**Sprint 2 MediCoin Token Documentation**

**Sprint backlog:**

1. Create storage for patient data on Blockchain:

* Code smart contract for Storage of patient data
* Deploy contract to Ethereum blockchain

1. Integrate token with website:

* Create html for ICO page
* Connect page to deployed contract using web3js
* Use webpack to bundle JavaScript file

**MediCoin Token Use Case:**

Smart Contracts are interacted via the frontend allowing user to purchase token. It is necessary for the user to interact with these contracts for them to have connected their Ethereum wallet first via Metamask.

**Flow Chart:**

**Diagram

Description automatically generated**

**User story 1:**

As a user, I want to be able to buy medicoin token with my metamask wallet on the website

**Wire frame:**

**Graphical user interface

Description automatically generated**

**Test case:**

Check response when buy transaction is initiated.

**Test steps:**

* Connect wallet via Metamask prompt
* Click Buy Token
* Accept transcation via metamask prompt
* Check transcation status on metamask or wait till wallet info balance increments

**Implementation:**

**Text

Description automatically generated**

**User story 2:**

As a user, I want to be able to view my Medi-coin wallet balance.

**Test case:**

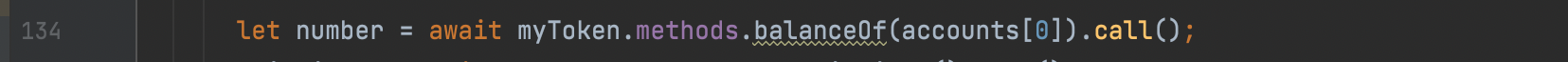
Check wallet info field after connecting metamask wallet.

**Test Steps:**

* Connect wallet via Metamask prompt after clicking ‘Connect Wallet’ button
* Check wallet info field

**Implementation:**

Number variable is equalled to the return value of the token contracts balance of function for the connected account and we set the DOM element to the string

****

****

**User story 3:**

As a user I want to be able to view the ICO remaining tokens.

**Test Case:**

Check response after metamask wallet connected.

**Tests steps:**

* Connect metamask wallet via metamask prompt

**Implementation:**

Get instances of contracts using ABI

Text

Description automatically generated

Call weiRaised method on the crowdsale contract to see how many tokens have been sold



Set DOM elements to these values

Graphical user interface, text

Description automatically generated

**Blockchain Use Case:**

The storage contract is interacted via web3js and when a user registers their information is added to the contract on the Ethereum blockchain.

**User story 1:**

As a admin, I want my users information to be stored on a blockchain

**Test case:**

When interacting with the deployed instance of the Storage contract the functions to add information to the contract is working and successful.

**Test steps:**

* Run truffle storage tests
* Check outcome

**Implementation:**

Test script is located on Gith in the tests folder of the TokenCrowdSale folder

**Sprint Retrospective Review:**

I feel like more progress could have been made in relation to integration of the storage of the patient data in the contract on the blockchain with the android app and web app even though as it was assigned to the next sprint but I ended up spending more time on understanding and tuning the integration of the token with the website for the ICO.