Block Chain Project

TOC

Requirements

- 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。

Be detailed in 2020.docx.

初版实现方案

- Database:

主码	债权人	债务人	还款日期	挂起	金额
id	creditor	debtor	ddl	pending	value
string	address	address	uint256	address	uint256

  注意！：这个id是因为这个奇怪的数据库需要主码，但实际上无用，所以均设为一个全局变量（这个主码竟然是可以重复的

- 功能一：债权人发起，（债权人，债务人，还款日期）如有则直接修改value，否则插入一条新记录
- 功能二：债权人发起，减少记录金额，插入一条新记录金额为减少的金额（“插入”同样按照功能一的方式先做判断，看是修改还是插入）
- 功能三：债权人发起，向银行账户申请转移债券，设置pending为银行账户；银行可以取消pending（设为0）将债权人改为自己（“改”同样按照功能一的方式先做判断，看是修改哪个）。
- 功能四：债权人发起，（判断时间）删除一条记录

最终实现

1. TOC

2. 数据库
   - debt表

所有者(主码)	债权人	债务人	还款日期	金额
owner	creditor	debtor	ddl	value
string	string	string	int	int

其中保证(owner,creditor,debtor,ddl)唯一确定一条记录。
将owner和creditor不一致的欠条定义为挂起态，将一致的欠条定义为正常态。

   - account表

用户名(主码)	公司类型
id	type
address	int

其中保证(id)唯一确定一条记录。

3. 返回码
   - MORTGAGE_TO_DEBTOR：以欠条向银行抵押，其中欠条的债务人就是这个银行，从而引发错误。
   - NOT_BANK：当某操作对象必须是银行，但不是银行，引发错误。
   - REGISTERED：该地址已注册账户，引发错误。
   - ID_EXIST：注册时该用户名已存在，引发错误。
   - OVERFLOW：转移欠条时，指定金额超出欠条金额，引发错误。
   - NOT_EXIST：该欠条不存在，引发错误。
   - DB_ERR：数据库操作出错。
   - SUCC：成功。
4. 函数
   - register
     - 描述：公司账户注册
     - 公有：是
     - 参数：
       - id：公司账户名
       - type：公司类型
     - 返回值：

        - 返回码
    - select
        - 描述：查询所有者为自己的全部debt
        - 公有：是
        - 参数：无
        - 返回值：
            - DEBT数组
    - insert_core
        - 描述：添加一个欠条，如果存在相关欠条更新其value，否则插入一条新的欠条
        - 公有：否
        - 参数：
            - id：欠条的onwner
            - creditor：欠条的creditor
            - debtor：欠条的debtor
            - ddl：欠条的ddl
            - value：欠条的value
        - 返回值：
            - 返回码
    - insert
        - 描述：添加一个欠条，其中owner和creditor均为自己
        - 公有：是
        - 参数：
            - debtor：欠条的debtor
            - ddl：欠条的ddl
            - value：欠条的value
        - 返回值：
            - 返回码
    - is_bank
        - 描述：判断该公司账户是不是银行
        - 公有：否
        - 参数：
            - bank：判断的银行账户名
        - 返回值：
            - 返回码
    - mortgage
        - 描述：用部分或全部正常态的欠条向银行申请抵押，将owner设为银行，creditor仍设为自己，欠条由正常态转变为挂起态。
        - 公有：是
        - 参数：
            - bank：银行账户名
            - debtor：欠条的debtor
            - ddl：欠条的ddl
            - value：需要抵押的value
        - 返回值：

        - 返回码
    - permit
        - 描述：银行处理指定抵押申请，若同意将creditor设为银行，若拒绝将owner设为申请者，欠条由挂起态转变为正常态。
        - 公有：是
        - 参数：
            - boolean：0为拒绝，非0为同意
            - creditor：欠条的creaditor
            - debtor：欠条的debtor
            - ddl：欠条的ddl
        - 返回值：
            - 返回码
    - assign
        - 描述：转移部分或全部正常态的欠条给别人，如果目标是欠条的debtor视为还款，如果目标不是欠条的debtor视为交易欠条
        - 公有：是
        - 参数：
            - creditor：转移目标
            - debtor：欠条的debtor
            - ddl：欠条的ddl
            - value：需要转移的value
        - 返回值
            - 返回码

测试及结果

- 创建账户

  在链上部署Debt合约后，首先利用拥有的账户地址进行注册。注册情况如下表所示：

	地址	id	类型
账户1	0x1f9ef5400aca6856fa0da557707b5066c7e80c2a	cmp3	OTHER
账户2	0x3ffa7b39dbc33deb632f8f93f5456431ea8f3b9f	cmp2	OTHER
账户3	0x5a7f68319d872c697fce19bb327f42c272f2db80	cmp1	OTHER
账户4	0xe1e2188ae94eb88e1295cd6fd11f2a1c0b582693	bank1	BANK

  注册过程调用的函数为register( )，以账户1的注册为例展示注册的具体过程如下图：

  注册过程还设置了保护机制，即无法用同一地址注册两个账户，保证了账户和地址一一对应的关系：

- 创建账单（功能一）
- 注册完成后，开始创建各个账户之间的账单。完成创建后账单的具体情况如下：

	owner	creditor	debtor	ddl	value
账单1	cmp2	cmp2	cmp1	500	500
账单2	cmp1	cmp1	cmp3	500	350
账单3	bank1	bank1	cmp3	500	400
账单4	bank1	bank1	cmp1	500	200

创建账单调用的函数为insert( )，以账单1的创建为例展示创建过程如下：

且该账单可以被拥有者（即cmp2）通过select( )函数查询：

该过程实现了要求中的功能一，即实现采购商品—签发应收账款交易上链。交易双方在完成商品交易后通过insert( )函数创建账单并存储在数据库中（上链）。

- 抵债和还款（功能二，功能四）

  这时cmp1试图通过将cmp3的欠款用于抵消其欠cmp2的款项：

  但由于数额超限（cmp3欠款为350，无法用于抵消500的欠款）而无法执行。

  调整抵消金额为200后成功执行：

  查询cmp1的账单，发现金额改变（由350变为150）：

  且在bank1的账单中，cmp3的欠款数额增加（由400变为600）：

  该过程实现了要求中的功能二：实现应收账款的转让上链。账户A可用账户B对其的欠款来偿还自己对另一账户C的欠款，从而将将债务关系转移到B与C之间。

  同时assign( )函数亦能够实现还款的功能（即功能四），只需把传入参数中的“creditor”和“debtor”设为相同的值（且必须是当前账户的债务人），则可实现还款，如下图：

  还款后，当前账户中的相应账单被清除：

- 贷款（功能三）

  这时cmp2试图将与cmp1的账单用于向bank1抵押贷款：

  但由于数额超限（cmp1欠款为500，无法用于贷款1000）而无法执行。

  修改贷款金额为200后成功：

  执行成功，且cmp2的账单金额发生变化：

  bank1通过permit( )函数接收贷款申请：

  bank1获得了cmp1的账单：

  该过程实现了要求中的功能三：利用应收账款向银行融资上链。账户A可用账户B对其的欠款来向银行贷款，从而将将债务关系转移到B与银行之间。

分工

- [x] 实现方案：谷正阳，陈振宇
- [x] Database：谷正阳
- [x] 功能一：陈嘉宁
- [x] 功能二：陈振宇
- [x] 功能三：陈嘉宁
- [x] 功能四：陈振宇

- [x] Debug及代码修改：谷正阳
- [x] 文档：谷正阳
- [x] 测试：陈嘉宁