CodeClan_

Bridging the digital skills gap.

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.

Please fill in each point with screenshot or diagram and description of what you are showing.

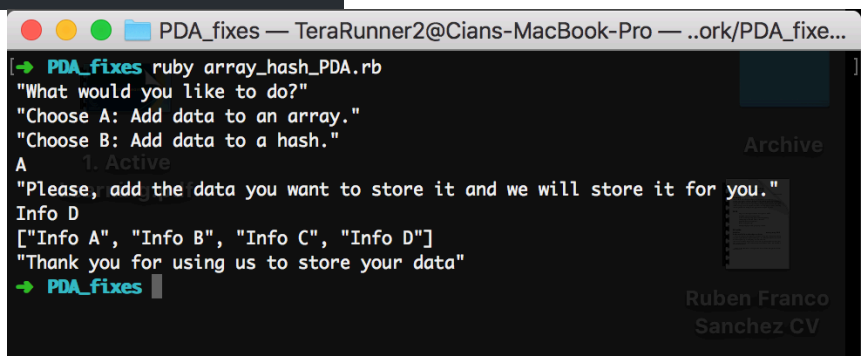
Each point requires details that cover each element of the Assessment Criteria, along with a brief description of the kind of things you should be showing.

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
		Description:

Paste Screenshot here

```ruby
@array_data = ["Info A", "Info B", "Info C"]

p "What would you like to do?"
p "Choose A: Add data to an array."
p "Choose B: Add data to a hash."

def start
  @input_start = gets.chomp

  case @input_start
    when "A"
      array_screenshot
    when "B"
      hash_screenshot
  end
end

def array_screenshot
  p "Please, add the data you want to store it and we will store it for you."
  @input = gets.chomp
  @array_data.push(@input)
  p @array_data
  p "Thank you for using us to store your data"
end

def hash_screenshot
  p "Please, to be add in our service, we will need your name and age."
  p "Please, answer the next questions please."
  p "Please, add your name."
  @input_name = gets.chomp
  p "Please, add your age."
  @input_age = gets.chomp
  @user = Hash.new
  @user[:name] = @input_name
  @user[:age] = @input_age
  p @user
  p "Thank you for creating a user."
end

start
```



Description here

Designed a function that takes an input from the user and add it to the array. It prints the array so the customer knows that his data is already on 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
		Description:

Paste Screenshot here

```ruby
@array_data = ["Info A", "Info B", "Info C"]

p "What would you like to do?"
p "Choose A: Add data to an array."
p "Choose B: Add data to a hash."

def start
  @input_start = gets.chomp

  case @input_start
    when "A"
      array_screenshot
    when "B"
      hash_screenshot
  end
end

def array_screenshot
  p "Please, add the data you want to store it and we will store it for you."
  @input = gets.chomp
  @array_data.push(@input)
  p @array_data
  p "Thank you for using us to store your data"
end

def hash_screenshot
  p "Please, to be add in our service, we will need your name and age."
  p "Please, answer the next questions please."
  p "Please, add your name."
  @input_name = gets.chomp
  p "Please, add your age."
  @input_age = gets.chomp
  @user = Hash.new
  @user[:name] = @input_name
  @user[:age] = @input_age
  p @user
  p "Thank you for creating a user."
end

start
```

```
[➜ PDA_fixes ruby array_hash_PDA.rb
"What would you like to do?"
"Choose A: Add data to an array."
"Choose B: Add data to a hash."
B
"Please, to be add in our service, we will need your name and age."
"Please, answer the next questions please."
"Please, add your name."
Ruben
"Please, add your age."
34
{:name=>"Ruben", :age=>"34"}
"Thank you for creating a user."
```

Description here

Designed a function that takes data from the customer and creates a hash with it. It prints back the hash.

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
		Description:

Paste Screenshot here

```
def self.all()
  sql = "SELECT * FROM stars"
  values = []
  stars = SqlRunner.run(sql, values)
  result = stars.map { |star| Star.new(star) }
  return result
end
```

Description here

Designed a function to search on the database and return the stars data from the database.

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
		Description:

Paste Screenshot here

```
def self.all_sorted_by_f_name()
  sql = "SELECT stars.first_name, stars.last_name FROM stars ORDER BY first_name ASC"
  values = []
  stars = SqlRunner.run(sql, values)
  result = stars.map {|star| Star.new(star)}
  return result
end
```

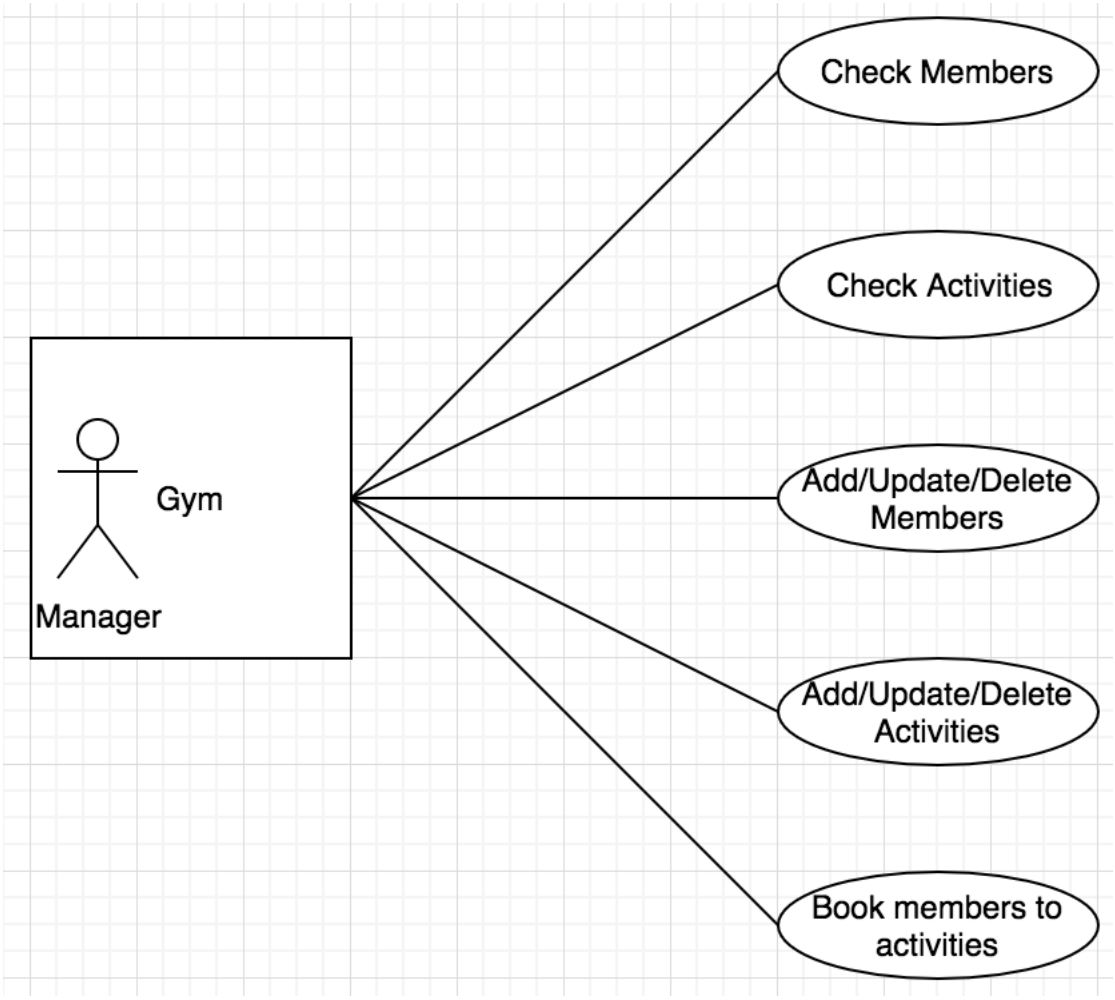
Description here

Modified a bit the previous code sorting the data by the first name.

Week 5 and 6

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

Paste Screenshot here

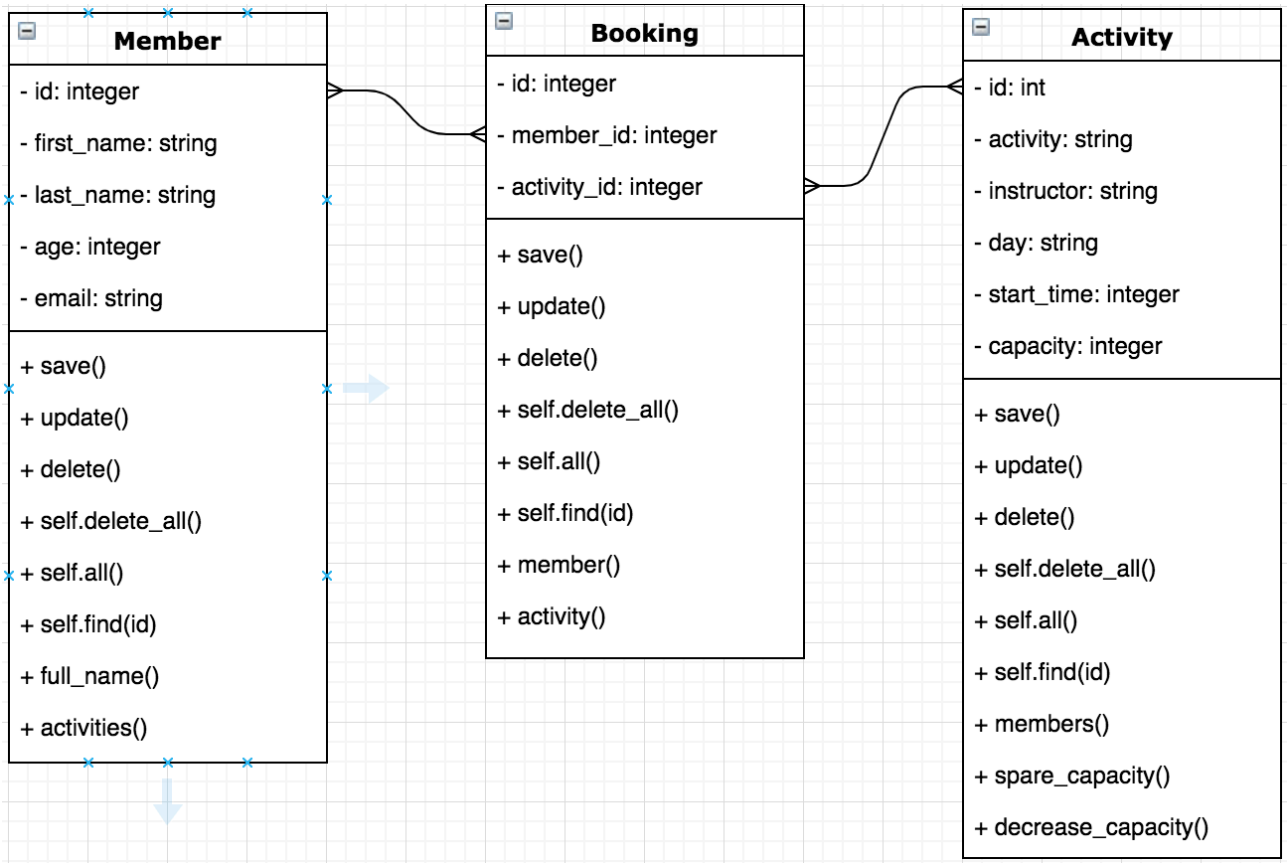


Description here

Case diagram for a manager and the options he/she has.

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

Paste Screenshot here

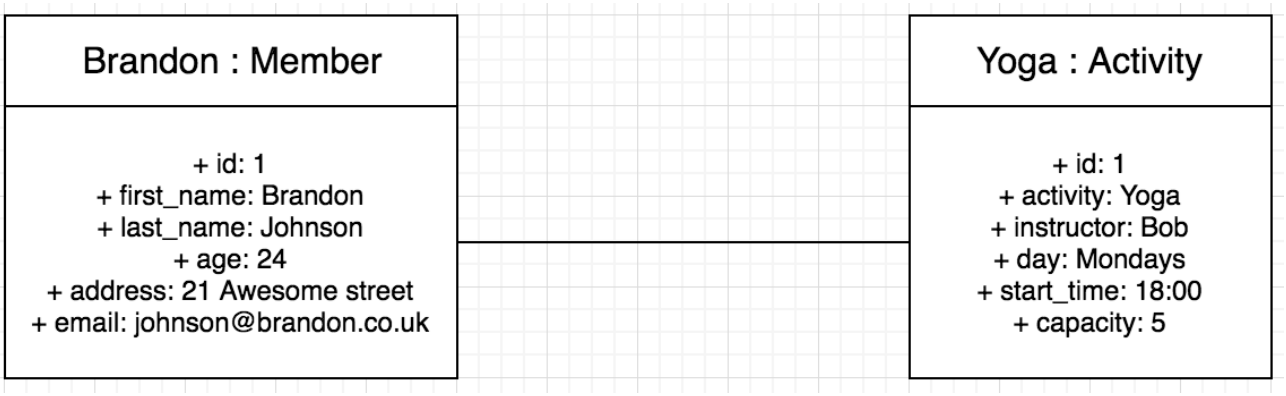


Description here

Class diagram showing the three main classes.

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

Paste Screenshot here

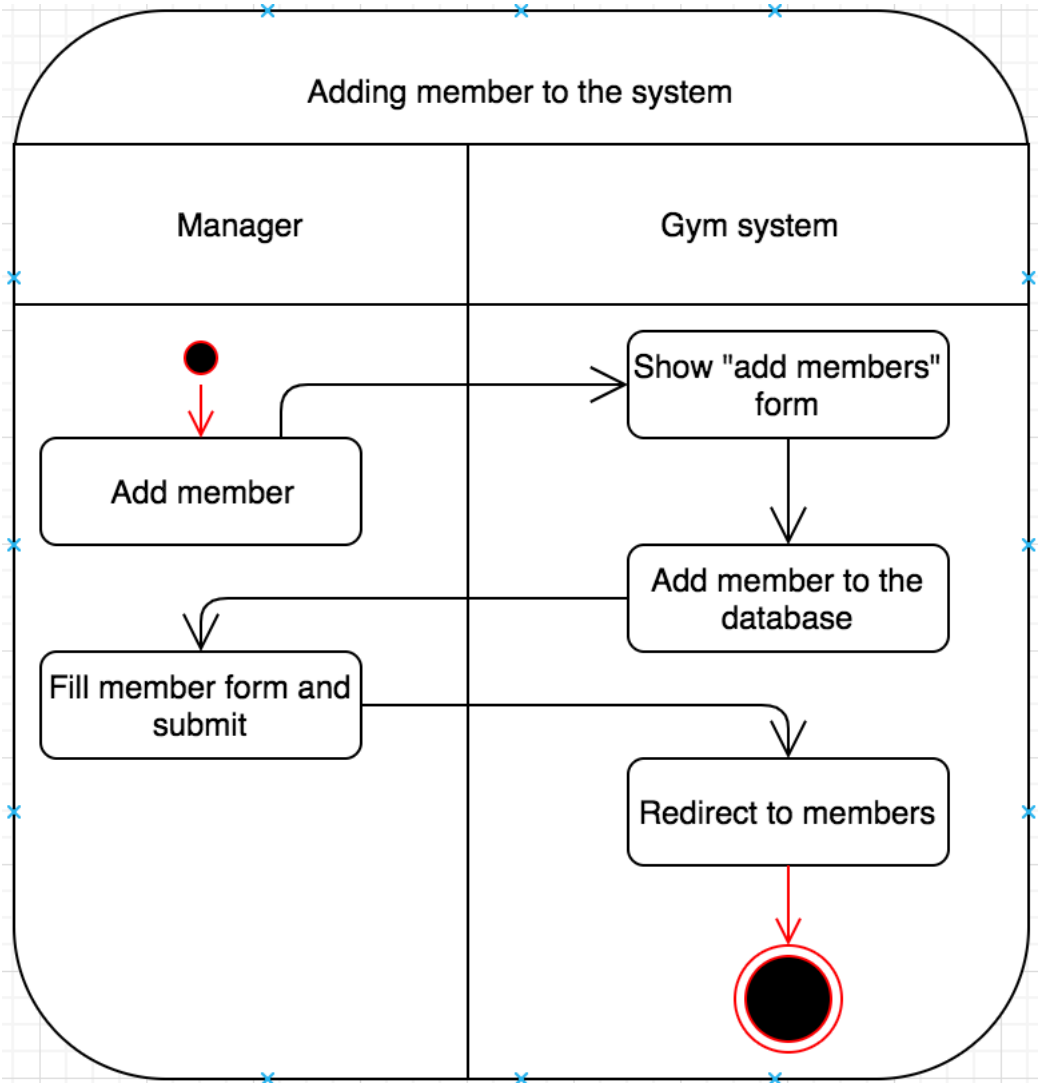


Description here

Object diagram from showing member (Brandon) - activity (Member) relation.

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

Paste Screenshot here



Description here

Activity diagram “adding a member” from my project

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
		Description:

Paste Screenshot here

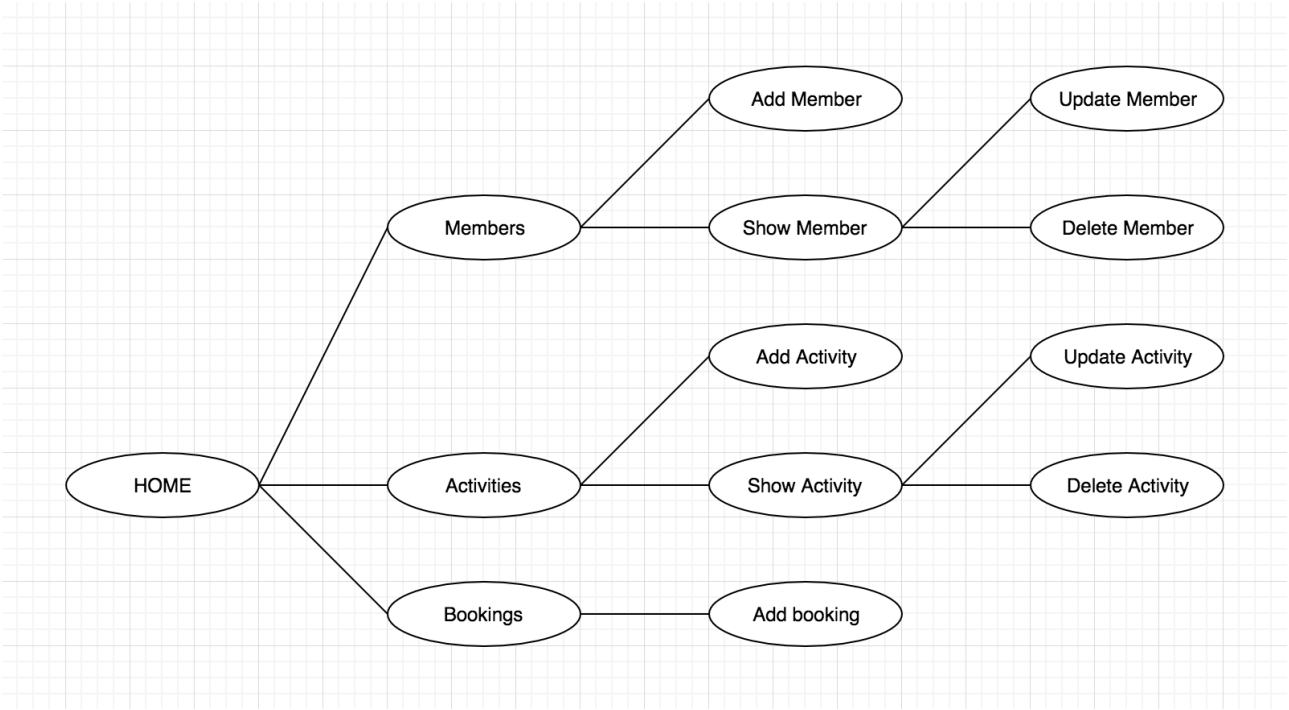
	Implementation Constraint	Solution
Hardware and software platforms	1-What could be a constraint on the product? Conflicts with another web browsers. 2- How could be a constraint of the product? When trying to open it with a non updated web browser the application won't launch. 3- Why is it a problem? You cannot use the application.	Use universals fonts, coding, having the web browsers updated, etc…
Performance requirements	1- Could go slow in slower machines. 2 - It would frustrate the user if the information takes time to load. 3 - The user could look for another faster product.	Refactor the code and test the application on several machines with different specifications.
Persistent storage and transactions	1 - Need a place to store the database 2 - If the user does not have enough space. Could be a problem. 3 - If the program need the space, and the user doesn't have it. It is not going to be installed.	Tell the customer that he will need space in his computer/server to store data or try to make a cloud server and store the information on the cloud.
Usability	1 - Some users could be colourblind people. 2 - If the application depends of the difference between colours, could be a problem. 3 - Colourblind people won't enjoy or even being able to continue with the application.	Use better contrast colours in CSS.
Budgets	1 - Not enough budget for the project. 2 - If there is not enough budget, some application will not be able to finish, or even the full software. 3 - If there is no application, no pay.	Reduce CSS and html style keeping the functionality. Maybe forget some extensions and keep it clean and simple.
Time	1 - Not enough time for everything. 2 - If the project has extensions, it would be impossible to reach them if the MVP is not achieved. 3 - Extensions normally improve the user experience.	Reduce time from other options and/or eliminate non-essentials extensions

Description here

Implementation constraint table with some examples from a possible web application.

Unit	Ref	Evidence
P	P.5	User Site Map
		Description:

Paste Screenshot here



Description here

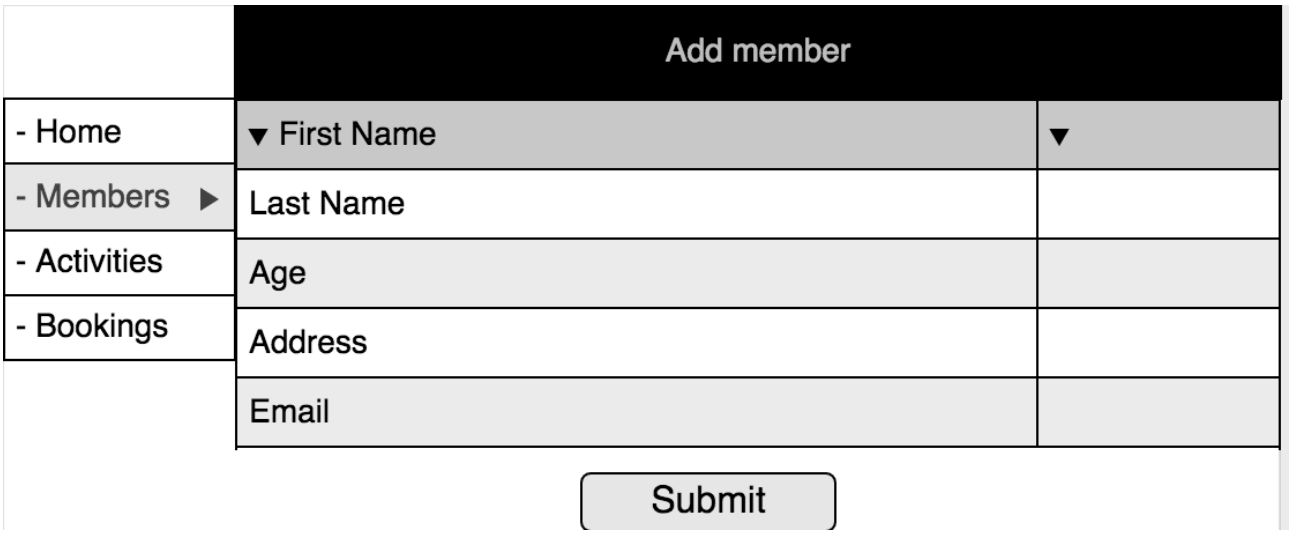
User site map from a project with three classes.

Unit	Ref	Evidence
P	P.6	2 Wireframe Diagrams
		Description:

Paste Screenshot here

Wireframe 1

Wireframe 2

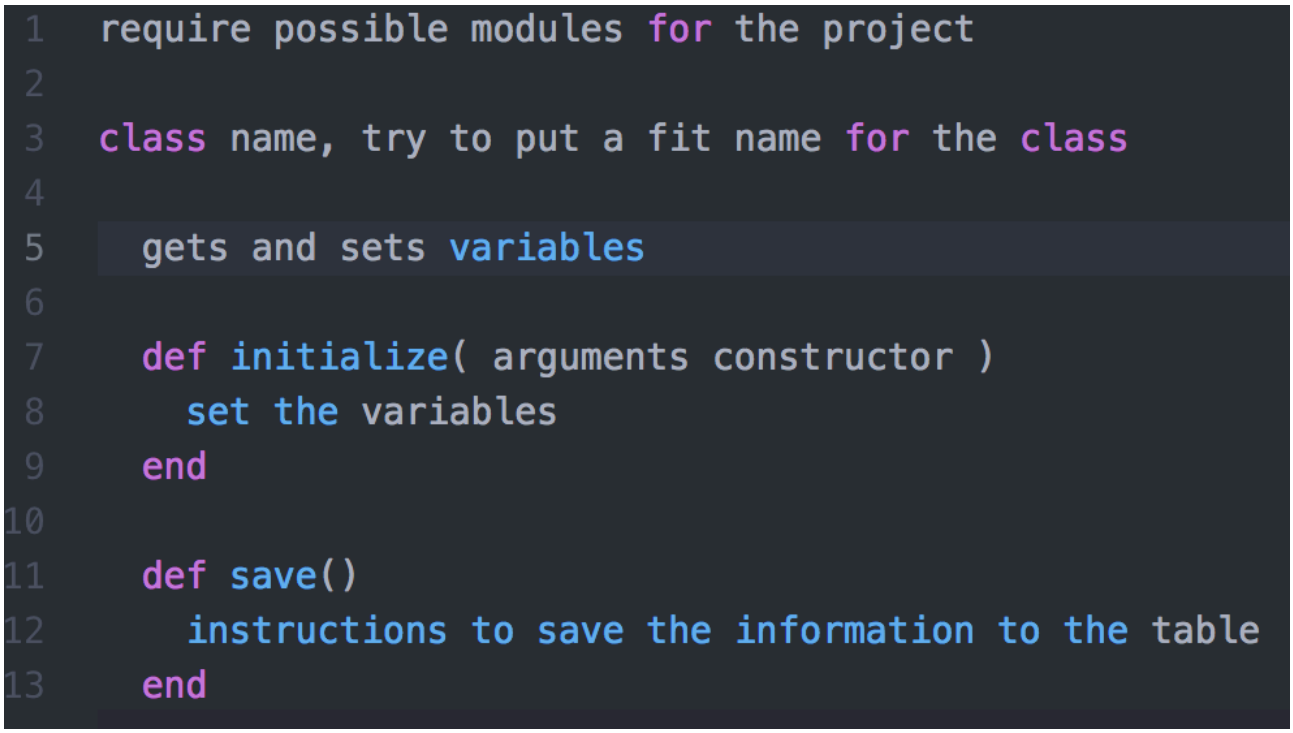


Description here

Wireframe 1 shows a possible members list from a project.
Wireframe 2 shows a form to add a new member.

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

Paste Screenshot here

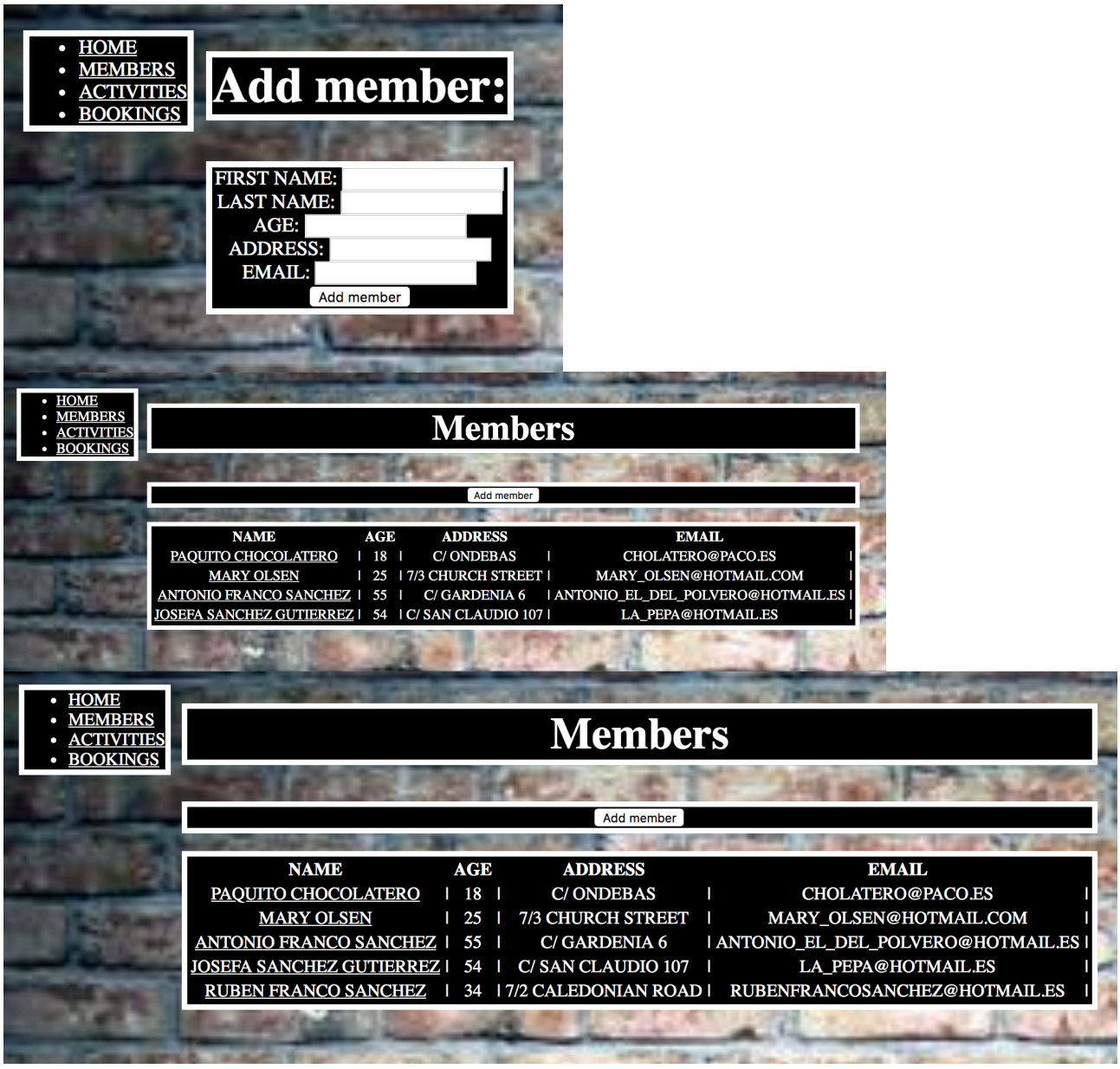


Description here

Pseudocode from the class booking and the first method, save.

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
		Description:

Paste Screenshot here

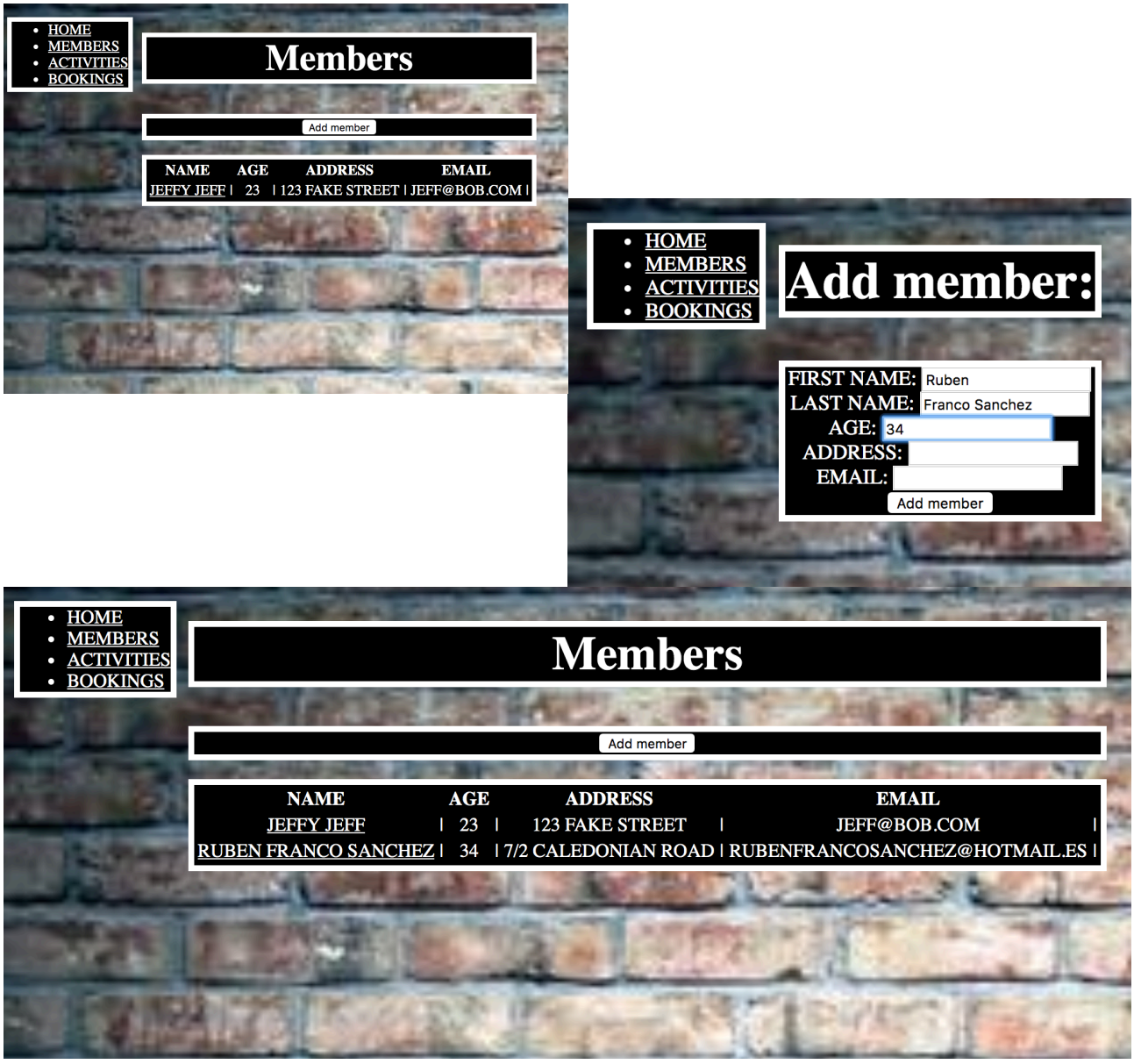


Description here

Showing how to create a new member.

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
		Description:

Paste Screenshot here

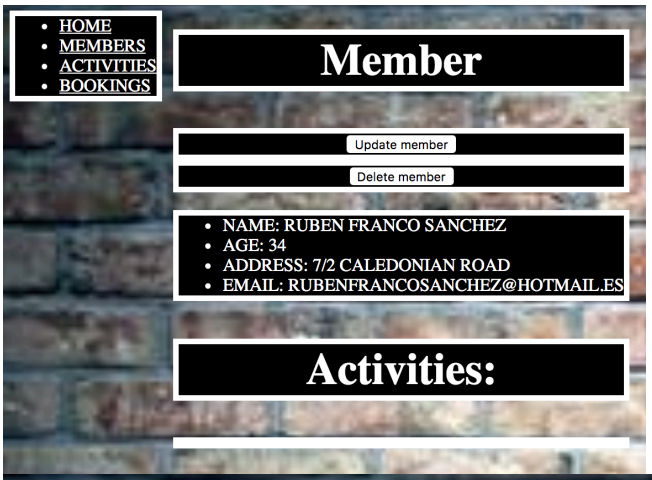


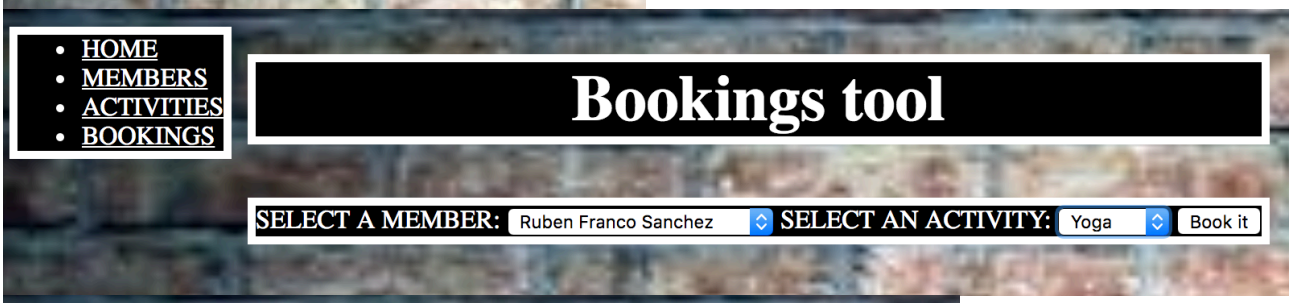
Description here

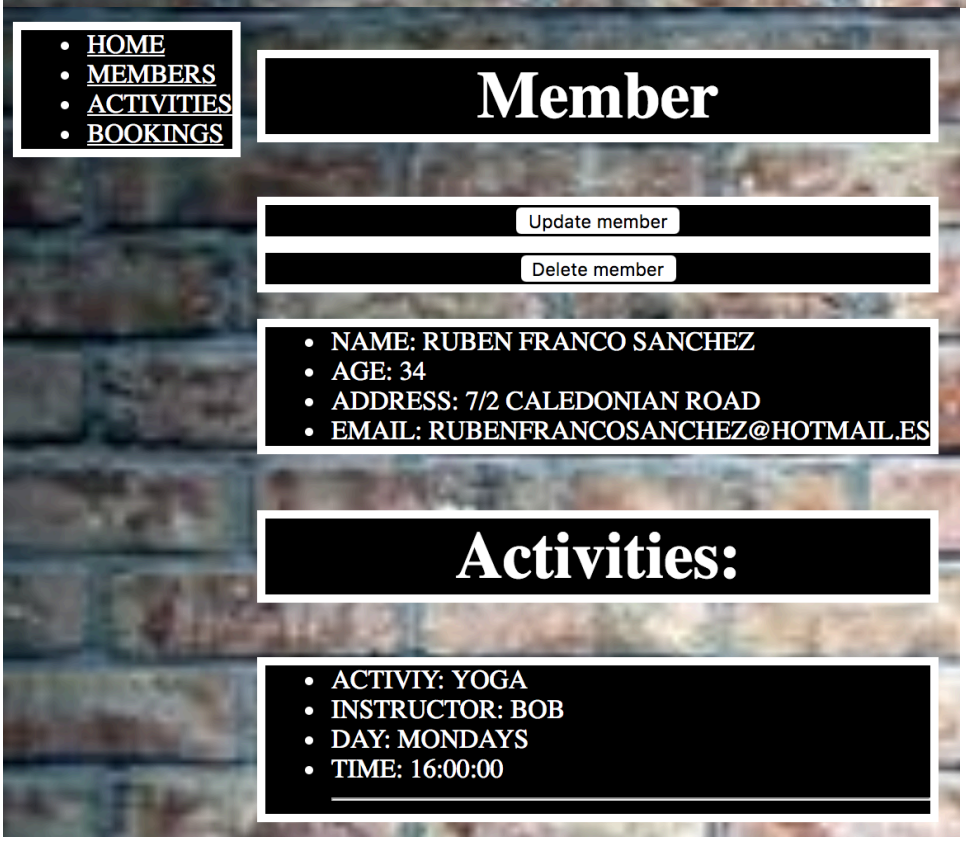
Screenshot 1: Actual information from the database.
Screenshot 2: Adding information.
Screenshot 3: Shows the information being saved.

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
		Description:

Paste Screenshot here





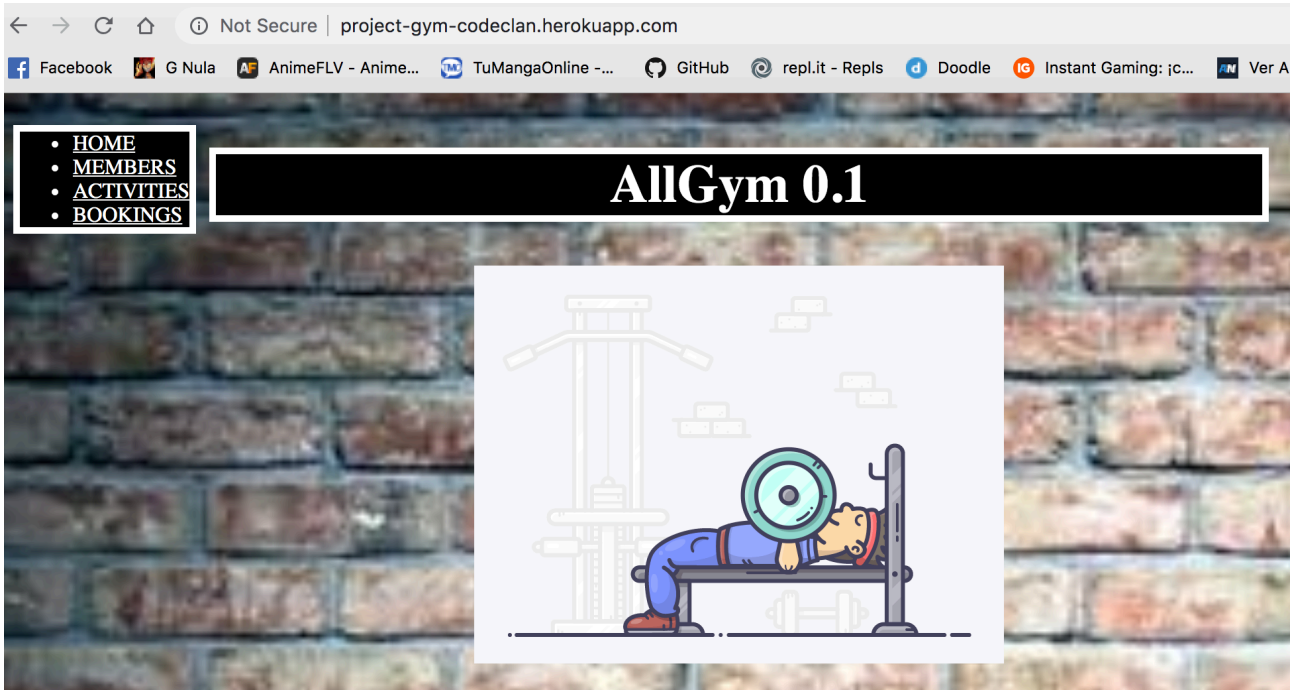


Description here

Booking an activity to a member.

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

Paste Screenshot here

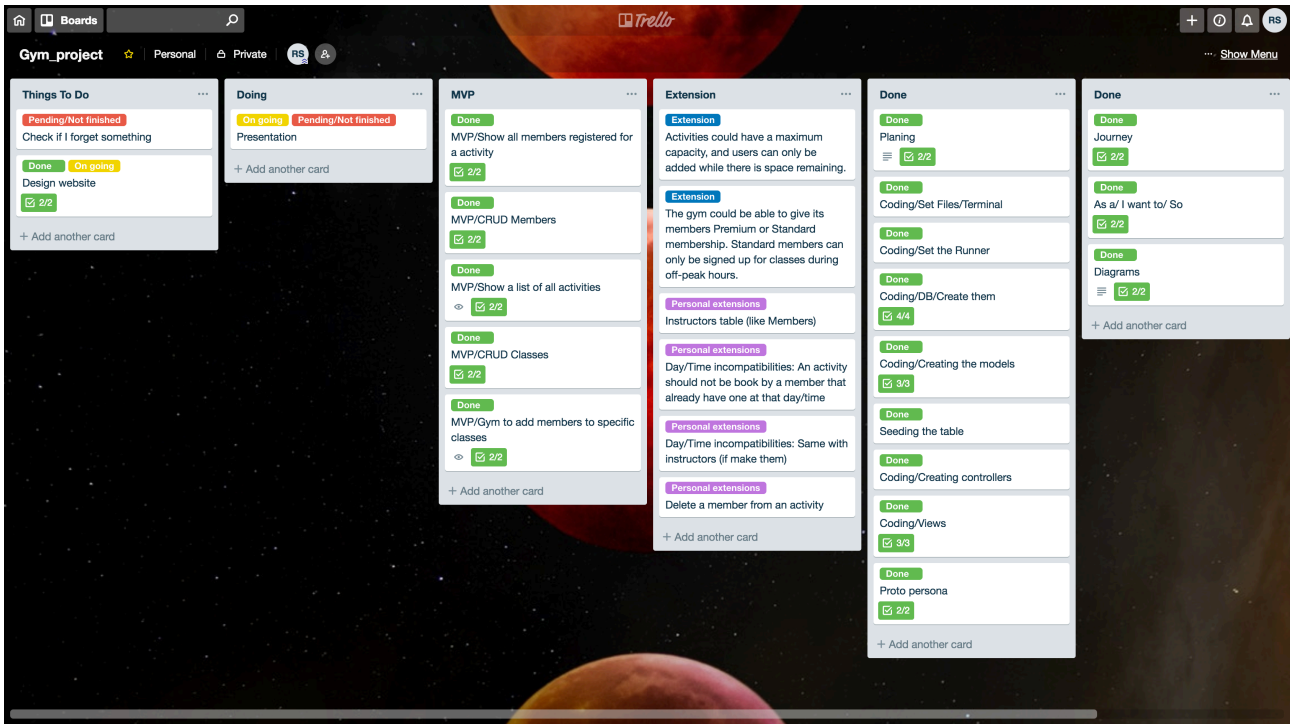


Description here

Main page from the project I work on my own.

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

Paste Screenshot here



Description here

Screenshots of the Trello showing the planning with labels showing the progress of the project.

Week 7

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
		Description:

Paste Screenshot here

```
const PubSub = require('../helpers/pub_sub.js');
const RequestHelper = require('../helpers/request_helper.js');

const Country = function () {
  this.text = null;
};

Country.prototype.getData = function () {
  const request = new RequestHelper('https://restcountries.eu/rest/v2/all');
  request.get((data) => {
    this.text = data;
    PubSub.publish('Country: Countries-Loaded', this.text);

    PubSub.subscribe('SelectView: Index Sent', (event) => {
      const selectedIndex = event.detail;
      this.publishCountryData(selectedIndex);
    });
  })

  Country.prototype.publishCountryData = function (selectedIndex) {
    const selectedCountry = this.text[selectedIndex];
    console.log(selectedCountry);
    PubSub.publish('Country:Object Ready to Send', selectedCountry);
  };
};


module.exports = Country;

```

Countries of the World

Select a country: Spain

Spain

Region:

Europe

Languages:

- Spanish

Description here

Screenshot 1: Code from the API.
Screenshot 2: API working.

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
		Description:

Paste Screenshot here

```
def test_check_for_Ace
  expected1 = true
  expected2 = false
  actual1 = checkforAce(1)
  actual2 = checkforAce(2)
  assert_equal(expected1, actual1)
  assert_equal(expected2,actual2)
end
```

```
require('minitest/autorun')
require('minitest/rg')
require_relative('../testing_task_2')
require_relative('../card')

class TestCard < Minitest::Test

  def setup
    @card1 = Card.new("Spades", 1)
    @card2 = Card.new("Hearts", 2)
    @cardGame = CardGame.new()
    @cards = [@card1, @card2]
  end

  def test_check_for_Ace()
    assert_equal(true, @cardGame.checkforAce(@card1))
    assert_equal(false, @cardGame.checkforAce(@card2))
  end
```

```
→ PDA_Static_and_Dynamic_Task_A git:(master) ✗ ruby specs/testing_task_2_specs.rb
Run options: --seed 37295

# Running:

E

Finished in 0.001152s, 868.0556 runs/s, 0.0000 assertions/s.

  1) Error:
CardGameSpec#test_check_for_Ace:
NoMethodError: undefined method `checkforAce' for #<CardGameSpec:0x007fea54875460>
    specs/testing_task_2_specs.rb:11:in `test_check_for_Ace'

1 runs, 0 assertions, 0 failures, 1 errors, 0 skips
→ PDA_Static_and_Dynamic_Task_A git:(master) ✗
```

```
require_relative('./card')
class CardGame

  def checkforAce(card)
    if card.value == 1
      return true
    else
      return false
    end
  end
```

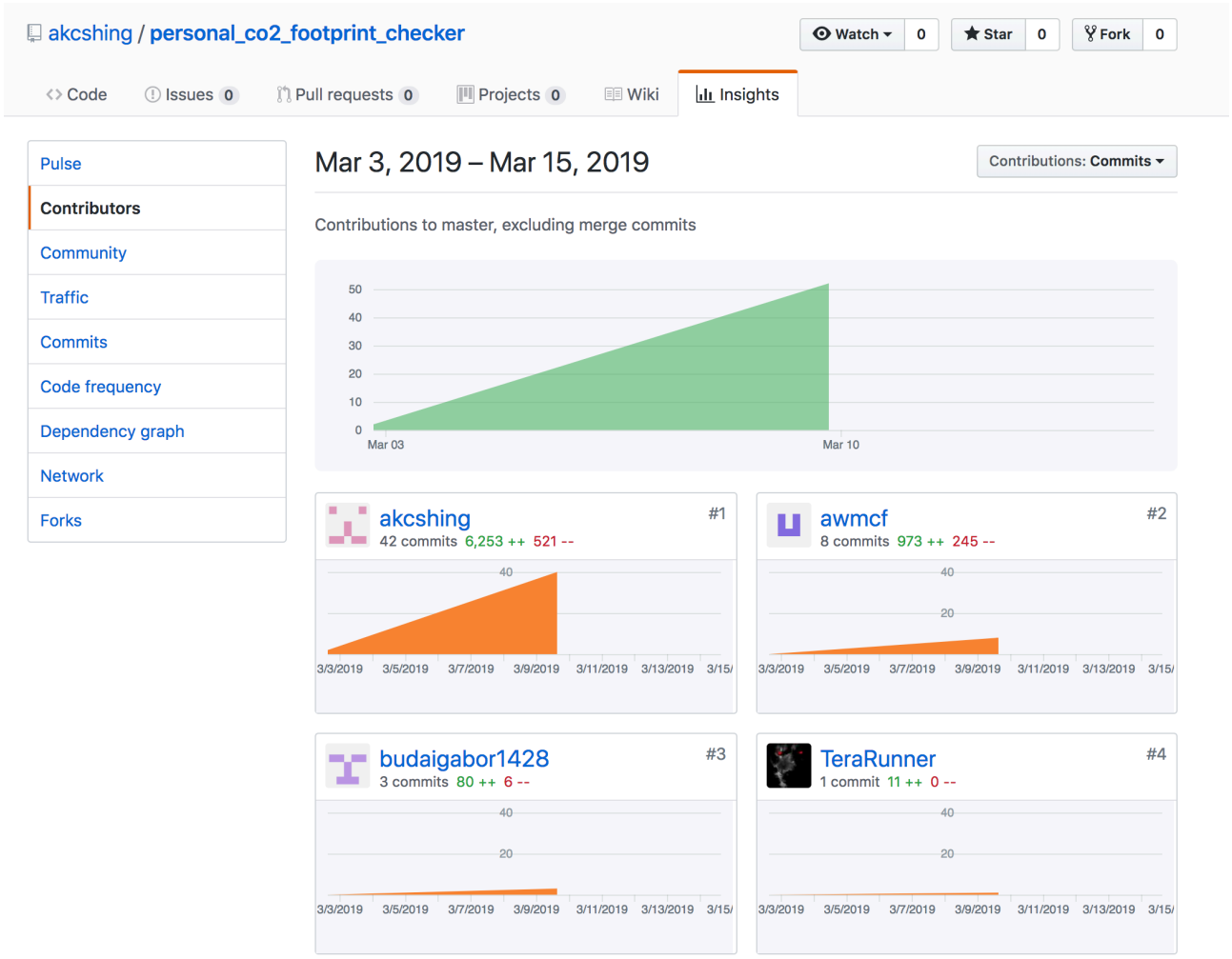
Description here

Screenshots showing test code failing (left) and corrected (right).
They also show the piece of code we use.

Week 9

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.
		Description:

Paste Screenshot here

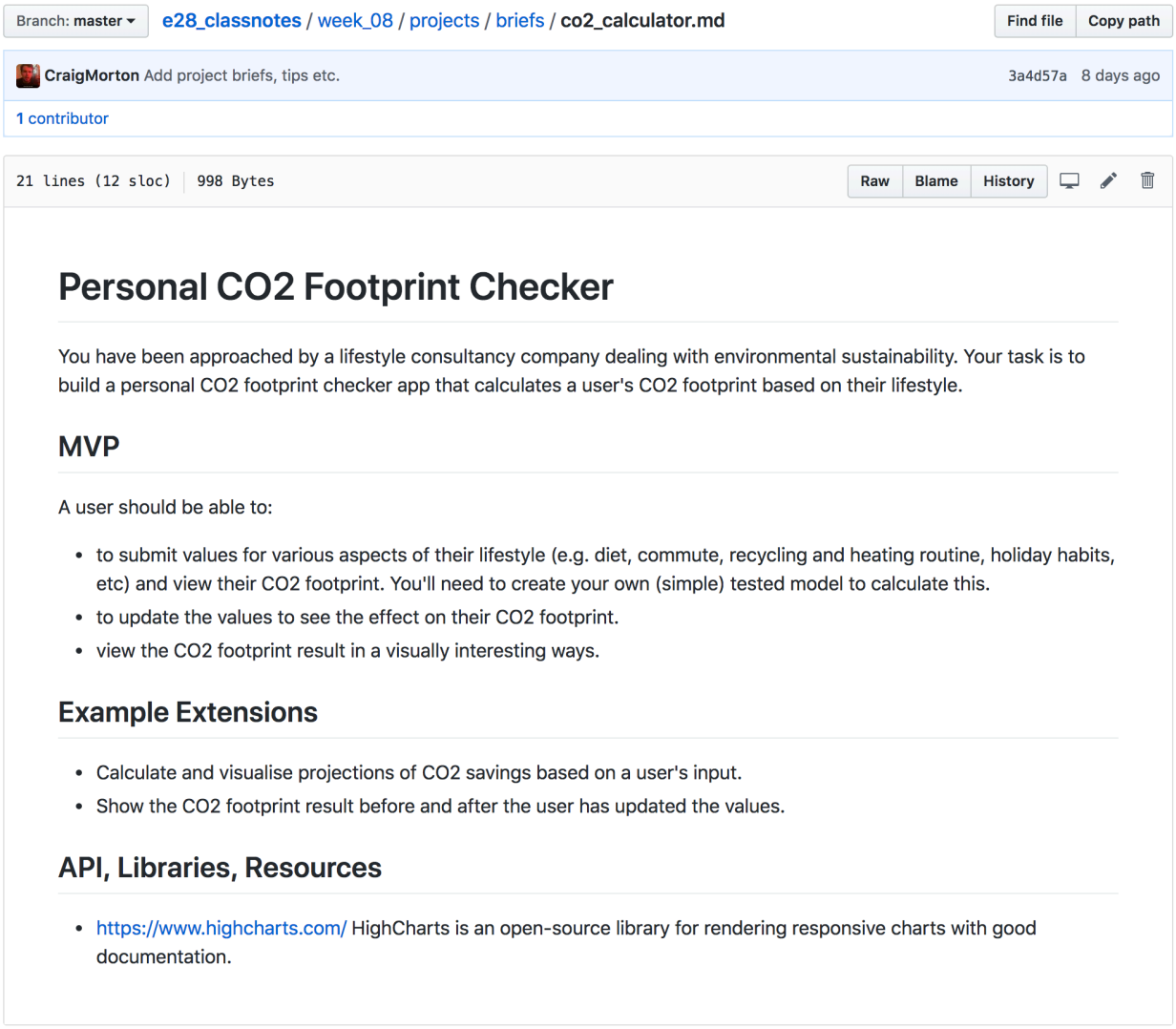


Description here

I am TeraRunner on this page.
Big part of the project we were mob programming.
Almost at the end, I pair programming with akcshing.
That is why I only have one commit.

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

Paste Screenshot here



Branch: master ▾ e28_classnotes / week_08 / projects / briefs / co2_calculator.md Find file Copy path

CraigMorton Add project briefs, tips etc. 3a4d57a 8 days ago

1 contributor

21 lines (12 sloc) 998 Bytes Raw Blame History

Personal CO2 Footprint Checker

You have been approached by a lifestyle consultancy company dealing with environmental sustainability. Your task is to build a personal CO2 footprint checker app that calculates a user's CO2 footprint based on their lifestyle.

MVP

A user should be able to:

- to submit values for various aspects of their lifestyle (e.g. diet, commute, recycling and heating routine, holiday habits, etc) and view their CO2 footprint. You'll need to create your own (simple) 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

- Calculate and visualise projections of CO2 savings based on a user's input.
- Show the CO2 footprint result before and after the user has updated the values.

API, Libraries, Resources

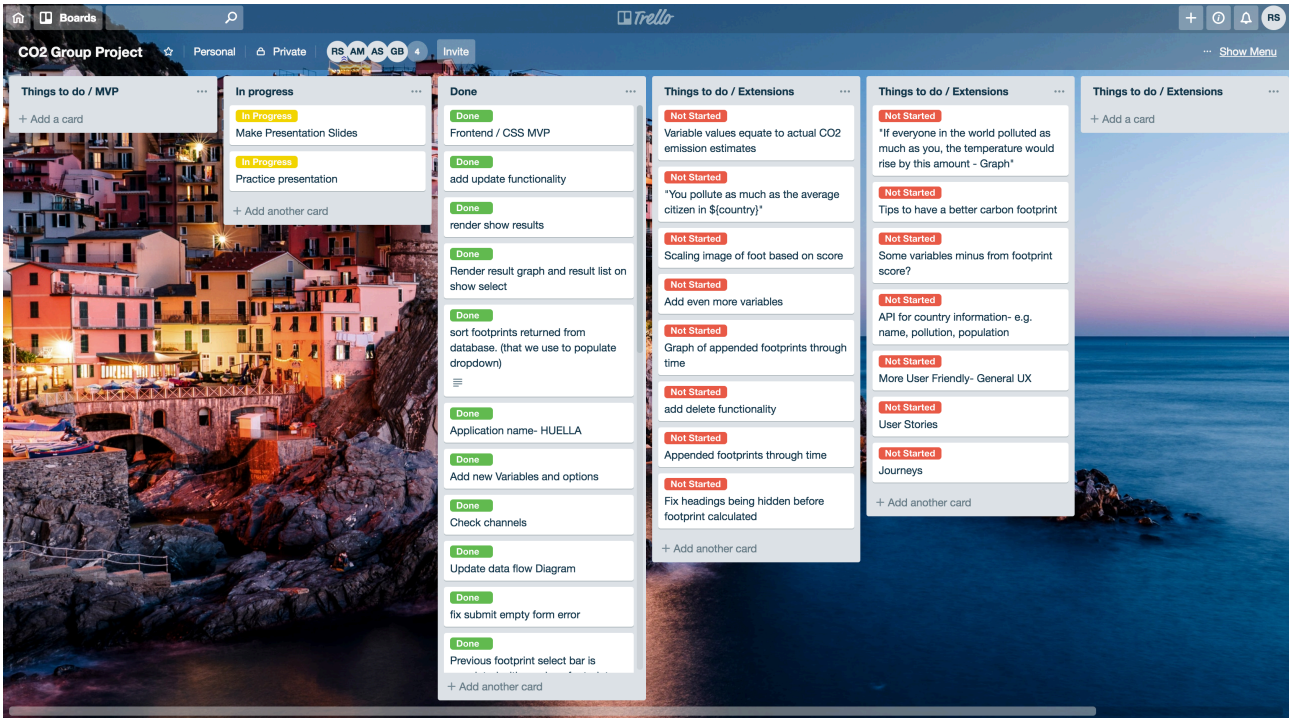
- https://www.highcharts.com/ HighCharts is an open-source library for rendering responsive charts with good documentation.

Description here

Brief from the project we choose as a group.

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

Paste Screenshot here



Description here

We used “Trello” for our planning.

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

Paste Screenshot here

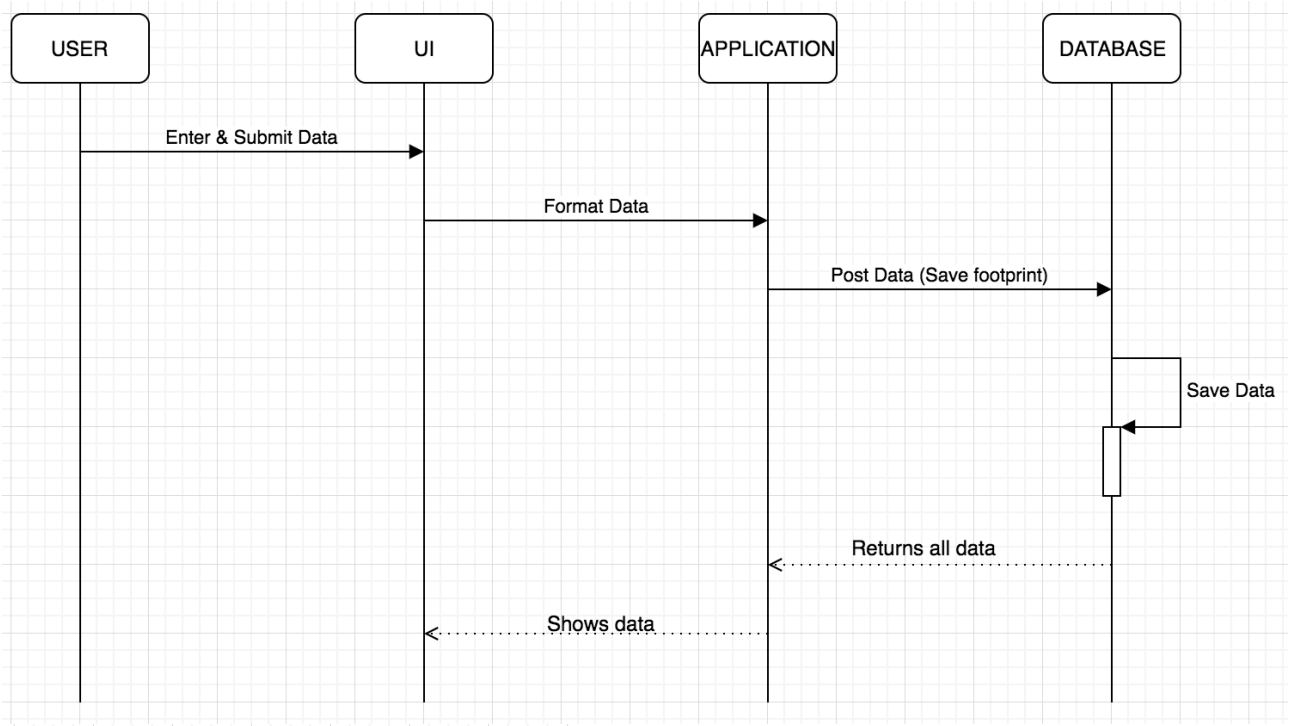
Acceptance Criteria	Expected Result	Pass/Fail
A user is able to calculate his footprint	Fill a form and receive a calculated footprint	Pass
A user is able to update/correct the information he add it	Saw the updated information on the web	Pass
A user is able to see his details in a fancy and detailed way	Receive a graph and a details trough views on the web	Pass
A user is able to see his footprint during a period of time	Receive a timeline graph with his details over time	Pass

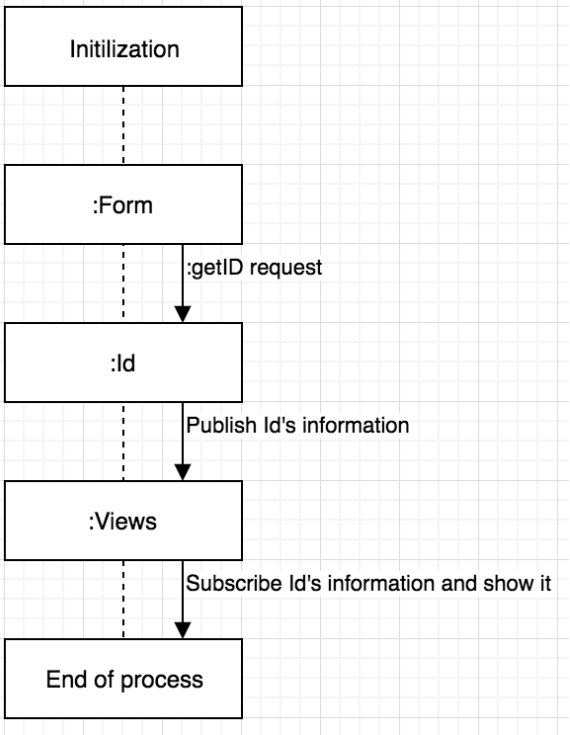
Description here

Acceptance criteria and test plan from the CO2 project.

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

Paste Screenshot here





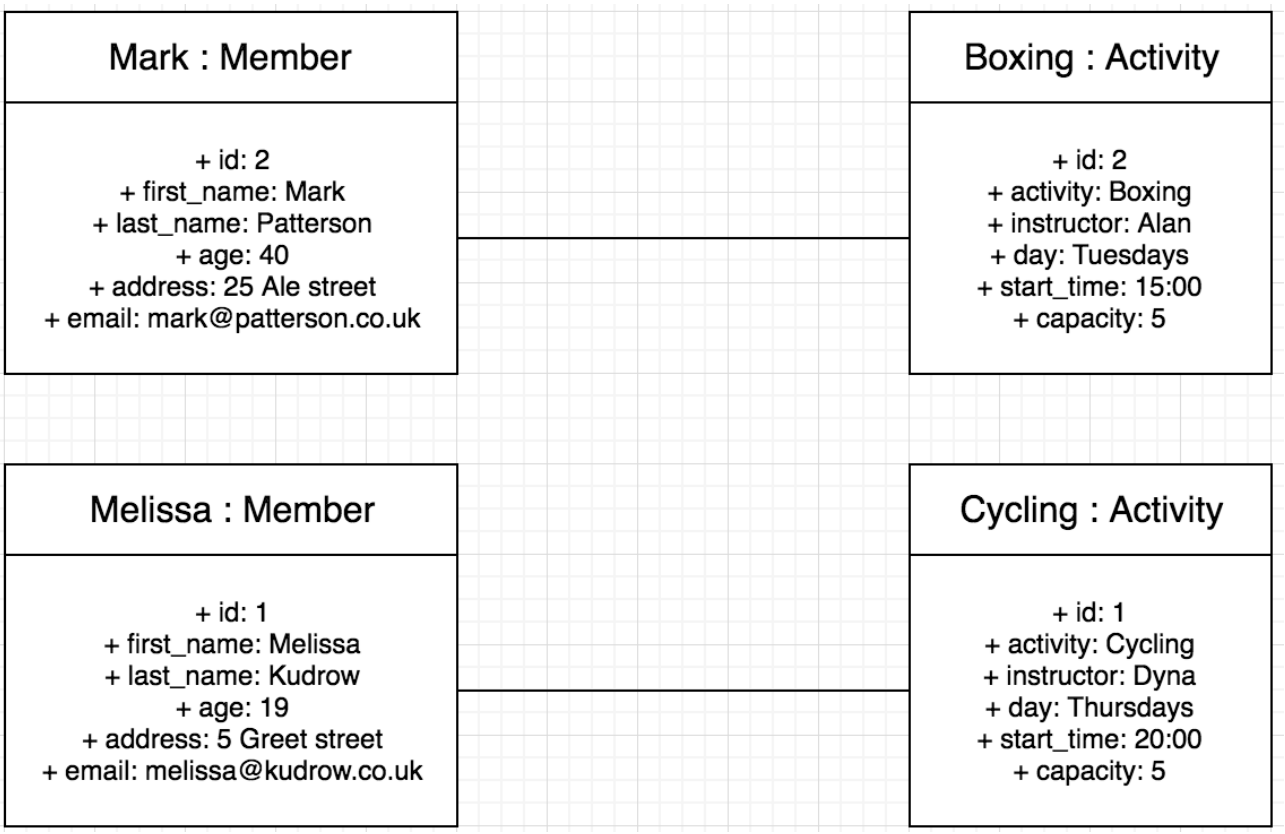
Description here

1.- Sequence Diagram when the user enter data to send it to our database.

2.- Collaboration diagram when the user select a previous date on the form and the application shows it.

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

Paste Screenshot here



Description here

Two objects diagrams showing different members that joined different activities.

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

Paste Screenshot here

Bug/Error	Solution	Date
Wrong information in the form (on select)	Refactor the bindevent in the view	12-3
Testing Mocha Error not running	Corrected the directory	12-3
Render function error appending instead of changing	Adding innerHTML: ' ' to clear first	13-3
Was not adding numbers from the form	Use of the ParseInt to transform the date to integers	13-3
Cannot render timeline chart	We have to use a different chart	14-3

Description here

Bug report from the CO2 Footprint project.

Week 12

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

Paste Screenshot here

```java
import java.util.*;

public class Network {
    private String name;
    private ArrayList<IConnect> devices;

    public Network(String name){
        this.devices = new ArrayList<IConnect>();
        this.name = name;
    }

    public String getName() { return name; }

    public int deviceCount() { return devices.size(); }

    public void connect(IConnect device) { devices.add(device); }

    public void disconnectAll() { devices.clear(); }
}
```

```java
public class Printer implements IConnect{

    public String print(String data){
        return "printing: " + data;
    }

    public String connect(String data) {
        return "connecting to " + data + " network";
    }
}
```

```java
public class Desktop implements IConnect {
    private String name;
    private String make;
    private String model;

    public Desktop(String name, String make, String model) {
        this.name = name;
        this.make = make;
        this.model = model;
    }

    public String getName() { return name; }

    public String getMake() { return make; }

    public String getModel() { return model; }

    public String connect(String data) { return "connecting to network: " + data; }
}
```

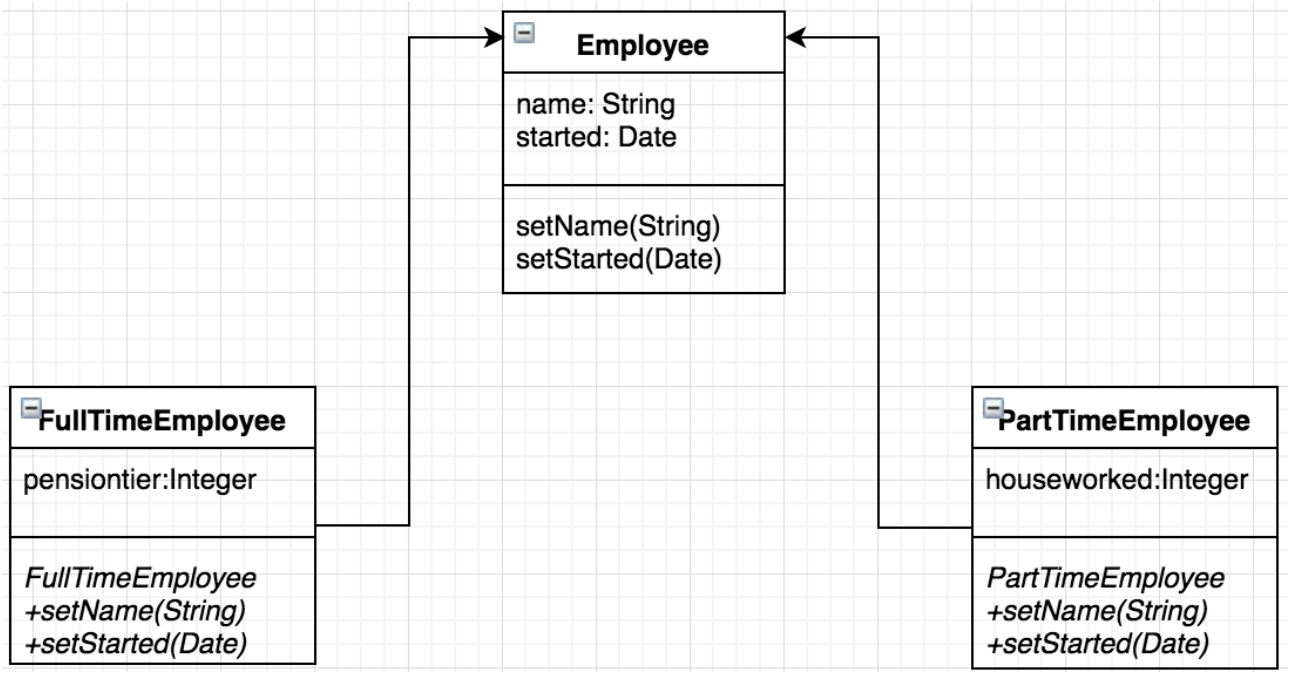
```java
public interface IConnect {
    public String connect(String data);
}
```

Description here

Use of polymorphism to connect several devices trough a network.

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

Paste Screenshot here

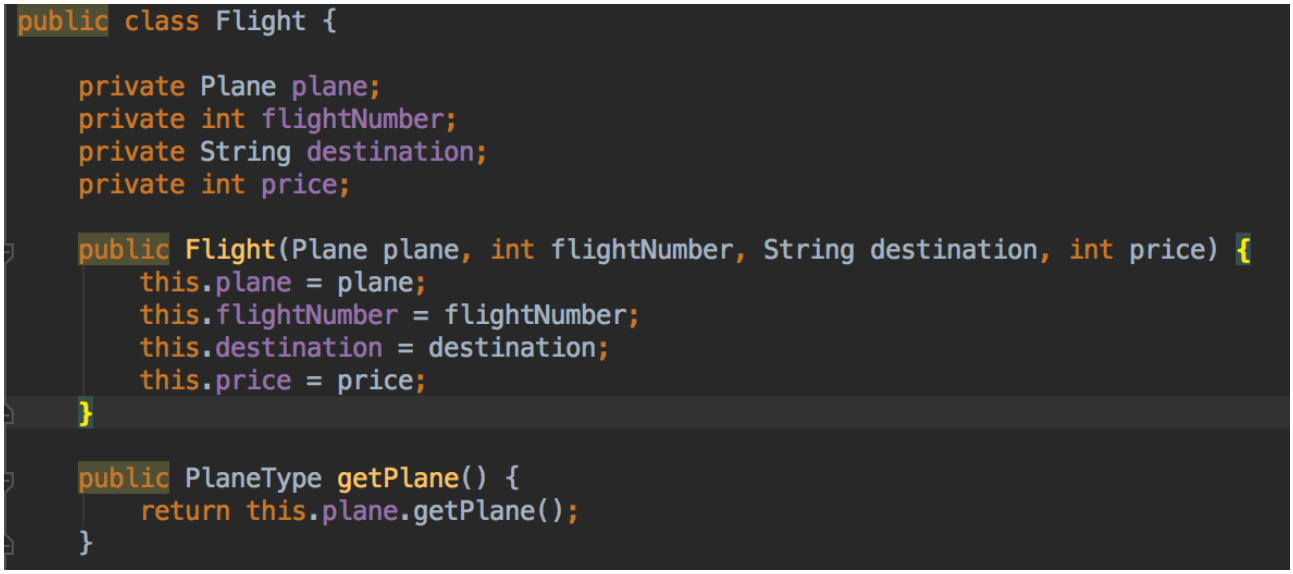


Description here

Inheritance diagram from an employee and the kinds of employee.

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

Paste Screenshot here



```
public class Flight {

    private Plane plane;
    private int flightNumber;
    private String destination;
    private int price;

    public Flight(Plane plane, int flightNumber, String destination, int price) {
        this.plane = plane;
        this.flightNumber = flightNumber;
        this.destination = destination;
        this.price = price;
    }

    public PlaneType getPlane() {
        return this.plane.getPlane();
    }
```

Description here

Encapsulation of flight. Declaring it's values as privates and calling them by public 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.
		Description:

Paste Screenshot here

Person.java

```java
public class Person {

    private String name;
    private String cohort;

    public Person(String name, String cohort) {
        this.name = name;
        this.cohort = cohort;
    }

    public String getName() { return this.name; }

    public String getCohort() { return this.cohort; }

    public void setName(String name) { this. name = name; }

    public void setCohort(String cohort) { this.cohort = cohort; }

    public String talk(String language) { return "I love " + language; }

}
```

Person › talk()

Student.java

```java
public class Student extends Person {

    public Student(String name, String cohort) { super(name, cohort); }

    public String talk(String language) { return "I love learning " + language; }

}
```

Description here

Inheritance from Person class and the student.

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.
		Description:

Paste Screenshot here

```
public int drinkPotion(HealingTool healingTool) {
    int lifeIncrement = 0;
    int healingToolDecrease = 0;
    while (this.getHp() < 20 && healingTool.getHealingToolDamage() > 0) {
        lifeIncrement = this.getHp() + 1;
        this.setHp(lifeIncrement);
        healingToolDecrease = healingTool.getHealingToolDamage() - 1;
        healingTool.setHealingToolDamage(healingToolDecrease);
    }
    return this.getHp();
}
```

```
public int damageDeal(Character enemyCharacter) {
    if (enemyCharacter.getArmour().getArmourDefense() >= weapon.getWeaponDamage()) {
        return enemyCharacter.hp -= 1;
    } else {
        int damage = weapon.getWeaponDamage() - enemyCharacter.getArmour().getArmourDefense();
        return enemyCharacter.hp -= damage;
    }
}
```

Description here

The first algorithm takes an attribute from the class and, meanwhile it is not reaching the top “lifepoints” and there is still enough “potion”, it will increase the hp (healing points). It will return the healed hp at the end.

The second algorithm will return the hp attribute from the passed class after it checks the armour from the objective class less the weapon damage.

I choose these two because I am proud of them. They teach me that I was wrong and they can be improved. They show me that this is a constant learning.