



成绩	
----	--

西北大学

本科毕业论文（设计）

题目：基于分布式微服务的
医院预约挂号平台
的设计与实现

学生姓名 庞晓宇

学 号 2020118100

指导教师 卢燕宁

院 系 信息科学与技术学院

专 业 软件工程

年 级 2020 级

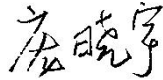
教务处制

二〇二四年六月

诚信声明

本人郑重声明：本人所呈交的毕业论文（设计），是在导师的指导下独立进行研究所取得的成果。毕业论文（设计）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或在网上发表的论文。

特此声明。

论文作者签名： 

日 期： 2024 年 6 月 10 日

摘 要

随着互联网技术的发展，线上挂号预约已经成为医疗服务的重要组成部分，市场上现有多元化的挂号平台在用户选择医院和医生时，面临信息碎片化和操作复杂的困扰。本文提出并构建一个多家医院资源整合的统一预约挂号平台以提升用户的就医体验和医院的服务质量。

本平台基于 Spring 微服务技术栈开发，根据数据的特点选用不同的数据持久化策略，通过消息队列实现服务间异步通信以保证系统的高性能运行。

本平台实现了医院的自助接入申请与后台审核流程，确保医疗资源的规范化管理和用户信息的安全保障。前台门户为用户提供了多维度的医院搜索、科室排班查询、便捷的预约挂号流程以及订单管理功能，并通过 API 接口实现医院信息的动态同步与展示，帮助用户更有效地获取医疗资源，减轻医疗资源分配不均衡的现象。

本平台经过严格的系统测试，在功能完备性、系统性能和用户体验上均达到了预期目标，不仅简化了用户挂号过程，优化了医疗资源配置，还为医院提供了便捷的运营管理工具。未来展望中，系统将继续迭代升级，融入更多先进技术，不断提升服务质量和用户体验。

关键词：分布式 微服务 Spring 技术栈 预约挂号平台

Abstract

With the development of Internet technology, online registration appointment has become an important part of medical services. The existing diversified registration platforms in the market face the problems of information fragmentation and complex operation when users choose hospitals and doctors. This paper proposes and constructs a unified appointment registration platform with multi-hospital resource integration to improve users' medical experience and hospital service quality.

This platform is developed based on the Spring microservice technology stack. According to the characteristics of data, different data persistence strategies are selected to realize asynchronous communication between services through message queue to ensure the high performance of the system.

This platform realizes the self-service access application and background review process of the hospital to ensure the standardized management of medical resources and the security of user information. The front desk portal provides users with multi-dimensional hospital search, department scheduling query, convenient appointment registration process and order management functions, and realizes dynamic synchronization and display of hospital information through API interface, helping users more effectively obtain medical resources and reducing the imbalance of medical resource allocation.

After rigorous system testing, the platform has reached the expected goals in terms of functional completeness, system performance and user experience, which not only simplifies the user registration process, optimizes the allocation of medical resources, but also provides a convenient operation management tool for hospitals. In the future outlook, the system will continue to iterate and upgrade, integrate more advanced technologies, and continuously improve the quality of service and user experience.

Keywords: Distributed Systems; Microservices; Spring technology stack; reservation registration platform

目录

1 绪论	1
1.1 设计选题的背景与意义	1
1.2 理论与实证准备	1
1.3 本论文结构及内容	1
2 需求分析	3
2.1 前台门户	3
2.1.1 医院部分	3
2.1.2 用户部分	3
2.1.3 其他	3
2.2 后台管理	3
2.3 业务流程	4
2.3.1 医院接入	5
2.3.2 数据同步与展示交互	5
2.3.3 用户交互与预约流程	5
2.3.4 支付与订单管理	5
2.3.5 订单变更与通知机制	5
2.3.6 现场服务与订单查询	6
3 系统设计	7
3.1 技术选型	7
3.2 架构设计	8
3.2.1 服务拆分与注册中心	8
3.2.2 微服务网关设计	8
3.2.3 持久化存储	8
3.2.4 异步处理与消息队列	9
3.2.5 前端技术栈	9
3.2.6 系统架构图	9
3.3 模块划分	9
3.4 数据模型	10
3.5 通用与配置抽取工具类	13
3.6 网关设计	15
3.7 微服务设计	15
3.7.1 数据字典微服务设计	15
3.7.2 医院微服务设计	16
3.7.3 用户微服务设计	17
3.7.4 短信微服务设计	18
3.7.5 对象存储微服务设计	19
3.7.6 订单微服务设计	20
3.7.7 定时任务微服务设计	20
3.7.8 统计分析微服务设计	21
3.8 微服务客户端设计	21
3.9 SDK 设计	22
3.10 数据库设计	23

- 4 系统实现 26
 - 4.1 开发环境配置 26
 - 4.2 数据字典模块实现 26
 - 4.3 医院管理模块实现 26
 - 4.4 用户模块实现 27
 - 4.5 短信服务模块实现 27
 - 4.6 对象存储模块实现 27
 - 4.7 订单管理模块实现 27
 - 4.8 定时任务模块实现 28
 - 4.9 统计分析模块实现 28
- 5 部署与测试 29
 - 5.1 部署 29
 - 5.2 后台管理测试 29
 - 5.2.1 后台管理界面展示 29
 - 5.2.2 后台管理测试 32
 - 5.3 前端门户测试 33
 - 5.3.1 前台门户界面展示 33
 - 5.3.2 前端门户测试 38
- 6 总结与展望 40
 - 6.1 总结 40
 - 6.2 展望 40
- 参考文献 42

1 绪论

1.1 设计选题的背景与意义

互联网技术的迅猛发展为传统行业带来了革新的机会，医疗行业也不例外。在线挂号预约服务作为医疗服务数字化转型的重要组成部分，有效解决了患者面临的诸多不便。传统的线下挂号方式不能满足现代社会的需求，这促使了在线挂号预约平台的发展。目前市场上存在众多此类平台均未实现资源整合与统一；这不仅给用户带来了额外的负担，还影响了用户体验^[1]。

面对这一状况，构建一个资源统一的预约挂号平台显得尤为必要。该平台可以整合各个医院的信息和挂号服务，为用户提供一个一站式的解决方案，极大地提高了就医效率和用户满意度。本文旨在设计和实现一个基于分布式微服务的预约挂号统一平台，通过高效的技术手段，解决现有系统的不足，为用户和医院双方提供更优质的服务。

1.2 理论与实证准备

首先对现有的在线挂号平台及其功能特点进行了深入探究，并学习了相关的技术理论，如分布式系统设计、数据库优化、用户交互设计等。通过问卷调查和访谈等手段收集了大量用户和医院的反馈信息，最终明确本系统的设计目标和功能需求。

本平台选择基于 Java、SpringBoot、SpringCloud 等技术栈进行开发。同时，使用 MySQL 和 MongoDB 进行数据持久化；利用 RabbitMQ 实现服务间的异步消息通信，确保系统的高效运行^[2]。

本系统的核心目标在于：前台构建一个标准化的操作界面以确保用户能够在统一平台上便捷地完成对多家医院的挂号预约操作；后台整合医院、科室和医生信息，实现信息的透明化展示，辅助用户做出明智选择；此外设计有效的推广机制和反馈系统，增进医患交流，使医院能根据用户反馈不断优化服务；同时通过智能推荐和资源调度算法，减少医疗资源的不合理分配。

1.3 本论文结构及内容

本论文分为六章：

第一章：绪论章节，探讨了设计选题的背景与意义，概览了理论基础与实证准备并规划了全文的结构。

第二章：需求分析章节，详尽解析了前台界面、后台管理系统及核心业务流程的需求，

细化至各功能模块的具体要求。

第三章：系统设计章节，明确了技术方案选择，全面展示了系统架构设计、模块切分策略，并借助图表直观呈现了各模块设计细节。

第四章：系统实现章节，描述了开发环境的搭建配置，还逐一对各模块实施的功能进行了细致说明。

第五章：部署与测试章节，具体说明了系统的部署步骤与方法，并报告了各项功能的测试结果。

第六章：总结与展望章节，归纳总结了本研究的主要成果，同时展望了该领域未来可能的研究方向与发展趋势。

2 需求分析

2.1 前台门户

2.1.1 医院部分

医院检索：提供多维度筛选条件，展示医院列表及其等级、地域、名称等信息，同时展示常见科室列表和平台公告。

医院详情：包含科室介绍、医院详细信息、预约须知、停诊通知等内容，并提供订单查询和取消功能。

门诊排班查询：展示每日各科室放号数量、剩余号源信息，以及医生具体排班时间和挂号费用。

挂号预约流程：用户可以按照实际需求选择排班、填写就诊人信息，完成预约挂号后生成订单，支持支付宝支付挂号费，并可在线取消预约订单。

2.1.2 用户部分

用户认证与登录：支持手机号码验证登录（短信验证码方式）和微信扫码（关注公众号）两种认证方式。

实名认证功能：用户填写自己的个人信息并上传相关的证件照片进行实名认证。

就诊人信息管理：用户可以管理多个就诊人的基本信息，包括查看、添加、编辑和删除就诊人信息。

个人订单管理：用户在个人页面可以查看自己的订单列表以及订单详情，针对未支付的订单可以进行支付，已支付的订单可以进行取消。

讨论区功能：建立用户交流社区，方便用户分享就医体验、获取健康咨询等。

2.1.3 其他

消息通知机制：系统在订单状态变更、就诊时间变动、停诊通知和平台公告发布时，应及时推送给相关用户或医院。

医院接入流程：提供医院自助接入申请功能，后台审核通过后发放密钥和 SDK，以便新医院快速加入平台。

2.2 后台管理

首页：设计一个仪表板用于数据概览，实时显示关键运营指标，包括注册用户总数、接入的医院总数、累计订单数量以及累计订单金额。

数据字典：使用一个树形表格实现对行政区划、医院等级、证件类型、学历和民族等基础数据的维护，支持批量导入与导出操作。

医院管理：构建一个可编辑的医院接入信息展示列表，允许管理员进行新增、删除、修改医院相关信息，如 API 基础路径、密钥、联系人信息，并支持启用和禁用医院账号的功能。

用户管理：该模块提供用户账户列表的浏览、搜索、新增、编辑和删除功能，同时具备用户状态（启用/禁用）的调整能力。

消息管理：支持查看所有已发送和待发送的消息记录，同时提供向用户或医院发送各类消息的通知功能。

对象存储：用于查看、管理和操作用户认证资料及其他相关图片等文件的存储情况，支持文件的预览、上传、下载和删除操作。

订单管理：提供订单列表的查询、筛选、详情展示，以及订单状态更改的操作权限。

定时任务：提供能够手动触发诸如发送就诊提醒、自动更新订单状态等定时任务的功能。

统计分析：通过可视化图表展现注册用户数、接入医院数、订单数和订单金额的趋势变化和统计分布。

2.3 业务流程

业务流程如图 2-1，医院首先需与平台完成对接并上传相关信息，随后平台负责展示医院的详细信息。当用户访问平台进行挂号时，平台将实时与医院进行交互。此过程支持科室、排班等的更新，订单的创建及更新等一系列交互。

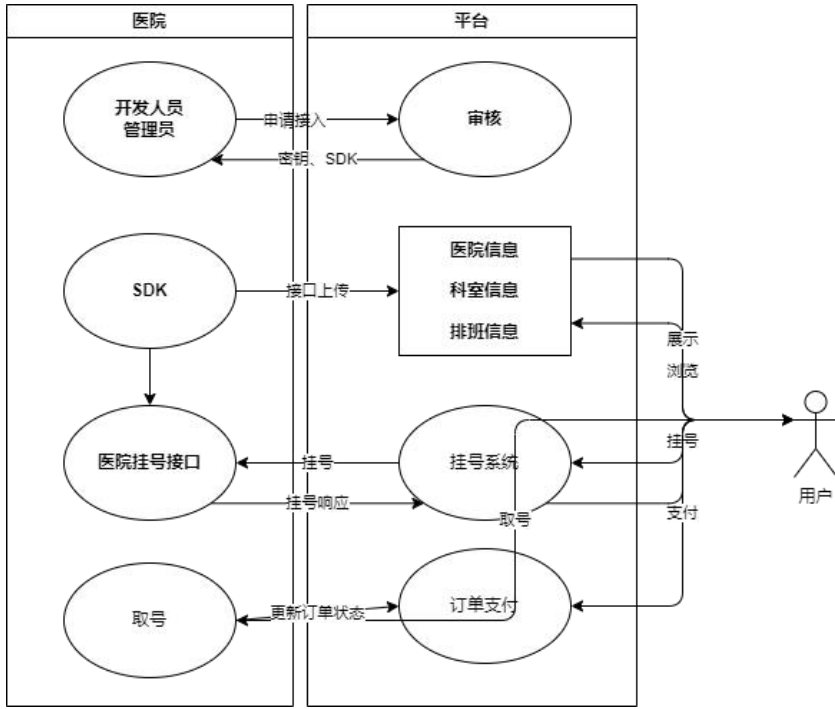


图 2-1 业务流程图

2.3.1 医院接入

医院通过前端门户提交接入申请，提供必要的基础信息，包括但不限于医院名称、地理位置、级别、联系方式等，并设定 API 基础路径。平台后台收到接入申请后，由管理员进行审核。审核内容包括确认医院资质和所提供信息的真实性与完整性。审核通过后，后台管理系统自动生成密钥和配套的 SDK，并将这些信息通过安全渠道发放给医院方。医院技术人员利用发放的 SDK 和密钥对接到平台，开始推送和更新医院的基础信息、科室信息以及医生排班表等相关数据至平台服务器。

2.3.2 数据同步与展示交互

医院通过授权的 SDK 接口成功将最新的医院基本信息、科室结构、医生专业介绍、实时的排班计划等关键数据上传至平台。平台端接收到这些数据后，运用高效的后台处理机制进行整合、验证和储存，并确保在前台门户页面即时更新展示，使用户能够随时查看并了解到所有已接入医院的详尽情况，包括但不限于医院等级认证、各科室的功能划分及其特色服务，以及各个医生的专业特长和当前可预约时段。

2.3.3 用户交互与预约流程

当用户访问平台时，可根据自己的需求对不同医院、科室和医生进行多维度筛选和比较，进而选定心仪的目标进行预约挂号。用户可在规定的预约时间段内自由选择合适的医生和时间，并在平台上完成挂号操作。系统会在用户下单瞬间基于预定规则创建订单，记录下预约详情，包括用户身份信息、就诊时间和医生信息等核心要素。

2.3.4 支付与订单管理

预约完成后，用户可以通过集成的安全支付通道完成在线支付，支持多种支付方式，如银行卡、第三方支付平台等。支付成功后，订单状态将被更新为已付款，并发送确认短信或邮件通知用户。此外，平台还会保留完整的电子账单供用户随时查阅。

2.3.5 订单变更与通知机制

在整个医疗服务过程中，平台保持与医院系统的紧密联动，定期同步医院方面的最新信息变化，如医生排班调整、科室临时关闭等。同时，医院亦可通过 SDK 主动推送更新信息至平台，共同确保用户所见信息的时效性和准确性。针对任何订单状态的更改，无论是用户发起的预约取消、改签请求，还是由于医院原因导致的就诊时间变动、临时停诊通知等情况，平台均能迅速响应并通过消息通知模块及时告知用户，以减少因信息滞后带来的不便。

2.3.6 现场服务与订单查询

到达就诊日，用户依据系统内的提示完成取号操作，正式进入诊疗流程。同时，用户不仅可以在就诊当天查询其预约订单的状态，还能够在任意时段登录平台查询历史订单记录，便于了解个人就诊历史和后续跟进事宜。通过这种闭环式的全流程管理，平台致力于为用户提供便捷、透明且高效的一站式医疗服务体验。

3 系统设计

3.1 技术选型

在调研与结合自身实际情况后，本系统根据不同的应用领域最终决定采用以下技术选型，如表 3.1-3.4.

表 3.1 后端开发技术选型

后端开发	技术选型	说明
基础框架	Spring Boot	Java 企业级应用程序的快速开发框架，具有简洁的编码风格和强大的自动配置能力 ^[3] 。
微服务框架	Spring Cloud + Alibaba	提供分布式系统开发的一整套解决方案，阿里巴巴开源实现进一步增强了微服务治理功能。
程序构建工具	Maven	用于 Java 项目的构建与依赖管理，简化构建过程和版本控制。
持久层框架	Mybatis-Plus	通过扩展 MyBatis 构建的持久层框架，对创建、读取、更新及删除等基本操作进行了优化封装，显著提升了开发工作的效率。

表 3.2 数据存储技术选型

数据存储	技术选型	说明
关系型数据库	MySQL	开源、广泛应用的关系型数据库，用于存储结构化、事务性强的数据，如用户信息、订单详情等 ^[4] 。
非关系型数据库	MongoDB	用于存储 JSON-like 文档，适合结构灵活、查询复杂的数据，如医院信息、科室信息、排班信息等。
缓存系统	Redis	高性能内存键值数据库，用于存储热点数据、会话信息等，提高系统响应速度。

表 3.3 分布式服务技术选型

后端开发	技术选型	说明
注册与配置中心	Nacos	用于服务的注册发现、配置管理，提供服务治理和配置变更推送功能。
认证机制	JWT	JSON Web Tokens，用于无状态、安全的用户认证和授权 ^[5] 。
API 网关	Spring	提供统一的路由、过滤器和熔断机制，实现服务的统一入口

	Gateway	和权限控制。
负载均衡	Nginx	高性能的反向代理服务器和负载均衡器，实现流量分配和故障转移。
消息中间件	RabbitMQ	实现系统间的异步解耦，处理如订单状态通知、消息推送等消息传递任务 ^[6] 。

表 3.4 前端开发技术选型

后端开发	技术选型	说明
打包工具	Vite	构建工具，提供更快热更新速度和更优的开发体验。
基础 JS 框架	Vue.js、Nuxt.js	Vue.js 用于构建用户界面，Nuxt.js 则为 Vue.js 提供服务端渲染(SSR)支持。
路由管理	Vue Router	Vue.js 官方路由库，用于管理 SPA 应用的路由导航。
状态管理	Vuex	Vue.js 的状态管理模式，集中管理组件的状态数据。
UI 组件库	Element-Plus	基于 Vue.js 的丰富 UI 组件库，用于快速构建页面。
网络请求	axios	基于 Promise 的 HTTP 客户端，用于前后端数据交互。
图表库	echarts	功能丰富的 JavaScript 图表库，用于数据可视化展示。

3.2 架构设计

3.2.1 服务拆分与注册中心

本系统采用分布式微服务架构设计，系统的核心功能模块化为八大微服务：数据字典服务、医院管理服务、用户服务、短信服务（依托阿里云 SMS）、对象存储服务（利用阿里云 OSS）、订单服务、定时任务服务以及统计分析服务。所有微服务均注册至 Nacos 注册中心，通过服务发现机制实现动态管理和通信^[7]。

3.2.2 微服务网关设计

使用统一的网关作为系统对外服务的入口，负责路由转发、权限控制、限流降级等非功能性需求，保证系统的稳定性和安全性^[8]。医院通过系统提供的 SDK 调用网关，进而访问各个微服务进行数据的同步与交互。

3.2.3 持久化存储

关系型数据存储：使用 MySQL 存储系统中涉及的关系性强、事务性要求高的数据^[9]，例如用户信息、订单详情等。文档型数据存储：选用 MongoDB 存放诸如文档模型的数据^[10]，

如医院信息、科室信息等结构不固定的数据。缓存服务：借助 Redis^[11]作为缓存中间件，提高系统性能，降低数据库压力，存储高频访问的数据如用户 session^[12]、订单状态等。

3.2.4 异步处理与消息队列

使用 RabbitMQ 作为消息队列，实现系统的异步解耦，例如订单状态变更、消息推送等场景，保证系统的高可用性和扩展性^[13]。

3.2.5 前端技术栈

后台管理系统采用 Vue.js 框架开发，实现丰富的数据展示和管理功能，满足后台管理人员对系统各项功能的精细化操作需求。前台门户采用 Nuxt.js 构建，结合 Vue.js 的 SSR 特性优化 SEO 效果和首屏加载速度，为用户提供流畅的浏览和交互体验^[14]。

3.2.6 系统架构图

架构图的架构如图 3-1，本系统通过合理的微服务架构设计、多样化的数据库支持、高效的 消息队列以及现代化的前端框架，构建了一个高性能、高可用、易于维护和扩展的预约 挂号统一平台。

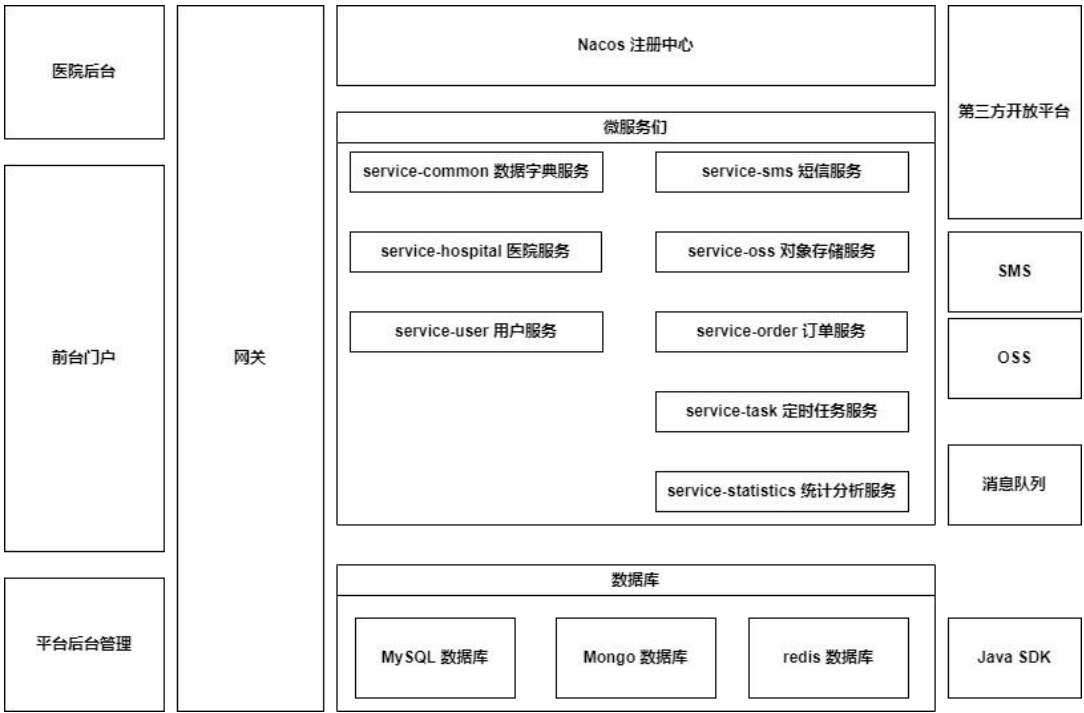


图 3-1 系统架构图

3.3 模块划分

图 3-2 所示是系统模块划分图，通常包括了所有核心模块以及它们之间的依赖和交互方式，直观地展示了系统的不同组成部分及其相互关系。

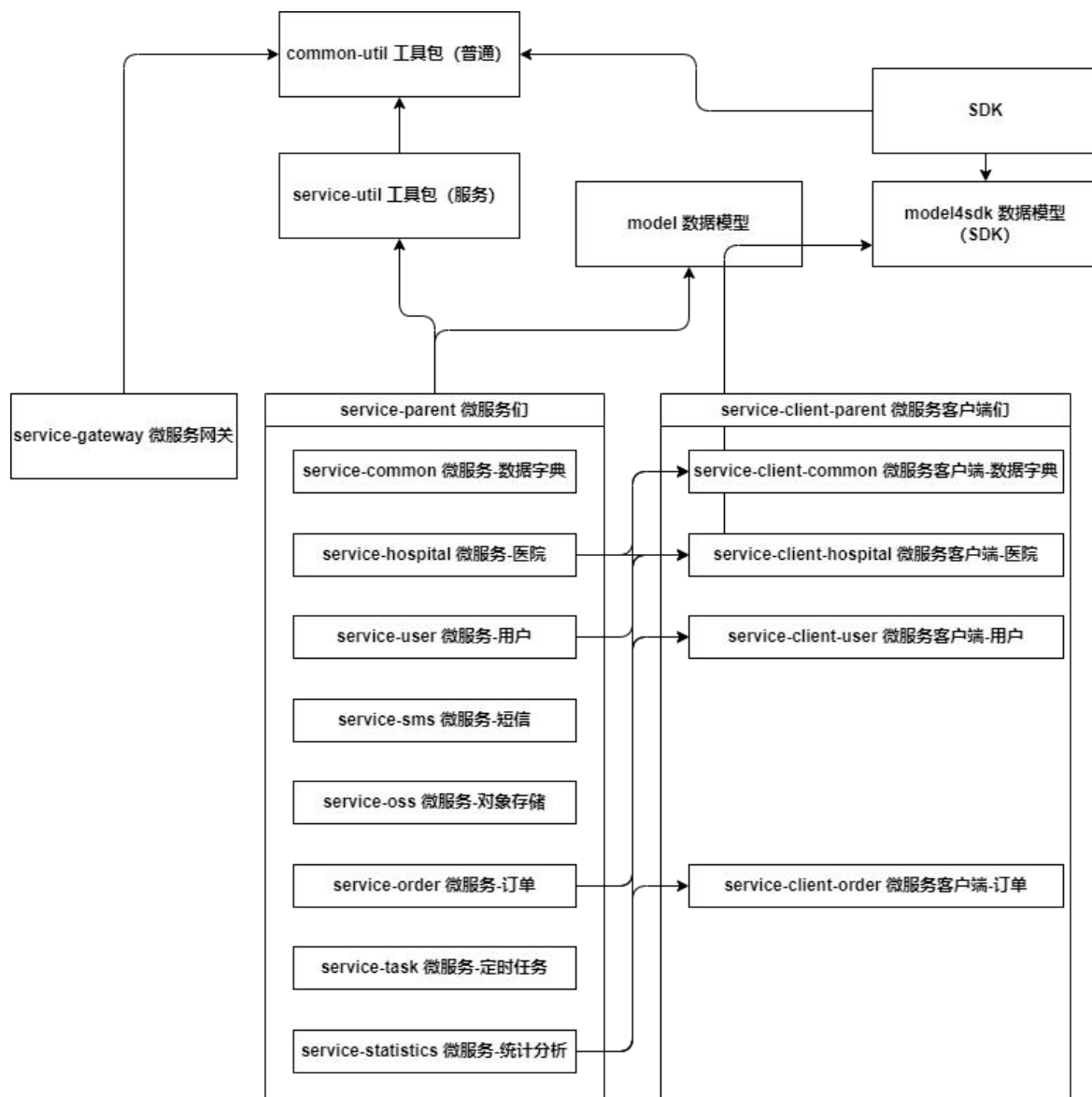


图 3-2 模块总览图

3.4 数据模型

图 3-3，图 3-4，图 3-5，图 3-6，图 3-7 定义了系统所使用到的实体类设计，涵盖了各个模块的数据库实体以及接口的参数值对象等。

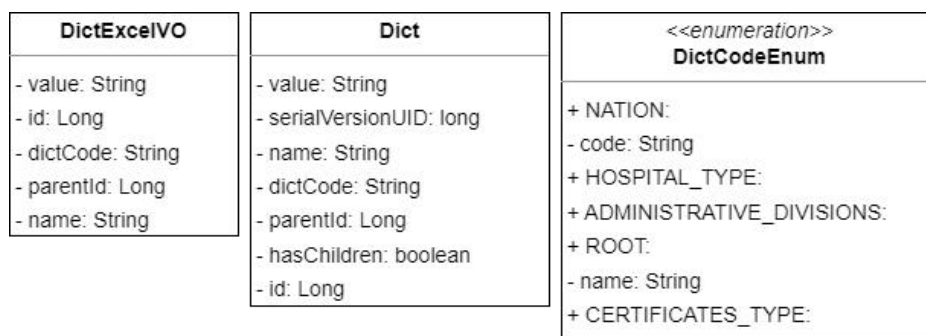


图 3-3 数据字典实体类图

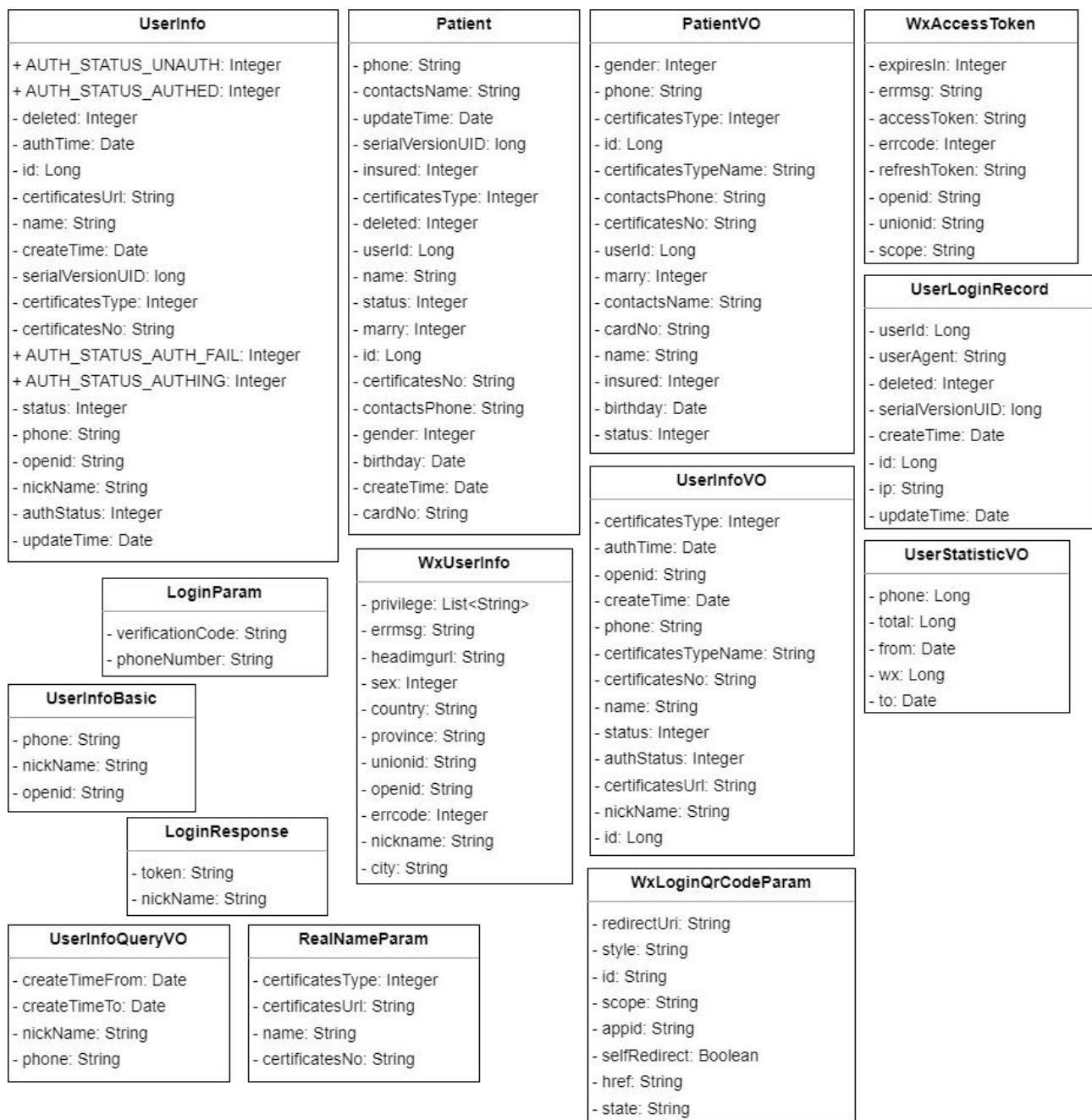


图 3-5 用户实体类图

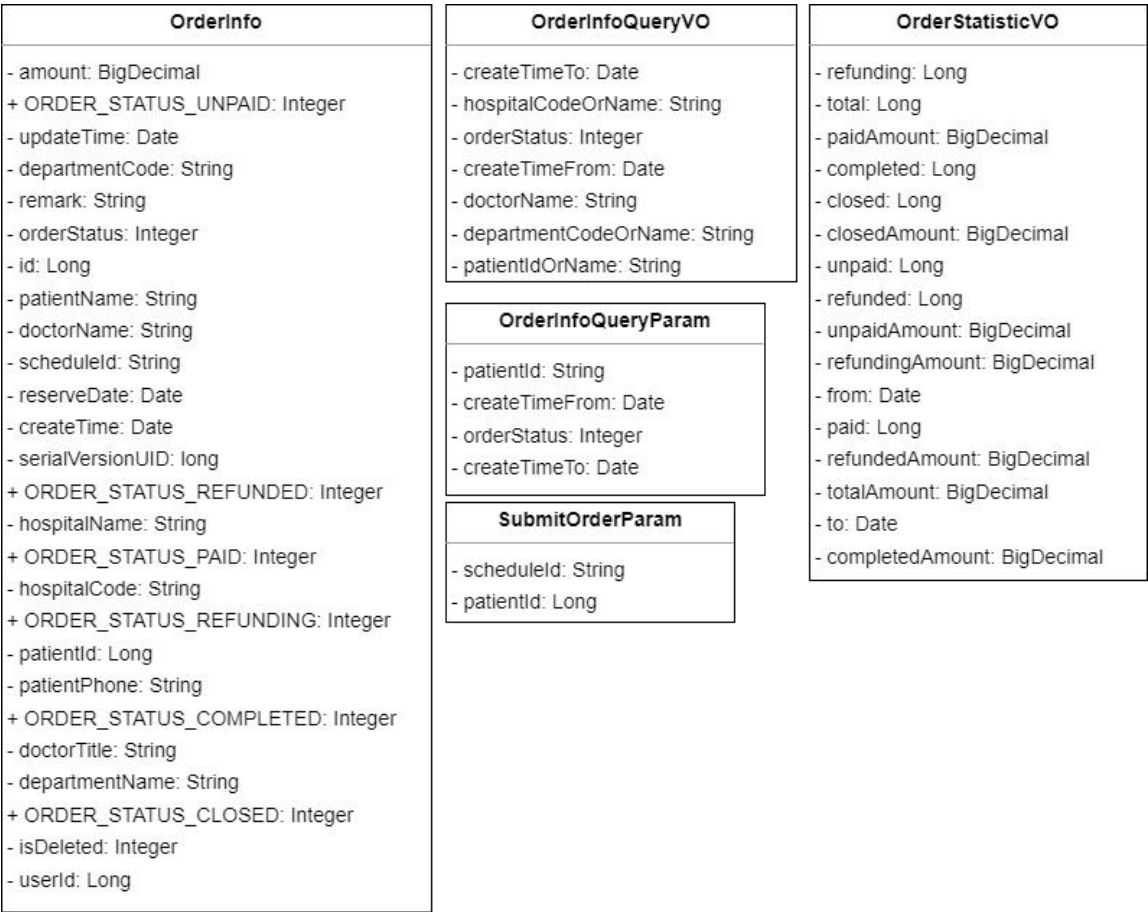


图 3-6 订单实体类图

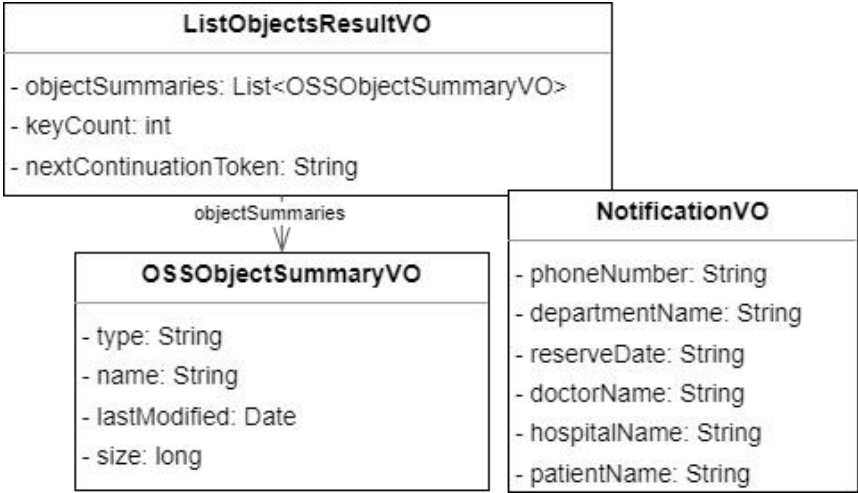


图 3-7 其他实体类图

3.5 通用与配置抽取工具类

图 3-8 是基于微服务的通用操作和配置抽取的工具类，包含用于微服务间通信、请求响应处理、异常处理、API 调用封装等功能的类，旨在简化服务层代码的复杂性，提高复用性和可维护性。

图 3-9 则定义了一些更通用的工具类，这些类封装了常见的、跨模块的实用功能，例如日期时间处理、字符串操作、加密解密、随机数处理、文件操作等。



图 3-8 服务工具类图

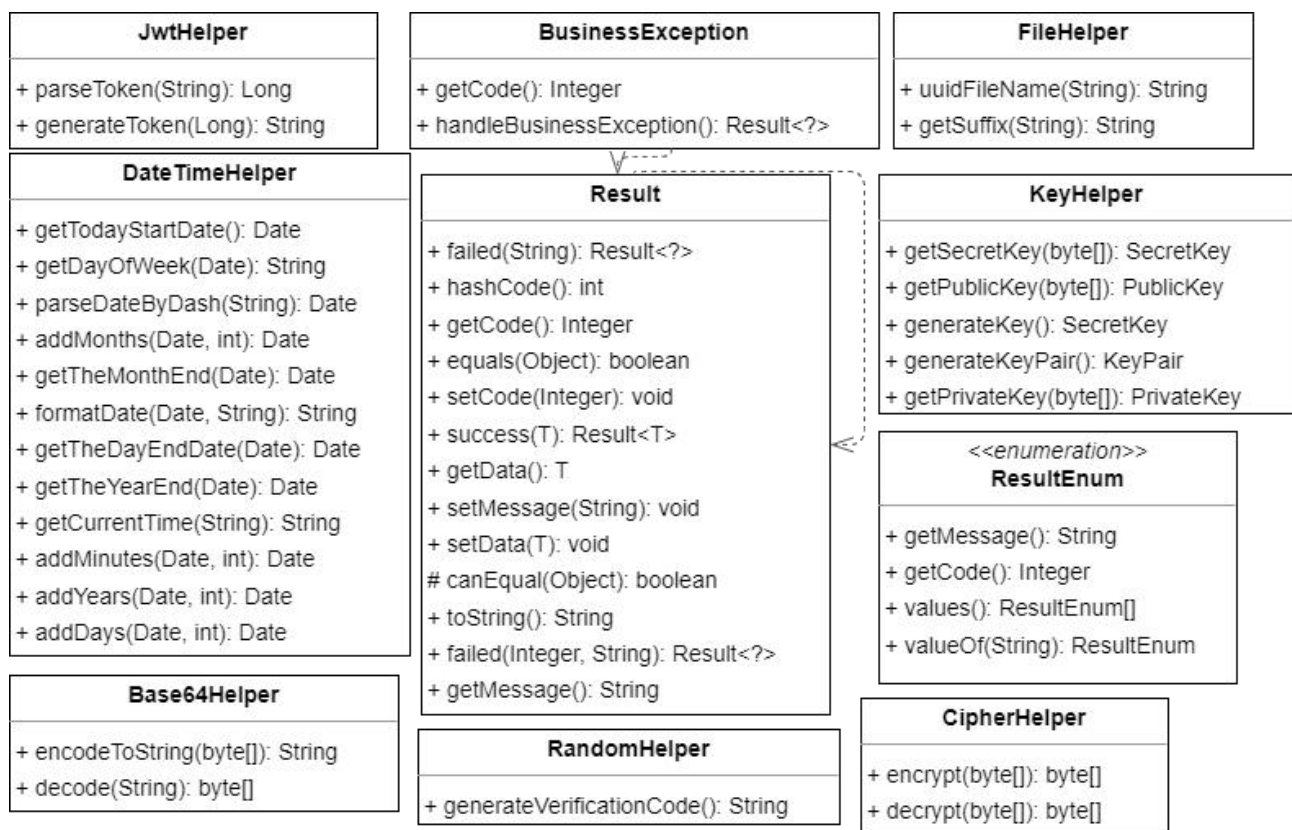


图 3-9 通用工具类图

3.6 网关设计

网关是系统对外暴露的 API 网关设计，包括其路由配置、身份验证、授权等方面的功能设计与实现，它是外部请求进入系统内部各微服务前的第一道屏障。

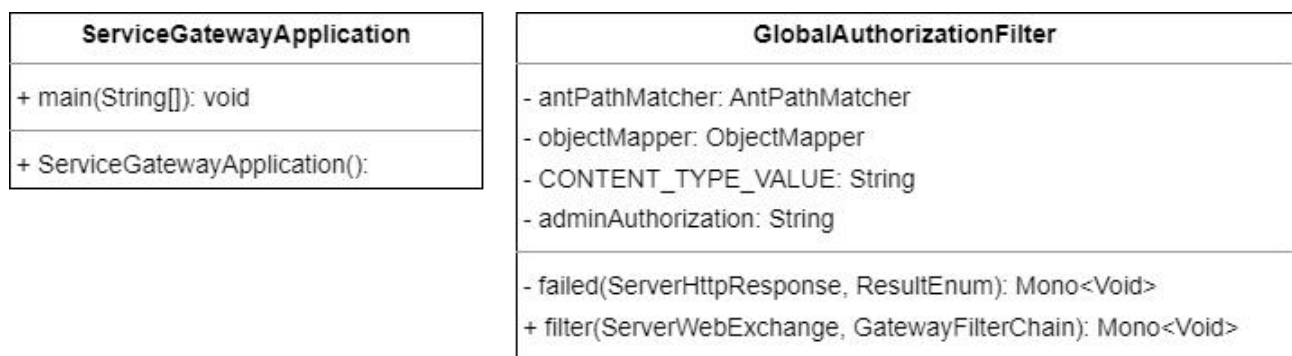


图 3-10 网关类图

3.7 微服务设计

3.7.1 数据字典微服务设计

用于管理系统的元数据，如枚举值、静态配置项等，并确保数据的一致性和准确性。

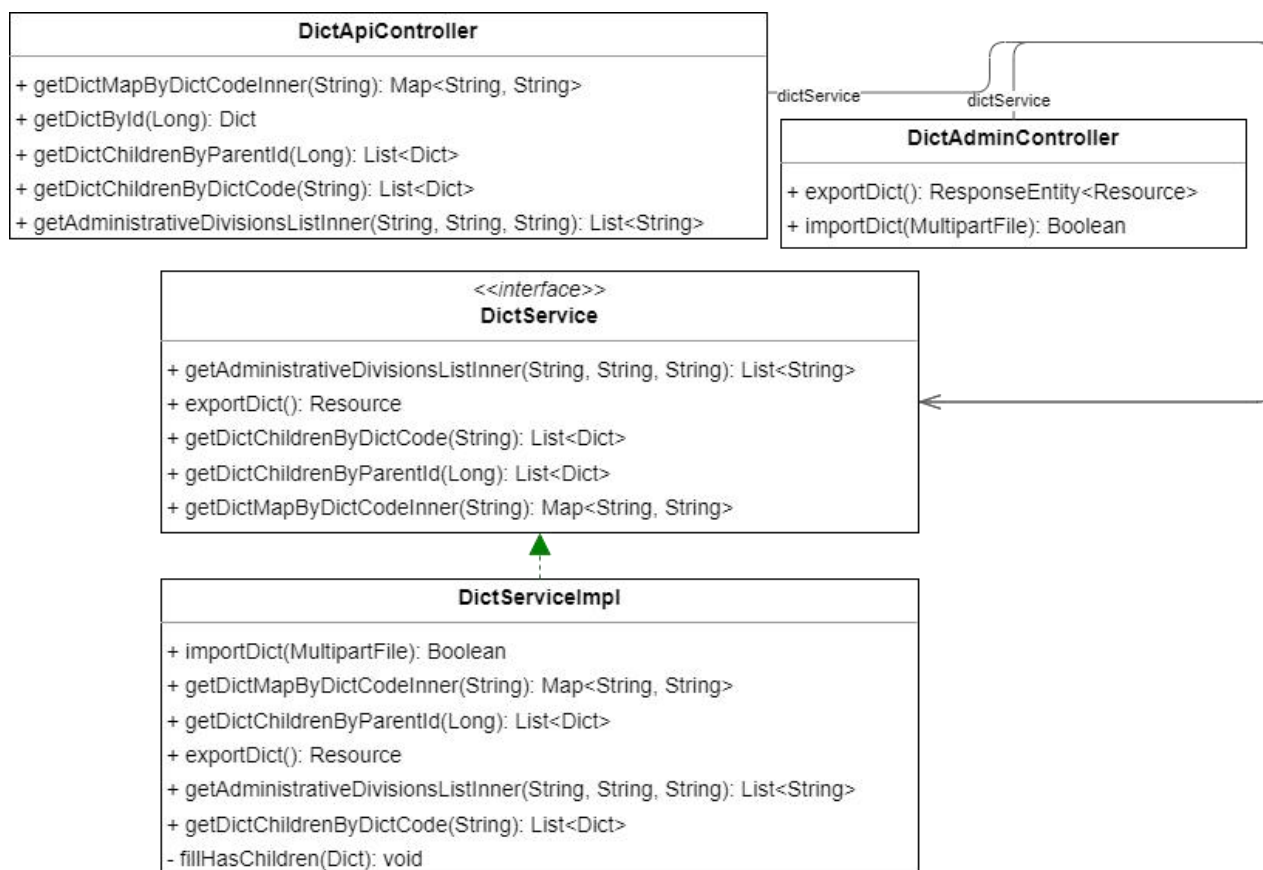


图 3-11 数据字典微服务类图

3.7.2 医院微服务设计

系统的核心微服务，提供了包括但不限于医院基本信息维护、科室组织管理、医生排班调度及患者预约挂号等关键服务功能。

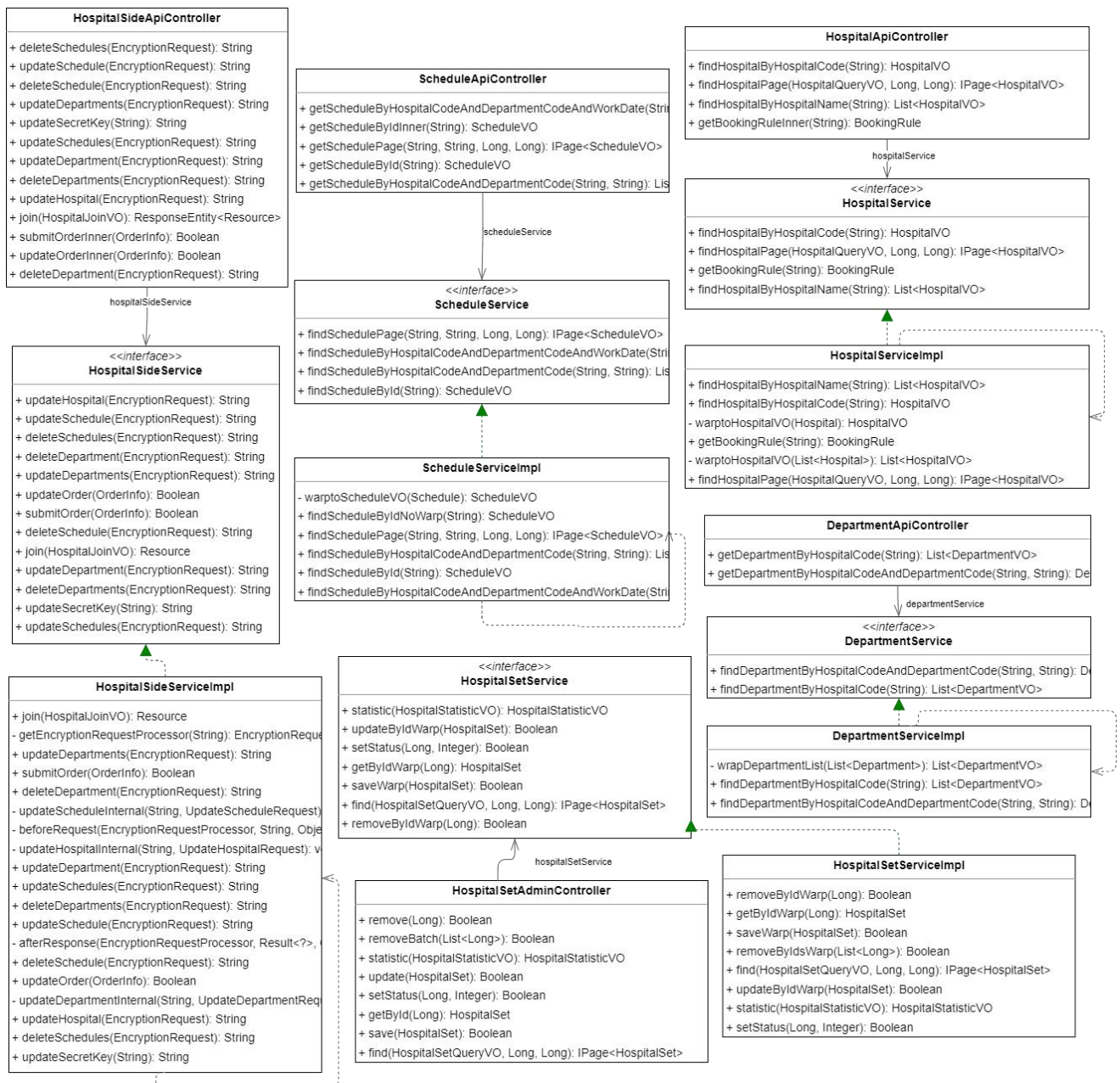


图 3-12 医院微服务类图

3.7.3 用户微服务设计

用户账号生命周期管理，包括用户注册、登录认证、就诊人信息的管理。

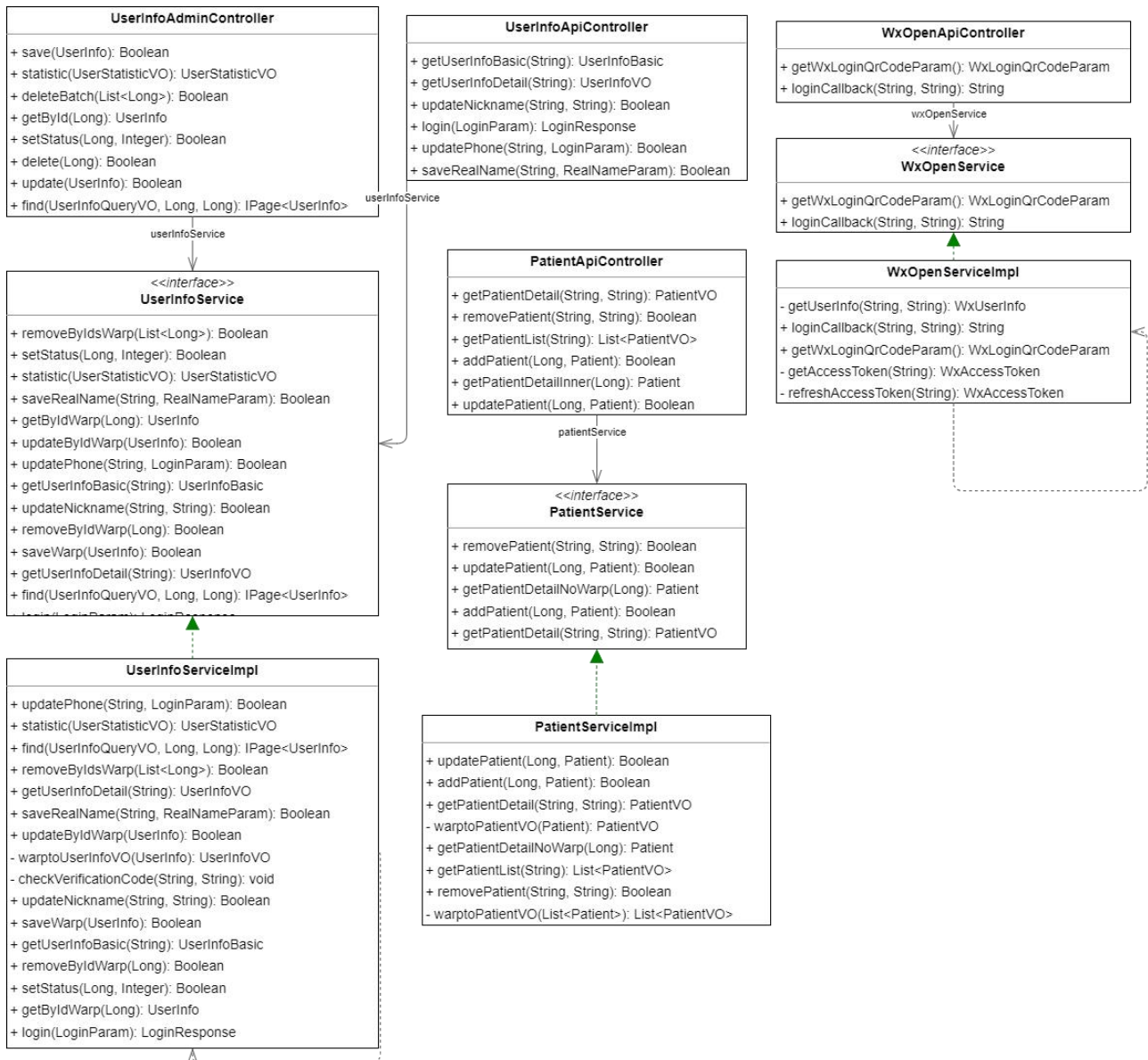


图 3-13 用户微服务类图

3.7.4 短信微服务设计

针对短信发送功能设计的微服务，集成阿里云 SMS 服务，处理验证码发送、通知消息推送等业务场景。

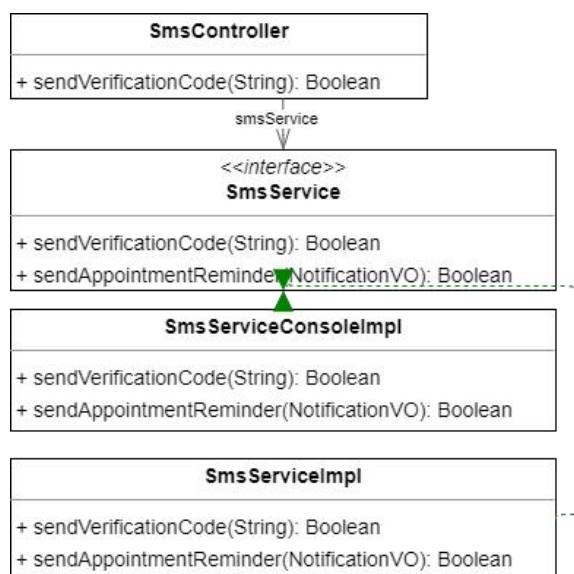


图 3-14 短信微服务类图

3.7.5 对象存储微服务设计

用于用户认证信息、文件的存储的微服务，通过阿里云 OSS 提供文件上传下载、存储桶管理等功能。

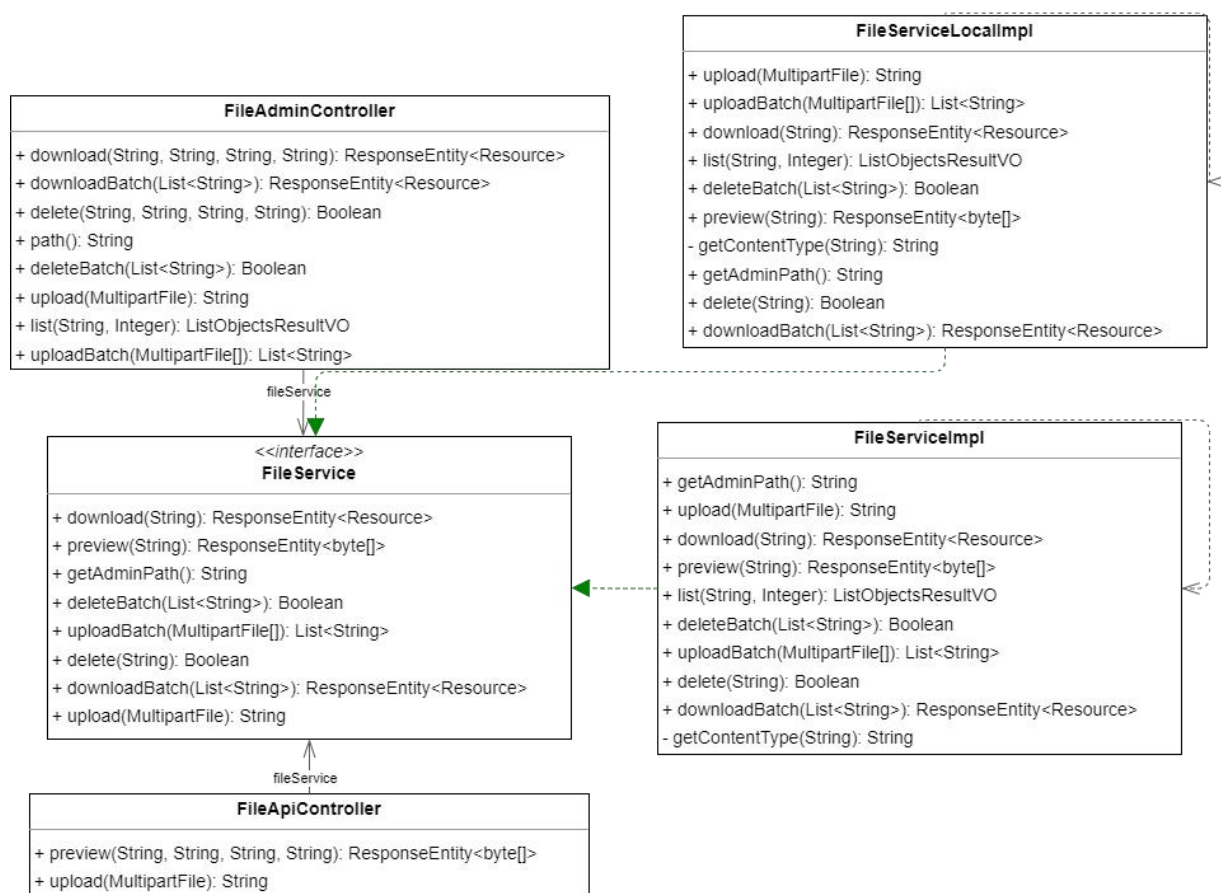


图 3-15 对象存储微服务类图

3.7.6 订单微服务设计

处理预约挂号、支付、退订等一系列与订单生命周期相关的业务流程。

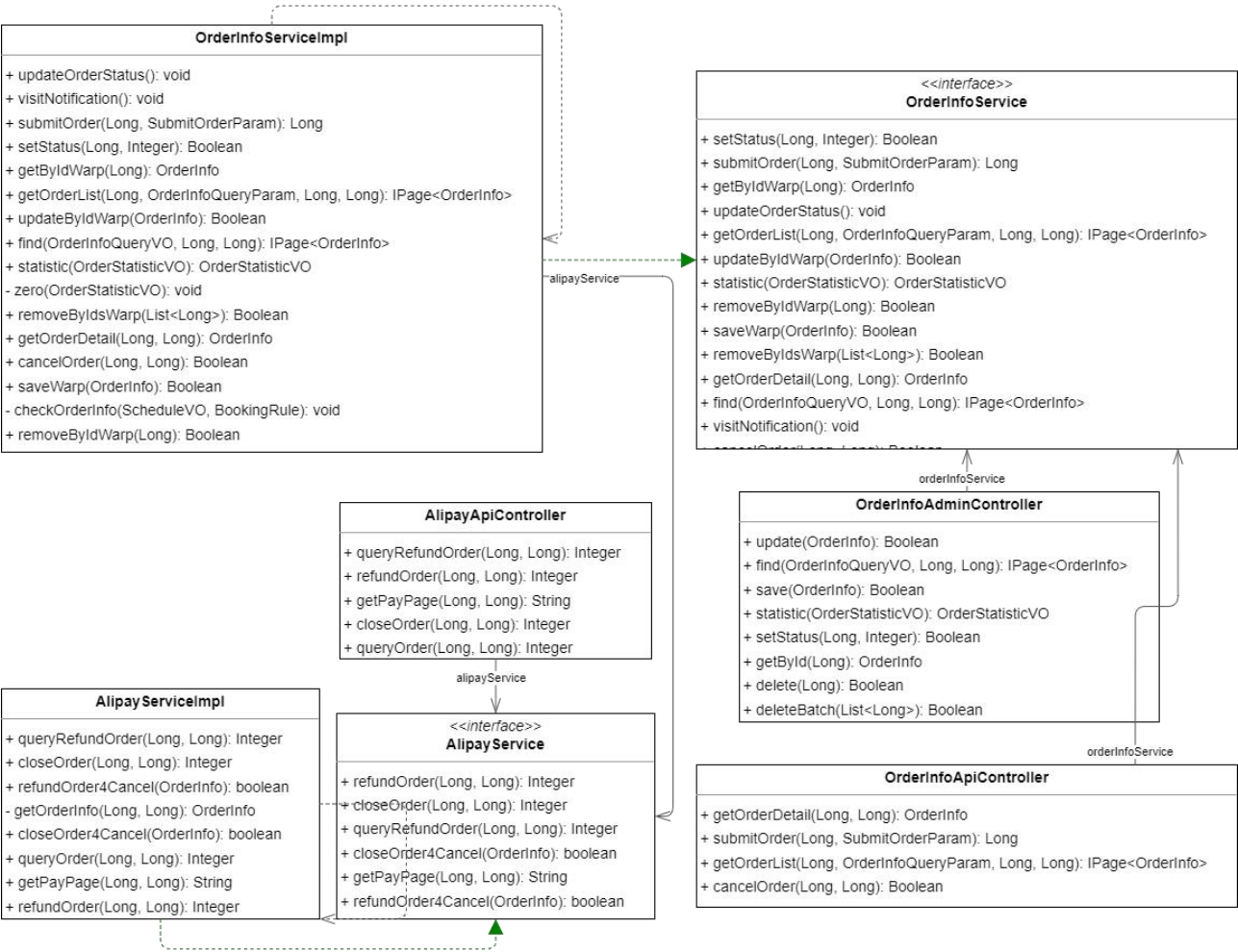


图 3-16 订单微服务类图

3.7.7 定时任务微服务设计

负责执行周期性任务，如通知用户订单状态更新、就诊通知、排班信息更新等后台作业。

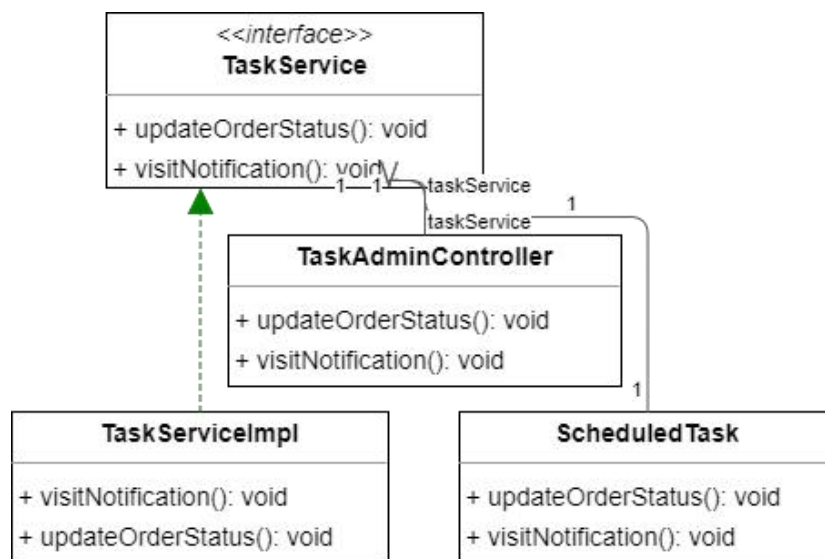


图 3-17 定时任务微服务类图

3.7.8 统计分析微服务设计

对注册用户、接入医院和挂号订单进行数据聚合以生成报表等统计分析功能。



图 3-18 统计分析微服务类图

3.8 微服务客户端设计

系统内部各微服务之间交互时所使用的客户端设计，包括微服务的注册发现与调用（服务间通信）。

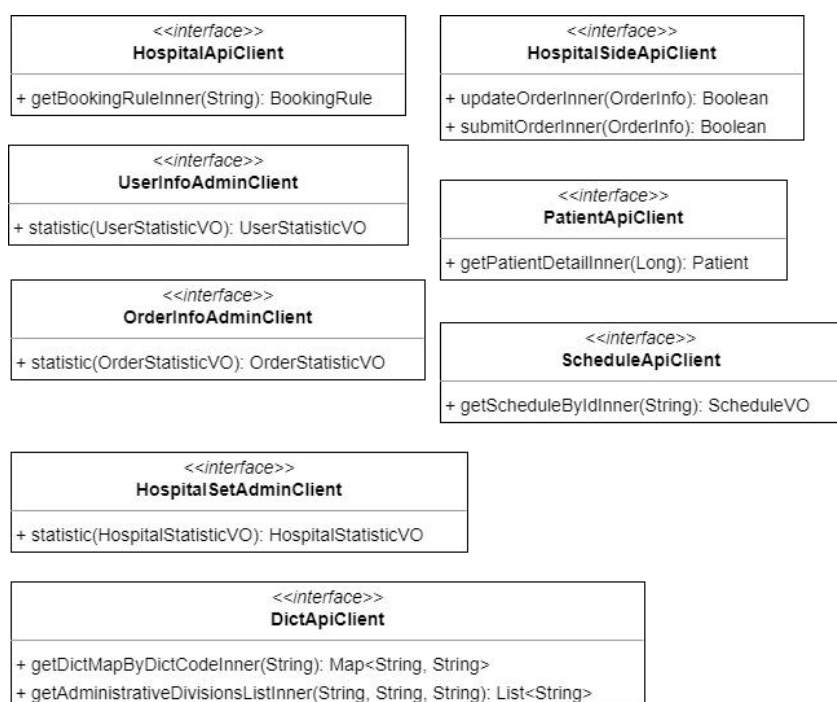


图 3-19 微服务客户端类图

3.9 SDK 设计

为医院侧提供的 SDK 设计，该 SDK 包含了与系统进行交互所需的接口、认证机制、错误处理等内容，方便医院便捷地整合并对接到系统中。



图 3-20 SDK 类图

3.10 数据库设计

依据前述需求分析、系统架构设计及各模块设计细节，本节重点展示了关系型数据库中表结构的具体设计方案，详述于表 3.5 至表 3.9 中。

表 3.5 数据字典表

列名	数据类型	空?	主键	自增	默认值	备注
id	bigint	否	是	否	NULL	id
parent_id	bigint	否	否	否	NULL	上级 id
name	varchar(127)	是	否	否	NULL	名称
value	varchar(32)	是	否	否	NULL	值
dict_code	varchar(32)	是	否	否	NULL	编码

表 3.6 医院设置表

列名	数据类型	空?	主键	自增	默认值	备注
id	bigint	否	是	否	NULL	id
hospital_code	varchar(32)	是	否	否	NULL	医院编号
hospital_name	varchar(127)	是	否	否	NULL	医院名称
api_url	varchar(127)	是	否	否	NULL	api 基础路径
sign_key	varchar(511)	是	否	否	NULL	签名密钥
contacts_name	varchar(16)	是	否	否	NULL	联系人
contacts_phone	varchar(16)	是	否	否	NULL	联系人手机
status	tinyint	否	否	否	1	状态
create_time	timestamp	否	否	否	当前时间	创建时间
update_time	timestamp	否	否	否	当前时间	更新时间
is_deleted	tinyint	否	否	否	0	逻辑删除

表 3.7 用户信息表

列名	数据类型	空?	主键	自增	默认值	备注
id	bigint	否	是	否		id
phone	varchar(16)	是	否	否		手机号
openid	varchar(127)	是	否	否		微信 openid

nick_name	varchar(64)	是	否	否		昵称
name	varchar(16)	是	否	否		姓名
certificates_type	tinyint	是	否	否		证件类型
certificates_no	varchar(64)	是	否	否		证件编号
certificates_url	varchar(255)	是	否	否		证件路径
auth_status	tinyint	否	否	否	0	状态
auth_time	datetime	是	否	否		认证有效期
status	tinyint	否	否	否	1	状态
create_time	timestamp	否	否	否	当前时间	创建时间
update_time	timestamp	否	否	否	当前时间	更新时间
is_deleted	tinyint	否	否	否	0	逻辑删除

表 3.8 就诊人信息表

列名	数据类型	空?	主键	自增	默认值	备注
id	bigint	否	是	否		id
user_id	bigint	否	否	否		用户 id
name	varchar(16)	是	否	否		姓名
phone	varchar(16)	是	否	否		手机号
certificates_type	tinyint	是	否	否		证件类型
certificates_no	varchar(64)	是	否	否		证件编号
gender	tinyint	是	否	否		性别 (
is_marry	tinyint	是	否	否		是否结婚
birthday	date	是	否	否		出生日期
is_insured	tinyint	是	否	否		是否有医保
card_no	varchar(64)	是	否	否		就诊卡号
contacts_name	varchar(16)	是	否	否		联系人姓名
contacts_phone	varchar(16)	是	否	否		联系人手机
status	tinyint	否	否	否	1	状态
create_time	timestamp	否	否	否	当前时间	创建时间

update_time	timestamp	否	否	否	当前时间	更新时间
is_deleted	tinyint	否	否	否	0	逻辑删除

表 3.9 订单表

列名	数据类型	空?	主键	自增	默认值	备注
id	bigint	否	是	否		id
user_id	bigint	否	否	否		用户 id
hospital_code	varchar(32)	是	否	否		医院编号
hospital_name	varchar(127)	是	否	否		医院名称
department_code	varchar(32)	是	否	否		科室编号
department_name	varchar(127)	是	否	否		科室名称
doctor_name	varchar(32)	是	否	否		医生名称
doctor_title	varchar(32)	是	否	否		医生职称
schedule_id	varchar(127)	是	否	否		排班 id
reserve_date	datetime	是	否	否		预约时间
patient_id	bigint	否	否	否		就诊人 id
patient_name	varchar(64)	是	否	否		就诊人名称
patient_phone	varchar(64)	是	否	否		就诊人手机
amount	decimal(10, 2)	是	否	否		订单金额
order_status	tinyint	否	否	否	0	订单状态
remark	varchar(512)	是	否	否		备注
create_time	timestamp	否	否	否	当前时间	创建时间
update_time	timestamp	否	否	否	当前时间	更新时间
is_deleted	tinyint	否	否	否	0	逻辑删除

4 系统实现

4.1 开发环境配置

在本系统开发过程中，采用了一系列主流且稳定的技术栈，具体版本信息如表 4.1 所示。

表 4.1 开发环境技术组件版本

技术组件	使用版本
JDK	21.0.1
Node.js	20.10.0
Nacos	2.3.0
MySQL	8.2.0
MongoDB	6.0.12
Redis	7.2.3
RabbitMQ	3.12

4.2 数据字典模块实现

本模块构建了一套全面的数据字典管理系统，其核心 API 接口囊括了数据字典的导入导出、检索和层级查询等功能：

- 数据导入接口：使用 Excel 进行数据字典的导入，方便用户批量录入或更新数据字典项。
- 数据导出接口：具备将所有字典项数据输出至 Excel 文件的功能，便于线下维护和备份。
- 单一查询接口：可通过 ID 或编码精确查找特定字典项信息。
- 子项查询接口：支持根据父级 ID 或编码递归获取下级字典子项。
- Map 型查询接口：提供编码映射服务，快速返回对应编码的子项信息。
- 地区编码查询接口：特别定制了基于省市区编码查询地区全名的服务。

4.3 医院管理模块实现

本模块完整地实现了对医院信息的 CRUD 操作，覆盖了从新增到状态变更的所有管理功能点，其中包括但不限于：

- 创建、编辑、删除及批量删除医院设置信息；

- 针对医院的全方位查询服务，支持按名称、编码等方式检索；
- 设计了动态的医院数据统计接口，可依据不同参数实时汇总医院相关数据；
- 构建了完善的医院申请加入流程，涵盖申请提交、密钥更新以及多层面信息（医院概况、科室结构、排班安排等）的维护和批量处理；
- 同步实现了与订单功能的深度集成，支持用户提交订单、跟踪订单状态、查询订单详情，以及获取医院科室和排班信息等。

4.4 用户模块实现

用户模块承载了用户账户生命周期的全程管理，主要包括：

- 基于增删改查（CRUD）的操作，可完成用户的注册、注销、批量管理及状态变更；
- 支持用户登录验证、实名认证流程，允许用户自主更新联系方式（如手机号码、昵称等）；
- 集成了微信扫码登录功能，实现二维码生成、登录回调处理等环节的安全便捷接入；
- 管理并维护用户与其关联的就诊人信息，提供就诊人添加、删除、查询及信息更新服务。

4.5 短信服务模块实现

本模块专注于短信通信服务，具体功能有：

- 提供验证码短信发送接口，可迅速向目标手机号码发送验证码，并实时反馈发送结果；
- 配合定时任务系统，执行诸如就诊提醒通知等自动化短信发送任务，确保用户及时接收到关键信息。

4.6 对象存储模块实现

为了高效管理和访问用户上传的文件资源，设计了丰富易用的对象存储服务接口，包括：

- 获取管理路径、上传单个或批量上传多个文件；
- 下载单个或选择多个文件批量打包下载；
- 删除单个或多个文件记录；
- 查看文件列表包括文件的相信信息；
- 专为用户场景优化，提供文件上传接口和安全预览服务。

4.7 订单管理模块实现

订单模块实现了全面而细致的订单资源管理功能，既满足后台管理需求，又兼顾用户端

操作权限：

- 创建订单：经由管理员或已验证用户，使用 POST 方式提交订单信息创建新的挂号订单；
- 删除订单：可以通过订单 ID 对订单进行删除，支持批量删除多个订单；
- 更新订单：不仅可以根据订单 ID 修改订单状态，而且也能根据业务需求全面支持订单其他信息的更新操作；
- 查询订单：对于订单的查询操作，除了可以使用订单 ID 进行查询外，还支持通过条件过滤（含分页），以及用户个人订单列表查询；
- 订单状态变更：通过专门的状态更新接口，使得订单流转过程清晰可控；
- 下单流程：用户借助安全的身份验证机制发起下单请求，系统将根据用户提交的下单请求生成订单；
- 订单详情获取：用户通过身份认证后，可查看其订单列表；在订单详情页面，用户能了解到订单的支付金额及其他具体信息；
- 支付相关接口：包括获取支付页面链接、查询交易状态、关闭交易、发起退款及查询退款状态等一系列支付流程辅助功能；
- 与第三方支付平台（例如支付宝）无缝对接，确保订单支付流程的安全与完整性。

4.8 定时任务模块实现

该模块支持人工触发或自动执行一系列关键任务，如：

- 手动触发发送就诊提醒通知的任务，保障患者按时就诊；
- 自动化更新订单状态任务，遵循业务逻辑规则，精准调控订单进展。

4.9 统计分析模块实现

为了深入洞察系统运行状况，统计分析模块提供了针对医院、用户及订单的多样统计分析接口：

- 接受自定义参数输入，灵活生成不同维度的统计数据报告；
- 助力运营决策，通过数据分析反映系统各项核心指标表现。

5 部署与测试

5.1 部署

在进行测试之前，首先需要部署整个系统环境。为了简化部署过程并确保一致性，本系统为此编写了一键部署脚本来完成环境准备、服务打包、环境初始化、服务配置和最终启动工作。脚本首先执行清理任务，删除所有服务容器并清除旧的 **Docker** 镜像（如果存在），确保开始新的部署前环境干净整洁。接下来，脚本会依次构建后端 **Java** 应用、前端资源和模拟服务。紧接着脚本将会进行环境初始化，创建并配置必要的服务依赖，如 **Nacos** 和 **Redis** 实例，并引导用户输入各微服务所需的配置以自动生成配置文件。之后，脚本将构建所有微服务的 **Docker** 镜像^[15]。

在镜像的构建完成后，脚本将以守护态模式启动所有服务容器。通过这个全面而自动化的部署脚本，可以快速高效地搭建起基于分布式微服务架构的预约挂号统一平台，并确保部署过程中的一致性和可靠性。

5.2 后台管理测试

5.2.1 后台管理界面展示

根据前面的介绍，这部分分为首页，数据字典，医院用户订单等的管理以及统计分析模块，演示见图 5-1 至 5-7。

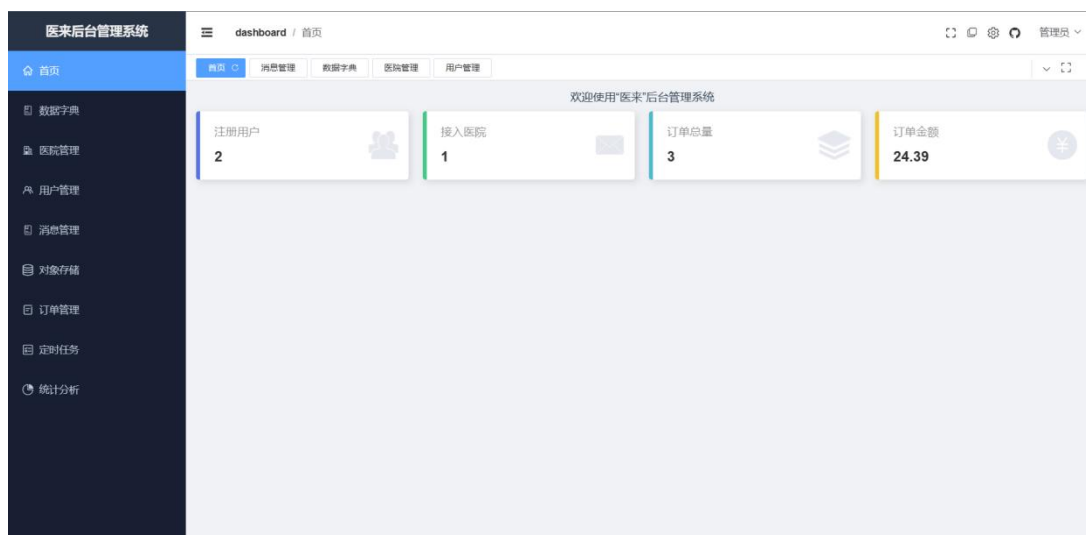


图 5-1 后台管理-首页

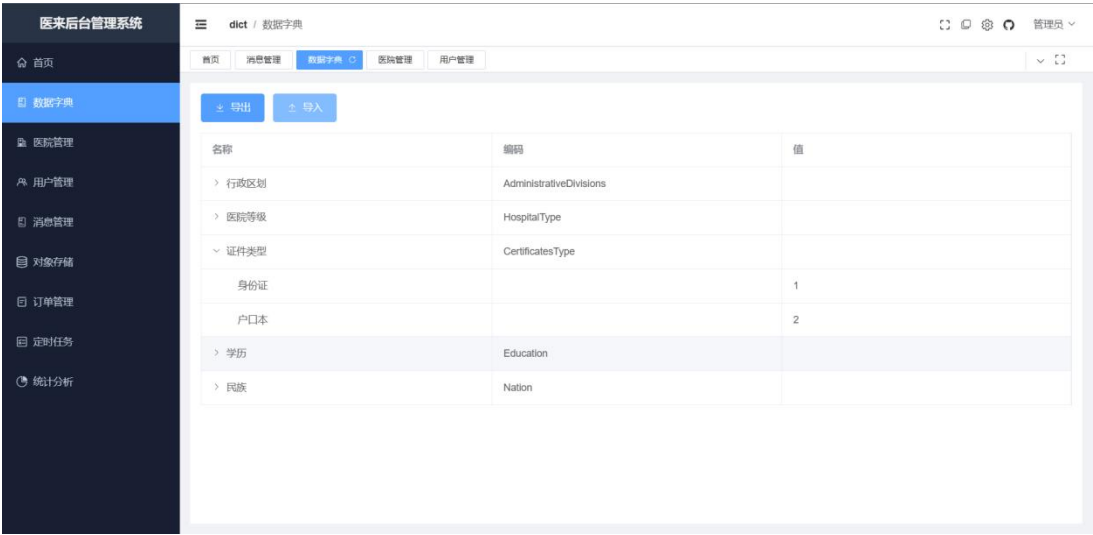


图 5-2 后台管理-数据字典

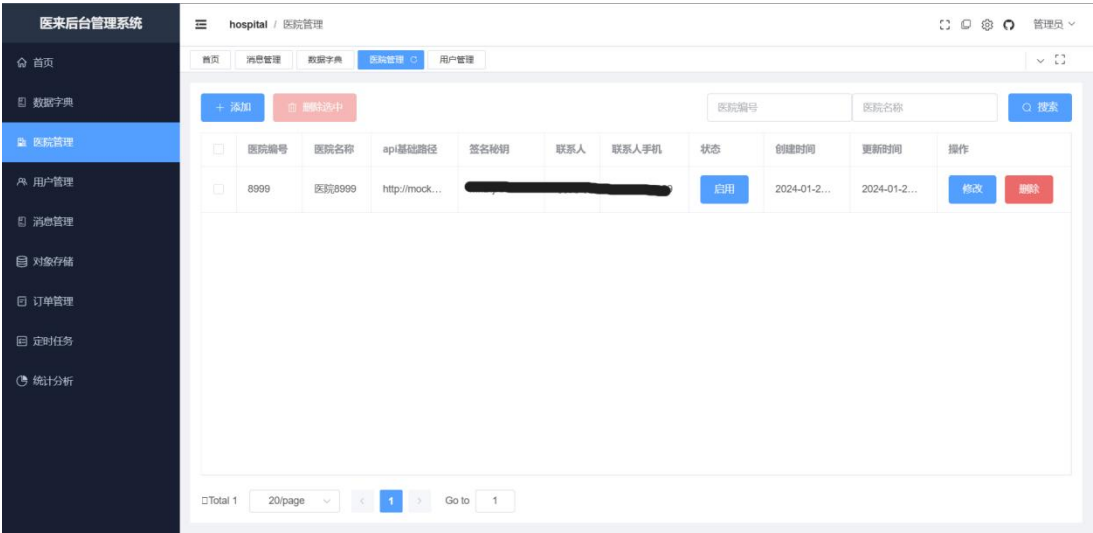


图 5-3 后台管理-医院管理



图 5-4 后台管理-用户管理

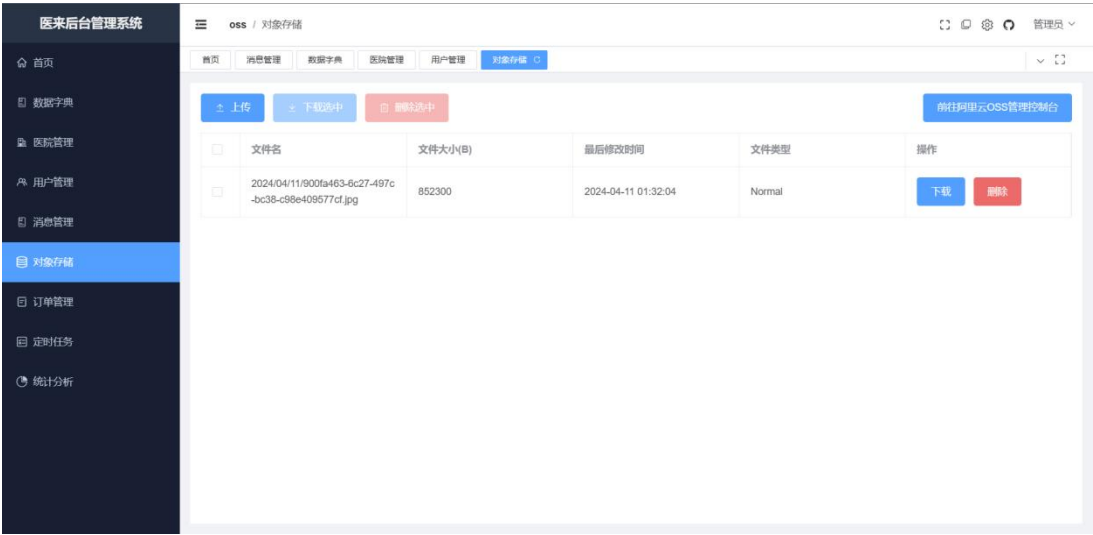


图 5-5 后台管理-文件管理

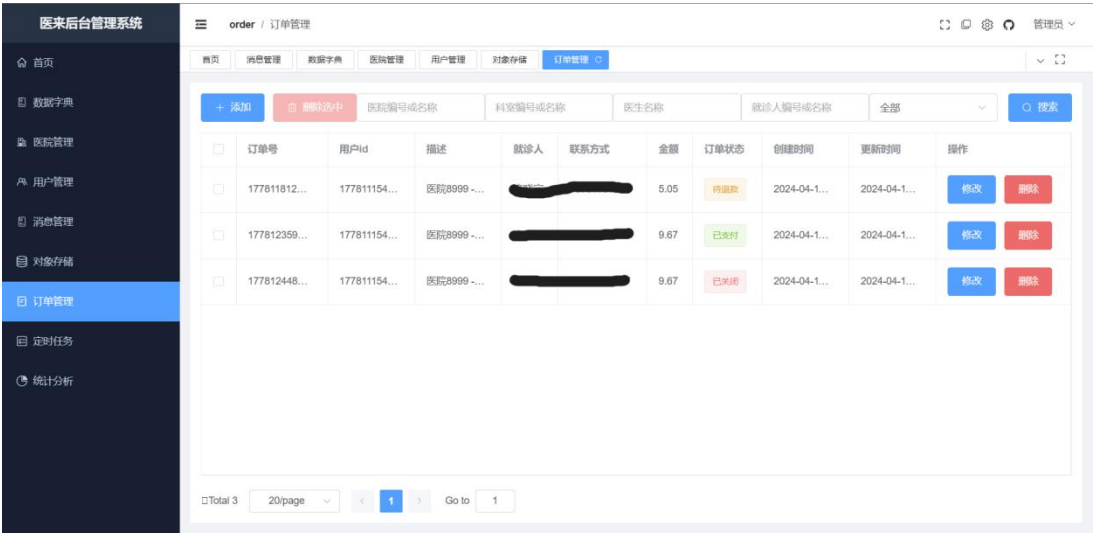


图 5-6 后台管理-订单管理

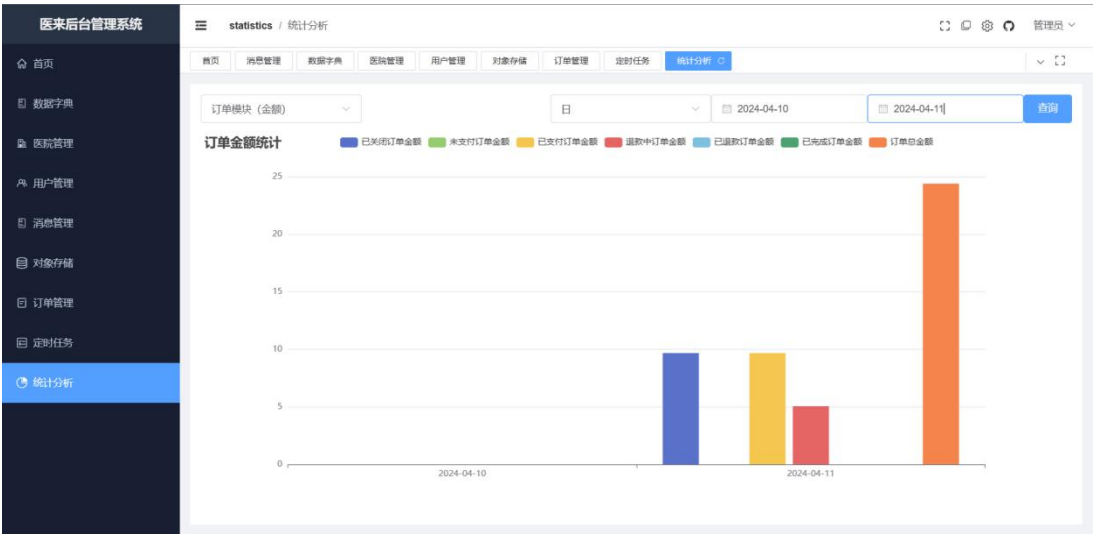


图 5-7 后台管理-统计分析

5.2.2 后台管理测试

该部分的测试主要包括后台管理员登录、登出；后台管理首页数据展示；数据字典展示、导入导出；医院管理增删改查，状态变更；用户管理增删改查，状态变更；文件上传、下载、删除；订单管理增删改查，状态变更；定时任务：发送就诊通知、更新订单状态；查看统计分析数据（医院、用户、订单），测试结果见表 5.1。

表 5.1 后台管理测试结果

测试项	测试结果
测试管理员能否通过正确的账号密码登录系统。	通过
测试错误的账号密码组合是否被拒绝登录，并显示相应提示。	通过
测试管理员登出功能是否正常，登出后是否回到登录页面。	通过
测试首页是否能展示预期的各项数据统计，包括但不限于用户数、医院数、订单数等。	通过
测试数据是否实时更新。	通过
测试数据字典项是否能正确展示。	通过
测试导入功能是否能正确读取 Excel 文件，并更新数据字典。	通过
测试导出功能是否能正确生成并下载包含所有数据字典项的 Excel 文件。	通过
测试是否能成功添加、编辑、删除医院信息。	通过
测试医院信息查询功能是否能准确返回搜索结果。	通过
测试医院状态变更功能（如启用/禁用）是否正常工作。	通过
测试是否能成功添加、编辑、删除用户信息。	通过
测试用户信息查询功能是否能准确返回搜索结果。	通过
测试用户状态变更功能（如启用/禁用）是否正常工作。	通过
测试文件上传功能是否支持各种格式文件，并正确保存到服务器。	通过
测试文件下载功能是否能正确提供下载链接，并保持文件完整性。	通过
测试文件删除功能是否能从服务器删除指定文件。	通过
测试是否能成功创建、编辑、删除订单。	通过
测试订单查询功能是否能准确返回搜索结果。	通过
测试订单状态变更功能（如已支付、已取消）是否正常工作。	通过

测试系统是否能定时发送就诊通知给用户。	通过
测试系统是否能定时更新订单状态。	通过
测试统计分析页面是否能正确显示医院、用户、订单的统计数据。	通过
测试各项数据是否能按预期的条件进行筛选和分析。	通过

5.3 前端门户测试

5.3.1 前台门户界面展示

根据前面的介绍，这部分实现了包括医院、科室排班的查询，用户登录、实名认证，添加就诊人挂号支付等功能，演示见图 5-8 至 5-15。



图 5-8 门户首页

“医来”预约挂号统一平台

Q 点击输入医院名称

搜索

帮助中心

登录/注册

预约挂号

医院8999 | 专科

多发性硬化专科门诊

医院详情

2024-04-11 — 2024-04-17

预约挂号

2024-04-11 周四

2024-04-12 周五

2024-04-13 周六

2024-04-14 周日

2024-04-15 周一

2024-04-16 周二

2024-04-17 周三

41 / 133

16 / 103

33 / 73

88 / 155

44 / 127

46 / 117

52 / 70

停诊信息

< 1 >

查询/取消

上午号源

主任医师 | 贾秀兰

专门处理心脏的疾病，如冠心病、心肌炎等

¥9.67

剩余 25

下午号源

副主任医师 | 高军

专门处理肾脏的疾病，如胃炎、肾结石等

¥7.93

剩余 63

京ICP备00000000号 电话挂号000-00000000 我的主页 项目首页 问题反馈 加入我们

图 5-9 排班展示

“医来”预约挂号统一平台

Q 点击输入医院名称

搜索

帮助中心

登录/注册

预约挂号

医院详情

预约挂号

停诊信息

查询/取消

验证码已发送至

598940 46s

马上登录

第三方账号登录

微信扫码—扫关注
“快速预约挂号”

扫一扫下载
“预约挂号”APP

xxxxxx 官方指定平台
快速挂号 安全放心

副主任医师 | 高军

专门处理肾脏的疾病，如胃炎、肾结石等

¥7.93

剩余 63

京ICP备00000000号 电话挂号000-00000000 我的主页 项目首页 问题反馈 加入我们

图 5-10 用户登录（手机号验证码）

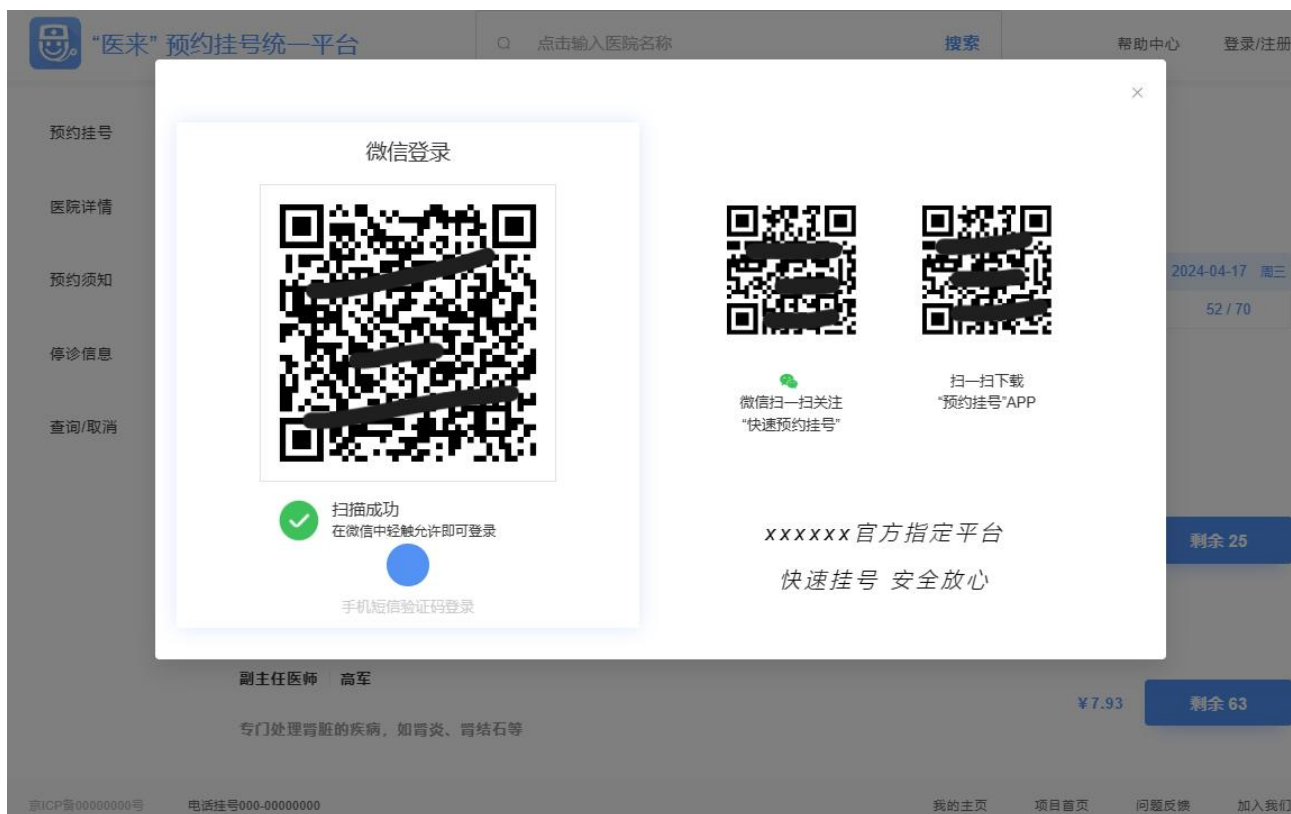


图 5-11 用户登录（微信扫码）



图 5-12 实名认证

“医来”预约挂号统一平台

Q

点击输入医院名称

搜索

帮助中心

通晓宇宙

用户资料

添加就诊人

实名认证

就诊人信息

就诊人管理

挂号订单

问题讨论

姓名:

证件类型:

身份证

证件号码:

6

性别:

男

女

出生日期:

20

手机号码:

建档信息 (完善后部分医院首次就诊不排队建档)

婚姻状况:

未婚

已婚

自费/医保:

自费

医保

医保卡号:

6

联系人信息 (选填)

姓名:

手机号码:

13

保存

京ICP备00000000号

电话挂号000-00000000

我的主页

项目首页

问题反馈

加入我们

图 5-13 添加就诊人

预约挂号

医院详情

预约须知

停诊信息

查询/取消

确认挂号信息

选择就诊人：

身份证
3
X

+

添加就诊人

选择就诊卡： 如您持社保卡就诊，请务必选择医保预约挂号，以保证正常医保报销

身份证

医保 身份证

挂号信息

就诊日期： 2024-04-14 周日 上午

就诊医院： 医院8999

就诊科室： 多发性硬化专科门诊

医生姓名： 贾秀兰

医生职称： 主任医师

医生专长： 专门处理心脏的疾病，如冠心病、心肌炎等

医事服务费： 9.67元

用户信息

就诊人手机号：

图 5-14 确认挂号

37

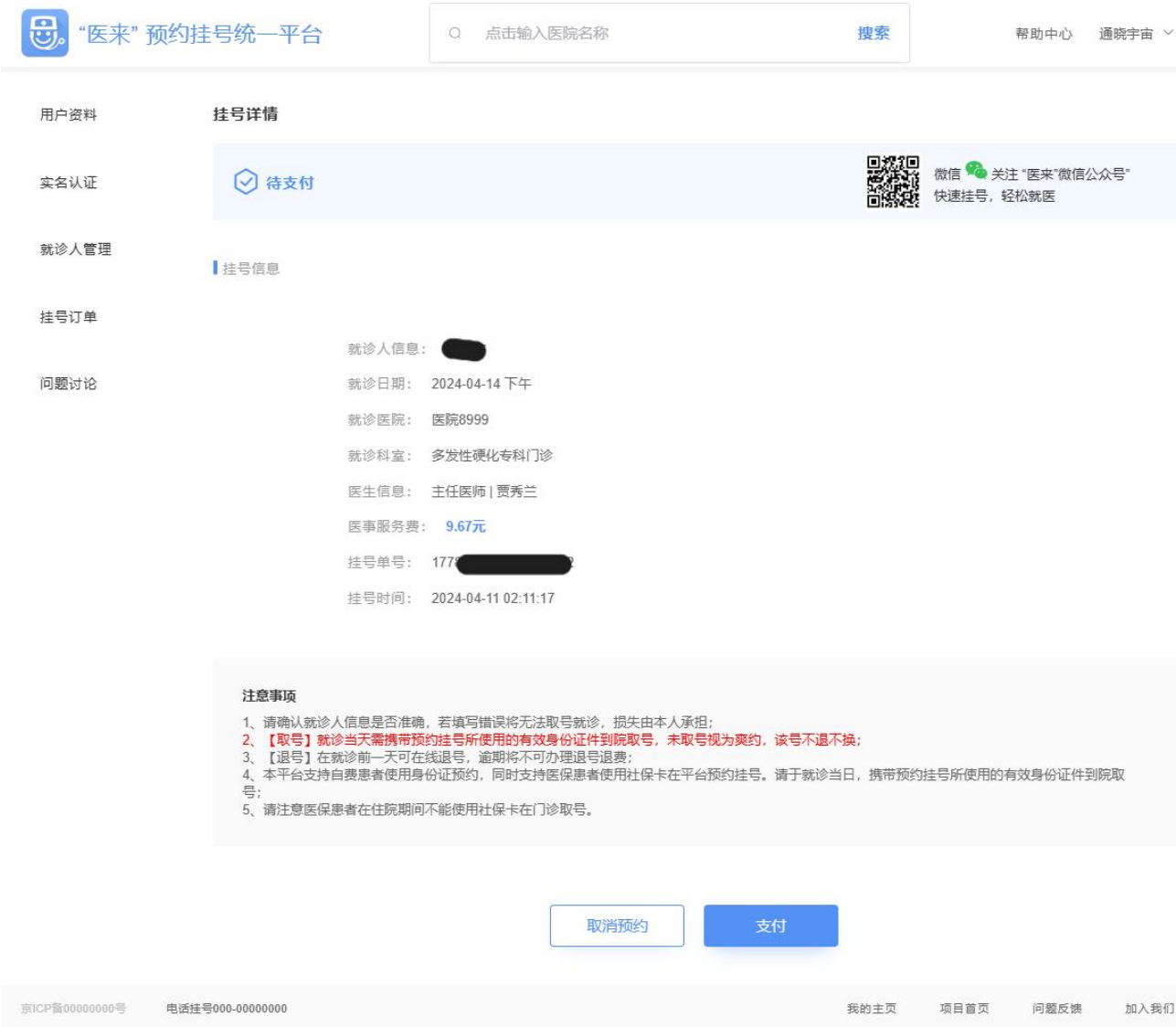


图 5-15 预约成功

5.3.2 前端门户测试

该部分的测试主要包括前台用户登录（手机验证码、微信扫码）、登出；医院列表展示、按等级、地区、名称查询；医院科室、详情展示；科室排班信息展示；预约挂号、创建订单、支付、取消订单；查看、修改个人信息，实名认证；就诊人管理，增删改查；用户讨论区，测试结果见表 5.2.

表 5.2 前台门户测试结果

测试项	测试结果
测试用户能否通过手机验证码登录系统。	通过
测试微信扫码登录功能是否正常，包括扫码、授权、登录流程。	通过
测试用户登出功能是否正常，登出后是否回到登录页面。	通过

测试医院列表是否能正确展示所有医院信息。	通过
测试列表是否支持按医院等级、地区、名称进行筛选查询。	通过
测试选定医院后是否能正确显示该医院的所有科室信息。	通过
测试医院详情页面是否包含完整的医院信息，包括地址、联系方式等。	通过
测试管理员登出功能是否正常，登出后是否回到登录页面。	通过
测试选定科室后是否能正确显示该科室的排班信息，包括医生、排班时间等。	通过
测试用户能否成功进行挂号预约，并创建订单。	通过
测试支付功能是否正常，包括支付流程和支付状态更新。	通过
测试用户能否在规定条件下取消订单，并正确更新订单状态。	通过
测试用户能否查看自己的个人信息。	通过
测试用户能否更新自己的个人信息，包括昵称、手机号等。	通过
测试实名认证功能是否正常，包括提交实名信息和认证状态更新。	通过
测试用户能否成功添加、编辑、删除就诊人信息。	通过
测试用户能否查询到已添加的就诊人信息列表。	通过
测试用户能否在讨论区发帖、回帖。	通过
测试帖子和回复是否按时间顺序正确显示。	通过

6 总结与展望

6.1 总结

本项目旨在满足当前社会中医院和患者日渐增长的在线预约挂号需求，利用 Spring 技术栈，实现了一个功能全面的网上预约挂号系统。该系统通过细致的需求分析、精心设计的体系结构与功能模块、合理的数据库设计及高效的系统实现，配合严格的系统测试，最终成功投入运行。该系统实现了网上预约挂号的核心功能，有效缓解了患者预约挂号难、就医难的问题；同时优化了就医流程，并为医院提供了一个高效的管理服务平台，有效提升了医疗资源的利用率，对患者和医院均产生了实质性的经济和社会效益。

项目在研发和实施中取得的主要成果概括如下：

- 采用基于 Spring Cloud + Alibaba 的微服务架构设计，改善了系统的内部聚合度和外部耦合度，提高了系统的可维护性和扩展性。
- 引入 Spring Cloud Gateway 作服务网关，提升了系统的并发处理能力和稳定性，满足了高并发场景需求。
- 系统包含数据字典、订单管理、短信服务、用户管理和业务日志记录等多个功能模块，全面满足了医院和患者的日常需求，同时支持多种支付和登录方式，提升了用户体验。
- 通过全面的功能和性能测试，确保系统能在实际运营中稳定可靠地服务于多家医院和广大患者，系统响应时间和性能均达到设计要求。
- 系统展现出强大的可扩展性和可扩充性，能够适应未来医疗服务需求的增长。

本项目的设计与开发过程中，详细的需求分析确保了系统设计的完备性和实用性；高效的架构设计和技术实现保证了系统的性能和稳定性。通过对用户管理、订单管理、短信服务等核心功能的综合测试，以及系统前端和后端界面的用户体验评估，不断优化系统功能和操作流程，最终打造出一款高质量的医疗服务平台。此外，完善的项目部署和测试流程也为系统的顺利上线和稳定运行提供了有力保障。随着系统的不断完善与优化，预计将能更好地服务于社会，为医院和患者之间架起更加便捷、高效的沟通桥梁。

6.2 展望

尽管本系统满足了患者对在线预约挂号系统的基本需求，但随着技术进步和用户需求的不断变化，还有许多潜在的改进和扩展空间。未来的发展方向可包括如下几个方面：

- 版本升级：系统可定期进行功能更新和优化，以提供更加稳定和高效的用户体验。

- **Elasticsearch 全文检索：**为了提高医院、科室和医生信息的检索速度和准确性，计划集成 Elasticsearch 作为系统的全文搜索引擎。这将使用户能够快速找到所需的服务和信息^[16]。
- **Spring Security 权限控制：**为了增强系统的安全性，可通过引入 Spring Security 进行细粒度的权限控制。通过角色管理、访问控制等机制，确保数据安全和用户隐私^[17]。
- **K8S 集群管理：**随着系统用户量的增加，服务的稳定性和可扩展性将成为关键。系统可通过 Kubernetes (K8S) 进行容器化部署和集群管理，将有助于实现系统的自动化部署、扩展和管理，提高系统的容错能力^[18]。
- **移动端 APP 开发：**考虑到用户移动端的便捷性，未来将开发适用于 iOS 和 Android 平台的移动端应用程序，提供更加丰富和便利的服务^[19]。
- **小程序开发：**为了进一步扩大用户群和提高服务的覆盖率，将开发微信小程序和支付宝小程序，让用户能够在常用社交平台上直接访问和使用我们的服务^[20]。

本系统期望不仅能够持续满足用户和医院不断变化的新需求，而且能够树立行业标杆，引领医疗预约挂号系统向更高水平发展。

参考文献

- [1]瞿晓琦,张晶,多吉加布.“互联网+”背景下医院信息化建设研究[J].信息系统工程,2024,(04):91-94.
- [2]曹璇,苏瑞芳,祝贺,等.住院患者出院随访系统的设计与实现[J].科技风,2023,(36):4-6.DOI:10.19392/j.cnki.1671-7341.202336002.
- [3]Łukasz W,Łukasz L,Małgorzata A K.A Comparative Assessment of JVM Frameworks to Develop Microservices[J].Applied Sciences,2023,13(3):1343-1343.
- [4]Gyorödi C,Gyorödi R,Olah A I, et al.A Comparative Study Between the Capabilities of MySQL Vs. MongoDB as a Back-End for an Online Platform[J].International Journal of Advanced Computer Science and Applications (IJACSA),2016,7(11):
- [5]Bucko A,Vishi K,Krasniqi B, et al.Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History[J].Computers,2023,12(4):
- [6]Ran L,Jiaqi Y,Huibiao Z, et al.Verification of RabbitMQ with Kerberos Using Timed Automata[J].Mobile Networks and Applications,2022,27(5):2049-2067.
- [7]闫燕,张旭亮,杨海申,等.基于云架构的物探基础数据库服务端开发及应用[C]//中国石油学会石油物探专业委员会.第二届中国石油物探学术年会论文集(下册).东方地球物理公司采集技术中心;河北省地震勘探数据采集技术创新中心;,2024:4.DOI:10.26914/c.cnkihy.2024.001536.
- [8]王魁生,杨柳.基于微服务的智能制造仓储系统研究与应用[J].电脑知识与技术,2024,20(06):68-70.DOI:10.14004/j.cnki.ckt.2024.0319.
- [9]刘淑敏.山西IPTV关于微服务应用架构的探索[J].广播电视信息,2024,31(03):46-49.DOI:10.16045/j.cnki.rti.2024.03.013.
- [10]Kotturu P,Kammula P S,Bunga S S, et al.Prevention of Session Hijacking and Authentication Providing to the Session Cookie[J].International Journal of Recent Technology and Engineering (IJRTE),2019,8(4):9685-9690.
- [11]郭玉龙,张瑜,王硕.基于微服务组件化的未来校园智慧服务平台[J].电脑编程技巧与维护,2023,(11):82-84+104.DOI:10.16184/j.cnki.comprg.2023.11.045.
- [12]李蓓.一个电子数据取证培训系统的设计与实现[J].电脑知识与技术,2022,18(09):27-29+35.DOI:10.14004/j.cnki.ckt.2022.0469.
- [13]张曲.存算分离数据库云端运维工具的设计与实现[D].华中科技大学,2022.DOI:10.27157/d.cnki.ghzku.2022.006526.
- [14]罗之宇.基于医疗知识图谱的检索算法研究[D].浙江科技大

学,2024.DOI:10.27840/d.cnki.gzjkj.2024.000032.

[15]周傲英.互联网时代分布式关系数据库的有益探索[J].计算机研究与发展,2024,61(03):539.

[16]曹勤琪,于泓涛,梁振.基于 Elasticsearch 的挂号系统设计与实现[J].中国医学装备,2024,21(02):109-113.

[17]杜成龙,曹海平,吴亮.基于 Spring Security 的自定义用户认证和授权策略研究[J].信息系统工程,2022,(09):75-78.

[18]周微.福建农业银行数据仓库元数据管理系统设计与实现[D].电子科技大学,2014.

[19]贺媛.健康传播视域下“微医APP”的内容分析[D].兰州财经大学,2023.DOI:10.27732/d.cnki.gnzsx.2023.000422.

[20]税俊洁,王黎光.基于微信小程序的医院预约挂号系统的设计与实现[J].电脑编程技巧与维护,2023,(10):64-67.DOI:10.16184/j.cnki.comprg.2023.10.033.

致谢

在这篇论文完成之际，我怀着无比感激的心情向所有在此过程中给予我帮助和支持的人们致以最诚挚的谢意。

首先，衷心感谢我的指导老师。您的学识渊博和严谨治学的态度对我影响深远。从选题到论文写作的每一个环节，您都给予了细致的指导和耐心的帮助，使我在学术道路上不断进步。您的鼓励和支持是我克服困难、不断前行的最大动力。

同时，我要特别感谢我的辅导员。您在我整个大学期间给予了无微不至的关怀和支持，不仅在学术上提供了指导，也在生活上给予了许多帮助。您的关怀让我在学业和生活中充满了信心和勇气。

我要感谢我的家人。感谢您们一直以来对我的理解和支持。无论我遇到多大的挑战和困难，您们始终是我坚强的后盾。您的关爱和鼓励让我在学术道路上勇往直前。

也要感谢我的朋友们。感谢您们在我感到迷茫和疲惫时给予的陪伴和鼓励。您的理解和支持让我在前进的路上不再孤单。

此外，我还要感谢开源社区的所有贡献者。通过参与开源项目和使用开源资源，我学到了很多实用的技术和知识，这对我的研究工作起到了重要的帮助作用。您的无私奉献让我深受感动，也激励我在未来的工作中继续回馈社区。

这篇论文的完成离不开所有人的支持和帮助，再次向您们致以最诚挚的谢意！