SignIT

SignIT is my personal project to build a proof of concept Ethereum-based Digital Signature platform. Using Blockchain, its goal is to create a system to manage a collection of agreements, that builds a user identity based on those agreements and relationships. Our GOAL is making easier and more democratic to manage your digital presence.

Getting Started

Clone this project from https://github.com/eduardopezzi/signIT_project.git On the project folder you will find the truffle project that you can find the Solidity contract and its tests. Start typing

```
npm install
```

The initial setup for running properly, you will need to run Ganache (Cli or desktop version), and copy the 12 mnemonic words and create .secret file at root folder and past the words inside the file. After that you can run truffle test command. Make sure that your Ganache is running under the port 7545.

Run the following commands to compile, deploy and test the contract

```
truffle compile
truffle migrate
truffle test
```

Before run truffle test, and after run truffle migrate, copy the contract address and past into the newDocument.js file at the folder ./signIt-React-App/src/data

Prerequisites

Npm ^6.13.4

Truffle 5.0.42

Version 2.3.0-beta

Metamask 7.7.8

IPFS

Tests

Truffle will run the following tests:

```
Contract: SignitDoc
    ✓ 1. SignitDoc should be deployed and store a Document (41ms)
```

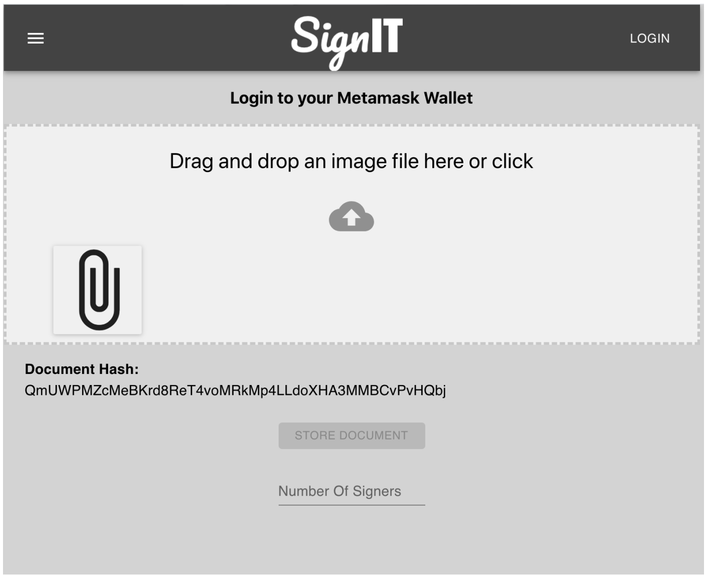
```
1970-01-19T09:01:13.361Z
     ✓ 2. get time of deployment
     ✓ 3. Verify function should recover signer address (56ms)
     ✓ 4.1. Document can be signed and event is emitted (204ms)
     ✓ 4.2. Document can NOT be signed twice by the same signer (167ms)
     ✓ 4.3. Document can be signed by Signer2 and status changed (260ms)
     ✓ 5. Document can NOT be signed, Signer already signed (121ms)
     ✓ 6. Document can NOT be signed for more than #ofSigners variable
(155ms)
     ✓ 7.1. Signature can be REVOKED by signer (126ms)
     ✓ 7.2.Document can be REVOKED (310ms)
     ✓ 7.3.Document REVOKED can NOT be signed (107ms)
```

Web Interface

Also you will find a React Web Interface to this application at the signIT-React-App folder. Open this folder and run

```
npm install
npm start
```

Also, if you want use IPFS decentrilized file system, it is necessary to install and run the server. To test this application you must have Metamask installed and setup a custom RPC to connect to your ganache project. Once everything setup, you can start using the app. First upload a document, second store the document (it will store the document information in the blockchain and the file in the IPFS network. Third, you can sign it and afterr that verify if the signature matches with signer's wallet and retrive who has signed the Document.


SignIT
LOGIN
Login to your Metamask Wallet
Drag and drop an image file here or click
STORE DOCUMENT
Number Of Signers

Document Hash:
QmUWPMZcMeBKrd8ReT4voMRkMp4LLdoXHA3MMBCvPvHQbj

Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

Authors

- **Eduardo Pezzi**

License