



浙江大學
Zhejiang University

【健康导航平台预约系统】

需求分析报告

主要撰写者：李右

子模块撰写者：C1全体组员

初审：赵嘉成

初稿撰写日期：2021.04.10

第一次修订日期：2021.04.10

目 录

第 1 章 引言	1
1.1 编写目的	1
1.2 相关背景	1
1.3 定义部分	2
1.3.1 Linux	2
1.3.2 SQL	2
1.3.3 ASP.NET Core	2
1.3.4 Vue2	2
1.3.5 用例	2
1.3.6 用例图	3
1.3.7 状态图	3
1.3.8 数据流图	3
1.3.9 CRC 卡	3
1.3.10 类图	3
1.4 参考资料	3
第 2 章 任务概述	5
2.1 系统概述	5
2.2 系统组成	5
2.3 用户特点	5
2.4 假定和约束	5
第 3 章 需求规定	7
3.1 功能要求	7
3.1.1 挂号预约功能	7
3.2 性能要求	7
3.3 接口要求	7
3.4 界面或人机接口要求	8
3.5 系统质量要求	8
3.6 质量控制要求	8
3.7 设备验收要求	8
第 4 章 用户场景以及状态图	9
4.1 用例	9
4.1.1 用例 1	9

4.1.2	用例 2	9
4.1.3	用例 3	9
4.1.4	用例 4	9
4.1.5	用例 5	9
4.1.6	用例 6	9
4.2	对应用例图	9
4.3	状态图	9
4.3.1	病人	9
4.3.2	医生	9
4.4	类图	9
4.5	数据流图	9
4.5.1	level1 预约	9
4.5.2	level1 取消预约	9
4.5.3	level1 支付模块	9
4.6	CRC 卡	9
第 5 章	运行环境规定	10
5.1	服务端硬件配置需求	10
5.2	服务端软件依赖要求	10
5.3	客户端软硬件配置要求	10
5.4	变更控制规范	10
5.4.1	需求变更	11
5.4.2	变更控制	11
5.5	故障处理	11
5.6	故障确认	11
5.7	故障自动恢复	12
5.8	故障连锁诊断	12
第 6 章	总结	13

第 1 章 引言

本章首先说明本需求报告的编写目的，再论述本项目的背景，定义一些本报告涉及到的一些专业用语，最后列出与本报告相关的参考文献。

1.1 编写目的

本需求报告是为了构建一个健康导航完整开发体系而编写，描述了系统的整体架构，将会使用的框架和技术栈，对于功能需求、性能需求和其他一些非功能性需求进行了详细的描述，明确标识了各功能的实现过程，并且阐述了适用的范围和背景。为软件的设计和编码提供了基础性指导，便于软件开发过程中的版本控制和代码管理。同时本需求说明书也是项目完成之后验收工程的依据。具体的目的可以列为以下几点：

1. 描述健康导航完整开发的框架和约束，前端和后端使用的框架和技术栈，为后续软件开发提供参。
2. 描述健康导航完整开发的功能和操作流程，厘清业务逻辑，为系统的开发和设计提供基础思路。
3. 为项目之后的测试与验收提供参考依据，本文档也将为后续的软件工程概要设计、软件工程详细设计的实施提供依据。

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 and 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

1.3.1 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。

1.3.2 SQL

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

1.3.3 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。

1.3.4 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。

1.3.5 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。

1.3.6 用例图

用例图 (Use Case Diagram) 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。

1.3.7 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。

1.3.8 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。

1.3.9 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。

1.3.10 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

[2] Roger S Pressman. Software Engineering: A Practitioner's Approach. McGraw-Hill Education, 9 edition, 2020.

[3] 本文档所参考的软件开发标准。《计算机软件产品开发文件编制指南 (GB/T 8567)》《计算机软件文档编制规范 (GB/T 8567-2006)》《计算机软件需求说明编制指南 (GB/T 9385)》

[4] 本文档所参考的医疗系统开发要求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. 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.

第 2 章 任务概述

2.1 系统概述

健康导航平台以互联网平台作为基础，紧密关注当前疫情下互联网医疗的巨大潜力，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。健康导航平台将从五个模块进行服务，分别为挂号预约系统、电子健康档案、健康检测、在线药房和健康论坛。小组主要实现挂号预约子系统，该系统同时对普通用户（患者）以及医生等医疗端提供服务。对于广大患者用户，能够在预约挂号系统中完成预约操作，选择自己预约的时间、科室和医生，提交个人信息，填写主诉，缴纳费用，完成预约，也能够有一定的时间限制内取消预约并退款；对于医生，能够通过该系统获得病人针对自己的预约情况，同时能够自动完成针对个人的排班计划。

2.2 系统组成

预约子系统主要包括医生端和病人端两部分。满足医疗场所对于线上预约的时效性、便捷性、自动化等的要求。产品成型后可供诊所、小型医院、保健站等医疗场所使用。

2.3 用户特点

本软件的用户主要是一线医务人员，在日常工作中需要接触大量的患者，从而产生大量的文字、符号、图表、影像等资料。他们对于病历、病案等医疗文书的格式比较熟悉，同时具有良好的计算机软件操作能力，能够快速掌握较大型软件的操作；但是，他们通常不具备较深层次的计算机知识和编程能力，在软件发生错误时很难自行更正。因此，对本系统的维护人员有较高的要求，需要有很好的计算机系统知识及一定的编程能力，并对本系统的设计与实现都有较好的理解。

2.4 假定和约束

本系统开发需要专业数据库服务器一台、配有基本编程功能的 PC 高档微机若干台；需要 C++ Visual Studio、数据库编程人员、需要界面设计的美工人员。预计整个系统开发时间为 2 个月，试运行时间一周。可行的验证计划：针对各个环节的模型设计和界面预想结果，应给出与实际过程比较的精度分析过程，要求全系统模型稳定性不低于 80%，软件稳定，有效，配套文档齐全。

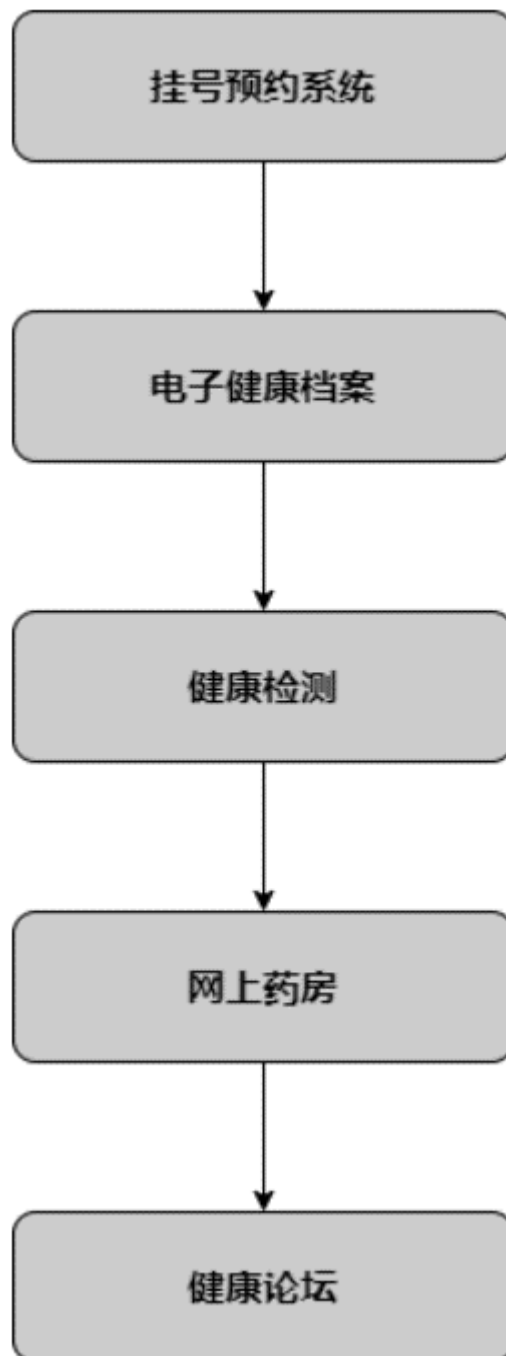


图 2.1 健康导航系统的模块组成

第 3 章 需求规定

本章将分别描述本项目的各种需求规定，包括：功能要求、性能要求、接口要求、界面或人机接口要求、系统质量要求、质量控制要求和验收要求。下面分别说明。

3.1 功能要求

本系统基于互联网平台，向民众提供在线挂号预约服务。具体功能说明如下：

3.1.1 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；

3.2 性能要求

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。

3.3 接口要求

一般要求：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。

扩展性要求：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。

3.4 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；

3.5 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

3.6 质量控制要求

由项目办组织进行软件需求分析评审，由承制方提交软件概要设计，并经过评审确认。在设计的过程中，项目办积极与承制方沟通，确认对方很好地明确该软件的需求，并对研制进度进行跟踪。软件编写完成后，总体必须参与软件的单元测试等软件测试；软件研发完成后，利用总体提供测试案例进行软件确认测试；承制方做软件研制总结，项目办组织评审确认。

3.7 设备验收要求

承制方经内部验收通过后，提前三天提交验收申请。项目办根据任务书，对软件开发过程中的版本管理、文档、各阶段的评审结论、测试结果和测试结论进行审查，召开验收评审会决定该产品是否通过验收。

第 4 章 用户场景以及状态图

4.1 用例

本节将描述系统的典型用例，包括病人的个人就诊卡管理，挂普通号和专家号、退号，以及医生层面查看个人排班、个人统计信息以及已预约病人信息。

4.1.1 病人

用例	个人就诊卡管理
主要参与者	病人
目标	管理（包括添加、修改与删除）预约平台上个人就诊卡数据
前提条件	用户已经通过登录检查，并且选择预约挂号界面
触发器	用户点击个人就诊卡管理功能
场景	1. 用户进行登录检查
	2. 选择预约挂号界面
	3. 点击个人就诊卡管理
	4. 新窗口中显示已有就诊卡信息，用户使用添加、修改或删除就诊卡功能
	5. 点击确定修改
	6. 服务器后台将更改信息存档并更新数据库信息
	7. 就诊卡界面显示新的就诊卡数据
异常	1. 无法通过登录检查
	2. 网站不可用（界面无法正常显示）
	3. 无法正常显示服务端储存相应就诊卡信息
	4. 无法保存做出的修改（无法修改服务器数据）
优先级	非常重要
使用频率	较频繁
使用方式	浏览器

个人就诊卡管理用例

用例	挂普通号
主要参与者	病人
目标	用户选择相应科室并确定时间进行挂号操作
前提条件	用户个人就诊卡信息完整
触发器	用户点击门诊中的普通门诊
场景	1. 用户补充完整个人就诊卡信息
	2. 点击门诊中的子选项普通门诊
	3. 选择相应科室
	4. 在跳转新界面中选择时间点击挂号
	5. 进入预约信息确认界面，点击确认
	6. 收到预约成功短信
异常	1. 网站不可用（界面无法正常显示）
	2. 无法获取就诊人信息
	3. 无法正常收到预约成功通知短信
优先级	重要
使用频率	非常频繁
使用方式	浏览器

挂普通号用例

用例	挂专家号
主要参与者	病人
目标	用户选择对应科室，确定就诊专家与就诊时间进行挂号操作
前提条件	用户个人就诊卡信息完整
触发器	用户点击门诊中的专家门诊
场景	1. 用户补充完整个人就诊卡信息
	2. 点击门诊中的子选项专家门诊
	3. 选择相应科室
	4. 在跳转新界面中选择相应专家与相应时间点击挂号
	5. 进入预约信息确认界面，点击确认
	6. 收到预约成功短信
异常	1. 网站不可用（界面无法正常显示）
	2. 无法获取就诊人信息
	3. 无法正常收到预约成功通知短信
优先级	重要
使用频率	频繁
使用方式	浏览器

挂专家号用例

用例	退号
主要参与者	病人
目标	用户将已经预约挂号但又未实际取号的记录撤销
前提条件	用户存在满足条件的预约挂号记录
触发器	用户对某挂号记录点击预约取消
场景	1. 用户进行登录检查
	2. 选择预约挂号记录界面
	3. 选择相应记录进行预约取消
	4. 取消成功，短信通知
异常	1. 网站不可用
	2. 无法获取挂号记录
	3. 挂号记录中不存在未取号的记录
	4. 未取号记录已超过取消截至时间
优先级	较重要
使用频率	不频繁
使用方式	浏览器

病人退号用例

4.1.2 医生

用例	查看个人排班情况
主要参与者	医生
目标	用户查看当周个人排班情况
前提条件	用户身份已经认证
触发器	用户点击查看当周排班情况
场景	1. 用户进行登录验证
	2. 在界面中选择查看当周排班情况
	3. （额外操作）点击某次排班进行请假
异常	1. 网站不可用
	2. 登录失败或登录异常
	3. 无法获取排班信息
	4. 请求请假失败
优先级	重要
使用频率	较频繁
使用方式	浏览器

查看个人排版情况用例

用例	查看个人统计信息
主要参与者	医生
目标	用户查看个人统计数据
前提条件	用户身份已经认证且获得权限
触发器	用户点击查看个人统计信息
场景	1. 用户进行登录验证
	2. 在界面中选择查看个人统计信息
	3. 显示统计数据
异常	1. 网站不可用
	2. 登陆失败或登录异常
	3. 无法获取个人统计信息
优先级	较重要
使用频率	不频繁
使用方式	浏览器

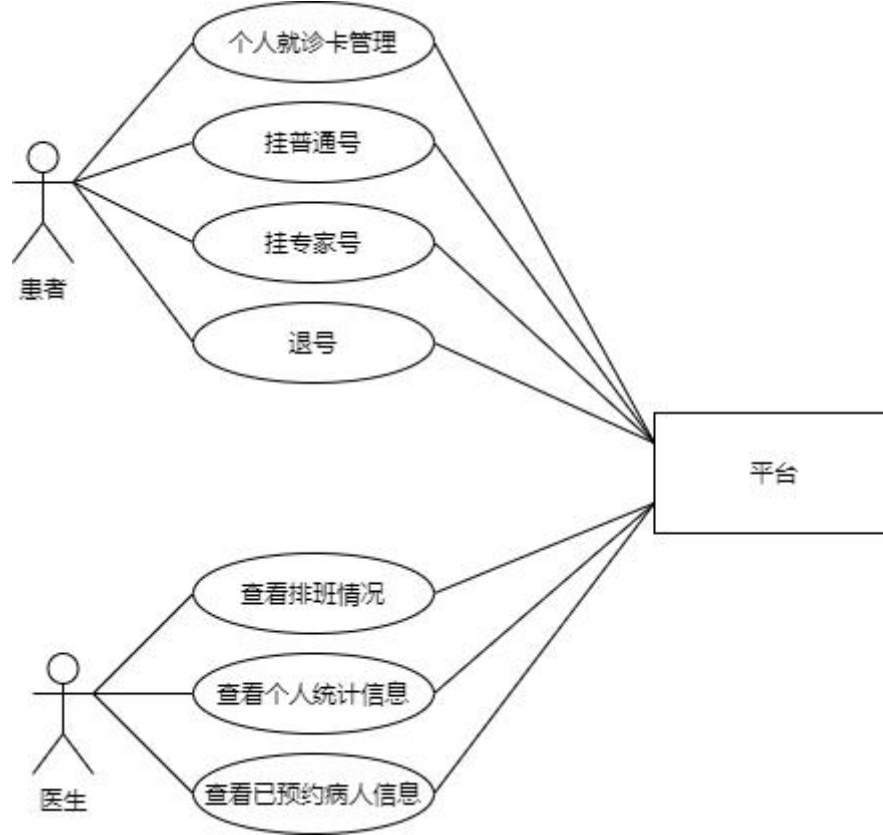
查看个人统计信息用例

用例	查看当周已预约病人信息
主要参与者	医生
目标	用户查看当周某已预约病人信息
前提条件	用户身份已经认证
触发器	用户点击查看当周病人预约列表
场景	1. 用户进行登录验证
	2. 在界面中选择当周病人预约列表
	3. 点击某个病人
	4. 显示其就诊卡信息
	5. （额外操作）对病人进行短信提醒
异常	1. 网站不可用
	2. 登录失败或登录异常
	3. 无法获取当周病人预约病人信息
	4. 无法使用短信进行提醒
优先级	不重要
使用频率	不频繁
使用方式	浏览器

查看当周已预约病人信息

4.2 用例图

本节将根据上面所说的 6 个用例，绘制出用例图。

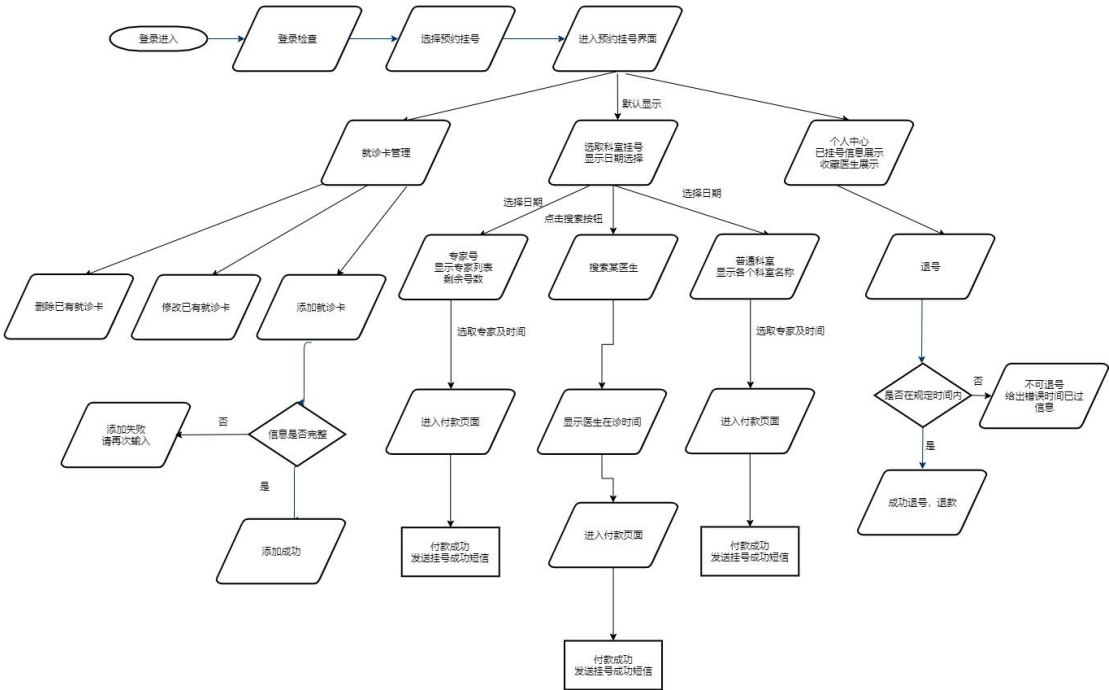


用例图

4.3 状态图

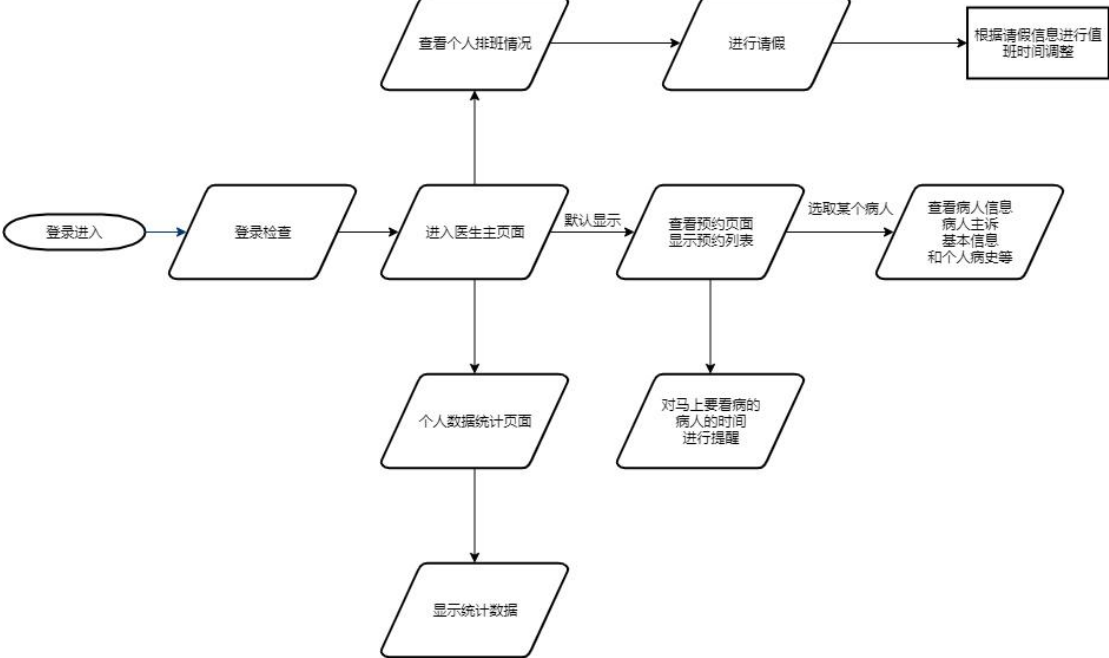
本节将展示子系统中病人和医生的系统状态图，对上面涉及到的用例进行进一步阐述，并为之后的系统实现打下基础。

4.3.1 病人



病人状态图

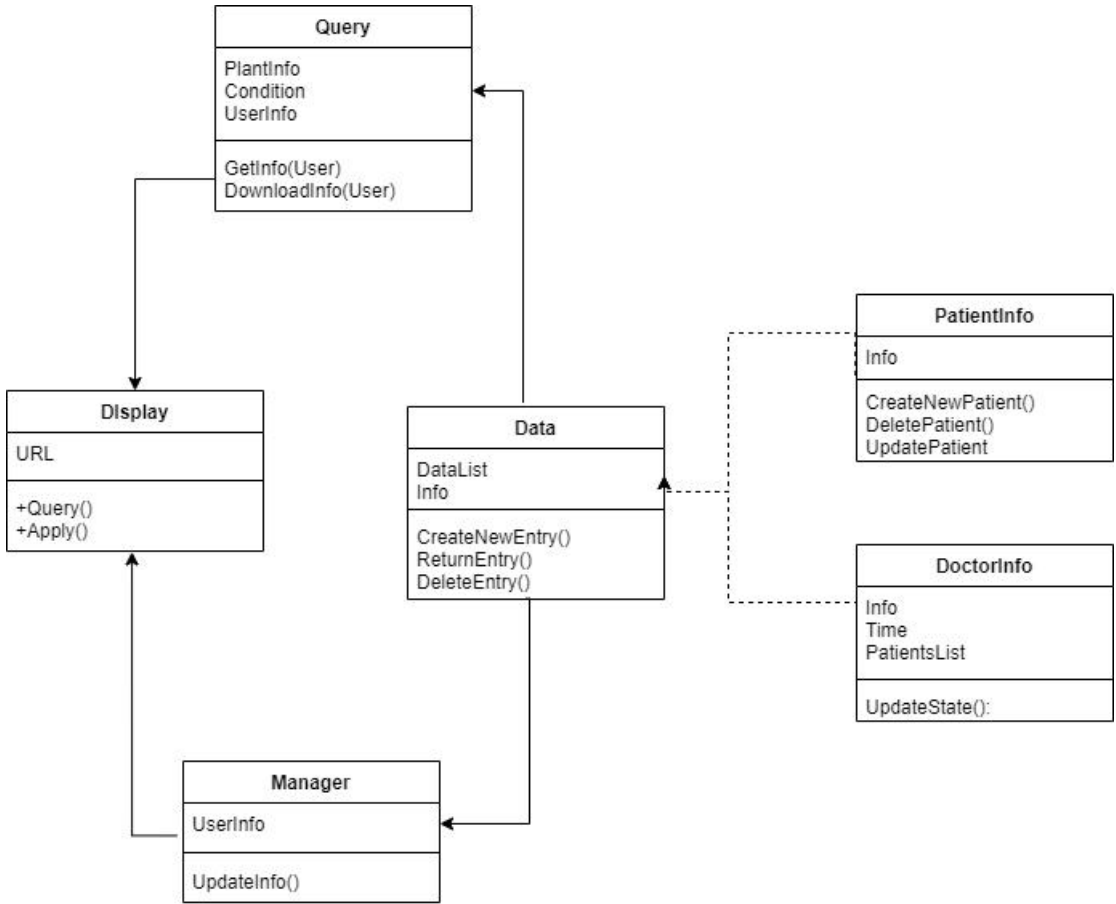
4.3.2 医生



医生状态图

4.4 类图

本节将用例场景具体实现的机制抽象为类图,展示实现系统所必须的相关类、类的属性、操作及其相互之间的关系,为之后的数据流图、CRC 卡等提供基础。

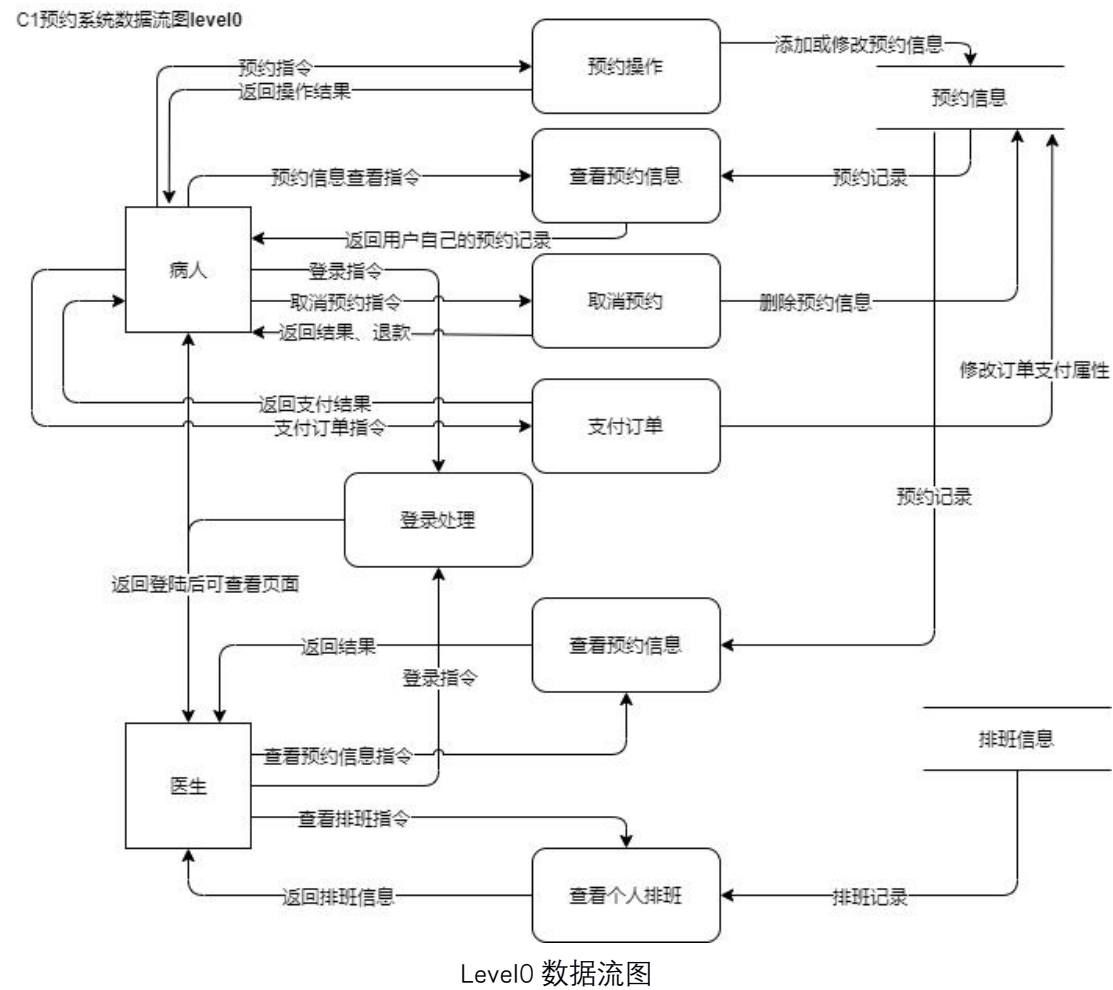


类图

4.5 数据流图

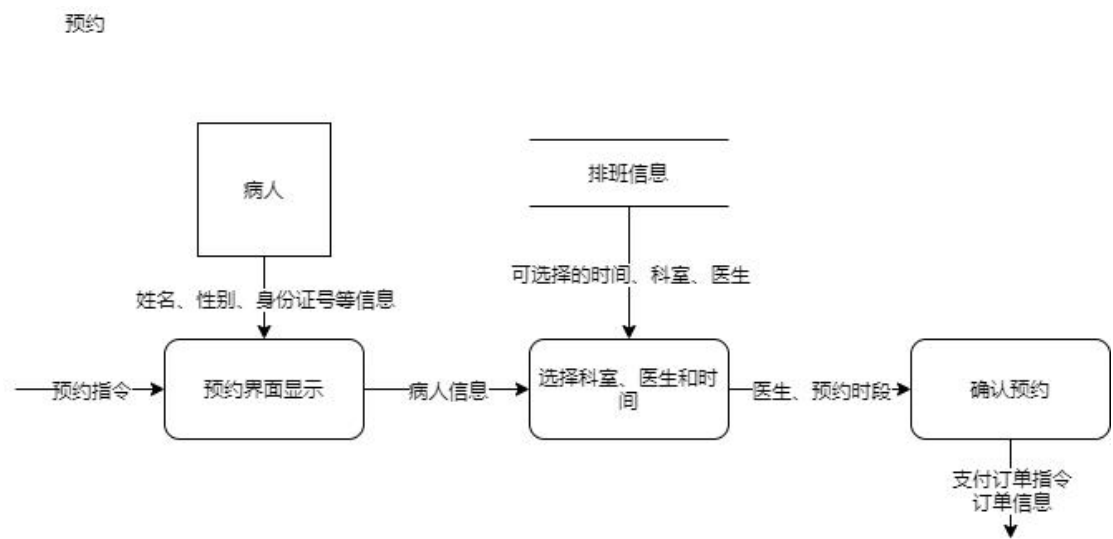
本节将展示子系统中较为常用的系统功能的相关指令、数据的流动方式。

4.5.1 level0



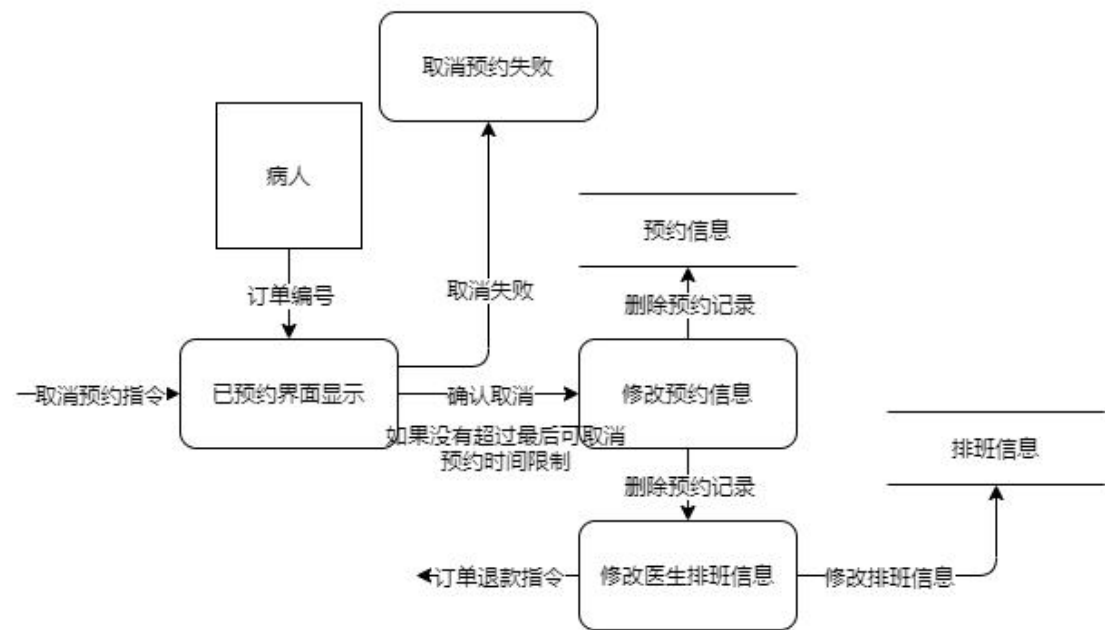
4.5.2 level1

4.5.2.1 预约

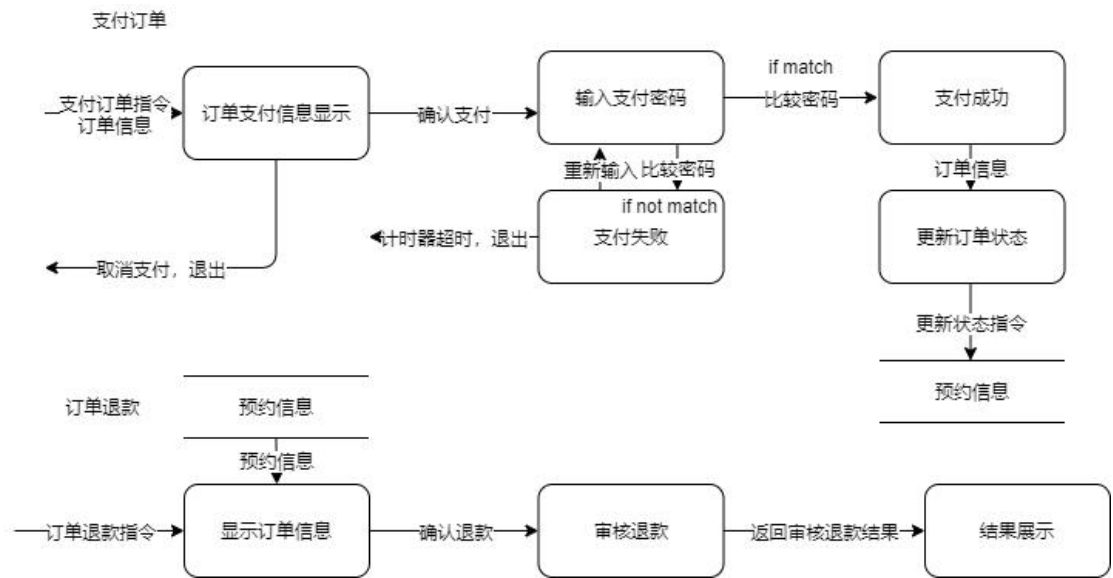


4.5.2.2 取消预约

取消预约



4.5.2.3 支付模块



4.6 CRC 卡

本节将在类图的基础上具体展示不同的类的职责及协作关系。

Class	DBManager
说明	数据库管理程序
职责	协作者
查询医生排班信息列表	Shift
变更医生排班信息列表	Shift, withdraw, regular&expert
查询某医生的预约病人列表	Reservation
变更某医生的预约病人列表	Reservation, withdraw。 regular&expert
查询某时间段空余医生列表	UserChoice
查询某医生的空余时间段列表	UserChoice
查询病人所有就诊卡	MedicalCard

Class	PatientHome
说明	病人主页
职责	协作者
登录	Login, Display
查询与修改就诊卡	MedicalCard, Display

Class	MedicalCard
说明	病人就诊卡
职责	协作者
获取就诊卡列表	DBManager
挂号	regular&expert
退号	withdraw

Class	regular&expert
说明	挂普通号或专家号
职责	协作者
获取用户挂号选择	UserChoice
添加就诊卡	DBManager

Class	withdraw
说明	病人退号
职责	协作者
删除就诊卡	DBManager

Class	UserChoice
说明	病人退号
职责	协作者
显示用户可选医生及其对应时间段	DBManager, Display
返回用户预约的医生与时间段	regular&expert

Class	DocHome
说明	医生个人统计信息
职责	协作者
提供登录校验信息	Login
提供个人统计信息	Display
个人统计信息变更	Shift, Reservation

Class	Reservation
说明	管理医生当周已预约信息
职责	协作者
查询登录情况	Login
获取预约病人列表	DBManager, Display
变更预约病人列表	DBManager
获取预约病人就诊卡	PatientHome, Display

Class	Shift
说明	管理医生当周排班信息
职责	协作者
查询登录情况	Login
获取排班信息列表	DBManager, Display
申请排班请假	DBManager

Class	Display
说明	显示各类页面信息
职责	协作者
显示登录界面与登录情况	Login
显示个人统计信息	Home
显示个人排班信息	Shift
显示当周已预约病人信息	Reservation

Class	Login
说明	用户登录校验
职责	协作者
获取并校验用户登录信息	DocHome, PatientHome
提供登录证明	Reservation, Shift

第 5 章 运行环境规定

下面分别介绍本系统开发运行的前后端软件环境，硬件环境，变更控制和故障处理的规范

5.1 服务端硬件配置需求

以下硬件环境作为软件开发、算法优化设计参考、真实运行硬件环境在此基础上可能产生变化，可能需要进行适当调试及修改，软件研制方如果对硬件环境有更严格的要求，应在软件研制过程中及时提出。本系统涉及到大量的并发需求与数据库操作，需要根据实际情况调整硬件需求

- 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

5.2 服务端软件依赖要求

- 操作系统：Linux
- 数据库：MySQL
- 数据库管理工具：常见 MySQL 管理软件（如 MySQL WorkBench）
- 开发工具：支持 ASP .NET Core 的后端开发工具，与支持 Vue2.0 的前端开发工具
- 测试工具：支持 ASP .NET Core
- 前端框架：Vue2
- 后端框架：ASP .NET Core

5.3 客户端软硬件配置要求

- 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

5.4 变更控制规范

本节说明变更控制的规范，包括：需求变更和变更控制。

5.4.1 需求变更

本系统的开发完全响应项目任务书中规定的功能，但也有可能涉及需求变更，需求的变更存在多种情况：

* 前期需求工作不到位，没有充分收集和整理需求的内容或者需求的理解上面存在一定偏差、甚至出现二义性。* 项目开发过程中遇到的需求难以实现的情况。

对于这两点，可以通过在项目启动之前或者启动的初期，对项目需求进行集中确认，项目执行当中保持经常性的交流和反馈，确保项目需求以及需求变化信息能够充分、准确地传达给项目执行者，以便项目执行始终是围绕着有效的需求进行。

5.4.2 变更控制

实际上不是每个需求都可以事先被确定下来，经验告诉我们有些需求的确是后期操作的时候逐步改进、逐步清晰的。对于这类问题，一方面要事先要能够预见，能够意识到用户很难在需求文本的层面上确认最后的需求，另一方面，技术上可以采取原型开发等方法能够及早检验需求是否合理，在技术框架的设计方面也要保持足够的灵活性，能够支持一定范围的变化。

另外，需求管理体系也需要按照规范建立以支持需求的管理与控制

5.5 故障处理

错误处理影响着系统从开发到使用整个过程中的自身质量及可维护水平；对非正常情况的处理，有以下基本要求：

1. 不论是硬件、软件、系统、用户的各种错误，系统都尽量提示并提出很简要的处理建议；2. 对于一些小的出错，系统容忍并按缺省及自动进入系统进行下一步处理；3. 尽量减少错误的波及面，任何错误不致导致系统崩溃，或引发无可挽回的损失。

5.6 故障确认

当系统发生故障时，系统应能够自动重新执行一次对该系统组件的数据查询，以确认确实是发生了故障，同时，故障确认按钮会激活，相关系统管理人员可以将处理故障的方法、过程等相关信息填入故障确认注释框里，并将该信息自动进行存储，以后遇到同类型的故障时，工程师可以通过查看故障确认日志获得相关参考信息。

故障确认功能可以避免系统管理人员进行重复性操作，提高对故障的处理速度和水平，帮助系统管理人员及时沟通故障处理情况。故障处理日志功能，更能为其它网络管理人员提供技术指导和参考，有效地提高系统管理人员的工作效率，为信息和服务平台的稳定运行提供可靠的保障。

5.7 故障自动恢复

故障自动恢复功能对网络管理人员具有非常重要的意义。当出现一个临时性的标准故障时，自动恢复功能就会自动执行指定脚本或 POST 数据到指定的相关程序，使服务恢复正常。把一个挂起的服务器操作通过自动启动功能而使它重新运行；把一个耗费系统过多资源的进程停止或者重新启动设备。

5.8 故障连锁诊断

在系统经常出现的问题中，有一些是属于流程性连锁故障，需要按照一定的顺序和特定的流程去排除，这些重复性操作繁杂且效率较低。因此需要提供通过故障连锁诊断功能，解决这一问题。

系统工程师可将常见故障的连锁诊断过程做成流程图发布到信息集成和服务系统中，一旦系统出现上述故障，系统将自动启动连锁诊断功能，帮助系统管理人员逐一排查故障，迅速发现、定位故障。

第 6 章 总结

下面对本次报告做一个总结，本次报告按照下面的步骤，系统性地确定了待开发软件“健康导航平台”中“挂号预约系统”的相关需求。

1. 说明了编写此需求分析报告的目的和背景，在这个部分定义了一些常用专业术语并做了概述；
2. 描述了本系统的组成和对应模块信息，从而规定了任务书中所定下的任务需求，包括对系统总体功能、各模块功能、性能、集成、接口、数据/模型、界面、系统质量、质量控制和设备验收的相关要求规定；
3. 实际调研并分析出了系统将要实现的用例、状态图、CRC 图、类图以及系统变化的状态图；
4. 规定了系统运行的软硬件环境，包括前端与后端；
5. 明确了对应模块的验收标准以及与其他模块的协作。

本次需求分析报告，将为此后的系统总体设计、详细设计打好基础，并为项目的顺利开发与验收提供了依据。