**实训报告**

题目**：** **糖尿病预治智能助手**

|  |  |
| --- | --- |
| 学 院 | 计算机科学技术 |
| 专 业 | 计算机科学与技术 |
| 班 级 | 计科04 |
| 学 号 | 2022204257 |
| 姓 名 | 付康 |

2025年 7 月 12 日

# 第一章 系统概述

## 1.1 背景介绍

## 在当今全球化的时代，糖尿病已然成为一个令人忧虑的全球性健康难题。随着生活方式的急剧转变，如高糖高脂饮食的普遍化、运动量的大幅减少，以及人口老龄化进程的加速，糖尿病的发病率正以惊人的速度持续攀升。世界卫生组织的数据显示，全球糖尿病患者数量已达数亿之多，并且这一数字仍在不断增加。糖尿病不仅会引发各种严重的并发症，如心血管疾病、肾病、失明等，严重威胁患者的生命健康和生活质量，还会给社会和家庭带来沉重的经济负担，包括高额的医疗费用、长期的护理成本以及生产力的损失。因此，如何科学有效地预防和治疗糖尿病，成为了全球公众和医疗界共同关注的焦点问题。

## “糖尿病预治智能助手”创新性地融合了 DeepSeek和 Dify的前沿智能体开发技术，精心构建了一个一站式的糖尿病预治管理体系。通过深度对接专业医学知识库，并运用DeepSeek大模型，能够对用户输入的健康数据进行全方位、深层次的分析，进而精准预测糖尿病风险。不仅如此，该助手还会根据用户的具体健康状况和糖尿病类型，量身定制个性化的防治方案，涵盖科学合理的饮食计划和针对性强的运动指导，助力用户有效管理病情。除此之外，还贴心地设置了与糖尿病领域专家在线交流的名医对话界面，让用户能够随时随地获得专业、权威的医疗建议。

## 1.2 产品使用范围

本平台面向普通人群与糖尿病患者，围绕“防、筛、管、助”四大场景，打造一站式、全周期健康管理服务。打开首页，权威科普专区滚动呈现糖尿病病因、并发症与日常误区，让大众“刷手机”即可掌握防病要点；点击“在线咨询”，三甲内分泌科医生实时接诊，图文、语音、视频皆可，患者无需排队即可得到专业指导。

平台内置 AI 风险引擎，可读取血糖、BMI、家族史、生活习惯等十余项指标，秒级输出个人糖尿病 3～10 年发病概率，并给出可视化风险雷达图。针对已确诊或高风险用户，系统会基于每日上传的血糖、饮食、运动、睡眠数据，自动生成个性化“控糖生活方案”：每餐主食克数、餐后运动时长、用药提醒、情绪管理建议一应俱全，且每周根据疗效动态调整。

为了让健康管理“看得见、坚持住”，平台提供“生活打卡”工具，用户可一键记录血糖、饮食、步数、饮水量等，系统自动生成周/月趋势报告；打卡信息管理中心支持数据补录、异常提醒、目标校准，帮助用户养成长期习惯。个人中心则集成档案管理、设备绑定、消息订阅、隐私设置，所有健康建议均可按个人口味、作息自由微调。

更贴心的是，平台搭载“糖尿病助手智能体”，以自然语言对话方式 7×24 小时在线：科普问答、指标解读、方案修改、用药疑惑，一句话即可触发 AI 精准回复，并可无缝转接真人医生。无论是想远离糖尿病的普通人，还是需要精细控糖的患者，都能在这里获得科学、个性、可持续的健康守护。

# 第二章 系统需求分析

## 2.1 任务概述

## 在进行可行性分析之前，应该对于可行性分析的含义和步骤进行大概的了解。对于可行性分析，我认为它的作用是要理清本项目中宿舍管理系统的各种需求和背景，主要内容包括：本项目的研究和开发目的，具体的需求和需要实现的功能，学校运行本系统的硬件设备和软件环境。在了解清楚这些之后，就需要对于以上的需求和背景进行分析，之后，制作出宿舍管理系统的物理模型，并抽象出逻辑模型。模型出来后，就要去思考系统的具体运行逻辑，并得出数据流图和数据字典，画出必要的程序示意图，如：顺序图、流程图等。还要写出数据字典和算法描述等，这样才是一个软件系统正规的开发顺序和开发要求。

## 2.2 整体项目架构

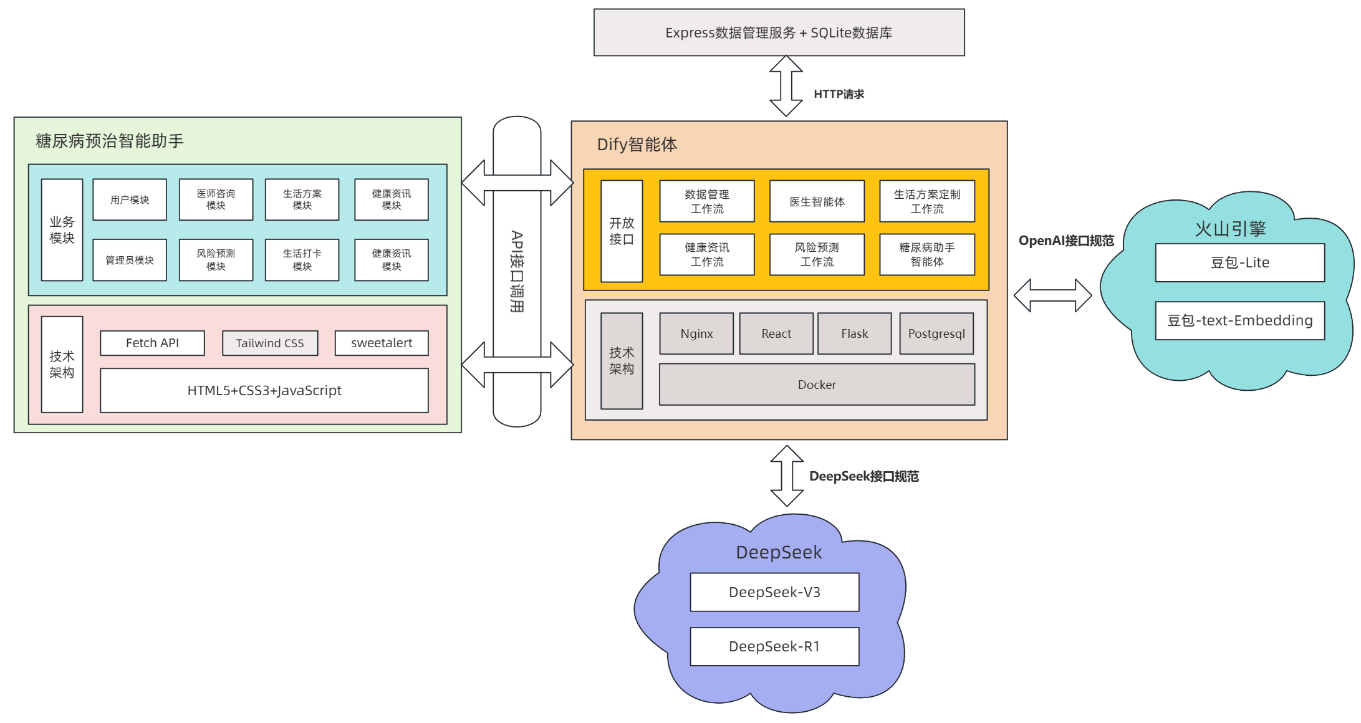


图1 项目架构图

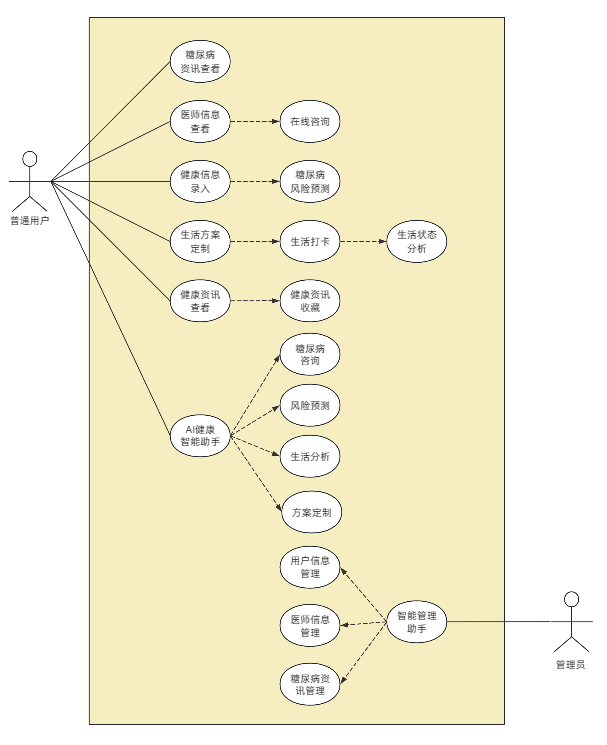


图2 系统用例图

## 2.3 可行性分析

《糖尿病助手》专注服务于糖尿病患者及相关人群。依托强大的大模型能力，它涵盖了糖尿病相关内容咨询、风险预测、个性化生活方案定制以及生活打卡等多项功能。平台可依据用户输入的信息，精准预测其患糖尿病的风险。借助智能体技术，用户能与 AI 医生展开在线对话，获取专业医疗建议。同时，平台会结合用户日常生活习惯，生成专属生活方案。用户可依照方案进行每日打卡，平台还能依据打卡信息，深入分析用户近期生活状态，助力用户更好地管理糖尿病。

糖尿病管理系统借助先进的传感器技术与移动互联手段，实现对患者血糖、血压、心率等生理数据的自动化采集与实时传输。与传统患者手动记录、就诊时提交数据的方式不同，该系统能通过可穿戴设备、家用智能检测仪器，不间断地收集多维度健康数据，如日常饮食摄入详情、运动轨迹与时长等。利用智能算法对这些数据进行即时分析，不仅保障了数据的精准性与及时性，还让医生能随时掌握患者身体状况，极大减少了患者手动记录的麻烦与数据遗漏、错误的可能性。

系统运用大数据深度分析和深度学习技术，对海量患者健康数据进行挖掘，探寻数据间潜在联系与规律。通过构建专业的并发症风险评估模型，系统能够精准预测糖尿病患者发生各类并发症的风险，例如糖尿病肾病、视网膜病变等。相较于传统仅依靠医生经验和简单指标判断病情风险的方式，新系统在风险防控方面展现出更高的精准度与科学性，助力医生提前制定干预方案，降低并发症发生几率。

糖尿病管理系统实现了从预约挂号、线上问诊、检查检验预约到开具处方、药品配送的全流程线上化操作。患者无需多次前往医院排队，在家就能与医生进行视频面诊，上传近期健康数据。与传统繁琐的线下就医流程相比，新系统大幅简化操作步骤，减少患者候诊时间，提升就医效率。同时，系统支持部分检验结果自动审核与智能诊断辅助功能，进一步提高医生诊疗效率。

该系统支持医生远程管理患者病情，即便患者身处异地，医生也能依据实时传输的数据，对治疗方案进行调整。系统还能生成详细的健康报表与趋势图表，如血糖波动曲线、饮食运动达标情况等，为医生制定个性化治疗决策提供数据支撑。相较于传统依赖患者定期到院、医生凭有限信息判断的管理方式，新系统在管理的便捷性与决策的精准性上优势显著，让患者能随时获得专业指导。

随着人们健康意识的提升以及老龄化社会的加剧，糖尿病患者数量持续上升，对高效、便捷的糖尿病管理服务需求愈发迫切。同时，国家大力推动 “互联网 + 医疗健康” 政策，鼓励医疗科技创新应用于慢性病管理领域，为糖尿病管理系统的发展营造了良好环境。此系统正是顺应这一市场需求的关键工具，能为糖尿病患者提供全方位、全周期的健康管理支持 。

伴随数字技术的迅猛发展与医疗健康产业的深度变革，糖尿病管理系统将迎来更广阔发展空间与丰富应用场景。未来，系统不仅局限于糖尿病日常管理，还将拓展至糖尿病前期筛查、康复护理指导等领域，为患者提供全病程服务。同时，系统将与 5G、人工智能、虚拟现实等前沿技术融合，进一步提升智能化水平，例如通过虚拟现实技术为患者定制沉浸式运动康复方案，增强患者管理疾病的参与度与依从性，全方位提升糖尿病管理的质量与效果。

## 2.4 需求分析

## 2.4.1 功能分析

|  |  |
| --- | --- |
| 功能类别 | 子功能 |
| 首页 | 在线问诊 |
| 文章科普 |
| 糖尿病科普 |
| 方案定制 | 运动方案展示 |
| 生活方案展示 |
| 每日打卡 |
| 方案生成 |
| 健康资讯 | 健康资讯标签展示 |
| 健康资讯文章展示 |
| 健康资讯文章收藏 |
| AI助手 | 糖尿病信息咨询 |
| 生活方案生成 |
| 糖尿病风险预测 |
| 用户信息修改 |
| 个人中心 | 登录/注册 |
| 个人信息管理 |
| 糖尿病风险预测 |
| 打卡管理 |
| 管理员AI智能管理 |

## 2.4.2 性能分析

站在最终用户的视角，软件性能就是“我点一下到看见结果”的等待时长——从手指离开屏幕到界面给出明确反馈的全部时间；越短，体验越丝滑。

对运维工程师而言，性能不仅是单个用户的“秒开”，更是“万人同时在线”时的系统表现：服务器扛不扛得住、CPU/内存/网络有没有瓶颈、数据库能否撑到峰值、配置和索引还能不能再调、集群能不能横向扩展，以及 7×24 小时持续运行会不会掉链子。

而开发工程师眼里，性能贯穿整个生命周期：

算法是否高效，必要时用 buffer 减少 I/O；

架构是否可水平扩展、无单点；

编码规范是否避免内存泄漏、死锁、资源竞争；

数据库设计、查询、缓存是否最优；

性能场景是否能自动化、可重复测试。

## 2.5.2 开发环境简介

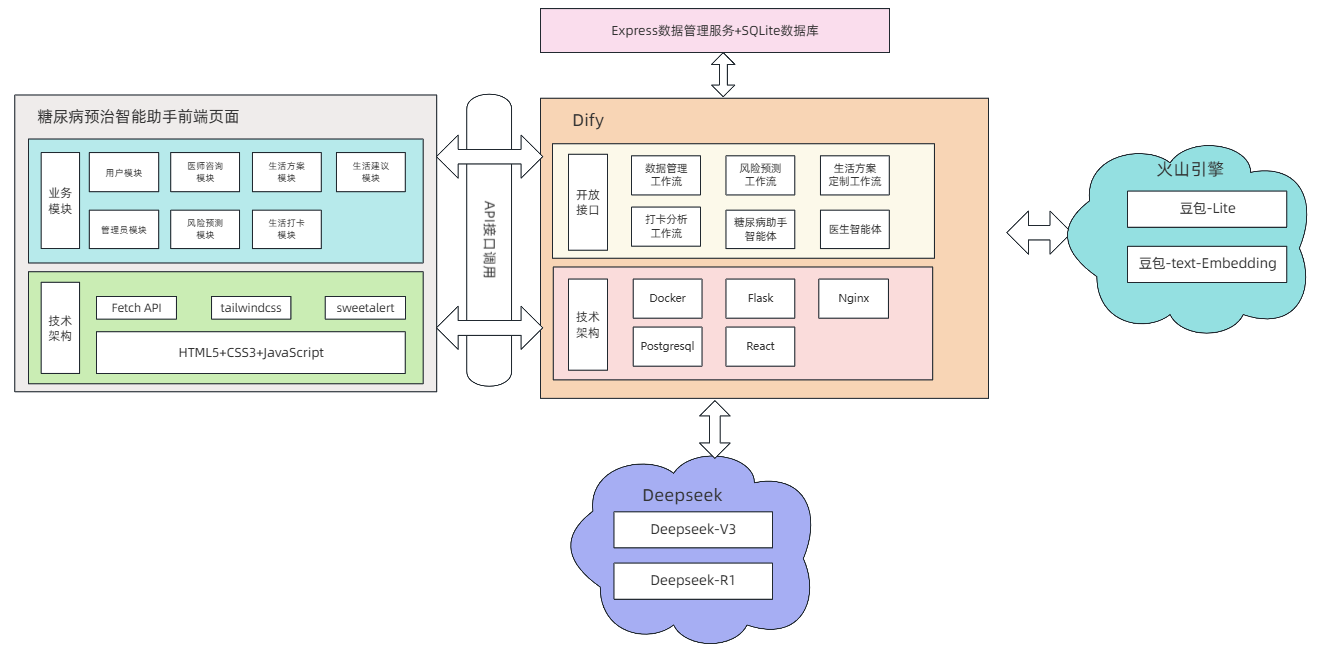
## 系统开发运行环境

虚拟机搭建dify环境。

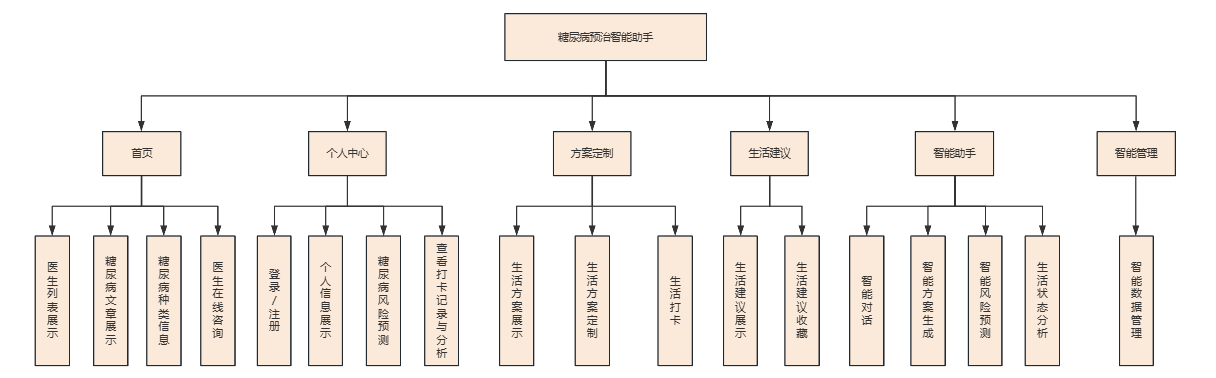
# 第三章 系统总体设计

## 3.1 系统总体功能设计

## 3.1.1总体系结构设计



## 3.1.2 功能设计



## 

## 3.2 界面设计

|  |  |  |  |
| --- | --- | --- | --- |
| **模块名** | **界面名称** | **界面标识** | **功能描述** |
| 首页 | 主页 | home.html | 医生列表展示、科普文章展示、糖尿病类型展示 |
| 在线咨询页面 | chat.html | 在线咨询页面中输入自己想问的问题，等待医生对问题进行回答 |
| 文章详情页面 | article.html | 点击文章封面跳转至文章详情页面 |
| 糖尿病类型页面 | diabetes.html | 点击糖尿病类型，跳转对应类型信息页面，在此页面中展示发病机制，临床表现，治疗 |
| 方案定制页面 | 方案展示页面 | scheme.html | 展示饮食管理与运动建议方案，可进行打卡，点击“调整生活方案”则跳转至方案调整界面。 |
| 方案跳转界面 | getScheme.html | 在此页面中可以选择自己的生活习惯，并给出自己对生活方案的方案建议，点击“生成”则生成生活方案信息。 |
| 生活建议 | 生活建议页面 | lifeAdvice.html | 根据用户当前状态展示生活建议信息标签，包括饮食建议、运动建议、生活指南、糖尿病科普 |
| 建议详情页 | lifeAdviceInfo.html | 点击建议标签跳转至建议详情页面，在详情页面中可以看到AI根据建议标签与用户状态生成的建议内容 |
| AI助手 | AI助手页面 | userAI.html | 用户在此页面中可以与AI助手聊天，AI助手功能包括为用户定制生活方案，给用户生活建议，解答用户与糖尿病相关的问题，为用户做糖尿病风险评估等。 |
| 个人中心 | 个人中心页面 | mine.html | 当未登录时点击个人中心，跳转至登录页面，已登录后则展示用户名称和菜单栏，包括个人信息、我的方案、我的建议、打卡记录、帮助中心等。 |

# 第四章 系统主要模块详细设计及实现

## 4.1 系统主要模块

## 4.1.1系统首页

进入MasterGo工作台（https://mastergo.com/files/home）然后完成登录，然后新建一个设计文件

在新页面中点击“AI生成界面”。

在AI生成界面，切换生成界面为“移动端”，输入提示词，系统会根据提示词生成详细的页面结构设计建议，如需修改，可继续提出完善意见提示词，系统会反复修改，满意后点击最下方的开始生成按钮，系统会生成UI界面。

一般界面生成提示词可以遵循以下结构：

a）是什么：该页面是xxx页面，用于xxx功能;

b）包括什么：该页面最上方为logo展示，上方为xxx，......;

c）什么风格：整体风格以浅蓝和白色调为主，页面简洁美观。

主页为糖尿病预治智能助手，界面最上方为logo与搜索栏，上方为轮播图，中间部分为专业医师团队，有三位医师，每个医师都展示头像、名称、科室、以及“立即咨询”按钮，医师团队下方为健康科普栏，展示健康科普文章，包括文章封面、文章标题、文章简介及浏览量，再下方为糖尿病类型栏，包括糖尿病的封面，名称及简介。最下方为菜单栏，包括“首页”，“方案定制，“生活习惯”，“A助手“及“个人中心”，整体以浅蓝色调和白色为主。

选取一张设计较好的，继续在聊天栏中与其对话，进行修改（注意本项目界面设计并不要求与界面设计参考图完全一致，可根据参考图自由发挥设计）。

生成一版比较满意的界面后，点击右下方的按钮，展开代码，然后点击“复制代码”，并在项目工程中创建“index.html”文件，将代码粘贴到此文件中。

创建糖尿病科普文章页面，页面包括标题、文章封面、正文内容、发布人、发布时间与点赞、评论、查看数量。界面以蓝白色调为主，简洁美观。

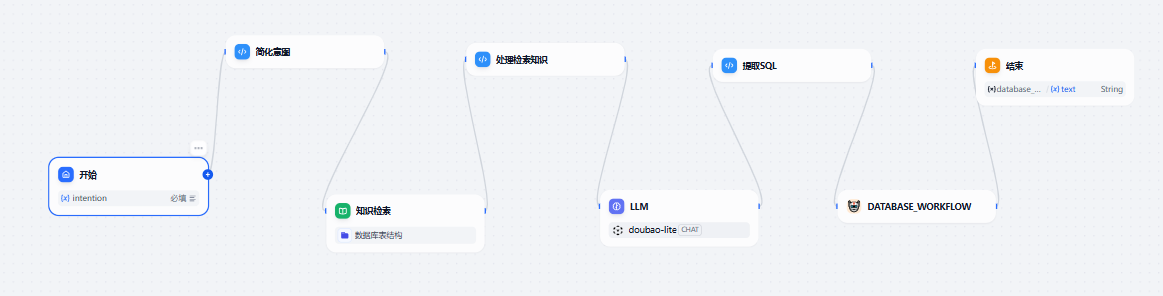
创建糖尿病类型展示界面，界面包括标题，糖尿病类型封面、发病机制、临床表现、治疗方案界面需要清晰分布信息，以蓝白色调为主，整体界面简介美观。

### 4.1.2 医师咨询

使用AI工具开发首页界面:

1. UI界面设计
2. 使用MasterGo设计UI界面
3. 使用DeepSeek&Cline优化代码

使用Dify开发数据管理工作流:



使用DeepSeek实现首页数据展示:

1. 通过数据管理工作流实现首页面数据加载功能
2. 首页数据缓存功能实现
3. 实现文章详情加载功能
4. 实现糖尿病类型加载功能

### 个人中心

使用AI工具设计个人中心页面与登录注册页面:

1. 个人中心页面设计
2. 登录注册页面设计

使用DeepSeek实现登录注册与个人中心展示功能:  
1. 登录注册功能实现

2.个人中心用户显示功能设计与功能实现

### 4.1.4 个人信息及糖尿病风险预测

使用AI工具设计个人信息及糖尿病风险预测页面:  
1. 功能分析

2.使用AI工具设计个人信息界面

3.使用AI工具设计其他页面

使用Dify开发个人信息管理及风险预测工作流:

### 4.2关键代码

个人中心加载方法

window.onload = function() {

    const user = getUserInfo()

    console.log(user)

    if (user) {

        document.getElementById('userAvatar').src = user.avatar\_url || '/img/user2.png'

        document.getElementById('userName').textContent = user.username || '用户'

        // Hide admin menu if not admin

        const adminMenuItem = document.querySelector('#menuList div:nth-child(6)')

        if (adminMenuItem && user.username !== 'admin') {

            adminMenuItem.style.display = 'none'

        }

    }

    // Add logout functionality

    document.getElementById('logoutBtn').addEventListener('click', function() {

        localStorage.removeItem('user')

        top.location.href = '/login.html'

    })

}

用户登录与注册实现

tailwind.config = {

    theme: {

        extend: {

            colors: {

                primary: '#2563eb',

                secondary: '#60a5fa'

            },

            borderRadius: {

                'none': '0px',

                'sm': '2px',

                DEFAULT: '4px',

                'md': '8px',

                'lg': '12px',

                'xl': '16px',

                '2xl': '20px',

                '3xl': '24px',

                'full': '9999px',

                'button': '4px'

            }

        }

    }

}

window.onload = function() {

    const loginTab = document.getElementById('loginTab');

    const registerTab = document.getElementById('registerTab');

    const loginContent = document.getElementById('loginContent');

    const registerContent = document.getElementById('registerContent');

    // 添加空值检查

    if (!loginTab || !registerTab || !loginContent || !registerContent) {

        console.error('One or more elements not found');

        return;

    }

    loginTab.addEventListener('click', () => {

        loginTab.classList.add('text-primary', 'border-primary');

        loginTab.classList.remove('text-gray-400', 'border-gray-200');

        registerTab.classList.add('text-gray-400', 'border-gray-200');

        registerTab.classList.remove('text-primary', 'border-primary');

        loginContent.classList.add('active');

        registerContent.classList.remove('active');

    });

    registerTab.addEventListener('click', () => {

        registerTab.classList.add('text-primary', 'border-primary');

        registerTab.classList.remove('text-gray-400', 'border-gray-200');

        loginTab.classList.add('text-gray-400', 'border-gray-200');

        loginTab.classList.remove('text-primary', 'border-primary');

        registerContent.classList.add('active');

        loginContent.classList.remove('active');

    });

}

function login() {

    const username = document.getElementById("username").value;

    const password = document.getElementById("password").value;

    if (username && password) {

        fetchSQLWorkflow(`查询用户名为${username}且密码为${password}的用户信息`, username)

            .then(data => {

                res = data.result

                console.log(res)

                // 没有查到相关的用户

                if (res.length == 0) {

                    showFloatingAlert("用户名或密码错误")

                } else {

                    localStorage.setItem("user", JSON.stringify(res[0]))

                    location.href = "index.html"

                }

            })

    } else {

        showFloatingAlert("请确保表单填写完整", 'warning')

    }

}

function register() {

    const regUsername = document.getElementById("regUsername").value;

    const regPassword = document.getElementById("regPassword").value;

    const secPassword = document.getElementById("secPassword").value;

    if (regUsername && regPassword & secPassword) {

        if (regPassword == secPassword) {

            fetchSQLWorkflow(`创建用户名为'${regUsername}' 密码为'${regPassword}'的用户`, regUsername)

                .then(data => {

                    res = data.result[0].result

                    console.log(res)

                    if (res == 1) {

                        showFloatingAlert("注册成功", 'success')

                    } else if (res.includes("UNIQUE constraint failed")) {

                        showFloatingAlert("注册失败,用户名已存在", 'error')

                    } else {

                        showFloatingAlert("服务器异常" + res, 'error')

                    }

                })

        } else {

            showFloatingAlert("两次输入的密码不一致", 'warning')

        }

    } else {

        showFloatingAlert("请确保表单填写完整", 'warning')

    }

}

# 第五章 系统测试

## 5.1 系统测试的目的

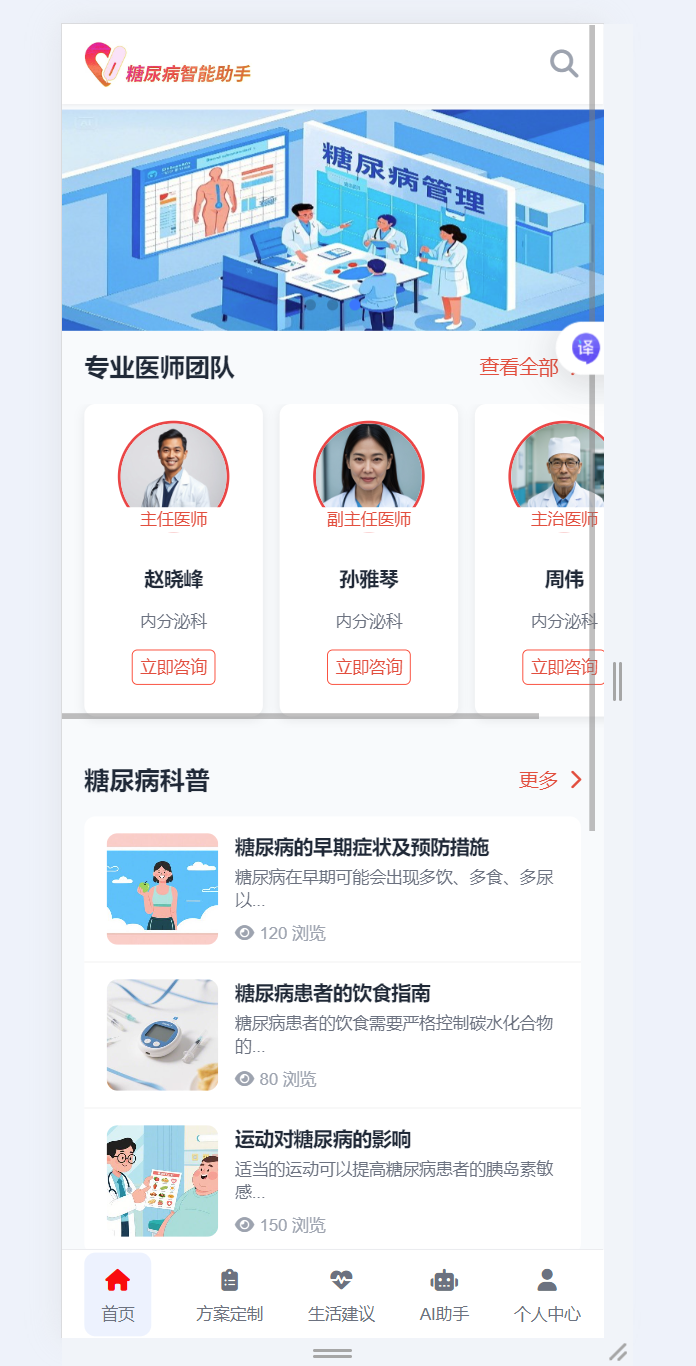
对于开发一个系统，测试是保证系统可以正常上线运行的重要部分。测试可以对于之前在需求分析，系统设计，功能编码等步骤中，没有发现或者考虑到的错误和缺陷进行标注，并进行下一步的修改。测试是对开发出来的源程序进行试运行，通过测试，发现程序问题，并及时的反馈给开发人员。这样可以有效地提高软件的开发质量，保证系统的进度。所以说，系统测试在软件开发中有着重要地位，在开发时，编写出一个功能模块后就需要对它进行必要的测试。

## 5.2 系统测试和功能模块测试

### 5.2.1 登录模块测试



### 5.2.2 首页测试



### 5.2.3 AI助手测试



### 5.2.4个人信息测试



### 5.3 测试总结

通过分析测试结果，糖尿病预治智能助手基本实现了需求所要求的功能，系统功能完整，操作逻辑严密，无重大缺陷，在进行部分细节修改后，即可进行系统试运行