

分类号 _____
学校代码 10487

学号 M201976354
密级 _____

华中科技大学
硕士学位论文
(学术型 ☐ 专业型 ☒)

网上预约挂号系统的设计与实现

学位申请人：游国强

学 科 专 业：软件工程

指 导 教 师：邱德红 教授

答 辩 日 期：2022 年 05 月 25 日

**A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Professional Master Degree**

**Design and Implementation of an Online Hospital
Appointment Registration System**

Candidate : YOU Guoqiang

Major : Software Engineering

Supervisor : Prof. QIU Dehong

Huazhong University of Science and Technology

Wuhan 430074, P. R. China

May, 2022

摘要

由于我国人口基数庞大，地区之间经济发展水平差异明显，人均医疗资源十分有限且分布不均衡，一方面患者面临“就医难、挂号难、预约难”问题，另一方面医院存在“病源不足，资源闲置”的问题。如何在医患之间建立起有效的信息交流平台，解决患者和医院面临的问题，不仅会产生经济效益，而且会带来积极的社会效应。因此，利用软件工程技术，开发一个功能完备、服务患者和医院的网上预约挂号系统具有重要意义。

网上预约挂号系统分为两个子系统，其一是供系统管理员使用的后台管理子系统，包括了统计模块、用户模块以及订单模块等多个功能模块；另一是供患者使用的前台用户子系统，该部分包括用户登录和注册、用户认证、添加患者、医院选择、科室选择、排班选择、订单支付、取消预约等多个模块。本系统后端使用的是 Spring Boot 与 Spring Cloud 相结合的方式搭建，前端采用 Vue 框架搭建，采用 MySQL 数据库保存医院数据、订单数据、支付数据、退款数据、用户数据、患者数据、数据字典数据等，MongoDB 数据库保存医院、科室和排班数据，Redis 数据库用于保存短信验证码、微信登录等各种支付形式的二维码，使用 MP 持久层框架来操作 MySQL 中的数据模型，Nacos 作为注册中心，在阿里云 OSS 建立文件系统用于保存患者的证件照片等信息，阿里云短信服务实现手机登录功能。

对网上预约挂号系统进行的功能和性能测试表明，在并发量为 3000 的情况下，系统的各项功能都能够正常运行，且各种请求接口的最大响应时间都在 1 秒钟左右。网上预约挂号系统实现了预期的设计目标，在实际应用中对于解决医患双方面临的问题产生了积极的作用。

关键词： 预约挂号；Spring Boot 框架；Vue 框架；Redis 数据库

Abstract

Due to the huge population base, the differences in economic development levels are obvious, and the per capita medical resources are very limited and unevenly distributed. On the one hand, patients face the problem of "difficulty in seeking medical treatment, registration, and appointment"; on the other hand, the hospital has the problem of "insufficient disease sources and idle resources". How to establish an effective information exchange bridge and solve the problems will not only produce economic benefits, but also bring positive social effects. Therefore, it is of great significance to use software engineering technology to develop an online appointment registration system.

The online appointment registration system consists of two parts, the first is the background management subsystem for system administrators, which includes multiple functional modules such as statistics module, user module and order module; the second is the front-end user subsystem, this part includes multiple modules such as user login and registration, user authentication, adding patient, hospital selection, department selection, shift selection, order payment, and cancellation of appointment. The back end of the system is built with Spring Boot and Spring Cloud, the front end is built with Vue framework, and MySQL database is used to save hospital data, order data, payment data, refund data, user data, patient data, data dictionary data, etc., and MongoDB saves Hospital, department and shift data, Redis database saves three forms of QR codes, the MyBatis-Plus framework is used to operate the MySQL database, Nacos is the registration center, and Alibaba Cloud OSS is distributed. The Alibaba Cloud SMS service implements the mobile phone login function.

The function and performance test of the online registration system shows that all functions of the system can run normally when the concurrency is 3000, and the maximum response time of various request interfaces is about 1 second. The online appointment and registration system has achieved the expected design goals, and has played a positive role in solving the problems faced by both doctors and patients in practical applications.

Key words: Appointment registration, Spring Boot framework, Vue framework, Redis database

目 录

1 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.3 本文研究内容.....	4
2 相关技术分析	6
2.1 Spring Boot 框架	6
2.2 Spring Cloud 框架	7
2.3 数据库相关技术.....	7
2.4 其它技术.....	9
2.5 本章小结.....	10
3 网上预约挂号系统的需求分析	12
3.1 业务概述.....	12
3.2 系统功能需求分析.....	13
3.3 系统非功能需求分析.....	16
3.4 本章小结.....	17
4 网上预约挂号系统的设计	18
4.1 系统架构设计.....	18
4.2 数据字典管理模块设计.....	20
4.3 订单管理模块设计.....	22
4.4 短信服务管理模块设计.....	24
4.5 用户管理模块设计.....	26
4.6 业务日志记录模块设计.....	28
4.7 系统数据库设计.....	30
4.8 本章小结.....	37
5 网上预约挂号系统的实现	38
5.1 系统开发环境.....	38
5.2 数据字典功能模块实现.....	38
5.3 订单管理功能模块实现.....	40
5.4 短信服务管理功能模块实现	44
5.5 用户管理功能模块实现.....	46
5.6 业务日志记录模块实现.....	48
5.7 本章小节.....	48
6 网上预约挂号系统的测试	49
6.1 系统的测试环境.....	49

6.2 功能测试.....	49
6.3 非功能测试.....	52
6.4 本章小结.....	55
7 总结与展望.....	56
7.1 总结.....	56
7.2 展望.....	56
参考文献.....	58

1 绪论

1.1 研究背景与意义

由于我国人口基数庞大,地区之间经济发展水平差异明显,人均医疗资源十分有限且分布不均匀,一方面导致患者面临“预约难、挂号难、就医难”,另一方面也有医院存在“病源不足、资源闲置”等问题。针对这些问题,预约挂号就是一种很好的解决方案,预约挂号就是患者提前预约就诊医院、就诊科室、就诊医生和就诊时间,并在指定的预约时间进行就诊^[1,2]。

网上预约挂号系统的后端采用 Spring Boot 框架与 Spring Cloud 框架搭建,前端采用 Vue 框架搭建,不仅可以保证系统的平稳运行,也可以提高可维护性和高扩展性。该系统一方面提供用户模块功能,方便用户登录注册、身份认证等操作;另一方面也提供在线预约挂号、取消预约和定时功能,便于医疗资源的科学合理利用,防止医疗资源浪费,便于患者合理安排自身时间;同时还提供后台配置、管理、查询和业务日志记录功能,方便对医院、科室和排班信息的管理,便于日后各种问题的排查。该系统使患者可以随时随地进行预约挂号,而无需每次都去医院,从而达到降低时间、金钱和精力的目的。不仅如此,患者还可以通过该系统对多家医院、科室和排班进行筛选,了解各个医生所擅长的领域和主治方向,为不懂这方面知识的患者减少许多不必要的麻烦,从而提高就医效率。同时还能够提升医院的知名度,让更多人可以从网络中了解到网络平台上的就医信息。在该系统上注册的医院,可以被全国范围内的用户们所搜索、查看、选择,从而提升医院的知名度,增加医院的就医人数,提高医疗资源的使用效率。

网上预约挂号系统的研究在医患之间建立起有效的信息交流平台,解决患者和医院面临的问题,使得患者不再面临“预约难、挂号难、就医难”的问题,并最大程度缓解医院“病源不足、医疗资源闲置”的问题。这样不仅可以减少患者就医等候时间、优化医院医疗力量配比、提高医院服务质量、提高患者就医体验,而且还会产生经济效益和积极的社会效应^[3]。

1.2 国内外研究现状

当前,最为常见的就医方式是患者亲临医院,导医根据患者的病态做出大致的就诊科室判断,推荐患者去指定挂号窗口排队挂相应诊断科室的号码,最后排队按号就医。互联网和移动通信技术的发展,使得患者能够利用互联网随时随地可以进行线上挂号。网上预约挂号系统的研发工作在国内外均有展开^[4]。以下对近几年来,国内外研发网上预约挂号系统的相关情况进行综述分析。

1.2.1 国外研究现状

在一些国家,譬如美国、加拿大、新加坡和日本,通过网上预约挂号系统进行预约挂号的现象已经普遍存在,在这些国家几乎大部分人都是通过这种方式,从而提高就医效率^[5,6]。

在美国、加拿大这样的经济和技术发达的国家,网上预约挂号系统的使用比较普遍。探其究竟,是因为他们的就医制度设置的比较完善^[7-9]。国外的大型医院在一般情况下只针对重型病人提供线下问诊机制,而对于其它的一般型病人,就必须先通过手机 APP、个人计算机等其它方式进行网上预约挂号,之后再根据结果去线下进行医治。如果医院暂时无法处理该病情,医院就会帮助患者开出转诊单,将患者转移到其它的医院进行就诊,同样也需要通过网上预约的方式来进行。这些就医手段和方式,不仅解决了医院存在的“病源不足、资源闲置”问题,同时也提高了患者的就医效率,并且在一定程度上也减少了医院线下的人流量,使得医院可以避免人满为患的状况^[10-12]。网上预约挂号系统在各个医院有较为成功且值得借鉴的经验。

又如新加坡,以初级综合诊所的预约系统为基础,推广电话预约。目前,为了在人满为患的情况下提高效率,患者需要等待很长时间。在新加坡,基层综合诊疗预约制已全面推行。无论是前来随访买药的患者,还是需要急诊的患者,都必须通过电话预约才能看病。但是他们的预约系统允许所有医院单独实施,他们还没建立一个全国统一的在线预约挂号平台^[13,14]。

在日本,部分部门实行预约制,尽管日本的经济是世界上最好的经济体之一,但并未实施在线预约系统。类似于我国传统的就医方式,就是患者到医院的门诊大厅排

队挂号，再凭号到相应的科室就诊^[15-17]。

1.2.2 国内研究现状

在我国，存在多种预约挂号的方式，包括线下挂号、手机电话预约挂号以及网上预约挂号。在我国虽然存在着除线下挂号以外的其它方式，但是也有许多医院，特别是小型医院由于各种原因并没有开放这些方式，导致这几种预约挂号方式的接受程度并不是很高，在近两年因为疫情的原因才导致网上预约人数逐渐上升。

使用人数最多的方式就是医院线下挂号，这是预约挂号的初级阶段，通过这种方式每位患者都需要到医院现场来进行预约挂号，并且需要患者在门诊窗口、服务中心等指定位置通过机器完成挂号，这样会导致医院比较拥挤，从而出现人满为患的现象。而且如果有些患者具有传染性的疾病，这种方式也将更容易导致疾病的快速传播。

手机电话预约挂号也是一种效率较高的方式，但是在我国的接受程度并不高。虽然许多医院也开放这种方式，但是却存在电话打不通、长时间未接通等一系列问题。这种方式一在定程度上也增加了人工成本，且通过这种方式无法实现实时付款，导致患者在预约成功后，仍然需要到现场排队付款、填写各种信息。这些问题都是导致该预约挂号方式不能成为主流方式的原因。

随着互联网技术的不断发展，通过互联网进行预约挂号的形式也逐渐成为主流。对于网上预约挂号这种方式，它可以解决线下挂号和手机电话挂号的种种弊端。但是对于不会使用智能手机或者电脑的老年人来说，如果要使用这种方式就需要子女帮忙完成；而且这种方式也需要使用微信或者支付宝来付款，部分中老年人不会使用这种付款方式仍需子女帮忙。在我国，虽然网上预约挂号的方式也在部分医院存在，但是推广效应还有待提高，例如在 2019 年江苏省的互联网医院也进入了快速建设期，在当时只有 7 家医院开通了互联网医院，因此使用的人数较少，在 2020 年出现疫情后才陆续增加了 80 家互联网医院，使用的人数逐渐上升。导致我国网上预约挂号的接受程度不高可能包括下述原因：1）医院未对外进行广泛宣传，导致了解的人比较少；2）系统功能的不完善，不仅导致就医效率没有得到提升，还在一定程度上适得其反^[18-20]。

国内对网上预约挂号系统的研发一直比较活跃。邢娜和郑蕾根据各大医院的实际情况和患者的用户需求,详细阐述了一个预约挂号系统的实际业务需要,总结了一个良好的预约挂号系统需要的一系列主体功能^[21];且之后使用 SSH 框架的 Java 技术实现网上预约挂号系统的一系列模块功能。此外,陈博、焦娣、吴雨柯和时娅楠综合当时已经存在的一些预约挂号系统,收集医院和患者的一系列需求,确定就医过程中的完整流程,提高预约公平性登记,进一步方便群众^[22-24]。张磊在他的文章中详细地阐述了一些现存医院的挂号现状,并对各类用户进行需求分析,从而将预约挂号系统划分成许多个模块,并设计各个功能模块各自业务流程,最后采用 MVC 模式来设计和实现目标系统^[25]。张雯霄在文章中收集了各个医院的资源,并进行整合,最后实现的网上预约挂号系统中提供统一的操作界面,从而满足医院和患者的各种需求^[26]。

黄巧和李慧宁在他们的文章中,使用 JavaWeb 作为系统的主要技术手段,包括其中的过滤器、Servlet 程序等,但系统使用到任何的框架,使得项目的开发工作变得繁琐^[27,28]。张明英、巩蕾在挂号系统开发使用到诸如 JSP 等技术,即 Java 服务器页面。JSP 作为一项古老的 Java 服务端页面技术,由于 JSP 页面在运行的时候会转换成 Servlet 程序,使得程序的运行效率下降,因此现在的项目基本上不怎么使用该技术^[29,30]。

刘晓龙、严春风和王永平在他们的项目中使用 SSM 框架来实现网上预约挂号系统。Spring 框架通过 IOC 和 AOP 特性,将对象之间的依赖关系交给框架控制,方便解耦,同时也简化程序开发;Spring MVC 框架拥有灵活的数据验证、格式化和数据绑定机制,对 web 层进行解耦,使得开发更简洁;MyBatis 框架采用映射文件的方式,解除 SQL 语句与代码之间的耦合度。但是随着技术的发展,也出现了诸如 Spring Boot 框架之类的技术,通过该框架无需过多复杂的配置即可搭建系统,因此 SSM 框架由于其配置文件复杂也逐渐被淘汰^[31-33]。

1.3 本文研究内容

本论文的主要任务是设计和开发网上预约挂号系统。根据当今社会中医院和患者的实际需求,利用软件工程技术,设计并开发出满足医院和患者需求的网上预约挂号

号系统。具体的研究内容如下：

（1）分析用户特特征，明确用户群体。针对不同类型的用户，进行系统功能需求分析，建立需求模型。同时进行性能需求分析，包括系统的吞吐量、稳定性、响应时间等多方面性能满足实用要求；

（2）设计网上预约挂号系统的体系结构，并细分为多个功能模块；对各个模块进行设计，包括对应的 UML 类图和时序图等；根据系统中数据的特点及其实体之间的关系先设计出对应的 E-R 图，然后根据 E-R 图转换成数据表；

（3）编码实现各个功能模块。基于 JavaWeb 开发相关的技术，实现数据字典模块、短信服务模块以及业务日志记录模块等多个模块功能；

（4）利用软件测试技术和工具，如 Apache 提供的 JMeter 压测工具，对开发的系统进行一系列功能和性能测试，确保系统质量。

2 相关技术分析

本章介绍在网上预约挂号系统的开发中应用到的主要技术。其中有用于后端开发的 Spring Boot 框架、Spring Cloud 框架以及数据库相关的技术，也有用于前端开发的 Vue 框架。

2.1 Spring Boot 框架

该网上预约挂号系统基于 Spring Boot 框架来搭建项目，搭建微服务项目所采用的技术是目前比较流行的 Spring Cloud 框架，并且和 Spring Boot 框架相匹配，可以极大地加快开发速度，简化开发过程，且许多第三方框架都与其有很好的融合。Spring Boot 框架设计出来的主要目的就是解决 Spring 框架使用中配置 XML 文件的痛苦，减少 Spring 框架中大量 XML 配置文件的配置操作带来的烦恼^[34]。

使用 Spring Boot 框架的主要目的就是代替之前使用非常复杂的 Spring 框架，以往使用 Spring 框架进行系统开发的时候，我们需要在 XML 配置文件中配置大量的 bean 标签，并且与 Spring MVC 框架和 MyBatis 框架等其它框架进行整合的时候也需要复杂的配置。在 XML 配置文件中，需要配置组件扫描、各种对象 bean、事务管理等，这些复杂的配置操作都会影响项目的开发效率。因此现在大部分项目的开发框架都会选择使用新诞生的 Spring Boot 框架，而不是陈旧的 SSM 框架。在 Spring Boot 框架中，我们不需要繁琐的配置文件，大部分的配置都是默认的，都是程序员们约定俗成的配置，即约定优于配置。如果需要修改默认的配置的话，只需要在 properties 配置文件或者 yaml 配置文件中进行简单配置^[35,36]。

Spring Boot 框架不仅具有开箱即用和约定优于配置的优良特性，同时能够很好地集成第三方插件。比如我们要在 Spring Boot 框架搭建的项目中使用 MyBatis-Plus 框架或者 Redis 数据库，仅需在 pom.xml 配置文件中添加对应依赖关系，即可轻松地集成第三方框架。此类项目都会集成一个父工程，该父工程的父工程是一个版本仲裁中心，管理着大部分依赖的版本号。我们添加依赖的时候，只需要写上 GAV 坐标中

的 G 和 A, 而不需要写上版本号, 这样会默认使用版本仲裁中心中所提供的版本号, 同时我们也可以显示地指定使用某个版本。

除此之外, Spring Boot 作为现今使用者最多的框架之一, 其官方也对它非常重视, 每隔一段时间都会有相应的版本更新迭代。

2.2 Spring Cloud 框架

该网上预约挂号系统使用 Spring Cloud 框架来搭建微服务系统。搭建微服务系统我们有多种框架可以选择, 除了 Spring Cloud 框架以外, 我们还可以选择 Dubbo 框架, 但是由于 Spring Cloud 框架具有一系列的优良特性, 比如适用于不同环境、约定优于配置, 此外还有快速启动、开箱即用等优良特性, 所以我们采取后者来搭建微服务系统^[37]。

Spring Cloud 框架基于 Spring Boot 简化分布式系统的各种基础设施的开发, 这些都可以通过 Spring Boot 框架来做到一键运行。Spring Cloud 框架也并不是重复造轮子, 它只是把现存的各种成熟的技术组合到一起, 通过 Spring Boot 框架的展现形式展现出来而已, 并掩盖其内部的复杂实现细节与各种各样的配置, 最后向外界提供一个简易工具包。

使用 Spring Cloud 框架搭建微服务应用的话也是非常的简便, 而且该框架与 Spring Boot 框架的集成也是非常高, 与其它一些经常要使用的第三方插件的融合度也很好, 配置也及其的简单。Spring Cloud 生态主要划分为两大体系, 我们主要使用的是 Spring Cloud Alibaba 系列的模块功能, 使用这些可以给我们系统提供良好且优秀的性能^[38,39]。

2.3 数据库相关技术

对于用户、订单和支付等数据保存在关系型数据库 MySQL 中, 因为对于这些数据而言, 不仅需要存储数据本身, 还需要存储它们之间的关系。在该系统中, 医院信

息、科室信息以及排班信息由于其庞大的数据量，将其保存至非关系型数据库 MongoDB 中，因为该数据库适用于表结构不明确且数据不断变大、数据量很大或将来会变得很大等场景。此外，在通过手机发送验证码、微信扫码登录、微信扫码支付或者支付宝扫码支付的时候，可把手机验证码、微信二维码或者支付宝二维码保存至 Redis 数据库中，且设置一定的过期时间，因为该数据库具有存取速度快、便于设置数据有效期等特点。MyBatis-Plus 作为 MyBatis 框架的升级版，由国人所开发，鉴于其效率的高效、操作的便捷，使用它用于对 MySQL 数据库的操作^[40]。

MySQL 不仅在各种规模的公司中十分流行，还与 Oracle 等数据库相比起来有着许多的优良特性。最为广大公司接受的优势就是它的免费，许多数据库都是需要收费的，而且收费数额也不低，这就导致许多中小型公司因无法承受高额的费用而放弃使用付费数据库。相反地，MySQL 数据库在这方面就吸引了众多的中小型公司，该数据库现在是大部分公司系统上主要使用的数据库。在 MySQL 数据中中可以划分为多个库，每个库中也划分多个数据表，每个数据表中会保存数据及其之间的联系^[41]。

可以作为持久层的框架有许多，曾经广为使用的 Hibernate 框架是一种全自动的 ORM（Object Relation Mapping，对象关系映射）框架。正由于该框架的全自动，当自己需要实现一些个性化需求时，就会显得十分麻烦，并且该框架在使用上与其它技术相比难度也是巨大。后来出现的 MyBatis 框架，是一种半自动的 ORM 框架，相比 Hibernate 框架来说在使用上和效率上都有较大的提升。在 MyBatis 框架中，可以将 SQL 语句单独写在一个 XML 文件中，称为 mapper 映射文件，通过调用 mapper 接口中的方法去匹配相应的 SQL 语句，而我们并不需要去实现 mapper 接口中的这些抽象方法。除此之外，该框架还提供了多种获取参数值、逆向工程等功能。基于此，MyBatis-Plus 框架也应运而生，其理念是为效率而生，在 MyBatis 上只做增强，不做改变。使用该框架大部分对单表的操作我们都不需要自己写 SQL，通过调用其提供的方法即可。它提供的通用 mapper 和通用 service 为持久层的开发提供了很大的便利，若涉及到多张表的操作，也同样可以使用 MyBatis 框架的方式来实现相应的功能。

Redis 数据库与 MySQL 数据库不同，它将需要保存的数据缓存在运行内存之中，

可以极大程度地提高系统的运行速度，与 MySQL 数据库把数据保存在磁盘中相比，是一种大优势。除此之外，在 Redis 数据库中还具有丰富的数据类型，每一种数据类型都有多个操作指令，也有自己所适应的场景。比如 string 数据类型适应做一些简单数据的缓存，比如缓存手机验证码，搭配时效性一起使用最佳；zset 数据类型保存的数据是有顺序的，就可以用来实现一些诸如排行榜的这种功能。Redis 作为一种内存级别数据库，在项目运作的时候把所有的数据都是存放在其运行的内存之中，由于内存中的数据的易失性，基于数据安全性等多方面的考虑，Redis 数据库同时也提供数据持久化策略，使用这些持久化策略就可以在一些必要的场合中将内存中的数据保存在文件中，在合适的时候又恢复到内存之中^[42,43]。

2.4 其它技术

该系统使用 Nacos 注册中心进行服务注册，系统会将所有的微服务注册到 Nacos 注册中心中，以供其它的微服务进行远程调用，同时也是与 Spring Cloud 框架搭配使用的一款优秀组件。系统还使用 Feign 组件用于微服务之间的远程调用，Feign 组件也是与 Spring Cloud 框架搭配使用的一款优秀组件。当前端访问不同微服务的时候，不同微服务的 IP 地址和端口号可能不同，这时如果由前端来处理该类问题，将会显得十分麻烦，这里我们采用 Spring Cloud Gateway 服务网关来处理该类问题，同时它也可以用来解决跨域问题，这是区别于使用 Nginx 反向代理服务器来解决该问题的重点^[44-46]。

在传统不使用 Maven 工具的系统中，需要手动地管理各个 jar 包及其版本，这个工作是十分麻烦的。当使用 Maven 工具后，只需要在其 pom.xml 配置文件中通过 GAV 坐标添加相应的依赖即可，如果依赖还有其需要的前置依赖，也会自动引入项目中。Maven 所管理的项目会被划分为多个生命周期，对应提供相应的执行命令，比如通过 mvn test-compile 命令来编译测试源代码。除此之外，还提供诸如中央仓库、本地仓库等概念，由于中央仓库放在国外，访问起来的速度相对较慢。因此我们通过在 Maven 的核心配置文件 settings.xml 中设置阿里镜像等方式，提高下载资源的速

度，还可以在配置文件中配置本地仓库路径等。

系统使用 Git 分布式版本控制工具来进行版本控制，使用 Gitee 代码托管中心来进行代码托管。GitHub 同样是一种代码托管中心，由于其服务器放置在国外，导致访问速度很慢，而 Gitee 由于服务器放置在国内，访问速度较快，因此系统使用 Gitee 作为代码托管中心。

在后台管理子系统中的数据管理模块中，保存着一些固定不变的数据，即一些数据字典数据，这里可以使用导入导出的功能以 Excel 表格为媒介对它们进行操作，这里使用阿里巴巴提供的 EasyExcel 技术来操作表格。在后台管理子系统中还会有统计模块，用于统计所有用户的一些预约挂号信息，这里我们采用 ECharts 插件来展示这些统计信息。在用户进行身份认证的时候，需要上传他们的证件照片，那么证件照片需要保存至某个地方，我们使用阿里云 OSS 分布式文件系统来作为存储证件照片的媒介。

ECharts 插件是一款非常方便的可视化工具，它提供多种各式各样的可视化形式，包括图表、柱状图、饼状图等等形状。除此之外，ECharts 插件的使用也非常的简单，而且在 ECharts 的官网上，对于每种图形的使用都有一些简单的例子来供我们自己学习，我们仅仅需要根据官网所提供的例子修改一些参数，提供指定形式的数据即可展示相应的可视化界面^[47]。

Apache JMeter 是 Apache 组织提供的一款高性能的测试工具，经常被广泛用于系统的压力测试。JMeter 工具简单易用，也易于安装，且官方提供绿色版本以供使用。相比其它的需要 http 测试工具而言，JMeter 最大的优点在于其扩展性强，JMeter 能够自动扫描其 lib/ext 子目录下 jar 文件中的插件，并且将其装载到内存，让用户通过不同的菜单调用^[48-50]。

2.5 本章小结

本章对网上预约挂号系统中所使用到的一些主要技术做了简单的介绍。这些技术包括用于系统搭建的 Spring Boot 框架和 Spring Cloud 框架，用于保存数据的

MySQL 数据库、Redis 数据库等；此外，还包括用于数据可视化的 ECharts 插件，提供发送短信服务的阿里云短信服务工具，以及用于存储用户的证件照片的阿里云 OSS 分布式文件系统。

3 网上预约挂号系统的需求分析

本章的主要任务是进行网上预约挂号系统的需求分析。结合挂号就医的业务流程，针对系统不同的用户群体，展开系统的功能性需求分析和性能需求进行分析。

3.1 业务概述

网上预约挂号系统使用于各大医院之间，为医院、患者们提供便捷服务。该系统面向的人群分为以下三类：系统管理员、患者和医院。网上预约挂号系统是为患者提供预约挂号服务的平台，也是分配、管理门诊医疗资源的工作平台，使用者以患者为主。具体地说，本系统需要实现以下需求：

（1）提供用户模块功能

该系统为用户提供多种登录方式，包括微信扫码和手机验证码等。对于已注册的用户而言，不仅需要提交身份认证信息并得到审核，挂号之前还需添加患者信息，以便后续的预约挂号。

（2）提供在线预约挂号、取消预约和定时功能

在该系统中，患者们可以根据自身实际情况选择目标医院、目标科室和目标排班，在此基础上选择已添加的患者，最后选择微信扫码、支付宝扫码或者银行卡的方式付款。为患者提供查询预约和取消预约的功能，便于医疗资源的科学合理利用，防止医疗资源浪费，便于患者合理安排自身时间。当取消预约时，系统会根据用户是否已付款判断是否退款。对于已预约并付款成功的患者，在就诊当日会定时收到就诊短信，提醒患者及时就诊。

（3）提供后台配置、管理和查询功能

系统管理员可以在后台管理子系统中配置各个医院的信息，对各个医院进行统一管理，比如上传医院信息、科室信息和排班信息。还可以对数据字典信息、用户信息、订单信息和统计信息管理，统计信息可以将预约信息以图表的形式展示。

（4）提供业务日志记录功能

系统的运转不可能总是正常的，当发生异常时需要定位问题、排查问题、解决问题。因此系统提供业务日志记录功能，记录系统运作过程中的各个细节，便于日后问题的排查。

3.2 系统功能需求分析

3.2.1 后台管理子系统

在后台管理子系统中包括多个功能模块，有医院管理、数据字典管理、用户管理、订单管理和统计管理功能模块。后台管理子系统的 UML 用例图如图 3-1 所示。

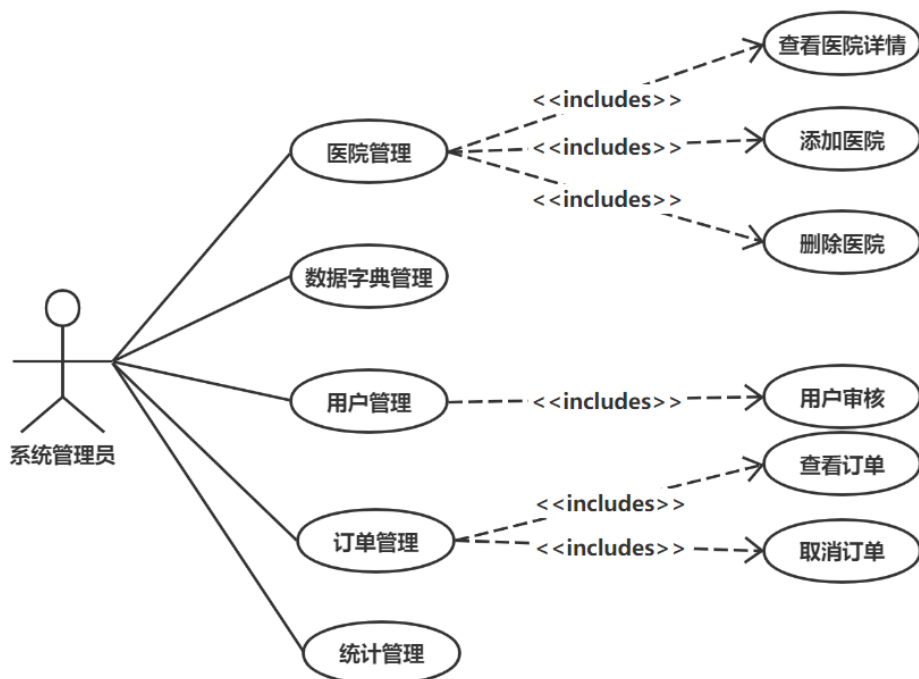


图 3-1 后台管理子系统用例图

(1) 在医院管理模块中，需要提供医院列表来展示所有医院的详情，且可以进行删除、锁定和修改操作，也可以进行查看、排班和下线等操作；还需要提供医院添加功能来添加医院；

(2) 在数据管理模块中保存一些固定不变的数据，包括行政区划信息、医院等级信息、民族信息和学历信息等，且可以通过 Excel 表格的方式进行导入导出操作；

(3) 在用户模块里首先需要有用户列表功能，已成功注册的用户信息将会展示

在此处。还需要提供用户认证审批功能，新注册的用户需要提交认证信息进行认证审核，这些用户的审核信息会显示在用户审核界面，系统管理员可以对这些用户审核，审核成功的用户才可以进行预约挂号；

(4) 在订单模块中对患者提交的订单进行管理，系统管理员可以查看订单、取消订单；

(5) 在最后一个统计管理模块中，可以统计预约挂号信息，比如查询指定时间范围内的预约挂号数量的变化趋势，通过折线图的方式进行展示，用于支持后续的数据分析操作。

3.2.2 前台用户子系统

在前台用户子系统中包括多个功能模块，有登录管理、身份认证管理、患者管理和预约挂号管理功能模块。前台用户子系统的 UML 用例图如图 3-2 所示。

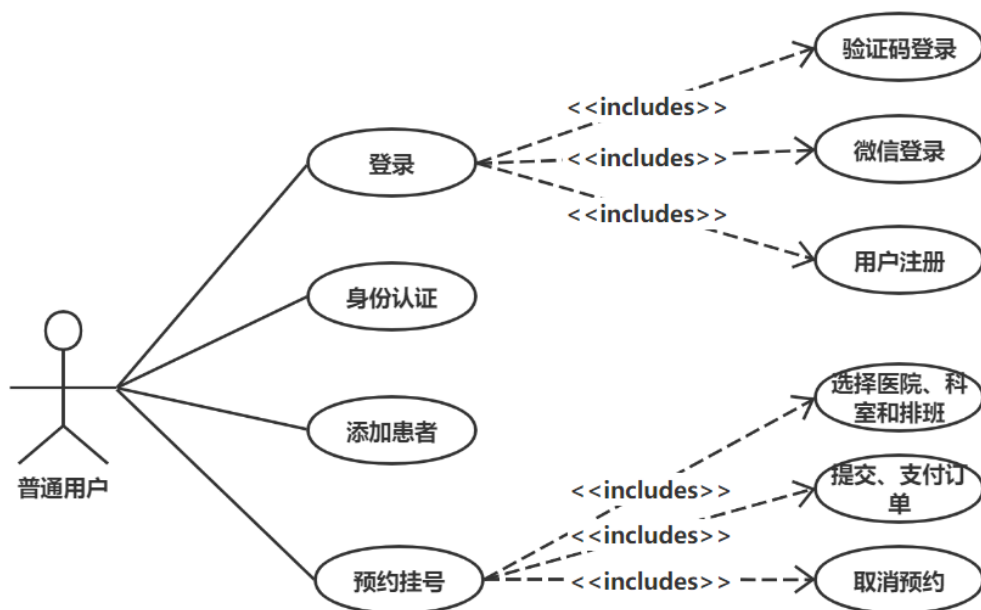


图 3-2 前台用户子系统用例图

(1) 首先需要包括用户登录、注册功能，用户可以使用短信验证码和微信扫码两种方式。对于第一次使用微信扫码登录的用户，需要绑定手机号码。第一次使用短信验证码方式登录时，若该手机号码是未注册的，系统会对该手机号码自动注册；

(2) 用户注册成功后，需要进行实名认证，即身份认证，认证成功并审核通

过后才可以预约挂号；

(3) 用户登录并通过身份认证，可以在个人信息中添加患者信息，这些信息在预约挂号时可以直接使用；

(4) 当用户进行实名认证并添加患者后，就可以预约挂号。首先需要确定目标医院、目标科室和目标排班，实现选择医院、科室和排班功能。其次需要提交订单并选择支付方式并生成支付二维码，实现提交并支付订单功能。最后用户可以选择取消预约，根据是否已付款进行退款流程，实现取消预约功能。

3.2.3 业务日志记录功能

网上预约挂号系统正式上线以后，除了系统的正常运行，还可能会出现各种问题，比如系统奔溃、运行错误以及系统响应速度慢，这些问题无一不是降低用户体验感的原因。为了解决这些问题，需要采取一些手段。一方面，可以通过操作系统的任务管理器来监控系统的运行情况，比如 CPU 占有率，但是该方法也仅仅只能大体查看系统运行状态，而对于系统运行的内部细节还是有所欠缺；另一方面，就是系统需要设计出业务日志记录功能，这样可以在系统运行的时候记录下各个方法的调用情况，包括方法名、方法参数以及调用的起始时间和结束时间。

实现系统的业务日志记录功能除了需要满足以上要求，该功能还不能对系统其它功能的正常运转产生任何影响。因此本系统遵循“对修改关闭，对扩展开放”的设计原则，该原则要求“添加一个功能应该是在已有代码基础上扩展代码，而非修改已有代码”，业务日志记录功能就是在现有功能的基础上进行添加，不对其他任何功能进行修改，不会对现有功能产生任何影响。其次，我们还需要能够根据业务日志记录的内容精确定位到系统出现故障的位置，并能够分析故障原因且解决它。基于这些点，对日志记录功能的要求归纳为如下两点：

(1) 不影响正常运行

业务日志记录功能的实现在系统原有功能的基础上添加上去，是对业务功能的日志记录，是对业务功能的修饰，因此不能影响到正常代码的正常执行，通过 Spring 框架所提供的 AOP (Aspect of Programming, 面向切面编程) 可以实现这个功能。

(2) 设置全局标识

系统日志的内容包括调用方法的方法名、调用的起始时间和形参列表等,该功能需要将这些信息包装起来,保存至日志记录队列中,最后通过一个单独的线程对日志记录队列中的信息保存至数据库中。对于每条记录都需要设置一个全局标识,用于区分日志记录,这样可以更方便查看记录、定位故障点以及解决问题。业务日志记录功能的 UML 用例图如图 3-3 所示。

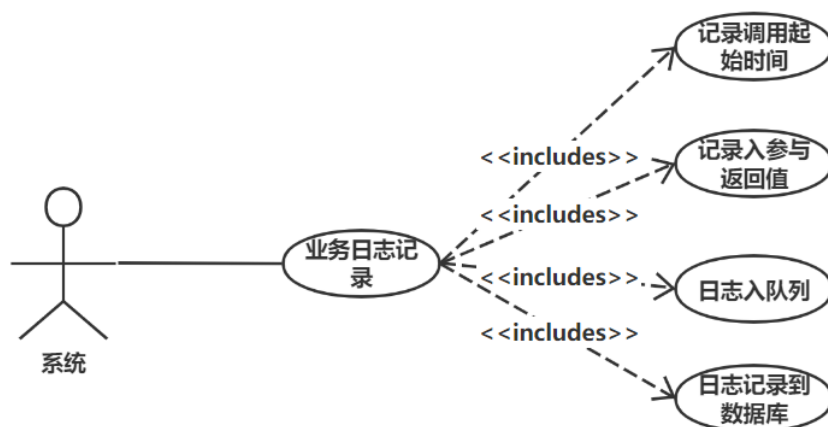


图 3-3 业务日志记录用例图

系统实现业务日志记录功能,首先需要记录业务方法调用的起始时间、业务方法的入参和返回值等信息,然后将这些信息封装成一个整体保存到日志队列中,最后启动一个单独线程将日志信息记录到指定数据库中。

3.3 系统非功能需求分析

网上预约挂号系统的非功能需求指的是非功能性的特性,这些功能用户在使用时无法明显感受,但是会隐式影响用户使用系统的体验,因此系统需要满足稳定性、可扩展性等多方面的性能需求。

(1) 稳定性。在系统正常运行的过程中,出现诸如页面乱码、用户访问速度缓慢以及系统奔溃等问题,就是出现系统稳定性问题。在设计系统时,需要把这些问题考虑进去,减少这类问题的发生,使得系统达到高内聚、松耦合的特征。在系统开发过程中也需要保证程序代码的质量、软件架构的稳定,这样当用户需求产生变动时,就可以把对系统的影响降到最小;

(2) 可扩充性和可维护性。世界万事万物都是普遍联系、永恒发展的，面对随时可能产生变动的技术体系和用户需求，设计出可维护性和可扩充性高的系统，以最低的成本应对随时可能发生的变化，对于系统的生命周期至关重要。此外，在系统进入运行的前期，也仅仅只会加入部分医院、部分科室和排班来测试系统，在后期，会根据系统的运行情况和医院的加入意愿来加入更多的医院，此时就更需要系统满足可扩充性和可维护性；

(3) 系统性能。主要是需要保证系统的并发性，即保证同一时间如果有大量用户访问该系统也可以提供稳定的服务。本系统的设计目标是支持 3000 个用户的并发访问；除了并发量的要求以外，系统性能还要求各种操作的响应时间符合预期，各个接口请求的最大响应时间都应该控制在 1s 秒钟左右，对于用户登录接口的响应时间要求在 500 毫秒左右，对于订单生成接口的响应时间由于受订单量的影响要求在 1s 左右。

3.4 本章小结

本章主要对系统进行需求分析。首先确定用户群体，并提出系统高层需求；其次从功能性需求出发将系统划分为后台管理子系统和前端用户子系统，并对它们的整体功能进行详细、正确的描述；最后进行系统的非功能性需求分析。

4 网上预约挂号系统的设计

本章的主要任务是进行网上预约挂号系统的系统设计。首先就软件层次架构展开分析，其次阐述系统各个功能模块的设计方案，并给出对应的 UML 类图和 UML 时序图，最后进行系统数据库的设计，根据需求画出 E-R 图，再根据 E-R 图设计出具体的表结构。

4.1 系统架构设计

架构相当于建造建筑物时的整体框架，系统架构类似于项目的结构。网上预约挂号系统的架构设计包括了两个方面，一方面是系统的软件层次架构，这里体现的是三层体系架构思想；另一方面是系统功能模块划分，后续针对这些模块进行设计与实现。

4.1.1 系统的软件层次架构

该网上预约挂号系统采用的是 B/S 模式，一共包括三次层次，分别为上层表现层，中层业务逻辑层和下层数据持久层。在业务逻辑层中包含一系列的微服务，分别为医院模块、订单模块、短信服务模块和用户模块等多个模块，这种使用不同的微服务代表不同的功能模块的方式不仅可以提高程序开发人员的开发效率，而且当某个功能模块需要进行修改时，只需将业务逻辑层中对应的逻辑代码进行修改就可实现。

这种多层架构模式的设计不但可以分离程序开发人员的关注点、做到模块无损替换，而且还能降低系统各个模块之间的耦合度，提高软件的可复用性与高扩展性。在前端 UI 层与业务逻辑层之间采用 Spring Cloud Gateway 服务网关和 Nacos 注册中心进行请求转换，数据层则采用 MySQL 数据库、Redis 数据库等三种类型的数据库用于保存不同特点的数据信息。图 4-1 展示了网上预约挂号系统的软件层次架构设计。

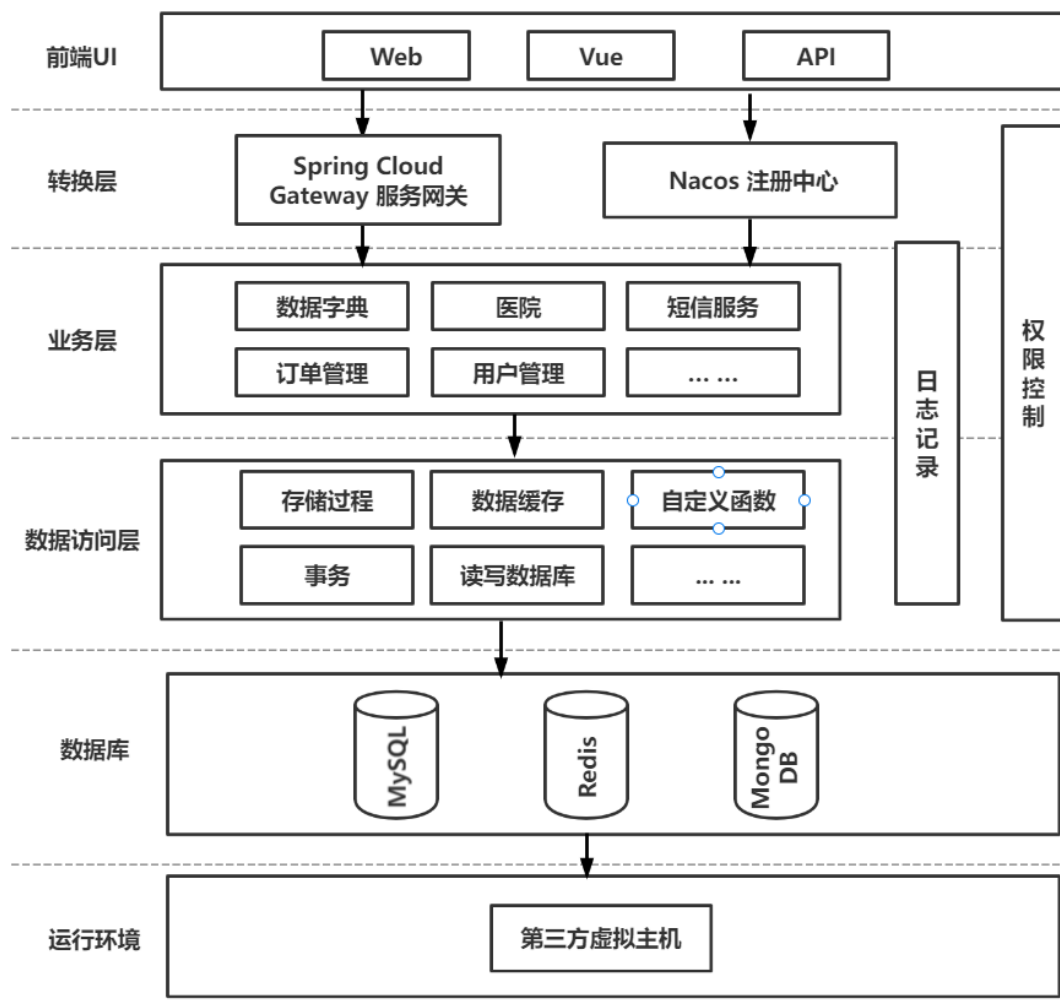


图 4-1 网上预约挂号系统的体系架构图

首先是前端 UI 设计，即用户可以直接看到的系统页面信息。在这里用户可以查看到系统所有页面上的前端信息，并且可以根据自己的需要进行操作，最终得到系统的反馈。前台用户子系统的表现层包括多个前端界面，有登录界面、用户界面、科室选择界面、排班选择界面、订单界面、支付界面、认证信息提交界面和添加患者信息界面等。

第二层为转换层。因为不同的微服务的 IP 地址和端口号可能不同，这种问题如果交给前端来处理会比较麻烦。这里采用的是 Spring Cloud Gateway 服务网关技术处理这类问题。此外，还使用 Nacos 注册中心来注册服务。

第三层为业务逻辑层。采用的是微服务的实现方式，不同的功能模块设置成不同的微服务，各个微服务之间的 IP 地址和端口号不一定会相等。该层首先接收前端发

出的 http 请求, 再进行一些处理, 包括业务处理、数据校验等操作, 最后通过数据层得到目标数据, 并返回给上一层使用。

第四层和第五层是数据访问层。在该层主要是对各种数据库的操作, 包括 MySQL 数据库和其它的非关系型数据库。

最后一层为系统的运行环境。在开发阶段, MySQL 数据库、Redis 数据库和 MongoDB 数据库都运行在本地的虚拟机中或阿里云服务器中, 而前后端代码以及 Nacos 注册中心则是运行在 Windows 本地。

4.1.2 系统功能模块划分

通过在上一章中对系统的需求分析, 可以将该系统划分多个模块, 包括数据字典、订单管理、短信服务管理、用户管理和业务日志记录等。数据字典模块下包括行政区划、医院等级以及民族信息等; 订单管理包括提交订单、支付订单以及取消订单等; 用户模块包括用户登录与注册、身份认证以及添加患者等; 短信服务模块包括发送验证码与校验验证码。除这些主要功能模块以外, 还提供系统的业务日志记录功能模块, 下面将对几个主要的功能模板进行设计。

4.2 数据字典管理模块设计

数据字典模块在该系统起着举足轻重的作用, 该模块维护一些固定不变的数据, 以供系统正常运行使用。维护的信息包括行政区划信息、医院等级信息和民族信息等。

(1) 行政区划信息管理。保存的是全国范围内所有的行政区划信息, 因为这些数据基本上是固定不变的, 因此将其作为数据字典管理。在后台管理子系统的数据管理模块中可以查看到这些数据, 并且在前台用户子系统的主页面可以根据该信息进行医院筛选的操作, 或者在提交用户认证信息、添加患者信息选择家庭住址时使用该数据;

(2) 医院等级信息管理。保存的是医院所有的等级划分信息, 因为这些数据基本上是固定不变的, 因此将其作为数据字典进行管理。在后台管理子系统的数据管

理模块中可以查看到这些数据，并且在前台用户子系统的主页面中可以根据该信息进行医院筛选的操作；

(3) 民族信息管理。保存的是全国五十六个民族信息，因为这些数据基本上是固定不变的，因此将其作为数据字典进行管理。在后台管理子系统中的数据管理模块中可以查看到这些数据，并且在前台用户子系统提交用户认证信息、添加患者信息选择民族时使用到。

在数据字典模块中，定义 Dict 实体类来保存数据字典信息。创建 DictController 类、DictService 接口、DictServiceImpl 类和 DictMapper 接口来分别表示控制层、业务逻辑层和数据持久层。其中 UML 类图设计如图 4-2 所示。

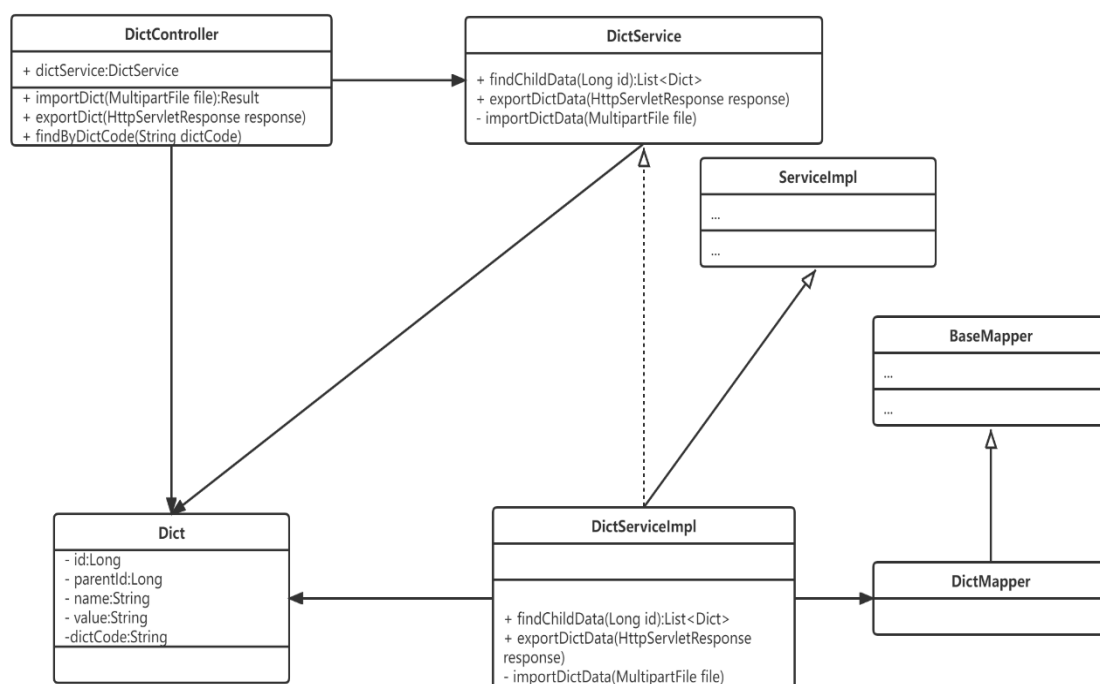


图 4-2 数据字典类图

系统查询数据字典信息的时候首先访问到 DictController 控制层，再通过 DictService 业务逻辑层来间接访问数据库进行查询，最后将得到的数据返回给系统。在该模块中系统可调用 importDict 方法与 exportDict 方法通过访问 DictController 控制层来实现数据字典信息的导入导出功能，并以 Excel 表格为数据传播媒介。数据字典模块的 UML 时序图如图 4-3 所示。

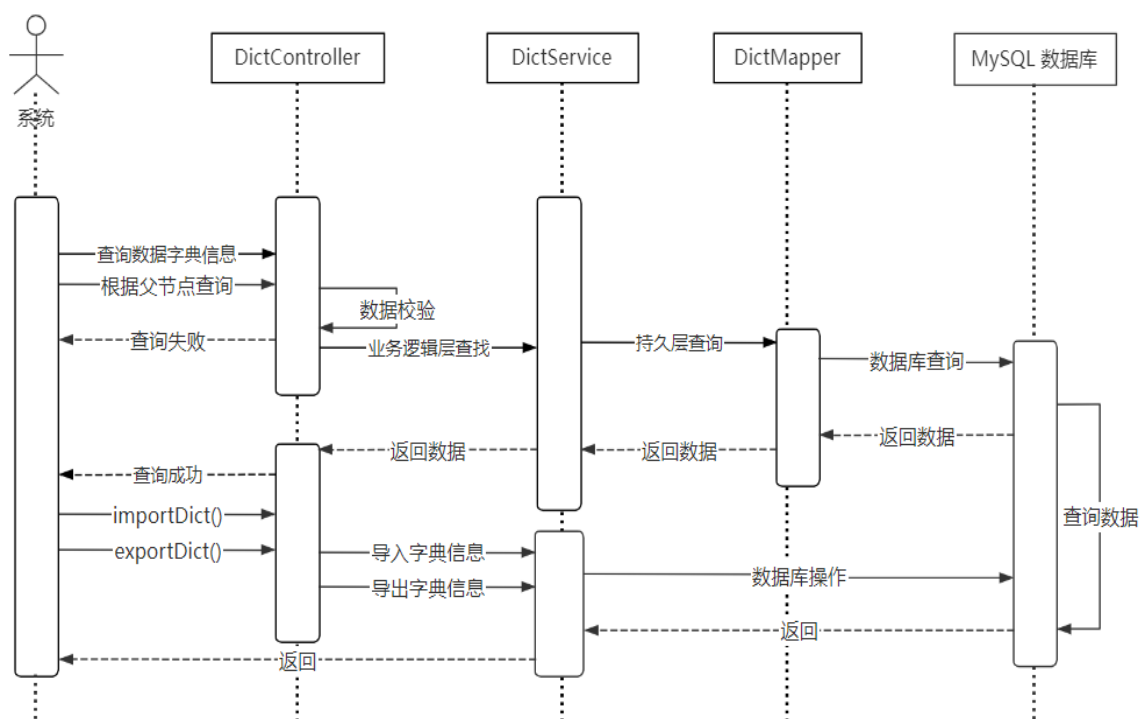


图 4-3 数据字典时序图

4.3 订单管理模块设计

该功能模块是系统核心模块之一，用户选择某个目标排班以后就会生成订单，对应着提交订单子模块；之后可以选择支付订单，对应着支付订单子模块；无论有没有进行支付订单，都可以选择取消预约，对应着取消订单子模块。

(1) 提交订单。用户首先需要在主界面选择目标医院，选择后可以查看到该医院下所有的科室信息，选择目标科室后可以看到该科室下所有的排班信息，然后可以选择上午或者下午的排班，也可以看到剩余预约数等信息。确定目标排班后，需要选择患者，最后提交即可完成提交订单；

(2) 支付订单。用户提交订单后可以支付订单，也可以取消订单，并且提供了各式各样的支付形式，包括微信支付、支付宝支付以及各种类型的银行卡的支付操作；

(3) 取消订单。在提交订单后，或者支付订单后，都可以选择取消预约来取消订单；如果是已支付订单，退款将会原路返回到付款账户中。

在该模块中，定义 `OrderInfo` 实体类保存订单信息，创建 `OrderController` 类、`OrderService` 接口、`OrderServiceImpl` 类和 `OrderMapper` 接口分别表示控制层、业务逻辑层和数据持久层，另外 `ServiceImpl` 类和 `BaseMapper` 接口是 MP 框架所提供的。其中 UML 类图设计如图 4-4 所示。

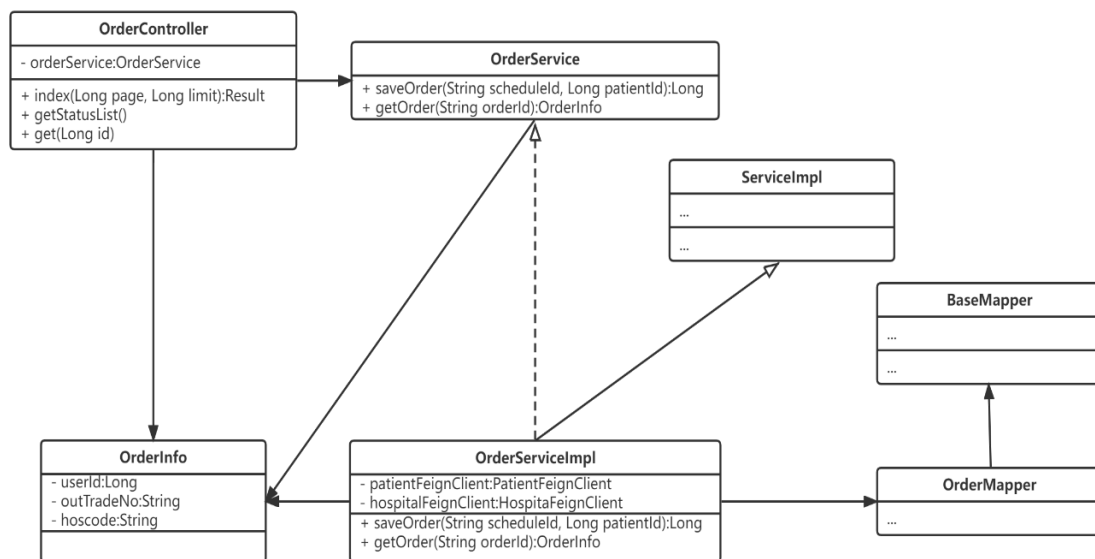


图 4-4 订单管理类图

用户进行预约挂号操作涉及到订单管理功能，当用户选择好目标医院、科室和排班后，就会进入到 `OrderController` 控制层中。在该层中首先会进行一个简单的登录判断，如果系统检测到用户尚未登录，则返回给用户相关信息，跳转到登录界面中。已登录用户提交订单，进入到 `OrderService` 业务逻辑层中，在该层中首先会判断号源数是否足够，若足够，订单提交成功，保存到 `MySQL` 数据库中，若号源数不够则返回不成功信息给用户。

提交订单以后，无论是否完成支付订单的操作，都可以取消订单，并返回结果给用户。首先将取消订单的请求发送到 `OrderController` 控制层中，从而间接去修改 `MySQL` 数据库中订单相关信息。其次会判断该订单是否完成支付，若已完成支付，则调用支付接口实现各种支付方式的退款操作。

在从 `OrderController` 控制层提交订单到 `OrderService` 业务逻辑层后，还会涉及到支付订单的操作，首先将支付订单的请求发送到 `OrderController` 控制层中，从而间接

去调用支付接口实现各种支付方式的支付功能，并修改 MySQL 数据库中订单相关信息，将成功信息返回。订单管理模块的 UML 时序图设计如图 4-5 所示。

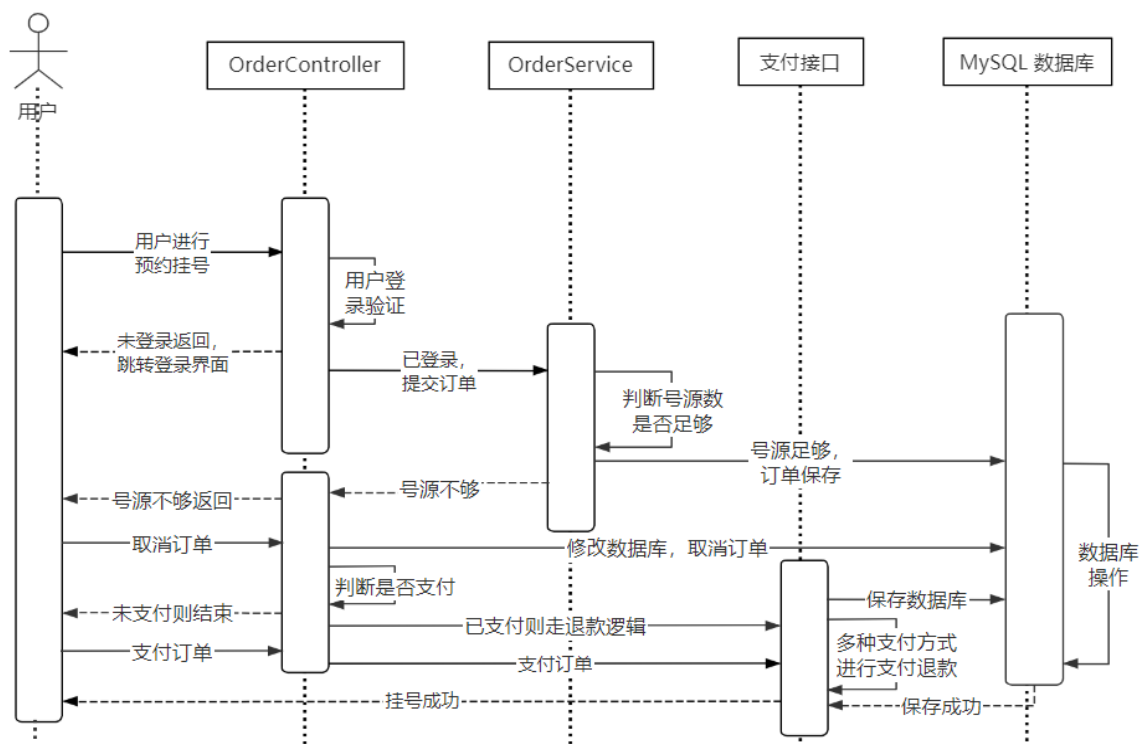


图 4-5 订单管理时序图

4.4 短信服务管理模块设计

短信服务管理模块在该系统中同样起着至关重要的作用，主要用在用户登录注册中。该模块包括发送手机验证码子模块和校验手机验证码子模块两方面。

(1) 发送手机验证码。当用户登录系统后，可以选择手机验证码登录，首次登录的用户会自动注册。这里后端会把手机号和验证码作为 key 和 value 保存至 Redis 数据库中，同时设置一分钟的时效性；

(2) 校验手机验证码。当用户输入手机验证码后，后端接收到该数据会去对比用户的输入和 Redis 数据库中保存的数据是否一致，根据比对结果来判断最终的登录成功与否。

在短信服务管理模块中，创建 MsmApiController 类作为控制层组件，MsmService

接口和 `MsmServiceImpl` 类作为业务逻辑层组件，为简化操作，在这里对 Redis 数据库的操作封装在业务逻辑层即可。由于这里涉及到阿里云短信服务和随机验证码的功能，还需要创建 `ConstantPropertiesUtils` 和 `RandomUtil` 两个工具类以供使用。其中 UML 类图如图 4-6 所示。

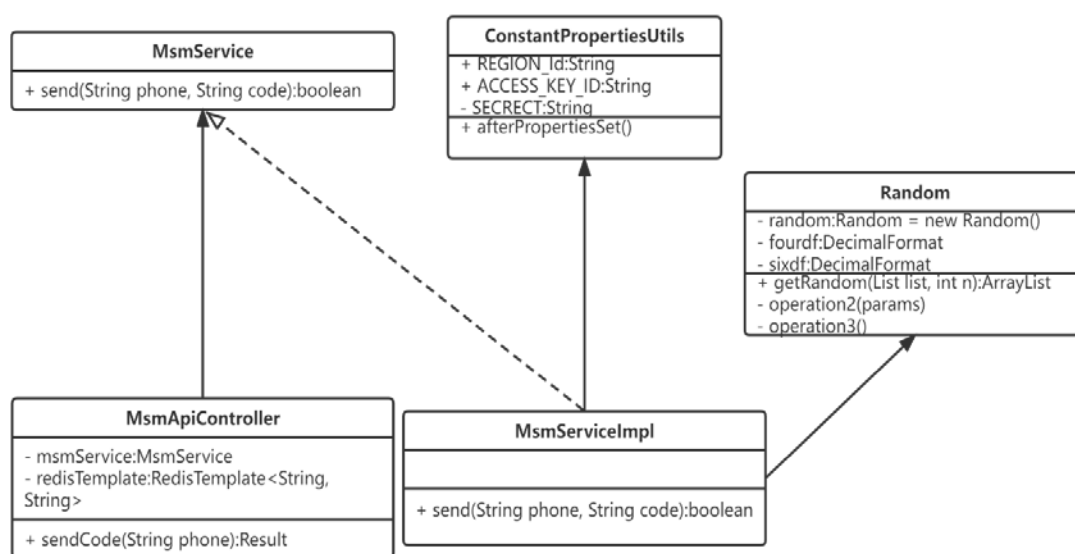


图 4-6 短信服务管理类图

在短信服务管理模块中，用户首先会点击发送验证码按钮，该请求就会进入到 `MsmApiController` 控制层中。控制层中发送请求查询 Redis 数据库中该手机号码对应的验证码是否存在，如果存在的话，就什么也不做直接返回给用户；如果验证码不存在则在 `MsmApiController` 控制层中使用工具类生成六位数的随机验证码，并通过 `MsmService` 业务逻辑层将验证码保存至 Redis 数据库中，同时访问阿里云短信服务发送短信。

当用户发送校验验证码请求到控制层时，先校验验证码和手机号的格式，若格式错误则返回错误信息给用户；否则去查询 Redis 数据库中该手机号码对应的验证码，并与用户输入的验证码进行对比，并将对比结果返回给用户。

在该模块中还提供 `ConstantPropertiesUtils` 工具类来操作阿里云短信服务，其中包括域 id 和密钥等属性。此外还提供 `RandomUtil` 工具类来产生随机验证码，支持六位数的验证码，底层采用 Java 基础类库中提供的 `Random` 类所提供的随机数生成方

法。短信服务管理模块 UML 时序图如图 4-7 所示。

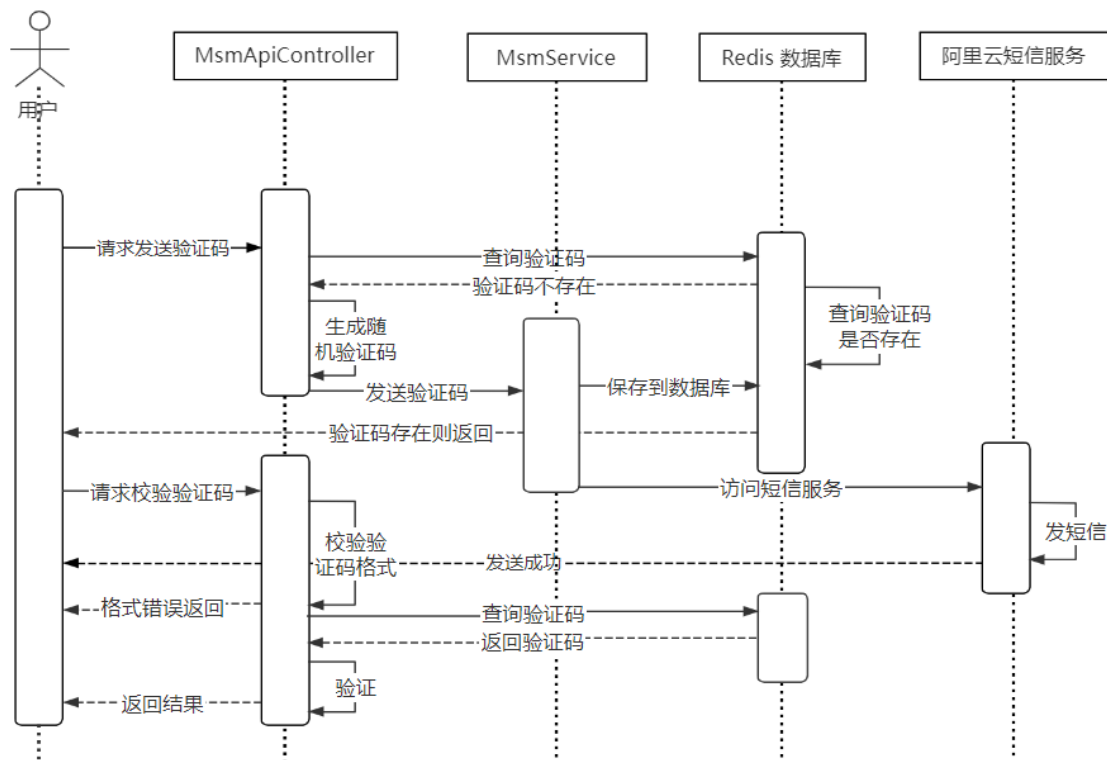


图 4-7 短信服务管理时序图

4.5 用户管理模块设计

一个优秀的系统对于用户管理模块的设计是非常重要的，对于用户数据的保护、身份的认证和患者信息的维护，在设计中都应该充分考虑到。该模块包括用户登录与注册、身份认证和添加患者子模块。

(1) 用户登录与注册。对于第一次使用网上预约挂号系统的用户，在没有登录的情况下只能显示一些基本医院和科室等信息，倘若需完成其它与挂号相关的操作就要先进行登录。登录支持手机验证码和微信扫码两种方式，并且第一次登录的用户会自动注册，第一次使用微信扫码的方式也需要绑定手机号；

(2) 身份认证。当用户第一次注册并登录后，仍然无法正常使用该系统中的功能。用户首先需要在身份认证界面进行身份认证，包括上传证件照片等，待审核通过

后即可使用该系统；

(3) 添加患者：在身份认证审核通过后可以给当前账号添加多个患者信息，这样在提交订单后可以直接选择已经被添加好的患者。用户管理模块 UML 类图如图 4-8 所示。

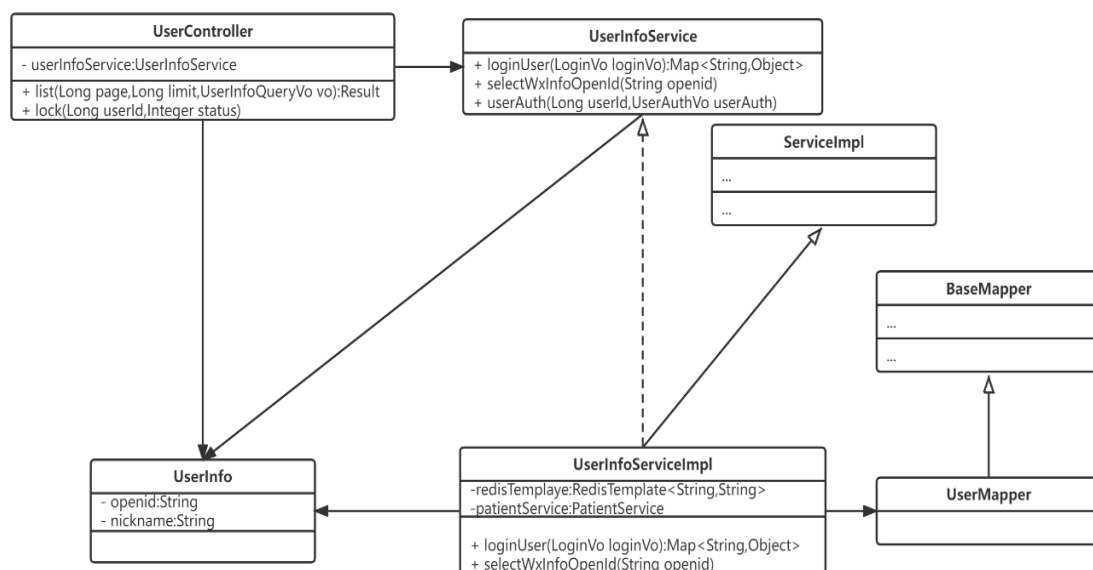


图 4-8 用户管理类图

在该模块中，创建 **UserInfo** 实体类用户保存用户信息，**Patient** 实体类保存患者信息，**UserController** 类作为控制层组件，**UserInfoService** 类和 **UserInfoServiceImpl** 类作为业务层组件，**UserMapper** 接口作为数据持久层组件，由于该模块涉及到微信扫码等功能，因此还需要创建 **ConstantWxPropertiesUtils** 工具类和 **HttpClientUtils** 工具类。

首先用户进行登录操作，可以选择多种登录方式。登录请求首先进入到 **UserController** 控制层中，第一步判断用户是否注册过，对于已注册过的用户而言，直接登录成功；对于尚未注册的用户而言，那么请求会进入到 **UserService** 业务逻辑层中进行用户注册操作，并把注册信息保存到 **MySQL** 数据库中，最后返回用户登录成功信息。

首次注册成功的用户需要进行用户认证操作，身份认证请求先进入到 **UserController** 控制层中，再进入到 **UserService** 业务逻辑层中，最终将认证信息保存

到 MySQL 数据库中，其中的证件照片会被保存到阿里云 OSS 分布式文件系统中，并把相应的链接地址保存在 MySQL 数据库中。同样地，添加患者信息时也从 UserController 控制层到 UserService 业务逻辑层中，最终将患者信息保存到 MySQL 数据库中，最后返回操作成功信息给用户。用户管理模块 UML 时序图如图 4-9 所示。

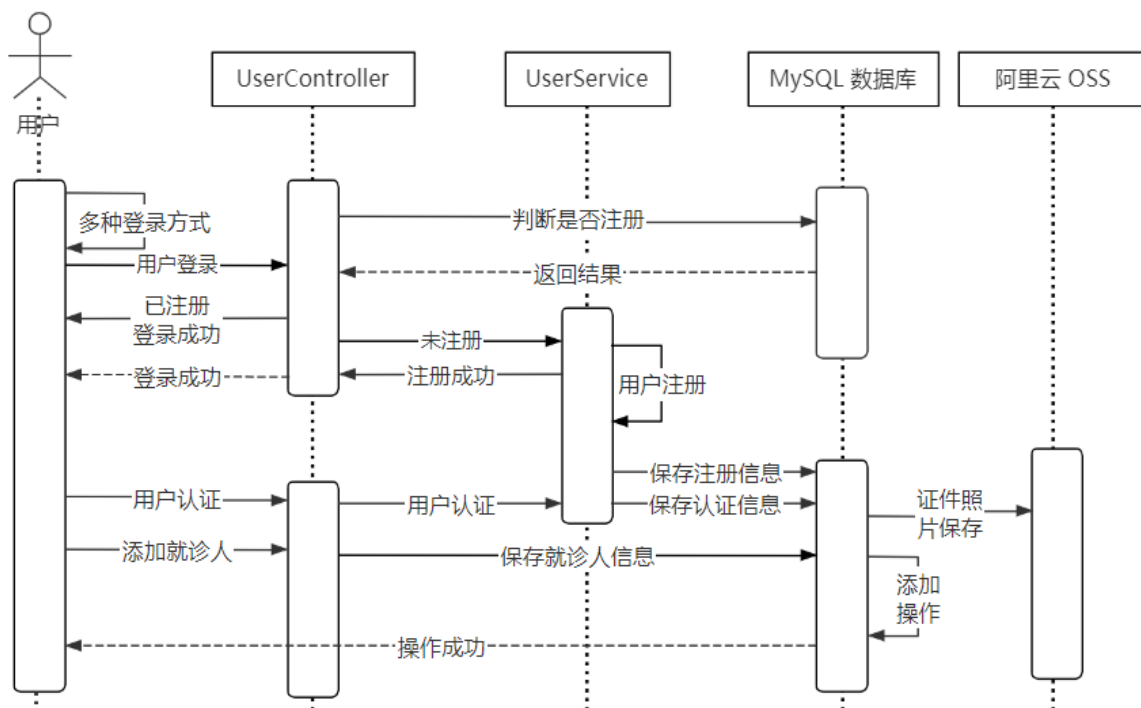


图 4-9 用户管理时序图

4.6 业务日志记录模块设计

业务日志记录模块的内容包括两个方面，分别是调用方法的信息和业务数据的记录。方法信息记录包括方法名、调用时间、形式参数。根据网上预约挂号系统的需要，本模块的设计需要遵循以下原则：一方面，本模块的执行过程不得影响系统中正常的业务流程运行；另一方面，系统中记录的所有日志信息都需要有一个唯一的标识符。

(1) 设计思想

根据系统业务日志的设计原则和系统的各种需求，本模块的设计采用面向切面的编程模型，将网上预约挂号系统分为两部分：首先是横切关注点，即需要加入功能

的位置；另一部分是切面，即需要加入的功能代码，需要使用表达式来定位目标业务方法。

(2) 业务日志功能设计

根据 Spring 框架中的 AOP 组件的使用场景和系统的具体需求，确定该功能模块使用 Spring 框架的 AOP 组件来设计完成。该模块的 UML 类图如图 4-10 所示。

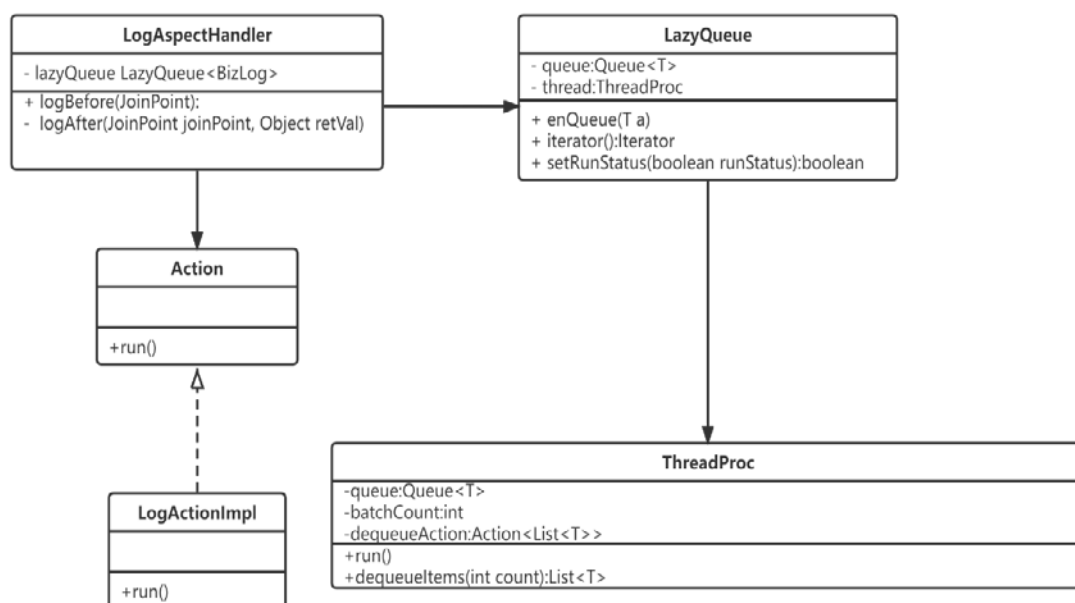


图 4-10 业务日志记录类图

在业务日志记录模块中，需要有一个 **LogAspectHandler** 类表示切面，其中定义了插入到业务代码中的逻辑，在这个系统中就是日志的输出。在该类中还需要有一个 **LazyQueue** 类型的成员作为属性，每一次的日志输出都保存到该队列，最终由单独的线程去执行数据库保存操作。

当系统发生业务事件时，例如某些业务方法的调用，那么系统即会发送记录日志的请求到 **LogController** 控制层中，在该层中首先会记录此次业务操作的各种操作参数，包括被调方法的方法名、方法调用的开始时间、结束时间以及方法的形式参数与返回值信息等。最后将这些信息保存到 **LazyQueue** 日志队列中，并返回保存成功信息。

在 **LazyQueue** 日志队列中保存日志信息后，紧接着系统会发送保存日志的请求

到 LogController 控制层中，并在控制层中启动一个独立的线程去操作 LazyQueue 日志队列中的日志信息，而不会影响正常的业务功能的运转。按照先进先出的原则依次保存到 MySQL 数据库中，最后返回操作成功信息到 LogController 控制层和系统中。业务日志记录模块的 UML 时序图如图 4-11 所示。

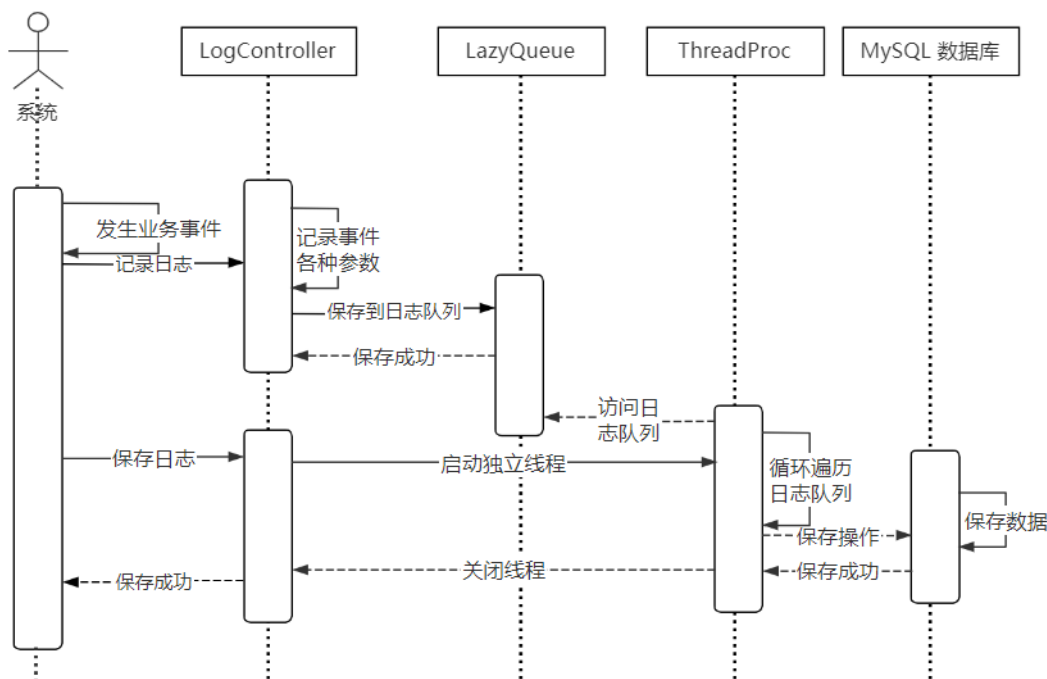


图 4-11 业务日志记录时序图

4.7 系统数据库设计

系统数据库的设计在系统开发周期中尤为重要，数据库设计的优秀与否在一定程度上不仅会影响日后系统的运行效果，还会影响该系统未来的扩展。数据库的设计不仅需要满足一定的范式，还需要将应包含的数据考虑周全，以及具有什么类型的字段。数据库的设计应该先设计出概念模型，即实体-关系图，紧接着再将其转换成数据表。

4.7.1 实体关系图

首先需要画出实体关系图，根据实际需求确定好系统具有哪些实体及其之间的内部联系，以及每个实体具有哪些属性，完成这些工作就方便后续创建对应的数据

表。由分析得知系统包括用户实体和患者实体，且它们为 1: n 的关系，即一个用户可以拥有多个患者，而一个患者从属于一个用户，且用户实体与患者实体都有姓名、身份证号码等属性。其次还有订单实体，用户实体与订单实体为 1: n 的关系，即一个用户可以提交多个订单，而一个订单属于一个用户，订单实体具有医院编号、科室编号以及排班编号等属性。订单实体与付款单实体为 1: 1 的关系，即一个订单付款生成一个付款单，一个付款单属于一个订单，且付款单具有付款类型、付款金额以及订单编号等属性。付款单实体与退款单实体为 1: 1 的关系，即一个付款单完成退款生成一个退款单，而一个退款单属于一个付款单，且退款单具有订单编号、退款状态等属性。最后还有医院、科室以及排班三个实体，与订单都是 1: n 的关系，即一个订单对应着一个医院、一个科室以及一个排班，反之一个医院、一个科室以及一个排班可以属于多个订单。排班实体与医生实体是 n: 1 的关系，即一个排班对应一个医生，一个医生对应多个排班。图 4-12 展示该系统的 E-R 图。

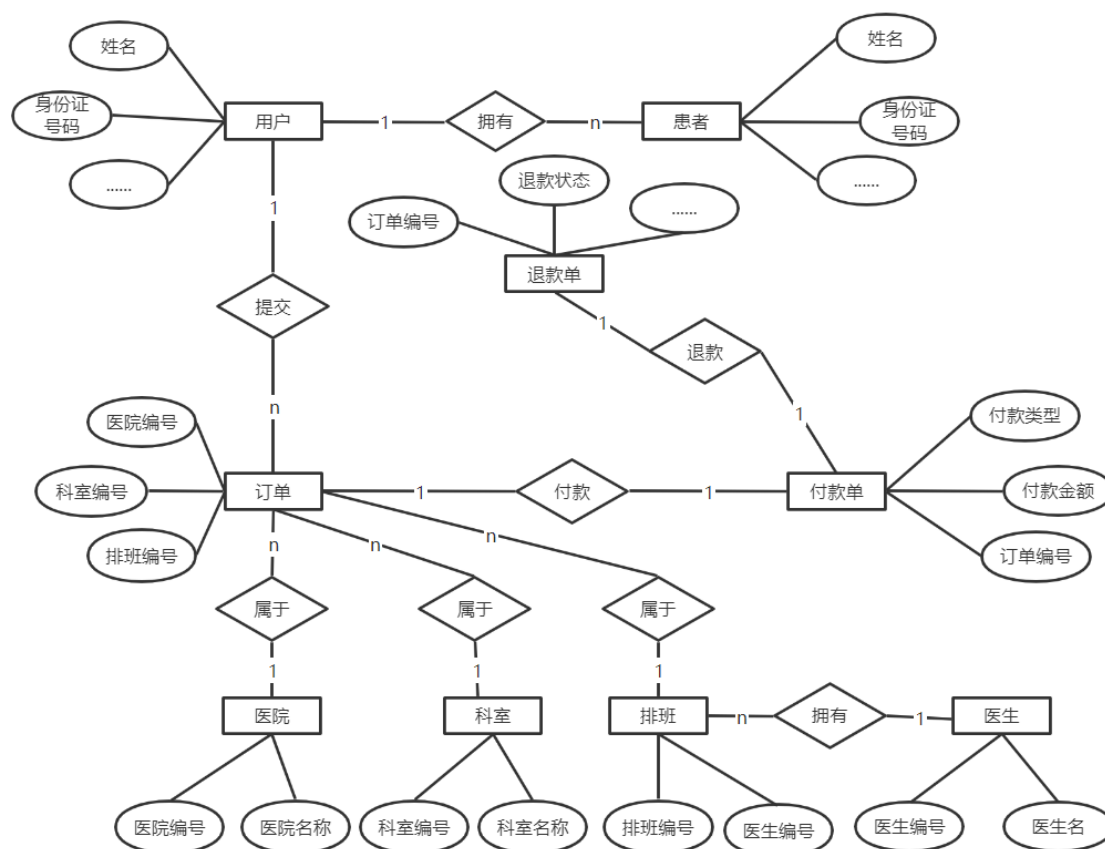


图 4-12 网上预约挂号系统 E-R 图

4.7.2 数据表设计

通过对 E-R 图的分析,得出该系统关联到的表有:用户表、患者表、订单表、支付信息表、退款信息表、医院信息表、科室信息表以及排班信息表。其中医院表、科室表以及排班表保存在 MongoDB 数据库中,其它的表数据则保存在 MySQL 数据库中,具体的数据库表的设计如下:

(1) 用户表

user_info 用户表保存的是在前台用户子系统中注册的用户信息,其中的字段包括用户编号、微信 openid、昵称、电话号码、用户姓名、证件类型、证件编号、证件路径、认证状态以及状态等,具体设计如表 4-1 所示。

表 4-1 用户表

编号	字段名	字段类型	是否可为空	注释
1	id	BIGINT(20)	否	用户编号
2	openid	VARCHAR(100)	是	微信 openid
3	nick_name	VARCHAR(20)	是	昵称
4	phone	VARCHAR(11)	否	手机号
5	name	VARCHAR(20)	是	用户姓名
6	certificates_type	VARCHAR(3)	是	证件类型
7	certificates_no	VARCHAR(30)	是	证件编号
8	certificates_url	VARCHAR(200)	是	证件路径
9	auth_status	TINYINT(3)	否	认证状态
10	status	TINYINT(3)	否	状态

(2) 患者表

每个已注册的用户可以添加多个患者信息,在后续提交订单时就可以选择某个患者,这些患者信息就保存在 patient 患者表中。其中的字段包括编号、用户 id、姓名、证件类型、证件编号、性别以及出生年月等,因为用户实体与患者实体之间为 1:n 的关系,因此在患者表中使用 user_id 属性作为外键,代表用户表中的用户 id,具体设计如表 4-2 所示。

表 4-2 患者表

编号	字段名	字段类型	是否可为空	注释
1	id	BIGINT(20)	否	编号
2	user_id	BIGINT(20)	是	用户 id
3	name	VARCHAR(20)	是	姓名
4	certificates_type	VARCHAR(3)	是	证件类型
5	certificates_no	VARCHAR(30)	是	证件编号
6	sex	VARCHAR(3)	是	性别
7	birthdate	DATE	是	出生年月

(3) 订单表

用户选择目标排班并确定患者后，会生成 order_info 订单表。订单表包括的字段有编号、用户 id、订单交易号、医院编号、医院名称等，因为用户实体与订单实体之间为 1: n 的关系，因此在订单表中定义 user_id 属性作为外键，代表用户表中的用户 id，具体设计如表 4-3 所示。

表 4-3 订单表

编号	字段名	字段类型	是否可为空	注释
1	id	BIGINT(20)	否	编号
2	user_id	BIGINT(20)	是	用户 id
3	out_trade_no	VARCHAR(300)	是	订单交易号
4	hoscode	VARCHAR(30)	是	医院编号
5	hosname	VARCHAR(100)	是	医院名称
6	depcode	VARCHAR(30)	是	科室编号
7	depname	VARCHAR(20)	是	科室名称
8	title	VARCHAR(20)	是	医生职称
9	hos_schedule_id	VARCHAR(50)	是	排班编号
10	reserve_date	DATE	是	安排日期
11	reserve_time	TINYINT(3)	否	安排时间
12	order_status	TINYINT(4)	是	状态

(4) 支付信息表

用户支付订单以后，会将信息保存到对应的 `payment_info` 支付信息表中。其中包括的字段有编号、对外业务编号、订单 `id`、支付类型、交易编号、支付金额以及支付状态等，因为订单实体与付款单实体之间是 1:1 的关系，因此在支付信息表中定义 `order_id` 属性作为外键，代表订单表中的订单 `id`，具体设计如表 4-4 所示。

表 4-4 支付信息表

编号	字段名	字段类型	是否可为空	注释
1	<code>id</code>	<code>INT(11)</code>	否	编号
2	<code>out_trade_no</code>	<code>VARCHAR(50)</code>	是	对外业务编号
3	<code>order_id</code>	<code>BIGINT(20)</code>	是	订单 <code>id</code>
4	<code>payment_type</code>	<code>TINYINT(3)</code>	是	支付类型
5	<code>trade_no</code>	<code>VARCHAR(50)</code>	是	交易编号
6	<code>total_amount</code>	<code>DECIMAL(10,2)</code>	是	支付金额
7	<code>payment_status</code>	<code>TINYINT(3)</code>	是	支付状态

(5) 退款信息表

已经完成支付的用户也可以随时取消预约，完成退款操作，相关的信息会保存到 `refund_info` 退款信息表中。其中包括的字段有编号、对外业务编号、订单编号、支付类型、交易编号、退款金额以及退款状态等，具体设计如表 4-5 所示。

表 4-5 退款信息表

编号	字段名	字段类型	是否可为空	注释
1	<code>id</code>	<code>INT(11)</code>	否	编号
2	<code>out_trade_no</code>	<code>VARCHAR(50)</code>	是	对外业务编号
3	<code>order_id</code>	<code>BIGINT(20)</code>	是	订单编号
4	<code>payment_type</code>	<code>TINYINT(3)</code>	是	支付类型 (
5	<code>trade_no</code>	<code>VARCHAR(50)</code>	是	交易编号
6	<code>refund_status</code>	<code>TINYINT(3)</code>	是	退款状态

(6) 医院信息表

各个医院的信息保存在 `Hospital` 医院信息表中，该表保存在 `MongoDB` 数据库中，包括的字段有医院编号、医院名称、省编号、市编号、区编号以及医院介绍等。

医院信息表中的数据由系统管理员在后台管理子系统中进行添加，具体数据则由具体医院提供。具体设计如表 4-6 所示。

表 4-6 医院信息表

编号	字段名	字段类型	是否可为空	注释
1	hoscode	String	否	医院编号
2	hosname	String	否	医院名称
3	provinceCode	String	否	省编号
4	cityCode	String	否	市编号
5	districtCode	String	否	区编号
6	intro	String	是	医院介绍

(7) 科室信息表

各个医院的科室信息保存在 Department 科室信息表中，该表保存在 MongoDB 数据库中，包括的字段有医院编号、科室编号、科室名称、科室介绍、大科室编号以及大科室名称等。科室信息表中的数据由系统管理员在后台管理子系统中进行添加，具体数据则由具体医院提供。具体设计如表 4-7 所示。

表 4-7 科室信息表

编号	字段名	字段类型	是否可为空	注释
1	hoscode	String	否	医院编号
2	depcode	String	否	科室编号
3	depname	String	否	科室名称
4	intro	String	是	科室介绍
5	bigcode	String	否	大科室编号
6	bigname	String	否	大科室名称

(8) 排班信息表

各个科室的排班信息保存在 Schedule 排班信息表中，该表保存在 MongoDB 数据库中，包括的字段有医院编号、科室编号、排班编号、职务、医生编号以及擅长领域等。排班信息表中的数据由系统管理员在后台管理子系统中进行添加，具体数据则由具体医院提供。具体设计如表 4-8 所示。

表 4-8 排班信息表

编号	字段名	字段类型	是否可为空	注释
1	hoscode	String	否	医院编号
2	depcode	String	否	科室编号
3	hosScheduleId	String	否	排班编号
4	docid	String	否	医生编号

（9）数据字典表

数据字典表用于保存一些固定的数据信息，比如行政区划信息、民族信息以及学历信息等，其中包括的字段有 id、上级 id、名称、值以及编码等。行政区划信息由全国行政区划信息查询平台提供，民族信息由中国民族信息网提供，学历信息由中国高等教育学生信息网提供，系统管理员可在后台管理子系统中通过 Excel 表格的形式对这些数据进行导入导出操作。具体设计如表 4-9 所示。

表 4-9 数据字典表

编号	字段名	字段类型	是否可为空	注释
1	id	BIGINT(20)	否	id
2	parent_id	BIGINT(20)	否	上级 id
3	name	VARCHAR(100)	否	名称
4	value	BIGINT(20)	是	值
5	dict_code	VARCHAR(20)	是	编码

（10）医生信息表

医生信息表保存医生信息，主要包括了 id、name、title 以及 skill 字段。具体设计如表 4-10 所示。

表 4-10 医生信息表

编号	字段名	字段类型	是否可为空	注释
1	id	BIGINT(20)	否	医生编号
2	name	VARCHAR(100)	否	医生名称
3	title	VARCHAR(100)	否	职称
4	skill	VARCHAR(20)	是	擅长领域

4.8 本章小结

本章对系统的设计方案做了详细的阐述。首先阐述网上预约挂号系统各个功能模块的设计方案；其次进行系统数据库的设计，根据需求画出 E-R 图，再根据 E-R 图设计出具体的表，并给出表的字段、字段类型和详细的数据类型约束。

5 网上预约挂号系统的实现

5.1 系统开发环境

网上预约挂号系统的前端使用 Vue 框架，后端使用 Java 8 开发。Java 作为一门有着许多优势的高级语言，官方提供了大量基础类库供开发者使用。系统具体环境配置如下：

- (1) 开发语言：Java 8;
- (2) 项目构建工具：Maven 3.3.6;
- (3) 集成开发环境：IntelliJ IDEA 2020.3;
- (4) 数据库环境：Redis 5.0.8、MySQL 5.6、MongoDB 4.4.1。

5.2 数据字典功能模块实现

登录后台管理子系统后，在主界面可以看到数据管理模块，包括数据字典子模块。数据管理模块的主要功能就是显示一些固定数据，包括行政区划信息、医院等级信息、证件类型信息、学历信息和民族信息。该界面左上方也提供了以 Excel 表格形式的数据导入导出功能。数据管理模块对应 yygh_cmn 数据库中的 dict 数据表，包括 id、parent_id、name、value 和 dict_code 字段，分别表示 id、父级 id、名称、值以及字典编码。

基于数据字典的表结构，后端实现中提供了关于数据字典相关的操作功能。包括根据 value 值的查询功能，可以根据 value 的值查询出对应的 name 值；根据 dictCode 和 value 值的联合查询功能，可以根据字典编码和 value 值联合查询出对应的 name 值；根据数据 id 值查询子数据列表的功能，可以根据 id 值查询出所有下级列表；根据 dictCode 值获取下级节点的功能，可以根据字典编码的值查询出所有下级列表；以及以 Excel 表格为媒介导入数据字典和导出数据字典的功能。

根据 value 的值查询出对应的 name 值，数据字典中每条数据的 value 值都不相同，因此就可以获取到对应的 name 的值。此外，还提供根据字典编码和 value 值联

合查询 name 的值。前端发送 http 请求到后端，请求参数为 value 值或字典编码和 value 值，后端通过 @GetMapping 和 @PathVariable 注解接收请求参数，并注入到对应的形式参数中，再通过访问 MySQL 数据库获取结果并返回。图 5-1 为后台管理员数据字典操作界面。

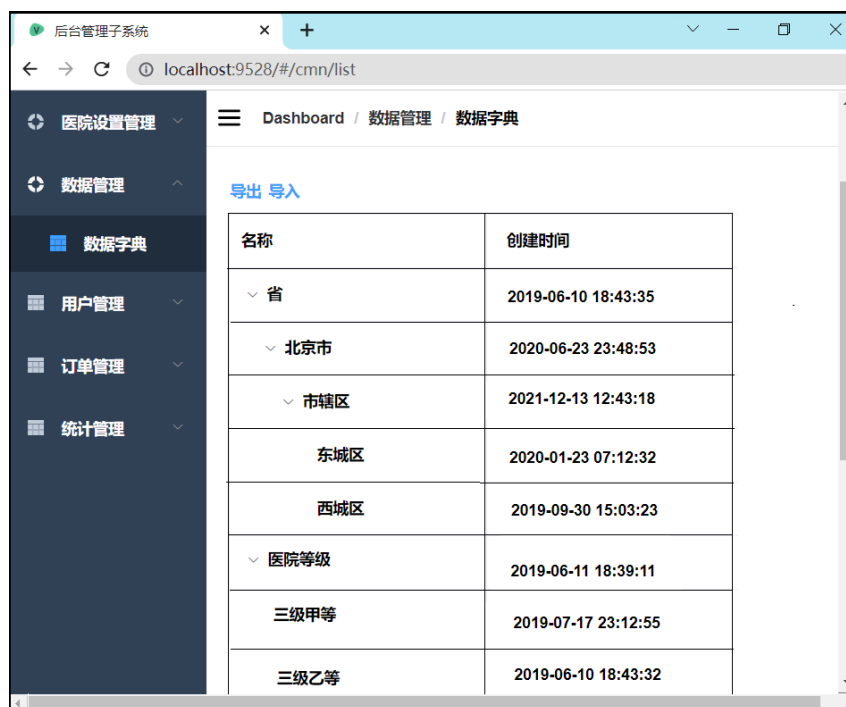


图 5-1 后台管理员数据字典操作界面

在该页面点击某个上级节点，会显示出该父节点的所有下级节点。该功能是根据父类 id 查询子数据列表，以及根据字典编码获取子数据列表实现。当点击某个上级节点，前端需要将父类 id 或者父类字典编码作为请求参数传递给后端，后端通过 @GetMapping 和 @PathVariable 注解接收请求参数，并注入到对应的形式参数中，再通过访问 MySQL 数据库获取结果并返回。

在数据管理模块页面的左上方有导入导出按钮，可将数据字典信息导出至 Excel 表格中，也可将 Excel 表格中的数据导入到数据字典中，前提是 Excel 表格中的数据必须满足特定的格式。该功能使用到阿里提供的 EasyExcel 技术，首先需在 pom.xml 配置文件中添加依赖，其次通过 EasyExcel 工具类提供的 read 方法和 write 方法来导入和导出数据字典数据。

数据字典模块应用在后台管理子系统中的医院列表模块。系统管理员可以通过

下拉菜单选择城市、地区和医院等级，查询出指定条件的医院，这些医院信息在医院在医院设置添加模块中被添加。当点击查询按钮，系统发送 http 请求到后端，并携带 provinceCode、cityCode 和 hosname 参数，分别代表省编码、城市编号和医院编号。后端将这些参数封装成 HospitalQueryVo 实体类的对象，并通过查询 MongoDB 数据库将医院信息返回。图 5-2 为后台管理员医院操作界面。



图 5-2 后台管理员医院操作界面

数据字典除了应用在后台管理子系统中，也应用在前台用户子系统主界面根据医院等级与地址筛选医院中。此处可根据医院等级来筛选目标医院，例如三级甲等、三级乙等、二级甲等或其它等级，也可根据医院地区来筛选医院，比如东城区、西城区或其它地区。为实现这种功能，后端定义 HospitalQueryVo 实体类，在该类中定义 hostype 属性表示医院类型，provideCode 属性、cityCode 属性和 districtCode 属性表示省编号、市编号和区编号。当浏览器通过医院等级和地区筛选医院时会向后端发起 http 请求，请求参数为医院等级和地区信息，后端将该请求参数封装成 HospitalQueryVo 实体类的对象，再使用该对象去 MongoDB 数据库中查询数据并返回。

5.3 订单管理功能模块实现

该功能模块的实现较为复杂，当用户选择目标医院、目标科室和目标排班后，进

入患者选择界面,选择目标患者并点击提交订单就会生成订单。选择目标医院、目标科室和目标排班实现在 service_hosp 微服务中,而订单的生成、订单支付、取消预约实现在 service_order 微服务中,关于选择患者信息实现在 service_user 微服务中。把这些不同功能实现在不同微服务中,可以极大程度提高系统运行效率;并且将功能模块进行划分,便于系统开发人员开发和理解。

在前台用户子系统主界面用户可以在搜索框中输入医院关键字,也可以在等级和地区栏中来筛选医院,最终符合筛选条件的医院都会以列表的方式在页面下方呈现给用户。当点击目标医院后,前端会将医院编号 hoscode 作为请求参数传递到后端中 HospApiController 控制层的 index 处理器,并通过注解的方式赋值给 hoscode 属性。后端获取到医院编号后,首先创建 List 集合保存 DepartmentVo 实体类的对象,其次根据医院编号从 MongoDB 数据库中查找到对应的科室数据保存至 List 集合中,最后根据大科室编号分组,获取每个大科室里的下级子科室,并把 List 集合返回给浏览器呈现给用户。图 5-3 为前台患者科室选择界面。



图 5-3 前台患者科室选择界面

排班选择界面的实现较为复杂,涉及到日期分类和排序操作。进入到该页面,已经确定医院和科室,即确定 hoscode 值和 deocode 值。在科室选择界面用户选择目标

科室后,前端发送 http 请求到后端,请求参数包括 hoscode 值和 deocode 值,后端使用注解的方式获取这两个参数,再去 MongoDB 数据库中查询排班信息,最后把这些结果返回到浏览器并呈现给用户。前台患者排班选择界面如图 5-4 所示。



图 5-4 前台患者排班选择界面

进入排班选择界面后,患者即可选择目标日期来查看号源信息,该界面是患者根据自身需求来选择合适的排班并进行下单操作。当点击目标日期,前端将医院编号 hoscode、科室编号 deocode 和日期 workDate 作为请求参数传递给后端。后端通过 @GetMapping 和 @PathVariable 注解将它们赋值给 hoscode、deocode 和 workDate 三个形式参数,并根据这三个参数访问 MongoDB 数据库获取对应的号源信息返回给浏览器并呈现给用户。选择目标日期后,页面就会显示当前页面的号源信息,包括医生名字、医生职称、剩余号数和价钱等信息。

当患者选择某个排班,即跳转到添加患者页面,需要选择目标患者信息。选择成功并提交订单,前端将排班编号 scheduleId 与患者编号 patientId 作为请求参数发送给后端。后端通过 @PostMapping 和 @PathVariable 注解的方式接收到参数并注入到 scheduleId 和 patientId 两个形式参数中,并通过调用业务逻辑层去生成订单,即把订单信息保存到 yygh_order 数据库中的 order_info 数据表中。

当用户选择患者并点击确认挂号,业务逻辑层首先通过患者编号 patientId 从

yygh 数据库中的 patient 数据表中获取到患者信息,并将结果封装成 Patient 实体类的对象。同时根据排班编号 scheduleId 从 MongoDB 数据库中获取排班信息,并将结果封装成 ScheduleOrderVo 实体类的对象。然后会判断当前预约时间是否可预约,如果不预约则直接返回;如果可预约则会创建 OrderInfo 实体类的对象,并将 Patient 对象和 ScheduleOrderVo 对象中与订单相关的信息封装到其中,最后使用 MyBatis-Plus 框架中提供的方法将封装好的 OrderInfo 实体类的对象中的数据保存到 yygh_order 数据表中的 order_info 数据表中,实现订单的生成并保存的操作。前台患者选择界面如图 5-5 所示。

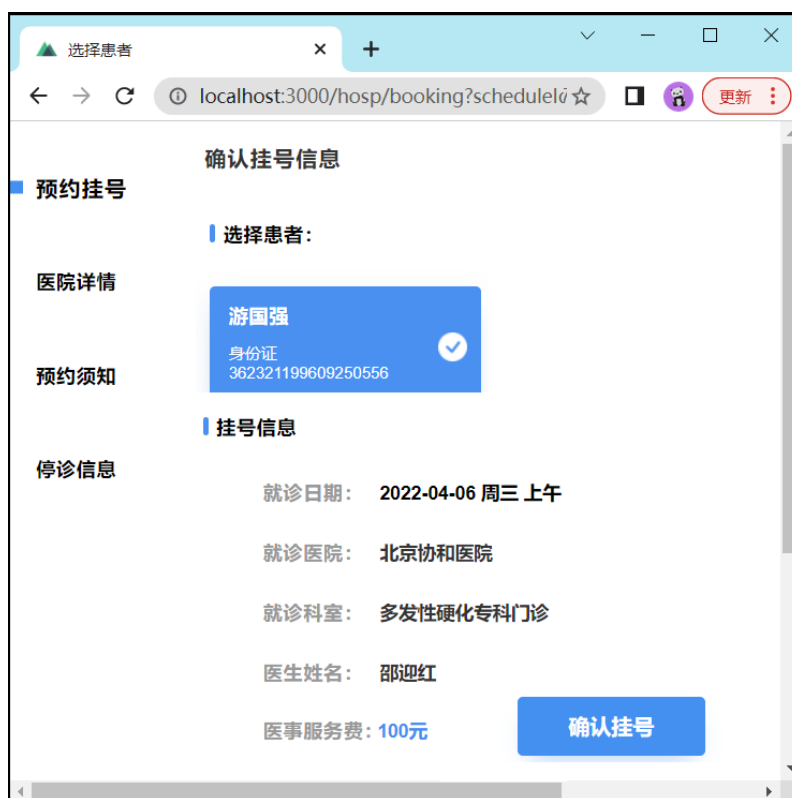


图 5-5 前台患者选择界面

订单生成之后可随时取消。当用户选择取消订单,前端将订单编号作为请求参数发送给后端,后端采用 @GetMapping 和 @PathVariable 注解的方式接收订单编号,即 Restful 风格的请求路径,并注入到形式参数中。后端通过订单编号从 MySQL 数据库中获取到订单,并封装成 OrderInfo 实体类的对象,判断该订单是否可取消。若不可取消则直接返回;若可取消则更新订单的状态为已取消状态。此外,在取消订单时,

还应判断当前订单是否已付款，如果已付款还需要进行退款，且退款按原路返回；如果未付款则什么也不需要做。前台订单支付界面如图 5-6 所示。



图 5-6 前台订单支付界面

用户点击支付订单，前端会将订单编号 `orderId` 作为请求参数传递到后端。在后端，首先判断该订单编号在 Redis 数据库中是否存在。如果存在则直接返回对应的 `value` 值（微信二维码，用于扫码支付使用）；如果不存在则调用微信支付相关的 API 生成支付二维码，并与订单编号作为 `value` 和 `key` 保存在 Redis 数据库，且设置一分钟的有效期；最后将支付二维码返回给浏览器并呈现给用户。付款成功会将支付记录保存到 `yygh_order` 数据库的 `payment_info` 数据表。之后用户如果取消预约，会将退款记录保存到 `yygh_order` 数据库的 `refund_info` 数据表，并更新订单表的状态和付款表的状态，且实现退款操作。

5.4 短信服务管理功能模块实现

短信服务管理功能模块使用在使用手机验证码登录，新用户在使用手机验证码

方式登录会自动注册。此外，定时模块中也会在每天固定时间点给患者们送短信提醒。在进入系统后，如果用户没有进行登录就进行预约挂号操作，即会跳转到登录界面，且支持多种登录方式，以手机验证码为主。即使使用微信扫码登录，在后续也需绑定手机号码，方便用户与账号之间的绑定。图 5-7 为前台手机登录界面。

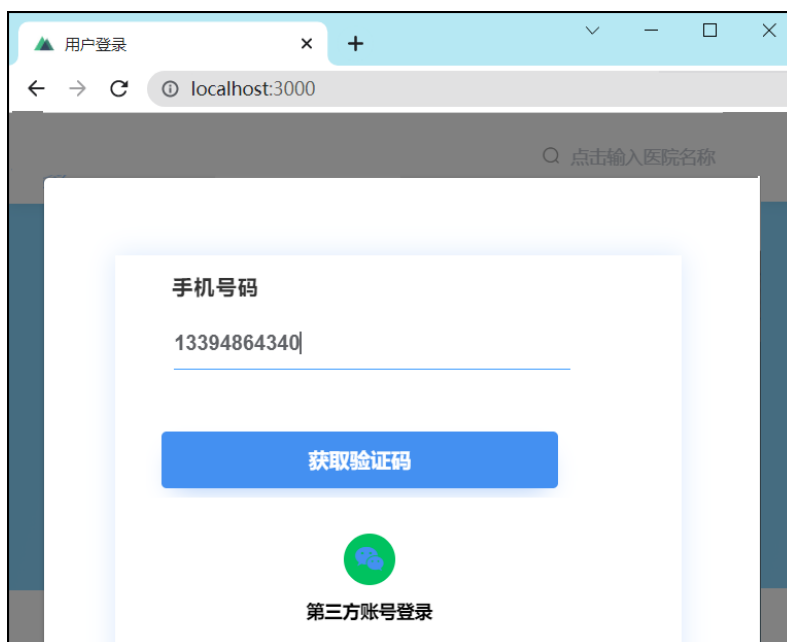


图 5-7 前台手机登录界面

短信相关的功能实现在 `service-msm` 微服务，发送短信功能对应 `public Result sendCode(@PathVariable String phone)` 方法。用户输入号码并点击获取验证码，前端会发送 `http` 请求到后端，请求参数为“`phone=手机号码`”，请求匹配到 `sendCode` 处理器，`phone` 参数通过 `@PathVariable` 注解会获取到参数值。后端首先去查看 `Redis` 数据库中是否保存了该号码，如果已保存表示已发送验证码给该号码且验证码未过期，直接返回即可。

如果 `Redis` 数据库没有保存该号码，则通过 `RandomUtil` 工具类生成六位数随机字符作为验证码，把手机号码与验证码作为 `key` 和 `value` 保存到 `Redis` 数据库中，并向该手机号码发送验证码短信，同时设置 `Redis` 数据库中该键值对的有效期为一分钟。

用户输入手机验证码并点击登录后，后端判断手机号是否在 `Redis` 数据库中保存，如果没有保存则返回验证码错误；如果已保存则匹配对应的验证码，匹配成功则

登录成功,进入系统主界面,否则返回验证码错误。其中主界面中的行政区划信息由全国行政区划信息查询平台提供,系统管理员可在后台管理子系统中通过 Excel 表格的形式对这些数据进行导入导出操作,登录成功后进入前台主界面如图 5-8 所示,在主界面可通过选择等级、地区信息来筛选目标医院,并通过点击目标医院可进入科室选择界面。



图 5-8 前台主界面

短信服务也使用在定时模块。定时模块的实现在 service 模块中的 service_task 子模块中,该模块在每日早上八点定时向当天就诊患者发送短信提醒。service_task 模块通过@EnableScheduling、@Scheduled 注解和 cron 表达式来实现定时功能。通过 cron 表达式可以实现每日早上八点定时执行目标方法。在目标方法的实现中,遍历所有未取消预约的订单,如果订单的工作日期是当天,则向患者发送短信提醒。

5.5 用户管理功能模块实现

该功能模块与就医用户相关。该模块实现的好坏会直接影响用户的体验,因此这一模块的实现也是至关重要,包括多个子模块。

用户登录与注册子模块实现在 service-user 微服务。用户输入手机号并填写验证码后提交,将 http 请求发送到后端,请求参数包括手机号和验证码。请求会匹配到 UserInfoApiController 控制层的 login 处理器,并通过 @PostMapping 和 @RequestBody

注解将请求参数注入到 LoginVo 实体类的对象，该对象的 phone 属性表示手机号，code 属性表示验证码。

后端首先从 LoginVo 实体类的对象提取出手机号与验证码，且需保证它们都是非空，若为空则作抛出异常处理。其次，需要去判断 Redis 数据库保存的验证码与传入的验证码是否一致，如果不一致抛异常处理。最后需要判断登录的用户是否已注册；如果未注册，则完成注册并登录成功；如果已注册直接登录成功。注册的用户被添加到 yygh_user 数据库的 user_info 数据表。

用户注册后，身份认证信息尚未完整，该用户无法进行预约挂号。此时需要完成身份认证，需要填写用户姓名、证件类型、证件号码以及证件照片等信息。当用户输入完这些信息点击提交，前端发送 http 请求给后端，请求参数包括上述填写的信息。后端则通过 @PostMapping 和 @RequestBody 注解的方式将请求参数封装成 UserAuthVo 实体类的对象，其中的 name 和 certificatesType 属性分别代表用户的名字和证件类型，certificatesNo 属性表示证件号码，certificatesUrl 属性表示证件路径，该证件路径代表阿里云 OSS 分布式文件系统的路径。图 5-9 所示为前台实名认证界面。

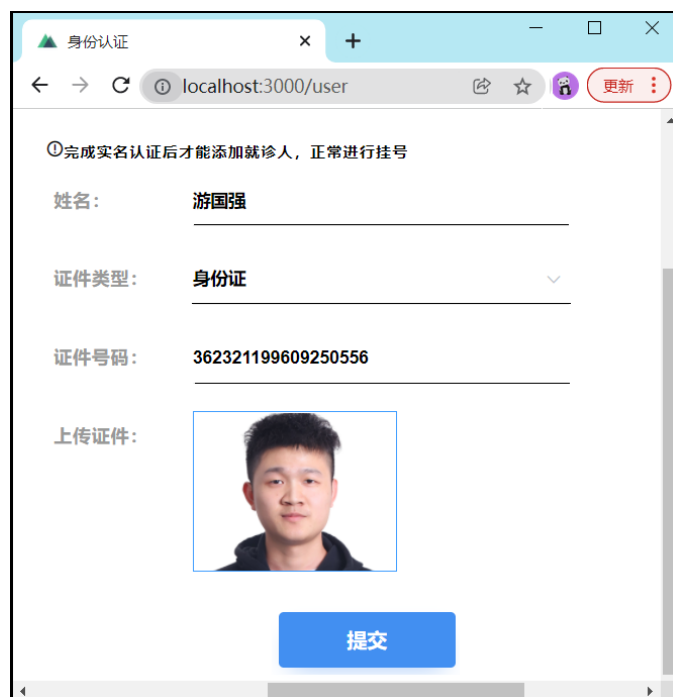


图 5-9 前台实名认证界面

最后一个子模块就是添加患者模块。用户在预约挂号之前，首先就需要添加患

者,在生成订单时可以直接选择已添加的患者。在患者管理界面中添加患者信息,需要输入患者姓名、证件类型、证件号码以及性别、民族等信息,点击保存后,前端发送的 http 请求会匹配到 PatientApiController 控制层的 savePatient 处理器,且将请求参数注入到 Patient 实体类的实例中,该类中的 userId 属性表示用户 id, name 属性表示用户姓名。将 Patient 实体类的对象封装好之后,调用 MyBatis-Plus 框架提供的方法保存到 yygh_user 数据库的 patient 数据表中。

5.6 业务日志记录模块实现

(1) 在 Spring 配置文件中配置<aop:aspectj-autoproxy>标签,用于生成动态代理实现类对象,因为 AOP 底层采用动态代理来实现。还需要配置三个 bean 标签,分别为 LogAspectHandler、LogActionImpl 和 LazyQueue 类,其中 LazyQueue 类通过 constructor-arg 标签自动调用带参构造器来创建对象,最后还配置了 logBefore 方法和 logAfter 方法表示业务方法调用前后运行其它代码的逻辑。

(2) 日志记录实现逻辑

首先定义配置文件中配置的三个类,并实现 logBefore 和 logAfter 方法,最后使用 @Aspect 注解定义切面,并且所有的 bean 都需要被 Spring IOC 容器来管理,否则 AOP 功能不会生效。采用切入点表达式的方式来将切面切入至横切关注点中,使得业务方法调用的时候自动调用 logAfter 和 logBefore 方法。在这两个方法中需要将调用的方法信息记录到日志队列中,最后创建一个单独线程记录至数据库。

5.7 本章小节

本章首先介绍网上预约挂号系统的开发环境,包括系统开发语言、集合开发环境和相关的数据库。其次介绍该系统各个功能模块的实现情况,主要涉及数据字典模块、订单管理模块、短信服务管理模块、用户管理模块和业务日志记录模块的具体实现情况。

6 网上预约挂号系统的测试

本章对网上预约挂号系统进行功能测试和性能测试。测试工作可以发现系统中隐藏的问题并确定缺陷存在的位置,然后对其进行修改,保障系统的功能和性能能够满足用户的需求。

6.1 系统的测试环境

测试工作在搭建好的测试环境下进行,包括软件环境和硬件环境,软件环境包括操作系统与应用软件,硬件环境包括计算机型号和 CPU/内存。表 6-1 为测试环境具体的配置信息。

表 6-1 测试环境配置表

测试工具	操作系统	应用软件	型号	CPU/内存
浏览器	Windows 10	IE/Firefox/Chrome	Acer TMP 259	i5-6200U 2.3Ghz/16G
Web 服务器	Windows 10	Tomcat 8.1	Acer TMP 259	i5-6200U 2.3Ghz/16G
数据库软件	Windows 10	MySQL 5.6 Redis 5.0.8 MongoDB 4.4.1	Acer TMP 259	i5-6200U 2.3Ghz/16G
压测工具	Windows 10	JMeter 5.4.3	Acer TMP 259	i5-6200U 2.3Ghz/16G

6.2 功能测试

(1) 数据字典管理模块测试

在后台管理子系统中,需要在数据管理模块展示行政区划信息、学历信息、医院等级信息和民族信息,还需要在其它多个模块能够根据行政区划信息或医院等级信息来筛选医院列表。在前台用户子系统中,主界面需要根据行政区划信息和医院等级信息筛选医院,且在身份认证和添加患者时,也需要填写民族信息。表 6-2 提供了数据字典管理模块的部分测试用例。

表 6-2 数据字典管理模块功能测试用例

前提步骤	执行详情	预期结果	最终情况
系统管理员登录	点击数据管理以及各个节点	页面显示行政区划信息、民族信息、学历信息以及医院等级信息等，还可以显示下级节点	通过
系统管理员登录	在医院设置中选择医院等级筛选医院	页面会根据选择的医院等级显示各个等级的医院	通过
普通用户登录	在主界面点击医院等级和行政区划选项	页面下方会根据所点击的选型显示相应的医院	通过
普通用户登录	填写身份认证信息，选择行政区划信息和民族信息	会显示多级菜单供选择省市区，以及下拉框选择民族信息	通过
普通用户登录	添加患者时选择省市区信息和民族信息	会显示多级菜单供选择省市区，以及下拉框选择民族信息	通过

（2）订单管理功能模块测试

这部分包括前台用户子系统和后台管理子系统的操作。首先在后台管理子系统中系统管理员能够查询到订单。其次在前台用户子系统中，患者选择目标医院，页面会显示所有科室信息；然后选择目标科室，页面会显示该科室下所有的排班信息；紧接着选择日期，页面会显示该日期中上午和下午的号源信息；最后选择目标排班，页面会让用户选择患者。提交订单后就会生成订单，此时我们可以选择取消预约和支付订单；当支付订单后也可以取消预约并退款。表 6-3 提供了订单管理模块的部分测试用例。

表 6-3 订单管理模块功能测试用例

前提步骤	执行详情	预期结果	最终情况
系统管理员登录	点击订单管理	页面显示已经提交的订单，包括取消预约用折线图的方式显示	通过
系统管理员登录	点击统计管理	订单随日期的变动情况	通过
普通用户登录	点击目标号源	页面显示添加患者，以及价格信息等	通过

续表 6-3 订单管理模块功能测试用例

前提步骤	执行详情	预期结果	最终情况
普通用户登录	添加患者，并提交订单	页面显示订单提交成功，此时可以取消预约和支付订单	通过
普通用户登录	点击支付订单	页面进入支付界面，支持支付宝扫码支付、微信扫码支付以及银行卡支付	通过
普通用户登录	扫码支付	提示支付成功，进入支付成功页面，该页面可以取消预约	通过
普通用户登录	点击取消预约	提示取消预约成功，并完成退款操作	通过

(3) 短信服务管理模块测试

该模块的实现在前台用户子系统的登录与注册中，支持短信验证码登录，而且第一次登录的用户会完成自动注册。表 6-4 提供了短信服务管理模块的部分测试用例。

表 6-4 短信服务管理模块功能测试用例

前提步骤	执行详情	预期结果	最终情况
普通用户登录	输入手机号码，点击发送验证码	手机收到验证码，有效期为一分钟	通过
普通用户登录	输入错误验证码，点击登录	提示验证码错误	通过
普通用户登录	待有效期过后输入验证码	提示验证码不存在	通过
普通用户登录	重复发送验证码	提示重复发送，请稍后重试	通过
普通用户登录	输入正确验证码，点击登录	登录成功，第一次登录自动注册	通过

(4) 用户管理功能模块测试

该部分是与用户相关的操作，也包括前台用户子系统和后台管理子系统两部分。在后台管理子系统中，系统管理员可以在用户管理模块查看到已注册的用户列表和认证列表，认证列表保存的是已上传认证信息但尚未通过认证的用户。在前台用户子系统中，可以实现登录注册、身份认证和添加患者的功能。表 6-5 提供了用户管理功能模块的部分测试用例。

表 6-5 用户管理模块功能测试用例

前提步骤	执行详情	预期结果	最终情况
系统管理员登录	点击用户管理中的用户列表	页面显示已经注册的用户，包括已认证与未认证	通过
系统管理员登录	点击用户管理中的认证列表	页面显示提交认证信息的用户	通过
普通用户登录	用户输入手机号获取验证码进行登录注册	手机收到验证码，登录成功	通过
普通用户登录	输入身份认证信息，点击提交	提交功能，后台管理系统中的认证列表会显示该用户	通过
普通用户登录	输入患者信息，点击保存	患者添加成功，在患者列表中可查看	通过

6.3 非功能测试

功能性测试满足需求是系统可以投入市场使用的前提条件。这节主要通过 Apache 提供的 JMeter 压测工具模拟在高并发场景下对系统的性能进行测试，测试结果可以通过 Apache JMeter 压测工具提供的聚合报告与图形结果来展示，测试的主要接口包括用户登录与生成订单。使用 Apache JMeter 压测工具模拟并发数为 3000 的情况下系统的响应情况，首先测试用户登录接口，测试后得到的聚合报告如图 6-1 所示。

Summary Report

Name:Summary Report

Comments:聚合报告

Write results to file / Read from file

Filename

Label	# Samples	Average	Min	Max
HTTP 请求	3000	506	502	611
TOTAL	3000	506	502	611

图 6-1 用户登录接口测试聚合报告

从 Apache JMeter 压测工具提供的聚合报告中能够看出在并发数为 3000 的情况

下，所有请求的平均请求延迟为 506 毫秒，最小响应时间为 502 毫秒，最大响应时间为 611 毫秒，满足需求分析中要求的预期目标。Apache JMeter 压测工具除了可以提供聚合报告形式的数据展示测试结果，还可通过图形结果的形式来查看，图形结果展示的数据在聚合报告的基础上增加了更多测试信息。图 6-2 为用户登录接口测试的图形结果。

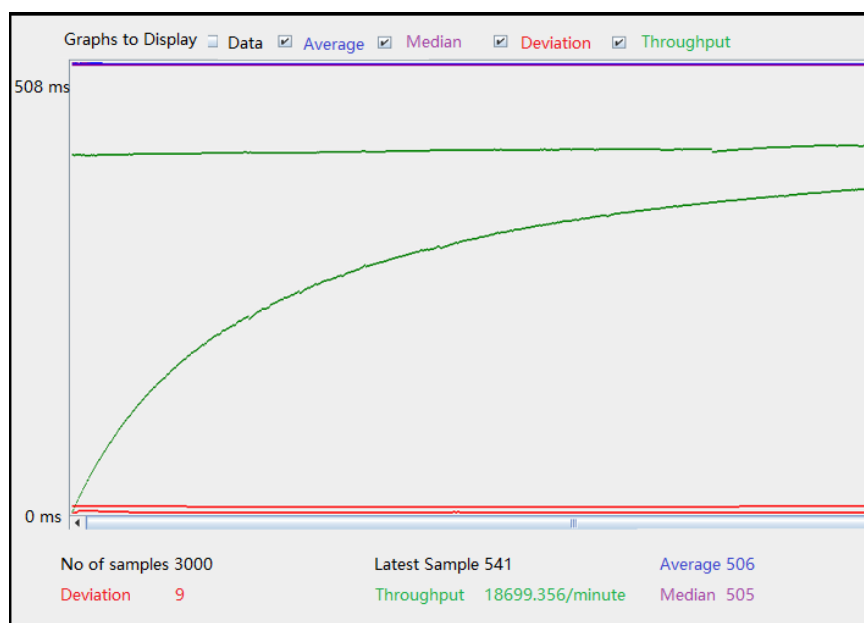


图 6-2 用户登录接口测试图形结果

从 Apache JMeter 压测工具得到的图形结果可以看出在最大并发数为 3000 的情况下，服务器响应最后一个用户登录接口请求的时间为 541 毫秒，并且服务器每分钟可以处理的请求数高达 18699 个，每一次请求的平均响应时间为 506 毫秒，图形结果展示的情况与上述聚合报告展示的情况一致。请求响应时间的中间值为 505 毫秒，即测试的请求中有一半请求的响应时间低于 505 毫秒，有一半请求的响应时间高于 505 秒。

通过 Apache JMeter 压测工具的测试，得知用户登录接口的性能要求满足需求分析中的预期目标。接下来测试的是生成订单接口，同样模拟的是在并发数为 3000 的情况下该接口的性能，在 Apache JMeter 中配置接口请求参数并测试得到的聚合报告如图 6-3 所示。

Summary Report

Name:

Summary Report

Comments:

聚合报告

Write results to file / Read from file

Filename

Label	# Samples	Average	Min	Max
HTTP 请求	3000	1005	1002	1142
TOTAL	3000	1005	1002	1142

图 6-3 生成订单接口测试聚合报告

从 Apache JMeter 压测工具提供的聚合报告中得到在并发量为 3000 的情况下，所有请求的平均请求延迟为 1005 毫秒，比登录接口的响应时间稍大一些。最小响应时间与最大响应时间分别为 1002 毫秒和 1142 毫秒，满足需求分析中要求的预期目标。

但也可以从该聚合报告中看出生成订单的接口请求的响应时间相较于其它接口请求的响应时间较长，导致这种情况的主要原因可能是由于订单数量的增长，导致数据表记录数的增大，最终导致生成订单响应时间的延长，可以通过改变数据库结构等方式来得到缓解。

Apache JMeter 工具除了提供聚合报告展示测试结果的方式，JMeter 工具还提供图形结果以供展示测试结果，在结果图中使用不同颜色的曲线代表不同的性能指标，结果图底端也统计了部分结果信息。

从 Apache JMeter 压测工具提供的图形结果也同样可以看出在并发数为 3000 的情况下，服务器响应最后一个请求的时间为 1007 毫秒，服务器每分钟可以处理的请求数为 5732，每一次请求的平均响应时间为 1005 毫秒，与上述聚合报告的结果大致吻合。请求响应时间的中间值为 1004 毫秒，即有一半请求的响应时间低于 1004 毫秒，有一半请求的响应时间高于 1004 毫秒。通过 Apache JMeter 压测工具的测试，得知订单生成接口的性能要求满足需求分析中的预期。图 6-4 为生成订单接口测试的图形结果。

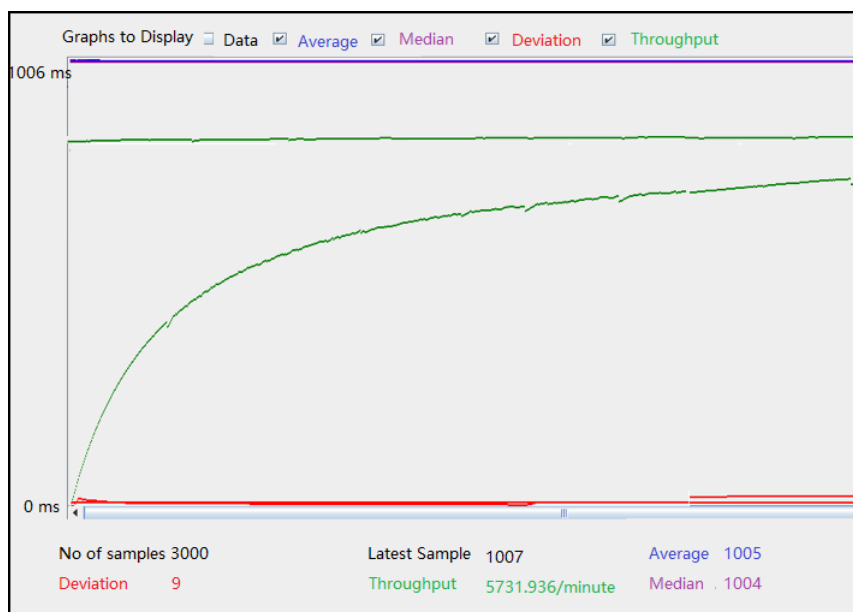


图 6-4 生成订单接口测试图形结果

6.4 本章小结

本章对网上预约挂号系统进行功能测试和性能测试。其中功能测试使用的是黑盒测试，且均通过测试。非功能测试使用 Apache JMeter 压测工具，根据测试结果得知在并发数为 3000 的场景下用户登录接口的平均请求响应时间为 506 毫秒，最小响应时间为 502 毫秒，最大响应时间为 611 毫秒；在并发数 3000 的场景下生成订单接口平均请求响应时间为 1005 毫秒，最小与最大响应时间分别为 1002 毫秒和 1142 毫秒，均满足预期要求。

7 总结与展望

7.1 总结

本项目面向当今社会中医院和患者的实际需求，利用软件工程技术开发网上预约挂号系统。完成了系统的需求分析、体系结构和功能模块设计、数据库设计、系统实现，并进行了系统测试。网上预约挂号系统的研发工作取得了以下结果：

（1）网上预约挂号系统在实际应用中实现了设计的功能，运行性能指标符合要求，比较有效地解决了患者“预约难、挂号难、就医难”，以及医院面临的“病源不足，资源闲置”的问题，不仅产生了经济效益，也产生了积极的社会效应；

（2）网上预约挂号系统包括数据字典、订单管理、短信服务、用户管理和业务日志记录五大功能模块，可以满足大部分医院和患者的实际需求。并且在后续的系统实现上，提供了多种支付方式和登录方式，满足当今社会日益增长的需求；

（3）在软件架构设计技术上，系统采用 Spring Cloud 框架搭建微服务的方式，不同的微服务各司其职，实现了“高内聚，松耦合”的目标；

（4）网上预约挂号系统使用 Spring Cloud Gateway 服务网关代替 Nginx 反向代理服务器，采用 Spring Cloud 微服务模式部署，能够降低实际物理资源的消耗，从而挺高对客户端的高并发能力和服务器效率。

通过功能测试与性能测试表明，在最大并发数 3000 的条件下，可以最大限度提高系统响应时间，各个接口最大响应时间都控制在 1 秒钟左右，且所实现的各个模块的功能满足预期需求。系统现已上线运行服务于多家中小型医院，满足用户的大部分需求，最终效果优良。同时该系统也表现出高可扩展性、高可扩充性，可以适应未来日益递增的需求。

7.2 展望

虽然在该系统的需求分析、设计、实现与测试等工作中做到了尽心尽力，但是由于时间和个人水平等原因，该系统仍然存在着许多功能没有实现、一些性能指标没能

做到最好等问题。因此在日后的版本更新中，将致力于将其完善，主要包括以几个方面：

（1）在用户登录的时候，仅仅实现了手机验证码的方式进行登录，对于微信扫码的方式尚未实现。现在大部分人都习惯使用微信扫码的方式登录，这样既便捷，又节约时间，因此该系统在后续版本更新中需要把微信扫码登录加入到其中；

（2）在支付订单时，目前仅实现了微信扫码支付，而支付宝扫码支付和银行卡支付的方式尚未实现，对于现在流行多方式支付的社会来说，仅仅只支持一种支付方式是无法满足广大人民群众需求的，因此该系统在后续版本迭代中需添加多种支付方式。

参考文献

- [1] 戴志鑫, 张鹭鹭. 上海市公立医院便捷就医服务现状研究. 解放军医院管理杂志, 2021, 28(08): 780-782
- [2] 俞越峰, 牛昕, 邵懿. 党建引领视域下公立医院文化建设研究—以上海市同仁医院为例. 领导科学论坛, 2022(04): 104-107
- [3] 樊超, 高汉. 基于不同性别患者的某肿瘤专科医院就医满意度差异研究. 中国医院管理, 2021, 41(08): 86-90
- [4] Xiaoyue Yin, Hui Xie. Hospital Marketing Strategies Based on the Humanistic Management Thought. Proceedings of 2017 2nd EERES International Conference on Transportation, Biomedical Research and Social Sciences, 2017: 429-433
- [5] Junyan Lan, Panlei Wu, Xiaoxin Chen, Junjie Zhang, Mengjia Ying. An Empirical Study on the Influencing Factors of Patients Accepting Behavior in Online Medical System. Proceedings of 2nd International Symposium on Economic Development and Management Innovation , 2020: 130-135
- [6] Fang Yang, Jun Yan, Songtao Li, Chen Yang. Risk Analysis and Safety Design of Mobile Smart Medical System. Proceedings of 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference, 2017: 189-192
- [7] Wang Yizhu, Lu Sifei, Zhou Ying, Zhou Siwen, Zhu Ruiji. Study on Optimum Design of Waiting Space in General Hospitals Based on the Analysis of Accompanying Family. Proceedings of the 14th International Conference on Environment-Behavior Studies, 2020: 1536-1543
- [8] Qin Bona, Zhou Ying. Study on the Architectural Planning of General Hospital from the Standpoint of Diurnal Population Distribution. Proceedings of the 14th International Conference on Environment-Behavior Studies, 2020: 758-764
- [9] Zhuoxi Zhang, Ming Yuan, Hanwei Qian. Research on MySQL Database Recovery and Forensics Based on Binlog. Proceedings of the 11th International Conference on Computer Engineering and Networks, 2021: 750-759
- [10] Qi Fei, Chen Yi, Fu Yuxin. The Impacts of Indoor Elements of Healthcare Facilities on Wayfinding Stress. Proceedings of the 14th International Conference on Environment-

- Behavior Studies, 2020: 1739-1745
- [11] Yichao Luan. Analysis on the Emotional Labor Effect of Medical Staff in Chinese Public Tertiary Hospitals. Proceedings of The first International Symposium on innovation management and Economics, 2021: 377-381
- [12] Haifeng Fan. Research on Network Security System Detection Based on Mysql. Proceedings of 2019 International Conference on Information Science, Medical and Health Informatics. Francis Academic Press, 2019: 689-693
- [13] Pengfei Jiu, Hongping Cheng. Construction of Quality Management System of Grade 3A Hospital Based on TQM Concept. Proceedings of the 3rd Asia-Pacific Social Science and Modern Education Conference, 2020: 145-149
- [14] Masaaki Minami, Ryoko Sakakibara, Taichi Imura, Mika Watanabe, Hideo Morita. Clinical Characteristics of Haemophilus influenzae at General Hospital in the Central Region of Japan. Proceedings of 2016 Workshop 6, 2016: 15-20
- [15] Zhu Ruiji, Zhou Ying, Lu Sifei, Zhou Siwen. Evaluation and Optimization of Hospital Waiting Space Based on Semantic Segmentation. Proceedings of the 14th International Conference on Environment-Behavior Studies, 2020: 91-98
- [16] Liu Chengjie, Cai Zhichang. A Comparative Study on the Spatial Layout of Emergency Departments in General Hospitals of China and Foreign Countries Based on Space Syntax. Proceedings of the 14th International Conference on Environment-Behavior Studies, 2020: 580-586
- [17] Xu Yi, Lu Zhenrui, Zhou Ying. Discussion on the Vertical Traffic Planning of Outpatient and Medical Department in Large General Hospital. Proceedings of the 14th International Conference on Environment-Behavior Studies, 2020: 1762-1769
- [18] 孙剑, 程建军. 基于微信小程序的医院远程挂号系统设计. 电子设计工程, 2021, 29(22): 51-54
- [19] 陶博, 杨妍. 基于贝叶斯理论的医院预约挂号反黄牛机制研究. 现代医药卫生, 2021, 37(18): 3094-3097
- [20] 沈威, 周璐, 姚海燕. 医院门诊系列改造在预约诊疗中的成效. 江苏卫生事业管理, 2021, 32(07): 931-932
- [21] 邢娜, 郑蕾, 王莉. 医院网站预约挂号系统设计. 解放军医院管理杂志, 2021,

28(10): 929-930

- [22]陈博, 焦娣. 门诊预约挂号系统的设计与应用. 智慧健康, 2022, 8(04): 1-4
- [23]吴雨柯. 基于 ASP.NET MVC 的线上预约挂号系统. 轻工科技, 2021, 37(09): 68-69
- [24]时娅楠, 鲍迪. 以号源池为核心预约挂号系统的设计与优化. 数码世界, 2020(03): 245-246
- [25]张磊. 医院预约挂号系统设计与应用探究. 中国信息化, 2018(12): 81-82
- [26]张雯霄, 凌佳, 余睿, 殷宪达, 俞婷婷. 预约挂号系统对患者就医行为影响因素及应用—基于模糊综合评价法的研究. 现代信息科技, 2022, 6(01): 140-143
- [27]黄巧, 曹奕, 李雪. 智慧医疗视角下产科门诊应用全预约挂号系统的回顾性研究. 现代医院, 2021, 21(06): 903-906
- [28]李慧宁, 王文军. 基于 JavaWeb 的医院在线挂号系统设计与实现. 电子元器件与信息技术, 2021, 22(02): 172-175
- [29]张明英. 基于 JSP 的医院在线预约挂号系统设计. 时代农机, 2017, 44(08): 131-132
- [30]巩蕾. 基于智能移动端的医院预约挂号系统设计与研究. 电脑编程技巧与维护, 2020(02): 48-49
- [31]刘晓龙, 万振. 预约挂号及门诊叫号系统的设计与优化. 医疗装备, 2017, 30(19): 35-37
- [32]严春风. 基于 asp.net 技术的医院预约挂号系统的实现. 电脑知识与技术, 2018, 14(12): 72-73
- [33]王永平, 张伟. 门诊预约挂号系统在兰州地区医院中的对接及应用. 甘肃科技, 2017, 33(19): 10-11
- [34]单树倩, 任佳勋. 基于 SpringBoot 和 Vue 框架的数据库原理网站设计与实现. 电脑知识与技术, 2021, 17(30): 40-50
- [35]周常志, 甘恒. 基于 SpringBoot 的智慧就业服务平台的设计与实现. 电脑知识与技术, 2021, 17(28): 182-184
- [36]刘娟, 任光芹, 左欣, 湛佳军, 崔忠伟. 基于 Spring Boot 的财政一体化管控平台设计与实现. 物联网技术, 2021, 11(12): 78-80

- [37] 马荣彦. Spring Cloud 微服务框架浅析. 现代电影技术, 2021, 20(10): 47-50
- [38] 杨思誉, 刘海霞, 童基均, 冉宇瑶. Spring Cloud 云原生应用开发与实现. 软件导刊, 2021, 20(06): 183-187
- [39] 黄静, 吴涵. 基于 Spring Cloud 的工厂可视化管理系统的设计与实现. 软件工程, 2021, 24(10): 59-62
- [40] 陈洁. MySQL 数据库安全课程教学方法优化分析. 无线互联科技, 2021, 18(24): 160-161
- [41] 赵学作. MySQL 数据库主从数据同步的设置. 网络安全和信息化, 2021(11): 94-97
- [42] 杨耀勇. 基于 WebLogic 与 Redis 的客服监控系统的设计实现. 信息与电脑, 2021, 33(10): 121-125
- [43] 仲灵毓. Redis 存储系统在广电监测系统中的应用. 现代电视技术, 2022(02): 139-142
- [44] 樊鸿涛. 基于 Nginx 的美术辅助课程信息学习微系统开发. 微型电脑应用, 2021, 37(08): 168-169
- [45] 许诺. 大数据下基于 Nginx 负载均衡的教务系统优化设计. 信息技术与信息化, 2021(06): 191-193
- [46] 赵琳. 基于 CentOS 7 系统实现 Nginx 负载均衡. 网络安全和信息化, 2022(02): 96-99
- [47] 邱敏, 梁婷婷, 梁天友. 基于 Echarts 数据分析的校园疫情防控系统设计. 物联网技术, 2021, 11(10): 100-102
- [48] 王曙燕, 郑佳妮, 孙家泽. 基于页面对象的 Web 应用测试用例生成方法. 计算机应用, 2020, 40(01): 212-217
- [49] 唐承玲, 王虎, 李光平, 唐春蓬. 基于 JMeter 的 Web 性能测试研究. 电脑与电信, 2021(06): 65-68
- [50] 严强强, 马骏, 夏迎舟. 基于 JMeter 的指挥信息系统软件性能测试研究. 信息化研究, 2020, 46(02): 74-78