Evidence Gathering Document for SQA Level 8 Professional Developer Award.

This document is designed for you to present your screenshots and diagrams relevant to the PDA and to also give a short description of what you are showing to clarify understanding for the assessor.

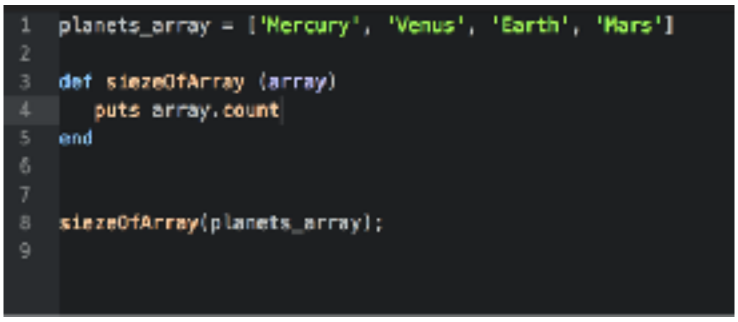
Each point that required details the Assessment Criteria (What you have to show) along with a brief description of the kind of things you should be showing.

Please fill in each point with screenshot or diagram and description.

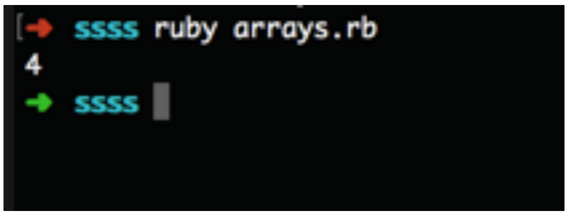
Week 2

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

Paste Screenshot here



```ruby
planets_array = ['Mercury', 'Venus', 'Earth', 'Mars']

def siezeOfArray (array)
   puts array.count
end


siezeOfArray(planets_array);

```



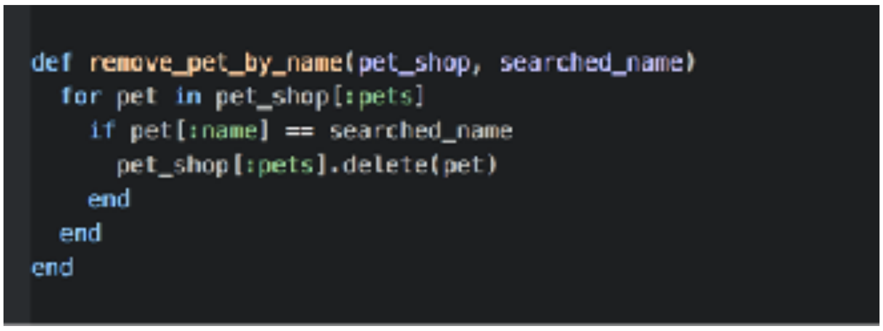
```
ssss ruby arrays.rb
4
ssss 
```

Description here

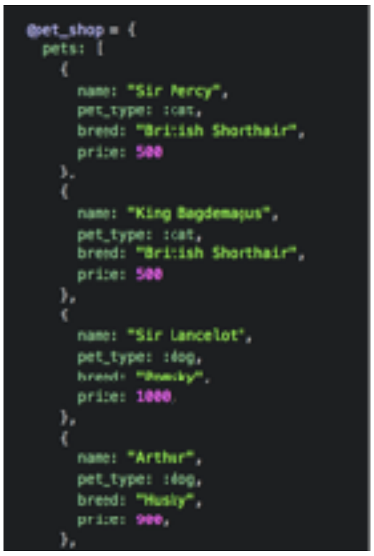
Planets array contains planets. sizeOfArray function counts how many elements are there in the array. When run in the terminal function returns 4 which is the size of the array.

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

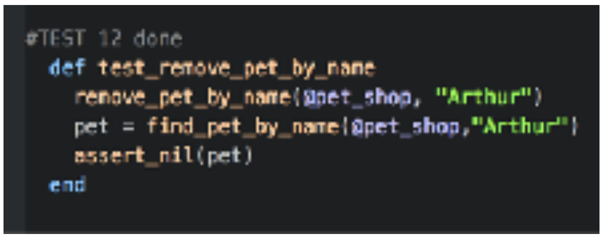
<u>Paste Screenshot here</u>



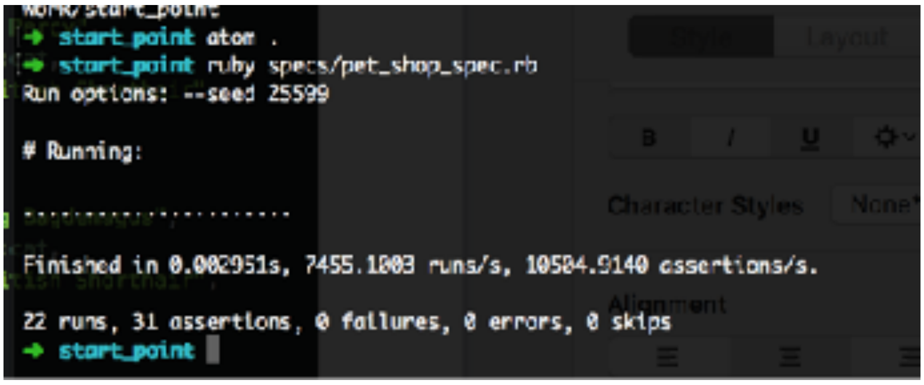
```ruby
def remove_pet_by_name(pet_shop, searched_name)
  for pet in pet_shop[:pets]
    if pet[:name] == searched_name
      pet_shop[:pets].delete(pet)
    end
  end
end
```



```ruby
@pet_shop = {
  pets: [
    {
      name: "Sir Percy",
      pet_type: :cat,
      breed: "British Shorthair",
      price: 500
    },
    {
      name: "King Bagdemagus",
      pet_type: :cat,
      breed: "British Shorthair",
      price: 500
    },
    {
      name: "Sir Lancelot",
      pet_type: :dog,
      breed: "Pomsky",
      price: 1000
    },
    {
      name: "Arthur",
      pet_type: :dog,
      breed: "Husky",
      price: 900,
    },
```



```ruby
#TEST 12 done
  def test_remove_pet_by_name
    remove_pet_by_name(@pet_shop, "Arthur")
    pet = find_pet_by_name(@pet_shop,"Arthur")
    assert_nil(pet)
  end
```



```
work/start_point
➜ start_point atom .
➜ start_point ruby specs/pet_shop_spec.rb
Run options: --seed 25599

# Running:

......................

Finished in 0.002951s, 7455.1003 runs/s, 10504.9140 assertions/s.

22 runs, 31 assertions, 0 failures, 0 errors, 0 skips
➜ start_point
```

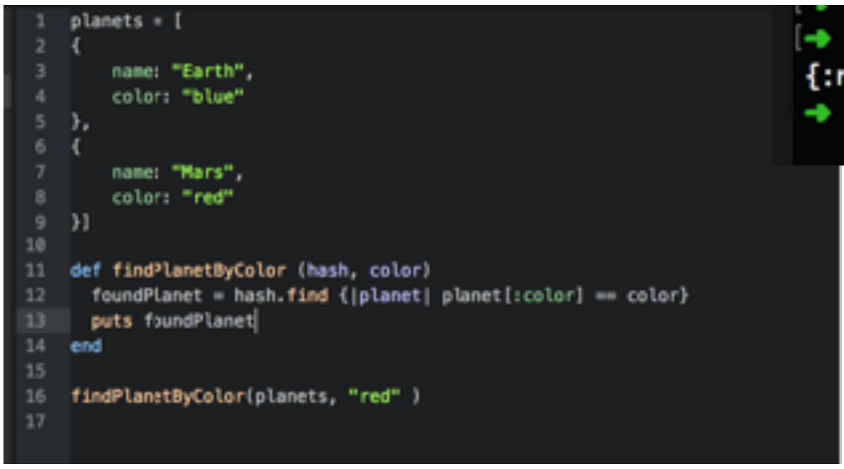
<u>Description here</u>

The function remove_pet_by_name takes in the hash pet shop and an argument for the type of the animal that will be searched for. It then searches through the hash to find all animal of the specific name using [:name] to access values inside of the hash. If it finds it, it then removes the pet that was found. This is shown by running the test and passing them on the screenshot.

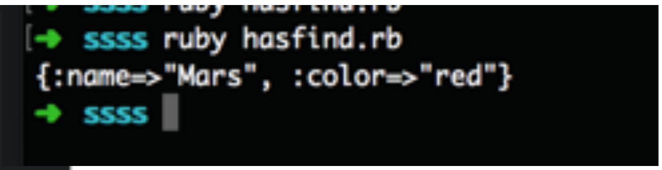
<u>Week 3</u>

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

Paste Screenshot here



```ruby
planets = [
{
    name: "Earth",
    color: "blue"
},
{
    name: "Mars",
    color: "red"
}]

def findPlanetByColor (hash, color)
  foundPlanet = hash.find {|planet| planet[:color] == color}
  puts foundPlanet
end

findPlanetByColor(planets, "red" )
```



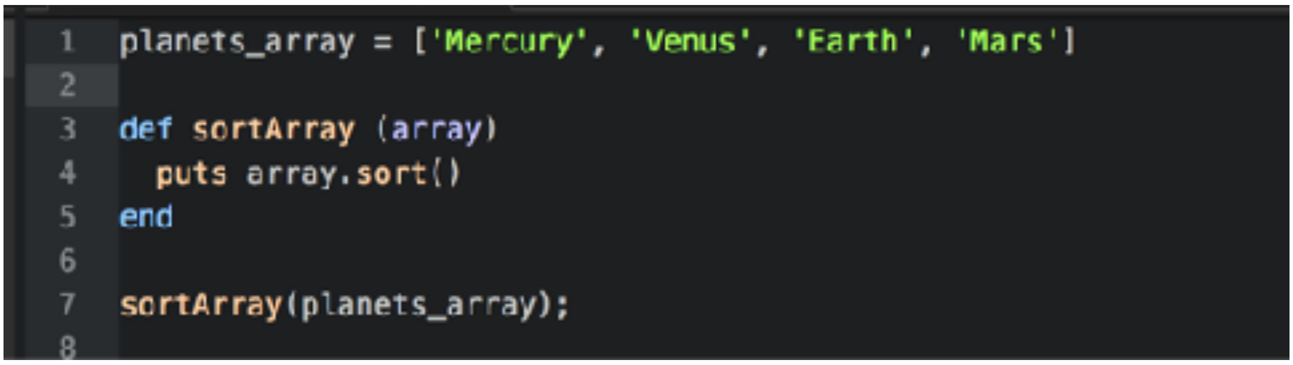
```
➜ ssss ruby hasfind.rb
{:name=>"Mars", :color=>"red"}
➜ ssss
```

Description here

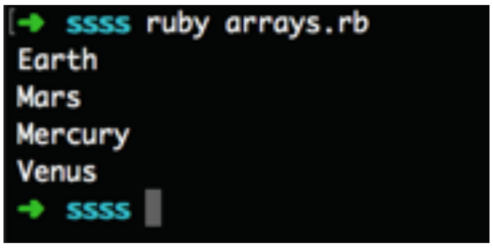
This function performs a search for a specific planet by its color. On the left - hash with planet elements and a function that looks for a planet by its color. Function takes a hash and the colour as parameters. Function is called with color ‘red’. On the right, terminal output showing the planet select is Mars as the colour is red.

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

Paste Screenshot here



```ruby
planets_array = ['Mercury', 'Venus', 'Earth', 'Mars']

def sortArray (array)
  puts array.sort()
end

sortArray(planets_array);
```



```
[➜  ssss ruby arrays.rb
Earth
Mars
Mercury
Venus
➜  ssss 
```

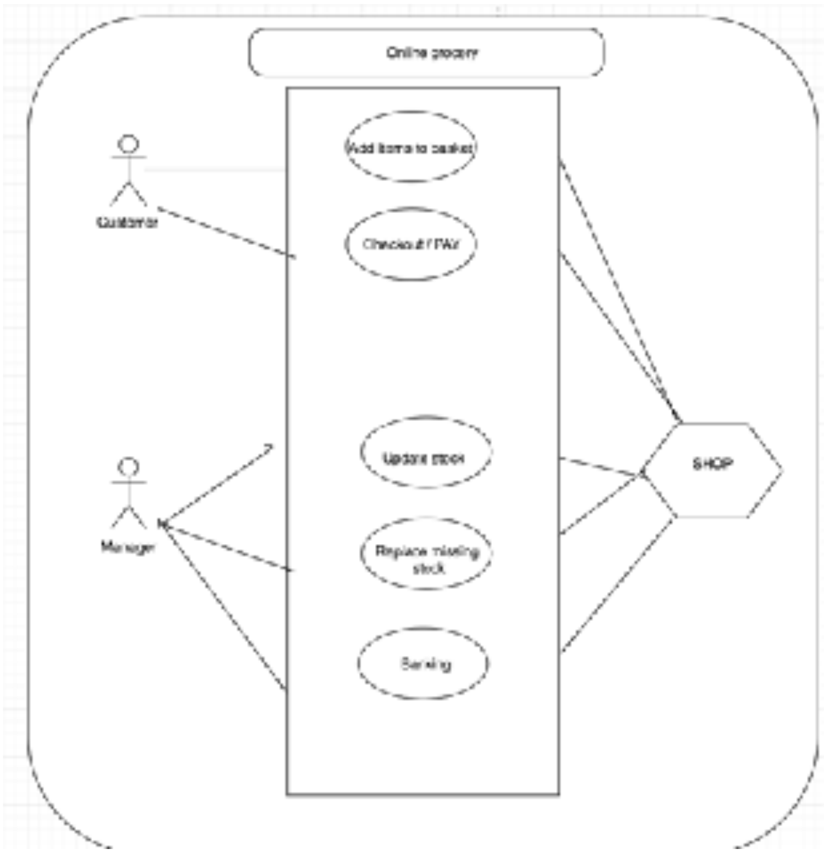
Description here

Planets array contains planets. sortArray function sorts elements of the array in the alphabetical order. When run in the terminal function returns planets in the alphabetical order as expected.

Week 5

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

Paste Screenshot here

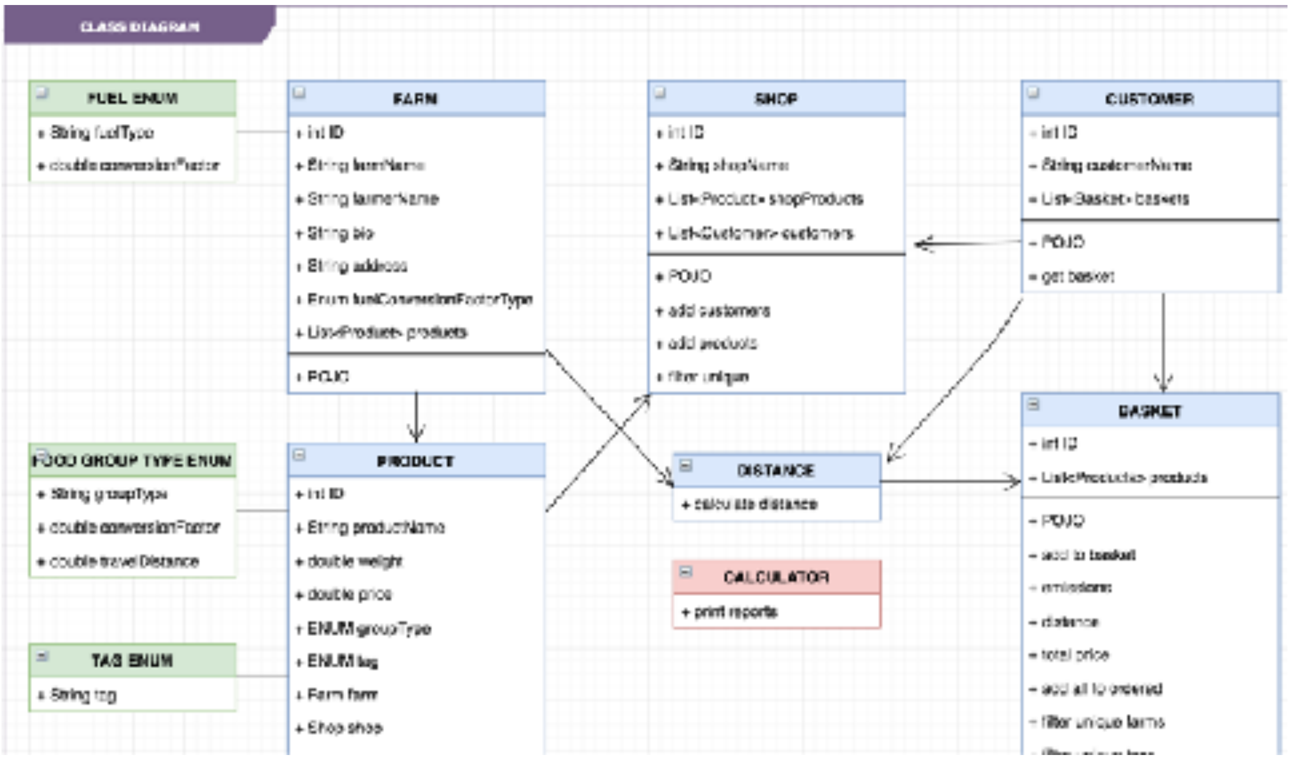


Description here

A use case diagram for the online grocery shopping.

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

<u>Paste Screenshot here</u>

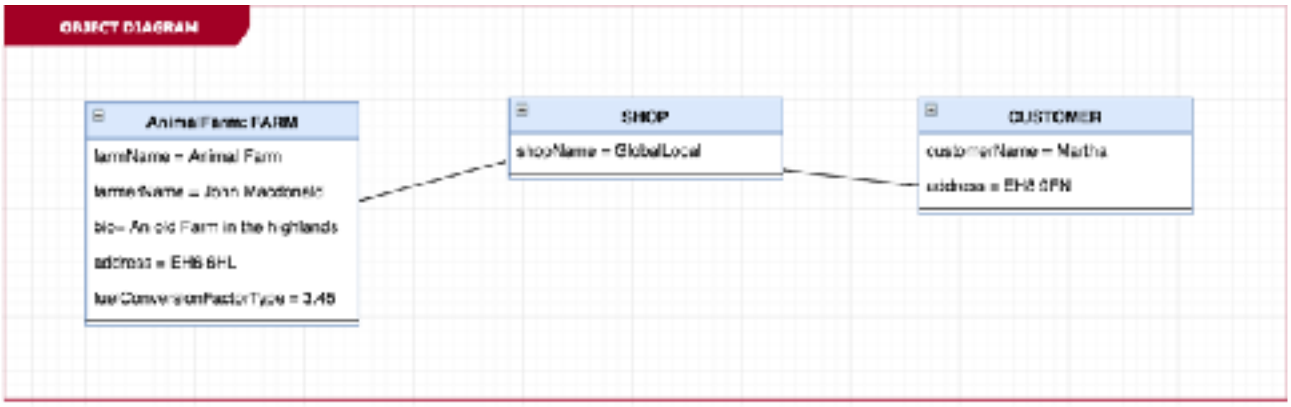


<u>Description here</u>

A class diagram of the online shop.

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

Paste Screenshot here

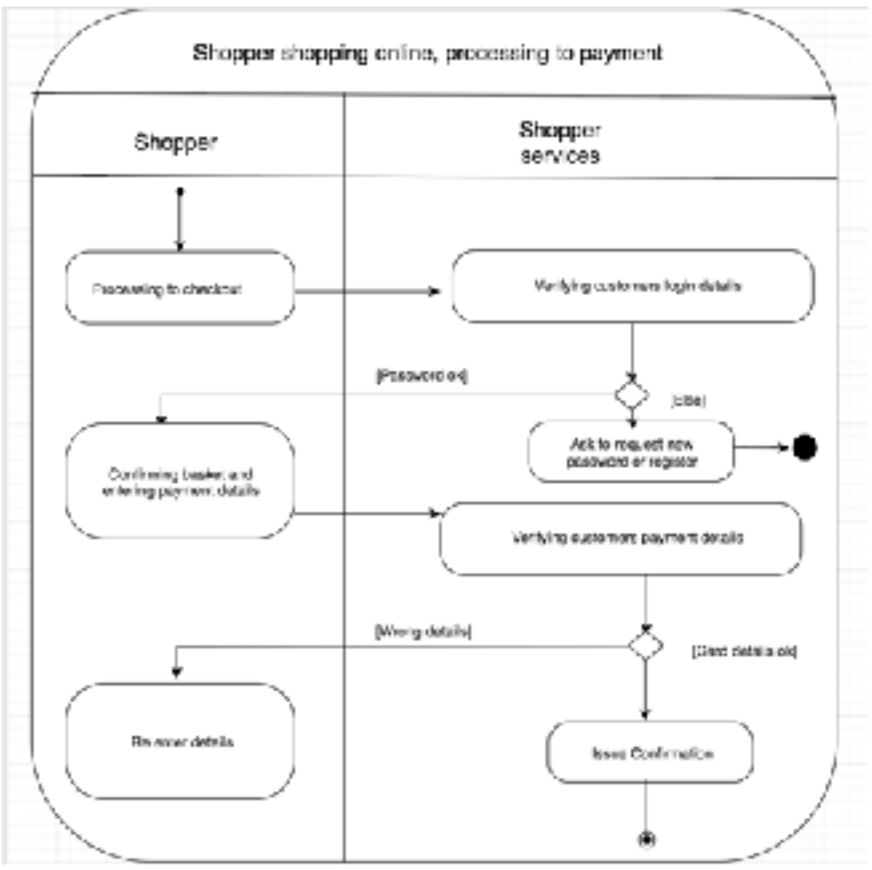


Description here

A simplified object diagram of the online shop.

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

Paste Screenshot here


Shopper shopping online, processing to payment
Shopper
Shopper services
Proceeding to checkout
Verifying customers login details
[Password ok]
[else]
Ask to request new password or register
Confirming basket and entering payment details
Verifying customers payment details
[Wrong details]
[Card details ok]
Re-enter details
Issue Confirmation

Description here

An activity diagram for the online shopping payment processing.

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

Paste Screenshot here

Topic	Possible effect on Constraints on Product	Solution
Persistent Storage and Transactions	There are data persistency issues because of the data storage capacity. The amount of data processed is slowing down the operating system. The data can be persisted only to a specific type of database e.g. Oracle. This can cause problems with persisting information or limit the type of platforms that the application can be used on.	Understanding the volume of the data processed and building in the accurate data storage and processing. Consider various data management systems - e.g. Oracle, MySQL and allow data to be used across all.

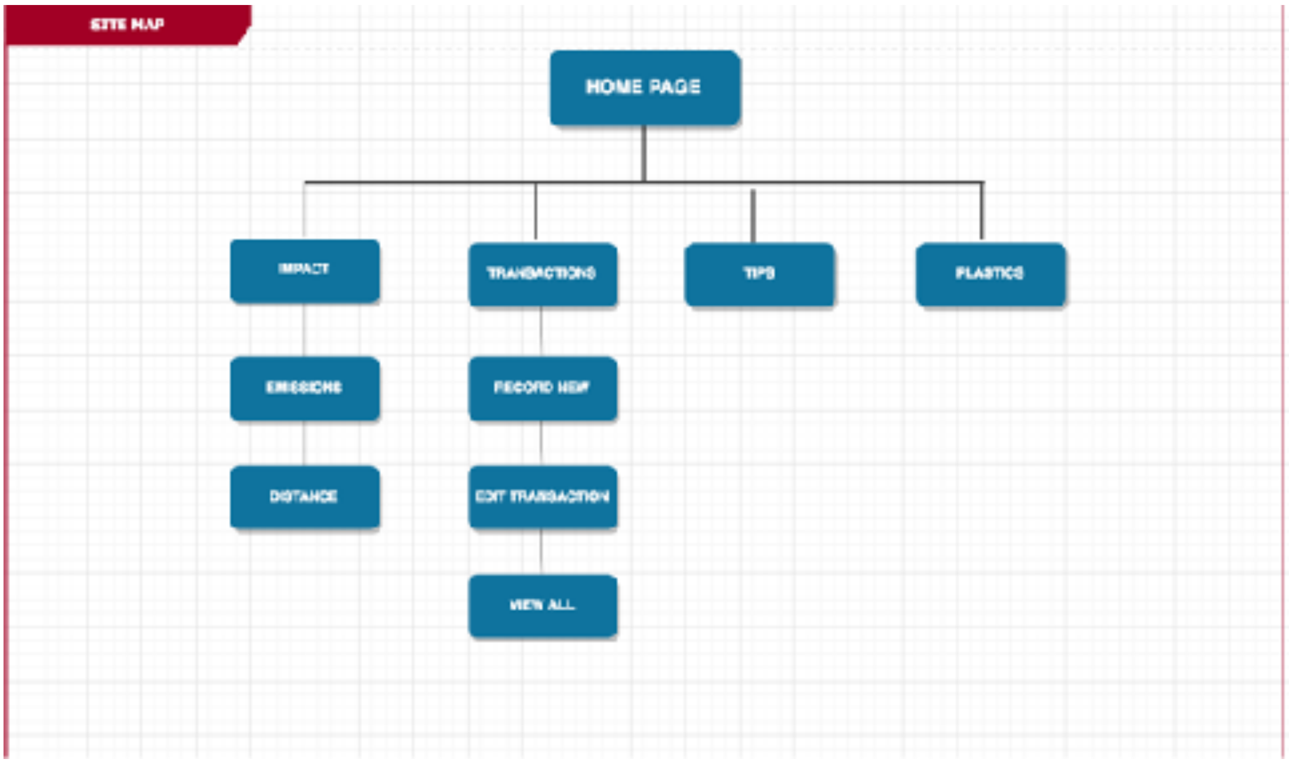
Hardware and Software Platforms	The application can be hosted on a specific type of a software applications. This means that there are certain limitations to where it could be used.	At the development stage different operating systems should be considered and the application should be developed with different operating software in mind.
Performance Requirements	There might be specific performance requirements for the product including where the software will be run, how efficient and how quickly does the software need to perform. This can limit how the product will be designed, how the data will be stored.	The data flow and loading time have to be thought through before designing the solution. The performance requirements have to be taken into the account before building the software.
Usability	Product might have user specific usability requirements including the way that the product is designed with allies accessibility features. This can limit some design features.	The usability requirements have to be well defined before the product development and revised to ensure that the target audience users can navigate the software intuitively.
Budgets	Limited budget might be allocated for the product delivery, and the amount available for might stop or limit development of some of the features.	Outlining proper plan of how the budget will be spent and what features are feasible within the budget.
Time limitations	Project or some features might not be deliverable within the identified timeline	Revise the product road map and identify which features can be delivered within the timescale. Amend the MVP if required .

Description here

Implementation constraints table that identifies possible solutions to avoid these.

Unit	Ref	Evidence
P	P.5	User Site Map

Paste Screenshot here

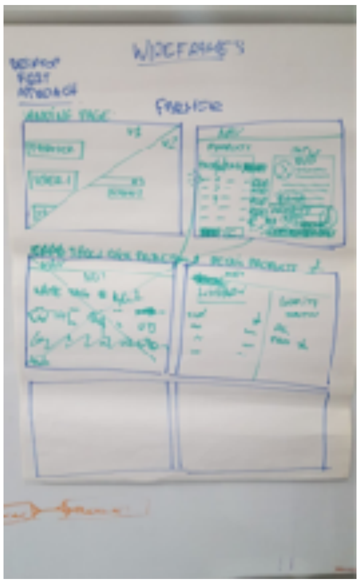


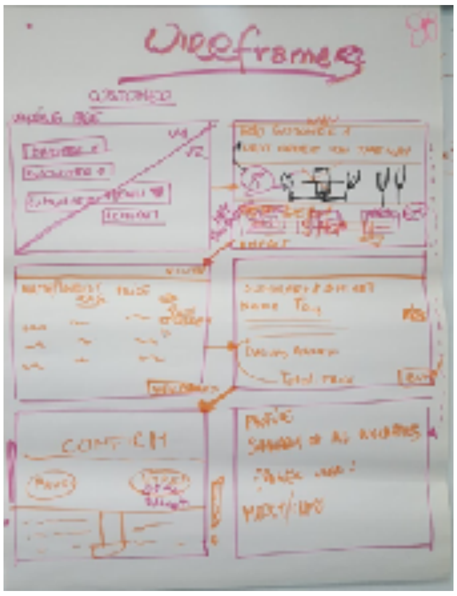
Description here

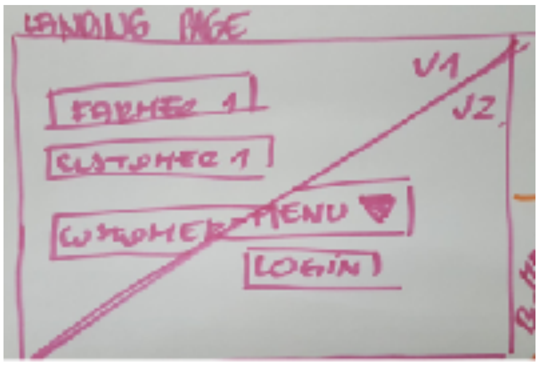
A user site map for my plastic solo project in Ruby

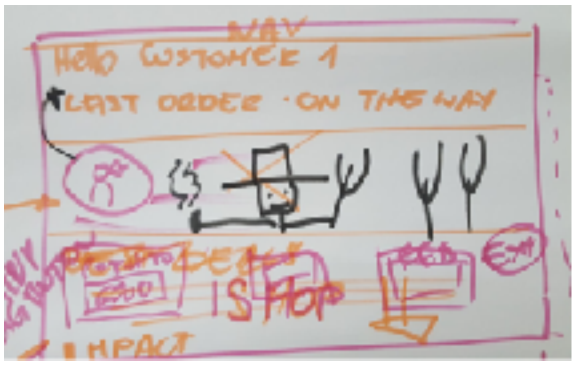
Unit	Ref	Evidence
P	P.6	2 Wireframe Diagrams

Paste Screenshot here





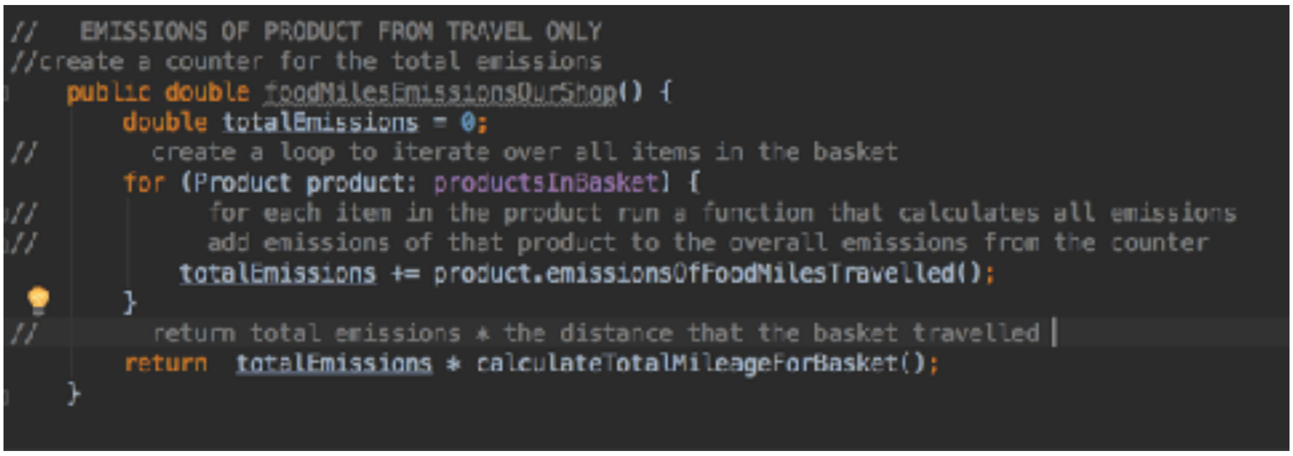




Description here
Two wireframe diagrams produced for the online shopping platform. It outlines how the user will interact with the web app and what screens will be built in to the program.

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

Paste Screenshot here



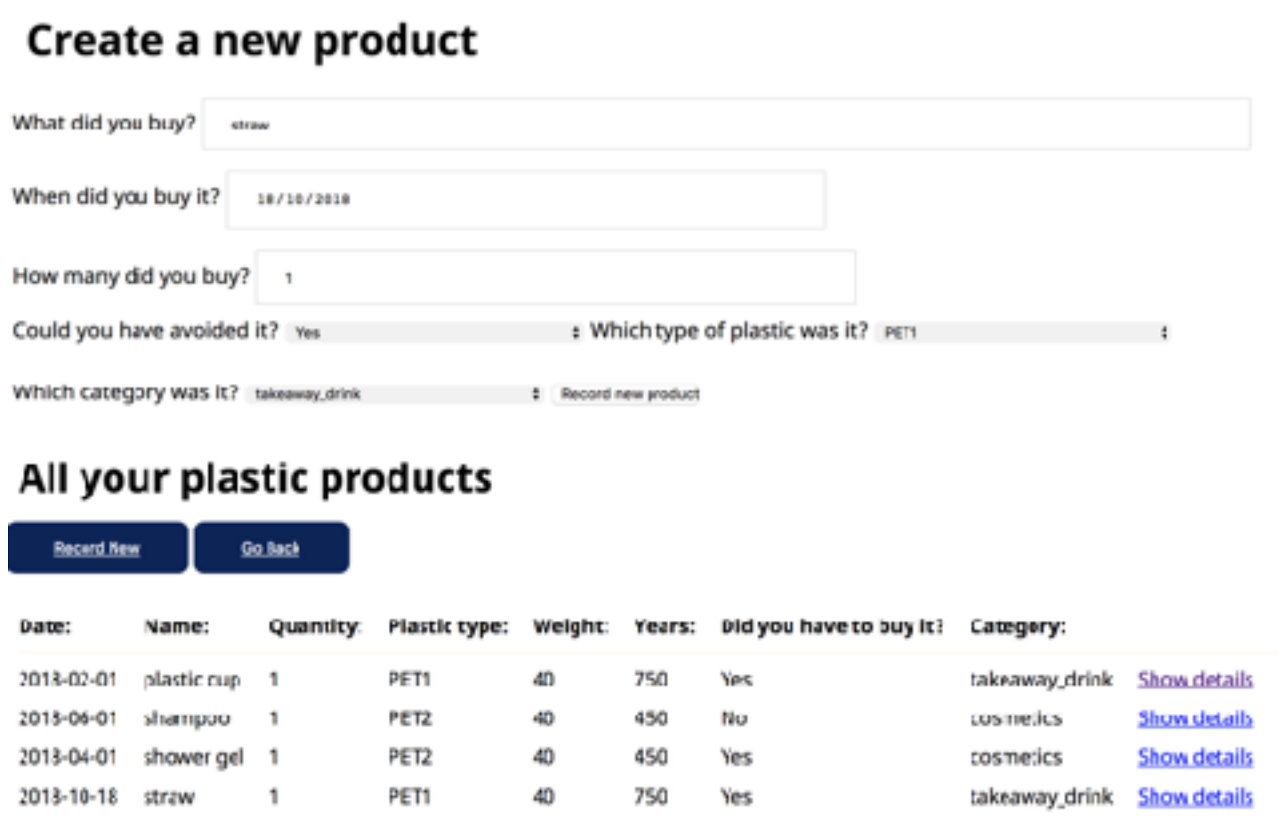
```
//   EMISSIONS OF PRODUCT FROM TRAVEL ONLY
//create a counter for the total emissions
    public double foodMilesEmissionsOurShop() {
        double totalEmissions = 0;
//          create a loop to iterate over all items in the basket
        for (Product product: productsInBasket) {
//              for each item in the product run a function that calculates all emissions
//              add emissions of that product to the overall emissions from the counter
            totalEmissions += product.emissionsOfFoodMilesTravelled();
        }
//          return total emissions * the distance that the basket travelled
        return  totalEmissions * calculateTotalMileageForBasket();
    }
```

Description here

Function that contains pseudo code that identifies steps of what the function should be doing once run.

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

Paste Screenshot here



Create a new product

What did you buy? straw

When did you buy it? 18/10/2018

How many did you buy? 1

Could you have avoided it? Yes   Which type of plastic was it? PET1

Which category was it? takeaway_drink   Record new product

All your plastic products

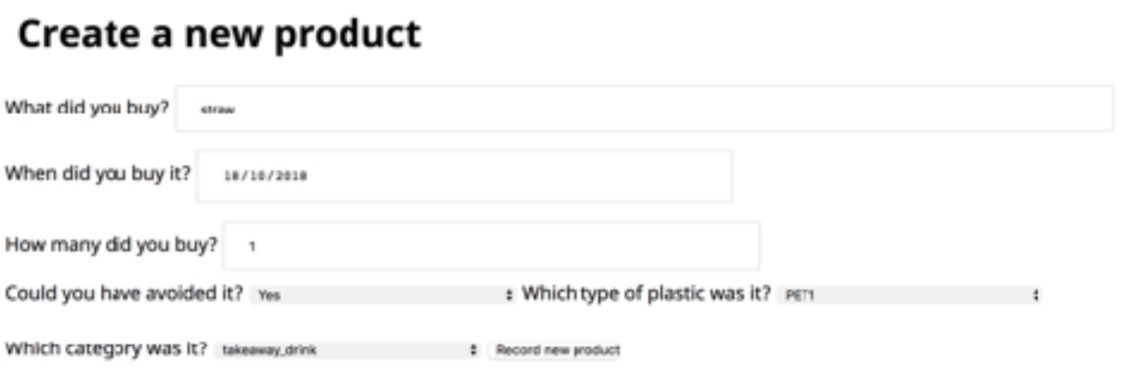
Date:	Name:	Quantity:	Plastic type:	Weight:	Years:	Did you have to buy it?	Category:	
2018-02-01	plastic cup	1	PET1	40	750	Yes	takeaway_drink	Show details
2018-06-01	shampoo	1	PET2	40	450	No	cosmetics	Show details
2018-04-01	shower gel	1	PET2	40	450	Yes	cosmetics	Show details
2018-10-18	straw	1	PET1	40	750	Yes	takeaway_drink	Show details

<u>Description here</u>

<u>A </u>screenshot of the user inputting details of the plastic product and then a screenshot of the table containing all products including the one that was entered by the user.

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

<u>Paste Screenshot here</u>



Create a new product

What did you buy? straw

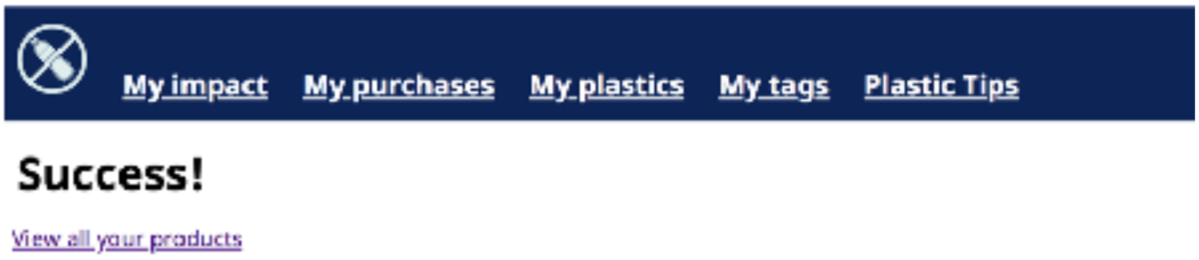
When did you buy it? 18/10/2018

How many did you buy? 1

Could you have avoided it? Yes

Which type of plastic was it? PET1

Which category was it? takeaway_drink



Success!

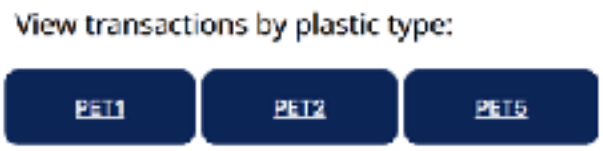
View all your products

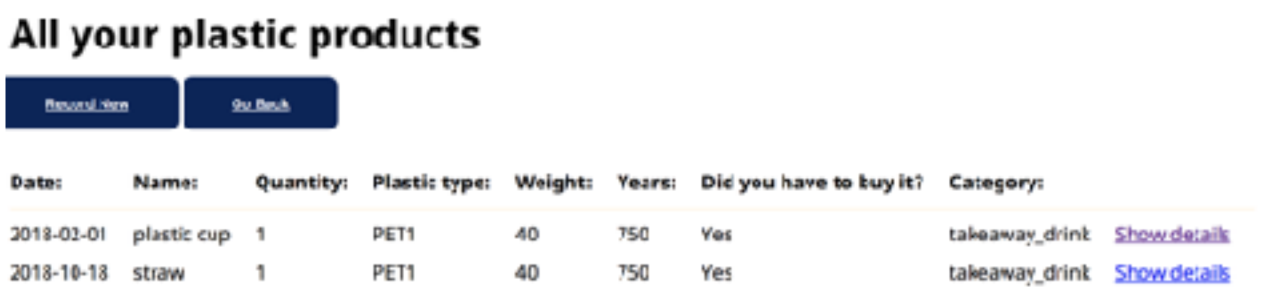
<u>Description here</u>

A screenshot of the user inputting details of the new product into the program and then a screenshot of the on screen message that is displayed to the user after the input has been 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

<u>Paste Screenshot here</u>


View transcriptions by plastic type:


All your plastic products

Date:	Name:	Quantity:	Plastic type:	Weight:	Years:	Did you have to buy it?	Category:	
2018-02-01	plastic cup	1	PET1	40	750	Yes	takeaway_drink	Show details
2018-10-18	straw	1	PET1	40	750	Yes	takeaway_drink	Show details

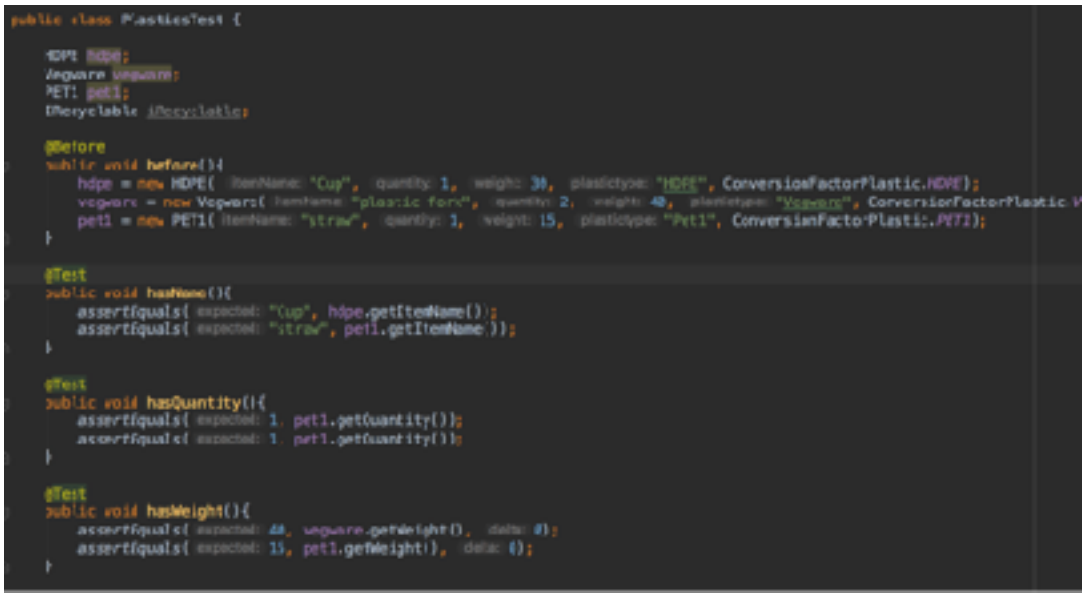
<u>Description here</u>

An example of user requesting to see all products of the specific plastic type. Screenshots show the buttons that outline options and then the table that displays all products of the particular plastic type.

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

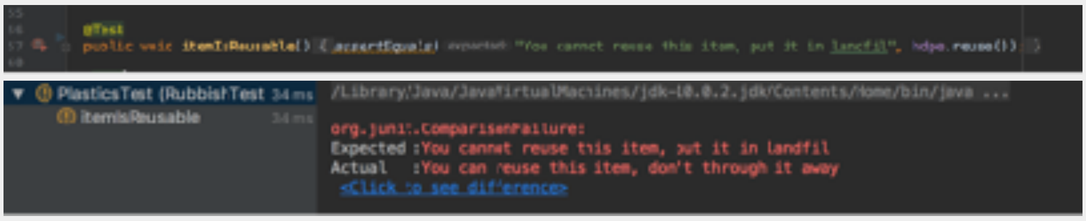
Paste Screenshot here

Example of a test code



```
public class PlasticsTest {

    HDPE hdpe;
    Vegware vegware;
    PET1 pet1;
    IRecyclable iRecyclable;

    @Before
    public void before(){
        hdpe = new HDPE( itemName: "Cup",  quantity: 1,  weight: 30,  plasticType: "HDPE", ConversionFactorPlastic.HDPE);
        vegware = new Vegware( itemName: "plastic fork",  quantity: 2,  weight: 40,  plasticType: "Vegware", ConversionFactorPlastic.V
        pet1 = new PET1( itemName: "straw",  quantity: 1,  weight: 15,  plasticType: "Pet1", ConversionFactorPlastic.PET1);
    }

    @Test
    public void hasName(){
        assertEquals( expected: "Cup", hdpe.getItemName());
        assertEquals( expected: "straw", pet1.getItemName());
    }

    @Test
    public void hasQuantity(){
        assertEquals( expected: 1, pet1.getQuantity());
        assertEquals( expected: 1, pet1.getQuantity());
    }

    @Test
    public void hasWeight(){
        assertEquals( expected: 40, vegware.getWeight(),  delta: 0);
        assertEquals( expected: 15, pet1.getWeight(),  delta: 0);
    }
```

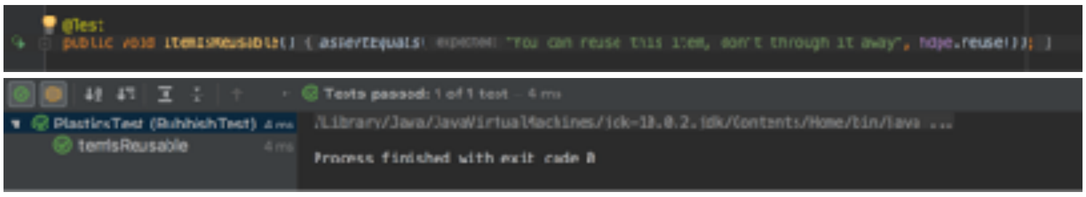
Test code failing to pass



```
    @Test
    public void itemIsReusable() { assertEquals( expected: "You cannot reuse this item, put it in landfil", hdpe.reuse()); }
```

```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...

org.junit.ComparisonFailure:
Expected :You cannot reuse this item, put it in landfil
Actual   :You can reuse this item, don't through it away
 <Click to see difference>
```

Example once errors were corrected

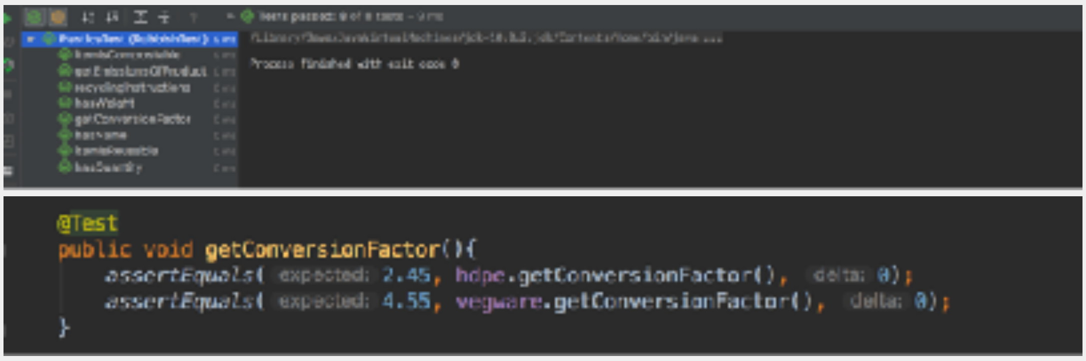


```
    @Test
    public void itemIsReusable() { assertEquals( expected: "You can reuse this item, don't through it away", hdpe.reuse()); }
```

Tests passed: 1 of 1 test – 4 ms
PlasticsTest (RubbishTest) 4 ms
itemIsReusable 4 ms

```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

Test code passing



```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

```
    @Test
    public void getConversionFactor(){
        assertEquals( expected: 2.45, hdpe.getConversionFactor(),  delta: 0);
        assertEquals( expected: 4.55, vegware.getConversionFactor(),  delta: 0);
    }
```

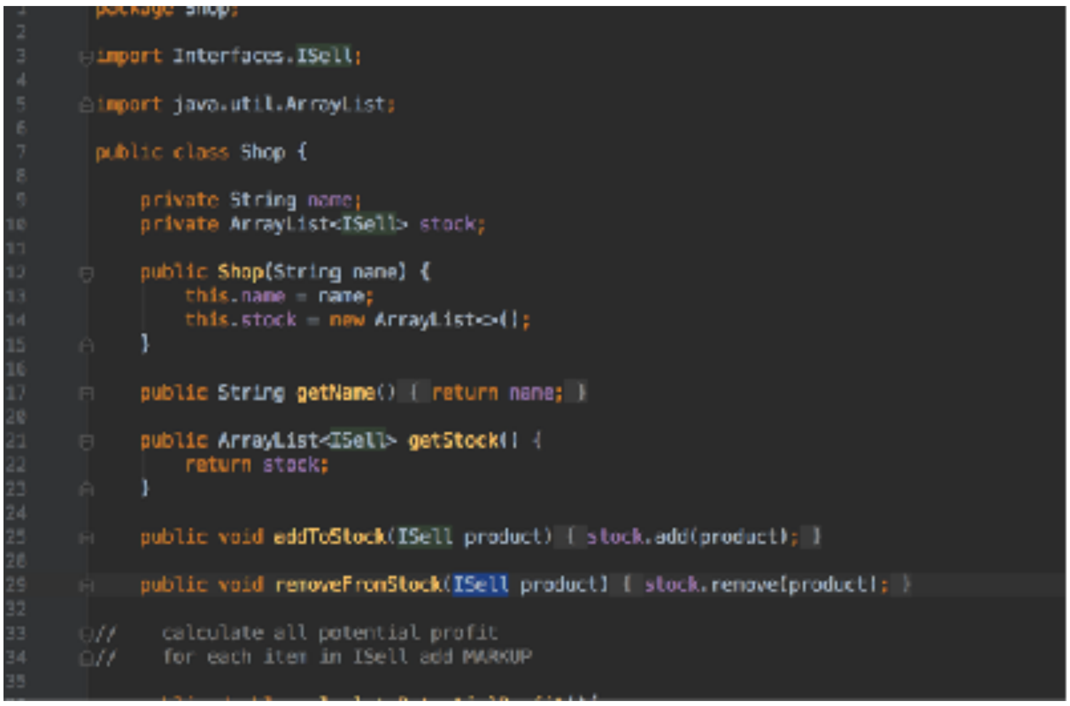
Description here

The code above shows a set of tests for the class, it then shows the failing itemIsReusable test for the function reuse() function and then the fixed passing test.

Week 7

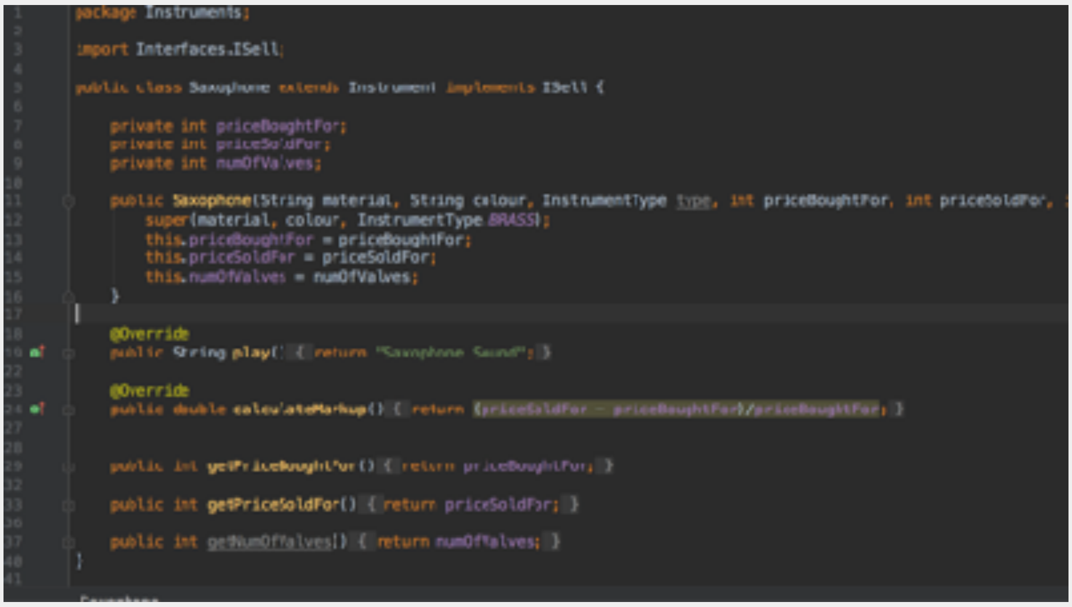
Unit	Ref	Evidence
I&T	I.T.7	The use of Polymorphism in a program and what it is doing.

Paste Screenshot here

The shop class.	

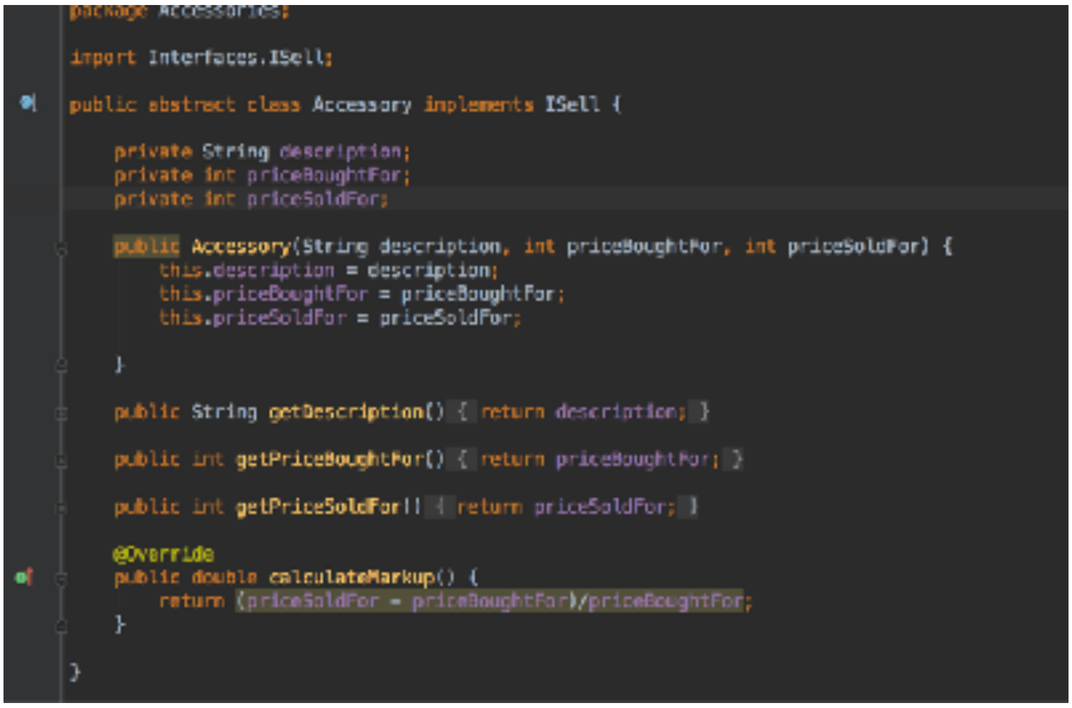
```java
package shop;

import Interfaces.ISell;

import java.util.ArrayList;

public class Shop {

    private String name;
    private ArrayList<ISell> stock;

    public Shop(String name) {
        this.name = name;
        this.stock = new ArrayList<>();
    }

    public String getName() { return name; }

    public ArrayList<ISell> getStock() {
        return stock;
    }

    public void addToStock(ISell product) { stock.add(product); }

    public void removeFromStock(ISell product) { stock.remove(product); }

//    calculate all potential profit
//    for each item in ISell add MARKUP
```

Clas implementing interface



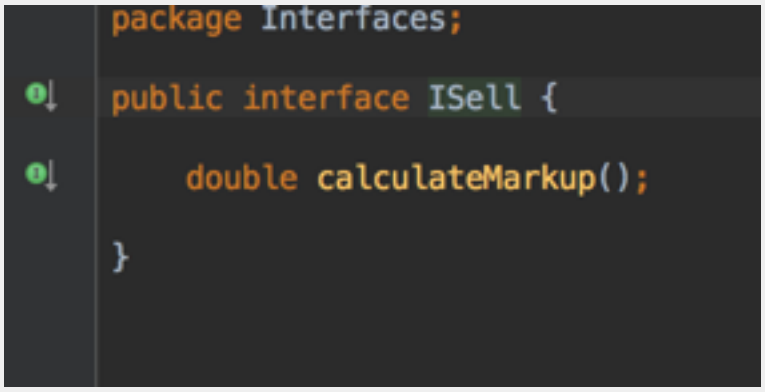
```java
package Instruments;

import Interfaces.ISell;

public class Saxophone extends Instrument implements ISell {

    private int priceBoughtFor;
    private int priceSoldFor;
    private int numOfValves;

    public Saxophone(String material, String colour, InstrumentType type, int priceBoughtFor, int priceSoldFor,
        super(material, colour, InstrumentType.BRASS);
        this.priceBoughtFor = priceBoughtFor;
        this.priceSoldFor = priceSoldFor;
        this.numOfValves = numOfValves;
    }

    @Override
    public String play() { return "Saxophone Sound"; }

    @Override
    public double calculateMarkup() { return (priceSoldFor - priceBoughtFor)/priceBoughtFor; }


    public int getPriceBoughtFor() { return priceBoughtFor; }

    public int getPriceSoldFor() { return priceSoldFor; }

    public int getNumOfValves() { return numOfValves; }
}
```

Clas implementing interface



```java
package Accessories;

import Interfaces.ISell;

public abstract class Accessory implements ISell {

    private String description;
    private int priceBoughtFor;
    private int priceSoldFor;

    public Accessory(String description, int priceBoughtFor, int priceSoldFor) {
        this.description = description;
        this.priceBoughtFor = priceBoughtFor;
        this.priceSoldFor = priceSoldFor;

    }

    public String getDescription() { return description; }

    public int getPriceBoughtFor() { return priceBoughtFor; }

    public int getPriceSoldFor() { return priceSoldFor; }

    @Override
    public double calculateMarkup() {
        return (priceSoldFor - priceBoughtFor)/priceBoughtFor;
    }

}
```

The interface



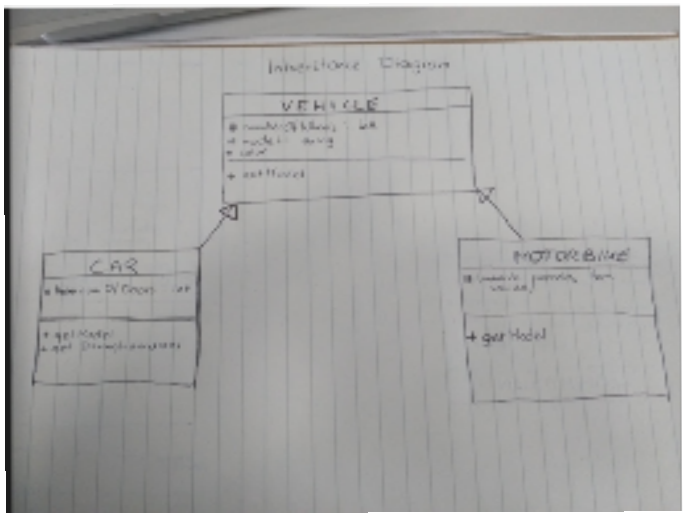
```java
package Interfaces;

public interface ISell {

    double calculateMarkup();

}
```

Description here

The code shows a class Shop that takes an Array List of ISell objects as it’s property. Next screenshots show two other classes that are implementing ISell interface and therefore take functions that the interface showed in the final screenshot has.

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

Paste Screenshot here

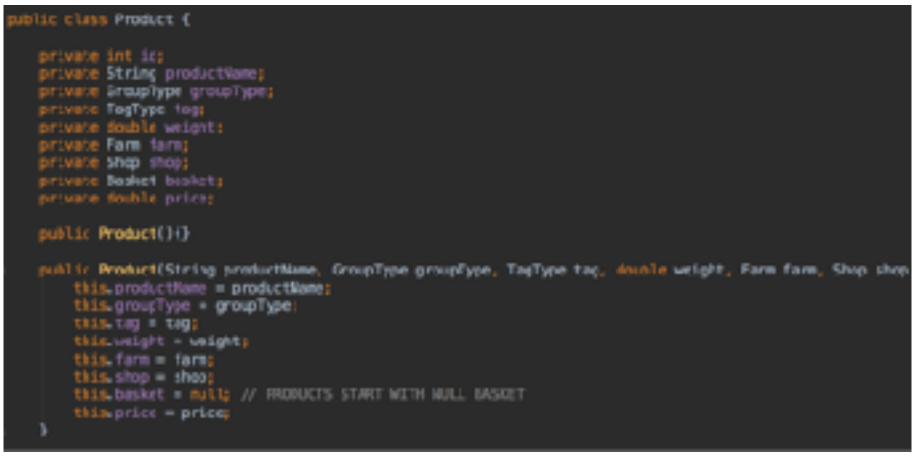


Description here

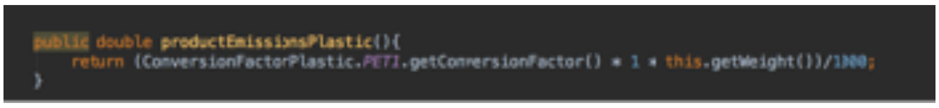
A diagram of inheritance, showing that car and motorbike class would inherit from the Vehicle class which is a parent to them.

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

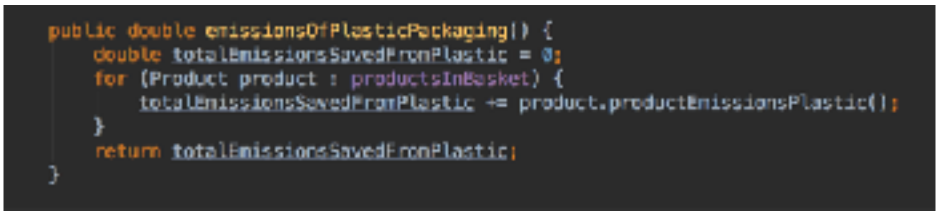
Paste Screenshot here



```
public class Product {

    private int id;
    private String productName;
    private GroupType groupType;
    private TagType tag;
    private double weight;
    private Farm farm;
    private Shop shop;
    private Basket basket;
    private double price;

    public Product(){}

    public Product(String productName, GroupType groupType, TagType tag, double weight, Farm farm, Shop shop
        this.productName = productName;
        this.groupType = groupType;
        this.tag = tag;
        this.weight = weight;
        this.farm = farm;
        this.shop = shop;
        this.basket = null; // PRODUCTS START WITH NULL BASKET
        this.price = price;
    }
```



```
public double productEmissionsPlastic(){
    return (ConversionFactorPlastic.PET1.getConversionFactor() * 1 * this.getWeight())/1000;
}
```



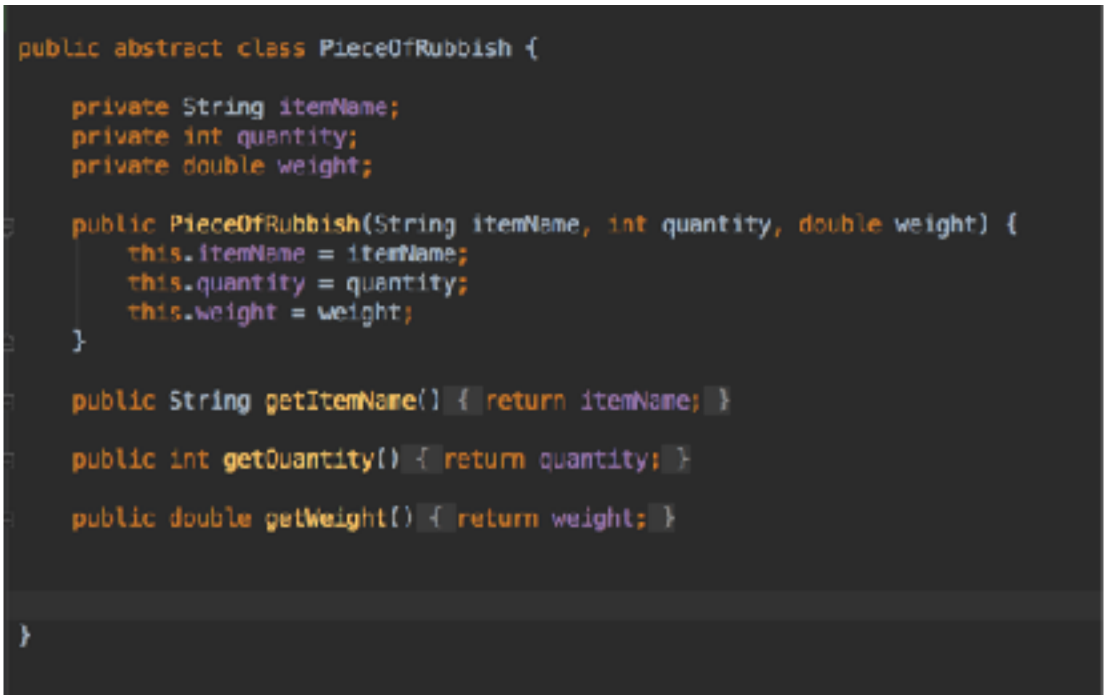
```
public double emissionsOfPlasticPackaging() {
    double totalEmissionsSavedFromPlastic = 0;
    for (Product product : productsInBasket) {
        totalEmissionsSavedFromPlastic += product.productEmissionsPlastic();
    }
    return totalEmissionsSavedFromPlastic;
}
```

Description here

Screenshots show the encapsulation, showing that the variables of a class will be hidden from other classes. The methods of this function can be accessed only through the methods of their current class or through inheritance.

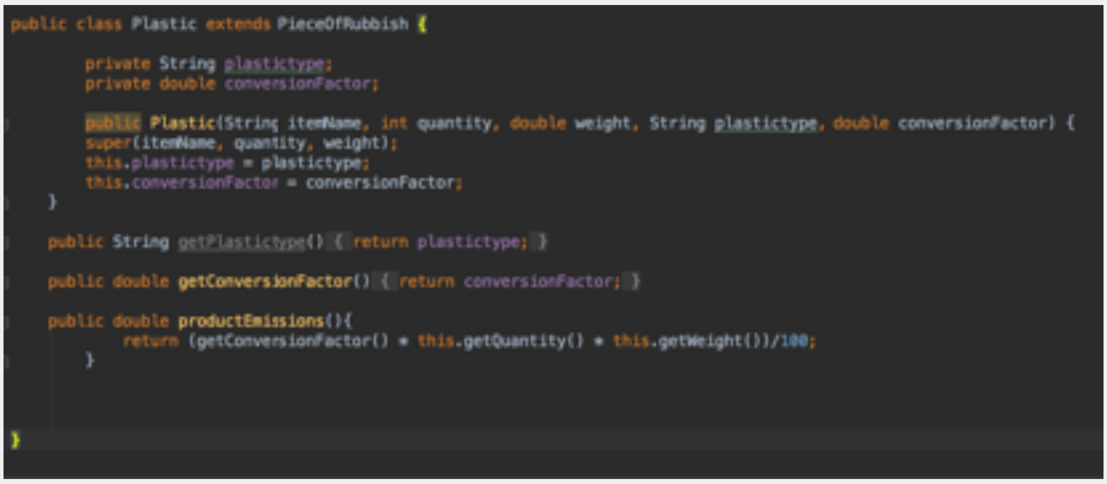
Unit	Ref	Evidence
I&T	I.T.2	Take a screenshot of the use of Inheritance in a program. Take screenshots of: *A Class *A Class that inherits from the previous class *An Object in the inherited class *A Method that uses the information inherited from another class.

Paste Screenshot here

A class	

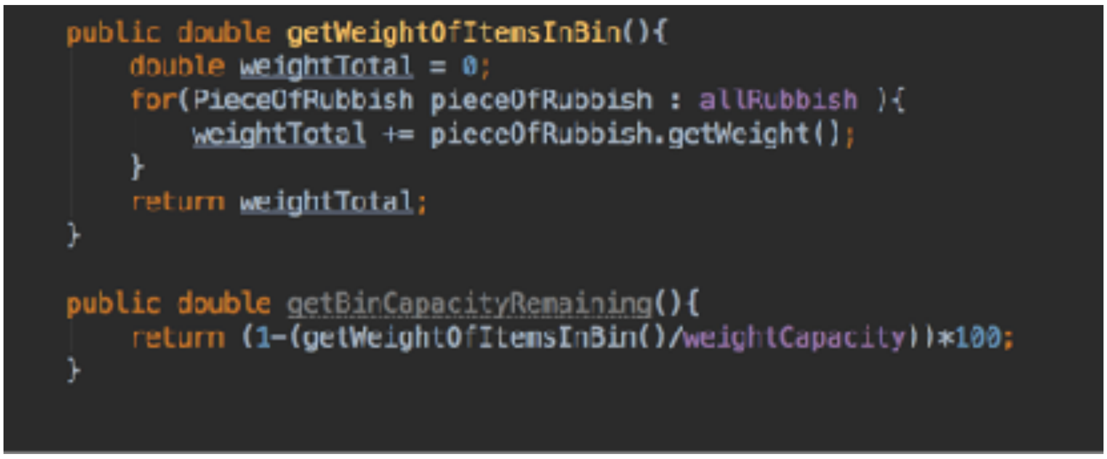
```
public abstract class PieceOfRubbish {

    private String itemName;
    private int quantity;
    private double weight;

    public PieceOfRubbish(String itemName, int quantity, double weight) {
        this.itemName = itemName;
        this.quantity = quantity;
        this.weight = weight;
    }

    public String getItemName() { return itemName; }

    public int getQuantity() { return quantity; }

    public double getWeight() { return weight; }


}
```

A class that inherits



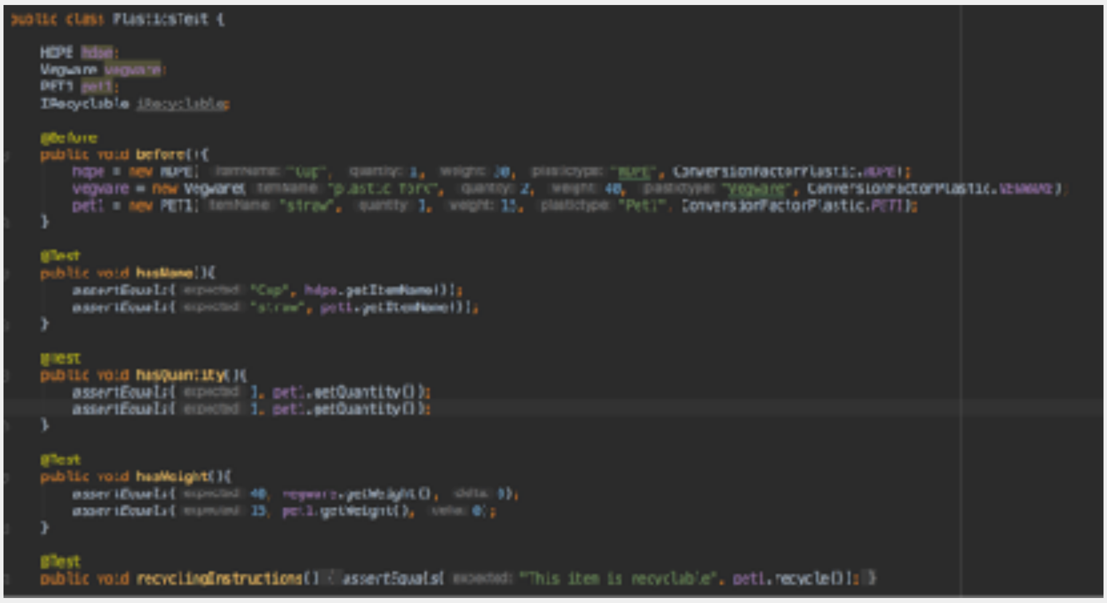
```
public class Plastic extends PieceOfRubbish {

        private String plastictype;
        private double conversionFactor;

        public Plastic(String itemName, int quantity, double weight, String plastictype, double conversionFactor) {
        super(itemName, quantity, weight);
        this.plastictype = plastictype;
        this.conversionFactor = conversionFactor;
    }

    public String getPlastictype() { return plastictype; }

    public double getConversionFactor() { return conversionFactor; }

    public double productEmissions(){
            return (getConversionFactor() * this.getQuantity() * this.getWeight())/100;
        }


}
```

A method

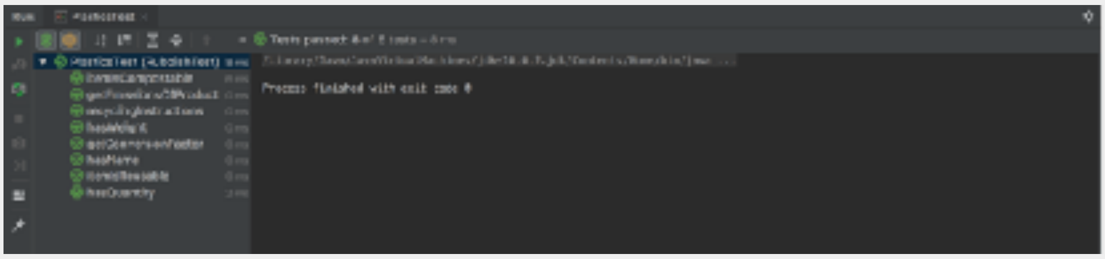


```
    public double getWeightOfItemsInBin(){
        double weightTotal = 0;
        for(PieceOfRubbish pieceOfRubbish : allRubbish ){
            weightTotal += pieceOfRubbish.getWeight();
        }
        return weightTotal;
    }

    public double getBinCapacityRemaining(){
        return (1-(getWeightOfItemsInBin()/weightCapacity))*100;
    }
```

A test and passing



```
public class PlasticsTest {

    HDPE hdpe;
    Vegware vegware;
    PET1 pet1;
    IRecyclable iRecyclable;

    @Before
    public void before(){
        hdpe = new HDPE( itemName: "Cup", quantity: 1, weight: 30, plastictype: "HDPE", ConversionFactorPlastic.HDPE);
        vegware = new Vegware( itemName: "plastic fork", quantity: 2, weight: 40, plastictype: "Vegware", ConversionFactorPlastic.VEGWARE);
        pet1 = new PET1( itemName: "straw", quantity: 1, weight: 15, plastictype: "Pet1", ConversionFactorPlastic.PET1);
    }

    @Test
    public void hasName(){
        assertEquals( expected: "Cup", hdpe.getItemName());
        assertEquals( expected: "straw", pet1.getItemName());
    }

    @Test
    public void hasQuantity(){
        assertEquals( expected: 1, pet1.getQuantity());
        assertEquals( expected: 1, pet1.getQuantity());
    }

    @Test
    public void hasWeight(){
        assertEquals( expected: 40, vegware.getWeight(), delta: 0);
        assertEquals( expected: 15, pet1.getWeight(), delta: 0);
    }

    @Test
    public void recyclingInstructions() { assertEquals( expected: "This item is recyclable", pet1.recycle()); }
```



```
Run: PlasticsTest
Tests passed: [illegible]
▼ PlasticsTest [illegible]
    itemIsCompostable
    getEmissionsOfProduct
    recyclingInstructions
    hasWeight
    getConversionFactor
    hasName
    itemIsRecyclable
    hasQuantity

[illegible]

Process finished with exit code 0
```

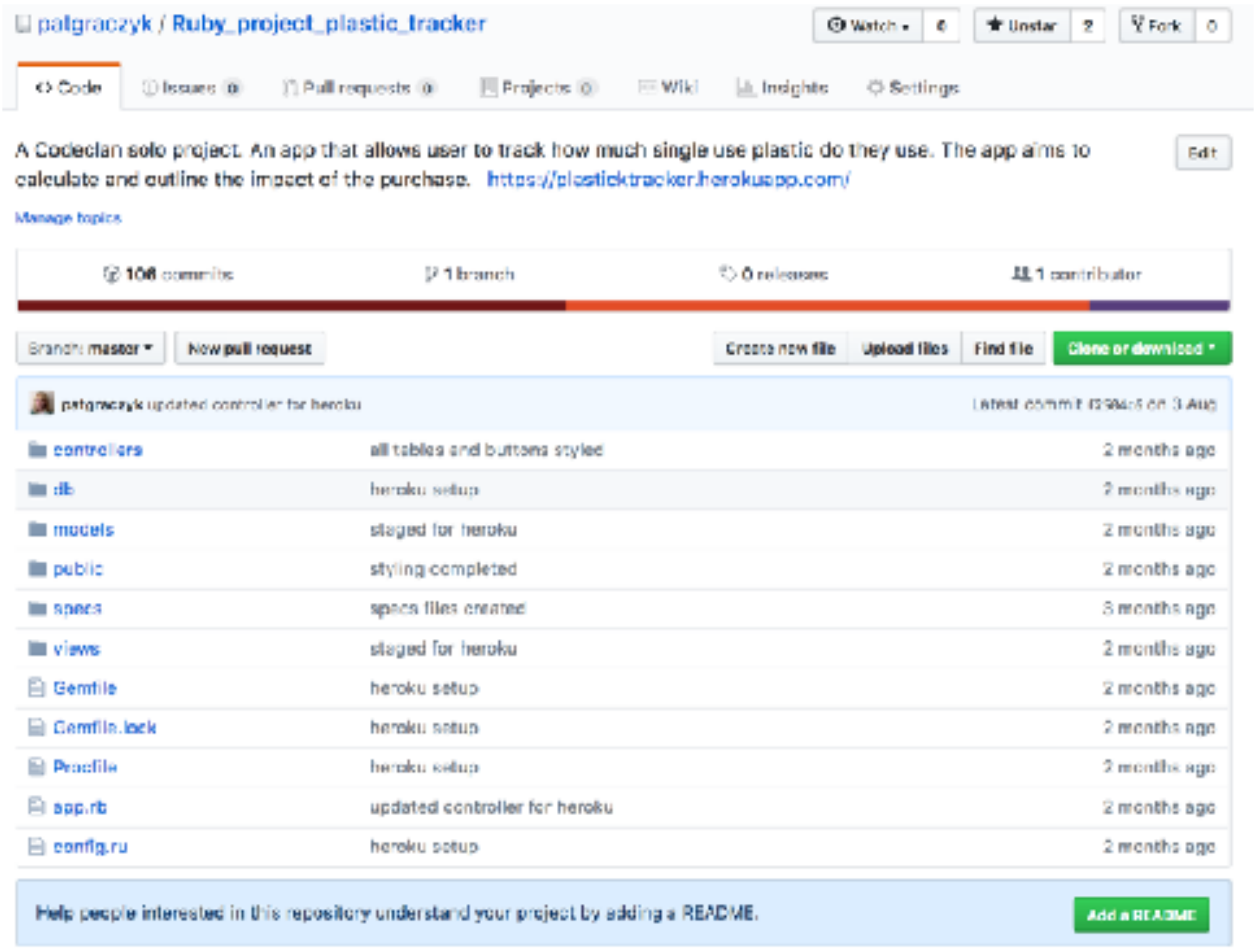
Description here

The code shows that the piece of rubbish is a parent class to the Plastic class. This means that the functions and properties from the Piece of Rubbish class are available and can be used by the Plastic class. This is demonstrated in the other screenshots where tests of the methods from the parent class are used to test the Plastic class and passing.

Week 10

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

Paste Screenshot here



A Codeclan solo project. An app that allows user to track how much single use plastic do they use. The app aims to calculate and outline the impact of the purchase. https://plastictracker.herokuapp.com/

108 commits · 1 branch · 0 releases · 1 contributor

patgraczyk updated controller for heroku — Latest commit [illegible] on 3 Aug

controllers	all tables and buttons styled	2 months ago
db	heroku setup	2 months ago
models	staged for heroku	2 months ago
public	styling completed	2 months ago
specs	specs files created	3 months ago
views	staged for heroku	2 months ago
Gemfile	heroku setup	2 months ago
Gemfile.lock	heroku setup	2 months ago
Procfile	heroku setup	2 months ago
app.rb	updated controller for heroku	2 months ago
config.ru	heroku setup	2 months ago

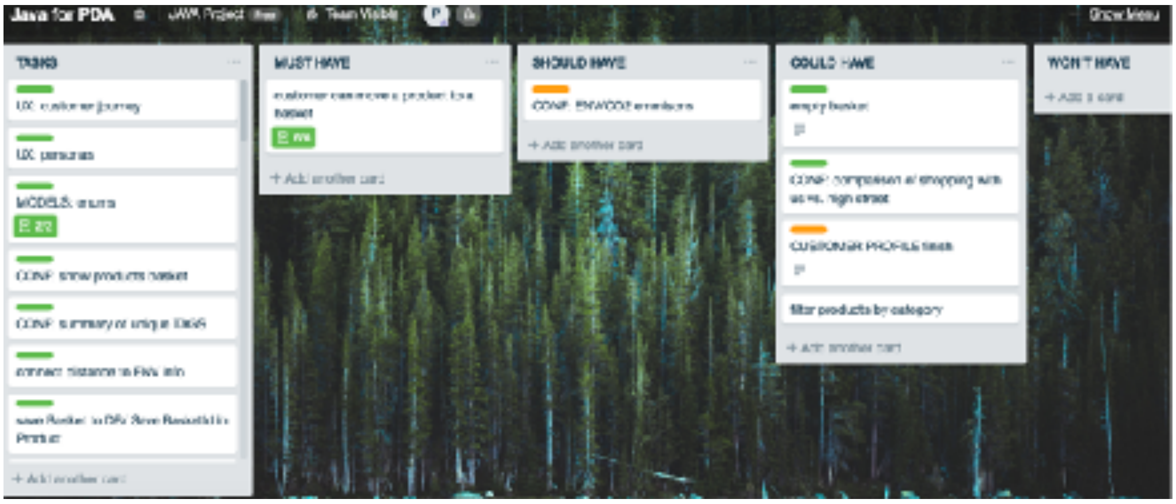
Help people interested in this repository understand your project by adding a README.

Description here

https://github.com/patgraczyk/Ruby_project_plastic_tracker

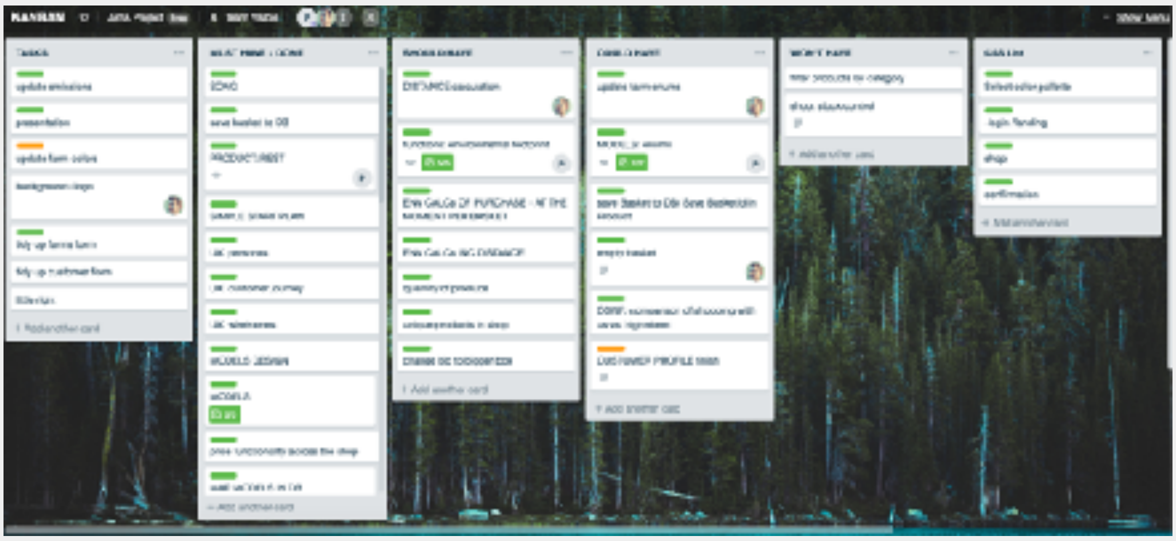
Unit	Ref	Evidence
P	P.12	Take screenshots or photos of your planning and the different stages of development to show changes.

Paste Screenshot here



Java for PDA | JAVA Project | Team Visible | Show Menu

TASKS	MUST HAVE	SHOULD HAVE	COULD HAVE	WON'T HAVE
UX customer journey	customer can move a product to a basket	CONF: ENCODE enumbers	empty basket	+ Add a card
UX personas	+ Add another card	+ Add another card	CONF: comparison of shopping with ... vs. high street	
MODELS: enums			CUSTOMER PROFILE team	
CONF: show products basket			filter products by category	
CONF: summary of unique ...			+ Add another card	
connect ... to ...				
save Basket to DB; Save Basket to Product				
+ Add another card				



KANBAN | JAVA Project | Team Visible | Show Menu

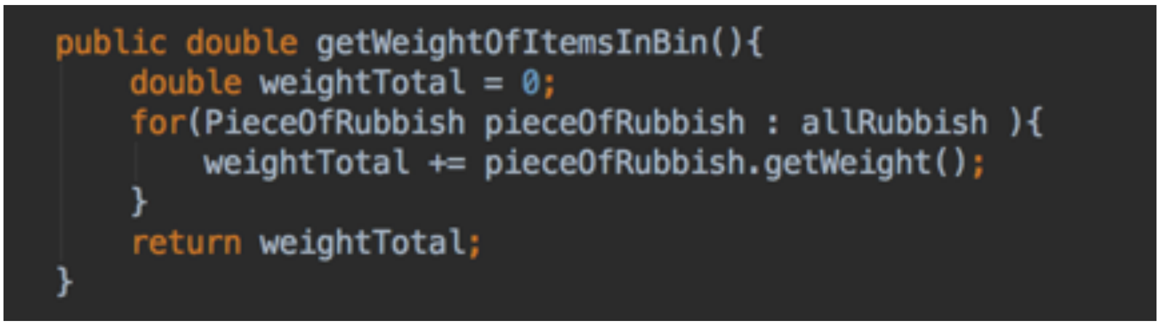
[illegible]

Description here

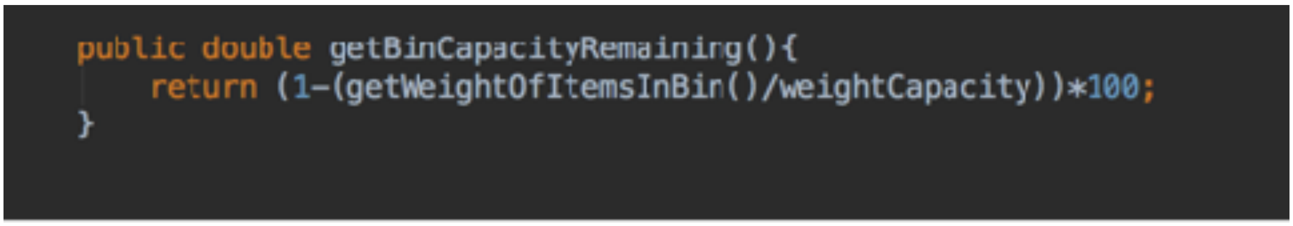
The screenshot shows the KANBAN Trello board at the different stages of the project. It clearly shows different set of tasks being added and moved as the coding and the project development continued.

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.

Paste Screenshot here



```
public double getWeightOfItemsInBin(){
    double weightTotal = 0;
    for(PieceOfRubbish pieceOfRubbish : allRubbish ){
        weightTotal += pieceOfRubbish.getWeight();
    }
    return weightTotal;
}
```



```
public double getBinCapacityRemaining(){
    return (1-(getWeightOfItemsInBin()/weightCapacity))*100;
}
```

Description here

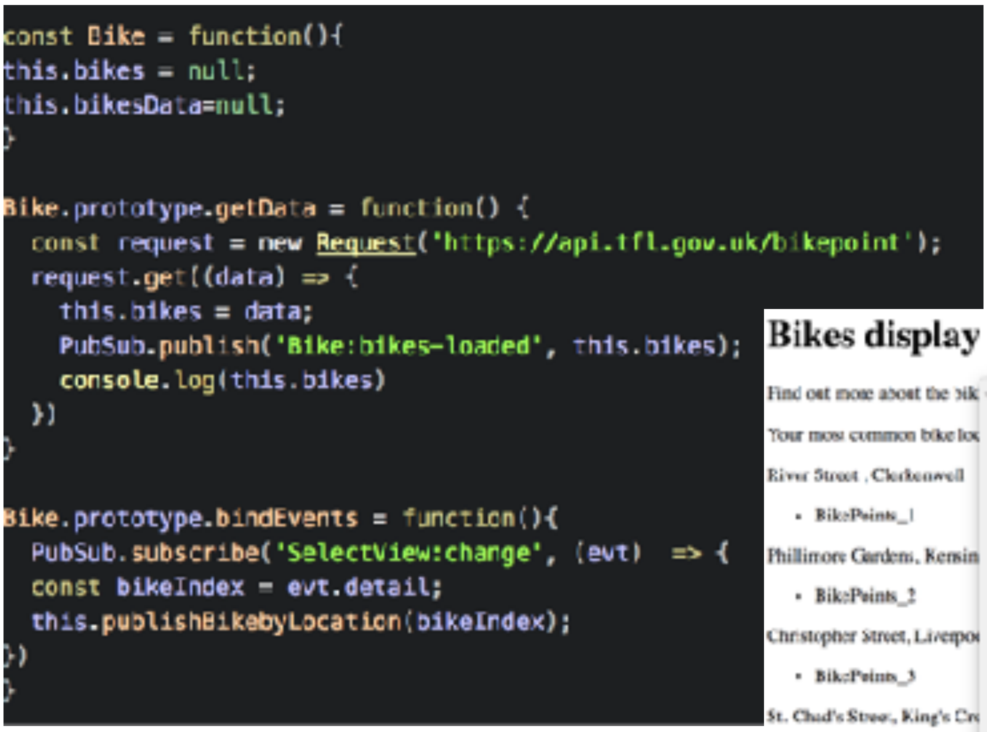
Algorithm number one is calculating the total weight of all of the instances of the bin class. It stores the total weight and iterates over all other bins and adding that value to the overall total it then returns that total.

The second one calculates the remaining capacity of that bin by taking the total weight, its capacity and calculating the % of the space that is left.

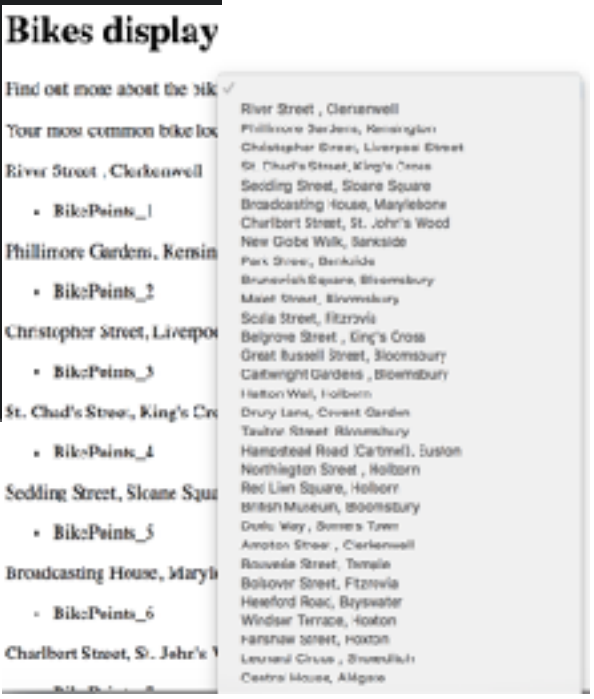
Week 12

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

Paste Screenshot here



```
const Bike = function(){
this.bikes = null;
this.bikesData=null;
}

Bike.prototype.getData = function() {
  const request = new Request('https://api.tfl.gov.uk/bikepoint');
  request.get((data) => {
    this.bikes = data;
    PubSub.publish('Bike:bikes-loaded', this.bikes);
    console.log(this.bikes)
  })
}

Bike.prototype.bindEvents = function(){
  PubSub.subscribe('SelectView:change', (evt)  => {
  const bikeIndex = evt.detail;
  this.publishBikebyLocation(bikeIndex);
})
}
```



Bikes display

Find out more about the bik

Your most common bike loc

River Street , Clerkenwell

- BikePoints_1

Phillimore Gardens, Kensin

- BikePoints_2

Christopher Street, Liverpo

- BikePoints_3

St. Chad's Street, King's Cr

- BikePoints_4

Sedding Street, Sloane Squa

- BikePoints_5

Broadcasting House, Maryl

- BikePoints_6

Charlbert Street, St. John's

River Street , Clerkenwell
Phillimore Gardens, Kensington
Christopher Street, Liverpool Street
St. Chad's Street, King's Cross
Sedding Street, Sloane Square
Broadcasting House, Marylebone
Charlbert Street, St. John's Wood
New Globe Walk, Bankside
Park Street, Bankside
Brunswick Square, Bloomsbury
Malet Street, Bloomsbury
Scala Street, Fitzrovia
Belgrove Street , King's Cross
Great Russell Street, Bloomsbury
Cartwright Gardens , Bloomsbury
Hatton Wall, Holborn
Drury Lane, Covent Garden
Taviton Street, Bloomsbury
Hampstead Road (Cartmel), Euston
Northington Street , Holborn
Red Lion Square, Holborn
British Museum, Bloomsbury
Doric Way , Somers Town
Ampton Street , Clerkenwell
Bouverie Street, Temple
Bolsover Street, Fitzrovia
Hereford Road, Bayswater
Windsor Terrace, Hoxton
Fanshaw Street, Hoxton
Leonard Circus , Shoreditch
Central House, Aldgate

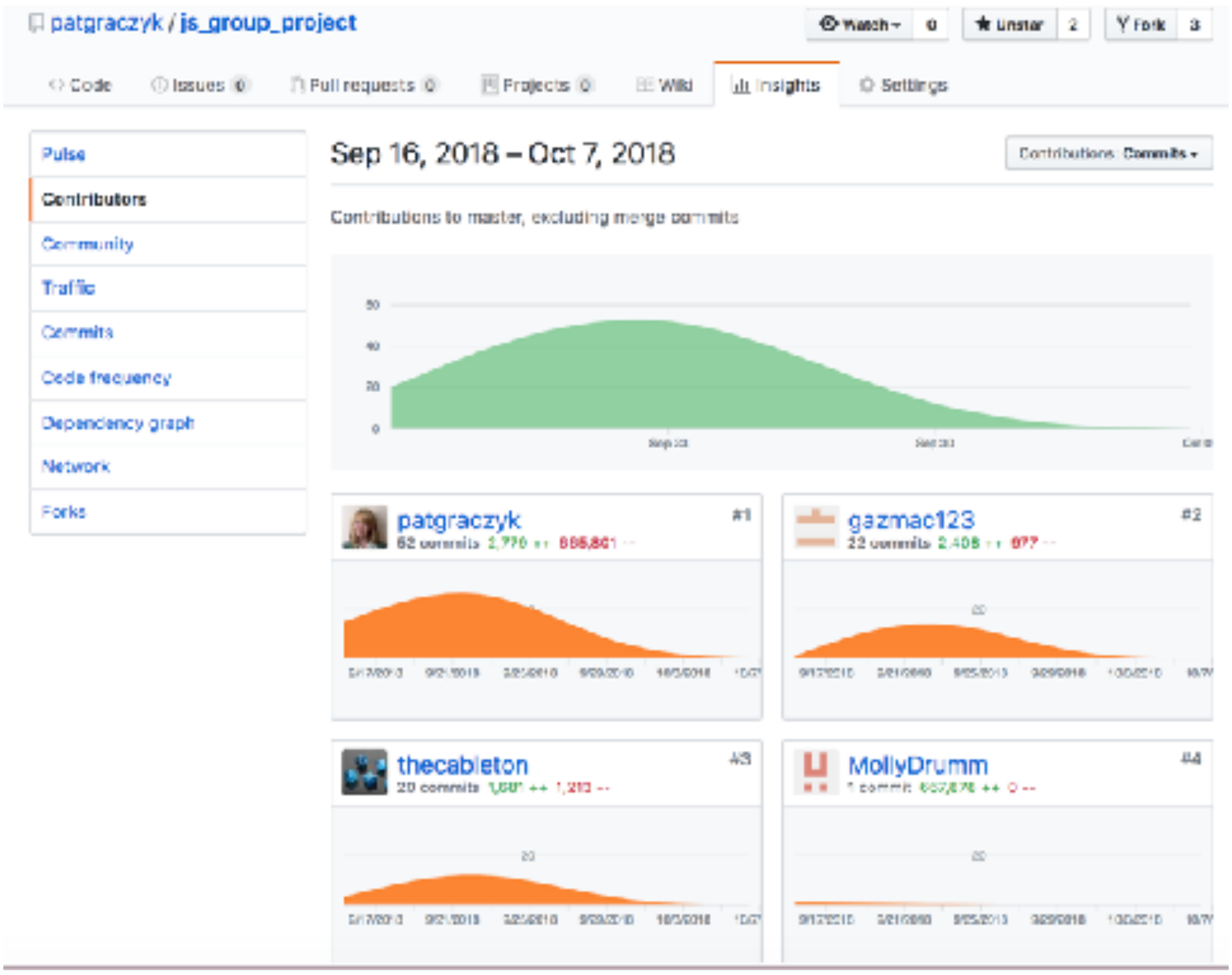
Description here

This program makes a request to the Transport for London API to get locations of all bikes (Boris bikes) in London. It then publishes that information through the 'Bikes:bikes-loaded' channel. The second screenshot shows the display through the dropdown and statically on the screen.

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.

Paste Screenshot here



Sep 16, 2018 – Oct 7, 2018

Contributions to master, excluding merge commits

patgraczyk #1
62 commits 2,770 ++ 886,861 --

gazmac123 #2
22 commits 2,408 ++ 877 --

thecableton #3
20 commits 1,887 ++ 1,213 --

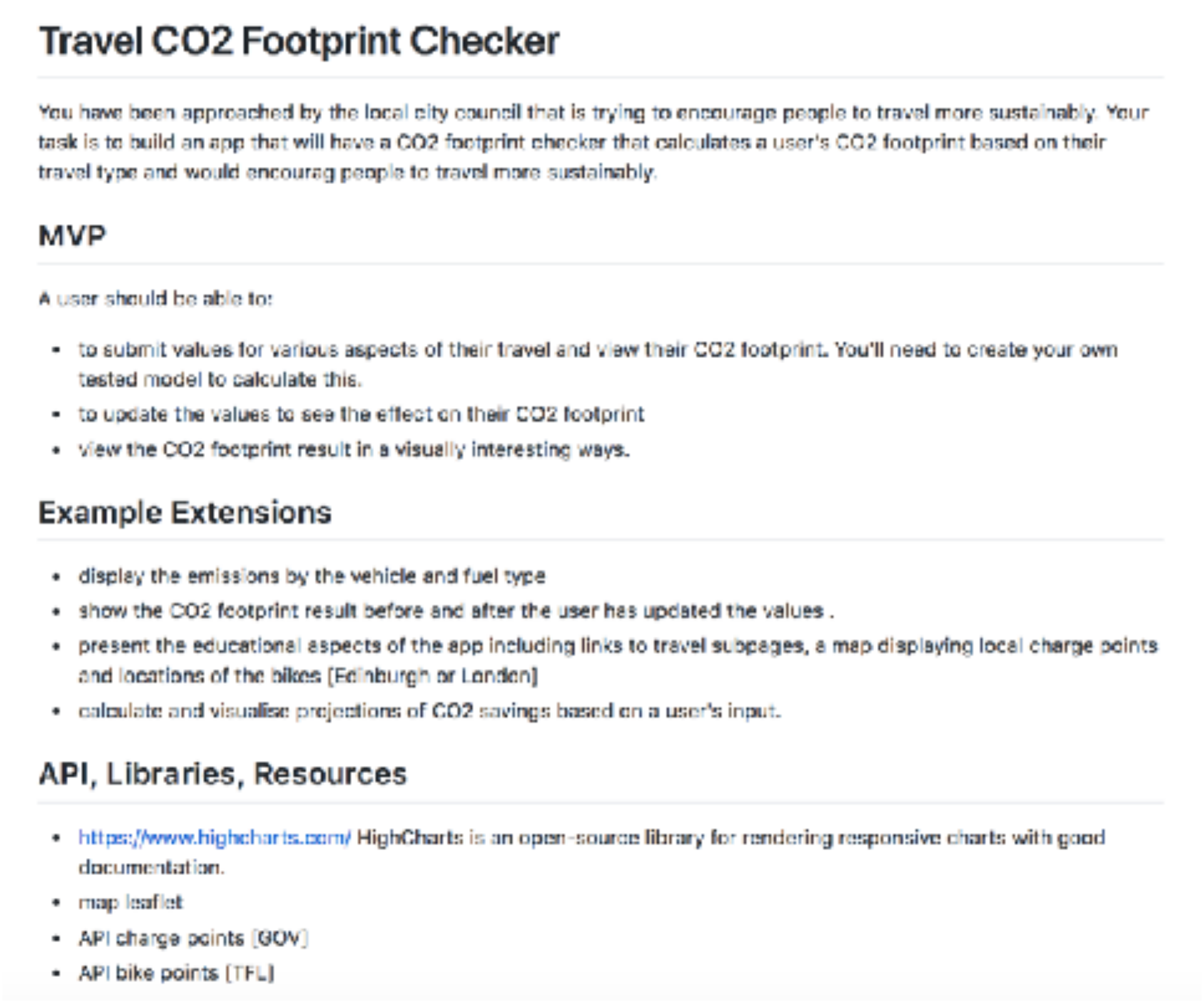
MollyDrumm #4
1 commit 857,878 ++ 0 --

Description here

Screenshot from our group project Github account.

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

Paste Screenshot here



Travel CO2 Footprint Checker

You have been approached by the local city council that is trying to encourage people to travel more sustainably. Your task is to build an app that will have a CO2 footprint checker that calculates a user's CO2 footprint based on their travel type and would encourag people to travel more sustainably.

MVP

A user should be able to:

- to submit values for various aspects of their travel and view their CO2 footprint. You'll need to create your own tested model to calculate this.
- to update the values to see the effect on their CO2 footprint
- view the CO2 footprint result in a visually interesting ways.

Example Extensions

- display the emissions by the vehicle and fuel type
- show the CO2 footprint result before and after the user has updated the values .
- present the educational aspects of the app including links to travel subpages, a map displaying local charge points and locations of the bikes (Edinburgh or London)
- calculate and visualise projections of CO2 savings based on a user's input.

API, Libraries, Resources

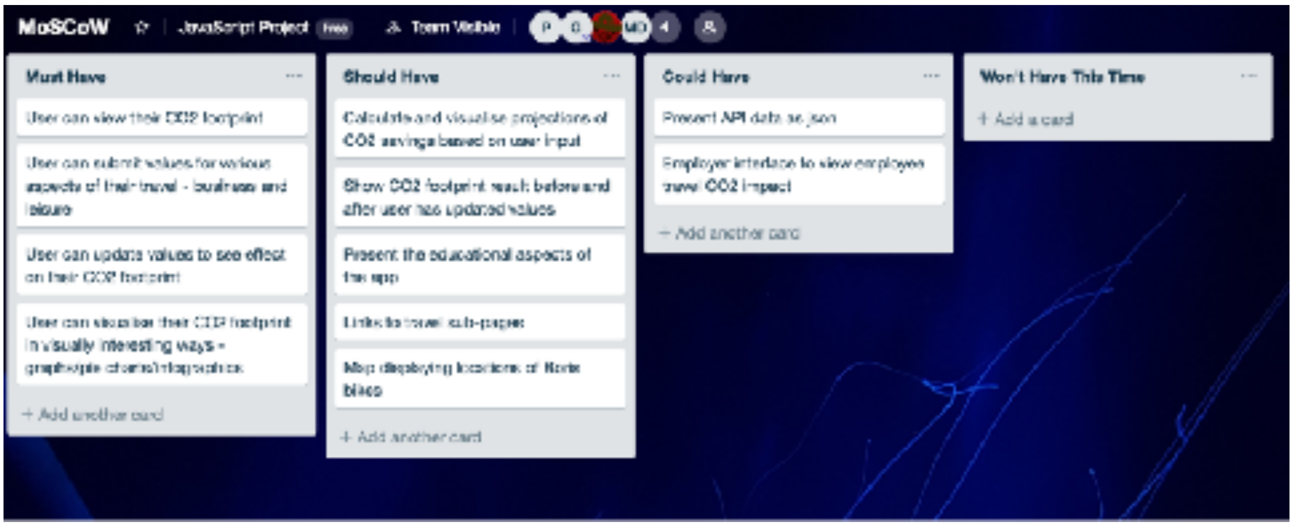
- https://www.highcharts.com/ HighCharts is an open-source library for rendering responsive charts with good documentation.
- map leaflet
- API charge points (GOV)
- API bike points (TFL)

Description here

A project brief that we have written for our JS group project.

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

<u>**Paste Screenshot here**</u>



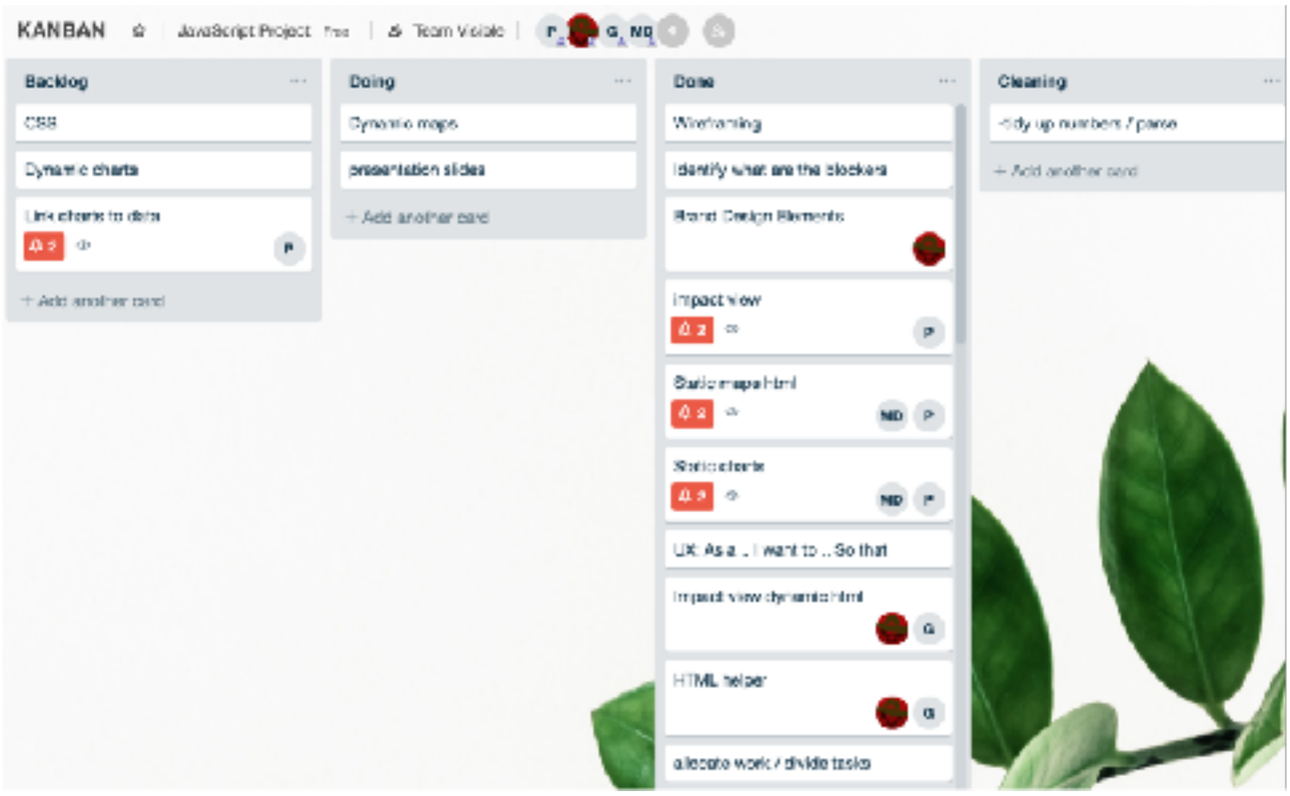
MoSCoW | JavaScript Project | Team Visible

Must Have
- User can view their CO2 footprint
- User can submit values for various aspects of their travel - business and leisure
- User can update values to see effect on their CO2 footprint
- User can visualise their CO2 footprint in visually interesting ways - graphs/pie charts/infographics
- + Add another card

Should Have
- Calculate and visualise projections of CO2 savings based on user input
- Show CO2 footprint result before and after user has updated values
- Present the educational aspects of the app
- Links to travel sub-pages
- Map displaying locations of Boris bikes
- + Add another card

Could Have
- Present API data as json
- Employer interface to view employee travel CO2 impact
- + Add another card

Won't Have This Time
- + Add a card



KANBAN | JavaScript Project | Team Visible

Backlog
- CSS
- Dynamic charts
- Link charts to data
- + Add another card

Doing
- Dynamic maps
- presentation slides
- + Add another card

Done
- Wireframing
- Identify what are the blockers
- Brand Design Elements
- Impact view
- Static maps html
- Static charts
- UX: As a ... I want to ... So that
- Impact view dynamic html
- HTML helper
- allocate work / divide tasks

Cleaning
- tidy up numbers / parse
- + Add another card

<u>**Description here**</u>

<u>**Screenshots**</u> of the KANBAN and MSOCOW Trello boards.

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

Paste Screenshot here

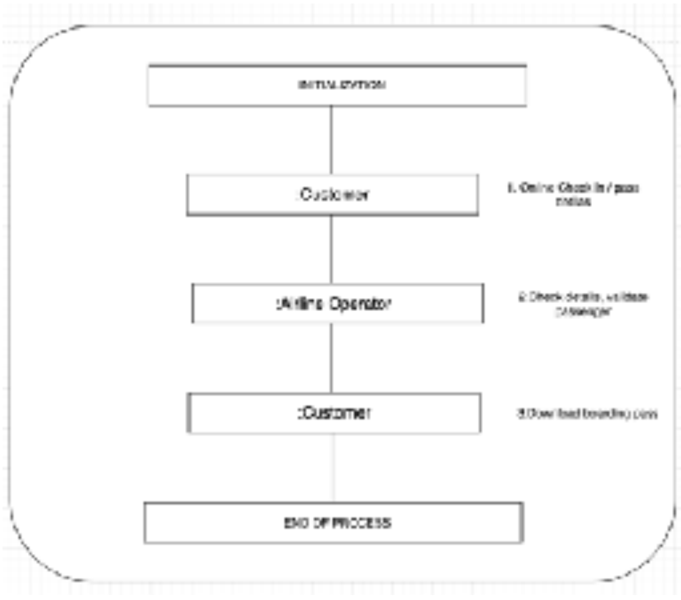
Acceptance Criteria	**Expected Result / Output**	**Pass / Fail**
User should be able to record new product they purchased	On the landing page and in the nav bar there is a category record product. This is a form with the fields to be filled in. Upon saving data is persisted to the DB.	PASS
User should be able to record a new plastic type	In the nav bar there is a category for plastics that allows user to record a new plastic type. Nav bar item is linked to the form that contains fields to be filled in and a save button. Upon saving data is persisted to the DB.	PASS
User should be able to view all products	In the navbar there is an item linked to the list of all recorded items. That list is in sync with the data persisted to the db.	PASS
User should be able to view transactions by plastic type	On the 'products' subpage underneath the list of all products there is a link to view products by a plastic type. On click this displays a list of all products of a specific plastic type.	PASS

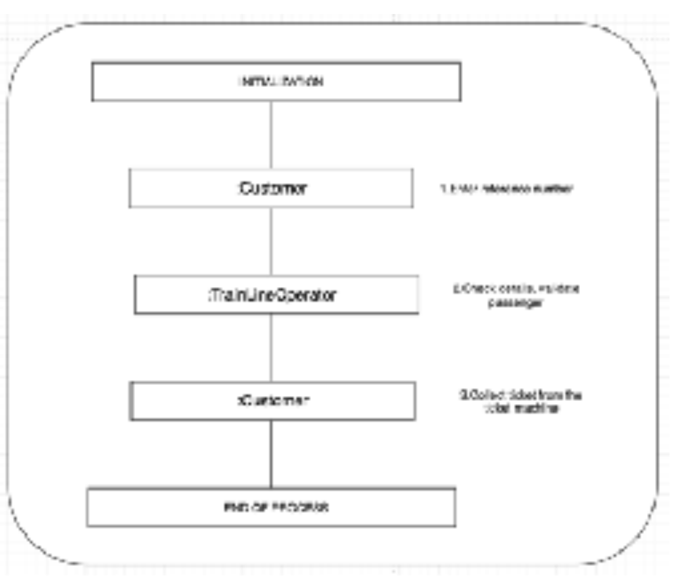
Description here

The table above shows the acceptance criteria and the test plan. It identifies key functions that the user should be able to perform while utilising the mobile / web app. It then outlines what the expected result is and if under those circumstances the test is passing.

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

<u>Paste Screenshot here</u>



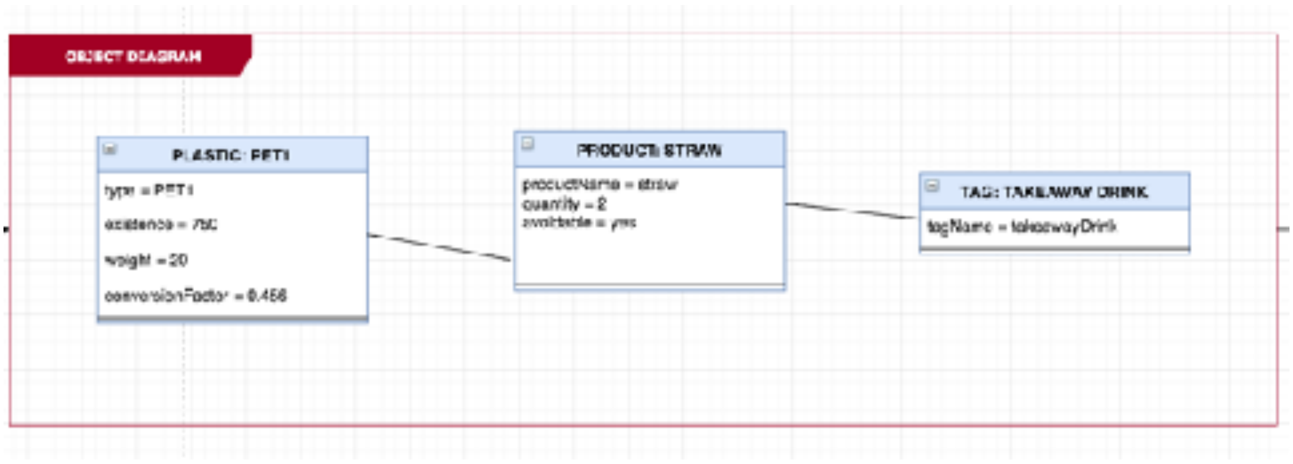

INITIALISATION
:Customer
1. Enter reference number
:TrainLineOperator
2.Check details, validate passenger
:Customer
3.Collect ticket from the ticket machine
END OF PROGRAM

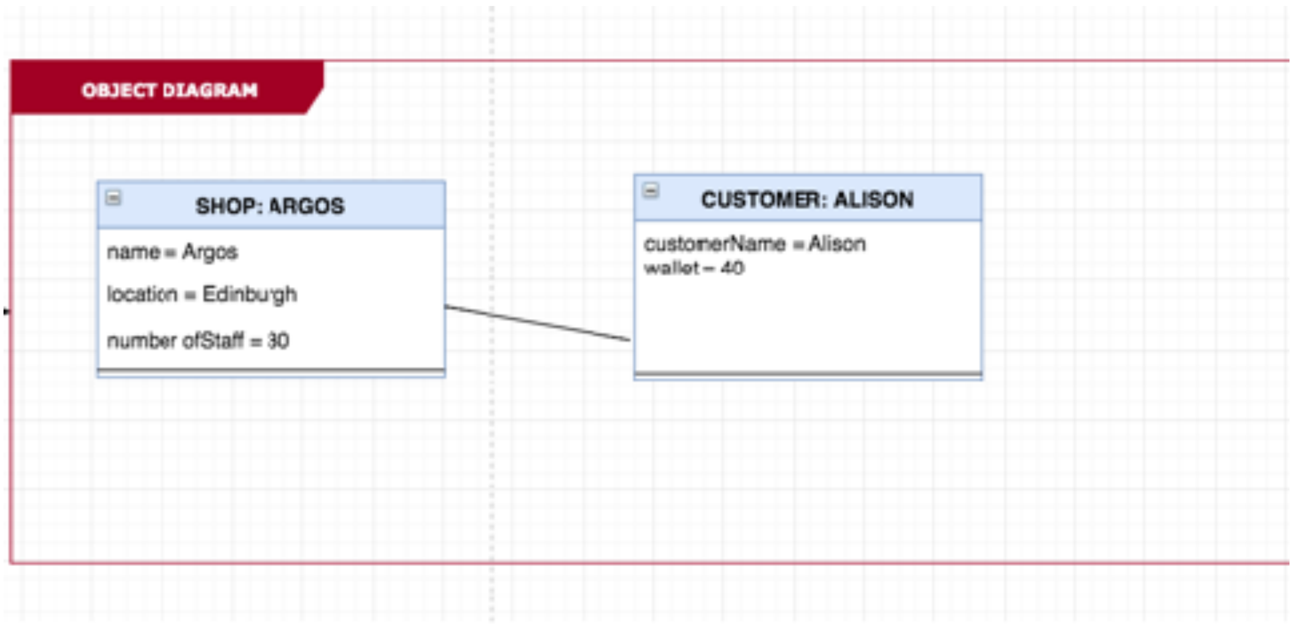
<u>Description here</u>

Two system interaction diagrams above. One is for an airline ticket booking (left) and second is for online train booking system (right).

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

<u>Paste Screenshot here</u>





<u>Description here</u>

Two object diagrams. The top one is for the plastic product, that has the plastic type and the tag. Second one is for a customer and an online shop.

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

Paste Screenshot here

User can record a unique transaction (new plastic used)	FAILED	Persist information to the db with a unique ID	PASSED
User can create a new tag	FAILED	Persist information to the db with a user input	PASSED
User can record a new plastic type	FAILED	Persist information to the db with a user input, assign unique ID	PASSED
User can view transactions by plastic	FAILED	Create relational DB with a foreign keys of plastic.	PASSED
User can assign a date to the purchase	FAILED	Use Date class accessible in each language and assign it. Save as a timestamp in SQL.	PASSED

Description here

A bug tracking report that shows what functionality should the web app have and highlights when it fails. It then identifies what needs to be done to fix the error and shows the new PASSED status.