**数据区块链存储系统-测试报告**

一、 概述 1

1.1项目简介 2

1.2参考资料 2

1.3相关文档 2

1.4版本更新信息 2

二、 目标系统功能需求 2

三、 目标系统其他需求 2

四、 功能测试报告 3

4.1测试环境 3

4.1.1硬件运行环境 3

4.1.2软件运行环境 3

4.2单元测试 3

4.2.1 测试描述 3

4.2.2测试用例 5

4.2.3测试结果 6

4.2.4测试结果记录 8

4.3集成测试 9

4.4系统测试 9

五、 性能测试报告 9

六、 其他测试结果 9

6.1 压力测试 9

6.2 安全性测试 9

6.3 可移植性测试 9

七、 不符合项列表 9

八、 测试结果 10

# 概述

## 1.1项目简介

区块链是一种按照时间顺序将数据区块以顺序相连的方式组合成的一种链式数据结构， 并以密码学方式保证的不可篡改和不可伪造的分布式账本。以其去中心化、开放性、自治性、信息不可篡改、匿名性等特性，被广泛运用在金融行业中。

Hyperledger Fabric是一个企业级，开源的分布式账本解决方案的区块链架构平台。其基础是模块化架构，提供了高保密性、弹性、灵活性和可伸缩性。它支持不同组件的拔插式实现，并能够适应整个经济生态系统中存在的复杂性。

本项目使用Hyperledger Fabric技术，对金融服务平台的基本功能进行封装，实现相关操作入链，为平台更上层的结构提供对应接口。

## 1.2参考资料

[1]吕云翔，软件工程实用教程，清华大学出版社，2015

[2]吕云翔，刘瀚诚，刘天毅，软件工程项目实训教程，清华大学出版社，2016

[3] Hyperledger Fabric官方文档

[4]Ian Sommerville著，软件工程（原书第10版），彭鑫等译，机械工业出版社，2018

[5]张海藩，吕云翔，软件工程（第4版），人民邮电出版社，2013

## 1.3相关文档

数据区块链存储系统《需求规格说明书》《软件开发计划书》《配置文档》以及需求报告编写指南。

## 1.4版本更新信息

版本更新信息：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 创建者 | 创建日期 | 维护者 | 维护日期 |
| V1.0 | 李威 | 2018/5/30 | 李威 | 2018/5/30 |
| V1.1 | 李威、许鸿智、闫坤 | 2018/6/5 | 许鸿智、闫坤 | 2018/6/5 |

# 目标系统功能需求

参见区块链存储系统的《需求规格说明书》

# 目标系统其他需求

参见区块链存储系统的《需求规格说明书》

# 功能测试报告

## 4.1测试环境

### 4.1.1硬件运行环境

Intel® Core™ i7-8700k CPU @ 3.70GHz

DDR4 16G

50G 机械硬盘

### 4.1.2软件运行环境

dubbo-monitor-simple 2.0.0

zookeeper-3.4.10

go1.9.4

Docker 18.03.0-ce

## 4.2单元测试

### 4.2.1 测试描述

单元测试主要使用白盒测试，对接口方法进行逻辑覆盖。测试分为以下四个方法，其名称及流程图如下：

1. InsertTransaction方法

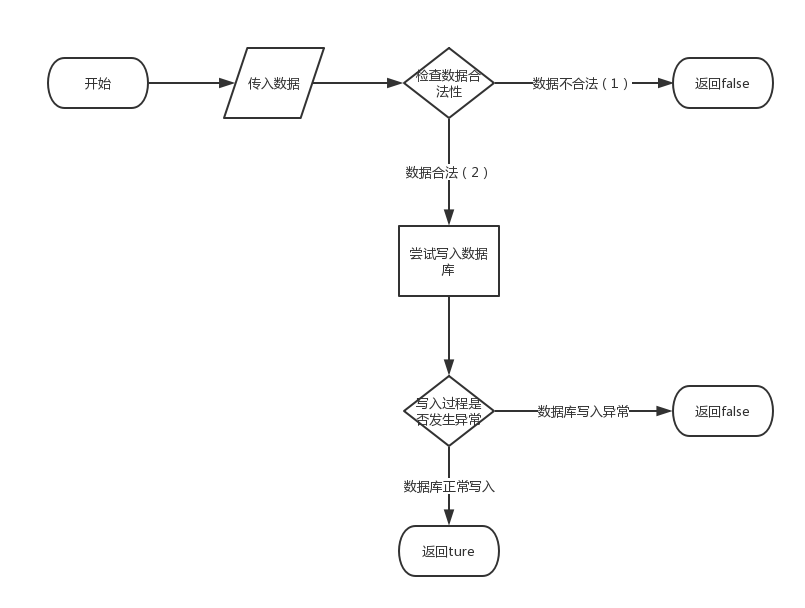


图4.1“InsertTransaction方法”流程图

1. QueryTransaction方法

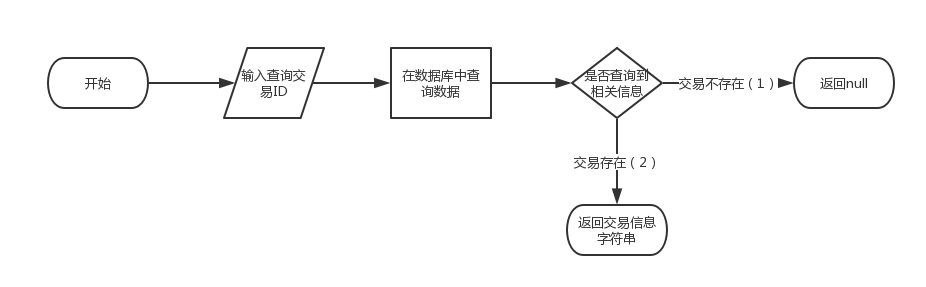


图4.2“QueryTransaction方法”流程图

1. InsertBalanceChange方法

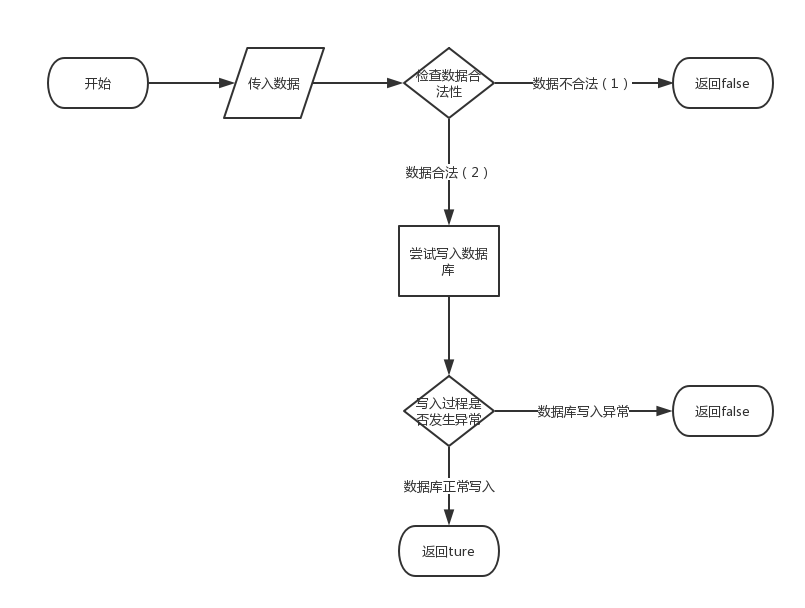


图4.3 “InsertBalanceChange方法”流程图

1. QueryBalanceChange方法

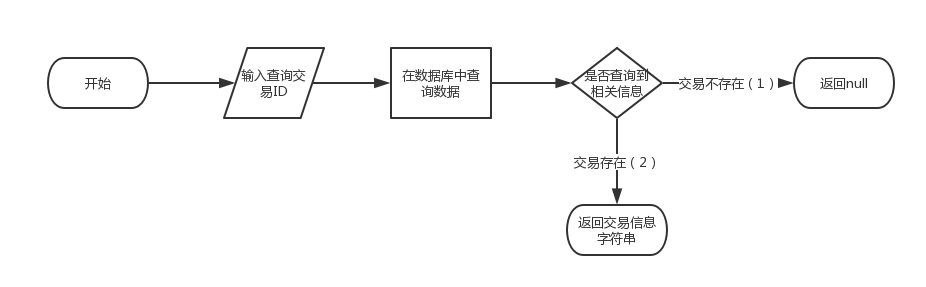


图4.4“QueryBalanceChange方法”流程图

### 4.2.2测试用例

1. InsertTransaction方法

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 输入数据 | | | | | | | | 预期输出 | 分支覆盖 |
| recordId | paymentInstitutionId | paymentUserId | collectionInstitutionId | collectionUserId | dateTime | TransactionType | sum |
| 1 | 1000 | 100 | 12 | 34 | 23 | “2018” | true | 23.33 | true | 2 |
| 2 | null | 100 | 12 | 34 | 23 | “2018” | true | 23.22 | false | 1 |

表格4.1 “InsertTransaction方法”测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入数据 | 预期输出 | 分支覆盖 |
| recordId |
| 1 | 1000 | {"recordId":"1000","paymentInstitutionId":"100","paymentUser\_id":"12""collectionInstitutionId":"34","collectionUser\_id":"23","dateTime":"2018" ,"type":"true","sum":"23.33"} | 2 |
| 2 | 998 | null | 1 |

1. QueryTransaction方法

表格4.2“QueryTransaction方法”测试用例

1. InsertBalanceChange方法

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 输入数据 | | | | | | 预期输出 | 分支覆盖 |
| recordId | institutionId | userId | dateTime | TransactionType | sum |
| 1 | 2000 | 123 | 234 | 2018 | true | 234.23 | true | 2 |
| 2 | null | 123 | 234 | 2018 | true | 234.23 | false | 1 |

表格4.3 “InsertBalanceChange方法”测试用例

1. QueryBalanceChange方法

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入数据 | 预期输出 | 分支覆盖 |
| recordId |
| 1 | 2000 | {"recordId":"2000","institutionId":"123","user\_id":"234","dateTime":"2018" ,"type":"true","sum":"232.234"} | 2 |
| 2 | 1998 | null | 1 |

表格4.4 “QueryBalanceChange方法”测试用例

### 4.2.3测试结果

InsertTransaction方法测试结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 1 | 功能名称 | InsertTransaction | |
| 功能描述 | 参见数据区块链存储系统《需求规格说明书》 | | | |
| 测试步骤 | 输入 | 交易记录ID (recordId)  付款方机构ID (paymentInstitutionId)  付款方用户ID (paymentUser\_id)  收款方机构ID (collectionInstitutionId)  收款方用户ID (collectionUser\_id)  交易日期 (dateTime)  交易类型 (type)  交易金额 (sum) | | |
| 输出 | 1.true  2.false | | |
| 测试结果 | 通过 | 发现问题 | | 无 |
| 测试人 | 许鸿智 | 测试时间 | | 2018-6-8 |

表格4.5 “InsertTransaction方法”测试结果

QueryTransaction方法测试结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 2 | 功能名称 | QueryTransaction | |
| 功能描述 | 参见数据区块链存储系统《需求规格说明书》 | | | |
| 测试步骤 | 输入 | 交易记录ID (recordId) | | |
| 输出 | 1.{"recordId":"1000","paymentInstitutionId":"100","paymentUser\_id":"12""collectionInstitutionId":"34","collectionUser\_id":"23","dateTime":"2018" ,"type":"true","sum":"23.33"}  2.null | | |
| 测试结果 | 通过 | 发现问题 | | 无 |
| 测试人 | 许鸿智 | 测试时间 | | 2018-6-8 |

表格4.6 “QueryTransaction方法”测试结果

InsertBalanceChange方法测试结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 3 | 功能名称 | InsertBalanceChange | |
| 功能描述 | 参见数据区块链存储系统《需求规格说明书》 | | | |
| 测试步骤 | 输入 | 交易记录ID (recordId)  机构ID (institutionId)  用户ID (user\_id)  交易日期 (dateTime)  交易类型 (type)  交易金额 (sum) | | |
| 输出 | 1.true  2.false | | |
| 测试结果 | 通过 | 发现问题 | | 无 |
| 测试人 | 许鸿智 | 测试时间 | | 2018-6-8 |

表格4.7 “InsertBalanceChange方法”测试结果

QueryBalanceChange方法测试结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 4 | 功能名称 | QueryBalanceChange | |
| 功能描述 | 参见数据区块链存储系统《需求规格说明书》 | | | |
| 测试步骤 | 输入 | 交易记录ID (recordId) | | |
| 输出 | 1.{"recordId":"2000","institutionId":"123","user\_id":"234","dateTime":"2018" ,"type":"true","sum":"232.234"}  2.null | | |
| 测试结果 | 通过 | 发现问题 | | 无 |
| 测试人 | 许鸿智 | 测试时间 | | 2018-6-8 |

表格4.8 “QueryBalanceChange方法”测试结果

### 4.2.4测试结果记录

“InsertTransaction方法”测试结果记录如表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 预期结果 | 实际结果 | 测试结果 | 测试人 | 测试时间 |
| 1 | True | True | 通过 | 许鸿智 | 2018-6-8 |
| 2 | False | False | 通过 | 许鸿智 | 2018-6-8 |

表格4.9 “InsertTransaction方法”测试结果记录

“QueryTransaction方法”测试结果记录如表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 预期结果 | 实际结果 | 测试结果 | 测试人 | 测试时间 |
| 1 | {"recordId":"1000","paymentInstitutionId":"100",  "paymentUser\_id":"12""collectionInstitutionId":"  34","collectionUser\_id":"23","dateTime":"2018" ,  "type":"true","sum":"23.33"} | {"recordId":"1000","paymentInstitutionId":"100",  "paymentUser\_id":"12""collectionInstitutionId":"  34","collectionUser\_id":"23","dateTime":"2018"  ,"type":"true","sum":"23.33"} | 通过 | 许鸿智 | 2018-6-8 |
| 2 | null | null | 通过 | 许鸿智 | 2018-6-8 |

表格4.10 “QueryTransaction方法”测试结果记录

“InsertBalanceChange方法”测试结果记录如表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 预期结果 | 实际结果 | 测试结果 | 测试人 | 测试时间 |
| 1 | True | True | 通过 | 许鸿智 | 2018-6-8 |
| 2 | False | False | 通过 | 许鸿智 | 2018-6-8 |

表格4.9 “InsertBanlanceChange方法”测试结果记录

“InsertTransaction方法”测试结果记录如表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 预期结果 | 实际结果 | 测试结果 | 测试人 | 测试时间 |
| 1 | {"recordId":"2000","institutionId":"123",  "user\_id":"234","dateTime":"2018" ,"typ  e":"true","sum":"232.234"} | {"recordId":"2000","institutionId":"123",  "user\_id":"234","dateTime":"2018" ,"typ  e":"true","sum":"232.234"} | 通过 | 许鸿智 | 2018-6-8 |
| 2 | False | False | 通过 | 许鸿智 | 2018-6-8 |

表格4.9 “QueryBalanceChange方法”测试结果记录

## 4.3集成测试

集成测试涉及系统五个模块的整合，故不在本报告中阐述。

## 4.4系统测试

集成测试涉及系统五个模块的整合，故不在本报告中阐述。

# 性能测试报告

经测试，接口方法的调用未发生异常。

# 其他测试结果

## 6.1 压力测试

对本模块使用压力测试，测试结果为服务器线程池最多可同时容纳200个查询指令。

## 6.2 安全性测试

系统可防止错误信息、异常修改带来的不良后果：

1. 交易信息在区块链中加密存储
2. 区块链信息的不可修改性
3. 保留日志文件
4. 信息插入前进行合法性检查

## 6.3 可移植性测试

经测试，本模块可在配置好必要软件环境的Linux、Windows、MacOS系统下正常运行，对硬件配置要求不高。

# 不符合项列表

无

# 测试结果

测试完成时间：2018-6-8

测试地点：学生公寓

测试环境：多个开发者不同配置的个人电脑

参与测试人员：许鸿智、闫坤、李威

系统强项：可移植性好，安全性较高

系统弱项：运行效率有待提升

不符合项的统计结果：无

组长签字：闫坤