<HONEYCOMB>

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018.9.8 | 0.1 | 添加部分基础用例的测试报告 | 曹金坤，谢添翼 |
| 2018.9.8 | 0.2 | 添加了部分非功能性用例的测试报告 | 曹金坤，谢添翼 |
| 2018.9.8 | 0.3 | 添加了性能测试部分的报告 | 朱锦昊 |
| 2018.9.8 | 1.0 | 添加了后端自动化测试部分的报告 | 马轲 |

[1. 简介 4](#_Toc524289433)

[1.1 目的 4](#_Toc524289434)

[1.2 范围 4](#_Toc524289435)

[1.3 定义、首字母缩写词和缩略语 4](#_Toc524289436)

[1.4 参考资料 4](#_Toc524289437)

[2. 测试概要 4](#_Toc524289438)

[3. 测试环境 4](#_Toc524289439)

[4. 测试结果及分析 4](#_Toc524289440)

[4.1 需求覆盖率及缺陷分布 4](#_Toc524289441)

[4.2 缺陷严重程度 5](#_Toc524289442)

[5. 缺陷清单 5](#_Toc524289443)

[5.1 功能性缺陷 5](#_Toc524289444)

[5.2 非功能性缺陷 6](#_Toc524289445)

[6. 测试结论与建议 6](#_Toc524289446)

测试报告

# 简介

## 目的

说明针对HoneyComb项目的测试用例和其他测试需求形成的结果。

## 范围

本文档定义的内容，涉及到的范围包括Honeycomb项目包含的所有源代码、文档和媒体文件。

## 定义、首字母缩写词和缩略语

SSH: Spring, Struts, Hibernate形成的后端框架

## 参考资料

《HoneyComb项目立项建议书》*,2018.6.20,Honeycomb小组*

《HoneyComb需求规约》*,2018.6.20,Honeycomb小组*

《HoneyComb测试用例》,*2018.9.6,HoneyComb小组*

# 测试概要

测试时间：2018.9.8 – 2018.9.9

测试人员：曹金坤、谢添翼、马轲、朱锦昊

测试方法：

1. 前端测试：基于使用用例的基础上设置测试用例进行手动测试
2. 后端测试：基于自动化测试程序，对主要的后端功能函数进行测试
3. 性能测试: 基于loadrunner进行模拟的压力和并发测试；
4. 部署测试：手动进行

# 测试环境

测试环境：

1. Macbook pro 2015 early，8G内存
2. Windows10, i7-7代标压，8G内存
3. Ubuntu 14.04, 2G内存

# 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** | 4 | 100% | 0 | 0 |  |
| **修改简介** | 1 | 100% | 0 | 0 |  |
| **创建板块** | 3 | 100% | 0 | 0 |  |
| **跟帖** | 2 | 100% | 0 | 0 |  |
| **评论帖子** | 2 | 100% | 0 | 0 |  |
| **用户登录** | 3 | 100% | 0 | 0 |  |
| **系统部署测试** | 1 | 100% | 0 | 0 |  |
| **主页搜索** | 2 | 100% | 0 | 0 |  |
| **条件搜索** | 3 | 100% | 0 | 0 |  |
| **浏览论文网络图** | 2 | 100% | 0 | 0 |  |
| **网络图中浏览论文信息和评论** | 1 | 100% | 0 | 0 |  |
| **网络图中选取制定层级节点** | 1 | 100% | 0 | 0 |  |
| **网络图中选区论文节点** | 1 | 100% | 0 | 0 |  |
| **扩展网络图** | 1 | 100% | 0 | 0 |  |
| **个人网络图** | 9 | 100% | 0 | 0 |  |
| **评论论文** | 2 | 100% | 0 | 0 |  |
| **论文收藏** | 4 | 100% | 0 | 0 |  |
| **用户社区记录** | 1 | 100% | 0 | 0 |  |
| **查看浏览历史记录** | 1 | 100% | 0 | 0 |  |
| **功能项小计** | 50 | 100% | 0 | 0 |  |
| **非功**  **能项** | **性能** | 1 | 100% | 0 | 0 | 基于loadrunner进行的性能测试，包含多项用户操作 |
| **部署测试** | 1 | 50% | 1 | 1 | 在服务器上的部署在数据库访问效率和网络上体现了缺陷，并且没有实现小程序平台部署 |
| **界面测试** | 7 | 100% | 3 | 42.9% | 部分前端内容没有实现统一的css库适配 |
|  | **非功能项小计** | 9 | 90% | 4 | 44.4% |  |
|  | **总计** |  |  |  |  |  |

## 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 | 3 | 1 | 4 |
| **占缺陷百分比** | 0 | 0 | 75% | 25% | 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** |

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 | Slow! | 细微 | 部署测试 | 服务器端部署过高延迟 | 在绑定了公网域名的服务器上部署HoneyComb项目时，发现了较大的延迟现象，导致部分操作可能会因为超时而失败 | 16 |
| 2 | Community\_non-bs | 一般 | 界面测试 | 社区模块风格不统一 | 在社区模块的风格设计中，引入了不满足预先确定的bootstrap／md-bootstrap风格库的设计 | 48 |
| 3 | Panel\_non-bs | 一般 | 界面测试 | Panel详情模块风格不统一 | 在Panel详情展示模块的风格设计中，引入了不满足预先确定的bootstrap／md-bootstrap风格库的设计 | 51 |
| 4 | Editor\_non-bs | 一般 | 界面测试 | 复用控件风格不统一 | 在编辑器模块的风格设计中，引入了不满足预先确定的bootstrap／md-bootstrap风格库的设计 | 53 |

# 测试结论与建议

经过前述的测试工作，我们得出了以下的主要测试结论：

1. 工程的功能性需求得到了完整的满足，基本符合立项时的计划，站内私信功能之后因为需求变更没有实现，相关说明已经在项目的总结报告中得到了体现；
2. 少数的功能性测试中出现了一些漏洞，部分测试用例的设计也不够完善，测试过程可以进一步完善；
3. 对于总共40W条的论文数据，在搜索和论文可视化时，工程体现出了可以察觉的延迟，有进一步的优化空间，但是基本满足了功能性的需求；
4. 对于错误的输入、前置条件等负测试用例，测试过程体现了工程基本的健壮性，但是反馈的错误信息有进一步优化的余地，比如对于已注册用户输错密码和登录的用户未注册，前端给出了同样的提示，有时会让用户迷惑；
5. 工程对于大量并发的支持性较弱，在loadrunner的压力测试中体现出来了较大的不足，可以通过进一步的代码优化乃至于深层次的重构和架构设计来解决；
6. 在远端服务器上的工程部署实现了从0到1的效果，但是实际使用体验不好，特别是宕机后的再恢复以及设计大量数据操作的用户操作；
7. 工程前端元素较为统一，基本基于bootstrap / md-bootstrap的设计规范开发，但是对于一些移动设备的适配还不够完美。