<饿了么>

测试报告

版本 <3.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <27/08/2018> | <1.0> | <添加单元测试 > | <王梦瑶> |
| <04/09/2018> | <2.0> | <添加测试用例> | <李凌荟&李珊> |
| <06/09/2018> | <3.0> | <添加压力测试> | <全体> |
| <07/09/2018> | <4.0> | <汇总完善> | <张伯宁> |

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

本测试报告为软件学院2016级学生于2018年暑假进行的软件项目《吃了吗》的测试报告，目的在于总结测试阶段的测试，以及分析测试结果，描述系统是否符合需求

## 范围

预期参考人员包括用户、测试人员、开发人员、项目管理者、其他质量管理人员和需要阅读本报告的产品经理

## 参考资料

《吃了吗项目软件需求规约》、《吃了吗项目测试用例》

## 概述

本项目包括系统功能测试以及系统压力测试

# 测试概要

测试时间：每次迭代结束之前进行测试

具体项目测试见修订历史记录

测试地点：软件学院

测试人员：项目小组成员

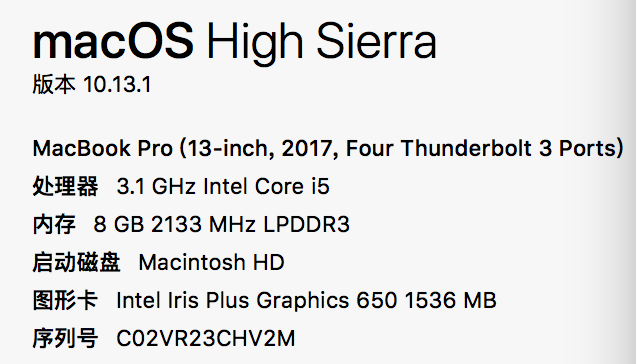
测试方法：使用Jmeter来监测前端和后端服务端口的响应正确性和响应性能

测试内容：包括系统正确性测试以及系统压力测试和系统单元测试

# 测试环境

硬件环境：i7CPU，960tx显卡，intel 512G SSD

软件环境：win10和 Jmeter（64bit）



# 测试结果及分析

## 需求覆盖率及缺陷分布

*[需求覆盖率是指经过测试的需求/功能和软件需求规约中所有需求/功能的比值，通常情况下要达到100%的目标。需给出功能测试及非功能测试的需求覆盖情况。对于功能测试，需给出各个主要功能模块的需求覆盖率。如果需求覆盖率未达到100%可在备注项中说明原因及未测试内容。缺陷率是指本项缺陷占总缺陷数的百分比。请尽量使用图、表进行描述。]*

**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** | **用户登录与注册** | 12 | 100% | 0 | 0% | 无 |
| **新用户引导** | 7 | 80% | 0 | 0% | 无 |
| **用户浏览窗口** | 15 | 100% | 0 | 0% | 无 |
| **用户浏览历史记录** | 4 | 100% | 0 | 0% | 无 |
| **功能项小计** | 38 | ------ | 0 | 0% | 无 |
| **非功**  **能项** | **性能** | 9 | 75% | 0 | 0% | 无 |
| **可靠性** | 21 | 100% | 0 | 0% | 无 |
| **易用性** | 2 | 100% | 0 | 0% | 无 |
|  | **非功能项小计** | 32 | *--------* | 0 | 0% | 无 |
|  | **总计** | 70 | *--------* | 0 | 0% | 无 |

## 缺陷严重程度

*[描述缺陷按严重程度分布情况，包括不同严重程度缺陷个数及占缺陷总数百分比等。尽量使用图、表表示。]*

**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** | 无 | 无 | 无 | 无 | 无 |
| **占缺陷百分比** | 无 | 无 | 无 | 无 | 无 |

# 缺陷清单

## 功能性缺陷

*[简要阐明功能缺陷汇总情况，并给出具体功能缺陷列表。]*

**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 | ---- | -- | --- | ----- | ---------- | ---- |
| 2 | ---- | -- | --- | ----- | ---------- | ---- |
| 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

*[简要阐明非功能缺陷汇总情况，并给出具体非功能缺陷列表。(根据所测系统的实际情况，可选择性地对系统进行必要的非功能性测试，包括系统的性能、可靠性、易用性、安全性、兼容性等。)请参考功能缺陷列表给出相关缺陷清单。]*

**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 | 无 |  |  |
| 2 |  |  | 性能 | 无 |  |  |
| n |  |  | 可靠性 | 无 |  |  |

# 测试结论与建议

[描述测试过程中所遇到的问题，根据测试中所遇到的问题、对测试结果所进行的分析，给出对本次测试的结论、及可取的建议措施等。]

首先感谢小组成员的同心协力以及耐心调试！

《吃了吗》项目的测试工作到今天算是全部结束了，除了后期维护必要的一些回归测试和用户使用手册的撰写外，整个测试阶段告一段落。

从6月底进入项目，在沈备军老师的帮助下开始学着写项目测试文档，到根据文档的单元测试及每日功能测试，再到整个项目进行迭代后对测试文档的重新架构及整体回归测试，直至最后的统一交付测试，小组项目提交总BUG数为244个。

在这244个BUG的提交和修正过程中，在测试文档的写作及修订中，我们对整个项目的逻辑及架构逐步清晰，对项目之间所需的复杂交互的认识也越发深入，对项目功能逻辑上的测试如何进行也更加明晰。

下面简单谈谈对项目的结论以及对未来项目的建议：

测试实为软件开发中即为重要的一环，以前对测试总是不置可否，经过这次的训练发现测试真的会驱动软件的整体开发，从单元测试到类测试到系统测试都会对整个项目产生不可或缺的影响。

对未来的建议：

首先团队项目在开发的时候一定要牢记各个软件的版本的统一一致：因为版本不一样导致出现了一些匪夷所思的bug令人无从下手。

其次在每次迭代交付之前一定要进行完整的测试，来保证整个项目的正确性得到充分保证。

很难忘的一次开发体验，很愉快的小组测试~