{项目名称}  
软件设计规格说明书

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  [ ] 正在修改 | 文件标识： |  |
| 当前版本： | V1.0 |
| 作 者： |  |
| 完成日期： | 2018-03-16 |

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V0.1 | 邵腾 | 宋博文 | 2018/3/8 |  |
| V1.0 | 邵腾 | 宋博文 | 2018/3/15 |  |
|  |  |  |  |  |

目录

[1 概述 3](#_Toc508974150)

[2 设计目标和约束 3](#_Toc508974151)

[3 用例视图 4](#_Toc508974152)

[3.1 用户信息登记用例实现 4](#_Toc508974153)

[3.2 用户信息维护用例实现 5](#_Toc508974154)

[3.3 病例信息登记用例实现 5](#_Toc508974155)

[3.4 病例信息维护用例实现 6](#_Toc508974156)

[3.5 病例检索用例实现 6](#_Toc508974157)

[3.6 考题信息登记用例实现 7](#_Toc508974158)

[3.7 考题信息维护用例实现 7](#_Toc508974159)

[3.8 试卷信息生成用例实现 8](#_Toc508974160)

[3.9 考试信息生成用例实现 8](#_Toc508974161)

[3.10 图片上传用例实现 9](#_Toc508974162)

[3.11 视频管理用例实现 9](#_Toc508974163)

[4 逻辑视图 10](#_Toc508974164)

[4.1 概述 10](#_Toc508974165)

[4.2 包设计 10](#_Toc508974166)

[5 进程视图 （进程，用户端，服务器端，进程之间的关系 类似构件图） 11](#_Toc508974167)

[6 部署视图 12](#_Toc508974168)

[7 实现视图 13](#_Toc508974169)

# 概述

描述本文档的目的是明确。需求，帮助团队更好地开发。

本系统适用于医疗教育领域，为宠物医院提供毕业实习生学习系统，非指定客户（医院）医学毕业实习生无法通过验证。

虚拟宠物医院学习系统分为·医院导览，职能学习，系统管理，病例管理，测试管理五大模块。旨在为有志于进入宠物医生行业的实习生提供一个自主学习，自我评测的平台，可以使得宠物工作者不去实体医院就能系统地学习各种宠物诊疗专业知识，能够使得毕业实习生了解宠物医院结构、科室、进行病例学习等，为今后的宠物医生的生涯打下良好基础。·

项目计划在4/26之前完成，开发成本控制在500元以内。计划在前端采用3D导览，目标是比较逼真。病例学习和测试模块目标是符合用户的使用习惯，界面友好美观。本软件计划设置不同岗位角色并配备约200个左右的真实病例。

对于投资方医院来说，虚拟宠物医院学习系统可以使实习医生的培训工作系统化，流程化，方法多元化。也节约了部分培训医生的人力和时间成本。同时，由于学习系统是在线的，因此可以不限时间和地点地进行回放，保证知识点不丢失。同时测试管理模块保证培训的质量。

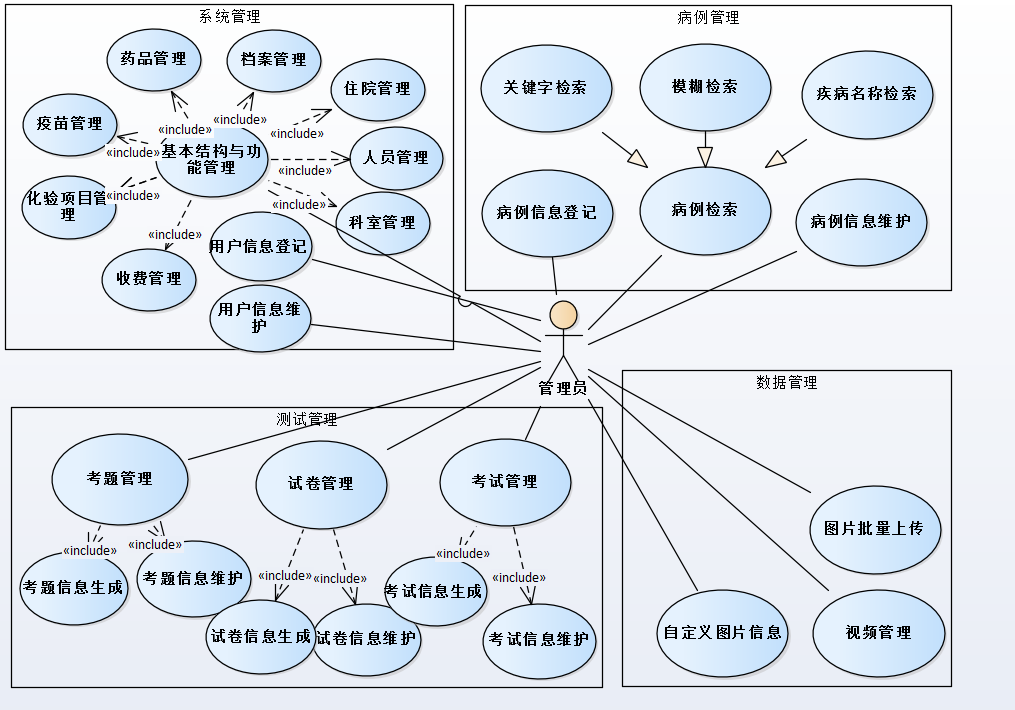
对于开发人员来说，开发的过程是一次很好的体验团队开发较大项目的体验，会更加熟悉版本控制，团队内和团队间协作，以及团队的分工等技能，是一次很好的真实开发的模拟，为大四学生求职，以及今后的开发生涯都是很好的储备。

# 设计目标和约束

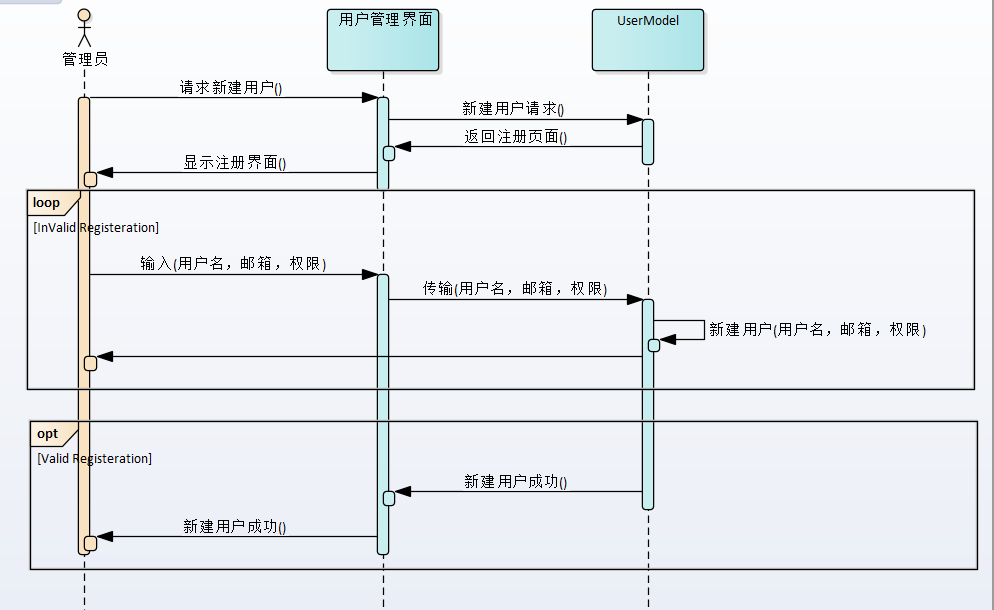
整个系统计划基于Python的Django框架来实现，数据库是mysql,前台的3D效果计划用WebGL实现，开发的集成环境是Pycharm。从需求设计到完成开发到完成测试，总项目时间预计8周。开发计划采用增量模型（Incremental Model）。代码的版本控制是通过github。4人团队结构是：产品经理+架构师+开发人员+测试人员。

本系统架构旨在开发完成项目宠物医院学习系统的后台部分，主要功能包括系统管理，病例管理，测试管理，数据管理。系统开发的目标给项目的管理员搭建一个良好的平台去更好地实现对整个项目后台部分的有效管理。

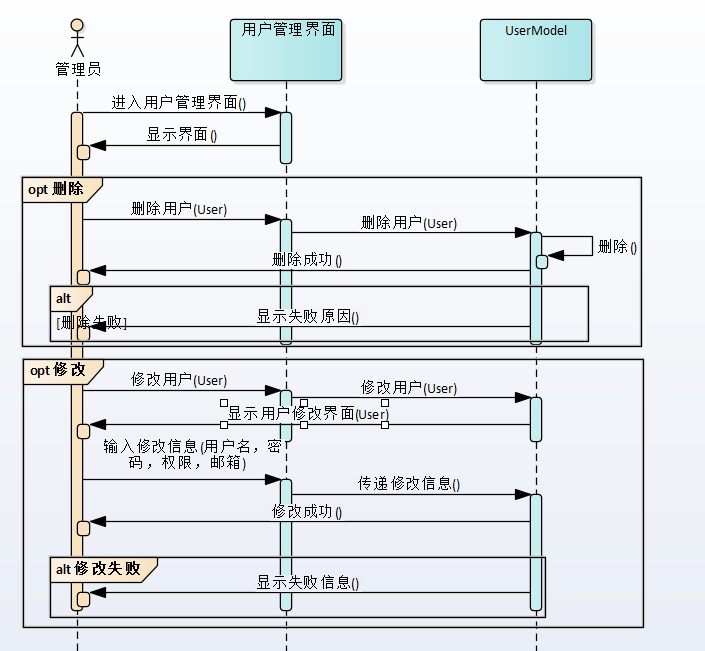
# 用例视图



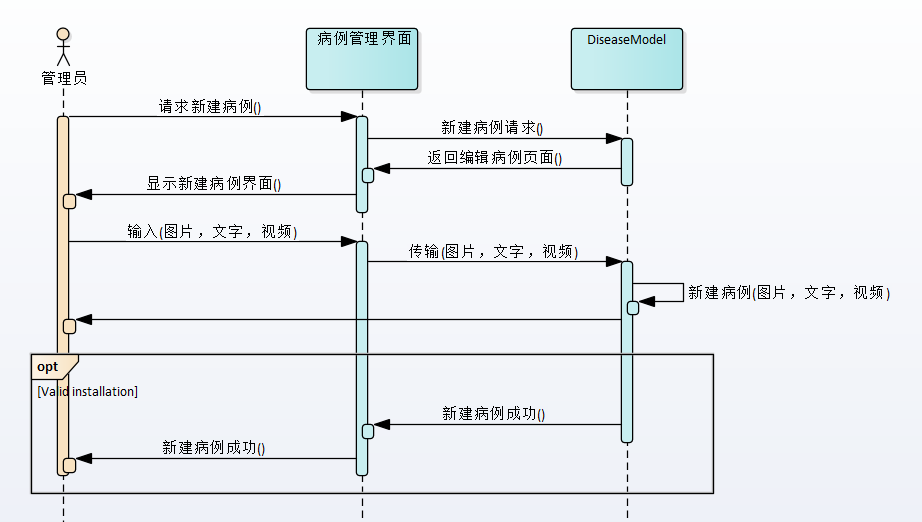
## 用户信息登记用例实现



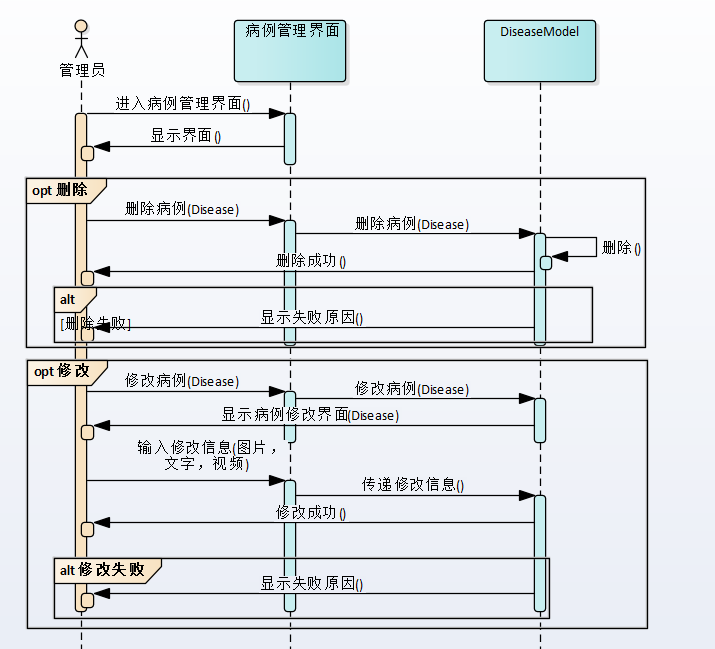
## 用户信息维护用例实现



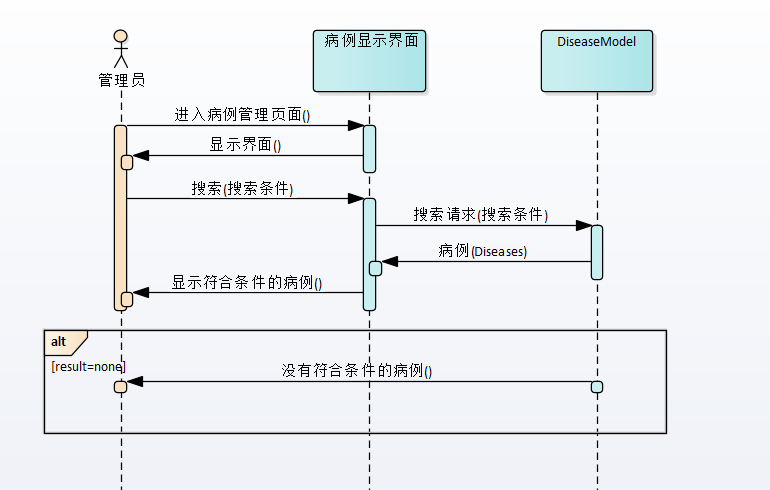
## 病例信息登记用例实现



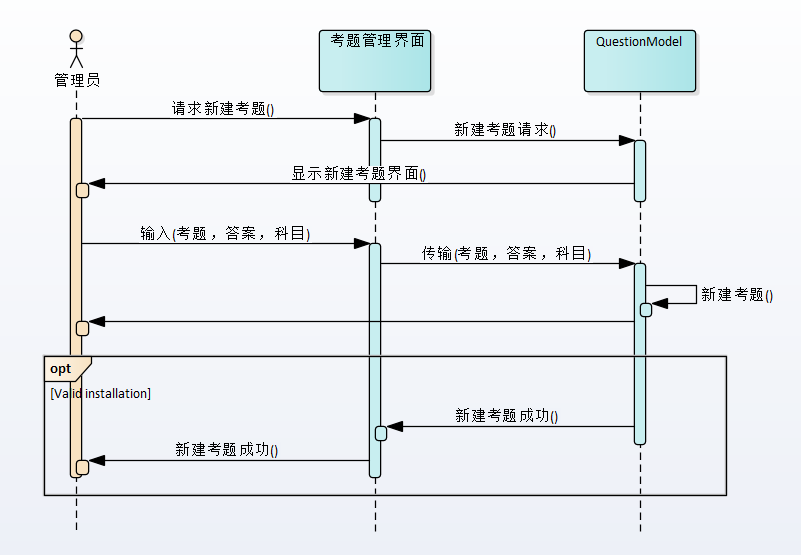
## 病例信息维护用例实现



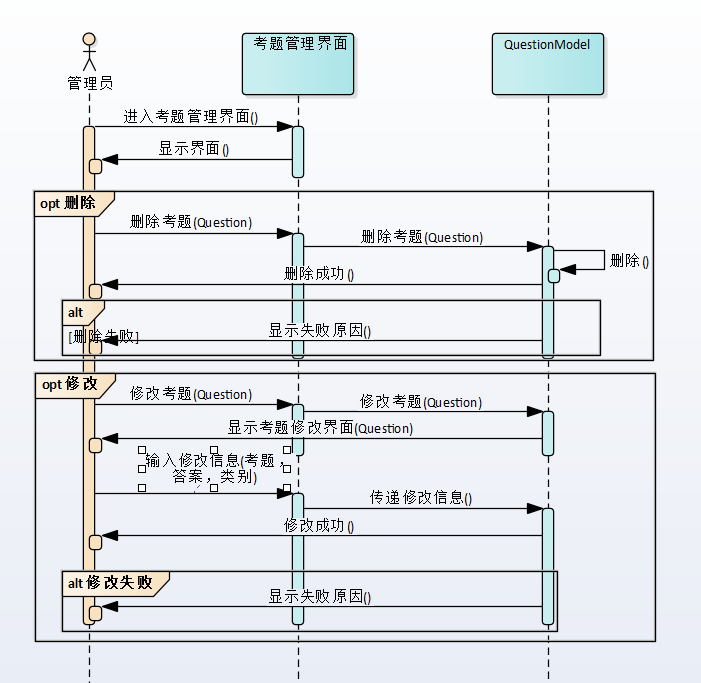
## 病例检索用例实现



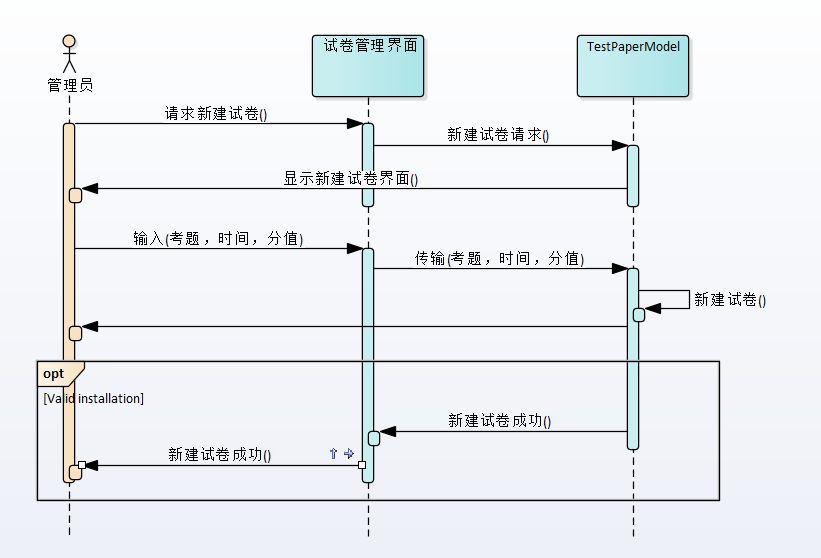
## 考题信息登记用例实现



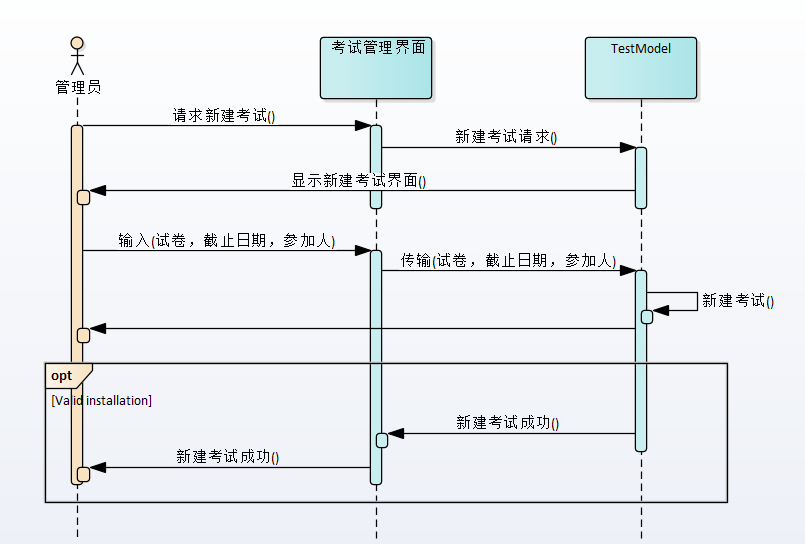
## 考题信息维护用例实现



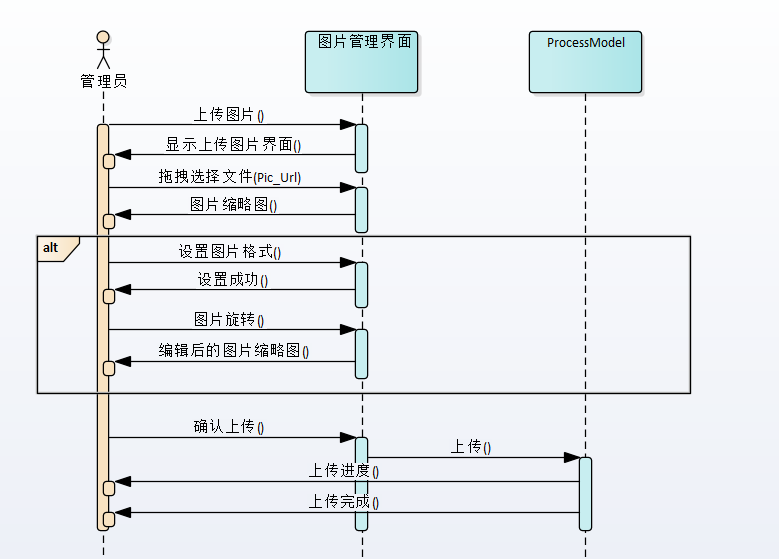
## 试卷信息生成用例实现



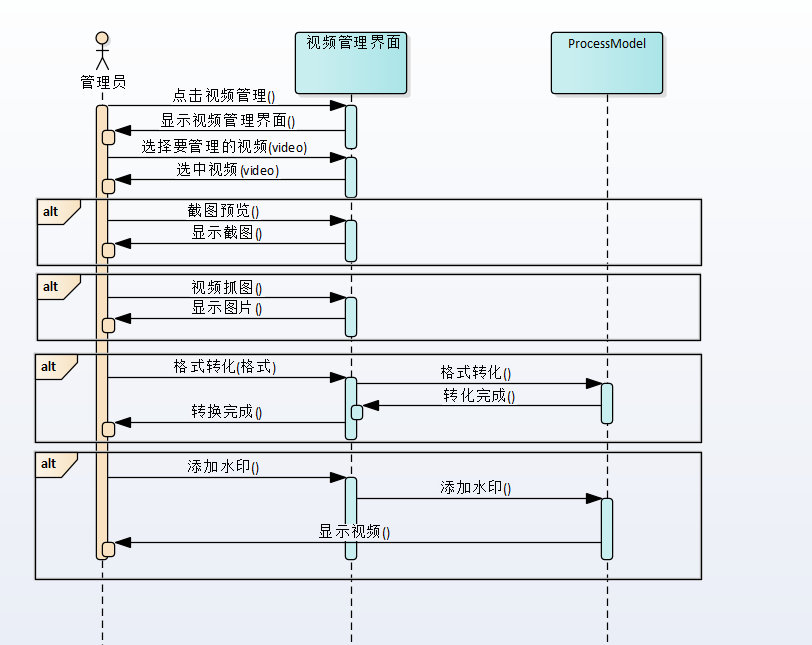
## 考试信息生成用例实现



## 图片上传用例实现



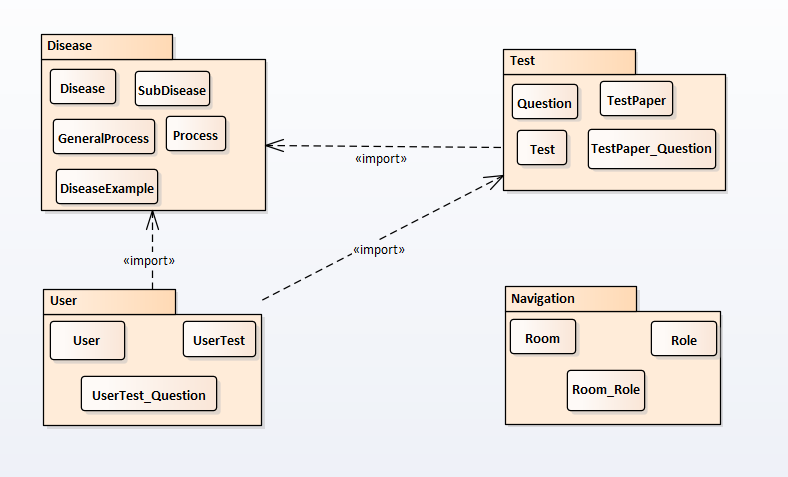
## 视频管理用例实现



# 逻辑视图

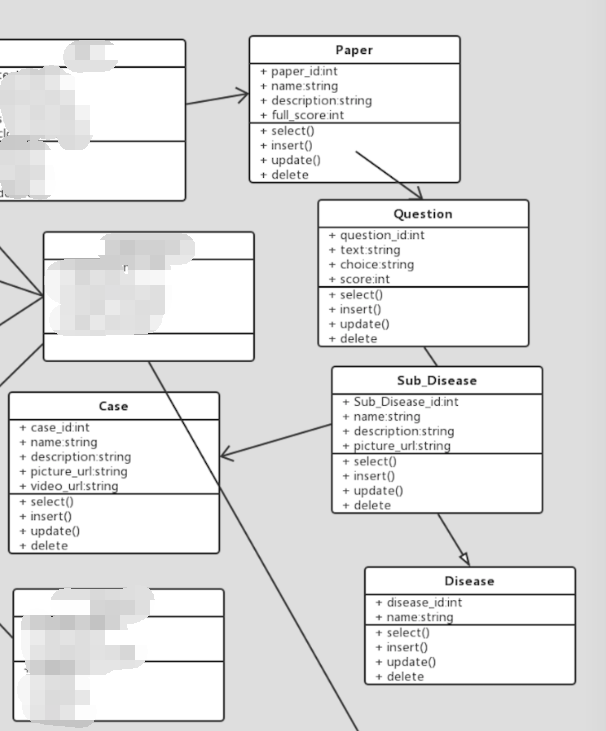
## 概述

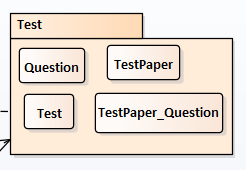
整个项目包括4个子功能模块：病例部分（包括病例学习和病例管理）Disease，测试部分（包括在线测试和测试管理）Test，用户部分（包括用户操作和用户管理）User，3D导览部分Navigation，系统管理

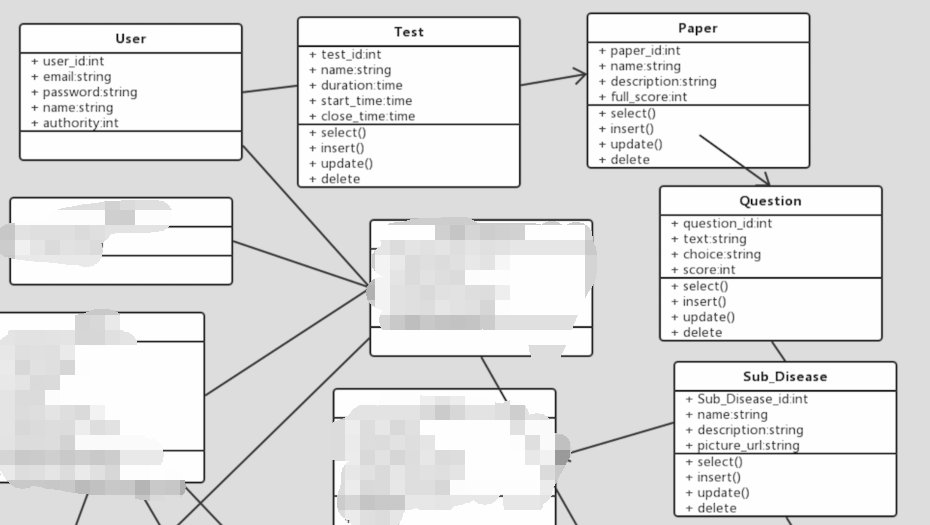


## 包设计

Disease包中有以下的表：Disease（疾病大类）, SubDisease（疾病细分）,GeneralProcess（疾病概述），DiseaseExample（疾病例子），Process（病例信息，包括文字/图片/视频的描述）。其中Diasease会与Test包中的Test表关联，因为试卷会按照病种来出。

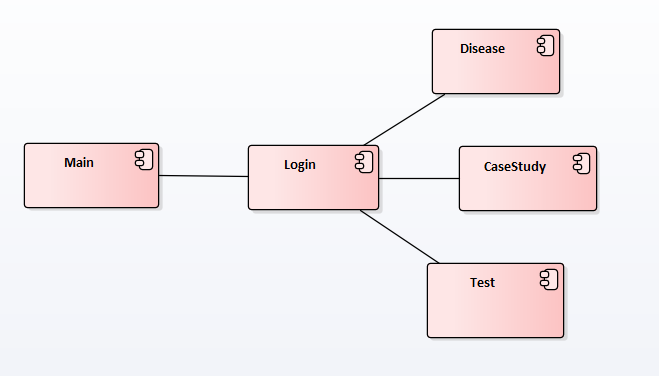


Test包中有一下的表：Question（试题），Choice（选项），TestPaper（试卷），Test（考试），TestPaper\_Question（试卷和试题的关系），其中Test会与User有一个多对多的关系，Question会与疾病种类有联系

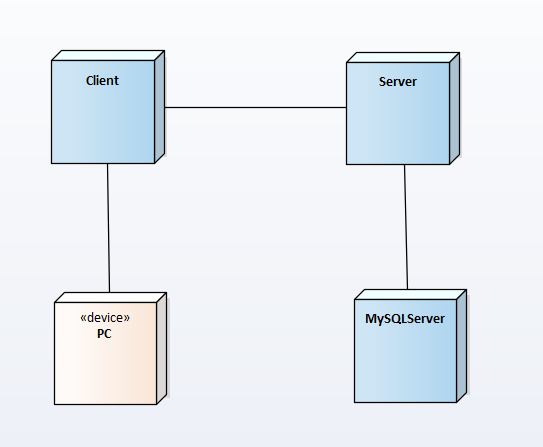


# 进程视图 （进程，用户端，服务器端，进程之间的关系 类似构件图）

提示：进程视图侧重系统的运行特性，关注非功能性的需求（性能，可用性）。服务于系统集成人员，方便后续性能测试。强调并发性、分布性、集成性、鲁棒性（容错）、可扩充性、吞吐量等。定义逻辑视图中的各个类的具体操作是在哪一个线程（Thread）中被执行。



# 部署视图



# 实现视图

