

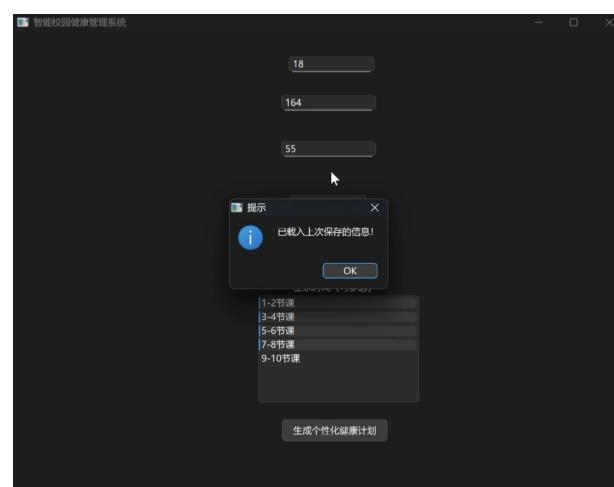
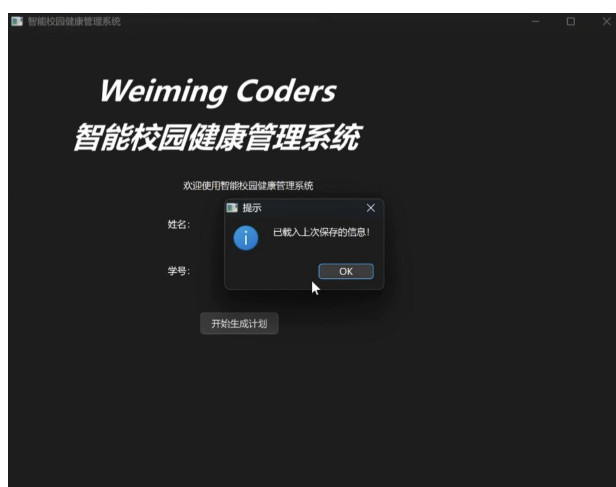
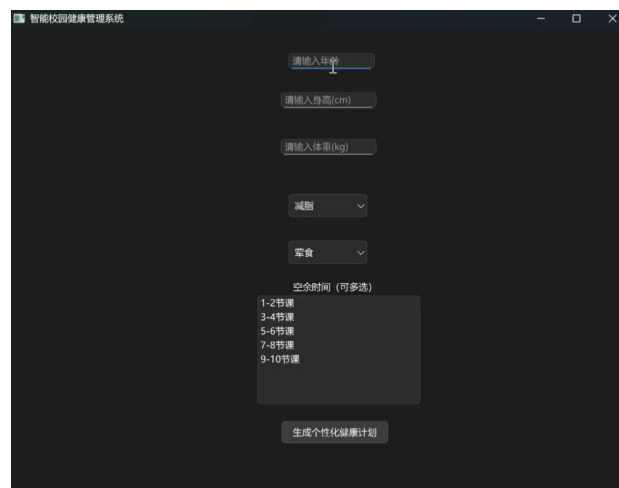
智能健康管理系统

程设大作业报告

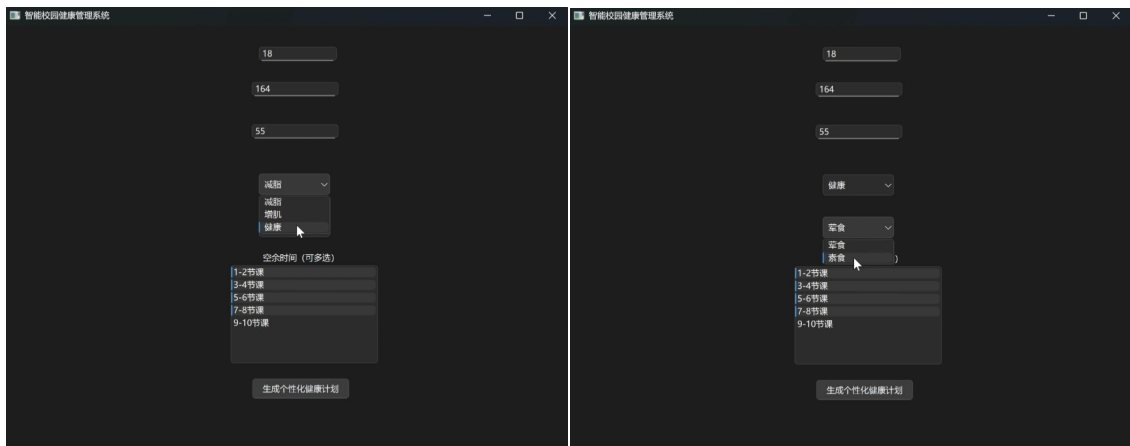
28 组 Weiming Coders

一、程序功能介绍

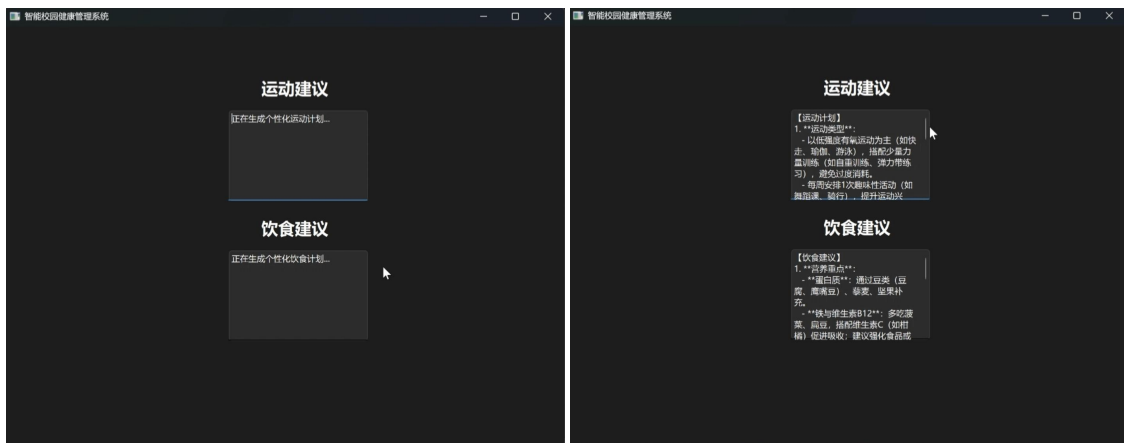
- 个人登录信息存储与读取。



- 通过用户的身高体重，可选择增肌/减脂/健康三种运动模式，以及荤食/素食两种饮食偏好，系统接入了 Deepseek V1，会根据不同选项以及身高体重来智能分配运动与饮食内容。



- 用户可选择自己工作日的空余时间（如第 3 节课-第 4 节课），Deepseek 会根据空余时间多少来安排运动量。



二、项目各模块与类设计细节

（一）项目概述

校园健康管理系统是一个基于 Qt 框架开发的桌面应用，旨在为学生提供个性化的健康管理方案。系统通过收集用户的基本信息、身体指标和健康目标，调用 DeepSeek V1 服务生成运动计划和饮食建议，并在本地提供备选方案。系统具有用户信息管理、BMI 计算、运动时间计算、AI 健康计划生成等核心功能。

（二）系统架构设计

1. 整体架构

系统采用分层架构设计，主要分为以下几个层次：

- 表示层：负责用户界面展示和交互，由 MainWindow 类实现
- 业务逻辑层：处理核心业务逻辑，包括用户信息管理、健康计算等
- 数据访问层：负责用户数据的存储和读取
- 服务层：封装 AI 服务调用，与 DeepSeek API 交互

2. 模块划分

系统主要由以下模块组成：

- 用户界面模块：MainWindow 类及其相关 UI 组件
- AI 服务模块：AIService 类，负责与 DeepSeek API 交互
- 用户信息模块：UserProfile 结构体，存储用户基本信息
- 数据存储模块：接口函数，处理用户数据的 JSON 存储和读取
- 健康计算模块：提供 BMI 计算、运动时间计算等功能

(三) 核心类设计与功能

1. AIService 类

类定义

```
cpp
class AIService : public QObject {
    Q_OBJECTpublic:
    explicit AIService(QObject *parent = nullptr);
    void generatePlan(float bmi, int exerciseTime, const QString &userInfo);

signals:
    void planGenerated(const QStringList &plan);
    void errorOccurred(const QString &msg);

private slots:
    void onReplyFinished(QNetworkReply *reply);
```

```
private:

    QNetworkAccessManager *manager;

    const QString apiUrl;

    const QString apiKey;};
```

功能描述

- **AI 服务封装**: 负责与 DeepSeek API 通信, 发送健康计划生成请求
- **网络请求处理**: 使用 QNetworkAccessManager 处理 HTTP 请求和响应
- **JSON 数据处理**: 构建符合 DeepSeek API 格式的请求体, 并解析响应
- **异步通信**: 通过信号槽机制实现异步请求, 不阻塞 UI 线程
- **错误处理**: 捕获网络错误和 API 错误, 并发送错误信号

关键方法

- generatePlan: 构建请求参数, 发送 API 请求
- onReplyFinished: 处理 API 响应, 解析结果并发送信号

2. MainWindow 类

类定义

```
cpp

class MainWindow : public QMainWindow{

    Q_OBJECT

public:

    explicit MainWindow(QWidget *parent = nullptr);

    ~MainWindow();

private slots:

    void on_generateButton_clicked();

    void on_submitButton_clicked();

    void onWorkoutPlanGenerated(const QStringList &plan);

    void onErrorOccurred(const QString &errorMsg);

private:

    Ui::MainWindow *ui;

    UserProfile currentUser;

    AIService *aiService;
```

```
int calculateTotalExerciseTime(const QStringList &freeTimes);
```

```
QStringList generateWorkoutPlan(const UserProfile &user);
```

```
QStringList generateMealPlan(const UserProfile &user);};
```

功能描述

- **用户界面管理**: 负责系统主界面的展示和交互
- **业务逻辑协调**: 协调各模块工作, 处理用户输入和系统响应
- **AI 服务集成**: 连接 AIService 的信号槽, 处理 AI 返回结果
- **本地备选方案**: 当 AI 服务出错时, 提供本地生成的健康计划
- **用户数据管理**: 收集、验证和保存用户输入的健康信息

关键方法

- on_submitButton_clicked: 处理用户提交的健康信息
- onWorkoutPlanGenerated: 处理 AI 返回的健康计划
- onErrorOccurred: 处理 AI 服务错误, 切换到本地方案

3. UserProfile 结构体

结构体定义

cpp

```
struct UserProfile{  
  
    QString name;  
  
    QString studentId;  
  
    int age;  
  
    float height;  
  
    float weight;  
  
    QString goal;  
  
    QStringList freeTimes;  
  
    QString dietType;  
  
  
    float calculateBMI() const;  
  
    int calculateExerciseTime() const;};
```

功能描述

- **用户信息存储**: 存储用户的基本信息、身体指标和健康目标
- **健康指标计算**: 提供 BMI 计算和运动时间计算功能
- **数据持久化**: 支持将用户信息序列化为 JSON 格式存储

关键方法

- calculateBMI: 计算用户身体质量指数
- calculateExerciseTime: 根据用户空闲时间计算总运动时间

(四) 模块交互流程

1. AI 健康计划生成流程

1. 用户在界面输入健康信息并提交
2. MainWindow 收集用户信息，计算 BMI 和运动时间
3. MainWindow 调用 AIService 的 generatePlan 方法
4. AIService 构建 API 请求，发送到 DeepSeek 服务器
5. DeepSeek 服务器处理请求，返回健康计划
6. AIService 解析响应，发送 planGenerated 信号
7. MainWindow 接收信号，解析并显示健康计划

具体 AI prompt:

```
void AIService::generatePlan(float bmi, int exerciseTime, const QString &userInfo) {  
    // 构建提示词 - 明确要求分两部分输出  
    QString prompt = QString(  
        "用户BMI: %1, 每周锻炼时间: %2分钟, 额外信息: %3\n\n"  
        "请生成个性化健康计划, 包含以下两部分内容: \n"  
        "1. 【运动计划】根据用户BMI和空闲时间, 设计合理的运动方案 (包含运动类型、频率、时长) \n"  
        "2. 【饮食建议】根据用户目标和饮食偏好, 提供营养建议 (包含三餐示例) \n\n"  
        "输出格式要求: \n"  
        "首先明确标注 '【运动计划】', 然后是运动内容\n"  
        "然后明确标注 '【饮食建议】', 然后是饮食内容\n"  
        "每部分内容请分点说明, 使用中文。")  
        .arg(bmi).arg(exerciseTime).arg(userInfo);  
}
```

2. 错误处理流程

1. AIService 检测到网络错误或 API 错误
2. AIService 发送 errorOccurred 信号
3. MainWindow 接收信号，显示错误信息
4. MainWindow 调用本地健康计划生成方法
5. 显示本地生成的健康计划作为备选方案

(五) 关键技术与实现细节

1. AI 服务集成

系统通过 QNetworkAccessManager 实现与 DeepSeek API 的通信, 使用 JSON 格式进行数据交换。关键实现点包括:

- **请求头设置:** 正确设置 Authorization 和 Content-Type 头
- **请求体构建:** 按照 DeepSeek API 规范构建 JSON 请求体
- **响应解析:** 从 JSON 响应中提取健康计划内容

2. 本地健康计算

系统提供本地健康计算功能, 包括:

- **BMI 计算:** 根据身高和体重计算身体质量指数
- **运动时间计算:** 根据用户选择的空闲时间计算总运动时间
- **本地健康计划生成:** 当 AI 服务不可用时, 根据用户信息生成备选计划

3. 用户数据管理

系统使用 JSON 格式存储用户数据, 实现了:

- **数据序列化:** 将 UserProfile 对象转换为 JSON 格式
- **数据持久化:** 将用户数据保存到本地文件
- **数据加载:** 从本地文件读取用户数据

(六) 扩展性设计

1. 服务提供者扩展

系统设计支持扩展其他 AI 服务提供者, 只需:

- 实现新的服务类, 继承自统一的接口
- 在 MainWindow 中配置使用新的服务类
- 保持接口一致性, 不影响上层逻辑

2. 健康计划算法扩展

系统支持扩展健康计划生成算法:

- 实现新的计划生成类，实现统一的生成接口
- 在 MainWindow 中注册新的计划生成算法
- 支持在配置中选择使用不同的生成算法

3. 数据存储扩展

系统支持扩展数据存储方式:

- 实现新的数据存储类，实现统一的数据接口
- 在接口函数中添加对新存储方式的支持
- 保持数据访问接口一致性，不影响上层逻辑

(六) 总结

校园健康管理系统通过合理的架构设计和模块划分，实现了用户信息管理、健康指标计算、AI 健康计划生成等核心功能。系统采用 Qt 框架实现了良好的用户界面，使用 DeepSeek API 提供智能健康计划生成服务，并在本地提供备选方案，保证了系统的可用性。

系统的设计注重可扩展性，通过接口设计和模块化实现，支持未来对 AI 服务、健康算法和数据存储的扩展。错误处理机制确保了系统在外部服务不可用时仍能提供基本功能，提升了用户体验。

该系统适合作为校园健康管理的辅助工具，帮助学生制定科学的运动和饮食计划，促进健康生活方式的养成。

三、小组成员分工情况

吴彦澍 2400013076: 框架搭建，UI 界面设计，用户信息存储与读取，作业报告撰写，演示视频解说。

唐天行 2400017794: 运动、饮食推荐算法，空闲时间推荐运动算法，程序实际运行录屏。

黄喆 2400013090: 界面设计修改美化，推荐算法优化，AI 接入，deepseek 模型生成个性化推荐，报告撰写

四、项目总结与反思

我们小组一开始值得肯定的是很迅速的定好了要做的方向与程序的具体内容，目标分工明确。但是在具体构建的过程中，由于组员运用 Mac 和 Windows

两个不同的操作系统，导致了很长一段时间的不兼容，并且加上对于 github 的不熟悉，导致很多时间都被浪费了，没有有效的利用。不过经过团队协作和协商，共同解决了协作问题，并且在完成的时候还加入了自己的创新点。在这次的项目里，我们学到了要一开始就尽快搭建好可以共同协作的环境，以便一起开发和推进工作。