{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： | V0.1 |
| 作 者： |  |
| 完成日期： | 2018-03-08 |

版 本 历 史

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

目录

[1 概述 3](#_Toc508308549)

[2 设计目标和约束 3](#_Toc508308550)

[3 用例视图 3](#_Toc508308551)

[3.1 用户信息登记用例实现 3](#_Toc508308552)

[3.2 用户信息维护用例实现 4](#_Toc508308553)

[3.3 病例信息登记用例实现 4](#_Toc508308554)

[3.4 病例信息维护用例实现 4](#_Toc508308555)

[3.5 病例检索用例实现 4](#_Toc508308556)

[3.6 考题信息登记用例实现 5](#_Toc508308557)

[3.7 考题信息维护用例实现 5](#_Toc508308558)

[3.8 试卷信息生成用例实现 5](#_Toc508308559)

[3.9 考试信息生成用例实现 6](#_Toc508308560)

[3.10 图片批量上传用例实现 6](#_Toc508308561)

[3.11 自定义上传图片信息用例实现 6](#_Toc508308562)

[3.12 视频管理用例实现 7](#_Toc508308563)

[4 逻辑视图 7](#_Toc508308564)

[4.1 概述 7](#_Toc508308565)

[4.2 包设计 7](#_Toc508308566)

[5 进程视图 8](#_Toc508308567)

[6 部署视图 8](#_Toc508308568)

[7 实现视图 8](#_Toc508308569)

# 概述

*提示：概要描述本文档的目的、范围、定义、首字母缩写、缩略语、引用和软件架构的概述*

描述本文档的目的是明确。

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。

# 设计目标和约束

*提示：本节描述会显著影响整个系统架构的系统需求和目标,例如,安全、安全、隐私、第三方开发包、可移植性、分配和重用等。同时，也定义了可能适用于设计和实现策略、开发工具、团队结构、时间表、遗留代码等的约束条件*

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

*列出与系统功能相关的用例和用例场景，并设计软件系统是如何实现这些用例及场景*

## 用户信息登记用例实现

|  |  |
| --- | --- |
| **用例名称** | 用户信息登记 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.管理员新建用户，输入用户名和邮箱，默认密码是111111，然后勾选新用户权限  2.新用户创建完成 |

## 用户信息维护用例实现

|  |  |
| --- | --- |
| **用例名称** | 用户信息维护 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.管理员进入用户的信息界面  2.管理员选择要修改的内容（用户名，密码，权限，邮箱），进行修改  2.1管理员删除用户  3.用户信息维护完成 |

## 病例信息登记用例实现

|  |  |
| --- | --- |
| **用例名称** | 病例信息登记 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.管理员点击新建病例  2.对于病例中的各个过程：患宠典型的临床症状、检查过程及结果，诊断结果及治疗方案等，选择图片/文字/视频加以描述  3.新用例登记实现 |

## 病例信息维护用例实现

|  |  |
| --- | --- |
| **用例名称** | 病例信息维护 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.管理员进入病例的信息界面开始维护  2.管理员选择病例中要修改的内容（图片，文字，视频），进行修改  2.1管理员删除病例  3.病例信息维护完成 |

## 病例检索用例实现

|  |  |
| --- | --- |
| **用例名称** | 病例检索 |
| **用例场景** | who：用户（管理员/实习生）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.用户在搜索框输入搜索的条件（支持模糊搜索，关键字搜索，病例名称搜索）  2.显示病例查询条件的相关病例  2.1如果没有符合查询条件的病例，ze1返回无  3.病例检索成功 |

## 考题信息登记用例实现

|  |  |
| --- | --- |
| **用例名称** | 考题信息登记 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.管理员点击新建考题  2.管理员输入试题信息，考题答案，以及该题所属病种  3.新考题信息登记实现 |

## 考题信息维护用例实现

|  |  |
| --- | --- |
| **用例名称** | 考题信息维护 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.管理员进入考题的信息界面开始维护  2.管理员选择考题中要修改的内容（考题，答案，所属病种），进行修改  2.1管理员删除考题  3.考题信息维护完成  3.1如果维护失败，则返回说明失败原因 |

## 试卷信息生成用例实现

|  |  |
| --- | --- |
| **用例名称** | 试卷信息登记 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **前置条件** | 题库中已经有考题 |
| **用例描述** | 1.管理员点击新建试卷  2.管理员选择试题添加到试卷，设置考试时间，设置每道题的分数以及试卷总分  3.新试卷信息登记实现 |

## 考试信息生成用例实现

|  |  |
| --- | --- |
| **用例名称** | 考试信息登记 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **前置条件** | 题库中已经有试卷 |
| **用例描述** | 1.管理员点击新建考试  2.管理员相应的试卷，设置考试的截止日期，以及可以参加这场考试的学生  3.新考试信息登记实现 |

## 图片批量上传用例实现

|  |  |
| --- | --- |
| **用例名称** | 图片批量上传 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.用户针对特定病例批量上传图片，且上传功能支持BMP，JPG，GIF，PNG等多种图片格式，在图片上传过程中支持拖拽文件以及图片自动生成缩略图  2.上传过程中自动生成图片缩略图，以及显示上传进度，在所选中的全部文件上传完毕后显示上传成功，如果在上传过程中发生错误就显示错误  3.图片批量上传成功  3.1如果上传失败，则返回并显示原因 |

## 自定义上传图片信息用例实现

|  |  |
| --- | --- |
| **用例名称** | 自定义上传图片信息 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **用例描述** | 1.用户编辑想要上传的图片信息  2.用户设置图片格式，快速编辑图片  3.自定义完成 |

## 视频管理用例实现

|  |  |
| --- | --- |
| **用例名称** | 视频管理 |
| **用例场景** | who：用户（管理员）  where：虚拟宠物医院管理系统 |
| **前置条件** | 用户已上传视频到后台 |
| **用例描述** | 1.用户选择想要管理的视频  2.用户对选中的视频进行操作（格式转化，视频抓图，截图预览以及添加水印等）  3.显示操作后的视频 |

# 逻辑视图

*提示：本节主要定义架构上重要部分的设计模型,例如分解成的子系统和包。并为每个重要的包,定义相关的类和实体类，描述其职责和一些重要的关系,操作,和属性。*

## 概述

*提示：可以使用包或者层的形式描述整个系统架构的组成模型*

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

## 包设计

描述重要包的构成，与其它包之间的关系

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）

# 进程视图

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

*提示：部署视图显示的是系统的实际部署情况，它是为了便于理解系统在一组处理节点上的物理分布。在系统中，只包含有一个部署视图，用来说明各种处理活动在系统各节点的分布。但是，这个部署视图可以在每次迭代过程中都加以改进。部署视图中包括进程、处理器和设备。进程是在自己的内存空间执行的线程；处理器是任何有处理功能的机器，一个进程可以在一个或多个处理器上运行；设备是指没有任何处理功能的机器。*

# 实现视图

提示：本节描述了实现模型的整体结构,包括在实现模型中的系统分解层次和子系统部件

