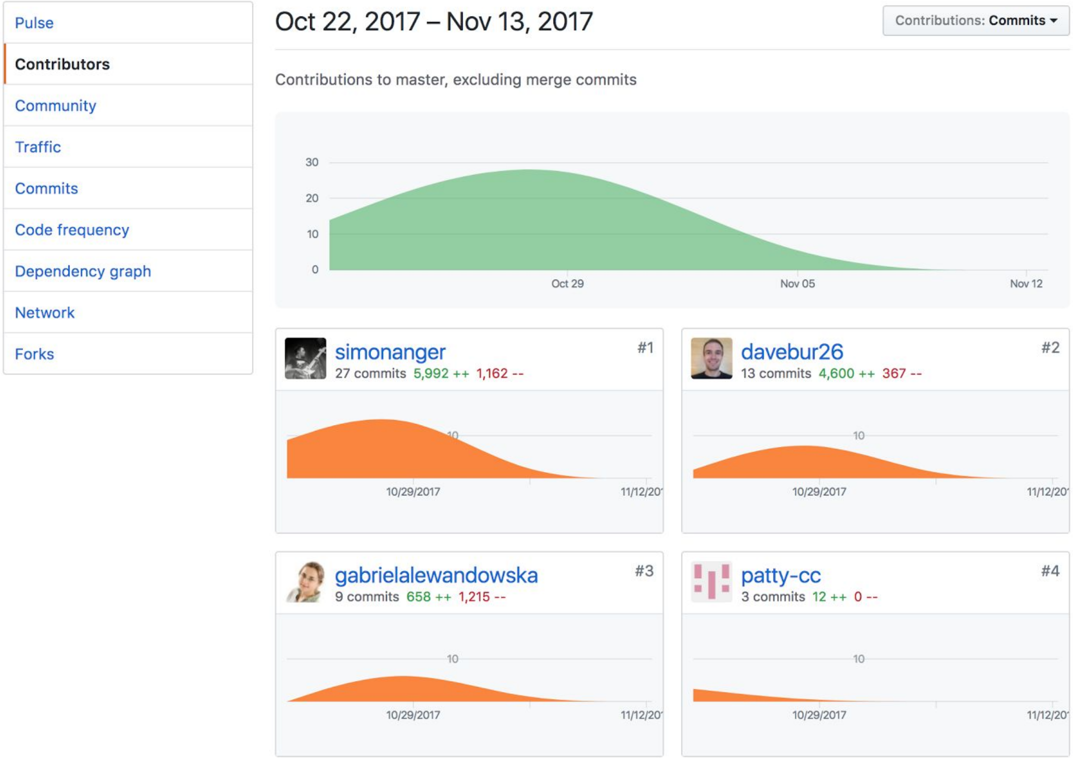
PDA: Software Development

Project Unit
Level 8

Iain Paterson
Cohort E15

P1

GitHub contributors page



Oct 22, 2017 – Nov 13, 2017

Contributions to master, excluding merge commits

simonanger #1
27 commits 5,992 ++ 1,162 --

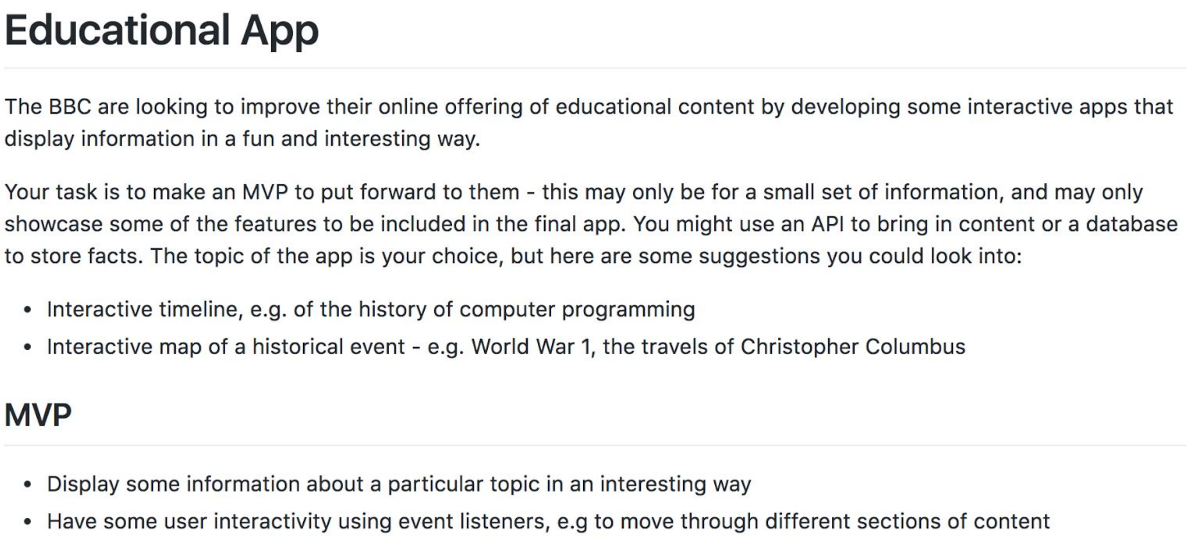
davebur26 #2
13 commits 4,600 ++ 367 --

gabrielalewandowska #3
9 commits 658 ++ 1,215 --

patty-cc #4
3 commits 12 ++ 0 --

P2

Project Brief for group project



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:

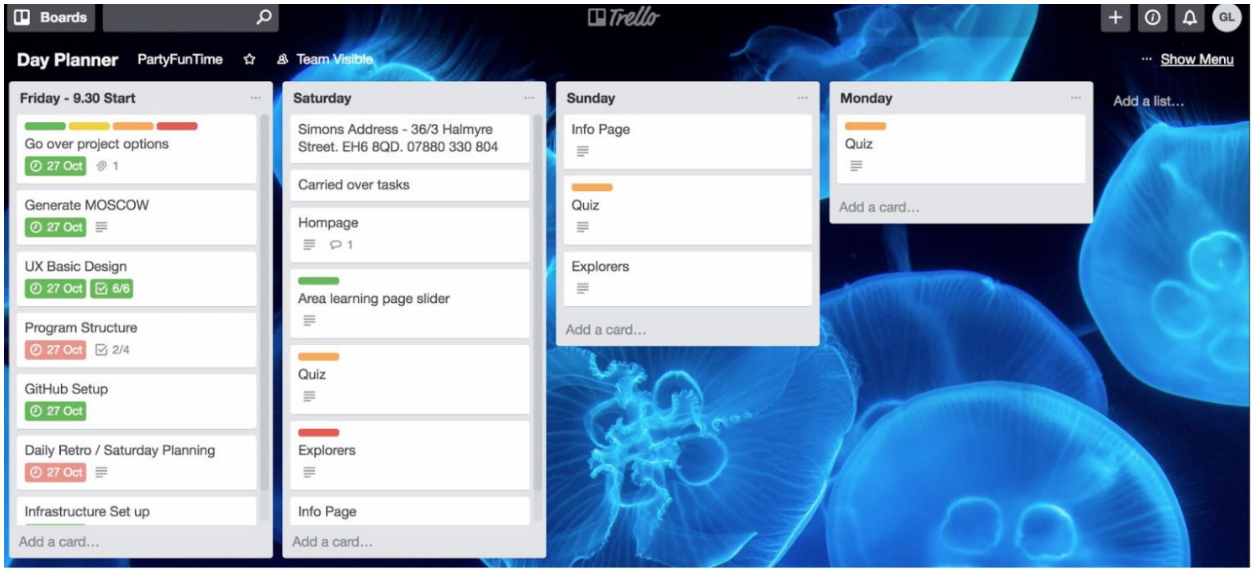
- Interactive timeline, e.g. of the history of computer programming
- Interactive map of a historical event - e.g. World War 1, the travels of Christopher Columbus

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

P3

Screenshots of group planning stages

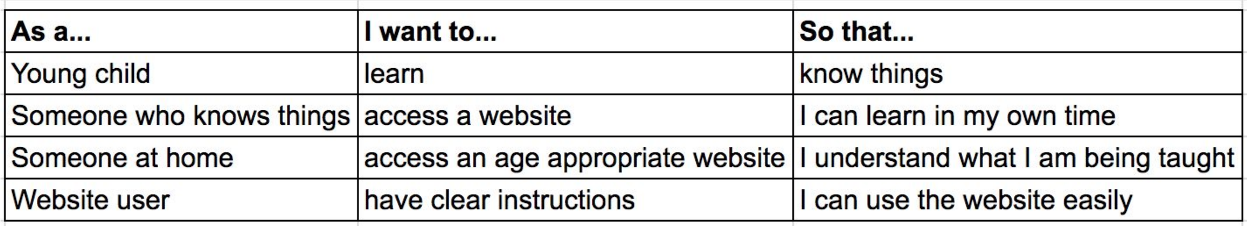


Friday - 9.30 Start
- Go over project options — 27 Oct — 1
- Generate MOSCOW — 27 Oct
- UX Basic Design — 27 Oct — 6/6
- Program Structure — 27 Oct — 2/4
- GitHub Setup — 27 Oct
- Daily Retro / Saturday Planning — 27 Oct
- Infrastructure Set up
- Add a card...

Saturday
- Simons Address - 36/3 Halmyre Street. EH6 8QD. 07880 330 804
- Carried over tasks
- Hompage — 1
- Area learning page slider
- Quiz
- Explorers
- Info Page
- Add a card...

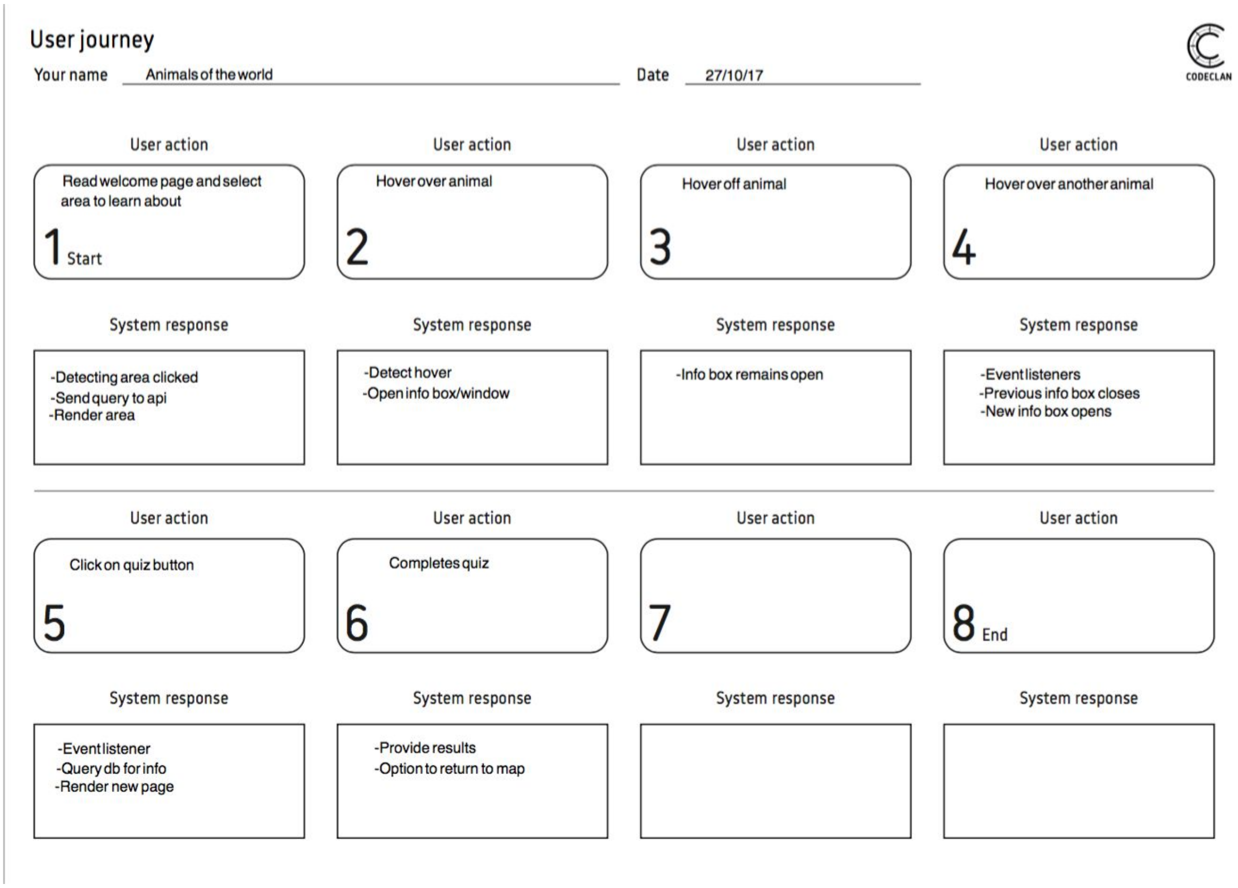
Sunday
- Info Page
- Quiz
- Explorers
- Add a card...

Monday
- Quiz
- Add a card...



As a...	I want to...	So that...
Young child	learn	know things
Someone who knows things	access a website	I can learn in my own time
Someone at home	access an age appropriate website	I understand what I am being taught
Website user	have clear instructions	I can use the website easily

User Journey



User journey

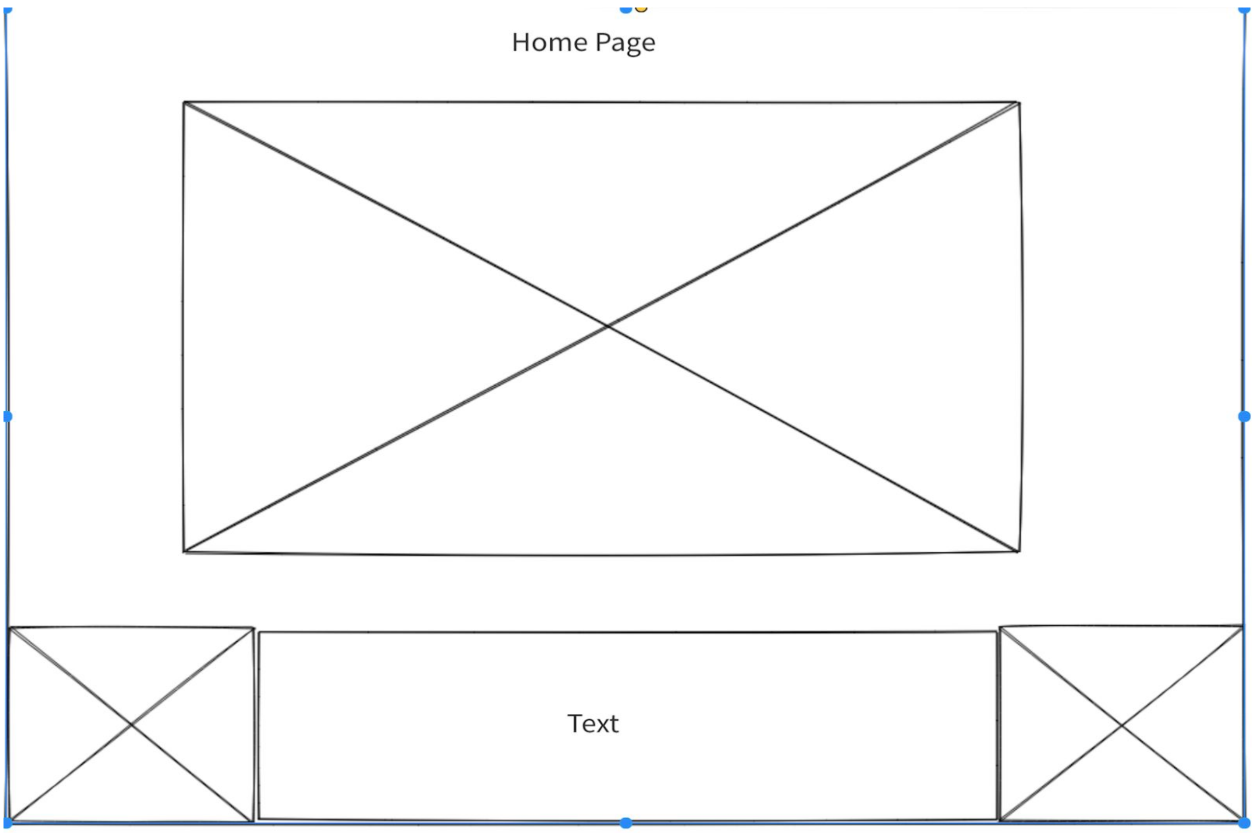
Your name Animals of the world

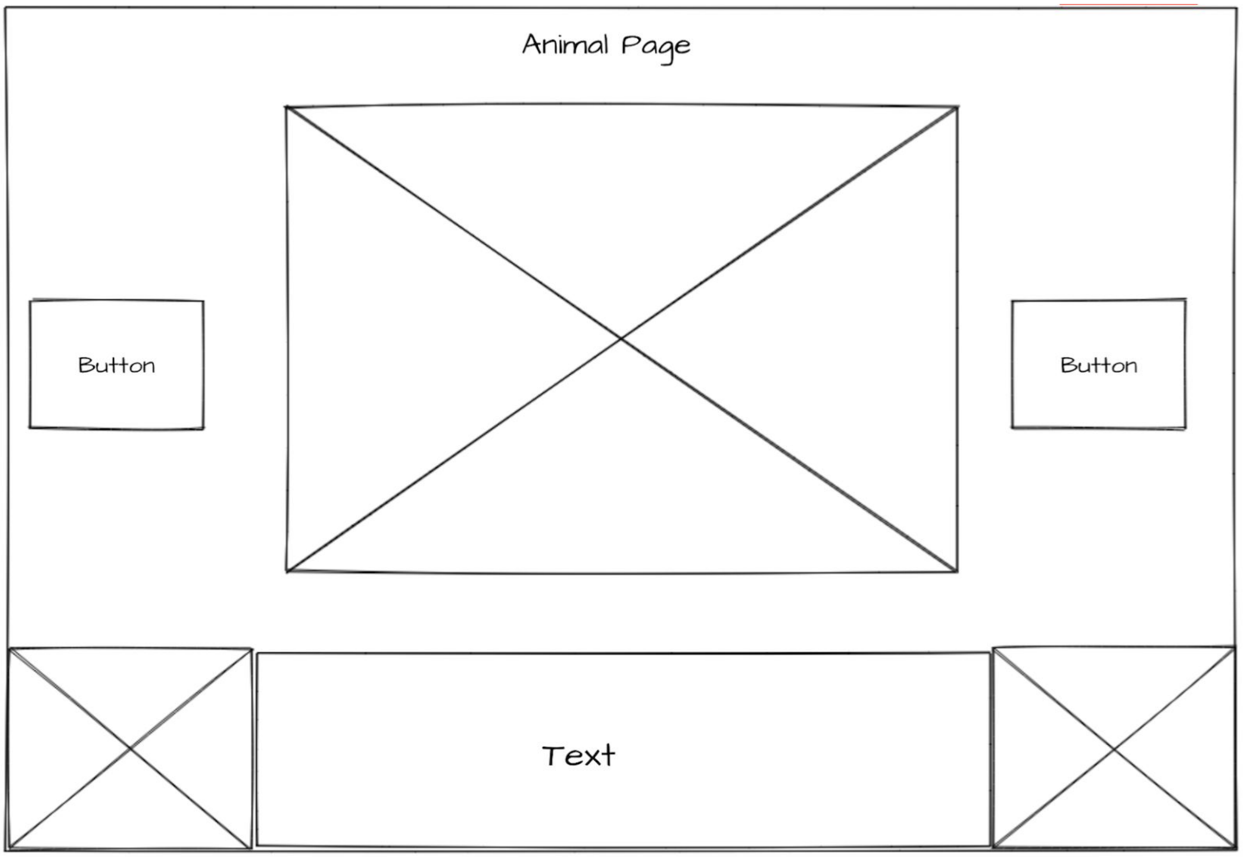
Date 27/10/17

User action	User action	User action	User action
Read welcome page and select area to learn about 1 Start	Hover over animal 2	Hover off animal 3	Hover over another animal 4
System response	**System response**	**System response**	**System response**
-Detecting area clicked -Send query to api -Render area	-Detect hover -Open info box/window	-Info box remains open	-Event listeners -Previous info box closes -New info box opens

User action	User action	User action	User action
Click on quiz button 5	Completes quiz 6	7	8 End
System response	**System response**	**System response**	**System response**
-Event listener -Query db for info -Render new page	-Provide results -Option to return to map		

Wireframes

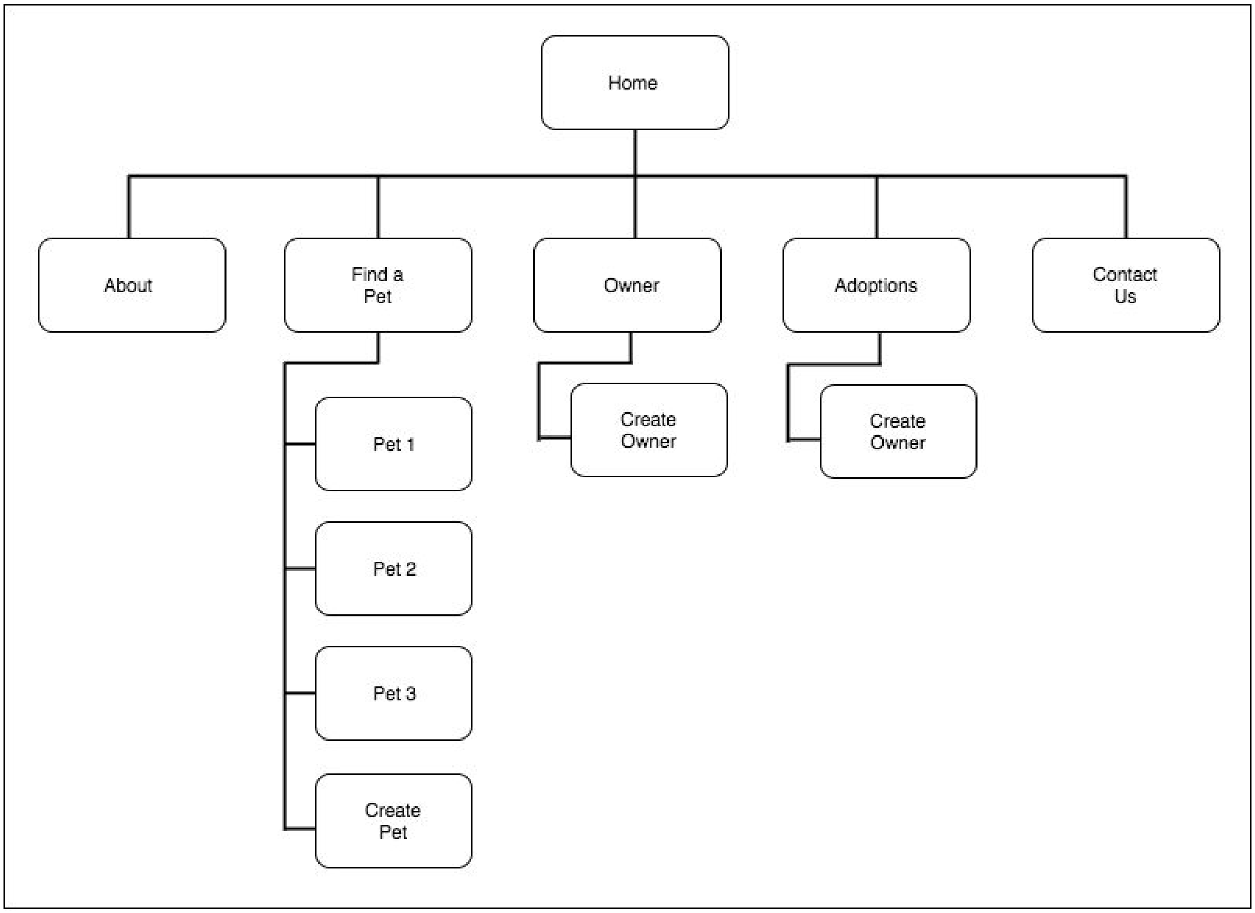

Home Page
Text


Animal Page
Button
Button
Text

P4
Acceptance Criteria

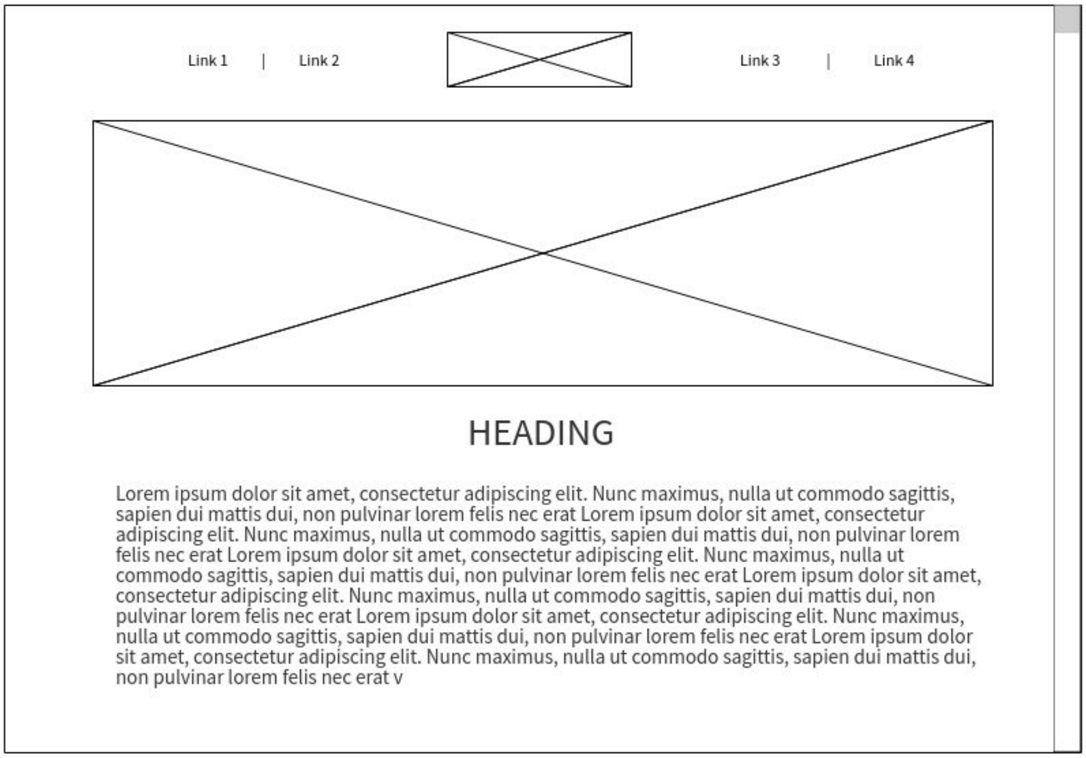
Acceptance Criteria	Expected Result/Output	Pass / Fail
User can select between pets in the vet and pets ready for adoption	A drop down menu allows user to filter between adoptable pets and pets in vets. Only showing filtered for pets	Pass
User can enter pet information and save it to a database	Clicking the ‘create pet’ button will save the given information to the database	Pass
User can delete pet information from the database	Once the ‘delete’ button has been pressed the pet information will be deleted from the database	Pass
User should be able to navigate to social media	Social media pages are displayed when the appropriate link has been pressed	Pass
User should be able to filter pets by type of animal	A dropdown displaying different types of animals is available for the user to select a type of pet. Once pressed only pets of that type will be displayed.	Fail

P5
Use of a Sitemap



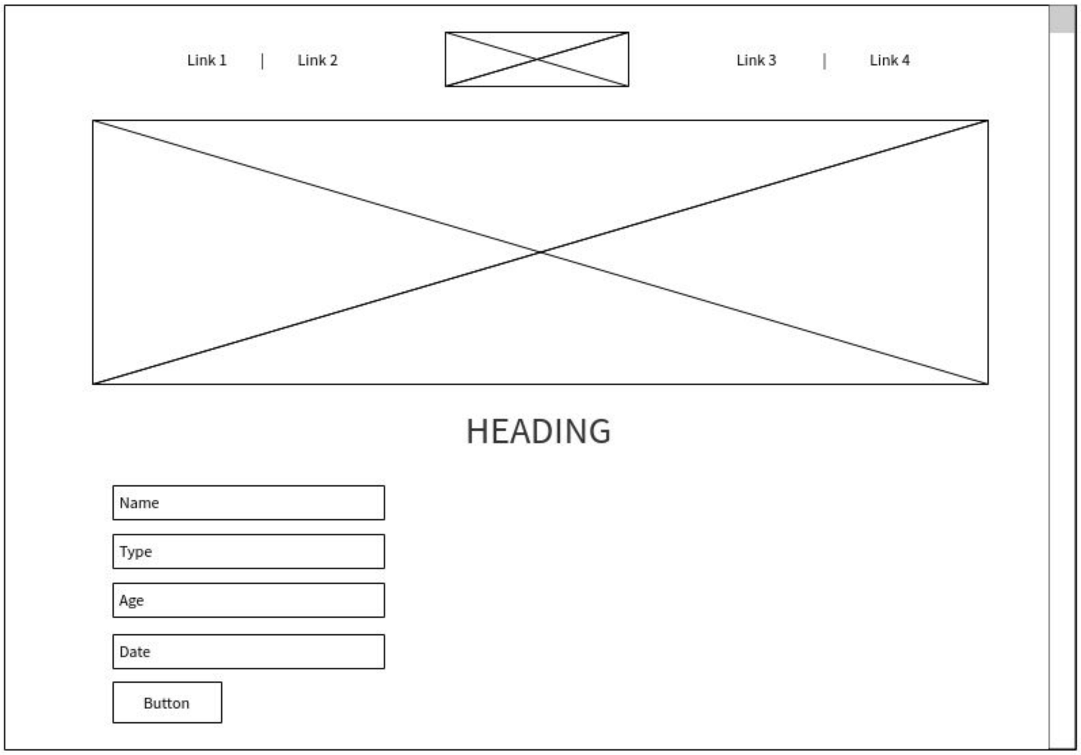
P6

Use of Wireframes



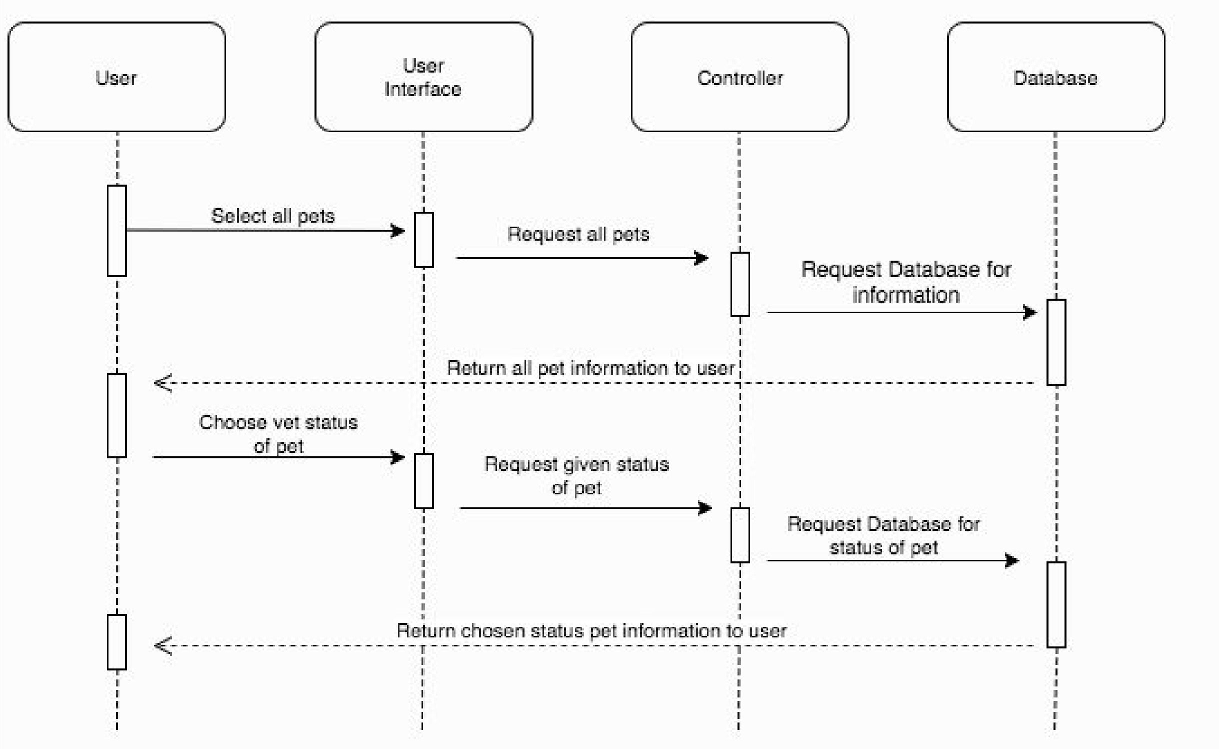
HEADING

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat v

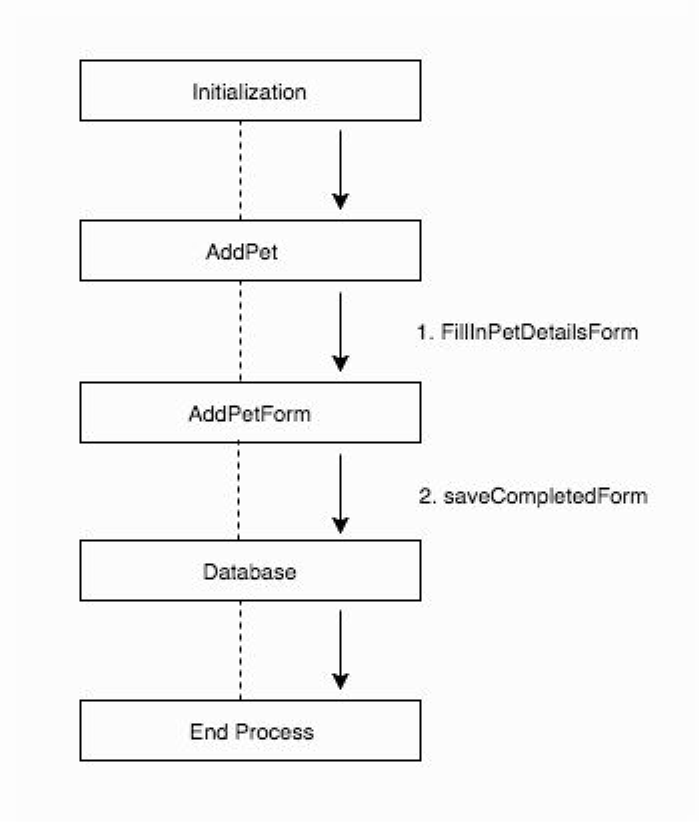


P7

Sequence Diagrams

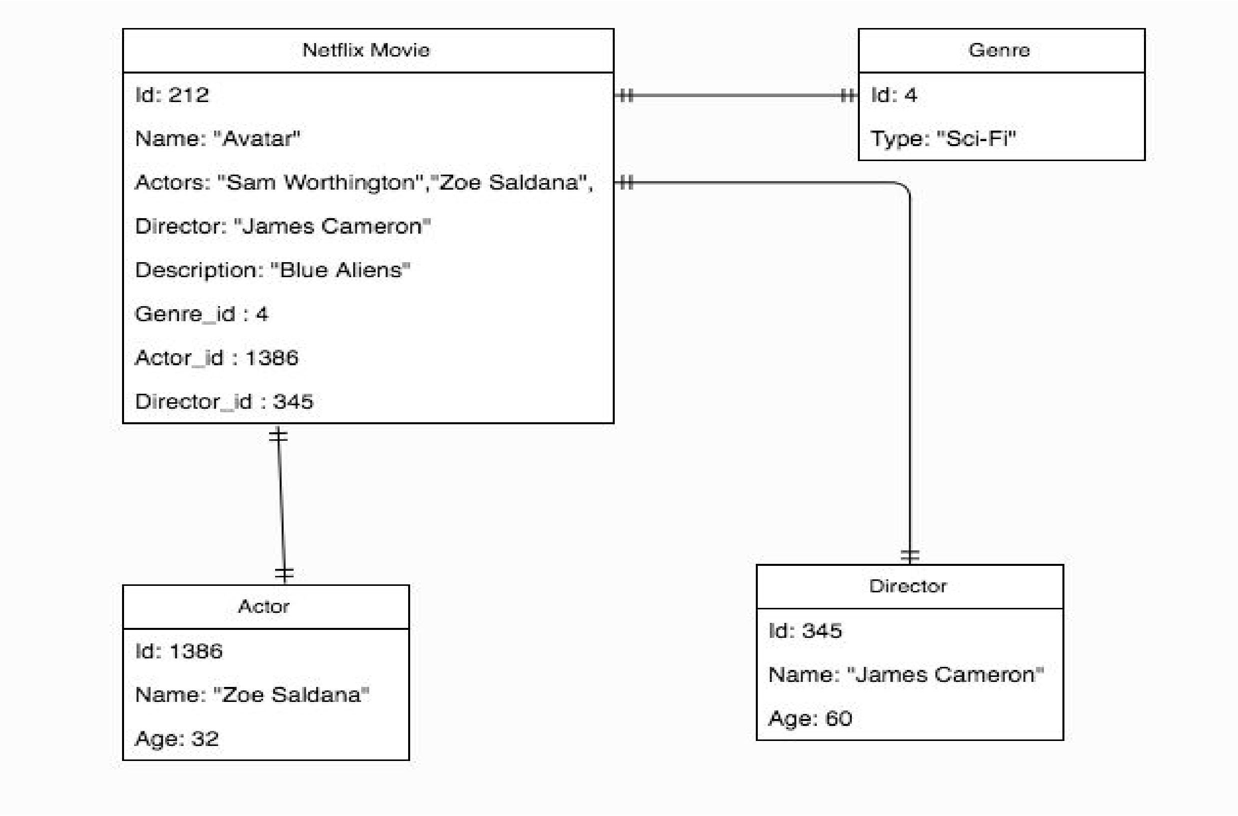

User
User Interface
Controller
Database
Select all pets
Request all pets
Request Database for information
Return all pet information to user
Choose vet status of pet
Request given status of pet
Request Database for status of pet
Return chosen status pet information to user

Collaboration Diagram



P8

Two Object Diagrams



Netflix Movie

Id: 212

Name: "Avatar"

Actors: "Sam Worthington","Zoe Saldana",

Director: "James Cameron"

Description: "Blue Aliens"

Genre_id : 4

Actor_id : 1386

Director_id : 345

Genre

Id: 4

Type: "Sci-Fi"

Actor

Id: 1386

Name: "Zoe Saldana"

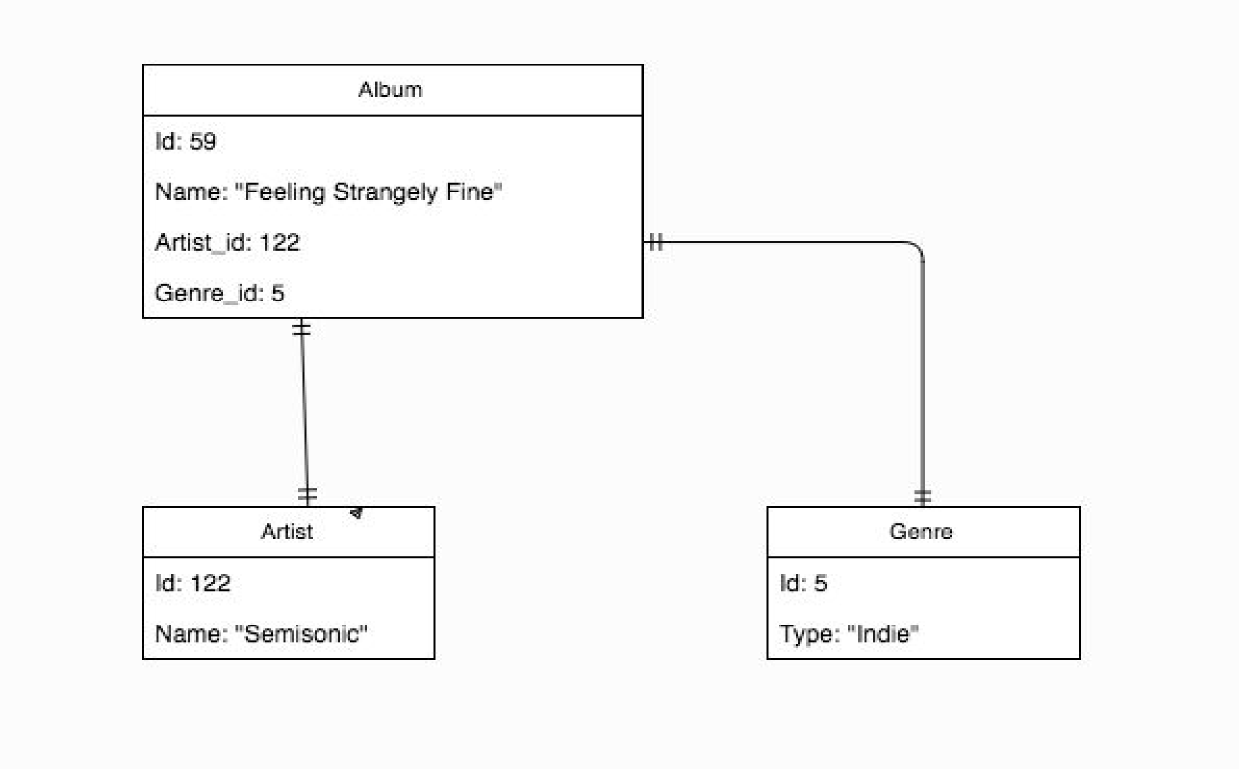
Age: 32

Director

Id: 345

Name: "James Cameron"

Age: 60



Album

Id: 59

Name: "Feeling Strangely Fine"

Artist_id: 122

Genre_id: 5

Artist

Id: 122

Name: "Semisonic"

Genre

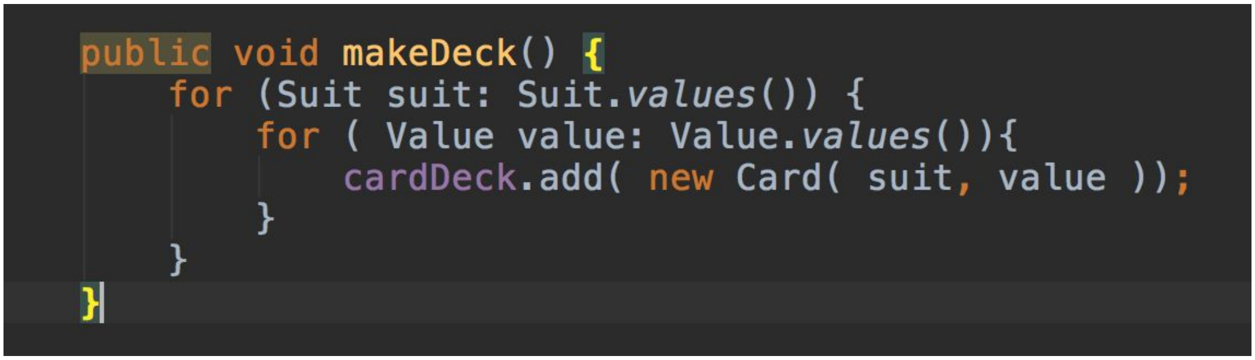
Id: 5

Type: "Indie"

P9

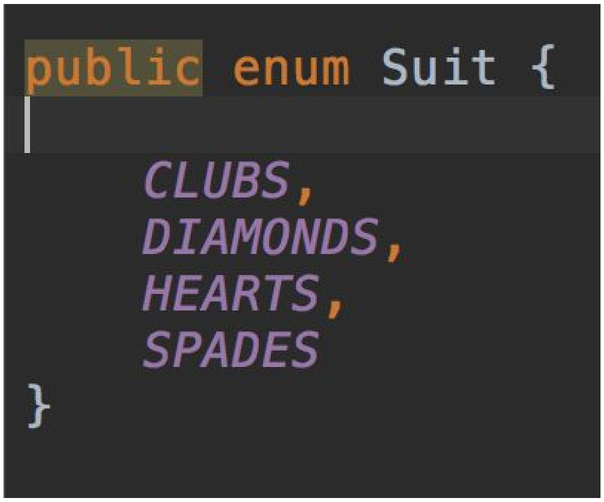
Algorithms implemented for a Black Jack card game app

makeDeck

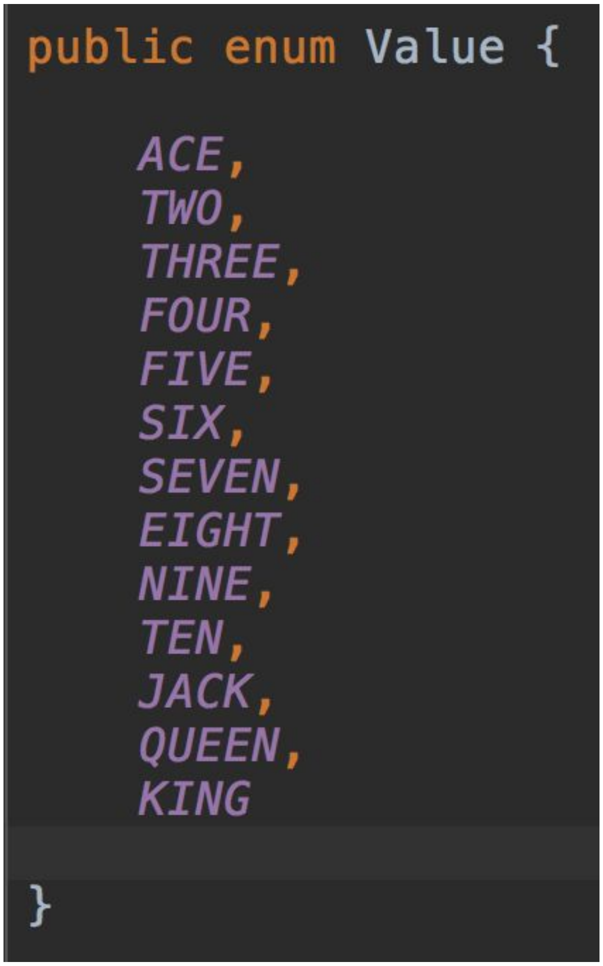


```
public void makeDeck() {
    for (Suit suit: Suit.values()) {
        for ( Value value: Value.values()){
            cardDeck.add( new Card( suit, value ));
        }
    }
}
```

Enums makeDeck uses



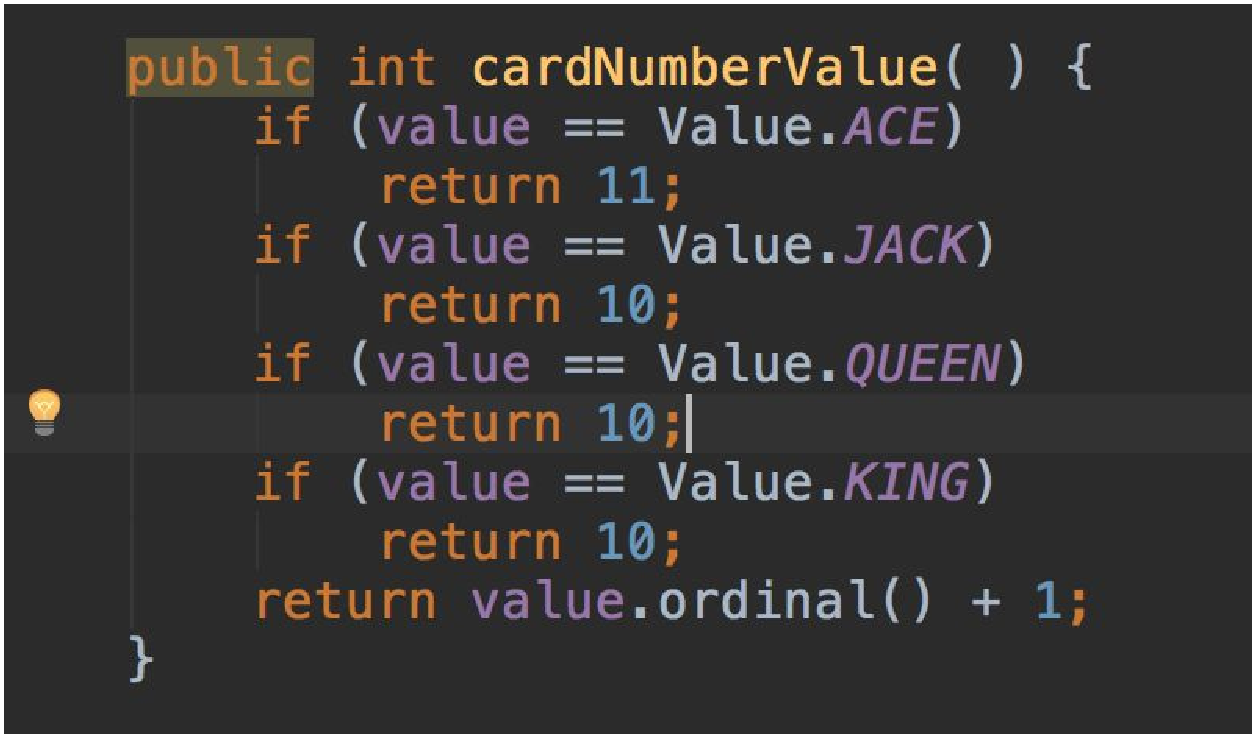
```
public enum Suit {

    CLUBS,
    DIAMONDS,
    HEARTS,
    SPADES
}
```



```
public enum Value {

    ACE,
    TWO,
    THREE,
    FOUR,
    FIVE,
    SIX,
    SEVEN,
    EIGHT,
    NINE,
    TEN,
    JACK,
    QUEEN,
    KING

}
```

This makeDeck algorithm was written to create a card deck from two sets of enums named Suit and Value. It firstly loops around the Suit enum to grab a card suit of either Clubs, Diamonds, Hearts or Spades. Once a suit is selected it will then loop through the Value enum and select a value between a low Ace and the high King. It will then pair these selected Value enum with the selected Suit enum and add them as a Card to the card deck. It will then repeat this process another 51 times to create a deck of 52 cards.

cardNumberValue

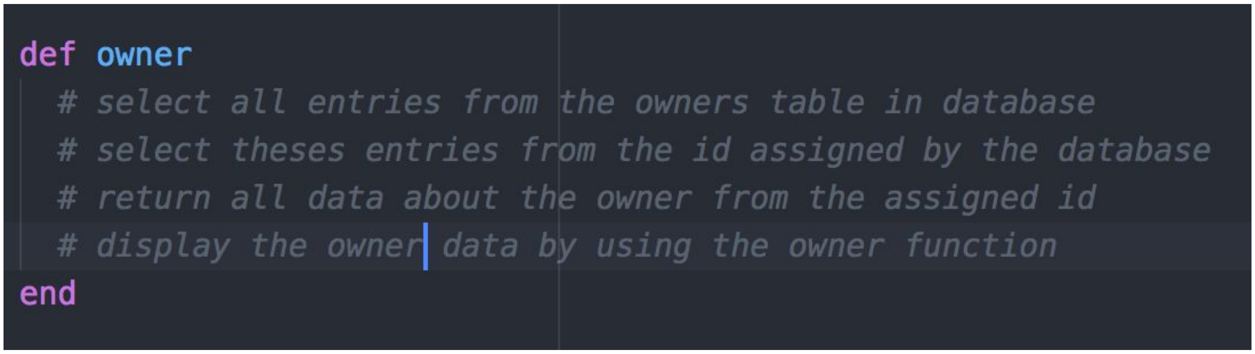


```
public int cardNumberValue( ) {
    if (value == Value.ACE)
        return 11;
    if (value == Value.JACK)
        return 10;
    if (value == Value.QUEEN)
        return 10;
    if (value == Value.KING)
        return 10;
    return value.ordinal() + 1;
}
```

The cardNumberValue algorithm is used to help assign a specific value to certain cards from the Value enum. Through the enum Ace would be assigned the value of 1, Jack would be 11, Queen would be 12 and King would be 13. For the purposes of the black jack card game Ace needs to equal 11 and Jack, Queen and King need to equal 10. So this algorithm takes the value of these cards and reassigns them to the appropriate value needed for the game. The value.ordinal() + 1 is there to add 1 to the initial values from the enum as they start from 0 to 12.

P10

Use of Pseudocode

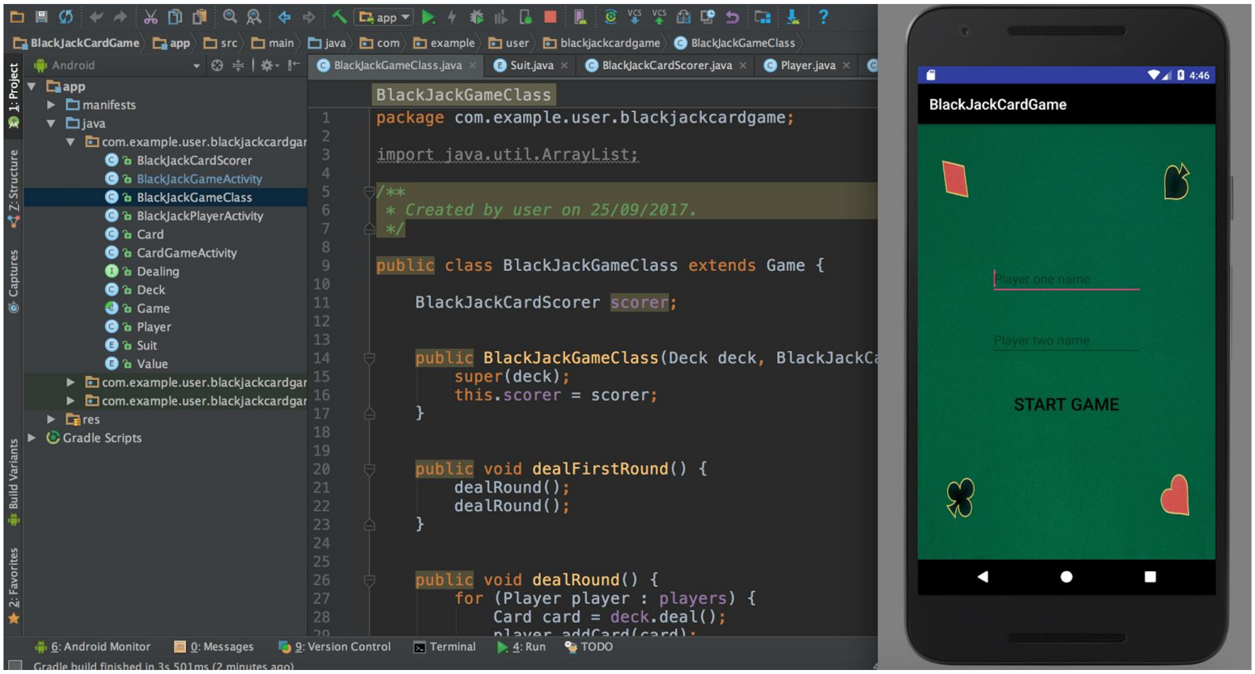


```
def owner
  # select all entries from the owners table in database
  # select theses entries from the id assigned by the database
  # return all data about the owner from the assigned id
  # display the owner data by using the owner function
end
```

P11

A solo project I have worked on

[*https://github.com/patty-cc/BlackJackCardGame*](https://github.com/patty-cc/BlackJackCardGame)



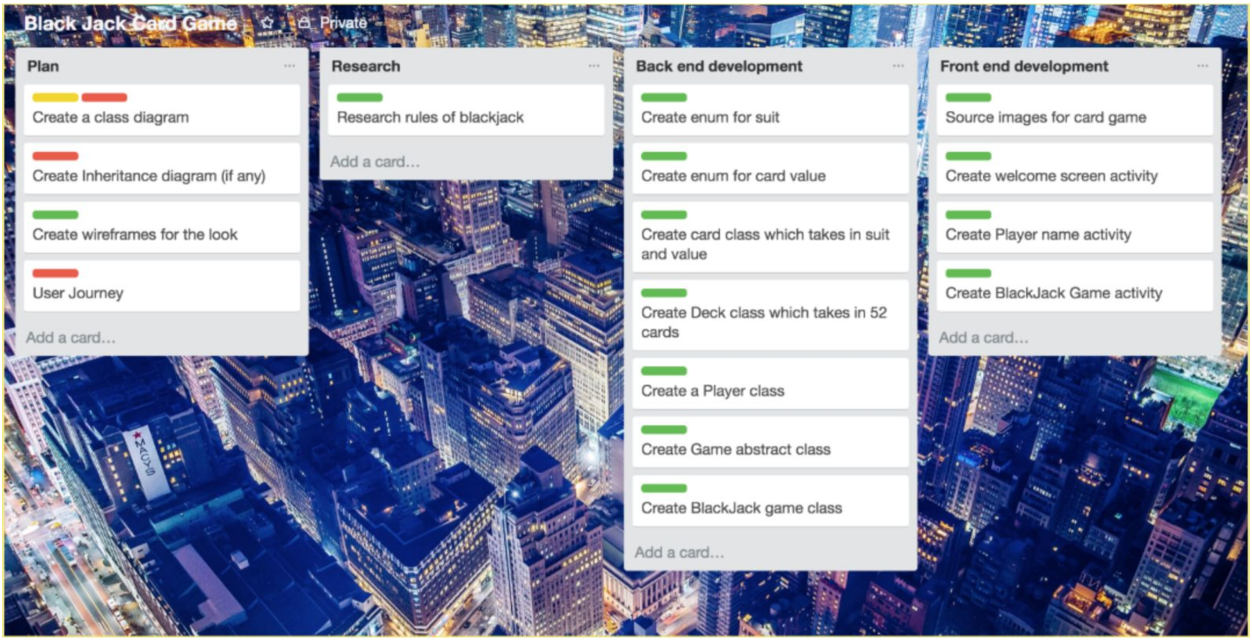
BlackJackGameClass

```
package com.example.user.blackjackcardgame;

import java.util.ArrayList;

/**
 * Created by user on 25/09/2017.
 */

public class BlackJackGameClass extends Game {

    BlackJackCardScorer scorer;


    public BlackJackGameClass(Deck deck, BlackJackCa
        super(deck);
        this.scorer = scorer;
    }


    public void dealFirstRound() {
        dealRound();
        dealRound();
    }


    public void dealRound() {
        for (Player player : players) {
            Card card = deck.deal();
            player.addCard(card);
```

Gradle build finished in 3s 501ms (2 minutes ago)

P12

Screenshots of planning



Black Jack Card Game ☆ Private

Plan

- 🟨🟥 Create a class diagram
- 🟥 Create Inheritance diagram (if any)
- 🟩 Create wireframes for the look
- 🟥 User Journey

Add a card…

Research

- 🟩 Research rules of blackjack

Add a card…

Back end development

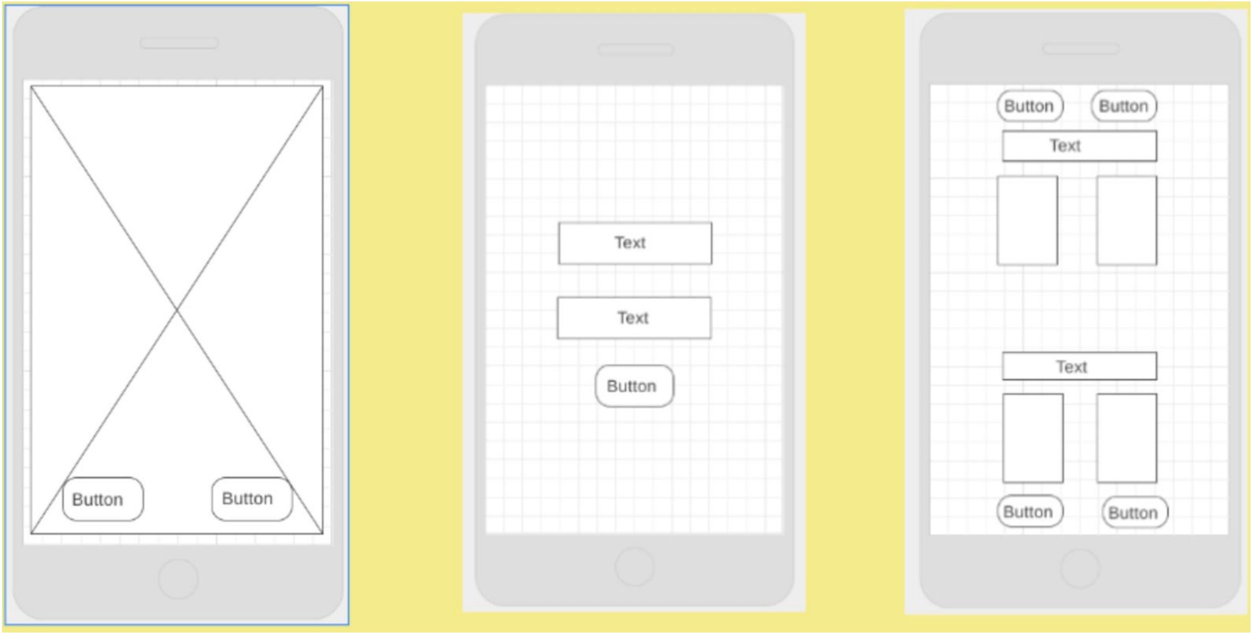
- 🟩 Create enum for suit
- 🟩 Create enum for card value
- 🟩 Create card class which takes in suit and value
- 🟩 Create Deck class which takes in 52 cards
- 🟩 Create a Player class
- 🟩 Create Game abstract class
- 🟩 Create BlackJack game class

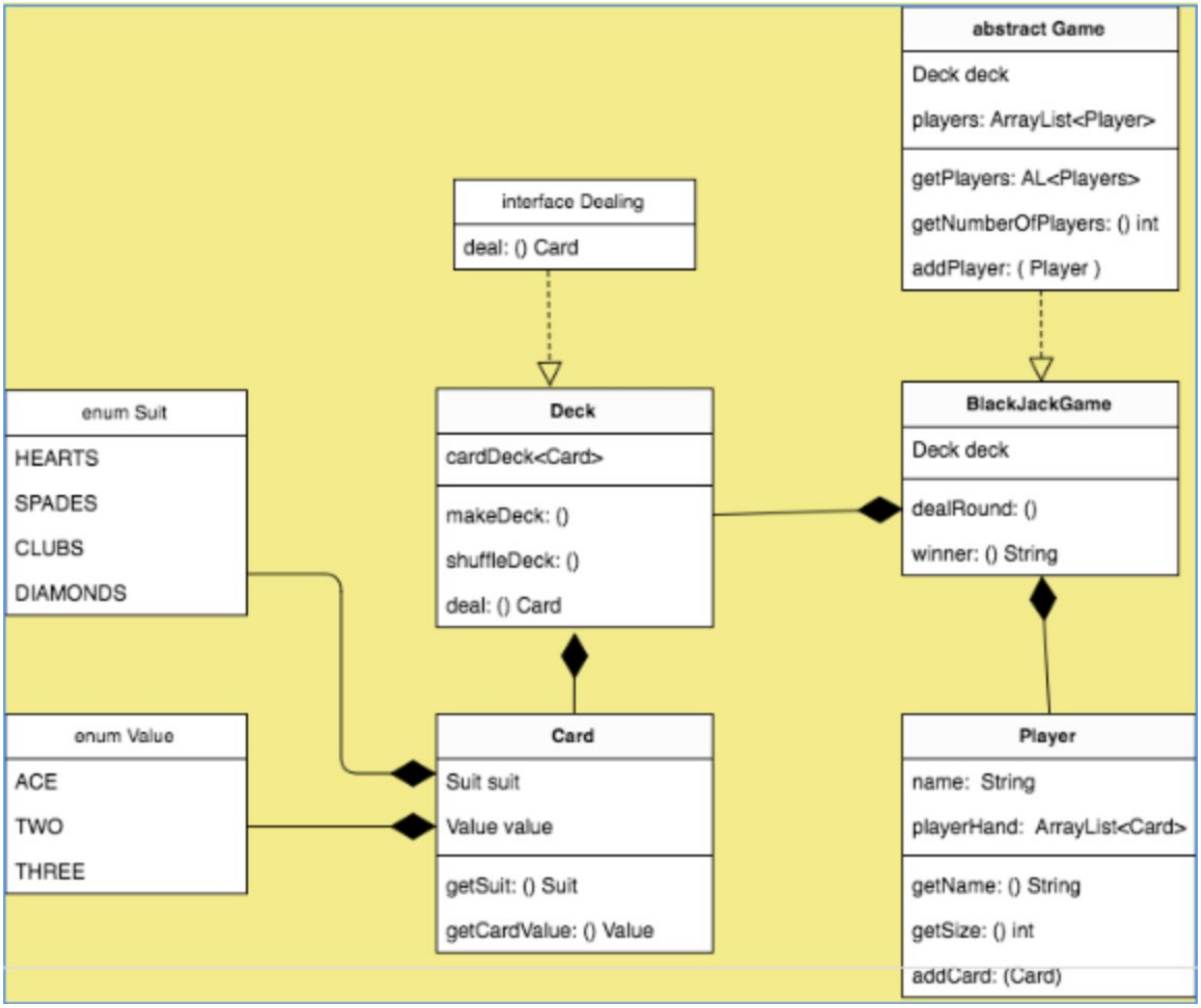
Add a card…

Front end development

- 🟩 Source images for card game
- 🟩 Create welcome screen activity
- 🟩 Create Player name activity
- 🟩 Create BlackJack Game activity

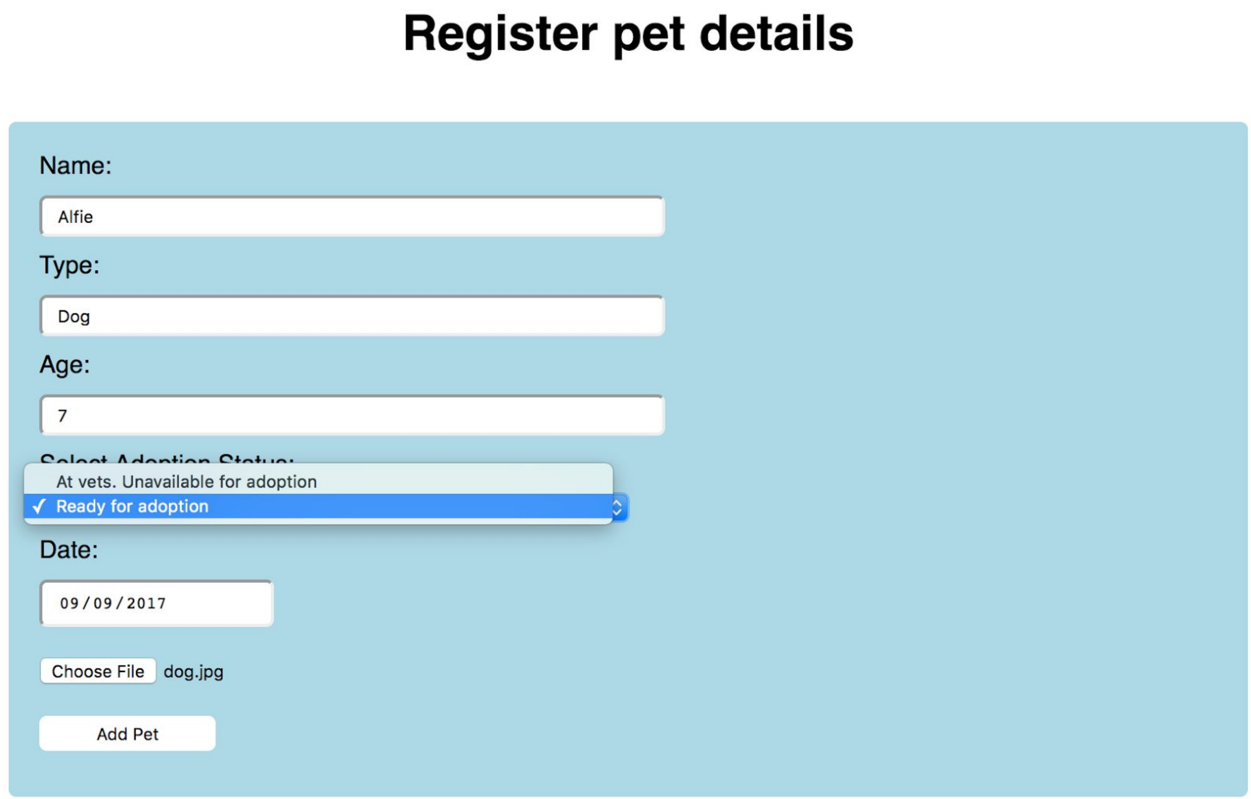
Add a card…





P13

User input being processed



Register pet details

Name:

Alfie

Type:

Dog

Age:

7

Select Adoption Status:

At vets. Unavailable for adoption

✓ Ready for adoption

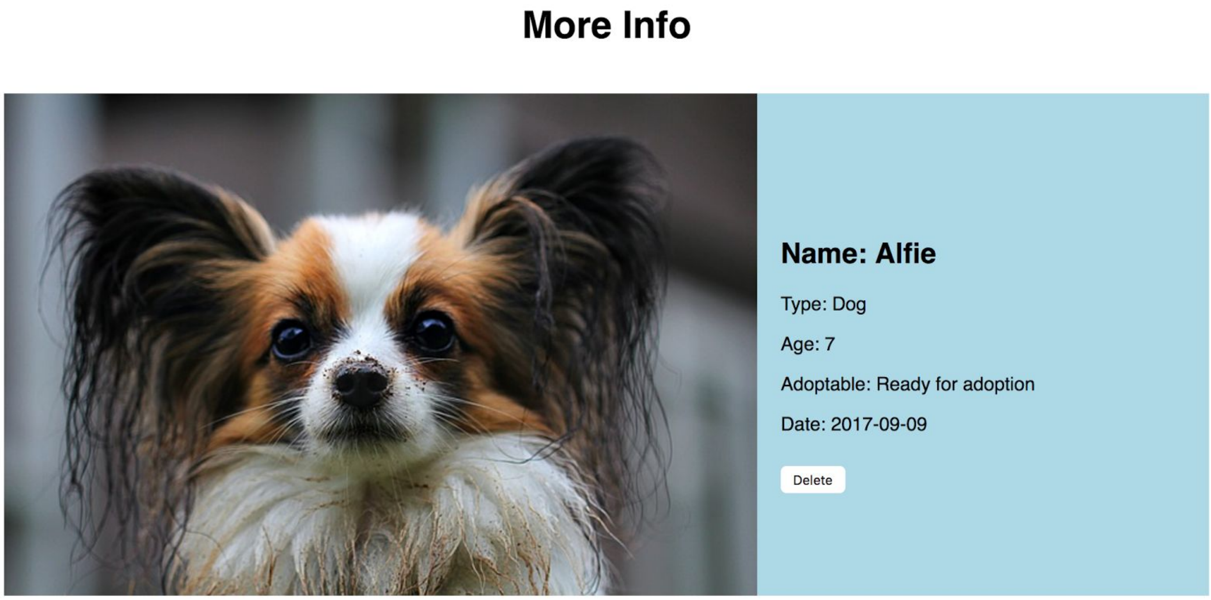
Date:

09/09/2017

Choose File dog.jpg

Add Pet

User input being used



More Info

Name: Alfie

Type: Dog

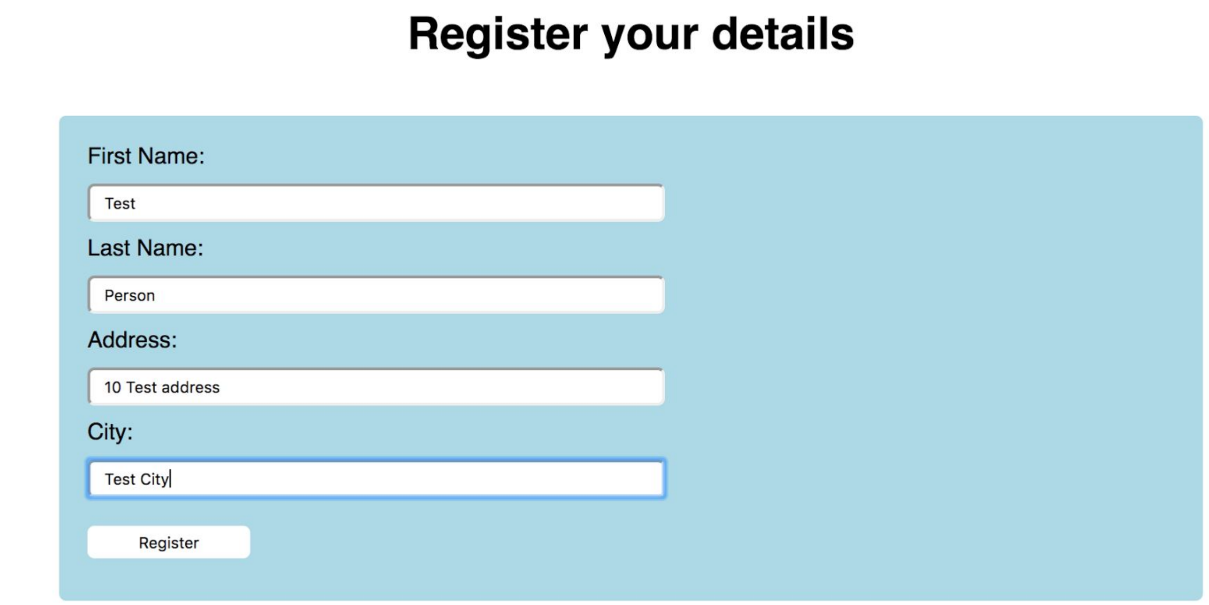
Age: 7

Adoptable: Ready for adoption

Date: 2017-09-09

P14

Interaction with data persistence



Register your details

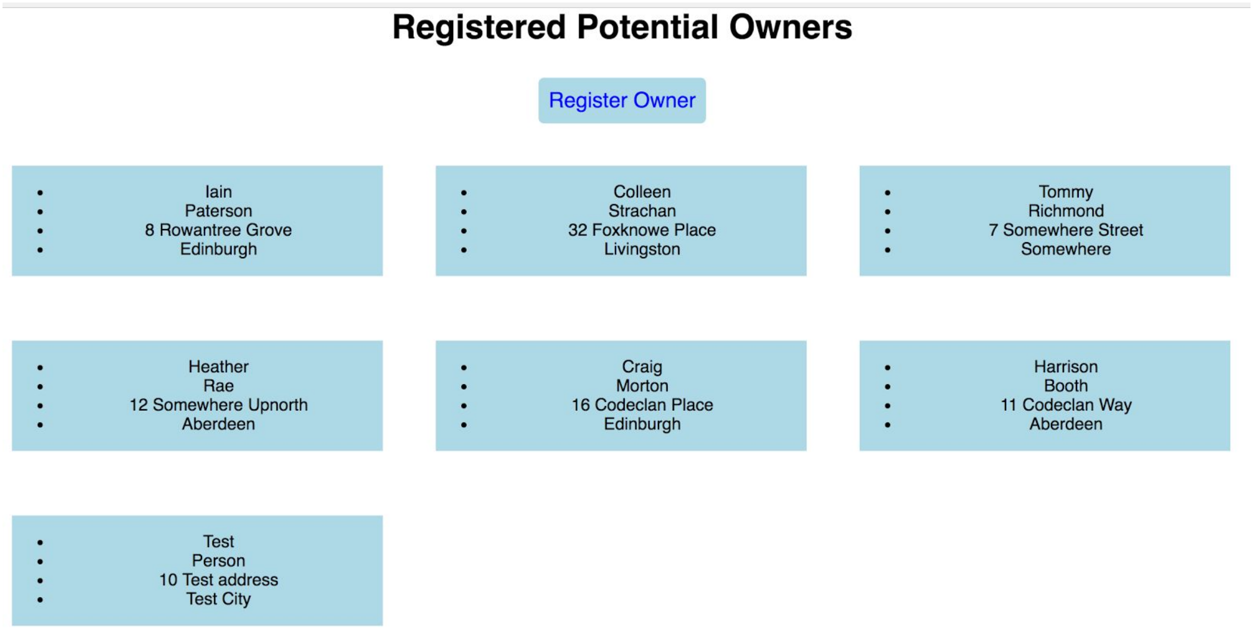
First Name: Test

Last Name: Person

Address: 10 Test address

City: Test City

Conformation of data being saved



Registered Potential Owners

- Iain
- Paterson
- 8 Rowantree Grove
- Edinburgh

- Colleen
- Strachan
- 32 Foxknowe Place
- Livingston

- Tommy
- Richmond
- 7 Somewhere Street
- Somewhere

- Heather
- Rae
- 12 Somewhere Upnorth
- Aberdeen

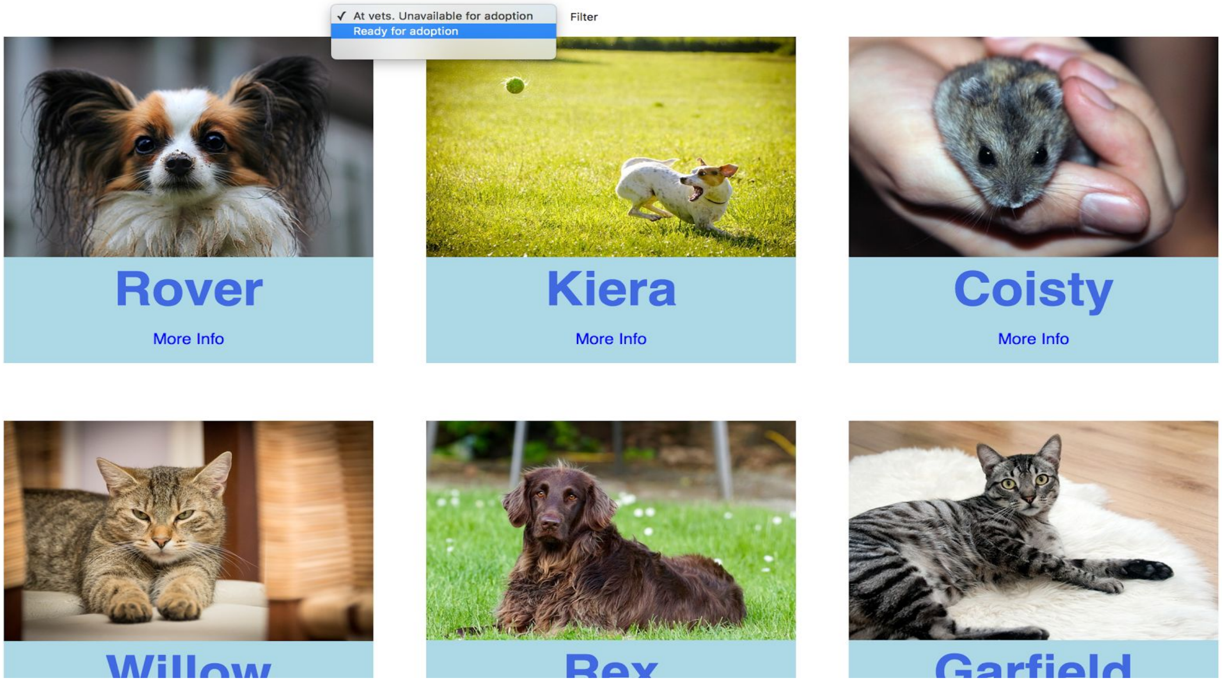
- Craig
- Morton
- 16 Codeclan Place
- Edinburgh

- Harrison
- Booth
- 11 Codeclan Way
- Aberdeen

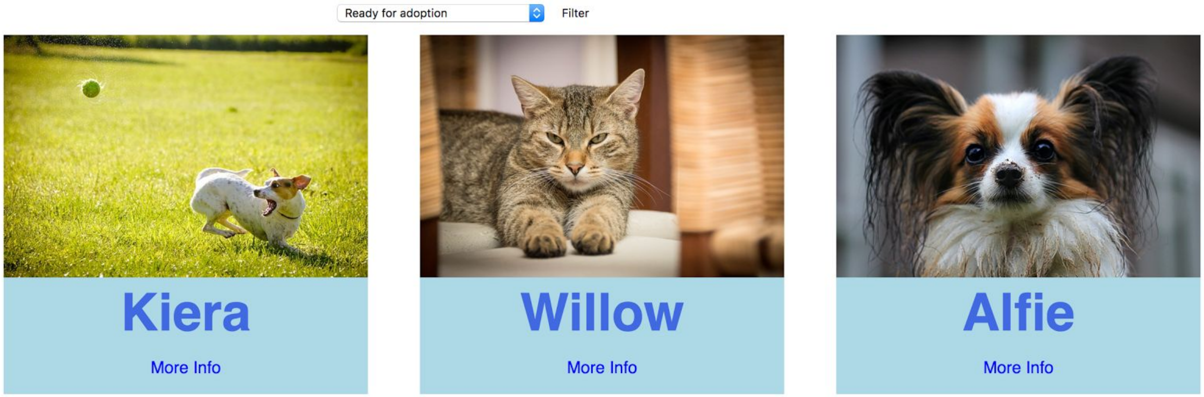
- Test
- Person
- 10 Test address
- Test City

P15

The user requesting information, picking ready for adoption.



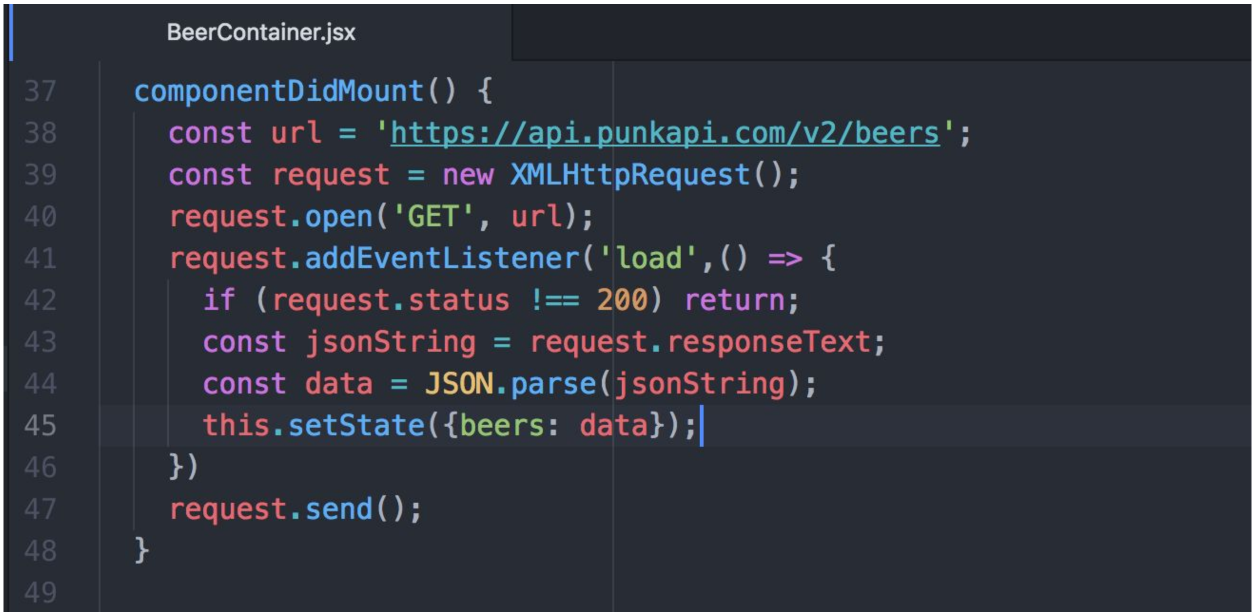
The user request being processed



P16

An API being used within a program

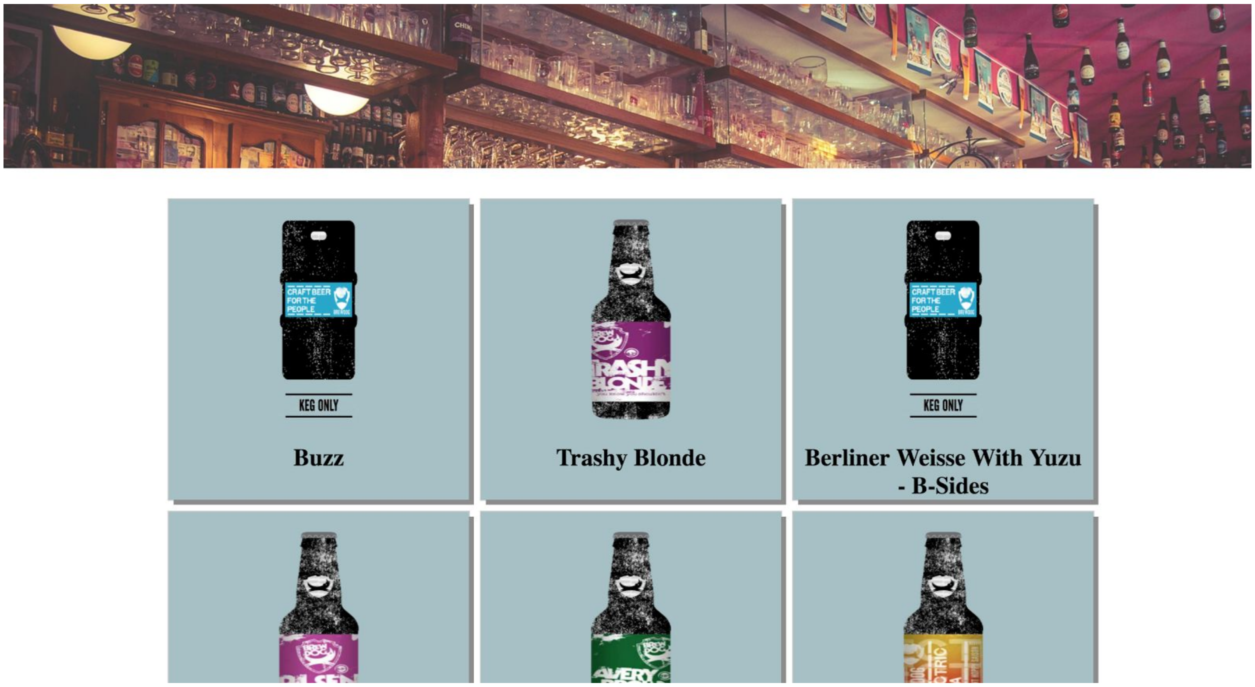
The code that uses the API



BeerContainer.jsx

```
  componentDidMount() {
    const url = 'https://api.punkapi.com/v2/beers';
    const request = new XMLHttpRequest();
    request.open('GET', url);
    request.addEventListener('load',() => {
      if (request.status !== 200) return;
      const jsonString = request.responseText;
      const data = JSON.parse(jsonString);
      this.setState({beers: data});
    })
    request.send();
  }

```

The API being used by the program



P17

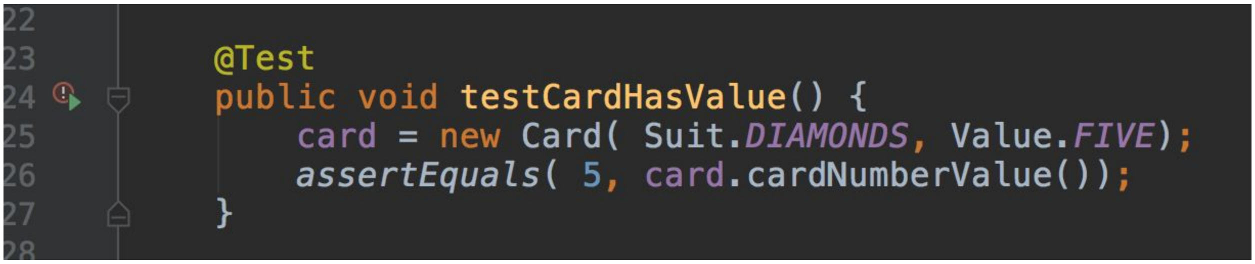
Bug tracking report

Users must be able to able to add adoptee details			PASS
Users must be able to filter between adoptable and non-adoptable pets	FAIL	Implemented logic to backend ruby code to filter between pets adoption status	PASS
Users must be able to access social media pages through social media links	FAIL	Added anchor tags to social media image links	PASS
Users must be able to delete a pet from the database			PASS
Name fields should only accept letter characters and not accept numbers	FAIL	Implemented validation logic to only accept letter characters	PASS

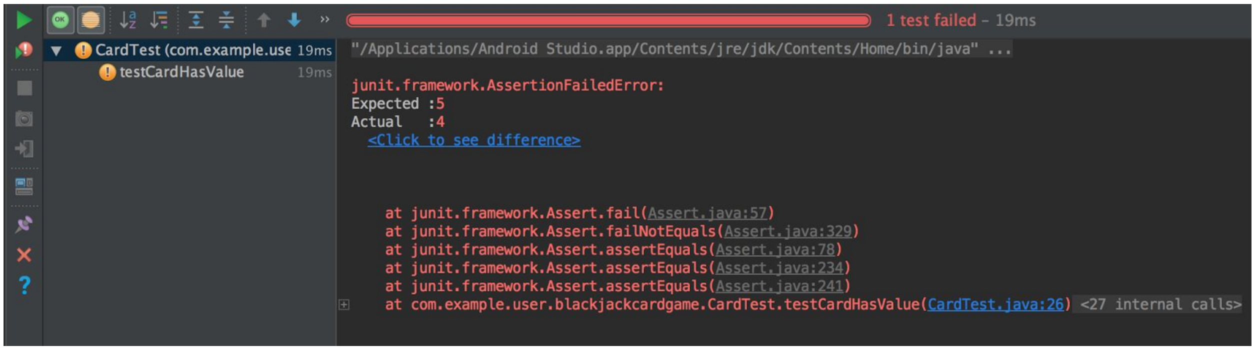
P18

Demonstration of testing

Test one failing to pass



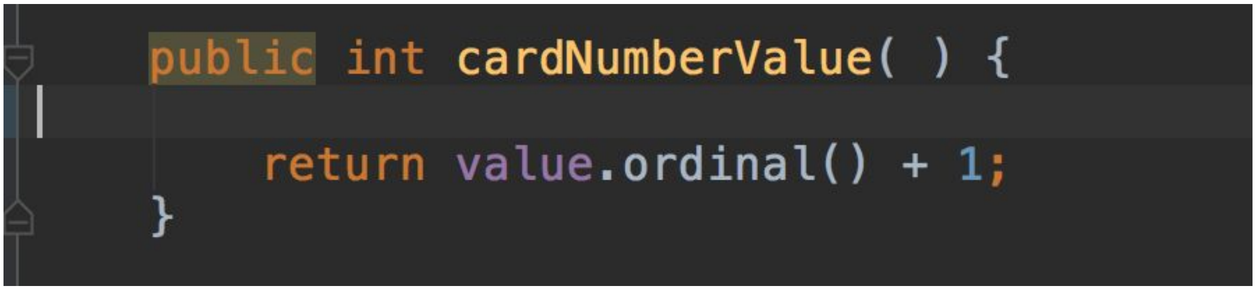
```
    @Test
    public void testCardHasValue() {
        card = new Card( Suit.DIAMONDS, Value.FIVE);
        assertEquals( 5, card.cardNumberValue());
    }
```



```
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

junit.framework.AssertionFailedError:
Expected :5
Actual   :4
  <Click to see difference>


    at junit.framework.Assert.fail(Assert.java:57)
    at junit.framework.Assert.failNotEquals(Assert.java:329)
    at junit.framework.Assert.assertEquals(Assert.java:78)
    at junit.framework.Assert.assertEquals(Assert.java:234)
    at junit.framework.Assert.assertEquals(Assert.java:241)
    at com.example.user.blackjackcardgame.CardTest.testCardHasValue(CardTest.java:26) <27 internal calls>
```

The test is expecting the value of FIVE to be 5 but because the Value enum starts from the point of 0 FIVE currently equates to 4 so the test will fail.

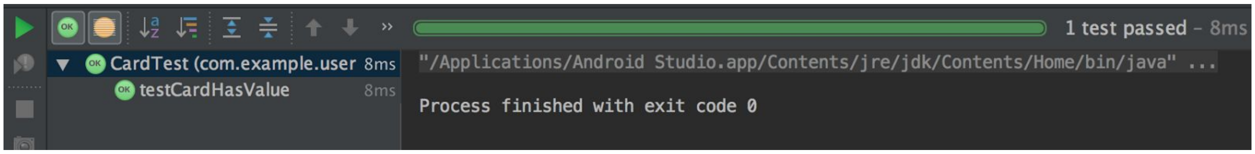
Added code to make test pass



```
    public int cardNumberValue( ) {

        return value.ordinal() + 1;
    }
```

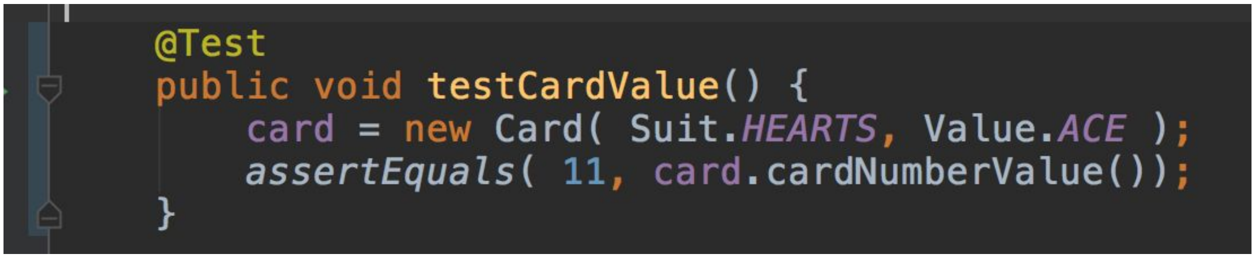
This code will make the enum value start from 1 instead of 0

Test now passing

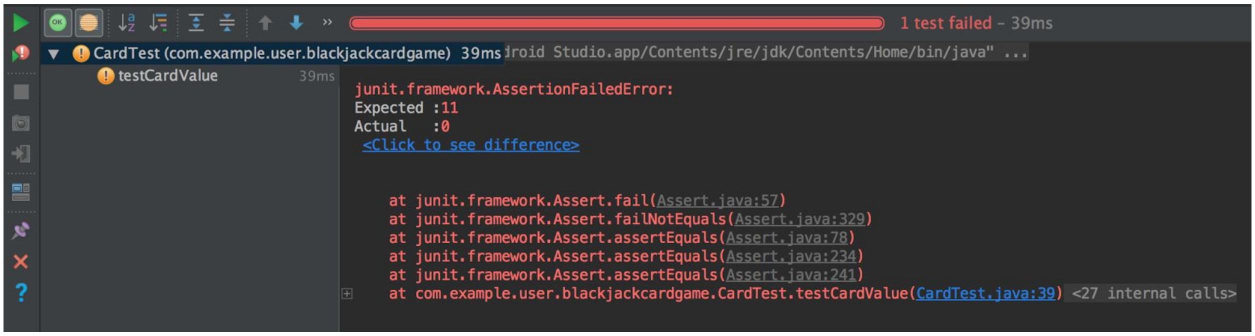


```
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

Process finished with exit code 0
```

Test two failing to pass



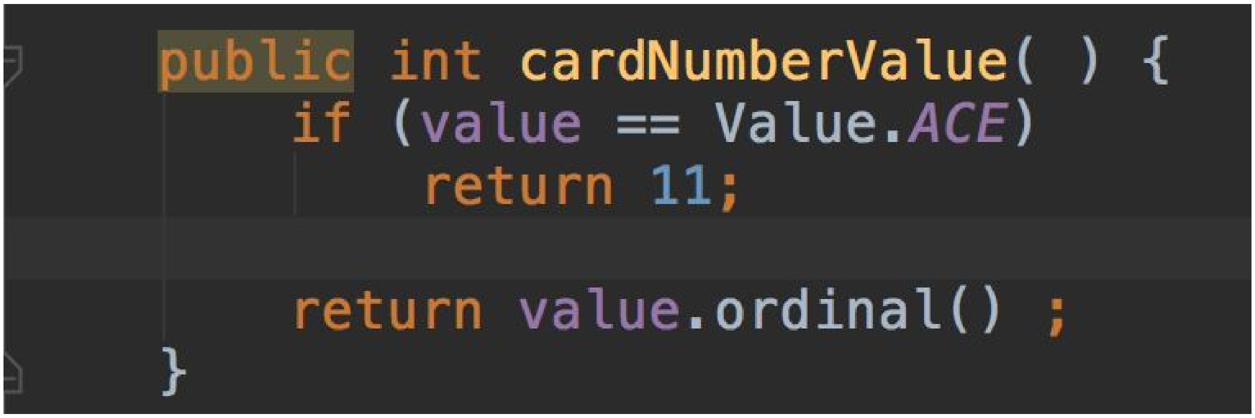
```
@Test
public void testCardValue() {
    card = new Card( Suit.HEARTS, Value.ACE );
    assertEquals( 11, card.cardNumberValue());
}
```



```
droid Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

junit.framework.AssertionFailedError:
Expected :11
Actual   :0
 <Click to see difference>


    at junit.framework.Assert.fail(Assert.java:57)
    at junit.framework.Assert.failNotEquals(Assert.java:329)
    at junit.framework.Assert.assertEquals(Assert.java:78)
    at junit.framework.Assert.assertEquals(Assert.java:234)
    at junit.framework.Assert.assertEquals(Assert.java:241)
    at com.example.user.blackjackcardgame.CardTest.testCardValue(CardTest.java:39) <27 internal calls>
```

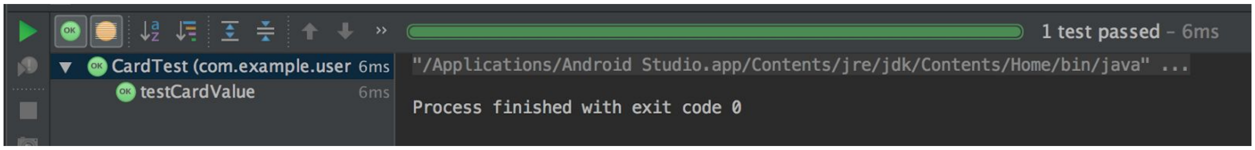
This test is failing as it is expecting the ACE to be of value 11 but currently it has a value of 0

Added code to make test pass



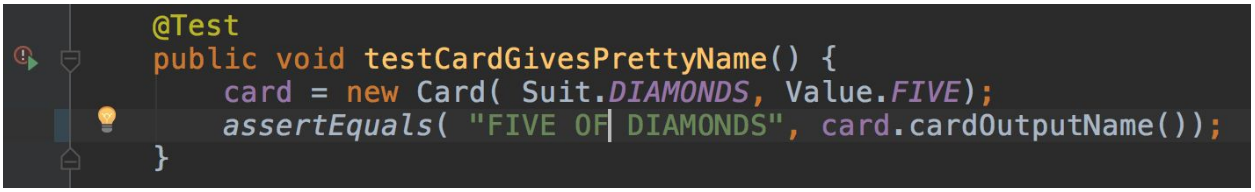
```
public int cardNumberValue( ) {
    if (value == Value.ACE)
      return 11;

    return value.ordinal() ;
}
```

Test now passing

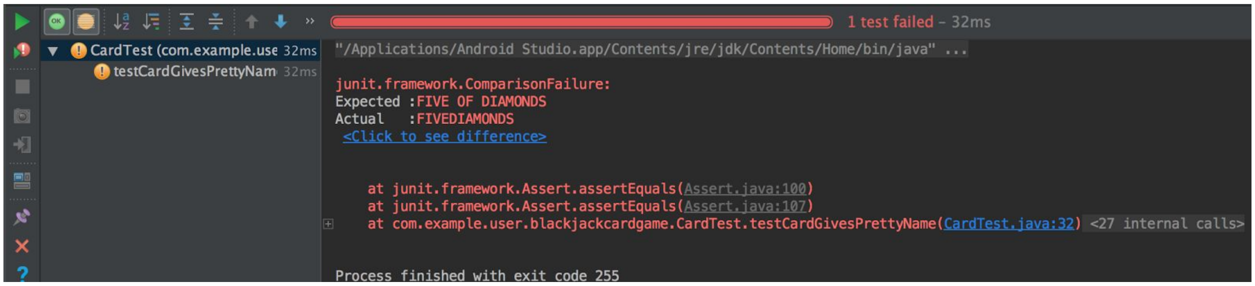


```
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

Process finished with exit code 0
```

Test three failing to pass



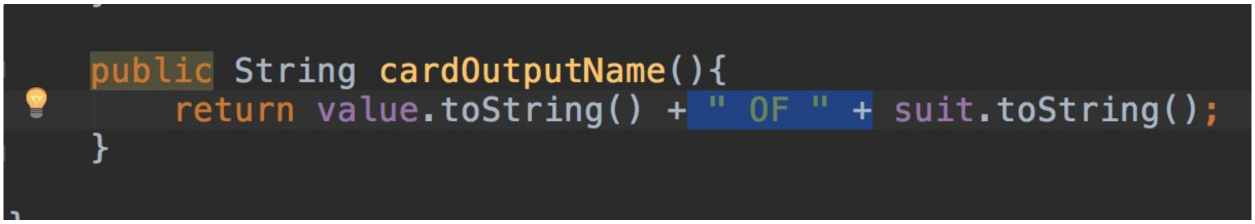
```
@Test
public void testCardGivesPrettyName() {
    card = new Card( Suit.DIAMONDS, Value.FIVE);
    assertEquals( "FIVE OF DIAMONDS", card.cardOutputName());
}
```



```
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

junit.framework.ComparisonFailure:
Expected :FIVE OF DIAMONDS
Actual   :FIVEDIAMONDS
 <Click to see difference>


    at junit.framework.Assert.assertEquals(Assert.java:100)
    at junit.framework.Assert.assertEquals(Assert.java:107)
    at com.example.user.blackjackcardgame.CardTest.testCardGivesPrettyName(CardTest.java:32) <27 internal calls>


Process finished with exit code 255
```

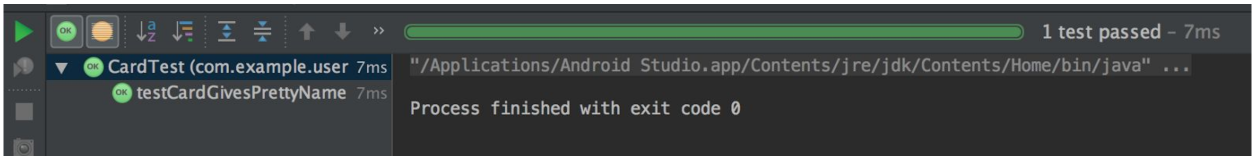
This is test is failing because it is looking for the value of the card to be ‘FIVE OF DIAMONDS’ but currently due to the code it is thinking the card is ‘FIVEDIAMONDS’

Code added to make test pass



```
    public String cardOutputName(){
        return value.toString() + " OF " + suit.toString();
    }
```

Test now passing



```
1 test passed – 7ms

CardTest (com.example.user 7ms
    testCardGivesPrettyName 7ms

"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java" ...

Process finished with exit code 0
```