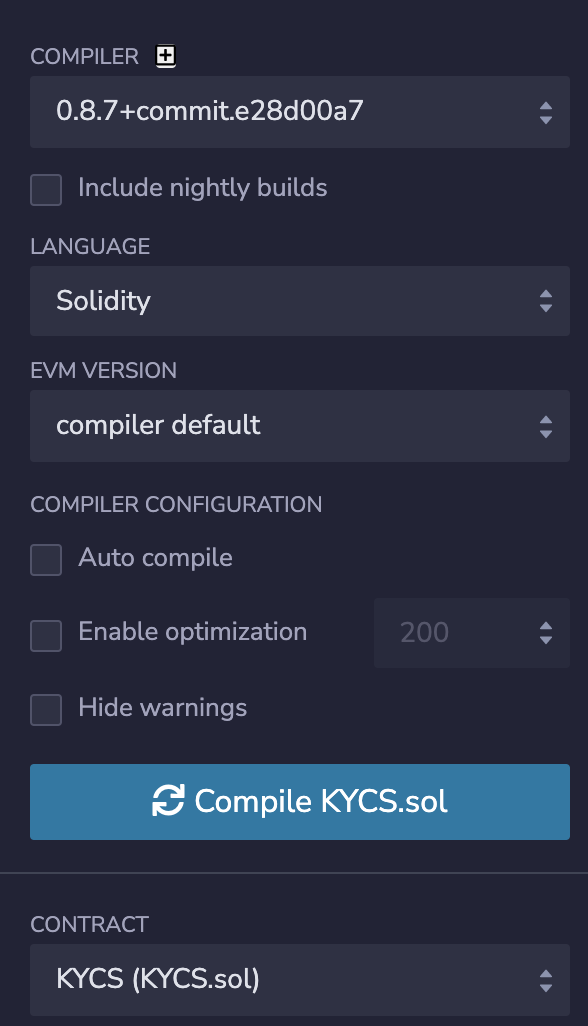
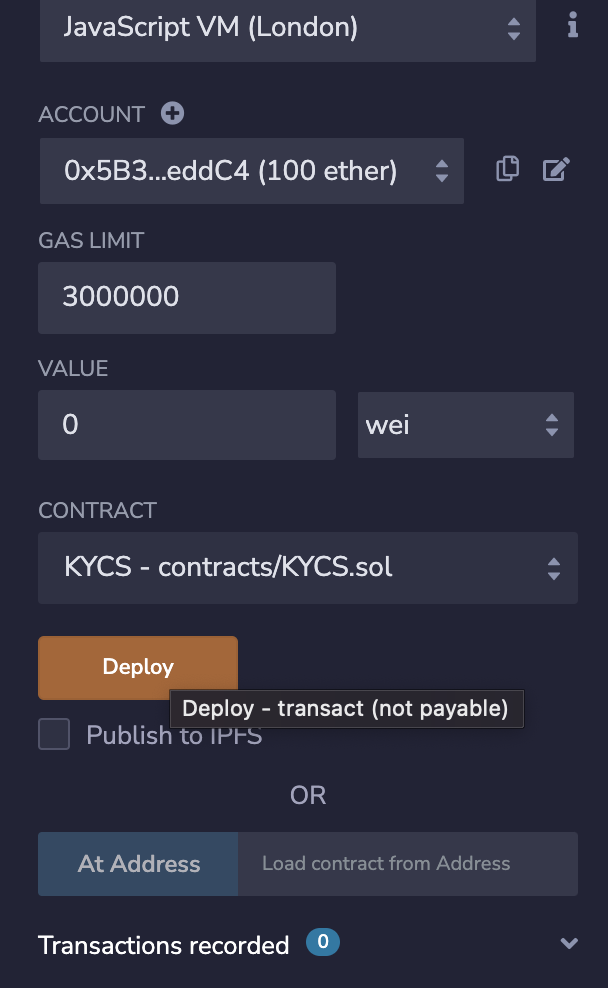
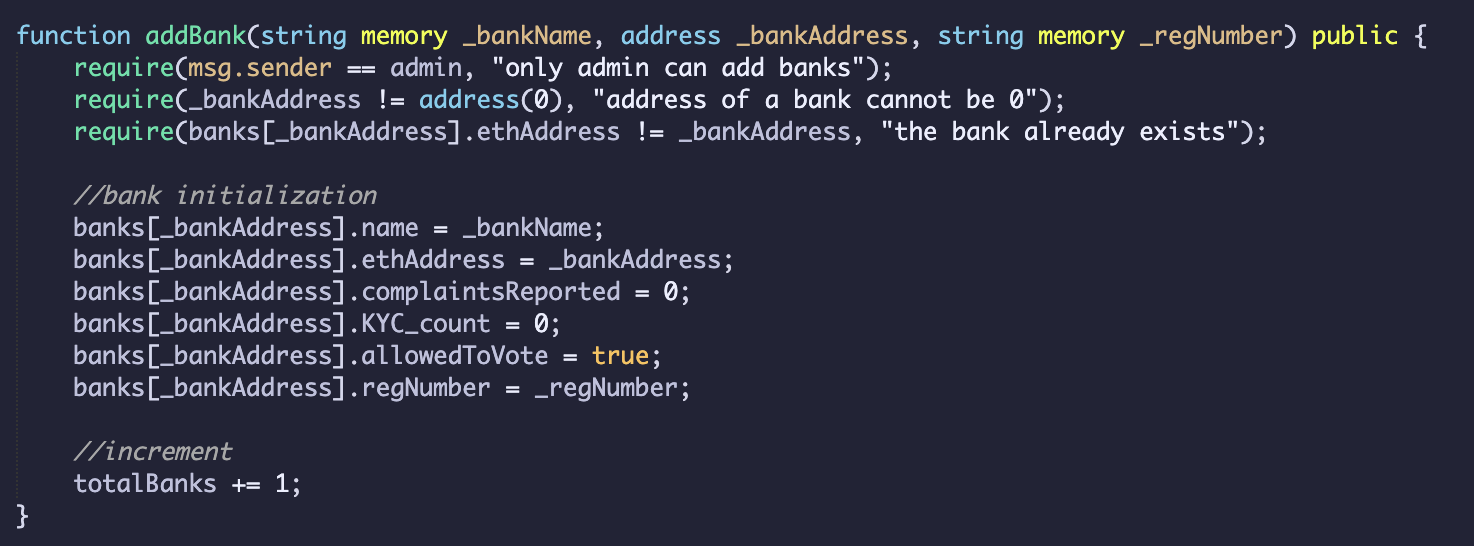
**Smart Contract Flow**

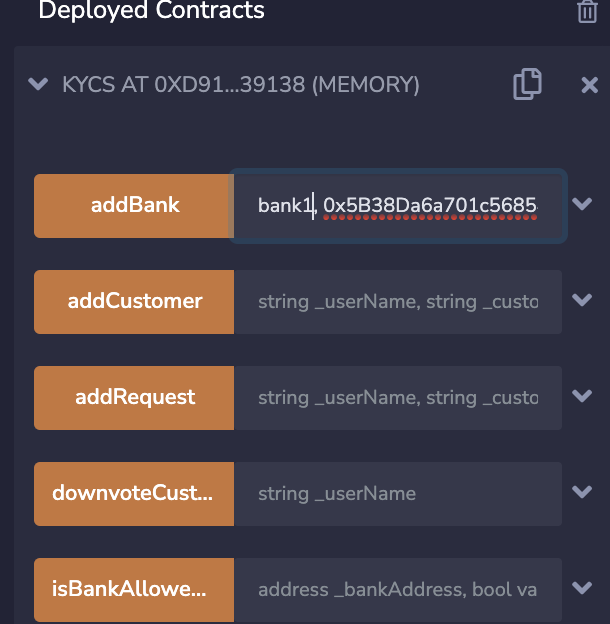
* Start by compiling the KYC contract file.



* After a successful compilation, deploy the contract to the network provided by Remix IDE. Note that the address that deploys the contract becomes the admin.  
    
  

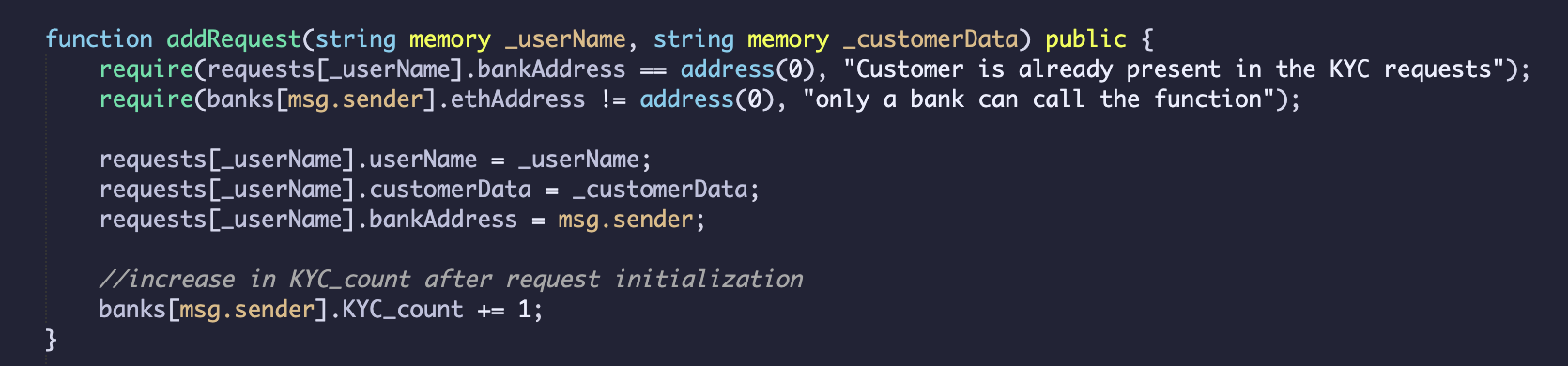
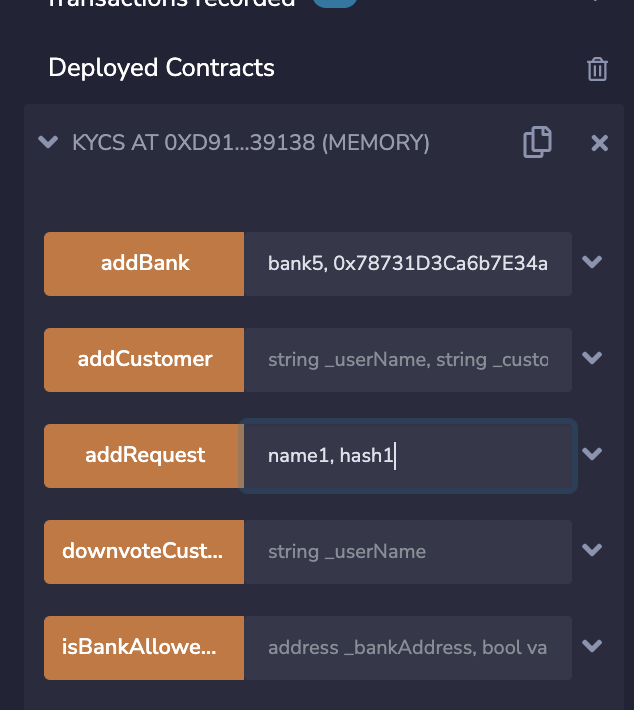
* The admin starts by adding a few banks to the network -



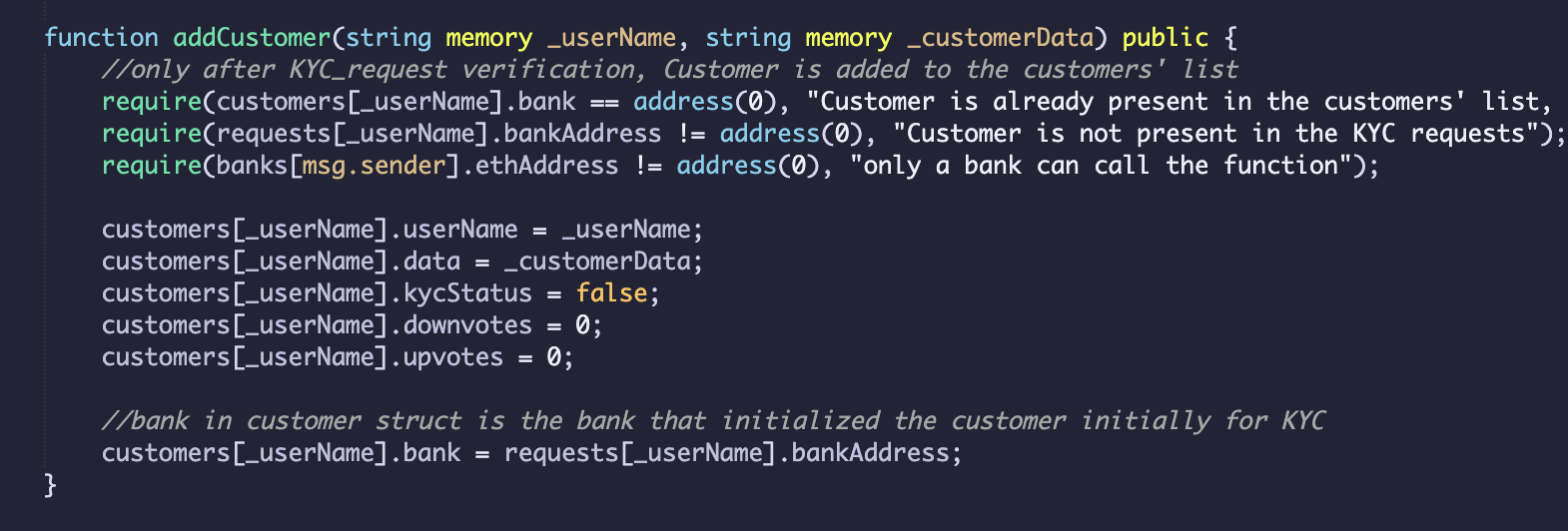
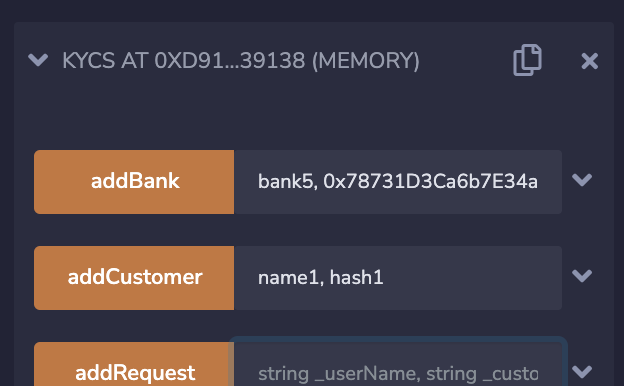
****

Admin can also remove the bank, set the bank’s voting access to true or false.

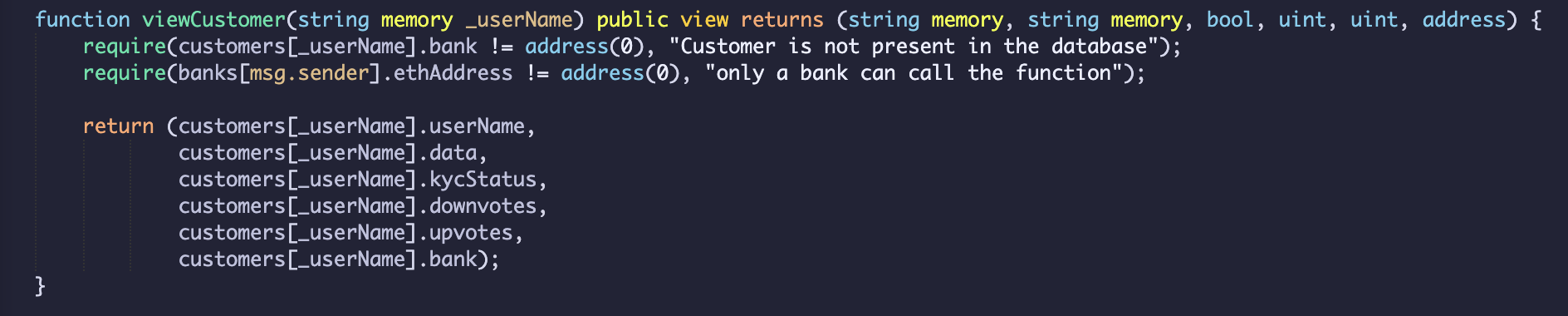
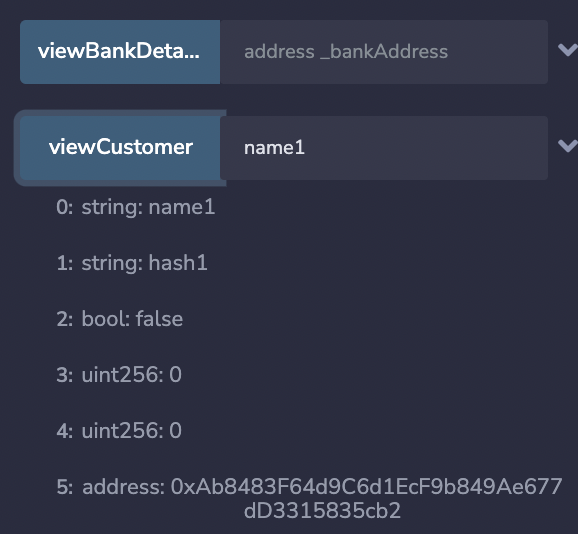
* Using any one of the banks(addresses) that were created, start up a KYC request for a customer.

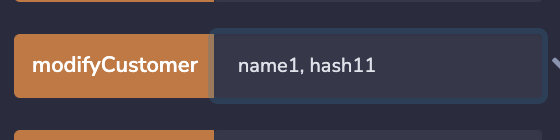
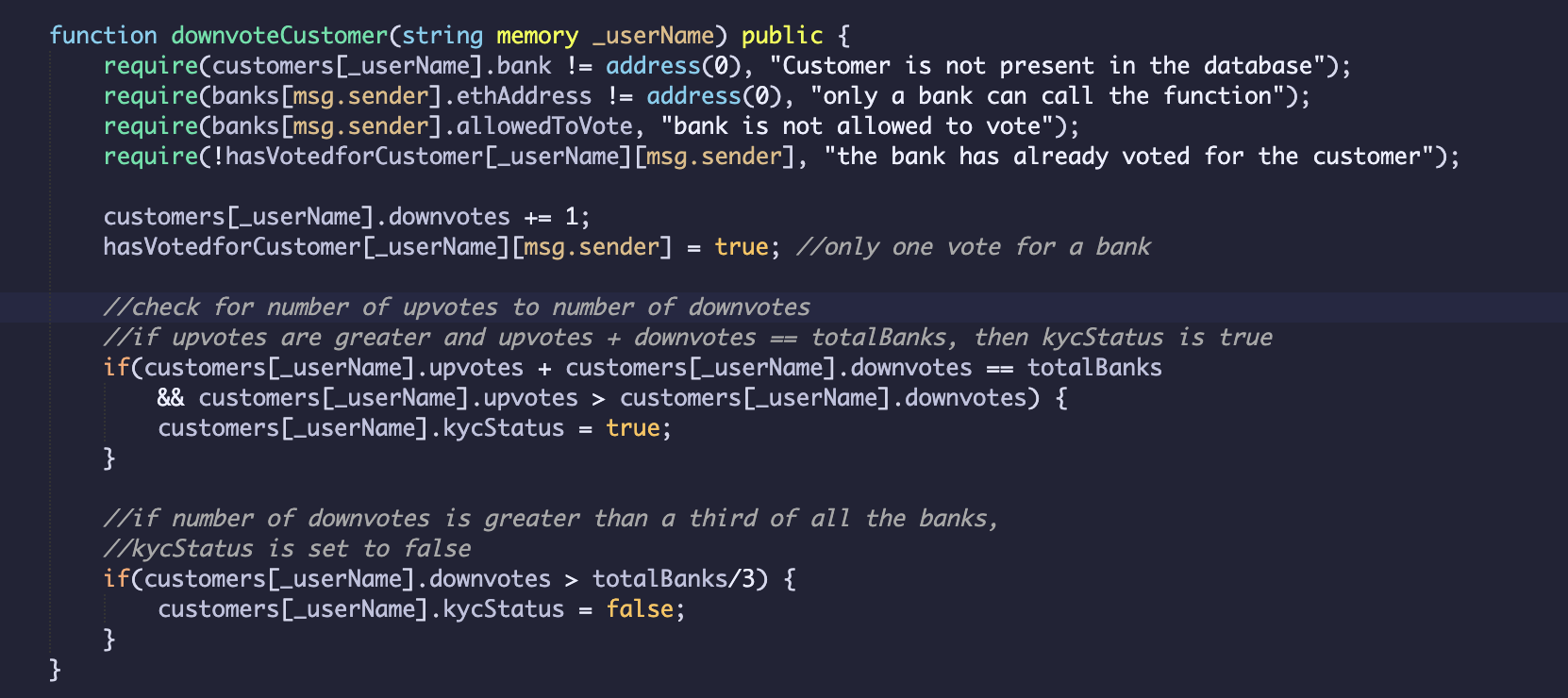
  
  


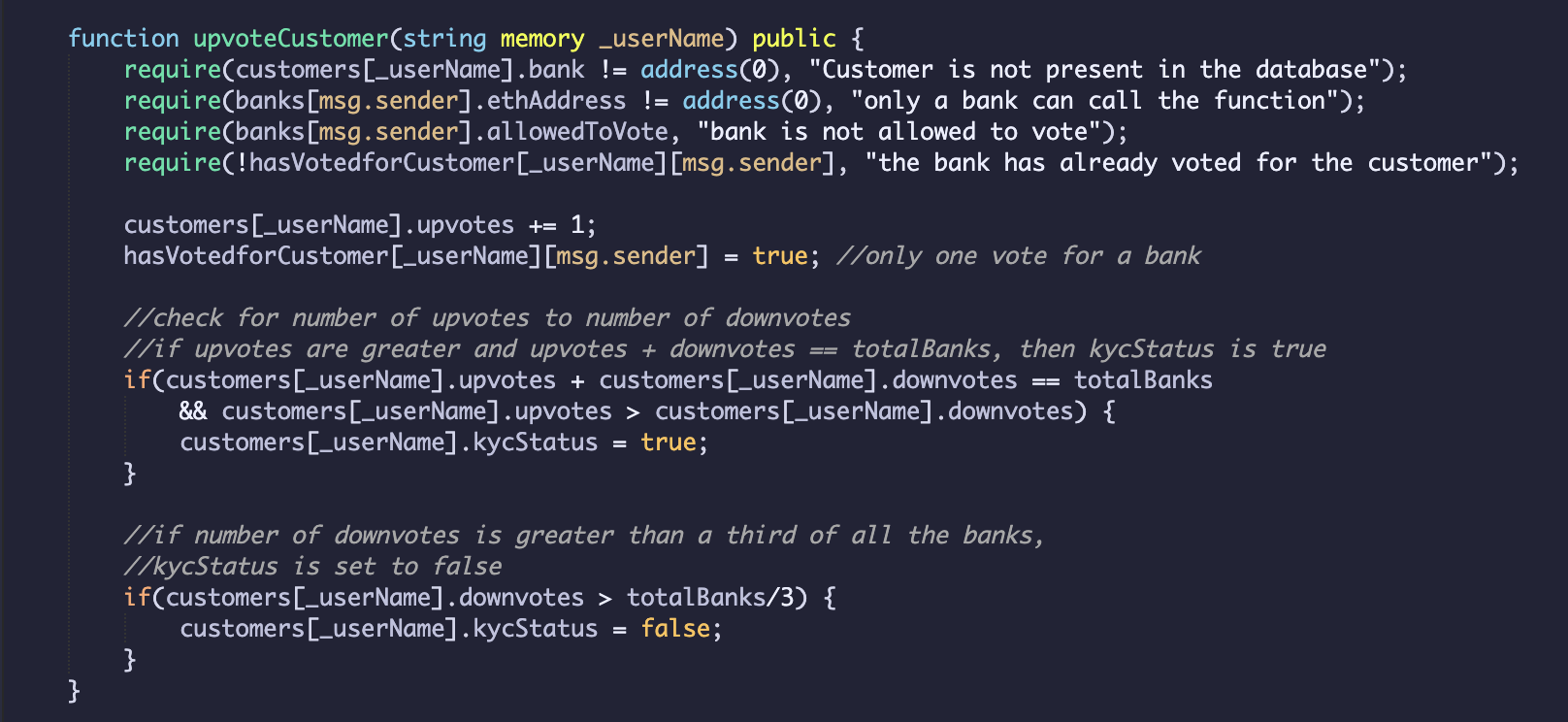
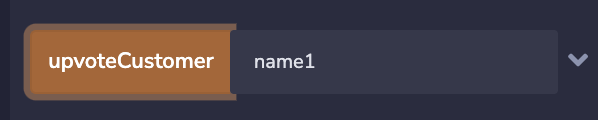
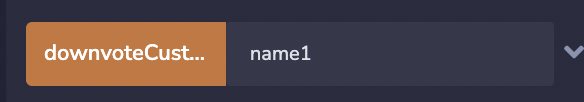
* After KYC verification, a bank can then add the customer to the customer’s list

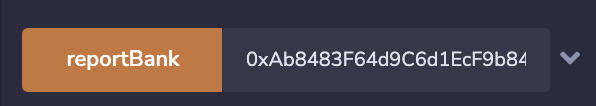
  
  


* To view, the details of a customer click on “viewCustomer”. This will give all the data regarding the customer like name, up-votes, down-votes and more.

* The customer data can also be modified if required. This will reset the hash of the customer data, up-votes, down-votes against the customer.  
    
  
* The banks are allowed to up-vote a customer or down-vote him/her. If the up-votes are more than the down-votes, then the “KYC\_status” of the customer is set to true. If the total number of down-votes is greater than a third of the total banks (even if the up-votes are more than the down-votes), the “KYC\_status” of the customer is set to false. Note - only one vote for one bank.  
    
  

* View the customer details again to check the “KYC\_status” of a customer.
* Banks can also report other banks(file complaints) if they think that a particular bank is committing fraud. Even in this case, if more than a third of the total number of banks have reported against a bank, then that bank can no longer up-vote/down-vote a customer.  
    
    
    
  
* The complaints against a bank can also be viewed.  
    
  