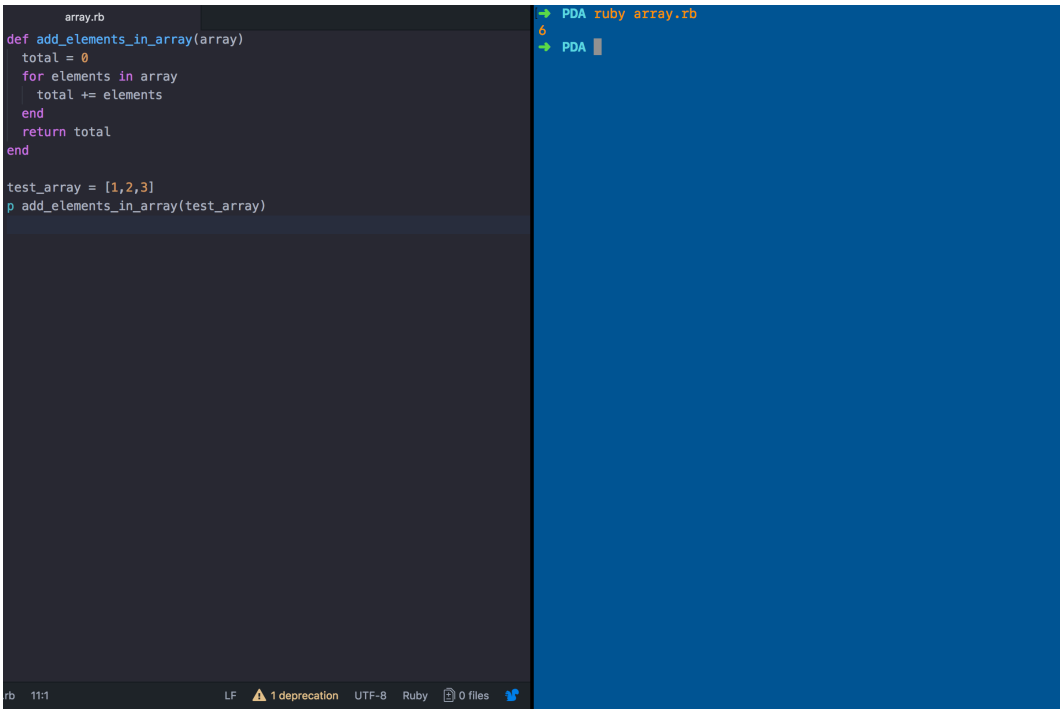
Evidence Gathering Document for SQA Level 8 Professional Developer Award.

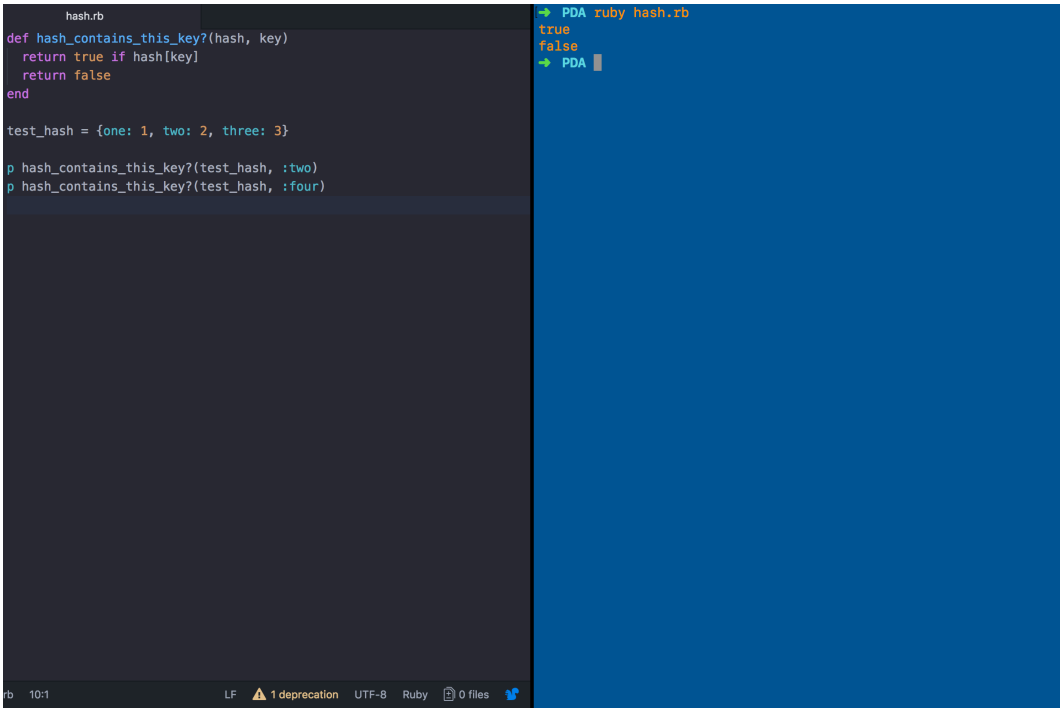
Week 2

Unit	Ref	Evidence
I&T	I.T.5	Demonstrate the use of an array in a program. Take screenshots of: *An array in a program *A function that uses the array *The result of the function running



The function to add elements in an array is defined and called in array.rb on the left. On the right you can see the result of running the program in the terminal.

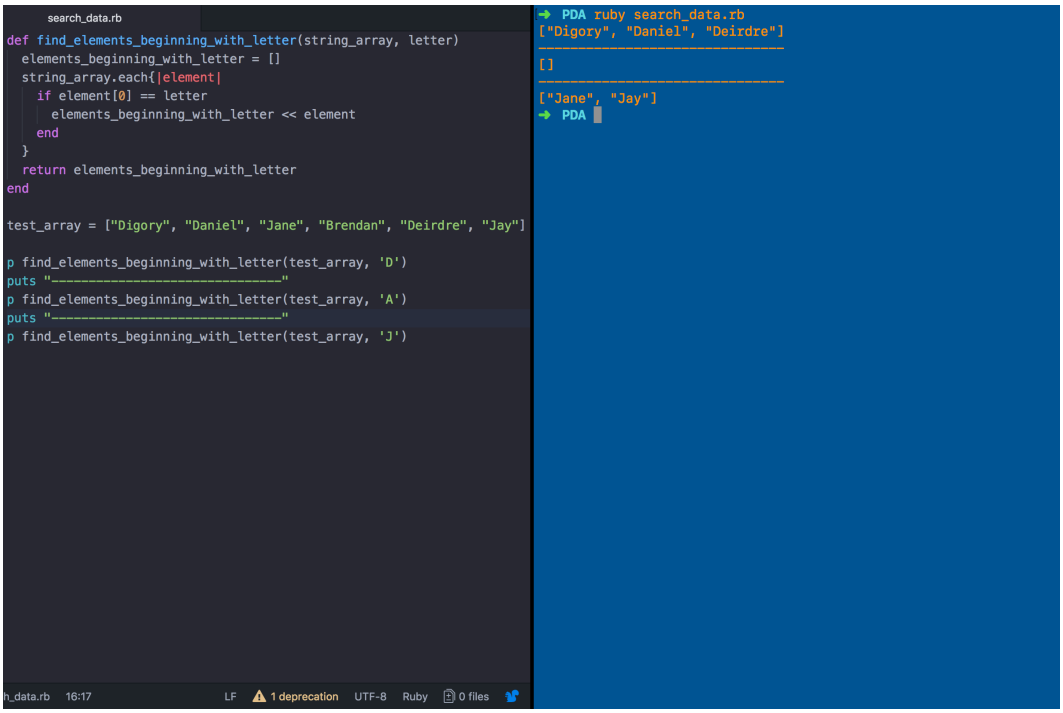
Unit	Ref	Evidence
I&T	I.T.6	Demonstrate the use of a hash in a program. Take screenshots of: *A hash in a program *A function that uses the hash *The result of the function running



The function defined in hash.rb checks to see if a specific key is present in a hash, returning a boolean true or false depending on the result.

Week 3

Unit	Ref	Evidence
I&T	I.T.3	Demonstrate searching data in a program. Take screenshots of: *Function that searches data *The result of the function running



The function defined in search_data.rb searches through each element of an array of strings. If the element's first letter is the same as the parameter letter, it is added to another array. This array is then returned.

Unit	Ref	Evidence
I&T	I.T.4	Demonstrate sorting data in a program. Take screenshots of: *Function that sorts data *The result of the function running

app.js

```
1  sortByLastLetterDescending = function(stringArray){
2      stringArray.sort((element1, element2)=>{
3          if(element1[element1.length-1] > element2[element2.length-1]){
4              return -1;
5          } else{
6              return 1;
7          }
8      });
9  }
10
11 const stringArray = ["is", "Digory", "cool"];
12 sortByLastLetterDescending(stringArray);
13 console.log(stringArray);
14
15
```

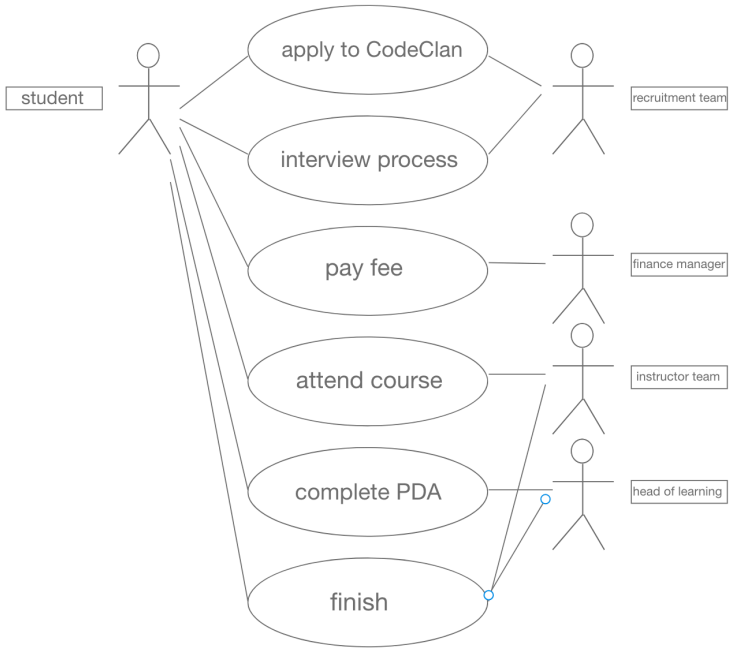
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
→  project_victual git:(master) ✗ node client/src/app.js
[ 'Digory', 'is', 'cool' ]
→  project_victual git:(master) ✗
```

The function defined in app.js accepts a parameter string array. This array is then sorted by last letter descending (i.e. word with last letter ‘z’ is before a word with last letter ‘a’)

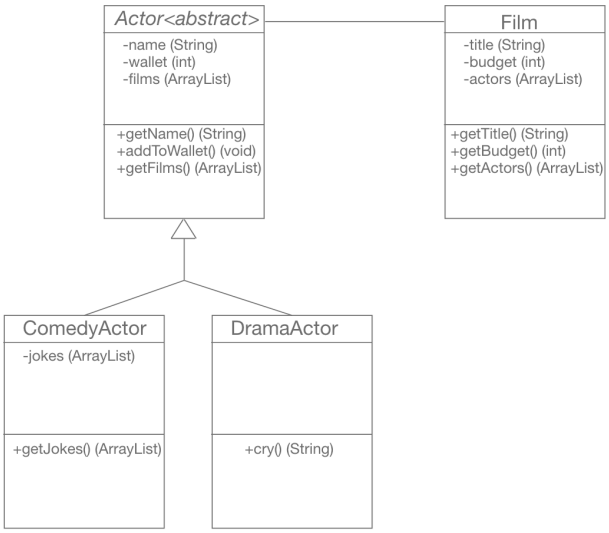
Week 5

Unit	Ref	Evidence
A&D	A.D.1	A Use Case Diagram



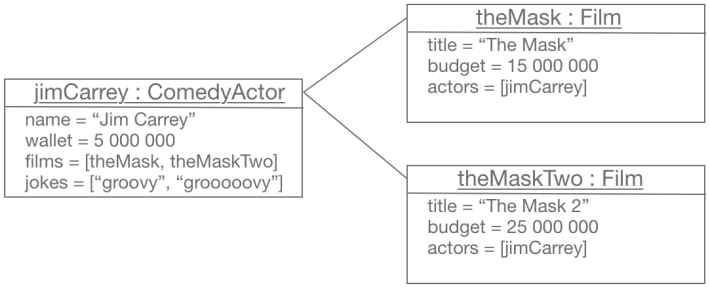
The diagram depicts the system in which a student completes the CodeClan Software Development course. The actors are the student, recruitment team, finance manager, instructor team and head of learning.

Unit	Ref	Evidence
A&D	A.D.2	A Class Diagram



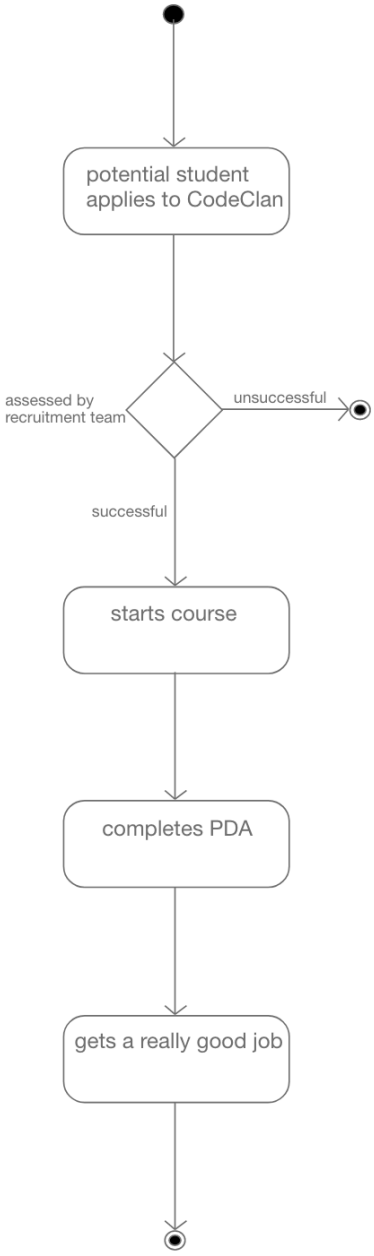
The above represents the relationships between four classes and the methods and attributes of each class. Actor is abstract and is inherited by ComedyActor and DramaActor. Film and Actor are aware of each other. The + and - signs show whether a method or attribute is public (+) or private (-).

Unit	Ref	Evidence
A&D	A.D.3	An Object Diagram



The above shows three objects created from the classes in the previous example.

Unit	Ref	Evidence
A&D	A.D.4	An Activity Diagram

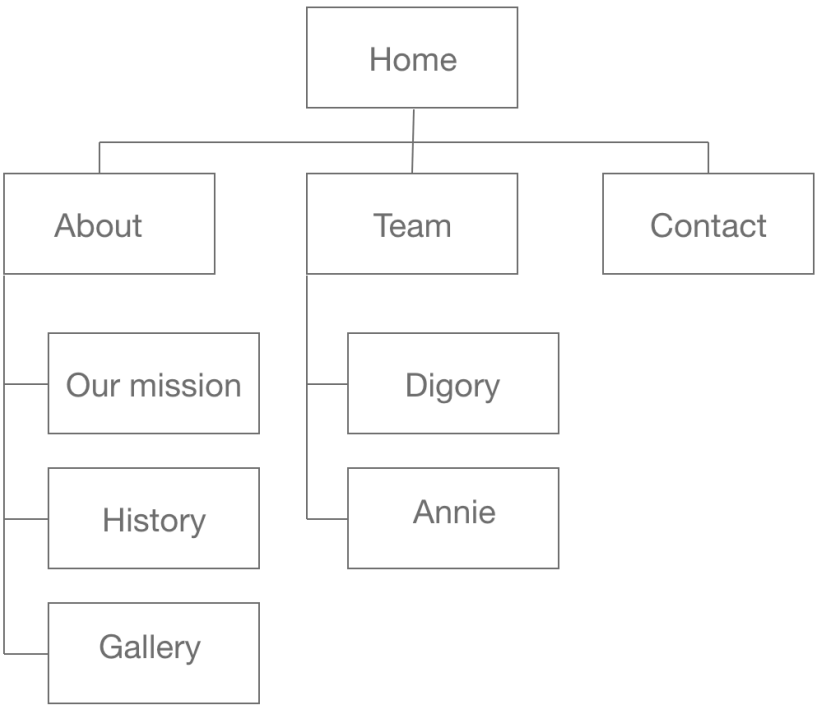


The above models the system whereby a student attends the CodeClan course. The diamond shaped box represents a condition check to determine whether they are successful in their application.

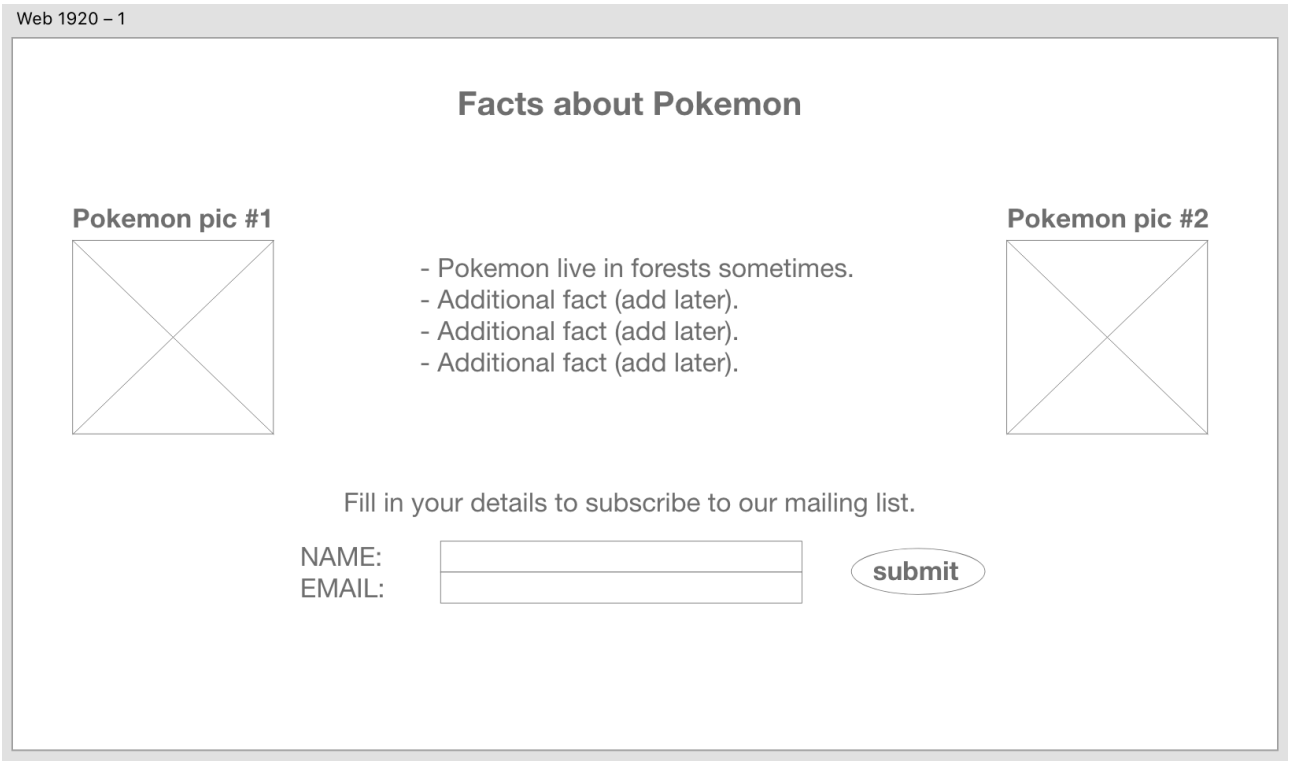
Unit	Ref	Evidence
A&D	A.D.6	Produce an Implementations Constraints plan detailing the following factors: *Hardware and software platforms *Performance requirements *Persistent storage and transactions *Usability *Budgets *Time

CONSTRAINT	POSSIBLE EFFECT OF CONSTRAINT ON PRODUCT	SOLUTION
Hardware and software platforms	The app is designed for viewing on a computer monitor and not for mobiles or tablets.	Use flexbox to make the app able to resize.
Performance requirements	It requires a high performance graphics card in order to be viewed.	Add in a lower resolution option.
Persistent storage and transactions	Storage of data remotely can increase loading times.	Cache data locally where possible
Usability	Customers using screen readers or other assistive technologies may find aspects of the app difficult to navigate.	Ensure that the app conforms to WebAIM standards.
Budgets	Our budget for external APIs is 0.	Research APIs that are free to use.
Time limitations	We have 5 days to complete the project.	Implement Agile practices within our team.

Unit	Ref	Evidence
P	P.5	User Site Map



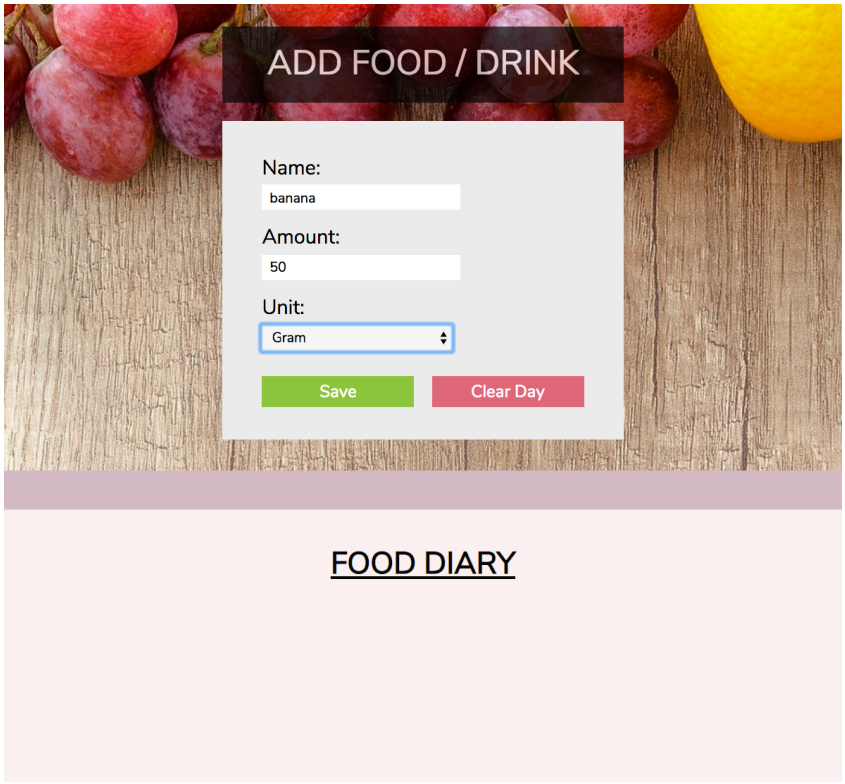
Unit	Ref	Evidence
P	P.6	2 Wireframe Diagrams

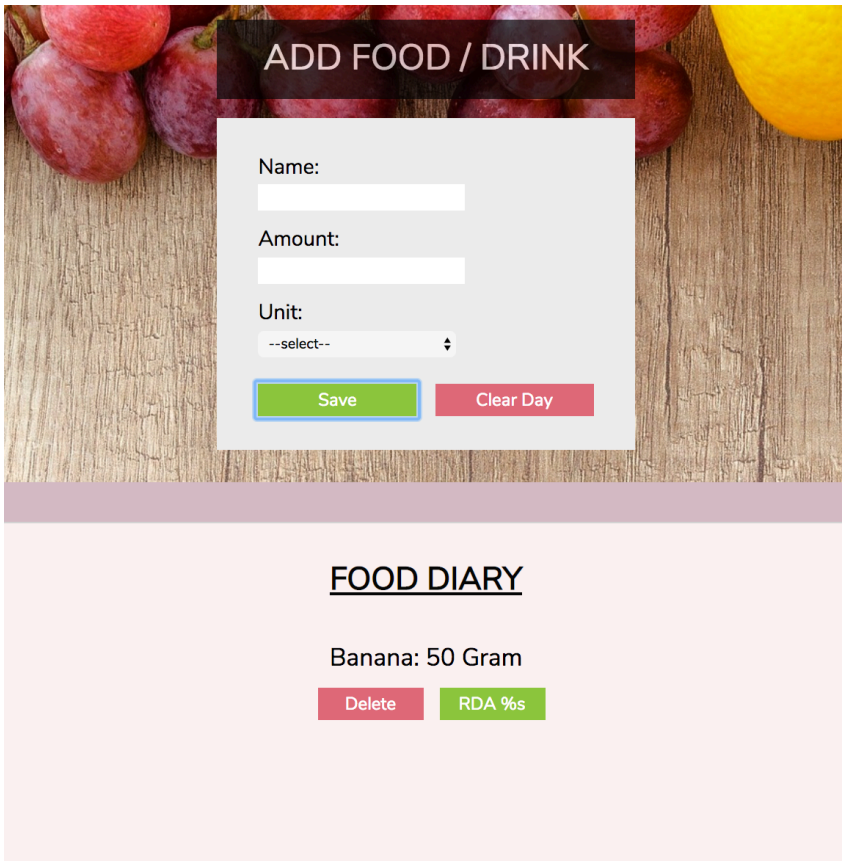


Unit	Ref	Evidence
P	P.10	Example of Pseudocode used for a method

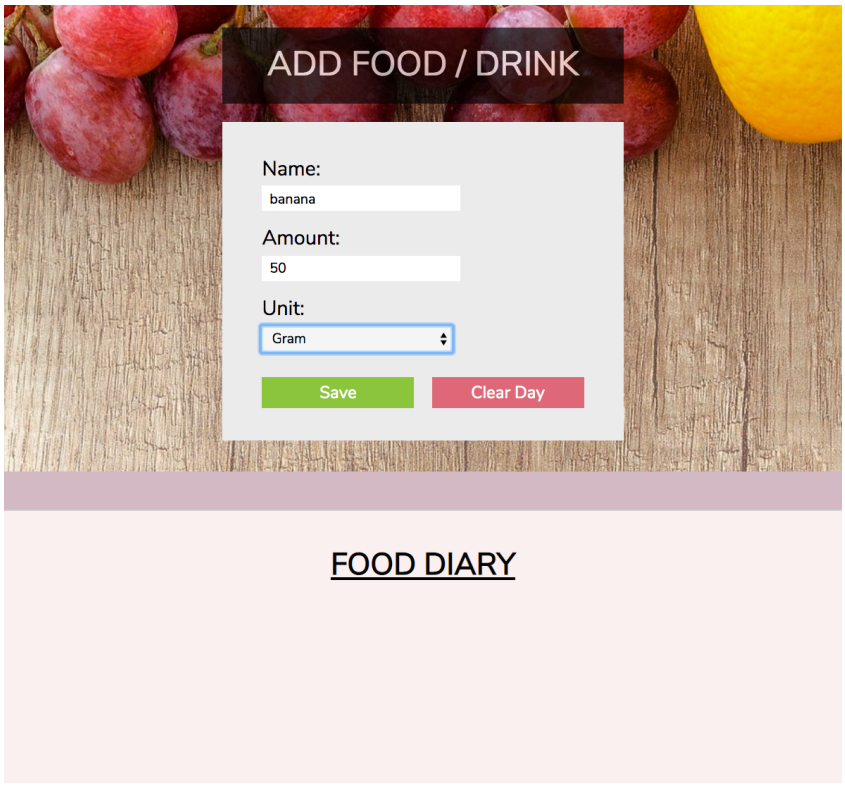
```
// Method for generating random Pseudocode for people doing the PDA.
App.prototype.pseudoCodeGenerator = function(){
    // Create a random number.
    // Use this to select an element from an array of words.
    // Loop through this again until a reasonable number of words is generated.
    // Check this combination of words hasn't been used before to prevent plagiarism.
    // Return the list of words as a string.
}
```

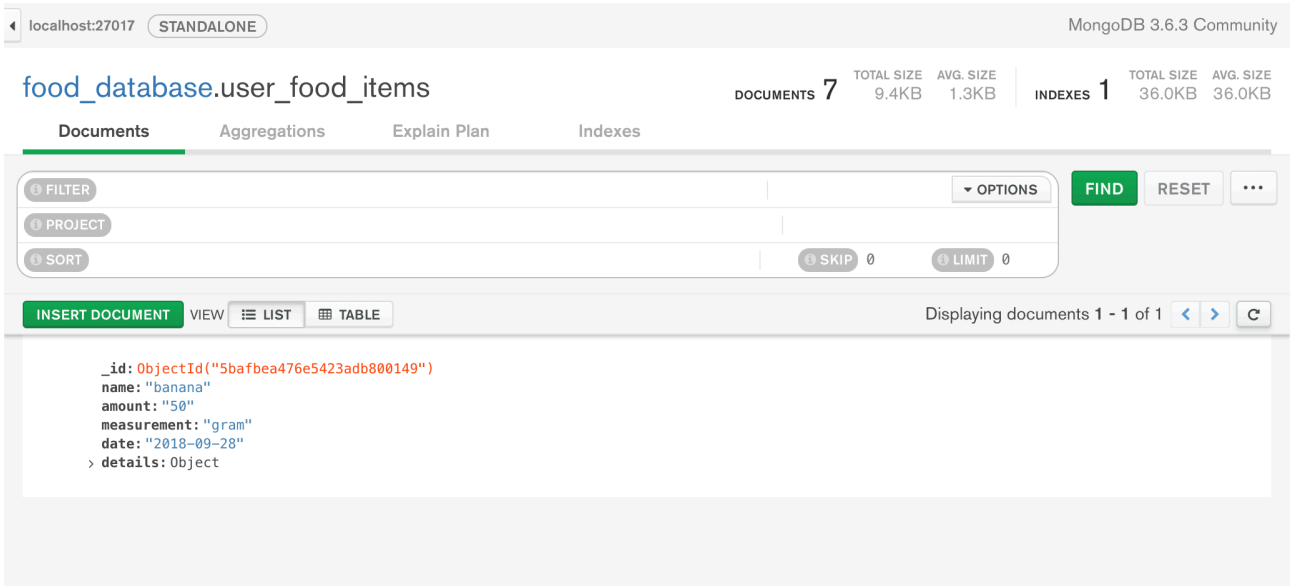
Unit	Ref	Evidence
P	P.13	Show user input being processed according to design requirements. Take a screenshot of: * The user inputting something into your program * The user input being saved or used in some way





Unit	Ref	Evidence
P	P.14	Show an interaction with data persistence. Take a screenshot of: * Data being inputted into your program * Confirmation of the data being saved





Unit	Ref	Evidence
P	P.15	Show the correct output of results and feedback to user. Take a screenshot of: * The user requesting information or an action to be performed * The user request being processed correctly and demonstrated in the program

RECIPES HIGH IN VITAMIN D

RICH HOMEMADE RICOTTA

SOLE MEUNIÈRE

GRILLED RED SNAPPER WITH ROASTED MIXED PEPPERS AND ASPARAGUS

DIETARY REQUIREMENTS

- ○ No Dietary Requirements
- ○ No Alcohol
- ◉ Vegan
- ○ Vegetarian
- ○ No Wheat

Enter any ingredient you do NOT want in the recipe. You can enter multiple ingredients by separating with a comma (eg. chickpea, garlic):

onion

Filter

The initial recipes suggested are at the top. The user then filters these.

RECIPES HIGH IN VITAMIN D

SALTED, SEEDY CHOCOLATE BARK RECIPES

MANGO CHILI SMOOTHIE RECIPE

BITTERSWEET GRANOLA RECIPES

DIETARY REQUIREMENTS

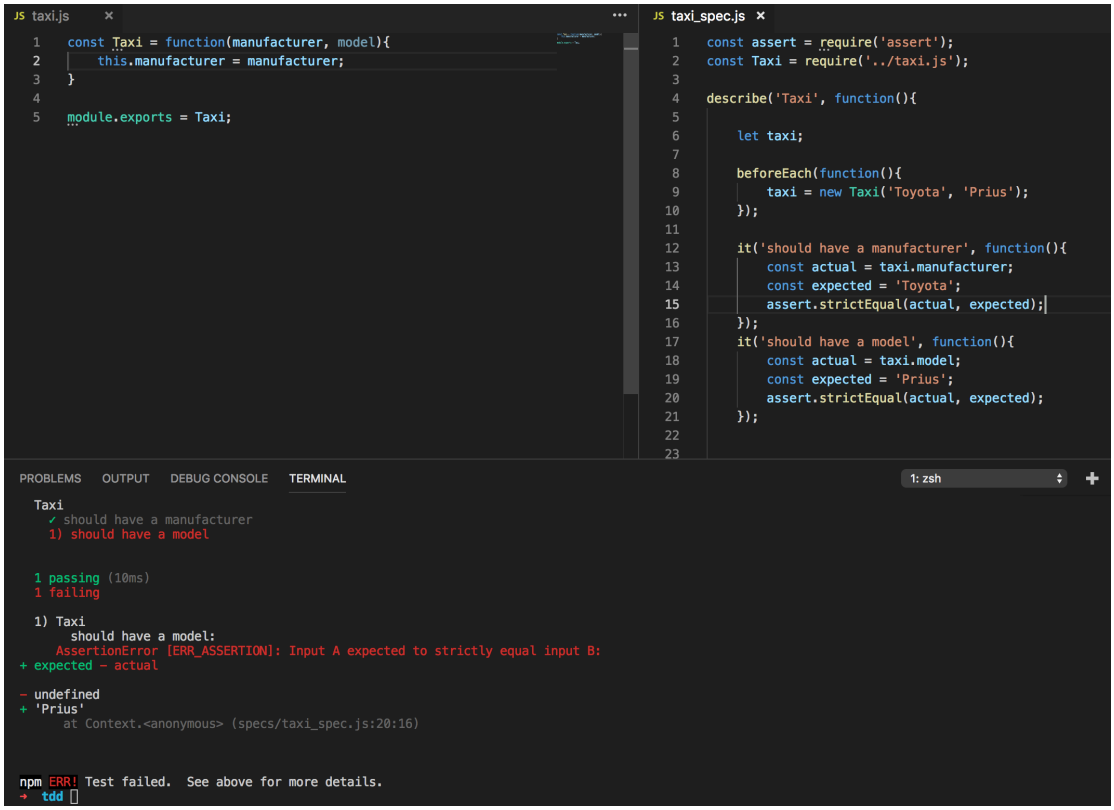
- ◉ No Dietary Requirements
- ○ No Alcohol
- ○ Vegan
- ○ Vegetarian
- ○ No Wheat

Enter any ingredient you do NOT want in the recipe. You can enter multiple ingredients by separating with a comma (eg. chickpea, garlic):

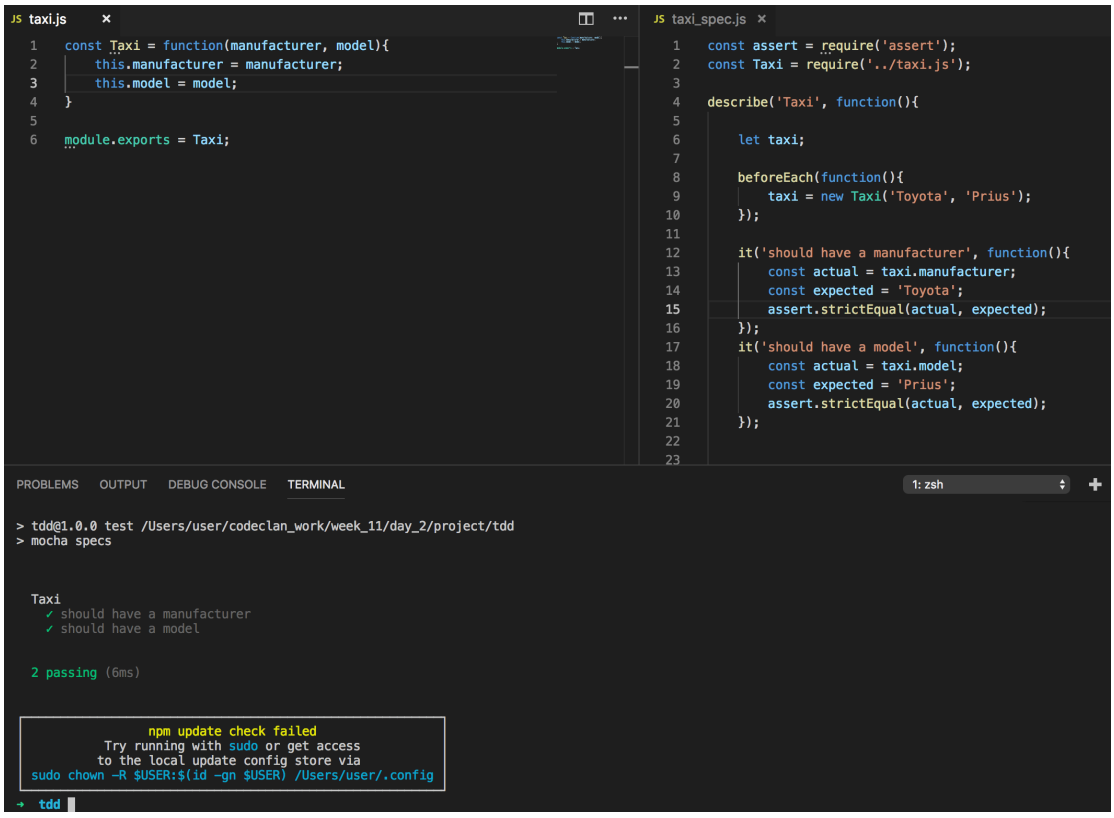
Enter ingredients...

Filter

Unit	Ref	Evidence
P	P.18	Demonstrate testing in your program. Take screenshots of: * Example of test code * The test code failing to pass * Example of the test code once errors have been corrected * The test code passing



One of the tests fail because the model is not being assigned to anything in the Taxi class.



Unit	Ref	Evidence
I&T	I.T.7	The use of Polymorphism in a program and what it is doing.

Actor.java

```java
package models;

public abstract class Actor {
    protected String name;
    protected FilmGenreType genreOfSpeciality;

    public Actor() {
    }

    public Actor(String name, FilmGenreType genreOfSpeciality) {
        this.name = name;
        this.genreOfSpeciality = genreOfSpeciality;
    }

    public FilmGenreType getGenreOfSpeciality(){
        return genreOfSpeciality;
    }

    public abstract String audition();
```

DramaActor.java

```java
package models;

public class DramaActor extends Actor{

    public DramaActor(){}

    public DramaActor(String name) {
        super(name, FilmGenreType.DRAMA);
    }

    public String audition() {
        return "I am a really serious actor guys.";
    }
}
```

Actor › getGenreOfSpeciality()

ComedyActor.java

```java
package models;

public class ComedyActor extends Actor{

    public ComedyActor(){}

    public ComedyActor(String name) {
        super(name, FilmGenreType.COMEDY);
    }

    @Override
    public String audition() {
        return "Hey guys, wanna hear a joke?";
    }

    public String tellAJoke() {
        return "So two whales are sitting in a bar, one of them says WOOOIIIIOIOIOOOOOWOIWWOOOOOOOIW" +
                "WWWWWWWOOOOOOOOOOOOOOOOOOIIOIOIOWWWWOWOWOWOWOWOIOIOIIIIIIIIIIIIOOOOWWWWWWWWWWWWWOWOW" +
                "WWWOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO. The other one says, Frank, you're drunk.";
    }
}
```

The Actor parent class is defined and ComedyActor and DramaActor extend from this. ComedyActor has an additional method tellAJoke().

Runner.java

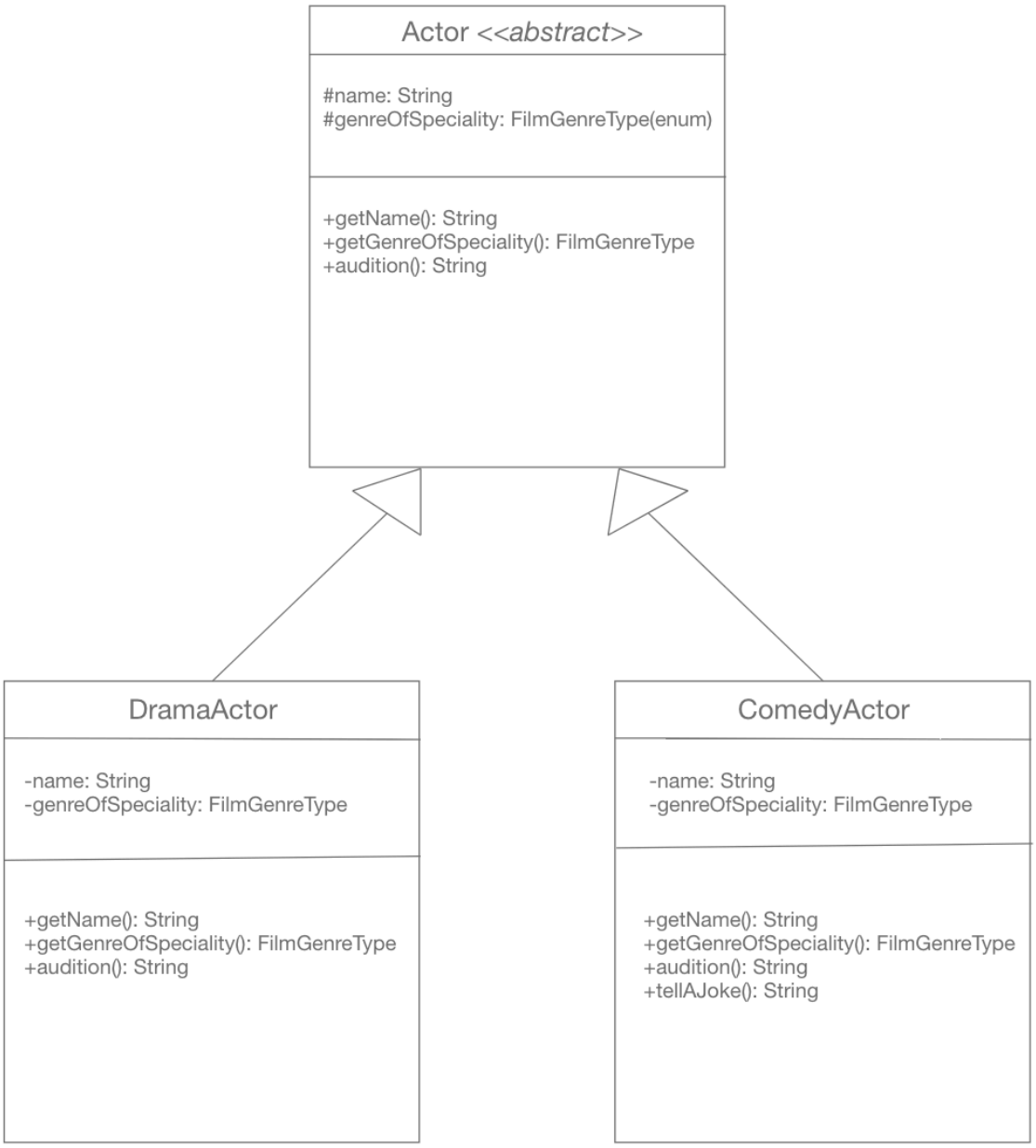
```java
import models.Actor;
import models.ComedyActor;
import models.DramaActor;
import models.FilmGenreType;

import java.util.ArrayList;
import java.util.List;

public class Runner {
    public static void main(String[] args) {

        DramaActor seriousMike = new DramaActor( name: "Mike the Spike");
        ComedyActor comedyJane = new ComedyActor( name: "Jane");

        List<Actor> actors = new ArrayList<Actor>();

        actors.add(seriousMike);
        actors.add(comedyJane);

        for(Actor element : actors){
            System.out.println(element.audition()+"\n");
            if(element.getGenreOfSpeciality() == FilmGenreType.COMEDY){
                System.out.println(((ComedyActor) element).tellAJoke());
            }
        }
    }
}
```

Run: Runner

```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...
I am a really serious actor guys.

Hey guys, wanna hear a joke?

So two whales are sitting in a bar, one of them says
 WOOOIIIIOIOIOOOOOWOIWWOOOOOOOIWWWWWWWWOOOOOOOOOOOOOOOOOOOOOIIOIOIOWWWWOWOWOWOWOWOIOIOIIIIIIIIIIIIOOOOWWWWWWWWWWWWWOWOWWWWWOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO.
 The other one says, Frank, you're drunk.
```

We go through the Actor ArrayList and make them audition. If they are a ComedyActor they are cast from Actor to ComedyActor to allow them to carry out the tellAJoke() method

Unit	Ref	Evidence
A&D	A.D.5	An Inheritance Diagram



Unit	Ref	Evidence
I&T	I.T.1	The use of Encapsulation in a program and what it is doing.

```java
public class Visitor {
    private int age;
    private int heightInCm;
    private double money;

    public Visitor(int age, int heightInCm, double money) {
        this.age = age;
        this.heightInCm = heightInCm;
        this.money = money;
    }

    public int getAge() {
        return age;
    }

    public int getHeightInCm() {
        return heightInCm;
    }

    public double getMoney() {
        return money;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public void setHeightInCm(int heightInCm) {
        this.heightInCm = heightInCm;
    }

    public void setMoney(double money) {
        this.money = money;
    }

}
```

Using encapsulation, the variables: age, heightInCm and money are only available to be read or modified through the getter and setter methods.

Unit	Ref	Evidence
I&T	I.T.2	Take a screenshot of the use of Inheritance in a program. Take screenshots of: *A Class *A Class that inherits from the previous class *An Object in the inherited class *A Method that uses the information inherited from another class.

Actor.java

```java
package models;

public abstract class Actor {
    protected String name;
    protected FilmGenreType genreOfSpeciality;

    public Actor(String name, FilmGenreType genreOfSpeciality) {
        this.name = name;
        this.genreOfSpeciality = genreOfSpeciality;
    }

    public FilmGenreType getGenreOfSpeciality(){
        return genreOfSpeciality;
    }

    public abstract String audition();

}
```

ComedyActor.java

```java
package models;

public class ComedyActor extends Actor{

    public ComedyActor(String name) {
        super(name, FilmGenreType.COMEDY);
    }

    @Override
    public String audition() {
        return "Hey guys, my genre of speciality is " + super.getGenreOfSpeciality() + " wanna hear a joke?";
    }

```

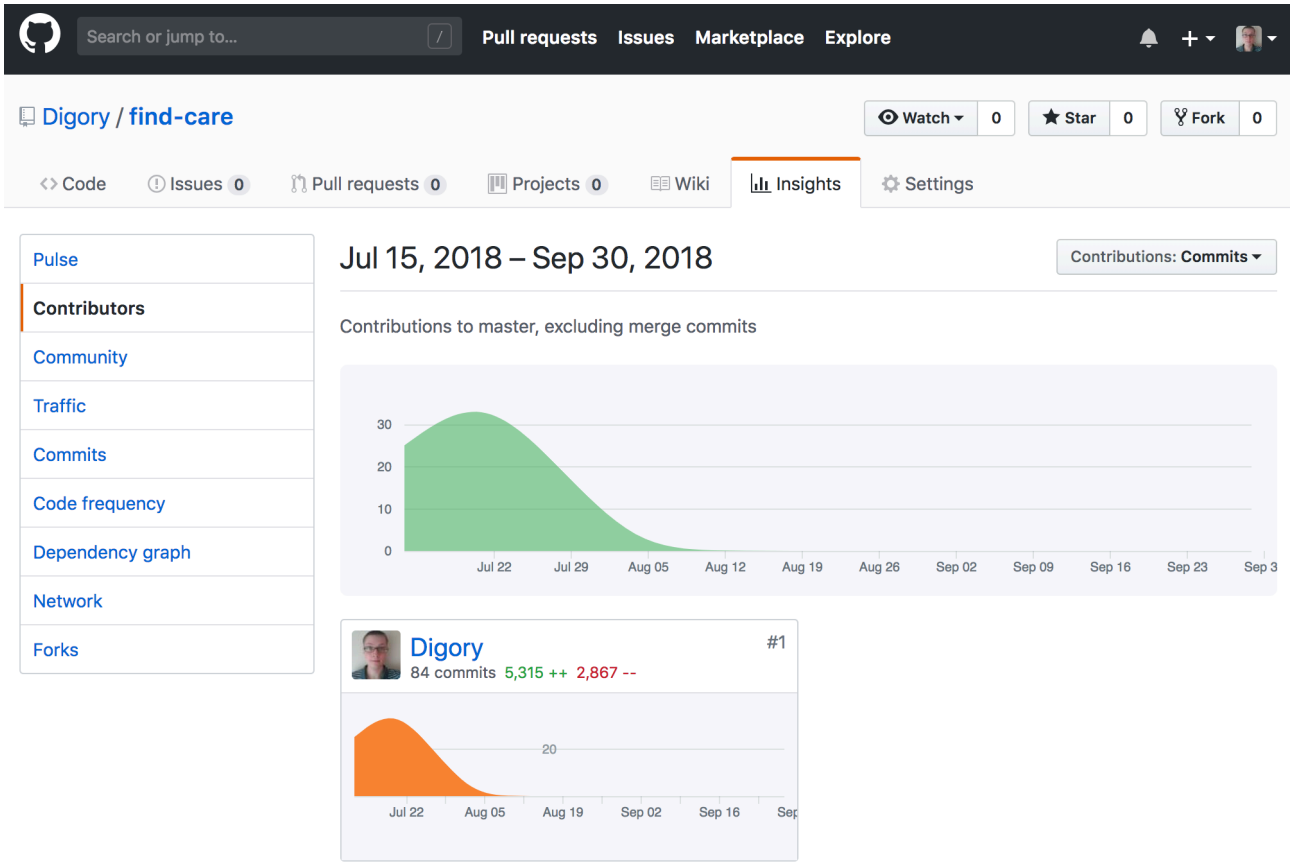
Runner.java

```java
import models.ComedyActor;

public class Runner {
    public static void main(String[] args) {

        ComedyActor comedyJane = new ComedyActor( name: "Jane");

        System.out.println(comedyJane.audition());

    }
}
```

Run: Runner

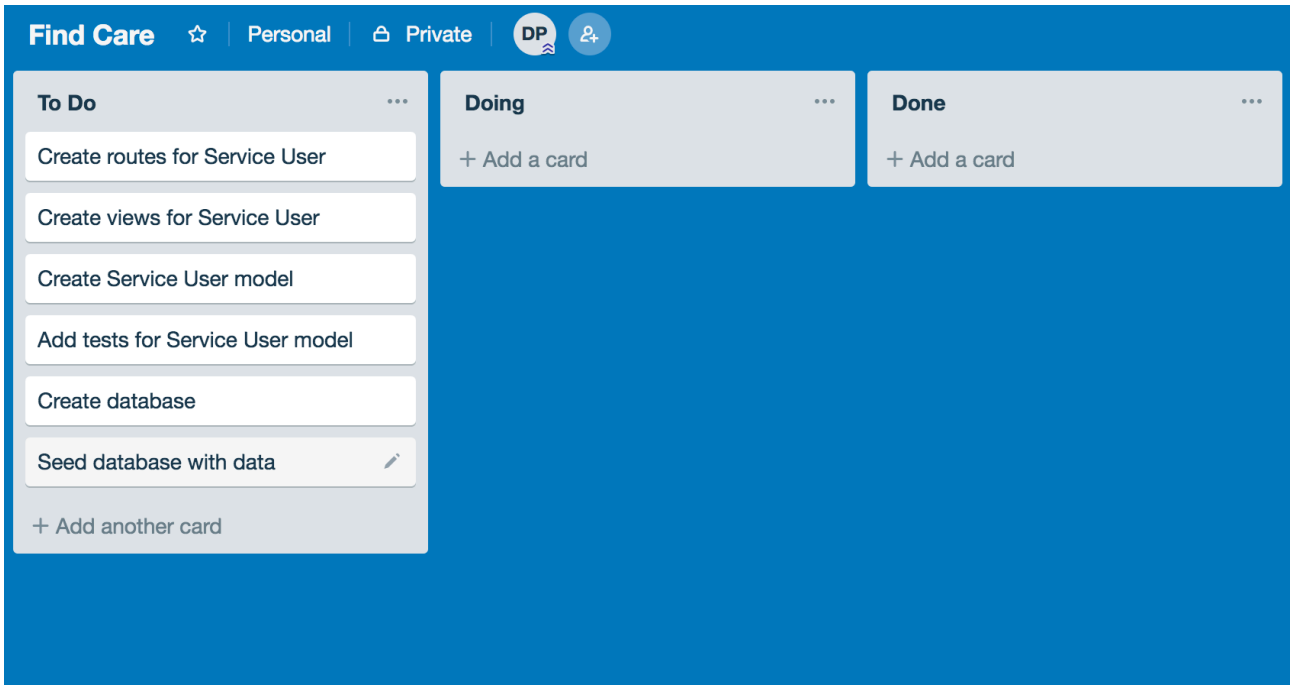
```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...
Hey guys, my genre of speciality is COMEDY wanna hear a joke?
```

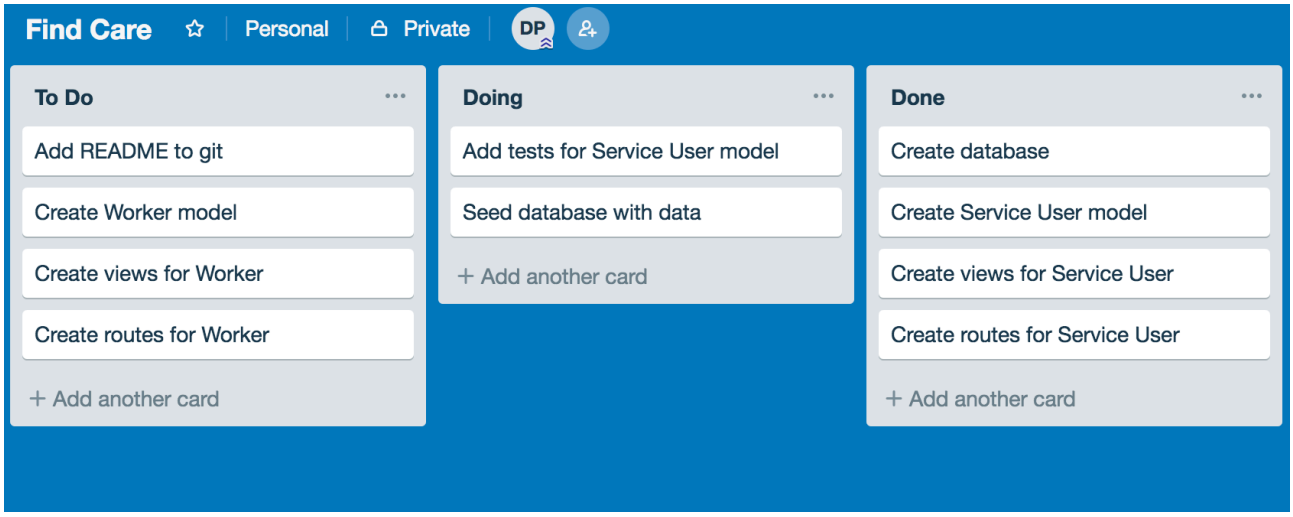
Unit	Ref	Evidence
P	P.11	Take a screenshot of one of your projects where you have worked alone and attach the Github link.



https://github.com/Digory/find-care

Unit	Ref	Evidence
P	P.12	Take screenshots or photos of your planning and the different stages of development to show changes.





Unit	Ref	Evidence
P	P.9	Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms.

```
def self.filtered_search(gender, can_drive, max_hourly_rate, searched_array)
  found_workers = []
  for worker in self.all()
    if worker.gender() == gender || gender == "a"
      if worker.can_drive() == true || can_drive == "a"
        if worker.hourly_rate() <= max_hourly_rate.to_f/$cost_multiplier
          if worker.does_experience_match_all_filters?(searched_array)
            found_workers << worker
          end
        end
      end
    end
  end
  return found_workers
end
```

```
def does_experience_match_all_filters?(searched_array)
  return true if searched_array.include?("any")
  worker_experience_array = @experience.split(" | ")
  return searched_array.all?{|string| worker_experience_array.include?(string)}
end
```

These are two algorithms I wrote for the solo project. The project was to create a web app for helping people with disabilities find support workers; and the algorithms are used for searching for workers based upon criteria inputted by the user.

In the “self.filtered_search” method, we go through all the workers on the list and check to see, firstly, if their gender matches the searched gender, then their driving status and maximum hourly rate. The final argument, “searched_array”, represents the user-inputted criteria for worker experience. For example, the user can request the worker is experienced in First-Aid and/or Moving and Handling. Since the user can request more than one experience this is an array.

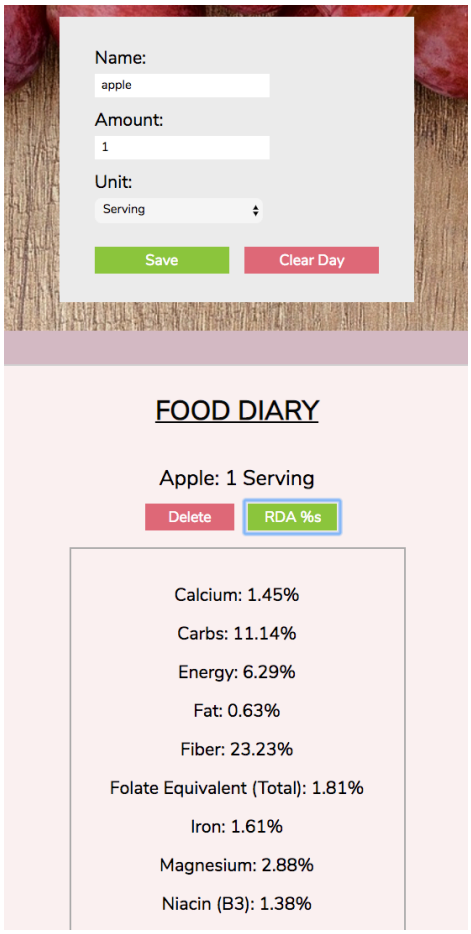
We use the helper method “does_experience_match_all_filters?” to check whether a worker has every type of experience in the array. It will return true only if they do, otherwise it will return false.

If a worker matches all of the previous criteria then they are added to an array “found_workers” and returned.

Week 12

Unit	Ref	Evidence
P	P.16	Show an API being used within your program. Take a screenshot of: * The code that uses or implements the API * The API being used by the program whilst running

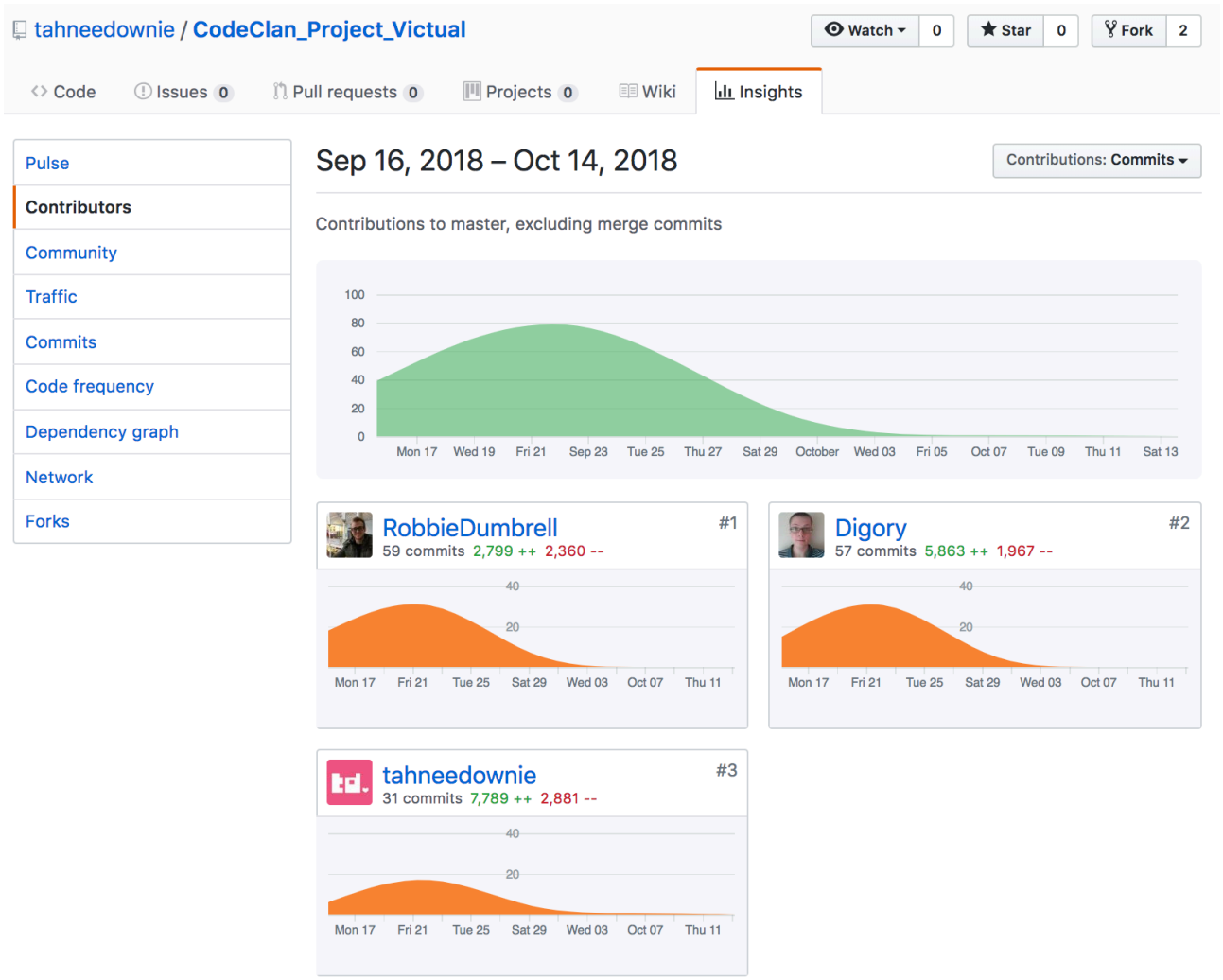
```
const dotenv = require('dotenv');
dotenv.load();
const apiKey = process.env.FOOD_API_KEY;
const applicationID = process.env.FOOD_APPLICATION_ID;

const FoodAPIRequest = function () {
};

FoodAPIRequest.prototype.get = function (foodWeWant) {
    return fetch(`https://api.edamam.com/api/food-database/parser?ingr=${foodWeWant}&app_id=${applicationID}&app_key=${apiKey}`)
        .then((response) => response.json());
};

FoodAPIRequest.prototype.post = function (payload) {
    return fetch(`https://api.edamam.com/api/food-database/nutrients?app_id=${applicationID}&app_key=${apiKey}`, {
        method: 'POST',
        body: JSON.stringify(payload),
        headers: { 'Content-Type': 'application/json' }
    })
        .then(response => response.json());
};

module.exports = FoodAPIRequest;
```



The API request finds the Recommended Daily Allowance of a specific food's nutrient content.

Week 15

Unit	Ref	Evidence
P	P.1	Take a screenshot of the contributor's page on Github from your group project to show the team you worked with.



Unit	Ref	Evidence
P	P.2	Take a screenshot of the project brief from your group project.

Project Victual

Aim:

Develop an application that helps monitors your daily nutritional intake and compare to recommended nutritional intake

MVP:

The app should:

Aim 1. Allows the user to keep a record of all the food / ingredients that the user submits on a particular day

Aim 2. Persist the food / ingredients entered so that the values can be displayed to user over time

Aim 3. Display nutritional value for each food / ingredient consumed

Aim 4. Display a summary of the nutritional intake for that day

How much of each item

What the nutritional of that item per g

Sum total of nutrients that day

API to be used:

Edamam

Extensions:

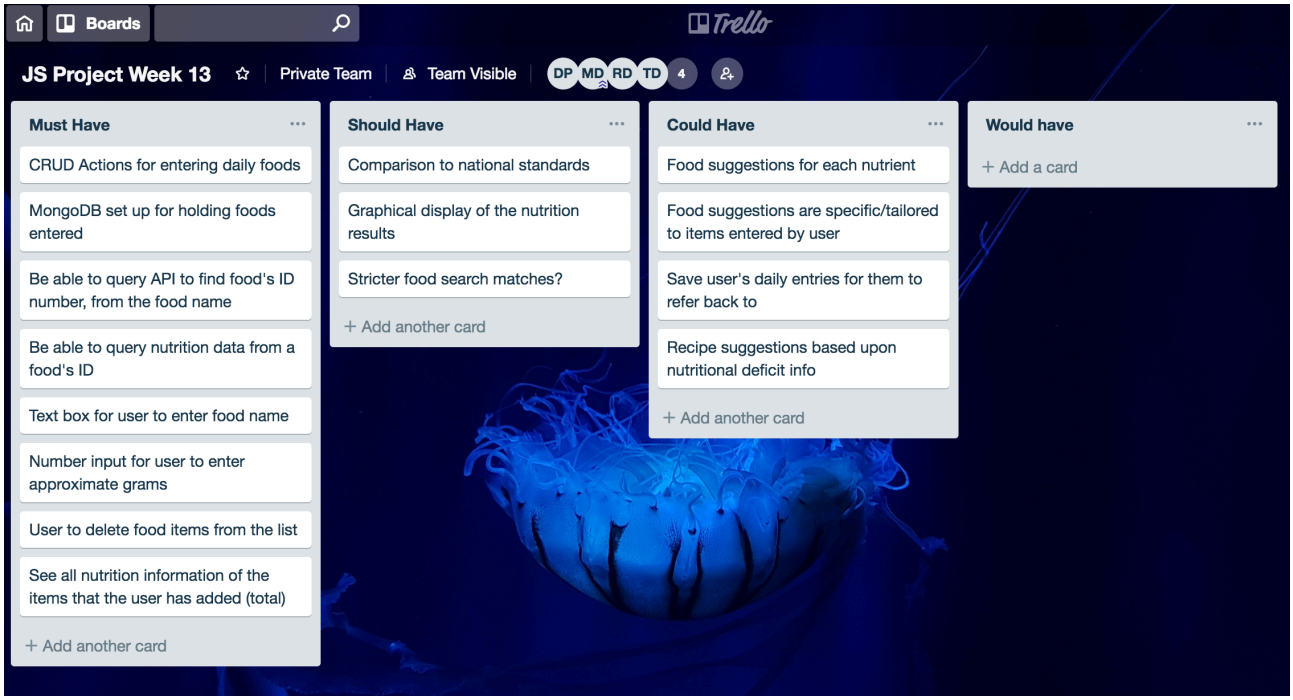
Compare actual versus expected nutrimental intake

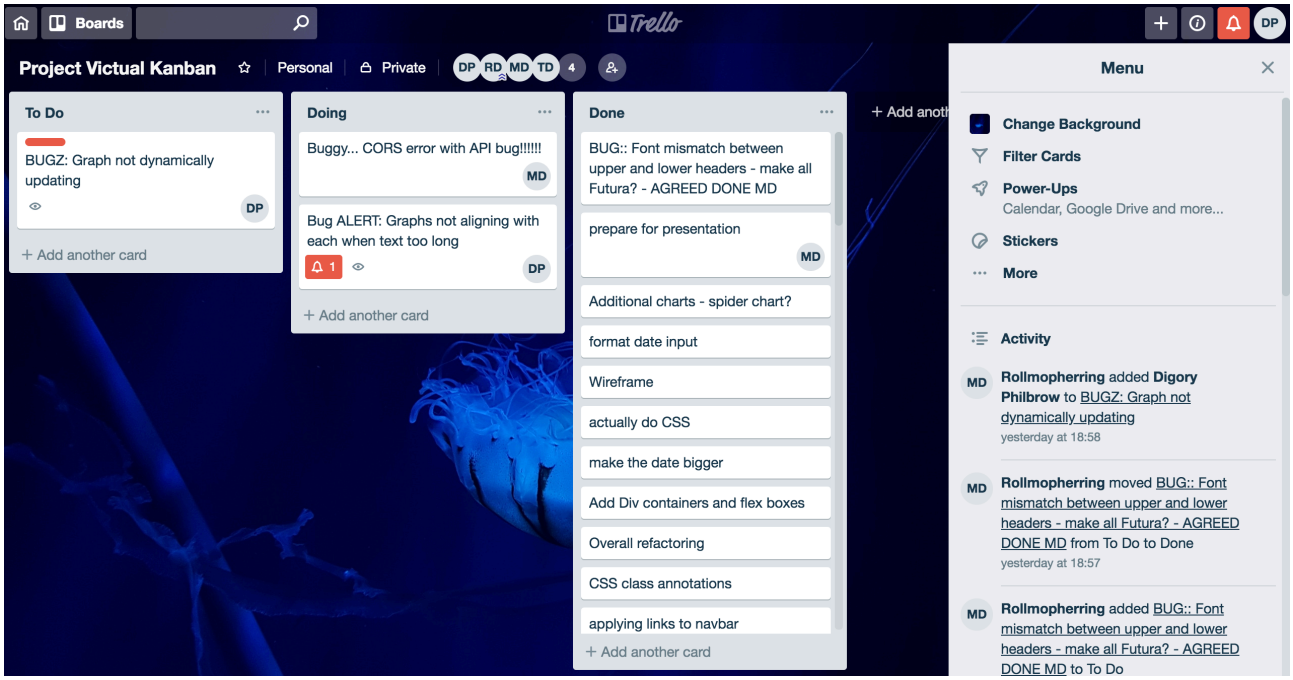
B. Graphical representation of the % difference

C. Suggestions for food intake changes

D. Suggestion of recipes that include the item that will compensate nutritional deficiencies

Unit	Ref	Evidence
P	P.3	Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.



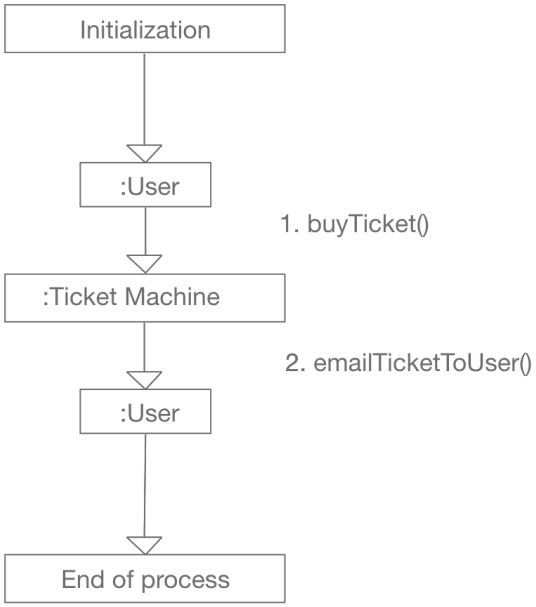


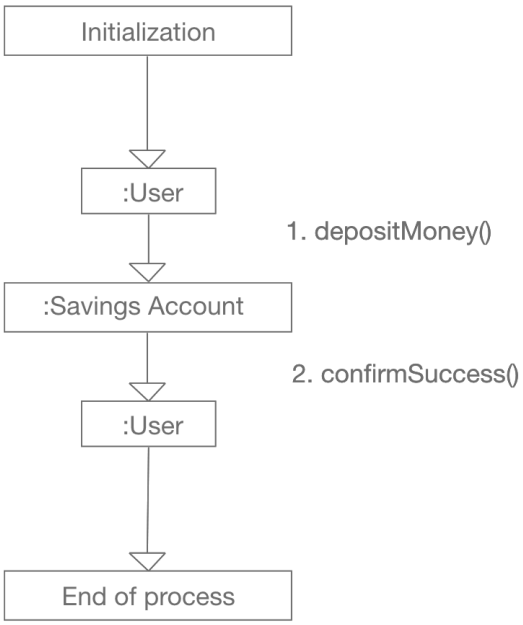
Unit	Ref	Evidence
P	P.4	Write an acceptance criteria and test plan.

Project Victual - nutrition app

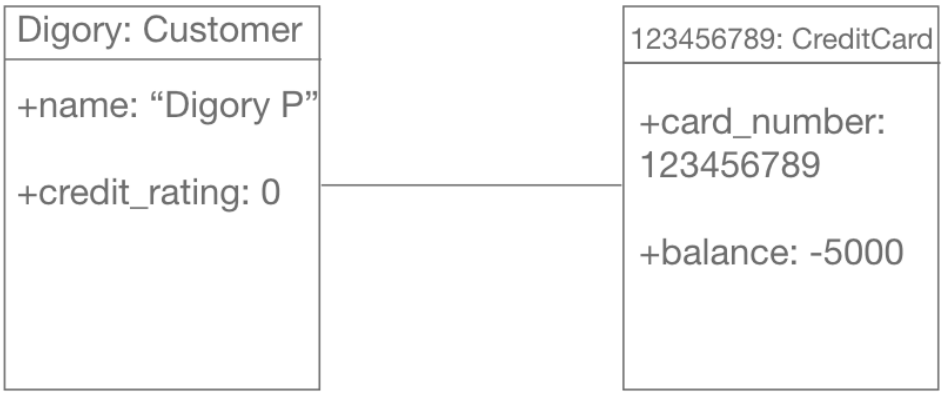
Acceptance Criteria	Expected result/output	Pass/Fail
A user is able to add a food to their food diary.	A form is displayed on the user homepage. When the user clicks the 'save' button the form is submitted and the food is saved to the database.	Pass
A user can view all of the foods in their food diary.	The user's list of foods in their diary is displayed on their homepage. When a food is added the diary is updated to reflect this.	Pass
A user can view each specific food's nutritional information.	An "RDA" button is displayed next to the food in the diary. When the button is clicked a list is expanded which shows the nutritional information.	Pass
A user is able to delete a food from their diary.	A 'delete' button is displayed next to the food in the diary. When the button is clicked the food is removed from the database and the food diary is updated to reflect this.	Pass
A user can view a graph of how their nutrient intake has changed over the past week.	Directly underneath the user's food diary is displayed a line graph with nutrient intake displayed over the previous week. The user can filter this graph with use of a dropdown menu which has options to show each specific nutrient on the graph.	Pass

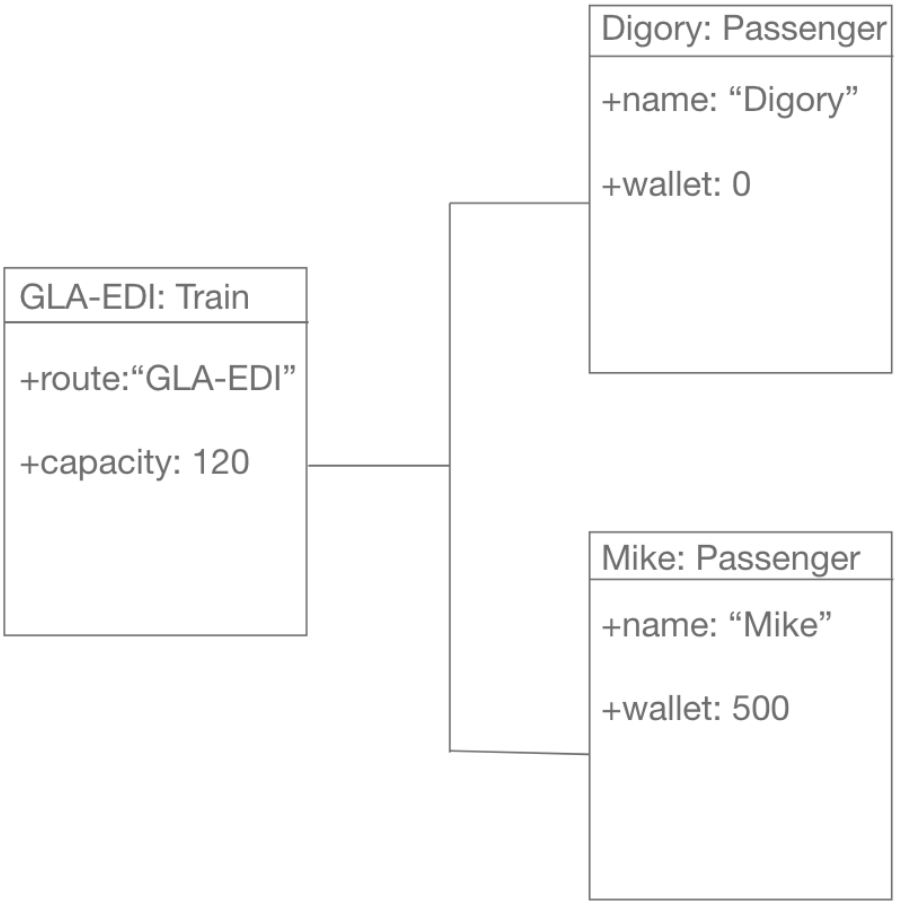
Unit	Ref	Evidence
P	P.7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).





Unit	Ref	Evidence
P	P.8	Produce two object diagrams.





Unit	Ref	Evidence
P	P.17	Produce a bug tracking report

User can add a food to their food diary.	FAIL	Link the information on the form submitted by user to our model, which saves the information to the database.	PASS
User can view a graph displaying their weekly nutrient intake.	FAIL	Calculate each day's nutrient intake over the past week and use Highcharts to display the information.	PASS
User can see recipes based upon their most deficient nutrients.	FAIL	Use the user's total daily nutrient intake to calculate the 3 most deficient nutrients. Use this to query the Edamam API to find recipes with those nutrients in. Display the recipes to the user.	PASS
User can filter recipes based upon their own dietary requirements.	FAIL	Add a form underneath the recipes with options for different dietary types e.g. vegan. Use the form input to query the API with the additional recipe restrictions.	PASS
User can change the date of their food diary by clicking a date on a calendar.	FAIL	Add a calendar input to the top of the page. Add an event listener to the calendar which listens for any change. Use this to refresh the page when a date is selected.	PASS