



北華大學

本科毕业设计（论文）

题目名称： 基于微信小程序的膳食营养管理系统的设计与实现

学 院： 计算机科学技术学院

专业年级： 软件工程 18 级

学生姓名： 常媛媛

班级学号： 4 班 27 号

校内指导教师： 王思飞

企业指导教师： 刘威(天津优度软件技术有限公司 组长)

二〇二二 年 五 月

摘 要

膳食营养管理已经成为目前乃至今后人们关注且热议的话题。近些年，我国也开始利用膳食管理来解决人类健康问题。本文以实际情况出发，对比传统的膳食营养推荐模式，人们现在更需要便携、易于操作、可针对自己的身体情况进行日常化管理的软件模式，这与传统的推荐 App 并不匹配，所以本文设计实现开发基于微信小程序的膳食营养管理系统，不仅微信小程序的操作方便简单，还能达到人们对饮食管理的要求。

本文所论述的基于微信小程序的膳食营养管理系统，针对所有关注健康的用户，从实际出发解决人们在日常膳食管理中遇到的问题。本文基于软件工程思想以系统的需求、可行性、系统的详细功能设计和编写测试等方向来分析论述该系统。系统采用开放式服务体系，以 Java 语言为主，利用 SpringBoot 框架和 Uniapp 框架搭建服务。系统具有简洁性、可靠性、响应快、易扩展等特点。系统的创新点在于采用微信小程序的方式开发系统并用穷举的方法推荐个性化膳食套餐，尽可能满足用户需求，依据国际《食物热量表》，合理科学推荐饮食套餐。让更多关注饮食的人可以用更少的时间保持健康。

关键词：膳食管理；开放式服务体系；Uniapp 框架；穷举

Abstract

Dietary nutrition management has become a hot topic at present and even in the future. In recent years, China has also begun to use dietary management to solve human health problems. Based on the actual situation, compared with the traditional dietary nutrition recommendation mode, people now need a software mode that is portable, easy to operate and can carry out daily management according to their own physical conditions, which does not match the traditional recommended app. Therefore, this paper designs and implements a dietary nutrition management system based on wechat applet, which is not only convenient and simple to operate, but also meets people's requirements for dietary management.

The diet and nutrition management system based on wechat applet discussed in this paper aims to solve the problems people encounter in daily diet management from the reality for all users who care about health. Based on the idea of software engineering, this paper analyzes and discusses the system from the aspects of system requirements, feasibility, detailed function design and writing test. The system adopts an open service system, mainly Java language, and uses springboot framework and uniapp framework to build services. The system has the characteristics of simplicity, reliability, fast response and easy expansion. The innovation of the system is to develop the system by means of wechat applet and recommend personalized meal packages by exhaustive methods to meet the needs of users as much as possible, and reasonably and scientifically recommend meal packages according to the international food calorie table. People who pay more attention to their diet can spend less time keeping healthy.

Key words: food management; open service system; uniapp framework; exhaustive

目 录

摘 要	I
Abstract	II
第 1 章 绪论	1
1.1 课题研究背景	1
1.2 国内外动态分析	1
1.3 研究意义	2
1.3.1 理论意义	2
1.3.2 实际意义	2
第 2 章 方案论证	3
2.1 架构选择	3
2.1.1 微信小程序	3
2.1.2 Web 系统	3
2.1.3 应用程序	3
2.1.4 架构选择	3
2.2 框架选择	4
2.2.1 Uniapp 框架	4
2.2.2 Mina 框架	4
2.2.3 SpringBoot 框架	4
2.2.4 SSM 框架	4
2.2.5 框架选择	5
2.3 数据库选择	5
2.3.1 Oracle 数据库	5
2.3.2 MySQL 数据库	5
2.3.3 数据库选择	5
2.4 算法选择	5
2.4.1 穷举法	5
2.4.2 贪心算法	5
2.4.2 基于内容的推荐算法	6
2.4.3 算法选择	6
2.5 创新点	6
2.6 开发工具	6
2.7 方案设计	6
2.8 可行性分析	7
2.8.1 技术可行性	7
2.8.2 操作可行性	7
2.8.3 法律可行性	7

2.8.4	经济可行性	7
2.8.5	总结	7
第 3 章	需求分析	8
3.1	总体目标	8
3.2	用户需求分析	8
3.2.1	用户角色需求分析	8
3.2.2	系统用例设计图	8
3.3	功能需求分析	10
3.3.1	总体设计类图	10
3.3.2	用户信息管理	11
3.3.3	膳食管理	12
3.3.4	饮食推荐	13
3.3.5	公告管理	14
3.3.6	反馈管理	14
3.3.7	推荐贴士管理	15
3.4	数据分析	16
3.4.1	数据流图	16
3.4.2	数据字典	17
3.4.3	数据库 E-R 图	19
3.5	性能需求分析	20
3.5.1	性能分析	20
3.5.2	运行需求	20
第 4 章	系统分析与设计	21
4.1	概要设计	21
4.1.1	体系结构设计	21
4.1.2	功能流程	22
4.2	数据设计	25
4.2.1	数据库逻辑结构设计	25
4.2.2	数据库表设计	27
4.3	详细设计	30
4.3.1	用户注册模块设计	30
4.3.2	用户登录模块设计	31
4.3.3	用户管理模块	33
4.3.4	饮食推荐贴士模块	35
4.3.5	公告管理模块	37
4.3.6	膳食管理模块	39
4.3.7	食谱推荐模块	42
第 5 章	系统测试与结果分析	48

5.1	测试导图	48
5.2	单元测试	48
5.3	系统功能测试	50
5.4	系统性能测试	51
5.5	系统测试报告	52
结 论	54
致 谢	55
参考文献	56
附 录	57

第 1 章 绪论

本章将从课题研究背景、国内外动态分析、研究意义这三个方面来为本系统的设计与实现做总体论述。

1.1 课题研究背景

随着我们生活的节奏越来越快,人们更多的注重于自身的食物喜好,而很少顾及营养素的合理搭配和均衡问题,并且不规律的饮食习惯也正是造成人类身体损伤的主要因素。据卫生研究资料证实:胃肠道疾病的发生率正逐渐增加,这也是现在快节奏生活带来的健康隐患。而且一天三餐对人们补充一天消耗和身体需要的营养是至关重要的,每日通过食物摄入的营养正是维持生命的根本物质,营养提供给细胞所需要的能量,以及到达身体各个组织部位的协调动作,使人体保持健康的生理机制。从而良好的应对了日常生活中的压力和问题。

针对以上涉及到的问题,本系统基于微信小程序的膳食营养管理系统,根据食物营养成分比,直观感受食物给我们带来的能量,合理评估科学健康的食物营养搭配方案,合理推荐饮食;并根据一日三餐个性化推荐套餐。让更多关注饮食的人可以用更少的时间保持健康,爱上健康生活。

1.2 国内外动态分析

目前,随着互联网与饮食文化的发展,国内外居民膳食营养也逐渐成为人们关注并热议的话题,因此膳食营养管理也正朝着更加科学、健康的方向发展。合理的营养膳食是保障人体健康,降低慢性病发作风险的重要举措。人们对食物属性的了解搭配和膳食套餐的选择,也是保证身体健康的有效手段。80 年代以来,食品与营养援助政策更加聚焦营养和健康,新增营养教育等项目进一步丰富了食品与营养援助政策工具箱,国内外对膳食营养的管理也推向高潮。

例如,国外流行的 **Paprika Recipe Manager** 是专业的食谱管理软件,可以帮助用户更好的管理食谱,制定膳食计划和创建购物清单。软件借助强大的子类组织和高级搜索工具,帮助使用者快速找到食谱;存储每种食谱的营养信息和个性化注释;智能杂货清单自动组合食材,并分类;支持无缝云同步,在所有设备无缝同步食谱等;支持离线访问、食谱导入等功能。让管理饮食成为触手可及的事。**Nutridays**、**MasterCook** 等软件都是国外用于管理膳食营养的热门软件。

近年来,国内同样做了有针对性的研究,上海医院与上海理工大学联合商讨并研发的“营养膳食分析与配置”软件,针对特殊疾病人群开发出膳食分析、日常食谱定

制、治疗膳食食谱定制等功能。并且我国为了控制逐年上升的肥胖症、“三高”、糖尿病等慢性疾病的发生，针对这些特定人群的膳食配餐系统已逐步完善并使用起来，确保了膳食营养管理高效率、高质量和智能化发展。

由国内外现状可知，膳食营养管理越来越收到重视；国内针对管理膳食营养提高人体机能的发展势头良好。

综上，本系统设计实现一个基于微信小程序的膳食营养管理系统对人们的生活有着重要意义。

1.3 研究意义

1.3.1 理论意义

基于微信小程序的膳食营养管理系统，通过软件的方式对人民的膳食营养进行管理，回避传统的昂贵膳食营养搭配医师的管理，利用计算机、手机辅助管理膳食营养，让科技走进人们的生活，对膳食营养管理以软硬件开发辅助的模式提供了佐证，开辟了新的市场和方向。

1.3.2 实际意义

相比于过去时代，人们已基本解决温饱问题，逐渐重视膳食营养搭配问题，基于微信小程序的膳食营养管理的发展也发生了根本性变化。为了更加便捷合理的为关注饮食健康人士服务，也为了更好地提升全民健康水平，将科学营养配比与个人实际相结合，借助计算机网络技术，建立一个完善的膳食营养管理小程序，对于合理膳食搭配，关注全民健康有十分重要的意义。

第 2 章 方案论证

利用软件工程思想解决问题时，确定系统开发方案，确定技术选型，是至关重要的一步，是整个项目的基石。合适且正确的开发方案，不仅可以提高开发效率，降低开发成本，更是确保软件开发成功的关键。本章将详细论述本系统的方案选择及可行性分析。

2.1 架构选择

2.1.1 微信小程序

微信小程序依托于微信这个十亿多流量的大平台，获客方便。并且小程序是一种不需下载安装即可使用的应用，用完即走，随时可用，无需安装和卸载，不会占用手机太多空间，节省流量。

2.1.2 Web 系统

网站式管理系统主要包括前端和后端对相互配合开发，一般为 B/S 模式，即浏览器和服务端。基于 Web 的管理系统，因其开发周期短、与用户平台无关、易于实现交互式应用的特点，逐渐成为开发管理系统的首选。

2.1.3 应用程序

应用程序即 App，移动 App 应用的优势主要是给我们的生活带来的方便、实用性。改变了以前人们浏览网页，上网购物只能通过浏览器来实现的缺点。手机 App 软件不仅实现了一键达到目的要求，更是延伸到了生活和工作的各个领域，是当今时代不可或缺的产品。

2.1.4 架构选择

本系统选择微信小程序架构开发。相比于 Web 系统开发，微信小程序更加方便快捷，受众面更加广泛。相比于手机应用程序 App 来说，微信小程序无需卸载安装，占用内存小，节省流量的特点更是用户选择倾向的关键，所以本系统最终选择微信小程序来开发。

2.2 框架选择

2.2.1 Uniapp 框架

Uniapp 是一个跨平台的前端框架，使用 Vue 的语法规则、微信小程序的标签和 API，支持多端开发，即开发者编写一套代码，可编译到 Android、小程序、QQ 小程序等多个平台。

Uniapp 具有更多交叉端、更好的性能体验、较低的学习成本的优点。Uniapp 框架使用 Cordova 的进阶版，也就是把 Web 代码打包到本地，另外 Uniapp 支持使用 Weex 框架拓展性能，通过桥的功能把 Vue 控件映射为原生控件进行渲染，提高开发性能。

2.2.2 Mina 框架

Mina 框架主要是基于 TCP/IP、UDP/IP 协议栈的通信框架，提供事件驱动、异步操作的编程模型。底层依赖 Java NIO 库，上层提供基于事件异步接口，核心是一个响应的数据绑定系统，只需关注上层逻辑，便可轻松保持数据和视图同步，快速构建应用。

2.2.3 SpringBoot 框架

SpringBoot 框架由 Pivotal 团队开发，是 Spring 框架的升华版。SpringBoot 框架整合了大量的开发框架，SpringBoot 框架的出现简化了 Spring 应用的搭建和开发过程，内嵌 Tomcat 服务器，支持 jar/war 项目部署方式，使得开发人员在保证功能实现的同时在配置上减少时间。

SpringBoot 框架是基于 Spring 开发的，SpringBoot 框架的核心思想是约定大于配置，默认进行很多设置，多数 SpringBoot 系统需要进行较少的 Spring 配置，减少了大量 xml 配置，自动依赖注入，简便快捷。

SpringBoot 框架集成 Mybatis。Mybatis 是 Java 数据持久层的一个开源框架。Mybatis 几乎消除了 JDBC 所有代码和手工设置的参数。Mybatis 提供了简单好用的 API 和数据库交互。在 SpringBoot 框架中，只需在 POM 文件中添加 Mybatis 的依赖包，实现数据库的持久化。

2.2.4 SSM 框架

Java SSM 框架即指 Spring+SpringMVC+MyBatis 的简称，由 Spring 框架和 Mybatis 框架，两个开源整合（SpringMVC 是 Spring 中的部分内容），常作为数据源较简单的 Web 项目的框架。

相比于之前的 SSH（Spring+Struts+Hibernate），SSM 更加轻量化、灵活化，是目前业界主流的 Java Web 开发框架

2.2.5 框架选择

本系统选择 Uniapp 框架和 SpringBoot 框架为后序程序开发的基础。Uniapp 不限制底层 API 调用, 在小程序端, Uniapp 支持直接编写微信原生代码。并且 Uniapp 封装的组件与微信小程序相似, 并且基于 Vue.js 组件齐全, 可视化开发, 再结合 HBuilder 开发速度快。SpringBoot 开发简化了 SSM 的开发配置, 并且 SpringBoot 不推荐使用 JSP, 主推的是 Thymeleaf 和 Freemarker 等模板引擎, 系统化开发, 语法简洁, 功能强大。Uniapp 与 SpringBoot 二者结合, 开发微信小程序方便快捷, 连接顺畅。

2.3 数据库选择

2.3.1 Oracle 数据库

Oracle 数据库支持大数据库、多用户的高性能的事务处理, 遵守数据存取语言、操作系统、用户接口和网络通信协议的工业标准(SQL)。且支持分布式数据库和分布处理, 具有可移植性、可兼容性和可连接性, 是全球化、跨平台的数据库。Oracle 相对于其他 RDBMS 来讲较复杂, 同时提供 GUI 和命令行, 在 WindowsNT 和 Unix 下操作相同。

2.3.2 MySQL 数据库

MySQL 中提供了多种数据库存储引擎, 每种引擎都有自己的优点, 适合不同的应用。MySQL 是一种关系型数据库, 几乎所有重要的 DBMS 都支持 SQL, 体积小, 灵活性强, 速度快, 且源码开放, 成本低, 可移植性强。

2.3.3 数据库选择

由于 Oracle 对于硬件要求更高, 价格昂贵。MySQL 开源且数量级足够, 所以最终选择 MySQL 数据库作为本文的最终数据库。

2.4 算法选择

2.4.1 穷举法

穷举法基本思想是逐一列举问题所涉及的所有情形, 并根据问题提出的条件检验哪些是问题的解。即罗列出所有问题的可能解, 找到最终的目标答案, 此算法准确性强, 理解简单。穷举法更多的依赖计算机的运行速度和精确度, 需要对所有解的情况进行测试, 直到找出最符合要求的答案。

2.4.2 贪心算法

贪心算法(又称贪婪算法)是指, 在对问题求解时, 它总是在当前做出最佳选择。贪心选择标准就是选择“当前最好”的决策, 贪心算法根据这个标准进行决策, 将原

问题变成一个相似但规模更小的子问题，而后每一步选出来的一定是原问题整体最优解的一部分。如果一个问题贪心后只剩下一个子问题且有最优子结构，那么该问题就可以使用贪心算法。当一个问题的整体最优解包含其子问题的最优解时，我们称次问题具有最优子结构性质。

2.4.2 基于内容的推荐算法

基于内容的推荐算法思路主要有三种：基于用户历史行为记录做推荐；用户和标的物特征都用显式的标签表示，利用该表示做推荐；用户和标的物嵌入到同一个向量空间，基于向量相似做推荐。其优点主要是：用户之间的独立性、好的解释性、新的标签可立即推荐。

2.4.3 算法选择

本程序最终选用穷举法解决饮食套餐推荐问题。相比于贪心算法，穷举法所列举推荐搭配更为全面，用户有更多选择。相比于基于内容的推荐算法，本系统不要求根据用户的饮食喜欢推荐，而是脱离不健康饮食喜好，生成独立健康的饮食内容，所以本系统选用穷举法，罗列选出饮食搭配套餐。

2.5 创新点

相比于传统的膳食管理 App，此基于微信小程序的膳食营养管理系统以微信小程序作为客户端，操作简单便捷，吸引各类用户参与使用。此外本系统采用 Json Web token 加密，同时加盐的方式进行校验登录，为每个用户生成唯一的 token 值，拦截器会根据当前用户的 token 值判断当前用户的登录状态，防止用户直接通过 URL 的方式侵入系统，大大提高了系统的安全性。本系统个性化推荐膳食营养套餐，根据每位用户身高体重等信息计算身体质量指数，并且根据国际膳食营养标准，推荐三组不同种类的膳食营养套餐，本膳食营养套餐的推荐采用穷举的算法方式，尽可能的满足用户对套餐的选择需求。

2.6 开发工具

系统能否正确、安全、可靠、高效的运行与选择的开发平台息息相关。此系统采用开放式服务体系，选用 Java 语言，Maven 构建。在微信开发者工具、HBuilderX 和 IDEA 中开发。

2.7 方案设计

综上本文对系统的开发框架，数据库进行了选型，最终确定本系统依照如下方案开发：

系统结构：开放式服务体系；
客户端开发框架：Uniapp；
服务端开发框架：SpringBoot；
算法：穷举法；
数据库：MySQL；
开发语言：Java；
开发环境：Win10。

2.8 可行性分析

2.8.1 技术可行性

膳食营养管理小程序的开发是利用 SpringBoot 框架、MySQL 数据库、Uniapp 框架，且这些都是开源的，并且有很多相关的博客会提供技术上的相关资料，国内外也有很多相关的案例以作参考。综上所述在技术层是可行的。

2.8.2 操作可行性

本系统操作简单，功能齐全，适用于追求健康合理饮食的人们。根据“人机交互”的可用性原则，我们可以得知系统的交互需要：易学习、高效率、高容错的特点。对于本系统而言，作为一款微信小程序，只要用户熟悉使用手机，即可操作此程序，并且产品的使用操作说明会在产品投入使用后提供。因此，本系统具备操作可行性。

2.8.3 法律可行性

本系统所选开发语言、算法皆为正规渠道合理获得使用。所用开发软件也是开源且经过验证的合法版本，SpringBoot 框架是根据 Apache License 协议开源的免费框架，MySQL 同样是一款开源数据库，不存在任何侵权或违反法律的内容，程序编译内容皆在法律允许范围内。因此本系统具备法律可行性。

2.8.4 经济可行性

本系统软件皆为开源免费软件，硬件设备只需一台电脑，本系统不需要额外开销，成本很低。若后续上线推广使用，可以开通会员制，升级上线付费内容，还可进行盈利。综上所述本系统在经济方面是可行的。

2.8.5 总结

综上，本系统具有技术可行性、操作可行性、法律可行性和经济可行性，无开发瓶颈，可以进行投入开发。

第3章 需求分析

系统需求分析部分概述膳食管理系统的功能需求，将系统进一步划分为模块，来详细的分析功能，并对核心业务进行分析，最后确定基于微信小程序的膳食营养管理系统的性能需求和运行需求。

3.1 总体目标

如今，随着人们对膳食营养健康关注度的提高，有关此类网络服务的需求不断增大。基于微信小程序的膳食营养管理系统是为了帮助用户合理安排膳食营养，依据国际膳食营养标准，根据用户身体状况合理推荐膳食营养安排。本系统的膳食营养管理指标为卡路里，考虑到虽然影响膳食营养的因素有很多，像碳水、蛋白质、微量元素等，但是从目前我国国民的身体情况出发，一些微量元素，蛋白质等在如此的社会发展条件下，并不是主要的关注点，现在更多的是以卡路里为标准，来管理推荐人们的膳食营养搭配。系统也会根据用户已摄入能量做后续的饮食推荐，合理调整膳食。系统推送当下人们关注、易懂的膳食营养贴士，实现每日关注健康饮食管理。

传统的膳食 app 要求系统配置，手机性能，而基于微信小程序开发，用户操作简单，功能性强，大大提高了用户的使用率。利用 Uniapp 框架开发，提供优美的 UI 界面，提高用户使用舒适度。

3.2 用户需求分析

3.2.1 用户角色需求分析

本系统用户主要分为两类，一类是用户，一类是管理员。

游客经注册后成为小程序用户，用户登录后可查看膳食营推送贴士，查看公告，修改密码等。用户可以输入个人信息，系统来推荐健康的膳食营养搭配；也可以在小程序中输入自己的饮食状况，系统计算后推荐后续可摄入营养，推荐接下来的饮食搭配。用户也可查看自己的饮食记录，以便每天查看监督。用户可对自己使用小程序的情况反馈给管理员，管理员视情况进行调整。

管理员拥有最高权限，可管理用户、公告、膳食信息，管理反馈意见，确保各类信息及时更新，系统及时得到维护。

3.2.2 系统用例设计图

用例图可以清楚的显示用户和相关用例的关系，清晰的展示系统的所有功能。由

以上分析可知，本系统主要有用户和管理员两种角色，这两张角色用例图如下：

1. 用户角色用例图

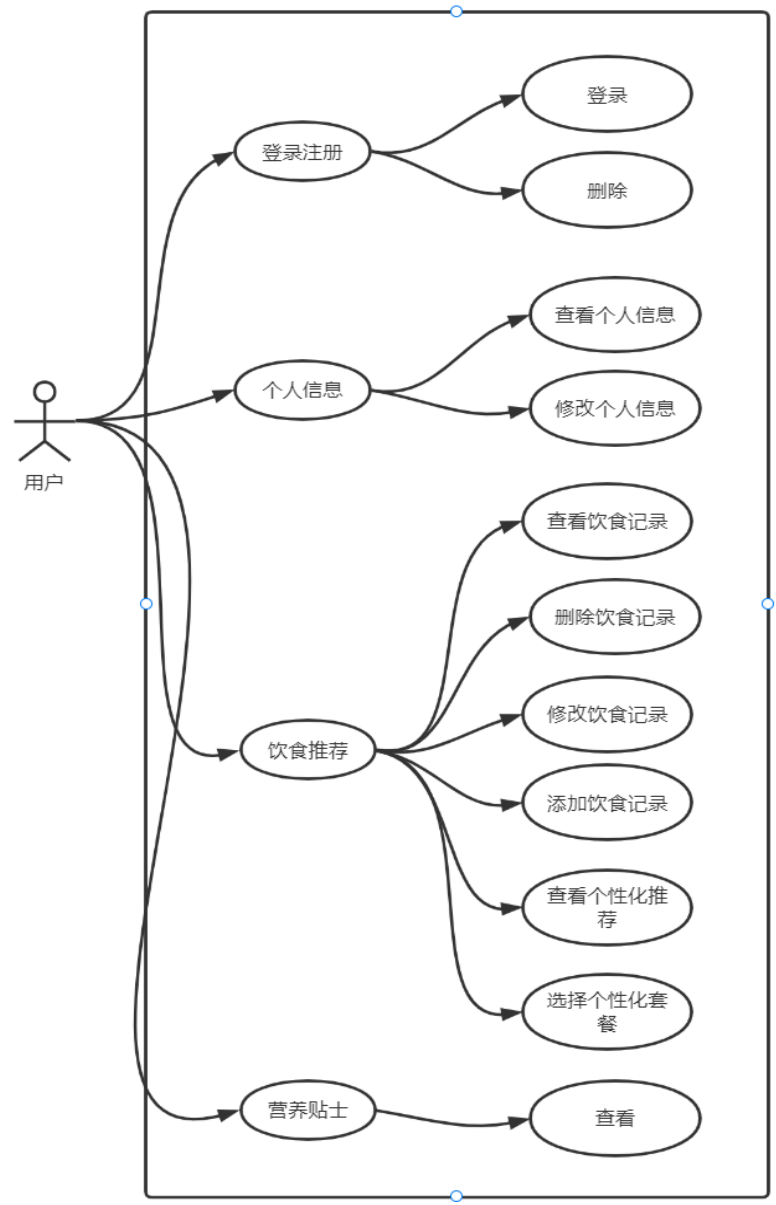


图 3-1 用户角色用例图

如图 3-1 所示，普通用户有注册、登录，个人信息管理以及个人饮食推荐管理、查看营养贴士等功能。用户注册登录后，输入自己个人及饮食信息，系统根据身体信息进行合理膳食营养推送，用户进行选择。

2. 管理员用例图

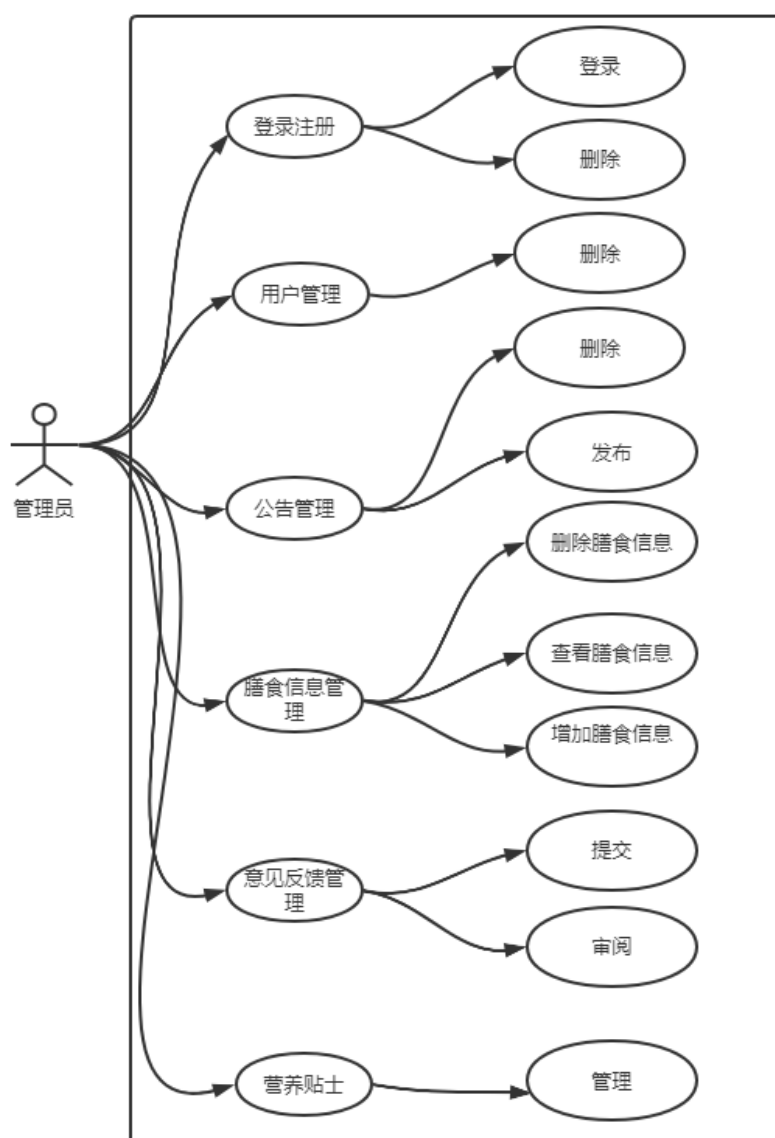


图 3-2 管理员用例图

如图 3-2 所示，管理员登录后可对用户进行管理，对公告进行发布和删除，根据国际《食物热量表》来维护膳食营养信息库，及时审阅意见反馈情况，管理膳食营养贴士信息等。

3.3 功能需求分析

3.3.1 总体设计类图

本系统旨在设计一个简洁便利的膳食营养管理微信小程序，现对其中涉及的功能服务以类进行为边界进行划分，以下会使用类图的形式来描述主要系统功能，主要功能分为：用户信息管理类、膳食管理类、饮食推荐类等。如图 3-3 为整体设计类图

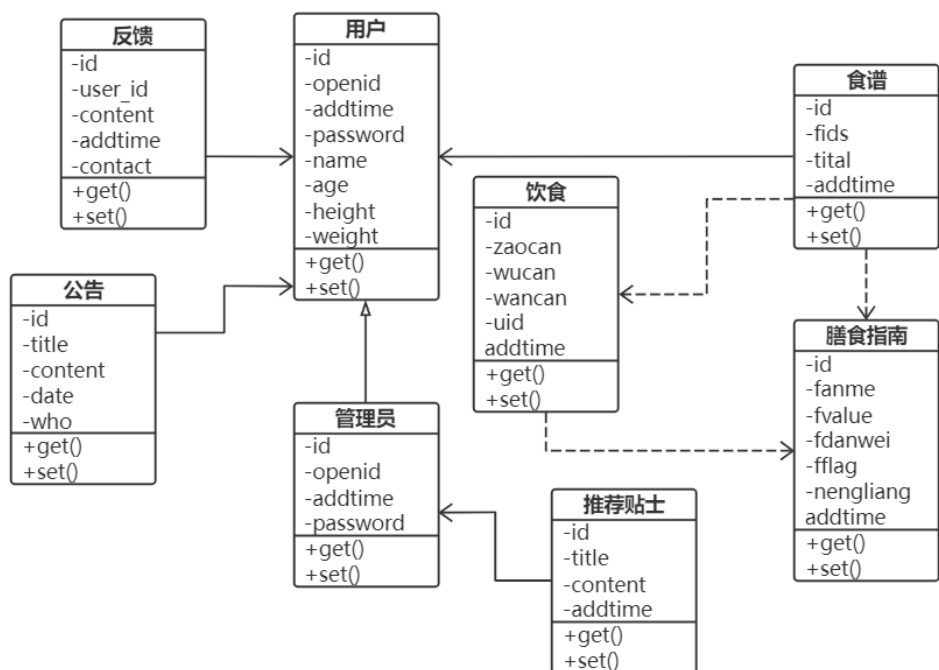


图 3-3 整体设计类图

3.3.2 用户信息管理

用户信息类参与实现了多个主要功能实现类，首先，用户开始使用小程序时，先进行注册、登录功能；系统根据用户状态码判断当前用户为普通用户还是管理员，管理员可根据用户 id 进行删除或启用用户的权限管理。用户注册、登录后可添加修改自己的年龄、身高、体重等个人信息，作为后续膳食推荐的重要依据。

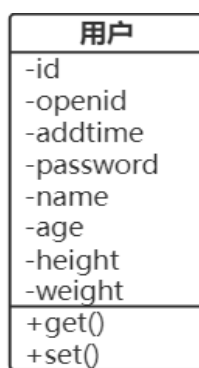


图 3-4 用户实体类图

用户实体类图如图 3-4 所示，描述了用户存储的实体信息。

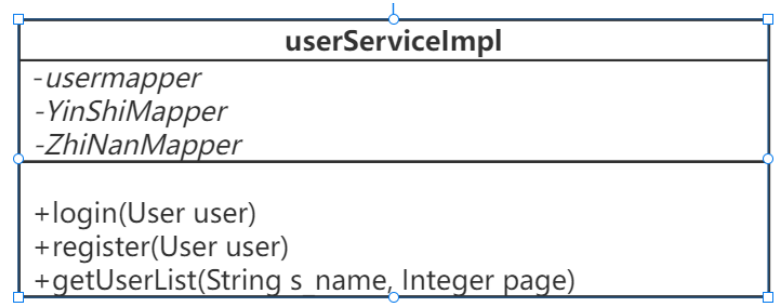


图 3-5 用户实现类图

用户实现类图如图 3-5 所示，描述了用户参与主要实现的服务接口：用户注册；用户登录；获取用户信息。

3.3.3 膳食管理

膳食管理的主要实体对象为管理员。管理员登录小程序后，根据国际《食物热量表》管理膳食模板，即管理不同食物的计量值、计量单位、能量等信息，作为后续饮食推荐的膳食模板，致力于推荐科学化、信息化的膳食套餐。此膳食管理类包含了对食物能量值的计算，当用户输入已摄取的食物后，程序会计算已摄入食物的能量值，同样为后续推荐做准备。



图 3-6 膳食指南类图

用户膳食指南类图如图 3-6 所示，描述了膳食指南实体类所包含的实体信息。

ZhiNanServiceImpl
-ShiPuMapper -YinShiMapper -ZhiNanMapper
+addZhiNanInfo(ZhiNan zhiNan) +getZhiNanList(String s_name, Integer userId, Integer page) +deleteZhiNan(Integer id) +sheru(Integer userId)

图 3-7 膳食指南实现类

用户膳食指南实现类如图 3-7 所示，描述了此实现类所实现的功能接口：添加指南、删除指南、查询指南信息、计算摄入量接口。

3.3.4 饮食推荐

用户进入小程序注册登录后，根据上述自身情况填写的个人信息，如身高体重等；系统根据不同用户的身体健康指数推荐食谱，此食谱的生成方式为：根据膳食模板中食物的能量值，以国际健康值为标准，推荐 3 种不同的食谱：减肥套餐、健康套餐、营养套餐。考虑到当前形势下蛋白质、微量元素等对人们的影响并不大，所以此食谱以目前人们最为关注的卡路里为推荐标准。并且系统会根据用户信息计算其理想体重，系统根据其理想体重和真实体重对比计算，首先选中推荐一种食谱，用户也可自行选择系统推送的套餐；用户也可以自选食物，程序计算能量值，用户参考食用；用户输入饮食信息，系统根据饮食信息中已摄入能量值，来为后续可摄入能量进行推送，若已超标，程序会推荐适量的运动。

食谱
-id -fids -tital -addtime +get() +set()

图 3-8 食谱类图

食谱类图如 3-8 所示，描述了食谱类所包含的实体信息。

ShiPuServiceImpl
-ShiPuMapper -ZhiNanMapper
+createShiPu() { +selectShiPu(String nengliang)

图 3-9 食谱实现类图

食谱实现类图如 3-9 所示，描述了食谱实现类所实现的功能接口，创建食谱和选择食谱接口。

3.3.5 公告管理

管理员登录小程序后可进行公告的增添删除等管理，用户进入小程序后可查看公告。

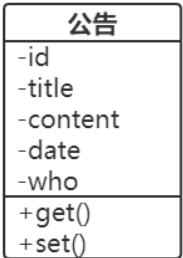


图 3-10 公告类图

公告管理类图如图 3-10 所示，描述了公告类所包含的实体信息。

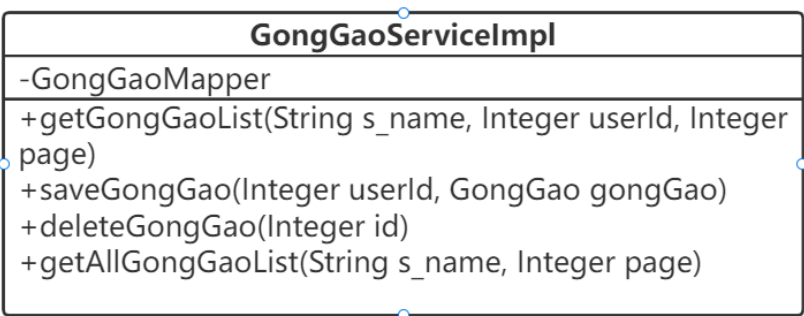


图 3-11 公告管理实现类图

公告管理实现类图如图 3-11 所示，公告实现类实现了公告的查询、添加、删除的功能，而未涉及公告修改功能，本系统考虑到公告具有严肃性和符合法律、道德要求的原则，为避免引起法务纠纷，提高公告的严肃性，所以没有设计公告的修改功能。

3.3.6 反馈管理

管理员和用户登录后都可进行反馈，但是管理员可查看所有用户反馈，用户只能查看自己的反馈列表。

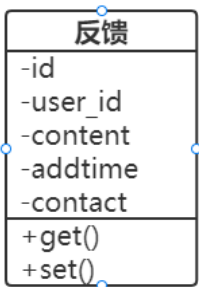


图 3-12 反馈实体类图

反馈实体类图如图 3-12 所示，描述了反馈类所包含的实体信息。

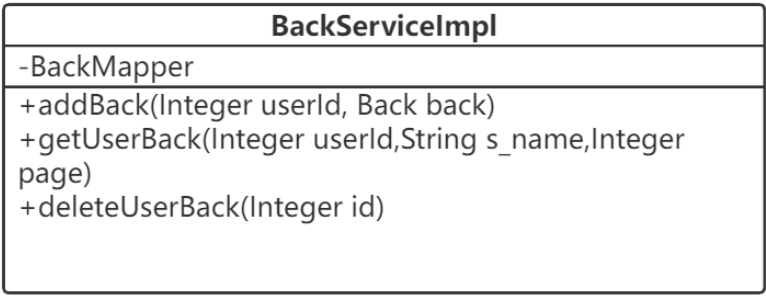


图 3-13 反馈实现类图

反馈实现类图如图 3-13 所示，管理员、用户登录后都可进行反馈，管理员可以查看反馈列表。

3.3.7 推荐贴士管理

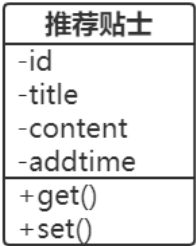


图 3-14 推荐贴士类图

推荐贴士类图如图 3-14 所示，描述了推荐类的实体信息。

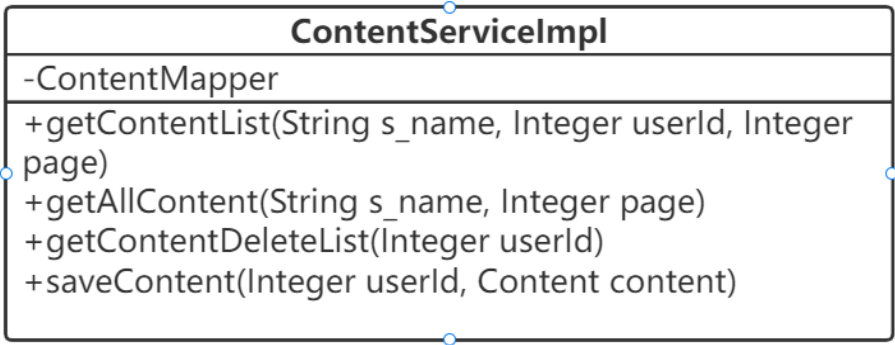


图 3-15 推荐贴士实现类图

推荐贴士实现类图如图 3-15 所示，实现类推荐查找和保存类。管理员登录后可对推荐信息进行发布增添，用户和管理员在首页都可查看、搜索感兴趣的推荐信息。

3.4 数据分析

3.4.1 数据流图

数据流图以图的形式表示系统逻辑。展示了系统的逻辑功能、系统内部数据的逻辑流向和逻辑变换的过程。

1. 顶层数据流分析

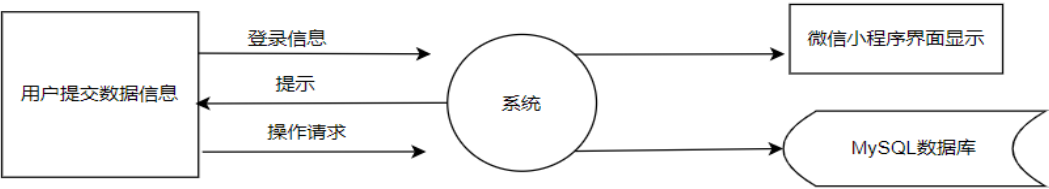


图 3-16 系统数据流图顶层图

如图 3-16 系统数据流图顶层图所示，此图为本系统的顶层数据流图，当用户向系统提交数据信息后，系统将提示信息返回，并将传入数据进行对比和处理，将数据信息写入数据库，完成系统数据流动。

2. 一层数据流分析

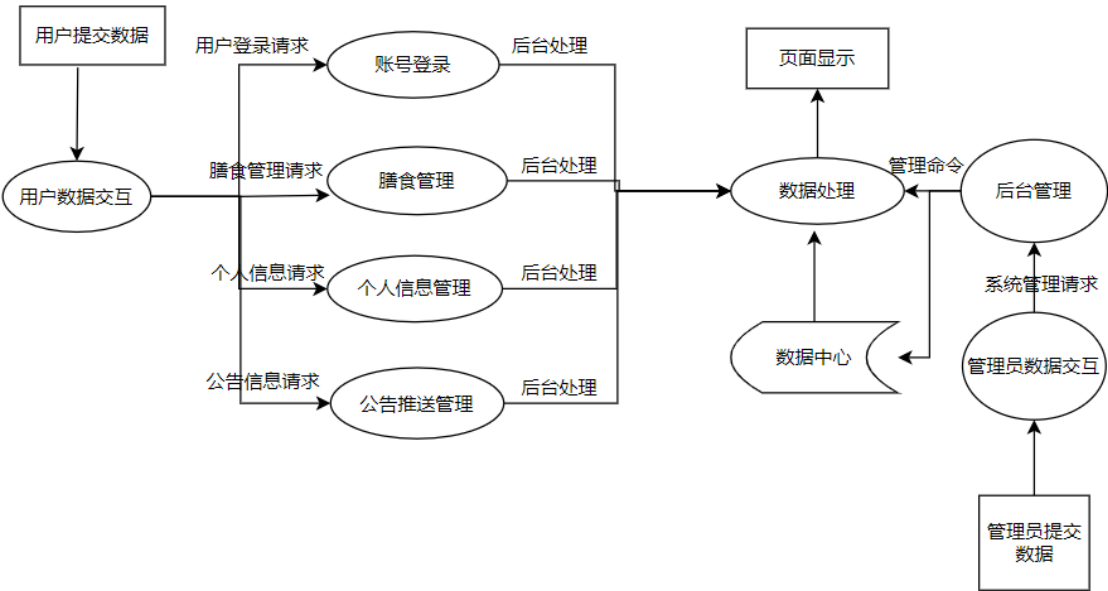


图 3-17 系统数据流图一层图

如图 3-17 为系统一层数据流图，用户提交数请求会分为用户登录请求、膳食管理请求、个人信息请求、公告信息请求几个部分。这几个部分请求会分别有对应的服务来处理，处理后保存到数据中心。管理员数据请求会在后台管理与数据中心直接交互。

3.4.2 数据字典

在对系统的整体框架进行构建时，数据字典是指对数据的数据项、数据结构、数据流、数据存储、处理逻辑、外部实体等进行定义和描述，数据字典如下所示：

1. 数据结构：keep-user

含义说明：用户是业务进行的载体，本数据结构记录了用户的基本数据信息。

表 3-1 keep-user 数据字典

数据项	含义说明
id	识别用户的唯一标识
openid	用户注册时所用昵称
password	用户设置的登录密码
addtime	用户注册时间
is_admin	用户和管理员的身份标识
status	用于标记该用户是否启用
name	记录用户姓名
age	用于记录用户年龄
height	用于记录用户身高
weight	用于记录用户体重
oddpasword	用于记录用户修改之前的密码

2. 数据结构：keep-yinshi

含义说明：用户记录饮食信息表，本数据结构记录了饮食的基本数据信息。

表 3-2 keep-yinshi 数据字典

数据项	含义说明
id	识别饮食的唯一标识
uid	联用户 id
zaocan	用于早餐食用信息
wucan	用户午餐食用信息
wancan	用户晚餐食用信息
count1	用户早餐食用量
count2	用户午餐食用量
count3	用户晚餐食用量
addtime	用于记录时间

3. 数据结构：keep-zhinan

含义说明：膳食营养信息表，本数据结构记录了每种食物的膳食营养信息的基本

数据。

表 3-3 keep-zhinan 数据字典

数据项	含义说明
fid	识别膳食的唯一标识
fname	膳食名称
fvlue	膳食计量值
fdanwei	膳食计量单位
nengliang	膳食卡路里数
fflag	膳食信息是否删除，0 为未删除，1 为已删除

4. 数据结构：keep-gonggao

含义说明：公告信息表，本数据结构记录了公告信息的基本数据。

表 3-4 keep-gonggao 数据字典

数据项	含义说明
id	识别公告的唯一标识
title	公告主题
content	公告内容
date	公告发布时间

5. 数据结构：keep-content

含义说明：膳食推荐信息表，本数据结构记录了膳食推荐的基本数据。

表 3-5 keep-content 数据字典

数据项	含义说明
id	识别推荐的唯一标识
title	推荐标题
content	推荐内容
addtime	推荐添加日期
is_top	推荐是否置顶

6. 数据结构：keep-back

含义说明：意见反馈表，本数据结构记录了用户反馈信息的基本数据。

表 3-6 keep-back 数据字典

数据项	含义说明
id	反馈的唯一标识
user_id	反馈该条意见的用户 id
content	具体反馈内容

addtime	反馈提交时间
contact	反馈用户手机号

7. 数据结构：keep-shipu
- 含义说明：食谱生成表，本书结构记录了用户食谱推荐组合的基本数据。

表 3-7 keep-shipu 数据字典

数据项	含义说明
id	食谱的唯一标识
fid	关联膳食表，表示组成食谱的膳食信息组合情况
total	该项组合食谱总能量值
addtime	食谱的生成时间

3.4.3 数据库 E-R 图

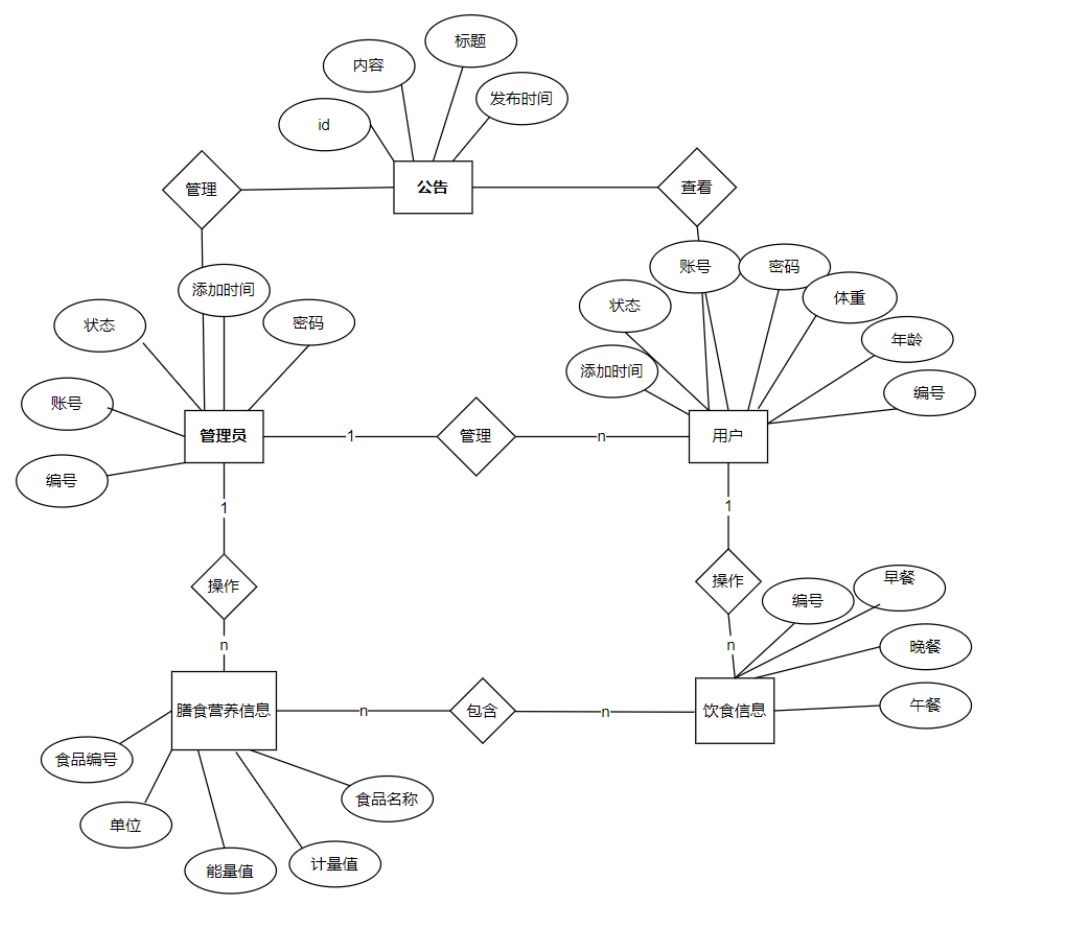


图 3-18 数据库 E-R 图

3.5 性能需求分析

3.5.1 性能分析

1. 简洁性

基于微信小程序的膳食营养管理系统，主要意图之一就是简化程序页面，不仅利用小程序开发的方便、简洁性，我们所设计的程序更要简单易懂，才能更加符合大众审美，利于满足广大用户的需求。

2. 可靠性

用户在使用小程序时其中涉及用户身份信息以及饮食信息，要求数据的存储十分可靠，多次根据实际情况，减少修改权限许可。也可在此通过 MySQL 数据库进行物理备份，而且备份存储的地方进行定制化选择，确保备份信息长期且无误存储，维持系统稳定。

3. 响应性

由于系统提供多种功能，不同功能的处理逻辑也有很大不同，响应时间也不同，甚至同一功能在不同输入数据的响应时间也不同，此系统基于微信小程序开发，要求简单便捷，响应时间理应缩短。可采用业务拆分，JVM 配置优化，SQL 查询优化等方式，缩短响应时间。

4. 易扩展性

关于软件系统的可扩展性，遵循开闭原则，对扩展开放对修改关闭。对本系统最核心的关于膳食营养来推荐饮食功能，可采用垂直扩展设计理念，增加相同膳食信息所对应的饮食推荐搭配，提高信息复用率，加快对需求变更的响应速度。本系统膳食推荐标准以卡路里为主要依据，减少当下对人们影响小并且不太关注的领域的开发，提高开发效率，并且所有膳食信息严格按照国际《膳食营养表》进行添加和管理，提高程序的科学性和易扩展性。

3.5.2 运行需求

系统运行需求如表 3-8 所示：

表 3-8 运行需求表

软件编号	软件名称	具体配置
1	客户端	微信开发者工具
2	语言环境	Java
3	框架	SpringBoot、Uniapp
4	数据库	MySQL5.5
5	开发工具	IDEA

第4章 系统分析与设计

基于微信小程序的膳食营养管理系统的设计阶段，首先从技术架构设计开始，从技术方面探索此系统。按照需求分析确定此系统的功能架构，给出动态建模和静态建模设计，为下面的详细设计做铺垫，最后确定数据库设计。

4.1 概要设计

4.1.1 体系结构设计

从需求分析的阐述中可以得知，基于微信小程序的膳食营养管理系统主要有用户和管理员两个角色，与之相关的核心业务是膳食管理和饮食推荐业务，为支撑这两个业务模块，将此系统细分为如下几个模块：

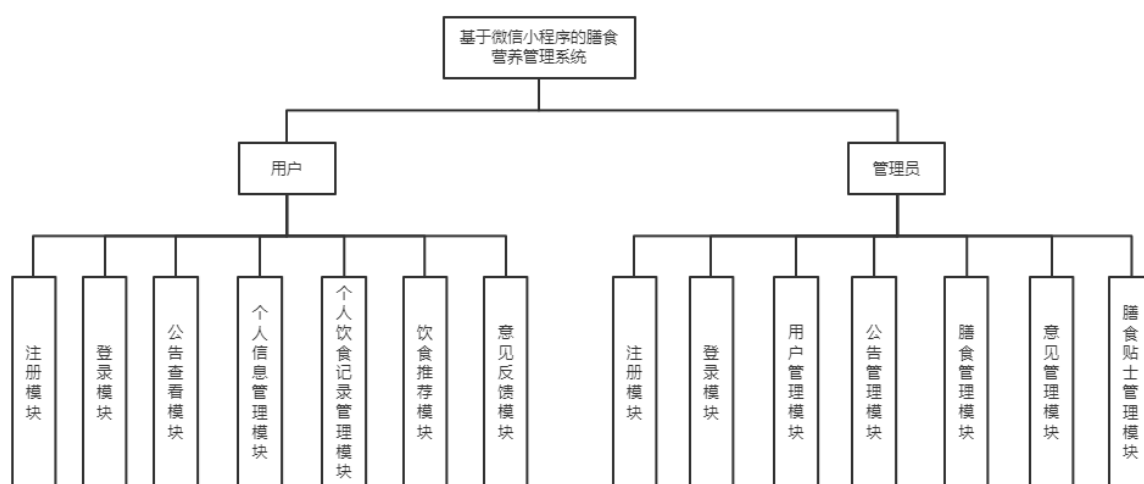


图 4-1 总体功能模块图

如图 4-1 所示，系统以用户和管理员两种角色组成。

1. 用户角色具体功能模块描述如下：

- (1) 注册模块
- (2) 登录模块
- (3) 公告查看模块
- (4) 膳食推荐查看模块
- (5) 个人信息管理模块
- (6) 个人饮食记录管理模块
- (7) 饮食推荐模块

(8) 意见反馈模块

2. 管理员角色具体功能模块描述如下：

- (1) 注册模块
- (2) 登录模块
- (3) 用户管理模块
- (4) 膳食贴士管理模块
- (5) 膳食管理模块
- (6) 公告管理模块
- (7) 意见管理模块
- (8) 修改用户密码模块

4.1.2 功能流程

由于本系统涉及两种角色：管理员和用户，程序流程有交叉，因此列举如下四种功能流程。

1. 注册流程

用户进入程序后可以先进行用户注册，程序接收前端输入的注册信息，由系统判断数据是否符合要求，进而查询数据库检索是否有重复名称占用，符合以上要求则注册成功，否则即为失败。注册流程图如图 4-2 所示。

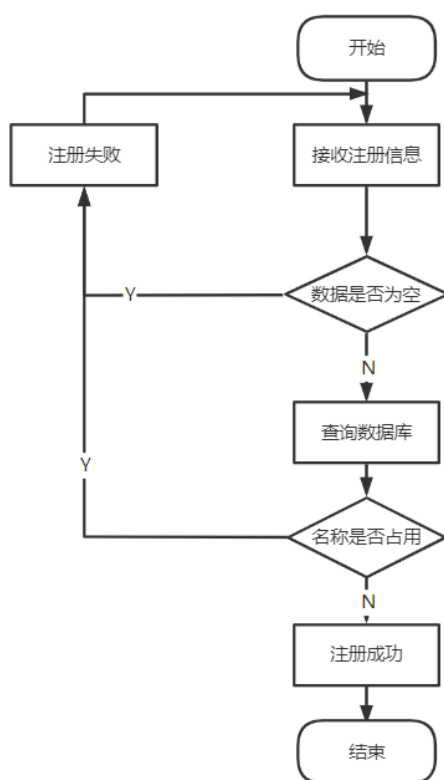


图 4-2 用户注册流程图

2. 登录流程

用户注册成功后，进行账号密码登录，系统和数据库信息进行对比，并且用 token 进行校验，采用 JWT 加密，同时加盐处理，确保每个用户有且仅有唯一的 token 值进行校验，认证成功，通过拦截器则登录成功，否则登录失败。用户登录流程图如图 4-3 所示。

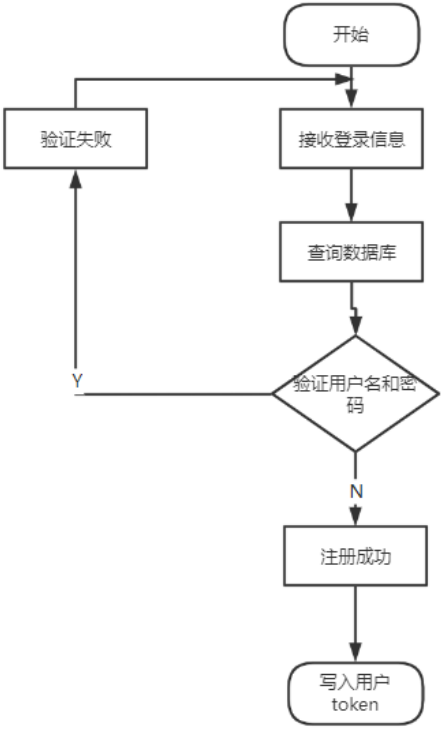


图 4-3 用户登录流程图

3. 膳食管理流程

在膳食管理过程中，管理员首先会按照国际《营养能量表》对食品膳食信息进行增添，严格按照国家食品安全规格进行补充或删除，为保证信息的严谨性，不支持修改功能。此后用户可根据实际情况填写自己食用食物信息，系统进行对已摄取能量的计算，为后面食谱推荐做准备。膳食管理流程如图 4-4 所示：

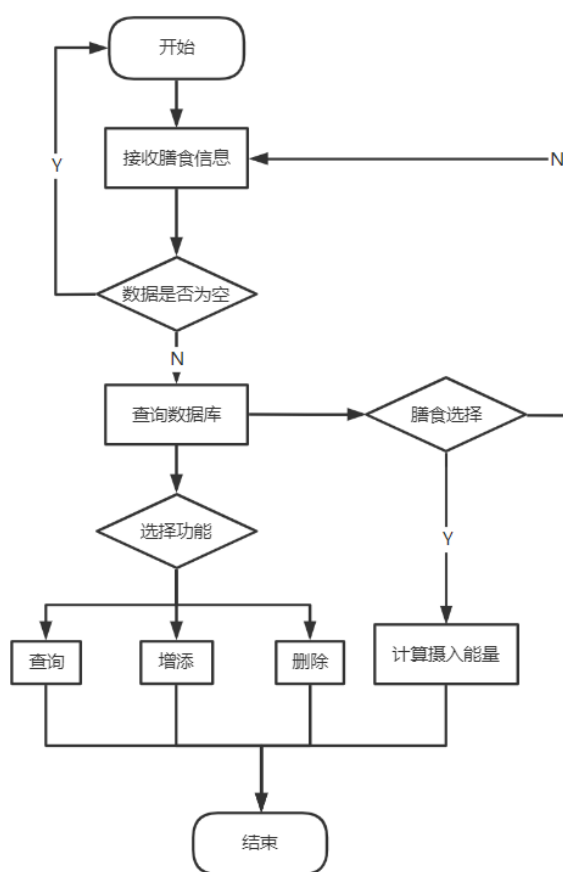


图 4-4 膳食管理流程图

4. 饮食推荐流程

饮食推荐过程中，系统首先根据用户输入的个人信息进行 BMI（身体质量指数）的计算，计算公式为 $BMI = \frac{\text{体重}}{\text{身高}^2}$ ，并推算用户标准体重，公式为：标准体重 = $(\text{身高} - 80) \times 0.7$ 。根据标准人体摄入营养能量进行计算系统根据标准体重判断当前用户的真实体重，若真实体重超过标准体重的 10% 则推荐减肥套餐，若真实体重小于标准体重的 10% 则推荐营养套餐，否则推荐健康套餐。本系统膳食营养推荐以卡路里为标准，对当前社会形势下对人体影响不大的蛋白质、微量元素等膳食指标不做过多考虑，更多的关注人们目前关注的且影响力更大的卡路里来衡量整个膳食推荐食谱，严格按照《膳食热量表》进行管理。若已摄入能量，即用户输入已用食物和食用量，系统则需按国家标准能量来减去已摄入能量，进行后续饮食的推荐，用户继而查看选择。用户若对推荐食谱不满，也可自行选择食物，系统计算后，自行参考选择。饮食推荐流程图如图 4-5 所示：

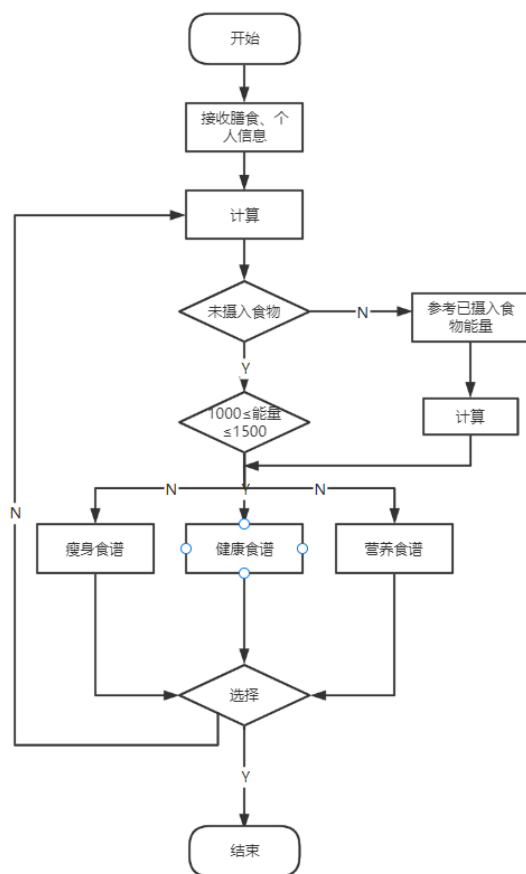


图 4-5 饮食推荐流程图

4.2 数据设计

数据库是对需求的直观反映和表达。因此，设计时必须真正满足用户的需求，需要与用户进行多次沟通，细化需求，并且每次变更，在数据库设计中也要逐一对应变更。本章主要论述系统概要设计与详细设计，从功能到数据详细到代码去论述系统的各个方面。

4.2.1 数据库逻辑结构设计

通过对基于微信小程序的膳食营养管理的功能模块和实体整体进行分析，可以得出数据库逻辑结构设计。

1. 实体属性图

系统可以分为以下实体：

- (1) 用户

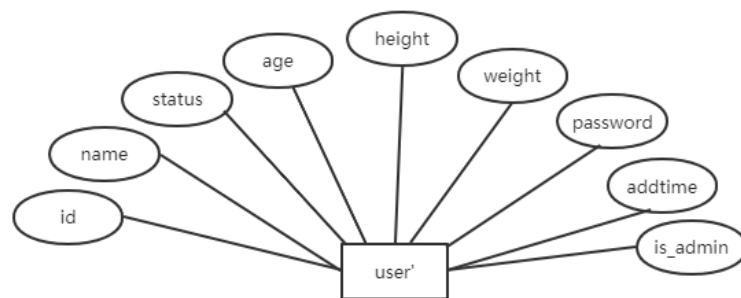


图 4-6 用户实体属性图

用户实体属性图如图 4-6 所示，用户是系统业务实现一个主要的实体。用户实体属性包括：用户 id，用户姓名、用户年龄、用户体重、用户状态、密码、注册时间。

(2) 饮食

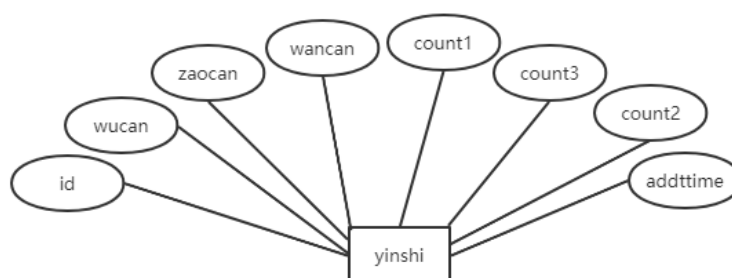


图 4-7 饮食实体属性图

饮食实体属性图如图 4-7 所示，用户饮食业务实体包括：饮食 id，早餐、午餐、晚餐、早餐食用量、午餐食用量、晚餐食用量和添加时间。

(3) 膳食

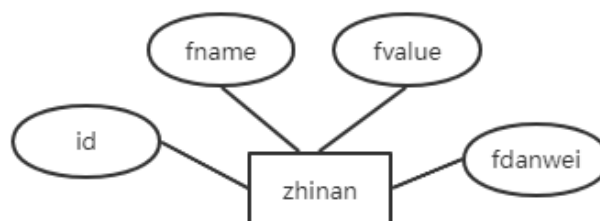


图 4-8 膳食指南实体属性图

膳食指南属性图如图 4-8 所示，膳食指南业务实体包括：膳食 id、膳食名称、膳食计量值、膳食计量单位、能量值。

(4) 公告

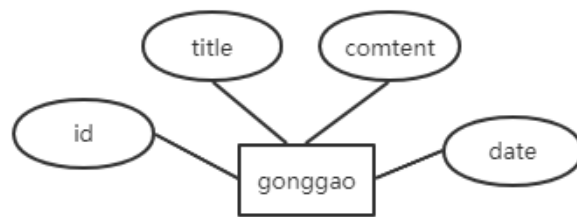


图 4-9 公告实体属性图

实体属性图如 4-9 所示，公告实体包括：公告 id、公告标题、公告内容、公告添加日期。

(5) 推荐

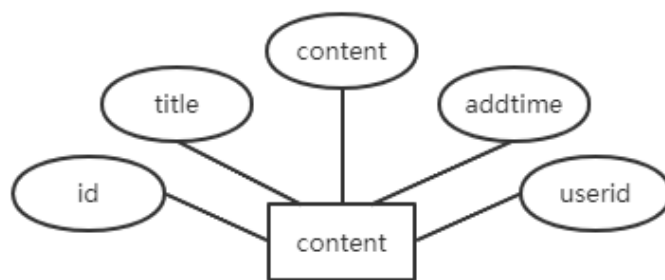


图 4-10 推荐实体属性图

推荐实体图如 4-10 所示，包括：推荐 id、推荐标题、推荐内容、推荐时间。

(6) 食谱

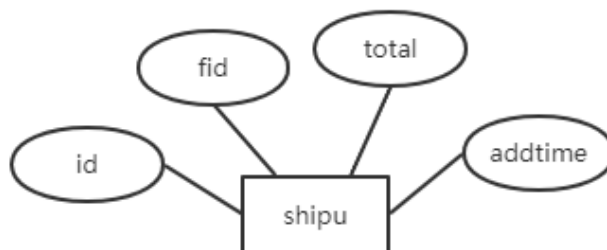


图 4-11 食谱实体属性图

食谱实体如图 4-11 所示，包括：食谱 id、膳食 fid、食谱总能量、添加时间。

4.2.2 数据库表设计

本系统使用 MySQL 数据库，数据库名为 shanshi，数据库表结构如表 4-1 用户互信息表至 4-7 食谱信息表。

1. 用户信息表

用户信息表记录了用户的基本数据信息。结构如表 4-1 所示：

表 4-1 用户信息表

属性名	存储名	类型	长度	是否为空	备注
用户 id	Id	Int	11	No	Null
用户名	name	varchar	255	No	Null
密码	Password	varchar	255	No	Null
年龄	age	varchar	30	Yes	Null
体重	Weight	varchar	255	Yes	Null
状态	State	Int	1	No	Null
身高	Height	varchar	255	Yes	Null
添加时间	Addtime	Datetime	100	No	Null

2. 饮食信息表

饮食信息表记录了饮食的基本数据信息。

表 4-2 饮食信息表

属性名	存储名	类型	长度	是否为空	备注
id	Id	Int	11	No	Null
早餐	Caocan	varchar	255	Yes	Null
午餐	wucan	varchar	255	Yes	Null
晚餐	Wancan	varchar	255	Yes	Null
早餐食用量	Count1	Int	11	Yes	Null
午餐食用量	Count2	Int	11	Yes	Null
晚餐食用量	Count3	Int	11	Yes	Null

3. 膳食指南信息表

膳食指南信息表记录了每种食物的膳食营养信息的基本数据。

表 4-3 膳食指南信息表

属性名	存储名	类型	长度	是否为空	备注
膳食 id	Id	Int	11	No	Null
膳食名称	fname	varchar	255	No	Null
膳食营养价值	fvalue	Int	11	Yes	Null
膳食计量单位	Fdanwei	varchar	255	Yes	Null

4. 公告信息表

公告信息表记录了公告信息的基本数据。

表 4-4 公告信息表

属性名	存储名	类型	长度	是否为空	备注
公告 id	Id	Int	10	No	Null
公告主题	Title	varchar	50	Yes	Null
公告内容	Content	varchar	20	Yes	Null
公告发布日期	Date	varchar	20	Yes	Null

5. 膳食贴士信息表

膳食贴士信息表记录推荐的膳食贴士信息的基本数据。

表 4-5 膳食贴士信息表

属性名	存储名	类型	长度	是否为空	备注
膳食贴士 id	Id	Int	11	No	Null
用户 id	User_id	varchar	11	Yes	Null
推荐主题	Title	varchar	100	No	Null
推荐内容	Content	varchar	255	Yes	Null

6. 反馈信息表

反馈信息表记录用户和管理员反馈信息的基本数据。

表 4-6 反馈信息表

属性名	存储名	类型	长度	是否为空	备注
反馈信息 id	Id	Int	11	No	Null
用户 id	User_id	varchar	11	Yes	Null
反馈内容	Content	varchar	100	No	Null
反馈时间	Addtime	varchar	100	Yes	Null
联系方式	Contact	varchar	100	Yes	Null

7. 食谱信息表

食谱信息表记录了生成食谱信息的基本数据。

表 4-7 食谱信息表

属性名	存储名	类型	长度	是否为空	备注
食谱信息 id	Id	Int	11	No	Null
膳食 id	Fids	varchar	255	Yes	Null
食谱总能量	total	varchar	255	Yes	Null
反馈时间	Addtime	varchar	100	Yes	Null

4.3 详细设计

根据以上对系统的需求设计，我们初步了解了本系统的架构、功能、开发方向，以下将是对本系统进一步的详细设计过程以及已完成功能的情况展示。

4.3.1 用户注册模块设计

游客进入小程序后，若未曾注册账号，可先行进行注册。



图 4-12 用户注册界面图

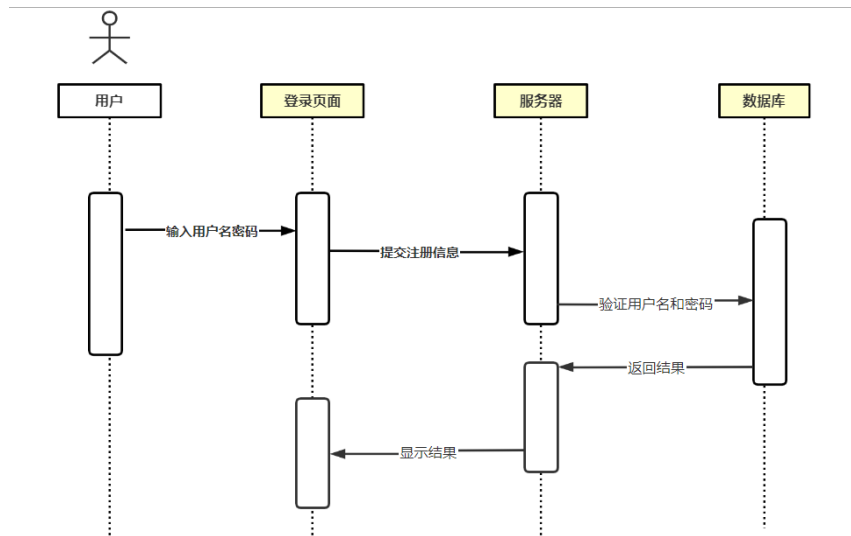


图 4-13 用户注册时序图

核心代码如下：

```

view class="login-input" :class="{active:distinguish.PassWord===true}">
  <image src="../../static/yx-login/password.png"></image>
  <input type="password" placeholder="请输入您要注册的密码"
placeholder-style="color:#fff" v-model="From.PassWord" v-if="SwiTch"/>
  <input type="text" placeholder="输入您要注册的密码" placeholder-
style="color:#fff" v-model="From.PassWord" v-else/>
  <image src="../../static/yx-login/delete.png" class="delete" v-
if="From.PassWord" @click="From.PassWord=null" style="padding-
right:10px;width:22px;"></image>
  <image src="../../static/yx-login/yanjing.png" class="switch"
@click="Switch()"></image>
</view>
<text class="tips" v-if="PassWordtips">{{PassWordtips}}</text>
</view>
</view>
<view class="login-button">
  <button @click="register">注册</button>
</view>

```

4.3.2 用户登录模块设计

运行该平台时，首先进入登录界面，进行登录验证。本系统采用 token 验证方式：采用 JWT 加密，同时进行加盐。统一的权限验证工具，防止用户直接通过 URL 进行侵入系统，拦截器会根据当前用户生成的 token 判断当前用户是否已经登录。如果没有登录提示您还没有登录信息，所有请求都会先通过此 token 拦截器，确保系统安

全性。



图 4-14 用户登录界面图

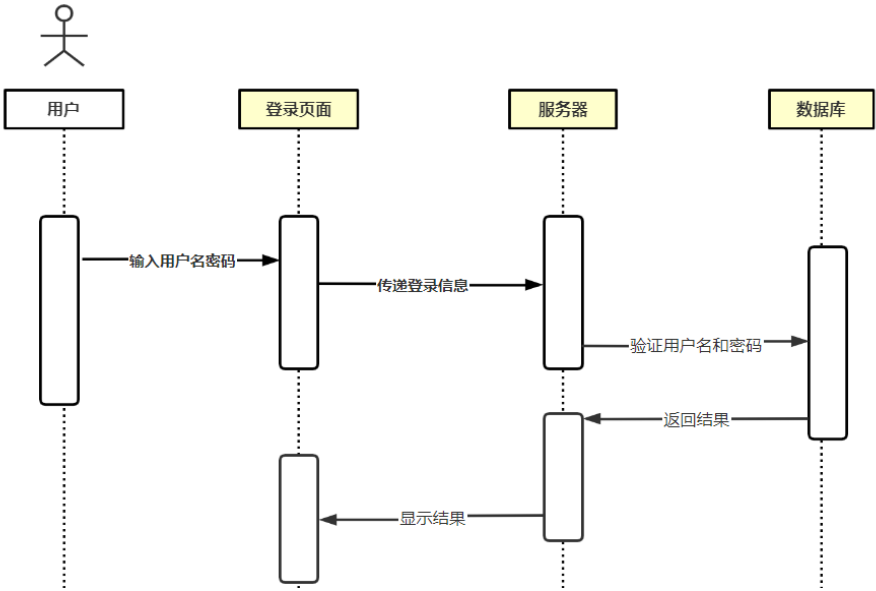


图 4-15 用户登录时序图

核心代码如下：

```
public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
Object handler) throws Exception {
    String requestURI = request.getRequestURI();
```

```

// 注册和登录接口直接返回 true
if (requestURI.equals("/api/user/register") ||
requestURI.equals("/api/user/login")) {
    return true;
}

//跨域请求会首先发一个 option 请求，直接返回正常状态并通过拦截器
if (request.getMethod().equals("OPTIONS")) {
    response.setStatus(HttpServletResponse.SC_OK);
    return true;
}
response.setCharacterEncoding("utf-8");
String token = request.getHeader("token");
if (token != null) {
    // token 校验
    boolean result = TokenUtil.verify(token);
    if (result) {
        System.out.println("通过拦截器");
        return true;
    }
}
response.setContentType("application/json; charset=utf-8");
try {
    JSONObject json = new JSONObject();
    json.put("msg", "您还没有登录");
    json.put("code", "401");
    response.getWriter().append(json.toString());
    System.out.println("认证失败，未通过拦截器");
} catch (Exception e) {
    return false;
}
}

```

4.3.3 用户管理模块

管理员登录后可对注册的用户进行管理。本系统基于微信小程序的开发理念和对于安全性的考虑，本系统只提供管理员是否启用用户的管理权限，更好的维护用户的私人信息。管理员删除用户后，数据库并不会删除用户的所有信息，其信息仍保存在数据库中，减少数据的流动。



图 4-16 用户管理界面图

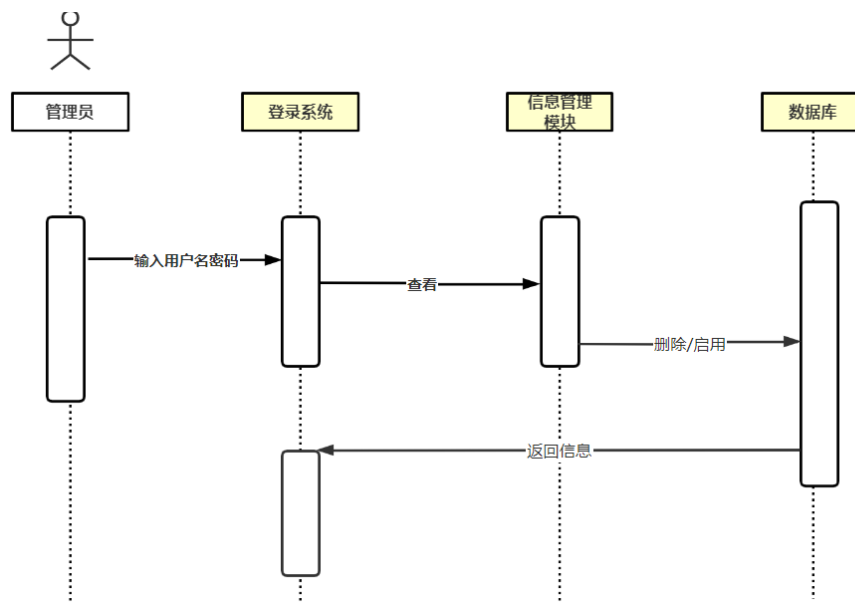


图 4-17 用户管理时序图

核心代码：

```

<view class="m-card-list">
  <view
    v-for="item in userList"
  >

```



```

:key="item.id"
:class="['m-card', item.isTop === 1 ? 'm-top' : '']"
>
<view class="m-date"
  ><text class="cuIcon-titles text-orange"></text>创建时间: {{
    item.addtime.slice(0, 10)
  }}</view>
>
<view class="m-title">用户名: {{ item.openid }}</view>
<view class="m-content" v-if="item.isAdmin == 0"> 角色: 普通用户
</view>
<view class="m-content" v-if="item.isAdmin == 1"> 角色: 管理员 </view>
<view class="m-content" v-if="item.status == 0">
  账户状态: 不可用
</view>
<view class="m-content" v-if="item.status == 1"> 账户状态: 可用 </view>
<view
  class="m-content"
  v-if="item.status == 1"
  @click="updateStatus(item.id, -1)"
>
  <button style="color: #ed1c24">删除</button>
</view>
<view
  class="m-content"
  v-if="item.status == 0"
  @click="updateStatus(item.id, 1)"
>
  <button style="color: #39b54a">启用</button>
</view>
</view>
</view>
</baseTabar></baseTabar>
</view>
</template>

```

4.3.4 饮食推荐贴士模块

管理员管理增添小程序首页中的膳食推荐小贴士,让人们更直观的感受关注健康

饮食给大家带来的影响。

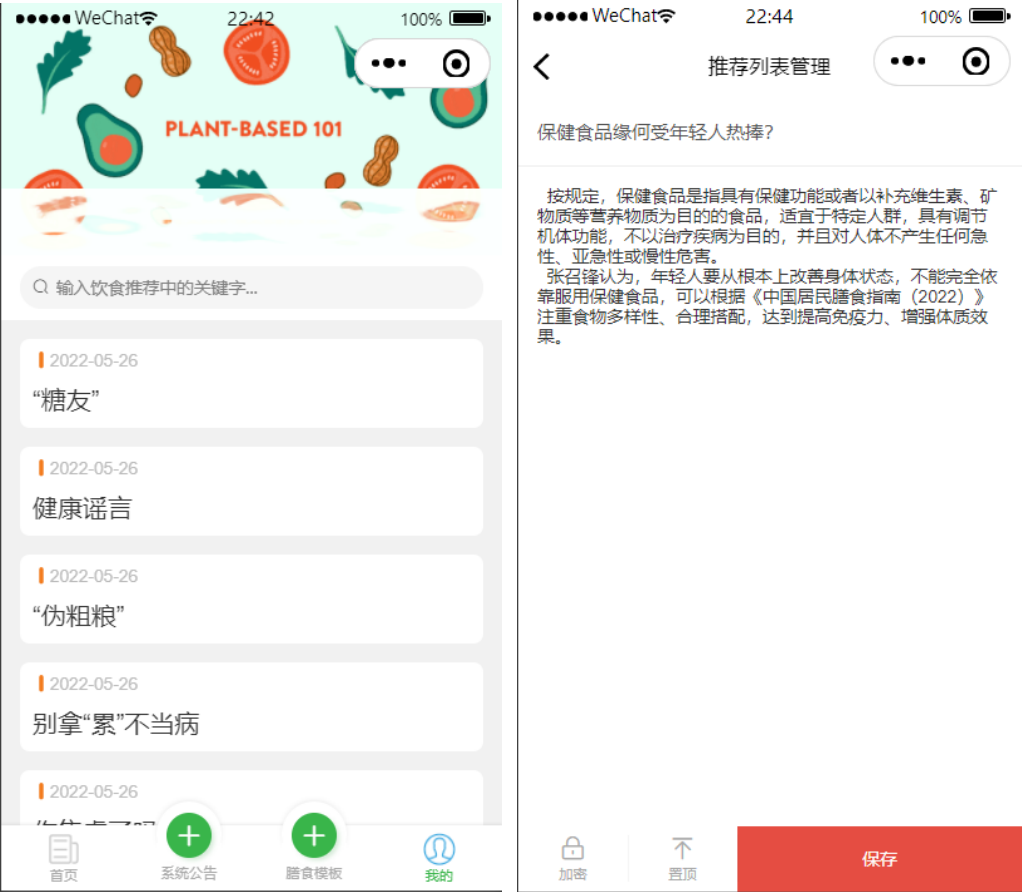


图 4-18 饮食推荐贴士界面图

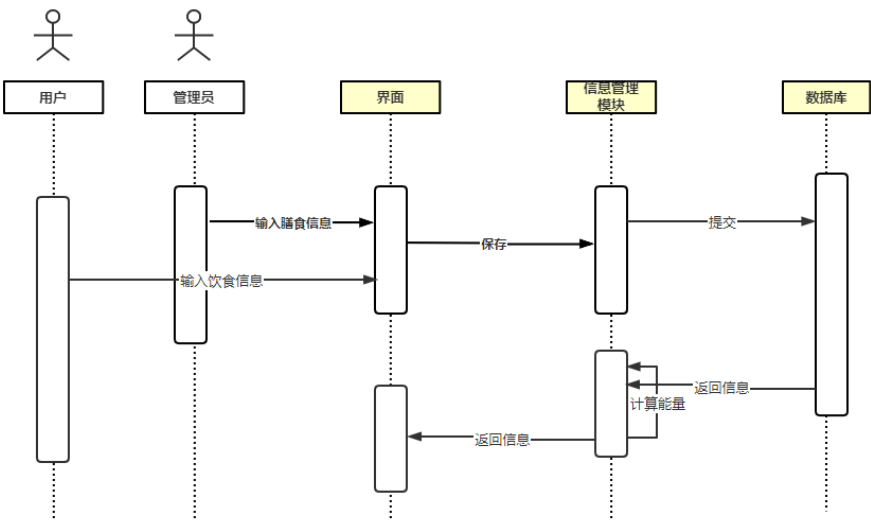


图 4-19 饮食推荐时序图

核心代码如下：

```
@ApiOperation(value = "查询饮食推荐贴士")
@GetMapping("/list_del")
```

```

public JsonResult getContentDeleteList() {
    Integer userId = GetLoginUser.getLoginUser();
    return contentService.getContentDeleteList(userId);
}

@ApiOperation(value = "保存/修改")
@PostMapping("/save")
public JsonResult saveContent(@RequestBody Content content) {
    Integer userId = GetLoginUser.getLoginUser();
    return contentService.saveContent(userId,content);
}

@ApiOperation(value = "根据 ID 查找备忘录")
@GetMapping("/by/{id}")
public JsonResult getById(@PathVariable(value = "id") int id) {
    return JsonResult.succ(contentService.getById(id));
}

@ApiOperation(value = "删除，并非真正删除")
@DeleteMapping("/del/{id}")
public JsonResult deleteUserContent(@PathVariable(value = "id") Integer id) {
    return contentService.deleteUserContent(id);
}

```

4.3.5 公告管理模块

管理员点击系统公告按钮后可以编写 公告主题和公告内容，本系统综合考虑法律可行性和实际运用情况，公告时常涉及系统内部或对外宣称的符合法律与道德约束的内容，有可能引起法律纠纷。所以本系统不涉及管理员的修改公告权限，允许增添和删除，并且删除后，数据库数据也将删除，避免法律纠纷。

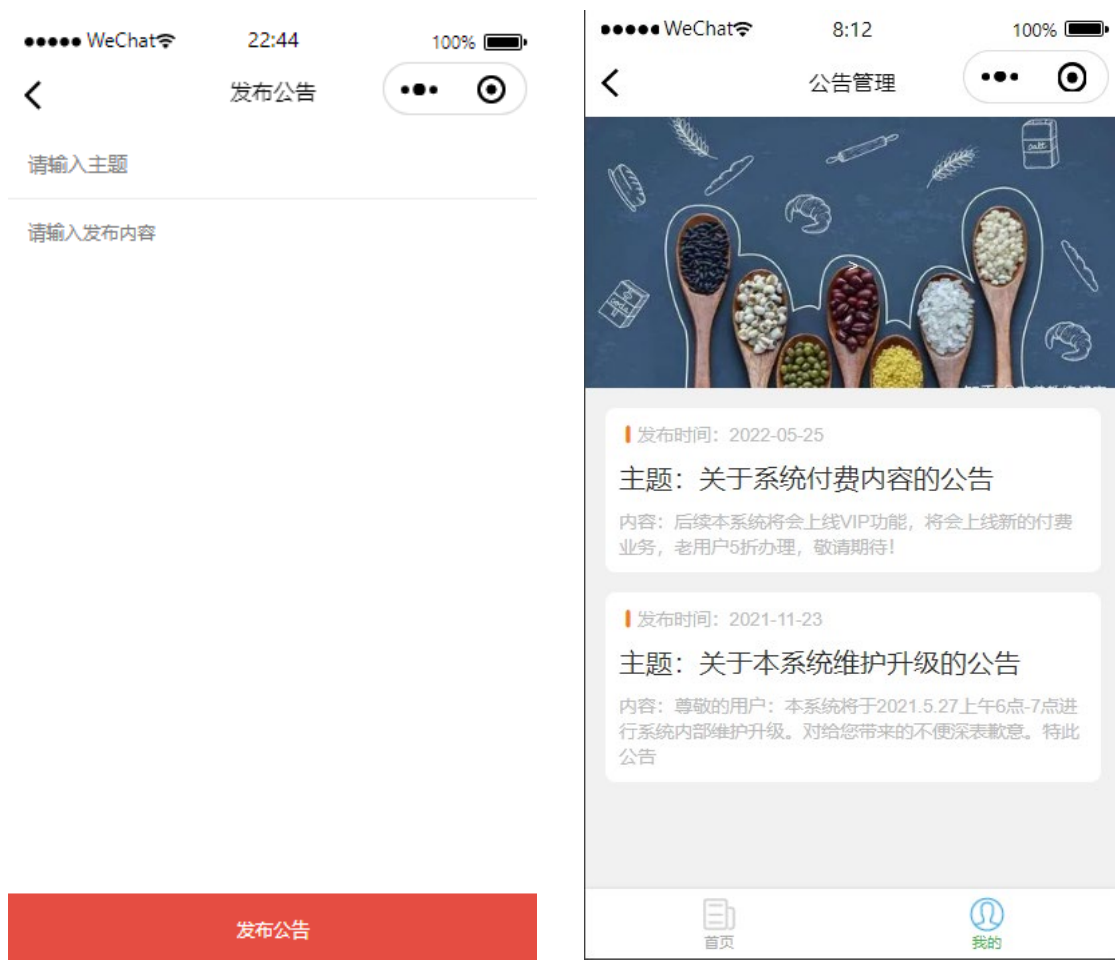


图 4-20 管理员管理系统公告界面图

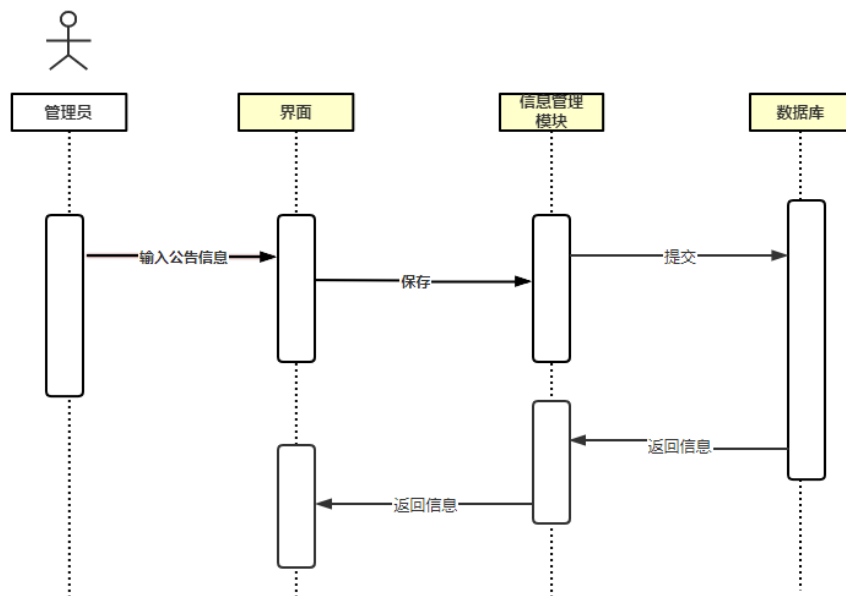


图 4-21 管理公告时序图

核心代码如下：

```
@ApiOperation(value = "分页查询公告")
@GetMapping("/list")
public JsonResult getContentList(String s_name,Integer
isAdmin,@RequestParam(defaultValue = "1") Integer page) {
    Integer userId = GetLoginUser.getLoginUser();
    if(isAdmin == 1) {
        return gongGaoService.getGongGaoList(s_name,userId,page);
    }else {
        //获取所有公告
        return gongGaoService.getAllGongGaoList(s_name,page);
    }
}

@ApiOperation(value = "保存/修改公告信息")
@PostMapping("/save")
public JsonResult saveGongGao(@RequestBody GongGao gongGao) {
    Integer userId = GetLoginUser.getLoginUser();
    return gongGaoService.saveGongGao(userId,gongGao);
}

@ApiOperation(value = "删除公告")
@DeleteMapping("/del/{id}")
public JsonResult deleteGongGao(@PathVariable(value = "id") Integer id) {
    return gongGaoService.deleteGongGao(id);
}
```

4.3.6 膳食管理模块

管理员登录后根据国际《营养能量表》添加管理膳食信息，维护膳食名称、计量值、单位及膳食能量值等数据，例如：300ml 牛奶 50 卡，保证数据的合理性，科学性。本系统不过多考虑当下对人们身体影响较小的指标如蛋白质、微量元素等，采用现在用户更加关注，对我们影响更大的卡路里作为推荐数据的衡量标准，提高数据的可扩展性。用户所得到的饮食推荐中这里的膳食信息将是重要依据，用来计算推荐套餐能量和摄入能量。

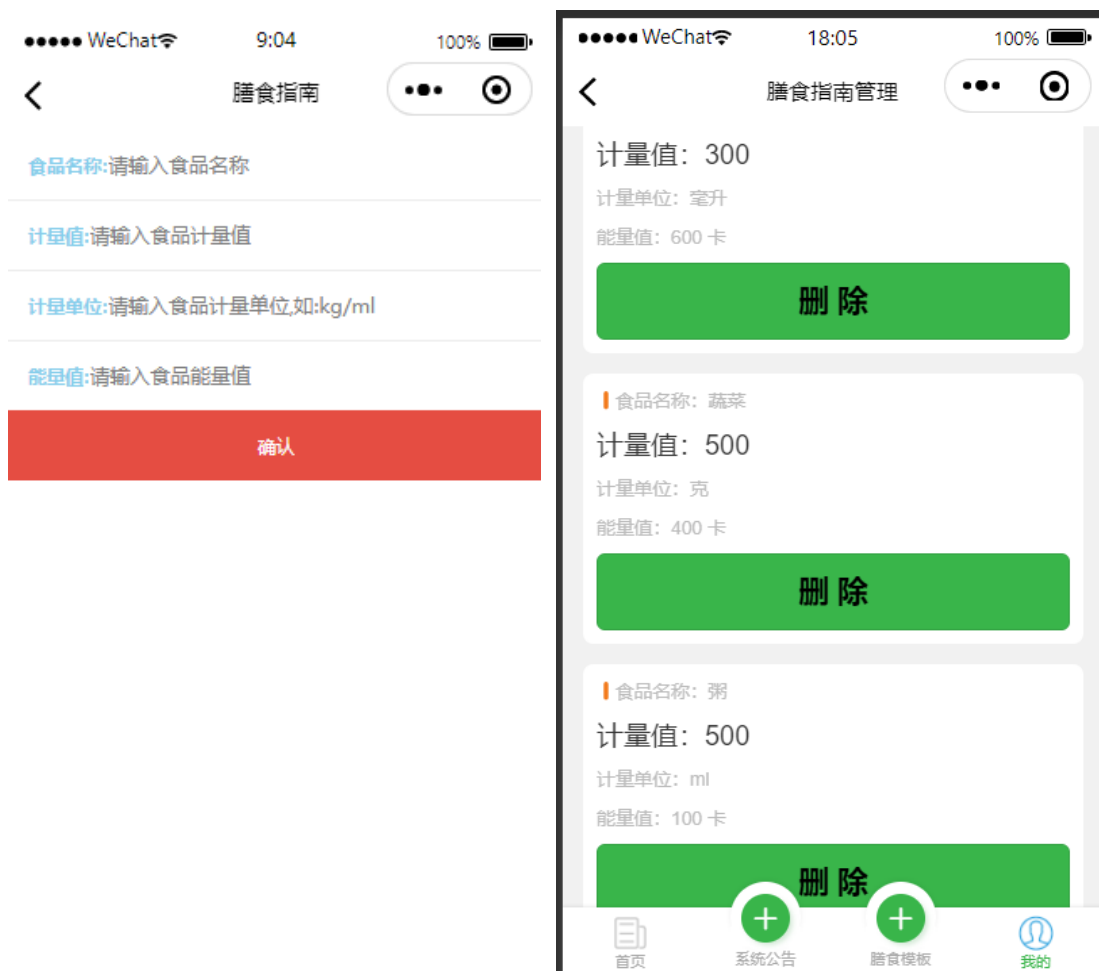


图 4-22 管理员管理膳食界面图

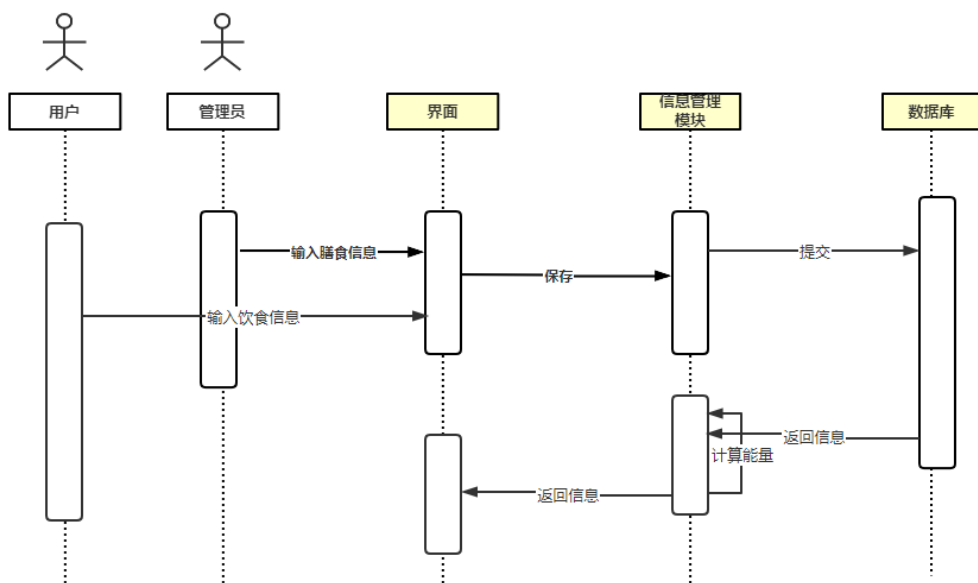


图 4-23 膳食管理时序图

核心代码如下：

```

@Override
public JsonResult getZhiNanList(String s_name, Integer userId, Integer page) {
    QueryWrapper<ZhiNan> queryWrapper = new QueryWrapper<>();//查询条件
生成器

    IPage<ZhiNan> contentIPage = new Page<>(page, 100);//参数一是当前页，参
数二是每页个数
    return JsonResult.succ(zhiNanMapper.selectPage(contentIPage, queryWrapper));
}
@Override
public String sheru(Integer userId) {
    BigDecimal sheru = BigDecimal.ZERO;

    QueryWrapper<YinShi> queryWrapper = new QueryWrapper<>();
    queryWrapper.lambda().eq(YinShi::getUid,userId)
        .ge(YinShi::getAddtime, DateUtil.beginOfDay(new Date()));
    queryWrapper.orderByDesc("addtime");
    queryWrapper.last(" limit 1");
    YinShi yinShi = yinShiMapper.selectOne(queryWrapper);

    if(null == yinShi){
        return sheru.toString();
    }
    if(StringUtil.isNotEmpty(yinShi.getZaocan())){
        QueryWrapper<ZhiNan> queryWrapper1 = new QueryWrapper<>();
        queryWrapper1.lambda().eq(ZhiNan::getFname,yinShi.getZaocan());
        ZhiNan zhiNan = zhiNanMapper.selectOne(queryWrapper1);
        if(null != zhiNan){
            sheru = sheru.add(new BigDecimal(zhiNan.getNengliang()));
        }
    }
    if(StringUtil.isNotEmpty(yinShi.getWucan())){
        QueryWrapper<ZhiNan> queryWrapper1 = new QueryWrapper<>();
        queryWrapper1.lambda().eq(ZhiNan::getFname,yinShi.getWucan());
        ZhiNan zhiNan = zhiNanMapper.selectOne(queryWrapper1);
        if(null != zhiNan){
            sheru =sheru.add(new BigDecimal(zhiNan.getNengliang()));
        }
    }
}

```

```

        if(StringUtil.isEmpty(yinShi.getWancan())){
            QueryWrapper<ZhiNan> queryWrapper1 = new QueryWrapper<>();
            queryWrapper1.lambda().eq(ZhiNan::getFname,yinShi.getWancan());
            ZhiNan zhiNan = zhiNanMapper.selectOne(queryWrapper1);
            if(null != zhiNan){
                sheru =sheru.add(new BigDecimal(zhiNan.getNengliang()));
            }
        }
        return sheru.toString();
    }
}

```

4.3.7 食谱推荐模块

食谱推荐管理是本系统最核心的模块。首先系统会根据用户的个人信息计算出用户的 BMI，即身体质量指数。公式为 $BMI = \text{体重(kg)} / \text{身高(m)}^2$ ，系统也会根据用户信息计算出理想体重，公式为 $(\text{身高}-80) * 0.7$ ，与用户的真实体重作对比，作为给用户选择套餐的依据。

此推荐套餐分为三种：减肥套餐、健康套餐、营养套餐，根据国际食品组织规定与实际情况结合，本系统选定减肥套餐标准能量值为 1000 卡，健康套餐为 1250 卡，营养套餐为 1500 卡。系统选用当下人们最关注且对身体健康影响最大的卡路里作为数据标准，关注食物大类进行推荐管理，每种套餐系统会推荐三种食物，食物总卡路里数最接近标准能量，后台会为每种套餐提供三种备选套餐，以供用户选择。用户若不喜欢此推荐可自行点击选择食物，系统会计算相应能量值，供用户参考。

此外，用户可添加今天的饮食日记，用户输入早餐/午餐/晚餐，和食用量，系统进行模糊查询并计算，点击添加按钮，系统累计食物热量，将所查膳食信息能量进行计算，系统会根据套餐标准能量值减去已摄入能量，做后续饮食推荐，若用户摄入能量已超标，系统则会建议运动为主。



图 4-24 饮食推荐界面

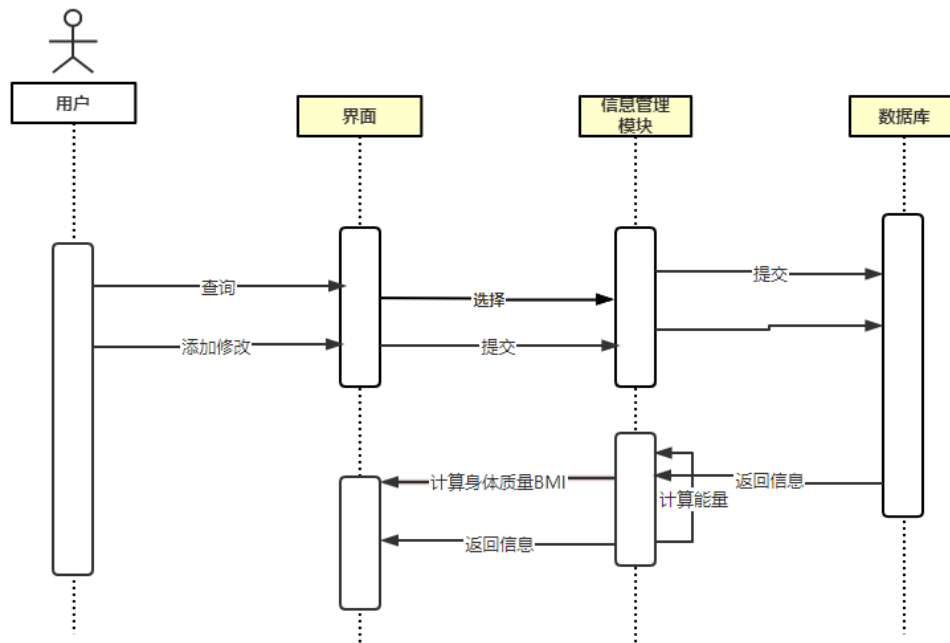


图 4-25 饮食推荐时序图

核心代码：

```

public void createShiPu() {
    QueryWrapper<ShiPu> queryWrapper1 = new QueryWrapper<>();
    shiPuMapper.delete(queryWrapper1);

    QueryWrapper<ZhiNan> queryWrapper = new QueryWrapper<>();
    queryWrapper.lambda().eq(ZhiNan::getFflag, 0);
    List<ZhiNan> zhiNans = zhiNanMapper.selectList(queryWrapper);
    //一种食物的食谱
    for (int m = 0; m < zhiNans.size(); m++) {
        ShiPu shiPu = new ShiPu();
        shiPu.setFids("," + zhiNans.get(m).getId() + ",");
        shiPu.setTotal(zhiNans.get(m).getNengliang());
        shiPu.setAddtime(new Date());
        shiPuMapper.insert(shiPu);
    }
    //两种食物的食谱
    for (int m = 0; m < zhiNans.size(); m++) {

```

```

for (int n = 0; n < zhiNans.size(); n++) {
    if (n <= m ){
        //排除重复
        continue;
    }
    ShiPu shiPu = new ShiPu();
    shiPu.setFids(","+zhiNans.get(m).getId()+","
        +zhiNans.get(n).getId()+",");
    shiPu.setTotal(new BigDecimal(zhiNans.get(m).getNengliang())
        .add(new BigDecimal(zhiNans.get(n).getNengliang()))
        .toString());
    shiPu.setAddtime(new Date());
    shiPuMapper.insert(shiPu);
}
}

//三种食物的食谱
for (int m = 0; m < zhiNans.size(); m++) {
    for (int n = 0; n < zhiNans.size(); n++) {
        if (n <= m ){
            //排除重复
            continue;
        }
        for (int k = 0; k < zhiNans.size(); k++) {
            if(k <= m || k <= n){
                //排除重复
                continue;
            }
            ShiPu shiPu = new ShiPu();
            shiPu.setFids(","+zhiNans.get(m).getId()+","
                +zhiNans.get(n).getId()+",")

```

```

        +zhiNans.get(k).getId()+","");
        shiPu.setTotal(new BigDecimal(zhiNans.get(m).getNengliang())
        .add(new
BigDecimal(zhiNans.get(n).getNengliang()))
        .add(new
BigDecimal(zhiNans.get(k).getNengliang()))
        .toString());
        shiPu.setAddtime(new Date());
        shiPuMapper.insert(shiPu);
    } } }

```

@Override

```

public ShiPu selectShiPu(String nengliang) {
    QueryWrapper<ShiPu> queryWrapper = new QueryWrapper<>();
    queryWrapper.lambda().le(ShiPu::getTotal,new BigDecimal(nengliang));
    queryWrapper.orderByDesc("total");
    queryWrapper.last("limit 3");
    List<ShiPu> shiPus = shiPuMapper.selectList(queryWrapper);
    if(shiPus.size() > 0){
        Random random = new Random();
        int num = random.nextInt(shiPus.size());
        return shiPus.get(num);
    } else {
        ShiPu shiPu = new ShiPu();
        shiPu.setFids("");
        shiPu.setTotal("0");
        return shiPu;
    } } }

```

本模块功能为此基于微信小程序的膳食营养管理系统最核心部分，采用穷举算法，依赖计算机的强大计算能力穷举出每一种可能的情况。穷举算法效率不高，但是适合一些没有明显规律可循的场合。算法的好坏并不完全依托其使用效率等因素，适合的

场景搭配使用适合的算法才是解决软件工程问题的重要思路。因为本系统目的是根据不同用户的不同身高、体重特点,推荐不同不同的饮食食谱,并且食谱并非单一选择。所以选用穷举法是最为合适的算法选择。

再此数据库中每个食品都有其唯一的食品 id, 算法根据食品 id 值, 排列组合出多种食谱。首先只有一种食物的食谱, 即为根据食品 id 的展开罗列; 两种食物的食谱, 将所有食物 id 排列组合, 排除重复组合即为所得食谱; 以此类推, 三种食物食谱算法理念相同, 本系统最多设定推荐三种符合标准能量值食物。用数学方法解释上述问题即为:

(一、二、、、: 食谱种类; 1, 2, 3,,, : 食物种类。假设有三种食物 1, 2, 3)

食谱 id	食物组合 id
一	1
二	2
三	3
四	1, 2
五	1, 3
六	2, 3
七	1, 2, 3

则共有 7 中搭配方式。

本系统即用穷举算法实现上述方式, 来推荐饮食套餐。


```

public class AddFileServiceTest {
    @Autowired
    private FileService service;
    /**
     * 测试单元 – 创建食谱
     * @throws AddException
     */
    @Test
    public void testAddFile() throws AddException {
        System.out.println(service.createFile(new AddFileRequest(
            151818002375896454448,
            " https://wangchuan-typora.oss-cn-
shanghai.aliyuncs.com/img/202204212100821.gif",
            "1518179574997934082" ,
            "b814b0d028bd398c909cda1db76e8027",
            ".jpg",
            "2022-04-24",
            "2022-04-24",
            "0",
            "/root",
            "shipu",
            null,
            915849
        )));
    }
    /**
     * 测试单元 – 选择食谱
     * @throws AddException
     */
    @Test
    public void testUpdateFile() throws AddException {
        System.out.println(service.updateFile(
            new FileRequest(
                1518180023742324738,
                " https://wangchuan-typora.oss-cn-
shanghai.aliyuncs.com/img/202204212100821.gif",
                "2022042121008215641" ,
                "b814b0d028bd398c909cda1db76e8027",

```

```

    "ppt",
    "2022-04-24",
    "2022-04-24",
    "0",
    "/root",
    "shipu",
    null,
    20992
    ));}
    }}

```

分别进行单元的集合测试后，将测试结果对比数据库与系统中的数据，结果表明测试全部通过，其他部分的测试同上，且通过测试。

5.3 系统功能测试

功能测试是对产品的各项功能进行验证，根据功能测试用例逐项测试，检查产品是否满足用户要求的功能。功能测试也称为黑盒测试。它只需要考虑要测试的每个功能，一般不需要考虑整个软件的内部结构和代码，从软件产品的接口和架构出发，根据需求编写测试用例，并在预期结果和实际结果之间进行评估，使产品满足用户的要求。测试分为两大模块：管理员测试模块、用户测试模块。

表 5-1 管理员测试模块测试用例表

测试点	测试用例	期望结果	实际结果
登录模块	微信开发者工具进行登录，登录管理员账号	登录成功	符合预期结果
推荐列表设置	点击推荐列表管理设置按钮	1.系统弹出推荐列表界面 2.输入推荐标题和内容。 3.添加成功，保存后在首页能够查看。	符合预期结果
用户管理设置	点击用户管理按钮	1.弹出用户 列表 2.选择指定用户 3.管理员授权用户是否启用。	符合预期结果
公告管理	点击系统公告按钮	1.弹出发布公告界面 2.点击“发布公告” 3.点击公告管理确认公告发布成功	符合预期结果
膳食指南管理	点击膳食模板按钮	1.弹出膳食指南添加界面 2.输入食品名称、计量值、单位、能量值 3.单击确认后在膳食指南管理中查看是否成功 4.管理员可进行删除管理	符合预期结果
意见反馈管理	点击意见反馈按钮	1.弹出意见反馈界面	符合预期结果

		2.输入反馈内容和联系方式 3.单击提交后可在反馈列表中进行查询	
--	--	-------------------------------------	--

表 5-2 用户测试模块测试用例表

测试点	测试用例	期望结果	实际结果
意见反馈管理	点击意见反馈按钮	1.弹出意见反馈界面 2.输入反馈内容和联系方式 3.单击提交	符合预期结果
公告管理	点击系统公告按钮	查看公告信息	符合预期结果
个人信息管理	点击查看个人信息按钮 后单击录入个人饮食信息按钮	1.弹出个人信息界面 2.添加和修改个人信息进行提交保存 3 弹出个人饮食信息界面 4.添加早/中/晚餐信息	符合预期结果
个性化推荐管理	单击个性化推荐按钮	1.弹出饮食建议界面 2.系统根据个人信息计算 BMI 身体质量，以此为基准来推荐饮食套餐 3.添加三餐饮食信息后，系统根据剩余可摄入能量继续推荐 4.用户选择查看	符合预期结果
密码修改管理	单击密码修改按钮	1.弹出密码修改界面 2.用户输入原始密码和新密码进行秘密修改	符合预期结果

5.4 系统性能测试

由于本系统尚未上线发布，没有涉及到大量并发。因此系统对硬件的要求并不是很高：以 1Ghz 处理器、2G 内存以及操作系统 Windows Server 2000 为基准即可。系统运行不要求服务器，对系统的执行速度没有过高要求，系统的响应实现控制在 50-100 毫秒以内。系统在一次响应过程中，大约 10 Mbyte 的数据流程系统，对数据库的压力在可控范围之内。系统中数据接收和发送的时间间隔大约是 20 毫秒。系统的吞吐量最大能达到 10 万 TPS。

性能测试采用 Jprofiler12 进行性能测试，在每秒 1000 次请求下系统性能如图 5-2 系统性能测图所示。

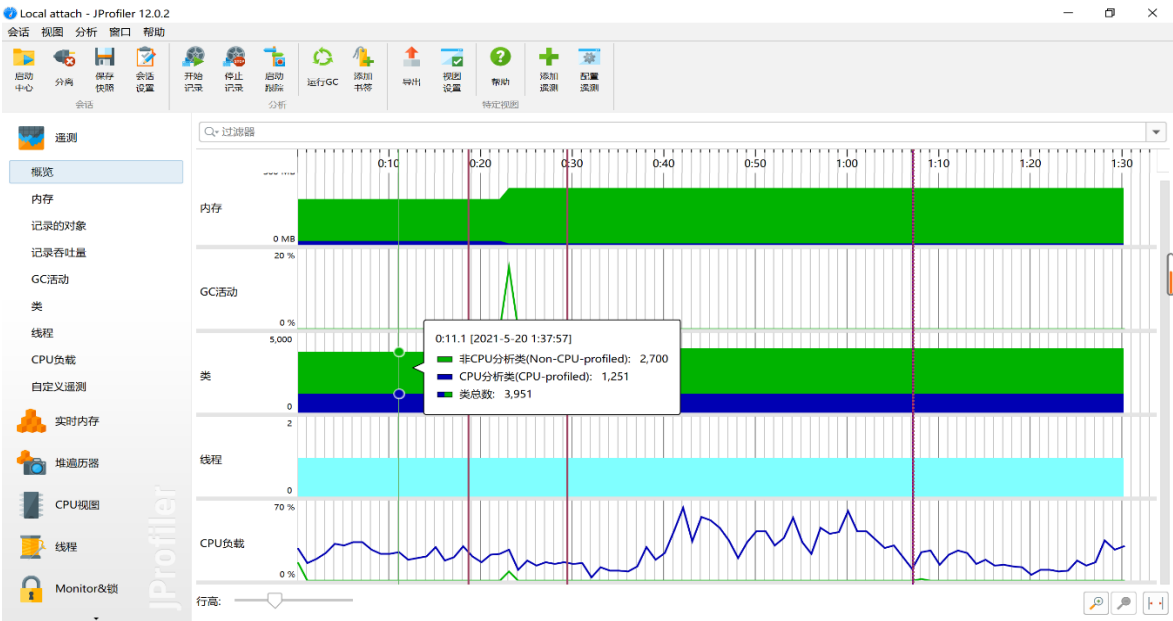


图 5-2 系统性能测图

5.5 系统测试报告

以下为基于微信小程序的膳食营养管理系统的设计与实现的测试报告。

表 5-3 测试模块

测试内容	测试情况及建议
登录模块	系统正常运行，符合结果预期，没有错误
推荐列表设置	推荐流程正确执行，符合结果预期，没有错误
用户管理设置	用户管理流程正确执行，符合结果预期，没有错误
公告管理	公告管理流程正确执行，符合结果预期，没有错误
膳食指南管理	能够正常显示膳食指南信息
意见反馈管理	能够正常显示已经反馈信息
个人信息管理	个人信息管理流程正确执行，符合结果预期，没有错误
个性化推荐管理	个性化推荐管理流程正确执行，符合结果预期，没有错误
密码修改管理	密码修改管理流程正确执行，符合结果预期，没有错误

表 5-4 风险项

序号	重要程度	问题描述	原因/备注
01	低	程序可能卡顿	本机系统老化

经过各项测试，分析其结果可得本系统能实现预期目标。具体分析过程如下：

1. 简洁性

通过测试，此微信小程序系统，界面美观，操作流畅，用户体验感舒适，总体来

说本系统具有简洁的特点。

2. 可靠性

本系统基于实际情况出发,考虑法律道德规定,且系统均为自行设计,安全性高,登录时也采用 JWT 同时数据加盐的方式,更好的保证用户登录的安全性,且没有信息泄露等风险,所以本系统具有可靠性。

3. 响应性

经测试系统的响应实现控制在 50-100 毫秒以内。系统在一次响应过程中,大约 10 Mbyte 的数据流程系统,对数据库的压力在可控范围之内。所以本系统响应性能好。

4. 易扩展性

本系统内数据严格按照国际《营养能量表》进行添加删除,食谱推荐严格按照国际化营养标准推荐,数据具有普适性、科学性,对不同管理员管理时数据具有易扩展性。后台多采用数据封装、接口封装的方式,对于后续程序开发也具有易扩展性。

在系统测试过程中,发现并解决了很多问题,逐步完成对项目的改善。同时对系统进行了简洁性、可靠性、响应性、易扩展性的测试,表明此系统符合上文需求说明中规定的标准。

结 论

针对当下人们越来越关注膳食营养,希望在日常生活中管理膳食来保持健康的问题,本文设计开发一个基于微信小程序的膳食营养管理系统。本系统后端采用 Java 语言,前端采用 Vue 等技术,选用开放式服务体系,即基于微信小程序开发,选择 SpringBoot 和 Uniapp 框架,数据库为 MySQL。本系统实现了用户对膳食营养管理的功能,解决了用户想要轻松便捷访问的需求问题,并且良好简洁的界面效果使得面对多职业,多种类人群仍然可以轻松使用。

本系统分为管理员和用户两种角色来管理使用此小程序,管理员主要参与用户管理、公告管理、反馈管理以及膳食模板管理等模块。用户参与个人信息、饮食情况以及饮食推荐管理等模块。为更好的迎合大众需求,本系统选择当下对人体健康影响最大且人们最关注的卡路里作为膳食推荐指标,食品信息严格按照国际《营养信息表》添加管理,食品推荐按国际食品健康标准。本系统采用穷举法来实现饮食套餐推荐,尽最大可能满足用户需求。

系统的不足之处是由于开发时间和本人目前能力技术限制,没有实现对膳食营养多种影响因素的管理实现,后续的改进方向为在推荐食谱时计算考虑碳水、蛋白质等;并且膳食贴士推荐仍是管理员手动添加,后续的改进方向为爬虫式获取膳食贴士推荐数据,以确保数据的完整正确,减少人力资源,提高可扩展性。

致 谢

大学四年生活如水流日沉般过去。回首种种，万般不舍，藏于心中。

一朝沐杏雨，一生念师恩。在刚开始确定选题时特别迷茫畏惧，担心自己不能完成毕业设计的开发实现，当时也正值考研最后冲刺阶段，很是慌乱，后来王思飞老师和我们单独电话会议，根据我们感兴趣和以后工作学习的方向，一同商量出最后的选题。老师当时总说，不用担心，毕设可以考研结束后开始着手，让我们放宽心，集中精力做最后的冲刺。虽然最后我并没有如愿考上研究生，但是不悔不惧，自己拼搏过努力过，这份坦然也来自老师在背后的付出，是他的提前了解，再三思考后再同我们商量确定内容，减少我们的焦虑，避免我们走岔路的可能。在程序编写和论文修改过程中，王思飞老师在学校工程认证的疲惫忙碌之中，仍抽时间同我们商量解决毕设中出现的问题，帮我们解答各种疑问，细致耐心，是老师的责任感，让我们更加努力想更好的完成这次毕业设计，对得起老师的付出，对得起自己的努力。在此我由衷的感谢我的毕业设计指导老师，王思飞老师，不仅是之前对课程的教授，更是现在认真负责的毕设指导，且陪我们走过大学四年中最后、最珍惜的时光，我很荣幸成为老师的学生。

当然大学四年，仍有很多我很想念和感谢的老师，玲玲老师，建慧老师，凤君老师，锴书记。。。脑子里涌现老师们上课的情形，五味杂陈，在此学生衷心祝愿老师们身体健康，万事顺遂，虽然我不是最好的学生，但您们是我最敬爱的老师，遇到您是我人生之幸。

山河不足重，重在遇知己。写到这里总想感叹，感叹时光流逝，感叹我们没能在一起拍毕业照片，感叹我们以后可能天南海北，感叹仍有千万言语却只能提笔寥寥数字。在此让我用最后的使用浪漫的权利，让你们的名字出现在我最重要的毕业论文中吧，周炎、安晚亭、刘钰钰、姜宏瑜。。。初见乍惊欢，久处仍怀然，人海茫茫，待以相遇，何其有幸。愿你们平安喜乐，前程似锦。

篇幅有限，话语太多，请让更多的话留在我们下次见面。行文至此，我的大学时光也即将落下帷幕，感恩父母、老师、朋友和自己，感恩遇见感谢相逢。愿我们健康快乐，未来可期！

参考文献

- [1] Reto Meier. Professional Android 2 Application Development[M]. Birmingham,UK: Wrox, 2018.
- [2] Dang Tran Khanh,Huy Ta Manh,Dang Ly Hoang,Le Hoang Nguyen. An Elastic Data Conversion Framework: A Case Study for MySQL and MongoDB[J]. SN Computer Science,2021,2(4).
- [3] Craig Walls, Ryan BreidenBach. Spring In Action[M]. Manning Publications. 2019.
- [4] 焦鹏琿. 基于 SpringBoot 和 Vue 框架的电子招投标系统的设计与实现[D].南京大学,2018:10-16.
- [5] 宋永鹏. 基于 MySQL 的数据库查询性能优化[J].电子设计工程,2021,29(12):43-47.
- [6] 赵斌. 软件测试技术经典教程[M]. 北京:电子工业出版社, 2011:1-72.
- [7] 孔祥盛. MySQL 数据库基础与实例教程[M]. 北京:人民邮电出版社, 2014:22-23.
- [8] 刘世峰,康来松,宫大庆. 面向群组的事件兴趣点推荐算法研究[J/OL].中国管理科学,2021.0383.
- [9] 周艳,李凯,付高宇,向铭杰,曹建波,叶栩见. 基于 TSVD 的协同过滤推荐算法研究[J].电脑知识与技术,2022:55-57.
- [10] 唐小健. 探究穷举法在 C 语言项目实践中的应用[J].电脑编程技巧与维护,2022(04):22-25.
- [11] 李海燕,陈善柳. 基于微信小程序的教学平台建设及应用[J].科技与创新,2022(09):174-176.
- [12] 刘永轩. 基于微信小程序的地名信息外业采集系统设计与开发[J].北京测绘,2022,36(04):503-506.
- [13] 张彤,姜攀. 基于行为特征识别的大学生心理测试微信小程序设计与实现[J].电脑编程技巧与维护,2021(08):50-54.
- [14] 杨冠慰. MySQL 数据库的数据隐私及安全机制研究[J].信息与电脑(理论版),2020,32(20):140-142.
- [15] 胡静. 浅析黑盒测试与白盒测试[J].衡水学院学报, 2018,10(1):16-19.

附 录

1. 系统使用说明书

系统管理员为小程序各类信息的维护者，用户为各行各业小程序的使用者。以下是系统使用流程。

管理员注册登录小程序后。根据按钮提示，可对首页的推荐小贴士进行查询管理，也可在“我的”里面进行增加修改；点击“系统公告”按钮可对公告进行增添管理，输入公告的标题和内容进行发布；点击“膳食模板”按钮，按规则填入膳食名称、食品计量值，食品计量单位和食品的能量值来添加管理膳食数据；点击“我的”管理员可对用户进行管理，决定是否启用该用户；公告管理可以对公告进行删除操作；点击“意见反馈”按钮，管理员可将自己的意见与联系方式提交给后台程序员；管理员也可查看用户反馈列表，进行查询管理。

用户注册登录小程序后。可在首页看到系统推送的健康小贴士，点击可进行查看内容。点击“我的”按钮，在意见反馈中填写用户对小程序的意见，并提交。用户可在“公告”按钮点开查看系统公告。点击查看“个人信息”按钮，用户可按要求输入或修改个人信息。点击“录入个人饮食信息”按钮，用户可添加早中晚的饮食情况，作为下面个性化推荐的重要依据。食谱推荐管理是本系统最核心模块。首先系统会根据用户的个人信息计算出用户的 BMI，即身体质量指数。公式为 $BMI = \text{体重(kg)} / \text{身高(m)}^2$ ，系统也会根据用户信息计算出理想体重，共给用户选择套餐的依据。

此推荐套餐分为三种：减肥套餐、健康套餐、营养套餐，根据国际食品组织规定与实际情况结合，本系统选定减肥套餐标准能量值为 1000 卡，健康套餐为 1250 卡，营养套餐为 1500 卡。每种套餐系统会推荐三种食物，食物总卡路里数最接近标准能量，后台会为每种套餐提供三种备选套餐，以供用户选择。用户若不喜欢此推荐可自行点击选择食物，系统会计算响应能量值，供用户参考。

此外，用户可添加今天的饮食日记，用户输入早餐/午餐/晚餐，系统会根据套餐标准能量值减去已摄入能量，做后续饮食推荐，若用户摄入能量已超标，系统则会建议运动为主。

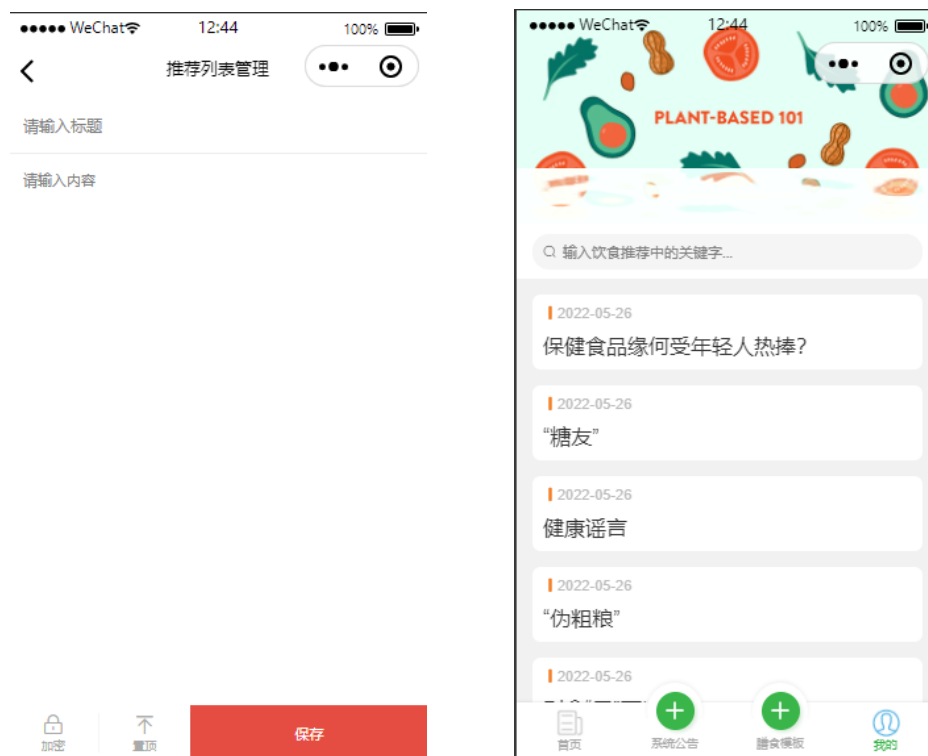
2. 运行结果如下所示

(1) 管理员登录微信小程序



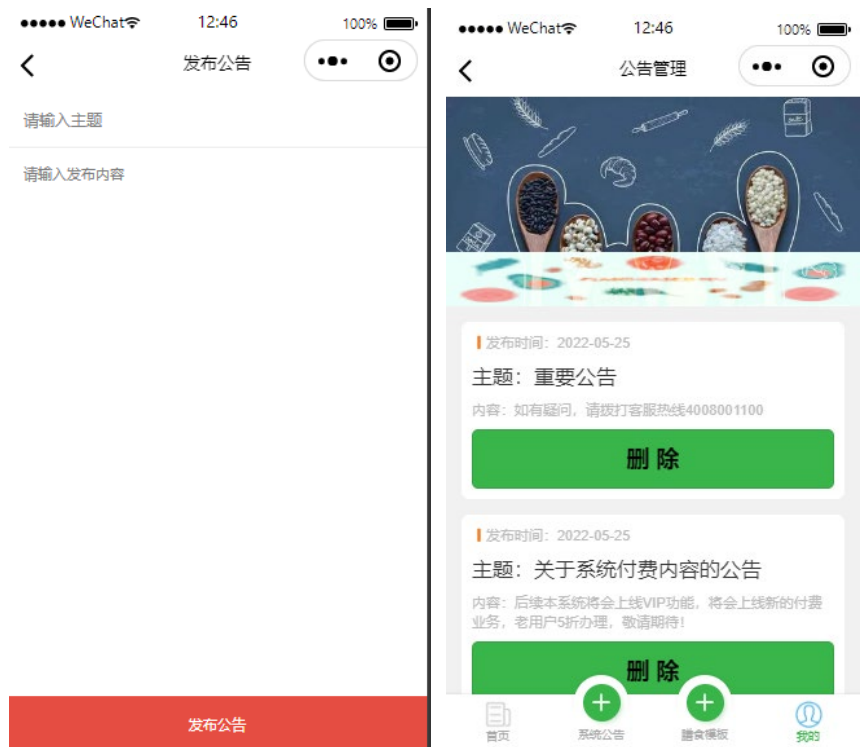
附图 1 用户登录页面

(2) 管理员登录微信小程序首页，管理推荐列表



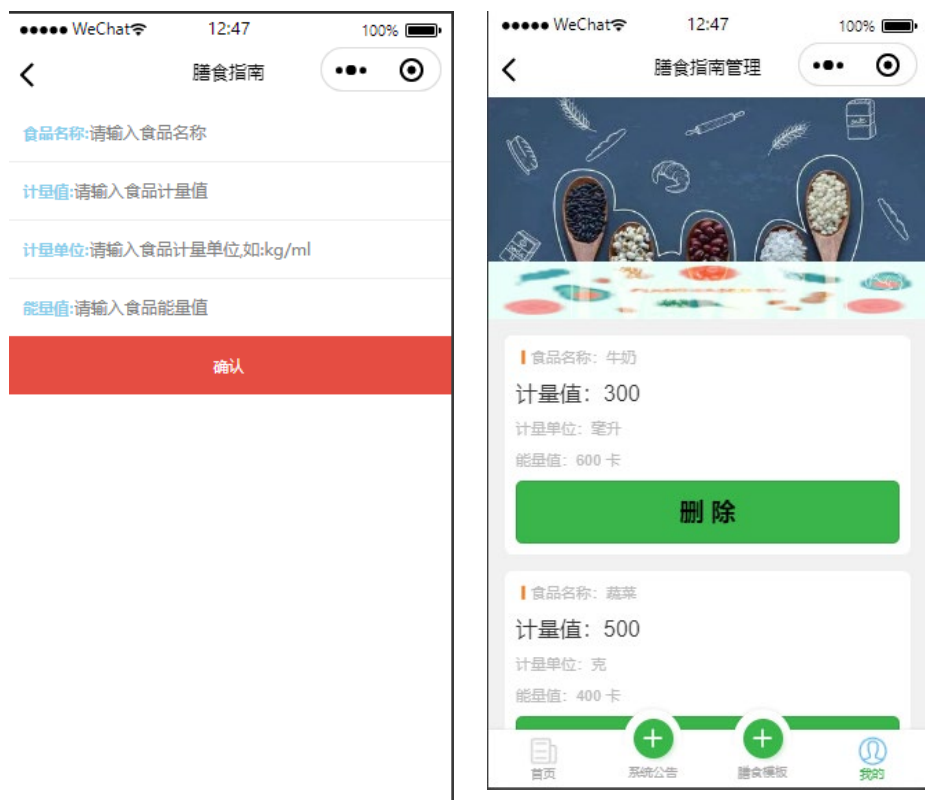
附图 2 管理员登录首页界面

(3) 管理员管理系统公告



附图 3 管理员公告管理界面

(4) 管理员管理膳食模板



附图 4 膳食管理界面

(5) 管理员管理用户



附图 5 管理用户界面

(6) 管理员管理意见反馈



附图 6 反馈管理界面

(7) 用户注册登录首页



附图 7 用户登录首页

(8) 用户意见反馈



附图 8 用户意见反馈界面

(9) 用户管理个人信息

个人信息

姓名: CVV

年龄: 22

身高: 165

体重: 50

提交

附图 9 个人信息管理界面

(10) 用户录入饮食情况

个人饮食信息

早餐:

牛奶

食用量: 60

+

午餐:

午餐吃了什么

食用量:

+

晚餐:

晚餐吃了什么

食用量:

+

保存

附图 10 饮食信息管理界面

(11) 用户查看饮食记录



附图 11 用户查看饮食记录界面

(12) 用户查看个性化饮食推荐



附图 12 用户查看饮食建议界面图

3. 程序核心源代码

(1) 食谱推荐代码

```
public void createShiPu() {
    QueryWrapper<ShiPu> queryWrapper1 = new QueryWrapper<>();
    shiPuMapper.delete(queryWrapper1);

    QueryWrapper<ZhiNan> queryWrapper = new
QueryWrapper<>();
    queryWrapper.lambda()
        .eq(ZhiNan::getFflag, 0);
    List<ZhiNan> zhiNans = zhiNanMapper.selectList(queryWrapper);

    //一种食物的食谱
    for (int m = 0; m < zhiNans.size(); m++) {
        ShiPu shiPu = new ShiPu();
        shiPu.setFids("," + zhiNans.get(m).getId() + ",");
        shiPu.setTotal(zhiNans.get(m).getNengliang());
        shiPu.setAddtime(new Date());
        shiPuMapper.insert(shiPu);
    }

    //两种食物的食谱
    for (int m = 0; m < zhiNans.size(); m++) {
        for (int n = 0; n < zhiNans.size(); n++) {
            if (n <= m ){
                //排除重复
                continue;
            }
            ShiPu shiPu = new ShiPu();
            shiPu.setFids("," + zhiNans.get(m).getId() + ","
                + zhiNans.get(n).getId() + ",");
            shiPu.setTotal(new
BigDecimal(zhiNans.get(m).getNengliang())
                .add(new
BigDecimal(zhiNans.get(n).getNengliang()))
                .toString());
            shiPu.setAddtime(new Date());
        }
    }
}
```

```

        shiPuMapper.insert(shiPu);
    }
}

//三种食物的食谱
for (int m = 0; m < zhiNans.size(); m++) {
    for (int n = 0; n < zhiNans.size(); n++) {
        if (n <= m ){
            //排除重复
            continue;
        }
        for (int k = 0; k < zhiNans.size(); k++) {
            if(k <= m || k <= n){
                //排除重复
                continue;
            }
            ShiPu shiPu = new ShiPu();
            shiPu.setFids("," + zhiNans.get(m).getId() + ","
                + zhiNans.get(n).getId() + ","
                + zhiNans.get(k).getId() + ",");
            shiPu.setTotal(new
BigDecimal(zhiNans.get(m).getNengliang())
                .add(new
BigDecimal(zhiNans.get(n).getNengliang())
                .add(new
BigDecimal(zhiNans.get(k).getNengliang())
                .toString()));
            shiPu.setAddtime(new Date());
            shiPuMapper.insert(shiPu);
        }
    }
}
}

```

(2) 食谱选择代码

```

@Override
public ShiPu selectShiPu(String nengliang) {
    QueryWrapper<ShiPu> queryWrapper = new QueryWrapper<>();
    queryWrapper.lambda().le(ShiPu::getTotal,new

```

```

BigDecimal(nengliang));
        queryWrapper.orderByDesc("total");
        queryWrapper.last("limit 3");
        List<ShiPu> shiPus = shiPuMapper.selectList(queryWrapper);
        if(shiPus.size() > 0){
            Random random = new Random();
            int num = random.nextInt(shiPus.size());
            return shiPus.get(num);
        }else {
            ShiPu shiPu = new ShiPu();
            shiPu.setFids("");
            shiPu.setTotal("0");
            return shiPu;
        }
    }
}

```

(3) 能量摄入计算

```

public String sheru(Integer userId) {
    BigDecimal sheru = BigDecimal.ZERO;

    QueryWrapper<YinShi> queryWrapper = new
QueryWrapper<>();
    queryWrapper.lambda().eq(YinShi::getUid,userId)
        .ge(YinShi::getAddtime,
DateUtil.beginOfDay(new Date()));
    queryWrapper.orderByDesc("addtime");
    queryWrapper.last(" limit 1");
    YinShi yinShi = yinShiMapper.selectOne(queryWrapper);

    if(null == yinShi){
        return sheru.toString();
    }

    if(StringUtil.isEmpty(yinShi.getZaocan())){
        QueryWrapper<ZhiNan> queryWrapper1 = new

```



```

QueryWrapper<>();

queryWrapper1.lambda().eq(ZhiNan::getFname,yinShi.getZaocan());
        ZhiNan zhiNan =
zhiNanMapper.selectOne(queryWrapper1);
        if(null != zhiNan){
            sheru = sheru.add(new
BigDecimal(zhiNan.getNengliang()));
        }
    }
    if(StringUtil.isEmpty(yinShi.getWucan())){
        QueryWrapper<ZhiNan> queryWrapper1 = new
QueryWrapper<>();

queryWrapper1.lambda().eq(ZhiNan::getFname,yinShi.getWucan());
        ZhiNan zhiNan =
zhiNanMapper.selectOne(queryWrapper1);
        if(null != zhiNan){
            sheru =sheru.add(new
BigDecimal(zhiNan.getNengliang()));
        }
    }
    if(StringUtil.isEmpty(yinShi.getWancan())){
        QueryWrapper<ZhiNan> queryWrapper1 = new
QueryWrapper<>();

queryWrapper1.lambda().eq(ZhiNan::getFname,yinShi.getWancan());
        ZhiNan zhiNan =
zhiNanMapper.selectOne(queryWrapper1);
        if(null != zhiNan){
            sheru =sheru.add(new
BigDecimal(zhiNan.getNengliang()));
        }
    }
    return sheru.toString();
}
}

```