基于区块链的电子病历共享信息系统

第三迭代测试评估报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018/11/4 | 0.1 | 增加简介和前端页面测试报告 | 蒋海波 |
| 2018/11/4 | 0.2 | 增加java后台接口测试报告 | 张宇燊 |
| 2018/11/4 | 0.3 | 增加获取数据地址的智能合约测试报告 | 朱良 |
| 2018/11/4 | 1.0 | 整理文档 | 蒋海波 |

目录

1. 简介 2

1.1 目的 2

1.2 范围 2

1.3 定义、首字母缩写词和缩略语 2

1.4 参考资料 2

1.5 概述 2

2. 测试结果摘要 2

3. 基于需求的测试覆盖 2

4. 基于代码的测试覆盖 2

5. 建议措施 2

6. 图 2

测试评估报告

# 简介

## 目的

本次测试旨在测试基于区块链的电子病历共享信息系统第三迭代实现的功能。

## 范围

本次测试范围为三个方面：前端的基础页面，java后台获取数据库数据的接口，获取数据地址的智能合约。

## 定义、首字母缩写词和缩略语

**Remix IDE:** 一款基于浏览器的Solidity语言的编译器。

# 前端页面测试

## 测试用例

|  |  |  |
| --- | --- | --- |
| 用例 | 步骤 | 预期 |
| 1 | 打开页面，输入全部医院信息，点击提交 | 页面正常显示，能正常输入医院信息，点击提交按钮后显示添加成功 |
| 2 | 打开页面，输入不完整的医院信息，点击提交 | 页面提示请输入参数 |

## 测试执行

|  |  |  |  |
| --- | --- | --- | --- |
| 测试序号 | 执行用例 | 通过 | Bug原因 |
| 1 | 1，2 | 1 | 判断输入参数是否为空的逻辑写错了 |
| 2 | 1，2 | 1，2 |  |

## 测试结果

最终前端页面功能均已实现并通过测试。其中发现bug数1个，修复完成bug数1个，未修复bug数0个。

## 图



图2-1 用例1结果



图2-1 用例2结果

# 后台接口测试

# 智能合约测试

## 测试结果摘要

本轮测试本别针对基于需求的测试覆盖和基于代码的测试覆盖两方面展开测试工作。

### 基于需求的测试覆盖

|  |  |  |  |
| --- | --- | --- | --- |
|  | 需求名称 | 需求详情 | 测试结果 |
| 功能性 | 接口一：  新增授权地址 | 1. 如果智能合约维护的授权地址列表中没有该地址，那么向列表中添加新地址； 2. 只有一个特定的地址能够使用改地址，其余地址调用该接口时，不生成有效区块，操作无效。 | 测试通过 |
| 接口二：  医院登陆接口 | 1. 授权的医院前端能够通过身份认证； 2. 同时，该接口返回一个随机生成的token，作为该医院的认证标识。 | 测试通过 |
| 接口三：  token验证接口 | 1. 能够验证医院的token是否真实，同时是否超时； 2. 两个条件都满足时，返回true，其余情况返回false。 | 测试通过 |
| 非功能性 | 易用性需求 | 接口便于前端开发人员使用。 | 测试通过 |
| 可靠性需求 | 平均可用时间≥94.00%；MTBF≥150天。 | 测试通过 |
| 性能需求 | 平均占用内存≈0.5 GB，占用内存≤8.0 GB。 | **测试未通过** |
| 可支持性需求 | 易于扩展接口，内部方法易于维护和调用。 | 测试通过 |

### 基于代码的测试覆盖

|  |  |  |
| --- | --- | --- |
| 测试种类 | 测试方法 | 测试结果 |
| 路径覆盖测试 | DD路径测试 | 通过 |

## 基于需求的测试覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 需求详情 | 实现方法 | 测试过程 | 结果 |
| 功能性 | 接口一：新增授权地址 | add\_authorized\_address( address \_addr ) | Input:  “0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c”  Check:  address [] authorized\_addr中新增输入地址。 | address [] authorized\_addr中新增输入地址。  功能实现。 |
| 接口二：医院登陆接口 | login () | 1. 以地址为“0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c”的前端调用接口二； 2. 以其他地址的前端调用接口二。 | 1. token\_item [] token\_list中新增token\_item，成员变量地址(addr)为“0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c”，隐式返回token为14995； 2. 系统提示：“transact to listAuthentication.login errored: VM error: revert. revert The transaction has been reverted to the initial state. Reason provided by the contract: "you have not authorized yet.".”，拒绝了该地址前端的登录访问。   功能实现。 |
| 接口三：token验证接口 | token\_verification ( uint \_token ) | Input: “14995”、“10000”  Output:  “true”、“false” | 功能实现。 |
| 非功能性 | 易用性：接口便于前端开发人员使用。 | | 检查接口的注释是否详细。 | 接口使用说明详细。测试通过。 |
| 可靠性：平均可用时间≥94.00%；MTBF≥150天。 | | 同时开启20个浏览器，运行Remix IDE，每分钟调用所有接口各三次，持续1小时不关闭，记录测试结果。 | 接口运行良好，所有浏览器重的remix IDE均正常运行。  测试通过。 |
| 性能：平均占用内存≈1.0 GB，占用内存≤8.0 GB。 | | 在5台内存为16GB的设备上，运行Remix IDE各一次，多次调用接口一、接口二和接口三，各1000次。 | 接口一和接口三运转正常，但占用2.0GB内存。接口二出现占用内存过大警告并出现重载现象。  **测试未通过。** |

## 基于代码的测试覆盖

### 新增授权地址接口的DD路径测试

基于add\_authorized\_address( address \_addr )代码的DD路径图，如图4-1所示。

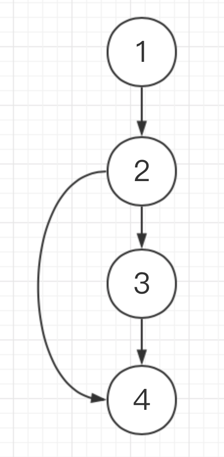


图4-1

关键节点对应的代码声明如下：

节点1:起始节点。

节点2:61-62行，require(!address\_verification(\_addr),

节点3:63行，"This address has already been authorized.");

节点4:72行，authorized\_addr.push( \_addr );

测试详情如下：

|  |  |  |
| --- | --- | --- |
| 路径 | 用例 | 测试结果 |
| 路径1:1-2-3-4 | 输入：0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c | 加入列表成功。  测试通过 |
| 路径2:1-2-4 | 输入0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c后再次输入0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c | 加入列表失败。  测试通过 |

### 医院登陆接口的DD路径测试

基于login()代码的DD路径图，如图4-2所示。

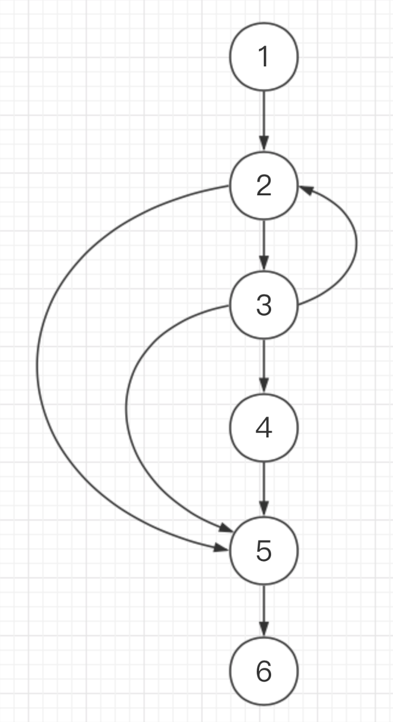


图4-2

关键节点对应的代码声明如下：

节点1:起始节点。

节点2:82行，for ( uint i = 0 ; i < list\_length ;

节点3:83行，if ( authorized\_addr[i] == addr\_tmp ){

节点4:84-85行，token\_list.push( token\_item({ token: ret\_token , addr: addr\_tmp , creation\_time: now}));break;

节点5: require(i==list\_length-1,"you have not authorized yet.");

节点6: return ret\_token;

测试详情如下：

|  |  |  |
| --- | --- | --- |
| 路径 | 用例 | 测试结果 |
| 路径1:1-2-3-4-5-6 | 地址为0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c申请登陆。 | 登陆成功，返回token为678445。  测试通过 |
| 路径2:1-2-5-6 | 清空授权列表后，尝试调用登陆接口。 | 登陆失败。  测试通过 |
| 路径3:1-2-3-5-6 | 使用未授权的地址登陆。 | 登陆失败。  测试通过 |
| 路径4:1-2-3-2-5-6 | 再新加入一个授权地址0xDB35b7d915458EF540aDe6068dFe2F44E8fa987e并使用改地址申请登陆。 | 登陆成功，返回token为45631。  测试通过 |

### token验证接口

基于token\_verification ( uint \_token )代码的DD路径图，如图4-3所示。

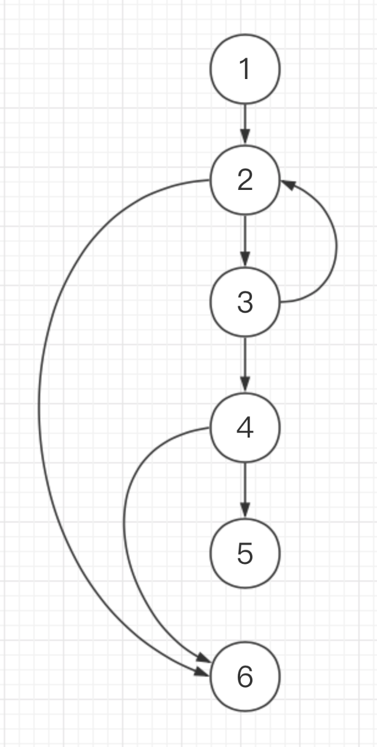


图4-3

关键节点对应的代码声明如下：

节点1:起始节点。

节点2:96行，for ( uint i = 0 ; i < list\_length ;

节点3:97行，if ( token\_list[i].token == \_token )

节点4:98行，if( token\_list[i].creation\_time + 2 hours >= now )

节点5:99行，return true;

节点6:104行，return false;

测试详情如下：

|  |  |  |
| --- | --- | --- |
| 路径 | 用例 | 测试结果 |
| 路径1:1-2-3-4-5 | 使用地址0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c在得到的token后的五分钟之内，调用该接口。 | 返回true。  测试通过 |
| 路径2:1-2-3-4-6 | 使用地址0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c在得到的token后的五分钟之内，调用该接口。 | token超时，返回false。  测试通过 |
| 路径3:1-2-6 | 清空token列表之后，尝试调用该接口。 | 返回false。  测试通过 |
| 路径4:1-2-3-2-6 | 加入多个token，并使用另外一个不再列表中的token为参数调用该接口。 | 返回false。  测试通过 |

## 建议措施

针对内存需求过大导致性能不佳的情况，需要进一步优化代码的信息存储机制。

# 总结

本次测试结果为项目第三迭代所需完成的功能均已实现并基本通过测试，部分未通过测试的内容将会在后续迭代中修改和优化。