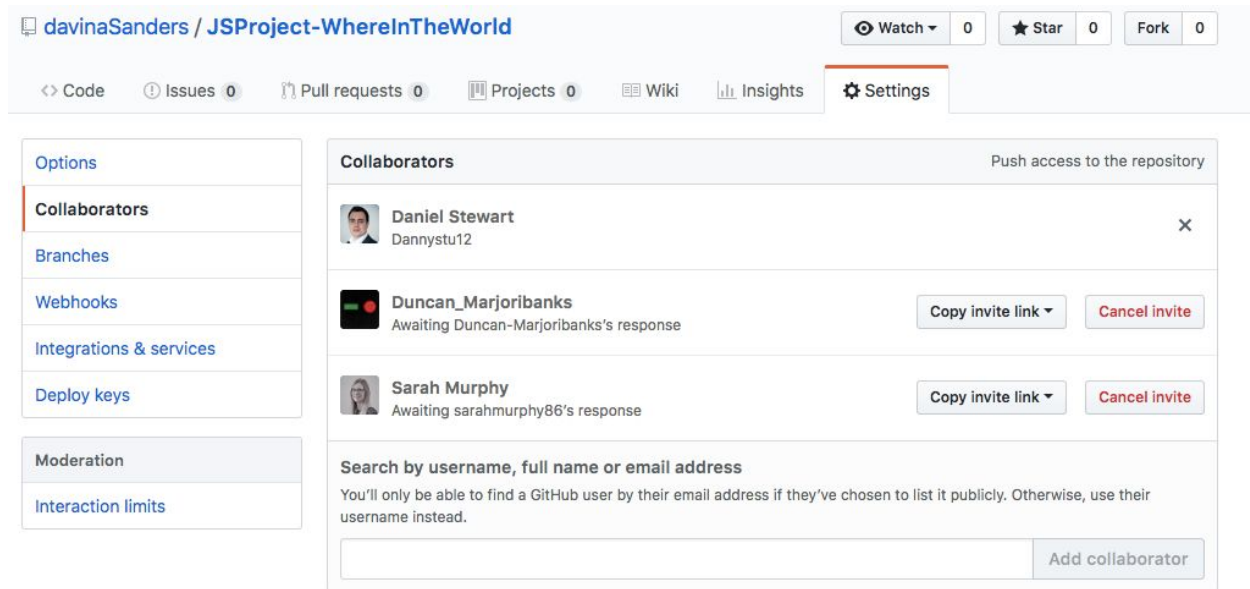


## Project Unit

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

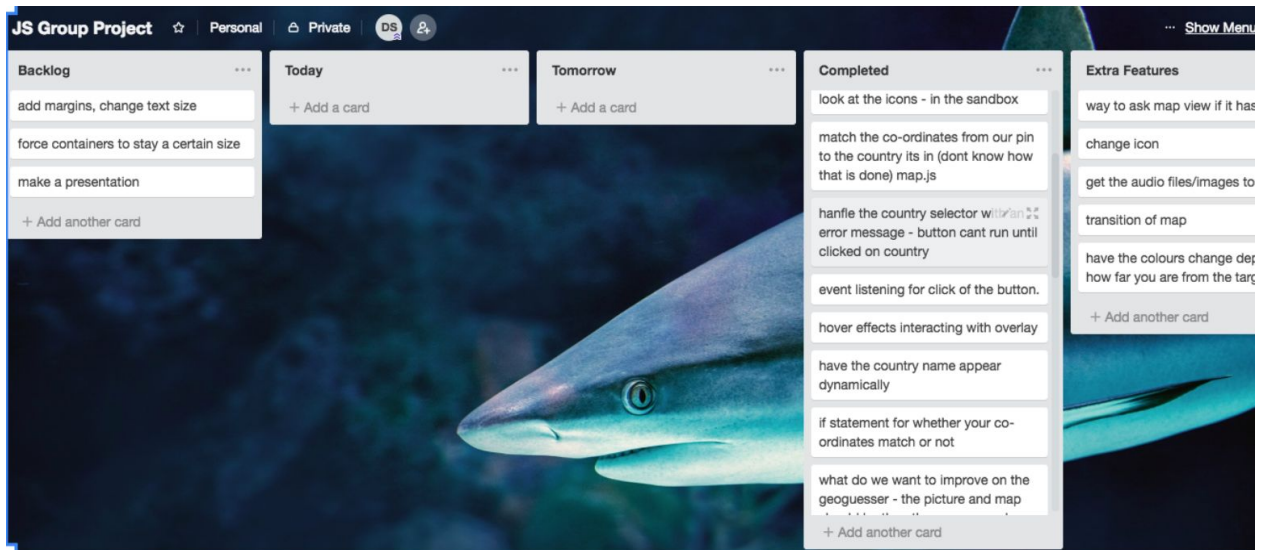
#### MVP

- 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

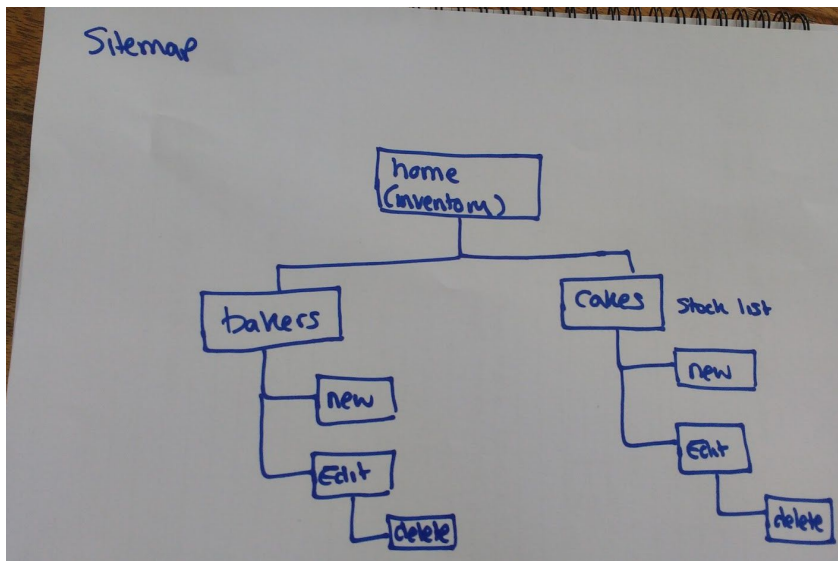


### P4 - Acceptance criteria and test plan

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

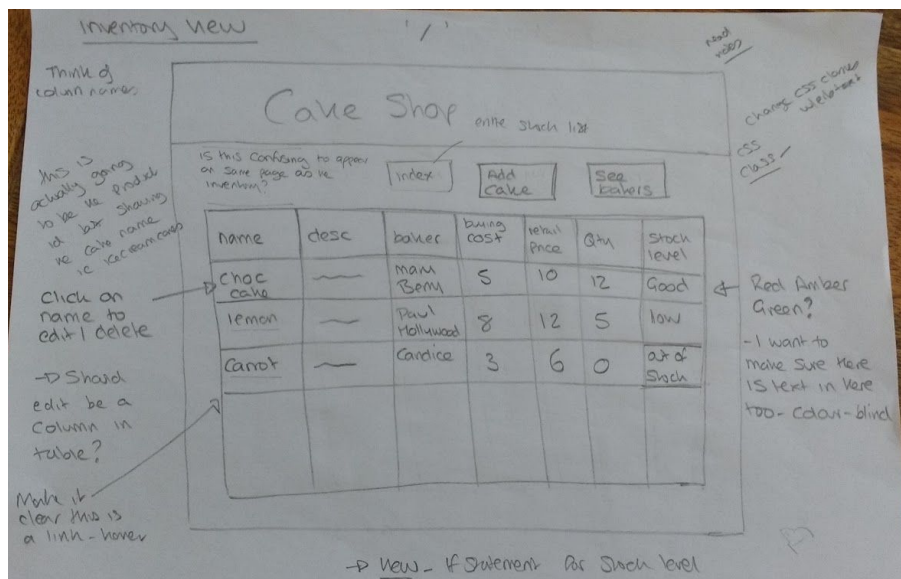
## P5 - Sitemap

For cake shop app:

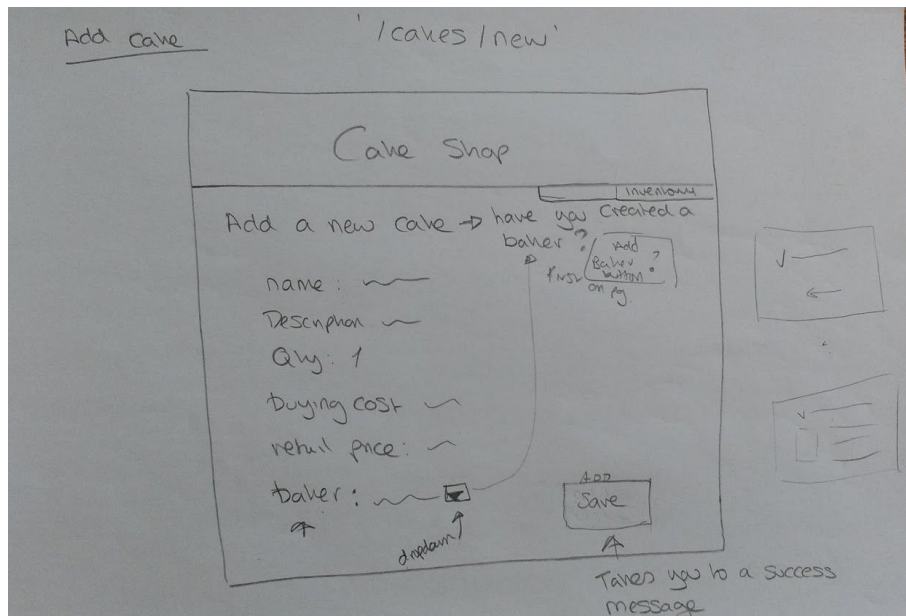


## P6 - Wireframes

Cake shop - inventory view:

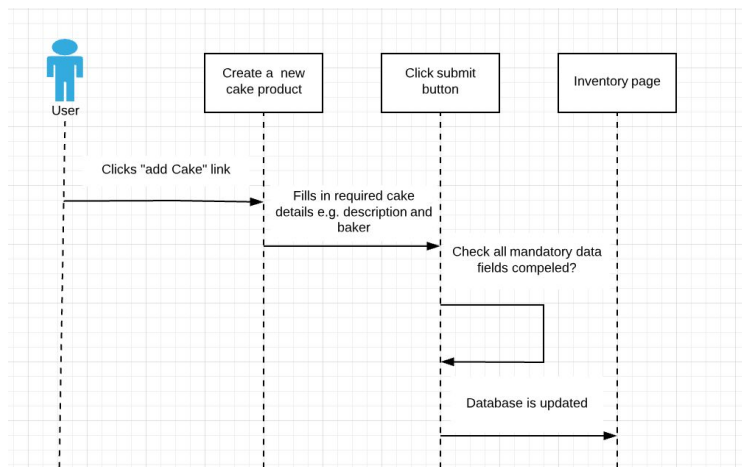


Cake shop - new cake view:

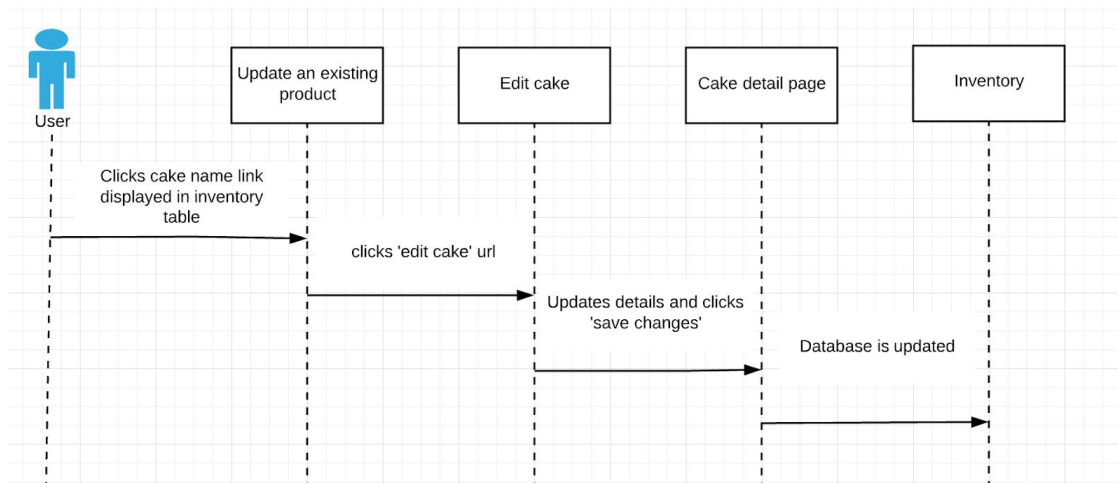


## P7- Two system interaction diagrams

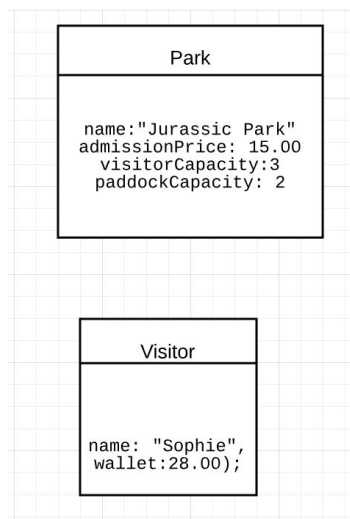
### Adding a cake(product)



## Update a cake(existing product)



## P8- Two object diagrams



## P9 - Two algorithms you have written

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

```
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

```
# 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
```



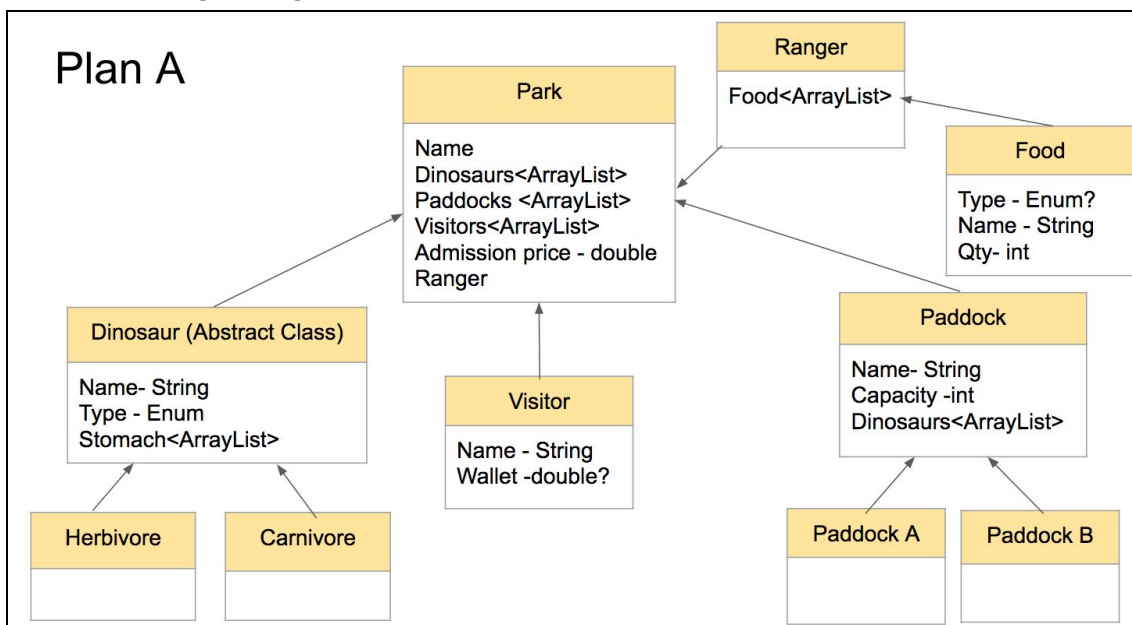
## P11 - Screenshot of project on GitHub:

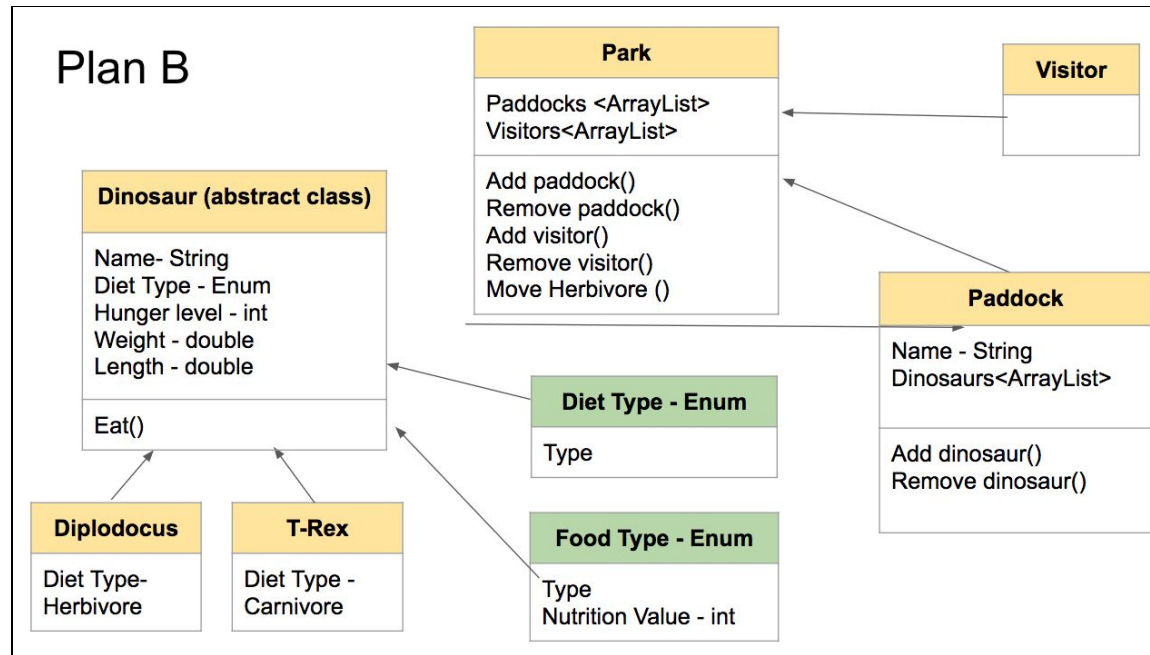
The screenshot shows the GitHub interface for the repository 'Jurassic\_Park\_Java\_Project' by user 'sarahmurphy86'. The repository has 32 commits, 1 branch, 0 releases, and 1 contributor. The 'Code' tab is selected, showing a list of files and their commit history. The files listed are: .gradle (Initial commit, 6 days ago), .idea (code for review, 6 hours ago), gradle/wrapper (Initial commit, 6 days ago), out (refactor park class, 23 hours ago), src (refactor park class, 23 hours ago), build.gradle (Initial commit, 6 days ago), and gradlew (Initial commit, 6 days ago). The repository has no description, website, or topics provided.

File	Commit	Time
.gradle	Initial commit	6 days ago
.idea	code for review	6 hours ago
gradle/wrapper	Initial commit	6 days ago
out	refactor park class	23 hours ago
src	refactor park class	23 hours ago
build.gradle	Initial commit	6 days ago
gradlew	Initial commit	6 days ago

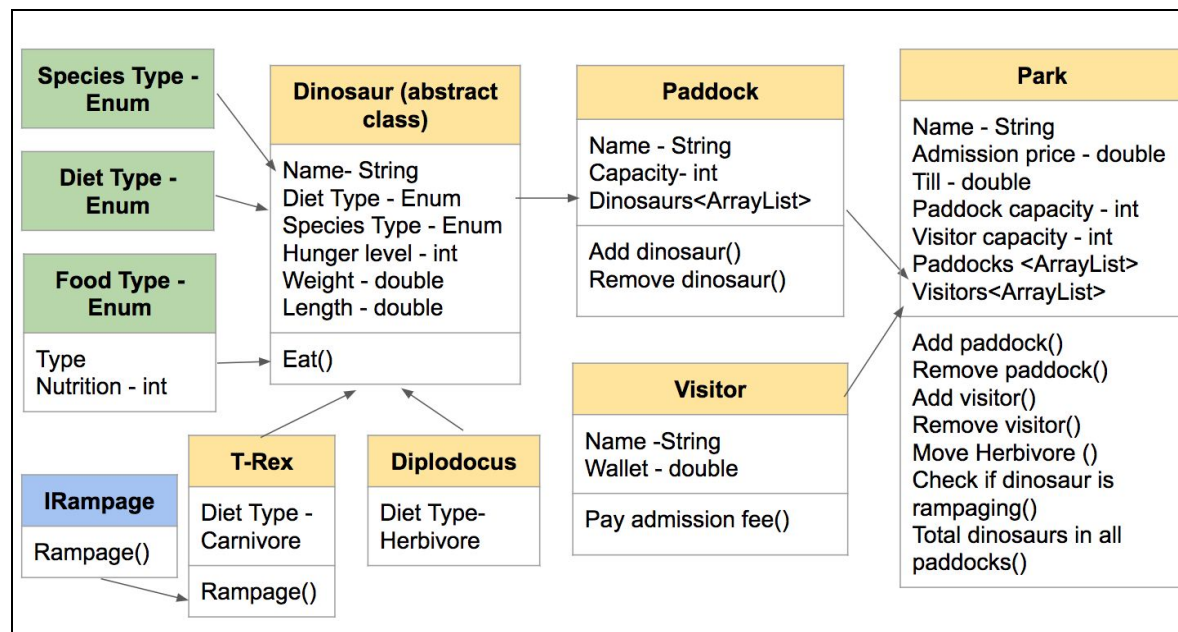
[https://github.com/sarahmurphy86/Jurassic\\_Park\\_Java\\_Project](https://github.com/sarahmurphy86/Jurassic_Park_Java_Project)

## P12 - Planning changes





### Final Plan:



### P13 - user input being processed

User inputting a baker name into the program



localhost:4567/bakers/new

Home Page Stock List Add Cake Baker List Add Baker

### Add new baker

Baker Name:

## User input saved and displayed in the Baker List

localhost:4567/bakers

Home Page Stock List Add Cake Baker List Add Baker

### Baker List

Baker Name	Quick Delete
<a href="#">Mary Berry</a>	<input type="button" value="Delete Mary Berry"/>
<a href="#">Paul Hollywood</a>	<input type="button" value="Delete Paul Hollywood"/>
<a href="#">Sarah Murphy</a>	<input type="button" value="Delete Sarah Murphy"/>

You can only delete a baker if they are not currently supplying your cakes

## P14 - Interaction with data persistence

### Edit existing data

localhost:4567/bakers/3

Home Page Stock List Add Cake Baker List Add Baker

### Sarah Murphy

[Edit Sarah Murphy](#)

### Data being updated in the program

localhost:4567/bakers/3/edit

Home Page Stock List Add Cake Baker List Add Baker

Baker Name:

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

[Home Page](#) [Stock List](#) [Add Cake](#) [Baker List](#) [Add Baker](#)

### Baker List

Baker Name	Quick Delete
<a href="#">Mary Berry</a>	<input type="button" value="Delete Mary Berry"/>
<a href="#">Paul Hollywood</a>	<input type="button" value="Delete Paul Hollywood"/>
<a href="#">Sarah Murphy2</a>	<input type="button" value="Delete Sarah Murphy2"/>

You can only delete a baker if they are not currently supplying your cakes

**P15 - correct output of results and feedback to user**

Inventory screen showing cake list

[Home Page](#) [Stock List](#) [Add Cake](#) [Baker List](#) [Add Baker](#)

### Cake shop inventory

Name	Description	Baker	Cake Type	Buying Cost	Retail Price	Amount in Stock	Markup
<a href="#">Lemon Drizzle</a>	Delicious lemon sponge soaked in zesty lemon sugar	Mary Berry	Classic Sponge	5	8	10	3
<a href="#">Chocolate</a>	Tasty chocolate sponge covered in a thick dark chocolate icing	Paul Hollywood	Birthday Cake	8	10	3	2
<a href="#">Carrot Cake</a>	Tasty carrot cake with cream cheese icing	Paul Hollywood	Birthday Cake	10	15	12	5

User selects delete cake:

[Home Page](#)
[Stock List](#)
[Add Cake](#)
[Baker List](#)
[Add Baker](#)

## Carrot Cake

Description: Tasty carrot cake with cream cheese icing

Baker Name: Paul Hollywood

Cake Type: Birthday Cake

Buying Cost: 10

Retail Price: 15

Quantity: 12

[Edit Carrot Cake](#)

[Delete Carrot Cake](#)

User gets message to say cake has been deleted:

[Home Page](#)
[Stock List](#)
[Add Cake](#)
[Baker List](#)
[Add Baker](#)

You have successfully deleted the cake!

Cake no longer displayed in inventory:

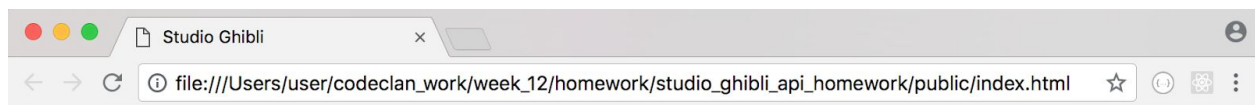
[Home Page](#)
[Stock List](#)
[Add Cake](#)
[Baker List](#)
[Add Baker](#)

### Cake shop inventory

Name	Description	Baker	Cake Type	Buying Cost	Retail Price	Amount in Stock	Markup
<a href="#">Lemon Drizzle</a>	Delicious lemon sponge soaked in zesty lemon sugar	Mary Berry	Classic Sponge	5	8	10	3
<a href="#">Chocolate</a>	Tasty chocolate sponge covered in a thick dark chocolate icing	Paul Hollywood	Birthday Cake	8	10	3	2

**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);
  });
};
```



## Welcome to Studio Ghibli

Select a film:

### Grave of the Fireflies

In the latter part of World War II, a boy and his sister, orphaned when their mother is killed in the firebombing of Tokyo, are left to survive on their own in what remains of civilian life in Japan. The plot follows this boy and his sister as they do their best to survive in the Japanese countryside, battling hunger, prejudice, and pride in their own quiet, personal battle.

### 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

```

it('should have a collection of films', function () {
  const actual = cinema.films;
  assert.strictEqual(actual, film);
});

it('should be able to get a list of film titles', function () {
  const actual = cinema.getAllFilmsByTitle();
  const filmsTitleArray = ['Moonlight', 'Blade Runner 2049', 'Dunkirk', 'Black Panther', 'T2 Trainspotting'];
  assert.strictEqual(actual, filmsTitleArray);
});

it('should be able to find a film by title', function () {
  const actual = cinema.getFilmByTitle(Moonlight);
  const film = moonlight;
  assert.strictEqual( actual, film);
}

```

```
specs — user@users-MacBook-Pro — ../Cinema/specs — zsh — 109x48
→ specs git:(master) * npm test

> lab_enumeration_callbacks_films_end@1.0.0 test /Users/user/codeclan_work/week_11/day_3/enumeration_lab/Lab-
Cinema
> mocha specs

Cinema
  1) should have a collection of films
  2) should be able to get a list of film titles
  3) should be able to find a film by title

Film
  ✓ should have a title
  ✓ should have a genre
  ✓ should have a year
  ✓ should have a length

4 passing (12ms)
3 failing

1) Cinema
   should have a collection of films:
     ReferenceError: film is not defined
       at Context.<anonymous> (specs/cinema_spec.js:20:36)

2) Cinema
   should be able to get a list of film titles:

     AssertionError [ERR_ASSERTION]: [ 'Moonlight',
'Blade Runner 2049',
'Dunkirk',
'Black Panther',
'T2 Trainspotting' ] === [ 'Moonlight',
'Blade Runner 2049',
'Dunkirk',
'Black Panther',
'T2 Trainspotting' ]
       + expected - actual

       at Context.<anonymous> (specs/cinema_spec.js:26:12)

3) Cinema
   should be able to find a film by title:
     ReferenceError: Moonlight is not defined
```



```

it('should have a collection of films', function () {
  const actual = cinema.films;
  assert.deepStrictEqual(actual, films);
});

it('should be able to get a list of film titles', function () {
  const actual = cinema.getAllFilmsByTitle();
  const filmsTitleArray = ['Moonlight', 'Blade Runner 2049', 'Dunkirk', 'Black Panther', 'T2 Trainspotting'];
  assert.deepStrictEqual(actual, filmsTitleArray);
});

it('should be able to find a film by title', function () {
  const actual = cinema.getFilmByTitle('Moonlight');
  const film = moonlight;
  assert.deepStrictEqual(actual, film);
});

```

→ specs git:(master) × npm test

```

> lab_enumeration_callbacks_films_end@1.0.0 test /Users/user/codeclan_work/week_11/day_3/enumeration_lab/Lab
Cinema
> mocha specs

```

#### Cinema

- ✓ should have a collection of films
- ✓ should be able to get a list of film titles
- ✓ should be able to find a film by title

#### Film

- ✓ should have a title
- ✓ should have a genre
- ✓ should have a year
- ✓ should have a length

7 passing (9ms)