

移动智能开发第一阶段

1.项目可行性报告（行业分析、竞争对手或同类产品分析，自身条件分析）

（1）行业分析

近年来，中国消费者对健康生活的追求持续加强，营养保健产业市场规模也有较高幅度的持续增强。消费者们的生活理念、健康管理行为正在发生变化，结合技术创新、产品与服务模式创新，市场需求也呈现出一些演变趋势。

对健康生活的追求导致人们对改善营养摄入产生更多需求。人们健康生活依赖三大支持：合理营养摄入；适当规律运动；良好休息睡眠。当代人健康生活意识的不断提升，使得对如何吃得健康营养也有了更多追求。

营养摄入方案可一定程度帮助解决多种健康问题，这里有几个类型。如图所示：改善人体机能降低患病风险（微量元素平衡、体重控制、提高免疫力）；辅助治疗控制各类慢性疾病（糖尿病、心血管疾病、精神疾病、慢性炎症、骨质疏松、其他代谢问题）；满足不同生理阶段人群健康需要（婴幼儿、青少年、孕产妇、更年期、中老年）；满足特殊职业健康需要（运动员、脑力工作者、商务人士、有害工种、吸烟人群、熬夜人群.....）。

在日常饮食中，要想吃得健康，首先人们通过食物成分搭配及量上的控制，来令餐食更有利于健康；此外，通过选择额外服用营养补充剂等保健品来弥补日常餐食中较为缺失的营养成分。

而就当下中国现状，营养健康知识鸿沟巨大，专业营养师服务的普及率在中国尚非常低；食物成分含量缺乏透明度；人们对营养产品及销售渠道缺乏信任；缺少满足个性化需求的营养产品；健康营养餐食长期坚持难；对营养品效果如何也缺乏有效的评估途径。这些都是目前营养产业消费市场的需求痛点，在后面关于营养产业数字创新场景的讨论中会再进一步对照需求痛点进行分析。

然而长久以来，一般消费者在更健康的进行饮食管理、营养调理上一直处于十分迷茫的状态，专业人士、产品与服务供应商、渠道商他们掌握的信息与消费者之前存在很大的鸿沟。当代消费者从各方面都有更强的自我掌控欲求，他们更期待缩小信息鸿沟，在一定程度上能吃得更明白。事实上，多数的市场需求痛点都归因于严重的信息鸿沟。

普通消费者们首先没能充分了解自身的健康状态，其次没有足够营养学知识来指导选购合适的食品、营养品以及相应摄入量摄入方法等。除此之外，对产品生产商、渠道商、服务商的认识也有限，最终因各维度、各环节的信息鸿沟而无法实现自我掌控。

而如今事态还在继续变得复杂。例如根据尼尔森的报告显示，人们为了获得健康，在做出相关产品采购时要考虑的因素变得更为复杂，许多消费者无法理解健康产品的相关说明，虽然产品提供了产品标签及说明资料，但这些信息往往过于专业，无法清晰理解，影响了消费者做出合适的采购选择。

与此同时，国际营养保健品热议的未来一大趋势——**精准营养**正逐渐成为保健行业的新宠儿。

与简单粗暴的传统“补身”相对应的，是通过基因的测定与分析，基于基因组和微生物组等大数据，制定个性化营养干预方案的精准营养。但是鉴于基因组检测等技术的高门槛，业界大多通过大量外在表现数据和个人感受与选择制定高度针对性的营养方案。

“精准营养”逐渐成为营养健康产业新热点，尤其在2020年新冠疫情影响下，全球健康意识普遍增强，个性化营养受到市场热捧。据Markets and markets研报数据显示，目前，个性化营养市场规模增长势头强劲，预计2025年将增至164亿美元。

精准营养行业现有的行业模式有哪些可以参考的信息？5月6日，国内精准营养领域的权威著作《精准营养白皮书》线上版抢先云发布。

该著作由汤臣倍健营养健康研究院、中国科学院上海营养与健康研究所、IFF(原杜邦营养与生物科技)、巴斯夫(BASF)、荷兰应用科学研究院(TNO)、庶正康讯科学与法规中心、中国老年保健医学研究会老年健康科技创新分会和中国营养保健食品协会研发专业委员会等联合发布。这项研究的学术成果——《精准营养研究与产业转化趋势》将由中国科学技术出版社于7月正式出版，它将系统性地总结国内外精准营养科研前沿和应用现状，深入分析未来科研、行业面临的机遇与挑战。

全球健康峰会(GWS)在2019年的趋势报告中，将「精准营养」列为全球八大健康趋势之一。各大品牌利用新的技术、数据工具、甚至是基因检测等终端设备来制定符合个人的营养解决方案。

(2) 竞争对手/同类型产品分析

I.香港食物安全中心官网：https://www.cfs.gov.hk/sc_chi/nutrient/search1.php

优势：信息来源真实可靠

缺点：页面简陋，无移动端应用，搜索结果极其模糊，信息筛选成本高



II.中国营养学会食物营养成分查询平台：<https://fq.chinafcd.org/>

优势：成分详细，检索成本低

缺点：无移动端应用，信息过少，无法记录保存信息



食物营养成分查询平台
Food Composition Database

中国疾病预防控制中心营养与健康所
中国营养学会

权威发布

搜索
Search

您现在的位置：食物营养成分查询平台 > 搜索词：油条

食物信息

食物名称

别名或俗名

英文名称

食部
Edible

水分
Water

能量
Energy

加入对比

油条

100%

21.8g

1636KJ

☐ 对比

上一页

1

下一页

中国疾病预防控制中心营养与健康所 版权所有 联系电话：010-67791292
地址：北京市西城区南纬路29号 邮编100050
京ICP备13005340号-2



中国疾病预防控制中心
营养与健康所
National Institute for Nutrition and Health
China Center for Disease Control and Prevention



中国营养学会
Chinese Nutrition Society

1



食物营养成分查询平台
Food Composition Database

中国疾病预防控制中心营养与健康所
中国营养学会

权威发布

搜索
Search

您现在的位置：食物营养成分查询平台 > 谷类及制品 > 小麦 > 油条

谷类及制品

小麦

稻米

玉米

大麦

小米、黄米

其它



暂无图片

油条

食物类：谷类及制品

亚 类：小麦

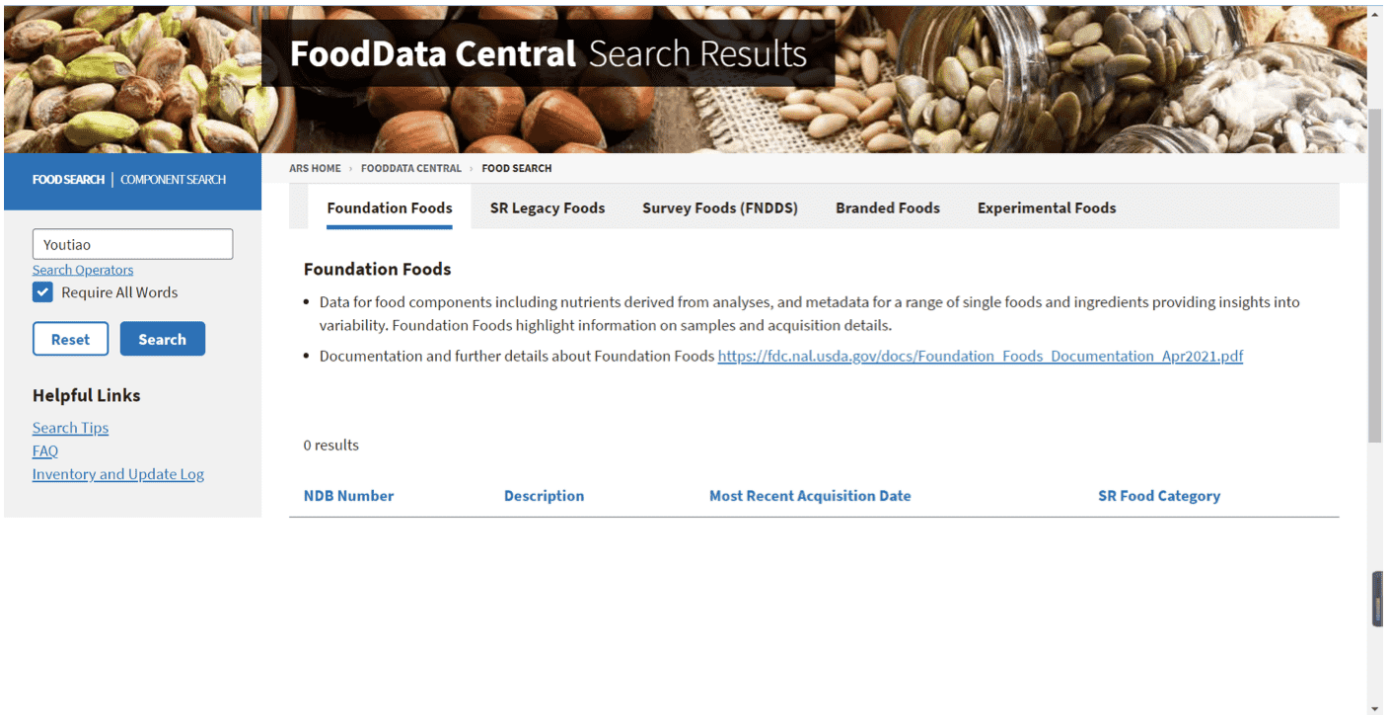
声明：本网站食物营养成分数据来源于中国疾病预防控制中心营养与健康所建立的中国食物成分表，未经中国疾病预防控制中心营养与健康所允许，任何单位和个人不得用于商业目的。

分享：

III.美国农业部网站的食物成分数据库：<https://fdc.nal.usda.gov/ndb/search/list>

优势：数据多样权威

缺点：访问困难，许多中国特色产品无法找到，页面信息全英文

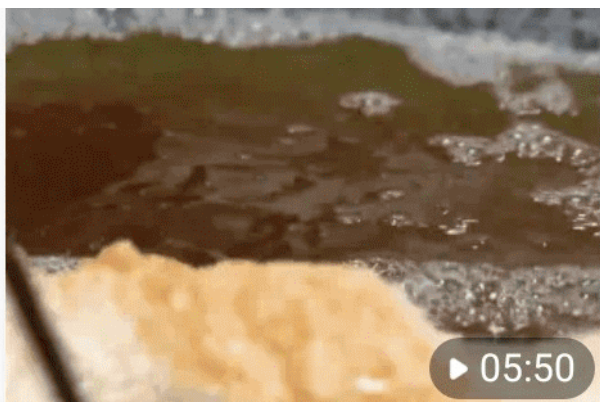


IV.香哈菜谱

优势：有移动端

缺点：没有详细的营养信息，无法满足痛点，应用信息过多显得臃肿，广告的出现更是进一步降低了使用体验





自制油条|轻松学会

刘仑 · 4861人学



油条丝瓜汤

誉誉的御膳 · 1.5万人学



芒

范理

推荐



自制无铝油条

面粉、鸡蛋、牛奶、干酵母、...
596.8万人做过



花恋花痴

♡ 6.4W



自制油条



解姐美食

♡ 1895





Q 油条



搜索

菜谱

综合

视频

笔记

用户

白云买菜就上叮咚买菜，生
鲜，肉禽，最快29分钟送...



会员低至**2.5元**/月

专业课程 学1000个名厨秘方

—— 您可获得 ——

✓ 1000个名厨秘方

✓ 上万部视频菜谱

炸油条

面粉、
14.2万



无广告打扰

立即开通

快速炸油条方法 麻辣德子



正点播出



2203

油条

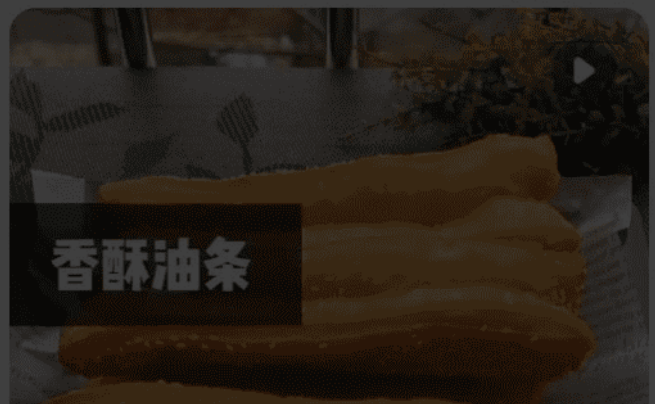
普通面粉、快速无铝油条膨松...
112.6万人做过



辛墨墨的小...



1.3W



没搜到？把你想找的菜谱告诉我们 >

2.产品定位及目标（用户群分析等）

1. 用户定位

产品主要供给想要规范化自己日常饮食的人群，如计划在减肥的肥胖人士，需要控制饮食的病人之类的。这类人群需要比较精确的知道食物的营养成分，让自己的饮食可以合理的提供所需的营养。而且这类人群有能力也会接受付费的定制化服务，消费水平可观。

2. 功能定位

- 可查询各种食物的营养成分，使用户可以准确的清楚吃什么食物能获得所需营养
- 收录各种常见疾病应该摄入的营养，提供相应的饮食方案

3. 风格定位

主要以查询功能为主，辅以提供建议的功能，帮助目标用户更好的规划自己的日常饮食

3.产品内容总策划（应用规划流程、设计与测试规范，开发日程表）

I.应用规划流程

为什么做？——“我们为什么要开发这个产品？”（战略层）

解决人们对于快速精准了解食物具体营养成分的需求。

做什么？——“我们要开发的是什么？”（范围层）

我们的产品主要以功能性为主，向用户精准快速地提供食物营养成分。

怎么做？——“我们如何达到上述抽象层面所确定的问题？”（结构层、框架层、表现层）

我们的产品将首先立足于最基本的需求痛点——通过用户输入食物名称或者相关信息返回相应的营养信息，供用户利用这些信息制定饮食方案。除此之外，我们将追求实现动态记录用户每一天的营养摄入情况，并且通过用户设定的营养目标动态地为用户提供饮食建议。

II.设计与测试规范

3.2.1.功能测试

接口测试

前后端分离项目开发实现过程中，后端接口编码调试通过后，要及时进行接口测试。

接口测试的重点是要检查数据的交换，传递和控制管理过程，以及系统间的相互逻辑依赖关系等。主要注意点：

- 1、业务功能正常场景、异常场景
- 2、输入输出边界测试：参数必填非必填；参数排序、个数、类型；参数长度；参数包含特殊字符

接口测试工具：postman、jmeter

功能测试

对软件功能进行的测试，主要检查软件功能是否实现了软件功能说明书（软件需求）上的功能要求，由测试负责人编写测试计划、测试用例，并实施，功能测试过程应记录缺陷并跟进解决。

功能测试一般进行如下几种情况的测试：

1. 正常情况
2. 非正常情况
3. 边界情况
4. 非法情况
5. 强度测试
6. 破坏性测试

7. 兼容性测试

8. 用户友好性测试

界面设计规范测试：

1、字体是否统一

2、字号是否符合规定

3、界面布局是否合理，整体效果如何

输入值测试：

1、数据类型

2、数据长度

3、约束条件是否满足，是否完整

性能测试

性能测试的主要范围：

1、系统的大用户压力；

2、系统的并发用户压力；

3、系统的数据库压力；

4、系统的稳定性等

输出《性能测试报告》，主要包括：性能指标：平均响应速度、吞吐量、系统用户的压力等数据。

测试工具：Loadrunner、jmeter

兼容性测试

兼容性测试主要包括web兼容性测试、APP兼容性测试、APP接口版本兼容性测试和数据兼容性测试

生产环境测试

将软件部署到用户实际生产环境后，由于环境差异，需要在用户现场进行确认测试，保证系统功能、性能完备，可正常运行。

3.2.2.测试用例标准

用例颗粒度原则：测试用例是执行的最小实体

用例划分基本原则是以最小功能模块来划分，为保障用例的可执行性、覆盖度，规范编写用例的粒度要求如下：

- 1.一个功能正常流程，编写一个测试用例；
- 2.一个功能中多个异常流程，应分开编写多个测试用例；
- 3.同一功能不同入口，可合并编写一个测试用例；
- 4.同一功能不同数据准备，应分开编写多个测试用例；
- 5.同一个功能用例的自动化用例和功能用例要匹配，若自动化用例不能完全覆盖功能用例，自动化用例和功能用例拆分两个互补测试用例；

用例编写要求规范

- 1.具有清晰名称、前提条件、操作步骤、期望结果的；
- 2.可被他人理解的；
- 3.可被他人执行的；

III.开发日程表



开发日程表



用户需求

软件开始开发前需要确定代价和所获得价值的对比，也就是 ROI（Return On investment），一旦确定需要创建，就需要安排一系列的资源来支撑这个软件的生存。这是需求的最原始描述。

为什么既要有用户需求，也要有产品需求？因为两者是有差异的，用户需求由用户提出，对技术一般不描述，只描述产品目标。产品需求是根据用户需求转化而来的技术实现需求，需要针对用户提出的产品目标进行细分，总结出每一个功能点，再针对每一个功能点细分为各种不同的操作流程，对每一个操作流程进行技术化定义。

用户需求和产品需求容易发生不一样，这是因为虽然大家都在谈需求，但是出发点可能不同，造成了双方关注点和思维方式不同。用户需求关注的是系统如何支持业务流程，背后的需求是“实现业务目标”。技术人员关注的是合理技术方案，背后的需求是“工作量”、“实现难度”和“系统性能”。

产品需求

我们需要弄清楚产品经理或项目需求提出者为什么要做这个项目？这是最本质的业务需求。需求分析确定的业务需求，都是从业务需求推导出来的，都必须为业务需求服务。

产品需求一般包括产品需求规格说明书和产品需求矩阵。产品需求矩阵一般按照子系统、功能集、执行单元的结构列出所有的功能需求，每列则对应每项功能的工作步骤以及每个步骤的工作量。

产品需求写完后，需要进行评审。在需求评审会上，产品、技术详细评审需求是否完整，产品功能的正常场景是什么？是否形成闭环？异常场景是什么？是否考虑周全？

需求评审后，开发和测试负责人，分别编写技术方案和测试用例。技术方案评审，开发负责人拉上涉及到其他系统的负责人一起讨论，技术方案中必须要有业务流程图和时序图，业务流程图是为了梳理开发对业务的理解，是否和需求一致。时序图是为了梳理本次需求涉及的系统交互。技术方案评审通过后，确认工作量和交付时间，反馈给产品。

总体设计

设计阶段的目标主要是对待开发系统的构架进行分析和设计，并建立系统构架的基线，以便为之后的实施工作提供一个稳定的基础。

设计阶段包括了系统架构的输出，一个好的系统架构设计可以帮助人类梳理业务逻辑且抓住核心需求，设计稳定可扩展的业务系统，评估业务开发周期和开发成本，有效的规避风险。例如盖房子的时候得有建筑图纸，有了图纸，才能核算施工周期。

总体设计是整个系统的框架型设计，意义及其重大，一般情况下不能省略（只有维护项目可以省略总体设计，因为基准项目已经设计完毕），所有的产品开发项目均需要首先进行总体设计，它是设计首要步骤，决不允许本末倒置，不能出现先编码后设计的情况，这是软件开发的第二大痛点（第一大是需求不明确、任意变更需求）。

总体设计分为三个阶段：

第一阶段：初始设计。在对给定的数据流图进行复审和精化的基础上，将其转化为初始的模块结构图。

第二阶段：精化设计。依据模块“高内聚低耦合”的原则，精化初始的模块结构图，并设计其中的全局数据结构和每一模块的接口。

第三阶段：设计复审阶段。对前两个阶段得到的高层软件结构进行复审，必要时还可能需要对软件结构做一些精化工作。

概要设计

概要设计的目的是描述系统的每个模块的内部设计，对总体设计和详细设计承担承上启下的作用。

概要设计按照结构化设计方法进行设计。结构化设计方法的基本思路是：按照问题域，将软件逐级细化，分解为不必再分解的模块，每个模块完成一定的功能，为一个或多个父模块服务（即接受调用），也接受一个或多个子模块的服务（即调用子模块）。模块的概念，和编程语言中的子程序或函数是对应的。

概要设计阶段把软件按照一定的原则分解为模块层次，赋予每个模块一定的任务，并确定模块间调用关系和接口。

在这个阶段，设计者会大致考虑并照顾模块的内部实现，但不过多纠缠于此。主要集中于划分模块、分配任务、定义调用关系。模块间的接口与传参在这个阶段要制定得十分细致明确，需要编写严谨的数据字典，避免后续设计产生不解或误解。概要设计一般不是一次就能做到位，而是反复地进行结构调整。典型的调整是合并功能重复的模块，或者进一步分解出可以复用的模块。在概要设计阶段，应最大限度地提取可以重用的模块，建立合理的结构体系，节省后续环节的工作量。

概要设计文档最重要的部分是分层数据流图、结构图、数据字典以及相应的文字说明等。以概要设计文档为依据，各个模块的详细设计就可以并行展开了。

详细设计

详细设计阶段就是依据概要设计阶段的分解，设计每个模块内的算法、流程，为每个模块完成的功能进行具体的描述，要把功能描述转变为精确的、结构化的过程描述。

详细设计这个阶段，各个模块可以分给不同的人去并行设计。设计者的工作对象是一个模块，根据概要设计赋予的局部任务和对外接口，设计并表达出模块的算法、流程、状态转换等内容。这里要注意，如果发现有结构调整（如分解出子模块等）的必要，必须返回到概要设计阶段，将调整反应到概要设计文档中，而不能就地解决，不打招呼。详细设计文档最重要的部分是模块的流程图、状态图、局部变量及相应的文字说明等。一个模块对应一篇详细设计文档。

概要设计阶段通常得到软件结构图，详细设计阶段常用的描述方式有：流程图、N-S图、PAD图、伪代码等。而详细设计的目的是描述某一个模块内部的处理流程、开发方法和编码技巧。一般来说，详细设计由项目简介、模块说明（具体说明每一个模块内部的流程、功能、逻辑、消耗以及未解决问题）、接口设计（包括内部接口和外部接口）、数据结构设计（包括物理结构和逻辑结构）、特殊处理等几个部分构成。软件的详细设计，最终是将软件系统的各个部分的具体设计方法、逻辑、功能采用文字方式进行表述。这样在实现过程中，编码人员原则上严格按此进行代码实现即可。

编写代码

编写代码可以遵循以下几点原则：

先做核心模块的压测：很多程序员，习惯把东西做完，然后等着快上线的时候才做性能测试，那么如果前面设计出了问题，这个就很头大了。当然，后期快上线的时候也要做性能测试，但前期的我认为还是很重要的。当然，做好这一点，需要懂一些业务，你要知道业务压力在哪里，业务请求的重心在哪里，很多时候，产品经理不讲，你也要问清楚。

确保过程可控：代码执行时一定要保持中间的输出，比如说，每处理 10 万条日志，写一条状态日志，记录处理的日志条目数和当前的执行时间。

多打日志：很多时候，代码写的自己也不是很满意，比如某个处理效率不够优化，某个处理的方法不够简洁，或者扩展性比较差，代码写的很弱智，但可能短时间没有办法想清楚最合理的解决方案，考虑到上线初期这里并不是重心所在，所以也不会特意去优化它，但这种情况下我往往会留下注释，并说明下一步优化的可能思路是什么，或者想到的可行方案是什么。

简单易懂的逻辑：千万不要把自己绕进去了，时间一长，谁都看不明白你的逻辑。如果逻辑真的很难在一个函数内完成，尝试切分。

不要沉迷于框架：框架最大的问题是什么？是过于繁冗的嵌套。为什么我一直很烦框架？因为经常遇到需要一秒钟几千次请求的处理场景，那么调优的时候，要从数不清的框架中寻找数据处理的逻辑，寻找性能卡点，可能改动代码只有两行，但是找问题需要两天。程序员记住，你的技术能力绝对不能被框架约束住。

使用熟悉、成熟的技术：很多人根本没搞明白自己的障碍和问题在哪里，根本不知道相关技术产品的优势和劣势在哪里，看一堆第三方的数据测评，脑子一热，去学新技术，然后，掉进坑里出不来，如果是创业公司，可能项目就死在里面了。使用新技术前，建议全面了解该技术的特征，适用范围，以及不适用的范围。

代码审核

众所周知，在团队中进行代码审查（Code Review）可以提升代码质量，分享项目知识、明确责任，最终达到构建更好的软件、更好的团队。

代码审核及其重要，一般来说每周都要做一次代码审核。首先，代码审核有利于你跟踪项目进展情况，我们能真实地看到手下的人进展如何，并且更早发现他们是否误入歧途。有时候，手下人会说“完成得差不多了！”，你去看代码时发现什么都没有或者只是一堆垃圾，诸如此类，总之离完成还很遥远。在管理中，这种情况是最让人讨厌的，所以我认为代码审查是避免这种麻烦的最佳途径。

单元测试

要认识单元测试，首先要明白什么是“单元（Unit）”。所谓“单元”指的是代码调用的最小单位，实际上指的是一个功能块（Function）或者方法（Method）。所以单元测试指的就是对这些代码调用单元的测试。

单元测试是一种白盒测试，就是必须要对单元的代码细节很清楚才能做的测试。所以，单元测试的编写和执行都是由软件工程师来做的。相对于单元测试，还有集成测试。集成测试基本都是黑盒测试，主要是由测试人员根据软件的功能手册来进行测试，需要有专门的测试环境配合。集成测试又分功能测试、回归测试等。

需要单元测试的代码实际上是开发人员自己写的逻辑，测试逻辑所依赖的环境是否正常不是单元测试的目的。在环境访问代码中引入逻辑，只会让逻辑更难测试，导致逻辑代码无法进行单元测试。因此，可单元测试的代码，才能够采用单元测试。判断可测试的代码还有一个方法，就是看这个方法能否用一个 main 函数直接运行，如果可以的话就是可单元测试的代码。可测试的代码还有另一个特征，就是该方法单元的参数，开发人员可以自由模拟，不需要依赖外部环境。

集成测试

集成测试，也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求组装成为子系统或系统，进行集成测试。实践表明，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常的工作。一些局部反映不出来的问题，在全局上很可能暴露出来。

集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。它根据集成测试计划，一边将模块或其他模块组合成越来越大的系统，一边运行该系统，以分析所组成的系统是否正确，各个组成部分是否合拍。集成测试的策略主要有自顶向下和自底向上两种。也可以理解为在软件设计单元、功能模块组装、集成为系统时，对应用系统的各个部件（软件单元、功能模块接口、链接等）进行的联合测试，以决定他们能否在一起共同工作，部件可以是代码块、独立的应用、网络上的客户端或服务端程序。

系统测试

系统测试阶段包括系统测试方案及用例编写、功能性测试、性能测试、稳定性测试。

为了验证需求分析确定的功能是否齐全并被正确实现，同时还要对安装、部署、适应性、安全性、界面等非功能性需求进行测试。系统测试也有测试人员负责，应该在需求分析完成后进行设计，在集成测试完成后进行实施。

功能性测试一般由独立测试小组采用黑盒方式来测试，主要测试系统是否符合“需求规格说明书”。在经过以上各阶段测试确认之后，把系统完整地模拟客户环境来进行的测试。系统测试是将已经确认的软件、计算机硬件、外设、网络等其他元素结合在一起，进行信息系统的各种组装测试和确认测试，其目的是通过与系统的需求相比较，发现所开发的系统与用户需求不符或矛盾的地方，从而提出更加完善的方案。

性能测试验证系统的稳定性和效率，检查系统是否满足规定的性能要求。性能测试通常选择一些典型的功能，检验这些功能在大量用户同时使用系统时系统是否稳定。性能测试由测试人员负责，可以在系统测试完成后进行，也可以对重要模块先进行性能测试，可以贯穿整个测试周期，目的是尽早发现系统的性能瓶颈并提早解决。

稳定性测试和性能测试都必须等到系统基本没问题、趋于稳定时再进行才有效果，否则很难顺利测下去，出现异常也不能定位究竟是系统架构的问题，还是功能上的缺陷。

稳定性测试（亦可称可靠性测试）通过给系统加载一定的业务压力，让系统持续运行一段时间（一般为 7x24 小时），检测系统是否能够稳定运行。

产品发布

产品发布是系统测试结束后的最后一步，通常在软件产品开发过程中不需要产品试制环节，可以直接上线，只需要系统测试员输出系统测试报告并批准产品发布（上线）就可以了。

产品发布前需要通过产品发布说明会形式，对整个产品开发过程从立项开始回溯过程，指出整个过程中的不足点，总结经验，为下一个项目提供经验案例。这一会议可以通过正式会议形式召开，需要召集产品经理、主要开发人员、测试人员、上级领导等参与，准备充分，尽最大可能说清楚这个产品发布之后的效果、效益，为上线后的价值评估做准备。这一环节不可缺少，即便在互联网公司，迭代速度很快的情况下，这一环节也需要满足。

4.技术解决方案

1.营养成分表查询接口

1.1根据食物名称获取营养成分

<https://www.tianapi.com/apiview/121#apicode>

使用营养成分表查询接口查询某种食物的营养成分

名称	类型	必填	示例值/默认值	说明
key	string	是	您自己的APIKEY（ 注册账号 后获得）	api密钥
word	string	是	油条	食品名称、类型名称、营养成分缩写
mode	int	是	0	搜索模式：0营养成分、1食品分类、2营养正序、3营养倒序
num	int	否	10	返回数量，默认10，最大20
page	int	否	1	翻页

返回示例

食品营养成分接口的种类列表和营养成分表

```
1  {
2    "code":200,
3    "msg":"success",
4    "newslist":[
5    {
6      "name":"油条",
7      "rl":386,
8      "las":0.01,
9      "gai":6,
10     "dbz":6.9,
11     "su":0.07,
12     "mei":19,
13     "zf":17.6,
```

```
14     "ys":0.7,
15     "tei":1,
16     "shhf":50.1,
17     "wsfc":0,
18     "meng":0.52,
19     "ssxw":0.9,
20     "wsse":3.19,
21     "xin":0.75,
22     "wssa":0,
23     "dgc":0,
24     "tong":0.19,
25     "lb":2.7,
26     "jia":227,
27     "ling":77,
28     "shc":21.8,
29     "la":585.2,
30     "xi":8.6,
31     "type":"谷类"
32   }]
33 }
```

2.FatsecretPlatform API接口

使用FatsecretPlatform API提供的各种接口

统一请求地址 <https://platform.fatsecret.com/rest/server.api> 请求方式：POST

统一返回数据：添加参数format=json

2.1 根据食物id获取食物的营养成分信息

参数：

```
1  food_id # :食物的id
2
3  method: food.get.v2 api对应的方法
```

响应数据

营养成分

只截取了部分常用数据

```
1  {
```



```
2   "food":
3   {
4     "brand_name": "Sainsbury's", // 品牌名称 (如果id对应的是某个品牌)
5     "food_id": "44234",
6     "food_name": "BGtY Beef Lasagne", // 食物名称
7     "food_type": "Brand", // 食物类型 (Brand--某种品牌, Generic--具体的食物)
8     "servings": {
9       "serving": {
10        "calories": "344", // 卡路里
11        "carbohydrate": "42.10", // 碳水化合物(克)
12        "fat": "8.90", // 脂肪 (克)
13        "fiber": "5.0", // 膳食纤维 (克)
14        "protein": "23.90", // 蛋白质 (克)
15        "sugar": "4.20" // 糖(克)
16      }
17    }
18  }
19 }
```

2.3 根据食物名称搜索

参数

```
1  method: foods.search
2
3  search_expression:搜索表达式 比如^hamburger$ page_number: 页码
4
5  max_results:搜索结果数
```

响应数据

```
1  {
2
3    "foods": {
4
5      "food": {
6
7        "food_description": "Per 3263g - Calories: 2223kcal | Fat: 83.70g | Carbs: 196.81g | Protein: 177.35g",
8
9        "food_id": "4743662",
10
11        "food_name": "Hamburger Soup",
```

```
12
13     "food_type": "Generic",
14
15     "food_url": "https://www.fatsecret.com/calories-
16
17     nutrition/generic/hamburger-soup"
18
19 },
20
21 "max_results": "1",
22
23 "page_number": "1",
24
25 "total_results": "650"
26
27 }
28
29 }
```

3.android中调用api查询接口

使用OkHttp发送请求

项目地址 <https://github.com/square/okhttp>

1.添加依赖

```
1  dependencies {
2
3      ...
4
5      implementation 'com.squareup.okhttp3:okhttp:4.1.0' }
```

2.设置网络访问权限

在AndroidManifest.xml文件中添加

```
1  <uses-permission android:name="android.permission.INTERNET" />
```

3.发送请求

```
1  public func test(){
2
3  // 要开启一个新线程，android4.0以上不能在主线程发网络请求
4      thread {
5
6          val key = "6c658c0160d42b5f5e4807a3dadfee91";
7
8          val word = "油条";
9
10         val mode = 0;
11
12         // 注意OkHttp3会默认使用密文传输，如果使用http协议，会报错
13
14         val apiURL = "https://api.tianapi.com/txapi/nutrient/index?
key=$key&word=$word&mode=$mode";
15
16         try {
17
18             // 1.构建一个httpClient
19
20             val client = OkHttpClient()
21
22             // 2.构建一个request请求
23
24             val request = Request.Builder()
25
26                 .url(apiURL)
27
28                 .build();
29
30             // 发送请求
31
32             val response = client.newCall(request).execute()
33
34             val result = response.body?.string();
35
36             if (result != null) {
37
38                 runOnUiThread(){
39
40                     Log.i("MAIN",result.toString())
41
42                 }
43
44             }
45
```

```
46         } catch (e:Exception) {  
47  
48             Log.e("MAIN",e.toString())  
49  
50         }  
51  
52     }  
53  
54 }
```

使用JSONObject解析响应数据

响应的json字符串

```
{"code":200,"msg":"success","newslst":[{"name":"油条","rl":386,"las":0.01,"gai":6,"dbz":6.9,"su":0.07,"mei":19,"zf":17.6,"ys":0.7,"tei":1,"shhf":50.1
```

解析后

I/Main: name:油条

rl:386

las:0.01

gai:6

dbz:6.9

su:0.07

mei:19|

zf:17.6

ys:0.7

tei:1

shhf:50.1

wsfc:0

meng:0.52

```
1  val jsonObject = JSONObject(result) // 字符串转换为对象
2  // 字符串转换为数组
3  val nutrition =
4  JSONArray(jsonObject.get("newslist").toString()).getJSONObject(0); for (k in nutrition.keys())
5  {
6  Log.i("Main", "$k:${nutrition.get(k)}")
7  }
```

5.推广方案

1.线上渠道

基础上线：各大下载市场、应用商店、大平台、下载站的覆盖Android版本发布渠道

应用商店：geogle商店、小米商城、魅族商店、oppo应用商店、华为应用商店等。

客户端：豌豆荚手机精灵、91手机助手、360软件管家等。

2.线下渠道

采用免费用户通过邀请新用户注册可解锁高级功能的模式

产品运营阶段

A、种子期：主要目的在于收集用户行为数据，与产品设计时的用户模型做对比，有目的性调优。

主要关注数据有：页面路径转化，按钮点击，启动次数，启动时间段，停留时长等。

这个阶段数据量不求大，但求真实，用户来源的话，可以先做免费渠道，如果能有一些首发资源更好了。

B、推广期：主要目的在于扩大影响，吸收用户。

主要关注数据有：新增，活跃，留存以及渠道数据。