# ShareToken specification (version-1)

## 1. Contract files

The implementation is uploaded on this github: <https://github.com/trungtt1981/smartcontract>

### 1.1. Test files

The test files are stored in the folder “test”. Test instruction is described in the file README.md

### 1.2. Solidity files

The Solidity contract files are stored in the folder “contracts”.

The main contract files are:

* **ShareToken.sol**
  + Implement properties and methods of the ShareToken token according to the requirement specification.
  + The implementation is inherited from standardized ERC20 interface.
  + Some important methods:
    - *sell(address buyer, uint tokens)*: transfer the main sale tokens issued to the buyer. The total amount of main sale tokens issued is stored in the variable “totalTokenIssued”. The token buyer is locked. The contract owner will unlock all token buyers manually.
    - *rewardAirdrop(address \_to, uint \_amount)*: transfer the reward tokens of Airdrop to the receiver. The receiver is locked and will be unlocked by the contract owner.
  + Included in the MainSale contract.
* **MainSale.sol**
  + Contain reference to the ShareToken contract and the WhiteListManager contract.
  + For deployment, addresses of the deployed ShareToken contract and WhiteListManager contract are to be input.
  + Implement the procedure of issuing tokens to the buyers. The daily updating of eth/usd rate is also implemented by using Oraclize API with some required fee. The actual cost (that has been observed) is about 0.013 ETH per updating. So if the main sale runs for 4 weeks, then the total fee will be 4 \* 7 \* 0.013 = 0.364 ETH. This fee is automatically deducted from the MainSale contract balance in ETH.
  + 2 methods for withdrawing all of the MainSale contract balance (in ETH):
    - withdrawToOwner()
    - withdrawTo(address \_to)
* **WhiteListManager.sol**
  + Contain a list of white-listed addresses which can be flexibly updated by the external WhiteListN contracts (e.g. WhiteList1, WhiteList2 and etc).
  + Implement a method used to check if a given address is white-listed or not.
  + Included in the MainSale contract.
* **WhiteListN.sol**
  + Used to hard-code the pre-determined list of presale and seed addresses together with associated tokens.
  + The deployment needs to be specified with the addresses of the deployed WhiteListManager and the deployed ShareToken.
  + During the deployment, the list of seed and presale addresses will be input to whitelist held by the WhiteListManager. In addition, tokens will also be transferred to the specified address.
  + It is quite possible to feed white-listed addresses (at any time) into the WhiteListManager by simply deploying the WhiteListN contracts.

## 2. Deployment procedure

Described in this document: <https://github.com/trungtt1981/smartcontract/blob/master/DeploymentProcedure.txt>

## 3. Contribution

Tran, Thi Thuy Trang

Trinh, Tien Trung