

测试计划（GB8567——88）

1 引言

1.1 编写目的

这次测试其主要目的是为了描述对该 APP 要测试的各个模块的功能、测试的方法、以及可能出现的健壮性问题。从而使该 APP 能够完全实现预定的功能。

1.2 背景

该软件名称为 EasyPick。

该项目在理想情况下源自于广大消费者的需求，使消费者在购买商品时能够通过该软件更快更准确地匹配到与自己最近并且符合要求的商铺，从而节省时间。此项目的使用用户和测试的计算中心均基于 Android 平台的手机，在实际测试中，必须配置好相关的软件运行环境以及设置好测试参数。

1.3 定义

图片	商品和商铺的实体图片
个人资料	展示用户的基本信息，并且可以更改
评价	允许用户对商品和商铺进行打分和评论

收藏	支持用户存储自己喜好的商铺，该功能会在APP界面上显示
定位	该APP调用高德API实现定位以及就近匹配的功能
推送	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及商铺

1.4 参考资料

《软件工程导论》
《详细设计说明书》
《项目开发计划》
《需求说明书》

2 计划

2.1 软件说明

提供一份图表，并逐项说明被测软件的功能、输入和输出等质量指标，作为叙述测试计划的提纲。

功能	子功能	输入	输出
个人账户管理模块	用户注册	用户名、密码、以及个人信息	状态 200
	登出	“先看看”命令	按点赞数排列的状态列表
	用户登录	用户名、密码、	状态 204
	修改个人信息	用户名加个人信息	状态 400
收藏管理模块	收藏	用户名和收藏的 url	状态 200
	浏览收藏	用户名	状态 200 和 url 参数列表
搜索与信息获取模块	搜索	商品或商家名称”	状态 303 和 url 列表
	商家信息获取	商家的 url	状态 200 和商家各项信息
	用户信息获取	用户的 iphone	状态 200 和用户的各项信息
	商品信息获取	商品的 url	状态 200 和商品的各项属性

功能	输入	输出
地理距离优化	从后端获取	附近的商家
高斯模糊算法效率	原始图片	处理后的图片
商品热度排名	商家商品	排名后的商品

标识符	测试内容	与预期差别
表测试	测试计划的所有内容	无
基本操作测试	测试计划的所有内容	无
查询测试	测试计划的所有内容	无

2.2 测试内容

一、

①单元测试

目的：发现各模块内部可能存在的各种差错

测试内容：模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试。

②集成测试

目的：发现各模块连接中可能发生的问题

测试内容：采用自顶向下集成或者自底向上集成的方式对所选程序进行模块非增量式集成测试。

③系统测试

目的：验证系统是否满足需求规格的定义

测试内容：功能测试、健壮性测试。

二、

列出组装测试和确认测试中的每一项测试内容的名称标识符、这些测试的进度安排以及这些测试的内容和目的，例如模块功能测试、接口正确性测试、数据文卷存取的测试、运行时间的测试、设计约束和极限的测试等。

三、检验数据库系统是否稳定以及是否有缺陷，为后端提供一个良好的基础。

2.3 测试 1（单元测试）

参与人员：殷熔磬

被测功能：软件运行过程中各个模块的功能

2.3.1 进度安排

测试进度为 1 天，测试内容主要包括模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试。

2.3.2 条件

所需硬件：

笔记本

所需软件：Windows 操作系统

开发平台：python3

开发框架：flask

API 测试工具

数据库管理系统：MySQL

所需工作人员：殷熔碑

2.3.3 测试资料

《需求规格说明书》
《详细设计说明》
被测程序源代码

2.3.4 测试培训

无

2.4 测试 2（集成测试）

参与单位：殷熔碑
被测功能：软件各模块之间的连接功能

2.4.1 进度安排

测试进度为 1 天，测试内容主要为采用自顶向下集成或者自底向上集成的方式对所选程序进行模块非增量式集成测试。

2.4.2 条件

所需硬件：
所需硬件：
笔记本
所需软件：Windows 操作系统
开发平台：python3
开发框架：flask
API 测试工具
数据库管理系统：MySQL
所需工作人员：殷熔碑

2.4.3 测试资料

《需求规格说明书》
《详细设计说明》
被测程序源代码

2.4.4 测试培训

无

2.5 测试 3（系统测试）

参与单位：殷熔碑
被测功能：软件各模块之间的连接功能

2.5.1 进度安排

测试进度为 1 天，测试内容主要包括功能测试、健壮性测试。

2.5.2 条件

所需硬件：
笔记本
所需软件：Windows 操作系统
开发平台：python3
开发框架：flask
API 测试工具
数据库管理系统：MySQL
所需工作人员：殷熔碑

2.5.3 测试资料

《需求规格说明书》

2.5.4 测试培训

无

2.6 地理距离算法耗时测试（d xp-01）

参与人员：丁鑫棚

被测功能：计算全城 poi 与用户所在地点的距离，选举距离用户 1Km 以内的店家显示，测试结果 ms

2.7 高斯模糊效率测试(d xp-02)

参与人员：丁鑫棚

被测功能：获取图片文件，通过计算实现高斯模糊，测试所耗时间，测试结果以 ms 输出。

2.8 表测试

参与人员：汪哲汇

被测功能：数据库表的基本操作

2.9 基本操作测试

参与人员：汪哲汇

被测功能：数据库表的基本操作

2.10 查询测试

参与人员：汪哲汇

被测功能：数据库表的基本操作

3 测试设计说明

3.1 测试 1（单元测试）

3.1.1 控制

人工输入

3.1.2 输入

该测试需要输入的信息为系统各个 url 及其参数

3.1.3 输出

预期的输出数据为函数的正确输出值。

3.1.4 过程

- ①依据被测程序的各个函数的注释设计测试用例
- ②选择被测单元
- ③创建测试用例，然后运行，看实际运行结果是否与预期匹配。
- ④重复②③，直到所有 api 均已被测。

3.2 测试 2（集成测试）

3.2.1 控制

人工输入

3.2.2 输入

该测试需要输入的信息为被调用函数的输入参数

3.2.3 输出

预期输出为正确的输出结果

3.2.4 过程

把底层模块组织成实现某个子功能的模块群；

开发一个测试驱动模块，控制测试数据的输入和测试结果的输出；

对每个模块群进行测试；

删除测试使用的驱动模块，用较高层模块把模块群组织成为完成更大功能的新模块群。
从第一步开始循环执行上述各步骤，直至整个程序构造完毕。

3.3 测试 3（系统测试）

3.3.1 控制

人工输入

3.3.2 输入

该测试需要输入的信息为软件运行中可能遇到的各种外在条件

3.3.3 输出

预期输出为该 api 对各种情况的正确响应。

3.3.4 过程

为测试 api 的输入信息设计出错处理通路，设计测试用例，模拟错误数据和系统可能发生的错误，进行系统测试的规划和设计。

3.4 地理距离算法耗时测试(dxp-01)

3.4.1 控制

获取所收集到的 poi 地理位置

3.4.2 输入

poi 的经纬度

3.4.3 输出

距离用户位置 1Km 内的 poi 以及耗时

3.4.4 过程

- 一. 为了简化测试，输入 10 个 poi 的经纬度
- 二. 输入用户经纬度
- 三. 计算用户与所有 poi 的距离
- 四. 输出距离 1Km 的 poi 和耗时

说明完成此项测试的一个个步骤和控制命令，包括测试的准备、初始化、中间步聚和运行结束方式。

3.5 高斯模糊算法效率测试(dxp-02)

3.5.1 控制

获取图片资源

3.5.2 输入

图片资源

3.5.3 输出

高斯模糊后的资源以及耗时

3.5.4 过程

- 一. 获取图片资源
- 二. 进行高斯模糊
- 三. 输出模糊后的资源以及耗时

3.6 商品热度排名测试(dxp-03)

3.6.1 控制

获取商品热度排名，并按照排名输出

3.6.2 输入

商品热度

3.6.3 输出

依据热度排名的商品

3.6.4 过程

- 一. 获取各商品的热度
- 二. 依据热度进行快排
- 三. 热度排序后由高到低排列商品

3.7 表测试

3.7.1 控制

输入变量数据类型正确。

3.7.2 输入

show tables; show columns from goods; show columns from shop; show columns from users;

3.7.3 输出

```
MySQL 5.7 Command Line Client
mysql> show tables;
+-----+
| Tables_in_base |
+-----+
| collect1       |
| collect2       |
| goods          |
| hobby          |
| pictures       |
| pictures1      |
| relations      |
| relationrg     |
| relations      |
| relationug     |
| remark1        |
| remark2        |
| shop           |
| shoucang1      |
| shoucang2      |
| users          |
+-----+
16 rows in set (0.00 sec)
```

```

MySQL 5.7 Command Line Client

+-----+
| goods | CREATE TABLE `goods` (
  `goodsname` varchar(30) NOT NULL,
  `price` int(5) NOT NULL,
  `label` varchar(30) NOT NULL,
  `rank` enum('very low','low','normal','good','very good') NOT NULL,
  `Amount` int(5) NOT NULL,
  `ShopName2` varchar(20) NOT NULL,
  KEY `fk_3` (`goodsname`),
  KEY `fk_4` (`ShopName2`),
  CONSTRAINT `fk_3` FOREIGN KEY (`goodsname`) REFERENCES `shop` (`Goods`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_4` FOREIGN KEY (`ShopName2`) REFERENCES `shop` (`ShopName`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+

```

```

MySQL 5.7 Command Line Client

+-----+
| Table | Create Table
+-----+

+-----+
| shop  | CREATE TABLE `shop` (
  `ShopName` varchar(20) NOT NULL,
  `Label` varchar(30) NOT NULL,
  `Cordx` float DEFAULT NULL,
  `Cordy` float DEFAULT NULL,
  `Runtime` varchar(30) NOT NULL,
  `Goods` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`ShopName`),
  UNIQUE KEY `Goods` (`Goods`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+

1 row in set (0.00 sec)

mysql>

```

```

MySQL 5.7 Command Line Client

+-----+
| users | CREATE TABLE `users` (
  `Id` varchar(20) NOT NULL,
  `Age` int(2) NOT NULL,
  `Nickname` varchar(20) NOT NULL,
  `Phone` int(13) NOT NULL,
  `Mima` varchar(20) NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE KEY `Nickname` (`Nickname`),
  UNIQUE KEY `Phone` (`Phone`),
  UNIQUE KEY `Mima` (`Mima`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+

1 row in set (0.00 sec)

mysql>

```

3.8 基本操作测试

3.8.1 控制

语法符合 MySQL 规范

3.8.2 输入

```
INSERT INTO Users VALUES('haha',21,'wo123',1377018242,'asd123')
INSERT INTO Users VALUES('wuyu',18,'3455',1345687242,'dsgsds')
INSERT INTO Users VALUES('3466',21,'cvnn',1397032242,'111111')
```

```
mysql> delete from Users where age=21;
Query OK, 2 rows affected (0.14 sec)
```

```
UPDATE Users
SET Age=24
WHERE Id='haha'
```

3.8.3 输出

```
mysql> select * from users;
```

Id	Age	Nickname	Phone	Mima
3466	21	cvnn	1397032242	111111
haha	21	wo123	1377018242	asd123
wuyu	18	3455	1345687242	dsgsds

```
3 rows in set (0.00 sec)
```

```
mysql> delete from Users where age=21;  
Query OK, 2 rows affected (0.14 sec)
```

```
mysql> select * from users;
```

Id	Age	Nickname	Phone	Mima
wuyu	18	3455	1345687242	dsgsds

```
1 row in set (0.00 sec)
```

```
mysql> select * from users;
```

Id	Age	Nickname	Phone	Mima
3466	21	cvnn	1397032242	111111
haha	24	wo123	1377018242	asd123
wuyu	18	3455	1345687242	dsgsds

```
3 rows in set (0.00 sec)
```

3.9 查询测试

在上述操作过程中已经实现了查询的测试，不再赘述。

4 评价准则

4.1 范围

范围局限于能够正常请求 http 协议的数据。以及符合 SQL 规范的语句。

4.2 数据整理

数据库自动整理生成方式

硬件资源：

笔记本

软件资源：Windows 操作系统

开发平台：python3

框架：flask

数据库管理系统：MySQL

4.3 尺度

输出结果的类型合理。