

## PrimeDefi - Smart Contract Test Cases Writeup

**function createIssuer**(string memory \_name, string memory \_creditRating) => Issuer create profile with meta data (name, credit rating)

- Return false if issuer (with same wallet address) already registered
- Input data format: must be strings
- issuerCount increased by 1

**function getIssuerMetaDataByISIN**(string memory \_ISIN) public view returns(  
string memory issuerName,  
string memory creditRating  
)

=> Any user can get details of issuer meta data by ISIN

- Returns issuer name and credit rating

**function getAllIssuerDetailsByAddr**(address \_addr) public view returns(  
string memory name,  
string memory creditRating,  
string[] memory dealsIssued,  
uint256 totalIssuedAmount,  
uint256 totalEscrowFundLockedIn,  
uint256 totalAvailBalance  
)

=> Issuer can get details of its own profile and wallet balances by calling this function

- Return false if caller is not issuer
- Return issuer name, credit rating, list of ISINs of deals issued, total issued amount, total escrow lock0in, and total available balance (to withdraw)

**function editIssuerMetaData**(address \_addr, string memory \_newName, string memory \_newCreditRating)

=> Issuer edit its own profile meta data (name, credit rating)

- Front end should pre-populate both data fields with existing data then user can modify
- Input data cannot be null
- Caller must be issuer

**function createDealIssuance**(  
string memory \_ISIN,  
string memory \_dealName,  
uint256 \_initSize,

```

uint256 _minSize,
uint256 _faceValue,
uint256 _offerPrice,
uint _offerCloseTime,
uint256 _term,
uint256 _interestRate,
uint[] memory _interestPaymentDates,
uint256 _upfrontFee
)

```

=> Issuer creates deal issuance (to kick start offering period):

- Return false if ISIN already used in existing issuances
- Input data format: Date/time in uint
- Input data format: Term in days
- Input data format: Rate and fee: assuming 3 decimals, need to convert to whole number before passing (i.e. use 1,250 for 1.25%)
- No empty fields allowed
- deal.state becomes OFFERLIVE
- deal.isOfferLive = true
- dealCount increased by 1

**function getIssuanceStateNameRateDatesDetailsByISIN**(string memory \_ISIN) public view returns(

```

State state,
string memory dealName,
uint offerStartTime,
uint offerCloseTime,
uint256 term,
uint256 interestRate,
uint[] memory interestPaymentDates
)

```

=> Any user can get details of a deal issuance (partially), looking up by its ISIN

- Return open offering state
- Return deal name
- Return offer starting time
- Return offer close time
- Return term of the deal (in # of days)
- Return interest rate (yield)
- Return a numerical list of interest payment dates

**function getIssuanceSizePriceTotalBidDetailsByISIN**(string memory \_ISIN) public view returns(

```
uint256 initSize,
uint256 minSize,
uint256 faceValue,
uint256 offerPrice,
uint256 upfrontFee,
uint256 escrowRatio,
uint256 totalInvestorBid ) => Any user can get details of a deal issuance
```

(partially), looking up by its ISIN

- Return deal initial size
- Return minimum size to launch
- Return deal face value
- Return offer price
- Return upfront fee charged by the issuer
- Return escrow ratio (use 5% for now)

```
function editDealIssuance(
    string memory _ISIN,
    string memory _newDealName,
    uint256 _newInitSize,
    uint256 _newMinSize,
    uint256 _newOfferPrice,
    uint _newOfferCloseTime,
    uint256 _newTerm,
    uint256 _newInterestRate,
    uint[] memory _newInterestPaymentDates,
    uint256 _newUpfrontFee,
    uint256 _newEscrowRatio
)
```

=> Issuer can edit its own issuance when the offer period is live:

- Return false if caller is not issuer
- Return false if deal offering period is not live
- Input data follow required format
- No empty fields allowed

```
function createTokenMetaData(string memory _ISIN, string memory _name)
```

=> Issuer sets up deal token meta data by ISIN

- Return false if deal offer is not live
- Return false if caller is not issuer (only issuer is allowed to call)
- Data format: string for token name (symbol will set the same as deal ISIN)

```
function getTokenDetails(string memory _ISIN) public view returns(
    string memory tokenName,
    string memory symbol,
    uint256 totalSupply
)
```

=> Any user can get details of a deal token already created, looking up by deal ISIN

- Return token name
- Return token symbol
- Return total supply (0 if the deal not closed yet)

```
function editTokenMetaData(string memory _ISIN, string memory _newName)
```

=> Issuer can edit token name:

- Return false if deal offer is not live
- Return false if caller is not issuer
- Token symbol will be replaced to new ISIN if existing ISIN was updated prior to this call

=====

```
function investorMakeDeposit(IERC20 _stableCoin, uint256 _amount) => Investor
deposit USDC stablecoin into their wallet in our app
```

TODO: test Kovan USDC implementation => dApp USDC balance increased by investor deposit amount

- Investor details - totalBalance increased by amount
- Investor details - availBalance increased by amount
- totalInvestorCount increased by 1 (only if new investor)

```
function getAllInvestorDetails(address _addr) public view returns(
    string[] memory dealsBidded,
    uint256 totalBalance,
    uint256 totalLockedInBid,
    uint256 availBalance
)
```

=> Investor can get details of its own profile and wallet details by calling this function

- Return a list of ISINs of all the issuances bedded
- Return total balance
- Return total lockedin bid (money spent on existing open or successful bids)
- Return available balance (to withdraw or place more bids)

```
function investorMakeBid(string memory _ISIN, uint256 _bidAmount)
```

-> Investor makes a bid to an open issuance:

- Return false if already bidded
- Return false if deal offering is not live
- Return false if insufficient fund in investor wallet in the app (need to make additional deposit first)
- Deal issuance details: add to totalInvestorBid
- Deal issuance details: add caller to this deal's investors list
- Deal issuance details: investorCount increased by 1
- Deal issuance details: set investorAlreadyBid to true for this investor on this deal
- Investor details: add ISIN to investor's dealsBidded list
- Investor details: add bidAmount to investor's totalLockedInBid wallet balance
- Investor details: reset investor wallet's availBalance to be totalBalance - totalLockedInBid
- Investor details: set investor wallet's lockedInBid balance for this deal to be bidAmount

**function getBidDetails**(string memory \_ISIN, address \_addr) public view returns(  
 bool isClosed,  
 bool isSuccessful,  
 uint256 lockedInAmount)

=> Investor gets details of a bid made earlier

- Returns 3 parameters pertaining to investor bid of the deal

**function investorEditBid**(string memory \_ISIN, address \_addr, uint256 \_newBidAmount)

-> Investor changes bid amount of an already-placed bid

- Return false if deal offering is not live
- Return false if investor hasn't made a bid yet
- Return false if insufficient fund in investor wallet in the app (need to make additional deposit first)
- Deal issuance details: change totalInvestorBid
- Investor details: change investor's totalLockedInBid wallet balance
- Investor details: reset investor wallet's availBalance to be totalBalance - totalLockedInBid
- Investor details: change investor wallet's lockedInBid balance to be the newBidAmount

**function closeDealOffering**(string memory \_ISIN) public returns(bool)

=> This function needs to be called by Chainlink keepers when timestamp clicks pre-set offerCloseTime

- Return false if deal offer is not live
- Return false if caller is not issuer

- Return false if total investor bid < minLaunchSize
- Token minted (with total supply = finalSize / faceValue)
- Token address generated
- Token owner is set to be the Deal DApp contract
- DealTokenManager receives approval and allowance to delegate token transfer up to total supply
- Token address added to tracker\_0x\_addresses the token list
- tokenCount increased by 1
- Calculate allocation ratio: - If totalInvestorBid > finalSize (oversubscribed), ratio = finalSize/totalInvestorBif - Otherwise ratio = 1
- Go through the deal's investor list one by one, adjust each investor's final allocation to be their initial bid \* ratio
- Deal's totalEscrowAmount is calculated and assigned to issuer wallet
- USDC stablecoins transferred from investor wallet to Deal DApp wallet (final allocation amount), & - Issuer wallet is earmarked by total amount and escrow amount (issuer decides when to transfer USDC out to their own wallet using issuerWithdraw function)
- Deal tokens transferred to each investor account by final allocation
- All deal, issuer, investors details updated and state saved
- Return true after successful deal closing and issuance launching

#### **INTERNAL Functions Implemented:**

=====

assessDealOfferingFinalStatus  
 adjustInvestorAllocation  
 transferStableCoin  
 transferDealToken  
 cancelDeal  
 mintToken  
 editInvestorDetailsPostNewIssuance  
 editIssuerDetailsPostNewIssuance