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）	软工
学号	17343024	姓名	董轩宇
电话	13242893089	Email	407380756@qq.com
开始日期	12-10	完成日期	12-13

一、 项目背景

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，

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。，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——签发应收账款签发应收账款 交易上链交易上链。例如车企从轮胎公司购买一批轮胎并。例如车企从轮胎公司购买一批轮胎并签订应收账款单据。签订应收账款单据。

功能二：实现应收账款的转让上链上链，轮胎公司从轮毂公司购买一笔，轮胎公司从轮毂公司购买一笔轮毂，便将于车企的轮毂，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。期时归还钱款。

功能三：利用应收账款向银行融资上链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，供应链上所有可以利用应收账款单据向银行申请融资。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，应收账款单据到期时核心企业向下游企业支付相相应的欠应的欠款

二、 方案设计

由于能力有限和时间安排不得当，在规定时间内只完成了后端和链端的内容，前端没有完成

·后端

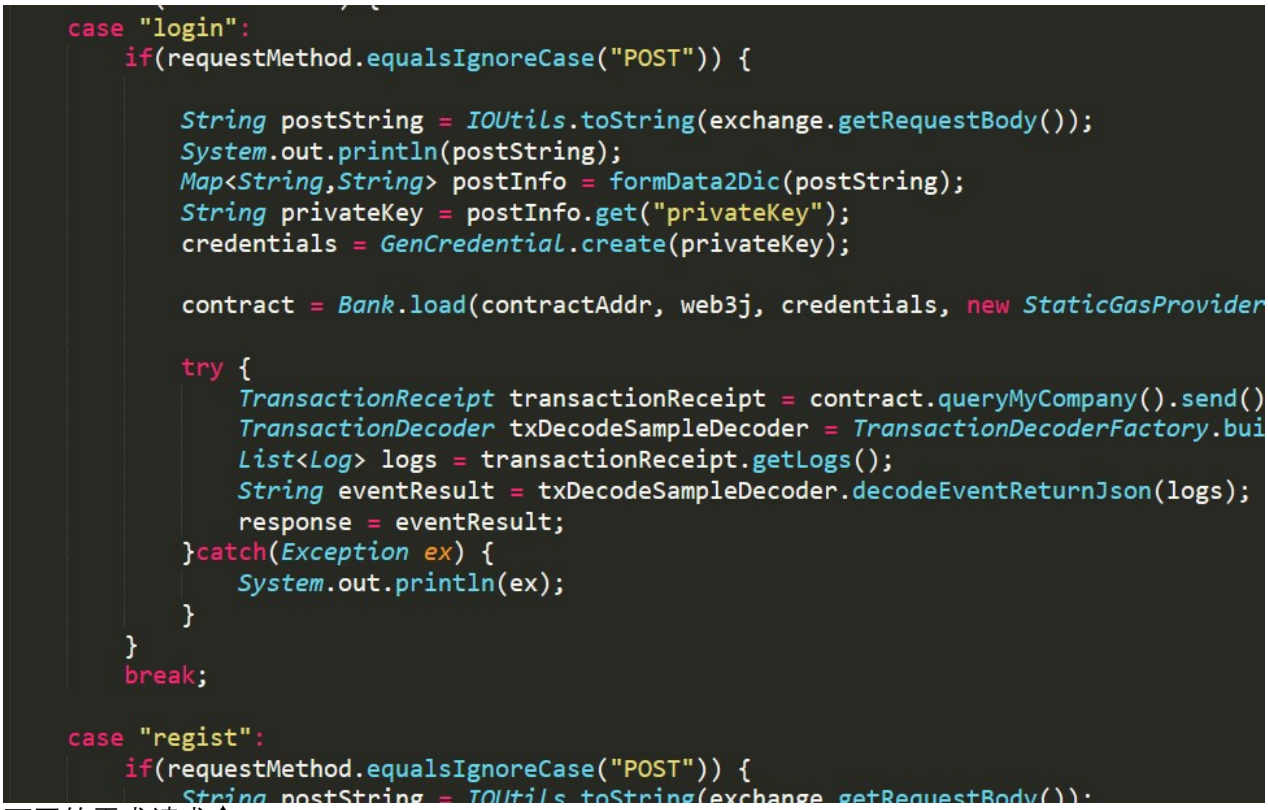
后端主要利用 SDK 进行交互

```
public static void init() throws Exception
{

    context = new ClassPathXmlApplicationContext("classpath:applicat

    service = context.getBean(Service.class);
    service.run();
    channelEthereumService = new ChannelEthereumService();
    channelEthereumService.setChannelService(service);
```

Sdk 初始化↑

```
396
397         HttpServer server = HttpServer.create(new InetSocketAddress(post), 0);
398         server.createContext("/api/login", new requestHandler("login"));
399         server.createContext("/api/regist", new requestHandler("regist"));
400         server.createContext("/api/bank/issue", new requestHandler("bank/issue"));
401         server.createContext("/api/bank/queryCompany", new requestHandler("bank/query
402         server.createContext("/api/company/send", new requestHandler("company/send"))
403         server.createContext("/api/company/showMyReceipt", new requestHandler("compan
404         server.createContext("/api/company/createReceipt", new requestHandler("compan
405         server.createContext("/api/company/transferReceipt", new requestHandler("comp
406         server.createContext("/api/company/bankFinancing", new requestHandler("compan
407         server.createContext("/api/company/payBack", new requestHandler("company/payB
408         server.start();
409         System.out.println("Server listening on localhost:"+post);
```

http 框架↑



```
case "login":
    if(requestMethod.equalsIgnoreCase("POST")) {

        String postString = IOUtils.toString(exchange.getRequestBody());
        System.out.println(postString);
        Map<String,String> postInfo = formData2Dic(postString);
        String privateKey = postInfo.get("privateKey");
        credentials = GenCredential.create(privateKey);

        contract = Bank.load(contractAddr, web3j, credentials, new StaticGasProvider

        try {
            TransactionReceipt transactionReceipt = contract.queryMyCompany().send()
            TransactionDecoder txDecodeSampleDecoder = TransactionDecoderFactory.bui
            List<Log> logs = transactionReceipt.getLogs();
            String eventResult = txDecodeSampleDecoder.decodeEventReturnJson(logs);
            response = eventResult;
        }catch(Exception ex) {
            System.out.println(ex);
        }
    }
    break;

case "regist":
    if(requestMethod.equalsIgnoreCase("POST")) {
        String postString = IOUtils.toString(exchange.getRequestBody());
```

不同的需求请求↑

·链段

链段部署按照官网上指示进行即可

合约：

login 部分：

```
/*登录*/
function login() public view returns(int){
    if (name2addr[addr2name[msg.sender]] != addres
        if (msg.sender == master){
            return 3;
        }
        return 1;
    }
    if (msg.sender == master){
        return 2;
    }
```

select 部分：

```
/*查询*/
function select(string memory account) public view returns(i

    Table table = openTable();

    Entries entries = table.select(account, table.newConditi
    uint256 asset_value = 0;
    if (0 == uint256(entries.size())) {
        return (-1, asset_value);
    } else {
        Entry entry = entries.get(0);
```

Register 部分：查询资产

```
/*信息 */
function register(string memory account, uint256 asset_value) p
    int256 ret_code = 0;
    int256 ret = 0;
    uint256 temp_asset_value = 0;

    (ret, temp_asset_value) = select(account);
    if(ret != 0) {
        name2addr[account] = msg.sender;
        addr2name[msg.sender] = account;
        accountList.push(account);
        Table table = openTable();

        Entry entry = table.newEntry();
        entry.set("account", account);
        if(msg.sender == master){
            entry.set("asset_value", int256(asset_value));
        }else{
            entry.set("asset_value", int256(0));
        }

        int count = table.insert(account, entry);
        if (count == 1) {

            ret_code = 0;
            if (msg.sender == master){
                ret_code = 1;
            }
```

Transfer 部分：收款的签发、转让

```
/*转移*/
function transfer(string memory to_account, uint256 amount) public returns

    string memory from_account = addr2name[msg.sender];
    int ret_code = 0;
    int256 ret = 0;
    uint256 from_asset_value = 0;
    uint256 to_asset_value = 0;


    (ret, from_asset_value) = select(from_account);
    if(ret != 0) {
        ret_code = -1;

        emit TransferEvent(ret_code, from_account, to_account, amount);
        return ret_code;

    }


    (ret, to_asset_value) = select(to_account);
    if(ret != 0) {
        ret_code = -2;

        emit TransferEvent(ret_code, from_account, to_account, amount);
        return ret_code;
    }

    if(from_asset_value < amount) {
        ret_code = -3;

        emit TransferEvent(ret_code, from_account, to_account, amount);
```

三、 心得体会

首先是因为个人能力有限和时间安排不当，在规定的时间内并没有完整的完成作业要求，只完成了后端和链段两个部分，并且大多数还是在与同学的讨论交流询问中才学会该如何完成，深感能力不足仍需进步，并为未能完成作业而道歉。

本次大作业最大的收获就是对于 fisco 的平台有了些许了解，fisco 的官网确实有着很丰富的资料，对于学习有着很大的帮助。对于项目的需求，更多部分主要的是对需求的理解，加以区块链知识的应用。但实现的更多方面，

主要是 solid 语言的实现。对于 solid 语言，虽然说是初学，但是加上以前的经验，还算是比较好掌握的；本次作业最大的问题是就是，环境的配置，平台的搭建等问题。并且由于缺乏数据库的相关知识，对于很多操作十分不熟练。

总体下来，感觉对于对于区块链的了解还是知之甚少。如果要对这方面加深学习研究，还需付出很大的努力。