浙江大學
ZHEJIANG UNIVERSITY

软件工程课程作业

设计报告

姓名 黄鸿宇 沈小康 周炜

学号 3210105703 3210103818 3210103790

院所 计算机科学与技术学院

2024 年 5 月 14 日

目录

1 引言

1.1 编写目的

在本阶段，我们已经完成了医疗预约管理系统中评价子模块的大致设计。本概要设计说明书的编写旨在进一步细化软件设计阶段的成果，将软件概貌加工成与源程序开发非常接近的软件标识。本文档的目标读者包括软件测试人员、程序开发员和软件分析员。随着信息技术的快速发展，公众对医疗服务的便捷性和效率提出了更高的要求。为满足这些需求，我们设计了一个综合性的医疗预约管理系统。该系统集成了在线预约、远程问诊、账单管理和用户反馈等功能，旨在显著提升医疗服务质量和患者体验。本文档详细阐述了系统的功能、性能和用户界面需求，以确保项目团队和管理者对系统需求有统一的理解，并指导后续的设计、开发、测试和维护工作。我们的医疗预约管理系统致力于为病人打造一个全面的医疗服务平台。系统的主要功能包括：

用户注册与登录：允许病人注册账户并登录，以便安全、便捷地使用系统服务。个人医疗信息管理：用户可以查看、管理和更新个人医疗信息，确保信息的准确性和时效性。AI 病情咨询服务：提供 AI 技术支持的病情咨询服务，为用户给出初步建议。科室与医生信息介绍：展示详细的科室和医生团队信息，帮助用户做出合适的选择。医生预约与时段选择：用户可以根据医生的可选时段进行预约，提高就诊便利性。医疗费用账单管理：用户可以在线提交和查看医疗费用账单，便于费用核对。缴费与退费：支持在线缴费和退费，简化费用处理流程。电子处方查询：用户可以在线查询医生开具的处方信息。电子病历搜索与访问：用户可以搜索和查看自己的电子病历记录。预约挂号与问诊：用户可以预约挂号，并在预约时间进行问诊。个性化预约建议：系统根据用户时间安排提供预约建议，满足个性化需求。电子问诊单与后续跟进：问诊后，用户可以接收电子问诊单，便于后续管理。医疗服务评价：用户可以评价医生和医院服务，帮助改进服务质量。处方与病历综合查询：用户可以查询处方并管理电子病历，全面了解健康状况。本报告作为设计模式报告，详细描述了系统的核心功能模块，并定义了各模块之间的交互方式和数据流向，确保了系统设计的一致性和完整性。报告中的蓝图覆盖了用户界面设计到后端逻辑处理的各个方面，旨在确保系统的易用性和技术实现的可行性。用户界面设计注重提供直观、友好的操作体验，而后端逻辑则着重于保障数据处理的准确性和安全性。报告还规范了系统的架构设计，包括技术选型、数据库设计、API 设计等关键技术点，为开发团队提供了明确的技术指导。在规范方面，报告强调了系统性能的要求，如响应时间、并发处理能力和系统稳定性等，确保系统在高负载情况下仍能保持流畅运行。同时，报告还提出了系统的安全性要求，包括数据加密、用户认证和访问控制等，以保护用户信息和系统安全。此外，报告还涉及了系统的可维护性和可扩展性，指导如何进行系统维护和更新，以及如何根据未来需求的变化对系统进行扩展和升级。这包括了代码的模块化设计、文档的完整性以及版本控制的最佳实践。通过这些措施，我们确保了系统的长期可持续发展，以适应未来医疗服务领域的变化和需求。

1.2 项目描述

在数字化和网络化技术迅速发展的背景下，中国的医疗行业正在经历一场重大的变革。尽管一些大型医疗机构已经部署了在线预约系统，但这些系统大多仅限于机构内部使用，未能实现不同医疗机构间的互联互通。同时，中小医疗机构由于技术和资金的限制，在线服务的普及仍然有待提高。鉴于此，开发一个全面且一体化的医疗预约管理系统显得尤为迫切，这不仅能够提高医疗服务的效率，还能促进医疗资源的共享。国内外众多在线预约诊疗服务平台的出现，为患者提供了一站式的便捷服务，包括注册、登录、查看个人信息、AI 病情咨询、科室浏览、医生预约、账单提交与缴费等。此外，患者还能参与问诊评价体系，为医疗服务提供反馈，帮助提升服务质量。在现代社会的快节奏生活中，公众对医疗服务的需求日益增长，传统的电话预约和现场排队方式已逐渐无法满足当前的需求。医疗预约管理系统利用互联网技术，使用户能够随时随地进行医疗服务预约，从而提高服务效率，减少等待时间，并改善用户体验。本项目的目标是开发一款全面的医疗预约管理系统，为患者提供一个便捷、高效的在线医疗服务平台。该系统将支持患者进行注册登录、查看个人信息、接收 AI 技术提供的病情咨询服务，以及在线查看医院科室信息和预约合适的医生，从而优化就诊流程。随着信息化时代的到来，人们对医疗服务的便捷性和个性化要求不断提升。因此，我们的系统设计特别注重用户体验，提供人性化的操作方式和多样化的功能，以满足不同患者的需求。展望未来，我们计划为系统扩展更多高级功能，如接入健康监测数据（例如心率、血压等）、提供个性化健康建议、支持语音输入创建事件等，以进一步提升医疗服务的质量和效率。

1.3 功能需求

本系统旨在通过一系列精心设计的核心功能，为患者提供一个全面而高效的医疗服务体验。以下是系统的主要功能需求：

- 用户注册与登录：病人可以注册账户并登录，以便安全、便捷地使用系统服务。
- 个人医疗信息管理：用户可以查看、管理和更新个人医疗信息，确保信息的准确性和时效性。
- AI 病情咨询服务：用户可以通过 AI 技术获得关于自己病情的初步建议。
- 科室与医生信息介绍：系统提供详细的科室和医生团队信息，帮助用户选择合适的科室和医生。
- 医生预约与时段选择：用户可以查看医生的可选时段并进行预约，提高就诊便利性。
- 医疗费用账单管理：用户可以在线提交和查看医疗费用账单，便于费用核对。
- 缴费与退费：系统支持在线缴费和退费，简化费用处理流程。
- 电子处方查询：用户可以在线查询医生开具的处方信息。
- 电子病历搜索与访问：用户可以搜索和查看自己的电子病历记录。

- 预约挂号与问诊：用户可以预约挂号，并在预约时间进行问诊。
- 个性化预约建议：系统根据用户时间安排提供预约建议，满足个性化需求。
- 电子问诊单与后续跟进：问诊后，用户可以接收电子问诊单，便于后续管理。
- 医疗服务评价：用户可以评价医生和医院服务，帮助改进服务质量。
- 处方与病历综合查询：用户可以查询处方并管理电子病历，全面了解健康状况。

通过这些功能的实现，本系统将极大地提升医疗服务的可及性和效率，确保用户能够享受到高质量的医疗服务体验。这不仅能够提高患者的满意度，还能促进医疗服务质量的整体提升和持续改进。

2 系统假设

为了确保医疗预约管理系统能够有效地服务于病人，我们基于以下假设进行系统设计：

- **用户能力假设**: 假设所有使用本系统的病人均具备操作智能手机或计算机的基本技能，并且有明确的需求进行医疗服务的预约和咨询。
- **技术环境假设**: 假设服务器配置能够满足系统运行的最低要求，包括操作系统、网络环境和必要的软件支持。同时，服务器安全性良好，能够抵御外部攻击，保证系统稳定运行。
- **网络依赖假设**: 虽然基本的医疗服务预约功能不依赖网络连接，但是部分高级功能，如在线支付和电子问诊单的接收，需要稳定的网络连接。特别是地点提醒功能，完全依赖于定位服务，因此需要一个流畅的网络环境来支持。
- **数据准确性假设**: 在系统运行过程中，依赖的第三方 API 和地图定位服务提供的数据是准确和可靠的。这确保了系统能够根据准确的数据为病人提供服务。

在开发医疗预约管理系统的过程中，我们遵循以下八项约束条件，以确保系统的稳定性、安全性、高效性及用户友好性。

通过遵循这些约束条件，我们的医疗预约管理系统将能够为病人提供一个全面、便捷的医疗服务体验，同时确保系统的长期稳定运行和用户数据的安全。

为了确保医疗预约管理系统能够有效地服务于病人，我们的系统设计和实现基于以下假设和术语定义：

通过这些术语的明确定义，我们希望病人能够更加顺畅地使用医疗预约管理系统，享受到全面、便捷的医疗服务体验。系统的设计旨在提高医疗服务的可及性和效率，简化病人的医疗服务流程，提升整体医疗服务质量。

约束项	描述
数据存储约束	系统后端采用标准化的 MySQL 数据库作为主要的数据存储解决方案，确保数据的持久化、一致性和安全性。实施定期备份和灾难恢复计划。
网络服务吞吐约束	系统设计考虑了高并发用户访问，确保网络服务具备足够的吞吐量，提供快速响应的用户体验。
数据安全约束	采取包括数据加密、访问控制和安全审计在内的多层次安全措施，保障用户数据的完整性、保密性和可用性。
性能要求约束	系统应能在各种设备上快速加载，提供流畅的用户体验，包括快速的页面响应时间和高效的数据处理能力。
用户界面约束	界面设计简洁直观，易于导航，确保所有用户群体都能轻松使用系统的各项功能。
兼容性约束	系统应在主流的操作系统和浏览器上运行良好，无需特殊配置即可访问所有功能。
可扩展性约束	系统架构设计应具备良好的可扩展性，便于未来增加新功能或升级现有功能，以适应不断变化的医疗需求。
法规遵从性约束	系统开发和运营需遵守所有相关的医疗保健法规和隐私政策，确保病人信息的合法处理和保护。
灾难恢复约束	系统应具备完善的灾难恢复计划和定期测试机制，确保在任何突发情况下系统的连续性和数据的完整性。

表 1: 医疗预约管理系统设计与实现的约束条件

术语	详细描述
医疗预约系统	一个综合性的在线服务平台，旨在为病人提供便捷的医疗服务预约体验。它允许用户远程预约挂号、查询医疗费用、查看电子问诊单据、评价医疗服务质量，并通过数据分析优化医疗资源分配。
注册登录	病人在使用医疗预约管理系统前必须进行的账户创建和身份验证过程。这确保了用户信息的安全性和隐私保护，同时为用户提供个性化的医疗服务。
AI 咨询	利用先进的人工智能技术，系统提供初步的病情分析和健康建议服务。AI 咨询能够根据病人提供的症状信息，给出可能的疾病诊断和建议的下一步行动。
科室浏览	系统提供的一个功能，允许病人查看医院内不同科室的详细信息，包括科室的专业领域、医生团队介绍和特色服务，以便病人能够根据自身需求选择合适的医疗服务。
预约挂号	病人可以通过系统选择心仪的医生和方便的时段进行预约。此功能通过智能排队和时间管理机制，最大化地减少病人的等待时间，提高就诊效率。
账单管理	一个集成在系统中的功能，使病人能够轻松查询、提交和支付医疗费用账单。账单管理功能支持多种支付方式，并提供详细的费用明细，以便病人了解费用构成。
电子问诊单	问诊结束后，病人将收到一份包含诊断结果、治疗建议和处方信息的电子文档。电子问诊单便于病人随时查看和保存，同时也为医生后续的跟踪治疗提供了便利。
问诊评价体系	医疗预约管理系统内置的评价机制，允许病人对接受的医疗服务进行评价。这些评价不仅为其他病人提供参考，也为医疗机构提供了改进服务质量的宝贵反馈。
处方查询	系统提供的一项功能，使病人能够在线查看医生开具的处方详情，包括药物名称、用法用量等。处方查询功能确保病人能够准确理解医嘱，并按需购买药品。
病历搜索	病人可以通过系统搜索并访问自己的历史医疗记录和病历资料。这项功能对于病人了解自己的健康状况、跟踪疾病进展和预防措施具有重要意义。

表 2: 医疗预约管理系统关键术语表

3 系统体系分析

本项目团队致力于开发一个综合性的医疗预约管理系统，该系统以事件为中心，旨在为患者提供全面而便捷的医疗服务体验。系统架构采用了经典的三层结构设计，包括表示层、业务逻辑层和数据访问层，每一层都承载特定的功能和责任，确保了系统的清晰性和可维护性。

在进行系统架构设计时，我们考虑了多种体系结构风格，并最终确定了最适合我们项目需求的架构。以下是我们参考的体系结构风格及其特点：

- **以数据为中心的体系结构**：这种风格专注于数据的组织和管理，适用于数据存储和处理至关重要的系统。它包括数据存储、数据访问层、业务逻辑层和表示层。

- **数据流体系结构**：专注于数据的移动和转换，通过一系列处理步骤的数据流来组织系统，适用于需要实时处理或分析的系统。

- **调用和返回体系结构**：也称为面向过程的架构，通过函数调用和返回进行通信，适用于过程编程语言。

- **面向对象体系结构**：围绕对象组织系统，对象是封装数据和行为的类的实例，适用于面向对象的编程语言。

- **分层次体系结构**：将系统组织成层，每一层为其上层提供服务并使用其下层的服务，适用于企业和 Web 应用程序。

- **客户端-服务器架构**：将系统分为处理用户界面的客户端和处理业务逻辑的服务器端，适用于网络应用程序。

- **浏览器-服务器架构**：一种特殊的客户端-服务器架构，客户端运行在用户 Web 浏览器中，适用于 Web 应用程序。

- **微服务架构**：将应用程序构建为小型独立服务的集合，每个服务都专注于特定的业务功能，适用于需要高度可扩展性和灵活性的系统。

- **事件驱动架构**：强调系统内发生的事件的产生、检测、消费和反应，适用于需要实时处理的应用程序。

我们的医疗预约管理系统采用了分层次体系结构，其中：

- **表示层**：负责与用户交互，展示信息和接收输入。

- **业务逻辑层**：处理用户请求，执行业务规则和逻辑。

- **数据访问层**：作为系统与数据库之间的桥梁，负责数据的存储和检索。

这种架构风格允许我们实现关注点分离，使得开发、测试和维护更加高效。同时，我们也考虑到了系统的可扩展性和未来的技术发展，确保系统能够适应不断变化的需求。

通过精心设计的架构和功能，我们的医疗预约管理系统将提供一个无缝、高效的医疗服务体验，提高医疗服务的可及性和效率，确保用户享受到高质量的医疗服务。这不仅将提升病人的满意度，也将促进医疗服务的整体改进和发展。

3.1 以数据为中心的体系结构

以数据为中心的体系结构是一种基于数据处理和数据存储的 IT 架构。它将数据的收集、处理、存储和使用放在整个系统的核心位置，从而使得数据成为整个系统的主要驱动力。以

数据为中心的体系结构是一种将数据处理和存储作为系统核心的设计方法。这种架构风格特别适用于那些以数据密集型操作为主的应用场景，如数据库系统、数据仓库、商业智能和大数据分析等。在这样的系统中，数据的组织、管理和优化是最为关键的考量因素。

在以数据为中心的体系结构中，通常包含以下几个关键组件：

1. **数据存储**：作为架构的核心，负责所有数据的存储和管理。这通常涉及到关系型或非关系型数据库、文件系统或其他形式的数据存储解决方案。数据存储的设计要确保数据的完整性、安全性和高可用性。

2. **数据访问层**：提供与数据存储交互的接口。这一层可能包含数据访问对象（DAO）、数据映射器或对象关系映射（ORM）框架等，以简化对数据存储的访问和操作。此外，数据访问层还可能负责处理缓存机制、数据库连接管理和事务控制。

3. **业务逻辑层**：处理与数据相关的业务规则和逻辑。这一层包含了实现系统功能的服务类、控制器和域对象等。业务逻辑层与数据访问层紧密交互，执行数据的检索、处理和存储。

4. **表示层**：负责将数据呈现给用户，并处理用户的输入。这可以包括图形用户界面、网页界面或 API 接口等。表示层与业务逻辑层交互，以展示数据和响应用户操作。

以数据为中心的架构强调了数据模型的设计质量，它可以带来一系列的好处，包括提高数据质量管理、优化系统性能以及简化系统的维护和演进。然而，这种架构也存在一些挑战，比如可能会增加系统的复杂性，并且需要专业的技能和工具来支持数据存储和处理的需求。

在医疗预约管理系统中，采用以数据为中心的体系结构可以确保患者信息、预约记录、医疗资源等关键数据的安全性和准确性。同时，通过精心设计的数据模型和访问层，可以提高系统的查询效率和响应速度，从而提升用户体验。尽管这可能需要更多的前期设计工作和专业技能，但从长远来看，它能够为系统提供一个坚实的数据基础，支持未来的扩展和维护。

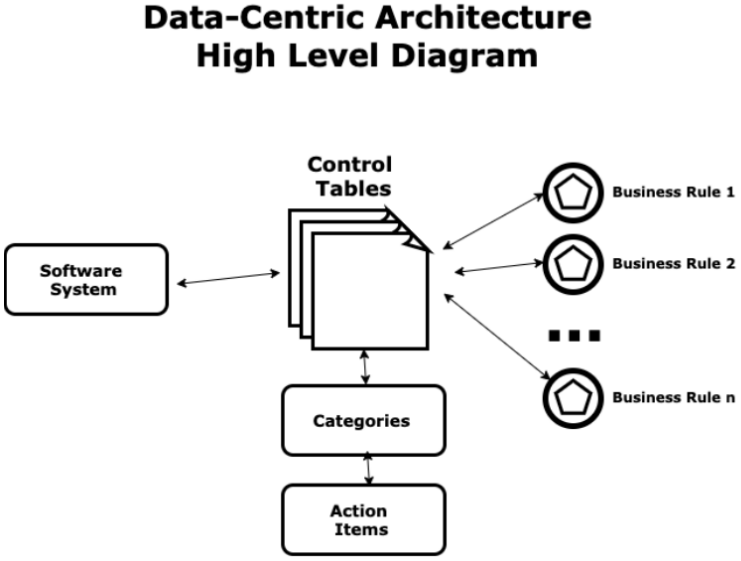

图 1: 以数据为中心的体系结构

3.2 数据流体系结构

数据流体系结构是一种特别关注于数据在软件系统中流动和转换方式的架构风格。与将系统划分为离散层次或组件的传统方法不同，数据流体系结构通过一系列连续的处理步骤来组织数据的流动。

在数据流体系结构中，以下是几个关键组件：

来源（Source）：这是系统接收初始输入数据的地方。来源可以是传感器、输入设备或其他数据源，负责提供数据流的起点。

处理器（Processor）：处理器对来自来源的数据执行各种处理步骤。这可能包括数据转换、数据清洗、数据聚合或应用机器学习算法等。

下沉（Sink）：下沉组件接收处理器的输出数据，并将其作为最终结果提供给用户或其他下游系统。下沉可以是输出设备、数据库或另一个系统，负责数据流的终止。

连接器（Connector）：连接器组件负责在来源、处理器和下沉之间提供通信和协调机制。这可能包括消息队列、事件中心、RESTful API 或其他中间件，确保数据能够顺畅地在系统中流动。

数据流体系结构在需要处理大量数据流，尤其是需要实时处理和分析的系统中特别有用。例如，在金融交易、医疗保健监控、物联网设备数据处理等领域，这种架构能够提供高性能、可扩展性和灵活性。

数据流体系结构的优点包括：

1. **性能改进**：通过优化数据处理步骤，系统可以更高效地处理大量数据。
2. **可扩展性**：系统可以通过增加处理器或调整数据流来轻松扩展。
3. **灵活性**：数据流可以根据需要灵活地调整或重新配置。

然而，这种架构也带来了一些挑战：

设计复杂性：设计一个高效的数据流可能非常复杂，需要仔细规划和考虑。调试难度：由于数据在多个处理步骤中流动，调试和识别问题来源可能比较困难。专业技能需求：实现和维护数据流体系结构可能需要特定的专业技能和工具。在医疗预约管理系统中，数据流体系结构可以用来优化患者数据的处理、预约流程的自动化管理以及实时更新医疗资源的状态。通过精心设计的数据流，系统可以实时响应预约变化，提高医疗服务的响应速度和整体效率。

3.3 调用和返回体系结构

调用和返回体系结构，亦称为面向过程的架构，是一种传统的软件系统组织方式。在这种架构风格中，系统由一系列的函数或过程组成，这些函数通过彼此之间的调用和返回来实现通信和数据交换。这种架构在早期编程语言中非常普遍，如 C 语言和 Fortran 语言，至今仍在某些场景下使用。

面向过程的架构包含以下几个关键组件：

1. **过程（Procedures）**：这些是系统的构建块，每个过程负责执行一个或一组特定的任务。过程之间可以通过调用来交互，实现任务的委托和数据的传递。

2. **数据（Data）**：在面向过程的架构中，数据通常以全局变量或局部变量的形式存在，可以在不同的过程间共享。数据通过函数参数传递，或者作为函数的返回值。

3. **控制流（Control Flow）**：系统的执行流程由函数调用的顺序决定。通常，一个 main 函数作为程序的入口点，根据程序的需要调用其他函数来完成任务。

面向过程的架构提供了以下优势：

1. 简单性：对于简单的程序，面向过程的架构易于理解和实现。
2. 模块化：程序被分解为模块化的函数，有助于代码的组织。
3. 易于测试和调试：由于程序被分解为独立的函数，测试和调试相对容易。

然而，面向过程的架构也面临一些挑战：

抽象有限：与面向对象的架构相比，面向过程的架构提供的抽象级别较低。**代码重复**：由于缺乏类和对象的重用机制，代码重复是一个常见问题。**状态管理困难**：在多个过程之间维护和同步状态数据可能会变得复杂。尽管面向过程的架构在某些特定场合下仍然适用，但它在很多情况下已经被更现代的架构风格所取代，如面向对象的架构和事件驱动的架构。这些现代架构提供了更高级的抽象和功能，更适合管理复杂的系统。

在医疗预约管理系统的背景下，面向过程的架构可能适用于一些简单的数据处理任务，但对于整个系统的开发和维护，可能需要考虑采用更先进的架构风格，以支持系统的可扩展性、灵活性和长期的可维护性。

3.4 面向对象体系结构

在软件架构设计中，面向对象体系结构（Object-Oriented Architecture, OOA）是一种以对象为中心的设计风格，它强调了数据和方法的封装、继承和多态性。面向对象的体系结构通常适用于像 Java、C++ 和 Python 这样的面向对象编程语言，其主要组件包括：

- **类（Class）**：定义了对象的数据结构和行为。类是对象的蓝图，可以通过实例化来创建具体的对象。类之间可以通过继承关系共享数据和行为。
- **对象（Object）**：是类的实例，封装了数据和方法。对象之间通过方法调用进行交互，这些方法可以操作对象内部的数据或返回新的数据。
- **封装（Encapsulation）**：是一种将对象的实现细节隐藏起来，同时只暴露一个公共接口的做法。封装有助于保护对象的内部状态不被外部直接修改，从而维护系统的完整性和一致性。
- **继承（Inheritance）**：允许新类从现有类中继承数据和行为，从而实现代码的重用，并能够创建出具有附加或修改行为的新类。
- **多态（Polymorphism）**：允许不同类的对象被视为同一类型的对象，只要它们共享一个公共接口或超类。多态性使得编写的代码能够对不同类型的对象执行相同的操作。

面向对象的体系结构提供了模块化、可重用性和可扩展性等显著优势。此外，面向对象的系统通常更易于测试和调试，并能提供对复杂系统的直观表示。

然而，面向对象的体系结构也带来了一些挑战，如管理类层次结构和依赖关系的复杂性，以及维护封装和信息隐藏的难度。

另一方面，分层体系结构（Layered Architecture）是一种将系统组织成多个层次的软件体系结构风格，每一层为其上层提供服务，同时利用其下层的服务。这种风格在企业和 Web 应用程序中非常常见，其主要组件包括：

- **表示层（Presentation Layer）**：负责向用户展示信息和接收用户输入。这一层可以包括用户界面、网页和应用程序屏幕。
- **应用层（Application Layer）**：处理用户输入，并实现系统的业务逻辑。这一层可以包含控制器、管理器和服务类等组件。
- **数据层（Data Layer）**：负责数据的存储和检索。这一层可以包括数据访问对象（DAO）、存储库和数据库连接器等组件。

每一层通过定义明确的接口或 API 与其上下层进行通信，上层向下层请求服务，下层通过上层定义的 API 提供服务。

分层体系结构的优势在于提供了关注点分离、模块化和可扩展性。可以根据需求添加或删除层，以适应不断变化的业务需求，并且每一层都可以独立测试和维护。

然而，分层架构也带来了一些挑战，如管理依赖关系和层间通信的复杂性，以及由于层与层之间的通信开销可能导致的性能问题。分层架构的变体包括 n 层架构和微服务架构，后者将系统分为小型、独立的服务，这些服务通过 API 相互通信。

在医疗预约管理系统的背景下，分层体系结构可以清晰地分离表示逻辑、业务逻辑和数据访问逻辑，有助于构建一个可维护、可扩展且易于测试的系统。

3.5 客户端-服务器架构

客户端-服务器架构（Client-Server Architecture）是一种常见的软件架构风格，它将系统的功能划分为两个主要部分：客户端和服务器端。这种架构广泛应用于网络应用程序，如 Web 应用程序、数据库系统和电子邮件系统等。

在客户端-服务器架构中，涉及以下几个关键组件：

- **客户端（Client）**：负责向用户展示信息和处理用户输入。客户端可以是桌面计算机、移动设备或 Web 浏览器上运行的应用程序。客户端通常负责处理用户界面的呈现和用户的交云操作。
- **服务器端（Server）**：负责实现系统的业务逻辑、数据存储和检索。服务器端通常包括用服务器端编程语言（如 Java、Python 或 PHP）编写的代码，以及数据库管理系统（DBMS）。
- **通讯层（Communication Layer）**：促进客户端和服务器端之间的通信。这一层通常使用 TCP/IP 或 HTTP(S) 等网络协议，包括 API、Web 服务以及在客户端和服务器端之间交换数据的其他方法。

客户端-服务器架构的优势包括：

- **可伸缩性（Scalability）**：服务器端可以轻松扩展以处理增加的流量和用户请求。
- **容错性（Fault Tolerance）**：系统设计可以实现在部分组件失败时继续运行。
- **模块化（Modularity）**：架构的分离性质允许独立的模块开发和维护，便于系统的扩展和维护。

然而，客户端-服务器架构也面临一些挑战：

- **安全漏洞（Security Vulnerabilities）**：架构需要妥善处理潜在的安全风险，如数据泄露和恶意攻击。
- **通信复杂性（Communication Complexity）**：管理客户端和服务器端之间的通信可能变得复杂，需要考虑网络延迟和带宽限制。
- **性能问题（Performance Issues）**：网络通信开销可能影响系统性能，特别是在高负载情况下。
- **跨平台一致性（Cross-Platform Consistency）**：由于客户端代码在用户的设备上运行，确保不同设备和平台上的性能和行为一致性是一个挑战。

在医疗预约管理系统的背景下，客户端-服务器架构可以提供一个稳定和高效的服务平台，允许用户通过客户端应用程序预约医生、管理个人信息和访问医疗服务。服务器端则集中处理业务逻辑、数据存储和安全性要求。通过精心设计通信层和采用适当的安全措施，可以确保系统的高效运行和数据安全。

3.6 浏览器-服务器架构

浏览器-服务器架构（Browser-Server Architecture, BS Architecture），通常称为 B/S 架构，是一种流行的软件架构模式，特别适用于 Web 应用程序的开发。在这种架构下，系统被划分为在用户 Web 浏览器中运行的客户端和在远程服务器上运行的服务器端。

以下是 BS 架构的主要组件：

- **浏览器（Browser）**：作为客户端，负责向用户展示信息并处理用户输入。浏览器中运行的 HTML、CSS 和 JavaScript 代码负责创建用户界面、处理用户交互并将用户请求发送到服务器端。
- **服务器端（Server）**：负责实现系统的业务逻辑、数据库的存储和检索操作，并生成动态内容发送回客户端。服务器端通常使用如 Java、Python 或 PHP 等服务器端编程语言来实现。
- **通讯层（Communication Layer）**：促进客户端和服务器端之间的通信。这一层通常采用 HTTP(S) 协议，并且可能包括 API、Web 服务以及客户端和服务器端之间交换数据的其他方法。

BS 架构的优点包括：

- **灵活性（Flexibility）**：可以在不影响服务器端的情况下更新客户端代码，反之亦然，使得系统维护和升级更加灵活。
- **可扩展性（Scalability）**：服务器端可以通过增加硬件资源或优化代码来轻松扩展，以处理更多的用户请求和数据流量。
- **易于维护（Maintainability）**：由于客户端和服务器端的分离，使得对任一端的维护和更新都不会影响到另一端。

然而，BS 架构也存在一些挑战：

- **安全漏洞（Security Vulnerabilities）**：需要特别注意保护系统免受如 SQL 注入、跨站脚本攻击（XSS）等网络安全威胁。
- **通信复杂性（Communication Complexity）**：客户端和服务器端之间的通信可能变得复杂，尤其是在处理大量并发请求时。
- **性能问题（Performance Issues）**：网络通信的开销可能影响系统性能，特别是在带宽受限或网络不稳定的情况下。
- **跨平台一致性（Cross-Platform Consistency）**：由于客户端代码在用户的设备上运行，确保在不同浏览器和设备上提供一致的用户体验是一个挑战。

在医疗预约管理系统的背景下，BS 架构可以提供一个用户友好的 Web 界面，允许患者轻松地预约挂号、查看医疗信息和接收健康咨询。服务器端则集中处理所有后端逻辑和数据存储，确保系统的稳定性和安全性。通过采用 BS 架构，医疗系统可以实现高效的资源利用和便捷的用户访问，同时降低系统维护和升级的复杂性。

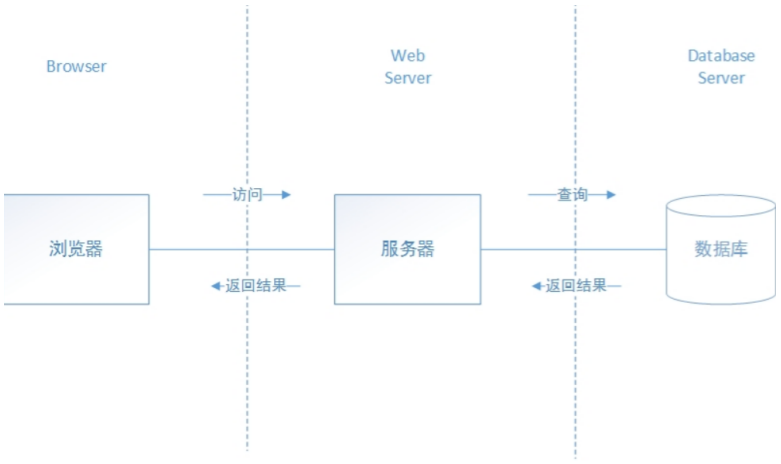

图 2: B/S 体系结构

3.7 微服务架构

微服务架构（Microservices Architecture）是现代软件开发中一种流行的架构风格，它将一个应用程序构建为一系列小型独立服务的集合，每个服务都专注于实现特定的业务功能。这些服务通过轻量级的通信协议，如 HTTP RESTful API 或消息队列（例如 AMQP 或 Kafka）进行交互。

微服务架构的关键特征包括：

- **小型独立服务**：每个微服务都是轻量级的，独立运行，并且专注于单一的业务功能。这种设计使得服务易于开发和维护。
- **去中心化架构**：在微服务架构中，没有中央控制点，每个服务都可以独立部署、升级和扩展。这种去中心化的特性提高了系统的灵活性和可维护性。
- **API 通信**：微服务之间通过定义良好的 API 进行通信，通常采用 RESTful 接口或轻量级的消息传递协议。这有助于实现服务之间的松耦合。
- **领域驱动设计**（DDD）：微服务架构通常与领域驱动设计原则结合使用，强调以业务领域为中心进行服务划分和设计。
- **开发运营文化**：微服务架构的实施通常伴随着 DevOps 文化的采纳，即开发和运营团队之间的紧密合作，以及自动化流程和持续交付的实践。

微服务架构的优势包括：

- **可扩展性**：可以独立地扩展系统中的单个服务，以应对增加的流量或负载。
- **弹性**：由于服务是独立运行的，一个服务的故障不太可能导致整个系统的崩溃。
- **灵活性**：团队可以选择最适合其服务的技术栈，并且可以独立地对服务进行更改和迭代。

然而，微服务架构也带来了一些挑战：

- **复杂性管理**：随着服务数量的增加，管理和协调这些服务的复杂性也随之增加。
- **API 通信开销**：服务之间的通信可能涉及额外的网络延迟和开销。
- **部署和监控**：需要复杂的工具和流程来部署、监控和维护大量的服务。
- **运营成本**：管理多个服务可能会增加运营成本，并且需要更多的 DevOps 实践来优化流程。

在医疗预约管理系统的背景下，微服务架构可以提高系统的可维护性和可扩展性，允许独立地开发和部署各个服务，如用户管理、预约调度、支付处理等。这种架构有助于快速迭代和发布新功能，同时保持系统的高可用性和稳定性。

3.8 事件驱动架构

事件驱动架构（Event-Driven Architecture, EDA）是一种软件架构风格，它侧重于系统中事件的生成、检测、消费和响应。这种架构特别适合于需要实时处理的应用程序，如金融交易、传感器网络、电子商务系统等。

在事件驱动架构中，以下是几个关键组件：

- **事件生产者（Event Producers）**：负责生成事件。这些事件可能源自用户操作、传感器读数或系统内部产生的事件。
- **事件消费者（Event Consumers）**：负责接收和处理事件。处理通常涉及执行特定的操作，如触发业务逻辑、更新数据库或发送通知。
- **事件总线（Event Bus）**：作为事件驱动架构的通信中枢，负责在生产者和消费者之间传递事件。事件总线可以使用如 Kafka、RabbitMQ 等消息队列系统，或 AWS SNS/SQS 等云服务来实现。
- **事件处理器（Event Processors）**：负责处理事件并执行适当的操作，如数据的聚合、转换或路由到特定的消费者。

事件驱动架构的优势包括：

- **可伸缩性（Scalability）**：由于组件的独立性，系统可以根据流量或负载的增加而轻松扩展。
- **模块化（Modularity）**：每个组件负责特定的任务，使得系统更加模块化，易于开发和维护。
- **容错性（Fault Tolerance）**：组件的独立性还意味着一个组件的故障不太可能影响整个系统的运行。
- **技术选择的灵活性（Technological Flexibility）**：团队可以根据需要选择最适合特定组件的技术栈。

然而，事件驱动架构也带来了一些挑战：

- **复杂性管理（Complexity Management）**：随着事件生产者和消费者的增加，管理这些组件的复杂性也随之增加。
- **事件总线开销（Event Bus Overhead）**：维护事件总线可能需要额外的资源和开销。
- **部署和监控（Deployment and Monitoring）**：需要复杂的工具和流程来部署、监控和维护分布式的事件驱动系统。
- **调试和排错（Debugging and Troubleshooting）**：异步事件流可能使得系统调试和排错变得更加困难。

在医疗预约管理系统中，事件驱动架构可以用于实时处理预约请求、支付确认、医生排班变更等事件。例如，当一个预约被创建或取消时，系统可以实时更新数据库、发送通知给相关方，并触发相关的业务流程。这种架构有助于提高系统的响应速度和可靠性，从而提升患者和医疗机构的整体体验。

3.9 本次设计采用的体系结构风格

在本次设计中，评价系统作为医疗预约管理系统的一个重要子模块，是一个基于 Web 的应用程序。因此，我们采用了多种架构风格的组合来构建这一系统：

- **浏览器-服务器（Browser-Server, B/S）架构：**由于评价系统是一个 Web 应用程序，它自然地采用了 B/S 架构。在这种架构中，用户的交互通过 Web 浏览器进行，而所有的业务逻辑和数据存储都在服务器端处理。
- **面向对象体系结构：**系统涉及到用户（包括病人、医生、管理员）和评价等实体，这些实体及其关系是通过对象来表示的。面向对象的方法允许我们通过类和对象来封装数据和行为，提供了一种自然的方式来模拟现实世界中的实体和它们之间的交互。
- **事件驱动架构：**评价系统中的许多活动，如添加评价、删除评价等，都可以被视为事件。系统需要检测这些事件并做出相应的响应。事件驱动架构允许系统组件在事件发生时进行通信和交互，从而实现实时响应。
- **层次体系结构：**我们的系统采用 Spring Boot 框架开发，该框架的层次结构从上至下可以分为五层：View 层、Controller 层、Service 层、Mapper 层（也称为 Dao 层）和 Model 层。每一层都有其特定的职责，且层与层之间通过定义良好的接口进行交互。
  - **View 层：**负责展示数据给用户，并接收用户输入。
  - **Controller 层：**负责处理用户的请求，调用 Service 层的业务逻辑，并准备响应数据。
  - **Service 层：**包含业务逻辑的应用设计和数据库操作的接口设计。
  - **Mapper 层：**负责数据持久化，提供 CRUD 操作。
  - **Model 层：**存放与数据库表字段相对应的实体类。
- **MVC 结构：**由于评价系统强调对评价记录等数据的增删改查，并通过中间件与 UI 界面连接，它也展现了 MVC（Model-View-Controller）结构的特点。Model 层负责数据，View 层负责显示，Controller 层负责逻辑。

综上所述，本次设计采用的 BS 体系架构，同时融合了层次体系架构、面向对象体系架构和事件驱动架构的一些特点。这种混合架构风格为评价系统提供了灵活性、可维护性和可扩展性，同时也确保了系统的实时响应能力和用户交互的直观性。通过这种方式，我们的评价系统能够高效地处理用户评价，提升医疗预约管理系统的整体性能和用户体验。

4 数据端与服务接口

4.1 服务端 UI 设计

大致设计如下图所示，页面中分为了多个模块，以医生个人信息模块为例，点击后即可展示医生个人信息。

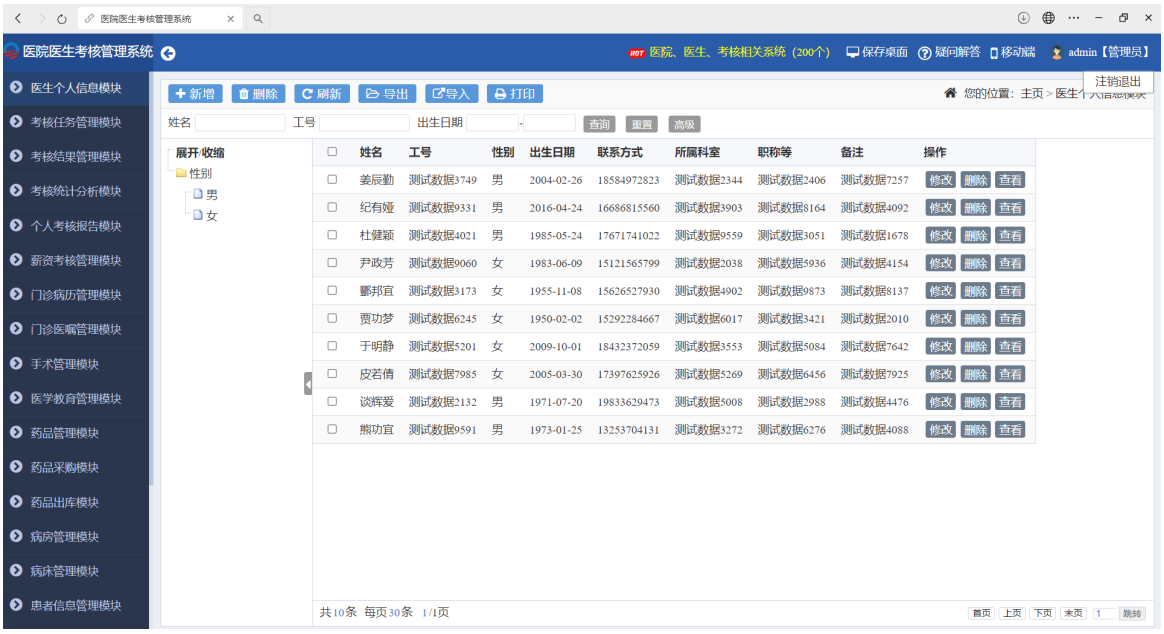

图 3: 服务端 UI 设计

进一步点击可以查看和修改数据，并且支持数据的导出：

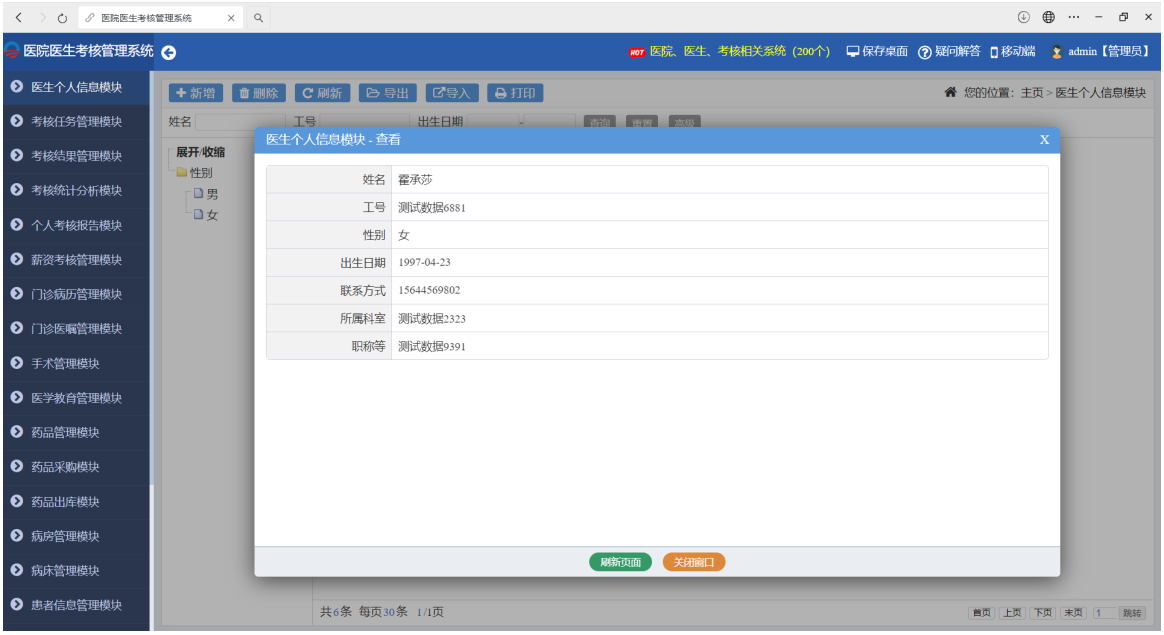

图 4: 查看和修改数据页面展示

其余功能不在一一给出，所有设计的功能见下章节

4.2 服务功能设计

4.2.1 用户注册与登录机制

病人可以通过注册账户并登录系统，以便安全、便捷地使用系统提供的各项服务。

注册	登录
用户访问注册页面，填写必要信息（如用户名、密码、邮箱等），提交注册请求。	用户访问登录页面，输入注册时的用户名和密码。
系统验证用户提供的信息是否符合要求，若符合则完成注册，向用户发送确认邮件。	系统验证用户输入的用户名和密码是否匹配注册时记录的信息。
用户收到确认邮件，点击确认链接完成账户激活。	登录成功后，用户可以访问系统提供的各项服务。

表 3: 用户注册与登录机制

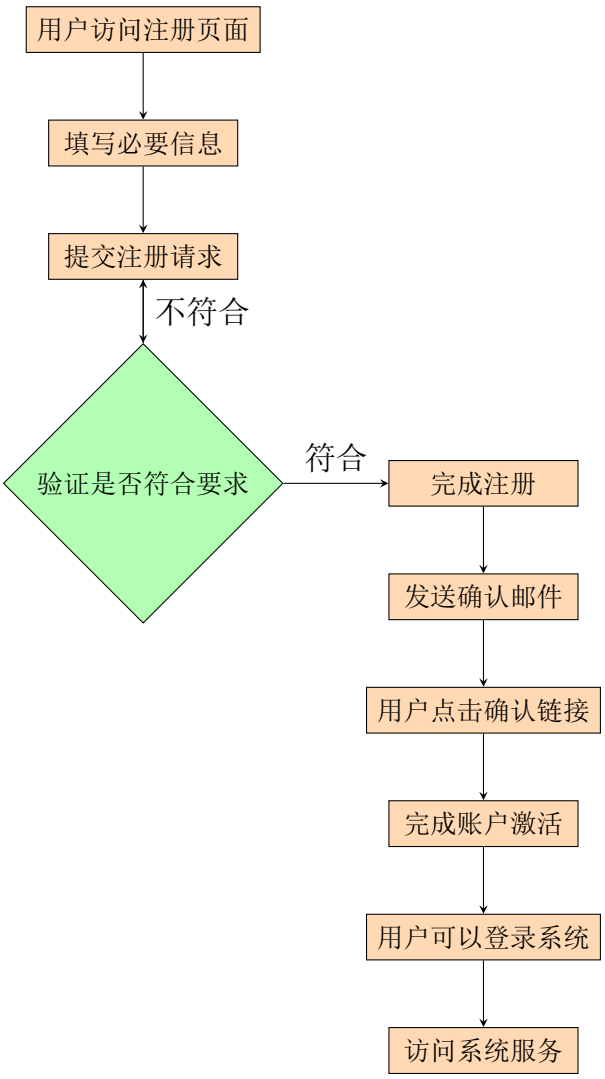
用户访问注册页面
填写必要信息
提交注册请求
不符合
验证是否符合要求
符合
完成注册
发送确认邮件
用户点击确认链接
完成账户激活
用户可以登录系统
访问系统服务

图 5: 用户注册流程

4.2.2 个人医疗信息管理

用户可以轻松查看、管理和更新自己的个人医疗信息，包括过往病史、药物过敏信息等，以确保信息的准确性和时效性。但是首先需要登录。

操作	描述
查看个人医疗信息	用户登录后，可以查看个人医疗信息，包括病史、药物过敏信息等。
更新个人医疗信息	用户可以更新病史、药物过敏等信息。需要通过系统审核确保信息的准确性。
授权访问	用户可以授权医生或家属访问特定的医疗信息。
查看访问记录	用户可以查看谁访问了他们的医疗记录，确保信息的安全。
接收系统提示	用户根据更新的医疗信息接收健康提示或提醒，比如药物相互作用警告。

表 4: 个人医疗信息管理操作（更新版）

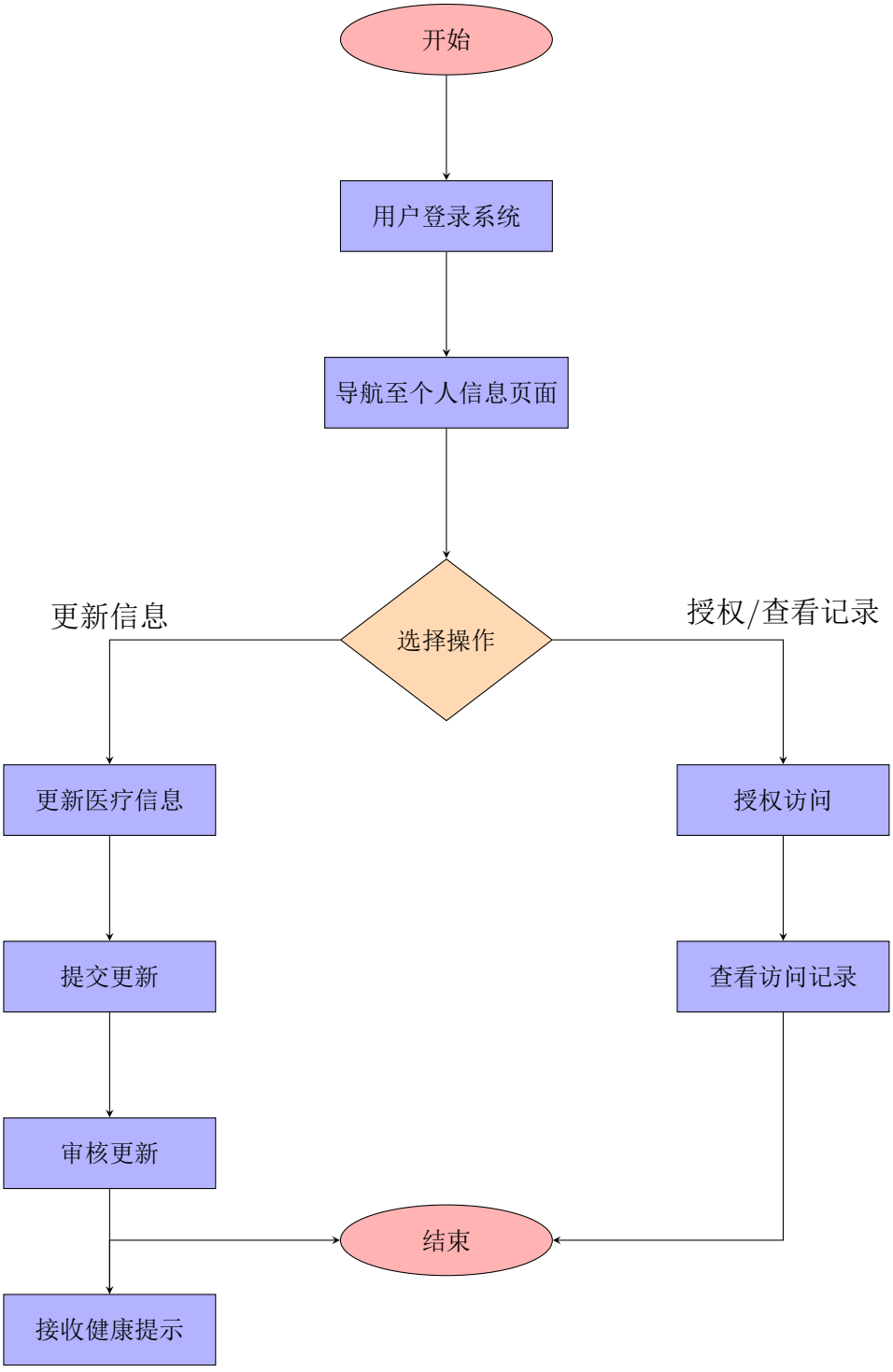

图 6: 个人医疗信息管理流程（更新版）

4.2.3 人工智能病情咨询服务

集成 AI 技术，用户可以咨询自己的病情，并获得初步的医疗建议，为进一步的诊断和治疗提供参考。

操作	描述
提交病情描述	用户通过系统输入自己的症状和相关信息，作为咨询的基础。
AI 分析病情	系统利用人工智能技术分析用户提交的信息，识别可能的病情。
获取医疗建议	根据 AI 分析结果，系统提供初步的医疗建议。
用户反馈	用户可以对提供的建议进行反馈，包括确认建议的有效性或请求更多信息。
医生审核	若用户请求，或 AI 系统不确定，医生会对病情进行人工审核，并提供进一步的建议。
追踪与跟进	系统定期追踪用户病情的变化，必要时提醒用户进行复查或更新病情信息。

表 5: 人工智能病情咨询服务操作

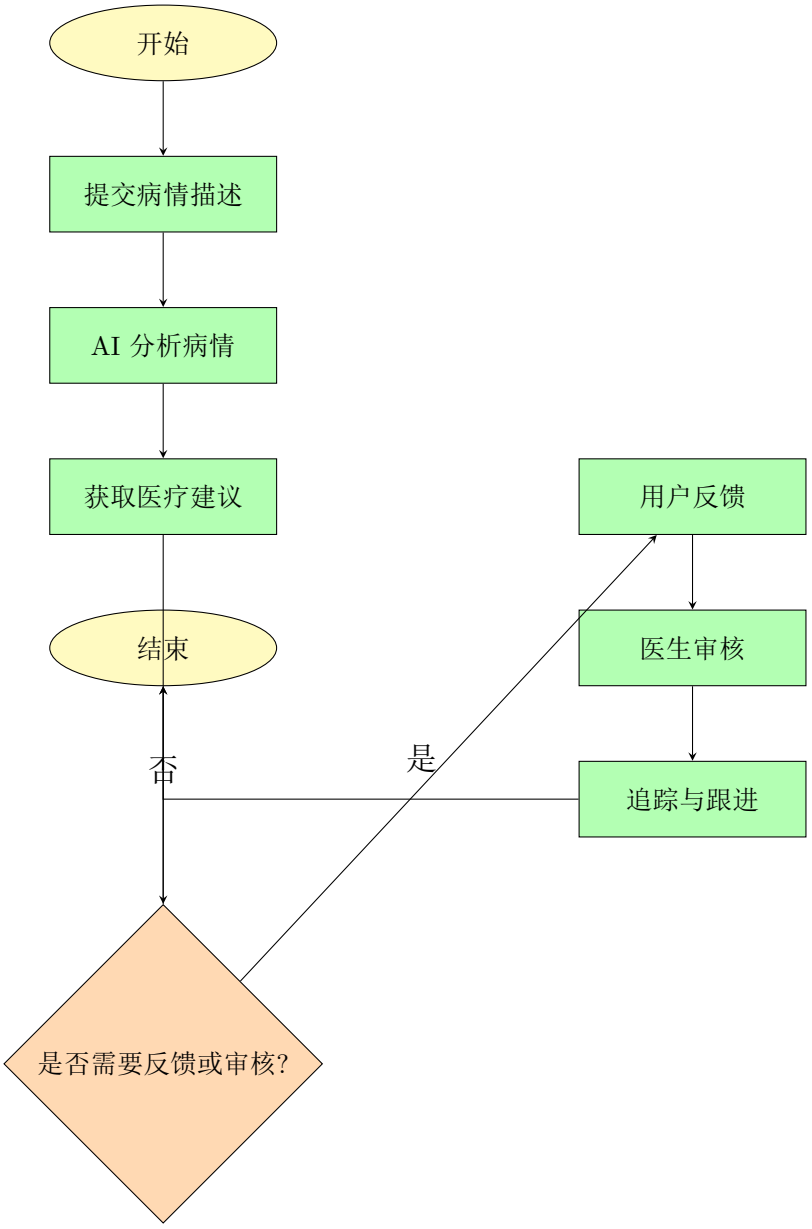

图 7: 人工智能病情咨询服务流程

4.2.4 科室与专业领域介绍

系统提供详细的科室信息，包括各科室的专业领域、医生团队介绍等，帮助用户了解并选择合适的科室。此表格旨在展示系统提供的科室信息及其内容：接下来，我们设计一个流

科室名称	提供的信息
内科	内科团队介绍、专业领域（如心脏病学、消化内科等）、常见病例处理
外科	外科团队介绍、专业领域（如普外科、神经外科等）、手术类型和案例
儿科	儿科团队介绍、儿童常见疾病处理、预防接种和健康管理
妇产科	妇产科团队介绍、孕期管理、生育服务和妇女健康问题

表 6: 科室与专业领域介绍

程图来描述用户如何利用系统了解科室信息，并做出选择：

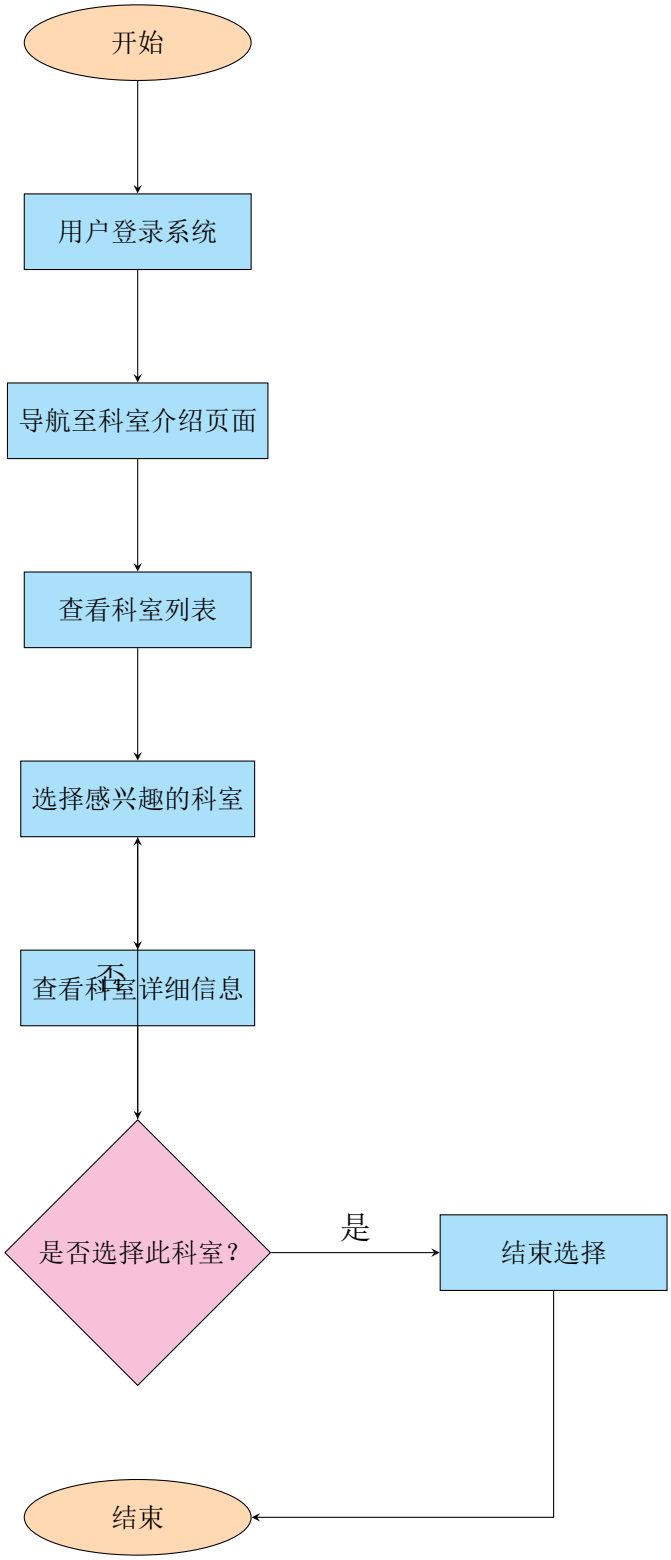
开始
用户登录系统
导航至科室介绍页面
查看科室列表
选择感兴趣的科室
查看科室详细信息
否
是否选择此科室？
是
结束选择
结束

图 8: 科室选择流程

4.2.5 医生预约时段选择

用户可以查看医生的可选时段，并根据自己的时间安排进行预约，提高就诊的灵活性和便利性。此表格展示了医生预约时段选择的相关操作及描述：

操作	描述
查看医生列表	用户可以浏览所有可预约的医生列表，包括医生的专业领域、评分及用户评价。
查看医生时段	选择一位医生后，用户可以查看该医生的可预约时段。
选择预约时段	用户根据自己的时间安排选择一个合适的预约时段。
确认预约	用户填写个人信息（如联系方式）并确认预约。
接收确认	预约成功后，用户将接收到预约确认信息，包括就诊时间和地点。
取消预约	用户可以在规定时间内取消预约，并重新安排。
提交评价	完成就诊后，用户可以提交对医生的评价。
医生推荐	根据用户的预约历史和评价，系统推荐医生。

表 7: 医生预约时段选择操作

下面是描述用户如何进行医生预约时段选择的流程图：

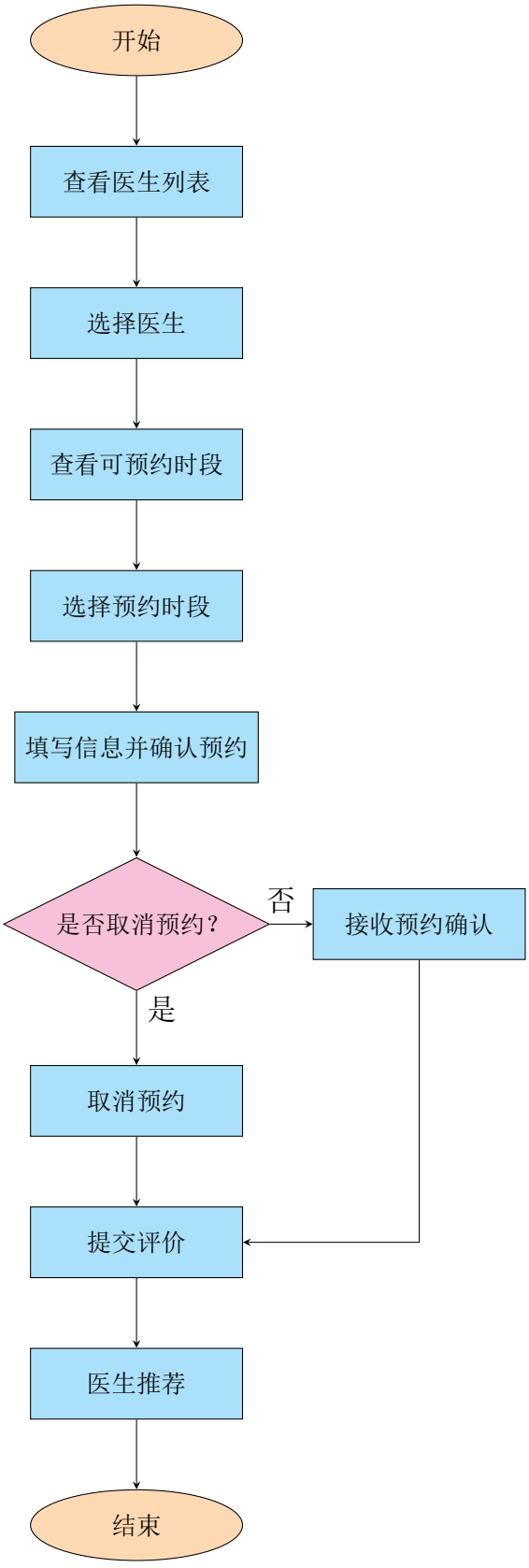

图 9: 医生预约时段选择流程

4.2.6 医生选择与预约确认

在选定时段后，用户可以根据自己的需求和医生的专业背景选择心仪的医生，并确认预约。此表格将展示用户在选择医生和确认预约时需执行的操作和相应的描述：

4.2.7 医生选择与预约确认

在选定时段后，用户可以根据自己的需求和医生的专业背景选择心仪的医生，并确认预约。此表格将展示用户在选择医生和确认预约时需执行的操作和相应的描述：

操作	描述
查看医生资料	用户可以查看各医生的详细资料，包括专业背景、擅长领域、工作经验及用户评价。
在线咨询	用户可以在线咨询医生，以便更好地了解医生的专业能力和态度。
选择医生	根据医生的资料和用户自己的需求，选择最适合的医生。
选择时段	用户选择医生后，根据可用时段表选择最合适的预约时间。
填写预约信息	输入必要的预约信息，如个人健康状况简介等。
修改/取消预约	用户可以在规定时间内修改或取消预约。
确认预约	完成所有信息的填写后，提交预约请求。系统会发送预约确认通知给用户。
预约反馈	预约后，用户可提交对预约过程的反馈。
预约跟踪	用户可以跟踪预约状态，包括提醒通知和就诊前的准备事项。

表 8: 医生选择与预约确认操作

接下来的流程图将描述用户如何进行医生的选择和预约确认的整个过程：

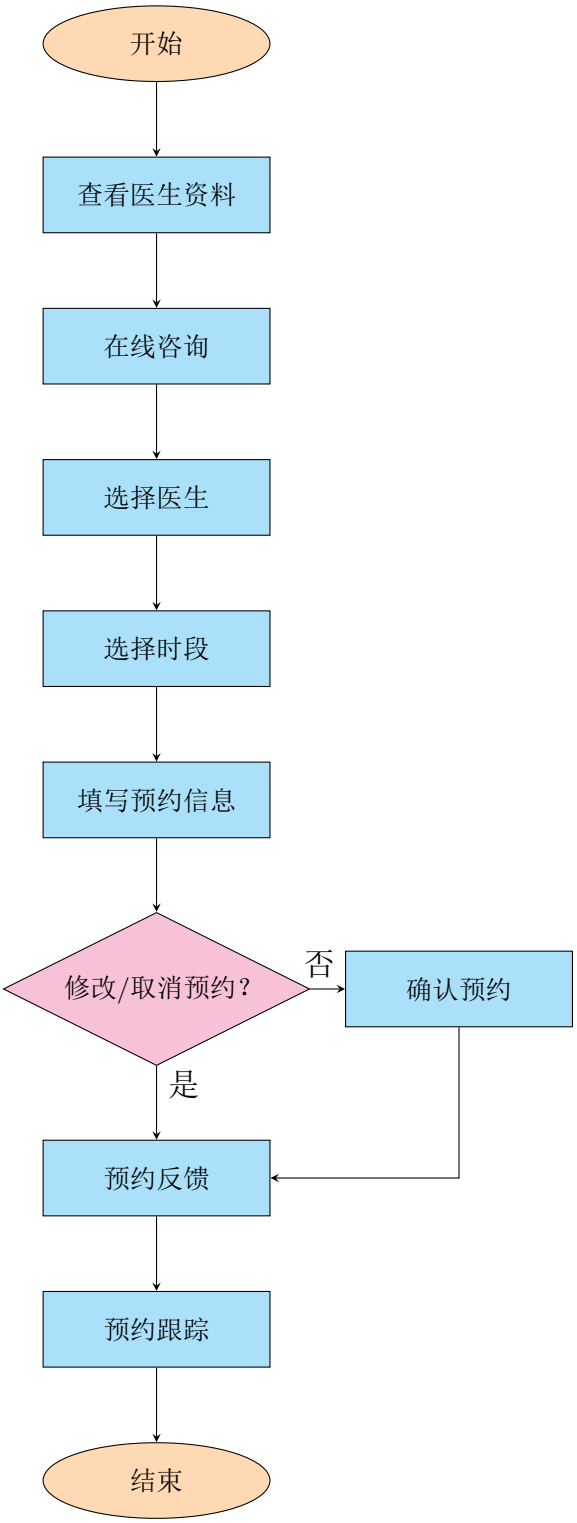

图 10: 医生选择与预约确认流程

4.2.8 医疗费用账单管理

用户可以在线提交和查看自己的医疗费用账单，包括详细的费用清单和总计，便于费用的核对和理解。此表格旨在展示与医疗费用账单管理相关的操作及其描述：以下流程图将描

操作	描述
提交医疗费用账单	用户可以在线提交自己的医疗费用账单，包括上传相关的医疗费用凭证。
查看费用账单	用户可以查看已提交的医疗费用账单及其详细的费用清单和总计。
费用账单审核	系统自动或人工审核提交的费用账单及凭证，确保费用的准确性。
费用账单异议	用户可以对账单中的某些费用项提出异议，要求重新审核或解释。
接收审核结果	用户接收到费用账单审核的最终结果，包括是否接受异议及调整后的费用总计。
在线支付费用	用户可以选择在线支付经审核后的医疗费用。

表 9: 医疗费用账单管理操作

述一个复杂的医疗费用账单管理流程，包括账单的提交、审核、异议处理和支付等步骤. 这个流程图涵盖了从用户提交医疗费用账单开始的全过程，包括账单的审核、对审核结果的异议处理，以及接收最终审核结果后的在线支付步骤。这个设计旨在提供一个全面的视角，展示医疗费用管理过程中可能涉及的各个环节，确保用户可以轻松地管理和理解自己的医疗费用。

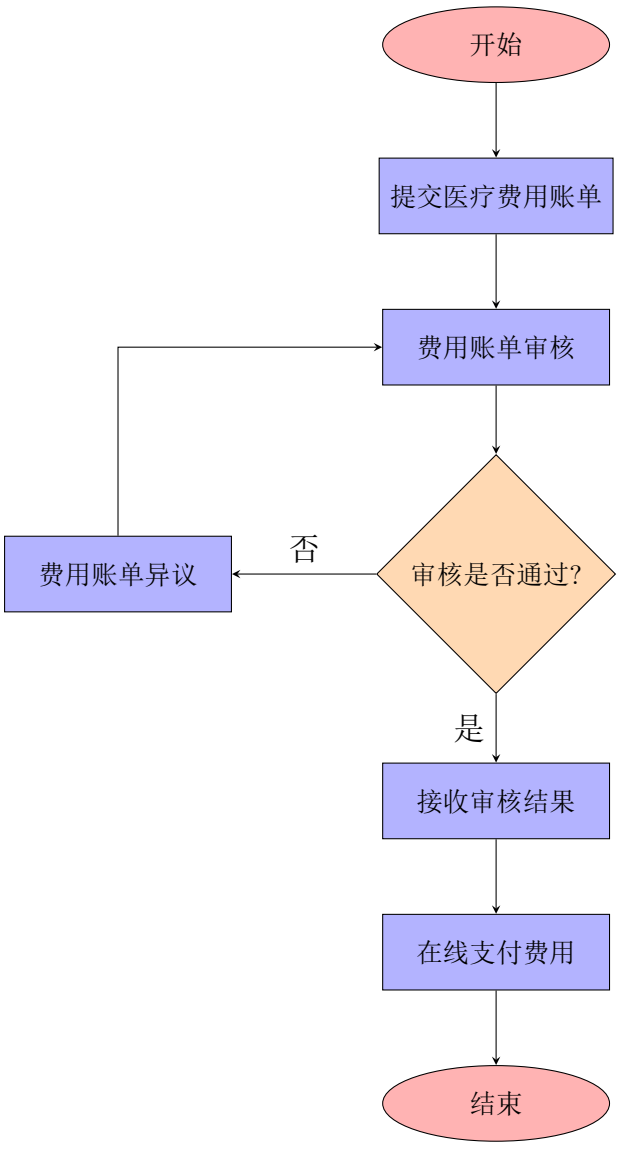

图 11: 医疗费用账单管理流程

4.2.9 缴费与退费流程

系统支持在线缴费和退费功能，简化了费用处理流程，减少了用户在医院的等待时间。此表格将介绍用户在系统中进行缴费和退费时需要执行的操作及其相应的描述：以下复杂的流

操作	描述
查看待缴费用	用户可以查看自己的待缴费用清单及总计。
选择缴费方式	用户选择合适的在线支付方式进行缴费，如信用卡、支付宝、微信等。
缴费确认	完成支付后，系统将显示缴费成功确认信息。
查看退费政策	用户可以查阅具体的退费政策，了解可能的退费条件和流程。
申请退费	在符合退费条件的情况下，用户可以提交退费申请，附上必要的证明材料。
退费审核	系统自动或人工审核退费申请及证明材料。
退费处理	审核通过后，系统将按照用户选择的方式退回费用。
接收退费确认	用户接收到退费成功的确认信息。

表 10: 缴费与退费流程操作

程图将展示缴费与退费过程中可能涉及的多个决策点和步骤. 在这个流程中，用户首先查看待缴费用，然后选择支付方式进行缴费，并接收缴费确认。同时，用户也可以查看退费政策，如果需要，可以提交退费申请并经过审核后完成退费，最终接收退费确认。这个流程图详细地展示了用户在缴费与退费过程中可能遇到的各种情况，确保用户能够清楚地了解和执行每一步。

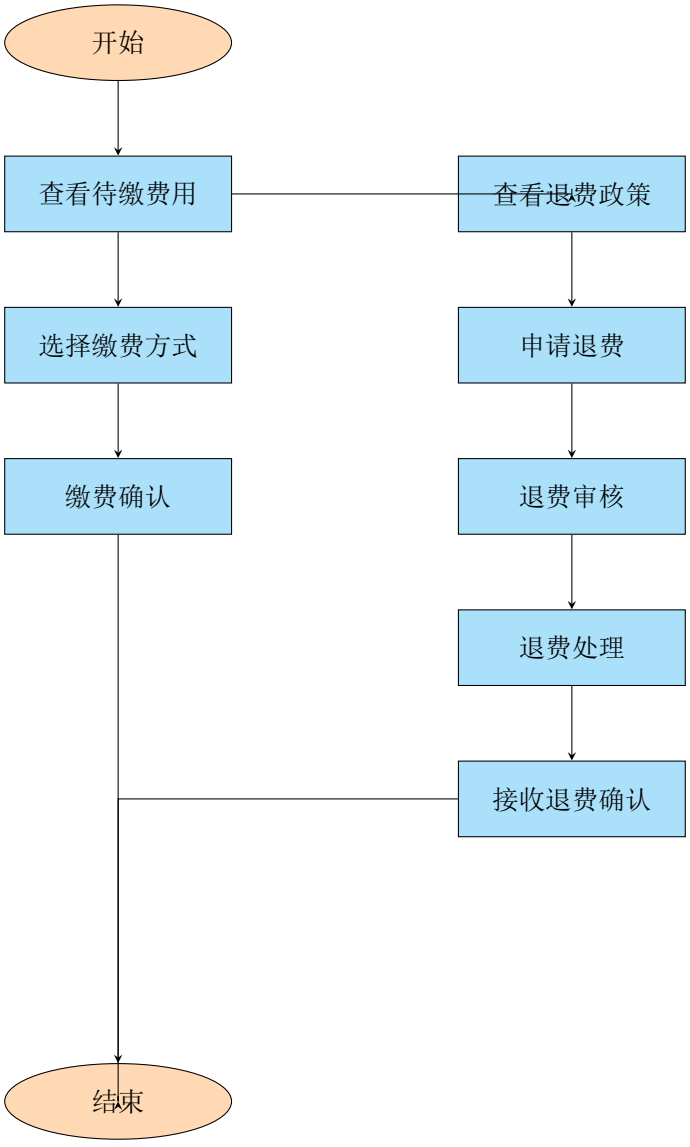

图 12: 缴费与退费流程图

4.2.10 电子处方查询

用户可以在线查询医生开具的处方信息，包括药物名称、用法用量等，便于用户理解和遵循医嘱。此表格展示了用户在线查询电子处方时的相关操作及其描述：

操作	描述
登录系统	用户需要登录系统以访问自己的电子处方信息。
选择查询处方	用户可以选择查看最新的处方或历史处方记录。
查看处方详情	展示选定处方的详细信息，包括药物名称、用法用量、开具医生等。
下载处方	用户可以下载处方信息，以便线下购药或备份。
咨询医生	若对处方有疑问，用户可以通过系统向开具处方的医生发起咨询。
评价服务	完成处方查询和使用后，用户可以对服务进行评价。

表 11: 电子处方查询操作

下面的流程图将展示一个涵盖多个可能操作和选择的复杂电子处方查询流程。这个流程图从用户登录系统开始，之后用户可以选择查看最新或历史的处方记录，查看处方详情，并根据需要决定是否下载处方。如果对处方有疑问，用户还可以选择咨询医生。最后，用户可以对使用过程中的服务进行评价。此流程图考虑了用户在查询电子处方过程中可能的各种行为，使其成为一个全面且复杂的流程描述。

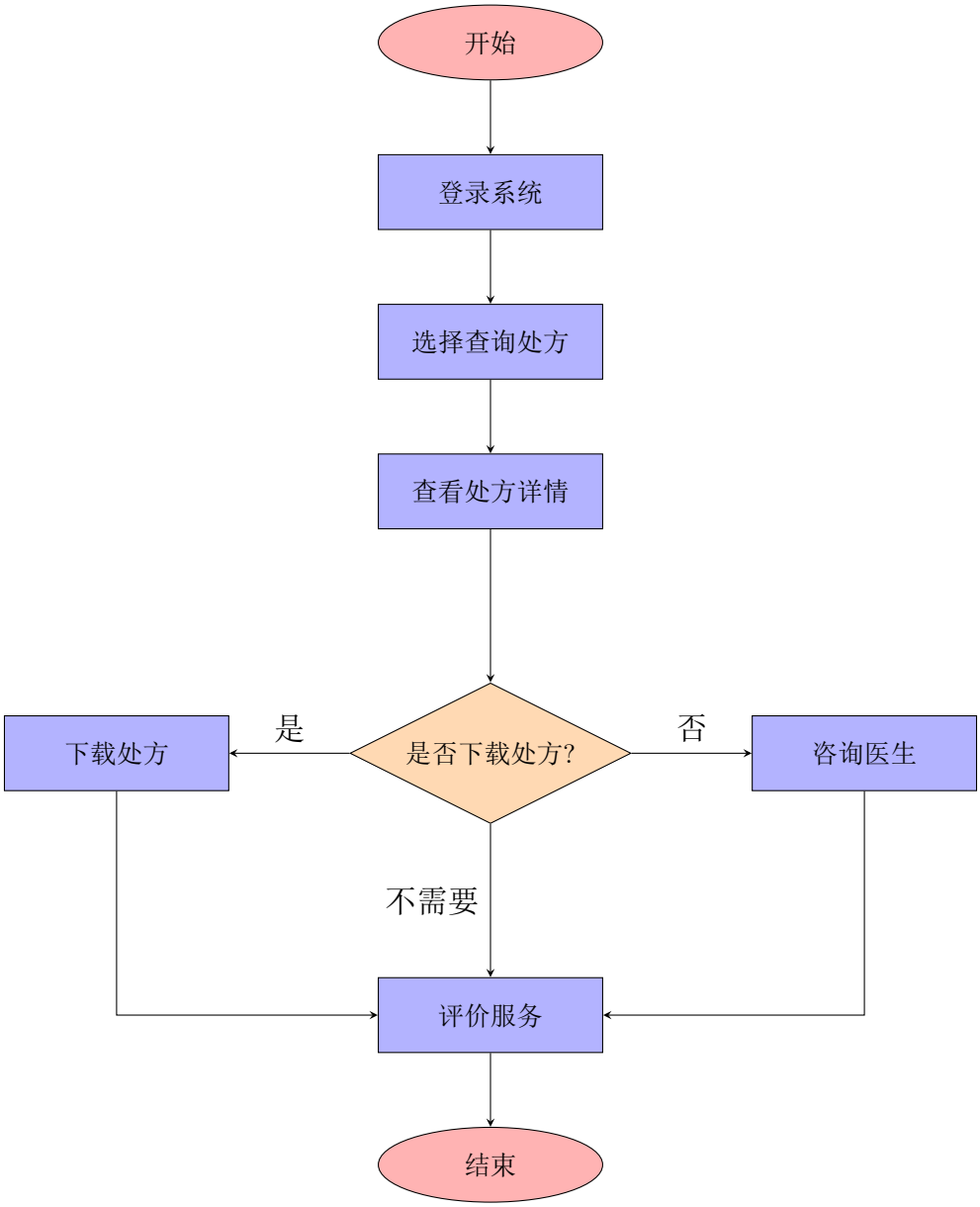

图 13: 电子处方查询流程

4.2.11 电子病历搜索与访问

用户可以搜索并查看自己的电子病历，包括历史就诊记录、检查结果等，便于健康管理和疾病预防。此表格将介绍用户在线查看电子病历时的相关操作及其描述：

操作	描述
登录系统	用户需要登录系统以访问自己的电子病历信息。
搜索病历	用户可以根据日期、疾病名称或医生姓名等关键词搜索相关的电子病历。
选择病历记录	从搜索结果中选择特定的病历记录以查看详细信息。
查看病历详情	显示选定病历的详细信息，包括诊断、治疗过程、检查结果等。
下载病历	用户可以下载电子病历的副本，以便离线查看或备份。
咨询医生	若对病历内容有疑问，用户可以通过系统向相关医生发起咨询。

表 12: 电子病历搜索与访问操作

下面的流程图将展示一个包含多个分支和决策点的复杂电子病历搜索与访问流程。这个流程图从用户登录系统开始，接着用户可以搜索特定的电子病历记录，并选择查看病历详情。根据需要，用户可以决定是否下载病历的副本。如果对病历内容有疑问，还可以选择咨询相关的医生。此流程图包含了多个选择和操作，旨在详细描述电子病历搜索与访问的全过程。

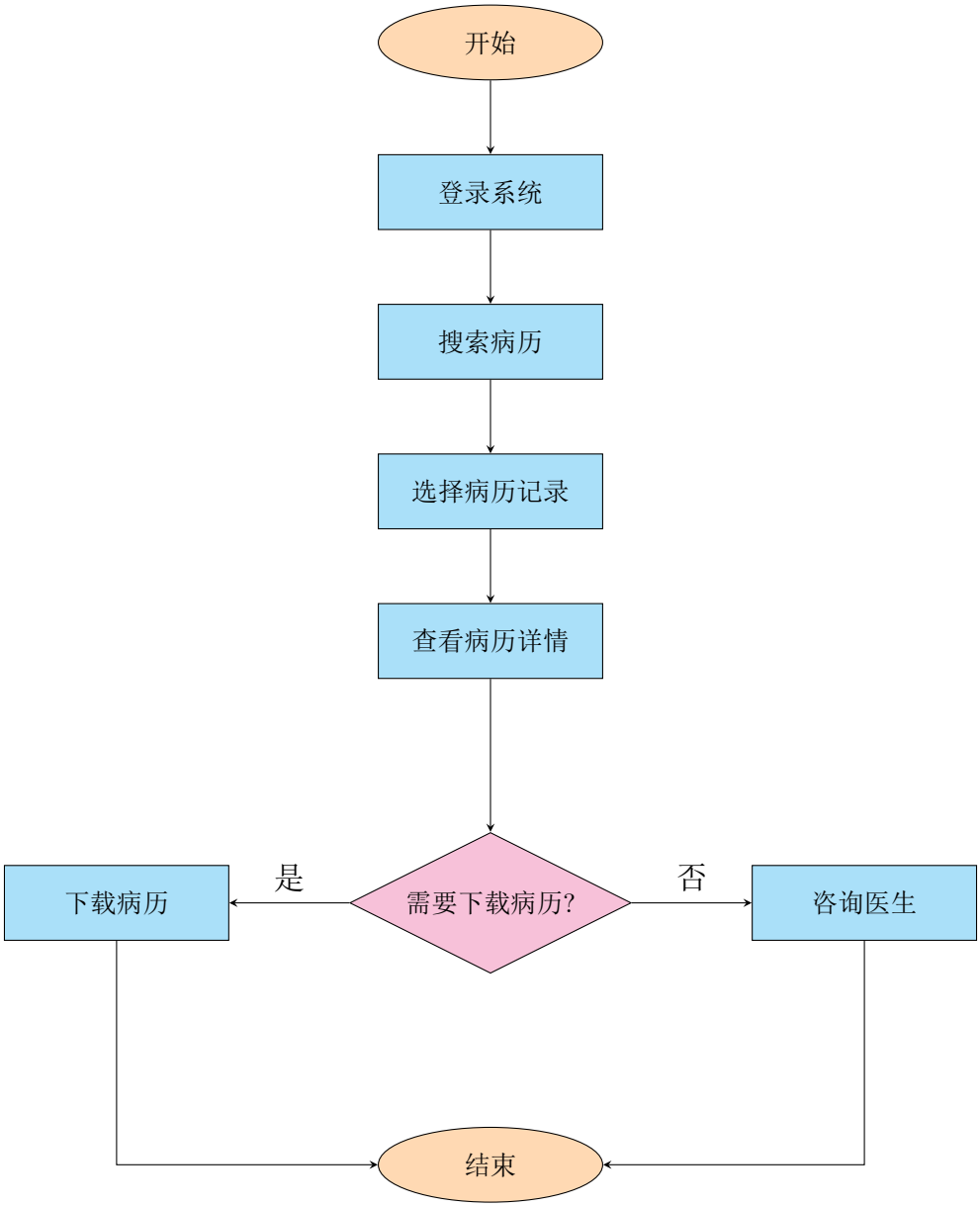

图 14: 电子病历搜索与访问流程

4.2.12 预约挂号与问诊服务

用户可以方便地预约挂号，并在预约时间进行在线或线下问诊，确保医疗服务的连续性和及时性。此表格展示了用户预约挂号与问诊时的相关操作及其描述：以下复杂的流程图展

操作	描述
选择医生或科室	用户可以根据自己的需求选择特定的医生或科室进行预约。
确定预约时间	用户从可用的时间段中选择一个最适合自己的时间进行预约。
填写个人信息	用户需填写个人信息，如姓名、联系方式等，以完成预约挂号。
预约确认	系统确认用户的预约请求，并发送预约详情给用户。
进行问诊	在预约时间，用户可以选择在线问诊或到医院进行线下问诊。
问诊结束	问诊结束后，医生提供诊断结果和治疗建议。

表 13: 预约挂号与问诊服务操作

示了从预约挂号到完成问诊的整个过程，包括多个操作和决策点。在这个流程图中，用户首先选择合适的医生或科室，然后选择一个预约时间并填写个人信息以完成预约挂号。预约确认后，用户需要决定是进行在线问诊还是到医院进行线下问诊。问诊结束后，医生将提供诊断结果和治疗建议。这个设计考虑了预约挂号与问诊服务中可能的各种情况，确保了医疗服务的连续性和及时性。

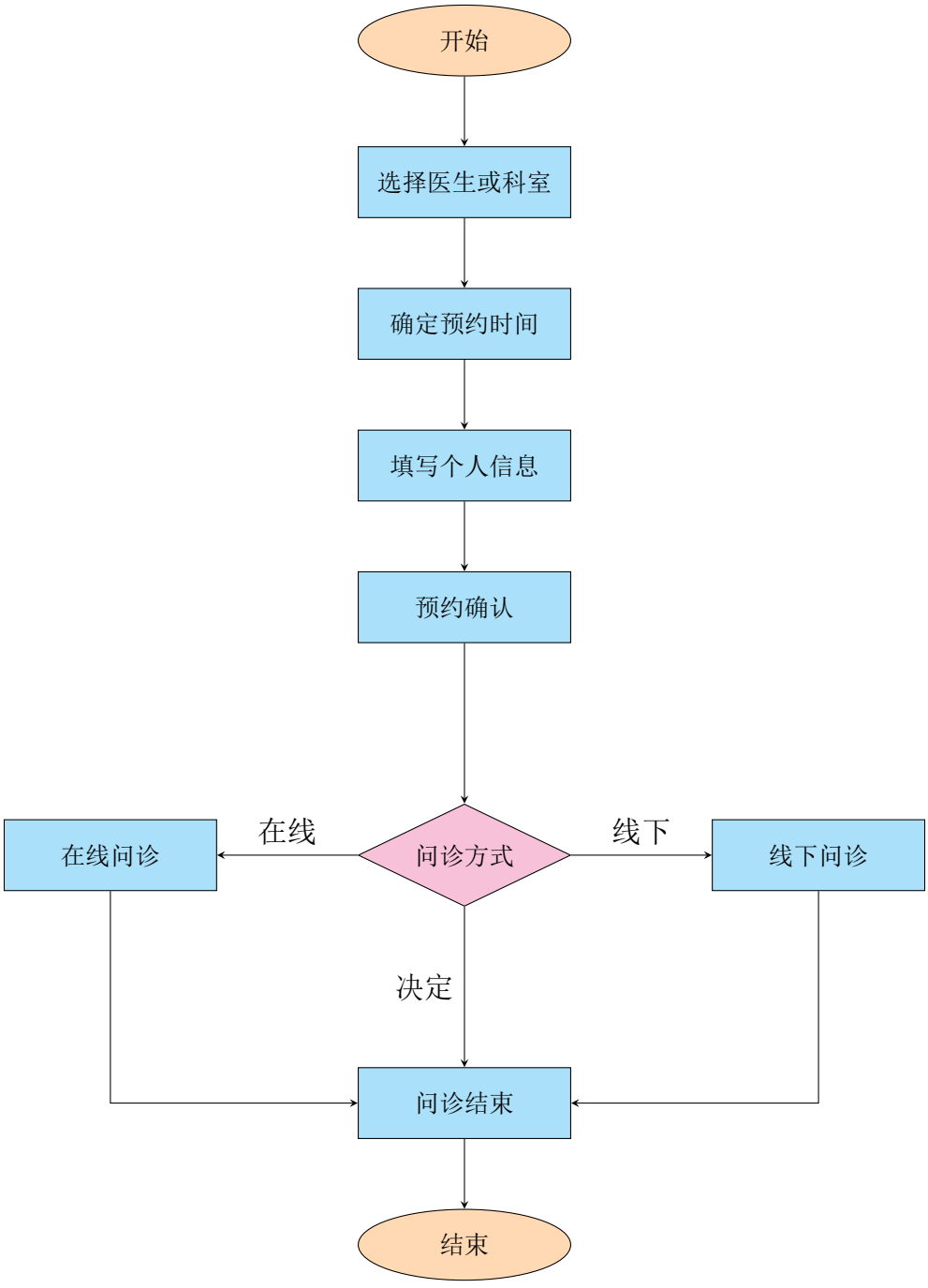

图 15: 预约挂号与问诊服务流程

4.2.13 时段灵活性与个性化预约

用户可以根据自己的时间安排，选择最佳的就诊时段，系统还会提供个性化的预约建议，以满足不同用户的需求。此表格展示了与时段灵活性和个性化预约相关的操作及其描述：

操作	描述
登录系统	用户需要登录系统以访问预约功能。
输入个人偏好	用户可以输入自己的时间偏好、医生偏好等信息。
接收预约建议	系统根据用户输入的偏好提供个性化的预约建议。
选择紧急预约选项	如有紧急情况，用户可以选择紧急预约，享有优先预约权。
选择就诊时段	用户从系统提供的建议中选择最适合自己的就诊时段。
确认预约	用户确认预约详情并完成预约过程。
预约成功	系统发送预约成功的通知，包括就诊时间和地点等信息。
就诊后评价	用户就诊后，可以提交对医生和就诊体验的评价。
自动更新偏好	系统根据用户的历史预约信息和评价自动更新个人偏好。

表 14: 时段灵活性与个性化预约操作

面的流程图将展示从登录系统到完成预约的整个过程，包括多个操作和决策点。在这个流程图中，用户首先登录系统并输入个人偏好信息，如可用的时间段和偏好的医生类型等。系统根据用户的偏好提供个性化的预约建议。用户从中选择一个最适合自己的就诊时段并确认预约。最后，用户接收到预约成功的通知。这个设计旨在提供灵活的时段选择和个性化的预约建议，以满足不同用户的需求。

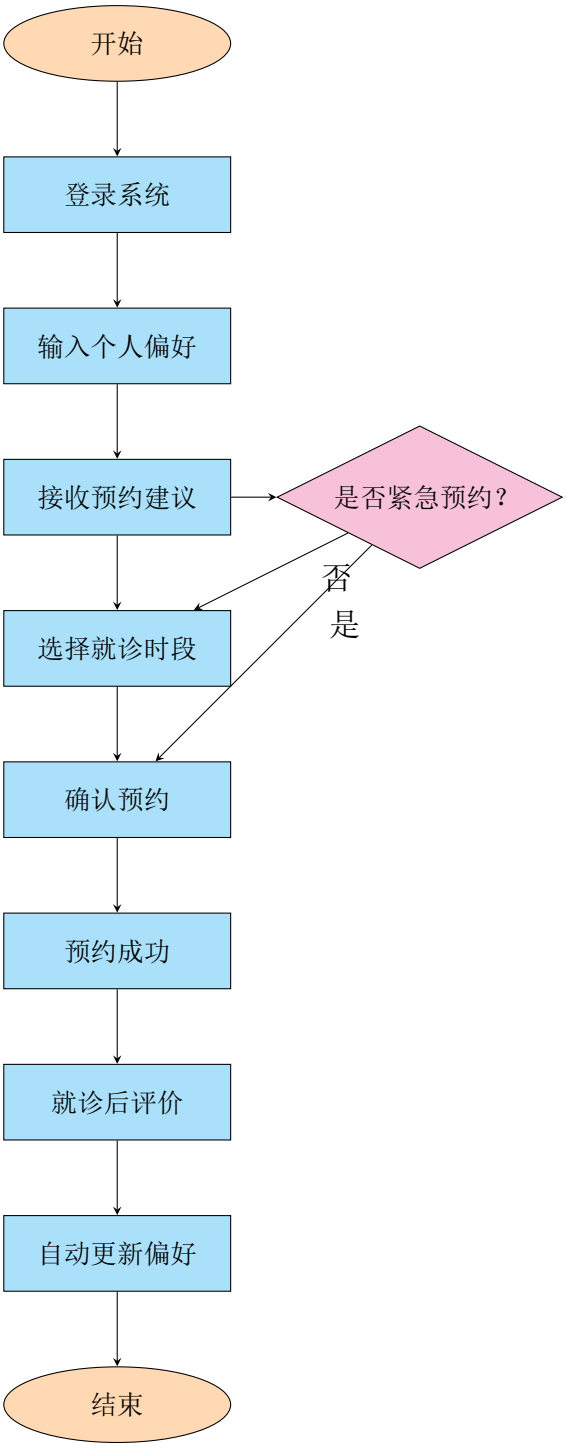

图 16: 时段灵活性与个性化预约流程

4.2.14 电子问诊单与后续跟进

问诊后，用户可以接收到电子问诊单，便于记录和后续跟进，确保医疗服务的完整性。此表格展示了与电子问诊单接收和后续跟进相关的操作及其描述：下面的流程图将展示一个包

操作	描述
完成问诊	用户完成在线或线下问诊后，系统自动生成电子问诊单。
查看问诊单	用户可以在系统中查看电子问诊单的内容，包括诊断结果、治疗建议等。
下载问诊单	用户有选项下载问诊单，以便于打印或电子存档。
咨询医生	对问诊单有疑问的用户可以直接通过系统咨询医生。
安排后续治疗	根据问诊单的建议，用户可以安排后续的治疗或复诊。

表 15: 电子问诊单与后续跟进操作

含多个步骤和决策点的复杂电子问诊单接收与后续跟进流程。在这个流程图中，用户首先完成在线或线下问诊，随后系统将自动生成电子问诊单。用户可以查看问诊单的详细内容，包括医生的诊断结果和治疗建议。用户有选项下载问诊单，若对问诊单内容有疑问或需要进一步的指导，可以选择咨询医生。根据问诊单的建议，用户还可以安排后续的治疗计划或复诊。这个设计旨在确保医疗服务的完整性，同时提供了灵活的后续跟进选项。

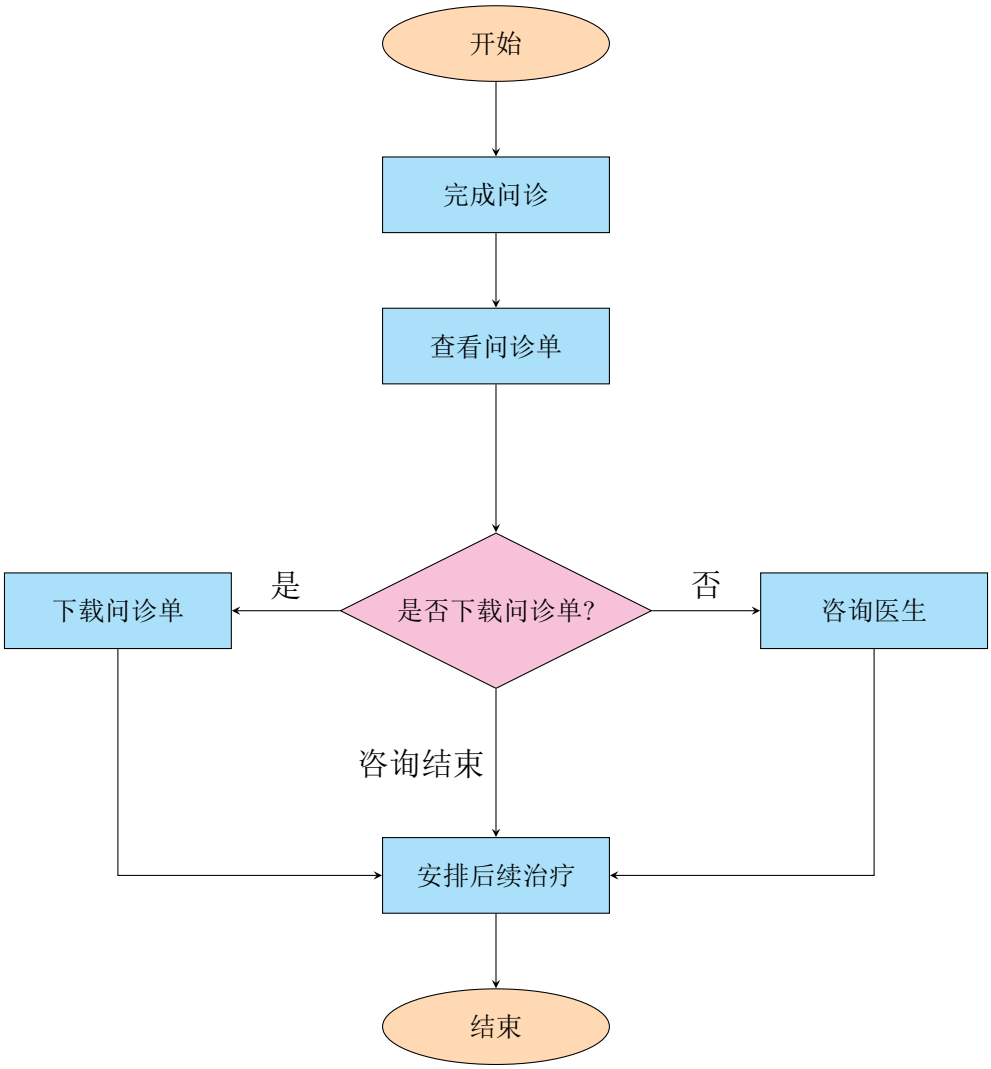

图 17: 电子问诊单与后续跟进流程

4.2.15 医疗服务评价体系参与

用户可以参与对医生和医院服务的评价，为其他患者提供参考，同时也帮助医疗机构改进服务质量。此表格展示了与医疗服务评价相关的操作及其描述：下面的流程图将展示用户

操作	描述
登录系统	用户需要登录系统才能参与评价。
选择评价对象	用户可以选择评价特定的医生或医院服务。
填写评价内容	用户填写关于医疗服务的评价，可以包括满意度、服务质量、环境等方面。
提交评价	用户提交填写好的评价内容。
查看评价反馈	用户可以查看自己的评价是否被医疗机构采纳或对服务进行了改进。

表 16: 医疗服务评价体系参与操作

参与医疗服务评价体系的整个过程，包括从登录系统到查看评价反馈的多个步骤。在这个流程图中，用户首先登录系统，然后选择要评价的医生或医疗服务。之后，用户填写评价内容，并提交评价。最后，用户可以查看自己的评价是否对医疗机构的服务质量产生了影响，例如是否有改进措施的反馈。这个设计旨在鼓励用户参与医疗服务评价，为其他患者提供参考，同时帮助医疗机构了解并改进服务质量。

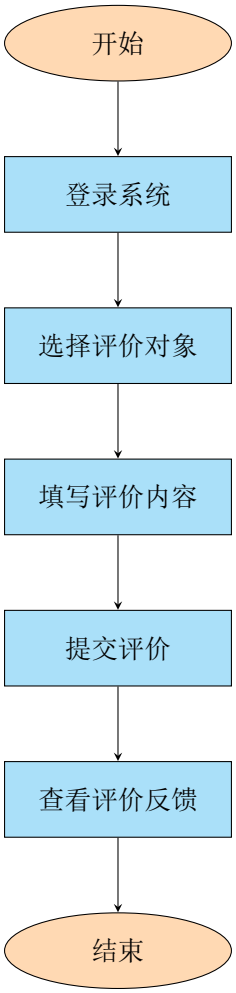

图 18: 医疗服务评价体系参与流程

4.2.16 处方与病历的综合查询

用户可以查询医生开具的处方，并访问自己的电子病历，便于对健康状况进行全面管理。此表格展示了与处方和病历综合查询相关的操作及其描述：下面的流程图将展示一个从登录

操作	描述
登录系统	用户需登录系统才能访问自己的处方和病历信息。
查询处方	用户可以根据日期、医生姓名等关键词查询医生开具的处方。
访问电子病历	用户可以查看和访问自己的电子病历，包括历史就诊记录、检查结果等。
下载文档	用户有选项下载电子处方和病历的副本，以便离线查看或打印。
咨询医生	对处方或病历内容有疑问的用户可以通过系统咨询相关医生。

表 17: 处方与病历的综合查询操作

系统到查询处方和病历，再到下载文档和咨询医生的复杂过程。在这个流程图中，用户首先登录系统，然后可以查询特定的处方或访问自己的电子病历记录。用户有选项下载处方或病历的副本，若对内容有疑问，可以选择咨询相关医生。这个设计旨在提供一个方便的方式，使用户能够全面管理自己的健康状况，同时也便于医生了解患者的医疗历史。

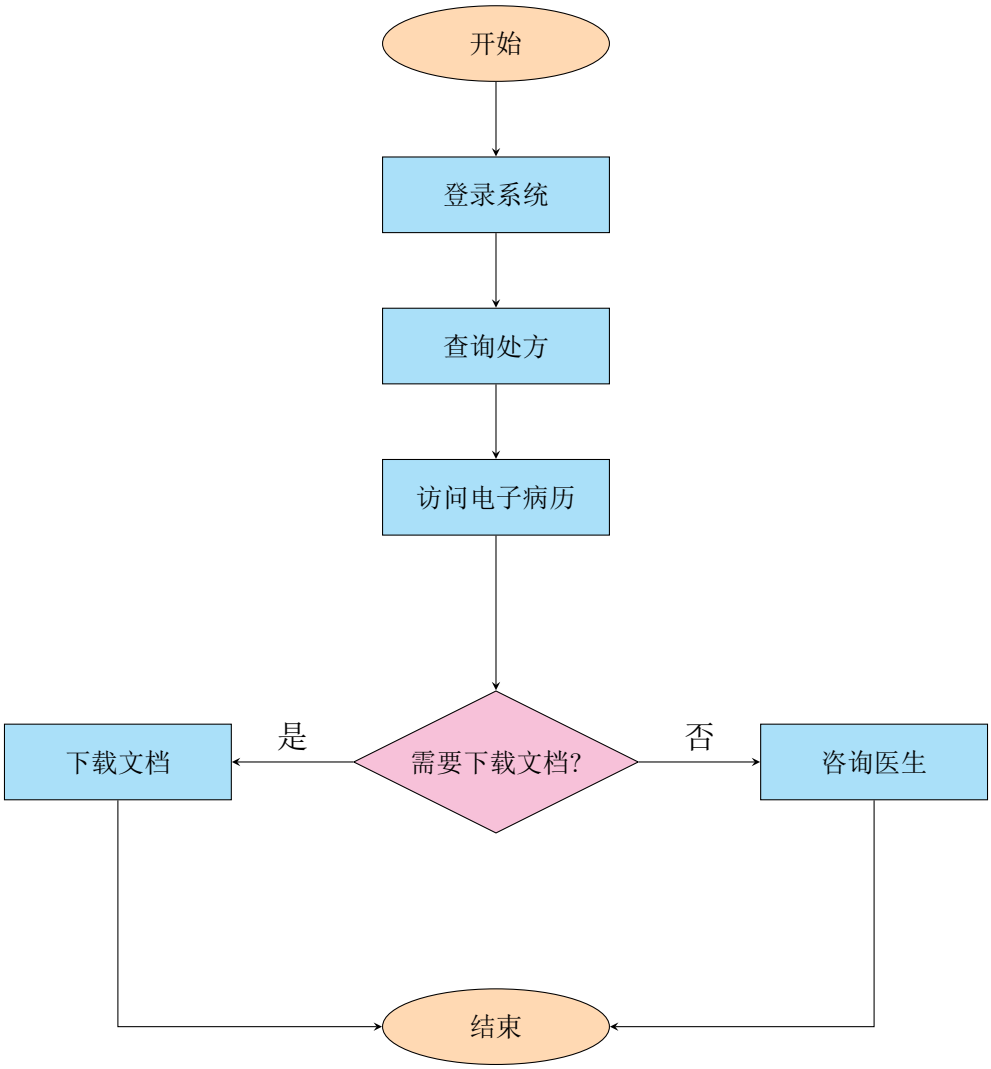

图 19: 处方与病历的综合查询流程

4.3 状态图

本节将详细解释前一小节中各类功能的状态 UML 图，该图展示了医疗预约管理系统中用户与系统交互的各个阶段。图23展示了用户从开始使用系统到最终结束服务的完整流程。

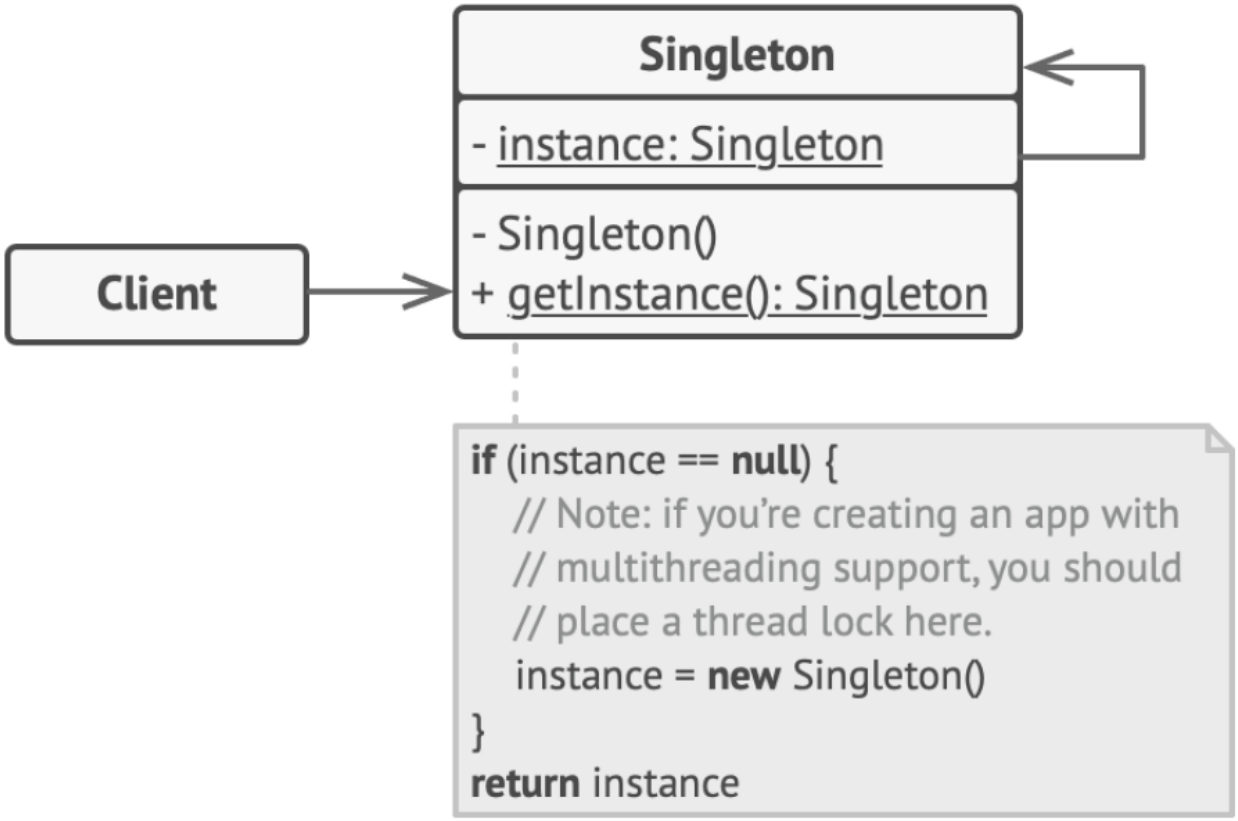

图 20: 功能状态 UML 图

初始状态用星号 ([*]) 表示，代表用户开始与系统交互。首先，用户访问登录页面，输入账号和密码。在这一阶段，有两个可能的结果：如果验证成功，用户将登录成功并进入服务选择状态；如果验证失败，用户将返回登录页面重新输入账号密码。

在服务选择状态，用户有多个选项，包括查看医生列表、查询处方、访问电子病历、参与医疗服务评价、查看科室信息、预约挂号与问诊服务、时段灵活性与个性化预约、医疗费用账单管理以及缴费与退费流程。每个选项都对应一个动作，用户可以根据自己的需求选择相应的服务。

如果用户选择预约挂号与问诊服务，他们将完成问诊，并接收电子问诊单，随后进入电子问诊单与后续跟进状态。在这里，用户可以查看问诊单详情，并根据需要进行后续操作。

当用户选择时段灵活性与个性化预约，他们将进行预约并确认预约时间。一旦预约成功，状态图将结束，表示用户已经完成了预约流程。

在缴费与退费流程中，用户需要完成在线支付。支付成功后，用户将回到服务选择状态，可以选择其他服务或结束当前会话。

医疗费用账单管理允许用户查看费用账单。用户可以查看详细的费用清单，并在需要时进行支付或其他相关操作。

查询处方和访问电子病历状态允许用户查看处方详情和病历详情，以便更好地了解自己的医疗状况和历史记录。

参与医疗服务评价状态鼓励用户对所接受的服务进行评价，这有助于医疗机构改进服务质量。

最后，每个服务的结束都将用户带回服务选择状态，用户可以选择继续使用其他服务或结束当前会话。当用户结束所有服务时，他们将退出系统，状态图也随之结束。

4.4 类图

4.4.1 用户类图

用户类图展示了用户账户的基本属性和行为。用户 (User) 类包含用户名 (username)、密码 (password)、电子邮件 (email) 以及与用户相关的处方 (Prescriptions) 和医疗记录 (MedicalRecords) 的列表。用户可以进行预约 (makeAppointment)、咨询医生 (consultDoctor)、查看处方 (viewPrescriptions) 和查看医疗记录 (viewMedicalRecords)。

处方 (Prescription) 类包含处方 ID(prescriptionID)、开具日期 (date)、开处方的医生姓名 (doctorName) 和处方详情 (details)。医疗记录 (MedicalRecord) 类包含记录 ID(recordID)、记录日期 (date)、医生姓名 (doctorName)、诊断 (diagnosis) 和治疗信息 (treatment)。

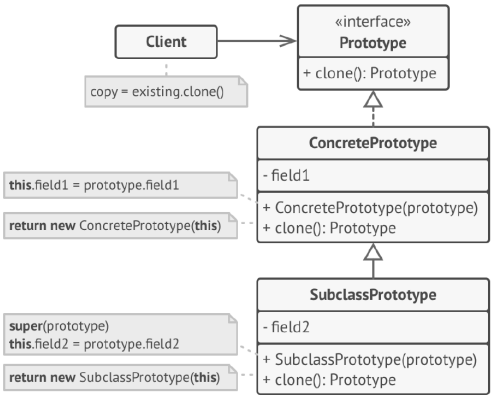

图 21: 功能状态 UML 图

4.4.2 预约与问诊服务类图

预约与问诊服务类图描述了预约系统的结构和行为。预约服务 (AppointmentService) 类负责创建 (makeAppointment) 和取消 (cancelAppointment) 预约。预约 (Appointment) 类包含预约时间 (datetime)、负责的医生 (doctor)、预约的用户 (user) 和预约状态 (status)。状态 (Status) 枚举定义了预约的可能状态，包括已安排 (Scheduled)、已取消 (Cancelled) 和已完成 (Completed)。

医生 (Doctor) 类包含医生 ID(doctorID)、姓名 (name)、专业 (specialty) 和医生的预约列表 (appointments)。医生可以进行咨询 (consult)。咨询服务 (ConsultationService) 类提供进行咨询 (conductConsultation)、提供处方 (providePrescription) 和提供医疗记录 (provideMedicalRecord) 的行为。

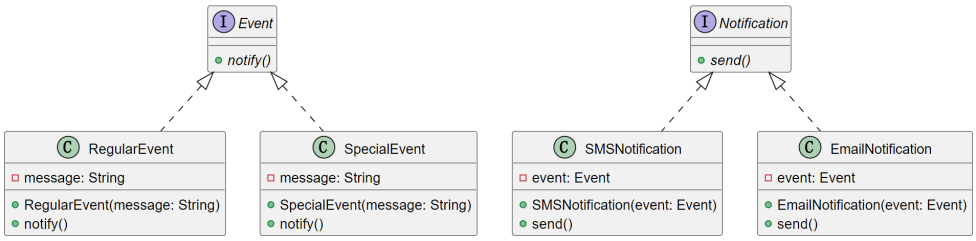

图 22: 功能状态 UML 图

4.4.3 医疗服务评价体系类图

医疗服务评价体系类图展示了评价系统的结构。评价系统 (EvaluationSystem) 类负责收集提交的评价 (submitEvaluation) 和查看医生的评价 (viewEvaluations)。评价 (Evaluation) 类包含评价 ID(evaluationID)、评价用户 (user)、被评价的医生 (doctor)、评价内容 (content)、评分 (rating) 和评价日期 (datetime)。

医生 (Doctor) 类在评价体系中增加了接收评价 (receiveEvaluation) 的行为。评价 (Evaluation) 类与用户 (User) 和医生 (Doctor) 类相关联，表示评价由用户提交，并且针对特定的医生。

这些类图构成了综合医疗服务系统的基础模型，涵盖了用户交互、预约管理、问诊服务和评价体系等多个方面。通过这些类和它们之间的关系，系统能够管理用户的医疗信息、提供预约和咨询服务，并收集用户对医疗服务的评价。

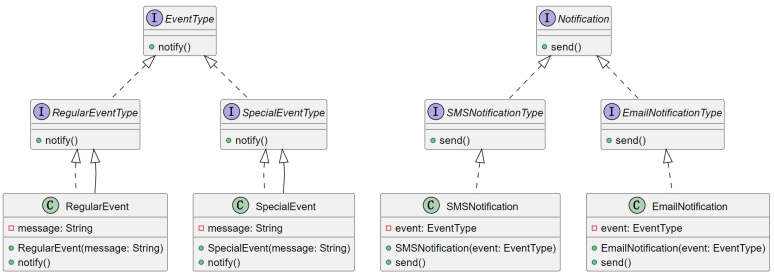

图 23: 功能状态 UML 图

4.4.4 类：普通用户（Ordinary Users）

普通用户类（ordinary users）代表了使用我们医疗预约管理系统的病人。他们可以通过用户登录功能（user login）访问系统，该功能由用户界面（User Interface）提供支持，并与用户数据库（User Database）协作以验证用户信息。普通用户还可以查询医疗举措信息（measure information），参与论坛讨论（forum），并在必要时进行论坛举报（report forum）。此外，他们可以查看医院信息（hospital information），预约医生（appointment system），使用 AI 医生进行问诊（AI doctor），并通过视频通话进行会诊（video consultation）。

类	职责	协作者
ordinary users	用户登录	User Interface
	查询举措（measure information）	Measure Database
	使用论坛（forum）	Forum
	论坛举报（report forum）	Report System
	查看医院信息（hospital information）	Hospital Database
	预约医生（appointment system）	Appointment System
	使用 AI 医生问诊（AI doctor）	AI System
	使用视频通话会诊（video consultation）	Video Conference System

表 18: 普通用户类的职责和协作者

类	职责	协作者
User Login	验证用户信息（validate user credentials）	User Database
	提供登录反馈（provide login feedback）	User Interface

表 19: 用户登录类的职责和协作者

4.4.5 类：用户登录（User Login）

用户登录类（User Login）负责处理用户的登录过程。它与用户数据库（User Database）协作以验证用户的凭据，并提供登录反馈给用户界面（User Interface）。

4.4.6 类：举措信息（Measure Information）

类	职责	协作者
Measure Information	提供举措信息（provide measure info）	Measure Database
	提供举措查询（perform measure query）	User Interface

表 20: 举措信息类的职责和协作者

举措信息类（Measure Information）负责向普通用户提供医疗举措的相关信息。它从举措数据库（Measure Database）中获取信息，并通过用户界面（User Interface）提供查询服务。

4.4.7 类：论坛（Forum）

论坛类（Forum）负责提供一个平台，让用户能够就医疗话题进行交流和讨论。它还负责管理论坛的帖子和用户评论，确保讨论的秩序和内容的适当性。此外，论坛类与用户界面（User Interface）协作，提供论坛交流功能，并与报告系统（Report System）协作，处理用户的举报。

类	职责	协作者
Forum	初始化论坛板块（initialize forum boards）	Board Management System
	管理用户帖子（manage user posts）	Post Management System
	审核用户评论（moderate user comments）	Comment Moderation Rules
	提供论坛交流（enable forum interaction）	User Interface
	处理论坛举报（handle forum reports）	Report System

表 21: 论坛类的职责和协作者

类	职责	协作者
Hospital Information	提供医院概况（provide hospital overview）	Hospital Database
	展示科室信息（display department info）	Department Catalog
	更新医院新闻（update hospital news）	News System
	提供医院查询（enable hospital search）	User Interface

表 22: 医院信息类的职责和协作者

4.4.8 类：医院信息（Hospital Information）

医院信息类（Hospital Information）负责存储和管理与医院相关的所有信息，包括医院概况、科室信息和最新新闻。它通过用户界面（User Interface）提供医院查询功能，使用户能够轻松找到所需的医院信息。

4.4.9 类：预约系统（Appointment System）

类	职责	协作者
Appointment System	检查医生可用性（check doctor availability）	Doctor Schedule
	安排预约时间（安排预约时间）	Calendar System
	确认预约（confirm appointments）	User Interface
	提供预约反馈（provide appointment feedback）	User Interface
	管理取消和重新预约（manage cancellations and reschedules）	Cancellation Policy

表 23: 预约系统类的职责和协作者

预约系统类（Appointment System）负责处理用户与医生之间的预约安排。它检查医生的时间表（Doctor Schedule），安排预约时间（Calendar System），并在用户界面（User Interface）上确认预约和提供反馈。此外，它还管理取消和重新预约的流程（Cancellation Policy）。

4.4.10 类：视频会诊（Video Consultation）

视频会诊类（Video Consultation）负责设置和管理在线视频会诊的过程。它使用视频通话系统（Video Call System）来初始化和维护会诊连接，并确保网络服务（Network Services）

类	职责	协作者
Video Consultation	初始化视频通话（initialize video calls）	Video Call System
	管理会诊连接（manage consultation connections）	Network Services
	录制会诊记录（record consultation records）	Recording System
	提供会诊反馈（provide consultation feedback）	User Interface

表 24: 视频会诊类的职责和协作者

的稳定性。此外，它还负责录制会诊记录（Recording System），并在会诊结束后通过用户界面（User Interface）提供反馈。

4.4.11 类：AI 医生（AI Doctor)

类	职责	协作者
AI Doctor	提供医疗咨询（provide medical consultation）	User Interface
	分析症状（analyze symptoms）	Symptom Analysis Engine
	推荐治疗方案（recommend treatment plans）	Treatment Database
	回答健康相关问题（answer health-related questions）	Health Knowledge Base
	更新患者健康记录（update patient health records）	Patient Record System

表 25: AI 医生类的职责和协作者

AI 医生类（AI Doctor）通过用户界面（User Interface）提供医疗咨询服务，分析用户输入的症状，并利用症状分析引擎（Symptom Analysis Engine）来识别可能的健康问题。它还可以推荐治疗方案，并从治疗数据库（Treatment Database）中检索相关信息。此外，AI 医生类负责回答用户关于健康的问题，并更新患者的健康记录。

4.4.12 类：视频会诊（Video Consultation)

类	职责	协作者
Video Consultation	初始化视频通话（initialize video calls）	Video Call System
	管理会诊连接（manage consultation connections）	Network Services
	录制会诊记录（record consultation records）	Recording System
	提供会诊反馈（provide consultation feedback）	User Interface
	调度会诊资源（dispatch consultation resources）	Resource Management System

表 26: 视频会诊类的职责和协作者

视频会诊类（Video Consultation）负责设置和管理在线视频会诊的过程。它使用视频通话系统（Video Call System）来初始化和维护会诊连接，并确保网络服务（Network Services）的稳定性。此外，它还负责录制会诊记录（Recording System），并在会诊结束后通过用户界面（User Interface）提供反馈。视频会诊类还与资源管理系统（Resource Management System）协作，以调度所需的会诊资源。

4.4.13 类：管理员（Admin)

类	职责	协作者
Admin	管理用户信息（manage user information）	User Database
	管理医生信息（manage doctor information）	Doctor Database
	管理医院信息（manage hospital information）	Hospital Database
	处理举报信息（handle report information）	Report System
	监控系统性能（monitor system performance）	Performance Monitoring Tools
	配置系统设置（configure system settings）	Configuration Manager

表 27: 管理员类的职责和协作者

管理员类（Admin）负责维护医疗预约管理系统的整体运行。它管理用户数据库（User Database）中的用户信息，包括注册、更新和删除用户账户。管理员还管理医生数据库（Doctor Database）中的医生信息，以及医院数据库（Hospital Database）中的医院信息。此外，管理员处理系统中的举报信息，并监控系统性能，确保系统稳定运行。管理员还可以配置系统设置，以优化系统性能和用户体验。

4.4.14 类：医生（Doctor)

类	职责	协作者
Doctor	提供医疗咨询（provide medical consultation）	User Interface
	管理预约信息（manage appointment information）	Appointment System
	参与视频会诊（participate in video consultations）	Video Consultation
	更新患者处方（update patient prescriptions）	Prescription Management System
	记录诊断和治疗建议（record diagnoses and treatment suggestions）	Medical Record System

表 28: 医生类的职责和协作者

医生类（Doctor）负责向病人提供专业的医疗服务。他们通过用户界面（User Interface）提供医疗咨询，并管理预约系统（Appointment System）中的预约信息。医生还参与视频会诊（Video Consultation），并使用处方管理系统（Prescription Management System）更新患

者的处方。此外，医生在医疗记录系统（Medical Record System）中记录诊断和治疗建议，以便跟踪患者的健康状况。

4.4.15 类：账单管理（Billing Management）

类	职责	协作者
Billing Management	生成和发送账单（generate and send bills） 管理账单支付（manage bill payments） 处理退款请求（process refund requests） 提供账单查询和历史记录（provide bill inquiries and history）	User Interface Payment Gateway Financial System Billing Database

表 29: 账单管理类的职责和协作者

账单管理类（Billing Management）负责处理系统中所有与账单相关的事务。它生成和发送账单给用户，管理账单支付过程，并通过支付网关（Payment Gateway）处理用户的支付。此外，它还处理用户的退款请求，并与财务系统（Financial System）协作。账单管理类还提供账单查询和历史记录的功能，让用户能够轻松访问自己的账单信息。

4.4.16 类：处方管理（Prescription Management）

类	职责	协作者
Prescription Management	创建和管理处方（create and manage prescriptions） 跟踪处方状态（track prescription status） 提供药物信息（provide drug information） 处理处方续签请求（handle prescription renewals）	Doctor Pharmacy System Drug Database Medical Record System

表 30: 处方管理类的职责和协作者

处方管理类（Prescription Management）负责创建和管理医生开具的处方。它跟踪处方的状态，并与药房系统（Pharmacy System）协作，确保药物按时送达。处方管理类还提供药物信息，并处理处方续签请求。它与医疗记录系统（Medical Record System）紧密协作，确保处方信息的准确性和可追溯性。

4.4.17 类：健康档案管理（Health Record Management）

健康档案管理类（Health Record Management）负责维护和更新病人的健康档案。它确保医疗记录的准确性和完整性，并在用户界面（User Interface）上提供健康档案的访问。健康档案管理类还保护病人的隐私，并与安全系统（Security System）协作，确保所有健康信息的安全。

类	职责	协作者
Health Record Management	维护病人健康档案（maintain patient health records） 更新医疗记录（update medical records） 提供健康档案访问（provide health record access） 保护病人隐私（protect patient privacy）	Doctor Billing Management User Interface Security System

表 31: 健康档案管理类的职责和协作者

4.4.18 类：报告和分析（Reporting & Analytics）

类	职责	协作者
Reporting & Analytics	生成医疗报告（generate medical reports） 分析医疗服务使用情况（analyze service usage） 提供决策支持（provide decision support） 监控服务质量（monitor service quality）	Health Record Management Data Analysis Tools Admin Quality Control System

表 32: 报告和分析类的职责和协作者

报告和分析类（Reporting & Analytics）负责生成医疗报告，并分析医疗服务的使用情况。它为管理员提供决策支持，并监控服务质量。报告和分析类使用数据分析工具（Data Analysis Tools）来处理和解释数据，确保管理层能够基于数据做出明智的决策。

通过添加这些类，我们的病人预约系统将能够提供更加全面和高效的医疗服务，满足病人和医疗服务提供者的各种需求。这些类的职责和协作者之间的关系确保了系统的整体性能和用户体验。

4.4.19 类：保险处理（Insurance Processing）

类	职责	协作者
Insurance Processing	验证保险信息（validate insurance information） 处理保险索赔（process insurance claims） 更新保险状态（update insurance status） 提供保险相关咨询（provide insurance consultation）	Insurance Company API Billing Management Patient Record System User Interface

表 33: 保险处理类的职责和协作者

保险处理类（Insurance Processing）负责验证病人的保险信息，并与保险公司的 API 进行交互以处理索赔。它更新病人的健康档案系统中的保险状态，并提供用户界面上的保险相关咨询服务。

4.4.20 类：预约提醒（Appointment Reminders）

类	职责	协作者
Appointment Reminders	安排预约提醒（schedule appointment reminders）	User Interface
	发送预约通知（send appointment notifications）	Notification System
	跟踪提醒状态（track reminder status）	Appointment System
	管理提醒偏好（manage reminder preferences）	User Profile System

表 34: 预约提醒类的职责和协作者

预约提醒类（Appointment Reminders）负责安排并通过通知系统（Notification System）发送预约提醒，确保病人不会错过他们的预约时间。它还跟踪提醒的状态，并管理用户在用户档案系统（User Profile System）中的提醒偏好设置。

4.4.21 类：在线支付（Online Payment）

类	职责	协作者
Online Payment	处理在线支付事务（process online payments）	Payment Gateway
	确认支付状态（confirm payment status）	Billing Management
	提供支付历史记录（provide payment history）	User Interface
	支持多种支付方式（support multiple payment methods）	Payment Service Providers

表 35: 在线支付类的职责和协作者

在线支付类（Online Payment）负责处理系统中的所有在线支付事务。它与支付网关（Payment Gateway）和支付服务提供商（Payment Service Providers）协作，确认支付状态，并在用户界面（User Interface）上提供支付历史记录。这个类支持多种支付方式，以满足不同用户的需求。

4.4.22 类：用户反馈（User Feedback）

类	职责	协作者
User Feedback	收集用户反馈（collect user feedback）	User Interface
	分析反馈数据（analyze feedback data）	Reporting & Analytics
	响应用户评论（respond to user comments）	Forum
	改善服务质量（improve service quality）	Service Improvement Team

表 36: 用户反馈类的职责和协作者

用户反馈类（User Feedback）负责收集和分析用户反馈数据，以便不断改善服务质量。它响应用户在论坛上的评论，并与服务改进团队（Service Improvement Team）协作，根据用户反馈采取行动。

4.5 数据词典

4.5.1 数据流定义表

数据流名称	来源/去向	内容	注释
用户登录信息	普通用户 → 系统	用户名，密码	用于验证用户的身份
预约请求	普通用户 → 预约系统	用户 ID，医生 ID，预约时间	用于创建新的预约
预约反馈	预约系统 → 普通用户	预约状态，预约时间，医生信息	用于告知用户预约的结果
医生信息查询	普通用户 → 医生数据库	医生 ID	用于获取医生的详细信息
医生信息	医生数据库 → 普通用户	医生姓名，专长，可预约时间	用于展示医生的详细信息
举报信息	普通用户 → 管理员	用户 ID，被举报内容，举报理由	用于处理用户的举报
管理操作	管理员 → 用户数据库/医生数据库/医院数据库	操作类型，目标 ID，新信息	用于更新数据库的信息
AI 问诊请求	普通用户 → AI 医生	用户 ID，病症描述	用于 AI 医生进行诊断
AI 诊断结果	AI 医生 → 普通用户	诊断结果，建议的治疗方案	用于告知用户 AI 医生的诊断结果
视频会诊请求	普通用户 → 视频会诊	用户 ID，医生 ID，预约时间	用于安排视频会诊
视频会诊反馈	视频会诊 → 普通用户	会诊状态，会诊时间，医生信息	用于告知用户会诊的结果

表 37: 系统中的数据流定义

4.5.2 数据元素定义表

数据元素名称	别名	定义	来源	示例	注释
用户 ID	UID	用户在系统中的唯一标识符	用户注册时生成	123456	用于识别用户
用户名	Username	用户在系统中的名称	用户在注册时提供	JohnDoe	用于显示用户的名称
密码	Password	用户的登录密码	用户在注册时提供	Password123	用于验证用户的身份
医生 ID	DID	医生在系统中的唯一标识符	医生注册时生成	D78901	用于识别医生
医生姓名	DoctorName	医生的姓名	医生在注册时提供	Dr. Smith	用于显示医生的姓名
专长	Specialty	医生的专长领域	医生在注册时提供	心脏病专家	用于帮助用户选择合适的医生
可预约时间	AvailableTime	医生可接受预约的时间	医生提供	2024 年 3 月 25 日 10:00-12:00	用于安排预约
预约时间	AppointmentTime	用户预约的时间	用户在预约时提供	2024 年 3 月 25 日 11:00	用于安排预约
病症描述	SymptomDescription	用户描述的症状信息	用户在 AI 问诊或视频会诊时提供	头痛，恶心	用于 AI 医生诊断或医生会诊
诊断结果	DiagnosisResult	AI 医生或医生的诊断结果	AI 医生或医生提供	脑震荡	用于告知用户诊断结果
治疗方案	TreatmentPlan	AI 医生或医生的建议治疗方案	AI 医生或医生提供	休息，避免剧烈运动	用于指导用户的治疗

表 38: 系统中的数据元素定义

4.5.3 外部项定义表

外部项名称	类型	来源/目的地	内容	注释
用户登录输入	输入	普通用户	用户名，密码	用于验证用户的身份
登录反馈输出	输出	普通用户	用户登录状态,用户信息（如果登录成功）	用于告知用户登录的结果
预约请求输入	输入	普通用户	用户 ID，医生 ID，预约时间	用于创建新的预约
预约反馈输出	输出	普通用户	预约状态，预约时间，医生信息	用于告知用户预约的结果
医生信息查询输入	输入	普通用户	医生 ID	用于获取医生的详细信息
医生信息反馈输出	输出	普通用户	医生姓名，专长，可预约时间	用于展示医生的详细信息
AI 问诊请求输入	输入	普通用户	用户 ID，病症描述	用于 AI 医生进行诊断
AI 诊断反馈输出	输出	普通用户	诊断结果,建议的治疗方案	用于告知用户 AI 医生的诊断结果
视频会诊请求输入	输入	普通用户	用户 ID，医生 ID，预约时间	用于安排视频会诊
视频会诊反馈输出	输出	普通用户	会诊状态，会诊时间，医生信息	用于告知用户会诊的结果
管理操作输入	输入	管理员	操作类型,目标 ID,新信息	用于更新数据库的信息
管理操作反馈输出	输出	管理员	操作结果，新的信息（如果操作成功）	用于告知管理员操作的结果

表 39: 系统中的外部项定义

4.5.4 数据精度表

数据元素名称	数据类型	精度	最小值	最大值	注释
用户 ID	整数	N/A	1	99999999	用户 ID 应为唯一的正整数
用户名	字符串	N/A	N/A	50 个字符	用户名应不超过 50 个字符
密码	字符串	8 个字符	20 个字符	N/A	密码应包含 8 到 20 个字符
医生 ID	整数	N/A	1	99999999	医生 ID 应为唯一的正整数
医生姓名	字符串	N/A	N/A	50 个字符	医生姓名应不超过 50 个字符
专长	字符串	N/A	N/A	100 个字符	专长描述应不超过 100 个字符
可预约时间	日期和时间（分、秒）	精确到分钟	2024 年 1 月 1 日 00:00	2030 年 12 月 31 日 23:59	可预约时间应精确到分钟
预约时间	日期和时间（分、秒）	精确到分钟	2024 年 1 月 1 日 00:00	2030 年 12 月 31 日 23:59	预约时间应精确到分钟
病症描述	字符串	N/A	N/A	500 个字符	病症描述应不超过 500 个字符
诊断结果	字符串	N/A	N/A	500 个字符	诊断结果应不超过 500 个字符
治疗方案	字符串	N/A	N/A	1000 个字符	治疗方案应不超过 1000 个字符

表 40: 系统中的数据精度要求

4.6 数据流图

4.6.1 指令数据流图

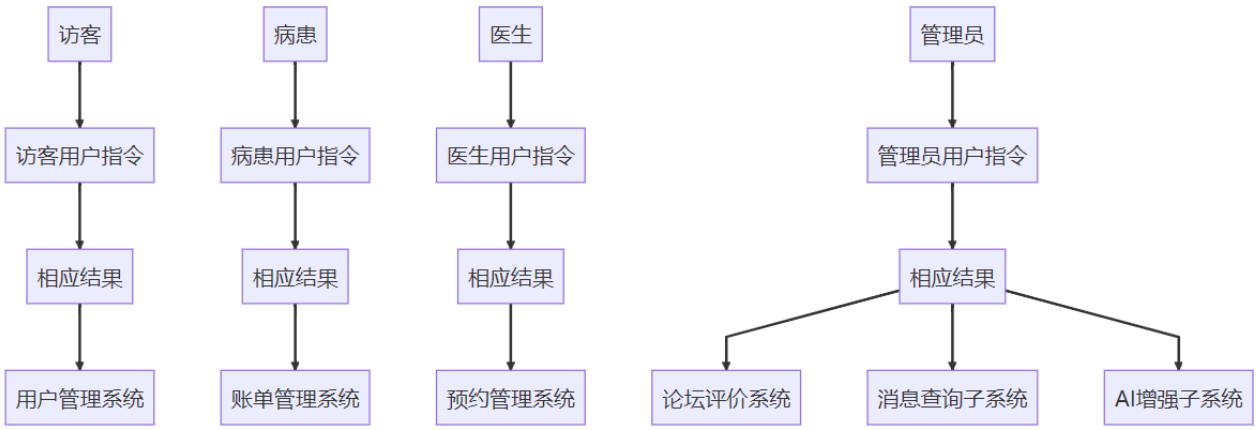

图 24: 指令数据流图

4.6.2 顶层数据流图

下面是整体的顶层数据流图，由于空间所限，这里放的图较小，更大的可见于附录??。该顶层数据流图，描述了用户从注册登录到使用各种服务的数据流动过程：

- **用户注册和登录**：用户访问注册页面，填写并提交注册信息。系统验证信息后完成注册，发送确认邮件。用户点击邮件链接激活账户，然后登录系统。
- **个人医疗信息管理**：用户登录后可以查看、更新医疗信息，授权他人访问并查看访问记录，接收系统提示和健康建议。
- **AI 医疗咨询服务**：用户提交症状描述，AI 系统分析并给出初步建议。用户反馈后，如有需要医生会审核。系统跟踪跟进病情。
- **科室和专业介绍**：用户登录后导航到科室页面，选择并查看感兴趣的科室信息，做出选择。
- **医生预约时间段选择**：用户查看医生列表和可用时间段，选择预约时间段并确认。系统发送确认信息，用户可提交医生评价。
- **医生选择和预约确认**：用户查看医生资料，在线咨询后选择医生。选择就诊时间段，填写预约信息并确认。系统跟踪预约状态。
- **医疗账单管理**：用户提交医疗账单后，可以查看明细。医生或相关人员审核验证后，用户可对项目提出争议。系统发送最终明细，用户在线支付。
- **支付和退款流程**：用户查看未支付费用，选择支付方式完成支付。系统发送确认信息。用户可查看退款政策，申请退款后相关人员审核。系统处理退款并通知用户。

4.6.3 中层数据流图

下面是整体的中层数据流图，描述了用户从注册登录到使用各种服务的数据流动过程。主要包含下面的内容：

- **用户注册和认证**：用户开始注册，输入并提交注册信息。系统验证信息后，发送确认邮件。用户确认邮件后，注册完成。用户登录时，输入凭证进行验证。验证通过后，用户登录成功。
- **个人健康记录管理**：用户访问健康记录，更新记录并提交。系统验证更新后，更新成功。用户可授权他人访问记录并进行审核。
- **AI 辅助咨询**：用户描述症状，AI 处理后给出建议。用户与 AI 交互，如有需要可请求医生审核。
- **预约管理**：用户查看医生可用时间，选择时间段并输入个人信息。提交预约请求后，系统确认时间段，用户可设置提醒和进行就诊后操作。
- **账单和支付管理**：用户查看费用，费用明细后提交支付。系统验证支付后，支付确认。如有差异，用户可提出争议并解决。
- **退款流程**：用户请求退款，管理员审核并处理请求。退款获批后，系统发行退款；否则，退款被拒绝。

4.6.4 病人医生交互

该交互图25，主要包含下面的内容：

- **病人提交咨询**：病人向医院系统提交咨询。
- **生成病历**：系统根据咨询生成病历。
- **医疗记录**：病历包括过去的测试结果。
- **查询病历**：医生可以查询病历。
- **查看病历**：医生查看病历并进行诊断。
- **诊断结果**：医生给出诊断结果，可能包括开具药方或建议检查。
- **建议检查和药方**：系统将建议检查和药方发送给病人。
- **填写反馈**：病人填写反馈信息。
- **更新病历**：系统更新病历以包括病人的反馈。
- **医院系统**：以上所有交互都发生在医院系统中。

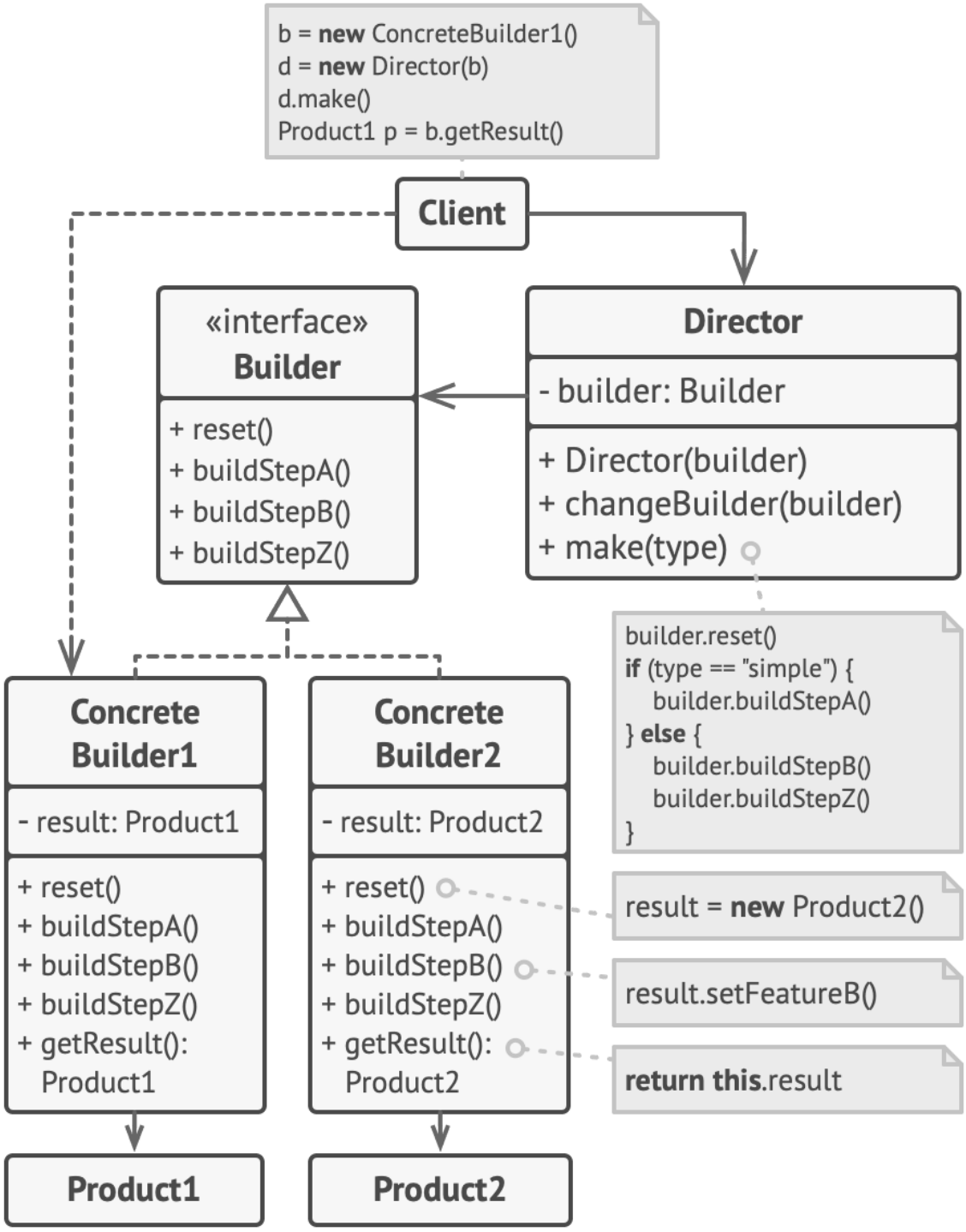
b = new ConcreteBuilder1()
d = new Director(b)
d.make()
Product1 p = b.getResult()
Client
«interface»
Builder
+ reset()
+ buildStepA()
+ buildStepB()
+ buildStepZ()
Director
- builder: Builder
+ Director(builder)
+ changeBuilder(builder)
+ make(type)
builder.reset()
if (type == "simple") {
builder.buildStepA()
} else {
builder.buildStepB()
builder.buildStepZ()
}
Concrete
Builder1
- result: Product1
+ reset()
+ buildStepA()
+ buildStepB()
+ buildStepZ()
+ getResult():
Product1
Concrete
Builder2
- result: Product2
+ reset()
+ buildStepA()
+ buildStepB()
+ buildStepZ()
+ getResult():
Product2
result = new Product2()
result.setFeatureB()
return this.result
Product1
Product2

图 25: 病人医生交互图

5 服务端

5.1 服务端硬件配置需求

5.2 服务器架构

本部分将详细介绍医疗预约管理系统服务器端的架构设计。服务器端的设计采用了分层架构，具体分为四个层次：Controller 层、Service 层、Mapper 层和 Model 层。

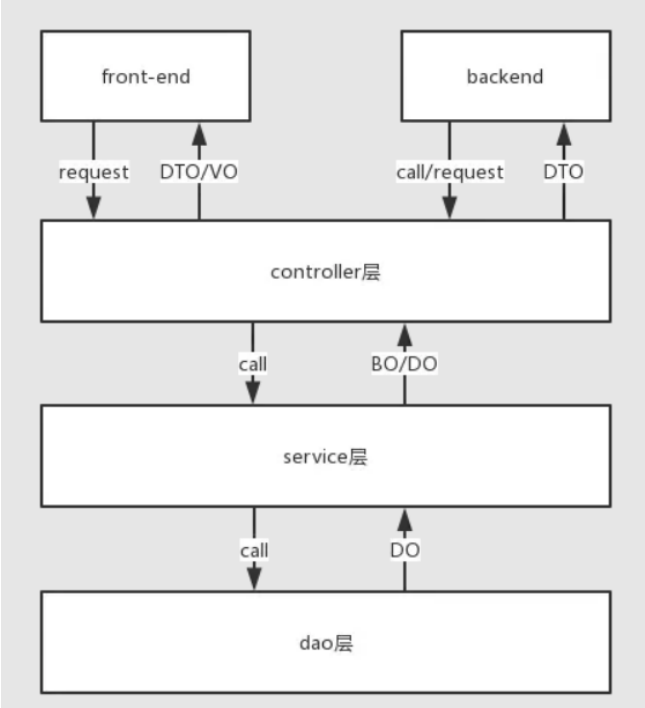

图 26: 代码结构

5.2.1 Controller 层

- **处理 HTTP 请求：**Controller 层接收来自前端的 HTTP 请求，并根据请求类型（如 GET、POST、PUT、DELETE）和请求的资源路径来执行相应的操作。
- **提取和验证数据：**从 HTTP 请求中提取必要的数据，并对其进行验证，确保数据的合法性和完整性。
- **调用业务逻辑：**将提取的数据传递给 Service 层，由 Service 层执行具体的业务逻辑处理。
- **返回响应：**根据 Service 层处理的结果，构造合适的 HTTP 响应并发送回客户端。
- **异常处理：**处理请求处理过程中可能发生的异常，并返回适当的错误响应。

Controller 层的 UI 设计原则

在用户界面（UI）设计中，Controller 层遵循以下原则以提升用户体验：

- **用户控制原则**：确保用户能够通过直观的操作来控制系统，例如，通过简化的导航和清晰的指令来引导用户。
- **减轻记忆负担原则**：减少用户需要记忆的信息量，例如，通过使用自动完成、历史记录或提供清晰的指引来减少用户的记忆负担。
- **界面一致性原则**：保持界面的一致性，使得用户无论在哪个模块都能有相同的操作体验，例如，使用统一的布局、颜色方案和控件。

Controller 层的设计旨在为用户提供一个直观、易用且响应迅速的交互界面，同时也确保了系统的灵活性和可维护性。通过精心设计的 Controller 层，我们的医疗预约管理系统能够提供高效、个性化的服务，满足不同用户的需求，从而提升整体的医疗服务体验。

在医疗预约管理系统的服务器端设计中，Service 层扮演着核心角色，负责实现系统的业务逻辑。以下是 Service 层及其相关层次的详细描述：

Service 层

Service 层是系统的核心业务逻辑层，它处理用户请求并执行相应的业务逻辑。在这一层中，我们细分了以下模块：

- **事件类**：负责管理所有事件信息，例如预约挂号和健康咨询等医疗服务事件。
- **关系类**：建立事件与提醒之间的联系，并定义它们之间的各种关系，如时间或地点的关联。
- **提醒类**：规定在特定事件发生时应采取的行动，例如发送通知给用户或启动提醒机制。

Mapper 层（Dao 层）

Mapper 层，也称为数据访问对象（Dao）层，是系统的数据访问层。它作为系统与数据库之间的桥梁，提供数据访问接口，使得 Service 层能够安全、准确地读写数据库中的数据。这一层的存在确保了系统的稳定性和数据的完整性。

Model 层

Model 层包含了与数据库表字段相对应的实体类。这些实体类封装了数据，并提供了标准的 set/get 方法，以支持数据的操作和访问。Model 层是 Service 层和 Mapper 层之间数据交互的媒介。

通过 Service 层、Mapper 层和 Model 层的协同工作，医疗预约管理系统能够为病人提供以下一站式医疗服务：

- 用户注册与登录，确保用户能够安全地访问系统服务。
- 查看和管理个人健康信息，提供个性化的医疗信息管理。
- AI 病情咨询服务，利用人工智能技术为病人提供初步的医疗建议。
- 科室选择和医生预约挂号，优化就诊流程，提高医疗服务的可及性和效率。

系统还支持电子问诊单的接收，便于病人记录和后续跟进，确保医疗服务的连续性。此外，病人可以通过参与医疗服务评价体系，为医疗服务提供宝贵的反馈，帮助医疗机构不断改进服务质量。

在技术实现方面，项目采用了面向对象的设计原则，前端框架的构建是开发团队的主要责任。我们选择了 Vue.js 作为前端框架，并使用 JavaScript 作为编程语言，以实现高效、响应式的用户界面。

总体而言，这些功能的实现旨在为病人提供一个无缝、高效的医疗服务体验。通过提升医疗服务的可及性和效率，这些设计不仅满足了病人的需求，也为医疗服务的整体改进和发展做出了重要贡献。

在医疗预约管理系统的服务器端设计中，除了核心的业务逻辑层和服务层之外，还需要考虑数据的安全性和稳定性。以下是数据备份与恢复以及安全性保障的详细描述：

数据备份与恢复

为了确保系统的稳定性和数据的安全性，服务器软件必须提供可靠的数据备份和恢复机制。这一机制的重要性体现在以下几个方面：

- **故障恢复**：在软硬件故障发生时，系统必须能够利用备份的数据迅速恢复运行环境，以减少系统停机时间，保证服务的连续性。
- **数据完整性**：用户信息管理模块的安全性至关重要，必须确保用户信息的完整性和准确性，包括个人资料、预约记录、健康信息等。
- **统计数据的准确性**：系统应定期进行数据备份，确保在任何时候都能够恢复到一个已知的、准确的数据状态。

为了实现这些目标，可以采取以下措施：

- 实施定期自动备份策略，包括全量备份和增量备份。
- 确保备份数据的存储安全，避免数据丢失或损坏。
- 测试备份数据的恢复流程，确保在紧急情况下能够快速有效地恢复数据。

安全性保障

安全性是医疗预约管理系统的另一个关键考虑因素。系统应实施以下加密措施来保障数据的保密性和安全性：

- **登录数据加密**：对用户的登录信息，如用户名和密码，进行加密处理，以防止未授权访问。
- **传输数据加密**：对在系统间传输的数据进行加密，无论是用户信息还是医疗记录，都应确保在传输过程中不被截获或篡改。
- **数据存储加密**：对存储在数据库中的敏感数据进行加密，即使数据被非法访问，也无法直接读取内容。

此外，系统还应采取其他安全措施，如：

- 实施强密码策略和定期密码更换提醒。
- 使用防火墙和入侵检测系统来防止未经授权的访问。
- 定期进行安全审计和漏洞扫描，以发现并修复潜在的安全问题。

通过这些措施，医疗预约管理系统能够为病人提供一个安全、可靠的医疗服务体验，保护病人的隐私和数据安全，同时也符合医疗行业的安全标准和法规要求。

6 总结

这份设计报告由黄鸿宇、周炜、沈小康三位完成，目的在于详细阐述智慧医疗预约管理系统的设计。报告深入探讨了系统的功能需求、性能标准、用户界面设计原则、后端逻辑处理机制、系统架构构建、技术选型策略、数据库设计方法以及 API 设计等关键技术要素。特别强调了确保系统设计的合理性和可交互性。

参考文献

[1] Andrea Burattin, Hugo A López, and Lasse Starklit. A monitoring and discovery approach for declarative processes based on streams. *arXiv preprint arXiv:2208.05364*, 2022.

[2] Mojtaba Eshghie, Wolfgang Ahrendt, Cyrille Artho, Thomas Troels Hildebrandt, and Gerardo Schneider. Capturing smart contract design with dcr graphs. In *International Conference on Software Engineering and Formal Methods*, pages 106–125. Springer, 2023.

[3] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Pearson Deutschland GmbH, 1995.

[4] Erich Gamma, Richard Helm, Ralph E Johnson, and John Vlissides. *设计模式: 可复用面向对象软件的基础*. Ji xie gong ye chu ban she, 2019.

[5] Lukas Heiland, Marius Hauser, and Justus Bogner. Design patterns for ai-based systems: A multivocal literature review and pattern repository. In *2023 IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI (CAIN)*, pages 184–196. IEEE, 2023.

[6] Wei Liu, Li Xia, Junyang Yu, and Xindi Huang. *Design Patterns*. Tsinghua University Press, 2nd edition edition, 12 2018.

[7] Katerina Paltoglou, Vassilis E Zafeiris, NA Diamantidis, and Emmanouel A Giakoumakis. Automated refactoring of legacy javascript code to es6 modules. *Journal of Systems and Software*, 181:111049, 2021.

[8] Kamalmeet Singh. *Java Design Patterns and Practices*. DianDu Education Bookstore, 07 2019. PDF available for download on Baidu Netdisk.