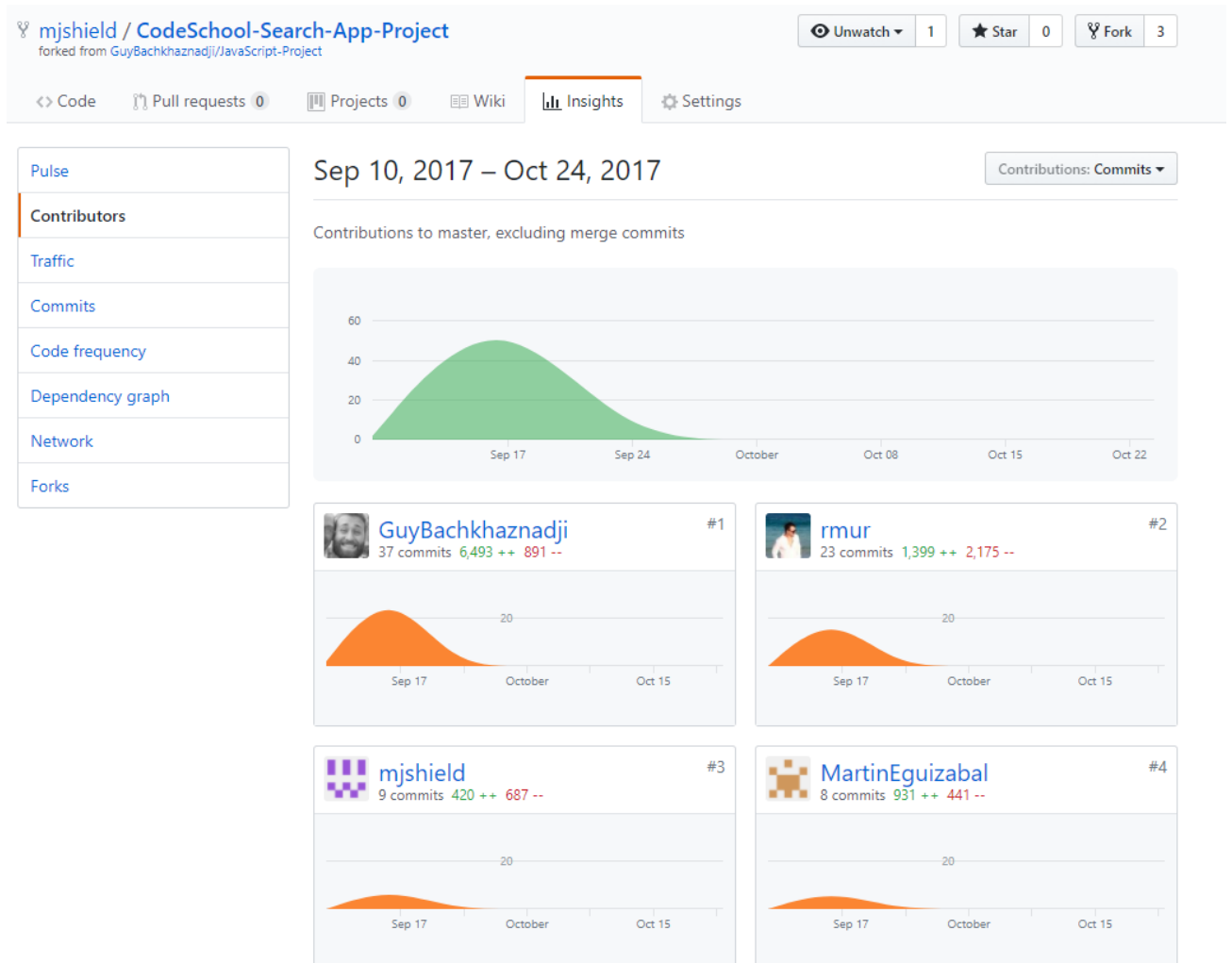


Project Unit Evidence

Matthew Shield

27th October 2017

P 1 – Take a screenshot of the contributor's page on Github from your group project to show the team you worked with:



P 2 – Take a screenshot of the project brief from your group project:

Group JavaScript Project - Operation Athena

A website which lists all the code boot camps in the UK you can attend and allows you to search through them by price, location, length or languages.

Technical Specs

A vanilla JavaScript project using Express, MongoDB, Google Maps API and a custom built API.

MVP

- Make custom API
- Display all bootcamps as list
- When clicked on show details of a bootcamp

Extensions

- Filter camps in list by Language
- Filter camps in list by Price
- Filter camps in list by Location
- Filter camps in list by Length
- Add google maps to the details page
- Favourites list which you can add to and remove from
- Compare two bootcamps
- Add cost of living for bootcamps to API

P 3 – Provide a screenshot of the planning your competed during your group project, e.g. Trello MOSCOW board:

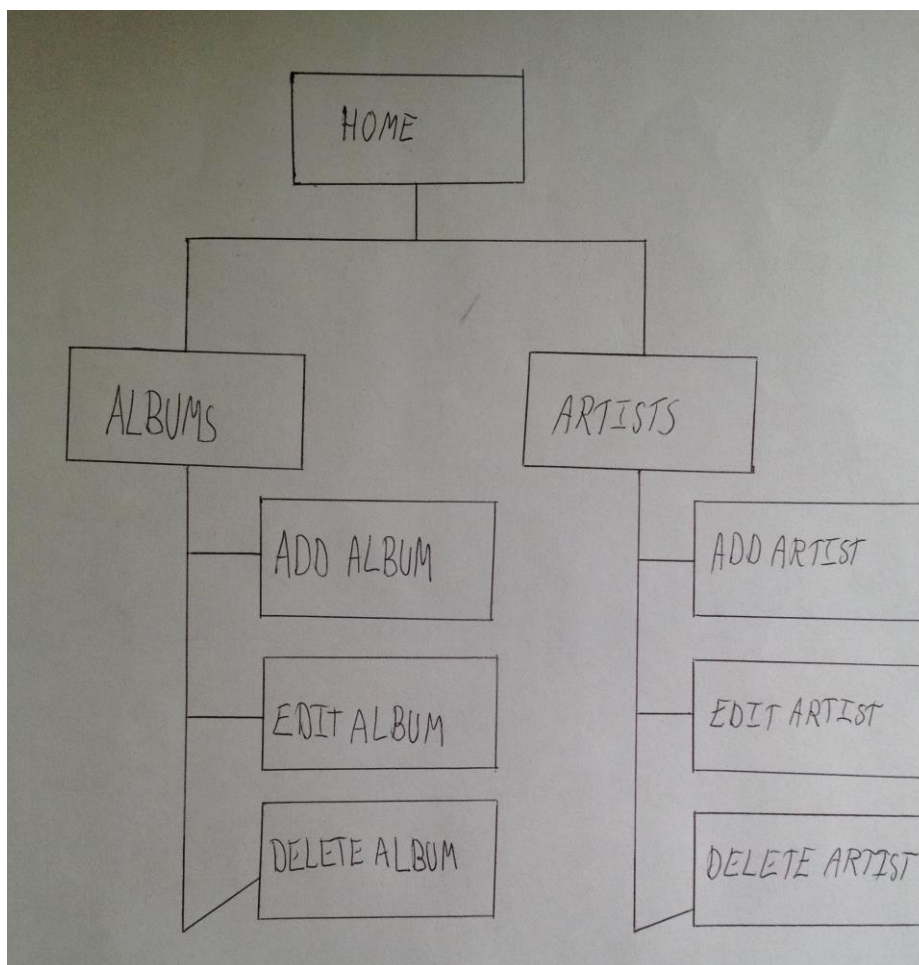
The screenshot shows a Trello board titled 'Operation Athena' with a 'Team Athena' label and 'Team Visible' permissions. The board is organized into four columns: 'Must', 'Should', 'Could', and 'Would'. Each column contains a list of tasks, each with a green progress bar indicating completion status. The 'Must' column has 10 tasks, 'Should' has 7, 'Could' has 5, and 'Would' has 3. Each column ends with an 'Add a card...' button.

Must	Should	Could	Would
Make API	Funding options	Favourites list	Rating of teachers?
Make File Structure	Filter bar	Add to favourites	Show ratings (if available)
Set up Git	Filter dropdowns	Remove from favourites	All details table
Show all bootcamps	Employment stats	Compare two bootcamps	Add a card...
Show details	Entry requirements (if available)	Make bootcamp logos fit/edit database	
Show cost	Add google map	Add a card...	
Show languages	Show cost of living		
Show address	Add a card...		
Show qualifications			
Add a card...			

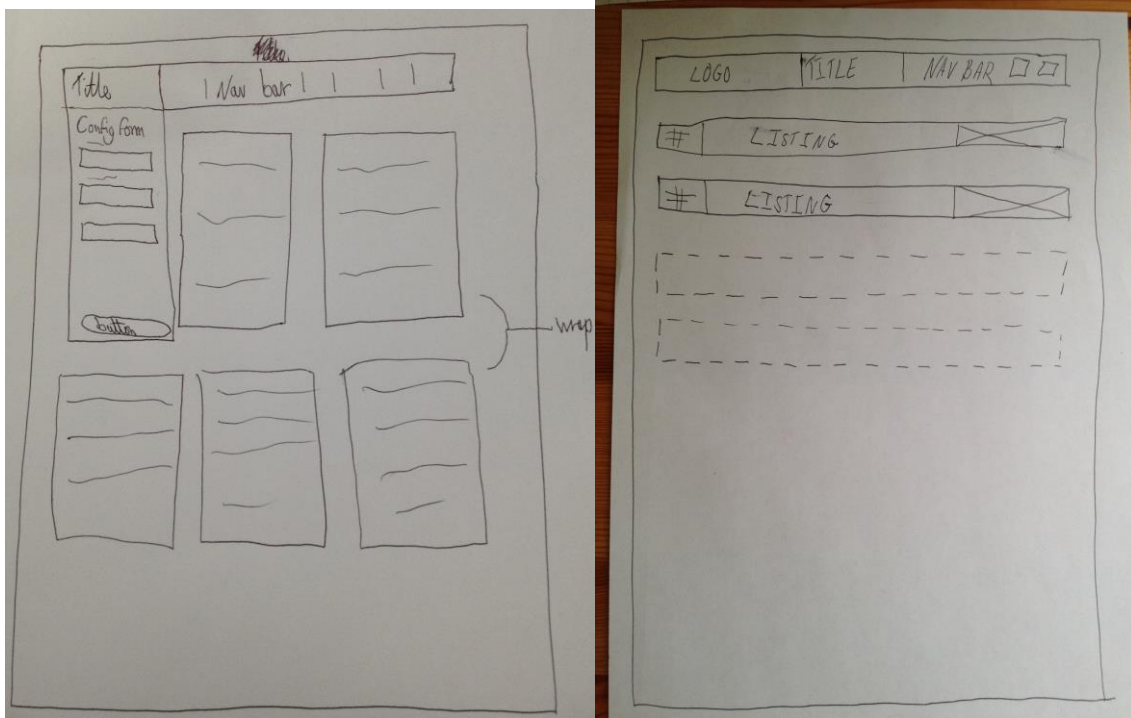
P 4 – Write an acceptance criteria and test plan:

Acceptance Criteria	Expectation/Result	Pass/Fail
A user can access the index of all albums.	A user clicks on the Albums button and the view is populated by all albums.	Pass
A user can access the details of a specific album.	A user clicks on the Details button and the appropriate album populates the view.	Pass
A user can return to the index of all albums from the details page.	A user clicks on the Albums button and the details view is replaced by a list of all albums.	Pass
A user can access the home page that lists nothing.	A user clicks the website title and is taken back to the original home view.	Fail

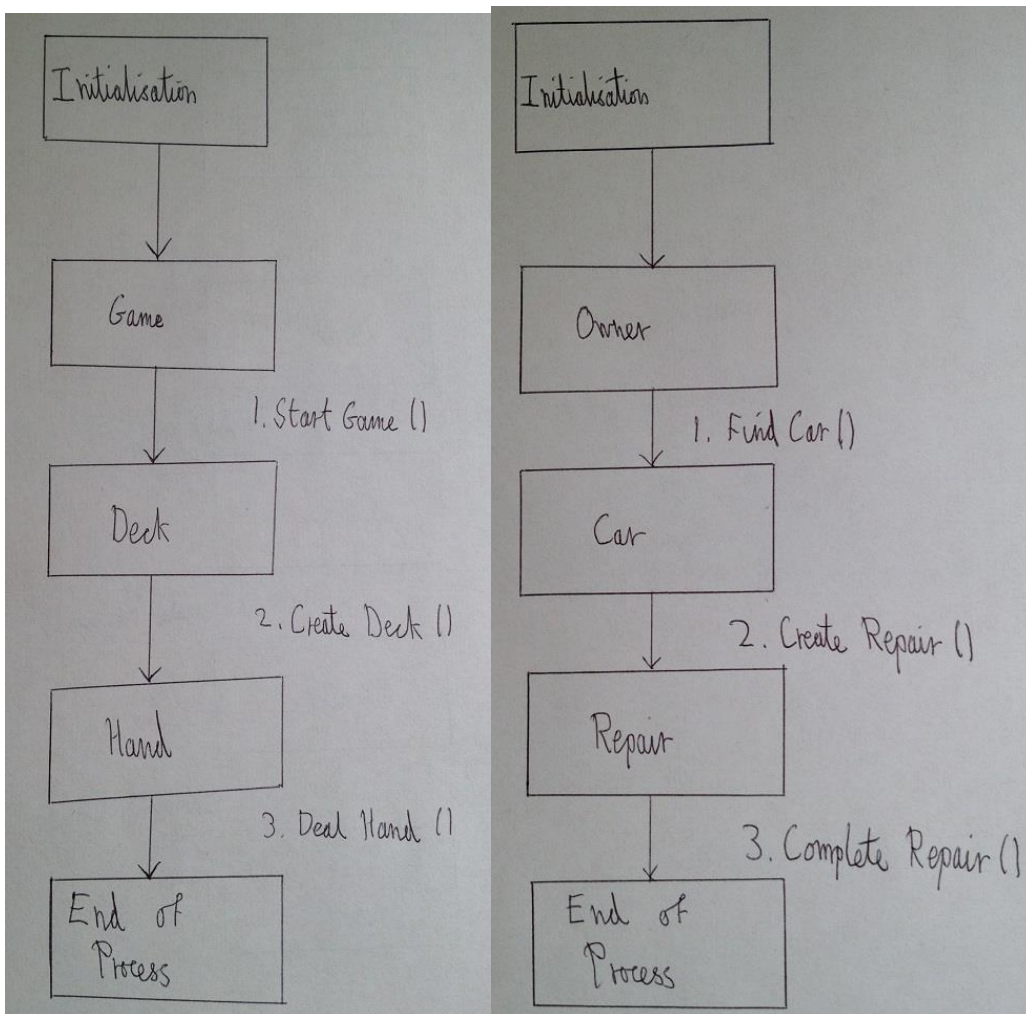
P 5 – Create a user sitemap:



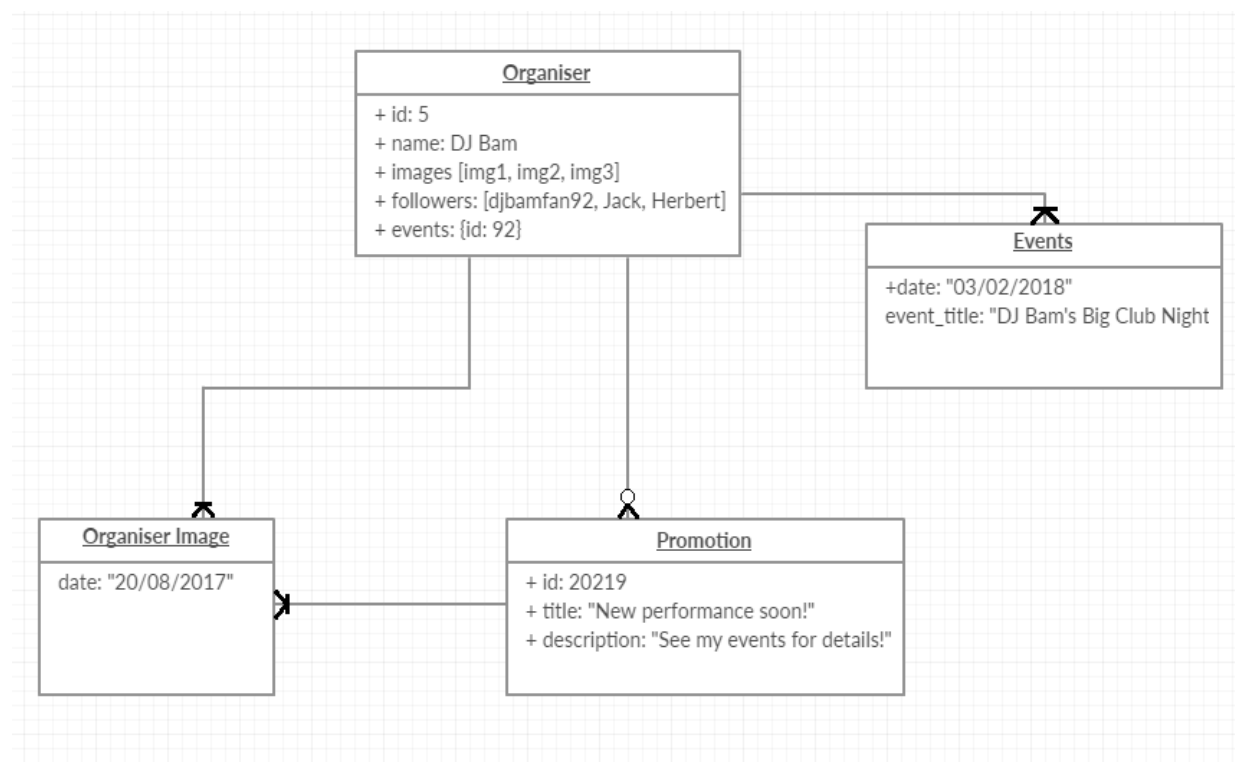
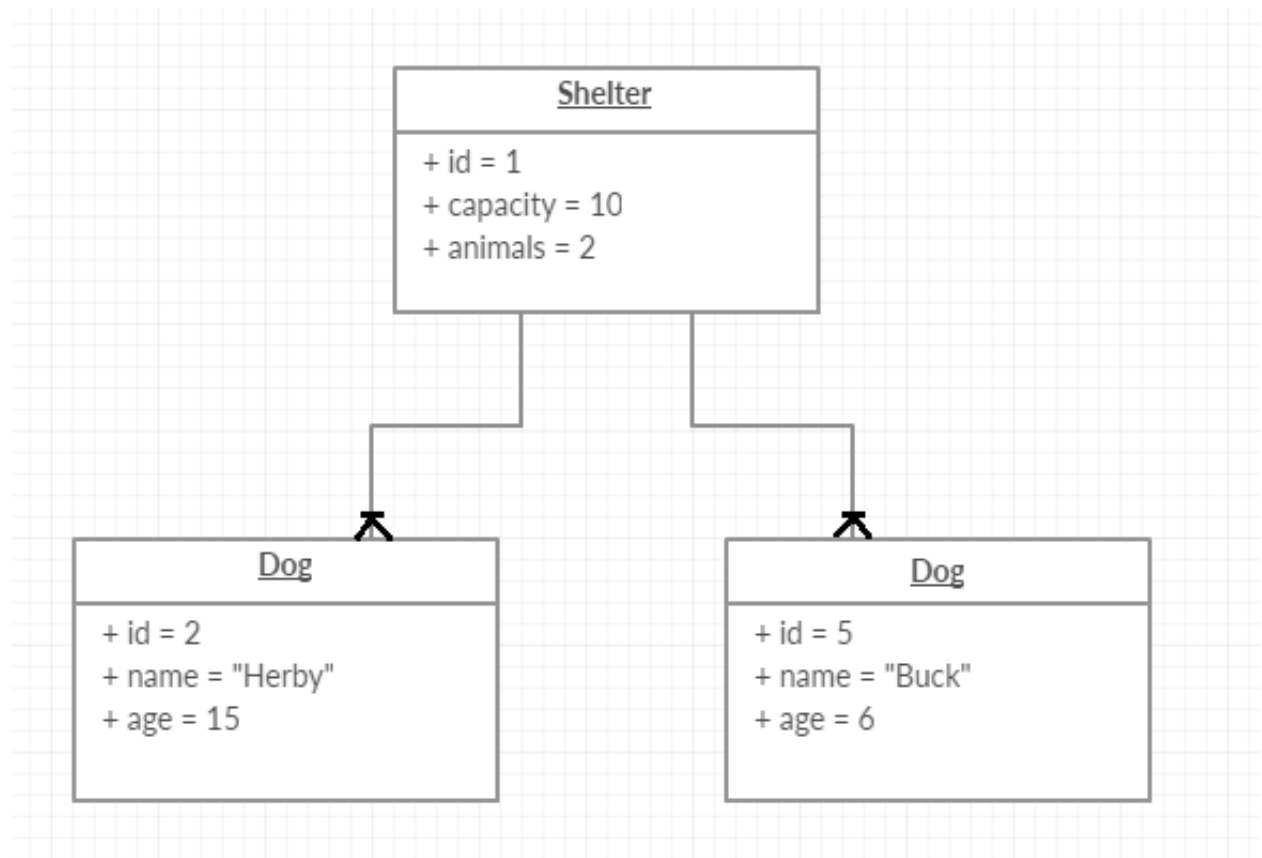
P 6 – Produce two wireframe designs:



P 7 – Produce two system interaction diagrams (sequence and/or collaboration diagrams):



P 8 – Produce two object diagrams:



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:

1. This algorithm is a Bubble Sort that I used to iterate through an array of hashes, each hash containing information on a cat. I used this because I needed a way to sort all of the cats from youngest to oldest. Although the Bubble Sort does not scale well to large data sets, this code was designed for a minimal number of cats that would realistically be restricted to the physical size of the adoption shelter that the program was built for.

```
1  def cat_sort(cats)
2      n = cats.length
3      loop do
4          swapped = false
5
6          (n-1).times do |i|
7              if cats[i][:age] > cats[i+1][:age]
8                  cats[i], cats[i+1] = cats[i+1], cats[i]
9                  swapped = true
10             end
11         end
12
13         break if not swapped
14     end
15     cats
16 end
```

2. This algorithm is a Merge Sort. It was used because I needed to sort a list of patients by a priority value with a higher value being a higher priority. Unlike the list of cats, a data set of patients can be rather large and so I needed a more efficient sorting method, which a Merge Sort provides.

```

def patient_priority_sort(patients)
  if patients.length <= 1
    patients
  else
    mid = (patients.length / 2).floor
    left = patient_priority_sort(patients[0..mid-1])
    right = patient_priority_sort(patients[mid..patients.length])
    merge(left, right)
  end
end

def merge(left, right)
  if left.empty?
    right
  elsif right.empty?
    left
  elsif left[0][:priority] > right[0][:priority]
    [left[0]] + merge(left[1..left.length], right)
  else
    [right[0]] + merge(left, right[1..right.length])
  end
end

```

P 10 – Take a screenshot of an example of pseudocode for a function:

```

def It_should_add_a_new_guest(new_guest)
  # It should check if the current number of occupants is less than 3
  # It should also check if the new guest has 10 money to pay for entry
  # It should then take the guests 10 money and add them to the list of occupants
  # If the new guest didn't have enough money or the room had 3 people, print out an error
end

```

P 11 – Take a screenshot of one of your projects where you have worked alone and attach the Github link:

<https://github.com/mjshield/Record-Shop-Project>

mjshield / Record-Shop-Project

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Stock-tracking website created in Ruby and styled in CSS and HTML, with use of Sinatra. [Add topics](#)

30 commits 1 branch 0 releases 1 contributor

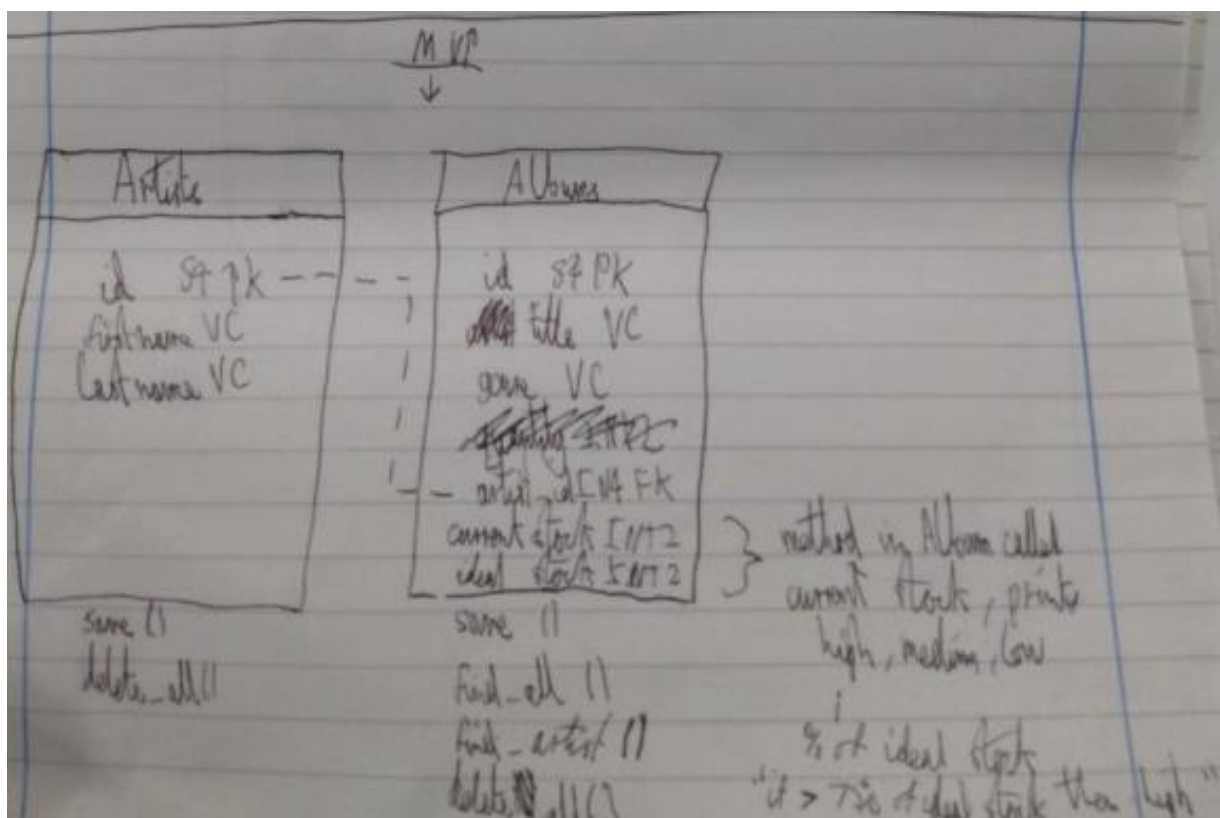
Branch: master New pull request Create new file Upload files Find file Clone or download

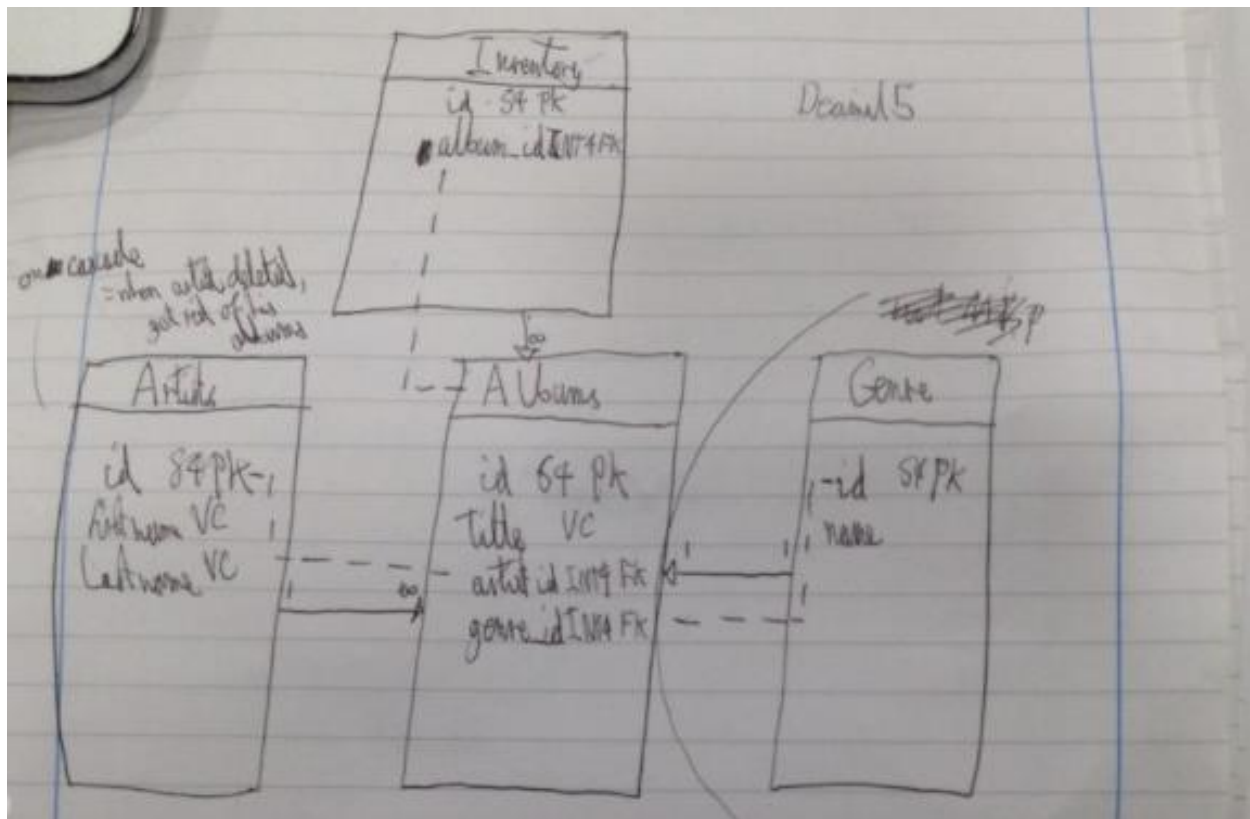
mjshield Removed header on '/' page Latest commit e8ca1a8 on 26 Jul

controllers	Style and UX improvements	3 months ago
db	Minor refactoring, final commit	3 months ago
models	Minor refactoring, final commit	3 months ago
public	Changed background and main header style	3 months ago
specs	Added create artists, adjusted stock calculation	3 months ago
views	Removed header on '/' page	3 months ago
app.rb	Added create artists, adjusted stock calculation	3 months ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

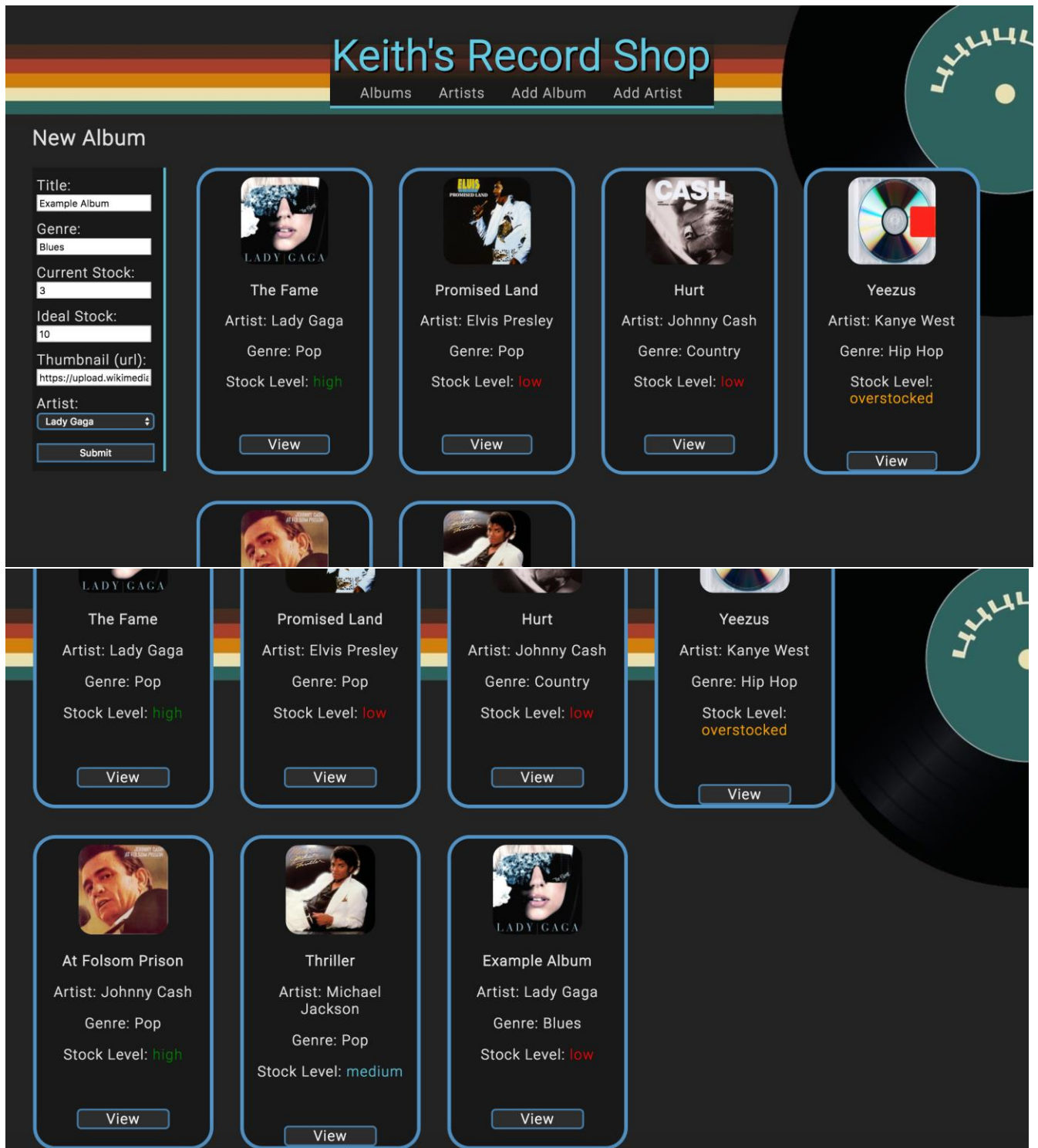
P12 – Take screenshots or photos of your planning and the different stages of development to show changes:





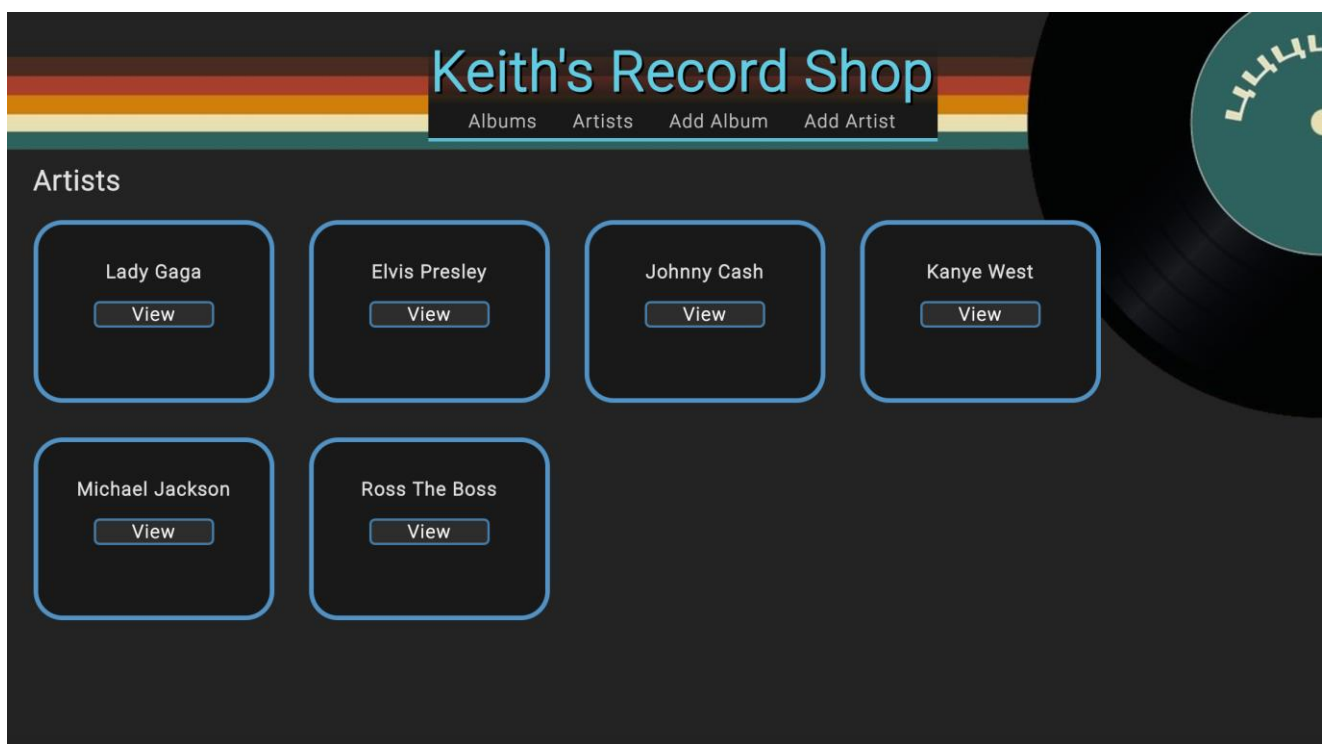
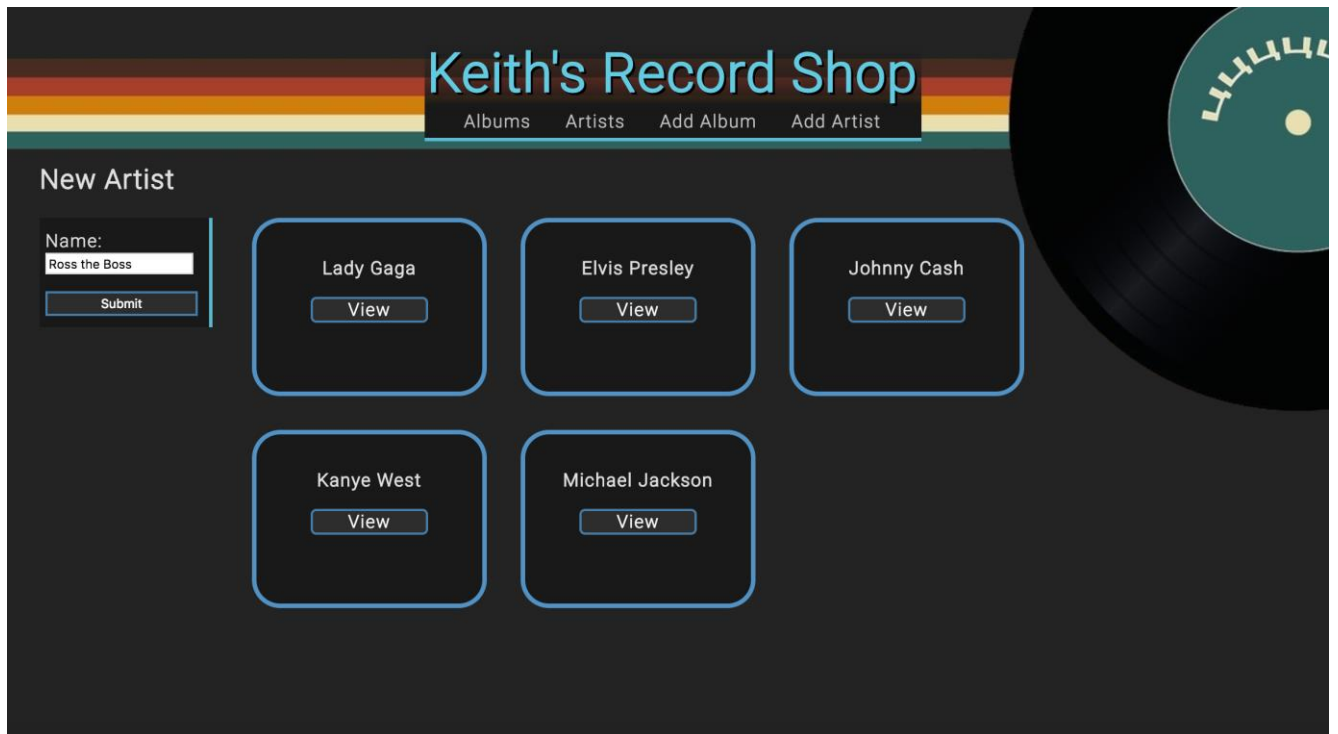
P 13 - Show user input being processed according to design requirements. Takes a screenshot of:

- The user inputting something into your program
- The user input being saved or used in some way



P 14 – Show an interaction with data persistence. Take a screenshot of:

- Data being inputted into your program
- Confirmation of the data being saved



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

Keith's Record Shop

AlbumsArtistsAdd AlbumAdd Artist

Edit Album

Title:
Promised Land

Genre:
Rock

Current Stock:
4

Ideal Stock:
20

Thumbnail (url):
<https://images-na.ssl-ir>

Artist:
Elvis Presley

Submit

Keith's Record Shop

AlbumsArtistsAdd AlbumAdd Artist

Album details:

Edit

Delete



Promised Land
Artist: Elvis presley
Genre: Rock
Stock Level: low

P 16 – Show an API being used with your program. Take a screenshot of:

- The code that uses or implements the API
- The API being used by the program whilst running

```
1 var api = {  
2   key: 'http://www.giantbomb.com/api/characters?api_key=438e1c03c59747526afe0b3233315fcc22dfac67&format=json&sort=date_last_updated:desc'  
3 }  
4
```

```

9  var makeRequest = function(url, callback) {
10     var request = new XMLHttpRequest();
11     request.open('GET', url);
12     request.addEventListener('load', callback);
13     request.send();
14 }
15
16 var requestComplete = function() {
17     if( this.status !== 200 ) return;
18
19     var jsonString = this.responseText;
20     var charactersObject = JSON.parse(jsonString);
21     console.log(charactersObject);
22
23     var characters = charactersObject.results;
24     console.log(characters);
25     populateList(characters);
26 }
27
28 var populateList = function(characters) {
29     var ul = document.querySelector('#character-list');
30
31     characters.forEach( function(character) {
32         var li = document.createElement('li');
33         var img = document.createElement('img')
34         var imageUrl = "https://upload.wikimedia.org/wikipedia/commons/a/ac/No_image_available.svg"
35         var gender = "Unknown"
36         var deck = "This character's description hasn't been filled in!"
37         var firstGame = "Unknown"
38
39         if (character.image !== null) {var imageUrl = character.image.small_url;}
40         if (character.deck !== null) {var deck = character.deck}
41         if (character.first_appeared_in_game !== null) {var firstGame = character.first_appeared_in_game.name}
42
43         if (character.gender !== null) {
44             if (character.gender === 1) {var gender = "Male"}
45             else if (character.gender === 2) {var gender = "Female"}
46             else {}
47         }
48
49         li.innerHTML =
50         '<p>' + character.date_last_updated + '</p>' +
51         '<p><a href="' + character.site_detail_url + '>' + '' + '</p>' +
53         '<p><b>' + character.name + '</b></a></p>' +
54         '<p><i>' + deck + '</i></p>' +
55         '<p><b>Gender:</b>' + gender + '</p>' +
56         '<p><b>First appeared in:</b>' + firstGame + '</p>';
57         ul.appendChild(li);
58     })
59 }

```

Recently Updated Character Pages:

- "2017-10-24 00:50:00"



[Kuriyama Mirai](#)

A female lead character in the anime "Beyond the boundaries" or "Kyoukai no Kanata". She possess the ability to manipulate blood, which is looked down upon between the Spirit World Warriors clans.

Gender: Unknown

First appeared in: Unknown

- "2017-10-24 00:49:23"



[Tsunayoshi Sawada](#)

The lead character of the anime "Reborn!" or "Katekyō Hitman Reborn!". He is first a cowardly character but due to unforeseen circumstances, he is forced to change and fulfill his role.

Gender: Unknown

First appeared in: Unknown

- "2017-10-24 00:49:01"



[Kashiwazaki Sena](#)

A female lead character in the anime "I don't have many friends" or "Boku wa Tomodachi ga Sukunai". She is the daughter of the headmaster of the school that she attends.

Bug Description	Pass/Fail	Bug Fix	Pass/Fail
A user can view an index of all albums	Failed	Added indexing function to initialisation of albums page	Passed
A user can edit an existing album	Failed	Added button to each album container to edit that album	Passed
A user can create a new artist	Failed	Added button for creating a new artist on the artist index page	Passed
A user can see all of a specific artist's albums	Failed	Added the album index to the artist details page, restricted to albums by that artist	Passed
A user can return to the landing page	Failed	The website title now functions as a home button which resets to the original view	Passed
A user will be informed that they cannot leave a field blank when adding a new artist	Failed	Added an error handler which stops artist submittal and informs the user that they must fill every field	Passed

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


```
1  require 'minitest/autorun'
2  require_relative '../record_shop'
3
4  class TestRecordShop < Minitest::Test
5
6    def setup
7
8      @record_shop = {
9        records: [
10          {
11            title: "Happy as Larry",
12            artist: "Larry"
13          },
14          {
15            title: "Cleo",
16            artist: "Catman"
17          }
18        ]
19      }
20    end
21
22
23    def test_stock_count
24      count = stock_count(@record_shop)
25      assert_equal(2, count)
26    end
27
28  end
29
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: powershell



Running:

F

Finished in 0.001811s, 552.1851 runs/s, 552.1851 assertions/s.

1) Failure:

TestRecordShop#test_stock_count [./record_shop_spec.rb:25]:

Expected: 2

Actual: 1

1 runs, 1 assertions, 1 failures, 0 errors, 0 skips

```
1 def stock_count(record_shop)
2   | return record_shop.count
3 end
```

```
1 def stock_count(record_shop)
2   | return record_shop[:records].count
3 end
```

Running:

.

Finished in 0.001721s, 581.1815 runs/s, 581.1815 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips