CrypToadz Analysis

Aviv Zohman
Tyler Lewis

NFT stands for non-fungible token. That means these tokens are unique; they exist independently. Specifically, they are irreplaceable and exchangeable digital files held on the blockchain that are traded over the internet.

Pokemon cards, for example, are reproducible and owned among many individuals. NFT's are like Pokemon cards, but they are unique and exclusively owned on the blockchain. NFT's are important. They are an instance of something much larger than we can currently imagine or predict – a manifestation of a new era of entrepreneurship, societal norms, and private property.

Each archaeological period has been characterized by its predominant material -- the Stone Age, Bronze Age, Iron Age, and Silicon Age. We are approaching a new era, the Meta Age, and NFT's are paramount to this revolution. From a high-level perspective, this is why Tyler and I have chosen this topic for our project – a head start, somewhat.

CrypToadz are small amphibious creatures that are NFT's. That is, CrypToadz is an NFT *collection*. Our project focuses on CrypToadz because they are highly-traded and we like their design. But NFT's are hard to appraise because their value is constantly fluctuating. Accordingly, CyrpToadz are hard to appraise, and there currently lacks a sufficient CrypToadz repository for straightforward price analysis.

Let's say an individual has a new-found interest in the NFT world, but is uninterested in

the blockchain, cryptocurrency, and the like. They know NFT's offer the grandeur of a hunk of cash and they want to take a stab at cashing in some chips. But NFT marketplaces *do* include this information and it's discouraging to the NFT novice. For instance, OpenSea is the world's first and largest NFT marketplace. It is elegant and fluid with a beautiful interface. But it contains a heap of data that is only helpful to an NFT aficionado.

An NFT new-comer typically looks for just a few pieces of information: the price of the NFT, the image of the NFT, and its last sold price (this last one is important because the user can know how much it is valued). Therefore, this is our program's top-priority – a simple interface that provides these pieces of data (among other functions) in one-click.

Building the database was our first critical checkpoint of this project and the basis of our back-end. We utilized OpenSea's API and Python to pull metadata off all 7025 CrypToadz in the collection. We organized this data into a relational database, which allows us to discard redundant information and efficiently query for desired results. The database is hosted with Google's Cloud Computing service, permitting synchronous changes and usability among several users.

The database stores information regarding CrypToadz owners, historical sales, current market values, possible traits, and the CryptToadz' source images and individual traits. Given 181 possible traits, there is quite a bit of variation between these NFTs. Consequently, we implemented an index on all traits for faster query speed and efficiency.

Our program emphasizes the three critical pieces of data every NFT novice should be equipped with. But our program would be one-dimensional if this is all we included. Hence, our program has additional functionality. CrypToadz' traits are incredibly important to their value, so

we've implemented a frame that returns a CrypToad based on the user's selected traits. Furthermore, the CrypToadz collection is a community, and it's handy to know the usernames of other owners and their quantity of CrypToadz. Perhaps an owner is deemed "trustworthy" if they own more toadz. Hence, there is a frame that accepts the Owner's ID and returns their username and quantity of toadz. Lastly, we've implemented a sandbox for our user, that is a playground for inserting, updating, and deleting hypothetical users. A high-profile user may prefer various aliases to purchase CrypToadz' under. After all, owning lots of CrypToadz may draw attention to your account, and this is undesirable.

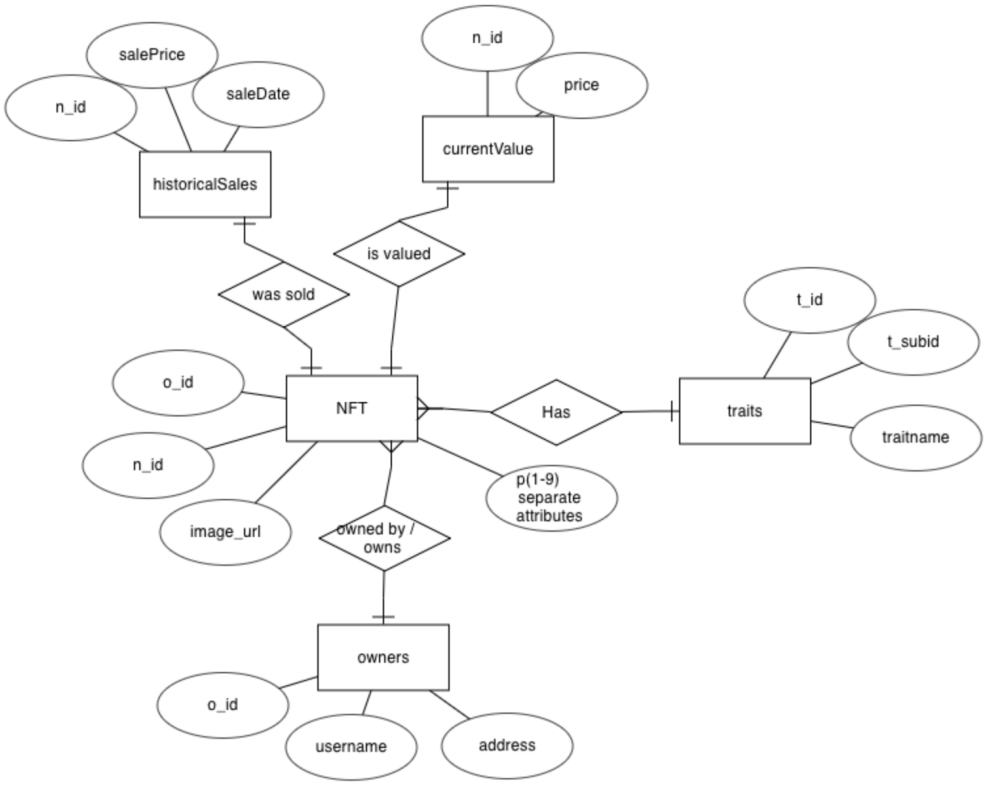
The front end software of our project is Python's Tkinter GUI package. The front end *design* stems from our functionality. That is, our design is simple and eloquent because our target audience is the NFT novice, and a convoluted interface only hinders a beginners comprehension. We want our interface to be intuitive, and therefore we drew inspiration from the most intuitive interface we deal with day-to-day. The iPhone. We have a single window that contains frames corresponding with their functionality, just as an iPhone contains one or more windows consisting of apps.

Going forward, we are considering scaling functionality to analyze any NFT project rather than specifically CrypToadz. An NFT's corresponding data can all be found on the blockchain, and would subsequently be a one stop shop for all NFT related data. Rather than continuing to query a SQL database, we would want to incorporate the ability to instantaneously read the blockchain for our data.

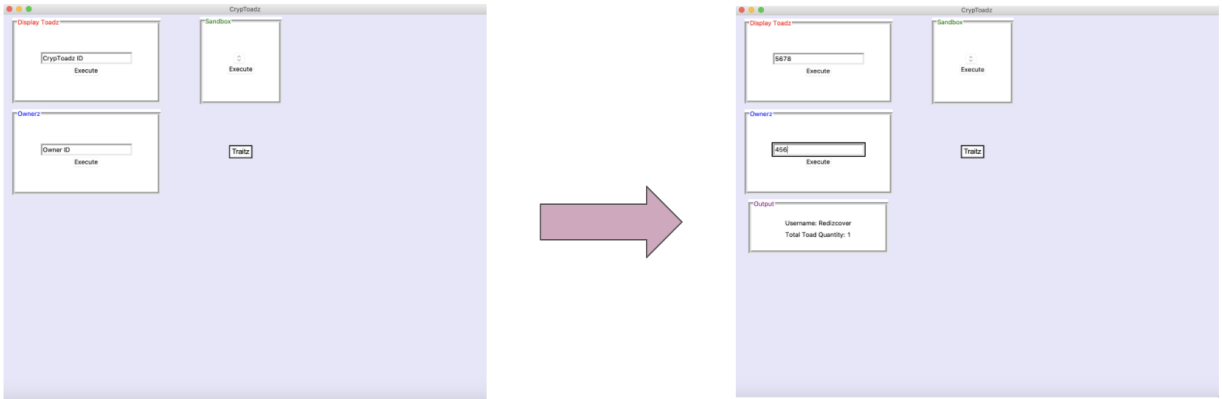
The crypto community has created standards for how NFTs store data including their image source and traits which allow for us to pull data from any project we want consistently

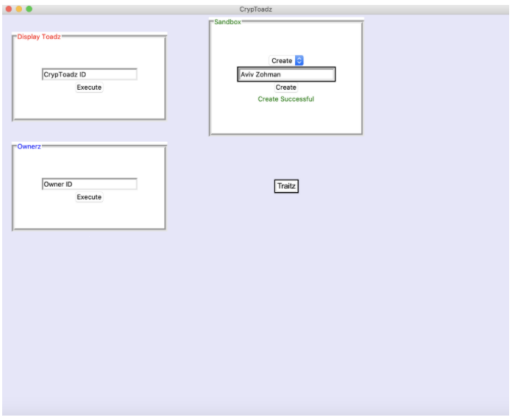
without oversight. This allows for the most up-to-date information being displayed and further cutting out the middleman API we have used to build our current database.

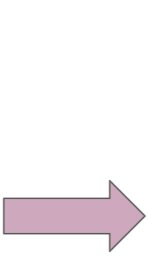
Schema Diagram

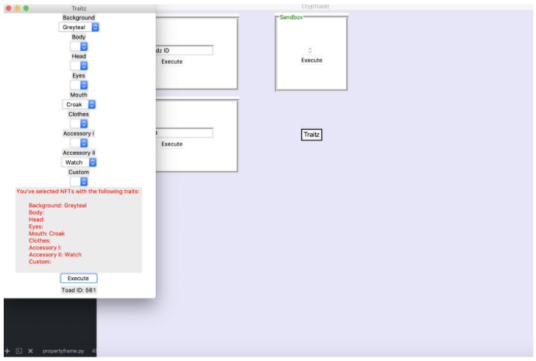


Application Images (details in screen recording)




CrypToadz
Display Toadz
CrypToadz ID
Execute
Owners
Owner ID
Execute
Sandbox
Create
Aviv Zohman
Create
Create Successful
Traitz




Traitz
Background
Greyteal
Body
Head
Eyes
Mouth
Croak
Clothes
Accessory I
Accessory II
Watch
Custom
You've selected NFTs with the following traits:
Background: Greyteal
Body:
Head:
Eyes:
Mouth: Croak
Clothes:
Accessory I:
Accessory II: Watch
Custom:
Execute
Toad ID: 561
CrypToadz
Sandbox
Execute
Traitz