CryptoTracker

Manage a portfolio of digital assets

What Is CryptoTracker?

CryptoTracker is designed for people who buy and sell decentralised digital assets... or 'cryptocurrency'.

In this MVP release, the user can create a portfolio of cryptos from the top 100 assets listed on CoinMarketCap.com and get real time price data to track their portfolio value.

Target Audience: Who is it for?

Cryptocurrency portfolio applications have existed for for several years.

There are online services that offer this, however most people are not comfortable entering their investments into an unknown website or cloud based service. They prefer anonymity. Many users rely on manually updated Excel spreadsheets.

This app aims to offer a better crypto tracking solution that is local and private, with real time pricing data and other metrics. It was designed to be simple, fast, easy to use and light weight. It requires Ruby and runs on Windows, Mac and Linux systems.

What Functionality Does it Provide?

Users can:

- View their portfolio (cached and live pricing)
- Add and remove crypto assets from their portfolio
- Get help

Admin can:

View users, Add users, Deactivate Users

Screenshots of CryptoTracker

```
+-----------------------------------------------+
| Crypto Portfolio Tracker - You Are Logged Out |
+-----------------------------------------------+
| 1. Login                                      |
| 2. Quit                                       |
+-----------------------------------------------+
```

```
Welcome Tester, you are logged in.
+---------------------------------------------+
| Crypto Portfolio Tracker Main Menu          |
+---------------------------------------------+
| 1. View Portfolio                           |
| 2. Add/Remove Crypto                        |
| 3. Help                                     |
| 4. Quit                                     |
+---------------------------------------------+
```

Screenshots of CryptoTracker

```
+-------------------+---------------+-----------------+-----------------+-----------------+
|                              Tester's Crypto Portfolio                                |
+-------------------+---------------+-----------------+-----------------+-----------------+
| Name              | Symbol        | Quantity        | Price USD       | Total USD       |
+-------------------+---------------+-----------------+-----------------+-----------------+
| Litecoin          | LTC           | 500.0           | $71.00          | $35,501.40      |
| Ethereum          | ETH           | 35.0            | $541.77         | $18,962.10      |
| Stellar           | XLM           | 5000.0          | $0.16           | $793.41         |
| NEM               | XEM           | 50.0            | $0.21           | $10.25          |
| XRP               | XRP           | 5000.0          | $0.55           | $2,772.31       |
|                   |               |                 |                 |                 |
| Grand Total       |               |                 |                 | $58,039.47      |
+-------------------+---------------+-----------------+-----------------+-----------------+
Data supplied by CoinMarketCap.com

Username: tester
2020-12-17 18:38
Your portfolio has a total of 5 digital assets.
```

Screenshots of CryptoTracker

```
Welcome Admin, you are logged in.
+-------------------------+
| Crypto Tracker Admin    |
+-------------------------+
| 1. Create User          |
| 2. Show All Users       |
| 3. Deactivate User      |
| 4. Quit                 |
+-------------------------+


* Create new user *
Name:
Gary
Username:
gazza
Password:
pass123
You are about to add a new user. Are you sure? y/n
```

```
+---------+----------+-----------+--------+------------+
|                       Users                         |
+---------+----------+-----------+--------+------------+
| Name    | Username | Pass      | Active | Joined     |
+---------+----------+-----------+--------+------------+
| Admin   | fusion22 | fusion22  | true   | 2020-01-01 |
+---------+----------+-----------+--------+------------+
| Nick    | graycode | pass1234  | true   | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Jay     | jay      | 1234      | false  | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Henry   | henzo    | pass1234  | true   | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Mary    | mazzy    | pass1234  | true   | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Glen    | glenfish | pass1234  | true   | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Kim     | kim      | pass1234  | true   | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| John    | jank     | 1234      | true   | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Liz     | liz55    | 1057F%8yC3# | true | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Bon     | bon      | 1234      | false  | 2020-12-15 |
+---------+----------+-----------+--------+------------+
| Zuma    | zoom     | 1234      | true   | 2020-12-16 |
+---------+----------+-----------+--------+------------+
| Test    | test     | test      | false  | 2020-12-16 |
+---------+----------+-----------+--------+------------+
| Ron     | ron      | 1234      | true   | 2020-12-16 |
+---------+----------+-----------+--------+------------+
| Gonzo   | gonzers  | pass1234  | true   | 2020-12-16 |
+---------+----------+-----------+--------+------------+
| Scott   | djscotty | 1234      | true   | 2020-12-16 |
+---------+----------+-----------+--------+------------+
| Jim     | jimbo    | 1234      | false  | 2020-12-16 |
+---------+----------+-----------+--------+------------+
```
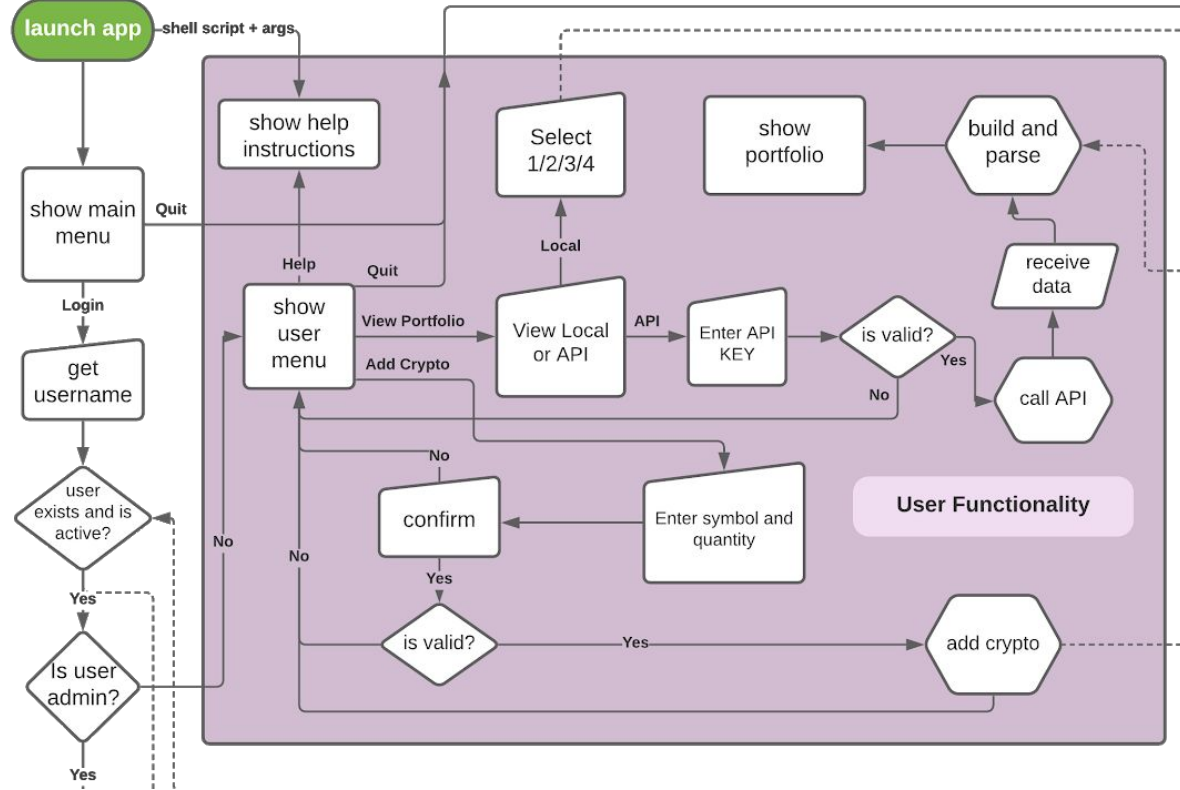
Flow Control
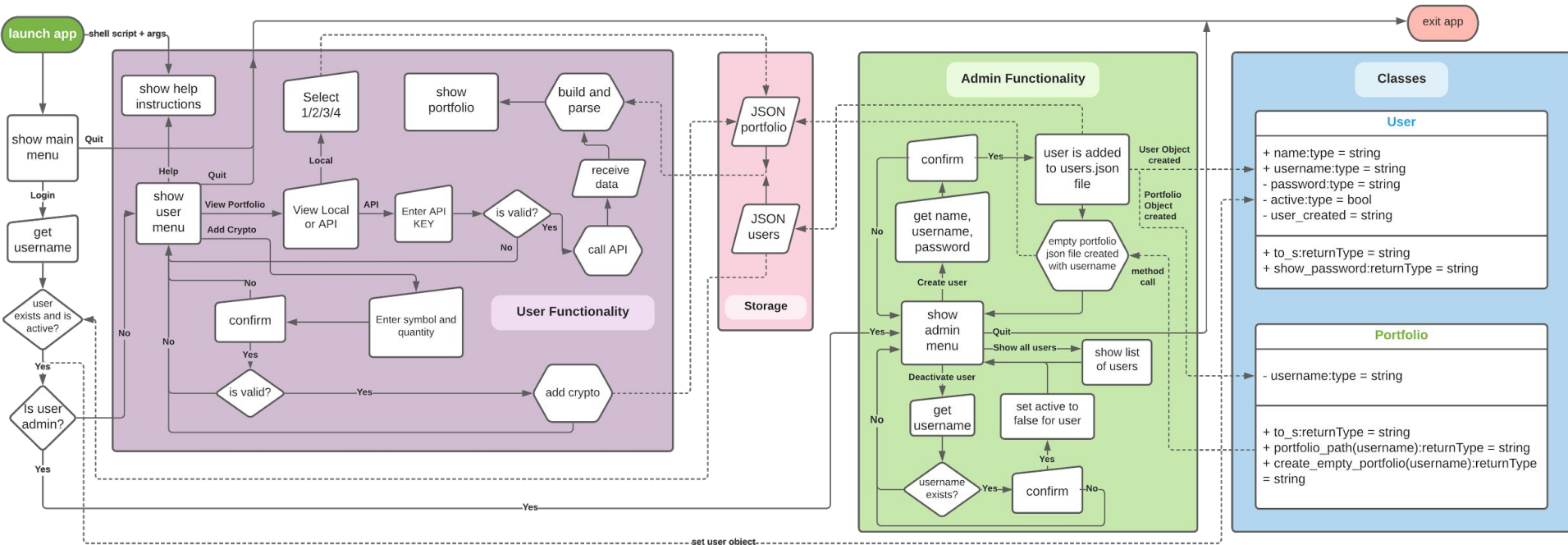
Flow Control: Users

User

Portfolio Management

View Portfolio

- Name
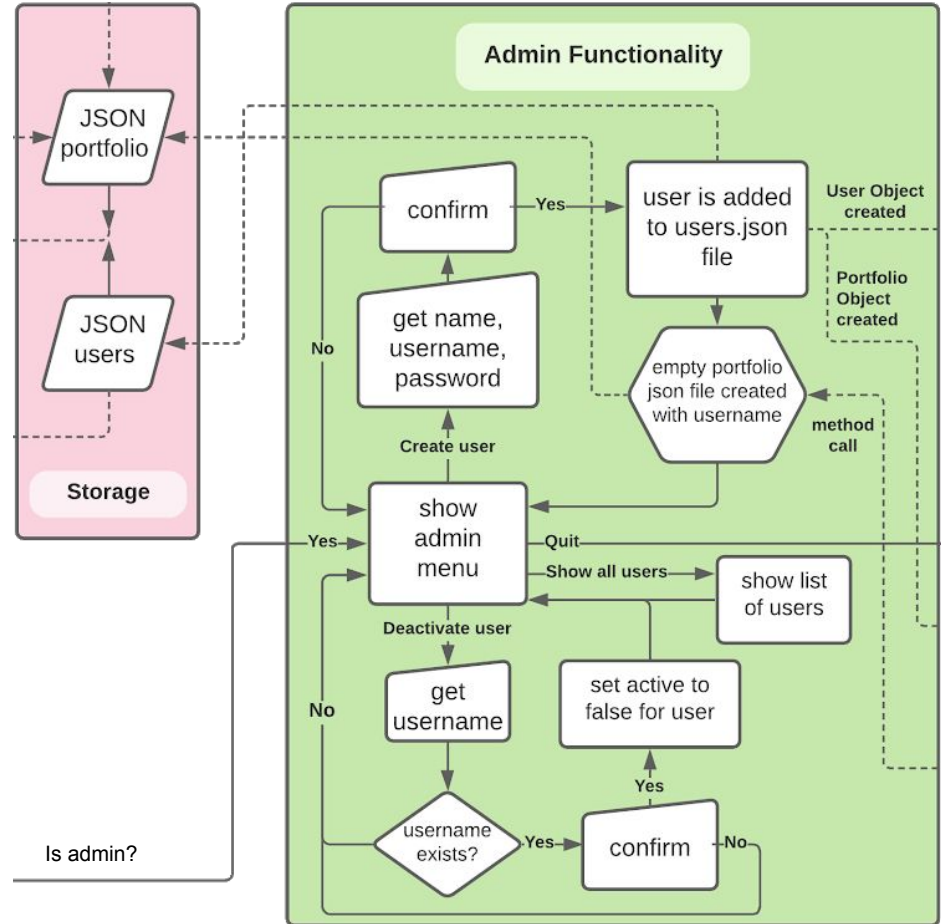- Symbol
- Quantity
- Price
- Value
- Grand Total

Flow Control: Users

Admin

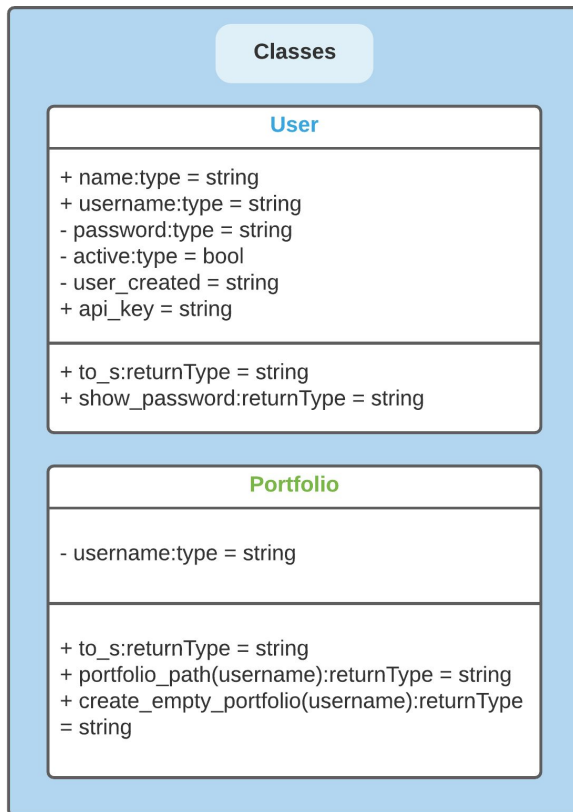Handles user creation and management

Interacts with JSON storage and menu

Classes & UML

User class used for user object attributes handling for the session

Portfolio class used for setting up a new portfolio And managing portfolio json file path

Classes

User

+ name:type = string
+ username:type = string
- password:type = string
- active:type = bool
- user_created:type = string
+ api_key = string

+ to_s:returnType = string
+ show_password:returnType = string

Portfolio

- username:type = string

+ to_s:returnType = string
+ portfolio_path(username):returnType = string
+ create_empty_portfolio(username):returnType = string

```ruby
class Portfolio
    attr_reader
    attr_writer
    attr_accessor

    def initialize(username)
        @username = username
    end

    # methods
    def to_s
        return @username
    end

    def portfolio_path
        return "./json/portfolios/#{@username}.json"
    end

    def create_empty_portfolio
        return {"username":"#{@username}","data":{"BTC":{"asset_name":"","asset_quantity":0,"asset_buy_date":"2020-12-14","asset_sell_date":"","usd_price":"","btc_price":"","usd_profit":"","btc_profit":""}}}
    end

end
```

Variables & Structure

Main.rb handles high level flow. Menu.rb handled all logged-out, User logged-in and Admin logged-in menu displays and selection handling

An array was returned from each menu selection which included the login state, the current menu action and any additional class objects such as the logged in user.

The *user object* was made available by passing it around to various methods.

```ruby
# logged in User menu selection handling
def logged_in_menu_selection(selection)
    clear
    case selection
    when 1
        # portfolio
        clear
        return [true, "show_portfolio"]
    when 2
        # add a crypto
        return [true, "add_crypto"]
    when 3
        # help
        clear
        return [true, "show_help"]
    when 4
        # quit
        return [false, "exit"]
    else
        return [true, "error"]
    end
end
```

Classes

The User class is called when a new user is created by an Admin account. This makes the user's data available for that session.

A Portfolio object is created which calls a class method to builds a blank portfolio.

```ruby
class User
    attr_reader :name, :username, :active, :api_key
    attr_writer
    attr_accessor

    def initialize(name, username, password, api_key)
        @name = name
        @username = username
        @password = password
        @api_key = api_key
        @active = true
        @user_created = Time.now.strftime("%Y-%m-%d")
    end



    # methods
    def to_s
        return "Name: #{@name} Username: #{@username}
 Password: ******** Active: #{@active} Created:
#{@user_created}"
    end

    def show_password
        return @password
    end

end
```

Application Programming Interface

- The API key is stored in a text file which is read by a method for security

- Cryptocurrency data for portfolios are stored in unique json files

- Four API calls for cryptocurrency data were stored locally for testing in the api_cached directory

- API data for cached files was limited to 100 of the top crypto/assets as listed on CoinMarketCap.com

BTC,ETH,XRP,USDT,BCH,LTC,LINK,ADA,DOT,BNB,XLM,USDC,BSV,EOS,XMR,WBTC,TRX,XEM,XTZ,LEO,FIL,CRO,NEO,DAI,VET,REV,ATOM,AAVE,DASH,WAVES,HT,MIOTA,UNI,ZEC,ETC,YFI,THETA,BUSD,COMP,CEL,MKR,SNX,OMG,DOGE,UMA,KSM,FTT,ONT,ZIL,ALGO,SUSHI,OKB,BTT,BAT,TUSD,RENBTC,DCR,NEXO,ZRX,DGB,PAX,HUSD,AVAX,REN,QTUM,HBAR,AMPL,ICX,ABBC,CELO,LRC,EGLD,HEDG,STX,LUNA,KNC,RSR,REP,EWT,LSK,OCEAN,BTG,SC,QNT,RUNE,CVT,NANO,BAND,MANA,ZB,NMR,ENJ,ANT,MAID,SNT,CHSB,XVG,NXM,RVN

Code: Shell Script flags

crypto.sh will run the app, which points to main.rb

As of this time, the '-h' flag is available to display help information when run via the shell script or directly.

Any of these will show help to a non logged in user:

crypto.sh -h
crypto.sh --h
crypto.sh -help
crypto.sh --help
ruby main.rb -h
ruby main.rb --h
ruby main.rb -help
ruby main.rb --help

help

Placeholder text

```
This is help content. This is help content. This is help content. This
is help content. This is help content. This is help content. This is he
lp content. This is help content. This is help content.

This is help content. This is help content. This is help content. This
is help content. This is help content. This is help content. This is he
lp content. This is help content. This is help content. This is help co
ntent. This is help content. This is help content. This is help content
. This is help content. This is help content. This is help content.

– This is help content.
– This is help content.
– This is help content.

This is help content. This is help content. This is help content.

[johndoe@glenfishmac:~/Documents/CryptoTracker/src]$
```

Additional flags for version and install info will be made available.

DRY Code ♻

Methods were broken up into logical groupings for portfolio and user operations and those files included in menu, which was in turn required in main.

Other methods such as json handling, api methods and help were put in their own files.

The Portfolio and User classes were both added to the classes.rb file

One users.json file contains all the user credentials.

```
require_relative '../api/api'
require_relative 'classes'
require_relative 'json-read-write'
require_relative 'portfolio'
require_relative 'help'
require_relative 'user'
```

Error Handling

- Decision to minimise the amount of onscreen error messaging

- Validation errors have been handled with rescue and rescue retry

- Invalid menu choices re-display the previous menu

- Incorrect API KEY - specific error message is displayed to alert the user

Ruby Gems

The following gems are used:

httparty   handles API requests

json   handles writing and reading

terminal-table   displays data in a table structure in the terminal

colorize   adds colour to text

```ruby
require 'httparty'
require 'json'
require 'terminal-table'
require 'colorize'
```

Github

Github was used for the project from the file structure through
to the finished MVP.

Everything was coded on 'main' branch, with the exception of
code cleanup which was done on 'refactoring' and then merged
locally.
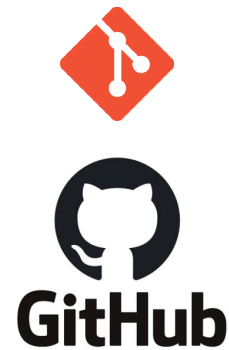
Trello Project Management

REQUIREMENTS

Files Required - checklist
☑ 5/5

README.md checklist
☑ 1/5

Software Development Plan - checklist
☑ 0/20

Slide Deck
☑ 0/6

Code Requirements
☑ 0/18

accept user input in the form of a file or text input
💬 1 ☑ 1/1 ⏱ 10m

produce printed output or interact with the file system
💬 1 ☑ 1/1 ⏱ 10m

+ Add another card

Backlog Donut

Handle Help selection for users who are not logged in and non-users

add arguments/flags to shell script

do manual testing and take screenshots

create slide-deck. take screenshots of system in action

+ Add another card

Backlog Sprinkles

write rspec tests

Add a subclass for admins and assign priviliges

create a Manage Users method for admins which includes activating and deleting
☑ 1/3

find a gem for producing PDFs or printable output and test

Additional/optional data for portfolio 24 hr change, market cap

only allow new API calls if more than 10 minutes since last call

+ Add another card

Bugs & Issues

in admin menu, entering a number that isnt a menu selection causes error

+ Add another card

In Progress

+ Add a card

To Do Today

Build Software Development Plan

Define Project in Readme.md file
📄 ⏱ 33m

+ Add another card

Testing

+ Add a card

Done

make control flow + UML document

when selecting 'help' and not logged in, it allows logged in access. Need to handle

Handle errors

create Portfolio Class
☑ 0/14

Method to add new digital assets to portfolios
📄

Method to add new portfolios
📄

change login greeting to show person's name

JSON handling w/Classes
📄 ☑ 6/6 ⏱ 1h 44m

create User Class

Admin can deactivate users
⏱ 30m

check to see if user exists before adding to users.json
⏱ 39m

change bottom row Total to Grand Total

Method to Add new users to Users JSON
☑ 1/1 ⏱ 30m

Store most recent API call as

+ Add another card

Development Issues

- store the data in a JSON format

- Making class attributes accessible to all my methods

- Testing with rspec - I did a few initial tests

?

Ethical Issues

Privacy of user data, especially financial information and investments is critical to users. At this stage of development, CryptoTracker is currently a working proof of concept.

The API key and user credentials are currently stored unencrypted in text and JSON files.

It is intended that password handling and secure storage and authentication is added in the next sprint.

CryptoTracker Live Presentation

If you want to follow along with the demo and see
real-time market prices reflected in the app, go to:

CoinMarketCap.com