# Princple of Distributed Ledger Coursework

Lifan Zhao

February 2020

---

# 1 Introduction

Cryptomon is a blockchain platform where players can lottery their cryptomons with Ethereum wallet, breed the new cryptomons from the ones they have owned, exchange cryptomons for the one they desire, and fight against each other's cryptomon.

Therefore, the platform is split into four different page: Trade, Breed, Fight, Profile:

- **Trade**: Trade page is designed for gaining cryptomons. The ways includes lottery, buy cryptomons and share cryptomons.
  - Lottery: By paying **lotteryFee** amount of WEI to the owner of the contract, the players can gain randomly five cryptomons from the contract's. **baseCryptomons**
  - Buy: The cryptomons that other players want to sell will be listed here. One can buy these cryptomons by paying the tagged **price** to the provider player.
  - Share: Since the breed mechanism and lottery, players may need some cryptomons with specific requirements and at the mean time get some duplicated cryptomons. They can share the one that no longer needs and raise the exchange requirement. Hence, all the shared cryptomons are listed here, only those who can provide the required cryptomon can completed the exchange. It will not charge any fee, other than gas.
- **Breed**: Breed page is designed for giving birth of new cryptomons. The breeding mechanism is designed in the following:
  - Only cryptomons belongs to the same **pokedexId** and from two different genders can breed.
  - The new cryptomons belongs to the same **pokedexId** as its parent. Its level will be the half of lower **level** of its parents. And the points from each layer will be distributed randomly to its **HP**, **ATK** and **DEF** attribute.
  - After successfully breeding, both of its parent require time to cool down.
- **Fight**: Fight page is designed for cryptomons fighting and level up. One can set his cryptomons as **forFight** so that they can be listed here for other players to attack. As a reward for being chosen, these cryptomons is the first to attack during fighting.
  Unfortunately, for the security reason that miner can view and utilize the random function to win every fight, in Cryptomon every fight is predictable. The fighting mechansim is designed in the following:
  - The listed cryptomons attack first. The damage is attacker's **ATK** subtracted by victim's DEF. If DEF is larger than **ATK**, then no damage applies.
  - Two cryptomons will fight until one's HP become zero.
  - The winner will slowly recover its **HP** to **maxHP**, while the loser will have to be remain unavailable until **requiredCoolDown** time elapse.
  - The winner will gain one **winPoint**. And with 10 **winPoint**, the cryptomons is able to level up. When levelup, the cryptomons can add one point to the attribute the play wants. Yet, at the same time, its **requiredCoolDown** will be extended.
- **Profile**: In Profile Page, players can see all of cryptomons they owned stated with different status. Players can also set price to **sellCryptomons**, set requirement to **shareCryptomon** or **setForFight**. Cancel options is also provided here.

Waht's more, there is a set of **baseCryptomons** on blockchain to which only owner of the contract can added new cryptomons. **baseCryptomons** is private to all. Owner only can only perform add action which I think can be good way to store so many pokemons without exceeding the size limit of smart contract.

# 2 Architecture

## 2.1 Solidity

The platform is built on Solidity. Again, Cryptomons Contract is splited into 4 different part:

- cryptomons.sol: in charge of cryptomons storage and creation function.
  - **buildCard(uint16 pokedexId, uint16 HP, uint16 ATK, uint16 DEF, uint16 requireCoolDown) public onlyOwner** which can only be called by owner to build the baseCryptomons where normal cryptomons get their settings from.
  - **createCryptomon(uint pokedexId, uint16 level) internal** which can only be called from other two public functions "breedCryptomons" and "lotteryCryptomons".
  - **lotteryCryptomons** which transfer amount of WEI to the owner of contract and get 5 random cryptomons from baseCryptomons with random gender and level 1.
  - some helper function to get mapping in struct.
  - **rand(uint256 _length, uint seed)** which use block.difficulty, block.timestamp and an input seed as random seed and returns random number from required range(length). Indeed, it is not safe. Yet, nothing important related to this random function, therefore I use it directly to decide gender and pokedexId.
  - **withdraw() external** which use for get the amount of money the buyer transferred.
- cryptomonsBreed.sol: in charge of cryptomons breed and related functions.
  - **breed(uint cryptomonId1, uint cryptomonId2) public onlyOwnerOf(cryptomonId1) onlyOwnerOf(cryptomonId2)** which requires msg.sender as the owner of two selected cryptomons, check the breeding requirements, generated a new cryptomon and set two cryptomons to cool down.
  - **addHP/addATK/addDEF(Cryptomon storage cryptomon, uint16 pointsToAdd) internal** which add level up points to attributes.
- cryptomonsFight.sol: in charge of cryptomons fight and related functions.
  - **fightCryptomon(uint target, uint cryptomonId) public onlyOwnerOf(cryptomonId) readyFighting(target, cryptomonId)** which requires cryptomons[cryptomonId] belongs to msg.sender and cryptomons[target] is either not for sale, for share, or cooling down. And calculate the fight result, give the winPoint to winner and set loser to cool down.
  - **setForFight(uint id, bool forFight) public onlyOwnerOf(id)** which only allow the owner of cryptomon to change the fight status.
- cryptomonsTrade.sol: in charge of cryptomons sell, purchase and exchange functions.
  - **sellCryptomon(uint id, uint price) public onlyOwnerOf(id)** which only allow the owner of the cryptomons to set the price and put it on market.
  - **retreiveCryptomon(uint id) public onlyOwnerOf(id)** which only allow the owner of the cryptomons to cancel the sale and remove it from market.
  - **buyCryptomon(uint id) public payable** which allow players to buy the cryptomons with its tagged price.
  - **sharingCryptomon(uint id, uint32 requiredPokedexId, uint32 level, uint32 gender) public onlyOwnerOf(id)** which only allow owner of the cryptomons to share it with stated requirements and put it on market.
  - **exchangeCryptomons(uint id, uint target) public onlyOwnerOf(id)** which allow the player to exchange the shared Cryptomon on market with the satisfied cryptomons he have.
  - **cancelSharingCryptomon(uint id) public onlyOwnerOf(id) returns (bool)** which remove the cryptomons from sharing market and reset its exchange requirements.

## 2.2 Web Interface

The frontend is implemented with React and web3 library. Player have to connect to port 8545 with MetaMask in browser to interact with the platform. The web interface as shown above divides into four pages where each page contains varieties number of CardList which iterates each Cryptomons as a Card. Most cards have button to interact with.

Noticed that due to limited frontend knowledge, UI feedback is not quite well, user might have to wait a while or refresh page to re-enter the page in order to see the changes.

No backend is implemented, because web3's interactions with Cryptomon Contract give the enough functionalities like backend's create, query data to database.

# 3 Threat Model

Since the platform did not utilize either ERC721 or SafeMath, there are several parts need to be taken care of. Firstly, Re-entrancy attacks exploits contracts' fallback function to execute certain lines of code unexpectedly. In the two payable function(lotteryCryptomon, buyCryptomons), using transfer() function limits the gas to 2300 so that no more execution can be performed. Also, utilizing the checks-effects-interactions patterns, the contract put all the state-changing code before the transfer/send code can prevent re-entrancy.

Secondly, underflow and overflow. The uints can change dramatically when performing calculation near the limits. There are uint32 and uint8 variables declared to shrink the Cryptomons size. And they will surely causing problems in not treated carefully.

- **Cryptomon.level, Cryptomon.winPoints**: which are uint16. All the calculations on them are addition and set to zero. Beside, after upgrade one level, cryptomon need to at least rest 1 minutes, and the cool down time increases as level increases. It will take at least 24855 years to overflow.
- **Cryptomon.properties**: which is mapping(string=>uint32) and the attributes inside will only be added one point once level up, therefore uint32 should not be underestimate to overflow.
- Other calculation has been compared with each other before calculating.

Meanwhile, the above assumption is established on the belief that the owner is not adversary. Since this Cryptomon is not quite decentralized at the beginning, and all the baseCryptomons are created only by the contract owner, this means that he or she can set the baseCryptomons attributes with an edge value and create an overflow. But, at the buildCryptomons, we use uint16 as parameter which is sufficiently enough for pokemon attributes and convert to uint32 in storage.

At last, the random number generator I used is indeed unsafe. The miner can utilize their convenience to get a fixed number. But, the scenario I use random number is deciding gender and pokedexId, which are both equally distributed, hence the threat is not malicious.

# 4 Usage and Output

## 4.1 Owner


Figure 1: Sidebar Content

The side bar navigate to all the four pages. And only the owner of contract should go into "Build Cryptomons"

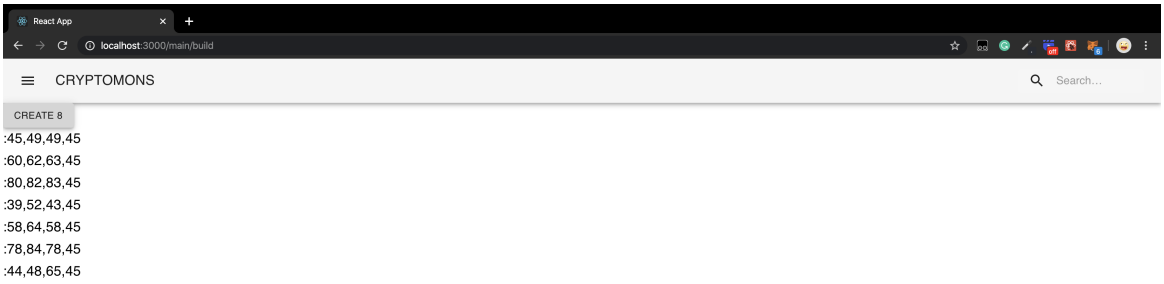



Figure 2: Press Button and Build Cryptomons

After pressed the button, it will fetch from pokeapi to get the correct data, and build base cryptomons to the contract. It will cost gas fee, which can be seen in the Metamask's blue bubble on upper left corner.

## 4.2 User 1

Now we assume a user come in with his account, and go to the trade page to lottery some Cryptomons. He can click the slot machine icon. And seeing this.

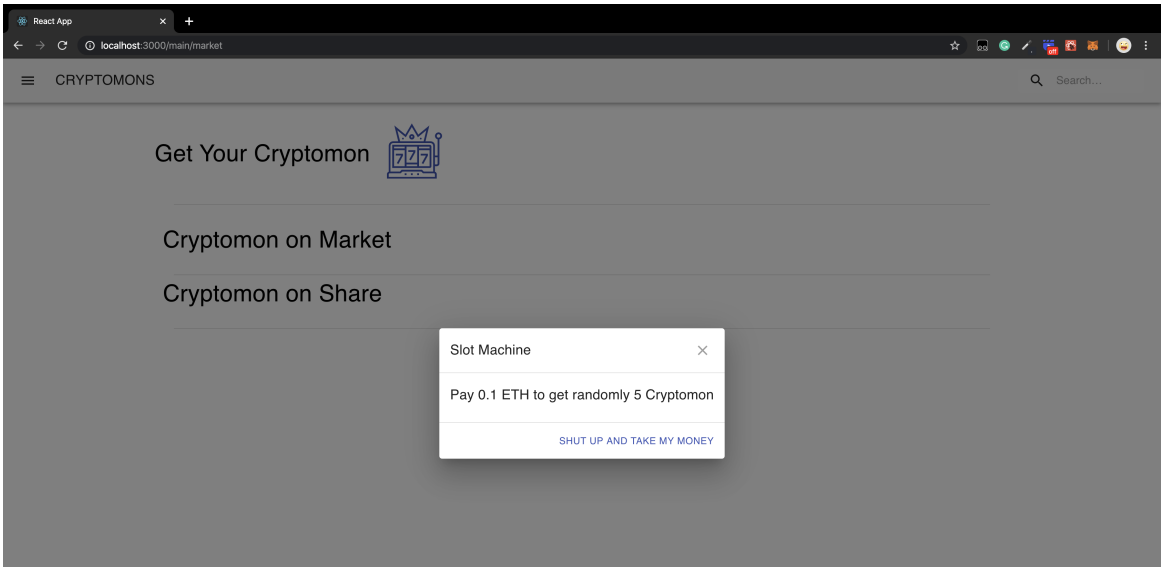



Figure 3: Lottery Start

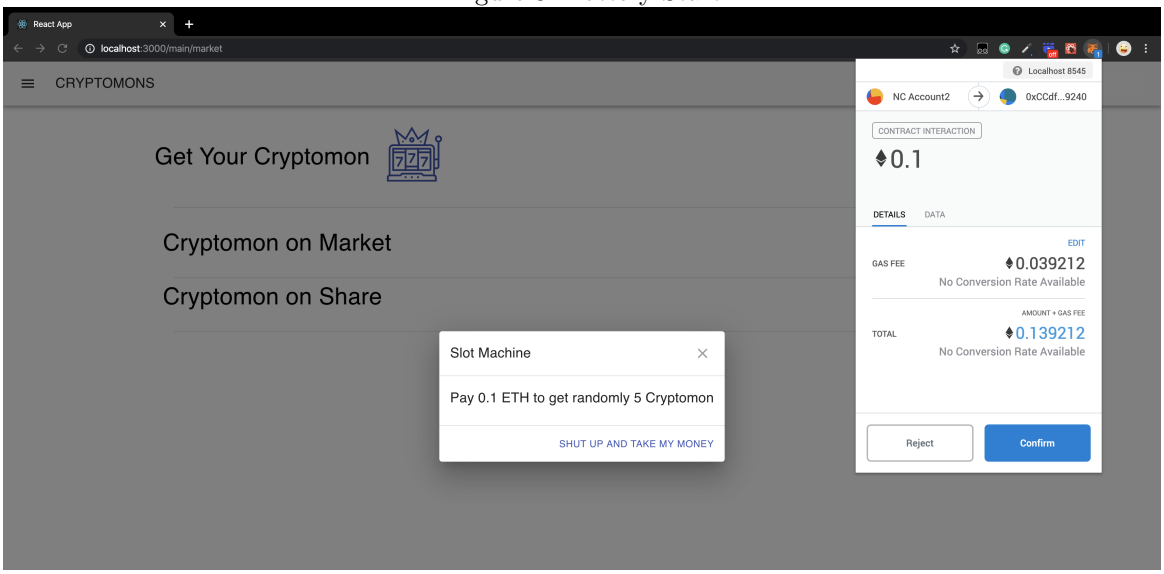



Figure 4: Lottery Confirm

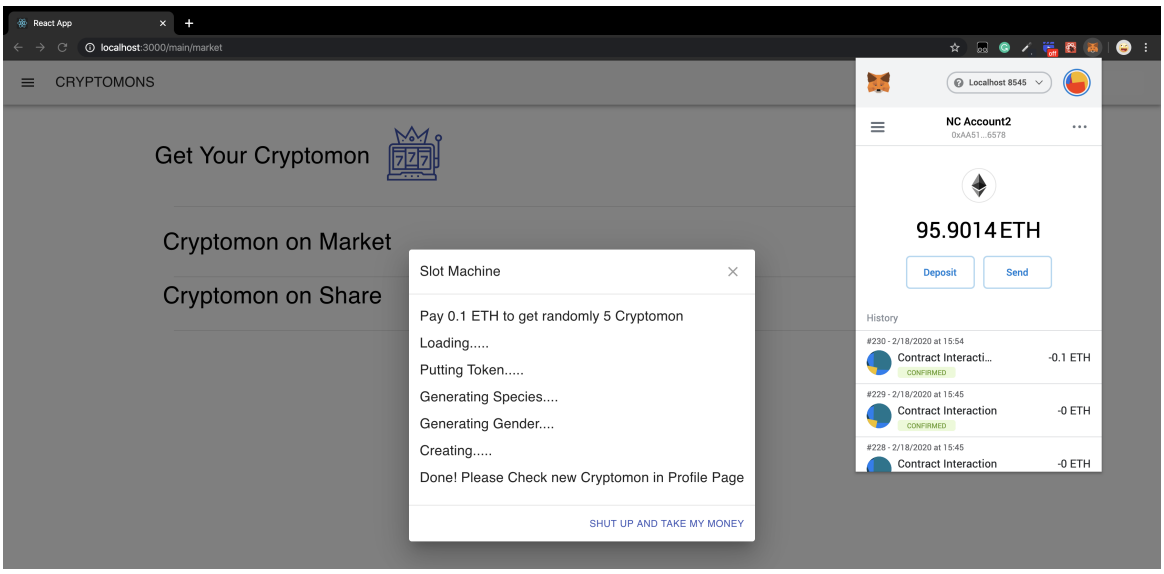



Figure 5: Lottery Finish

Now let's go to Profile Page to check your Cryptomons

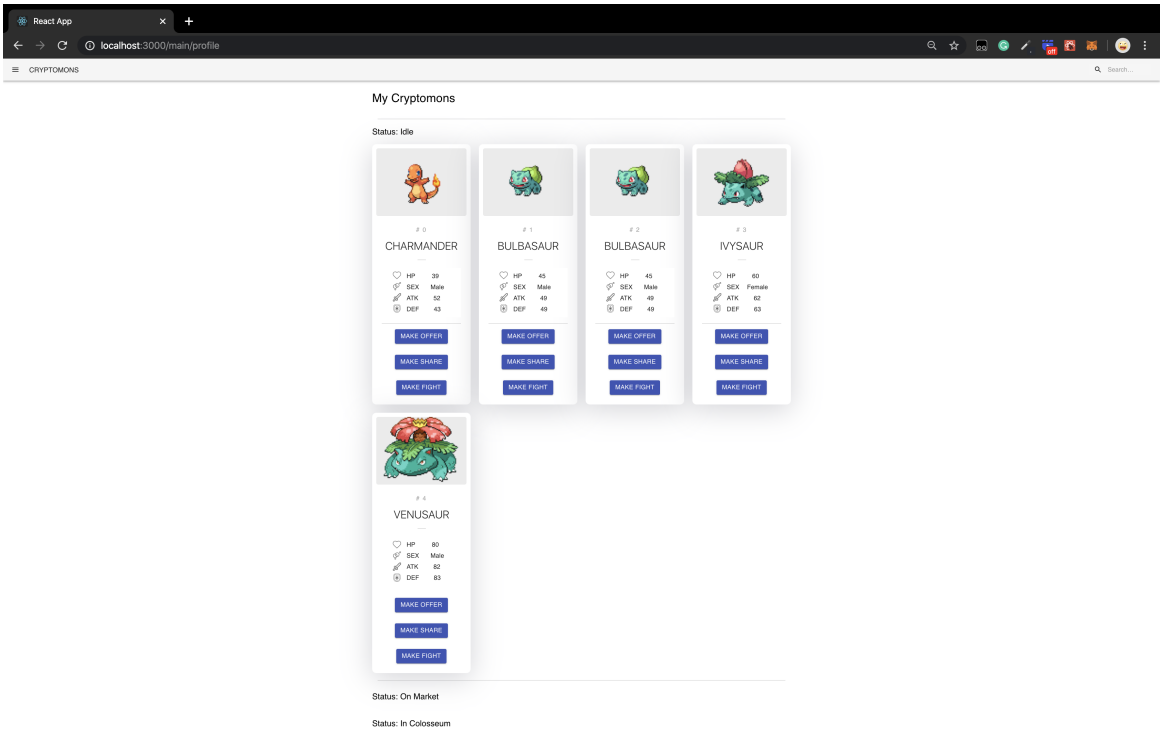



Figure 6: Profile Page After Lottery

And Now we can try to sell some cryptomons with the input price.

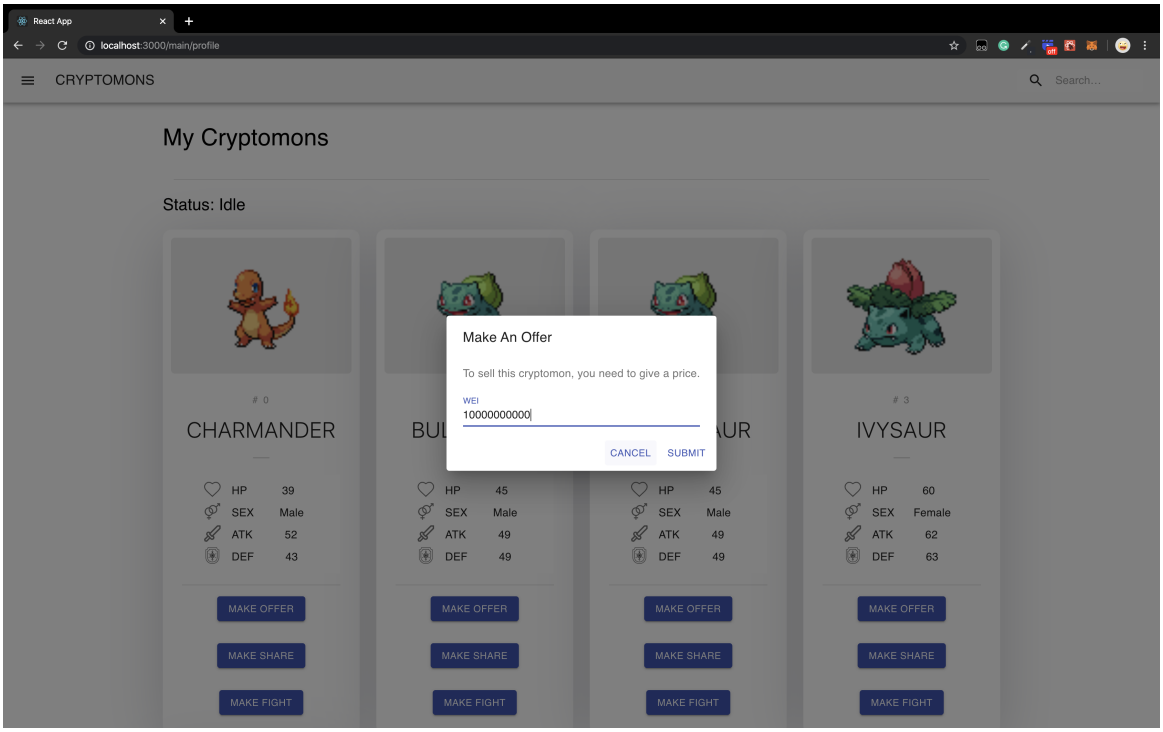



Figure 7: Sell Cryptomon

We can also share the cryptomons in exchange of other things. By entering the required pokedexId, required level, and required gender, and Submit the sharing.

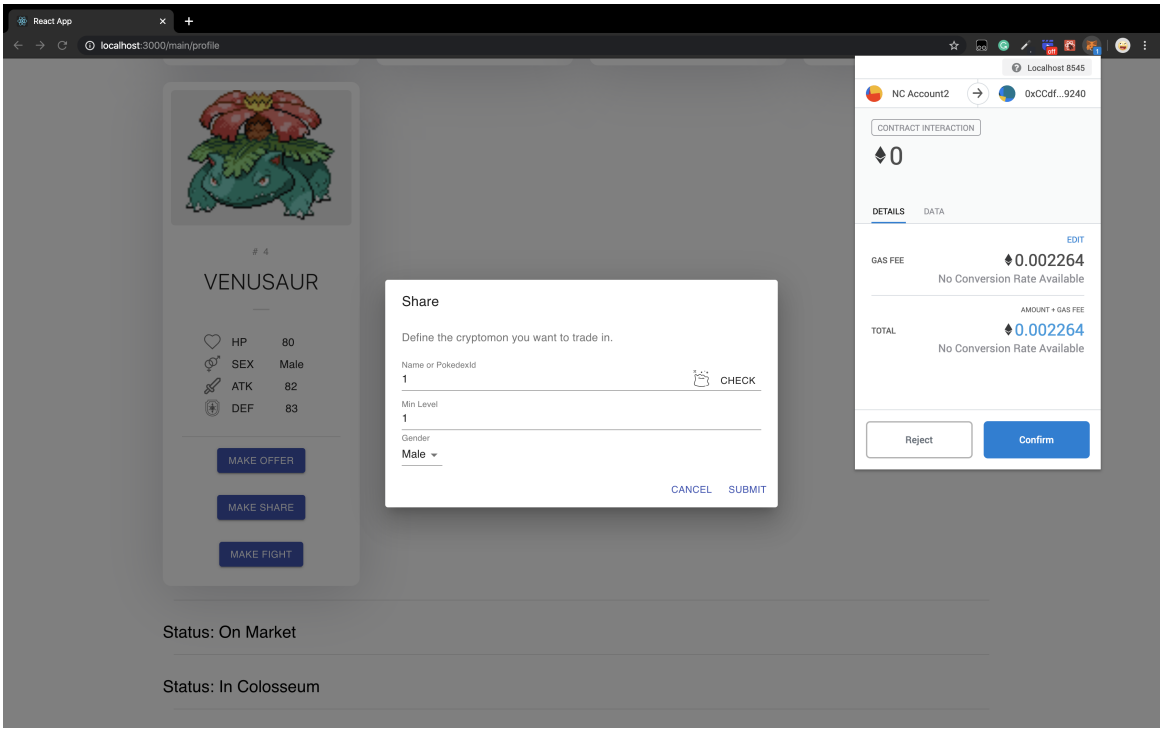



Figure 8: Share Cryptomon

At last, when we update this profile page, we can see the changed cryptomons have been listed in different area.

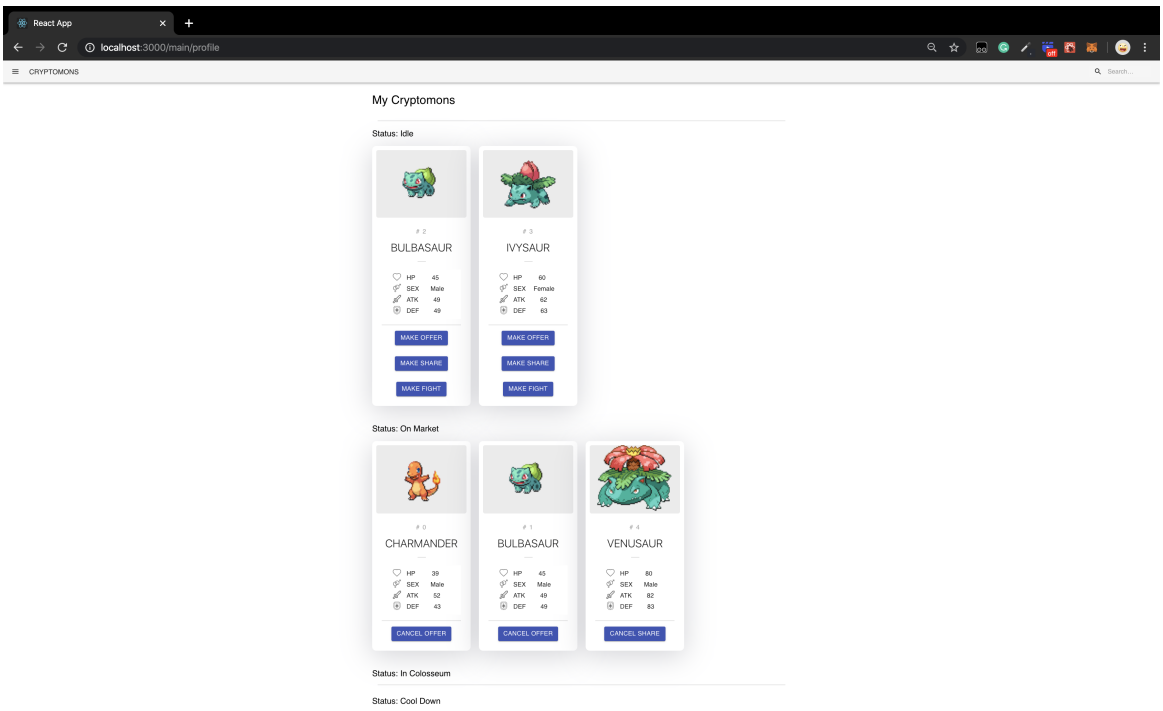
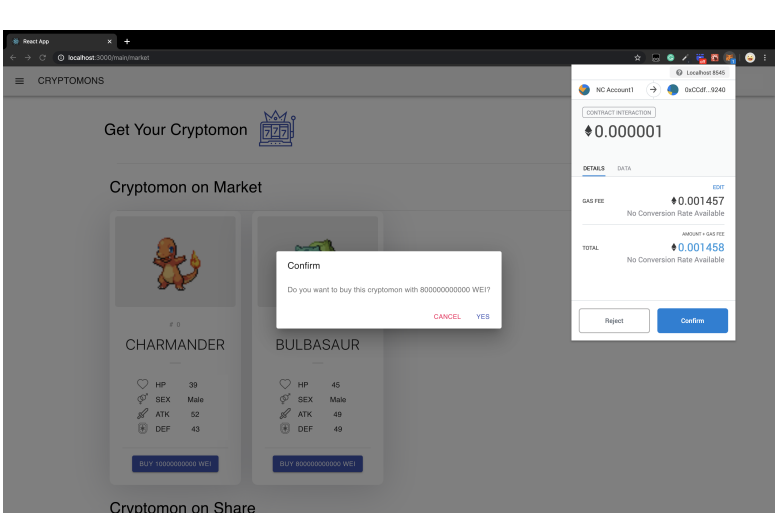


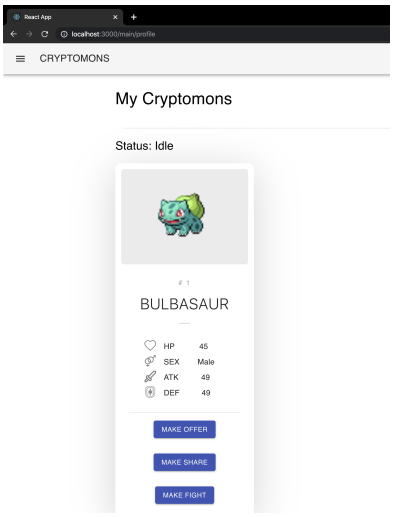

Figure 9: Profile Page Updated

## 4.3 User 2

Now we can change to another account, and see how the market looks like now. The forSale crytomons and forShare Cryptomons are listed. We can as well buy them.




(a) Buy




(b) Owned

Figure 10: Trade Page

The transaction detail from Ganache is following:

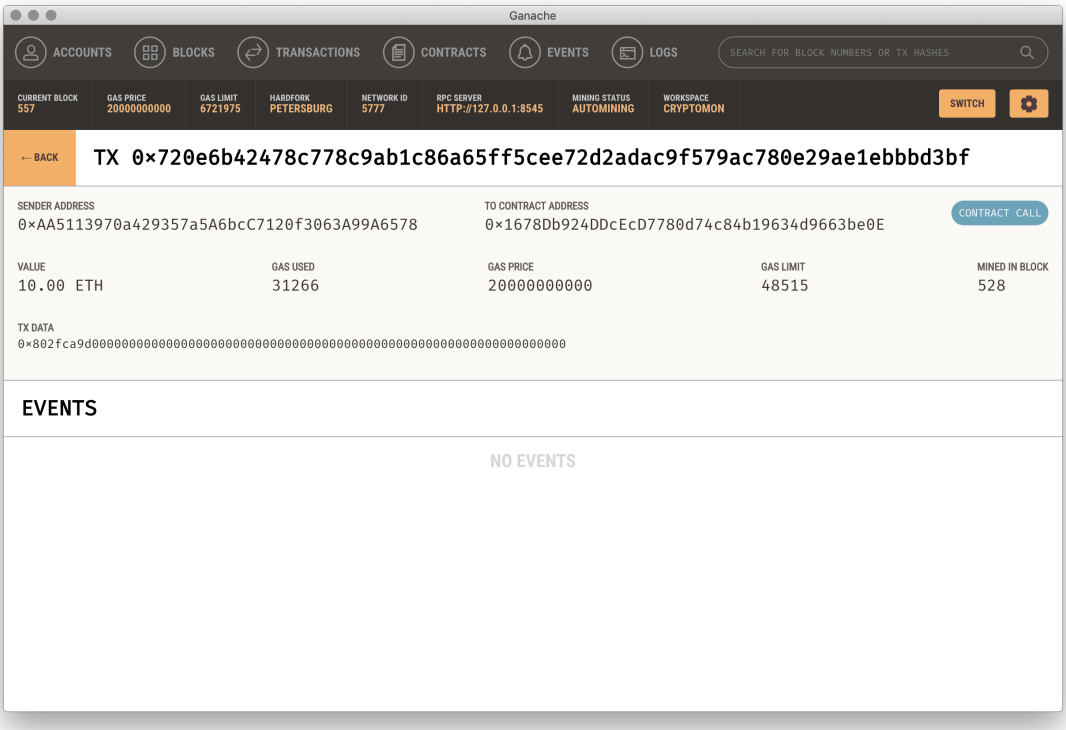
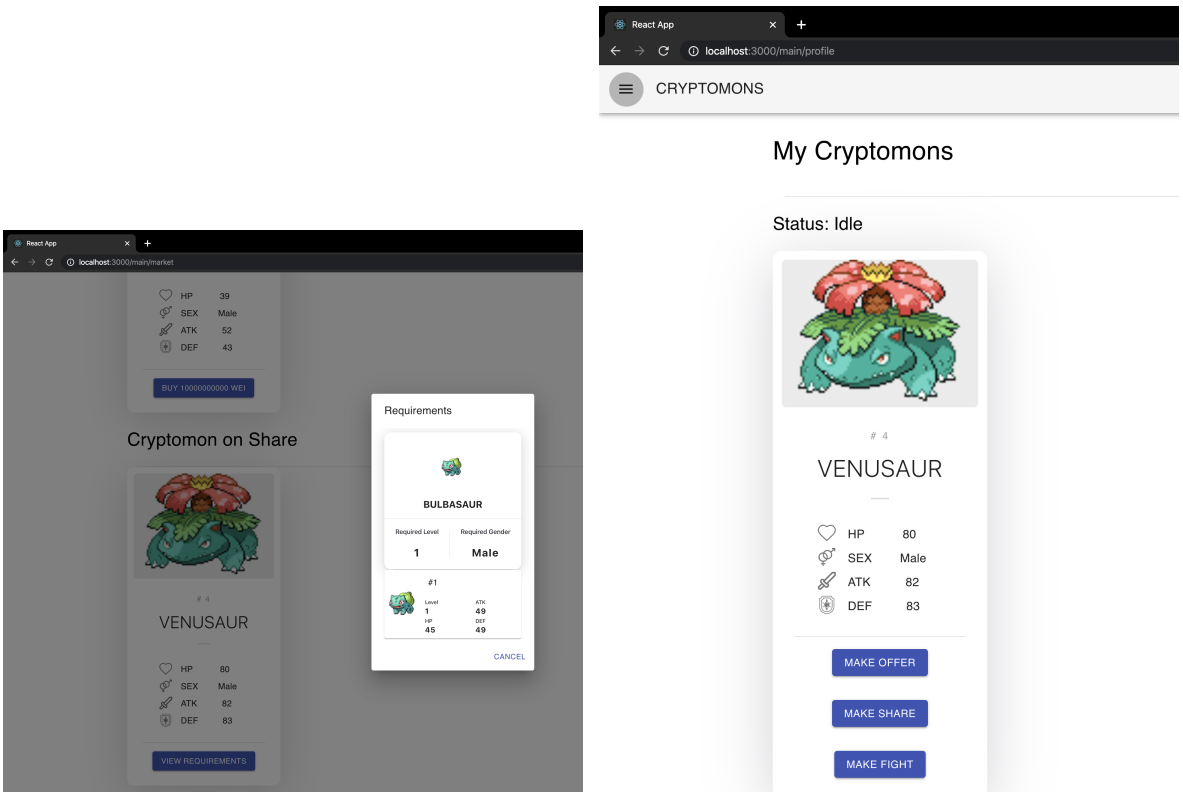



Figure 11: transaction

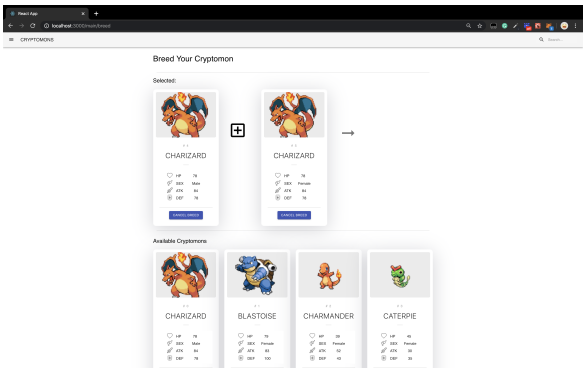
As we just purchased a Bulbasaur, we now can exchange the venusaur with our new bulbasaur.



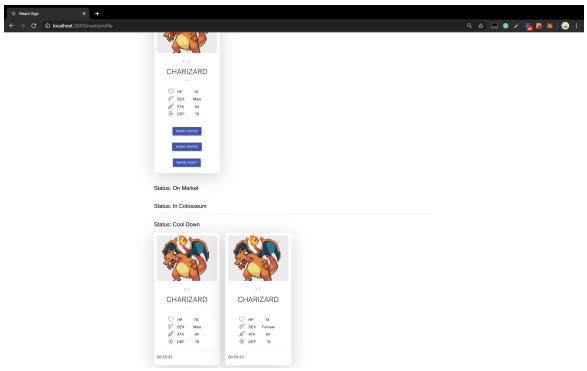

(a) Exchange

(b) Owned

Figure 12: Share Page

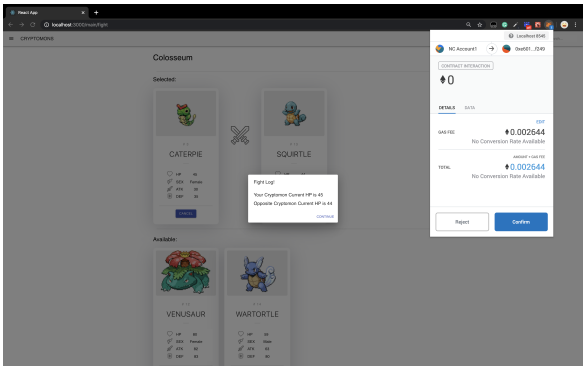



(a) Breed

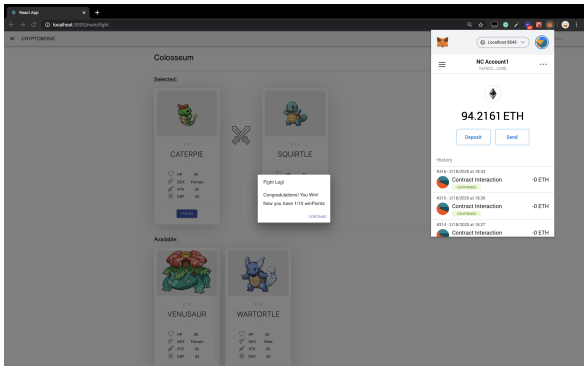



(b) Set Cool Down After Breeding

Figure 13: Breed Page




(a) Fight




(b) Fight Result

Figure 14: Fight Page