

**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.




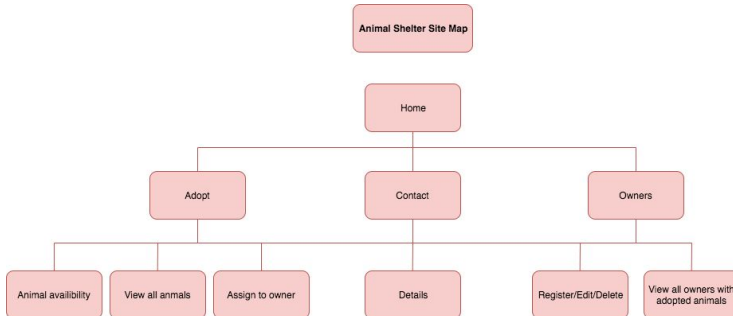
	Unit	Ref.	Evidence	Done
Week 2	I & T	I.T 5	<p>Demonstrate the use of an array in a program. Take screenshots of:</p> <p>*An array in a program</p> <pre> 1 require 'minitest/autorun' 2 require_relative '../array' 3 4 class TestArray < MiniTest::Test 5 6 def setup 7 @numbers = [3, 6, 9, 12, 15, 18, 21] 8 end 9 10 def test_all_numbers_in_array 11 12 assert_equal([3, 6, 9, 12, 15, 18, 21], @numbers) 13 end 14 end </pre> <p>*A function that uses the array</p> <pre> 1 def all_numbers_in_array 2 return numbers 3 end </pre> <p>*The result of the function running</p> <pre> 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 </pre>	✓

	I & T	I.T 6	<p>Demonstrate the use of a hash in a program. Take screenshots of:</p> <p>*A hash in a program</p> <pre data-bbox="539 331 1061 931"> @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, },], } </pre> <p>*A function that uses the hash</p> <pre data-bbox="539 987 1158 1375"> 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 </pre> <p>*The result of the function running</p> <pre data-bbox="539 1431 1257 1621"> # 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) ✖ █ </pre>	✓
	I & T		<p>Static and Dynamic testing task A</p> <p>https://github.com/AlphaHydroxy/Professional-Development-Awards/tree/master/Section_A_Static_and_Dynamic_Testing</p>	✓

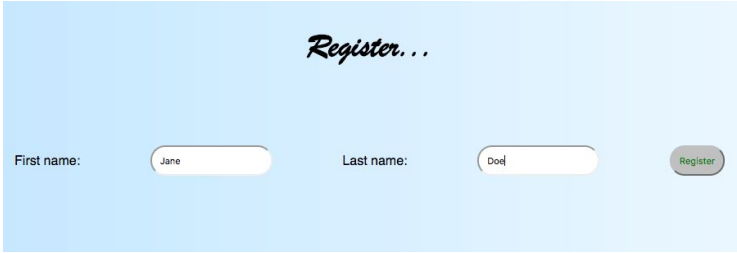

	Unit	Ref.	Evidence	Done
Week 3	I & T	I.T 3	<p>Demonstrate searching data in a program. Take screenshots of:</p> <p>*Function that searches data</p> <pre>require_relative '../db/sql_runner' class Animal attr_accessor :id, :name, :picture, :type, :breed, :age, : date_added def initialize (options) @id = 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 end</pre> <p>*The result of the function running</p> <pre>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)</pre>	✓
	I & T	I.T 4	<p>Demonstrate sorting data in a program. Take screenshots of:</p> <p>*Function that sorts data</p> <pre>def sort_by_date() sql = "SELECT date_added FROM animals WHERE animals.date;" SqlRunner.run(sql) end</pre> <p>*The result of the function running</p> <pre>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)</pre>	✓


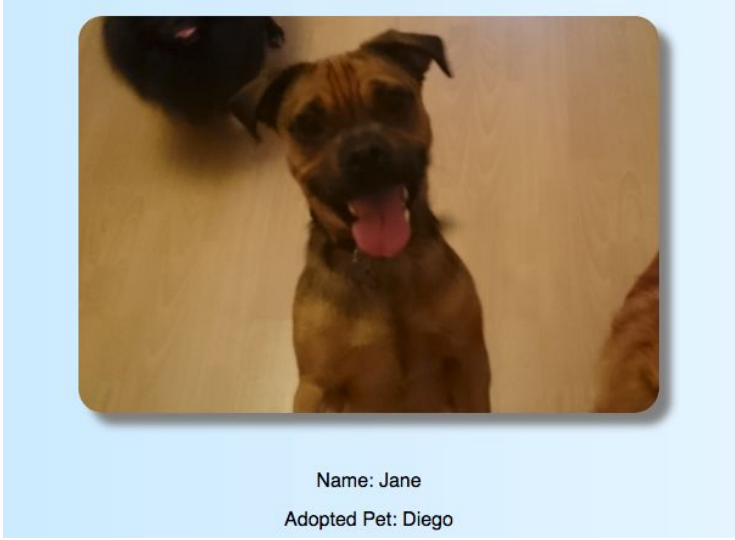

Week 5	Unit	Ref.	Evidence	Done
	A & D	A.D 1	<p>A Use Case Diagram</p>	✓
	A & D	A.D 2	<p>A Class diagram.</p>	✓
	A & D	A.D 3	<p>An Object diagram.</p>	✓

	A & D	A.D 4	<p>An Activity Diagram</p> <pre> graph TD subgraph "Activity Diagram Animal Shelter" direction TB subgraph "User" Start(()) --> Visit[Visit website] Visit --> RequestInfo[Request animal information] RequestInfo --> RequestShelter[Request shelter details] RequestShelter --> Register[Register as new user] Register --> End((())) end subgraph "Animal Shelter" DisplayHome[Display homepage] DisplayInfo[Display animal information] DisplayShelter[Display shelter details] SaveDB[Save user details to database] ConfirmReg[Confirm registration] end Visit --> DisplayHome DisplayHome --> RequestInfo RequestInfo --> DisplayInfo DisplayInfo --> RequestShelter RequestShelter --> DisplayShelter DisplayShelter --> Register Register --> SaveDB SaveDB --> ConfirmReg ConfirmReg --> End end </pre> <p>The diagram illustrates the process of interacting with an animal shelter website. It features two lifelines: User and Animal Shelter. The process begins with the User visiting the website, which triggers the Animal Shelter to display the homepage. From the homepage, the User can request animal information, leading the Animal Shelter to display animal information. Similarly, the User can request shelter details, leading the Animal Shelter to display shelter details. The User can also register as a new user, which triggers the Animal Shelter to save the user details to the database and then confirm the registration. The process ends with the User's registration confirmed.</p>	✓
--	-------	-------	--	---

A & D	A.D 6	<p>Produce an Implementations Constraints plan detailing the following factors:</p> <ul style="list-style-type: none">*Hardware and software platforms*Performance requirements*Persistent storage and transactions*Usability*Budgets*Time <p><u>Implementation Constraints Plan</u></p> <table><thead><tr><th>Constraint</th><th>Possible Effect of Constraint</th><th>Solution</th></tr></thead><tbody><tr><td>Hardware and software platforms</td><td>Certain features may not be compatible</td><td>Prepare for different browsers, screen sizes</td></tr><tr><td>Performance requirements</td><td>Experience longer load times</td><td>Ensure smaller files (e.g) are used</td></tr><tr><td>Persistent storage</td><td>Too many users querying the database at the same time</td><td>Improve/expand network</td></tr><tr><td>Usability</td><td>Limited usage due to personal restrictions</td><td>Use semantic mark up for screen readers, intuitive layouts</td></tr><tr><td>Budgets</td><td>No budget</td><td>-</td></tr><tr><td>Time limitations</td><td>6 days to complete project - may not have additional functions featured</td><td>-</td></tr></tbody></table>	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 ) 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	-	✓
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 ) 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	<p>Create a user sitemap.</p>  <pre>graph TD; A[Animal Shelter Site Map] --> B[Home]; B --> C[Adopt]; B --> D[Contact]; B --> E[Owners]; C --> F[Animal availability]; C --> G[View all animals]; C --> H[Assign to owner]; D --> I[Details]; E --> J[Register/Edit/Delete]; E --> K[View all owners with adopted animals];</pre>	✓																					

	P	P 6	<div>Produce two wireframe designs.</div> <div><div><div><div><div><div></div><div></div><div></div></div><div>Animal Shelter</div><div><div>Adopt</div><div>Owners</div><div>Contact</div></div></div><div><div><div>Heading</div><div>Subheading</div><div><div></div></div><div><p>Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis. Te facilisis mnesarchum qui posse omnium mediocritatem est cu. Modus argumentum ne qui tation efficiendi in eos. Ei mea falli legere efficiantur et tollit aliquip debitis mei. No deserunt mediocritatem mei. Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis. Te facilisis mnesarchum qui posse omnium mediocritatem est cu. Modus argumentum ne qui tation efficiendi in eos. Ei mea falli legere efficiantur et tollit aliquip debitis mei. Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis. Te facilisis mnesarchum qui posse omnium mediocritatem est cu. Modus argumentum ne qui tation efficiendi in eos. Ei mea falli legere efficiantur et tollit aliquip debitis mei. Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis.</p></div></div><div><div></div></div></div><div><div></div></div></div></div><div><div><div>Adopt</div><div>Owners</div><div>Contact</div></div><div><div></div><div></div><div></div></div><div><p>Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis. Te facilisis mnesarchum qui posse omnium mediocritatem est cu. Modus argumentum ne qui tation efficiendi in eos. Ei mea falli legere efficiantur et tollit aliquip debitis mei. No deserunt mediocritatem mei. Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis. Te facilisis mnesarchum qui posse omnium mediocritatem est cu. Modus argumentum ne qui tation efficiendi in eos. Ei mea falli legere efficiantur et tollit aliquip debitis mei. Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis.</p></div><div><div></div></div></div><div><div></div><div></div><div></div></div><div><p>Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis. Te facilisis mnesarchum qui posse omnium mediocritatem est cu. Modus argumentum ne qui tation efficiendi in eos. Ei mea falli legere efficiantur et tollit aliquip debitis mei. No deserunt mediocritatem mei. Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis. Te facilisis mnesarchum qui posse omnium mediocritatem est cu. Modus argumentum ne qui tation efficiendi in eos. Ei mea falli legere efficiantur et tollit aliquip debitis mei. Lorem ipsum dolor sit amet et delectus accommodare his consul copiosae legendos at vix ad putent delectus delicata usu. Vidit dissentiet eos cu eum an brute copiosae hendrerit. Eos erant dolorum an. Per facer affert ut. Mei isque mentitum moderatus cu. Sit munere facilis accusam eu dicat falli consuluatu at vis.</p></div><div><div></div></div></div> <div><div></div></div>
--	---	-----	--

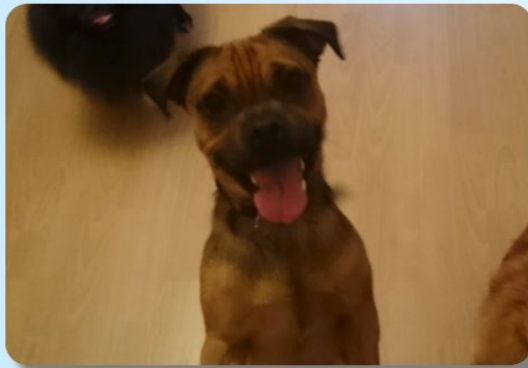
P	P 10	<p>Take a screenshot of an example of pseudocode for a function.</p> <pre> #write a method on Adoptions Class to create table entry for adoptions def save() #insert and return values into adoptions table sql = "INSERT INTO adoptions (owner_id, animal_id) VALUES ({@owner_id}, {@animal_id}) RETURNING *;" #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 </pre>	✓
P	P 13	<p>Show user input being processed according to design requirements. Take a screenshot of:</p> <ul style="list-style-type: none"> * The user inputting something into your program  <ul style="list-style-type: none"> * The user input being saved or used in some way 	✓

P	P 14	<p>Show an interaction with data persistence. Take a screenshot of:</p> <ul style="list-style-type: none"> * Data being inputted into your program  <ul style="list-style-type: none"> * Confirmation of the data being saved 	✓
P	P 15	<p>Show the correct output of results and feedback to user. Take a screenshot of:</p> <ul style="list-style-type: none"> * 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	<p>Demonstrate the use of Polymorphism in a program.</p> <pre> 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; } } </pre>	✓

	Unit	Ref.	Evidence	Done
Week 7	A & D	A.D 5	<p>An Inheritance Diagram</p> <pre> classDiagram class Vehicle { +start_engine() +drive() } class Car { +start_engine() +drive() } class Motorbike { +start_engine() +drive() } Vehicle < -- Car Vehicle < -- Motorbike </pre>	✓
	I & T	I.T 1	<p>Take a screenshot of an example of encapsulation in a program.</p> <pre> 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; } } </pre>	✓

	I & T	I.T 2	<p>Take a screenshot of the use of Inheritance in a program. Take screenshots of:</p> <p>*A Class</p> <pre> 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; } } </pre>	✓
--	-------	-------	--	---

*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.setSelectedMealTime(selectedMealTime);
eatFoodEvent.setUser(new User("Jia", "Female", 56));
AddMealActivity.this.allTheEaterFood.add(eatFoodEvent);
```

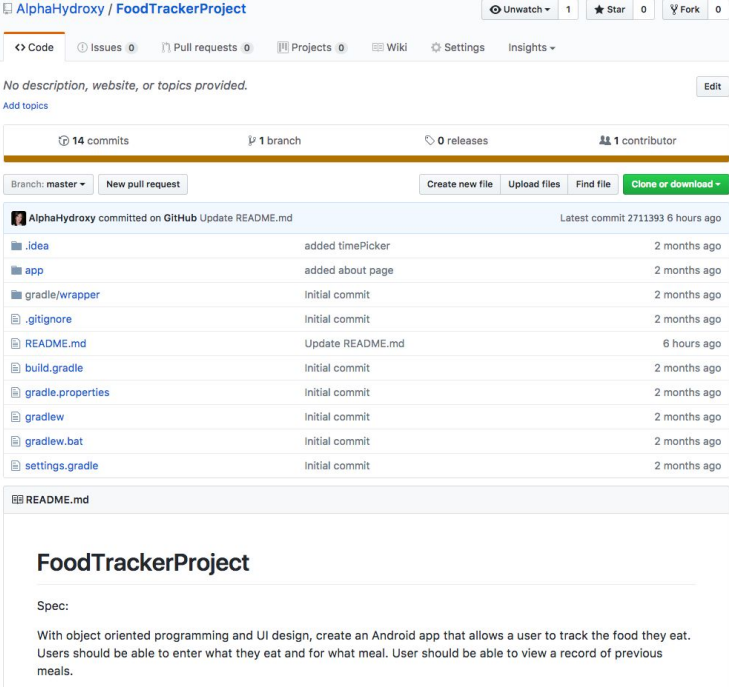
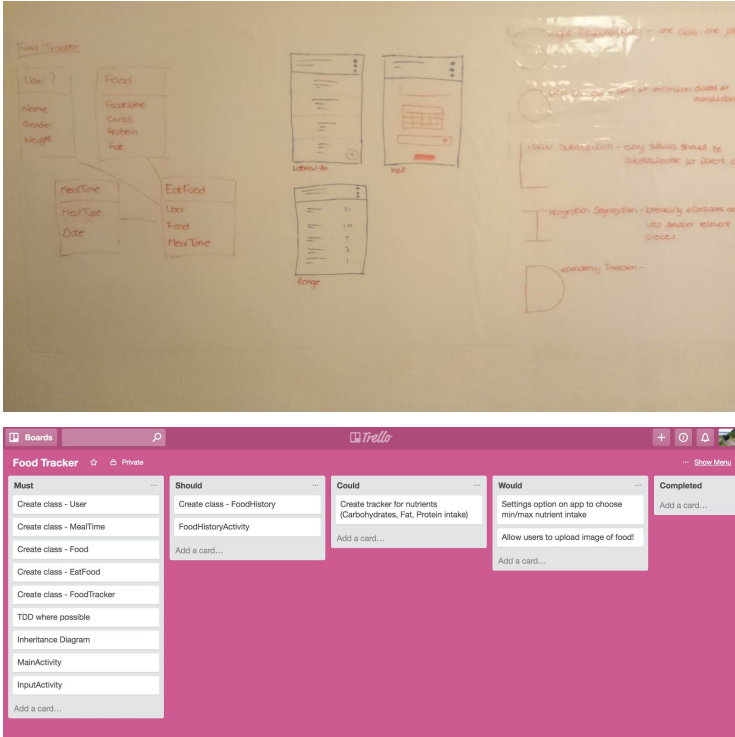
*A Method that uses the information inherited from another class.

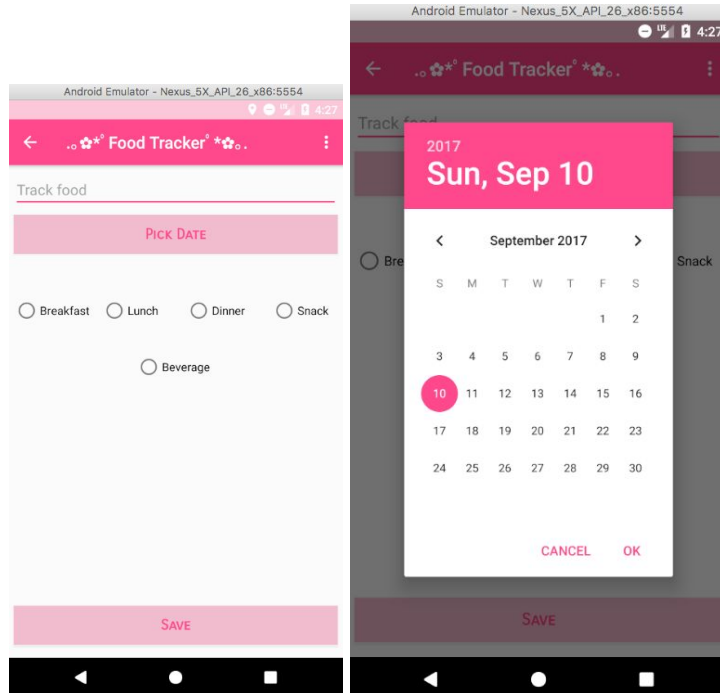
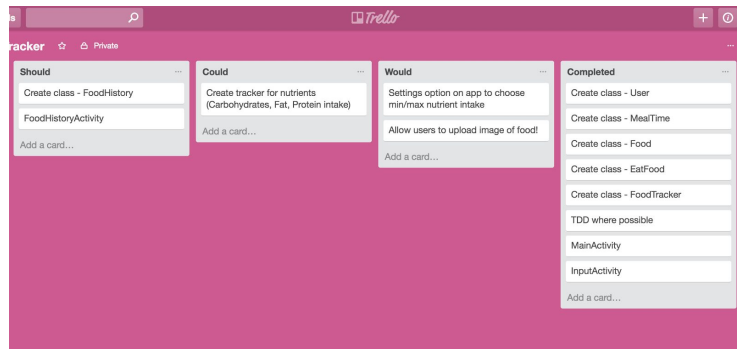
```
public String toString(){
    return "User: " + this.user.getName() + " | Weight: " + user.getWeight() + "kg" + "\nFood : "
        + food.getName() + "\nMeal Time: " + mealTime.getPrimary() + " " +
        mealTime.getSecondary() + "\n" + "Date : " + mealTime.getDate();
}

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);
}
```

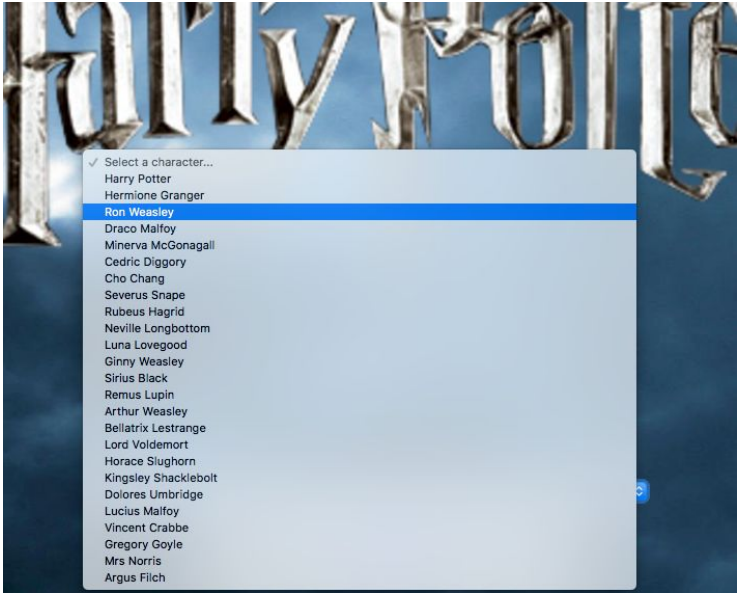
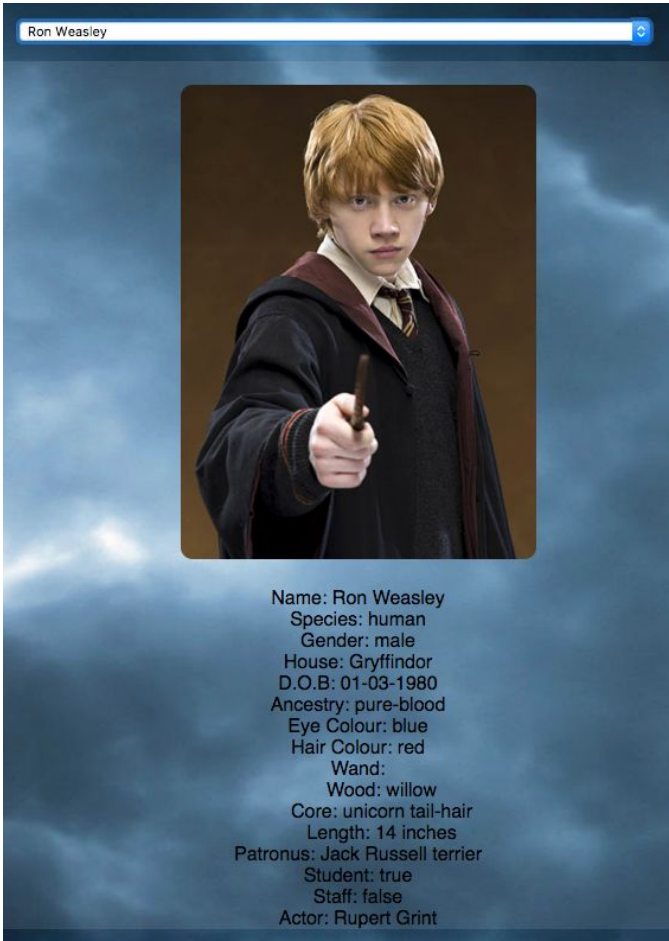
	P	P 11	<p>Take a screenshot of one of your projects where you have worked alone and attach the Github link.</p>  <p>https://github.com/AlphaHydroxy/FoodTrackerProject</p>	✓
	P	P 12	<p>Take screenshots or photos of your planning and the different stages of development to show changes.</p> 	✓

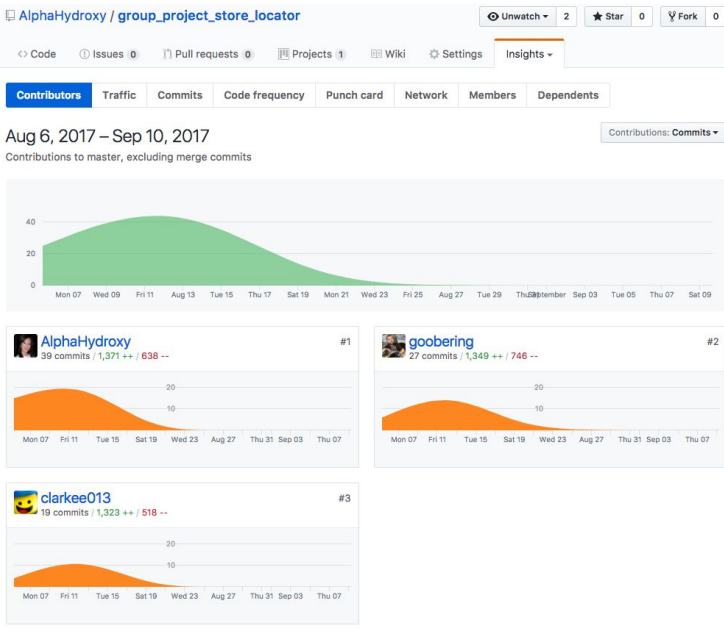
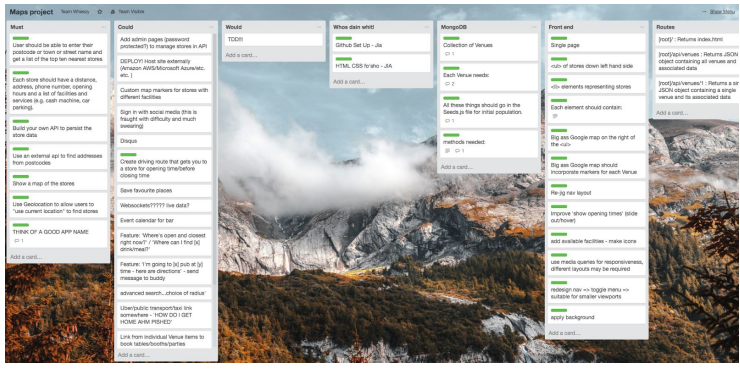


	Unit	Ref.	Evidence	Done
Week 10	P	P 18	<p>Demonstrate testing in your program. Take screenshots of:</p> <p>* Example of test code</p> <pre> public class UserTest { User user; @Before public void before() { user = new User("Jia", "female", 56); } @Test public void getUsersName(){ assertEquals("Jia", user.getName()); } @Test public void getUsersGender(){ assertEquals("female", user.getGender()); } @Test public void getUsersWeight(){ assertEquals(56, user.getWeight()); } } </pre> <p>* The test code failing to pass</p> <pre> 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()); } } </pre> <p>All 3 tests failed</p> <p>* Example of the test code once errors have been corrected</p>	✓

			<pre> 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()); } } </pre> <p>* The test code passing</p> 	
--	--	--	--	--

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

	P	P 16	<p>Show an API being used within your program. Take a screenshot of:</p> <ul style="list-style-type: none"> * The code that uses or implements the API <pre>var app = function() { var url = "http://hp-api.herokuapp.com/api/characters"; makeRequest(url, requestComplete); }</pre> <ul style="list-style-type: none"> * The API being used by the program whilst running  	✓
--	---	------	--	---

Week 13	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>Store Finder</p> <p>Global pub giant Withoutspoon, wants a pub finder that allows users to "Find your nearest store" for their website.</p> <p>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	P 3	<p>Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.</p> 	✓

P	P 4	Write an acceptance criteria and test plan.	✓																								
<table><thead><tr><th>Acceptance Criteria</th><th>Expected Result/Output</th><th>Pass /Fail</th></tr></thead><tbody><tr><td>User should be able to enter postcode/address get list of nearest pubs</td><td>List of pubs to show on homepage</td><td>Pass</td></tr><tr><td>Each pub should have distance, address, phone number, opening hours & other facilities</td><td>list view on homepage for the details & facilities should be seen</td><td>Pass</td></tr><tr><td>Facilities to show to load from internal API storage</td><td>The pubs and facilities should show correctly on the list on the list view of the webpage</td><td>Pass</td></tr><tr><td>External API to be used for address look up and populates to map view</td><td>Markers for pubs saved in internal API showing on map</td><td>Pass</td></tr><tr><td>Geolocation to allow user to see pubs around their location</td><td>Pubs shown should be within distance of the user</td><td>Pass</td></tr><tr><td>User should be able to click on pub markers to display walking route</td><td>Walking route displayed on map from current location</td><td>Pass</td></tr><tr><td>User should be able to save favourite pub</td><td>not implemented yet</td><td>Fail</td></tr></tbody></table>				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
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>Initial page load</p> <pre>sequenceDiagram actor User participant index as controller/index.js participant map as map/index.js participant venues as models/venues.js participant venues_ctrl as controller/venues.js participant Database participant GoogleMaps as Google Maps API User->>index: HTTP GET index.html activate index index->>map: getHtmlWrapper activate map map->>GoogleMaps: Get Google Map widget activate GoogleMaps GoogleMaps-->>map: Google Map widget deactivate GoogleMaps map->>venues_ctrl: mapWrapper.compassAutoCompleteWidget() activate venues_ctrl venues_ctrl->>GoogleMaps: Get Google Place AutoComplete widget activate GoogleMaps GoogleMaps-->>venues_ctrl: Google Place AutoComplete widget deactivate GoogleMaps venues_ctrl-->>index: clearMarkers() deactivate venues_ctrl index->>venues: addMarker(userPos) activate venues venues->>index: getCenter(userPos) deactivate venues index->>venues: webOpen(2) activate venues venues->>venues_ctrl: venues.getNearest(userPos) deactivate venues venues_ctrl->>venues: HTTP POST userPos activate venues venues->>venues_ctrl: venuesCtrlRange deactivate venues venues_ctrl->>venues: getDistanceRangeOfUserPos activate venues venues->>venues_ctrl: venuesCtrlRange deactivate venues venues_ctrl-->>index: venuesCtrlRange deactivate venues_ctrl deactivate index</pre> <p>User location changed using search box</p> <pre>sequenceDiagram actor User participant map as map/index.js participant GoogleMaps as Google Maps API participant venues as models/venues.js participant venues_ctrl as controller/venues.js participant Database User->>map: getAutoComplete().getPlace() activate map map->>GoogleMaps: getPlace() activate GoogleMaps GoogleMaps-->>map: place.geometry.location deactivate GoogleMaps map-->>User: place.geometry.location deactivate map User->>map: clearRoute() activate map map-->>User: clearRoute() deactivate map User->>map: clearRouteMarkers() activate map map-->>User: clearRouteMarkers() deactivate map User->>map: clearMarkers() activate map map-->>User: clearMarkers() deactivate map User->>map: getCenter(place.geometry.location) activate map map-->>User: getCenter(place.geometry.location) deactivate map User->>map: webOpen(2) activate map map-->>User: webOpen(2) deactivate map User->>map: addMarker(place.geometry.location) activate map map-->>User: addMarker(place.geometry.location) deactivate map User->>venues_ctrl: venues.getNearest(place.geometry.location) activate venues_ctrl venues_ctrl->>venues: HTTP POST place.geometry.location activate venues venues->>venues_ctrl: venuesCtrlRange deactivate venues venues_ctrl->>venues: getDistanceRangeOfPlace.geometry.location activate venues venues->>venues_ctrl: venuesCtrlRange deactivate venues venues_ctrl-->>Database: venuesCtrlRange deactivate venues_ctrl</pre>																											

	P	P 8	<p>Produce two object diagrams.</p> <pre> classDiagram class FoodTracker { +EatFood } class EatFood { +ArrayList<EatFood> } class User { +Name: Jia +Gender: Female +Weight: 56kg } class Mealtype { +Primary: Breakfast +Secondary: Beverage +Date: 22-07-17 } class Food { +FoodName: Cereal +Carbs: 7g +Protein: 2g } FoodTracker --> EatFood EatFood --> User EatFood --> Mealtype EatFood --> Food class PubFinder { +User } class User2 { +Geolocation: Glasgow +API: Client side API } class Pub { +Name: Counting House +Address: St Vincent Pl +Town: Glasgow +Region: Lanarkshire +Postcode: G1 0GG +Phone: 0141 123 4567 +Facilities: wifi, room hire +Opening times: Mon 9-1 } PubFinder --> User2 User2 --> Pub </pre>	✓

	P	P 9	<p>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.</p> <div data-bbox="528 371 1265 636"> <pre>public void populate() { for (Suit suit : Suit.values()) { for (Rank rank : Rank.values()) { cards.add(new Card(rank, suit)); } } }</pre> <p>Looping through enums in Suit and enums in Rank and assigning each possible pair to an ArrayList of cards</p> </div> <div data-bbox="528 651 1265 976"> <pre>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; }</pre> <p>Loop through array of words and increment by one each word is repeated</p> </div> <div data-bbox="528 992 1265 1612"> <pre>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); }</pre> <p>if radio button is checked, assign string to variable</p> </div>	✓
--	---	-----	---	---

P	P 17	Produce a bug tracking report	✓																								
		<table><tr><td>User must be able to update their location using the search box.</td><td>Failed</td><td>Added code to prevent return keypress from submitting entire page as form.</td><td>Passed</td></tr><tr><td>Application must respond appropriately when user enters unfindable location in search box.</td><td>Failed</td><td>Added check on location to ensure it exists, and displayed an error alert on unfindable location input.</td><td>Passed</td></tr><tr><td>User must be able to create a walking route between venues by holding the Ctrl key while clicking venues on the map.</td><td>Failed</td><td>Added global cross-platform onkeydown/onkeyup handlers to detect Ctrl keypress.</td><td>Passed</td></tr><tr><td>User must be able to create a walking (not driving/public transport/etc.) route between venues</td><td>Failed</td><td>Added travelMode: 'WALKING' to MapWrapper.prototype.showRoute method. Google's map route default is driving.</td><td>Passed</td></tr><tr><td>Application should correctly retrieve a list of facilities available at each venue from the database.</td><td>Failed</td><td>Corrected seeds file to populate the database with 'facilities', rather than 'facilites'.</td><td>Passed</td></tr><tr><td>User must be able to clear a previously created walking route</td><td>Failed</td><td>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.</td><td>Passed</td></tr></table>	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 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	
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 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																								