虚拟宠物医院学习系统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： | 李沐凡 |
| 完成日期： | 2018-3-8 |

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| v0.1 | 李沐凡 | 李沐凡 | 2018/3/8~2018/3/8 |  |
|  |  |  |  |  |
|  |  |  |  |  |

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 登录用例实现 3](#_Toc507590263)

[3.2 注销用例实现 3](#_Toc507590264)

[3.3 医院导览用例实现 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

## 编写目的

1. 本文档主要介绍了宠物医院学习系统前台的初步设计。
2. 本文档中对系统结构进行明确，安排项目规划和进度、指定初步的开发和测试计划。
3. 本文档供软件开发各方人员进行参考。

## 范围

本文档主要包含宠物医院学习系统前台的各项设计。

## 软件架构

宠物医院学习系统前台包括展示层、业务逻辑层、数据层三层。

其中展示层主要负责向用户展示信息和数据。

业务逻辑层主要负责处理各种业务逻辑，主要使用Python进行开发。

数据库层负责数据的存储，使用MySQL对数据进行存储，并使用Redis对常用数据进行缓存。

# 设计目标和约束

## 安全

用户在登录系统的时候需要进行身份验证并获得一个用于在后续操作中进行身份验证的Token。

用户名和密码均以加密的方式进行发送。

在输入框中使用字符替换、存储过程等方式进行防注入处理。

前后台分离，确保前台用户无法访问后台控制系统。

## 隐私

用户信息存储在后端数据服务器上，数据服务器不运行外网访问。

## 第三方开发包

Nginx反向代理、Redis内存数据库、Flask框架、Vue.js框架、Mysql数据库等

可移植性：

使用可移植语言开发，保证在多种系统上迁移的时不需要进行修改

## 重用

前端代码中数据展示方式相同的页面进行重用。

## 实现策略

展示页面和业务逻辑相分离，两者之间通过网络进行通信。

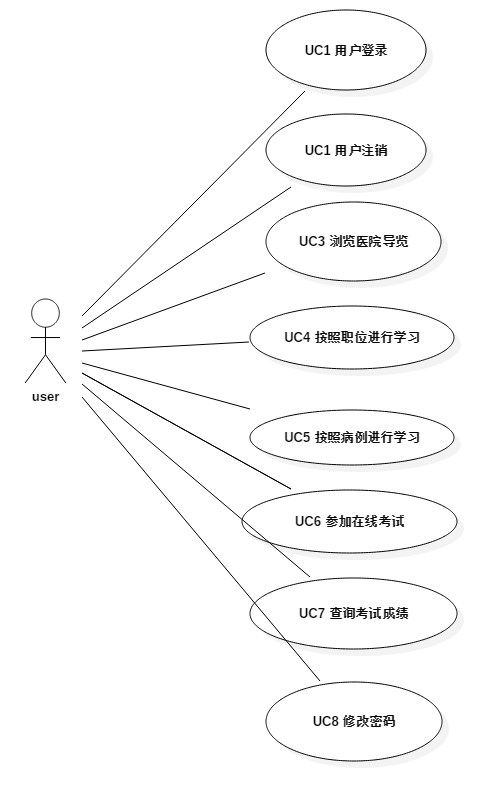
## 开发工具

展示层页面主要使用Veu.js进行开发，业务逻辑主要由基于Python语言的Flask进行开发。两者直接使用Http请求进行通信，通信数据使用Google Protocol buffer进行序列化和反序列化。数据层使用MySQL和Redis进行开发。

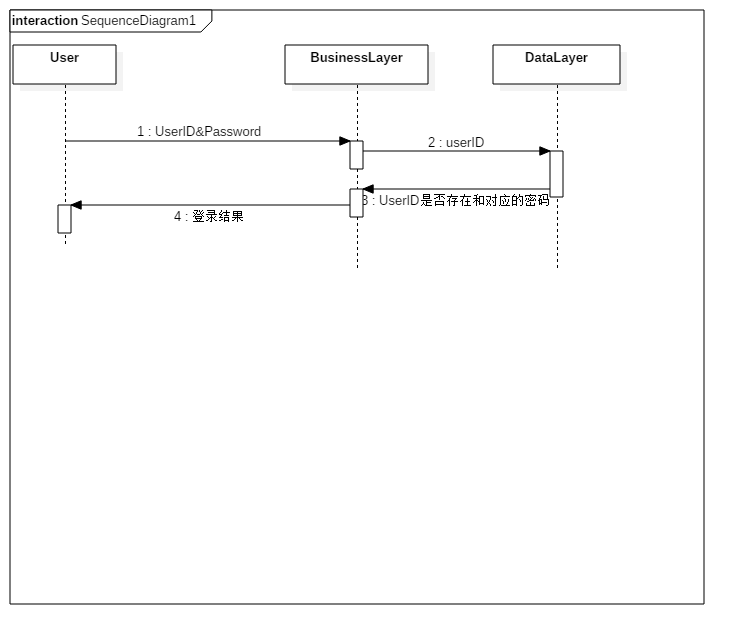
# 用例视图

*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*

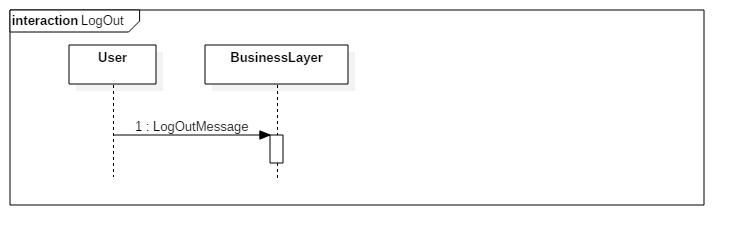
用例视图：

**

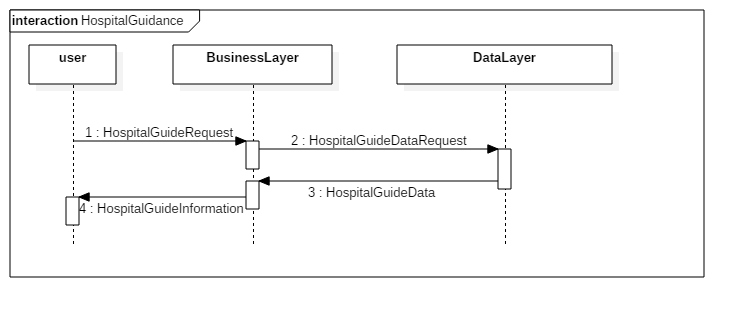
## UC-1登录用例实现



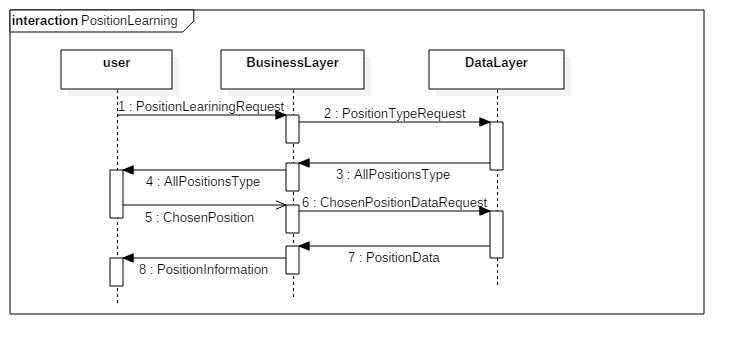
## UC-2 注销用例实现



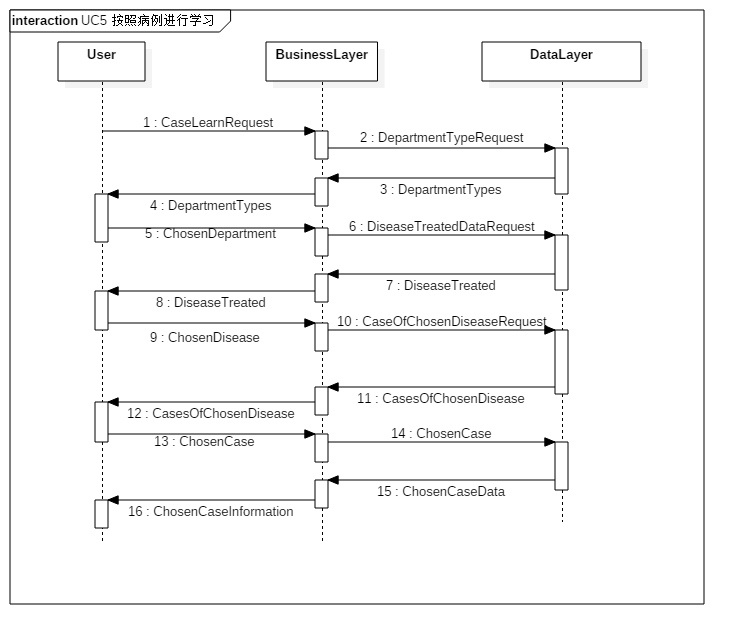
## UC-3浏览医院导览用例实现



## UC-4 按照职位进行学习



## UC-5 按照病例进行学习



# 逻辑视图

## 概述

本系统按照层次分主要分为三个层，视图层、业务逻辑层和数据层。

按照包分为登录登出包、信息展示包、考试包、

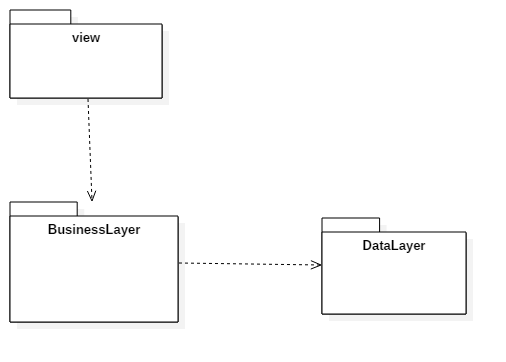
## 包设计

View包主要包括前端页面中的html代码和JavaScript代码。

BusinessLayer包主要包含处理业务逻辑的Python代码。

DataLayer包主要包含数据库表结构设计和Prorobuf数据定义文件。

包之间的关系如下图所示

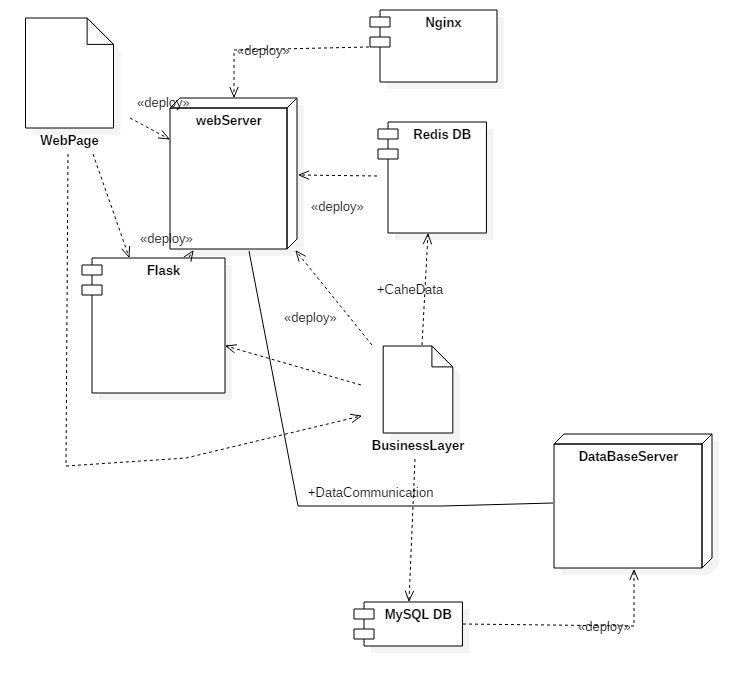


# 进程视图

无

# 部署视图

前端页面和业务逻辑层部署在多台网站服务器上，使用Flask框架处理对前端页面的请求，使用Nginx进行反向代理。Redis数据库也部署在服务器上。MySQL单独部署在数据库服务器上。



# 实现视图

暂无