优邻购

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2019/7/11 | <1.0> | sprint1结束，初次编写 | 褚宇轩，邓霄 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 4

4. 测试结果及分析 4

4.1 需求覆盖率及缺陷分布 4

4.2 缺陷严重程度 5

5. 缺陷清单 5

5.1 功能性缺陷 5

5.2 非功能性缺陷 6

6. 测试结论与建议 6

测试报告

# 简介

## 目的

该测试报告将项目的测试过程及结果写成文档，对发现的问题和缺陷进行分析，为纠正软件存在的质量问题提供依据，同时为软件验收和交付打下基础。

## 范围

此报告覆盖的是优邻购web端的管理员部分和测试优邻购系统的后端用户部分。

## 定义、首字母缩写词和缩略语

UserDao：用户数据访问层

OrderInfoDao：订单数据访问层

Id：用户的id

Valid：是否可用

Phone：电话号码

Status：订单执行状态

regDate：注册日期

certification：验证码

password：密码

username：用户名

address：用户地址

gender：性别

latitude：纬度

longitude：经度

storeId：店家Id

storeName：店家名字

orderItemId：商品id

orderItemId：商品名称

amount：数量

price：单价

totalPrice：总价

createDate：订单创建日期

## 参考资料

《sprint1测试用例》

《优邻购软件需求规约》

## 概述

本文档主要包含测试的施行情况和出现的问题

# 测试概要

测试时间：2019/7/10-2019/7/11

测试地点：上海交通大学软件学院3108教室

测试方法：单元测试，集成测试， 黑盒测试

测试人员：王笑然，邓霄，褚宇轩，王志远

测试内容：使用junit对底层访问逻辑进行测试，使用Postman手动对接口进行测试

# 测试环境

软件：Junit，Postman

硬件：笔记本电脑

# 测试结果及分析

## 需求覆盖率及缺陷分布

下面表格中的需求覆盖率针对的是sprint1迭代计划中的需求

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **经销商管理** | 7 | 100% | 0 | 0 |  |
| **店铺管理** | 7 | 100% | 0 | 0 |  |
| **商品信息管理** | 4 | 50% | 2 | 50% | 第二次迭代解决 |
| **登录** | 4 | 100% | 0 | 0 |  |
| **用户查询** | 4 | 100% | 0 | 0 |  |
| **订单查询** | 3 | 100% | 0 | 0 |  |
| **用户注册** | 2 | 100% | 0 | 0 |  |
| **用户登陆** | 2 | 100% | 0 | 0 |  |
| **用户修改个人信息** | 1 | 100% | 0 | 0 |  |
| **用户查看个人信息** | 1 | 100% | 0 | 0 |  |
| **新增订单** | 1 | 100% | 0 | 0 |  |
| **查看订单** | 2 | 100% | 0 | 0 |  |
| **修改订单状态** | 2 | 100% | 0 | 0 |  |
| **取消订单** | 2 | 100% | 0 | 0 |  |
| **总计** | 42 | 95% | 2 | 5% |  |
| **非功**  **能项** | **暂无** | | | | | |
|  | **非功能项小计** | *0* | *0* | *0* | *0* | *0* |
|  | **总计** | *42* | *95%* | *2* | *5%* |  |

## 缺陷严重程度

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** |  | 2 |  |  |  |
| **占缺陷百分比** |  | 100% |  |  |  |

# 缺陷清单

## 功能性缺陷

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | 1 | 严重 | 商品信息管理 | 商品信息添加 | 时间原因没有实现 |  |
| 2 | 2 | 严重 | 商品信息管理 | 商品信息修改 | 时间原因没有实现 |  |
| 3 |  |  |  |  |  |  |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

**表4 非功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **测试类型** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 |  |  | 性能 |  |  |  |
| 2 |  |  | 性能 |  |  |  |
| … |  |  | … |  |  |  |
| n |  |  | 可靠性 |  |  |  |

# 测试结论与建议

本次第一次迭代主要是以实现功能为主，系统整体还不完善，测试也以单元测试为主.

使用的是以Postman的手动测试为主，测试不够便捷与快捷，后面要尝试实现自动化测试。测试用例数量不够，无法保证潜在的错误完全排除。