DoggyData

SIT305 ANDROID & IOS MOBILE PROGRAMMING

Trimester 1, 2020

Leesa Ward 217493588

INTRODUCTION & FEATURES

DoggyData is a pet breed lookup application that allows the user to search for dog and cat breeds by a range of criteria.

The app, and the first target audience, is inspired by a child I met who was carrying a book about cat breeds. When I told her I had rescue cats and I wasn't sure what breeds they were, her eyes lit up and she asked if I could show her photos of them so she could try to work out what breed they might be.

The target audiences of the application are:

- Children interested in learning about dogs and cats
- Adults also interested in learning about dogs or cats, or who are trying to decide upon a breed before getting a pet.

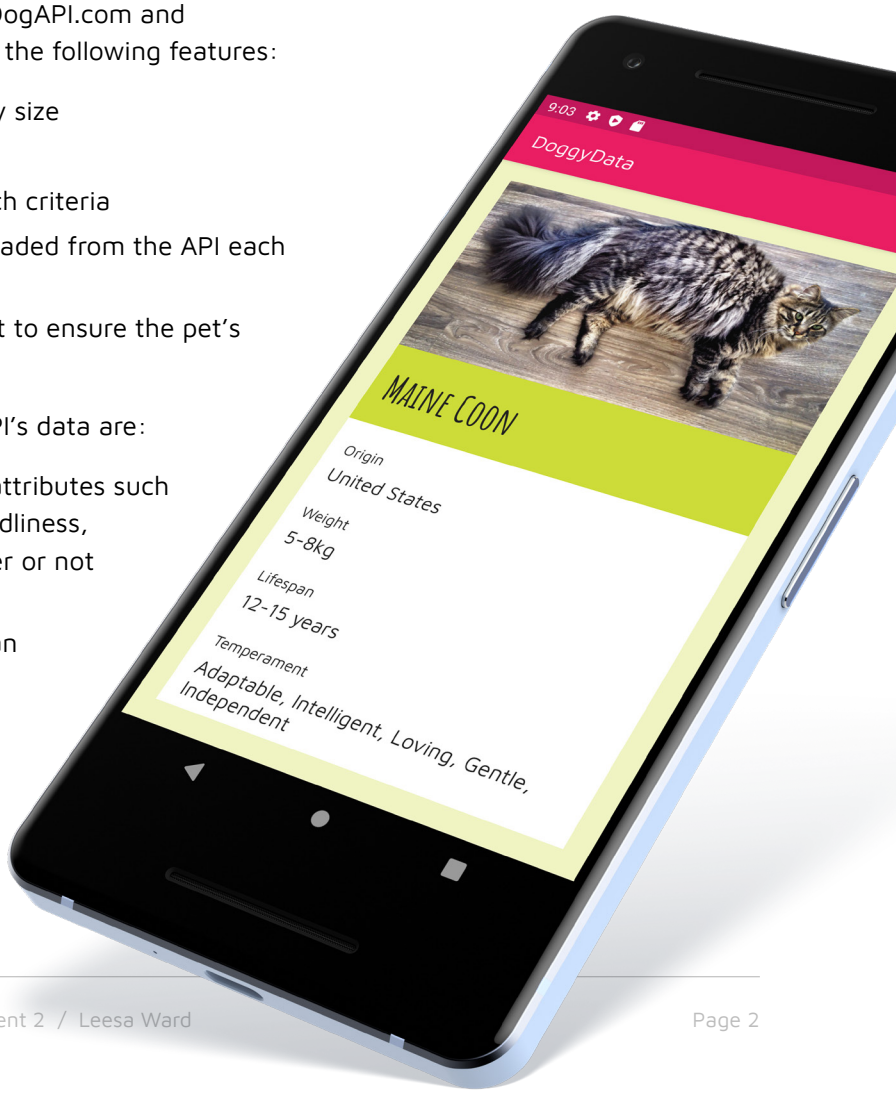
FUNCTIONALITY

DoggyData sources its information from TheDogAPI.com and TheCatAPI.com using the Volley library. It has the following features:

- Use height and weight values to search by size
- Search by temperament
- View details of a breed that met the search criteria
- A random image of the chosen breed is loaded from the API each time a breed detail screen is viewed
- A face detection library is used to attempt to ensure the pet's face is shown in the loaded image.

Future planned features possible using the API's data are:

- Choosing cat breeds based on ratings of attributes such as affection level, adaptability, child-friendliness, dog-friendliness, energy level and whether or not the breed is considered a "lap cat"
- Choosing breeds based on average lifespan (this information is displayed in the detail screen, but search for it is not yet implemented)
- Development of an algorithm to use height and weight values to group breeds (e.g. small, medium and



large dogs)

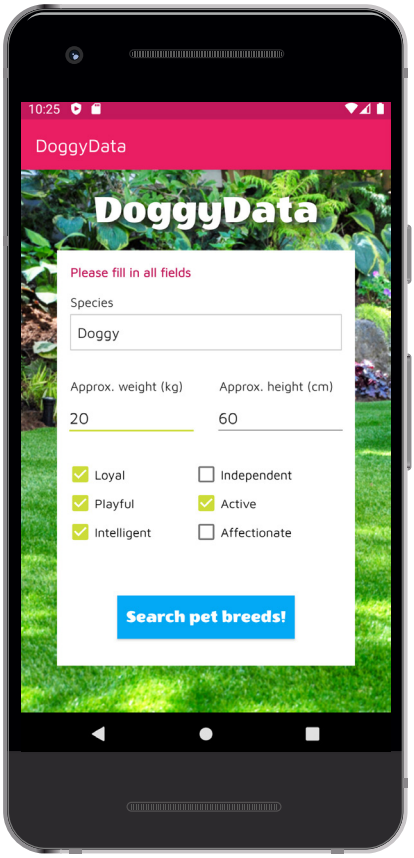
- ✔ Expansion of the temperament search functionality.

The API does not support direct querying in the manner that the application requires, so there is a lot of data processing that takes place with each request. For better efficiency and the ability to add more data (such as coat colour, a key feature a child would use when trying to classify a pet they met), I plan to have the API data inserted into a database (that could be updated with new API data using a simple button within the app) which the app would use instead of directly querying the API.

DESIGN & STRUCTURE

The design of the application is kept deliberately simple, with three screens: The home/search screen, search result screen, and breed detail screen. Clicking the native *back* button takes the user back to where they were previously - i.e. to go from the detail screen back to the search results screen, or the search results screen back to the home/search screen.

The design adopts a bright and colourful colour scheme with round and friendly fonts to appeal to children.



HOME/SEARCH SCREEN

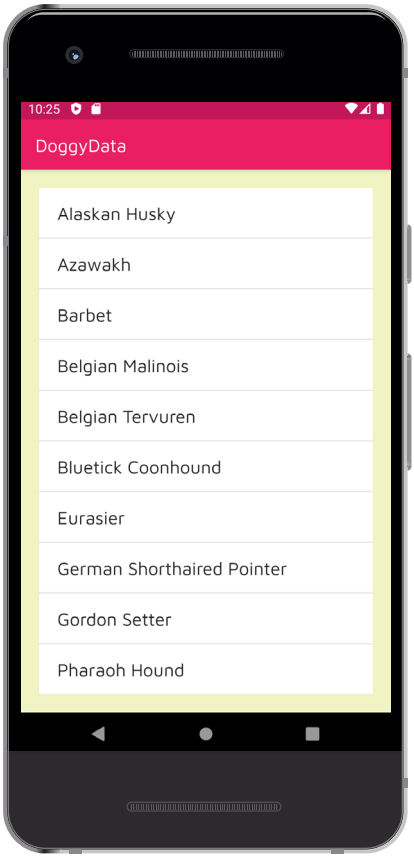
The app is designed with a single activity with multiple fragments. The Main Activity is the search screen that the user sees when the app is loaded. It also uses a utility class to check that an internet connection is available and log an error if there isn't.

Layout files:

- activity_main.xml

Java classes:

- MainActivity.java
- ConnectionCheck.java



SEARCH RESULT SCREEN

The search result screen, or breed list fragment, is where all the action happens. It makes API requests using the Volley library, processing the returned data, creating pet objects (Dog and Cat subclasses), and saving those objects to a list if they meet the search criteria.

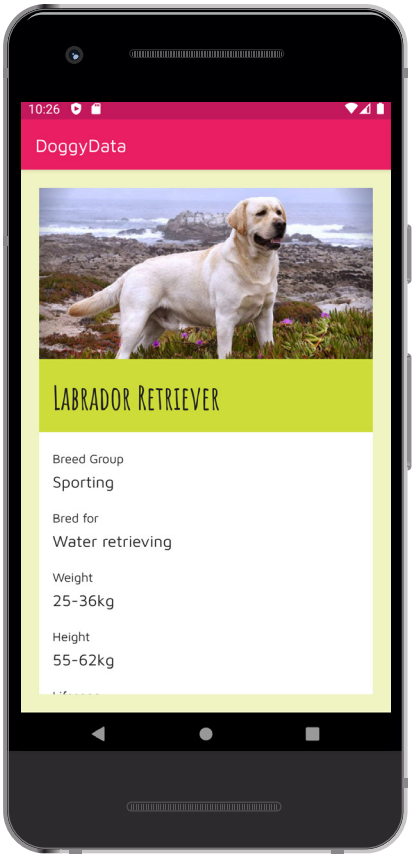
This fragment contains a RecyclerView which loads in the pets from that list using the breed list item adapter, and passes the pet object to the breed detail fragment (see below).

Layout files:

- fragment_breedlist.xml
- listitem_breed.xml

Java classes:

- BreedListFragmentLoader.java
- BreedListFragment.java
- BreedListItemAdapter.java
- Dog.java
- Cat.java
- Pet.java



BREED DETAIL SCREEN

The breed detail fragment receives a pet object from the breed list fragment and uses it to display a profile for that breed. The details are contained within a ScrollView to ensure the user can access all details regardless of the height of their device, by scrolling in that region.

Layout files:

- fragment_breeddetail.xml

Java classes:

- BreedDetailFragment.java
- Dog.java
- Cat.java
- Pet.java