【健康导航平台】

设计报告

主要撰写者：于嘉伟

子模块具体设计：C 组各小组全体成员

初审：赵嘉成

初稿撰写日期：2021.4.23

第一次修订日期：2021.4.23

目录

Chapter1. 引言

本章首先说明本设计报告的编写目的和背景；接着定义一些涉及本报告的专业术语，最后列出与本报告相关的参考文献。

1.1 编写目的

从本阶段开始，项目进入正式开发阶段。这份总体设计报告的编写目的，是以本项目的需求分析说明书为依据，从总体设计的角度，明确健康导航完整开发的总体架构、流程、数据结构、数据库设计。

目的在于:

- 为开供依据
- 为修改、测试、维护提供条件
- 明确各模块外部接口、内部接口、用户接口

预期读者:

- 软件客户
- 项目经理
- 项目开发人员
- 软件质量分析人员
- 系统维护人员

1.2 相关背景

互联网医疗是互联网在医疗行业的新应用，具体指以互联网平台为基础的医疗咨询或服务，包括健康教育、医疗信息查询、电子健康档案、疾病风险评估、在线疾病咨询、电子处方、远程会诊、远程治疗与康复等医疗服务，其为民众提供更便捷的医疗知识普及与医疗服务，有利于解决中国医疗资源不平衡和人们日益增加的健康医疗需求之间的矛盾。

随着移动互联网技术的发展，医疗与电商平台结合紧密，行业产业链得到延伸，互联网医疗潜在巨大 规模得以激活，根据相关数据显示，2020 年互联网医疗市场规模超 900 亿元，同时疫情逐渐培养了消费者 对在线医疗的认识，提高了人民对在线医疗的认可度，人们通过手机 APP 咨询获取专业健康的需求也越 来越旺盛。

2020 年 7 月国务院办公厅印发的《关于进一步优化营商环境更好服务市场主体的实施意见》中提出 “在保证医疗安全和质量前提下，进一步放宽互联网诊疗范围，将符合条件的互联网医疗服务纳入医保报销范围……”，同时为响应浙江省医疗卫生服务领域“最多跑一次”改革，推动“互联网 + 智慧医疗”实行， 故本次软工项目实践为在《互联网医疗》主题下的一个健康导航平台，整体上分为——挂号预约系统、电子健康档案、健康检测、在线药房和健康论坛五个模块，基本可满足医疗场所对于网上服务的要求，产品 成型后可供诊所、小型医院、保健站等场所使用。

1.3 定义部分

Linux Linux，全称 GNU/Linux，是一种免费使用和自由传播的类 UNIX 操作系统。是一个基于 POSIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。它能运行主要的 Unix 工具软件、应用程序和网络协议。它支持 32 位和 64 位硬件。

MySQL MySQL 是一种关系型数据库管理系统，将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。其拥有体积小、速度快、总体拥有成本低，开放源码的特点。

ASP.NET Core ASP.NET Core 是一个跨平台的高性能开源框架，用于生成启用云且连接 Internet 的新式应用。它是一个模块化框架，既可以 Windows 上的完整.NET Framework 上运行，也可以在跨平台.NET Core 上运行。该框架是一个完整的重写，它将先前单独的 ASP.NET MVC 和 ASP.NET Web API 整合到单一编程模型中。尽管它是一个新的框架，建立在新的 Web 栈上，但它与 ASP.NET MVC 具有高度的概念兼容性。ASP.NET Core 应用程序支持并排版本控制，其中运行在同一台机器上的不同应用程序可以以 不同版本的 ASP.NET Core 为目标。

Vue2 Vue 是一套用于构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用单文件组件和 Vue 生态系统支持的库开发的复杂单页应用。Vue.js 的目标 是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。

用例 用例 (Use Case) 是 UML(United Modeling Language, 统一建模语言)中的重要概念。在 UML 的文档中，用例是在不展现一个系统或子系统内部结构的情况下，对系统或子系统的某个连贯的功能单元的定义和描述。用例是一种通过用户的使用场景来获取需求的技术。每个用例提供了一个或多个场景，该场景说明了系统是如何和最终用户或其它系统互动，也就是谁可以用系统做什么，从而获得一个明确的业务目标。

用例图 用例图(Use Case Diagram)是是用户与系统交互的最简表示形式，是被称为参与者的外部用户所能观察到的系统功能的模型图，展现了参与者和与他相关的用例之间的关系。用例图通过对系统、子系统或类的行为的可视化，展示了用例之间以及同用例参与者之间如何相互联系，同时帮助开发者和用户理解这些元素，并进行对应的实现。

状态图 状态图 (State Diagram) 是描述一个实体基于事件反应的动态行为，显示了该实体如何根据当前所处的状态对不同的事件做出反应的。通常我们创建一个 UML 状态图是为了以下的研究目的:研究类、角色、子系统或组件的复杂行为。

数据流图 数据流图(Data Flow Diagram):简称 DFD，它从数据传递和加工角度，以图形方式来描述系统中数据流程，从而表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。它标志了一个系统的逻辑输入和逻辑输出，以及把逻辑输入转换逻辑输出所需的加工处理。

CRC卡 CRC (Class-Responsibility-Collaborator) Card 是目前比较流行的面向对象分析建模方法。CRC卡是一个标准索引卡集合，包括类名、类的职责、类的协作关系三部分，每一张卡片表示一个类。类代表一系列对象的集合，这些对象是对系统设计的抽象建模，可以是一个人、一件物品等等;职责包括这个类对自身信息的了解，以及这些信息将如何运用;协作指代另一个类，我们通过这个类获取我们想要的信息 或者相关操作。

类图 类图 (Class diagram) 是显示类、类的内部结构、类之间的关系的静态结构。类图是面向对象建模的 主要组成部分。它既用于应用程序的系统分类的一般概念建模，也用于详细建模，将模型转换成编程代码。类图也可用于数据建模。

1.4 参考资料

1. 本项目的经核准的计划任务书
2. 本项目的需求分析报告书
3. Roger S Pressman. Software Engineering: A Practitioner's Approach. McGraw-Hill Education, 9 edition, 2020.
4. 本文档所参考的软件开发标准。
   a) 《计算机软件产品开发文件编制指南(GB/T 8567)》
   b) 《计算机软件文档编制规范(GB/T 8567-2006)》
   c) 《计算机软件需求说明编制指南(GB/T 9385)》
5. 本文档所参考的医疗系统开发要求
   a) 中华人民共和国国家卫生健康委员会. 电子病历基本数据集, 2014.
   b) 国家卫生计生委. 电子病历共享文档规范第 1 部分:病历概要等 57 项卫生行业标准.

Chapter2. 总体设计

2.1 需求规定

2.1.1 功能需求

挂号预约功能：

- 用户端能够进行挂号和退号处理，能够查询自己的挂号信息；
- 医生端能够查看与自己相关的预约记录，能够查看预约患者的主诉和病史；
- 医生端能够标记和查看病人的违约情况；
- 能够自动调整排班，根据医生调班需要调整预约时间或者分流预约患者；
- 能够统计医生处理的预约订单数据。

健康检测功能：

- 用户端能够进行核酸检测和体检的分时段预约；
- 用户端能够查看过往核酸检测和体检的报告；
- 医生端能够修改核酸检测和体检的各时段容量；
- 医生端能够发布核酸检测和体检的报告。

电子健康档案功能：

- 用户端能够浏览个人基本信息，能够绑定医保卡和病历卡；
- 用户能够查询和浏览过往病历资料；
- 能够实现日常提醒功能；
- 能够实现基本的统计功能。

网上药房功能：

- 能够实现药品的管理和搜索；
- 能够实现病人便捷取药；
- 能够实现药品购物车和订单交易；
- 能够实现防疫物品快捷购买；
- 能够实现个人信息管理。

健康论坛功能：

- 能够实现用户之间在线问答；
- 能够实现用户发帖跟帖交流；
- 能够实现用户个人主页与认证信息。

2.1.2 性能需求

- 系统要具备良好的反应速度，给用户良好的使用体验；
- 在保证网络条件良好的情况下，系统应具备如下时间特性要求：
    - 单个用户在线：
        - 响应用户动作时间小于1 秒；
        - 信息检索响应时间小于2 秒；
    - 多用户同时在线：
        - 响应用户动作时间小于2 秒；
        - 信息检索响应时间小于5 秒；
- 系统至少要支持同一时间内500个用户并发访问。

2.1.3 其他需求

接口要求：

- 各子系统之间的耦合关系必须明确清楚，整体系统的耦合算法必须准确并可根据需要进行调整。
- 系统框架内外部接口关系良好，可以与子系统方便地进行交互和数据传递。
- 各子系统的接口和参数具有很好的开放性，可以方便查阅和调整，系统框架和子系统应具备独立运行自检的功能。
- 本系统应具备扩充服务功能的能力。总体架构设计，应考虑到功能的可扩展性。

界面或人机接口要求：

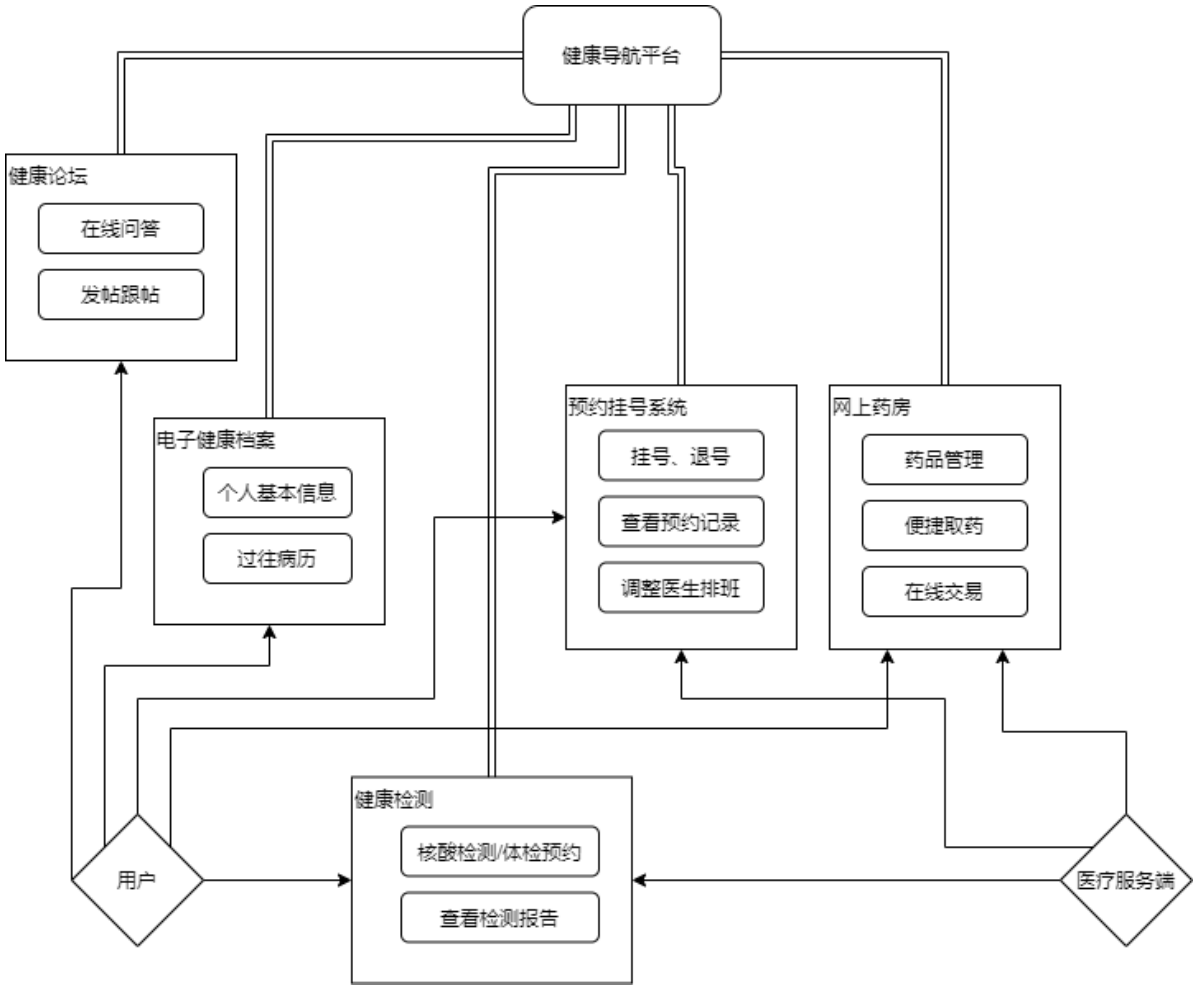
- 系统的界面应简洁美观、布局合理，突出重点内容，点选按键和显示内容应大小适宜；
- 各项服务的操作流程应简明合理且方便，降低用户的学习成本。
- 对不同移动设备应具有较好的兼容性，保证用户体验的完整性和一致性。

系统质量要求：

- 正确性：系统框架模型、算法编写要正确，调用接口与工作流程要准确。
- 易用性：系统界面友好使用方便，系统的部署与使用简便。
- 健壮性：系统应有一定的抗用户误操作而崩溃的能力，在一定的错误范围内能够恢复到正常运行状态。
- 可靠性：对用户的输入信息能及时自动保存，当系统异常停止运行时数据要容易恢复。
- 安全性：系统不应被设备上的其他软件影响运行，用户的私密数据不能被泄露，系统的重要数据不能被非法篡改或盗取。
- 可扩展性：本系统可扩展性良好，可灵活增加新的服务内容。
- 可维护性：输入、输出和运行日志易于存储和备份。
- 开放性：立足开放性接口和服务，能够与各医院的系统联结或联结其他服务提供商，提供更灵活的使用方式

2.2 总体模块组成

健康导航平台以互联网平台作为基础，从医疗场所和患者对于网上服务的要求出发，并考虑到当前疫情下的各类医疗需求，实现一个综合了各类医疗服务的互联网医疗平台。平台的功能组成如下图所示，主要包括为挂号预约系统、电子健康档案、健康检测、在线药房和健康论坛五个模块，并针对不同的用户类型给予不同的用户服务，一类是普通用户及患者，一类是诊所，医院等医疗服务。对于广大患者用户，可以在平台上进行医疗挂号及相关缴费，完成健康检测预约，查看个人电子健康档案和相关病历查询，购买药品、防疫用品，及在平台提供的论坛进行健康发帖讨论。对于医疗服务端，可以快速管理病人预约信息，发布健康检测信息，发布相关的医疗信息。

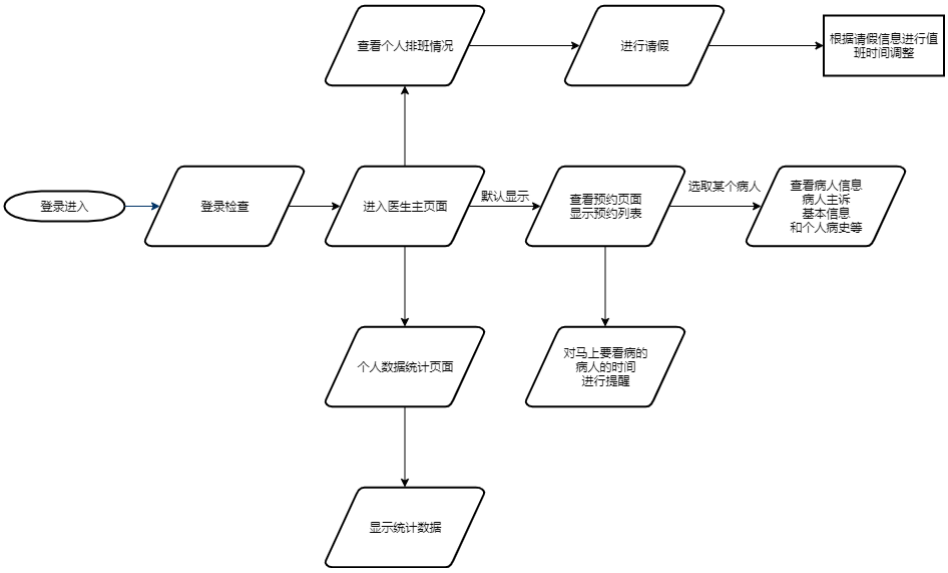


2.3 功能及性能分配

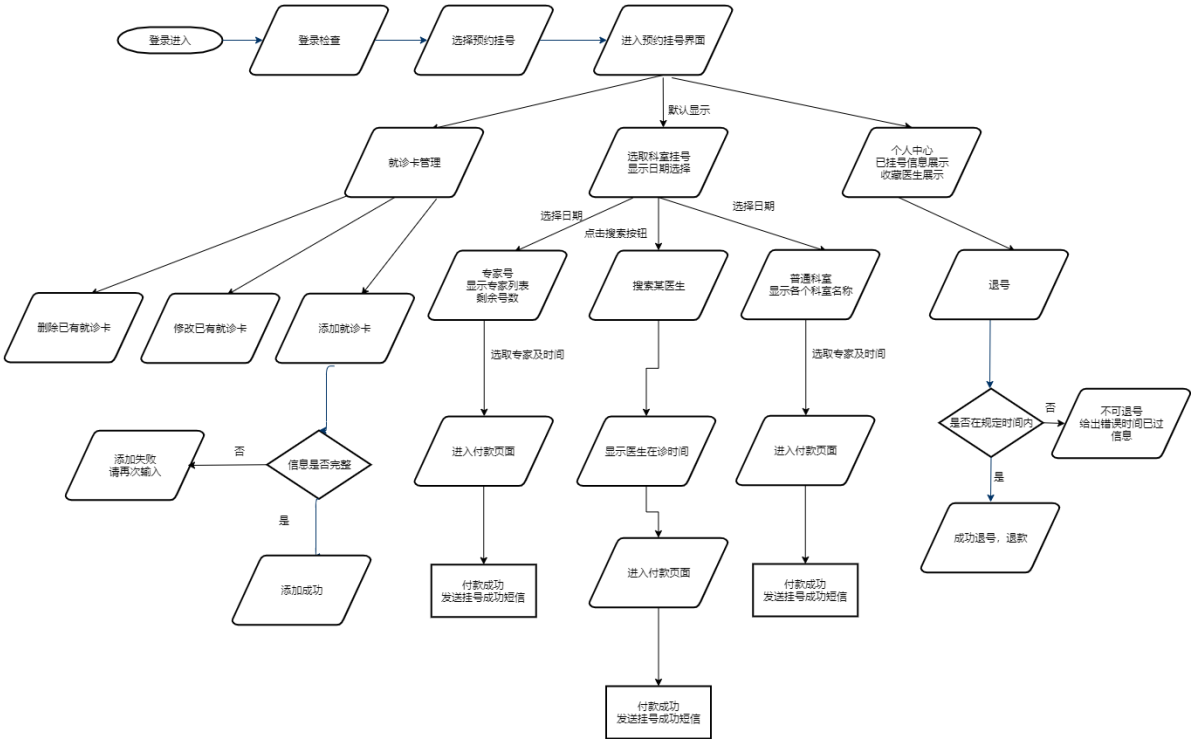
根据上节的系统模块组成及功能分析,我们可将上述五个主要基本性能分配用流程图分别描述如下：

挂号预约：

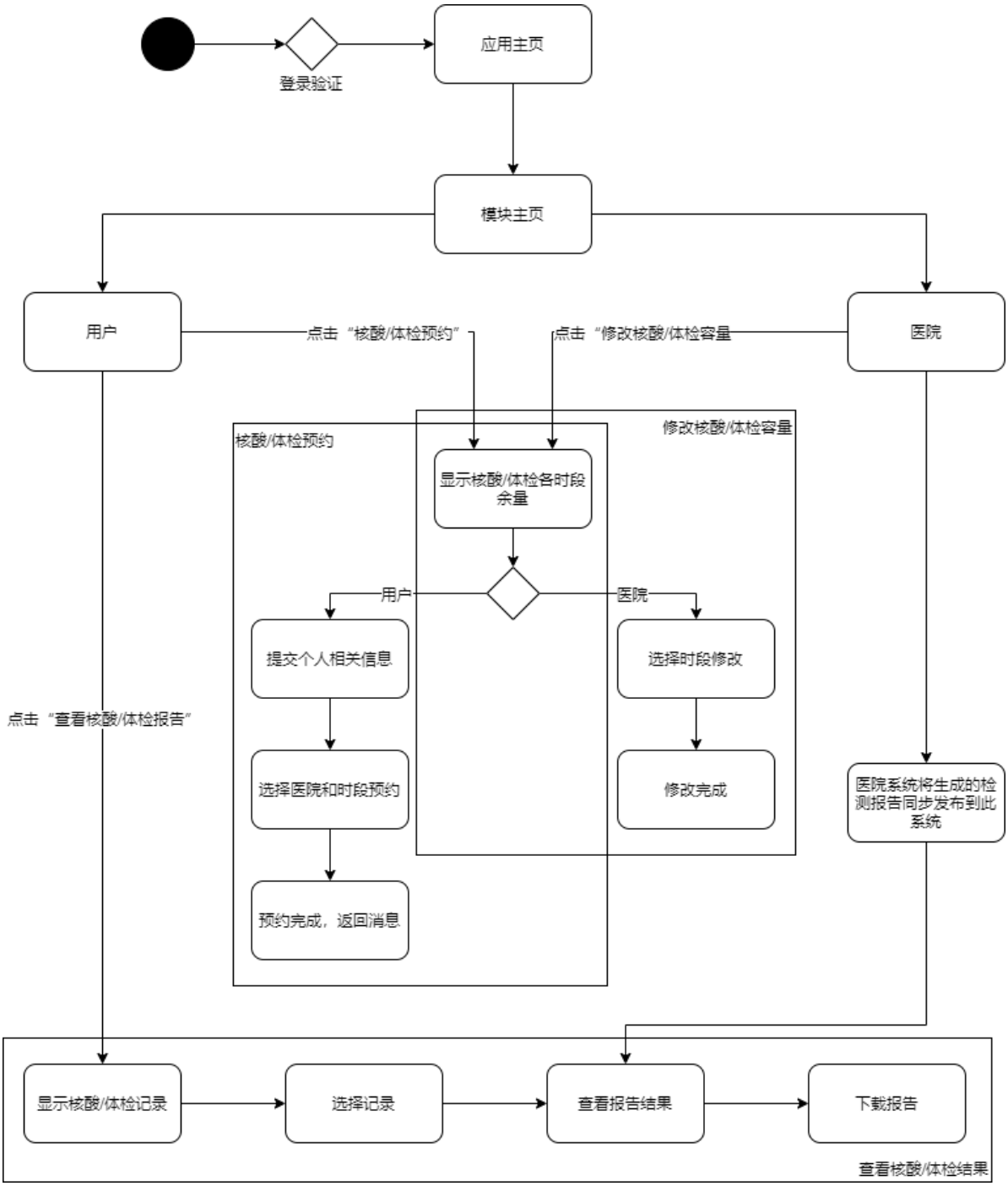
医生：



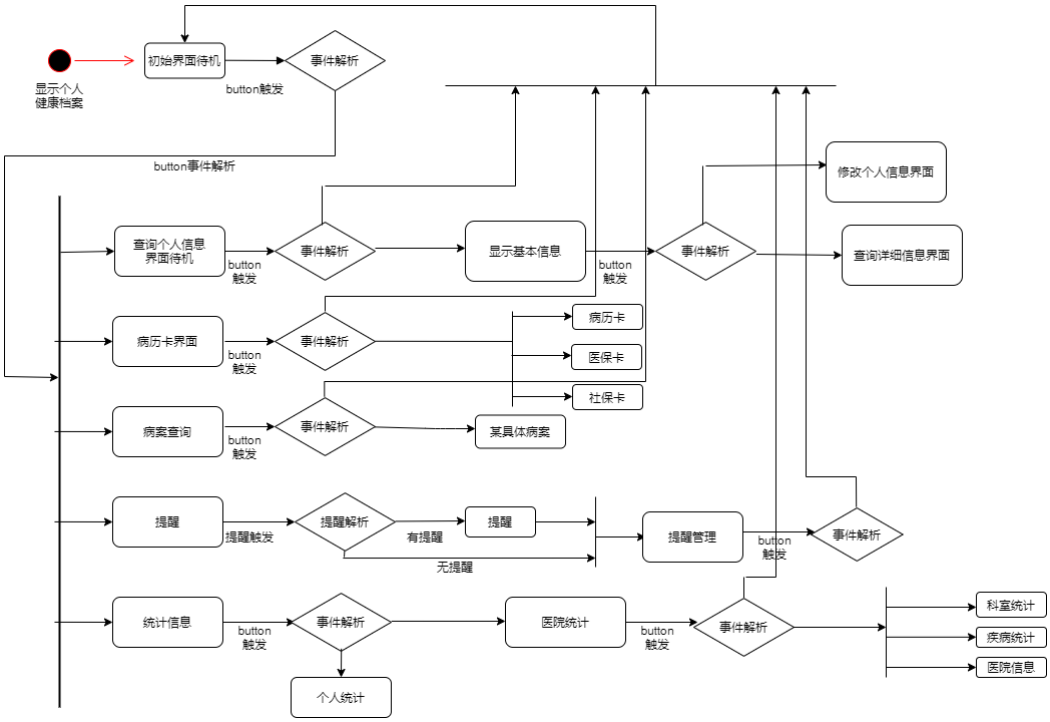
患者：



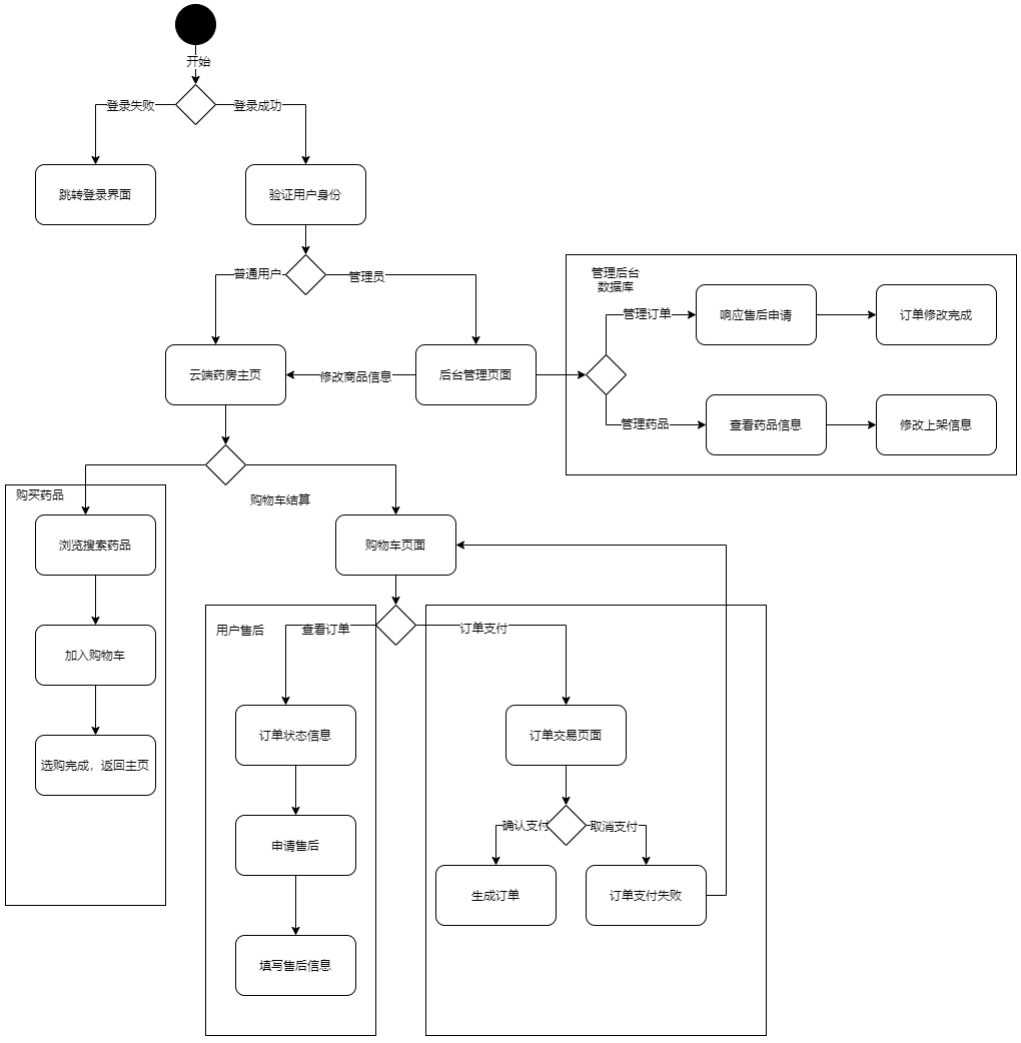
健康检测：


应用主页
登录验证
模块主页
用户
点击“核酸/体检预约”
点击“修改核酸/体检容量
医院
核酸/体检预约
修改核酸/体检容量
显示核酸/体检各时段余量
用户
医院
提交个人相关信息
选择时段修改
选择医院和时段预约
修改完成
预约完成，返回消息
点击“查看核酸/体检报告”
医院系统将生成的检测报告同步发布到此系统
显示核酸/体检记录
选择记录
查看报告结果
下载报告
查看核酸/体检结果

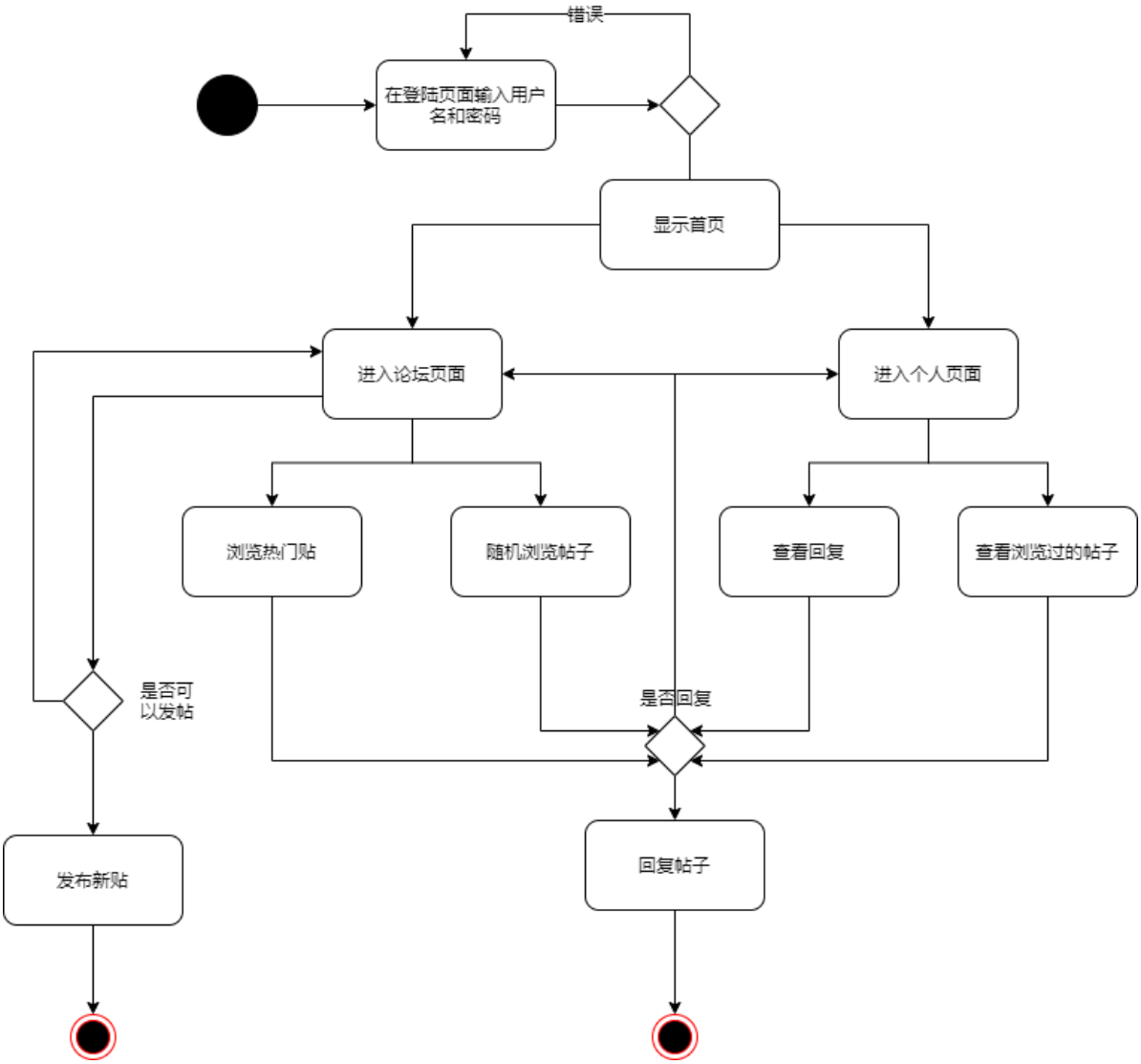
电子健康档案：


显示个人健康档案
初始界面待机
button触发
事件解析
button事件解析
修改个人信息界面
查询个人信息界面待机
button触发
事件解析
显示基本信息
button触发
事件解析
查询详细信息界面
病历卡界面
button触发
事件解析
病历卡
医保卡
社保卡
病案查询
button触发
事件解析
某具体病案
提醒
提醒触发
提醒解析
有提醒
提醒
无提醒
提醒管理
button触发
事件解析
统计信息
button触发
事件解析
医院统计
button触发
事件解析
科室统计
疾病统计
医院信息
个人统计

网上药房：


开始
登录失败
登录成功
跳转登录界面
验证用户身份
普通用户
管理员
管理后台数据库
管理订单
响应售后申请
订单修改完成
云端药房主页
修改商品信息
后台管理页面
管理药品
查看药品信息
修改上架信息
购买药品
浏览搜索药品
加入购物车
选购完成，返回主页
购物车结算
购物车页面
用户售后
查看订单
订单状态信息
申请售后
填写售后信息
订单支付
订单交易页面
确认支付
取消支付
生成订单
订单支付失败

健康论坛：


错误
在登陆页面输入用户名和密码
显示首页
进入论坛页面
进入个人页面
浏览热门贴
随机浏览帖子
查看回复
查看浏览过的帖子
是否可以发帖
是否回复
发布新贴
回复帖子

2.4 运行环境

2.5.1 服务端

- CPU：不小于2.0GHz
- 内存：不小于2.0GB
- 硬盘：不小于40GB
- 网络带宽：不小于10Mbps
- 操作系统：Linux
- 数据库：MySQL
- 后端框架：ASP .NET Core
- 开发工具：支持ASP .NET Core的后端开发工具
- 测试工具：支持ASP .NET Core

2.5.2 客户端

- CPU：不小于1.0GHz
- 内存：不小于4.0GB
- 平台载体：具备小程序功能的APP（如微信或支付宝）
- 网络功能：具备基本的网络连接功能
- 前端框架：Vue2
- 开发工具：支持Vue2的前端开发工具
- 测试工具：支持Vue2

2.5 软件架构设计

本系统基于互联网，以小程序为载体向用户提供服务，系统的正常运行依赖着后台服务器的正确工作。在内部，系统的5个模块的功能上重合度与交互度并不高，但用户和医疗服务端的有关数据需要共享，该数据存在后台服务器供各模块调取使用，且由特定模块进行维护更新。基于此特点，研发过程中更多的关注点在于前后端的通信与协作。

系统各模块的组成与关系如**图xx**所示，具体说明参见第5章各模块详细设计。

Chapter3. 软件接口设计

电子健康档案模块

3.1 用户接口

对于电子病历系统模块，主要有：

基本信息浏览界面

- 本信息窗口：显示姓名年龄身高体重等
- 过敏信息窗口：显示过敏食物、物品、药物等
- 遗传病史窗口：显示遗传病
- 住院记录窗口：显示住院记录

多卡绑定界面

- 病历卡界面：用户基本信息、医院ID等
- 医保卡界面：医保卡号、持有人个人身份信息等
- 社保卡界面：社保卡号、持有人个人身份信息等

病例查询界面

- 工作人员查询界面：某科室、某疾病的病例搜索栏
- 普通患者查询界面：个人病例列表

日常提醒界面

- 提醒设置界面：提示设置
- 提醒界面：显示提醒内容

医院统计界面

- 工作人员查询界面：信息搜索栏与展示栏
- 普通患者查询界面：信息搜索栏与展示栏（查询结果受限）

3.2 后端接口

对于电子病历系统模块，所有后端请求全部面对数据库，主要有：

请求类型	PATH	描述
GET	/api/healthrecord/person_info/information	获取患者个人信息
GET	/api/healthrecord/case/{user_phone}	获取病例病案信息

3.3 用户界面

- 个人信息


中国移动
11:58
健康档案
我的家人
自己
25岁 | 未婚 | 体重标准
健康评测
体检报告
分析报告
就医记录
监测记录
运动 步
1528
消耗23千卡
17/09/31
体重 kg
50.5
BMI 20.08
17/09/31 10:35
血压 mmHg
140/105
心率 65次/分钟
17/09/31 10:35
血糖 mmol/L
--
--
尿酸 mmol/L
0.11
17/09/31 10:35
血氧 %
94
17/09/31 10:35

- 病历卡


UI SCREEN
病历卡
Show Baotu
22岁
医保卡就诊
检测科室
心血管常规疾病
病例详情
三级高血压，高血脂等
确诊疾病
高血压三级
包图 心虚管疾病专家
509人向他提问过
提问
病情资料
头晕，眼花，四肢无力.....
院内挂号
血样采集检测
待检测状态
立即检测

- 病案介绍


UI SCREEN
中国移动
9:41 AM
100%
病历预览
2019-10-11
主诉: 最近腰椎老是莫名酸痛，也没有久坐就是莫名其妙的酸痛
一般病史:
现病史: 此处为病人现病史信息
过敏史: 此处为病人过敏史信息
既往史: 此处为病人既往史信息
体格检查:
身高 175CM
体重 74KG
体温 38度
来源 口腔
脉搏 90次/分
血压 120MMHG
症状: 最近腰椎老是莫名酸痛，也没有久坐就是莫名其妙的酸痛
妊娠
哺乳
确定

云端药房模块

3.1 用户接口

对于云端药房模块，主要有：

- 云端药房主界面
    - 药品简略信息区：显示药品简略信息
    - 分类选择区：选择不同类别的药品
    - 搜索栏：按照名称搜索药品
- 药品详情界面
    - 药品图片区：显示药品图片
    - 药品详细信息区：显示药品详细信息

3.2 后端接口

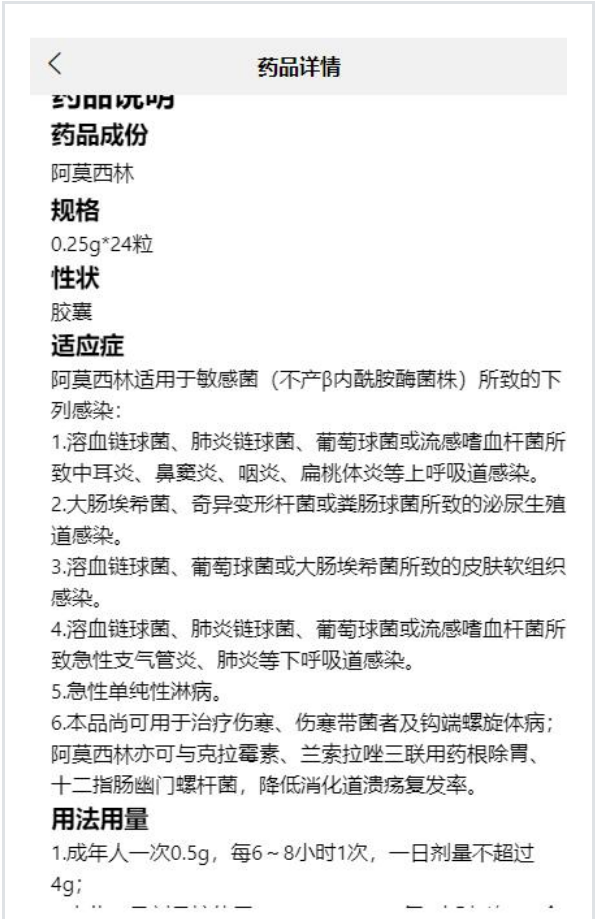
对于云端药房模块，主要有：

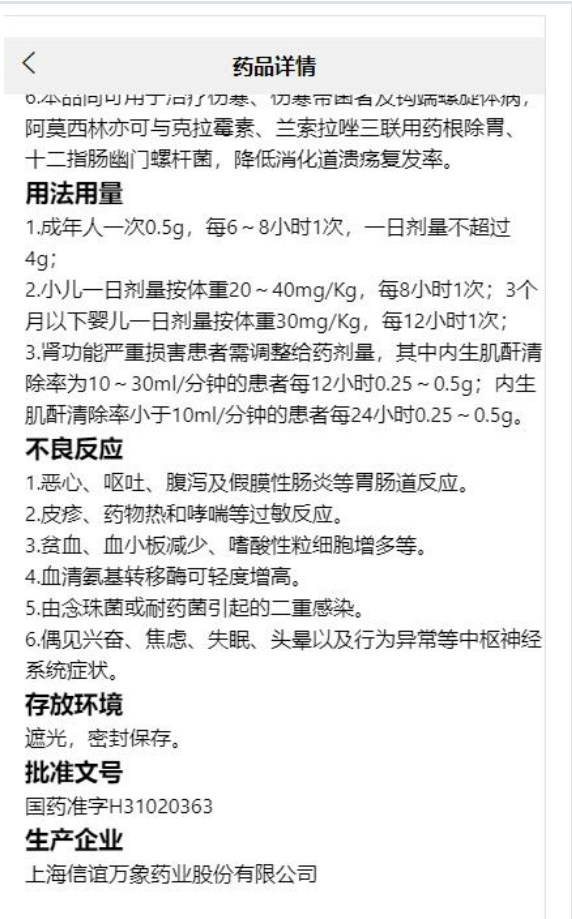
请求类型	PATH	描述
GET	/api/phar/list	获取药品列表
GET	/api/phar/detail	获取药品详情
GET	/api/phar/search	根据药品名搜索药品（暂时不用做模糊匹配）
GET	/api/phar/chart/list	获取购物车列表
POST	/api/phar/chart/add	添加商品
DELETE	/api/phar/chart/delete	删除购物车中若干个条目
GET	/api/phar/order/list	获取订单列表
GET	/api/phar/order/detail	获取订单详情
POST	/api/phar/order/add	添加订单
PUT	/api/phar/order/cancel	取消订单

3.3 用户界面


在线药房
感冒头痛
肠胃消化
心脑血管
风湿骨伤
皮肤用药
呼吸用药
五官用药
妇科用药
感冒头痛
阿司匹林
对血小板的聚集有抑制作用
￥14.99
阿莫西林
治疗伤寒、其他沙门菌感染
￥15.99
肠胃消化
感冒灵胶囊
解热镇痛
￥24.99
黄连上清片
滋阴养胃
￥13.99
心脑血管
养胃舒颗粒
散风清热，泻火止痛
￥23.99
健胃消食片
健胃消食
￥6.99
风湿骨伤
测试页


药品详情
阿莫西林胶囊
AMOXICILLIN CAPSULES
0.25g
阿莫西林
库存0
加入购物车
13.99￥/盒
药品说明
药品成份
阿莫西林
规格
0.25g*24粒


药品详情
药品成份
阿莫西林
规格
0.25g*24粒
性状
胶囊
适应症
阿莫西林适用于敏感菌（不产β内酰胺酶菌株）所致的下列感染：
1.溶血链球菌、肺炎链球菌、葡萄球菌或流感嗜血杆菌所致中耳炎、鼻窦炎、咽炎、扁桃体炎等上呼吸道感染。
2.大肠埃希菌、奇异变形杆菌或粪肠球菌所致的泌尿生殖道感染。
3.溶血链球菌、葡萄球菌或大肠埃希菌所致的皮肤软组织感染。
4.溶血链球菌、肺炎链球菌、葡萄球菌或流感嗜血杆菌所致急性支气管炎、肺炎等下呼吸道感染。
5.急性单纯性淋病。
6.本品尚可用于治疗伤寒、伤寒带菌者及钩端螺旋体病；阿莫西林亦可与克拉霉素、兰索拉唑三联用药根除胃、十二指肠幽门螺杆菌，降低消化道溃疡复发率。
用法用量
1.成年人一次0.5g，每6～8小时1次，一日剂量不超过4g；


药品详情
阿莫西林亦可与克拉霉素、兰索拉唑三联用药根除胃、十二指肠幽门螺杆菌，降低消化道溃疡复发率。
用法用量
1.成年人一次0.5g，每6～8小时1次，一日剂量不超过4g；
2.小儿一日剂量按体重20～40mg/Kg，每8小时1次；3个月以下婴儿一日剂量按体重30mg/Kg，每12小时1次；
3.肾功能严重损害患者需调整给药剂量，其中内生肌酐清除率为10～30ml/分钟的患者每12小时0.25～0.5g；内生肌酐清除率小于10ml/分钟的患者每24小时0.25～0.5g。
不良反应
1.恶心、呕吐、腹泻及假膜性肠炎等胃肠道反应。
2.皮疹、药物热和哮喘等过敏反应。
3.贫血、血小板减少、嗜酸性粒细胞增多等。
4.血清氨基转移酶可轻度增高。
5.由念珠菌或耐药菌引起的二重感染。
6.偶见兴奋、焦虑、失眠、头晕以及行为异常等中枢神经系统症状。
存放环境
遮光，密封保存。
批准文号
国药准字H31020363
生产企业
上海信谊万象药业股份有限公司

体检预约模块

3.1 用户接口

主界面：体检预约入口、核酸检测预约入口；

体检预约界面：查看体检报告入口、医院选择栏、日期选择栏；

体检预约时段选择界面：时段选择栏；

体检时段余量修改界面：时段余量修改栏；

核酸检测预约界面：查看体检报告入口、医院选择栏、日期选择栏；

核酸检测预约时段选择界面：时段选择栏；

核酸检测时段余量修改界面：时段余量修改栏；

3.2 后端接口

体检预约部分：

请求类型	PATH	描述
GET	/api/exam/physical/hospital	查询医院列表
GET	/api/exam/physical/remainder	查询预约余量
POST	/api/exam/physical/appointment	新增预约
GET	/api/exam/physical/appointment/user_phone	查询预约信息
GET	/api/exam/physical/report/appoint_id	获取体检报告
GET	/api/exam/physical/setting	查询余量设置
PUT	/api/exam/physical/setting	修改余量设置

核酸检测预约部分：

请求类型	PATH	描述
GET	/api/exam/covid/hospital	查询医院列表
GET	/api/exam/covid/remainder	查询预约余量
POST	/api/exam/covid/appointment	新增预约
GET	/api/exam/covid/appointment/user_phone	查询预约信息
GET	/api/exam/covid/report/appoint_id	获取体检报告
GET	/api/exam/covid/setting	查询余量设置
PUT	/api/exam/covid/setting	修改余量设置

3.3 用户界面

图1 主界面

图2 体检/核酸检测预约界面

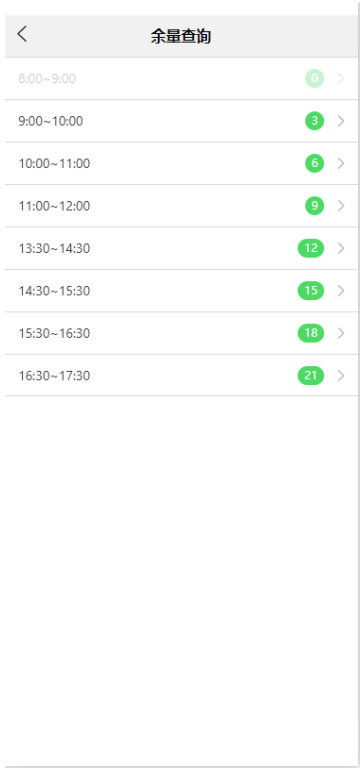


图3 体检/核酸检测预约时段选择界面

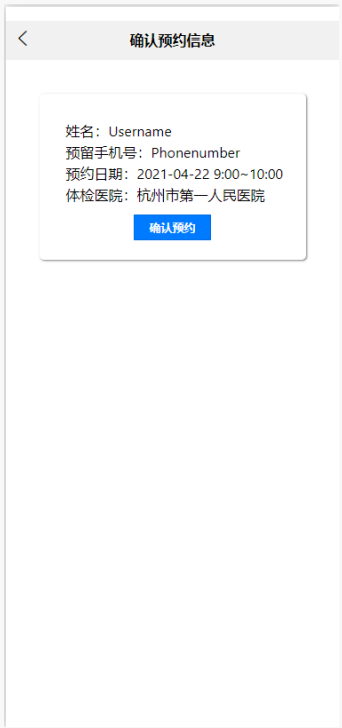


图4 体检/核酸检测预约确认界面

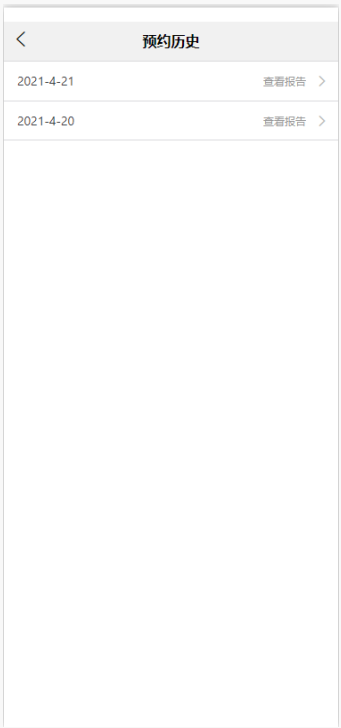


图5 体检/核酸检测查看报告界面

健康论坛模块

3.1 用户接口

对于健康论坛模块，主要有：

- 贴子列表界面
  - 贴子列表
  - 发帖按钮
- 贴子浏览界面
  - 给贴子或其回复点赞、踩，收藏
  - 回复贴子
- 发帖/回复界面
  - 贴子编辑框（包括标题输入，文字样式选择等）
- 问题列表界面
  - 问题列表
  - 发起提问
- 提问界面
  - 提问编辑框（包括问题标题输入，问题细节描述输入等）
- 问题与回答简介浏览界面
  - 浏览问题以及下面的回答
  - 发起回答
- 发起回答界面
  - 回答内容编辑框
- 回答内容浏览界面
  - 浏览回答
  - 给回答点赞、踩，收藏
  - 查看该回答下的回复
  - 回复该回答
- 回复浏览界面
  - 浏览回复
  - 给回复点赞、踩
- 回复某一回答界面
  - 回复内容编辑框
- 查看他人信息时
  - 显示其用户信息、最近发帖、最近提问、最近回答
- 查看自己的信息时
  - 除了上述信息之外，还要展示自己收藏的贴子、回答

3.2 后端接口

对于健康论坛模块，主要有：

请求类型	PATH	描述
GET	/api/forum/post	查询主题贴列表
GET	/api/forum/post/{id}	查询单个主题贴的信息
POST	/api/forum/post	新增主题贴
PUT	/api/forum/post/{id}	贴子内容编辑
DELETE	/api/forum/post/{id}	删除贴子

3.3 用户界面

挂号预约模块

3.1.1 病人端用户接口

1. 就诊人管理界面：
    - 已有就诊人选择查看区：点击已有就诊人，跳转页面显示详细信息并可选择是否修改与保存
    - 添加就诊人按钮：点击按钮，跳转界面填写信息，选择是否确认添加
2. 预约医院院区选择界面
    - 选择不同医院横条区：点击不同横条区，跳转界面显示不同医院科室信息
3. 科室选择界面
    - 选择不同科室组件区：使用组件聚合不同层次门诊与对应具体科室，左侧显示门诊类型，右侧显示具体科室；选择科室，跳转新页面
4. 挂号选择界面

   根据不同层次门诊，不同选择方式
    - 专家门诊选择区：上方显示具体日期，下方显示医生列表，点击医生区域显示组件，选择挂号时间段，跳转新页面
    - 普通门诊选择区：仅显示挂号日期列表，点击日期区域显示组件，选择挂号时间段，跳转新页面
5. 挂号确认页面
    - 更换就诊人区域：点击当前就诊人，跳转就诊人列表进行更换
    - 确认预约按钮：点击按钮，完成本次预约，跳转
6. 退号查询界面
    - 更换就诊人区域：点击当前就诊人，跳转就诊人列表进行更换
    - 确认查询按钮：点击按钮，跳转新页面
7. 挂号列表界面
    - 不同医院选择区：上方点击不同区域，下方显示对应就诊人预约挂号记录
    - 退号按钮：选择带有退号按钮的记录，显示新组件，提示确认退号，确认后退号完成

3.1.2 医生端用户接口

1. 认证界面：
    - 用户名及密码输入框：点击分别输入字符，密码加密显示
    - 确认登录按钮：点击，跳转界面
2. 已挂号病人列表界面：
    - 选择某日期组件：点击选择具体某日，确认显示内容
    - 某日病人列表选择区：(病人按照挂号时间段以及挂号时间前后顺序显示)点击某病人对应区域，跳转新界面
3. 病人信息界面

3.2.1 病人部分后端接口

1. 挂号部分：实现用户个人信息表的查询，预约记录表的查询和增改

接口	地址	描述
GET	/api/appoinment/register/hospital	查询医院列表
GET	/api/appoinment/register/room/{hospital}	查询某医院科室列表
GET	/api/appoinment/register/doctor_list	查询某科室的医生列表
GET	/api/appoinment/register/doctor_remainder	查询某医生预约数量
POST	/api/appoinment/register/record	新增预约
GET	/api/appoinment/register/record/user_phone	查询预约信息

2. 退号部分：实现预约记录表的查询与具体某条记录的删除

接口	地址	描述
GET	/api/appoinment/register/record/user_phone	查询预约信息
DELETE	/api/appoinment/withdraw/record/user_phone	删除具体某条预约记录

3.2.2 医生部分后端接口

实现医生对挂号病人列表与某具体病人详细信息的查询

接口	地址	描述
GET	/api/appoinment/doctor/{doctor_id}register_list	查看病人挂号列表
GET	/api/appoinment/doctor/{patient_id}/infomation?appoint_date=xxxx	查看病人详细信息

3.3 用户界面

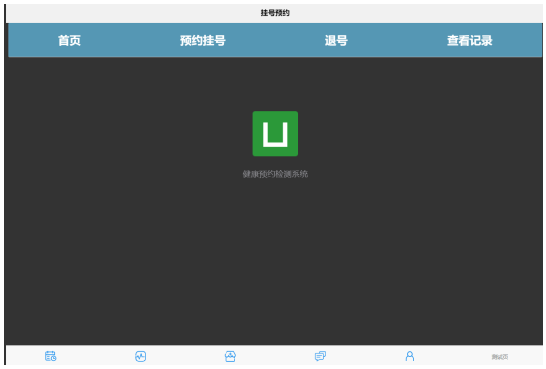


图1 病人端预约子模块UI

图2 预约医院和科室选择界面

图3 预约医院和科室选择界面

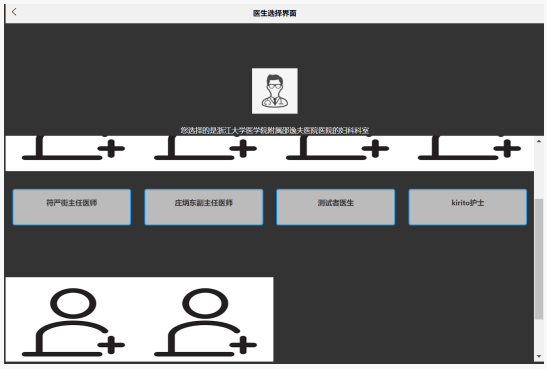


图4 医生选择界面

Chapter4. 运行设计

电子健康档案模块

4.1 运行页面模块组合

网页内部分布为五个子模块，点击不同图标可以进入对应的模块界面。默认从挂号预约开始，从左到右依次是：挂号预约、健康检测、在线药房、健康论坛、个人中心。

4.2 运行控制

健康导航平台控制逻辑对用户的响应如下：

用户选择某个病历

1. index.vue调用api模块导出的获取列表接口
2. api模块构造请求，向后端发送请求以获取病案列表
3. 在请求返回后，触发的方法会在index中的相关容器中填入获取到的数据

用户选择某个病案详情

1. index.vue获取用户所点击的病案id，根据id向api模块发送查询详细数据的异步请求
2. api模块构造请求，向后端发送请求以获取病案详细信息
3. index.vue携带部分源数据进行页面跳转，进入到病案详情页，展示加载动画
4. 请求返回后，触发的方法会在详情页中的相关容器中填入获取到的数据

用户在健康论坛讨论

index.vue获取用户所发送的信息，传送给后端服务器，然后转发给所有用户

4.3 运行时间

主页面加载时间小于5s，模块页面切换时间小于2s，任何搜索功能的用户搜索时间小于3s，健康论坛各用户时延小于1s。

云端药房模块

本章首先描述系统页面组合，再阐述系统各个子模块的运行控制流程，最后说明系统各个模块的运行时间。

4.1 运行页面模块组合

见文件pharmacy.drawio

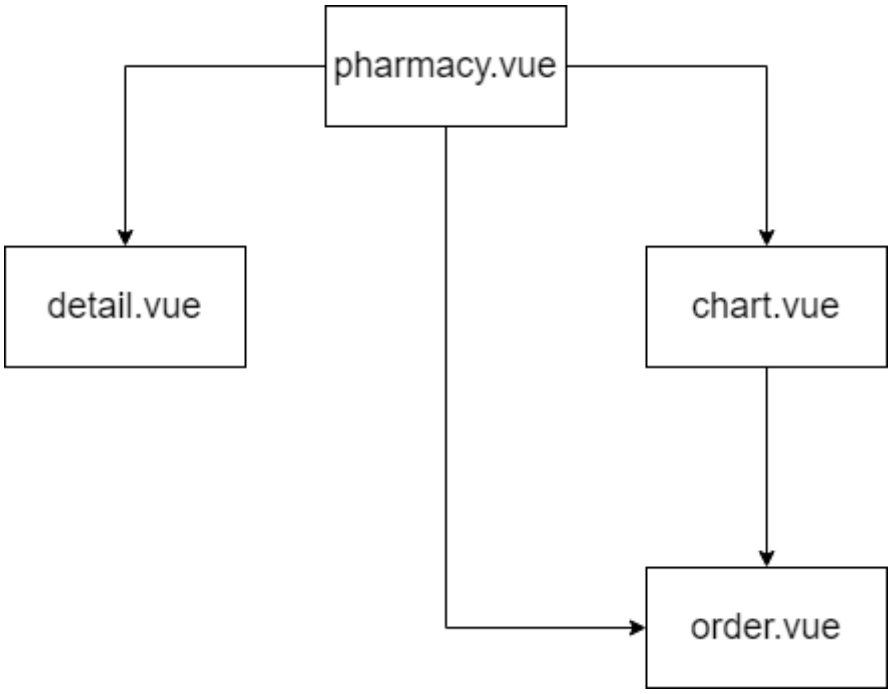


图7 K-means算法流程图

4.2 运行控制

云端药房模块主要控制流程：

- 主程序运行
- 用户点击云端药房导航栏
- 加载并显示药品列表
- 用户选择药品分类
- 用户按照药品名称进行搜索
- 用户点击某种药品跳转到详情页
- 用户从详情页点击返回跳转到主页面

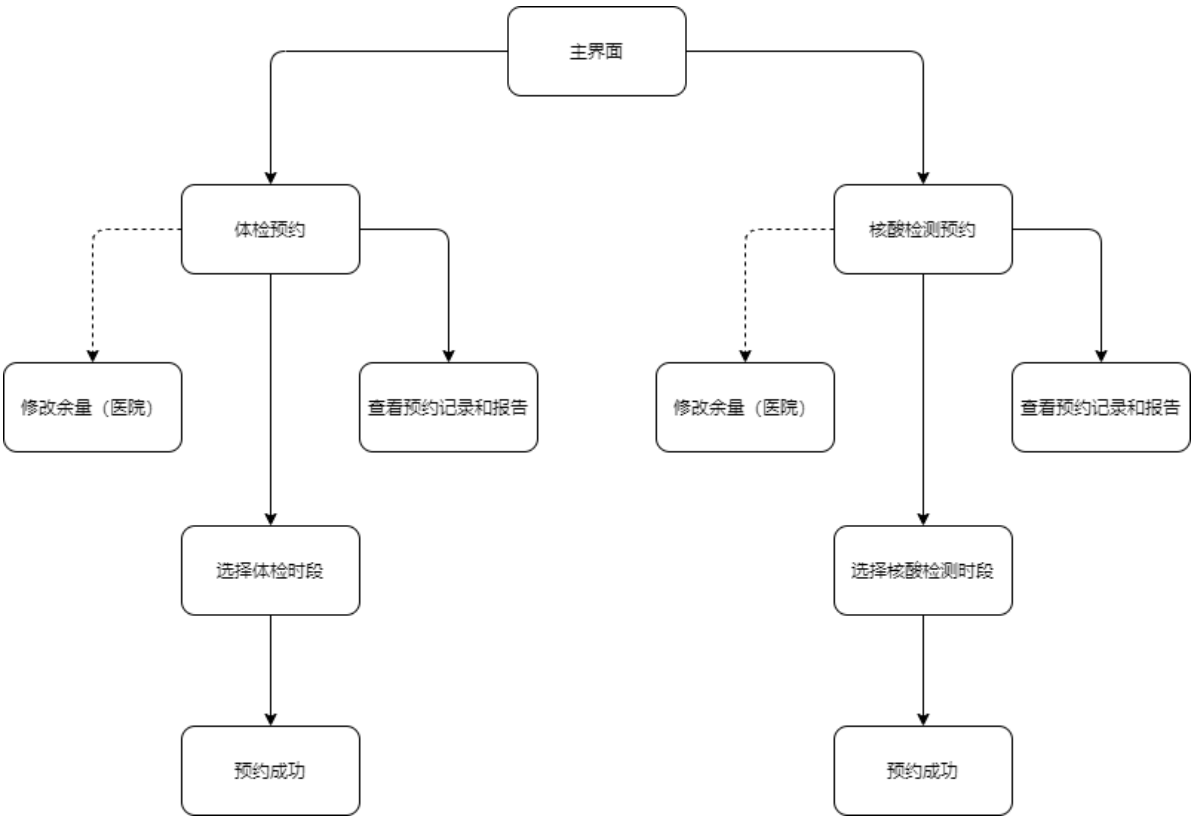
4.3 运行时间

主页面加载时间小于5s，响应用户选择分类时间小于2s，响应用户搜索时间小于3s，从主页到药品详情页跳转时间小于2s，从详情页返回到药房主页时间小于2s。

体检预约模块

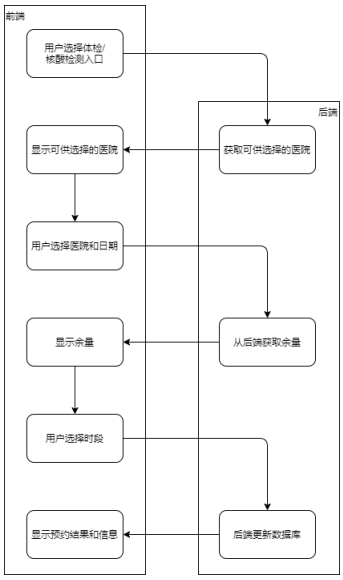
本章首先描述系统页面组合，再阐述系统各个子模块的运行控制流程，最后说明系统各个模块的运行时间。

4.1 运行页面模块组合

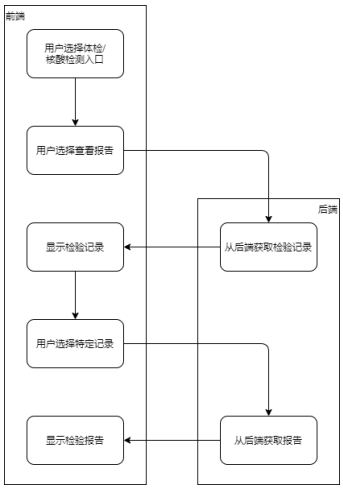


4.2 运行控制

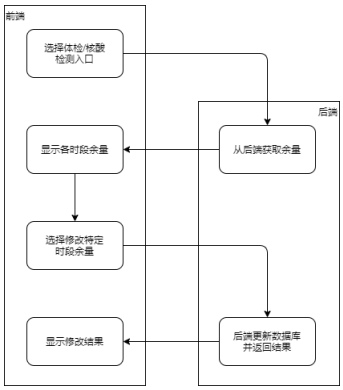
用户进行体检/核酸检测预约：
用户从主界面选择体检/核酸检测入口 →
从后端获取可供选择的医院 → 页面显示可供选择的医院和日期 →
用户选择医院和日期 →
从后端获取各时段余量 → 页面显示各时段及其余量 →
用户选择时段 →
后端更新数据库并返回结果 → 页面显示预约结果和信息



用户查看体检/核酸检测报告：
用户从主界面选择体检/核酸检测入口 →
用户选择查看报告 →
从后端获取用户的检验记录 → 页面显示用户的检验记录 →
用户选择特定记录查看报告 →
从后端获取用户的报告 → 页面显示检验报告



医院修改体检/核酸检测容量：
用户从主界面选择体检/核酸检测入口 →
从后端获取各时段余量 → 页面显示各时段余量 →
用户选择特定时段余量进行修改 →
后端更新数据库并返回结果 → 页面显示修改结果



4.3 运行时间

响应用户动作时间不超过1s，与后端交互时间不超过2s，页面加载、刷新、渲染时间不超过2s。

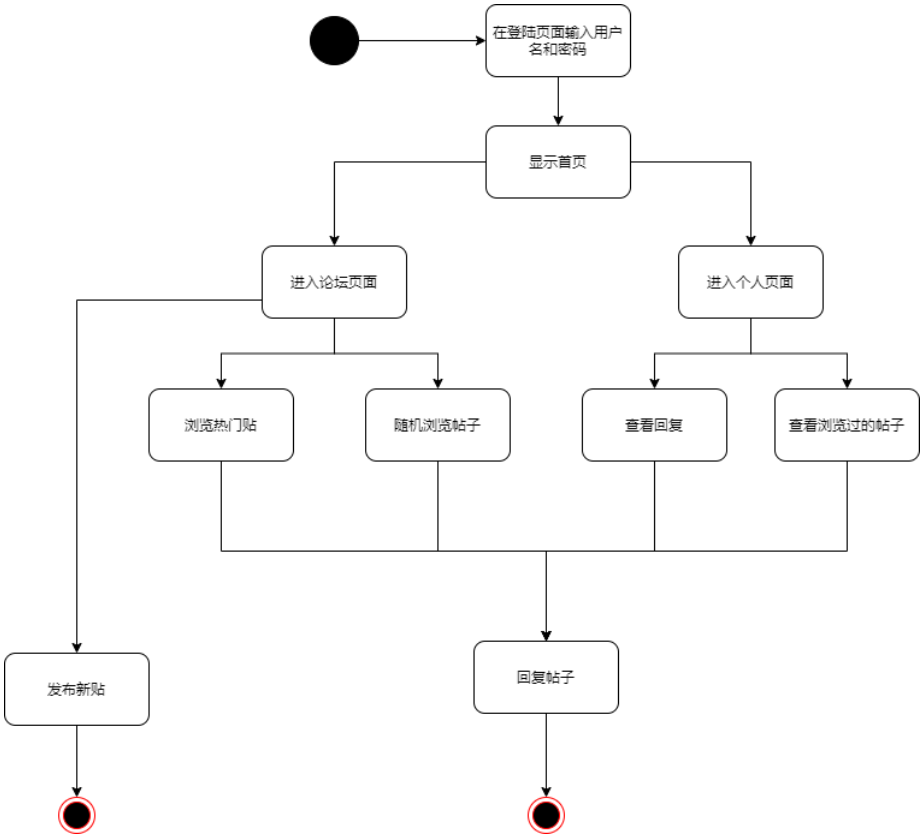
健康论坛模块

本章首先描述系统页面组合，再阐述系统各个子模块的运行控制流程，最后说明系统各个模块的运行时间。

4.1 运行页面模块组合

健康论坛场景模式：

此模式如下图所示。

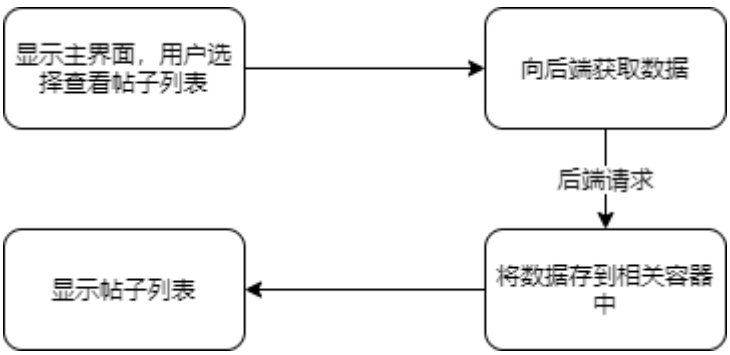


4.2 运行控制

健康论坛控制逻辑对用户的响应如下：

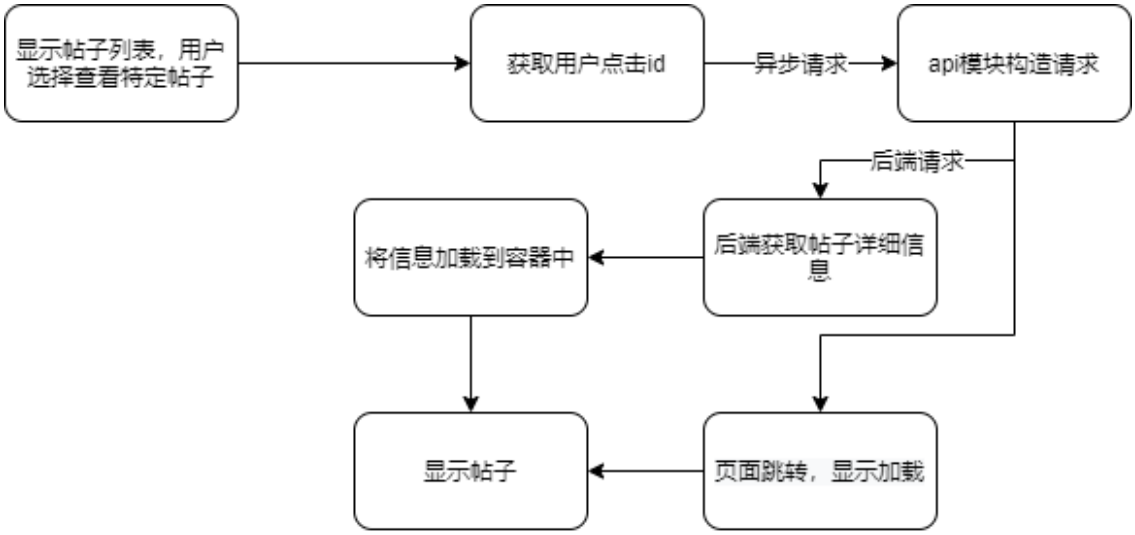
- 查看帖子列表
  index.vue调用api模块导出的获取帖子列表接口
  api模块构造请求，向后端发送请求以获取帖子列表
  在请求返回后，触发的方法会在index中的相关容器中填入获取到的数据

  对应流程图如下：



- 查看帖子内容
  index.vue获取用户所点击的帖子id，根据id向api模块发送查询详细数据的异步请求
  api模块构造请求，向后端发送请求以获取帖子详细信息
  index.vue携带部分源数据进行页面跳转，进入到药品详情页，展示加载动画
  请求返回后，触发的方法会在详情页中的相关容器中填入获取到的数据

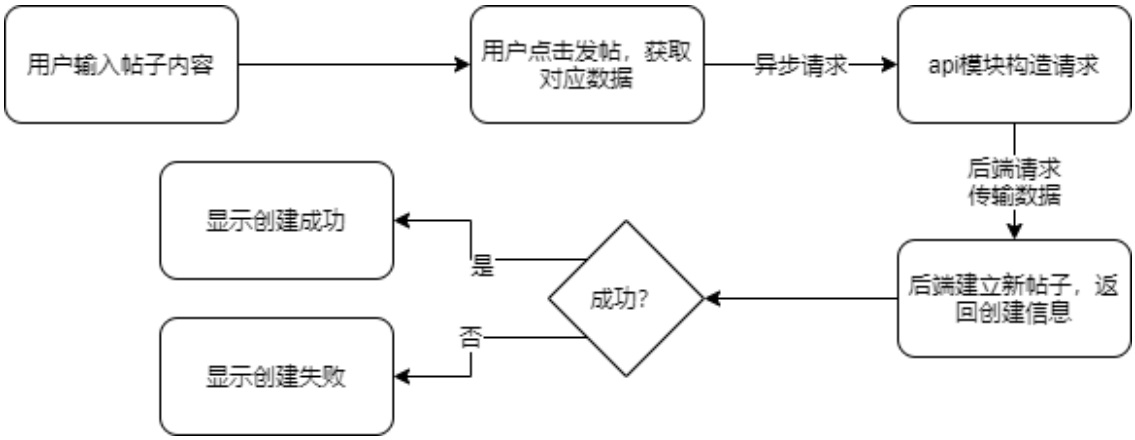
  对应流程图如下：



- 发帖

  用户点击发布的按钮后，index.vue 触发获取用户所输入的帖子内容，向 api 模块发送发帖的异步请求
  api模块构造请求，向后端发送请求同时传输数据
  请求返回后，根据返回内容向用户显示发帖成功或失败

  对应流程图如下：

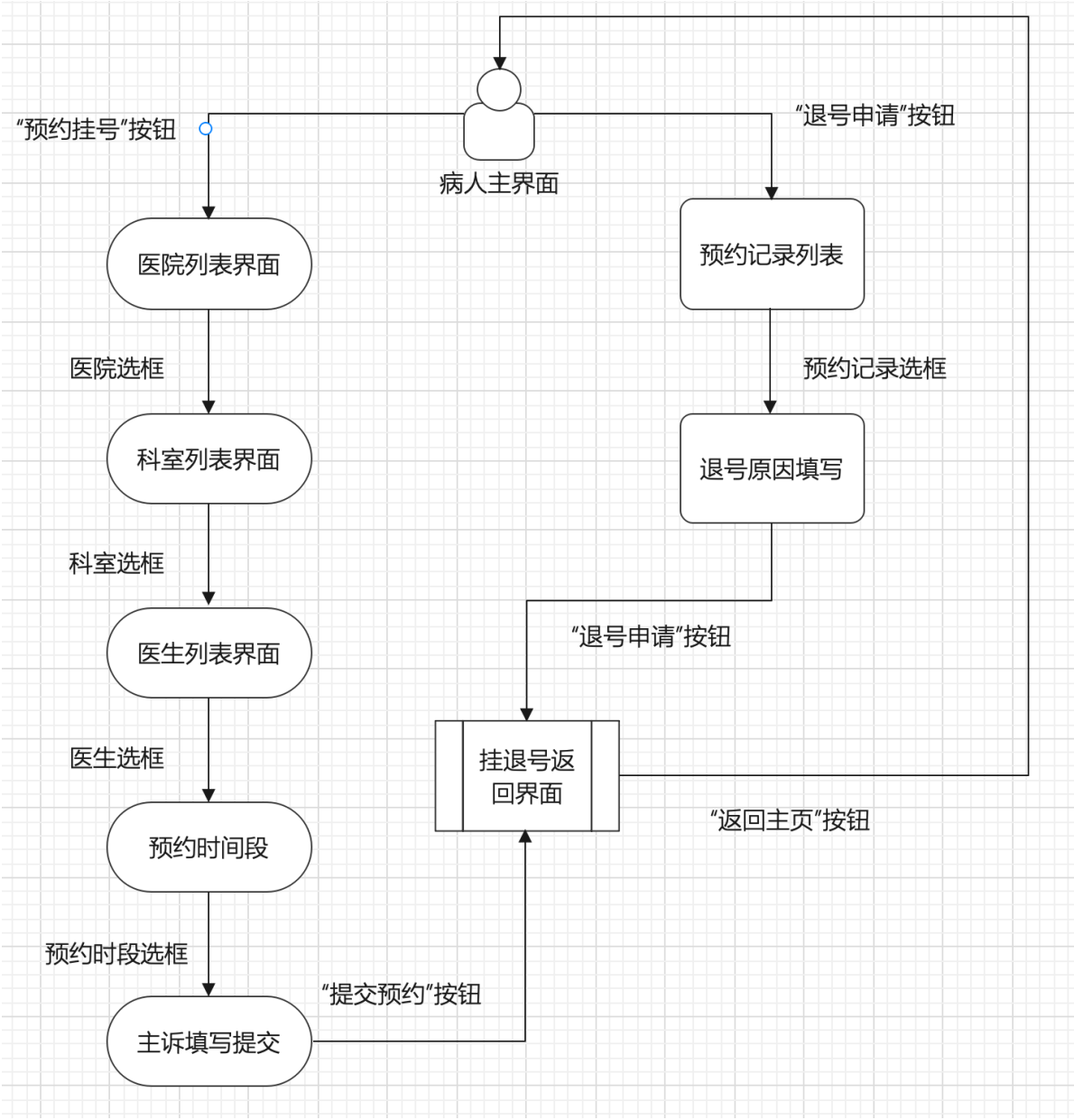


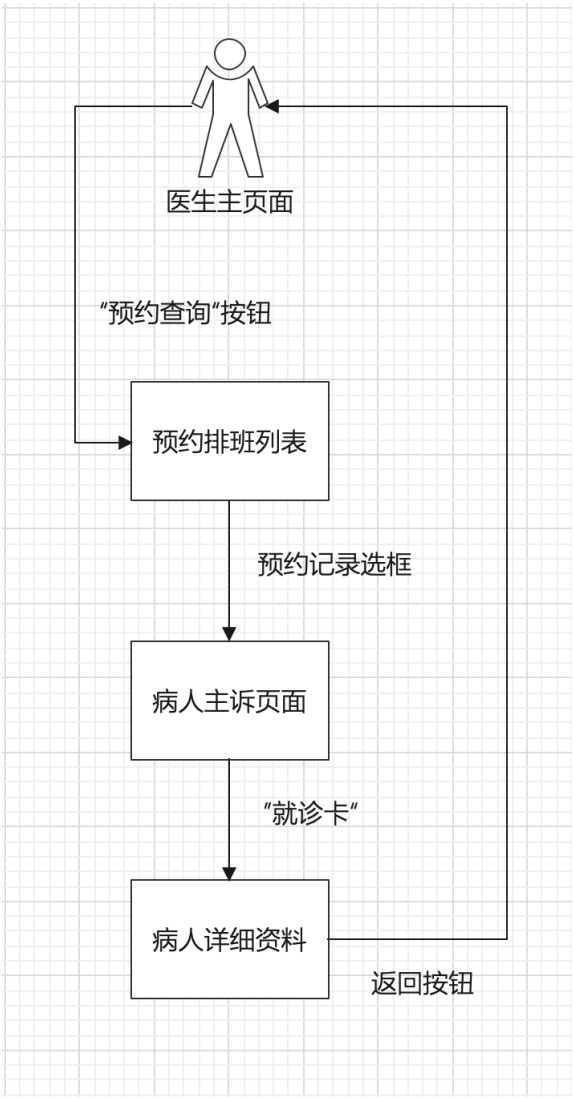
4.3 运行时间

- 单个用户在线时，应保证响应用户动作时间小于1秒，信息检索响应时间小于2秒。
- 多个用户同时在线时（至少支持500用户），应保证响应用户动作时间小于2秒，信息检索响应时间小于5秒。

挂号预约模块

4.1 模块组合





病人挂号与退号子模块组合见上图；

医生预约查询子模块见左图；

4.2 运行控制

预约子系统模块主要控制流程

- 主程序运行
- 等待用户交互动作
- 根据用户动作调用子模块

1. **患者预约挂号子模块**
    - 主程序运行
    - 用户点击“预约挂号”按钮
    - 向数据库请求医院列表并以选框展示
    - 用户点击医院选框
    - 向数据库请求医院可选科室列表并以选框展示
    - 用户点击科室选框
    - 向数据库请求科室医生列表并以选框展示
    - 用户点击医生选框
    - 向数据库请求医生有余量的时间段并以选框展示
    - 用户点击时间段选框
    - 用户填写主诉信息并点击提交按钮
    - 向数据库提交预约记录并等待响应
    - UI显示预约成功界面或返回预约失败原因

2. **患者退号子模块**
    - 主程序运行
    - 用户点击“我要退号”按钮
    - 向数据库请求返回患者预约列表
    - 用户选择退号选框
    - 用户输入退号原因
    - 请求数据库删除对应记录并等待响应
    - UI显示退号成功或退号失败原因
3. **医生排号列表子模块**
    - 主程序运行
    - 用户点击“查询预约排号”按钮
    - 请求数据库返回医生名下的预约列表，并以选框形式展示
    - 医生点击预约记录
    - 向数据库请求预约病人病例、就诊卡、主诉等详细信息
    - UI显示病人详情

4.3 运行时间

主要有患者预约挂号、退号和医生查看排号等子模块，下面分别描述。

1. 患者预约挂号子模块

   模块对用户按钮与输入交互的响应不应大于0.5秒；

   数据库请求医院、科室、医生列表以及可预约时段的耗时不超过5s；

   数据库响应新增记录耗时不超过1s；
2. 患者退号子模块

   模块对用户按钮与输入交互的响应不应大于0.5秒；

   数据库请求患者预约记录的耗时不超过5s；

   数据库响应删除记录耗时不超过1s；
3. 医生排号列表子模块

   模块对用户按钮与输入交互的响应不应大于0.5秒；

   数据库请求医生名下的预约列表耗时不超过5s；

   数据库请求病人病例、就诊卡等详细信息耗时不超过15s；

   UI显示病人详情响应时间不超过0.5秒；

Chapter5. 各模块详细设计

电子健康档案模块

5.1 功能性能实现情况

电子健康档案模块已经实现了最基本的功能，包括基本信息浏览，用户点击查看过敏药物、遗传疾病等更详细的基本信息，进入既往病历查询界面，选择病历中的某一个病案查看详细信息，跳转回到电子健康档案模块首页。页面加载时间平均在5s以内，响应用户点击时间平均在3s以内。

5.2 UI设计结构

电子健康档案模块主要页面有用户基本信息页和既往病历展示页。

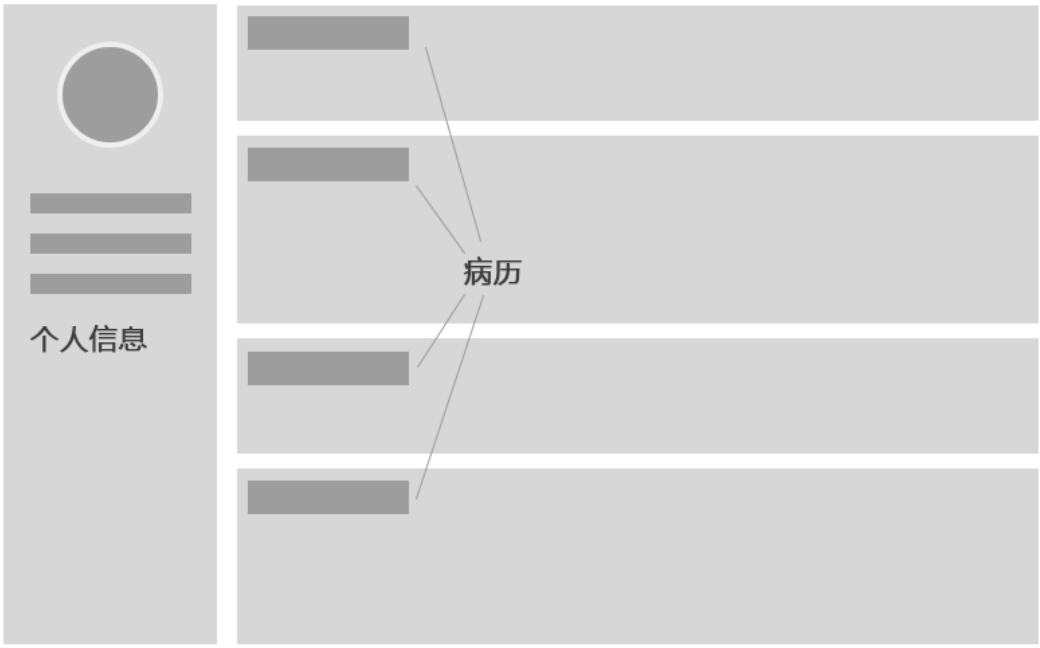
- 用户个人信息页UI结构

前端用到的数据主要包括用户电话号码、姓名、邮箱、性别、身高和体重，以及用户的头像。

```
data() {
    isLogin: false
    phone_number: null
    name: null
    email: null
    gender: null
    height: null
    weight: null
    portrait: null
}
```

- 既往病历展示页UI结构

  既往病历展示页由多个病案构成，结构大致如下：



左侧边栏显示简要个人信息和头像，右侧按日期顺序显示病案简介。若病案数量较多，超出屏幕的容纳范围，则在窗口右侧显示滑动条。

点击病案位置(即右侧浅灰色矩形区域)即可查看病案的详细信息。

5.3 功能设计结构

电子健康档案模块主要功能有用户基本信息展示、既往病历查询、病案详细信息展示。

- 基本信息展示实现方法

  点击按钮，进入用户个人信息页之后，首先获取用户的个人信息。用户登录之后，可以获得其手机号等信息。根据登录信息，使用恰当的url对后端发起请求。如，我们可以使用HTTP客户端`axios`，对后端请求用户个人信息：

  ```
  axios
    .get(/*url*/) /* 获取数据 */
    .then(response => (this.userInfo = response.data.userInfo))/*服务器反馈*/
    .catch(error => console.log(error)) /* 错误信息处理*/
  ```

  将获得的信息在前端界面进行展示。

- 既往病历查询实现方法

  首先，根据用户的`authType`信息，判断用户的身份。仅当`authType = 1`时，即用户的身份为病人，既往病历栏中才有内容。

  导航栏中既往病历的设置如下，点击之后，跳转到既往病历页面：

  ```
  <router-link class="..." :to="{ path: '/med_history'}" tag="li">
      既往病历
  </router-link>
  ```

  同时，对后端发出请求，在数据库中获取用户的病历内容。

云端药房模块

5.1 功能性能实现情况

云端药房模块已经实现了最基本的功能，包括显示主页面显示药品列表，用户选择药品类别，用户根据药品名称进行搜索，用户点击药品后跳转到药品详情子页面，用户点击返回跳转到药房主页。同时页面加载时间平均在2s以内，响应用户点击时间平均在1s以内。

5.2 UI设计结构

云端药房模块主要页面有药品展台页和药品详情页

- 药品展台页UI结构

```
<template>
    <view class="list_box">
        <!-- 左侧导航菜单-->
        <view class="left">
            <scroll-view scroll-y="true" :style="{ 'height':scrollHeight }">
                <!-- 单个菜单项，显示药品分类-->
                <view class="item" v-for="(item,index) in leftArray"
:key="index" :class="{ 'active':index==leftIndex }"
                 :data-index="index" @tap="leftTap">{{item.id}}</view>
            </scroll-view>
        </view>
        <!-- 右侧主要内容-->
        <view class="main">
            <scroll-view scroll-y="true" :style="{ 'height':scrollHeight }"
@scroll="mainScroll" :scroll-into-view="scrollInto"
             scroll-with-animation="true" @touchstart="mainTouch"
id="scroll-el">
                <!-- 为每一个分类循环赋值-->
                <block v-for="(item,index) in mainArray" :key="index">
                    <view class="item" :id="'item-'+index">
                        <!-- 本分类的名称-->
                        <view class="title">
                            <view>{{item.title}}</view>
                        </view>
                         <!-- 为每一种药品循环赋值-->
                        <view class="goods" v-for="(item2,index2) in
item.list" :key="index2" @tap="onItemClicked(index, index2)">
                            <image :src="item2.picPath" mode=""></image>
                            <view>
                                <!-- 药品名称-->
                                <view>{{item2.name}}</view>
                                <!-- 药品简略描述-->
                                <view class="describe">{{item2.description}}
</view>
                                <!-- 药品价格-->
                                <text class="text">￥{{item2.price}}</text>
                            </view>
                        </view>
                    </view>
                </block>
            </scroll-view>
        </view>
    </view>
</template>
```

- 药品详情页UI结构

```
<template>
    <transition name="medicine-detail">
        <view class="medicine">
            <!--规定页面滑动方式-->
            <scroll-view :scroll-top="scrollTop" scroll-y="true"
class="scroll-Y" @scrolltoupper="upper" @scrolltolower="lower"
@scroll="scroll">
                <!--药品详情页容器-->
                <view class="medicine-wrapper">
                    <!--药品简略内容-->
                    <view class="medicine-content">...</view>
                    <view class="split"></view>
                    <!--药品详细信息-->
                    <view class="description-content">...</view>
                </view>
            </scroll-view>
        </view>
    </transition>
</template>
```

5.3 功能设计结构

云端药房模块主要功能有药品列表展示，药品搜索，药品详细信息展示

- 列表展示实现方法

```
<script>
    //从api模块中获取静态资源的方法
    import {
        getStatic
    } from "../../fetch/api.js"
    export default {
        //为页面绑定默认元素
        data() {
            return {...}
        },
        onLoad() {
            /*获取系统配置信息*/
            uni.getSystemInfo(...);
        },
        computed: {},
        mounted() {
            this.getListData();
        },
        methods: {
            /* 获取列表数据 */
            getListData() {...},
            //获取距离顶部的高度
            getScrollTop(selector) {...},
            /* 获取元素顶部信息 */
            async getElementTop() {...},
            /* 主区域滚动监听 */
            mainScroll(e) {...},
```

```
            /* 主区域触摸 */
            mainTouch() {...},
            /* 左侧导航点击 */
            leftTap(e) {...},
            /*药品点击监听事件*/
            onItemClicked(index, index2) {...}
        }
    }
</script>
```

- 详细信息实现方法

```
<script>
    //从api模块中获取静态资源的方法
    import {
        getStatic
    } from "../../fetch/api.js"
    export default {
        //为页面绑定默认元素
        data() {
            return {...}
        },
        onLoad: function(option) {
            //从静态资源中获取元素值
            getStatic("/pharmacy/data.json").then((res) => {...})
        },
        methods: {
            //监听页面滑动至顶部
            upper: function(e) {
                console.log(e)
            },
            //监听页面滑动至底部
            lower: function(e) {
                console.log(e)
            },
            //监听页面滑动
            scroll: function(e) {
                console.log(e)
                this.old.scrollTop = e.detail.scrollTop
            },
        },
    }
</script>
```

体检预约模块

5.1 功能性能实现情况

健康检测模块的基本功能已经完成，包括显示体检/核酸检测可供预约的医院、日期，显示可供预约的时段，完成预约，显示预约记录，显示报告，修改预约时段的余量等，部分功能还有待完善。系统对用户的操作响应时间不超过1s，与后端交互时间不超过2s，刷新渲染页面不超过2s

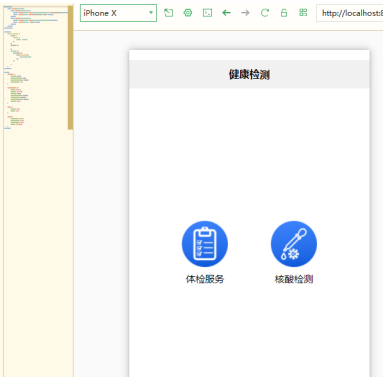
5.2 UI设计结构

健康检测模块主要有主入口、体检/核酸检测预约界面、预约时段选择界面、查看报告界面

主入口:

主入口设置体检和核酸检测服务的两大入口。

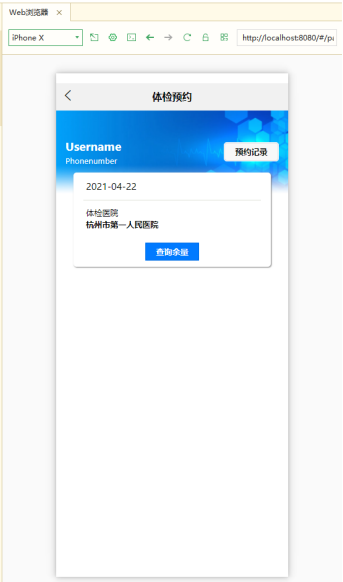
```vue
<template>
    <view class="content">
        <view class="switch-item">
            <image class="logo" src="/static/exam_phy.png" @click="gotoPhy"></image>
            <text class="title" @click="gotoPhy"> 体检服务 </text>
        </view>
        <view class="switch-item">
            <image class="logo" src="/static/exam_covid.png"></image>
            <text class="title"> 核酸检测 </text>
        </view>
    </view>
</template>

<script>
    export default {
        data() {
            return {
                title: 'Hello'
            }
        },
```



体检/核酸检测预约界面

此界面供用户选择日期与医院。加载此界面时，医院列表需要从后端获取。

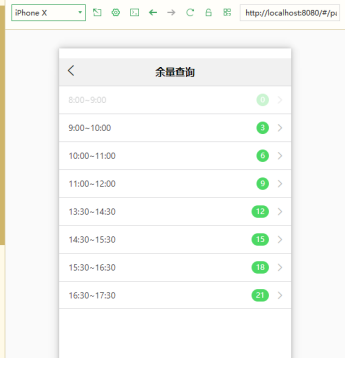
```vue
<template>
    <view class="page">
        <view class="title">
            <view class="Info">
                <text style="color: #FFFFFF ; font-size: larger; font-weight: 700;">  Username </text>
                <text style="color: #FFFFFF ;font-size: smaller;"> Phonenumber </text>
            </view>
            <button size="mini" style="border-width: 3rpx; border-color: #ffffff; margin-right: 15px;" type="default" @c
        </view>
        <view class="content">
            <view class="datesel">
                <view style=" margin-left: 20px; margin-top: 10px; margin-bottom: 10px;" @click="onShowDatePicker('date'
            </view>
            <view class="divider"/>
            <view class="uni-list">
                <view class="uni-list-cell" style=" margin-left: 20px; margin-top: 10px; margin-bottom: 10px;">
                    <view class="uni-list-cell-left" style="font-size: small;">
                        体检医院
                    </view>
                    <view class="uni-list-cell-db" style="font-size: small; font-weight: 700;">
                        <picker @change="bindPickerChange" :value="index" :range="hospitals">
                            <view class="uni-input">{{hospitals[index]}}</view>
                        </picker>
                    </view>
                </view>
            </view>
            <button size="mini" style="align-self: center; border-radius: 1rpx; border-color: #ffffff;" type="primary" @
        </view>
        <mx-date-picker :show="showPicker" :type="type" :value="value" :show-tips="true" :begin-text="'入住'" :end-text="
    </view>
</template>
```



预约时段选择界面

此界面供用户选择预约的时段。余量为0的时段不可选择。

```vue
<template>
    <view>
        <uni-list>
            <uni-list-item  v-for="(val, key) in remainder" :key="key" link="" @click="onClick"
             :title="convertText[key]" showArrow show-badge="true" :badge-text="val.toString()"
             :disabled="val<=0">
            </uni-list-item>
        </uni-list>
    </view>

</template>

<script>
    import {fake_fetchGet, fake_fetchPost} from "../../fake_backend.js";
    export default {
        components: {

        },

        data() {
```



查看报告界面

此界面供用户查看预约的历史和查看生成报告。

5.3 功能设计结构

健康检测模块的主要功能是供用户预约体检或核酸检测，也需要给医疗端提供修改预约时段余量的功能。功能中没有算法的问题，都是与数据库的交互，由后端完成，此处展示等同的数据库操作（数据表定义请见第6章）。

获取医院列表：

```
select distinct hospital from healthguide_exam_covid_capacity;
```

查询时段余量：

```
select section, remainder from healthguide_exam_covid_remainder
where hospital=XXX, appoint_date=XXX;
```

新增预约信息：

```
insert into healthguide_exam_covid_appointment (user_phone, hospital,
appoint_date, section)
values (XXX, XXX, XXX, XXX);
```

查询预约历史：

```
select appoint_id, hospital, appoint_date, section
from healthguide_exam_covid_appointment
where user_phone=XXX;
```

修改余量：

```
update healthguide_exam_covid_remainder set remainder=XXX
where hospital=XXX, appoint_date=XXX, section=XXX;
```

健康论坛模块

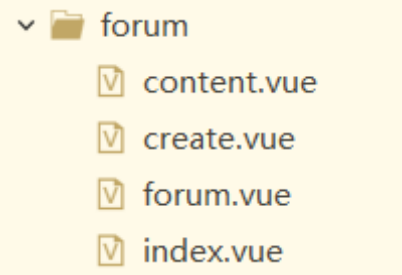
5.1 功能性能实现情况

健康论坛模块目前实现了交流空间的初步UI布局和简单交互设计，具有进入交流空间、查看讨论贴详情、加载更多讨论贴、进入发布新讨论贴的编辑页面等交互功能。不过，由于与程序后端尚未进行有效对接，交流空间内的所有讨论贴均为从本地JS文件中读取，因而无法有效衡量本模块的性能。

5.2 UI设计结构

健康论坛模块主要包括交流空间和在线问答两个子模块。目前只实现了交流空间的初步UI布局设计，而图标、字体字号、颜色风格尚未完善，与后端的交互接口也没有进行有效对接，故此处仅使用本地JS文件中储存的模拟讨论贴数据，展示初步的布局和交互。

到撰写本报告时，初步建立的页面结构如下：



健康论坛入口处的初步界面如下：

相关代码如下：

这一部分的代码较为简单，只需要将两个按钮的点击事件与部分的navigateTo函数相关联，并将目标地址作为参数传递给navigateTo函数即可。而后，navigateTo函数会根据目标地址自动导航到目标页面

交流空间

讨论贴列表页UI布局如下图所示：

因与后端的对接还未完成，本次demo中的数据主要从一个嵌入项目中的JS中读取。所以，首先需要导入数据。

```
import user_data from '../../data.js';
```

而后，从导入的数据中取15条出来作为初始展示的讨论贴。

```
topics: user_data.data.data.slice(0,15)
```

通过@click关键字，将讨论贴的点击事件与载入新页面的函数相关联。

```
<view v-for="(item,i) in topics">
    <view class = "topic" @click="seeDetails(i)">
```

其中i为被点击讨论贴在topics数组中的下标。待与后端相对接后这里传入的参数应该改成讨论贴在数据库中的ID

seeDetails函数的功能为导航到内容详情页，并将讨论贴ID作为参数传入

```
seeDetails: function(i) {
    uni.navigateTo(
        url: './content?topic_id' + i,
        ......
    )
}
```

而“加载更多”按钮则和loadMore函数相绑定，在demo中只需要在topics变量新加入15个展示的主题贴即可

```
<button class="load-button" @click="loadMore()">加载更多</button>
```

```
loadMore: function() {
    this.topics=user_data.data.data.slice(0, this.topic.length+15);
}
```

而等与后端的对接完成后，此部分的代码逻辑应该为向后端请求发送接下来的15条讨论贴的标题、作者昵称和头像等数据，并拼接到原来的讨论贴数组之后。

点击右上角的“+”按钮后，程序会导航到发布新讨论贴的编辑页面，页面布局如下图所示：

而点击任意讨论贴，即可进入详情页面查看讨论贴主题内容和回复情况。讨论贴详情页布局如下：

该页面的实现逻辑较为简单，只需要根据上层页面传递的主题贴ID信息载入其标题、作者、内容和回复等相关信息，再用从Vue中继承的v-for关键词将回复信息展开显示即可。待与后端的对接完善后，应该先根据主题贴ID向后端数据库发送请求，得到相关数据后再进行展示。

在线问答

在线问答子模块的编写尚未开始，但大致上可以借鉴交流空间的设计方法来设计在线问答区的UI。值得注意的是，在线问答区应有选择不同类型疾病的选项卡，以便用户能迅速找到与特定病种相关的问答信息。而问答详情页面中，也应该高亮显示来自经过认证的专业医师的回答，以突显其权威性。

5.3 功能设计结构

由于尚未与后端进行对接，这里的多数功能都无法展示具体的实现方法，只能描述设想中的功能实现方法。

1. 讨论贴或问答贴及其回复列表展示：

   前端程序向后端发送请求，收到讨论贴/问答贴的相关信息后，按照访问次数由高到低排序，使用v-for关键词进行展开并显示即可。点击进入讨论贴/问答贴后，同样可以按照回复的点赞数量/权威程度/回复时间进行加权排序，而后用v-for关键词进行列表展开显示。

2. 讨论贴或问答贴搜索

   前端程序以搜索关键词为参数向后端发送请求，后端根据关键词从数据库中检索标题、内容或回复中包含有关键词的讨论贴或者问答贴，将其信息返回给前端，随后前端即可加以显示。

3. 点赞/按踩

   进入讨论贴或者问答贴后，若对其内容或者部分回复有赞同/反对的情绪，则可以点击点赞/按踩键。点击后，前端程序会以讨论贴/问答贴/回复的ID以及点赞/按踩为参数向后端发送请求，而后端在接收到请求后，即可对数据库中该讨论贴/问答贴/回复的赞踩信息进行修改、更新

4. 发布新贴/新回复

   编辑好新贴/新回复的信息后，点击提交按钮，前段程序就会以作者ID、新贴内容/回复信息为参数向后端发送请求，而后端在接收到请求后，则会向相应的数据库中插入新的值，从而完成了新贴/新回复的发布。随后前端程序也会刷新页面，以显示刚刚发布的贴子/回复。

5. 举报

   发现讨论贴/问答贴中有违规内容时，用户可以点击举报按钮并选择违规种类、填写违规信息，而后前端会以贴子ID、举报内容等相关信息为参数将请求发送至后台。后台会在一个数据库内维护单独的举报信息队列，一便工作人员能够延时处理这些违规举报。举报得到回复后，相关的反馈信息也会回复给原用户。

挂号预约系统

5.1 功能性能实现情况

系统目前实现三个子功能模块，分别是预约挂号，退号、医生端的记录查询模块。在预约挂号模块的查询界面，用户可以通过个人信息查询预约记录表单以及查询不同的医院科室等预约情况的大致状况。在挂号界面，用户需要输入自己的详细信息以及所需申请的科室、医生、时间段等信息。若果符合预约的要求，系统则会新增一个预约信息，挂号结束，同时用户可以选择查询自己的预约记录。

在退号模块中，用户可以在查询界面输入预约的信息来进行简单的预约记录表查询。在退号界面，用户可以输入自己预约的信息以边继续宁预约记录的筛选，之后如果选择退号，系统会根据当前的信息判断并在允许的情况下删除该条预约记录

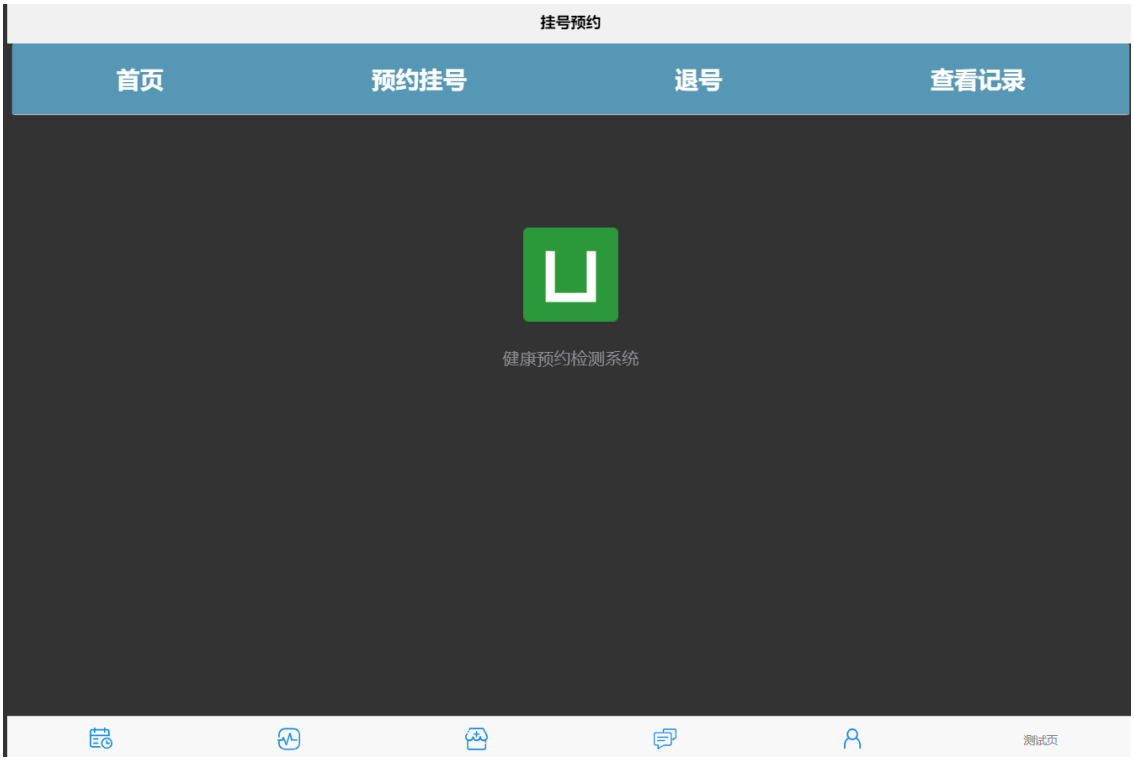
在医生记录查询模块，在查询界面，医生可以查看当前的挂号列表、病人预约号码等信息。同时医生可以输入某个病人的ID或者就诊卡ID来随时查看改名病人的详细信息

在性能要求上，系统确保用户在登陆验证后能够在较短时间内能够访问所属权限下的系统资源并能够正常使用系统所提供的功能，响应时间应小于1s，加载时间应在7s以内。页面设计保持合理简约，同时可以根据使得用户能够在较短时间内了解并使用。

5.2 UI结构设计

预约挂号模块主要包括预约、退号、预约查询和医生端预约板块。目前开发进度比较少，仅仅开发了病人端的预约子模块UI；目前开发的界面比较少，界面结构如图：

- 病人端预约子模块UI



首页内容如图所示。通过点击导航栏中的不同栏目可以跳转到不同的功能区域。

```
<template>
    <view class="content">
        <!--
        <button @click = "appointment">{{title1}}</button>
        <button @click = "withdraw">{{title2}}</button>
        <button @click = "show">{{title3}}</button>
        -->
        <nav  v-on:click.prevent>
            <!-- 当菜单上的链接被点击时，我们调用了 makeActive 方法，该方法在 Vue 实例中创建。 -->
            <a href="#" class="home" @click="jump_appointment">首页</a>
            <a href="#" class="projects" @click="jump_register">{{title1}}</a>
            <a href="#" class="services" @click="jump_withdraw">{{title2}}</a>
            <a href="#" class="contact" @click="jump_show">{{title3}}</a>
        </nav>
        <image class="logo" src="/static/logo.png"></image>
        <view class="text-area">
            <text class="title">{{title}}</text>
        </view>
    </view>
</template>
```

具体UI代码如上图所示，主要设计了nav导航栏，并设定针对性的样式。当用户点击不同的栏目，会触发不同的click动作，进行页面跳转。

后续预计在当前页面的其余部分展示公益标语或者医院的宣传广告，并实现轮转。

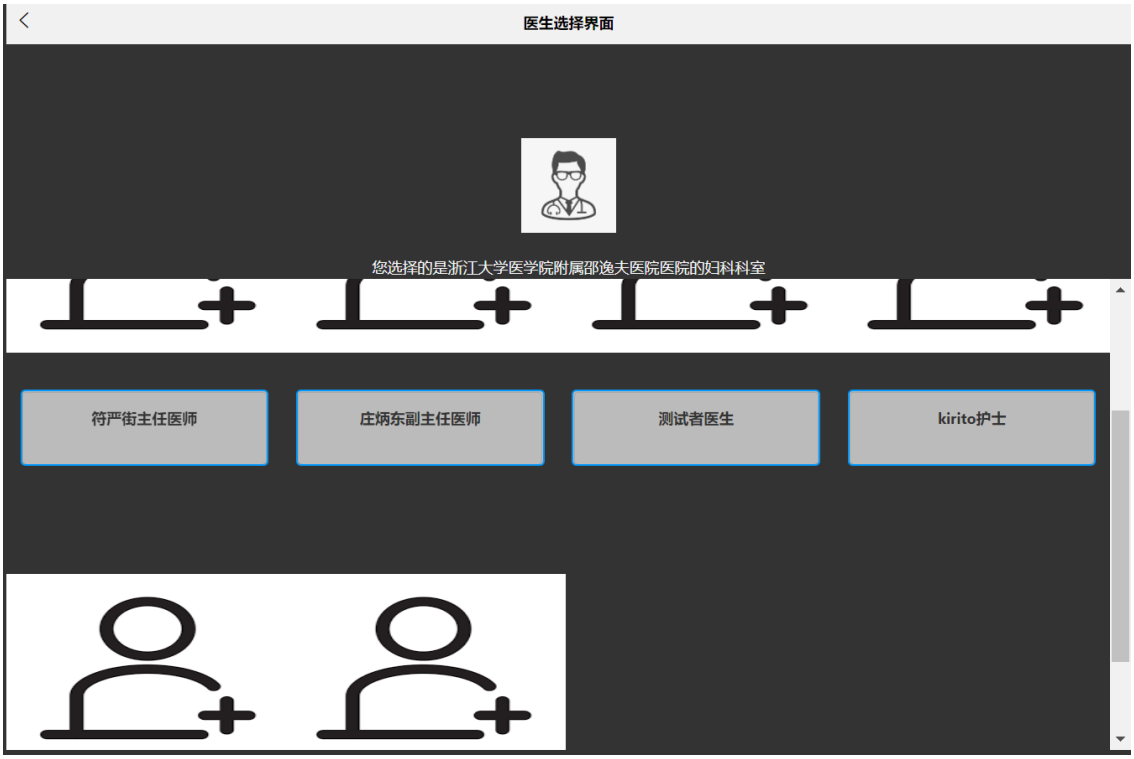
- 预约医院和科室选择界面

register界面可以选择需要预约的医院和科室。目前制作成下拉框的形式。当前没有和后端进行对接，因此目前的数据是从前端自己设计的json数据对象中获取并展示的。选择完医院和科室之后，跳转到医生选择界面。

```
<template>
    <view class="content">
        <view class="text-area">
            <text class="title">{{title}}</text>
        </view>
        <image class="logo" src="/static/icon.png"></image>
        <text class = "title">选择医院</text>
        <select v-model = "hospitals">
            <option v-for = "item in array" >{{item.name}}</option>
        </select>
        <text class = "title">选择科室</text>
        <select  v-model = "rooms">
            <option v-for = "items in room_array">{{items.name}}</option>
        </select>
        <button @click = "jump_doctor">确认选择</button>
    </view>
</template>
```

目前仍在进一步考虑当前页面的展示形式，当科室较多的时候，简单的下拉框可能不适合选择，同时也需要考虑微信小程序形式开发下信息的展示方式，最终给出一套方案。

- 医生选择界面



当跳转到医生界面的时候，会获取当前在register页面选择的医院和科室数据，并在界面中进行说明展示。之后会采用滑动条的形式将该医院科室的所有的医生的信息展示出来。

```
<template>
    <view class="content">
        <image class="logo" src="/static/doctor.png"></image>
        <view class="text-area">
```

```
            <text class="status">您选择的是
{{JSON.parse(decodeURIComponent(this.$route.query.hospital))}}医院的
{{JSON.parse(decodeURIComponent(this.$route.query.room))}}科室</text>
        </view>
        <scroll-view scroll-y style="height: 500px;" class = "vui">
            <!--  style="width: 60%; height: 70%; position:
absolute;left:30%;"  -->
            <view class="user" v-for = "item in array" >
                <img src="/static/user.png" alt="">
                <button style="width: 90%; height: 27%; left: 5%;background-
color:#BBBBBB;">{{item.name}}{{item.reputation}}</button>
            </view>
        </scroll-view>
    </view>
</template>
```

5.3 功能设计结构

目前实现的功能主要包括界面跳转、参数传递等比较基础的功能，还没有和后端具体对接。

界面跳转：

```
methods: {
            jump_register: function(){
                        this.$router.push( {path:
'/pages/registration/register'});
                    }
        }
```

采用vue2的路由跳转方式，如图所示的方式，当用户点击某一个按钮或者导航栏，触发对应的click函数，修改路由路径。

参数传递

首先要获取待传递的参数。

```
<select  v-model = "rooms">
            <option v-for = "items in room_array">{{items.name}}</option>
        </select>
```

当用户在下拉框中选择对应的医院或者科室的时候，通过v-model能够在rooms中获得用户当前选择的科室的值。

```
jump_doctor:function()
            {
                this.$router.push({
                    path: '/pages/registration/doctor',
                    query:{

hospital:encodeURIComponent(JSON.stringify(this.hospitals)),
                        room:encodeURIComponent(JSON.stringify(this.rooms))
                    }
                });
            }
```

在跳转的时候，通过query传递参数，这样在新的页面进行刷新的时候参数不会丢失。将获取到的参数传递到新的页面，因为query采用GET的url传值，所以最好对参数进行加密处理，在目标页面中进行解密。

```
<view class="text-area">
            <text class="status">您选择的是
{{JSON.parse(decodeURIComponent(this.$route.query.hospital))}}医院的
{{JSON.parse(decodeURIComponent(this.$route.query.room))}}科室</text>
        </view>
```

在目标页面中，通过this.$route.query可以获取到对应的参数，对这些参数进行解密并展示就可以了。

Chapter6. 数据结构详细设计

电子健康档案模块

用户个人信息

person_info(user_phone, auth_type, user_name, user_email, user_gender, user_height, user_weight, user_ID_number)

病案信息

healthguide_healthrecord_case_record(case_id, patient_phone, create_time, hospital, doctor_name, doctor_phone, case_result, cost, medicine)

用户个人信息

字段	类型	是否为空	是否主键	备注
user_phone	varchar(40)	否	是	用户电话，作为用户唯一标识
auth_type	tinyint	否	否	用户权限等级：[0] 未登陆[1] 病人[2] 医生[3] 管理员[4] 高管
user_name	varchar(40)	否	否	用户姓名
user_email	varchar(40)	是	否	用户邮箱
user_gender	enum('male','female)	否	否	用户性别
user_height	varchar(10)	否	否	用户身高
user_weight	varchar(10)	否	否	用户体重
user_ID_number	varchar(40)	否	否	用户身份证号

病案信息

字段	类型	是否为空	是否主键	备注
case_id	varchar(40)	否	是	病案id,作为病案唯一标识符
patient_phone	varchar(40)	否	否	病人电话(标识病人id)
create_time	datetime	否	否	创建日期
hospital	varchar(60)	否	否	所在医院信息
doctor_name	varchar(40)	否	否	负责医生姓名
doctor_phone	varchar(40)	否	否	负责医生电话
case_result	varchar(80)	否	否	病案诊断结果
cost	varchar(10)	否	否	诊断费用
medicine	varchar(40)	否	否	推荐药品名词

云端药房模块

1）healthguide_exam_medicine_list——存储所有药品信息的总表

属性名	数据类型	not null	备注	其他
id	int	y	药品id	primary key，auto_increment
name	varchar(40)	y	药品名	/
cata	varchar(40)	y	商品分类	/
type	varchar(40)	y	药品类型	/
price	decimal(10,2)	y	药品价格	/
sIndication	tinytext	n	简要适用症	/
lIndication	text	n	适用症	/
usageAndDosage	text	n	用法用量	/
adr	text	n	不良反应	/
contraindications	text	n	禁忌	/

2）healthguide_phar_chart_list——存储购物车信息的总表、

属性名	数据类型	not null	备注	其他
phone	varchar(40)	y	电话	外键，参考 user_info(user_phone)
id	int	y	药品id	外键，参考 healthguide_phar_booth_medicine(id)
count	int	y	数量	/

3）healthguide_phar_order_list——存储订单信息的总表

属性名	数据类型	not null	备注	其他
order_id	int	y	订单id	primary key auto_increment
phone	varchar(40)	y	电话	外键，参考user_info(user_phone)
medicine_id	int	y	药品id	外键，参考 healthguide_phar_booth_medicine(id)
count	int	y	数量	/
create_date	date	y	订单创建时间	/
status	enum('未支付','已支付','取消','已完成')	y	订单状态	/

体检预约模块

本子系统包含体检预约和核酸检测预约两部分，两者所用的数据表的字段是相同的。

healthguide_exam_physical_capacity——设置每个医院的时段容量默认值：
(healthguide_exam_covid_capacity)

属性名	数据类型	not null	备注	其他
hospital	varchar(60)	y	医院名	primary key
default_capacity	int	y	各时段默认容量	

healthguide_exam_physical_remainder——设置每个医院每天各时段实时余量：
(healthguide_exam_covid_remainder)

属性名	数据类型	not null	备注	其他
hospital	varchar(60)	y	医院名	primary key (composite)
appoint_date	date	y	日期	primary key (composite)
section	int	y	时段	primary key (composite)
remainder	int	y	实时余量	

healthguide_exam_physical_appointment——预约信息：
(healthguide_exam_covid_appointment)

属性名	数据类型	not null	备注	其他
appoint_id	bigint	y	预约序列号	primary key, auto_inc
user_phone	varchar(40)	y	用户电话	key 'idx_phone'
hospital	varchar(60)	y	医院名	
appoint_date	date	y	预约日期	
section	int	y	预约时段	
create_time	datetime	y	记录创建时间	default current_timestamp

healthguide_exam_physical_report——储存体检报告：
(healthguide_exam_covid_report)

属性名	数据类型	not null	备注	其他
appoint_id	bigint	y	预约序列号	primary key
user_phone	varchar(40)	y	用户电话	key 'idx_phone'
report	MediumBlob	n	报告	default null

健康论坛模块

系统使用的数据库为MySQL，主要包含以下几个表：

- forum_post_topic
- forum_post_reply
- forum_post_topic_favorite
- forum_qa_question
- forum_qa_answer
- forum_qa_reply
- forum_qa_answer_favorite

forum_post_topic 存放所有主题帖的表

属性	数据类型	not null	备注	其他
topic_id	bigint	y	主题贴ID	primary key, auto_increment
session	varchar(10)	y	所属版块	/
title	varchar(40)	y	标题	/
author_name	varchar(40)	y	作者姓名	
author_id	varchar(40)	y	作者ID	foreign key(user)
content	text	y	内容	/
create_time	Datetime	y	创建时间	/
last_edit_time	Datetime	y	最后编辑时间	/
reply_cnt	int	y	回复数	/
view_cnt	int	y	浏览数	/
like_cnt	int	y	点赞数	/
dislike_cnt	int	y	点踩数	/

forum_post_reply 存放主题帖之下回复的表

属性	数据类型	not null	备注	其他
reply_id	bigint	y	回复ID	primary key, auto_increment
topic_id	bigint	y	主题帖ID	foreign key(forum_post_topic)
author_name	varchar(40)	y	作者姓名	
author_id	varchar(40)	y	作者ID	foreign key(user)
floor	int	y	楼层	/
content	text	y	内容	/
create_time	Datetime	y	创建时间	/
last_edit_time	Datetime	y	最后编辑时间	/
like_cnt	int	y	点赞数	/
dislike_cnt	int	y	点踩数	/

forum_post_topic_like_or_not 存放主题贴赞踩信息的表

属性	数据类型	not null	备注	其他
topic_id	bigint	y	主题贴ID	primary key,foreign key(forum_post_topic)
user_id	varchar(40)	y	赞踩的用户ID	primary key, foreign key(user)
like_or_dislike	bool	y	1表示赞，0表示踩	/

forum_post_reply_like_or_not 存放回复赞踩信息的表

属性	s数据类型	not null	备注	其他
reply_id	bigint	y	回复ID	primary key, foreign key(forum_post_reply)
user_id	varchar(40)	y	赞踩的用户ID	primary key, foreign key(user)
like_or_dislike	bool	y	1表示赞，0表示踩	/

forum_post_topic_favorite 存放主题贴收藏信息的表

属性	数据类型	not null	备注	其他
favorite_id	bigint	y	收藏ID	primary key, auto_increment
topic_id	bigint	y	主题贴ID	foreign key(forum_post_topic)
user_id	varchar(40)	y	收藏的用户ID	foreign key(user)
title	varchar(40)	y	标题	/
author_name	varchar(40)	y	作者	

forum_qa_question 存放提问的表

属性	数据类型	not null	备注	其他
question_id	bigint	y	问题ID	primary key, auto_increment
session	varchar(10)	y	所属版块	/
title	varchar(40)	y	标题	/
author_name	varchar(40)	y	作者姓名	
author_id	varchar(40)	y	作者ID	foreign key(user)
content	text	y	问题内容	/
create_time	Datetime	y	创建时间	/
last_edit_time	Datetime	y	最后编辑时间	/
answer_cnt	int	y	回答数	/
view_cnt	int	y	浏览数	/

forum_qa_answer 存放回答的表

属性	数据类型	not null	备注	其他
answer_id	bigint	y	回答ID	primary key, auto_increment
question_id	bigint	y	问题ID	foreign key(forum_qa_question)
author_name	varchar(40)	y	作者姓名	
author_id	varchar(40)	y	作者ID	foreign key(user)
content	text	y	回答内容	/
create_time	Datetime	y	创建时间	/
last_edit_time	Datetime	y	最后编辑时间	/
reply_cnt	int	y	回复数	/
view_cnt	int	y	浏览数	/
like_cnt	int	y	点赞数	/
dislike_cnt	int	y	点踩数	/

forum_qa_reply 存放回答之下评论的表

属性	数据类型	not null	备注	其他
reply_id	bigint	y	回复ID	primary key, auto_increment
answer_id	bigint	y	回答ID	foreign key(forum_qa_answer)
author_name	varchar(40)	y	作者姓名	
author_id	varchar(40)	y	作者ID	foreign key(user)
content	text	y	内容	/
create_time	Datetime	y	创建时间	/
last_edit_time	Datetime	y	最后编辑时间	/
like	int	y	点赞数	/
dislike	int	y	点踩数	/

forum_qa_answer_like_or_not 存放回答赞踩信息的表

属性	数据类型	not null	备注	其他
answer_id	bigint	y	回答ID	primary key , foreign key(forum_qa_answer)
user_id	varchar(40)	y	赞踩的用户ID	primary key , foreign key(user)
like_or_dislike	bool	y	1表示赞，0表示踩	/

forum_qa_reply_like_or_not 存放回复赞踩信息的表

属性	数据类型	not null	备注	其他
reply_id	bigint	y	回复ID	primary key , foreign key(forum_qa_reply)
user_id	varchar(40)	y	赞踩的用户ID	primary key , foreign key(user)
like_or_dislike	bool	y	1表示赞，0表示踩	/

forum_qa_answer_favorite 存放回答收藏信息的表

属性	数据类型	not null	备注	其他
favorite_id	bigint	y	收藏ID	primary key
answer_id	bigint	y	回答ID	foreign key(forum_qa_answer)
user_id	varchar(40)	y	收藏的用户ID	foreign key(user)
title	varchar(40)	y	标题	/
author_name	varchar(40)	y	作者	foreign key(user)

挂号预约系统

完成挂号预约功能时，病人先要选择具体的医院和科室信息，然后选择该科室的相关医生。在实现这个功能的过程中，医生的职务、信息、在固定时段的可预约余量都需要展示出来。因此需要查询存储有医院、医院对应科室、科室中对应医生、医生个人信息的一张表，同时，为了实现预约余量查询和具体的预约、退号操作，需要存储预约记录信息。

因此，目前主要涉及了两张表，如下所示：

表1：healthguide_appointment_register_hospital

名	类型	长度	非空	键	注释
hospital	varchar	60	非空		医院名
room	varchar	40	非空		医院科室
doctor_phone	varchar	40	非空	主键	医生id
doctor_info	varchar	255	非空		医生信息

建立该表的SQL语句如下：

```
create table `healthguide_appointment_register_hospital` (
  `hospital` varchar(60) not null comment '医院名' ,
  `room` varchar(40) not null comment '医院科室',
  `doctor_phone` varchar(40) not null comment '医生id' primary key,
  `doctor_info` varchar(255) not null comment '医生信息'
) engine=InnoDB default charset=utf8mb4 comment='各医院科室和医生信息';
```

```
insert into healthguide_appointment_register_hospital value
("浙江大学医学院附属第一医院", "儿科", "赵嘉成","主任医师");
//示例数据
```

表2：healthguide_appointment_register_record

名	类型	长度	非空	键	注释
appoint_id	int	11	非空	主键	预约时段id
patient_phone	varchar	40	非空	主键	病人id
doctor_phone	varchar	40	非空	主键	医生id
room	varchar	40	非空		科室
hospital	varchar	40	非空		医院
appoint_date	date_time	0	非空		预约日期

创建这张数据表的SQL语句如下：

```
create table `healthguide_appointment_register_record` (
  `appoint_id` int not null comment '预约时间段编号',
  `patient_phone` varchar(40) not null comment '病人ID',
  `doctor_phone` varchar(40) not null comment '医生ID',
  `room` varchar(40) not null comment '科室',
  `hospital` varchar(40) not null comment '医院',
  `appoint_date` date_time not null comment '预约日期'
  primary key (`doctor_phone`,`appoint_id`,`patient_phone`)
) engine=InnoDB default charset=utf8mb4 comment='医院预约记录';
```

系统为每一个医生开放10个预约时段，每个预约时段都有一定的容量，从早到晚用1-10的id进行标识。当用户选择某一个时段的时候，会自动转换成对应的id存储到预约记录中。查询剩余余量的时候，从预约记录中筛选该医生目前的预约记录，并返回不同时段的预约记录数，从而确定有余量的时段。

Chapter7. 软件容错报错设计

本章说明系统运行时可能出现的集中主要错误，然后提出相应的修复措施。

7.1 出错信息表

序号	出错情况	解决方案
1	服务器宕机	重启云服务器
2	服务器性能不足	租用配置更高的云服务器（2核1G）
3	程序崩溃	重启并优化程序
4	存储空间不足	增大服务器硬盘空间
5	页面乱码	更改软件编码格式

7.2 补救措施

- 如果遇到常规错误，那么可以按照上面的方式解决。
- 当程序出现较大问题时我们要第一时间启动备份程序。
- 如果无法启动备份程序那么只能暂停应用程序的服务并进行维护。
- 如遇其他特殊错误及时反馈给相关开发人员，以便于第一时间进行修复。

Chapter8. 软件维护、测试、验证设计

本章主要阐述系统版本控制，软件开发过程以及软件的测试方法。

8.1 软件版本控制

8.1.1 版本控制

该系统采用git版本控制软件进行代码的版本控制和数据备份。

git 在本地磁盘上就保存着所有有关当前项目的历史更新，处理速度快；git 中的绝大多数操作都只需要访问本地文件和资源，不用实时联网，适合分布式开发。

版本号控制策略：

1. 软件系统的版本号由指定人员确定，进行版本控制工作。
2. 各子系统的版本号独立。
3. 初版本号通常为1.0.0或0.0.0。
4. 当项目在进行了局部修改或bug修正时，主版本号和子版本号都不变，修正版本号加1。
5. 当项目在原有的基础上增加了部分功能时，主版本号不变，子版本号加1，修正版本号复位为0，因而可以被忽略掉。
6. 当项目在进行了重大修改或局部修正累积较多，而导致项目整体发生全局变化时，主版本号加1，子版本号和修正版本号复位0。

8.1.2 工作流程

我们使用 githubflow 来进行分支管理，个人在开发的时候需要经过以下六个步骤：

- 创建分支
  在特定的分支上实现自己的工作
- 添加提交
  将修改的内容提交的服务器
- 提出Pull请求
  Pull 请求开启了一个关于你的提交内容的讨论，
- 讨论和评估你的代码
  其他人来审查你的代码
- 部署
  部署来做最终的测试
- 合并
  在生产环境中得到验证，合并到主分支上。

GitHub flow 的核心优势在于其流程带来的自动化可能性，能够做到其它流程无法实现的检查过程，并极大简化开发团队的体力劳动，真正发挥自身的价值。

8.2 软件开发过程

在该软件开发过程中，我们采用敏捷开发的方法。

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法。在敏捷开发中，软件项目的构建被切分成多个子项目，各个子项目的成果都经过测试，具备集成和可运行的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

敏捷开发有以下主要特征：

- 敏捷开发方法是“适应性”而非“预设性”
- 敏捷开发方法是“面向人”而非“面向过程”

8.3 软件测试方法

软件测试，描述一种用来促进鉴定软件的正确性、完整性、安全性和质量的过程。在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。我们对软件做如下的测试。

8.3.1 软件测试进程

Alpha测试
Alpha测试通常是阶段性的开发完成后所开始进行，一直持续到进入Beta测试阶段前的阶段。Alpha测试是一种验证测试，在模拟的环境中以模拟的资料来运行。

在这个阶段中，通常是在开发单位由开发人员与测试的测试人员，以模拟或实际操作性的方式进行验证测试。

Beta测试
在系统测试中通常先进行Alpha测试以验证信息系统符合用户以及设计需求所期望的功能。当Alpha阶段完成后，开发过程进入到Beta阶段，由公众参与的测试的阶段。Beta测试可称为确认测试，在一个真实的环境中以实际的资料来运行测试，以确认性能，系统运行有效率，系统撤销与备份作业正常，透过测试让信息系统日后可以更趋完善。

8.3.2 软件测试阶段

单元测试
单元测试是对软件组成单元进行测试，其目的是检验软件基本组成单位的正确性，测试的对象是软件设计的最小单位：函数。
并且使用假资料测试不同状况下功能使用情况，单元测试还有助于开发人员编写更好的代码。
单元测试是基于code的:可读性、可测试性，它们与开发代码的构建方式密切相关。因此开发人员最清楚哪些测试最有意义。

集成测试
集成测试也称综合测试、组装测试、联合测试，将程序模块采用适当的集成策略组装起来，对系统的接口及集成后的功能进行正确性检测的测试工作。其主要目的是检查软件单位之间的接口是否正确，集成测试的对象是已经经过单元测试的模块。

系统测试

系统测试主要包括功能测试、界面测试、可靠性测试、易用性测试、性能测试。 功能测试主要针对包括功能可用性、功能实现程度（功能流程&业务流程、数据处理&业务数据处理）方面测试。

回归测试

回归测试指在软件维护阶段，为了检测代码修改而引入的错误所进行的测试活动。回归测试是软件维护阶段的重要工作，有研究表明，回归测试带来的耗费占软件生命周期的1/3总费用以上。

与普通的测试不同，在回归测试过程开始的时候，测试者有一个完整的测试用例集可供使用，因此，如何根据代码的修改情况对已有测试用例集进行有效的复用是回归测试研究的重要方向，此外，回归测试的研究方向还涉及自动化工具，面向对象回归测试，测试用例优先级，回归测试用例补充生成等。

- 测试原有功能
- 测试新加入的功能是否有副作用

Chapter9. 软件功能维护设计

9.1 功能需求满足度分析

电子健康档案模块

功能要求	是否已完成
1. 用户登录模块	是
2. 用户可查看个人信息	是
3. 病人可查看历史兵力	是
4. 病历数据统计功能	否

云端药房模块

功能要求	是否已完成
1. 显示药品列表	是
2. 根据名称搜索药品	是
3. 选择不同类别药品	是
4. 查看药品详情信息	是
5. 添加药品到购物车	有待完成
6. 查看订单	有待完成

体检预约模块

功能要求	是否已完成
1. 用户端能够进行核酸检测和体检的分时段预约	是
2. 用户端能够查看过往核酸检测和体检的报告	是
3. 医生端能够修改核酸检测和体检的各时段容量	是
4. 医生端能够发布核酸检测和体检的报告	是

健康论坛模块

功能	是否已完成
1. 能够浏览贴子	是
2. 能够浏览贴子回复	是
3. 能够发帖	否
4. 能够收藏贴子	否
5. 能够赞、踩贴子	否
6. 与问答相关的内容	否
7. 与个人信息相关的内容	否

挂号预约系统

功能	是否已完成
1. 病人能正常使用挂号	是
2. 病人能查看已挂号	是
3. 病人能正常使用退号	是
4. 医生能查看已挂号列表	是
5. 医生能查看排班情况	是
6. 医生能请假，系统自动排班	是

9.2 性能需求满足度分析

性能要求	是否已完成
1.页面平均加载时间小于5s	是
2.界面友好，操作简单	是
3.健壮性	是
4.可靠性	是
5.安全性	是
6.可扩展性	是
7.可维护性	是
8.开放性	是

Chapter10. 总结

本概要设计报告，主要基于本项目经核准的计划任务书及《“健康导航平台”用户需求分析报告》，首先说明了报告编写的背景和目的，定义了本报告中的一些常用专业术语及相关参考文献；然后阐述了系统实现的总体设计，包括系统的需求规定、总体模块组成、功能及性能分配、各模块原理及系统的运行环境；再描述了系统的接口设计，包括用户接口、外部接口、内部接口及界面设计；然后说明了系统的运行设计，包括运行模块组合、运行控制、运行时间；接着阐述了系统各模块的详细设计，对每一个小模块都描述了其功能性能实现情况、输入输出、逻辑流程以及算法描述；在系统容错报错设计一章中，主要说明了软件维护、测试和验证设计，最后在软件需求满足度分析中，分析了功能需求满足度和性能需求满足度。本概要与详细设计报告，将为此后系统实施、测试、验收提供依据及技术路线支持。