器乐练习辅助app——小海螺测试报告

版本 1.0

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 24/7/2019 | 1.0 | 增加后端的功能性测试 | 张怡天 |
| 26/7/2019 | 1.0 | 增加app的功能性测试 | 刘庆星 |
|  |  |  |  |
|  |  |  |  |

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

本报告针对本项目第二次迭代完成的相关内容，对于在其基础上进行的相关测试，进行相关的分析。

## 范围

本次报告的范围是项目的第二次迭代。

## 定义、首字母缩写词和缩略语

无

## 参考资料

无

## 概述

本次测试主要针对前后端的相关功能，检验其是否正常，非法输入能否通过以及与数据库的连接情况。

# 测试概要

7月22日，我们对开发出的用户信息访问和乐谱获取功能进行了功能测试，主要测试人员是张怡天；

7月22日，我们对开发出的乐谱搜索、调音和错误标记功能进行了功能测试，主要测试人员是刘庆星；

# 测试环境

后端测试在Windows系统下运行，测试使用的环境为Android Studio及其附带的模拟器。

前端测试在Mac OS系统下运行，测试使用环境为Xcode及ios12.2 Simulator

# 测试结果及分析

本次共对5个功能点进行了测试。使用了12个测试用例，没有发现缺陷。

## 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** | **请求用户数据** | 3 | 100% | 0 | 0% | 无 |
| **请求乐谱信息** | 3 | 100% | 0 | 0% | 无 |
| **乐谱搜索** | 2 | / | 0 |  |  |
| **调音** | 2 | / | 0 |  |  |
| **错误标记** | 2 | / | 0 |  |  |
| **……** |  |  |  |  |  |
| **功能项小计** | 12 | 100% | 0 | 0% | 无 |
| **非功**  **能项** | **性能** |  |  |  |  |  |
| **可靠性** |  |  |  |  |  |
| **…** |  |  |  |  |  |
|  | **非功能项小计** | 0 | -- | -- | -- | 无 |
|  | **总计** | 12 | 100% | 0 | 0% | 无 |

## 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 | 0 | 0 |
| **占缺陷百分比** | 0% | 0% | 0% | 0% | 0% |

# 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 |  |  |  |  |  |  |
| 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

**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

本次测试测试了前后端基本的功能运行，基本测试到了整个项目所需的功能，但目前来说还是比较基础，下一次迭代将用更多用例进行性能测试以确保我们的系统稳定运行