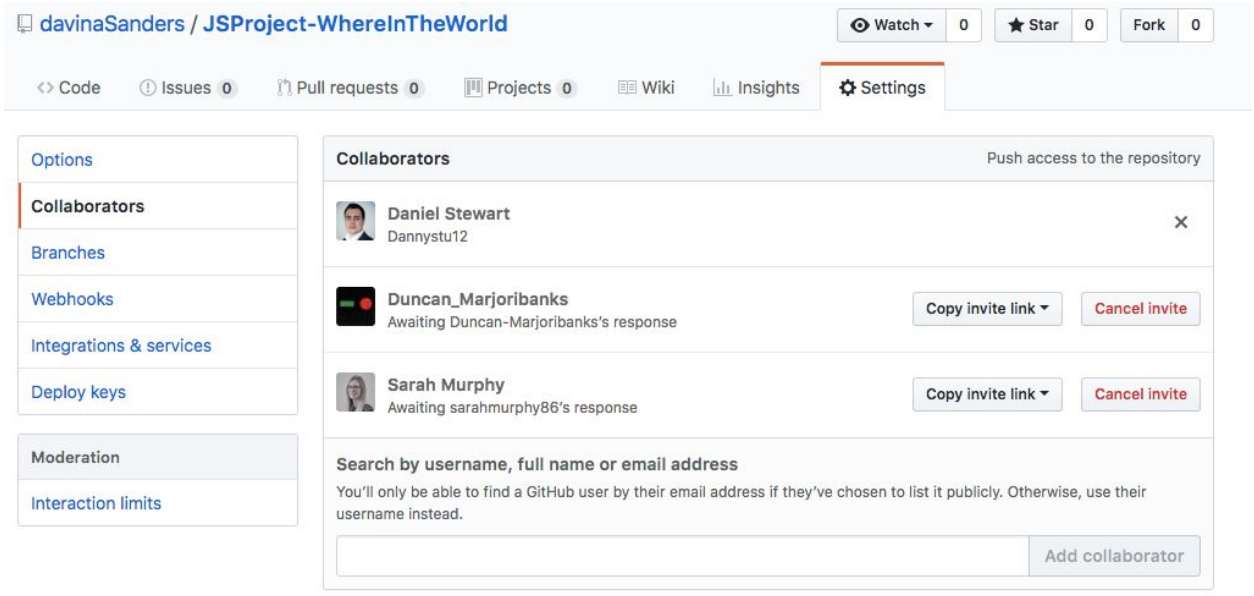
Project Unit

Sarah Murphy
Cohort E21

P1- Screenshot of the contributors page on Github



P2 - Screenshot of the project brief from your group project

Educational App

The BBC are looking to improve their online offering of educational content by developing some interactive apps that display information in a fun and interesting way. Your task is to make an MVP to put forward to them - this may only be for a small set of information, and may only showcase some of the features to be included in the final app. You might use an API to bring in content or a database to store facts.

The topic of the app is your choice, but here are some suggestions you could look into:

- Interactive timeline, e.g. of the history of computer programming
- Explore the Solar System - navigate through planets and display information
- Interactive map of a historical event - e.g. World War 1, the travels of Christopher Columbus

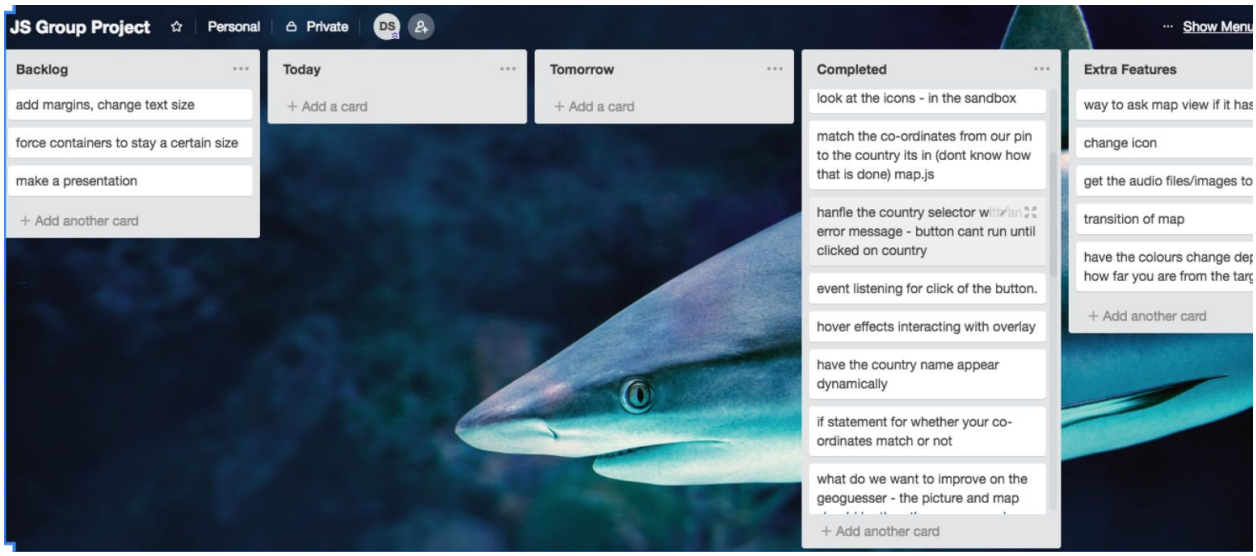
MVP

- Display some information about a particular topic in an interesting way
- Have some interactivity that enables a user to move through different sections of content

Examples of further features

- Bring in data using an API or create your own
- Use charts or maps to display your information

P3 - Planning you completed during your group project

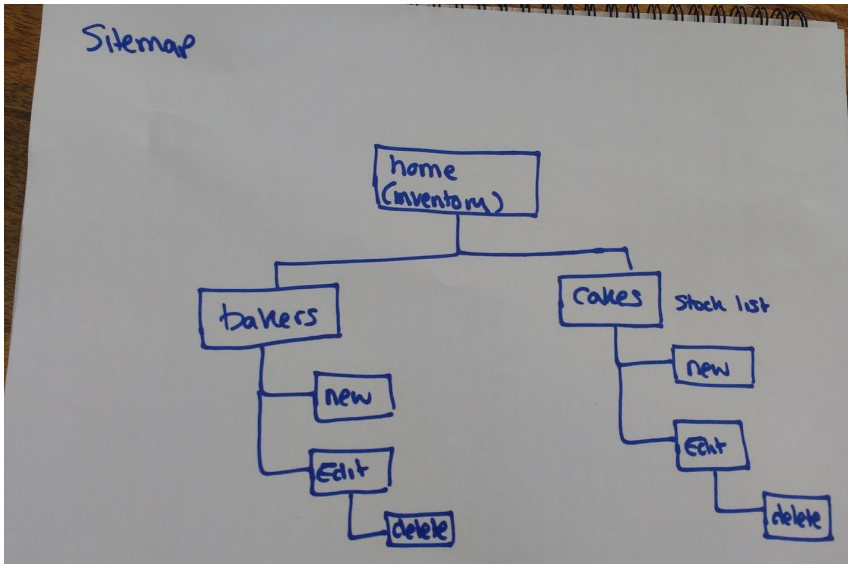


P4 - Acceptance criteria and test plan

Acceptance Criteria	Expected Result/Output	Pass/ Fail
The user can click on a country on an interactive map	A pop up indicates the name of the country and the country is highlighted	Pass
The user can then select a different country to the one previously selected	The new country is highlighted and the country name appears as a pop-up	Pass
A user is able to click on a submit button	They user is taken to a new page	Pass
The 'correct' country is displayed	A pin drops on the 'correct' country	Pass
A voice-over indicates whether the users guess is correct	Positive feedback is given for a correct answer and negative feedback is given for an incorrect answer	Pass

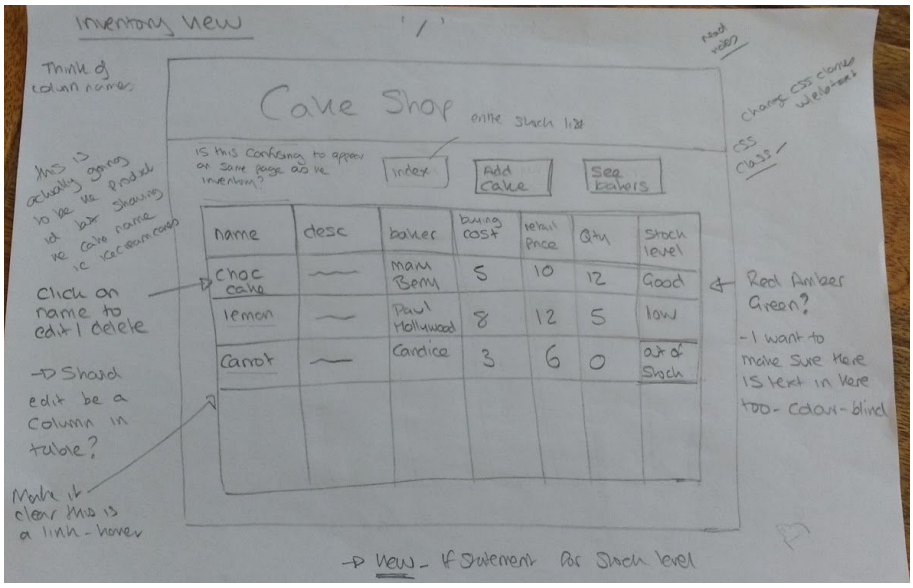
P5 - Sitemap

For cake shop app:

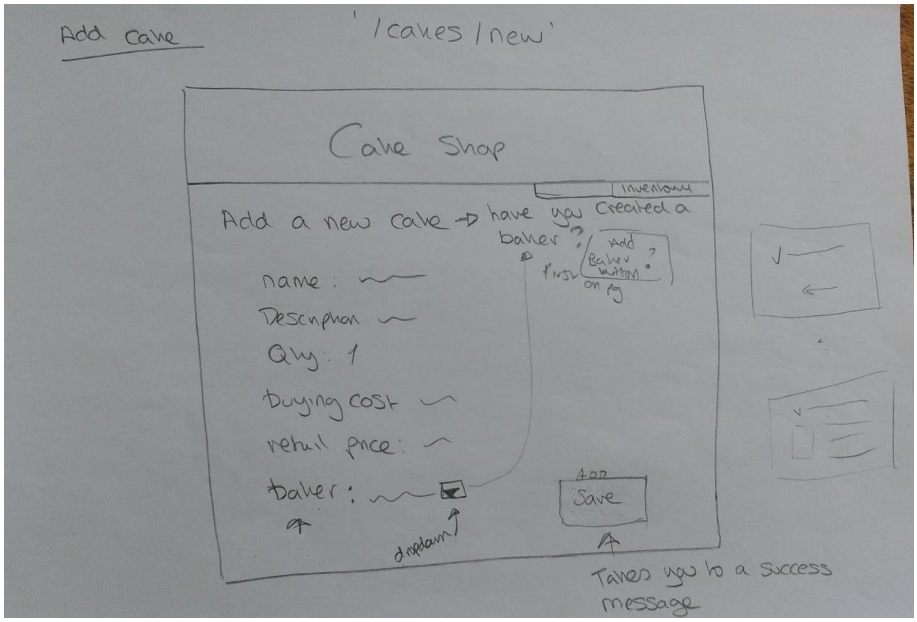


P6 - Wireframes

Cake shop - inventory view:

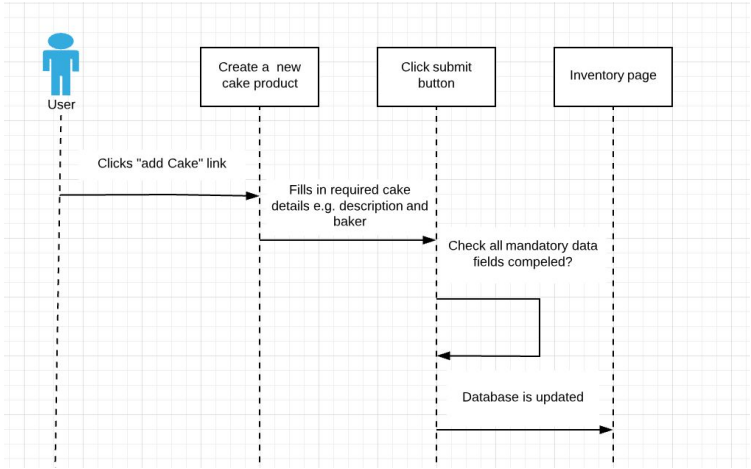


Cake shop - new cake view:

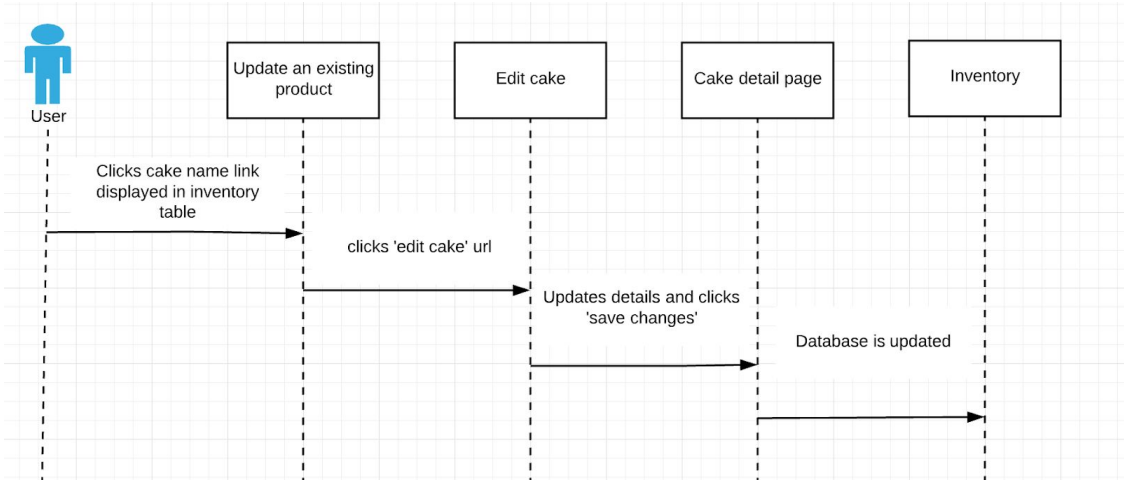


P7- Two system interaction diagrams

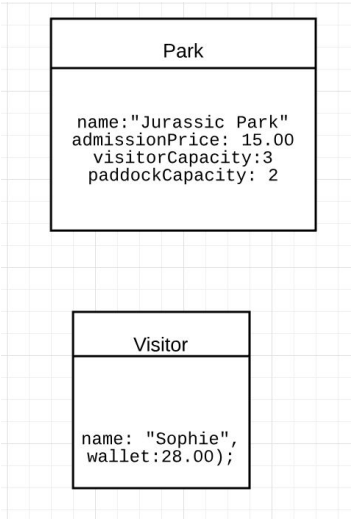
Adding a cake(product)



Update a cake(existing product)



P8- Two object diagrams



P9 - Two algorithms you have written

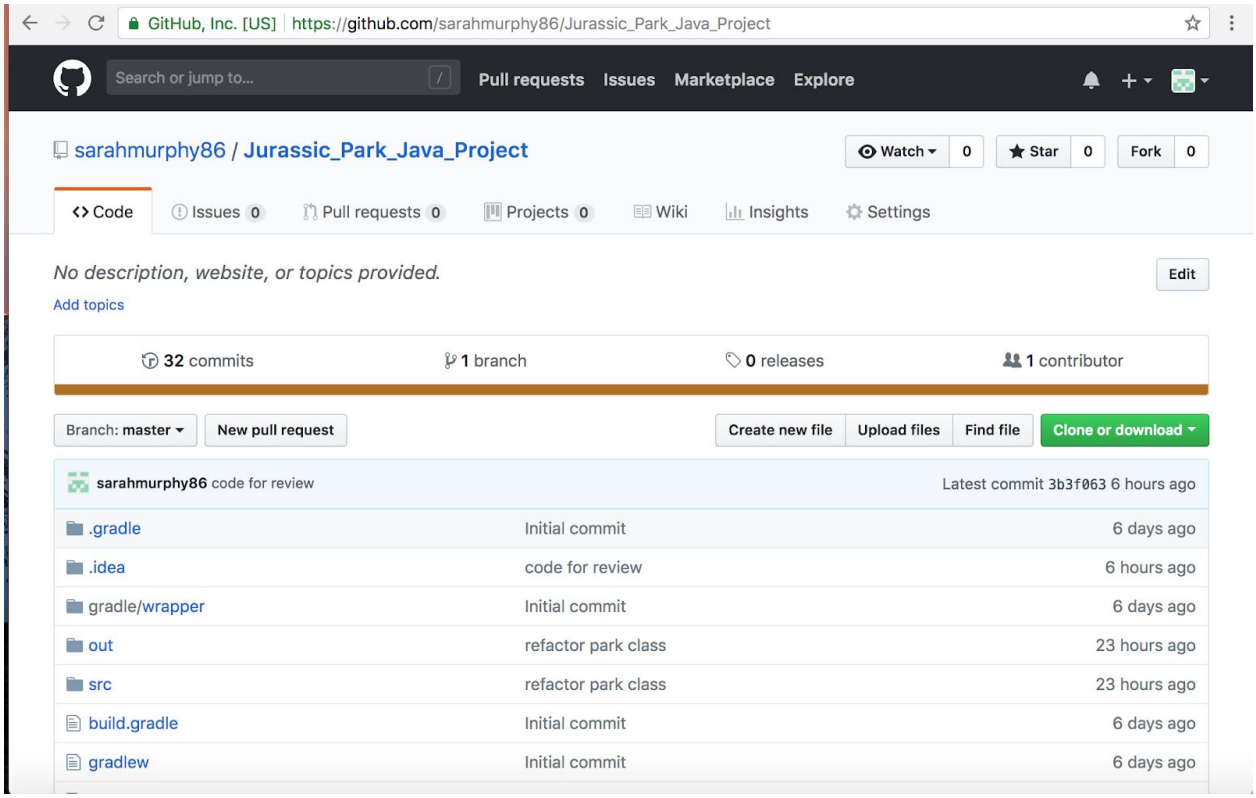
```ruby
def stock_level()
  if @quantity >= 8
    return "green"
  elsif @quantity >=3 && @quantity <=7
    return "amber"
  else
    return "red"
  end
end
```

```java
  public double totalCostOfItems() {
      double total = 0;

      for (Item item : this.items) {
          total += item.getSellingPrice();
      }

      if (total <= 20) {

          return total;
      } else return total - (total / 10);

  }
```

P10 - Pseudocode for a function

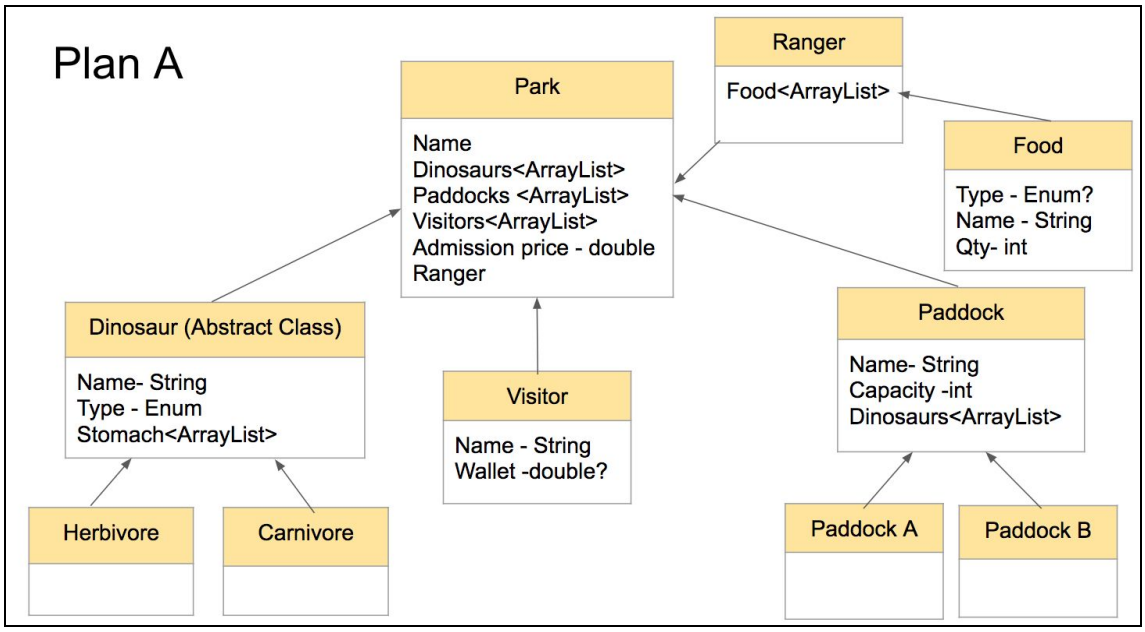
```ruby
# Method for Stock Level
def stock_level()
# If quantity is greater than 8 return green
# If quantity is greater than 3 and quantity
is less than or equal to 7 return amber
# Otherwise return red
  if @quantity >= 8
    return "green"
  elsif @quantity >=3 && @quantity <=7
    return "amber"
  else
    return "red"
  end
end
```

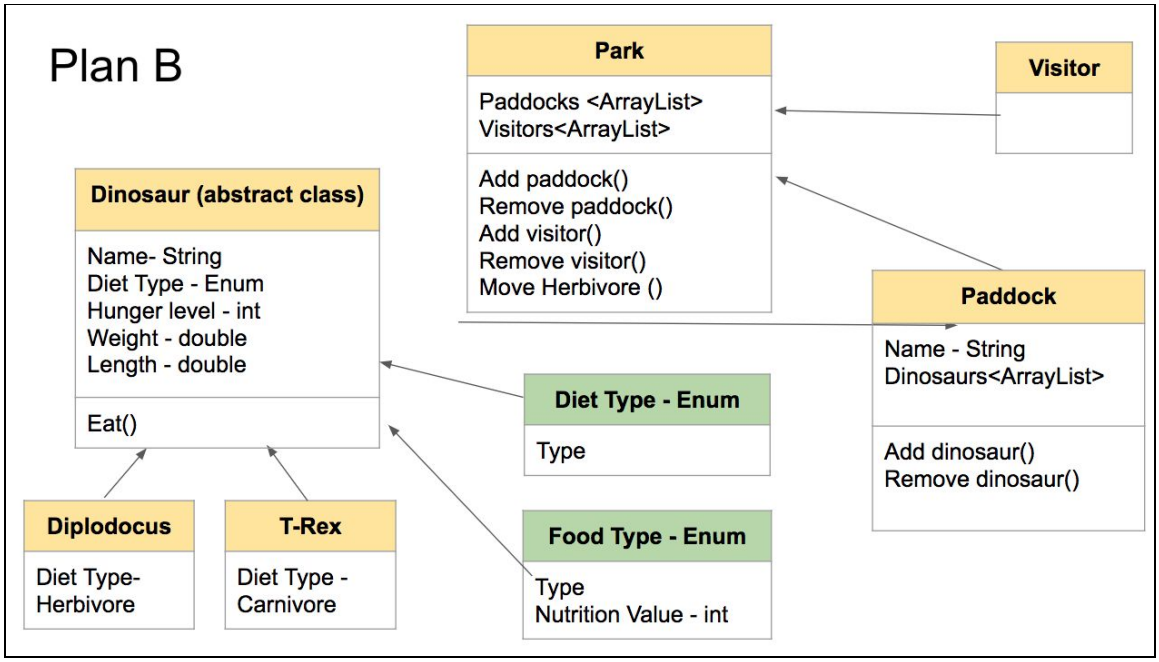
P11 - Screenshot of project on GitHub:



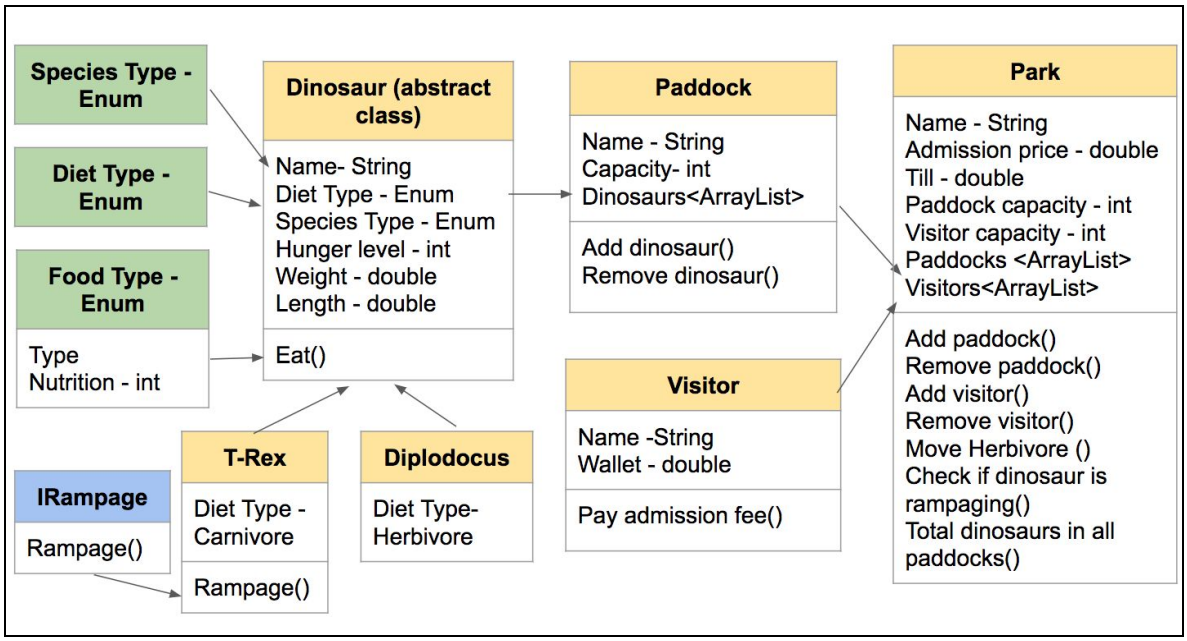
https://github.com/sarahmurphy86/Jurassic_Park_Java_Project

P12 - Planning changes



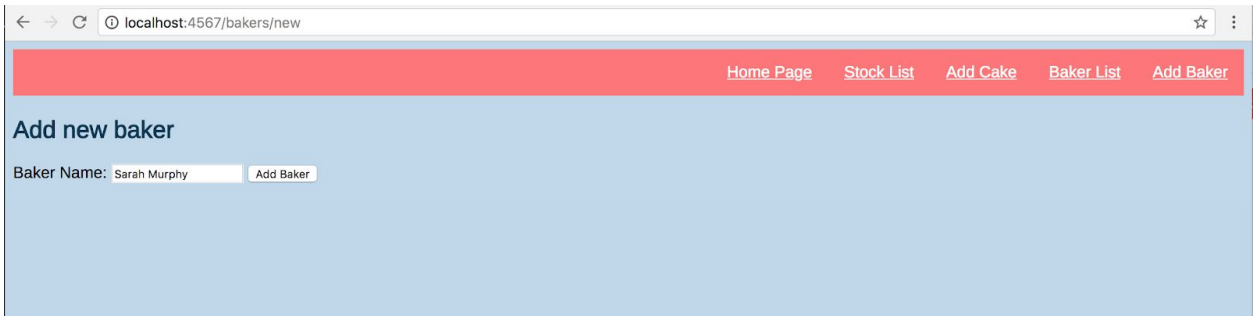


Final Plan:



P13 - user input being processed

User inputting a baker name into the program

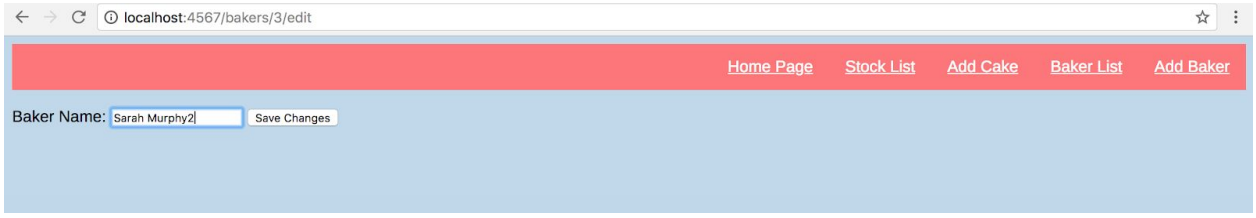


User input saved and displayed in the Baker List

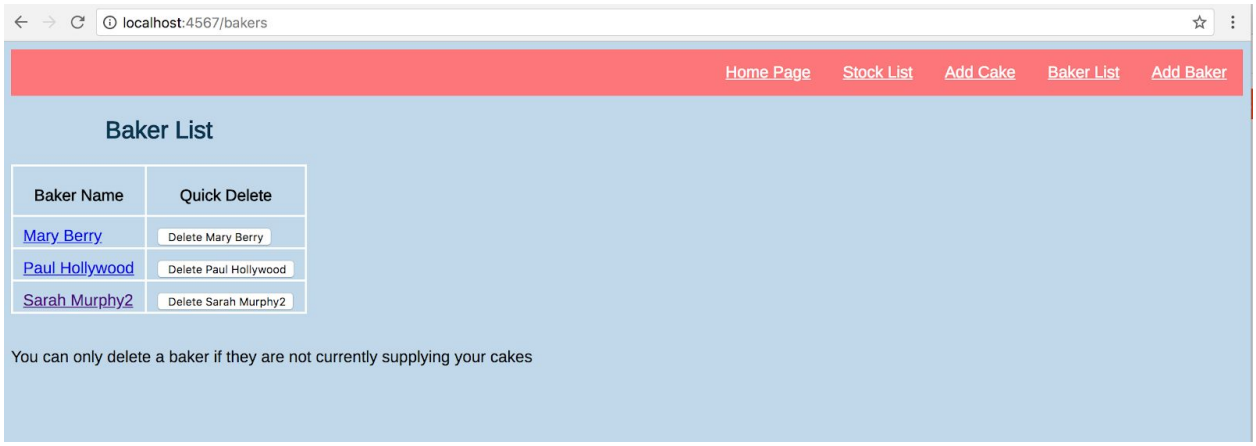
P14 - Interaction with data persistence

Edit existing data

Data being updated in the program



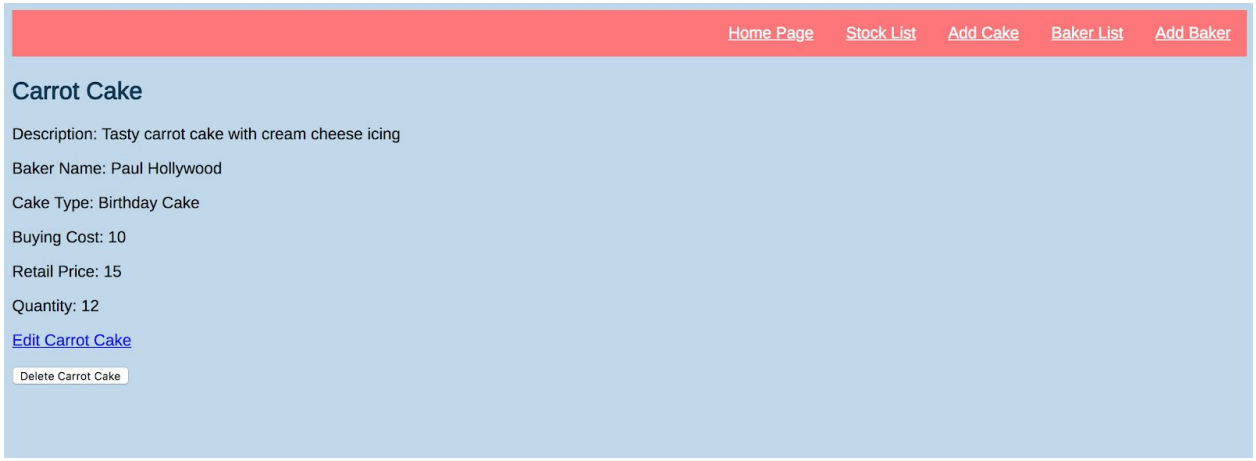
Confirmation of the data being saved - ie new name showing on baker list



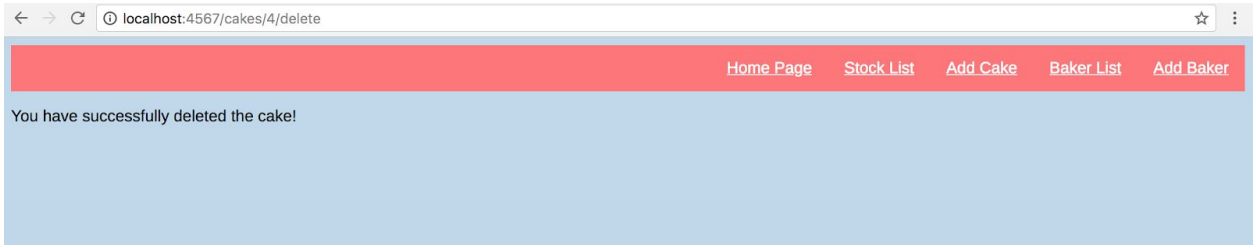
P15 - correct output of results and feedback to user

Inventory screen showing cake list

User selects delete cake:



User gets message to say cake has been deleted:



Cake no longer displayed in inventory:

P16 - Show an API being used in your program

```
Films.prototype.getData = function () {
  const helper = new Helper('https://ghibliapi.herokuapp.com/films')
  helper.get((allFilmData) => {
  // console.log(allFilmData);
  this.filmsData = allFilmData;
  PubSub.publish('Films:film-name', allFilmData);
  });

};
```

P17 - Produce a bug tracking report

Activity	Status	Solution	Final Status
Cake must have a name	You could add a cake without a name	Made this field not Null in SQL table	Fixed
You cannot delete a baker if they still have cakes on the system	Sinatra Crashed	Redirect user to a page to say "you cannot delete this baker as they still have cakes on the system"	Fixed
Stock quantity, buying price and retail cost should be greater than zero	You could add in minus numbers	Changed the minimum quantity to be 0	Fixed
The inventory should highlight stock amber in colour if the quantity was between 3 and 7	Stock with a quantity of 7 was not being highlighted amber	A less than operator was being used, instead of a less than or equal to	Fixed

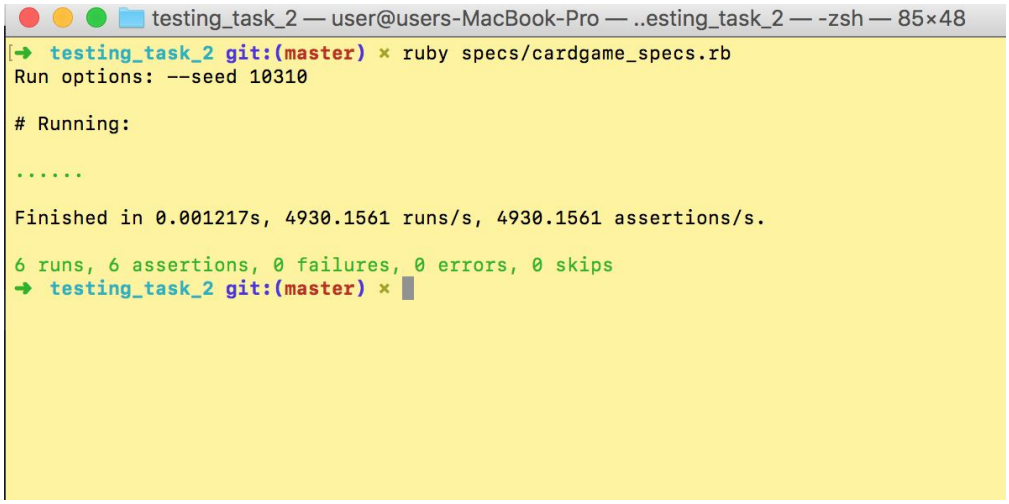
When you go to edit the cake, the existing information should be displayed	Drop down menu on the cake details defaulted to first item in the drop down when you went to edit it	Amended this so that when you went into edit the cake it retained the existing information	Fixed

P18 - Testing Code in your programme

```
  dif highest_card(card1 card2)
  if card1.value > card2.value
    return card.name
  else
    card2
  end
end
end
```

```
➜  testing_task_2 ruby specs/cardgame_specs.rb
specs/cardgame_specs.rb:4:in `require_relative': /Users/user/Desktop/PDA_Static_
and_Dynamic_Task_A/testing_task_2/cardgame.rb:26: syntax error, unexpected keywo
rd_end, expecting end-of-input (SyntaxError)
  end
    ^
      from specs/cardgame_specs.rb:4:in `<main>'
```

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


testing_task_2 — user@users-MacBook-Pro — ..esting_task_2 — -zsh — 85×48
➜ testing_task_2 git:(master) × ruby specs/cardgame_specs.rb
Run options: --seed 10310
Running:
......
Finished in 0.001217s, 4930.1561 runs/s, 4930.1561 assertions/s.
6 runs, 6 assertions, 0 failures, 0 errors, 0 skips
➜ testing_task_2 git:(master) ×