

饮食助手系统-需求分析及可行性分析

一、概述

1.1 课题背景

在当今世界，随着生活水平的提高以及食品的种类的不断丰富，人们对于“吃”有了更高的要求。对于不同的人其饮食需求往往不同：有的人希望更改自己的饮食计划来减肥，有的人寻求着更健康的饮食方式，也有的人希望制定一个适合自己的增肌或减脂的饮食方案。尽管大多数人都有这样的需求，但是却很少有一个非常便捷的平台来帮助用户制定自己的饮食计划。

针对这一现状，有必要设计一个饮食助手系统，用户可以使用这个系统来查找各种食物的热量及各种营养素含量，且该系统还可以帮助用户进行定制化的饮食计划分析，并给出推荐的饮食方案。

1.2 编写目的

该文档将从项目的基本结构及功能进行描述，试图给出项目的大致轮廓。再深入对产品的功能，特点和实现方法进行详细的介绍，同时对用户需求进行分析。便于用户进行理解和交流，反映出用户问题的结构，可以作为开发工作的基础和依据以及确认测试和验收的依据。该文档的预期读者为用户与开发人员。在阅读本文档时，首先要了解产品的功能概貌，然后可以根据自身的需要对每一功能进行适当的了解。

1.3 适用人群

本产品是针对对饮食健康方面有需求的用户而开发，在需求上，充分考虑了具体用户的实际情况。在功能上，主要用于为用户提供食物热量查询、饮食计划推荐等服务。本产品适用于各年龄段人群，并且对于不同年龄段人群提供的功能侧重点不同。

1.4 产品功能

作为官方，将为用户提供以下功能：

- 用户可以通过本应用查找各类食材、菜肴的热量及营养素含量
- 用户可以通过本应用查找自己每日所需摄入的各类营养素的量
- 用户可以通过本应用记录自己的饮食计划，并且系统会为用户分析其饮食计划的合理性
- 系统可以根据用户的需求及信息（包括身体信息、所在地信息、希望在健康饮食上的花费金额等）为用户自动推荐饮食计划

作为用户，可以作为贡献者为其他用户提供以下数据：

- 用户可以自己上传各类菜肴所用的食材以及其做法，并给出整体的热量及所包含的各类营养素含量
- 用户可以将自己制定的饮食计划上传，并分享给社区

1.5 产品特点

本产品并未选择在当下较火的移动端上开发。虽然在移动端上更加容易推广，且大多数用户更习惯于移动端的应用，移动端的用户基数也更大。但当下移动端已经有较为成熟的饮食方面的 app，而对于 PC 端上此类应用非常少，且在 PC 端上开发可以添加更多移动端无法实现的功能，提高用户制定饮食计划的效率。此外，本产品会根据用户愿意在饮食上消费的金额以及其所在地综合来分析并给出推荐的饮食计划，这是其他同类应用很少具备的功能。

1.6 一般约束

约束名称	说明
开发期限	2019/10/10-2020/5/1
经费开支	租赁云服务器以及云数据库需要经费
高级语言	JavaScript, HTML 和 CSS
可靠性需求	该应用程序为用户提供定制的饮食计划，如果在分析用户数据时出现错误，给出错误的饮食计划推荐会对用户产生误导，影响用户健康及体验。因此对数据处理的准确性有非常高的要求。

二、需求分析

2.1 用户需求

- 食物数据库
可以查询到生活中各种常见食物的准确热量及各种营养素含量
- 健康评估
可以根据用户身体情况计算其每日应摄取的各种营养素的量
- 饮食推荐
可以推荐给用户一些饮食建议、计划等
- 评估助手
可以记录用户身体情况变化，评估用户每日的饮食计划

2.2 安全性需求

- 重要数据加密

对一些重要的数据按一定的算法进行加密，如用户密码、重要参数等

- 数据恢复

将各个用户的数据进行备份，允许用户申请数据的恢复，防止用户数据丢失记录日志

- 记录日志

本系统应该能够记录系统运行时所发生的所有错误，包括本机错误和网络错误。这些错误记录便于查找错误的原因。日志同时记录用户的关键性操作信息。

2.3 性能需求

- 内存消耗小

运行时不会占用过多的内存

- 反馈速度快

在用户查询或更新界面后即时进行反馈

- 抗干扰能力好

在非硬件故障或非通讯故障时，系统能够保证终端能正常运行

2.4 界面需求

- 界面风格采用当前流行的简化风格、扁平化的设计

- 把握界面之间的逻辑关系，将主要的内容集中在 2、3 个页面中，虽然次级界面有助于 我们把握逻辑关系，但是过多的页面跳转会使用户感到焦虑

- 用户所需的所有数据内容均按先后次序合理显示

- 所有的重要选项都要在主页显示

- 各条目是否合理分类于各逻辑区，并运用标题将各区域进行清晰划分

2.5 运行环境需求

需要在 Windows 7/Windows 10 操作系统上运行，后期拟移植到 Android 操作系统上。

2.6 维护性需求

- 源代码格式标准化

各个变量、文件、类、函数等命名规范进行统一

- 文档格式标准化

对于软件开发的相关文档的格式进行统一，提高维护效率

2.7 可靠性需求

除电源、硬件、操作系统、服务器管理系统外，程序不允许异常退出或崩溃。

2.8 数据字典

Food

Attribute	Data Type	Accept Null Value?	Description
Calories	Single	no	
Protein	Single	no	
Total Fat	Single	no	Including saturated fat and trans fat
Total Carbonhydrates	Single	no	Including dietary fiber and sugar
Sodium	Single	no	
Cholesterol	Single	no	
Calcium	Single	no	
Iron	Single	no	
Vitamin C	Single	no	

食物数据字典

Dish

Attribute	Data Type	Accept Null Value?	Description
IngredientName1	varchar(255)	no	
...			
IngredientName30	varchar(255)	no	
Protein	Single	no	
Total Fat	Single	no	Including saturated fat and trans fat
Total Carbonhydrates	Single	no	Including dietary fiber and sugar
Sodium	Single	no	
Cholesterol	Single	no	
Calcium	Single	no	
Iron	Single	no	
Vitamin C	Single	no	

菜肴数据字典

Meal

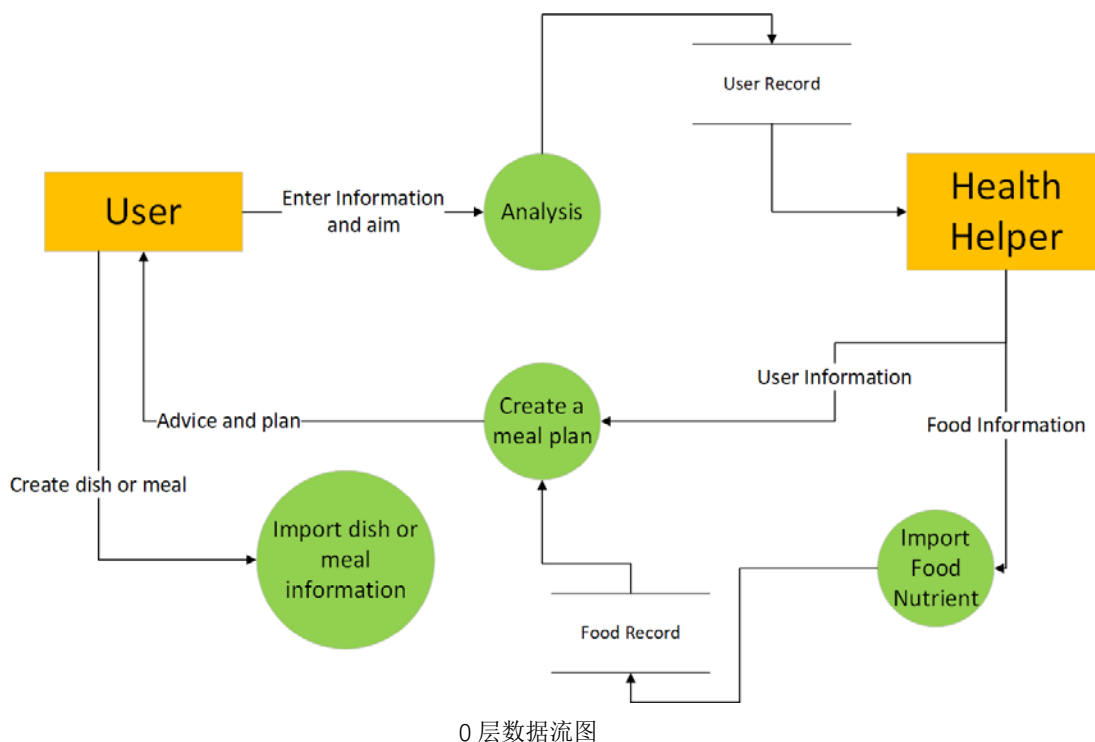
Attribute	Data Type	Accept Null Value?	Description
DishName1	varchar(255)	yes	
...			
DishName20	varchar(255)	yes	
FoodName1	varchar(255)	yes	
...			
FoodName20	varchar(255)	yes	
Protein	Single	no	
Total Fat	Single	no	Including saturated fat and trans fat
Total Carbonhydrates	Single	no	Including dietary fiber and sugar
Sodium	Single	no	
Cholesterol	Single	no	
Calcium	Single	no	
Iron	Single	no	
Vitamin C	Single	no	

一餐数据字典

2.9 数据流图



顶层数据流图



三、 可行性分析

3.1 技术可行性

考虑到时间充裕以及系统核心功能不是过于复杂等因素，个人开发饮食助手是具有可行性的。本人已经经历过之前的软件开发，对于前后端的配置已经有了一定基础。仅需画上一段时间再学习巩固一下相关知识便可进行开发。

核心功能已在上文提到，本系统不算很复杂，后期可以尝试加入更多的扩展功能，比如说经验分享，饮食计划分享等功能。

开发方面前端拟采用 Electron 框架，使用 JavaScript，HTML 和 CSS 来构建跨平台的桌面应用。Electron 可以大大提高开发者的效率，且构建出的 GUI 简洁美观，符合当下审美需求。

开发后端方面拟采用 Electron + Oracle，服务器可使用腾讯提供的云服务器，数据库也可使用其提供的云数据库。

3.2 应用可行性

就需求方面来讲，根据上文分析的结果可以看出这一方面有很大的用户需求。而就推广方面来讲，如今的各类互联网平台为个体开发者提供了更多的展示自己应用程序的机会。对于自己创建的桌面应用可以上传到 GitHub、各类应用商店、各类博客等来获得关注甚至其他开发者的加入。此外，当今越来越多的人拥有自己的个人电脑，PC 的用户基数越来越大，因此无需担心吸引不到足够的用户。