

This solution implements a custom ERC20 token that is mintable, not burnable, has a max supply, has additional role for setting transfer restrictions and implements management of admin and transfer admin roles.

## Limitations

---

The assignment is implemented, except for the following limitations:

- Voting mechanism for sending higher than tmax amount of tokens is not implemented
- Similarly voting mechanism for exception for sending higher than tmax amount of tokens then TRANSFERLIMIT is not implemented
- Web3 app is not implemented and events are not emitted
- To vote on TMAX, majority of users need to call the proposeTMAX function with the same value. However, the other users don't know which values were already proposed. Better solution would be to have separate propose and vote functions with unique proposal ids.

## Structure

---

The modified ERC20 token is implemented in the file `contracts/ModifiedERC20.sol`. It is based on the OpenZeppelin ERC20 implementation.

Tests for the token are implemented in the file `test/TestToken`. The tests are ran against local ganache blockchain

## Gas used

---

uses default gas limit of ~6700000

Gas used for minting: 122114

Gas used for minting in batch (2 recipients): 159499

Gas used for proposing TMAX: 53727

Gas used for proposing a new mintAdmin: 57839

Gas used for a transfer: 96852