



江西中醫藥大學

## 2025 届本科生毕业论文

儿童健康跟踪管理系统的设计与实现

学 号： 202101014083

姓 名： 徐博亮

学 院： 计算机学院

指导老师： 章新友

专 业： 计算机科学与技术

完成时间： 2025.05.12

## 江西中医药大学学士学位论文原创性声明

本人郑重声明:所呈交的论文(设计)是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的地方外,论文(设计)不包含任何其他个人或集体已经发表或撰写的成果作品。对本论文(设计)的研究作出重要贡献的个人和集体,均已在文中以明确的方式表明。本人完全明白本声明的法律后果由本人承担。

签名:徐增亮 2025 年 05 月 15 日

## 关于论文(设计)使用授权的说明

本人完全了解江西中医药大学有关保留、使用毕业论文(设计)的规定。相关规定的内容为:学校有权保留送交论文(设计)的原件,允许论文(设计)被查阅和借阅;学校可以公布论文(设计)的全部或部分内容,可以采用影印、缩印或其他复制手段保存论文(设计)和汇编本论文(设计)。

签名:徐增亮 2025 年 05 月 15 日

## 摘 要

社会对儿童健康问题的关注度持续升高，医疗信息化进程不断加速，在此背景下儿童健康管理在实际应用里的地位变得越发关键，本研究围绕构建一套适合日常管理场景的儿童健康跟踪管理系统进行，系统开发时选用 Vue3 作为前端框架，Spring Boot 作为后端支撑技术，还配合 MySQL 数据库完成数据持久化设计。系统设计的最关键的是实现对儿童健康相关信息的全面记录以及动态跟踪，支持生长发育、营养摄入、体温变化、疫苗接种等多个维度的数据管理功能，可协助家长及医务人员更高效地掌握儿童健康状况，前端界面基于 Vue3 与 Element Plus 构建，注重操作的直观性以及数据访问的便利性，后端部分借助 Spring Boot 框架搭建，采用 RESTful API 风格达成前后端的数据高效通信。系统引入 MySQL 数据库对各类信息统一管理，提高了整体的数据安全性与稳定性。

**关键词：**儿童健康跟踪管理系统 Vue3 ; Element Plus; Java ; Spring Boot ;MySQL ; 数据库；生长发育；饮食管理；疫苗接种；电子健康档案

## Abstract

The society's attention to children's health issues has been continuously increasing, and the process of medical informatization has been accelerating. Against this background, the position of children's health management in practical applications has become increasingly crucial. This study focuses on constructing a set of children's health tracking and management systems suitable for daily management scenarios. When developing the system, Vue3 was selected as the front-end framework. Spring Boot, as the back-end supporting technology, also cooperates with the MySQL database to complete the design of data persistence. The most crucial aspect of the system design is to achieve comprehensive recording and dynamic tracking of information related to children's health, support data management functions in multiple dimensions such as growth and development, nutritional intake, body temperature changes, and vaccination, and can assist parents and medical staff in grasping children's health conditions more efficiently. The front-end interface is constructed based on Vue3 and Element Plus. Emphasizing the intuitiveness of operation and the convenience of data access, the back-end part is built with the help of the Spring Boot framework and adopts the RESTful API style to achieve efficient data communication between the front-end and back-end. The system introduces the MySQL database to uniformly manage all kinds of information, improving the overall data security and stability.

**Keywords:** Children's Health Tracking Management System Vue3; Element Plus; Java ; Spring Boot ; MySQL ; Database; growth and development; dietary management; vaccination; electronic health record

# 目 录

摘 要.....	I
Abstract.....	II
目录.....	III~IV
<b>1 绪论</b> .....	1
1.1 研究背景.....	1
1.2 研究现状.....	2
1.3 发展趋势.....	2
<b>2 相关技术简介</b> .....	3
2.1 vue3,Java,springboot,element-plus 技术.....	3
2.2 B/S 结构 .....	3
2.3 MYSQL 数据库 .....	4
2.4 技术架构集成.....	4
<b>3 系统分析</b> .....	5
3.1 可行性分析.....	5
3.1.1 技术可行性.....	5
3.1.2 经济可行性.....	5
3.1.3 操作可行性.....	5
3.1.3 法律可行性.....	6
3.2 系统性能分析.....	6
3.3 系统功能分析.....	7
3.4 系统流程分析.....	7
3.4.1 操作流程.....	7
3.4.2 添加信息流程.....	9
3.4.3 编辑信息流程.....	11
3.4.4 删除信息流程.....	11
3.4.4 查找信息流程.....	12
<b>4 系统设计</b> .....	12
4.1 系统结构设计.....	12
4.2 系统顺序图设计.....	14
4.2.1 登陆模块时序图.....	14
4.2.2 用户管理模块时序图.....	14
4.3 数据库设计.....	15
4.3.1 数据库 E-R 图设计 .....	15
4.3.2 数据库概述.....	16

---

4.3.3 数据库表结构设计 .....	16
4.3.4 数据库性能优化 .....	16
4.3.5 数据库安全设计 .....	17
<b>5 系统实现的主要功能 .....</b>	<b>17</b>
5.1 登录注册模块的实现 .....	18
5.2 用户信息管理 .....	21
5.2.1 管理员对用户管理界面 .....	22
5.2.2 管理员添加用户 .....	22
5.2.3 管理员编辑用户 .....	23
5.3 儿童健康管理模块的实现 .....	24
5.3.1 儿童基本信息管理界面 .....	25
5.3.2 儿童生长记录管理界面 .....	25
5.3.3 儿童体温记录管理界面 .....	27
5.3.4 儿童饮食记录管理界面 .....	28
5.4 医疗信息模块的实现 .....	28
5.4.1 就医记录管理界面 .....	28
5.4.1 用药记录管理界面 .....	29
5.4.1 疫苗接种管理界面 .....	30
5.4.1 过敏史管理界面 .....	31
5.4.1 用户基本信息修改界面 .....	31
<b>6 系统测试 .....</b>	<b>33</b>
6.1 测试目标 .....	33
6.2 测试策略 .....	33
6.3 测试任务 .....	33
6.4 测试结果 .....	34
<b>七、总结与展望 .....</b>	<b>35</b>
<b>参考文献 .....</b>	<b>36</b>
<b>致谢 .....</b>	<b>38</b>
<b>个人简介 .....</b>	<b>39</b>

# 1 绪论

## 1.1 研究背景

随着社会持续发展以及居民生活质量持续提升，儿童健康问题渐渐成为公众极为关注的话题，近些年来，儿童群体里出现的肥胖倾向、视力状况以及营养摄入不均衡等情形变得日益常见，自新冠疫情出现以来，儿童健康监测的意义被突显出来，随着生育政策的变动，家庭在养育孩子方面投入的关注度与精力较大增多，预防性健康管理的观念也跟着有所提升。不过当下依旧面临如医疗资源分配不均衡、儿科专业人才匮乏等实际问题，家长与医护人员之间在信息沟通方面也存在较大妨碍。

从技术角度来讲，互联网医疗的迅速发展为解决上述难题提供了全新思路，在“互联网 + 医疗健康”政策的带动下，远程诊疗以及在线健康服务渐渐融入到日常的生活当中。Vue3 等现代前端框架的广泛运用、借助 Spring Boot 构建的后端系统稳定性提高，以及数据库安全机制的不断优化，都为搭建高效的儿童健康管理平台提供了可行的技术支持，然而目前仍有不少机构依靠传统纸质档案来记录儿童健康信息，这种模式已无法契合当前对于信息实时性与完整性的需求，电子化、结构化的健康数据管理体系急需建立。

## 1.2 研究现状

### 1.2.1 国内研究现状：

当下我国于儿童健康管理领域开展的信息化建设已然有了一定成果，不过总体而言，尚处于初级阶段，未来有较大的发展空间，在一些经济发展水平较高的地区，部分医疗机构已经渐渐引入了电子健康档案系统，以此来开展基础的儿童健康信息管理工作，比如北京、上海等一线城市的一些儿童专科医院搭建起了比较规范的电子病历平台，基本上达成了健康数据的录入与查询功能。

即便如此，现有的系统存在着不少限制，多数平台功能比较单一，在数据共享与交互能力方面表现欠佳，而且用户界面设计以及操作便捷性方面以及待提高，许多系统依旧以传统的病历信息管理作为主要内容，缺少对儿童成长过程里的关键健康指标进行系统性监测的能力，例如在生长发育记录、营养摄入追踪、预防接种计划等方面，目前的电子系统尚未构建起统一、完善的管理机制，不能契合实际应用中对于综合化、动态化数据管理的要求。

### 1.1.2 国外研究现状

在儿童健康管理信息化这一领域当中，发达国家占据着领先的位置，比如说美国的 Epic 系统以及欧洲的一些医疗信息系统，它们已然达成了相对成熟的儿童健康管理功能，这些系统一般拥有以下这些特点：

- 1、数据全面：可囊括从新生儿一直到成年阶段的完整健康档案
- 2、互操作性：可支持不同医疗机构之间的数据进行共享
- 3、智能化水平高：拥有健康预警以及个性化健康指导的功能
- 4、家庭参与度高：为家长提供便捷的健康管理工具，以此促进家庭成员的积极参与

### 1.1.3 发展趋势

近年来，儿童健康信息系统的开发及应用渐渐引起了社会各界的普遍关注，不少相关研究试着依靠持续跟踪儿童健康数据，来提高健康管理的效率以及科学性，在技术实现层面，这类系统一般采用相似的技术栈，像是运用前后端分离架构以及主流数据库技术，以此保证系统运行的稳定性和数据处理的可靠性。

在用户界面设计方面，现代健康管理平台变得日益重视交互体验以及界面的友好程度，像 Vue3 和 Element Plus 等前端技术的引入，让界面在视觉呈现以及功能使用上更加直观，方便家长和医务人员对儿童健康信息进行快速录入与查询，这种以用户为中心的设计理念，提升了系统的操作便捷性，也契合当前技术发展的趋势。

系统安全性方面，后端大多时候使用 Java 语言结合 Spring Boot 框架进行开发，搭配 MySQL 数据库完成数据的存储与管理，为提高数据隐私保护水平，系统在设计时引入了加密存储、分权限控制等机制，降低敏感信息泄露的风险，在当前数据安全监管变得日益严格的背景下，保障用户信息的安全性成为系统建设中必不可少的一部分。



---

## 2 相关技术简介

### 2.1 前后端相关技术：

#### 2.1.1 Vue3 框架

Vue3 作为当下的前端框架，给予了如 Composition API 等新特性，在代码组织方面更为出色，性能表现也更为优越，依靠响应式系统以及虚拟 DOM 技术，Vue3 可有效构建用户界面，同时支持数据驱动的视图更新，TypeScript 的集成对提升开发的稳定性与可靠性同样有所帮助。

#### 2.1.2 Element Plus

Element Plus 是一款依托 Vue3 构建的 UI 组件库，其提供了诸多常用的预置组件，像表单、表格以及导航等，此组件库有统一的设计风格，交互体验颇为不错，可有效提升开发效率以及用户满意度，在本系统里，Element Plus 的组件于各个功能模块中得到广泛应用，保证了用户界面有专业性与整体一致性。

#### 2.1.3 Spring Boot 框架

Spring Boot 作为当下被广泛运用的 Java 后端框架，对 Spring 应用的初始化以及开发流程起到了简化作用，Spring Boot 遵循“约定优于配置”这一原则，使得配置工作大幅减少，而且凭借其强大的自动配置特性与丰富的启动器，可有效提升开发效率。在本系统里，Spring Boot 主要承担处理 HTTP 请求、执行业务逻辑以及达成数据访问等后端任务。

### 2.2 B/S 架构

浏览器/服务器架构乃是现代 Web 应用当中主要的架构模式之一，于本系统里，采用了前后端分离的 B/S 架构设计，具体如下：

- 前端：运用 Vue3 来构建单页面应用
- 后端：借助 Spring Boot 提供 RESTful API 服务
- 通信：前后端之间依靠 HTTP 协议来进行数据交互

**采用这样的架构有着诸多优势：**

1. 跨平台性：用户只要借助浏览器便可访问系统
2. 部署的灵活性：前端静态资源和后端服务可分别独立进行部署
3. 良好的扩展性：可针对前端或者后端单独实施扩展
4. 较低的维护成本：客户端不需要进行维护，服务端统一开展更新

## **2.3 MYSQL 数据库**

MySQL 作为一款在众多领域有着广泛应用的关系型数据库管理系统，在当前所讨论的这个系统里，它肩负着对于数据进行存储以及实施管理的关键任务：

1. 数据持久化：其用途在于存储有关儿童的各类基本信息，以及涉及健康档案等在内的关键数据
2. 事务处理：要切实保障数据操作有原子性以及数据保持一致性
3. 数据安全：需设置用户身份验证举措以及权限控制机制
4. 高性能：借助索引与查询优化手段，达成高效的数据访问目的

本系统数据库设计依照规范化设计原则展开，创建了涉及儿童信息、生长记录、饮食记录、体温记录以及疫苗接种记录等多方面的相关数据表，构建起较为完善的数据存储体系。

## **2.4 技术架构集成**

对上述各项技术给予有效整合之后，成功构建出了一个完整的全栈解决方案：

1. 前端：运用 Vue3 以及 Element Plus 来达成用户界面及其交互功能的构建
2. 后端：借助 Spring Boot 来为应用程序供给业务逻辑以及数据服务
3. 数据库：选用 MySQL 来承担数据的存储以及管理工作
4. 通信：借助 RESTful API 达成前后端的数据交换操作

这种技术架构保障了系统有可靠的性能以及可维护的特性，同时也为未来的功能拓展和性能提升打造了稳固的根基。

## 3 系统分析

### 3.1 可行性分析

可行性分析要依据用户需求，明确儿童健康跟踪管理系统的研究方向与目标，经多角度可行性评估，确定系统整体框架以及功能模块设计，这部分会从技术、经济、操作、法律这四个方面展开详细剖析。

#### 3.1.1 技术可行性

本系统运用主流的 Vue3 加上 Spring Boot 技术栈开展开发工作，前端借助 Vue3 框架搭配 Element Plus 组件库，可迅速搭建出美观又功能完备的用户界面，后端采用 Spring Boot 框架，其拥有成熟的生态系统以及丰富的开发文档，为系统开发给予了可靠的技术支撑。项目组成员对 Web 开发技术有着较为深刻的理解，且在前期做了充足的技术储备以及原型验证，MySQL 数据库的运用也保证了数据存储的可靠性与安全性，从技术层面而言，本系统的开发有充分的可行性。

#### 3.1.2 经济可行性

本系统开发成本主要覆盖开发环境搭建以及后期运维这两个部分，像 Vue3、Spring Boot、MySQL 这类开发工具皆属于开源软件，无需支付授权费用，开发环境所需硬件设备仅需基本的用于开发的计算机即可，在系统部署层面，可选择性价比高的云服务器来部署，运维成本在可控范围之内。相较于传统纸质档案管理模式，本系统能提高工作效率，降低人力成本，有着不错的投资回报比例，从经济角度而言，本系统的开发与维护成本是可接受的。

#### 3.1.3 操作可行性

系统设计秉持“以用户为中心”理念，运用直观的界面设计以及清晰的操作流程，借助 Element Plus 组件库所提供的标准化 UI 组件，达成了界面风格的统一以及操作的一致，系统依据不同角色，也就是家长、医生、管理员，设计了差异化的功能模块与操作界面，还给出了必要的操作提示和帮助信息。用户仅需经过简单培训，便可迅速掌握系统的使用方法，系统的响应式设计保证了在不同设备上有良好的使用体验，从操作层面来讲，本系统拥有较高的可用性与易用性。

### 3.1.3 法律可行性

本系统开展开发工作时严格遵循相关法律法规，所运用的技术框架以及组件皆是开源软件或者已经获取合法授权的，在数据处理这一方面，系统严格依照《中华人民共和国个人信息保护法》等相关法律规定，针对用户数据给予加密存储以及安全管理，系统于设计和实现的过程当中并未侵犯任何第三方的知识产权，所有功能模块都是自主进行开发的。系统还达成了完善的权限管理机制，以此保证用户数据的访问契合相关法律要求，从法律角度而言，本系统的开发以及使用是完全合规的。

## 3.2 系统性能分析

（1）**系统的存储性：**本儿童健康跟踪管理系统要处理以及存储诸多的健康数据，涉及生长发育记录、体温记录、饮食记录、疫苗接种记录等多个维度的信息，该系统选用 MySQL 数据库当作数据存储方案，有高效的数据存储与检索能力，可支持大规模数据的快速访问，并且系统达成了完善的数据备份和恢复机制，以此保障数据安全，还支持分表分库策略，为未来系统的扩展创造了可能性。

（2）**系统的易学性：**系统设计遵循“简单易用”准则，运用直观的界面设计，契合用户使用习惯，借助清晰的功能分类以及导航结构，再结合智能表单填写与必要的操作提示，有效降低了用户的学习成本，系统有数据可视化展示功能，可使复杂的健康数据更便于理解和分析，还支持快捷操作与批量处理，提高了工作效率。

（3）**系统的可靠性：**该系统构建起一套周全的安全保护机制，于数据安全性层面，运用加密传输以及存储机制，达成了完备的用户认证与授权管理，还构建起定期数据备份以及灾难恢复机制，在系统稳定性方面，借助分层架构设计削减了模块间耦合度，达成了完备的异常处理以及日志记录机制，并且支持系统监控以及性能优化，以此保障系统可稳定运行。

（4）**系统响应性能：**系统实施了优化举措，前端运用组件懒加载、数据缓存处理以及资源按需加载等技术来提升用户体验，后端借助数据库索引优化、查询语句优化以及接口响应缓存等方式保证快速响应，另外系统的模块化设计以及标准化接口为未来功能扩展和第三方系统集成带来便利，可配置化设计还可以让系统灵活应对业务规则调整。

(5) **系统安全性：**该系统达成了基于角色的权限管理、多因素身份认证以及操作日志审计等访问控制机制，针对敏感数据，运用加密存储以及严格的访问控制方式，切实保护了系统与用户数据的安全，这些性能特征相互结合，保证了系统可安全、稳定且高效地运行，充分契合儿童健康管理的各项需求。

### **3.3 系统功能分析**

#### **3.3.1 系统总体功能：**

本儿童健康跟踪管理系统服务家长、医生以及管理员这三类关键用户群体，为保障信息安全并实现操作便捷，系统构建了基于角色的权限管理体系，针对不同用户类型开放定制化功能模块，其核心功能包含儿童基本信息维护、生长发育指标记录、膳食营养管理、体温动态监测以及疫苗接种计划管理等诸多方面，希望能够打造一个全面的儿童健康管理平台。展开来说，本系统设定了管理员、家长和医生三种用户角色，用户需先凭借各自的用户名和密码进行身份验证登录系统，系统在数据库层面设计了专门字段用以区分用户角色，在用户成功登录后，呈现与其身份相符的系统导航菜单以及功能界面，这种精细化的角色权限设计，可有效提升系统的专业性与易用性。

#### **3.3.2 用户角色分析：**

##### **3.3.2.1 家长用户功能：**

- (1) 信息管理：查看和更新自己孩子的基本信息
- (2) 生长发育跟踪：记录身高、体重等生长数据，查看发育曲线
- (3) 饮食记录：记录每日饮食情况，获取营养建议
- 1. 体温监测：记录体温数据，查看体温变化趋势
- 2. 疫苗接种管理：查看接种记录
- 3. 就医记录：查看就医历史，管理用药信息

##### **3.3.2.2 医生用户功能：**

- (1) 患者管理：查看负责的儿童健康信息
- 1. 健康评估：进行生长发育评估，提供健康建议
- 2. 接种管理：记录疫苗接种信息，制定接种计划

3. 就医记录：记录诊疗信息，开具用药建议

3、管理员用户功能：

- 1. 用户管理：管理系统用户账号和权限
- 2. 医生管理：维护医生信息和工作分配
- 3. 系统维护：进行系统参数配置和日志管理

系统用例图如图 3-2 所示。

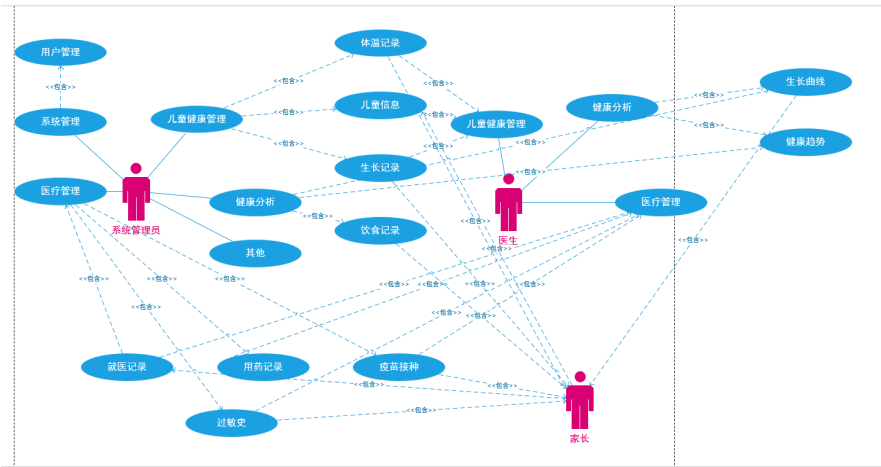


图 3-2 系统用例图

3.4 系统流程分析

3.4.1 操作流程

三种用户要想进入系统进行管理操作，必须登录系统，在登录界面输入登录账号和密码，系统前端首先会校验登录信息是否正确，错误则会自动清除账号和密码的输入框，并给与对应的提示信息；正确则进行后端联合数据库的查询登陆信息是否正确，正确则做三件事：1、登录进入功能界面；2、根据登陆的用户来记录用户的角色，再根据角色来返回与其相对应的系统导航栏菜单；3、返回登陆的 token 并记录，以后的每一次请求都需要携带 token。可进行功能处理，反之登录失败。系统操作的流程图，如图 3-4 所示。

该系统的 token 的工作原理如下：

- (1) 当用户登录时：
  - 用户提交用户名和密码
  - 后端验证用户名和密码正确后，使用 JWT (JSON Web Token) 生成 token

- token 中包含了用户的关键信息（如用户 ID、角色 ID 等）
- 后端将 token 返回给前端

// 登录成功后生成 token

```
Map<String, Object> claims = new HashMap<>();
claims.put("id", user.getUserId());
claims.put("username", user.getUsername());
claims.put("roleId", user.getRoleId());
```

```
String token = JwtUtil.genToken(claims);
```

然后前端把返回的 token 存储到持久化的“pinia”里面，每次请求都要通过请求拦截器将 token 携带到请求头里去。

```
//添加请求拦截器
instance.interceptors.request.use(
  (config) => {
    //在发送请求之前做什么
    let tokenStore = useTokenStore()
    console.log(tokenStore.token)
    //如果 token 中有值，在携带
    if (tokenStore.token) {
      config.headers.Authorization = tokenStore.token
    }
    return config
  },
  (err) => {
    //如果请求错误做什么
    Promise.reject(err)
  }
)
```

### 3.4.2 添加信息流程

添加信息时，系统采用自动增号的模式，在添加信息输入信息，系统会自动对信息数据进行验证，信息合法则验证成功添加至数据库，信息不合法提示添加失败，重新输入信息。添加信息流程，如图 3-5 所示。

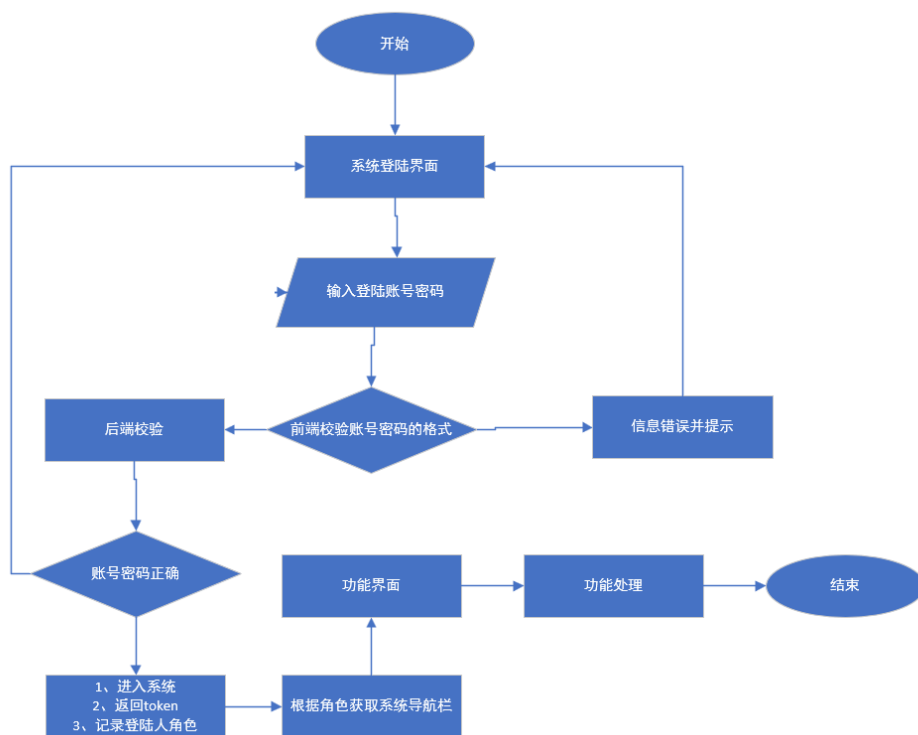


图 3-4 操作流程圖

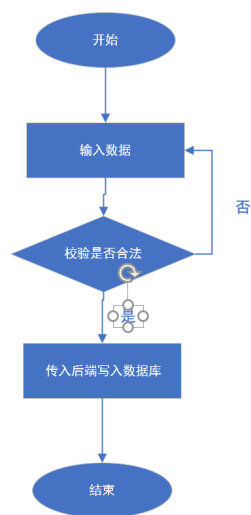


图 3-5 添加信息流程图



### 3.5.3 编辑信息流程

在进行编辑信息操作时，首先进入编辑信息界面，输入编辑信息数据，然后将信息更新至数据库，信息不合法则修改失败，重新输入。编辑信息流程图，如图 3-6 所示。

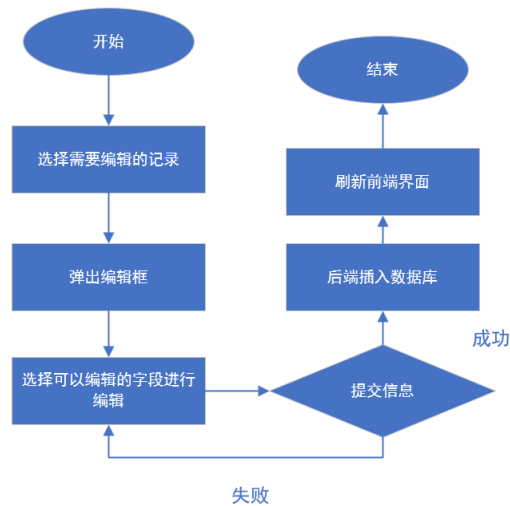


图 3-6 编辑信息流程图

### 3.4.4 删除信息流程

选择要删除的信息，单击删除按钮，系统则提示是否确定删除信息，选择确定删除，则删除信息成功，系统数据库将信息进行删除。删除信息流程图如图 3-7 所示。

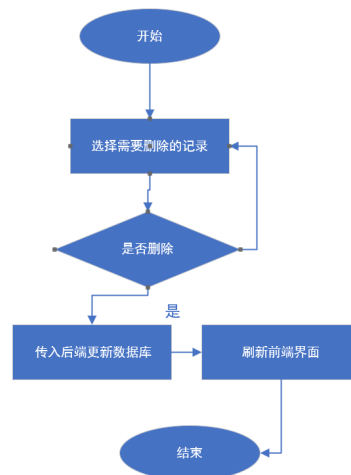


图 3-7 删除信息流程图

### 3.4.5 查询信息流程

通过搜索框输入信息，再敲回车键，通过 传入需要查询的信息到后端，再由后端进行数据库的查询，将数据返回给前端。值得注意的是，每一次的请求需要携带 token 到请求头才可以请求成功，否则请求不成功。查询支持模糊查询。然后将返回的数据由前端的 element-plus 的 el-table 组件进行渲染到页面上，如图 3-8 所示。

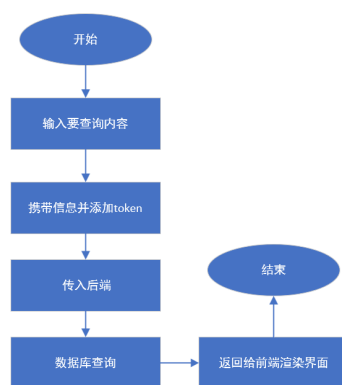


图 3-8 查询信息流程图

## 四 系统设计

### 4.1 系统结构设计

系统结构设计是一个将一个庞大的任务细分为多个小的任务的过程，这些小的任务分段完成后，组合在一起形成一个完整的任务。本儿童健康跟踪管理系统的功能结构设计如图 4-1 所示。

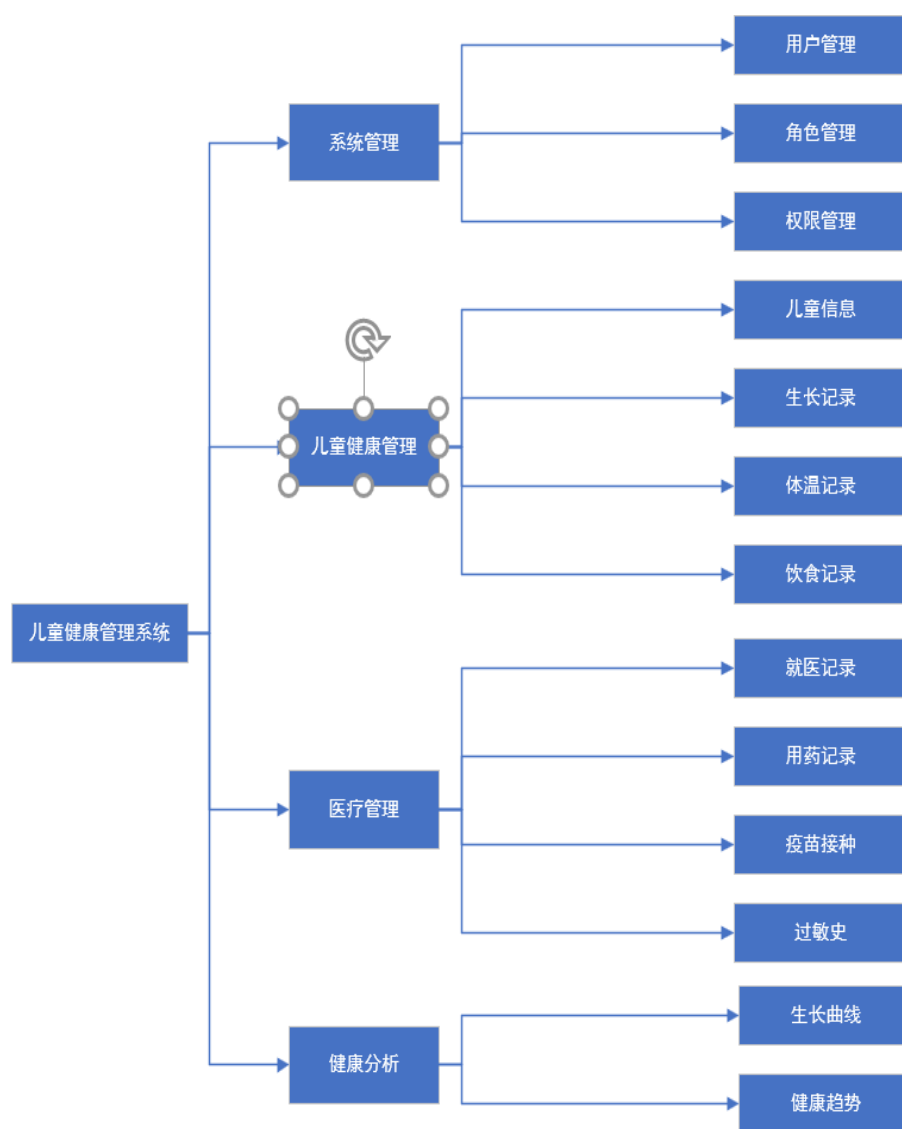


图 4-1 系统结构图

## 4.2 系统时序图设计

### 4.2.1 登陆模块时序图

登陆模块满足了用户人员的登陆，登陆模块时序图，如图 4-2 所示。

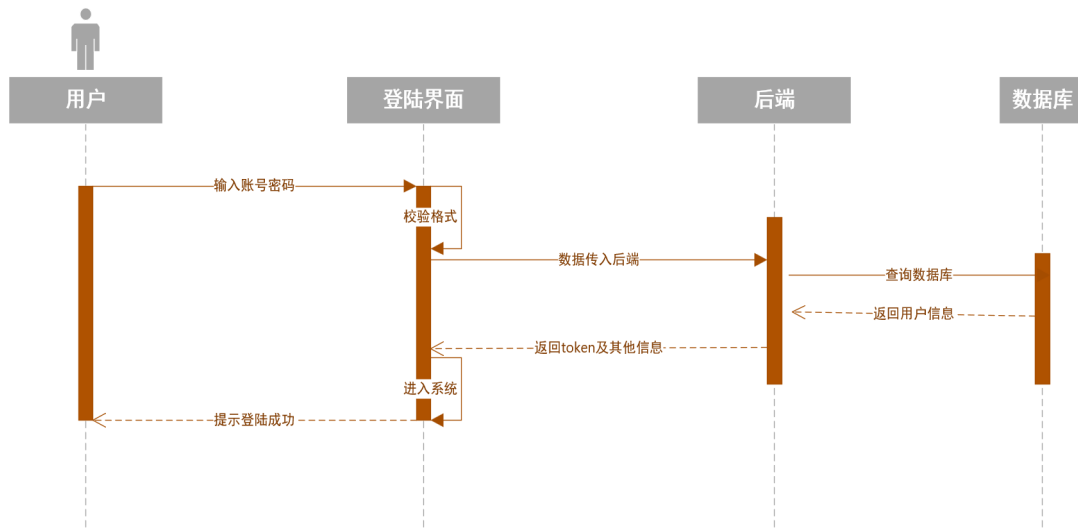


图 4-2 登陆模块顺序图

### 4.2.2 用户管理模块时序图

用户管理模块的顺序图，如图 4-3 所示。

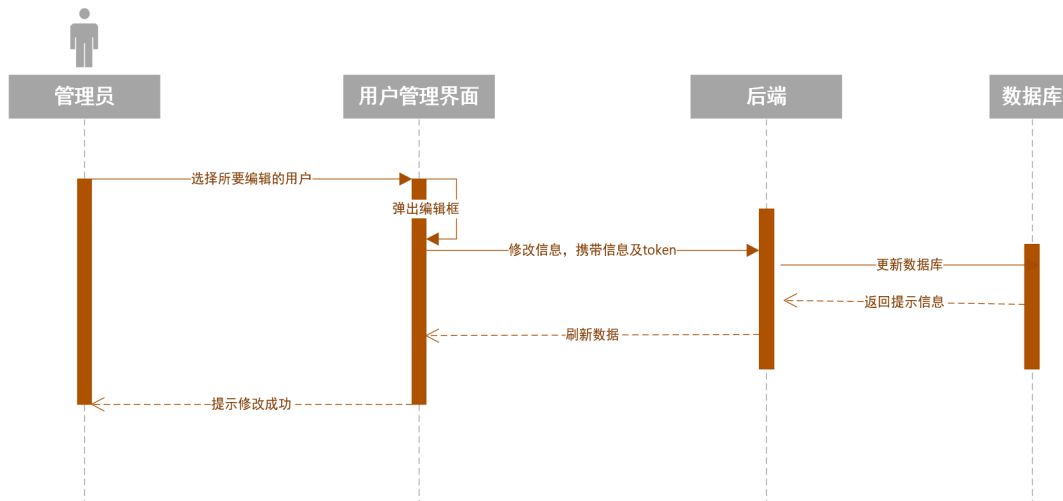


图 4-3 用户管理模块顺序图

## 4.3 数据库设计

### 4.3.1 数据库 E-R 图设计

系统 E-R 图就是系统的实体关系图，它是用来描述某一组织(单位)的概念模型，提供了表示实体、属性和联系的方法。构成 E-R 图的基本要素是实体、属性和关系。实体是指客观存在并可相互区分的事特；属性指指实体所具有的每一个特性。

设计概念就是在数据分析的基础上自下而上的对整个系统的数据库概念结构进行设计。从用户的角度对视图进行开发，然后集成视图，最后分析从而取得最后的结果。

本数据库概念模型是为了将现实世界中信息进行抽象而设计的，从而实现信息世界的建模，因此，概念模型是进行数据库设计的重要工具。数据库的概念模型设计可以通过 E-R 图来现实世界的概念模型，本系统的 E-R 图表现了系统中各个实体之间的联系。

根据系统的一般要求，通过对整个系统功能、运行过程的分析，形成了反应信息需求的概念数据模型。概念模型可以通过 E-R 图来表示，根据数据库中的几个表分别绘制数据库的实体图。系统主要功能的实体 E-R 图，如图 4-4 所示。

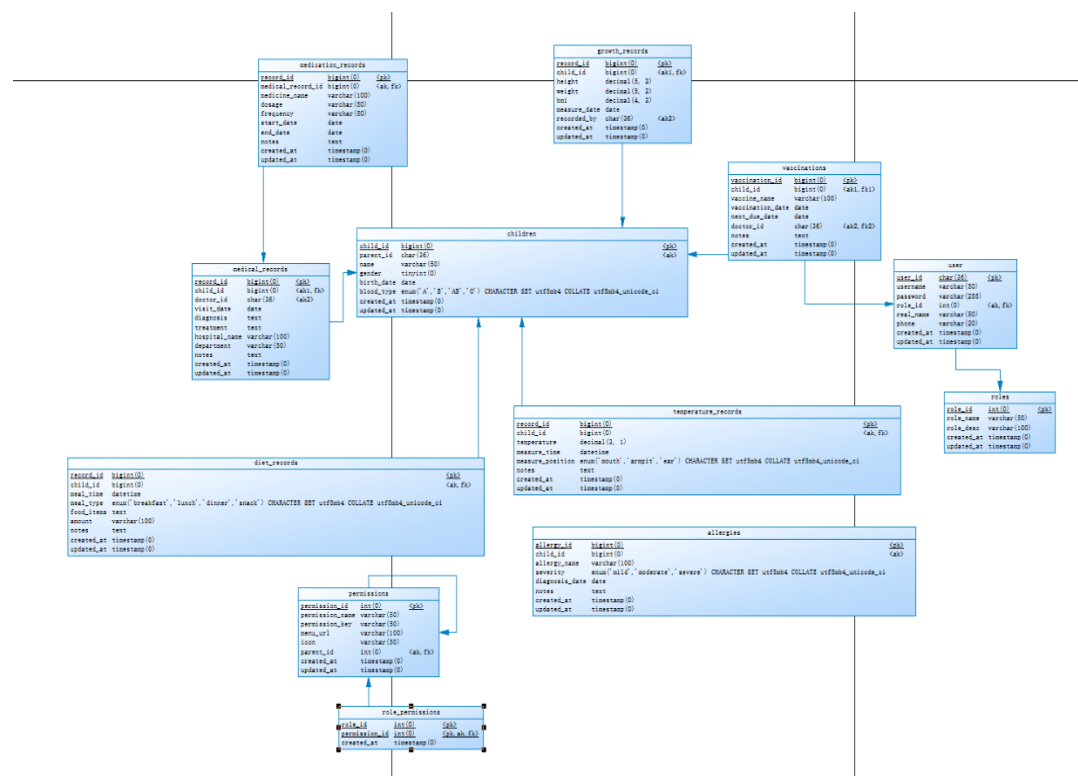


图 4-4 系统主要功能的实体 E-R 图

### 4.3.2 数据库概述

本系统的后台数据存储以及管理选用了 MySQL 8.0 数据库管理系统，并且构建了名为 `child_health_system` 的数据库，在构建数据库时，着重遵循了数据库设计的一些基本准则，涉及数据完整性保障、模式规范化处理、系统性能优化以及未来可扩展性考量。为保证数据准确性和一致性，系统设计中细致运用了主键和外键约束，为避免数据冗余并提高数据组织效率，数据库结构设计采用了第三范式，另外为提升数据检索效率，对关键字段进行了合理索引设置，本文在设计实践中发现，综合应用上述策略，对构建稳定高效且易于维护的儿童健康管理系统很关键。

### 4.3.3 数据库表结构设计

为构建一套完备的儿童健康管理体系，本系统精心设计了八个核心数据表，其中用户表用于存储系统各类用户的基本信息，包括管理员、医生以及家长这三类账户的信息，且密码采用 BCrypt 算法进行加密存储，以此提高账户的安全性，儿童信息表记录儿童个体的基本资料，并依靠 `parent_id` 字段与用户表建立关联，明确家长与子女的对应关系。

生长记录表着重跟踪记录儿童的身高、体重等生长发育指标，系统可自动计算身体质量指数，为健康评估提供数据支持，体温记录表详细记录每次体温监测的数据，囊括具体的测量时间、测量部位等关键信息，这对及时发现儿童的健康异常颇为关键，饮食记录表用于记录儿童每日的膳食情况，包含具体的用餐时段、食物种类以及摄入量等信息，为系统的营养管理模块提供详实的数据基础。

就医记录表与用药记录表一同构建起完整的医疗信息管理体系，前者主要记录儿童的就诊相关信息，比如初步诊断结果和具体的治疗方案，后者则详细记录儿童的用药情况，包括药品名称、使用剂量以及用药时长等关键信息，疫苗接种表专门用于管理儿童的疫苗接种档案，记录每次接种的日期、疫苗种类，还有记录下次疫苗接种提醒的功能。笔者觉得，这些数据表的精细化设计，为系统各项功能的实现奠定了坚实的数据基础。

### 4.3.4 数据库性能优化

为保障系统运行流畅高效，在数据库设计阶段融入多项性能优化考量，对于查询频率较高的字段，谨慎建立相应索引，如用户表的 `username` 字段、儿童信

息表的 `parent_id` 字段以及各类记录表的 `child_id` 字段等，实践说明，合理运用这些索引能提升数据检索速度，在数据量庞大时，查询性能提升明显。

经过对字段类型做细致挑选以及对存储空间给予精心优化，数据存储成本得以有效降低，比如对于性别、状态这类取值范围固定的字段，选用了相对较小的数据类型，对于时间相关字段，依据实际应用需求，灵活选择 `DATE` 或 `DATETIME` 类型，凭借构建恰当外键关系，保证了数据引用完整性，也为执行关联查询提供了性能保障。大家认为，这些细致优化措施是保证系统稳定高效运行的关键环节。

#### **4.3.5 数据库安全设计**

数据安全性在本系统设计里是一项极为关键的考量因素，基于此，我们打造了一套完备的数据访问控制体系，展开来说，系统运用了基于角色的访问控制也就是 `RBAC` 机制，对不同用户的数据访问权限实施了严格的限定，比如系统规定家长用户仅可查看其关联子女的健康档案信息，医生用户则只可以访问其负责管理儿童的相关数据，如此一来可切实防止未经授权的数据访问以及潜在的数据泄露风险，保障系统中数据的安全性与隐私性。

在数据保护层面，采用了多层次的安全举措，借助数据加密保障敏感信息的安全，像用户密码运用 `Md5Util` 算法加密存储，保证即便数据库遭遇攻击也不会泄露用户密码。

---

## 五 系统实现的主要功能

### 5.1 登录模块的实现

在系统的用户认证设计当中，选用了基于 JWT 也就是 JSON Web Token 的身份验证机制，借助这一机制来构建一个相对安全且稳定的登录与注册流程，前端部分采用 Vue3 框架展开开发，并且结合 Element Plus 组件库，达成了界面直观、交互友好的用户体验。为提升表单交互的准确性，系统在输入校验方面进行了精心设计：登录表单含有基础的用户名与密码字段，注册表单在登录表单基础上增添了密码确认功能，还引入了实时验证机制，比如对于用户名设定了 5 到 16 个字符的长度限制，同时针对密码强度、空格输入等方面也做出了必要限制。前端凭借封装 API 接口与后端服务开展数据交互，同时运用 Pinia 状态管理工具，对用户的登录状态和权限信息进行统一维护，为保障后台操作权限的合理分配，系统规定用户要登录后才可进入后台管理模块，登录成功后，系统会依据用户所有的角色动态分配对应的菜单项，登录界面如图 5.1 所示。

#### 5.1.1 整体架构

系统登录模块运用前后端分离架构思路，前端借助 Vue3 框架开发登录界面及相关交互逻辑，后端基于 Spring Boot 框架处理登录验证请求并生成用户身份令牌 token，整个登录流程由多环节协同组成，涉及用户输入合法性校验、登录请求发送、令牌创建与服务端存储以及登录状态持续维护等。实际实现中，前后端依约定接口规范通信，保障登录过程兼具安全性与良好交互体验。

#### 5.1.2 详细细节

在后端架构层面，系统整体依托 Spring Boot 框架搭建而成，依照典型的三层架构模式来架构，以此提高项目的可维护性能与扩展性能，当中控制器层也就是 Controller，主要职责是处理客户端发起的 HTTP 请求，借助 @RestController 注解达成 RESTful 风格的接口定义，覆盖用户注册这个功能，具体路径是 /user/register，以及登录功能，具体路径是 /user/login 等核心功能。服务层也就是 Service，承担着系统主要业务逻辑的封装以及实现工作，包含用户信息校验、密码加密，这里采用的是 MD5 算法，以及 Token 的生成与管理等工作，数据访问层也就是 Mapper，借助 MyBatis 框架实现与 MySQL 数据库的交互，负责达成用户相关数据的持久化操作。



为强化系统安全性，后端引入基于拦截器的身份认证机制，针对所有需鉴权的请求开展 JWT Token 验证，借助 ThreadLocal 技术达成用户会话信息于请求生命周期内的隔离与管理，切实保障了接口调用过程中的认证一致性以及数据安全。

在数据库设计方面，系统构建了以用户表为核心的数据模型，表中涉及用户 ID、用户名、加密后的密码、角色 ID、真实姓名、联系电话等基础字段，且借助外键与角色表建立关联，达成基于角色的访问控制。系统支持三类角色的灵活配置，分别为管理员、医生和家长，每类角色拥有不同的权限范围，用户登录成功后，系统会生成包含身份标识的 JWT Token，并在后续请求中经由 Authorization 请求头完成身份验证，另外系统还有用户信息管理功能，支持用户进行个人资料修改、密码变更以及头像上传等操作。所有接口统一经由自定义的响应封装类返回数据，保证前后端通信格式的一致性和规范性。

### 5.2.3 登陆界面

系统登录界面的设计十分注重视觉简洁性以及用户操作的直观性，整体采用了现代简约风格，并且是基于 Element Plus 组件库来构建的，在布局设计方面，页面采用左右分栏结构，左侧用于展示系统品牌，其背景运用柔和的渐变色调，这样能让“CHILD”标志以及“儿童健康跟踪管理系统”的系统名称在视觉上更具辨识度，提高平台的专业形象和信任感。右侧区域是用户交互模块，集成了登录与注册功能，借助组件的动态切换机制实现单页面内的多功能交互，提升了使用的便捷程度，具体到表单交互层面，登录界面围绕用户体验进行了细节优化，用户名与密码的输入框是基于 el-input 组件构建的，并且分别嵌入了用户图标与锁形图标，以此提高信息输入的识别性以及界面美观度。系统在输入验证方面设计了实时反馈机制，用户名要求是 5 至 16 位的非空字符，密码同样限制在相同长度范围内，且不能包含空格，一旦用户输入不符合规则，系统会即时提示具体的错误信息，帮助用户快速修正，登录按钮采用了与系统主题色调协调的主色风格，点击后会呈现加载动画，提供了良好的交互反馈，也提高了界面响应的可感知性。

### 5.2.4 优化用户体验

平台在登录模块实施登录状态持久化管理机制，同时结合实时表单校验以及清晰错误反馈设计，以此降低用户操作失误率，提升系统用户体验与安全保障能力，在安全性上，系统对用户密码加密存储，借助 Token 机制完成身份认证，还严格

验证前端请求参数格式与合法性，减少潜在非法访问风险。这些机制共同构成儿童健康管理系统用户认证基础保障体系，系统登录界面如图 5-1 所示。



图 5-1 系统登录界面

系统使用 Pinia 进行状态管理，如下是创建了专门的 token 存储模块，用于管理用户登录状态程序如下。

```
// stores/token.js

export const useTokenStore = defineStore('token', {

  state: () => ({

    token: localStorage.getItem('token') || ""

  }),

  actions: {

    setToken(token) {

      this.token = token

      localStorage.setItem('token', token)

    },

    removeToken() {

      this.token = ""

      localStorage.removeItem('token')
```

```
}  
  
}  
  
})
```

系统的注册功能是按照前后端分离架构来开发的，前端跟后端借助 RESTful API 接口达成数据交互，达成模块分工结构清晰、职责明确，为保障用户注册过程安全稳定，系统在前后两端都实现了严格的数据验证机制，还搭配适当的交互反馈来提升使用体验，在前端界面设计里，注册表单和登录表单整合在同一页面右侧区域，用户能依靠按钮切换来完成交互视图的切换。注册模块主要针对家长群体，支持他们自助创建账号，表单是由 Element Plus 提供的 el-form 组件构建的，有用户名、密码以及确认密码这三个核心字段，并且都有实时验证逻辑，防止无效或不规范的数据提交，系统给新注册用户默认分配“家长”角色，注册成功后，前端会自动引导用户跳转到登录界面，还会清空原表单输入信息，保证用户体验的连续性。另外为提高用户对操作状态的感知，系统在注册过程中引入了消息提示机制，依靠 ElMessage 组件实现成功与失败结果的即时反馈，对异常情况进行统一处理，注册界面整体效果如图 5-2 所示。

图

5-2 系统注册录界面

首系统首页在界面设计方面十分重视实用性以及操作便捷性二者的融合，其布局结构较为简洁清晰，用户可在较短时间之内定位到所需要的功能入口以及关键信息区域，借助统一的视觉风格以及模块划分，不同功能块在视觉上形成了清晰的分区，这样便降低了用户操作时的认知负担，界面构建运用了 **Element Plus** 组件库，这提升了开发效率，又保障了系统在交互风格上的一致性。首页以管理员身份作为例子展示了用户基本信息以及相关数据的布局方式，具体效果可见图 5-3。

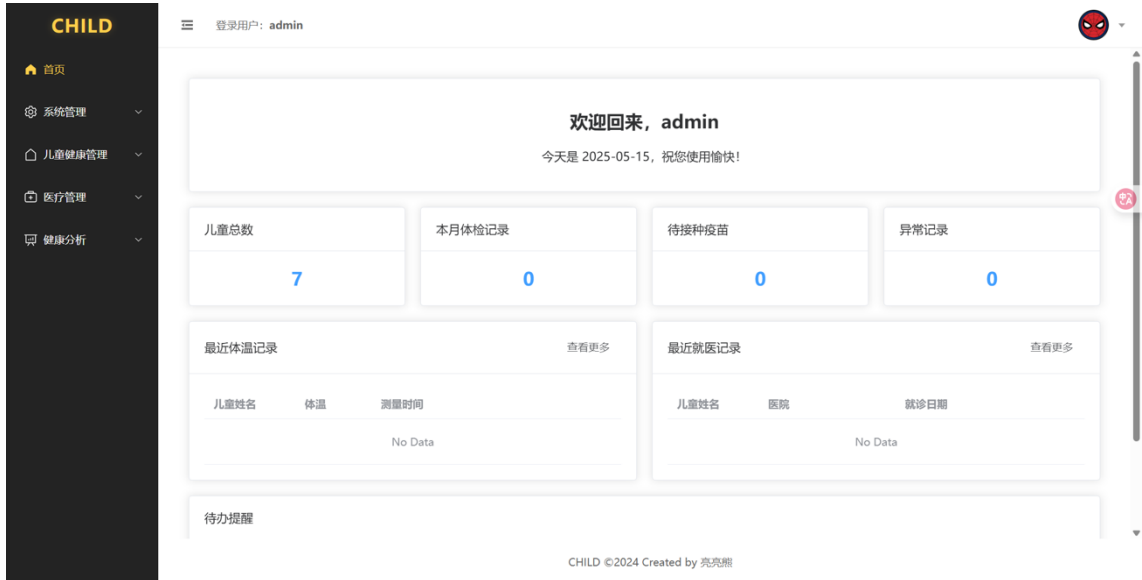


图 5-3 主界面展示

## 5.2 用户信息管理

用户信息管理模块运用了典型的前后端分离设计理念，其功能囊括了用户基本资料的查看、更新，密码修改以及头像上传等操作，从整体上保证了信息维护有灵活性，操作拥有安全性。

前端页面依据 **Vue3** 以及 **Element Plus** 搭建而成，主要借助 **UserInfo.vue** 达成用户个人资料管理功能。界面采用卡片式布局方式，把用户 ID、用户名、真实姓名、联系电话等字段划分到基础资料区域，并且提供可编辑表单用于修改，表单验证逻辑由 **Element Plus** 表单校验组件来完成，其中针对手机号字段运用正则表达式 `^1[3-9]\d{9}$` 实施格式校验，以此防止不符合规范的数据录入。

系统采用 **JWT** 进行用户身份校验，所有敏感请求需携带合法 **Token**，后端通过自定义拦截器 **LoginInterceptor** 拦截验证请求，并利用 **ThreadLocal** 实现用户会话信息在请求线程间的传递，以保障数据调用的准确性。

数据库使用 MySQL 管理用户数据，核心表 user 包含 user\_id（UUID 主键）、username、password（加密存储）、real\_name、phone、role\_id（与 roles 表外键关联）等字段，通过角色字段实现了权限控制逻辑，不同类型用户（如管理员、医生、家长）在接口访问与字段修改权限上均作了限制。系统同时引入了事务控制与异常处理机制，以提升数据操作的稳定性与一致性。

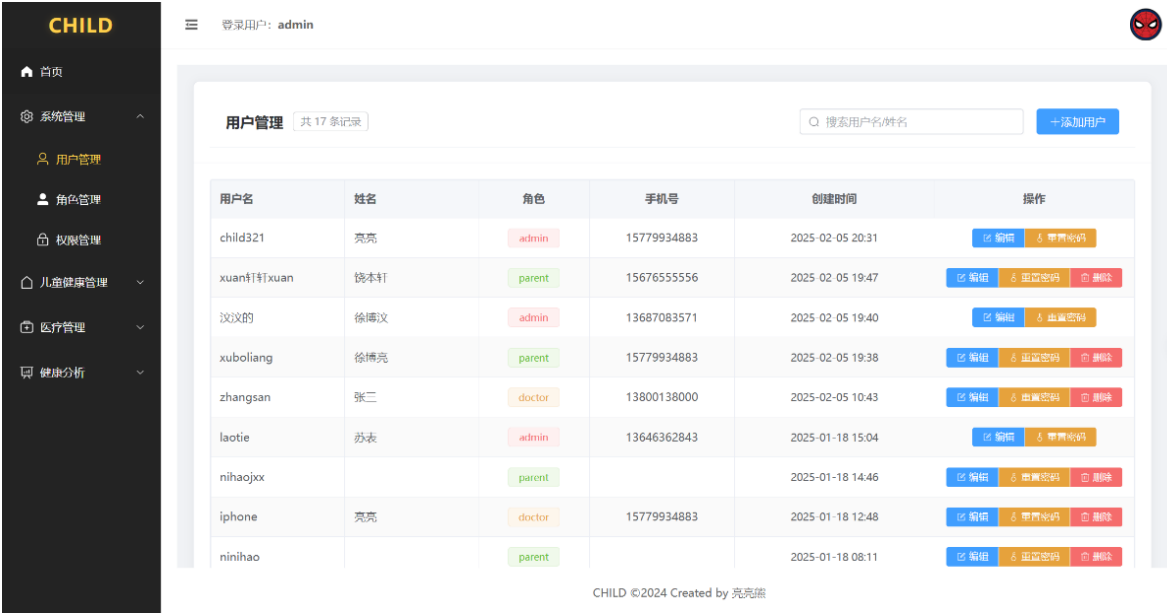


图 5-4 管理页面

添加用户的界面，如图 5-5 所示。

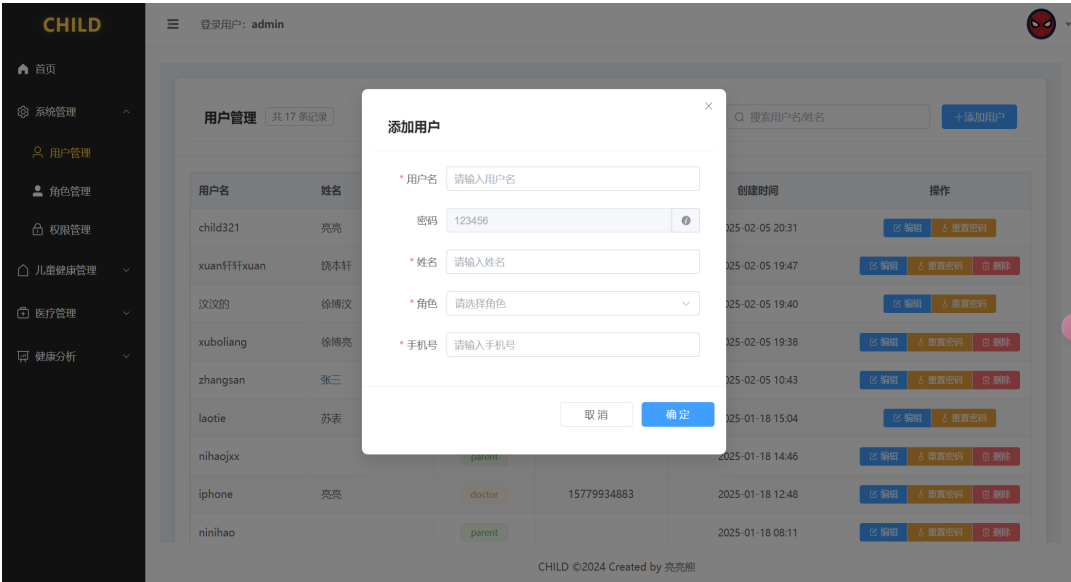


图 5-5 添加用户的界面

在用户管理模块里，“添加用户”以及“编辑用户”这两项功能是专门为管理员角色打造的，整个实现流程着重把控权限控制以及输入的合规性，前端部分，选用了 Element Plus 的 el-dialog 弹窗组件，借助弹出式表单的形式来开展用户信息的录入与修改操作，表单字段覆盖了用户名、密码、真实姓名、手机号以及角色选择等内容。

为保证输入符合规范，系统针对各个字段都设定了实时校验规则：用户名被限定为 5 至 16 个非空字符，密码也被限制在同样的长度范围内，并且禁止包含空格，手机号需要符合国内通用格式校验，真实姓名字段是必填项，不允许留空，角色选择借助下拉框来实现，选项内容依据当前用户权限进行动态加载，以此保证管理员在创建新用户时，不能赋予其超出权限范围的角色。

在业务逻辑层，UserServiceImpl 实现了用户添加与编辑的关键处理流程，添加用户时，系统会先校验用户名是否重复，接着使用 UUID 生成用户主键，再对密码进行 MD5 加密后存储，编辑用户时，系统会首先校验当前操作用户的权限等级，保证其无法对权限更高的账号进行修改。数据持久化由 UserMapper 接口完成，核心 SQL 语句如下：

-- 添加用户

```
INSERT INTO user (user_id, username, password, role_id, real_name, phone,
created_at, updated_at)

VALUES ({userId}, {username}, {password}, {roleId}, {realName}, {phone},
now(), now());
```

-- 更新用户信息

```
UPDATE user

SET real_name = {realName},

    phone = {phone},

    role_id = {roleId},

    updated_at = now()

WHERE user_id = {userId};
```

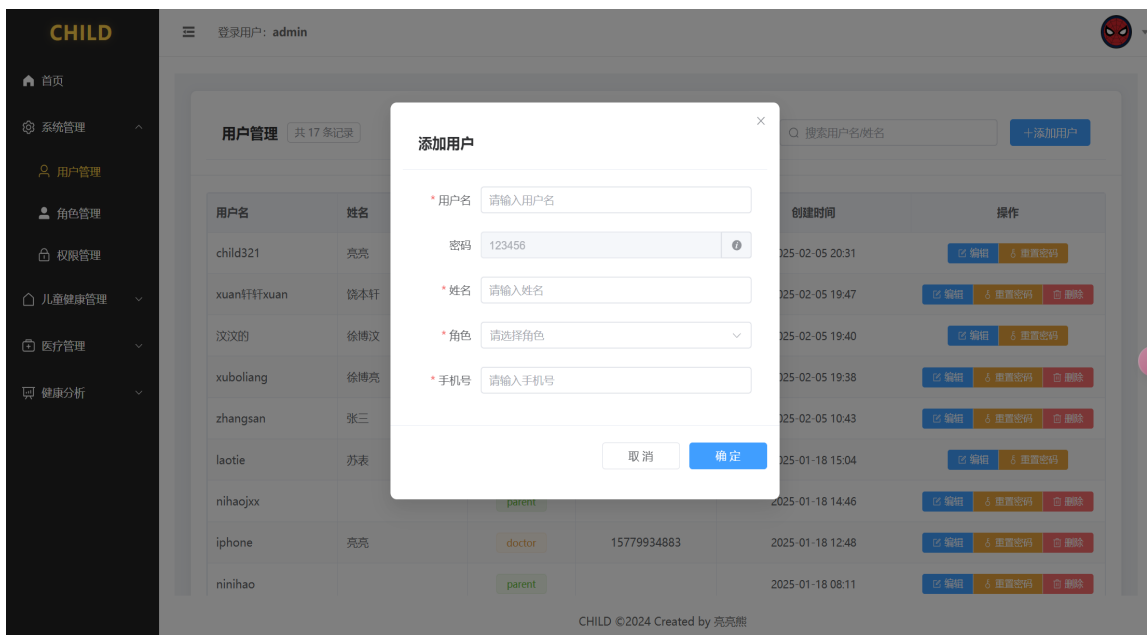


图 5-5 编辑用户信息的界面

## 5.3 儿童信息管理

### 5.3.1 数据库设计部分

本系统当中的儿童信息管理模块借助精心设计的数据库结构达成了对儿童成长数据的有效存储以及管理，其核心设计覆盖两个表，分别是儿童信息表以及成长记录表，儿童信息表以 `UUID` 作为主键，用来存储每个儿童的基本资料，像是姓名、性别、出生日期以及监护人信息等字段。为达成家长与儿童之间的关联，表中借助外键 `parent_id` 关联到用户表，以此保证每个儿童信息都可与对应家长的账户信息相匹配，成长记录表专门用于记录儿童在成长过程里各项关键指标的变化，例如身高、体重等数据，这些数据凭借 `child_id` 与儿童信息表中的记录建立联系，呈现出一对多的关系，也就是说一个儿童可能存在多条成长记录。这样的设计为系统提供了稳固的数据基础，支撑了儿童成长的持续追踪以及数据管理。

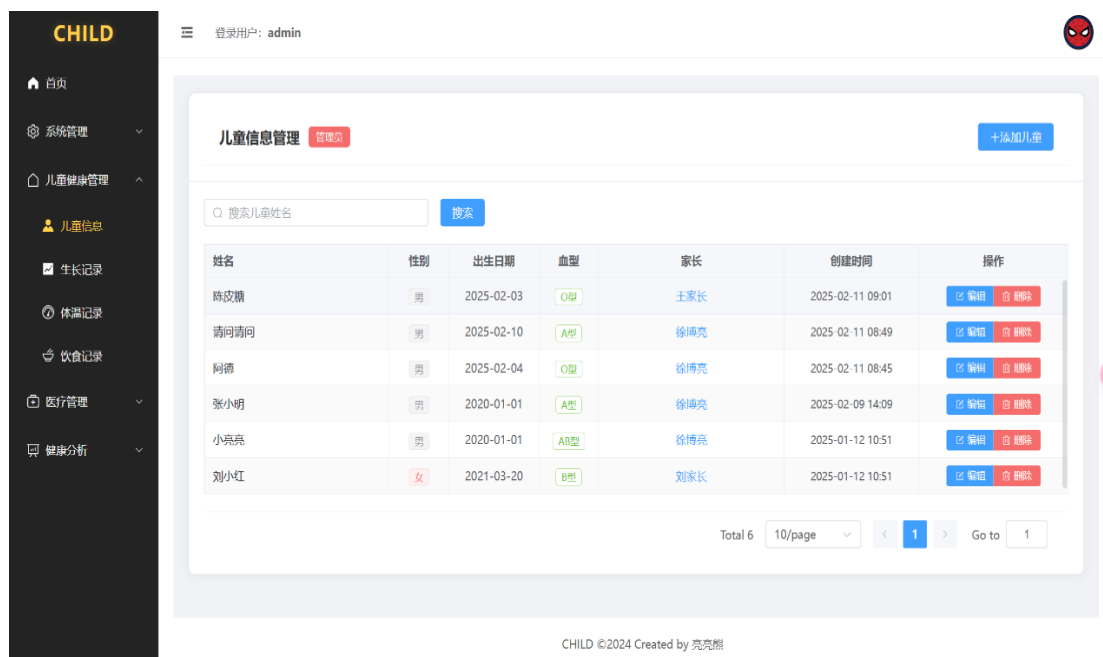


图 5-6 儿童信息管理的界面

## 5.4 儿童生长记录管理

儿童的生长数据管理是基于数据库里的 `growth_records` 表来设计并实现的，此表设有好些个字段，用来存儿童的身高、体重、测量时间等信息，主键是 `record_id`，借助 `child_id` 字段和儿童信息表关联起来，并且记录人能借助 `recorded_by` 字段和用户表建立联系。另外该表还设了 `bmi` 字段，这是依据身高体重算出来的体质指数，可用来评估儿童的生长状况，后端系统是用 `Spring Boot` 框架开发的，`GrowthRecordController` 负责处理跟生长记录有关的接口请求，具体功能有分页获取记录列表、按儿童 ID 查其历史记录、添加新记录以及修改已有记录等。在 `GrowthRecordServiceImpl` 里实现了必要的校验逻辑，像限制身高在 50 到 250cm 之间、体重在 2 到 200kg 之间，以此保证录入数据的合理性，还内置了 BMI 值的自动计算逻辑，防止前端重复计算，权限控制也被放进处理流程里，系统会校验当前用户有没有权限访问相关儿童的记录信息，保障数据安全。前端用 `Vue3` 框架搭配 `Element Plus` 构建交互界面，页面凭借 `el-table` 展示生长数据清单，还提供新增与修改记录的功能表单，表单输入项包含身高、体重和测量



日期等核心字段，界面整体布局挺合理，用户操作挺便捷，图 5-7 展示了该功能模块的实际页面效果。

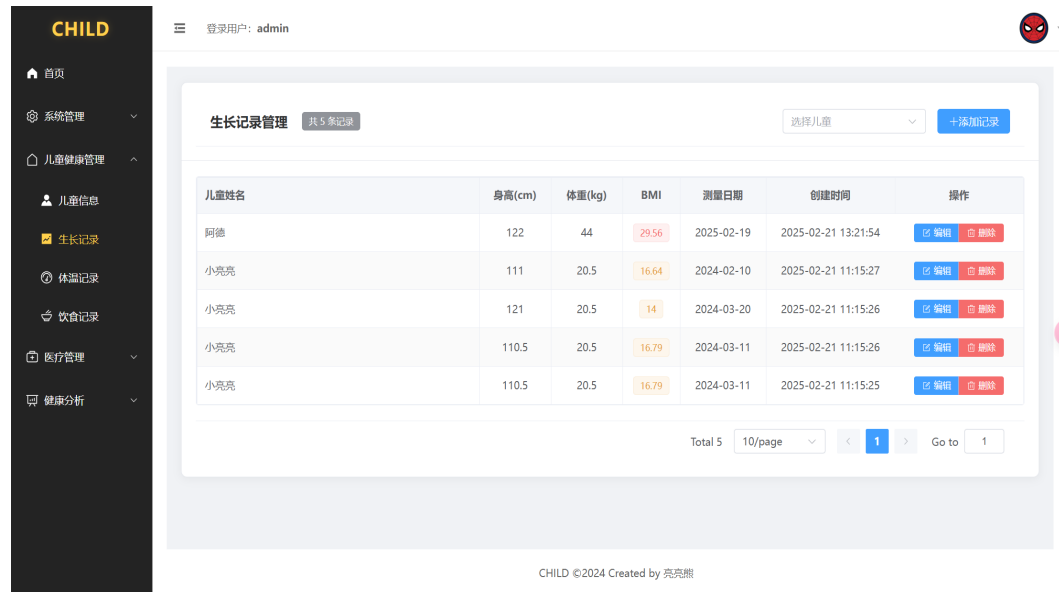


图 5-7 儿童生长管理的界面

## 5.5 儿童体温记录管理

儿童体温记录管理模块以数据库中的 `temperature_records` 表为核心构建，全面记录了儿童在不同时间和部位的体温测量数据。该表结构设计合理，主要字段包括：`record_id`（自增主键，用于标识唯一记录）、`child_id`（儿童 ID，外键关联儿童信息表）、`temperature`（体温值，采用 `decimal(3,1)` 类型以保留一位小数）、`measure_time`（测量时间）、`measure_position`（测量部位，枚举类型，支持“口腔 `mouth`”“腋下 `armpit`”和“耳温 `ear`”）、`notes`（备注信息）、`created_at` 和 `updated_at`（记录的创建与更新时间戳）。借助 `child_id` 字段同儿童表构建外键关联，以此保证数据结构有完整性以及数据之间存在联系，在后端进行实现时，系统依据 Spring Boot 框架的标准三层架构来开展开发工作，控制层由

`TemperatureController` 承担处理和体温记录相关的各类 HTTP 请求的任务，这些请求包含新增、查询以及统计等操作。服务层的 `TemperatureService` 实现了核心业务逻辑，主要有体温范围校验这一功能，也就是要保证录入的值处于  $35.0^{\circ}\text{C}$  至  $42.0^{\circ}\text{C}$  之间，同时还有异常体温识别功能，系统会按照不同的测量部位设定正常值区间，可自动判断测量结果是不是异常，提升了数据处理的智能化程度。数据访问层借助 MyBatis 框架开展数据库操作，保障了系统性能以及数据处理效率，图 5-8 呈现出了该模块的界面效果。

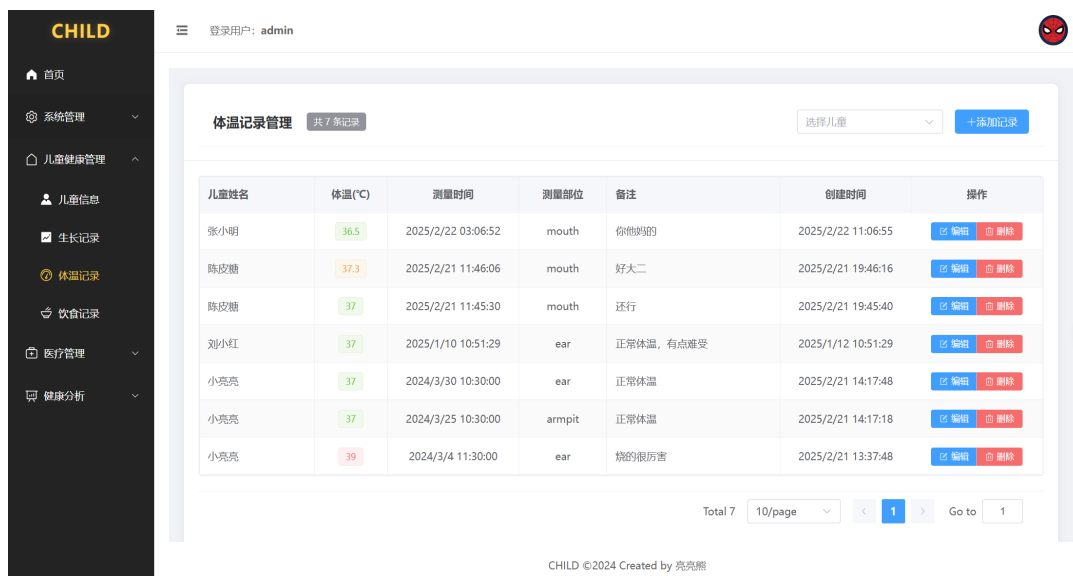


图 5-8 儿童体温记录管理的界面

## 5.6 儿童饮食记录管理

关于儿童饮食记录的管理工作，系统专门设计了一个称作 `diet_records` 的数据库表，用以存储与之相关的数据，此表所记录的信息包含记录编号即 `record_id`、所属儿童编号也就是 `child_id`、用餐时间 `meal_time`、餐别类型 `meal_type`、食物内容 `food_items`、摄入量 `amount` 以及备注 `notes` 等关键字段，并且借助外键把它与 `children` 表构建起一对多的关联关系，以此保证每一条饮食记录都可清晰地对应到某个儿童。后端运用 **Spring Boot** 框架来开展开发工作，控制层由 `DietRecordController` 负责处理前端发起的请求，这些请求功能包含饮食记录的查看、添加、编辑以及删除等。在服务层中，`DietRecordService` 实现了数据有效性校验、权限判定和记录之间的关联操作等关键逻辑，比如确保提交的用餐时间格式正确、食物项不能为空等。前端界面方面，系统使用了 **Vue3** 框架和 **Element Plus UI** 组件库来实现页面交互。用户可以通过饮食记录列表页面查看历史记录，支持添加新记录或编辑已有内容，所有操作都通过表单完成，整体操作流程简洁明了。页面中的用餐时间选择、餐别选项和备注填写等功能。系统对不同角色的用户也做了权限控制，比如家长账户只能管理自己孩子的数据，医生只能查看自己负责的儿童饮食信息，而管理员则可以访问全系统的所有记录。该模块的实际界面如图 5-9 所示。

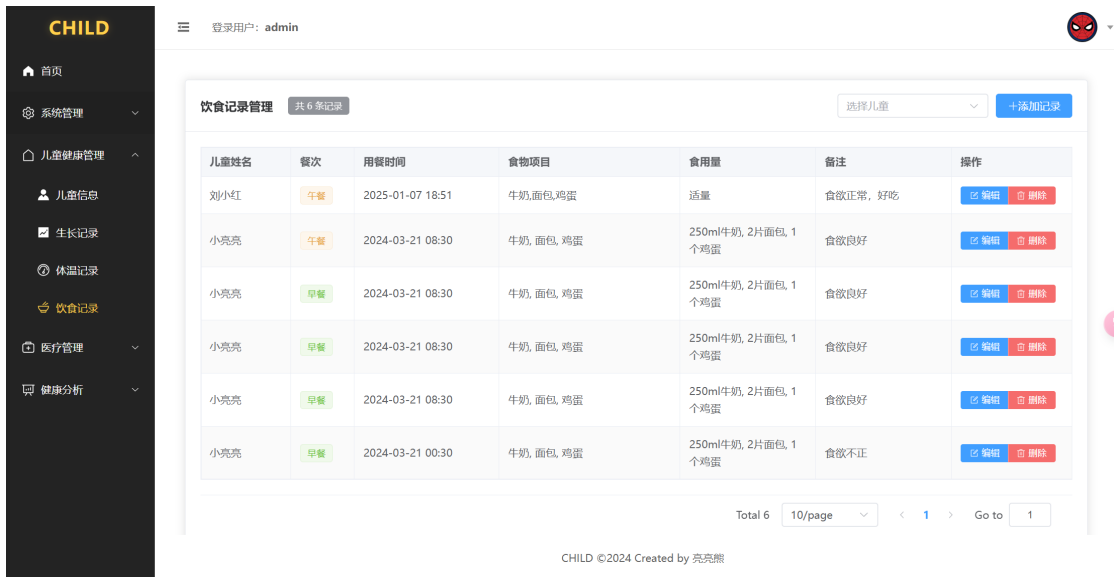


图 5-9 儿童饮食记录管理的界面

## 5.8 就医记录管理

本系统的就医记录管理模块采用前后端分离架构，后端以 `medical_records` 表为核心，记录儿童 ID、医生 ID、就诊时间、诊断结果、治疗方案、医院名称、科室及备注等信息。相关用药信息存储在 `medication_records` 表中，通过医疗记录 ID 建立关联。前端基于 `Vue3` 和 `Element Plus` 实现，就医记录页面使用 `el-table` 展示数据，添加/编辑功能通过 `el-dialog` 弹窗完成，表单验证覆盖就诊时间、医院、科室等关键字段。后端由 `Spring Boot` 提供 `REST` 接口，控制层通过 `MedicalRecordController` 处理请求，业务逻辑封装在 `MedicalRecordService` 中，包括权限校验和医生自动关联。数据访问层使用 `MyBatis`，支持多表查询。系统采用基于角色的权限控制，医生可添加修改记录，家长仅可查看其孩子的就医数据。该模块的实际界面如图 5-10 所示。

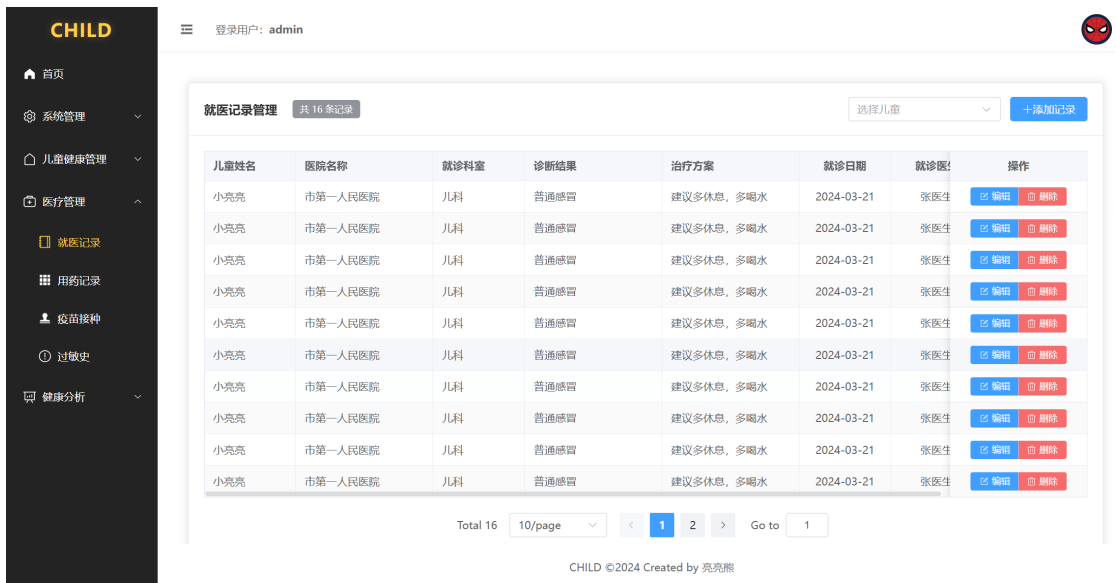


图 5-10 儿就医记录管理的界面

5.9 用药记录管理

本系统的用药信息管理模块采用前后端分离的设计理念进行开发。在数据库结构上，采用 medication\_records 表作为主要存储载体，以 record\_id 为自增主键，并通过 medical\_record\_id 外键与 medical\_records 表建立数据关联。该表记录了儿童用药过程中的关键信息，包括药品名称 (medicine\_name)、剂量 (dosage)、服药频率 (frequency)、用药起止日期 (start\_date、end\_date) 及备注说明 (notes)。系统会借助 created\_at 和 updated\_at 字段自动维护记录的创建时间与修改时间，以此提高数据的可追踪性以及历史回溯能力，前端部分是基于 Vue3 框架来开发的，主要页面处在 /views/medical/medications/index.vue 这个位置，依靠 Element Plus 组件库达成用户交互界面。药品记录展示运用 el-table 表格组件，并且支持依据就医记录编号、药品名称以及日期范围来进行过滤检索，数据录入与编辑是借助 el-dialog 弹窗组件完成的，利用 el-form 组件实现表单验证机制，针对药品名称、剂量、频率以及用药日期等核心字段设置了必填项以及格式校验规则。双向绑定采用 v-model 指令来保证数据和视图可实时同步，用药记录功能界面如图 5-11 所示。

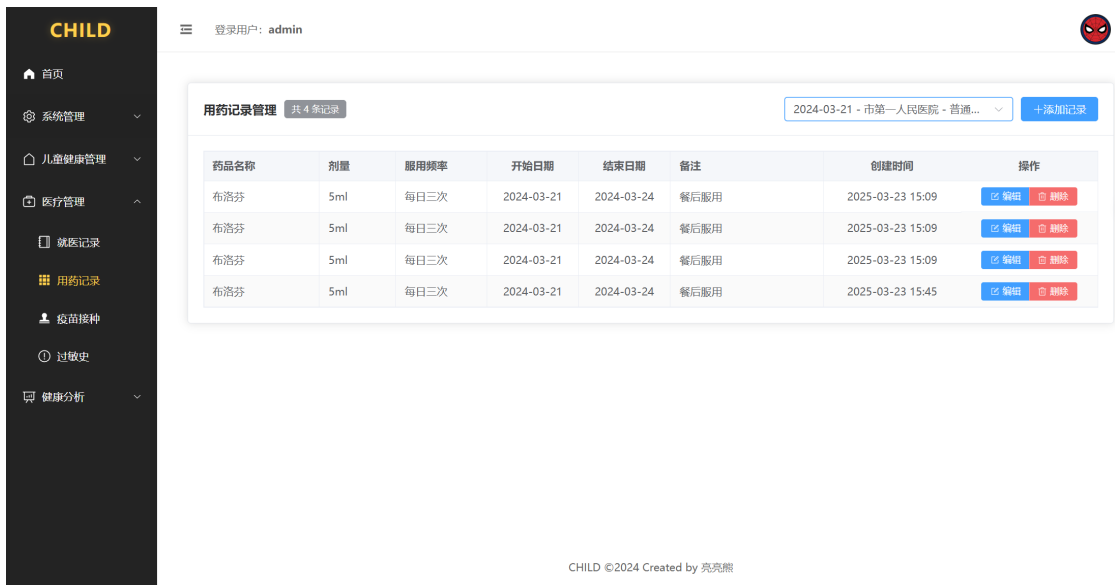


图 5-11 用药记录管理的界面

## 5.10 疫苗接种记录管理

本系统之中的疫苗接种管理模块主要是用来记录以及管理儿童的疫苗接种相关信息的，在数据库设计这一方面，运用 `vaccinations` 表来存储接种记录的基础信息，其中囊括了接种儿童、疫苗名称、接种日期、下次接种日期以及接种医生等关键信息，系统的前端页面借助 `Vue3` 框架进行开发，主要划分成三个功能区域：接种记录列表、接种信息录入表单以及接种日历提醒。在记录列表里面，家长可较为便利地查看孩子的历史接种记录，而医生则可添加并且管理接种信息，疫苗接种记录管理的界面，具体如图 5-12 所示。

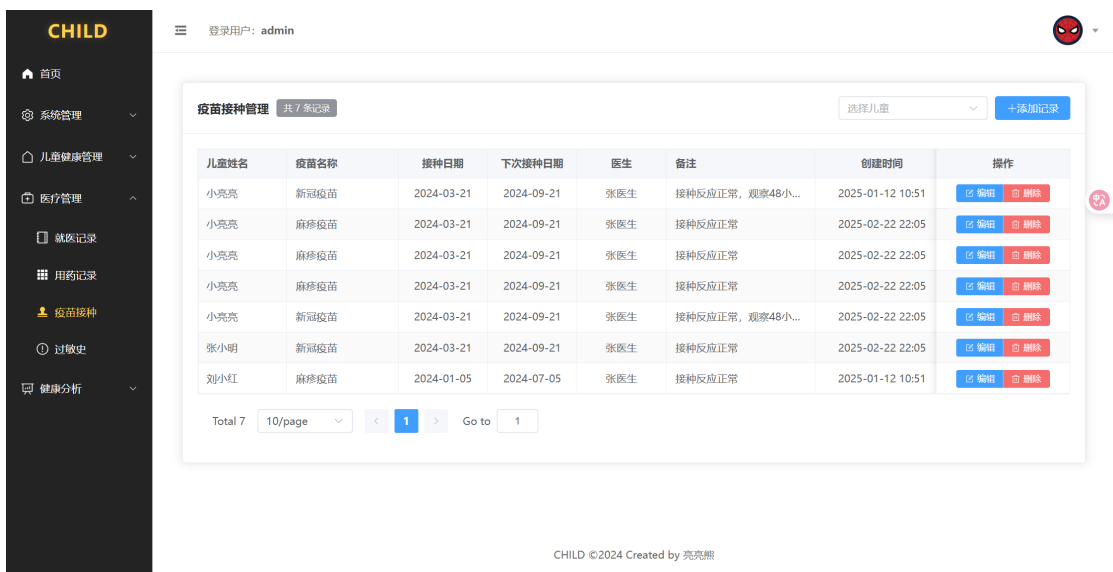


图 5-12 疫苗接种记录管理的界面

5.11 过敏史记录管理

本系统有过敏史记录管理功能,可对儿童过敏信息展开系统性收集与维护工作,在数据模型设计环节,系统借助 allergies 表实施数据存储操作,此表记录了诸多关键字段,如儿童编号,它与儿童基础信息表建立了外键关联,过敏源名称,过敏反应的严重等级,该等级划分为轻度、中度以及重度,诊断时间,以及备注等内容,以此保障数据的完整性与可追溯性。过敏信息管理功能的界面呈现如图 5-13 所示。

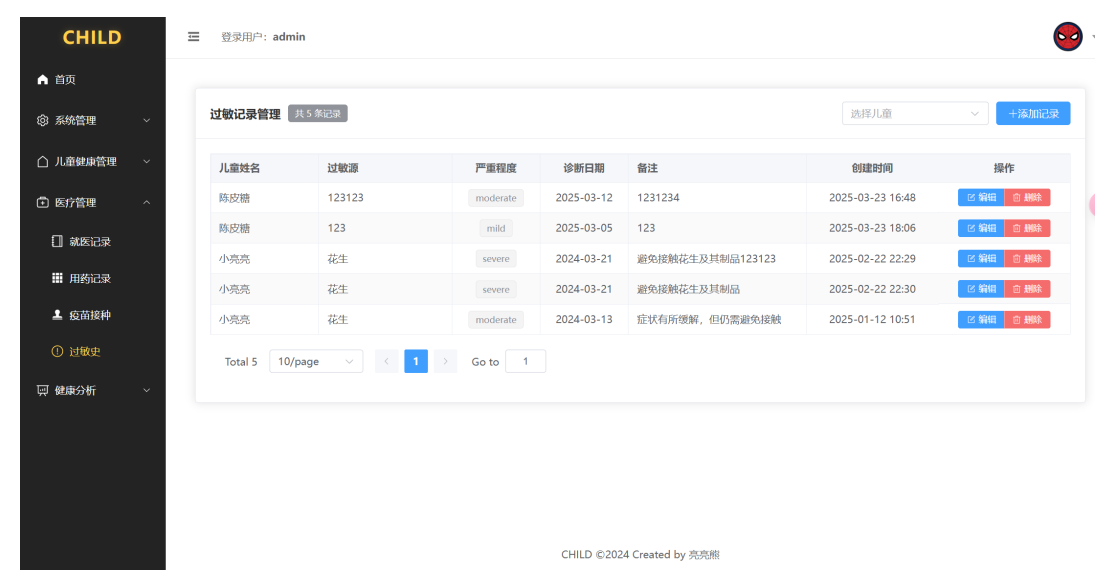


图 5-13 过敏史记录管理的界面

5.12 用户基本信息修改

本系统的用户基本信息编辑功能采用前后端分离的架构设计,数据库层面以 user 表为基础,主键使用 UUID 生成方式,记录用户的基本资料信息,包括登录账号、加密后的密码、所属角色编号、真实姓名以及联系电话等字段。前端采用 Vue3 框架开发,具体功能集成在 /views/user/UserInfo.vue 文件中。用户界面使用 Element Plus 的 el-card 组件进行结构组织,表单中包含用户编号、登录名称(均为只读)、角色信息、真实姓名以及联系方式等数据项,并通过 v-model 实现数据的双向绑定。表单验证方面,系统利用 Element Plus 提供的校验机制,对真实姓名字段设置了必填规则,手机号字段则采用正则表达式 ^1[3-9]\\d{9}\$ 进行格式校验。用户需点击“编辑”按钮才能激活可修改状态,状态由 isEdit 变量进行控制。用户基本资料管理界面如图 5-14 所示。

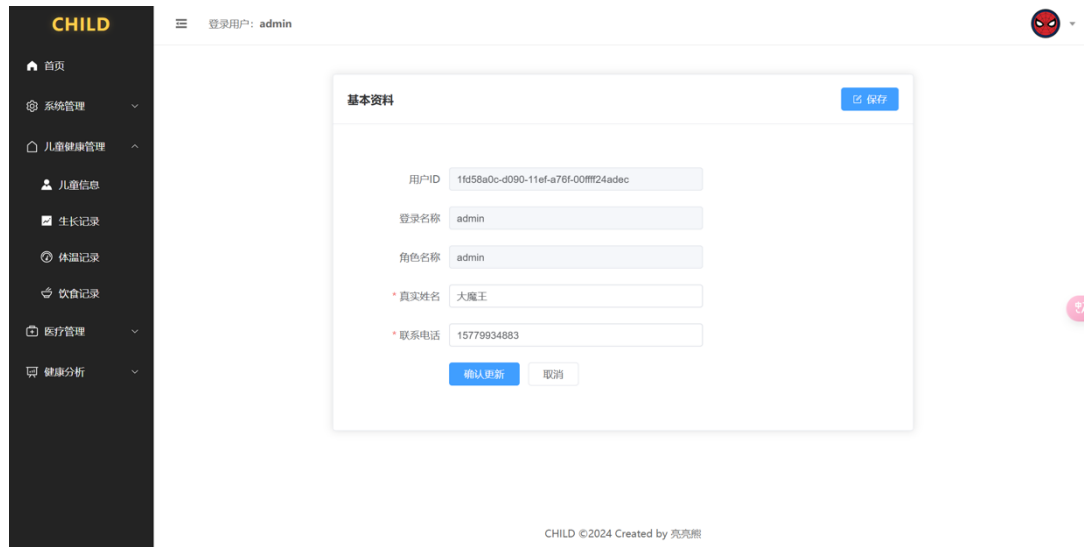


图 5-14 用户基本信息修改的界面

## 六 系统测试

### 6.1 测试目标

完成儿童健康跟踪管理系统开发工作后，要对系统开展全面测试，来评估其整体运行的稳定可靠程度，测试的核心来确认系统各项功能是否完全契合需求文档里的设定标准，具体覆盖：验证不同角色用户在权限控制方面的行为是否符合权限设计要求，检查系统里各类数据的准确和一致情况，评估平台的安全机制与运行稳定状况，并且从用户角度考量，测试界面交互的友好程度以及操作流程的合理状况。

### 6.2 测试策略

本系统于测试进程里运用了分层次且覆盖全面的策略以保障整体质量，起初在单元测试阶段，着重针对关键业务逻辑模块展开验证，生长记录与体温记录等和数据处理紧密相关的功能，保证服务层中各个方法的逻辑准确无误，并且检查数据访问层中 SQL 语句的执行状况是否契合预期。紧接着是接口测试，主要集中于系统所提供的 RESTful API，测试其功能是否完备，同时对错误响应机制是否合理、权限拦截是否生效等方面给予评估，开展了前端层面的功能测试，包括页面路由跳转是否顺畅、表单交互及数据提交是否正常、组件之间的联动逻辑是否清晰等内容。

### 6.3 测试任务

当我们的软件开发完成后，还没有给用户使用之前，程序开发人员需要做的工作就是要去发现开发出来的软件是否存在错误，一旦发现要及时作出修改。

程序在测试过程中需要做的任务有下面几点：

- (1) 测试程序就是需要在程序中发现 Bug，这是最基本的任务
- (2) 看一看所开发出来的软件在功能上面有没有缺陷
- (3) 开发出来的软件质量上面有没得保障
- (4) 系统使用者对系统功能的需求要一直不断的查看

系统测试工作包含以下四个方面，首先是用户身份认证相关测试，着重验证登录功能是否有效，token 生成与验证流程是否稳定，权限控制机制能否准确限制访问范围，其次针对儿童信息模块，测试儿童资料新增、编辑、删除及查询



---

操作是否正常，验证家长与儿童绑定关系准确性，确认数据库约束条件能否有效保证数据一致性与完整性。第三部分测试聚焦健康记录管理，覆盖生长数据、体温信息和饮食记录等模块功能是否健全，验证就医记录与用药信息关联是否正确实现，最后在系统安全性测试中，重点检查未登录用户访问拦截措施是否生效，敏感信息是否加密存储或传输。

## 6.4 测试结果

由于经验不足，写代码时出现了一些考虑不周的系统缺陷，写代码的时候会出现与设想不一致，比如说代码不规范导致接口与接口之间出现问题，功能与客户的要求不符合，这样导致系统不能过关。所以系统在上线前必须反复测试，经过反复测试，修改，再测试，再修改，系统才能够不断完善。在整个系统测试中，根据需求文档和设计文档，逐一对功能进行检测并写好测试用例，有效避免残片缺陷，因为系统出现缺陷不仅影响功能，而且可以导致数据的不准确，导致系统性能的降低，经过测试，才能使得系统的稳定性和成熟度得到极大的提升，系统性能也才有保证。

经过多轮的联调与优化，目前系统基本实现了初期设定的各项功能目标。在功能层面，各模块功能已全面落地，模块间逻辑顺畅、数据交互稳定，确保了整体业务流程的严密性。系统响应及时、页面渲染顺畅，后台数据库查询经过优化后效率可以，系统资源的使用保持在合理水平。在安全保障方面，系统不仅设置了粗略的角色权限管理，还对敏感数据传输进行了加密处理。

---

## 七 总结与展望

在本次毕业设计，我开发了一个儿童健康跟踪管理系统。该系统的主要目的是为了解决当前儿童健康信息管理中存在的若干问题，例如健康数据整理效率低下等健康状况。

我本系统采用了 Vue3、Spring Boot 和 MySQL 等技术进行开发。其中，Vue3 结合 Element Plus 实现了美观且易用的前端界面，为家长、医生和管理员提供了便捷的操作体验；Spring Boot 负责处理后端的各项功能，例如请求处理和业务逻辑管理；MySQL 数据库则用于存储各类儿童健康数据，确保数据的安全性和可靠性。

现在系统虽然完成了，但还有很多可以改进的地方。以后我想让系统功能更强大，比如加一些智能分析的功能。现在只是简单记录数据，以后可以用算法分析这些数据，提前发现孩子可能出现的健康问题，像根据孩子平时吃的东西、身高体重变化，预测会不会变胖，然后给家长一些具体的建议，像该让孩子吃什么、做多少运动。

---

## 参考文献

- [1] 王小明, 李华. 基于 Vue 的前端开发框架研究与实践[J]. 计算机工程与应用, 2020, 56(10): 12-18.
- [2] 张晓东, 刘强. Spring Boot 框架的应用与发展[J]. 软件学报, 2019, 30(6): 1023-1032.
- [3] 张华, 王蕾. 基于 MySQL 数据库优化的技术研究[J]. 数据库技术与应用, 2018, 26(4): 45-51.
- [4] 李四, 陈晨. 《现代软件工程》[M]. 北京: 清华大学出版社, 2021.
- [5] 刘强, 黄凯. Vue3 框架应用与前端开发优化[J]. 前端技术研究, 2021, 12(2): 45-50.
- [6] 刘智勇, 邓泽伟. 电子健康档案系统的设计与实现[J]. 信息技术与应用, 2020, 28(5): 66-72.
- [7] 高云. 《数据库系统概论》[M]. 北京: 高等教育出版社, 2019.
- [8] 张宇, 李浩. 面向大数据的儿童健康信息管理系统研究[J]. 数据分析与应用, 2019, 15(8): 99-104.
- [9] 陈力, 刘志刚. 基于云计算的健康管理系统架构设计[J]. 云计算技术与应用, 2019, 7(3): 123-130.
- [10] 王浩, 杨林. 移动医疗健康应用技术研究及展望[J]. 现代医疗技术, 2020, 35(1): 45-50.
- [11] 李明, 张婷. 基于 Vue 和 Spring Boot 的电商平台系统设计与实现[J]. 电子商务与技术, 2021, 14(4): 75-81.
- [12] 刘强, 王永刚. 基于 Spring Boot 的微服务架构设计与实现[J]. 计算机与通信, 2018, 26(12): 32-39.
- [13] 赵宏, 陈悦. 基于数据分析的健康管理系统研究[J]. 电子信息技术与应用, 2021, 32(7): 11-17.
- [14] 邱岩, 徐杰. 基于 Spring Boot 的 RESTful API 设计与实践[J]. 软件开发与技术, 2020, 16(5): 58-64.
- [15] 张涛, 刘鑫. 基于 Vue 的单页面应用性能优化研究[J]. Web 开发技术, 2020, 22(2): 50-56.
- [16] 罗丹, 王俊. 基于 MySQL 数据库的性能优化技术[J]. 数据库与系统管理, 2020, 33(6): 88-92.
- [17] 陈飞, 李丽. 基于大数据的儿童健康管理信息系统[J]. 大数据技术与应用, 2021, 10(4): 41-47.
- [18] 周强, 黄宇. 互联网+健康管理模式探析[J]. 网络健康技术, 2020, 18(3): 61-66.
- [19] 王辉, 张梅. 基于 Vue3 的现代 Web 开发技术研究[J]. Web 技术与应用, 2021, 15(2): 25-31.
- [20] 刘锐, 徐辉. 基于 Spring Boot 的健康数据管理系统设计[J]. 计算机系统设计与应用, 2019, 25(4): 75-81.
- [21] 孙浩, 刘峰. 基于 Vue 和 Spring Boot 的图书管理系统的设计与实现[J]. 信息技术与管理, 2020, 18(9): 102-108.

- 
- [22] 王彬, 赵磊. 基于 RESTful 架构的医疗健康管理系统设计与实现[J]. 信息系统与管理, 2021, 29(1): 67-73.
- [23] 高峰, 刘明. Vue3 与 Element Plus 技术在企业管理系统中的应用[J]. 软件与网络技术, 2020, 28(5): 99-105.
- [24] 李瑞, 王宏. 面向云平台的儿童健康管理系统设计实现[J]. 云计算与大数据, 2019, 18(7): 35-42.
- [25] 张剑, 陈文. 数据库安全性与健康信息保护研究[J]. 信息安全与技术, 2020, 15(2): 43-49.

---

## 致谢

撰写本次毕业设计论文时，我历经了从构建系统架构到实现功能模块，再到撰写与完善论文的过程，这是对我本科四年所学知识的综合检验，也是一次富有挑战且收获满满的实践经历，在此我诚挚感谢所有在这段历程中给予我帮助与支持的人。

诚挚感谢我的指导教师章新友教授，在毕业设计全过程中，您于论文撰写方面给了我许多宝贵建议。您耐心细致的指导与严谨的治学态度让我获益匪浅，正是在您悉心指导下，我才顺利完成此论文。

我要感谢计算机学院的各位授课教师，在四年学习里，各位老师的谆谆教导使我掌握了计算机科学与技术领域的核心知识，为本次毕业设计奠定了坚实理论基础，从数据结构、算法、数据库管理到前后端开发、系统架构设计，这些知识为我独立完成本系统开发工作提供了关键支撑。

衷心感谢江西中医药大学计算机学院为我提供的学习平台，让我能在本科四年系统学习计算机专业知识，本次毕业设计是对我大学四年学习成果的集中展示，也是对我个人独立开发能力的关键实践，在项目中，我学习并应用了 Vue3 + Element Plus 前端技术栈、Spring Boot + MySQL 后端架构，结合 RESTful API 实现了前后端的数据交互。这段经历让我对系统开发有了更系统化的理解，也让我更明确了未来发展方向，本项目顺利完成离不开各方支持与帮助，我会一直铭记这段充实而宝贵的学习经历，并在未来学习和工作中继续努力，不断提升自己。

毕业并非终点，而是新起点，在未来学习和工作中，我会继续保持对技术的热爱，不断学习新知识，提升自己开发能力，并将所学运用到实际项目中，为社会创造更大价值。

---

## 个人简介

### 一基础信息

姓名：徐博亮

性别：男

出生日期：2022 年 09 月

民族：汉族

籍贯：江西省上饶市

政治面貌：共青团员

学校：江西中医药大学

专业：计算机科学与技术

### 二教育经历

2021 年 9 月-2025 年 6 月，江西中医药大学，计算机科学与技术，本科。