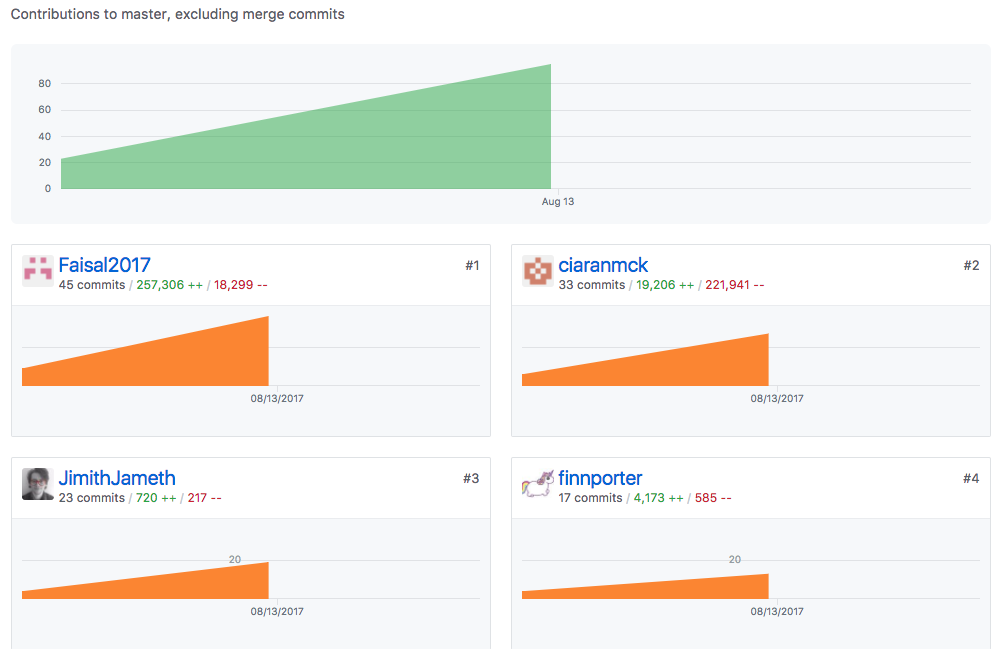
**Evidence for Project Unit**

Finn Porter

Cohort 13

23 June 2017

**P- 1 Github Contributors page**



**P- 2 Project Brief**

**Brief: Educational App**

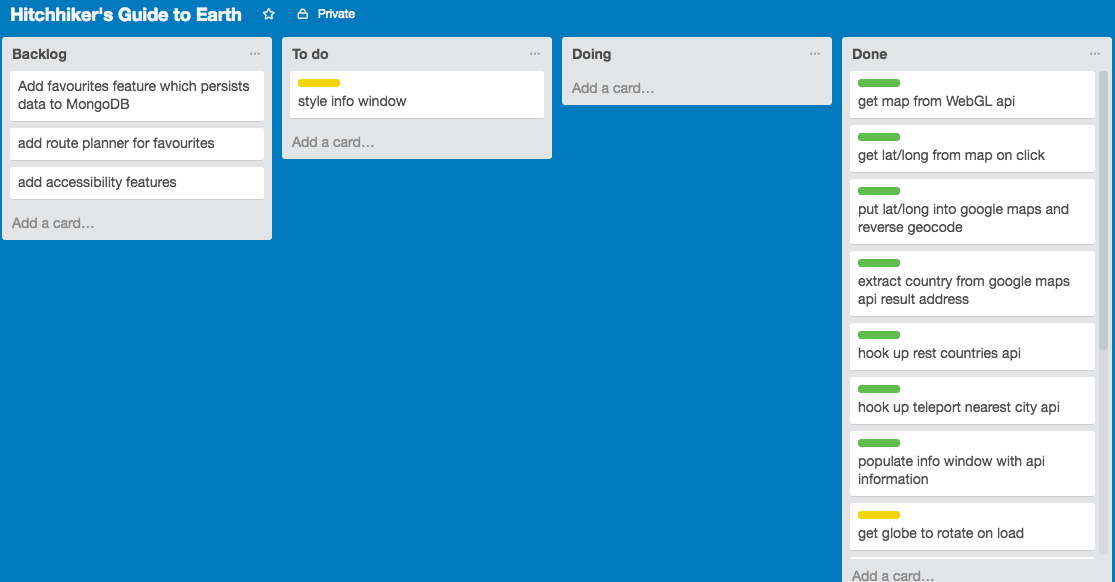
The BBC are looking to improve their online offering of educational content by developing some interactive apps that display information in a fun and interesting way.

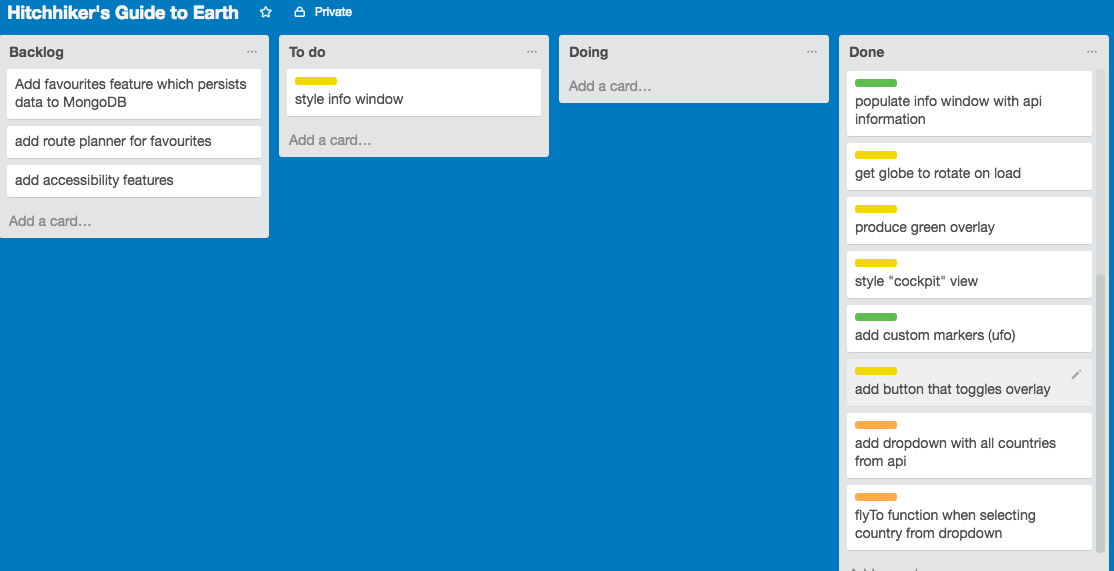
Your task is to make an MVP to put forward to them - this may only be for a small set of information, and may only showcase some of the features to be included in the final app. You might use an API to bring in content or a database to store facts. The topic of the app is your choice, but here are some suggestions you could look into:

Interactive timeline, e.g. of the history of computer programming Interactive map of a historical event - e.g. World War 1, the travels of Christopher Columbus MVP

* Display some information about a particular topic in an interesting way
* Have some user interactivity using event listeners, e.g to move through different sections of content

**P-3 Use of Trello**

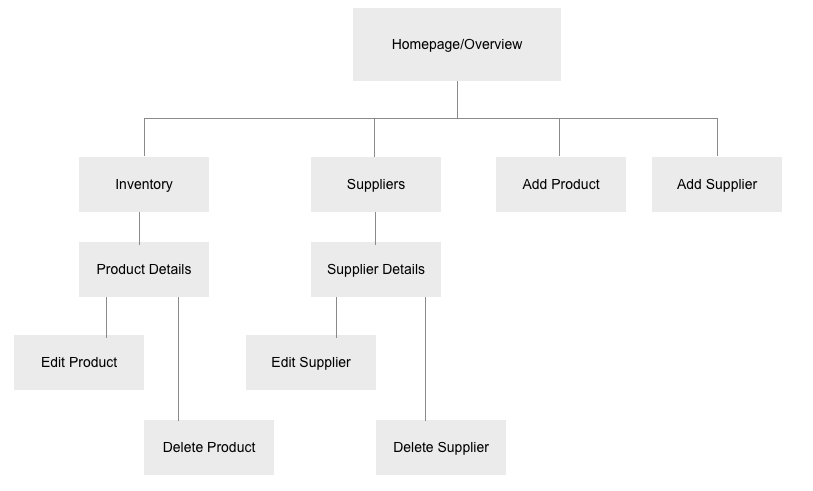




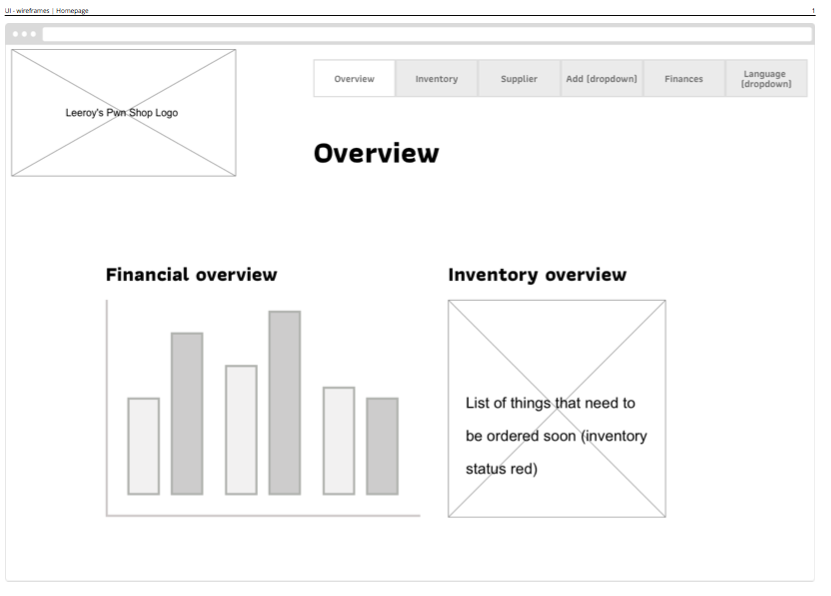
**P-4 Acceptance Criteria**

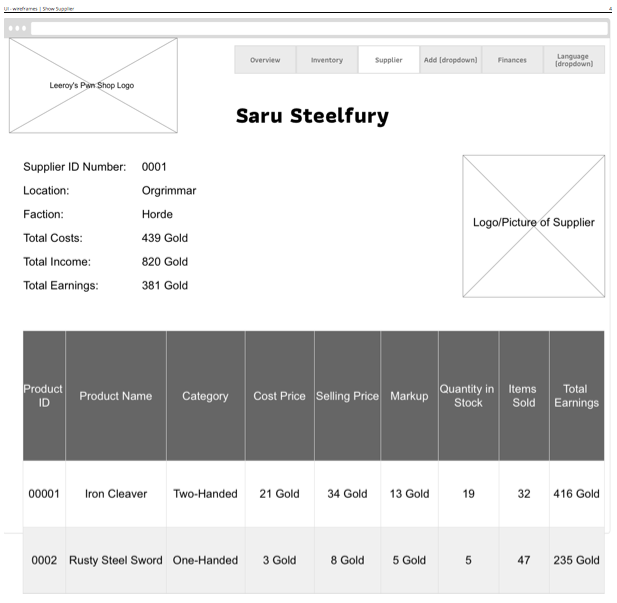
|  |  |  |
| --- | --- | --- |
| **Acceptance Criteria** | **Expected Result** | **Pass/Fail** |
| **User can mark spots on the globe** | **A marker appears on double click** | **pass** |
| **User can stop and start globe rotation** | **A button toggles earth rotation** | **pass** |
| **Api displays the data user requests** | **On click the marker produces information for the user** | **pass** |
| **User can zoom to a specific country** | **On selecting a country from a dropdown the globe automatically zooms in to show that country** | **pass** |
| **User can save favourite places** | **There is no option to save places yet** | **fail** |

**P-5 User sitemap**

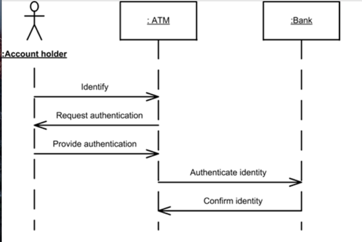
****

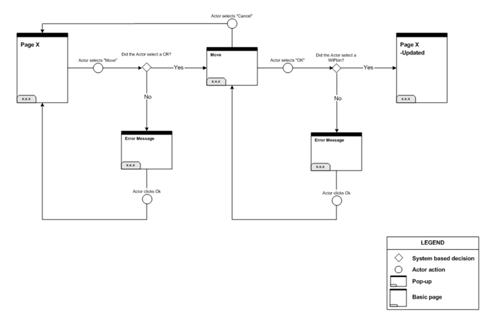
**P-6 Wireframes designs**

****

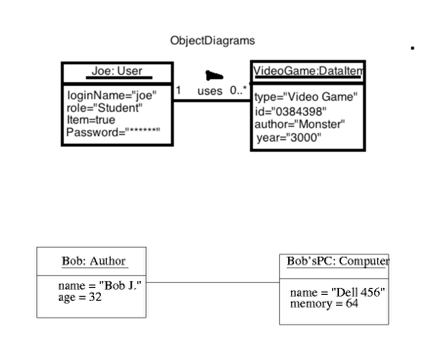
****

**P-7 System interactions diagrams**





**P-8 Two Object Diagrams**



**P- 9 D.T.- a Choice of two algorithms (find the algorithms on a program you might have written, show the code you have used. )**

**On this example please take a screenshot and write what it is doing and why u decided to use it.**

**A - Search Algorithm- For one of the projects I carried out I had to find items in a warehouse, by bays and rows. The best way to do this was to use a search algorithm, where the items had an ID. I had passed the ID into the function and iterated through the items checking the ID I was looking for.**

**B- Delete Algorithm - In the same project I had to delete items from the warehouse. The delete algorithm allowed me to go and find the item by ID and delete it from the array of items, in each bay.**

**P - 10 Example of Pseudocode**

**class TestInventory < MiniTest::Test**

**def #inventory has to have products**

**#for each product it has to have {**

**# name of product,**

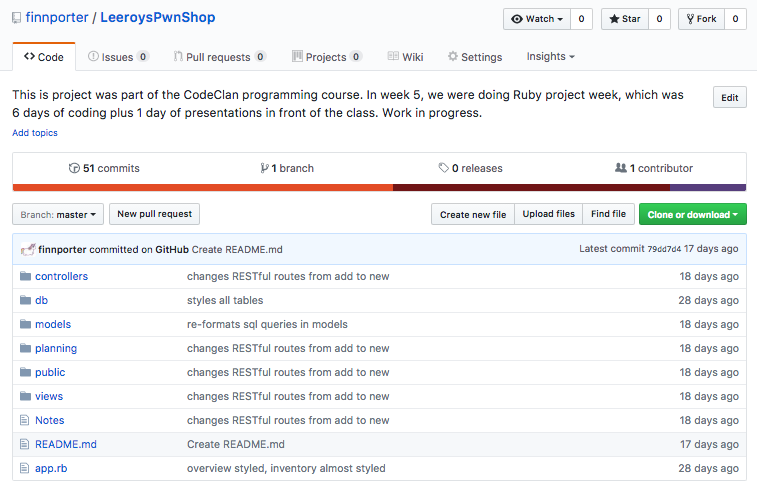
**# buy price,**

**# sell price,**

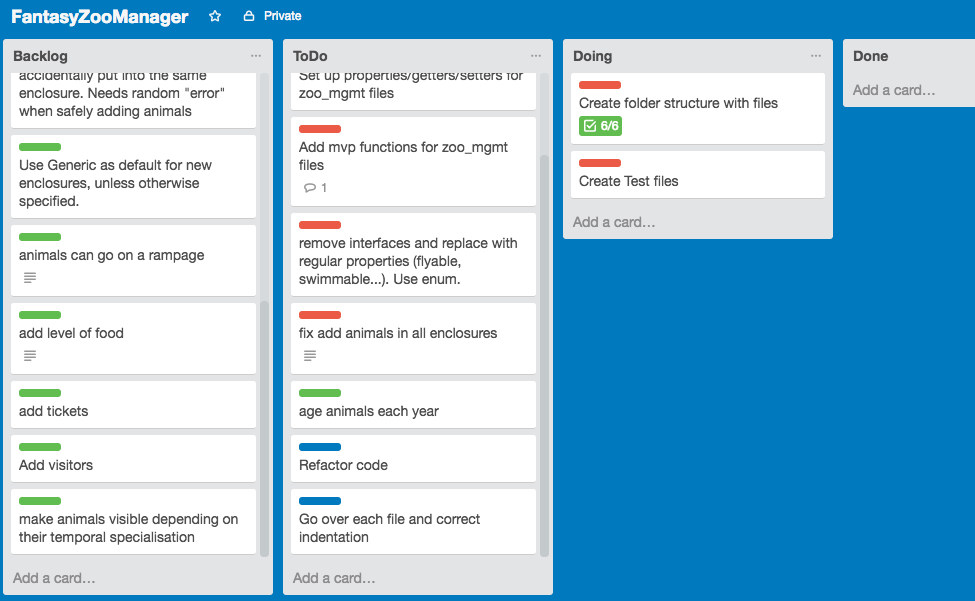
**# quantity on stock**

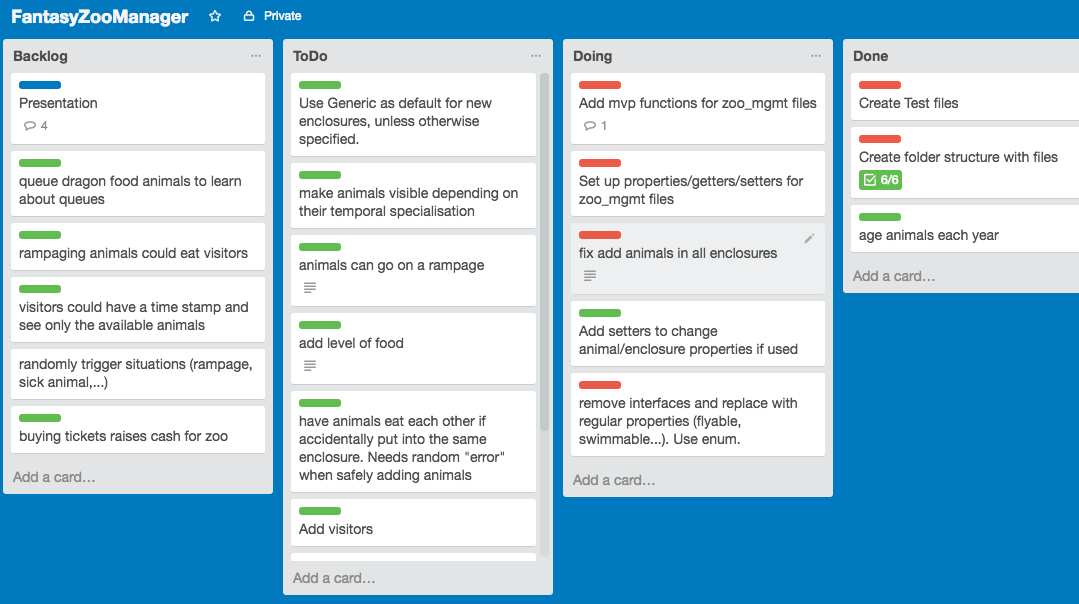
**}**

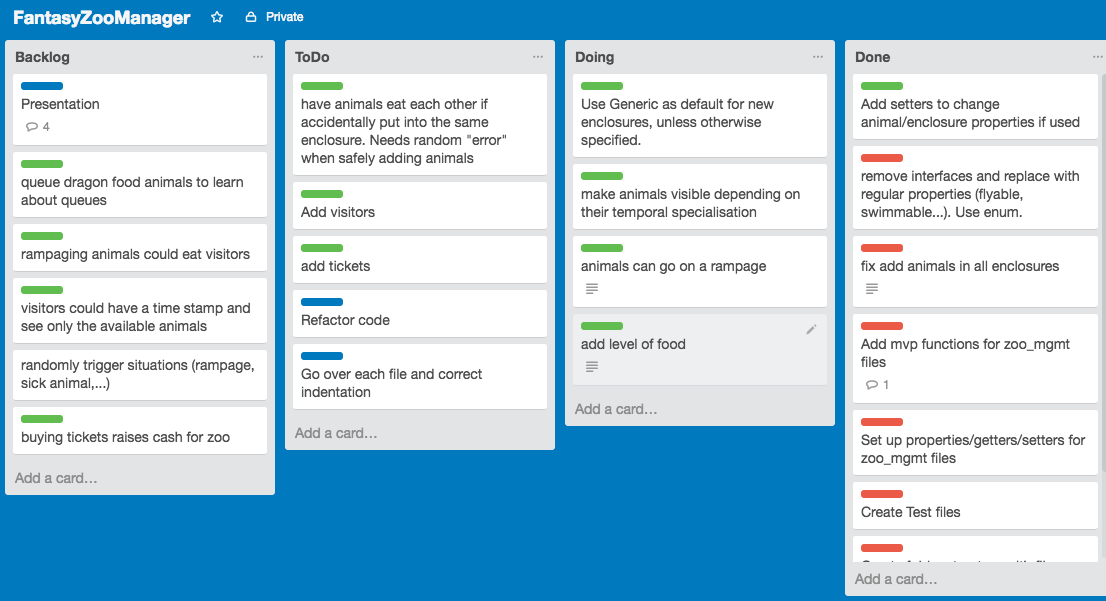
**P - 11 Github link to one of your projects**

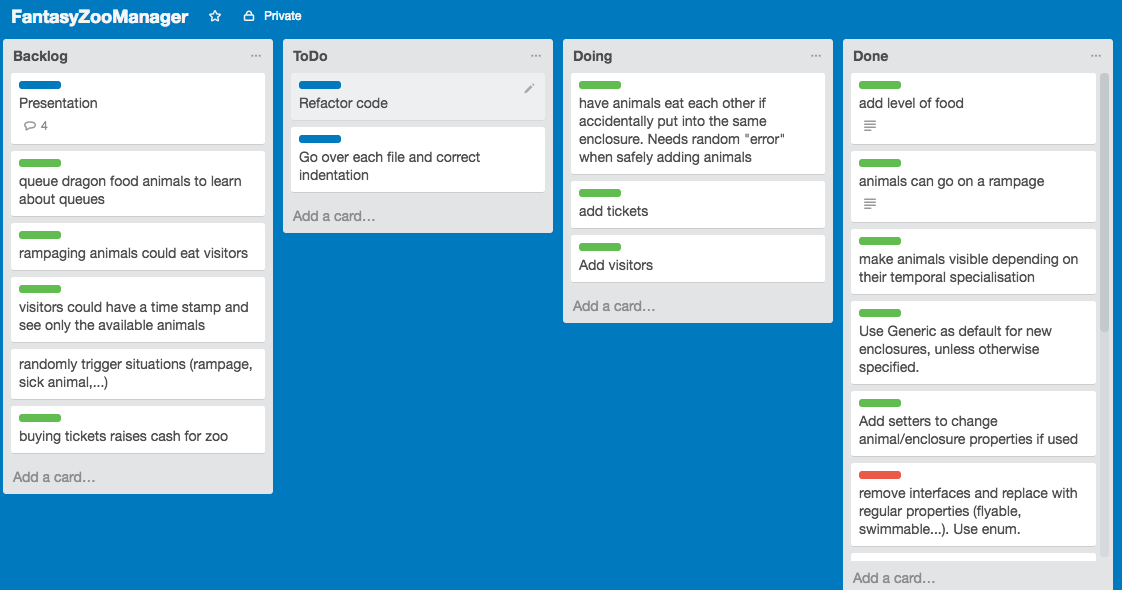


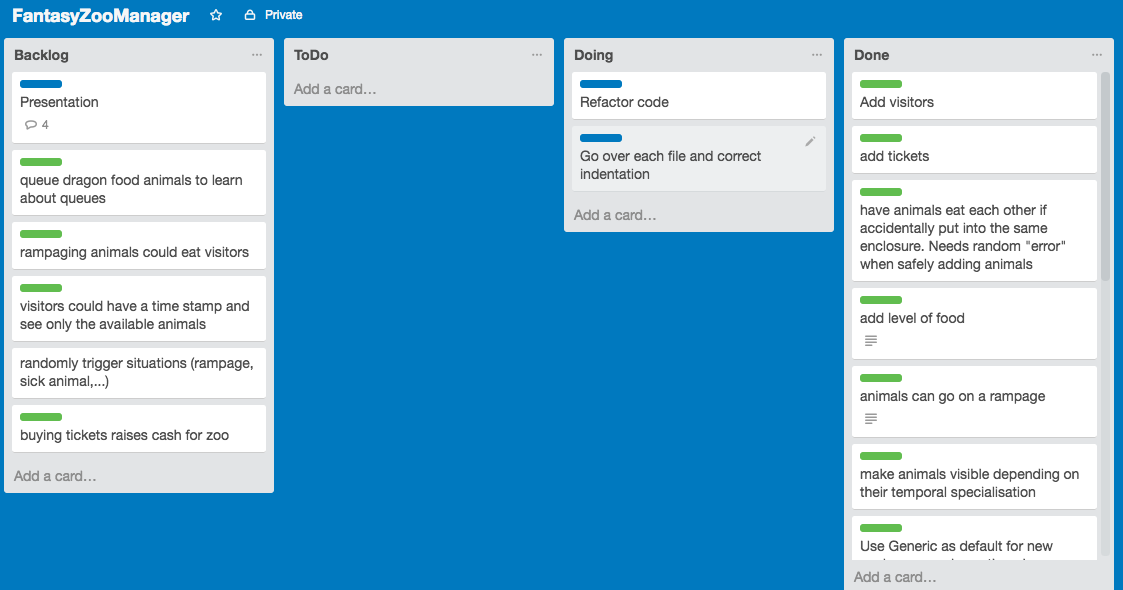
**P - 12 Screenshot of your planning and the different stages of development to show changes.**

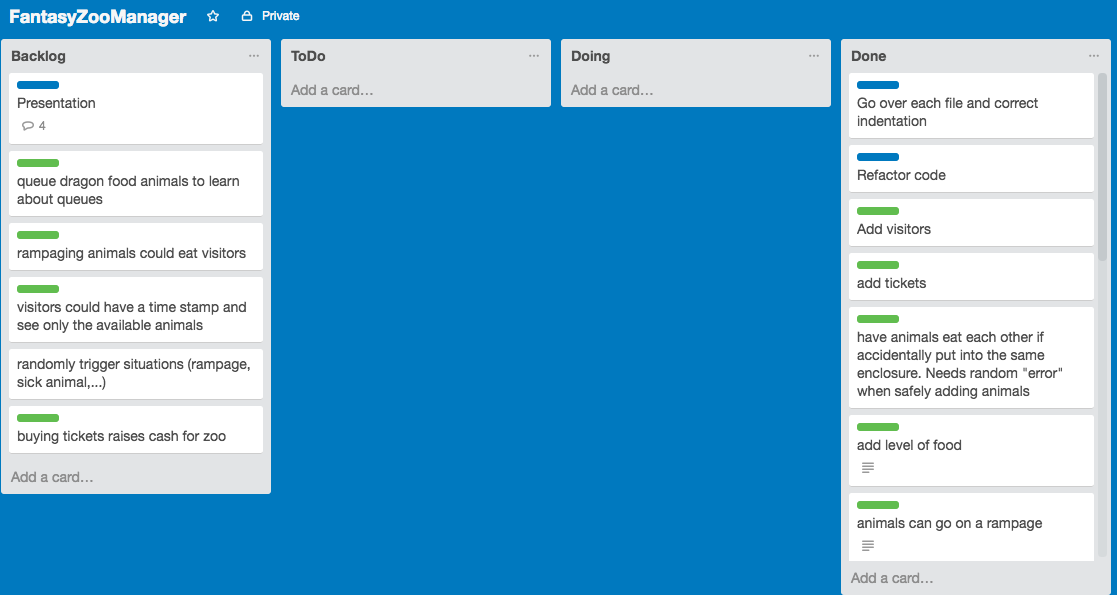
****

****

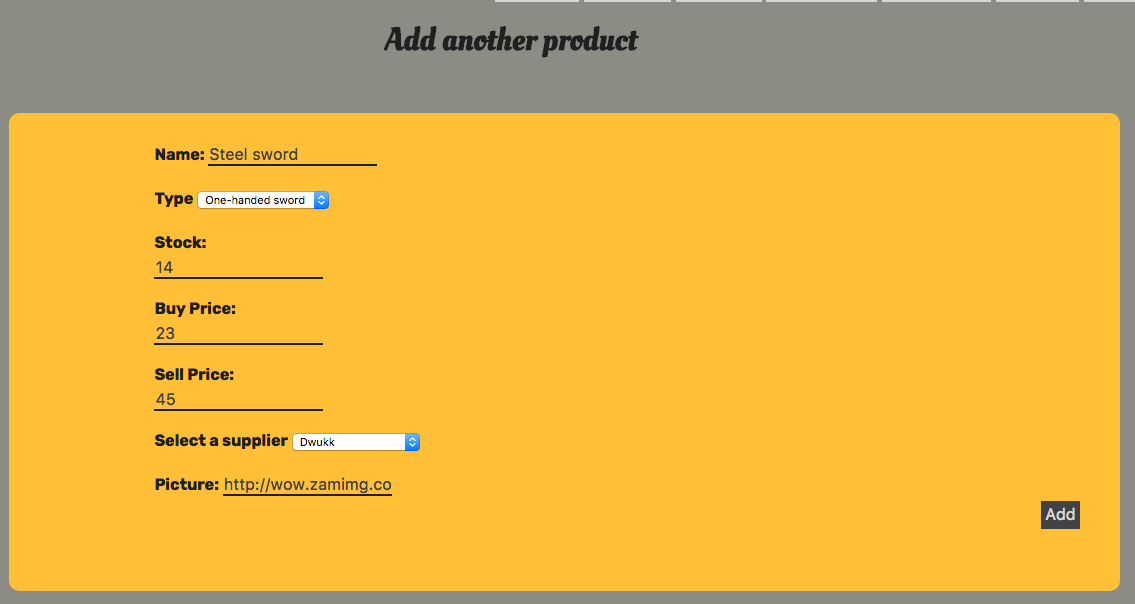
****

****

****

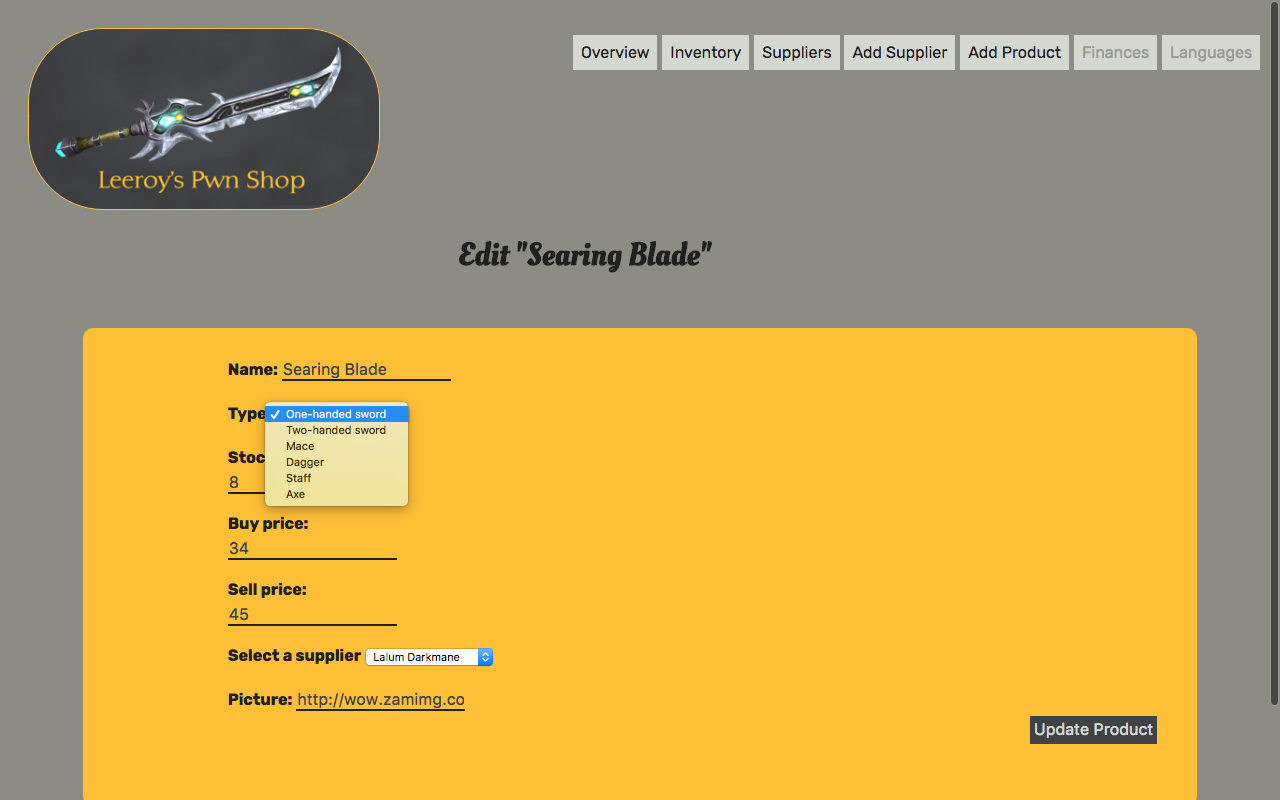
****

**P - 13 User input**

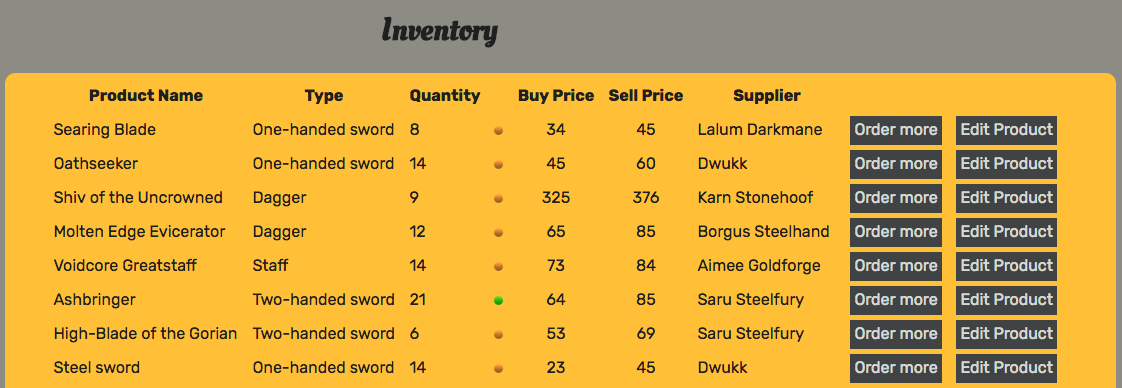
****

**Make sure u show the input being added.**

**P - 14 Interaction with data persistence**

****

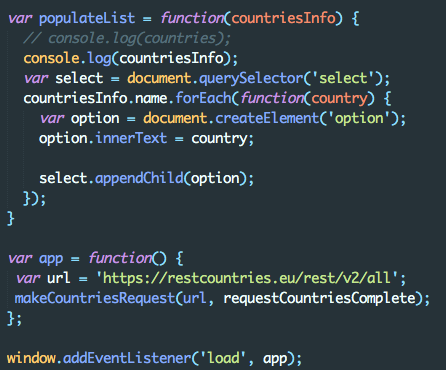
**P - 15 User output result**

****

Product “Steel Sword” from P-13 being saved in the inventory.

**P - 16 Show an API being used within your program.**

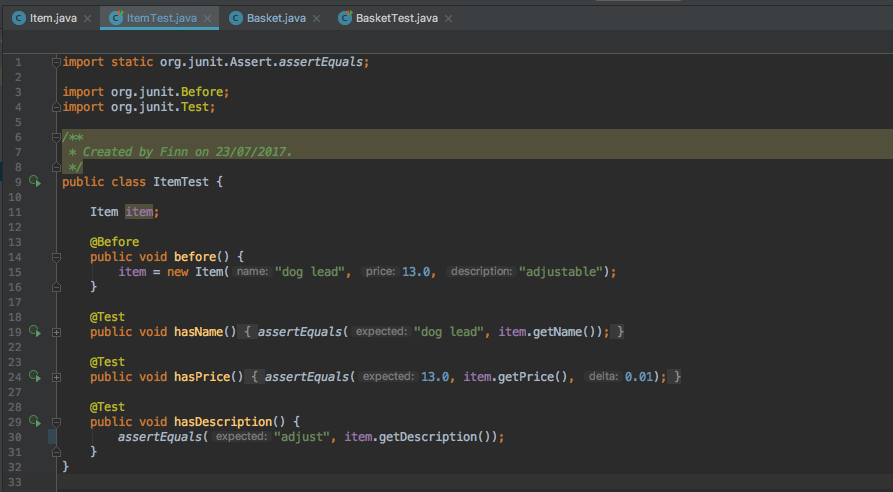
****

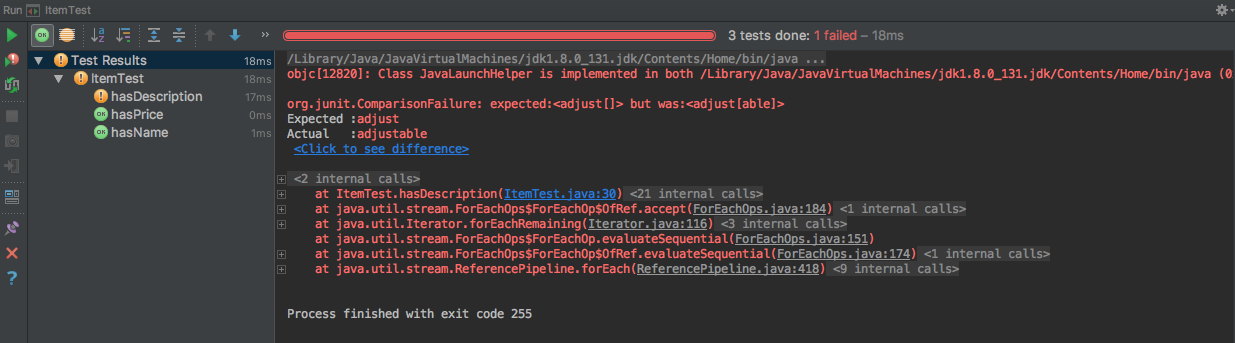
****

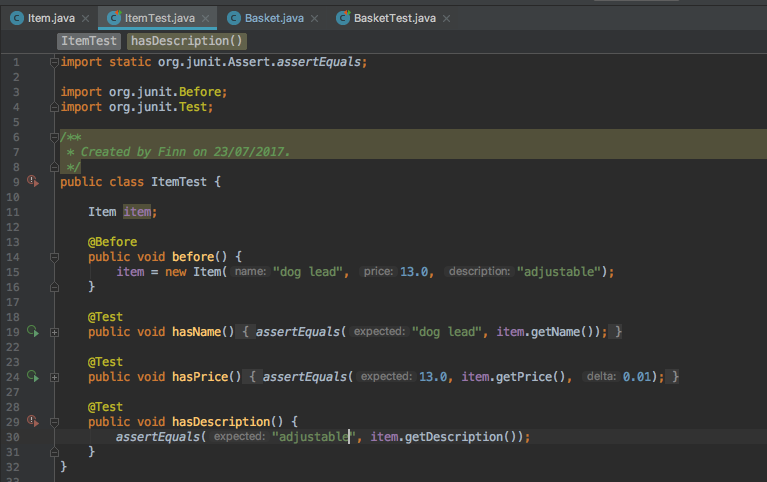
****

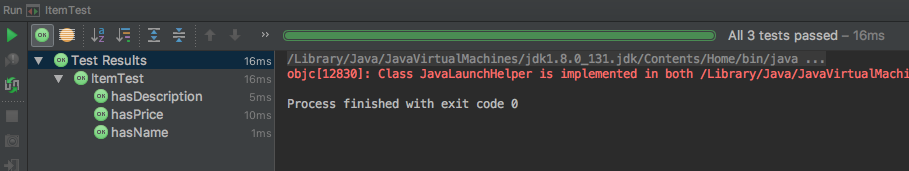
**P - 17 Bug tracking report showing the errors diagnosed and corrected.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **User must be able to see info when clicking marker** | **Failed** | **Info window populates with data from the api** | **Passed** |
| **The info is generated with each new marker** | **Failed** | **The population works and the nearest city info has been fixed to be be accurate** | **Passed** |
| **Trip date cannot be made for dates passed** | **Failed** | **Added validations to stop creation of trips with past dates** | **Passed** |
| **Trip can only have a number of available spaces** | **Failed** | **Set a number of spaces available per trip.** | **Passed** |

**P -18 Testing your program**

****

****

****

**P - 18 Acceptance test plan.**

