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	17343105	姓名	田皓
电话		Email	
开始日期		完成日期	

一、 项目背景

基 于 已有的 开 源 区 块 链 系 统 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。**

场景介绍：传统供应链金融：

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，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。

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

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

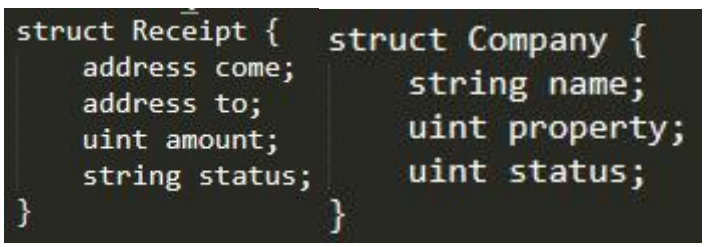
期时归还钱款。

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。

二、 方案设计

存储设计：



使用两个结构体来储存账单和公司的信息。

核心功能介绍：

部署合约：

```
1.  public void deployAssetAndRecordAddr() {
2.          try {
3.              Asset asset = Asset.deploy(web3j, credentials, new StaticGasProvider(gasPric
    e, gasLimit)).send();
4.              System.out.println(" deploy Asset success, contract address is " + asset.get
    ContractAddress());
5.              recordAssetAddr(asset.getContractAddress());
6.          } catch (Exception e) {
7.              System.out.println(" deploy Asset contract failed, error message is  " + e.g
    etMessage());
8.          }
```

```
    }
```

使用 Asset 项目原本的合约部署函数即可

公司注册：

```
public void register(String addr, String name, BigInteger amount, BigInteger status) {
        try {
            String contractAddress = loadAssetAddr();
            Test2 asset = Test2.load(contractAddress, web3j, credentials, new StaticGasProvider(gasPrice, gasLimit));
            TransactionReceipt receipt = asset.register(addr, name, amount).send();
            System.out.println(receipt);
            List<RegisterEventResponse> response = asset.getRegisterEvents(receipt);
            if (!response.isEmpty()) {
                System.out.println("register successful:" + response.get(0).name);
            } else {
                System.out.println("register error.\n");
            }
        } catch (Exception e) {
            System.out.println("register error" + e.getMessage());
        }
    }
```

签订账单：

```
public void receivable_account(String addr, BigInteger amount) {
        try {
            String contractAddress = loadAssetAddr();
            Test2 asset = Test2.load(contractAddress, web3j, credentials, new StaticGasProvider(gasPrice, gasLimit));
            TransactionReceipt receipt = asset.receivable_account(addr, amount, "0").send();
            List<AccountEventResponse> response = asset.getAccountEvents(receipt);
            if (!response.isEmpty()) {
                System.out.printf("completing transaction! %s give %s a receipt, amount is %s\n", response.get(0)._from, response.get(0)._to, response.get(0).amount.toString());
            } else {
                System.out.println("transaction error.\n");
            }
        } catch (Exception e) {
            System.out.println("error message:" + e.getMessage());
        }
    }
```

转让账单：

```
public void transfer(String assetAccount, BigInteger amount) {
        try {
            String contractAddress = loadAssetAddr();
            Test2 asset = Test2.load(contractAddress, web3j, credentials, new StaticGasProvider(gasPrice, gasLimit));
            asset.transfer(assetAccount, amount, "0").send();
            System.out.println("success");
        } catch (Exception e) {
            System.out.println("error message:" + e.getMessage());
        }
    }
```

申请融资：

```
public void financing(BigInteger amount) {
        try {
            String contractAddress = loadAssetAddr();
            Test2 asset = Test2.load(contractAddress, web3j, credentials, new StaticGasProvider(gasPrice, gasLimit));
            TransactionReceipt receipt = asset.financing(amount).send();
            System.out.println("financing error.\n");
            List<FinancingEventResponse> response = asset.getFinancingEvents(receipt);
            if (!response.isEmpty()) {
                System.out.printf("financing %s amount: %s\n", response.get(0)._from, response.get(0).amount.toString());
            }
            else {
                System.out.println("financing error.\n");
            }
        } catch (Exception e) {
            System.out.println("error message:" + e.getMessage());
        }
    }
```

还款：

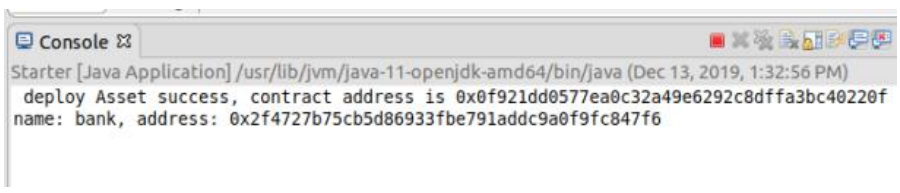
```
public void account_settlement() {
        try {
            String contractAddress = loadAssetAddr();
            Test2 asset = Test2.load(contractAddress, web3j, credentials, new StaticGasProvider(gasPrice, gasLimit));
            asset.account_settlement().send();
            System.out.println("success");
        } catch (Exception e) {
            System.out.println("error message:" + e.getMessage());
        }
    }
```

```
event Register(string name, uint amount);
    event Account(string _from, string _to, uint amount);
    event Transfer(string _from, string _to, string next_from, uint amount);
    event Financing(string _from, uint amount);
    event Settlement();
```

三、 功能测试与界面展示

登录界面

首先注册银行



然后注册三个公司


```
name: bank, address: 0x2f4727b75cb5d86933fbe791addc9a0f9fc847f6
create a account
name: car, address: 0x718811e2d1170db844d0c5de6d276b299f2916a9
create a account
name: wheel, address: 0xce1c349523c39e42769d690af28c34e2adc67dd4
create a account
name: lunguchang, address: 0x9cad951a57d174a8f0a425d5ee6920415165fdaa
```

签订单据，car 欠 wheel100，转让单据：car 欠 wheel50，car 欠 lg50

```
transaction: car to wheel, amount:100
transfer successful
receipt:0 thamount: 50
```

现在 lg 申请融资，融资成功


```
Financing successfully
amount: 10000012
```

Car 支付账单。


```
Successful settlement!Accomplished 1 receipt for wheel:
receipt:0 , amount: 100
 property: 1899900
```