Exort

测试报告

版本 <0.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 11/7/2019 | 0.1 | 第一次迭代验收测试报告 | 张万强 |
| 22/7/2019 | 0.2.1 | 第二次迭代验收测试报告 | 邓公淯 |
|  |  |  |  |
|  |  |  |  |

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

第二次迭代测试，主要是针对各模块间的协同和整体功能集合做一次完整的功能型测试。

## 范围

本次测试分为两个部分，一个是基于Junit在maven中进行的单元测试，另一部分是通过继续完善的前端服务页面来对后端提供的接口进行对接测试。

## 定义、首字母缩写词和缩略语

## 参考资料

## 概述

# 测试概要

|  |  |
| --- | --- |
| 名称 | 描述 |
| 时间 | 2019.07.22 |
| 地点 | 软件学院3101 |
| 人员 | 张万强，邓公淯，沈小洲 |
| 测试方法 | 单元测试以及人工功能性测试 |
| 测试内容 | 用户登陆注册以及管理员进行用户管理 |

# 测试环境

前端：尚未编写自动化测试，采用手动测试

后端: Junit,同时采用Postman进行补充的手动测试

# 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** | **用户信息及账户管理** | 11 | 100% | 0 | 0 | 0 |
| **活动管理** | 7 | 100% | 0 | 0 | 0 |
| **社团管理** | 9 | 100% | 4 | 45% | 0 |
| **社团成员管理** | 17 | 100% | 0 | 0 | 0 |
| **权限管理** | 38 | 100% | 0 | 0 | 0 |
| **接口层接口转调** | 5 | 90% | 2 | 67% | 0 |
|  | **功能项小计** | 87 | 100% | 6 | 7% | 0 |
| **非功能项** | **可靠性** | 1 | 80% | 0 | 0 | 0 |
|  | **非功能项小计** | 1 | 80% | 0 | 0 | 0 |
|  | **总计** | 88 | 100% | 6 | 7% | 0 |

## 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 | 4 | 0 | 4 |
| **占缺陷百分比** | 0 | 0 | 100% | 0 | 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 | Apiserver-2 | 一般 | 可靠性 | 社团管理接口转发错误 | API接口转发存在错误 |  |
| 2 | Apiserver-3 | 一般 | 可靠性 | 活动管理接口转发错误 | API接口转发存在错误 |  |
| 3 |  |  |  |  |  |  |

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** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# 测试结论与建议

由于本周的工作主要集中在进行前后端接口的对接以及各模块功能的整合，所以大部分测试都是针对各模块功能的正确性以及前后端对接的正确性进行测试，性能型测试要放在下周在整体的完整型确定之后再进行测试，当前的可靠性测试也表现出并不特别乐观的结果，各模块之间的相互协调之间依旧存在问题。