**毕设注意**

**计算机毕设基本没有学校对代码进行查重，只会对论文进行查询**，只要把论文写好就可以了。

写论文最简单，最直接的方法，就是去找类似的论文，最新的论文照着写。

加下自己的创新和优化。

# 系统优化

## 系统分析

整体来看是一个偏向百科查询类的系统。

**数据是怎么来的呢？**

通过mybatis-plus简化了sql编写，从数据库中查询。

### 角色介绍

系统共设计了三个角色：游客、用户、管理员。

1. 游客：尚未进行注册和登录。具备登录注册、疾病、药品的搜索、查询详情等权限。
2. 用户：除了游客的功能权限外，还具备登录、个人资料的修改、登录密码修改、意见反馈、智能医生咨询等权限。
3. 管理员：除了用户的功能权限外，还具备疾病管理、药品管理、反馈管理等权限。

### 系统遇到的问题：

1. Ai会回复医疗之外的问题。 【还需优化】

解决方案有几种:

1. 关键字过滤（低复杂度）。设置一组与医疗相关的关键词列表，用户输入的问题中必须包含这些关键词，系统才会响应。
2. 意图分类模型（中等复杂度）。需要一定的 NLP 和机器学习知识
3. 主题识别和语义分析（较高复杂度）。利用 NLP 技术进行语义分析，识别用户输入的主题和语境，判断是否与医疗领域相关。
4. 基于知识库的限制（中等复杂度）。

本项目采用第一种。不过还可以尝试另外三种。

1. UI优化问题，游客点击【智能医生】会重定向本页。
2. 图片问题，上传图片代码在哪？ -- 图片存在阿里云仓库，需重新上传图片
3. 药品和疾病的数据可以爬虫吗。

### 依赖

依赖涉及广泛的功能，从 Spring Boot 基础框架、数据库操作，到邮件、短信、AI 调用、文件处理等，几乎涵盖了完整的应用开发需求。每个依赖都在各自领域提供了非常成熟的功能，帮助实现智慧医疗系统中的各项业务需求。

### 优化建议

对频繁查询的数据（如疾病分类）加入缓存，减少数据库压力。

**邮箱验证码认证**：

* 用户输入邮箱后，系统生成随机验证码并通过邮件发送。
* 验证码存储在 Session，有效期内（如5分钟）可验证通过。

**加密存储**：

* 密码在存储前通过加密（如 MD5、SHA-256）处理，防止明文泄露。
* 可结合盐值（Salt）增强密码复杂性。

加个knf4j接口文档

# 页面

**邮件怎么配置？**

· MAIL\_USERNAME：邮件服务的用户名（如QQ邮箱）。

· MAIL\_PASSWORD：邮件服务的密码（如QQ邮箱授权码）。

· host、port、properties：邮件服务的SMTP设置，这里配置了ssl（安全连接）和starttls（加密连接）等。

· valid：验证码有效期，设置为5分钟。

· title和template：邮件的标题和内容模板。

**通义千问怎么配置？**

ai-key：阿里云通义千问的API密钥。

**对象存储怎么配置？**

· bucket-name：阿里云OSS的存储桶名称。

· end-point：OSS的地域节点（如广州区域）。

· access-key、access-secret：用于访问阿里云OSS的凭证

**@Value("${spring.mail.template}") private String template; 值是怎么注入的？**

@Value 注解中的 ${} 括起来的内容是 Spring 占位符表达式，表示从配置文件中获取对应的值。

## 首页-权限控制器

点击【智能医生】，没登录就返回首页。

点击【所有反馈】，没登录就返回首页。

点击【我的资料】，没登录就返回首页。

点击【添加疾病】，没登录就返回首页。

点击【添加药品】，没登录就返回首页。

点击【疾病管理】，没登录就返回首页。

点击【药品管理】，没登录就返回首页。



## 注册

### 发送邮箱验证码怎么实现？

· **生成验证码**：利用RandomUtil.randomNumbers(6)生成6位数字验证码。

· **发送邮件**：使用JavaMailSenderImpl和MimeMessageHelper构建并发送邮件。

· **存储验证码**：将验证码存入session，以便后续验证。

· **验证验证码**：用户提交验证码时，从session获取并检查验证码是否正确及是否过期。

## 登录

### 登录怎么实现的？

使用登录控制器接受login请求。流程分支有登录成功、账户尚未注册、密码错误。

### 权限控制怎么做的？

登录成功会将用户对象保存在session中。

修改成登录拦截器了。

**1、游客与用户、管理员的权限在于智能医生模块的区分。**

修改前：使用系统跳转控制器SystemController，对智能医生、所有反馈、我的资料、添加疾病、添加药品、疾病管理、药品管理对未登录用户限制访问。

修改后：系统跳转控制器SystemController，将判断登录用户逻辑去掉了。改为控制拦截器中进行统一判断。

**2、用户和管理员的权限进行了区分。**

角色状态，1代表管理员，0普通用户

在前端进行限制，条件判断登录用户的角色状态是否是管理员，是则显示模块li标签。

## 公共部分 common-bar.html





### 搜索

### 提交反馈

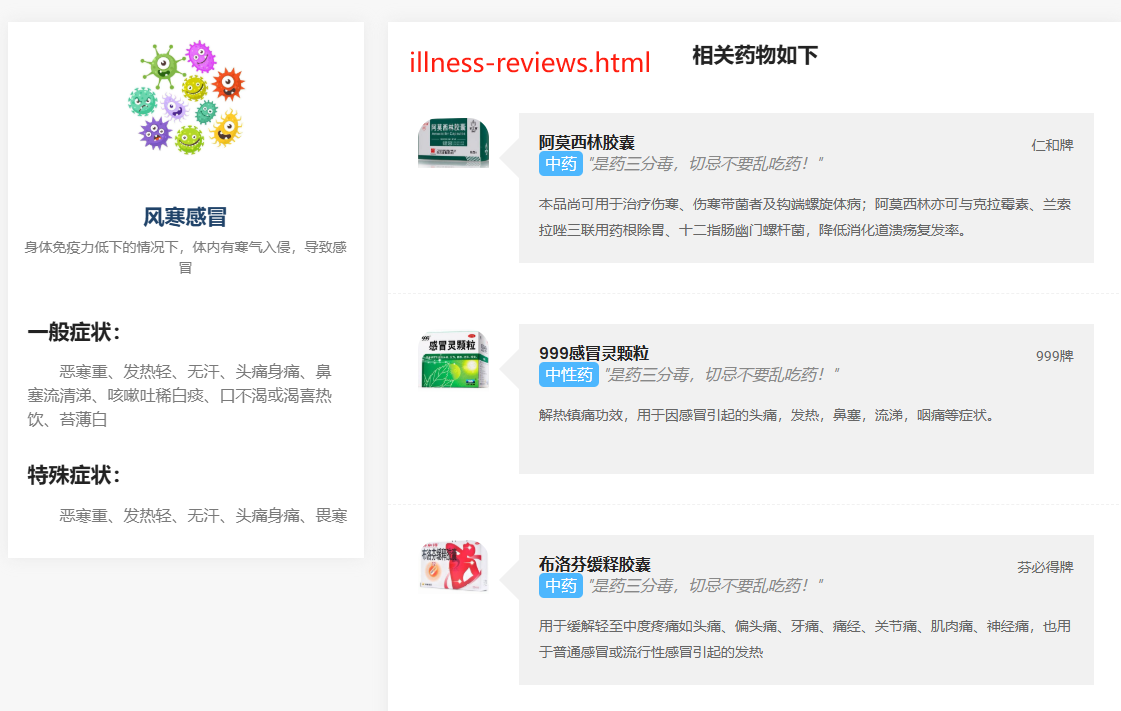
### 文件上传

## 疾病管理

### 查看疾病

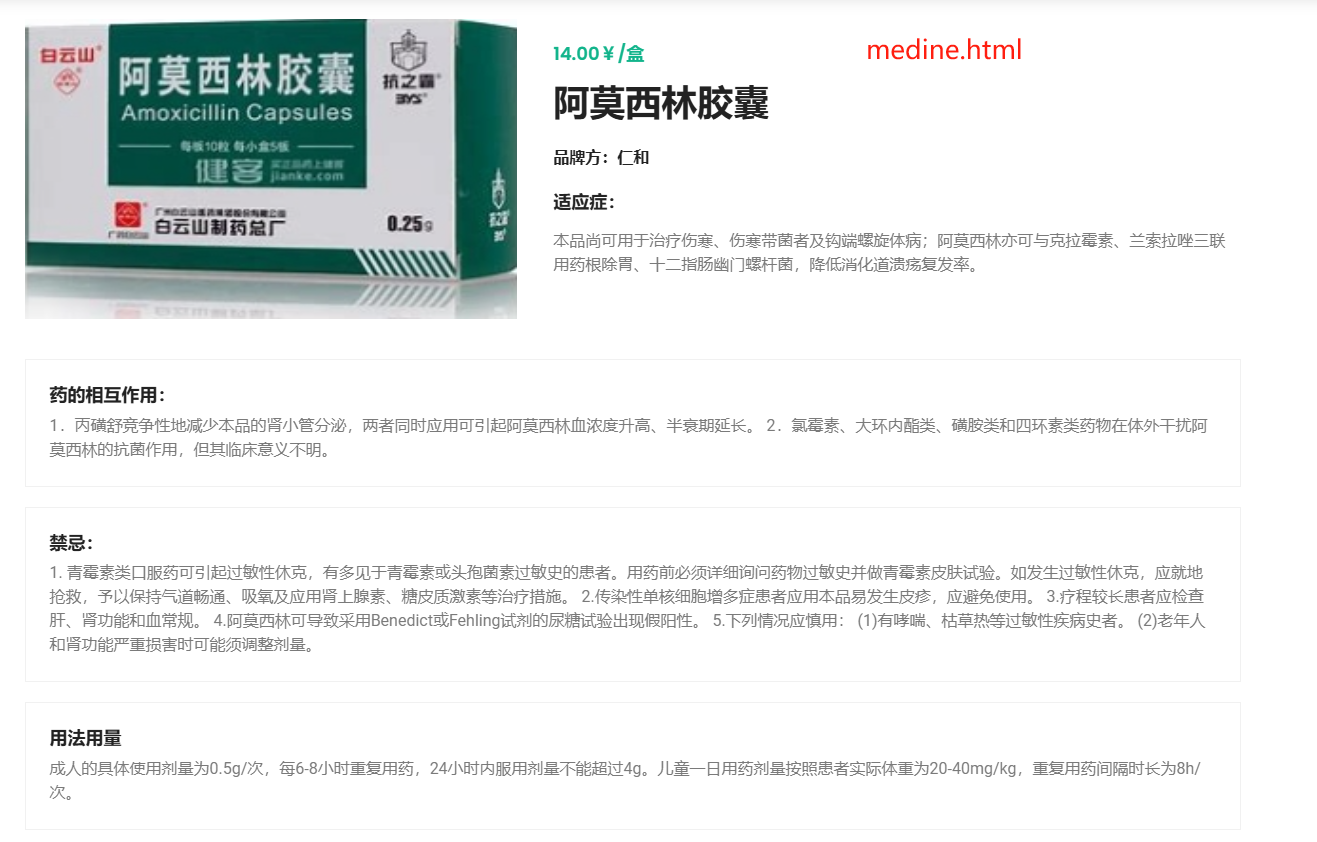
**查看疾病怎么进行跳转？**

1. “所有疾病”中点击【查看疾病】
2. 生成url： findIllnessOne/id
3. 在controller层 findIllnessOne 方法中查询相关疾病下的药信息
4. 返回疾病页

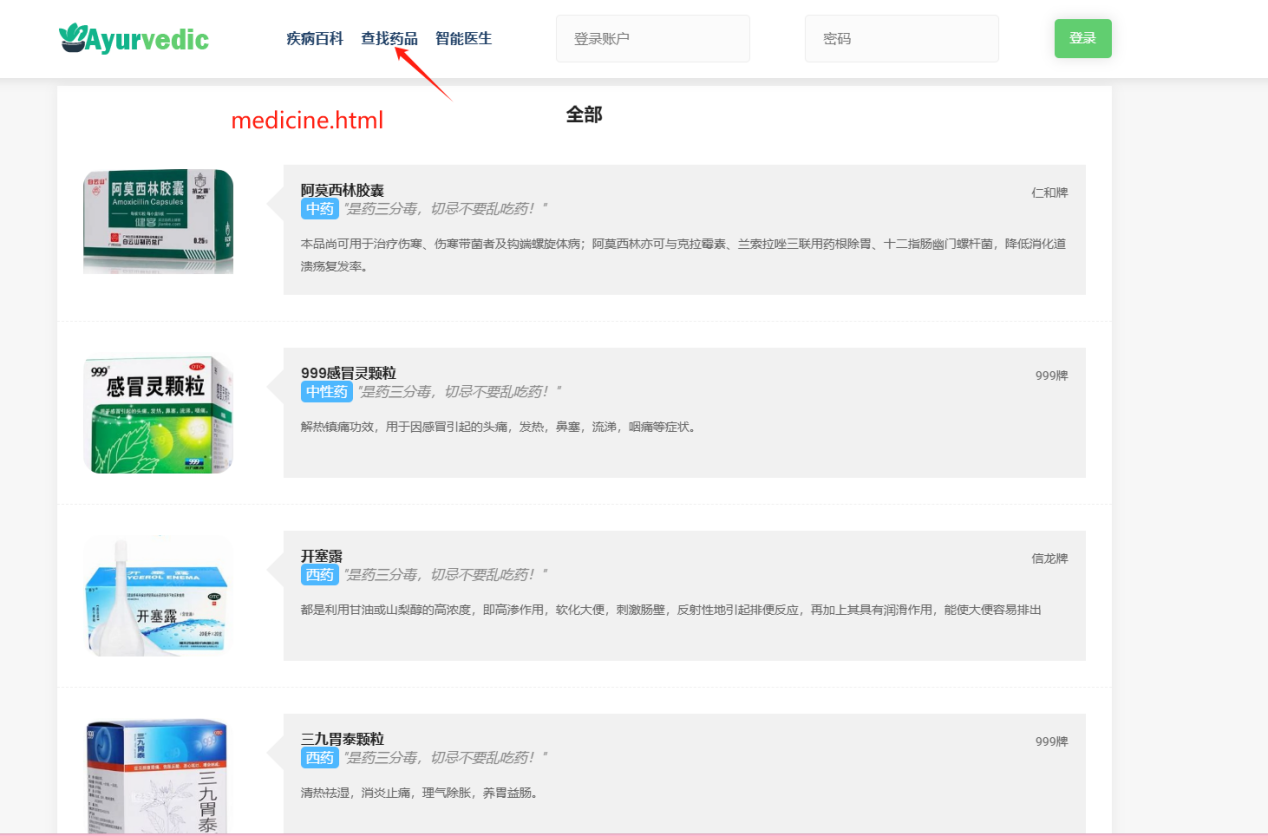


### 从疾病详情中查看药





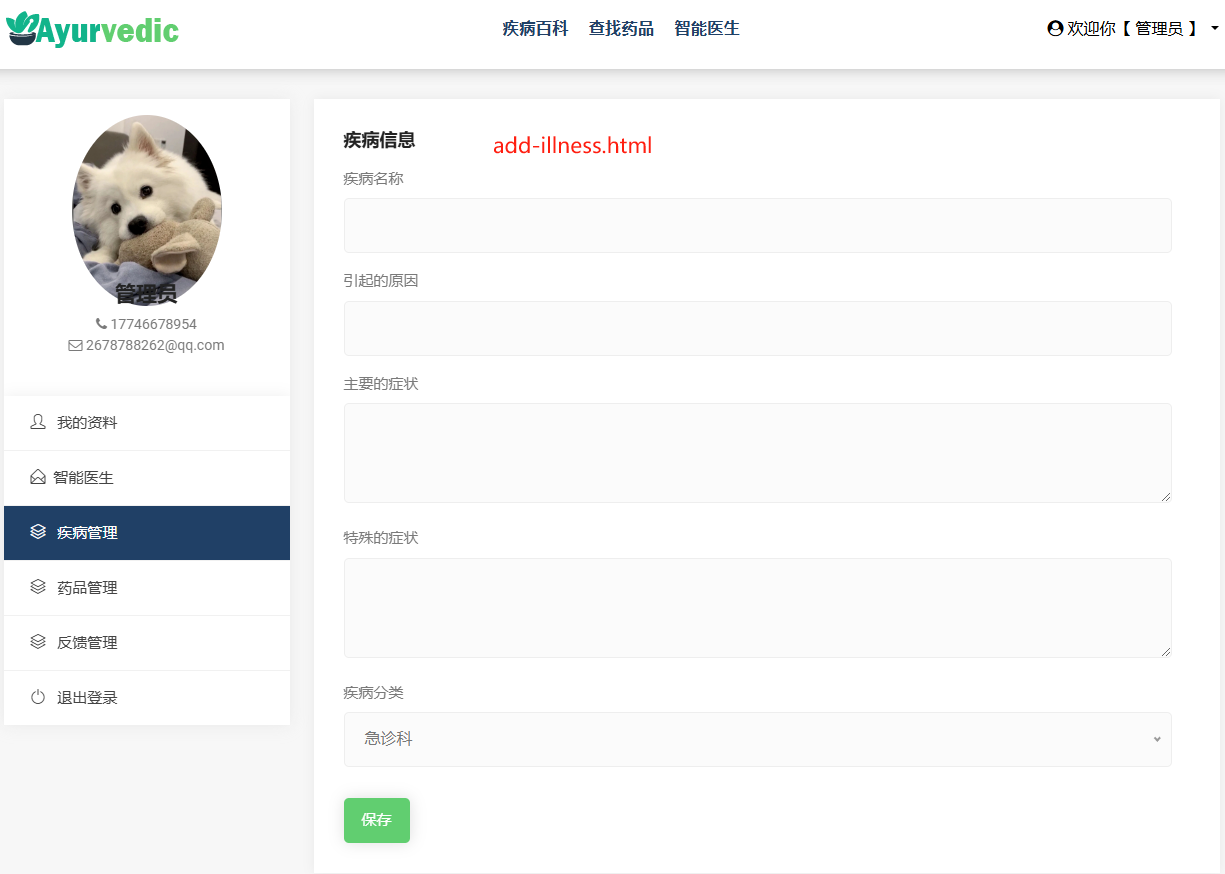
### 查找药品



### 所有疾病



### 新增疾病



**新增疾病的流程是怎么样的？**

1. 疾病管理>全部疾病，点击【新增疾病】
2. 跳转systemController的保存方法，填入信息，获取疾病种类。
3. 跳转add-illness页面，输入信息，点击【保存】
4. 调用custom.js文件的保存疾病方法，去带参数访问后端url，疾病controller及basecontroller的save，然后调用BaseService继承IService的save方法，根据多态，动态性绑定，其实是调用IllnessService.save。启动时： 当 Spring 初始化 IllnessController Bean 时，会解析其父类 BaseController<Illness>，并找到合适的 BaseService 实现。依赖注入时： 当 Spring 执行 @Autowired 注入时，发现 service 需要注入，Spring 从容器中获取具体的 IllnessService 实例
5. illness.save()，数据有id说明数据库有数据是保存则执行插入，否则执行修改，最后返回更新的本条数据。

**为什么可以从controller返回“页面名称”，就跳转到前端页面？ 这是什么语法？**

视图解析器会根据该逻辑视图名称找到对应的物理视图文件。视图解析器根据配置的前后缀（例如，/WEB-INF/views/ 和 .jsp）来定位实际的视图文件路径，最终渲染该视图。视图解析器（ViewResolver） 的工作原理

### 修改疾病

流程：

1. 不管新增还是编辑，都会跳转Controtller，填入信息，根据疾病id获取疾病详情，再获取所有疾病种类，用于下拉框使用，返回add-illness.html
2. 填写信息，点击【保存】，调用保存方法，发送ajax请求
3. 调用编辑。
4. 返回数据

### 删除疾病

删除流程：

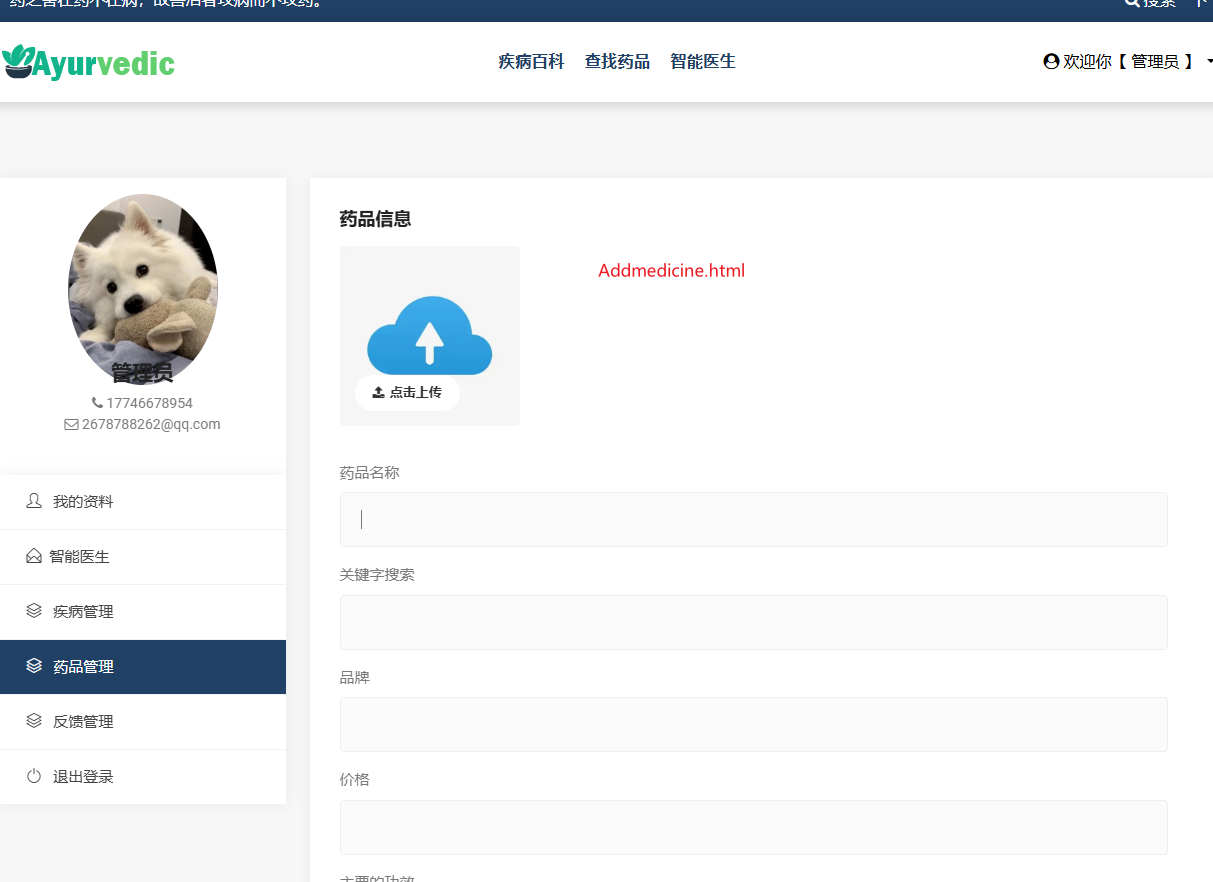
1. 点击【删除】，调用deleteIllness，发送ajax请求
2. controller进行接受，调用IllnessService删除
3. controller根据删除的行数判断删除成功，删除失败并返回数据给前端

## 药品管理

### 所有药品

### 修改

### 新增药品



### 删除

## 反馈管理

### 所有反馈



### 删除反馈

## 智能医生

流程：

1. 用户输入消息并点击“发送”按钮 ，send
2. 发送 AJAX 请求到后端， /message/query 该函数获取输入框中的文本，若为空则不发送请求。发送请求时，将消息作为 content 参数发送给后端 /message/query 接口。
3. MessageController接收到请求后，调用 ApiService.query() 方法进行处理。
4. ApiService 会检查消息是否包含医疗相关的关键词。
5. 如果是医疗问题，调用 AIGC 模型生成回复；如果不是，返回一个提示消息。

