系统设计说明书

团队：银河护胃队

学院：计算机与大数据学院

专业：数据科学与大数据技术

年级：2022级

2024年 10 月 31 日

目录

一、 引言

1.1 编写目的

1. 在养食记项目的需求分析阶段中，已经将系统用户对本系统的需求做了详细的阐述，这些用户需求已经在上一阶段中的网络问卷调研中获得，并在需求规格说明书中得到详尽得叙述及阐明。
2. 本阶段已在需求分析的基础上，对养食记系统做概要设计。主要解决了实现该系统需求的程序模块设计问题，接口设计，数据库设计与系统安全性等问题。在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明。
3. 开发人员可参考此概要设计报告，在概要设计对系统所做的模块结构设计的基础上，对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书，以便于了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误

1.2 项目背景

在现代快节奏的生活中，人们对于自身健康和饮食的关注度日益提高。然而，面对市场上种类繁多的食物，如何选择适合自己身体状况和目标的食物成为了一个难题。一方面，不合理的饮食可能导致各种健康问题，如肥胖、营养不良、慢性疾病等；另一方面，对于有特定目标的人群，如健身塑形、控制血糖血压、提高免疫力等，缺乏针对性的饮食指导。 同时，传统的饮食建议往往比较宽泛，缺乏个性化，无法满足不同个体的特殊需求。而且，人们在获取饮食信息时，需要花费大量时间和精力去筛选和甄别。随着人工智能技术的发展，开发一个能够根据用户填写的身体信息和期望目标，在对话过程中为用户精准推荐食物的智能食物推荐系统成为迫切需求。

- 用户群体：
  - 健康关注者：注重日常饮食对身体影响，希望通过合理饮食保持健康的人群。
  - 健身人群：包括增肌、减脂等有特定健身目标，需要相应营养支持的人。
  - 慢性疾病患者：如糖尿病、高血压、高血脂等患者，需要特殊饮食安排来辅助控制病情。
  - 特殊需求人群：例如孕妇、老年人、素食者等有特殊身体状况或饮食限制的群体。
- 应用场景：
  - 日常饮食推荐：根据用户基本身体信息（如年龄、性别、体重、身高、过敏史等），为用户推荐日常健康饮食。
  - 目标导向饮食：针对用户的特定目标（如减肥、增肌、提高免疫力等），给出符合目标的食物建议。
  - 疾病管理饮食：为慢性疾病患者推荐有助于控制病情的食物，并考虑食物与药物之间的相互作用。
  - 特殊时期饮食：为孕妇、哺乳期妇女、老年人等特殊时期或群体提供合适的食物推荐。
  - 饮食调整建议：当用户反馈身体不适或饮食变化时，系统及时调整推荐的食物。

1.3 参考资料

- UML 之类图
- UML 类图详解
- 软件系统详细设计说明书实际项目模板
- 系统功能架构图 - hanease - 博客园
- 《产品需求规格说明书》
- 详解设计模式六大原则
- 《概要设计说明书》

二、 系统总体设计

2.1 体系结构设计

三层架构

架构介绍

三层架构将系统分为表示层、业务逻辑层和数据访问层。

表示层（Presentation Layer）：这是用户与系统交互的界面。它负责接收用户输入（如在界面中填写身体信息、选择目标等），并将系统的输出展示给用户（如推荐的食物列表）。可以是网页、移动应用的界面等。

业务逻辑层（Business Logic Layer）：此层包含了系统的核心业务逻辑，相当于 MVC 中的控制器和部分模型功能。它处理表示层传来的请求，例如根据用户输入的身体信息和目标，调用相关算法和规则来生成食物推荐逻辑。同时，它还协调数据访问层和表示层之间的交互，对数据进行必要的处理和转换。

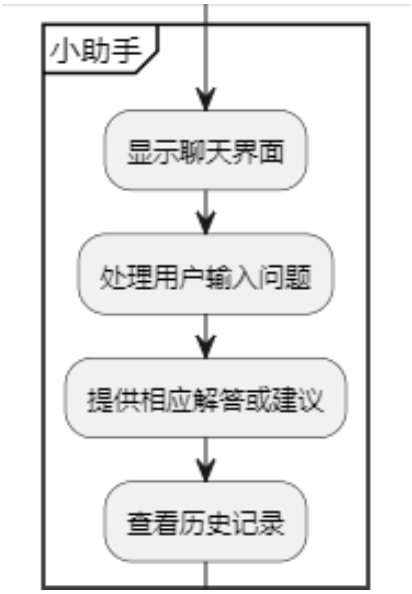
数据访问层（Data Access Layer）：主要负责与数据库或其他数据存储系统进行交互，如存储和读取用户信息、食物营养数据等。这一层确保业务逻辑层能够方便地获取和更新所需的数据。

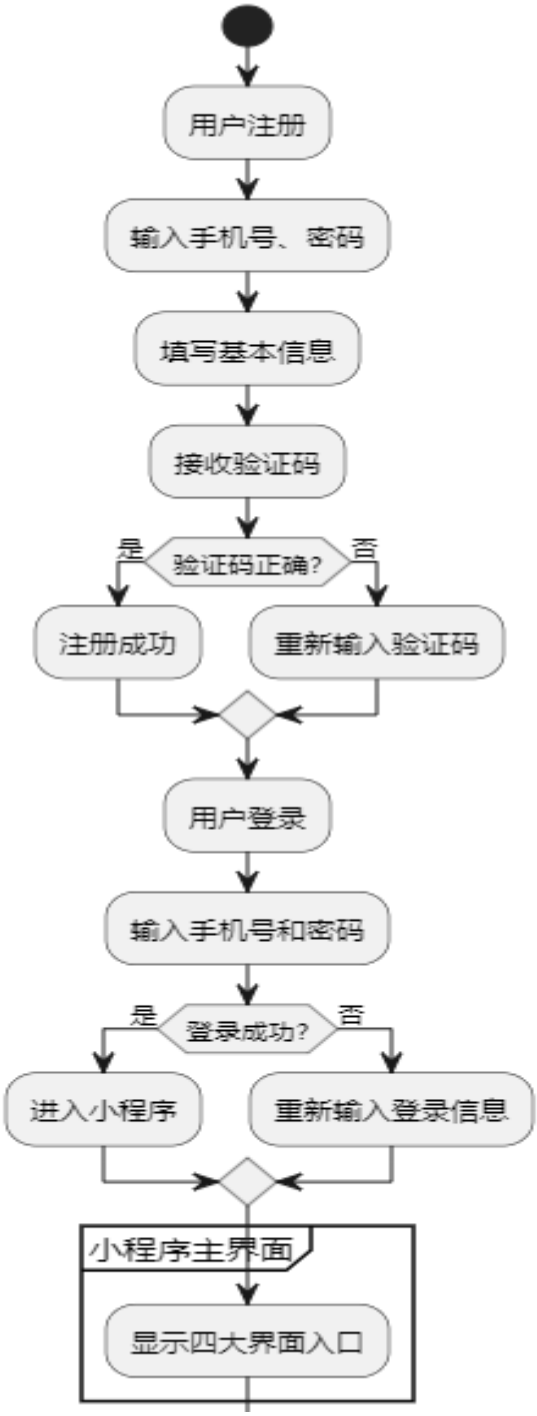
工作流程

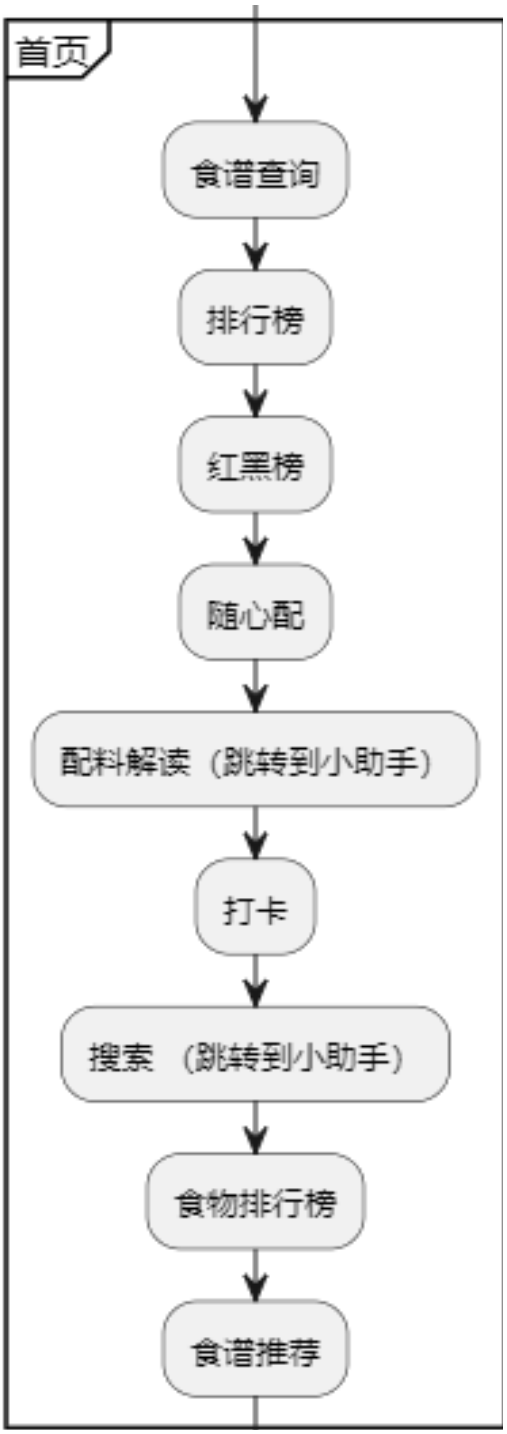
- 用户在表示层输入身体信息和目标。
- 表示层将这些信息传递给业务逻辑层。
- 业务逻辑层根据接收到的信息，运用内置的推荐算法和业务规则，向数据访问层请求相关数据（如符合特定条件的食物数据）。
- 数据访问层从数据库中检索所需数据并返回给业务逻辑层。
- 业务逻辑层处理数据后，将推荐的食物信息返回给表示层进行展示。

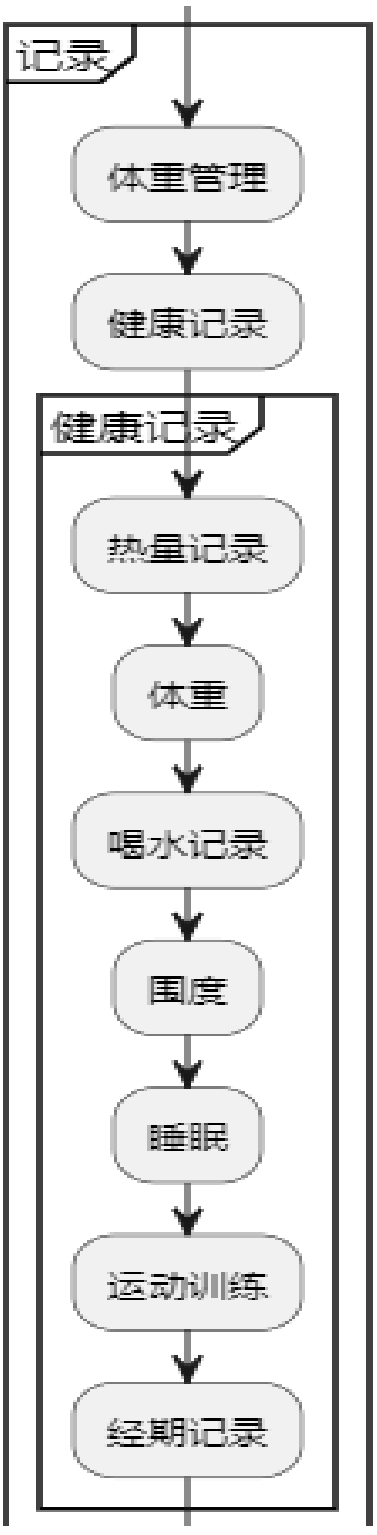
2.2 功能模块流程图

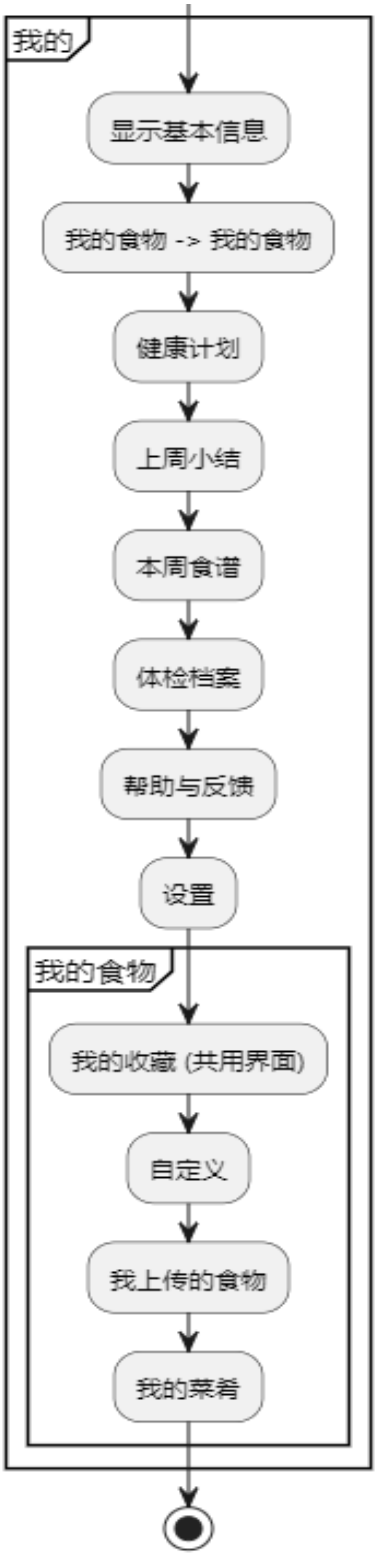
小程序端总共分为五个模块（注册登录、首页、小助手、记录、我的），划分依据主要是为了更好地满足用户需求，明晰功能定位，确保用户能够快速找到所需功能，同时保持操作的直观性和便捷性。




用户注册
输入手机号、密码
填写基本信息
接收验证码
是
验证码正确?
否
注册成功
重新输入验证码
用户登录
输入手机号和密码
是
登录成功?
否
进入小程序
重新输入登录信息
小程序主界面
显示四大界面入口


首页
食谱查询
排行榜
红黑榜
随心配
配料解读（跳转到小助手）
打卡
搜索 （跳转到小助手）
食物排行榜
食谱推荐





2.3 整体技术架构

1. 框架：前端（微信小程序开发框架）+后端（Node框架）

2. 开发工具：vscode+HBuilderX+微信开发工具+mysql+tomcat
3. 开发语言：html+css+js+java+sql
4. 数据库连接：mysql
5. 中间件：Redis、RocketNQ，Minio

2.4 设计目标

（一）提供精准个性化推荐

1. 深度定制饮食方案：通过全面分析用户的身体信息，包括但不限于年龄、性别、身高、体重、身体成分（如体脂率、肌肉量）、基础代谢率、健康状况（有无疾病、过敏史等）以及详细的饮食偏好和限制，为每个用户量身定制精准的食物推荐方案。确保推荐的食物不仅符合用户的营养需求，还能契合其口味偏好，从而提高用户对推荐食物的接受度和依从性。
2. 动态适应目标变化：实时跟踪用户设定的目标变化，无论是短期的如为了某一特殊活动进行的快速体重管理，还是长期的健康改善目标（如改善血糖控制、提升免疫力等），系统都能及时调整推荐策略。根据用户目标的进展情况，如体重的阶段性变化、健康指标的改善趋势等，动态优化食物推荐，助力用户更高效地实现目标。

（二）优化用户体验与交互

1. 便捷高效的交互流程：设计简洁直观、易于操作的用户界面和交互流程，使用户能够轻松输入身体信息和目标，并快速获取推荐食物列表。减少用户的操作步骤和输入时间，提供清晰的指引和提示，确保即使是不熟悉技术的用户也能顺利使用系统。例如，采用智能输入提示、下拉菜单选择等方式优化数据输入过程，以图表、列表等清晰易懂的形式展示推荐结果。
2. 个性化反馈与建议：在提供食物推荐的同时，为用户提供详细的营养分析和个性化的饮食建议。解释推荐食物的选择依据，如为何某种食物对于该用户的特定目标有益，以及其所含的关键营养素对身体的作用。针对用户的日常饮食习惯，给出合理的调整建议，帮助用户逐步建立健康的饮食模式。此外，根据用户的反馈（如对推荐食物的喜好或不适应情况），及时调整后续的推荐内容，实现更加个性化的服务。

（三）支持健康管理与教育

1. 健康数据跟踪与分析：根据用户的健康数据，如运动数据、睡眠数据、身体指标变化等。通过对这些多维度数据的综合分析，更全面地了解用户的生活方式和健康状况，为食物推荐提供更精准的依据。

（四）促进系统可持续发展

1. 数据驱动的优化与创新：建立完善的数据收集和分析机制，持续收集用户与系统的交互数据、用户对推荐食物的反馈数据、食物营养数据的更新以及健康领域的研究成果等。运用数据分析技术深入挖掘数据价值，发现用户需求和行为模式的变化趋势，以及系统运行中的潜在问题和优化空间。基于数据分析结果，不断优化推荐算法、完善系统功能、更新食物数据库，以保持系统的先进性和有效性，满足用户日益增长的需求和不断变化的市场环境。

2.5 设计原则

在设计软件系统时，我们遵循一系列核心原则以确保系统的可维护性、可扩展性和可靠性。这些原则包括:

（一）用户为中心原则

1. 需求导向设计：深入了解用户的需求、期望和使用场景，将用户需求贯穿于系统设计的全过程。通过用户调研、反馈收集、数据分析等方式，不断挖掘用户的潜在需求，确保系统功能和界面设计能够切实满足用户在获取个性化食物推荐、健康管理和饮食教育等方面的需求。
2. 易用性优先：注重系统的易用性和用户体验，从用户的角度出发进行交互设计和功能布局。简化操作流程，减少用户的认知负担，提供清晰明确的信息提示和引导。确保系统的界面美观、简洁，功能操作直观、便捷，使用户能够轻松上手并愉快地使用系统，提高用户的满意度和忠诚度。

（二）科学性与准确性原则

1. 基于权威知识：食物推荐算法和营养分析模型应以科学的营养学理论、研究成果和临床实践为基础。参考权威的营养指南、学术文献以及专业机构发布的标准，确保推荐的食物组合符合人体营养需求和健康原则。对食物的营养成分数据进行严格审核和验证，保证数据的准确性和可靠性。
2. 精准数据处理：在数据采集、存储和分析过程中，采用精确的算法和技术，确保用户输入的身体信息、健康目标以及食物数据等得到准确处理和解读。避免数据误差和错误分析导致的不合理推荐，为用户提供科学、准确的饮食建议，保障用户的健康权益。

（三）智能化与个性化原则

1. 先进算法应用：运用先进的人工智能和机器学习算法，如深度学习算法中的神经网络、推荐系统中的协同过滤算法和基于内容的推荐算法等，对用户数据进行深度挖掘和分析。通过分析用户的历史行为、偏好、身体特征以及与其他用户的相似性等多维度信息，实现智能化的食物推荐，为每个用户提供独特、符合其个体需求的推荐结果。
2. 持续学习与优化：系统应具备自我学习和优化的能力，随着用户数据的不断积累和更新，能够自动调整推荐模型和算法参数，不断提高推荐的准确性和个性化程度。定期对系统的推荐效果进行评估和验证，根据用户的反馈和实际使用情况，持续改进系统性能，以适应用户需求的变化和个体差异的动态性。

（四）安全性与隐私保护原则

1. 数据安全保障：建立健全的数据安全管理体系，采取严格的数据加密技术、访问控制机制和备份恢复策略，确保用户的个人信息和健康数据在存储、传输和使用过程中的安全性。防止数据泄露、篡改和丢失，保障用户数据的完整性和保密性。
2. 隐私保护合规：严格遵守相关的法律法规和隐私政策，明确告知用户数据的收集、使用和共享方式，并获得用户的明确同意。在系统设计中，充分考虑用户隐私保护的需求，对用户敏感信息进行匿名化处理和严格的权限管理，确保用户的隐私不被侵犯，使用户能够放心使用系统。

（五）可扩展性与兼容性原则

1. 系统架构灵活：采用模块化、分层化的系统架构设计，使系统具有良好的可扩展性和灵活性。能够方便地添加新的功能模块、集成新的数据源和算法，以及适应不同的技术平台和设备。确保系统在面对不断增长的用户需求和业务发展时，能够快速进行功能扩展和升级，保持系统的稳定性和性能。

（六）可靠性与稳定性原则

1. 质量保证测试：在系统开发过程中，进行全面的质量保证测试，包括单元测试、集成测试、系统测试、性能测试和安全测试等。通过严格的测试流程和方法，及时发现并修复系统中的漏洞、错误和性能瓶颈，确保系统在各种情况下都能稳定运行，为用户提供可靠的服务。
2. 容错与恢复机制：设计完善的容错机制和故障恢复策略，当系统遇到异常情况或错误时，能够自动检测并进行相应的处理，避免系统崩溃或数据丢失。例如，对用户输入的异常数据进行合理的容错处理，对系统运行中的突发故障能够快速恢复，保证系统的连续性和可用性，降低系统故障对用户使用的影响。

三、系统功能模块详细设计

3.1 首页模块

- 功能描述：首页展示有食谱查询，排行榜，红黑榜，随心配，配料解读，打卡，搜索，食物排行榜，食谱推荐的功能
  - 点击食谱查询，可查看各种食物的成分，健康指数和价格
  - 点击食物榜单，可查看各种食物的卡路里等成分的排行榜
  - 点击红黑榜，可查看在各种情况下（如减肥，糖尿病）的推荐食用的和谨慎食用的食物与相关信息
  - 点击随心配，可随心搭配当日的食谱
  - 点击配料解读和搜索都会跳转到小助手界面
  - 点击食物排行榜可查看各个方面的食物排行
  - 点击食谱推荐可直接查看系统推荐的更适合用户的食谱
- 原型界面展示：


搜索食物营养和热量
打卡
13
食谱查询
排行榜
红黑榜
随心配
配料解读
营养日记
草莓
百变冬日小甜心，低卡低糖，营养不减。
食物排行榜
益智健脑宜吃
亚麻籽
热量：505.7千卡/100.0克
查看
核桃（鲜）
热量：336.0千卡/100.0克
查看
夏威夷果
热量：718千卡/100.0克
查看
食谱推荐
更适合你的食谱
换一换
首页
小助手
记录
我的


食谱查询
筛选
土豆炖牛腩
¥25.6
土豆、牛肉、番茄、小麦、燕麦、豌豆、玉米
健康指数
155 千卡
0.5 千克
50% 蛋白质
98 脂肪
酱汁巴沙鱼
¥49.9
巴沙鱼、黄瓜、胡萝卜、酱油、空心菜、芥菜
健康指数
128 千卡
1.0 千克
60% 蛋白质
57 脂肪
菲力牛排
¥88.8
牛肉、青菜、小辣椒、玫瑰盐、黑胡椒、芦笋、橄榄油
健康指数
433 千卡
0.5 千克
50% 蛋白质
98 脂肪
地中海风情烤鱼
¥46.5
鲈鱼、柠檬汁、橄榄油、黑胡椒、罗勒、彩椒
健康指数
433 千卡
0.5 千克
50% 蛋白质
98 脂肪


食物榜单
低热量-水果类
芦荟（鲜）
4 千卡/100克
白粉桃
26 千卡/100克
甜瓜
26 千卡/100克
青木瓜
29 千卡/100克
西瓜（黄肉小瓜）
29 千卡/100克
木瓜
30 千卡/100克
杨梅
30 千卡/100克
杨桃
31 千卡/100克
减肥食物红黑榜
推荐食用
谨慎食用
主食类
肉蛋类
乳类
水果类
蔬菜类
坚果类
饮品
全部
白米虾
高蛋白：17.3g
河蚌
高蛋白：10.9g
海参
高蛋白：6g
牛肉(里脊肉)
高蛋白：22.2g
沙丁鱼
高蛋白：19.8g
黑鱼
高蛋白：18.5g
石斑鱼
高蛋白：20.2g

3.2 小助手模块

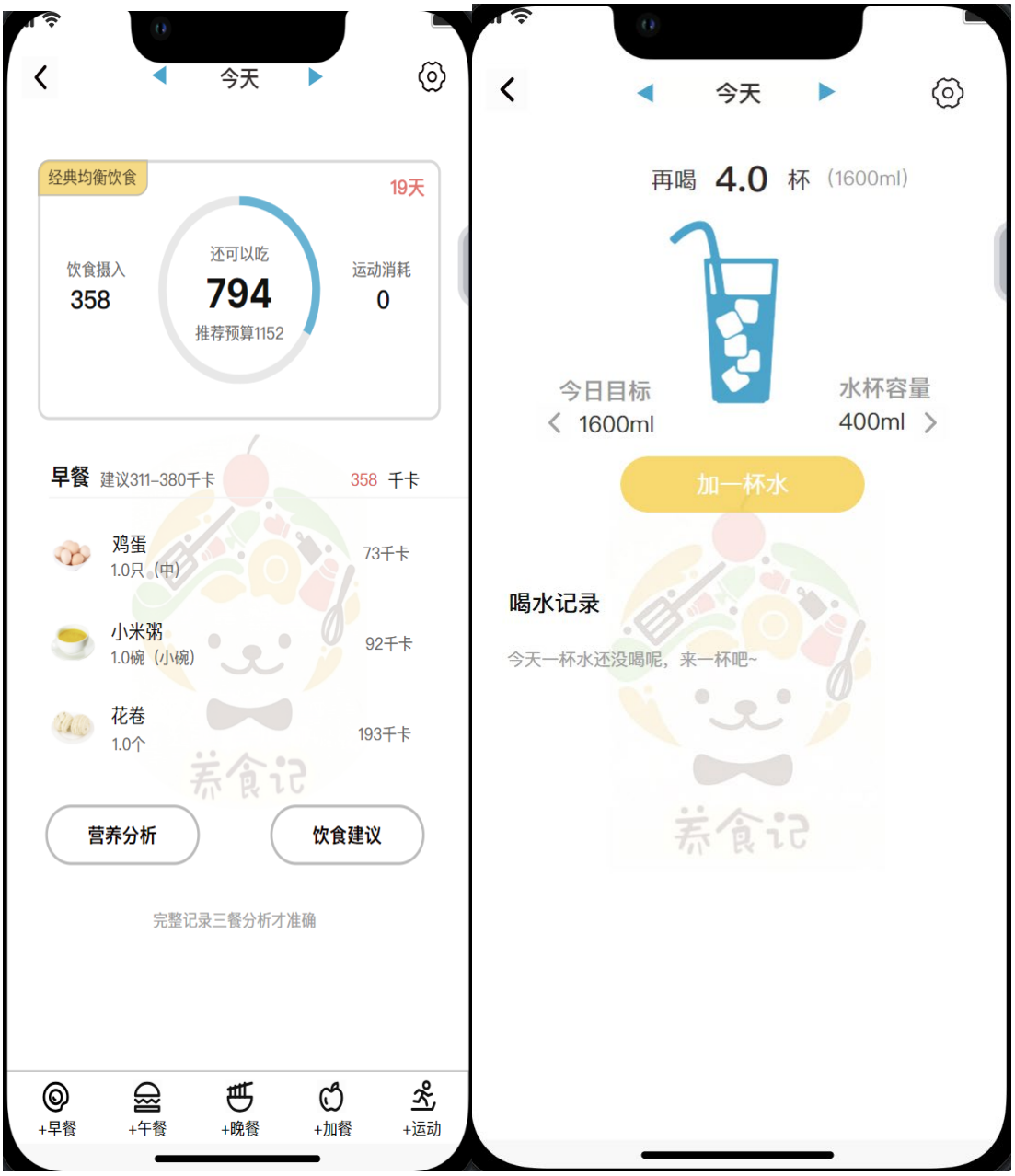
- 功能描述：用户能够与小助手对话，小助手通过分析用户的需求和用户本身的身体情况，智能回答用户的问题，用户还可查看聊天的历史记录
    - 点击搜索框，提出问题，得到答案
    - 点击查看历史记录，查看之前的聊天记录
- 原型界面展示：

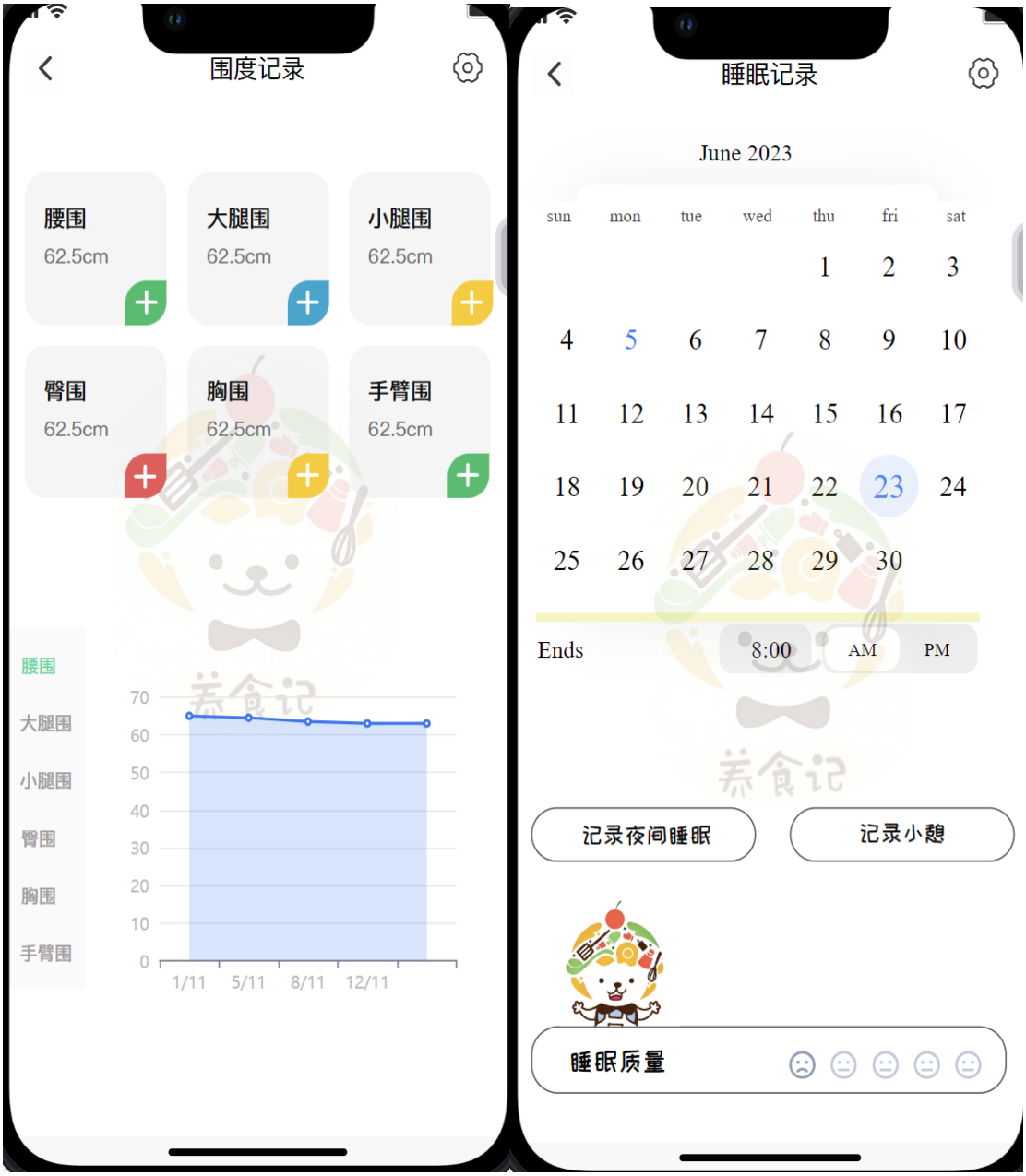

12:00
--点击查看历史记录--
你可能想问:
帮我生成适合减脂的一周的食谱
水果黄瓜与普通黄瓜的营养成分区别
我不知道今天该吃什么，请帮我选一个
人一天最多能吃几个鸡蛋?
换一批
关于食物的事情都可以问养养哦!
首页
小助手
记录
我的
12:00
--点击查看历史记录--
减肥能吃肠粉吗
减肥时可以吃肠粉，但要注意适量和选择健康的配料。肠粉通常含有米浆和淀粉，热量相对较高，所以建议搭配更多的蔬菜和瘦肉，减少油脂和酱料的使用。此外，控制摄入的总热量也是关键。如果能搭配均衡的饮食和适量的运动，适量享用肠粉是可以的。
关于食物的事情都可以问养养哦!
首页
小助手
记录
我的

3.3 记录模块

- 功能描述：记录展示有体重管理和健康记录，其中健康记录有热量记录，体重，喝水记录，维度，睡眠。
    - 点击体重管理，可记录当前体重和设定目标体重
    - 点击热量记录，可记录一天中所摄入的热量
    - 点击喝水记录，可记录今日饮水量和设定今日饮水目标
    - 点击围度，可记录自己的六围
    - 点击睡眠可记录自己一天中的睡眠时间
- 原型界面展示:


体重管理
单位：公斤
已减少2.0
51.5
初始
49.5
当前
44.5
目标
健康记录
热量记录 15：03更新
还可吃 794 千卡
+早餐
+午餐
+晚餐
+加餐
+运动
体重
1天前更新
49.5 公斤
目标 45.0 公斤
喝水记录
还要喝 1600 毫升
围度
一天前更新
睡眠
08：05更新
平均 6 小时
运动训练
一天前更新
经期记录
1天前更新
首页
小助手
记录
我的
体重记录
斤
公斤
已减去
2.0
目标44.5
记录体重
设定新目标
计划每周减重0.70公斤
51.5公斤
初始体重
开始减重
2021年11月1日
49.5公斤
最新体重（1个月前）
减重第15天
2021年11月16日
44.5公斤
目标体重
目标日期
2022年1月12


今天
经典均衡饮食
19天
饮食摄入
358
还可以吃
794
推荐预算1152
运动消耗
0
早餐 建议311-380千卡
358 千卡
鸡蛋
1.0只（中）
73千卡
小米粥
1.0碗（小碗）
92千卡
花卷
1.0个
193千卡
营养分析
饮食建议
完整记录三餐分析才准确
+早餐
+午餐
+晚餐
+加餐
+运动
今天
再喝 4.0 杯 (1600ml)
今日目标
1600ml
水杯容量
400ml
加一杯水
喝水记录
今天一杯水还没喝呢，来一杯吧~



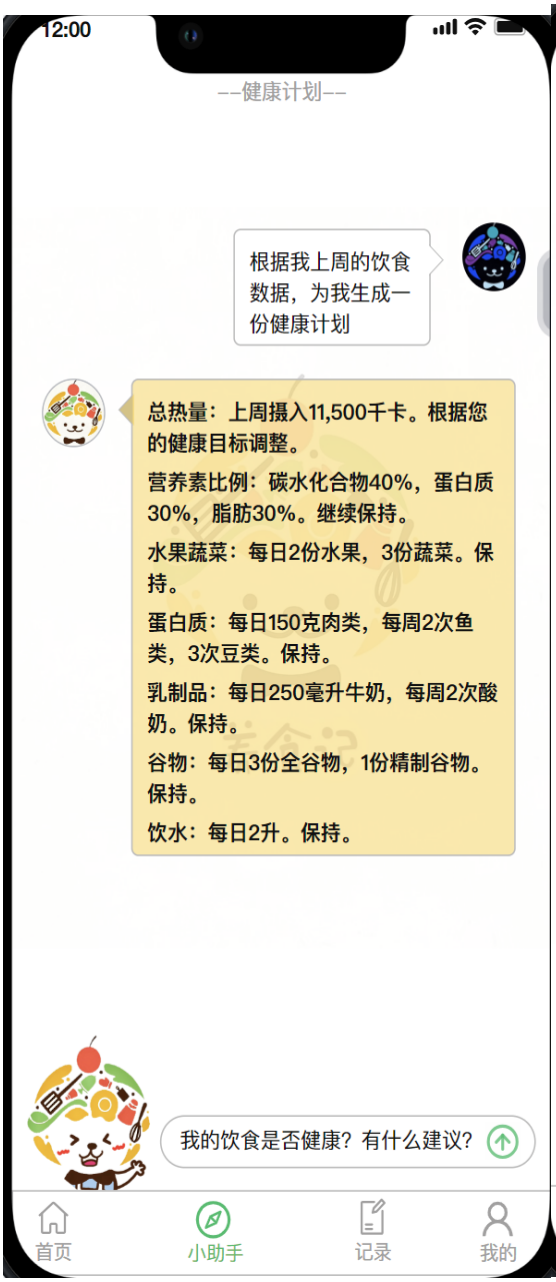
3.4 我的模块

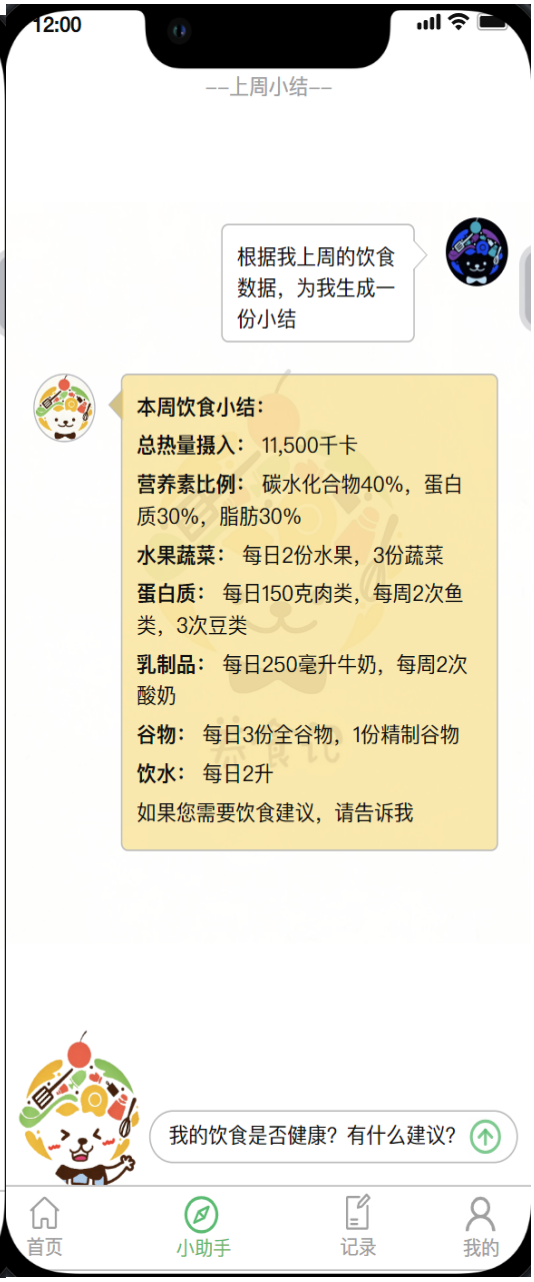
- 功能描述：我的展示有基本信息，我的食物，我的收藏，健康计划，上周小结，本周食谱，体检档案，帮助与反馈功能
    - 点击基本信息，可记录个人的基本信息
    - 点击我的食物，可查看我收藏食物，自定义食物，上传食物，我的菜肴
    - 点击我的收藏，可查看和添加收藏食物
    - 点击健康计划，可让小助手生成健康计划
    - 点击上周小结，可让小助手生成小结
    - 点击本周食谱，可查看本周的全部食谱
    - 点击体检档案，可修改查看自己的健康与生活与体检情况
    - 点击帮助与反馈，用户能提交使用遇到的问题，让我们进行改进
- 原型界面展示：


我的
13
熊养养（已黑化）
我的勋章
基本信息
我的食物
我的收藏
健康计划
上周小结
本周食谱
体检档案
帮助与反馈
首页
小助手
记录
我的
基本信息
性别
女
出生日期
2020-01-29
身高
160.5厘米
最新体重
51.5公斤
特殊时期
无
管理目标
减重
目标体重
44.5公斤
预估达成时间
10周2022-02-10
减重阶段
刚开始减重
特殊情况
无
修改资料后，预算热量可能会发生变化


我的
13
熊养养（已黑化）
我的勋章
基本信息
我的食物
我的收藏
健康计划
上周小结
本周食谱
体检档案
帮助与反馈
首页
小助手
记录
我的


我的收藏
水煮鸡胸肉
低脂高蛋白的健身餐界“清流”，让你在减脂路上“鸡”不可失！
水煮虾
海洋的鲜甜在舌尖跳跃，低卡又高蛋白，让你轻松“虾”出健康新高度！
蔬菜汤
轻盈如云，滋味浓郁，一碗下肚，暖胃又瘦身，蔬菜界的“瘦身魔法师”！
牛油果鸡肉卷
绿意盎然的牛油果拥抱嫩滑鸡肉，卷起健康与美味的完美邂逅，每一口都是对味蕾的温柔拥抱！


12:00
--健康计划--
根据我上周的饮食数据，为我生成一份健康计划
总热量：上周摄入11,500千卡。根据您的健康目标调整。
营养素比例：碳水化合物40%，蛋白质30%，脂肪30%。继续保持。
水果蔬菜：每日2份水果，3份蔬菜。保持。
蛋白质：每日150克肉类，每周2次鱼类，3次豆类。保持。
乳制品：每日250毫升牛奶，每周2次酸奶。保持。
谷物：每日3份全谷物，1份精制谷物。保持。
饮水：每日2升。保持。
我的饮食是否健康？有什么建议？
首页
小助手
记录
我的


12:00
--上周小结--
根据我上周的饮食数据，为我生成一份小结
本周饮食小结：
总热量摄入： 11,500千卡
营养素比例： 碳水化合物40%，蛋白质30%，脂肪30%
水果蔬菜： 每日2份水果，3份蔬菜
蛋白质： 每日150克肉类，每周2次鱼类，3次豆类
乳制品： 每日250毫升牛奶，每周2次酸奶
谷物： 每日3份全谷物，1份精制谷物
饮水： 每日2升
如果您需要饮食建议，请告诉我
我的饮食是否健康？有什么建议？
首页
小助手
记录
我的


周一
经典均衡饮食
饮食预算
1946
依个人信息生成
早餐 建议311-380千卡
鸡蛋
1.0只（中）
73千卡
小米粥
1.0碗（小碗）
92千卡
花卷
1.0个
193千卡
营养分析
饮食建议
完整记录三餐分析才准确
早餐
午餐
晚餐


体检档案
1.健康状况：
无慢性疾病
无过敏现象
2.生活习惯：
轻度运动
经常久坐
3.体检结果：
血糖水平正常
血脂水平正常
肾肝功能正常


12:00
您的意见，我们认真倾听
请告诉我们您遇到的问题
请留下您的联系方式（可选）
提交反馈

四、类图

4.1 用户相关类图设计思路

4.1.1 用户类（User）

1. **设计目的**：作为整个系统的核心参与者之一，用于表示使用智能食物推荐系统的个体。用户类包含了用户登录和注册所必需的信息，如手机号和密码，这些信息用于系统的身份验证和用户管理。
2. **方法设计**：注册（register）和登录（login）方法是用户与系统交互的入口操作。通过注册方法，新用户可以创建自己在系统中的账号，而登录方法则允许已注册用户访问其个人信息和系统功能。

4.1.2 基本信息类（Basic Information）

1. **设计目的**：用于存储用户的基本个人资料，这些信息对于后续的健康分析和食物推荐至关重要。例如，年龄、性别、身高和体重等信息可以用于计算身体质量指数（BMI），进而为用户的健康状况评估提供依据。
2. **方法设计**：计算 BMI（calculateBMI）方法根据用户的身高和体重数据计算出 BMI 值，这是一个衡量身体胖瘦程度与健康状况的重要指标，有助于系统更全面地了解用户的身体特征。

4.1.3 健康目标类（Health Goal）

1. **设计目的**：此类别用于记录用户在健康和饮食方面的目标设定。目标描述和目标体重等属性明确了用户期望达成的方向，例如减肥、增肌或维持特定的健康指标等。
2. **方法设计**：设定目标（setGoal）方法允许用户在系统中输入自己的目标，而评估目标进度（evaluateGoalProgress）方法则可以根据用户当前的身体数据和饮食情况，为用户提供目标达成情况的反馈，帮助用户了解自己距离目标还有多远。

4.1.4 生活与健康记录类（Life and Health Record）

1. **设计目的**：全面收集用户的生活方式、健康状况相关信息。包括血糖、血压、吸烟史、饮酒史、运动记录、喜好口味、病史、用药记录和过敏史等，这些信息为个性化的饮食推荐提供了丰富的依据。例如，系统可以根据用户的血糖情况推荐适合的食物，或者根据过敏史排除可能引起过敏的食物。

4.1.5 饮食推荐类（Diet Recommendation）

1. **设计目的**：是系统为用户提供服务的关键部分，根据用户的各类信息生成饮食推荐。
2. **方法设计**：智能推荐（intelligentRecommendation）方法运用系统内置的算法，结合用户的基本信息、健康目标和生活健康记录，为用户生成精准的饮食推荐。随机推荐（randomRecommendation）方法则可能在一些特定场景下，为用户提供一些多样化的食物选择建议。

4.2 商家相关类图设计思路

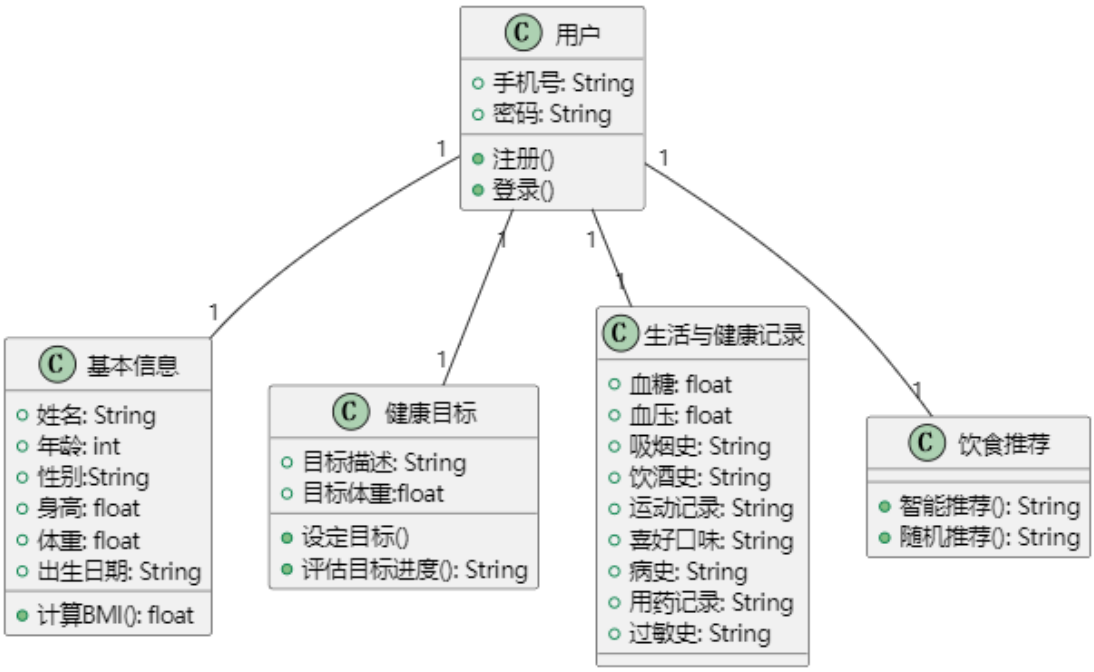
4.2.1 商家类（Merchant）

1. **设计目的**：用于表示在系统中提供食物相关服务的商家实体。商家 id、商家名称和地址等属性用于唯一标识和定位商家，菜单属性存储了商家所提供的菜品信息，商家图标则用于在用户界面中展示商家形象。
2. **方法设计**：获取商家信息（getMerchantInformation）方法可以返回商家的基本信息，方便用户浏览和选择。获取菜单（getMenu）方法用于获取商家的菜品列表，这是用户点餐或查看可购买食物的重要途径。添加菜品（addDish）、删除菜品（deleteDish）和更新商家信息（updateMerchantInformation）方法则用于商家对自己的信息和菜品进行管理，以保持信息的准确性和及时性。

4.3 菜品相关类图设计思路

4.3.1 菜品类（Dish）

1. **设计目的**：详细描述系统中每一道菜品的信息，包括菜品的基本属性、食材构成、营养成分、口味、价格、所属商家等，这些信息对于用户了解菜品和系统进行饮食推荐都非常关键。
2. **方法设计**：获取菜品信息（getDishInformation）方法用于返回菜品的详细信息，方便用户查看。计算营养成分总量（calculateTotalNutrition）方法可以根据食材及含量和营养成分等数据，计算出菜品的总营养信息，这对于饮食推荐系统分析菜品对用户健康的影响至关重要。判断过敏成分（checkAllergyIngredients）方法可以根据用户的过敏史，判断菜品是否适合用户食用。更新评分（updateRating）方法用于用户对菜品进行评价后，更新菜品的用户评分，以便其他用户参考和系统根据用户反馈优化推荐。




商家
商家id: String
商家名称: String
地址: String
菜单: List<菜品>
商家图标: String
获取商家信息(): String
获取菜单(): List<菜品>
添加菜品(菜品 新菜品): void
删除菜品(String 菜品id): void
更新商家信息(String 名称, String 地址, String 图标): void


菜品
菜品id: String
菜品名称: String
食材: List<String>
食材及含量: Map<String, Float>
营养成分: Map<String, Float>
口味: String
价格: Float
所属商家id: String
菜品图片url: String
用户评分: Float
过敏成分: List<String>
获取菜品信息(): String
计算营养成分总量(): Map<String, Float>
判断过敏成分(List<String> 用户过敏成分): Boolean
更新评分(Float 新评分): void

五、 系统安全和权限设计

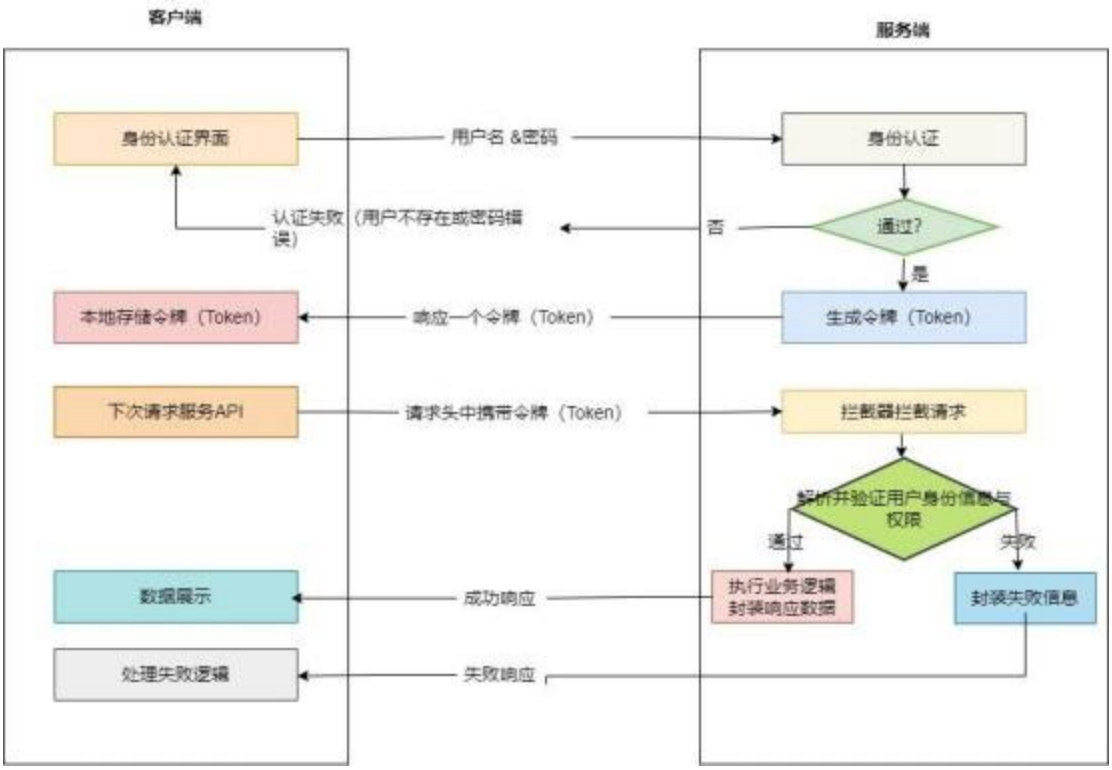
5.1 安全性

1.XSS（跨站脚本攻击）和 CSRF（跨站请求伪造） 问题

我们采用 Spring Security 安全框架负责对用户的认证和对每个系统请求的路由的鉴权。

• 使用 JWT 认证机制，采用白名单保存令牌，确保只有令牌持有者可以访问受保护的资源。

- 认证流程
  - 客户端发起认证请求：用户通过登录表单将用户名和密码发送到后端的接 口，这一过程通常是一个 HTTP POST 请求。
  - 服务端生成令牌（Token）：服务端在核对用户名和密码成功后，会将用户 的 id 等其他信息作为 JWT Payload（负载），并签名生成一个 JWT (Token)，形成的 JWT 是一个形如 xxx.yyy.zzz 的字符串。
  - 前端保存令牌（Token）：JWT 字符串作为登录成功的返回结果返回给前端，前端将返回的结果保存在本地浏览器的 localStorage 或 sessionStorage 中。
  - 后续请求携带令牌（Token）：后续用户每次请求服务端资源时，都需要将 JWT 放入 HTTP Header 的 Authorization 位（Bearer + Token），避免了XSS（跨站脚本攻击）和 CSRF（跨站请求伪造） 问题。
  - 服务端拦截请求解析并验证令牌（Token）：后端会拦截请求，检查请求头 中是否携带令牌（Token），如果存在则进行解析并验证其有效性 。例，检查签名是否正确，检查 Token 是否过期，检查 Token 的接收方是否是自己等。
  - 响应结果：验证通过后，后端使用 JWT 中包含的用户信息进行其他逻辑操 作，返回相应结果。



2.SQL 注入问题

• **使用参数化查询 ：** MyBatis Plus 支持参数化查询，这是防止 SQL 注入的最有效方法 之一 。通过使用参数化查询，用户输入的数据不会被解释为 SQL 代码，而是作为参数 传递给预编译的 SQL 语句 。例如，MyBatis Plus 中，可以使用 #{} 占位符来实现参数 化查询，这会自动将用户输入作为参数绑定到 SQL 语句中，而不是直接拼接，从而避 免了 SQL 注入的风险 。

• **配置 SQL 注入过滤器：** MyBatis Plus 允许配置 SQL 注入过滤器， 以对输入数据进行过滤和检查，防止恶意 SQL 代码的注入 。这些过滤器可以在 SQL 执行前对 SQL 语句进行扫描和处理，从而提供额外的安全层 。

• **使用预编译的 SQL 语句：** MyBatis Plus 在执行 SQL 语句时，会使用预编译的 SQL 语句，这种方式可以提高安全性， 因为 SQL 注入攻击通常发生在 SQL 语句的编译阶 段。通过预编译，MyBatis Plus 确保了用户输入不会被解释为 SQL 代码的一部分 。 （SQL 语句的预编译是指在数据库操作之前，先对 SQL 语句进行编译，然后在执行 时，将参数直接替换到编译好的 SQL 语句模板中的过程 。）

3.DDOS 攻击

将服务器部署在华为云服务器上，利用华为云服务器提供的一套完整 DDoS 解决方案，和丰富的带宽资源抵御 DDOS 攻击。

4.敏感数据保护

数据存储层面，统一采用采用 MD5 消息摘要算法对隐私信息进行保护， 以免泄露重要信息。

5.2 权限设计

1. 访客： 浏览到小程序点进来查看小程序 。这类用户可以看到随机的饮食推荐信息 、食物排行榜信息 、 以及使用小助手的简单功能。

2. 用户：用户注册账号，登录后可看到根据个人定制化的饮食推荐信息 、个人打卡记录、完整的小助手功能、完整的健康记录功能。可以进行更新个人信息，AI互动问答、健康食谱制作等操作。

3.后台管理员：使用网页端 。需要了解用户的活跃情况 、食物排行榜等， 并据此进行数据分析，作出业务决策与查看后台操作日志；根据数据分析得到的潜在用户需求更新营养库；需要收集用户的历史饮食记录和菜品点击率选择合适的个性化推荐算法，配置是基于菜品热门推荐还是基于菜品分类推荐。

六、性能设计

- 文件的最大上传大小为 200MB，视情况予以修改，而上传文件的时间不应超过 5 秒
- 用户的普遍接口响应时间应控制在 1 秒以内。
- 如果页面长时间无响应，系统应提示 403 错误，并建议用户刷新页面。
- 系统应具有良好的吞吐量， 以应对高并发场景下的性能问题。

七、系统处理规定

7.1 输入输出要求

• 系统输入和输出数据必须以 JSON 格式进行 。这包括但不限于 API 请求和响应 、文件上传和下载 、 以及系统日志的记录 。JSON 格式应遵循RFC 8259 标准，确保数据 结构的清晰和一致性。

• 输入数据必须进行有效性验证，包括数据类型 、格式和值的范围， 以确保数据的准确 性和完整性。

• 输出数据应包含必要的元数据，如状态码， 以便于数据的追踪和调试。

7.2 数据管理能力要求

• 系统应具备高效的数据存储 、检索 、更新和删除能力 。这包括但不限于数据库的索引 优化 、查询优化和数据缓存策略。

• 数据备份和恢复机制必须定期进行， 以防止数据丢失和系统故障。

• 系统应支持数据的安全性要求，包括数据加密 、访问控制和审计日志。

7.3 故障处理要求

• 故障处理流程应包括故障转移 、数据恢复和系统重启等机制， 以确保系统的高可用性。

• 系统应记录详细的故障日志，包括故障时间 、故障类型 、影响范围和处理结果， 以便 于事后分析和改进。

7.4 系统出错处理设计

1. 数据库连接错误

a. 可能原因：

。 数据库服务器地址错误：连接字符串中的服务器地址可能有误，导致无法 找到数据库服务器。

。 端口号错误：数据库服务可能在非默认端口上运行，而连接字符串中使用 的是默认端口。

。 数据库服务未运行：数据库服务器可能没有启动，或者服务出现故障。

。 网络问题：网络配置错误或网络连接问题可能导致无法连接到数据库服务 器。

。 认证信息错误：用户名或密码错误，或者用户没有足够的权限访问数据 库。

b. 应对措施：

。 检查连接字符串：核实数据库服务器地址 、端口号 、数据库名称 、用户名 和密码 等信息是否正确。

。 检查数据库服务状态：确保数据库服务正在运行，并且可以接受连接。

。 检查网络连接：使用如 ping等工具检查网络连接是否正常。

。 检查权限：确保数据库用户具有连接和操作数据库的权限。

2. 输入错误

a. 可能原因：

。 用户输入不准确：用户在输入数据时可能犯了拼写错误或输入了不完整的 信息。

。 缺乏输入验证：应用程序可能没有正确地验证输入数据，导致无效或错误 的数据被 接受。

b. 应对措施：

。 改进用户界面：确保输入字段清晰标记，并且用户容易理解所需输入的数 据类型。

。 输入验证：在前端和后端实施严格的输入验证，确保数据的正确性和完整 性。

。 错误提示：为用户提供即时的反馈，指出输入错误并指导他们如何纠正。

3. 第三方服务错误

a. 可能原因：

。 服务不可用：依赖的第三方服务出现故障或维护，导致无法使用。

。 API 变更：第三方服务 API 更新，导致现有集成出现问题。

。 配额限制：超出了第三方服务的使用配额。

b. 应对措施：

。 服务监控：监控依赖的第三方服务状态，及时发现问题。

。 版本控制：跟踪第三方服务的版本更新，及时适配 API 变更。

。 配额管理：监控使用情况，确保不超过配额限制，必要时申请增加配额。

。 备选方案：为关键服务准备备选方案， 以应对第三方服务的不可用。

7.5 其他专门要求

系统应遵循相关的法律法规和行业标准，包括数据保护法规、行业安全标准等。

系统应具备良好的扩展性和可维护性，以便于未来的升级和维护。

系统应进行定期的性能测试和安全审计，以确保系统的稳定性和安全性。