**{{project\_name}}测试报告**

**撰 写 人： {{writer}}**

**撰 写 日 期：{{write\_time}}**

# 编写目的

本测试报告为{{project\_name}}的测试报告，目的在于总结测试阶段的测试以及分析测试结果，描述系统是否符合需求，并依据结果对该产品做出评价和建议，以供后续项目参考。

Bug统计与分析

于项目来言，Bug分为有效、无效。无效Bug是指解决方案为不是问题、设计如此、无法重现、不予解决等不需要处理的问题。在本报告中，只统计有效Bug，重点关注实际处理的问题，突出项目层面的总结。其判断依据为：解决方案=需求变更、已解决。

数据统计截止时间：{{write\_time}}前。

本项目有效Bug总计：{{bug\_number}}个。

## 按模块

<<module\_table>>

<<module\_image>>

**分析：**

就Bug数量而言，{{max\_bug\_module}}模块最多，但其相较其余模块逻辑较复杂，属正常情况；其余模块Bug数量也在正常范围内。

## 按严重程度

|  |  |
| --- | --- |
| 严重程度 | 说明 |
| A | Blocker：阻碍开发或测试工作的问题。   * 造成系统崩溃、死机、死循环，系统无法再使用 * 导致数据库数据丢失 * 与数据库连接错误 * 主要功能丧失 * 基本模块缺失等问题 * 界面显示引起客户极大误解，法律风险，甚至政治风险   如：代码错误、死循环、数据库发生死锁、重要的一级菜单功能不能使用等 |
| B | Critical：系统无法执行、崩溃或严重资源不足、应用模块无法启动或异常退出、无法测试、造成系统不稳定。具体基本上可分为：   * 严重花屏 * 内存泄漏 * 用户数据丢失或破坏 * 系统崩溃/死机/冻结，杀掉进程或者刷新还可以正常使用其他功能 * 模块无法启动或异常退出 * 严重的数值计算错误 * 功能设计与需求严重不符 * 用户权限问题（涉及到数据安全问题，比如试用用户能看到正式用户的专属资源；中央平台，运营型不能操作包装型标签这种算C等级） * 安全问题 * 其它导致无法测试的错误 |
| C | Major：影响系统功能或操作，主要功能存在严重缺陷，但不会影响到系统稳定性具体基本上可分为：   * 功能未实现或与需求不符 * 功能错误 * 系统刷新错误 * 语音或数据通讯错误 * 轻微的数值计算错误 * 边界条件下错误 * 系统所提供的功能或服务受明显的影响 |
| D | Minor：界面、性能缺陷 具体基本上可分为：   * 操作界面错误（包括数据窗口内列名定义、含义是否一致） * 提示信息错误（包括未给出信息、信息提示错误等） * 长时间操作无进度提示 * 系统未优化（性能问题） * 光标跳转设置不好，鼠标（光标）定位错误 |
| E | Trivial：易用性及建议性问题具体基本上可分为：   * 界面格式等不规范 * 辅助说明描述不清楚 * 操作时未给用户提示 * 可输入区域和只读区域没有明显的区分标志 * 个别不影响产品理解的错别字 * 文字排列不整齐等一些小问题 * 建议 |

<<bug\_level\_table>>

<<bug\_level\_image>>

**分析：**

1. C类占比最多，是由于开发时间较短，研读需求不仔细，导致该问题，开发前应认真阅读需求，减少bug数量。
2. 无A类、B类bug，说明开发提交的代码质量较好。

## 按解决方案

<<resolve\_table>>

<<resolve\_image>>

**分析：**

1. 解决方案为{{max\_bug\_resolve}}的Bug数量最多，占比为{{fix\_proportion}}，说明主要问题是由代码造成的。
2. 其中延期解决的Bug有{{second\_bug\_number}}个。

## 按激活次数

激活次数：如果开发修复问题后，测试不通过，重新激活指派给开发，激活次数就会+1。

<<activated\_count\_table>>

<<activated\_count\_image>>

**分析：**

1. {{activated\_zero}}的Bug，是一次性解决通过的。
2. 在可接受范围内，修改效率良好。

## 按Bug类型

<<bug\_type\_table>>

<<bug\_type\_image>>

**分析：**

<<Pareto\_Chart>>

1. 从帕累托图可以看到，符合二八原理。即百分八十的问题是由百分之二十的原因造成的。
2. 代码错误问题比较突出，需要适当减少。

## 按开发人员

<<exploitation\_table>>

<<exploitation\_image>>

## 按测试人员每天提交的bug数

<<test\_submit\_days>>

## 按测试人员每天关闭的bug数

<<test\_close\_days>>

## 遗留Bug清单

<<postpone\_bug\_table>>

# 测试情况

本次测试的测试对象为{{project\_name}}。

测试类型为功能测试，同时对兼容性、数据、异常处理、易用性等也进行了验证。

测试方法使用了边界值、等价类划分、场景法、错误推测法、发散测试、随机测试等。

本次测试共前中后台三个模块，前台分为PC端及移动端，对各个模块进行了用例评审。本次主要是由于项目时间较为紧张，且开发阅读需求文档不仔细，导致bug数量较多。

从DI值上来看，预期是{{expect\_DI}}个，实际有{{actual\_DI}}个。{{diff}}

**项目测试下来，也发现测试方面存在一些不足之处，期望后续改进：**

1. 业务、算法这块，需要多自学一些，多主动了解一些
2. 质量把控方面，数据这块，属于核心内容，还有些欠缺，对于数据模块的了解不够，导致在测试中耗费的时间较久

评价

## 测试结论

经测试，本系统在功能上达到需求要求，测试通过。

满足系统测试出口准则：

1. 按照系统测试计划完成了系统测试。
2. 测试用例的需求覆盖率和测试用例执行率均达到100%。
3. 累计发现缺陷数和累计关闭缺陷数两条曲线稳定并趋向重合，连续三天每天新增BUG≤3个，且无A类、B类发现。
4. 系统遗留Bug得到相关负责人的确认，并有对应的风险防范措施。

## 缺陷、限制及建议

需求方面，数据的取法做了多次的改动，且有些特殊值的取法未在文档中体现，导致给的内容比较混乱，不够明确，希望以后尽量完善算法文档。

在测试过程中，数据模块较为复杂，测试时间较为紧张，希望在后续的项目中，能预留更多的时间给到测试。

附：后期优化改造内容

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 描述 | 影响 | 处理建议 |
| 1 | 暂无 | -- | -- |