

华北水利水电大学

North China University of Water Resources and Electric Power

《软件工程概论》本科生课程结课报告

医院挂号系统设计与开发

专业	:	人工智能
学号	:	20218526
班级	:	2020185
姓名	:	高树林

2023 年 6 月 20 日

目录

1	可行性分析	1
1.1	项目概述	1
1.2	项目论证	2
1.3	项目计划	3
1.4	可行性分析	4
2	总体设计	6
2.1	功能需求分析	6
2.2	性能分析	7
2.3	数据需求分析	9
3	详细设计	11
3.1	处理流程	11
3.2	总体结构设计	12
3.3	数据结构	14
4	系统实现	26
4.1	实现工具	26
4.2	数据库	27
4.3	开发平台	28
5	测试报告书	30
5.1	测试方法	30
5.2	测试示例	31
	参考文献	34

1 可行性分析

1.1 项目概述

医院作为人们日常生活中重要的健康服务提供者，其管理效率和服务质量对于社会公众的健康至关重要。为了提升医院的管理效能和服务水平，我们计划引入一套先进的医院挂号系统。该系统旨在开发一个高效、可靠的医院挂号系统，为患者和医生提供便捷的挂号服务，优化医院的运营效率。该系统将使用现代化的技术和流程，以提高医疗服务的质量和效率为目标。

该医院挂号系统将具备以下主要功能：

1. 患者预约挂号：患者可以通过在线平台或移动应用程序预约医生的挂号。系统将提供医生的详细信息、专长、时间表和可预约的挂号数量等相关信息，以便患者做出明智的选择。
2. 智能排队管理：通过该系统，患者可以根据自己的需求和医生的排班情况，选择合适的时间段进行预约挂号。系统将自动分配合理的时间和顺序，以最大程度地减少患者的等待时间，并提供实时的排队信息。
3. 医生信息管理：系统将维护医生的个人资料、专业背景、工作经验和就诊时间表等信息，并提供快速搜索和筛选功能，以便患者选择适合自己需求的医生。
4. 患者档案管理：系统将保存患者的基本信息、病历记录、药物过敏信息和诊断结果等重要数据，确保医生能够方便地查看和更新患者的医疗历史，提供更加个性化的治疗服务。
5. 费用结算与支付：系统将提供便捷的费用结算和支付功能，包括在线支付、医保报销和第三方支付等选项，为患者提供灵活的支付方式，并确保费用的准确计算和及时结算。
6. 医生工作站：系统将提供医生工作站，让医生能够轻松查看和管理患者的预约、诊断结果和处方等信息，提高工作效率，并支持电子处方和电子病历的生成，以减少纸质文档的使用。
7. 管理报表和统计：系统将生成各类报表和统计数据，用于医院管理层进行绩效评估、资源分配和决策制定。这些报表包括挂号量、医生工作量、患者满意度等指标，帮助医院

管理层及时调整和改进医疗服务。

通过该医院挂号系统，患者能够更加方便地预约挂号、减少患者的等待时间和排队繁琐的过程。患者可以在任何时间、任何地点通过网络或移动应用程序进行预约挂号，无需亲自前往医院排队等候。系统将根据患者的需求和医生的排班情况，智能地分配挂号时间，减少了患者在医院等待的时间，并提供实时的排队信息，让患者可以更好地安排自己的时间。

此外，医生也将从该系统中受益。他们可以方便地查看和管理患者的预约信息、诊断结果和处方等内容。医生工作站提供了一个集中管理患者信息的平台，使医生能够更加高效地提供个性化的诊疗服务。系统还支持电子处方和电子病历的生成，减少了纸质文档的使用，提高了医生的工作效率。

管理层可以通过系统生成的各类报表和统计数据进行评估、资源分配和决策制定。他们可以根据挂号量、医生工作量和患者满意度等指标，及时调整和改进医疗服务，提升医院的整体运营效率和服务质量。

该医院挂号系统的目标是通过现代化的技术和流程，为患者提供便捷的预约挂号服务，减少患者的等待时间和排队过程，提高医生的工作效率和个性化诊疗服务，以及为管理层提供数据支持，实现医院运营的优化和改进。

1.2 项目论证

1.2.1 现状分析

目前，许多医院仍然采用传统的人工挂号方式，导致患者排队等待时间长、挂号效率低下，并且易出现挂号信息错误的问题。此外，医院管理者在人工处理大量挂号信息时容易出现疏忽，造成资源浪费和服务不均衡。

1.2.2 问题识别

传统的人工挂号方式存在以下问题：

- a) 人工排队等待时间长，影响患者就医体验；
- b) 容易出现挂号信息错误，导致就诊混乱；
- c) 医院管理者难以实时获取和分析挂号数据，影响决策效率。

1.2.3 解决方案：

引入一套先进的医院挂号系统可以解决上述问题：

- a) 通过在线挂号，患者可以避免长时间排队等待，提升就医体验；
- b) 引入自动化挂号流程，减少人为错误，提高挂号准确性；
- c) 提供实时数据分析和报告功能，帮助医院管理者优化资源配置和决策。

1.3 项目计划

1.3.1 项目目标：

开发和部署一套全面的医院挂号系统，实现以下目标：

- a) 提高挂号效率，缩短患者等待时间；
- b) 提升挂号准确性，减少信息错误；
- c) 提供数据分析和报告功能，优化医院资源配置。

1.3.2 项目步骤：

- a) 需求分析：与医院管理部门合作，明确系统功能和性能要求；
- b) 系统设计：基于需求分析结果，设计系统架构和界面；
- c) 开发实施：编码、测试和部署系统；
- d) 数据迁移：将现有挂号数据迁移到新系统中；
- e) 培训和推广：培训医院工作人员并推广系统使用；
- f) 系统维护：持续监控系统运行状态，及时处理问题。

1.3.3 时间规划：

时间规划是指在项目实施过程中，对各个任务和阶段进行时间安排和计划的过程。通过合理的时间规划，可以确保项目按时完成，并且有效地管理项目资源和进度。以下是医院挂号系统项目的时间规划示例：

- 1) 需求分析阶段：预计持续2周，包括与医院管理部门沟通、收集需求和定义系统功能和性能要求。
- 2) 系统设计阶段：预计持续3周，包括系统架构设计、界面设计和数据库设计等。
- 3) 开发实施阶段：预计持续8周，包括编码、测试和系统部署等。
- 4) 数据迁移阶段：预计持续1周，将现有挂号数据迁移到新系统中，并确保数据的完整性和准确性。

5) 培训和推广阶段：预计持续2周，包括培训医院工作人员使用新系统，并进行推广和宣传。

6) 系统维护阶段：持续跟踪和监控系统运行状态，及时处理问题，保证系统的稳定性和可用性。

总计预计项目实施时间为16周，其中包括开发和实施阶段的主要工作。然而，实际的时间规划可能会根据项目的规模、复杂性和资源可用性等因素进行调整和优化。

1.4 可行性分析

在决定实施医院挂号系统项目之前，进行可行性分析是必要的。本文将从技术可行性、经济可行性和操作可行性三个方面对该项目进行评估，以确保项目的成功实施和运行。

1.4.1 技术可行性

技术评估：医院挂号系统依赖于先进的信息技术和软件开发，需要评估现有技术是否能够满足项目需求。通过分析市场上的可行解决方案和技术趋势，确认项目的技术可行性。

系统集成：医院挂号系统需要与现有医院管理系统、电子病历系统等进行集成，确保数据的交互和一致性。通过评估系统集成的难度和可行性，确定系统能否无缝融入现有医院环境。

安全性考虑：医院挂号系统涉及患者的敏感信息，包括个人身份和健康记录等。评估系统的安全性要求和数据保护措施，确保系统的安全性和隐私保护。

1.4.2 经济可行性

成本评估：评估项目的开发成本、硬件设备和软件许可费用，包括系统开发、系统集成、培训和推广等。同时，考虑项目带来的长期运维成本和系统升级的成本。

收益预测：分析医院挂号系统的预期收益，如患者就医效率的提升、医院资源的优化配置和患者满意度的提高等。通过与医院管理部门合作，进行收益预测和效益评估，确保项目的经济可行性。

投资回报周期：评估医院挂号系统项目的投资回报周期，即项目实施后多长时间能够回收项目投资成本，并开始产生正面的经济效益。确保项目的投资回报周期在可接受的范围内。

1.4.3 操作可行性

人员培训：评估医院现有人员的技术水平和培训需求，以确保他们能够熟练使用新的挂号系统。制定培训计划和培训材料，提供必要的技术支持，确保系统的顺利过渡和运行。

用户接受度：通过与医院管理部门和患者进行沟通，了解他们对新系统的期待和接受程度。收集用户反馈和意见，对系统进行必要的调整和改进，以提高用户满意度和系统的操作可行性。

运营支持：评估医院现有的技术支持和运维团队的能力，确保能够提供系统的稳定运行和及时的技术支持。制定运营支持计划，包括故障排除、系统更新和性能监控等，以保证系统的持续可用性和稳定性。

综上所述，通过对医院挂号系统项目的技术可行性、经济可行性和操作可行性进行评估，可以得出以下结论：技术上，现有的信息技术和系统集成能力支持项目的实施，确保系统的稳定性和安全性。经济上，项目的投资成本和预期收益相符，且投资回报周期在可接受的范围内。操作上，医院现有的人员培训和支持体系可满足项目的需求，用户接受度较高。

因此，基于可行性分析的结果，医院挂号系统项目具备可行性，值得进一步推进和实施。在项目实施过程中，应充分考虑技术、经济和操作方面的要求，并与医院管理部门、用户和相关利益相关方密切合作，确保项目的顺利实施和成功运营。

2 总体设计

2.1 功能需求分析

目功能需求分析是医院挂号系统项目中至关重要的一部分，它涉及对系统功能的详细定义和规划。在功能需求分析中，从功能和流程两个方面来说明系统的需求。

2.1.1 功能需求分析

1. 患者挂号功能：

- 提供在线挂号功能，允许患者通过系统选择科室、医生和就诊时间。
- 支持患者填写个人信息、病历和过敏史等相关信息。
- 自动生成挂号号码和就诊时间，并向患者发送挂号确认信息。

2. 医生管理功能：

- 医生信息管理，包括医生个人信息、专长、排班和出诊时间等。
- 支持医生就诊时间的设置和调整，确保挂号系统与医生的实际出诊时间一致。
- 医生值班表管理，确保医生的排班信息准确无误。

3. 排队管理功能：

- 实时显示患者的挂号信息、就诊进度和等待时间。
- 提供分诊功能，将患者根据病情严重程度和就诊需求分配到不同的医生或科室。
- 提供排队叫号功能，通知患者就诊的顺序和窗口号码。

4. 就诊记录和病历管理功能：

- 记录患者的就诊历史、诊断结果、治疗方案和药物处方等信息。
- 支持医生查看和更新患者的病历记录，并提供数据导出和打印功能。

5. 数据统计和分析功能：

- 收集和分析挂号数据，包括患者数量、科室就诊量和医生工作量等。
- 提供报表和图表展示，帮助医院管理者了解挂号情况和资源利用情况。
- 支持生成统计报告，用于决策和优化医院资源配置。

2.1.2 流程分析

1. 患者挂号流程：

- 患者通过系统选择科室和医生，填写个人信息并提交挂号请求。
- 系统验证患者信息的准确性和合法性，并生成挂号记录。
- 患者收到挂号确认信息，包括挂号号码和就诊时间。

2. 排队和分诊流程：

- 患者到达医院后，提供挂号号码或个人信息以确认身份。
- 系统根据挂号号码和就诊进度，显示患者的等待时间和就诊窗口。
- 医院工作人员根据患者的病情严重程度和就诊需求，将其分配到相应的医生或科室。
- 分诊结果更新至系统，并通知患者就诊的顺序和窗口号码。

3. 就诊记录和病历管理流程：

- 医生根据患者的病情进行诊断和治疗，并记录相关的就诊信息。
- 就诊信息包括诊断结果、治疗方案、药物处方等，更新至患者的电子病历中。
- 医生可以随时查看和更新患者的病历记录，确保病历的完整性和准确性。

4. 数据统计和分析流程：

- 系统收集挂号数据、就诊数据和医生工作量等相关信息。
- 数据经过统计和分析，生成报表和图表展示，反映挂号情况和资源利用情况。
- 医院管理者可以根据报表和图表的分析结果，进行决策和优化医院资源配置。

通过功能需求分析和数据流分析，我们明确了医院挂号系统的各项功能和相关数据流程。这些功能和数据流程的定义和规划将为系统的开发和实施提供指导，并确保系统能够顺利满足医院挂号管理的需求，提升患者就医体验和医院工作效率。

2.2 性能分析

目性能分析是医院挂号系统项目中至关重要的一部分，它涉及对系统的性能需求和性能指标进行评估和定义。以下是对医院挂号系统的性能分析的一些关键考虑因素：

1. 响应时间：

- 系统应能够在患者提交挂号请求后快速响应，提供即时的挂号确认信息。
- 医生或工作人员在处理挂号请求、查询患者信息和更新病历等操作时，系统应具

备高效的响应速度。

2. 并发处理能力:

- 医院挂号系统可能面对大量的并发操作，如多个患者同时提交挂号请求或医生同时查询患者信息。

- 系统应具备良好的并发处理能力，能够有效地处理和响应多个同时进行的操作，以保证系统的稳定性和性能表现。

3. 可扩展性:

- 随着医院规模和患者数量的增加，系统需要具备良好的可扩展性，能够支持更多的用户和同时进行的操作。

- 系统的架构和设计应考虑到未来的扩展需求，可以通过增加服务器、优化数据库等方式进行扩展。

4. 数据库性能:

- 医院挂号系统需要高效地管理和存储大量的患者信息、挂号记录和病历数据等。

- 数据库应具备良好的性能，能够快速响应查询和更新操作，并具备有效的索引和优化策略，提高数据访问的效率。

5. 安全性和数据保护:

- 医院挂号系统涉及患者的个人隐私和敏感信息，包括身份证号码、病历记录等。

- 系统应具备高度的安全性，包括访问控制、加密传输、数据备份和恢复等措施，保护患者数据的安全和完整性。

6. 错误处理和容错性:

- 系统应具备良好的错误处理机制，能够捕捉和处理各种异常情况，如用户输入错误、系统故障等。

- 系统应具备一定的容错性，即在发生错误或故障时，能够提供适当的错误提示或备用方案，以保证用户的良好体验。

通过对医院挂号系统的性能分析，可以确保系统能够满足高并发、快速响应和安全可靠等关键性能要求。同时，对系统进行性能测试和优化，以确保系统在实际运行中能够稳定、高效地运行。以下是一些性能优化的策略和方法：

1. 资源优化：对系统的硬件和软件资源进行合理配置和利用，确保系统具备足够的计算能力、存储空间和网络带宽。通过监控系统的资源利用率，及时进行调整和扩展。

2. 缓存机制：使用缓存技术减轻数据库负载，提高系统的响应速度。将经常访问的数据存储在缓存中，减少数据库查询次数。

3. 并发控制：采用合适的并发控制策略，如锁机制、事务管理等，避免数据冲突和并发操作的竞争。同时，合理设计数据库表结构和索引，提高数据库的并发性能。

4. 前端优化：优化前端页面的加载速度和用户交互体验，减少不必要的网络请求和数据传输。使用合适的前端框架和技术，减少页面渲染时间。

5. 异步处理：采用异步处理机制，将耗时的操作放到后台进行，避免阻塞用户界面和影响系统的响应速度。例如，将邮件通知、数据同步等操作放入后台任务队列进行处理。

6. 日志和监控：记录系统的日志信息，包括错误日志、性能日志等，方便进行系统性能分析和问题排查。同时，建立监控系统，实时监测系统的运行状态和性能指标，及时发现并解决潜在的性能问题。

7. 安全防护：采取合适的安全措施，保护系统免受恶意攻击和数据泄露。包括防火墙、入侵检测系统、加密通信等技术手段，确保系统的安全性和可靠性。

通过以上性能分析和优化措施，可以提升医院挂号系统的响应速度、并发处理能力和安全性，确保系统能够高效、稳定地运行，满足用户和医院管理者的需求。同时，不断进行性能监测和调优，保持系统的持续性能优化。

2.3 数据需求分析

数据需求分析是医院挂号系统项目中至关重要的一部分，它涉及对系统所需的数据进行详细定义和规划。以下是对医院挂号系统的数据需求分析的一些关键考虑因素：

1. 患者数据：

- 患者个人信息：包括姓名、性别、年龄、联系方式等基本信息。
- 患者病历信息：包括既往病史、过敏史、家族病史等与就诊相关的信息。
- 患者挂号记录：包括挂号时间、科室、医生、就诊状态等挂号相关的信息。

2. 医生数据：

- 医生个人信息：包括姓名、性别、职称、专长等基本信息。

- 医生排班信息：包括出诊时间、科室、就诊类型等排班相关的信息。

3. 科室和诊室数据：

- 科室信息：包括科室名称、科室负责人、科室电话等科室相关的信息。

- 诊室信息：包括诊室编号、位置、设备等与诊室使用相关的信息。

4. 就诊记录和病历数据：

- 就诊记录：包括就诊时间、挂号号码、诊断结果、治疗方案等与就诊相关的信息。

- 电子病历：包括患者的诊断记录、检查结果、药物处方等与患者健康管理相关的信息。

5. 挂号统计和报表数据：

- 挂号统计数据：包括每日挂号量、就诊科室统计、医生工作量等与挂号统计相关的信息。

- 报表数据：包括生成的统计报表、图表展示等用于决策和资源优化的数据。

6. 系统日志数据：

- 系统操作日志：记录用户的操作日志，包括登录、挂号、查询等操作的记录。

- 错误日志：记录系统发生的错误和异常情况，方便故障排查和问题解决。

在数据需求分析中，需要明确每个数据项的属性、关联关系、访问权限等。同时，需要考虑数据的安全性、一致性和完整性，确保数据的准确性和可靠性。

通过数据需求分析，可以确保医院挂号系统能够满足各种业务需求和数据管理的要求。同时，合理规划和设计数据库结构、数据接口和数据流程，确保数据的有效管理和流动，为系统的正常运行和决策提供有

3 详细设计

3.1 处理流程

医院挂号系统的处理流程涵盖了从患者挂号到就诊完成的整个过程。以下是本次设计的医院挂号系统的处理流程图：

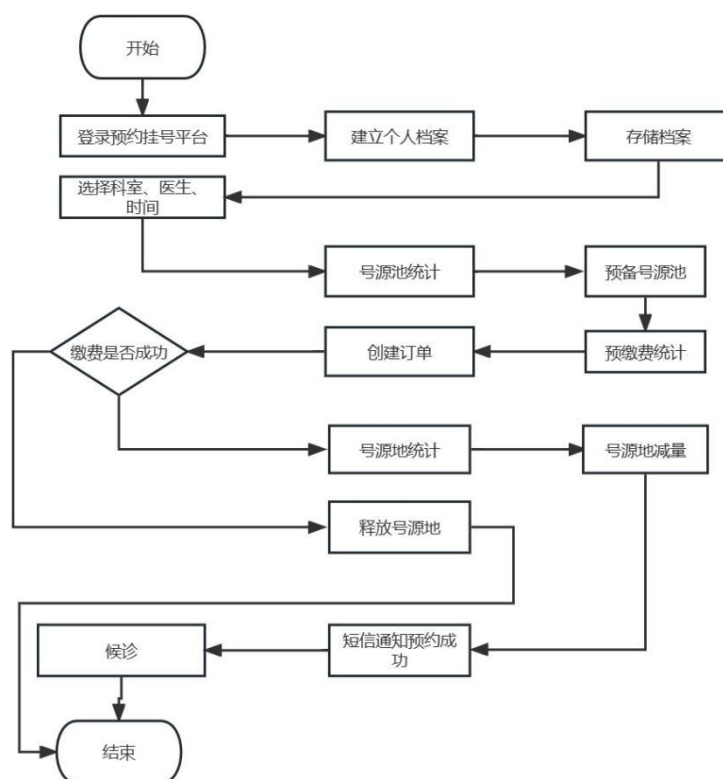


图 1 医院挂号系统流程图

1. 患者挂号：

- 患者通过医院挂号系统登录或注册账号。
- 患者选择就诊科室和医生，并选择预约时间段。
- 患者提交挂号请求，系统进行验证和处理。
- 系统生成挂号号码和预约确认信息，发送给患者。

2. 分诊：

- 医院工作人员根据患者的病情严重程度和就诊需求，将其分配到相应的医生或科室。

- 分诊结果更新至系统，并通知患者就诊的顺序和窗口号码。

3. 就诊：

- 患者按照分诊结果指示的顺序到达相应的诊室。
- 医生根据患者的挂号号码或个人信息，确认患者身份和挂号信息。
- 医生进行病情询问、体格检查和必要的辅助检查。
- 医生根据患者的病情进行诊断和制定治疗方案。
- 医生将诊断结果、治疗方案和药物处方等记录在患者的电子病历中。

4. 结算：

- 就诊完成后，患者前往结算窗口。
- 医院工作人员根据患者的就诊记录和费用标准，进行费用计算和结算。
- 患者支付相应的医疗费用，或通过医保支付。

5. 数据管理：

- 医院挂号系统实时记录和管理患者的挂号信息、就诊记录和电子病历。
- 医生可以随时查看和更新患者的病历记录，确保病历的完整性和准确性。
- 挂号统计和报表数据定期生成，反映挂号情况和资源利用情况。

6. 系统安全和维护：

- 系统具备安全机制，保护患者的隐私和数据安全。
- 系统进行定期备份和数据恢复，以防止数据丢失。
- 系统进行性能监测和故障排查，确保系统的稳定运行。

医院挂号系统的处理流程旨在提供高效的挂号和就诊服务，简化患者流程、减少排队时间，并提高医院资源的利用效率。通过规范的流程和

3.2 总体结构设计

医院挂号系统的总体结构设计是确保系统能够高效运行和实现各项功能的关键。本系统是一个典型的医院挂号系统的总体结构设计，它具有：

1. 用户界面层：

- 用户界面层是系统与患者、医生、工作人员等各类用户进行交互的接口。
- 用户界面提供友好的界面设计，包括登录、挂号、查询、预约等功能的界面展示

和操作。

- 用户界面可以根据不同用户角色进行权限控制，确保用户只能访问其具备权限的功能和数据。

2. 应用逻辑层：

- 应用逻辑层是医院挂号系统的核心处理层，负责处理用户请求、协调各个模块之间的交互。

- 应用逻辑层包括挂号管理、排班管理、就诊管理、数据查询等功能模块。

- 每个功能模块具有独立的业务逻辑，负责实现相应的业务规则和流程。

3. 数据访问层：

- 数据访问层负责与数据库进行交互，实现数据的读取、存储和更新。

- 数据访问层通过数据库操作，对患者信息、医生信息、排班信息、就诊记录等进行管理。

- 数据访问层采用合适的数据库访问技术，如SQL语句、ORM框架等，与数据库进行有效的数据交互。

4. 数据库层：

- 数据库层是存储和管理医院挂号系统所需数据的核心组件。

- 数据库层根据系统的数据需求，设计合适的数据库结构和表关系。

- 数据库层应具备良好的性能和可扩展性，通过索引、分区等技术进行优化，确保数据的快速访问和高效存储。

5. 安全性与权限控制：

- 系统的安全性和权限控制是保护患者隐私和数据安全的重要环节。

- 系统应具备用户认证和授权机制，确保只有经过授权的用户可以访问系统功能和数据。

- 敏感数据应采取加密传输和存储等安全措施，防止数据泄露和恶意攻击。

6. 第三方接口和集成：

- 医院挂号系统可能需要与其他系统进行集成，如医保系统、支付系统等。

- 系统需要与第三方接口进行数据交互，确保挂号信息、费用结算等的准确传递和

处理。

通过合理的总体结构设计，医院挂号

3.3 数据结构

目医院挂号系统的数据结构是指系统中存储和管理数据的组织形式和方式。良好的数据结构设计能够提高数据的存储效率和查询性能，同时确保数据的一致性和完整性。以下是医院挂号系统常见的数据结构：

1. 患者数据结构：

- 患者信息：包括患者ID、姓名、性别、年龄、联系方式等基本信息。
- 病历信息：包括既往病史、过敏史、家族病史等与就诊相关的信息。
- 挂号记录：包括挂号ID、挂号时间、科室、医生、就诊状态等挂号相关的信息。

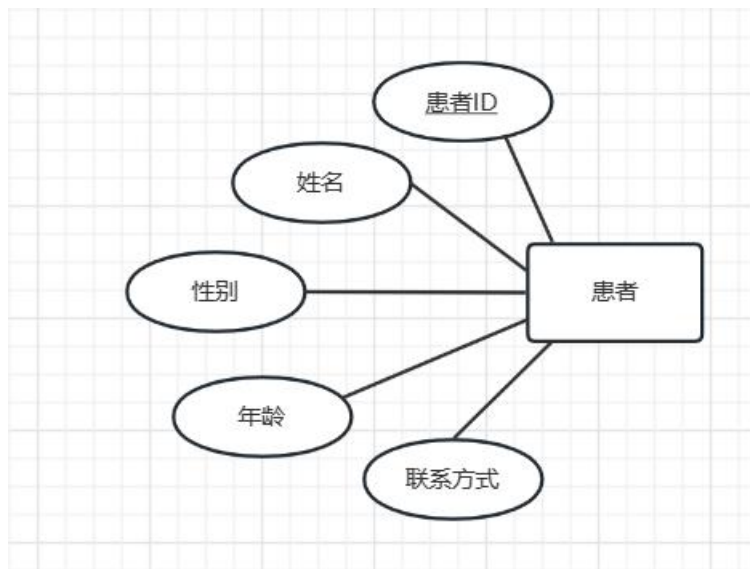


图 2 患者的数据结构

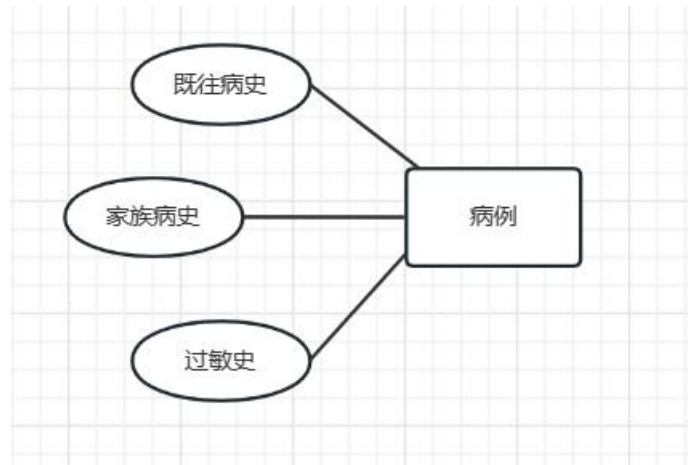


图3 病例数据结构

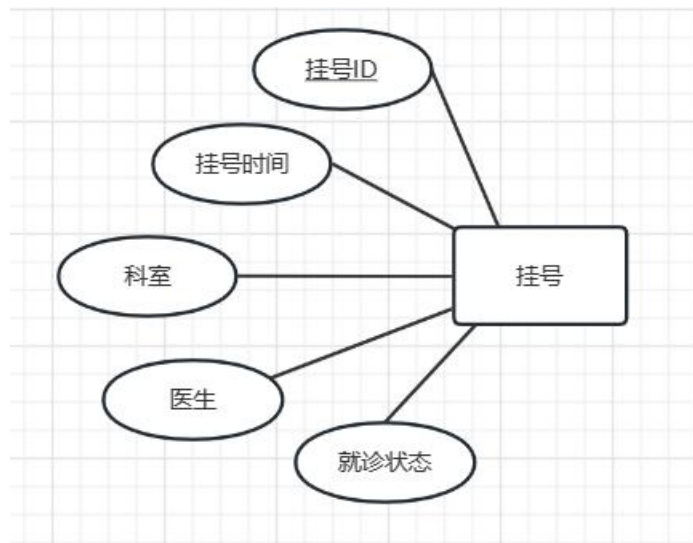


图4 挂号数据结构

2. 医生数据结构:

- 医生信息: 包括医生ID、姓名、性别、职称、专长等基本信息。
- 排班信息: 包括排班ID、科室、出诊时间、就诊类型等排班相关的信息。

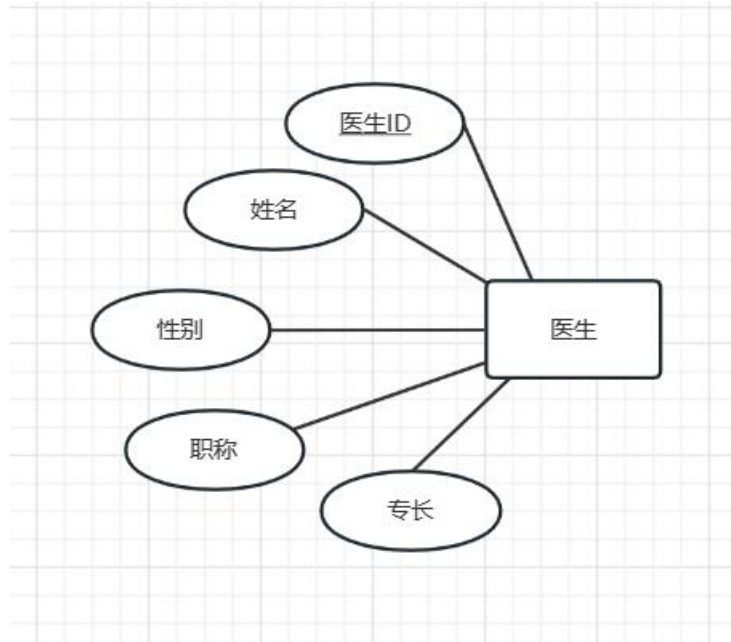


图 5 医生数据结构

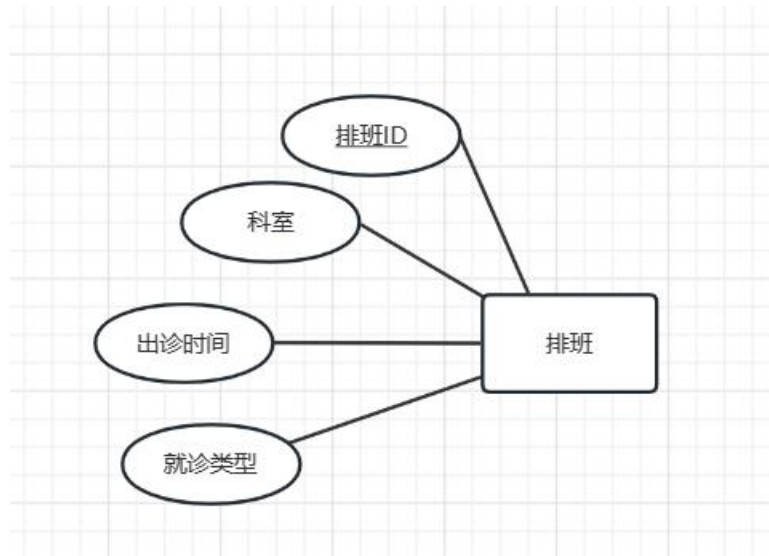


图 6 排班的数据结构

3. 科室和诊室数据结构:

- 科室信息: 包括科室ID、科室名称、科室负责人、科室电话等科室相关的信息。
- 诊室信息: 包括诊室ID、诊室编号、位置、设备等与诊室使用相关的信息。

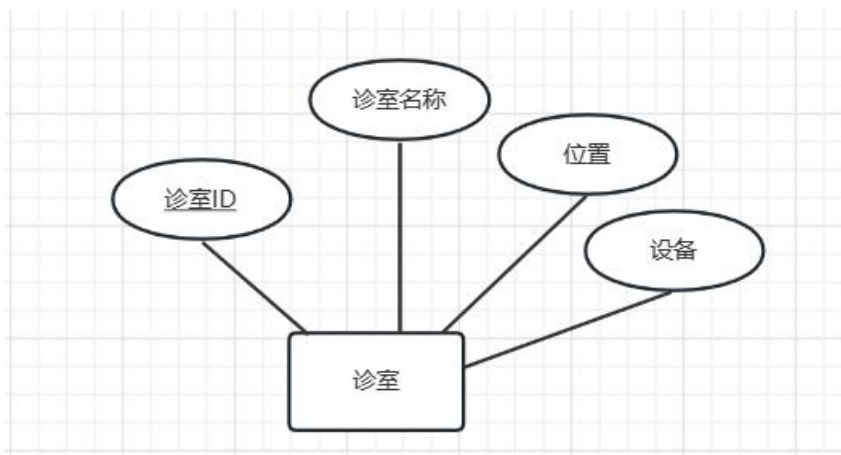


图 7 诊室数据结构

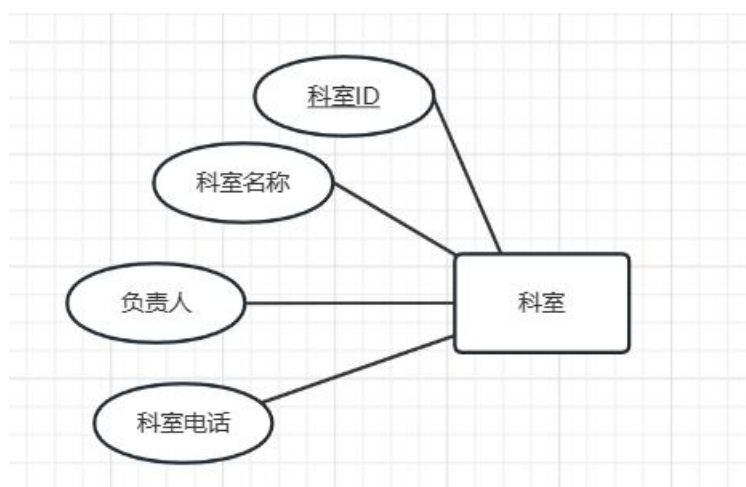


图 8 科室数据结构

4. 就诊记录和病历数据结构:

- 就诊记录: 包括就诊ID、就诊时间、患者ID、医生ID、诊断结果、治疗方案等与就诊相关的信息。

- 电子病历: 包括病历ID、患者ID、就诊ID、诊断记录、检查结果、药物处方等与患者健康管理相关的信息。

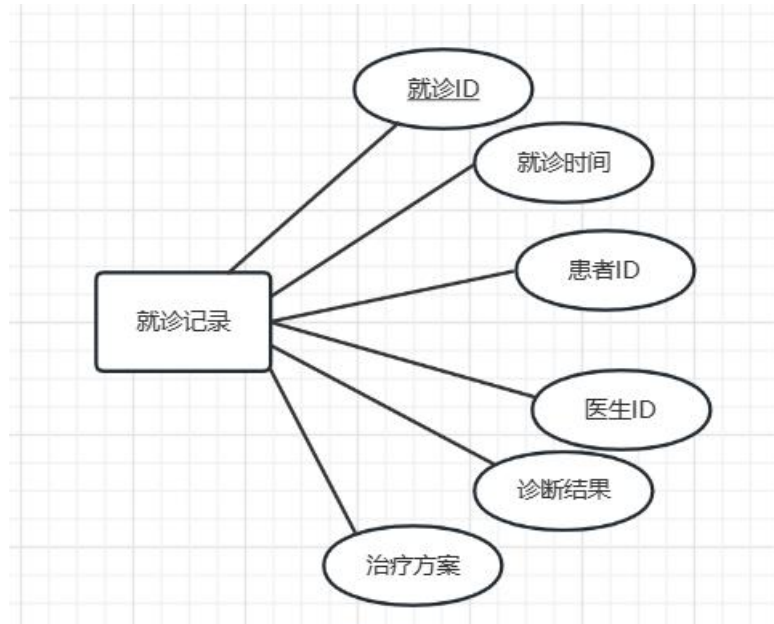


图 9 就诊记录的数据结构

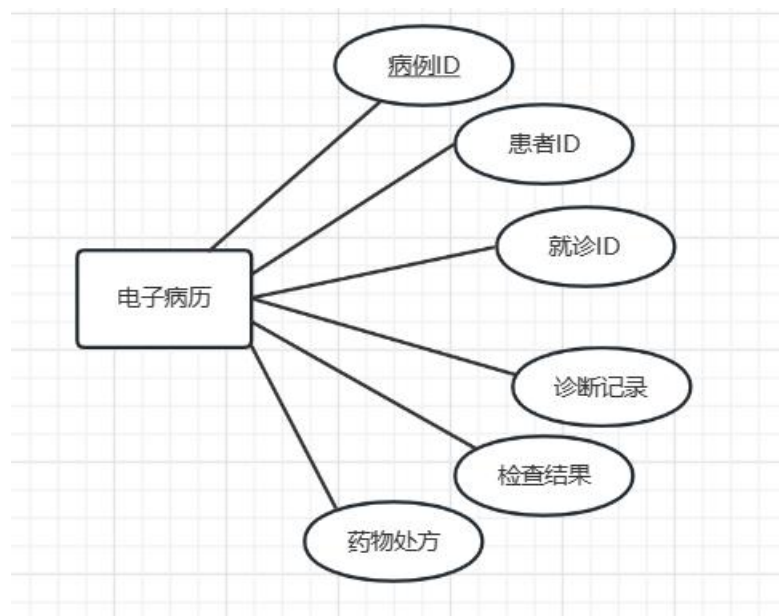


图 10 电子病历的数据结构

5. 挂号统计和报表数据结构：

- 挂号统计数据：包括统计日期、挂号量、就诊科室统计、医生工作量等与挂号统计相关的信息。

- 报表数据：包括报表ID、报表名称、生成日期、图表数据等用于决策和资源优化的数据。

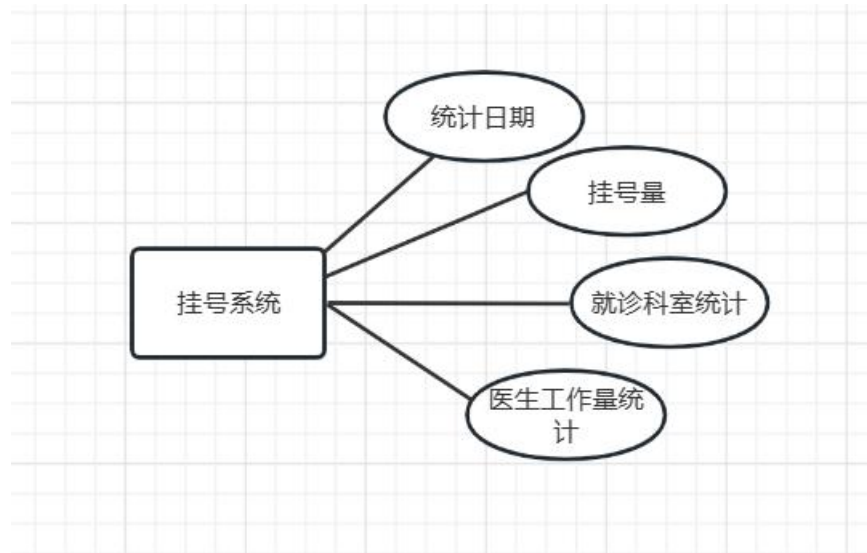


图 11 挂号统计数据结构

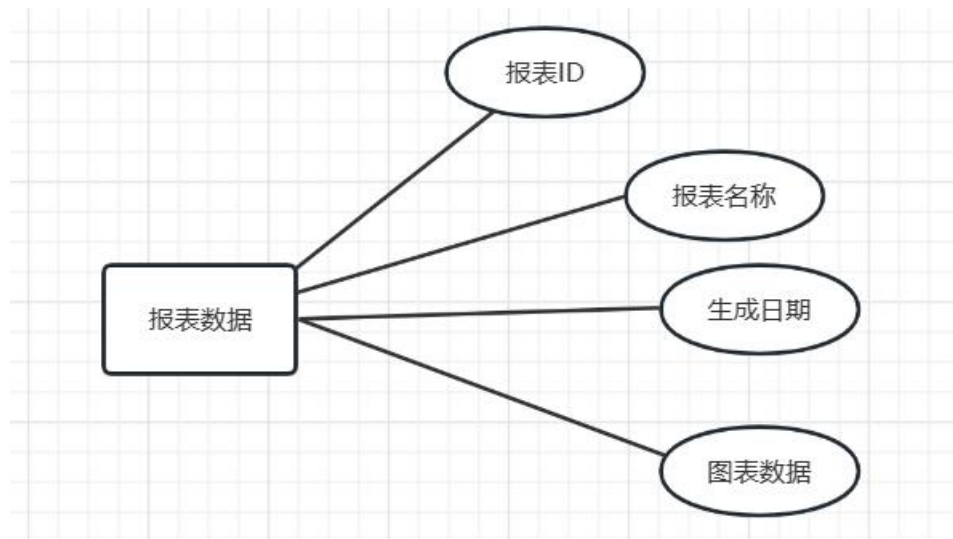


图 12 报表数据数据结构

6. 系统日志数据结构:

- 系统操作日志: 记录用户ID、操作时间、操作类型、操作内容等用户操作的日志信息。
- 错误日志: 记录错误ID、错误时间、错误类型、错误信息等系统发生的错误和异常情况。

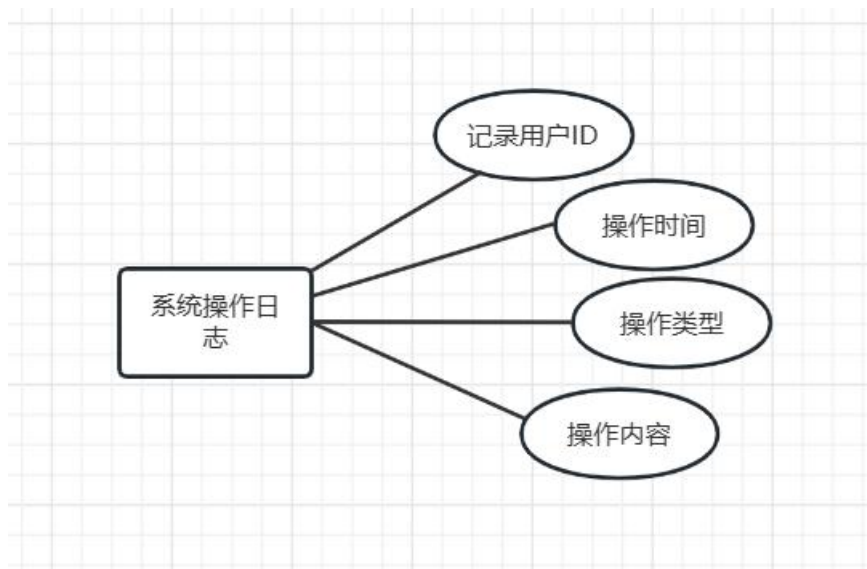


图 13 系统操作日志

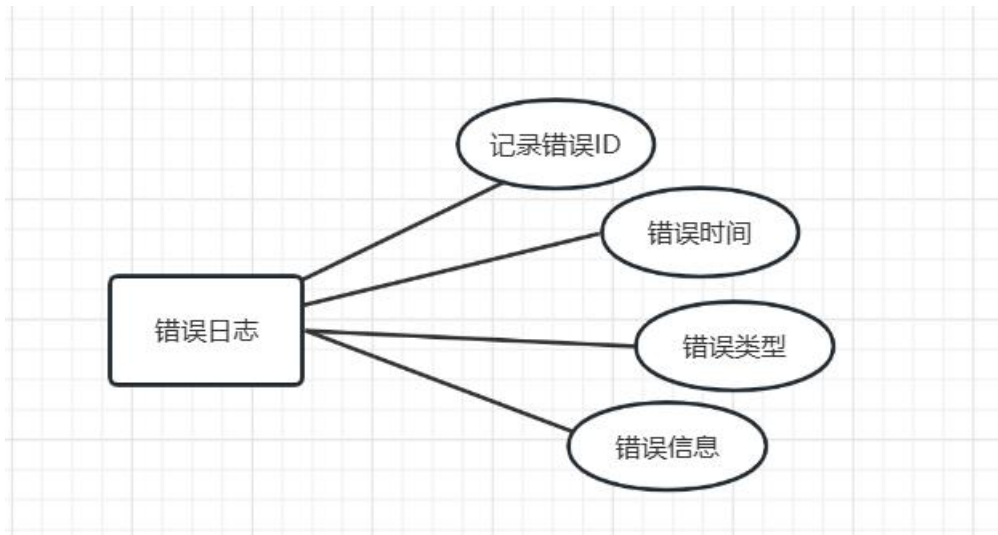


图 14 错误日志

在设计数据结构时，需要考虑数据之间的关联关系和依赖关系，使用适当的数据类型和约束条件来确保数据的有效性和完整性。此外，可以使用索引和关系数据库等技术来优化数据的查询和存储效率。

7. 订单和费用数据结构：

- 挂号订单：包括订单 ID、患者 ID、医生 ID、挂号时间、支付状态等挂号相关的信息。

- 就诊费用：包括费用 ID、患者 ID、就诊 ID、费用类型、费用金额等与就诊费用相关的信息。

- 支付记录：包括支付 ID、订单 ID、支付时间、支付方式、支付金额等与支付过程相关的信息。

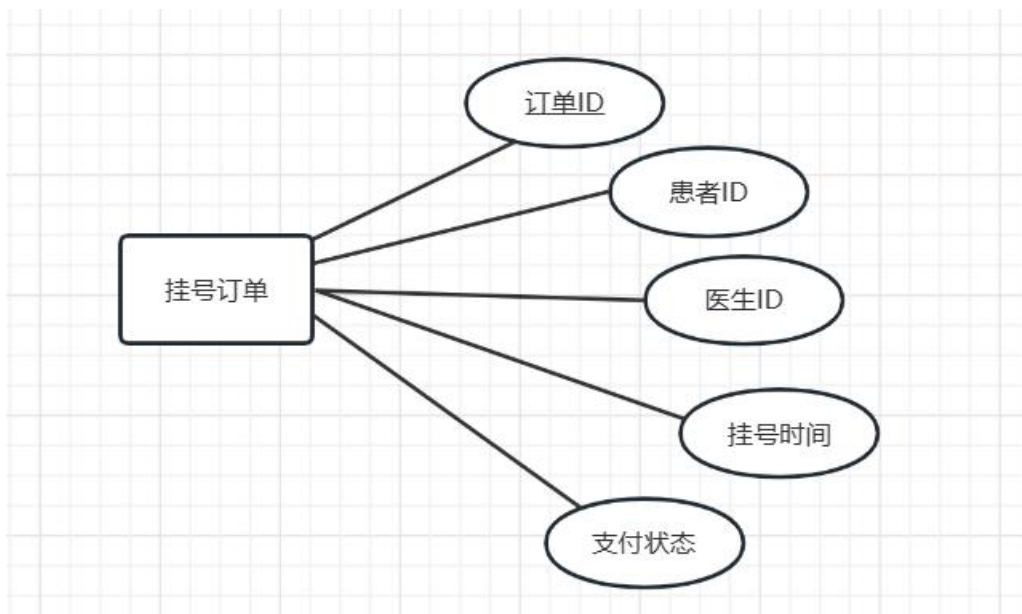


图 15 挂号订单数据结构

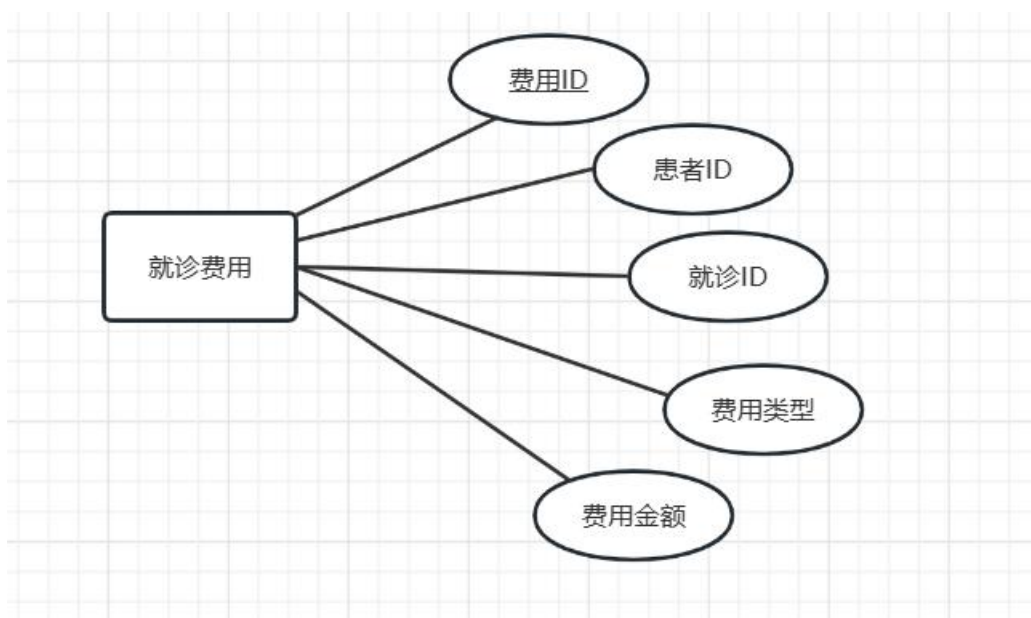


图 16 就诊费用数据结构图

8. 用户角色和权限数据结构:

- 用户角色: 包括角色 ID、角色名称、角色描述等用户角色相关的信息, 用于定义不同用户的权限和功能访问。
- 用户权限: 包括权限 ID、权限名称、权限描述等用户权限相关的信息, 用于控制用户对系统功能和数据的访问权限。

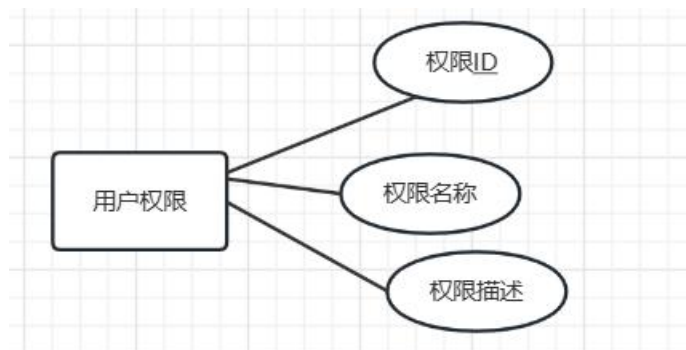


图 17 用户权限数据结构图

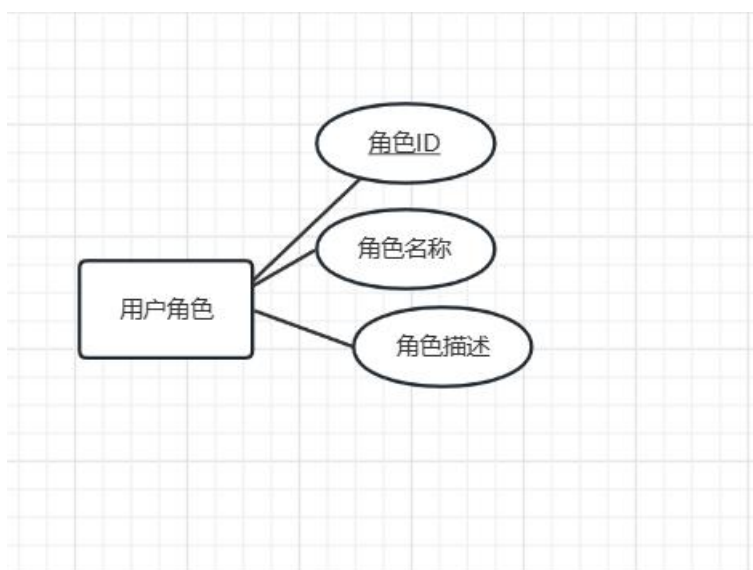


图 18 用户角色数据结构图

9. 第三方接口数据结构:

- 第三方系统接口: 包括接口 ID、接口名称、接口描述、接口参数等与第三方系统集成相关的信息。

- 数据交互记录：包括交互记录 ID、接口 ID、交互时间、交互类型、交互结果等与第三方系统数据交互相关的信息。

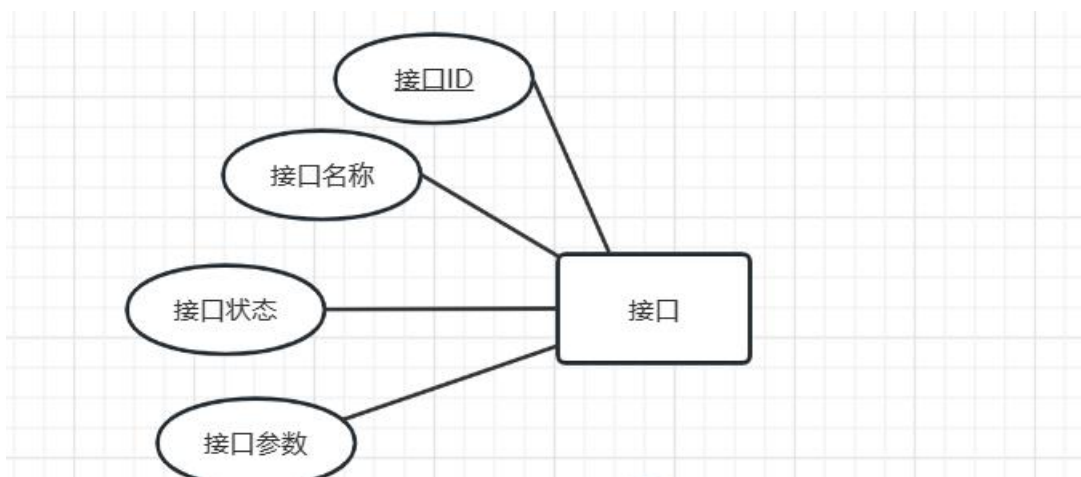


图 19 接口数据结构

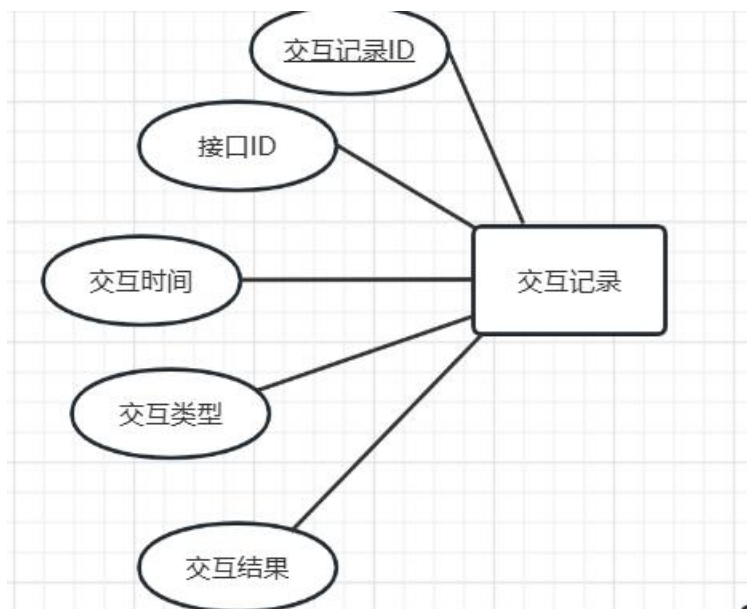


图 20 交互记录数据结构

10. 系统配置和参数数据结构：

- 系统配置项：包括配置项 ID、配置项名称、配置项值等与系统运行参数相关的信息。
- 系统参数：包括参数 ID、参数名称、参数值等与系统功能和行为相关的信息。

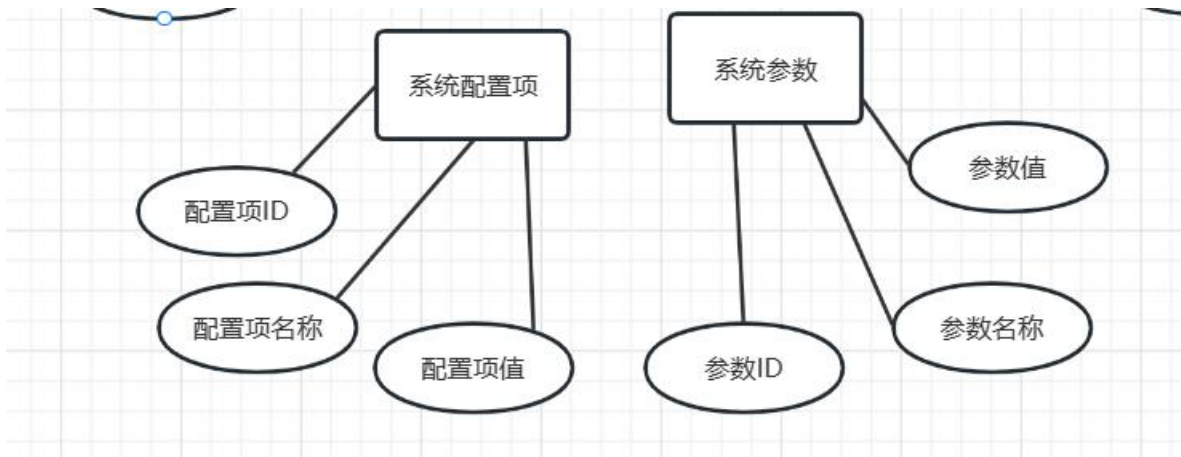


图 21 系统配置和参数配置数据结构

以上数据结构仅是医院挂号系统中的一部分，具体的数据结构设计还需根据实际需求进行进一步规划和定义。在设计数据结构时，需要考虑数据的一致性、完整性和可扩展性，确保系统能够高效地存储和管理数据，并满足系统功能的要求。

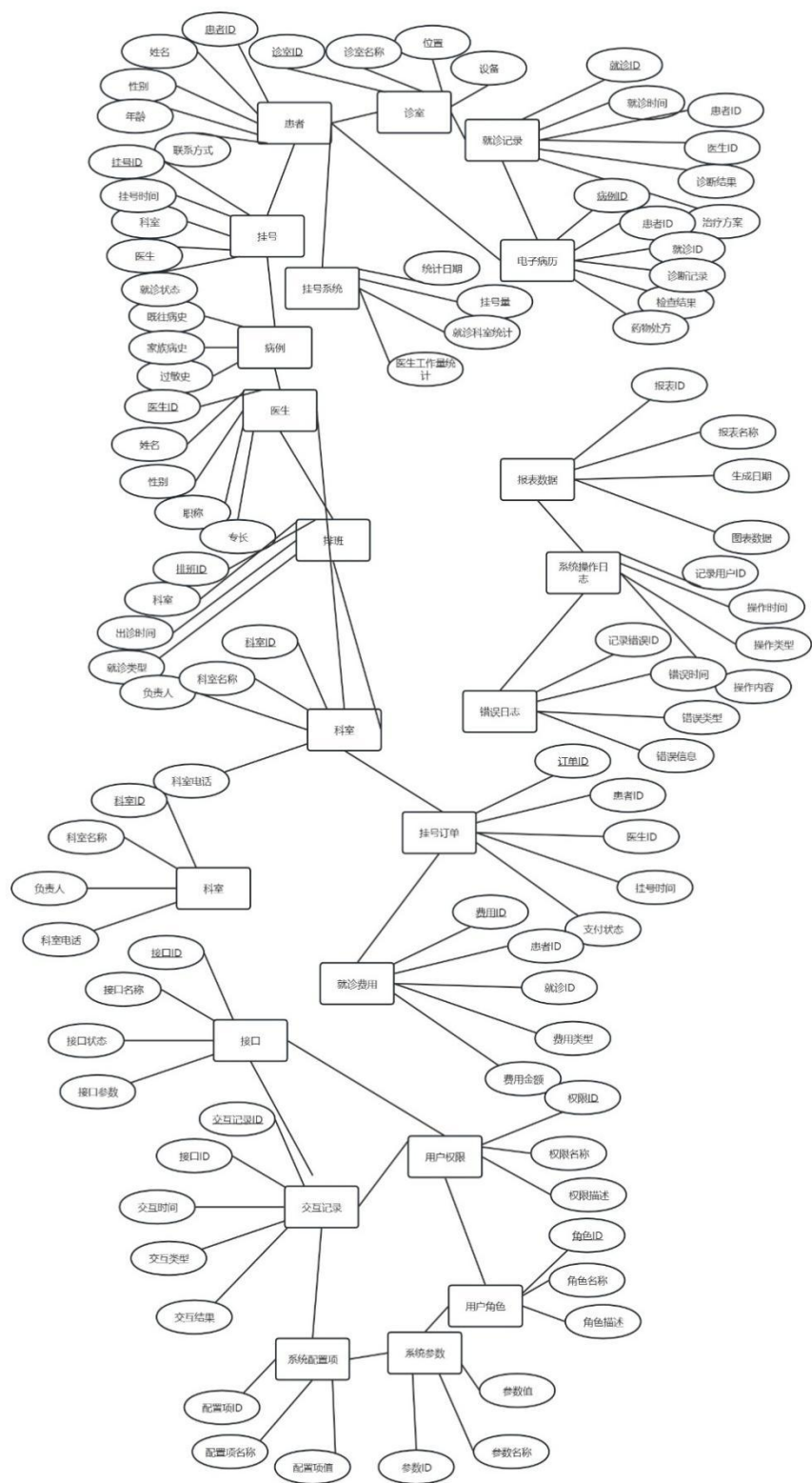


图 22 本系统的数据结构

4 系统实现

4.1 实现工具

目在开发医院挂号系统时，可以使用多种工具和技术来实现系统的各个方面。以下是可能用到的一些主要工具和技术：

1. 编程语言：选择合适的编程语言来实现系统的后端和前端逻辑。常见的编程语言包括Java、Python、C#等，本项目选择Java语言为开发语言。

2. 后端框架：使用后端框架来加速系统的开发过程，并提供一些常用的功能和工具。常见的后端框架包括Spring Boot、Django、ASP.NET等，本系统使用Java语言实现，因此后端框架用到的为Spring Boot。

3. 前端技术：使用HTML、CSS和JavaScript等前端技术来实现系统的用户界面。同时，可以使用流行的前端框架如React、Angular、Vue.js等来简化前端开发和提升用户体验。

4. 数据库管理系统：选择合适的数据库管理系统（DBMS）来存储和管理系统的数据。常见的DBMS包括MySQL、PostgreSQL、Oracle等，由于本系统需要高度可靠的数据库进行存储，因此本系统采用甲骨文公司的Oracle数据库。

5. 数据库工具：使用数据库工具来设计数据库结构、进行数据建模和管理数据库。本实验选取的数据库工具为JetBrain公司的DataGrip

6. 版本控制系统：使用版本控制系统来管理源代码的版本和协同开发。常用的版本控制系统包括Git，可以利用Git来管理代码仓库、分支和合并等。

7. 统一建模语言（UML）工具：使用UML工具来进行系统的建模和设计。UML工具可以帮助开发团队绘制用例图、类图、时序图等，以便更好地理解 and 沟通系统的设计。

8. 测试工具：使用自动化测试工具来编写和执行系统的单元测试、集成测试和端到端测试。常见的测试工具包括JUnit、Selenium、Postman等，可以提高系统的质量和稳定性。

9. 集成开发环境（IDE）：选择适合开发语言的IDE来进行开发工作。常用的IDE包括Eclipse、IntelliJ IDEA、Visual Studio等，提供代码编辑、调试、自动完成等功能，本系统选取的集成开发环境为IntelliJ IDEA。

10. 项目管理工具：使用项目管理工具来管理开发过程和任务分配。常见的项目管理工具包括JIRA、Trello、Microsoft Project等，可以帮助团队协调工作、追踪进度和管理需

求。

4.2 数据库

数据库在医院挂号系统中扮演着重要的角色，需要综合考虑系统的需求、性能、安全性和可扩展性等因素，选择合适的数据库类型，并进行合理的数据库结构设计、性能优化和备份恢复策略，以确保系统的数据管理和运行的可靠性和高效性。

在医院挂号系统的开发中，数据库是非常重要的组成部分，用于存储和管理系统中的各种数据。以下是在本系统中关于数据库的考虑和决策。

4.2.1 数据库类型选择：

在选择数据库类型时，需要考虑系统的规模、性能需求和团队的熟悉程度。常见的数据库类型包括关系型数据库（如MySQL、PostgreSQL、Oracle）和非关系型数据库（如MongoDB、Redis）。关系型数据库适合处理结构化数据和复杂查询，而非关系型数据库适合处理半结构化数据和高并发访问。

2. 数据库结构设计：

数据库结构设计应考虑系统的实际需求和数据关系。使用数据库建模工具进行实体关系建模（ER建模），绘制实体-关系图（ER图）来表示表之间的关系。合理的数据库结构设计能够提高数据的一致性、完整性和查询效率。

3. 数据表设计：

在设计数据表时，需要考虑每个表的字段和数据类型，确定主键、外键和索引等。字段应该明确、准确地表示数据的含义，并设置适当的约束条件，如唯一约束、非空约束等，以确保数据的有效性和完整性。

4. 数据库性能优化：

为了提高数据库的性能，可以采取一些优化措施。例如，创建适当的索引来加速查询操作，合理分配表空间和文件组，定期执行数据库维护任务（如优化查询计划、清理过期数据），并使用缓存技术来减轻数据库的负载。

5. 数据库安全性：

数据库安全性是非常重要的，需要采取措施来保护数据的机密性和完整性。确保数据库的访问受到身份验证和授权的限制，限制敏感数据的访问权限，使用加密技术对敏感数

据进行保护，并定期备份和恢复数据库以防止数据丢失。

6. 数据库备份和恢复：

建立数据库备份和恢复策略，定期备份数据库以防止数据丢失，同时测试和验证备份数据的可恢复性。备份可以采用完全备份、增量备份或差异备份等方式，根据实际需求来选择。

7. 数据库迁移和升级：

在系统的演进过程中，可能需要进行数据库迁移和升级。确保在数据库迁移和升级过程中，数据的完整性和一致性得到保证，进行充分的测试和验证，以避免数据丢失和系统故障。

8. 数据库监控和性能调优：

建立数据库监控系统，定期监测数据库的性能指标，如响应时间、查询执行时间、资源利用率等。通过监控可以及时发现潜在的性能问题，并进行性能调优。性能调优包括优化查询语句、调整数据库参数、增加硬件资源等，以提高数据库的响应速度和吞吐量。

9. 数据库备份和灾难恢复：

设计和实施数据库的备份和灾难恢复策略，确保数据的可靠性和可恢复性。定期进行全量备份和增量备份，并将备份数据存储在安全的位置。同时，进行灾难恢复演练，验证备份数据的可用性，并准备应对可能发生的灾难情况。

10. 数据库的扩展性和高可用性：

考虑到系统的扩展性和高可用性需求，可以采用数据库集群、主从复制、读写分离等技术来实现数据库的水平扩展和高可用性。这样可以提高系统的并发处理能力和故障恢复能力。

数据库在医院挂号系统中扮演着重要的角色，需要综合考虑系统的需求、性能、安全性和可扩展性等因素，选择合适的数据库类型，并进行合理的数据库结构设计、性能优化和备份恢复策略，以确保系统的数据管理和运行的可靠性和高效性。

4.3 开发平台

目在医院挂号系统的开发中，选择适合的开发平台是至关重要的。以下是一些常见的开发平台选项：

1. Web开发平台：使用Web开发平台可以构建跨平台、可访问的应用程序。常见的Web开发平台包括Java的Spring Boot、Python的Django、Node.js的Express等。这些平台提供了开发框架和工具，简化了开发过程并提供了丰富的功能和组件。

2. 移动应用开发平台：如果需要开发移动端的医院挂号应用程序，可以选择移动应用开发平台。例如，使用React Native、Flutter或Ionic等跨平台开发框架，可以在多个移动操作系统上构建一致的用户体验。

3. 桌面应用开发平台：如果需要开发桌面应用程序，可以选择合适的桌面应用开发平台。例如，使用Electron或Java的Swing框架可以构建功能强大的跨平台桌面应用程序。

4. 云平台：利用云平台可以实现快速开发、部署和扩展。例如，使用AWS、Microsoft Azure或Google Cloud等云平台提供的服务和工具，可以轻松部署和管理医院挂号系统。

5. 前端开发框架：选择合适的前端开发框架可以加速用户界面的开发。常见的前端开发框架包括React、Angular、Vue.js等，它们提供了丰富的UI组件和开发工具。

选择适合的开发平台可以提高开发效率、降低开发成本，并提供丰富的功能和工具支持。根据具体的需求和团队的技术背景，选择合适的开发平台对于医院挂号系统的成功开发和部署至关重要。

5 测试报告书

5.1 测试方法

在开发医院挂号系统时，测试是确保系统质量和功能完整性的重要环节。下面是一些常用的测试方法和策略：

单元测试：单元测试是针对系统中最小的可测试单元（如函数、方法）进行的测试。通过编写针对每个单元的测试用例，验证其正确性和预期行为。使用单元测试框架（如JUnit、PyTest）来自动执行测试用例，并检查输出是否符合预期。

集成测试：集成测试用于测试系统各个模块之间的协作和交互。通过编写集成测试用例，模拟真实场景下的模块集成，并验证它们之间的接口和数据流是否正确。

系统测试：系统测试是对整个医院挂号系统进行的综合性测试，验证系统在实际使用环境中的功能和性能。系统测试可以包括功能测试、性能测试、安全性测试等。功能测试验证系统的功能是否符合需求，性能测试评估系统在负载和并发条件下的性能，安全性测试检查系统的安全漏洞和弱点。

用户验收测试：用户验收测试是由最终用户或代表用户的人员进行的测试，目的是验证系统是否满足用户需求和预期。通过模拟实际使用场景，用户可以评估系统的易用性、功能完整性和符合性。

冒烟测试：冒烟测试是在系统开发的早期阶段进行的一组基本测试，用于快速检查系统的主要功能是否正常工作。冒烟测试有助于尽早发现系统的严重问题，确保系统能够进行进一步的详细测试和开发。

回归测试：回归测试用于确保系统在进行修改或更新后仍然保持原有功能的稳定性。在每次系统更新后，执行回归测试以验证已有功能是否正常运行，并检查是否引入新的错误或问题。

性能测试：性能测试旨在评估系统在不同负载条件下的性能表现。通过模拟大量用户并发访问系统，测试其响应时间、吞吐量和资源利用情况。常见的性能测试工具包括JMeter、LoadRunner等。

安全性测试：安全性测试用于评估系统的安全性和防护能力。通过模拟攻击和渗透测试，发现系统的安全漏洞和脆弱点，并采取相应措施进行修复和防范。

自动化测试：自动化测试是使用脚本和自动化测试工具来执行测试用例的过程。自动化测试可以提高测试效率和一致性，并减少人工测试的工作量。通过编写测试脚本，可以自动执行重复性的测试任务，并生成详细的测试报告。

数据驱动测试：数据驱动测试是一种测试方法，其中测试用例的输入和预期输出是基于一组测试数据进行定义。通过使用不同的测试数据集合，可以覆盖更多的测试场景和边界条件，提高测试覆盖率。

真实环境测试：在系统测试阶段，需要在真实的生产环境中进行测试，以模拟实际的用户行为和系统负载。这可以帮助发现系统与外部环境的交互问题，并评估系统的可靠性和稳定性。

异常情况测试：异常情况测试用于验证系统在处理异常或边界条件时的行为。通过模拟各种异常情况，如输入错误、无效数据、网络故障等，可以确保系统能够正确地处理和响应这些情况，避免系统崩溃或数据损坏。

完整性测试：完整性测试旨在验证系统的数据完整性和一致性。通过对数据输入、处理和输出进行全面的测试，确保数据在系统中的存储、更新和检索过程中不会出现错误或丢失。

用户界面测试：用户界面测试用于验证系统的用户界面是否符合设计和用户体验要求。测试包括界面布局、导航、交互等方面，以确保用户能够轻松使用系统并完成任务。

并发测试：并发测试用于测试系统在高并发负载下的性能和稳定性。通过模拟多个同时访问系统的用户，检查系统在并发条件下的响应时间、资源利用率和数据一致性。

综合使用以上不同的测试方法和策略，可以全面评估医院挂号系统的质量、功能和性能。测试过程应该从早期阶段开始，并持续进行，以确保系统在开发周期内及时发现和解决问题，以交付高质量的挂号系统。

5.2 测试示例

以下是医院挂号系统的一些测试示例，用于说明测试用例的类型和内容：

5.2.1 单元测试示例

测试目标：验证用户注册功能的正确性。

测试用例：

- 输入有效的用户名和密码，验证用户能够成功注册。
- 输入重复的用户名和密码，验证系统是否能够提示错误信息。
- 输入无效的用户名和密码，验证系统是否能够拒绝注册并给出相应的错误提示。

5.2.2. 集成测试示例

测试目标：验证挂号系统的预约和支付功能的正确性。

测试用例：

- 模拟用户选择预约时间和科室，验证系统能够正确预约。
- 模拟用户选择支付方式并完成支付，验证系统能够正确处理支付流程。
- 检查预约和支付系统之间的接口和数据流是否正常。

5.2.3. 系统测试示例

测试目标：验证整个医院挂号系统的功能和性能。

测试用例：

- 测试不同用户角色的登录和权限管理功能。
- 模拟多个用户同时访问系统，测试系统在高并发负载下的性能和稳定性。
- 测试系统在各种异常情况下的处理能力，如网络故障、无效输入等。

5.2.4. 用户验收测试示例

测试目标：验证系统是否符合用户需求和期望。

测试用例：

- 用户注册并登录系统，验证系统界面的易用性和导航流程。
- 模拟用户进行挂号、预约、取消预约等操作，验证系统是否按照预期执行。
- 用户测试支付功能，验证支付过程的安全性和可靠性。

5.2.5. 性能测试示例：

测试目标：评估医院挂号系统在负载和并发条件下的性能。

测试用例：

- 模拟多个用户同时进行挂号和预约操作，测试系统的响应时间和吞吐量。
- 增加系统负载，测试系统在高负载条件下的性能表现和资源利用率。

- 测试系统在长时间运行和大量数据处理时的稳定性和内存管理。

这些测试示例涵盖了不同层次和类型的测试，包括单元测试、集成测试、系统测试、用户验收测试和性能测试。通过执行这些测试用例，可以全面评估医院挂号系统的功能、性能、安全性和用户体验，确保系统的质量和可靠性。

参考文献

- [1] 郭爽.基于 SSH 框架的全民健康预约挂号系统的设计与实现[J].中国高新技术企业, [2] 2011-11-01.
- [3] 邱金水等. 基于 SSH 的 Web 医院信息系统的设计与实现[J] , 昆明理工大学, 2013-04-01.
- [4] 胡晟娅. 医院预约挂号系统设计与应用[J]. 通讯世界, 2016-03-12 , (02) : 页码.
- [5] 张莹 李韶斌.医院预约挂号系统的设计与实现[D]. 中国卫生信息管理杂志, 2011-04-20
- [6] 陈功贵. 医院门诊挂号管理信息系统的设计与实现[J]. 宿州教育学院学报, 2005-10-15.
- [7] 王希望. 基于 SSH 架构的 Web 系统的开发方法[D].河北农业大学学报, 2010-12-15.
- [8] 马郑萍.对医院预约挂号系统实施过程中存在问题的分析.西部中医药, 2013.
- [9] 罗本骞.数据库软件应用.《网络安全与技术应用》,2014.
- [10] 刘波.浅谈医院预约挂号系统的设计.《科技创新与应用》.2012.
- [11] LIAO TIEJUN. Catering industry material purchasing mobile terminal system.CHONGQING YUNBAKE COMMERCE AND TRADE CO LTD,2016-08-03.
- [12] LU XIAOJUN. Data processing method for high-concurrency platform server[J].UNIV ELECTRONIC SCIENCE & TECH CHINA,2016-08-31.
- [13] CHEN PENGFEI. General paging query method and system based on SSH framework[J].SHANGHAI SUNINFO TECHNOLOGY CO LTD. 2014-08-13