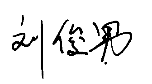
|  |
| --- |
| 河南大学 2022 届本科毕业论文 |
|  |
| **基于Spring Cloud框架的医院预约就诊系统设计与实现** |
|  |
| **Design and Implementation of Online Appointment Registration System Based on Spring Cloud Framework** |

|  |  |  |
| --- | --- | --- |
| 论文作者姓名 | ： | 艾鑫鑫 |
| 作者学号 | ： | 1825010094 |
| 所在学院 | ： | 软件学院 |
| 所学专业 | ： | 网络工程 |
| 导师姓名 | ： | 刘俊男 |
| 导师职称 | ： | 讲师 |
|  |  |  |
| **2022 年4月25日** | | |

**河南大学 2022 届本科生毕业论文（设计、创作）任务书**

|  |  |  |  |
| --- | --- | --- | --- |
| **论文（设计、创作）题目** | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | |
| **所在学院** | 软件学院 | **专 业** | 网络工程 |
| **学生姓名** | 艾鑫鑫 | **学 号** | 1825010094 |
| **一、毕业论文（设计、创作）要求**  **1．设计内容**  通过设计一个分布式微服务的网上预约就诊系统，实现用户网上注册登录和预约的主要功能。  **2．基本要求**  利用Spring框架平台搭建一个微服务项目。能够完整支撑用户进行正常使用预约功能。  第一个模块是用户管理模块，用户管理模块主要负责维护数据库用户表。首先注册功能的实现需要添加用户功能，添加用户前需要查询该用户是否已经注册，则需要根据手机号查询用户功能。登录功能需要使用到用户的查询功能。  第二个模块是医生管理模块，医生管理模块主要负责数据库医生表的维护。用户点击选择科室后，会跳转进科室详细信息页面。该页面会展示该科室中的所有医生列表，我们需要调用查询该科室所有医生功能返回数据给前端页面。选择医生后，页面上方用来展医生的详细信息，这里需要调用根据医生id查询功能。下方展示医生可用预约时间段。后台医生管理需要对医生表进行删除、添加、修改操作。  第三个模块是预约时间表管理模块，预约时间表管理模块主要负责管理预约时间表，后台管理页面可以选择医生进行操作该医生的时间表。主要操作有添加时间表、修改时间表、修改时间表放号数。  第四个模块是，订单管理模块，订单管理模块主要负责订单表和预约时间表的管理。用户预约成功时会向订单表里添加一条新的预约数据，同时并修改预约时间表里的剩余放号数。用户取消预约时释放号源，并修改预约时间表。  **3．要解决的问题**  分布式微服务架构的搭建；如何实现预约流程中所用到的每个功能，并能完成预约操作。  **4．要求提交的成果**  毕业设计源程序及论文。 | | | |
| **二、毕业论文（设计、创作）进度安排**  2021年10月至2021年11月完成并提交开题报告；  2021年11月至2022年1月底完成毕设系统搭建并填充好各部分功能，提交中期检查报告；  2022年2月至2022年3月中旬完成毕业论文初稿。  2022年3月中旬到2022年4月完成论文修改和查重，达到答辩要求。 | | | |
| **三、需收集的资料和指导性参考文献**  [1]张丹峰. 开放统一预约系统设计[D].西安电子科技大学,2018.  [2]刘洋,赵博.浅析医院自助服务信息化管理[J].经营管理者,2018(12):84.  [3]赵欣蝶,茅梦颖,毛雨静,曾驿涵.以网上挂号为例探究老年人群新媒介接触现状[J].传媒论坛,2021,4(01):22-23+26.  [4] 黄金链,郭丹媚.医院门诊预约挂号服务分析[J].实用临床护理学电子杂志,2017,2(41):150-151.  [5]张明,杨晗.网上预约系统在医院的应用[J].现代经济信息,2018(17):103.  [6]邓倩. 基于spring框架的网上预约挂号平台的设计与实现[D].电子科技大学,2016.  [7]童良. 基于B/S的医院信息管理系统的设计与实现[D].电子科技大学,2020.DOI:10.27005/d.cnki.gdzku.2020.001182.  [8]夏小涵. 私立儿童医院管理系统的设计与实现[D].吉林大学,2018.  [9]邢娜,郑蕾,王莉.医院网站预约挂号系统设计[J].解放军医院管理杂志,2021,28(10):929-930+969.DOI:10.16770/J.cnki.1008-9985.2021.10.012.  [10]魏星. 基于B/S架构的预约挂号系统的设计与实现[D].天津工业大学,2018.  [11]杨茂琼,杨成.多元化预约挂号在三级甲等综合医院门诊管理中的应用研究[J].人人健康,2021(22):84-85.  [12]张建.预约挂号要等等老年人[J].晚晴,2021(03):5.  [13]陈喜芳,刘绮才,周风齐.门诊预约挂号对患者服务质量及满意度的调查研究[J].实用临床护理学电子杂志,2020,5(20):171+173. | | | |

**指导教师签名**

**2021年 11 月 1 日**

**河南大学 2022 届本科生毕业论文（设计、创作）开题报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **论文（设计、创作）题目** | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | | |
| **所在学院** | 软件学院 | | **专 业** | 网络工程 |
| **学生姓名** | 艾鑫鑫 | | **学 号** | 1825010094 |
| **一、本课题研究意义**  网上预约就诊系统，是借助互联网，面向患者个人，实现各类就诊预约的远程服务系统，是医院预约就诊发展的一个趋势，也是医院将来逐步对外提供预约服务的一个重要组成部分，是以患者为中心开展医疗服务的重要革新措施，可有效地解决门诊病人高峰时就诊困难问题，能大大缓解医院门诊窗口的压力，有利于患者提前安排就医计划，减少候诊时间，也有利于医院提升管理水平，提高医疗服务效率，对患者、医院都有好处。  随着网络信息技术的不断发展,越来越多的传统行业利用信息技术对其生产活动进行了升级改造，无论从饮食、娱乐、交通、购物等方面都涌现出了很多在线服务。通过提供在线服务可以有效的提升业务的信息化处理水平并且提高资源的统筹管理效率[1]。  正确、方便、快捷的预约系统是患者获得医疗服务的前提保证，随着计算机的普遍应用和信息化的发展,人工挂号的比例将会逐渐下降。网上预约系统实现了互联网技术与传统窗口挂号服务的有机结合,能够让患者足不出户就可以选择并预约所需的专家及就诊时间。能够设计出一个对用户来说易懂，易用的预约就诊系统是非常有必要的。 | | | | |
| **二、国内外有关本课题的研究动态**  一个设计良好的预约系统能够节省许多人力资源，和极大的降低管理的成本。随着社会的信息化发展，医院信息化管理也是必然的趋势。现在部分医院也有自己的预约系统，但是患者对预约系统的了解程度及操作熟练度均有所不同，所以有很大部分患者感觉使用体验繁琐；[2]对于有智能手机老年人来说网上预约挂号部分操作还比较困难，比如验证码的填写较为困难、操作页面字体太小、操作流程不明了等问题[3]。所以设计系统方面必须要考虑各个人群对于系统的使用是否容易接受。现在依然大部分人还是通过现场排队预约，但是网上预约已经呈现逐步增加的趋势。因为网上预约可以避免排队的过程，还有部分人不愿意到医院现场进行预约。随着越来越多的人学会使用网上预约，未来医院网上预约会成为主流的预约方式。  本次研究结果显示，现场预约(91.7%）依然是门诊挂号的主要方式不同的约访方式，但其优势并不显著，相比之下，电话预约的预约满意度明显提升，网上预约的流程难度、人力消耗以及时间消耗均较大，但其预约成功率与预约信息准确率最高(91.6%、98.0%）且爽约率最低(8.2%）。由此可见，不同的门诊预约就诊方式，其优缺点也存在一定的差异性[4]。每种预约方式都不可或缺，各有优缺点。所以为医院设计出一个网上预约就诊系统是非常有必要的。 | | | | |
| **三、本课题研究的基本内容**  分析设计门诊预约就诊的必备功能，并采用合适的开发工具完成相应的预约就诊系统，具体为：  1.用户的注册和登录。用户分为患者、医师和系统管理员三种。实现网上注册和登录功能，根据登录人员的权限等级设计不同的操作权限。  2.检查项的后台管理。  3.后台预约订单的管理。  4.管理每项每日可预约的数量。 | | | | |
| **四、本课题拟解决的主要问题**  1.预约就诊系统过于繁琐的问题：系统界面以及流程如何设计才能满足大部分患者更方便的进行预约就诊。  2.用户爽约问题：如何提高通过网上预约就诊的用户预约成功后就诊率。  3.用户不知道根据自己的病症要挂什么科的问题。  4.如何让后台编码更符合规范，即有一个合理的设计模式，以便于以后功能的扩展以及加强。  5.如何让本系统能与其他功能模块无缝的连接起来。 | | | | |
| **五、研究方法**  1.本系统采用java语言， jdk1.8 开发。前端技术栈有html、bootstrap、vue。分布式及安全技术栈有zookeeper、dubbo、Sringboot、SpringSecurity。持久化技术栈有MyBtis-Plus和Mysql。  2.过查阅网络和阅读相关书籍搜集有价值的资料，进行实际需求调查，确定系统的主要实现模块。  3.分析以往类似系统的设计思想，选择合适的模块组成架构。根据实际情况设计合适的数据库结构。  4.对设计好的系统进行调试，通过调试发现存在的问题，从而达到完善系统的目的。 | | | | |
| **六、主要创新点**  1.网上预约黑名单功能：对于半年内预约两次不来且未取消预约的，将其加入网上预约黑名单，封禁该用户三个月的网上预约功能。该功能可有效提高就诊率和提高用户对网上预约的重视程度。  2.智能搜索推荐功能：用户可以在预约页面输入病症的一些关键词，如“湿疹”、“红斑”等关键词系统会推荐挂皮肤科，关键词“咳嗽”、“喉咙痒”等关键词会推荐挂呼吸科等等。而且会把最多人挂的科室放在明显的位置，提高用户的使用体验。  3.候补预约功能：当预约人数已满的适合提供候补预约功能，当有人取消预约后会按照候补预约的顺序将预约资格发放给候补预约的人。 | | | | |
| **七、主要参考文献**  [1] 张丹峰. 开放统一预约系统设计[D].西安电子科技大学,2018.  [2]刘洋,赵博.浅析医院自助服务信息化管理[J].经营管理者,2018(12):84.  [3]赵欣蝶,茅梦颖,毛雨静,曾驿涵.以网上挂号为例探究老年人群新媒介接触现状[J].传媒论坛,2021,4(01):22-23+26.  [4]黄金链,郭丹媚.医院门诊预约挂号服务分析[J].实用临床护理学电子杂志,2017,2(41):150-151.  [5] 张明,杨晗.网上预约系统在医院的应用[J].现代经济信息,2018(17):103.  [6]邓倩. 基于spring框架的网上预约挂号平台的设计与实现[D].电子科技大学,2016.  [7]童良. 基于B/S的医院信息管理系统的设计与实现[D].电子科技大学,2020.DOI:10.27005/d.cnki.gdzku.2020.001182.  [8]夏小涵. 私立儿童医院管理系统的设计与实现[D].吉林大学,2018.  [9]邢娜,郑蕾,王莉.医院网站预约挂号系统设计[J].解放军医院管理杂志,2021,28(10):929-930+969.DOI:10.16770/J.cnki.1008-9985.2021.10.012.  [10]魏星. 基于B/S架构的预约挂号系统的设计与实现[D].天津工业大学,2018.  [11]张岩,孙莉,王慧玲,王晨,陈莲莲.基于Java Web的医院网上预约挂号系统[J].淮南师范学院学报  [12]邢娜,郑蕾,王莉.医院网站预约挂号系统设计[J].解放军医院管理杂志,2021,28(10):929-930+969.,2016,18(03):112-114. | | | | |
| **具体时间及写作进度安排** | | | | |
| **起止日期** | | **主要工作内容** | | |
| 2021.09.20-2021.11.14 | | 与导师交流，查找资料，确定毕业设计选题，撰写开题报告。 | | |
| 2021.11.15-2021.11.20 | | 修改完善开题报告，提交给导师审核。 | | |
| 2021.11.21- 2021.11.30 | | 答辩委员会（组）完成开题审查。 | | |
| 2021.12.01-2022.03.15 | | 根据所选课题进行毕业设计，填写中期检查表提交给导师，导师负责审查，答辩委员会（组）审核。 | | |
| 2022.03.16-2022.04.20 | | 根据导师的建议和指导，撰写毕业论文，完成论文查重。 | | |
| 2022.04.21-2022.04.30 | | 论文评阅和答辩。 | | |
| 2022.05.01-2022.05.10 | | 根据答辩建议修改毕业论文。 | | |
| **指导教师对开题报告的意见**  学生艾鑫鑫的选题《基于Spring Cloud框架的医院预约就诊系统设计与实现》，紧扣专业方向、紧扣现实，做到理论与实践结合、与实习体会结合，有现实意义，有完成选题的能力和条件，其开题报告体现了我院的人才培养目标的要求。同意开题。  **指导教师签名：**  **2021 年11月30日** | | | | |
| **系（室）对本课题开题的意见**  同意开题。  **微信图片_20220503201221**  **负责人签名： 2021 年11月30日** | | | | |

**河南大学 2022 届本科生毕业论文（设计、创作）**

**中期进展情况检查表**

|  |  |  |  |
| --- | --- | --- | --- |
| **论文（设计、创作）题目** | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | |
| **所在学院** | 软件学院 | **专 业** | 网络工程 |
| **学生姓名** | 艾鑫鑫 | **学 号** | 1825010094 |
| **一、毕业论文（设计、创作）进展情况**  1.网站的的前端项目搭建完成。  2.后端功能编写也已经基本完成。  3.微服务框架搭建完成。  4.毕业论文概论编写完成。 | | | |
| **二、毕业论文（设计、创作）存在问题及解决方案**  存在问题：   1. 前后端跨域问题。 2. 数据库时间和查询出的时间不一致。   解决方案：   1. 使用@CrossOrigin注解。 2. jackson解析器配置spring.jackson.time-zone=GMT+8 | | | |
| **三、指导教师对学生论文（设计、创作）进展等方面的评价**  医院预约就诊系统目前已基本实现核心功能，但也存排版问题，后期需要继续完善。目前的成果而言，同意通过中期检查，建议对研究内容进行总结，开始论文撰写工作。目前的成果而言，建议优化页面。项目完成后需要以严谨的语言，合理的布局，对研究内容总结，开始进行论文撰写。  **指导教师签名： 2022 年 3 月 15 日** | | | |
| **系（室）对本课题中期检查的意见**  同意通过中期审核。  **微信图片_20220503201221**  **负责人签名： 2022 年 3 月 15 日** | | | |

**河南大学 2022 届毕业论文（设计、创作）指导教师评价表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **论文（设计、创作）题目** | | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | | | |
| **所在学院** | | 软件学院 | **专 业** | 网络工程 | | |
| **学生姓名** | | 艾鑫鑫 | **学 号** | 1825010094 | | |
| **指导教师** | | 刘俊男 | **指导教师职称** | 讲师 | | |
| **指导教师评分** | **序号** | **评分项目** | **评分参考** | | | **得分** |
| 评阅指标（优秀标准） | | 满分（100分） |  |
| 1 | 选题质量 | 符合专业培养目标；有实际意义和推广价值。 | | 20 | 18 |
| 2 | 文献资料 应用能力 | 能独立查阅文献，具有收集、加工各种信息及获取新知识的能力。 | | 10 | 8 |
| 3 | 调查研究能力 | 能准确理解课题任务；研究方案设计合理；能独立从事调查研究；能综合运用所学知识发现与解决实际问题。 | | 30 | 27 |
| 4 | 论文（设计、创作）质量 | 格式、图表（或图纸）规范，符合要求；结构严谨，逻辑性强；内容翔实，表达准确流畅；学术价值或实用价值高。 | | 20 | 18 |
| 5 | 创新能力 | 观点独到，方法新颖，角度新颖。 | | 10 | 7 |
| 6 | 工作量及态度 | 工作量饱满；能圆满完成任务书规定的各项工作。 | | 10 | 7 |
| 总得分 | | | | 85 | |
| **指导教师评定意见** | 参照上述评价标准及学生论文（设计、创作）完成情况，做出具体评价： | | | | | |
| 论文《基于Spring Cloud框架的医院预约就诊系统设计与实现》，内容较为完整，篇幅符合学院规定，层次结构安排科学，论点突出，论述紧扣主题。语言表达流畅，格式符合规范要求；参考了较为丰富的文献资料，能够较为完整地实现相关功能；未发现抄袭现象，是一篇较好的毕业论文。同意该生参加毕业论文答辩。 | | | | | |
| 指导教师签名： 2022年4月25日 | | | | | |

**河南大学 2022 届毕业论文（设计、创作）评阅教师评价表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **论文（设计、创作）题目** | | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | | |
| **所在学院** | | 软件学院 | **专 业** | 网络工程 | |
| **学生姓名** | | 艾鑫鑫 | **学 号** | 1825010094 | |
| **评阅教师** | |  | **评阅教师职称** |  | |
| **评阅教师评分** | **序号** | **评分项目** | **评分参考** | | **得分** |
| 评阅指标（优秀标准） | 满分 （100分） |  |
| 1 | 选题质量 | 符合专业培养目标：有实际意义和推广价值。 | 20 | 17 |
| 2 | 文献资料应用能力 | 能独立查看文献，具有收集，加工各种信息及获取新知识的能力。 | 10 | 8 |
| 3 | 调查研究能力 | 能准确理解课题任务：研究方案设计合理:能独立从事调查研究，能综合运用所学知识发现与解决实际问题。 | 30 | 26 |
| 4 | 论文（设计、创作）  质量 | 格式，图表（或图纸）规范符合要求：结构严谨，逻辑性强 内容翔实，表达准确流畅：学术价值或使用价值高。 | 30 | 27 |
| 5 | 创新能力 | 观点独到，方法新颖，角度新颖。 | 10 | 7 |
| 总得分 | | | 85 | |
| **评阅教师评定意见** | 参照上述评价标准及论文（设计、创作）内容，做出具体评价： | | | | |
| 论文《基于JSP的河南旅游网站的设计与实现》，能够解决游客选择景点的问题，选题具有一定的实践指导意义。符合专业培养目标，能够达到综合训练目的。论文材料丰富，内容详实，较为全面地分析了问题，综合运用知识能力较强，是一篇较好的毕业论文。同意该生参加毕业论文答辩。 | | | | |
| 评阅教师签名： 郭念 2022 年4 月29 日 | | | | |

**河南大学 2022 届本科生毕业论文（设计、创作）**

**答辩成绩表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **论文（设计、创作）题目** | | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | | | | |
| **所在学院** | | 软件学院 | | **专 业** | | 网络工程 | |
| **学生姓名** | | 艾鑫鑫 | | **学 号** | | 1825010094 | |
| **答**  **辩**  **委**  **员**  **会**  **（组）**  **评**  **分**  **及**  **评**  **定**  **结**  **论** | **评分**  **项目**  **及**  **分值** | **答辩委员会（组）专家评分** | | | | | |
| **答 辩 情 况** | | **论 文 质 量** | | | **合计**  （100分） |
| 内容表达 情况  （15分） | 回答问题 情况  （25分） | 规范要求与文字表达  （20分） | 论文（设计、创作）质量和创新意识  （40分） | |
| **得分** | 12 | 22 | 16 | 35 | | 85 |
| **答辩**  **委员**  **会**  **（组）**  **评定**  **结论** | 论文《基于Spring Cloud框架的医院预约就诊系统设计与实现》选题合理，具有一定的现实性、针对性和实用性。文章篇幅符合学校的相关规定，内容较为完整，层次结构安排科学，主要工作基本阐述清楚。 语言表达基本流畅，总体而言满足毕业设计的要求。通过毕业答辩。  答辩委员会（组）签字：  王瀛 游大涛 郭念 蒋磊  2022 年 5 月 4 日 | | | | | |
| 毕业论文（设计、创作）答辩成绩： 85 分  答辩委员会（组）负责人签字：**微信图片_20220503201221**  2022 年 5 月 4 日 | | | | | | | |

**河南大学 2022 届本科生毕业论文（设计、创作）**

**答辩纪要**

|  |  |  |  |
| --- | --- | --- | --- |
| **论文（设计、创作）题目** | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | |
| **答辩人姓名** | 艾鑫鑫 | **指导教师** | 刘俊男 |
| **答辩时间** | 2022年5月4号 | **答辩地点** | 2022/5/4 腾讯会议471-741-050 |
| **答辩委员会(组)负责人** | 韩志杰 | **答辩委员会(组)成员** | 王瀛  游大涛  郭念  蒋磊 |
| 答辩中提出的问题及回答的简要情况记录：   1. 问：为什么选择使用SpringCloud技术进行开发，而不是SSH或者SSM，你对SpringCloud怎么考虑的？是之前对这个技术比较熟悉，还是自己摸索的？   答：因为这是一个比较新的框架，国内许多公司使用的技术还都比较老，并且我觉得使用新的技术也是一种技术层面上的创新。自己是在决定使用SpringCloud进行开发后自己通过视频学习这种开发技术的。   1. 摘要写的还行，技术写的不多，大量的解释性的话，去掉“潜移默化”，去掉主观性的评价。 2. 2.1功能模块框架图，框图不用箭头，就用一个框。 3. 预约流程图画的不对。 4. 缺一个完整的E-R图。 5. 摘要开头需要修稿，做到开门见山，别人读开头就知道你做的是什么。 6. 论文中不要出现大面积空白。   记录人签名： 2022 年 5 月 4 日 | | | |

**河南大学 2022 届本科生毕业论文（设计、创作）**

**综合成绩表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **论文（设计、创作）题目** | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | | | |
| **所在学院** | 软件学院 | | **专 业** | 网络工程 | |
| **学生姓名** | 艾鑫鑫 | | **学 号** | 1825010094 | |
| **指导教师评分** | 85 | **评阅教师评分** | 85 | **答辩评分** | 85 |
| **综合成绩** | 85 | **成绩等级** |  | **是否推优** |  |
| 所在系（室）意见 | 成绩合格。  **微信图片_20220503201221**  负责人签名： 2022 年5月4日 | | | | |
| 学院意见（仅推优论文填写） | 负责人签名（盖章）： 2022 年5月7日 | | | | |

**河南大学本科生毕业论文（设计、创作）承诺书**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **论文（设计、创作）题目** | | 基于Spring Cloud框架的医院预约就诊系统设计与实现 | | | | | | | |
| **姓名** | 艾鑫鑫 | | | **学 号** | | 1825010094 | | **专 业** | 网络工程 |
| **指导教师姓名** | | | 刘俊男 | | **职 称** | | 讲师 | | |
| **完成时间** | | | 2021 年9 月 20 日 — 2022 年 4 月 30 日 | | | | | | |
| **承诺内容：**   1. **本毕业论文（设计、创作）是学生 艾鑫鑫 在导师 刘俊男 的指导下独立完成的，没有抄袭、剽窃他人成果，没有请人代做，若在毕业论文（设计、创作）的各种检查、评比中被发现有以上行为，愿按学校有关规定接受处理，并承担相应的法律责任。** 2. **学校有权保留并向上级有关部门送交本毕业论文（设计、创作）的复印件和电子稿件。学校可以将毕业论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本论文。**   **备注：**    **学生签名： 指导教师签名：**  **2022 年 5 月 7 日 2022 年 5 月 7 日** | | | | | | | | | |

摘 要

近些年随着智能手机的普及和我国5G网络的建设，很多大型医院都配备了网上预约系统，这样节约了用户许多的时间和精力。但是在三四线城市和县城的医院依然还只有线下挂号一种模式，还没有自己的网上预约就诊系统。如果这些中小型医院也能轻松部署属于自己的网上预约系统，将会给医院预约系统减轻许多的压力。所以本课题想要设计出一个方便中小型医院部署和使用的模块化系统。

本课题围绕构建网上预约管理系统进行论述，对网上预约系统所需功能进行模块化设计。用户可以直接访问网址或者使用手机扫描二维码来进入网站进行预约。用户首先选择自己要预约的科室，然后可以选择自己想要预约的医生。页面会展示医生个人简介和可预约时间段。选择就诊人后，可以成功创建订单完成预约。

系统使用Spring Cloud框架和Mybatis进行设计，采用分布式微服务架构，前端页面使用ElementUI进行设计。将不同的功能模块封装为不同的微服务。不同功能的微服务通过注册中心进行服务发现和远程调用。并且如果用户量大的话可以将功能独立的微服务单独部署在一台服务器上，访问量较少的微服务可以多个部署在一台服务器上，部署起来较为灵活。本系统可以减轻医院线下挂号系统的压力，并且提高了医院用户的预约体验。

关键词：网上预约就诊； Spring Cloud；MyBatis；微服务

ABSTRACT

In recent years, with the popularity of smart phones and the construction of 5G networks in China, many large hospitals have been equipped with online reservation systems, which save users a lot of time and energy. However, hospitals in the third and fourth tier cities and counties are still only offline registration mode, and there is no online appointment system of their own. If these small and medium-sized hospitals can easily deploy their own online reservation system, it will reduce a lot of pressure on the hospital reservation system. Therefore, this project wants to design a modular system that is convenient for small and medium-sized hospitals to deploy and use.

This topic around the construction of online reservation management system is discussed, the online reservation system required functions for modular design. Users can visit the website directly or use their phone to scan a QR code to enter the site and make an appointment. Users first choose the department they want to make an appointment with, and then they can choose the doctor they want to make an appointment with. The page displays the doctor's personal profile and available appointments. After selecting the patient, you can successfully create an order to complete the appointment.

Spring Cloud framework and Mybatis are used to design the system, distributed micro-service architecture is adopted, and ElementUI is used to design the front-end page. Encapsulate different functional modules into different microservices. Microservices with different functions are discovered and remotely invoked through a registry. In addition, if the number of users is large, the micro-services with independent functions can be deployed on a single server, and the micro-services with less visits can be deployed on multiple servers, which makes the deployment more flexible. This system can reduce the pressure of offline registration system in hospital and improve the booking experience of hospital users.

**Keywords:** Hospital Medical Appointment System; Spring Cloud; MyBatis; micro service

目 录

摘 要 I

ABSTRACT III

第 1 章 绪论 1

1.1 课题来源 1

1.2 课题背景 1

1.3 国内外在该方向的研究现状及分析 1

1.3.1 国内现状 1

1.3.2 国外现状 2

1.4 相关技术介绍 2

1.4.1 Spring 2

1.4.2 MyBatis 3

1.4.3 Spring Cloud 3

1.5 论文组织结构 3

第 2 章 系统需求分析和相关技术分析 4

2.1 系统功能模块概述 4

2.1.1 功能模块架构 4

2.1.2 各功能模块结构描述 4

2.2 各功能详细需求 5

2.2.1 用户管理模块功能需求 5

2.2.2 医生管理模块功能需求 6

2.2.3 预约时间表管理模块功能需求 6

2.2.4 订单管理模块功能需求 6

2.3 预约就诊流程分析 6

2.4 本章小结 7

第 3 章 系统设计 9

3.1 系统体系结构 9

3.2 功能模块设计 9

3.3 数据库概念设计 10

3.4 数据库逻辑设计 14

3.5 微服务架构设计 17

3.6 本章小结 18

第 4 章 系统实现 19

4.1 前台模块的实现 19

4.1.1 注册 19

4.1.2 密码登录 21

4.1.3 发送短信验证码 23

4.1.4 短信验证码登录 23

4.1.5 科室展示 24

4.1.6 医生列表 25

4.1.7 创建预约订单 26

4.2 后台管理模块的实现 27

4.2.1 医师管理 27

4.2.2 医师排班 30

4.3 本章小结 32

第 5 章 系统测试 33

5.1 测试环境配置 33

5.2 测试方案 33

5.3 测试过程 34

5.4 测试结论 37

5.5 本章小结 37

第 6 章 总结与展望 38

参考文献 39

查重报告 40

格式检测报告 41

# 绪论

本章首先简要介绍本自选课题的研究背景，之后对于国内外预约挂号系统的分析，和在国内外使用技术方面差异及优缺点。最后对本系统使用到的技术进行简要的介绍。

## 课题来源

该课题为自选课题。

## 课题背景

随着智能手机的普及和信息化社会的发展，几乎每个行业都与互联网产生了微妙联系。互联网的加入给我们的日常生活也带来了诸多便利。人们足不出户便能完成之前许多需要付出大量精力完成的事。例如和我们息息相关的最新的网上社区团购买菜，今天下单明天就能拿到运来的新鲜蔬菜和各种杂货，其便利性和隔天快速送达的属性深受人们的喜爱。我认为网络的优势就在于能够让我们能够在空间距离上能做到像是我们就在现场一样，让我们付出更少的时间和精力来做到之前需要花费大量时间和精力事情。选择医院预约挂号作为课题是出于想要设计出一套功能完善的系统，作为医院线下预约挂号的一种补充。希望通过这个系统能够让网上预约挂号更加方便和人性化。

## 国内外在该方向的研究现状及分析

每一个医院的客户最先接触到的就是医院的就是预约就诊功能，用户如果能够在预约就诊环节得到很好的体验，也会间接影响到用户对医院的印象和对医院的评价。本小节分别对国内和国外的医院网上预约就诊平台使用到的技术以及现状进行分析和论述。

### 国内现状

国内许多医院虽然有网上预约挂号系统，但是其放号量还不到线下放号量的一半。许多用户想要通过网上预约挂号，但是苦于一直没有号放出，不得已到医院线下去挂号。国内现在越来越多的人都使用智能手机，许多老年人也学会了使用智能手机网上支付和聊天。未来网上挂号的需求量必然会越来越大，而且网上预约挂号还可以在新型冠状病毒疫情期间能尽量减少医院这种人员密集场所的防疫负担。

目前许多国内医院预约系统使用的技术还停留在使用jsp的阶段。代码前后端不分离的项目使得代码耦合度太高，非常不利于后期的维护和功能的拓展。开发过程一般为前端工程师写好html页面后再由后端工程师改写为jsp页面，这样的执行流程往往效率低且容易出现错误。在实际使用中每一次请求jsp页面都需要服务器把jsp编译成servlet，系统开销也比较大。国内这种前后端不分离的开发方式也逐渐在被淘汰。

### 国外现状

国外预约制度发展的比国内要早许多并且和国内预约挂号制度有些许不同。例如美国的预约挂号不是抢号占号，而是预约医生的某一段时间。在美国看病处处都需要预约，对比国内提供一站式服务的方式，许多人会觉得美国的看病效率会低许多。在国内预约爽约似乎是一种毫无惩戒的事，而国外对于爽约行为是非常严厉的，屡次爽约的用户会被医院标记，所以设计系统的时候要考虑对用户爽约行为的惩戒。设计预约挂号系统的时候要综合国内用户的习惯来制定适合中国人使用的系统。

目前国外java作为后端来开发主流使用到的技术是Spring Boot和Spring Cloud来做前后端分离的项目。当项目规模比较大用户量比较多的时候就可以考虑使用Spring Cloud来搭建分布式微服务的项目。前后端可以由不同的部门来负责，只要根据开发文档来写好对应的功能接口就可以很好的做到前后端协同开发。这样一来开发效率会得到很大的提升，并且出现错误的概率也会大大降低。

## 相关技术介绍

本小节对于设计预约就诊系统所用到的技术进行分析。

### Spring

Spring框架是目前主流的一款Java后端全栈式框架。与我们印象中所熟知的全栈不同，它是指后端所有层次都用Spring自己的技术来实现。使用Spring框架会给我们的开发节省许多时间。利用IOC可以帮助我们的系统解耦，利于后期开发的维护。

### MyBatis

MyBatis是一款持久层框架，是为了提升我们的开发效率而生。如果不使用MyBatis，我们在对数据库进行操作的时候会花费许多时间在创建数据库连接、释放连接、创建Session上。而MyBatis替我们封装好了JDBC，所以我们只需要关注于SQL语句的书写即可。

### Spring Cloud

Spring Cloud是指一个构成微服务框架的集合体。Spring Cloud的主要模块有：Eureka用于服务发现、Feign用于远程服务调用、Hystrix熔断器、Zuul微服务网关、Config分布式配置等。使用Spring Cloud微服务框架构建项目，我们可以按照需求将项目拆分为不同的微服务然后部署在不同的主机上，实现分布式微服务项目。还可以搭建多个Eureka注册中心，配置多个Eureka注册中心的通讯关系即可实现高可用。即使一个注册中心挂掉了还有其他注册中心代替其功能。

## 论文组织结构

本文共分为以下几个章节：

第一章：介绍课题提出的背景以及原因，分析国内外在医院往事预约方面的现状并进行分析。

第二章：对系统所需功能进行分析，对各功能模块进行介绍，并对各功能模块的功能的详细需求进行分析。对网上预约就诊的流程进行分析和介绍，并画出预约流程图。最后介绍了本系统设计过程中会用到的开发技术。

第三章：对系统的体系结构进行设计，划分功能模块并进行设计。根基各功能模块的功能需求来设计与每个相关的数据库表。

第四章：详细描述预约系统各功能的具体实现以及前端页面和后端代码的展示。

第五章：对系统各功能模块进行测试，设计出合适的测试用例对系统进行测试，找出系统中的bug并进行修改，确保系统可以正常运作。

第六章：详细的对预约就诊系统的设计进行总结和分析。找出系统中存在不足的地方，以及提出改进方案。

# 系统需求分析和相关技术分析

在本章对医院预约就诊系统的模块设计和各模块负责的功能进行分析。

## 系统功能模块概述

该小节分析系统模块设计和模块功能分析。

### 功能模块架构

预约就诊系统总共细分为四个功能模块：用户管理模块、医生管理模块、预约时间表管理模块和订单管理模块。用户管理模块包括用户管理功能、验证码发送和验证功能、注册功能和登录功能。医生管理模块包括医生查询功能、医生添加功能、医生批量删除功能以及医生信息修改功能。预约时间表管理模块包括添加预约时间表、预约时间表修改、预约时间表查询。订单管理模块包括所有订单查询功能、批量订单删除功能。系统功能模块图如图2.1所示。

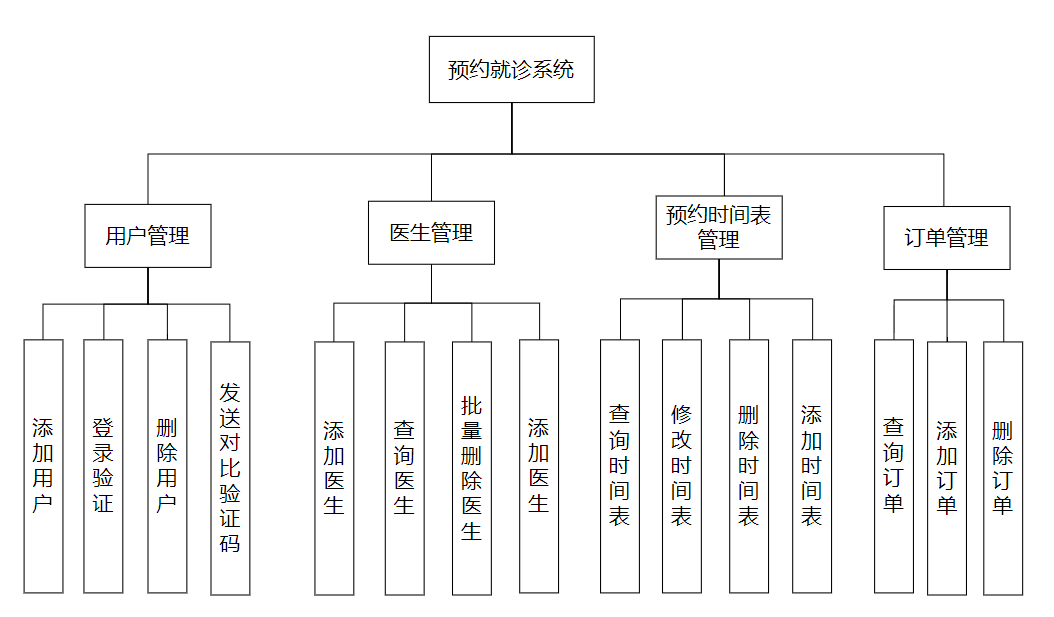


图2.1 系统功能模块图

### 各功能模块结构描述

（1）用户管理模块

用户管理模块主要负责用户信息管理和登录注册功能。首先注册功能的实现需要添加用户功能，添加用户前需要查询该用户是否已经注册，则需要根据手机号查询用户功能。登录功能需要使用到用户的查询功能。

（2）医生管理模块

医生管理模块主要负责数据库医生表的维护。用户点击选择科室后，会跳转进科室详细信息页面。该页面会展示该科室中的所有医生列表，我们需要调用查询该科室所有医生功能返回数据给前端页面。选择医生后，页面上方用来展医生的详细信息，这里需要调用根据医生id查询功能。下方展示医生可用预约时间段。后台医生管理需要对医生表进行删除、添加、修改操作。

（3）预约时间表管理模块

预约时间表管理模块主要负责管理预约时间表，后台管理页面可以选择医生进行操作该医生的时间表。主要操作有添加时间表、修改时间表、修改时间表放号数。

（4）订单管理模块

订单管理模块需要实现的功能有订单表和预约时间表的查询和添加功能。用户预约成功时会向订单表里添加一条新的预约数据，同时并修改预约时间表里的剩余放号数。当用户取消预约时系统会释放号源，删除预约订单的同时修改预约时间表的放号数量。

## 各功能详细需求

该小节通过对业务系统功能的分析，我们可以首先明确我们需要设计出什么功能，以及大致了解功能所需的数据，可以帮助我们进一步构造出系统的大体结构。

### 用户管理模块功能需求

现在网站由于需要防止恶意注册和黄牛卖号，系统需要限制用户使用手机号来进行账号的注册。所以我们设计的注册方式需要绑定手机号来进行注册，一个手机号只允许注册一个账号。用户注册账号时首先需要填写用户和手机号后点击“发送验证码”按钮。用户需要根据收到的短信验证码在输入框内输入短信验证码，最后可以设置账号的登录密码和个人基本信息。

登录功能设计两种当前主流登录方式：使用账号密码登录和账号短信验证码登录。账号密码登录用户只需要输入账号和密码以及图形验证码就可以完成登录。短信验证码登录需要用户输入注册时使用的手机号用于接收短信验证码，后台确认验证码正确即可完成登录。

### 医生管理模块功能需求

预约就诊管理后台可以对医院所有医生进行管理，主要功能有添加医生、删除医生、批量删除医生、修改医生信息、搜索医生。在后台管理页面可以很便捷的查看所有医生并进行管理。

### 预约时间表管理模块功能需求

该模块是用于各科室管理医生可预约时间段以及放号数量。管理人员可依据医生要求灵活分配就诊时间和放数量。可以根据医生id查询医生所有时间表，并且可以修改、添加新的时间表和删除预约时间表。

### 订单管理模块功能需求

用户在主页面选择科室和医生后会展示该医生所有可预约时间段，用户选择时间段后可以选择就诊人，当用户点击“确认预约”按钮后可以完成预约。用户需要查看自己账号下的所有的预约订单，所以还需要根据用户id查询用户id下所有订单的功能。该模块可以使医院更方便的对订单进行管理和统计。

## 预约就诊流程分析

首先用户需要登录账号，在系统主页面里选择自己需要预约的科室。进入科室预约页面，会展示所选科室里所有的医生列表。用户在医生列表页面可以选择自己想要预约的医生。在医生预约页面会显示医生个人简介帮助用户了解医生信息，页面会显示医生所有可预约的时间段，用户选择自己合适的时间来预约。进入生成订预约单页面，用户需要选择给哪位患者进行预约，如果还没有保存就诊人信息，则需要添加就诊人信息。点击确认预约按钮后会生成订单，并修改数据库里的数据。预约就诊流程如图2.2所示。

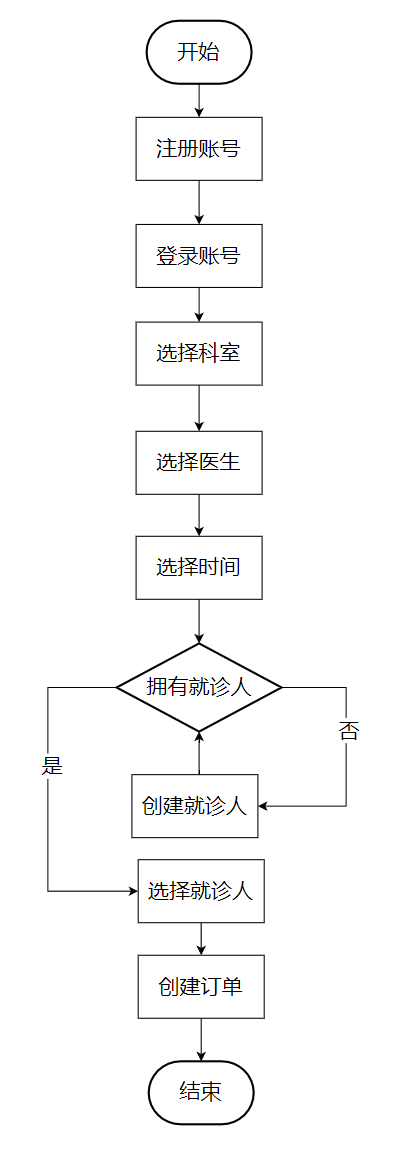


图2.2 预约流程图

## 本章小结

本章首先对系统各个功能模块进行了分类，并对每一个模块的功能进行了描述。其次对每一个功能模块的功能需求进行了分析，初步拟定了系统需要实现的几个主要功能。还对网上预约就诊的流程进行了分析，让预约流程所需要的各个功能进行了补充。最后对系统将使用到的主要技术进行了分析及解释为什么要用到这个技术和用到这个技术的优点。

# 系统设计

本章是对系统体系机构和功能模块进行设计，还设计了微服务系统的组成结构，对数据库概念和逻辑进行设计。通过本章设计出预约就诊系统的雏形。

## 系统体系结构

本系统是前后端分离的项目，前端用于展示数据后端用于处理前端发送过来的请求。不同的请求由不同的模块负责，后端分别有订单管理模块、用户管理模块、预约时间表管理模块和医生管理模块。各个微服务为Eureka客户端，当需要调用其他微服务的方法时，需要向Eureka服务器查询该微服务的主机IP地址和端口等信息。系统体系结构如图3.1所示。

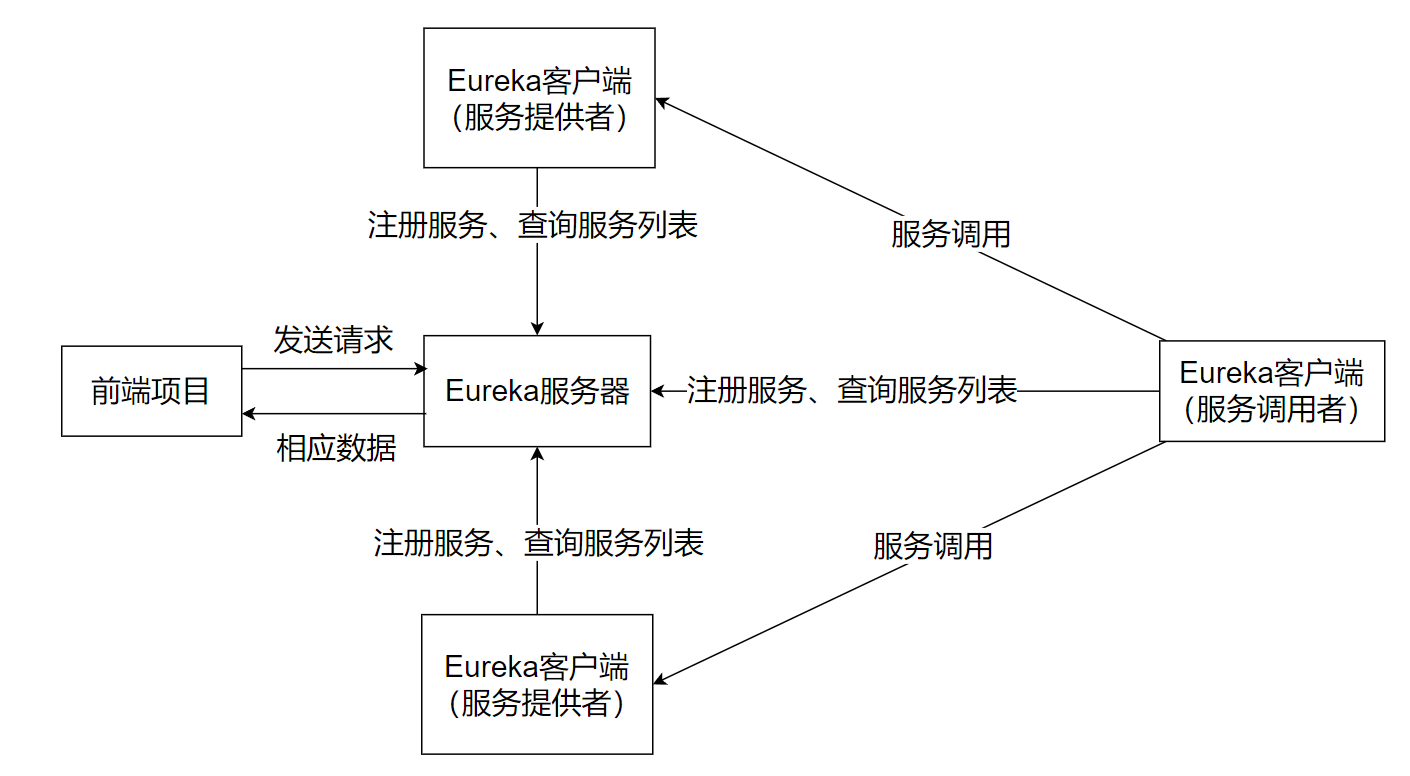


图3.1 系统体系结构图

## 功能模块设计

根据之前的系统需求分析，可以将系统分为四个功能模块：用户管理模块、医生管理模块、预约时间表管理模块和预约订单管理模块。

用户管理功能模块中包括注册添加用户功能、发送和对比短信验证码功能、登录用户验证功能；

医生管理包括添加医生、查询医生信息、批量删除医生和修改医生信息。

预约时间表管理模块包括：查询医生的预约时间表、修改预约时间表、删除预约时间表、添加预约时间表；

订单管理模块包括：查询订单、添加订单和删除订单。

## 数据库概念设计

数据库概念设计是指对各个不同类别的实体类进行设计，具体描述每种实体的属性和各个实体之间的关联。

（1）用户

如图3.2为普通用户的E-R图，用户实例中包括了：编号、手机号、密码、昵称、账号状态和用户角色。

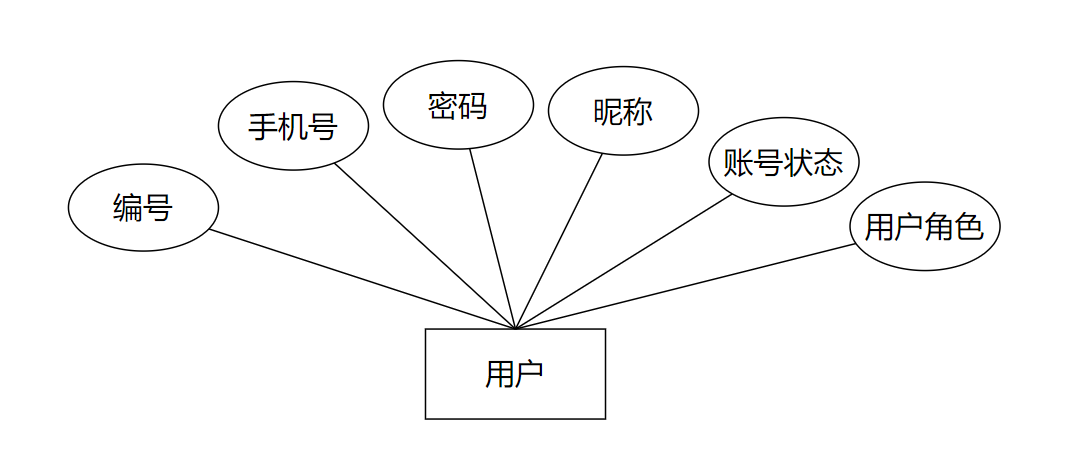


图3.2 用户E-R图

（2）医生

如图3.3为医生的E-R图。医生实例不仅包括了个人基本信息如：姓名、年龄、性别和电话号码，还包括了部门号、职位、擅长和个人简介，来帮助用户进一步了解医生。

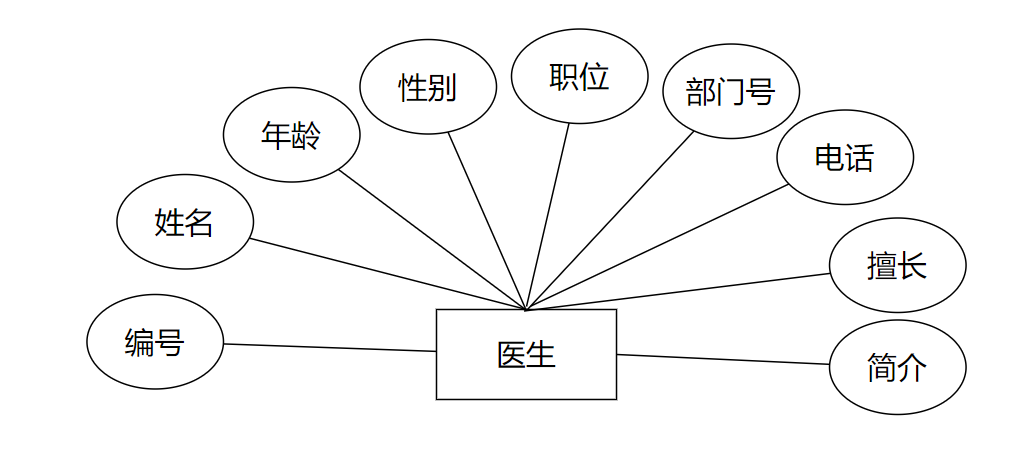


图3.3 医生E-R图

（3）管理员

管理员E-R图如图3.4所示。管理员实例包括了编号、用户名、密码、昵称和账号状态。

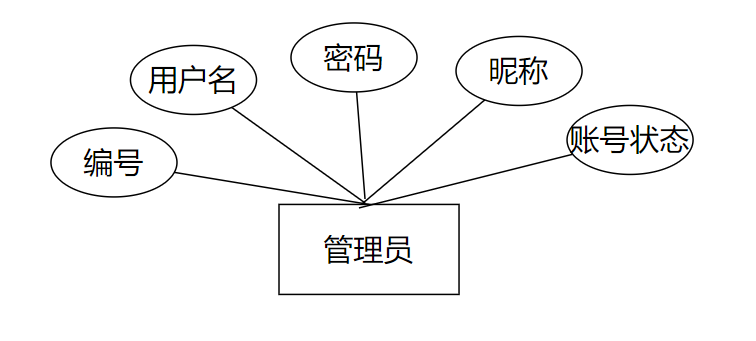


图3.4 管理员E-R图

（4）就诊人

图3.5为就诊人E-R图。就诊人实例包含了编号、姓名、性别、身份证号、身份证号码和用户编号。

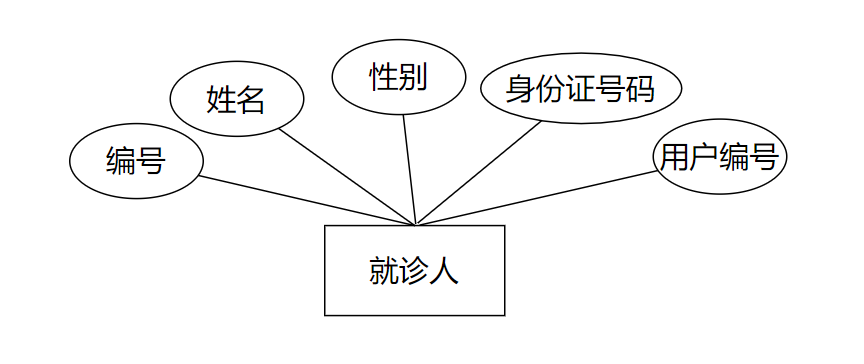


图3.5 就诊人E-R图

（5）一级科室

图3.6是一级科室的E-R图。一级科室实例包括了编号和科室名称。

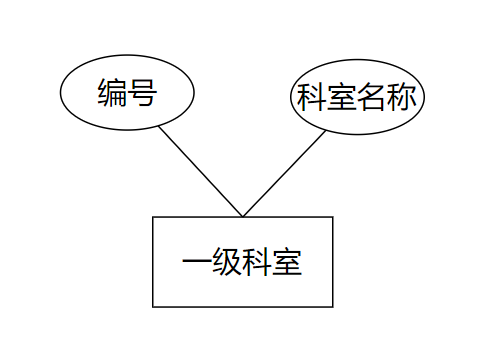


图3.6 一级科室E-R图

（6）二级科室

图3.7是二级科室的E-R图。二级科室不仅包括了编号和科室名称，还添加了一号科室的编号。

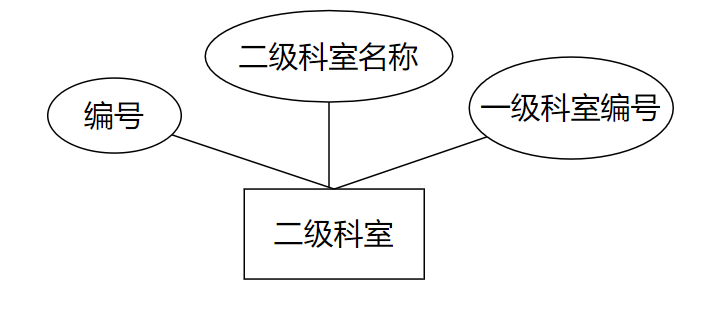


图3.7 二级科室E-R图

（7）时间表

时间表E-R图如图3.8所示。时间表实例包括时间表编号、医生编号、放号数量和时间表日期。

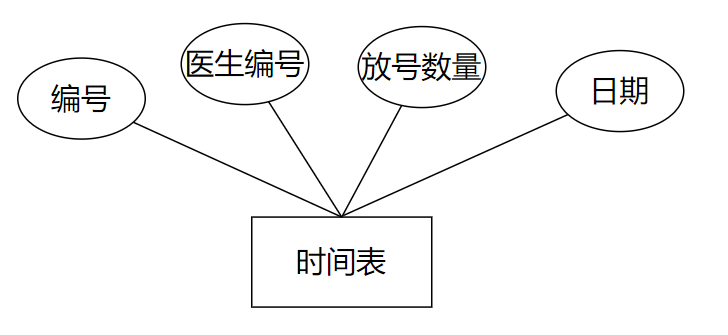


图3.8 时间表E-R图

（8）订单

订单E-R图如图3.9所示。订单实例包括编号、用户编号、就诊人编号和时间表编号。

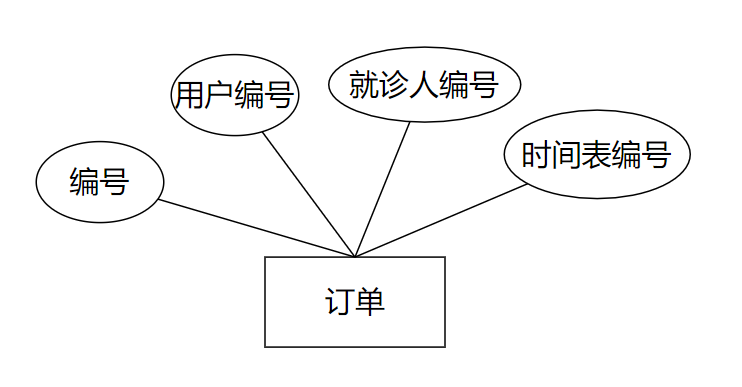


图3.9 订单E-R图

（9）系统整体E-R图如图3.10所示。

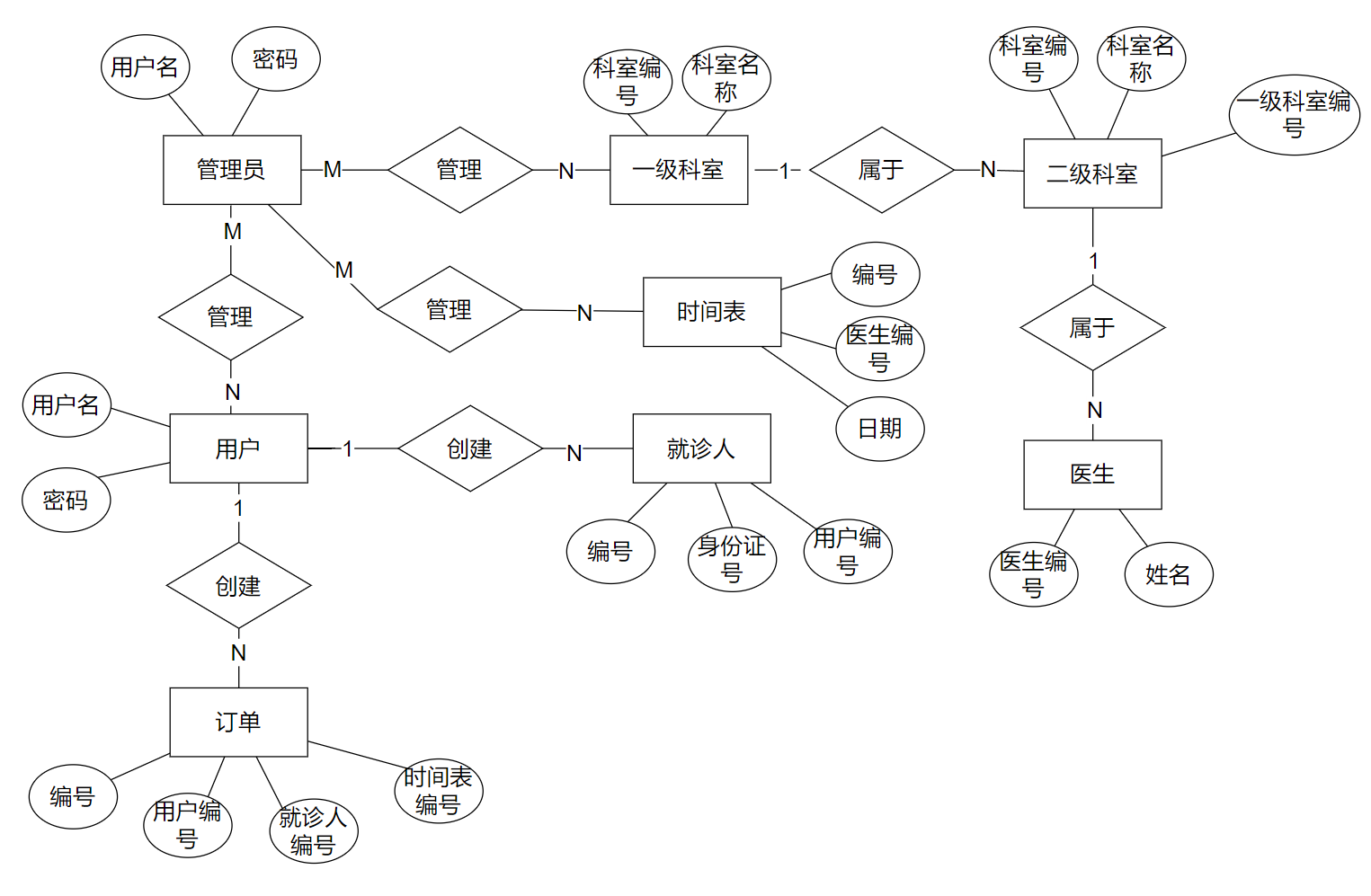


图3.10 系统整体E-R图

## 数据库逻辑设计

基于上一章系统需求分析，我们可以根据需求创建所需的数据库表。具体设计有用户管理模块：用户表、管理员表、就诊人表；医生管理模块：医生表、一级科室表、二级科室表；预约时间表模块：预约时间表；订单管理模块：订单表；

（1）用户表

用户表是用来存储注册用户账号信息的表。其中主要存储用户id、昵称、手机号、登录密码、账号状态。登录和注册需要通过请求查询和修改用户表。如表3.1所示。

表3.1 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 手机号 | phone | varchar(11) | NO |
| 密码 | password | varchar(32) | NO |
| 昵称 | nickname | varchar(32) | NO |
| 账号状态 | status | int(1) | NO |

（2）管理员表

管理员表不允许网页直接修改，只允许通过修改数据库来修改数据。管理页面只允许登录过管理员账号的进行管理操作。如表3.2所示。

表3.2 管理员表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 用户名 | username | varchar(16) | NO |
| 密码 | password | varchar(32) | NO |
| 昵称 | nickname | varchar(32) | NO |
| 账号状态 | status | int(1) | NO |

（3）就诊人表

就诊人表用于维护创建订单时选择的就诊人，一个用户账号下可以创建多个就诊人。其中就诊人的userID绑定用户表的主键id作为外键来使用。如表3.3所示。

表3.3 就诊人表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 用户名 | name | varchar(16) | NO |
| 密码 | gender | varchar(32) | NO |
| 身份证号码 | id\_number | varchar(18) | NO |
| 用户id | user\_id | int(11) | NO |

（4）一级科室表

一级科室是比较笼统的科室，例如内科是一级科室，内科下面的二级科室还有神经内科、消化内科、心血管内科和呼吸科等。科室的一二级分类可以方便用户的查找和分类管理。一级科室表如表3.4所示。

表3.4 一级科室表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 科室名称 | department\_name | varchar(16) | NO |

（5）二级科室表

二级科室隶属于一级科室，所以一级科室的主键作为二级科室的外键进行约束。二级科室表如表3.5所示。

表3.5 二级科室表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 科室名称 | department\_name | varchar(16) | NO |
| 一级科室id | first\_department | int(16) | NO |

（6）医生表

医生表主要存储医生的个人信息，用户在选择医生的时候需要了解每个医师的个人信息，例如医生年龄、擅长领域和职位都是帮助用户选择医生的条件。录入这些信息可以更好的让用户来了解信息。医生表如表3.6所示。

表3.6 医生表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 姓名 | name | varchar(16) | NO |
| 年龄 | age | int(8) | NO |
| 性别 | gender | varchar(1) | NO |
| 职位 | position | varchar(16) | NO |
| 科室id | department\_id | int(16) | NO |
| 手机号码 | phone | varchar(11) | NO |
| 擅长 | major | varchar(256) | YES |
| 医生简介 | doctor\_desc | varchar(256) | YES |

（7）预约时间表

预约时间表是用于存储医生的时间表的。用户选择预约时间时会查询该表中的数据，根据医生的id进行查询所有订单表。本表中可以设置时间和每个时间段的放号数量。如表3.7所示。

表3.7 预约时间表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 医生id | doctor\_id | int(8) | NO |
| 放号数 | permit | int(16) | NO |
| 预约日期 | date\_time | date | NO |

（8）订单表

订单表中主要存储的是用户信息、时间表id和就诊人id。每当成功生成订单的时候和取消订单时会修改本表的数据。如表3.8所示。

表3.8 订单表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 字段名 | 数据类型 | 可为空 |
| 编号 | id | int(16) | NO |
| 用户id | user\_id | varchar(16) | NO |
| 就诊人id | patient\_id | int(16) | NO |
| 时间表id | schedule\_id | int(16) | NO |

## 微服务架构设计

首先项目所有内容都由一个父工程所包含，在夫工程中引入了很多Spring Cloud未来会用到的依赖，这样以后需要的时候不用在导入依赖。common模块是用来存放一些用到的工具类。common-db是数据访问模块，该模块中主要由common模块、mysql-connector-java、druid、my-batis等依赖组成，当一个微服务需要访问数据库的时候，就可以依赖本模块来实现。Eureka模块是微服务的注册中心，通过配置yml文件和写好启动类，就可以创建好我们自己的注册中心。service模块主要是所有微服务的父模块，为了方便项目管理，所有微服务都创建在该模块中。每一个微服务模块中都有controller、dao、和service包。Service-api模块主要存放每个微服务会用到的pojo和需要使用到的feign接口。同时每一个service都依赖了其对应的api，如果需要调用其他微服务的feign接口也需要依赖其对应的service-api。微服务系统架构设计图如图3.11所示。

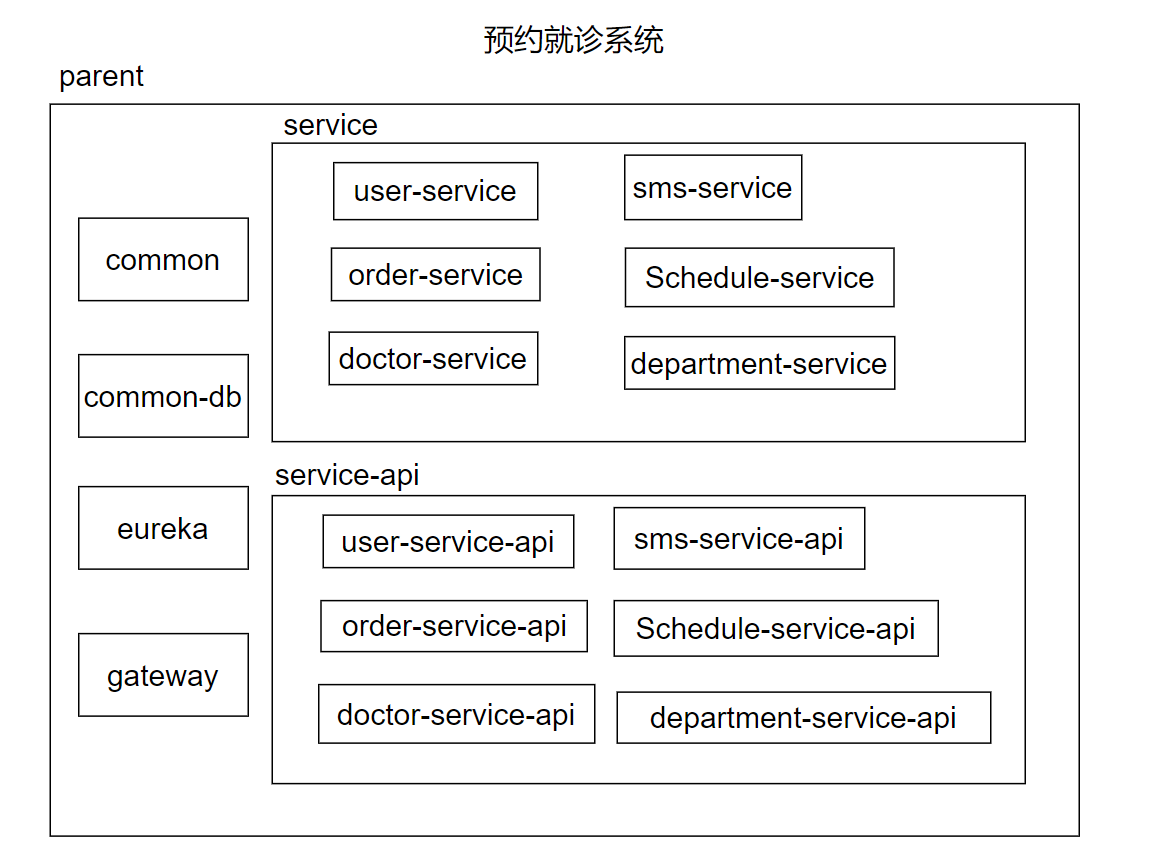


图3.11 微服务架构设计图

## 本章小结

本章主要对系统整体架构进行设计，明确各个模块之间的关系和功能调用。功能模块的设计可以更加明确各模块的功能设计。对各功能的世界可以引出各功能模块负责的数据库表的设计，对所需功能的字段进行设计。

# 系统实现

本章主要描述系统各个功能模块的具体实现流程。对各个功能的核心代码进行讲解和展示并对数据流的处理过程进行了说明。前端页面的实现也进行了详细的说明，以及数据格式的转换思路和具体代码实现。

## 前台模块的实现

本小节主要描述预约系统各个功能模块的主要功能的实现。

### 注册

后端代码执行流程以本功能为例进行介绍，其他功能的实现与之类似，所及不再一一赘述。

用户首先需要输入手机号码，前端会对输入的手机号进行验证。如果符合要求在点击“获取验证码”按钮来获取验证码后会成功发送验证码，否则提醒电话号码格式不正确。然后用户可以设置登录密码和用户昵称。点击注册按钮后，后台会对手机号进行校验查看是否已经存在该账号，如果已经存在会返回错误信息。如果查询到该手机号还未注册，则校验用户输入的验证码是否正确。如果验证码正确则完成注册。注册页面如图4.1所示。

图形用户界面, 文本, 应用程序

描述已自动生成

图4.1 注册

前端页面注册用到的表单使用ElementUI的el-form表单进行实现，将表单数据和myform对象进行双向绑定，当修改页面数据的时候myform的数据也会随之变化。使用@click将register方法绑定到“注册”按钮的click方法上，将getCode()方法绑定到“获取短信验证码”按钮上。这样当点击按钮时就能触发给它绑定的方法。具体前端代码如下：

<el-form ref="myform" :model="myform" label-position="right" ref="registerForm" label-width="100px"

class="demo-ruleForm">

<h1>注册账号</h1>

<el-form-item label="手机号">

<el-input v-model="myform.phone" placeholder="请输入你的手机号码" clearable style="width:300px;"></el-input>

</el-form-item>

<el-popover placement="top-start" title="提示" width="200" trigger="hover"

content="登录验证码短信将发送到你的手机上，请查看手机短信并输入验证码。">

<el-button type="primary" plain slot="reference" @click="getCode">获取验证码</el-button>

</el-popover>

<el-form-item label="验证码">

<el-input placeholder="请输入短信验证码" v-model="myform.code" clearable style="width: 300px;"></el-input>

</el-form-item>

<el-form-item label="设置密码">

<el-input placeholder="请设置你的密码" v-model="myform.password" show-password clearable style="width: 300px;">

</el-input>

</el-form-item>

<el-form-item label="设置昵称">

<el-input placeholder="请设置你的昵称" v-model="myform.nickname" clearable style="width: 300px;"></el-input>

</el-form-item>

<el-button type="primary" @click="register">注册</el-button>

<el-button @click="gologin">去登录</el-button>

</el-form>

在后台中注册请求会到达UserController的register()方法，方法中使用@RequestBody注解来接收请求的请求体中的数据给我们写好的形参Map。对map对象使用get()方法即可根据键值来取走要用的数据。

首先要取出手机号来查看数据库中是否已经存在该账号，如果存在则返回错误并携带错误信息“该手机号已注册”。如果数据库中没有该用户则继续执行下面的流程。具体代码如下:

String phone = (String) map.get("phone");

User byPhone = userService.findByPhone(phone);

if (byPhone != null){

return new Result(false, 301, "该手机号已注册", null);

}

然后从Redis中取出发送验证码赋值给sysCode变量，再拿sysCode与用户输入的验证码进行对比，如果相同则进行后续创建操作，如果验证码错误则返回错误信息“验证码错误”。具体代码如下：

String sysCode = (String) redisTemplate.boundHashOps("smscode").get(phone);

if (sysCode == null && !sysCode.equals(code)) {

System.out.println("验证码错误");

return new Result(true, 300, "验证码错误", null);

}

创建用户操作，首先先新创建一个用户对象，将用户输入的数据复制给新用户对象。由于Service层的UserService继承了IService，所以可以直接调用UserService的save方法，传入参数为user。具体代码如下:

User user = new User();

user.setPhone(phone);

user.setPassword(password);

user.setNickname(nickname);

user.setRole(0);

user.setStatus(1);

userService.save(user);

return new Result(true,200,"注册成功，即将为你跳转到登录页面",null);

### 密码登录

用户想要进入主页面使用预约功能需要先登录，用户可以选择账号密码登录或者短信验证码登录。如图4.2所示。

图形用户界面, 文本, 应用程序

描述已自动生成

图4.2 密码登录

当用户进入或者刷新页面的时候页面会触发钩子函数changeCode()进行调用后台的getBaseCaptcha()方法，该方法会随机产生一个4位的验证码并生成图像返回给页面，并将验证码存入session中，方便后续取出验证。具体代码如下：

HttpSession session=request.getSession();

Object[] objs = RandomValidateCodeUtil.createMumAndCharTwo();

//将验证码存入Session

session.setAttribute("imageCode",objs[0]);

//将图片输出给浏览器

BufferedImage image = (BufferedImage) objs[1];

response.setContentType("image/png");

OutputStream os = response.getOutputStream();

ImageIO.write(image, "png", os);

用户点击登录时会调用userlogin()方法，页面会先调用checkCode()方法检验验证码是否正确如果正确则将passCodeCheck置为true否则则提示“验证码错误”。具体代码如下：

checkCode() {

if (this.code == "" || this.code.trim() == "") {

this.$message({

message: '验证码不能为空',

type: 'warning'

});

} else {

this.msg = '';

// 发送请求判断验证码是否正确

axios.post('http://localhost:9001/user/checkImgCode', {

code: this.code

}).then(response => {

if (response.data.code == 200) {

console.log(true);

this.passCodeCheck = true;

} else {

this.$message({

message: '验证码错误',

type: 'warning'

});

}

})

}

}

后端接收到手机号和输入的验证码后会取出存在session中的imageCode和用户输入的验证码code进行对比，如果相同返回true否则返回false。具体代码如下：

@PostMapping("/checkImgCode")

public Result checkImgCode(@RequestBody Map map,HttpServletRequest request){

HttpSession session = request.getSession();

String imageCode = (String) session.getAttribute("imageCode");

String code = (String) map.get("code");

System.out.println(code);

System.out.println(imageCode);

if (code.equalsIgnoreCase(imageCode)){

return new Result(true,200,"验证码正确");

}else {

return new Result(false,300,"验证码错误");

}

}

只有图形验证码校验通过后才会发送登录请求，将手机号和密码传入后端的login()方法。调用userService的findByPhone()方法查询用户，取出数据库中的密码在与用户输入的进行对比来判断是否允许用户登录，具体代码如下：

String phone = loginuser.getPhone();

String password = loginuser.getPassword();

User user = userService.findByPhone(phone);

if (user != null && user.getPassword().equals(password)) {

System.out.println("登陆成功");

return new Result(true, 200, "登录成功", null);

} else {

System.out.println("登陆失败");

return new Result(false, 300, "账号或密码错误", null);

}

### 发送短信验证码

前端传入手机号后, 使用PhoneFormatCheckUtils工具类的isPhoneLegal()方法校验手机号是否规范，如果校验通过后台就会自动产生一条六位数的短信验证码。把即将发出的短信存入消息队列中,并把手机号和短信验证码存入Redis中。发送短信时微服务会先提取消息队列中的验证码,并调用阿里云购买的短信发送端口完成手机验证码的发送。具体代码如下：

//检验手机号是否正确

if (!PhoneFormatCheckUtils.isPhoneLegal(phone)) {

return new Result(false, StatusCode.ERROR, "手机号格式不正确");

}

//生成6位随机数

String code = (long) (Math.random() \* 1000000) + "";

//将验证码存入Redis缓存

redisTemplate.boundHashOps("smscode").put(phone, code);

//发送到RabbitMQ

Map map = new HashMap();

map.put("mobile", phone);

map.put("code", code);

rabbitTemplate.convertAndSend("axx.sms.queue", map);

RabbitMQ监听端会持续监测消息队列中的消息，当有新消息到达时会触发监听端的getMessage()方法。getMessage()方法会取出map中的手机号和验证码，并调用短信发送接口sendSms()进行短信发送。具体代码如下：

String mobile = map.get("mobile");

String code = map.get("code");

// 发送短信

smsUtil.sendSms(mobile,code);

### 短信验证码登录

前台用户输入手机号phone和短信验证码code传给后台进行校验，后台从Redis中取出验证码sysCode与用户输入的验证码code进行比较，如果相同则校验通过，如果不同则提示用户输入错误。具体代码实现如下：

String sysCode = (String) redisTemplate.boundHashOps("smscode").get(phone);

if (sysCode != null && sysCode.equals(code)) {……}

### 科室展示

用户进入主页面会展示科室列表，用户可以选择一级科室查看其下级所有科室。根据科室分级，用户可以更快的找到自己要预约的科室。这里为了方便用户查看科室信息使用了ElementUI的el-cascader-panel级联选择器。由于该组件对数据格式有限制，所以在查询完一级科室和二级科室信息后需要对格式进行调整。具体代码如下：

// 获取一级id列表

List<DepartmentFirst> all = departmentFirstService.findAll();

ArrayList<Integer> ids = new ArrayList<>();

for (DepartmentFirst departmentFirst : all) {

ids.add(departmentFirst.getId());

}

// 查询二级科室

ArrayList<List<DepartmentSecond>> departmentSeconds = new ArrayList<>();

for (Integer id : ids) {

List<DepartmentSecond> listById = departmentSecondService.findListById(id);

departmentSeconds.add(listById);

}

ArrayList<Object> objects = new ArrayList<>();

for (int i = 0; i < all.size(); i++) {

HashMap<String, Object> map = new HashMap<>();

map.put("value",all.get(i).getId());

map.put("label",all.get(i).getDepartmentName());

ArrayList<Object> children = new ArrayList<>();

for (DepartmentSecond departmentSecond : departmentSeconds.get(i)) {

HashMap<String, Object> department = new HashMap<>();

department.put("value",departmentSecond.getId());

department.put("label",departmentSecond.getDepartmentName());

children.add(department);

}

map.put("children",children);

objects.add(map);

}

首先查询所有一级科室的并用all 接受，然后遍历all ArrayList拿出每一个一级科室的id放入ids中。然后再根据每一个id查询其下的二级科室列表并存入departmentSeconds。然后就可以根据要求格式将数据转化为:

options[{

value: 'zhinan',

label: '指南',

children: [{

value: '',

label: ' '}]}]

页面展示效果如图4.3所示。

一些文字和图片的手机截图

描述已自动生成

图4.3 科室展示

### 医生列表

点进科室页面会展示该科室的所有医生和简介信息。用户可以选择自己想要预约的医生。点击“预约”按钮即可进入预约界面。医生列表的展示使用了ElementUI的el-table，只需要将医生列表数据复制给tableData并与el-table进行双向绑定即可。具体代码如下：

<el-table :data="doctors" @selection-change="handleSelectionChange" size="medium" highlight-current-row="true"

style="width: 100%">

<el-table-column type="selection" width="55" prop="uid">

</el-table-column>

<el-table-column width="50px" label="序号" type="index">

</el-table-column>

<el-table-column label="医生姓名" prop="name">

</el-table-column>

</el-table-column>

<el-table-column label="年龄" prop="age">

</el-table-column>

</el-table>

页面展示效果如图4.4所示。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图4.4 医生列表

### 创建预约订单

用户创建预约订单时需要选择就诊人。一个用户可以创建多个就诊人。当创建订单需要选择一个就诊人进行预约。创建订单如图4.5所示。

图形用户界面, 文本, 应用程序

描述已自动生成

图4.5 创建预约订单

创建订单时需要同时操作order表和schedule表两个表，在order表里添加一条预约记录的同时需要修改时间表的剩余号数减1，所以需要在方法上使用SpringBoot注解@Transactional来开启事务。具体代码实现如下：

@PostMapping

@Transactional

public Result add(@ApiParam(name = "Order对象",value = "传入JSON数据",required = true) Order order){

//调用OrderService实现添加Order

try {

//调用OrderService实现添加Order

orderService.add(order);

// 余量减一

scheduleService.reduceSchedule(String.valueOf(order.getScheduleId()));

} catch (Exception e) {

e.printStackTrace();

return new Result(false,300,"失败");

}

return new Result(true,200,"添加成功");

}

其中reduceSchedule()是需要手动编写的方法，需要传入的参数是时间表的id，方法会判断是否大于0并且对剩余放号量进行减1的操作。具体代码实现如下：

@Update("UPDATE schedule SET permit=permit-1 WHERE id = #{scheduleId} AND permit>= 1 ")

int reduceSchedule(@Param("scheduleId") String id);

## 后台管理模块的实现

本小节主要描述后台管理功能模块的主要功能的实现。后台管理模块主要包括医师管理和医师排班管理。

### 医师管理

主要是对医师表进行维护。可以进行对医师的基本操作如：修改医师信息、添加和删除医师。使用的也是ElementUI的el-table进行数据的展示。前台页面如图4.6所示。

图形用户界面, 应用程序

描述已自动生成

图4.6 医师管理

（1）展示医师列表

当用户进入页面的时候会触发钩子函数findAllDoctor()，前端会发起请求查询医师信息。并且会携带参数currentPage和pageSize，分别代表当前页和每一页的数据量。currentPage默认为第一页，pageSize大小默认为5。用户点击页面或者前进后退按钮可以调整当前页，点击每页条数选项卡可以选择每一页的展示条数。这两个数据改变后都会触发findAllDoctor()方法来刷新页面数据。具体代码实现如下：

findAllDoctor() {

let str = "http://localhost:9003/doctor/search/" +

this.currentPage + "/" + this.pageSize;

axios.post(str, {

name: this.search

}).then(response => {

//为表格展示的用户数组赋值

this.tableData = response.data.data.rows;

//为总数据量赋值

this.totalCount = response.data.data.total;

})

}

//每页数据量发生改变的方法

handleSizeChange: function (size) {

this.pageSize = size; //先改变每页数据量的值

this.findAllDoctor(); //重新调用获取用户的方法

}

//当前页码发生改变的方法

handleCurrentChange: function (currentPage) {

this.currentPage = currentPage; //先改变当前页码的值

this.findAllDoctor(); //重新调用获取用户的方法

}

后台接收查询医生的请求的方法是findPage，取出请求中的参数并调用doctorService的findPage方法进行查询。返回的PageResult<Doctor>给页面进行数据的展示。具体代码实现如下：

@PostMapping(value = "/search/{page}/{size}" )

public Result<PageResult<Doctor>> findPage(@RequestBody(required = false) @ApiParam(name = "Doctor对象",value = "传入JSON数据",required = false) Doctor doctor, @PathVariable int page, @PathVariable int size){

//调用DoctorService实现分页条件查询Doctor

PageResult<Doctor> pageResult = doctorService.findPage(doctor, page, size);

return new Result(true,StatusCode.OK,"查询成功",pageResult);

}

（2）添加医师

输入医师的姓名、年龄、性别登信息后可以完成医师的添加。必填项为姓名、年龄和手机号。后端接收到传入的doctor对象后会调用doctorService.add(doctor)方法进行添加。具体代码如下：

@PostMapping

public Result add(@RequestBody @ApiParam(name = "Doctor对象",value = "传入JSON数据",required = true) Doctor doctor){

//调用DoctorService实现添加Doctor

try {

doctorService.add(doctor);

} catch (Exception e) {

e.printStackTrace();

return new Result(true,300,"添加失败");

}

return new Result(true,200,"添加成功");

}

（3）修改医师

管理员可以点击“修改”按钮可以修改当前行的医师信息。当点击“详情”按钮时，会先读取当前行医师的信息存入临时对象，然后复制给新弹出的对话框，这样新弹出的对话框中就会显示医师的现有信息，方便我们进行修改操作。具体代码如下：

//点击修改按钮的方法

handleEdit(index, row) {

//出现修改的对话框

this.dialogVisible = true;

//赋值数据源

this.ruleForm.id = row.id;

this.ruleForm.name = row.name;

this.ruleForm.age = row.age;

this.ruleForm.gender = row.gender;

this.ruleForm.position = row.position;

this.ruleForm.departmentId = row.departmentId;

this.ruleForm.phone = row.phone;

this.ruleForm.major = row.major;

this.ruleForm.doctorDesc = row.doctorDesc;

}

修改完信息后用户点击确定按钮会触发submitForm()方法，并将表单中的数据作为形参传入方法中，方法调用axios发送post请求将医师信息发送给后端。后端使用doctorService的update方法更新数据库。具体代码实现如下：

@PostMapping(value="/update")

public Result update(@RequestBody @ApiParam(name = "Doctor对象",value = "传入JSON数据",required = false) Doctor doctor){

//调用DoctorService实现修改Doctor

doctorService.update(doctor);

return new Result(true,200,"修改成功");

}

（4）查看详情

点击“详情”按钮时，会记录当前行医师的个人信息，这样打开新的对话框时将信息赋值给ruleForm就可以显示医师的详细信息了。具体代码实现如下：

//点击详情按钮的方法 index下标，row就是当前这一行中的数据！

handlelook(index, row) {

//出现详情的对话框

this.dialogVisible1 = true;

//赋值数据源

this.ruleForm = row;

this.dialogDepartment = this.secondDepartments[row.departmentId -1 ].departmentName;

}

（5）删除医师

用户可以在列表多选框处选择一个或者多个医生进行删除操作。点击“删除”后系统会弹窗询问“是否确认要删除所选中的医师”以防止误操作。继续点击“确认”后会进行删除操作。

删除请求由于前端会将要删除的医生id放入delarr数组中然后拼接到请求url中，后端接收到请求后要对参数进行拆分处理成字符串数组。之后可以遍历字符串数组，依次取出每一个id进行删除操作。具体代码实现如下：

@DeleteMapping(value = "/deleteByIds" )

public Result deleteByIds(@RequestParam String ids){

try{

String[] delids = ids.split(",");

for (String id : delids) {

doctorService.delete(Integer.valueOf(id));

}

} catch (NumberFormatException e) {

e.printStackTrace();

return new Result(false,300,"删除失败");

}

return new Result(true,200,"删除成功");

}

### 医师排班

医师排班管理可以对医师时间表进行操作。基本可用操作和医师管理基本一致。如图4.7所示。



图4.7 排班管理

（1）添加时间表

用户需要选择要进行操作的医师，然后输入剩余可用号、预约时间等信息。后端接收到请求后对参数进行解析。然后调用时间表scheduleService的add方法进行添加操作。具体代码实现如下:

@PostMapping

public Result add(@RequestBody @ApiParam(name = "Schedule对象",value = "传入JSON数据",required = true) Schedule schedule){

//调用ScheduleService实现添加Schedule

try {

scheduleService.add(schedule);

} catch (Exception e) {

e.printStackTrace();

return new Result(false,300,"添加失败");

}

return new Result(true,200,"添加成功");

}

/\*\*

\* 修改Schedule

\*

\* @param schedule

\*/

@Override

public void update(Schedule schedule) {

this.updateById(schedule);

}

（2）修改时间表

管理员可以点击每一行的“修改”按钮来修改当前行的时间表。前台发送请求时携带当前时间表的id方便我们后端进行修改。后端将id赋值给接收到的schedule对象，然后调用scheduleService的update()方法进行数据跟新。具体代码实现如下：

@PutMapping(value="/{id}")

public Result update(@RequestBody Schedule schedule,@PathVariable Integer id){

//设置主键值

schedule.setId(id);

//调用ScheduleService实现修改Schedule

scheduleService.update(schedule);

return new Result(true,StatusCode.OK,"修改成功");

}

（3）删除时间表

删除时间表时会根据时间表的id进行查询是否存在该时间表的订单，如果存在则无法删除，如果不存在则可以进行删除操作。具体代码实现如下：

@DeleteMapping(value = "/{id}" )

public Result delete(@PathVariable Integer id){

//创建订单对象并设置订单预约表id

Order order = new Order();

order.setScheduleId(id);

//根据id查询是否存在相关订单

List<Order> list = orderService.findList(order);

if (list == null){

//调用ScheduleService实现根据主键删除

scheduleService.delete(id);

## 本章小结

对各功能模块的各功能进行了代码上的实现，并可以完成预定的功能。前端页面展示实现了科室的二级联动菜单的展示、按照科室来查询医生列表、查询医生名下所有时间表等功能。后端管理实现了医生的管理页面的展示和各个操作数据库的实现和时间表管理的各个功能的代码实现。逐一展示了后端主要代码的编写和阐述了各个微服务之间方法的调用。

# 系统测试

系统测试是为了能够发现在开发过程中系统中存在的问题并及时解决，并对系统中的功能进行测试。本章会对测试环境、测试方案、测试结果及测试分析进行详细论述。系统测试的目的是用来测试来确保是否我们的预约就诊系统能否正常的运行。

## 测试环境配置

系统测试所需的服务器的部署环境和测试环境，分别如图5.1和图5.2所示。

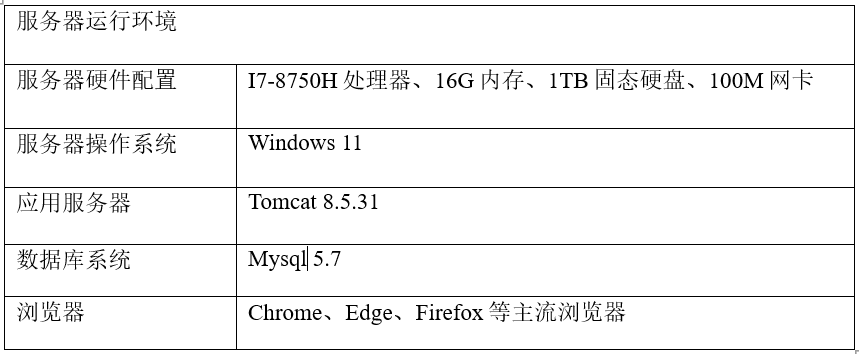


图5.1 部署环境



图5.2 测试环境

## 测试方案

（1）测试目的

根据系统的功能设计和功模块进行测试各功能的完成情况，及异常情况下系统能否正常处理异常。

（2）测试范围

本次测试的范围为系统在预约流程上所需要用到的基础功能如登录、注册、预约和后台管理等主要功能。

（3）测试方法

测试方法为人工使用本系统进行测试，并对测试结果进行记录。

## 测试过程

医院网上预约就诊系统在预约流程用到的基本功能包括注册、登录、浏览信息、预约医生。

设计用户注册的测试用例，来验证注册功能是否能够成功完成新用户的注册，新注册的用户的数据能否添加进数据库中。如表5.1所示。

表5.1 用户注册测试用例

|  |  |
| --- | --- |
| 名称 | 内容 |
| 编号 | Test01 |
| 名称 | 用户注册 |
| 前提 | P1、Tomcat服务器已启动 |
| 操作步骤 | S1、输入手机号码后接收短信验证码  S2、在输入框输入验证码  S3、设置密码  S4、设置昵称  S5、点击注册 |
| 期望结果 | 1、页面提示注册成功  2、转跳到登录页面  3、数据库中有注册信息  4、可以使用注册的账号登录 |
| 实际结果 | 符合期望 |

设计用户登录的测试用例，测试使用账号能否正常登录。如表5.2所示。

表5.2 用户登录测试用例

|  |  |
| --- | --- |
| 名称 | 内容 |
| 编号 | Test02 |
| 名称 | 用户登录 |
| 前提 | P1、Tomcat服务器已启动  P2、用户已注册 |
| 操作步骤 | S1、输入手机号码后接收短信验证码  S2、在输入框输入验证码  S3、登录 |
| 期望结果 | 1、页面提示登录成功  2、转跳到主页面 |
| 实际结果 | 符合期望 |

设计用户进行预约就诊的流程的测试用例，测试用户在登录过后是否可以正常进行预约。如表5.3所示。

表5.3 预约挂号测试用例

|  |  |  |
| --- | --- | --- |
| 名称 | 内容 | |
| 编号 | | Test03 |
| 名称 | | 预约挂号 |
| 前提 | | P1、Tomcat服务器已启动  P2、用户已登录 |
| 操作步骤 | | S1、选择科室  S2、选择医生  S3、选择时间  S4、选择就诊人  S5、预约 |
| 期望结果 | | 1、页面提示预约成功  2、数据库中订单表和时间表同时修改成功 |
| 实际结果 | | 符合期望 |

设计医师管理的测试用例，测试后台管理页面医师管理功能模块各功能是否可以正常完成。如表5.4所示。

表5.4 医生管理测试用例

|  |  |  |
| --- | --- | --- |
| 名称 | 内容 | |
| 编号 | | Test04 |
| 名称 | | 医生管理 |
| 前提 | | P1、Tomcat服务器已启动  P2、管理员账号已登录 |
| 操作步骤 | | S1、查看所有医生  S2、查看医生详细信息  S3、修改医生详细信息  S4、添加医生  S5、删除医生 |
| 期望结果 | | 1、各操作均能显示操作成功  2、数据库能被成功修改 |
| 实际结果 | | 符合期望 |

设计时间表管理的测试用例，测试时间表管理各项功能是否可以正常使用。如表5.5所示。

表5.5 时间表管理测试用例

|  |  |  |
| --- | --- | --- |
| 名称 | 内容 | |
| 编号 | | Test05 |
| 名称 | | 时间表管理 |
| 前提 | | P1、Tomcat服务器已启动  P2、管理员账号已登录 |
| 操作步骤 | | S1、查看所有时间表  S2、查看时间表详细信息  S3、修改时间表  S4、添加时间表  S5、删除时间表 |
| 期望结果 | | 1、各操作均能显示操作成功  2、数据库能被成功修改 |
| 实际结果 | | 符合期望 |

## 测试结论

根据人工进行系统功能测试，医院网上预约就诊系统各个功能模块下的基本功能进行测试，并修改相关bug，最终系统所有功能可以正常使用，因此预约就诊系统可以正常部署并上线使用。

## 本章小结

系统测试是系统上线前必备的流程，通过该流程我们可以对系统中可能出现的问题进行预测和测试。对于出现的问题可以及时调整，有利于系统的稳定性和可维护性。对预约就诊系统的各个功能进行了详细的测试。首先通过设计测试用例和科学的测试方法。之后根据设计的测试用例严谨的验证了各个功能的可用性和稳定性。通过本章的各个流程测试，理论和实际上都保证了系统可以正常使用和部署。

# 总结与展望

本篇比较系统的阐述了网上预约就诊服务体系的基本设计思想和具体实现流程。。首先通过分析如今网上预约的现状和国内外预约形式的异同来初步拟定要设计一个什么样的系统。再对系统各个功能模块进行设计和各模块功能需求的分析，让我们对系统架构有一个初步的认识。后面分别对各个模块的功能进行设计，每个功能之间的调用关系进行梳理。然后就是对设计出的功能进行代码上的实现、进行服务的搭建和数据库表的实现。最后对预约就诊系统的各个功能进行了科学的测试。使用测试用例检测系统运行过程中可能出现的问题并进行解决。可以得出设计出的系统是可以正常运行部署的结论。

本网上预约就诊系统的设计可以有效解决中小型医院和诊所的网上预约就诊系统的部署难题，极大程度上降低了部署和管理的成本。本系统的上线可以有效分担线下挂号的负担。非常简单的预约流程也可以降低系统的使用门槛，即使是刚接触智能手机的老年人也可以轻松使用本系统。

随着研究的深入和分析，本系统需要改善的地方：

1. 验证码的发送没有时间的限制，应该和当前的主流验证码设计保持一致，可以设计成系统判断一分钟内只能点击发送一次验证码，如果不加限制会有人恶意利用来消耗系统资源。
2. 数据库的密码没有加密在传输的过程中有可能信息被劫持导致密码泄露。并且如果数据库内容被窃取大量明文密码将会泄露，对系统和用户的个人信息造成影响。
3. 科室信息这种不会经常改变的信息可以不是每次都从数据库中查询。固定不败你或者不经常改变并且经常查询的数据可以放到Redis中，这样查询效率会高很多，并且可以节省很多频繁调用数据库产生的资源浪费。

参考文献

[1]张丹峰. 开放统一预约系统设计[D].西安电子科技大学,2018.

[2]刘洋,赵博.浅析医院自助服务信息化管理[J].经营管理者,2018(12):84.

[3]赵欣蝶,茅梦颖,毛雨静,曾驿涵.以网上挂号为例探究老年人群新媒介接触现状[J].传媒论坛,2021,4(01):22-23+26.

[4] 黄金链,郭丹媚.医院门诊预约挂号服务分析[J].实用临床护理学电子杂志,2017,2(41):150-151.

[5]张明,杨晗.网上预约系统在医院的应用[J].现代经济信息,2018(17):103.

[6]邓倩. 基于spring框架的网上预约挂号平台的设计与实现[D].电子科技大学,2016.

[7]童良. 基于B/S的医院信息管理系统的设计与实现[D].电子科技大学,2020.DOI:10.27005/d.cnki.gdzku.2020.001182.

[8]夏小涵. 私立儿童医院管理系统的设计与实现[D].吉林大学,2018.

[9]邢娜,郑蕾,王莉.医院网站预约挂号系统设计[J].解放军医院管理杂志,2021,28(10):929-930+969.DOI:10.16770/J.cnki.1008-9985.2021.10.012.

[10]魏星. 基于B/S架构的预约挂号系统的设计与实现[D].天津工业大学,2018.

[11]杨茂琼,杨成.多元化预约挂号在三级甲等综合医院门诊管理中的应用研究[J].人人健康,2021(22):84-85.

[12]张建.预约挂号要等等老年人[J].晚晴,2021(03):5.

[13]陈喜芳,刘绮才,周风齐.门诊预约挂号对患者服务质量及满意度的调查研究[J].实用临床护理学电子杂志,2020,5(20):171+173.

查重报告



格式检测报告

