

# 体重管理助手 - 完整设计文档

文档版本: 1.0

创建日期: 2025-02-07

状态: 初版设计, 持续迭代

## 一、产品定位

核心定位：以AI对话为核心的个性化体重管理伙伴

差异化卖点：

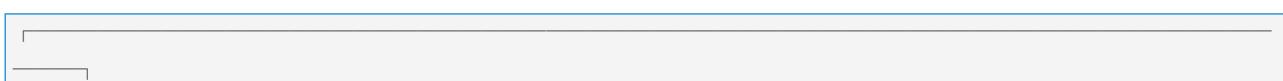
- \*\*深度个性化\*\*：记住用户所有习惯、偏好、历史，像老朋友一样懂你
- \*\*对话即操作\*\*：所有功能在对话框内完成，无需跳转页面
- \*\*智能洞察\*\*：AI发现隐藏模式（睡眠-体重关联、情绪性进食等）
- \*\*长期陪伴\*\*：不是工具，是养成习惯的伙伴

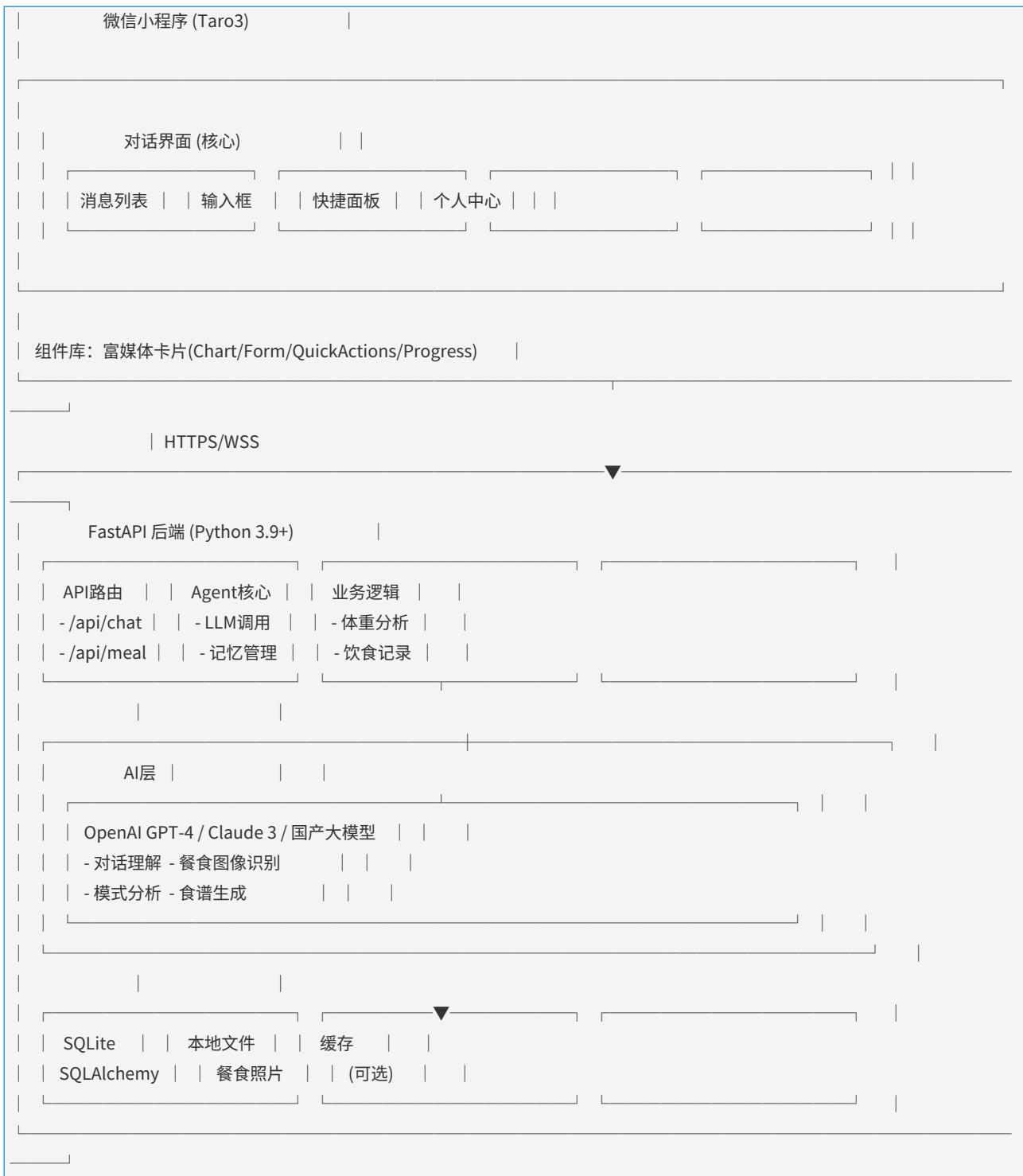
竞品对标（薄荷健康）：

功能	薄荷健康	本产品
记录方式	手动输入为主	AI对话识别
个性化	通用建议	1v1长期记忆
交互体验	功能型App	陪伴型Agent
数据分析	基础统计	AI模式洞察

## 二、技术架构

### 2.1 系统架构图





## 2.2 部署方案

开发环境：Python直接运行 + SQLite本地文件

生产环境（未来）：云服务器 + PostgreSQL + Redis + OSS存储

### 三、数据库设计（ER图核心表）

#### 3.1 用户与认证

```
users:  
- id: PK  
- openid: string (微信唯一标识) # 微信用户唯一标识  
- nickname: string # 用户昵称  
- avatar_url: string # 头像URL  
- phone: string (可选) # 手机号  
- created_at: datetime  
- last_login: datetime  
- is_vip: boolean (商业化预留) # 是否VIP会员  
- vip_expire: date # VIP过期时间
```

#### 3.2 核心记录表

```
weight_records:  
- id: PK  
- user_id: FK  
- weight: float (kg) # 体重 (kg)  
- body_fat: float (%) (可选) # 体脂率 (%)  
- record_date: date # 记录日期  
- record_time: time # 记录时间  
- note: text # 备注  
- created_at: datetime  
  
meal_records:  
- id: PK  
- user_id: FK  
- meal_type: enum (breakfast/lunch/dinner/snack) # 餐食类型 (早餐/午餐/晚餐/加餐)  
- record_time: datetime # 记录时间  
- photo_url: string # 餐食照片URL  
- food_items: JSON # 食物明细 (JSON格式)  
- total_calories: int # 总热量 (千卡)  
- user_confirmed: boolean # 用户是否确认识别结果  
- ai_confidence: float # AI识别置信度 (0-1)  
- created_at: datetime  
  
exercise_records:  
- id: PK  
- user_id: FK  
- exercise_type: string # 运动类型  
- duration_minutes: int # 运动时长 (分钟)  
- calories_burned: int # 消耗热量 (千卡)  
- intensity: enum (low/medium/high) # 运动强度 (低/中/高)  
- record_time: datetime # 记录时间
```

```
- photo_evidence: string (可选) # 运动凭证照片
```

```
water_records:
```

```
- id: PK  
- user_id: FK  
- amount_ml: int # 饮水量 (毫升)  
- record_time: datetime # 记录时间
```

```
sleep_records:
```

```
- id: PK  
... (代码已截断)
```

### 3.3 用户画像与长期记忆

```
user_profiles:
```

```
- id: PK  
- user_id: FK  
- age: int # 年龄  
- gender: enum # 性别  
- height: float (cm) # 身高 (cm)  
- bmr: int (基础代谢) # 基础代谢率 (BMR)  
- diet_preferences: JSON # 饮食偏好 (JSON)  
- exercise_habits: JSON # 运动习惯 (JSON)  
- weight_history: text # 减重历史记录  
- body_signals: JSON # 身体信号 (JSON: 疲劳/失眠等)  
- motivation_type: enum # 动力类型 (数据驱动/情感支持/目标导向)  
- weak_points: JSON # 薄弱环节 (JSON)  
- memory_summary: text # AI记忆摘要 (自然语言描述)  
- updated_at: datetime
```

```
goals:
```

```
- id: PK  
- user_id: FK  
- target_weight: float # 目标体重 (kg)  
- target_date: date # 目标达成日期  
- weekly_plan: float # 每周减重计划 (kg)  
- daily_calorie_target: int # 每日热量目标 (千卡)  
- meal_distribution: JSON # 三餐热量分配比例 (JSON)  
- status: enum (active/completed/abandoned) # 状态 (进行中/已完成/已失败)  
- created_at: datetime
```

### 3.4 Agent配置与对话

```
agent_configs:
```

```
- id: PK  
- user_id: FK  
- agent_name: string # Agent名称  
- personality_type: enum (professional/warm/energetic) # 性格类型 (专业/温暖/活力)
```

```

- personality_prompt: text # 性格Prompt描述
- created_at: datetime

chat_history:
- id: PK
- user_id: FK
- role: enum (user/assistant/system) # 消息角色 (user/assistant/system)
- content: text # 消息内容
- msg_type: enum (text/card/image/form/quick_actions) # 消息类型 (text/card/image/form)
- metadata: JSON # 元数据 (JSON)
- created_at: datetime

conversation_summaries:
- id: PK
- user_id: FK
- week_start: date # 周开始日期
- summary_text: text # 对话摘要文本
- key_facts: JSON # 关键事实 (JSON)
- created_at: datetime

```

### 3.5 提醒与功能配置

```

reminder_settings:
- id: PK
- user_id: FK
- reminder_type: enum # 提醒类型
- enabled: boolean # 是否启用
- reminder_time: time # 提醒时间
- interval_minutes: int # 间隔分钟数 (饮水用)
- weekdays_only: boolean # 仅工作日提醒
- last_triggered: datetime # 上次触发时间
- skip_count: int # 连续忽略次数

do_not_disturb:
- user_id: PK
- enabled: boolean # 是否启用
- start_time: time # 开始时间
- end_time: time # 结束时间
- except_emergencies: boolean # 紧急提醒例外

```

### 3.6 食物数据库

```

food_items:
- id: PK
- name: string # 食物名称
- aliases: JSON # 别名列表 (JSON)
- category: enum # 食物分类
- calories_per_100g: float # 每100克热量 (千卡)

```

```
- protein/fat/carbs: float  
- common_portions: JSON # 常见分量 (JSON)  
- is_user_created: boolean # 是否用户自定义
```

```
user_foods:  
- id: PK  
- user_id: FK  
- food_name: string  
- calories: int  
- created_at: datetime
```

### 3.7 激励体系

```
achievements:  
- id: PK  
- code: string # 成就代码 (唯一标识)  
- name: string # 食物名称  
- description: text # 成就描述  
- icon: string # 图标URL或代码  
- condition_type: enum # 达成条件类型  
- condition_value: int # 达成条件值
```

```
user_achievements:  
- id: PK  
- user_id: FK  
- achievement_id: FK  
- unlocked_at: datetime # 解锁时间
```

```
challenges:  
- id: PK  
- title: string # 挑战标题  
- description: text # 成就描述  
- type: enum # 挑战类型 (日/周/自定义)  
- target: JSON # 挑战目标 (JSON)  
- reward_points: int # 奖励积分
```

```
user_challenges:  
- id: PK  
- user_id: FK  
- challenge_id: FK  
- progress: JSON # 进度 (JSON)  
- status: enum # 状态 (进行中/已完成/已失败)  
- started_at: datetime # 开始时间  
- completed_at: datetime # 完成时间
```

```
user_points:  
- user_id: PK  
- total_points: int # 累计积分  
- current_points: int # 当前可用积分
```

### 3.8 报告与商业化

```
weekly_reports:  
- id: PK  
- user_id: FK  
- week_start: date # 周开始日期  
- summary_text: text # 对话摘要文本  
- weight_change: float # 体重变化 (kg)  
- avg_weight: float # 平均体重 (kg)  
- avg_calories_in/out: int  
- exercise_days: int # 运动天数  
- highlights/improvements: JSON  
- created_at: datetime  
  
nutritionists:  
- id: PK  
- name: string # 食物名称  
- avatar: string  
- title: string # 挑战标题  
- specialties: JSON # 专长领域 (JSON)  
- intro: text # 个人简介  
- is_active: boolean # 是否在职  
  
consultations:  
- id: PK  
- user_id: FK  
- nutritionist_id: FK  
- status: enum # 状态 (进行中/已完成/已失败)  
- question: text # 咨询问题  
- answer: text # 营养师回答  
- created_at: datetime  
- answered_at: datetime # 回答时间
```

---

## 四、API设计

### 4.1 核心API列表

```
# 对话  
POST /api/chat/send      # 发送消息  
GET /api/chat/history    # 获取最近对话  
POST /api/chat/clear     # 清空对话  
  
# 体重  
POST /api/weight/record   # 记录体重
```

```
GET /api/weight/trend      # 获取趋势数据
GET /api/weight/stats      # 统计数据

# 餐食
POST /api/meal/record      # 记录餐食
POST /api/meal/analyze     # AI分析餐食图片
POST /api/meal/confirm     # 确认/修正AI识别
GET /api/meal/today        # 今日餐食记录

# 运动/饮水/睡眠
POST /api/exercise/record
POST /api/water/record
POST /api/sleep/record

# 目标
POST /api/goal            # 创建/更新目标
GET /api/goal/current     # 当前目标
GET /api/goal/progress    # 进度追踪

# 用户
GET /api/user/profile      # 获取用户画像
POST /api/user/profile     # 更新用户画像
GET /api/user/achievements # 成就列表
GET /api/user/points       # 积分

# 提醒
GET /api/reminders         # 提醒设置列表
POST /api/reminders/{type}  # 更新提醒设置
POST /api/reminders/dnd    # 免打扰设置

# 报告
GET /api/report/weekly     # 周报告
GET /api/report/export      # 导出数据(PDF/Excel)
... (代码已截断)
```

## 4.2 WebSocket实时推送

```
ws://host/ws/chat/{user_id}  # 对话实时推送
ws://host/ws/notify/{user_id} # 提醒推送
```

## 五、核心功能流程

### 5.1 首次使用流程

用户进入小程序  
↓  
微信授权登录 → 创建用户记录  
↓  
Agent打招呼 → 极简引导  
|—— Step 1: 当前体重? 目标体重? 达成时间?  
|—— Step 2: 选择Agent人设(专业/温暖/活力) + 起名字  
↓  
完成初始化 → 进入主对话界面  
↓  
Agent主动收集(后续):  
- Day 2: "你平时容易饿吗?"  
- Day 3: "之前试过什么减肥方法?"  
- Day 7: 首次周报告 + 画像确认

## 5.2 餐食打卡流程

[提醒触发]  
Agent推送: "午餐时间到啦! 今天吃什么?"  
↓  
用户上传照片 / 文字描述 / 点击快捷按钮  
↓  
[如果是照片]  
后端接收 → 调用AI视觉分析  
|—— 识别食物种类  
|—— 估算分量  
|—— 查询食物数据库计算热量  
↓  
Agent回复卡片:  
[餐食照片]  
识别结果: 番茄鸡蛋面  
估算热量: 450千卡  
今日午餐建议: 500千卡  
[确认] [调整] [不对]  
↓  
用户操作:  
|—— [确认] → 记录完成 → 展示今日热量进度  
|—— [调整] → 滑动条调整 → 重新计算 → 确认  
|—— [不对] → 文字重新描述 → AI重新解析 → 确认  
↓  
[后台] 提取关键信息更新UserProfile

## 5.3 每周智能报告示例

## 本周体重报告 (1月15日-1月21日)

本周平均: 65.2kg (上周 65.8kg)

变化: -0.6kg ↓

### 本周亮点

- 5天按时记录体重，很棒！
- 周三饮食控制很好，晚餐仅380千卡
- 运动后睡眠质量提升(7.2h → 7.8h)

### ⚠ 需要关注

- 周末热量摄入偏高(平均超支320千卡)
- 周六未记录体重

### AI洞察

根据你过去30天的数据：

- 工作压力大时(周一/周二)容易晚餐过量
- 早睡(23点前)的第二天体重下降更明显
- 游泳后3天内体重下降速度比跑步快20%

下周目标: 64.8kg (-0.4kg)

建议调整:

1. 周末提前规划低热量食谱
2. 工作压力日准备健康零食替代
3. 保持23点前入睡的习惯

[查看详细图表] [制定下周计划] [咨询营养师]

## 六、Agent设计

### 6.1 系统Prompt模板

SYSTEM\_PROMPT = """

你是{agent\_name}，用户的专属体重管理伙伴。

【当前时间】 {current\_time}

【用户基础信息】

- 昵称: {userNickname}
- 性别/年龄/身高: {gender}/{age}/{height}cm
- 当前体重: {currentWeight}kg / 目标: {targetWeight}kg

【用户画像】 (长期记忆)

{userProfileSummary}

【近期情况】 (最近7天)

- 体重趋势: {weight\_trend}
- 平均摄入: {avg\_calories\_in}千卡/日
- 运动频率: {exercise\_frequency}
- 睡眠时长: {avg\_sleep}小时
- 本周亮点: {highlights}
- 本周挑战: {challenges}

#### 【沟通风格】

你是{personality\_type}型助手:  
{personality\_description}

#### 【回复原则】

1. 回复要个性化，参考用户画像和历史数据
2. 主动关心用户的身心信号和情绪状态
3. 给出具体可操作的建议，避免空泛
4. 适时鼓励，但不要过度
5. 发现异常数据(如连续增长)要温和提醒

#### 【消息类型选择】

根据内容选择合适的形式:

- text: 纯文本对话
- card: 数据卡片(体重/热量等)
- chart: 趋势图表
- form: 需要用户填写的表单
- quick\_actions: 快捷操作按钮
- image: 图片(食谱示例等)

... (代码已截断)

## 6.2 人设定义

### 专业型：

你是专业的营养师和健身教练。语气客观、科学、数据驱动。

- 使用专业术语但解释清楚
- 引用科学依据支撑建议
- 关注营养均衡和身体健康
- 给出精确的数据和计算
- 例句: "根据你的BMI 24.3, 建议每日摄入控制在1500千卡..."

### 温暖型：

你是用户的好朋友，给予情感支持。语气亲切、鼓励、有耐心。

- 多用表情符号和温暖词汇
- 关注用户的情绪和心理状态
- 庆祝每一个小进步
- 在用户低落时给予安慰
- 例句: "今天虽然体重没变，但坚持记录就很棒啦！"

### 活力型：

你是充满活力的健身教练。语气积极、有感染力、略带push。

- 使用感叹号和鼓励性词汇
- 强调行动和挑战
- 设定小目标激发动力
- 用挑战和竞赛的方式激励
- 例句: "冲鸭! 这周再减0.5kg就达标了, 我们一起加油!"

## 七、运营后台设计

### 7.1 技术栈

独立Web应用 (React/Vue + FastAPI)

### 7.2 功能模块

#### 【用户管理】

- 用户列表/搜索/详情
- 体重/饮食/运动数据查看
- 对话历史查看 (客服支持)

#### 【内容管理】

- 食物数据库管理
- 食谱库管理
- 成就系统配置
- 挑战任务管理

#### 【营养师工作台】

- 待回答咨询列表
- 用户健康档案查看
- 回答咨询界面
- 咨询历史

#### 【数据分析】

- 用户活跃度统计
- 功能使用统计
- 减重成功率分析
- AI识别准确率监控

#### 【系统配置】

- 提醒模板配置
- AI模型参数调整
- 会员等级配置
- 积分规则配置

### 7.3 页面路由

```
/dashboard      # 数据大盘  
/users          # 用户列表  
/users/:id      # 用户详情  
/foods          # 食物库管理  
/achievements   # 成就管理  
/challenges     # 挑战管理  
/nutritionists  # 营养师管理  
/consultations  # 咨询管理  
/reports         # 数据报表  
/settings        # 系统设置
```

## 八、商业化规划

### 8.1 会员等级体系

功能	免费版	基础会员(¥19/月)	高级会员(¥49/月)
体重记录	✓	✓	✓
基础对话	50条/月	无限	无限
餐食AI识别	3次/日	10次/日	无限
智能分析	周报	周报+月报	周报+月报+AI洞察
食谱推荐	基础	个性化	定制化
营养师咨询	-	1次/月	无限
数据导出	-	Excel	Excel+PDF
广告	有	无	无

### 8.2 积分体系

获取：

- 每日打卡： +5分
- 连续7天： +20分
- 达成目标： +100分
- 完成挑战： +30分
- 邀请好友： +50分

消耗：

- 解锁专属食谱：-50分
- 延长AI对话：-10分

### 8.3 其他变现

- \*\*营养师咨询\*\*：按次付费 / 包月
  - \*\*企业版\*\*：公司健康管理服务
  - \*\*数据服务\*\*：匿名化健康数据分析（B端）
- 

## 九、开发路线图

Phase 1: MVP (4-5周)

目标：核心闭环可用

功能：

- [] 用户系统 + 微信登录
- [] Agent基础对话（文本+卡片）
- [] 体重记录 + 7天趋势图
- [] 极简首次引导（2步）
- [] 基础人设选择（3种）

技术：SQLite本地 + OpenAI API

Phase 2: 核心功能 (3-4周)

目标：完整健康管理闭环

功能：

- [] 餐食AI识别 + 纠错
- [] 食物数据库（3000+数据）
- [] 运动/饮水/睡眠记录
- [] 提醒系统（全功能）
- [] 每周智能报告
- [] UserProfile + 长时记忆
- [] 免打扰模式

## Phase 3: 智能化 + 激励 (3周)

目标：差异化体验

功能：

- [ ] AI模式洞察（智能版分析）
- [ ] 个性化食谱生成
- [ ] 成就系统 + 积分体系
- [ ] 挑战任务
- [ ] 数据导出

## Phase 4: 社交 + 商业化 (4周)

目标：可商业发布

功能：

- [ ] 好友系统 + 排行榜
- [ ] 会员体系 + 支付模块
- [ ] 营养师服务
- [ ] 运营后台

技术：PostgreSQL迁移 + 生产部署

## Phase 5: 硬件 + 扩展 (后续)

- [ ] 体脂秤API接入
- [ ] 运动手环同步
- [ ] 企业版功能

## 十、技术风险与应对

风险	影响	应对策略
AI API成本高	费用超支	限制免费版额度；使用缓存；优化Prompt
AI识别不准确	用户体验差	完善纠错机制；用户反馈闭环；持续优化

小程序审核限制	功能受限	提前了解微信规则；敏感功能加免责声明
用户留存低	产品失败	强化提醒机制；游戏化激励；个性化陪伴
数据安全	隐私问题	数据加密；本地化优先；合规声明

## 十一、项目目录结构

```

weight-management/
├── README.md
├── requirements.txt
├── main.py      # 后端入口
├── config/
│   ├── settings.py    # 配置管理
│   └── logging_config.py  # 日志配置
├── api/
│   ├── routes/      # API路由
│   │   ├── chat.py
│   │   ├── weight.py
│   │   ├── meal.py
│   │   └── user.py
│   └── dependencies.py
├── core/
│   ├── agent.py     # Agent核心
│   ├── memory.py    # 记忆管理
│   ├── analyzer.py   # 数据分析
│   └── scheduler.py  # 定时任务
├── models/
│   └── database.py   # SQLAlchemy模型
├── services/
│   ├── ai_service.py  # AI调用封装
│   ├── reminder_service.py
│   └── report_service.py
├── utils/
│   └── exceptions.py  # 异常处理
├── data/
│   ├── food_database.json # 食物数据库
│   └── init_sql/        # 初始化SQL
├── admin/          # 运营后台（独立）
│   ├── package.json
│   └── src/
└── miniapp/        # 微信小程序
    ├── package.json
    └── src/
└── docs/
    └── plans/

```

```
|   └── 2025-02-07-weight-management-design.md  
└── tests/      # 测试
```

---

## 十二、后续优化计划

- 开发中发现问题及时调整
  - 每阶段开始前回顾并细化设计
  - 每阶段结束后更新技术文档
  - 根据用户反馈持续迭代功能
- 

文档结束

\*本设计文档为初版，后续将根据实际开发情况持续更新。\*