**《软件工程》**

**实验报告五 ：面向对象的系统建模C**

**姓 名： 周子雄 学 号： 202210120127**

**院 系： 计算机与信息学院 专 业： 计算机科学与技术**

**实 验 室： j1307 实验日期： 2024/11/16**

**总评成绩： 审阅教师：**

# 实验目的

1. 掌握用例图建模；
2. 编写用例规格说明

# 实验环境

boardmix，idea

# 实验要求

1. 阅读下面材料，Rational Rose绘制用例图，并编写用例规格说明（选择2-3个关键用例编写即可，预约和支付用例必须编写用例规格说明，活动图建模），以一个Word文件的形式提交（用例图贴到文档的适当位置）
2. 参考案例 word文档《案例-旅店预约系统用例图和用例文档.doc》或PDF格式《案例-旅店预约系统用例图和用例文档.pdf》格式编写
3. 文件以“学号-姓名-软件工程实验五.doc”的方式命名，提交到长江雨课堂“软件工程实验五”

# 实验内容

“医院预约挂号系统”问题陈述

为了规范和推动医院预约挂号服务，卫生部2019年8月在其官方网站发布了《关于在公立医院施行预约诊疗服务工作的意见(征求意见稿)》，要求在推动医院开展预约挂号工作的同时，提高对预约挂号服务工作的认识、加强对预约挂号服务工作的管理、并认真做好相关组织工作。

某IT公司瞄准此次契机，决定着手开发一个通用的“医院预约挂号系统”，以满足各级公立医院的预约挂号需求。

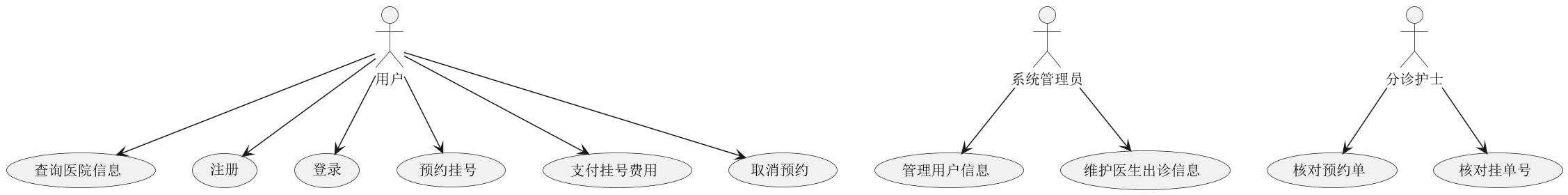
系统的基本流程如下：未注册用户可以通过该系统查询医院、相关科室、各科室的医生等各类信息，但不能使用其它与预约相关的业务。需要进行预约挂号的用户必须通过该网站利用身份证号进行实名注册，并提供手机号，注册信息由系统管理员进行审核，审核通过后，用户才可使用手机号和验证码登录该系统。为了方便用户登录，系统还支持第三方（如微信公众号）登录。微信公众号登录必须首先绑定手机号用户才能登录。预约挂号时，用户首先选择需要预约的医院，之后选择要预约的科室和时间（指定某个日期的上午或下午）；此时，系统应自动显示该时间段内该科室所有出诊的医生。需要注意的是，每个医生每次出诊所能看病的人数有一定的限制，当某个医生的预约人数满员后即不可预约。用户可以选择一个可预约的医生进行预约，一个用户每个时间段最多只能预约5位医生。预约成功后，用户可以打印预约单。用户还可以通过第三方的支付系统（I期只支持淘宝的支付宝，后续支持各类信用卡）网上支付挂号费，也可以暂不交费。年龄超过60岁的患者免费。已交费的用户还可打印挂号单，并在看病当天拿着预约单和挂号单直接去医院相应的科室分诊台进行分诊，分诊台的护士核查预约单和挂号单无误后盖章确认，即允许用户看病。未交费的用户需要拿着预约单到医院的挂号处交费，挂号处核查预约单，并打印出挂号单，盖章确认后交给分诊台护士后进行分诊。

在看病的前一天，用户可随时取消预约记录，系统不收取任何费用，已缴的费用会自动退回到用户的帐号。看病当天的预约记录只能在医院挂号处现场取消，也不收取费用。但是，对于那些在网上预约成功，却不去看病也不按时取消的用户，系统会进行警告：已收取的费用不再退回，每出现一次用户的信用等级下降1级；当用户信用等级将为0时，不再允许使用该系统。用户的初始信用等级是在审核用户注册信息时设定的。

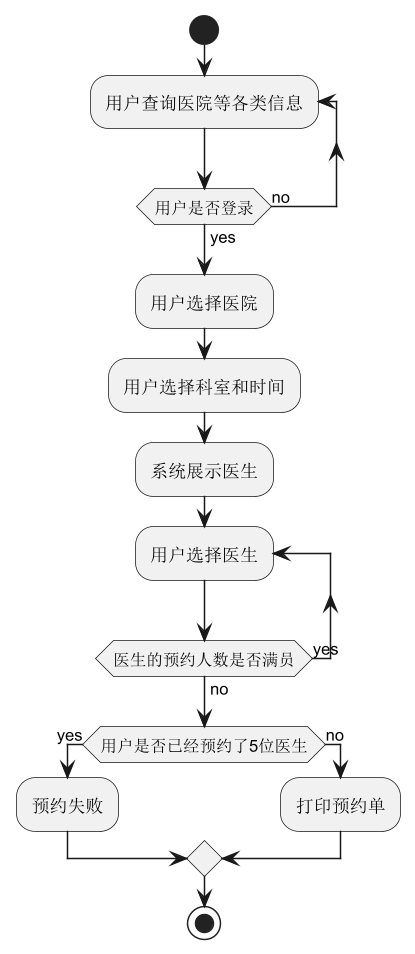
此外，有关医生的出诊信息可以由系统管理员手动维护，也可通过定制一些规则后由系统提前若干天（具体多少天可以由系统管理员设置）生成某日的出诊信息。当用户预约成功、缴费成功、退费成功、预约时间到达前2小时提醒、预约未使用未取消时系统均要求使用短信通知用户。

支持登录用户代患者挂号功能，以解决对于无法使用APP的用户如婴幼儿或老年人也能够使用该预约挂号系统。

系统用例图



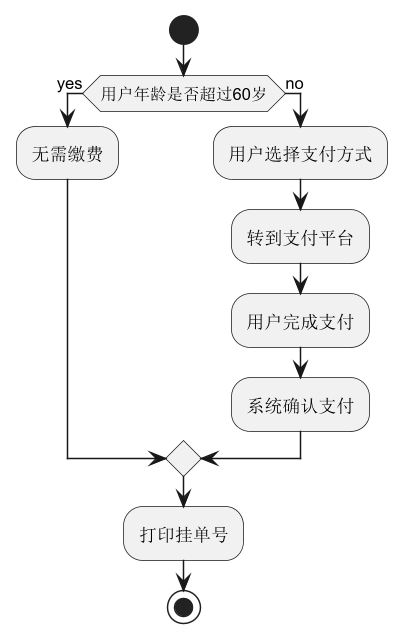
预约活动图



**表1 “预约”用例文档**

|  |  |
| --- | --- |
| **用例名** | 预约 |
| **简要描述** | 用户预约医生 |
| **参与者** | 用户 |
| **涉众** | 用户：预约医生  医生：给用户看病 |
| **相关用例** | 暂无 |
| **前置条件** | 用户正确地登录到该系统 |
| **后置条件** | 如果预约成功，则系统记录本次预约信息，相关的医生被预约次数更新 |
| **基本事件流**  1. 该用例起始于用户查询医院等信息；  2. 系统按照用户的要求设定查询条件（D-1）来查询医院（D-2）；  3. 系统显示所有科室列表（A-1）；  4. 用户选择科室和时间（A-2）；  5. 系统展示医生（B-1）；  6. 用户选择医生（A-3）（B-2）；  7. 系统保存本次预约信息（D-3），显示预约成功消息（A-4）；  8. 系统打印预约单后，用例结束。 | |
| **备选事件流**  A-\* 系统在提交保存预约信息前，随时都可能中止该用例  1. 系统提醒系统当前所有操作都会被取消；  2. 系统确认后，当前用例结束，也可选择取消，则继续后续操作。  A-1 没有找到满足可预约需求的医院  1. 系统显示没有找到所需的医院；  2. 系统可以重新设定查询条件，也可以选择结束该用例。  A-2 所有医院中没有用户满意的科室和时间  系统可以重新查询其他科室，也可选择结束该用例；  A-3 用户预约医生人数上限  系统可以重新为用户选择医生，或修改预约时间，也可以选择结束该用例  A-4 系统保存失败  1. 系统显示保存失败，并提示系统重新提交；  2. 系统可以重新提交本次预约信息，也可以结束该用例。 | |
| **补充约束-数据需求**  D-1 查询条件包括：医院名称，科室名称，医生名称等。  D-2 医院信息包括：医院名称  D-3 预约信息包括：用户的基本信息（姓名、身份证）和本次预约信息（医院名称、科室、医生、时间）  **补充约束-业务规则**  B-1 每一个医生每天都有预约上限  B-2 一个用户每个时间段最多只能预约5位医生  **补充约束-非功能性需求**  无 | |
| **待解决问题**  （暂无） | |
| **相关图**  （暂无） | |

支付活动图



**表1 “支付”用例文档**

|  |  |
| --- | --- |
| **用例名** | 支付 |
| **简要描述** | 用户支付单号 |
| **参与者** | 用户 |
| **涉众** | 用户：支付单号 |
| **相关用例** | 暂无 |
| **前置条件** | 用户正确地登录到该系统 |
| **后置条件** | 用户成功地预约医生 |
| **基本事件流**  1. 该用例起始于用户预约；  2. 用户选择支付方式（B-1）；  3. 转到支付平台  4. 用户完成支付（A-1）  5. 系统确认支付  6. 系统打印挂单号后，用例结束。（D-1） | |
| **备选事件流**  A-\* 系统在提交保存预约信息前，随时都可能中止该用例  1. 系统提醒系统当前所有操作都会被取消；  2. 系统确认后，当前用例结束，也可选择取消，则继续后续操作。  A-1 支付失败  1. 支付平台支付失败  2. 用户可以重新支付，也可以选择结束该用例。 | |
| **补充约束-数据需求**  D-1 挂单号新信息：（用户姓名，支付金额等）  **补充约束-业务规则**  B-1 用户年龄如果60岁以上，则不用支付  **补充约束-非功能性需求**  无 | |
| **待解决问题**  （暂无） | |
| **相关图**  （暂无） | |

# 实验结论

在进行“医院预约挂号系统”实验的过程中，我不仅加深了对用例图和用例规格说明的理解，还学习了如何将业务需求转化为具体的系统设计，尤其是在面向对象分析与设计（OOAD）方面的应用。以下是我对本次实验的一些心得体会：

在系统设计中，首先需要对业务需求有清晰的理解。在实验开始前，仔细阅读问题陈述，理清每个功能的实际意义和业务流程，这是我们能够准确设计出系统的前提。例如，用户预约挂号的流程涉及多个环节，包括医院选择、科室选择、医生选择、支付、挂号等，每个环节都需要仔细梳理其逻辑关系，确保系统能够顺利执行。

用例图是整个系统功能的高层次视图，可以帮助我们快速了解系统的主要参与者及其互动。通过绘制用例图，我能够清晰地看到各个用户角色与系统之间的关系以及他们能够进行的操作。这对于后续的系统设计和开发具有重要的指导意义。在本次实验中，我通过用例图展示了用户、系统管理员、第三方支付平台等角色的互动，帮助我更好地理解了系统的功能需求。

用例规格说明是对每个功能的详细描述，清晰列出了用例的触发条件、主要流程、后置条件等关键信息。通过编写用例规格说明，我深入思考了系统每个功能的实现细节。例如，用户预约挂号的用例规格说明不仅涉及预约的基本流程，还考虑到了医生预约人数限制、用户预约医生的限制、预约成功后的后续操作等，这些都需要在设计中予以充分考虑和处理。

活动图提供了一种可视化的方式来展示系统内部的工作流程。在本次实验中，我通过活动图展示了用户预约挂号和支付挂号费的具体流程。活动图让我更清晰地理解了每个操作步骤之间的顺序关系，以及不同决策点对系统流程的影响。例如，在支付挂号费的流程中，用户的选择直接决定了支付平台的跳转和支付状态的确认，这些逻辑通过活动图呈现出来，更容易识别潜在的问题和优化的空间。

实验中，除了注重技术实现，还考虑到了用户体验。例如，系统设计了用户通过手机验证码登录、第三方微信登录等多种登录方式，增强了系统的便利性和用户体验。此外，系统还设计了预约成功后打印预约单、挂号费支付和退费等功能，确保了系统在实际应用中的可操作性和灵活性。

在设计系统时，除了常规流程外，异常情况和边界条件的处理也至关重要。例如，用户预约挂号时，如果医生的预约人数已满，系统应及时提示用户并提供其他可预约医生。对于未按时取消预约的用户，系统还需要考虑到信用等级的管理和罚款机制，确保系统的公平性和约束力。这些异常情况的处理不仅提升了系统的稳定性，还增强了系统的容错性。

通过本次“医院预约挂号系统”的实验，我进一步加深了对面向对象分析与设计的理解，并学会了如何使用用例图、用例规格说明、活动图等工具来设计一个系统。这些工具不仅能帮助我们从不同角度分析问题，还能确保系统设计的完整性与可行性。我也更加明白了系统设计中每个环节的细节和重要性，未来在进行类似项目时，我会更加注重需求分析、系统设计以及边界条件的考虑，从而确保项目的高效实施和优质交付。

仓库地址

https://github.com/Li-Keanu/school-software