Evidence Gathering Document for SQA Level 8 Professional Developer Award.

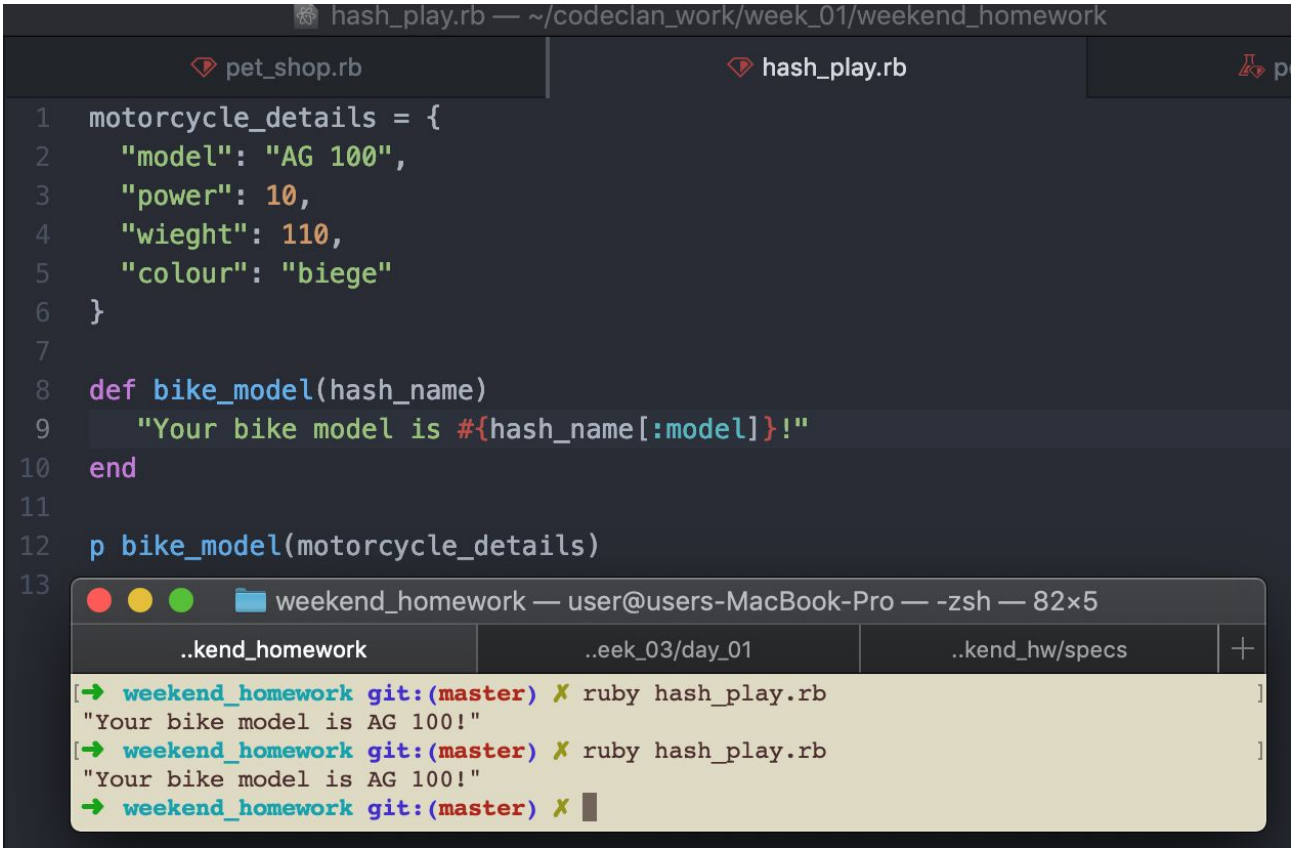
This document is designed for you to present your screenshots and diagrams relevant to the PDA and to also give a short description of what you are showing to clarify understanding for the assessor.

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	

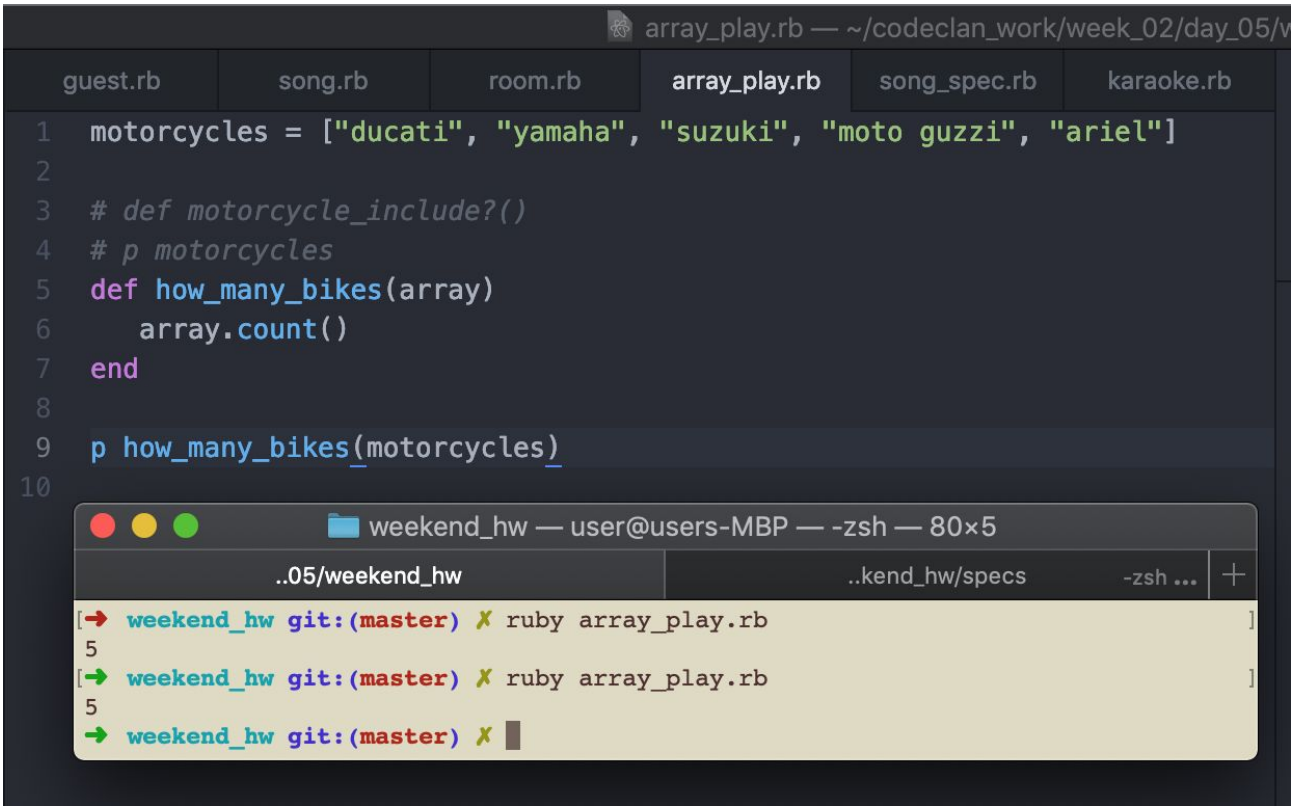


Here is an example of a hash being used to store the details of a motorcycle. The function is bike_model() is then called to print what is the model name.

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	

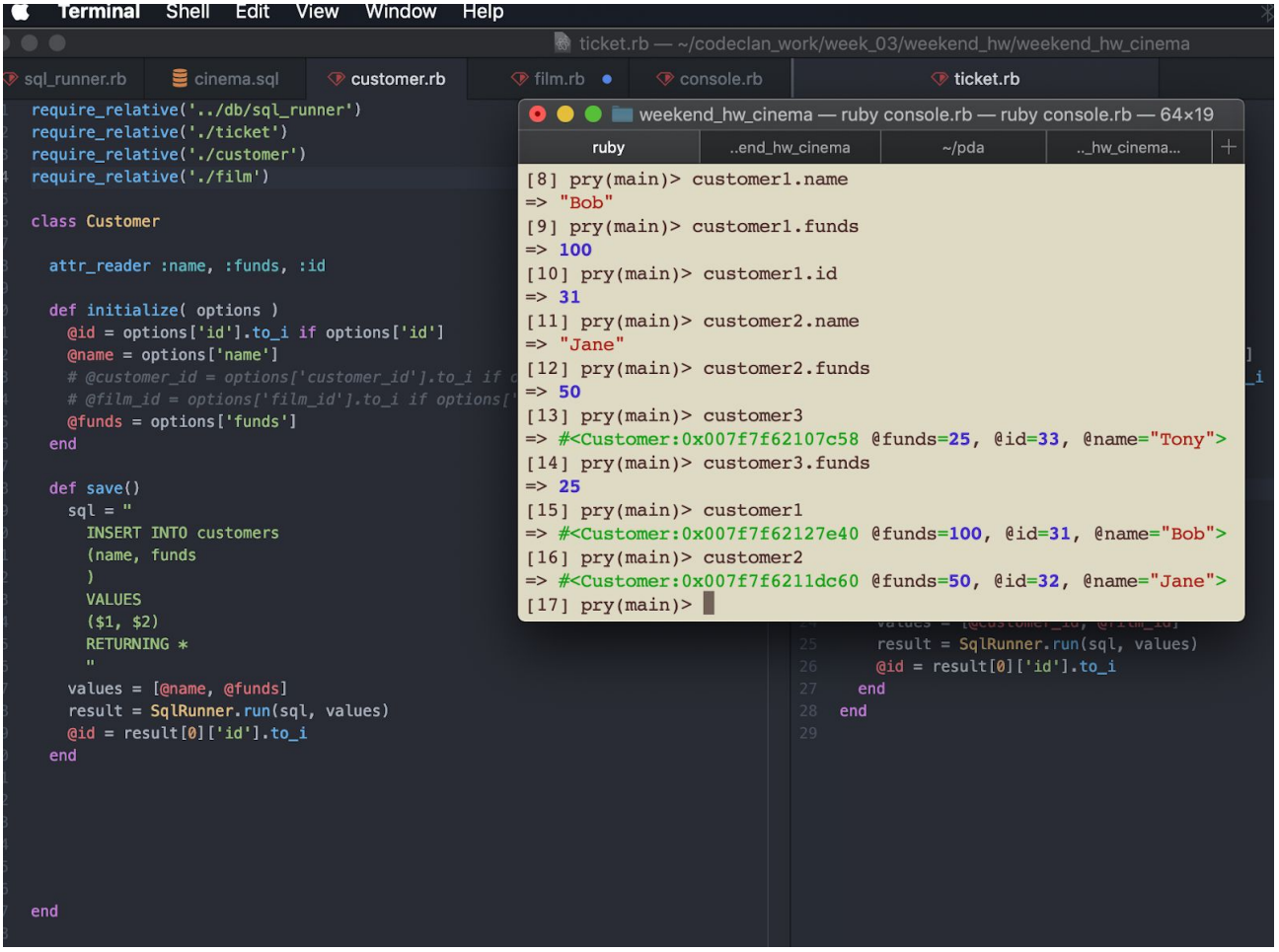
Here is an array of motorcycles, and the function how_many_bikes() returns the number of bikes



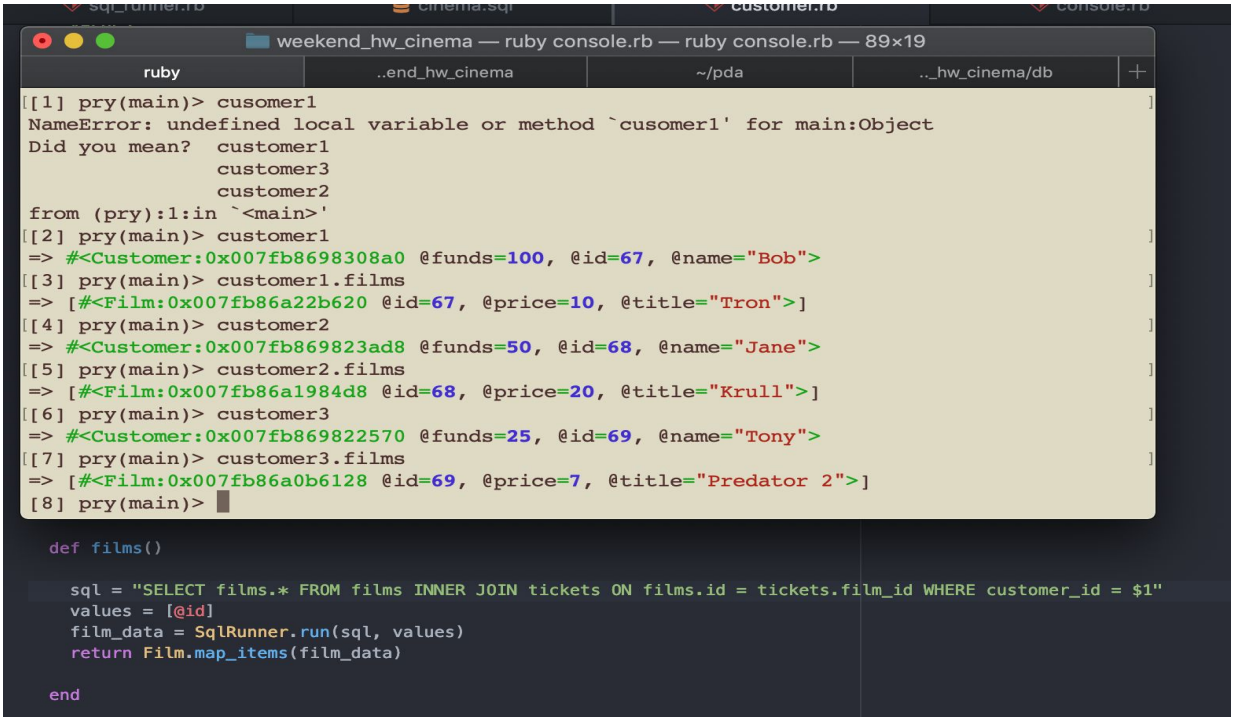
Week 3

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	

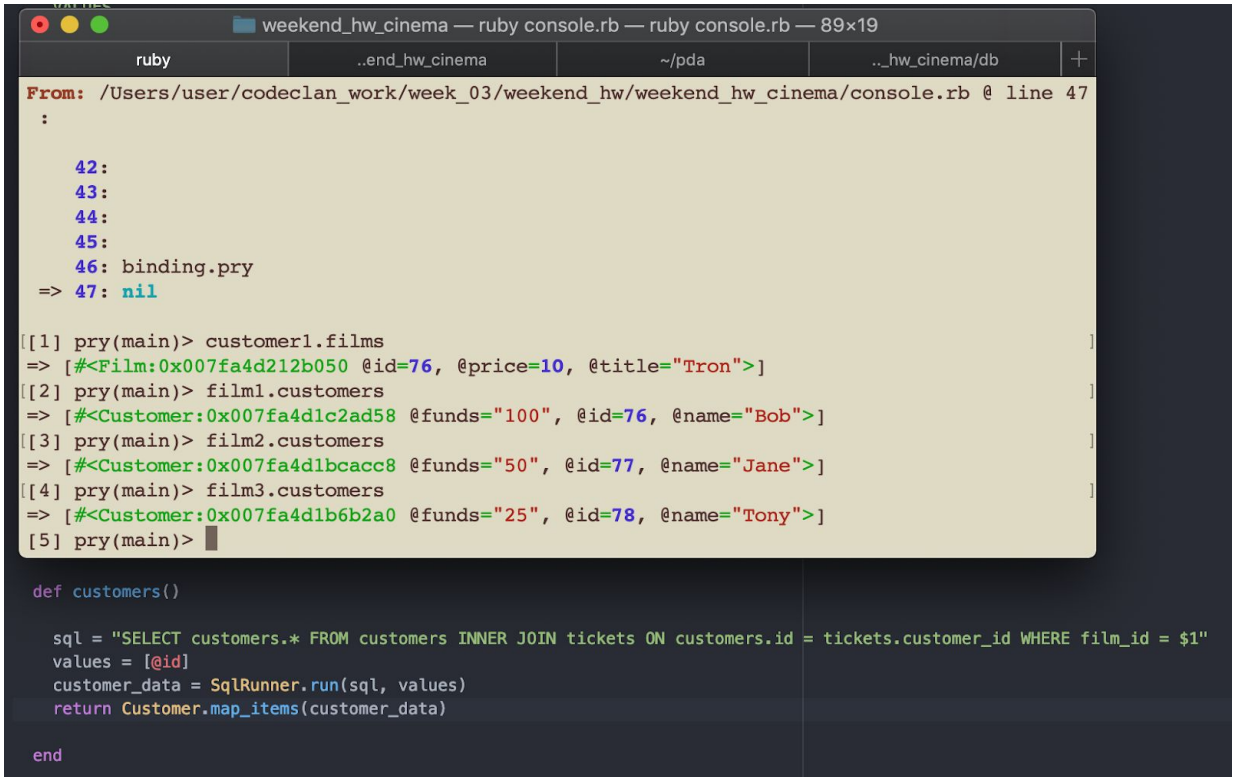
Here is a function running that can return data when called. Here we can find the customer name, id and funds individually by calling .name, .funds or .id respectively. Alternatively all the customers data can be displayed by simply calling customer1 etc.



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	

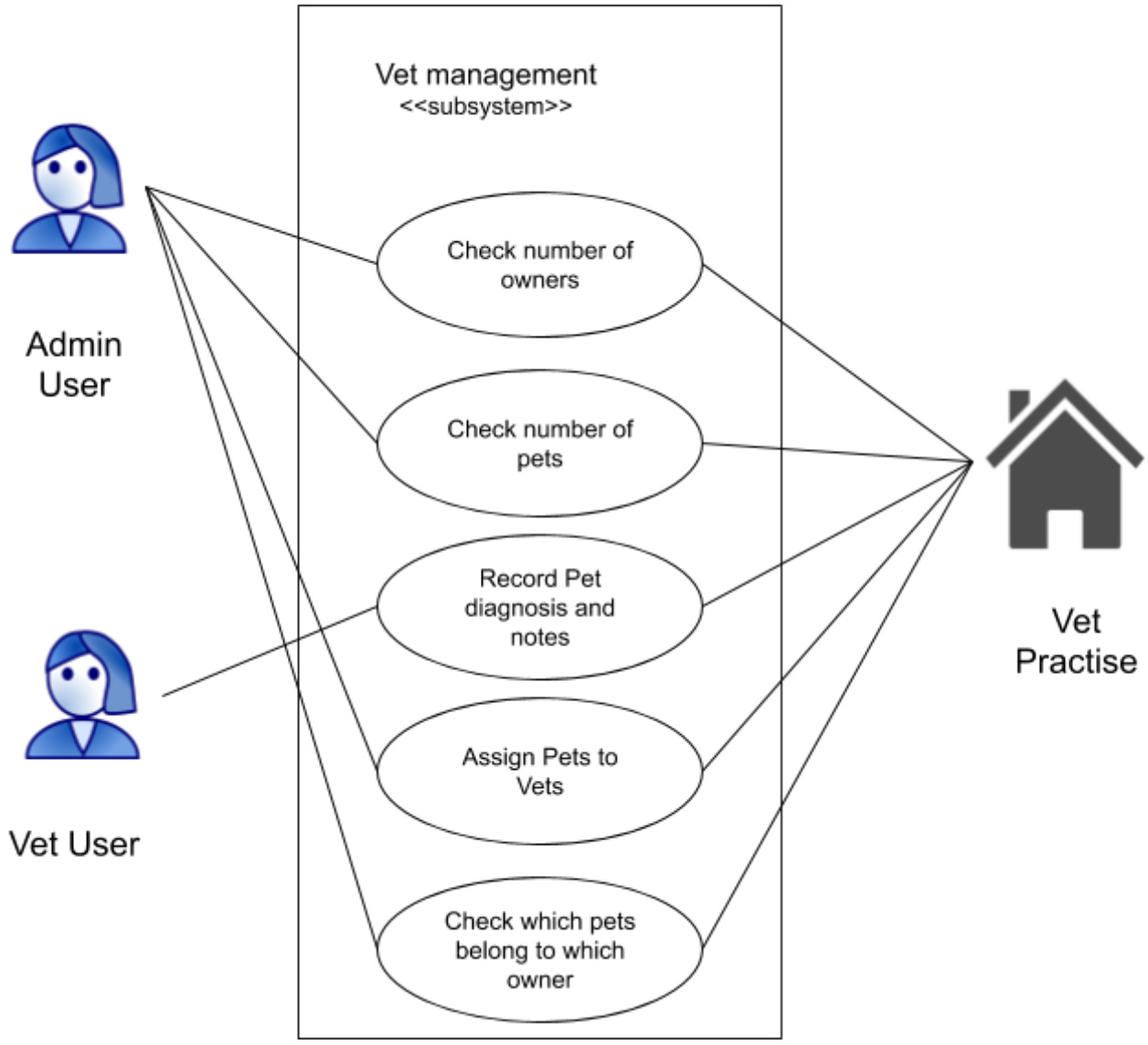


Here data can be sorted by finding the films a particular customer has been to see. Below is the opposite sort, here we can see all customers who have seen a particular film

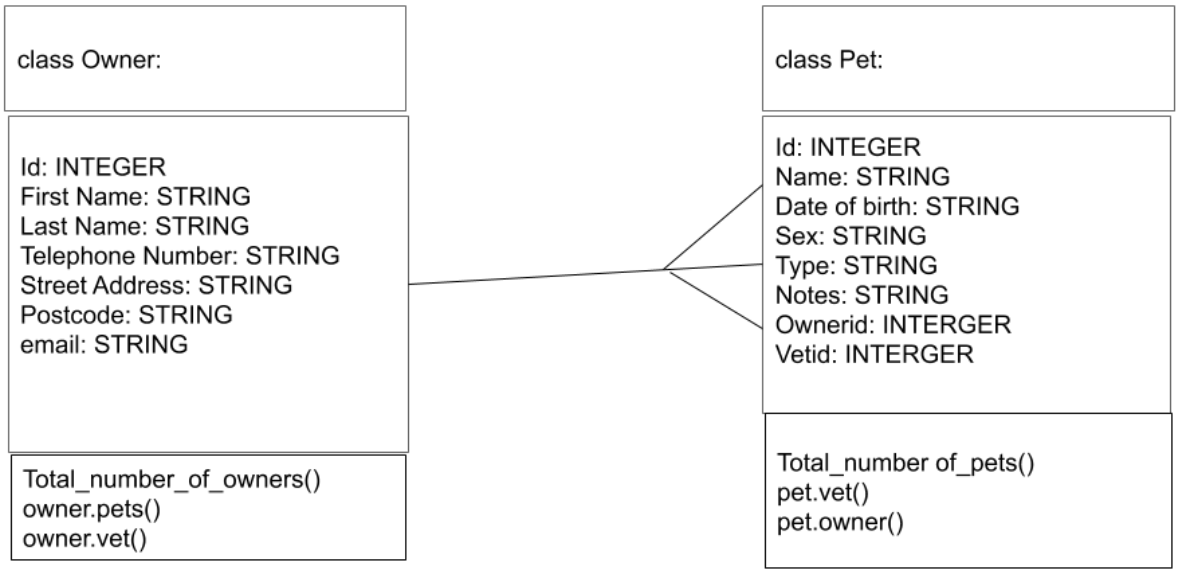


Week 4

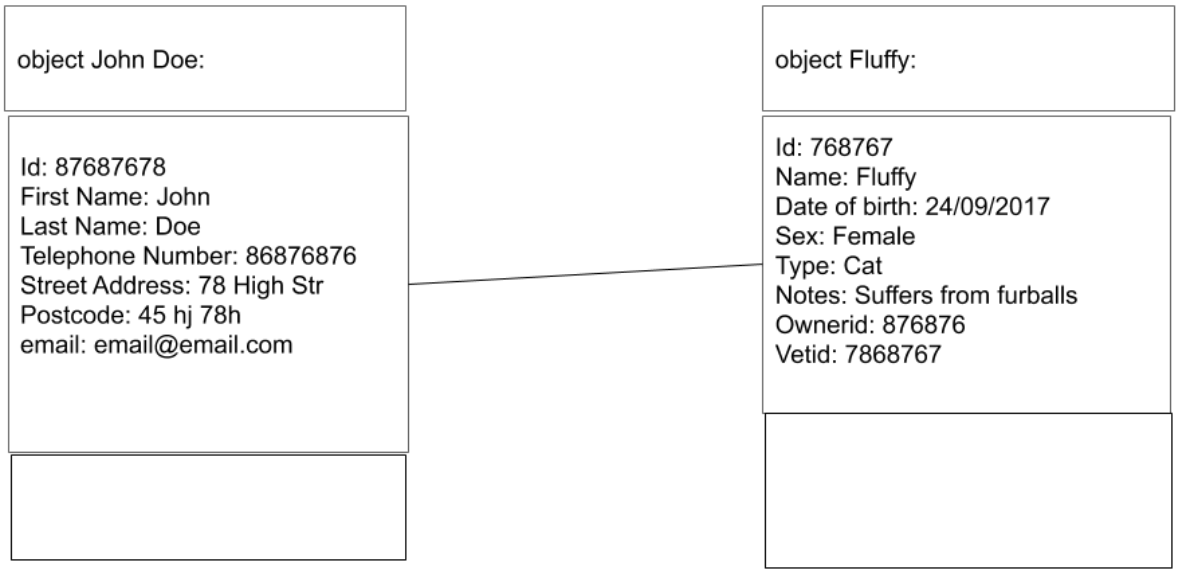
Unit	Ref	Evidence	
A&D	A.D.1	A Use Case Diagram	



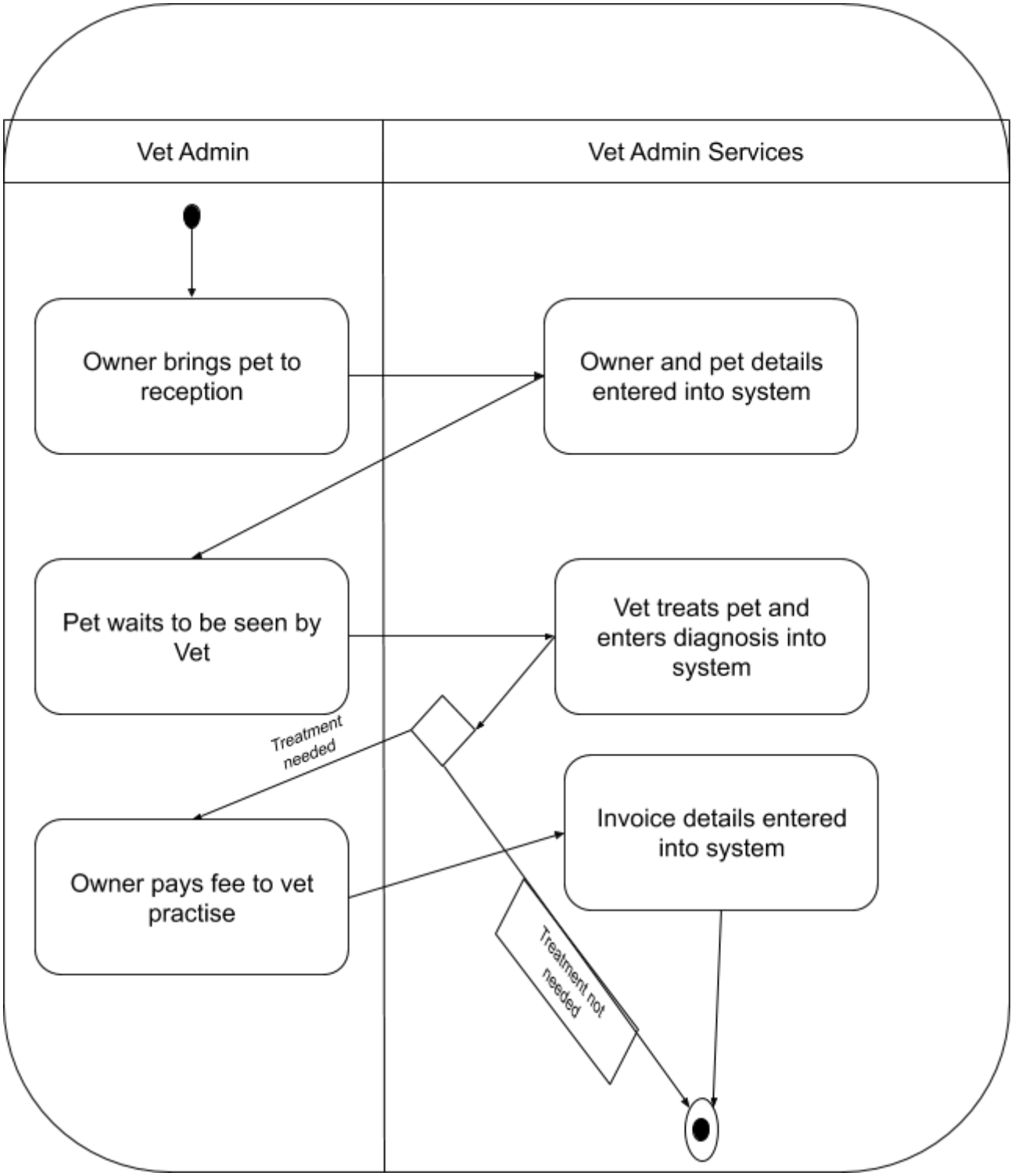
Unit	Ref	Evidence	
A&D	A.D.2	A Class Diagram	



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



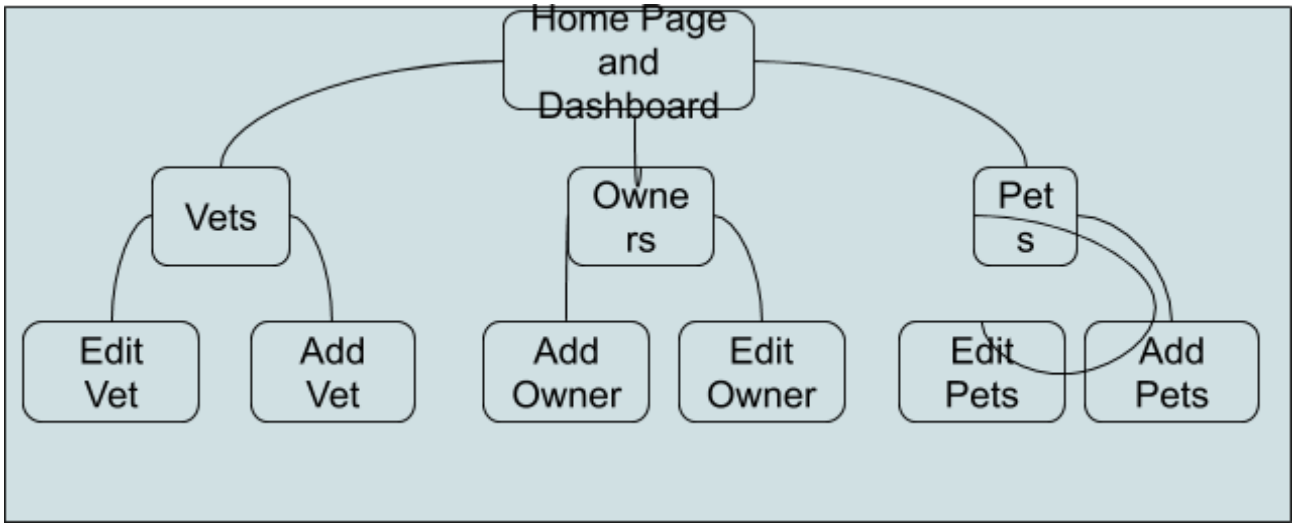
Unit	Ref	Evidence	
A&D	A.D.4	An Activity Diagram	



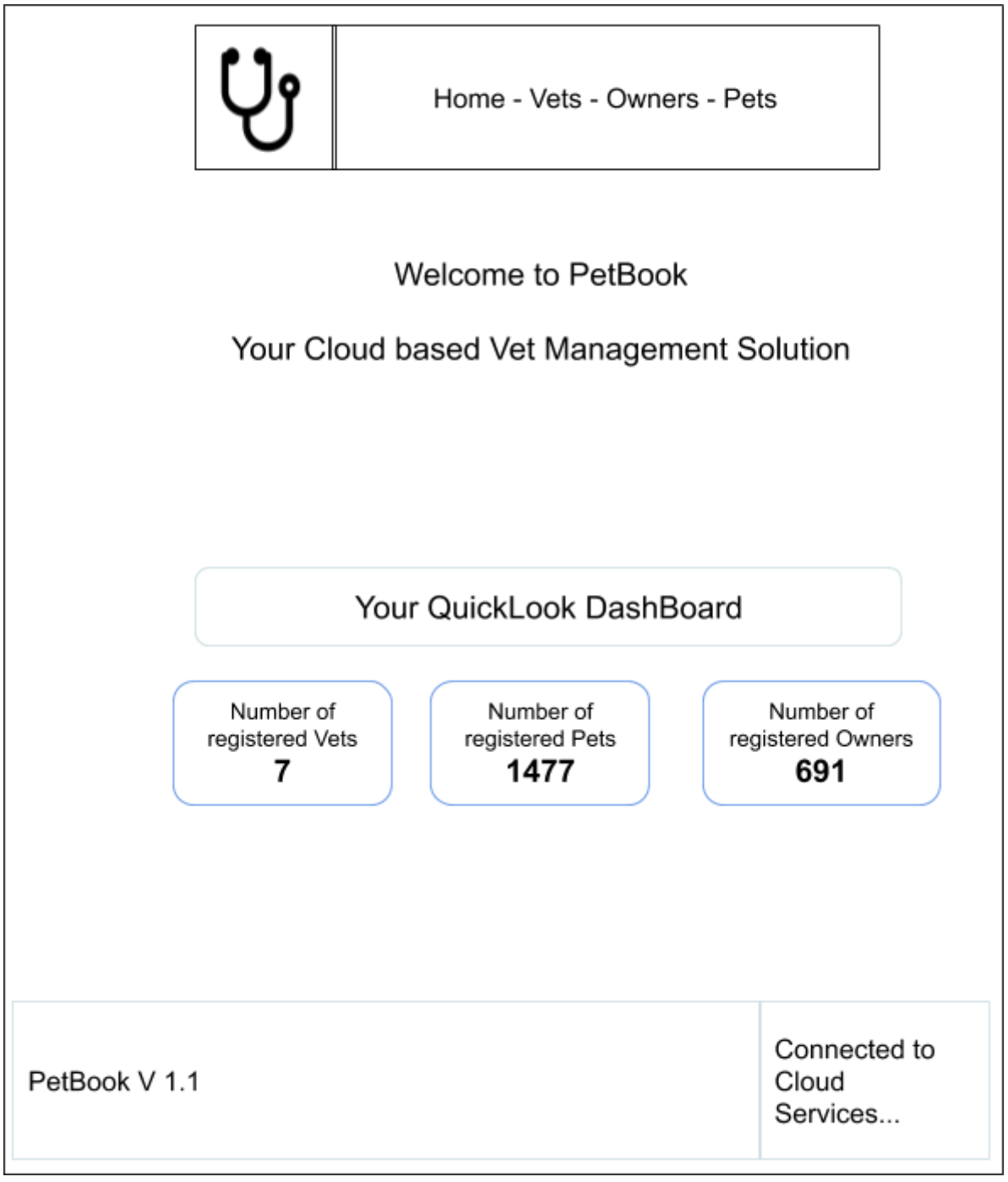
Unit	Ref	Evidence	
A&D	A.D.6	Produce an Implementations Constraints plan detailing the following factors: *Hardware and software platforms *Performance requirements *Persistent storage and transactions *Usability *Budgets *Time	

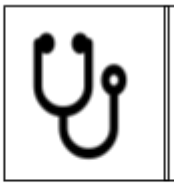
Constraint Category	**Implementation Constraint**	**Solution**
Hardware and Software Platforms	Not enough memory to run the desired software	Increase the available ram
Performance Requirements	Fast enough to server x number of concurrent users	Purchase extra system resources
Persistent Storage and Transactions	The current size and type of the storage solution	Increase the size and upgrade the type of storage solution ie Cloud Based
Usability	Difficult for user to efficiently use the system	Streamline the UX to improve workflow
Budgets	Limited funding prevents additional features and functionality being added	Secure additional funding resources
Time Limitations	The product has to be delivered to the client in a certain date	Increase the amount of personal working on the product to speed up delivery times.

Unit	Ref	Evidence	
P	P.5	User Site Map	



Unit	Ref	Evidence	
P	P.6	2 Wireframe Diagrams	





Home - Vets - Owners - Pets

Owners First Name:

Owners Last Name:

Owners Telephone:

Owners Address:

Owners Postcode:

Add New Owner

Week 5

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

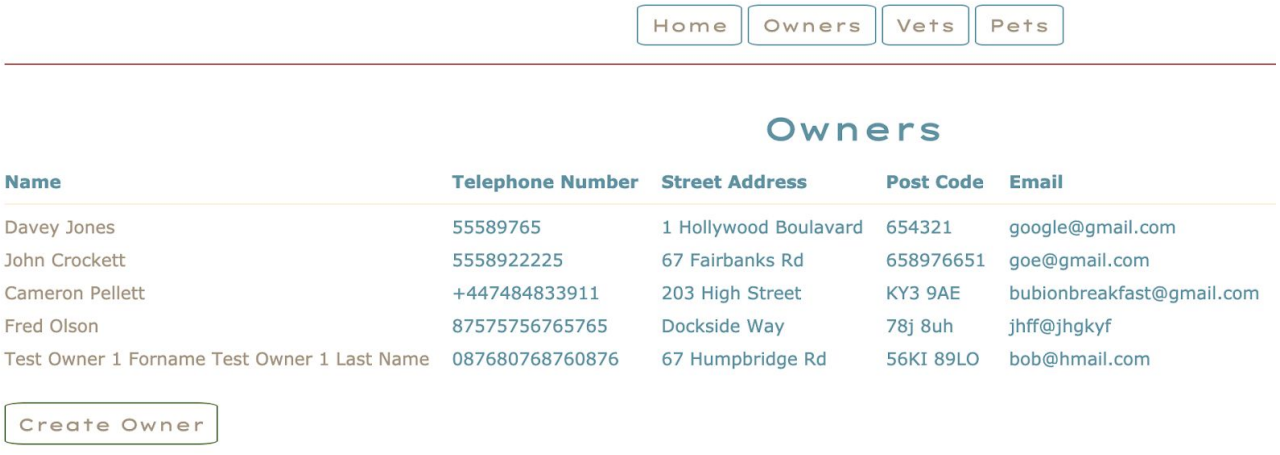
```
#Concatinate the first and last names of the vets
#Use string interploation with the instance variables for the first and last names
#Return or output the the concatinated string.
  def pretty_name()
    return "#{@first_name} #{@last_name}"
  end
```

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	

The current owner list

A new owner being added

The updated owners list with the new owner appearing

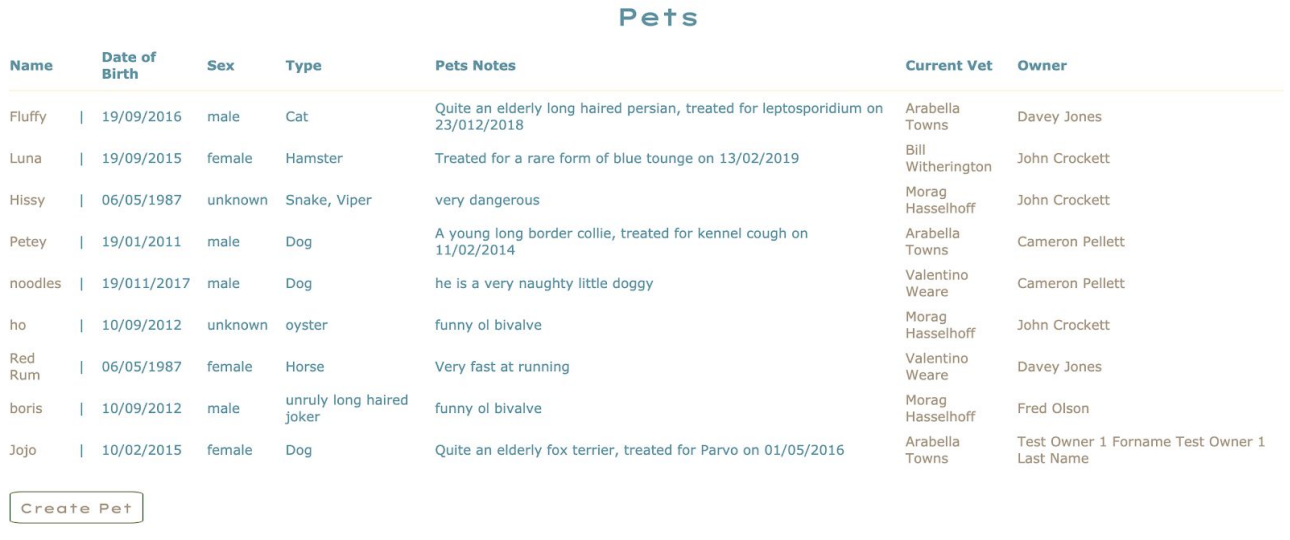


Unit	Ref	Evidence	
P	P.14	Show an interaction with data persistence. Take a screenshot of: * Data being inputted into your program * Confirmation of the data being saved	

Cameron Pellett	+447484833911	203 High Street	KY3 9AE	bubionbreakfast@gmail.com	Petey noodles
Fred Olson	87575756765765	Dockside Way	78j 8uh	jhff@jhgkyf	boris
Test Owner 1 Forname Test Owner 1 Last Name	087680768760876	67 Humpbridge Rd	56KI 89LO	bob@hmail.com	

Test owner 1 has not yet been assigned to a pet.

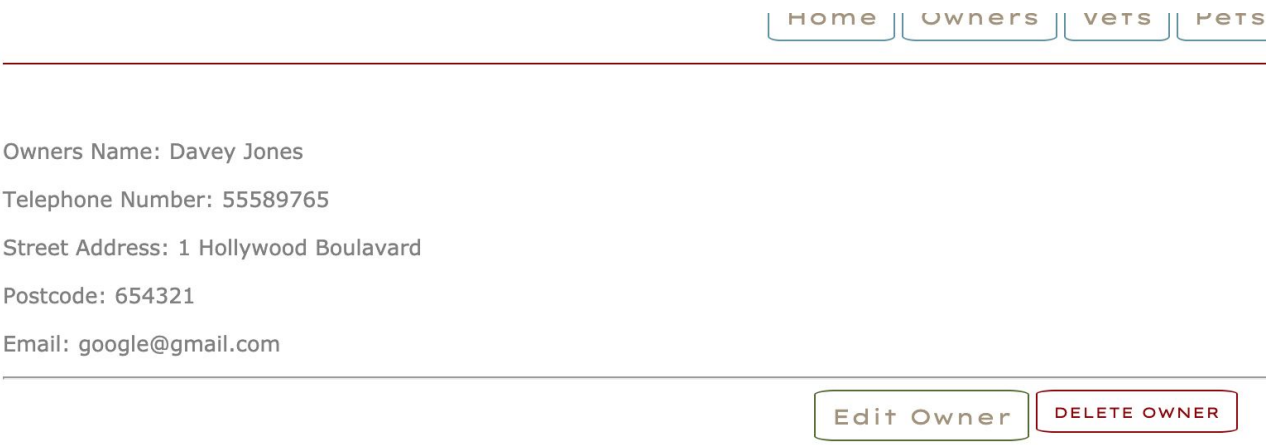
Test owner has been assigned to a pet



Confirmation that the owner is now assigned to the pet.

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	

The user can edit the owners phone number



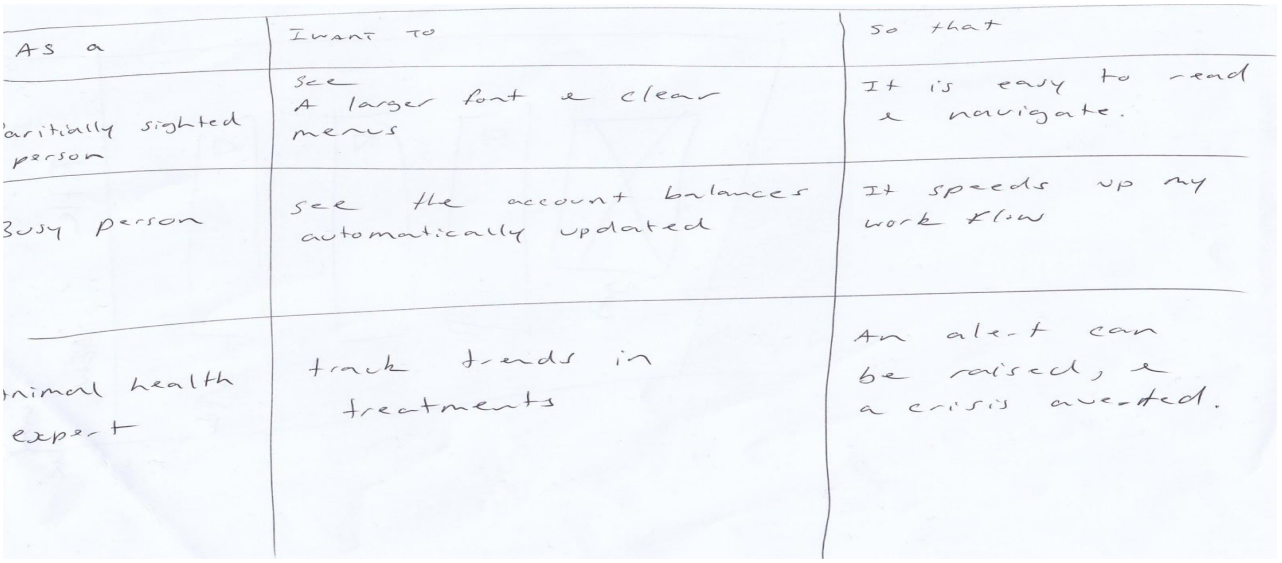
The owners phone number is now edited

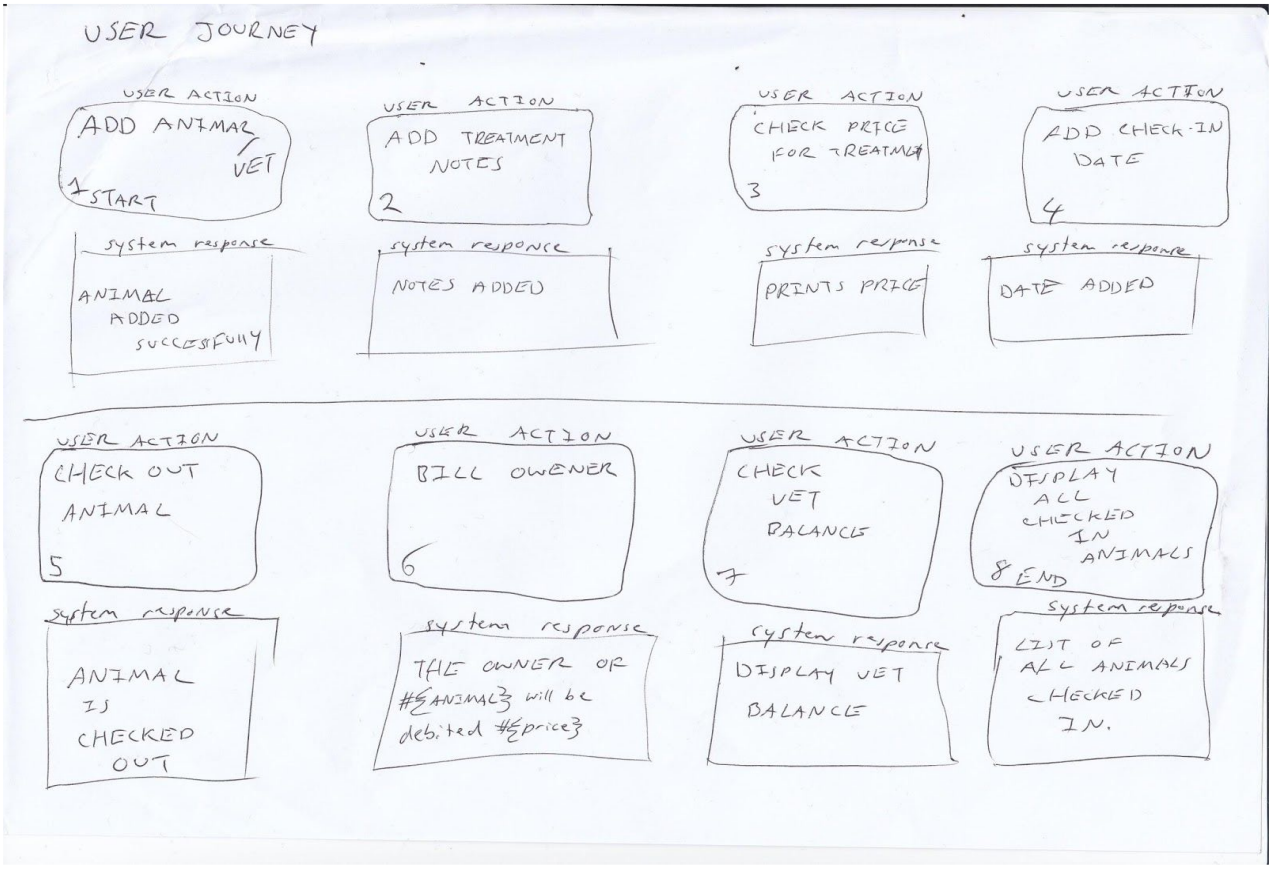
Unit	Ref	Evidence	
P	P.11	Take a screenshot of one of your projects where you have worked alone and attach the Github link.	

Vet Management App A veterinary practice has approached you to build a web application to help them manage their animals and vets. A vet may look after many animals at a time. An animal is registered with only one vet. This App is built using an Object Oriented Model. It has Classes that define individual instances of each object. This App is written in: Ruby HTML CSS Uses RESTful Routes with Sinatra

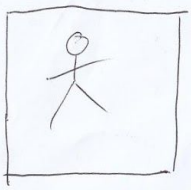
https://github.com/bubionbreakfast/vet_app

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





PROTO - PERSONA



NAME:

JANE DOE

BEHAVIOURS

QUALITY ~~CARE~~ INFORMATION
EASE of USE
FAST
ACCURATE INFORMATION

DEMOGRAPHICS

40 YEAR OLD
FEMALE
~~WITH 2 x CATS~~
COMPUTER LITERATE

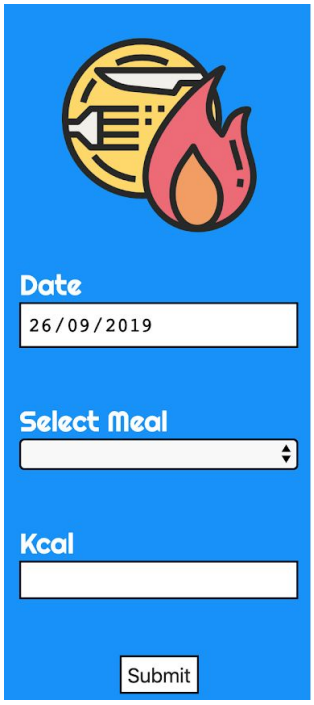
NEEDS & GOALS

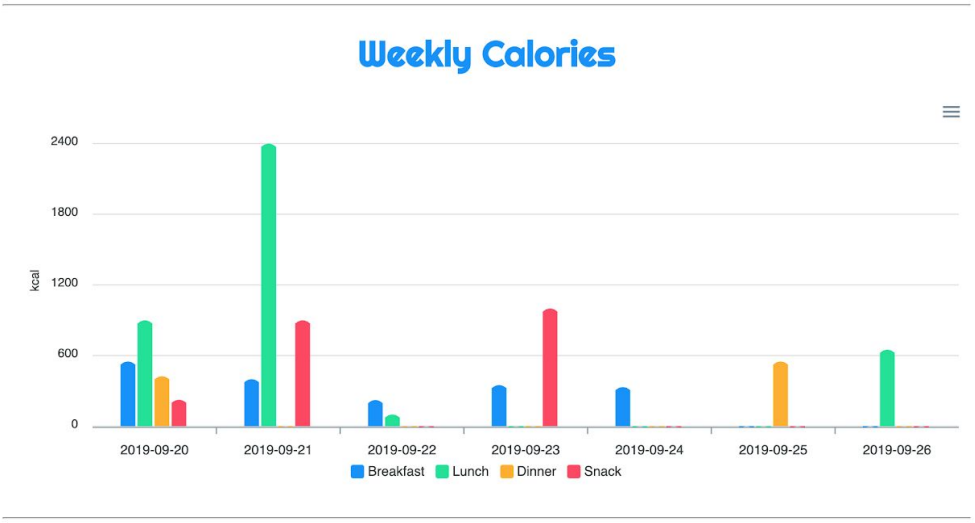
GET ~~ACCOUNT~~ ACCOUNT BALANCE
GET WHICH VET IS CARING FOR WHICH PET
RAISE ALERT FOR CERTAIN TREATMENTS.
NUMBER OF ANIMALS IN CARE

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	

```
MongoClient.connect('mongodb://localhost:27017')
  .then((client) => {
    const db = client.db('habitTracker');
    const mealsCollection = db.collection('meals');
    const mealsRouter = createRouter(mealsCollection);
    app.use('/api/meals', mealsRouter);
  })
  .catch(console.err);
```





Week 8

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

Habit Tracker

Nowadays everyone is trying to build or break a habit. But it's tricky to keep track of them. Identify a habit you'd like to help someone break or build (e.g. alcohol consumption, smoking, calories, exercise, healthy eating...) and make an app to help.

MVP

A user should be able to:

- Make CRUD entries on the front-end that are persisted on a MongoDB database on the back-end
- Display the data in visually interesting / insightful ways.

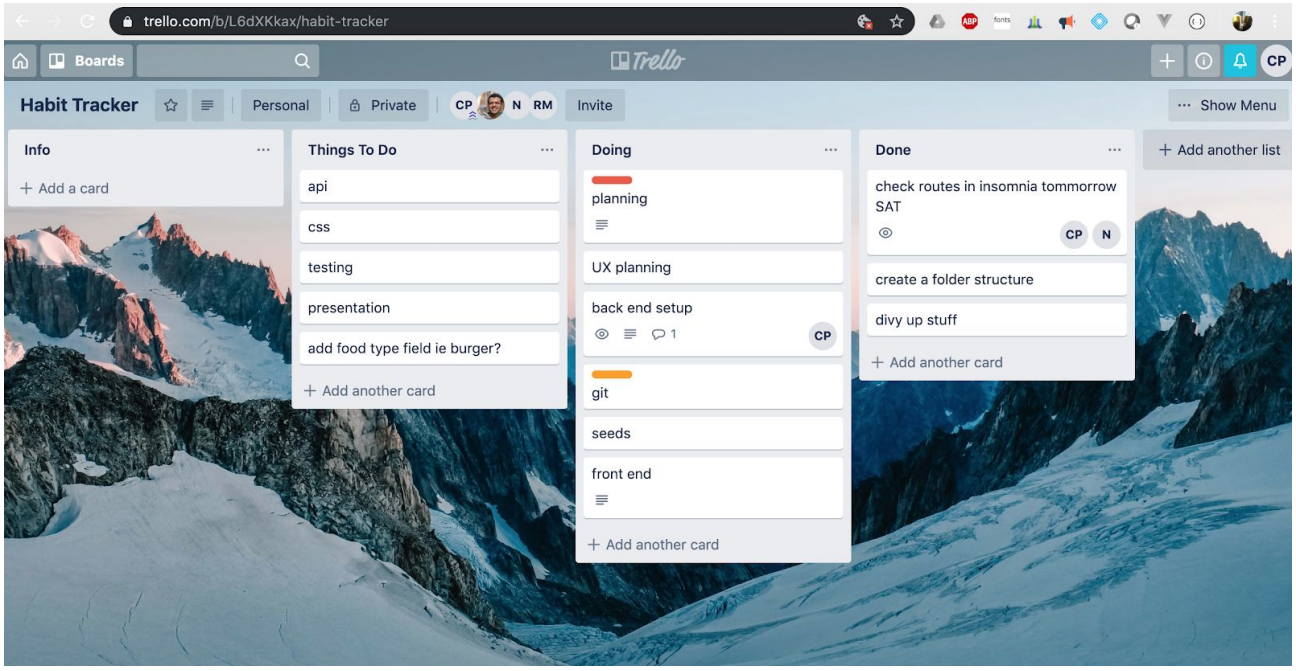
Example Extension

- Bring in an external API to provide nutritional info, exercises, beers etc
- Handle dates elegantly - let a user filter by week, month to see progress over time

Resources

- HighCharts is an open-source library for rendering responsive charts with good documentation.

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

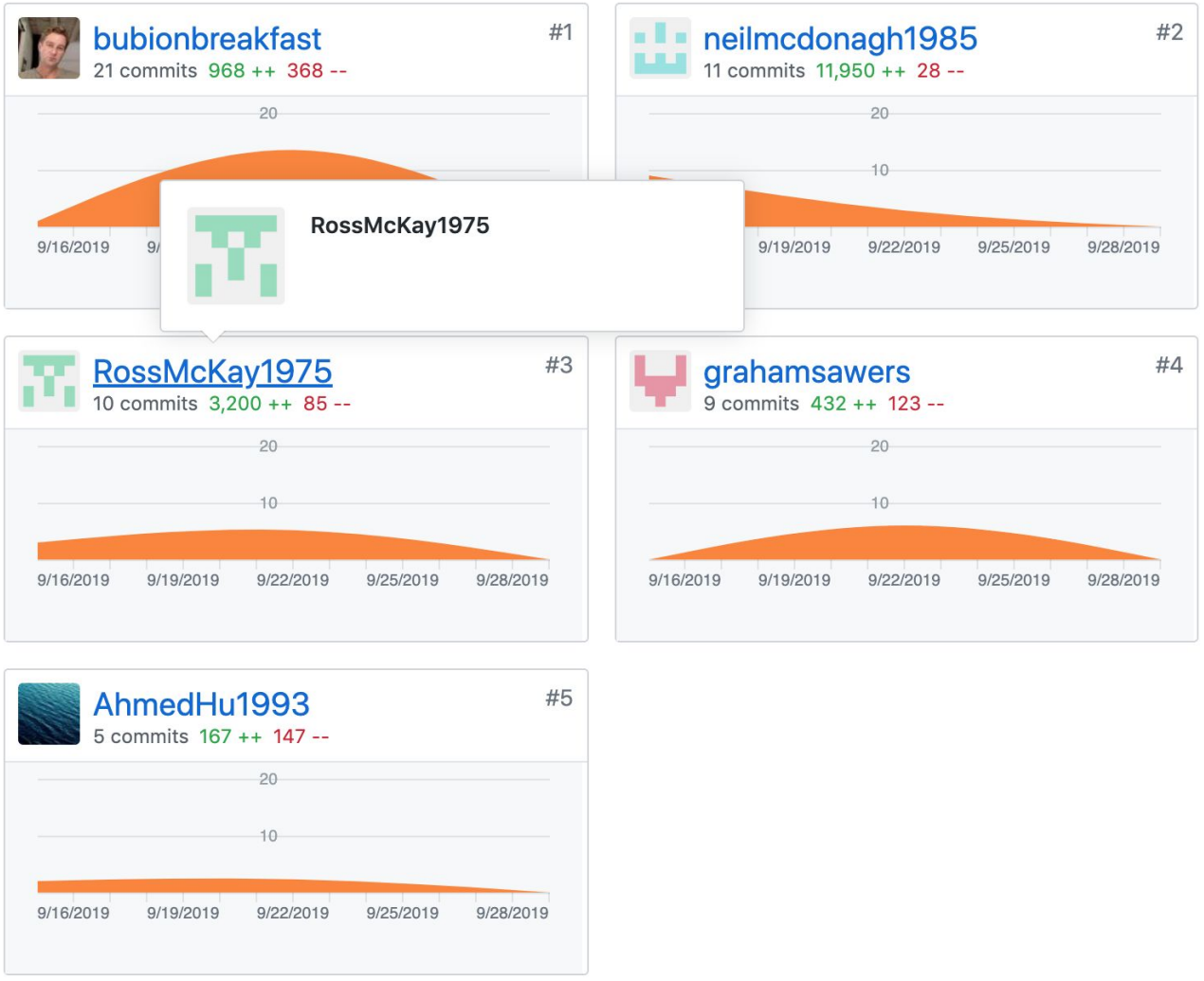


Unit	Ref	Evidence	
P	P.4	Write an acceptance criteria and test plan.	

Acceptance Criteria	**Expected Result**	**Pass/Fail**
A user is able to add a meal	breakfast appears	**pass**
A user is able to add calories	the number of calories appears next to the meal	**pass**
A user is able to view 7 days of data	A graph with 7 days of recorded data appears	**pass**
A user can see calories trends over time	A graph with all of the users data appears showing increase/decrease	**pass**
A user can delete a meal if added erroneously	The meal data is destroyed, and removed from the list	**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.	



Week 11

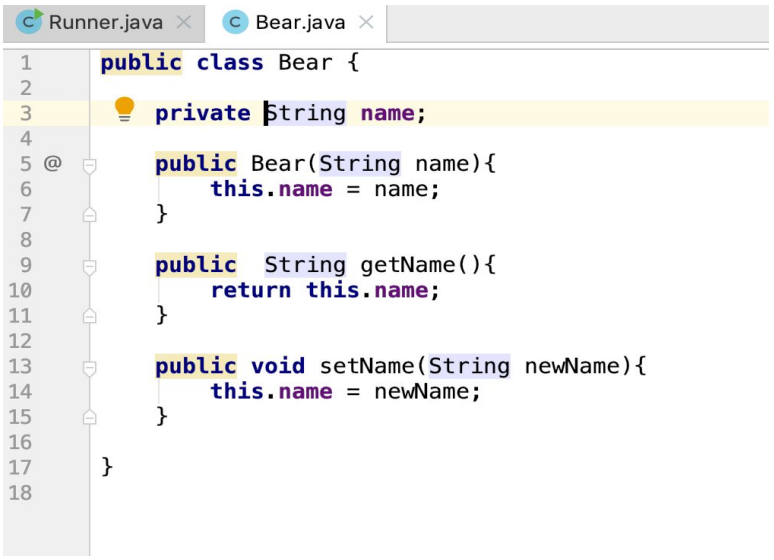
Unit	Ref	Evidence	
P	P.18	Demonstrate testing in your program. Take screenshots of: * Example of test code * The test code failing to pass * Example of the test code once errors have been corrected * The test code passing	

```
13
14    it('add 1 to 4 and get 5', function(){
15      const actual = calculator.add(4)
16      assert.equal(actual, 5)
17    })
```

```
[➜ js_calculator_start_point git:(master) ✗ npm test

> js_calculator_start_point@1.0.0 test /Users/user/e33_classnotes/week_
art_point
> mocha tests/unit/calculator_spec.js



  calculator
    ✓ it has a sample test
    1) add 1 to 4 and get 5


  1 passing (8ms)
  1 failing

  1) calculator
       add 1 to 4 and get 5:

      AssertionError [ERR_ASSERTION]: 4 == 5
      + expected - actual

      -4
      +5

      at Context.<anonymous> (tests/unit/calculator_spec.js:16:12)
      at processImmediate (internal/timers.js:439:21)


npm ERR! Test failed.  See above for more details.
```

```
[➜ js_calculator_start_point git:(master) ✗ npm test

> js_calculator_start_point@1.0.0 test /Users/user/e33_classnotes
art_point
> mocha tests/unit/calculator_spec.js



  calculator
    ✓ it has a sample test
    ✓ add 1 to 4 and get 5


  2 passing (8ms)
```

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



Here the Bear class encapsulates the data in the form of the name of the Bear. This data is restricted to be private to the class of Bear. The class includes a method in the form of “getName” to make this data available for other parts of the program. Therefore the Bear class is an example of encapsulation

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

Description here

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

Paste Screenshot here

Description here

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

Description here

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

Description here

Unit	Ref	Evidence	

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here

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

Paste Screenshot here

Description here