

Evidence for Project Unit

Brendan Prado

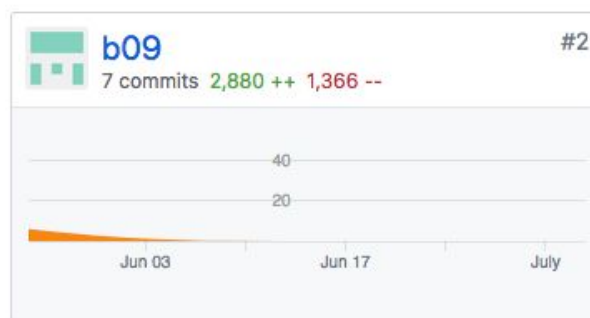
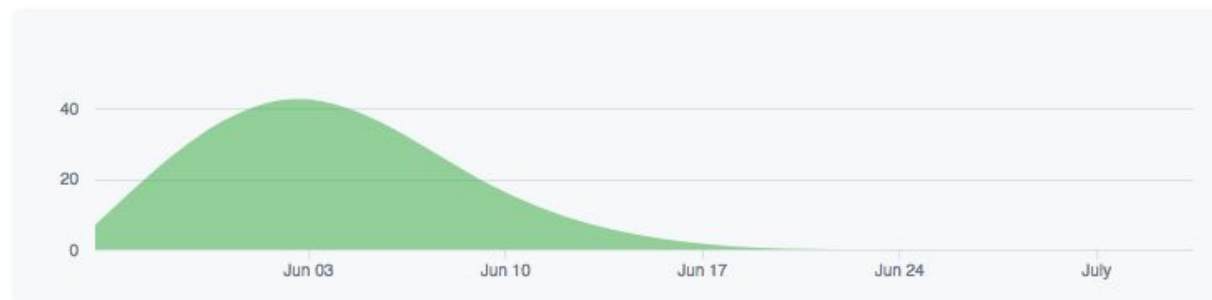
Cohort E20

P. 1 Github contributors page

May 27, 2018 – Jul 5, 2018

Contributions: **Commits** ▾

Contributions to master, excluding merge commits



P. 2 Project Brief

NASA International Space Station Tracker

Now deployed to Heroku at <https://nasa-iss-tracker.herokuapp.com/>

Project definition

An interactive educational app built with JavaScript using Node.js, webpack and Express for the server side. It uses positional data from the International Space Station (ISS) as a fun way for users to retrieve educational material about locations around the Earth! Created within a 4-person Agile team over the course of a week.

MVP

- Take user selection of a location and display a prediction of the next time the ISS will be above that location.
- A dynamic map showing the ISS current location related to city selection.

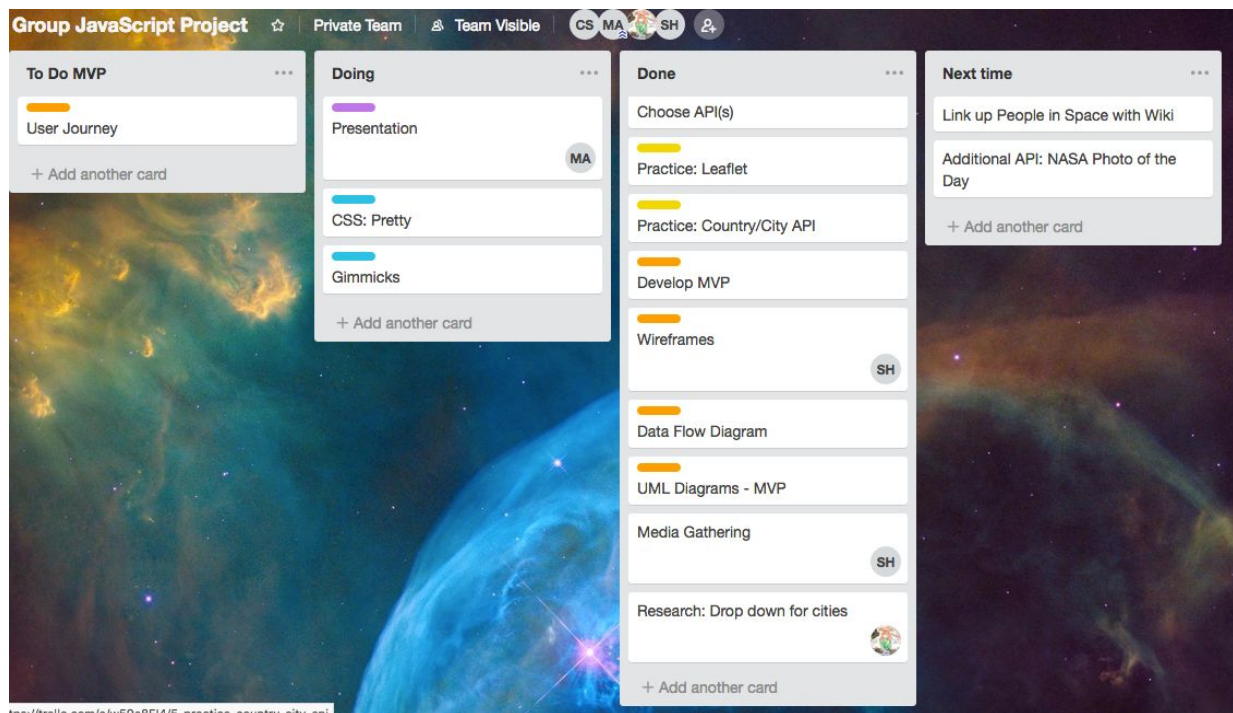
Extensions

- Take user selection of a location and display educational material relevant to that location.
- Use NASA media resources to provide relevant and appealing background content.

Data sources used

- [Open Notify API](#) for the ISS data,
- [Leaflet](#) for map rendering
- [Wikipedia API](#) for additional information about locations
- [Countries](#) and [cities](#) npm packages

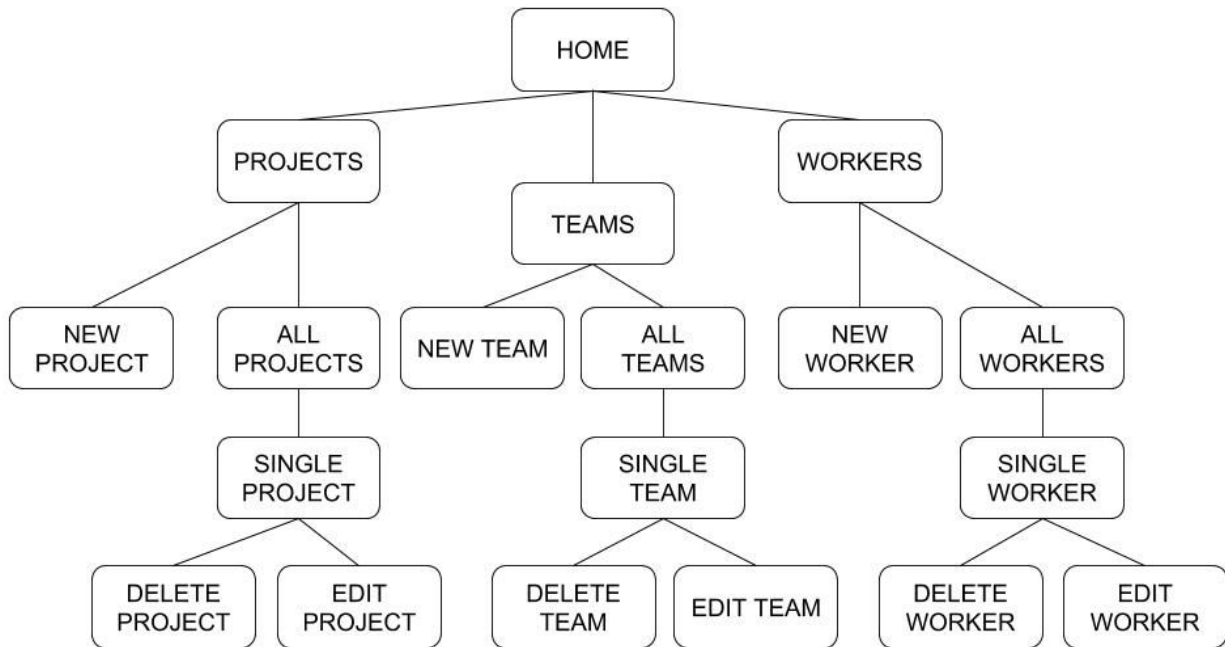
P. 3 Use of Trello



P. 4 Acceptance Criteria

Acceptance Criteria	Expected Result/Output	Pass/Fail
Marker on map should move every 5 seconds	App makes an API request every 5 seconds for current geo location of the ISS	PASS
User can select a city after input of first three characters of city name	Method searches through node module containing all world cities and checks for character match from user input, but only starting on the third inputted character	PASS
User can see list of people in space	App makes an API request on startup for current people in space	PASS
Upon selecting a city, user can see wiki articles relevant to city location	App makes an API request which is triggered when city name is selected and uses city coordinates to specify API request	PASS
Upon selecting a button, user can see wiki articles relevant to ISS location	App makes an API request which is triggered when a button is selected which grabs the current ISS coordinates and uses them to specify the API request	PASS
User can see message if no wiki articles are available	If an API request of a wiki article contains nothing, either because there are not articles for that location or some other reason, a message will appear warning user of no content for that location	PASS

P. 5 User Sitemap



P. 6 Wireframes designs (2)

HOMEPROJECTSWORKERSTEAMS

NEWDATEBUDGETTYPE

PROJECT	BUDGET	START	EMP. PR.	TYPE	MORE
NAME	\$\$\$	DATE	#	FRONT END	INFO
NAME	\$\$\$	DATE	#	FRONT END	INFO
NAME	\$\$\$	DATE	#	FRONT END	INFO
NAME	\$\$\$	DATE	#	FRONT END	INFO
NAME	\$\$\$	DATE	#	FRONT END	INFO
NAME	\$\$\$	DATE	#	FRONT END	INFO

Project Info:

[VIEW ALL](#)

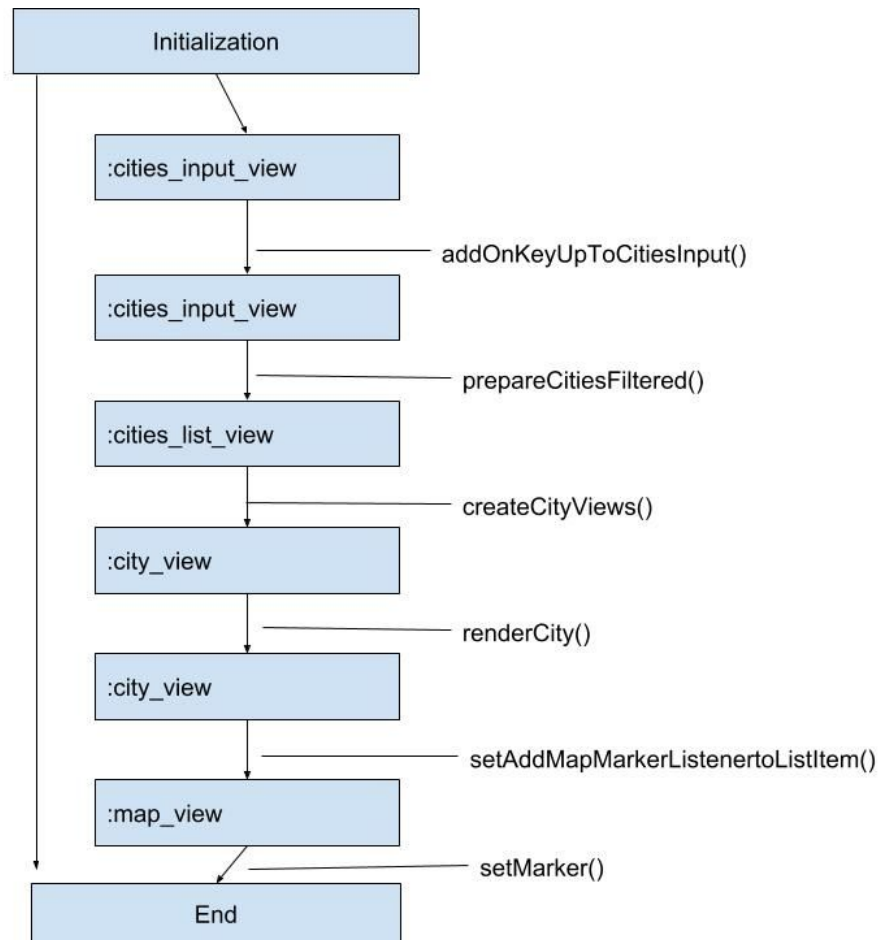
Name:	Name of Project
Budget:	\$10000
Date:	2016 - 03 - 05
Members:	First Lastname, First Lastname
Monthly Payroll:	\$100 amount of payroll based on salary of workers
Project Details:	Descript

[DELETE](#)

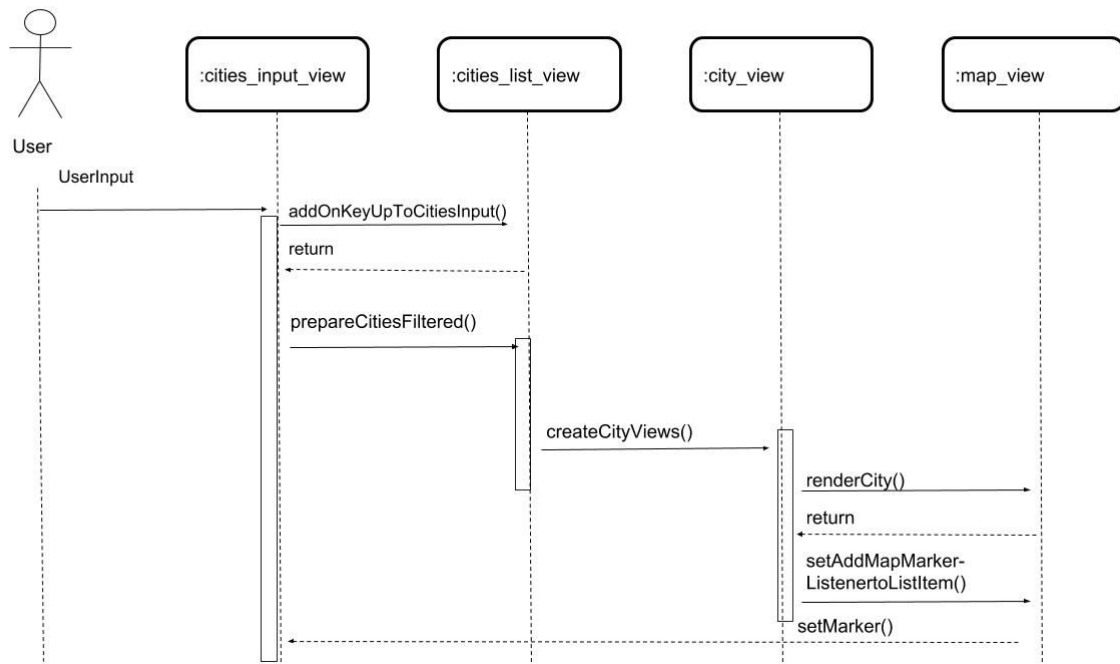
[EDIT](#)

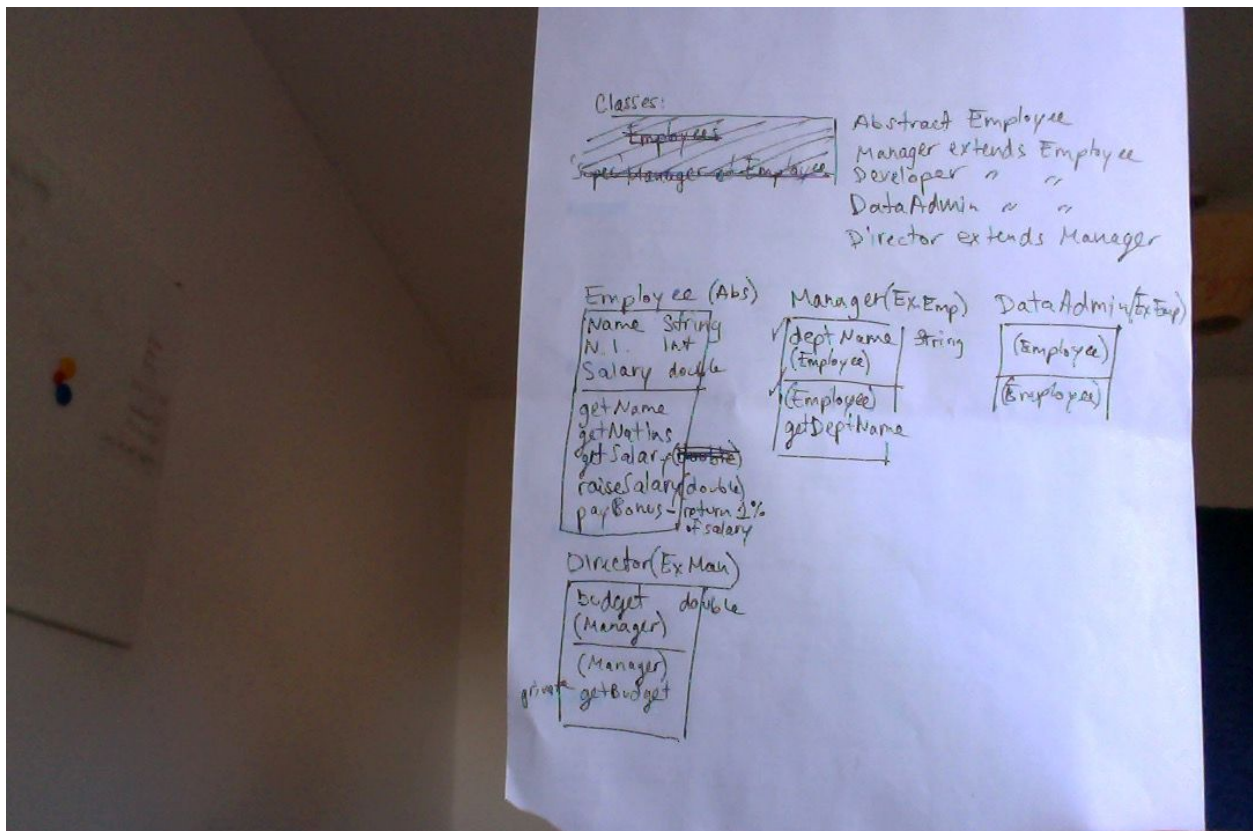
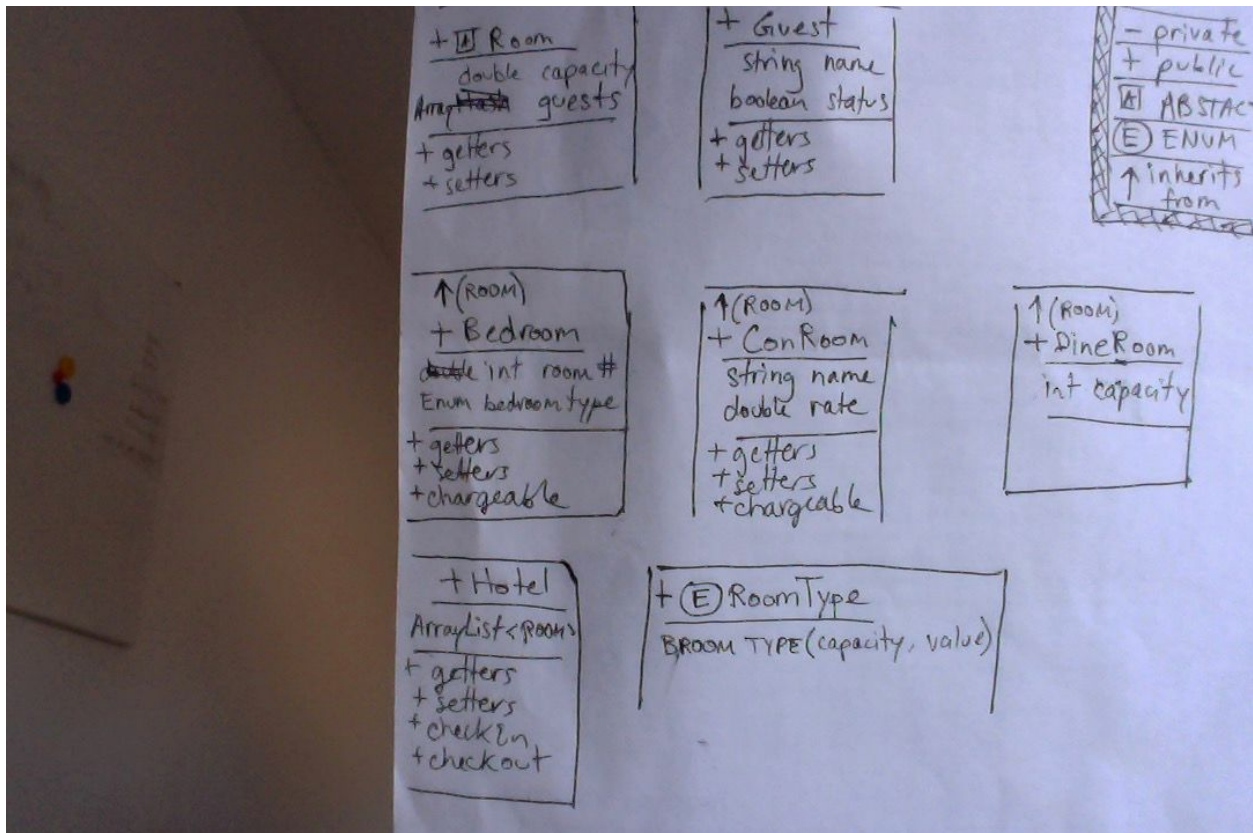
P. 7 System interaction diagrams (2)

Collaboration Diagram: input and select city



Interaction Diagram: input and select city





P. 9 Choice of two algorithms (find algorithms on a program you might have written, show the code you have used. (On this example take a screenshot and write what it is doing and why you decided to use it))

```

public static String morseToAlpha(String morse) {
    String[] morseWords = morse.split(" ");
    StringBuilder alphaWordOutput = new StringBuilder();
    for (String word : morseWords) {
        for (String letter : word.split(" ")) {
            for (MorseAlphabet alphaEnumRow : MorseAlphabet.values()) {
                if (alphaEnumRow.getMorse().equals(letter)) {
                    alphaWordOutput.append(alphaEnumRow.getAlphabet());
                }
            }
        }
        alphaWordOutput.append(" ");
    }
    return alphaWordOutput.toString();
}

```

The above algorithm takes in a string of morse symbols, and splits each morse word by finding each double space in the string, and appends each morse word to an array. For each element of the array (a word written in morse), it will separate each group of dash and dot symbols that correspond to a normal letter, find their appropriate letter value, and append that letter to java class that builds a string from an array of letters.

```

19
20 handleCellSelected(index){
21     const newBoard = this.state.board.slice(0)
22
23     if (this.crossesTurn === true) {
24         if (newBoard[index] !== '0')
25         {
26             newBoard[index] = 'X'
27             this.crossesTurn = false
28         }
29     } else {
30         if (newBoard[index] !== 'X')
31         {
32             newBoard[index] = '0'
33             this.crossesTurn = true
34         }
35     }
36     this.setState({board: newBoard})
37 }

```

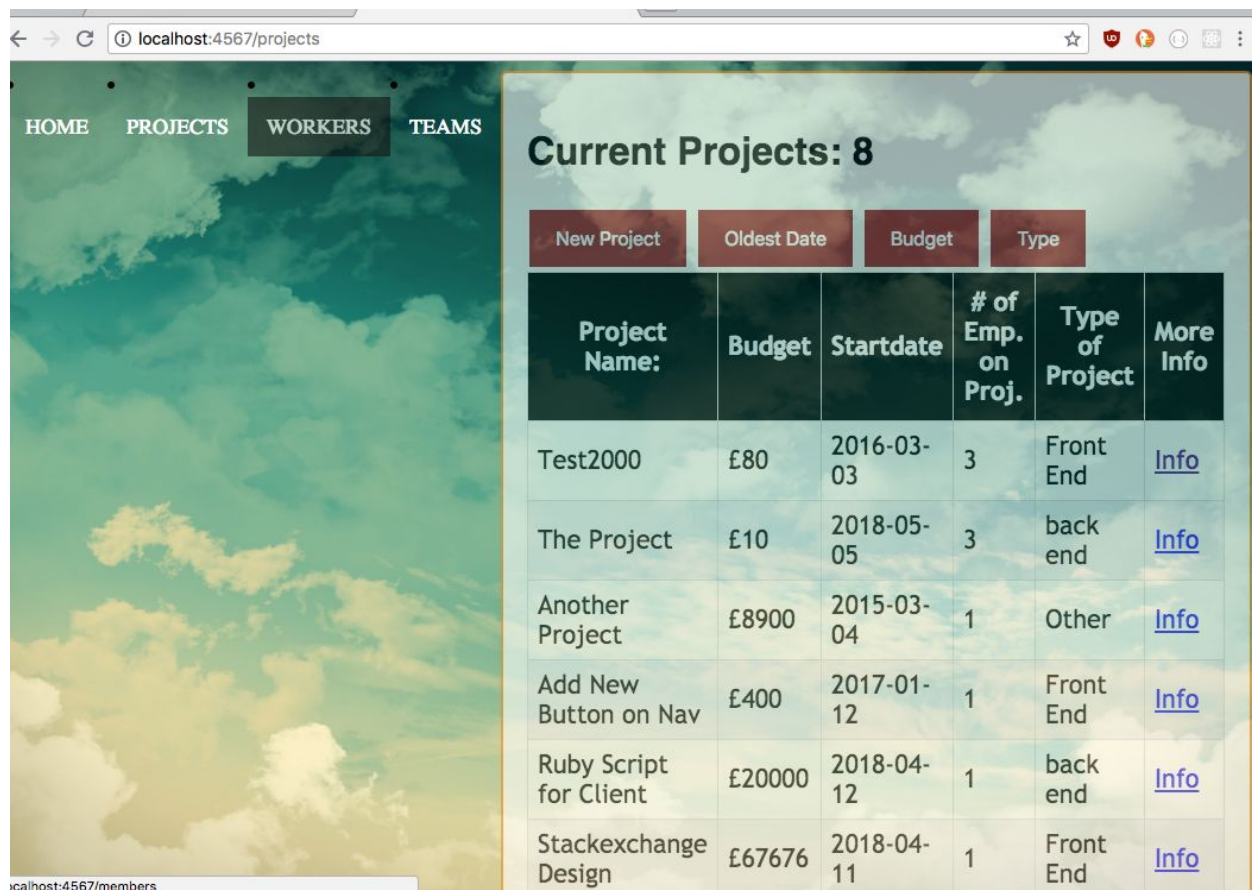

The above algorithm is used in tic-tac-toe game. It checks if the selected space has a corresponding value of 'X' or 'O', and if there is no value, it will add the corresponding value to the cell depending on whose turn it is to play.

P. 10 Example of Pseudocode

```
1
2
3 def method
4   split long string of dots and dashes into an array of
5   short strings at every double space. compare each
6   element of the array to the enum of corresponding
7   letters for the dots and dashes and append the value
8   onto a new array. concatenate the elements of the
9   second array and print as a string.
10  end
```

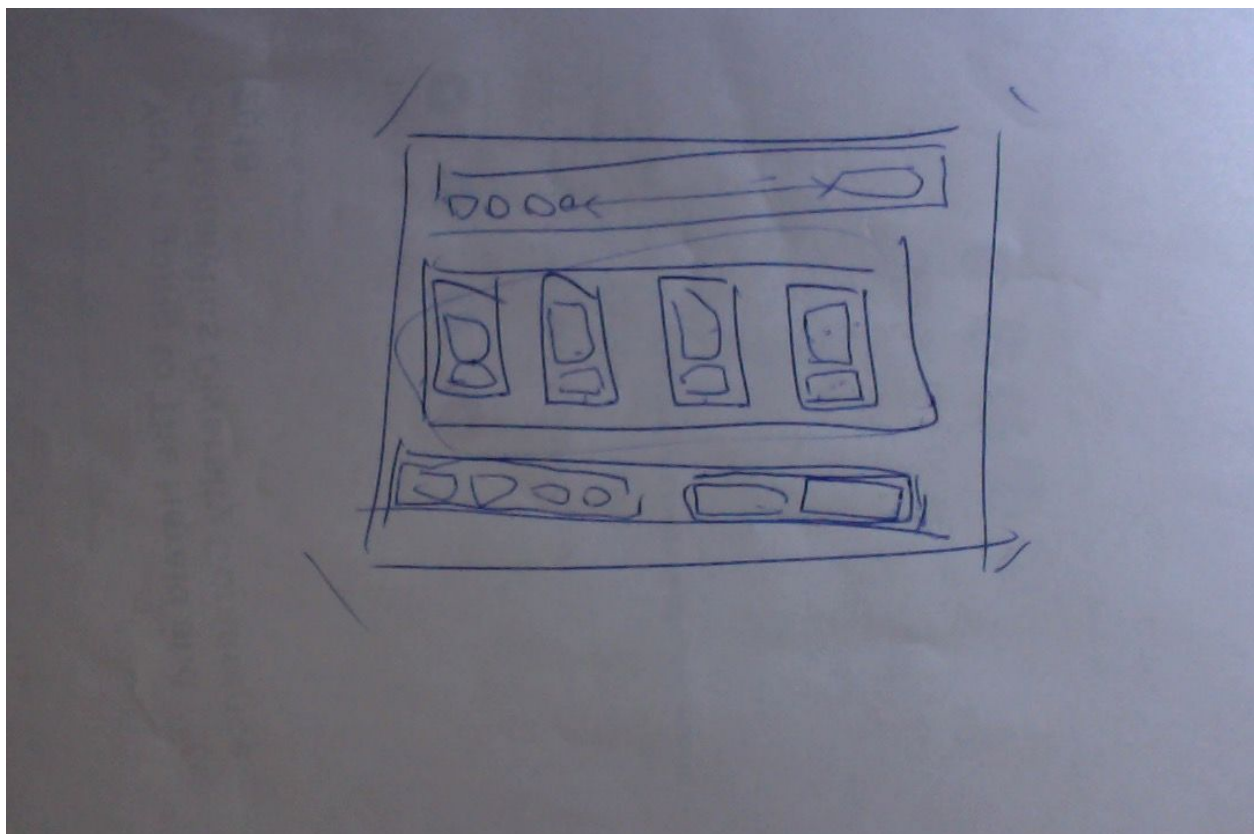
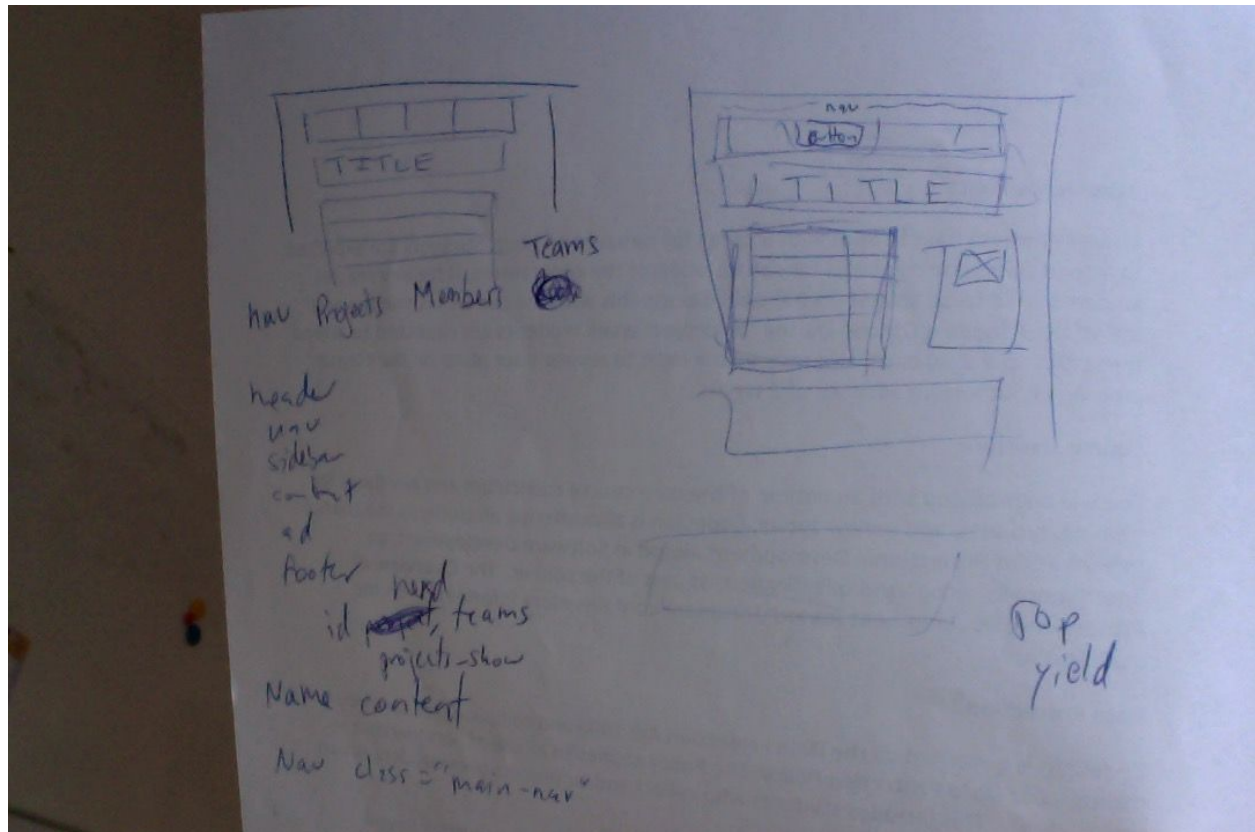
P. 11 Github link to on of you projects

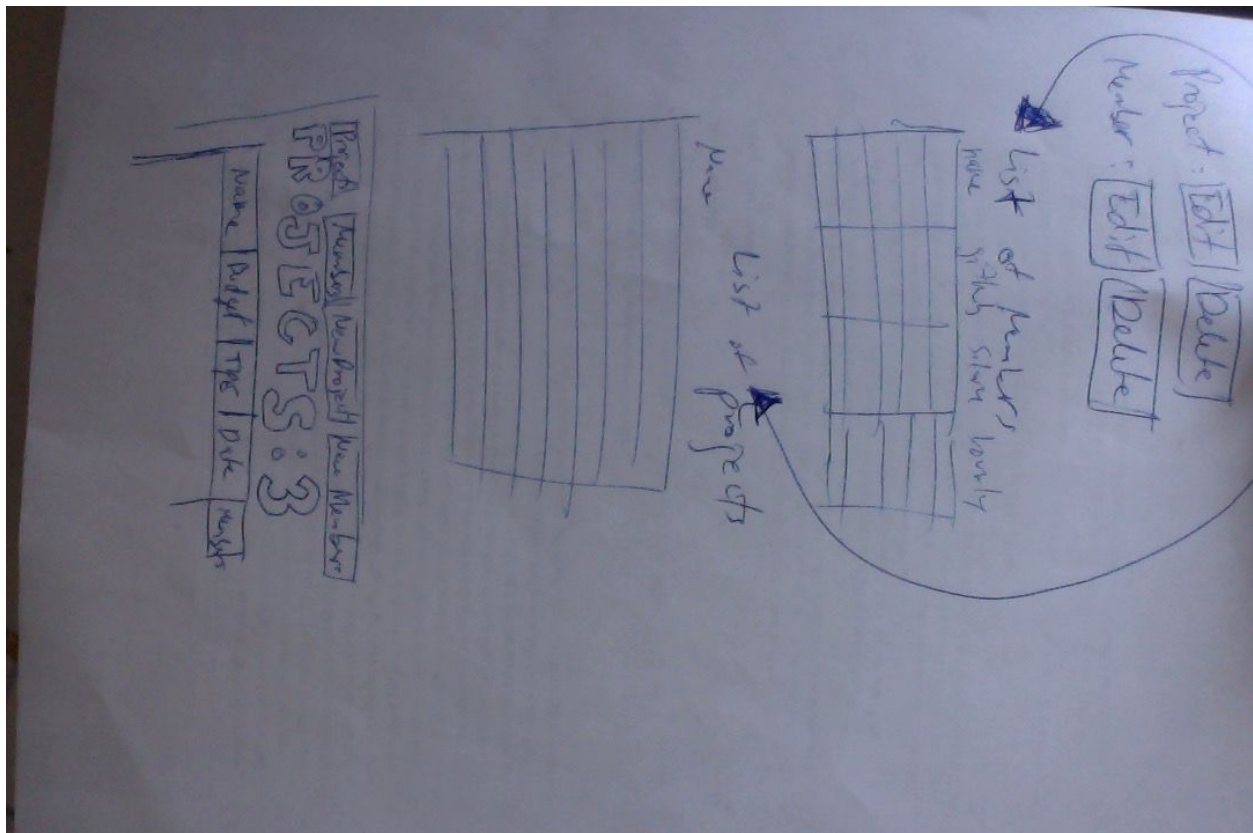
Github link: https://github.com/b09/project_dashboard



Current Projects: 8					
New Project Oldest Date Budget Type					
Project Name:	Budget	Startdate	# of Emp. on Proj.	Type of Project	More Info
Test2000	£80	2016-03-03	3	Front End	Info
The Project	£10	2018-05-05	3	back end	Info
Another Project	£8900	2015-03-04	1	Other	Info
Add New Button on Nav	£400	2017-01-12	1	Front End	Info
Ruby Script for Client	£20000	2018-04-12	1	back end	Info
Stackexchange Design	£67676	2018-04-11	1	Front End	Info

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





passing a string to controller p

none for dropdown
add
none for dropdown
remove

get	members	members/index	@members = Mem.All
members/new	members/new	members/new	
member.id/edit	member/edit	member/edit	@mem = Mem.And
member.id	member/show	member/show	@mem = Mem.And

post

members	get members	Mem.new → mem.save
members/:id	members/params[id]	Mem.new → mem.update
members/:id/delete	members	Mem.delete(params[id])

get projects	@Proj = Proj.All, @mems = Mem.All
get projects/new	
get projects/:id/edit	@mems = Mem.All

P. 13 User input (make sure to show input being added) & P. 14 Interaction with data persistence

Current Projects: 8

New project selected

New Project	Oldest Date	Budget	Type		
Project Name:	Budget	Startdate	# of Emp. on Proj.	Type of Project	More Info
Test2000	£80	2016-03-03	3	Front End	Info
The Project	£10	2018-05-05	3	back end	Info
Another Project	£8900	2015-03-04	1	Other	Info
Add New Button on Nav	£400	2017-01-12	1	Front End	Info

New Project Form

Name:

Alloted Budget:

Select start date:

Project Description:

Front End

Create Project

Cancel

Create Project button selected after filling out form

Current Projects: 9

Project now viewable on website

New Project	Oldest Date	Budget	Type		
Project Name:	Budget	Startdate	# of Emp. on Proj.	Type of Project	More Info
PDA Data Persistence	£9	2018-01-01	0	Front End	Info
The Project	£10	2018-05-05	3	back end	Info
Test2000	£80	2016-03-03	3	Front End	Info
Add New Button		2017-01-12			

P. 15 User output result

Current Projects: 9

Budgets are unordered

New Project	Oldest Date	Budget	Type		
Project Name:	Budget	Startdate	# of Emp. on Proj.	Type of Project	More Info
Test2000	£80	2016-03-03	3	Front End	Info
The Project	£10	2018-05-05	3	back end	Info
Another Project	£8900	2015-03-04	1	Other	Info
Add New Button on Nav	£400	2017-01-12	1	Front End	Info
Ruby Script for Client	£20000	2018-04-12	1	back end	Info
Stackexchange Design	£67676	2018-04-11	1	Front End	Info
New Project For	£80000	2017-04-12	1	Front End	Info

Current Projects: 9

Budget button selected

All budgets ordered from lowest to highest

New Project	Oldest Date	Budget	Type		
Project Name:	Budget	Startdate	# of Emp. on Proj.	Type of Project	More Info
PDA Data Persistence	£9	2018-01-01	0	Front End	Info
The Project	£10	2018-05-05	3	back end	Info
Test2000	£80	2016-03-03	3	Front End	Info
Add New Button on Nav	£400	2017-01-12	1	Front End	Info
Another Project	£8900	2015-03-04	1	Other	Info
Breakfast App	£9000	2017-03-05	0	Front End	Info
Ruby Script for Client	£20000	2018-04-12	1	back end	Info

P. 16 Show API used within your program

```
app.get(`/iss-data/astronauts`, (req, res) => {
  const url = 'http://api.open-notify.org/astros.json';

  fetch(url)
    .then(jsonData => jsonData.json())
    .then(data => res.json(data));
});
```

People in Space

Oleg Artemyev

Andrew Feustel

Richard Arnold

Sergey Prokopyev

Alexander Gerst

Serena Aunon-Chancellor

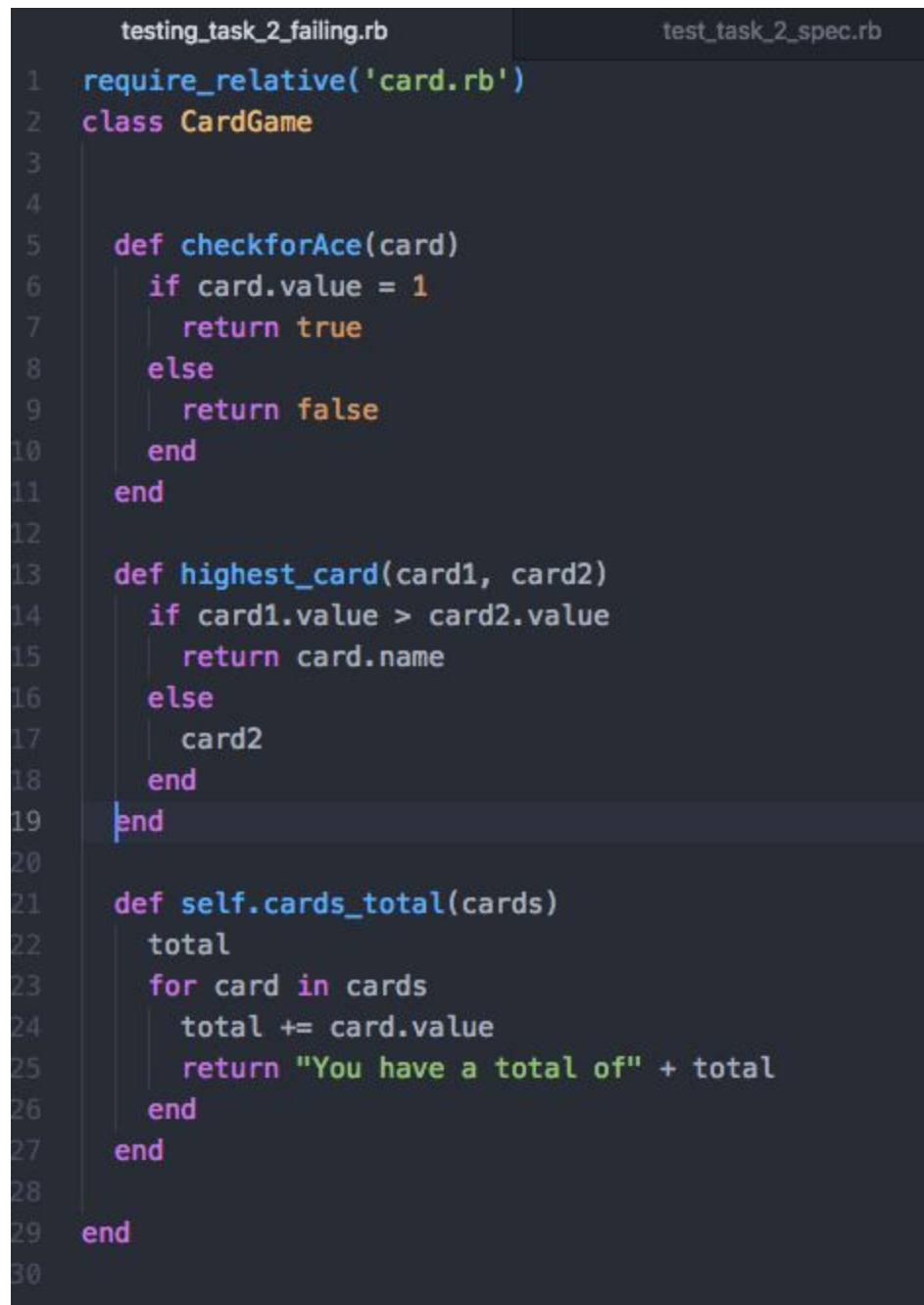


P. 17 Bug tracking report showing the errors diagnosed and corrected

City names load within one second	FAIL	Populate cities only after they match with three inputted characters to reduce search load	PASS
Wikipedia articles are available based on exact gps coordinates	FAIL	Request wikipedia articles within a large enough radius to get results	PASS
Make an Ajax request directly from the browser	FAIL	Route the ajax requests through a local server	PASS
Video from external website works	FAIL	Change video link to another video	PASS
Map loading properly	FAIL	Change order of render methods	PASS

P. 18 Demonstrate testing: include example of test code, test code failing, corrected test code, test passing

1)Failing script



The image shows a code editor with two tabs: 'testing_task_2_failing.rb' and 'test_task_2_spec.rb'. The 'testing_task_2_failing.rb' tab is active, displaying the following Ruby code:

```
1 require_relative('card.rb')
2 class CardGame
3
4
5   def checkforAce(card)
6     if card.value = 1
7       return true
8     else
9       return false
10    end
11  end
12
13  def highest_card(card1, card2)
14    if card1.value > card2.value
15      return card.name
16    else
17      card2
18    end
19  end
20
21  def self.cards_total(cards)
22    total
23    for card in cards
24      total += card.value
25      return "You have a total of" + total
26    end
27  end
28
29 end
30
```

2) Failing test in terminal

```

→ specs git:(master) x ruby test_task_2_spec.rb
Run options: --seed 63294

# Running:

F

Failure:
TestCardGame#test_for_highest_card [test_task_2_spec.rb:29]:
--- expected
+++ actual
@@ -1, +1 @@
-4
+#{<Card:0XXXXXX @suit="Diamonds", @value=4>

bin/rails test test_task_2_spec.rb:27

E

Error:
TestCardGame#test_for_cards_total:
NameError: undefined local variable or method `total' for CardGame:Class
/Users/macbook/codeclan/e20/workfiles/pda_related/pda_evidence/Static
test_task_2_spec.rb:33:in `test_for_cards_total'

bin/rails test test_task_2_spec.rb:32

E

Error:
TestCardGame#test_for_ace__true:
NoMethodError: undefined method `check_for_ace' for #{<CardGame:0x007f8c3
Did you mean? checkforAce
test_task_2_spec.rb:18:in `test_for_ace__true'

bin/rails test test_task_2_spec.rb:17

E

Error:
TestCardGame#test_for_ace__false:

```

3) Passing script

```

require_relative('card.rb')

class CardGame

  def check_for_ace(card)
    if card.value == 1
      return true
    else
      return false
    end
  end

  def highest_card(card1, card2)
    if card1.value > card2.value
      return card1.value
    else
      return card2.value
    end
  end

  def self.cards_total(cards)
    total = 0
    for card in cards
      total += card.value
    end
    return "You have a total of " + total.to_s
  end
end

```

4) Passing test in terminal

```

→ specs git:(master) x ruby test_task_2_spec.rb
Run options: --seed 5783

# Running:

....

Finished in 0.002064s, 1937.9845 runs/s, 1937.9845 assertions/s.
4 runs, 4 assertions, 0 failures, 0 errors, 0 skips

```

5) Test script


```

testing_task_2_failing.rb | test_task_2_spec.rb
require('minitest/autorun')
require('minitest/rg')
require('../testing_task_2_failing.rb')
require('../card.rb')

class TestCardGame < Minitest::Test

  def setup
    @card1 = Card.new('Hearts', 1)
    @card2 = Card.new('Spades', 2)
    @card3 = Card.new('Clubs', 3)
    @card4 = Card.new('Diamonds', 4)
    @card_game = CardGame.new()
    @all_cards = [@card1, @card2, @card3, @card4]
  end

  def test_for_ace__true
    result = @card_game.check_for_ace(@card1)
    assert_equal(true, result)
  end

  def test_for_ace__false
    result = @card_game.check_for_ace(@card2)
    assert_equal(false, result)
  end

  def test_for_highest_card
    result = @card_game.highest_card(@card3, @card4)
    assert_equal(4, result)
  end

  def test_for_cards_total
    result = CardGame.cards_total(@all_cards)
    assert_equal("You have a total of 10", result)
  end
end
end

```