

# Deliverable #2

The Programmer's Manual

By:

Curatolo, Christopher

Harms, Mathew

Howse, Jason

# Requirements Documentation

## Introduction

### 1.1 Purpose:

The purpose of the documentation is to explain the requirements of the iNFT application.

### 1.2. Scope of the Product:

The iNFT app will generate NFTs (Non-Fungible Tokens) on the Ethereum network.

### 1.3. Glossary:

- Dapp - A decentralized application that uses a decentralized network (e.g. Ethereum, Bitcoin).
- Blockchain: An immutable, ledger database of transactions distributed across a network.
- Crypto wallet: A tool that manages a collection of accounts, public keys, private keys, and interacts with a blockchain network.
- Crypto Keys
  - ✧ Asymmetric cryptography involves pairs of keys.
  - ✧ Generation of Keys depends on the cryptographic algorithm.
  - ✧ The algorithm generates a private and public key
  - ✧ Depending on your use case, one key signs and encrypts messages, the other key verifies and decrypts messages.
- Public address - A hash of the account's public key. An Ethereum public address is digested from a keccak256 hash function where the public key is the input.
- Ethereum - A blockchain platform that facilitates cryptocurrency and smart contracts.
- Ether - The cryptocurrency on the Ethereum blockchain.
- Gas - The fee required to conduct a transaction on the blockchain.
- Smart contract - Self-executed code on the blockchain.
- NFT - (Non Fungible Tokens) A smart contract that has a standard, transfers ownership, and mints unique tokens of a digital asset. According to [wikipedia.org/wiki/Non-fungible\\_token](https://wikipedia.org/wiki/Non-fungible_token), "NFTs are tracked on blockchains to provide the owner with a proof of ownership that is separate from copyright."
- IPFS - (Interplanetary File System) is an immutable file system, once an item is pinned to the IPFS network it cannot be removed.

### 1.4. References:

- IEEE 830-1998

### 1.5 Overview

The rest of this document will go over the requirements of the iNFT in both general terms and more specific terms.

## General Description

### 2.1 Product perspective:

The iNFT application is an NFT minter. That is, it will create a token on the Ethereum block chain network that points to a particular file on the IPFS network. This allows users to make a claim on the authenticity of an Item posted to IPFS. It is effectively a form of copy right. Another use for this technology is users may mint deeds as a proof of property ownership. The blockchain provides the integrity of an object while the IPFS network provides the availability of the information.

### 2.2 Product functions:

The product will allow users with an Ethereum private key to Mint NFTs. Users will log onto an Ethereum network (currently only the local and Ropsten Test networks are available). Once the user logs in with their private key the user will be able to view a list of all available NFTs. If the user selects any of the NFT it will be downloaded to their local machine and the file's path can be copied. If the file is an approved file type it will be displayed by the app. The user will also either be able to type a file path into the text box at the top or select a file from the file picker. If a the file is of an approved type it will display and the user can select the mint button to post the file to IPFS and the Ethereum network

### 2.3 User characteristics:

There is only one type of user of this application. That is a person who wishes to mint NFT tokens or view minted NFT tokens. A person with a valid private key can log into the system and mint tokens. A person with a random hex string can also unlock the system as the Ethereum network only tracks private keys that have made transactions. A person using an unsecured private key may have their assets stolen. The person utilizing the system should have a very basic knowledge of how computers work particularly capable of selecting a file from the file picker.

### 2.4 General constraints:

The contract hasn't been posted to the Mainnet due to the cost. The system does require an active Ethereum network, either a local network with the following properties. The local address set to: HTTP://127.0.0.1:7545 and Chain ID/Network ID is 5777. The NFT and iNFT contracts would have to be deployed to the network as well. If an active internet connection exists then the Test network Ropsten can be posted to. The user would need to have enough ether on their account to make the required transaction. It should be noted that the larger the file type that has been chosen to mint will take longer to post to the IPFS network and retrieve from the IPFS network.

### 2.5 Assumptions and dependencies:

The only dependency for the end user would be that they are running on a Windows 10 platform. An active internet connection is required to use the Ropsten network.

## Specific Requirements Front End

### Login Block

- D1.0.0 User shall be able to log in.
- D1.1.0 The System shall set the UI height and width.
  - D1.1.1 The System shall set the UI height to 160px.
  - D1.1.2 The System shall set the UI width to 400px.
- D1.2.0 User shall be able log into an environment.
  - D1.2.1 User shall be able to choose an environment.
  - D1.2.2 The System shall configure the local environment to connect to RPC Server HTTP://127.0.0.1:7545 Network ID/Chain ID 5777.
  - D1.2.3 The System shall configure the test environment to connect to RPC Server https://ropsten.infura.io/v3/c403a4afb4f5439588595f1f242e7c75 Network ID/Chain ID 3.
  - D1.2.4 The System shall configure the main net environment to connect to https://mainnet.infura.io/v3/c403a4afb4f5439588595f1f242e7c75 Network ID/Chain ID 1.
  - D1.2.5 If no environment is chosen and the login button is clicked, the System shall pop a toast message error requesting the user to choose an environment.
  - D1.2.6 The System shall configure the current environment based on the User's environment selection.
- D1.3.0 User shall be able log in with a private key hex key.
  - D1.3.1 If an environment is selected, no key entered, and the login button is clicked, the System shall pop a toast message error requesting the user enter a private key.
  - D1.3.2 If an environment is selected, the key contains non hex characters, and the login button is clicked, the System shall pop a toast message error stating the private key is invalid.
  - D1.3.3 If the selected network is unavailable, then the System shall pop a toast error message stating the environment is not available and details will be printed to the log file.
- D1.4.0 If login is successful, then the System shall pop a toast with the current address's balance.
  - D1.4.1 If the account balance is greater than 0, then the System shall use the primary toast message format and show the balance.
  - D1.4.2 If the account balance is 0, then the System shall use the warning toast message format and warn the user that the current balance is 0.
  - D1.4.3 The System shall populate a combo box with all NFTs associated with the users address.
  - D1.4.4 The user's private key shall be held as a variable in memory.

### Main Block

- D2.0.0 The System shall instantiate the main block
- D2.1.0 The System shall expand UI height and width.
  - D2.1.1 The System shall set the UI height to 450px.
  - D2.1.2 The System shall set the UI width to 800px.
- D2.2.0 The System shall instantiate a Logout button, a browse button, a file text box, and a NFT combo box.

### Mint NFT

- D3.0.0 The User shall be able to mint a file.
- D3.1.0 The User shall be able to select a file.
  - D3.1.1 The User shall be able to write a file name into the text box.
  - D3.1.2 The User shall be able to pick a file from the file picker.
  - D3.1.2 The System shall remove the image or text displayed when the text box is updated.
- D3.2.0 The System shall display image files.
  - D3.2.1 If the file bytecode type is "webp", "jpeg", "png", "gif", or "jpg", then the System shall display the image.
- D3.3.0 The System shall display plain text files.

D3.3.1 If the file bytecode type is "txt", "html", "xml", "css", "js", "htm", or "json", then the System shall display the text.

D3.4.0 If there is any text in the file text box, then the System shall make the mint button available.

D3.4.1 If the file does not exist then, then the System shall pop a toast message error stating no such file exists.

D3.4.2 If the file does exist, the System shall attempt to pin the file to the IPFS network.

D3.4.3 If the file does not successfully pin to the IPFS network, the System shall pop a toast message error stating the file did not post to IPFS and post the error to the log file.

D3.4.4 If the file successfully pins to the IPFS network, the System shall take the hash code generated and attempt to mint a token to the Ethereum network.

D3.4.5 If the system successfully mints the token, the System shall pop a toast message primary stating Success!

D3.4.6 If the system fails to mint the token, the System shall pop a toast message error stating the item failed to mint and point error logs to the log file.

D3.4.7 The System shall populate a combo box with all NFTs associated with the user's address.

#### Get NFT

D4.0.0 The User shall be able to select any NFT from the combo box.

D4.1.0 If the user selects an NFT, the System shall download the NFT to the local system.

D4.1.1 The System shall assign a generic name to the file.

D4.1.2 The System shall use miming to determine the file type is and assign the proper extension to the file.

D4.2.0 The System shall display image files.

D4.2.1 If the file bytecode type is "webp", "jpeg", "png", "gif", or "jpg", then the System shall display the image.

D4.3.0 The System shall display plain text files.

D4.3.1 If the file bytecode type is "txt", "html", "xml", "css", "js", "htm", or "json", then the System shall display the text in a text block.

D4.4.0 The System shall post the file name to a text box.

D4.5.0 The User shall be able to copy the file path to the clipboard.

#### Transfer NFT (Not Implemented)

D5.0.0 Not Implemented.

#### Logout

D6.0.0 The user shall be able to logout.

D6.1.0 The system shall set the account value to null.

D6.2.0 The system shall set user environment to null.

D6.3.0 The system shall set to the generic environment settings.

D6.4.0 The system shall display the Login screen.

#### Logging

D7.0.0 The System shall maintain a log of important events.

D7.1.0 The System shall record important events as info events.

D7.1.1 The System shall log a record that will display the url to the item posted to IPFS.

D7.1.2 The System shall log a record the transaction hash.

D7.2.0 The System shall record system errors from the MainWindow.xaml class as error events.

D7.2.1 The System shall log a record when the environment fails to connect.

D7.2.2 The System shall log a record when an error occurs during login.

D7.2.3 The System shall log a record when the System fails to display an image.

D7.2.4 The System shall log a record when the System fails to display a plain text file

D7.2.5 The System shall log a record when the System fails to get a file from the IPFS network.

D7.2.6 The System shall log a record when the System fails to pin a file to IPFS.

D7.2.7 The System shall log a record when the System fails to mint a token to the Ethereum network.

D7.3.0 The System shall record system errors from the Ethereum\_Interact class as error events.

D7.3.1 The System shall log a record when the System fails to create a local web3 object.

D7.3.2 The System shall log a record when the System fails to create a test web3 object.

D7.3.3 The System shall log a record when the System fails to create a production web3 object.

D7.3.4 The System shall log a record when the System has an error when checking that a token is associated with a particular account.

D7.3.5 The System shall log a record when the System has an error minting the token.

D7.4.0 The System shall record system errors from the Helpers class as error events.

D7.4.1 The System shall log a record when the System fails to deserialize the JSON object selected.

D7.5.0 The System shall record system errors from the IPFS\_Interact class as error event.

D7.5.1 The System shall log a record when the System fails to write the IPFS file to local storage.

D7.5.2 The System shall log a record when the system fails to convert a stream to byte array or the mime guesser throws an error.

D7.5.3 The System shall log a record when the system fails to pin a file to IPFS.

D7.5.4 The System shall log a record when the system fails trying to delete all stored files.

## Specific Requirements Back End

- B1.0.0 The System shall deploy the NFT contract to the blockchain network
  - B1.1.0 The User shall be able to mint an NFT against the deployed contract
  - B1.1.1 The System shall be able to call the mint function of the NFT contract
  - B1.1.2 The System shall produce a unique token ID for each NFT minted
  - B1.1.3 The System shall take in a user's wallet address and the token ID for the minted NFT
  - B1.1.4 The user shall be able to connect to a deployed NFT smart contract
  - B1.1.5 The user shall receive a unique token ID mapped to their wallet address as well as the user's tokenURI after a successful mint
  - B1.1.6 The user shall be able to mint images continuously to their wallet address
  - B1.1.7 The user shall be defined as the owner of the minted NFT via the metadata
  - B1.1.8 The user shall receive the tokens for each minted NFT

## Design Documentation

Diagram 1: Front End UML

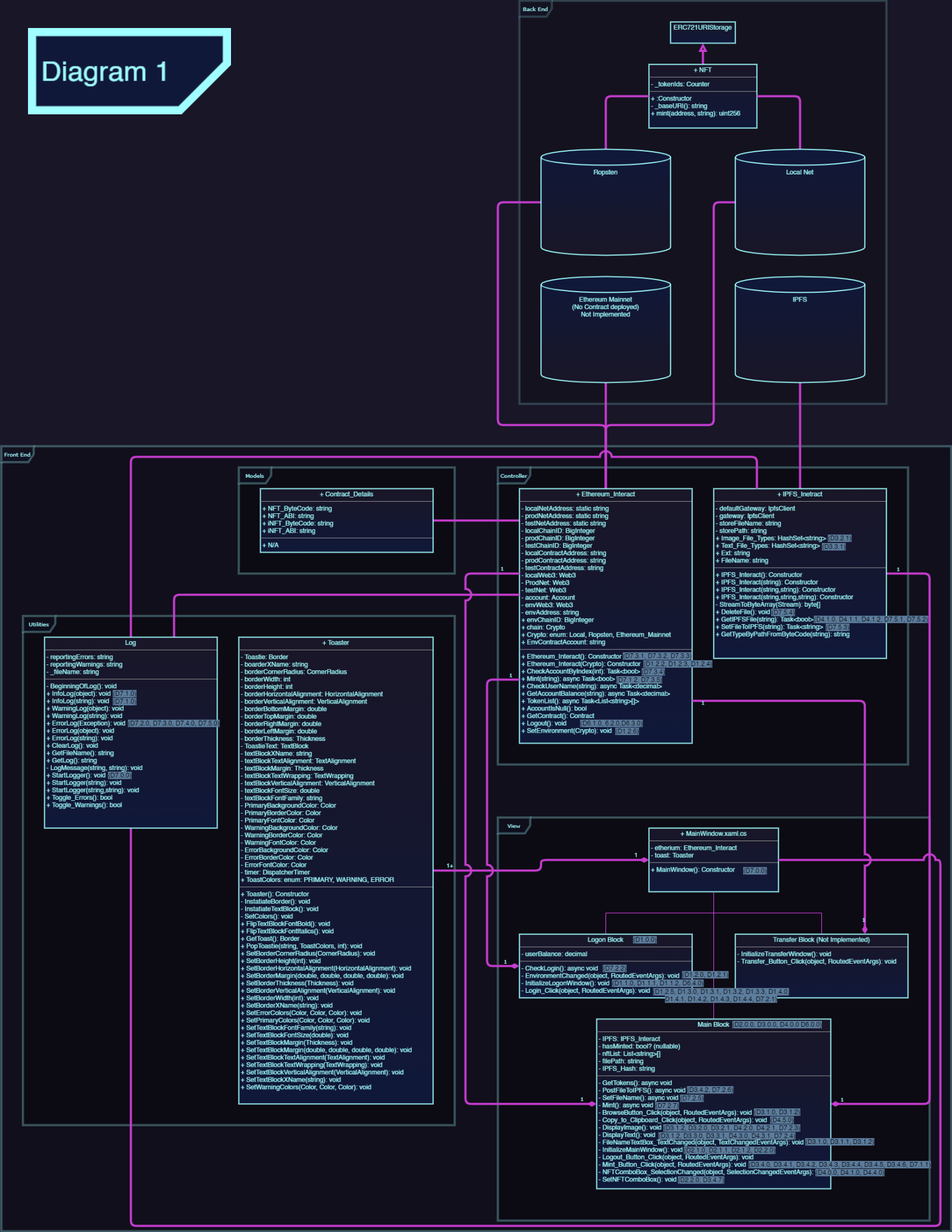
Diagram 2: Front End Flow

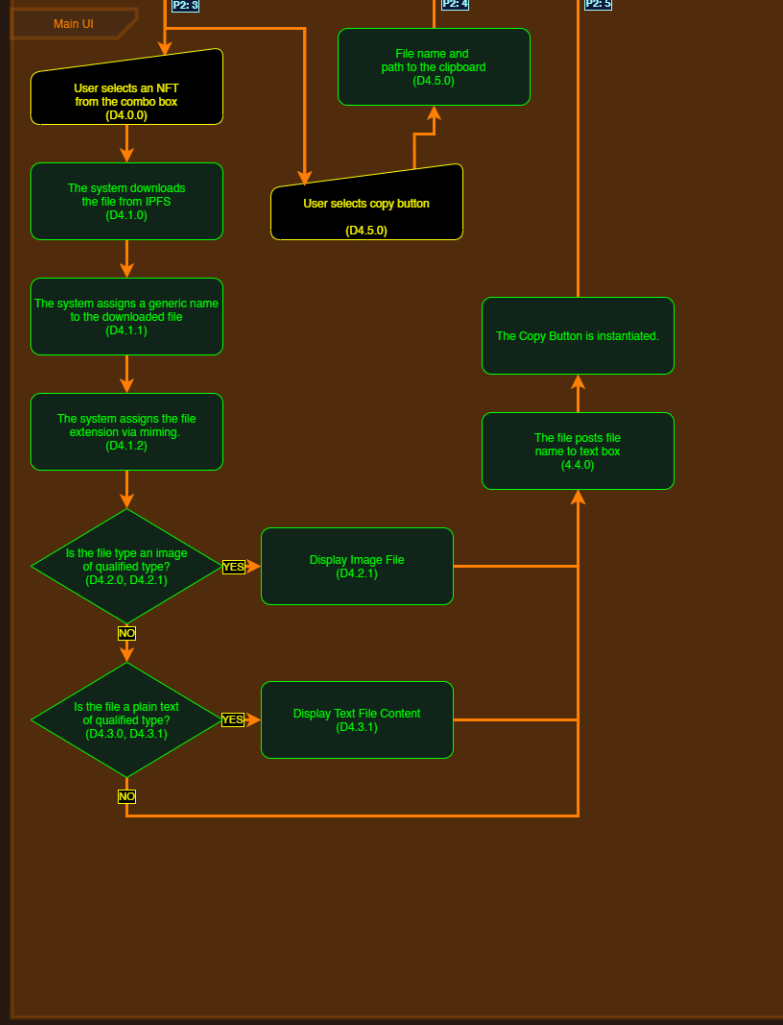
Diagram 3: Back End Class Diagram

Diagram 4: Back End Dev Flow Chart



# Diagram 1





## Diagram 2

Diagram 3

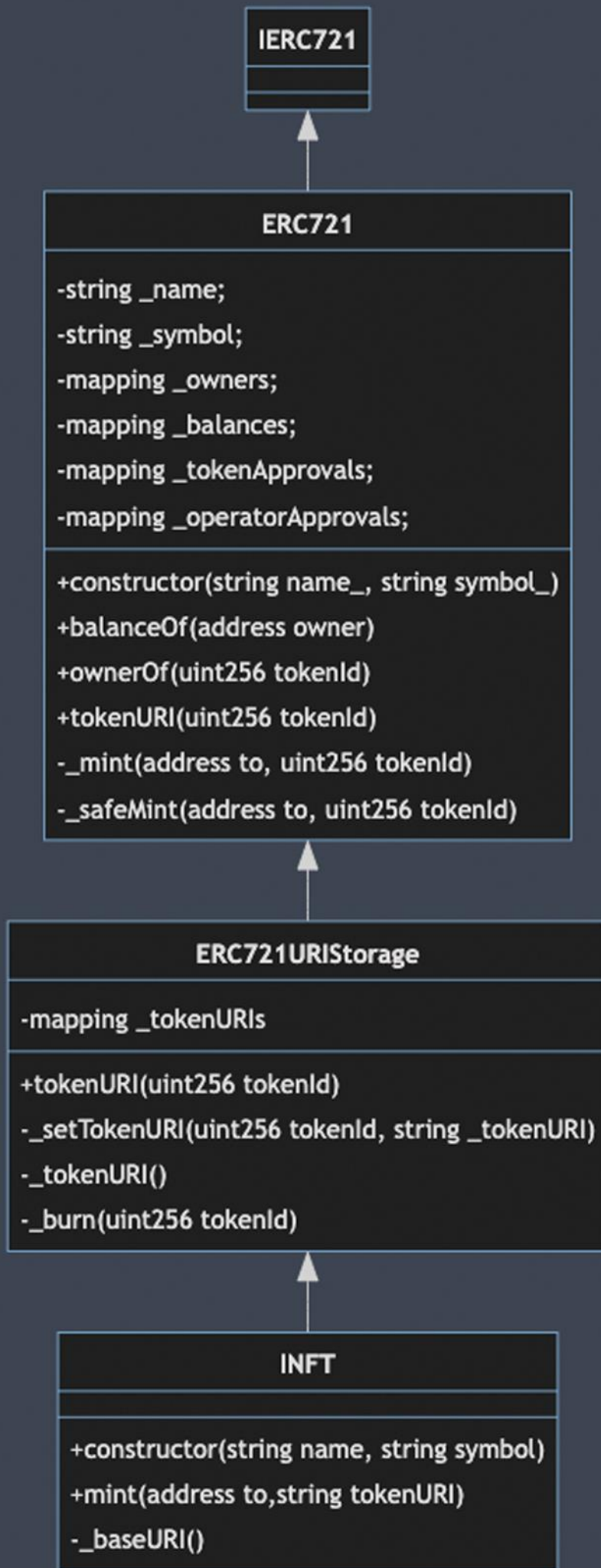


Diagram 4

