

## 二、需求规格说明书

1. 引言 .....	3
1.1 编写目的 .....	3
1.2 项目背景 .....	3
1.3 定义 .....	3
1.4 参考资料 .....	4
2. 任务概述 .....	5
2.1 目标 .....	5
2.2 运行环境 .....	5
2.3 条件与限制 .....	5
3. 数据描述 .....	6
3.1 静态数据 .....	6
3.2 动态数据 .....	6
3.3 数据库介绍 .....	6
3.4 数据词典 .....	7
3.5 实体-联系图 .....	12
3.6 状态转换图 .....	13
3.7 数据采集 .....	13
4. 功能需求 .....	14
4.1 功能划分 .....	14
4.2 功能描述 .....	14

5. 性能需求 .....	14
5.1 数据精确度 .....	14
5.2 时间特性 .....	15
5.3 适应性 .....	16
6. 运行需求 .....	16
6.1 用户界面 .....	16
6.2 硬件接口 .....	17
6.3 软件接口 .....	17
6.4 故障处理 .....	17
7. 其它需求 .....	17
7.1 可使用性 .....	17
7.2 可维护性 .....	17
7.3 可移植性 .....	18

# 1. 引言

## 1.1 编写目的

本需求规格说明书的编写旨在明确智能营养膳食系统（以下简称“系统”）的需求，为后续的项目规划、进度安排、软件开发与测试提供详细的指导。本文档旨在确保所有项目参与者对系统的预期功能和性能有共同的理解。本文档供项目管理人员、设计人员、开发人员参考。

## 1.2 项目背景

项目的委托单位：北京语言大学信息科学学院软件工程课

开发单位：22 级软件工程“智能营养膳食系统”小组

主管部门：22 级软件工程“智能营养膳食系统”小组

本系统旨在服务于对食物营养成分有特定需求的用户，包括健身爱好者、营养师、特殊疾病患者、想要合理规划饮食的人群等。系统将提供食物营养成分查询、特殊群体食物推荐、餐盘热量计算等功能。

## 1.3 定义

术语：

- 营养成分 (Nutritional Components)：食物中所含有

的维持生命与健康所必需的物质，包括碳水化合物、蛋白质、脂肪、维生素等。

- **卡路里 (Calories)**: 一种能量单位，用于衡量食物的能量含量。
- **热量**: 指食物在体内氧化过程中释放的能量，通常用卡路里来表示。

缩写词:

- **UI**: 用户界面 (User Interface)
- **API**: 应用程序编程接口 (Application Programming Interface)

## 1.4 参考资料

[1] 张海藩. 软件工程导论(第五版). 北京: 清华大学出版社, 2008

[2] 窦万峰. 软件工程方法与实践(第三版). 北京: 机械工业出版社, 2016

[3] 普莱斯曼. 软件工程: 实践者的研究方法(原书第 8 版). 北京: 机械工业出版社, 2016

## 2. 任务概述

### 2.1 目标

开发一个“智能营养膳食系统”，旨在为用户提供一个平台，通过该平台用户可以查询食物的营养成分、卡路里、维生素含量等信息，并为特定健康需求的用户提供个性化的食物推荐，系统还需提供餐盘热量计算功能，帮助用户管理日常饮食。

### 2.2 运行环境

操作系统: Microsoft Windows 2000 Advanced Server

支持环境: IIS 5.0

数据 库: Microsoft SQL Server 2000

### 2.3 条件与限制

系统必须连接相对应的数据库，必须在指定的运行环境下才能稳定运行。系统的查询为精准查询，暂不支持模糊查询。由于受到数据库数据量的限制，我们可供查询的数据量有限。同时，对于系统无法解答的问题，我们接入了大模型星火讯飞的 api，但是会有字数及运行速度上的限制。

### 3. 数据描述

#### 3.1 静态数据

食物数据库：包含食物名称、类别、营养成分信息、热量等。

#### 3.2 动态数据

**输入数据：**

用户查询请求：食物名称或类别或固定模板问题。

**输出数据：**

营养查询功能：食物的详细信息，包括营养成分、卡路里等。

慢性病膳食查询功能：根据询问的相关疾病问题推荐的食物列表及其作用。

餐盘热量计算功能：用户餐盘的总热量。

#### 3.3 数据库介绍

**名称：** ruangong fooodb

**类型：** 关系型数据库

**食物表 (food)**

字段名	数据类型	长度	描述
id	INT	10	主键，自增
name	VARCHAR	100	食物名称

image_url	VARCHAR	255	食物图片的 URL 链接
Detail_url	VARCHAR	255	食物详细信息页面的 URL 链接
calories	DECIMAL	5, 2	每 100 克热量
fat	DECIMAL	5, 2	每 100 克脂肪含量
protein	DECIMAL	5, 2	每 100 克蛋白质含量
fiber	DECIMAL	5, 2	每 100 克纤维含量
category	VARCHAR	50	食物类别

食物特征表 (food\_feature)

字段名	数据类型	长度	描述
id	INT	10	主键, 自增
food_id	VARCHAR	100	食物 ID
feature	VARCHAR	50	食物特征

### 3.4 数据词典

#### 食物信息

- **名字:** 食物信息
- **描述:** 存储关于食物种类、营养成分和卡路里的详细信息。
- **定义:** 食物信息 = 食物 ID + 食物名 + 食物图片 URL  
+ 详细信息 URL + 每 100 克热量 + 每 100 克蛋白质含量 + 每 100 克脂肪含量 + 每 100 克纤维含量 + 食物类别

### 类别

- **位置:** 食物信息数据库表
- **组织方式:** 关系型数据库表, 每种食物一行记录, 以食物 ID (food\_id) 为主键

### 食物 ID (food\_id)

- **名字:** 食物 ID
- **描述:** 系统中唯一标识每种食物的数字标识符
- **定义:**
  - **数据类型:** INT
  - **长度:** 10
- **位置:** 食物表主键列
- **组织方式:** 数值型, 自增

### 食物名 (name)

- **名字:** 食物名
- **描述:** 食物的名称, 用于用户识别和查询
- **定义:**
  - **数据类型:** VARCHAR
  - **长度:** 100
- **位置:** 食物表食物名列
- **组织方式:** 字符型

### 食物图片 URL (image\_ur1)

- **名字:** 食物图片 URL

- **描述:** 食物图片的网络链接地址, 用于显示食物图片
- **定义:**
  - **数据类型:** VARCHAR
  - **长度:** 255
- **位置:** 食物表图片 URL 列
- **组织方式:** 字符型

#### 详细信息 URL (detail\_url)

- **名字:** 详细信息 URL
- **描述:** 食物详细信息页面的网络链接地址, 用户可以点击以获取更多关于食物的信息
- **定义:**
  - **数据类型:** VARCHAR
  - **长度:** 255
- **位置:** 食物表详细信息 URL 列
- **组织方式:** 字符型

#### 食物类别 (category)

- **名字:** 食物类别
- **描述:** 食物的分类, 如水果、蔬菜、肉类等
- **定义:**
  - **数据类型:** VARCHAR
  - **长度:** 50
- **位置:** 食物表类别列

- 组织方式: 字符型

### 热量 (calories)

- 名字: 热量
- 描述: 每 100 克食物所含的卡路里数
- 定义:
  - 数据类型: DECIMAL
  - 长度: 5, 2
- 位置: 食物表热量列
- 组织方式: 数值型

### 蛋白质 (protein)

- 名字: 蛋白质
- 描述: 每 100 克食物中的蛋白质含量
- 定义:
  - 数据类型: DECIMAL
  - 长度: 5, 2
- 位置: 食物表蛋白质列
- 组织方式: 数值型

### 脂肪 (fat)

- 名字: 脂肪
- 描述: 每 100 克食物中的脂肪含量
- 定义:
  - 数据类型: DECIMAL

。 长度: 5, 2

- 位置: 食物表脂肪列
- 组织方式: 数值型

### 纤维 (fiber)

- 名字: 纤维
- 描述: 每 100 克食物中的纤维含量
- 定义:
  - 数据类型: DECIMAL
  - 长度: 5, 2
- 位置: 食物表纤维列
- 组织方式: 数值型

### 食物特征

- 名字: 食物特征
- 描述: 存储食物的特定特征, 如是否为过敏原、是否适合特定饮食 (如素食、无麸质等)
- 定义: 食物特征 = 特征 ID + 食物 ID + 特征
- 位置: 食物特征数据库表
- 组织方式: 关系型数据库表, 每种食物特征一行记录,以特征 ID (feature\_id) 为主键

### 特征 ID (feature\_id)

- 名字: 特征 ID
- 描述: 系统中唯一标识每种食物特征的数字标识符

- 定义：
  - 数据类型： INT
  - 长度： 10
- 位置： 食物特征表主键列
- 组织方式： 数值型， 自增

### 特征 (feature)

- 名字： 特征
- 描述： 描述食物的特定属性或特征，例如“低脂”、“无麸质”、“高纤维”、“过敏原”等
- 定义：
  - 数据类型： VARCHAR
  - 长度： 50
- 位置： 食物特征表特征列
- 组织方式： 字符型

## 3.5 实体-联系图

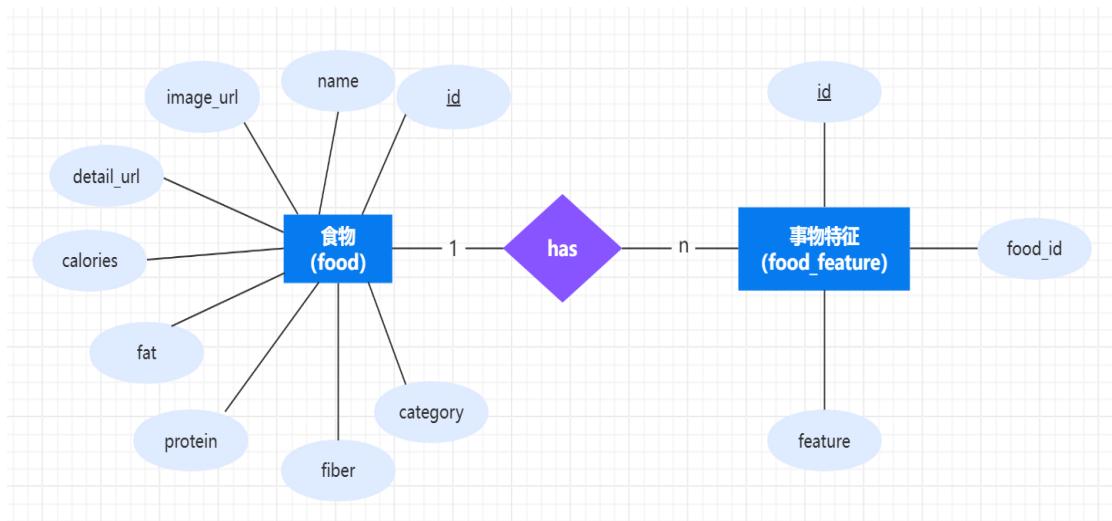


图 1 实体—联系图

### 3.6 状态转换图

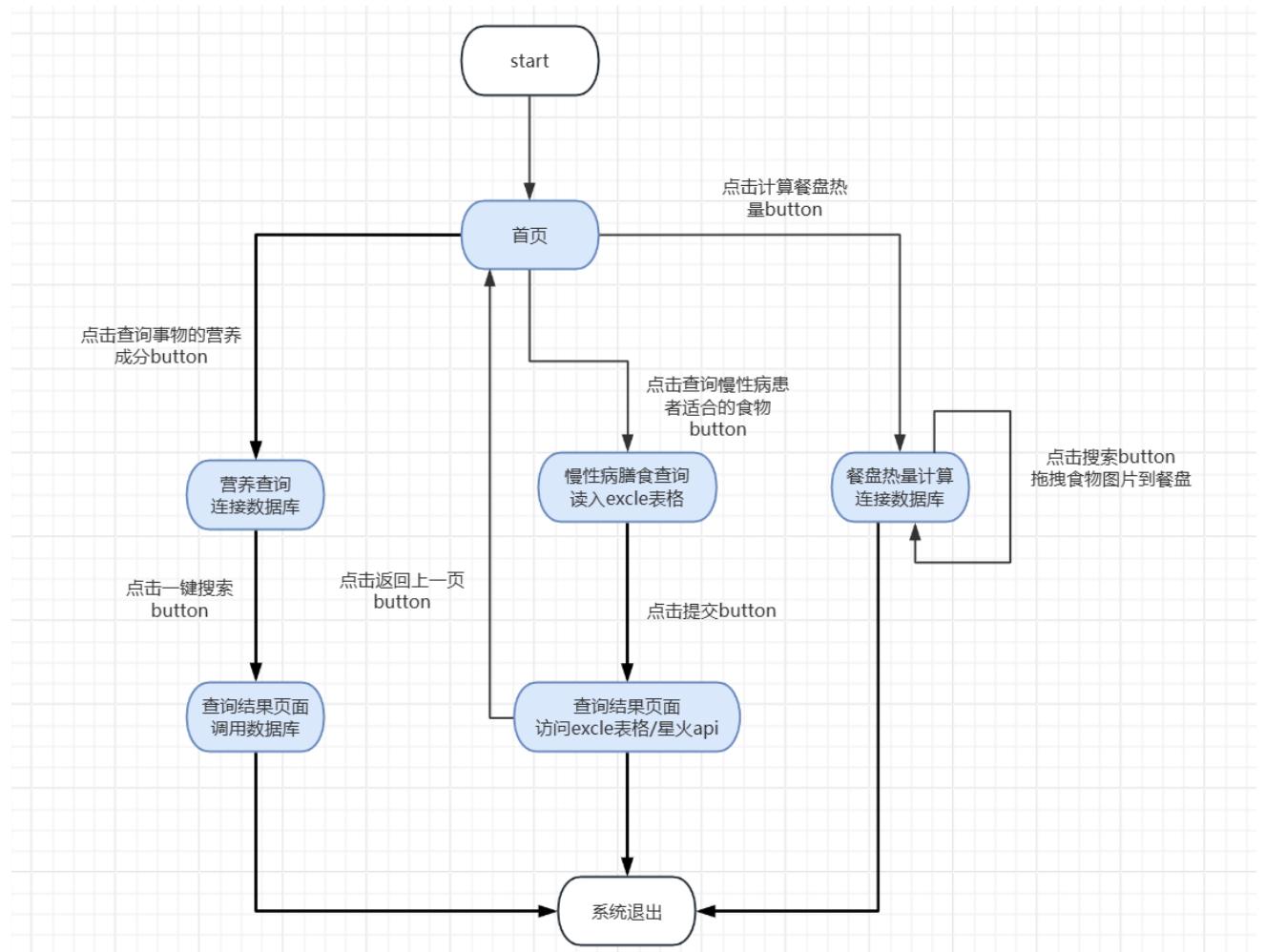


图 2 状态转换图

### 3.7 数据采集

爬取薄荷网 (<https://www.boohee.com/>) 和唤醒食物 (<https://www.foodwake.cn/>) 的数据得到。

## 4. 功能需求

### 4.1 功能划分

- 营养查询功能
- 慢性病膳食查询功能
- 餐盘热量计算功能

### 4.2 功能描述

- **营养查询功能**: 允许用户通过食物名称或类别查询营养成分和卡路里。
- **慢性病膳食查询功能**: 根据用户输入的固定模板的疾病查询问题返回适宜的食物或作用。
- **餐盘热量计算功能**: 计算用户餐盘的总热量。

## 5. 性能需求

### 5.1 数据精确度

食物表 (food) 精度表格

字段名	精度	备注
id (food_id)	INT	无需小数点，整数精度
name	VARCHAR(100)	字符串长度不超过 100 字符

image_url	VARCHAR(255)	URL 链接, 字符串长度不超过 255 字符
detail_url	VARCHAR(255)	URL 链接, 字符串长度不超过 255 字符
calories	DECIMAL(5, 2)	每 100 克热量, 整数部分 3 位, 小数 2 位
fat	DECIMAL(5, 2)	每 100 克脂肪含量, 整数部分 3 位, 小数 2 位
protein	DECIMAL(5, 2)	每 100 克蛋白质含量, 整数部分 3 位, 小数 2 位
fiber	DECIMAL(5, 2)	每 100 克纤维含量, 整数部分 3 位, 小数 2 位
category	VARCHAR(50)	食物类别, 字符串长度不超过 50 字符

食物特征表 (food\_feature) 精度表格

字段名	精度	备注
id	INT	无需小数点, 整数精度
food_id	VARCHAR(100)	食物 ID, 字符串长度不超过 100 字符
feature	VARCHAR(50)	食物特征, 字符串长度不超过 50 字符

查询结果的准确性需达到 95% 以上。

## 5.2 时间特性

**响应时间:** 系统应在 2 秒内响应用户查询。

**更新处理时间:** 数据库更新应在 5 分钟内完成。

**数据转换与传输时间:** 数据传输和转换应在 1 分钟内完成。

**运行时间:** 程序应在 5 秒内完成运行。

## 5.3 适应性

系统应能在不同的 Windows 服务器版本上运行。

系统应能适应不同的网络环境和数据库更新。

# 6. 运行需求

## 6.1 用户界面

1. 首页：导航页面，提供三个不同功能的分页面的进入 button，同时会给出一些食物营养小知识进行科普。提供页面介绍及联系方式。
2. 营养查询功能：页面显示“食物搜索”输入框，搜索框内输入“食物名称”后就显示该食物相关信息。
3. 慢性病膳食查询功能：用户通过询问固定模板的问题，即会跳转到下一页返回结果。该页面会给出可询问问题的固定模板。
4. 餐盘热量计算功能：用户通过输入不同类别的摄入食物，将搜索结果的食物图片拖拽到餐盘上，再输入食物克重，即可计算得到总热量。
  - a) 屏幕格式：清晰、直观的布局，易于用户操作。
  - b) 报表格式：提供可打印的详细报告。
  - c) 菜单格式：简洁明了的菜单选项。
  - d) 输入输出时间：用户输入应在 2 秒内得到系统响应。

## 6.2 硬件接口

内存: 512M 以上

磁盘空间: 40G 以上

CPU: 233Mhz 以上

硬盘空间: 1.5G 以上

## 6.3 软件接口

数据库接口: 与 Microsoft SQL Server 2000 的接口。

网络接口: 支持 HTTP/HTTPS 协议。

操作系统: Microsoft Windows 10

## 6.4 故障处理

系统应具备自动错误检测和恢复机制。

# 7. 其它需求

## 7.1 可使用性

系统将提供用户手册和在线帮助文档。

## 7.2 可维护性

系统的设计为模块化，根据不同功能进行不同界面的设计与程序的撰写，便于维护和升级。

### 7.3 可移植性

系统应能在不同的环境中快速部署。