Hogwarts

系统测试报告

版本 <3.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <01/01/2021> | <1.0> | 初始版本 | 熊添晨 |
| <05/01/2021> | <2.0> | 新增部分测试 | 郭黛筠 |
| <08/01/2021> | <3.0> | 更新部分测试结果 | 熊添晨 |
|  |  |  |  |

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

*为霍格沃茨导游APP提供系统测试报告，为本项目修改提供方向，并为课程提交材料。*

## 范围

*霍格沃茨导游APP项目*

## 定义、首字母缩写词和缩略语

*无*

## 参考资料

*《软件工程原理教材》*

## 概述

*本测试报告包含功能性与非功能性测试以及缺陷清单。*

# 测试概要

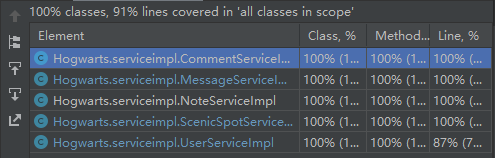
测试使用Junit，Jmeter，Prometheus，Granafana以及flutter自带的测试工具进行相关测试，测试地点为小组学习室，成员为全体成员参与，时间为12/10-1/7。

# 测试环境

处理器：Intel®Core™i5-8300H CPU @ 2.30GHz

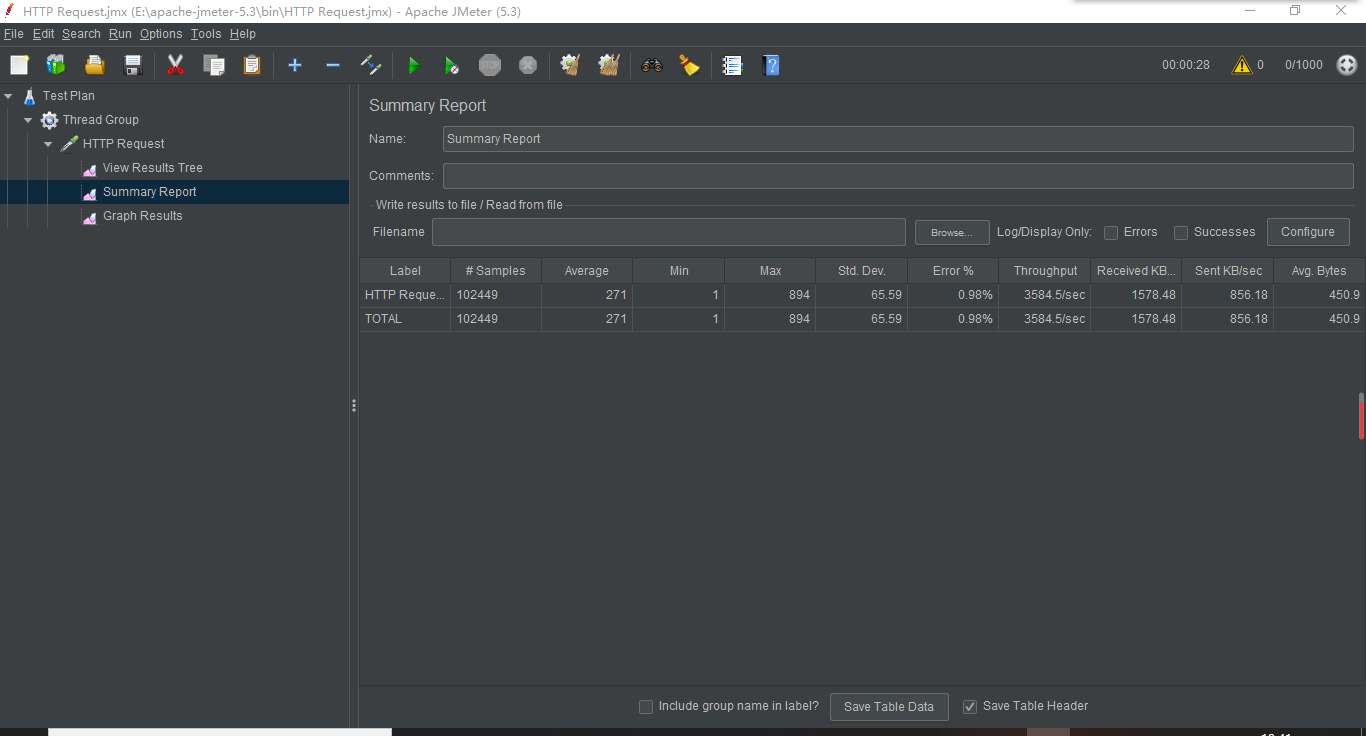
# 测试结果及分析

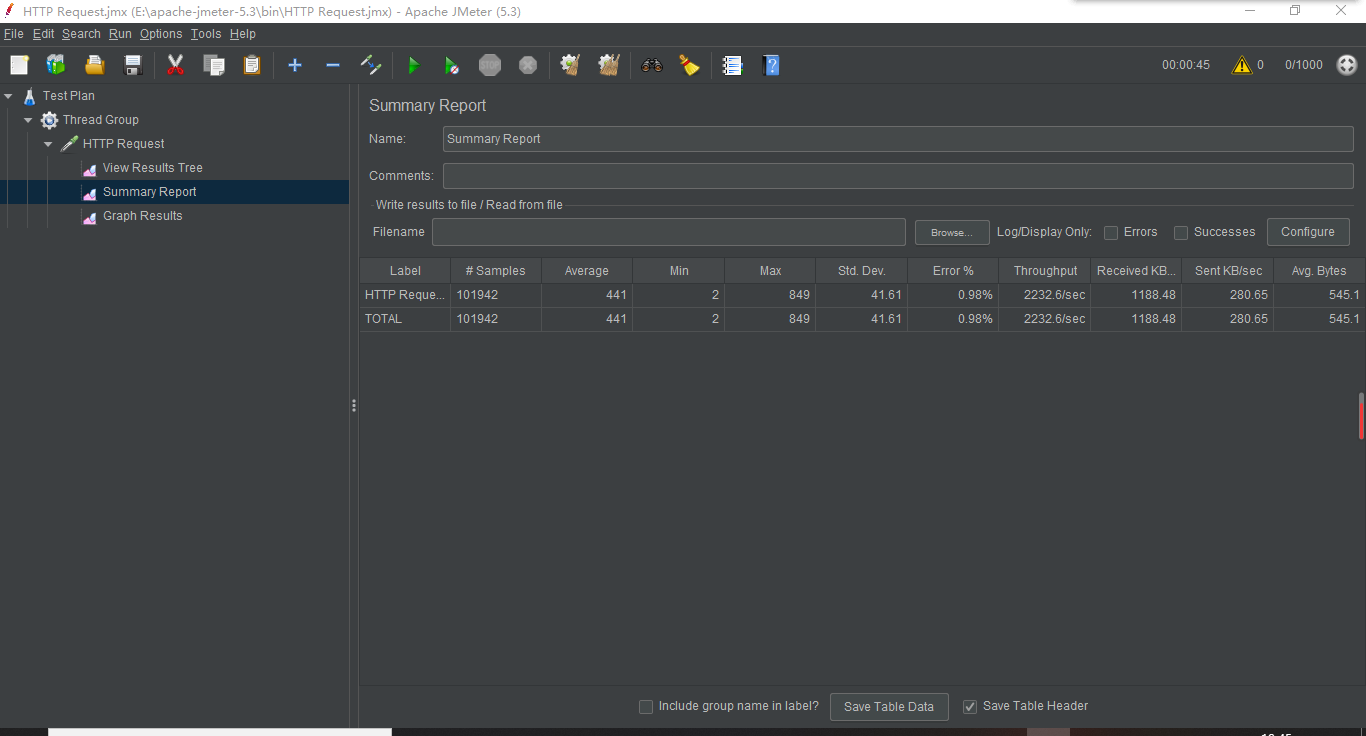
单元测试:



单元测试类覆盖率，方法覆盖率100%，代码覆盖率只有少数错误处理函数未被测试。

性能测试：1000并发条件下，最大响应时间均小于1s，第一张图为组队共享性能测试，第二张图为普通功能获取用户信息测试。





## 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** | **导游** | 5 | 100% | 0 | 0 | 已解决 |
| **游记生成** | 3 | 100% | 0 | 0 | 已解决 |
| **景点评价** | 1 | 100% | 0 | 0 | 无 |
| **个人信息修改** | 2 | 100% | 0 | 0 | 无 |
| **功能项小计** | 11 | 100% | 0 | 0 | 无 |
| **非功**  **能项** | **性能** | 12 | 100% | 0 | 0 | 1000并发下已控制为最大1s |
| **可靠性** | 2 | 100% | 0 | 0 | 无 |
| **界面测试** | 2 | 100% | 0 | 0 | 无 |
| **安全性测试** | 1 | 100% | 0 | 0 | 无 |
| **可靠性测试** | 1 | 100% | 0 | 0 | 无 |
| **部署测试** | 1 | 100% | 0 | 0 | 无 |
| **兼容性测试** | 1 | 100% | 0 | 0 | 无 |
| **…** |  |  |  |  |  |
|  | **非功能项小计** | *20* | *100%* | *0* | *0* | *无* |
|  | **总计** | *32* | *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

*无功能性缺陷（在前几个版本中存在一些缺陷，但最终版本均已解决）*

**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** |

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

*无（在前几个版本中存在一些非功能性缺陷，但最终版本均已成功解决。）*

# 测试结论与建议

由于项目开始很早，虽然在前几个版本内存在一些功能或非功能缺陷，但是由于项目成员不懈努力与修改，所有已发现的功能缺陷以及非功能缺陷均已解决，故在最终版本系统测试报告中，所有测试均已通过且所有已发现的缺陷均已排除。