

电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE



论文题目 某医院挂号预约管理系统设计与实现

专业学位类别 工 程 硕 士

学 号 201651080118

作 者 姓 名 叶基林

指 导 教 师 王伟 教授

分类号_____密级_____

UDC 注 1 _____

学 位 论 文

某医院挂号预约管理系统设计与实现

(题名和副题名)

叶基林

(作者姓名)

指导教师

王 伟

教 授

电子科技大学

成 都

霍军亚

高级工程师

美的集团家用空调事业部

广 东

(姓名、职称、单位名称)

申请学位级别

硕士

学科专业

工程硕士

工程领域名称

软件工程

提交论文日期

2021.3

论文答辩日期

2021.5

学位授予单位和日期

电子科技大学

2021 年 6 月

答辩委员会主席

评阅人

Design and Realization of The Hospital Registration and Appointment Management System

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Discipline: **Master of Engineering**

Author: **Jilin Ye**

Supervisor: **Prof Wei Wang**

School: **School of Mechanical Electrical Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：叶曼林

日期：2021年6月19日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：叶曼林

导师签名：王

日期：2021年6月20日

摘要

医院的挂号预约服务可以基于移动互联技术的基础上,提升准确性和及时性,让患者的等候医生时间减短,精准性提高,提高患者的满意度。本论文的医院位于广东省佛山市某镇区,属于丙级医院,目前的挂号预约的模式是:电话预约或者现场预约,但是随着镇区的发展,人口增加,基于电话预约或者现场预约的模式,已经无法满足日益增长的需求,患者抱怨多,意见大。

本论文研究是基于移动互联网下,选取医院挂号预约系统作为研究对象。结合某医院挂号的现状分析,主要研究内容如下:

1) 指出了医院挂号面临的困难,指出构建本系统重要价值,对国内外系统的应用情况、文献研究情况进行了分析,明确了研究的主要目的与重要意义。然后介绍了 APP 端、WEB 端两个子系统使用的开发技术,主要包括 MVC 框架技术、数据库 MySQL、UML 建模以及 Android 等技术的工作原理、应用价值以及主要特点。并对系统建设进行经济、技术与操作等方面的可行性分析,描述了用户角色构成与可使用的功能。

2) 介绍了系统总体功能,通过功能用例图描述了功能的层次功能构成,描述了系统核心功能的进行流程与数据流程,确定了业务处理过程以及涉及到的数据实体,并描述了系统在性能、安全等领域的非功能性需求。

3) 对系统进行详细设计,搭建了系统应用、技术与网络等方面的框架,梳理了功能构成、开发技术与硬件部署等内容,设计了系统功能结构图,对系统用户管理、科室/医生管理、排班管理、预约管理、支付处理、消息通知等核心功能进行程序流程图、接口等方面设计,重点设计了系统数据库概念、物理模型。

4) 对系统进行 APP 与 web 端的开发,提供核心代码与实现界面,描述了实现场景。搭建了测试环境,描述了各项测试参数配置情况,进行了数据、功能、性能、非功能性等方面测试,验证通过,确认了系统设计与实现效果达到了预期目标。

通过挂号预约管理系统的构建,实现了用户快速预约,快速就诊,构建了公正透明的挂号秩序,显著减少了现场挂号带来的时间浪费,提高了医疗服务质量。降低了人工挂号带来医疗管理成本,为患者提供更加精准的服务。

关键词: 医院挂号, 预约管理, APP 预约

ABSTRACT

The hospital's registration appointment service can be changed based on the existing network technology and mobile internet technology to improve accuracy and timeliness, so that patients' waiting time for doctors is shortened, accuracy is improved, and patient satisfaction is improved. The hospital in this thesis is located in a township in Foshan City, Guangdong Province, which is a Class C hospital. The current mode of registered appointment is: telephone appointment or on-site appointment. However, with the development of the town, the population increases, based on telephone appointment or on-site appointment. The model has been unable to meet the increasing demand, and patients have complained a lot and opinions are big.

The research in this thesis is based on the mobile Internet, and the hospital registration appointment system is selected as the research object. Combining the analysis of the status quo of a hospital registration and the problems existing in the hospital registration method, the main research content is as follows:

1) The research in this thesis points out the difficulties faced by hospital registration, points out the important value of building this system, analyzes the application of the system at home and abroad and the literature research, and makes clear the main purpose and significance of the research. Then it introduces the development technologies used in the APP and Web subsystems of the system, mainly including the working principle, application value and main characteristics of SSH framework technology, MySQL, UML and Android technology. The feasibility analysis of economy, technology and operation of the system construction is carried out, and the composition of user roles and the functions available are described.

2) The research in this thesis introduces the overall function of the system, describes the hierarchical function structure of the function through the function use case diagram, describes the process and data flow of the core function of the system, determines the business processing process and the data entities involved, and describes the non-functional requirements of the system in the performance, security and other fields.

3) The research in this thesis is to the detailed design of system, set up the system application, technology and the network framework, combed the function structure,

development technology and hardware deploy content, design the system function structure, the system user management, department/doctor management, scheduling management, booking management, payment processing, alerts, and other core functions of program flow chart and interface design, The concept and physical model of system database are designed emphatically.

4) The APP and Web end of the system are developed, the core code and implementation interface are provided, and the implementation scenario is described. The test environment was built, the configuration of various test parameters was described, and the data, function, performance, non-functional and other aspects of the test were tested. The verification was passed, confirming that the system design and implementation effect reached the expected goal.

Through the establishment of the registration appointment management system, users can make a quick appointment, see a doctor quickly, build a fair and transparent registration order, significantly reduce the waste of time caused by on-site registration, and improve the quality of medical service. It reduces the medical management cost brought by manual registration and provides more accurate services for patients.

Keywords: Hospital Registration, Appointment Management, APP Appointment

目 录

第一章 绪 论	1
1.1 课题的背景与意义	1
1.2 国内外现状	2
1.3 系统的主要贡献	4
1.4 论文的组织结构	4
第二章 系统开发技术概述	6
2.1 J2EE 技术	6
2.2 MVC 框架技术	7
2.2.1 Spring 框架	7
2.2.2 Struts2 框架	7
2.2.3 Hibernate 框架	8
2.3 MySQL 技术	9
2.4 UML 建模技术	10
2.5 WEB 技术	10
2.6 Android 技术	10
2.7 本章小结	11
第三章 挂号预约管理系统需求分析	12
3.1 系统总体需求	12
3.2 系统可行性分析	13
3.3 系统角色分析	13
3.4 系统功能需求分析	14
3.4.1 APP 挂号预约	14
3.4.2 WEB 后台管理	17
3.5 系统流程分析	21
3.6 非功能性需求分析	24
3.7 本章小结	25
第四章 挂号预约管理系统设计	26
4.1 系统应用架构设计	26
4.2 系统技术架构设计	27

4.3 系统网络拓扑设计	29
4.4 系统功能架构设计	30
4.4.1 APP 挂号预约	31
4.4.2 WEB 后台管理	33
4.5 系统接口设计	36
4.6 系统数据库设计	37
4.6.1 数据库表结构设计	37
4.6.2 数据并发处理	42
4.7 本章小结	42
第五章 挂号预约管理系统实现	44
5.1 APP 挂号预约系统实现	44
5.1.1 登录注册	44
5.1.2 在线预约	46
5.1.3 支付处理	48
5.2 WEB 后台管理实现	50
5.2.1 后台用户管理	50
5.2.2 预约管理	51
5.2.3 科室/医生管理	52
5.2.3 排班管理	55
5.2.4 消息通知	55
5.3 本章小结	56
第六章 系统测试	57
6.1 测试总体设计	57
6.2 测试用例	58
6.2.1 用户管理	58
6.2.2 科室/医生管理	58
6.2.3 排班管理	59
6.2.4 预约管理	59
6.2.5 支付处理	62
6.3 本章小结	62
第七章 总结和展望	63
7.1 全文总结	63

7.2 展望	63
致 谢	65
参考文献	66

第一章 绪 论

1.1 课题的背景与意义

随着我国进入老年化社会,老年人数量剧增,糖尿病、高血压等慢性疾病是影响老年人长寿的重要因素,就医人数呈现出显著上涨趋势,2019 年全国医疗卫生机构总共接待了不同类型的患者进行治疗总诊疗,人数增长了 4.1 亿人次,增长率达到了 4.9%^[1]。城市生活节奏的不断加快,让许多年轻人也产生了各种压力,容易在身体、心理上产生相关的疾病。国内一二三线城市三甲医院具有优质的医疗资源,导致许多患者都往这些医院来排队就诊,患者人数多,需要花费大量的时间来排队,而且还不一定能够取得想要时段的号源。即使有号,也可能形成上午排队取号、下午才能就诊,大部分时间花在了排队等候上^[2]。实际看病可能只花了三到五分钟,时效比太低,患者体验感太差^[3];有些热门门诊与名医还被黄牛垄断号源,然后超高价格出售给患者,提高了患者就医成本。如何提供快速便捷的就诊方式,节省患者的候诊时间,创造一个公平透明、快速高效的挂号流程和就诊环境,成为医院体制深化改革获得患者满意度亟需解决的首要问题。许多医院已就该问题,根据实际采取了不同的尝试和手段来提高效率,节约资源。

当前社会计算机信息化建设水平高,创造了良好的数字化应用环境,医疗卫生领域也在积极引进这些技术融入到对医疗机构治疗管理工作中,有效提升了相关单位的工作效率。就医人数的日益上升,迫切需要医疗机构加强自身数字化建设力度,更加高效地利用各种设备资源,以优质的管理与服务,解决医疗机构接诊压力问题。随着国家卫生部对患者就诊时间问题的重视及指导性文件的发放,全国各地医院想方设法、依托移动互联技术提供线上预约服务,患者可以通过手机进行预约挂号,为患者提供了优质医疗服务,缩短患者挂号等待时间,尽可能减少排队轮候时间,大大缩短了就医实际成本^[4]。

某医院位于广东省佛山市的镇区,属于丙级医院,存在看病人员多,许多用户(患者)到该医院看病,都需要花费很长的时间,轮候排队,而且即使排上号,也不能看到自己心仪的医生,并且在患者中,有大部分属于工薪阶层,漫长的等待往往耗费其大部分的时间,甚至出现了为了免除等待时间,雇人排队,为了看上心仪的医生,不惜重金从黄牛哪里买号。该医院目前的预约挂号方式为:用户(患者)到医院现场挂号、人工电话挂号、网络挂号等方式。但这几种方式存在自身的不足,首先是不能非常及时地响应用户请求,当存在突发事件导致患者所挂号的医生不能坐诊,医院难以快速反馈给患者用户,会带给患者很大的不便与

反感^[5]。

该医院就如何提供快速便捷的就诊方式，节省用户（患者）的候诊时间，创造一个公平透明、快速高效的挂号流程和就诊环境，成为医院获得用户（患者）满意度亟需解决的首要问题。该医院内部也就此问题，根据实际情况采取了不同的尝试和手段来提高效率，节约资源，依托移动互联技术同步开展了形式多样的门诊预约服务，患者可以通过移动网络进行挂号，减少用户（患者）挂号排队等待时间，实现用户（患者）到医院后就能在 30 分钟内完成诊断。

该医院基于移动互联网技术，研究一个用户可不分时间段全天候的预约系统，让用户可以按自身需求通过各种途径快速预约到医生资源，避免不必要的排队和轮候的时间，提升用户（患者）的满意度，是该医院目前的迫切需要。医院预约挂号系统是目前该医院信息化、数据化、简易流程化建设的重要组成因素，给患者提供了就医方便的服务，确保患者就医更加高效。本文提出的医院预约挂号系统能够有效的提升挂号效率。

1.2 国内外现状

预约挂号是当前现代化医疗体系不可缺少的内容，患者就医可以提前挂号，能够有效提高看病治疗的效率^[6]。许多市民为了抢夺优质医疗资源，往往集中到知名三甲医院排队就诊，但是因为排队人数多，时间成本很高，让黄牛找到了巨大市场，导致就医秩序的混乱，破坏了公平就医的环境。随着互联网特别是移动互联网的快速发展，在线预约挂号发展成为了医疗就诊的重要环节，受到了市场的高度认可。

在国外欧美地区，几乎以实现完全预约诊疗服务的方式来达到效率提升、患者满意的目的。在欧美等发达国家，其对患者就诊时间的安排制度与国内不同，其国家的医生大部分是私营，家庭医生与患者建立了稳定的合作关系，并根据患者档案能够掌握对方的身体状况^[7]。患者有病的就诊前，首先想到的是家庭对口的医生，通过与家庭医生交流确定是否需要就诊。如美国麻省理工医院医疗技术非常先进，每天门诊平均不到 5000 人次，但是基本都是提前预约就诊^[8]。通常患者会按照常规流程进行就医^[9]：第一步，预约家庭医生。预约形式是电话、电子邮件通知家庭医生居多，当患者需要就诊时，医生会安排就诊医院，用户通过医院平台进行预约并确定就诊时间。病人根据家庭医生的建议，携带转诊单，直接通过电话告知接诊的医疗中心或专科医院来预约就诊时间。可见，国外如欧美地区国家，基本建立了以家庭或社区为主线的分诊的家庭预约体系，医疗机构能够为患者提供各个专科提供位置最近的预约服务与医疗资源，并满足他们高效就诊需求，

同时也能够快速实现医疗机构盈利。

西方国家在医院预约挂号领域的技术研发方面,通常使用 Parnas 方法,可以实现软件重用和组件连接^[10]。在软件工程领域引入了 Jira 敏捷软件开发、极限编程等方法,采用模型驱动体系结构,提出了测试驱动开发与面向 Agent 软件开发^[11]。客户端 APP 方面有使用基于 PhoneGap 的开源版本 Cordova 构建 ionic 技术。前端会采用 Node.js 来进行界面开发,在开发过程使用 Docker 软件容器化平台来管理程序,另外开发平台包括 IntelliJ 集成开发平台,采用 Git 作为开源版本控制系统^[12]。A. G. Philippa 提出了基于社交软件优势,开发了医院挂号系统设计与实现,方便患者挂号就医^[13]。M. Sergio 开发了智能化的移动预约挂号平台,为医患双方提供便利^[14]。G. D. Smith 为了节约患者排队挂号时间,设计了预约挂号系统,提高医院工作效率^[15]。J. Badreddine 设计了医院叫号系统,通过医生信息发布与展示,有序引导患者就医^[16]。K. Hichem 设计了预约挂号系统,改变传统就诊模式^[17]。S. Williams 设计的挂号平台通过数据分析,科学安排患者就诊和医生排班^[18]。

我国目前的一二线城市三甲医院普遍建设独立预约诊疗系统,通过微信公众号或电话号码进行预约;但医生专家资源稀缺,预约少,医院之间缺乏互联互通。三四线及以下城市,普遍缺少独立预约诊疗系统^[19]。另外,预约挂号系统还没有与医院的 HIS 系统实现数据的完全对接,共享程度不高。研究一个可以用户可不分时间段全天候的预约系统,让用户可以按自身需求通过各种途径快速预约到医技资源,避免不必要的排队和轮候的时间,提升患者的满意度^[20]。

目前针对医院预约挂号的应用与数字化建设方面的研究较多。其中,马静璇探讨了现有预约挂号方式存在的不足,对基于互联网+平台的远程挂号展开了深入研究^[21]。李雅琼等在医院预约系统开发中提出对 IP 地址限制、身份证实名制挂号、随机放号的方式,来尽量满足公平性原则,降低黄牛囤票的风险,支持 Webservice 方式访问号类资源^[22]。梁景云提出了基于门诊全预约挂号流程的优化,通过对比分析,指出利用预约挂号能够提高挂号成功率、预约满意度,就诊人员更有时间掌控能力^[23]。杨靖祎提出开发医院微信公众号,提供预约挂号服务^[24]。巩蕾提出了开发智能手机端的预约 APP,支持短信验证码预约以及到医院扫码取纸质号,从而提高实名认证约束水平^[25]。冯清分析开发口腔门诊挂号系统 APP 能够带来的重要应用价值,对其业务流程进行了分析,然后设计了基于患者版 APP 和医务人员版 APP,并总结其应用效果^[26]。张鹏利用移动设备操作系统 Android 的优点,应用在医院的预约挂号系统中,提高了信息资源利用率,方便广大患者更好就医^[27]。孙亚琦设计了医院挂号 APP,构建了 UML 动静态模型,设计了功能模块,保障了系统性能,使用 Objective-C 语言实现交互功能^[28]。姜慢开发医院预约挂号系

统，方便患者进行线上挂号^[29]。王珂设计开发了基于 Android 系统的医院移动预约挂号系统，通过研究发现系统 CPU、内存使用率都不高，系统运行稳定^[30]。张磊提出构建医院预约挂号系统，提供患者与医疗机构的交互性建设，实现在线答疑解惑^[31]。这些研究从开发技术、应用程度、建设价值等多方面进行了在线预约挂号方面信息化探讨。

1.3 系统的主要贡献

本系统选取医院挂号预约系统作为研究对象。结合某医院挂号的现状与方式进行问题分析，挂号系统必须根据用户（患者）的需求，实现预约挂号的功能，其途径是通过完善要素和功能来实现，以减少用户（患者）对于挂号时间长，就诊时间长等问题的抱怨。

硬件上，基于医院现有的内网系统的硬件为基础，在系统上开发外置的预约挂号的软件接口，增加必要的硬件终端，保证用户（患者）可以通过手机、网络、电话、现场终端等方式快速与挂号系统相连，实现快速就诊。

使用 Android 和 Java，开发手机终端的开发 APP 预约挂号系统，通过 UML 和 WEB 技术，使用 MySQL 数据库开发后台管理系统，后台管理包括了用户管理、预约管理、消息通知、排班管理、科室/医生管理等子系统，建立用户信息数据库，以便实现用户（患者）可快速预约需求的功能。

为了防止医院的预约号被恶意占用，首先，系统采取身份证实名制预约，在进行用户注册操作时，需要验证对身份证和手机随机码是否正确。在现场就医取号时需要提供手机短信和用户身份证进行核验；系统设置用户在同时段范围内只能进行一次预约操作，当天不能超过 2 次预约。系统能对根据用户操作异常，对用户 IP 地址进行监测，当确定用户进行刷号操作，系统能够自动屏蔽该 IP 地址访问请求。其次，增加动态识别码技术，来降低黄牛抢票速度，为其他患者赢得抢票时间。这些技术也能避免恶意“刷号软件”的恶意攻击。

1.4 论文的组织结构

本系统研究医院预约管理系统，主要研究内容包括预约挂号功能的 APP 应用设计、后台管理系统设计。

本系统的章节结构安排如下：

第一章 绪论：指出广东省佛山市镇区的某丙级医院的医疗资源紧张、挂号方式落后的现实问题，提出构建在线预约系统来解决问题。然后分析了国内外在医院挂号方面、信息化建设方面以及文献研究方面的情况。指出系统的主要贡献。

第二章 系统开发技术概述，介绍了系统开发用到的技术，如：J2EE、MVC 框架技术、MySQL、UML、WEB、Android 等技术。

第三章 需求分析：基于医院的现状，结合现有的挂号模式和硬件能力，确定系统总体需求，从 APP 挂号预约、后台管理等两个方面介绍系统功能，使用用例图描述用户与功能之间的关系，对系统业务流程进行描述，介绍系统技术可行性和安全设计原则与措施等。

第四章 系统设计：本章阐述了系统的应用架构设计在数据层、业务逻辑层、数据层的采取的设计方法，阐述了 WEB 端和 APP 端的技术架构设计，同时也阐述了医院预约挂号管理系统网络拓扑设计，并且说明了系统功能设计、接口设计、数据库设计的方式。

第五章 系统实现：描述系统界面与类图的功能实现，每个功能的功用。

第六章 部署和测试 本章主要包括系统的发布，部署和测试，介绍了用户管理、科室/医生管理、排班管理、预约管理、支付处理的测试用例，分析了测试结果。

第七章 总结和展望：本章总结了在系统开发过程中存在不足的地方，以及后期的扩展和维护进行总结。

第二章 系统开发技术概述

本文研究的医院预约挂号系统主要是基于 Java 语言进行开发的,采用 SSH 框架进行集成开发,能够满足企业级应用开发,可以为用户提供性能稳定、功能完善、数据正常的系统。

2.1 J2EE 技术

J2EE 支持开发企业级应用,可以为软件开发提供技术指南,科学规范了程序开发过程,有效解决程序部署问题,是层次结构清晰的优秀框架,如图 2-1。

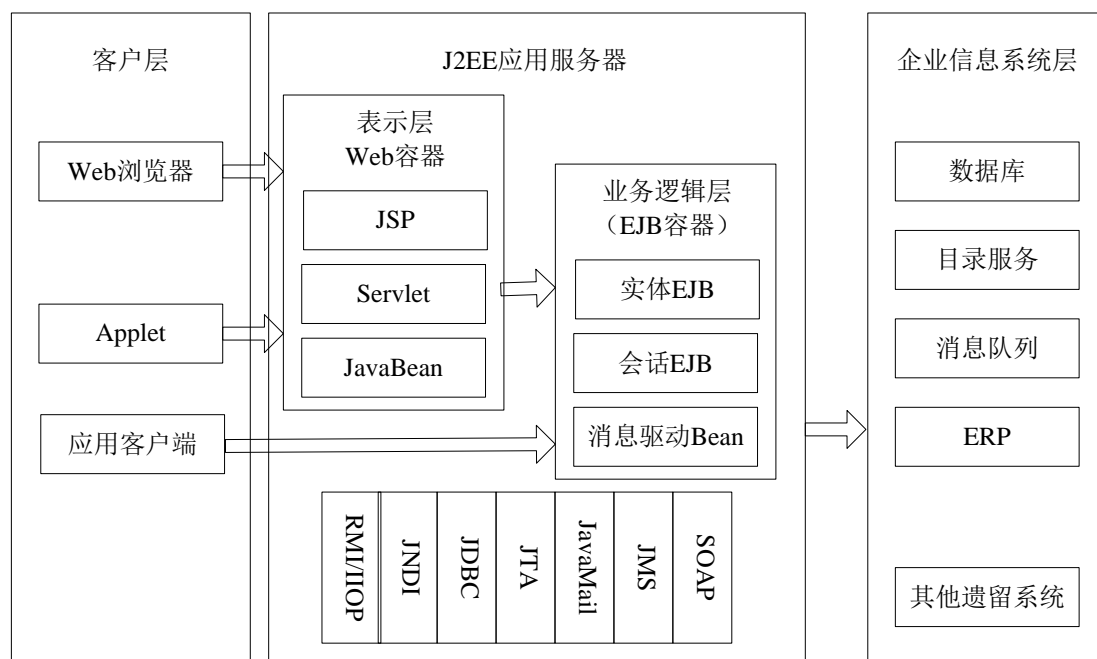


图 2-1 J2EE 层次结构

从图示结构可知,层次结构简洁,各级相互之间紧密联系,其构成包括由 J2EE 应用服务层、企业信息系统层、客户层等,相互间发挥着十分重要的作用。客户层是建立与客户对接关系,客户能够直观看到并与之交互的层面,同时也支持用户可操作的组件,使得用户可利用 http 协议实现直接访问,支持服务器将 XML、HTML 格式的 web 界面输出;基于 java 虚拟机的 Applet 在浏览器中嵌入运行。J2EE 应用服务层由 web 容器和 EJB 容器组成,解析并响应用户发出的请求程序,处理用户需求内容与业务逻辑,对全生命周期进行管理。服务层可提供各类组件和插件的通信接口。企业信息系统层提供基础数据管理、检验检索、目录匹配服务。

2.2 MVC 框架技术

2.2.1 Spring 框架

Spring 作为开源性特点的应用程序开发框架，可以提供友好的框架环境，兼容和整合其他一些框架和插件，同步提供模块化开发，使得程序员的开发效率得到有效地提高。该框架核心作用是控制页面跳转和数据输出，支持用户通过服务接口进行远程调用的业务逻辑处理。

Spring 框架特性构成如图 2-2 所示，具有高内聚、低耦合特征，有效减少对 APIs 的依赖^[25]。Spring 框架内嵌封装的 jar 包，程序员通过调用已完成封装的服务和方法来提高程序开发效率，降低反复开发和测试同类代码的工作量。同时，通过对响应比来提升框的对数据访问效率。Spring 框架的核心特点是控制反转 (IOC) 和 DI 依赖注入，是框架被广泛应用的基础，在对象创建和组件方面发挥作用。面向切面编程 AOP 也是该框架其中的一个功能，能够很好控制系统的运行状态。

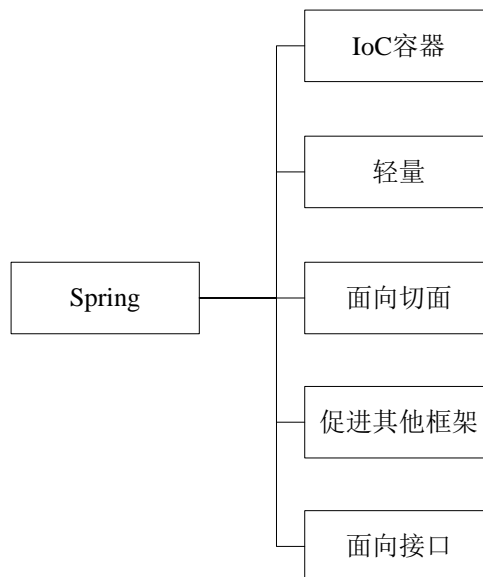


图 2-2 Spring 特性构成

2.2.2 Struts2 框架

Struts2 框架是 MVC 开发模式中的 View。它在页面输出，通过内嵌页面样式控制标签库来输出美观的界面发挥着重要的作用，迅速提升了程序员在前端 JSP 编写效率，其控制页面间跳转是通过 xml 配置文件来实现的。该轻量级框架能与 Spring 框进行集成运用，在许多创新性元素基础上升级了原有框架。

Struts2 体系结构如图 2-3 所示，客户浏览器发送访问请求给 jsp，Servlet 容器接收到请求后采用 filters() 方法识别并解析请求得内容，同步提供页面导航功能。

接着 `FilterDispatcher` 控制器合理分配相关操作权限后跳转，通过配置管理器 `ConfigurationManager` 来对各项基本参数进行配置。在 `web.xml` 文档里进行统一配置，是为了加强业务功能处理过程的管控的目的。提供丰富的视图组件来满足动态页面输出需要是 `Struts2` 的扩展功能，能够提供各种数据与业务接口来完成系统的构建。

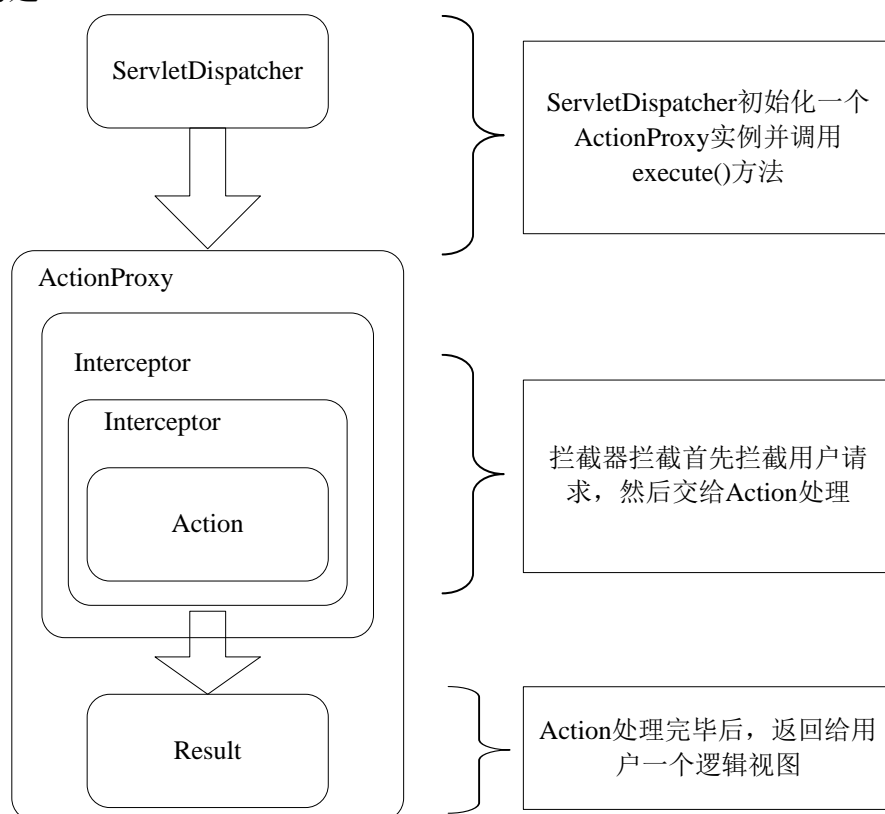


图 2-3 Struts2 体系结构

2.2.3 Hibernate 框架

`Hibernate` 的实现是建立于对数据库访问程序常用 `JDBC` 语句的进行封装的基础上，其特点是建立应用程序与数据库连接的开放源代码的对象关系映射框架，简化数据访问层繁琐的重复性代码和减少后期编程数量。`Hibernate` 为了实体属性、类型与数据库表字段、类型四者的一致性，建立实体对象与数据库表的映射关系的功能。同时 `Hibernate` 提供了数据快速查询的接口，实现了对各类型事务与任务并发工鞣呢进行高效管理。`Hibernate` 的核心接口图如图 2-4 所示，分别负责建立会话关系、配置、事务处理、查询、标准化等五个类型的接口。`Hibernate` 能够确保一定时间范围内不需反复访问数据库的稳定连接以实现 `jsp` 的数据来源。`Hibernate` 功能是十分强大的，其实体数据映射关系可由 `.hbm.xml` 文件来配置，实验与与 `DB2`、`mySQL`、`Oracle` 等主流数据库的访问连接，并且兼容不同的应用服

务器，比如 Tomcat 等。

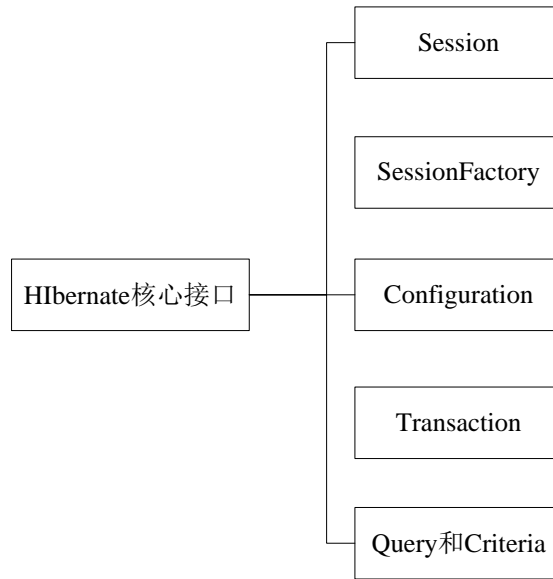


图 2-4 hibernate 核心接口

2.3 MySQL 技术

MySQL 的数据库系统，其特点是免费、开源给程序员更改，并且占用空间小、运行速度快。MySQL 数据库管理系统由于其程序代码公开透明，能够兼容不同开发语言，它的运行原理非常简单和明确。

MySQL 数据库基础控件非常丰富，其数据库表结构的存储关系型数据能够将字符、数字、日期等类型数据校验合格后进行保存。同时也通过提供索引来提高数据检索效率，定期执行相关事务是利用存储过程来实现的，并生成指定格式的报表、指标数据。由于 MySQL 兼容不同编程语言，其也能灵活配置各种参数来提供不同的数据库访问持久化中间件提供服务接口。

MySQL 数据库特点如下：

- 1) 由于其跨平台、跨语言运行和服务能力，同时能够支持 java、C++等各种类型编程语言访问，所以在 Linux、Unix、Windows 等类型操作系统上稳定运行。
- 2) 用户可以从客户端可视化管理界面访问到数据库，在数据编辑与维护，交互方面可通过 pl/sql 工具直接查看数据库记录。
- 3) 扩展性非常好，同一时间可接受不同客户端访问，并保持稳定。
- 4) 存储空间大。

2.4 UML 建模技术

UML 是功能构成与数据处理过程的模型语言。UML 通过提供可视化图形信息，来帮助程序员更好的处理需求分析、程序设计。UML 是程序员在进行程序编写之前不可或缺的工具。其在绘图功能、提供模型库、导航功能、模型互换功能上，通过简单易懂、可视化的方式展示了软件组件构成，展示了不同关联对象的内在关系与外部交互作用过程。描述事物处理机制和数据对象可以通过 UML 对静态与动态结构进行建模完成。UML 的用例图是以多层次关系描述功能与子功能、功能之间的关系的静态类型结构图；描述事物动态发展过程与处理机制则是时序图和协作图。

2.5 WEB 技术

Web Service 是借助标准通信协议实现程序模块的发布，实现分布式互联网系统构建的基本部件。SOAP 是 Web Service 使用的通信协议。标准的互联网协议-XML、HTTP 和 TCP/IP，是 Web Service 主要的优点，可实现在不同的平台和不同语言上开发出来的程序在 Web Service 使用。Web Service 通过常用的技术标志语言中的 HTTP 以及 XML 为多个组织之间的业务流程的集成提供一个通用机制，同时借助 XML 语言对其他各种语言编写的接口进行封装，利用 Web service 实现数据的访问以及查询功能。

2.6 Android 技术

Android 是 Google 公司研发的开源的操作系统，该系统是基于 Linux 基础之上，能够为手机 APP 运行提供基础环境。该操作系统是开放源代码的，程序员能够在此基础上开发组件，并且可以兼容不同的应用平台^[28]。Android 平台的开发目的是资源整合，包括了底层 Linux 操作系统、中间层的插件以及上层的 Java 程序。它提供了 KeyApps、SDK、LinuxKernel、Middleware 等四个部分在内的手机全套软件方案^[30]。程序员可按实际需求来更改系统功能，定制化需要的插件，使用软件美化 UI 界面美，整合封装成 jar 包进行调用。Android 也支持提供国际化插件的功能，在数据安全存储和全面管理时用 SQLite。Android 系统上同时支持多线程可运行多个应用程序。Android 系统也可进行辅助进程的管理，最大程度优化内存，快速响应程序组件，同时对数据共享、视频播放提供基于框架下安全保障。

2.7 本章小结

本章简要介绍了医院挂号预约管理系统开发过程用到的关键技术，对源代码编写的语言、MVC 开发框架、数据存储 MySQL、UML 建模、WEB 技术以及 Android 等技术的特点、工作原理等进行了描述，为系统开发创造条件。

第三章 挂号预约管理系统需求分析

3.1 系统总体需求

目前，某医院的现在整个患者看病流程，如图 3-1 所示。本次开发是结合某医院的实际情况和需求，尽可能基于现有医院的医院现场挂号、电话人工或自动挂号系统的基础上，增加和设计一个基于移动互联技术上开发的挂号预约管理，让用户（患者）根据自身的需求，可不分时间段全天候的通过各种途径和方式（如医院网站、电话、手机、医院终端等）快速完成挂号，预约到医生资源，按预约时间找到对应的医生及医生所在的业务科室，实现快速就诊。

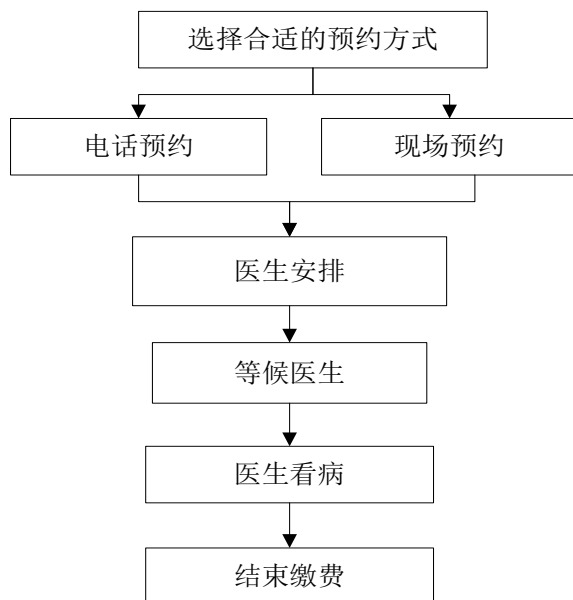


图 3-1 某医院患者看病流程

预计某医院的预约挂号系统由 APP 预约挂号和后台管理组成。

预约挂号 APP 是本文的主要研究的内容，用户（患者）和医生通过移动通讯工具（手机、IPD）的界面，如图 3-2。以此实现即时及迅速就医的目的，减少用户（患者）排队等候医生的时间。

本次是开发的明确的目标是结合医院目前的实际业务流程，实现一个基于某医院的预约挂号 APP 子功能，丰富医院的挂号模式的多样化。根据上述开发思维，我们对现实生活中在医院挂号的业务流程进行抽象、提炼，形成实验挂号预约系统的功能子域，并区分这些子域为不同的类别，以便后续实施合适策略。

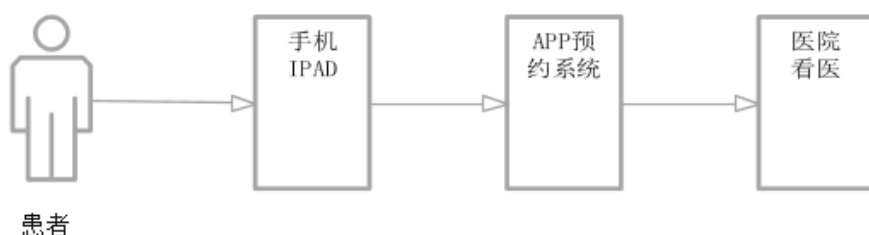


图 3-2 看病预约

3.2 系统可行性分析

1) 技术可行性分析

本系统手机端采用 Andorid 系统平台架构，后台 WEB 服务器端使用 B/S 的模式来实现用户与系统之间的互动。软件后台采用 J2EE、Spring MVC 等技术来实现系统的开发。系统经由 MyEclipse 软件进行相应的研发，运用 J2EE 模式完成 Java 编码工作，之后应用 Webservice 完成信息调用过程，通过 Webservice 的技术达到远程调用的目标，最终经由 SQL 语言完成对信息的高速检索。

2) 经济上的可行性

某医院在医院系统的数字化、智能化建设上非常重视，为满足患者需求和医院现有系统的管理需求，将在医院信息化建设上投入大量资金，所以医院挂号预约管理系统的建立是有充足的资金，为专业开发、硬件设备采购等上提供保障。同时挂号预约管理系统正式投入使用，也会节省电话挂号和现场挂号的管理和服务成本，将会医院带来显著的经济效益，并提升了医院形象。

3) 操作上的简单性

系统开发会结合医院行业特色来设计人性化、个性化的 UI 界面和导航功能，并且基于用户使用习惯来设计系统功能样式、布局，让用户快速找到所需的功能，提高人机交互水平与用户体验感。

3.3 系统角色分析

医院挂号预约管理系统包括 WEB 端和 APP 端，每个子系统中包含系统管理员、患者等不同的角色，如表 3-1 所示。

表 3-1 系统角色功能表

序号	角色	系统	功能简述
1	系统管理员	WEB 端	维护科室、医生、排班等基础信息，对预约信息、账号等进行管理。
2	患者	APP 端	设置和收到挂号预约消息，进行在线挂号预约，可以查阅历史预约记录，取消预约，在线支付。

3.4 系统功能需求分析

本预约挂号管理系统重要的功能需求包括 App 挂号预约：用户（患者）注册、在线预约、支付处理和后台管理：预约管理、科室/医生管理、排班管理、用户管理、消息通知。

3.4.1 APP 挂号预约

APP 挂号预约的功能是能够为用户（患者）提供通过基于移动互联技术，通过手机、IPAD 等移动工具来进行远程在家在公司在路上的无限互联的预约服务，用户（患者）如果要使用该医院的在线预约服务，可通过微信小程序，或 APP 来实现，其界面管理的挂号预约管理如图 3-3 所示，包括登录注册、个人中心、在线预约、在线支付等核心功能。

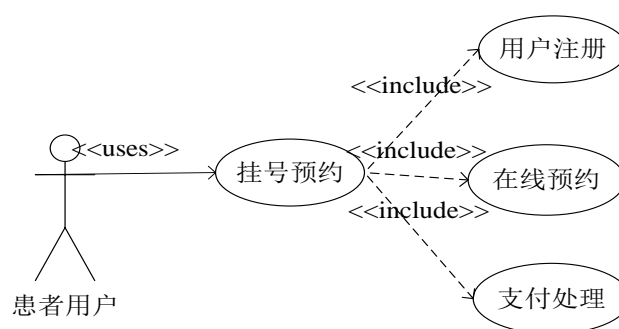


图 3-3 APP 挂号预约

1) 用户（患者）登录注册

用户登录注册，医院一般称为用户（患者）建档。用户登录 APP 后，为了在该医院预约到心仪的医生，需要注册账户，通过身份验证才可以在系统内进行预

约，采取实名制预约。用户账号登录过程需要对输入的密码进行加密保存。注册过程可以用手机号码注册和随机码来进行验证，消息推送到手机号码。如图 3-4 所示系统登录注册功能逻辑图。如果忘记密码，可以用手机验证码进行重置。

同时提供个人中心给患者维护基础资料，查看个人信箱是否收到系统推送的消息，重新设置密码。支持单个账号自己家庭内的用户（患者）进行预约，对于不存在的用户（患者）可以删掉。个人中心模块逻辑图如图 3-5 所示。

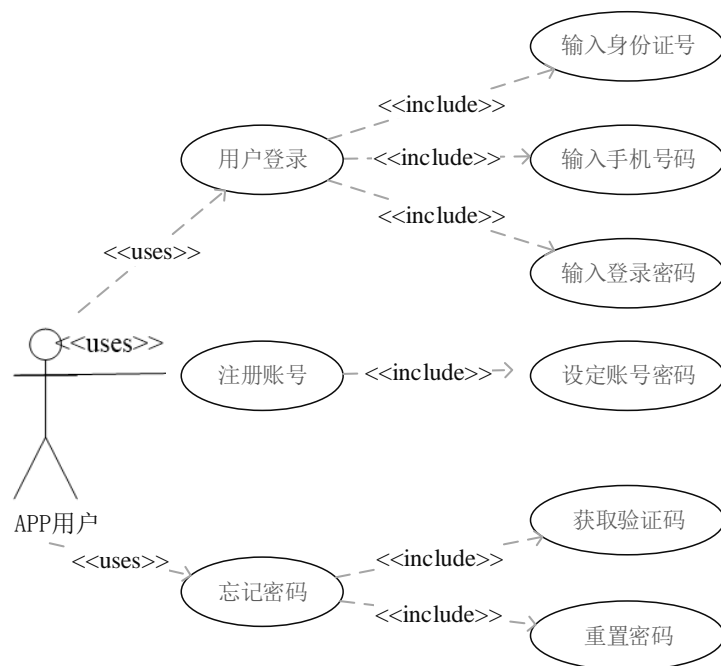


图 3-4 登录注册功能用例图

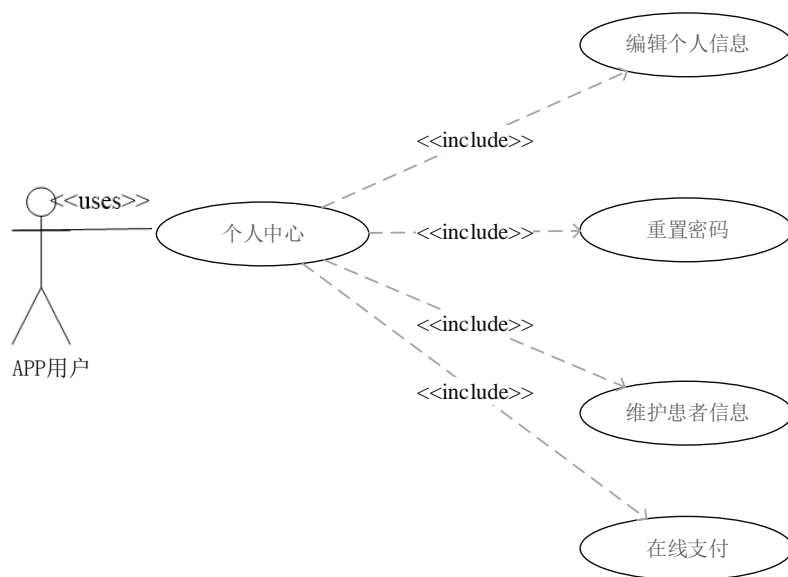


图 3-5 个人中心用例图

2) 在线预约

用户（患者）通过在线预约模块可以选择自己需要的科室医生，确定看病时间段，完成预约功能。同时可以在预约列表可以查询自己历次过往的预约记录，并支持点击某条记录可以查看详情，支持根据实际需要决定是否取消预约。在线预约逻辑关系图如图 3-6 所示。

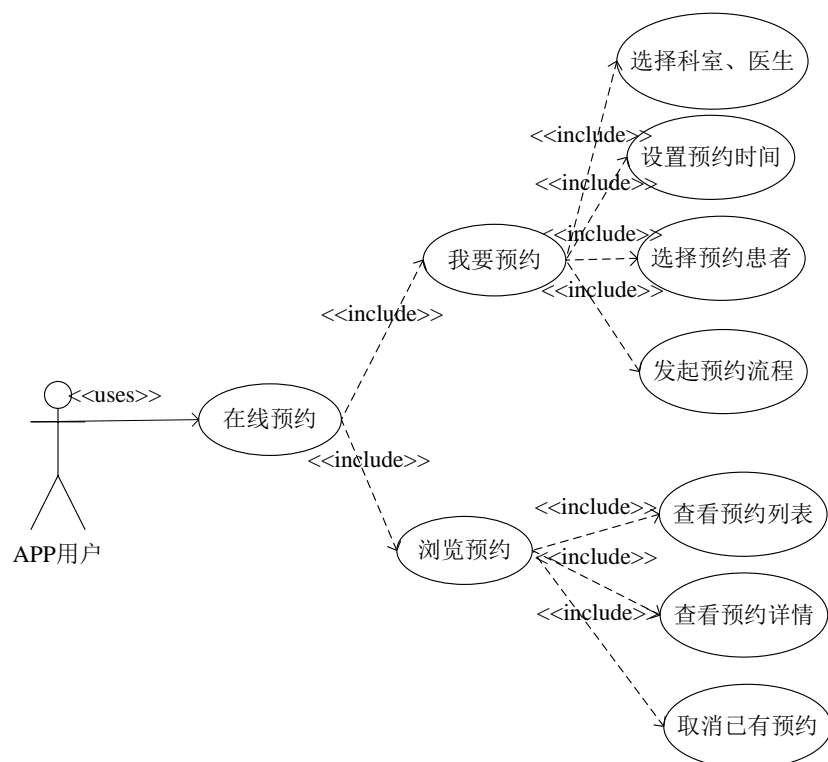


图 3-6 在线预约用例图

同时为避免信息错误和避免黄牛炒号，浪费医疗资源，预约是必须提供个人身份证信息、手机号进行验证的。过程中系统会发生短信方式来告知用户是否预约成功，并发送预约成功的科室、坐诊医生、预约时段，提醒用户（患者）提前半小时到等短消息给用户。系统上的时间定义逻辑为：预约时间范围不能超过 7 天，限制个人每天的预约次数为一次^[32]。

系统惩罚逻辑对于预约然后不负责任的失约的用户（患者），会建立惩罚制度，累计取消次数超过 3 次，会在短时间内限制网上预约，时间长度 7 天，到期后会解除限制。预约时段通常为半小时为单位进行计算，约定在该时段来就诊。预约过程需要系统显示剩余名额，超出名额后，该时段不可预约。

3) 支付处理

在线支付是实现用户（患者）在预约医生和时间段，完成费用支付的一站式

线上服务。

系统通过接入微信系统，实现预约费用的收取。用户（患者）在提交预约请求后，系统根据预约的内容生成预约订单，并按照微信平台的规则生成支付二维码进行支付。当支付完成后，微信平台会返回交易的信息，系统通过对信息的解析更改订单的状态。当支付成功后，系统更改用户（患者）的预约状态为“成功预约”^[33]。在预约时间段的 24 小时之前，用户（患者）可以发起取消预约的请求，此时系统将向微信平台发起退款请求，如退款成功，微信平台将返回相应的交易信息。用户（患者）取消预约成功，款项将于次日打入用户（患者）的原支付账号。在线支付逻辑图如图 3-7 所示。

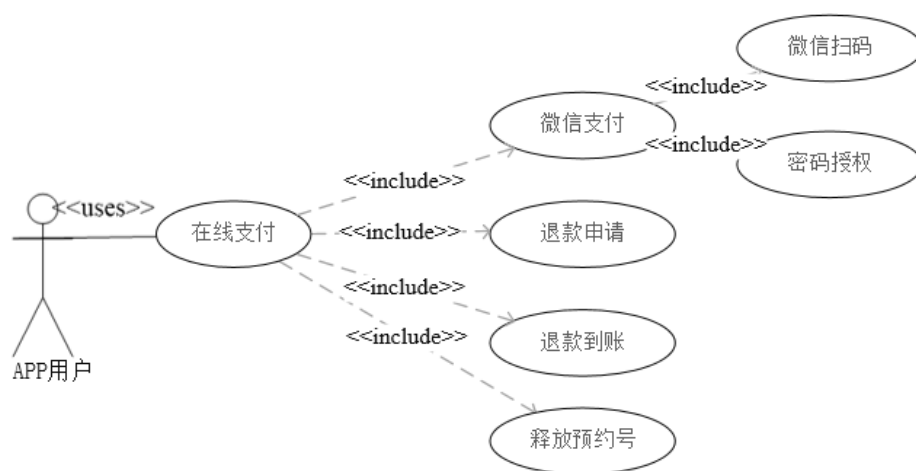


图 3-7 在线支付用例图

3.4.2 WEB 后台管理

后台管理主要是医院管理人员进行操作，可以设置科室、医生等信息，能够管理排班管理（号资源），对 APP 用户进行维护。后台管理逻辑图如图 3-8 所示。支持管理员账号登录系统，维护所有用户的预约信息。后台管理包括预约管理、用户管理、科室/医生管理、排班管理、消息通知。

1) 预约管理

预约管理对于 APP 预约是实现：

（1）输出 7 天内科室/医生排班情况给到 APP 预约端，供用户（患者）查看，进行预约。

（2）输出科室信息，医生信息、资历，从事该科的时间，历届的履历同步到 APP 预约端。

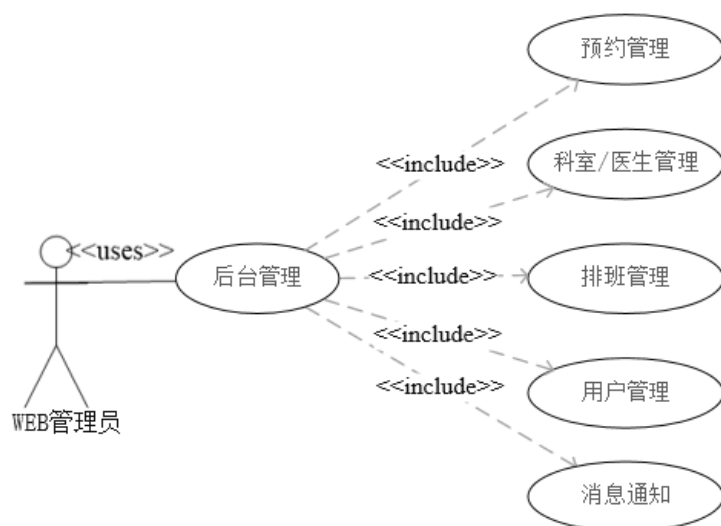


图 3-8 后台管理用例图

预约管理的后端管理是实现：

- (1) 科室医生快速查看到本科室的预约人数和时间段。
- (2) 医生可快速查看到本人的坐诊时间是否冲突，是否满足坐诊时间的要求，同时也可以看预约班次、预约类型、病人姓名、病人身份证、病人电话、预约登记者、预约时间、挂号时间。
- (3) 支持对预约信息进行快速检索和退号，查看预约详情，支持集中退号。
- (4) 一个用户（患者）只能预约同一个科室/医生同一班次的一个号。
- (5) 生成预约信息，然后调用短信接口将预约内容传递给用户。

预约管理用例图如图 3-9 所示。

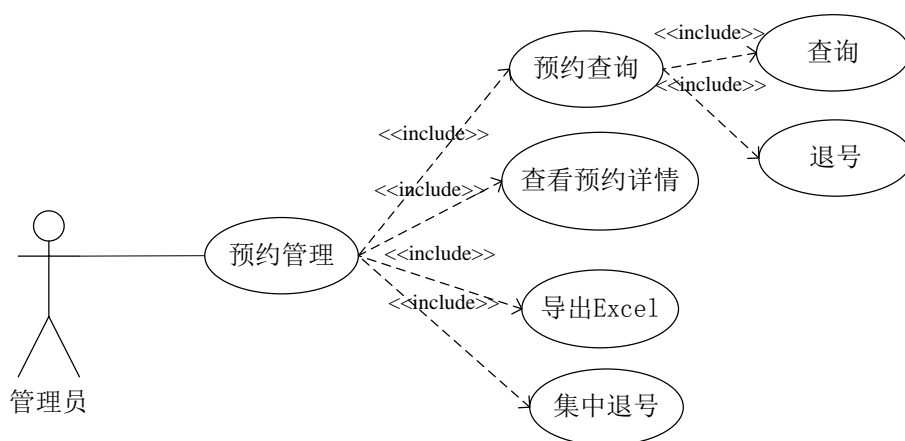


图 3-9 预约管理用例图

2) 科室/医生管理

科室/医生管理实现挂号预约的关键因素。用户（患者）使用预约挂号的系统功能的目的是预约到看病的科室和医生。

科室管理逻辑图如图 3-10 所示。科室管理支持录入科室资料，管理员使用本系统的科室管理，用户为拥有管理员权限的医护人员。

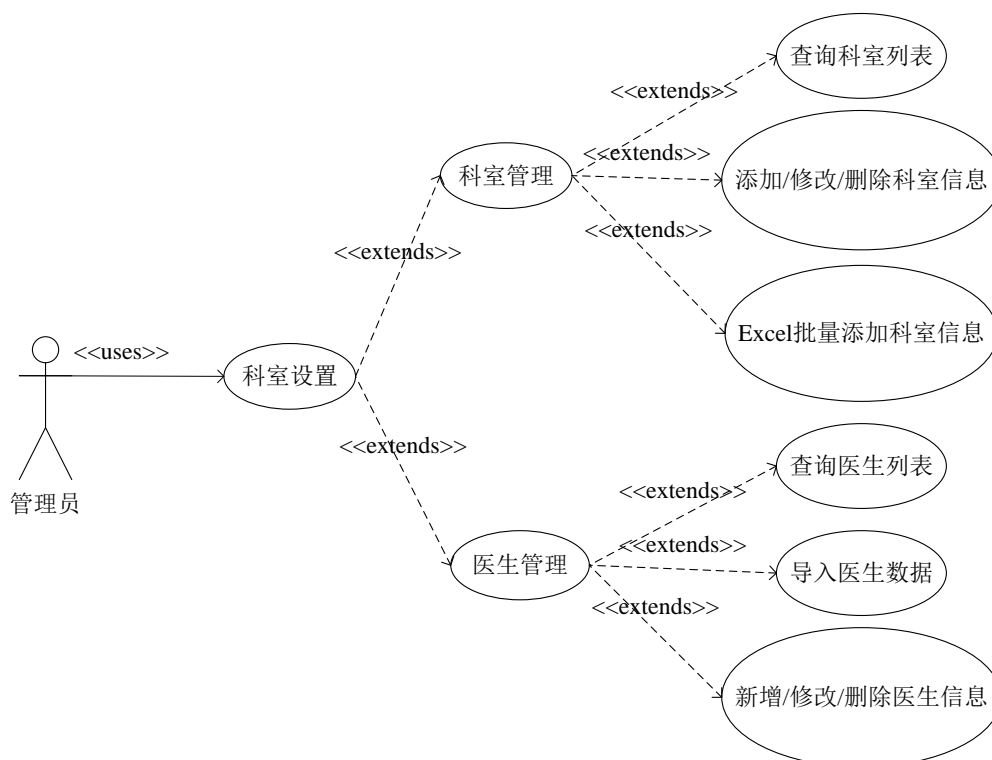


图 3-10 科室设置用例图

拥有科室管理权限的管理员成功登录系统，添加科室过程，输入信息为科室编码、科室名称。输出结果为添加是否成功、问题反馈、匹配数据问题。管理员进行科室资料的录入。基本流程为管理员登录成功并进入科室管理界面输入科室信息并点击添加。系统校验科室信息，提示是否添加成功。

医生管理是分配医生给所属科室，对不同科室的医生基本信息进行管理，支持维护医生信息。管理员成功登录系统，且该角色拥有用户管理权限。通过给科室进行分配医生，建立科室编码、医生编码等之间的关系，并提示分配是否成功。管理员进行医生归属科室的分配过程，管理员登录成功并进入科室管理界面。管理员选择需要分配的医生，并拖动到归属的科室下。管理员选择需要兼职医生，并拖动到临时归属的科室下。系统校验一个医生只能归属于一个科室，此功能校验包括兼职医生。提示是否分配成功。支持批量导入医生信息，会更新导入后的

医生列表信息，可以浏览医生的排班情况和号资源。

3) 排班管理

排班管理实现用户（患者）在某一时间可以预约到自己心仪的医生。

排班管理也称呼为号资源管理，其包括：

（1）用户（患者）提前 7 天可以查询到医院各科室/医生的排班信息（又称号资源）。排班信息包含科室、医生、职称、预约费用、日期、班次、开始时间、结束时间、放号数量等。

（2）添加号资源，可以进行批量导入和导出。

（3）排班信息包括时间段、医生资历。

（4）可以提供给用户导出历史资源表。

（5）医生仅仅隶属于某个科室。一个医生在同一班次只能有一个排班。支持查看。

排班管理（又称：号资源管理）逻辑图如图 3-11 所示。

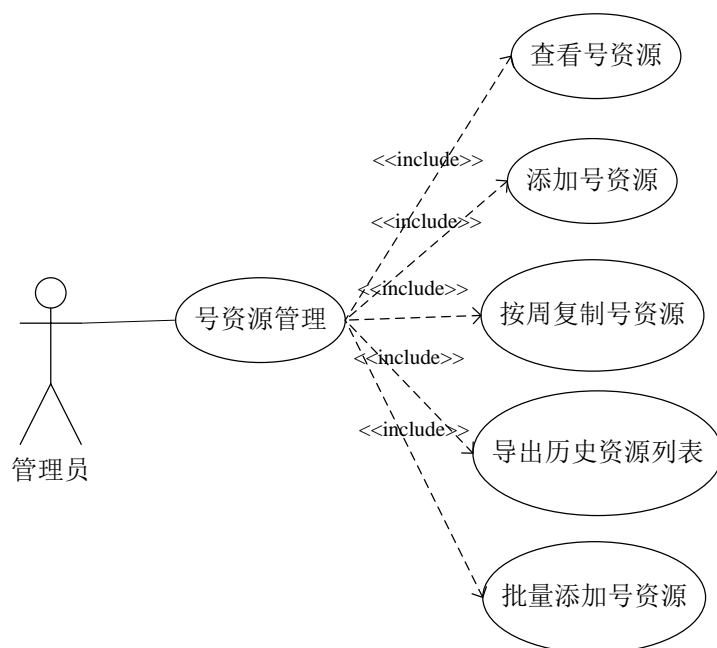


图 3-11 排班管理用例图

4) 后台用户管理

后台用户管理包括 APP 用户（患者）管理和用户（医院管理员）管理功能，功能用例如图 3-12 所示。

用户（医院管理员）作为系统用户，成功登录系统，该角色拥有用户管理权限。医护人员输入用户的信息，并点击添加，如发现信息不全，则可进行删除用

户。系统校验用户信息，提示是否添加成功。拥有用户管理权限的角色，才能进入用户管理界面，进行用户管理操作和用户管理权限的角色赋值。

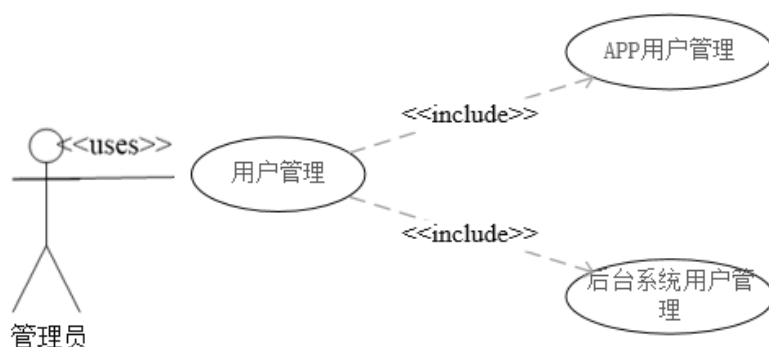


图 3-12 后台用户管理用例图

5) 消息通知

本系统主要采用发送短信通知方式来告知 APP 用户，功能逻辑如图 3-13 所示。用户（患者）成功预约/取消预约时，系统自动生成短信通知，并调用合作的短信运营商的接口发送短信。在用户（患者）已经成功预约/取消预约情况下，已配置好短信营运商端口设置，会发送短信给用户。当用户（患者）成功预约/取消预约时，系统根据预设的消息模板生成具体的短信内容，并调用短信运营商的接口进行短信发送。其中，需要确保用户（患者）在本系统预留的手机号码有效。合作的运营商的端口 URL 及用户名密码有效。

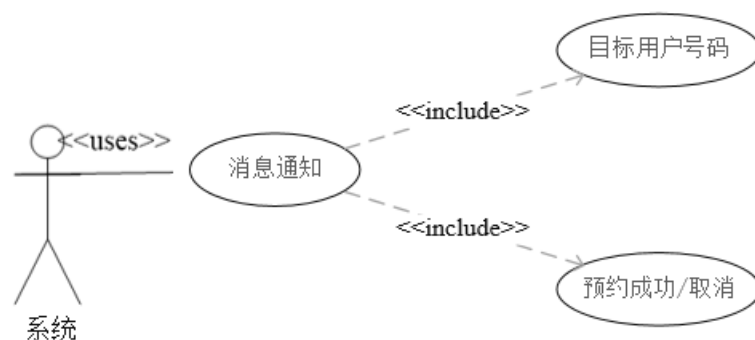


图 3-13 消息通知用例图

3.5 系统流程分析

1) 业务流程分析

挂号预约业务总体流程如图 3-14 所示。用户注册需要采用身份证实名制方式进行挂号，系统登录是身份证、手机号码的方式。登录过程进行身份验证，确保

系统安全。登录后可以选择科室、医生并根据自己的实际情况选择预约时间段，预约成功后，页面指示，选择支付平台缴费，缴费完毕后系统会自动推送预约成功与否消息的。预约成功的短消息包括了时间段、医生名、科室等信息。如果用户需要取消现有预约记录，也可同步进行。用户（患者）可以在约定时间前半小时达到医院，凭借身份证、就诊卡打印输出挂号单。然后到就诊区的分诊处扫码候诊，排队等候叫号。

预约过程中的身份信息是从用户注册信息自动带出来和匹配，无需用户重新填写，但用户（患者）需确保注册的信息与到场就医人员信息的一致。预约的医生如果因为特殊原因不能坐诊，医生内部取消坐诊后，系统会根据其他医生的实际情况匹配和分配当天实际坐诊医生。

如果用户（患者）取消预约，系统设定取消预约时间是 24 小时之前，如果在 24 小时内不能如约就诊而取消预约，不给予退费。一天之前取消退费，可以全额退款，但是违约记录累计达到 3 次以上，系统为给予锁定账号的处罚，1 个月内用户不能利用该账号进行预约。

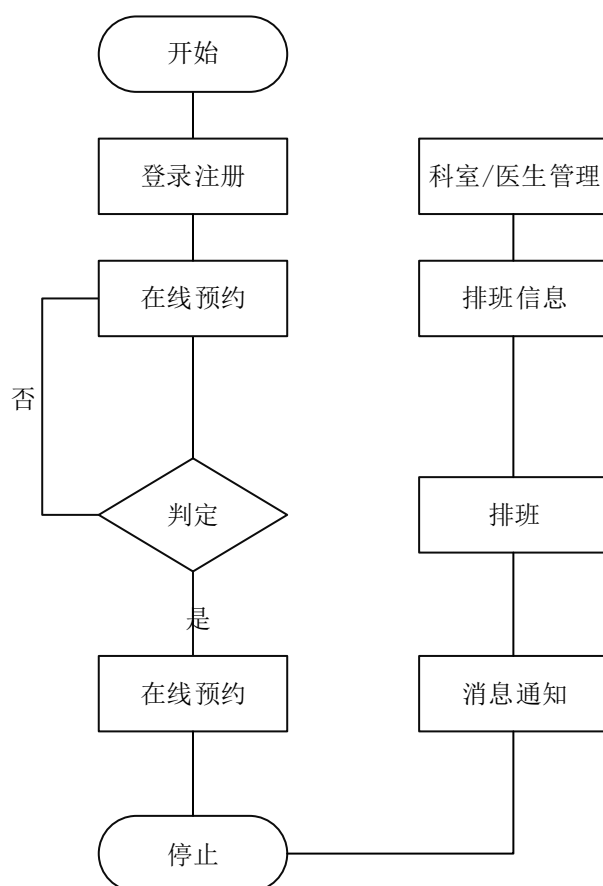


图 3-14 系统业务流程

2) 系统数据流程分析

在医院挂号预约过程会产生各种业务数据，成为了系统建设的重要组成部分，这些数据为功能管理提供资源。本文将对系统管理涉及到的用户、数据对象等进行描述。医院挂号预约过程执行不同业务管理，涉及各种实体数据，数据流程图如图 3-15 所示。其中：

(1) 用户主要包括两类，分别是患者 APP 用户、系统管理员。

(2) 数据存储对象包括 D1 患者信息，D2 医生信息，D3 系统管理员，D4 在线预约记录，D5 缴费记录，D6 预约取消记录，D7 消息记录，D8 科室信息。

(3) 数据操作：对于患者 APP 用户他主要执行的功能操作录像：1.1 用户注册账号、1.2 用户登录系统、1.3 专家记录检索、1.4 预约记录查询、1.5 用户基本资料编辑，1.6 在线预约在挂号，1.7 支持在线缴费，1.8 完成在线挂号过程，1.9 取消预约完成退费；web 端管理员用户能够完成基本操作：2.1 知名或者普通专家信息管理、2.2 对所有患者的预约记录进行管理，支持取消预约；2.3 患者管理，维护患者信息；2.4 管理员管理，维护管理员账号信息；2.5 科室设置，可以管理科室资料。

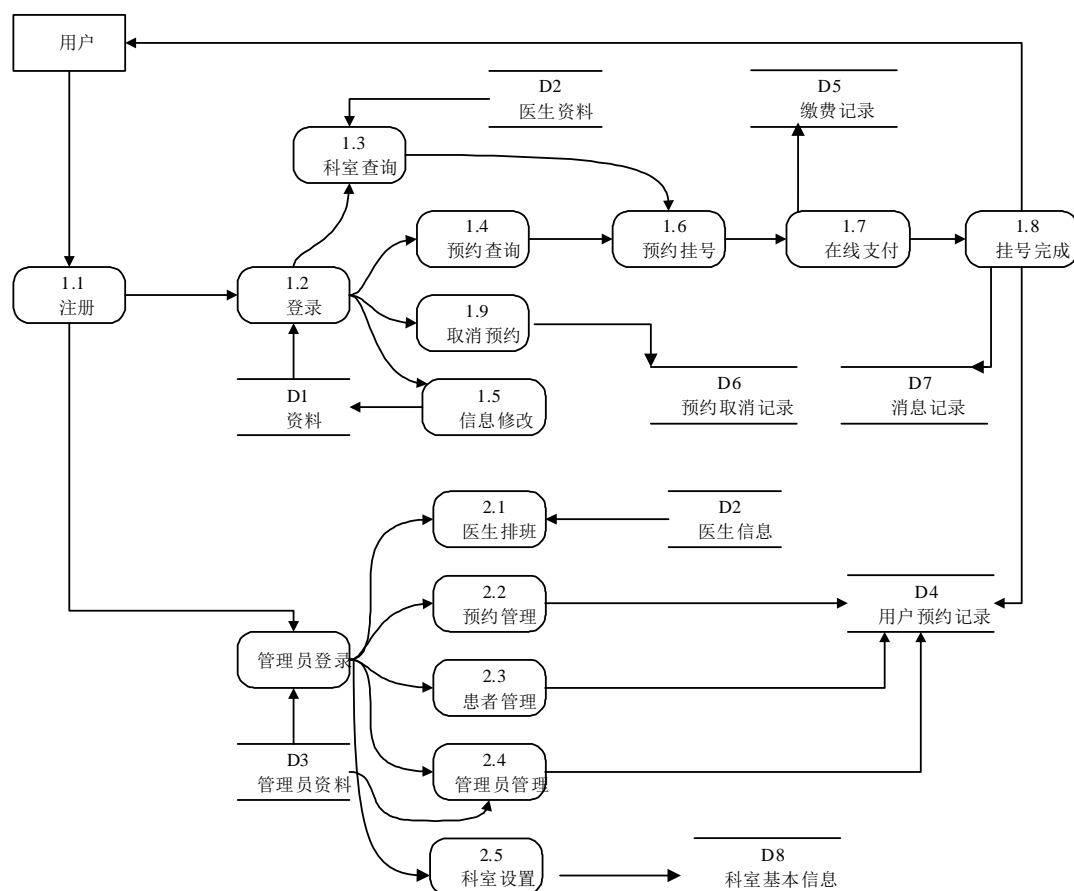


图 3-15 系统数据流程

3.6 非功能性需求分析

本文研究的医院挂号预约管理系统，需要加强在系统可靠性、安全等方面的建设，确保系统的稳定性。

1) 系统可靠性

为应对医院挂号预约管理系统每天高达 10 万人次甚至更多人次的访问量，系统的可靠性是保证医院挂号预约管理系统可持续服务的重要条件。系统不仅要配置性能高的硬件服务器，也必须搭建快速、稳定、互联互通的网络平台，确保服务器在大量用户访问时不会造成崩溃、瘫痪。同时也要配置备份数据服务器同步备份数据，为万一系统崩溃迅速恢复创造条件。同时也要在系统的后期代码、硬件设备、数据、系统可编程上加深可维护性。

2) 系统安全性

系统从安全角度考虑，应遵循 4 个原则：

(1) 网络安全设计原则，应遵守表 3-2 列举的原则。

(2) 终端操作设备身份唯一性

终端操作设备应具有唯一的身份标识，便于统一管理与维护，然后可以以虚拟身份分布式部署到网络平台，提高应用与数据的安全性。

(3) 终端操作设备软件完整性

终端设备软件能够安装自动扫描漏洞的软件，提高对周围环境的监控能力，降低遭受外部攻击与恶意修改的风险。在设备软件升级过程，需要加入各项参数的验证，确保操作员信息的完整。

(4) 系统设计防止反汇编、反调试的能力

软件应具备防止反汇编、反调试的能力。

为实现系统安全性，需要遵循以下措施：

首先需要满足网络硬件设备的核心参数需要，具有强大的业务处理能力。其次，需要对存储业务数据、应用程序、文件等功用不同的服务器分配到不同安全区域，设置不同的安全防控级别。再次，设置合理的网络 IP 地址策略，满足路由和交换过程所有数据的正常、安全传输。最后，设计冗余系统线路和网络设备，确保当某条线路发生异常情况，可以用备用线路来提高系统的可用性。同时需要加强安全设备的防护管控工作，利用杀毒软件与防火墙、日志等措施来提高系统安全性，加强远程用户安全访问网络的管理^[35]。

表 3-2 网络安全设计原则

原则	说明
最小权限数原则	每个用户都有属于自己的角色，角色基本能够决定用户操作的模块权限，并将进程处理过程设置为最小权限，从而可以防止用户主观破坏或者操作失误导致系统异常的情况发生。
权限分离原则	维护基础数据与核心业务数据，必须由高级管理员来完成，并且限制管理员的账号数量不超出 5 个。普通的业务数据维护可以交给一般用户完成，他们分别保存在不同权限数据表中，实现了权限分离的目的 ^[34] 。
外来代码编译安全设计原则	通过接口调用外部服务的过程中，必须核实相关方法与插件来源的合法性，并建立身份识别机制，避免相关代码带来的破坏性问题。
标准符合性原则	应根据国家标准、行业标准规定开发软件。
可扩充外接接口设计	预留插件或接口，方便功能扩展增加预备入口、资源。

3.7 本章小结

本章结合医院现有的电话预约或者现场预约的模式，已经无法满足日益增长的需求，患者抱怨意见多的情况，提出了基于移动互联网下，选取医院挂号预约系统作为研究对象。结合某医院挂号的现状分析，医院挂号方式存在的问题，建立标准医院挂号预约管理系统的设计与实现方案。

本文主要介绍医院挂号预约管理的设计目标、流程分析和功能需求：

- 1) 对医院挂号预约管理设计需求进行了阐述。
- 2) 对系统的技术可行性、经济上的可行性、操作简单性进行简述。
- 3) 分析了系统的角色包括了 **WED** 管理和患者用户，阐述了两者的功能。
- 4) 介绍了医院挂号预约管理系统中 **APP** 挂号预约和 **WEB** 后台管理的需求。
- 5) 分析了业务流程和数据流程。
- 6) 指出医院挂号预约系统的可靠性和安全性等非功能性需求。

从而保证系统能够根据实际需求，在功能操作方便性，网络速度的快速性，系统的安全性进行设计。

第四章 挂号预约管理系统设计

4.1 系统应用架构设计

挂号预约管理系统独立部署运行，通过约定的标准化接口与医院 HIS 系统对接，院方管理子系统即由医院系统实现，向订单子系统提供院方的必要信息。系统在表现层是采用 Android，实现用户的界面操作设计。在业务层是使用 Java,数据库选是 MySQL^[41]。应用架构如图 4-1 所示，由多个层次构成，整合了 APP 和后台管理，分别为表现层、业务逻辑层与数据层，每个层次负责对应的功能，层次间紧密合作。

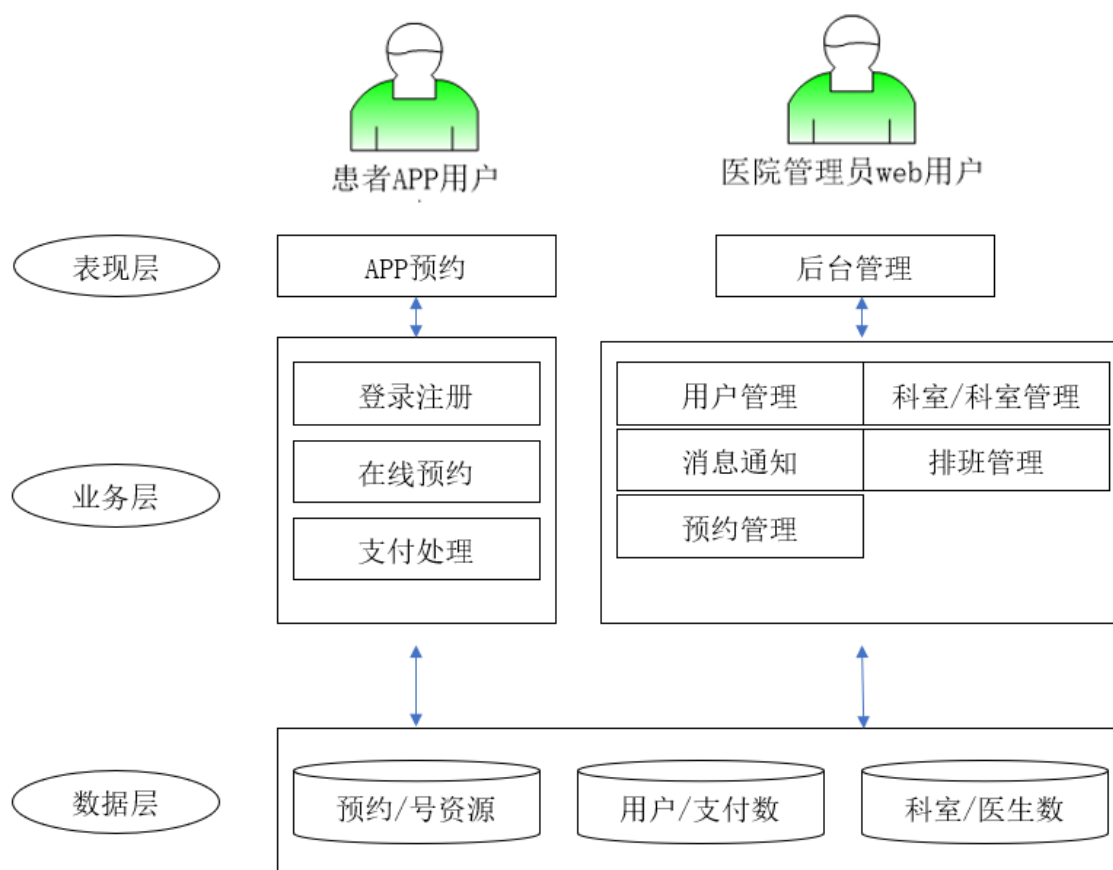


图 4-1 WEB 端技术架构

1) 数据层

数据层主要负责存储预约/号源、在线支付、科室、医生、用户、用户（患者）档案等各种数据，其关键是数据安全存储，合理组织数据格式，对数据进行管理和维护，保障数据的正确输出。数据层是应用架构的底层，也是整个系统最关键

和核心的数据存储空间。也是为系统正常运行和服务于用户的内在依据。数据层可以保存数据，也以缓存方式提高服务效率，明确缓存对象与缓存有效时长，确定缓存位置。是否采用缓存、设置多大缓存等需要分析数据使用频率情况、文件大小，缓存需要定期清理^[40]。数据层可以向业务逻辑层提供服务接口 API，提供统一数据访问接口，采取不同策略来保障网络畅通。数据层可以为业务层提供数据，并与医院其他业务系统进行对接，提供数据共享服务，帮助进行数据快速交换。另外数据层具备比较完善的备份机制。

2) 业务层

该层主要负责处理功能管理过程的业务执行逻辑，及时响应用户请求，对数据层的数据进行准确性、一致性检查，对业务分支情况进行处理，可以返回结果到表现层。业务层包括 APP 和后台系统两部分，APP 包括用户（患者）登录注册、在线预约、在线支付等功能；后台系统包括预约管理、科室/医生管理、用户管理、排班管理、消息通知等功能。

3) 表现层

用户直接能够查看到的界面，是人机交互的核心所在，其由用户端的 APP 操作界面和后台系统维护界面，维护业务数据，开展各项功能管理。用户采用 B/S 模式访问系统，不需要升级系统就能够查看到更新后的系统功能。系统表现层的用户包括 APP 用户（患者）用户、后台管理员用户。表现层可以浏览到系统布局、文字图片信息等，通过标准化、规范化设计，保持界面的简洁大方，方便使用。

4.2 系统技术架构设计

医院挂号预约管理系统分成 APP 挂号预约与 web 后台管理，技术架构分成 web 架构与 APP 架构。搭建系统时，独立设计成两个子应用程序，系统间的数据和功能可以相互调用、相互共享。系统开发工具为 myEclipse2019，数据库为 mySQL。

1) WEB 端技术架构

医院挂号预约管理系统 web 端实现用户通过浏览器接入系统，提供 B/S 服务模式，WEB 端技术架构如图 4-2 所示。系统为能集成化开发，整合了 SSH 框架资源，采用面向对象编程语言 java，支持多层次开发。

系统前台页面设计控制插件用 jquery，动态输出用 jsp，页面样式用 CSS，通过控制各种组件与文字的输出效果来实现一次登录、多系统自由切换。系统需要进行用户登录身份识别与验证。页面局部动态加载更新功能是通过 Ajax 实现的。提交的 form 表单数据类型、长度的合法性是 JavaScript 验证的，除此之外 JavaScript

还可以进行页面跳转控制。**Struts2** 框架通过自身标签库控制输出效果来实现与 **JSP** 的整合，响应请求和识别。通过 **XML** 定义数据格式和交换方式，**JSP** 快速动态输出和展示数据。**Spring** 框架整合 **Hibernate** 与 **Struts2**，负责 **MVC** 模式的相应功能。**Web service** 独立封装好完成某项事务的功能方法，支持接口调用，提升效率。**JDBC** 程序代码，映射实体与数据库表结构的数据：挂号、医生、号资源、科室、支付等各种业务数据的安全存储。

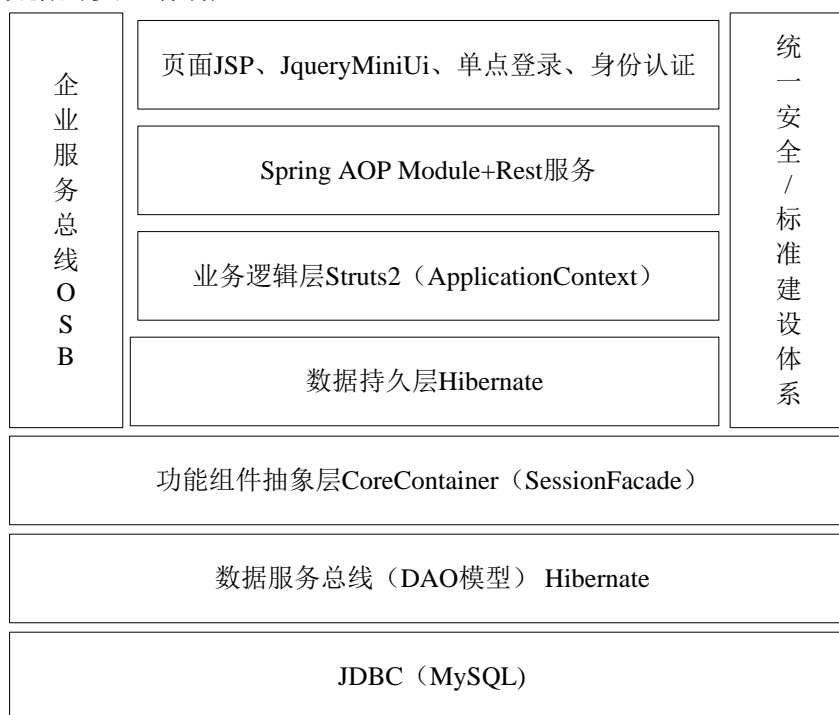


图 4-2 WEB 端技术架构

2) APP 端技术架构

医院挂号预约管理系统 APP 端技术架构如图 4-3 所示。

(1) 视图展示：所有用户通过访问 APP 功能界面，实现在线支付、预约记录、取消现有预约，在线预约并支付等功能。

(2) 业务逻辑：通过 **Presenter** 数据交互逻辑和 **Model** 数据存储，实现了数据存储对象封装，控制数据交换过程，并且承载上层的数据通信和页面跳转，从实现业务基本管理功能。图片处理过程使用 **fresco** 技术。

(3) 基础框架：提供了包括网络模块、普通界面 **Common UI**、基础工具类 **Base Utils**、**IM** 等基础组件和服务接口。支持使用 **http** 通信过程的 **OkHttp** 技术，以此稳定、可持续性的通讯。

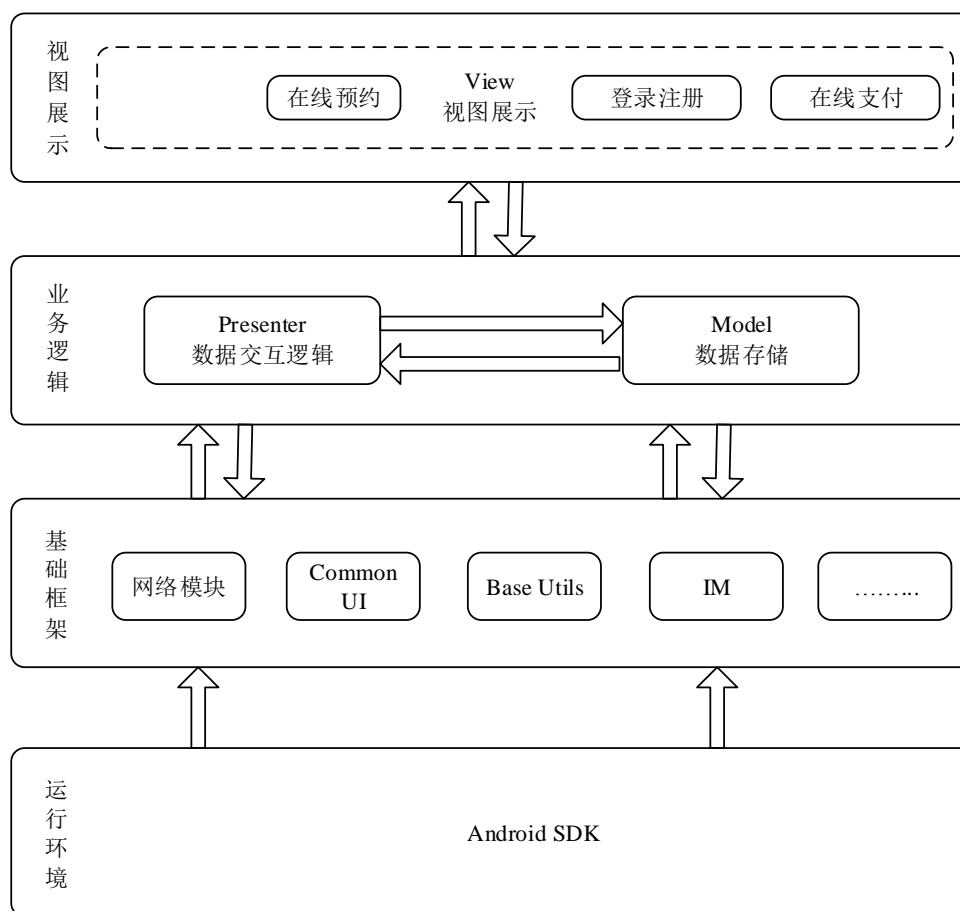


图 4-3 APP 端技术架构

（4）运行环境：通过 Android SDK 实现基础运行环境，来对 java 编程语言进行编译，实现系统正常运行。

4.3 系统网络拓扑设计

医院挂号预约管理系统通过提供硬件设施的部署来搭建稳定的网络拓扑，网络拓扑结构图如图 4-4 所示。预约挂号系统配置包括应用程序、数据库以及备份等功能服务器在内的各种类型服务器。考虑到数据库服务器在存储科室、医生、挂号、支付等数据长期积累带来的存储压力，需要配置大容量存储空间，并提供数据访问接口。通过 tomcat 容器进行部署应用程序服务器发布程序包，不定期进行升级，提高维护效率。系统所有数据都可以利用备份服务器进行备份，避免意外情况发生，能够及时恢复数据。系统提供短信服务器，可以向预约客户推送预约、支付相关情况短信。系统数据库可以同步更新，确保 APP 与 web 端数据的一致性。

系统为保证每个患者浏览到的界面基本一致，需提供统一的外部预约接口，保证患者看到最新可预约的医生、号源数量、时间段情况。系统通过防火墙对内

外进行隔离保护，避免内网 HIS 数据遭到外部侵入，支持内外网数据同步的更新，支持以 webservice 方式实现跨平台、跨系统的数据交换和资源共享。支持系统批量导入各种数据，维护个人基本信息，在线及时预约和取消预约等操作。患者到医院后可以打印出挂号单，然后去就诊区排队等候。

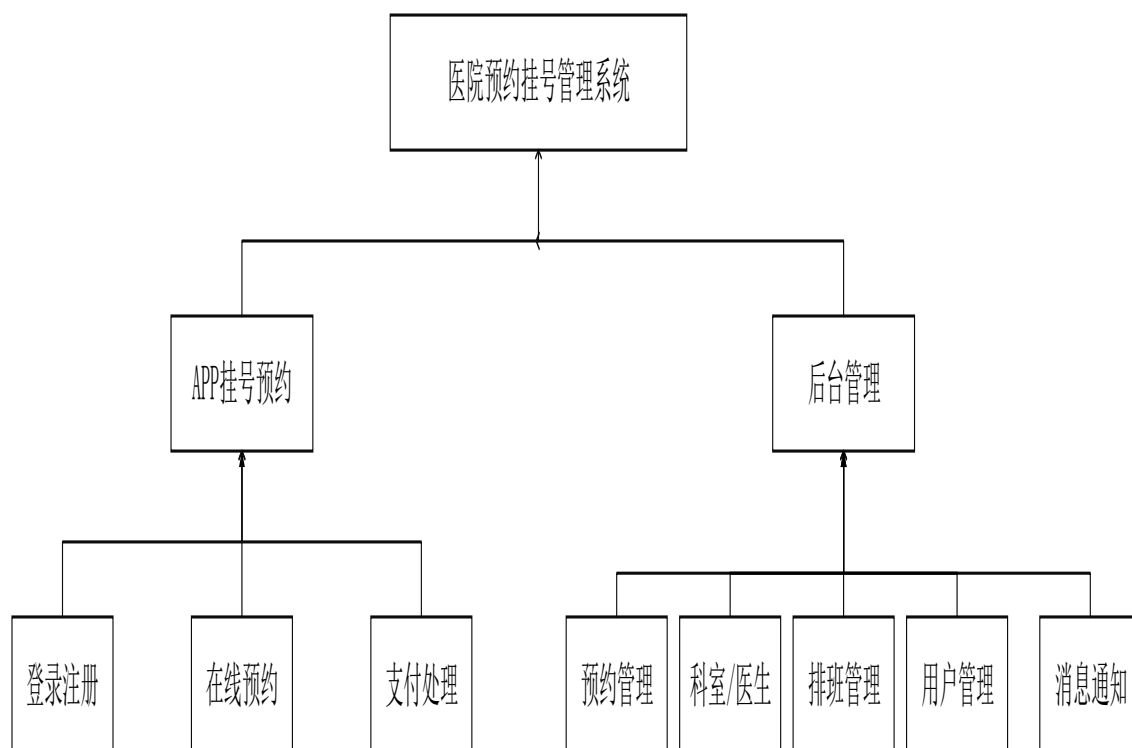


图 4-4 系统网络拓扑图

4.4 系统功能架构设计

本系统重要的功能需求包括 App 预约挂号：用户（患者）建档、支付处理、在线预约和后台管理：预约管理、科室/医生管理、排班管理、用户管理、消息通知。用户管理功能可以建立患者档案信息；对用户权限进行控制。科室/医生管理负责对科室信息和医生信息进行管理。排班管理可以对医生会诊进行安排。预约管理支持用户在线预约，并对预约信息进行处理。支付功能管理用户支付信息和退款信息。消息功能提供短信通知，推送微信消息等。各子系统包括若干模块，结构示意图如图 4-5 所示：

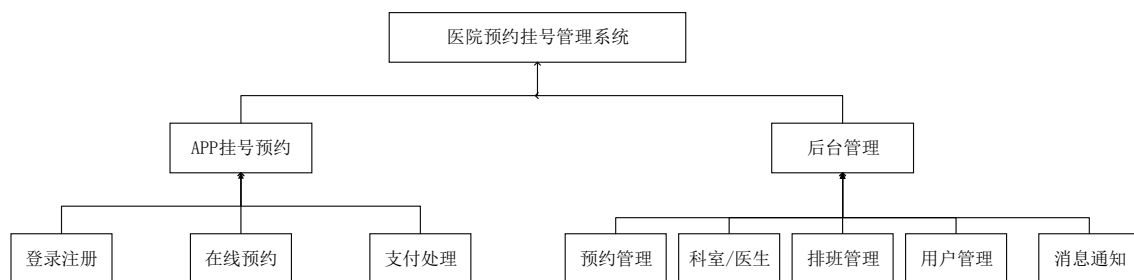


图 4-5 功能模块结构示意图

4.4.1 APP 挂号预约

APP 挂号预约子系统，如图 4-6 所示，包括了登录注册、在线预约、在线支付三个功能模块，为完成各功能的需求，设计了各子功能的支撑子域。

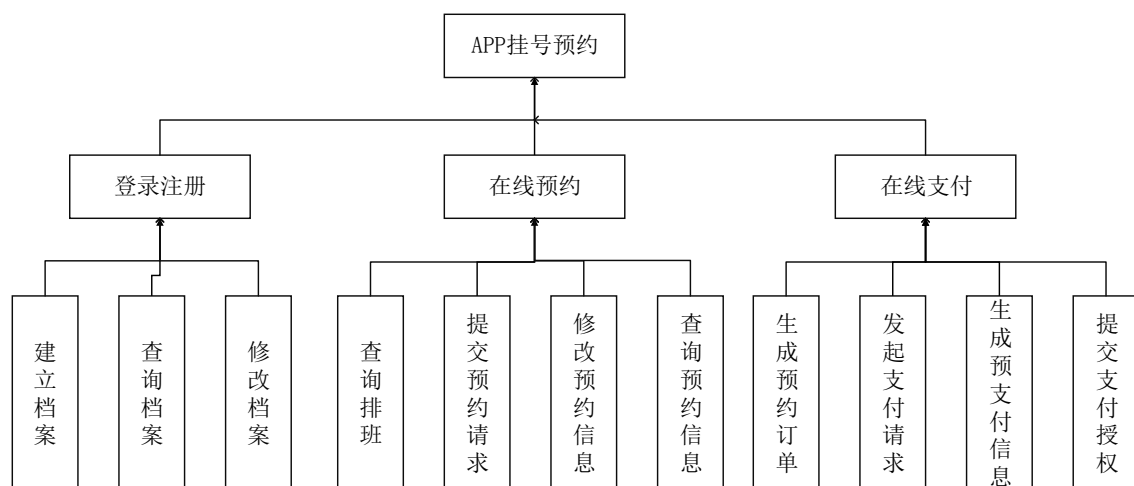


图 4-6 APP 挂号预约

1) 登录注册

登录注册，是针对用户（患者）作为应用对象的。

（1）用户（患者）在显示的界面注册，建立电子档案，保存用户基本资料，是用户（患者）使用预约功能的基础。电子档案为每个用户（患者）分配一个唯一的编号，作为使用本预约系统的身份识别，通过数据库设计存储于 HIS 服务器。

（2）用户（患者）本人可在界面上查看用户（患者）的电子档案，用户（患者）的电子档案数据的取数来与 HIS 服务器之前的登记和储存的信息。但是有权限分级管理。

（3）用户（患者）本人可通过网页界面的特定修改档案的界面，修改自身的信息，通过数据库设计存储于 HIS 服务器。

2) 在线预约

用户（患者）为满足就医的时间及看到心仪的医生的需求，会首先查询科室/医生的排班情况，再发起预约。

（1）查询排班信息，用户（患者）可以在系统上通过输入需预约医生的姓名、需预约科室的名称中的任意条件，对相关科室/出诊医生的排班信息进行查询。

（2）用户（患者）提交预约信息时可以选择出诊医生，查询该医生该班次的预约的具体放号信息。可对预约状态为“可预约”的排班明细信息，进行门诊预约。

（3）发送预约。

（4）查询预约信息，用户（患者）在预约成功后，可以通过用户（患者）姓名、身份证号码等条件查询预约情况。

（5）修改预约信息，用户（患者）可以进行修改或取消预约操作。修改或取消预约，必须在该已预约班次时段（看诊时间）的 24 小时之前进行。修改或取消预约成功后，系统同样会发送短信通知。

3) 支付处理

（1）支付处理包括生成预约订单生成、发送支付请求、生成预付交易、提交支付授权、交易通知等功能。系统通过调用支付系统进行操作，通过服务接口将结果返回给业务系统。当消息反馈异常时，用户需要进行查询，同步支付结果。

（2）预约订单生成，用户提交预约成功后，选择支付方式，如银行支付、微信、银行卡等。在业务界面上，直接生成选择支付页面，点击具体支付方式后直接进入支付页面。

（3）成功提交预约请求后，会自动动态占用预约号.发送支付请求，以接入微信的支付方式。具体的操作流程是，提交预约请求之后，系统生成支付二维码。用微信扫描二维码进行支付。用户（患者）输入密码进行授权。微信平台验证交易，系统根据微信异步通知更改交易状态。最后系统通知用户（患者）付款和预约结果。支付过程，预支付订单保留 10 分钟，超时自动取消，同时占用的预约号也自动释放。在线支付过程，输入数据为病人 ID、标价金额、交易时间、交易类型。输出结果为：病人 ID、交易单编号、业务结果、错误代码、错误代码描述、交易类型。付款成功后会自动改变支付状态，并更新支付相关的所有参数值，并记录交易前的信息，如果中途发生异常，可以帮助回滚，确保交易数据的前后一致与安全。

（4）提交支付授权，输入密码确认交付，借助第三方平台，完成用户请求授权、提交支付授权、验证支付信息。提交预约请求之后，待支付订单将为用户保留 10 分钟，超时未完成支付，订单将自动取消，同时预约也将取消。

(5) 交易通知, 发送交易短信, 在交易时, 可临时更改接收预约短信的电话号码, 系统自动带排班信息, 包括科室、医生在对应的科室、对应的时段有空闲的情况下, 反馈至 HIS 服务器, HIS 服务器信息对比后, 默认此次交易成功时, 预约短信将会发送至临时填写的手机号码, 或微信通知。

(6) 取消预约, 用户想要取消预约申请, 需在预定时间内发起取消申请, 取消通过中间层分发取消预约接口, 由 HIS 服务器确认, 递交需求到第三方平台。

(7) 发起退款请求, 收到第三方平台的请求后, 在预约时段开始前 24 小时可以发起退款申请, 款项退回支付的原微信账号。根据微信的规定, 申请退款成功, 退款金额会在 24 小时内退还。完成退款后, 预约订单状态将改变。

(8) 系统根据退款结果更改用户(患者)的预约状态。并释放被占用的预约号, 系统通知用户(患者)退款结果及预约取消结果。

4.4.2 WEB 后台管理

WEB 后台管理模块, 如图 4-10 所示, 包括预约管理、后台用户管理、科室/医生管理、排班管理、消息通知等功能模块, 为完成各功能的需求, 设计了各子功能的支撑子域。

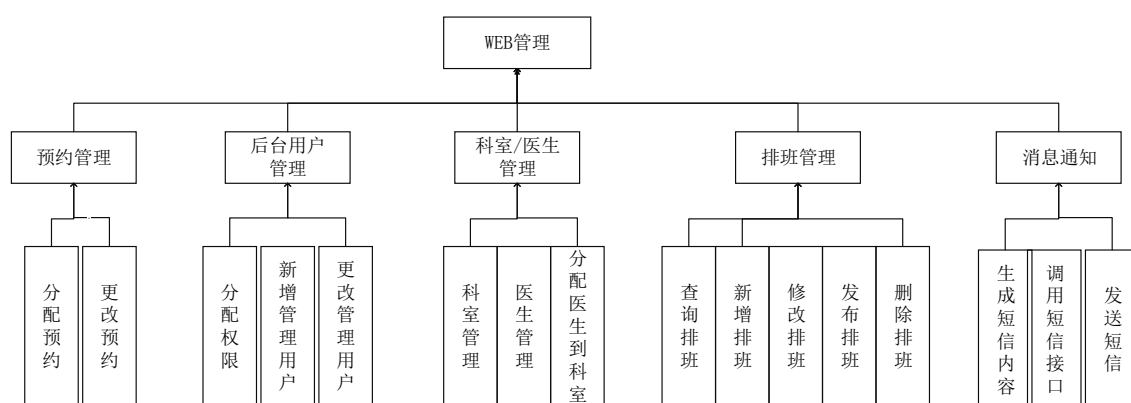


图 4-7 后台管理

1) 预约管理

(1) 系统在收到 APP 挂号系统传递过来的用户(患者)预约请求后, 对收到的预约请求进行自动分配到相关的科室/医生, 并形成排班通知到科室/医生。

(2) 系统在收到 APP 挂号系统传递过来的用户(患者)预约取消请求后, 需自动取消已经分配的科室/医生, 并重新接受新的分配预约请求。

2) 后台用户管理

(1) 提供给医院系统管理员使用的, 医院系统管理员通过网页界面对用户进

行管理，病历填写。系统可给不同职位的医护人员分配不同的系统角色。不同的系统角色拥有不同的系统权限，通过数据库设计存储于 HIS 服务器。

(2) 更改用户，需特定权限医护人员和进行，在网页界面上更改用户资料，通过数据库设计更改用户接口分发，记录更改用户资料信息存储于 HIS 服务器。

(3) 新增用户，需特定权限医护人员和进行，在网页界面上新增用户资料，通过数据库设计新增用户接口分发，记录废除用户资料信息存储于 HIS 服务器。

3) 科室/医生管理

(1) 科室管理，主要任务是管控科室的信息，功能包括：管理科室资料，科室管理接口分发、记录科室资料，包括添加、修改，删除科室的基本信息，建立索引，包括：科室 ID、名称等。可以查看科室基本信息和科室医生信息等。

(2) 医生管理，维护医生基本信息。在管理医生的信息上，应先录入科室和医生的资料，再进行分配医生所属科室的操作，每个医生只能归属于一个科室。当一个医生已经拥有所属的科室，再把他分配到另外科室时，需移除他在原有科室的记录，并把他添加进新科室。

(3) 分配医生所属科室，包括分配医生所属科室、分配医生所属科室、记录医生所属科室，以此实现医生轮值和临时调班管理，在管理医生轮值信息方面，主要功能包括：a、医生排班信息维护，管理医生轮值信息。b、排班查询，支持检索医生、科室的排班情况。

3) 排班管理

(1) 排班管理功能可以对排班信息进行增删查改操作，发布排班信息等。

a、排班的关键在于设计模板，合理的模板设计可以大大减轻排班人员的工作量，故而在系统中将模板及正式排班合并到同一套表中，模板的日期设置为系统启用之前的日期用于区别正式排班。

b、排班的另一关键点在于生成排班时需要判断当前是星期几（因日常出诊中习惯用星期几来判断具体日期而不是用几月几日来界定），此处使用了两个函数来进行处理星期。

c、因排班时需要直观地将排班表列出，但由于排班的时间段是动态的（比如可以一次性排一周的，也可以一次排一个月的班。），所以显示的表格采用动态控件生成字段处理^[36]。

(2) 在查询排班信息，医生通过输入医生/科室名称进行查询，系统通过服务器和 HIS 服务器调用医生的排班信息。每个医生在一个班次时段里只能有最多一个排班。

(3) 在新增排班信息，由空闲医生的提交可接受门诊的时间，医生可以选择

没有安排排班的班次时段，新增该班次时段的排班信息，包括科室、医生、诊室、开始时间、结束时间、排班类型等。新增成功后，需要发布排班信息，方能生效。以供用户（患者）对该班次进行门诊预约。系统在 HIS 服务器中选取没有安排的时段、科室、诊室、医生的信息，信息自动比对后提交，新增排班接口分发信息至 HIS 服务器，由 HIS 服务器记录新增排班信息，并自动调用。

（4）修改排班信息，医生可根据个人的情况，科室的安排，选择已经存在的排班信息，进行修改，修改后通过修改排班接口分发到 HIS 服务器，由 HIS 服务器记录排班信息成功之后需要再次发布排班信息，才能生效。

a、修改排班的关键在于要判断修改后的排班诊室及时间段是否存在有效排班，存在则不允许修改。

b、修改排班后需要发通知预约用户（患者）改诊信息。

（5）删除排班信息，医生可根据个人的情况，科室的安排，选择已经存在的排班信息，进行删除，删除后通过修改排班接口分发到 HIS 服务器，由 HIS 服务器记录排班信息成功之后需要再次发布排班信息，方能生效。

删除排班时需要检查是否有用户（患者）已预约，有的话不能删除：

（6）发布排班信息，信息来源于新增排班、修改排班、删除排班，由系统通过获得的信息发送到修改排班状态接口进行分发，由 HIS 服务器记录并完成更新发布。临时调班管理可以进行医生调班操作，对调班信息进行维护，可以通过医生 ID 号、班次、等关键词进行查询。

4) 消息通知

（1）消息模块设计，这个模块的业务是负责整个项目所有的消息出入口，包含短信发送，APP 推送。

（2）生成短信内容，在用户成功预约/取消预约时，系统会根据预设的消息模板，在交易时，可临时更改接收预约短信的电话号码，反馈至 HIS 服务器，HIS 服务器信息对比后，默认此次交易成功时，预约短信将会发送至临时填写的手机号码。生成短信内容的通过调用合作的短信运营商的接口，发送短信通知。

（3）短信模板内容可根据消息配置中心进行设置。

（4）发送消息模块，包括生成消息的内容、调用接口、发送消息等。

（5）在用户成功预约/取消预约时，系统会根据预设的消息模板，生成消息内容，并将通知发送给用户。

（6）消息模板内容可以消息配置中心进行设置。

4.5 系统接口设计

预约方式主要包括是 APP 预约, 通过对数据获取, 分析, 发送到中间层, 进行暂时储存, 识别, 发送到用户管理、预约管理模块, 来进行实现挂号预约功能。

各子系统之间的交互接口清单及功能概要如表 4-1 所示:

表 4-1 各子系统之间的交互接口清单及功能概要

子系统	模块	接口	功能概要
APP 挂号预约	登录注册	用户信息接口	为后台管理系统提供用户基本信息
	在线预约	预约接口	为后台管理系统提供预约信息
		排班查询	调用后台管理系统提供的排班信息
	支付	支付接口	提供多种方式的扣费功能
		退款接口	提供多种方式的退费功能
后台管理	预约管理	预约信息接口	管理 APP 预约系统预约信息
	后台用户	后台用户接口	管理用户权限和信息
	科室/医生	科室结构查询接口	为 APP 预约系统提供科室信息
		医生信息查询接口	为 APP 预约系统提供医生基本信息
	排班	排班信息查询接口	为 APP 预约系统提供医生排班信息
	消息处理	短消息发送接口	为 APP 预约系统提供发送手机短消息通知的功能
		微信消息发送接口	为 APP 预约系统提供发送微信消息推送的功能

4.6 系统数据库设计

数据库是预约系统的基础和核心部分所在，其负责系统的整个数据流的传输和交换，设计质量的好坏，速度优化的快和慢，直接影响到系统的运算效率，关系到用户体验的感受。数据库设计是根据医院业务的需求，用户的需求、信息处理需求、外部平台信息对接交换口的需求等，可以保证数据的一致性^[41]。

数据库设计主要是构造数据库概念、物理等两种模型，解决数据存储问题。在数据的信息交流和传输中，实施适当的优化，以便能长久有效的管理数据，并且能响应足够多用户请求的，不出现断卡，满足各类用户的应用需求，其中包括信息管理要求和信息应用需求，以及数据应用，系统操作的要求。数据库采用结构化设计。结构化设计是对概要设计的进一步优化，其目标是为软件结构逻辑中每个模块提供可给程序员编程实现的具体算法^[42]。

本医院挂号预约管理系统中需用到五种数据流，包括预约信息流，登录数据流，科室/医生数据流，外部平台接口数据流。预约管理数据流主要是储存用户（患者）的信息，用户（患者）挂号预约时填写的各种信息。登录管理数据流包括用户名、密码、权限、后台管理员权限、授权信息等。科室/医生数据流包括医生名称、科室、预约人数等信息。外部数据流主要是消息运行商接口和微信平台接口调用。以下是数据库的数据流大概方向设计：

用户表包括用户名、角色表包括角色编号、角色功能表包括角色功能编号、系统功能表包括功能编号、爽约记录表包括排班号，预约号明细表、检查信息表包括排班号，预约号明细表、患者档案表包括用户（患者）编号、挂号表包括排班号，排班明细表、支付表包括支付 ID，支付明细表、支付配置表包括公共账号 ID、排班表包括排班次编号、医生表包括科室编号、医生编号、排班次定义表包括科室编号、班次编号、科室表包括科室编号、预约类型表包括排班次编号、医技类型表包括医技室编号、检查项目表包括科室编号、检查项目编号。

4.6.1 数据库表结构设计

本数据库包括科室信息表 T_ORG、医生信息表 T_DOCTOR、系统消息表 T_MESSAGE、支付信息表 T_PAYMENT 等八个数据表为例进行设计。实体关系图如图 4-8 所示。

1) 科室信息表 T_ORG

科室信息表结构如表 4-2 所示，负责保存医院所有科室基础信息，包括描述、名称以及编码等字段。

2) 医生信息表 T_DOCTOR

医生信息表结构如表 4-3 所示，负责保存不同科室、不同级别的专家与医生信息，包括简介、照片、入院时间、名称以及编码等字段。

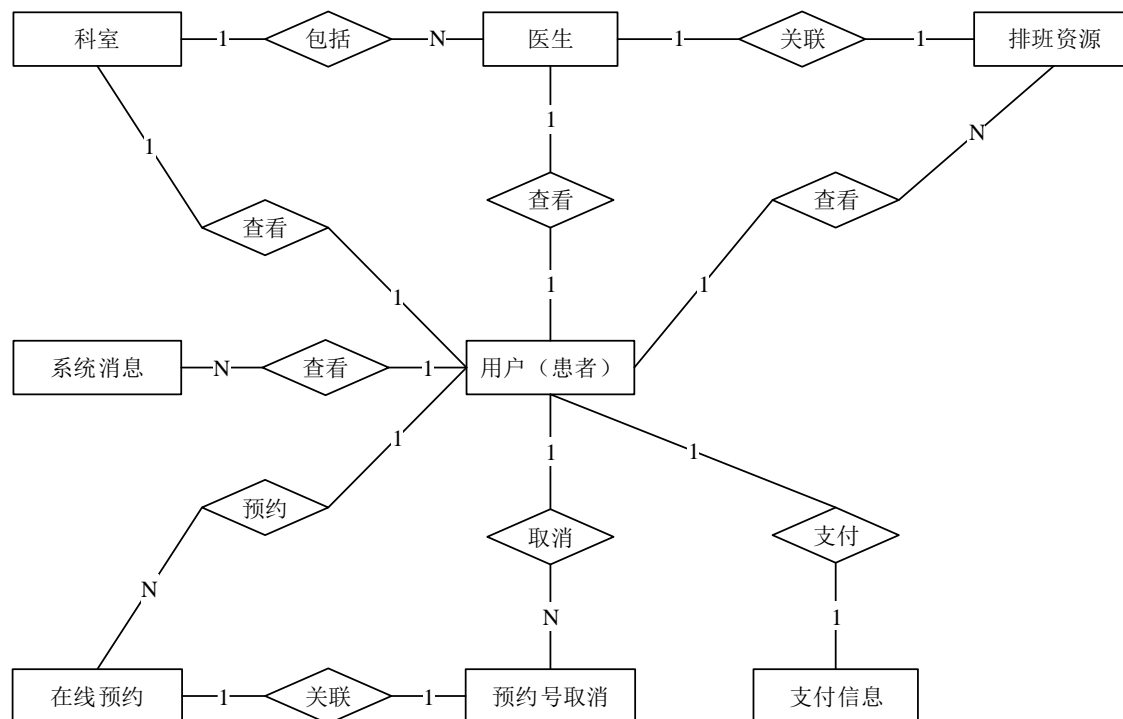


图 4-8 系统 E-R 图

表 4-2 科室信息表

字段	类型	说明	备注
organization_id	varchar(12)	科室编码	Pk, 非空
org_name	varchar(36)	科室名称	非空
org_discription	varchar(128)	科室描述	
Status	varchar(1)	状态	非空

表 4-3 医生信息表

字段	类型	说明	备注
doctor_code	varchar(12)	医生编码	Pk
organization_id	varchar(12)	科室编码	fk
doctor_name	varchar(18)	医生姓名	非空
description	varchar(128)	简介	

续表 4-3 医生信息表

字段	类型	说明	备注
Come_time	date	入院时间	非空
picture	blob	照片	
status	varchar(1)	状态	非空

表 4-4 用户信息表

字段	类型	说明	备注
user_name	varchar(18)	用户名	Pk, 非空
organization_id	varchar(12)	所属科室	Fk, 非空
telephone	varchar(11)	手机号码	非空
opera_name	varchar(18)	用户姓名	非空
id_no	varchar(18)	证件号码	非空
id_type	varchar(1)	证件类型	非空
password	varchar(255)	密码	非空
role	varchar(1)	用户角色	非空
enable_flag	varchar(1)	启用标志	非空

3) 用户信息表 T_USER

用户信息表结构如表 4-4 所示, 负责保存 APP、web 端两个子系统的用户信息, 包括用户名、类型、密码等字段。

4) 排班资源表 T_SCHEDULE

排班资源表表结构如表 4-5 所示, 负责保存医院所有预约号资源分布情况, 包括资源编号、科室编码、医生编码等字段。

表 4-5 排班资源表

字段	类型	说明	备注
Source_id	varchar(12)	排班资源 编码	Pk, 非空
schedule_date	date	排班日期	非空
shift	varchar(12)	班次	非空
begin_time	date	开始时间	非空

续表 4-5 排班资源表

字段	类型	说明	备注
end_time	date	结束时间	非空
dept_code	varchar(12)	科室编码	非空
source_type	varchar(1)	号资源类型	非空
appoint_fee	nubmer(8)	挂号费	非空
doctor_code	varchar(10)	医生编码	非空

5) 在线预约挂号表 T_APPOINTMENT

预约挂号表结构如表 4-6 所示, 负责保存 APP 端用户(患者)在线预约记录, 包括预约编码、预约时间等字段。

表 4-6 在线预约挂号表

字段	类型	说明	备注
reservation_id	varchar(12)	挂号编号	pk, 非空
Source_id	varchar(12)	预约号资源编码	非空
book_enable	varchar(1)	是否可预约	非空
advice_time	datetime	建议就诊时间	非空
reg_id	varchar(12)	病人 ID	非空
patient_name	varchar(18)	姓名	非空
book_idcard	varchar(18)	预约身份证	非空
book_date	date	预约时间	非空
Is_pay	varchar(1)	是否付费	非空

6) 系统消息表 T_MESSAGE

系统消息表结构如表 4-7 所示, 负责保存 APP 平台用户接收到的消息记录, 包括消息编号、消息名称等字段。

表 4-7 系统消息表

字段	类型	说明	备注
message_id	varchar(12)	消息 id	PK, 非空
organization_id	varchar(12)	科室编码	非空

续表 4-7 系统消息表

字段	类型	说明	备注
reg_id	varchar(12)	病人 ID	非空
mobile	varchar(11)	接收手机号	非空
content	varchar (256)	内容	非空
msg_type	varchar(1)	信息类型	1: 短信; 2: 微信; 非空
send_time	date	发送时间	非空
receipt	varchar(24)	回执	默认 0: 未收到; 1: 已收到。非空

7) 支付信息表 T_PAYMENT

系统消息表结构如表 4-8 所示, 负责保存用户 (患者) 通过 APP 支付的挂号费用, 包括支付编号、机构编码、收费类型等字段。

表 4-8 支付信息表

字段	类型	说明	备注
pm_id	varchar(12)	支付 id	PK, 非空
reg_id	varchar(12)	病人 ID	非空
dept_code	varchar(12)	科室编码	非空
doctor_code	varchar(12)	医生编码	非空
amount	nubmer	金额	非空
pay_type	varchar(12)	收费类型	非空
pay_direct	varchar(12)	支付方向	1: 支付; 2: 退款; 非空
pay_time	date	支付时间	非空
pay_state	varchar(1)	支付状态	默认 0: 未支付; 1: 已 成功; 2: 支付中; 非空

8) 预约号取消表 T_CANCEL

表 4-9 预约号取消表

字段	类型	说明	备注
reservation_id	varchar(12)	挂号编号	pk, 非空
Source_id	varchar(12)	预约号资源编 码	非空
cancel_enable	varchar(1)	预约取消	非空
reg_id	varchar(12)	病人 ID	非空
patient_name	varchar(18)	姓名	非空
book_idcard	varchar(18)	预约身份证	非空
book_date	date	预约时间	非空

4.6.2 数据并发处理

在本系统中，数据并发处理的优化处理，是一个重要的解决点。系统上数据并发主要存在于：在某时间段，系统突然接入大量用户访问数据。在某时间段，同时有多个缓则预约或者挂同一个号。针对系统突然接入大量用户访问数据，解决方法：

1) 数据库优化：支持采用多元化方式来优化数据库设计与管理工作的主体，能够提高数据执行语句的速度，建立多表连接查询模式，优化语法与表结构，建立索引来提高检索效率，支持数据分区、分表进行独立管理，提高数据逻辑处理能力。

2) 读写分离：对主从服务器，写用主数据库，读用从数据库进行多个数据库服务器配置。

3) 批量读取和延迟修改：支持用户批量操作数据库数据，提高系统并发处理能力，能够将经常使用的数据保存在缓存中，提高数据访问效率。

针对同时有多个缓则预约或者挂同一个号，采用“锁号”的功能，当其中一个用户（患者）首先拿到预约号，系统会自动分配该预约号到该用户进行“锁定”，其他用户是则被提示：该号已被另外的用户（患者）锁定，提示重新预约另外的号。直到该号被重新释放，才可以被重新接受预约。

4.7 本章小结

本章是根据医院的预约挂号系统的需求，设计 APP 管理预约和后台管理，主要研究如下：

1) 介绍了系统应用架构设计和技术架构设计。

2) 介绍了预约管理系统的软件设计方案，APP 挂号预约：用户（患者）注册、支付处理、在线预约和后台管理：预约管理、科室/医生管理、排班管理、后台用户管理、消息通知来实现挂号预约功能。

3) 介绍了数据库接口，以及数据并发处理的两种情形：大量用户访问和同一预约号被多个用户（患者）同时预约的处理方法。

通过以上研究，基本实现了挂号预约系统的需求。

第五章 挂号预约管理系统实现

本文研究的医院预约挂号系统主要是基于 Java 语言进行开发,采用 SSH 框架进行集成开发,能够满足企业级应用开发,可以为用户提供性能稳定、功能完善、数据正常的系统。由 J2EE 应用服务层、医院管理系统层、客户层等构成的。在界面设计上,针对 APP 用的是 Android。通过简单、简洁、快速、友好的人机互动界面来增强用户好感,实行人机交互,提升用户的满意度。

5.1 APP 挂号预约系统实现

5.1.1 登录注册

某医院挂号预约管理系统 APP 启动后,经过短暂图标寻转后,用户便可以马上进行注册或者登录的页面,如图 5-1 所示。用户登录后,后台系统会自动配置给用户一个字段,用户可通过手机号码进行注册,同时通过获取到随机码来验证手机用户身份信息,并录入初始化登录密码。注册用户需同意 APP 使用协议,确保用户按照遵守协议的相关规定,不违规操作。用户在登录过程使用手机号和密码即可登录进入,方便快捷,如图 5-2 所示。



图 5-1 注册登录页面

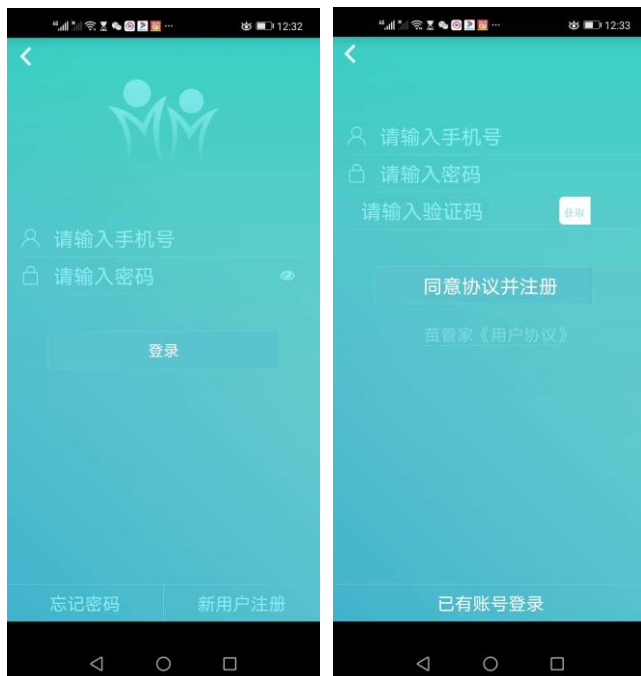


图 5-2 APP 登录与注册页面

用于检验密码和账户的代码如下：

//检验是否正确

```
public boolean insLogin(String account, String password, String
code){if(list.size()==0){return false;
}
for(int i=0;i<list.size();i++){
String str=list.get(i);
String[]a=str.split('-',2);
if(a.equals(account)&&a[1].equals(password))return true;
return false;
```

后台会对获取到的账户与密码信息展开检验，如果接收到的反馈值是 0，那么输出 false，反之则输出 true。在用户输入的账户和密码一致之后，会转入主页面，并展示小窗口“登录完成”；获取到的账户信息不正确，小窗口会显示“账户密码不匹配！”。

代码具体内容为：

```
if(MainAcitivity.main.isLogin(account, password, code)){
MainAcitivity.main.setContentView(R.layout.menu);
MainAcitivity.main.menu=new Menu();
oast.makeText(MainAcitivity.main.”登录成功！”，
Toast.LENGTH_LONG).show();
}else{
Toast.makeText(MainAcitivity.main,”账号或密码或验证码错误！”
//
Toast.LENGTH_LONG).show();passwordText.setText("");
}
}else{
Toast.makeText(MainAcitivity.main,”账号或密码或验证码错误！”，
Toast.LENGTH_LONG).show();
}
}
})
```

5.1.2 在线预约

在线预约过程，用户（患者）登录后可以采取以下方式进行查询可进行的预约：

- 1) 输入时间，查询未来 7 天的排班信息。
- 2) 输入科室、医生的部分姓名作为选取条件，如图 5-3 所示。
- 3) 用户（患者）可以在医生页面上看到医生的介绍，如图 5-4 所示。并可进行预约，如图 5-5 所示。



图 5-3 科室列表页面



图 5-4 医生页面

图 5-5 预约确认

4) 在预约信息栏中, 用户(患者)可以通过检索类型 + 相应检索条件查询获取个人信息。获取个人信息成功后, 可点击一条预约状态为“可预约”的排班明细信息, 进行门诊预约。在预约时, 可临时更改接收预约短信的电话号码, 则本次预约成功时, 预约短信将会发送至临时填写的手机号码。如需永久更改接收预约短信的号码, 请前往用户信息管理进行更改。

5) 预约成功后, 用户(患者)可以在预约信息栏处通过检索类型 + 相应检索条件查询个人已预约明细信息。已预约信息在排班明细中展示。用户(患者)可以点击一条已预约的排班明细信息, 进行修改或取消预约操作。修改或取消预约, 必须在该已预约班次时段(建议看诊时间)的 24 小时之前进行。修改或取消预约成功后, 系统同样会发送短信通知。

部分代码如下:

```
<div style="margin-top: 100px;margin-left: 300px">
  <div id="page-inner">
    <div class="container">
      <div class="h3 alert-info info">我的申请</div>
      <%--显示我的申请--%>
      <table class="table table-hover table-striped">
        <thead>
          <tr>
```

```

        <td>序号</td>
        <td>申请</td>
        <td>星期</td>
        <td>申请时间</td>
    </tr>
</thead>
<c:forEach var="item" items="${maps}" varStatus="statu">
    <tr>
        <td>${statu.index+1}</td>
        <td>${item.request}</td>
        <td>星期${item.worktime}${item.ampm}</td>
        <td>${item.applytime}</td>
        <td>${item.reason}</td>
        <td>${item.state}</td> <td class="btn btn-info
btn-link" data-aid="${item.aid}" data-toggle="modal" data-target="#myModal">
        <c:if test="${item.state=='等待处理'}">
            取消
        </c:if></td>
    </tr>
</c:forEach>
</div>
</div>
</div>

```

5.1.3 支付处理

本系统采用接入微信的支付方式。用户（患者）提交预约请求之后，系统会弹出支付界面，用户扫描二维码进行支付。成功之后，微信会回调系统的接口。系统将会自动检测到交易成功，则会更改支付状态为“完成”，同时预约完成。微信预支付订单保留 10 分钟，用户（患者）需在 10 分钟内完成支付。若逾时，预支付订单自动取消。同时，为用户（患者）占用的预约号也自动释放。在线支付如图 5-6 所示。

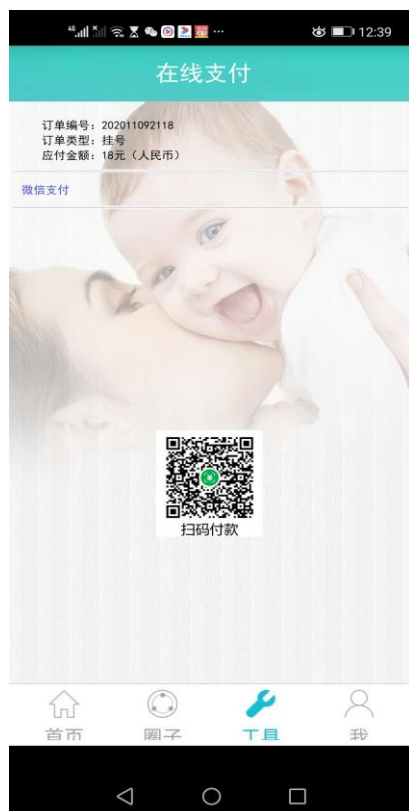


图 5-6 在线支付

代码如下:

```
public void CreateOrder() throws Exception{
    //请求 URL
    HttpPost= new
HttpPost("https://api.mch.weixin.qq.com/v3/pay/partner/transactions/jsapi");
    // 请求 body 参数
    String reqdata = "{"
        + "\"attach\": \"自定义数据说明\","
        + "\"notify_url\": \"https://www.weixin.qq.com/wxpay/pay.php\","
        + "\"settle_info\": {"
        + "\"profit_sharing\": false,"
        + "},"
        + "\"amount\": {"
        + "\"total\": 100,"
        + "\"currency\": \"CNY\""
        + "}, {"
```

```

        + "\"goods_name\": \"guahao\","
        + "\"wechatpay_goods_id\": \"1001\","
        + "\"quantity\": 1,"
        + "\"merchant_goods_id\": \"挂号费用\","
        + "\"unit_price\": $string"
        + "}"
        + "};

StringEntity entity = new StringEntity(reqdata);
entity.setContentType("application/json");
httpPost.setEntity(entity);
httpPost.setHeader("Accept", "application/json");
//完成签名并执行请求
CloseableHttpResponse response = httpClient.execute(httpPost);
try {
    int statusCode = response.getStatusLine().getStatusCode();
    if (statusCode == 200) { //处理成功
        System.out.println("success,return body = " +
EntityUtils.toString(response.getEntity()));
    } else if (statusCode == 204) { //处理成功，无返回 Body
        System.out.println("success");
    } else {
        System.out.println("failed,resp code = " + statusCode + ",return body =
" + EntityUtils.toString(response.getEntity()));
        throw new IOException("request failed");
    }
} finally {
    response.close();
}
}

```

5.2 WEB 后台管理实现

5.2.1 后台用户管理

1) 用户成功登录后可进入用户管理界面，如图 5-7 所示，同时能针对相对应的管理权限进行使用，本系统预设了系统管理员、管理员、操作员、医生、用户

(患者) 5 类别管理角色。

2) 系统管理员为最高级别角色, 其可以设置、修改、维护系统设置; 设置、修改、维护电信运营商端口地址; 设置、修改、维护系统账号与相关密码设置标准; 设置、修改、维护其余类别的管理用户使用权限; 设置、修改、维护消息模板等。

3) 管理员为科室类型管理角色, 可以按照实际需求设置院长室、内科、外科、骨科、中医科、财务科、住院部、门诊部、药剂科、统计科等不同职能的管理员, 其可以设置、修改本身用户密码; 设置、修改、维护操作员、医生这 2 个类别的管理用户使用权限; 设置、修改、维护上述 2 个类别管理用户的账号与相关密码; 设置、修改、维护科室消息模板等。



图 5-7 后台登录页面

4) 操作员为业务操作类型管理角色, 可以根据不同科室的不同业务范畴进行相应设置, 其可以设置、修改、维护医生等用户的基本信息; 其中的预约排班、挂号操作员可以制定、修改、维护各相应科室的医生出诊排班表、病人挂号顺序表等。

5) 医生为个人用户类型使用角色, 其可以设置、修改本身用户密码; 设置、修改、维护本身出诊安排; 查询、修改预约挂号本身的病人挂号顺序表。

5.2.2 预约管理

1) 可以通过输入需预约医生的姓名、需预约科室的名称中的任意条件, 对相关科室/出诊医生的排班信息进行查询。查询的时间范围为当天至未来一星期。

2) 可以对查询结果根据医生出诊日期进行进一步筛选。在出诊医生列表中, 列出符合筛选条件的出诊医生的排班信息, 展示的信息包括医生姓名、出诊日期(星期几)、上午班次还是下午班次、医生的职称、预约挂号费用、该医生该班次剩余可预约号数等, 如图 5-8 所示。

3) 在排班明细中, 列出符合筛选条件的用户(患者)预约明细信息。展示的信息包括班次、序号、建议看诊时间、预约状态、用户(患者)姓名、用户(患者)电话、预约类型、登记员、登记时间等。

5.2.3 科室/医生管理

科室管理是对所有科室信息进行维护和调配, 支持后台批量导入数据的功能, 也支持逐条新增。新增科室描述如图 5-9 所示。系统会对新增记录会在一定的时间间隙内自动刷新或手工刷新。

科室医生																																																																															
预约医生 <input type="text"/>	预约科室 <input type="text"/>																																																																														
预约信息																																																																															
检索类型 <input type="text" value="手机号"/>	检索条件 <input type="text"/>																																																																														
患者姓名 <input type="text"/>	电话号码 <input type="text"/>																																																																														
身份证号 <input type="text"/>																																																																															
预约备注 <input type="text"/>																																																																															
出诊医生	排班明细																																																																														
日期 <input type="text"/>	时段 <input type="text"/> 状态 <input type="text" value="可预约"/>																																																																														
<table border="1"><thead><tr><th>医生</th><th>日期(星期)</th><th>出诊身份</th><th>剩余号数</th></tr></thead><tbody><tr><td>王医生</td><td>2020-10-30 (二) 上午</td><td>教授出诊9元</td><td>28</td></tr><tr><td>王医生</td><td>2020-11-02 (五) 上午</td><td>教授出诊9元</td><td>30</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table>	医生	日期(星期)	出诊身份	剩余号数	王医生	2020-10-30 (二) 上午	教授出诊9元	28	王医生	2020-11-02 (五) 上午	教授出诊9元	30													<table border="1"><thead><tr><th>班次</th><th>序号</th><th>建议时间</th><th>状态</th><th>姓名</th><th>电话</th><th>预约类型</th><th>登记员</th><th>登记时间</th></tr></thead><tbody><tr><td>上午</td><td>1</td><td>2020-10-30 08:00:00</td><td>可预约</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>2</td><td>2020-10-30 08:00:00</td><td>已预约</td><td>XXX</td><td>136XXXX</td><td>现场</td><td>AAA</td><td>2020-10-29 08:00:00</td></tr><tr><td></td><td>3</td><td>2020-10-30 08:00:00</td><td>可预约</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>4</td><td>2020-10-30 08:00:00</td><td>可预约</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>5</td><td>2020-10-30 08:00:00</td><td>可预约</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>	班次	序号	建议时间	状态	姓名	电话	预约类型	登记员	登记时间	上午	1	2020-10-30 08:00:00	可预约							2	2020-10-30 08:00:00	已预约	XXX	136XXXX	现场	AAA	2020-10-29 08:00:00		3	2020-10-30 08:00:00	可预约							4	2020-10-30 08:00:00	可预约							5	2020-10-30 08:00:00	可预约					
医生	日期(星期)	出诊身份	剩余号数																																																																												
王医生	2020-10-30 (二) 上午	教授出诊9元	28																																																																												
王医生	2020-11-02 (五) 上午	教授出诊9元	30																																																																												
班次	序号	建议时间	状态	姓名	电话	预约类型	登记员	登记时间																																																																							
上午	1	2020-10-30 08:00:00	可预约																																																																												
	2	2020-10-30 08:00:00	已预约	XXX	136XXXX	现场	AAA	2020-10-29 08:00:00																																																																							
	3	2020-10-30 08:00:00	可预约																																																																												
	4	2020-10-30 08:00:00	可预约																																																																												
	5	2020-10-30 08:00:00	可预约																																																																												

图 5-8 挂号预约查询记录页面

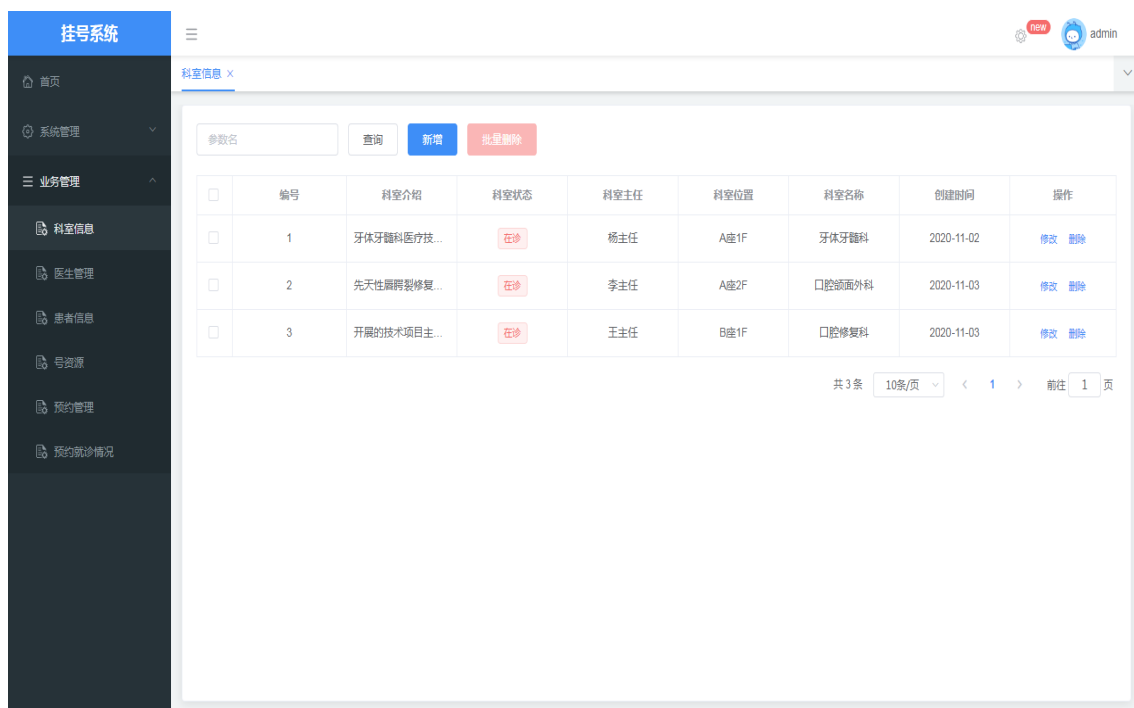


图 5-9 新增科室查询列表

科室查询列表部分代码如下：

会执行 `listDepartment ()` 方法：

```
@RequestMapping(value = "/listDepartment", method = RequestMethod.GET)
```

```
public String listDepartment (HttpServletRequest response,
```

```
HttpServletRequest request) {
```

```
//传入科室名称参数
```

```
String departmentName = request.getParameter("departmentName");
```

```
//创建部门对象
```

```
Department dm= new Department ();
```

```
//将部门名称设置为查询条件 departmentName
```

```
dm.setName(departmentName);
```

```
//执行查询语句
```

```
List< Department > departments
```

```
= departmentsService.findList(department);
```

```
request.setAttribute("departments ", departments);
```

```
return " departmentsList";
```

```
}
```

同时后台也支持对科室名称、描述等信息进行修改。后台开发批量模板，可

下载 EXCEL 格式模板，按要求填写数据即可导入。需要注意的是：每个科室绑定了相关医生信息是唯一的，删除完毕所有医生后才可删除科室信息。

医生管理是对科室内医生信息进行添加、修改或删除。当需新增医生，需要更新医生数据库，可以单条录入。当然，如果医院录入了一批新医生的信息，后台开发批量固定模板，如 EXCEL 格式模板导入科室数据，以 EXCEL 表格批量导入医生。添加医生如图 5-10 所示。

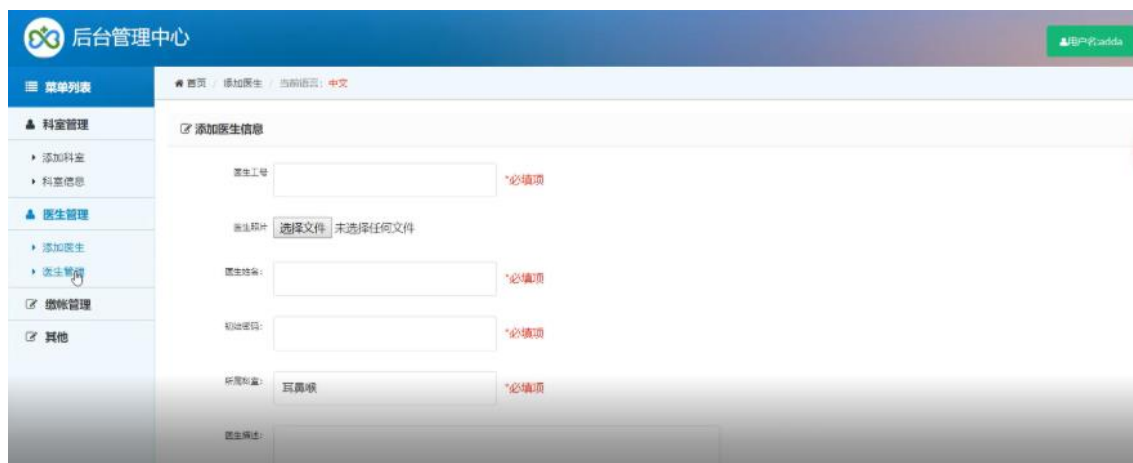


图 5-10 医生管理

加入医生资料需求，运行 Doctor ManageAction 中的 createDoctPersonInfo() 算法，传递至 CommonStatusResult 目标，代码为：

```
@RequestMapping(value="/createDoctPersonInfo",
method=RequestMethod.POST)
@ResponseBody
public CommonStatusResult createDoctPersonInfo(@RequestBody(required=true)
DTODOctPersonInfo dto) {
    DoctPersonInfo    PersonInfo=    DTODOctPersonInfo.    toEntity(dto);    if
(DoctPersonInfo=null){throw
FoodExecption.returnException(ErrorMessageConstant,CODE_PARAM_ERROR);
}
channelService, createDoctPersonInfo(DoctPersonInfo);
return success(CODE_SUCCESS, Doctor:eInfo.get Id 0);
```


5.2.4 排班管理

号资源查询界面如图 5-11 所示，可以查看资源来源、医生工号等信息。

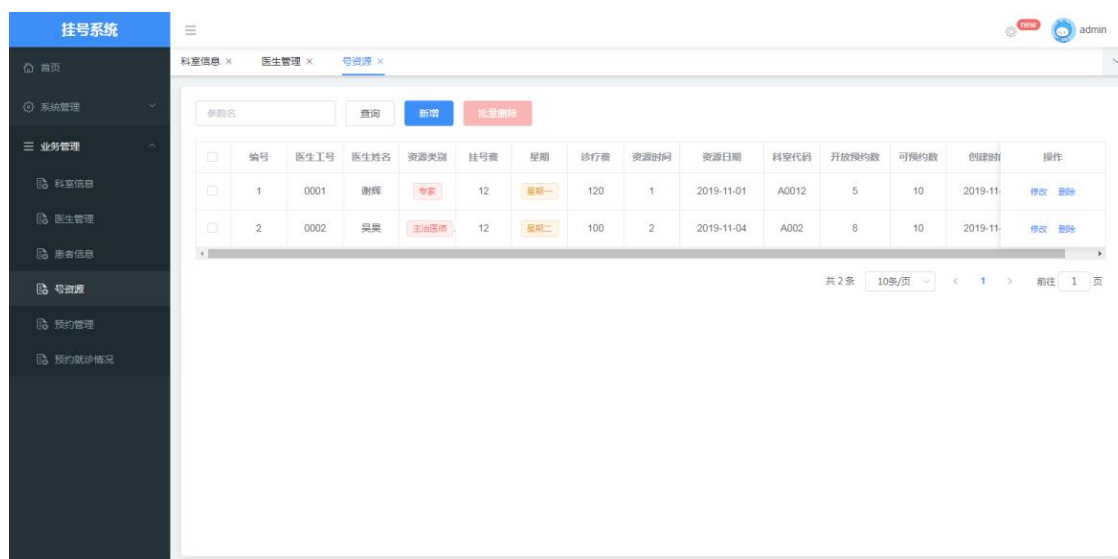


图 5-11 号资源查询页面

可以对医生进行排班，生成各种号资源。用户（患者）可以输入时间，查询未来 7 天的排班信息。用户（患者）可以输入医院、班次、科室、医生的部分姓名作为可选查询条件。查询结果按日期、班次、科室、医生姓名、时段进行升序排序。用户（患者）可以在查询结果中选择一条排班信息，查看其相关的详细预约号信息。用户（患者）可以选择一条状态为“可预约”的预约号，进行预约。此时，该预约号状态变为“占用”，用户（患者）需要在 10 分钟内完成微信。如支付成功完成，则该预约号状态变为“已预约”。如逾期未完成支付，或用户（患者）取消预约，则该预约号释放，状态变为“可预约”。系统会自动根据科室、医生的上班時間，进行自动分配，并输出排班表，并通知到相关的医生，如果医生对此有异议，则需向上反馈，重新更改或调整时间。

5.2.5 消息通知

消息发布之前，需要设定具体的模板，选择短信 URL。

1) 可以预先通过设置消息模板，设置生成的消息内容。在消息模板中，可以使用以下的变量进行内容自动替换：日期[\$date]、医院[\$org]、科室[\$dept]、医生[\$doctor]、诊室[\$room]、班次[\$shift]、建议就诊时间[\$time]、用户（患者）姓名[\$name]、就诊号[\$seq]、挂号费[\$fee]。

2) 在消息配置界面，需要填写合作的短信营运商的发短信端口地址及用户名

密码

3) 当预约成功/取消预约时, 如系统中保存了用户(患者)的手机号码/微信openid, 则会同时发送短信和微信消息通知。

5.3 本章小结

本章在挂号预约管理系统的实现, 介绍医院挂号预约管理系统的手机终端应用 APP 设计和后台管理, 主要研究如下:

- 1) 场景化的操作界面和操作步骤设计, 包括用户的注册、登录、预约、支付。
- 2) 后台管理, 包括科室/医生的管理、预约管理、排班管理、消息通知。

通过上述设计, 系统简单、简洁、快速、友好的人机互动界面能增强用户好感, 实行人机交互, 提升用户的满意度。

第六章 系统测试

6.1 测试总体设计

1) 测试设计

系统测试是测试系统设计的功能能否按设定的逻辑和程序正常运行，数据传输、用户端和后台数据是否一致，展示的数据内容显示是否正常、正确，系统性能是否平稳。

本系统测试方法是黑盒测试和白盒测试。黑盒测试是测试人员按步骤测试功能按钮、链接、菜单、文本域等各种控件功能是否正常，查找问题，同时也可以设定某些功能范围的误操作模式，来模拟系统会不会对误操作进行正确的判断^[43]。白盒测试是通过设计测试用例，核实系统功能，逐一验证，确保系统符合要求。同时也对代码跟踪每个参数值、流转情况以及数据库存储情况进行确认，同时进行集成测试，将不同模块整合在一起进行联合测试，掌握运行情况^[44]。

同时针对来模拟大量用户同时访问并操作系统，看过程中是否有变慢或用户数量过多无法访问的情况出现。通过系统自动记录测试数据，本此测试采用的LoadRunner 12.6 测试工具来建立一定数量的用户数据^[45]。

本文中测试主要是针对用户管理、科室/医生管理、预约管理、支付处理进行白盒测试，同时也着重对数据并发处理中的同一预约号多个用户（患者）预约的“锁定”功能进行测试。

2) 测试环境配置

(1) 服务器端

a、Web 服务器

软件配置：Windows10、JDK14、tomcat9.0、MyEclipse2019。

硬件配置：CPUP4 2.6G,内存 8G,硬盘 2TB。

b、数据库服务器

软件配置：Windows Server 2019、MySQL。

硬件配置：CPUP4 2.4G 内存 8G,硬盘 2TB。

(2) 客户端

软件配置：Windows10、IE10.0。

硬件配置：CPU 英特尔酷睿 i5,内存 8G,硬盘 1TB。

6.2 测试用例

6.2.1 用户管理

用户登录模块可以让用户（患者）和医护人员两种角色进行操作，用例目的是对用户密码、手机号码校验检查。用户登录测试用例如表 6-1 所示。

表 6-1 用户登录测试用例

测试用例 ID	主要规则	预置条件	测试步骤	期望结果
UT_01_001	输入错误的用户名或密码	信息数据库中不匹配	1.在登录界面输入错误的用户名或密码； 2.点击登录按钮	1.无法登录系统； 2.界面弹出提示“用户名或密码错误”。

6.2.2 科室/医生管理

科室/医生管理功能模块可以让管理员角色进行操作，用例目的是对科室和同一医院医生姓名唯一性检查。用户（患者）建档用例如表 6-2 所示。

表 6-2 科室/医生管理用例

测试用例 ID	主要规则	测试步骤	期望结果
新增科室 UT_04_001	同一医院，不能有相同的科室名称	1.成功登录本系统并进入科室/医生管理界面； 2.输入需要新增的科室数据；3.点击添加按钮。	1.检查输入的科室名称是否已存在，如存在则提示科室名称已存在，添加失败； 2.如输入的科室名称不存在数据库中同一医院中，则添加科室数据到数据库中，刷新显示该医院的科室列表。
新增医生 UT_04_002	同一医院，如出现相同名字的医生，则需要名字上加区分	1.成功登录本系统并进入科室/医生管理界面； 2.在文本框中输入需要新增的医生数据；3.点击添加医生按钮。	1.检查输入的医生姓名是否已存在数据库中同一医院中，如存在，则提示医生姓名已存在，添加失败；2.如输入的医生姓名不存在数据库中，则添加医生数据到数据库中，并刷新显示该医院的医生列表。

6.2.3 排班管理

排班管理功能模块可以让用户（患者）/医生、管理员角色进行操作。排班管理功能测试用例如表 6-3 所示。

表 6-3 排班管理功能测试用例

测试用例 ID	主要规则	用例目的	测试步骤	期望结果
查询排班 UT_05_001	输入科室名称进行准确查询或医生姓名进行模糊查询	自动识别输入的是科室名称还是医生姓名，进行相应的查询	1.医生成功登录本系统并进入排班管理界面； 2.选择具体科室作为查询条件； 3.点击查询按钮。	1.显示符合查询条件的排班数据；2.选中一条排班信息，在右方表格显示这条排班信息相关的所有预约号信息。
修改排班 UT_05_004	排班医生/时间段唯一性检查	同一时间段医生排班唯一性检查	1. 管理员成功登录本系统并进入排班管理界面；2.选择日期并查询需要修改的排班信息；3.选中需要修改的排班信息，点击修改按钮；4.修改排班信息；5.点击保存修改按钮。	1.检查输入的修改后的排班信息，同一时间段，一个医生只能有一条排班信息；2.在修改排班信息的同时，修改相关的预约号的待预约信息。
发布排班 UT_05_006	排班信息发布后才能生效	排班确认操作检查	1. 管理员成功登录系统并进入排班管理界面；2.选择开始和结束日期；3.确认需发布的排班信息填写正确但未发布；4.点击发布按钮。	1.排班信息发布后，排班信息才生效，用户（患者）只能看到发布后的排班信息。

6.2.4 预约管理

预约管理功能模块可以让用户（患者）进行操作。预约管理功能测试用例如表 6-4 所示。

表 6-4 预约管理功能测试用例

测试用例 ID	主要规则	用例目的	测试步骤	期望结果
新增预约 UT_06_001	同一时间段用户（患者）只能预约一个号；预约的时间段尚有未被占用的预约号	同一时间段用户（患者）预约号唯一性检查	1.成功登录本系统并进入预约界面；2.选择需要预约的日期、班次；3.可选择科室或填写医生姓名作为查询条件；4.查询获得相关的排班信息；5.选中需要进行预约的排班信息；6.在右方表格选择未被占用且需要预约的时间段内的预约号；7.点击预约按钮；8.进行支付。	1.在预约的该医生的该班次该时间段内，有一个预约号被该用户（患者）占用；2.该预约号在释放之前，不能分配给别的用户（患者）；3.该用户（患者）在该预约号释放之前，不能再预约同一时间段内别的预约号。

预约锁号及未支付解锁测试用例 ID：UT_06_001

以发热门诊（0098000） 2020-03-02 日 李雪梅医生的号为例，如下图 6-1 原始状态如下：

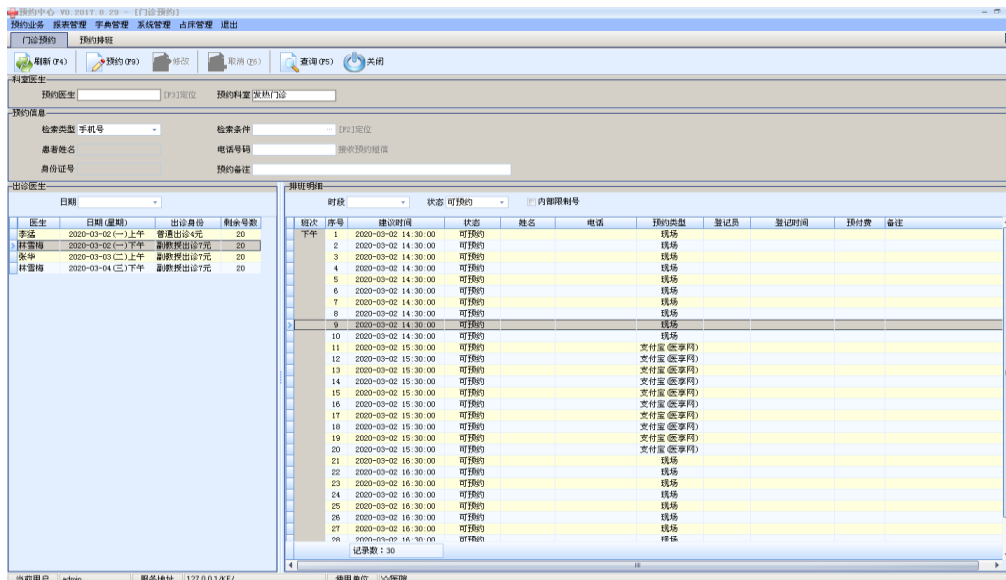


图 6-1 预约管理用例

测试状态下，两位用户同时锁定 8 号，其中一台先将 8 号锁定，另一预约用户提交时提示“选定的预约号已被其他用户锁定”。

锁号成功,如图 6-2 所示,查询数据可看出被锁定的 8 号已为“不可预约”状态。
解锁未支付的锁号的定时任务,每 1 分钟运行清理一次,语句如下:

```
update gh_base_child a
```

```
set a.gh_operate = 'a.gh_date = null,a.serial_no = null,a.book_enable = 1, charge_flag = 0
```

```
where a.gh_date < sysdate - 15/(1*24*60) -- 锁号15分钟未支付的
```

```
and NVL(a.charge_flag,0) = 0 -- 未支付
and nvl(a.serial_no,'') <> ''
```

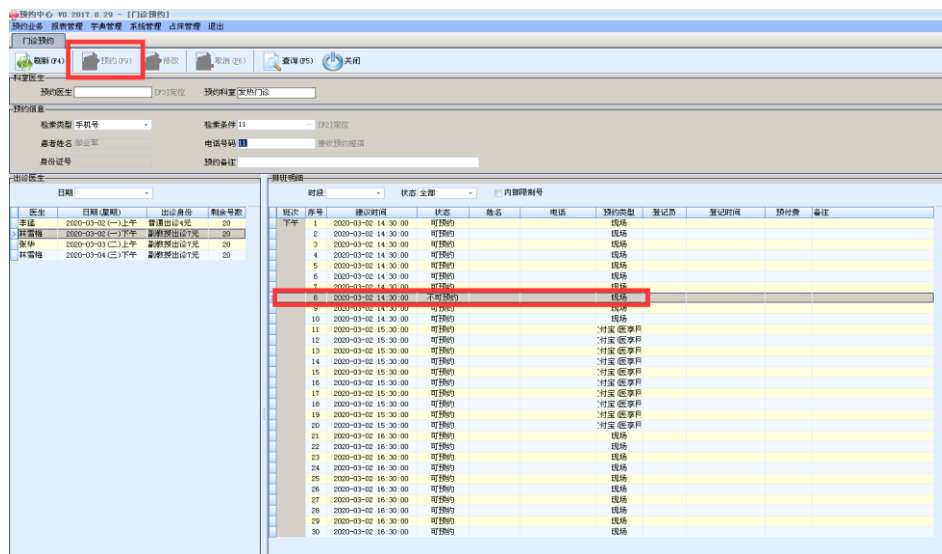


图 6-2 预约管理：不可预约

解锁后,预约查询界面可看到不可预约的 8 号已重新变为“可预约”,如图 6-3 所示。测试结果:预约锁号成功,同时锁号但未在规定时间内支付的号解锁成功。

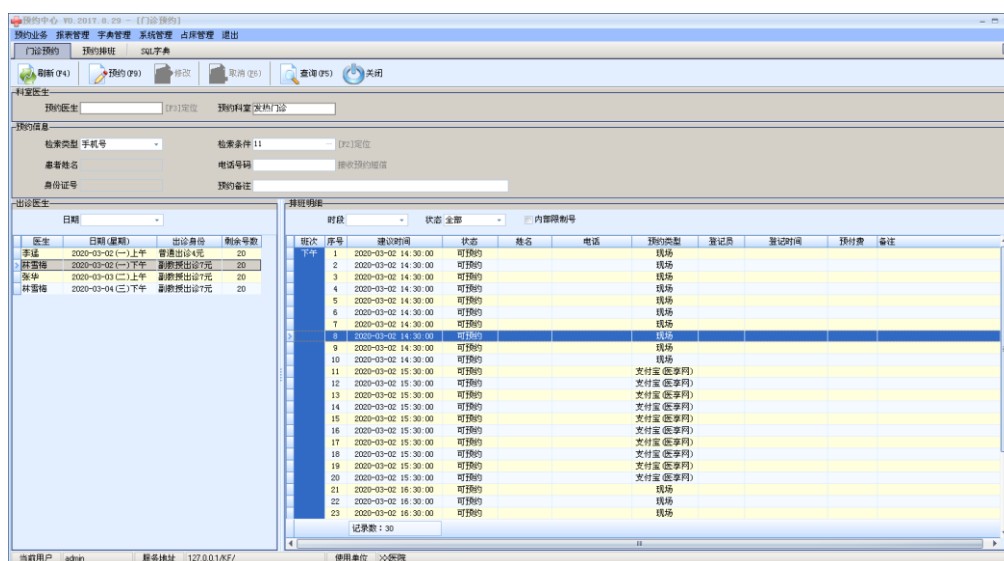


图 6-3 预约管理：预约解锁

6.2.5 支付处理

支付处理功能模块可以让用户（患者）进行支付或退款操作。支付处理功能测试用例如表 6-5 所示。

表 6-5 支付处理功能测试用例

测试用例 ID	主要规则	用例目的	预置条件	测试步骤	期望结果
用户支付 UT_07_001	用户（患者）对预约进行支付，系统根据微信平台回调信息更改支付状态和预约状态	用户预约和微信流程检查	用户（患者）提交预约，已占用预约号但尚进行支付	1.成功提交预约，弹出支付窗口； 2.扫描二维码并支付。	1.微信成功，支付金额为预约金额；2.成功预约，可在相关预约号查询得到相关的预约记录。
用户支付 UT_07_002	用户（患者）取消预约，系统自动进行微信退款，并根据微信平台的回调信息更改预约状态	用户取消预约和微信退款流程检查	用户（患者）进行取消预约操作	1.提交预约申请； 2.等待系统向微信平台提交退款申请。	1.预约成功取消，占用的预约号释放，状态为“可预约”；2.次日收到微信退款。

6.3 本章小结

本章是对挂号预约系统功能能否按设定的逻辑和程序正常运行，数据传输、用户端和后台数据是否一致，展示的数据内容显示是否正常、正确，系统性能是否平稳进行测试，以此来验证系统的安全性和功能满足性，主要研究如下：

1) 介绍挂号预约管理系统模块测试方案的设计和采用的测试方法：黑盒法是通过操作系统进行测试，白盒法是通过设定测试用例进行判断。

2) 介绍了测试的设备配置。

3) 介绍了用户管理、科室/医生管理、排版管理、预约管理、支付处理的部分测试用例，进行了测试结果的分析，达到预期的测试目的。

通过上述测试的验证，证明了系统的功能设计的合理性和安全性。

第七章 总结和展望

7.1 全文总结

本医院挂号预约管理系统是基于结合某医院挂号的现状分析，在医院以电话、现场挂号方式的现状下，基于移动互联的技术基础上，开发了挂号预约管理系统前端 APP 和后台管理。硬件上，以医院现有的内网系统的硬件为基础，在系统上开发外置的预约挂号的软件接口，增加必要的硬件终端，保证用户（患者）可以通过手机、网络、电话、现场终端等方式快速与挂号系统相连，实现快速就诊。软件上，开发了手机终端 APP 应用软件实现了用户（患者）可以通过手机 APP 进行预约挂号，自助缴费的功能。后台管理上，开发了系统管理应用，实现用户管理、预约管理、消息通知、排班管理、科室/医生管理、支付处理等功能，建立用户信息数据库，以便实现用户（患者）可快速预约需求的功能。

但是由于该医院管理由于涉及到多个业务的实际场景，按场景的实践情况开发，整个系统庞大，大系统又根据实际情况拆分成若干个子系统进行开发，每个子系统又根据实际情况进行细化和拆分，拆分成若干模块，采用了强大的数据库编程工具，达到功能高效、操作人性化，本挂号预约管理系统主要有以下特点：

1) 基于 B/S 多层架构比较完整，同时基于互联网技术，可以满足基于移动互联技术的远程挂号和信息查询的要求，同时系统兼有对大量数据进行后台快速高效处理功能，在数据的时效性和准确性方面提供有效的技术流，界面上提供了一个高效、直观、便捷、操作人性化的挂号平台，方便用户（患者）和医生。

2) 在数据信息收集方面基本放弃了现场挂号由医护人员负责手动输入用户（患者）信息的方式，此方式工作量大，容易出错，改由用户（患者）根据网上预约挂号系统，注册时输入信息，在用户（患者）在医院看病时候，通过条形码的方式，将其信息与系统关联，提供了工作效率，降低了出错率。

3) 接口扩展性好，标准、规范、接口技术统一化，同时采用聚合模式，根据业务场景拆分不同功能模块，并模块化设计，方便随时扩展，系统的组成部分相对简单，易于调整，可为以后升级留出更多的空间。

7.2 展望

医院挂号预约管理系统的开发是一项复杂的工作，规模庞大，需要许多人员的参与，开发过程中需要涉及到的问题很多，由于个人的时间和精力限制，系统

开发研究还存在许多不足之处，本系统的主要开发工作是围绕需求和系统分析的功能开发，因此，该系统还有很大的空间可以改进。系统性能优化和程序编写有待完善，需要提高系统功能的个性化和智能性，页面布局只是实现了正常的模块显示，界面不够美观。所提供的功能不完善，还需要扩展其他模块，在后续的设计开发中，将根据用户要求进一步开发。

致 谢

在这里对读研和论文写作期间给予帮助的老师 and 同事表示感谢。在我读研期间，各位老师为了不影响我们的工作，在放假期间给我们授课，老师们制定了丰富的课件资源方便我们学习专业知识，对我提出的问题耐心给予解答。我的导师王伟教授给予我很多帮助，对我遇到的问题进行细心指导，让我能够顺利完成软件工程专业课程的学习。在论文撰写期间给我提供指导，引导我结合工作制定课题研究方向。对我所研究的课题进行了客观的评估，对我论文能够顺利完成给予了大量支持。公司领导积极鼓励我们继续学习提高专业知识，对我读研提供了大量支持，使我们能够顺利完成研究生学业。另外，在我课题研究期间，同事在百忙中抽出时间，给予我非常好的建议和帮助，他们分别是霍军亚等，谢谢你们对我的支持和帮助！最后，感谢专家评委，认真负责的对我的结题论文进行评审！

参考文献

- [1] 张奕珠. 医院信息管理系统存在的问题及设想[J]. 辽宁师专学报, 2021, 3(1): 107-108
- [2] 黄晓红, 王思贤, 何楚. 基于 Web 医疗信息管理系统的设计与实现[J]. 计算机应用研究, 2020, 29(10): 99-102
- [3] 尚武, 杨涌. 我国医疗信息系统及发展趋势[J]. 中华医学图书情报杂志. 2017, 11(4):14-17
- [4] 陈康, 彭浩文, 余俊蓉. 医疗门诊系统的设计与应用[J]. 计算技术与自动化, 2018, 17(4): 35-36
- [5] 肖春玲, 陈彦方. 医疗信息系统与电子病历[J]. 国际精神病学杂志, 2016, 33 (1) : 12-15
- [6] 陈飞霞. 现阶段我院的挂号系统模式及未来发展形势[J]. 中华医院管理杂志, 2005(02): 119
- [7] 杨立杰. 基于 B/S 模式的医院信息管理系统的设计及实现[D]. 电子科技大学. 2019
- [8] D. Sobhy, M. AbouElans. Cloud:Health care cloud computing System[C]/Internet Technology And Secured Transactions,2012 International Conference for.IEEE,2012: 161-166
- [9] Betty A Levine Information System and Integration[C]. Proceeding of IMC
- [10] Walter Savitch. Java an introduction to computer science&programming[M]. 北京: 清华大学出版社,2015,1.
- [11] Bruce Eckel. Thinking in Java[J]. Pearson Education,2015(12): 101-105
- [12] Luo yan. Design and Implementation of E-Learning System Based on j2ee[J]. Journal of Electronic Science and Technology of China, 2015(09): 79-82
- [13] A. G. Philippa. Design and implementation of hospital registration system[J]. Journal of Medical Informatics. 2019,40(04): 101-105
- [14] M. Sergio. Design and implementation of hospital mobile appointment registration platform[J]. Electronic Technology and Software Engineering. 2019,(06): 77-79
- [15] G. D. Smith. Research on the design and application of hospital appointment registration system[J]. Medical Information. 2018,(12): 123-127
- [16] J. Badreddine. About the design and implementation of the appointment and registration system of the hospital outpatient department[J]. Science and Technology Wind. 2017, (24): 113-118
- [17] K. Hichem. Design of hospital appointment registration system[J]. Electronic Technology and Software Engineering. 2017, (13): 223-227
- [18] S. Williams. Design and research of hospital appointment registration system[J]. Science and Technology Wind. 2017, (08): 98-101
- [19] 王波. 基于 B/S 模式医院网上预约挂号系统实现[J]. 医学信息杂志, 2011,(09): 23-25

- [20] 孟涛, 赵健, 孙萍. 预约挂号在医院门诊应用的探讨[J]. 中国医疗前沿. 2020(08): 45-49
- [21] 马静璇. 预约挂号在医院门诊应用的探讨[J]. 企业科技与发展. 2020,(11): 102-106
- [22] 李雅琼; 王丹凤; 夏令强. 门诊统一号源池预约系统设计[J]. 解放军医院管理杂志. 2020, 27(10): 77-82
- [23] 梁景云. 基于门诊全预约挂号流程的优化与排队挂号系统的研究[J]. 医学信息与健康. 2020, 18(19): 220-225
- [24] 杨靖祎. 浅析微信挂号系统的设计与应用[J]. 南方农机 2020,51(09): 157-159
- [25] 巩蕾. 基于智能移动端的医院预约挂号系统设计与研究[J]. 电脑编程技巧与维护 2020,(02): 109-113
- [26] 冯清; 于琦; 吕艳华. 口腔门诊挂号系统 APP 研究与设计[J]. 医学信息学杂志 2016, 37(09): 99
- [27] 张鹏. 集约式预约挂号系统的设计与实施[J]. 中国医学教育技术. 2018(03): 76-79
- [28] 孙亚琦. 医院挂号 APP 的设计与实现[D]. 南京: 南京理工大学[D]. 2018
- [29] 姜熳; 严正仲; 陈功. 医院预约挂号平台的集成设计与实现[J]. 中国卫生信息管理杂志 2018,15(06): 77-82
- [30] 王珂. 基于 Android 系统的医院移动预约挂号平台的设计与实现[D]. 郑州: 郑州大学. 2019
- [31] 张磊. 医院预约挂号系统设计与应用探究[J]. 中国信息化. 2018(12): 110
- [32] 任晓尘. 基于 J2EE 的移动预约挂号系统的设计与实现[D]. 苏州: 苏州大学, 2016
- [33] 陈卫民. 基于 .NET 平台的安全网上支付系统设计与实现[J]. 怀化学院学报, 2017, (05): 45-47
- [34] 刘洪涛. 网络数据库开发实战[M]. 中国铁道出版社. 2013
- [35] 刘晓华, 张健, 周慧贞. JSP 应用开发详解[M]. 北京: 电子工业出版社, 2017, 1
- [36] 汪孝宣, 刘中兵, 徐佳晶. JSP 数据库开发实例精粹[M]. 北京: 电子工业出版社, 2015, 1
- [37] 陈佳. 信息系统开发教程[M]. 清华大学出版社, 2020
- [38] 涵章. 现代医学管理全书[M]. 浙江杭州出版社, 2015
- [39] 龙劲松. Delphi6 开发人员指南. 机械工业出版社, 2015
- [40] 萨师煊, 王珊. 数据库系统概论[M]. 北京: 高等教育出版社, 2020, 2
- [41] 刘建宏. MySQL 数据库优化与集群[J]. 数字通信世界. 2017(07): 112
- [42] 邓林. MySQL 存储过程调试技巧[J]. 四川职业技术学院学报. 2017(04): 89-102
- [43] 闫术卓, 杨强. Struts 2 技术详解[M]. 北京: 电子工业出版社, 2018.6
- [44] 邵刚. 基于 Spring 框架的 MVC 控制器的优化与改进[D]. 济南: 山东大学, 2021
- [45] 李刚. 轻量级 J2EE 企业应用实战[M]. 北京: 电子工业出版社, 2017