|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | 虚拟宠物医院学习系统软件架构设计说明书 |
| 当前版本： | 1.0 |
| 作 者： | 项目第十五组全体成员 |
| 完成日期： | 2024-3-22 |

虚拟宠物医院学习系统

软件架构设计说明书

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| v0.1 | 项目第十五组 | 项目第十五组 | 2024/3/13-2024/3/15 | 架构设计初稿 |
| v1.0 | 项目第十五组 | 项目第十五组 | 2024/3/16-2024/3/22 | 架构设计基线 |
|  |  |  |  |  |

目录

[1 概述 1](#_Toc161840731)

[2 设计目标和约束 2](#_Toc161840732)

[3 架构设计 3](#_Toc161840733)

[3.1 总体方案 3](#_Toc161840734)

[3.2 架构说明 3](#_Toc161840735)

[3.2.1 架构图及说明 3](#_Toc161840736)

[3.2.2 架构设计关键点 4](#_Toc161840737)

[3.2.3 高可用性设计 4](#_Toc161840738)

[3.2.4 高性能设计 4](#_Toc161840739)

[3.2.5 可扩展性设计 4](#_Toc161840740)

[3.2.6 安全性设计 4](#_Toc161840741)

[3.2.7 其他设计 5](#_Toc161840742)

[4 部署方案 6](#_Toc161840743)

# 概述

（1）目的：

本文档表述了虚拟宠物医院学习系统的框架，描述本系统的各个主要方面，记录了项目成员对系统架构的决策。

项目经理根据架构定义的构件结构制定项目的开发计划；程序开发设计人员据此进行各构架的设计；测试设计员据此设计系统总体测试框架；架构文档还可指导各构件的实施、集成及测试。

本架构文档的预期阅读人员为项目第十五组成员。

（2）范围：本架构文档适用于虚拟宠物医院学习系统的总体应用框架。

1. 定义：

前端：

1. Vue3：流行前端框架Vue的第三个主要版本
2. JavaScript/TypeScript：前端开发语言，实现交互效果
3. Vite：新型前端构建工具
4. element-plus：一种基于Vue3的组件库
5. Swaager:统一接口管理
6. IDEA2022:软件开发工具
7. ApiFox: 前后端接口交互

后端：

1. Spring Boot3.0.5：后端开发框架
2. Java：后端开发语言
3. Maven：项目管理工具
4. JDK17: 后端Java版本
5. Redis7,0,1:数据库缓存
6. Mysql8.0.30:数据库
7. Navicat15：主流数据库管理工具
8. Mybatis-plus：增强版数据管理持久层框架
9. Microsoft.Garnet 1.0.0：微软远程缓存存储

1. 首字母缩写：
2. 缩略语：
3. 引用：
4. 软件架构概述：

# 设计目标和约束

目标：

1. 性能：响应时间不超过3秒
2. 成本：成本低，可忽略
3. 时间：2024年3月至2024年4月中旬
4. 可靠性：在任何情况下可保持正常运行
5. 安全性：防止未授权的访问，确保账户权限管理正常，保障数据机密性、完整性
6. 合规性：遵循质量管理体系、行业标准、安全级别
7. 技术性：采用流行、稳定的技术栈与工具
8. 兼容性：确保系统在不同操作系统、硬件平台、软件环境保持一致性、稳定性

约束：

1. 技术与工具约束：
2. 技术栈：前端小组为vue3 + typescript + element-plus + vite，

后端小组为SpringBoot+Mysql+Maven+Redis+Mybatis-Plus

1. 工具：前端小组为VSCode + figma，

后端小组为IDEA+Vmware+Navicat

前后端接口对接：ApiFox

1. 团队结构约束：

|  |  |  |
| --- | --- | --- |
| **人员** | **角色** | **职责** |
| 王胜煜 | 前端开发/架构 | 搭建前端环境、页面开发 |
| 张韵 | 前端开发/测试 | 设计UI界面、页面功能测试 |
| 陈泽镐 | 前端开发/测试 | 设计UI界面、页面开发 |
| 潘煦森 | 后端开发/项目经理 | 搭建后端环境，功能开发 |
| 张寒 | 后端开发 | 功能测试，功能开发 |
| 王盛阳 | 后端开发 | 规划整体架构，功能开发 |

# 架构设计

## 总体方案

## 架构说明

### 架构图及说明

图3.1 架构图

本项目采用前后端分离的架构模式，前端部分在访问层实现用户交互页面，使用html，Css和element-plus组件库实现前端UI，在交互层构建页面组件下载依赖、实现路由跳转以及调用后端提供的接口。

后端部分主体框架为controller，service和dao三层架构，在controller 层调用service层的功能并通过提供封装成供前端使用的接口实现用户请求和响应，service层负责实现业务逻辑，dao层着眼于对数据库数据的调用以及增删改查，数据层则定义了后端各层的实体类和数据对象，数据库存放系统的数据。

### 架构设计关键点

在Controller层采用Restful风格编写满足Swagger要求的api接口，在数据Model层，后端将前台请求的参数封装为Param，将提供给前端的视图参数封装在vo中，做到了高内聚低耦合。整体架构层层解耦，将每个大的功能拆分为小的功能，通过各个层级的交互来实现。

### 高可用性设计

1. 采用Redis、Garnet与数据库的多级缓存机制，提高了数据处理的效率

### 高性能设计

1. 权限信息采用位图设计，便于位运算，提高了性能
2. 异步加载，当用户点击了页面后才开始加载对应的页面，避免了同一时间内加载过多内容导致性能降低
3. 采用多级索引，提高数据的查找效率

### 可扩展性设计

1. 通过独立的药品库、科室库、人员库，添加依赖，避免修改、删除项时对数据表造成大量改动与空值错误
2. 用户身份权限采用位图设计，节省数据库表空间的同时支持新身份的加入
3. 需求功能模块化，采用低耦合设计，便于新增其他功能

### 安全性设计

1. 权限管理

用户完成注册账户后统一由系统分配为学习者角色权限，如果实际身份为管理人员则向超级管理员提交申请请求超级管理员分配相应权限。用户权限与用户账号绑定，一个账户只能对应一个身份享有相应身份对应的权限，如果要变更账户权限则需要向超级管理员申请改变账户角色，由超级管理员统一授予用户权限。

没有权限的用户无法访问管理员用户的界面，也不可通过修改url的方式越权访问。

用户信息的数据表内包含权限项，在登录时读取该项确定用户的权限。

1. 预防SQL注入

采用字符串过滤的方式，避免SQL注入攻击

1. 预防XSS注入

对输入框、表单等文本交互之处过滤关键字

1. 互斥锁

在数据管理员修改数据时，其他用户不可对正在修改的项进行读写。

### 其他设计

指定开发语言：JavaScript、CSS、html，Java、Sql

# 部署方案

图4.1 部署视图

前端将服务器部署在Nginx代理服务器上进行反向代理，并将网络请求分发到后端部署的应用级服务器Tomcat和云服务器上实现负载均衡。后端服务器先向Redis查询缓存，再请求Mysql数据库。