<Hobbitat校园兴趣爱好分享社区>

系统测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <2023/1/10> | <1.0> | <第一版系统测试报告> | <徐雨杰> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

本文档是对“Hobbitat校园兴趣爱好分享社区”项目进行系统测试后的报告文档

## 范围

本文档影响项目开发组（CS3604 第二组）开发的“Hobbitat校园兴趣爱好分享社区”网站。本网站将面向上海交通大学全体在校师生，为他们提供分享兴趣爱好的线上交流平台。

## 定义、首字母缩写词和缩略语

本组：指 CS3604 第二组的四名成员；

校园：指上海交通大学；

兴趣爱好：指用户倾向于从事的业余活动；

关注(follow)：指某用户关注其他用户的各种动向的关系；

帖子(post)：由用户主动发布的包含文字、图片等内容。

评论/回帖(comment)：名词，在已经存在的帖子下发表的内容，通常与原帖子内容相关；

回复(reply)：动词，指发布 comment 的行为。

分享：指在网站上发布包含自己从事的某项活动的相关文字、图片的帖子，或在他人评论区评论此类内容；

分区：按照兴趣爱好种类划分。通常一个分区中的帖子有同一主题，即属于同一大类的兴趣爱好。

互动：包括一个用户为其他用户点赞、评论、关注其他用户、私聊其他用户的行为。

## 参考资料

[1] 沈备军, 陈昊鹏, 陈雨亭. 软件工程原理[M]. 高等教育出版社, 2013.

## 概述

本文档接下来的部分将包括测试概要说明，测试环境说明，测试结果及分析，缺陷清单，测试结论及分析。

# 测试概要

全体项目开发组成员对照系统测试用例对项目进行了测试

# 测试环境

Windows系统

Python 3.8+

其他模块的版本见requirements.txt

# 测试结果及分析

## 需求覆盖率及缺陷分布

*[需求覆盖率是指经过测试的需求/功能和软件需求规约中所有需求/功能的比值，通常情况下要达到100%的目标。需给出功能测试及非功能测试的需求覆盖情况。对于功能测试，需给出各个主要功能模块的需求覆盖率。如果需求覆盖率未达到100%可在备注项中说明原因及未测试内容。缺陷率是指本项缺陷占总缺陷数的百分比。请尽量使用图、表进行描述。]*

**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** | 2 | 100% | 0 | 0% |  |
| **注册** | 3 | 100% | 1 | 33% |  |
| **注销** | 1 | 100% | 0 | 0% |  |
| **发帖** | 1 | 100% | 0 | 0% |  |
| **回复** | 1 | 100% | 0 | 0% |  |
| **点赞** | 1 | 100% | 0 | 0% |  |
| **收藏** | 2 | 100% | 0 | 0% |  |
| **搜索** | 1 | 100% | 0 | 0% |  |
| **关注** | 2 | 100% | 0 | 0% |  |
| **举报** | 2 | 100% | 0 | 0% |  |
| **通知** | 2 | 100% | 0 | 0% |  |
| **删除** | 1 | 100% | 0 | 0% |  |
| **管理员删帖** | 1 | 100% | 0 | 0% |  |
| **个人主页管理** | 2 | 100% | 1 | 50% |  |
| **内容推荐** | 1 | 100% | 0 | 0% |  |
| **私聊** | 1 | 100% | 0 | 0% |  |
| **功能项小计** | 24 | 100% | 2 | 8.3% |  |
| **非功**  **能项** | **性能** | 2 | 100% | 0 | 0% |  |
| **可靠性** | 3 | 100% | 0 | 0% |  |
| **易用性** | 3 | 100% | 1 | 33% |  |
| **安全性** | 1 | 100% | 0 | 0% |  |
| **兼容性** | 1 | 100% | 1 | 100% |  |
| **界面** | 3 | 100% | 0 | 0% |  |
| **压力承载** | 1 | 100% | 0 | 0% |  |
| **非功能项小计** | *15* | *100%* | *2* | *13%* |  |
|  | **总计** | *39* | *100%* | *4* | *10%* |  |

## 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 | 4 | 4 |
| **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 | Bug001 | Minor | 注册 | 无提示 | 注册出现失败时未给予用户提示 | ZJ-006 |
| 2 | Bug002 | Minor | 个人主页 | 未处理 | 上传头像未压缩和剪裁 | ZC-002 |

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 | Bug003 | Minor | 易用性 | 不美观 | 部分页面不够美观，缺乏吸引力 | USE-003 |
| 2 | Bug004 | Minor | 兼容性 | 不兼容 | 与移动设备兼容性差 | COMP-001 |

# 测试结论与建议

经过本轮测试，我们发现大部分需求已经实现并通过了测试。

对某些细微的缺陷，其修复需要花费大量时间。而部分缺陷则是在设计之初就没有考虑到。

这给予了我们一些启示。即要在设计阶段就让系统少出错，并具有拓展性和容错性。

对这些缺陷，我们需要花时间进行改进，并进行重新测试。