

# 测试报告

<Cealgull>

测试报告

版本 <1.0>

## 修订历史记录

日期	版本	说明	作者
14/7/2023	0.1	前端测试工具选取与初步使用	任柏俊，王浩丞
9/9/2023	0.2	前端测试全覆盖与完备	任柏俊，王浩丞
9/9/2023	1.0	后端测试覆盖	吴逸洋，徐轲

## 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

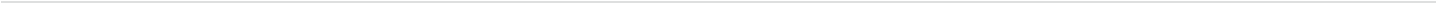
4.2缺陷严重程度5

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

1. 简介

目的

通过测试，熟悉测试基本思想与要求，配置测试工具与环境，验证实现功能函数输出符合预期值，确保界面用户操作逻辑可以正常实现。在测试中生成完备详尽的测试报告，发现测试的缺陷并加以评估。

范围

前端测试范围涉及：

测试项目Cealgull.app, 测试环境react native, 测试工具Jest, react-native-testing-library

后端测试范围设计：

Middleware, Verify 测试环境golang Chaincodes 测试环境 Hyperledger Fabric 测试工具go test, mockery

定义、首字母缩写词和缩略语

无

参考资料

无

概述

测试文档将主要介绍测试流程进度，测试结果，分析测试缺陷和原因，并最终总结得出结论

## 2. 测试概要

### 前端测试概要：

测试阶段	测试时间	测试地点	测试人员	测试方法	测试内容
第一阶段	2023年6月19日-2023年7月16日	软件大楼开发教室	任柏俊，王浩丞	在react native中使用jest，react-native-testing-library等测试框架进行标准化测试  对verify服务器进行单元测试覆盖	已经开发完成的前端页面框架，部分已完成组件的行为逻辑，以及页面间跳转操作的实现测试  完成证书签发、签名验证、重签名
第二阶段	2023年7月17日-2023年9月9日	学校，住所等	任柏俊，王浩丞	继续熟悉前端测试相关思想与工具，熟练应用其解决更多测试问题  对于网络请求测试需要与后端分离的要求，引入新的工具miragejs 来模拟网络请求的后端反馈  对middleware服务器进行单元测试覆盖  对智能合约进行单元测试覆盖 对后端系统整体进行性能以及冒烟测试	在前端内容的不断开发中跟进相关测试。不但要实现基本的渲染测试，还要实现交互操作测试，网络请求测试，错误处理测试等各方面内容  后端内容开发中需要不断更新单元测试代码，还需要编写对整个系统正确性验证的冒烟测试，同时到最后需要对整个系统添加性能测试。

### 后端测试概要：

测试阶段	测试时间	测试地点	测试人员	测试方法	测试内容
第一阶段	2023年6月19日-2023年7月16日	软件大楼开发教室	吴逸洋，徐轲	对verify服务器进行单元测试覆盖	完成证书签发、签名验证、重签名
第二阶段	2023年7月17日-	学校，住所等	吴逸洋，徐轲	对middleware服务器进行单元测试覆盖	后端内容开发中需要不断更新单元测试代码，还需要编写对整个系统正确性验证的冒烟测试，同

	-2023年 9月9日		对智能合约进行单元测试覆盖 对 后端系统整体进行性能以及冒烟 测试	时在最后需要对整个系统添加性 能测试。
--	----------------	--	---	------------------------

### 3. 测试环境

前端环境：

React Native，TypeScript，测试工具 Jest， react-native-testing-library，miragejs

后端环境：

mockery golang，测试工具mockery locust python requests

### 4. 测试结果及分析

#### 4.1 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	前端：界面绘制	57	98%	1	2%	
	前端：交互反馈与状态保存	92	98%	2	2%	
	前端：网络请求	42	100%	0	0%	
	前端：错误处理	105	100%	0	0%	
	前端：非用户可见功能支持	44	97.7%	1	2.3%	
	后端：智能合约	44	100%	0	0%	
		220	100%	0	0%	

	后端： Middlew are					
	后端： Verify	33	100%	0	0%	
	功能项小 计	413	99.3%	3	0.7%	
非功 能项	性能	50	98%	1	2%	
	可靠性	20	100%	0	0%	
	非功能项 小计	70	98.6%	1	1.4%	
	总计	483	99.2%	4	0.8%	

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	2	2	4
占缺陷百分比	0%	0%	50%	50%	100%

5. 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	1	一般	前端：交互反馈与状态保存	图片裁剪缺陷	图片选取后无法进行即刻裁剪，从而修正至预期结果	ImagePicker_test
2	2	细微	前端：界面绘制	Toast提示窗口缺陷	缺乏简洁的Toast提示窗口，采用了一种格式较为固定的外部库实现，未达到简洁的结果	Toast_test
3	3	一般	前端：交互反馈与状态保存	分页设置	限于性能，前端一次不能请求全部话题，需要分页获取，对于新获取的话题页，无法稳定实现将其“推入”底部的显示逻辑，只能刷新现存的一页内容	Pagination_test
4	4	细微	前端：用户非可见功能	用户登陆状态保存	登录用户状态获取函数useUser ()只获取本地内容，但是这往往是过时的，我们不得不在每次获取个人信息时放弃缓存，改为重新发送网络请求，用户数据缓存的效率提升有限	useUser_test

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。
  - 1. 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。
  - 2. 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。
  - 3. 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。
  - 4. 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。

- 3. 功能模块：所测试并出现该缺陷的功能模块名称。
- 4. 缺陷标题：描述缺陷的标题。
- 5. 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	1	一般	性能	前端话题请求缺陷	当一次性请求超过50个时，前端相应将明显变慢	forum.topic.list_test

6. 测试结论与建议

前端测试

遇到的问题：

- 1. 前端单元测试中，部分函数无法在单个测试组件中正常执行，需要mock该类函数以使测试按照预期正常进行
- 2. 前端网络测试中，需要和后端完全分离，只考察前端获取网络数据后的处理与反馈，需要引入工具mirage来模拟后端数据反馈
- 3. 测试结构分析：测试结果实现分支，行，函数高覆盖，充分测试了界面渲染，交互逻辑，网络请求，错误处理等相关测试要点，测试比较成功，较好的符合预期

结论与建议：

- 1. 前端测试应该以用户操作逻辑直觉展开测试，并保持这一原则的高优先度，尽量涉及和抓取用户可见，可交互的组件进行重点测试与开发，以确保前端服务型逻辑的正常运行
- 2. 面对前端测试的众多技术要求，应该在一定程度上借鉴开源社区的相关测试库，以提供更加强大与全面的测试功能

3. 前端测试不仅限于数据校验，需求复杂，为了提高测试的准确性和测试的覆盖率，需要更多时间与精力的投入。

## 后端测试

### 结论与建议：

1. 减少If分支，或者使用防御性编程来提高单元测试编写键单独
2. 性能测试需要混合编写，使用locust来测试用户的实际行为
3. 冒烟测试要符合前端操作逻辑，来测试系统更新后软件的正确性