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

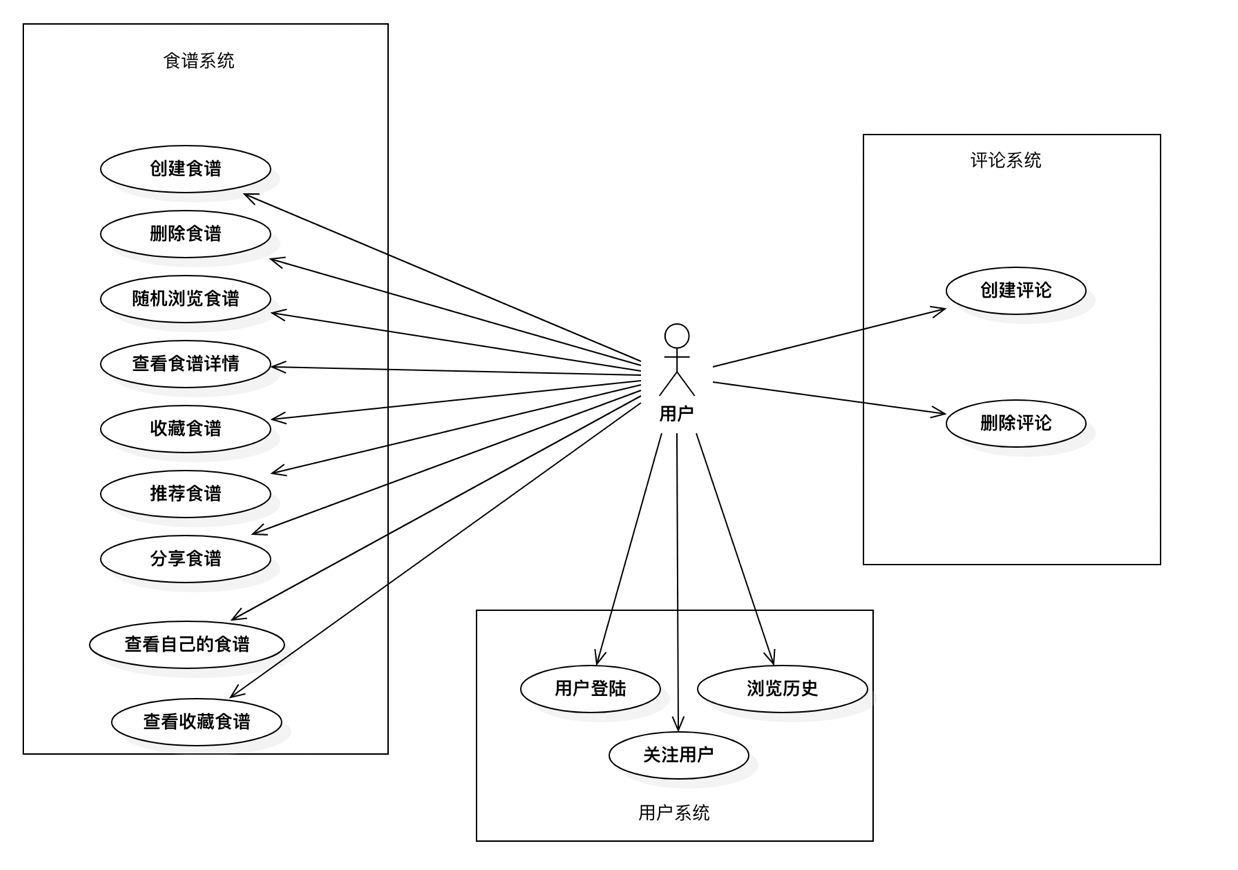
项目成员开发技术掌握不够熟练，故在项目前期，需要对相应的开发技能进行学习并实践，在开发后期，开发技能熟练之后，需要加快开发节奏，完成开发任务。

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

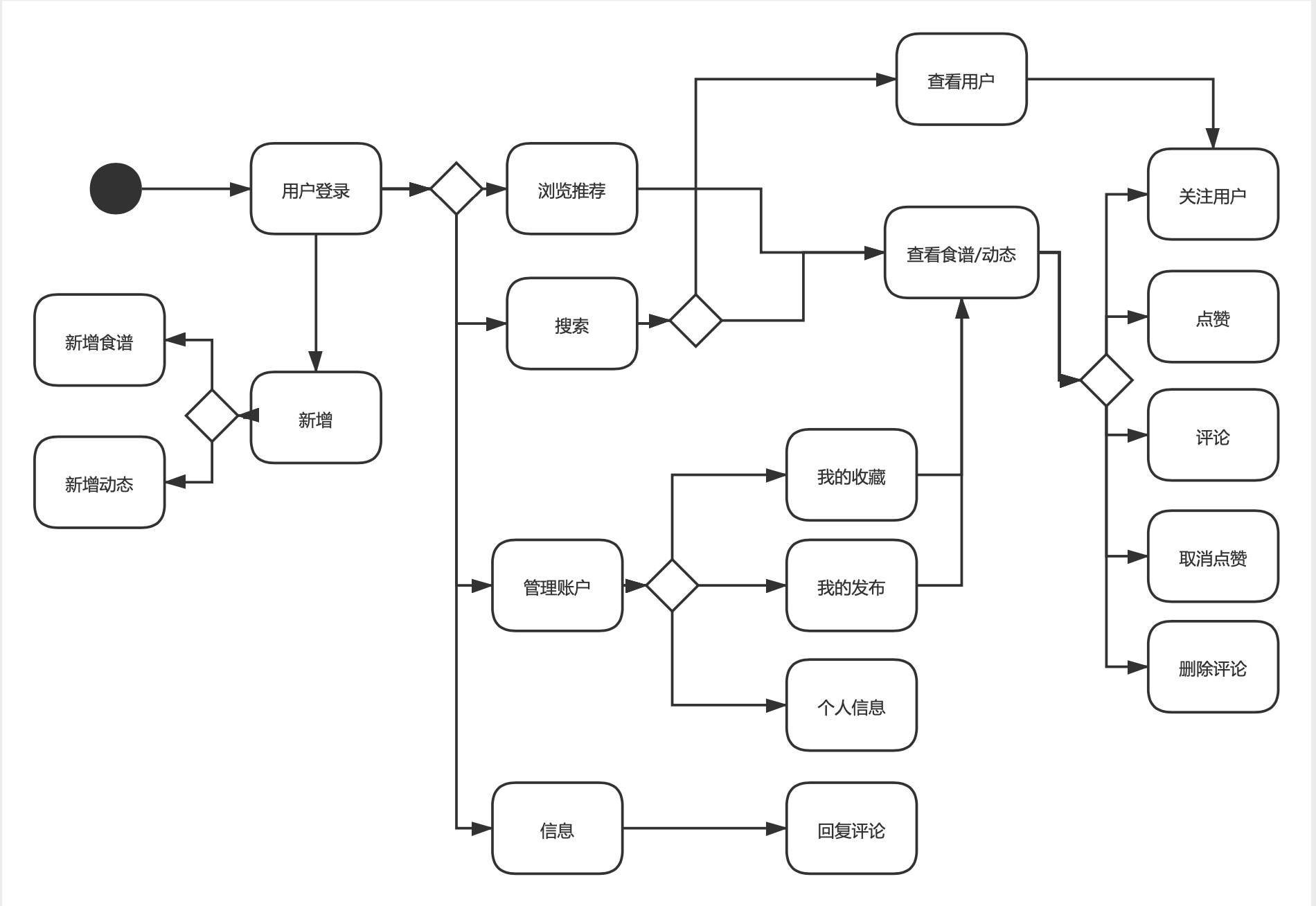
|  |
| --- |
| 用例: 分享食谱 |
| ID：UC12 |
| 参与者： 用户 |
| 前置条件: 用户拥有登录的账号、用户自己或其他用户已创建过至少一条食谱 |
| 基本事件流**:**   1. 用户点击待分享食谱，进入目的食谱详情界面 2. 用户点击“分享”按钮 3. 系统弹出分享界面 4. 用户在分享界面的文字输入框中输入分享理由（可选） 5. 用户点击“确认分享”按钮 |
| 后置条件：用户成功分享一条食谱 |

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

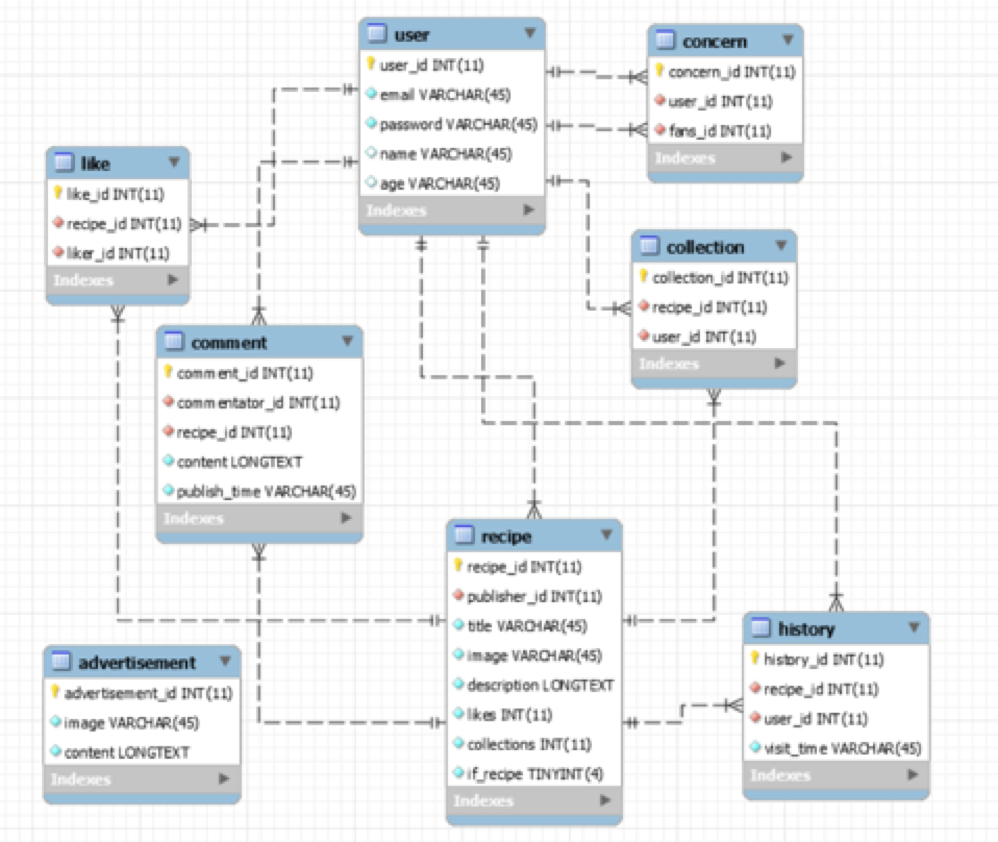
4.1 系统架构

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** | | |