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, "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 IFPS 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 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, 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 will 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 exists then, then the System shall pop a toast message error stating no such file exists.

D3.4.2 If the file does exists, 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 files 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 will all NFTs associated with the users 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 clip board.

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 Etherium_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

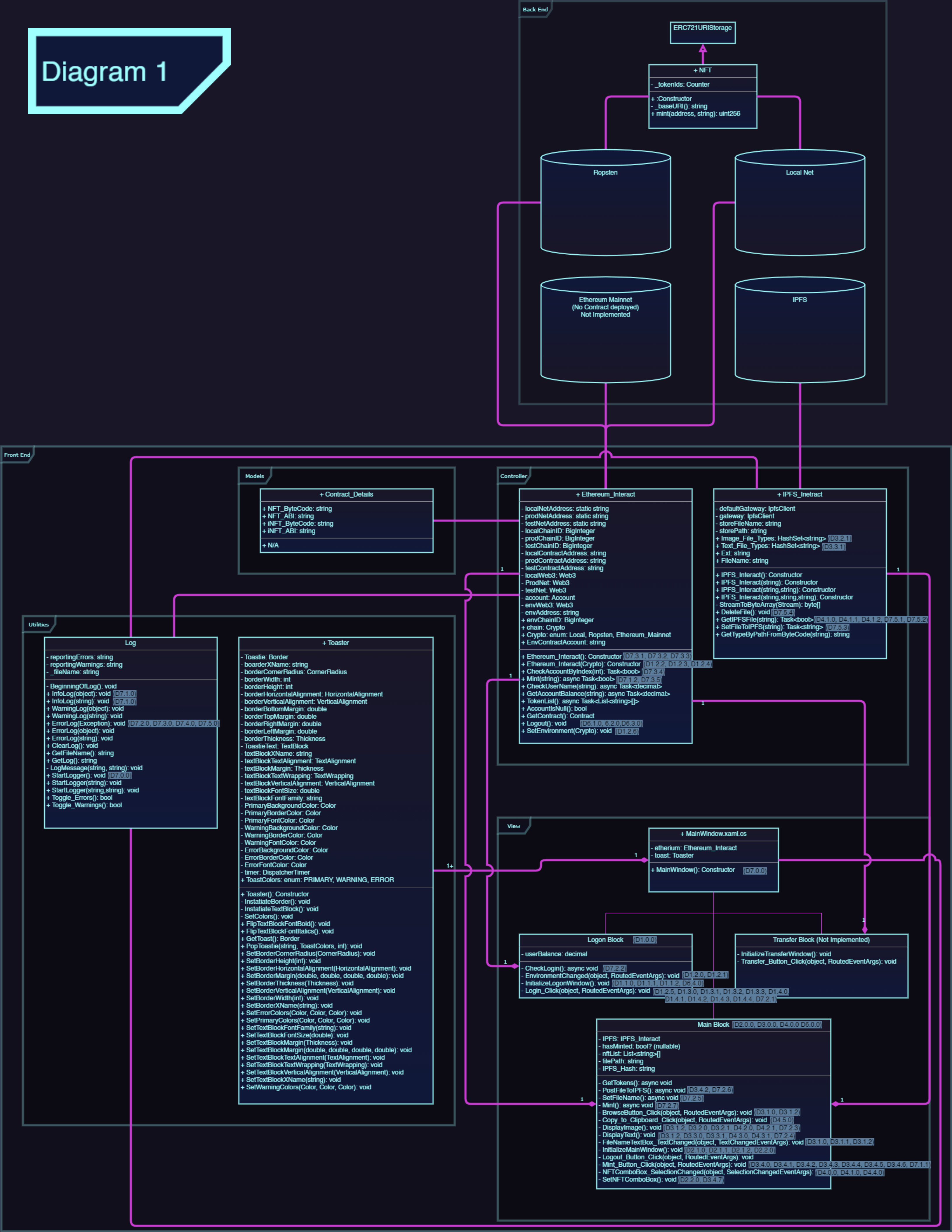
Diagram 1: Front End UML

Diagram 2: Front End Flow

Diagram 3: Back End Class Diagram

Diagram 4: Back End Dev Flow Chart

Diagram 1



```
ERC721URIStorage
```

```
+ NFT
- _tokenIds: Counter
+ :Constructor
- _baseURI(): string
+ mint(address, string): uint256
```

```
+ Contract_Details
+ NFT_ByteCode: string
+ NFT_ABI: string
+ iNFT_ByteCode: string
+ iNFT_ABI: string
+ N/A
```

```
+ Ethereum_Interact
- localNetAddress: static string
- prodNetAddress: static string
- testNetAddress: static string
- localChainID: BigInteger
- prodChainID: BigInteger
- testChainID: BigInteger
- localContractAddress: string
- prodContractAddress: string
- testContractAddress: string
- localWeb3: Web3
- ProdNet: Web3
- testNet: Web3
- account: Account
- envWeb3: Web3
- envAddress: string
+ envChainID: BigInteger
+ chain: Crypto
+ Crypto: enum: Local, Ropsten, Ethereum_Mainnet
+ EnvContractAccount: string
+ Ethereum_Interact(): Constructor (D7.3.1, D7.3.2, D7.3.3)
+ Ethereum_Interact(Crypto): Constructor (D1.2.2, D1.2.3, D1.2.4)
+ CheckAccountByIndex(int): Task<bool> (D7.3.4)
+ Mint(string): async Task<bool> (D7.1.2, D7.3.5)
+ CheckUserName(string): async Task<decimal>
+ GetAccountBalance(string): async Task<decimal>
+ TokenList(): async Task<List<string>[]>
+ AccountIsNull(): bool
+ GetContract(): Contract
+ Logout(): void      (D6.1.0, 6.2.0,D6.3.0)
+ SetEnvironment(Crypto): void (D1.2.6)
```

```
+ IPFS_Inetract
- defaultGateway: IpfsClient
- gateway: IpfsClient
- storeFileName: string
- storePath: string
+ Image_File_Types: HashSet<string> (D3.2.1)
+ Text_File_Types: HashSet<string> (D3.3.1)
+ Ext: string
+ FileName: string
+ IPFS_Interact(): Constructor
+ IPFS_Interact(string): Constructor
+ IPFS_Interact(string,string): Constructor
+ IPFS_Interact(string,string,string): Constructor
- StreamToByteArray(Stream): byte[]
+ DeleteFile(): void (D7.5.4)
+ GetIPFSFile(string): Task<bool>(D4.1.0, D4.1.1, D4.1.2, D7.5.1, D7.5.2)
+ SetFileToIPFS(string): Task<string> (D7.5.3)
+ GetTypeByPathFromByteCode(string): string
```

```
Log
- reportingErrors: string
- reportingWarnings: string
- _fileName: string
- BeginningOfLog(): void
+ InfoLog(object): void (D7.1.0)
+ InfoLog(string): void (D7.1.0)
+ WarningLog(object): void
+ WarningLog(string): void
+ ErrorLog(Exception): void (D7.2.0, D7.3.0, D7.4.0, D7.5.0)
+ ErrorLog(object): void
+ ErrorLog(string): void
+ ClearLog(): void
+ GetFileName(): string
+ GetLog(): string
- LogMessage(string, string): void
+ StartLogger(): void (D7.0.0)
+ StartLogger(string): void
+ StartLogger(string,string): void
+ Toggle_Errors(): bool
+ Toggle_Warnings(): bool
```

```
+ Toaster
- Toastie: Border
- boarderXName: string
- borderCornerRadius: CornerRadius
- borderWidth: int
- borderHeight: int
- borderHorizontalAlignment: HorizontalAlignment
- borderVerticalAlignment: VerticalAlignment
- borderBottomMargin: double
- borderTopMargin: double
- borderRightMargin: double
- borderLeftMargin: double
- borderThickness: Thickness
- ToastieText: TextBlock
- textBlockXName: string
- textBlockTextAlignment: TextAlignment
- textBlockMargin: Thickness
- textBlockTextWrapping: TextWrapping
- textBlockVerticalAlignment: VerticalAlignment
- textBlockFontSize: double
- textBlockFontFamily: string
- PrimaryBackgroundColor: Color
- PrimaryBorderColor: Color
- PrimaryFontColor: Color
- WarningBackgroundColor: Color
- WarningBorderColor: Color
- WarningFontColor: Color
- ErrorBackgroundColor: Color
- ErrorBorderColor: Color
- ErrorFontColor: Color
- timer: DispatcherTimer
+ ToastColors: enum: PRIMARY, WARNING, ERROR
+ Toaster(): Constructor
- InstatiateBorder(): void
- InstatiateTextBlock(): void
- SetColors(): void
+ FlipTextBlockFontBold(): void
+ FlipTextBlockFontItalics(): void
+ GetToast(): Border
+ PopToastie(string, ToastColors, int): void
+ SetBorderCornerRadius(CornerRadius): void
+ SetBorderHeight(int): void
+ SetBorderHorizontalAlignment(HorizontalAlignment): void
+ SetBorderMargin(double, double, double, double): void
+ SetBorderThickness(Thickness): void
+ SetBorderVerticalAlignment(VerticalAlignment): void
+ SetBorderWidth(int): void
+ SetBorderXName(string): void
+ SetErrorColors(Color, Color, Color): void
+ SetPrimaryColors(Color, Color, Color): void
+ SetTextBlockFontFamily(string): void
+ SetTextBlockFontSize(double): void
+ SetTextBlockMargin(Thickness): void
+ SetTextBlockMargin(double, double, double, double): void
+ SetTextBlockTextAlignment(TextAlignment): void
+ SetTextBlockTextWrapping(TextWrapping): void
+ SetTextBlockVerticalAlignment(VerticalAlignment): void
+ SetTextBlockXName(string): void
+ SetWarningColors(Color, Color, Color): void
```

```
+ MainWindow.xaml.cs
- etherium: Ethereum_Interact
- toast: Toaster
+ MainWindow(): Constructor    (D7.0.0)
```

```
Logon Block    (D1.0.0)
- userBalance: decimal
- CheckLogin(): async void  (D7.2.2)
- EnvironmentChanged(object, RoutedEventArgs): void (D1.2.0, D1.2.1)
- InitializeLogonWindow(): void (D1.1.0, D1.1.1, D1.1.2, D6.4.0)
- Login_Click(object, RoutedEventArgs): void (D1.2.5, D1.3.0, D1.3.1, D1.3.2, D1.3.3, D1.4.0,
D1.4.1, D1.4.2, D1.4.3, D1.4.4, D7.2.1)
```

```
Transfer Block (Not Implemented)
- InitializeTransferWindow(): void
- Transfer_Button_Click(object, RoutedEventArgs): void
```

```
Main Block (D2.0.0, D3.0.0, D4.0.0 D6.0.0)
- IPFS: IPFS_Interact
- hasMinted: bool? (nullable)
- nftList: List<string>[]
- filePath: string
- IPFS_Hash: string
- GetTokens(): async void
- PostFileToIPFS(): async void (D3.4.2, D7.2.6)
- SetFileName(): async void (D7.2.5)
- Mint(): async void (D7.2.7)
- BrowseButton_Click(object, RoutedEventArgs): void (D3.1.0, D3.1.2)
- Copy_to_Clipboard_Click(object, RoutedEventArgs): void  (D4.5.0)
- DisplayImage(): void (D3.1.2, D3.2.0, D3.2.1, D4.2.0, D4.2.1, D7.2.3)
- DisplayText(): void  (D3.1.2, D3.3.0, D3.3.1, D4.3.0, D4.3.1, D7.2.4)
- FileNameTextBox_TextChanged(object, TextChangedEventArgs): void (D3.1.0, D3.1.1, D3.1.2)
- InitializeMainWindow(): void (D2.1.0, D2.1.1, D2.1.2, D2.2.0)
- Logout_Button_Click(object, RoutedEventArgs): void
- Mint_Button_Click(object, RoutedEventArgs): void (D3.4.0, D3.4.1, D3.4.2, D3.4.3, D3.4.4, D3.4.5, D3.4.6, D7.1.1)
- NFTComboBox_SelectionChanged(object, SelectionChangedEventArgs): (D4.0.0, D4.1.0, D4.4.0)
- SetNFTComboBox(): void (D2.2.0, D3.4.7)
```

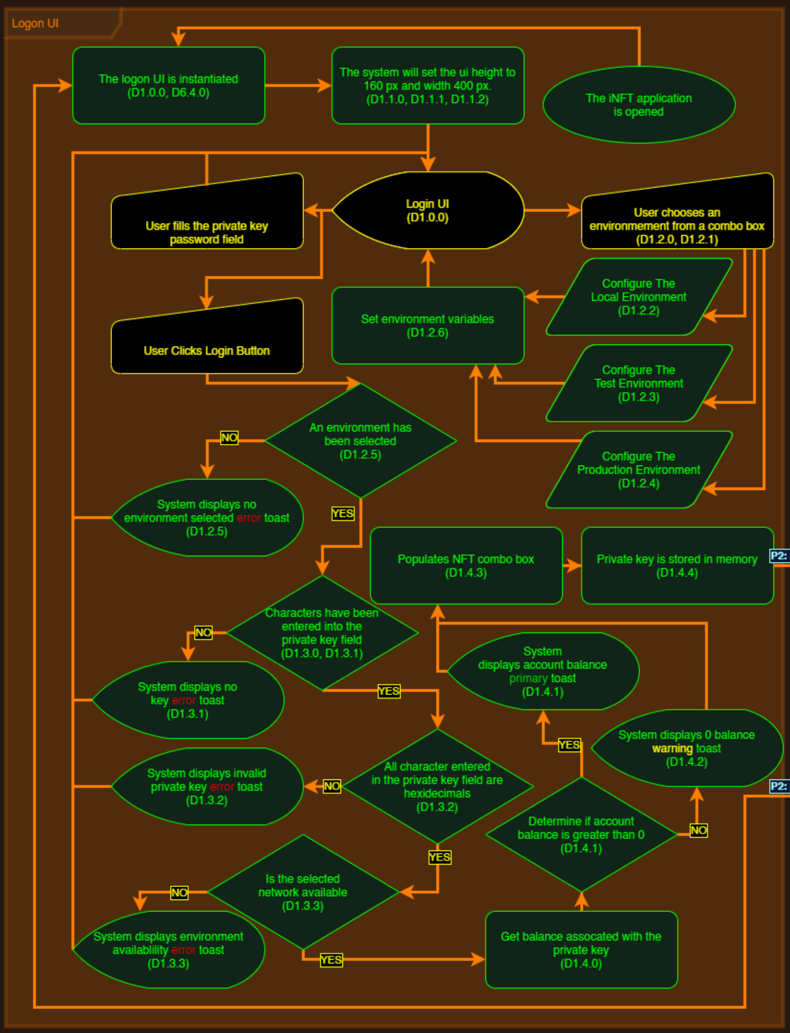

Logon UI
The logon UI is instantiated (D1.0.0, D6.4.0)
The system will set the ui height to 160 px and width 400 px. (D1.1.0, D1.1.1, D1.1.2)
The iNFT application is opened
Login UI (D1.0.0)
User fills the private key password field
User chooses an environmenment from a combo box (D1.2.0, D1.2.1)
Configure The Local Environment (D1.2.2)
Set environment variables (D1.2.6)
User Clicks Login Button
Configure The Test Environment (D1.2.3)
An environment has been selected (D1.2.5)
NO
YES
Configure The Production Environment (D1.2.4)
System displays no environment selected error toast (D1.2.5)
Populates NFT combo box (D1.4.3)
Private key is stored in memory (D1.4.4)
P2: 1
P1: 1
Characters have been entered into the private key field (D1.3.0, D1.3.1)
NO
YES
System displays no key error toast (D1.3.1)
System displays account balance primary toast (D1.4.1)
YES
System displays 0 balance warning toast (D1.4.2)
System displays invalid private key error toast (D1.3.2)
NO
All character entered in the private key field are hexidecimals (D1.3.2)
YES
Determine if account balance is greater than 0 (D1.4.1)
NO
P2: 2
P1: 2
Is the selected network available (D1.3.3)
NO
YES
System displays environment availablility error toast (D1.3.3)
Get balance assocated with the private key (D1.4.0)

Diagram 2

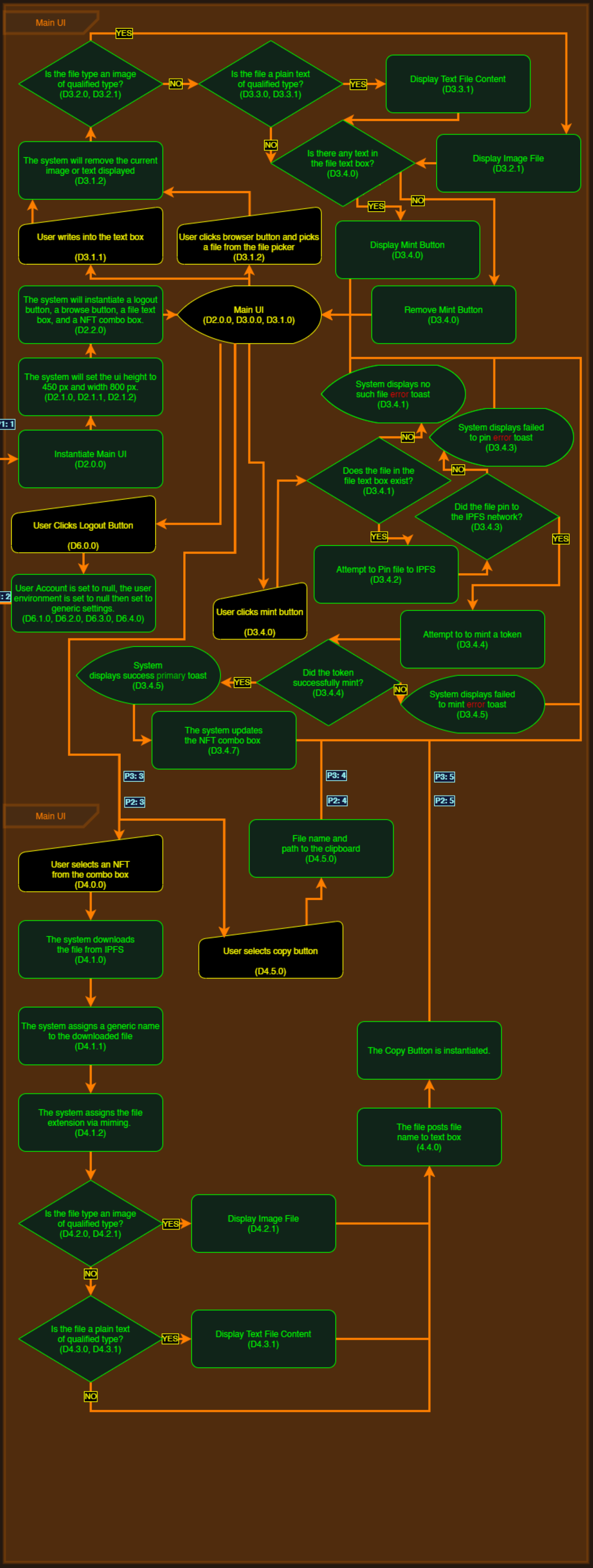

Main UI
YES
Is the file type an image of qualified type? (D3.2.0, D3.2.1)
NO
Is the file a plain text of qualified type? (D3.3.0, D3.3.1)
YES
Display Text File Content (D3.3.1)
The system will remove the current image or text displayed (D3.1.2)
NO
Is there any text in the file text box? (D3.4.0)
Display Image File (D3.2.1)
YES
NO
User writes into the text box (D3.1.1)
User clicks browser button and picks a file from the file picker (D3.1.2)
Display Mint Button (D3.4.0)
The system will instantiate a logout button, a browse button, a file text box, and a NFT combo box. (D2.2.0)
Main UI (D2.0.0, D3.0.0, D3.1.0)
Remove Mint Button (D3.4.0)
The system will set the ui height to 450 px and width 800 px. (D2.1.0, D2.1.1, D2.1.2)
System displays no such file error toast (D3.4.1)
System displays failed to pin error toast (D3.4.3)
Instantiate Main UI (D2.0.0)
NO
Does the file in the file text box exist? (D3.4.1)
NO
Did the file pin to the IPFS network? (D3.4.3)
User Clicks Logout Button (D6.0.0)
YES
YES
Attempt to Pin file to IPFS (D3.4.2)
User Account is set to null, the user environment is set to null then set to generic settings. (D6.1.0, D6.2.0, D6.3.0, D6.4.0)
User clicks mint button (D3.4.0)
Attempt to to mint a token (D3.4.4)
System displays success primary toast (D3.4.5)
YES
Did the token successfully mint? (D3.4.4)
NO
System displays failed to mint error toast (D3.4.5)
The system updates the NFT combo box (D3.4.7)
P3: 3
P2: 3
P3: 4
P2: 4
P3: 5
P2: 5
Main UI
User selects an NFT from the combo box (D4.0.0)
File name and path to the clipboard (D4.5.0)
The system downloads the file from IPFS (D4.1.0)
User selects copy button (D4.5.0)
The system assigns a generic name to the downloaded file (D4.1.1)
The Copy Button is instantiated.
The system assigns the file extension via miming. (D4.1.2)
The file posts file name to text box (4.4.0)
Is the file type an image of qualified type? (D4.2.0, D4.2.1)
YES
Display Image File (D4.2.1)
NO
Is the file a plain text of qualified type? (D4.3.0, D4.3.1)
YES
Display Text File Content (D4.3.1)
NO

Diagram 3

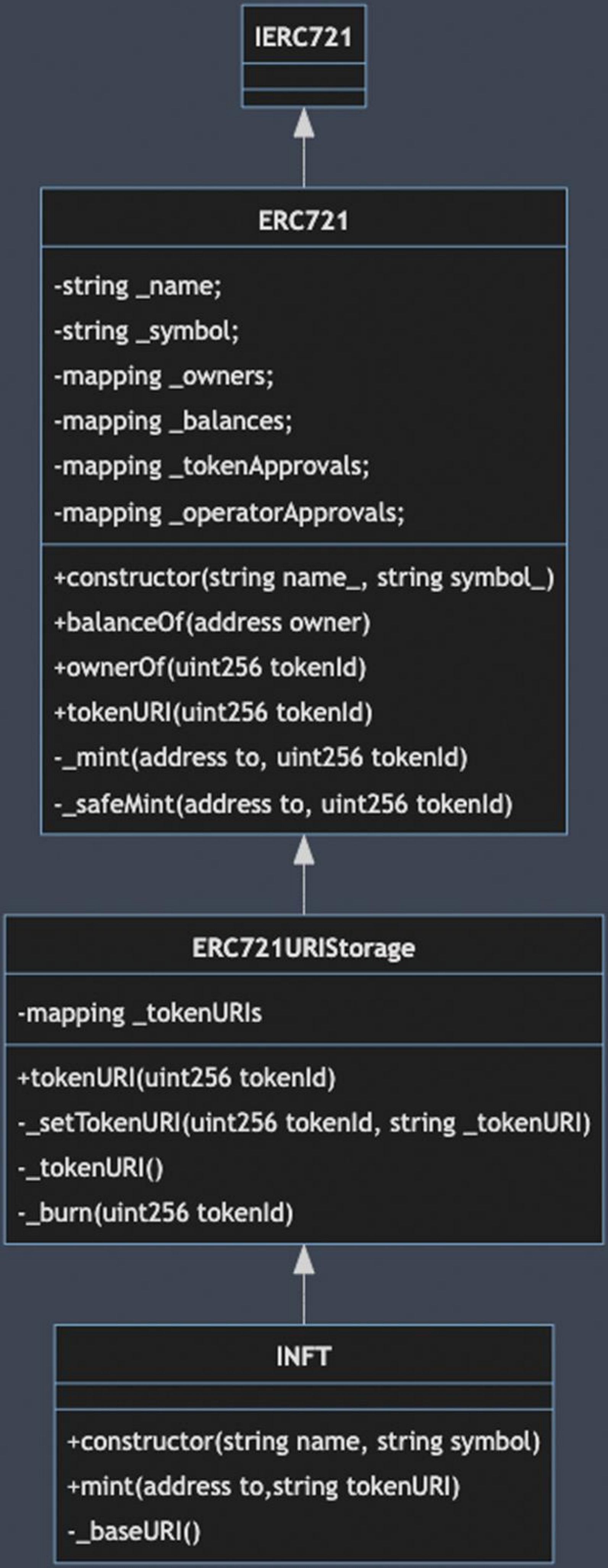
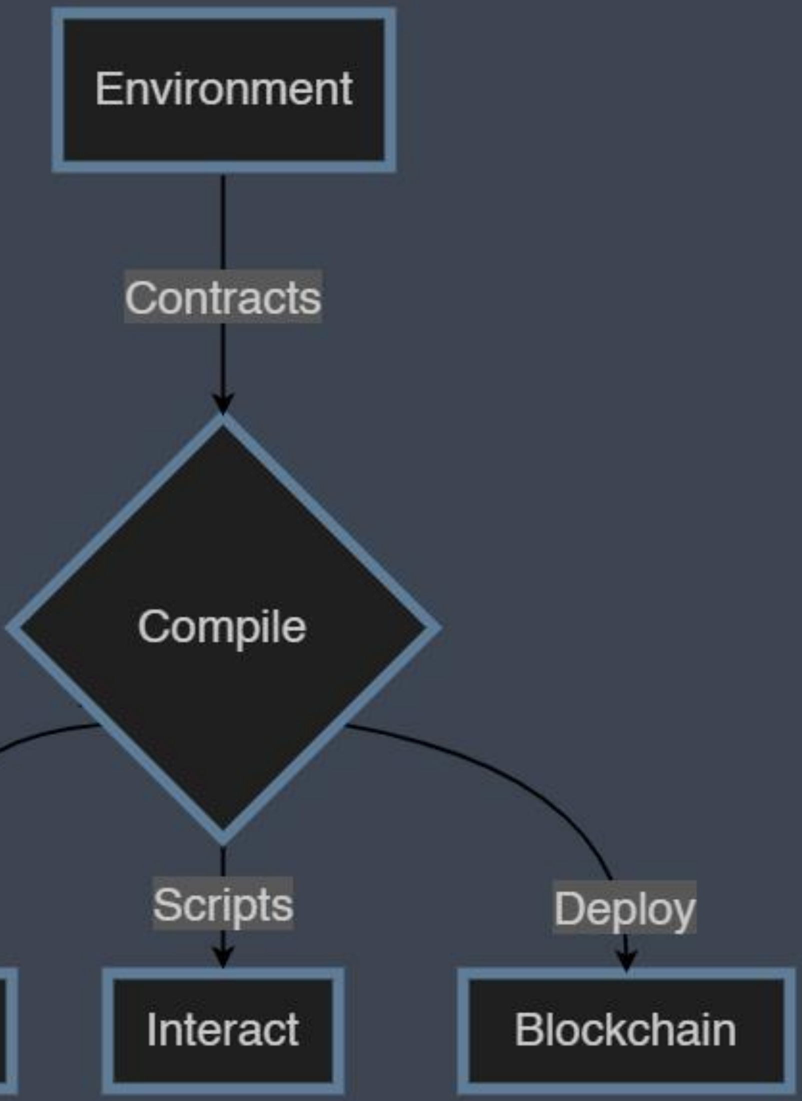

IERC721
ERC721
-string _name;
-string _symbol;
-mapping _owners;
-mapping _balances;
-mapping _tokenApprovals;
-mapping _operatorApprovals;
+constructor(string name_, string symbol_)
+balanceOf(address owner)
+ownerOf(uint256 tokenId)
+tokenURI(uint256 tokenId)
-_mint(address to, uint256 tokenId)
-_safeMint(address to, uint256 tokenId)
ERC721URIStorage
-mapping _tokenURIs
+tokenURI(uint256 tokenId)
-_setTokenURI(uint256 tokenId, string _tokenURI)
-_tokenURI()
-_burn(uint256 tokenId)
INFT
+constructor(string name, string symbol)
+mint(address to,string tokenURI)
-_baseURI()

Diagram 4


Environment
Contracts
Compile
Test
Scripts
Deploy
Debug
Interact
Blockchain

Testing Documentation

Test Case #	Requirement Tested	Rational	Input(s)	Expected Output	Pass/ Fail
1	D1.0.0	User shall be able to log in.	private key/wallet address	login successful message based on if the wallet is located given the input ID	Pass
2	D1.1.0	The System shall set the UI height and width.	n/a	UI should set height and width to the coded values	Pass
3	D1.1.1	The System shall set the UI height to 160px.	n/a	UI height should be set to 160px	Pass
4	D1.1.2	The System shall set the UI width to 400px.	n/a	UI Width should be set to 400px	Pass
5	D1.2.0	User shall be able log into an environment.	private key/wallet address	login successful message based on if the wallet is located given the input ID	Pass
6	D1.2.1	User shall be able to choose an environment.	Options between Local, Ropsten, Eth main net	Selected option should persist when login button is pushed	Pass
7	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.	Local Env chosen	Connection to local server with port is established	Pass
8	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.	Ropsten env chosen	connection to ropsten server at the given address with the chain id is established	Pass
9	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.	Eth main net env chosen	connection to the ethereum main net at the given address with the chain id is established	Pass
10	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.	specifically no input to env field	error message should pop up telling the user to select an environment	Pass

11	D1.2.6	The System shall configure the current environment based on the User's environment selection.	Options between Local, Ropsten, Eth main net	Whichever option is chosen, the configuration should be setup as per the previous connection tests	Pass
12	D1.3.0	User shall be able log in with a private key hex key.	a private key	Given the private key, the application should move forward into setting up the environment with this key as the user for this instance	Pass
13	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.	any given environment option and NO key value	a toast message (popup) should appear to let the user know to enter a key	Pass
14	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.	any given environment option and an incorrect key value	a toast message (popup) should appear to let the user know the key is invalid	Pass
15	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.	any given environment option	a line in the log file will be printed upon incomplete server connection	Pass
16	D1.4.0	If login is successful, then the System shall pop a toast with the current address's balance.	an accepted private key and environment option	a toast message will appear showing the current ether value of the given private key	Pass
17	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.	an accepted private key with an ether balance	a toast message should appear green and display the wallet ether balance	Pass
18	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.	an accepted private key with an ether balance	a toast message should appear yellow and display the wallet balance = 0 and display "warning"	Pass
19	D1.4.3	The System shall populate a combo box will all NFTs associated with the users address.	an accepted private key with an ether balance	the combo box will display all NFT hashes that have been minted by this application	Pass
20	D1.4.4	The user's private key shall be held as a variable in memory.	an accepted private key value	the address private key should be held in memory	Pass

				as the instance of iNFT login is open	
21	D2.0.0	The System shall instantiate the main block	key login values (private key and environment)	The main block should launch after the user clicks through "login"	Pass
22	D2.1.0	The System shall expand UI height and width.	key login values	the main block dimensions should be instantiated per the specs below	Pass
23	D2.1.1	The System shall set the UI height to 450px.	key login values	the main block height will be set to 450px	Pass
24	D2.1.2	The System shall set the UI width to 800px.	key login values	the main block width will be set to 800px	Pass
25	D2.2.0	The System shall instantiate a Logout button, a browse button, a file text box, and a NFT combo box.	key login values	The logout button, browse button, file next box, and combo nft box should initiate and be functional	Pass
26	D3.0.0	The User shall be able to mint a file.	any file	the user should be able to select any given file and be able to process the mint for it	Pass
27	D3.1.0	The User shall be able to select a file.	any file	the browse button should pull up the file explorer where the user can select a file from their machine	Pass
28	D3.1.1	The User shall be able to write a file name into the text box.	text from keyboard	the user can input text such as their file name/file path in the input box and the 'mint' button should appear	Pass
29	D3.1.2	The User shall be able to pick a file from the file picker.	any file	the file selector should allow the user to select a file from their machine and the file path should populate in the text box	Pass
30	D3.1.2	The System shall remove the image or text displayed when the text box is updated.	any file	when the text in the inbox is altered, the	Pass

				preview image should be removed	
31	D3.2.0	The System shall display image files.	any image file	if the file has one of the following image extensions (webp, jpeg, png, gif, or jpg) then a preview of the image should be displayed in the empty space of the main box	Pass
32	D3.2.1	If the file bytecode type is "webp", "jpeg", "png", "gif", or "jpg", then the System shall display the image.	any image file	if the file has one of the following image extensions (webp, jpeg, png, gif, or jpg) then a preview of the image should be displayed in the empty space of the main box	Pass
33	D3.3.0	The System shall display plain text files.	a text file	if the file has one of the following image extensions (txt, html, xml, css, js, htm, json) then a preview of the file will be displayed in the main box	Pass
34	D3.3.1	If the file bytecode type is "txt", "html", "xml", "css", "js", "htm", or "json", then the System shall display the text.	a text file	if the file has one of the following image extensions (txt, html, xml, css, js, htm, json) then a preview of the file will be displayed in the main box	Pass
35	D3.4.0	If there is any text in the file text box, then the System shall make the mint button available.	any text	when text is entered in the file text box, any given character combination, including a space, should make the mint button visible	Pass
36	D3.4.1	If the file does not exists then, then the System shall pop a toast message error stating no such file exists.	any text/file	hitting the mint button when the text in the file text box cannot be recognized as a file on the	Pass

				machine, an error toast message should appear	
37	D3.4.2	If the file does exists, the System shall attempt to pin the file to the IPFS network.	any file	if the file is recognized by the machine, then the system will attach it to the IPFS network	Pass
38	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.	any file	if the IPFS network is not reached or there is an error with the pin to IPFS, then an error toast message will appear	Pass
39	D3.4.4	If the files successfully pins to the IPFS network, the System shall take the hash code generated and attempt to mint a token to the Ethereum network.	any file	post pin the to the IPFS network, the file will be hashed and sent to the eth network	Pass
40	D3.4.5	If the system successfully mints the token, the System shall pop a toast message primary stating Success!	any file	if the connection to eth is made, and the file is sent along for minting, the success toast message will appear	Pass
41	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.	any file	if the connection to eth network is unsuccesful, a toast failure message will appear	Pass
42	D3.4.7	The System shall populate a combo box will all NFTs associated with the users address.	a private key that has minted an nft through the application	once an address has a minted nft, the combo box will populate the hashes of nft's minted to that address. This process is slow and takes a relog refresh after the mint completes	Pass
43	D4.0.0	The User shall be able to select any NFT from the combo box.	a private key that has minted an nft through the application	any hash selected should show a preview of the image	Pass

44	D4.1.0	If the user selects an NFT, the System shall download the NFT to the local system.	a private key that has minted an nft through the application	Copying to clipboard will copy the file path on the system to the image	Pass
45	D4.1.1	The System shall assign a generic name to the file.	a private key that has minted an nft through the application	the file is stored with the name "StoredFile.png" to the local machine.	Pass
46	D4.1.2	The System shall use miming to determine the file type is and assign the proper extension to the file.	a private key that has minted an nft through the application	the system will determine what the nft's type was at mint and assign that extension to the local file	Pass
47	D4.2.0	The System shall display image files.	a private key that has minted an nft through the application	any hash selected should show a preview of the image	Pass
48	D4.2.1	If the file bytecode type is "webp", "jpeg", "png", "gif", or "jpg", then the System shall display the image.	a private key that has minted an nft through the application	any hash selected should show a preview of the image	Pass
49	D4.3.0	The System shall display plain text files.	a private key that has minted an nft through the application	if the nft was a text file, then it will display the text within the file in the display portion of the application	Pass
50	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.	a private key that has minted an nft through the application	if the nft was a text file, then it will display the text within the file in the display portion of the application	Pass
51	D4.4.0	The System shall post the file name to a text box.	a private key that has minted an nft through the application	the file is stored with the name "StoredFile.png" to the local machine.	Pass
52	D5.0.0	Not implemented	n/a	n/a, functionality not completed	Fail
53	D4.5.0	The User shall be able to copy the file path to the clip board.	a private key that has minted an nft through the application	Copying to clipboard will copy the file path on the system to the image	Pass

54	D6.0.0	The user shall be able to logout.	a private key that has minted an nft through the application	the logout button should function by closing out the current instance of the application	Pass
55	D6.1.0	The system shall set the account value to null.	n/a	the account value, set by the private key value, will be reset upon logout	Pass
56	D6.2.0	The system shall set user environment to null.	n/a	the environment value, set by the environment value, will be reset upon logout	Pass
57	D6.3.0	The system shall set to the generic environment settings.	n/a	the generic environment setting is set upon logout to allow for the options to be selected upon relogin	Pass
58	D6.4.0	The system shall display the Login screen.	n/a	the logout button should close the main application frame and reopen the login frame	Pass
59	D7.0.0	The System shall maintain a log of important events.	n/a	Important events include nft mints, as well as errors throughout the connection and minting process	Pass
60	D7.1.0	The System shall record important events as info events.	n/a	the log file highlights these events (essentially all events) as 'INFO'	Pass
61	D7.1.1	The System shall log a record that will display the url to the item posted to IPFS.	n/a	the URL for the nft minted is displayed in the log file	Pass
62	D7.1.2	The System shall log a record the transaction hash.	n/a	the transaction hash is stored in the log file under the ipfs url	Pass
63	D7.2.0	The System shall record system errors from the MainWindow.xaml class as error events.	n/a	any errors encountered around the MainWindow.xaml file are displayed in the log file	Pass
64	D7.2.1	The System shall log a record when the environment fails to connect.	n/a	any environment connection issues will log a line to the log file	Pass

65	D7.2.2	The System shall log a record when an error occurs during login.	n/a	any login issues that would throw errors outside of the key validation would get logged	Pass
66	D7.2.3	The System shall log a record when the System fails to display an image.	n/a	When the system is unable to display an image file a message will be logged	Pass
67	D7.2.4	The System shall log a record when the System fails to display a plain text file	n/a	when the system is unable to display the contents of the file, a message will be logged	Pass
68	D7.2.5	The System shall log a record when the System fails to get a file from the IPFS network.	n/a	if there is an issue retrieving a requested file from IPFS, a message will be logged	Pass
69	D7.2.6	The System shall log a record when the System fails to pin a file to IPFS.	n/a	if there is an issue posting to IPFS, a message will be logged	Pass
70	D7.2.7	The System shall log a record when the System fails to mint a token to the Ethereum network.	n/a	if the Ethereum network isnt reached when attempting to mint a file, a message will be logged	Pass
71	D7.3.0	The System shall record system errors from the Etherium_Interact class as error events.	n/a	if the Ethereum network isnt reached when attempting to mint a file, a message will be logged	Pass
72	D7.3.1	The System shall log a record when the System fails to create a local web3 object.	n/a	When a hash is retrieved from a past minted NFT and cannot be created locally, a message will be logged	Pass
73	D7.3.2	The System shall log a record when the System fails to create a test web3 object.	n/a	When a hash is retrieved from a past minted NFT and cannot be created locally, a message will be logged	Pass

74	D7.3.3	The System shall log a record when the System fails to create a production web3 object.	n/a	When a hash is retrieved from a past minted NFT and cannot be created locally, a message will be logged	Pass
75	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.	n/a	If the system can't retrieve the wallet address for the token of a past mint, a log message is added to the log file	Pass
76	D7.3.5	The System shall log a record when the System has an error minting the token.	n/a	If the minting process fails, a log message is added to the log file	Pass
77	D7.4.0	The System shall record system errors from the Helpers class as error events.	n/a	a log message is added to the log file if the Helpers class returns any errors	Pass
78	D7.4.1	The System shall log a record when the System fails to deserialize the JSON object selected.	n/a	a log message is added to the log file if the JSON file processing is hungup	Pass
79	D7.5.0	The System shall record system errors from the IPFS_Interact class as error event.	n/a	a log message is added to the log file if interactions with the IPFS system fail	Pass
80	D7.5.1	The System shall log a record when the System fails to write the IPFS file to local storage.	n/a	a log message is added to the log file if the connection to post to IPFS fails	Pass
81	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.	n/a	a log message is added to the log file if the returned hash cannot be mimed for extension to the local saved asset	Pass
82	D7.5.3	The System shall log a record when the system fails to pin a file to IPFS.	n/a	a log message is added to the log file if posting to IPFS fails	Pass
83	D7.5.4	The System shall log a record when the system fails trying to delete all stored files.	n/a	a log message is added to the log file if the system cannot delete stored files for any reason	Pass

Bugs

1. Over nonlocal networks the NFT combo box will not update with the minted tokens right away. This is due to the minting process. The combo box will show minted items after the list is polled again. Polling only currently occurs immediately after the mint occurs and it takes 1-5 minutes to mine the transaction. The fix for this is to have the list of available tokens polled every minute asynchronously and have an event be thrown when there is a change between what is available and what is shown and to update what is shown. This change is planned for version 1.1.0.

2. Environmental are hardcoded and should be using a configuration file for users to use local environments. This change is planned for version 1.2.0

3. The Main net is not available due to the cost of deploying the contract. This change is not planned.