**Andrew Trail Crypto-Currency Tracker**

**V2 – See previous submission for anything missing here**

Contents

[Background Research 1](#_Toc513715636)

[Resources: 1](#_Toc513715637)

[Project Aims 1](#_Toc513715638)

[Data analysis 1](#_Toc513715639)

[Data Dictionary 1](#_Toc513715640)

[Project Plan Updated with Milestones 2](#_Toc513715641)

[Detailed Use Case 2](#_Toc513715642)

[Interface Design 2](#_Toc513715643)

# Background Research

I did some research into different ways to develop a GUI for python. The method I am using (appJar) seems to be the simplest, and given both the timescale and my previous experience this is what led to my decision to use it.

Some of the packages I studied do have more functionality and would lead to a better UI. In a later version these could be used. PyGObject and PyQt were the two that appeared to be best suited for my uses.

# Resources:

Sqlite3 library – Used to create and edit database files

Matplotlib library – Use to generate charts

# Project Aims

* Create a simple interface to keep track of currencies
* Allow users to see real-time prices of crypto currencies
* Improve coding ability and knowledge of python and libraries

# 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.

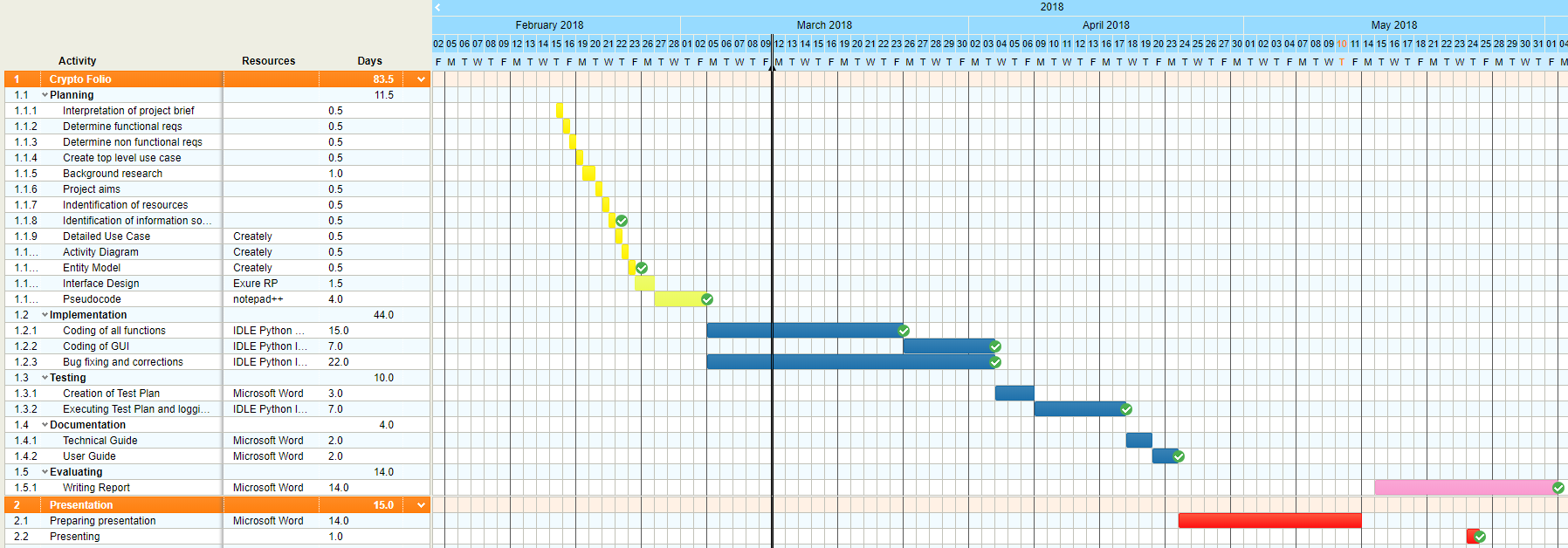
Holdings: This key is a secondary key and a float. Used to hold the users’ current holdings of each coin.

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.

# Data Dictionary

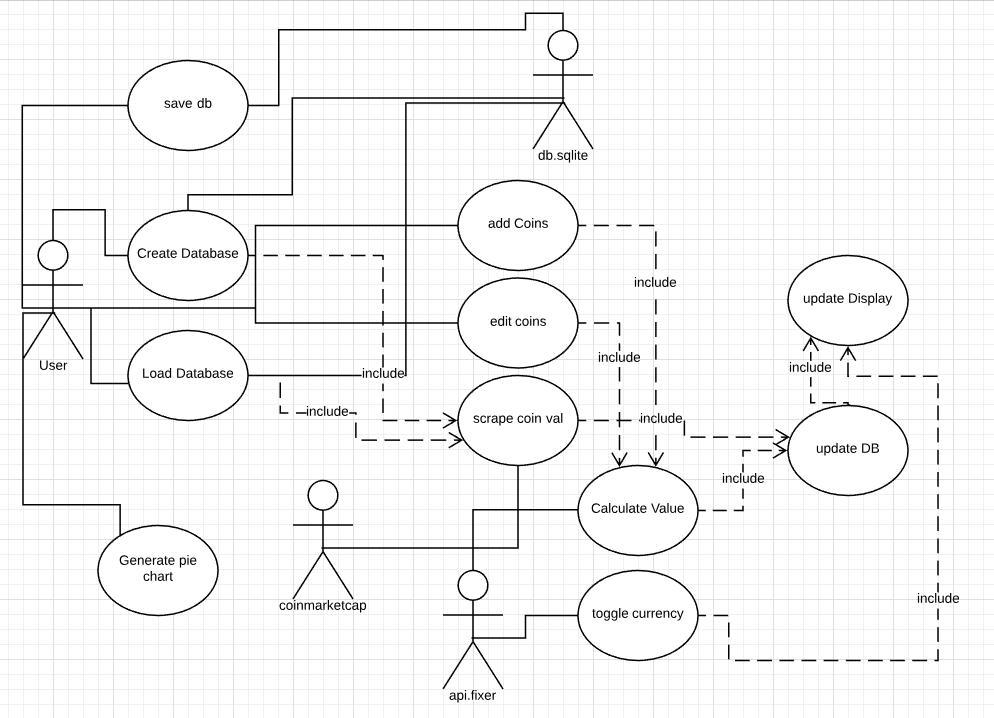
|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Attribute** | **Data type** | **Constraints/comments** |
| Coin | Varchar(10) | Primary | Not null |
| Price | Float | Secondary | Not null |
| Holdings | Float | Secondary | Not null |
| Holdings\_Value | Float | Secondary | Not Null |

# Project Plan Updated with Milestones



NB – Green ticks indicate when a milestone should have been reached.

# Detailed Use Case



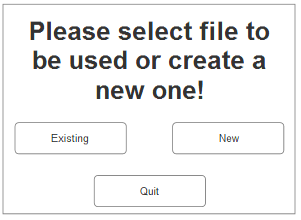
# Interface Design

The interface is largely as designed previously, but due to some constraints with appJar there are a couple of differences which will be noted alongside the wireframes.

It was designed to be clear for the user to use without checking the help documents provided, however these documents are provided if the user needs them.

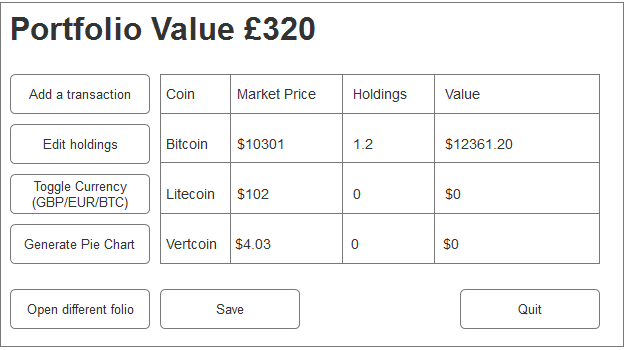
**Initial Form:**

* “Existing” button takes you to [a windows open dialog](#Open) window to choose a database file.
* “New” button takes you to [a windows save dialog](#Save) window to create a database file
* “Quit” button exits the program
* Once either existing or new is selected, and a database is created or loaded the main form opens



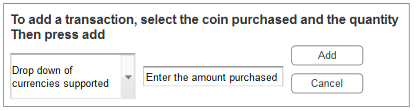
**Main Form**

* “Portfolio Value £xxx” at the top is the grand total of all holdings entered into your folio
* “Add a transaction” takes you to the [Add Transaction](#Add) form to add coins to your folio
* “Edit holdings” takes you to the [Edit](#Edit) Transaction form to edit your coins in folio
* **“**Toggle Currency” toggles the display currency between GBP, EUR and USD
* “Generate Pie Chart” opens [a windows save dialog](#Save) to choose a location to save a pie chart of holdings
* **“**Open Different Folio” opens [a windows open dialog](#Open) to choose a different database to load
* “Save” has the same functionality as “Save As” in most windows apps and allows you to save the current database to another file without destroying the copy currently open
* “Quit” exits the program

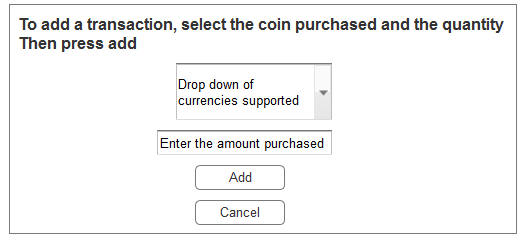


**Add Transaction Form**

* Form has changed. See updated version below.
* Due to limitation with appJar the options are displayed vertically as opposed to horizontally. Still provides a clear flow for the user.

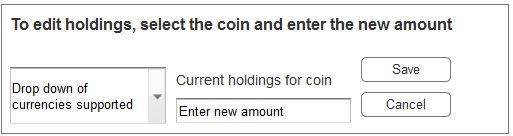


**New Add Transaction form:**

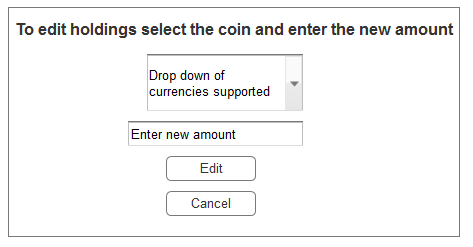


**Edit Transaction Form**

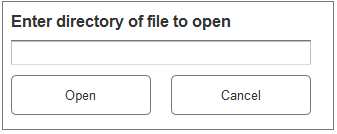
* Form has changed. See updated version below.
* Due to limitation with appJar the options are displayed vertically as opposed to horizontally. Still provides a clear flow for the user.



**New Edit transaction form**



**Old File selection form**

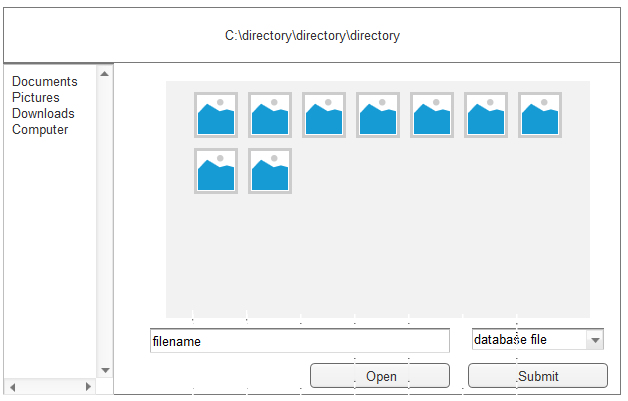


**Old Save dialog**



**New File selection form**

* Should improve usability for the user with default windows open and save forms as they will likely have used these before.



**New Save form**

