|  |  |  |
| --- | --- | --- |
| 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： | 0.1 |
| 作 者： | 项目第十五组全体成员 |
| 完成日期： | 2024-3-15 |

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 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 概述 1](#_Toc1505410650)

[2 设计目标和约束 2](#_Toc2040240239)

[3 架构设计 4](#_Toc1510147426)

[3.1 总体方案 4](#_Toc1338270268)

[3.2 架构说明 5](#_Toc710285280)

[3.2.1 架构图及说明 5](#_Toc2096613311)

[3.2.2 架构设计关键点 6](#_Toc258310603)

[3.2.3 高可用性设计 6](#_Toc359567983)

[3.2.4 高性能设计 6](#_Toc507179532)

[3.2.5 可扩展性设计 6](#_Toc1024032099)

[3.2.6 安全性设计 6](#_Toc651751081)

[3.2.7 其他设计 6](#_Toc866747371)

[4 部署方案 6](#_Toc1842790388)

# 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

1. 首字母缩写：
2. 缩略语：
3. 引用：
4. 软件架构概述：

*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*

# 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+Redis+Mybatis

1. 工具：前端小组为VSCode + figma，

后端小组为IDEA+VMWare+Navicat

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 |

1. 时间表：

|  |  |  |  |
| --- | --- | --- | --- |
| **任务名称** | **责任人** | **任务起止时间** | **输出产物** |
| UI界面设计 | 张韵、陈泽镐 | 2024/3/5-2024/3/13 | 完成界面设计图 |
| 前端环境搭建与首页框架搭建 | 王胜煜 | 2024/3/5-2024/3/13 | 前端环境的搭建 |
| 医院导览建模 | 陈泽镐 | 2024/3/12-2024/3/23 | 医院3d导览建模 |
| 登录注册功能 | 王胜煜/陈泽镐/张韵 | 2024/3/13-2024/3/14 | 登录注册页面 |
| 用户中心功能、用户管理功能 | 王胜煜/陈泽镐/张韵 | 2024/3/14-2024/3/15 | 用户中心页面、用户管理页面 |
| 病例管理功能 | 王胜煜/陈泽镐/张韵 | 2024/3/15-2024/3/16 | 病例管理页面 |
| 试题管理功能 | 王胜煜/陈泽镐/张韵 | 2024/3/16-2024/3/17 | 试题管理页面 |
| 考试管理功能 | 王胜煜/陈泽镐/张韵 | 2024/3/17-2024/3/18 | 考试管理页面 |
| 病例学习功能 | 王胜煜/陈泽镐/张韵 | 2024/3/18-2024/3/19 | 病例学习页面 |
| 试题测试功能 | 王胜煜/陈泽镐/张韵 | 2024/3/19-2024/3/20 | 试题测试页面 |
| 医院管理功能 | 王胜煜/陈泽镐/张韵 | 2024/3/20-2024/3/22 | 医院管理页面 |
| 医院导览功能 | 王胜煜/陈泽镐/张韵 | 2024/3/22-2024/3/23 | 医院导览页面 |
| 职能扮演功能 | 王胜煜/陈泽镐/张韵 | 2024/3/23-2024/3/24 | 职能扮演页面 |
| ai助教功能 | 王胜煜/陈泽镐/张韵 | 2024/3/24-2024/3/25 | ai助教页面 |
| 前端测试 | 王胜煜/陈泽镐/张韵 | 2024/3/25-2024/3/29 | 排除前端页面bug |
| 前后端对接 | 所有人 | 2024/3/29-2024/4-5 | 前后端完成对接，可正常使用 |
| **后端任务** | **责任人** | **任务起止时间** | **输出产物** |
| 数据库表设计 | 王盛阳 | 2024/3/5-2024/3/13 | 数据库表 |
| 后端环境搭建 | 潘煦森 | 2024/3/5-2024/3/13 | 后端环境的搭建 |
| 后端接口的简单实现 | 潘煦森，张寒，王盛阳 | 2024/3/13-2024/3/16 | 实现基本功能 |
| 后端接口测试 | 张寒 | 2024/3/17-2024/3/20 | 测试后端接口并提交bug，后端人员完成修改 |
| 登录注册 | 潘煦森 | 2024/3/13-2024/3/16 | 登录注册接口 |
| 用户管理 | 张寒、王盛阳 | 2024/3/15-2024/3/20 | 用户管理接口 |
| 医院导览科室管理 | 潘煦森、张寒 | 2024/3/17-2024/3/24 | 科室管理接口 |
| 病例管理、试题管理 | 王盛阳 | 2024/3/17-2024/3/21 | 病例管理和试题管理的接口 |
| 考试管理 | 潘煦森 | 2024/3/21-2024/3/24 | 考试功能的接口 |
| 职能扮演 | 张寒 | 2024/3/17-2024/3/20 | 职能扮演接口 |
| 智能助教开发 | 王盛阳、潘煦森 | 2024/3/25-2024/3/28 | 智能问答接口 |
| 数据上传管理 | 潘煦森、王盛阳、张寒 | 2024/3/28-2024/3/30 | 文件图片上传功能 |
| 后端测试 | 所有人 | 2024/3/30-2024/4-1 | 后端完成测试 |
| 前后端对接 | 所有人 | 2024/4/1-2024/4-5 | 前后端完成对接，可正常使用 |

*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, 包括性能 Performance、成本 Cost、时间 Time、可靠性 Reliability、安全性 Security、合规性 Compliance、技术性 Technology、兼容性 Compatibility。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。*

# 架构设计

## 总体方案

*提示：从整体上描述方案的结构，其核心内容就是架构图以及针对架构图的描述，包括模块或者子系统的职责描述、核心流程*

## 架构说明

### 架构图及说明

图3.2.1 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 层实现用户请求和响应并调用service层封装好的接口，service层负责实现业务逻辑并将dao层数据处理的构造方法封装成供前端使用的接口，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

### 高可用性设计

1.采用Redis与数据库的多级缓存机制，提高了数据处理的效率

*提示：支持系统各个功能满足可靠性的设计方案，即系统如何在其应提供服务时间段提供正确的服务*

### 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。

*提示：保证系统信息安全的设计方案。例如系统提供权限控制功能，权限控制包括两部分：身份识别和人员权限控制。身份识别如何设计，人员权限控制如何设,*

### 其他设计

指定开发语言：JavaScript、CSS、html，Java、Sql

*提示：上述以外的其他设计考虑点，例如指定开发语言、符合公司的某些标准等*

# 部署方案

图4.1 部署视图

*提示：部署视图显示的是系统的实际部署情况，包括硬件要求、服务器部署方式、组网方式，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。*