PDA: Software Development
Level 8
Student Evidence Checklist

Full name	Jia Sin Wong
Cohort	G2

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.

	Unit	Ref.	Evidence	Done
Week 2	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	✔

```ruby
require 'minitest/autorun'
require_relative '../array'

class TestArray < MiniTest::Test

  def setup
    @numbers = [3, 6, 9, 12, 15, 18, 21]
  end

  def test_all_numbers_in_array

    assert_equal([3, 6, 9, 12, 15, 18, 21], @numbers)
  end
end
```

```ruby
def all_numbers_in_array
  return numbers
end
```

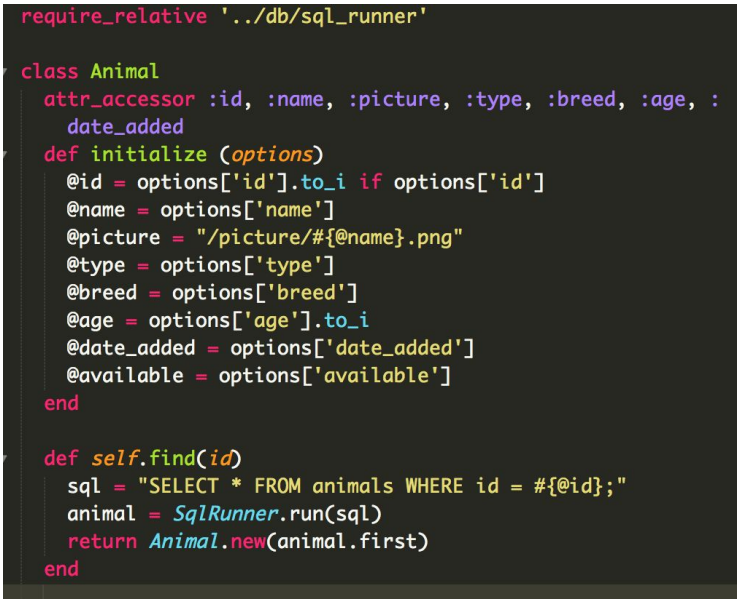
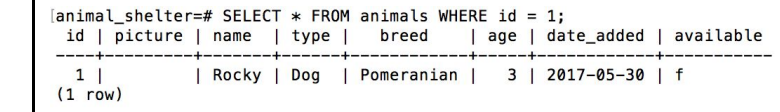
```
Run options: --seed 24060

# Running:

.

Finished in 0.000885s, 1129.4751 runs/s, 1129.4751 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
```

	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 	✔
	I & T		Static and Dynamic testing task A https://github.com/AlphaHydroxy/Professional-Development-Awards/tree/master/Section_A_Static_and_Dynamic_Testing	✔

```
@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,
      },
```

```
def pets_by_breed(pet_shop, breed)
  dog_found = []
  for pet in pet_shop[:pets]
    if pet [:breed] == breed
    dog_found.push(pet)
    end
  end
  return dog_found
end
```

```
# Running:

.

Finished in 0.000874s, 1144.2918 runs/s, 1144.2918 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
→  evidence_week_2_hash git:(master) ✗
```

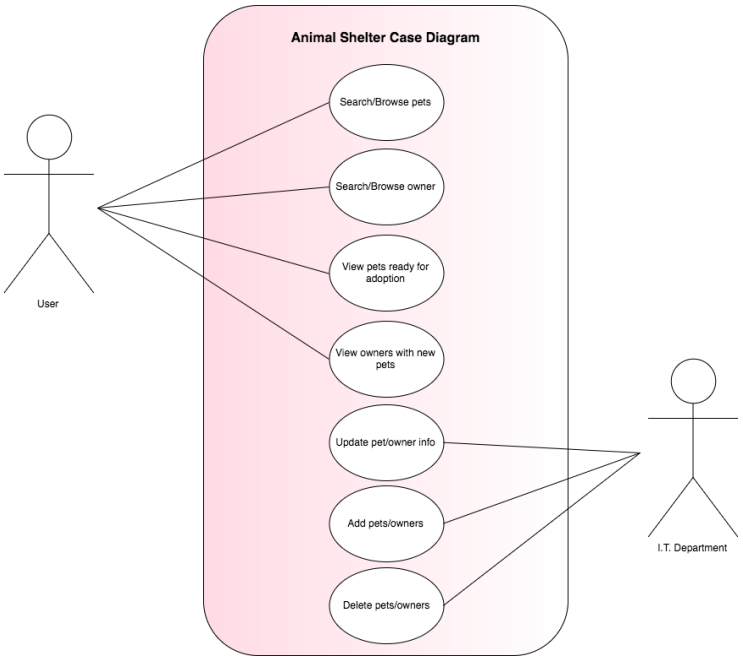
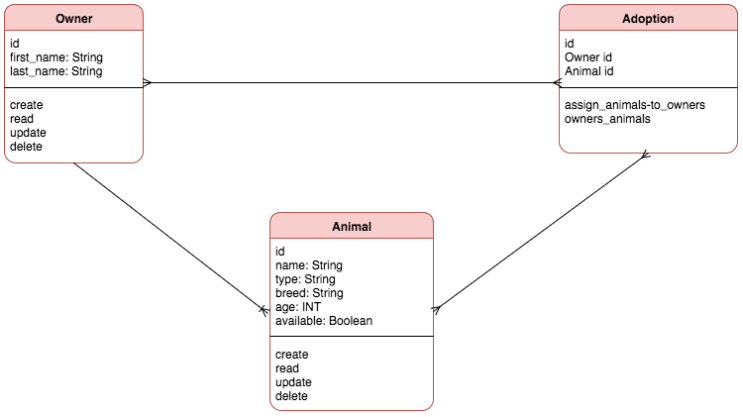
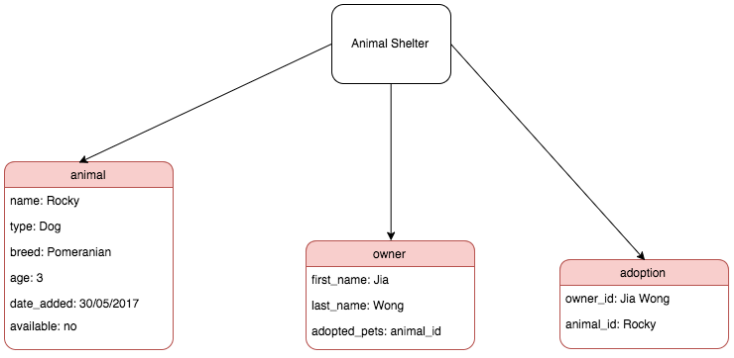
	Unit	Ref.	Evidence	Done
Week 3	I & T	I.T 3	Demonstrate searching data in a program. Take screenshots of: *Function that searches data  *The result of the function running 	✔
	I & T	I.T 4	Demonstrate sorting data in a program. Take screenshots of: *Function that sorts data  *The result of the function running 	✔

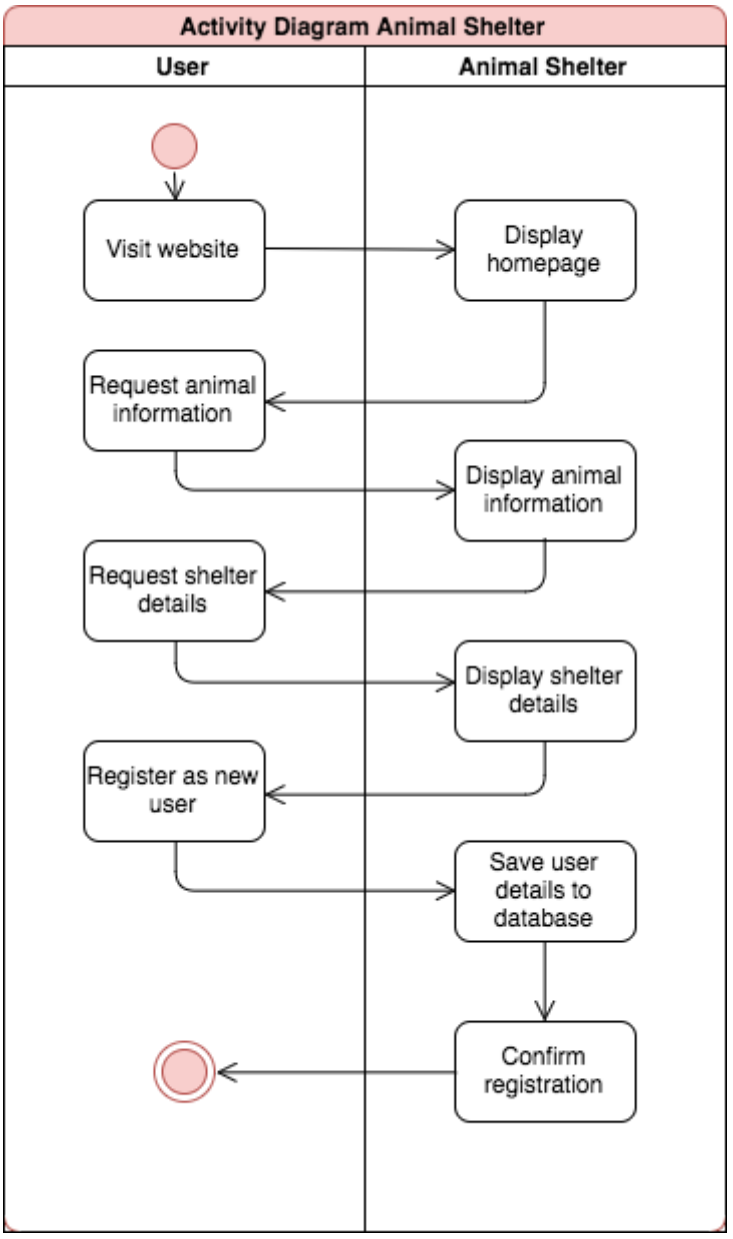
```ruby
require_relative '../db/sql_runner'

class Animal
  attr_accessor :id, :name, :picture, :type, :breed, :age, :
    date_added
  def initialize (options)
    @id = options['id'].to_i if options['id']
    @name = options['name']
    @picture = "/picture/#{@name}.png"
    @type = options['type']
    @breed = options['breed']
    @age = options['age'].to_i
    @date_added = options['date_added']
    @available = options['available']
  end

  def self.find(id)
    sql = "SELECT * FROM animals WHERE id = #{@id};"
    animal = SqlRunner.run(sql)
    return Animal.new(animal.first)
  end
```

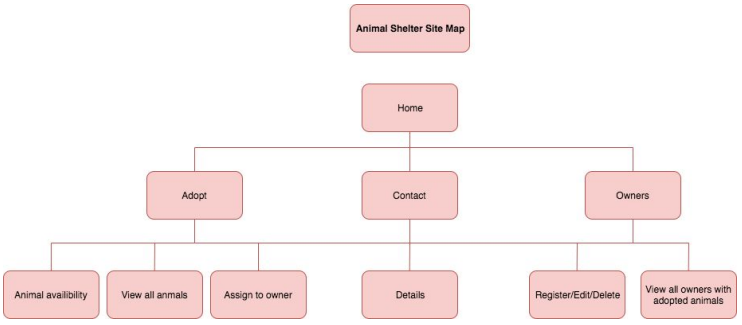
```
animal_shelter=# SELECT * FROM animals WHERE id = 1;
 id | picture | name  | type |   breed    | age | date_added | available
----+---------+-------+------+------------+-----+------------+-----------
  1 |         | Rocky | Dog  | Pomeranian |   3 | 2017-05-30 | f
(1 row)
```

```ruby
  def sort_by_date()
    sql = "SELECT date_added FROM animals WHERE animals.date;"
    SqlRunner.run(sql)
  end
```

```
animal_shelter=# SELECT * FROM animals ORDER BY date_added;
 id | picture |  name  |  type   |       breed        | age | date_added | available
----+---------+--------+---------+--------------------+-----+------------+-----------
  3 |         | Lilly  | Reptile | Terrapin           |   9 | 2017-05-22 | t
  2 |         | Terry  | Reptile | Terrapin           |   9 | 2017-05-22 | f
  1 |         | Rocky  | Dog     | Pomeranian         |   3 | 2017-05-30 | f
  5 |         | Diego  | Dog     | Jack Russell x Pug |   2 | 2017-06-02 | t
  6 |         | Freya  | Dog     | German Shepherd    |   8 | 2017-06-08 | t
  7 |         | Minnie | Dog     | Pomeranian         |   4 | 2017-06-09 | t
  4 |         | Chase  | Dog     | Jack Russell       |   5 | 2017-06-12 | t
(7 rows)
```

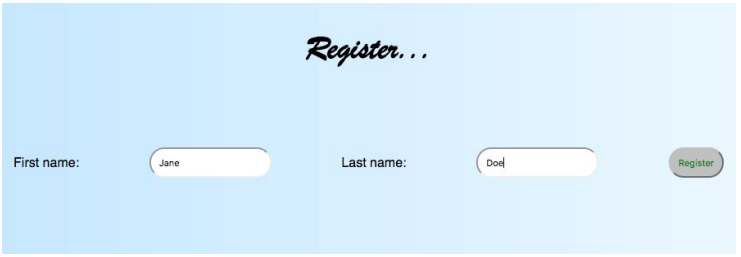
	Unit	Ref.	Evidence	Done
	A & D	A.D 1	A Use Case Diagram 	✔
	A & D	A.D 2	A Class diagram. 	✔
Week 5	A & D	A.D 3	An Object diagram. 	✔

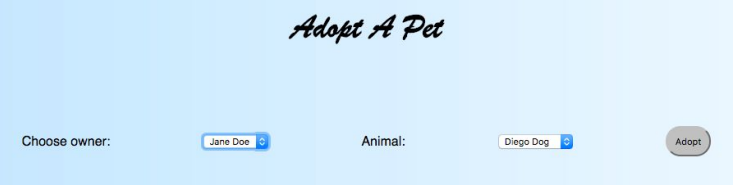
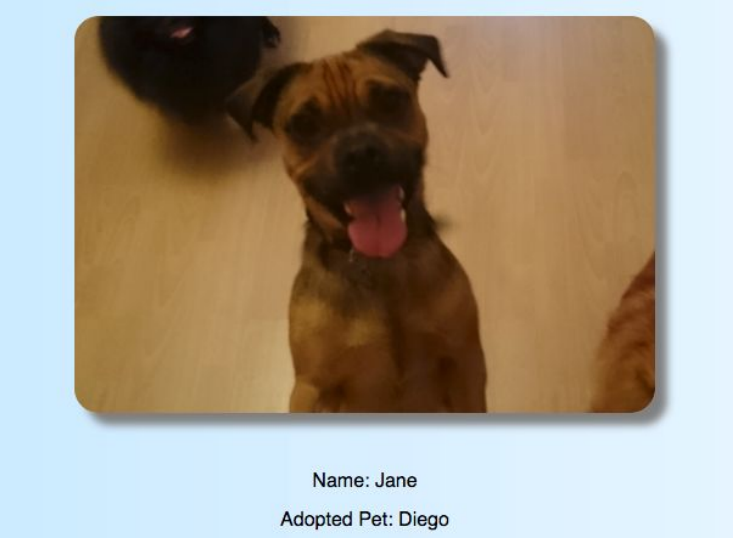
	A & D	A.D 4	An Activity Diagram 	✔

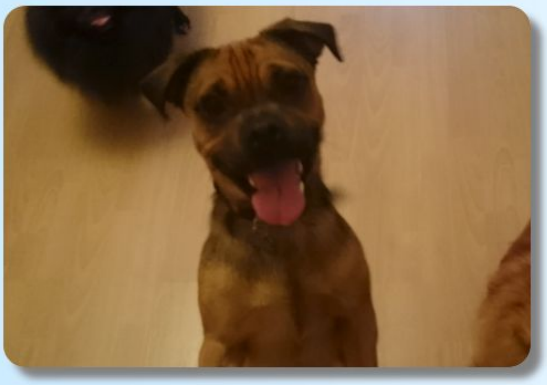
	A & D	A.D 6	Produce an Implementations Constraints plan detailing the following factors: *Hardware and software platforms *Performance requirements *Persistent storage and transactions *Usability *Budgets *Time Implementation Constraints Plan **Constraint** / **Possible Effect of Constraint** / **Solution** Hardware and software platforms — Certain features may not be compatible — Prepare for different browsers, screen sizes Performance requirements — Experience longer load times — Ensure smaller files (e.g img) are used Persistent storage — Too many users querying the database at the same time — Improve/expand network Usability — Limited usage due to personal restrictions — Use semantic mark up for screen readers, intuitive layouts Budgets — No budget — - Time limitations — 6 days to complete project - may not have additional functions featured — -	✔
	P	P 5	Create a user sitemap.  	✔

	P	P 6	Produce two wireframe designs.	✔

	P	P 10	Take a screenshot of an example of pseudocode for a function.
			`#write a method on Adoptions Class to create table entry for adoptions`
			`def save()`
			`#insert and return values into adoptions table`
			`sql = "INSERT INTO adoptions (owner_id, animal_id) VALUES (#{@owner_id}, #{@animal_id}) RETURNING *;"`
			`#assign local variable to run SQL code through SqlRunner class`
			`adoption = SqlRunner.run(sql).first`
			`#assign ID to new entry in adoptions table`
			`@id = adoption['id'].to_i`
			`#save animal id to pet_adopted method`
			`pet_adopted(adoption['animal_id'])`
			`end`
				✔

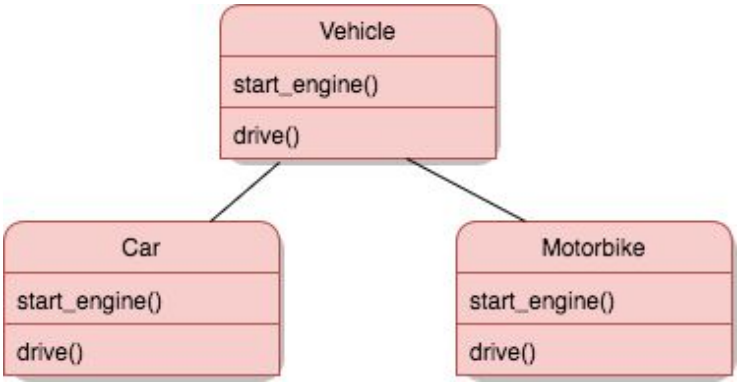
	P	P 13	Show user input being processed according to design requirements. Take a screenshot of: * The user inputting something into your program  * The user input being saved or used in some way 	✔

	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 	✔
	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  Name: Jia Adopted Pet: Terry  Name: Jane Adopted Pet: Diego	

	Unit	Ref.	Evidence	Done

Week 6	I & T	I.T 7	Demonstrate the use of Polymorphism in a program.
			`public class Employee {`
			`private int empId;`
			`private String name;`
			`private String ssn;`
			`private double salary;`
			`public Employee(int empId, String name, String ssn, double salary) {`
			`this.empId = empId;`
			`this.name = name;`
			`this.ssn = ssn;`
			`this.salary = salary;`
			`}`
			`public class Manager extends Employee {`
			`public String deptName;`
			`public Manager(int empId, String name, String ssn, double salary, String deptName) {`
			`super(empId, name, ssn, salary);`
			`this.deptName = deptName;`
			`}`
			`public class Director extends Manager {`
			`private double budget;`
			`public Director(int empId, String name, String ssn, double salary, String deptName, double budget){`
			`super(empId, name, ssn, salary, deptName);`
			`this.budget = budget;`
			`}`
				✔

	Unit	Ref.	Evidence	Done
Week 7	A & D	A.D 5	An Inheritance Diagram 	✔

	I & T	I.T 1	Take a screenshot of an example of encapsulation in a program.
			`public class Food implements Serializable{`
			`private String name;`
			`private double carbohydrate;`
			`private double protein;`
			`private double fat;`
			`public Food(String name, double carbohydrate, double protein, double fat){`
			`this.name = name;`
			`this.carbohydrate = carbohydrate;`
			`this.protein = protein;`
			`this.fat = fat;`
			`}`
				✔

	I & T	I.T 2	Take a screenshot of the use of Inheritance in a program. Take screenshots of: *A Class 	✔

```java
public class User implements Serializable{

    private String name;
    private String gender;
    private int weight;

    public User(String name, String gender, int weight){
        this.name = name;
        this.gender = gender;
        this.weight = weight;
    }

    public String getName() {
        return name;
    }

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

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public int getWeight() {
        return weight;
    }

    public void setWeight(int weight) {
        this.weight = weight;
    }
}
```

*A Class that inherits from the previous class

```
public class EatFood implements Serializable{

    User user;
    MealTime mealTime;
    Food food;

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public MealTime getMealTime() {
        return mealTime;
    }

    public void setMealTime(MealTime mealTime) {
        this.mealTime = mealTime;
    }

    public Food getFood() {
        return food;
    }

    public void setFood(Food food) {
        this.food = food;
    }
```

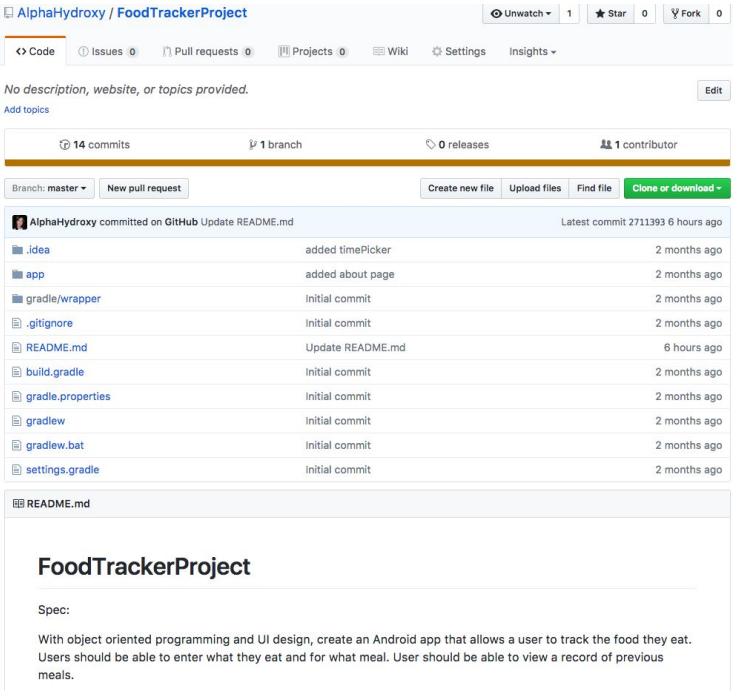
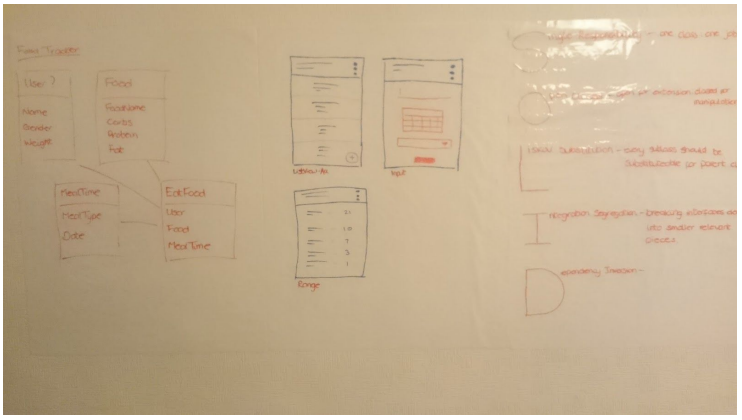
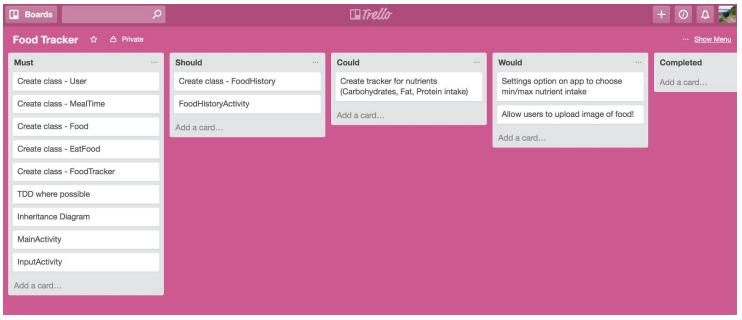
*An Object in the inherited class

```
eatFoodEvent.setUser(new User("Jia", "Female", 56));
```

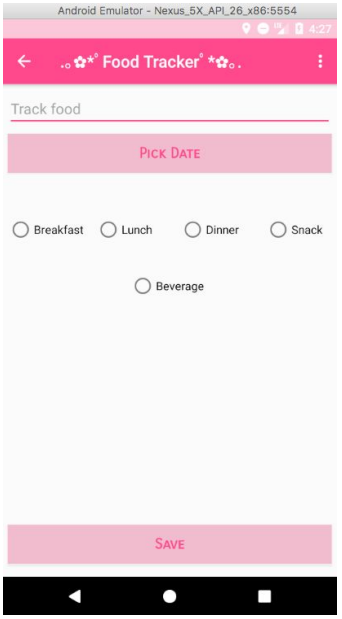
*A Method that uses the information inherited from another class.

```
public String toString(){
  return "User: " + this.user.getName() + " | Weight: " + user.getWeight() + "kg" + "\nFood : "
         + food.getName() + "\nMeal Time: "  +  mealTime.getPrimary() + " "  +
         mealTime.getSecondary() + "\n" + "Date : " + mealTime.getDate();
}
```

```
private MealTime getSelectedMealTime(Date selectedDate) {
    String primary = "";
    String secondary = "";

    if (radio_breakfast.isChecked()) {
        primary = "Breakfast";
    }
    if (radio_lunch.isChecked()) {
        primary = "Lunch";
    }
    if (radio_dinner.isChecked()) {
        primary = "Dinner";
    }

    if (radio_snack.isChecked()) {
        primary = "Snack";
    }
    if (radio_beverage.isChecked()) {
        secondary = "Beverage";
    }
    return new MealTime(primary, secondary, selectedDate);
}
```

	P	P 11	Take a screenshot of one of your projects where you have worked alone and attach the Github link.  https://github.com/AlphaHydroxy/FoodTrackerProject	✔
	P	P 12	Take screenshots or photos of your planning and the different stages of development to show changes.   	✔


Trello
Private
Should
Create class - FoodHistory
FoodHistoryActivity
Add a card...
Could
Create tracker for nutrients (Carbohydrates, Fat, Protein intake)
Add a card...
Would
Settings option on app to choose min/max nutrient intake
Allow users to upload image of food!
Add a card...
Completed
Create class - User
Create class - MealTime
Create class - Food
Create class - EatFood
Create class - FoodTracker
TDD where possible
MainActivity
InputActivity
Add a card...


Android Emulator - Nexus_5X_API_26_x86:5554
4:27
Food Tracker
Track food
PICK DATE
Breakfast
Lunch
Dinner
Snack
Beverage
SAVE


Android Emulator - Nexus_5X_API_26_x86:5554
4:27
Food Tracker
2017
Sun, Sep 10
September 2017
S M T W T F S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
CANCEL
OK
SAVE

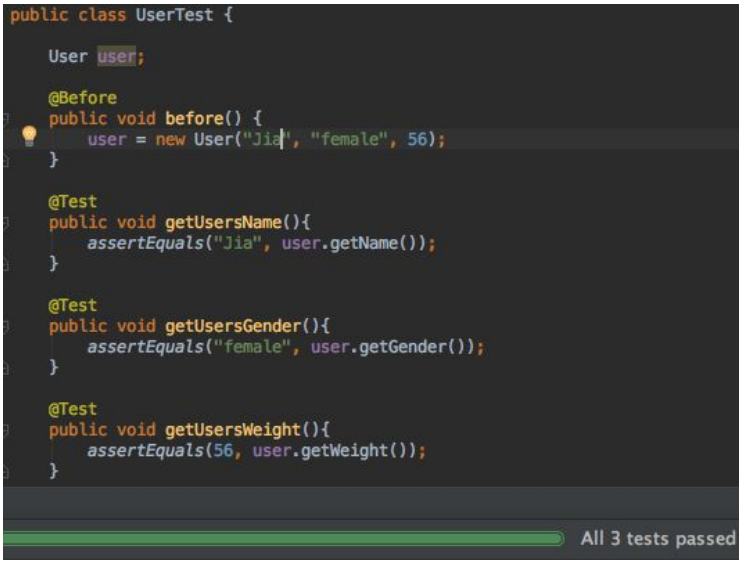

Android Emulator - Nexus_5X_API_26_x86:5554
4:27
Food Tracker
User: Jia | Weight: 56kg
Food : Coffee
Meal Time: Beverage
Date : Sun 10-09-17
User: Jia | Weight: 56kg
Food : Testing
Meal Time: Breakfast
Beverage
Date : Sun 10-09-17
User: Jia | Weight: 56kg
Food : Burger
Meal Time: Breakfast
Beverage
Date : Sun 10-09-17
User: Jia | Weight: 56kg
Food : Coffee
Meal Time: Beverage
Date : Sun 10-09-17

	Unit	Ref.	Evidence	Done
Week 10	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	✔

```
public class UserTest {

    User user;

    @Before
    public void before() {
        user = new User("Jia", "female", 56);
    }

    @Test
    public void getUsersName(){
        assertEquals("Jia", user.getName());
    }

    @Test
    public void getUsersGender(){
        assertEquals("female", user.getGender());
    }

    @Test
    public void getUsersWeight(){
        assertEquals(56, user.getWeight());
    }

}
```

```
public class UserTest {

    User user;

    @Before
    public void before() {
        user = new User("Ji", "male", 55);
    }

    @Test
    public void getUsersName(){
        assertEquals("Jia", user.getName());
    }

    @Test
    public void getUsersGender(){
        assertEquals("female", user.getGender());
    }

    @Test
    public void getUsersWeight(){
        assertEquals(56, user.getWeight());
    }
```

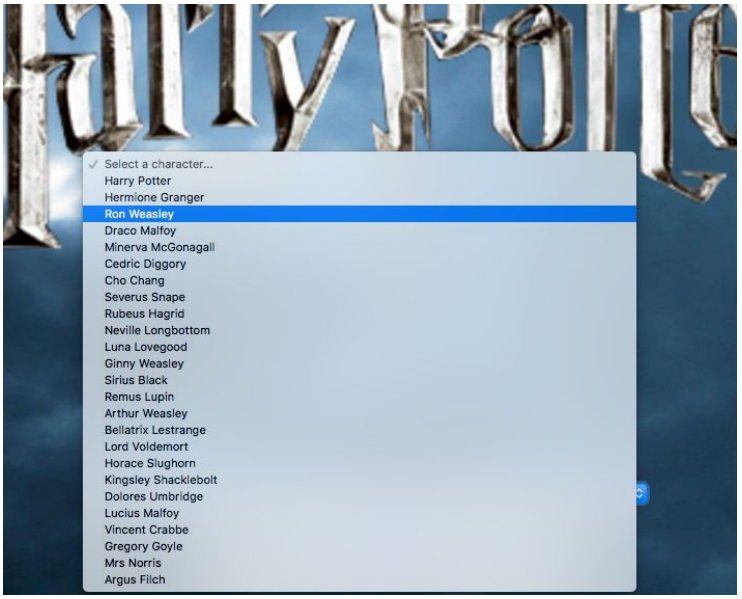
All 3 tests failed

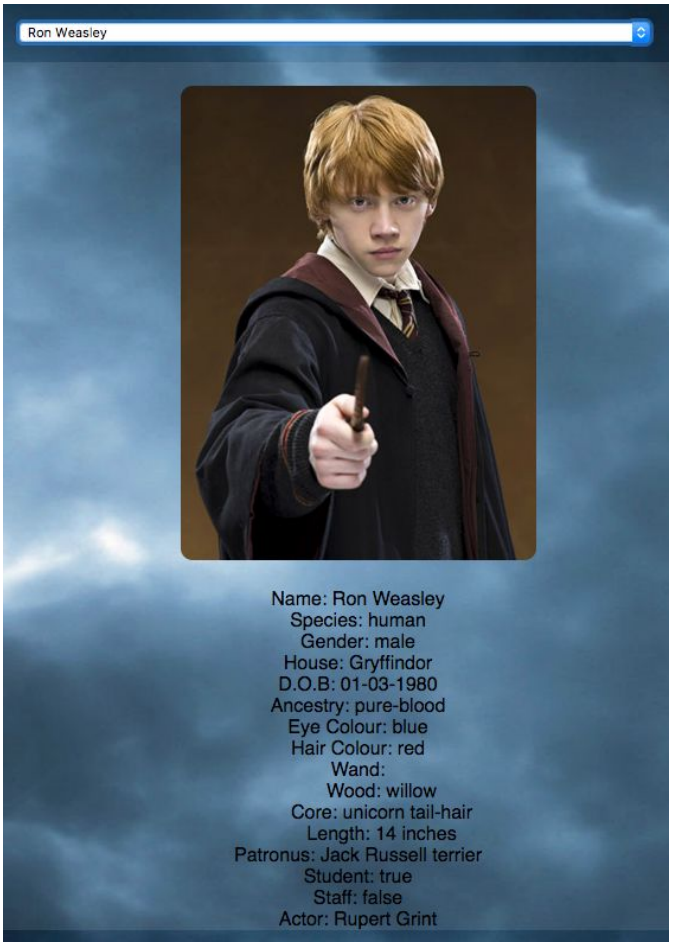
			 * The test code passing 	

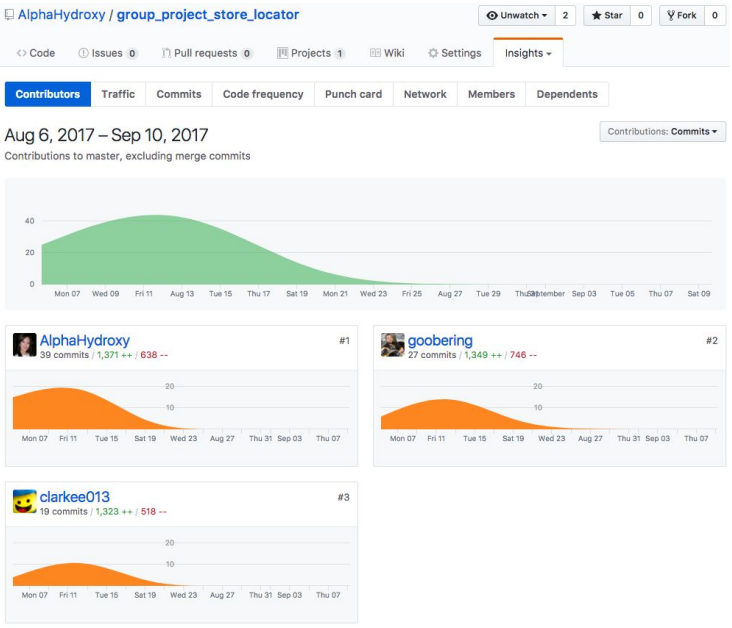
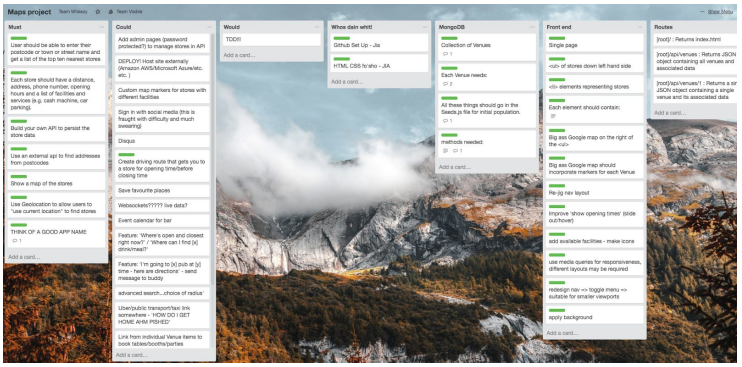
	Unit	Ref.	Evidence	Done
Week 11	I & T		Unit, integration and acceptance testing task B https://github.com/AlphaHydroxy/unit-and-integration-task	✔

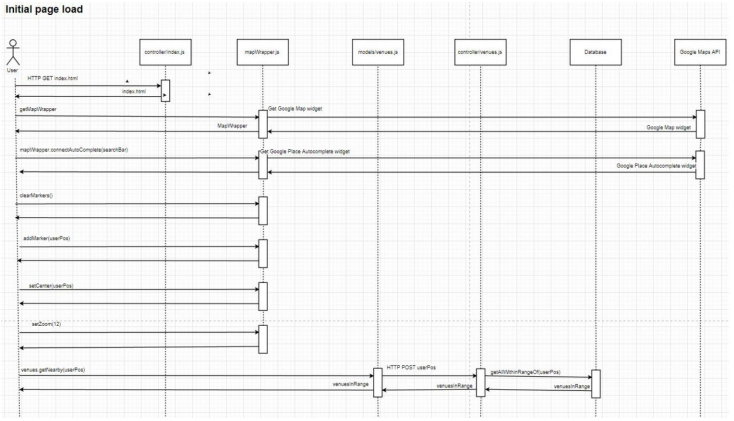
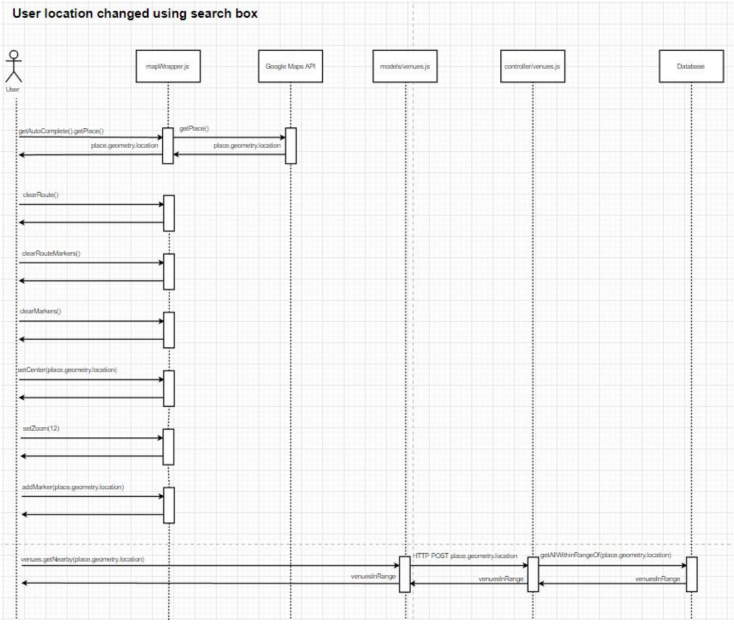
	P	P 16	Show an API being used within your program. Take a screenshot of: * The code that uses or implements the API * The API being used by the program whilst running	✔

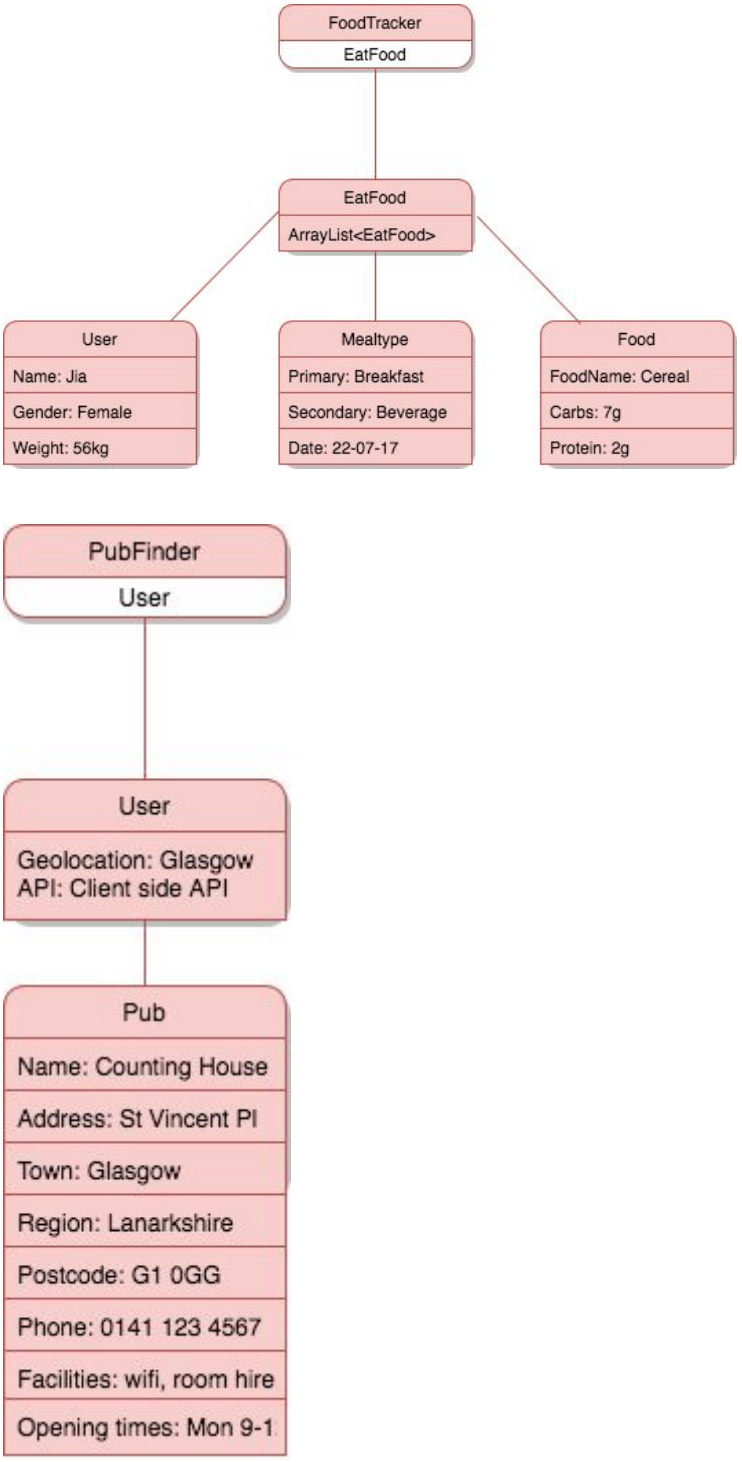
```
var app = function(){
  var url = "http://hp-api.herokuapp.com/api/characters";
  makeRequest(url, requestComplete);
```





	Unit	Ref.	Evidence	Done
Week 13	P	P 1	Take a screenshot of the contributor's page on Github from your group project to show the team you worked with. 	✔
	P	P 2	Take a screenshot of the project brief from your group project. Store Finder Global pub giant Withoutaspoon, wants a pub finder that allows users to "Find your nearest store" for their website. User can enter their postcode or town or street name and get a list of the venues within 3 miles, and a map showing all of the venues. Each has a distance, address, phone number, opening hours and a list of facilities and services (e.g. cash machine, car parking).	✔
	P	P 3	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 Criteria** / **Expected Result/Output** / **Pass /Fail** User should be able to enter postcode/address get list of nearest pubs — List of pubs to show on homepage — Pass Each pub should have distance, address, phone number, opening hours & other facilities — list view on homepage for the details & facilities should be seen — Pass Facilities to show to load from internal API storage — The pubs and facilities should show correctly on the list on the list view of the webpage — Pass External API to be used for address look up and populates to map view — Markers for pubs saved in internal API showing on map — Pass Geolocation to allow user to see pubs around their location — Pubs shown should be within distance of the user — Pass User should be able to click on pub markers to display walking route — Walking route displayed on map from current location — Pass User should be able to save favourite pub — not implemented yet — Fail	✔
	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. **Looping through enums in Suit and enums in Rank and assigning each possible pair to an ArrayList of cards** **loop through array of words and increment by one each word is repeated** **if radio button is checked, assign string to variable**	✔

```
public void populate() {
    for (Suit suit : Suit.values()) {
        for (Rank rank : Rank.values()) {
            cards.add(new Card(rank, suit));
        }
    }
}
```

```
public String countWordOccurrence() {
    String[] wordOccurrence = this.word.trim().split("[(' '),.-]");
    Map<String, Integer> map = new HashMap<>();
    for (String str : wordOccurrence) {
        if (map.get(str) != null) {
            map.put(str, map.get(str) +1);
        } else {
            map.put(str, 1);
        }
    }
    Set<String> str2 = map.keySet();
    String concatenatedOutput = "";
    for(String str : str2){
        concatenatedOutput += ("Words: " + str + " repeated " + map.get(str) + " times\n");
    }
    return concatenatedOutput;
}
```

```
private MealTime getSelectedMealTime(Date selectedDate) {
    String primary = "";
    String secondary = "";

    if (radio_breakfast.isChecked()) {
        primary = "Breakfast";
    }
    if (radio_lunch.isChecked()) {
        primary = "Lunch";
    }
    if (radio_dinner.isChecked()) {
        primary = "Dinner";
    }

    if (radio_snack.isChecked()) {
        primary = "Snack";
    }
    if (radio_beverage.isChecked()) {
        secondary = "Beverage";
    }
    return new MealTime(primary, secondary, selectedDate);
}
```

	P	P 17	Produce a bug tracking report	✔

User must be able to update their location using the search box.	Failed	Added code to prevent return keypress from submitting entire page as form.	Passed
Application must respond appropriately when user enters unfindable location in search box.	Failed	Added check on location to ensure it exists, and displayed an error alert on unfindable location input.	Passed
User must be able to to create a walking route between venues by holding the Ctrl key while clicking venues on the map.	Failed	Added global cross-platform onkeydown/onkeyup handlers to detect Ctrl keypress.	Passed
User must be able to create a walking (not driving/public transport/etc.) route between venues	Failed	Added travelMode: 'WALKING' to MapWrapper.prototype.showRoute method. Google's map route default is driving.	Passed
Application should correctly retrieve a list of facilities available at each venue from the database.	Failed	Corrected seeds file to populate the database with 'facilities', rather than 'facilites'.	Passed
User must be able to clear a previously created walking route	Failed	First attempt tried to modify the marker collection while looping over it. Corrected this using a while loop to monitor the collection's length and remove an element on each iteration.	Passed