

**Evidence Gathering Document for SQA Level 8 Professional Developer Award.**

This document is designed for you to present your screenshots and diagrams relevant to the PDA and to also give a short description of what you are showing to clarify understanding for the assessor.

Each point that required details the Assessment Criteria (What you have to show) along with a brief description of the kind of things you should be showing.

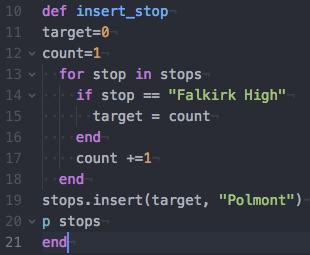
Please fill in each point with screenshot or diagram and description of what you are showing.

**Week 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **I&T** | I.T.5 | Demonstrate the use of an array in a program. Take screenshots of:  \*An array in a program  \*A function that uses the array  \*The result of the function running | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

****

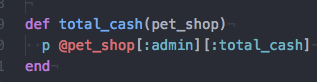
****

**Description here**

“insert\_stop” function inserts the stop “Polmont” at a point in the stops array after stop “Falkirk High”

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **I&T** | I.T.6 | Demonstrate the use of a hash in a program. Take screenshots of:  \*A hash in a program  \*A function that uses the hash  \*The result of the function running | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

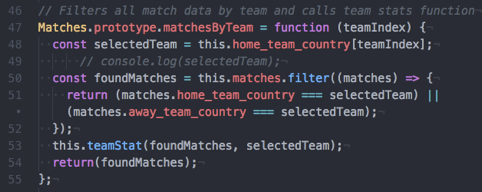
**Description here**

Function retrieves the value of “:total\_cash” from the admin hash within the @pet\_shop hash

**Week 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **I&T** | I.T.3 | Demonstrate searching data in a program. Take screenshots of:  \*Function that searches data  \*The result of the function running | |
|  |  | **Description:** | |

**Paste Screenshot here**

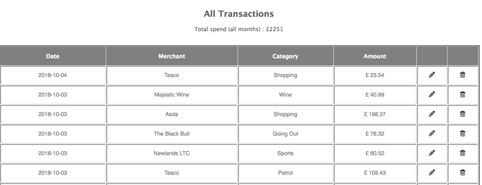
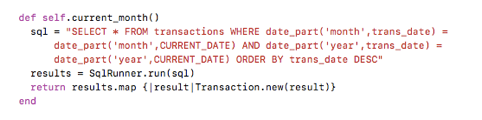
****

**Description here**

Function searches through an array of all matches at the recent football world cup then displays all matches (and various statistics) for the team selected by the user via a dropdown menu.

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **I&T** | I.T.4 | Demonstrate sorting data in a program. Take screenshots of:  \*Function that sorts data  \*The result of the function running | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

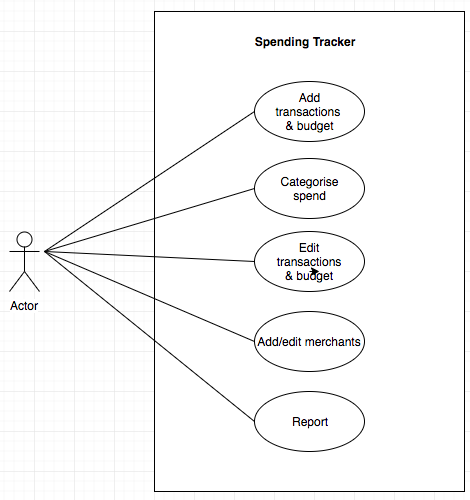
**Description here**

Function selects transactions where the transaction month equals the selected month then sorts the transactions in date ascending order before returning the transactions.

**Week 5 and 6**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **A&D** | A.D.1 | A Use Case Diagram | |
|  |  | **Description:** | |

**Paste Screenshot here**

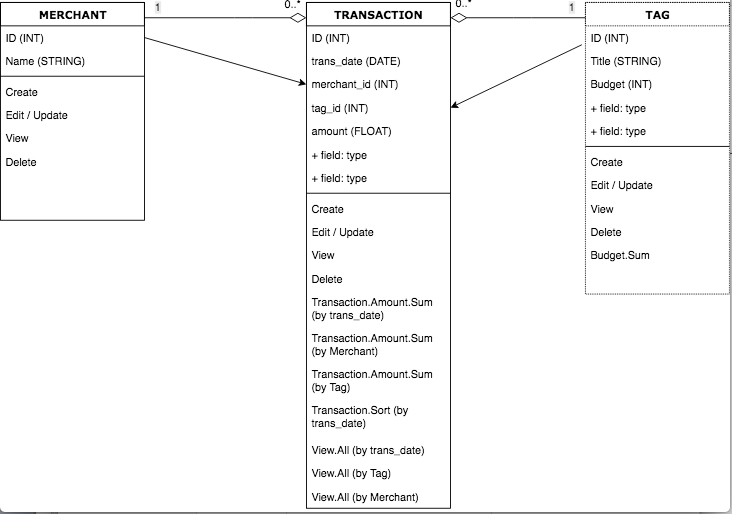
****

**Description here**

User Diagram shows a set of actions a user of my Week5 Spending Tracker can undertake.

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **A&D** | A.D.2 | A Class Diagram | |
|  |  | **Description:** | |

**Paste Screenshot her**

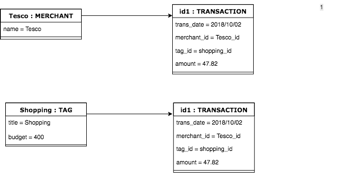
****

**Description here**

Class diagram of my Spending Tracker app showing the 3 classes – Merchant, Transaction and Tag (later changed name to budget in the app).

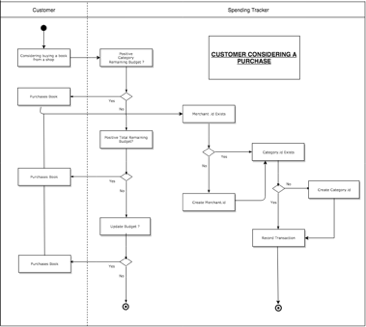
|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **A&D** | A.D.3 | An Object Diagram | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **A&D** | A.D.4 | An Activity Diagram | |
|  |  | **Description:** | |

**Paste Screenshot here**

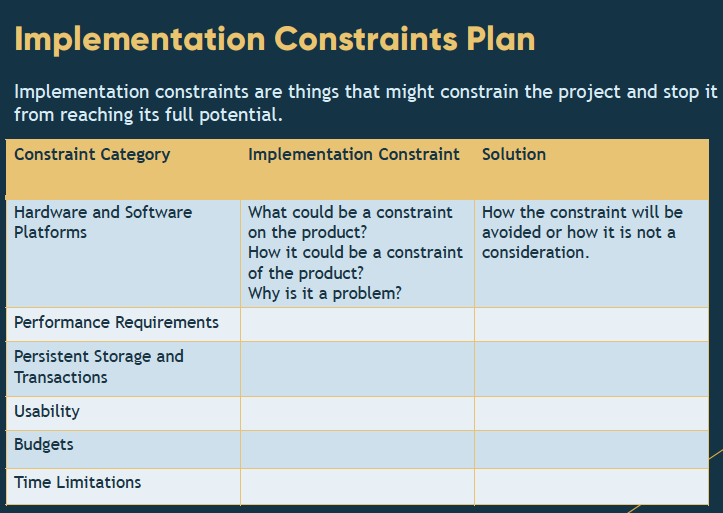
****

**Description here**

Diagram show user interacting with the spending tracker app when contemplating and then executing a transaction.

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **A&D** | A.D.6 | Produce an Implementations Constraints plan detailing the following factors:  \*Hardware and software platforms  \*Performance requirements  \*Persistent storage and transactions  \*Usability  \*Budgets  \*Time | |
|  |  | **Description:** | |

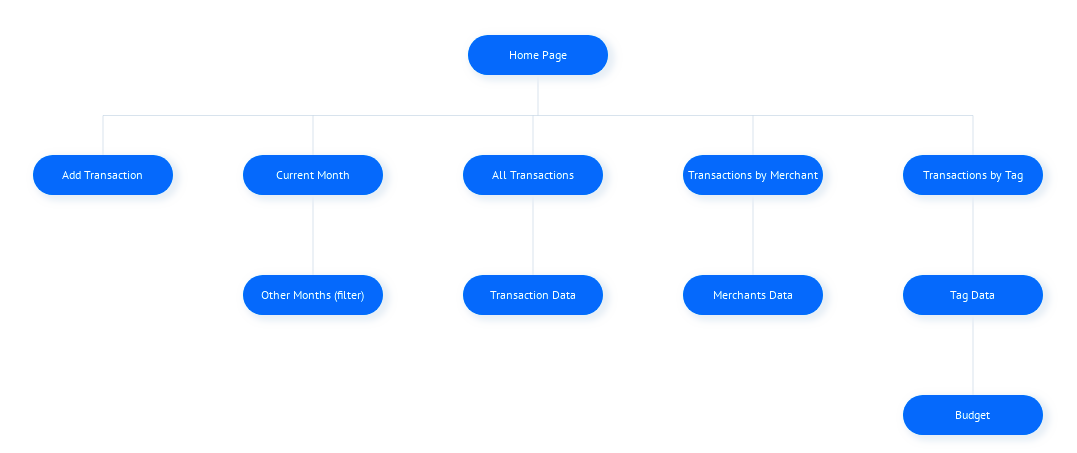
**Paste Screenshot here**

****

**Description here**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.5 | User Site Map | |
|  |  | **Description:** | |

**Paste Screenshot here**

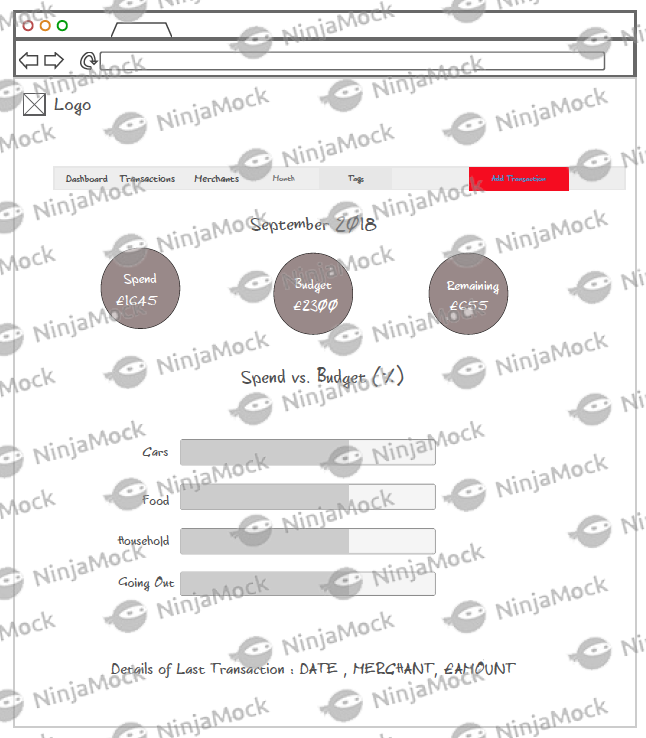
****

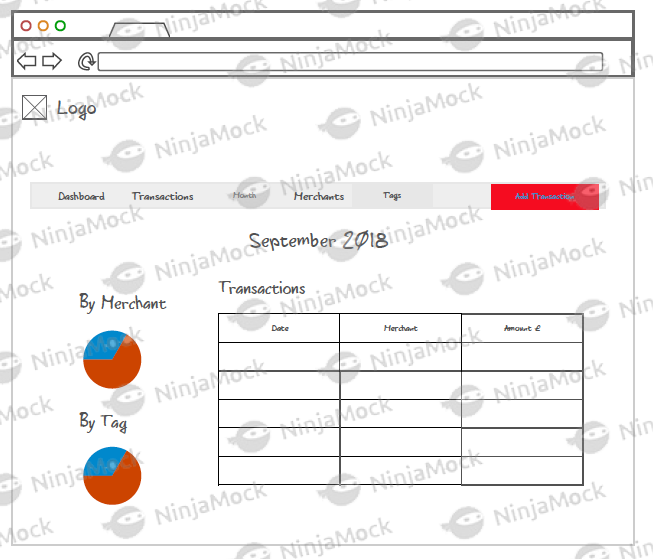
**Description here**

Site map of my spending tracker app showing the various view/pages.

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.6 | 2 Wireframe Diagrams | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

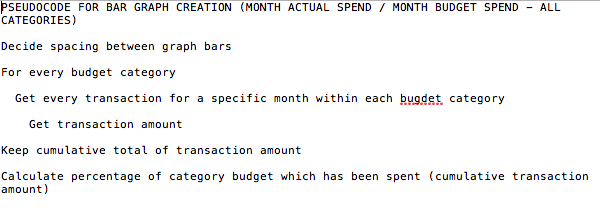
****

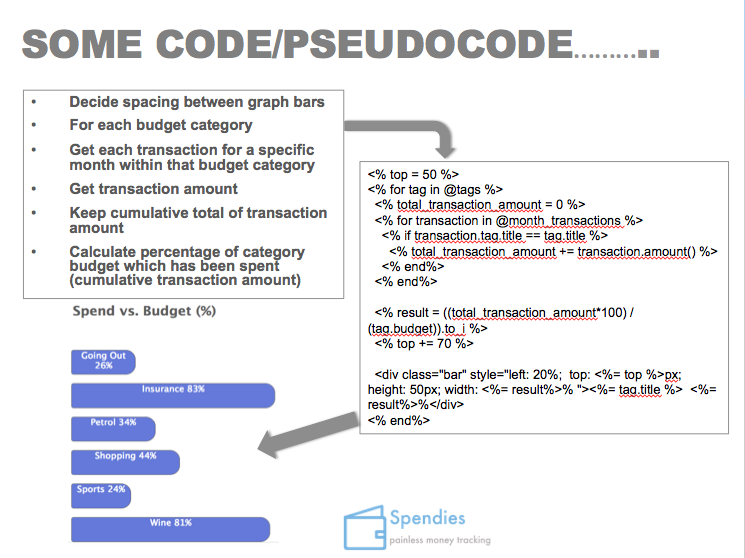
**Description here**

Wireframes of the spending tracker app showing 1) Home Page and 2) Page showing the spend for a given month

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.10 | Example of Pseudocode used for a method | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

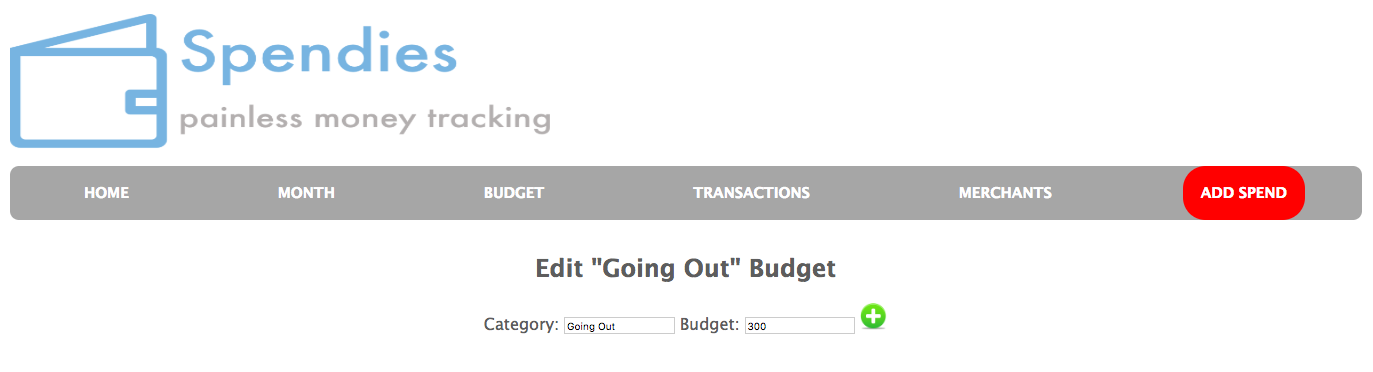
****

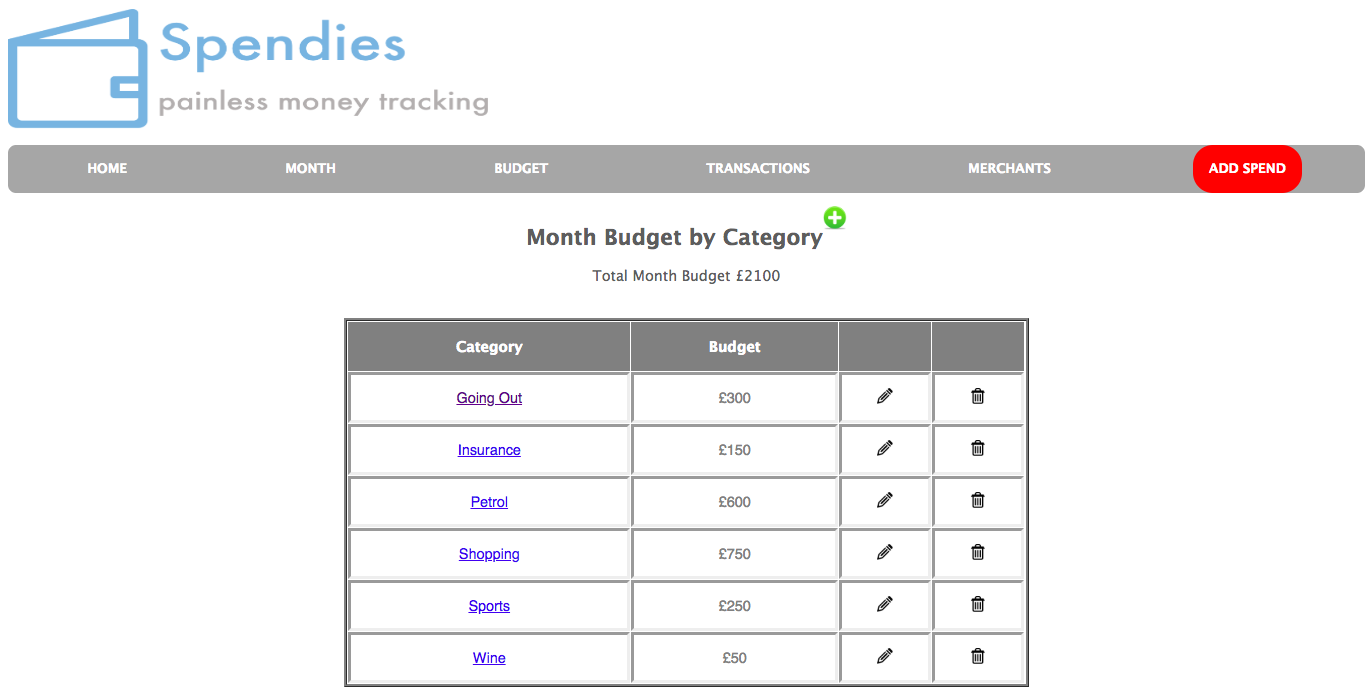
**Description here**

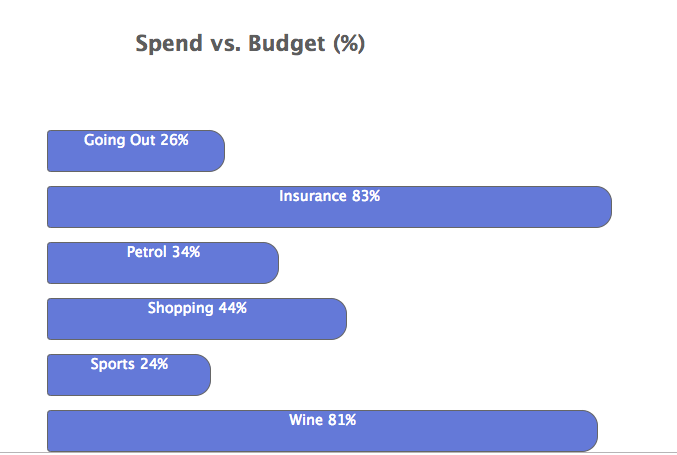
Pseudocode to create a bar graph in ruby which would then show the percentage of the month’s budget for a given category which has actually been spent in the month to date.

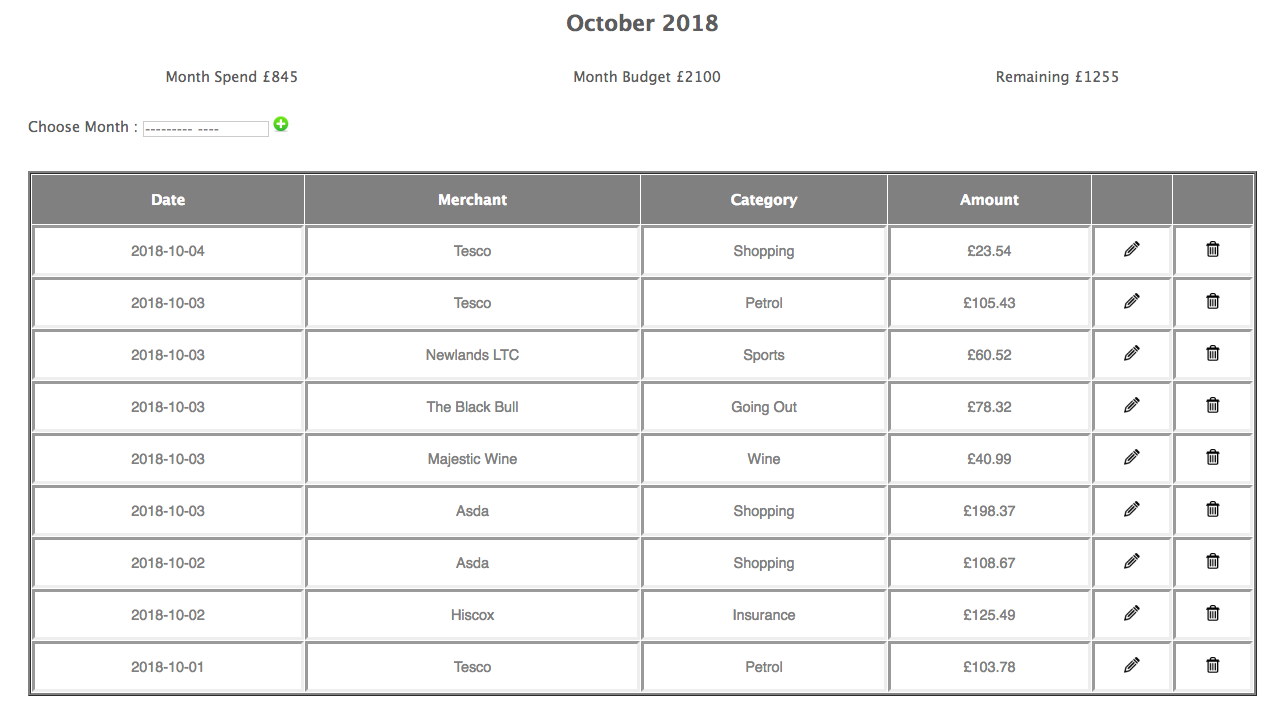
|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.13 | Show user input being processed according to design requirements. Take a screenshot of:  \* The user inputting something into your program  \* The user input being saved or used in some way | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

****

****

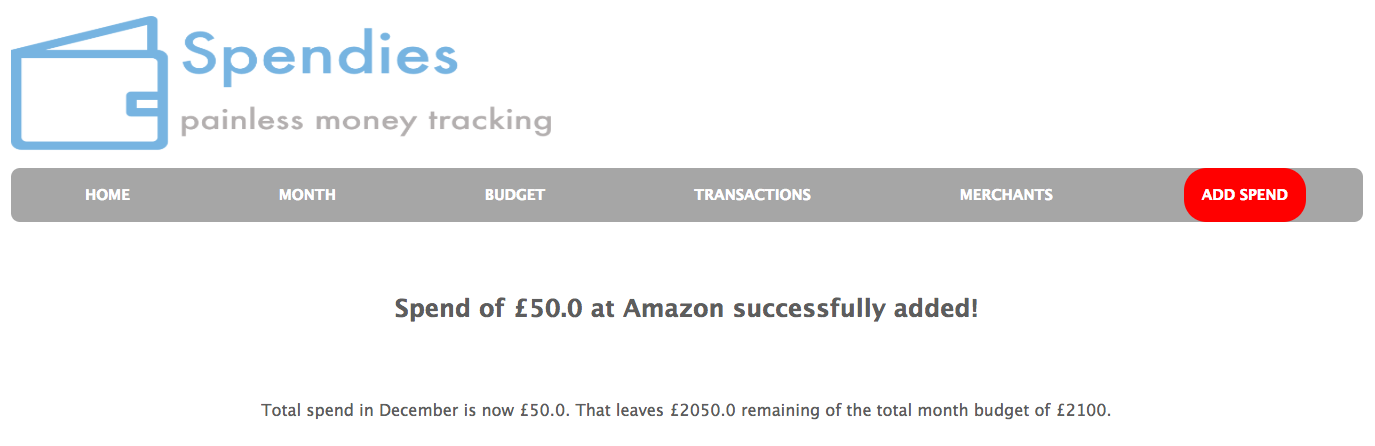
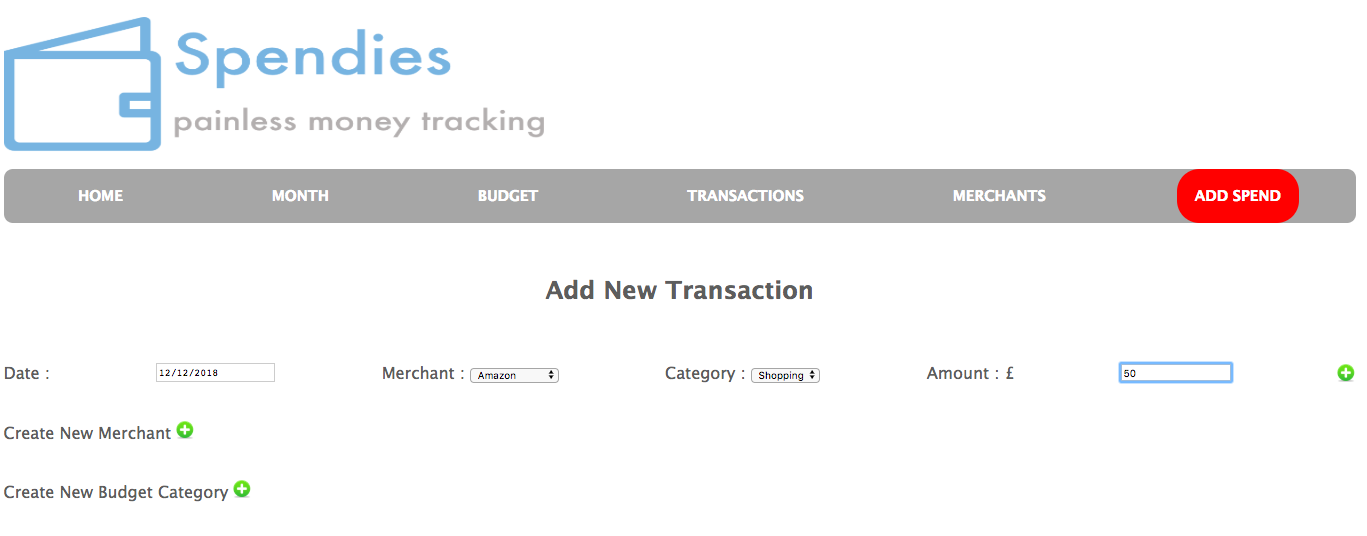
****

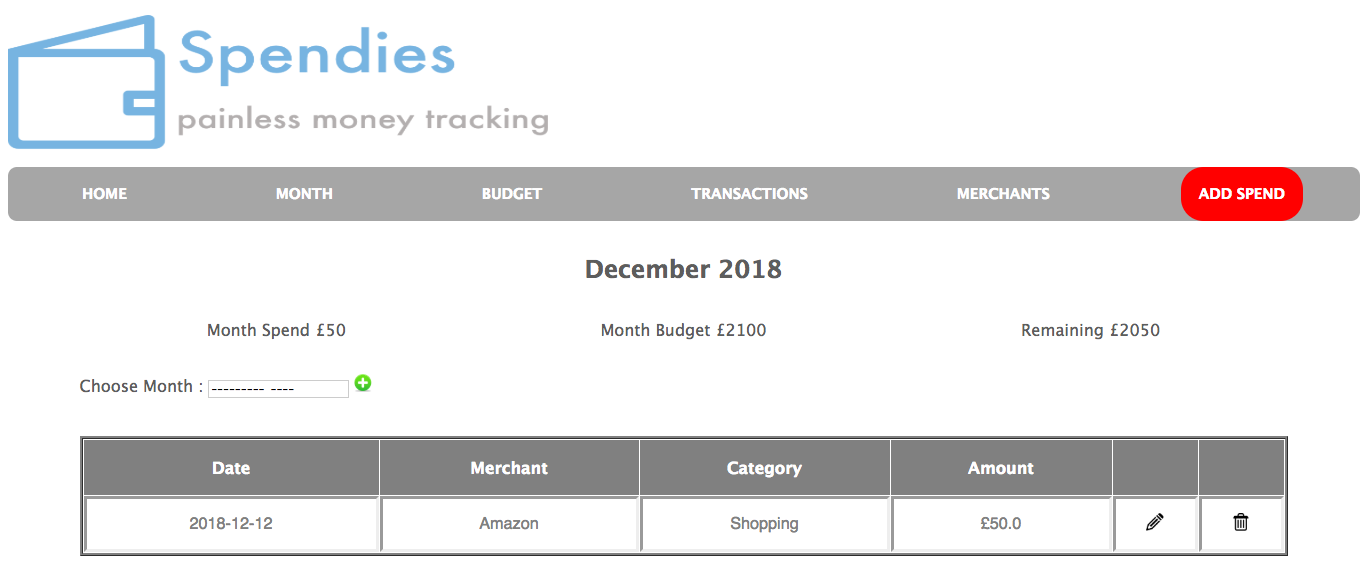
**Description here**

User inputs £300 as the budget limit for the category “Going out” (Screenshot 1). This is then saved and displayed in the total monthly budget table (screenshot 2) and is also used in the spend as a percentage of budget graph in Screenshot 3 (26% calculated as 78.32 / 300 where £78.32 is the total monthly spend for “going out” as shown in Screenshot4).).

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.14 | Show an interaction with data persistence. Take a screenshot of:  \* Data being inputted into your program  \* Confirmation of the data being saved | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

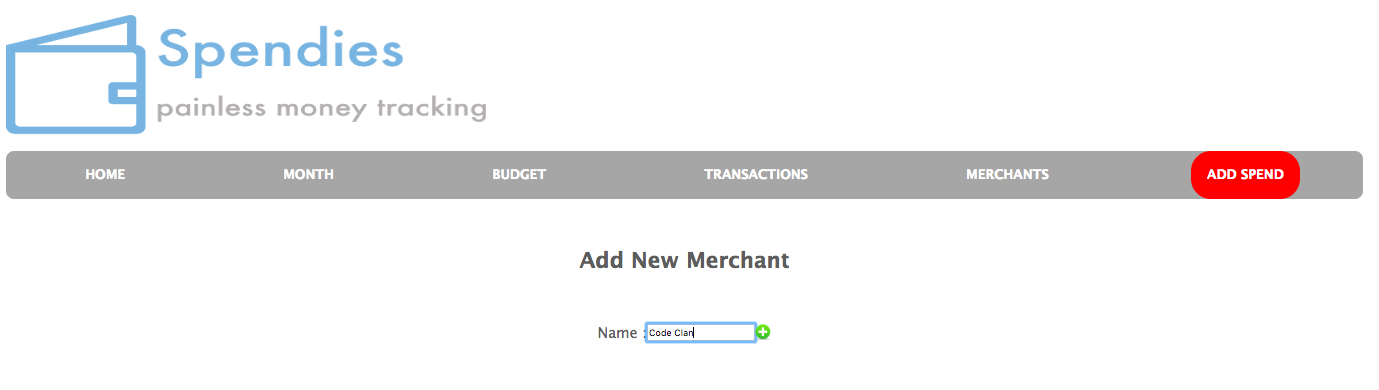
****

**Description here**

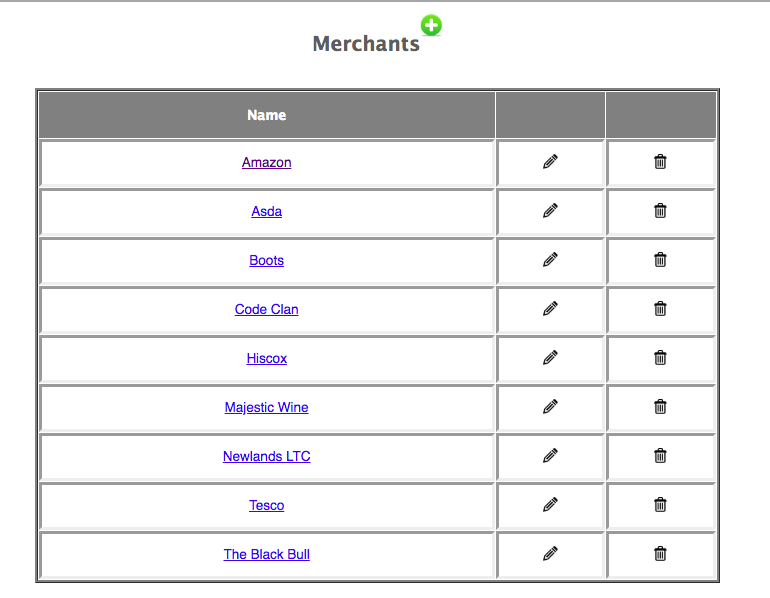
User enters a £50 transaction in screenshot 1. Screenshot 2 advises user that the transaction was successfully added. Screenshot 3 shows the list of the month’s transactions and the £50 transaction can be seen in the list (only transaction for month).

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.15 | Show the correct output of results and feedback to user. Take a screenshot of:  \* The user requesting information or an action to be performed  \* The user request being processed correctly and demonstrated in the program | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

****

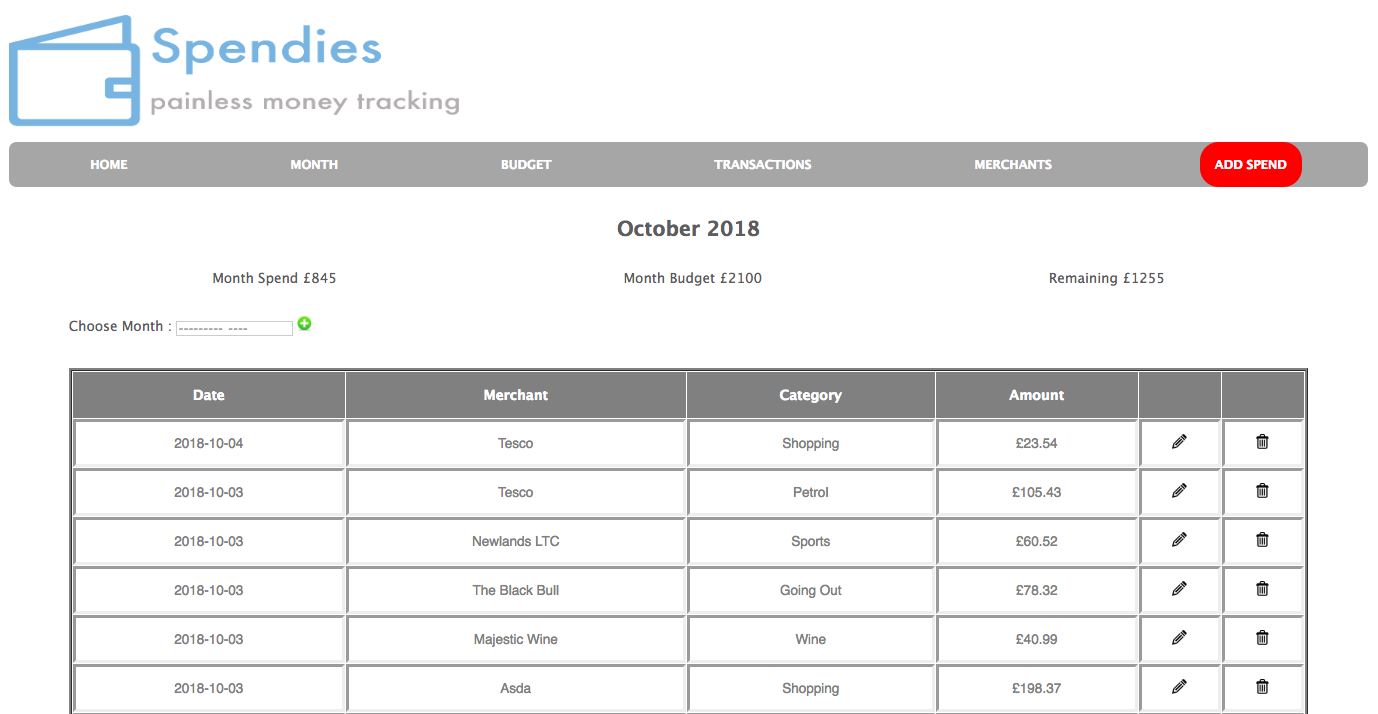
****

**Description here**

User requests that the merchant “Code Clan” is added to the merchant list (Screenshot1). This is confirmed in Screenshot2 and Screenshot3 shows the list of all merchants which now includes “Code Clan”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.11 | Take a screenshot of one of your projects where you have worked alone and attach the Github link. | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

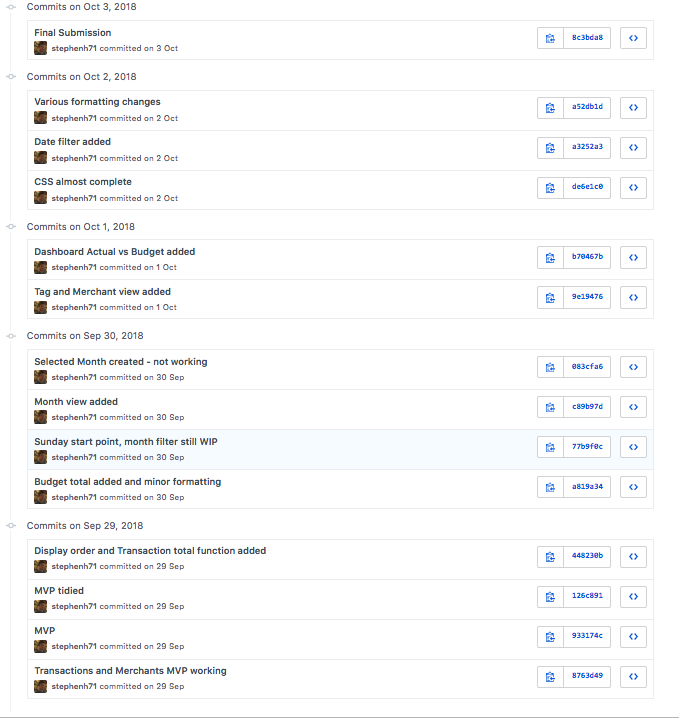
**Description here**

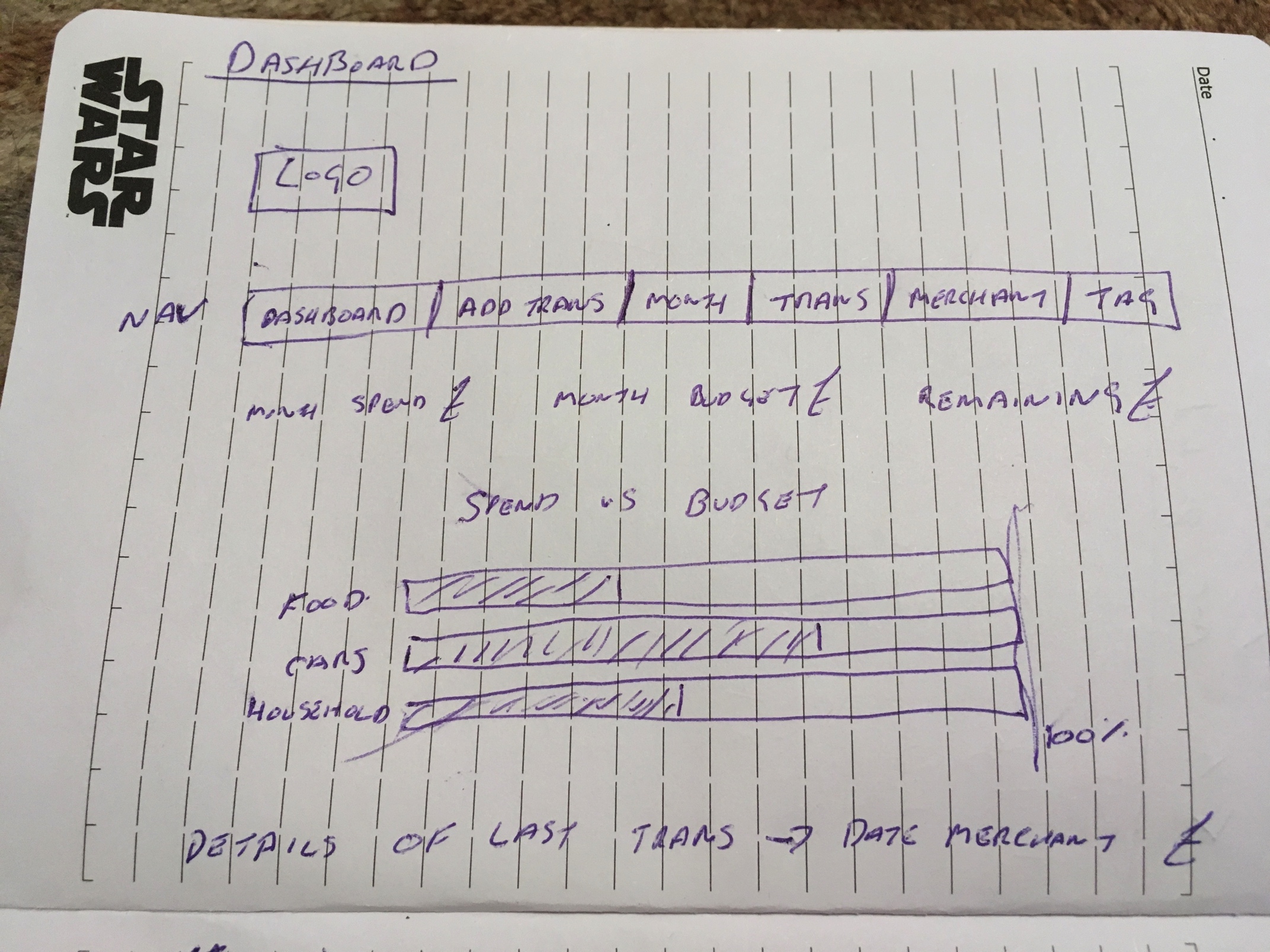
Spending Tracker project undertaken in week4 at the end of the Ruby module.

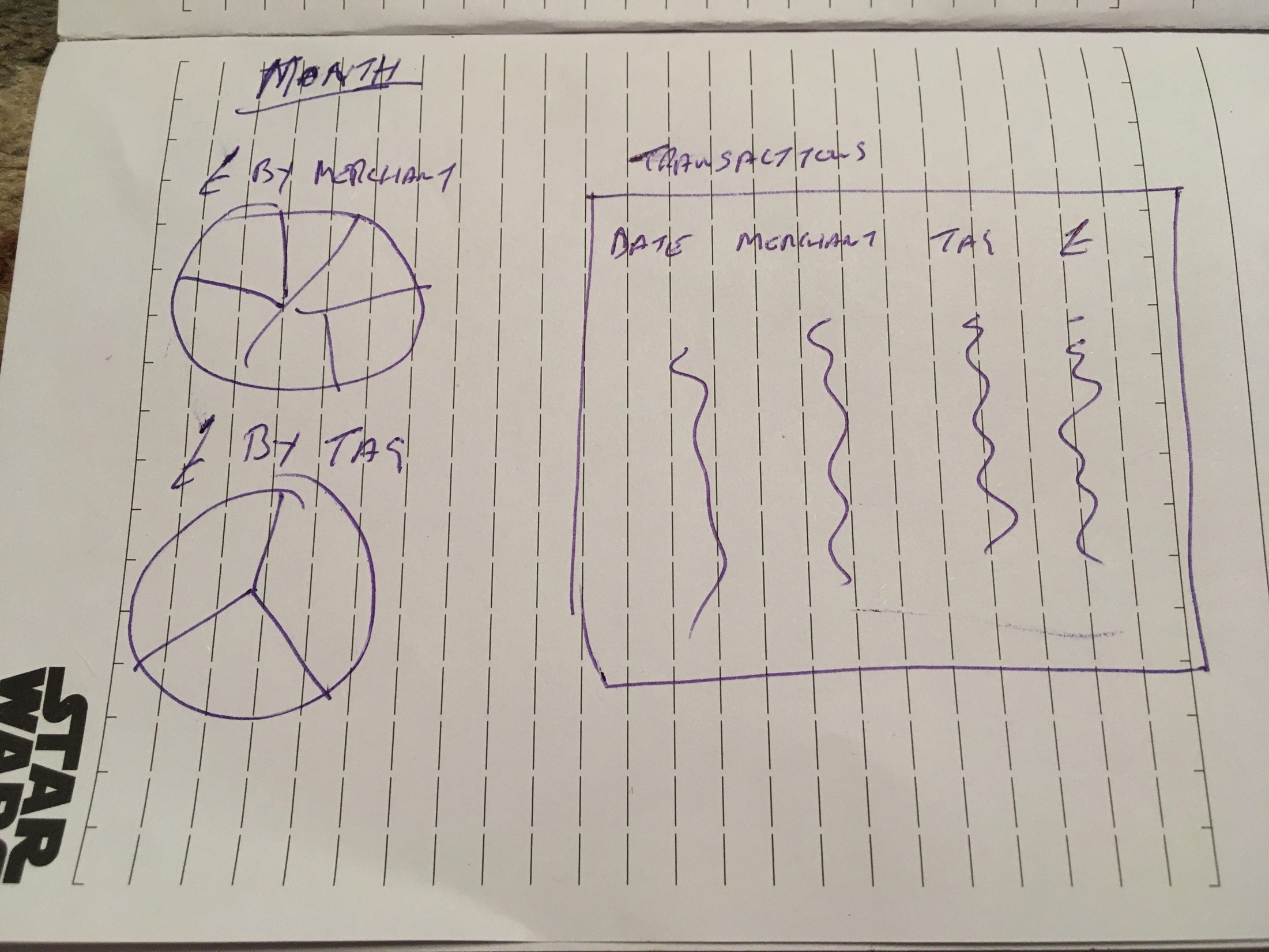
https://github.com/stephenh71/Week4\_Project\_v2

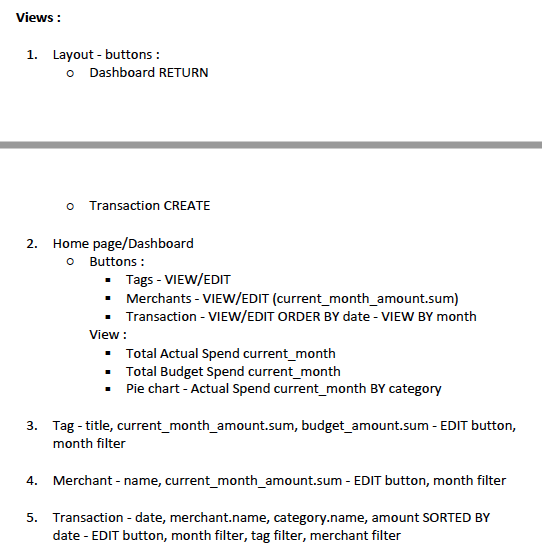
|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.12 | Take screenshots or photos of your planning and the different stages of development to show changes. | |
|  |  | **Description:** | |

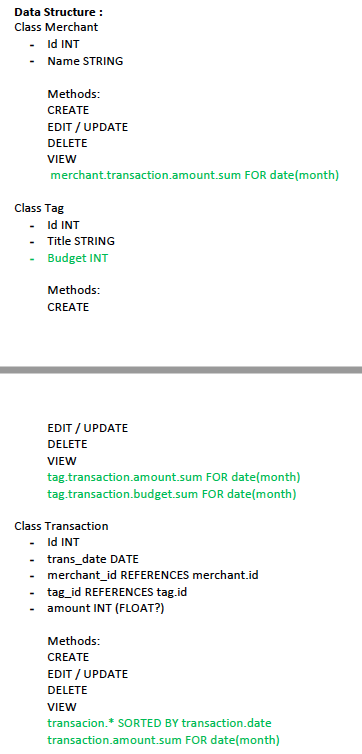
**Paste Screenshot here**

****

****

****

****

****

**Description here**

Screenshot 1 -Github commit history showing stages of development

Screenshots 2&3 – Early stage rough wireframe diagrams

Screenshot 4 – initial planning on views/site map

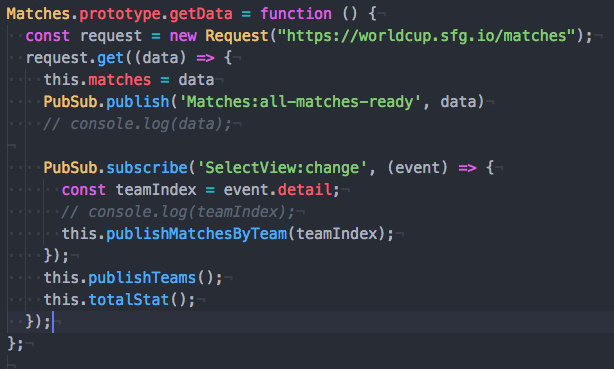
Screenshot 5 – early class diagram

Later stage planning docs already appended above.

**Week 7**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.16 | Show an API being used within your program. Take a screenshot of:  \* The code that uses or implements the API  \* The API being used by the program whilst running | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

****

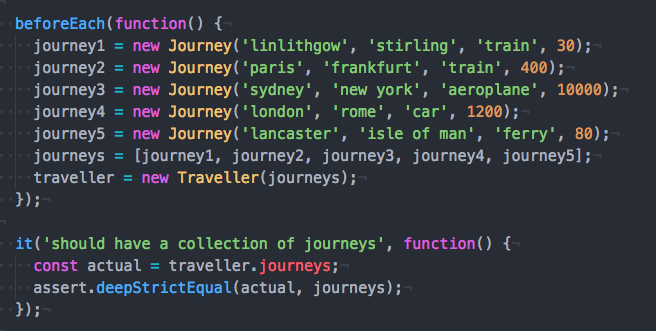
****

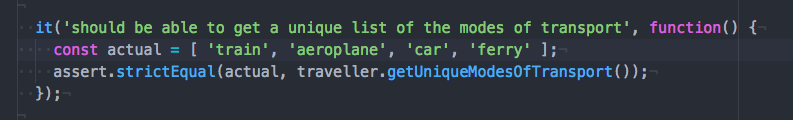
**Description here**

API requests details of all matches at the recent world cup from the FIFA website (Screenshot1). These are then used to populate a drop-down menu with the name of each participating team (Screenshot2) which when selected displays details of all their matches at the world cup and some statistics (Screenshot3).

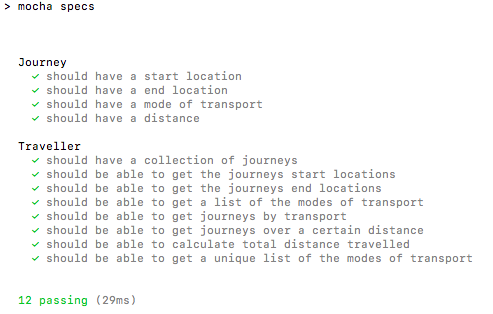
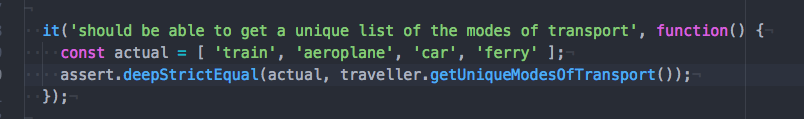
|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.18 | Demonstrate testing in your program. Take screenshots of:  \* Example of test code  \* The test code failing to pass  \* Example of the test code once errors have been corrected  \* The test code passing | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

****



****

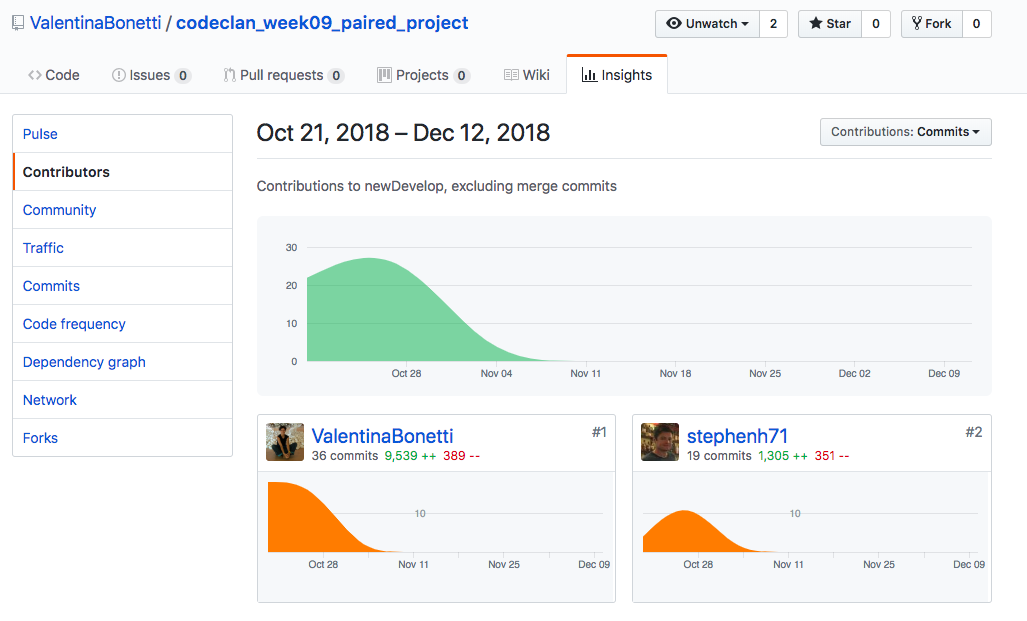
**Description here**

Screenshot 1 & 2 – test code example including test 12 which fails in Screenshot 3 because the test assertion was “strictEqual” whereas a “DeepStrictEqual” assertion was required. Screenshot 4 shows the test code after the correction and Screenshot 5 shows all tests passing after the correction.

**Week 9**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.1 | Take a screenshot of the contributor’s page on Github from your group project to show the team you worked with. | |
|  |  | **Description:** | |

**Paste Screenshot here**

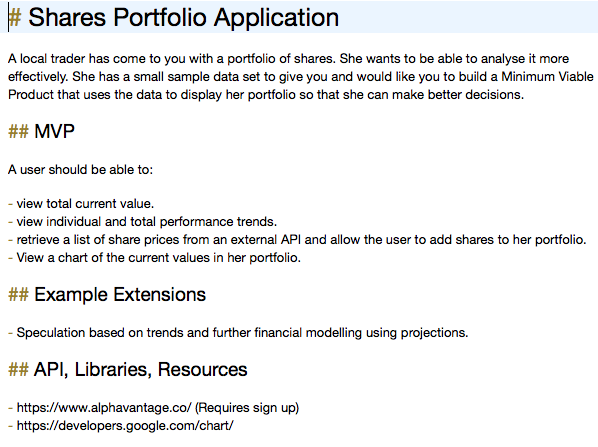
****

**Description here**

Worked with Valentina Bonetti on a share price/portfolio tracker - https://github.com/ValentinaBonetti/codeclan\_week09\_paired\_project

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.2 | Take a screenshot of the project brief from your group project. | |
|  |  | **Description:** | |

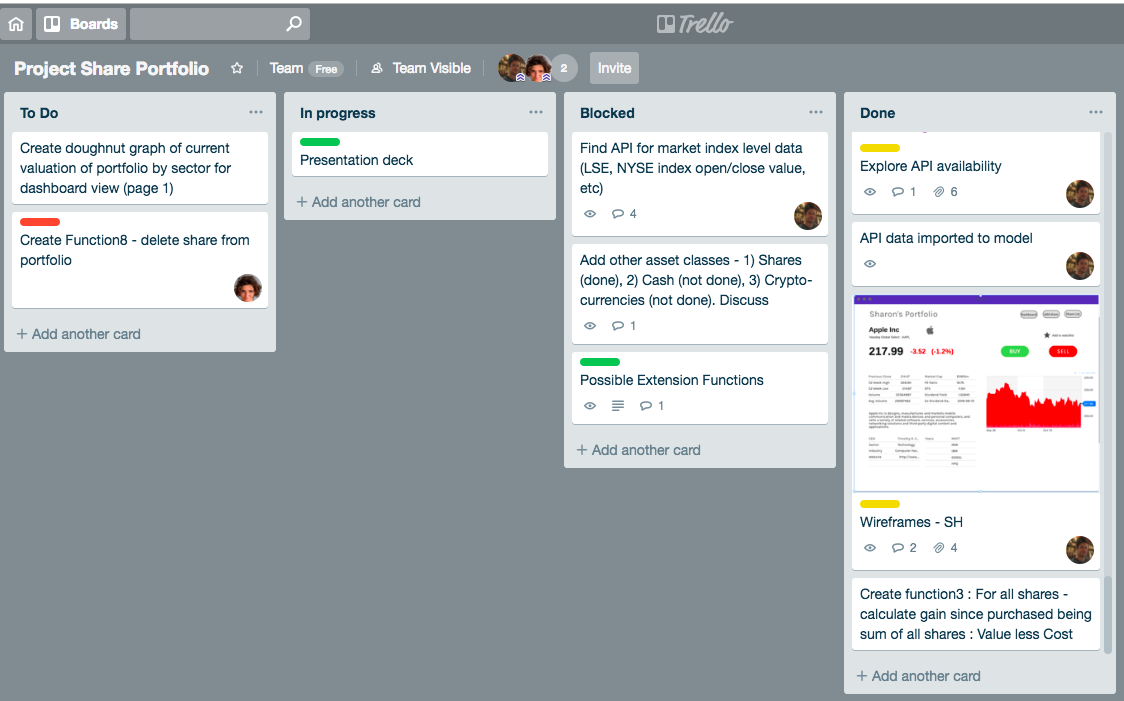
**Paste Screenshot here**

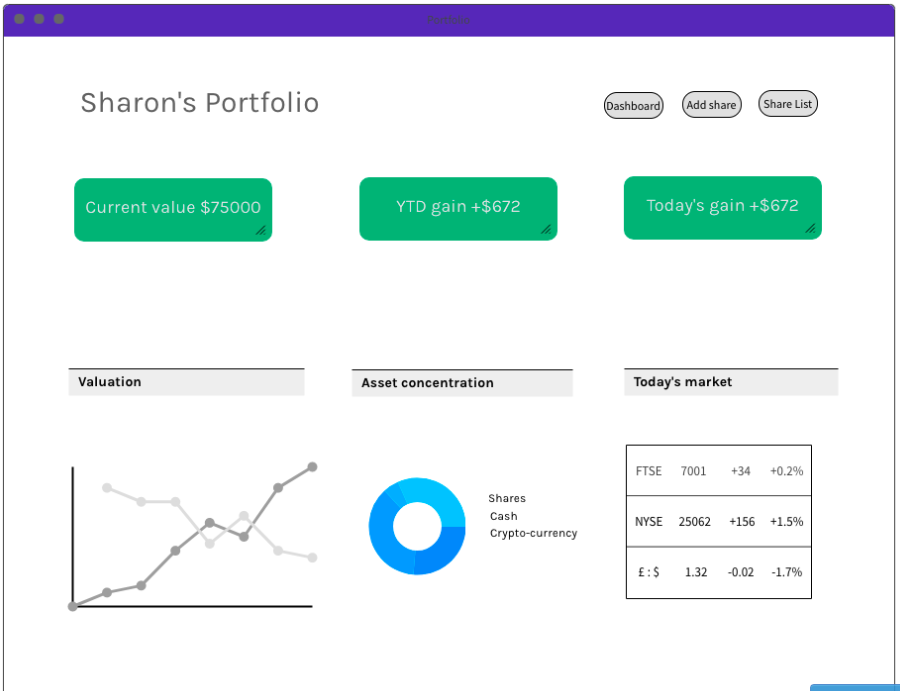
****

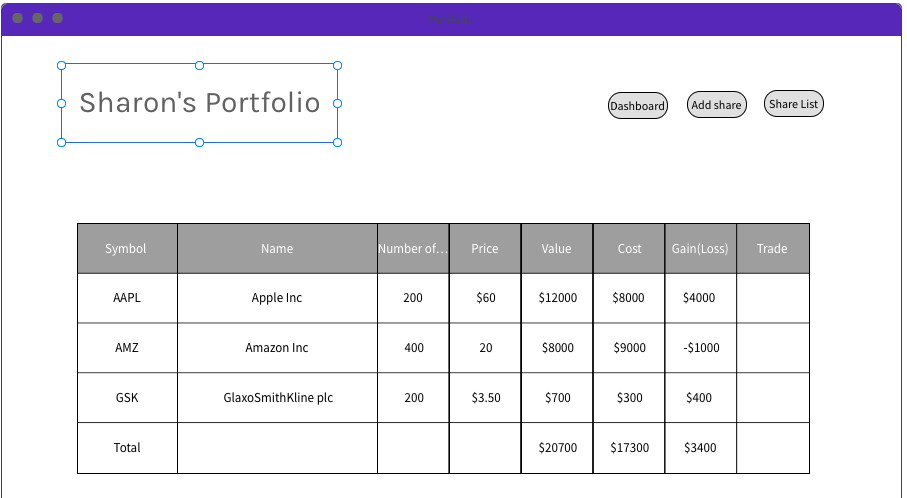
**Description here**

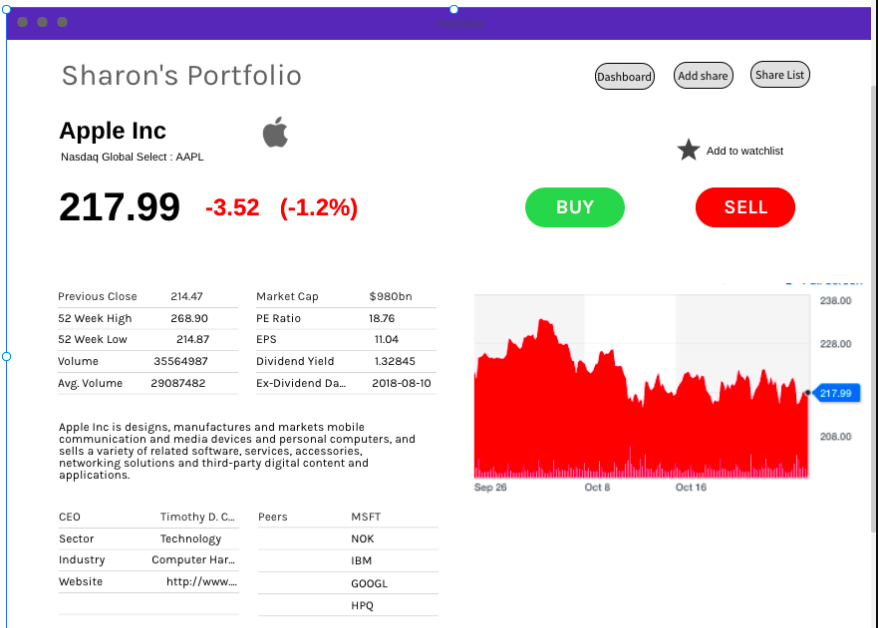
|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.3 | Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board. | |
|  |  | **Description:** | |

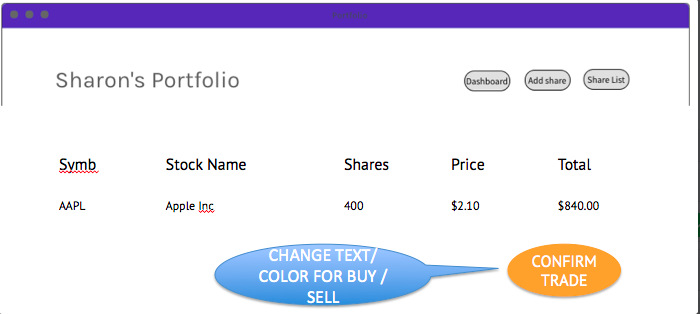
**Paste Screenshot here**

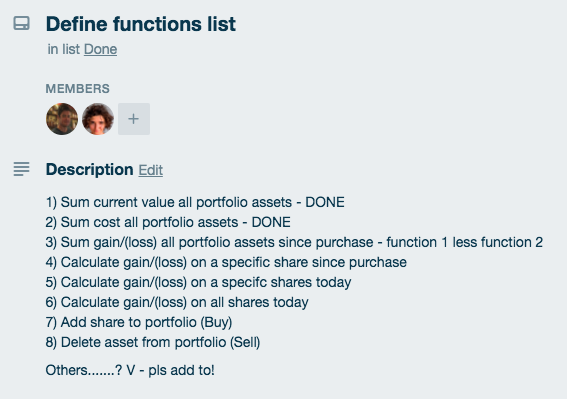
****

****

****

****

****

****

**Description here**

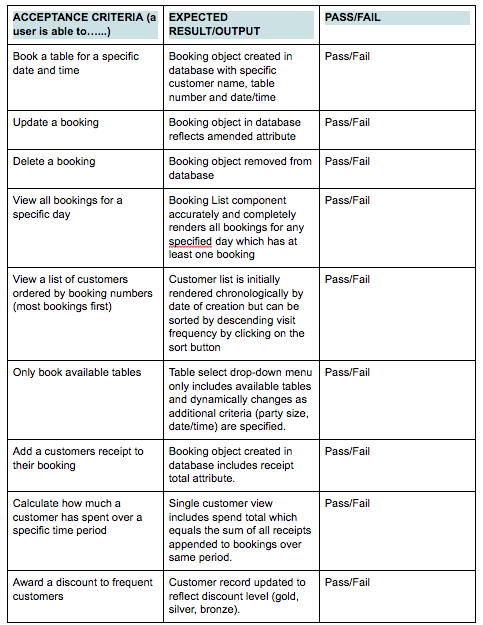
Screenshot1 – Trello Board

Screenshts2-5 – various wireframes

Screenshot 6 – functionality/features list

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.4 | Write an acceptance criteria and test plan. | |
|  |  |  | |

**Paste Screenshot here**

****

**Description here**

Acceptance criteria from group project no.2 (restaurant booking system).

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.7 | Produce two system interaction diagrams (sequence and/or collaboration diagrams). | |
|  |  | **Description:** | |

**Paste Screenshot here**

**Description here**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.8 | Produce two object diagrams. | |
|  |  | **Description:** | |

**Paste Screenshot here**

**Description here**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.17 | Produce a bug tracking report | |
|  |  | **Description:** | |

**Paste Screenshot here**

**Description here**

**Week 12**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **I&T** | I.T.7 | The use of Polymorphism in a program and what it is doing. | |
|  |  | **Description**: | |

**Paste Screenshot here**

**Description here**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **A&D** | A.D.5 | An Inheritance Diagram | |
|  |  | **Description:** | |

**Paste Screenshot here**

**Description here**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **I&T** | I.T.1 | The use of Encapsulation in a program and what it is doing. | |
|  |  | **Description:** | |

**Paste Screenshot here**

**Description here**

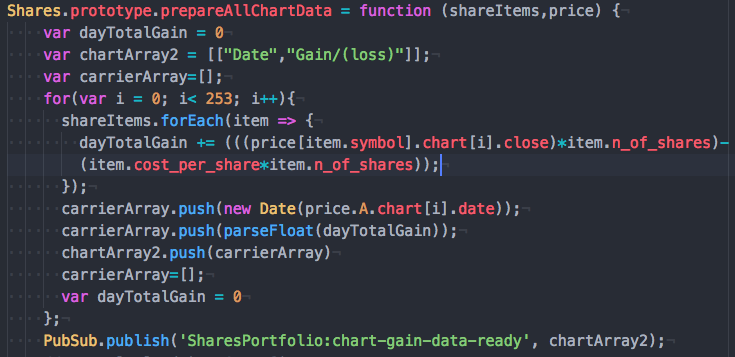
|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **I&T** | I.T.2 | Take a screenshot of the use of Inheritance in a program. Take screenshots of:  \*A Class  \*A Class that inherits from the previous class  \*An Object in the inherited class  \*A Method that uses the information inherited from another class. | |
|  |  | **Description:** | |

**Paste Screenshot here**

**Description here**

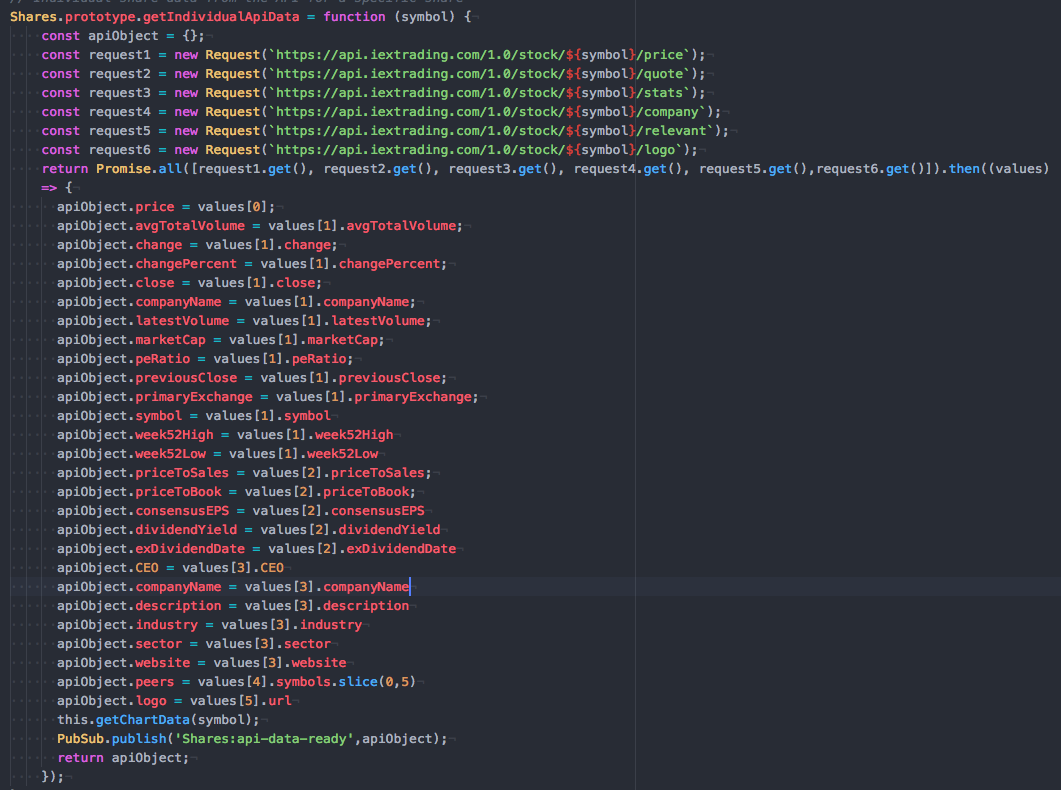
|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | **Ref** | **Evidence** |  |
| **P** | P.9 | Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms. | |
|  |  | **Description:** | |

**Paste Screenshot here**

****

**Description here**

Chosen this algorithm because I thought it was relatively complicated (for my standard) and involved a loop within a loop to take data originally sourced from a relatively unfriendly data structure (taken initially from an external API) to then format an extracted data set in the form of a new array for use in Google graphs.



Chosen this algorithm because it pulls on requests from multiple API’s to form data set which is then used in the rest of the app. We couldn’t find one API to provide all our data needs and this was the first time I’d used multiple API’s. It was also the first time I’d come across “asynchronicity”(?) and the need to use Promises to wait for the API data to be returned before proceeding with the rest of the code.