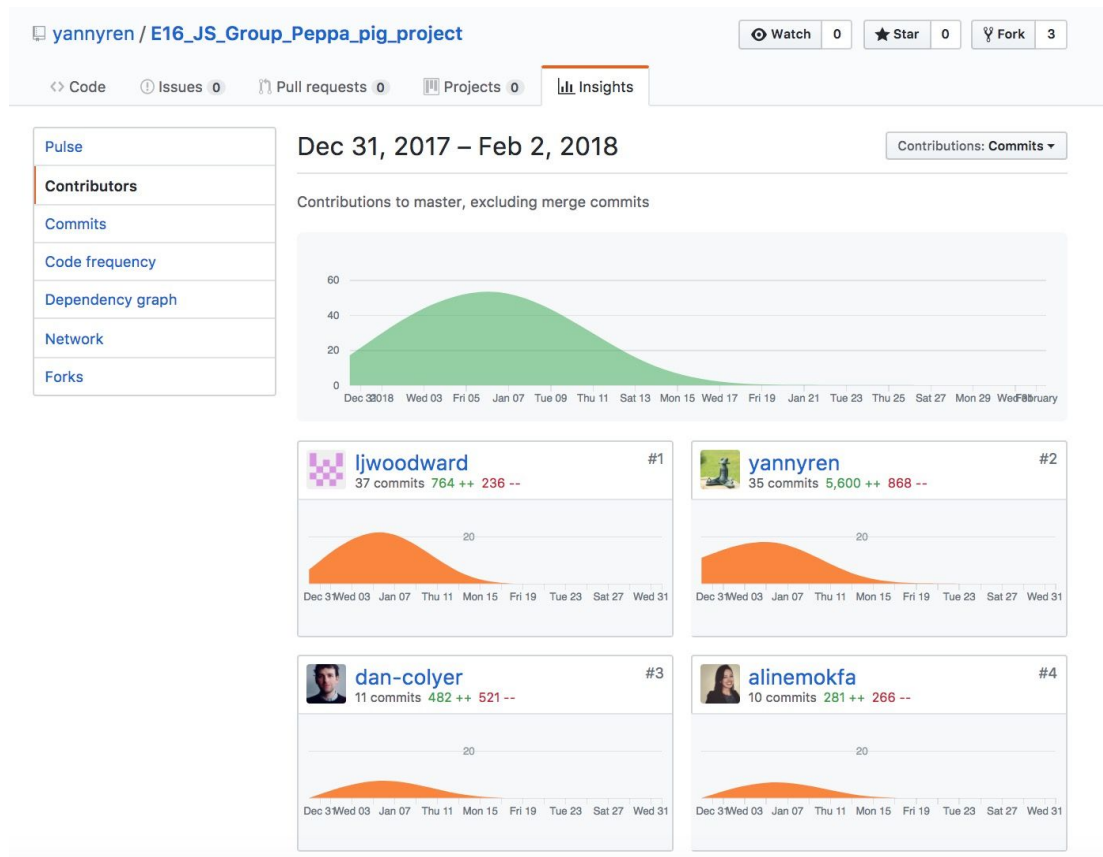


Evidence for Project Unit

Aline Mokfa
Cohort E16
02/Feb/2018

P 1 - Group Project | Contributor's Github page



P 2 - Group project | Brief

Educational App

The BBC are looking to improve their online offering of educational content by developing some interactive apps that display information in a fun and interesting way.

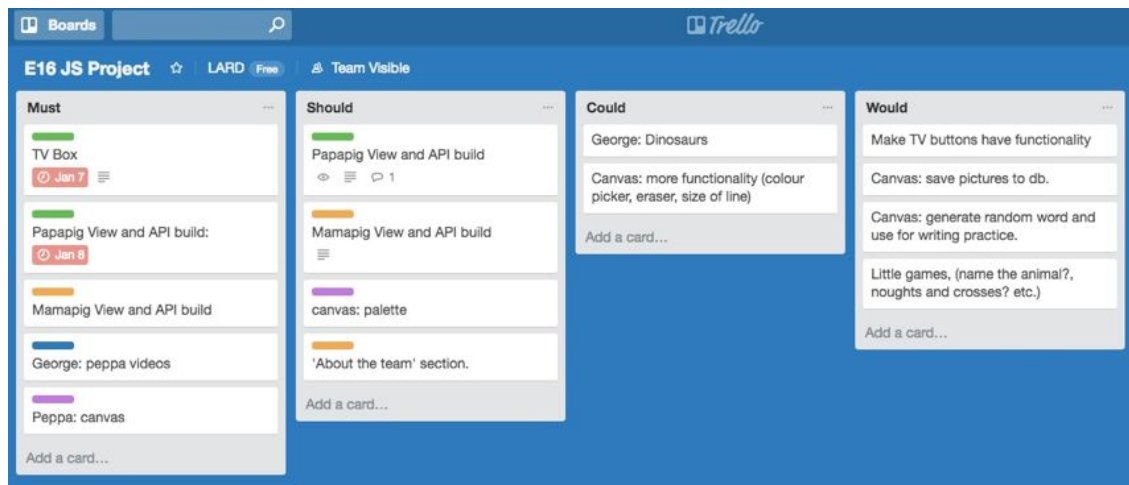
Your task is to make an MVP to put forward to them - this may only be for a small set of information, and may only showcase some of the features to be included in the final app. You might use an API to bring in content or a database to store facts. The topic of the app is your choice, but here are some suggestions you could look into:

- Interactive map of a historical event - e.g. World War 1, the travels of Christopher Columbus

MVP

- Display some information about a particular topic in an interesting way
- Have some user interactivity using event listeners, e.g to move through different sections of content

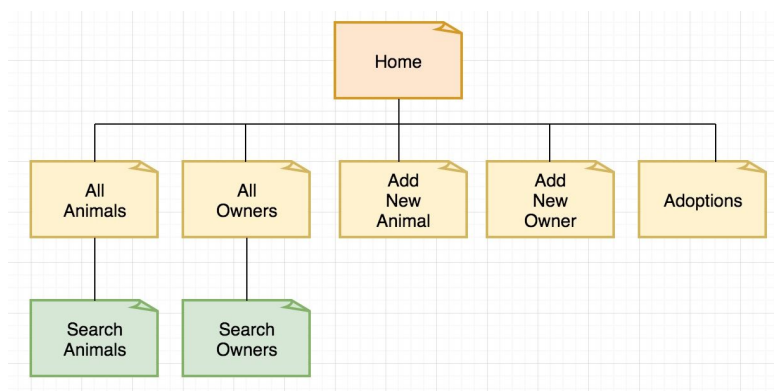
P 3 - Group project | Planning



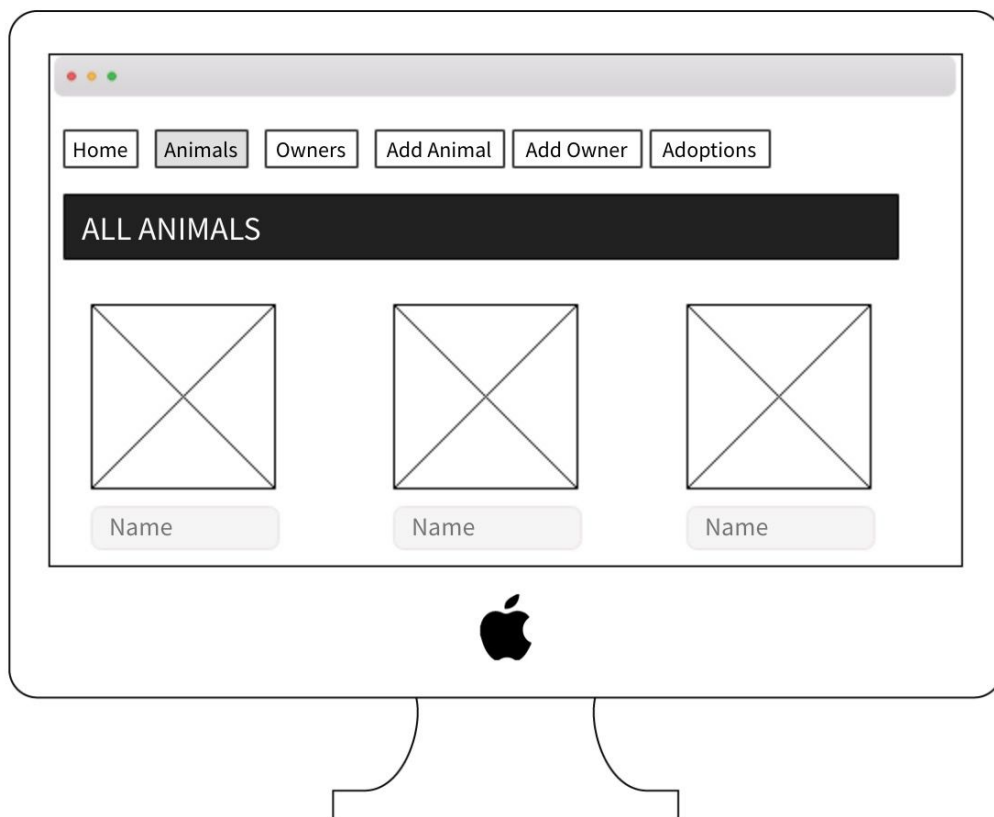
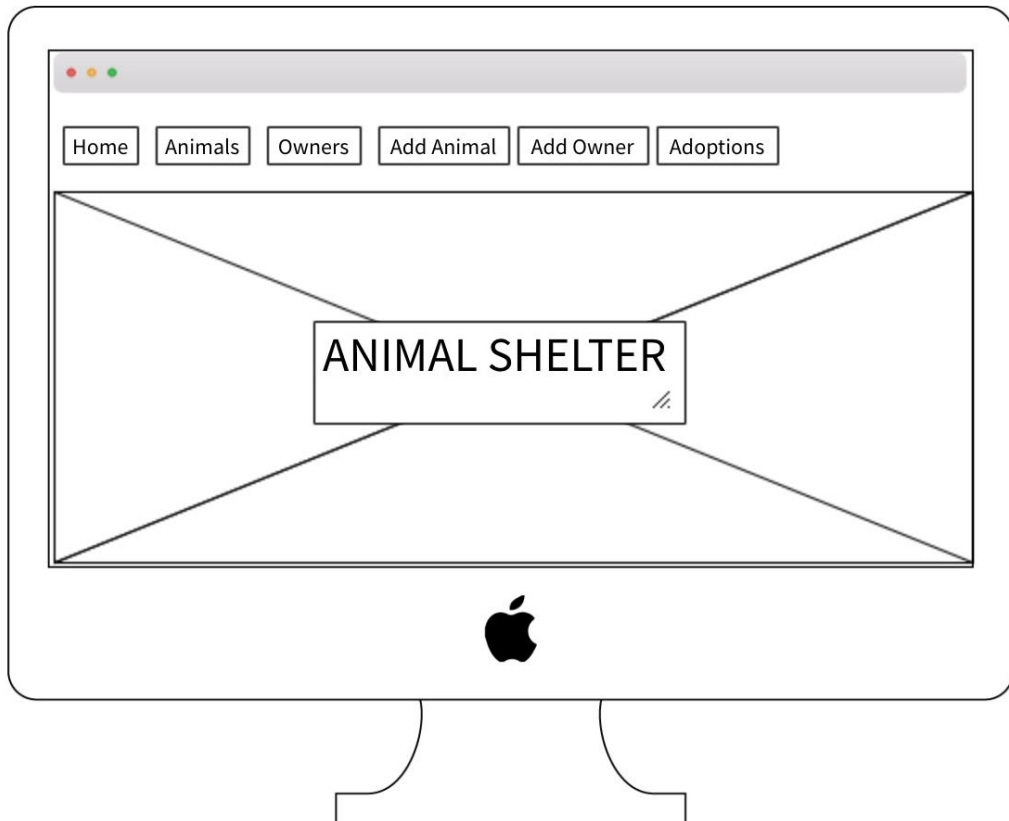
P 4 - Acceptance Criteria and Test Plan

Acceptance Criteria	Expected Result/Output	Pass / Fail
A user is able to search for a particular animal by name or type	Animals that match the name or type are displayed below the search box	PASS
A user is able to add/register a new animal	Animal is saved to database when the 'Add New Animal' button is clicked and displayed on the 'All Animals' page	PASS
A user can see a list of owners	Owners are displayed when the 'All Owners' page is visited by clicking the owners page link	PASS
A user can see a list of successful adoptions	Adoptions are displayed when clicking the adoptions page link	PASS

P 5 - User sitemap

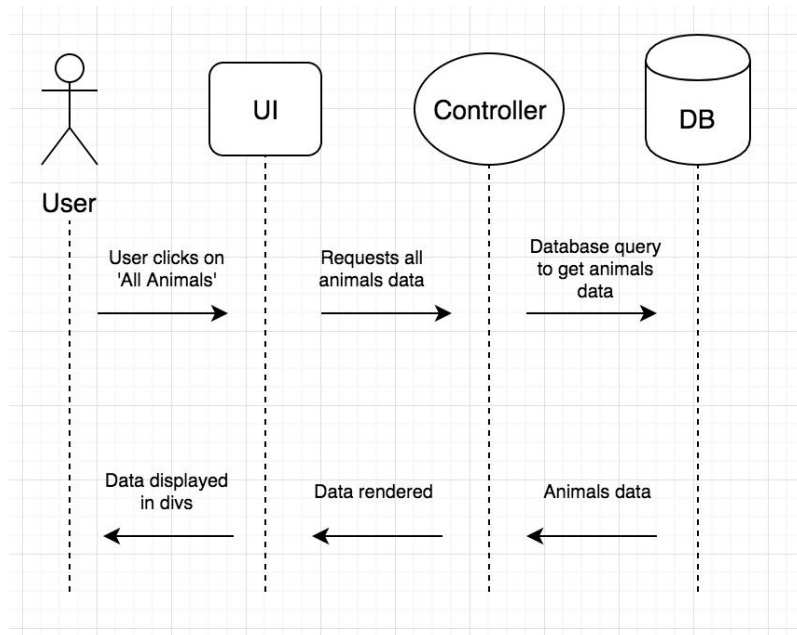


P 6 - Wireframe designs

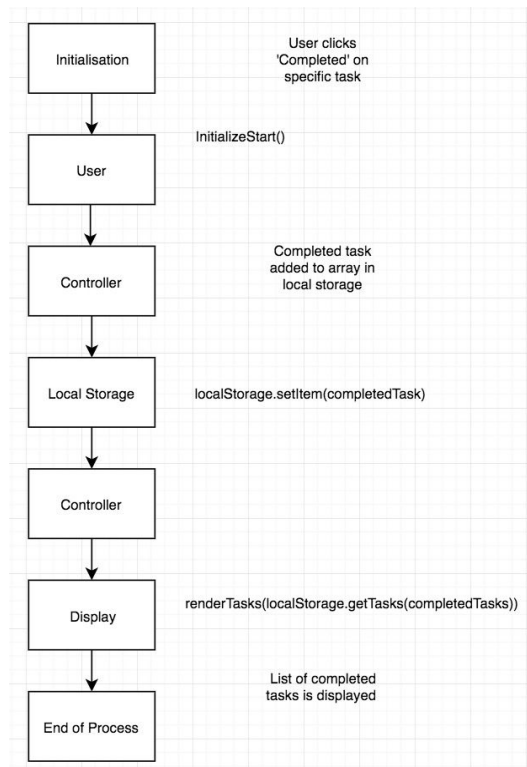


P 7 - System interactions diagrams

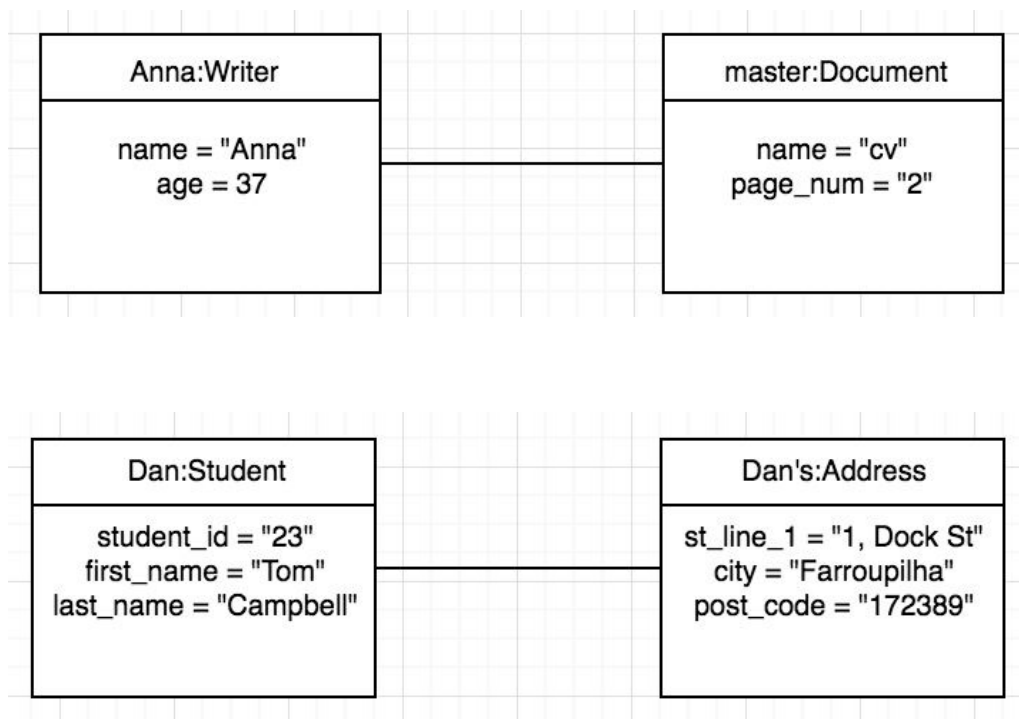
Animal Shelter Web App



To Do List App



P 8 - Two Object Diagrams



P 9 - Two Algorithms

Search Algorithm

For my first project, I added a search feature so the staff could quickly find animals registered with the shelter. I decided to use a search algorithm where the animal could be found by name, type, or breed. The results were then returned in a new array, after iteration.

```
def self.find_in_search(search)
  sql = "SELECT * FROM animals WHERE
  name LIKE $1
  OR type LIKE $1
  OR breed LIKE $1"
  values = ["%#{search.capitalize}%"]
  results = SqlRunner.run(sql, values)
  return results.map { |animal| Animal.new(animal)}
end
```

Update Algorithm

In the same project, I added an update button so any animal's details could be easily changed. The update algorithm allowed me to go and find the item by ID and update it, saving it to the database. The algorithm was written using SQL injection prevention and data validation.

```

def update()
  sql = "UPDATE animals
    SET (
      name,
      type,
      breed,
      adoption_status,
      image_url,
      owner_id
    ) =
    (
      $1, $2, $3, $4, $5, $6
    ) WHERE id = $7"

  if(@owner_id != nil)
    @adoption_status = 'f'
  end

  values = [@name, @type, @breed, @adoption_status, @image_url, @owner_id, @id]
  SqlRunner.run(sql, values)
end

```

P 10 - Pseudocode

```

#create function to search for an entry in an address book
#get first entry
#name entry 'address'
#while 'address' ins't the searched for entry
#get next entry
#end loop
#once entry is found, return it
#end function

```

P 11 - Solo project

https://github.com/alinemokfa/to_do_list_android_solo_project/tree/master/ToDoList

alinemokfa / to_do_list_android_solo_project

Watch 0 Star 0 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

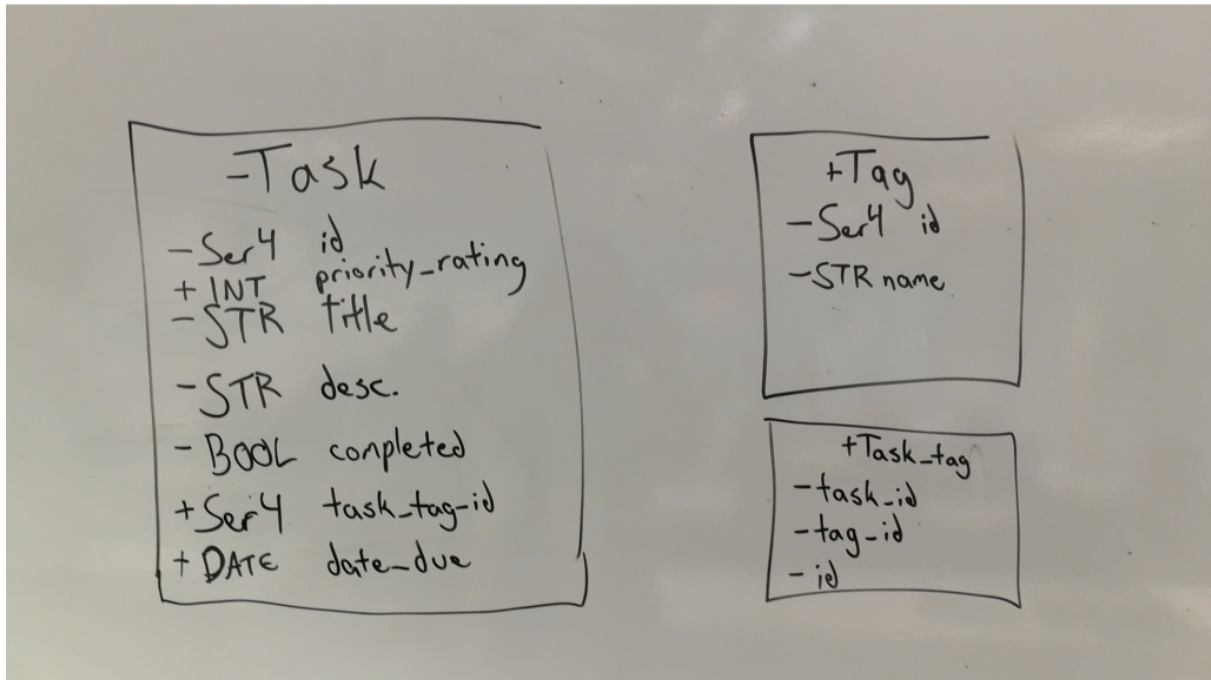
Branch: master to_do_list_android_solo_project / ToDoList / Create new file Upload files Find file History

alinemokfa changed app name Latest commit 7fe2448 on 24 Nov 2017

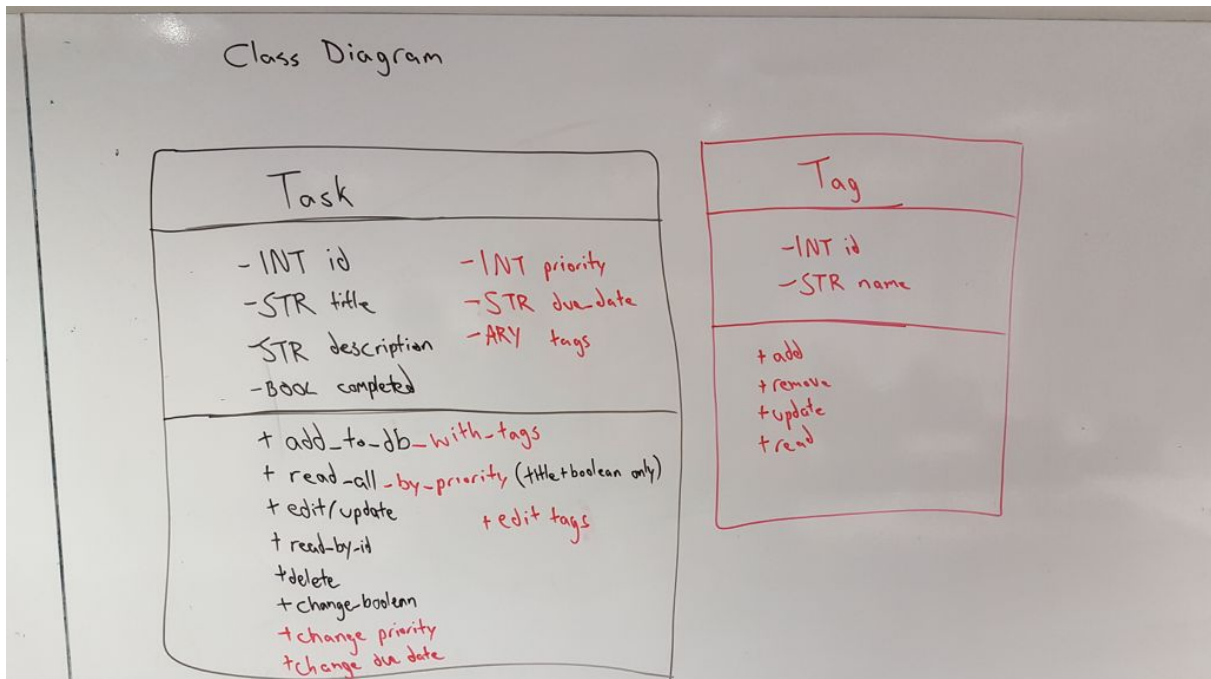
..		
.idea	set up + DatabaseHelper class created and tested	3 months ago
app	changed app name	2 months ago
gradle/wrapper	set up + DatabaseHelper class created and tested	3 months ago
.gitignore	set up + DatabaseHelper class created and tested	3 months ago
build.gradle	set up + DatabaseHelper class created and tested	3 months ago
gradle.properties	set up + DatabaseHelper class created and tested	3 months ago
gradlew	set up + DatabaseHelper class created and tested	3 months ago
gradlew.bat	set up + DatabaseHelper class created and tested	3 months ago
settings.gradle	set up + DatabaseHelper class created and tested	3 months ago

P 12 - Screenshots/photos of planning and the different stages of development to show changes.

Before:



After:



P 13 - User input

[HOME](#) [ALL ANIMALS](#) [ALL OWNERS](#) [ADD NEW ANIMAL](#) [ADD NEW OWNER](#) [ADOPTIONS](#)

NEW OWNER

First Name:

Last Name:

Address:

Email:

Phone Number:

Photo link:

Add New Owner

[HOME](#) [ALL ANIMALS](#) [ALL OWNERS](#) [ADD NEW ANIMAL](#) [ADD NEW OWNER](#) [ADOPTIONS](#)


Adrian Robson

Address: 5, Happy Lane

Email address: a.rob12@coldmail.com

Phone number: 76574829

Adopted animals: No animals.



Edit

Delete

P 14 - Interaction with data persistence

NEW ANIMAL

Name:

Bandit

Type:

Horse

Breed:

Arabian

Date of admission:

05 / 02 / 2018

Adoption Status:

Not Ready for Adoption

Photo link:

ie07236890d96eff5657

Add New Animal

Bandit

Breed: Arabian

Admission Date: 2018-02-05


Adoption status: Not Ready For Adoption



P 15 - User output result

HOME ALL ANIMALS ALL OWNERS ADD NEW ANIMAL ADD NEW OWNER ADOPTIONS

Search: dog



Name: Chew Barka

Type: Dog

P 16 - API being used in a program

Code:

```
JS MusicContainer.js JS TitleContainer.js
1 import React from 'react';
2 import TitleContainer from './TitleContainer.js'
3 import SearchContainer from './SearchContainer.js'
4 import SearchResult from '../components/SearchResult.js'
5
6 class MusicContainer extends React.Component {
7
8   constructor(props){
9     super(props);
10    this.state = {
11      results: []
12    }
13  }
14
15  componentDidMount() {
16    const url = 'https://itunes.apple.com/search?term=white+stripes';
17    const request = new XMLHttpRequest();
18    request.open('GET', url);
19    request.send();
20
21    request.addEventListener('load', () => {
22      if(request.status !== 200) return;
23      const jsonString = request.responseText;
24      const results = JSON.parse(jsonString);
25      console.log(results);
26      this.setState({results: results});
27    })
28  }
29 }
```

API being used:

iTunes Search

White + Stripes + Eleph Search

Artist:

The White Stripes

Album:

Elephant



P 17 - Bug tracking report showing the errors diagnosed and corrected

Bug		Resolution implemented	
Animal cannot be adopted if 'Not Ready for Adop.' is selected	Failed	Added validations so animals can only be adopted when ready	Passed
Owner can adopt an animal	Failed	Register an owner, use animal's ID to proceed with adoption	Passed
New animal's adoption status should be, by default, set to 'Not Ready for Adop.'	Failed	Hardcoded this option in New Animal form	Passed

P 18 - Testing your program

Test code:

```
require 'minitest/autorun'
require_relative '../models/owner.rb'

class TestOwner < Minitest::Test

  def setup
    @owner = Owner.new({
      'first_name' => 'Rose',
      'last_name' => 'Smith',
      'image_url' =>
        'https://pixabay.com/get/e834b90f20f0063ed1534705fb0938c9bd2
        2ffd41cb1184196f9c678a3/african-american-1180847_1920.jpg',
      'address' => '14/8 Dream Road',
      'email_address' => 'rose_dream@coldmail.com',
      'phone_number' => '07566728009'
    })
  end

  def test_pretty_name
    assert_equal( 'Rose Smith', owner.pretty_name() )
  end

end
```

Test code failing to pass:

```
→ animal_shelter git:(master) ✗ ruby specs/owner_spec.rb
Run options: --seed 58391

# Running:

E

Finished in 0.001020s, 980.3921 runs/s, 0.0000 assertions/s.

1) Error:
TestOwner#test_pretty_name:
NameError: undefined local variable or method `owner' for #<TestOwner:0x007fe4e0
0dd7f8>
Did you mean?  @owner
               specs/owner_spec.rb:18:in `test_pretty_name'

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

Test code once corrected:

```
require 'minitest/autorun'
require_relative '../models/owner.rb'

class TestOwner < Minitest::Test

  def setup
    @owner = Owner.new({
      'first_name' => 'Rose',
      'last_name' => 'Smith',
      'image_url' =>
        'https://pixabay.com/get/e834b90f20f0063ed1534705fb0938c9bd2
        2ffd41cb1184196f9c678a3/african-american-1180847_1920.jpg',
      'address' => '14/8 Dream Road',
      'email_address' => 'rose_dream@coldmail.com',
      'phone_number' => '07566728009'
    })
  end

  def test_pretty_name
    assert_equal( 'Rose Smith', @owner.pretty_name() )
  end

end
```

Test code passing:

```
➔ animal_shelter git:(master) ✖ ruby specs/owner_spec.rb
Run options: --seed 64716

# Running:

.

Finished in 0.001087s, 919.9632 runs/s, 919.9632 assertions/s.

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