Blockchain Proof-of-Concept proposal

*Blockchain Development individual assignment.*

[Insert specific use-case topic]

Prepared by: [Student name]

Due date: 18 June, 18:00 Tbilisi time.

|  |  |
| --- | --- |
|  | Dear students, your task is to create a proof-of-concept blockchain solution for a specific use-case.  Imagine you are a blockchain developer and you need to convince your client/boss/colleagues that a business problem (specific use-case) can be solved using smart contracts. To do that, you will create a proof-of-concept dApp demonstrating the basic functionality that solves that specific problem.  For that purpose, you will need to do the following:   1. Choose a use-case of your interest (voting, booking rooms, a game, buying IP rights, etc..) that you believe is suitable for a blockchain solution. 2. Use this template document to create a short description of your project and provide the reasoning behind your choice of use-case as well as describing the smart contract solution. 3. Design and create a smart contract solution that implements your chosen use-case, using Solidity. The smart contract should do at least the following:  * Changing the state of the ledger (state fields) * To call other smart contracts using both functions: call() and ‘named call’ * To be able to receive cryptocurrency (ETH) and send crypto using both functions: transfer() and send() * To implement function modifiers with appropriate restrictors * To use both structures: array and mapping * Emitting events * Using msg.sender and msg.value * Explicitly declare constructor(), fallback() and receive() functions * Try to conform to best practices and avoid security risks   Use all the knowledge you gained in order to write your smart contract(s) as best as you can.   * Additionally, after your smart contract is complete, add a special function with the following signature:   //can be invoked only by the contract owner who deployed it function projectSubmitted(string memory \_codeHash, string memory \_authorName, address \_sendHashTo) external onlyOwner  This function should send the SHA256 hash of your solidity file to a special smart contract. Successfully sending your hash code using this function will mean you submitted your work officially! The address of this special smart contract will be provided later. In case of more than one smart contracts, put them all in the same solidity file and then hash that file.  Win10 natively supports hashing of files. You can use the following command in command prompt:  certutil -hashfile C:\Users\......\ExampleSmartContract.sol SHA256  Provide the absolute path to the file, and no spaces in the name of the solidity file.  1. Deploy your smart contract solution on the Ropsten test network. 2. Invoke your special projectSubmitted() function. 3. Upload your 2 files (this document coverted to pdf and the .sol file with your smart contracts) in classter. When we evaluate your projects, the uploaded solidity file will be hashed and the resulting hash will be compared to the hash sent to the special smart contract using the projectSubmitted() function of your smart contract. If they are the same, that will mean that the solidity files are identical and have not been changed. 4. Drink a beer to a job well done 😊 |
|  | Remember, this is a proof-of-concept solution (prototype), it does not need to be full of unnecessary complex functionality. Rather, it must demonstrate the basic workflow for your specific problem/case and your ability to describe basic solutions using Solidity programming language according to the above guidelines.  This document should be brief, but make sure it looks formal and has a final and “complete” feel, and not look like just a collection of notes. You can add more information if you feel it will provide a better description of your solution.  Content in [ ] brackets should be replaced with relevant content for your use-case.  Once your document is ready, you can delete these (and the rest) info panels. |

Dear Mr./Mrs. [Client],  
  
We would like to present a smart contracts-based blockchain solution to the discussed use-case. Please get acquainted with the proposed prototype to understand the structure and process of our solution below.

# Summary

|  |  |
| --- | --- |
|  | Provide a brief summary of the chosen use-case, its relevance, and argumentation for using smart contracts to solve it. |

[Brief introduction into the use case and the motivation behind the current smart contract proposal]

## The Solution

|  |  |
| --- | --- |
|  | Brief explanation of the high-level benefits of using blockchain smart contracts for your specific use case. |

* General benefit 1: Recommendation # 1 statement
* General benefit 2: Recommendation # 2 statement
* …

# The blockchain solution

## Smart contracts specification

|  |  |
| --- | --- |
|  | Give a specification-style description of your smart contracts. The structure of the smart contracts should be clear from a glance. Use this style:  contract Contract1 {  fields  events  modifiers  constructor    list of all function signatures/prototypes ; Brief explanation what each function does. }  contract Contract2 {  fields  events  modifiers  constructor    list of all function signatures/prototypes ; Brief explanation what each function does. } |
|  |  |

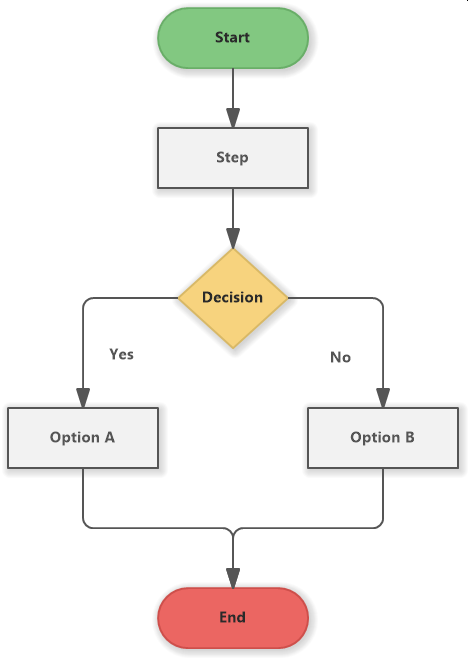
## Smart contracts explanation

|  |  |
| --- | --- |
|  | Give an explanation of what each smart contract does as part of the entire solution. Eg. Contract1 handles the booking process, while Contract 2 manages customer data… It should be clear the purpose of each smart contract in delivering the final results. |

* Contract1 is used to…
* Contract2 is used to…

## Smart contracts process flow

|  |  |
| --- | --- |
|  | Provide a diagram showing how the process flows through the different components of the solution. Interaction between users and smart contracts (and between contracts themselves) should be articulated. |



**Example Flowchart 1**. Name of example flowchart.

[Description of Flowchart 1 and what the data means.]

# Expected business results

|  |  |
| --- | --- |
|  | Briefly Describe the results expected from the project and why your approach will achieve those results. |

We expect our proposed solution to Client’s Company’s requirements to provide the following results:

## Technical Benefits

* Result 1. Detailed description of desired result and how it is achieved via the proposed solution
* Result 2. Detailed description of desired result and how it is achieved via the proposed solution
* Result 3. Detailed description of desired result and how it is achieved via the proposed solution

## Other Benefits

|  |  |
| --- | --- |
|  | Use this section to describe less tangible benefits such as increased morale or improved customer satisfaction. Any other benefits relevant to the use case should be described here. You can ignore this section if you feel it is not necessary for your use-case. |

* Result 1. Detailed description of desired result and how it is achieved via the proposed solution
* Result 2. Detailed description of desired result and how it is achieved via the proposed solution

# Conclusion

|  |  |
| --- | --- |
|  | Wrap up summarizing the problem, the need of a solution, and the benefits of the proposed blockchain solution. Include potential next steps for upgrading after implementing the blockchain solution to expand the positive results. Think of long-term opportunities. |

Thank you for your consideration,

[Name of solution proposal author]