<时间管理app>

系统测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <5/1/2021> | <1.0> | <创建> | <沈佳威> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

系统测试报告

# 简介

## 目的

本测试报告为时间管理app软件项目的系统测试报告，目的在于对系统开发和实施后的结果进行测试以及测试结果分析，发现系统中存在的问题，描述系统是否符合项目需求规约中规定的功能和非功能需求。

## 范围

项目名称：时间管理app一心（ISSUES）

预期参考人员：用户、测试人员、开发人员、项目管理者、其他质量管理人员

## 定义、首字母缩写词和缩略语

系统测试：按照需求规约对系统整体功能进行的测试

功能测试：测试软件各个功能模块是否正确，逻辑是否正确

系统测试分析：对测试的结果进行分析，形成报告、便于交流和保存

闹钟：能够在预定时间发出铃声提醒用户的APP，在本项目中需要完成一些小任务才能关闭铃声。

任务：本文中指用来关闭闹钟铃声的小任务，包括小游戏、摇晃手机、唱歌、数学题等。

学习模式：所有手机通知将被屏蔽，除部分应用外的应用将被锁定的APP模式，但仍可接听电话，拍照和使用紧急呼叫。

学习青蛙：类似于Forest的种树，学习模式的卡通化，当用户使用学习模式时学习青蛙会开始学习。

激励机制：完成任务或学习后的奖励，量化为学习青蛙的学习成果，可以给用户带来成就感和激励感。

等级：按某一标准区分的高下差别，在本项目中，根据学习时间的长短学习青蛙划分为不同学历。

用户信息：指用户名称、用户性别、用户联系方式等一些关于用户的基本资料。

好友：通过该项目APP可以互设任务闹钟。

## 参考资料

1）《时间管理app软件需求规约》

2）《时间管理app项目测试用例》

## 概述

本文档将从测试环境、测试结果及分析、缺陷清单和测试结论及建议这四个方面完成系统测试。

# 测试概要

时间：2021年1月1日——2021年1月7日  
地点：图书馆小组自习室  
人员：所有小组成员

测试方法：用Jmeter进行测试

测试内容：本测试按照软件需求规约分别测试功能测试和系统测试。功能测试覆盖各个子系统的功能模块，本测试针对在现有产品功能模块以及实施结果分别进行测试，测试整个系统是否达到需求规约中要求实现的功能以及测试系统的易用性、用户界面的友好性。系统测试包括系统的易用性、可靠性、安全性、可维护性等，整个系统集成后提供服务的能力，还包括系统服务性能测试。

# 测试环境

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 用途 | 硬件环境 | 软件环境 |
| 1 | 服务器 | 内存：2GB  物理处理器：Intel Xeon Family  时钟速度：2.5Ghz | 操作系统：Amazon Linux AMI 数据库软件：Mysql、PostgreSQL等 |
| 2 | App端 | 内存：4GB | 操作系统：Android |
| 3 | Web端 | 内存：8GB  磁盘：512GB | 操作系统：Windows 浏览器：  Chrome |
| 4 | 网络要求 | 2Mbps带宽 | |

# 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** | **登录注册服务** | 3 | 100% | 0 | 0% |  |
| **闹钟编辑** | 4 | 100% | 0 | 0% |  |
| **闹钟任务** | 7 | 100% | 1 | 2% | 照片识别需要较长时间 |
| **闹钟唤醒** | 2 | 100% | 0 | 0% |  |
| **学习服务** | 2 | 80% | 0 | 0% | 分享功能尚未实现 |
| **好友服务** | 3 | 100% | 0 | 0% |  |
| **个人中心** | 4 | 100% | 0 | 0% |  |
| **报表服务** | 1 | 80% | 0 | 0% | 分享功能尚未实现 |
| **功能项小计** | 26 | 84% | 1 | 2% |  |
| **非功**  **能项** | **性能** | 7 | 100% | 1 | 2% | 登录响应时间较长 |
| **可靠性** | 3 | 100% | 0 | 0% |  |
| **安全性** | 2 | 100% | 0 | 0% |  |
| **兼容性** | 2 | 100% | 0 | 0% |  |
| **可支持性** | 2 | 100% | 0 | 0% |  |
| **易用性** | 2 | 100% | 0 | 0% |  |
|  | **非功能项小计** | *18* | *100%* | *1* | *2%* |  |
|  | **总计** | *44* | *91%* | *2* | *2%* |  |

## 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 | 1 | 1 |
| **占缺陷百分比** | 0 | 0 | 0 | 100% | 1 |

# 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 | 闹钟任务 | 指定物品拍照识别时延迟 | 指定物品拍照识别时有一定延迟、大约在10-20s | ISSUES-testcase-007 |

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 | Bug2 | 细微 | 性能 | 登录响应时间较长 | /login 接口的平均反应时间超过了 2s | ISSUES-testcase-023 |

# 测试结论与建议

1. 测试过程中的问题：
2. 对微服务架构的单元测试较为陌生
3. 对微服务架构的部署较为陌生，而基于此的系统测试做的比较粗糙
4. 对测试结果的分析：

功能需求完成率为84%，其中未完成的是分享服务（微信申请未通过）；非功能需求基本完成，有一些细微的缺陷，如登录响应时间相对较长。  
3）测试的结论：

经过对测试结果的分析，项目的设计和实施基本达到了需求规约中要求的功能需求和非功能需求。  
4）可取的建议措施：

1.要加强对性能压力测试的重视程度

2.对于快速响应，不能一味依赖接口，而希望自己就能快速响应

3.对于软件开发周期的时间把控需要更加精细，以助于完成所有功能需求