**Andrew Trail Crypto-Currency Tracker**

**Brief**

An application to allow users to store and track their crypto currency holdings. The software will have a GUI. It will allow users to input quantities purchased and save and read to and from an external database. They will have the option to output a pie chart of their holdings to another external file. The application will display a total portfolio value, along with a detailed breakdown.

**Analysis**

**Functional Requirements:**

Initial form requires a database to be selected or created

Edit or add balances for each coin

Total folio price displayed

Folio display currency can be changed (USD/GBP/EUR)

Can open different database file from main form

Can save current folio to new database file

Generate a pie chart of monetary value of crypto holdings and save to external file.

Displayed with a GUI

Display current value of all currencies included

**Non-Functional Requirements:**

Should be saved as a portable .exe file

Code should be modular, with clear comments and meaningful variable names

The program should not use a large amount of processing power, or storage space

Planning to be completed by 2/3/18

Development to be completed by 11/5/18

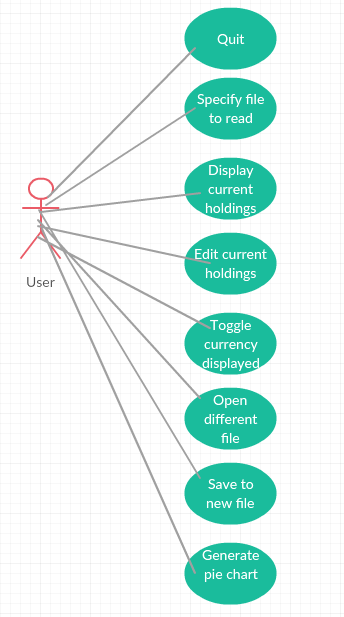
Presentation to be given between 14/5/18 and 25/5/18

Evaluation due by 1/6/18

**Project Aim:**

The aim of this project is to create a simple, intuitive desktop application to keep track of user crypto currency holdings.

**Top Level Use Case**



**Background Research**

I have studied a similar application to this, and this helped me to decide what I would

include with my application. The app is called Blockfolio, and it is an Android/iOS app for

mobile phones.

I felt the interface had a clunky, unintuitive design in places, and that I could improve upon it. In addition, I wanted to create a desktop application, rather than a mobile app, for use when I’m at home on my desktop.

The application also does not have the ability to manage multiple user accounts, which I will

be including. Finally, there is no option to display a visual breakdown of the distribution of your crypto holdings.

**Resources**

Python 3 – I have elected to use python as it has readily available packages for many functions, such as the coinmarketcap API, and plotly for generating a pie chart. In addition I am familiar with the language.

Creatly – I will use this as it is a free online environment to generate UML diagrams

Tomsplanner – This is an online environment for creating a Project plan, with detailed times, deadlines. Can also display this information in a Gantt Chart

App.jar – This is a library I will use within Python in order to generate the GUI

Sqlite3 library – Used to create and edit database files

Json library – used to scrape GBP/USD/EUR price ratios from the net

Coinmarketcap library – Used to get current coin prices from their webpage

Matplotlib library – Used to generate charts

Axure – Used to generate the wireframes

**Information Sources**

I am familiar with Python; however, will undoubtedly encounter problems I haven’t seen before. To remedy any such problem, I will use primarily stackoverflow.com, and if I can’t find the solution I require I will move to google.

I will also use the matplotlib webpage which shows lists of commands to generate the chart I require.

Finally, I will use the documentation provided alongside the coinmarketcap API for command lists also.

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Attribute** | **Data type** | **Constraints/comments** |
| Coin | Varchar(10) | Primary | Not null |
| Price | Float | Secondary | Not null |
| Holdings | Float | Secondary | Not null |
| Holdings\_Value | Float | Secondary | Not Null |

**Data analysis**

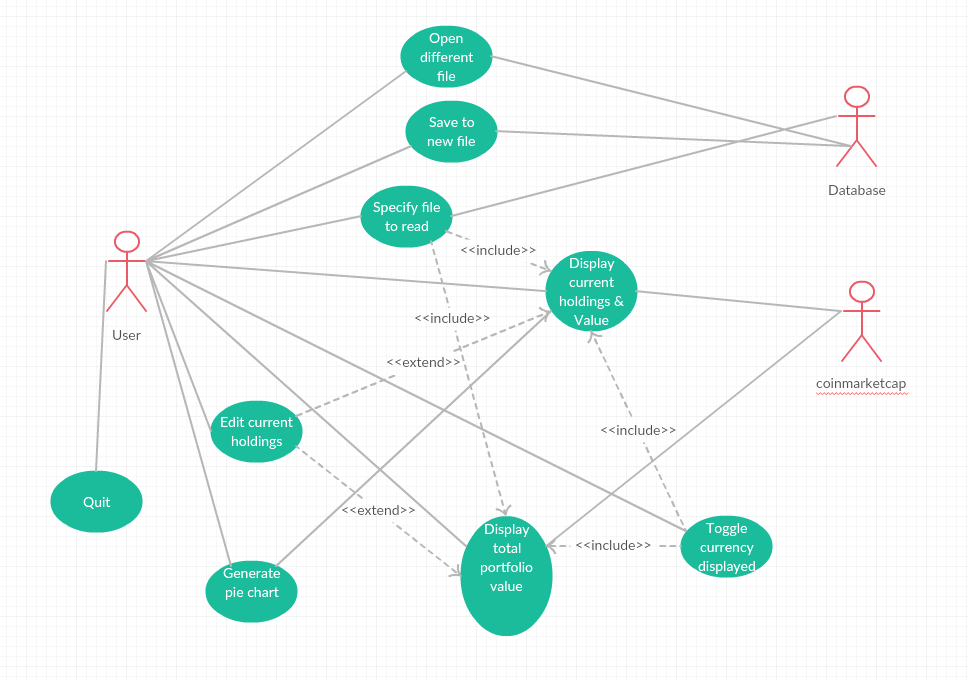
Coin: This is the primary key, used to identify the coins by name.

Price: This key is a secondary key and a float. I opted not to use the currency type here as calculations need to be done on the price within the code. Instead the output is formatted to include currency symbols and rounding to 2 decimal places. This method also improves accuracy when it comes to calculations and conversions.

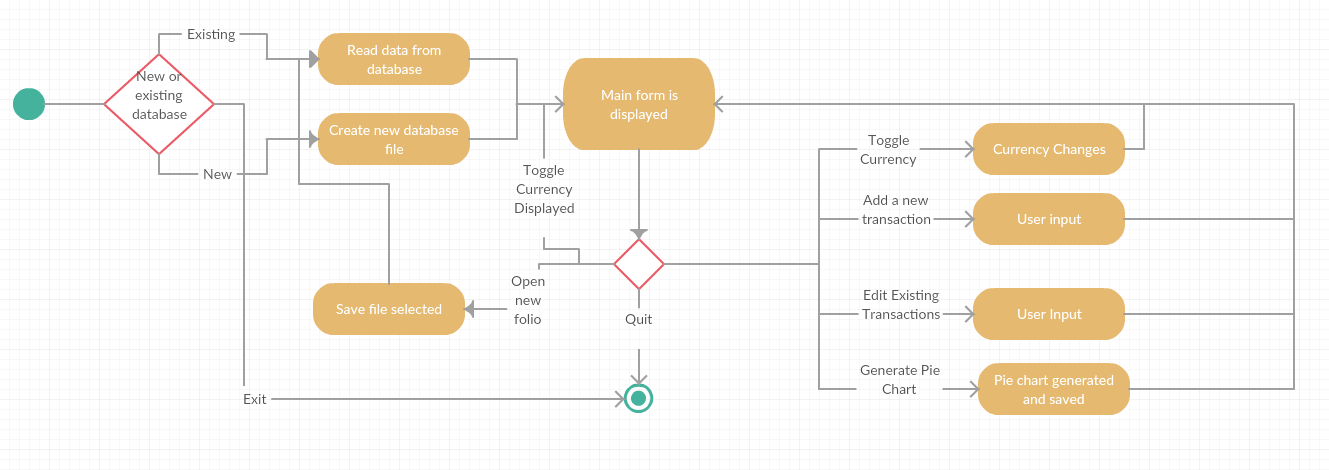
Holdings: This key is a secondary key and a float. Used to hold the users’ current holdings of each coin.

Holdings\_Value: This is a secondary key and a float. It’s used to hold the users’ holdings value. Again, I opted not to use the currency type, as calculation are done within the code.

**Detailed Use Case Diagram**

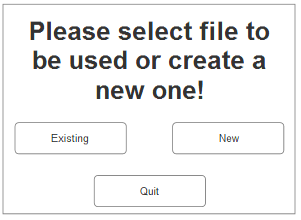


**Activity Diagram**

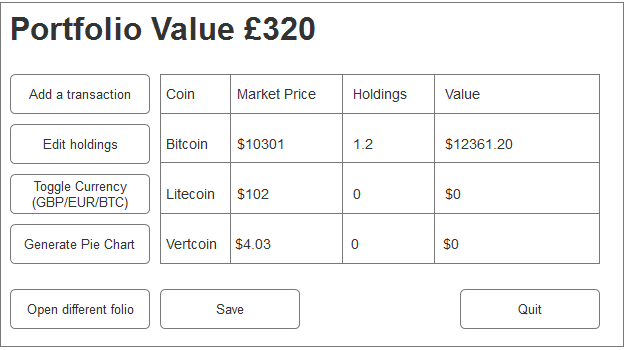


**Interface Design**

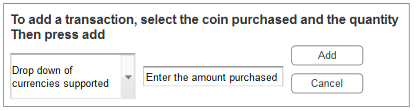
**Initial Form**



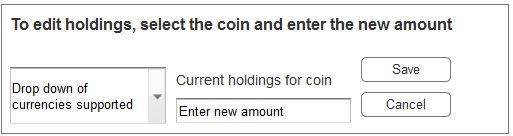
**Main Form**



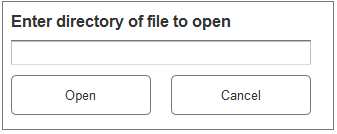
**Add Transaction Form**



**Edit Transaction Form**



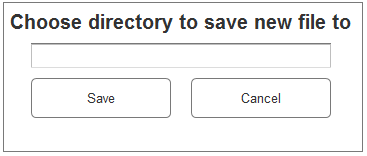
**File selection form**



**Save dialog**



**New file dialog**



**Project Plan:**

