

# 详细设计说明书

## 1 引言

### 1.1 编写目的

《详细设计书》的主要任务是在《概要设计书》基础上，对概要设计中产生的功能模块进行过程描述，设计功能模块的内部细节，包括类的详细逻辑和接口及类之间的调用关系，流程中的逻辑，算法的功能及其实现，性能的需求。为编写源代码提供必要的说明。

本文档的目的旨在使得该 APP 的各个功能模块的设计具体化、规范化，使设计人员对模块设计有清晰的概念和思路并且遵循统一的详细设计书写规范，节省制作文档的时间、软件开发时间，降低 APP 实现的风险，做到 APP 设计的规范性与全面性，以利于该 APP 的实现、测试、维护、版本升级等。

概要设计解决了该 APP 总体结构设计的问题，包括整个 APP 系统的结构、模块划分、模块功能和模块间的联系等。详细设计则要解决如何实现各个模块的内部功能，即模块设计。具体的说，模块设计就是要详细描述各模块中包含的功能的处理逻辑及其提供接口的参数格式、请求方式。但这并不等同于系统实现阶段用具体的语言编码，它只是对实现细节作精确的描述，这样编码阶段就可以将详细设计中对功能实现的描述，直接翻译、转化为用某种程序

设计语言书写的程序。详细设计的详细程度，应达到可以编写程序的水平。

### 1.2 背景与依据

该软件为广大购物爱好者提供了大量的商铺及商品资源，使用户能够直接找到自己所需商品的具体地理位置。同时也为店家用户提供了更好的线上展示机会，从而方便其更好地吸引消费者。

### 1.3 参考资料

- [1]概要设计规格
- [2]数据库系统概念
- [3]第一行代码
- [4]推荐系统算法
- [5]flask web 开发实战
- [6]Android 开发艺术探索

# 客户端：

## 2 程序系统的结构

用一系列图表列出本程序系统内的每个程序（包括每个模块和子程序）的名称、标识符和它们之间的层次结构关系。

## 3 登录（d xp-dengl u）设计说明

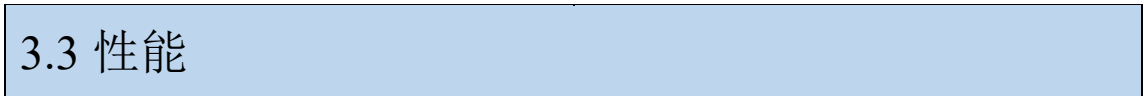
## 3.1 程序描述

实现登录功能，对用户输入的手机以及密码做简要判断



### 3.2 功能

对用户的输入的用户名及密码做简要判断，如手机 11 位，密码大于 6 位，符合即交于后端验证，失败提示错误，成功即登录，并将登录状态保存，下次无需登录。



### 3.3 性能



### 3.4 输入项

手机号	11 位数字
密码	大于 6 位



### 3.5 输出项

错误码	说明
400	手机号未注册或错误
401	密码错误
200	验证成功

## 3.6 算法

匹配算法

## 3.7 接口

```
builder.add("phonenum",string);  
Builder.add("password",string);
```

# 4 登录（dyp-zhuce）设计说明

## 4.1 程序描述

实现注册功能，需要用户输入注册信息，并把信息转换成 json 数据格式，通过 http 发送给后端

## 4.2 功能

## 

### 4.3 性能

### 4.4 输入项

手机号	11 位数字
密码	大于 6 位
性别	男/女
年龄	>1
昵称	不超过 12 个字符

### 4.5 输出项

错误码	说明
400	手机号已注册
200	注册成功

### 4.6 算法

匹配算法

### 4.7 接口



```
builder.add("onenumber",int);  
Builder.add("password",string); builder.add("sex",string);  
Builder.add("username",string);
```

## 5 定位功能（dxp-dingwei）设计说明

### 5.1 程序描述

实现定位功能获取用户当前所在经纬度

### 5.2 功能

用户第一次进入，后台直接运行定位，并保存经纬度。之后可以选择手动定位

### 5.3 性能

### 5.4 输入项

自动或者用户按定位按钮



## 5.5 输出项

用户所在的经纬度，以及城市

## 5.6 算法

调用高德 sdk

## 5.7 接口

```
amapLocation.getLatitude();//获取纬度  
amapLocation.getLongitude();//获取经  
度
```

# 6 推荐功能（dxp-tuijian）设计说明

## 6.1 程序描述

根据所收集到的用户个人以及和用户类似的用户兴趣数据以及所设置的商品的商标推送商品

## 6.2 功能

用户第一次进入，会在“猜我喜欢”这一栏看到被推送的商品

## 6.3 性能

由于本系统仅采用协同过滤推荐系统，而且加上自身水平的有限，所以会导致推送的内容不太准确，不过能符合基本需求。

## 6.4 输入项

用户第一次进入，后台刷新，当用户下拉时再次刷新

## 6.5 输出项

推送用户可能喜欢的商品

## 6.6 算法

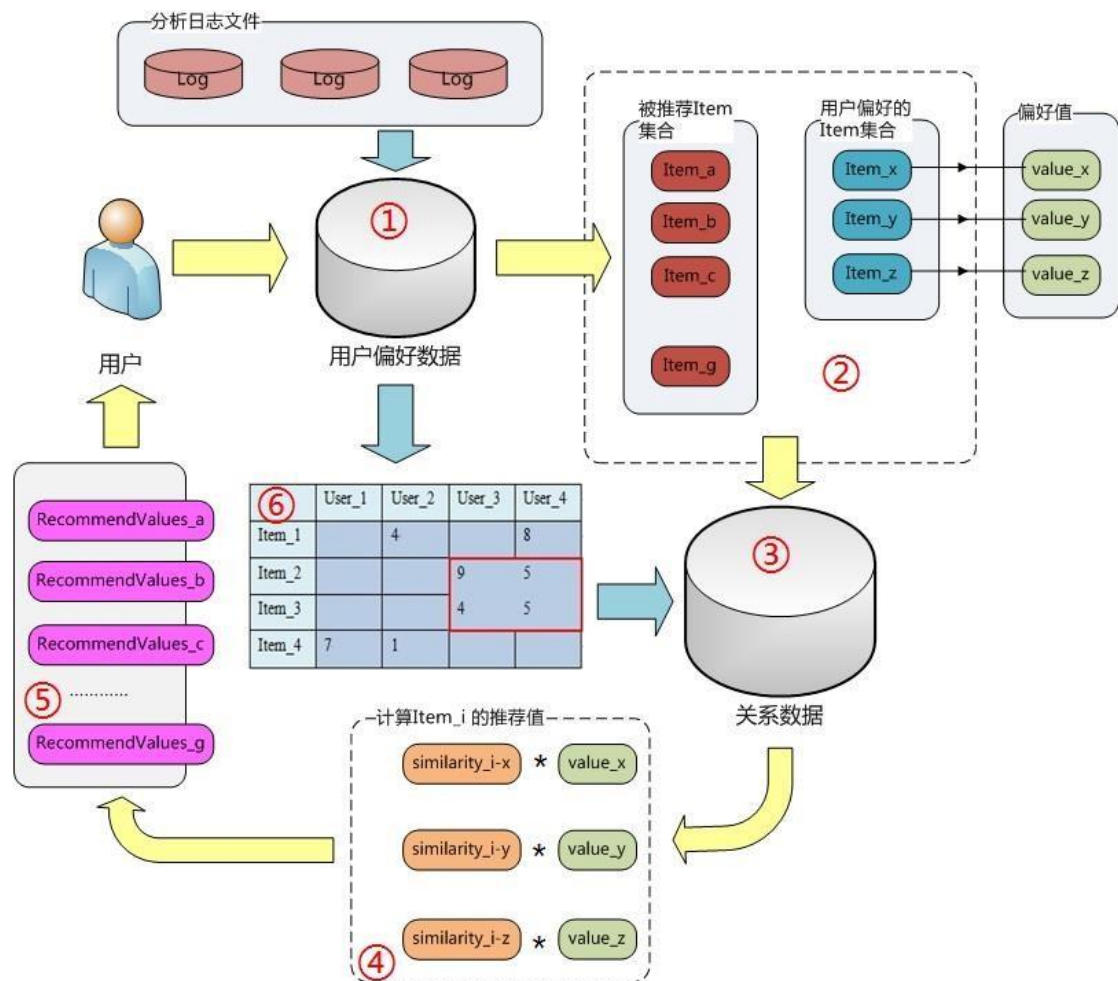
注：以下算法参考自：<http://my.oschina.net/BreathL/blog/62519>

推荐中的协同过滤算法简单说明下：

首先，通过分析用户的偏好行为，来挖掘出里面物品与物品、或人与人之间的关联。其次，通过对这些关联的关系做一定的运算，得出人与物品间喜欢程度的猜测，即推荐值。

最后，将推荐值高的物品推送给特定的人，以完成一次推荐。

协同过滤算法大致可分为两类，基于物品的与基于用户的；区分很简单，根据上面的逻辑，若你挖掘的关系是物品与物品间的，就是基于物品的协同过滤算法，若你挖掘的关系是用户与用户间的，就是基于用户的协同过滤算法；由于它们实现是有所不同，所以我分开整理，先来看看基于物品的协同过滤实现，我自己画了一幅图：



我通过数字的顺序，来标示数据变化的方向（由小到大）；下面分析下每一个步骤的功能以及实现。

首先，说明下两个大的数据源，**用户偏好数据**：UserID、ItemID、Preference：表示一个对一个物品的喜好程度；**关系数据**：ItemIDA(UserIDA)、ItemIDB(UserIDB)、Similarity：表示两个人或物品间的相似程度；接着一个用户来了，我们需要为其推荐，得拿到他的身份标示，一般是 UserID，于是：

①. 查找这个用户喜欢过的物品（即偏好的产品，并查出偏好值后面会用），以及还没有喜欢过的商品，前者是推荐运算的根据，后者作为一个产生推荐的一个集合；如②画的那样。

②. 这里是一个可扩展的地方（我自己理解）；因为这两部分的数据的作用非常明显，修改这两个集合对后面产生的推荐结果可产生非常直观的影响，比如清洗过滤，或根据用户属性缩小集合；不仅使后面推荐效果更优，运算性能也可以大幅度提高。

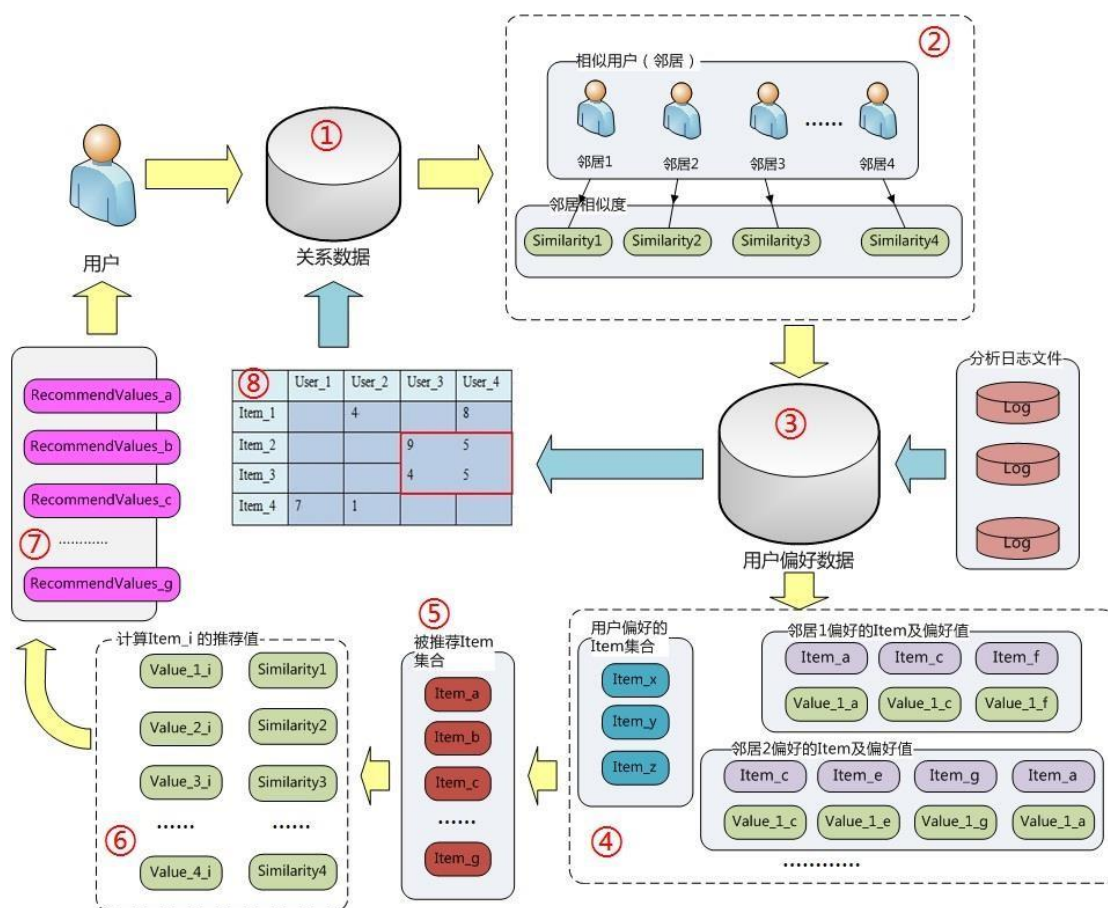
③. 查找这两个集合之间的关系，这是一对多的关系：一个没有偏好过的物品与该用户所有偏好过的物品间的关系，有一个值来衡量这个关系叫相似度 Similarity；这个关系怎么来的，看蓝色箭头的指向。步骤⑥

④. 得到这个一对多的关系后，就可以计算这个物品对于这个用户的推荐值了，图中  $\text{similarity}_{i-x}$  表示  $\text{Item}_i$  与  $\text{Item}_x$  之间的相似度， $\text{Item}_x$  是该用户偏好过的，该用户对其偏好值记为  $\text{value}_x$ ，相乘； $\text{Item}_i$  与该用户偏好过的所有物品以此做以上运算后，得到的值取平均值 便是  $\text{Item}_i$  的推荐值了。注：有可能  $\text{Item}_i$  不是与所有 该用户偏好过的物品都存在相似性，不存在的，不计算即可；另外这里方便理解介绍的都是最简单的实现；你也可以考一些复杂的数学元素，比如方差来判断离散性等。

⑤. 这步就简单多了，刚才对该用户没有偏好过的集合中的所有 Item 都计算了推荐值，这里就会得到一个 list，按推荐值由大到小排序，返回前面的一个子集即可。

⑥. 前面已经提到，关系数据时怎么来的，也是根据用户的偏好数据；你把其看成一个矩阵，横着看过来，参考两个 Item 间的共同用户，以及共同用户的偏好的值的接近度；这里的可选择的相似度算法很多，不一——介绍了，前面提到的 IBM 博客也详细讲解了。

基于物品的协同过滤算法分析完了，下面是基于用户的协同过滤算法，还是自己画了一幅图：



①. 同样也是查询，只是查询的对象不一样了，查询的是与该用户相似的用户，所以一来直接查了关系数据源。以及相似用户与该用户的相似度。

②. 与刚才类似，也是对数据集的一个优化，不过作用可能没那么大。（个人感觉）

③. 查询关系数据源，得到相似用户即邻居偏好过的物品；如步骤④；图中由于空间小，没有把所有邻居的偏好关系都列出来，用.....表示。其次还要得到该用户偏好过的物品集合。

- ④. 被推荐的 Item 集合是由该用户的所有邻居的偏好过的物品的并集，同时再去掉该用户自己偏好过的物品。作用就是得到你的相似用户喜欢的物品，而你还没喜欢过的。
- ⑤. 集合优化同基于物品的协同过滤算法的步骤②。
- ⑥. 也是对应类似的，依次计算被推荐集合中 Item\_i 的推荐值，计算的方式略有不同，Value\_1\_i 表示邻居 1 对，Item\_i 的偏好值，乘以该用户与邻居 1 的相似度 Similarity1；若某个邻居对 Item\_i 偏好过，就重复上述运算，然后取平均值；得到 Item\_i 的推荐值。
- ⑦、⑧. 与上一个算法的最后两部完全类似，只是步骤 ⑧你竖着看，判断两个用户相似的法子和判断两个物品相似的法子一样。

**由于以上实现设计到数据挖掘等高深技术，所以仅是参考，我们用 java 写了小型的实现，可见源代码**

## 7 高斯模糊功能（dxp-gaosimohu）设计说明

### 7.1 程序描述

现在越来越多的开发者喜欢在自定义控件的时候加上各种模糊背景，我也非常这种设计。比如说在个人中心加入如下背景模糊效果：



## 7.2 功能

用户可选择背景图片，之后通过算法将其模糊

## 7.3 性能

Android 中渲染一帧的时间应该不超过 16ms (60fps)，但如果在 UI 线程中做模糊处理就会让帧率降到了 17fps。所以由于算法不够优化还是会有些许延迟。

## 7.4 输入项

用户选择图片

## 7.5 输出项

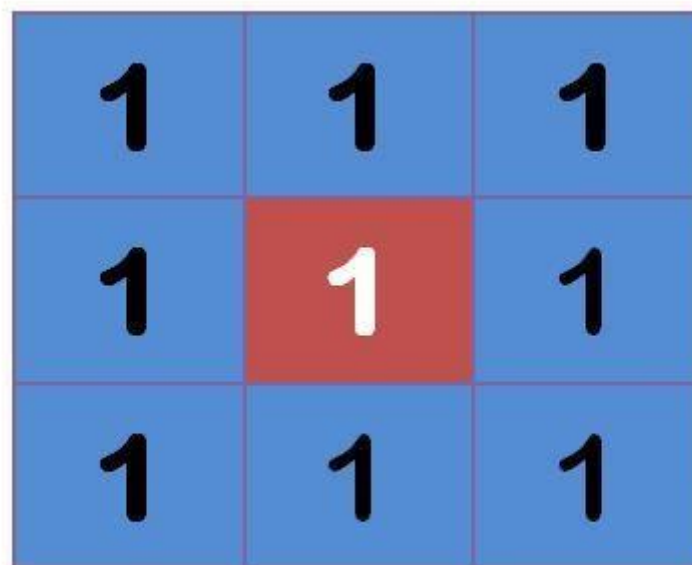
模糊后的图片

## 7.6 算法

一、高斯模糊的原理 所谓"模糊",可以理解成每一个像素都取周边像素的平均值。



上图中，2 是中间点，周边点都是 1。



"中间点"取"周围点"的平均值，就会变成 1。在数值上，这是一种"平滑化"。在图形上，就相当于产生"模糊"效果，"中间点"失去细节。

以上最简单的均值模糊,这种模糊对细节的保留不够,图片模糊后显得很僵硬,不美观,不符合要求,就直接舍弃了,接下来选择了高斯模糊,这种算法的思想是离模糊中心像素越近的元素对该模糊后的像素影响越大,所以应占据更大的比重,越远的像素占据的比重越小，所以我们一般采用高斯函数模糊：

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

用此函数计算权重：



1.32638	1.77477	1.51587
2.83963	3.69403	3.07627
3.22121	4.14113	3.4107

具体实现可参考源代码。

## 7.7 接口

```
Bitmap bitmap = blurBitmap(getBitmapFromDrawable(), this);
```

## 8 商家商品热度排名功能（dxp-hotrank）设计说明

## 8.1 程序描述

通过用户点击商品的次数来保存其热度，并通过热度排名展示给用户

## 8.2 功能

通过热度将商品排序

## 8.3 性能

采用快排时间复杂度  $O(n \log n)$

## 8.4 输入项

商家输入商品，系统记录用户点击过商品

## 8.5 输出项

排列过得商品

## 8.6 算法

商家输入商品-----》用户点击过商品----》此商品热度+1-----》所以商品通过热度快排-----》展示

## 9 图片加载及显示功能(dxp-fresco)设计说明

### 9.1 程序描述

由于本程序是一款可供用户在线浏览附近商家商品的软件，所以图片加载就显得格外重要，尤其是在这么多图片的情况下，处理图片的内存管理也十分重要。本软件采用的图片加载框架为 Facebook 开发的开源图片加载框架 fresco。

### 9.2 功能

#### 内存管理

解压后的图片，即 *Android* 中的 *Bitmap*，占用大量的内存。大的内存占用势必引发更加频繁的 *GC*。在 *5.0* 以下，*GC* 将会显著地引发界面卡顿。

在 *5.0* 以下系统，*Fresco* 将图片放到一个特别的内存区域。当然，在图片不显示的时候，占用的内存会自动被释放。这会使得 *APP* 更加流畅，减少因图片内存占用而引发的 *OOM*。

*Fresco* 在低端机器上表现一样出色，你再也不用因图片内存占用而思前想后。

## 图片的渐进式呈现

渐进式的 *JPEG* 图片格式已经流行数年了，渐进式图片格式先呈现大致的图片轮廓，然后随着图片下载的继续，呈现逐渐清晰的图片，这对于移动设备，尤其是慢网络有极大的利好，可带来更好的用户体验。

*Android* 本身的图片库不支持此格式，但是 *Fresco* 支持。使用时，和往常一样，仅仅需要提供一个图片的 *URI* 即可，剩下的事情，*Fresco* 会处理。

*Gif* 图和 *WebP* 格式

是的，支持加载 *Gif* 图，支持 *WebP* 格式。

## 图像的呈现

*Fresco* 的 *Drawees* 设计，带来一些有用的特性：

- 自定义居中焦点(对人脸等图片显示非常有帮助)
- 圆角图，当然圆圈也行。
- 下载失败之后，点击重现下载
- 自定义占位图，自定义 *overlay*, 或者进度条
- 指定用户按压时的 *overlay*

## 图像的加载

*Fresco* 的 *image pipeline* 设计，允许用户在多方面控制图片的加载：

- 为同一个图片指定不同的远程路径，或者使用已经存在本地缓存中的图片

- 先显示一个低解析度的图片，等高清图下载完之后再显示高清图
- 加载完成回调通知
- 对于本地图，如有 *EXIF* 缩略图，在大图加载完成之前，可先显示缩略图
- 缩放或者旋转图片
- 处理已下载的图片
- *WebP* 支持

## 9.3 性能

*Fresco* 是一个强大的图片加载组件。

*Fresco* 中设计有一个叫做 *image pipeline* 的模块。它负责从网络，从本地文件系统，本地资源加载图片。为了最大限度节省空间和 *CPU* 时间，它含有 **3** 级缓存设计（**2** 级内存，**1** 级文件）。

*Fresco* 中设计有一个叫做 *Drawees* 模块，方便地显示 *loading* 图，当图片不再显示在屏幕上时，及时地释放内存和空间占用。

## 9.4 输入项

上传或 web 图片

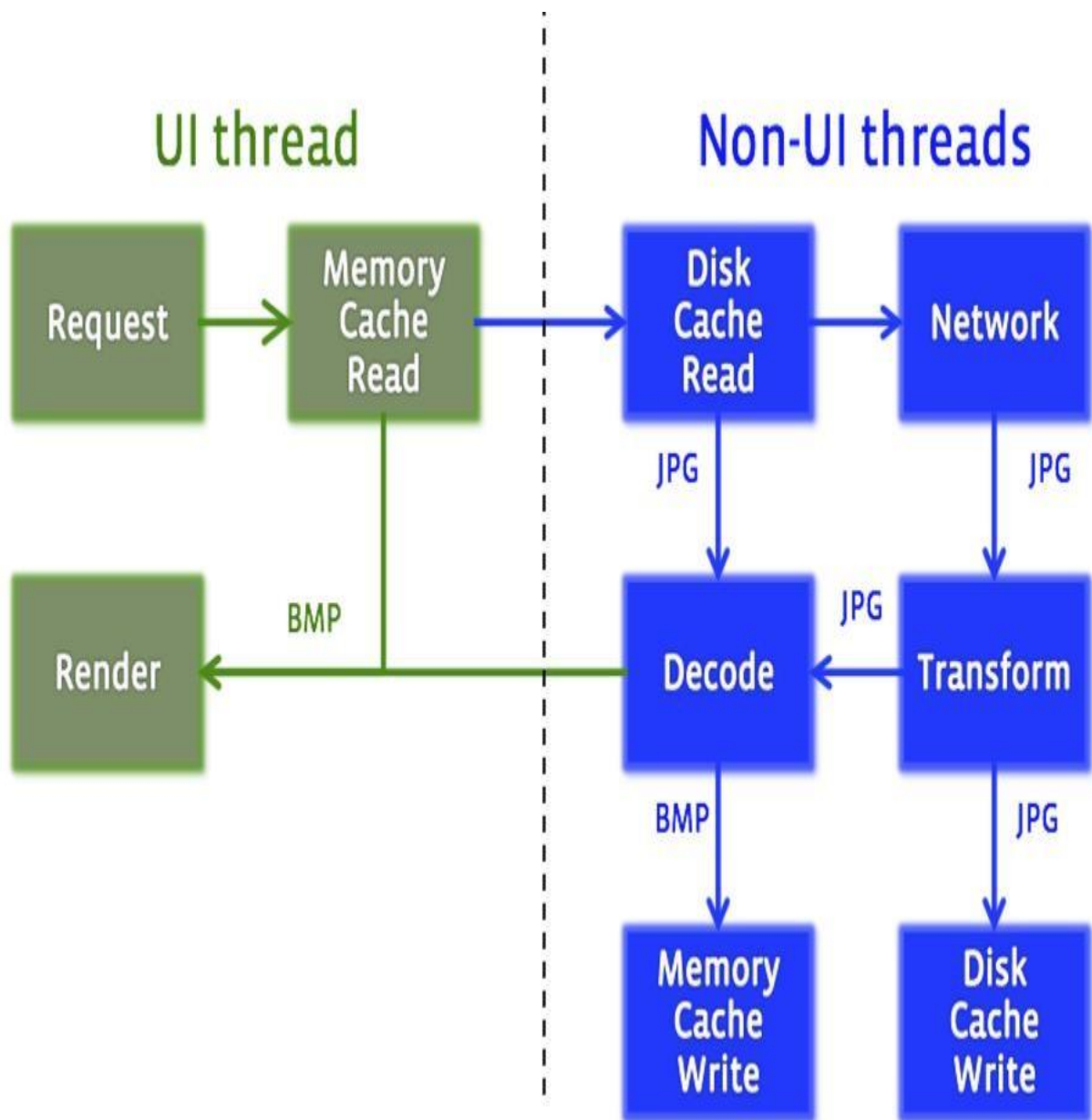
## 9.5 输出项

显示图片并显示图片加载过程



## 9.6 算法

Fresco 框架的 ImagePipeline 设计图



从设计图中可以看出，UIThread 只做图片的显示和从内存缓存中加载图片这两件事，而其它事情如：图片的 Decode、内存缓存的写、硬盘缓存的 IO 操作、网络操作等都由非 UITHread 来处理了，这使得 UITHread 专注界面的显示，而其它工作由其它线程完成，使 UI 更加流畅。

## Fresco 中的 MVC 模式

Fresco 框架整体是一个 MVC 模式

DraweeView——View

DraweeController——Control

DraweeHierarchy——Model 它们之间的关系大致如下：



DraweeHierarchy 意为视图的层次结构，用来存储和描述图片的信息，同时也封装了一些图片的显示和视图层级的方法。

DraweeView 用来显示顶层视图 ( getTopLevelDrawable() )。DraweeController 控制加载图片的配置、顶层显示哪个视图以及控制事件的分发。

【注】DraweeView 目前版本时继承于 ImageView，但这并不意味着我们可以随意的使用 ImageView 相关的方法（如：

setScaleType 等），官方并不建议我们使用，因为后期 DraweeView 将继承于 View，所以最好只使用 DraweeView 控件内置的方法。

## 10 地理距离计算显示附近商家（dxp-fujin）设计说明

### 10.1 程序描述

实现筛选附近商家的功能，并呈现出来

### 10.2 功能

当用户进入商家页，请求离我最近时，需要计算一遍用户位置与全城所有 POI 的距离。然后将 1Km 以内的商家呈现出来

### 10.3 性能

POI 数目 耗时 (ms)
----------------

```

5  0.1*10^-4 10
    0.3*10^-4
100  4*10^-4

```

## 10.4 输入项

获取用户经纬与 poi 经纬

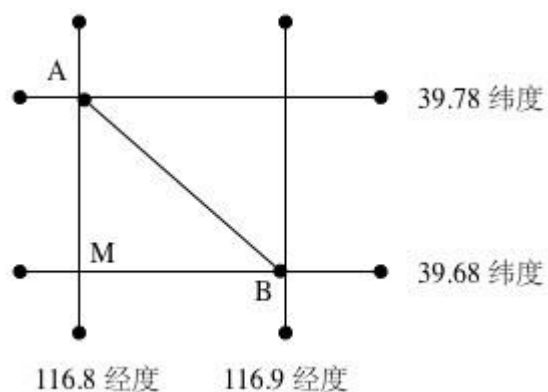
## 10.5 输出项

显示用户与 poi 相距 1Km 以内的商家

## 10.6 算法

### 1) 基本思路

我们的业务场景仅仅是在一个城市范围内进行距离计算，也就是说两个点之间的距离一般不会超过 200 多千米。由于范围小，可以认为经线和纬线是垂直的，如图所示，要求 A（116.8，39.78）和 B（116.9，39.68）两点的距离，我们可以先求出南北方向距离 AM，然后求出东西方向距离 BM，最后求矩形对角线距离，即  $\sqrt{AM^2 + BM^2}$ 。



南北方向

$AM = R \times \text{纬度差} \times \text{Math.PI}/180.0;$

东西方向  $BM = R \cdot \text{经度差} \cdot \cos<\text{当地纬度数} \cdot \text{Math.PI}/180.0>$  这种方式仅仅需要计算一次  $\cos$  函数。

2) 进一步优化我们看到这里计算了一次  $\cos$  这一三角函数，如果我们能消除此三角函数，那么将进一步提高计算效率。

如何消除  $\cos$  三角函数呢？采用多项式来拟合  $\cos$  三角函数，这样不就可以将三角函数转换为加减乘除了嘛！首先决定多项式的最高次数，次数为 1 和 2 显然都无法很好拟合  $\cos$  函数，那么我们选择 3 先尝试吧，注：最高次数不是越多越好，次数越高会产生过拟合问题。

使用 `org.apache.commons.math3` 这一数学工具包来进行拟合。中国的纬度范围在 10~60 之间，即我们将此区间离散成 `Length` 份作为我们的训练集。

## 10.7 接口

```
public static double distanceSimplifyMore(double lat1, double lng1, double lat2, double lng2, double[] a)
```

# Web 端：

个人账户管理模块：

### 1.1 模块描述

提供用户的注册、登陆、登出、账户信息修改等功能

### 1.2 功能

- (1)注册 (2)登陆
- (3)登出
- (4)修改个人信息

### 1.3 模块详细设计

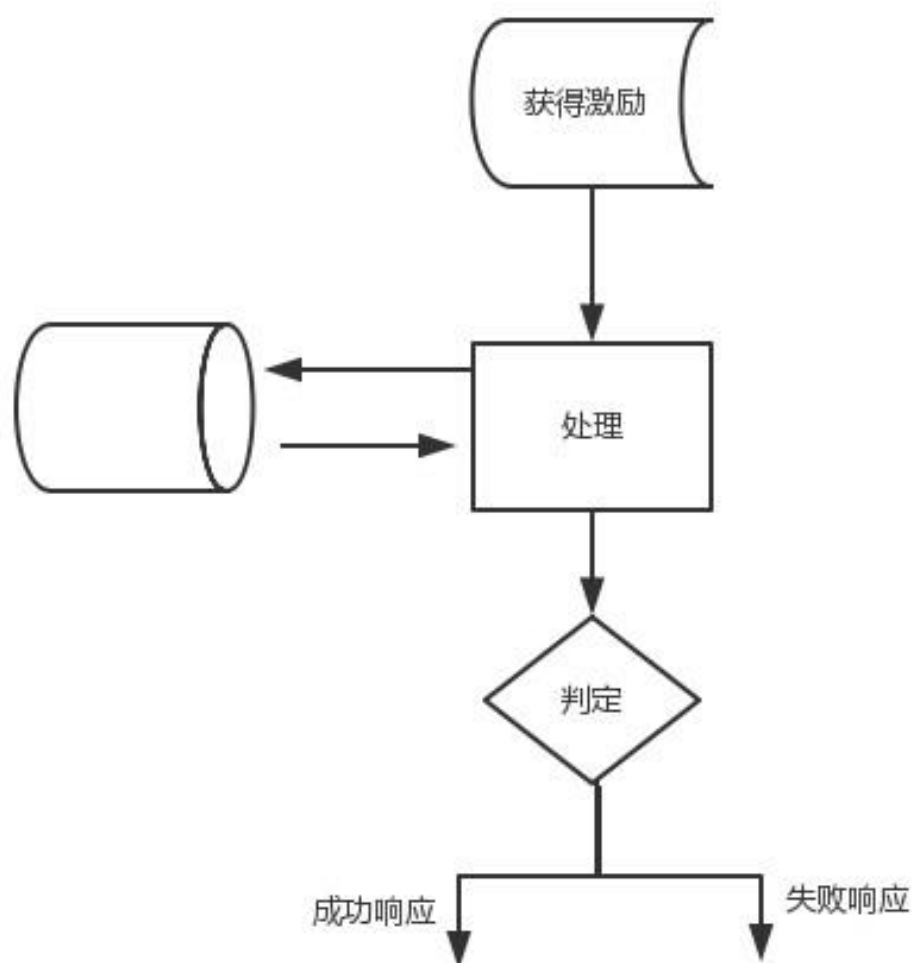
#### 1.3.1 注册详细设计

##### 1.3.1.1 输入激励

- (1)用户名由 http 协议中的 form 获得
- (2)昵称由 http 协议中的 form 获得
- (3)手机号码由 http 协议中的 form 获得
- (4)性别由 http 协议中的 form 获得
- (5)年龄由 http 协议中的 form 获得 •
- (6)密码由 http 协议中的 form 获得
- (7)兴趣爱好由 http 协议中的 form 获得

#### 1.3.1.2 输出项

- (1)201 状态成功标志
- (2)用户名已存在



#### 1.3.1.4 接口

API 输入激励	/regist		
请求方式	post		
参数说明	参数名	参数范围	可选性

用户名	username	小于 10 位字母 数字组合和	必选
昵称	nickname	大于 6 位	必选
手机号码	phone	11 位数字	必选
性别	sex	1/0	必选
年龄	age	0 到 100 以内 数字	必选
密码	password	大于 6 位数字 字母组合字	必选
兴趣爱好标签	hobby		
API 成功响应输出	201		
参数说明	参数名	参数范围	可选性
失败响应	403		
参数说明	参数名	参数范围	可选性
用户名	username	existed	必选

#### 1.3.1.5 限制条件及出错处理

- (1)用户名已经存在，则需要用户更换用户名重新注册

### 1.3.2 登陆详细设计

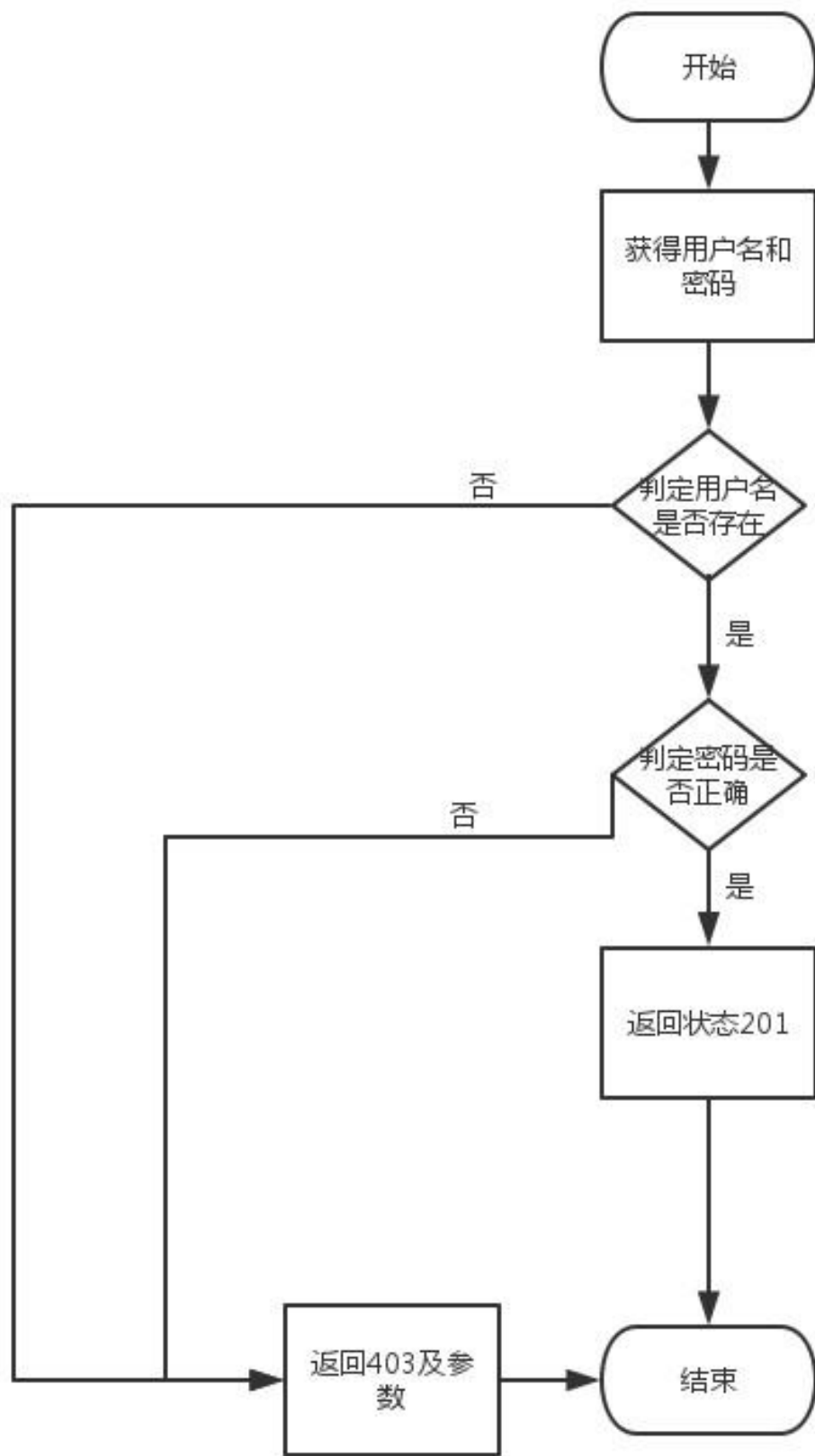
#### 1.3.2.1 输入激励

- (1)用户名由 http 协议中的 form 获得
- (2)密码由 http 协议中的 form 获得

#### 1.3.2.2 输出项

- (1)204 状态成功标志
- (2)用户名错误
- (3)密码错误

### 1.3.2.3 逻辑流程



#### 1.3.2.4 接口

API 输入	/login		
+C1+B2:G21			
请求方式	get		
参数说明	参数名	参数范围	可选性
用户名	username	小于 10 位字母和数字组合	必选
密码	password	大于 6 位数字字母组合	必选
成功响应	状态 204		
参数说明	参数名	参数范围	可选性
失败响应	状态 403		
用户名错误	参数名	参数范围	可选性
密码错误	username	ture/false	必选
	password	ture/false	必选

#### 1.3.2.5 限制条件及出错处理

- (1)用户名错误，提示用户输入正确的用户名
- (2)密码错误，提示用户输入正确的密码

### 1.3.3 账户信息修改详细设计

#### 1.3.3.1 输入激励

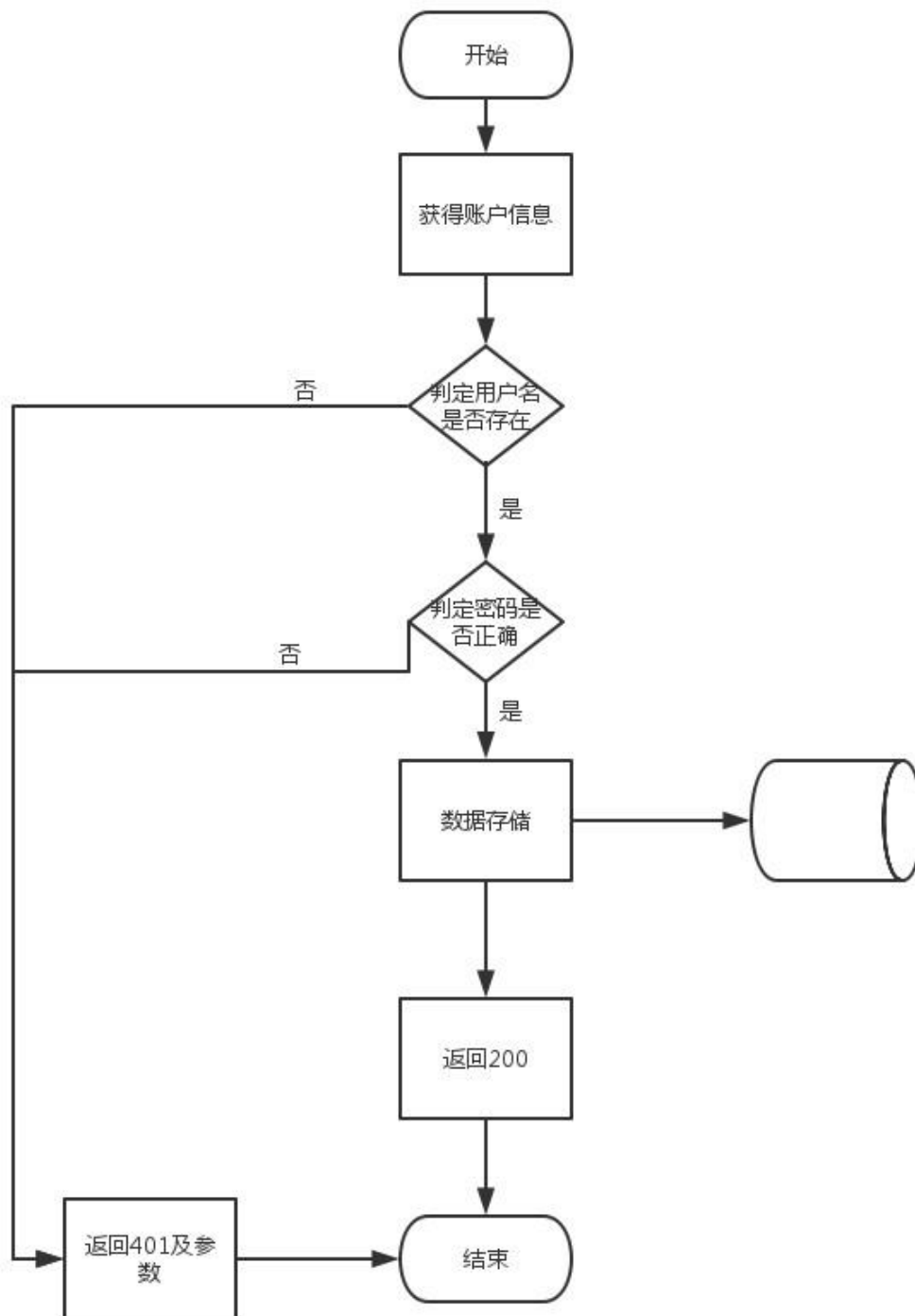
- (1)用户名由 http 协议中的 json 获得、
- (2)登陆标志

#### • 1.3.3.2 输出项

- (1)204 状态成功标志
- (2)用户名错误
- (3)未登录

#### 1.3.3.3 逻辑流程





#### 1.3.3.4 接口

API 输入激励	/ serinformationamed		
请求方式	get		
参数说明	参数名	参数范围	可选性
用户名	username	小于 10 位字母和数字组合	必选

手机号	iphone	11 位数字	必选
兴趣爱好标签	hobby	列表	必选
昵称	nickname	大于 6 位	必选
性别	sex	1/0	必选
年龄	age	0 到 100 以内数字	必选
地址	address	30 个字符以内的字符串	必选
API 成功响应输出	200		
参数说明	参数名	参数范围	可选性
失败响应	401		
参数说明	参数名	参数范围	可选性
说明信息	error	unauthorized	必选
失败响应	404		
参数说明	参数名	参数范围	可选性
说明信息	error	notexisted	必选

#### 1.3.3.5 限制条件及出错处理

- (1)用户未登陆，提示用户登录后才能登出
- (2)无此用户名，提示输入正确的用户名

#### 1.3.4 登出详细设计

##### 1.3.4.1 输入激励

(1)用户名由 http 协议中的 json 获得

(2)已登陆 flag http 协议中的 json 获得

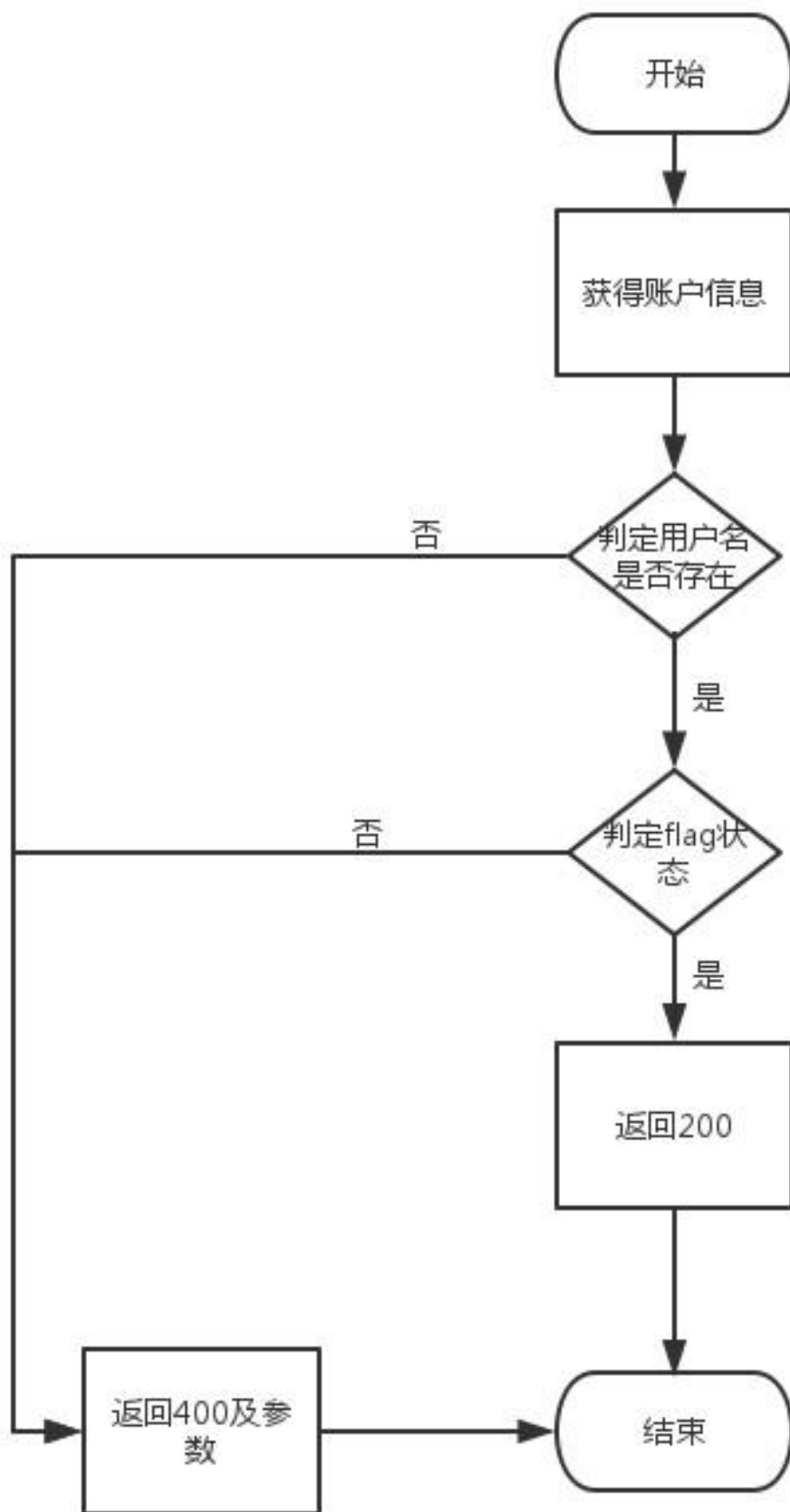
- 1.3.4.2 输出项

(1)200 状态成功标志

(2)404 错误，用户不存在

(3)401 错误，用户未未登录

1.3.4.3 逻辑流程



#### 1.3.4.4 接口

API 输入激励	/logout				
请求方式	get				
参数说明	参数名	参数范围			可选性
用户名	username	小于 10 位字母和数字组合			必选
登陆标志	flag	0 或 1			必选
成功响应	状态 204				
参数说明	参数名	参数范围			可选性
失败响应	状态 400				
参数说明	参数名	参数范围			可选性
用户名错误	username	ture/false			必选
未登录	flag	ture/false			必选

#### 1.3.4.5 限制条件及出错处理

- (1)用户未登陆，提示用户登录后才能修改个人信息
- (2)无此用户名，提示输入正确的用户名

### 收藏管理模块：

#### 2.1 模块描述

提供用户对商品、商店的收藏，浏览收藏的商品、商店等功能

#### 2.2 功能

- (1)收藏
- (2)浏览收藏

#### 2.3 模块详细设计

### 2.3.1 收藏详细设计

#### 2.3.1.1 输入激励

(1)用户名由 http 协议中的 json 获得

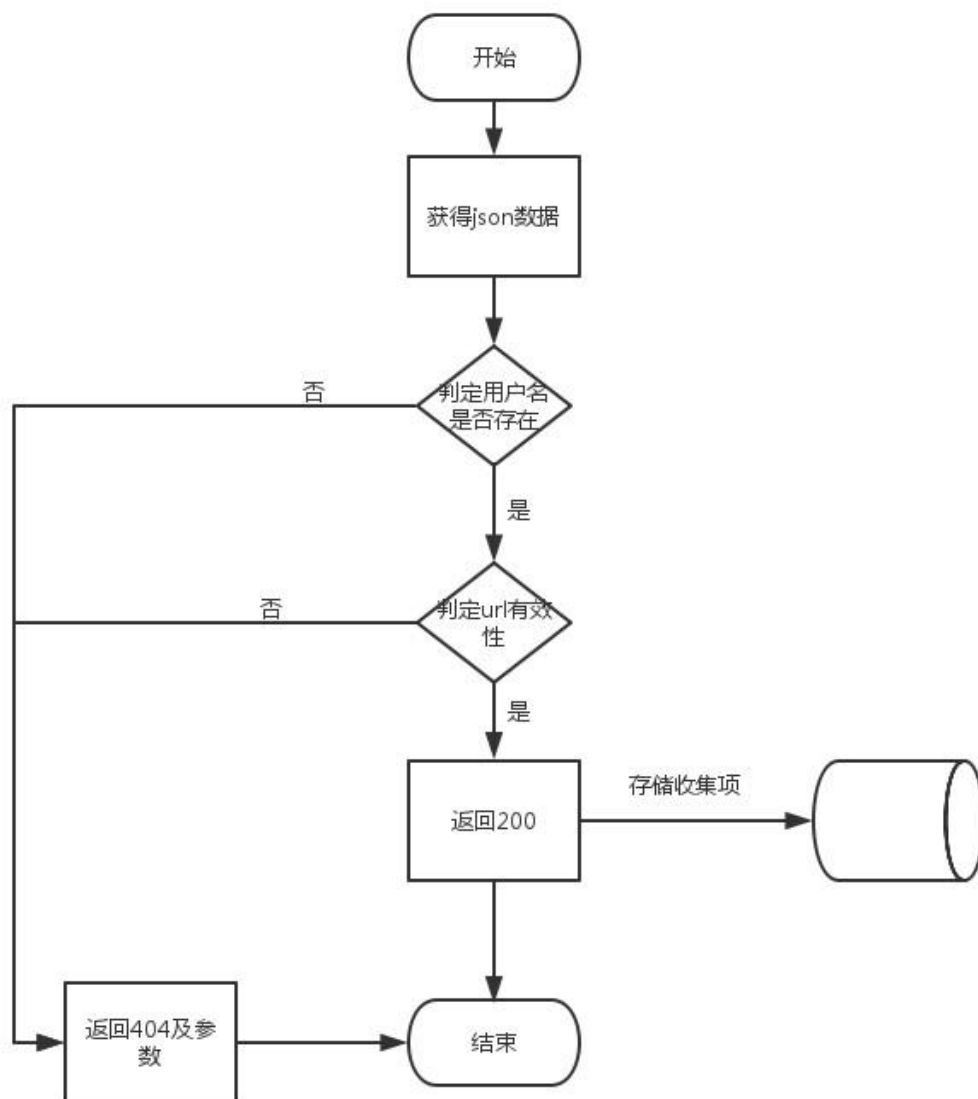
(2)收藏项由 http 协议中的 json 获得

#### • 1.3.1.2 输出项

(1)200 状态成功标志

(2)404 错误，用户不存在或收藏项不存在

#### 1.3.1.3 逻辑流程



#### 1.3.1.4 接口

API 输入激励	/collect
请求方式	get

参数说明	参数名	参数范围	可选性
用户名	username	小于 10 位字母和数字组合	必选
收藏项	tocollect	url	必选
API 成功响应输出		200	
参数说明	参数名	参数范围	可选性
失败响应		404	
参数说明	参数名	参数范围	可选性
用户名	username	notexisted	可选
收藏项	tocollect	notexisted	可选

#### 1.3.1.5 限制条件及出错处理

- (1)收藏项，提示收藏项不存在，输入正确的收藏项 url
- (2)无此用户名，提示输入正确的用户名

### 2.3.2 浏览收藏详细设计

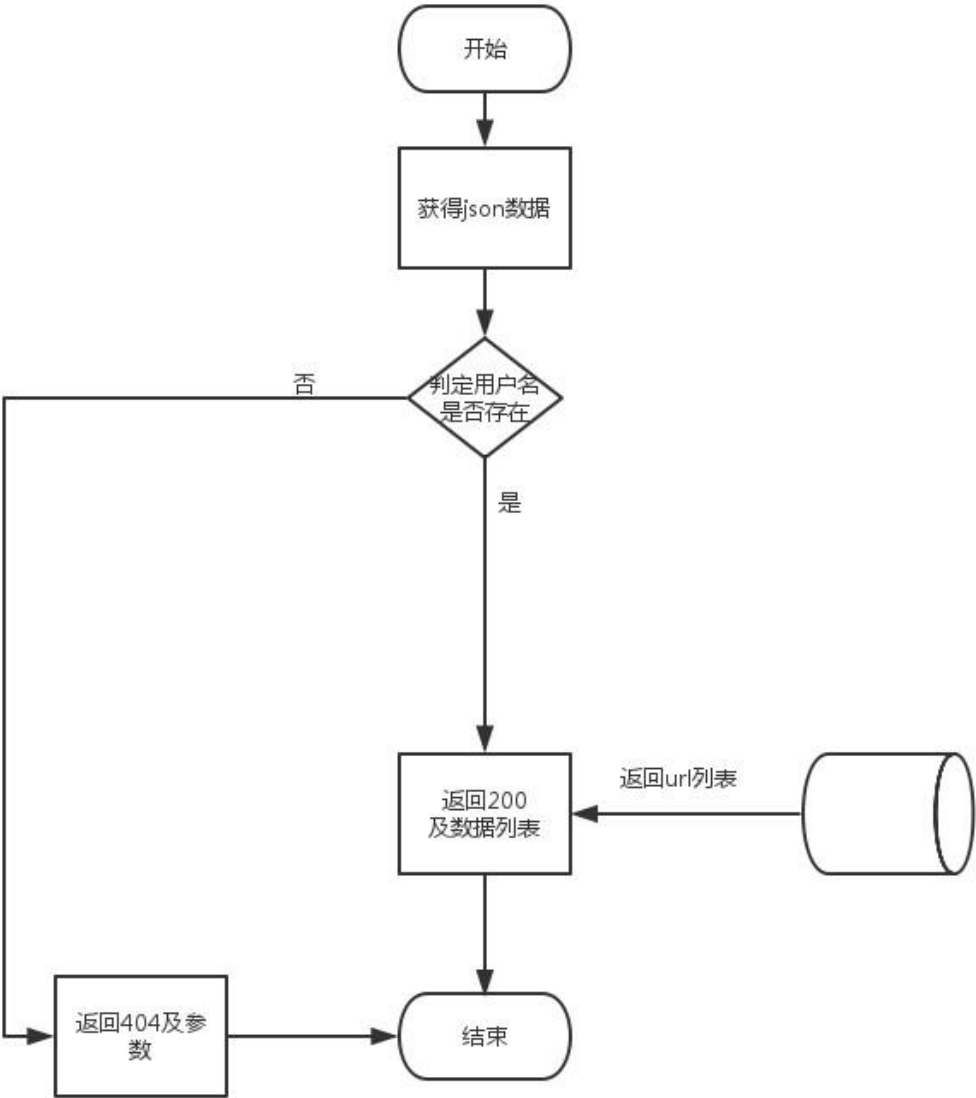
#### 2.3.2.1 输入激励

- (1)用户名由 http 协议中的 json 获得

#### • 1.3.2.2 输出项

- (1)商品的 url 列表
- (2)收藏商店 url 列表

1.3.2.3 逻辑流程



1.3.2.4 接口

API 输入激励	/collection		
请求方式	get		
参数说明	参数名	参数范围	可选性
用户名	username	小于 10 位字母 数字组合和	必选



API 成功响应输出	200		
参数说明	参数名	参数范围	可选性
商品列表	item	url 列表	必选
收藏的店家	merchant	url 列表	必选
失败响应	404		
参数说明	参数名	参数范围	可选性
说明信息	error	iphonewro ig	必选

#### 1.3.2.5 限制条件及出错处理

(1)无此用户名，提示输入正确的用户名

## 搜索与信息获取模块：

### 3.1 模块描述

提供商品的搜索，商品、商店、用户的信息获取

### 3.2 功能

(1)搜索

(2)信息获取

### 3.3 模块详细设计

#### 3.3.2 搜索详细设计

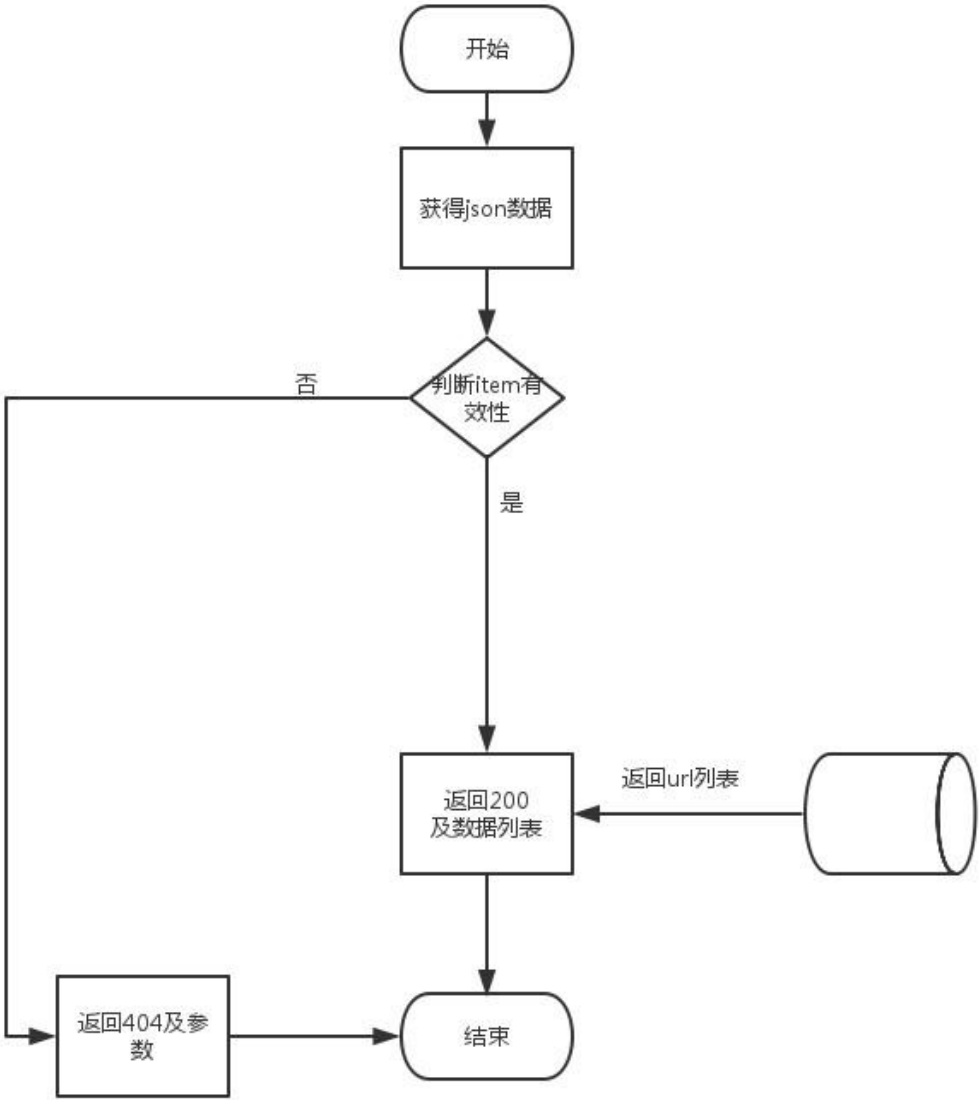
##### 3.3.2.1 输入激励

(1)商品名，由 http 协议中的 json 获得

##### • 3.3.2.2 输出项

(1)商品信息的 url 列表

3.3.2.3 逻辑流程



3.3.2.4 接口

API 输入激励	/search		
请求方式	get		
参数说明	参数名	参数范围	可选性
商品名	item	连续的字符串	必选

API 成功响应输出	303		
参数说明	参数名	参数范围	可选性
商品信息	itemimfornation	url	必选
失败响应	404		
参数说明	参数名	参数范围	可选性

#### 3.3.2.5 限制条件及出错处理

(1)响应失败，可能是后端数据出错，客户端需要提醒用户输入有误，并重新输入搜索信息。

### 3.3.2 商家信息获取设计

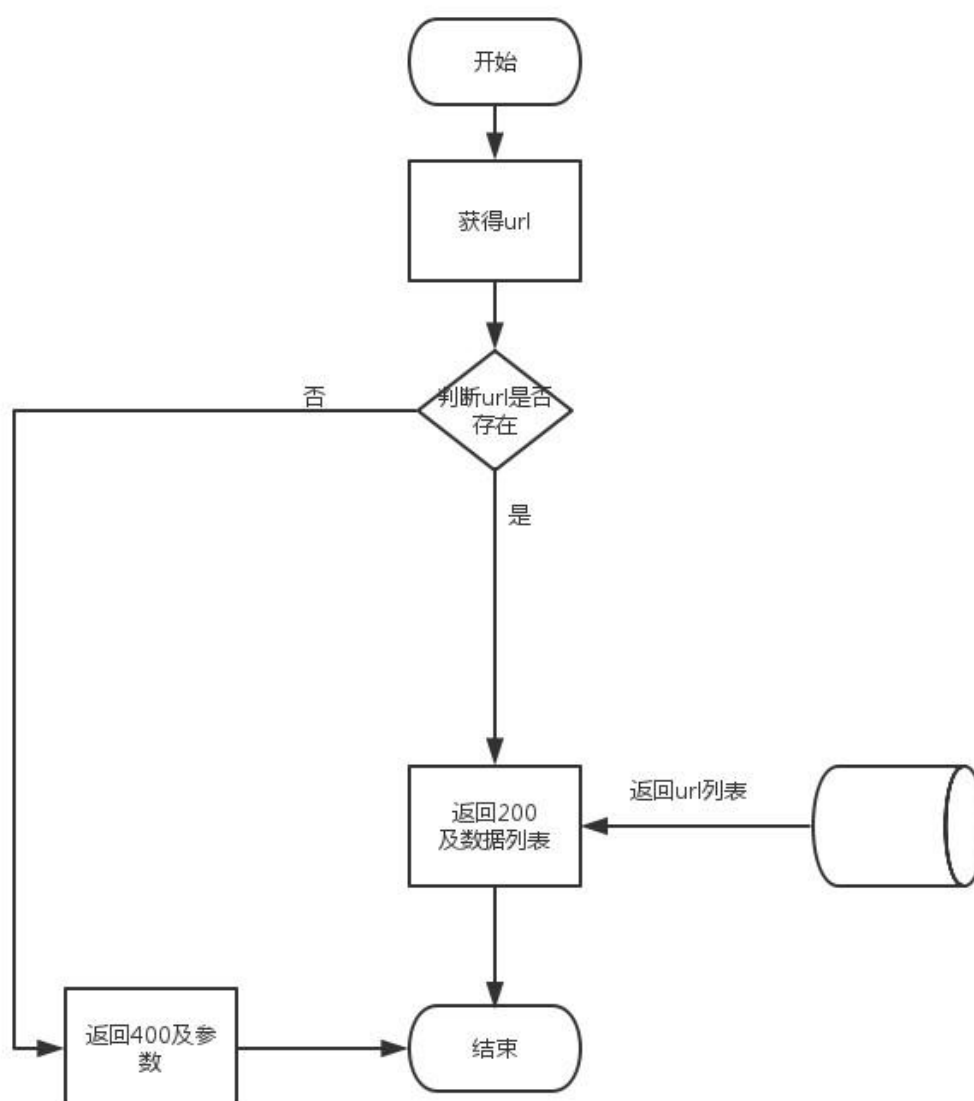
#### 3.3.2.1 输入激励

(1)商家 url

#### • 3.3.2.2 输出项

- (1) 商家名称
- (2) 商家的地理位置
- (3) 商家营业时间
- (4) 商家的评论
- (5) 商家拥有的商品 url 列表

#### 3.3.2.3 逻辑流程



### 3.3.2.4 接口

API 输入激励	/<商家 url>		
请求方式	get		
参数说明	参数名	参数范围	可 选 性

API 成功响应输出	200				
参数说明	参数名	参数范围		可 选 性	
商家名称	name	20 长度以内的字符串		必选	
商家的地理位置	location	100 长度以内的字符串		必选	
商家的营业时间	time	00 : 00 到 24 : 59 之间		必选	
对商家的评论	marks	100 长度以内的字符串列表		必选	
商家所拥有的商品	item	url 列表		必选	
失败响应	400				
参数说明	参数名	参数范围		可选性	

#### 3.3.2.5 限制条件及出错处理

(1)未输入正确的 url，提示输入正确的 url 后响应

### 3.3.3 商品信息获取设计

#### 3.3.3.1 输入激励

(1)商品 url

#### • 3.3.3.2 输出项

- (1) 商品价格 (2) 具体位置
- (3) 商品评分
- (4) 商品库存量
- (5) 商品所属商家
- (6) 商品以往品论列表
- (7) 所属标签

#### 3.3.3.3 逻辑流程

同上

#### 3.3.3.4 接口

API 输入激励	/<商品 url>				
请求方式	get				
参数说明	参数名	参数范围			可 选 性
API 成功响应输出	200				
参数说明	参数名	参数范围			可 选 性
商品价格	price	数字			必选
具体位置	location	20 长度内字符串			必选
商品的评分	mark	1 到 10 数字			必选
商品库存量	storenumber	大于 0 数字			必选
商品所属商家	merchant	20 长度内字符串			必选
商品编号	number	10 位数字			必选
商品以往评论	marks	100 长度以内的字符串列表			必选
所属标签	flag'	20 长度以内字符串			必选
失败响应	404				
参数说明	参数名	参数范围			可选性

### 3.3.3.5 限制条件及出错处理

(1)未输入正确的 url，提示输入正确的 url 后响应

## 3.3.4 用户信息获取设计

### 3.3.4.1 输入激励

(1)用户手机号

• 3.3.4.2 输出项

(1) 用户名

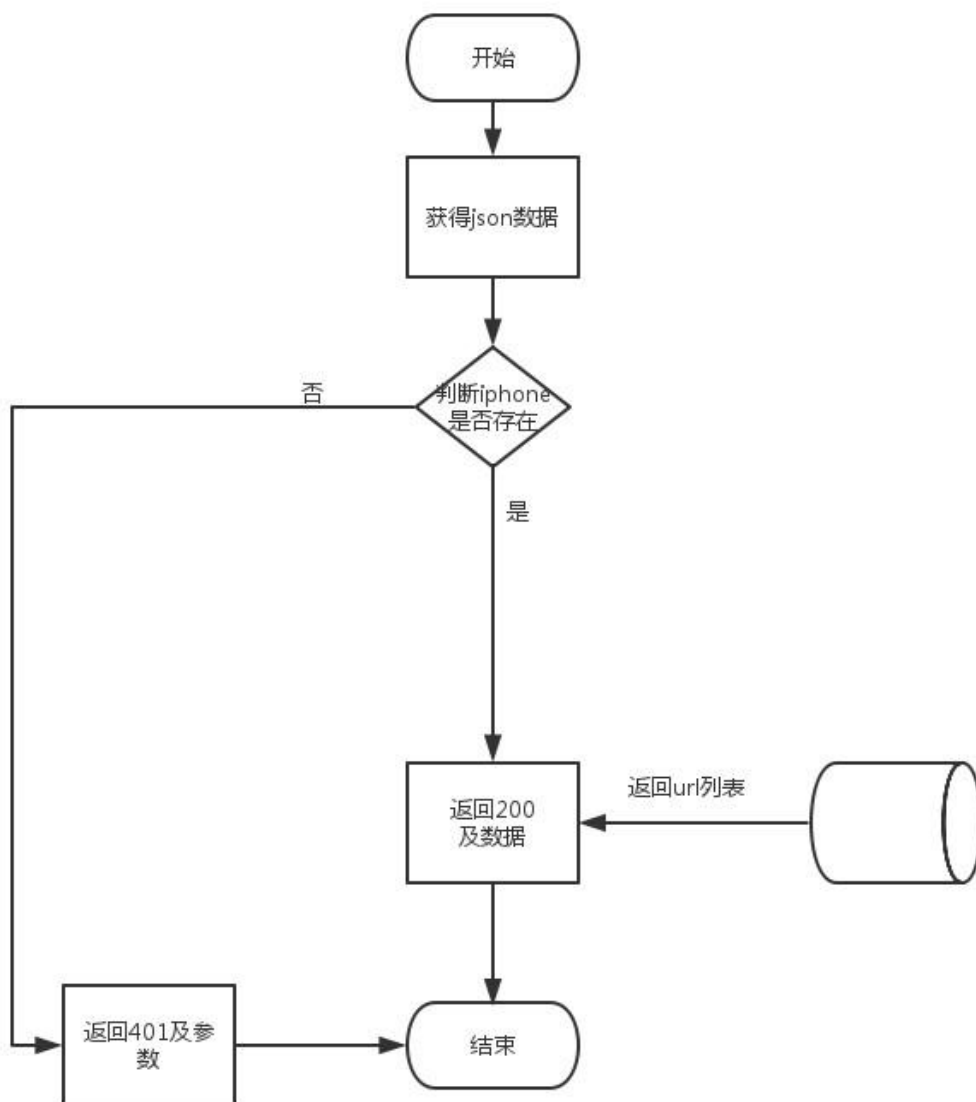
(2) 手机号

(3) 兴趣爱好

(4) 昵称 (5) 性别 (6) 年龄

(7) 地址

3.3.4.3 逻辑流程



3.3.4.4 接口

API 输入激励	/userinformation
请求方式	get

参数说明	参数名	参数范围	可 选 性
用户手机号	iphone	11 位数字	必选
API 成功响应输出	200		
参数说明	参数名	参数范围	可 选 性
用户名	username	小于 10 位字母和数字组合	必选
手机号	iphone	11 位数字	必选
兴趣爱好标签	hobby	列表	必选
昵称	nickname	大于 6 位	必选
性别	sex	1/0	必选
年龄	age	0 到 100 以内数字	必选
地址	address	30 个字符以内的字符串	必选
失败响应	401		
参数说明	参数名	参数范围	可选性
用户不存在	error	notexisted	必选

#### 3.3.4.5 限制条件及出错处理

(1)未输入正确的手机号，提示输入正确的手机号获取用户信息



# 数据库：

一、系统概述建立此数据库的意义即为存储用户与商家的信息，从而能实现根据用户的喜好来匹配与之距离最近的商品所在的位置，从而节省用户购物时间。

该数据库主要记录如下数据：

- 1、 用户（用户名、密码、昵称、手机号码、爱好、年龄、性别）
- 2、 商品（所属商家、图片、商品名、评价、价格、标签、库存）
- 3、 商铺（商铺名、商品名、标签、地理位置、图片、运营时间）
- 4、 对于商品的评价（所在商铺、评论者、内容） 5、对于商铺评论（所属商品、评论者、内容）但是考虑到其中有些属性值为多值属性，考虑到范式的需求，在实际建表中将会将一些属性拿出单独建表。

## 二、实体关系

一件商品可以被多个用户查找，一个用户也可以查找多个商品，所以用户实体与商品实体之间所建立的查找关系是多对多的。

一种商品可以属于多个商铺，一个商铺也可以有多种商品，所以商品实体和商铺实体之间所建立的查找关系是多对多的。

某种商品可以被多个用户收藏，一个用户也可以收藏多种商品，所以其建立的关系是多对多的。

某个商铺可以被多个用户收藏，一个用户也可以收藏多种商品，所以其建立的关系是多对多的。

某种商品可以被多个用户评论，但一个评价只能对应一种商品，所以其关系是一对多的。

某个商铺可以被多个用户评论，但一个评价只能对应一种商铺，所以其关系是一对多的。

## 三、实体属性

用户表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Id	用户名	Varchar(20)	是	否	是	否	无	否
Age	年龄	Int(2)	否	否	是	否	无	否
Nickname	昵称	Varchar(20)	否	否	是	是	无	否
Phone	手机号	Int(13)	否	否	是	是	无	否
Mima	密码	Varchar(20)	否	否	是	是	无	否

爱好表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
-----	------	------	----	----	----	----	-----	----

Id1	用户名	Varchar(20)	否	是	是	否	无	否
Hobby	爱好	Varchar(20)	否	是	否	否	无	否

商店表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
ShopName	店铺名	Varchar(20)	是	否	是	否	无	否
Cordx	坐标 x	Float(3)	否	否	是	否	无	否
Cordy	坐标 y	Float(3)	否	否	是	否	无	否
Label	标签	Varchar(30)	否	否	是	否	无	否
Runtime	运营时间	Varchar(30)	否	否	是	否	无	否
Goods	商品	Varchar(30)	否	否	否	是	无	否

图片表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Picture	图片	Medium blob	否	否	是	否	无	无
ShopName1	店铺名	Varchar(20)	否	是	是	否	无	无

商品表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
goodsname	商品名	Varchar(30)	否	否	是	否	无	否
price	价格	Int(5)	否	否	是	否	无	否
Label	标签	Varchar(30)	否	否	是	否	无	否
Rank	评级	enum	否	否	是	否	无	否
Amount	库存	Int(5)	否	否	是	否	无	否
ShopName2	商铺名	Varchar(20)	否	是	是	否	无	否

图片表

1

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Picture1	图片	medium BLOB	否	否	是	否	无	否
Goodsname1	商品名	Varchar(30)	否	是	是	否	无	否

收藏表

1

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Shopname3	店铺名	Varchar(30)	否	是	是	否	无	否
Thetime	收藏时间	Datetime	否	否	是	否	无	否
Id	用户名	Varchar(30)	否	是	是	否	无	否

收藏表 2

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Goodsname2	商品名	Varchar(30)	否	是	是	否	无	否
Thetime	收藏时间	Datetime	否	否	是	否	无	否
Id	用户名	Varchar(30)	否	是	是	否	无	否

评论表 1

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Shopname4	店铺名	Varchar(20)	否	是	是	否	无	无
Id2	用户名	Varchar(20)	否	是	是	否	无	无
Content	评论内容	Medium text	否	否	是	否	无	无

评论表 2

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增

Goodsname3	商品名	Varchar(20)	否	是	是	否	无	无
Id2	用户名	Varchar(20)	否	是	是	否	无	无
Content	评论内容	Medium text	否	否	是	否	无	无

用户商品关系表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Id4	用户名	Varchar(20)	否	是	是	否	否	否
ShopName5	店铺名	Varchar(20)	否	是	是	否	否	否

商品商铺关系表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Goodsname4	商品名	Varchar(20)	否	是	是	否	否	否
ShopName6	店铺名	Varchar(20)	否	是	是	否	否	否

商品评论关系表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Goodsname5	商品名	Varchar(20)	否	是	是	否	否	否
Id6	用户名	Varchar(20)	否	是	是	否	否	否

商铺评论关系表

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Shopname7	商铺名	Varchar(20)	否	是	是	否	否	否
Id7	用户名	Varchar(20)	否	是	是	否	否	否

收藏关系表 1

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Shopname8	商铺名	Varchar(20)	否	是	是	否	否	否
Id8	用户名	Varchar(20)	否	是	是	否	否	否
Dates1	收藏时间	Datetime	否	否	是	否	否	否

收藏关系表 2

字段名	字段描述	数据类型	主键	外键	非空	唯一	默认值	自增
Goodsname9	商品名	Varchar(20)	否	是	是	否	否	否
Id9	用户名	Varchar(20)	否	是	是	否	否	否
Dates2	收藏时间	Datetime	否	否	是	否	否	否

四、索引索引的目的旨在方便系统能够对所需内容进行快速查找，在使用 MySQLworkbench 开发平台进行数据库建立时，已为字段自动建好了相应的 Btree 索引，所以无需再多赘述。