

中山大学数据科学与计算机学院本科生实验报告

(2019 年秋季学期)

课程名称：区块链原理与技术

任课教师：郑子彬

年级	2017 级	专业 (方向)	软件工程
学号	16341011	姓名	李裕民
电话	15819838552	Email	1186695278@qq.com
开始日期	2019 年 10 月 18 日	完成日期	2019 年 12 月 12 日

最终制品放在 github 网址如下：

<https://github.com/Allensuki/BlockChain-Final-HW>

一、项目背景

1. 传统供应链金融：

在传统供应链金融中，只有在金融机构（银行）对该车企的信用评级很高，才可作为被认可的企业，认为他有很大的风险承担的能力。如果该公司需要借款时，金融机构认可该车企（核心企业）的还款能力，因此愿意借款给轮胎公司。但是，这样的信任关系并不会往下游传递。

这个过程将增加很多经济成本，而这个问题主要是由于该车企的信用无法在整个供应链中传递以及交易信息不透明化所导致的。这无疑不利于中小型企业的发展，因为只要不是直接与被认证企业有关系，其他企业都是很难相信这个企业的。

2. 区块链+供应链金融：

将供应链上的每一笔交易和应收账款单据上链，同时引入第三方可信机构来确认这些信息的交易，例如银行，物流公司等，确保交易和单据的真实性。同时，支持应收账款的转让，融资，清算等，让核心企业的信用可以传递到供应链的下游企业，减小中小企业的融资难度。

3. 要求：

基于已有的开源区块链系统 FISCO-BCOS (<https://github.com/FISCO-BCOS/FISCO-BCOS>)，以联盟链为主，开发基于区块链或区块链智能合约的供应链金融平台，实现供应链应收账款资产的溯源、流转。

二、 方案设计

本次设计综合参考了老师和助教给出的教学资料以及在网上查找了许许多多其他的案例，从而完成了此次的大作业内容。

1.存储设计

对于银行和公司，这里需要创建的内容有资产账户、资产金额、以及信用额度。在此可以创建两个表来进行存储设计。一个对应资产账户和资产金额，另一个则对应于资产账户和信用额度。

在此我们可以用函数 `function createTable() private{}来` 进行创建。

其中的主要语句如下：

```
TableFactory create =TableFactory(0x1001);  
create.createTable("t_asset","account","value");  
create.createTable("t_credit","account","credit");
```

对于任何公司和银行来说，注册用户即可获得两个初始值，资产金额、以及信用额度，分别存放到以上所说的两个表里面。对于银行则初始化时按照最大的额度进行初始化即可。

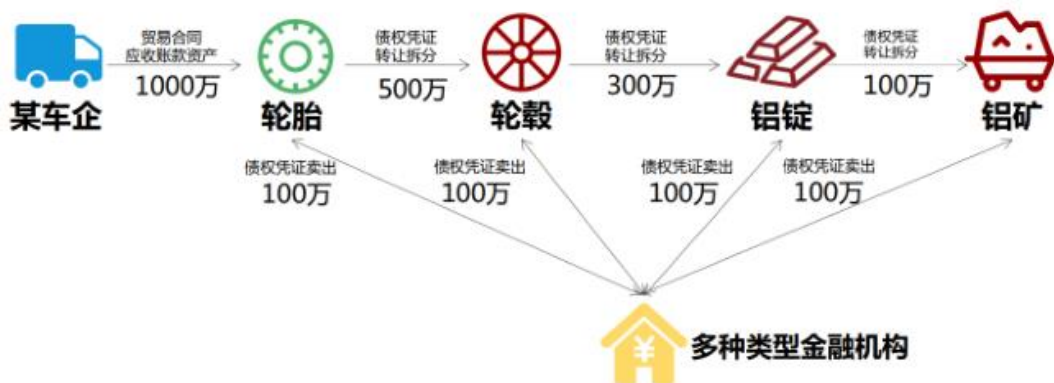
以上即为存储设计。

2 数据流图

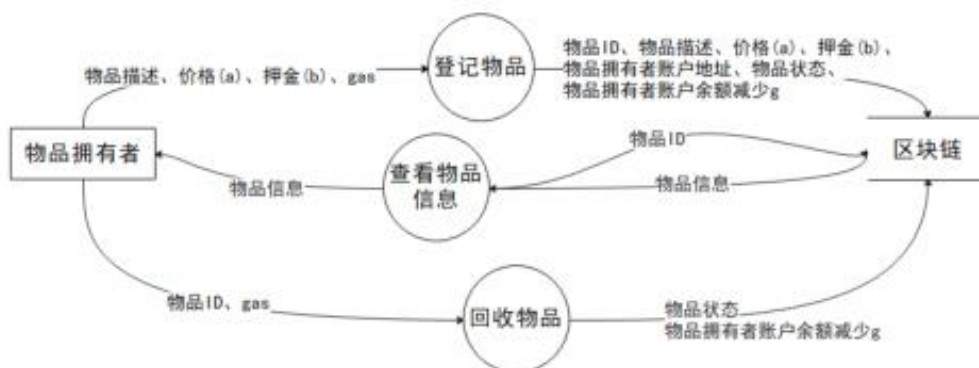
供应链应收账款资产的溯源、流转：



场景介绍：



根据老师及助教发给我们的资料可以知道数据流图有如下设计：



3.核心功能介绍

核心功能有如下几点：

首先我们需要创建用户这一简单的操作，对于银行则初始化时按照最大的额度进行初始化即可。同时创建出错时或者账户名重复时需要返回报错。代码如下：

```
if(ret != 0){
    Table table = openTable();
```

```

Entry entry =table.newEntry();
entry.set("account", account);
entry.set("asset value", int256(value));
int count=table.insert(account, entry);
table =opentable1();
entry =table. newEntry();
entry.set("account", account);
entry.set("credit", int256(credit))
count= table.insert(account, entry);
if (count==1){
ret_code=0;
}
else{
ret_code=-2;//创建失败。
}
else{
ret_code =-1;//账户重复。
}

```

此外，我们还需要查询用户信息的功能。用来查询用户信息：资产账户、资产金额、以及信用额度。代码如下：

```

else{
Entry entry= entries. get(0);
uint256 res1=uint256(entry getInt("value"));
table =openTable1();
entries table.select(account, table.newCondition());
entry =entries.get(0);
uint256 res2=uint256(entry.getInt("credit"));
return (0, res1, res2);
}

```

同时，两个账户之间可以进行交易，交易时可以直接进行简单的加减操作即可，并在表格中进行更新。代码如下：

```

Table table= openTable1();
Entry entry0 =table.newEntry();
entry0.set("account", from_account);
entry0.set("credit", int256(from_credit -amount));
int count =table.update(from_account, entry0, table.newCondition());
if(count != 1){
ret_code =-5;
emit TransferEvent(ret_code, from_account, to_account, amount);
return ret_code;
}

```

```

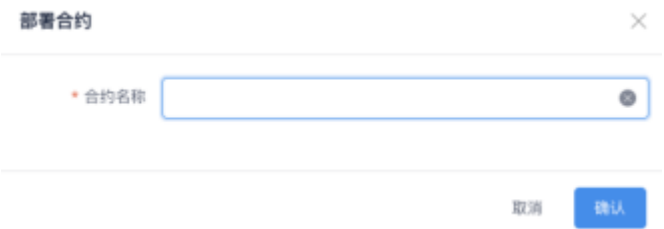
Entry entry1 =table.newEntry();
entry1.set("account", to_account);
entry1.set("credit", int256(to_credit + amount));
table.update(to_account, entry1, table.newCondition());
emit TransferEvent(ret_code, from_account, to_account, amount);
return ret_code ;

```

同时，在交易之前，我们需要传递相关的参数，即交易的双方账户名以及金额。我们还需要进行简单的判断，例如判断交易双方的账户是否存在，同时判断交易后账户的金额是否会溢出等等。进行相应的细节上的补充。

三、 功能测试

首先，我们部署合约如下：



创建一个用户如图所示：



然后我们创建两个用户分别为 A 和 B,A 和 B 的用户资产设为 100 和 200。如图所示：

用户名称	用户资产	用户信用额度
A	100	200
B	100	200

此时，我们假设 A 向 B 使用信用买了 100 的东西，那么 A 的信用将减少 100，变成 100，B 的信用将会增加，变为 300。结果如图所示：

用户名称	用户资产	用户信用额度
A	100	100
B	100	300

功能测试成功。

四、 界面展示

界面展示如图所示：

<input type="checkbox"/>	用户名称	用户资产	用户信用额度
暂无数据			
共 10 条 < 1 > 跳至 1 页			

<input type="checkbox"/>	合约名称	合约地址
<input type="checkbox"/>	暂无数据	
共 10 条 < 1 > 跳至 1 页		

五、 心得体会

本学期选修了老师的《区块链原理与技术》课程，使我学习到了许多有关区块链以及区块链货币方面的知识。

这次区块链的大作业是要求我们基于已有的开源区块链系统 FISCO-BCOS (<https://github.com/FISCO-BCOS/FISCO-BCOS>)，以联盟链为主，开发基于区块链或区块链智能合约的供应链金融平台，实现供应链应收账款资产的溯源、流转。这次的实验对我来说难度很大，主要在于前端和后端的开发方面上。API 接口方面也十分的令人困惑。再加上就是编程方面，由于自己的水平很差，所以在这方面的内容存在很大的困难。前端和后端应该如何去写，如何去设计我对此都没有很清楚的认知。不过好在老师及助教上课时的认真讲解下，还有同学的耐心帮助之下，我都一一克服了以上的困难。在此对老师及助教表达由衷的感谢之情，谢谢！

最后，由于我曾因病休学一年，所以自己的功底薄弱，但我也会在今后的学习生活中继续学习有关区块链的知识与技能。在这次大作业中肯定存在许多错误和纰漏，还恳请老师及助教多多理解与通融，感激不尽谢谢！