即应匹配信息发布平台

测试报告

版本 <5.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <8日9/月2019/年> | <5.0> | 迭代五测试报告 | 第11组 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

本测试报告是 用来记录“即应匹配信息发布平台”开发过程中的功能及非功能需求的完成情况，主要目的是发现设计和实现中的缺陷和不足，然后进行修复和改善。

## 范围

此测试报告撰写于迭代三功能完成之后，测试范围包含对迭代三的功能测试和非功能测试，包括tag生成，推荐活动等功能。此测试报告基于测试完成之后，对测试中使用的方法、测试的有效程度进行总结。

## 定义、首字母缩写词和缩略语

即应——即应匹配信息发布平台

Auth service —— 即应微服务架构中的用于登录、验证的服务

User service —— 即应微服务架构中用于用户信息的添加、修改、查询、删除的服务

Api gateway —— 即应微服务架构中用于禁停http请求，将请求转发，调用其他服务的服务

Activity service —— 即应微服务架构中用于活动的发起、删除、修改、查询，评论，以及tag生成，活动推荐的服务

Feedback service —— 即应微服务架构中用于对用户提供反馈，包括反馈的发起，删除，评论功能。

## 参考资料

《即应项目软件需求规约》 v1.2

《即应第五次迭代计划》 v1.0

《即应匹配信息发布平台 第五次迭代 测试用例》 v1.0

## 概述

本测试报告包含测试概要、测试环境、测试结果及分析、测试缺陷报告等。

# 测试概要

## 功能性测试

时间：2019年8月

人员：柳清源，荆家振，赵胜龙，戴方越

测试方法：单元测试，人工测试

测试内容：对项目的所有功能进行系统测试，包括后端单独测试以及前后端集成测试

# 测试环境

## 功能性测试

人工测试对后端使用Postman，运行于windows 10 1803系统。前端使用人工点击的方式进行测试。单元测试使用golang 1.12 自带的单元测试模块进行测试。

# 测试结果及分析

[对于所做的各种测试，指出其结果。例如测试功能点数、测试用例数、缺陷数等。]

[给出主要测试结果分析的图形或表格表示，如需求覆盖率、缺陷分布图等。]

## 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** | **发布反馈** | 4 | 100% |  |  |  |
| **查找反馈** | 3 | 100% |  |  |  |
| **评论反馈** | 4 | 100% |  |  |  |
| **删除反馈** | 5 | 100% |  |  |  |
| **发布活动** | 9 | 100% |  |  |  |
| **查询单条活动** | 3 | 100% |  |  |  |
| **修改活动** | 13 | 100% |  |  |  |
| **查询我的活动** | 2 | 100% |  |  |  |
| **添加评论** | 8 | 100% | 1 |  | 12.5% |
| **参与活动** | 9 | 100% | 1 |  | 11.1% |
| **查看活动参与申请** | 4 | 100% |  |  |  |
| **接受活动参与申请** | 9 | 100% | 1 |  | 11.1% |
| **获取用户已发送且未被接受的申请** | 3 | 100% |  |  |  |
| **查询特定种类的活动** | 4 | 100% |  |  |  |
| **查询活动成员及评价** | 4 | 100% |  |  |  |
| **退出活动** | 7 | 100% |  |  |  |
| **查看活动退出概率** | 5 | 100% |  |  |  |
| **拒绝活动申请** | 6 | 100% |  |  |  |
| **查看被拒绝的活动** | 1 | 100% |  |  |  |
| **确认申请被拒绝** | 2 | 100% |  |  |  |
| **关注用户** | 6 | 100% | 1 |  | 16.7% |
| **取消关注用户** | 4 | 100% | 3 |  | 75% |
| **查看已关注的人** | 2 | 100% |  |  |  |
| **查看好友** | 1 | 100% |  |  |  |
| **上传头像** | 1 | 100% |  |  |  |
| **搜索活动** | 2 | 100% |  |  |  |
| **分页** | 1 | 100% |  |  |  |
| **搜索外卖店家** | 2 | 100% |  |  |  |
| **查看隐私的用户信息** | 3 | 100% |  |  |  |
| **浏览活动** | 3 | 100% |  |  |  |
| **登陆** | 2 | 100% |  |  |  |
| **查看活动详情信息** | 1 | 100% |  |  |  |
| **接受活动申请** | 1 | 100% |  |  |  |
| **查看用户信息** | 1 | 100% |  |  |  |
| **分类搜索** | 1 | 100% |  |  |  |
| **我的相关信息** | 4 | 100% |  |  |  |
| **浏览历史记录** | 1 | 100% |  |  |  |
| **编辑个人信息** | 1 | 100% |  |  |  |
| **功能项小计** | 142 | 100% |  |  |  |

## 缺陷严重程度

*[描述缺陷按严重程度分布情况，包括不同严重程度缺陷个数及占缺陷总数百分比等。尽量使用图、表表示。]*

**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 | 1 | 1 | 5 | 7 |
| **占缺陷百分比** |  | 14% | 14% | 72% |  |

# 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 | 1 | Low | 添加评论 | 添加评论未对receiver不存在的情况做检查 | 如果后端添加评论时 receiver不存在时，本应返回错误，但是却返回“评论成功” | jing-testcase-156 |
| 2 | 2 | Urgent | 参与活动 | 加入不存在的活动时错误 | 在发送请求的时候，若是act-id对应的活动不存在，本应返回错误，但是却返回空指针错误 | jing-testcase-163 |
| 3 | 3 | Low | 接受活动参与申请 | 测试申请对应的活动已被删除的情况时错误 | 本应返回404错误，但是却返回403错误 | jing-testcase-176 |
| 4 | 4 | Medium | 关注用户 | 测试用户是否可以关注不存在的用户  时错误 | 当用户关注不存在的用户时，本应返回400错误但是却返回200。 | jing-testcase-214 |
| 5 | 5 | Low | 取消关注用户 | 测试用户是否可以取消关注未关注的用户时错误 | 本应返回400错误但是却返回200 | jing-testcase-216 |
| 6 | 6 | Low | 取消关注用户 | 测试用户是否可以取消关注不存在的用户时错误 | 本应返回400错误但是却返回200 | jing-testcase-217 |
| 7 | 7 | Low | 取消关注用户 | 测试用户是否可以取消关注自己  时错误 | 本应返回400错误但是却返回200 | jing-testcase-217 |

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。

# 测试结论与建议

本次测试主要是对之前的测试进行补充测试和集成测试，距离编写代码的时间比较远，因此在修复bug的时候工作量较大。之后测试应当尽早进行