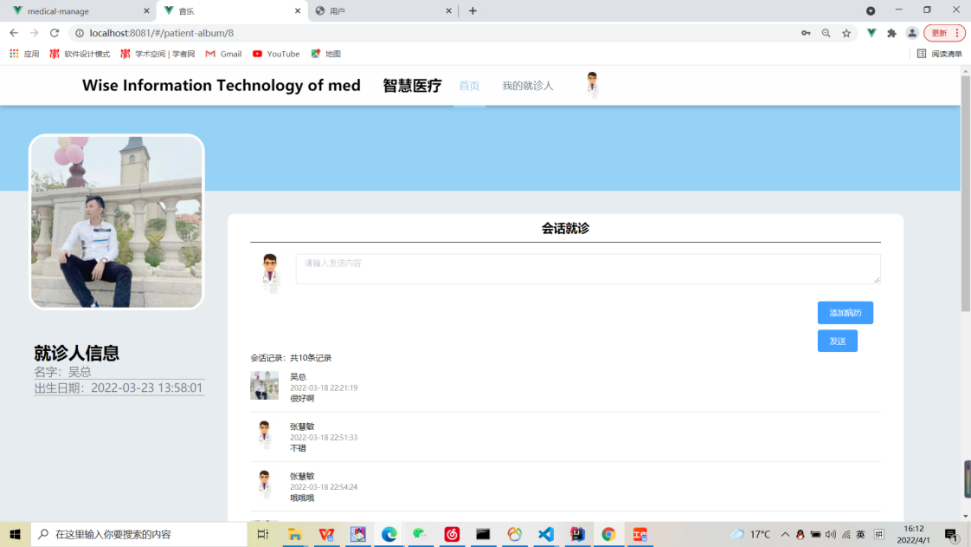


图5-10 医生会话就诊图

### 5.3.3 管理员首页

系统左侧导航栏，管理员登录进来以后，系统直接切换到系统首页，首页会显示系统的部分统计信息，比如对医生性别比例的统计信息，对医生根据职称划分的人才比例的统计信息，包括了对医生总数，就诊人数，药品，处方的数量的统计；左侧分别显示管理员对医护人员、患者、病历、药品、处方等管理入口，对数据进行的直接的操作；系统顶部信息展示系统后台信息，其中包括了管理员的照片信息，账户名称，在最右侧下拉菜单显示退出登录信息与展示页面全屏操作。

系统首页前端方法说明如表5-11所示。

表5-11 首页前端方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| GetByPatientNumber() | 统计患者数量信息，用于界面统计展示。 |
| GetByDoctorNumber() | 统计医生数量信息，用于界面统计展示。 |
| GetByDoctorSex() | 统计医生的性别比例信息。 |
| GetByDoctorTitle() | 根据医生的不同职称，展示人才比例信息 |

系统首页设计效果如图5-11、5-12所示。

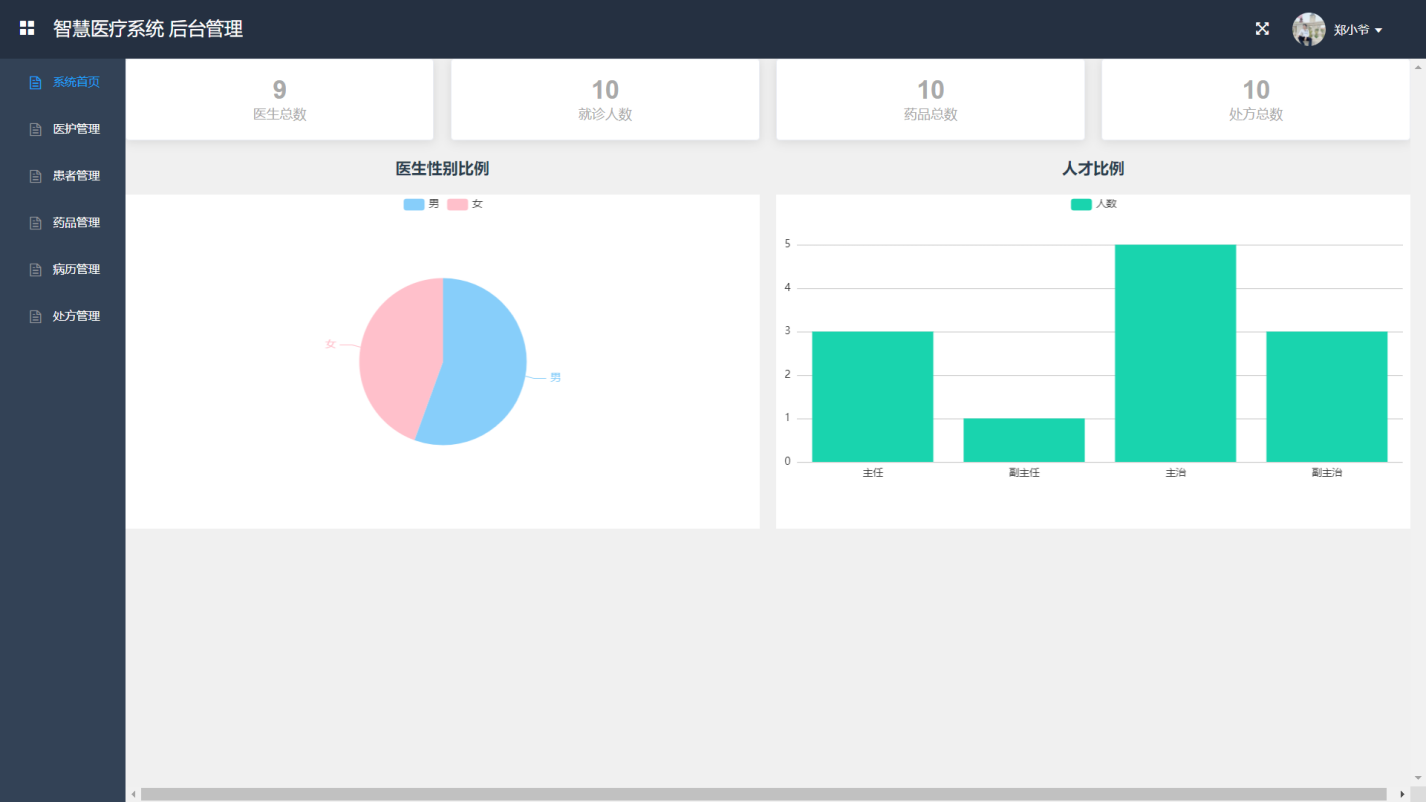


图5-11 左侧导航栏



图5-12 顶部导航栏

## 5.4 后台管理

### 5.4.1 医护管理

医护管理模块只有系统管理员才有权限访问，管理员在进入该页面之后，可以对该页面的医护信息进行添加、删除、修改、查询。其中，删除功能包括了批量删除信息与单项删除；页面的医护信息编辑对应的每一条记录，都有相对应的操作菜单；查询功能采用了对医护人员姓名进行模糊查询，并且能够根据输入的查询条件做到信息的实时展示，提高了页面的刷新效率。后台可以在不经查询的情况下，根据患者管理，直接获取每一位医生下的患者信息；在该页下端，采用了分页设置，当人员信息数量超过一定值的时候，会显示分页信息，可供选择。

医护管理模块前端相关方法说明如表5-12所示。

表5-12 前端相关方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| HandleCurrentChange() | 界面信息切换，负责根据界面查询条件展示不同信息。 |
| GetDoctorData() | 获取当前界面所有医生信息。 |
| UploadUrl() | 更新图片信息，加载医生的图片。 |
| AddDoctor() | 添加医生方法，负责响应后端接口。 |
| HandleEdit() | 编辑页面，保存可编辑的对应信息。 |
| EditSave() | 保存编辑页面信息，响应后端接口。 |
| DeleteRow() | 根据每行信息进行批量删除，调用单项删除方法。 |

医护管理模块操作类和类方法说明如表5-13、5-14所示。

表5-13 医护信息操作相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| DoctorController | 医护信息控制器，调用医护信息操作所需的接口。 |
| Doctor | 医护实体类，与数据库中医护信息相对应。 |
| DoctorMapper | 数据交互类，后端操作医护信息的方法。 |
| DoctorService | 医护操作服务类，封装与医护信息操作相关的逻辑调用。 |

表5-14 类方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| DoctorController.addDoctor() | 接收前端的参数，添加医护信息。 |
| DoctorController. updateDoctor() | 接收前端的参数，更新医护信息。 |
| DoctorController.deleteDoctor() | 接收前端的参数，删除医护信息。 |
| DoctorController. selectByPrimaryKey() | 根据传入主键，获取整个医护对象信息。 |
| DoctorController. doctorOfName() | 接收前端传入的值，进行模糊查询。 |
| DoctorController. updateDoctorPicture() | 上传/更新医生图片，接收图片信息并加载到项目中。 |
| DoctorController. login() | 接收医护信息，匹配数据库中账号密码。 |

医护管理页面设计效果如图5-13所示。

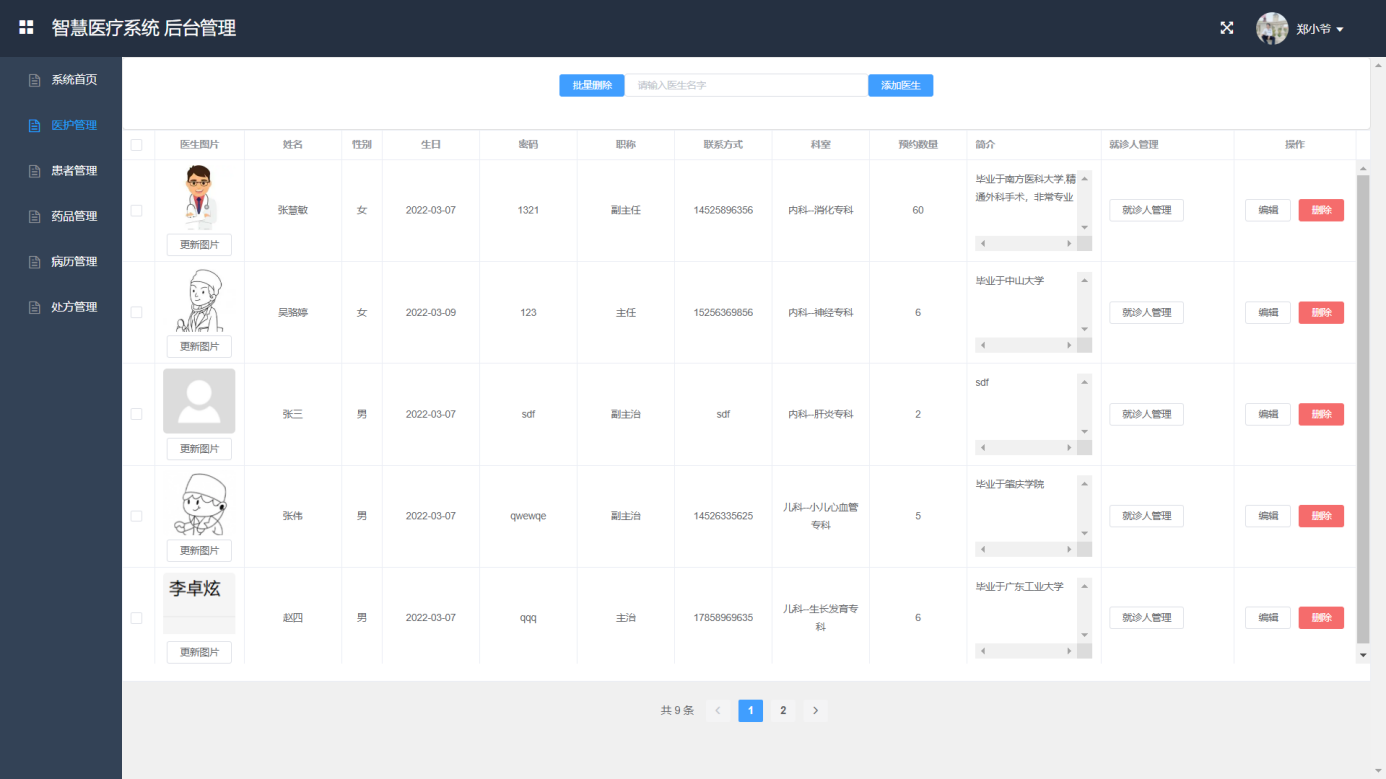


图5-13 医护管理

### 5.4.2 患者管理

患者管理模块只有系统管理员才有权限访问，管理员在进入该页面之后，可以对该页面的患者信息进行添加、删除、修改。其中，删除功能包括了批量删除信息与单项删除；页面的患者信息编辑对应的每一条记录，都有相对应的操作菜单；查询功能采用了对医护人员姓名进行模糊查询，并且能够根据输入的查询条件做到信息的实时展示，提高了页面的刷新效率。在该页下端，采用了分页设置，当人员信息数量超过一定值的时候，会显示分页信息，可供浏览。

患者管理模块前端相关方法说明如表5-15所示。

表5-15 前端相关方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| HandleCurrentChange() | 界面信息切换，负责根据界面查询条件展示不同信息。 |
| GetPatientData() | 获取当前界面所有患者信息。 |
| UploadUrl() | 更新图片信息，加载患者的图片。 |
| AddPatient() | 添加患者方法，负责响应后端接口。 |
| HandleEdit() | 编辑页面，保存可编辑的对应信息。 |
| EditSave() | 保存编辑页面信息，响应后端接口。 |
| DeleteRow() | 根据每行信息进行批量删除，调用单项删除方法。 |

患者管理模块操作类和类方法说明如表5-16、5-17所示。

表5-16 患者信息操作相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| PatientController | 患者信息控制器，调用患者信息操作所需的接口。 |
| Patient | 患者实体类，与数据库中患者信息相对应。 |
| PatientMapper | 数据交互类，后端操作患者信息的方法。 |
| PatientService | 患者操作服务类，封装与患者信息操作相关的逻辑调用。 |

表5-17 类方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| PatientController.addPatient() | 接收前端的参数，添加患者信息。 |
| PatientController. updatePatient() | 接收前端的参数，更新患者信息。 |
| PatientController.deletePatient() | 接收前端的参数，删除患者信息。 |
| PatientController. selectByPrimaryKey() | 根据传入主键，获取整个患者对象信息。 |
| PatientController. patientOfName() | 接收前端传入的值，进行模糊查询。 |
| PatientController. updatePatientPicture() | 上传/更新患者图片，接收图片信息并加载到项目中。 |
| PatientController. login() | 接收患者信息，匹配数据库中账号密码。 |

患者管理页面设计效果如图5-14所示。

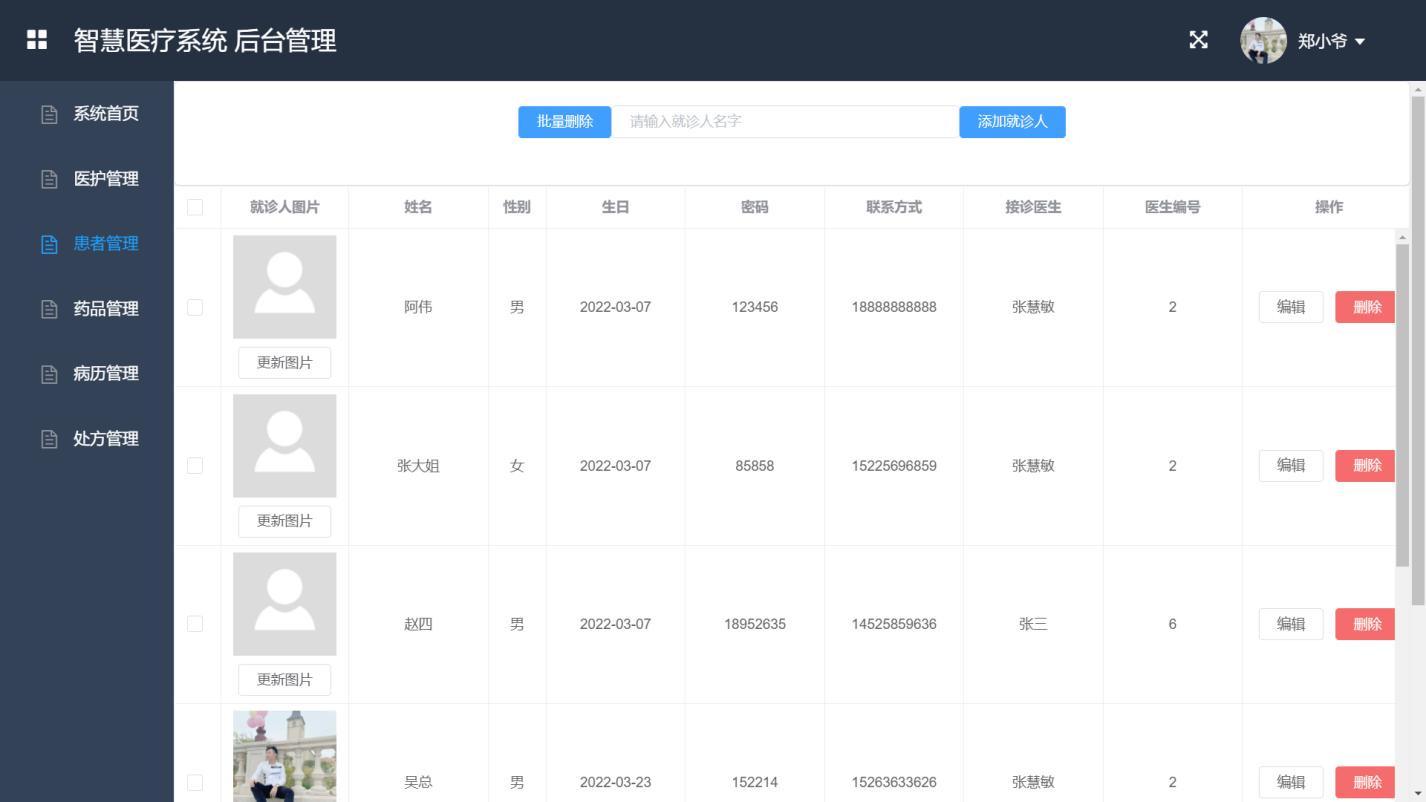


图5-14 患者管理

### 5.4.3 药品管理

药品管理模块只有系统管理员才有权限访问，管理员在进入该页面之后，可以对该页面的药品信息进行添加、删除、修改。其中，删除功能包括了批量删除信息与单项删除；页面的药品信息编辑对应的每一条记录，都有相对应的操作菜单；查询功能采用了对药品名称进行模糊查询，并且能够根据输入的查询条件做到信息的实时展示，提高了页面的刷新效率。在该页下端，采用了分页设置，当药品信息数量超过一定值的时候，会显示分页信息，可供浏览。

药品管理模块前端相关方法说明如表5-18所示。

表5-18 前端相关方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| HandleCurrentChange() | 界面信息切换，负责根据界面查询条件展示不同信息。 |
| GetDrugData() | 获取当前界面所有药品信息。 |
| AddDrug() | 添加药品方法，负责响应后端接口。 |
| HandleEdit() | 编辑页面，保存可编辑的对应信息。 |
| EditSave() | 保存编辑页面信息，响应后端接口。 |
| DeleteRow() | 根据每行信息进行批量删除，调用单项删除方法。 |

药品管理模块操作类和类方法说明如表5-19、5-20所示。

表5-19 药品信息操作相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| DrugController | 药品信息控制器，调用药品信息操作所需的接口。 |
| Drug | 药品实体类，与数据库中药品信息相对应。 |
| DrugMapper | 数据交互类，后端操作药品信息的方法。 |
| DrugService | 药品操作服务类，封装与药品信息操作相关的逻辑调用。 |

表5-20 类方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| DrugController.addDrug() | 接收前端的参数，添加药品信息。 |
| DrugController. updateDrug() | 接收前端的参数，更新药品信息。 |
| DrugController.deleteDrug() | 接收前端的参数，删除药品信息。 |
| DrugController. selectByPrimaryKey() | 根据传入主键，获取整个药品对象信息。 |
| DrugController. DrugOfName() | 接收前端传入的值，进行模糊查询。 |

药品管理页面设计效果如图5-15所示。

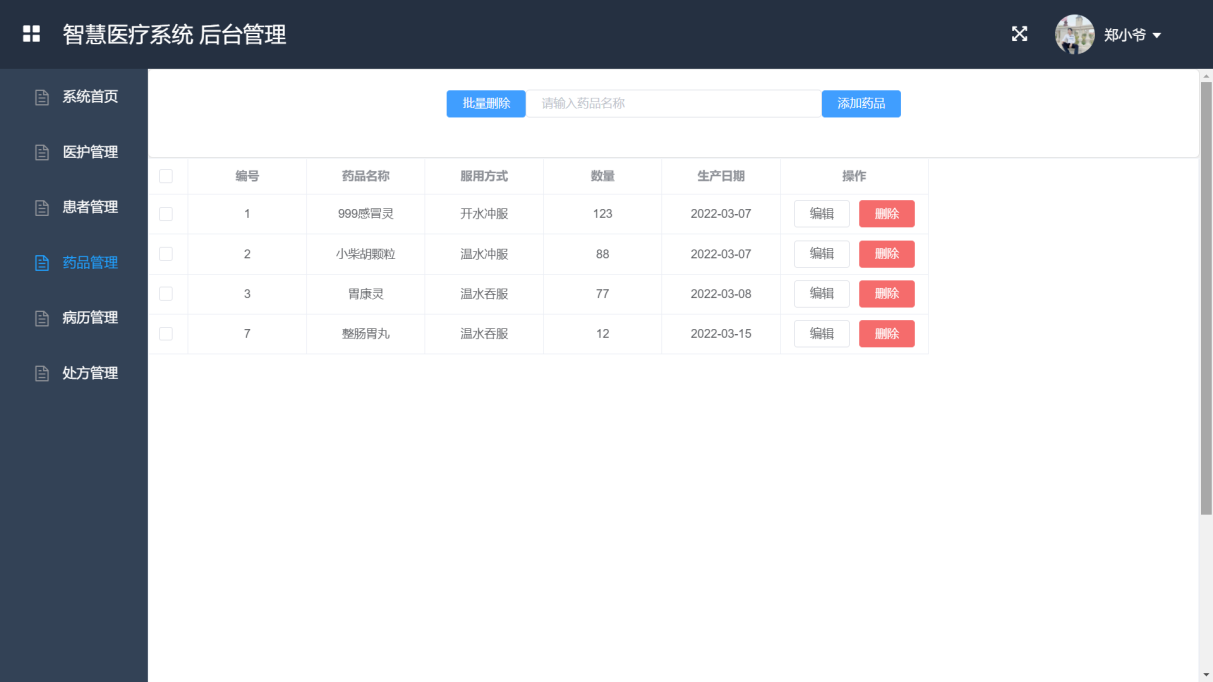


图5-15 患者管理

### 5.4.4 病历管理

病历管理模块只有系统管理员才有权限访问，管理员在进入该页面之后，可以对该页面的病历信息进行添加、删除、修改。其中，删除功能包括了批量删除信息与单项删除；页面的病历信息编辑对应的每一条记录，都有相对应的操作菜单；查询功能采用了对病历中患者姓名进行模糊查询，并且能够根据输入的查询条件做到信息的实时展示，提高了页面的刷新效率。在该页下端，采用了分页设置，当病历信息数量超过一定值的时候，会显示分页信息，可供浏览。

病历管理模块前端相关方法说明如表5-21所示。

表5-21 前端相关方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| HandleCurrentChange() | 界面信息切换，负责根据界面查询条件展示不同信息。 |
| GetRecordData() | 获取当前界面所有病历信息。 |
| AddRecord() | 添加病历方法，负责响应后端接口。 |
| HandleEdit() | 编辑页面，保存可编辑的对应信息。 |
| EditSave() | 保存编辑页面信息，响应后端接口。 |
| DeleteRow() | 根据每行信息进行批量删除，调用单项删除方法。 |

病历管理模块操作类和类方法说明如表5-22、5-23所示。

表5-22 病历信息操作相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| RecordController | 病历信息控制器，调用病历信息操作所需的接口。 |
| Record | 病历实体类，与数据库中病历信息相对应。 |
| RecordMapper | 数据交互类，后端操作病历信息的方法。 |
| RecordService | 病历操作服务类，封装与病历信息操作相关的逻辑调用。 |

表5-23 类方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| RecordController.addRecord() | 接收前端的参数，添加病历信息。 |
| RecordController. updateRecord() | 接收前端的参数，更新病历信息。 |
| RecordController.deleteRecord() | 接收前端的参数，删除病历信息。 |
| RecordController. selectByPrimaryKey() | 根据传入主键，获取整个病历对象信息。 |
| RecordController. RecordOfName() | 接收前端传入的值，进行模糊查询。 |

病历管理页面设计效果如图5-16所示。

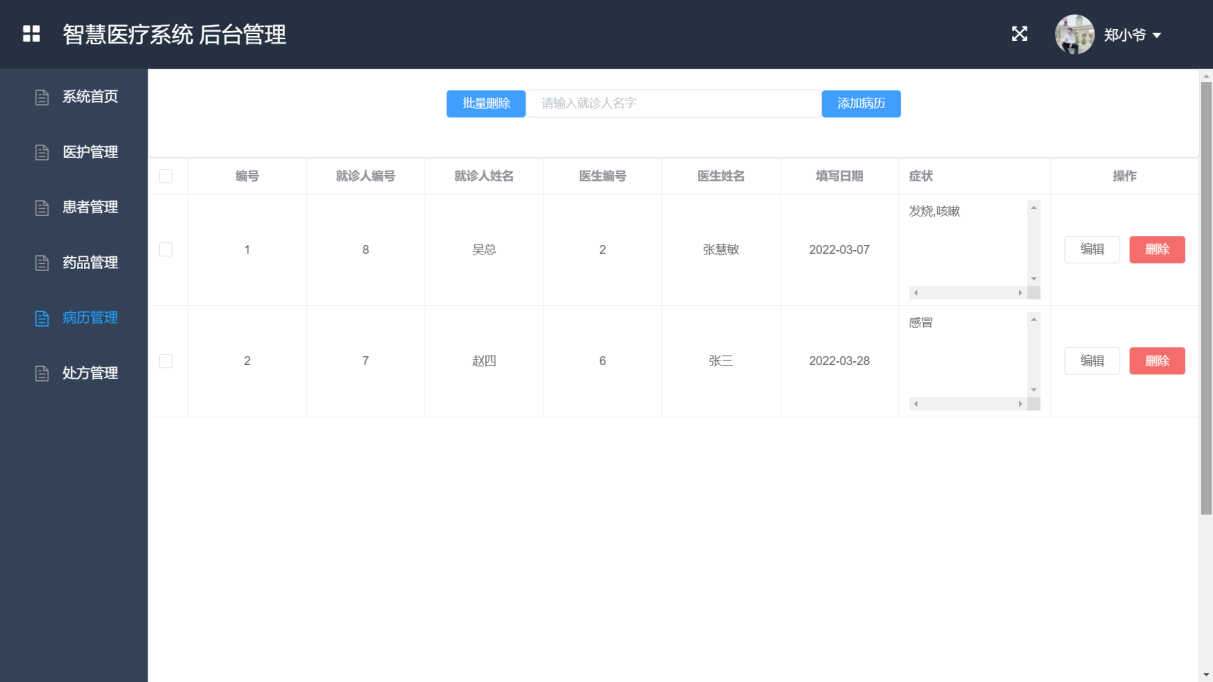


图5-16 病历管理

### 5.4.5 就诊会话

就诊会话功能主要分为医生与患者进行沟通所需，会话功能为医生或者患者登录系统之后，可以根据自己的需求分别选择不同的医生，或者医生选择自己相对应负责的患者，之后进行会话交流，会话过程中，每条会话信息显示相对应的发送者的内容，头像，发送时间等，在消息进行发送之后，会第一时间刷新界面并显示消息发送是否成功。每位医生与患者的会话信息仅限于两者查看，会话信息内容做到了对其他医生与患者之间的屏蔽，保证了会话内容的隐私性与保护性。会话结束后，医生界面可为患者添加病历信息。

会话信息模块前端相关方法说明如表5-24所示。

表5-24 前端相关方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| PostDialog() | 获取该界面的登录信息，将评论内容及相关信息发送到后端接口。 |
| GetDialog() | 获取对应的会话信息内容，调用后端会话接口。 |

会话信息模块操作类和类方法说明如表5-25、5-26所示。

表5-25 会话信息操作相关类说明

|  |  |
| --- | --- |
| 类名 | 类说明 |
| DialogController | 会话信息控制器，调用会话信息操作所需的接口。 |
| Dialog | 会话实体类，与数据库中会话信息相对应。 |
| DialogMapper | 数据交互类，后端操作会话信息的方法。 |
| DialogService | 会话操作服务类，封装与会话信息操作相关的逻辑调用。 |

表5-26 类方法说明

|  |  |
| --- | --- |
| 方法名 | 方法说明 |
| DialogController.addDialog() | 接收前端的参数，添加会话信息。 |
| DialogController. updateDialog() | 接收前端的参数，更新会话信息。 |
| DialogController.getDialog() | 接收前端的参数，获取会话信息。 |

会话信息页面设计效果如图5-17所示。

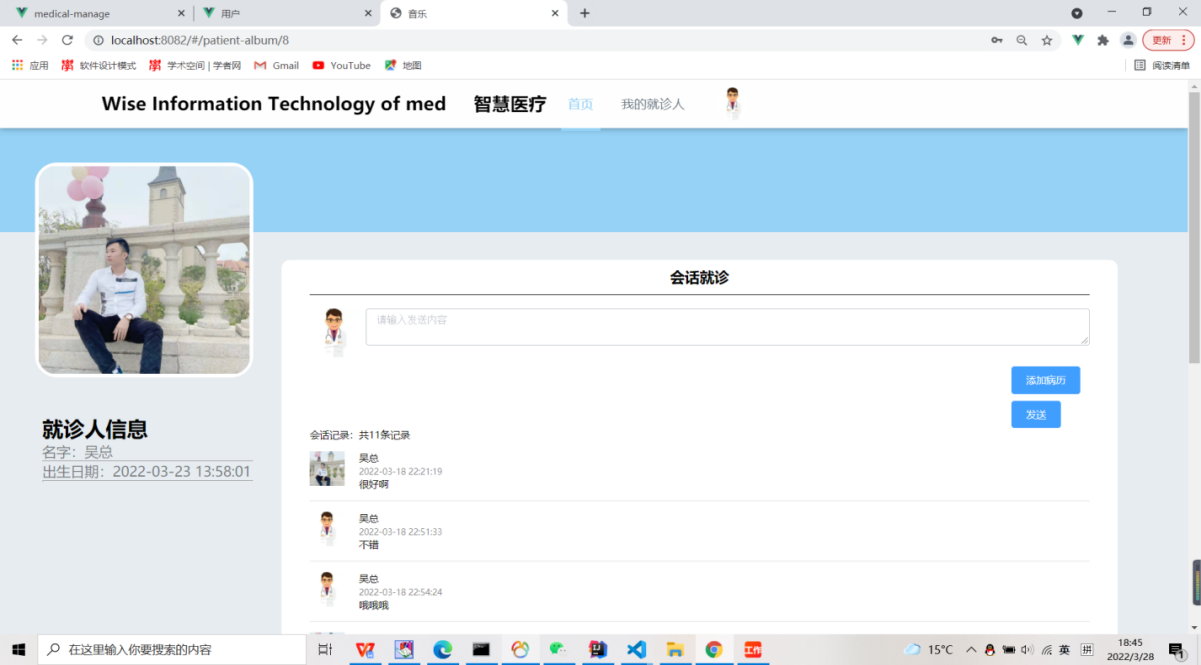


图5-17 会话信息

# 6 系统测试

本小节主要是针对智慧医疗系统的功能模块进行的测试，本次测试主要是用于测试系统所开发的功能与需求是否匹配，是否达到了用户的需求以及系统中可能出现的错误，在此前提下对最终的测试结果进行分析，最终判断出系统是否与需求分析中的功能相符合。

## 6.1 登录注册功能测试

登录注册功能为系统的核心业务，在不同身份人员登录系统后，分别对系统拥有不同的使用权限，不具有系统信息的人员进行注册信息后进行登录，本节对系统的登录功能编写测试用例并进行测试。

系统注册测试用例以及结果如下表6-1所示。

表6-1 系统注册登录功能测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 测试内容 | 用户操作 | 预期结果 | 测试结果 |
| 用户登录 | 输入正确的账号及密码 | 登录成功 | 通过 |
| 输入错误的账号或者密码 | 给出错误的提示信息，提示重新输入 | 通过 |
| 用户注册 | 输入系统要求的用户基本信息 | 确认提交后，跳转到登录界面进行登录 | 通过 |

## 6.2 后端管理功能测试

系统后端管理功能为系统管理员对医疗系统的核心管理业务，本小节对系统的后端功能部分编写测试用例并进行测试。

系统后端功能测试用例以及结果如下表6-2所示。

表6-2 系统后端功能测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 测试内容 | 用户操作 | 预期结果 | 测试结果 |
| 批量删除 | 选中需要删除的信息的复选框，选中之后点击批量删除 | 确认提交后，选中的信息被删除，并提示删除成功 | 通过 |
| 更新图片 | 点击要更新的图片下方的更新图片，在文件夹中挑选更新后的图片 | 确认提交后，页面更新选中的图片，并提示更新图片成功 | 通过 |
| 查询功能 | 点击要查询的文本框，输入要查询的名字，之后点击查询 | 查询到与该查询信息相对应的人员的全部信息，在下方列表进行显示 | 通过 |
| 添加功能 | 点击添加信息，之后在显示的添加框输入人员信息，之后点击添加 | 确认提交后，显示添加成功，下方列表更新添加后的人员信息总列表 | 通过 |
| 编辑功能 | 点击编辑信息，根据需要编辑的内容进行修改，之后点击修改 | 确认提交后，显示修改成功，在下方列表中显示修改后的人员信息 | 通过 |

## 6.3 就诊会话功能测试

就诊会话功能为医生与患者的沟通所在，也是系统的核心功能。本小节针对就诊会话功能编写测试用例并进行测试。

系统就诊会话功能测试用例以及结果如下表6-3所示。

表6-3 系统就诊会话功能测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 测试内容 | 用户操作 | 预期结果 | 测试结果 |
| 医生会话功能 | 医生进入患者界面后，发起会话信息 | 提交后，提示发送会话成功 | 通过 |
| 患者会话功能 | 患者进入医生界面后，发起会话信息 | 提交后，提示发送会话成功 | 通过 |

# 7 结论

本论文所构建的智慧医疗系统，主要是由管理者，医生，患者三种不同身份的人员进行使用，主要实现的功能为管理员具有本系统的最高权限，能够实现对医护人员信息的管理，即实现信息的增删改查等功能，并对系统中的数据做一个简单的统计。医生与患者能够通过该系统实现就诊会话功能，根据不同患者，能够做到患者与患者之间的会话记录的相互屏蔽。在前端平台，医生与患者分别采用不同的端口进行系统登录，管理员直接后端系统的管理平台界面，结合后端分离技术实现端口异域登录。由于本系统的开发时间有限，无法对系统的需求分析做到尽可能的完善，因此，在开发过程中出现了不同程度的需求变更，从而延长了系统的开发时间，独立开发系统，难以做到对系统的数据做出安全性与保密性的提升，在线程并发方面，不能很好的保证随着患者数量的上升，系统响应机制能否做出适当的线程安排。尽管系统在功能使用上，衔接性仍然不够，但目前而言，已经能够实现大部分的功能，保证了医生与患者就诊的可能性，基本实现了远程就诊，数据后台管理，最终初步达到了智慧医疗的概念。

# 参考文献:

1. 毛戈,李晶,姚弘毅.基于智慧医院的电子病历应用和设计[J].湖北大学学报(自然科学版),2021,43(06):706-712.
2. 孙海波,齐顺利,孙平,于晓倩.智慧医疗系统设计与实现[J].网络安全技术与应用,2021(10):130-131.
3. 程凯,谷志峰,朱炳旭,王君洋,靳豪杰.智慧医疗管理平台的设计与实现[J].无线互联科技,2021,18(17):50-51.
4. 周寅,张振方,周振涛,张扬.基于Java Web的智慧医疗问诊管理系统的设计与应用[J].中国医学装备,2021,18(08):132-135.
5. 刘宇航.基于智慧医疗的医院信息系统设计与应用[J].智慧中国,2021(07):82-83.
6. 童俊.基于微服务架构智慧门诊建设提升医院医疗服务质量[J].数字技术与应用,2021,39(07):129-131.
7. 游先铭,王欣.基于智慧医疗的挂号服务系统的应用价值探析[J].中国设备工程,2021(13):97-98.
8. 崔广森.基于大数据的智慧移动医疗信息系统结构分析[J].信息记录材料,2021,22(07):107-109.
9. 黄巧,曹奕,李雪.智慧医疗视角下产科门诊应用全预约挂号系统的回顾性研究[J].现代医院,2021,21(06):903-906.
10. 陈子璇.智慧医院管理背景下的医院信息化建设研究[J].现代信息科技,2021,5(11):124-127.
11. 尹康杰. 面向智慧医疗的物联网管理平台设计及实现[D].北京邮电大学,2021.
12. 崔昌云.智慧城市大数据平台的研究与应用探究[J].中国信息化,2021(04):112-113.
13. 左琳.基于智慧医疗的物联网技术与医疗设备管理系统的融合[J].信息系统工程,2021(04):119-120+124.
14. 王曼维,杨荻,李岩,及松洁.基于SpringBoot框架的智慧医疗问诊系统设计与实现[J].中国医学装备,2022,19(03):133-136.
15. 尹慧子. 智慧医疗情境下信息交互及效果评价研究[D].吉林大学,2020.DOI:10.27162/d.cnki.gjlin.2020.007386.
16. 马国强,王哲.基于JavaWeb的智慧医疗问诊系统设计与实现[J].信息技术与信息化,2017(09):105-106.
17. 刘毅. 基于智慧医疗的个人健康门户系统设计与实现[D].电子科技大学,2012.
18. 罗长阳. 智慧医疗监控终端软件系统的设计与实现[D].电子科技大学,2012.
19. Seera M , Lim C P . A hybrid intelligent system for medical data classification[J]. Expert Systems with Applications, 2014, 41(5):2239-2249.
20. Saadawi G , Azevedo R , Castine M , et al. Factors affecting feeling-of-knowing in a medical intelligent tutoring system: the role of immediate feedback as a metacognitive scaffold[J]. Advances in Health Sciences Education, 2010, 15(1):9-30.
21. Feyzi-Behnagh R , Azevedo R , E Legowski…. Metacognitive scaffolds improve self-judgments of accuracy in a medical intelligent tutoring system[J]. Instructional Science, 2014, 42(2):159-181.

**Design and implementation of intelligent medical system based on Springboot+Vue**

**Abstract:** China is a country with uneven development, and medical resources are relatively scarce for people in remote and underdeveloped areas. Based on this, this paper designs and implements data visualization based on Spingboot back-end technology, MySQL as database, Vue as front-end framework, and ElementUi technology.The intelligent medical system is designed and implemented.With the help of the Internet, the medical system realizes the remote treatment between patients and doctors, and the medical department stores and grasps the information of patients and medical treatment in real time, realizing the convenient management of medical information, helping to maximize the utilization of medical resources, and preliminarily realizing the concept of "smart medicine".

**Key words:** Intelligent medical treatment; Front and rear end separation; The database

# 致 谢

行文至此，大学时光落下帷幕。从一个懵懂的少年，满怀梦想，到走完象牙塔，四年时光悄然而逝。我也清楚，未来与过去不可交接，我的最后一个大学梦想也未能实现，我用自己的努力，从一个差生走进了本科，与警校失之交臂，与政法大学，与研究生未能有过邂逅，只能说来日方长。我很喜欢汪国真的一句话，既然选择了远方，便只顾风雨兼程。这里只是我另一条路的开始，我还想去追逐自己的梦想，就在这里，预示下一个梦想的开始吧！

非常感谢我的指导老师在论文完成过程中对我的指导。从开题到进展汇报，到论文的修改，李芳老师都给予我细心的指导。在此向李芳老师致以诚挚的谢意和崇高的敬意！