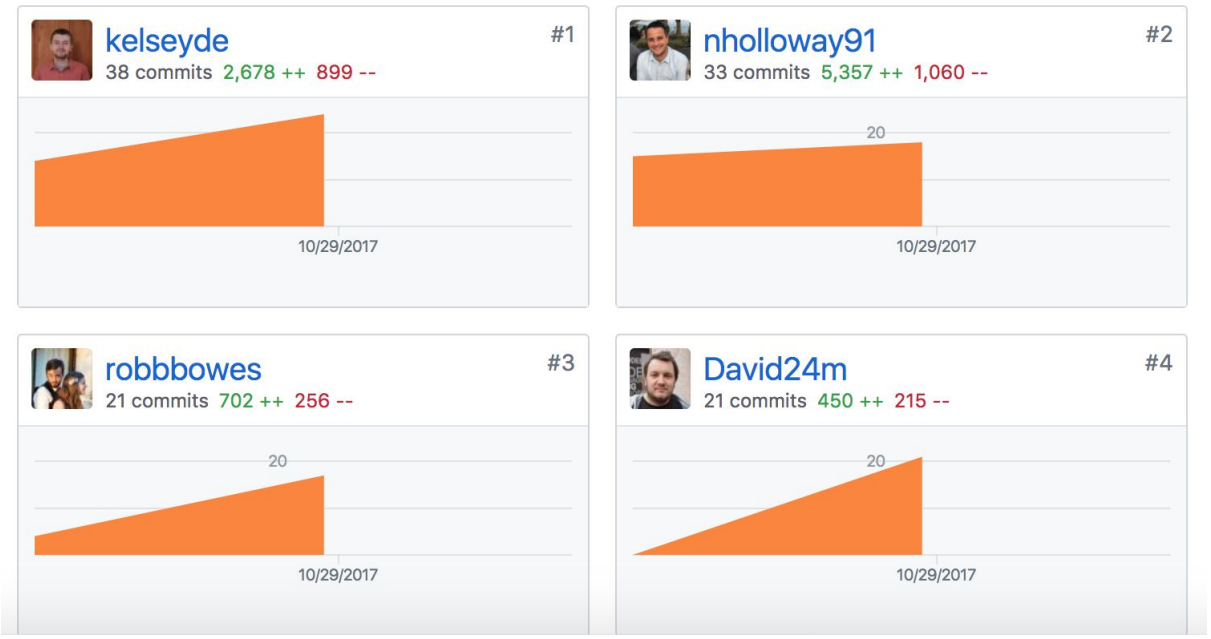
Robert James Bowes

PDA: Software Development Level 8
P - Project Unit

P 1



P 2

- **SPORTS DASHBOARD**

- Sports fans want to be able to view relevant sporting events on a dashboard. With a sport of your choice, use an existing API or create a new API to display information about fixtures, news and travel information for events.

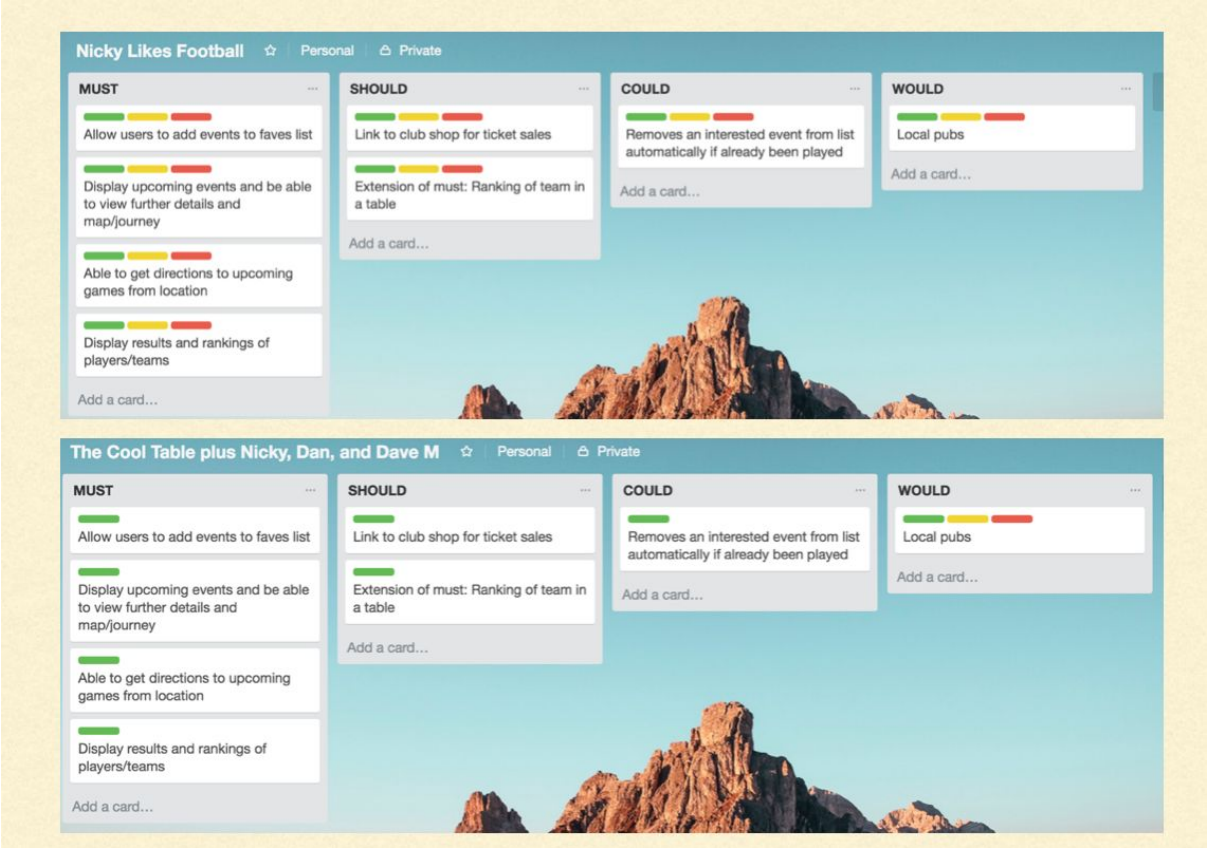
- MVP: 1) Display upcoming events on a map 2) Display results and ranking of players/ teams 3) Allow users to add events to a favourites list

- **ROUTE PLANNER**

- Visit Scotland are look for ways to encourage people to walk and cycle. Your task is to create an app that allows users to search for cycling and hiking routes, view routes on a map, save routes to a wish list and mark a route done.

- MVP: 1) Select start and finish locations for their route 2) Save routes to a wish list 3) Mark completed routes as 'done'

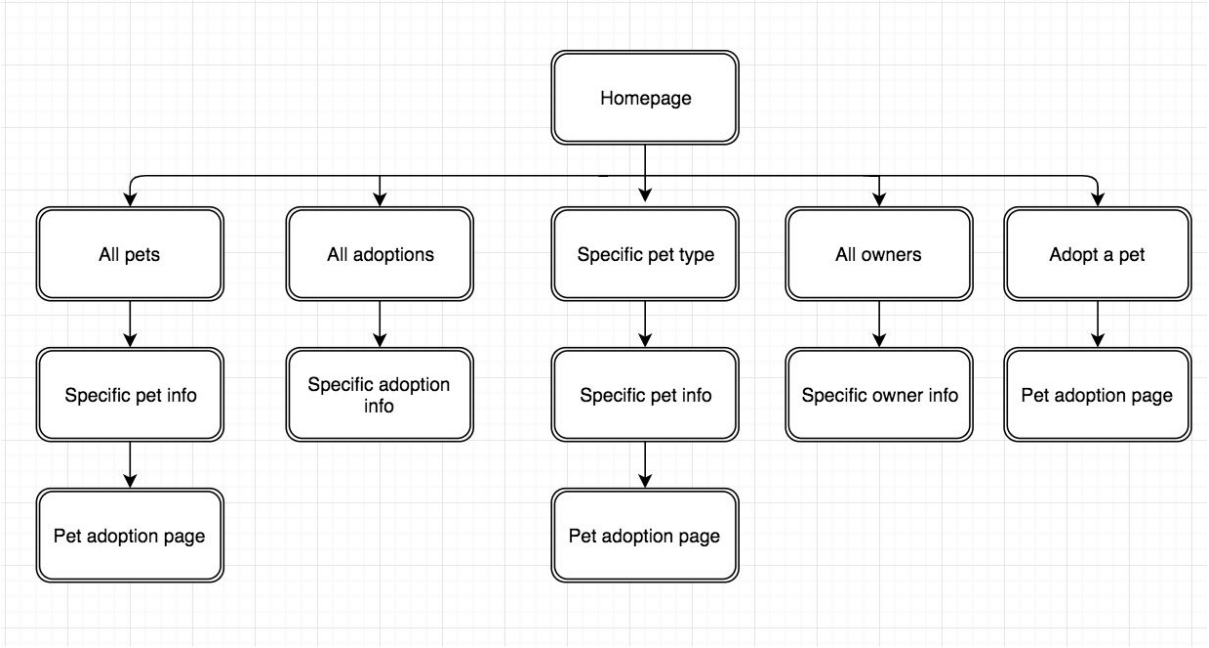
P3



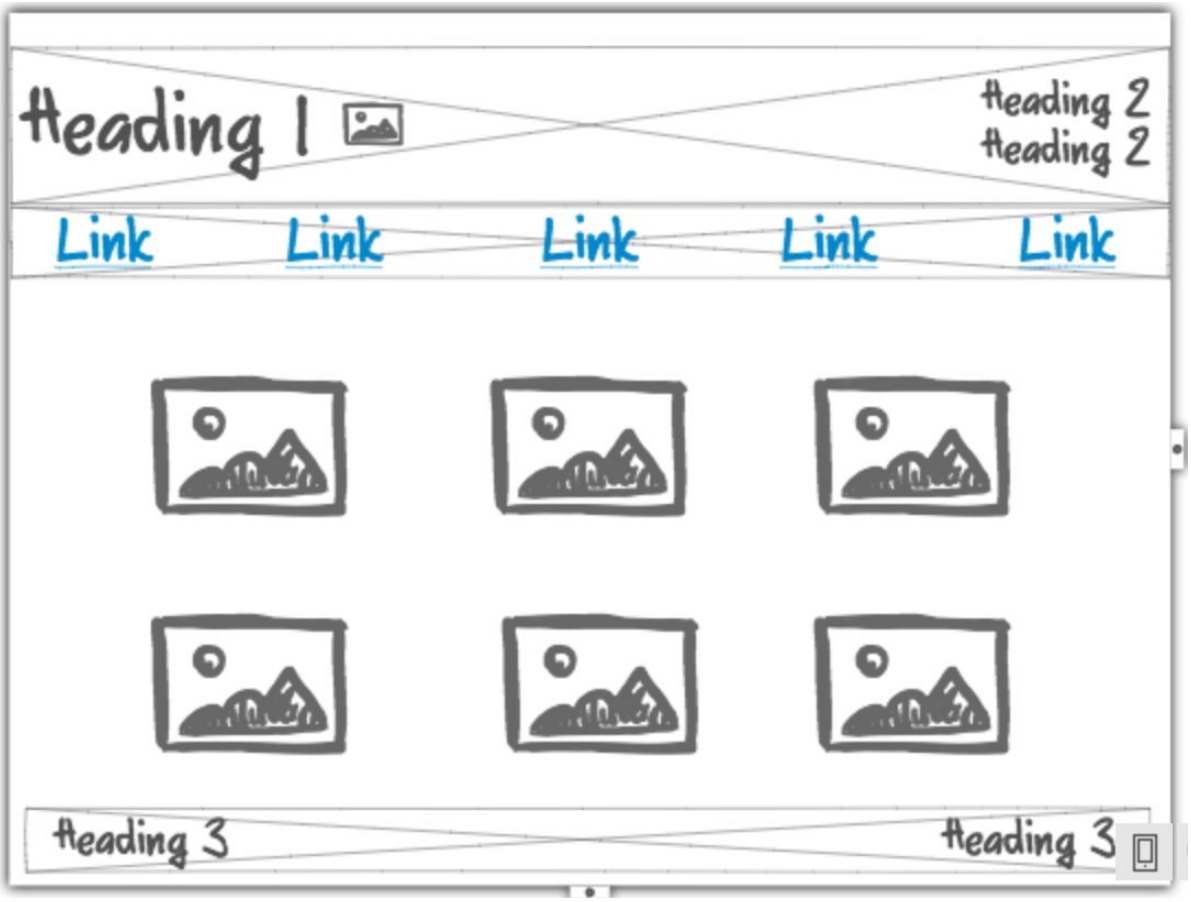
P - P4

Acceptance Criteria	Expected Result / Output	Pass / Fail
Users can see all upcoming away matches	Select drop down menu and selecting their team populates the away fixtures	Pass
Users can favourite games that they wish to add to favourites	Clicking the star icon on the away fixtures and/or the individual fixture page will add the fixture to a list of favourite fixtures	Pass
Users can get directions from their current location to the game they wish to attend	Clicking the get directions button on the away fixtures page will take the user to a page which shows directions using google maps	Pass
Users can be directed to a website in order to purchase tickets for the away fixture	Clicking the buy tickets button will take the user to the ticket ordering page of the club	Pass
Users can choose to drive, use public transport, or walk to the fixture	Choosing a different option on the select drop down menu will regenerate instructions on how to get to the fixture	Pass
Users have a custom banner at the top of the page	Choosing a team from the select drop down will programmatically change the CSS styling of the top banner and the team crest	Pass

P P5


Homepage
All pets
All adoptions
Specific pet type
All owners
Adopt a pet
Specific pet info
Specific adoption info
Specific pet info
Specific owner info
Pet adoption page
Pet adoption page
Pet adoption page

P P6


Heading 1
Heading 2
Heading 2
Link
Link
Link
Link
Link
Heading 3
Heading 3

Heading 1

Heading 3 Heading 3 Heading 3

Heading 2

Animal Type

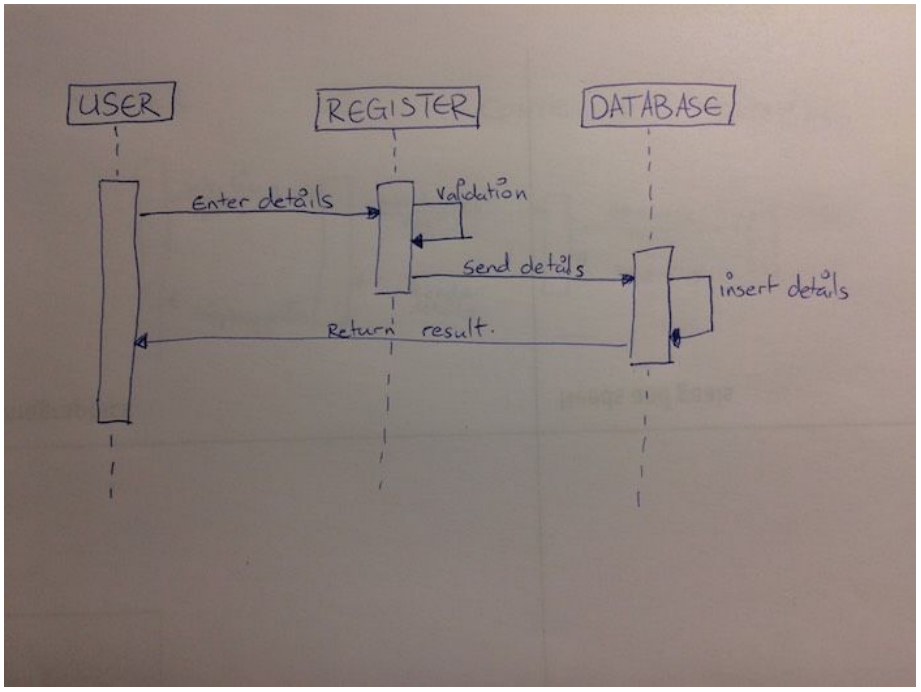
Animal Type

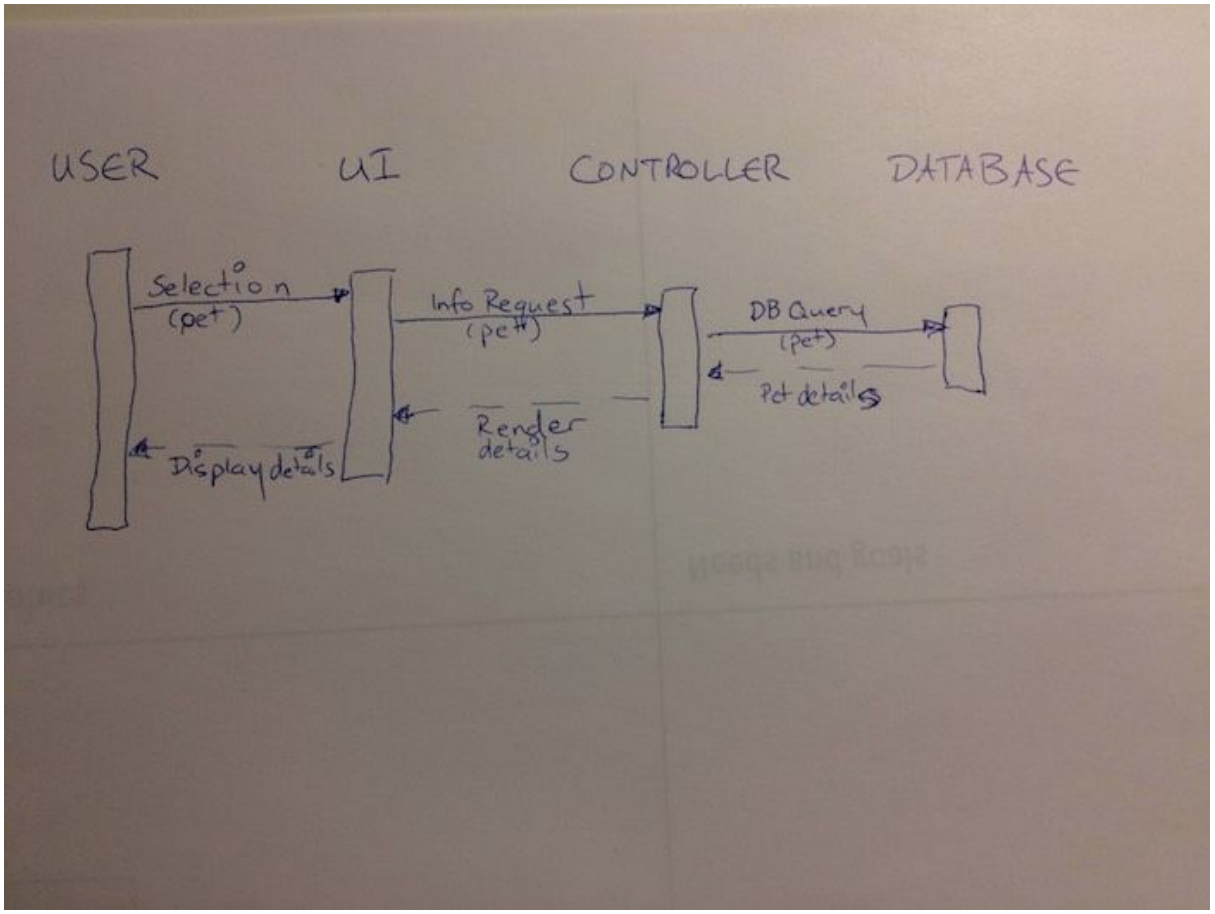
Animal Type

Animal Type

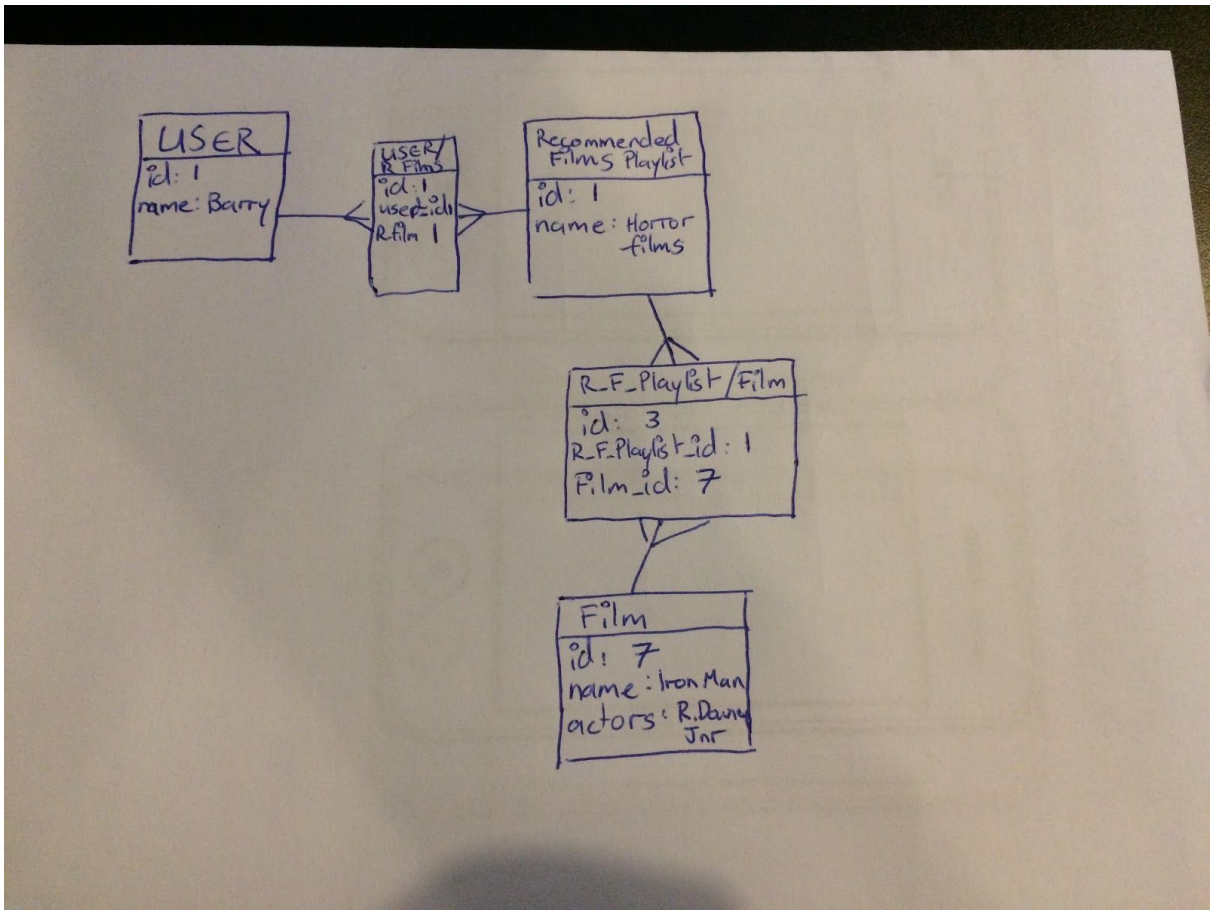
Heading 2

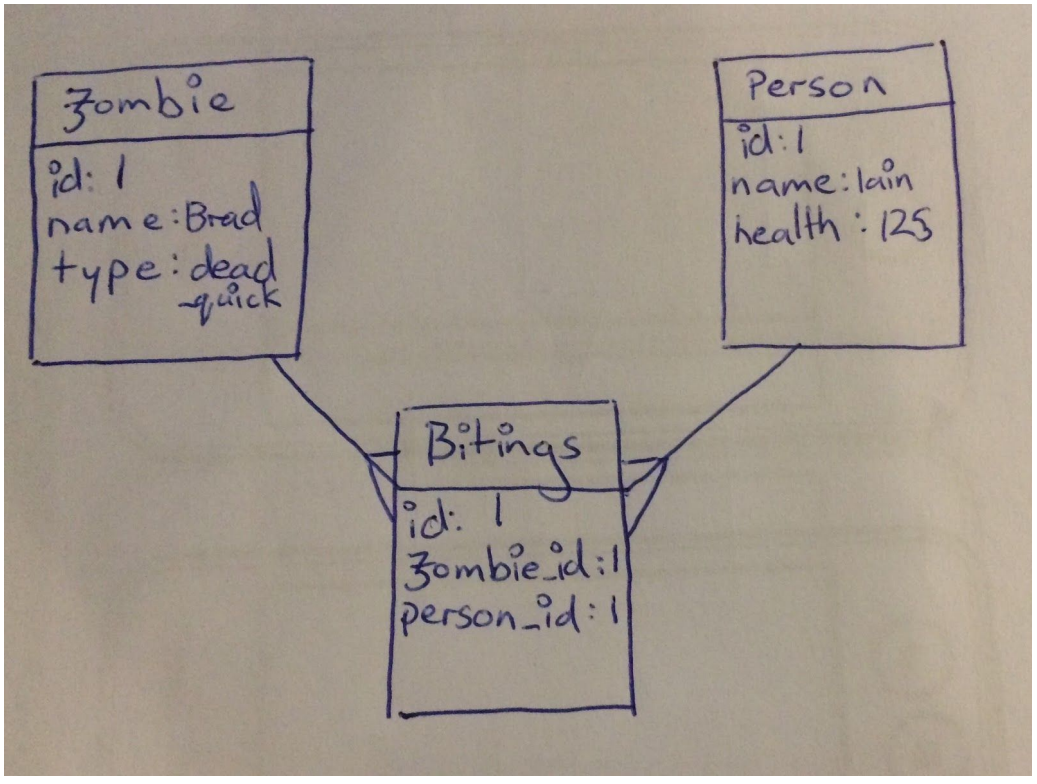
P7


USER
REGISTER
DATABASE
Enter details
Validation
Send details
Insert details
Return result.


USER
UI
CONTROLLER
DATABASE
Selection (pet)
Info Request (pet)
DB Query (pet)
Pet details
Render details
Display details

P8


USER
id: 1
name: Barry
USER/R Films
id: 1
user_id: 1
R-film 1
Recommended Films Playlist
id: 1
name: Horror films
R_F_Playlist/Film
id: 3
R_F_Playlist_id: 1
Film_id: 7
Film
id: 7
name: Iron Man
actors: R.Downey Jnr


Zombie
id: 1
name: Brad
type: dead quick
Person
id: 1
name: Iain
health: 125
Bitings
id: 1
Zombie_id: 1
person_id: 1

P9

```
int getHandWorth() {

    HashMap<Value, Integer> values = new HashMap<>();
    values.put(Value.TWO, 2);
    values.put(Value.THREE, 3);
    values.put(Value.FOUR, 4);
    values.put(Value.FIVE, 5);
    values.put(Value.SIX, 6);
    values.put(Value.SEVEN, 7);
    values.put(Value.EIGHT, 8);
    values.put(Value.NINE, 9);
    values.put(Value.TEN, 10);
    values.put(Value.JACK, 10);
    values.put(Value.QUEEN, 10);
    values.put(Value.KING, 10);
    values.put(Value.ACE, 1);

    int handWorth = 0;
    boolean aceCounter = false;

    for (Card card : hand) {
        if(card.getValue() == Value.ACE){
            aceCounter = true;
        }
        handWorth += values.get(card.getValue());
    }
    if (aceCounter && ((handWorth + 10) < 22) ) {
        return handWorth + 10;
    } else {
        return handWorth;
    }
}
```

I wrote this algorithm in order to work out the value of a player's hand. A hash was created with the card key/value pairs, then a handWorth variable was created and set as 0.

For each card in a players hand the handWorth was incremented by the value of the card. If the hand contained an Ace the aceCounter boolean would be marked as true in order to accommodate the fact that an Ace could be both high and low.

```
public String decideWinnerString() {
    dealerWhileUnder17();
    player.isBust();
    dealerPlayer.isBust();
    playerBlackjack();
    dealerBlackjack();
    if ( player.isBlackjack() && ( !dealerPlayer.isBlackjack() ) ) {
        player.setWinner(true);
        return "BLACKJACK!";
    }
    else if( ( player.getHandWorth() > dealerPlayer.getHandWorth() ) && !player.isBust() ) {
        player.setWinner(true);
        return "You win!";
    }
    else if( !player.isBust() && dealerPlayer.isBust() ) {
        player.setWinner(true);
        return "You win! The dealer is bust!";
    }
    else if( player.isBust() && dealerPlayer.isBust() ) {
        return "Nobody won this time around!";
    }
    else {
        dealerPlayer.setWinner(true);
        return "The dealer won!";
    }
}
```

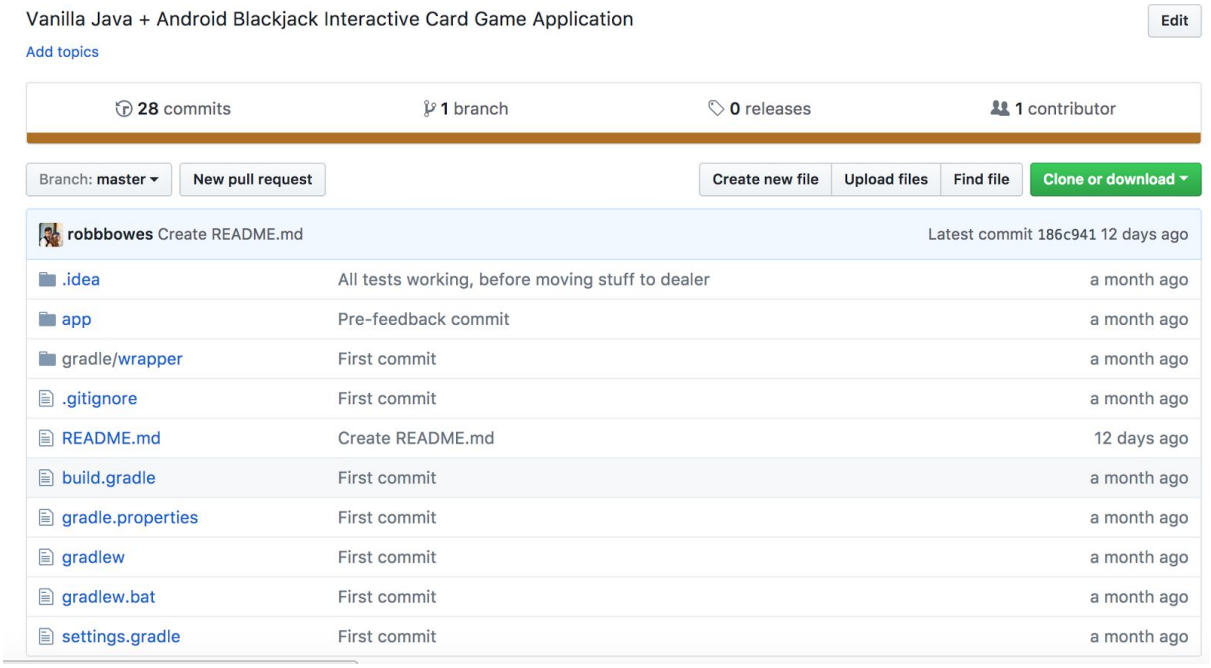
This algorithm was written in order to decide the winner of the game of Blackjack depending on the hand that the player and dealer both have in there hand at the start of either of their turns.
It checks to see whether the dealer's hand is with at least 17, if not they must take another card.
It then checks to see whether either the player or the dealer is bust depending the value of their hand, before checking whether either of them have blackjack.
It then sets the player or dealer as the winner and returns a string informing the player of the result of the game.

P10

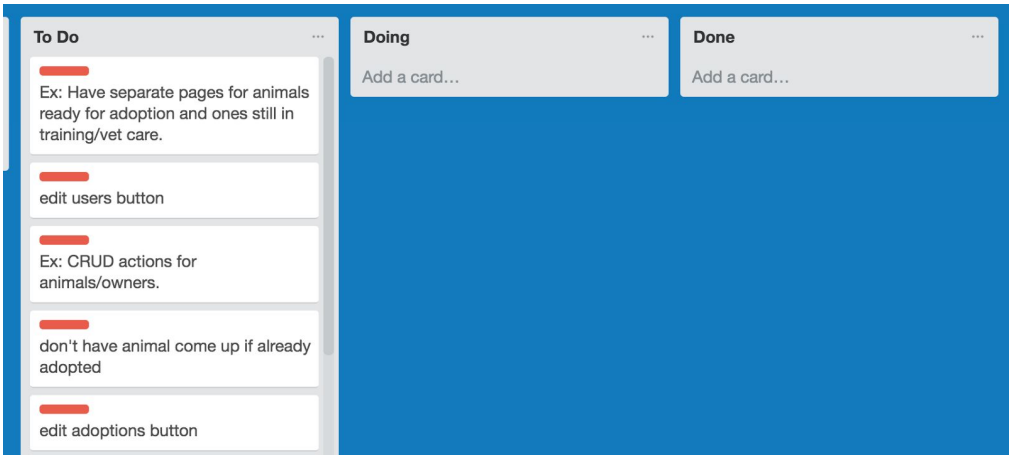
```ruby
def pets()
  # I want to select all pets from the pets table
  # Inner join on by using adoption id
  # return the details and map them to the pet class
  sql = "SELECT pets.*
  FROM pets
  INNER JOIN adoptions
  ON adoptions.pet_id = pets.id
  WHERE owner_id = $1"
  values = [@id]
  pet_data = SqlRunner.run(sql, values)
  return pet_data.map { |pet| Pet.new(pet)}
end
```

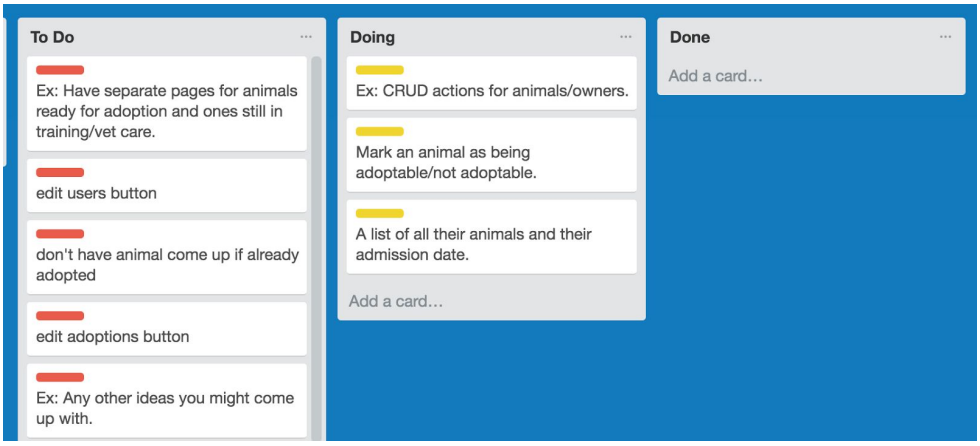
P 11

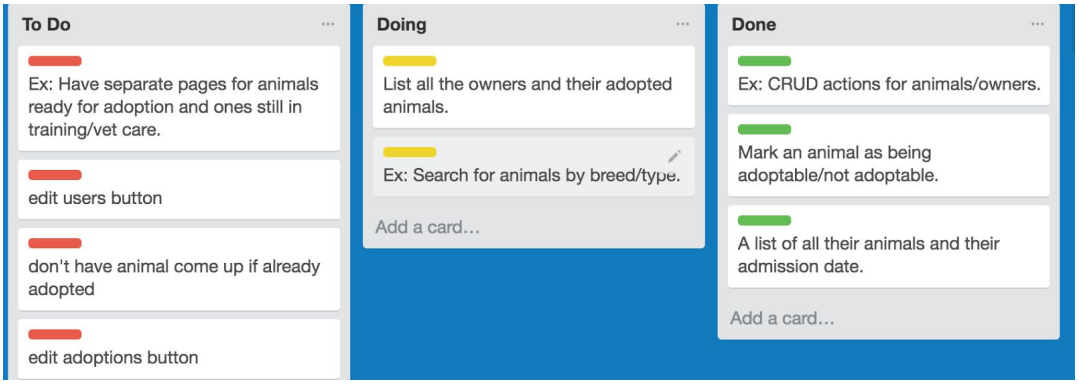
https://github.com/robbbowes/Java_Android_Blackjack



P12


To Do
Ex: Have separate pages for animals ready for adoption and ones still in training/vet care.
edit users button
Ex: CRUD actions for animals/owners.
don't have animal come up if already adopted
edit adoptions button
Doing
Add a card...
Done
Add a card...


To Do
Ex: Have separate pages for animals ready for adoption and ones still in training/vet care.
edit users button
don't have animal come up if already adopted
edit adoptions button
Ex: Any other ideas you might come up with.
Doing
Ex: CRUD actions for animals/owners.
Mark an animal as being adoptable/not adoptable.
A list of all their animals and their admission date.
Add a card...
Done
Add a card...


To Do
Ex: Have separate pages for animals ready for adoption and ones still in training/vet care.
edit users button
don't have animal come up if already adopted
edit adoptions button
Doing
List all the owners and their adopted animals.
Ex: Search for animals by breed/type.
Add a card...
Done
Ex: CRUD actions for animals/owners.
Mark an animal as being adoptable/not adoptable.
A list of all their animals and their admission date.
Add a card...

P13

All owners | Register an ow

First name: Simon

Surname: Anger

Address: 1 Streety Road

Address: Leith

Town: Embra

Picture: simon_anger.jpg

Register

Road

© 2017 Jurassic Bark Animal Shelter

Town: Embra

Edit owner?

P14

Name: Dean Anderson | Animal type: other

Dean Anderson has been with us since: 2017-08-22

Dean Anderson is currently up for adoption!

Adopt me!

Edit Dean Anderson

More others

Edit Dean Anderson (188)

Name: Dean Anderson Edited

Type: other

Admission date: 22 / 08 / 2017

Adoptable: Aye

Picture: /dean_anderson.jpg

Edit pet

Name: Dean Anderson Edited | Animal type: other

Dean Anderson Edited has been with us since: 2017-08-22

Dean Anderson Edited is currently up for adoption!

Adopt me!

Edit Dean Anderson Edited

More others

All owners | Register an owner | All our pets | Re

Here is a list of all our successful adoptions

All owners | Register an

To adopt a pet please register first.

Owner: Simon Anger

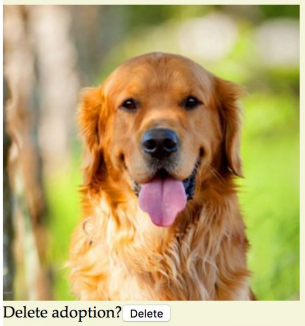
Pet: Douglas McKenzie

Adopt!

All owners | Register an owner | All our pets | R

Here is a list of all our successful adoptions

Douglas McKenzie has found a lovely home with Simon Anger!



Delete adoption? Delete

P16

```
  <head>
    <title>Harry Potter Characters!</title>
    <script src='index.js'></script>
    <link rel='stylesheet' type='text/css' href='main.css'>
```

```
var renderCharacters = function(charactersthing){
  var ulTag = document.querySelector("#character-list")
  charactersthing.forEach(function(character){
    ulTag.appendChild(createLiTags(character));
  })
}

var url = "http://hp-api.herokuapp.com/api/characters";
var request = new XMLHttpRequest();
request.open("GET", url);
request.addEventListener("load", function(){
  if (this.status === 200) {
    var characters = JSON.parse(this.responseText);
    renderCharacters(characters);
  }
})
request.send();
```

- Harry Potter
- Hermione Granger
- Ron Weasley
- Draco Malfoy
- Minerva McGonagall
- Cedric Diggory
- Cho Chang
- Severus Snape
- Rubeus Hagrid
- Neville Longbottom
- Luna Lovegood
- Ginny Weasley
- Sirius Black
- Remus Lupin
- Arthur Weasley
- Bellatrix Lestrange
- Lord Voldemort
- Horace Slughorn
- Kingsley Shacklebolt
- Dolores Umbridge
- Lucius Malfoy
- Vincent Crabbe
- Gregory Goyle
- Mrs Norris
- Argus Filch

P17

User can select their favourite team from drop down menu			**Passed**
User can add away games to their favourite fixtures	**Failed**	Away fixtures persisted in local storage	**Passed**
Users can decide to change their method of getting to the fixture	**Failed**	New request to directions api with updated method of transport	**Passed**
User can buy get a link to website in order to buy tickets for the fixture	**Failed**	Add the opposing teams' ticket office url in local mongodb and draw website address from there	**Passed**
Users favourite club influences website design	**Failed**	Programmatically change the CSS styling according to the selected club by assigning a class to the main header and club crest divs	**Passed**