美食拍-中期总结文档

大型主机专业综合项目

同济大学软件学院 - 2020年6月7日

1753402 谢康

1752079 尤皓

1752124 朴雪

1754133 黄金鑫

1651290 夏宇宁

1754188 谢尚汝

一、引言

1.1 概述

中国是一个幅员辽阔的国家，各个地方的人们创造了灿烂的美食文化，但是目前仍然没有一个简单、方便、热门的美食制作分享的软件，已有的一些软件都是机械的列出菜谱，都没有针对用户的爱好、习惯、区域做一些推荐与过滤。

除此之外，对于热爱美食的烹饪学习者来说，他们想要学习美食制作也面临着信息虚假、过程简略等问题，美食爱好者没有一个合适的平台去学习。

近些年来，互联网在中国快速发展，目前已成为社会经济、人民生活不可或缺的部分；在互联网中，任何一个细分市场都有着巨大的流量，而目前以美食的垂直分享平台仍为空白，这一领域必可大有作为。

1.2 组织目标

本产品“美食拍”垂直分享社交网站主要是提供给广大“吃货”一个吃货交友、美食讨论和美食组团的平台，同时还可以为美食商家提供广告位。产品的潜在客户数量巨大，技术要求简单发展前景广阔。而且产品所需开发时间短，维护简单，可以很好赶上并且满足社会及用户的需求。

在产品投入市场后，占领不成熟的美食社交领域，并且在用户需求下逐渐改善优化产品功能及其性能，在一到三年内，逐步成为功能齐全涉及广泛值得用户信赖的垂直型社交网站。

1.3 产品定义

本产品为面向美食及其所涉及对象的垂直分享社交型微信小程序，为有相同兴趣的用户提供先进科学以及高效的沟通，并且为此类用户提供安全便捷的网络周边环境。

1.4 产品开发背景

（1）随着人民生活水平的改善，越来越多的人开始关注生活质量，”吃得好”变成了大多数人的追求，产品的潜在客户数量巨大，技术要求简单发展前景广阔。

（2）当下媒体社交流行，以美食为契机分享的同时，打开社交功能，适应潮流。

（3）微信的高覆盖率，其小程序的潜在用户也十分巨大，同时开发难度并不太高。

1.5 产品功能与特色

功能：分享推荐、留言评论、点赞收藏、在线聊天、信息检索、论坛、个人信息管理等

特色：本产品可以针对用户所浏览的内容，进行大数据智能推荐信息，可以设置自己喜欢的tag。

1.6 产品范围

适用领域：适用于大多数领域，其中以美食领域及社交领域为核心

主要包含内容： 美食新闻 、美食文化 、食物的做法 、百科 、美食与养生 、广告及好友推送。

二、项目概述

2.1 用户故事

1、用户可以注册并登录账号

2、用户可以关注其他用户

3、用户可以查看自己的浏览历史

4、用户可以创建并发布食谱

5、用户可以删除自己发布的食谱

6、用户可以随机浏览评论

7、用户可以查看食谱的详情

8、用户可以收藏食谱

9、用户可以推荐食谱

10、用户可以分享食谱

11、用户可以查看自己的食谱

12、用户可以查看自己收藏的食谱

13、用户可以对食谱进行评论

14、用户可以删除对食谱的评论

2.2 项目功能

| 编号： | 功能点： | 参与者： |
| --- | --- | --- |
| UC-1 | 用户登录 | 用户 |
| UC-2 | 关注用户 | 用户 |
| UC-3 | 查看浏览历史 | 用户 |
| UC-4 | 创建评论 | 用户 |
| UC-5 | 删除评论 | 用户 |
| UC-6 | 创建食谱 | 用户 |
| UC-7 | 删除食谱 | 用户 |
| UC-8 | 随机浏览食谱 | 用户 |
| UC-9 | 查看食谱详情 | 用户 |
| UC-10 | 收藏食谱 | 用户 |
| UC-11 | 推荐食谱 | 用户 |
| UC-12 | 分享食谱 | 用户 |
| UC-13 | 查看自己的食谱 | 用户 |
| UC-14 | 查看收藏食谱 | 用户 |

2.3 用户和特征

用户：指已注册的账号使用者。

2.4 操作环境

服务端：

* + 1. 操作系统：Linux、macOS、windows
    2. 运行环境：JRE
    3. 数据库：mysql

客户端：

* + 1. 操作系统：Linux、macOS、windows、IOS、android等
    2. 浏览器：Google Chrome、Safari、Edge等主流浏览器

2.5 设计和实现约束

* 1. 开发环境约束

前端框架：官方框架

后端框架：SpringBoot

开发语言：JavaScript、Java

开发工具：Idea、webStorm、vsCode

开发测试浏览器：Google Chrome

* 1. 时间约束

该项目需要在1个月内完成开发，故需合理规划项目开发计划，并完成所需求的主要功能。

* 1. 技术约束

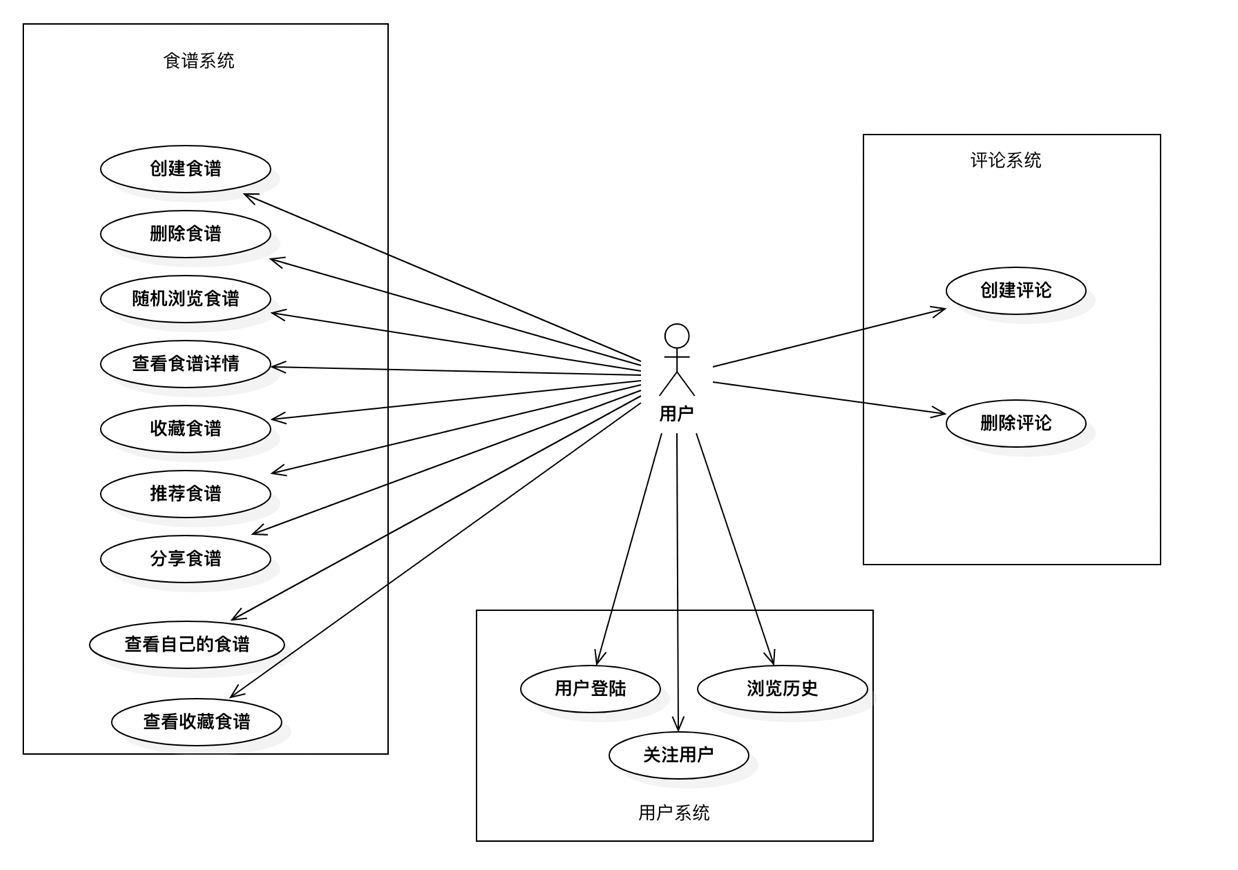
项目成员开发技术掌握不够熟练，故在项目前期，需要对相应的开发技能进行学习并实践，在开发后期，开发技能熟练之后，需要加快开发节奏，完成开发任务。

2.6 假设和依赖性

本项目是否能够成功实施，主要取决于以下条件：

1. 项目规划合理，时间安排符合所有开发人员的需求；
2. 项目难度符合预期，未出现某个开发人员无法解决的需求；
3. 开发任务无特殊情况，如突然加重的课程压力、突然出现的其他项目开发要求等；
4. 开发人员学习能力符合预期，能够熟练掌握相应的开发技能，并按时完成开发要求。

三、需求分析

3.1 功能性需求

3.1.1 登录

|  |
| --- |
| 用例: 登录 |
| ID：UC1 |
| 参与者： 用户 |
| 前置条件: 用户拥有微信账号 |
| 基本事件流**:**  1. 用户点击“登录”按钮  2. 用户调用微信进行快捷登陆 |
| 后置条件：用户可以使用系统提供的所有功能 |

3.1.2 关注用户

|  |
| --- |
| 用例: 关注用户 |
| ID：UC2 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号 |
| 基本事件流**:**   1. 用户点击他人头像查看详情 2. 用户点击关注加入关注列表 3. 用户可以在个人关注列表界面看到关注的人 |
| 后置条件：用户可以从关注列表访问关注的人 |

3.1.3 查看浏览历史

|  |
| --- |
| 用例: 查看浏览历史 |
| ID：UC3 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号 |
| 基本事件流**:**   1. 用户点击浏览历史按钮 2. 显示用户浏览过的历史 |
| 后置条件：用户可以查看历史食谱详情 |

3.1.4 创建评论

|  |
| --- |
| 用例: 创建评论 |
| ID：UC4 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户展开食谱详情界面 |
| 基本事件流**:**   1. 用户点击评论按钮 2. 用户在弹出的文本编辑界面编辑评论 3. 用户点击发送按钮发表评论 |
| 后置条件：用户可以查看自己发布的评论 |

3.1.5 删除评论

|  |
| --- |
| 用例: 删除评论 |
| ID：UC5 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户已发表某条评论 |
| 基本事件流**:**   1. 用户找到自己发表的评论 2. 点击删除按钮删除评论 |
| 后置条件：用户评论被移除 |

3.1.6 创建食谱

|  |
| --- |
| 用例: 创建食谱 |
| ID：UC6 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号 |
| 基本事件流**:**   1. 用户点击“创建食谱”按钮，进入食谱创建界面 2. 用户上传食谱图片并编辑文字 3. 点击“我要发布”按钮发布菜谱 |
| 后置条件：用户成功创建一条食谱 |

3.1.7 删除食谱

|  |
| --- |
| 用例: 删除食谱 |
| ID：UC7 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户已创建过至少一条食谱 |
| 基本事件流**:**   1. 用户点击待删除食谱，进入目的食谱详情界面 2. 用户点击删除食谱按钮 3. 系统弹出“是否确认删除”界面 4. 用户在弹出的“是否确认删除”界面选择是，则系统删除该条食谱；选择否，则系统不删除该条食谱 |
| 后置条件：用户成功删除一条食谱、用户成功取消删除一条食谱 |

3.1.8 随机浏览食谱

|  |
| --- |
| 用例: 随机浏览食谱 |
| ID：UC8 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户自己或其他用户已创建过至少一条食谱 |
| 基本事件流**:**   1. 用户进入“主页”页面 2. 系统检索并以信息流方式显示系统所拥有的所有食谱 3. 用户可以上下滑动信息流浏览系统中的食谱 |
| 后置条件：用户可以在“主页”查看系统中所有用户已发布的食谱信息流 |

3.1.9 查看食谱详情

|  |
| --- |
| 用例: 查看食谱详情 |
| ID：UC9 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户自己或其他用户已创建过至少一条食谱 |
| 基本事件流**:**   1. 用户进入“主页”页面 2. 系统检索并以信息流方式显示系统所拥有的所有食谱 3. 用户可以上下滑动信息流浏览系统中的食谱 4. 用户可以选择他（她）感兴趣的食谱，查看食谱详情 |
| 后置条件：用户可以查看系统中所有用户已发布的食谱详情 |

3.1.10 收藏食谱

|  |
| --- |
| 用例: 收藏食谱 |
| ID：UC10 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户自己或其他用户已创建过至少一条食谱 |
| 基本事件流**:**   1. 用户点击待收藏食谱，进入目的食谱详情界面 2. 若该食谱未被用户收藏过，用户点击收藏食谱按钮，系统将该条食谱加入用户收藏列表；若已被收藏过，用户点击收藏食谱按钮，系统将该条食谱移出用户收藏列表 |
| 后置条件：用户成功收藏一条食谱、用户成功取消收藏一条列表 |

3.1.11 推荐食谱

|  |
| --- |
| 用例: 推荐食谱 |
| ID：UC11 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户自己或其他用户已创建过至少一条食谱 |
| 基本事件流**:**   1. 用户点击待推荐食谱，进入目的食谱详情界面 2. 用户点击“推荐”按钮 |
| 后置条件：用户成功推荐一条食谱 |

3.1.12 分享食谱

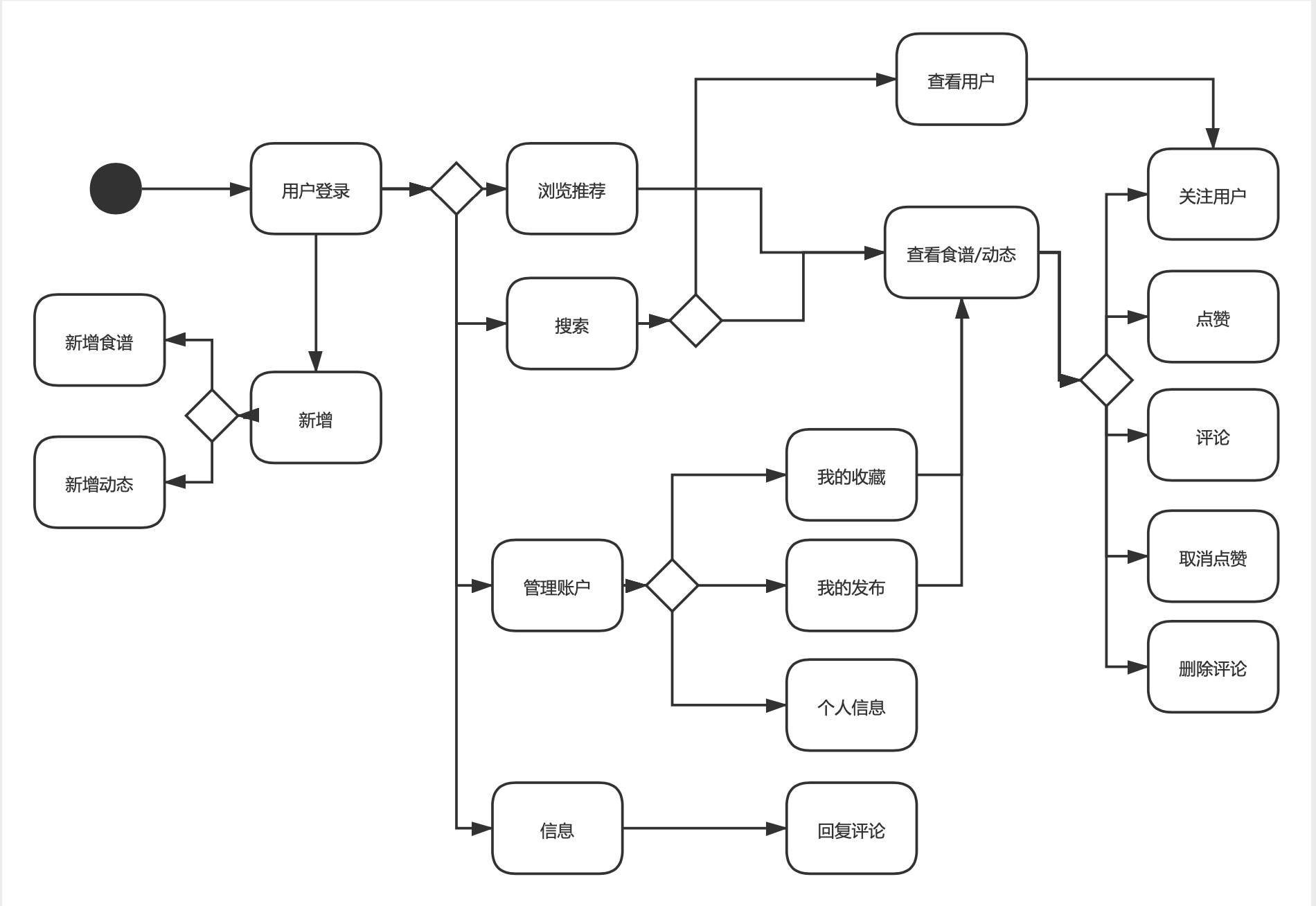
|  |
| --- |
| 用例: 分享食谱 |
| ID：UC12 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户自己或其他用户已创建过至少一条食谱 |
| 基本事件流**:**   1. 用户点击待分享食谱，进入目的食谱详情界面 2. 用户点击“分享”按钮 3. 系统弹出分享界面 4. 用户在分享界面的文字输入框中输入分享理由（可选） 5. 用户点击“确认分享”按钮 |
| 后置条件：用户成功分享一条食谱 |

3.1.13 查看自己的食谱

|  |
| --- |
| 用例: 查看自己的食谱 |
| ID：UC13 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号 |
| 基本事件流**:**   1. 用户点击已发布的食谱，进入相应界面 2. 显示用户已发布的食谱 |
| 后置条件：用户可查看食谱详情 |

3.1.14 查看收藏食谱

|  |
| --- |
| 用例: 查看收藏食谱 |
| ID：UC14 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号 |
| 基本事件流**:**   1. 用户点击已收藏的食谱，进入相应界面 2. 显示已收藏的食谱 |
| 后置条件：用户可查看食谱详情 |

3.2 用户行为

3.3 非功能性需求

3.3.1 易用性

界面风格统一，其友好的用户交互设计，让用户更方便地获取自己想要的信息。

尽量减少用户的学习成本，方便操作，操作流程合理，操作完成时有统一规范的提示信息。

支持没有计算机使用经验、计算机使用经验较少及有较多计算机使用经验的用户均能方便地使用本系统

系统具有一定的容错和抗干扰能力，在非硬件故障或非通讯故障时，系统能够保证正常运行，并有足够的提示信息帮助用户有效正确地完成任务。

系统具有一定的灵活性，可以提示用户如何操作

系统的通用性、操作简单易学，实用性强

网站功能精简，不含冗余恼人的其它功能。

3.3.2 可维护性

可维护性是指在不影响系统其他部分的情况下修改现有系统功能中问题或缺陷的能⼒。开发⼈员创建和设计系统架构时，为了提高系统的可维护性，必须考虑以下几个方⾯的要素：低耦合、高内聚合系统。

本系统将采用严格的软件工程的规范进行开发，并采用良好的设计模式保证系统各模块之间的低耦合 及模块之间的高内聚。

本系统的所有代码将会被详细注释，对于系统所有代码，我们会生成详尽的技术文档。对于系统开发过程可能出现的报错，我们将以文档的方式详细罗列报错码及对应的报错信息。

根据兼职行业的现状，本系统不仅要考虑现有的数据集成和功能集成，还要充分考虑因时新需求导致的对现有功能的新要求。

因而系统的设计方案应易于理解

相关的文档资料应清晰、完整、一致

同时要有良好的功能、数据可扩展性以及对环境适应能力。

3.3.3 可拓展性

系统建成后，应在现行系统上不需要做⼤的改动或不影响整个系统结构，就可以增加功能模块，这就必须在系统设计时留有接口，使其具有可扩展性和维护性，这样就方便在后期的维护过程中根据用户的需求添加相应的功能，同时也不会影响系统其他功能模块的正常运行。

除此之外，还应当保证所有文档说明的详细、完整度，使其他接收者或新加入开发者能够快速地了解系统的情况，并且正确地在原有基础上进行更新和加强。

3.3.4 稳定性

网站能够正常完成所有的功能需求，且保证服务器长时间运行无障碍。故障修复时间一般控制在3小时内，除非出现恶性攻击、病毒破坏、系统资源不足、硬件故障、操作系统崩溃网络中断等情况，不允许出现内存泄露。要求系统可以承受告警风暴和大数据量的冲击，不出现内存溢出等现象。

3.3.5 灾难恢复

即使黑客入侵随意篡改数据，或者发生灾难造成数据丢失，但由于区块链系统中，任一节点的损坏或者失去都会不影响整个系统的运作，并且可在链上取下当前准确数据，因此对数据恢复尤为容易

四、系统模块

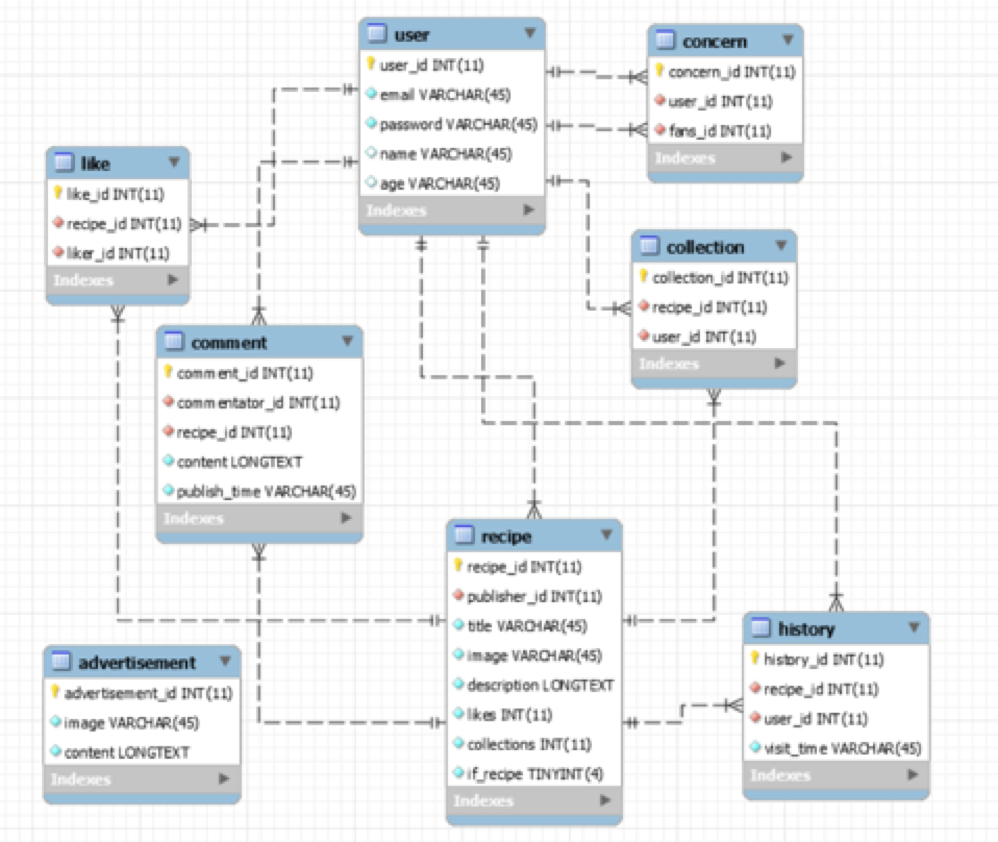
4.1 系统架构

本项目采用了前后端分离的微服务架构，后端接口通过网关进行路由控制，前端通过访问网关api进行数据交换。

4.2 数据库设计

数据库是对系统的全部数据进行有效的存储，实现数据的增、删、查、改等操作，满足各类用户的应用需求。本小节首先通过E-R图抽象出系统的数据结构，以及各数据表之间的联系，然后对支撑业务的一些主要数据表做具体说明。

4.2.1 系统整体E-R图



“美食拍”核心业务E-R图

4.2.2 数据表具体设计

“美食拍”小程序涉及食谱、用户、动态等多个对象，这些对象之间有着不同的关联关系，目前为止，本项目一共设计了8个数据表，下面就是对这些数据表的具体介绍。

（1）用户表。存储本小程序用户的相关信息，包括用户的登录邮箱、密码等，普通用户的账号由个人通过邮箱注册得到。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| USER\_ID | INT(11) | 否 | Y | 用户id，主键，自增 |
| EMAIL | VARCHAR(45) | 否 |  | 用户邮箱 |
| PASSWORD | VARCHAR(45) | 否 |  | 密码 |
| NAME | VARCHAR(45) | 是 |  | 用户昵称 |
| AGE | INT(11) | 是 |  | 年龄 |

（2）食谱表。也是实现小程序业务的核心表。存储发布的食谱信息，包括食谱的图片、标题、介绍以及发布时间等。表中有一标志字段用于判断该发布内容是食谱还是动态。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| RECIPE\_ID | INT(11) | 否 | Y | 食谱id，主键，自增 |
| PUBLISHER\_ID | INT(11) | 否 |  | 发布者ID，外键 |
| TITLE | VARCHAR(45) | 否 |  | 标题 |
| IMAGE | VARCHAR(45) | 否 |  | 食谱配图，存储路径 |
| DESCRIPTION | LONGTEXT | 否 |  | 食谱具体介绍 |
| LIKES | INT(11) | 否 |  | 点赞人数 |
| COLLECTIONS | INT(11) | 否 |  | 收藏人数 |
| IF\_RECIPE | TINYINT(4) | 否 |  | 是否是食谱 |

（3）评论表。该表存储了用户对于某个动态或是食谱的评论，包括评论的食谱id，评论人id，评论内容，发布时间。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| COMMENT\_ID | INT(11) | 否 | Y | 评论id，主键，自增 |
| COMMENTATOR\_ID | INT(11) | 否 |  | 评论人id，外键 |
| RECIPE\_ID | INT(11) | 否 |  | 评论食谱id，外键 |
| CONTENT | LONGTEXT | 否 |  | 评论内容 |
| PUBLISH\_TIME | VARCHAR(45) | 否 |  | 发布时间 |

（4）浏览历史表。该表存放用户的浏览食谱/动态的历史，包括浏览的食谱/动态id，浏览的人的id，访问时间。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| HISTORY\_ID | INT(11) | 否 | Y | 浏览历史id，主键，自增 |
| RECIPE\_ID | INT(11) | 否 |  | 浏览的食谱/动态id，外键 |
| USER\_ID | INT(11) | 否 |  | 浏览的人的id |
| VISIT\_TIME | VARCHAR(45) | 否 |  | 访问时间 |

（5）关注表。本小程序设计了关注功能，一个用户可以被多个用户关注，一个用户也可以关注多个用户。该表存放被关注的用户id，以及该用户的粉丝的id。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| CONCERN\_ID | INT(11) | 否 | Y | 关注id，主键，自增 |
| USER\_ID | INT(11) | 否 |  | 被关注的人的id，外键 |
| FANS\_ID | INT(11) | 否 |  | 关注者的id，外键 |

（6）点赞表。对一个用户所发出的动态或食谱，可以进行点赞。点赞表存放的就是点赞的食谱id以及点赞人的id。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| LIKE\_ID | INT(11) | 否 | Y | 点赞id，主键，自增 |
| RECIPE\_ID | INT(11) | 否 |  | 点赞的食谱id，外键 |
| LIKER\_ID | INT(11) | 否 |  | 点赞者的id，外键 |

（7）对于感兴趣的食谱/动态，用户可以进行收藏操作，一遍以后能够快速找到相应的食谱/动态。收藏表存放收藏的食谱id以及收藏用户的id。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| COLLECTION\_ID | INT(11) | 否 | Y | 收藏id，主键，自增 |
| RECIPE\_ID | INT(11) | 否 |  | 收藏的食谱id，外键 |
| USER\_ID | INT(11) | 否 |  | 收藏者的id，外键 |

（8）考虑到可能会有广告展示等需求，本项目还有一个表用来存取广告的相关信息，该表与其他表之间没有关联的部分。主要存储广告的图片、内容。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | NULL | PK | 说明 |
| ADVERTISEMENT\_ID | INT(11) | 否 | Y | 广告id，主键，自增 |
| IMAGE | VARCHAR(45) | 否 |  | 广告图片 |
| CONTENT | LONGTEXT | 否 |  | 广告内容 |

4.3 接口要求

4.3.1 用户接口

本产品需要用户在用户界面进行相应的操作，当进入主界面后，用户通过点击对应的按钮，分别进入相对应的页面。用户界面应当操作简便，简介而功能齐全，用户友好，并且采用菜单界面驱动方式。

4.3.2 硬件接口

本系统的规模暂时不需要特定的硬件支持，一般情况下通用终端设备都可运行。

4.3.3 软件接口

客户端：

操作系统：Windows、linux、macOS、Android、IOS。

浏览器：IE 10, Google Chrome, Microsoft Edge等。

服务端：

操作系统：Windows、linux、macOS。

Java jdk：1.8及以上版本

数据库：mysql

开发软件：Intellij IDEA

路由：zuul网关。

4.3.4 通信接口

网络通讯协议：HTTP协议。

数据传输协议：TCP/IP协议。

邮件服务协议：SMIP协议。

4.4 接口设计

4.4.1 用户注册

| 用户注册 | | | |
| --- | --- | --- | --- |
| 描述 | 用户使用注册服务 | | |
| 请求方式 | POST | | |
| 请求地址 | /users/register | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| email | String | 用户账户 |
| password | String | 用户密码 |
| name | String | 用户昵称 |
| 返回类型 | json:boolean | | |

4.4.2 用户登录

| 用户登录 | | | |
| --- | --- | --- | --- |
| 描述 | 用户登录账号 | | |
| 请求方式 | GET | | |
| 请求地址 | /users/login | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| email | String | 用户账户 |
| password | String | 用户密码 |
| 返回类型 | json:boolean | | |

4.4.3 更新历史记录

| 更新浏览历史 | | | |
| --- | --- | --- | --- |
| 描述 | 更新用户的浏览历史 | | |
| 请求方式 | POST | | |
| 请求地址 | /users/updateHistory | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| userId | int | 用户id |
| recipeId | Int | 食谱序号 |
| 返回类型 | json:boolean | | |

4.4.4 获取浏览历史

| 获取浏览历史 | | | |
| --- | --- | --- | --- |
| 描述 | 获取用户的浏览历史 | | |
| 请求方式 | GET | | |
| 请求地址 | /users/getHistory | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| userId | int | 用户id |
| 返回类型 | json:List | | |

4.4.5 创建评论

| 创建评论 | | | |
| --- | --- | --- | --- |
| 描述 | 对某一个食谱进行评论 | | |
| 请求方式 | POST | | |
| 请求地址 | /users/comment | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| userId | int | 用户id |
|  | recipeId | Int | 食谱序号 |
|  | content | String | 评论内容 |
| 返回类型 | json:boolean | | |

4.4.6 删除评论

| 删除评论 | | | |
| --- | --- | --- | --- |
| 描述 | 对自己对食谱的评论进行删除 | | |
| 请求方式 | DELETE | | |
| 请求地址 | /users/deleteComment | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| userId | int | 用户id |
|  | recipeId | Int | 食谱序号 |
| 返回类型 | json:boolean | | |

4.4.7 创建食谱

|  |  |  |  |
| --- | --- | --- | --- |
| **创建食谱** | | | |
| **描述** | **用户可以在发布界面创建并发布食谱** | | |
| **请求方式** | **POST** | | |
| **请求地址** | **/recipe** | | |
| **请求参数** | **字段名称** | **类型** | **描述** |
| **publisherId** | **int** | **发布者id** |
| **title** | **String** | **食谱标题** |
| **image** | **String** | **食谱图片** |
| **description** | **String** | **食谱详细描述** |
|  | **likes** | **int** | **点赞个数** |
|  | **collections** | **int** | **收藏人数** |
|  | **If\_recipe** | **boolean** | **是否是食谱** |
| **返回类型** | **Json:boolean** | | |

4.4.8 删除食谱

| **删除食谱** | | | |
| --- | --- | --- | --- |
| 描述 | 用户可以对自己已经发布的食谱进行删除 | | |
| 请求方式 | DELETE | | |
| 请求地址 | /recipe/{recipeId} | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| recipeId | int | 食谱id |
| 返回类型 | Json: boolean | | |

4.4.9 随机显示食谱

| **随机显示食谱** | | | |
| --- | --- | --- | --- |
| 描述 | **用户可以在发布界面创建并发布食谱** | | |
| 请求方式 | **GET** | | |
| 请求地址 | **/recipe** | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| 返回类型 | **Json:List** | | |

4.4.10 查看食谱详情

| **查看食谱详情** | | | |
| --- | --- | --- | --- |
| 描述 | **用户可以点击食谱查看食谱详情** | | |
| 请求方式 | **GET** | | |
| 请求地址 | **/recipe/{recipeId}** | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| **recipeId** | **int** | **发布者id** |
| 返回类型 | **Json:Map** | | |

4.4.11 收藏食谱

| **收藏食谱** | | | |
| --- | --- | --- | --- |
| 描述 | **用户可以收藏自己喜欢的食谱** | | |
| 请求方式 | **POST** | | |
| 请求地址 | **/collections** | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| **UserId** | **int** | 用户**id** |
| **recipeId** | **int** | **收藏的食谱id** |
| 返回类型 | **Json:boolean** | | |

4.4.12 推荐食谱

| **推荐食谱** | | | |
| --- | --- | --- | --- |
| 描述 | **用户对自己喜欢的食谱进行点赞推荐** | | |
| 请求方式 | **POST** | | |
| 请求地址 | **/likes** | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| **UserId** | **int** | 用户**id** |
| **recipeId** | **int** | **点赞的食谱id** |
| 返回类型 | **Json:boolean** | | |

4.4.13 查看食谱

| **查看食谱** | | | |
| --- | --- | --- | --- |
| 描述 | **用户可以查看自己喜欢的食谱** | | |
| 请求方式 | **GET** | | |
| 请求地址 | **/recipe/{userId}** | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| **UserId** | **int** | 用户**id** |
| 返回类型 | **Json:List** | | |

4.4.14 查看收藏的食谱

| **查看收藏的食谱** | | | |
| --- | --- | --- | --- |
| 描述 | **用户可以查看自己收藏的所有食谱** | | |
| 请求方式 | **GET** | | |
| 请求地址 | **/collections/{userId}** | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| **UserId** | **int** | 用户**id** |
| 返回类型 | **Json:List** | | |

4.4.15 删除收藏的食谱

| **删除收藏的食谱** | | | |
| --- | --- | --- | --- |
| 描述 | **用户可以删除自己收藏的食谱** | | |
| 请求方式 | **DELETE** | | |
| 请求地址 | **/collections/{collectionId}** | | |
| 请求参数 | 字段名称 | 类型 | 描述 |
| **collectionId** | **int** | **收藏id** |
| 返回类型 | **Json:List** | | |

五、浏览模块

5.1 模块描述

浏览模块包括了用户所有可以浏览的内容，包括食谱、动态、评论等。浏览模块的所有功能不需要登录即可使用。点击食谱标题可以查看食谱的详情和食谱评论。未登录用户的浏览动态只能查看所有人的动态。

5.2 功能

1）搜索食谱：用户可以通过关键词、时间、类型中的一个或多个条件搜索对应的食谱进行浏览。

2）搜索动态：用户可以通过关键词、时间、关注列表中的一个或多个条件搜索对应的动态进行浏览。

3）查看所有动态：用户可以在动态页面的名为“所有“的子页面查看到所有用户发布的动态。未登录的用户也可以查看到所有动态。

4）查看食谱：用户可以在食谱页面找到自己感兴趣的食谱，并点击进入食谱详情页面，下拉可以浏览食谱全部内容。

5）查看食谱评论：用户可以在食谱内容下方看到所有用户对于该食谱的评论。

5.3 模块详细设计

5.3.1 搜索食谱的详细设计

5.3.1.1 输入项

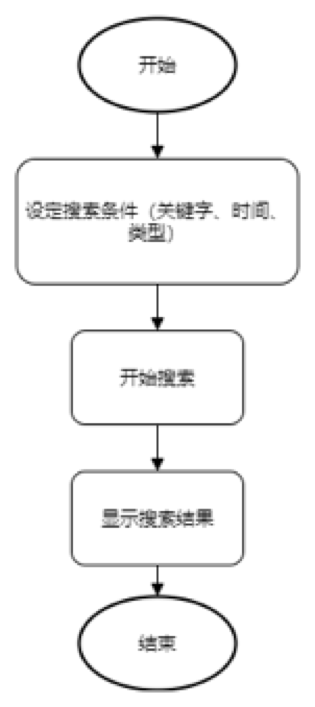
1）搜索食谱类别

2）搜索食谱关键词

3）搜索食谱时间

5.3.1.2 输出项

1）食谱搜索结果的页面

5.3.1.3 逻辑流程

5.3.1.4 限制条件与出错处理

1）限制所有搜索条件都为模式匹配的子字符串。

2）当搜索结果为空时，搜索结果页面显示空白。

5.3.2 搜索动态的详细设计

5.3.2.1 输入项

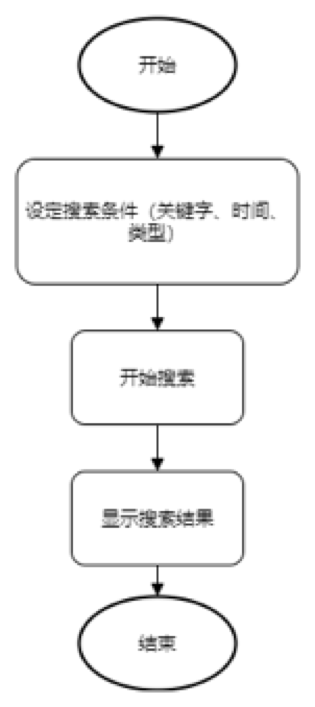
1）搜索动态所属好友

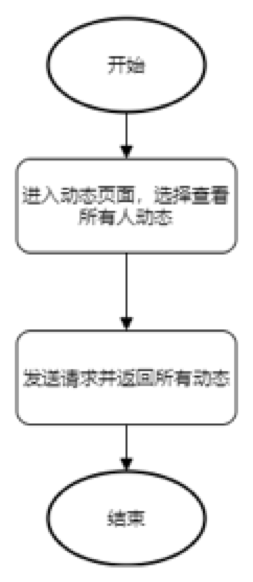
2）搜索动态关键词

3）搜索动态时间

5.3.2.2 输出项

1）动态搜索结果的页面

5.3.2.3 逻辑流程

5.3.2.4 限制条件与出错处理

1）限制所有搜索条件都为模式匹配的子字符串。

2）当搜索结果为空时，搜索结果页面显示空白。

5.3.3 查看所有动态的详细设计

5.3.3.1 输入项

无

5.3.3.2 输出项

1）显示所有动态的页面

5.3.3.3 逻辑流程

5.3.4 查看食谱的详细设计

5.3.4.1 输入项

1）用户所选择的食谱ID

5.3.4.2 输出项

1）显示食谱详情的标题，作者，时间和详细内容的页面

5.3.4.3 逻辑流程

5.3.5 查看食谱评论的详细设计

5.3.5.1 输入项

1）用户所选择的食谱ID

5.3.5.2 输出项

1）显示食谱的评论列表，包括评论作者，时间，评论内容

5.3.5.3 逻辑流程

5.3.5.4 限制条件与出错处理

1）某条食谱无评论时，评论列表为空。

六、社交模块

6.1 模块描述

社交模块包括了用户和其他用户互动的所有功能，例如查看自己或好友的动态，评论好友动态，查看自己的粉丝和关注的用户，关注或取消关注用户，点赞、取消点赞、评论食谱/动态等。尽已登录的用户可以使用社交模块的功能。

6.2 功能

1）查看关注/我的动态：已登录的用户可以在动态页面顶部的动态栏中选择查看自己的动态以及关注的用户发布的动态，若用户未登录则需先进行登录。

2）评论动态：若用户关注了动态的发布者，则用户可以在该动态的详细页面中评论该动态。

3）关注用户：用户可以在动态大厅以及食谱评论中选择自己感兴趣的用户进行关注，关注之后用户成为被关注用户的粉丝。

4）取消关注：如果用户对关注的用户不再感兴趣，或者不想看到该用户发布的动态，可以选择在关注界面取消关注。

5）评论食谱：用户可以在食谱详情页面评论食谱，用户评论某条食谱时，系统将自动将该条评论作为动态发布。该动态还包括该食谱的链接。

6）点赞食谱/动态：用户可以选择一条食谱或动态进行点赞。

7）取消点赞：如果用户对点赞的内容不再感兴趣，可以取消点赞。

6.3 模块详细设计

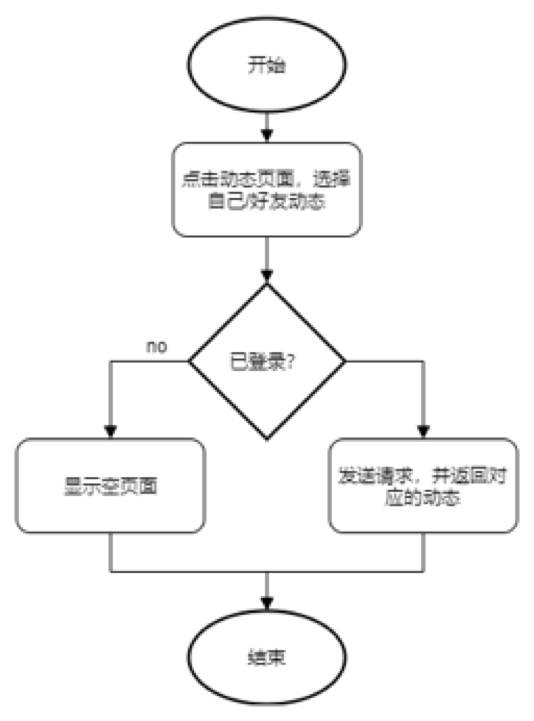
6.3.1 查看关注/我的动态的详细设计

6.3.1.1 输入项

1）用户ID

6.3.1.2 输出项

1）包括该用户自己和所有其关注用户的动态页面

6.3.1.3 逻辑流程

6.3.1.4 限制条件与出错处理

1）限制用户在查看动态前应登录账号

2）如果用户未登录，则动态页面显示为空

6.3.2 评论动态的详细设计

6.3.2.1 输入项

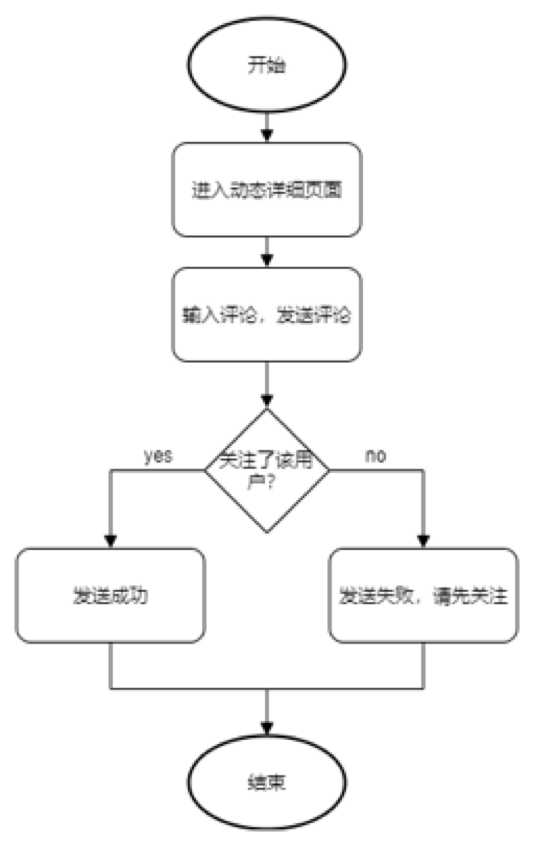
1）用户ID

2）评论的动态ID

3）评论的详细内容

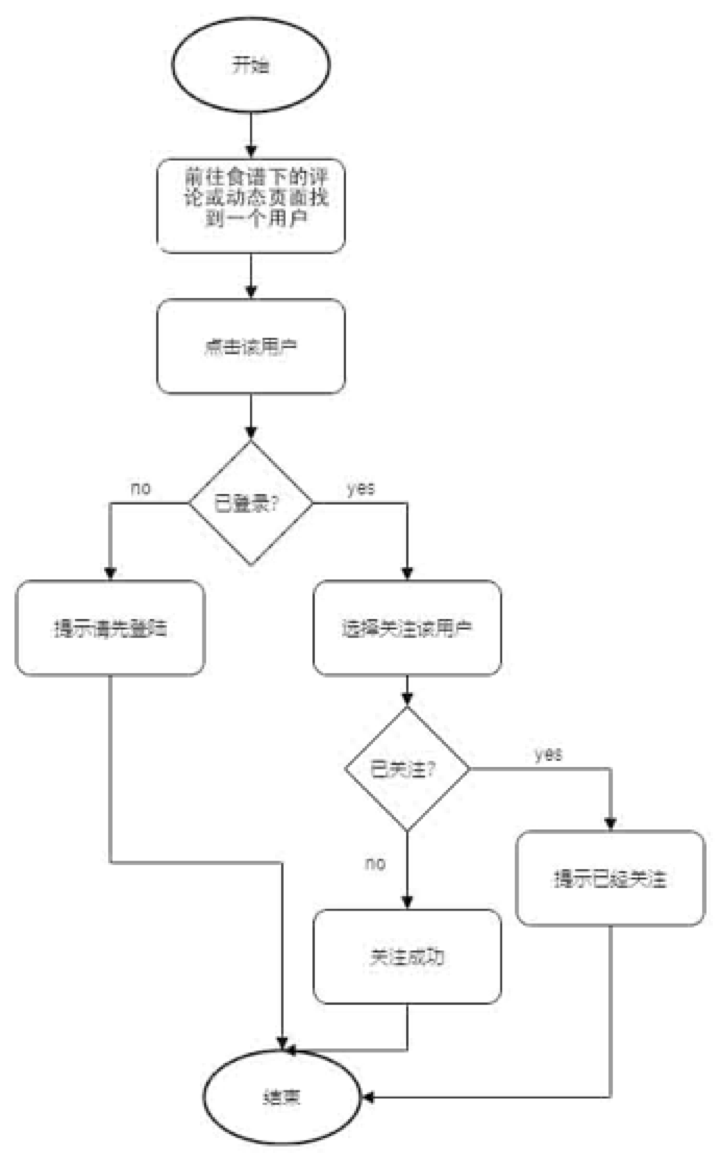
6.3.2.2 输出项

1）评论成功的示信息

6.3.2.3 逻辑流程

6.3.2.4 限制条件与出错处理

1）用户应当在登录后打开动态并对动态进行评论

2）如果用户未登录，则显示请先登录的提示信息

6.3.3 关注用户的详细设计

6.3.3.1 输入项

1）用户ID

2）关注用户ID

6.3.3.2 输出项

1）关注成功的提示信息

6.3.3.3 逻辑流程

6.3.3.4 限制条件与出错处理

1）用户应当在登录后进行关注操作

2）用户不能重复关注同一个用户

3）如果用户未登录，显示请先登录的提示信息

4）如果用户尝试关注一个已关注用户，显示已关注，不要重复点击的提示信息

6.3.4 取消关注的详细设计

6.3.4.1 输入项

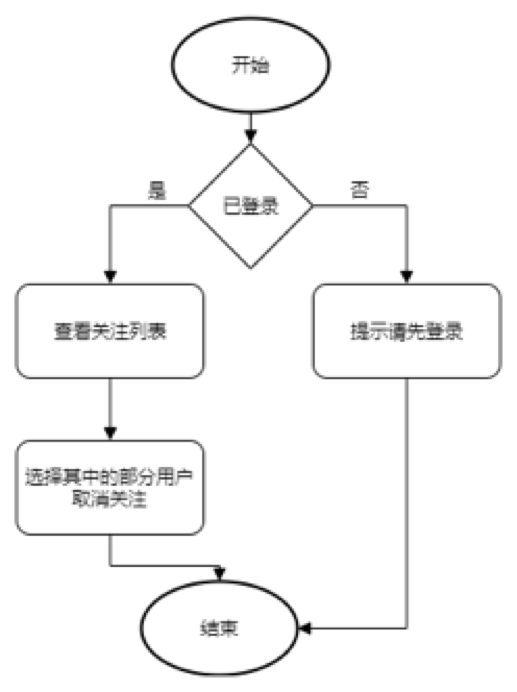
1）用户ID

2）取消关注用户ID

6.3.4.2 输出项

1）取消关注成功的提示信息

2）刷新后的关注列表页面

6.3.4.3 逻辑流程

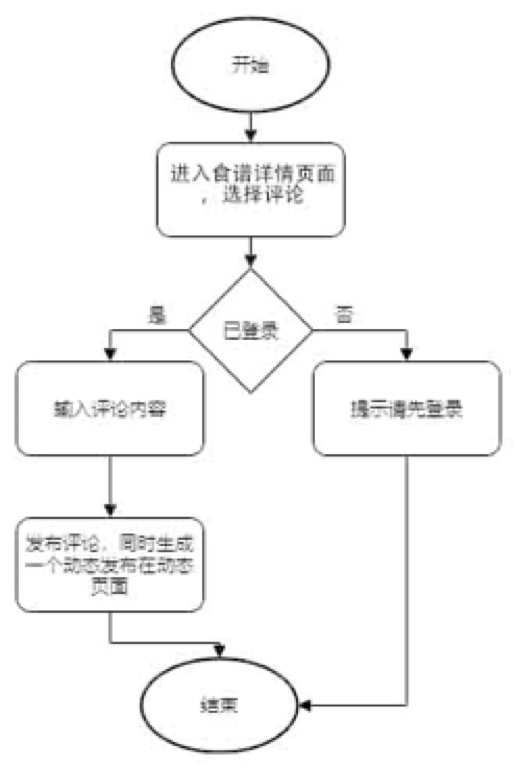
6.3.4.4 限制条件与出错处理

1）用户应当在登录后在关注列表中取消关注

2）如果用户未登录，无法打开关注列表

6.3.5 评论食谱的详细设计

6.3.5.1 输入项

1）用户ID

2）评论食谱ID

3）评论的详细内容

6.3.5.2 输出项

1）评论成功的提示信息

6.3.5.3 逻辑流程

6.3.5.4 限制条件与出错处理

1）用户应当在登录后打开食谱并对食谱进行评论

2）如果用户未登录，则显示请先登录的提示信息

6.3.6 点赞食谱/动态的详细设计

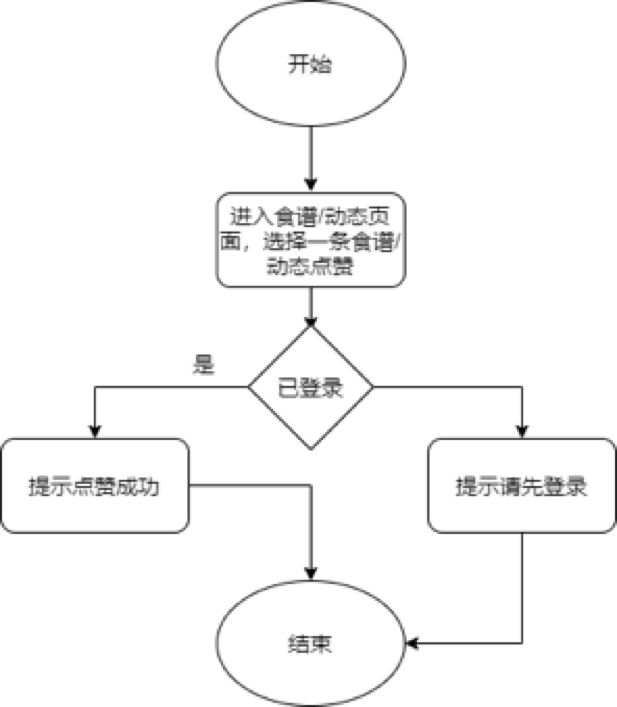
6.3.6.1 输入项

1）用户ID

2）点赞的食谱/动态ID

6.3.6.2 输出项

1）点赞成功的提示信息

6.3.6.3 逻辑流程

6.3.6.4 限制条件与出错处理

1）用户应当在登录后打开食谱/动态并对其进行点赞

2）如果用户未登录，则显示请先登录的提示信息

6.3.7 取消点赞的详细设计

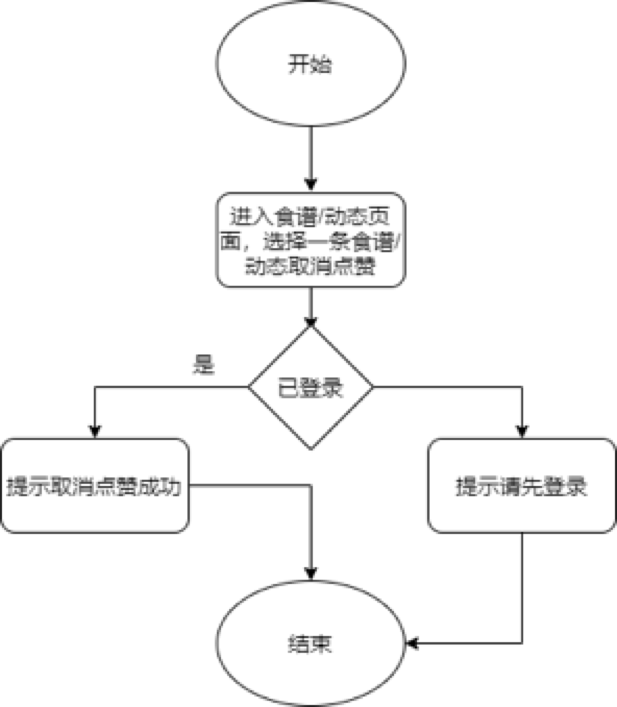
6.3.7.1 输入项

1）用户ID

2）取消点赞的食谱/动态ID

6.3.7.2 输出项

1）取消点赞成功的提示信息

6.3.7.3 逻辑流程

6.3.7.4 限制条件与出错处理

1）用户应在登录后取消点赞

2）如果用户未登录，显示请先登录的的提示信息

七、个人收藏发布模块

7.1 模块描述

个人收藏模块位于导航栏“个人“下。其中包括收藏列表和已发布列表两部分。此外用户对自己感兴趣的内容进行收藏以便后续可以便捷的查看已收藏内容的功能也包括在此模块。用户可以在浏览一条动态(食谱)时对该动态(食谱)进行收藏，可以浏览自己收藏的动态(食谱)的列表，并对其中某条动态(食谱)取消收藏。

7.2 功能

1）收藏动态(食谱)：用户可以在动态(食谱)详情页面收藏自己感兴趣的动态(食谱)，以便下次更快的找到这条动态(食谱)。

2）查看收藏：用户可以查看自己的动态(食谱)收藏列表。

3）取消收藏：用户可以将已收藏的动态(食谱)取消收藏。

4）查看已发布列表：用户可以查看自己发布的的动态(食谱)列表。

7.3 模块详细设计

7.3.1 收藏动态(食谱)的详细设计

7.3.1.1 输入项

1）用户ID

2）收藏动态(食谱)ID

7.3.1.2 输出项

1）收藏成功的提示信息

7.3.1.3 逻辑流程

7.3.1.4 限制条件与出错处理

1）用户应在登录后收藏动态(食谱)

2）用户不可以重复收藏同一条动态(食谱)

3）如果用户未登录，显示请先登录的提示信息

4）如果用户尝试收藏同一条动态(食谱)，显示已收藏，请不要重复点击的提示信息

7.3.2 查看收藏的详细设计

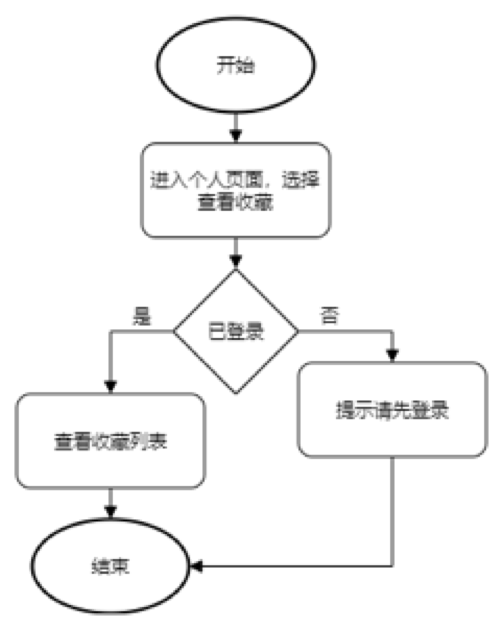
7.3.2.1 输入项

1）用户ID

7.3.2.2 输出项

1）包括用户收藏列表的页面

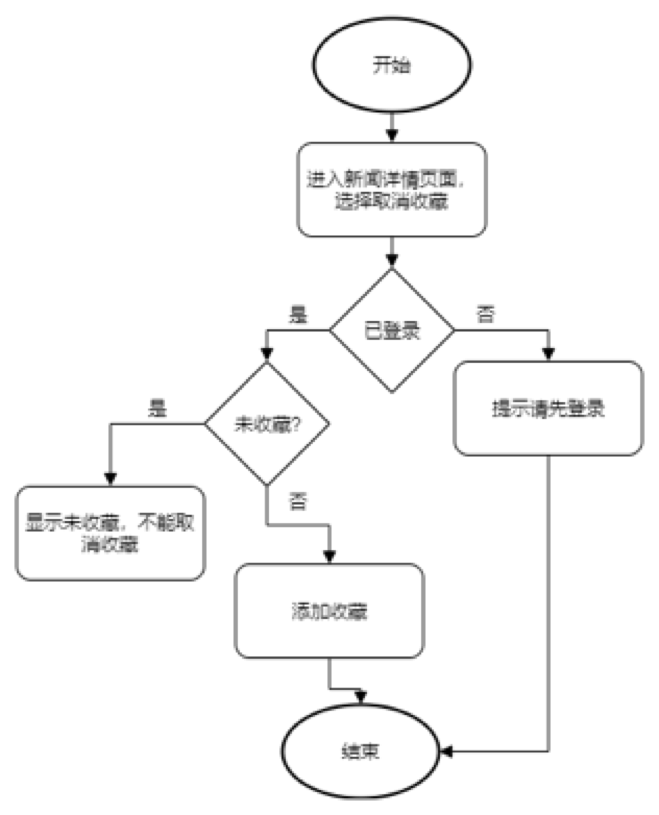
7.3.2.3 逻辑流程



7.3.2.4 限制条件与出错处理

1）用户应在登录后查看收藏

2）如果用户未登录，显示请先登录的提示信息

7.3.3 取消收藏的详细设计

7.3.3.1 输入项

1）用户ID

2）取消收藏的动态(食谱)ID

7.3.3.2 输出项

1）取消收藏成功的提示信息

7.3.3.3 逻辑流程

7.3.3.4 限制条件与出错处理

1）用户应在登录后取消收藏

2）用户不能取消收藏未收藏的动态(食谱)

3）如果用户未登录，显示请先登录的提示信息

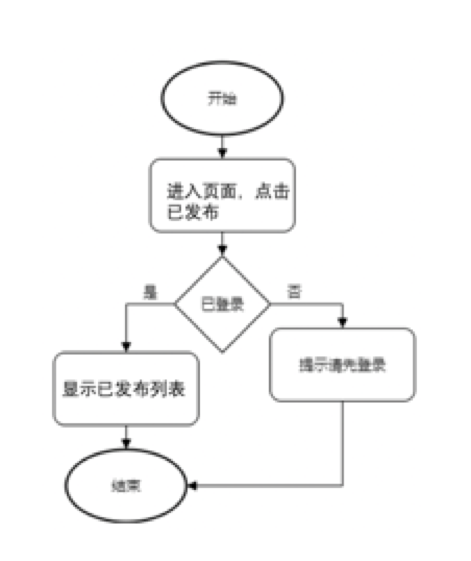
4）如果用户尝试取消收藏一条未收藏的动态(食谱)，显示未收藏，不可取消收藏的提示信息

7.3.4 查看已发布食谱（动态）的详细设计

7.3.4.1 输入项

1）用户ID

7.3.4.2 输出项

1）已经发布的食谱（动态）列表

7.3.4.3 逻辑流程

7.3.4.4 限制条件与出错处理

1）用户应在登录后查看已经发布的动态（食谱）

2）如果用户未登录，显示请先登录的提示信息

八、账户管理模块

8.1 模块描述

账户管理模块包括用户权限的管理功能，如基础的注册登录功能，以及修改个人昵称和密码的功能。

8.2 功能

1）注册：用户可以选择注册一个个人账号。

2）登陆后在微信小程序上与其他用户进行互动。

3）修改昵称：登录之后用户可以修改昵称，此昵称将作为登录账号凭证，并且是唯一的。

4）修改密码：用户可以选择修改密码，首先输入当前密码再将新密码输入两次，两次密码相同则可以进行密码修改。

8.3 模块详细设计

8.3.1 注册的详细设计

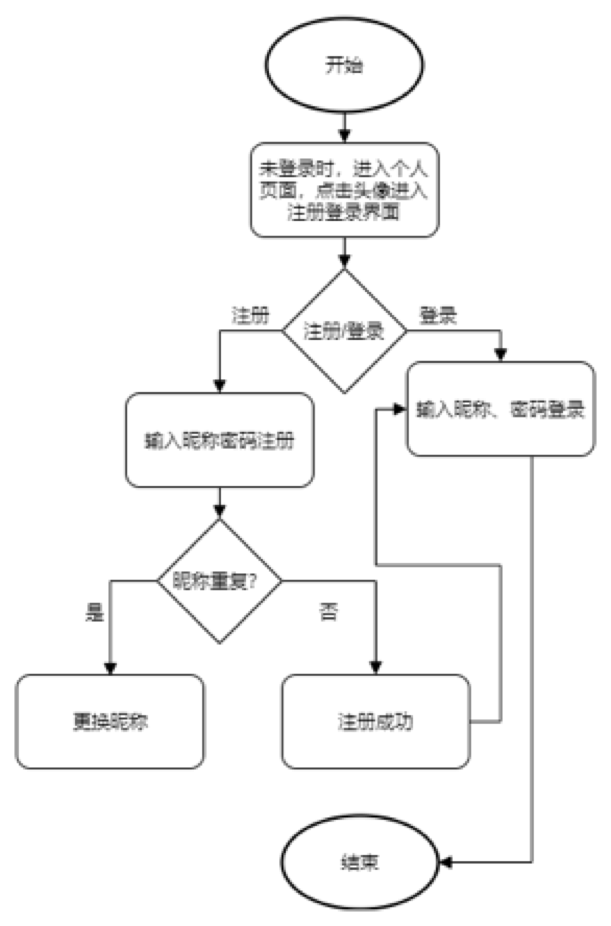
8.3.1.1 输入项

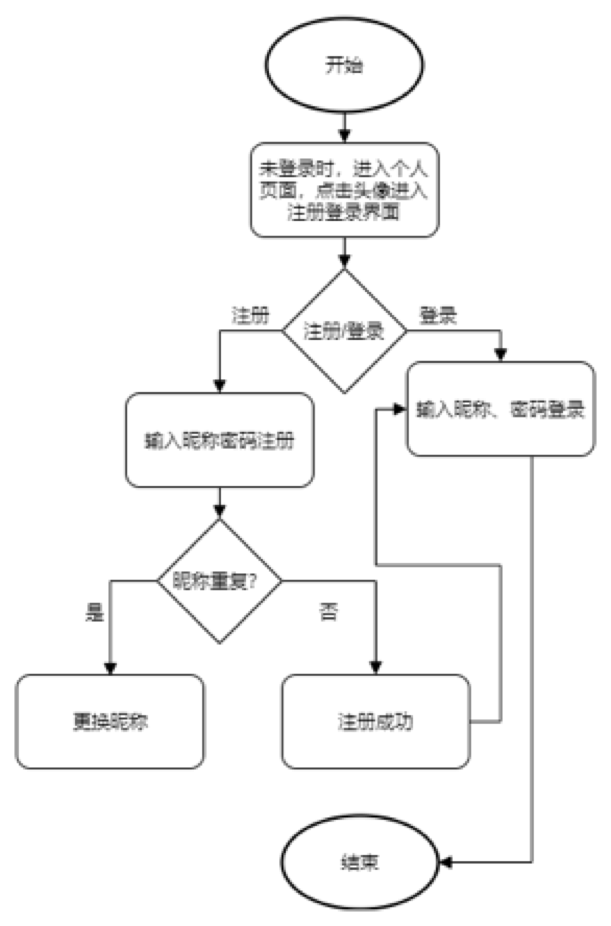
1）注册用户昵称

2）注册用户密码

8.3.1.2 输出项

1）注册成功，请登录的提示信息

8.3.1.3 逻辑流程



8.3.1.4 限制条件与出错处理

1）注册用户昵称不能在数据库中有重复昵称

2）如果用户尝试使用重复的注册昵称，显示昵称重复的提示信息

8.3.2 登录的详细设计

8.3.2.1 输入项

1）登录用户昵称

2）登录用户密码

8.3.2.2 输出项

1）动态(食谱)的浏览页面

8.3.2.3 逻辑流程

8.3.2.4 限制条件与出错处理

1）登录用户的昵称和密码应当匹配

2）如果用户输入的昵称和密码不匹配，显示用户名或密码错误的提示信息

8.3.3 修改昵称的详细设计

8.3.3.1 输入项

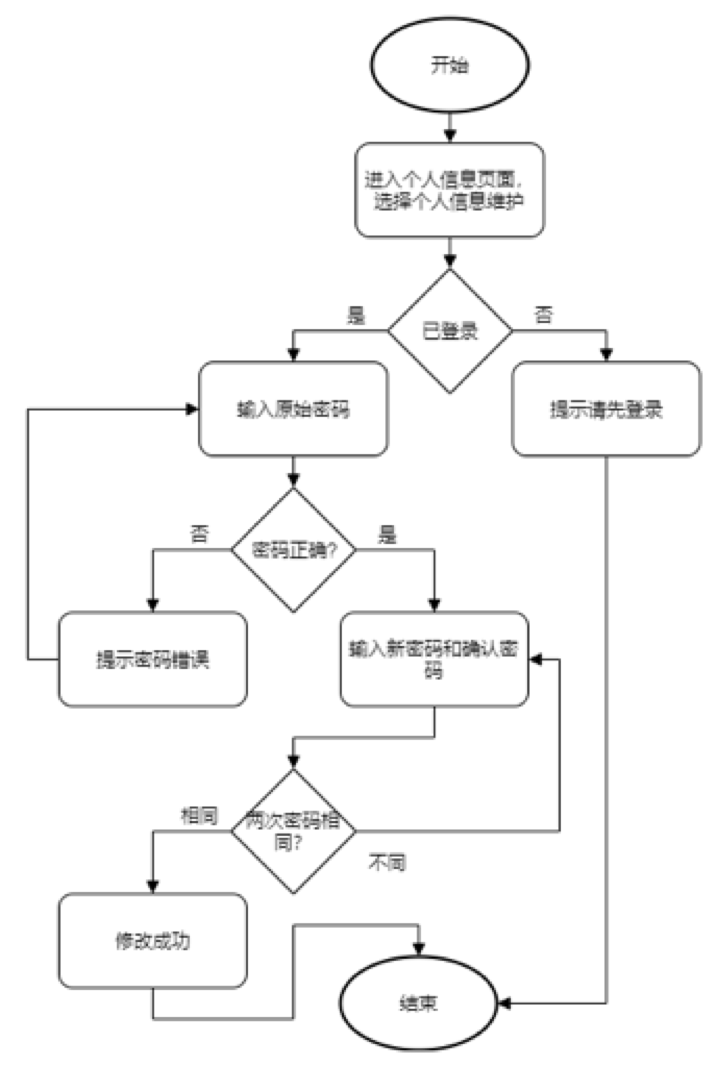
1）用户ID

2）修改的用户昵称

8.3.3.2 输出项

1）修改成功的提示信息

8.3.3.3 逻辑流程



8.3.3.5 限制条件与出错处理

1）修改的用户昵称不能在数据库中有重复昵称

2）如果用户尝试使用一个重复的昵称，显示昵称重复的提示信息

8.3.4 修改密码的详细设计

8.3.4.1 输入项

1）用户ID

2）用户的原始密码

3）修改的用户密码

4）二次输入的修改的密码

8.3.4.2 输出项

1）显示修改成功的提示信息，并自动退出登陆状态，返回动态(食谱)中心

8.3.4.3 逻辑流程

8.3.4.4 限制条件与出错处理

1）用户应当先正确输入原始密码后修改密码

2）用户应当保证两次输入的密码相同

3）如果用户输入了错误的原始密码，显示密码错误的提示信息

4）如果用户两次输入的修改密码不一致，显示两次输入密码不一致的提示信息

九、食谱（动态）模块

9.1 模块描述

该模块负责用户发布一个食谱，发布一个动态，并在后续进行修改和删除操作。

9.2 功能

1）发布动态（食谱）

2）发布后可以选择修改已发布的动态（食谱）。

3）可以删除已经发布的动态（食谱）

9.3 模块详细设计

9.3.1 发布动态（食谱）的详细设计

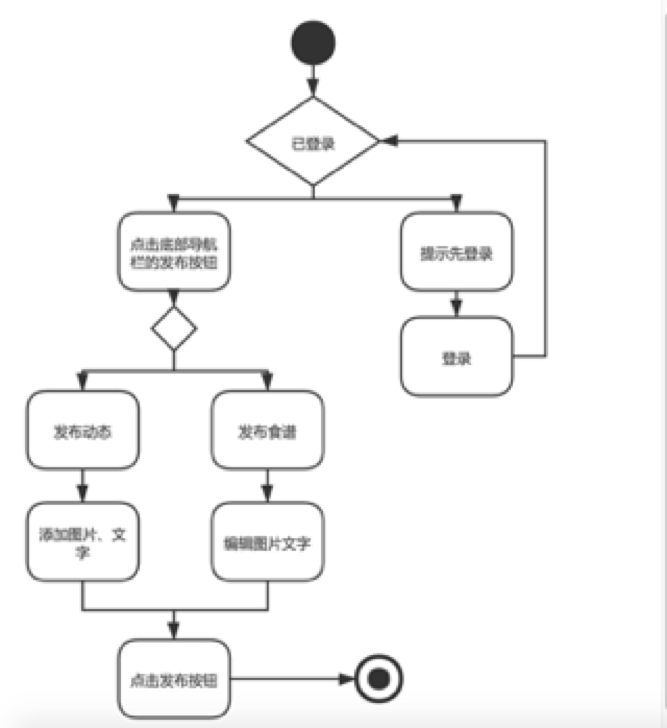
9.3.1.1 输入项

1）图片、文字信息

2）用户ID

9.3.1.2 输出项

1）动态（食谱）ID，success信息

9.3.1.3 逻辑流程

9.3.1.4 限制条件与出错处理

1）必须在登录状态才能发布

2）动态中限制图片数量为最多一张

9.3.2 修改动态（食谱）的详细设计

9.3.1.1 输入项

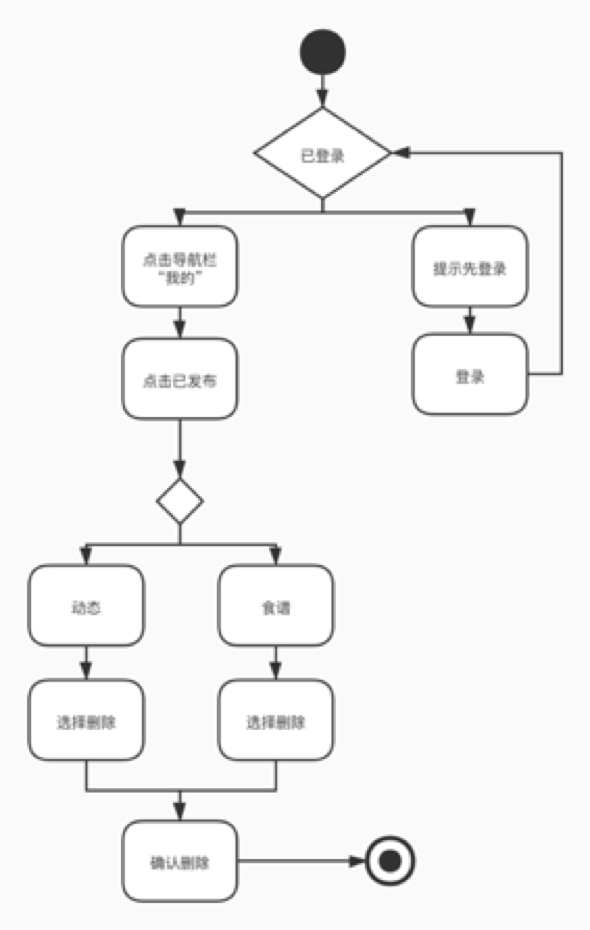
1）动态（食谱）ID

2）用户ID

3）修改后的信息

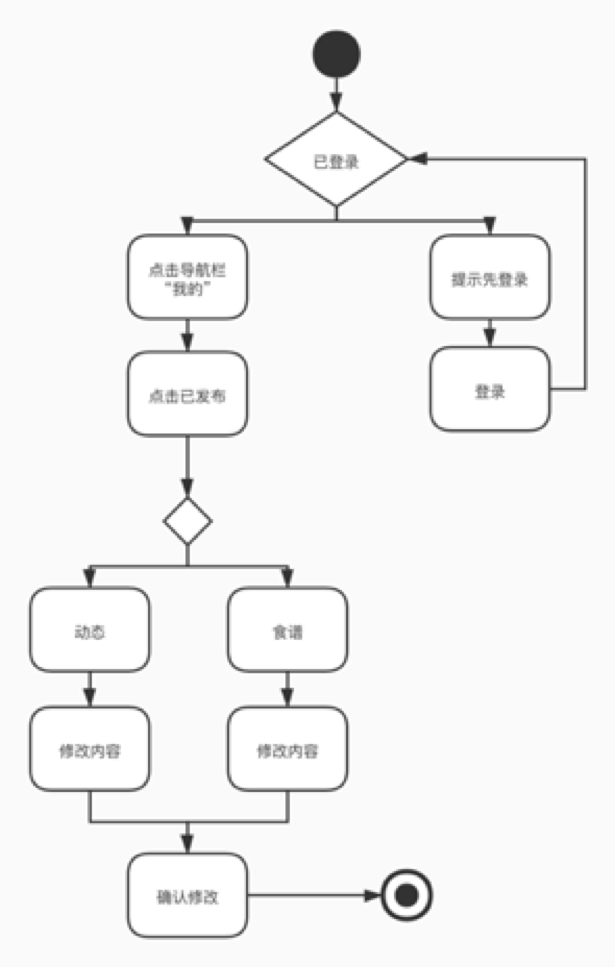
9.3.1.2 输出项

1）修改成功

9.3.1.3 逻辑流程

9.3.1.4 限制条件与出错处理

1）必须在登录状态才能修改



9.3.2 删除动态（食谱）的详细设计

9.3.1.1 输入项

1）图片、文字信息

2）用户ID

9.3.1.2 输出项

1）删除成功

9.3.1.3 逻辑流程

9.3.1.4 限制条件与出错处理

1）必须在登录状态才能删除

2）只能删除本人发布的动态（食谱）