

Instructions KYCContract.sol:

Testing KYCcontract functions on Remix:

1) Contract compiled successfully:

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'SOLIDITY COMPILER' panel shows the compiler version as '0.5.9+commit.c68bc34e' and the language set to 'Solidity'. The 'EVM VERSION' is set to 'compiler default'. The 'COMPILER CONFIGURATION' section has 'Auto compile' checked and 'Enable optimization' set to '200'. A blue button labeled 'Compile KYCContract.sol' is visible. Below the compiler panel, the 'CONTRACT' section shows 'KYCContract (KYCContract.sol)' selected, with buttons for 'Publish on Ipfs', 'Compilation Details', 'ABI', and 'Bytecode'.

The central code editor displays the Solidity code for 'KYCContract.sol'. The code includes a pragma statement for Solidity 0.5.9, a contract definition for 'KYCContract' using 'SafeMath', and several structs: 'Customer' (with fields for username, data, kycStatus, upVotes, downVotes, and bank), 'Bank' (with fields for bankName, ethAddress, complaints, kycCount, isAllowedToVote, and regNumber), and 'Request'.

The bottom panel shows transaction logs. The first log is a successful transaction: '[vm] from: 0xCA3...a733c to: KYC.upVoteCustomer(string) 0x332...D486D value: 0 wei data: 0x569...00000 logs: 0 hash: 0x0de...da204'. Below this, a call to 'KYC.viewCustomer' is shown. The second log is a successful call: '[call] from: 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c to: KYC.viewCustomer(string) data: 0x107...00000'. Each log has a 'Debug' button next to it.

2) Deploy Contract on Java script VM:

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is active, showing the 'ENVIRONMENT' set to 'JavaScript VM (London)', the 'ACCOUNT' as '0x5B3...eddC4 (99.999999%)', and the 'GAS LIMIT' as '3000000'. The 'VALUE' is set to '0' in 'wei'. The 'CONTRACT' is 'KYCContract - contracts/KYCContract.sol'. A 'Deploy' button is visible, along with options to 'Publish to IPFS' or 'At Address'. Below, it shows 'Transactions recorded: 1' and 'Deployed Contracts'.

The main editor shows the Solidity code for 'KYCContract.sol'. The code includes mappings for customers, banks, and requests, a constructor function, and a comment indicating the contract is deployed on the JavaScript VM. The constructor function is highlighted in blue.

```
41 // mapping (list) of customers
42 mapping(string => Customer) customers;
43
44 //mapping (list) of banks
45 mapping(address => Bank) banks;
46
47 //mapping of KYC requests
48 mapping(string => Request) requests;
49
50
51 /**
52  * constructor function
53  * - Define Admin user
54  * - reset number of banks
55  */
56
57 constructor() public {
58
59     //when the contract is deployed, set the account address as the admin user;
60     admin = msg.sender;
61     noOfBanks = 0;
62 }
63
64
65 /** Admin functions */
66
67
68
```

Below the code editor, the 'ContractDefinition KYCContract' is shown with '0 reference(s)'. A search bar is present with the text 'Search with transaction hash or address'.

The transaction details are displayed below the search bar:

status	transaction hash	from	to	gas	transaction cost	execution cost	hash	input	decoded input
true Transaction mined and execution succeed	0xfd53b68fb12956b7720ace9caa8a048d808ca2c0dbedaad98b49abfe599cd6c6	0x5B380a6a701c568545dCfcB03FcB875f56beddc4	KYCContract.(constructor)	8000000 gas	8000000 gas	4227254 gas	0xfd53b68fb12956b7720ace9caa8a048d808ca2c0dbedaad98b49abfe599cd6c6	0x608...90032	{}

Used test address provided by the remix environment for testing:

admin: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

Bank Name:	Address	- regNo.
bank1:	0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2	- reg1
bank2:	0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c	- reg2
bank3:	0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db	- reg3
bank4:	0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB	- reg4
bank5:	0x617F2E2fD72FD9D5503197092aC168c91465E7f2	- reg5
bank6:	0x17F6AD8Ef982297579C203069C1DbfFE4348c372	- reg6

1) Admin Functionality

-- only admin can add Bank or remove bank or update voting rights

Activities Google Chrome Sep 7 00:25

Remix - Ethereum IDE

remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.9+commit.c68bc34e.js

DEPLOY & RUN TRANSACTIONS

VALUE: 0 wei

CONTRACT: KYCContract - contracts/KYCContract.sol

Deploy

☐ Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 2

Deployed Contracts

- KYCContract AT 0xD7C...FAC5F (MEI)

addBank "bank1","0xAb84B3F64d9Ct"

addCustomer string _userName, string _cur

downVoteCust... string _userName

modifyCustomer string _userName, string _nei

modifyisAllow... address _ethAddress, bool _li

raiseKYC string _userName, string _cur

```
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
```

```

/* Admin functions */
/**
 * Add Bank - This function is used by the admin to add a bank to the KYC Contract.
 */
modifier isAdmin(address _sender) {
    require(_sender == admin, "Not authorised for admin transactions");
}
function addBank(string memory _bankName, address _ethAddress, string memory _regNumber)
    public
    isAdmin(msg.sender) {
    //validate bank does not exist for same address
    require(banks[_ethAddress].ethAddress == address(0), "Another bank exists with same address");
    banks[_ethAddress].bankName = _bankName;
    banks[_ethAddress].ethAddress = _ethAddress;
    banks[_ethAddress].complaints = 0;
    banks[_ethAddress].kycCount = 0;
    banks[_ethAddress].isAllowedToVote = true;
    banks[_ethAddress].regNumber = _regNumber;
}

```

ContractDefinition KYCContract 0 reference(s)

0 listen on network Search with transaction hash or address

transact to KYCContract.addBank pending ...

[vm] from: 0xAb8...35cb2 to: KYCContract.addBank(string,address,string) 0xd7c...faC5F value: 0 wei data: 0xccf...00000 logs: 0 hash: 0x66b...64987

transact to KYCContract.addBank errored: VM error: revert.

revert

The transaction has been reverted to the initial state.
Reason provided by the contract: "Not authorised for admin transactions".
Debug the transaction to get more information.

Debug

Activities Google Chrome Sep 7 00:26

Remix - Ethereum IDE

remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.9+commit.c68bc34e.js

DEPLOY & RUN TRANSACTIONS

Deployed Contracts

KYCContract AT 0XD7C...FAC5F (MEI)

- addBank "bank1","0xab8483f64d9c...
- addCustomer string _userName, string _cu...
- downVoteCust... string _userName
- modifyCustomer string _userName, string _nei...
- modifyisAllow... address _ethAddress, bool _li...
- raiseKYC string _userName, string _cu...
- removeBank "0xab8483f64d9c6d1ecf9...
- removeKYC string _userName
- reportBank address _bank
- upVoteCustom... string _userName
- getBankCompl... address _ethAddress
- noOfBanks

```
59 //when the contract is deployed, set the account address as the admin user;
60 admin = msg.sender;
61 noOfBanks = 0;
62 }
63
64 /*****
65 * Admin functions *****/
66 */
67
68 /**
69 * Add Bank - This function is used by the admin to add a bank to the KYC Contract.
70 *
71 */
72 modifier isAdmin(address _sender) {
73     require(_sender == admin, "Not authorised for admin transactions");
74 }
75
76
77 function addBank(string memory _bankName, address _ethAddress, string memory _regNumber)
78     public
79     isAdmin(msg.sender) {
80
81     //validate bank does not exists for same address
82     require(banks[_ethAddress].ethAddress == address(0), "Another bank exists with same address");
83
84     banks[_ethAddress].bankName = _bankName;
85     banks[_ethAddress].ethAddress = _ethAddress;
86     banks[_ethAddress].complaints = 0;
```

ContractDefinition KYCContract 0 reference(s)

0 listen on network Search with transaction hash or address

transact to KYCContract.removeBank pending ...

[vm] from: 0xab8...35cb2 to: KYCContract.removeBank(address) 0xd7c...fac5f value: 0 wei data: 0x964...35cb2 logs: 0 hash: 0xee7...da13c **Debug**

transact to KYCContract.removeBank errored: VM error: revert.

revert

The transaction has been reverted to the initial state.
Reason provided by the contract: "Not authorised for admin transactions".
Debug the transaction to get more information.

Activities

Google Chrome

Sep 7 00:26

Remix - Ethereum IDE

remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.9+commit.c68bc34e.js

DEPLOY & RUN TRANSACTIONS

Deployed Contracts

KYCCONTRACT AT 0xD7C...FAC5F (MEI)

addBank

"bank1","0xab8483f64d9c6d1ecf99"

addCustomer

string _userName, string _cust...

downVoteCust...

string _userName

modifyCustomer

string _userName, string _nei...

modifyisAllow...

"0xab8483f64d9c6d1ecf99"

raiseKYC

string _userName, string _cust...

removeBank

"0xab8483f64d9c6d1ecf99"

removeKYC

string _userName

reportBank

address _bank

upVoteCustom...

string _userName

getBankCompl...

address _ethAddress

noOfBanks

Home

kyc.sol

KYCCONTRACT.sol

3 tabs

```
59 //when the contract is deployed, set the account address as the admin user;
60 admin = msg.sender;
61 noOfBanks = 0;
62 }
63
64 //*****
65 * Admin functions *****
66 */
67
68 /**
69 * Add Bank - This function is used by the admin to add a bank to the KYC Contract.
70 *
71 */
72 modifier isAdmin(address _sender) {
73     require(_sender == admin,"Not authorised for admin transactions");
74 }
75
76
77 function addBank(string memory _bankName,address _ethAddress,string memory _regNumber)
78     public
79     isAdmin(msg.sender){
80
81     //validate bank does not exists for same address
82     require(banks[_ethAddress].ethAddress == address(0),"Another bank exists with same address");
83
84     banks[_ethAddress].bankName = _bankName;
85     banks[_ethAddress].ethAddress = _ethAddress;
86     banks[_ethAddress].complaints = 0;
```

ContractDefinition KYCContract

0 reference(s)

0

listen on network

Search with transaction hash or address

transact to KYCContract.modifyisAllowedToVote pending ...

[vm] from: 0xab8...35cb2 to: KYCContract.modifyisAllowedToVote(address,bool) 0xd7c...fac5f value: 0 wei data: 0x8a8...00000 logs: 0

hash: 0x985...184ba

transact to KYCContract.modifyisAllowedToVote errored: VM error: revert.

revert

The transaction has been reverted to the initial state.

Reason provided by the contract: "Not authorised for admin transactions".

Debug the transaction to get more information.

>

Debug

Admin is allowed to create new Bank

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel lists various functions for the KYCContract, including 'addBank', 'addCustomer', 'downVoteCust...', 'modifyCustomer', 'modifyisAllow...', 'raiseKYC', 'removeBank', 'removeKYC', 'reportBank', 'upVoteCustom...', 'getBankCompl...', 'noOfBanks', and 'viewBankDeta...'. The 'addBank' function is selected, showing its parameters: 'bank1' (string), '0xab8483f64d9c6d1ecf99' (address), and '0' (uint8).

The main editor shows the Solidity code for the KYCContract. The 'addBank' function is defined as follows:

```
71 modifier isAdmin(address _sender) {
72     require(_sender == admin, "Not authorised for admin transactions");
73 }
74
75
76
77 function addBank(string memory _bankName, address _ethAddress, uint8 _regNumber)
78     public
79     isAdmin(msg.sender) {
80
81     //validate bank does not exists for same address
82     require(banks[_ethAddress].ethAddress == address(0), "Another bank exists with same address");
83
84     banks[_ethAddress].bankName = _bankName;
85     banks[_ethAddress].ethAddress = _ethAddress;
86     banks[_ethAddress].complaints = 0;
87     banks[_ethAddress].kycCount = 0;
88     banks[_ethAddress].isAllowedToVote = true;
89     banks[_ethAddress].regNumber = _regNumber;
90
91     noOfBanks = noOfBanks.add(1);
92 }
93
94
95 /**
96  * Modify Bank isAllowedToVote - This function can only be used by the admin to change the status of isAllowedToVote of any of the banks at any point in time.
97  *
98  */
```

The bottom panel shows the transaction details for the 'addBank' function. The transaction was successful, with a status of 'true Transaction mined and execution succeed'. The transaction hash is '0x7f99fb9376921f1620bb04250da45c604356ac801378e769aba62d70f9eb103'. The transaction was sent from '0x5B38Da6a701c568545dcfcb875f56beddc4' to 'KYCContract.addBank(string,address,string) 0xd7Ca4e99f7c17189ea20e80d3363c47009afac5F'. The transaction cost was 8000000 gas, and the execution cost was 142266 gas.

View Bank Details: (no of banks made public for testing only)

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is active, showing a list of functions. The 'viewBankDetails...' function is selected, and its parameters are displayed: 0: string: _bankName bank1, 1: address: _ethAddress1 0xab8483f64d9c6d1ecf9b849ae677dD3315835cb2, 2: uint256: _complaints 0, 3: uint256: _kycCount 0, 4: bool: _isAllowedToVote true, and 5: string: _regNumber reg1. Below this, the 'viewCustomer' function is also visible. The main editor shows the Solidity code for the KYCContract, with the 'addBank' function highlighted. The bottom panel shows the transaction details for the 'addBank' function, indicating a successful execution. The transaction hash is 0x7f99fb9376921f1620bb04250da45c604356ac801378e769aba62d70f9eb103. The transaction cost is 8000000 gas, and the execution cost is 142266 gas.

ContractDefinition KYCContract 0 reference(s)

[vm] from: 0x5B3...eddC4 to: KYCContract.addBank(string,address,string) 0xd7C...faC5F value: 0 wei data: 0xccf...00000 logs: 0
hash: 0x7f9...eb103

status true Transaction mined and execution succeed

transaction hash 0x7f99fb9376921f1620bb04250da45c604356ac801378e769aba62d70f9eb103

from 0x5B38Da6a701c568545dcfcb83fcb875f56beddc4

to KYCContract.addBank(string,address,string) 0xd7Ca4e99F7C171B9ea20e80d3363c47009afaC5F

gas 8000000 gas

transaction cost 8000000 gas

execution cost 142266 gas

Added total 6 banks.

- noOfBanks counter made public for testing
- Logic for $\frac{1}{3}$ rd banks only possible if number of banks > 5

The screenshot shows a dark-themed web application interface. At the top, there is a button labeled 'noOfBanks'. Below it, the text '0: uint256: 6' is displayed. Below this, there is a dropdown menu with the label 'viewBankData...' and a selected value '"0x17F6AD8Ef982297579C"'. Below the dropdown, there is a list of six items, each with an index and a description:

- 0: string: _bankName bank6
- 1: address: _ethAddress1 0x17F6AD8Ef982297579C203069C1DbfFE4348c372
- 2: uint256: _complaints 0
- 3: uint256: _kycCount 0
- 4: bool: _isAllowedToVote true
- 5: string: _regNumber reg6

Add customers:

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is active, showing the 'KYCContract' file. Below the file list, there are buttons for 'addBank', 'addCustomer', 'downVoteCust...', 'modifyCustomer', 'modifyisAllow...', 'raiseKYC', 'removeBank', and 'removeKYC'. The 'addCustomer' button is highlighted. The main editor area shows the Solidity code for the KYCContract, with the 'addCustomer' function highlighted. The function code is as follows:

```
141 * bank functions *****
142 */
143
144 /**
145  * Create add a customer to the customer list.
146  * - Validate Bank Exists
147  */
148
149
150 function addCustomer(string memory _userName, string memory _customerData)
151 public
152 bankExists(msg.sender) {
153
154 //validate if customer is present in the database
155 require(customers[_userName].bank == address(0), "Customer is already present, please call modifyCustomer to edit the customer data");
156
157 customers[_userName].userName = _userName;
158 customers[_userName].data = _customerData;
159 customers[_userName].bank = msg.sender;
160
161 //set KYC status = false for new customer & reset upvotes and downvotes
162 customers[_userName].kycStatus = false;
163 customers[_userName].upVotes = 0;
164 customers[_userName].downVotes = 0;
165
166 }
167
168 /**
```

Below the code editor, the 'ContractDefinition' section shows the 'KYCContract' definition. The 'Run' button is visible. The bottom panel shows the transaction details for the 'addCustomer' function. The transaction was successful, with the following details:

Field	Value
status	true Transaction mined and execution succeed
transaction hash	0x3ef60ac7cfc4d9357dd6d1570fd445ab89e6d27307d3c04a416c03a581339f45
from	0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2
to	KYCContract.addCustomer(string,string) 0xd7Ca4e99F7C171B9ea2De80d3363c47009afaC5F
gas	8000000 gas
transaction cost	8000000 gas
execution cost	101279 gas

Add KYC request:

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is active, showing a list of transactions. The 'raiseKYC' transaction is highlighted, with parameters: 'sanal', 'adhar1hash'. The main editor shows the Solidity code for the KYC contract, with line 251 highlighted. The bottom panel shows the transaction details for the 'raiseKYC' function call, indicating a successful execution.

DEPLOY & RUN TRANSACTIONS

Transactions recorded: 12

Deployed Contracts: KYCContract at 0xD7C...FAC5F (MEI)

addBank: "bank6", "0x17F6AD8E9822"

addCustomer: "sanal", "adhar1hash"

downVoteCust...: string _userName

modifyCustomer: string _userName, string _nei

modifyisAllow...: "0xAb8483F64d9C6d1EcF9i"

raiseKYC: "sanal", "adhar1hash"

removeBank: "0xAb8483F64d9C6d1EcF9i"

removeKYC: string _userName

reportBank: address _bank

upVoteCustom...: string _userName

getBankCompl...: address _ethAddress

Contract Definition: KYCContract

```
231
232
233 function raiseKYC(string memory _userName, string memory _customerDataHash)
234 public
235 bankExists(msg.sender)
236 customerExists(_userName) {
237
238
239
240 //update if KYC already exists (update data for same customer and bank )
241 if (requests[_userName].bank == msg.sender) {
242
243 //validate if customer data has changed
244 require(keccak256(abi.encodePacked(requests[_userName].data)) != keccak256(abi.encodePacked(_customerDataHash)), "Request already exists");
245
246 //if there is a change in data hash, update KYC request
247 requests[_userName].data = _customerDataHash;
248
249 } else {
250
251 //create new KYC request for the customer
252 requests[_userName].userName = _userName;
253 requests[_userName].bank = msg.sender;
254 requests[_userName].data = _customerDataHash;
255
256 }
257
258 /**
259 * Remove Request - This function allows a bank to remove KYC request of a customer.
260 */
261 }
```

Transaction Details:

- status: true Transaction mined and execution succeed
- transaction hash: 0xab6d858c7b536493d7b37b47d7aded68e28003826526d572bb2fe07c69ebcec2
- from: 0xAb8483F64d9C6d1EcF9b849Ae677d03315835cb2
- to: KYCContract.raiseKYC(string,string) 0xd7Ca4e99F7C171B9ea2De80d3363c47009afaC5F
- gas: 8000000 gas
- transaction cost: 8000000 gas
- execution cost: 96157 gas

Update Customer:

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar shows a list of transactions, including 'modifyCustomer' with parameters 'sanal' and 'adhar1hash'. The main editor shows the Solidity code for 'KYCContract.sol', with the 'modifyCustomer' function highlighted. The function code is as follows:

```
171 *  
172 */  
173  
174 function modifyCustomer(string memory _userName, string memory _newcustomerData)  
175 public  
176 bankExists(msg.sender)  
177 customerExists(_userName) {  
178  
179 //validate if data has changed  
180 require(keccak256(abi.encodePacked(customers[_userName].data)) != keccak256(abi.encodePacked(_newcustomerData)), "Customer Data has not changed");  
181  
182 //Update Customer Data  
183 customers[_userName].data = _newcustomerData;  
184  
185 //set KYC status = false when customer data is updated and reset upvotes and downVotes  
186 customers[_userName].kycStatus = false;  
187 customers[_userName].upVotes = 0;  
188 customers[_userName].downVotes = 0;  
189  
190  
191 modifier customerExists(string memory _userName) {  
192 //validate if customer is present in the database  
193 require(customers[_userName].bank != address(0), "could not find the customer in the database");  
194 _;  
195 }  
196  
197  
198 /**
```

Below the code editor, the transaction details are shown, indicating a successful execution:

```
[vm] from: 0xAb8...35cb2 to: KYCContract.modifyCustomer(string,string) 0xd7C...faC5F value: 0 wei data: 0xf72...00000 logs: 0  
hash: 0xddb...27fb2  
  
status true Transaction mined and execution succeed  
transaction hash 0xddb93b87d8917684b4395360a7aaa6541aafad797cafe87811e1e4559e27fb2  
from 0xAb8483f64d9c6d1EcF9b849Ae677d03315835cb2  
to KYCContract.modifyCustomer(string,string) 0xd7Ca4e99f7C171B9ea2De88d3363c47009afaC5F  
gas 8000000 gas  
transaction cost 8000000 gas  
execution cost 44364 gas
```

Banks upvote customer: if upvote > downvote -> KYC Status = true

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows several functions: 'reportBank', 'upVoteCustom...', 'getBankCompl...', 'noOfBanks', 'viewBankData...', and 'viewCustomer'. The 'viewCustomer' function is selected, showing its parameters and return values. The main editor displays the 'KYCContract.sol' file, with the 'upVoteCustomer' function highlighted. The function logic includes checking if the customer's data hash matches the KYC request, updating the customer's upvotes, and setting the KYC status to true if the number of upvotes is greater than the number of downvotes. The bottom panel shows the execution of the 'upVoteCustomer' function, with a status of 'true Transaction mined and execution succeed'. The transaction details, including the hash, from address, to address, gas, and costs, are displayed.

```
282 modifier bankIsAllowedToVote(address _bank){
283     require(banks[_bank].IsAllowedToVote == true,"Bank is not allowed to Vote");
284     _;
285 }
286
287 function upVoteCustomer(string memory _userName)
288     public
289     bankExists(msg.sender)
290     bankIsAllowedToVote(msg.sender) {
291     // check customer data hash against the data hash in the KYC request
292     require(keccak256(abi.encodePacked(customers[_userName].data)) == keccak256(abi.encodePacked(requests[_userName].data)),"Data is not same as KYC request");
293
294     // upvote customer
295     customers[_userName].upVotes += 1;
296
297     //if one-third of the total number of banks downvote the customer, then the KYC status is set to false
298
299     if((noOfBanks > 5) && (customers[_userName].downVotes >= (noOfBanks / 3))) {
300         customers[_userName].kycStatus = false;
301     } else {
302         //For the KYC status to be true for any customer, the number of upvotes should be greater than the number of downvotes
303         if(customers[_userName].upVotes > customers[_userName].downVotes) {
304             customers[_userName].kycStatus = true;
305         }
306     }
307 }
308
309
```

ContractDefinition KYCContract 0 reference(s)

[vm] from: 0xAb8...35cb2 to: KYCContract.upVoteCustomer(string) 0xd7C...faC5F value: 0 wei data: 0x569...00000 logs: 0
hash: 0x2bb...e3390

status true Transaction mined and execution succeed

transaction hash 0x2bb53e5243fd9a37bf48b14b56ec8298fd33265819c03dfdb29c6c08f2e3390

from 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

to KYCContract.upVoteCustomer(string) 0xd7Ca4e99F7C171B9ea2De80d3363c47009afaC5F

gas 80000000 gas

transaction cost 80000000 gas

execution cost 82711 gas

hash 0x2bb53e5243fd9a37bf48b14b56ec8298fd33265819c03dfdb29c6c08f2e3390

If more than $\frac{1}{3}$ rd of banks downvote, customer KYC is false:

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows the contract's state variables and their values:

- `noOfBanks`: 0: uint256: 6
- `viewBankData...`: "0x17F6AD8EF982297579C"
- `0: string: _bankName bank6`
- `1: address: _ethAddress1 0x17F6AD8EF982297579C203069C1DbfFE4348c372`
- `2: uint256: _complaints 0`
- `3: uint256: _kycCount 0`
- `4: bool: _isAllowedToVote true`
- `5: string: _regNumber reg6`

The 'viewCustomer' function is selected, showing its arguments:

- `0: string: sanal`
- `1: string: adhar1hash`
- `2: bool: false`
- `3: uint256: 1`
- `4: uint256: 3`
- `5: address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2`

The 'Low level interactions' panel shows the transaction details:

- `CALLDATA`: [Empty]
- `Transact` button

The main editor shows the Solidity code for the `KYCCContract.sol` contract. The `downVoteCustomer` function is highlighted, showing the logic for updating the customer's KYC status based on the number of downvotes.

```
314 * - validate customer is valid
315 * - validate bank is valid
316 * - todo - avoid duplicate downvote or upvote by a bank
317 */
318
319 function downVoteCustomer(string memory _userName)
320 public
321 bankExists(msg.sender)
322 bankIsAllowedToVote(msg.sender) {
323
324     // check customer data hash against the data hash in the KYC request
325     require(keccak256(abi.encodePacked(customers[_userName].data)) == keccak256(abi.encodePacked(requests[_userName].data)), "Data is not same as KYC request");
326
327     // downvote the customer
328     customers[_userName].downVotes += 1;
329
330     //if one-third of the total number of banks downvote the customer, then the KYC status is set to false
331
332     if((noOfBanks > 5) && (customers[_userName].downVotes >= (noOfBanks / 3))) {
333         customers[_userName].kycStatus = false;
334     } else {
335         //For the KYC status to be true for any customer, the number of upvotes should be greater than the number of downvotes
336
337         if(customers[_userName].upVotes > customers[_userName].downVotes) {
338             customers[_userName].kycStatus = true;
339         }
340     }
341 }
```

The bottom panel shows the execution results:

- `transact to KYCCContract.downVoteCustomer pending ...`
- `[vm] from: 0xAb8...35cb2 to: KYCCContract.downVoteCustomer(string) 0xd7C...faC5F value: 0 wei data: 0x82e...00000 logs: 0`
- `hash: 0x2bf...01273`
- `call to KYCCContract.viewCustomer`
- `[call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: KYCCContract.viewCustomer(string) data: 0x187...00000`
- `transaction hash`: 0xc83276f6e114ee9f79cc39bdcbe3a346a7b04bd6861eb5e00b0cd55fa148ab3
- `from`: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

If upvotes is greater than downvotes, KYC is valid:

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows a contract named 'KYCContract.sol' with several variables and functions. The 'viewBankData...' function is selected, showing its parameters: a string for bank name, an address for bank address, a uint256 for complaints, a uint256 for KYC count, a bool for KYC status, and a string for registration number. The 'viewCustomer' function is also visible, showing its parameters: a string for customer name, a string for address hash, a bool for KYC status, a uint256 for KYC count, and a string for registration number. The 'Low level interactions' section shows the 'CALLDATA' field and a 'Transact' button.

The main editor displays the Solidity code for 'KYCContract.sol'. The code includes a modifier 'bankIsAllowedToVote' and a function 'upVoteCustomer'. The 'upVoteCustomer' function checks if the customer's KYC status is valid based on the number of upvotes and downvotes. If the number of upvotes is greater than the number of downvotes, the KYC status is set to true.

The bottom panel shows the execution results. It includes a search bar for transaction hash or address. The results show a transaction to 'KYCContract.upVoteCustomer' with a value of 0 wei and data '0x569...00000'. The transaction is pending. Below this, a transaction to 'KYCContract.viewCustomer' is shown with data '0x107...00000'. The transaction is pending.

Banks report another bank:

If more than $\frac{1}{3}$ rd of the bank report, banks voting rights are removed.

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows several functions: `removeBank`, `removeKYC`, `reportBank`, `upVoteCustom...`, `getBankCompl...`, `noOfBanks`, `viewBankDeta...`, and `viewCustomer`. The `viewBankDeta...` function is selected, showing its arguments: `0: string: _bankName bank5`, `1: address: _ethAddress1 0x617F2E2fD72FD9D5503197092aC168c91465E7f2`, `2: uint256: _complaints 0`, `3: uint256: _kycCount 0`, `4: bool: _isAllowedToVote true`, and `5: string: _regNumber reg5`.

The main editor shows the Solidity code for the `KYCContract.sol` file. The code includes a `viewBankDetails` function that returns bank details and a `reportBank` function that reports a complaint against a bank. The code is as follows:

```
357   
358   
359   
360   
361   
362   
363   
364   
365   
366   
367   
368   
369   
370   
371   
372   
373   
374   
375   
376   
377   
378   
379   
380   
381   
382   
383   
384   
385   
386   
387   
388   
389   
390   
391   
392   
393   
394   
395   
396   
397   
398   
399   
400   
401   
402   
403   
404   
405   
406   
407   
408   
409   
410   
411   
412   
413   
414   
415   
416   
417   
418   
419   
420   
421   
422   
423   
424   
425   
426   
427   
428   
429   
430   
431   
432   
433   
434   
435   
436   
437   
438   
439   
440   
441   
442   
443   
444   
445   
446   
447   
448   
449   
450   
451   
452   
453   
454   
455   
456   
457   
458   
459   
460   
461   
462   
463   
464   
465   
466   
467   
468   
469   
470   
471   
472   
473   
474   
475   
476   
477   
478   
479   
480   
481   
482   
483   
484   
485   
486   
487   
488   
489   
490   
491   
492   
493   
494   
495   
496   
497   
498   
499   
500   
501   
502   
503   
504   
505   
506   
507   
508   
509   
510   
511   
512   
513   
514   
515   
516   
517   
518   
519   
520   
521   
522   
523   
524   
525   
526   
527   
528   
529   
530   
531   
532   
533   
534   
535   
536   
537   
538   
539   
540   
541   
542   
543   
544   
545   
546   
547   
548   
549   
550   
551   
552   
553   
554   
555   
556   
557   
558   
559   
560   
561   
562   
563   
564   
565   
566   
567   
568   
569   
570   
571   
572   
573   
574   
575   
576   
577   
578   
579   
580   
581   
582   
583   
584   
585   
586   
587   
588   
589   
590   
591   
592   
593   
594   
595   
596   
597   
598   
599   
600   
601   
602   
603   
604   
605   
606   
607   
608   
609   
610   
611   
612   
613   
614   
615   
616   
617   
618   
619   
620   
621   
622   
623   
624   
625   
626   
627   
628   
629   
630   
631   
632   
633   
634   
635   
636   
637   
638   
639   
640   
641   
642   
643   
644   
645   
646   
647   
648   
649   
650   
651   
652   
653   
654   
655   
656   
657   
658   
659   
660   
661   
662   
663   
664   
665   
666   
667   
668   
669   
670   
671   
672   
673   
674   
675   
676   
677   
678   
679   
680   
681   
682   
683   
684   
685   
686   
687   
688   
689   
690   
691   
692   
693   
694   
695   
696   
697   
698   
699   
700   
701   
702   
703   
704   
705   
706   
707   
708   
709   
710   
711   
712   
713   
714   
715   
716   
717   
718   
719   
720   
721   
722   
723   
724   
725   
726   
727   
728   
729   
730   
731   
732   
733   
734   
735   
736   
737   
738   
739   
740   
741   
742   
743   
744   
745   
746   
747   
748   
749   
750   
751   
752   
753   
754   
755   
756   
757   
758   
759   
760   
761   
762   
763   
764   
765   
766   
767   
768   
769   
770   
771   
772   
773   
774   
775   
776   
777   
778   
779   
780   
781   
782   
783   
784   
785   
786   
787   
788   
789   
790   
791   
792   
793   
794   
795   
796   
797   
798   
799   
800   
801   
802   
803   
804   
805   
806   
807   
808   
809   
810   
811   
812   
813   
814   
815   
816   
817   
818   
819   
820   
821   
822   
823   
824   
825   
826   
827   
828   
829   
830   
831   
832   
833   
834   
835   
836   
837   
838   
839   
840   
841   
842   
843   
844   
845   
846   
847   
848   
849   
850   
851   
852   
853   
854   
855   
856   
857   
858   
859   
860   
861   
862   
863   
864   
865   
866   
867   
868   
869   
870   
871   
872   
873   
874   
875   
876   
877   
878   
879   
880   
881   
882   
883   
884   
885   
886   
887   
888   
889   
890   
891   
892   
893   
894   
895   
896   
897   
898   
899   
900   
901   
902   
903   
904   
905   
906   
907   
908   
909   
910   
911   
912   
913   
914   
915   
916   
917   
918   
919   
920   
921   
922   
923   
924   
925   
926   
927   
928   
929   
930   
931   
932   
933   
934   
935   
936   
937   
938   
939   
940   
941   
942   
943   
944   
945   
946   
947   
948   
949   
950   
951   
952   
953   
954   
955   
956   
957   
958   
959   
960   
961   
962   
963   
964   
965   
966   
967   
968   
969   
970   
971   
972   
973   
974   
975   
976   
977   
978   
979   
980   
981   
982   
983   
984   
985   
986   
987   
988   
989   
990   
991   
992   
993   
994   
995   
996   
997   
998   
999   
1000   
1001   
1002   
1003   
1004   
1005   
1006   
1007   
1008   
1009   
1010   
1011   
1012   
1013   
1014   
1015   
1016   
1017   
1018   
1019   
1020   
1021   
1022   
1023   
1024   
1025   
1026   
1027   
1028   
1029   
1030   
1031   
1032   
1033   
1034   
1035   
1036   
1037   
1038   
1039   
1040   
1041   
1042   
1043   
1044   
1045   
1046   
1047   
1048   
1049   
1050   
1051   
1052   
1053   
1054   
1055   
1056   
1057   
1058   
1059   
1060   
1061   
1062   
1063   
1064   
1065   
1066   
1067   
1068   
1069   
1070   
1071   
1072   
1073   
1074   
1075   
1076   
1077   
1078   
1079   
1080   
1081   
1082   
1083   
1084   
1085   
1086   
1087   
1088   
1089   
1090   
1091   
1092   
1093   
1094   
1095   
1096   
1097   
1098   
1099   
1100   
1101   
1102   
1103   
1104   
1105   
1106   
1107   
1108   
1109   
1110   
1111   
1112   
1113   
1114   
1115   
1116   
1117   
1118   
1119   
1120   
1121   
1122   
1123   
1124   
1125   
1126   
1127   
1128   
1129   
1130   
1131   
1132   
1133   
1134   
1135   
1136   
1137   
1138   
1139   
1140   
1141   
1142   
1143   
1144   
1145   
1146   
1147   
1148   
1149   
1150   
1151   
1152   
1153   
1154   
1155   
1156   
1157   
1158   
1159   
1160   
1161   
1162   
1163   
1164   
1165   
1166   
1167   
1168   
1169   
1170   
1171   
1172   
1173   
1174   
1175   
1176   
1177   
1178   
1179   
1180   
1181   
1182   
1183   
1184   
1185   
1186   
1187   
1188   
1189   
1190   
1191   
1192   
1193   
1194   
1195   
1196   
1197   
1198   
1199   
1200   
1201   
1202   
1203   
1204   
1205   
1206   
1207   
1208   
1209   
1210   
1211   
1212   
1213   
1214   
1215   
1216   
1217   
1218   
1219   
1220   
1221   
1222   
1223   
1224   
1225   
1226   
1227   
1228   
1229   
1230   
1231   
1232   
1233   
1234   
1235   
1236   
1237   
1238   
1239   
1240   
1241   
1242   
1243   
1244   
1245   
1246   
1247   
1248   
1249   
1250   
1251   
1252   
1253   
1254   
1255   
1256   
1257   
1258   
1259   
1260   
1261   
1262   
1263   
1264   
1265   
1266   
1267   
1268   
1269   
1270   
1271   
1272   
1273   
1274   
1275   
1276   
1277   
1278   
1279   
1280   
1281   
1282   
1283   
1284   
1285   
1286   
1287   
1288   
1289   
1290   
1291   
1292   
1293   
1294   
1295   
1296   
1297   
1298   
1299   
1300   
1301   
1302   
1303   
1304   
1305   
1306   
1307   
1308   
1309   
1310   
1311   
1312   
1313   
1314   
1315   
1316   
1317   
1318   
1319   
1320   
1321   
1322   
1323   
1324   
1325   
1326   
1327   
1328   
1329   
1330   
1331   
1332   
1333   
1334   
1335   
1336   
1337   
1338   
1339   
1340   
1341   
1342   
1343   
1344   
1345   
1346   
1347   
1348   
1349   
1350   
1351   
1352   
1353   
1354   
1355   
1356   
1357   
1358   
1359   
1360   
1361   
1362   
1363   
1364   
1365   
1366   
1367   
1368   
1369   
1370   
1371   
1372   
1373   
1374   
1375   
1376   
1377   
1378   
1379   
1380   
1381   
1382   
1383   
1384   
1385   
1386   
1387   
1388   
1389   
1390   
1391   
1392   
1393   
1394   
1395   
1396   
1397   
1398   
1399   
1400   
1401   
1402   
1403   
1404   
1405   
1406   
1407   
1408   
1409   
1410   
1411   
1412   
1413   
1414   
1415   
1416   
1417   
1418   
1419   
1420   
1421   
1422   
1423   
1424   
1425   
1426   
1427   
1428   
1429   
1430   
1431   
1432   
1433   
1434   
1435   
1436   
1437   
1438   
1439   
1440   
1441   
1442   
1443   
1444   
1445   
1446   
1447   
1448   
1449   
1450   
1451   
1452   
1453   
1454   
1455   
1456   
1457   
1458   
1459   
1460   
1461   
1462   
1463   
1464   
1465   
1466   
1467   
1468   
1469   
1470   
1471   
1472   
1473   
1474   
1475   
1476   
1477   
1478   
1479   
1480   
1481   
1482   
1483   
1484   
1485   
1486   
1487   
1488   
1489   
1490   
1491   
1492   
1493   
1494   
1495   
1496   
1497   
1498   
1499   
1500   
1501   
1502   
1503   
1504   
1505   
1506   
1507   
1508   
1509   
1510   
1511   
1512   
1513   
1514   
1515   
1516   
1517   
1518   
1519   
1520   
1521   
1522   
1523   
1524   
1525   
1526   
1527   
1528   
1529   
1530   
1531   
1532   
1533   
1534   
1535   
1536   
1537   
1538   
1539   
1540   
1541   
1542   
1543   
1544   
1545   
1546   
1547   
1548   
1549   
1550   
1551   
1552   
1553   
1554   
1555   
1556   
1557   
1558   
1559   
1560   
1561   
1562   
1563   
1564   
1565   
1566   
1567   
1568   
1569   
1570   
1571   
1572   
1573   
1574   
1575   
1576   
1577   
1578   
1579   
1580   
1581   
1582   
1583   
1584   
1585   
1586   
1587   
1588   
1589   
1590   
1591   
1592   
1593   
1594   
1595   
1596   
1597   
1598   
1599   
1600   
1601   
1602   
1603   
1604   
1605   
1606   
1607   
1608   
1609   
1610   
1611   
1612   
1613   
1614   
1615   
1616   
1617   
1618   
1619   
1620   
1621   
1622   
1623   
1624   
1625   
1626   
1627   
1628   
1629   
1630   
1631   
1632   
1633   
1634   
1635   
1636   
1637   
1638   
1639   
1640   
1641   
1642   
1643   
1644   
1645   
1646   
1647   
1648   
1649   
1650   
1651   
1652   
1653   
1654   
1655   
1656   
1657   
1658   
1659   
1660   
1661   
1662   
1663   
1664   
1665   
1666   
1667   
1668   
1669   
1670   
1671   
1672   
1673   
1674   
1675   
1676   
1677   
1678   
1679   
1680   
1681   
1682   
1683   
1684   
1685   
1686   
1687   
1688   
1689   
1690   
1691   
1692   
1693   
1694   
1695   
1696   
1697   
1698   
1699   
1700   
1701   
1702   
1703   
1704   
1705   
1706   
1707   
1708   
1709   
1710   
1711   
1712   
1713   
1714   
1715   
1716   
1717   
1718   
1719   
1720   
1721   
1722   
1723   
1724   
1725   
1726   
1727   
1728   
1729   
1730   
1731   
1732   
1733   
1734   
1735   
1736   
1737   
1738   
1739   
1740   
1741   
1742   
1743   
1744   
1745   
1746   
1747   
1748   
1749   
1750   
1751   
1752   
1753   
1754   
1755   
1756   
1757   
1758   
1759   
1760   
1761   
1762   
1763   
1764   
1765   
1766   
1767   
1768   
1769   
1770   
1771   
1772   
1773   
1774   
1775   
1776   
1777   
1778   
1779   
1780   
1781   
1782   
1783   
1784   
1785   
1786   
1787   
1788   
1789   
1790   
1791   
1792   
1793   
1794   
1795   
1796   
1797   
1798   
1799   
1800   
1801   
1802   
1803   
1804   
1805   
1806   
1807   
1808   
1809   
1810   
1811   
1812   
1813   
1814   
1815   
1816   
1817   
1818   
1819   
1820   
1821   
1822   
1823   
1824   
1825   
1826   
1827   
1828   
1829   
1830   
1831   
1832   
1833   
1834   
1835   
1836   
1837   
1838   
1839   
1840   
1841   
1842   
1843   
1844   
1845   
1846   
1847   
1848   
1849   
1850   
1851   
1852   
1853   
1854   
1855   
1856   
1857   
1858   
1859   
1860   
1861   
1862   
1863   
1864   
1865   
1866   
1867   
1868   
1869   
1870   
1871   
1872   
1873   
1874   
1875   
1876   
1877   
1878   
1879   
1880   
1881   
1882   
1883   
1884   
1885   
1886   
1887   
1888   
1889   
1890   
1891   
1892   
1893   
1894   
1895   
1896   
1897   
1898   
1899   
1900   
1901   
1902   
1903   
1904   
1905   
1906   
1907   
1908   
1909   
1910   
1911   
1912   
1913   
1914   
1915   
1916   
1917   
1918   
1919   
1920   
1921   
1922   
1923   
1924   
1925   
1926   
1927   
1928   
1929   
1930   
1931   
1932   
1933   
1934   
1935   
1936   
1937   
1938   
1939   
1940   
1941   
1942   
1943   
1944   
1945   
1946   
1947   
1948   
1949   
1950   
1951   
1952   
1953   
1954   
1955   
1956   
1957   
1958   
1959   
1960   
1961   
1962   
1963   
1964   
1965   
1966   
1967   
1968   
1969   
1970   
1971   
1972   
1973   
1974   
1975   
1976   
1977   
1978   
1979   
1980   
1981   
1982   
1983   
1984   
1985   
1986   
1987   
1988   
1989   
1990   
1991   
1992   
1993   
1994   
1995   
1996   
1997   
1998   
1999   
2000   
2001   
2002   
2003   
2004   
2005   
2006   
2007   
2008   
2009   
2010   
2011   
2012   
2013   
2014   
2015   
2016   
2017   
2018   
2019   
2020   
2021   
2022   
2023   
2024   
2025   
2026   
2027   
2028   
2029   
2030   
2031   
2032   
2033   
2034   
2035   
2036   
2037   
2038   
2039   
2040   
2041   
2042   
2043   
2044   
2045   
2046   
2047   
2048   
2049   
2050   
2051   
2052   
2053   
2054   
2055   
2056   
2057   
2058   
2059   
2060   
2061   
2062   
2063   
2064   
2065   
2066   
2067   
2068   
2069   
2070   
2071   
2072   
2073   
2074   
2075   
2076   
2077   
2078   
2079   
2080   
2081   
2082   
2083   
2084   
2085   
2086   
2087   
2088   
2089   
2090   
2091   
2092   
2093   
2094   
2095   
2096   
2097   
2098   
2099   
2100   
2101   
2102   
2103   
2104   
2105   
2106   
2107   
2108   
2109   
2110   
2111   
2112   
2113   
2114   
2115   
2116   
2117   
2118   
2119   
2120   
2121   
2122   
2123   
2124   
2125   
2126   
2127   
2128   
2129   
2130   
2131   
2132   
2133   
2134   
2135   
2136   
2137   
2138   
2139   
2140   
2141   
2142   
2143   
2144   
2145   
2146   
2147   
2148   
2149   
21
```


Activities Google Chrome Sep 7 00:43

Remix - Ethereum IDE

remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.9+commit.c68bc34e.js

DEPLOY & RUN TRANSACTIONS

- raiseKYC "anu","adhar2hash"
- removeBank "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2"
- removeKYC string _userName
- reportBank "0x617F2E2d72FD9D5503"
- upVoteCustom... "anu"
- getBankCompl... "0x617F2E2d72FD9D5503"
- noOfBanks
- viewBankDeta... "0x617F2E2d72FD9D5503"
- viewCustomer "anu"

0: uint256: _complaints 4

0: uint256: 6

0: string: _bankName bank5

1: address: _ethAddress1 0x617F2E2d72FD9D5503197092aC168c91465E7f2

2: uint256: _complaints 4

3: uint256: _kycCount 0

4: bool: _isAllowedToVote false

5: string: _regNumber reg5

0: string: anu

1: string: adhar2hash

```
382  /**
383  * Report Bank - This function is used to report a complaint against any bank in the network.
384  * - validate if bank is validate
385  * - validate if the reporting bank is a valid bank
386  *
387  *todo: validate a bank is allowed to complaint only once.
388  */
389
390  function reportBank(address _bank)
391  public
392  bankExists(_bank)
393  bankExists(msg.sender) {
394
395
396      banks[_bank].complaints += 1; // increment number of complaints by 1
397
398      //modify the isAllowedToVote status of the bank -
399      // - if any bank gets reported more than one-third of the banks present in the network, it will not be allowed to do KYC anymore.
400
401      if(( noOfBanks > 5) && (banks[_bank].complaints >= (noOfBanks / 3))) {
402          banks[_bank].isAllowedToVote = false;
403
404          noOfBanks = noOfBanks.sub(1);
405      }
406  }
407
408
409
```

mapping(string => struct KYCContract.Customer) customers 33 reference(s)

listen on network Search with transaction hash or address

[call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: KYCContract.getBankComplaints(address) data: 0x5ec...5e7f2 Debug

call to KYCContract.viewBankDetails

[call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: KYCContract.viewBankDetails(address) data: 0x88c...5e7f2 Debug

transaction hash 0x8cf39cc60b6ef94604261b71be9d9d421b383d4f4a8ff264a3914898f4d64aef

from 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

to KYCContract.viewBankDetails(address) 0xd7Ca4e99f7C171B9ea2De80d3363c47009afaC5F

execution cost 37161 gas (Cost only applies when called by a contract)