**浙江大学城市学院**

**系统测试报告**



组序： G23 ；

组长： 乔寒月 ；

组内成员： 李欣飏 ；

指导老师： 杨枨 ；

提交日期： 2019/5/26 ；

版本控制

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 批准人 | 起止日期 | 备注 |
| 1.0 | 乔寒月 | 李欣飏 | 杨枨 | 起：  2019/5/21  止：  2019/5/26 |  |
| 1.1 | 乔寒月 | 李欣飏 | 杨枨 | 起：  2019/5/28  止：  2019/5/31 |  |
| 1.2 | 乔寒月 | 李欣飏 | 杨枨 | 起：  2019/6/11  止：  2019/6/16 |  |

[1.引言 4](#_Toc10241008)

[1.1编写目的 4](#_Toc10241009)

[1.2背景 4](#_Toc10241010)

[1.3定义 4](#_Toc10241011)

[1.4参考资料 4](#_Toc10241012)

[2测试概要 5](#_Toc10241013)

[2.1测试目的 5](#_Toc10241014)

[2.2测试类别 5](#_Toc10241015)

[2.3测试模块 5](#_Toc10241016)

[2.4测试管理 7](#_Toc10241017)

[2.4.1过程管理 7](#_Toc10241018)

[2.4.2配置管理 7](#_Toc10241019)

[3测试结果及发现 8](#_Toc10241020)

[3.1单元测试 8](#_Toc10241021)

[3.2集成测试 8](#_Toc10241022)

[3.3确认测试 8](#_Toc10241023)

[3.3.1 α测试 8](#_Toc10241024)

[3.3.2 β测试 9](#_Toc10241025)

[3.4白盒测试 9](#_Toc10241026)

[3.5黑盒测试 10](#_Toc10241027)

[4测试总结 11](#_Toc10241028)

[4.1测试时间 11](#_Toc10241029)

[4.2测试范围 11](#_Toc10241030)

[4.3测试问题与风险分析 11](#_Toc10241031)

[4.4测试结果分析 11](#_Toc10241032)

[5综合评价 11](#_Toc10241033)

[5.1软件能力 11](#_Toc10241034)

[5.2建议 12](#_Toc10241035)

[5.3缺陷和限制 12](#_Toc10241036)

[5.4测试消耗资源 12](#_Toc10241037)

# 1.引言

## 1.1编写目的

目的：

说明对程序系统的设计考虑，包括程序系统的模块设计说明

包括模块描述、功能说明、性能解释、输出项输入项，设计方法（算法）流程逻辑 接口，存储分配，注释设计、限制条件、测试计划以及尚未解决的问题进行详细的说明，为实现代码做好充足的准备

预期读者：

软件制作小组成员：乔寒月、李欣飏

软件用户代表：杨枨老师、信管1702 郭伊娜、统计1801 朱芳颖。

## 1.2背景

软件名称：承闲二手购物小程序；

任务提出者：李欣飏；

开发者：乔寒月、李欣飏；

用户：杨枨老师、信管1702 郭伊娜、统计1801 朱芳颖。

## 1.3定义

暂无。

## 1.4参考资料

张海藩、牟永敏，软件工程导论（第六版），清华大学出版社

乔寒月、李欣飏、吴智宏，项目计划书

乔寒月、李欣飏、吴智宏，可行性分析报告

乔寒月、李欣飏、吴智宏，软件需求说明

乔寒月、李欣飏、吴智宏，总体设计报告

乔寒月、李欣飏，详细设计报告

# 2测试概要

## 2.1测试目的

1.验证软件是否满足项目计划、需求说明；

2.通过测试，发现软件缺陷；

3.为软件产品的质量测量和评价提供依据。

## 2.2测试类别

1.单元测试

2.集成测试

3.确认测试

4.白盒测试

5.黑盒测试

## 2.3测试模块

|  |  |  |
| --- | --- | --- |
| 测试模块：首页 | | |
| 标识符：1 | | |
| 测试项：   1. 用户通过输入关键字查询包含关键字的商品信息 2. 用户通过划动屏幕进行全部商品浏览 | | |
| 测试环境：微信 | | |
| 前置条件：能够运行微信小程序 | | |
| 操作步骤：  1.点击输入框输入关键字，点击搜索，浏览包含关键字的商品  2.划动屏幕浏览全部商品 | | |
| 输入数据 | 期望输出 | 实际结果 |
| 关键字 | 包含关键字的全部商品 | 包含关键字的全部商品 |
| 划动屏幕 | 全部商品 | 全部商品 |

|  |  |  |
| --- | --- | --- |
| 测试模块：分类 | | |
| 标识符：2 | | |
| 测试项：  1.用户通过输入关键字查询包含关键字的商品信息  2.用户通过点击分类图标查询该分类全部商品 | | |
| 测试环境：微信 | | |
| 前置条件：能够运行微信小程序 | | |
| 操作步骤：  1.点击输入框输入关键字，点击搜索，浏览包含关键字的商品  2.点击分类图标 | | |
| 输入数据 | 期望输出 | 实际结果 |
| 关键字 | 包含关键字的全部商品 | 包含关键字的全部商品 |
| 点击分类图标 | 该分类全部商品 | 该分类全部商品 |

|  |  |  |
| --- | --- | --- |
| 测试模块：购物车 | | |
| 标识符：3 | | |
| 测试项：  1.用户在商品详细信息界面进行加购，在购物车显示  2.用户通过点击增/减按钮对加购商品进行操作，左滑删除商品。 | | |
| 测试环境：微信 | | |
| 前置条件：能够运行微信小程序 | | |
| 操作步骤：  1. 用户在商品详细信息界面点击加购按钮进行加购  2.用户在购物车界面点击删除、增/减按钮对加购商品进行操作 | | |
| 输入数据 | 期望输出 | 实际结果 |
| 点击加购 | 加购商品在购物车显示 | 加购商品在购物车显示 |
| 左滑删除 | 删除加购商品 | 删除加购商品 |
| 点击增/减 | 增/减加购商品（大于0、小于商品总数量） | 增/减加购商品（大于0、小于商品总数量） |

|  |  |  |
| --- | --- | --- |
| 测试模块：个人中心 | | |
| 标识符：4 | | |
| 测试项：  1.点击我的订单，查看全部订单  2.待收货：显示已经提交还未收货的订单  3.待评价：显示已经收货还未评价的订单  4.申诉：显示正在申诉的订单，及申诉状态  5.我的收藏：显示收藏商品  6.我的评价：显示自己的全部评价  7.发布闲置  8.我的闲置：显示自己发布的全部商品  9.实名认证：显示自己的实名信息 | | |
| 测试环境：微信 | | |
| 前置条件：能够运行微信小程序 | | |
| 操作步骤：   1. 点击相应图标查看我的订单、待收货、待评价、申诉、我的收藏、我的评价、我的闲置 2. 点击发布闲置，输入相关数据，点击确定 3. 点击实名认证，输入相关数据，点击确定 | | |
| 输入数据 | 期望输出 | 实际结果 |
| 点击全部订单 | 显示全部订单 | 显示全部订单 |
| 点击待发货 | 显示还未发货订单 | 显示还未发货订单 |
| 点击待收货 | 显示还未收货的订单 | 显示还未收货的订单 |
| 点击已收货 | 显示已收货订单 | 显示已收货订单 |
| 点击我的收藏 | 显示收藏商品 | 显示收藏商品 |
| 点击我的评价 | 显示自己的全部评价 | 尚未显示 |
| 点击发布闲置 | 显示发布界面 | 显示发布界面，但还不能进行操作 |
| 点击我的闲置 | 显示自己发布的全部商品 | 尚未显示 |
| 点击实名认证 | 显示自己的实名信息 | 尚未显示 |
| 点击收货地址 | 显示收货地址 | 显示收货地址 |
| 点击反馈申诉 | 显示申诉页面 | 显示申诉页面 |

## 2.4测试管理

### 2.4.1过程管理

软件测试人员配备情况

测试项目负责人：乔寒月

测试分析员：李欣飏

测试设计员：李欣飏 乔寒月

测试程序员：李欣飏 乔寒月

测试员：李欣飏 乔寒月

测试系统管理员：李欣飏 乔寒月

配置管理员：李欣飏 乔寒月

具备的准入条件：

1. 具有项目计划
2. 具有总体设计 详细设计 项目需求文档
3. 所提交的被测试文件受web微信开发者工具控制
4. 软件源代码正确通过编译

尚未具备准出条件

### 2.4.2配置管理

配置管理工具：GitHub。

# 3测试结果及发现

## 3.1单元测试

1.编写目的：

（1）对单元测试结果进行整理汇总，形成正确的测试文档

（2）为软件的评审验收提供依据

（3）纳入软件产品配置管理库

2.软件单元描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 被测试单元 | 与之相关的单元 | 所属子系统 | 单元需求 |
| cart.json | cat类 | pages | 购物车功能函数编写 |
| cart.wxml | cart类 | pages | 购物车界面的的组件方位正确摆放 |
| cart.wxss | cart类 | pages | 购物车界面样式显示 |
| details.json | details类 | goods | 商品详情的功能函数 |
| details.wxml | details类 | goods | 商品详情页的组件方位格式正确 |
| details.wxss | details类 | goods | 商品详情页的色彩样式正确 |
| details.js | details类 | goods | 商品详情页的调用函数正确 |
| index.wxss | index类 | pages | 首页界面的色彩样式正确 |
| index.wxml | index类 | pages | 首页界面的组件方位格式正确 |
| index.json | index类 | pages | 首页界面的功能函数 |
| shopList.wxml | shopList类 | pages | 已经加入购物车的商品清单排列 |
| shopList.wxss | shopList类 | pages | 已经加入购物车的商品清单色彩样式显示 |
| seller-order  .wxml | seller-order  类 | pages | 个人中心界面的的组件方位正确摆放 |
| seller-order  .wxss | seller-order  类 | pages | 个人中心界面样式显示 |
| seller-order  .json | seller-order  类 | pages | 个人中心功能函数编写 |

测试工具集：

[miniprogram-simulate](https://github.com/wechat-miniprogram/miniprogram-simulate)：

测试代码框架

// /test/components/index.test.js

const simulate = require('miniprogram-simulate')

test('components/index', () => {

const id = simulate.load('/components/index') // 此处必须传入绝对路径

const comp = simulate.render(id) // 渲染成自定义组件树实例

const 类名= document.createElement('parent-wrapper')

comp.attach(类名)

const view = comp.querySelector('.index') // 获取子组件 view

expect(view.dom.innerHTML).toBe('index.properties') // 测试渲染结果

expect(window.getComputedStyle(view.dom).color).toBe('green') // 测试渲染结果

})

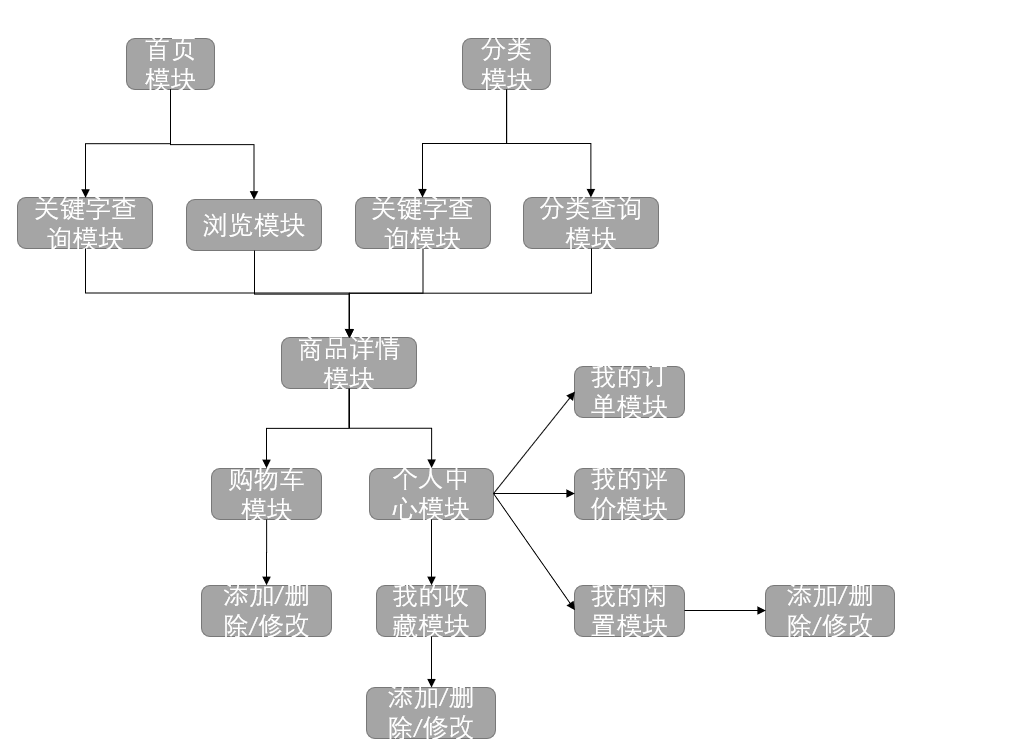
PS：测试工具集中的 wx 对象和内置组件都不会实现真正的功能，如果需要测试一些特殊场景的话，可以自行覆盖掉测试工具集中的 api 接口和内置组件。

PS：目前因为有部分自定义组件功能仍未支持（如抽象节点等），故测试工具暂无法全部覆盖自定义组件的特性，后续会继续完善。

测试工具集中提供了一些方便测试的接口，比如：

* 模拟 touch 事件、自定义事件触发
* 选取子节点
* 更新自定义组件数据
* 触发生命周期

## 3.2集成测试



1.先对主控制模块进行测试，将存根程序代替所有直接附属与主控制模块的模块。

2.每次用一个实际模块代换一个存根程序。

3.在结合进一个模块的同时进行测试

4.在不断进行回归测试

5.重复以上步骤。

未发现各模块接口间存在的问题

## 3.3确认测试

### 3.3.1 α测试

时间：2019/5/26

地点：慕贤楼

人员：乔寒月、李欣飏、郭伊娜

测试背景：在承闲二手购物小程序开发版下进行测试

测试结果：

郭伊娜同学在指导人员的指导下进行测试操作，由于在手机上不能显示数据库内信息，一些功能无法测试，认为界面略微单调和常规，没有视觉冲击，但是可以接受，布局简单易懂，操作方便。

功能：能够完成关键字搜索、分类查询、加购、收藏、查看订单等功能。

局域化：各数据库表是所有功能通用的，各模块之间联系较高

可用性：功能初步完善

可靠性：较低

性能：响应时间可以接受

支持：支持能够运行微信的移动设备

### β测试

尚未进行β测试。

## 3.4白盒测试

路径覆盖：

1. 实名认证→认证成功
2. 实名认证→认证失败
3. 关键字查询→查询等待提示→显示包含关键字商品
4. 关键字查询→查询失败→失败原因
5. 分类浏览→显示该分类商品
6. 分类浏览→浏览失败→失败原因
7. 收藏→收藏成功
8. 收藏→收藏失败
9. 加购→加购成功
10. 加购→加购失败
11. 提交订单→提交成功
12. 提交订单→提交失败→失败原因
13. 查询订单信息→显示订单信息
14. 查询订单信息→查询失败→失败原因
15. 待发货→待发货商品
16. 待收货→待收货商品
17. 已收货→已收货商品
18. 发布商品→发布成功
19. 发布商品→发布失败→失败原因
20. 下架商品→确认下架→下架成功
21. 下架商品→确认下架→下架失败→失败原因
22. 评价→评价成功
23. 评价→评价失败→失败原因
24. 反馈申诉→提交反馈→反馈成功
25. 反馈申诉→提交反馈→反馈失败→失败原因
26. 添加收货地址→输入信息→添加成功
27. 添加收货地址→输入信息→添加失败→失败原因

## 3.5黑盒测试

1.等价类划分

输入等价类：

（1）输入的商品id 评价编号 申诉编号

（2）评价内容 申诉内容 发布闲置的闲置商品描述

（3）购物车增减数目

（4）身份证id

（5）学号id

有效等价类：

与以上等价类一一对应：

（1）1——99999999999999999999

（2）1——2000

（3）1——无穷

（4）18位

（5）00000000——99999999

无效等价类：

（1）<1|| >99999999999999999999

（2）<1||>2000

（3）<1

（4）!=18位

（5）不满足8位数字

2.边界值分析

商品编号、评价编号、申诉编号字符数量取值范围：1<x<20

身份证id字符数量=18

学号id字符数量=8

评价内容和申诉内容闲置商品描述字符数量1<x<2000

3.状态测试

测试用例：

1. 输入的商品id
2. 评价编号
3. 反馈申诉编号

（4） 评价内容

（5） 反馈申诉内容

（6） 发布闲置的闲置商品描述

（7） 购物车增减数目

（8） 身份证id

（9） 学号id

测试数据 测试范围 期望结果

（1）1 1 1 1 1 1 1 1 1 等价类1234567无效等价类89 无效

（2）0 0 0 1 1 1 3 0 1 等价类4567无效等价类12389 无效

（3）1 1 1 1 1 1 1 4225865111119 31508416 等价类123456789 有效

# 4测试总结

## 4.1测试时间

2019/5/26

## 4.2测试范围

本次测试覆盖的范围包括：

1.功能测试

2.数据迁移测试

3.接口测试

4.兼容性测试

5.性能测试

6.安全性测试

## 4.3测试问题与风险分析

用户的实名认证信息需要重点保护。

## 4.4测试结果分析

能够实现购物小程序的基本功能，功能还需进一步完善。

界面可以更加丰富，加入一些元素给人焕然一新的感觉。

# 5综合评价

## 5.1软件能力

目前我们的微信购物小程序可以满足在安卓和IOS系统的手机上进行发布，只需要通过关注我们的公众号：乘闲 就可以使用我们的关联小程序了。

在该小程序中总共分为四个模块：首页、分类、购物车、我的

用户可以在首页模块中看到一些推荐的商品连接，直接点击图片即可进入商品的详情页

在商品详情页中，用户可以选择收藏商品或者加入购物车

在分类模块中，用户可以看到各种已经分好类的商品，可以根据自己的需求进行商品选择

在购物车模块中，用户可自行添加商品数量，自动显示总价，并可选择提交订单

在我的模块中，用户可以查看自己的交易进度，选择确认收货或者进行申诉 同时用户也可以评价已经确认收货的商品。用户可以在我的—发布闲置中自行发布闲置商品

在我的—我的闲置中可以更改闲置商品信息或者进行商品的下架处理

## 5.2建议

可以尝试在已有的基础上更加完善尤其是发布闲置的功能，对于上传的照片可以扩大系统的容量

## 5.3缺陷和限制

由于我们的技术还不完善我们的商品分类还不够完善，做出来的界面也比较粗陋不够精细

在商品的通货流动方面还有很大的进步空间

## 5.4测试消耗资源

占用了组内开发人员的大量时间，经常进行开会完善以及修改代码找出bug

十分耗费体力

不断地进行测试，由于只有两个开发人员，难度系数很大，较为吃力，耗费半个月左右的时间进行代码的编写以及开发技术的自我学习