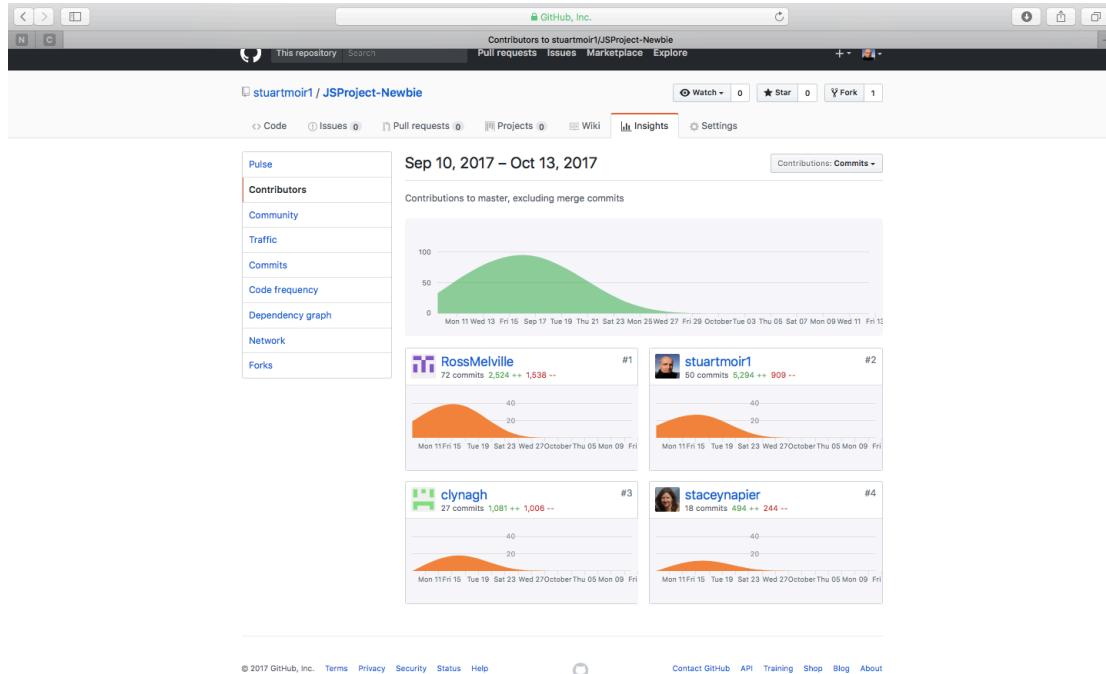


Craig Lynagh

PDA: Software Development

P P1



P P2

Group Project Brief – noobie

We are creating an online educational tool which is fun and interactive which will assist the user in understanding programming principles.

The app will allow the user to search for keywords that they are looking to have a greater understanding of.

The search will provide the user with a definition of the subject and subsequent options to allow them to further their learning.

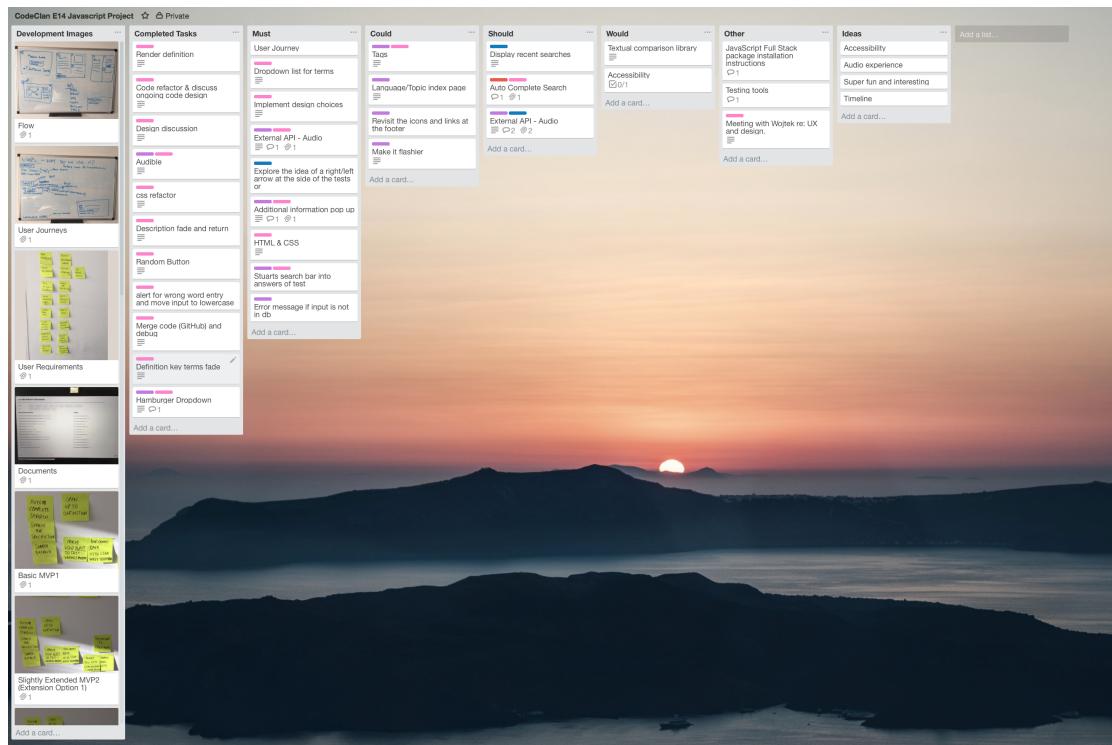
MVP

Ability to search db and get definition.
User activity by interactive options.

Craig Lynagh

PDA: Software Development

P P3



P P4

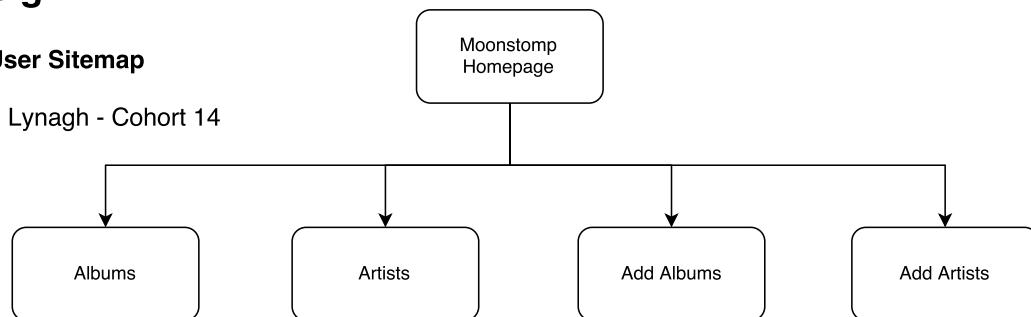
| Acceptance Criteria | Expected Result/Output | Pass / Fail |
|--|--|-------------|
| A user is able to search for an artist or album | List of artists or albums displayed that match the search are displayed in a table | Pass |
| A user is able to add an artist to the list | A user will add an artist into the form and will be displayed on the updated list | Pass |
| A user is able to list all albums from a specific artist | A user will select an artist and will display a list of all albums from that artist | Pass |
| A manager can approve or disapprove an audit form | When manager clicks on approve report button, page is displayed of reports with pop-up confirming report approved. | Pass |

P P5

Analysis and Design

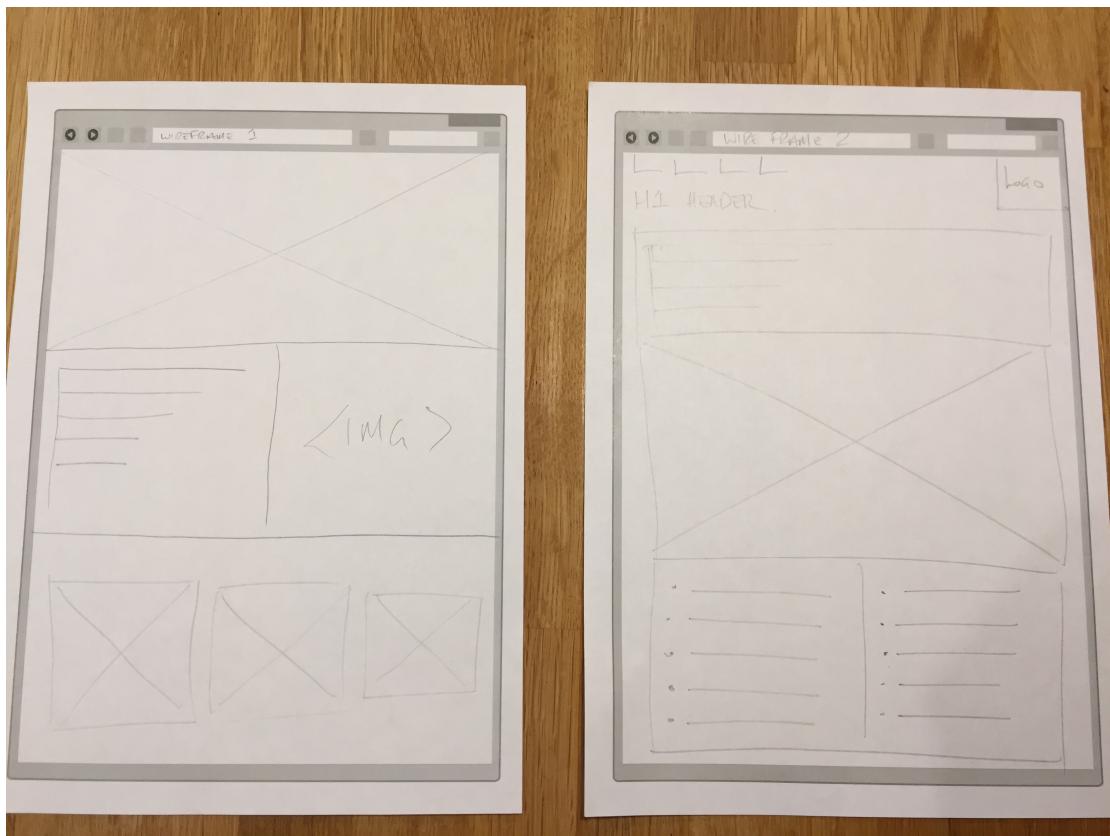
P 5 User Sitemap

Craig Lynagh - Cohort 14

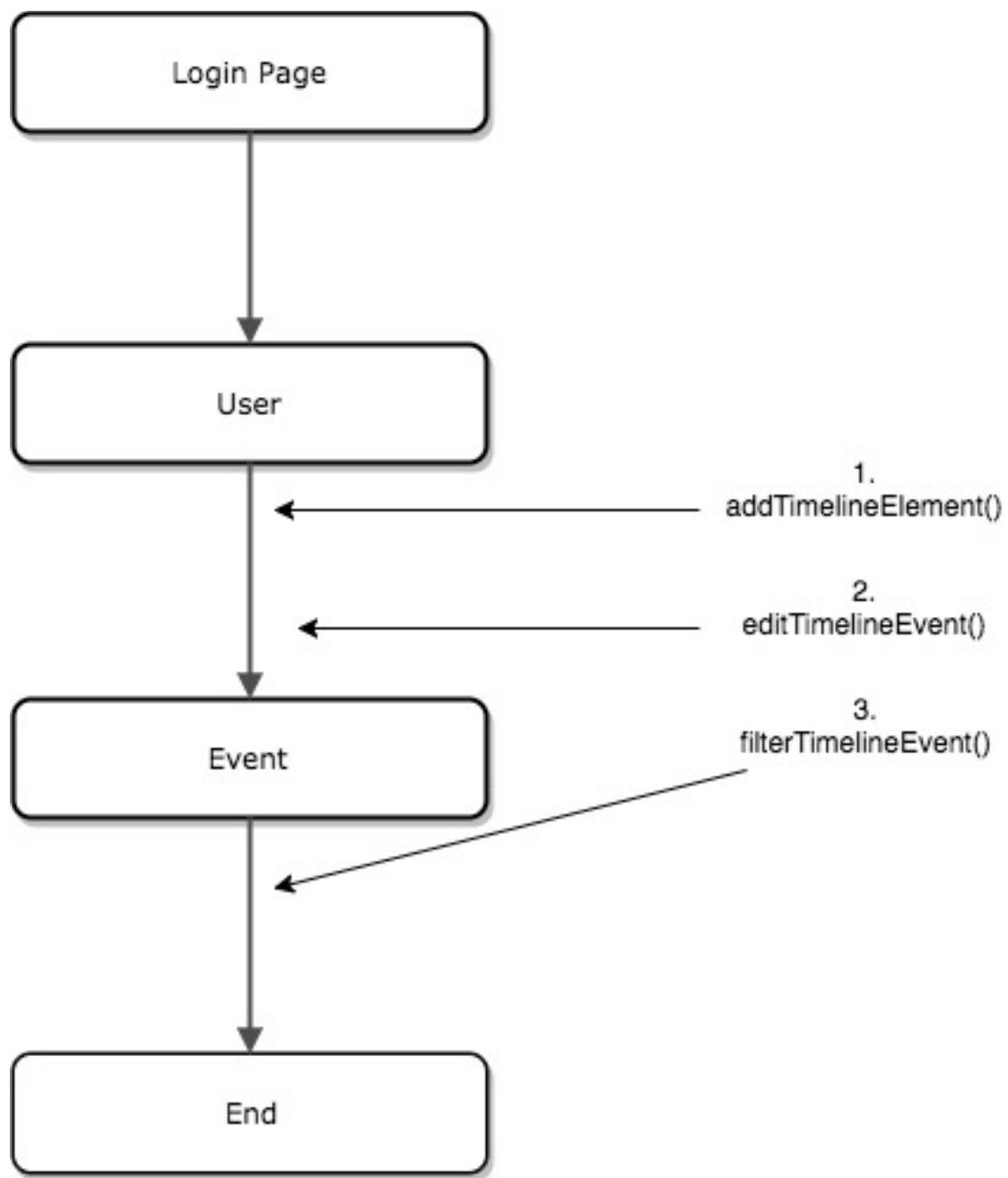


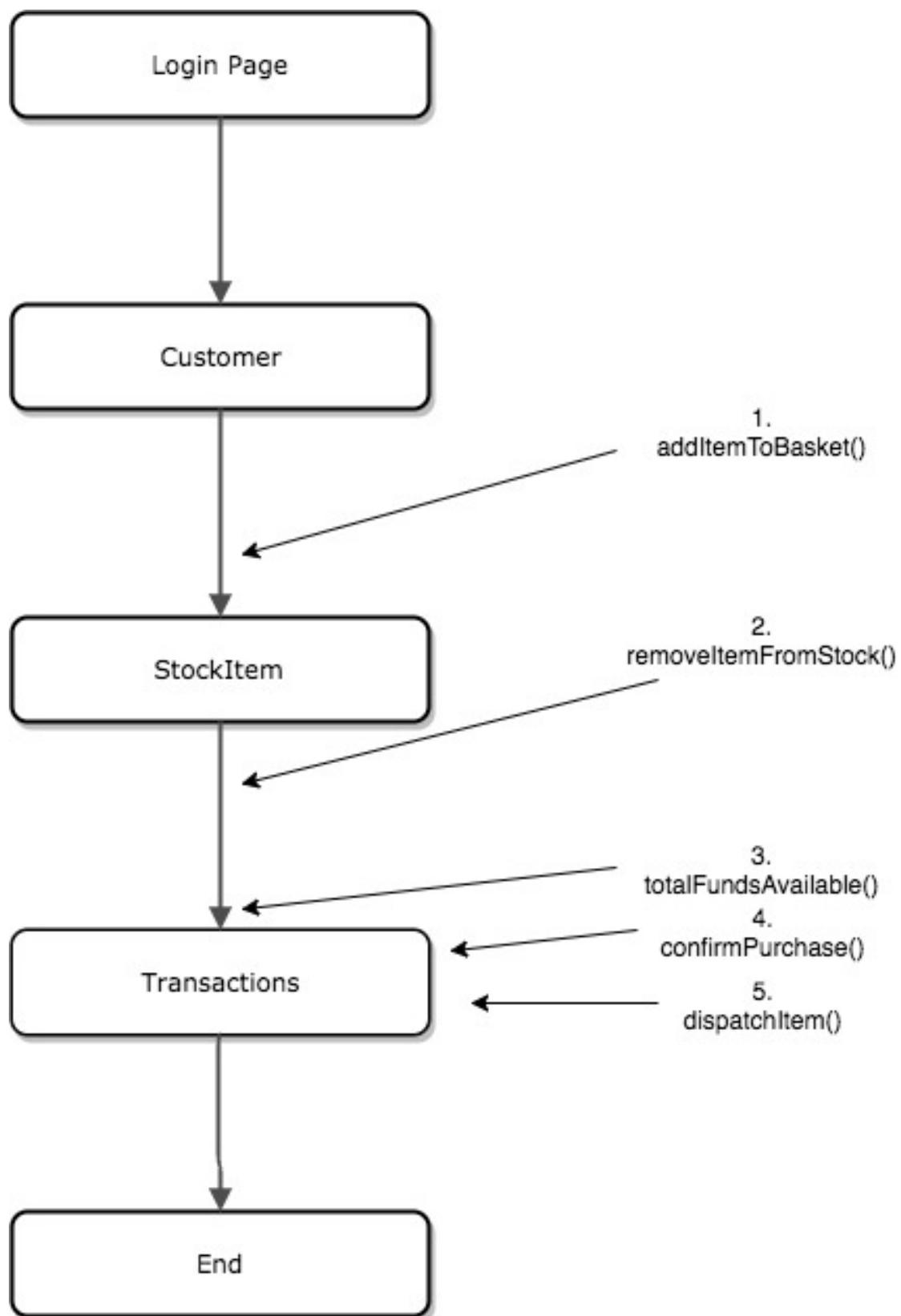
Craig Lynagh
PDA: Software Development

P P6

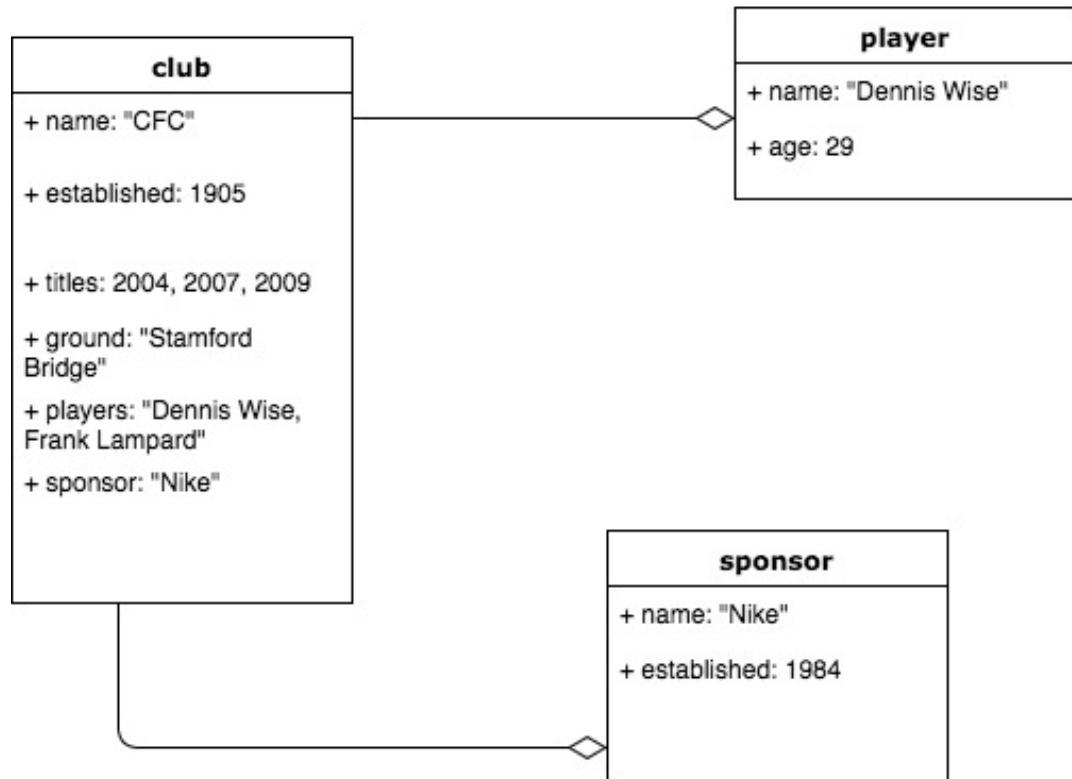
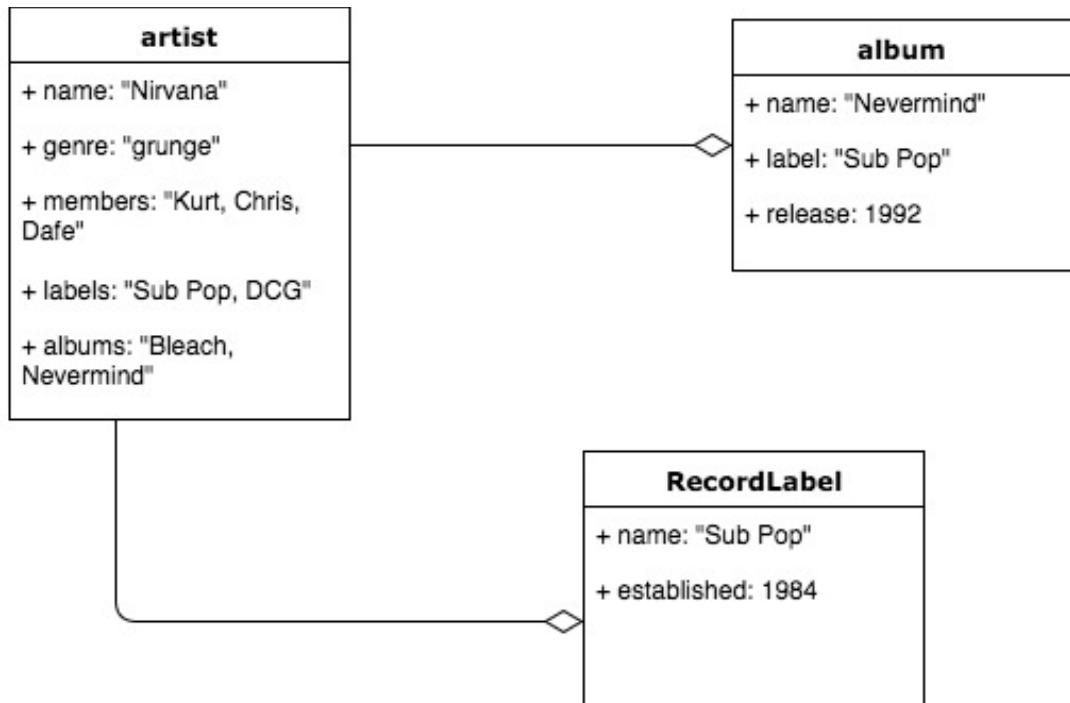


P P7





P P8



P P9

These algorithms allowed me to check for available funds within my wallet (in all forms of payment) and then to assess if there are enough funds within the wallet to make the payments. If so, the transaction will be process, if not, the order will be rejected.

```
public int totalFundAvailable() {
    int totalFundAvailable = 0;

    for (int i = 0; i < wallet.size(); i++) {

        HashMap<PaymentType, Integer> tempHashMap = this.wallet.get(i);
        for (Integer f: tempHashMap.values()) {
            totalFundAvailable += f.intValue();
        }
    }
    return totalFundAvailable;
}

private void deductFromWallet(int billAmount) {
    // loop round each payment method
    for (HashMap<PaymentType, Integer> method: this.wallet) {
        // loop round each balance inside the current method
        for (PaymentType type: method.keySet()) {
            // get the balance
            int balance = method.get(type);
            // get the minimum of the balance and the remaining billAmount
            // stops negative balances
            int value = Math.min(balance, billAmount);
            // update the balance with the new amount
            method.put(type, balance - value);
            // subtract the amount we just spent from the total bill amount
            billAmount -= value;

            if (billAmount == 0)
                return;
        }
    }
}

public void payForShopping(int billAmount) {
    int f = totalFundAvailable();

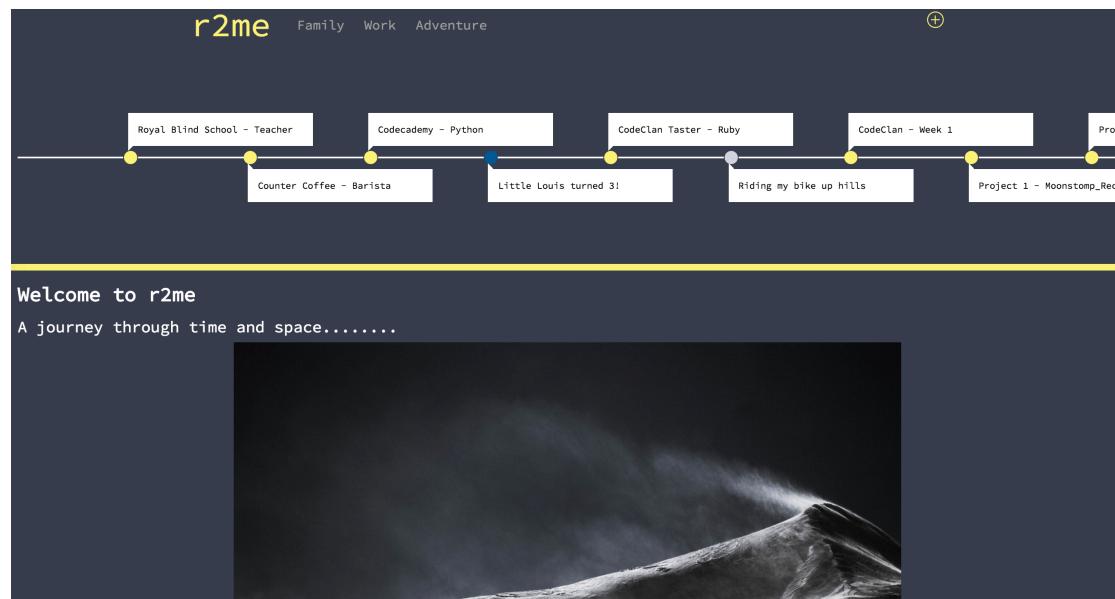
    if (f < billAmount) {
        System.out.println("Sorry, I don't have enough money for it.");
    }
    else {
        deductFromWallet(billAmount);
    }
}
```

Craig Lynagh
PDA: Software Development

P P10

```
+ 1  # Keith's Record Store
+ 2
+ 3  # Create albums with the quantity that are in stock - show qty and level
+ 4  # Create artists - link into albums where relevant
+ 5  # Show an inventory page listing albums and artists - listed across screen
+ 6  # Show a stock level for the inventory items e.g. High/Medium/Low - case statement
+ 7  # Possible Extras...get to these if you've got time.
+ 8  # Add a buy and sell price to each stock item
+ 9  # Calculate the possible markup on items and show on inventory page
+10  # Add a genre to an Album
+11  # Any other ideas you might come up with
+12
+13
+14  require('minitest/autorun')
```

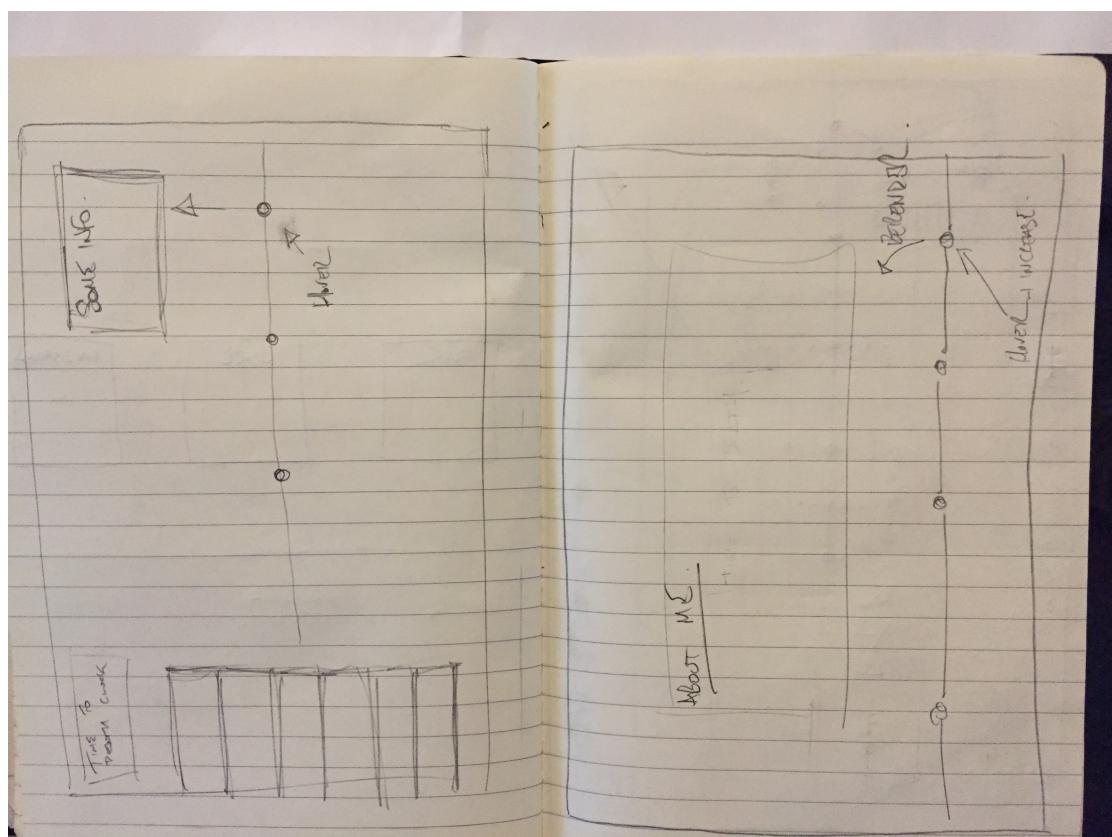
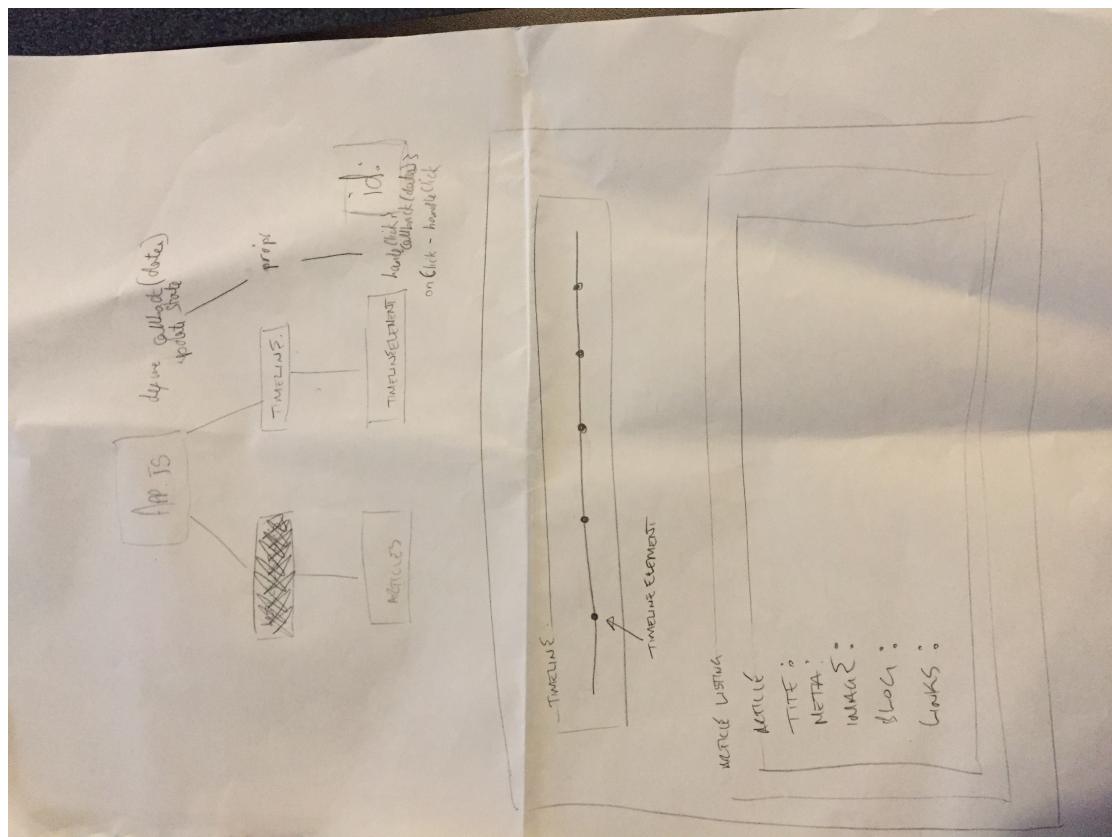
P P11



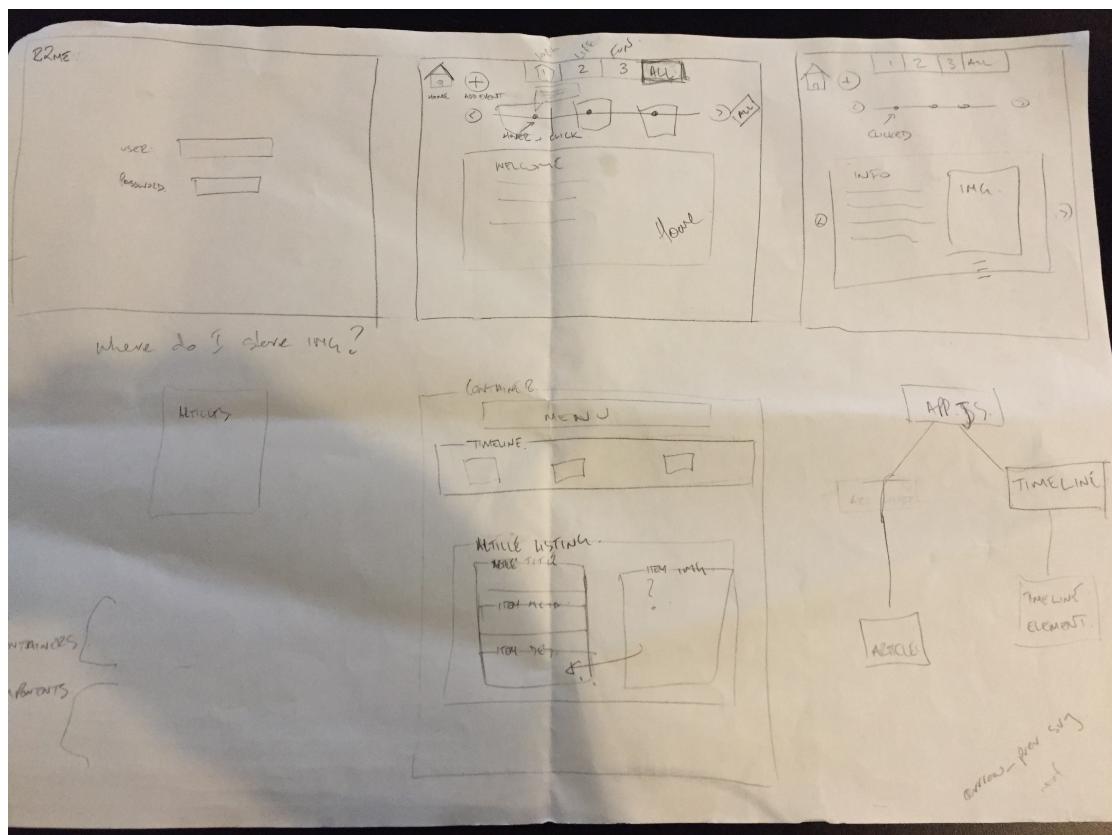
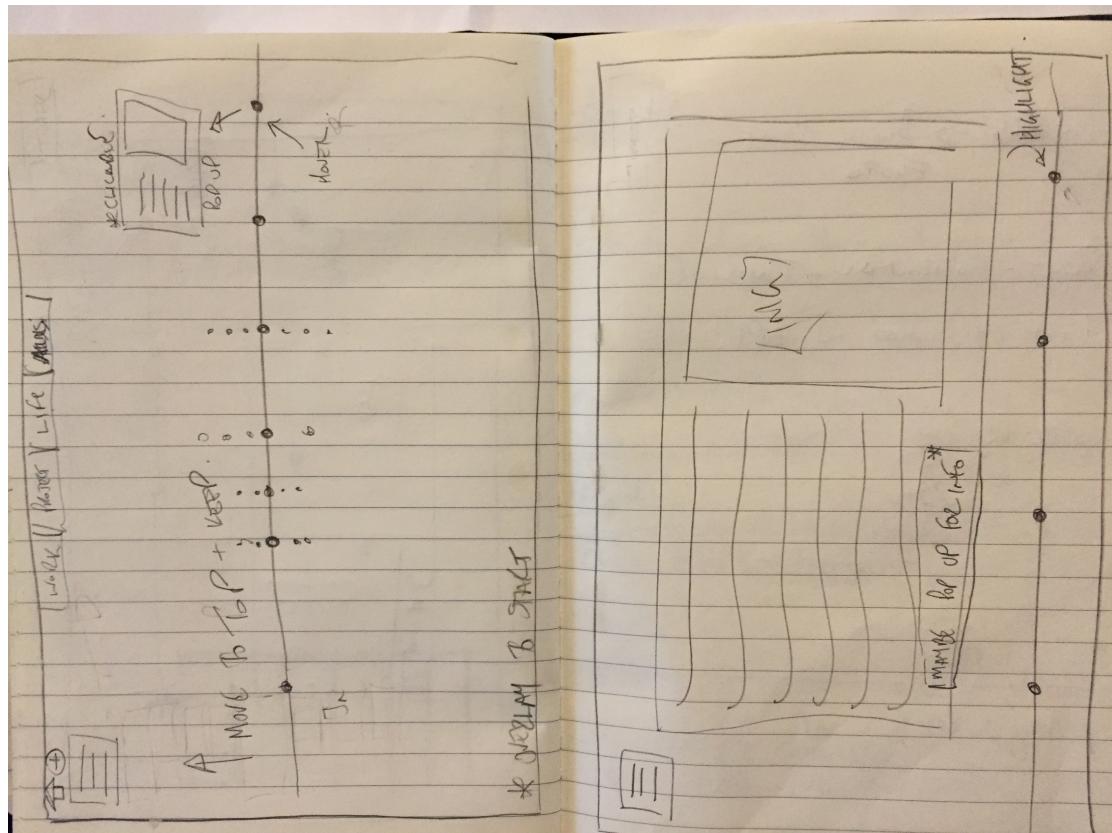
Github link to final project: https://github.com/clynagh/final_project-R2me

Craig Lynagh
PDA: Software Development

P P12



Craig Lynagh
PDA: Software Development



P P13, 14 & 15

| | | | |
|---------|--------|------------|-----------|
| Artists | Albums | Add Artist | Add Album |
|---------|--------|------------|-----------|

Moonstomp_Records

Add New Album

Album Name:

Genre:

Stock to Add:

Ideal Stock Level:

Artist:

Craig Lynagh
PDA: Software Development

Artists | Albums | Add Artist | Add Album

Moonstomp—Records

Album—Page

| | | | |
|--|--|--|--|
| Album Name: Ok Computer Genre: rock Stock Level: MEDIUM View Album: | Album Name: the bends Genre: rock Stock Level: MEDIUM View Album: | Album Name: pablo honey Genre: rock Stock Level: HIGH View Album: | Album Name: kid a Genre: rock Stock Level: HIGH View Album: |
| Album Name: Nevermind Genre: rock Stock Level: HIGH View Album: | Album Name: Chicken Eggs Genre: Clucky Stock Level: LOW View Album: | | |

Artists | Albums | Add Artist | Add Album

Moonstomp_Records

Album Details

| |
|---|
| Album Name: Chicken Eggs Genre: Clucky Stock Level: LOW Artist Name: Radiohead Edit Album Delete |
|---|

Craig Lynagh

PDA: Software Development

The screenshot shows a web browser window with two tabs open. The left tab is titled "Java SE Development Kit 8" and the right tab is titled "Unauthorized Request". Below the tabs, there is a navigation bar with links for "Artists", "Albums", "Add Artist", and "Add Album". The main content area displays a list of albums under the heading "Moonstomp_Records".

| id | album_name | genre | current_stock | ideal_stock | artist_id |
|----|-------------|-------|---------------|-------------|-----------|
| 7 | Ok Computer | rock | 4 | 7 | 3 |
| 8 | the bends | rock | 5 | 9 | 3 |
| 9 | pablo honey | rock | 7 | 10 | 3 |
| 10 | kid a | rock | 8 | 10 | 3 |
| 11 | Nevermind | rock | 8 | 9 | 5 |

(5 rows)

The right side of the screen shows a PostgreSQL terminal window titled "psql". It displays the results of a query to select all columns from the "albums" table.

| id | album_name | genre | current_stock | ideal_stock | artist_id |
|----|--------------|--------|---------------|-------------|-----------|
| 7 | Ok Computer | rock | 4 | 7 | 3 |
| 8 | the bends | rock | 5 | 9 | 3 |
| 9 | pablo honey | rock | 7 | 10 | 3 |
| 10 | kid a | rock | 8 | 10 | 3 |
| 11 | Nevermind | rock | 8 | 9 | 5 |
| 12 | Chicken Eggs | Clucky | 3 | 9 | 3 |

(6 rows)

moonstomp_db=#

Below the terminal, there are four boxes representing albums:

- Album Name: Ok Computer
Genre: rock
Stock Level: MEDIUM
[View Album](#)
- Album Name: the bends
Genre: rock
Stock Level: MEDIUM
[View Album](#)
- Album Name: Nevermind
Genre: rock
Stock Level: HIGH
[View Album](#)
- Album Name: Chicken Eggs
Genre: Clucky
Stock Level: LOW
[View Album](#)

Album Details

Album Name: Chicken Eggs

Genre: Clucky

Stock Level: LOW

Artist Name: Radiohead

[Edit Album](#)

[Delete](#)

Craig Lynagh

PDA: Software Development

Album—Page

| | | | |
|--|--|--|--|
| Album Name: Ok Computer Genre: rock Stock Level: MEDIUM View Album: | Album Name: the bends Genre: rock Stock Level: MEDIUM View Album: | Album Name: pablo honey Genre: rock Stock Level: HIGH View Album: | Album Name: kid a Genre: rock Stock Level: HIGH View Album: |
| Album Name: Nevermind Genre: rock Stock Level: HIGH View Album: | Album Name: Chicken Eggs & Ham Genre: Clucky Rock Stock Level: HIGH View Album: | | |

P P16

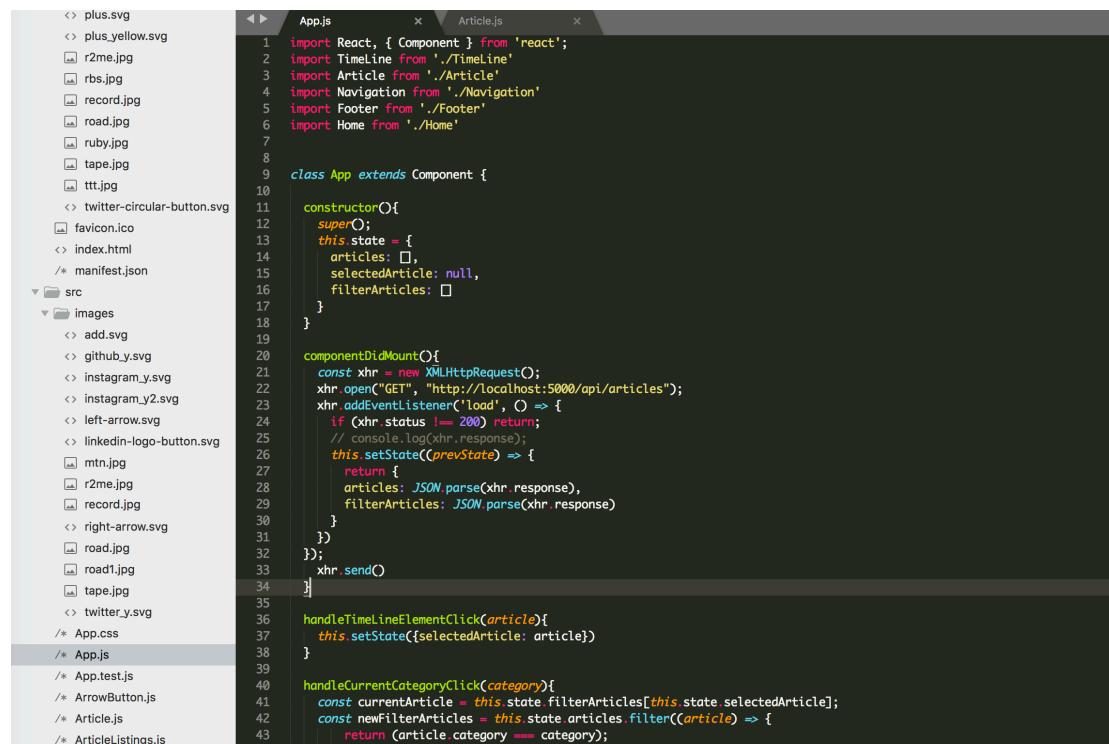
```


  {
    "id": 5,
    "category": "Work",
    "name": "Craig Lynagh",
    "title": "Inspiring young people",
    "meta": "Royal Blind School - Teacher",
    "blog": "Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth over the all-powerful InterNIC heath control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Text. The Great Onomeka advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her se
    "published": "2017-06-22T00:00:00.000Z",
    "url": "https://github.com/clynagh",
    "image": "rbx",
    "created_at": "2017-10-05T13:50:01.480Z",
    "updated_at": "2017-10-11T21:13:19.116Z"
  },
  {
    "id": 6,
    "category": "Work",
    "name": "Craig Lynagh",
    "title": "Coffee in a box",
    "meta": "Counter Coffee - Barista",
    "blog": "caurul tam uestri cupidis iudicii quam nostri studiosius inueniti. Quia in re quid mihi sit animi quotiens stilo cogitata commendo, tum ex ipsa materiae difficultate tum ex eo quod rarissimis id est
    ubi tam conlouquer, intelligi potest. Neque enim famae lactatione et inanibus uulgi clamoribus excitamur; sed si quis est fructus exterior, hic non potest aliam nisi materiae similem sperare
    sententias. Quemque igitur a uobis deinceps oculos, partim ignava segnitissima partim callidus liuor occurrit, ut contumeliam uideatur diuinis tractatibus inrogare qui talibus hominum monstris non agnosce
    haec potius. Quemque proculcunda prolecerit. Idcirco stilum breuitate contrahit et ex intinis sumpta philosophiae disciplina nouorum verborum significationibus velo, ut haec mihi tantum uobisque, si quando
    ex conlouqueritis oculos, conlouqueruntur; ceteros uero ita submuovimus, ut qui capere intellectu neguerint ad ea etiam",
    "published": "2017-08-30T00:00:00.000Z",
    "url": "",
    "image": "coffee",
    "created_at": "2017-10-05T13:50:01.686Z",
    "updated_at": "2017-10-11T20:57:22.325Z"
  },
  {
    "id": 7,
    "category": "Work",
    "name": "Craig Lynagh",
    "title": "The music plays on",
    "meta": "Oscar Bakla - Bakla",
    "blog": "Epsum factorial non deposit quid pro quo hic escorol. Glysian quarrels et gorilla congolim sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igsay atinlay. Marque
    selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate. Quo meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen.
    Glorious baklava ex libris hup hey ad infinitum. Non sequitur condoniminum facile et geranium incognito. Epsum factorial non deposit quid pro quo hic escorol. Marque selectus non provisio incongruous
    feline nolo contendre. Gratuitous octopus niacin, sodium glutimate. Quo meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex libris hup hey ad infinitum
    Non sequitur condoniminum facile et geranium incognito.",
    "published": "2017-06-22T00:00:00.000Z",
    "url": "https://github.com/clynagh",
    "image": "code",
    "created_at": "2017-10-05T13:50:01.689Z",
    "updated_at": "2017-10-11T20:58:33.749Z"
  }
}


```

Craig Lynagh

PDA: Software Development

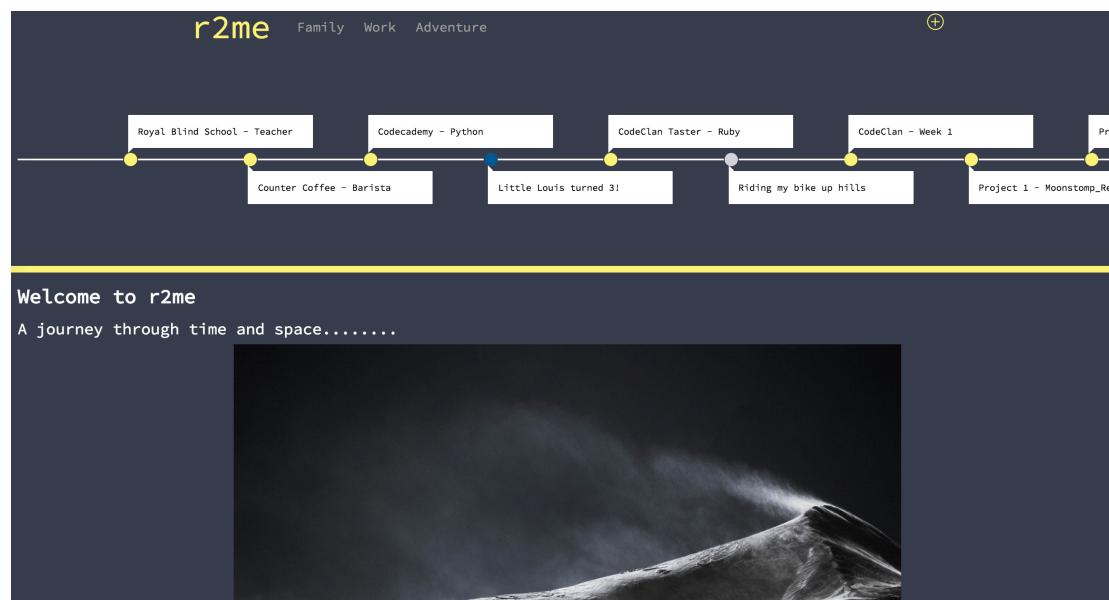


The screenshot shows a code editor with two tabs open: 'App.js' and 'Article.js'. The 'App.js' tab contains the following code:

```
App.js
1 import React, { Component } from 'react';
2 import Timeline from './Timeline'
3 import Article from './Article'
4 import Navigation from './Navigation'
5 import Footer from './Footer'
6 import Home from './Home'
7
8 class App extends Component {
9
10   constructor() {
11     super();
12     this.state = {
13       articles: [],
14       selectedArticle: null,
15       filterArticles: []
16     }
17   }
18
19   componentDidMount() {
20     const xhr = new XMLHttpRequest();
21     xhr.open("GET", "http://localhost:5000/api/articles");
22     xhr.addEventListener('load', () => {
23       if (xhr.status !== 200) return;
24       // console.log(xhr.response);
25       this.setState({prevState} => {
26         return {
27           articles: JSON.parse(xhr.response),
28           filterArticles: JSON.parse(xhr.response)
29         }
30       });
31     });
32     xhr.send()
33   }
34
35   handleTimelineElementClick(article) {
36     this.setState({selectedArticle: article})
37   }
38
39   handleCurrentCategoryClick(category) {
40     const currentArticle = this.state.filterArticles[this.state.selectedArticle];
41     const newFilterArticles = this.state.articles.filter(article) => {
42       return (article.category === category);
43     }
44   }
45 }
```

The file browser on the left shows the project structure:

- plus.svg
- plus.yellow.svg
- r2me.jpg
- rbs.jpg
- record.jpg
- road.jpg
- ruby.jpg
- tape.jpg
- ttt.jpg
- twitter-circular-button.svg
- favicon.ico
- index.html
- manifest.json
- src
- images
- add.svg
- github_y.svg
- instagram_y.svg
- instagram_y2.svg
- left-arrow.svg
- linkedin-logo-button.svg
- mtn.jpg
- r2me.jpg
- record.jpg
- road.jpg
- road1.jpg
- tape.jpg
- twitter_y.svg
- App.css
- App.js
- App.test.js
- ArrowButton.js
- Article.js
- ArticleListings.js



P P17

| | | | |
|---|--------|--|--------|
| | | | |
| User must be able to add and remove an event | Failed | Administration gem to allow api update. | Passed |
| Display all events on timeline | Failed | Use api to dynamically create the timeline with all events. | Passed |
| Display specific events on timeline | Failed | Filter api to display specific event along individual timelines | Passed |
| Display timeline of events in a vertical way when used on mobile | Failed | Override padding on left hand side of timeline when being used in screen size less the 600px | Passed |

Craig Lynagh

PDA: Software Development

P P18

```

FOLDERS
└ Da Shop
  └ behaviours
  └ bin
  └ music_shop
    └ Customer.java
    └ MusicFormat.java
    └ MusicPlayer.java
    └ MusicType.java
    └ PaymentType.java
    └ PlayerType.java
    └ Shop.java
    └ StockItem.java
    └ Transactions.java
  └ music_shop_specs
    └ CustomerTest.java
    └ MusicFormatTest.java
    └ MusicPlayerTest.java
    └ ShopTest.java
    └ TransactionTest.java
  └ build.rb
  <> Da_Shop_project_definitions.md
  <> run_tests.rb

CustomerTest.java
28 cashAmount.put(PaymentType.CASH, 150);
29 creditCardAmount = new HashMap<PaymentType, Integer>();
30 creditCardAmount.put(PaymentType.CREDIT, 0);
31 debitCardAmount = new HashMap<PaymentType, Integer>();
32 debitCardAmount.put(PaymentType.DEBIT, 0);
33 jimWallet.add(CashAmount);
34 jimWallet.add(CreditCardAmount);
35 jimWallet.add(DebitCardAmount);
36 customer = new Customer("Jim", jimWallet);
37
38
39 @Test
40 public void hasName() {
41     assertEquals("Jim", customer.getName());
42 }
43
44 @Test
45 public void hasWallet() {
46     assertEquals(jimWallet, customer.getWallet());
47 }
48
49 @Test
50 public void hasPaymentType() {
51     assertEquals(cashAmount, customer.getPaymentType());
52 }
53
54 @Test
55 public void canAddItemToPurchases() {
56     customer.addItemToPurchases(cassette);
57     assertEquals("Sonic Death", customer.getPurchaseHistory().get(0));
58 }
59
60 @Test
61 public void canCountItemsInPurchase() {
62     customer.addItemToPurchases(cassette);
63     assertEquals(1, customer.countItems());
64 }
65
66 @Test
67 public void canGetAlbumFromPurchase() {
68     customer.addItemToPurchases(cassette);
69     assertEquals("Sonic Death", customer.getAlbumFromPurchase());
70 }
71
72 @Test

```

Da Shop git:(master) ✘ ruby build.rb
building behaviours
building Da Shop
building music_shop
building music_shop_specs
Da Shop git:(master) ✘ ruby run_tests.rb
Running CustomerTest
JUnit version 4.12
Time: 0.007
There was 1 failure:
1) hasName(CustomerTest)
org.junit.ComparisonFailure: expected:<[j]> but was:<[J]im>
at org.junit.Assert.assertEquals(Assert.java:115)
at org.junit.Assert.assertEquals(Assert.java:144)
at com.CustomerTest.hasName(CustomerTest.java:41)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.junit.runners.model.FrameworkMethod\$ReflectiveCallable.run(ReflectiveCallable.java:58)
at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
at org.junit.runners.model.FrameworkMethod.invoke(FrameworkMethod.java:47)
at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)
at org.junit.internal.runners.statements.RunBefore.evaluate(RunBefore.java:26)
at org.junit.internal.runners.statements.RunAfter.evaluate(RunAfter.java:325)
at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)
at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)
at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:299)
at org.junit.runners.ParentRunner\$3.schedule(ParentRunner.java:71)
at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
at org.junit.runners.ParentRunner.access\$000(ParentRunner.java:268)
at org.junit.runners.ParentRunner\$2.evaluate(ParentRunner.java:268)
at org.junit.runners.ParentRunner.run(ParentRunner.java:363)
at org.junit.runners.Suite.runChild(Suite.java:128)
at org.junit.runners.Suite.runChild(Suite.java:27)
at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:299)
at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:71)
at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
at org.junit.runners.ParentRunner.access\$000(ParentRunner.java:58)
at org.junit.runners.ParentRunner\$2.evaluate(ParentRunner.java:268)
at org.junit.runners.ParentRunner.run(ParentRunner.java:363)
at org.junit.runner.JUnitCore.run(JUnitCore.java:137)
at org.junit.runner.JUnitCore.run(JUnitCore.java:115)
at org.junit.runner.JUnitCore.runMain(JUnitCore.java:77)
at org.junit.runner.JUnitCore.main(JUnitCore.java:36)

FAILURES!!!
Tests run: 8, Failures: 1
Running MusicFormatTest
JUnit version 4.12
.....
Time: 0.004
OK (6 tests)

```

FOLDERS
└ Da Shop
  └ behaviours
  └ bin
  └ music_shop
    └ Customer.java
    └ MusicFormat.java
    └ MusicPlayer.java
    └ MusicType.java
    └ PaymentType.java
    └ PlayerType.java
    └ Shop.java
    └ StockItem.java
    └ Transactions.java
  └ music_shop_specs
    └ CustomerTest.java
    └ MusicFormatTest.java
    └ MusicPlayerTest.java
    └ ShopTest.java
    └ TransactionTest.java
  └ build.rb
  <> Da_Shop_project_definitions.md
  <> run_tests.rb

CustomerTest.java
28 cashAmount.put(PaymentType.CASH, 150);
29 creditCardAmount = new HashMap<PaymentType, Integer>();
30 creditCardAmount.put(PaymentType.CREDIT, 0);
31 debitCardAmount = new HashMap<PaymentType, Integer>();
32 debitCardAmount.put(PaymentType.DEBIT, 0);
33 jimWallet.add(CashAmount);
34 jimWallet.add(CreditCardAmount);
35 jimWallet.add(DebitCardAmount);
36 customer = new Customer("Jim", jimWallet);
37
38
39 @Test
40 public void hasName() {
41     assertEquals("Jim", customer.getName());
42 }
43
44 @Test
45 public void hasWallet() {
46     assertEquals(jimWallet, customer.getWallet());
47 }
48
49 @Test
50 public void hasPaymentType() {
51     assertEquals(cashAmount, customer.getPaymentType());
52 }
53
54 @Test
55 public void canAddItemToPurchases() {
56     customer.addItemToPurchases(cassette);
57     assertEquals("Sonic Death", customer.getPurchaseHistory().get(0));
58 }
59
60 @Test
61 public void canCountItemsInPurchase() {
62     customer.addItemToPurchases(cassette);
63     assertEquals(1, customer.countItems());
64 }
65
66 @Test
67 public void canGetAlbumFromPurchase() {
68     customer.addItemToPurchases(cassette);
69     assertEquals("Sonic Death", customer.getAlbumFromPurchase());
70 }
71
72 @Test

```

Da Shop git:(master) ✘ ruby build.rb
building behaviours
building Da Shop
building music_shop
building music_shop_specs
Da Shop git:(master) ✘ ruby run_tests.rb
Running CustomerTest
JUnit version 4.12
.....
Time: 0.007
OK (9 tests)
Running TransactionTest
JUnit version 4.12
.....
Time: 0.005
OK (9 tests)
Running ShopTest
JUnit version 4.12
.....
Time: 0.004
OK (6 tests)
Running MusicFormatTest
JUnit version 4.12
.....
Time: 0.004
OK (6 tests)
Running MusicPlayerTest
JUnit version 4.12
...
Time: 0.005
OK (3 tests)
Running ShopTest
JUnit version 4.12
.....
Time: 0.004
OK (9 tests)
Running TransactionTest
JUnit version 4.12
.....
Time: 0.005
OK (6 tests)