# 可行性研究过程

1. 复查系统规模和目标

分析员访问关键人员，仔细阅读和分析有关的材料，以便对问题定义阶段书写的关于规模和目标的报告书进一步复查确认，改正含糊或不确切的叙述，清晰地描述对目标系统的一切限制和约束。这个步骤的工作，实质上是为了确保分析员正在解决的问题确实是要求他解决的问题。

产品目标：为了在当今数据量巨大、选择众多的生活中，提供数字化、智能化的方式辅助用户做出判断，此外，也可以根据用户自身个性化的需求精准推荐。

因此，我们产品的目标是，提供食物推荐。而食物推荐情况分为两种：

一种是在无特殊要求下，软件可以根据用户的历史搜索记录分析用户的饮食习惯或用户选定的tag筛选菜单，辅助用户做出选择

另一种是根据用户设定的特殊要求，如纯素食，热量少，无麸质，非乳制品等进行过滤，从而满足客户的特殊需求。

简而言之，就是面向使用外卖软件的用户，对外卖平台菜品提供基于推荐算法的菜品推荐和基于用户个性化的精准推荐。从而解决用户面对繁多菜品时的选择困难问题和有特殊需求问题。

1. 研究目前正在使用的系统

现有的系统是信息的重要来源。显然，如果目前有一个系统正被人使用，那么这个系统必定能完成某些有用的工作，因此，新的目标系统必须也能完成它的基本功能；另一方面，如果现有的系统是完美无缺的，用户自然不会提出开发新系统的要求，因此，现有的系统必然有某些缺点，新系统必须能解决旧系统中存在的问题。应该仔细阅读分析现有系统的文档资料和使用手册，也要实地考察现有的系统。常见的错误做法是花费过多时间去分析现有的系统。没有一个系统是在“真空”中运行的，绝大多数系统都和其他系统有联系。

目前正在使用的代表系统是饿了么，美团两大外卖平台。两平台均有独有的推荐算法，给予用户以菜品推荐，在一定程度上解决用户的基本选择问题。但根据研究显示，作为商业软件，两平台在进行推荐时不会以菜品本身作为最核心的部分，其外卖推荐算法中主要是把效率最大化的目标与算法目标做了一些关联。基于现有流量及供给，通过深入挖掘、精准匹配及智能化排序机制去实现用户体验和平台效率最大化，是效率最大化的基础上的流量再分配。在商家里，也有部分商家通过好评返现等手段刷好评，迷惑算法。由于这些原因，其推荐算法无法最大程度受益用户。且两平台均没有对用户的特殊需求做出处理，在其排序筛选功能中，无法对诸如过敏因素，素食需求做出筛选。



1. 导出新系统的高层逻辑模型

优秀的设计过程通常是从现有的物理系统出发，导出现有系统的逻辑模型，再参考现有系统的逻辑模型，设想目标系统的逻辑模型，最后根据目标系统的逻辑模型建造新的物理系统。

新系统的逻辑模型：

智能推荐部分：根据用户过往消费记录，根据相似度算法来给出推荐；给每一个菜品标签化，再根据用户的tag筛选列出筛选后的菜品。

特殊需求部分：根据可能存在的用户需求制定tag，同样根据用户筛选给出菜品。

1. 进一步定义问题

目前针对的问题是当下主流的外卖软件、餐饮市场没有对一些拥有特殊需求的用户做出对于推荐菜品的筛选，如生病、过敏等用户，都需要自己手动确认甚至是询问商家才能够得知是否含有自己无法吃的食物成分，同时也没有对用户的饮食偏好进行分类推荐，因此我们需要解决的就是用户的饮食偏好推荐以及屏蔽用户无法吃或无法接受的食物的问题。

6、推荐行动方针（6从5中选择并填入计划书因此5略去）

解决这个问题，首先需要知道如何去分类食物来让用户进行对应筛选。

因此我们采取对各个菜单进行标签化的分类方法；同时我们还能够让用户自己添加tag，并在后台整合这些tag或增加这些tag来丰富我们的标签数据库，这样才能够使菜品能够有足够的分化程度来供用户使用。当有足够的数据集时，我们能够训练一个模型来帮助我们对菜品进行分类，来减少我们对标签分类的人力成本

其次，用户该如何使用。考虑到现在的外卖餐饮大多集中在移动设备上，用户都喜欢在手机上进行餐饮的选择，在手机上点单以及在手机上查看自己想去的参观等，并且大多餐饮店铺也是使用移动设备来进行自身菜品的销售，因此我们应该是针对移动设备来开发对应的应用前端及用户图形界面使用来满足用户的需求。

从操作层面来说，本项目的最终目的为开发一个基于android系统的菜单推荐软件，因此语言及开发软件的选择上需要能够在android系统开发。Java语言能够满足在android平台开发的需求，Android Studio是专门开发安卓软件的平台，因此本项目主体采用Java语言在Android studio上进行开发。菜品模型则采用Python语言，训练对应数据集后产生的模型来对菜品进行分类。用户数据采用MySql数据库进行管理。在评估项目小组现有技术力后，本小组认为在技术要求层面是具有可行性的。

从成本层面来说，训练模型需要一定的成本去完善数据集并测试，且拥有较大的时间成本，但是仍然优于手动筛选标签的人力成本。在维护层面来说，有维护数据库服务器的经济维护成本以及用户数据的维护成本。

1. 草拟开发计划

第一阶段（2023/10/11~2023/11/13）

项目计划书、ppt初稿 10/11~10/24

项目计划书第一版审核 10/17

项目计划书、ppt修改 10/18~10/23

最后审核 10/24

可行性分析 10/25~10/31

可行性分析审核 10/31

需求分析 11/1~11/14

可行性研究报告 11/1~11/6

审核 11/7

审核后再优化 11/7~11/13

最后审核 11/14

第二阶段（2023/11/15~2023/12/26）

总体设计 11/15~11/21

详细设计 11/22~11/28

项目设计 11/29~12/5

项目实现 11/15~12/12

数据收集 11/15~11/19

数据标注11/20~11/26

设计和训练分类模型11/20~11/26

设计和训练推荐模型11/20~11/26

设计前端11/15~11/26

融合和优化11/27~11/30

测试和修改 12/1~12/5

编写代码清单12/6~12/9

编写测试用例12/6~12/9

制作ppt 12/10~12/12

维护 12/12~12/19

项目测试评审 12/20~12/26

第三阶段（2023/12/27~2024/1/16）

总结 12/27~1/9

修改总结、ppt，定稿 1/10~1/16