{虚拟宠物医院学习软件}  
 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： | 0.1 |
| 作 者： | 李琪琦、潘星辰 |
| 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 |
| V0.1 | 李琪琦 | 潘星辰 | 2018/03/08 |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 概述 3](#_Toc507590260)

[2 设计目标和约束 3](#_Toc507590261)

[3 用例视图 3](#_Toc507590262)

[3.1 XX用例实现 3](#_Toc507590263)

[3.2 XX用例实现 3](#_Toc507590264)

[3.3 XXX用例实现 3](#_Toc507590265)

[4 逻辑视图 3](#_Toc507590266)

[4.1 概述 3](#_Toc507590267)

[4.2 包设计 3](#_Toc507590268)

[5 进程视图 3](#_Toc507590269)

[6 部署视图 4](#_Toc507590270)

[7 实现视图 4](#_Toc507590271)

# 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*

1.1文档目的：

说明虚拟宠物医院项目的需求分析后的产物，包括项目的结构设计、实体类设计、功能说明、用例设计等内容。其最终目的是根据《虚拟宠物医院学习软件规格说明书》中的需求，设计出一套安全可行的软件系统模型。

1.2范围与作用：

本文档包括了设计的目标；团队人员分配、开发环境、进度计划等约束；功能具体描述说明；逻辑结构以及系统层次等设计说明。供项目经理、开发人员、测试人员查阅。

1.3软件架构：

该虚拟宠物医院学习软件系统主要采用前后端分离的方式，采用Spring REST 微服务的方式，拦截前端发送的URL请求，并返回处理结果。通过独立的模块化地设计来达到高内聚、低耦合的架构目标。

# 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*

2.1设计目标：

构建一套满足需求并且符合高内聚、低耦合、线程安全等目标的软件系统。

2.2 约束：

2.2.1隐私与安全

在架构的过程中，要考虑在高并发情况下的线程安全以及用户信息隐私安全的约束。

2.2.2 第三方开发包

通过Maven进行项目中的软件包管理，以约束引用软件的版本控制。

2.2.3 设计与实现策略：

关系型数据做数据存取；利用Spring做类实例管理，利用Spring boot应用来启用后端服务，架构一个可执行的微服务。

2.2.4 开发工具

数据库：MySQL

开发语言：JAVA 1.8  
后端框架：Spring Boot

依赖管理：Maven

集成开发环境：Eclipse

2.2.5团队结构

项目经理1名，架构师1名，配置管理员1名，测试人员1名。这四人共同承担开发工作。

2.2.6时间表

开发时长：2018年3月2日-2018年4月26号。

# 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*

## XX用例实现

## XX用例实现

## XXX用例实现

# 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*

项目架构模型：Spring REST Web Service.

## 包设计

描述重要包的构成，与其它包之间的关系

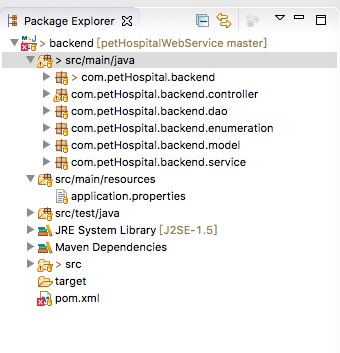
Controller包：拦截HTTP请求，处理接受后的数据信息；

Service包：供控制器调用，处理复杂的业务逻辑；

DAO：持久化层，利用JPA与Hibernate访问MySQL 数据库中的数据；

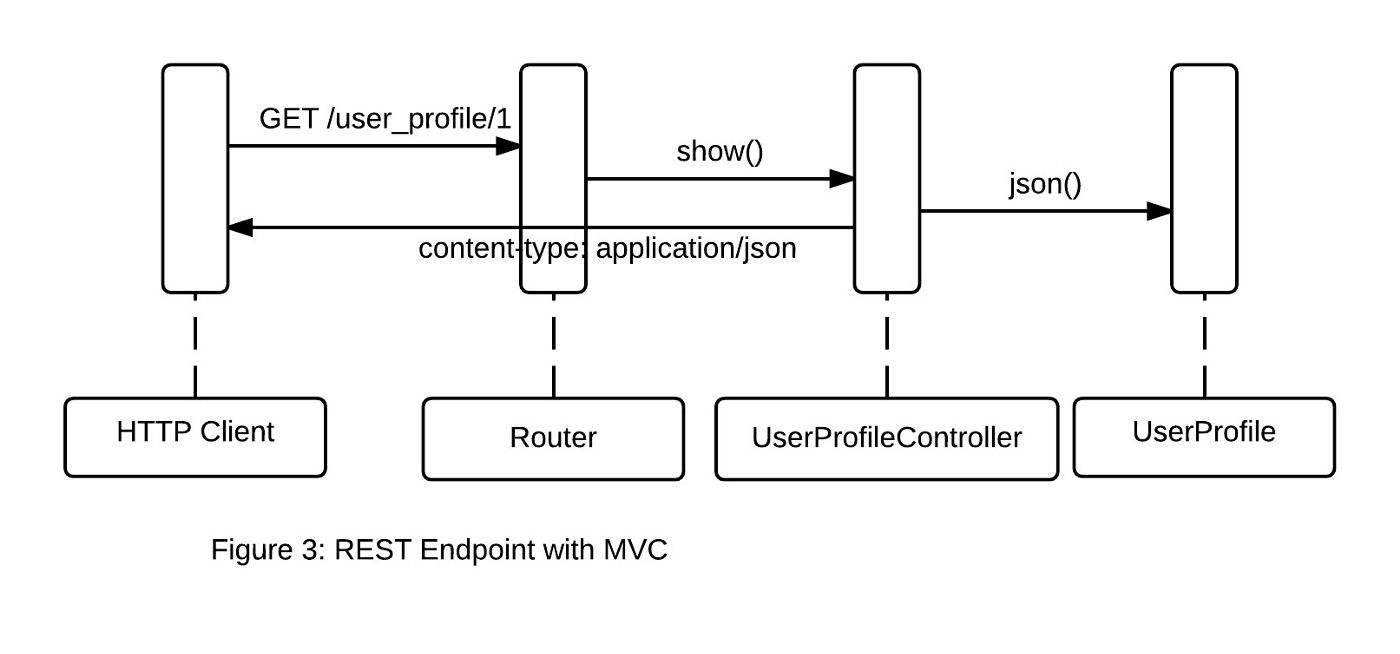
Model：实体类层：映射数据库表结构。

Enum包：放入一些以枚举来表示的特殊数据类型，以便清晰属性含义。



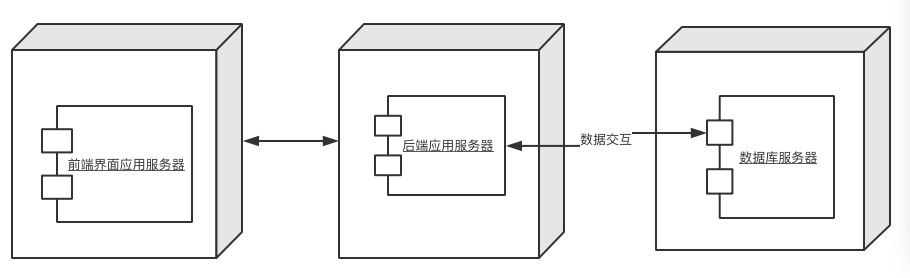
# 进程视图

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

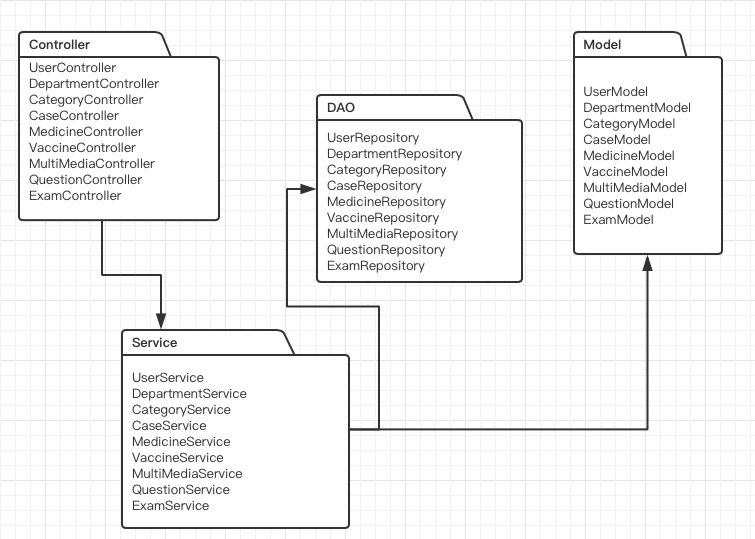
*提示：部署视图显示的是系统的实际部署情况，它是为了便于理解系统在一组处理节点上的物理分布。在系统中，只包含有一个部署视图，用来说明各种处理活动在系统各节点的分布。但是，这个部署视图可以在每次迭代过程中都加以改进。部署视图中包括进程、处理器和设备。进程是在自己的内存空间执行的线程；处理器是任何有处理功能的机器，一个进程可以在一个或多个处理器上运行；设备是指没有任何处理功能的机器。*

**

# 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

层次模型：



Entity模型：

