牙科医院管理系统

软件架构文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 27/4/2022 | <1.0> | 本文档将从架构方面对牙科医院管理系统进行综合概述 | 宋金骋、谷金龙、李昱翰、时宪 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 逻辑视图 4

3. 进程视图 4

4. 部署视图 4

5. 实现视图 5

6. 技术视图 5

7. 核心算法设计（可选） 5

8. 质量属性的设计 5

软件架构文档

# 简介

## 目的

本文档将从架构方面对牙科医院管理系统进行综合概述，阐述了系统的总体框架，包括逻辑视图、进程视图、部署视图、实现视图、技术视图、核心算法设计和质量属性的设计，说明了体系结构采取的设计策略和技术。为今后的编码设计提供了参考模板并提高效率，为客户提供了指引与说明，方便客户上手。

## 参考资料

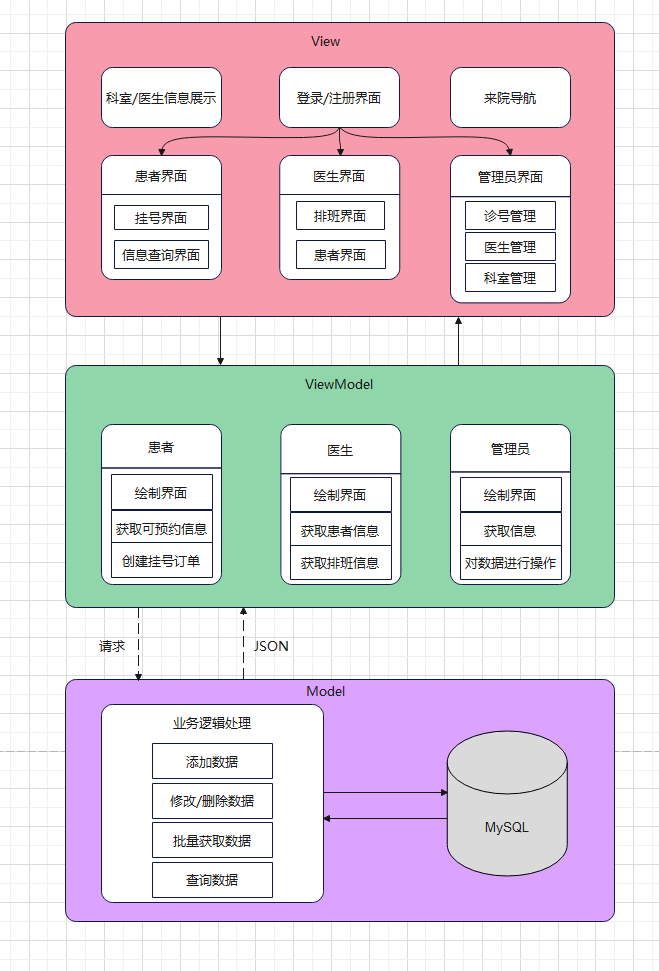
MVVM架构:[MVVM\_百度百科 (baidu.com)](https://baike.baidu.com/item/MVVM/96310)

Spring Boot官网: [Spring Boot](https://spring.io/projects/spring-boot)

UmiJS官网: [介绍 (umijs.org)](https://umijs.org/zh-CN/docs)

IntelliJ IDEA官网: [IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains](https://www.jetbrains.com/idea/)

# 逻辑视图



View：

展示ViewModel层的数据和状态

ViewModel：

暴露View需要的层数据，处理View层的具体业务逻辑，向Model层发送请求并接受JSON文件

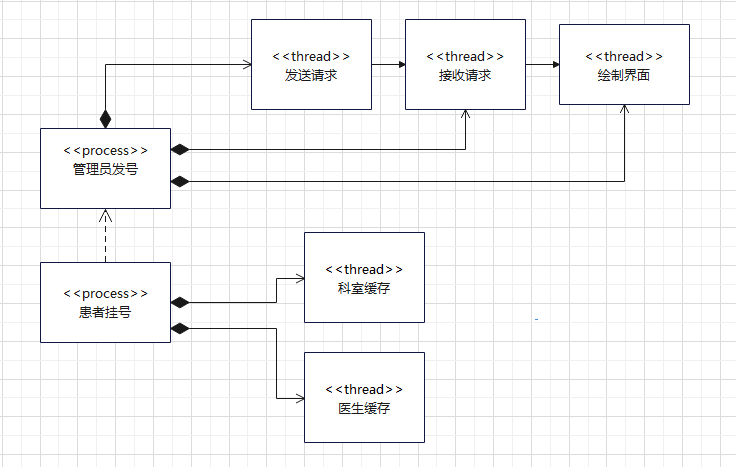
Model：

数据的存储、操作，处理ViewModel层的请求

优点：

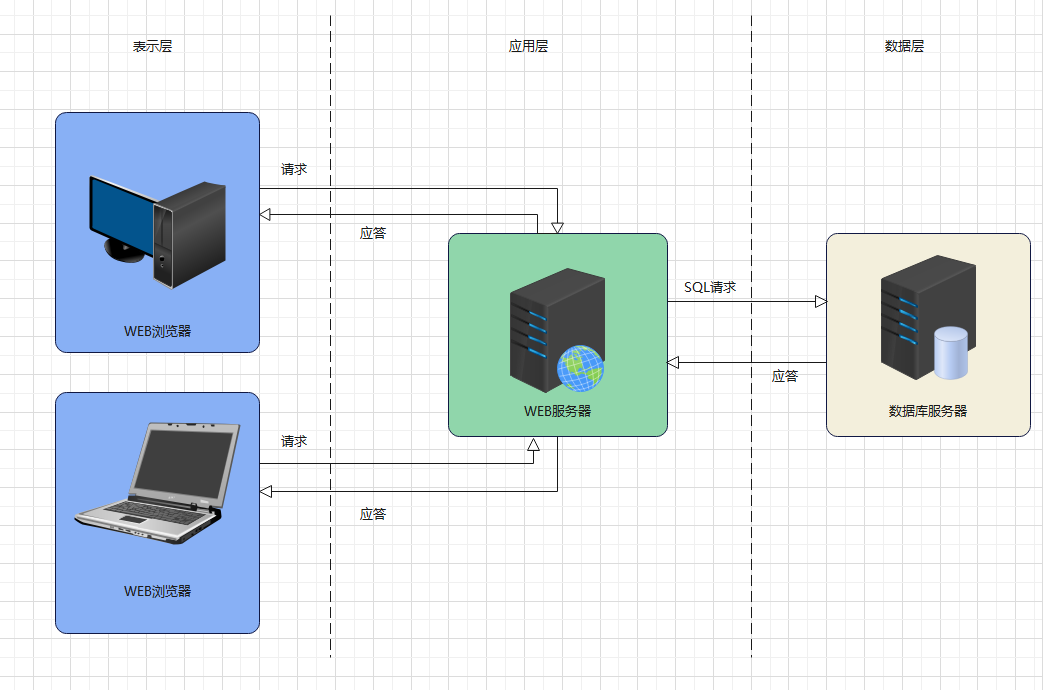
采用MVVM将视图UI和业务逻辑分离，有利于提高效率、降低耦合、提高复用性

# 进程视图



挂号功能的进程视图，患者挂号的进程依赖于管理员发号的进程，管理员发号进程由发送请求、接受请求、绘制界面三个线程组成。

# 部署视图

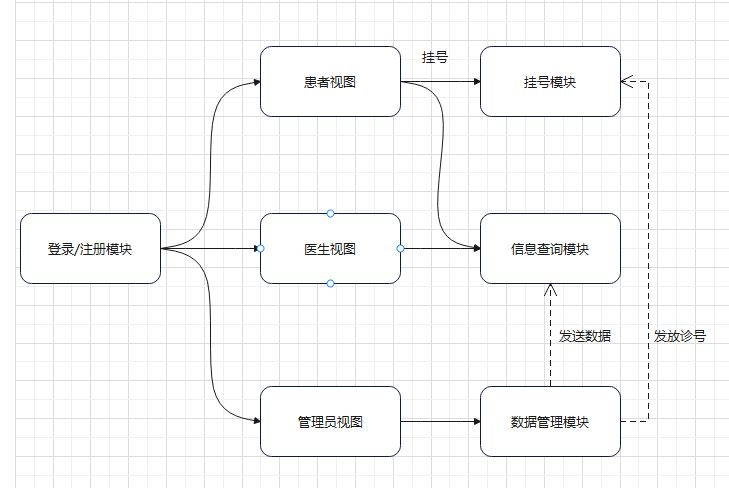


本牙科医院管理系统的部署视图采用B/S三层架构，分为表示层、应用层、数据层三层。工作流程：表示层为客户端，运行在用户的PC浏览器上；浏览器发送http请求到服务器，服务器端的web服务器接收并处理请求，向数据库服务器请求数据；数据库服务器将web服务器请求的数据返回，再由web服务器将数据返回给浏览器；浏览器解释执行html文件，呈现界面，显示信息并供用户交互。

优点：采用B/S架构可以省去安装过程，且适应性强、业务拓展方便、维护较简单。

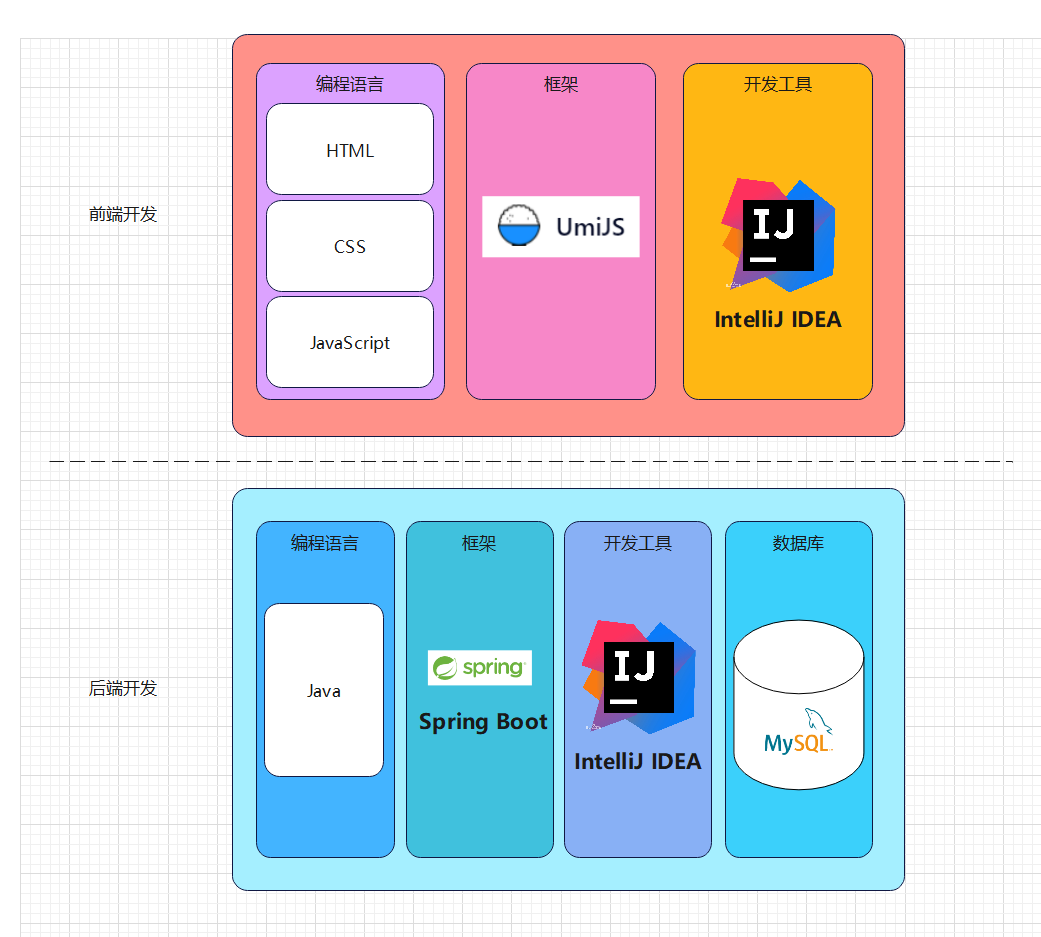
缺点：服务器压力大、性能受网速影响、兼容问题较多。

# 实现视图



挂号功能的实现如图，用户通过登录后进入到不同的视图，患者可以进行信息查询和挂号，医生可以查询信息，管理员通过数据管理模块对信息查询模块发送数据或者对挂号模块发放诊号

# 技术视图

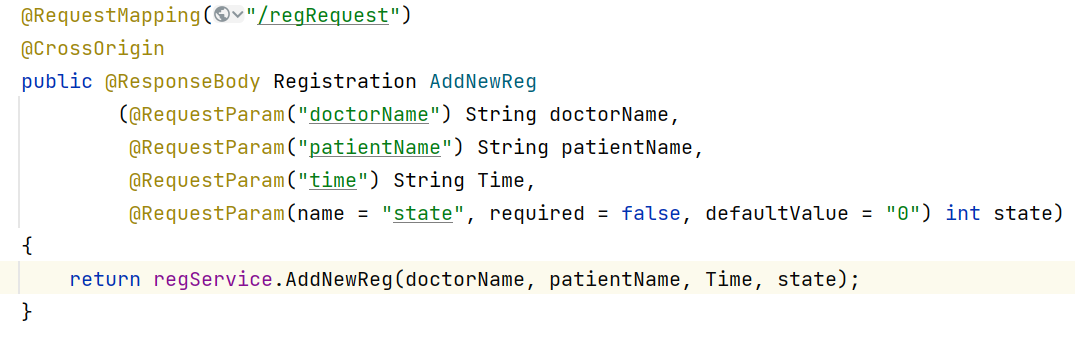


本项目前端选择UmiJS作为框架进行开发，开发工具为IntelliJ IDEA，使用编程语言为HTML、CSS、JavaScript

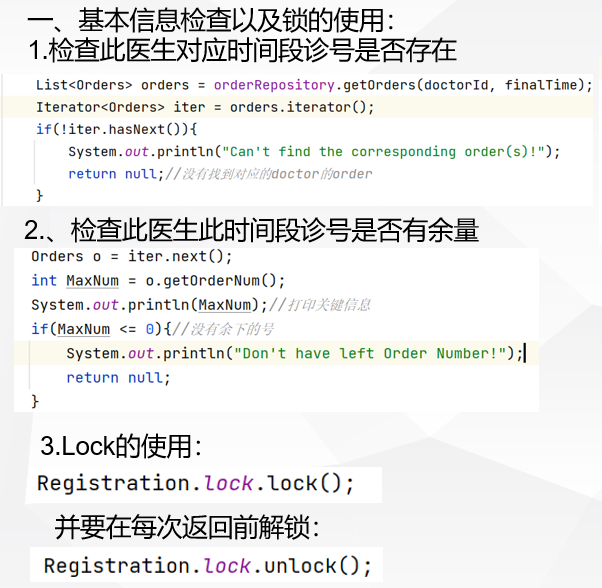
本项目后端选择Spring Boot作为框架进行开发，开发工具为IntelliJ IDEA，使用编程语言为JAVA，数据库查询语言SQL，数据库选择MySQL

# 核心算法设计

1.后端接收前端请求接口



2.后端具体实现逻辑





3.后端接口测试

# 质量属性的设计

该系统使用MVVM的逻辑架构及B/S三层物理架构，有利于降低耦合，提高适应性和复用性

项目开发选择较为成熟的框架Umi和Spring Boot，简化配置和部署，节约学习成本

开发工具使用IntelliJ IDEA，易于上手