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	17	专业（方向）	软件工程
学号	17343129	姓名	熊伟淇
电话	13246806552	Email	1024386569@qq.com
开始日期	2019.11.2	完成日期	2019.11.13

一、 项目背景

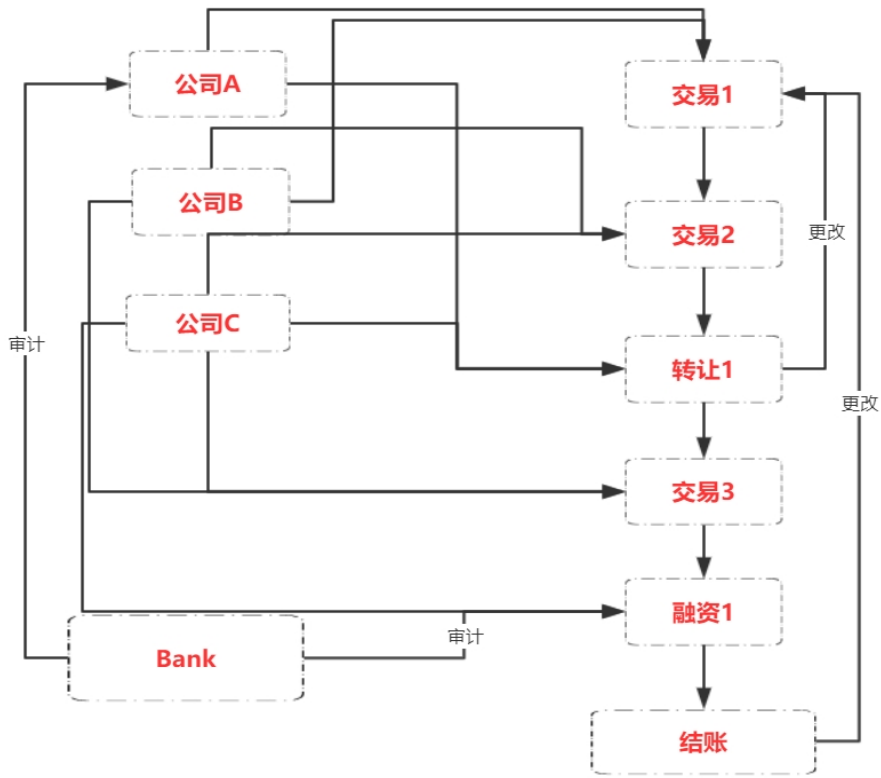
基于已有的开源区 块 链 系 统 FISCO-BCOS
（https://github.com/FISCO-BCOS/FISCO-BCOS），以联盟链为主，开发基于区块链或区块链智能合约的供应链金融平台，实现供应链应收账款资产的溯源、流转。
环境搭建基于 WEBase 以及 FISCO-BCOS 官方文档
https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/
场景介绍：
某车企（宝马）因为其造车技术特别牛，消费者口碑好，所以其在同行业中占据绝对优势地位。因此，在金融机构（银行）对该车企的信用评级将很高，认为他有很大的风险承担的能力。在某次交易中，该车企从轮胎公司购买了一批轮胎，但由于资金暂时短缺向轮胎公司签订了 1000 万的应收账款单据，承诺 1 年后归还轮胎公司 1000 万。这个过程可以拉上金融机构例如银行来对这笔交易作见证，确认这笔交易的真实性。在接下里的几个月里，轮胎公司因为资金短缺需要融资，这个时候它可以凭借跟某车企签订的应收账款单据向金融结构借款，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。在某个交易中，轮胎公司从轮毂公司购买了一批轮毂，但由于租金暂时短缺向轮胎公司签订了 500 万的应收账款单据，承诺 1 年后归还轮胎公司 500 万。当轮毂公司想利用这个应收账款单据向金融机构借款融资的时候，金融机构因为不认可轮胎公司的还款能力，需要对轮胎公司进行详细的信用分析以评估其还款能力同时验证应收账款单据的真实性，才能决定是否借款给轮毂公司。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。
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。

二、 方案设计

存储设计：
自定义结构 Company 和 Proposal，Bank 作为特殊的 Company 具体为：
struct Company {

```
    string name;    //公司名
    uint credibility;       //信誉度
    bool valid;       //注销
    uint property; //资产
}

struct Proposal {
    address owner;       //收款地址
    address apayer;   //付款地址
    string payer;       //付款方
    string payee;       //收款方
    uint amount;        //金额
    uint credibility;   //信誉度
    Status status;      //应收账款状态
}
```

数据流



实现功能：

功能一：实现采购商品—签发应收账款 交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。

功能二：实现应收账款的转让上链，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款。

功能三：利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资。

功能四：应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款

源码

```
function transaction(address receiver, uint amount) public{
        if(receiver==msg.sender)
        return ;
        proposals.push(Proposal(receiver,msg.sender,com[msg.sender].name, com[receiver].name, amount, com[msg.sender].credibility, Status.Active));
        emit Transaction(com[msg.sender].name, com[receiver].name, amount);
    }

    //转让收据，有三种情况，优先用本公司拥有的收款公司的收据来抵消，不够再将高信誉度的公司的收据整合起来交给对方，还是不够两公司之间就生成收据
    function transfer(address receiver, uint amount) public{
        if(receiver==msg.sender)
            return;

        uint i;
        uint tot=amount;

        //优先转让对方的收据
        for(i = 0; i < proposals.length; i++){
            if(proposals[i].owner==msg.sender&&proposals[i].status == Status.Active&&proposals[i].credibility>0){
                if(proposals[i].amount<=tot){
                    proposals[i].status=Status.Accomplished;
                    proposals.push(Proposal(receiver,proposals[i].apayer,proposals[i].payer, com[receiver].name, proposals[i].amount,proposals[i].credibility, Status.Active));
                    tot-=proposals[i].amount;
                }
                else{
                    proposals[i].status=Status.Accomplished;
                    proposals.push(Proposal(receiver,proposals[i].apayer,proposals[i].payer, com[receiver].name, tot,proposals[i].credibility, Status.Active));
                    proposals.push(Proposal(proposals[i].owner,proposals[i].apayer,proposals[i].payer, proposals[i].payee, proposals[i].amount-tot,proposals[i].credibility, Status.Active));
                    tot = 0;
                    break;
                }
            }
        }
        if(tot !=0)
            com[msg.sender].property=com[msg.sender].property-tot;
        emit Transfer(com[msg.sender].name, com[receiver].name, com[msg.sender].name, amount);
    }
```

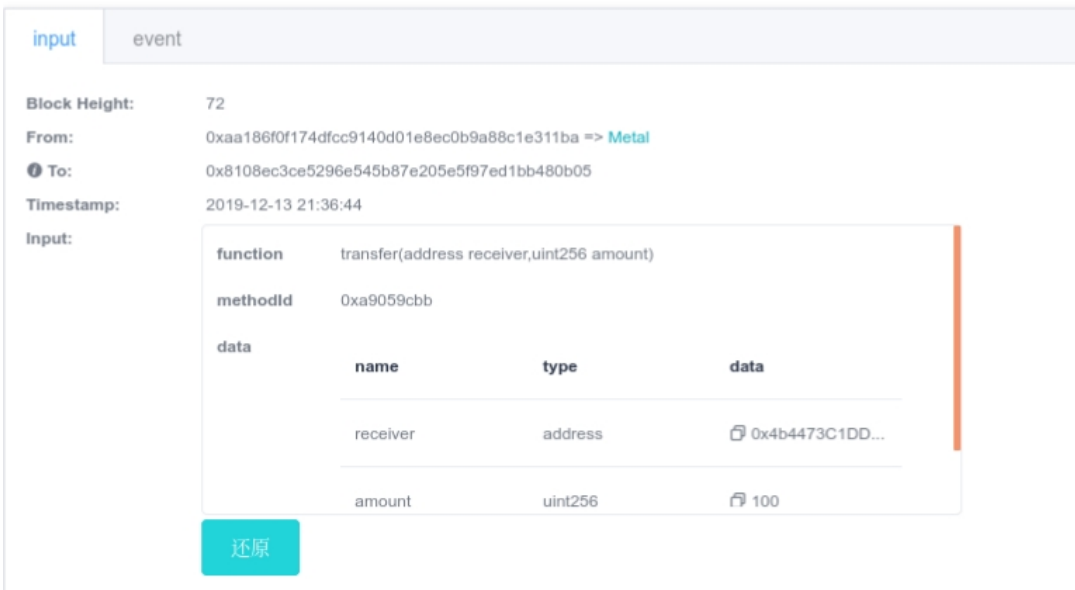
```
    //融资，高信誉度公司可以任意融资，非高信誉度公司根据所持有高信誉度公司的收据来决定融资的额度
    function financing(uint amount) public{
        uint i;

        //高信誉度公司可以任意融资
        if(com[msg.sender].credibility >0){
            proposals.push(Proposal(bank,msg.sender,com[msg.sender].name, sbank, amount,
com[msg.sender].credibility, Status.Active));
            com[msg.sender].property += amount;
            emit Financing(com[msg.sender].name, com[msg.sender].property, 1);
        }
        else{
            uint used = 0;
            uint sum = 0;

            //已经使用的融资额度
            for(i = 0; i < proposals.length; i++){
                if(proposals[i].owner==bank&&proposals[i].status == Status.Active&&proposals[i].credibility>0)
                    used += proposals[i].amount;
            }

            //计算剩余融资额度是否超过所需金额，如果不超过，则失败
            for(i = 0; i < proposals.length; i++){
                if(proposals[i].owner==msg.sender&&proposals[i].status ==
Status.Active&&proposals[i].credibility>0){
                    sum+=proposals[i].amount;
                    if(used+amount>sum)
                    emit Financing(com[msg.sender].name, com[msg.sender].property, 0);
                    return ;
                }
            }
            proposals.push(Proposal(bank,msg.sender,com[msg.sender].name, sbank, amount,
com[msg.sender].credibility, Status.Active));
            com[msg.sender].property += amount;
            emit Financing(com[msg.sender].name, com[msg.sender].property, 1);
        }
    }
    //结账，参数为欲偿还公司的地址，根据收据创建先后顺序自动还款，直到剩余资产无法还款
    function settlement(address receiver) public{
        //收款公司未注册
        if(receiver==msg.sender)
            return;

        for(uint i = 0;  i < proposals.length; i++){
            if(proposals[i].status == Status.Active&&proposals[i].apayer==msg.sender){
                //资产足够偿还收据金额
```

```
            if(com[msg.sender].property >= proposals[i].amount){
                com[msg.sender].property -= proposals[i].amount;
                com[receiver].property += proposals[i].amount;
                 proposals[i].status = Status.Accomplished;
                emit Settlement(com[msg.sender].name, com[receiver].name, proposals[i].amount,
com[msg.sender].property);
            }
            else
                break;
        }
    }
}
```

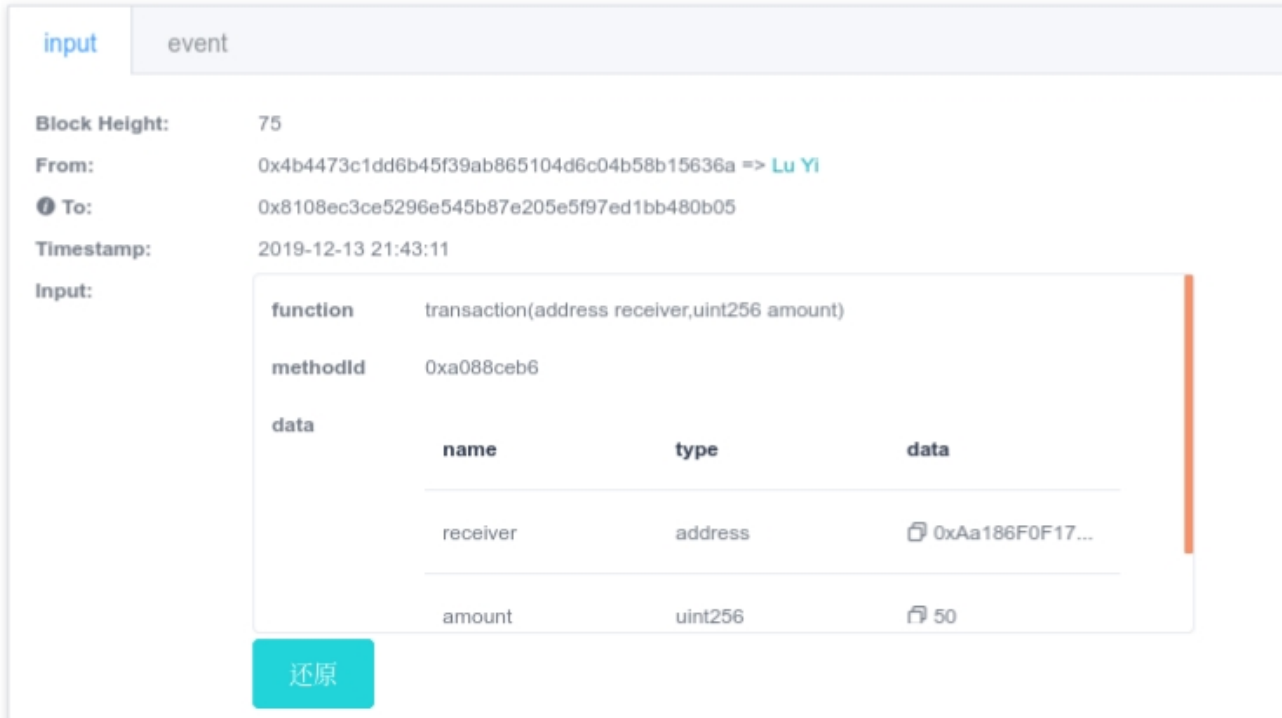
三、 功能测试

实验截图实现功能：

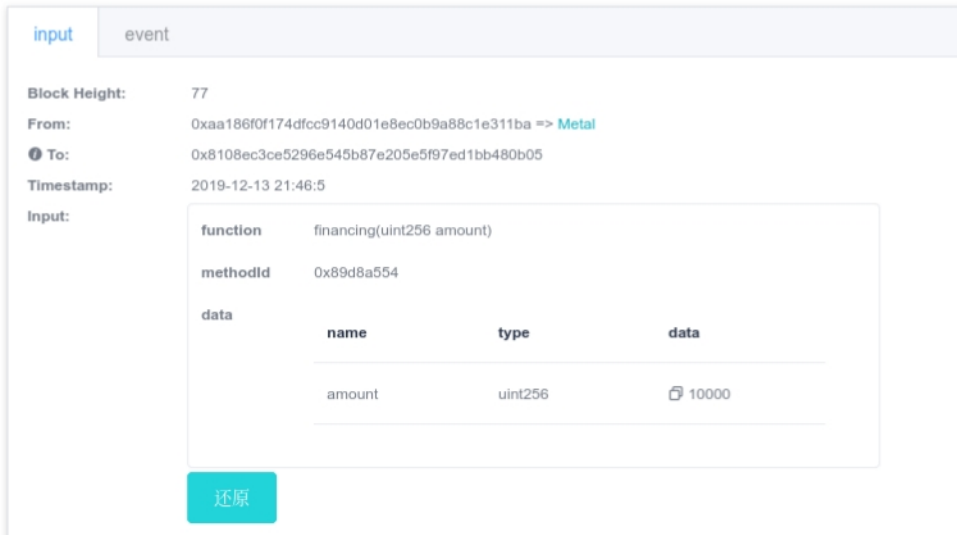
功能一：实现采购商品—签发应收账款 交易上链。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。



功能二：实现应收账款的转让上链，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的应收账款单据部分转让给轮毂公司。轮毂公司可以利用这个新的单据去融资或者要求车企到期时归还钱款



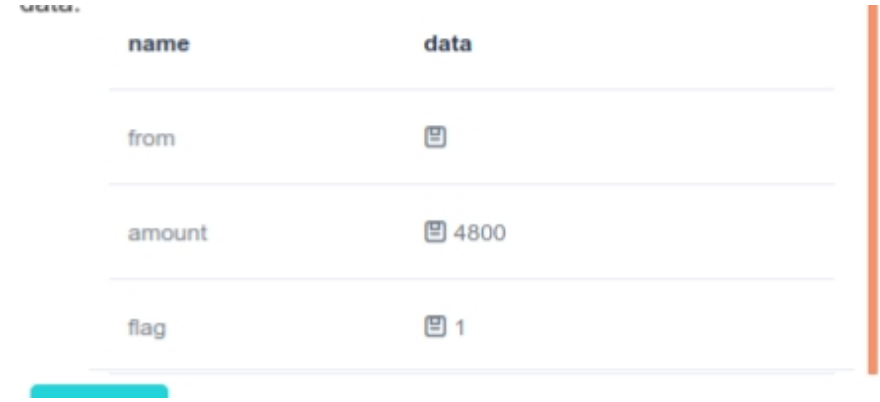
功能三：利用应收账款向银行融资上链，供应链上所有可以利用应收账款单据向银行申请融资



功能四：应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款

交易内容

交易后：



功能四：应收账款支付结算上链，应收账款单据到期时核心企业向下游企业支付相应的欠款

四、 界面展示

基于 WEBase 的前端网页显示交易及节点信息

五、 心得体会

本次大作业主要是综合考察学生对于区块链相关概念的理解以及智能合约开发能力，在大作业提供的应用背景下，对相关应用场景的供应链金融开发主要有以下优点：

易于双方信任的获取传统金融业为了维护交易双方的信任关系，催生了大量的高成本、低效率、单点为核心的中介机构，包括银行、第三方支付平台、托管机构、公证人、交易所等，专门从事信息搜集、信息处理和风险甄别能力，缓解交易双方的信息不对称，从而解决逆向选择和道德风险问题。而区块链技术的特性却恰好可以跨过这些第三方中介机构，直接实现信任的获取。理论上，在技术识别能力足够的情况下，区块链能让交易双方无须借助任何第三方信用中介即可开展经济活动，完全实现交易双方之间的信息公开共享，从而实现全球低成本的价值转移。

提高数据安全保密性区块链技术通过复杂的计算机加密算法，利用分布式存储的技术架构，完整地保存了所有历史交易信息，极大地提高了区块链交易系统中数据的安全性和保密性。其主要的关键技术包含两个方面：一是数据加密和签名机制；二是共识算法。

降低信息安全风险传统互联网金融业务依托在中心服务器上，往往伴随着网站安全、数据安全、交易安全、制度建设等等问题，这就给了各类网络黑客和贪婪的内幕交易员有了可趁之机。而基于区块链技术的新金融业务系统，充分利用区块链技术的特点构建了一个完全自治的系统，采用 P2P 网络的方式分布式存储，因此不会出现中心模式下的网站安全问题；系统中所有数据都采用计算机加密算法，所有数据不可被修改，因此区块链系统中数据更加安全可靠；系统中发生的所有交易都需要向全网公开且得到确认之后才会加入区块链节点，且所有历史交易数据都被全部保存并可追溯，因此区块链系统中的交易更加真实可靠。

要实现如此多的优势，区块链开发过程也对设计者有着极大的考验，在合约的架构方面，不仅要考虑数据结构的合理性，避免冗余，也要顾及数据流的走向，设计者常常会忽略去中心化的数据结构，造成合约的不健全。在功能实现方面则需要考虑使用者的实际需求，预留接口，达到优化用户体验的目的。

经过漫长的开发过程，我对自己的能力局限性也有了充足的认识，在今后会不断加强。