<项目名称>

系统测试报告

版本 <1.0>

[注：其中包括用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File >Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit> Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <日/月/年> | <x.x> | <详细信息> | <姓名> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

针对项目的最终部署做最终测试，包括功能需求和非功能需求。

## 范围

测试报告相关范围：对后端代码的service层做测试，并对后端api进行压力测试

## 定义、首字母缩写词和缩略语

无

## 参考资料

《系统测试用例》，《软件工程原理》，《qaq问答平台压测报告.csv》

《后端单元测试report》

## 概述：

本文档针对最终系统测试，对后端整体业务逻辑，以及性能，做最终测试

包括测试概要、测试环境、测试结果与分析，测试结论与建议

其中，测试概要说明了测试的时间、地点、人员。测试方法、测试内容等

测试环境简要描述了测试所使用的软硬件环境及配置

测试结果及分析只出了测试的结果，并给出了主要测试结果的图形表格

缺陷清单简要阐明了功能缺陷汇总情况，并给出了具体的功能缺陷列表

# 测试概要

测试时间：2021年1月8日星期五

地点：上海交通大学学生宿舍

测试内容与方法：利用junit对后端代码进行单元测试，并利用jmeter对重要端口做性能测试

功能测试：根据需求，针对每个用例的基本流与备选流设计正向/反向测试用例，测试正常操作能否正常执行通过，非法操作能否触发系统反应。

性能测试：对后端的主要接口设计性能测试用例，使用DBforge插入万级数据,Jmeter模拟高并发访问，实现98.3%的主要接口在1000并发下平均访问时间小于3秒

部署测试：即安装测试，测试后端部署各种服务包时，系统前端正常访问能否被支持

# 测试环境

硬件：谷歌云服务器

软件：jmeter，junit5

# 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** | 3 | 100% | 0 | 0 |  |
| **用户登录** | 9 | 66% | 3 | 33% | 用户封禁，激活，不存在仍可正常登录 |
| **用户退出** | 1 | 0 | 1 | 100% | 用户退出之后界面逻辑处理不当 |
| **用户提问** | 2 | 100% | 0 | 0 |  |
| **用户回答** | 2 | 100% | 0 | 0 |  |
| **用户评论** | 3 | 100% | 0 | 0 |  |
| **用户通知** | 1 | 0 | 1 | 100% | 页面显示微小bug |
| **用户问题管理** | 4 | 75% | 1 | 25% | 删除之后仍会显示 |
| **用户信息管理** | 3 | 66% | 1 | 33% | 用户头像更改存在bug |
| **关注** | 4 | 100% | 0 | 0 | 对用户和问题的关注与取消 |
| **功能项小计** | 32 | 78.13% | 7 | 21.87% |  |
| **非功**  **能项** | **性能** | 23 | 100% | 0 | 0 |  |
| **界面** | 3 | 100% | 0 | 0 |  |
| **单元测试** | 8 | 75% | 2 | 25% |  |
| **兼容性测试** | 1 | 100% | 0 | 0 |  |
| **部署** | 3 | 100% | 0 | 0 |  |
|  | **非功能项小计** | 38 | 95% | 2 | 5% |  |
|  | **总计** | 70 | 87% | 9 | 13% |  |

## 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** |  | 1 | 4 | 3 | 7 |
| **占缺陷百分比** |  | 14.3% | 57.1% | 42.6% |  |

# 缺陷清单

## 功能性缺陷

*[简要阐明功能缺陷汇总情况，并给出具体功能缺陷列表。]*

**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 | Bug0 | 一般 | 用户登录 | 未激活用户 | 未激活用户也能正常登录 | 12 |
| 2 | Bug1 | 细微 | 用户登录 | 用户不存在 | 用户不存在时不能正常报错，但确实未登录 | 9 |
| 3 | Bug2 | 一般 | 用户退出 | 退出异常 | 用户退出之后页面不能显示未登录状态 | 14 |
| 4 | Bug3 | 一般 | 问题管理 | 问题删除 | 问题删除后仍会显示 | 33 |
| 5 | Bug4 | 严重 | 用户编辑 | 编辑逻辑 | 编辑组件有严重逻辑，导致上传头像失败，以及用户信息丢失 | 38 |
| 6 | Bug5 | 一般 | 用户登录 | 用户被封禁 | 被封禁用户也能正常登录 | 11 |
| 7 | Bug6 | 细微 | 用户通知 | 通知页面缺陷 | 通知页面有些许缺陷 | 31 |

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 | 001 | 细微 | 单元测试 | 单元测试覆盖率缺陷 | 对userService层的代码覆盖率为93%，未达到95% | 045 |
| 2 | 002 | 细微 | 单元测试 | 单元测试覆盖率缺陷 | 对用户关注问题的service代码覆盖率为92%，未达到95% | 045 |

# 测试结论与建议

测试过程中遇到的问题：

使用jmeter进行测试时部分端口显示500

一些数据获取的接口响应时间过长，延迟较高

测试结果进行的分析：

本次测试结果暴露了平台的功能性需求的完成度较低，许多高优先级的需求并没有很好地做到位，

同时压力测试的数据量不足，可能导致测试结果的不准确。

单元测试在service层的代码覆盖率高，算是比较可靠，但是面对前端的许多非法请求并没有很好处理，由于缺乏端到端测试，导致了我们答辩的不足

本次测试的结论：

本次测试暴露了系统的诸多不足，需要修复的bug较多，但严重bug不多，不会影响用户的正常使用与体验

可取的建议措施：

本次项目给了我们很大的教训。在小组项目中，合作是关键，分工混乱不明确的情况下，很容易导致返工和失败，在未来的项目开发中，要及时进行功能和非功能测试，增强前后端的工作沟通，做好分工，严格按照迭代计划进行执行，才可以避免项目的失败