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.

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 1

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

Paste Screenshot here

```ruby
chickens = [
  {name:'Margaret', age:2, eggs: 0},
  {name:'Hetty', age:5, eggs: 7},
  {name:'Audrey', age:4, eggs: 8},
]

total_eggs = 0
for chicken in chickens
# p "#{ chicken[:name]} is #{chicken[:age]}"
  total_eggs += chicken[:eggs]
  p 'wooo eggs' if chicken[:eggs] > 0
  chicken[:eggs] = 0
end
p total_eggs
```

```
→  day_3 ruby loops.rb
"wooo eggs"
"wooo eggs"
15
```

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

An array in a program

```
fruits = ['apple', 'banana', 'grape', 'orange']
p fruits
```

Function that uses the array

```
fruits = fruits[2] + fruits[1] + fruits[0]
p fruits
```

A result of a function running

```
["apple", "banana", "grape", "orange"]
["kiwi", "apple", "banana", "grape", "orange", "lemon"]
"bananaapplekiwi"
```

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

Paste Screenshot here

```
def artist
  sql = "SELECT * FROM artists
  WHERE id = $1"
  values = [@artist_id]
  results = SqlRunner.run(sql, values)
  artist_data = results[0]
  artist = Artist.new(artist_data)
  return artist
end
```

```
=> [#<Artist:0x007f9c40a9d858 @id=1, @name="Tool">,
 #<Artist:0x007f9c40a9d7b8 @id=2, @name="Queen">,
 #<Artist:0x007f9c40a9d718 @id=3, @name="Pavarotti">]
```

On the left you can see an example of a sql query searching for all artist in the database and above you can see the result of this query.

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
def self.all
  sql= "SELECT * FROM customers ORDER BY funds"
  customer_data = SqlRunner.run(sql)
  return Customer.map_items(customer_data)
end
```

```
1 SELECT* FROM Customers
2 ORDER BY funds
```

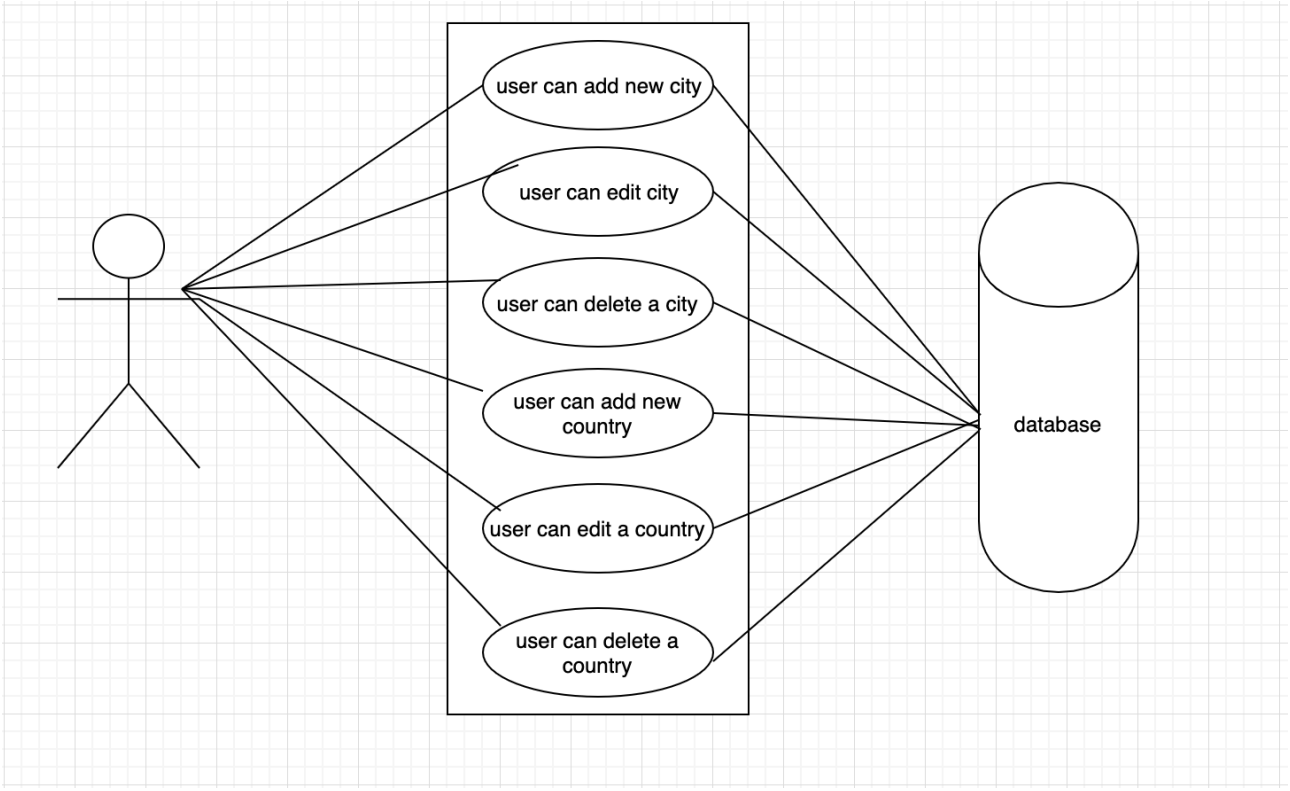
Load Query... Save Query...

d	name	funds
5	Gemma	25
1	Mark	30
4	Monica	30
2	John	45
3	Nicky	50

Week 4

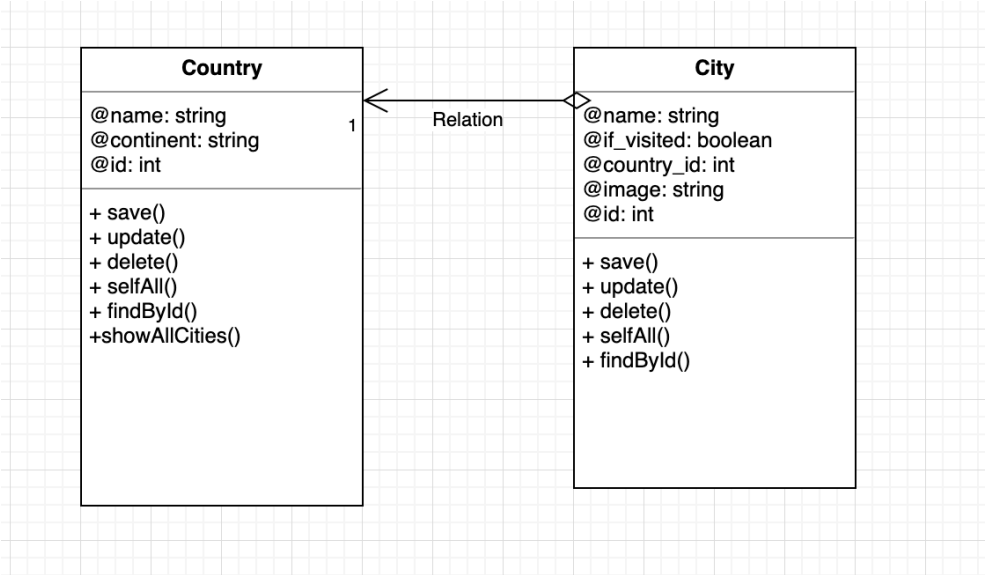
Unit	Ref	Evidence
A&D	A.D.1	A Use Case Diagram

Paste Screenshot here



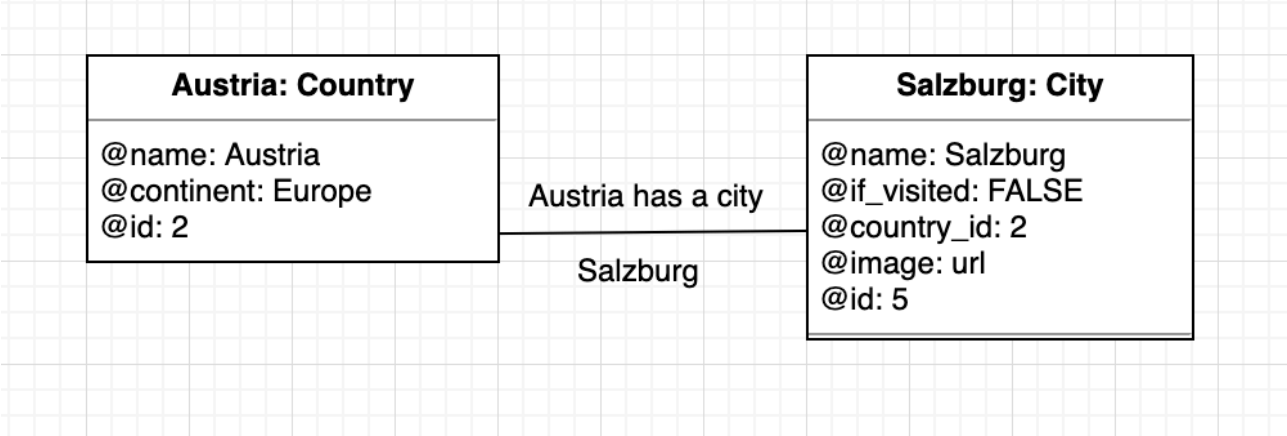
Unit	Ref	Evidence
A&D	A.D.2	A Class Diagram

Paste Screenshot here



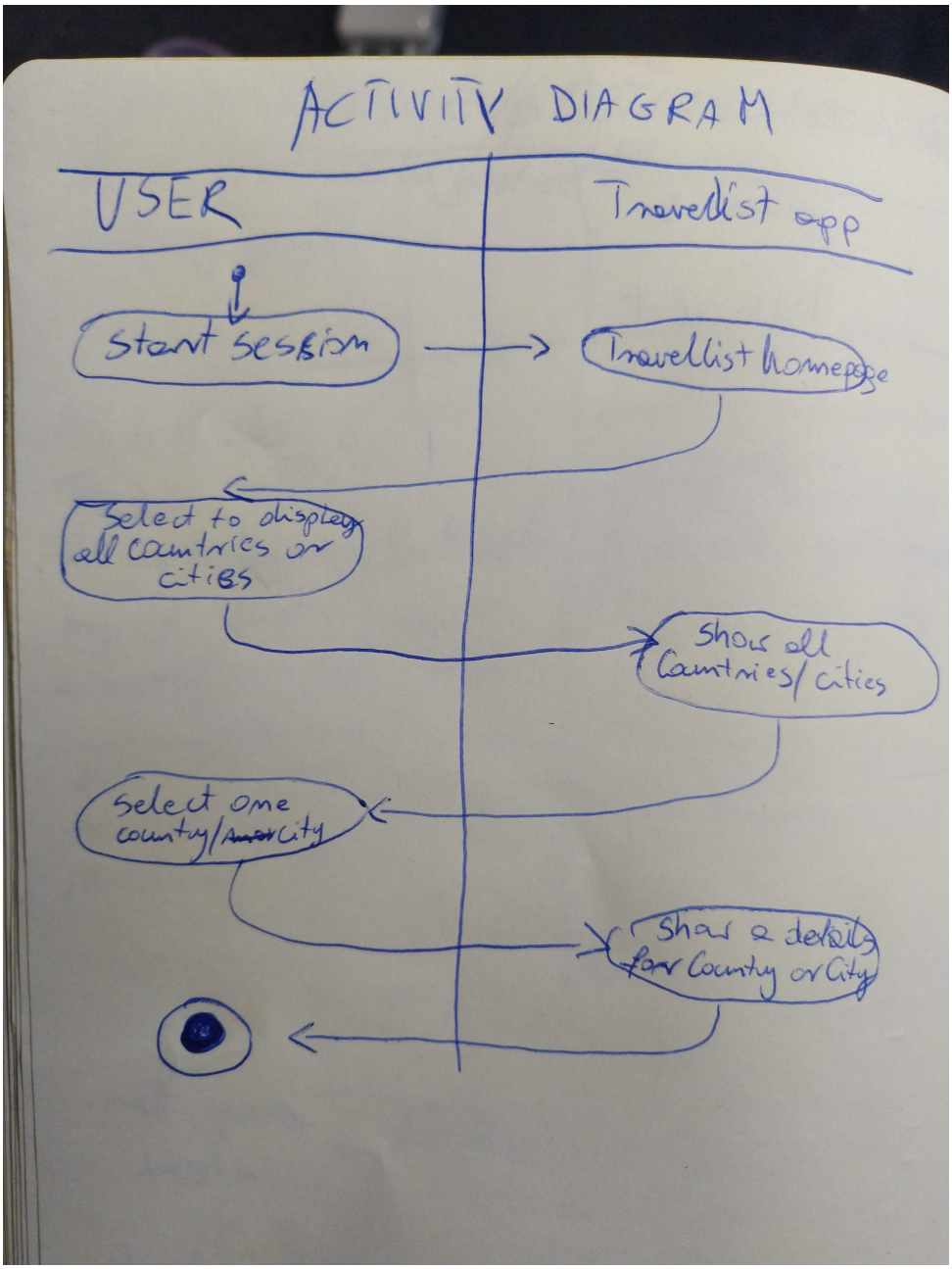
Unit	Ref	Evidence
A&D	A.D.3	An Object Diagram

Paste Screenshot here



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

Paste Screenshot here



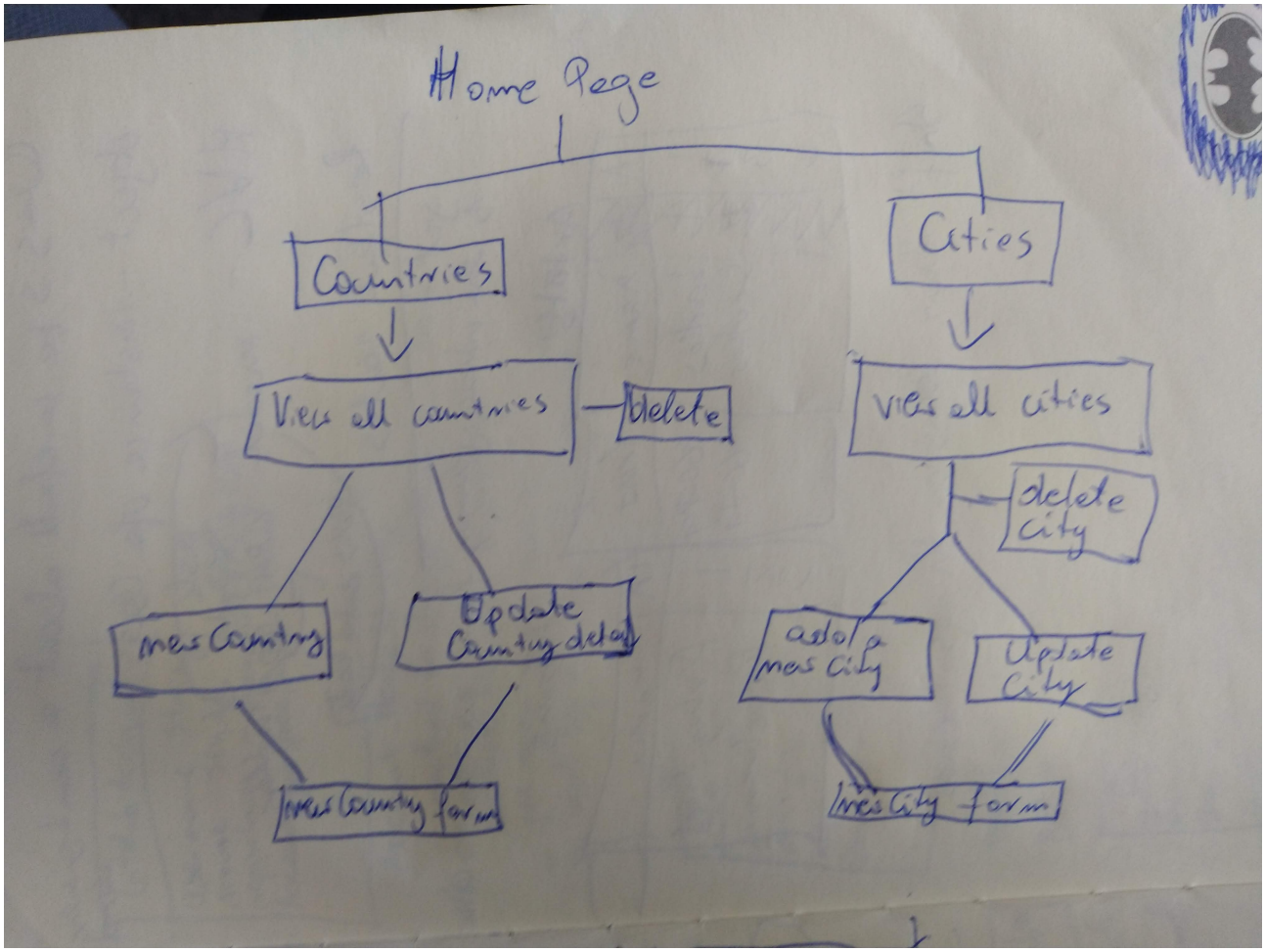
Unit	Ref	Evidence
A&D	A.D.6	Produce an Implementations Constraints plan detailing the following factors: *Hardware and software platforms *Performance requirements *Persistent storage and transactions *Usability *Budgets *Time

Paste Screenshot here

Constraint Category	Implementation Constraint	Solution
Hardware and software platforms	Built and tested on Apple laptop. Requires Ruby and Sinatra to work.	Make sure that the required technology is installed and up to date.
Performance requirements	Requires Wi-fi connection.	
Persistent storage and transactions	Requires PSQL and a database to store data.	Create appropriately named tables in POSTgres database for different classes.
Usability	At the moment, the Ruby file needs to run in order for the application to work.	Enable the application to operate entirely in the cloud.
Budgets	N/A	Personal project completed as part of a web development course.
Time	6 days until project's deadline.	Finish MVP and break workload into smaller features to be completed one at the time.

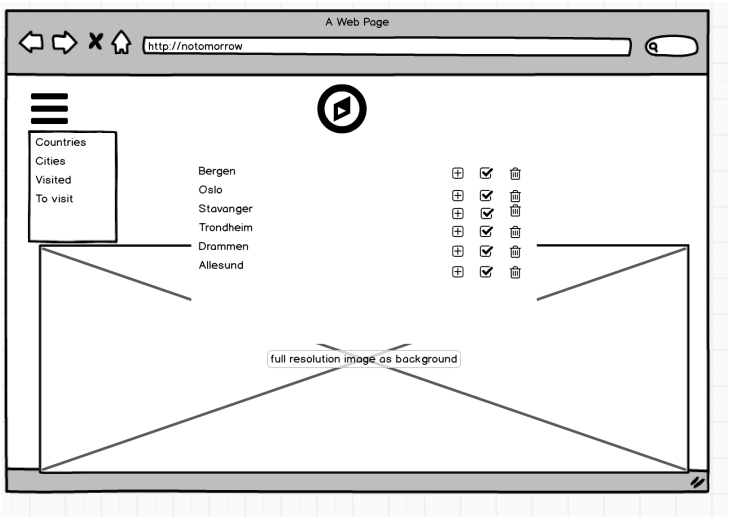
Unit	Ref	Evidence
P	P.5	User Site Map

Paste Screenshot here



Unit	Ref	Evidence
P	P.6	2 Wireframe Diagrams

Paste Screenshot here



Week 5

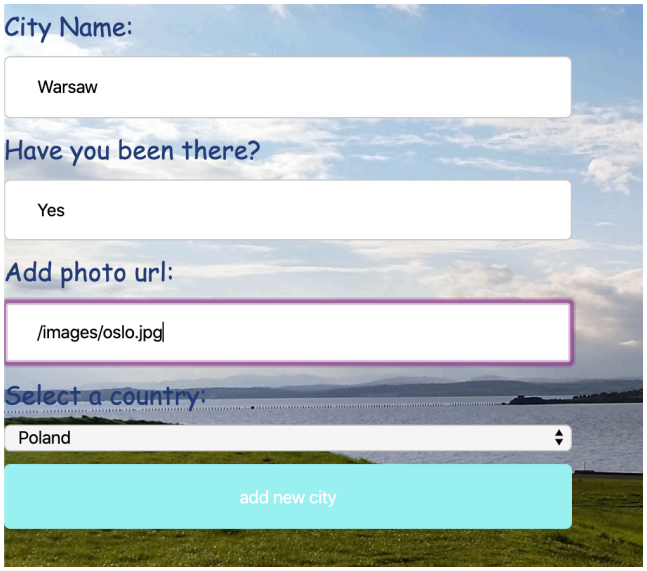
Unit	Ref	Evidence
P	P.10	Example of Pseudocode used for a method

Paste Screenshot here

```
#send a query to a database in order to show all albums and store in an array.
def Album.all
  sql = "SELECT * FROM albums"
  albums = SqlRunner.run(sql)
  return albums.map { |album| Album.new(album) }
end

#delete all albums from database with this query
def Album.delete_all
  sql = "DELETE FROM albums"
  SqlRunner.run(sql)
end
```

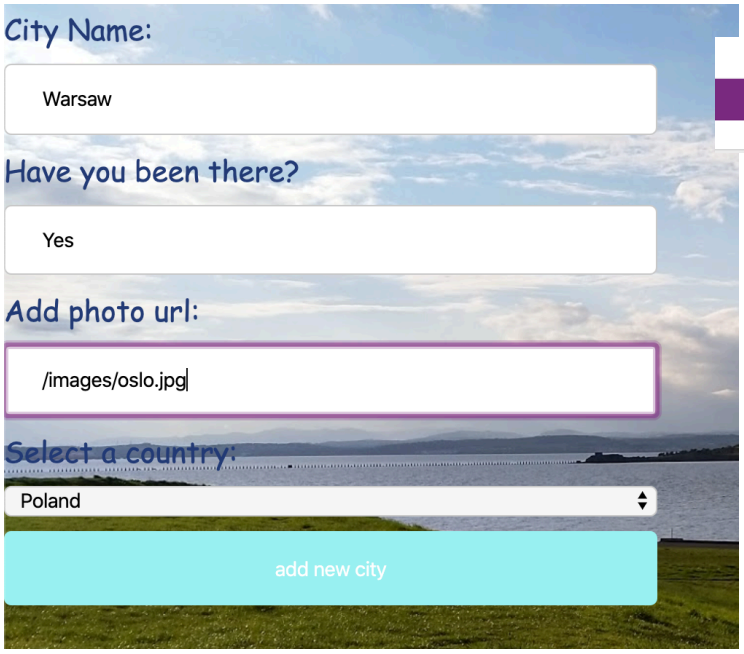
Unit	Ref	Evidence
P	P.13	Show user input being processed according to design requirements. Take a screenshot of: * The user inputting something into your program * The user input being saved or used in some way

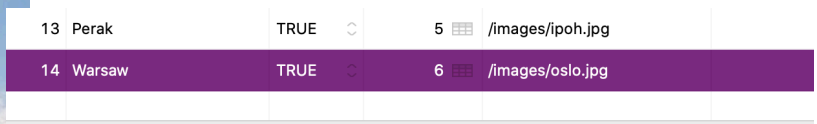
Paste Screenshot here



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

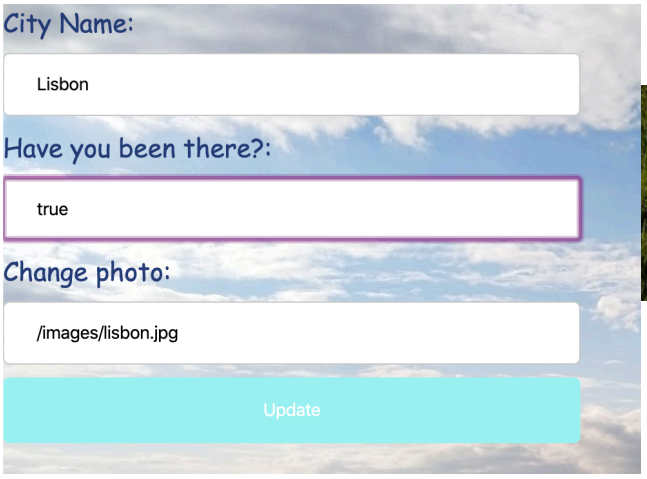
Paste Screenshot here

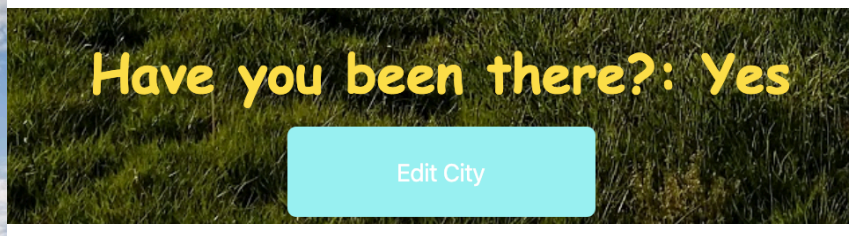




Unit	Ref	Evidence
P	P.15	Show the correct output of results and feedback to user. Take a screenshot of: * The user requesting information or an action to be performed * The user request being processed correctly and demonstrated in the program

Paste Screenshot here



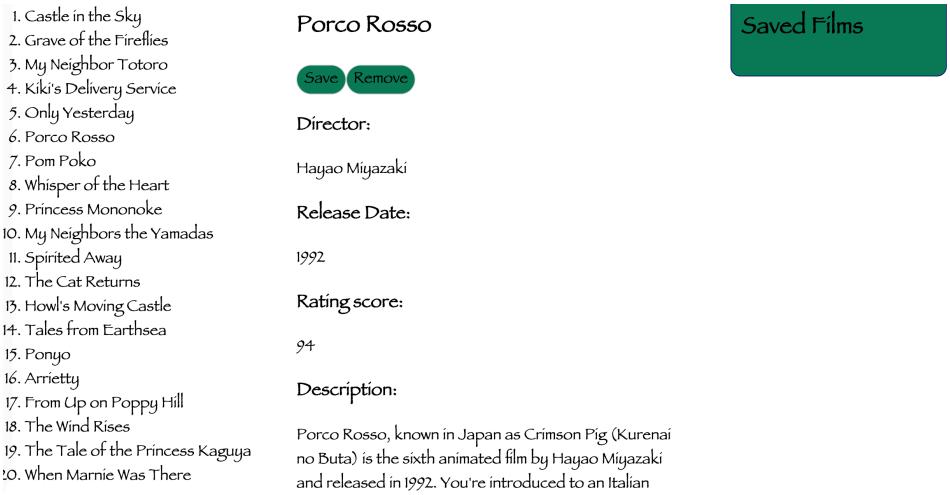


User request an update on a City, wants to change if visited property from false to true and it is displayed as yes or no..On the right side you can see change value.

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

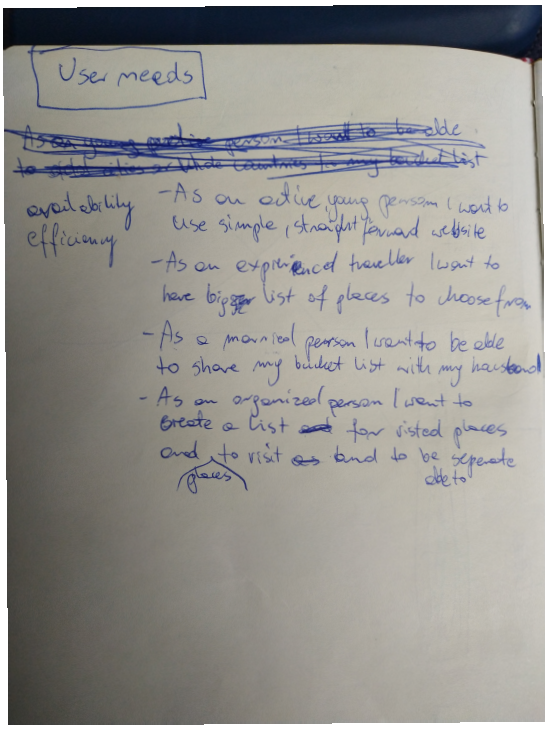
Paste Screenshot here

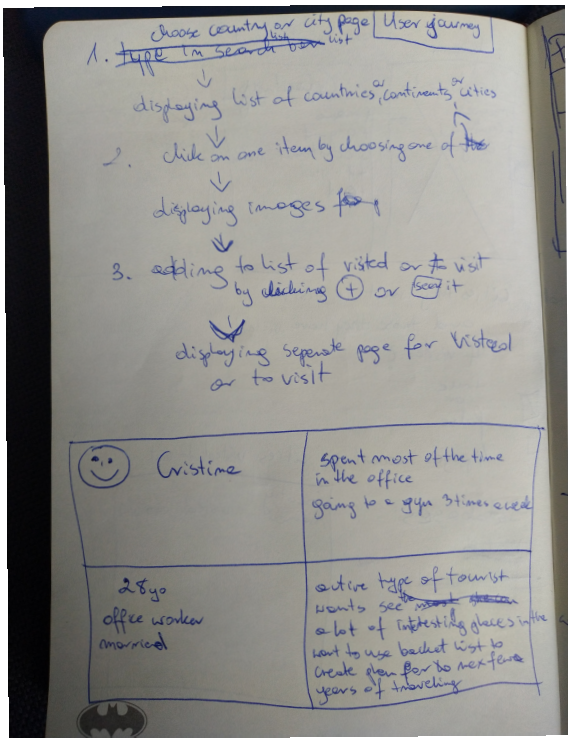
https://github.com/Bruss87/vue-app-hw

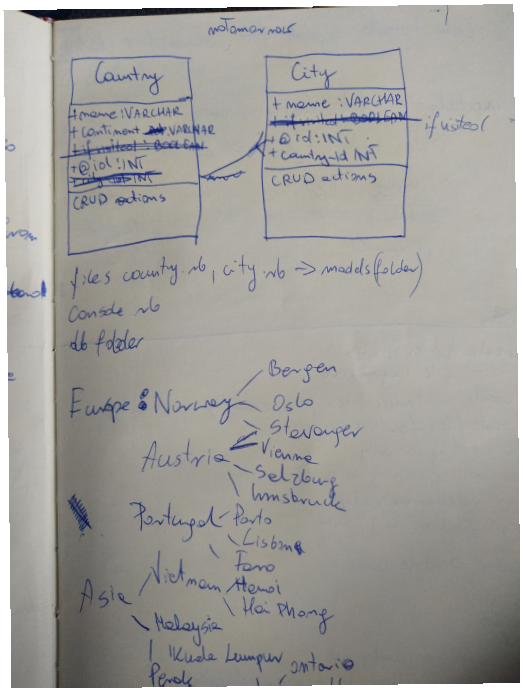


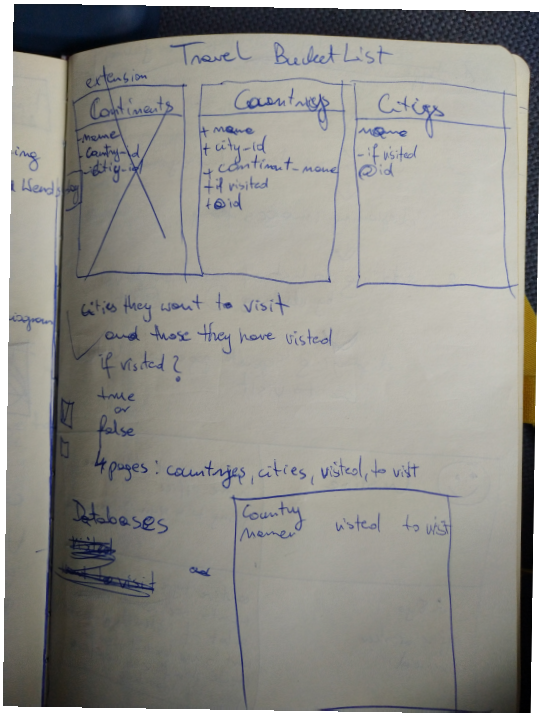
Unit	Ref	Evidence
P	P.12	Take screenshots or photos of your planning and the different stages of development to show changes.

Paste Screenshot here









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

Paste Screenshot here

```
mounted(){
fetch('https://ghibliapi.herokuapp.com/films?items=50')
  .then(res => res.json())
  .then(films => this.films = films)

  eventBus.$on('selected-film', film => this.selectedFilm = film)

  eventBus.$on('saved-film', savedFilm => this.savedFilms.push(savedFilm))

  eventBus.$on("removed-film", film => this.savedFilms.splice(-1, 1))
}
```

Week 8

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

Educational App

The BBC are looking to improve their online offering of educational content by developing some interactive browser applications that display information in a fun and interesting way. Your task is to make an a Minimum Viable Product or prototype to put forward to them - this may only be for a small set of information, and may only showcase some of the features to be included in the final app.

MVP

A user should be able to:

- view some educational content on a particular topic
- be able to interact with the page to move through different sections of content

Example Extensions

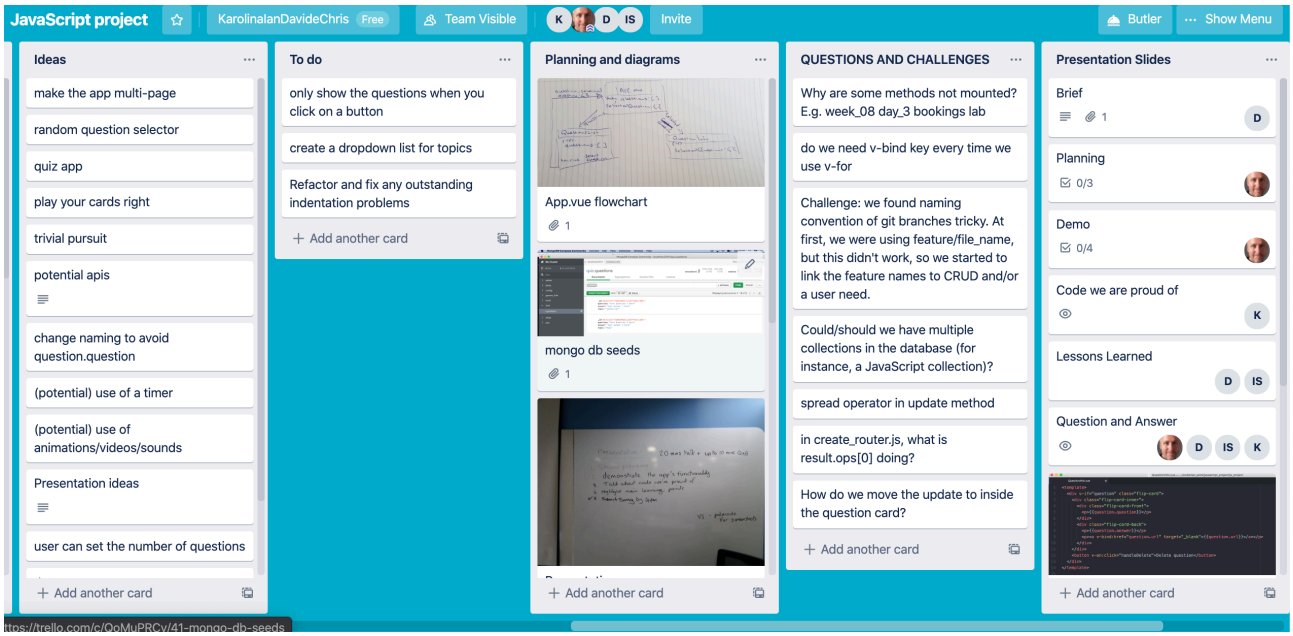
- Use an API to bring in content or a database to store information.
- Use charts or maps to display your information to the page.

API, Libraries, Resources

- https://www.highcharts.com/ HighCharts is an open-source library for rendering responsive charts.
- https://korigan.github.io/Vue2Leaflet/#/ Leaflet is an open-source library for rendering maps and map functionality.

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

Paste Screenshot here



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

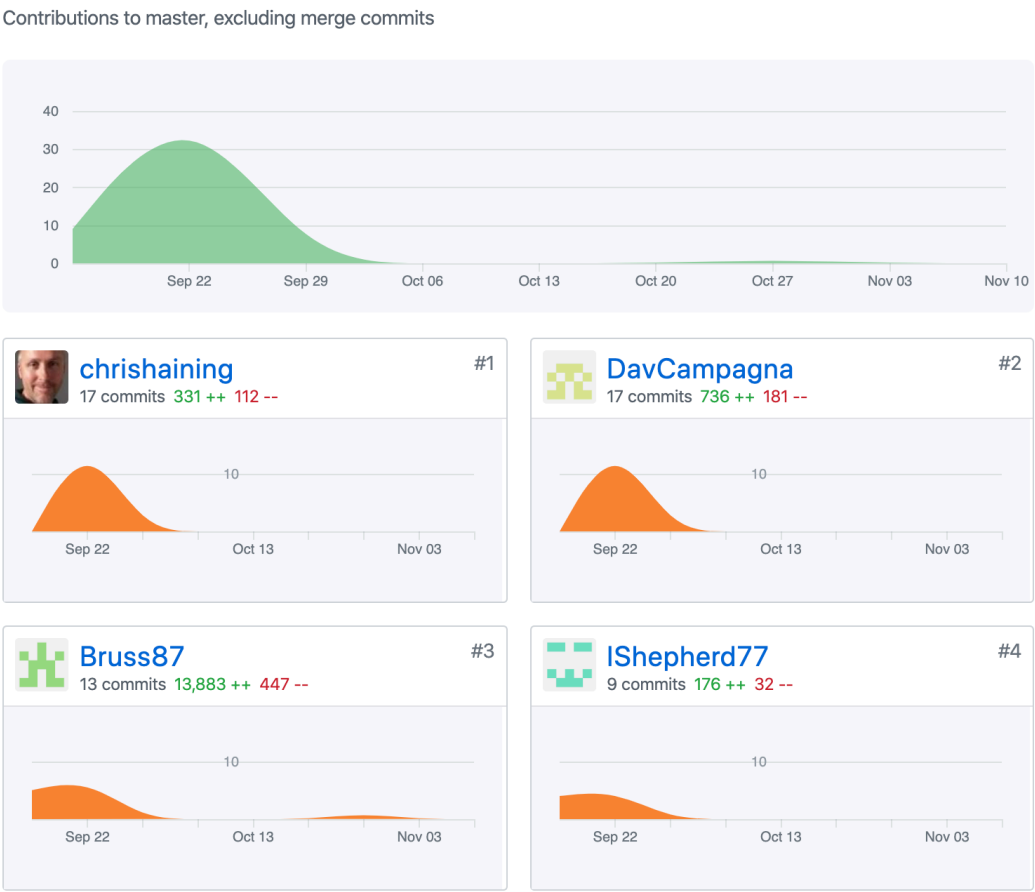
Paste Screenshot here

Acceptance Criteria	Expected Result	Pass/Fail
A user is able to	The … does … when	Pass/Fail
A user is able to clicks on two images that match	The image result returns true and the cards images remain visible	Pass
A user clicks on two images that don't match	The images return false and return to being covered	Pass
A user matches all the images in the grid	A message is sent to the screen	Pass

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.

Paste Screenshot here

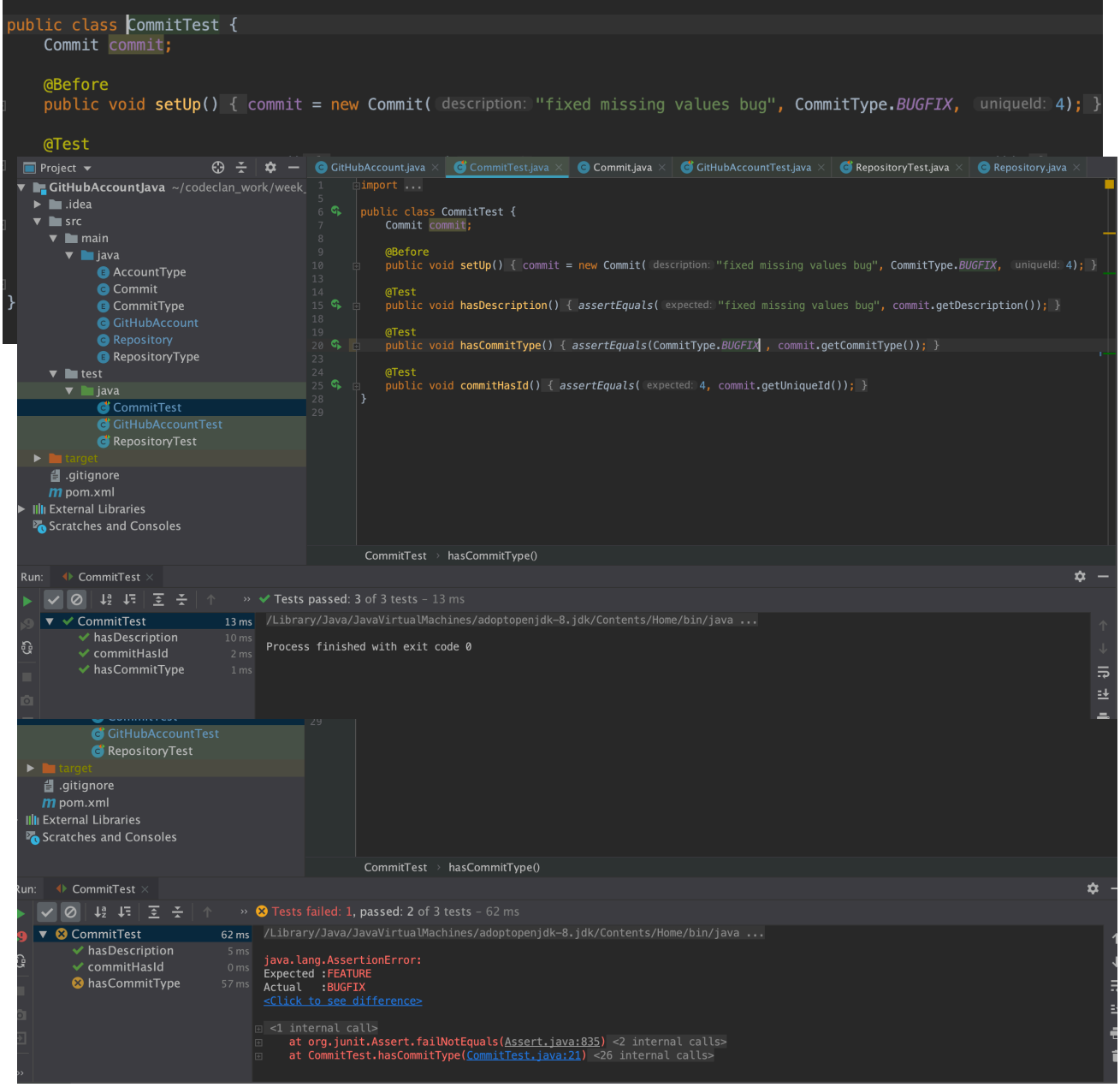


Week11

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

Description here



On those 3 screenshots above you can see test code failing to pass when commitType does not match the one in the setUp and test code that passes when commit Type match setUp.

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

Paste Screenshot here
Description here

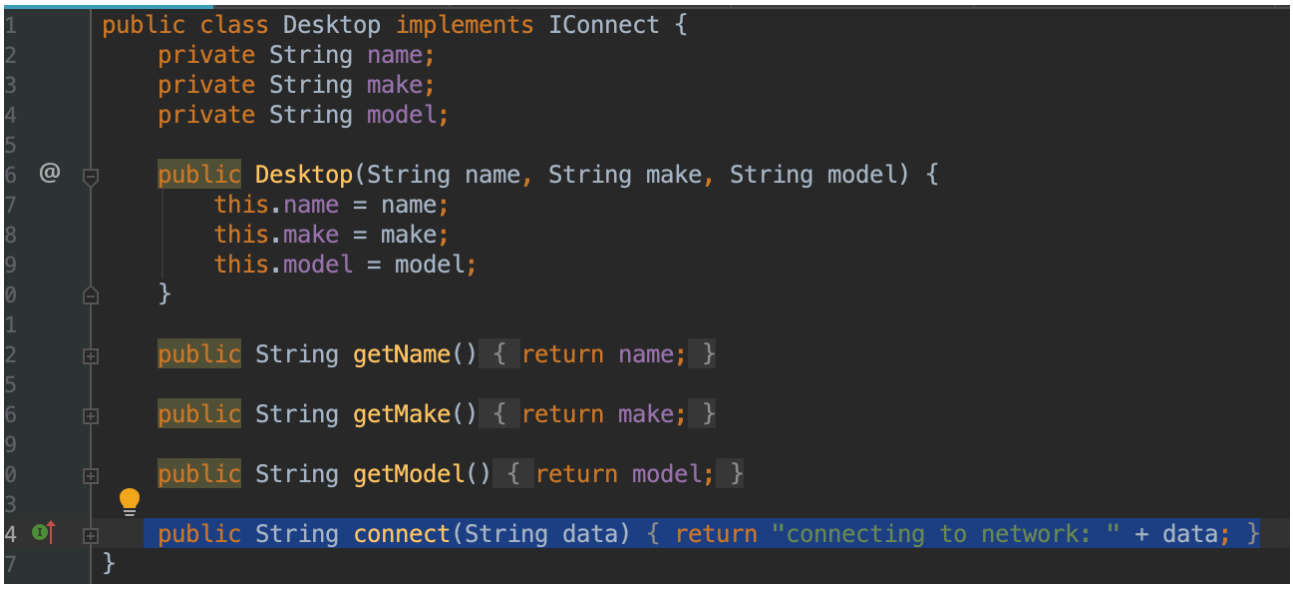
```java
public class GitHubAccount {

    private String userName;
    private String name;
    private AccountType accountType;
    private HashMap<String, Repository> repositories;

    public GitHubAccount(String userName, String name, AccountType accountType) {
        this.userName = userName;
        this.name = name;
        this.accountType = accountType.FREE;
        this.repositories = new HashMap<>();
    }

    public String getUserName() {

        return this.userName;
    }

    public String getName() {

        return this.name;
    }

    public AccountType getAccountType() { return this.accountType; }

    public void setAccountType(AccountType accountType){
        this.accountType = accountType;
    }


    public int countCollectionOfRepos() { return this.repositories.size(); }

    public void addRepoToCollectionOfRepos(Repository repository) {
        this.repositories.put(repository.getName(), repository );
    }

    public Repository findRepositoryByName(String name) { return this.repositories.get(name); }
//
//    public Repository findRepositoryByNoOfCommits(){
```

On the screenshot you can see an example of encapsulation of repository class within GitHubAccount class

Week 12

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

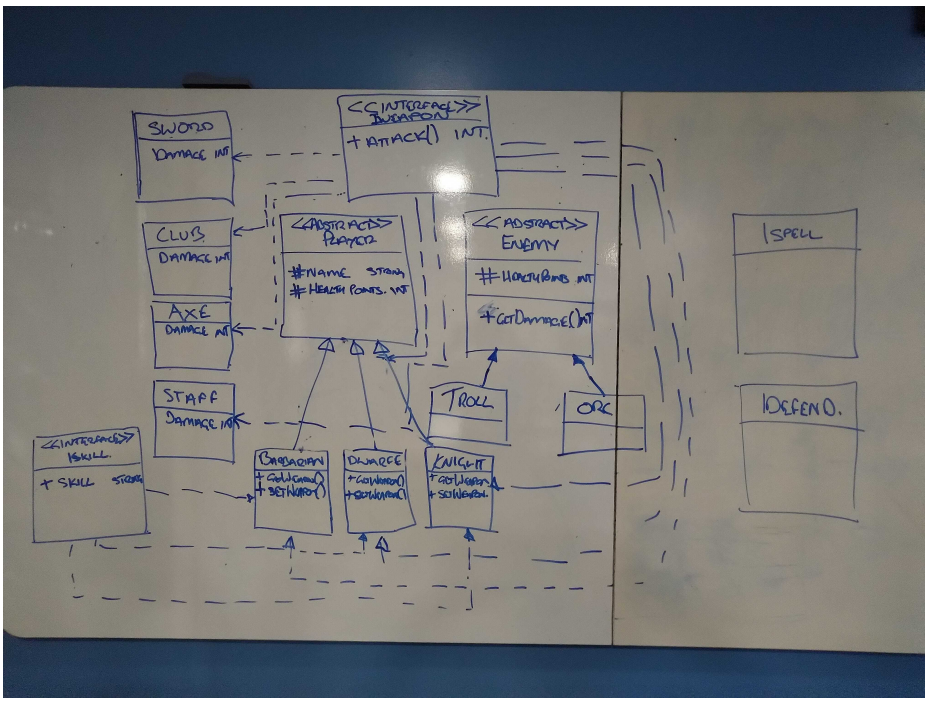
Paste Screenshot here



In the screenshot above you can see an example of Desktop class that implements an interface IConnect in order to use connect method to print a string.

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

Paste Screenshot here

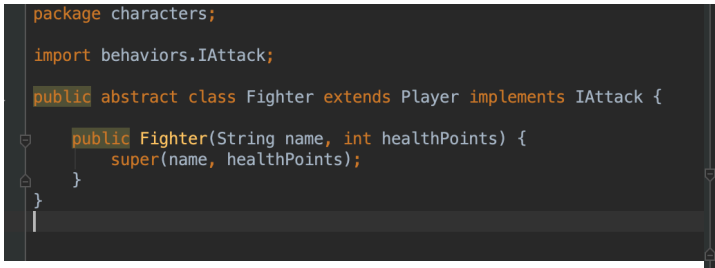


Description here

You can see above an example of inheritance diagram in a fantasy world game were you can implement different interfaces in order to use same method in a different classes also classes that inherit from another class like troll class inherits from enemy abstract class.

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



```java
package characters;

import behaviors.IAttack;

public abstract class Fighter extends Player implements IAttack {

    public Fighter(String name, int healthPoints) {
        super(name, healthPoints);
    }
}
```

```java
package characters;

import behaviors.IAttack;

public class Knight extends Fighter implements IAttack {

        IAttack weapon;

    public  Knight (String name, int healthPoints, IAttack weapon){
        super(name, healthPoints);
        this.weapon = weapon;

    }

    public IAttack getWeapon() { return this.weapon; }

    public int attack() { return this.healthPoints += this.weapon.attack(); }

    public void changeWeapon(IAttack newWeapon) { this.weapon = newWeapon; }

}
```

Description here

On the first screenshot we can see an example of an abstract class (Fighter), on the second screenshot we can see:
an example of class(Knight) that inherits from Fighter and
An example of an object of a Knight ,
An example of a method that uses healthPoints and weapon that were inherited from Fighter

Week 14

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 getTotal() {
        double total = 0;
        for (Item item : this.items) {
            total += item.getPrice() * item.getQuantity();
        }
        return total;
    }

}
```

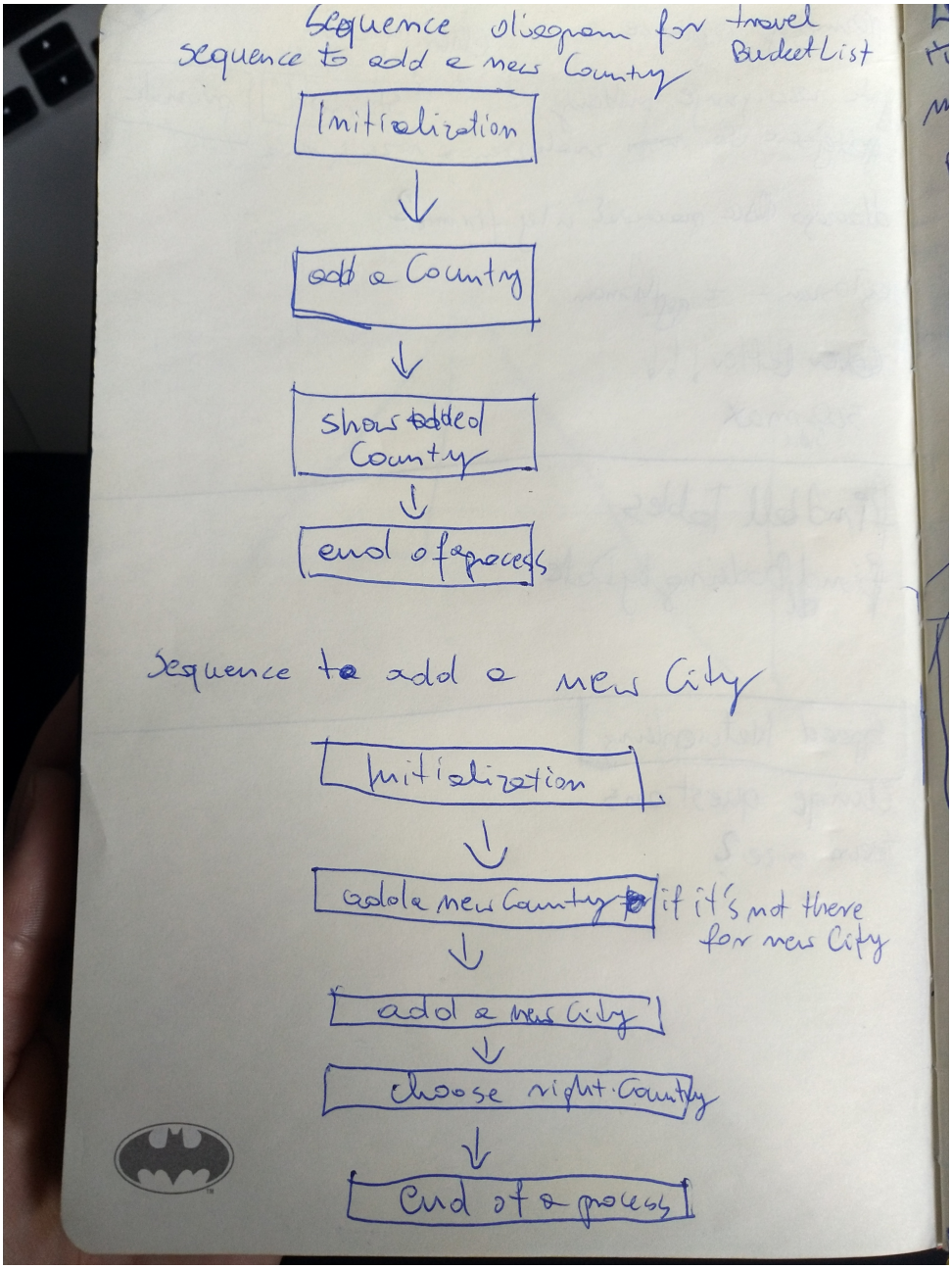
```
public void addGuest(Guest guest) {
    if (this.countCollectionOfGuests() < this.capacity) {
        this.collectionOfGuests.add(guest);
    }
}
```

Description here

On the first screenshot you can see an example of an algorithm which I used to solve the problem of adding a guest to a fully booked room, so this algorithm
Is comparing the size of a room vs its capacity and then is adding a guest if room is not full.
On the second screenshot you can see an example of an algorithm which I used in order to get the total value of a shopping basket based on the price and quantity of all items.

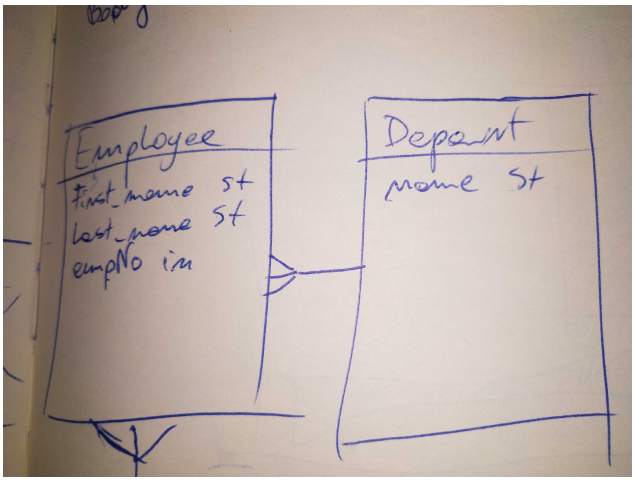
Unit	Ref	Evidence
P	P.7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).

Paste Screenshot here



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

Paste Screenshot here



Description here

On the screenshot above you can see an example of an employee class and department class

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

Paste Screenshot here

Bug/error	Solution	Date
Syntax error, missing coma	Add coma	23.09.2019
End expression expected	Moved end keyword from the middle of a file to the end of the file	05.10.2019
Parameter expected for a method	Added a variable into parentheses	20.10.2019
Can't map object of undefined	Added missing import from another class	21.10.2019