**uToken (Function wise functionality is described in Code)**

**Description of working of Smart Contract is as follows:**

* **Functionalities**

- Wrapping Tokens or Coins

- Unwrapping Tokens or Coins

- Transfer uTokens

- Fee Collection System

- Reward System

- Whitelist Mechanism

- Password and Recovery Phrase Mechanism to perform any action

* **Wrapping Tokens and Coins**

\*\*\_Definition:\_\*\* Locking Tokens or Coins and receiving uToken(s) which is/are equal in numbers to respective tokens or coins. For example: user want to wrap 10 USDT, he will enter 10 USDT and our smart contract will take 10 USDT from user and mint 10 uUSDT tokens in account of user.

\*\*\_Procedure\_\*\*

- Enter desired amount to be wrapped

- System will check that this is new user or not. In case of new user system will ask him to create a password and he will be provided at time a recovery phrase which will be only shown once and will be used by the user to recover his password in case of forgot password. In case of already registered user, system will just only show the form to take user desired amount of token or coin to be wrapped.

- Smart contract will save user password and recovery phrase.

- Smart contract will mint amount of respective uTokens in account of user.

- User can come and unwrap it any time.

* **Un-Wrapping Tokens and Coins**

\*\*\_Definition:\_\*\* Unlocking Tokens or Coins and receiving Token(s) or coin(s) back. For example: user want to unwrap 10 uUSDT, he will enter 10 uUSDT and our smart contract will burn 10 uUSDT and return 10 USDT to the user.

\*\*\_Procedure\_\*\*

- Enter desired amount to be unwrapped

- Smart contract will burn amount of respective uTokens from account of user.

- Respective tokens or coins will be transferred back to the user.

* **Transferring uTokens**

\*\*\_Description:\_\*\* Any time user can transfer uTokens to anyone with some restrictions explained in the procedure section.

\*\*\_Procedure\_\*\*

- Select uToken and click on transfer button.

- Enter recipient address and amount.

- Enter password Click on transfer button.

- Smart contract will transfer uTokens from sender to recipient.

- Now recipient can come to UI and transfer or unwrap any time.

* **Fee Collection System**

\*\*\_Description:\_\*\* Each time user wrap tokens or coins, a specific percentage (0.369%) will be deducted from the entered amount and remaining amount will be wrapped.

\*\*\_Fee Collectors\_\*\*

1. 30% of 0.369% will be transferred to the reward distributor address which will send reward to the winner later.

2. 30% of 0.369% will be transferred to the charity address.

3. 30% of 0.369% will be transferred to the fund address.

4. 10% of 0.369% will be transferred to the programmers address.

\*\*\_Procedure\_\*\*

- Fee will be deducted at time time of wrapping.

* **Reward System**

\*\*\_Description:\_\*\* Upon start of each time period, a winner will be selected which will receive all the reward collected in previous non-collected time periods.

\*\*\_Procedure\_\*\*

- There will be specific time period to collect reward for each time period.

- Each detail will be shown on the UI regarding collectable assets.

- User have to click within reward time limit on the button.

- Smart contract will transfer all the reward collected within previous un-collected time periods.

* **Whitelist Mechanism**

\*\*Definition:\*\* Whitelist users dosn't need to enter password or to perform a transaction from UI for transferring funds.

\*\*Implementation Guide:\*\*

Implementation can be usderstand in two ways.

1. Sender is EOA:

We will not implement that one as for any/all EOA user will have to input their password and as you said >> In the first case, all addresses will be required to enter password. ✅

2. Sender is Uniswap or AAVE:

In the second case, we will check if the sender is a contract address, and then we will allow it to transfer uTokens without any password. ✅

\*\*Strategy to Implement Whitelist Functinality:\*\*

There are two ways to check that caller is a contract address or not:

- if(msg.sender == tx.origin): in case of true, caller is EOA otherwise caller is a contract.

- if(msg.sender.code.length == 0): in case of true, caller is an EOA otherwise caller is a contract.

- \*\*\_Note:\_\*\*

1. Any of above can be used to check whether caller is an EOA or a contract.

2. isContract() function of Address Library is deprecat