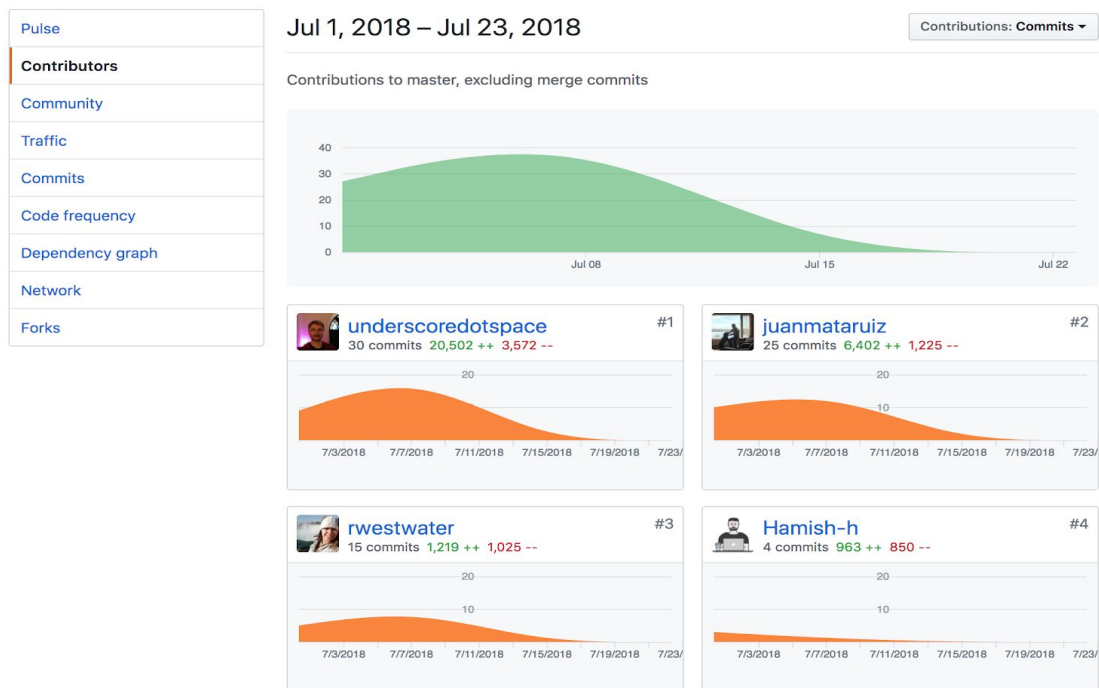


Evidence for Project Unit

Juan Mata Ruiz
E21

P. 1 Github Contributors page



P. 2 Project Brief

Shares App

A local trader has come to you with a portfolio of shares. She wants to be able to analyse it more effectively. She has a small sample data set to give you and would like you to build a minimal viable product (MVP) that uses the data to display her portfolio in useful ways so that she can make better decisions.

MVP

- View total current value
- View individual and total performance trends
- Retrieve a list of share prices from an external API and allow the user to add shares to her portfolio
- Provide a chart of the current values in her portfolio

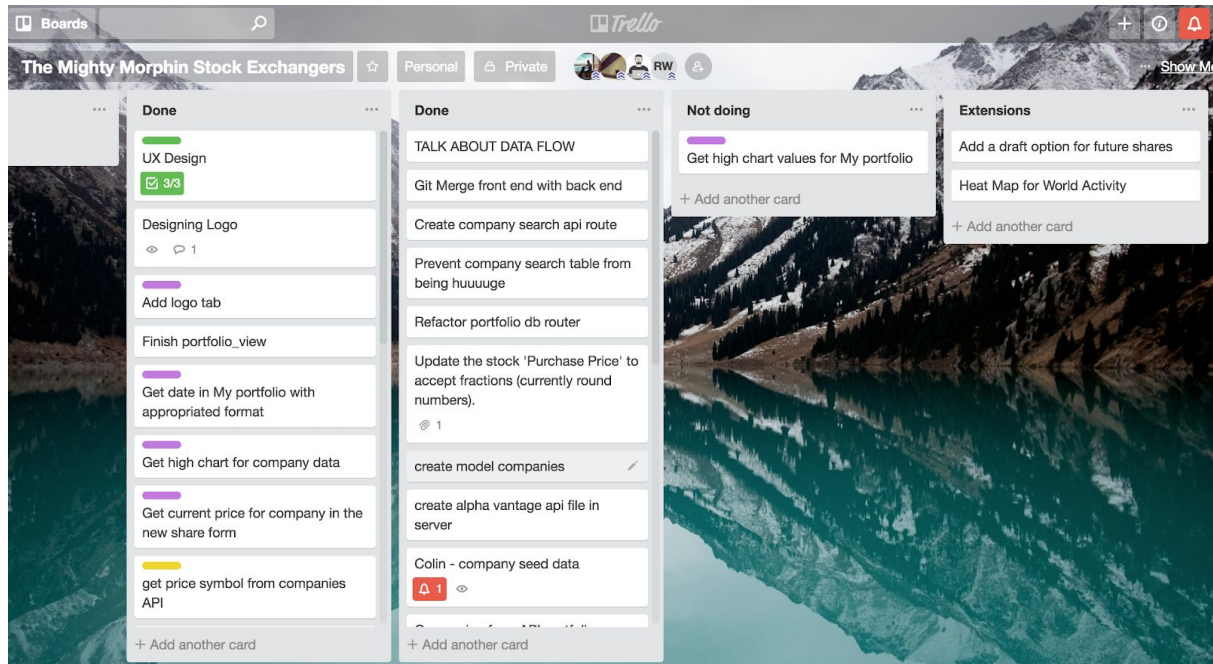
Examples of Further Features

- Speculation based on trends and further financial modelling using projections.

API, Libraries, Resources

- <https://www.alphavantage.co/> (Requires sign up)
- <https://www.highcharts.com/> HighCharts is an open-source library for rendering responsive charts.

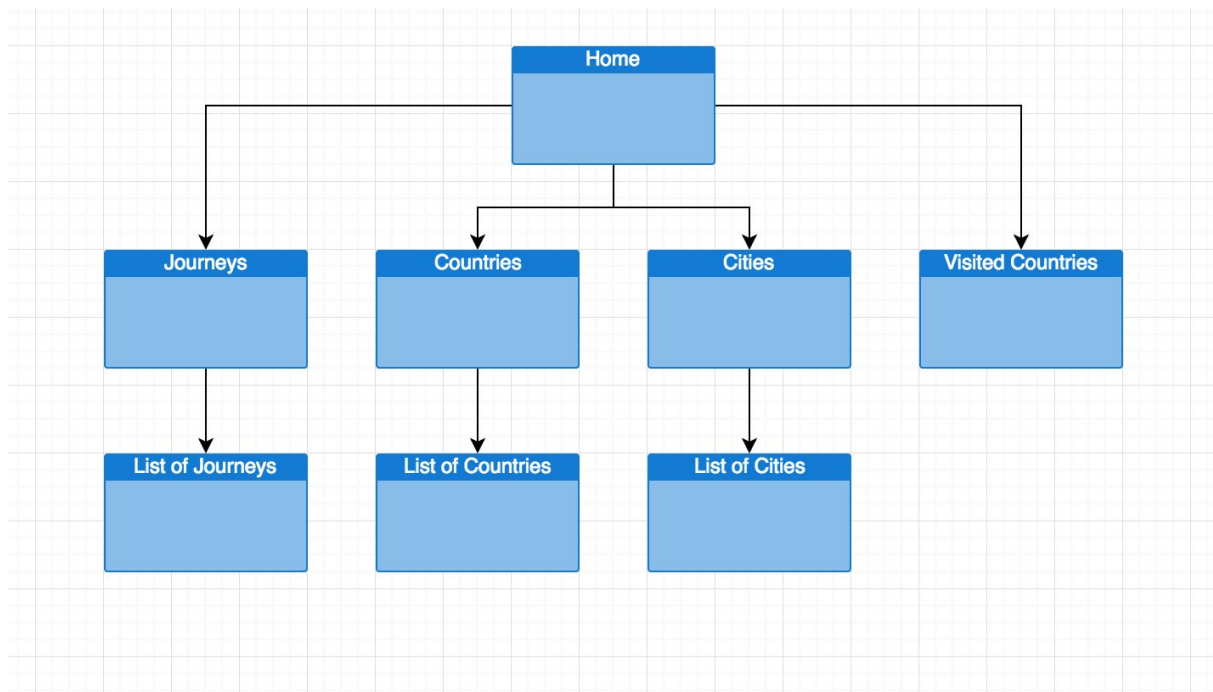
P. 3 Use of Trello



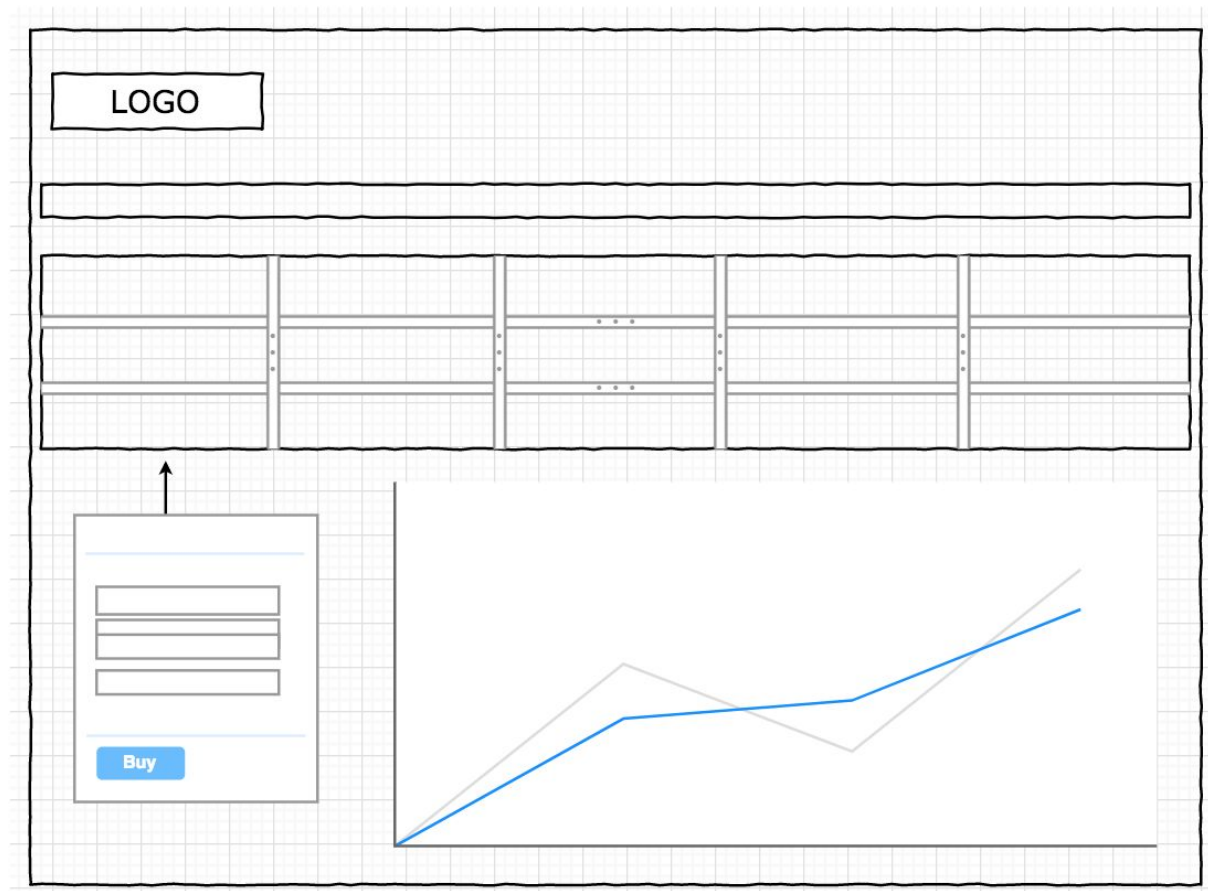
P. 4 Acceptance Criteria

Acceptance Criteria	Expected Output	Result
User can't move herbivore and carnivore dinosaurs together	Iterate through my dinosaurs arrayList looking for tall dinosaurs of the same type	Pass
User can feed dinosaurs if they have enough experience	Dinosaurs are fed if user has enough experience	Pass
User can add visitor to the park if the dinosaurs are rampaging	No visitor is allowed to enter the park if a dinosaur is on a rampage mode	Pass
User is able to test randomness	Everytime a random value is tested it passed	Pass

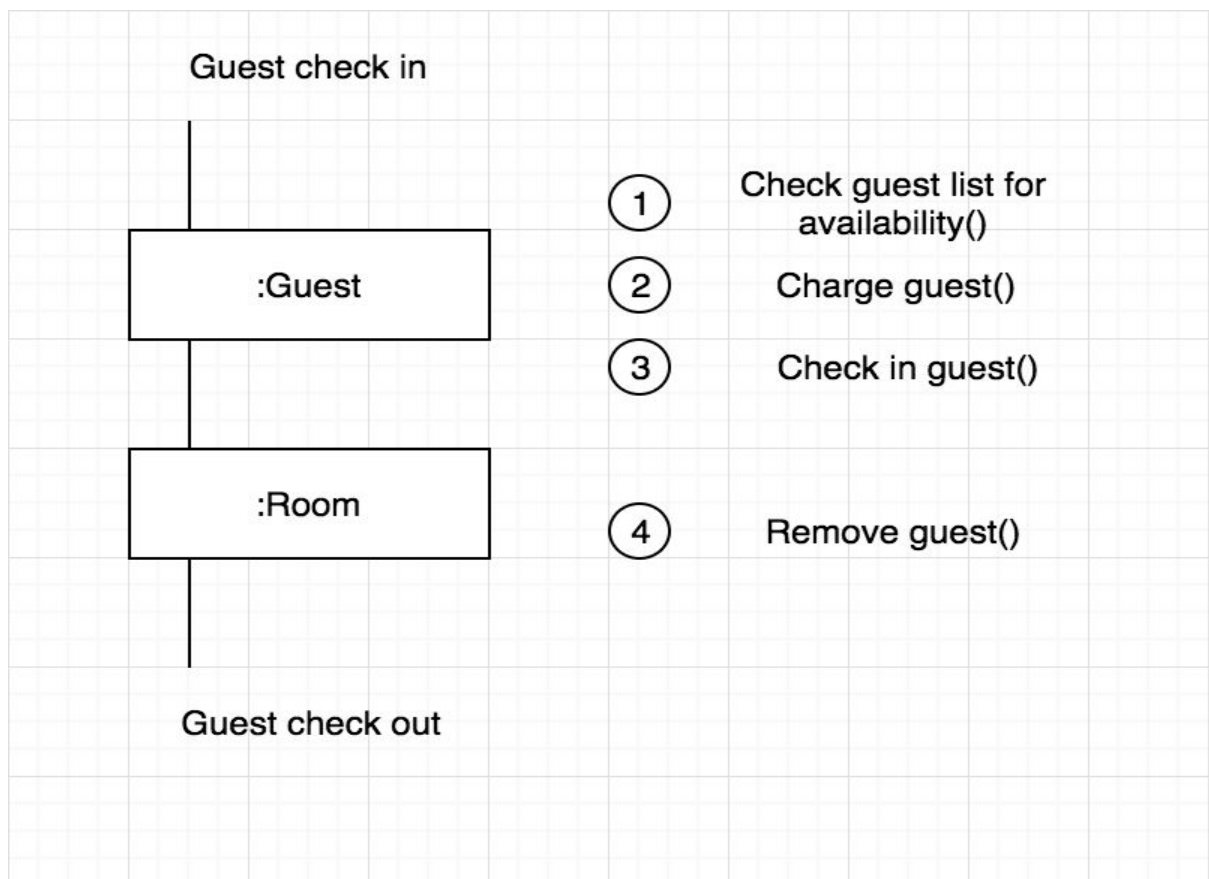
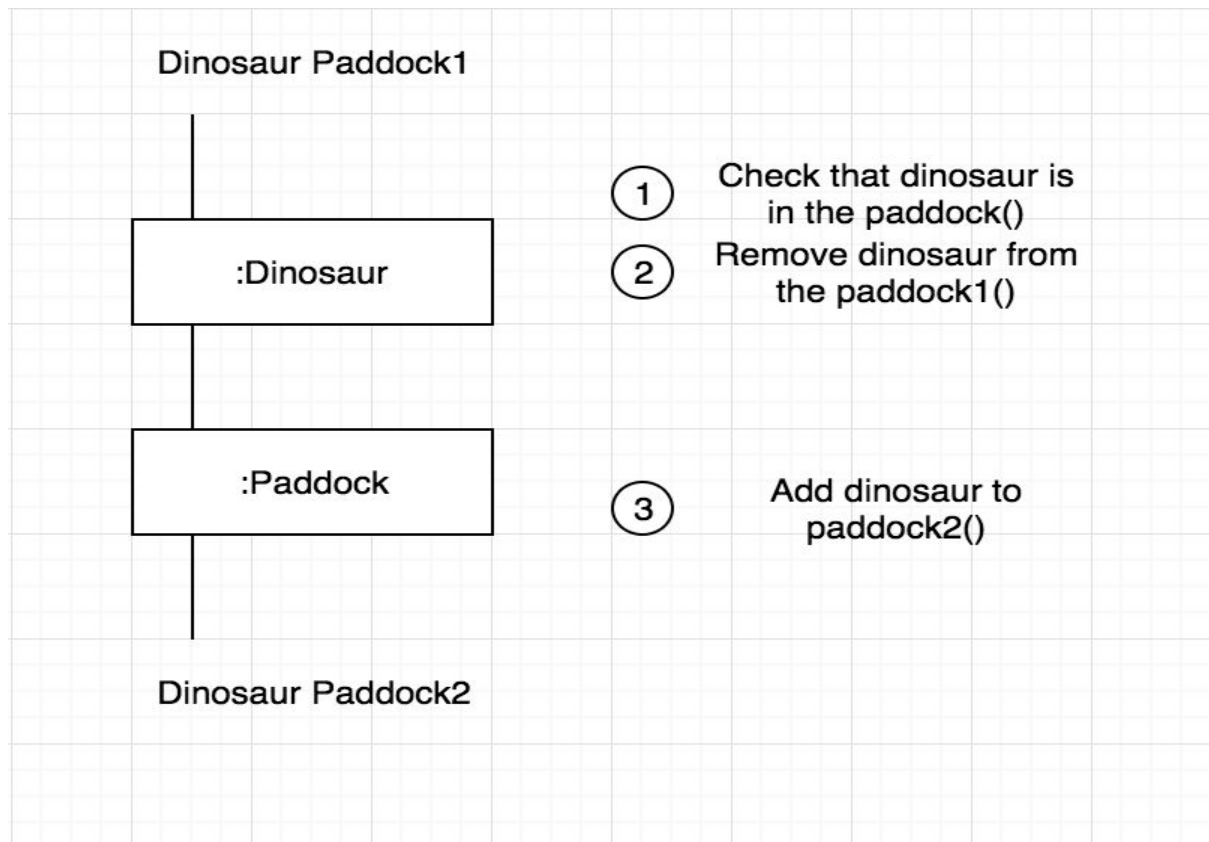
P. 5 User sitemap



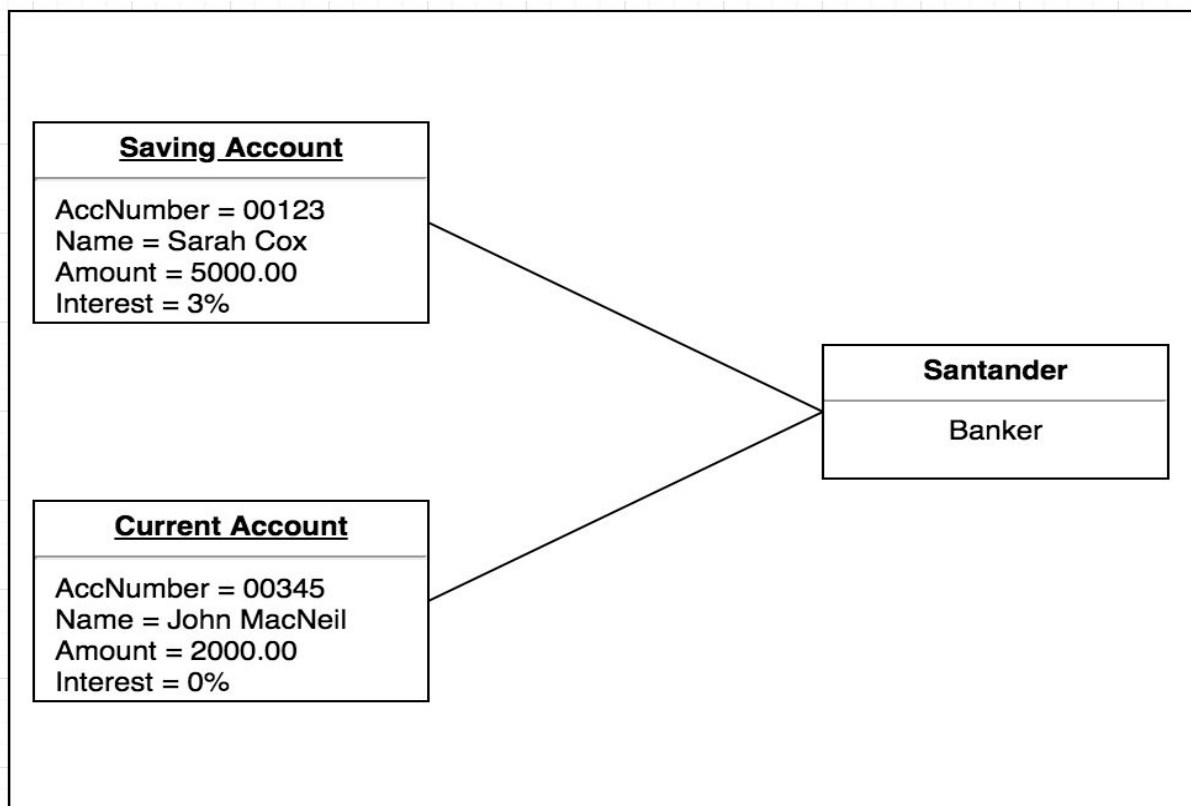
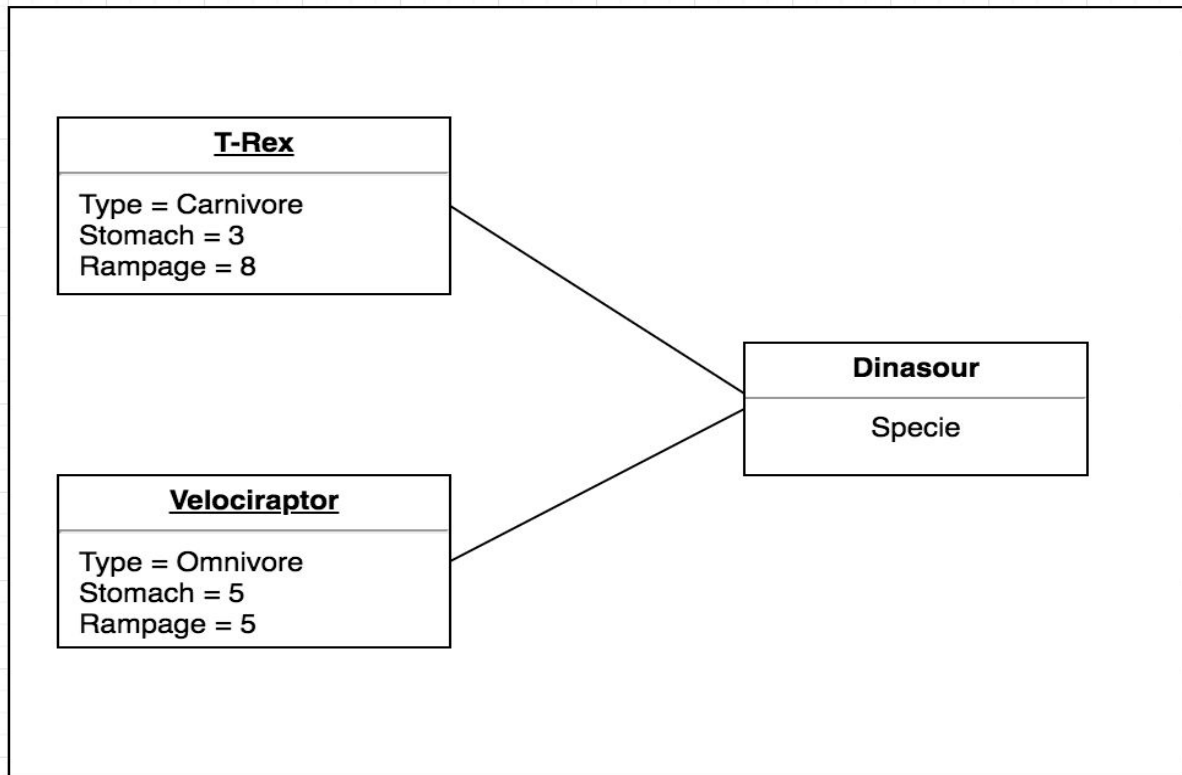
P. 6 Wireframes designs



P. 7 System interactions diagrams



P. 8 Two object diagrams



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

```
rps.rb x
1  class Game
2
3    def initialize(hand1, hand2)
4      @hand1 = hand1
5      @hand2 = hand2
6    end
7
8    def play()
9      if (@hand1 == "rock") && (@hand2 == "rock")
10         return "Draw"
11      elsif (@hand1 == "rock") && (@hand2 == "scissors")
12         return "Rock wins"
13      elsif (@hand1 == "rock") && (@hand2 == "paper")
14         return "Paper wins"
15      elsif (@hand1 == "paper") && (@hand2 == "rock")
16         return "Paper wins"
17      elsif (@hand1 == "paper") && (@hand2 == "scissors")
18         return "Scissors wins"
19      elsif (@hand1 == "paper") && (@hand2 == "paper")
20         return "Draw"
21      elsif (@hand1 == "scissors") && (@hand2 == "scissors")
22         return "Draw"
23      elsif (@hand1 == "scissors") && (@hand2 == "rock")
24         return "Rock wins"
25      elsif (@hand1 == "scissors") && (@hand2 == "paper")
26         return "Scissors wins"
27      else
28         return "Invalid route, try again"
29      end
30    end
31  end
32
33
```

```
def favourite_song(song)
  message = "We don't have your favourite song"
  @songs.each { |fav| if fav == song
    message = "Whoo! We have your favourite song."
  }
  return message
end
```

P. 10 Example of pseudocode

```
//park must have an array of paddocks
//park must have an array of visitors
//park must be able to count the number of paddocks
//park must be able to add paddock
//park must be able to remove paddock
//park must be able to get the total visitor in the park
//park must be able to add visitor
//park must be able to remove visitor
//park must be able to transfer dinosaur between paddocks
```

P. 11 Github link to one of your projects

https://github.com/juanmataruiz/Java_Project-JurassicPark

The screenshot shows the GitHub repository page for 'Java_Project-JurassicPark' by user 'juanmataruiz'. The repository has 24 commits, 1 branch, 0 releases, and 1 contributor. The latest commit is 'ec5e00' on 6 Jun. The repository contains several files and folders, including .gradle, .idea, gradle/wrapper, out, src, build.gradle, gradlew, gradlew.bat, and settings.gradle. A banner at the bottom encourages adding a README file.

GitHub, Inc. [US] github.com/juanmataruiz/Java_Project-JurassicPark

juanmataruiz / Java_Project-JurassicPark

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. [Add topics](#)

24 commits 1 branch 0 releases 1 contributor

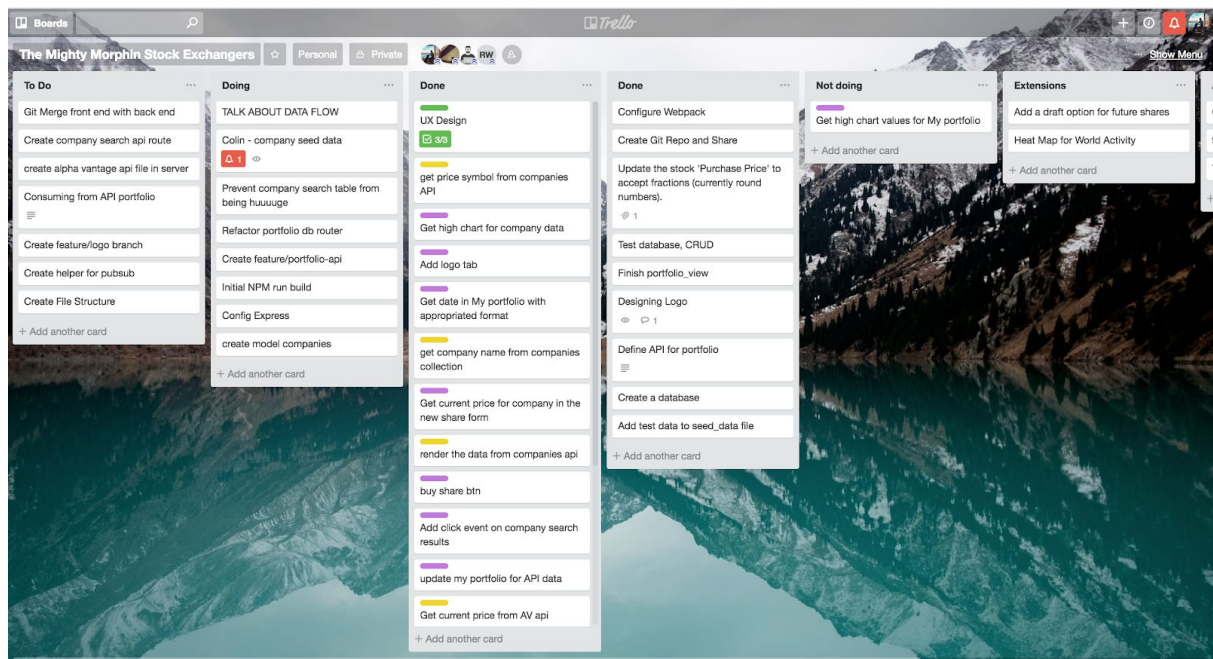
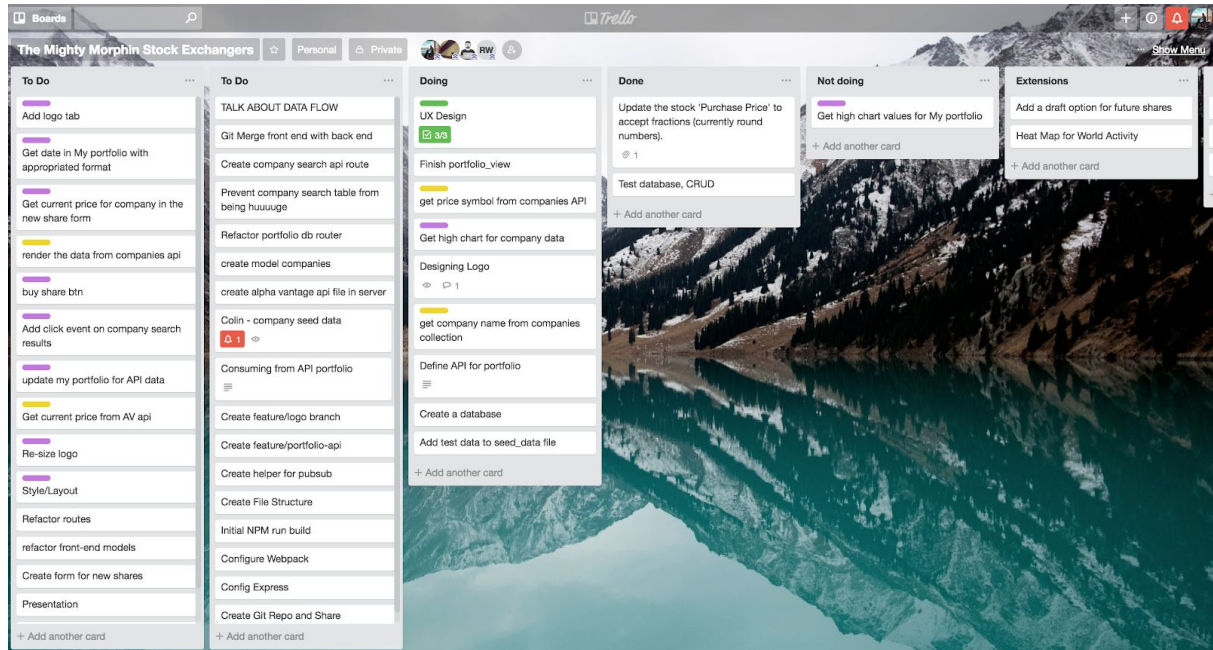
Branch: master New pull request Create new file Upload files Find file Clone or download

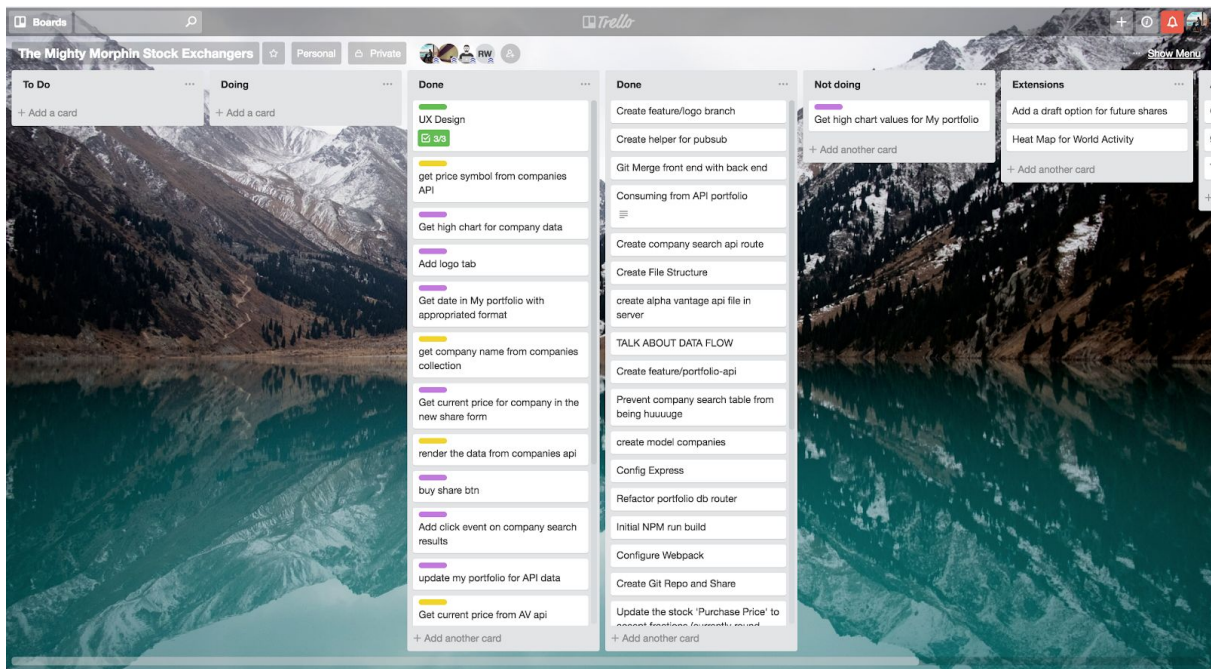
juanmataruiz ankylosaurus class can rampage and break free Latest commit ec5e00 on 6 Jun

.gradle	dinosaur can rampage and hunt	2 months ago
.idea	ankylosaurus class can rampage and break free	2 months ago
gradle/wrapper	dinosaur can rampage and hunt	2 months ago
out	ankylosaurus class can rampage and break free	2 months ago
src	ankylosaurus class can rampage and break free	2 months ago
build.gradle	dinosaur can rampage and hunt	2 months ago
gradlew	initial commit	2 months ago
gradlew.bat	initial commit	2 months ago
settings.gradle	initial commit	2 months ago

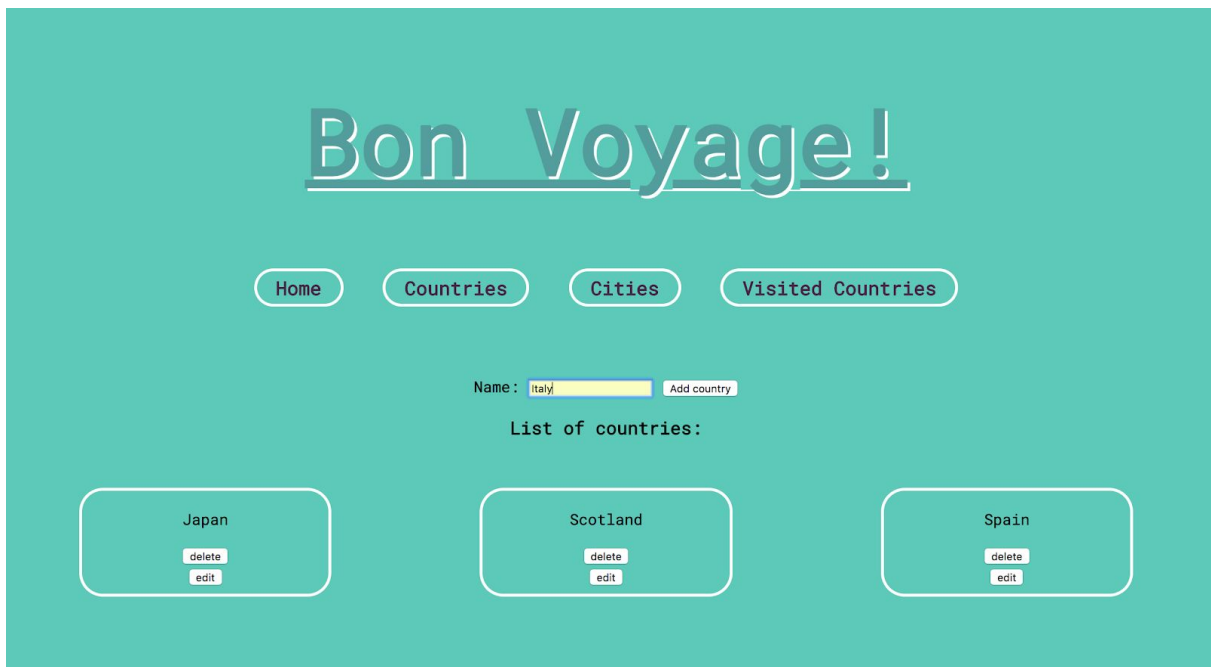
Help people interested in this repository understand your project by adding a README. [Add a README](#)

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





P. 13 User input *



P. 14 Interaction with data persistence

Bon Voyage!

Home Countries Cities Visited Countries

Name:

List of countries:

Japan

Scotland

Spain

Italy

P. 15 User output result

Bon Voyage!

Home Countries Cities Visited Countries

Name:

List of countries:

Japan

Scotland

Spain

Italy

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

User must be able to add a new city	Failed	Dropped database to refresh the content for the previous method to work	Passed
User must be able to add a new country	Failed	Spelling mistake in save method corrected	Passed
User must be able to update existing journeys	Failed	Reviewed RESTful route and corrected the path to /update	Passed
User can select existing countries from the drop down menu	Failed	Syntax error in PostgreSQL corrected	Passed
User has access to cities and countries in the same page	Failed	Create a double inner joiner to join both tables together	Passed

P.17 Testing your program

```
require_relative '../dynamic_testing.rb'
require_relative '../card.rb'
require 'minitest/autorun'
require('minitest/rng')

class TestCard < MiniTest::Test

  def setup
    @card1 = Card.new("One", 1)
    @card2 = Card.new("Two", 2)
    @card_game = CardGame.new
    @cards = [@card1, @card2]
  end

  def test_check_for_ace__True
    assert_equal(false, @card_game.check_for_ace(@card1))
  end

  def test_check_for_ace__False
    assert_equal(true, @card_game.check_for_ace(@card2))
  end

  def test_highest_card
    assert_equal(3, @card_game.highest_card(@card2, @card1))
  end

  def test_cards_total
    assert_equal("You have a total of 3", CardGame.cards_total(@cards))
  end

end
```

```
Static_and_Dynamic_Task_A — user@users-MacBook-Pro — .ynamic_Task_A —...
➔ Static_and_Dynamic_Task_A ruby specs/dynamic_testing_spec.rb
Run options: --seed 59556

# Running:

FFFF

Finished in 0.001285s, 3112.8405 runs/s, 3112.8405 assertions/s.

  1) Failure:
TestCard#test_cards_total [specs/dynamic_testing_spec.rb:29]:
Expected: "You have a total of 3"
Actual: "You have a total of 1"

  2) Failure:
TestCard#test_check_for_ace__False [specs/dynamic_testing_spec.rb:21]:
Expected: true
Actual: false

  3) Failure:
TestCard#test_highest_card [specs/dynamic_testing_spec.rb:25]:
Expected: 3
Actual: 2

  4) Failure:
TestCard#test_check_for_ace__True [specs/dynamic_testing_spec.rb:17]:
Expected: false
Actual: true

4 runs, 4 assertions, 4 failures, 0 errors, 0 skips
➔ Static_and_Dynamic_Task_A
```

```

require_relative '../dynamic_testing.rb'
require_relative '../card.rb'
require 'minitest/autorun'
require('minitest/rng')

class TestCard < MiniTest::Test

  def setup
    @card1 = Card.new("One", 1)
    @card2 = Card.new("Two", 2)
    @card_game = CardGame.new
    @cards = [@card1, @card2]
  end

  def test_check_for_ace__True
    assert_equal(true, @card_game.check_for_ace(@card1))
  end

  def test_check_for_ace__False
    assert_equal(false, @card_game.check_for_ace(@card2))
  end

  def test_highest_card
    assert_equal(2, @card_game.highest_card(@card2, @card1))
  end

  def test_cards_total
    assert_equal("You have a total of 1", CardGame.cards_total(@cards))
  end

end

```

```

Static_and_Dynamic_Task_A — user@users-MacBook-Pro — .ynamic_Task_A —...
➔ Static_and_Dynamic_Task_A ruby specs/dynamic_testing_spec.rb
Run options: --seed 46932

# Running:

....

Finished in 0.001150s, 3478.2609 runs/s, 3478.2609 assertions/s.

4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
➔ Static_and_Dynamic_Task_A

```