

基于 Java Web 的医疗咨询平台的设计与实现

韩定坤

上海市奉贤区精神卫生中心，上海 201400

摘要：社会的发展蒸蒸日上，传统医疗已经无法满足人们日益增加的医疗健康的需求，网络技术的不断变迁和迭代，医疗行业面临新的机遇，新的挑战。目前的医疗咨询方式受限于时间、空间等因素，无法满足人们的需求，因此需要一种更便捷、快速的咨询方式。而基于 JavaWeb 的医疗咨询平台正是针对这一需求而开发出来的。第一、提高医患沟通效率，传统的医疗咨询方式需要患者亲自前往医院或门诊，而基于 JavaWeb 的医疗咨询平台可以通过网络直接与医生进行咨询，大大提高了医患沟通的效率。第二、方便患者就医，基于 JavaWeb 的医疗咨询平台可以为患者提供更加便捷、快速的看病方式，同时也为医生提供了更多的就诊渠道，更好地服务于患者。第三、促进医疗信息化建设，基于 JavaWeb 的医疗咨询平台可以促进医疗信息化建设，推动医疗行业的数字化转型，提高医院的信息化水平。本论文基于 Java Web 技术，设计并实现了一个医疗咨询平台，旨在提高医疗服务的效率和便捷性。经过本小组的需求和系统的分析，采用 Spring Boot 框架和 MySQL 数据库，实现了医生管理、在线咨询、预约挂号等功能。系统经过详细的测试，包括单元测试、集成测试、性能测试和安全测试，保证了系统的稳定性和安全性。最终实现了一个功能完善、用户友好的医疗咨询平台，为用户提供了高效便捷的医疗服务体验。

关键词：Java Web 医疗咨询平台 Spring Boot

第 1 章 绪论

1.1 研究背景

随着互联网技术的快速发展，传统医疗已经无法满足人们日益增加的医疗健康的需求。目前的医疗咨询方式现存很多隐患，如资料不公开、时间成本高等，难以满足现代社会对医疗服务的高效性和便捷性需求。因此，设计和实现基于 Java Web 的医疗咨询平台对提升医疗服务水平具有积极意义。

1.2 项目研究的意义与内容

1.2.1 项目研究的意义

基于 Java Web 的医疗咨询平台的设计与实现具有重要意义：医疗咨询平台为病患和医师都提供了方便，病患可以方便地获取问题答案、进行在线提问，医生可以利用闲暇为病患提供专业知识，化简医患沟通流程，提升医疗服务体验。化简医患沟通流程：患者可简单的与信任医生进行交流与对话，加强医患之间的关系巩固。提高医疗服务水准。

1.2.2 项目研究的内容

本项目的研究内容主要包括以下：

需求分析：搞清楚医疗咨询平台功能与非功能需求，确定医疗咨询平台系统的最终目的与追求。

系统设计：设计系统的架构和模块，包括设计模式、层次结构、数据库设计等。系统实现：基于 Spring Boot 技术实现系统的核心功能，包括用户注册、登录、医生管理、在线咨询、预约挂号等。

总结展望：探索医疗咨询平台的发展前途与机遇，为此项目谋求更进一步的发展。通过以上内容的研究和实践，可以建立一个功能完善、用户友好的基于 Java Web 的医疗咨询平台，提升医疗服务水平，满足用户对便捷、高效医疗服务的需求。

第 2 章 系统分析

2.1 功能性需求

2.1.1 总体模块需求分析

医疗咨询平台分为病人端和医生端两个客户端，功能包括登录注册、问题提出、回答问题、查看问题与评论等模块。病人可以提出医疗健康相关问题，包括症状描述、病史等，可以选择是否指定医生回答问题。病人可以浏览医师的专业答案，若有疑问可追问使自己疑虑消除。病人还可以查看其他病人提出的问题和医生的回答，并对问题和回答进行评论和点赞。

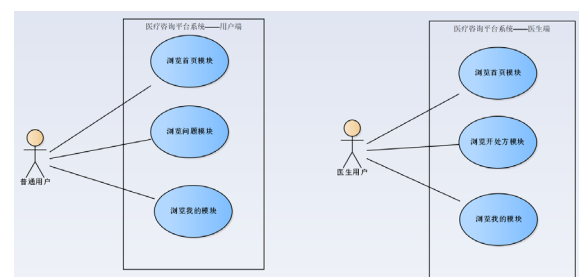


图 2-1 总体需求的用例图

2.1.2 单元功能模块需求分析

1. 病人提出问题模块

提问题模块是软件重要模块之一，含有对问题的提出、修改、查看、删除等功能。病人也可向指定医生提出问题也可在问题广场提出问题。软件为问题进行分类，使问题的查找与浏览更加有条理。

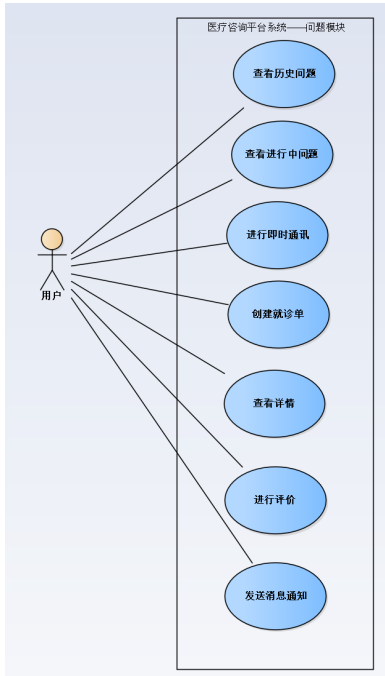


图 2-2 提问题模块的用例图

2. 医生回答问题模块

医生回答问题模块是医疗咨询平台中非常重要的一部分，主要包括指定问题和非指定问题两种情况：

指定问题：当患者在咨询时提出具体问题，医生需要对这些问题进行回答。

非指定问题：在一些情况下，患者可能并没有明确的问题，而是希望医生就某个主题或病情进行讨论和解释。

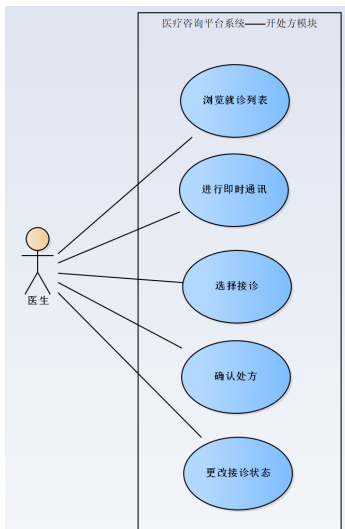


图 2-3 回答问题模块的用例图

3. 查看问题与评论模块

查看问题与评论模块是医疗咨询平台中患者和医生交流的重要环节，主要包括以下功能：

查看问题：患者可以在平台上发布问题，表达自己的疑问和症状。医生可浏览病患的描述，了解患者的疑问和症状。

问题分类与标签：平台可以根据问题的内容和主题进行分类和标签，方便患者和医生查找相关问题。分类和标签可以帮助提高问题的曝光度，使更多的医生和患者参与讨论。

2.2 项目的可行性分析

技术可行性：技术选择：Java Web 和 Spring Boot 是当前流行的技术栈，具备强大的开发和部署能力，能够满足医疗咨询平台的需求。技术支持：Java 和 Spring Boot 有庞大的开发者社区和丰富的文档资料，能够提供充足的技术支持。

经济可行性：成本预算：医疗咨询平台的开发和运营成本包括硬件设备、软件开发、人力资源等方面，需要进行详细的成本预算。收益预期：医疗咨询平台可以通过广告投放、服务费用等方式获取收益，需要对收益进行合理预期。

合规性：平台的运营和服务需符合医疗行业的规范和要求，保证服务的合法性和合规性。

社会可行性：市场需求：医疗咨询服务在社会上有较大的需求，尤其是在疫情期间，线上医疗咨询更受欢迎。

社会影响：医疗咨询平台可以提高医疗资源利用效率，方便患者就医，具有积极的社会影响。

第 3 章 详细设计

3.1 软件设计框架

MVC(Model-View-Controller)是一种软件设计模式，用于将应用程序分为三个核心部分：模型 (Model)、视图 (View) 和控制器 (Controller)。每个部分负责不同的功能，协同工作以实现应用程序的分层和解耦。

模型 (Model)：模型表示应用程序的数据和业务逻辑。它负责处理数据的存储、检索、更新和删除，并定义了应用程序的行为规则。

视图 (View)：视图负责显示模型中的数据以及与用户的交互界面。它将模型的数据转换为用户可以理解的形式，并将用户的输入转发给控制器。

控制器 (Controller)：控制器负责接收用户的输入，并根据输入更新模型或选择合适的视图。它充当了模型和视图之间的中介，协调应用程序的流程。

图 3-4 系统 ER 图

3.4.2 数据库单元设计

1. 用户表 (user)：此表保存着病患的个人信息，使每个病患有独特的标识。

字段名	数据类型	注释
ABC user_id	char(20)	用户id
ABC user_name	char(20)	用户昵称
ABC user_password	char(20)	用户密码
ABC user_img	char(20)	用户头像
ABC user_phone	char(20)	用户电话

图 3-5 用户表图

2. 医生表 (doctor)：此表包含了医生的个人信息及管理，包括医师证的认证，使医师信息更加全面有说服力。

字段名	数据类型	注释
ABC doctor_id	varchar(64)	医生id
ABC doctor_name	varchar(64)	医生名字
ABC sex	varchar(16)	性别
ABC age	varchar(16)	年龄
ABC doctor_certificate	varchar(32)	资格证
ABC doctor_introduction	varchar(1024)	个人介绍
ABC doctor_score	varchar(16)	得分
ABC doctor_frequency	varchar(32)	回答次数
ABC doctor_password	varchar(32)	密码
ABC doctor_phone	varchar(16)	电话
ABC doctor_img	varchar(256)	头像

图 3-6 医生表图

3. 问题表 (question)：

该表存储了病患提出的病症问题相关信息。存储着谁回答这个问题或者谁回答了这个问题和问题的全部描述。

字段名	数据类型	注释
ABC question_id	char(20)	问题的id
ABC user_id	char(20)	用户的id
ABC doctor_id	char(20)	医生的id
ABC question_introduction	char(60)	问题介绍

图 3-7 问题表图

4. 答案表 (answer)：此表包含了医生和病患的各种问题和答案。

字段名	数据类型	注释
ABC answer_id	char(60)	回答id
ABC user_id	char(20)	用户id
ABC doctor_id	char(20)	医生id
ABC question_id	char(60)	问题id
ABC answer_all	char(60)	问题回答
ABC comment	char(60)	评论
ABC probe	char(60)	追问

图 3-8 答案表图

第 4 章 系统实现

4.1 用户模块

对于医疗咨询平台的用户模块，分为医生端和患者端。主要包括用户登录和管理本人信息等功能。用户登录后可浏览问题广场和本人问题，进行查看问题详情并进行提出问题指定医生回答、修改问题、删除问题等操作。将 Spring Boot 与 JPA 相结合进行数据的增删改查。

4.1.1 软件首页面

用户未登录会进入首页面，之后进行登录选择患者或医生端。

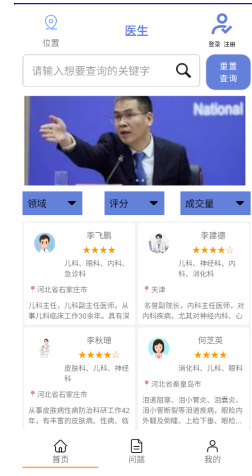


图 4-1 首页面

```
public class CookServiceImpl implements HomeCookService {
    2 usages
    @Autowired
    private HomeCookRepository homeCookRepository;

    1 usage xdxu108 *
    @Override
    public List<Doctor> findDoctors(Doctor doctor) {
        Specification<Doctor> specification = (root, criteriaQuery, criteriaBuilder) -> {
            List<Predicate> predicates = new ArrayList<>();
            return criteriaBuilder.and(predicates.toArray(new Predicate[predicates.size()]));
        };
        Sort sort = Sort.By(Sort.Order.desc( property: "cookScore"));
        List<Doctor> doctorList = homeCookRepository.findAll(specification, sort);
        return doctorList;
    }

    1 usage xdxu108 *
    @Override
    public Doctor findCookById(String cookId) {
        Optional<Doctor> optional = homeCookRepository.findById(cookId);
        return optional.get();
    }
}
```

图 4-1-1 首页面代码

4.1.2 患者端 / 医生端页面

当选择患者 / 医生端进入时，会进入相应界面：



图 4-2 医生端

用户端页面与首页面相同。同样进行增删改查。

```

@Override
public List<Orders> findOrdersById(String userId) {
    Specification<Doctor> specification = (root, criteriaQuery, criteriaBuilder) -> {
        List<Predicate> predicates = new ArrayList<>();
        if (!StringUtil.isEmpty(userId)) {
            predicates.add(criteriaBuilder.equal(root.get("userId"), userId));
        }
        return criteriaBuilder.and(predicates.toArray(new Predicate[predicates.size()]));
    };
    List<Orders> orderList = orderRepository.findAll(specification);
    return orderList;
}

1 usage  xiaodu108
@Override
public String updateOrderStatusById(String orderId) {
    Orders order = orderRepository.findById(orderId).get();
    order.setOrderStatus("已完成");
    orderRepository.save(order);
    return "success";
}

```

图 4-2-2 用户查看诊单

4.1.3 患者 / 医生自我管理

患者 / 医生可进入我的的界面来完善个人信息



图 4-3 医生我的界面



图 4-4 患者我的界面

都用的是查找更新自己的信息大同小异，只粘贴一个界面的代码了。

```

public User findUserById(String userId) {
    Optional<User> byId = mineRepository.findById(userId);
    User user = byId.get();
    return user;
}

1 usage  wqg2002
@Override
public User updateUser(User user) {
    Optional<User> optional = mineRepository.findById(user.getId());
    if (optional.isPresent()){
        User userInfo = optional.get();
        if (user.getUserName() == null){
            user.setUserName(userInfo.getUserName());
        }
        if (user.getUserPassword() == null){
            user.setUserPassword(userInfo.getUserPassword());
        }
        if (user.getUserPhone() == null){
            user.setUserPhone(userInfo.getUserPhone());
        }
        if (user.getUserLocation() == null){
            user.setUserLocation(userInfo.getUserLocation());
        }
        if (user.getUserImg() == null){
            user.setUserImg(userInfo.getUserImg());
        }
        User update = mineRepository.saveAndFlush(user);
        return update;
    }else{
        User create = createUser(user);
        return create;
    }
}

```

图 4-4-1 我的界面代码

4.2 病人提出问题模块

患者进入病人端后可提出自己的问题，包括自身状况、病理状况、年龄、性别等，问题可指定医生回答也可不指定投入问题广场等待回答。



图 4-5 提出问题界面

```

@Override
public List<Orders> findAllOrder() {
    List<Orders> orders_1 = orderRepository.findAll();
    return orders_1;
}

1 usage  xialone +1 *
@Override
public Orders findorderById(String doctor_id) {
    Optional<Orders> optional = orderRepository.findById(doctor_id);
    Set<Orders> ordersSet = new HashSet<>();
    Orders orders = optional.get();
    ordersSet.add(orders);
    return orders;
}

1 usage  xialone
@Override
public Orders updateStatus(String order_id,String status) {
    Optional<Orders> optional=orderRepository.findById(order_id);
    Orders orders=optional.get();
    orders.setOrderStatus(status);
    Orders save = orderRepository.save(orders);
    return orders;
}

```

图 4-5-1 诊单页面代码

点击聊天可与医生进行会诊交流

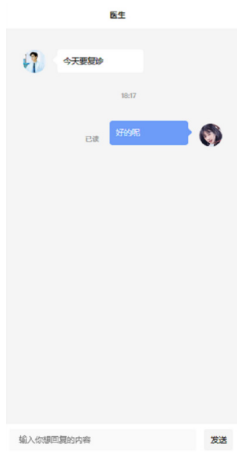


图 4-6 交流页面

交流是搭入了融云框架，后端代码如下：

```
public String getToken(@RequestParam(required = true) String userID) {
    StringBuffer req = new StringBuffer();
    String url = "http://api.cn.ronghub.com/user/getToken.json";
    String App_Key = "18y99t12222"; //开发平台分配的 App_Key;
    String App_Secret = "72124970797"; //开发平台分配的 App_Secret;
    String Timestamp = String.valueOf(System.currentTimeMillis() / 1000); //时间戳, 从 1970 年 1 月 1 日 0 时 0 分 0 秒开始到现在的秒数;
    String Nonce = String.valueOf(Math.floor(Math.random() * (99999 - 10000) + 10000)); //随机数, 五位数随机;

    String Signature = sha1(App_Key + App_Secret + Nonce + Timestamp); //签名生成;
    //Logger.i(signature);
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(url);
    httpPost.setHeader("App_Key", App_Key);
    httpPost.setHeader("App_Secret", App_Secret);
    httpPost.setHeader("Timestamp", Timestamp);
    httpPost.setHeader("Nonce", Nonce);
    httpPost.setHeader("Signature", Signature);
    httpPost.setHeader("Content-Type", "application/x-www-form-urlencoded");
    ListNameValuePair nameValuePair = new ArrayList<>(1);
    // nameValuePair.add(new BasicNameValuePair("name", userName)); //姓名 (例如张三, 必须是中文, 且长度在 2-10 之间);
    nameValuePair.add(new BasicNameValuePair("name", userID)); // 用户id (根据自己返回, 自己生成一个唯一id, UUID即可)
    // nameValuePair.add(new BasicNameValuePair("portraitUrl", userHead)); //头像(存储在服务器的路径)
    HttpResponse httpResponse = null;
    try {
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair, "utf-8"));
        httpResponse = (HttpResponse) httpClient.execute(httpPost);
        BufferedReader br = new BufferedReader(new InputStreamReader(httpResponse.getEntity().getContent()));
        String line = null;
        while ((line = br.readLine()) != null) {
            res.append(line);
        }
    }
}
```

图 4-6-1 交流界面后端代码

4.3 医生回答问题模块

医生可查看自己问题列表来选择问题回答或者去问题广场选择问题回答, 点进问题详情知晓后可进行回答, 代码和上述一样, 不再展示:



图 4-7 医生回答页面

点击聊天之后可与患者进行交流



图 4-8 医患交流页面

4.4 查看问题与评论模块

医生可查看问题详情然后进行状态跟进

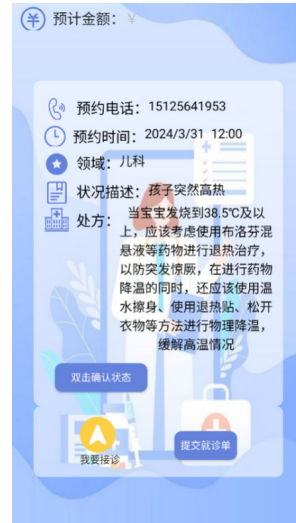


图 4-9 问题详情界面

患者可以查看自己的历史问题来进行问题复查

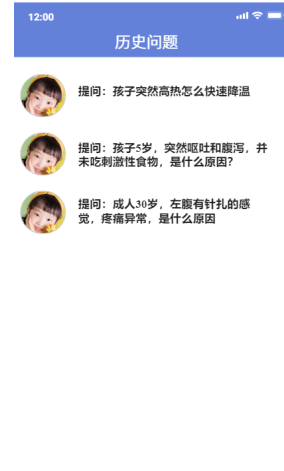


图 4-10 历史问题界面

医生也可以看自己的就诊记录来进行评级



4-11 历史就诊记录界面

患者之后可以进行评论来评价自己的满意程度



图 4-12 评论界面

4.5 登录注册模块

医生患者一开始都需进行统一注册, 之后可通过账号密码登录。一开始会有医生患者用户端选择。



4-13 登录注册首界面

医生端注册还需要医师证明, 更严密一些。



图 4-14 医生注册界面

病患注册界面



图 4-15 用户注册界面

第 5 章 系统测试

测试目标：对医疗咨询平台的登录注册、病人提出问题、医生回答问题、查看问题与评论模块进行系统测试，验证其功能和性能是否符合需求。

测试内容：登录注册模块：包括病患注册、病患登录等功能。病人提出问题模块：包括病患提问、浏览状态、搜索相关病状等功能。医生回答问题模块：包括医生回答问题、问题分类、医生资质验证等功能。查看问题与评论模块：包括问题查看、评论回复、评论管理等功能。

表 5-1 登录注册模块用例测试表

测试项	测试内容	预期结果	是否通过
用户注册	输入有效用户名、密码进行注册	注册成功，能够成功登录系统	是
用户登录	使用注册信息登录系统	能够成功登录系统	是
密码找回	测试密码找回功能	能够成功找回密码	是
注销登录	用户可以注销登录	能够成功注销登录	是

表 5-2 病人提出问题测试用例表

测试项	测试内容	预期结果	是否通过
提出问题	人可以提出咨询问题	够成功提交问题，问题显示在系统中	是
问题查看	人可以查看自己提出的问题	能够成功查看问题	是
问题搜索	病人可以根据关键词搜索问题	能够成功搜索到相关问题	是

表 5-3 医生回答问题测试用例表

测试项	测试内容	预期结果	是否通过
查看问题	可以查看病人提出的问题	能够成功查看问题	是
问题分类	可以回答病人提出的问题	能够成功回答问题	是
回答问题	医生可以对问题进行分类	能够成功分类问题	是
资质验证	系统可以验证医生的资质	成功验证医生的资质	是

表 5-4 查看问题用户评论测试用例表

测试项	测试内容	预期结果	是否通过
查看问题	用户可以查看问题和相关评论	能够成功查看问题和评论	是
发表评论	用户可以对问题进行评论	能够成功发表评论	是
评论回复	用户可以回复他人的评论	能够成功回复评论	是
评论删除	用户可以删除自己的评论	能够成功删除评论	是

第 6 章 总结与展望

本文基于 Java Web 下的 Spring Boot 开发了一款医疗咨询平台，运用软件工程中的各种设计和架构方法、测试方法等，使医疗咨询平台越来越完善。通过对整个项目的设计和实现，取得了一定的成果。

综上所述，医疗咨询平台具有广阔的发展前景和重要的社会意义。未来，伴随着人们日益增长的医疗服务需求，医疗咨询平台可以为医疗健康产业出一份力成为其重要帮手，为人们提供更便捷、更高效、更智能的医疗健康服务。

参考文献

[1] 马征辉,李宛娜,王蕾等.关于线上医疗的初探[J].电脑知识与技术,2020,16(14):110-111.
 [2] 易观:2017 年第 2 季度医药电商:紧密结合医疗产业,占据未来处方药网售时代缺口 [EB/OL].(2017-08-24)[2019-05-20].
 [3] 陈家和,马锦炉,张育玮.互联网医疗下患者持

续线上咨询和线下就诊意愿影响因素研究 [J]. 中国全科医学,2020,23(25):3164-3169.

[4] 张芝子,李伟,吴洋,等.“联网+健康医疗”服务新模式分析 [J]. 价值工程,2017,36(8):56-58.
 [5] 张宏升.软件架构的非功能性需求指标和区域化支持 [J]. 电脑知识与技术,2011,7(09):2085-2086+2089.
 [6] 牛博学.在线医疗社区信息对患者线下转线上医疗的影响研究 [D]. 西安理工大学,2023.
 [7] 林志吟,顾莹.推线上义诊、打通药品配送 互联网医疗企业为“保就医”奔波 [N]. 第一财经日报,2022-04-11(A07).
 [8] 李俊萌.计算机软件测试技术与开发应用策略分析 [J]. 信息记录材料,2023,24(03):50-52.
 [9] Sims M J. Communities of practice: Telemedicine and online medical communities [J]. Technological Forecasting Social Change.2018.126:53-63.