<LETMEKNOW通知app>

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <9/9/0217> | <1.0> | <编写测试报告> | <景致> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

保证工程各部分以及整体整合的正确性。确保工程的各项功能都能够运转。考察工程在固定硬件设施和网络条件下的性能。

## 范围

测试涵盖了整个工程（包括安卓端，windows管理员端，服务器端）的功能性需求和非功能性需求。

## 参考资料

无

## 概述

本报告为各个测试的总结，提供测试用例的数据统计，详细的测试用例见《测试用例》。

# 测试概要

测试时间：9/8-9/9

测试人员：杨健邦，陈世昊，陈天睿，景 致

测试方法：单元测试、集成测试、黑盒测试、白盒测试

# 测试环境

阿里云服务器

配置： CPU:1核 内存：2GB I/O传输：1Mbps

Android测试机

nexus 6p （android 8.0）

# 测试结果及分析

## 需求覆盖率及缺陷分布

**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** | **安卓端功能** | 21 | 100% | 1 | 4.8% |  |
| **管理员端功能** | 13 | 100% | 0 | 0% |  |
| **功能项小计** | 34 | 100% | 1 | 2.9% |  |
| **非功**  **能项** | **性能** | 10 | 100% | 1 | 10% |  |
| **易用性** | 5 | 100% | 1 | 20% |  |
|  | **非功能项小计** | *15* | *100%* | *2* | *13.3%* |  |
|  | **总计** | *49* | *100%* | *3* | *6.1%* |  |

## 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** | 0 | 0 | 0 | 3 | 3 |
| **占缺陷百分比** | 0% | 0% | 0% | 100% | 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 | Bug001 | 细微 | 推送 | 推送不及时 | 在app进程被完全杀死之后可能无法推送消息 | T1 |

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 | Bug006 | 细微 | 易用性 | 管理员界面边框 | 管理员界面白色区域缺少边框，后面如果是白色背景会与之融为一体，难以分辨 | Z1 |
| 2 | Bug007 | 细微 | 性能 | 回执读取速度 | 由于回执读取要向服务器请求数据，所以在网络不佳的时候速度慢 | Z2 |

# 测试结论与建议

本测试结果显示该工程满足了大部分的功能性需求和非功能性需求。在UI美化方面和细节方面仍有缺陷，服务器在高并发下的性能有待于进一步测试。