To do list

- update readme, you changed part of the project idea.

- learn js, react

- learn how to create dapp with react (with back end server), web3 and if possible, uport.

- learn how to integrate IPFS (https://flyingzumwalt.gitbooks.io/decentralized-web-primer/content/SUMMARY.html)

**Requirements**

● User Interface Requirements:

○ Run the app on a dev server locally for testing/grading

○ You should be able to visit a URL and interact with the application

■ App recognizes current Metamask account and displays it

■ Sign transactions using MetaMask

■ Contract state is updated

■ Update reflected in UI

● Test Requirements:

○ Write 5 tests for each contract you wrote

■ Solidity or JavaScript

○ Explain why you wrote those tests

○ Tests run with truffle test and all must pass and are properly structured

○ All tests must provide adequate coverage

● Design Pattern Requirements:

○ Implement emergency stop

○ What other design patterns have you used / not used?

■ Why did you choose the patterns that you did?

■ Why not others?

● Security Tools / Common Attacks:

○ Explain what measures you’ve taken to ensure that your contracts are

not susceptible to common attacks

● Import sth

○ Library or via EthPM

● Testnet deployment

○ The addresses provided in deployed\_addresses.txt correctly point to deployed contracts on the rinkeby testnet.

● Documentations

○ https://solidity.readthedocs.io/en/v0.4.21/layout-of-source-files.html#comments

● Stretch requirements:

○ uPort integration? (not sure how to link the 2)

○ IPFS integration

○ Cart functionality

○ Review functionality

○ Search functionality

○ Give store owners option to accept any ERC-20 tokens

○ implement in vyper

○ upgradable registry or delegation

○ Oracle (to get price?)

**User stories**

*Admin*

* Marketplace is managed by a group of administrators
* Admin opens web app, web app reads the address and identifies them as the admins, showing them only admin functions.
* Admin functions include
  + Adding and removing addresses from list of store owners with status, approved or unapproved.
  + can list all store owners as well.
  + Adding and removing other admins can only be done by 2 or more admins except for owner of contract.
  + Any admin can approve a storeowner.

*Approved store owners*

* Approved store owner opens web app, web app reads the address and identifies them as the approved store owners, showing them only store owner functions
* Store owner functions includes
  + Create, list, delete storefronts
  + Click on storefront to manage it
  + Add and remove items in storefront
  + Update price, description, quantity of item
  + Withdraw funds that the store has collected from sales

*Unapproved store owners*

* Unapproved store owner opens web app, web app reads the address and identifies them as the unapproved store owners, informing them that their application is in process.

*Registered shoppers*

* Registered shoppers open web app and logs in
* smart contract doesn’t recognise the address as an admin or (un)approved store owner and registers it as a generic shopper.
* Registered shoppers are able to do everything an unregistered shopper.
* Registered shoppers can also do the following:
  + Access cart and proceed to checkout
  + Items purchased will be sent to their purchase history
  + Amount debited from their balance and transferred to the store owner’s account
* Cart functionality (every address has a cart with has X items in it)

*Unregistered shoppers*

* Unregistered shoppers opens web app without logging in.
* They are still able to do the following:
  + See the marketplace (a list of items without name, description, price and quantity).
  + Add search functionality? Search name of product and description?
  + Click on the items to go to the product page which will display the description, quantity in stock and price. (potentially with reviews?)
  + List price in BTC (in sats) but you can convert to ETH (in wei)?
  + In the product page, you can also see the name of the store front, with logo.
  + If you click on the name of the store front, you will be directed to the store’s page which will display:
    - a list of items (with no description, price and quantity)
    - the description of the store
    - logo of the store

They are still able to see the browse stores, items to read their description etc. but they are not able to purchase anything. Clicking on the add to cart functionality will prompt them to log in.

All products need to have the following:

1. Name
2. Description – can be added later, defaults to “please update product description.”
3. Picture – can be added later, defaults to a pic with the words “to be added later”
4. Price, defaults to 0
5. Quantity in stock, defaults to 1. If 0, = out of stock.
6. Review?

All can be added, deleted, updated except reviews unless product is deleted. Can also update quantity.

All stores need to have the following:

1. Name
2. Description – can be added later, defaults to “please update product description.”
3. Logo – can be added later, defaults to a pic with the words “to be added later”
4. List of items – can be added later

All can be added, deleted, updated

uport metamask integration - https://medium.com/swapynetwork/using-uport-to-manage-identity-and-signing-transactions-in-swapy-exchange-fdcffea19326