Smart contract integration testing

```
Feature: ERC20 token & token sale deploy
As an ICO organizer
I want to create a token
So that I can issue them in an ICO

    Scenario: Token creation
        Given an owner account, and an administrator account
        When the owner creates the token
        Then the administrator account should hold all(959.805.000) tokens
        And the token name should be NOON
        And the tokens decimal value should be 18
        And the token should implement the ERC20 interface
        And the owner of the token sale should be the deployer
        And the owner of the token should be the deployer
        And the MIA of the token sale should be the mia parameter
        And the fund collector of the should be the fundCollector parameter
        And the white list manager of the token should be the WLA parameter
        And the secondaryMarketManager of the token should be the token sale

    Scenario: Changing administrators
        Given that I am the owner
        Then I can change the MIA
        And I can change the WLA
        And I can change the TSA
        And I can change the fundCollector
        And I can change the token owner
        And I can change the token sale owner

    Scenario: Changing administrators to invalid address
        Given that I have the necessary permissions
        When I try to change any of the addresses to 0x0
        Then the operation should fail

Feature: Trading tokens
As a token owner
I want to trade tokens
So I get something in exchange (?)

    Scenario: Transferring tokens from my account
        Given that I have X tokens on my account
        When I transfer Y <= X tokens from my account to a recipient
        Then the recipient should have Y more tokens then before
        And I should have X - Y tokens

    Scenario: Transferring too many tokens
        Given that I have X tokens on my account
        When I transfer Y > X tokens from my account to a recipient
        Then the operation should fail

    Scenario: Allowing an other account (spender) to handle my tokens
        Given that I own an account
        And I approved an address to spend X tokens
        Then their allowance should be X

    Scenario: Spending approved tokens
        Given that I have X tokens
        And an account has Y allowance to my account
        When they transfer Z <= Y, Z <= X tokens from my account
        Then the operation should be successful
        And their allowance should be Y - Z
        And I should have X - Z tokens

    Scenario: Spending more than approved tokens
        Given that I have X tokens
        And an account has Y allowance to my account
        When they transfer Z > Y, Z <= X tokens from my account
        Then the operation should fail

    Scenario: Spending more than available but less than approved tokens
```

```
        Given that I have X allowance on an account
        And an account has Y tokens
        When I try to transfer Z (Y < Z < X) tokens from the account
        Then the operation should fail

    Scenario: Transferring tokens without access
        Given that I did not approve an address
        When that address transfers tokens from my account
        Then the operation should fail

    Scenario: Modifying allowance of spenders
        Given that a spender is allowed to transfer at most X tokens
        When I approve for that spender Y tokens
        Then the their new allowance should be Y

    Scenario: Spending too many tokens from aprroved account
        Given that I have X allowance and have Y tokens on the account ( X > Y )
        When I try to spend Z tokens (X>Z>Y)
        Then the operation should fail
        And no tokens should change owner
        And the allowance should stay the same

Feature: USD / ETH / Token exchange ratio
As a token issuer
I want to issue tokens on a fix USD rate
So that the token price is not affected by the ETH exchange rate

    Scenario: Setting the exchange rate
        Given that I am the MIA
        When I set the exchange ration
        Then it should override the previous rate

    Scenario: Setting the exchange rate as a regular user
        Given that I am not the MIA
        When I set the exchange ration
        Then the operation should fail

Feature: Purchasing tokens
As an investor
I want to purchase tokens
So that I can trade them later

    Scenario: Purchasing tokens
        Given that I have an ethereum account
        And there is X remaining tokens to sale
        And the exchange rate is N token per ETH
        When I purchase tokens for Y ETH where N * Y <= X
        Then I should receive N * Y tokens
        And the remaining tokens should be X - N * Y

    Scenario: Purchasing too much tokens
        Given that I have an ethereum account
        And there is X remaining tokens to sale
        And the exchange rate is N token per ETH
        When I purchase tokens for Y ETH where N * Y > X
        Then I should receive X tokens
        And I should receive the remaining (Y - X)/X ETH
        And the remaining tokens should be 0

    Scenario: Changing fund collector
        Given that I changed the fundCollector from the original value
        When someone purchases tokens
        Then the price of the tokens should transfer to the new address
        And the original address should receive nothing

Feature: Issuing tokens manually
As an investor
I want to purchase tokens for USD
So that I don't have to buy ETH

    Scenario: Issuing tokens
        Given that I am the MIA
        And there is X remaining tokens to sale
        And the exchange rate is N token per ETH
        When I issue tokens for Y tokens to an address, where N * Y <= X
        Then the address should receive N * Y tokens
        And the remaining tokens should be X - N * Y

    Scenario: Issue too many tokens
        Given that I have an ethereum account
        And there is X remaining tokens to sale
        And the exchange rate is N token per ETH
        When I issue tokens for Y tokens to an address, where N * Y <= X
        Then the address should receive X tokens
        And the remaining tokens should be 0

    Scenario: Issuing tokens as regular user
        Given that I am not the MIA
        When I issue tokens for an address
        Then the operation should fail

Feature: Minimum token amount
As a token issuer
I want to set a minimum token amount MIN
So the sale is much more efficient, because the price of the KYC will be small compared to the investments.

    Scenario: Buying less than the minimum amount
        Given that I am an investor
        When I make an otherwise valid purchase of X tokens where X < MIN
        Then the operation should fail

Feature: White list
As an organizer
I want to have a white list
So that only verified investors can own my tokens

    Scenario: Purchase tokens without white list approval
        Given any account
        When it is not on the white list
        Then they shouldn't be able to purchase tokens

    Scenario: Purchase tokens when on white list
        Given an account present on the white list
        When they try to purchase tokens
        Then they should succeed

    Scenario: Transferring tokens to white listed account
        Given that the recipient is white listed
        When I make a valid transfer to the recipient
        Then the operation should succeed

    Scenario: Transferring token to non white listed account
        Given that the recipient is not white listed
        When I make an otherwise valid transfer to the recipient
        Then the operation should fail

    Scenario:Transfer from token to non white listed account
        Given that the recipient is not white listed
        When I make an otherwise valid transfer from an other account to the recipient
        Then the operation should fail

    Scenario: Transfer approved tokens to white listed account
        Given that the account have enough tokens
        And I have allowance to spend
        And the recipient is white listed
        When I transfer tokens
        Then the recipient should receive the tokens

    Scenario: Transfer not approved tokens to white listed account
        Given that the account have enough tokens
        And I don't have allowance to spend
        And the recipient is white listed
        When I transfer tokens
        Then the recipient should receive the tokens
```

```
Feature: White list administration
As a white list administrator
I want to add members to the white list
So that they can purchase tokens

    Scenario: Adding member to white list as a WLA
        Given that I am the white list administrator
        When I add a member to the white list
        Then the member should be white listed

    Scenario: Adding member to white list as anyone else
        Given that I am not the white list administrator
        When I try to add a member to the white list
        Then the transaction should fail

    Scenario: Changing the WLA
        Given that I am the token owner
        When I change the white list administrator
        Then the white list administrator should be changed

Feature: Secondary market
As the administrator
I want to be able to open the secondary market (only once)
So that people can start trading with the token

    Scenario: Transfer tokens on closed market
        Given that the secondary market is closed
        When a user tries to transfer tokens
        Then the transaction should fail

    Scenario: Transfer token on open market
        Given that the market is open
        And the user has enough tokens
        And the recipient is whitelisted
        When the user transfers tokens to the recipient
        Then the recipient should receive the tokens
        And the users balance should be decreased with the transferred amount

    Scenario: TransferFrom tokens on closed market
        Given that the secondary market is closed
        And I have allowance to spend tokens
        When a I try to transferFrom tokens
        Then the transaction should fail

    Scenario: TransferFrom token on open market
        Given that the market is open
        And the user has enough tokens
        And the recipient is whitelisted
        And I have allowance to spend tokens
        When a I try to transferFrom tokens
        Then the recipient should receive the tokens
        And the users balance should be decreased with the transferred amount

    Scenario: Opening market
        Given that the secondary market is closed
        When the administrator opens market
        Then the secondary market should be opened

Feature: Emergency mode
As the token owner
I want to be able do declare and cancel emergency
So that no one can trade any tokens when it's an emergency

    Scenario: Turning on emergency mode
        Given that I am the owner
        And the emergency mode is off
        When I declare emergency
        Then the transaction should succeed

    Scenario: Turning on emergency mode as non owner
        Given that I am not the owner
        And the emergency mode is off
        When I declare emergency
        Then the transaction should fail
    Scenario: Token purchase in emergency mode
        Given that the emergency mode is declared
        When any user tries to transfer tokens
        Then the transaction should fail

    Scenario: Turning off emergency mode
        Given that the emergency mode is declared
        When the owner turns off the emergency mode
        Then future valid transactions should succeed

Feature: Forbid manager accounts to receive tokens
As a token owner
I want to ban the manager accounts from token trading
So that investors trust the ICO more(?)

    Scenario: Forbid token manager accounts to trade or purchase tokens
        Given that I am the white list manager
        When I try to transfer to, transferFrom to, or issueToken to any of the managers or the owner to the white list
        Then the operation should fail

Feature: TSA management
As a token owner
I want to manage the TSA account
So that I can change it when necessary or it has been compromised or lost

    Scenario: Changing the TSA
        Given that I am the token owner or the TSA
        When I change the TSA account
        Then the new account should be the TSA
        And the old TSA should loose their privileges

    Scenario: CHanging the TSA as a regular user
        Given that I am not the token owner nor the TSA
        When I change the TSA account
        Then the operation should fail

Feature: MIA management
As a token owner
I want to manage the MIA account
So that I can change it when necessary or it has been compromised or lost

    Scenario: Changing the MIA
        Given that I am the token owner or the MIA
        When I change the MIA account
        Then the new account should be the MIA
        And the old MIA should loose their privileges

    Scenario: Changing the MIA as a regular user
        Given that I am not the token owner nor the MIA
        When I change the MIA account
        Then the operation should fail

Feature:Secondary market
As the TSA
I want to close the token sale and open the secondary market
So that token owners can trade their tokens by their own

    Scenario: Opening the secondary market manually
        Given that I am the TSA and the token sale is running
        When I close the token sale
        Then the token sale should close
        And the secondary market should open
        Then the purchase of tokens should fail
        And the remaining tokens should transferred to the token owner
        And the TSA and MIA account should be set to the owner

    Scenario: Opening the secondary market automatically
        Given that the token sale is running
        When the last token is purchased or issued manually
        Then the token sale should close
        And the secondary market should open
        And the remaining tokens should transferred to the token owner
        And the TSA and MIA account should be set to the owner

    Scenario: Opening the secondary market as a regular user
        Given that I am not the TSA
        When I try to open the secondary market
        Then the operation should fail

    Scenario: Purchasing token in closed token sale
        Given that the token sale is closed
        When a user tries to purchase tokens
        Then the operation should fail

    Scenario: Issuing token in closed sale
        Given that the sale is closed
        When the TSA issues tokens manually
        Then the operation should fail

    Scenario: Trading tokens in open secondary market
        Given that the secondary market is open
        When I make an otherwise valid transfer
        Then the transfer should be successful

    Scenario: Trading tokens in closed secondary market
        Given that the secondary market is open
        When I make an otherwise valid transfer
        Then the transfer should fail
```