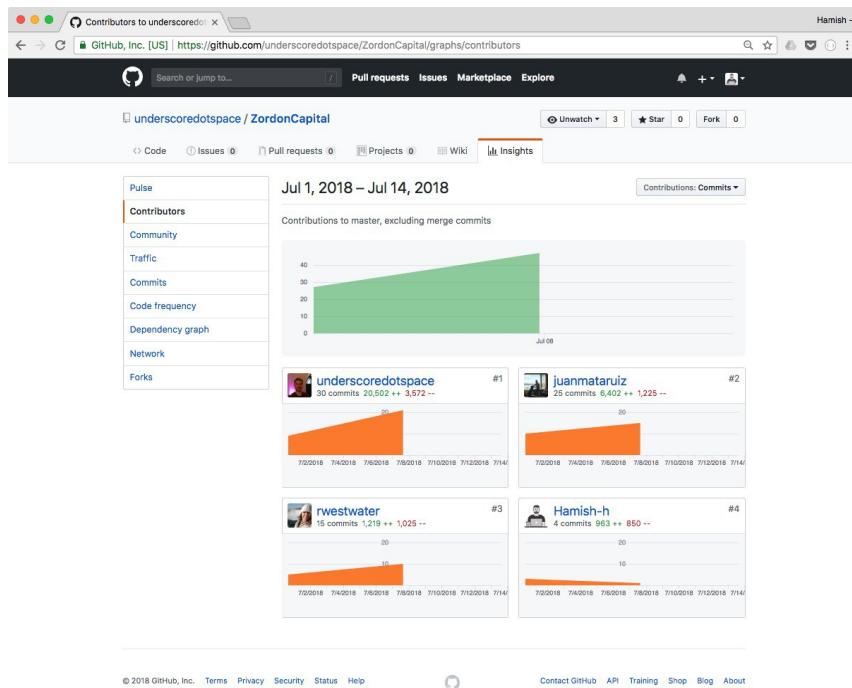


Evidence for Project Unit

Hamish Hoad
Cohort E-21

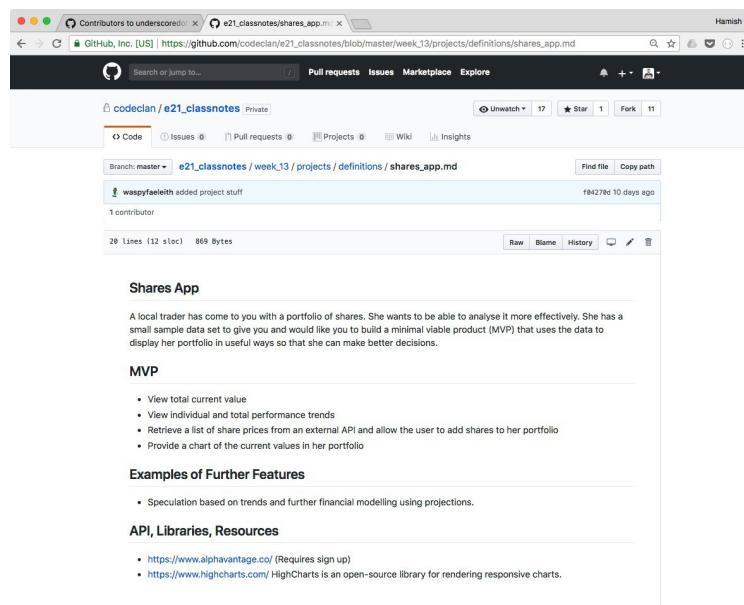
P. 1 Github Contributors page



Evidence for unit

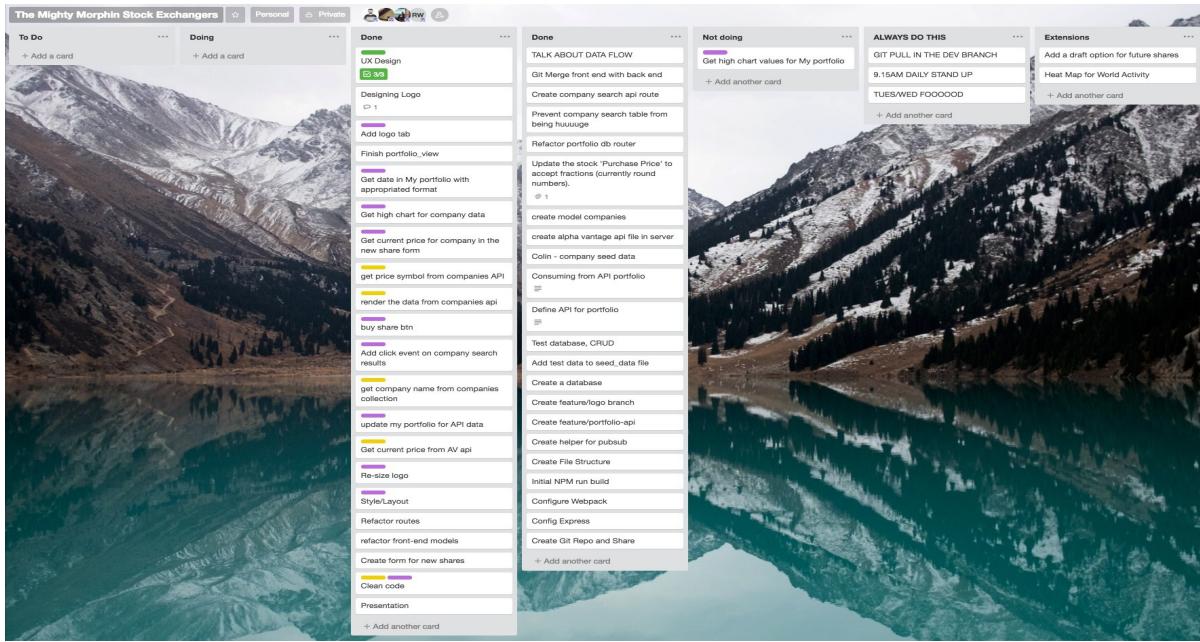
P. 2 Project Brief

Evidence for unit



P. 3 Use of Trello

Evidence for unit



P. 4 Acceptance Criteria

Evidence for unit

Acceptance Criteria

User Story: To display and analyse a portfolio of shares in useful ways to make better decisions.

Criteria:

1. The user should be able to see a portfolio list of purchased shares with the number purchased, the date of purchase, the price paid, the current valuation and the profit and losses that have resulted.
2. The user should be able to search for new shares and see the historical share price with its profits and losses to assess the viability for a purchase.
3. The user should be able to purchase shares which are added to the portfolio.
4. The user should not be able to purchase shares unless mandatory fields have been completed and a buy process initiated.

Acceptance Testing Plan

Process

1. Open the application.

An active news feed (ticker) is displayed containing; a link to each news feed.

'My Portfolio' of shares is displayed containing the;
Company symbol
Company name
Date purchased
Volume purchased
Price bought

*Current price
Profit or Loss
Profit/loss indicator
Total bought
Total current price
Total profit/loss
Totals profit/loss indicator.*

*A search option is displayed containing;
an input box
a search button.*

*A buy box is displayed containing;
Symbol
Company Name
Purchase Price
Volume
Buy Button*

2. Enter a full/part company name in the input box.

3. Click the search button.

A list of search results should be displayed.

4. Select an option from the list.

*The purchase box is then be populated with;
Symbol
Company Name
Purchase Price.*

*The graph is then be populated containing;
The Company Name
Close Price (axis)
Date (axis)
Graph line.*

5. Move / hover the cursor over points along the graph line

*A popup box indicates the
Date
Price of the share.*

6. Enter a valid number in the Volume input box

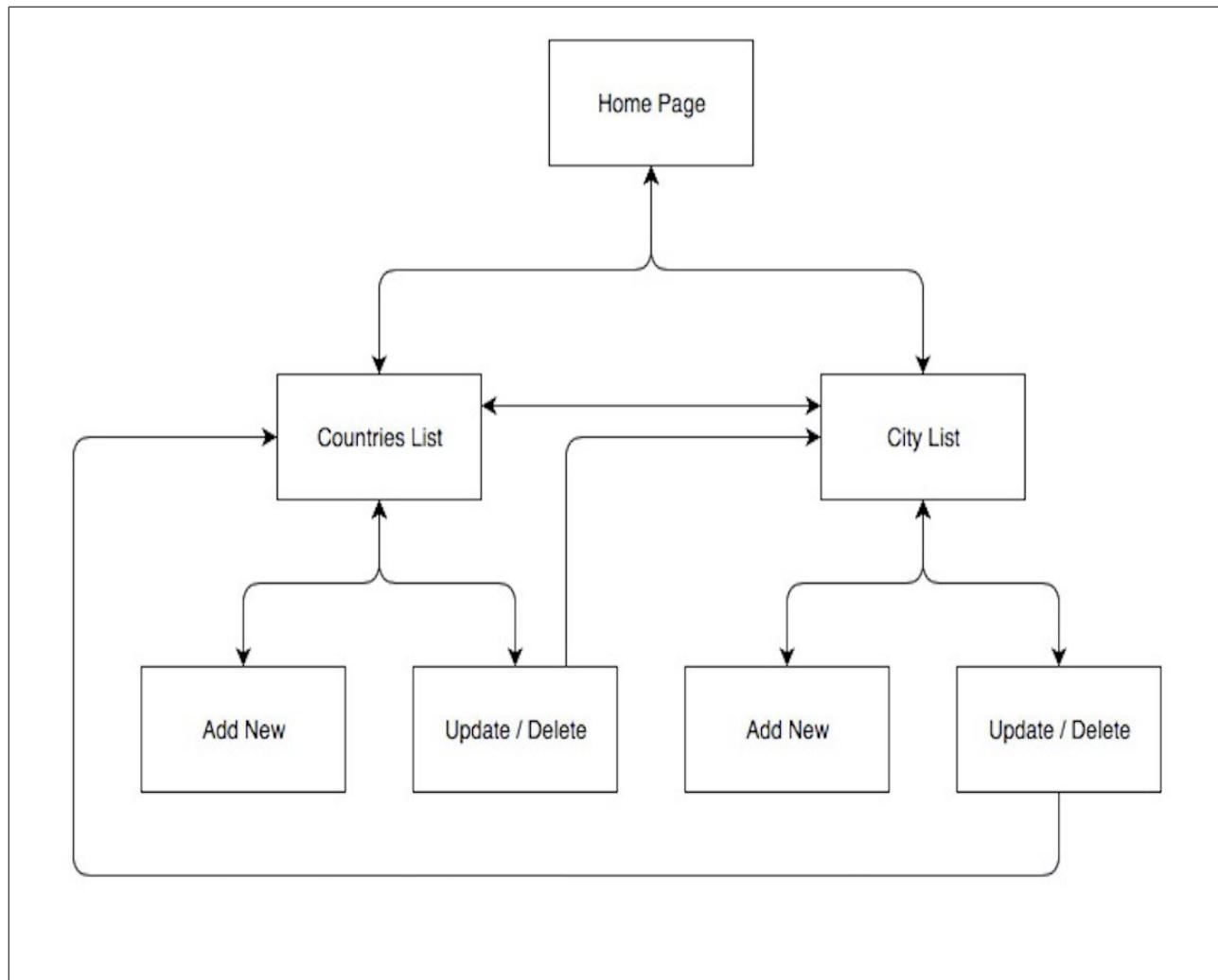
7. Click the buy button.

*The graph will clear of information / be hidden
The buy box will clear of input data
The screen will (or may need to) refresh
The purchase will appear on the screens portfolio list.*

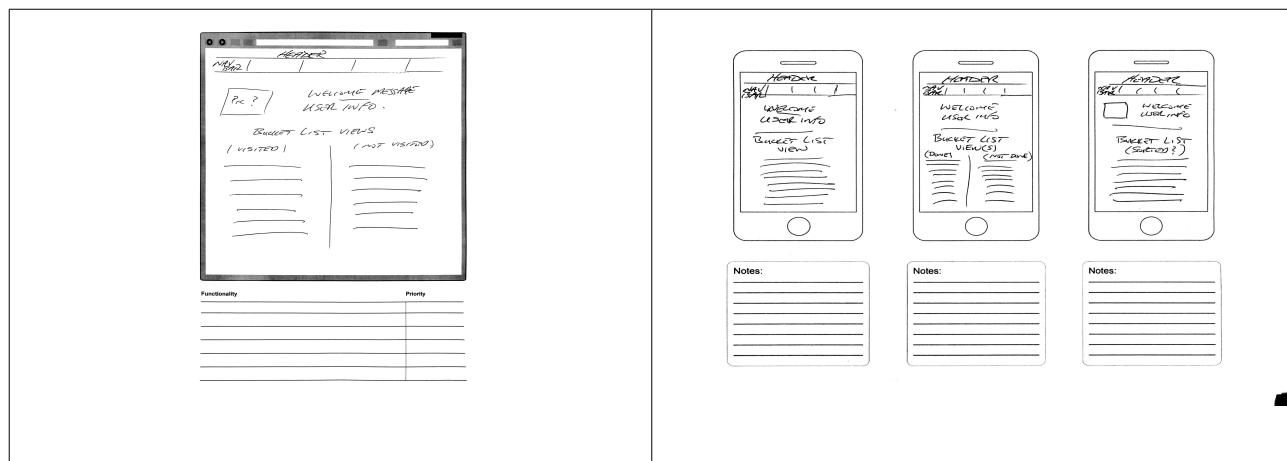
8. Refresh the browser.

All expected results will be displayed.

P. 5 User sitemap

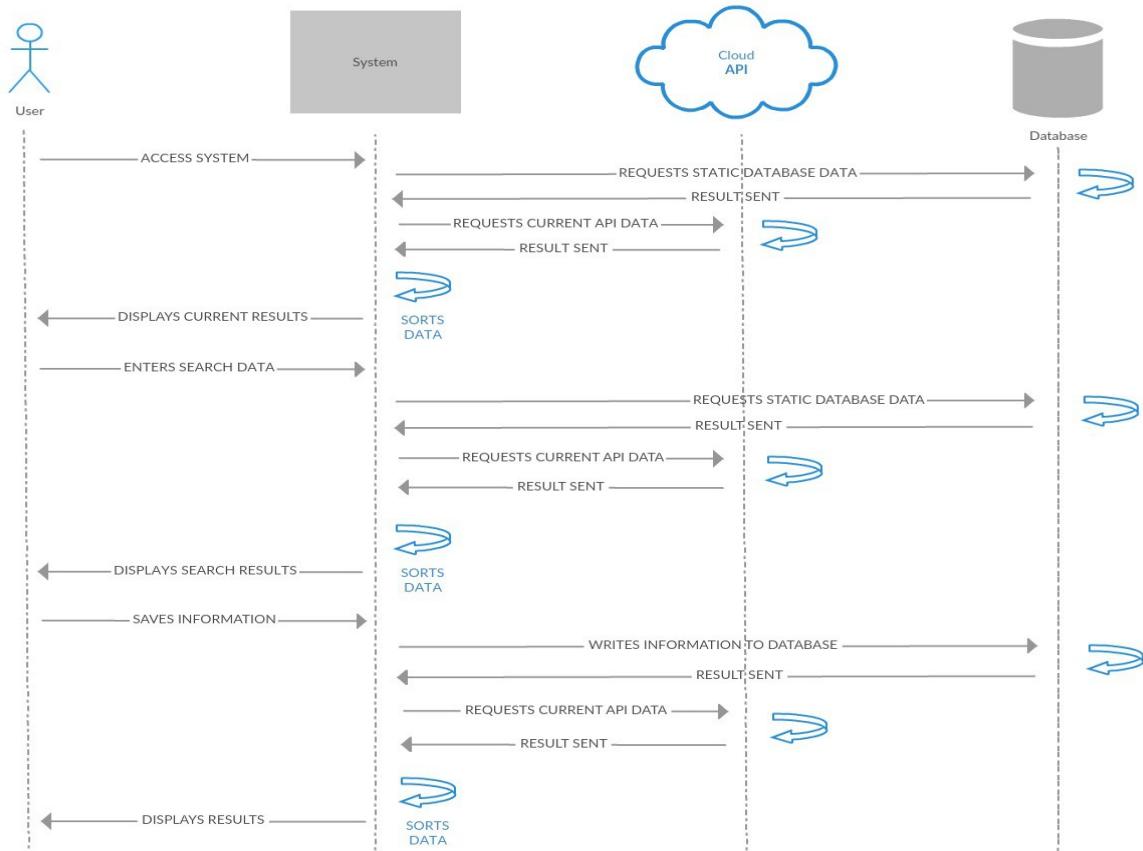


P. 6 Wireframes designs

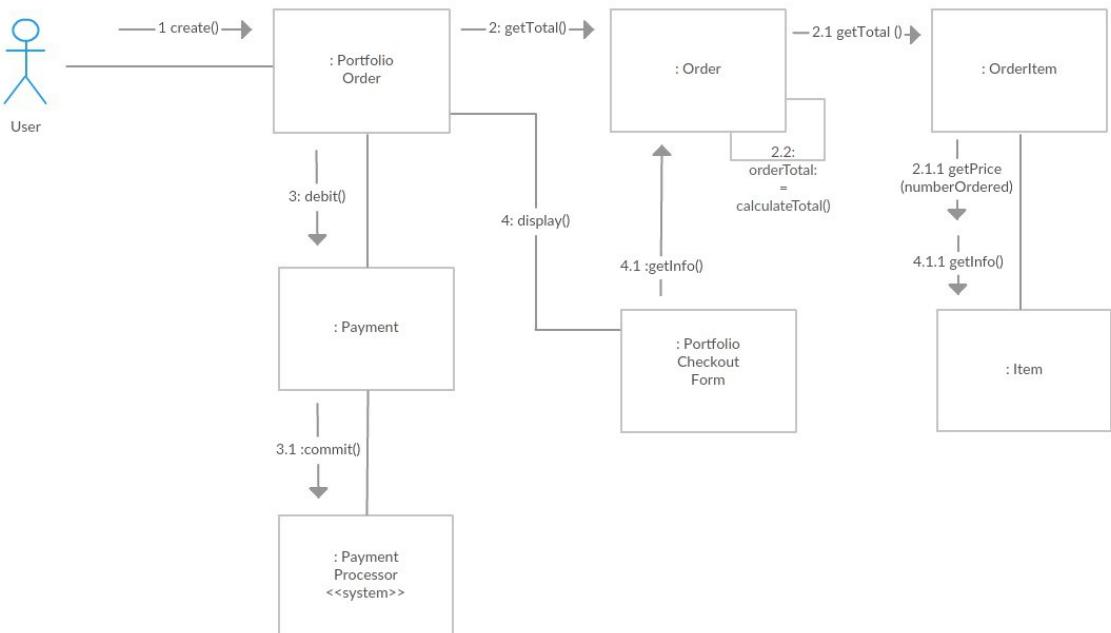


P. 7 System interactions diagrams

Sequence diagram

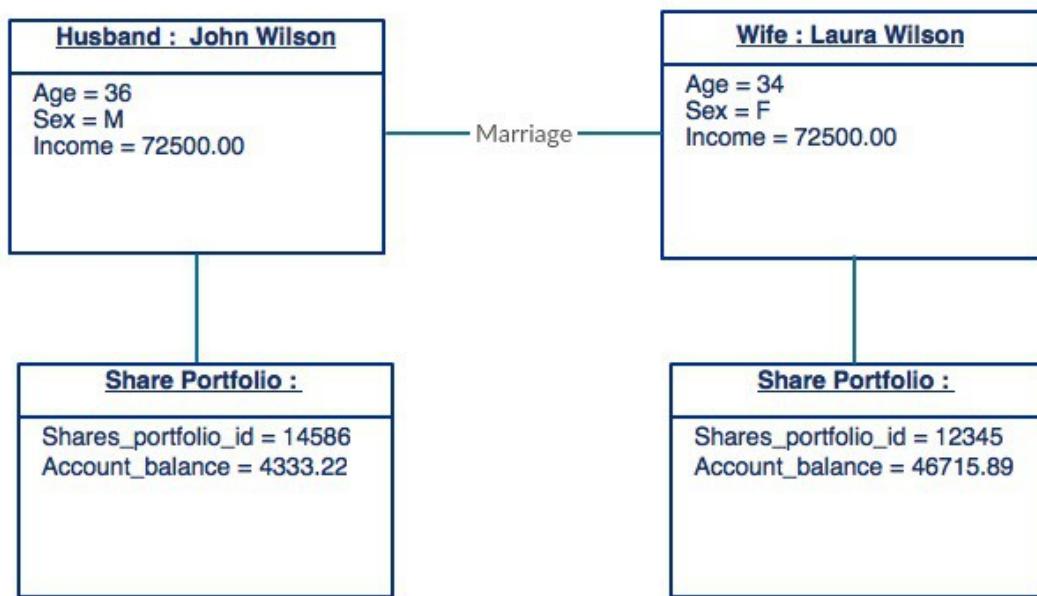
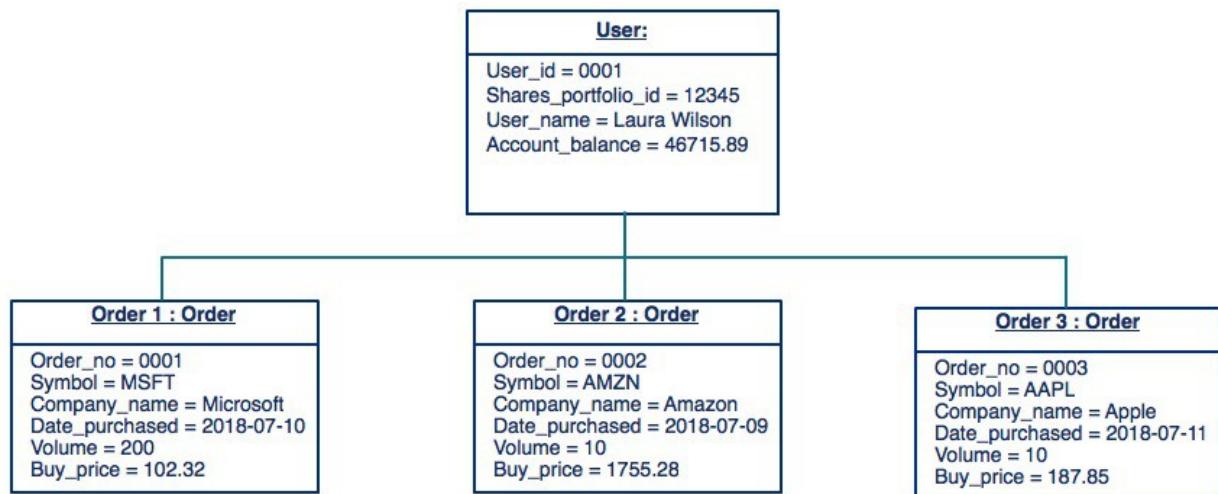


Collaboration diagram



P. 8 Two Object Diagrams

Evidence for unit Evidence for unit

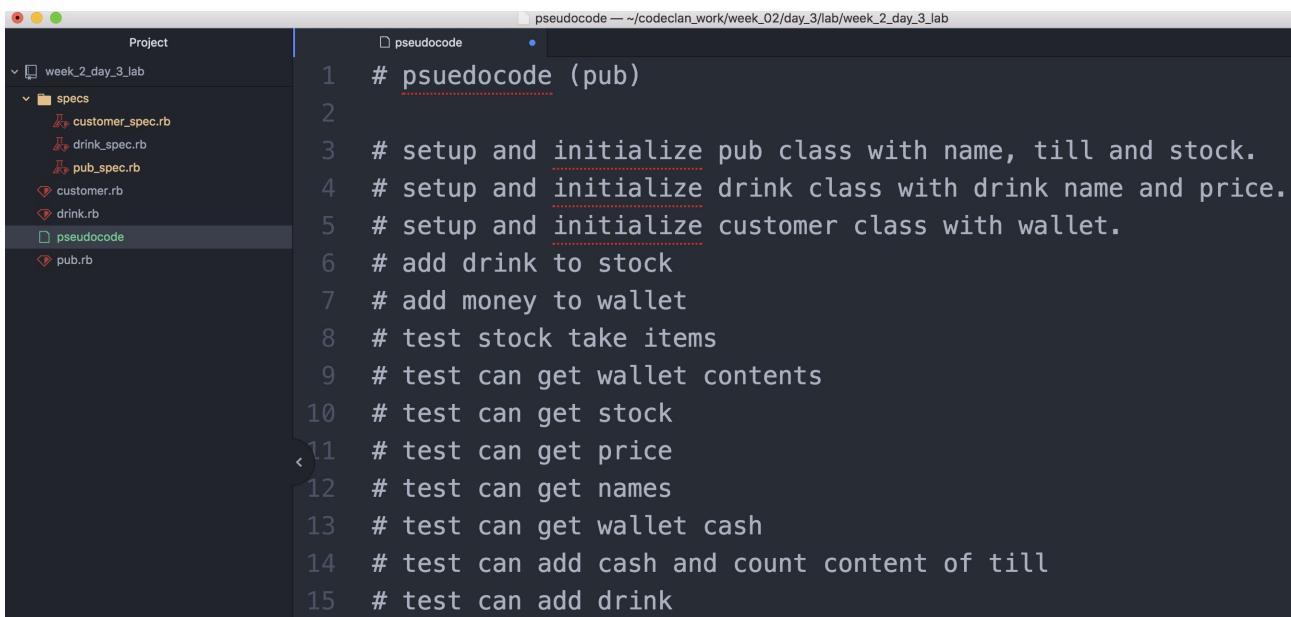


P. 9 Choice of two algorithms (find the algorithms on a program you might have written, show the code you have used.)

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

P. 10 Example of Pseudocode

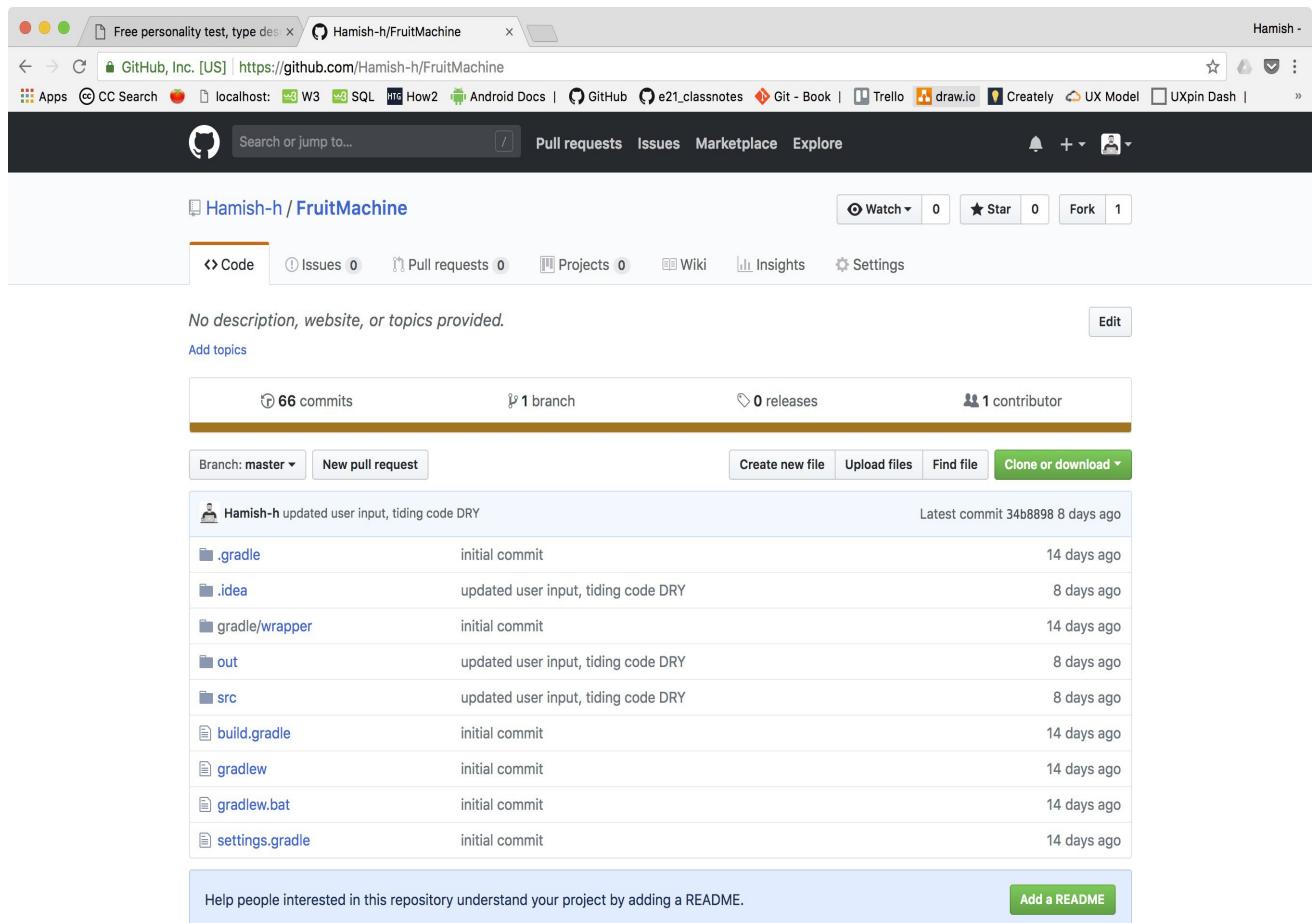
Evidence for unit



```
# pseudocode (pub)
# setup and initialize pub class with name, till and stock.
# setup and initialize drink class with drink name and price.
# setup and initialize customer class with wallet.
# add drink to stock
# add money to wallet
# test stock take items
# test can get wallet contents
# test can get stock
# test can get price
# test can get names
# test can get wallet cash
# test can add cash and count content of till
# test can add drink
```

P. 11 Github link to one of your projects

<https://github.com/Hamish-h/FruitMachine>



The screenshot shows a GitHub repository page for 'Hamish-h/FruitMachine'. The repository has 66 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was 34b8898, 8 days ago. The repository contains files like .gradle, .idea, gradle/wrapper, out, src, build.gradle, gradlew, gradlew.bat, and settings.gradle. A message at the bottom encourages adding a README.

No description, website, or topics provided.

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

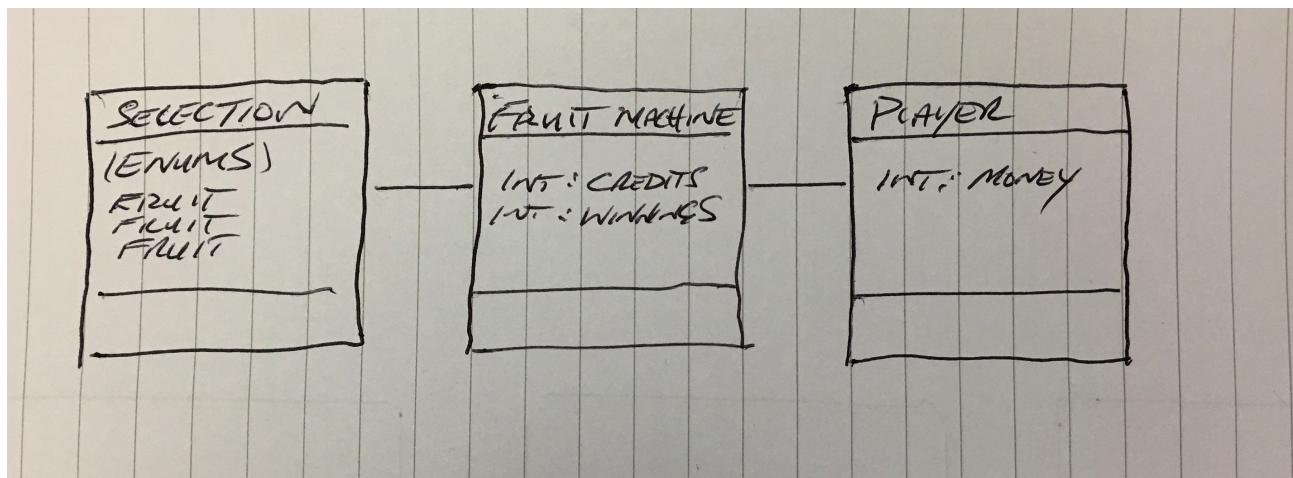
File	Commit Message	Date
.gradle	initial commit	14 days ago
.idea	updated user input, tiding code DRY	8 days ago
gradle/wrapper	initial commit	14 days ago
out	updated user input, tiding code DRY	8 days ago
src	updated user input, tiding code DRY	8 days ago
build.gradle	initial commit	14 days ago
gradlew	initial commit	14 days ago
gradlew.bat	initial commit	14 days ago
settings.gradle	initial commit	14 days ago

Help people interested in this repository understand your project by adding a README. Add a README

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

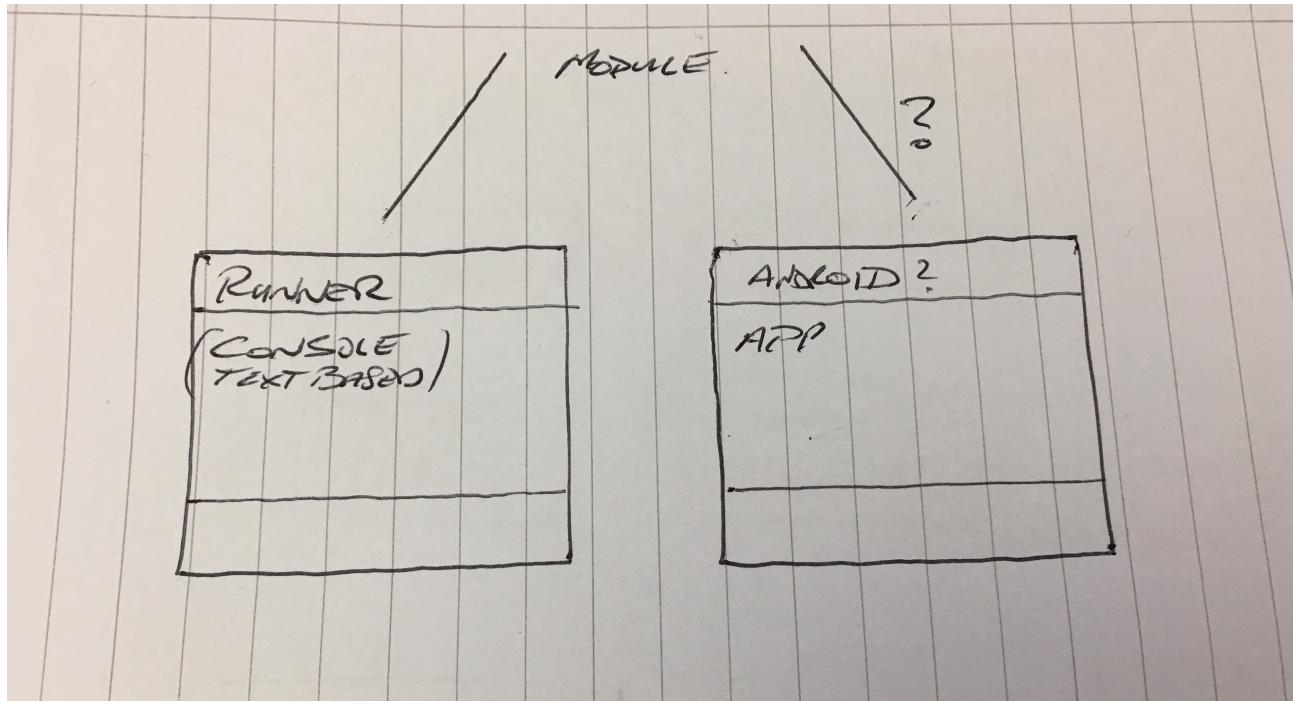
Planning and development – fruit machine

Rough plan

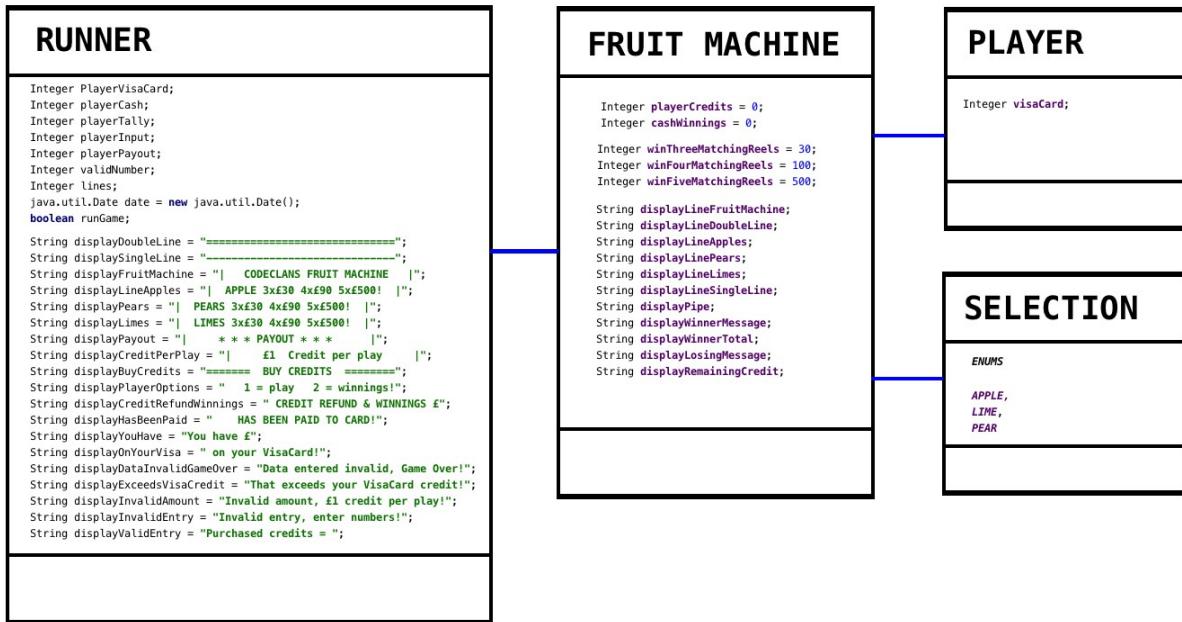


Rough plan for the control function

Java runner and Andriod app.



Detailed plan



Coding steps

The screenshot shows the Java code for the `Runner` class in an IDE. The code is identical to the one shown in the Detailed Plan diagram. The output window shows the game's interface and a sample run:

```

=====
| CODECLANS FRUIT MACHINE |
| APPLE 3xE30 4xE90 5xE500! |
| PEARS 3xE30 4xE90 5xE500! |
| LIMES 3xE30 4xE90 5xE500! |

[Pear|Apple|Pear|Lime|Apple]
=====

Unlucky this time, try again?

WINNINGS TOTAL £0

Your remaining credit is £59

You have £40 on your VisaCard!

1 = play 2 = winnings!

```

Coding Steps

Coding Steps

```

// WINNING LINES

// winning line of five apples £500
if ((reelOne == Selection.APPLE) && (reelTwo == Selection.APPLE) && (reelThree == Selection.APPLE) && (reelFour == Selection.APPLE) && (reelFive == Selection.APPLE)){ cashWinnings = (cashWinnings + winFiveMatchingReels);
    System.out.println(displayPipe + reelOne + displayPipe + reelTwo + displayPipe + reelThree + displayPipe + reelFour + displayPipe + reelFive + displayPipe + "\n" + displayLineDoubleLine + "\n"
        + displayWinnerMessage + "\n" + displayLineSingleLine + "\n" + displayWinnerTotal + cashWinnings + "\n" + displayLineSingleLine + "\n" + displayRemainingCredit + playerCredits);

// winning lines of four apples £100
else if ((reelOne == Selection.APPLE) && (reelTwo == Selection.APPLE) && (reelThree == Selection.APPLE) && (reelFour == Selection.APPLE)) { cashWinnings = (cashWinnings + 100);
    System.out.println(displayPipe + reelOne + displayPipe + reelTwo + displayPipe + reelThree + displayPipe + reelFour + displayPipe + reelFive + displayPipe + "\n" + displayLineDoubleLine + "\n"
        + displayWinnerMessage + "\n" + displayLineSingleLine + "\n" + displayWinnerTotal + cashWinnings + "\n" + displayLineSingleLine + "\n" + displayRemainingCredit + playerCredits);
}

// winning lines of four apples £100
else if ((reelTwo == Selection.APPLE) && (reelThree == Selection.APPLE) && (reelFour == Selection.APPLE) && (reelFive == Selection.APPLE)) { cashWinnings = (cashWinnings + 100);
    System.out.println(displayPipe + reelOne + displayPipe + reelTwo + displayPipe + reelThree + displayPipe + reelFour + displayPipe + reelFive + displayPipe + "\n" + displayLineDoubleLine + "\n"
        + displayWinnerMessage + "\n" + displayLineSingleLine + "\n" + displayWinnerTotal + cashWinnings + "\n" + displayLineSingleLine + "\n" + displayRemainingCredit + playerCredits);
}

// winning lines of three apples £30
else if ((reelOne == Selection.APPLE) && (reelTwo == Selection.APPLE) && (reelThree == Selection.APPLE)) { cashWinnings = (cashWinnings + winThreeMatchingReels);
    System.out.println(displayPipe + reelOne + displayPipe + reelTwo + displayPipe + reelThree + displayPipe + reelFour + displayPipe + reelFive + displayPipe + "\n" + displayLineDoubleLine + "\n"
        + displayWinnerMessage + "\n" + displayLineSingleLine + "\n" + displayWinnerTotal + cashWinnings + "\n" + displayLineSingleLine + "\n" + displayRemainingCredit + playerCredits);
}

// winning lines of three apples £30
else if ((reelTwo == Selection.APPLE) && (reelThree == Selection.APPLE) && (reelFour == Selection.APPLE)) { cashWinnings = (cashWinnings + winThreeMatchingReels);
    System.out.println(displayPipe + reelOne + displayPipe + reelTwo + displayPipe + reelThree + displayPipe + reelFour + displayPipe + reelFive + displayPipe + "\n" + displayLineDoubleLine + "\n"
        + displayWinnerMessage + "\n" + displayLineSingleLine + "\n" + displayWinnerTotal + cashWinnings + "\n" + displayLineSingleLine + "\n" + displayRemainingCredit + playerCredits);
}

// winning lines of three apples £30
else if ((reelThree == Selection.APPLE) && (reelFour == Selection.APPLE) && (reelFive == Selection.APPLE)) { cashWinnings = (cashWinnings + winThreeMatchingReels);
    System.out.println(displayPipe + reelOne + displayPipe + reelTwo + displayPipe + reelThree + displayPipe + reelFour + displayPipe + reelFive + displayPipe + "\n" + displayLineDoubleLine + "\n"
        + displayWinnerMessage + "\n" + displayLineSingleLine + "\n" + displayWinnerTotal + cashWinnings + "\n" + displayLineSingleLine + "\n" + displayRemainingCredit + playerCredits);
}

```

Output

Tests

```
=====
| CODECLAN FRUIT MACHINE |
| APPLE 3x£30 4x£90 5x£500 ! |
| PEARS 3x£30 4x£90 5x£500 ! |
| LIMES 3x£30 4x£90 5x£500 ! |
=====
|PEAR|APPLE|PEAR|LIME|APPLE|
=====
Unlucky this time, try again?
=====
WINNINGS TOTAL £0
=====
Your remaining credit is £59
=====
You have £40 on your VisaCard!
=====
1 = play   2 = winnings!
```

- ▶  **FruitMachineTest**
 - ✓ canGetWinThreeMatchingReels
 - ✓ canGetCashWinnings
 - ✓ CanGameSpin
 - ✓ canGetWinFiveMatchingReels
 - ✓ CanGetAnyRandomAnswerReelFive
 - ✓ CanGetAnyRandomAnswerReelFour
 - ✓ canGetPlayerCredits
 - ✓ CanGetAnyRandomAnswerReelOne
 - ✓ CanGetAnyRandomAnswerReelTwo
 - ✓ canGetWinFourMatchingReels
 - ✓ CanGetAnyRandomAnswerReelThree

P. 13 User input

User inputs and edits a City, adding places to visit.

The screenshot shows the project structure and the code for the edit view of a city. The code includes forms for updating the city's name and review, and for deleting it. The browser displays the 'Edit City' page for Edinburgh, showing a review about visiting Edinburgh Castle and the Royal Mile. The terminal shows the server logs for the session.

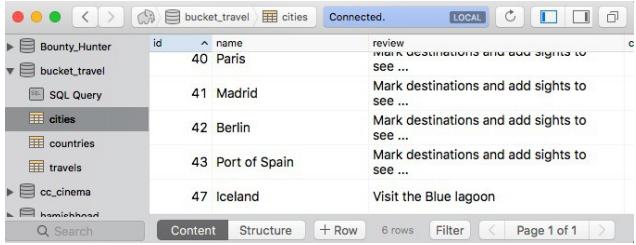
P. 14 Interaction with data persistence

Make sure you show the input being added.

User inputs a City, adds places to visit and saves the information which is then added to the database as shown.

The screenshot shows the project structure and the code for the edit view of a city. The code includes forms for updating the city's name and review, and for deleting it. The browser displays the 'Edit City' page for Edinburgh, showing a review about visiting Edinburgh Castle and the Royal Mile. The database table 'cities' is shown in the bottom left, containing rows for Edinburgh, Paris, Madrid, and Berlin. The terminal shows the server logs for the session.

P. 15 User output result

 <p>User inputs information and clicks add city which adds information to the database.</p>	 <p>The system then returns a list of cities showing the new city has been added to the database.</p>
 <p>The user can then update (or delete) city information, such as a place to visit.</p>	 <p>The information is then added to the database as shown in the above tables.</p>

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

The code that uses / implements the API.

```

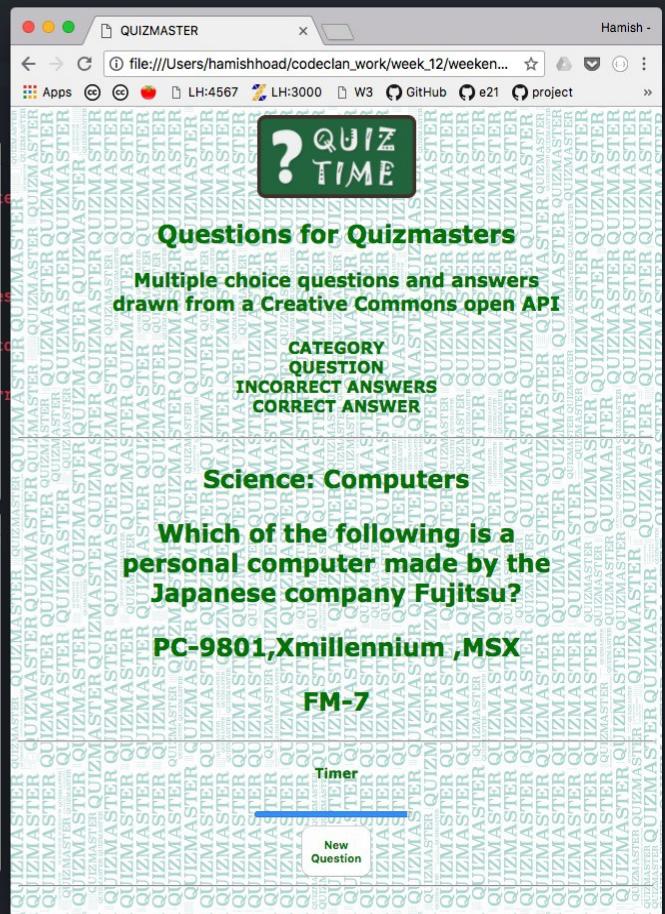
1  const Request = require('../helpers/request_helper.js');
2  const PubSub = require('../helpers/pub_sub.js');
3
4  const QuizMaster = function () {
5    // null or "" empty string
6    this.text = null;
7  }
8  // get data
9  QuizMaster.prototype.getData = function () {
10    // set the url
11    const request = new Request('https://opentdb.com/api.php?amount=1&difficulty=medium&type=multiple');
12    // oncomplete
13    request.get((data) => {
14      // console.log("request get results ", data);
15      // collect and assign data
16      this.category = data.results;
17      this.question = data.results;
18      this.incorrect_answers = data.results;
19      this.correct_answer = data.results;
20      // console.log("this.category", this.category);
21      // publish category
22      PubSub.publish('QuizMaster:quizMaster-loaded', this.category[0].category);
23      console.log(this.category[0].category);
24      console.log(this.question[0].question);
25      // publish
26      PubSub.publish('QuizMaster:quizMaster-loaded', this.question[0].question);
27      // publish incorrect answers
28      PubSub.publish('QuizMaster:quizMaster-loaded', this.incorrect_answers[0].incorrect_answers);
29      // publish correct answer
30      PubSub.publish('QuizMaster:quizMaster-loaded', this.correct_answer[0].correct_answer);
31    });
32  }
33  // export
34  module.exports = QuizMaster;
35

```

The API being used by the program while running.

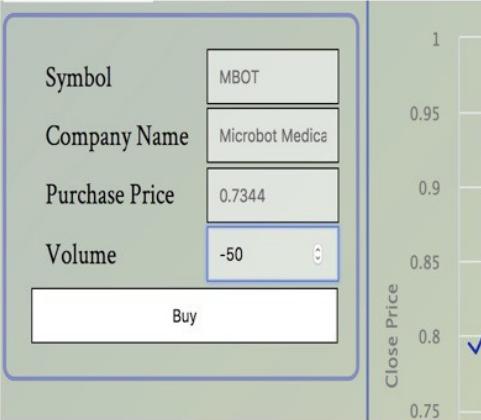
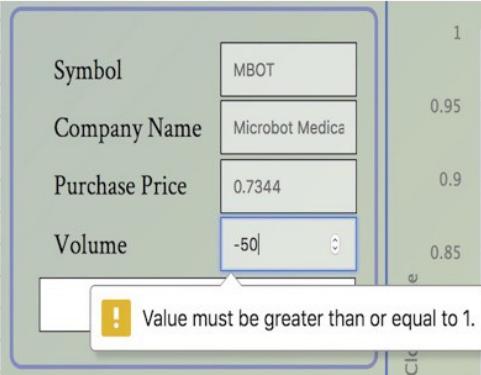
```
1 const Request = require('../helpers/request_helper.js');
2 const PubSub = require('../helpers/pub_sub.js');
3
4 const QuizMaster = function () {
5   // null or "" empty string
6   this.text = null;
7 }
8 // get data
9 QuizMaster.prototype.getData = function () {
10   // set the url
11   const request = new Request('https://opentdb.com/api.php?amount=1&difficulty=medium&type=multiple');
12   // oncomplete
13   request.get((data) => {
14     // console.log("request get results ", data);
15     // collect and assign data
16     this.category = data.results;
17     this.question = data.results;
18     this.incorrect_answers = data.results;
19     this.correct_answer = data.results;
20
21   npm run build — npm — node -n pm
22 [.src/models/quizmaster.js] 1.21 KiB {main} [built]
23 [.src/views/quizmaster_view.js] 823 bytes {main} [built]
24
25 ➜ repo git:(master) ✘ npm run build
26 > repo@1.0.0 build /Users/hamishhood/codeclan_work/week_12/weekend_homework/repo
27 > webpack -w
28
29 Webpack is watching the files...
30
31 Hash: f3d7f194be2d3fb4434
32 Version: webpack 4.13.0
33 Time: 88ms
34 Built at: 16/07/2018 00:34:48
35 Asset Size Chunks Names
36 bundle.js 8.62 KiB main [emitted] main
37 [.src/app.js] 455 bytes {main} [built]
38 [.src/helpers/pub_sub.js] 301 bytes {main} [built]
39 [.src/helpers/request_helper.js] 397 bytes {main} [built]
40 [.src/models/quizmaster.js] 1.21 KiB {main} [built]
41 [.src/views/quizmaster_view.js] 823 bytes {main} [built]
```

```
mongod — mongod — mongod
te access to data and configuration is unrestricted.
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten]
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten] ** WARNING: This server
is bound to localhost.
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten] ** Remote syste
ms will be unable to connect to this server.
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten] ** Start the se
rver with --bind_ip <address> to specify which IP
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten] ** addresses it
should serve responses from, or with --bind_ip_all to
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten] ** bind to all
interfaces. If this behavior is desired, start the se
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten] ** rver with
--bind_ip 127.0.0.1 to disable this warning.
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten]
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten]
2018-07-16T00:34:43.552+0100 I CONTROL [initandlisten] ** WARNING: soft rlimits
too low. Number of files is 256, should be at least 1000
2018-07-16T00:34:43.569+0100 I FTDC [initandlisten] Initializing full-time d
iagnostic data capture with directory '/data/db/diagnostic.data'
2018-07-16T00:34:43.569+0100 I NETWORK [initandlisten] waiting for connections
on port 27017
```



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

Evidence for unit

Bug ID	Category	Label	Value
	ID number	#001	
	Name	Portfolio: Volume of shares bought	
	Reporter	Project team	
Submit Date		07/11/18	
Summary		The share purchase option allows negative numbers to be entered i.e., buy -25 shares	
URL		local host:	
		 <p>The screenshot shows a user interface for buying shares. It includes fields for Symbol (MBOT), Company Name (Microbot Medica), Purchase Price (0.7344), and Volume (-50). A 'Buy' button is at the bottom. To the right is a vertical bar chart titled 'Close Price' with values from 0.75 to 1.0. A blue checkmark is visible near the 0.8 mark.</p>	
Bug overview	Screenshot		
	Platform	Macintosh	
	Operating System	OS High Sierra 10.13.6	
Environment	Browser	Chrome Version 67.0.3396.99 (Official Build) (64-bit)	
	Steps to reproduce	search for shares > select share from results > add negative purchase volume > click "buy" button (see screenshot) > refresh screen > negative purchase is shown	
	Expected result	The buy option should reject negative numbers	
	Actual result	The buy option accepts negative numbers	
Bug details	Description	Restrict the buy option to only accept valid input of a number equal to or greater than 1	
	Severity	Major	
	Assigned to	Project team (Juan)	
Bug tracking	Priority	Medium	
Notes	Notes	bug corrected. The buy option now rejects negative numbers and only accepts valid input	
		 <p>The screenshot shows the same share purchase interface as before, but now it displays an error message: 'Value must be greater than or equal to 1.' The volume field still contains '-50'.</p>	

P. 18 Testing your program

Show the test code, the test not passing.....and then the test fixed.

Example of test code

with the test failing to pass

NOT Passing

```
Project
```

```
card.rb
```

```
1
2 class Card
3   attr_reader :suit, :value
4
5   def initialize(suit, value)
6     @suit = suit
7     @value = value;
8   end
9 end
```

```
testing_task_2.rb
```

```
6 require_relative('card.rb')
7 class CardGame
8
9   def checkforAce(card)
10    if card.value == 1
11      return true
12    else
13      return false
14    end
15  end
16
17  def highest_card(card1, card2)
18    if card1.value > card2.value
19      return card.name
20    else
21      card2
22    end
23
24
25  def self.cards_total(cards)
26    total
27    for card in cards
28      total += card.value
29    end
30    return "You have a total of" + total
31  end
```

```
testing_task_2_spec.rb
```

```
7
8 class CardGameTest < MiniTest::Test
9
10 def setup()
11   @card = CardGame.new("Clubs", 1)
12 end
13
14 def test_check_for_ace
15   assert_equal("Clubs", @card.value)
16 end
17
18 end
```

```
hamishnair@Hamishis-MBP:~/ynamic_Task_A--> rake
```

```
from specs/testing_task_2_spec.rb:in `require_relative'
```

```
rake aborted!
```

```
POM_Static_and_Dynamic_Task_A_X_Ruby spec:testing_task_2_spec.rb
```

```
task :rake, :type => :rake do |t|
  system("rake")
end
```

```
spec:testing_task_2_spec.rb:in `eval': undefined local variable or method `@card' for main:Object (NameError)
```

```
spec:testing_task_2_spec.rb:in `block in <main>'
```

```
spec:testing_task_2_spec.rb:in `each'
```

```
spec:testing_task_2_spec.rb:in `eval'
```

```
spec:testing_task_2_spec.rb:in `require_relative': /Users/hamishnair/ynamic_Task_A--> rake
```

```
POM_Static_and_Dynamic_Task_A_X_Ruby spec:testing_task_2_spec.rb
```

```
spec:testing_task_2_spec.rb:in `eval': undefined local variable or method `@card' for main:Object (NameError)
```

```
spec:testing_task_2_spec.rb:in `block in <main>'
```

```
spec:testing_task_2_spec.rb:in `each'
```

```
spec:testing_task_2_spec.rb:in `eval'
```

```
spec:testing_task_2_spec.rb:in `require_relative': /Users/hamishnair/ynamic_Task_A--> rake
```

```
POM_Static_and_Dynamic_Task_A_X_Ruby spec:testing_task_2_spec.rb
```

```
spec:testing_task_2_spec.rb:in `eval': unexpected end-of-input, expecting keyword_end (SyntaxError)
```

```
spec:testing_task_2_spec.rb:in `block in <main>'
```

```
spec:testing_task_2_spec.rb:in `each'
```

```
spec:testing_task_2_spec.rb:in `eval'
```

```
spec:testing_task_2_spec.rb:in `require_relative': /Users/hamishnair/ynamic_Task_A--> rake
```

```
POM_Static_and_Dynamic_Task_A_X_Ruby spec:testing_task_2_spec.rb
```

Example of test code once errors have been corrected

with the test then passing, fixed

Passing

```
Project
```

```
card.rb
```

```
1
2 class Card
3   attr_reader :suit, :value
4
5   def initialize(suit, value)
6     @suit = suit
7     @value = value
8   end
9 end
```

```
testing_task_2.rb
```

```
5
6 require_relative('card.rb')
7 class CardGame
8
9   def check_for_ace(card)
10    if card.value == 1
11      return true
12    else
13      return false
14    end
15  end
16
17  def highest_card(card1, card2)
18    if card1.value > card2.value
19      return card.name
20    else
21      card2
22    end
23
24
25  def self.cards_total(cards)
26    total
27    for card in cards
28      total += card.value
29    end
30    return "You have a total of" + total
31  end
32
33 end
```

```
testing_task_2_spec.rb
```

```
1 #!/usr/bin/env ruby
2 require('minitest/autorun')
3 require('minitest/rg')
4
5 require_relative('../card.rb')
6 require_relative('../testing_task_2.rb')
7
8 class CardGameTest < MiniTest::Test
9
10 def setup()
11   @card1 = Card.new("Clubs", 1)
12   @card2 = Card.new("Clubs", 2)
13 end
14
15 def test_check_for_ace
16   assert_equal(1, @card1.value())
17 end
18
19 end
```

```
hamishnair@Hamishis-MBP:~/ynamic_Task_A--> rake
```

```
Run options: --seed 68720
```

```
# Running:
```

```
Finishes in 0.000000s, 1600.1651 runs/s, 1600.1651 assertions/s.
```

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

```
+ POM_Static_and_Dynamic_Task_A_X_Ruby spec:testing_task_2_spec.rb
```