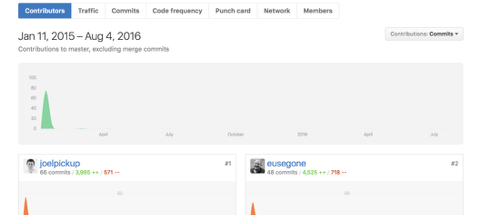
**Evidence for Project Unit**

Your name here

Your Cohort

Date here

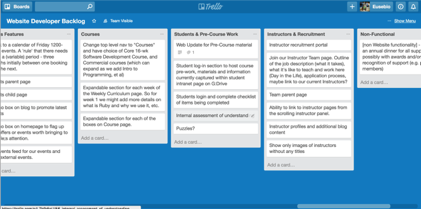
**P- 1 Github Contributors page**



**P- 2 Project Brief**

Write your project brief here for the group project, if you cannot remember ask one of the instructors or try to write one yourself based on the project you have created.

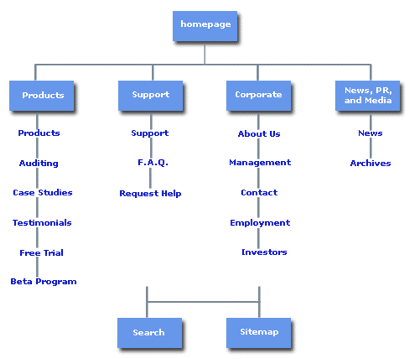
**P-3 Use of Trello**



**P-4 Acceptance Criteria**

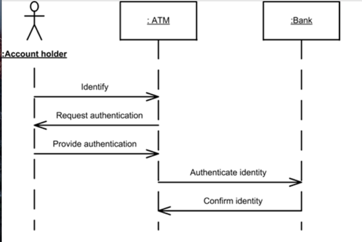
|  |  |  |
| --- | --- | --- |
| **Acceptance Criteria** | **Expected Result** | **Pass/Fail** |
| **User can click an area on the map and drop a ufo in** | **User is able to double click an leave a ufo marker for the country they have clicked.** | **pass** |
| **User can remove overlay by pressing button** | **User removes the overlay by pressing the button and can now select a country to place a ufo marker.** | **pass** |
| **Api displays the data user requests** | **The info box that appears for each marker shows us the data gathered to show** |  |
| **User can save favourite movie/movies** |  |  |
| **User can see list of favourite saved movies** |  |  |
|  |  |  |

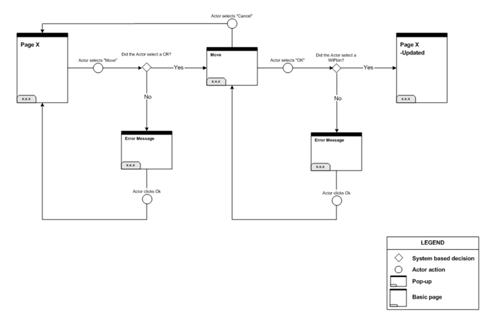
**P-5 User sitemap**

****

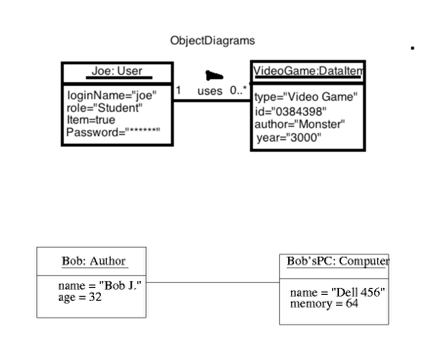
**P-6 Wireframes designs**

**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 TestBankAccount < MiniTest::Test**

**def #bank account has to have a name**

**# for each account it has to have a user = {**

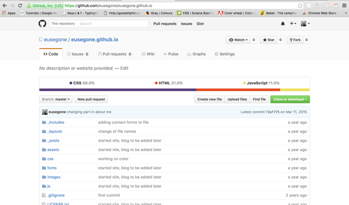
**# name of account holder: "Name",**

**# amount of cash in account: amount,**

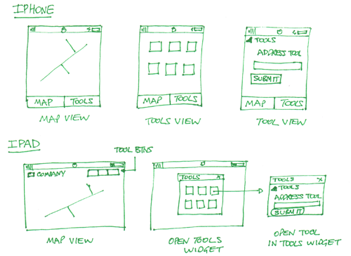
**# type of account: “personal” or "business"**

**}**

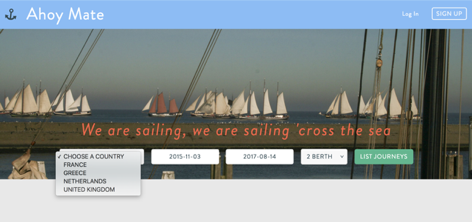
**P - 11 Github link to one of your projects**

****

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

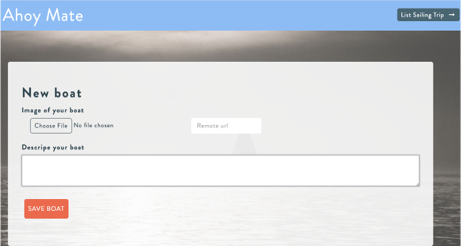
****

**P - 13 User input**

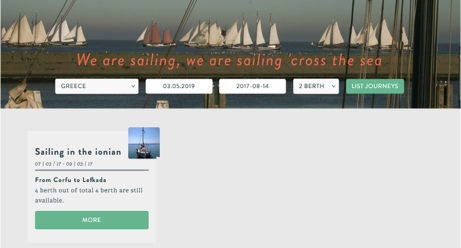
****

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

**P - 14 Interaction with data persistence**

****

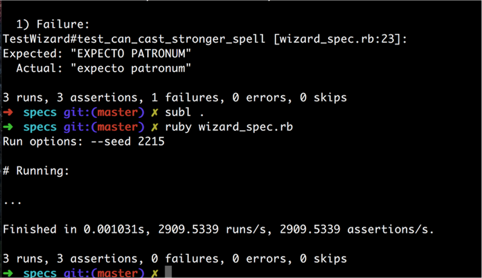
**P - 15 User output result**

****

**P - 16 Bug tracking report showing the errors diagnosed and corrected.**

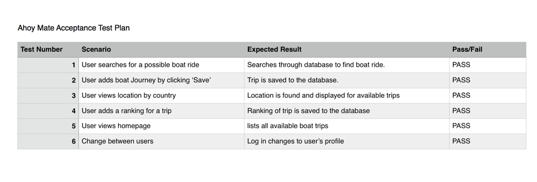
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **User must be able to add a trip** | **Failed** | **Saving a user, using the ID to assign a trip** | **Passed** |
| **Trip has a starting and end date** |  |  | **Passed** |
| **Trip date cannot be made for dates passed** | **Failed** | **Added 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 -17 Testing your program**



**Show the test not passing…..and then the test fixed.**

**P - 18 Acceptance test plan.**

****