Text

Description automatically generated

**KULLIYYAH OF INFORMATION COMMUNICATION & TECHNOLOGY**

**SEMESTER 1 2021/2022**

**BLOCKCHAIN AND APPLICATION**

**(CSCI 4312)**

**SECTION 01**

**REPORT:**

***Project Part 2***

**PREPARED BY:**

|  |  |
| --- | --- |
| **NAME** | **MATRIC NO.** |
| MUHAMMAD AZRI BIN ABDUL RAHIM | 1812031 |
| MUHAMMAD OWISH BIN MUHAMMAD RIZAL | 1816401 |
| IZZUL ISMAIL BIN IBRAHIM | 1817963 |

**LECTURER**

Dr. Norzariyah Binti Yahya

Full Repositories for Blockchain Project

<https://github.com/muhdazri1412/blockchain_project>

Contribution by parts

* Web user interface and Development (Azri)
* Unit-Testing and Smart Contracts (Owish)
* Migration, Deploy and Report (Izzul)

File

* Test.js

Code for 3 type of unit testing:

- Good day behavior (no unusual transaction)

- Check for Money Laundering (deposit exceeding threshold)

- Possible Money Laundering (smart contract balance exceeding threshold)

* Index.html

Web user interface for smart contract.

Code

1. Test.js

// const fs = require('fs');

// const contract = JSON.parse(fs.readFileSync('./build/contracts/transactiondetector.json', 'utf8'));

// console.log(JSON.stringify(contract.abi));

// var contractAddress = 0x96D6DE9D88f5747e19c9AFd91288BdA674528238;

// var abi = contract.abi;

// var ClientReceipt = new web3.eth.Contract(abi);

// var clientReceiptContract = ClientReceipt.at(contractAddress);

// var event = clientReceiptContract.Deposit(function(error, result) {

// if (!error)console.log(result);

// });

const transactiondetector = artifacts.require("transactiondetector");

contract("Check for transaction", function(accounts){

it("DEPOSIT 5 ether from account 1", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[1], value: 5 \* 10\*\*18});

});

it("DEPOSIT 5 ether from account 2", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[2], value: 5 \* 10\*\*18});

});

it("Check for transaction more than threshold", async() =>{

const NGO = await transactiondetector.deployed();

let hugeTransactors, amountHuge = await NGO.getHugeTransactors();

console.log(hugeTransactors);

console.log(amountHuge);

});

it("Check the balance if more than threshold and display potential launderers", async() =>{

const NGO = await transactiondetector.deployed();

let potential\_launderers = await NGO.getpotentialLaunderers();

console.log(potential\_launderers);

});

it("Smart Contract Balance", async() =>{

const NGO = await transactiondetector.deployed();

let scBalance = await NGO.getBalance();

scBalance = Number(scBalance);

console.log(scBalance);

});

});

contract("Check for money laundering (deposit exceeding threshold)", function(accounts){

it("DEPOSIT 15 ether from account 3", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[3], value: 15 \* 10\*\*18});

});

it("DEPOSIT 15 ether from account 4", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[4], value: 15 \* 10\*\*18});

});

it("Check for transaction more than threshold", async() =>{

const NGO = await transactiondetector.deployed();

let hugeTransactors, amountHuge = await NGO.getHugeTransactors();

console.log(hugeTransactors);

console.log(amountHuge);

});

it("Check the balance if more than threshold and display potential launderers", async() =>{

const NGO = await transactiondetector.deployed();

let potential\_launderers = await NGO.getpotentialLaunderers();

console.log(potential\_launderers);

});

it("Smart Contract Balance", async() =>{

const NGO = await transactiondetector.deployed();

let scBalance = await NGO.getBalance();

scBalance = Number(scBalance);

console.log(scBalance);

});

});

contract("POSSIBLE MONEY LAUNDERING (SMART CONTRACT BALANCE EXCEED THRESHOLD)", function(accounts){

it("DEPOSIT 9 ether from account 1", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[1], value: 9 \* 10\*\*18});

});

it("WITHDRAW 8 ether to account 2", async () => {

const NGO = await transactiondetector.deployed();

NGO.withdraw(8, {from: accounts[2]});

});

it("DEPOSIT 6 ether from account 2", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[2], value: 6 \* 10\*\*18});

});

it("DEPOSIT 25 ether from account 1", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[1], value: 25 \* 10\*\*18});

});

it("WITHDRAW 2 ether to account 2", async () => {

const NGO = await transactiondetector.deployed();

NGO.withdraw(2, {from: accounts[2]});

});

it("DEPOSIT 30 ether from account 3", async () => {

const NGO = await transactiondetector.deployed();

NGO.deposit({from: accounts[3], value: 30 \* 10\*\*18});

});

it("Check for transaction more than threshold", async() =>{

const NGO = await transactiondetector.deployed();

let hugeTransactors, amountHuge = await NGO.getHugeTransactors();

console.log(hugeTransactors);

console.log(amountHuge);

});

it("Check the balance if more than threshold and display potential launderers", async() =>{

const NGO = await transactiondetector.deployed();

let potential\_launderers = await NGO.getpotentialLaunderers();

console.log(potential\_launderers);

});

it("Smart Contract Balance", async() =>{

const NGO = await transactiondetector.deployed();

let scBalance = await NGO.getBalance();

scBalance = Number(scBalance);

console.log(scBalance);

});

});

1. Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" href="src/image/ngo.png">

<title>Home</title>

<!-- Bootstrap -->

<link href="src/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-2"></div>

<div class="col-md-8">

<br>

<div class="row">

<div class="col-md-4"></div>

<div class="col-md-4"><img src="" alt=""></div>

<div class="col-md-4"></div>

</div>

<br>

<h1 class="text-center">suspicious transactions detector</h1>

<br/>

</div>

</div>

<div class="row">

<div class="col-md-6">

<p>Account Balance : <b id="bal"> </b><b>ether</b></p>

</div>

<div class="col-md-6"></div>

</div>

<div class="row">

<div class="col-md-4">

<label >Insert Withdraw Amount : </label>

<input type="text" class="form-control"><br>

<button class="btn btn-primary">Withdraw</button>

</div>

<div class="col-md-8"></div>

</div>

<br>

<div class="row">

<div class="col-md-4">

<label for="">Insert Deposit Amount : </label>

<input type="text" class="form-control"><br>

<button class="btn btn-primary">Deposit</button>

</div>

<div class="col-md-8"></div>

</div>

<br><br>

</div>

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<!-- Include all compiled plugins (below), or include individual files as needed -->

<script src="src/js/bootstrap.min.js"></script>

<script src="src/js/web3.min.js"></script>

<script src="src/js/truffle-contract.js"></script>

<script src="src/js/app.js"></script>

</body>

</html>