智能有声书

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2019/07/08 | 1.1 | 第一次迭代测试 | 徐轶喆 |
| 2019/07/14 | 1.2 | 第一次迭代测试 | 徐轶喆 |
|  |  |  |  |
|  |  |  |  |

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

软件测试是为了发现错误而执行程序的过程。测试并不仅仅是为了要找出错误。通过分析错误产生的原因和错误的分布特征。可以帮助项目管理者发现当前所采用的软件过程的缺陷，以便改进。同时，通过分析也能帮助我们设计出有针对性的检测方法，改善测试的有效性。

撰写本篇报告的目的，是为了能够在软件测试的过程中使用系统性的策略和方法，对软件分不同方面进行尽可能完备的测试，同时对测试的用例编写，测试的过程，测试的结果以及测试的分析有一个统一的分析，更易于对软件现状作出正确的评估。

## 范围

本文档的范围限于对智能有声书项目系统版本1进行的功能测试、性能测试、界面测试、部署测试，以及各项测试的统计结果。本文档将用于对系统版本1发布前，进行错误的纠察及修正，及时完善发布版本。

## 定义、首字母缩写词和缩略语

历史记录:

智能有声书系统中用户曾经阅读过的有声书记录。

隐私设置:

用户个人信息设置是否公开。

## 参考资料

1. 《软件工程原理》.作者:沈备军、陈昊鹏、陈雨亭.高等教育出版社.2014.12版
2. 《软件测试的目的、原则及流程》.作者:Shine\_安.2018.04.04

<https://blog.csdn.net/u010885548/article/details/79817141>

## 概述

本文档将依次说明测试概要、测试环境、测试结果及分析、缺陷分析和结论总结。每个部分都将对其相应的内容进行详细列述。

# 测试概要

时间: 20190709

地点：软件学院实验室

人员：徐轶喆

测试方法：详见测试用例

测试内容：项目的功能测试、性能测试、界面测试、部署测试

# 测试环境

前端框架：MUI

前端开发软件：HbuilderX，夜神多开手机模拟器

后端框架：SSM

后端开发软件：IDEA

服务器：腾讯云

数据库管理：Mysql，MongoDB

# 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** | **用户注册** | 13 | 100% | 2 | 15.4% |  |
| **用户登录** | 11 | 100% | 2 | 18.2% |  |
| **历史记录** | 3 | 100% | 0 | 0 | 未测用例数:  2 |
| **用户收藏** | 3 | 100% | 0 | 0 |  |
| **用户信息** | 3 | 100% | 0 | 0 |  |
| **隐私设置** | 2 | 100% | 0 | 0 |  |
| **有声书制作** | 2 | 100% | 0 | 0 |  |
| **有声书操作** | 4 | 100% | 0 | 0 |  |
| **功能项小计** | 41 | 100% | 4 | 10.2% | 未测用例数:  2 |
| **非功**  **能项** | **部署** | 1 | 100% | 0 | 0 |  |
| **界面** | 6 | 100% | 1 | 16.7% |  |
|  | **非功能项小计** | 7 | 100% | 1 | 14.3% |  |
|  | **总计** | 48 | 100% | 5 | 10.9% | 未测用例数:  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 | 5 | 5 |
| **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 | Gbug001 | 细微 | 用户登录 | 用户名是否存在提示 | 输入不存在的用户名，应提示不存，测试结果为:系统提示用户账号密码不匹配 | AudioBook-testcase-022 |
| 2 | Gbug002 | 细微 | 用户登录 | 用户名长度提示 | 输入超出16位的用户名，应提示用户名过长，测试结果为:系统提示账号密码不匹配 | AudioBook-testcase-017 |
| 3 | Gbug003 | 细微 | 用户注册 | 用户名格式检测提示 | 输入错误格式的用户名，应提示用户名格式错误，测试结果为:系统提示用户名长度应在1~16位 | AudioBook-testcase-012 |
| 4 | Gbug004 | 细微 | 用户注册 | 用户名密码格式提示 | 输入错误格式的密码，应提示密码格式错误，测试结果为:系统提示密码长度应在6~16位 | AudioBook-testcase-013 |

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 | Fbug001 | 细微 | 界面 | 系统响应时间 | 系统响应时间测试目标为不超过2秒，实际最长响应时间为3~4秒 | AudioBook-testcase-045 |

# 测试结论与建议

在第一次迭代的测试中，我们对开发的系统版本1的高优先级5大功能项都进行了测试。测试结果表明，第一次迭代发布的版本基本可以满足需求规约上的详细需求。

在测试中，我们发现，尽管第一次迭代的系统版本软件，可以满足优先需求，但软件在界面美化、功能和性能上都有缺陷。由于生命周期在MUI架构下App应用开发中并不适用，导致我们前端页面对数据的请求和处理都遇到了很大的困难；页面跳转也是开发中的一个难题，MUI并没有封装好的现有路由组件，现阶段我们只能通过新开页面的方式来实现跳转。

由于这些产生的问题，我们在测试时也发现了一些问题。除了测试用例中发现的缺陷之外，我们在使用系统版本1的过程中也发现了一些问题。例如，跳转页面后原页面的数据的去留不一，一些操作是否成功没有给出具体的提示，界面不够美观，布局设计出现偏差等等。

对于这些问题，我们的建议是对问题进行统一总结后进行罗列，在测试之后的一周之内一一解决。我们小组的前端开发组员也在对软件界面的性能和美化进行修正，预期在一周之内，可以就第一次迭代测试结果，完善系统版本1 。