**校叮当校园app项目详细设计报告**



|  |  |  |
| --- | --- | --- |
| 文件状态  【 】草稿  【 】正式发布  【√】正在修改 | 文件标识 | SE-2018春-G15-校叮当校园app详细设计报告 |
| 当前版本 | 0.1.0 |
| 作者 | 陈佳敏、徐毓茜、江亮儒、马益亮 |
| 完成日期 | 2018-5-16 |

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 参与者 | 起止日期 | 备注 |
| 0.1.0 | 陈佳敏、徐毓茜、江亮儒、马益亮 | 2018-5-14至2018-5-16 | 完成详细设计，查找模板，各方面进行讨论书写。 |

**目录**

[一、 引言 4](#_Toc6056)

[1.1编写目的 4](#_Toc7894)

[1.2项目背景 4](#_Toc32652)

[1.3定义 5](#_Toc27415)

[1.4参考资料 6](#_Toc28195)

[二、 系统结构 7](#_Toc2025)

[三、 模块设计说明 13](#_Toc13828)

[3.1模块描述 14](#_Toc19673)

[3.2功能 14](#_Toc10905)

[3.3性能 15](#_Toc22925)

[3.3.1精度 15](#_Toc10054)

[3.3.2时间特性要求 16](#_Toc21703)

[3.3.3灵活性 16](#_Toc21747)

[3.4输入项 16](#_Toc5205)

[3.5输出项 17](#_Toc18603)

[3.6设计方法 19](#_Toc32166)

[3.7逻辑流程 20](#_Toc11374)

[3.8接口 21](#_Toc11228)

[3.9存储分配 28](#_Toc27503)

[3.10注释设计 28](#_Toc3984)

[3.11限制条件 29](#_Toc25932)

[3.12测试计划 30](#_Toc7595)

[3.13尚未解决的问题 30](#_Toc2055)

[四、 伪代码 31](#_Toc3575)

[4.1用户模块 31](#_Toc28362)

[4.2搜索模块 33](#_Toc28509)

[4.3商品模块 34](#_Toc21429)

[五、 PAD图 36](#_Toc18151)

[六、 Jackson方法 37](#_Toc21841)

# 引言

## 1.1编写目的

本文档用于指导软件开发者开发基于安卓的浙江大学城市学院校 叮当校园二手交易app的过程。通过规范开发人员的开发过程达到保证 项目团队完成软件项目的基本要求，提高软件质量的目的。开发者应根 据本文档进行软件开发和编制软件开发文档。本文档是对软件项目开发 人员的基本要求。

## 1.2项目背景

1. 项目的名称

经过需求分析，现为学校学生设计了名为校叮当的校园系统项目。

1. 项目的委托单位

G15小组

1. 项目的用户：

浙江大学城市学院的在校大学生。

1. 项目的任务提出者

G15小组

1. 项目的主要承担部门

G15小组

1. 项目背景

随着中国互联网用户群的日益庞大，互联网产业正扮演着市 场经济的重要角色。与此同时，信息时代的来临彻底改变了人们 传统的通信手段。从寄信到打电话，从打电话到通话手机，从通 话手机到能获取大信息量的智能手机，人们获取信息的主要手段 发生了质的变化。人们不会仅仅满足于手机只有通话与发短信的 功能，而更多是把手机当成接受讯息的工具，了解时事的工具， 建立社交网络的工具，便利生活的工具。而大学生对于各种信息 平台的接受度和依赖度普遍非常强，但是市面上的生活服务类应 用软件鱼龙混杂，针对大学生的需求，以大学生的利益为基本出 发点提供信息方便大学生的学习生活的应用软件还少之又少。互 联网技术和移动智能终端的飞速发展，促使越来越多的高校学生 更愿意通过移动智能终端平台连接互联网。于是针对大学生群体 的校园APP也诞生了。与此同时，我们应于课程要求，需要在 一个学期内开发项目，并在期末顺利结束，于是经过小组沟通， 我们决定做校叮当这样一个项目，以此项目计划书来展示整体计 划。

1. 软件系统与机构的关系

本软件不需外包，由G15小组独立研发。

## 1.3定义

|  |
| --- |
| 技术定义 |
| **JavaScript:** 这是一种直译式脚本语言，内置支持类型。浏览器通过 JavaScript 向服务器发出建立 WebSocket 连接的请求，  连接建立以后，客户端和服务器端就可以通过 TCP 连  接直接交换数据。 |
| **CSS（**界面优化）：通过CSS进行优化。 |
| **WebSocket API**:WebSocket是HTML5开始提供的一种在单个TCP连接 上进行全双工通讯的协议，在WebSocket API中，浏  览器和服务器只需要做一个握手的动作，然后，浏览  器和服务器之间就形成了一条快速通道。两者之间就  直接可以数据互相传送。在实现WebSocket连线过程  中浏览器发出WebSocket连线请求，然后服务器发出  响应，这个过程称为“握手”。 |
| **Node.js**（建设网站）： Node.js是一款web框架，用来建设网站。  简单的说 Node.js 就是运行在服务端的  JavaScript。Node.js 是一个基于Chrome  JavaScript 运行时建立的一个平台。Node.js  是一个事件驱动I/O服务端JavaScript环境，  基于Google的V8引擎，V8引擎执行  Javascript 的速度非常快，性能非常好。  Node.js就是让JavaScript(js)可以实现服务  器上的开发。JavaScript(前端语言)+node.js(JavaScript的运行环境)=实现服务器上的开发(后端技术)，也就是说js+node.js实现了后端开发的技术。 |

**C/S结构**：全称Client/Server,即客户机/服务器降低系统通讯开销。

**超级管理员**：对APP内交易的用户进行管理的人员，根据实际需要 对管理员赋予适当的权限。

**用户名ID**：用户输入的用于登录校园交易APP的身份标识。

**管理员身份验证**：根据管理员名和密码判断管理员是否为系统合法 管理员。

**数据有效性验证**：对用户输入的数据进行检查，判断是否合乎系统 要求。

## 1.4参考资料

1.朴灵. 深入浅出NodeJS[M]. 第1版. 人民邮电出版社出版社, 2013.

2.马尔奇·哈弗贝克. JavaScript编程精解[M]. 第2版. 机械工程出 版社, 2016.

1. ustbhuangyi. Vue.js 高仿饿了么外卖APP 收藏 [EB/OL]. [2018-3-30 3:30pm].

<https://coding.imooc.com/class/chapter/74.html#Anchor>.

4.计算分院老师. 数据库系统设计与开发[M]. 第1版. 浙江大学城市 学院, 2017.

5.上野宣. 图解HTTP[M]. 第1班. 人民邮电出版社, 2014.

6.网页： 模板：<https://blog.csdn.net/zerodanny/article/details/2077013>

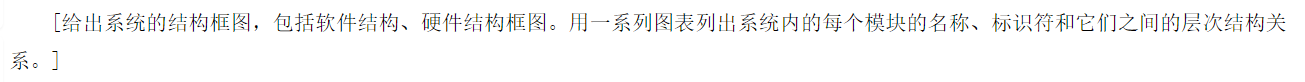
[2018/5/13 9:00am].

书写参考：<http://wxphp.com/wxd_5l6lo8742n1x2cx44e354ncj33s2bw019q3_6.html>

<https://wenku.baidu.com/view/f6ec96c46137ee06eff918e0.html>

<https://www.xzbu.com/1/view-6335577.htm>

# 系统结构



**系统程序功能层级图**



**表1 程序功能与模块的关系**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 用户登录 | 收藏单 | 订单 | 用户信息 | 评论模块 | 管理员登陆 | 商品信息 |
| 搜索商品 | √ |  |  |  |  |  | √ |
| 上架商品 | √ |  |  | √ |  |  | √ |
| 发布请求 | √ |  |  | √ |  |  |  |
| 收藏商品 | √ | √ |  | √ |  |  | √ |
| 发布评论 | √ |  |  | √ | √ |  | √ |
| 修改用户信息 | √ |  |  | √ |  |  |  |
| 订购商品 | √ |  | √ | √ |  |  | √ |
| 查询订单 | √ |  | √ |  |  |  |  |
| 修改商品信息 | √ |  |  |  |  |  | √ |
| 维护数据  库信息 |  |  |  | √ |  | √ | √ |
| 删除用户 |  |  |  | √ |  | √ | √ |
| 订单处理 |  |  | √ |  |  |  |  |

用户：

1. 搜索商品：根据商品的一部分物品名或者一部分关键词，调用商品模糊查询模块进行检索；



**图2搜索物品程序**

1. 上架商品：需要用户登录，需要用户输入正确的商品信息，系统也会登记上架人的信息；



**图3 上架商品程序**

1. 发布请求：需要用户登录，需要用户输入请求内容（如拼车，拼单买东西等），系统也会登记发布者以及参与者的信息；



**图4 发布请求程序**

1. 收藏：需要用户登录，会自动修改用户收藏界面的信息，读取对应商品的信息并修改其收藏量的值；



**图5 收藏商品程序**

1. 发布评论：需要用户登录，需要用户输入请求评论内容，系统也会登记评论者及其评论的商品；



**图6 评论商品程序**

1. 修改用户信息：需要用户登录，修改部分用户信息，包括学号，密码，手机号，邮箱，照片等信息，来保证用户信息的正确性；



**图7 修改用户信息程序**

1. 订购商品：需要用户登录，用户选择需要订购的商品后，系统会登记订购者以及所订购商品的信息，制作成订单放在订单表中；



**图8 订购商品程序**

1. 查询订单：需要用户登录，即可访问已购买或者已出售的订单信息；



**图9 查询订单程序**

1. 修改商品信息：需要用户登录，即可修改自己上架的商品的信息，但对于已经订购的商品无法进行修改；



**图10 修改商品信息程序**

1. 维护数据库信息：需要管理员登录，手动修改数据库中产生错误的部分，并分析产生错误的原因，修改程序；



**图11 维护数据库信息程序**

1. 删除用户：需要管理员登录，管理员手动删除严重违规违法的用户账号，并清除其上架的商品信息；



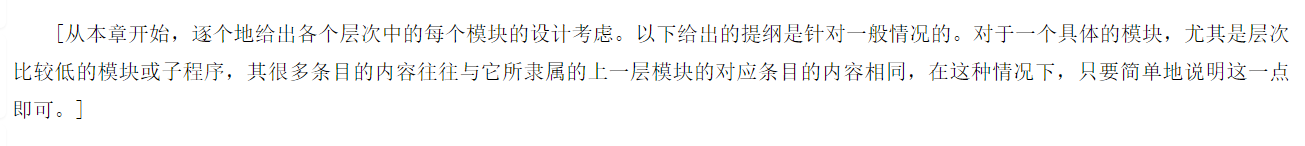
**图12 删除用户程序**

1. 订单处理：系统自动对生成的订单进行处理，根据订单中的用户信息和商品信息分别向买家和卖家发出订单成功生成的信息，并提醒用户在约定时间，约定地点进行交易；

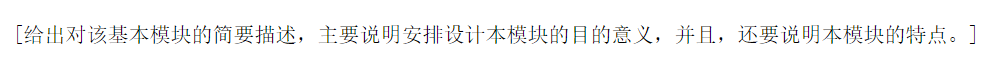


**图13 订单处理程序**

# 模块设计说明

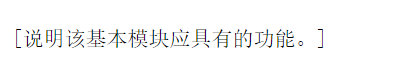


## 3.1模块描述



本程序包括用户搜索物品、加入收藏、发布评论、提交订单四个子模块，其中搜索商品、提交订单模块是有着前后的逻辑顺序，即后面的以前面的为基础，其目的是为了实现用户搜索商品、收藏商品、评价商品、购买商品的功能。本程序的特点是非常驻内存、是子程序、不可重人的、无覆盖要求、是顺序处理。

## 3.2功能





**图14 用户购买商品程序IPO图**

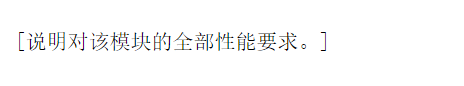
1. 搜索商品模块输入的是商品的部分名字或者部分关键词信息，然后经过系统对数据库查找，输出与之匹配的物品信息。

2. 在用户搜索商品之后，对于喜欢的商品可以加入收藏，便于下次购买。

3. 用户对于商品可以进行评论，系统会上传用户的评论信息，并发布在商品的评论区中。

4. 用户对于喜欢的商品可以直接购买，购买之后会生成订单通知买家和卖家双方。

## 3.3性能



### 3.3.1精度

查询物品请求信息：文本

有关价格信息：货币人民币，整数，最高位数为6位

有关日期时间信息：年-月-日 时：分：秒格式

输出信息全部为文本信息。

### 3.3.2时间特性要求

用户搜索，系统及时响应（<=3s）。

加入收藏，系统响应时间（<=3s）

发布评论，系统响应时间（<=3s）

用户提交订单，系统及时处理（<=3s）。

### 3.3.3灵活性

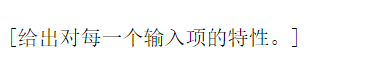
1．操作方式上的变化：用户可以根据关键字对物品信息进行模 糊查询。

2．同其他软件的接口的变化：数据库接口发生变化时，修改相 应的模式接口，以保证页面显示不变。

3．精度和有效时限的变化：数据输入精度不符合要求就会阻止 输入，并提示输入格式错误

4. 计划的变化或改进：对业务要求的变化，团队统一讨论、 修改、确定并执行。

## 3.4输入项



**表2 用户购物品程序输入项**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 名称 | 标识 | 格式类型 | 有效范围 | 输入方式 | 输入媒体 | 数据来源 | 安全保密 |
| 商品名 | Commodity\_name | varchar | 长度不超过30 | 用户输入 | 键盘 | 用户输入 | 无 |
| 商品关键词 | Commodity\_key | Varchar | 长度不超过30 | 用户输入 | 键盘 | 用户输入 | 无 |

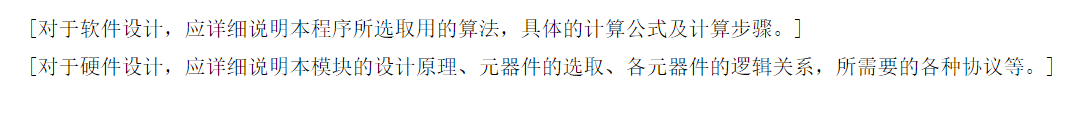
## 3.5输出项



**表3 用户购物程序输出项**

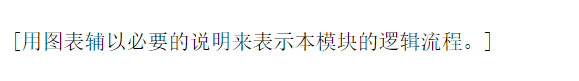
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 标识 | 格式类型 | 有效范围 | 输出方式 | 输出媒体 | 安全保密 |
| 商品名 | Commodity\_name | varchar | 长度不超过30 | 系统输出 | 屏幕 | 无 |
| 商品关键词 | Commodity\_key | Varchar | 长度不超过30 | 系统输出 | 屏幕 | 无 |
| 商品编号 | Commodity\_id | int |  | 系统输出 | 屏幕 | 无 |
| 商品图片 | Commodity\_picture | Varchar | 长度不超过100 | 系统输出 | 屏幕 | 无 |
| 价格 | Commodity\_price | int |  | 系统输出 | 屏幕 | 无 |
| 上架者id | User\_id | int |  | 系统输出 | 屏幕 | 无 |
| 商品简介 | Commodity\_intro | varchar | 长度不超过100 | 系统输出 | 屏幕 | 无 |
| 商品收藏数 | Commodity\_scl | int |  | 系统输出 | 屏幕 | 无 |
| 商品评论数 | Commodity\_pll | int |  | 系统输出 | 屏幕 | 无 |
| 商品上架时间 | Commodity\_uptime | datatime |  | 系统输出 | 屏幕 | 无 |
| 商品上架所需积分 | Commodity\_score | Int |  | 系统输出 | 屏幕 | 无 |

## 3.6设计方法



无特殊算法

## 3.7逻辑流程



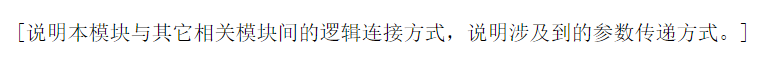


**图 15 用户购物品流程图**

以上流程图是用户购物品流程，用户首先登录系统，然后进入首页搜索商品，可以按照商品的部分名字或部分关键词搜索商品，然后对商品进行收藏或者评论，然后如果想继续购物品可以重复上面的操

作，如果用户已经买够了，最终将订单提交，交由系统处理。

## 3.8接口



**表4 检索物品**

|  |  |
| --- | --- |
| 执行条件 | 输入搜索内容，部分物品名、部分关键词 |
| 属性 | var key;  var connection=mysql.createConnection({  host:'localhost',  user:'root',  password:'zucc',  port:'3306',  database:'teamwork'  });  sql="select \* from commodity where commodity\_name like ? or commodity\_key like ?";  sqlparams=["%"+key+"%","%"+key+"%"]; |
| 方法 | connection.query(sql,sqlparams,function(err,result){  if(err){  console.log(err.message);  return;  }  else{  console.log(JSON.stringify(result));  res.end(JSON.stringify(result));  }  }) |

表5 加入收藏

|  |  |
| --- | --- |
| 执行条件 | 进入商品页面点击收藏按钮 |
| 属性 | var sc\_price,commodity\_id,user\_id;  var t1=new Date().toLocaleDateString();  var t2=new Date().toLocaleTimeString();  var t3=t1+" "+t2;  sql="insert into sc(sc\_time,sc\_price,commodity\_id,user\_id) values(?,?,?,?)";  sqlparams=[t3,sc\_price,commodity\_id,user\_id];  var connection=mysql.createConnection({  host:'localhost',  user:'root',  password:'zucc',  port:'3306',  database:'teamwork'  }); |
| 方法 | connection.query(sql,sqlparams,function(err,result){  if(err){  console.log(err.message);  return;  }  else{  res.write("收藏成功");  res.end();  }  }) |

表6 发布评论

|  |  |
| --- | --- |
| 执行条件 | 进入商品的评论区输入评论内容并点击发布评论 |
| 属性 | var commodity\_id,user\_id,pl\_content;  var t1=new Date().toLocaleDateString();  var t2=new Date().toLocaleTimeString();  var t3=t1+" "+t2;  sql="insert into pl(commodity\_id,user\_id,pl\_content,pl\_time) values(?,?,?,?)"; sqlparams=[commodity\_id,user\_id,pl\_content,t3];  var connection=mysql.createConnection({  host:'localhost',  user:'root',  password:'zucc',  port:'3306',  database:'teamwork'  }); |
| 方法 | connection.query(sql,sqlparams,function(err,result){  if(err){  console.log(err.message);  return;  }  else{  connection.query(sql,sqlparams,function(err,result){  if(err){  console.log(err.message);  return;  }  else{  res.write("评论发布成功");  res.end();  }  })  }  }) |

各个模块与主模块会员购物品模块及其他模块的关系如图所示：



**图16模块关系图**

以上四个模块与主模块及其他模块的关系如下：

**a.搜索商品模块**

①直接被主模块用户购物品模块调用，生成物品检索页面，用户在文本框中输入检索内容，页面显示物品信息；

②该模块通过链接调用评论商品模块，收藏商品模块，提交订单模块，用户可以收藏商品，发布评论，购买商品。

**b.评论商品模块**

①通过检索模块被主模块调用，把商品信息和用户信息添加到评论模块；

②在评论模块中输入评论内容，并且对应用户信息和商品信息，生成评论信息，发布在商品评论区中

**c.收藏模块**

①通过检索模块被主模块调用，把商品信息添加到收藏模块；

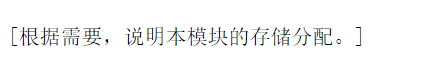
②收藏模块将商品的简要信息和链接生成一个小模块放在用户的收藏栏中

**d.提交订单模块**

①通过检索模块被主模块调用，把商品信息和用户信息添加到提交订单模块；

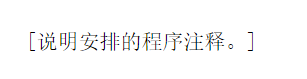
②该模块结合用户信息和商品信息生成订单，将生成的订单发到买家和卖家的客户端。

## 3.9存储分配



无特殊要求

## 3.10注释设计



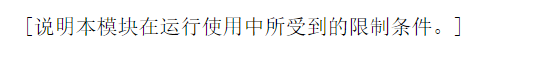
为每一个类编写类注释，说明本该类的作用、使用方法、作者列表、当前版本和完成时间、参考类；

为每一个方法编写方法注释，说明该方法的作用、输入参数的作用和取值范围、返回值、抛出的异常；

为每一个属性编写属性注释，说明其作用、取值范围、注意事项，为局部变量编写行末注释；

对使用的循环、分支、抛出异常块编写注释，说明该逻辑实现的功能。

## 3.11限制条件



**a.检索物品信息**

1. 输入的检索信息不能为空；
2. 输入的检索信息必须是由若干个中文、英文、数字组成的组合，组合最大长度为30

**b.提交订单**

订单中物品数量不能为0。

**3.12测试计划**

**a.测试技术要求**

1.进行人工静态检查，保证代码的逻辑正确性、清晰性、规范性、一致性；

2.进行动态测试，通过设计测试用例，执行待测程序来跟踪比较实际结果与预期结果来发现错误或缺陷。

**b.检索物品**

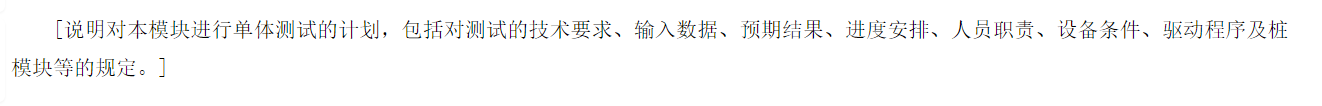
1.测试搜索框的输入数据长度是否符合**3.11**中的规定，如果有不符合，提示该项“格式错误”，若符合，则满足要求；

2.测试搜索框的内容是否为空，若为空，提示“该框不能为空”，若不为空，则满足要求。

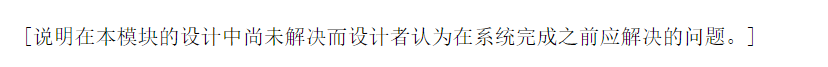
**c.提交订单**

测试订单中物品数量是否为0，若为0，提示“物品数量不能为0”，若大于0，则满足要求。

## 3.12测试计划



## 3.13尚未解决的问题



无

# 伪代码

## 4.1用户模块

#### 1.1用户注册界面

|  |  |  |
| --- | --- | --- |
| 用户注册 | | |
| 账号输入\* | 【user\_id】 | 【btn检测账号】 |
| 密码输入\* | 【user\_pw】 | |
| 确认密码\* | 【user\_pw】 | |
| 手机号\* | 【user\_phone】 | |
| 【btn确认注册】 | | |

If（账号==空）{

提示：密码不能为空

}

else{

useridbtn 检验数据库中是否有账号相同的情况

if（用户名存在）{

提示：用户名已经存在

}

}

if（密码==空）{

提示：密码不能为空

}

else if（密码输入不符合规范）{

提示：请按照标准输入密码

}

if（确认密码跟原密码不同）{

提示：两次密码输入不一致，请重新输入确认密码

}

if（手机号==空）{

提示：手机号不能为空

}

#### 1.2用户登录模块

|  |  |  |
| --- | --- | --- |
| 用户登陆 | | |
| 账号输入\* | 【user\_id】 |
| 密码输入\* | 【user\_pw】 【btn忘记密码？】 | |
| 【btn注册账户】 | 【user\_pw】 | |
| 【btn确认登陆】 | | |

if（（用户名输入 || 密码输入） == 空）{

提示：用户名，密码不能为空

}

if（忘记密码btn）{

进入忘记密码界面

}

if（确认登录btn）{

if（该账号在数据库里不存在）{

提示：帐号密码错误

}else if（该账号所对密码不正确）{

提示：帐号密码错误

}

返回用户界面

}

#### 1.3用户个人信息界面

|  |  |
| --- | --- |
| 用户个人信息界面 | |
| 照片 | 【user\_picture】 |
| 用户名 | 【user\_name】 |
| 生日 | 【user\_birth】 |
| 手机号 | 【user\_phone】 |
| 邮箱地址 | 【user\_email】 |
| 拥有积分 | 【user\_score】 |
| 学号 | 【user\_sid】 |
| 个人简述 | 【user\_resume】 |
| 【设置btn】 | |

if（邮箱格式不规范）{

提示：邮箱格式输入错误，请重新输入

}

modifybtn

if（用户名在数据库中存在）{

提示：用户名存在

}

## 4.2搜索模块

|  |  |
| --- | --- |
| 搜索栏 | 分类搜索btn |

#### 2.1全文检索

input 商品全称or商品关键字

查询数据库

if（商品在数据库中不存在）{

return false

主界面显示‘搜索失败’界面

} else{

返回商品信息对应的json数据

return true

主界面显示商品信息

}

#### 2.2分类搜索

分类搜索btn点击

跳转页面

|  |  |  |  |
| --- | --- | --- | --- |
| 主分类 | 子分类 | | |
| 分类一 | 分类一：子分类一 | 分类一：子分类二 | 分类一：子分类三 |
| 分类二 |
| 分类三 | 分类一：子分类四 | 分类一：子分类五 | 分类一：子分类六 |
| 分类四 |
| 分类五 | 分类一：子分类七 | 分类一：子分类八 | 分类一：子分类九 |
| 分类六 |

点击主分类

子分类页面跳转

点击子分类

跳转对应分类商品界面

## 4.3商品模块

#### 3.1商品发布模块

|  |  |
| --- | --- |
| 商品名称\* | 【commodity\_name】 |
| 商品价格 | 【commodity\_price】 |
| 商品关键词 | 【commodity\_key】 |
| 商品图片  【commodity\_picture】 | |
| 商品介绍  详细描述：【commodity\_intro】 | |
| 【取消发布btn】 【确认发布btn】 | |

if（某一项==空）{

提示：该项不能为空

}

if(commodity\_name不是由中文、英文、数字组成 ){

提示：商品格式错误

}

if(commodity\_key不是由中文、英文、数字组成 ){

提示：关键词格式错误

}

if(commodity\_price不是由数字组成 ){

提示：价格格式错误

}

if(commodity\_picture 根据提供的图片路径未找到该图片 ){

提示：图片不存在

}

if（点击确认发布btn && 上述条件都满足）{

提示：上架成功

该商品数据被存在数据库中

commodity\_id++

货架界面新增该商品简略信息

}

if（点击取消发布btn）{

返回上一级界面

}

#### 3.2货架界面

|  |  |
| --- | --- |
| 我的货架 | |
| 商品一 | 商品二 |
| 商品三 | 商品四 |
| 【下架商品btn】 | |

if（商品发布成功）{

货架新增该商品信息

}

if( 点击下架商品 btn)｛

If（商品已被订购）｛

下架商品失败

提示：该商品已被用户订购

｝

Else{

在数据库中删除该商品信息

在货架上删除该商品信息

提示:下架成功

}

｝

#### 3.3收藏界面

|  |
| --- |
| 收藏商品一 |
| 收藏商品二 |
| 收藏商品三 |

if（点击收藏商品 btn）{

修改数据，该商品的收藏字段数值+1

在收藏栏显示该收藏商品简略信息

}

If(在收藏栏点击某收藏商品 btn){

跳转到该商品的详细信息

}

if(点击取消收藏商品 btn)｛

修改数据库，该商品的收藏字段的数值-1；

删除收藏界面该商品的信息

｝

#### 3.3评论界面

if(点击发布评论 btn)｛

if（pl\_content 内容为空）｛

发布评论失败

提示：不能发布空评论

｝

else{

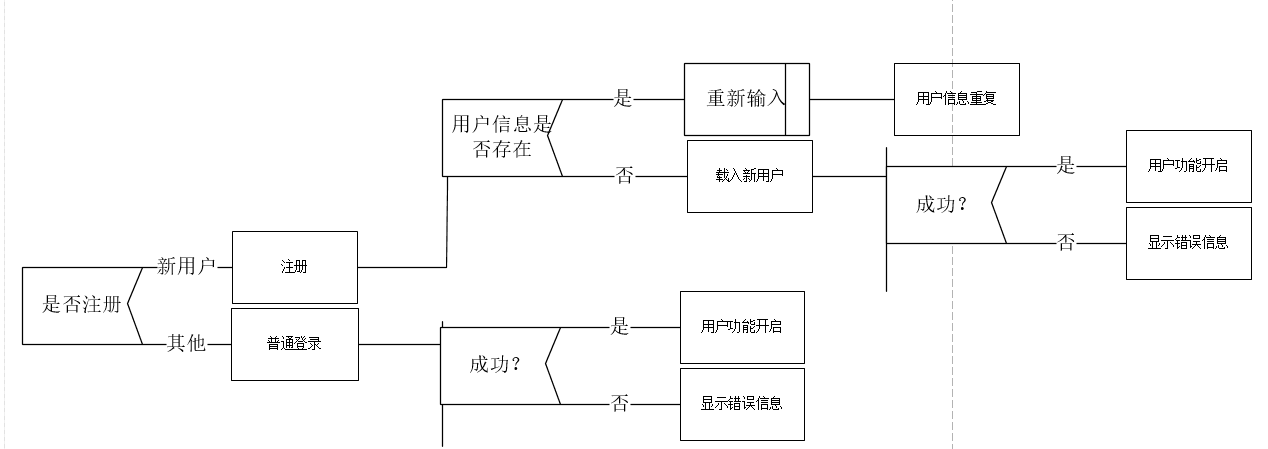
修改数据库，该商品的评论数数值+1，将该评论的内容保存到数据库

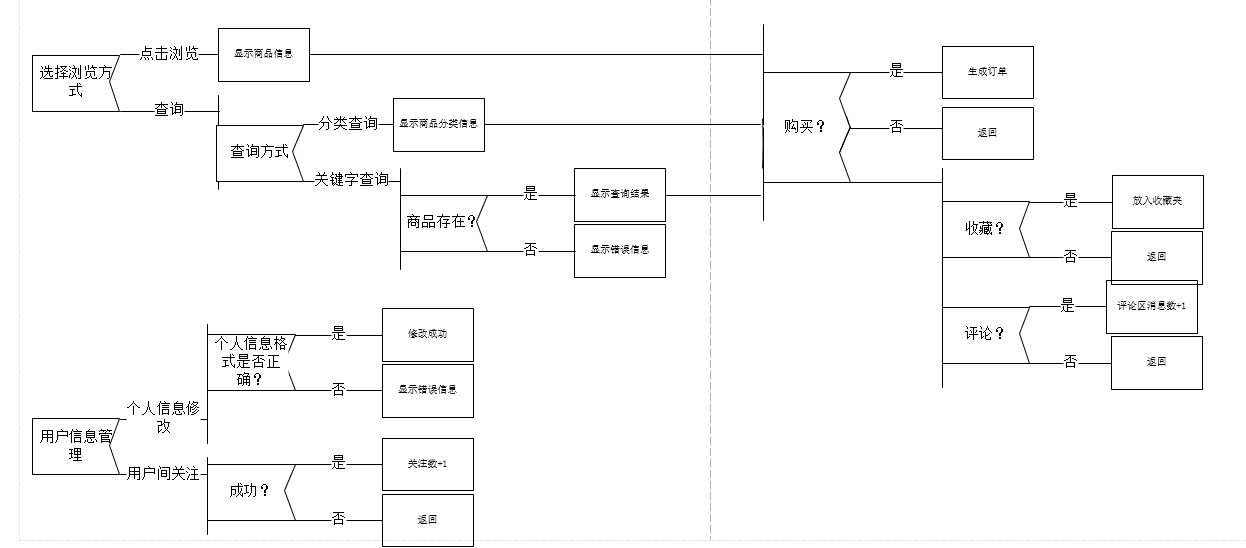
提示：评论成功

}

｝

# PAD图





# Jackson方法