

**PDA: Software Development  
Level 8  
Student Evidence Checklist**


<b>Full name</b>	<b>Emma Roberts</b>
<b>Cohort</b>	<b>G4</b>

The evidence required can be taken from your assignments, homework that you have completed on your own or by creating a specific example for the PDA.

	<b>Unit</b>	<b>Ref.</b>	<b>Evidence</b>	<b>Done</b>
--	-------------	-------------	-----------------	-------------

Week 2

I & T	I.T 5	<p>Demonstrate the use of an array in a program. Take screenshots of:</p> <ul style="list-style-type: none"> <li>*An array in a program</li> <li>*A function that uses the array</li> <li>*The result of the function running</li> </ul>	
		 <pre> 1 class Room 2 3   attr_reader :number_of_rooms 4 5   def initialize(number_of_rooms) 6     @number_of_rooms = [1, 2, 3, 4, 5] 7     @booked_rooms = [] 8     @room_playlist = [] 9     @room_capacity = [] 10  end 11 12  def check_guests_in() 13    @booked_rooms &lt;&lt; @guests 14    return @booked_rooms.count() 15  end 16 17  def test_check_guests_in() 18    assert_equal(1, @room.check_guests_in()) 19  end 20 21 end </pre> <p>     1= <code>room = Room.new</code> ruby room.rb      Run options: --seed 23584      # Running:      .....      Finished in 0.000054s, 4500.000 runs/s, 4500.000 assertions/s.      5 runs, 5 assertions, 0 failures, 0 errors, 0 skips      = <code>room.new</code> </p>	

I & T	I.T 6	<p>Demonstrate the use of a hash in a program. Take screenshots of:</p> <ul style="list-style-type: none"> <li>*A hash in a program</li> <li>*A function that uses the hash</li> <li>*The result of the function running</li> </ul>	
		 <pre> countries = {   uk: {capital: "London",        population: "6 million",        languages: ["English", "Gaelic", "Welsh"]     },    germany: {     capital: "Berlin",     population: "4 million",     languages: ["German", "German Deutsch", "Turkish"]   } }  p countries[:germany][:capital] p countries[:uk][:languages][0] p countries[:germany][:languages][-1] p countries[:germany][:languages].last()  + day_3 git:(master) x ruby hashes.rb "Berlin" "English" "Turkish" "Turkish" </pre>	
I & T		<p>Static and Dynamic testing task A</p> <p><a href="https://github.com/emrob/PDA">https://github.com/emrob/PDA</a></p>	

Unit	Ref.	Evidence	Done
------	------	----------	------

## Week 3

I & T

I.T 3

Demonstrate searching data in a program. Take screenshots of:  
 \*Function that searches data  
 \*The result of the function running

```
Park.prototype.moreThanTwoOffspring = function() {
  var total = [];
  for (var dino of this.enclosure) {
    if (dino.offspring > 2) {
      total.push(dino);
    }
  }
  return total;
}
```

```
it('should get all the dinosaurs with an offspring count of more than 1. function()
  park.addDino(dino1);
  park.addDino(dino2);
  park.addDino(dino3);
  expect(dino1.offspring).toBe(1);
  expect(dino2.offspring).toBe(2);
  expect(dino3.offspring).toBe(3);
  expect(park.moreThanTwoOffspring().length).toBe(2);
});
```

```
➤ homework@1:~/code/learn-work/week_11/Day2/homework$ npm run test
➤ homework@1:~/code/learn-work/week_11/Day2/homework$ npm run test
➤ homework@1:~/code/learn-work/week_11/Day2/homework$ npm run test
```

```

Dinosaur
  ✓ should have a type
  ✓ should have a number of offspring
  ✓ number of offspring is 1000 per year

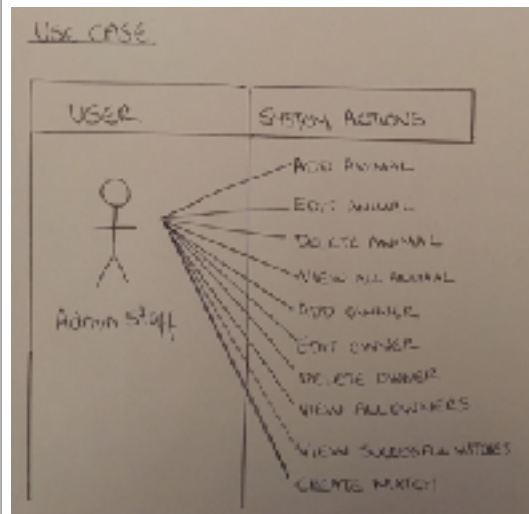
Park
  ✓ enclosure should start empty
  ✓ should be able to add dinosaur
  ✓ should be able to remove all dinosaurs of a particular type
  ✓ should get all the dinosaurs with an offspring count of more than 2

7 pending (0ms)
```

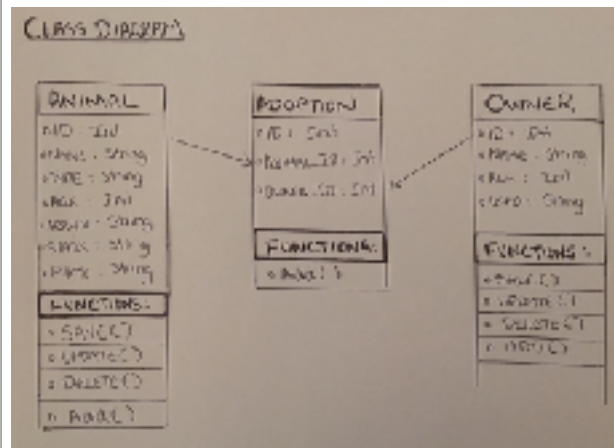
I & T	I.T 4	Demonstrate sorting data in a program. Take screenshots of: *Function that sorts data *The result of the function running
-------	-------	---

	Unit	Ref.	Evidence	Done
--	------	------	----------	------

A & D    A.D 1    A Use Case Diagram



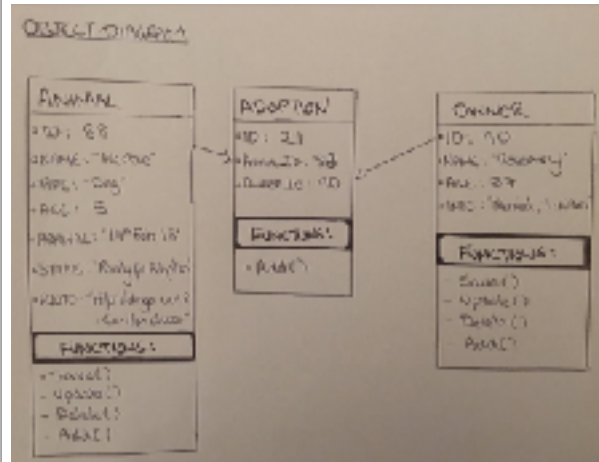
A & D    A.D 2    A Class diagram.



A &amp; D

A.D 3

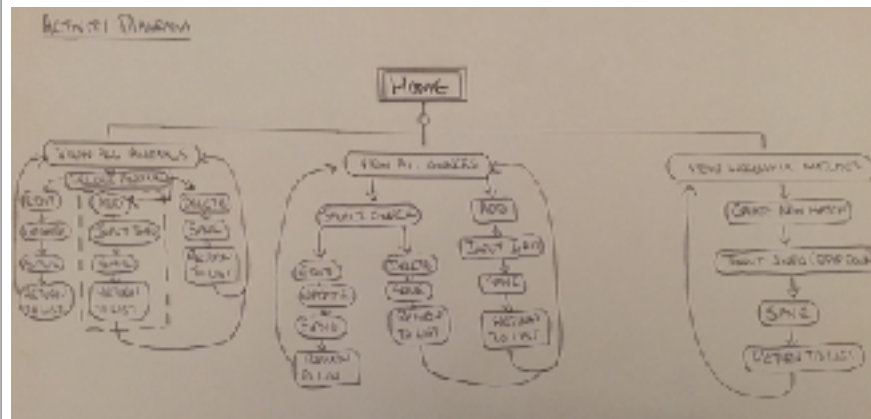
An Object diagram.

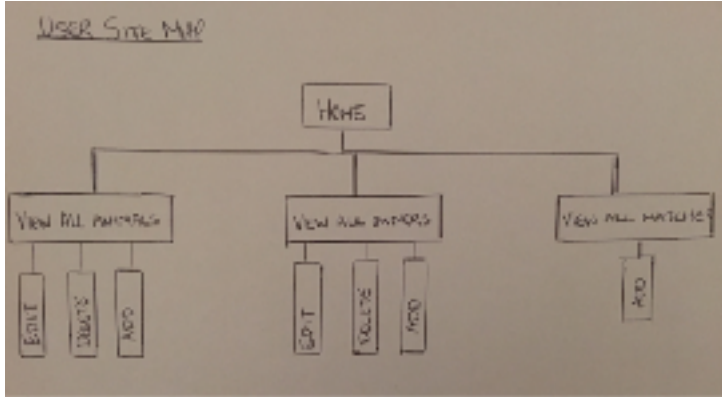


A &amp; D

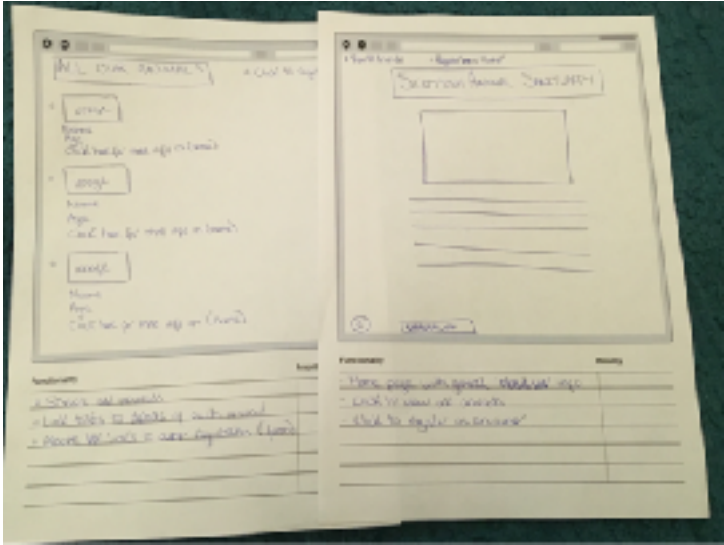
A.D 4

An Activity Diagram



A & D	A.D 6	<p>Produce an Implementations Constraints plan detailing the following factors:</p> <ul style="list-style-type: none"> <li>*Hardware and software platforms</li> <li>*Performance requirements</li> <li>*Persistent storage and transactions</li> <li>*Usability</li> <li>*Budgets</li> <li>*Time</li> </ul>	
P	P 5	<p>Create a user sitemap.</p>  <pre> graph TD     HOME[HOME] --&gt; PRODUCTS[VIEW ALL PRODUCTS]     HOME --&gt; EMPLOYEES[VIEW ALL EMPLOYEES]     HOME --&gt; MATERIALS[VIEW ALL MATERIALS]     PRODUCTS --&gt; EDIT_P[EDIT]     PRODUCTS --&gt; DELETE_P[DELETE]     PRODUCTS --&gt; ADD_P[ADD]     EMPLOYEES --&gt; EDIT_E[EDIT]     EMPLOYEES --&gt; DELETE_E[DELETE]     EMPLOYEES --&gt; ADD_E[ADD]     MATERIALS --&gt; ADD_M[ADD] </pre>	



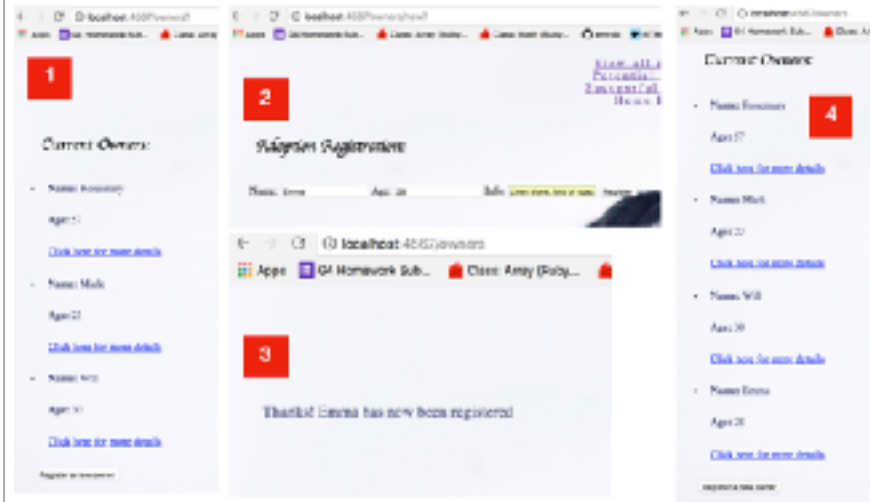
	P  P 6	<p>Produce two wireframe designs.</p> 	
Week 5	P  P 10	<p>Take a screenshot of an example of pseudocode for a function.</p> <pre> 30 31 findDuplicates: function (arr) { 32   let result = []; 33 34   arr.forEach(function(element, index) { 35 36     // Find if there is a duplicate or not 37     if (arr.indexOf(element, index + 1) &gt; -1) { 38 39       // Find if the element is already in the 40       // result array or not 41       if (result.indexOf(element) === -1) { 42         result.push(element); 43       } 44     }); 45 46     return result; 47   }, </pre>	

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



Show an interaction with data persistence. Take a screenshot of:

- \* Data being inputted into your program
- \* Confirmation of the data being saved

The screenshot displays a Ruby application interface. At the top, there are tabs for 'read.rb', 'show.rb', 'homepage.rb', and 'shelter\_controller.rb'. The 'read.rb' tab is active, showing a Ruby script that defines an animal with the following attributes: name 'Herbie', type 'Dog', age 5, arrival\_date 'Feb 14 2018', adoption\_status 'Ready for adoption', and a photo URL. Below the script, the 'ruby' tab is active, showing the command prompt output for running the script. The output confirms the creation of the animal object with all the specified attributes. At the bottom, there is a preview of the animal's profile page, titled 'Herbie Shwartz', which includes a photo of a small dog and some basic information.

- Show an interaction with data persistence. Take a screenshot of:
- \* Data being inputted into your program
  - \* Confirmation of the data being saved
- 
- The screenshot displays a Ruby application interface. At the top, there are tabs for 'read.rb', 'show.rb', 'homepage.rb', and 'shelter\_controller.rb'. The 'read.rb' tab is active, showing a Ruby script that defines an animal with the following attributes: name 'Herbie', type 'Dog', age 5, arrival\_date 'Feb 14 2018', adoption\_status 'Ready for adoption', and a photo URL 'https://dingo.care2.com/pictures/greenLiving/uploads/2017/18/Happy-small-dog.jpg'. Below the script, the 'ruby' button is highlighted. The output shows the animal object being created with all the specified attributes. At the bottom, there is a preview of the animal's profile page, titled 'Herbie Shwartz', featuring a photo of a small dog and a 'Read More' link.

Show an interaction with data persistence. Take a screenshot of:

- \* Data being inputted into your program
- \* Confirmation of the data being saved

The screenshot shows a Ruby console session. The first part shows the creation of an `Animal` object with the following attributes:

```

1: animal4 = Animal.new(
2:   :name => "Herbie",
3:   :type => "Dog",
4:   :age => 5,
5:   :arrival_date => "Feb 14 2018",
6:   :adoption_status => "Ready for adoption",
7:   :photo => "https://dingo.care2.com/pictures/greenliving/uploads/2017/18/happy-small-dog.jpg"
8: )
9: animal4.save!

```

The second part shows the output of `pry(main)> animal4`, which returns an `Animal` object with the following attributes:

```

=> #<Animal:0x007f8252429ec0
  @adoption_status="Ready for adoption",
  @age=5,
  @arrival_date="Feb 14 2018",
  @id=74,
  @name="Herbie",
  @photo=
    "https://dingo.care2.com/pictures/greenliving/uploads/2017/18/happy-small-dog.
jpg",
  @type="Dog">

```

The screenshot shows a web browser displaying the details of a dog named Herbie. The page has a title "Herbie Details" and a large photo of a small dog. Below the photo, there is a section titled "Herbie Details" with the following information:

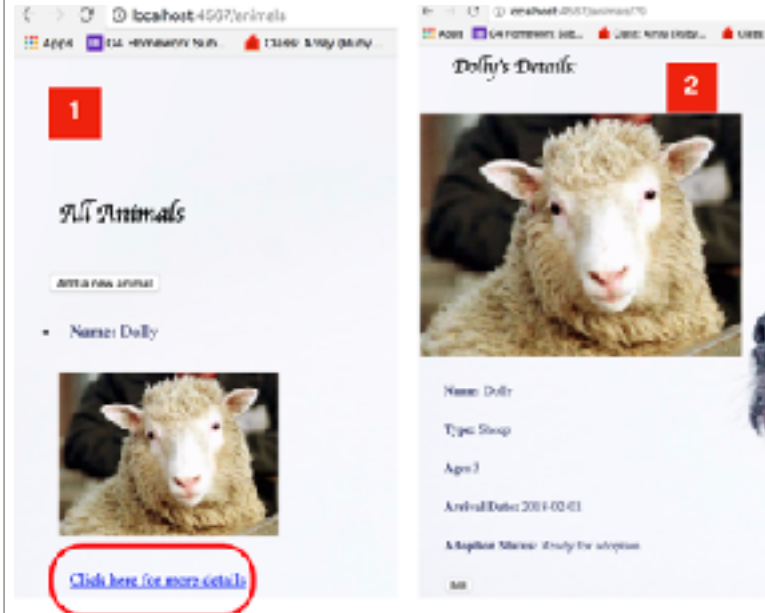
- Name: Herbie
- Age: 5
- Adoption Status: Ready for adoption
- Adoption Date: Feb 14 2018

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





I & T	I.T 7	<p>Demonstrate the use of Polymorphism in a program.</p> <pre> public class Developer extends Employee {     public Developer(String name, String niNumber, double salary){         super(name, niNumber, salary);     } }  public abstract class Employee {     public String name;     public String niNumber;     public double salary;      public Employee(String name, String niNumber, double salary ){         this.name = name;         this.niNumber = niNumber;         this.salary = salary;     } } </pre>	
A & D	A.D 5	An Inheritance Diagram	
I & T	I.T 1	<p>Take a screenshot of an example of encapsulation in a program.</p> <pre> public class Director extends Manager {     private double budget;      public Director(String name, String niNumber, double salary, String deptName, double budget){         super(name, niNumber, salary, deptName);         this.budget = budget;     }      public double getBudget() {         return budget;     } } </pre>	

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

The first screenshot shows the following code:

```

@InheritanceStrategy
abstract class Consumable {
    private String name;
    private FoodCategory foodCategory;
    private Meal meals;

    Consumable() {
    }

    Consumable(String name, FoodCategory foodCategory, Meal meals) {
        this.name = name;
        this.foodCategory = foodCategory;
        this.meals = meals;
    }

    abstract void print();
}

abstract class InheritanceStrategy {
    abstract void print();
}

class TestFood {
    Food food;
    Meal meals;

    TestFood() {
        food = new Food("Food", FoodCategory.BREAD, 100, 1, "Bread");
        meals = new Meal("Meals");
    }

    void print() {
        food.print();
        meals.print();
    }
}

```

The second screenshot shows the following code:

```

import java.util.*;
import java.util.*;

@InheritanceStrategy
public class Food extends Consumable {
    private String name;
    private FoodCategory foodCategory;
    private Meal meals;
}

```

The third screenshot shows the following code:

```

public class TestFood {
    Food food;
    Meal meals;

    TestFood() {
        food = new Food("Food", FoodCategory.BREAD, 100, 1, "Bread");
        meals = new Meal("Meals");
    }

    void print() {
        food.print();
        meals.print();
    }
}

```

# Take a screenshot of the use of Inheritance

- \*A Class
- \*A Class that inherits from
- \*A Method that uses the information

```

class Consumable {
    private String mealType;
    private FoodCategory foodCategory;
    private Meal meal;

    public Consumable(String mealType, FoodCategory foodCategory, Meal meal) {
        this.mealType = mealType;
        this.foodCategory = foodCategory;
        this.meal = meal;
    }

    public String getMealType() {
        return mealType;
    }

    public FoodCategory getFoodCategory() {
        return foodCategory;
    }

    public Meal getMeal() {
        return meal;
    }
}

class Food extends Consumable {
    public Food(String mealType, FoodCategory foodCategory, Meal meal) {
        super(mealType, foodCategory, meal);
    }

    @Override
    public String getMealType() {
        return "Food";
    }
}

```

```

import java.util.*;
import java.util.*;

@Inheritance(strategy = InheritanceType.JOINED)
public class Food extends Consumable {
    private String mealType;
    private FoodCategory foodCategory;
    private Meal meal;

    public Food(String mealType, FoodCategory foodCategory, Meal meal) {
        super(mealType, foodCategory, meal);
    }

    @Override
    public String getMealType() {
        return "Food";
    }
}

```

**Week 7**

P

Take a screenshot of one of your projects where you have worked alone and attach the Github link.

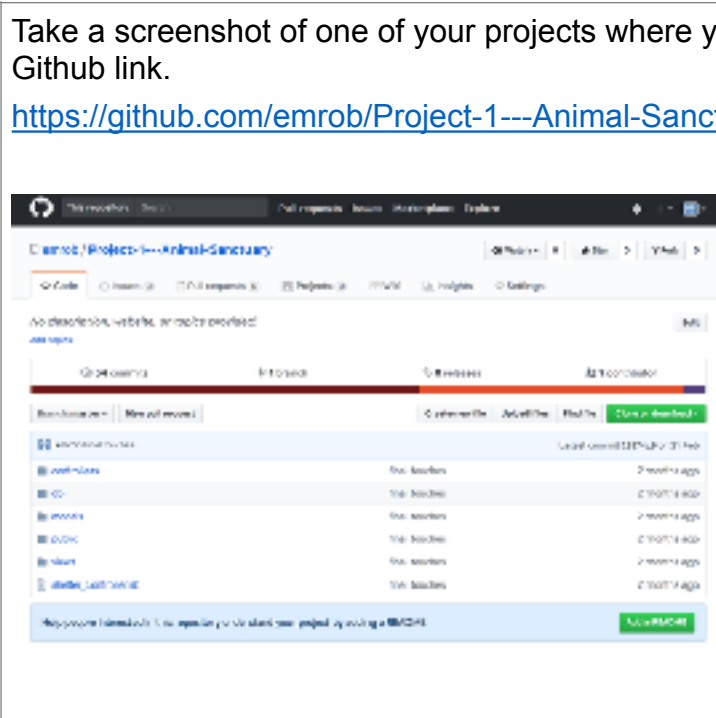
<https://github.com/emrob/Project-1---Animal-Sanctuary>

The screenshot shows the GitHub interface for the repository 'emrob/Project-1---Animal-Sanctuary'. The 'Code' tab is selected, showing a list of files and folders. The files listed are 'animal-sanctuary', 'api', 'models', 'views', and 'utils', each with a file icon and a commit count (e.g., 2 months ago). A green button 'Clone or download' is visible at the bottom right of the file list.

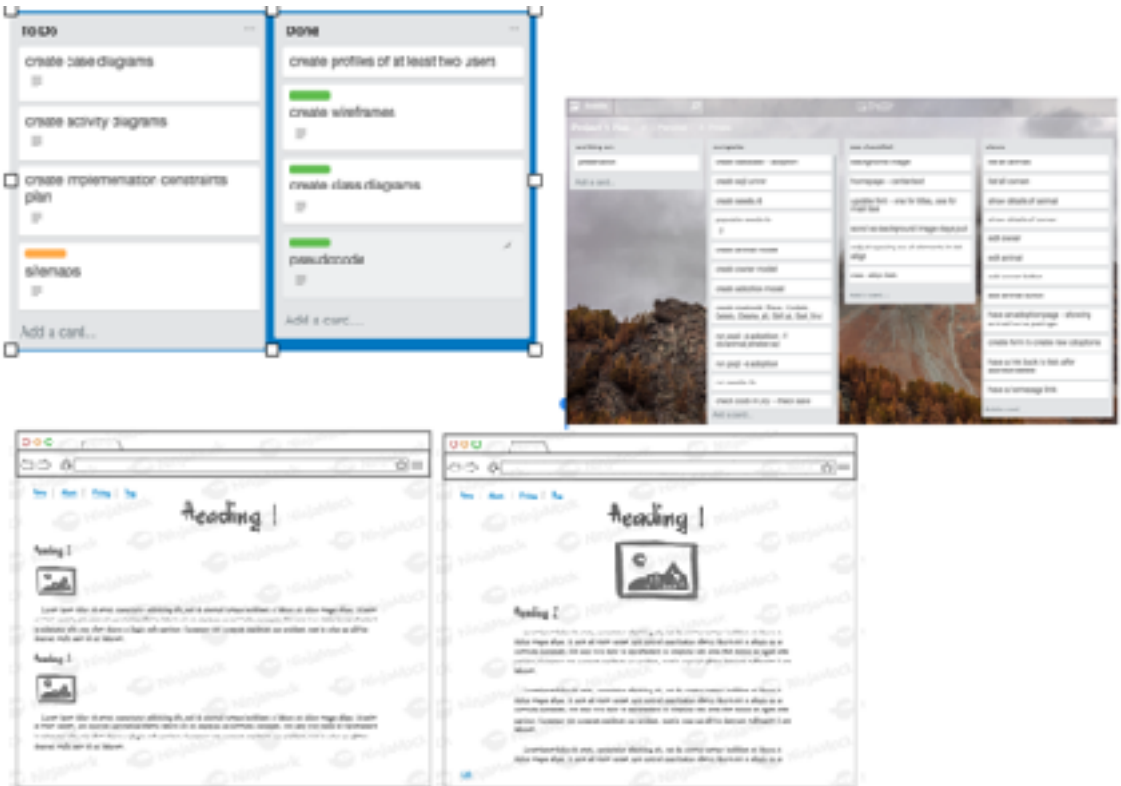
Take a screenshot of one of your projects where you have worked alone and attach the Github link.

<https://github.com/emrob/Project-1---Animal-Sanctuary>

The screenshot shows the GitHub interface for the repository 'emrob/Project-1---Animal-Sanctuary'. The 'Code' tab is selected, showing a list of files and folders. The files listed are 'animal-sanctuary', 'api', 'models', 'views', and 'utils', each with a file icon and a commit count (e.g., 2 months ago). A green button 'Clone or download' is visible at the bottom right of the file list.





P	P 12	<p>Take screenshots or photos of your planning and the different stages of development to show changes.</p> 	
---	------	--	--

Unit	Ref.	Evidence	Done
I & T		<p>Unit, integration and acceptance testing task B</p> <p><a href="https://github.com/emrob/JavaScript-Testing">https://github.com/emrob/JavaScript-Testing</a></p>	

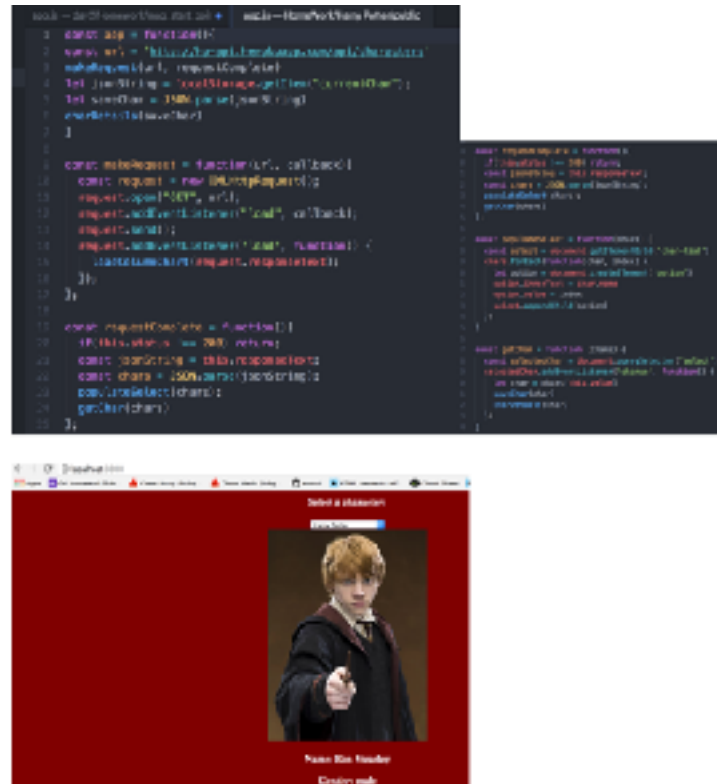
Week  
11

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

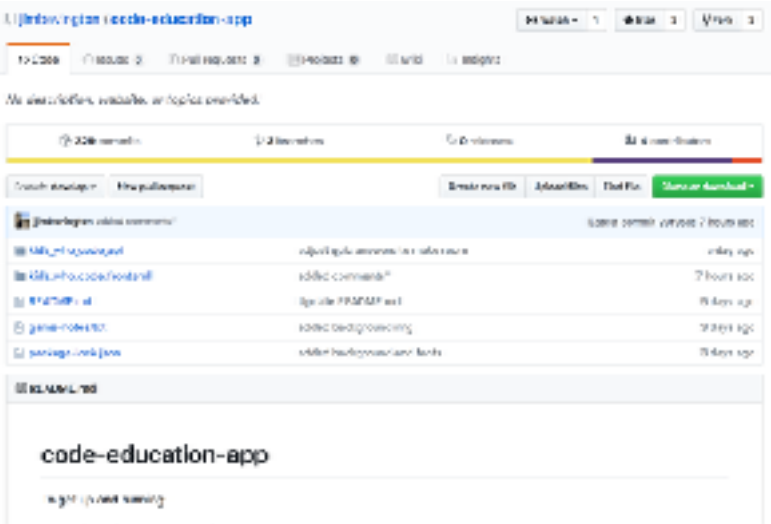



Unit

Ref.

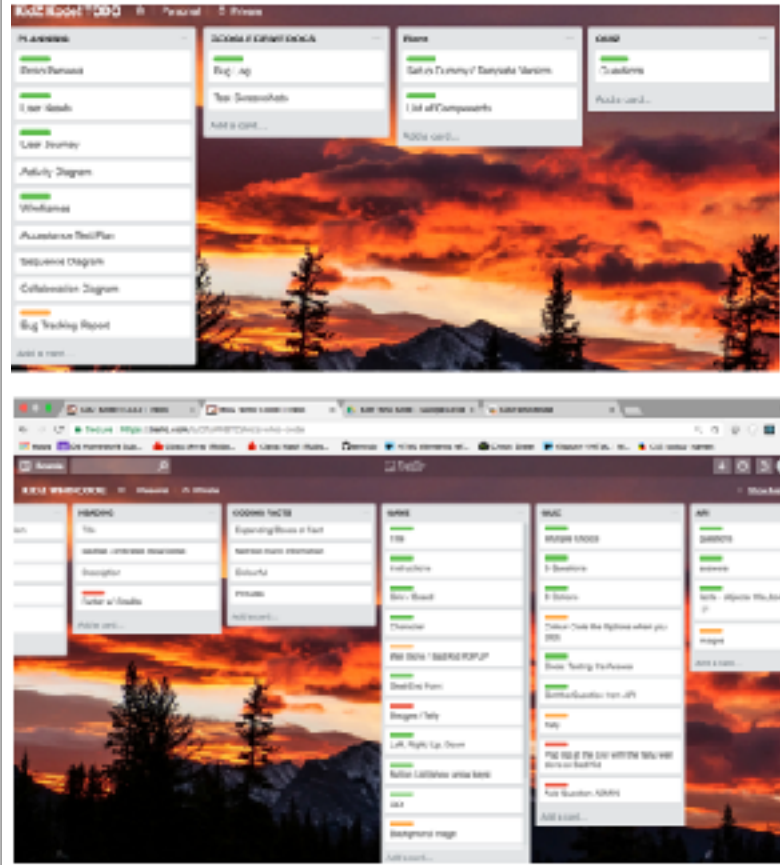
Evidence

Done

P	P 1	<p>Take a screenshot of the contributor's page on Github from your group project to show the team you worked with.</p> 	
P	P 2	<p>Take a screenshot of the project brief from your group project.</p> 	

P

Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.



P

P 4

Write an acceptance criteria and test plan.

Acceptance Test Plan

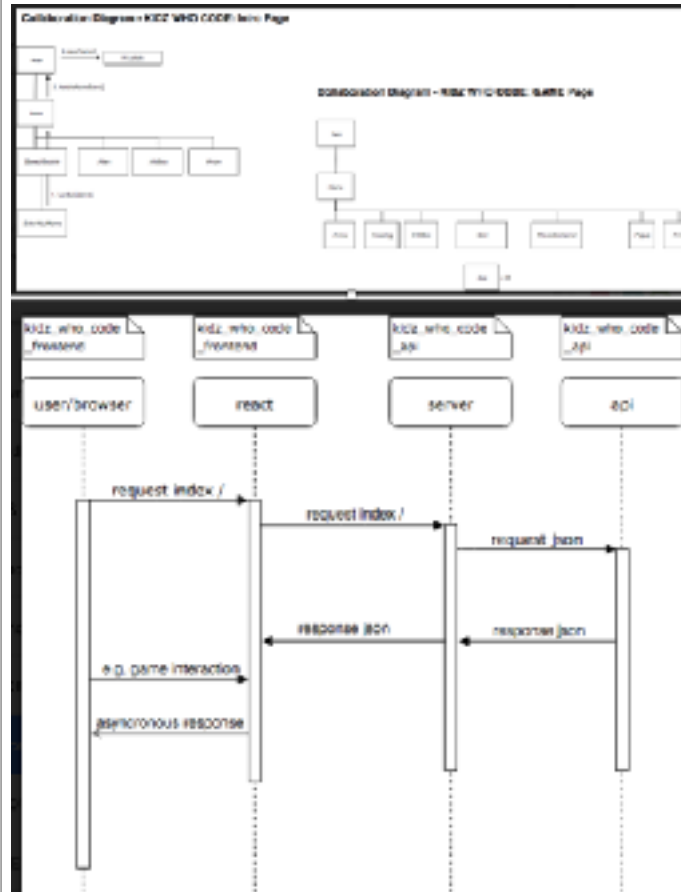
File Edit View Insert Format Data Tools Actions Help

File Edit View Insert Format Data Tools Actions Help

	A	B	C	D
1				
2		<b>Acceptance Criteria</b>	<b>Expected Results/Output</b>	<b>Pass/Fail</b>
3		A user is able to read each fact	When a user clicks on a question, a related fact is shown within 2 seconds	Pass
4		A user is able to navigate to each page	When a user clicks on a given name, they should be taken to the next page.	Pass
5		On the game page, navigation buttons are displayed for the user to use	User should see the navigation buttons they've pressed below their click log	Pass
6		A user navigates the robot to the robot on the game page	Robot should move according to the action buttons pressed by the user once the GO button is clicked	Pass
7		A user should receive confirmation after playing the game	If the user has played correctly they will be told so and prompted to take the quiz. Otherwise they will be given the option to try again.	Pass
8		ONE QUESTION IS SHOWN AT A TIME AT THE END PAGE	User sees one question at a time, with it at the end of the page	Pass
9		User is told whether they have answered correctly or not	Confirmation will show if the answer entered is chosen, along with a next question button. Confirmation that user is prompted to try again	Pass
10		Confirmation is received once the user has completed the quiz	A message displays once all five questions have been answered	Pass

Week

P	P 7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).	
P	P 8	Produce two object diagrams.	



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.

```
<label for="adoption_status">Adoption Status:</label>
<select name="adoption_status" id="adoption_status">
  <% for adoption_status in @adoption_status %>
    <option value="<%= adoption_status %>"
      <%= "selected" if adoption_status == @animal.adoption_status %>
    ><%= adoption_status %></option>
  <% end %>
</select>
```

```
<label>
  Select an animal:
  <select name="animal_id">
    <% for animal in @animals %>
      <% if animal.adoption_status == "Ready for adoption" %>
        <option value="<%= animal.id %>"><%= animal.name %>
      </option>
    <% end %>
  <% end %>
</select>
</label>
```

I have chosen the above because they transformed my app from just displaying data, to showing only relevant data. So if an animal was adopted or needing to be trained, it would no longer show on the drop down list - the drop down list would only show those ready for adoption

P

P 17

Produce a bug tracking report

**BUG TRACKING REPORT** ☆

File Edit View Insert Format Data Tools Window Help Last edited was made 3 days ago

100% - \$ % .00 123 - Arial - 18 - B I U

**REQUIREMENT**

	A	B	C	D
	REQUIREMENT	FAILED Log	Debug FA	PASSED Log
1	describe task	task if failed	describe action taken to fix bug	task Passed
2	Game: cellStates same var should store move value when Move button clicked. Failed when Move changed to img rather than button		Refactored addMove to use getAttribute("col") from Move img object.	
3	Game: Rocco: Ship should move around whole grid. Failing to land in last cell		Refactored bounds check to <= rather than > getRow	
4	Game: Rocco: ship should stop at Grid left/right boundaries.		added further checks in executeMoves to stop execution of left/right moves when at grid bounds	
5	SpeechBubble should be able to contain an EnterYourName form alongside p tags		refactored SpeechBubble to put things in p tags and other elements in article tags	
6	EnterYourName form should pass username to Ago state		made EnterYourName a container with state of name, updating when text input changes	