***DOC\_174\_REV\_C\_ASSESMENT AND IV FRONT SHEET\_INDIVIDUAL***

***CRITERIA\_CLASSTER.docx***

**ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)**

**(Note: This version is to be used for an assignment brief issued to students via Classter)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Title** | **BSc Software/Multimedia Development** | | | **Lecturer Name & Surname** | **Enrico Zanardo** | | |
| **Unit Number & Title** | | **Blockchain-ITBCK-60602101** | | | | | |
| **Assignment Number, Title**  **/ Type** | | Assignment 2 | | | | | |
| **Date Set** | | 05.01.2022 | **Deadline Date** | 28.01.2022 | | | |
| **Student Name** |  | | **ID Number** |  | | **Class / Group** |  |

|  |  |
| --- | --- |
| **Assessment Criteria** | **Maximum Mark** |
| *KU2.1: Show how to build decentralized applications..* | 5 |
| *AA2.3: Demonstrate techniques to read data from a smart contract.* | 7 |
| *AA2.5: Examine client-side techniques for a decentralized application.* | 7 |
| *AA3.1: Examine how patterns for smart contract programming are used.* | 7 |
| *KU3.2: Explain how to make use of external storage solutions.* | 5 |
| *SE2.6: Develop a working system with multiple functionalities* | 10 |
| *SE3.3: Develop a system using smart contract tokens.* | 10 |
| *KU4.1: Show the process of conducting unit testing for an error free decentralized application.* | 5 |
| *KU4.2: Present a well-tested application ready for release.* | 5 |
| **Total Mark** | 61 |

* This assignment brief has been approved and released by the Internal Verifier through Classter.
* Assessment marks and feedback by the lecturer will be available online via Classter (Http://mcast.classter.com) following release by the Internal Verifier
* Students submitting their assignment on Moodle/Turnitin will be requested to confirm online the following statements:

**Student’s declaration prior to handing-in of assignment**

* + I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy

**Student’s declaration on assessment special arrangements**

* + I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.
  + I declare that I refused the special support offered by the Institute.

**Notes to Students:**

***MCAST Controlled and approved document Unauthorised copying or communication strictly prohibited***



**Assignment 2**

# Blockchain Technologies - MCAST

## General Guidelines

Any reference from class notes, books or internet are allowed;

Plagiarism is strictly prohibited and will be penalized in accordance with the college’s disciplinary procedures;

All tasks should be attempted;

Deadline to submit the assignment is on **28.01.2022**; Assignment total marks: **61**;

# Project

Develop a complete Dapp from scratch for the Ethereum ecosystem with the following properties:

Define a personal domain application or keep the one used during the lectures [X-Men]; Define a proper Token that will be used to buy/sell domain objects [X-Men];

Define a skeleton of smart-contracts that takes in inheritance standard contracts such as openzeppeling ERC20 smart contract;

Develop custom compile and deploy functions that will compile and deploy a smart contract to Rinkeby Test network;

Build a local environment that use Truffle and Ganache-cli to interact with your smart contracts;

Develop a command line interface (CLI) that will help the user to run the unit tests, compile and deploy the smart contracts;

Develop a dedicated frontend application using vanilla javascript, React/Vue framework that will allow the users to play with the Dapp and connect your application to Metamask plug in;

The frontend application must interate through the blockchain (locally) with at least two functions;

Use as much as possible the patters seen during the course to write smart contracts using Solidity SDK version 0.8.0 or later;

Write a README.md file that describe all the required steps that are needed in order to run properly you Dapp on a local environment;

## Expected output:

Work with a git repository!!

Submit (on Moodle platform) only one zip file with all the code,

***[YOURFIRSTNAME\_YOURLASTNAME].zip***.

Documentation should be part of the source code.

* **Define the difference between storing variables on memory or on storage**

The major difference is that storing o the memory when the user turns off the pc the data is lost while on a storage the data will be kept and reused unless the storage device gets corrupted, hacked or the data is outdated. In theory if the information is not important sore on the memory other wise store the information on a digital/physical storage device.

|  |  |
| --- | --- |
| **KU2.1** | **Show how to build decentralized applications.** |
| To achieve KU2.1 [5 marks]: | |
| a) Setup a skeleton project locally that use custom javascript/typescript files or third party libraries (such as Truffle and Ganache) that compile and deploy a given smart contract written in solidity programming language [5 marks]; | |

|  |  |
| --- | --- |
| **AA2.3** | **Demonstrate techniques to read data from a smart contract.** |
| To achieve AA2.3 [7 marks]: | |
| a) Develop a unit test that read and use the **public** variables that are declared inside the smart contract [3.5 marks]; | |
| b) Develop a unit test that read and use the **private** variables that are declared inside the smart contract [3.5 marks]; | |

|  |  |
| --- | --- |
| **AA2.5** | **Examine client-side techniques for a decentralized application.** |
| To achieve AA2.5 [7 marks]: | |
| a) Develop a simple frontend web application using Vanilla Javascript, React/Vue frameworks and call a method (function) defined into a smart contract [7 marks]; | |

|  |  |
| --- | --- |
| **AA3.1** | **Examine how patterns for smart contract programming are used.** |
| To achieve AA3.1 [7 marks]: | |
| a) Define and use mapping objects to store objects that related to each other. [2 marks]; | |
| b) Define and use structs to collect objects properties [3 marks]; | |
| c) Define and use modifiers to generate appropriate filters (i.e. isOwner) [2 marks]; | |

|  |  |
| --- | --- |
| **KU3.2** | **Explain how to make use of external storage solutions.** |
| To achieve KU3.2 [5 marks]: | |
| a) Define the difference between storing variables on memory or on storage [2 marks]; | |
| b) Utilize the memory being included in the smart contract appropriately based on the kind of the variable created [3 marks]; | |

|  |  |
| --- | --- |
| **KU4.1** | **Show the process of conducting unit testing for an error free decentralized application.** |
| To achieve KU4.1 [5 marks]: | |
| a) Build a proper unit test for each method that is defined into the smart contract [5 marks]; | |

|  |  |
| --- | --- |
| **KU4.2** | **Present a well-tested application ready for release.** |
| To achieve KU4.2 [5 marks]: | |
| a) The frontend application must execute the business logic according to the logic defined into the smart contract [2 marks]; | |
| b) The deploy functionality must be developed to enable the backend to deploy the smart contract to the Rinkeby Test network. [3 marks]; | |

|  |  |
| --- | --- |
| **SE3.3** | **Develop a system using smart contract tokens.** |
| To achieve SE3.3 [10 marks]: | |
| a) Define a custom token that will be used inside the distributed application [5 marks]; | |
| b) Define certain specific actions that will distribute generated tokens to users engaging with the distributed application via the frontend. [5 marks]; | |

|  |  |
| --- | --- |
| **SE2.6** | **Develop a working system with multiple functionalities.** |
| To achieve SE2.6 [10 marks]: | |
| a) The frontend application must connect with Metamask in order to process the actions/events made by the developed application in the **Ganache** Test network [5 marks]; | |
| b) The frontend application must connect with Metamask in order to process the actions/events made by the developed application in the **Rinkeby** Test network [5 marks]; | |