pragma solidity ^0.4.19;

import "./Manager.sol";

import "./Distrubuter.sol";

contract Escrow is main {

address private escrow=0xcddb5e49b709b2e38171e36263c656e1c3bcf047;

uint private start;

function checkOrder(uint orderId) public {

// require(BalanceOfMoney[itemMap[orderId].distrubuter]>itemMap[orderId].totalCost);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]+itemMap[orderId].totalCost;

BalanceOfMoney[itemMap[orderId].distrubuter]=BalanceOfMoney[itemMap[orderId].distrubuter]-itemMap[orderId].totalCost;

bankConfirmation[orderId]= true;

}

function payBalance(uint orderId) public {

require(keccak256(statsMap[orderId].checkPoint)== keccak256("OrderReceivedByDistrubuter"));

uint i;

uint remaining;

bool complete=false;

if(flowOfObject[orderId].Addresses[0]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[0]]=BalanceOfMoney[flowOfObject[orderId].Addresses[0]]+(commisionAmount[flowOfObject[orderId].Addresses[0]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[0]]\*itemMap[orderId].totalCost/100);

}

if(flowOfObject[orderId].Addresses[1]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[1]]=BalanceOfMoney[flowOfObject[orderId].Addresses[1]]+(commisionAmount[flowOfObject[orderId].Addresses[1]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[1]]\*itemMap[orderId].totalCost/100);

}

if(flowOfObject[orderId].Addresses[2]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[2]]=BalanceOfMoney[flowOfObject[orderId].Addresses[2]]+(commisionAmount[flowOfObject[orderId].Addresses[2]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[2]]\*itemMap[orderId].totalCost/100);

}

if(flowOfObject[orderId].Addresses[3]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[3]]=BalanceOfMoney[flowOfObject[orderId].Addresses[3]]+(commisionAmount[flowOfObject[orderId].Addresses[3]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[3]]\*itemMap[orderId].totalCost/100);

}

if(flowOfObject[orderId].Addresses[4]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[4]]=BalanceOfMoney[flowOfObject[orderId].Addresses[4]]+(commisionAmount[flowOfObject[orderId].Addresses[4]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[4]]\*itemMap[orderId].totalCost/100);

}

if(flowOfObject[orderId].Addresses[5]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[5]]=BalanceOfMoney[flowOfObject[orderId].Addresses[5]]+(commisionAmount[flowOfObject[orderId].Addresses[5]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[5]]\*itemMap[orderId].totalCost/100);

}

if(flowOfObject[orderId].Addresses[6]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[6]]=BalanceOfMoney[flowOfObject[orderId].Addresses[6]]+(commisionAmount[flowOfObject[orderId].Addresses[6]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[6]]\*itemMap[orderId].totalCost/100);

}

if(flowOfObject[orderId].Addresses[7]!=0x0000000000000000000000000000000000000000){

BalanceOfMoney[flowOfObject[orderId].Addresses[7]]=BalanceOfMoney[flowOfObject[orderId].Addresses[7]]+(commisionAmount[flowOfObject[orderId].Addresses[7]]\*itemMap[orderId].totalCost/100);

BalanceOfMoney[escrow]=BalanceOfMoney[escrow]-(commisionAmount[flowOfObject[orderId].Addresses[7]]\*itemMap[orderId].totalCost/100);

}

remaining=BalanceOfMoney[escrow];

BalanceOfMoney[itemMap[orderId].manufacturer]=BalanceOfMoney[itemMap[orderId].manufacturer]+remaining;

BalanceOfMoney[escrow]=0;

}

function checkBalance(address accountNumber)constant public returns(uint) {

return(BalanceOfMoney[accountNumber]);

}

}