CPEN 475/575

Project 4

A multithreaded networked application that uses ViewPager2 and RecyclerView.Adapter

Deliverables

Please clean your Android Studio project, then zip the containing directory and submit.

Gradle Info

compileSdkVersion 30
minSdkVersion 24
targetSdkVersion 30

Testing

I will compile and test your solution on a Google Pixel running API 30.

Overview:

This is a port of project 3 to use a ViewPager2 and RecyclerView.Adapter.
Do not worry about screen rotations for this project.

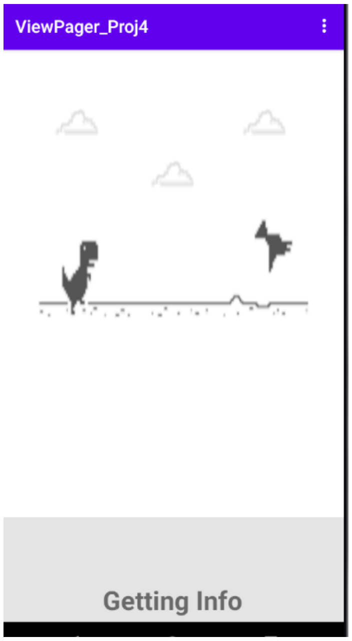
Features

Create an application that has/does the following.

1. Please create an application with an appbar.
2. Please embed a ViewPager2 in the MainActivity for this project. Please use an Adapter that extends RecyclerView.Adapter to populate this ViewPager. Please be sure to use ViewHolders that extends RecyclerView.ViewHolder
3. The appbar should have a overflow icon that leads to settings.
4. Please create a settings screen with one ListPreference. It contains a choice of 2 URLs (see Helpful Bits). These sites are where both the json list of pet information and individual pet pics are stored.
5. Please create a preference change listener on this preference. On preference change, this listener causes the ViewPager2 to repopulate.
6. Please make all network calls on a separate thread. You will have to download both text (JSON) and images.

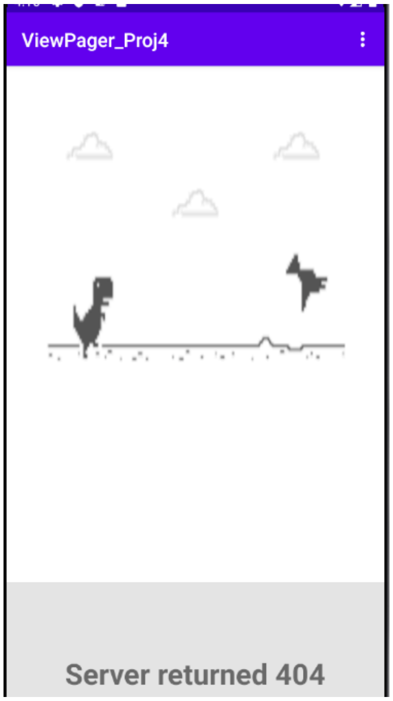
Typical Use

1. On startup the app will download the json file pets.json from the site chosen by the user in the settings (or if initial app install, default to the CNU site). In the interim, while the app is waiting to download the JSON, the app should display a screen similar to the following:



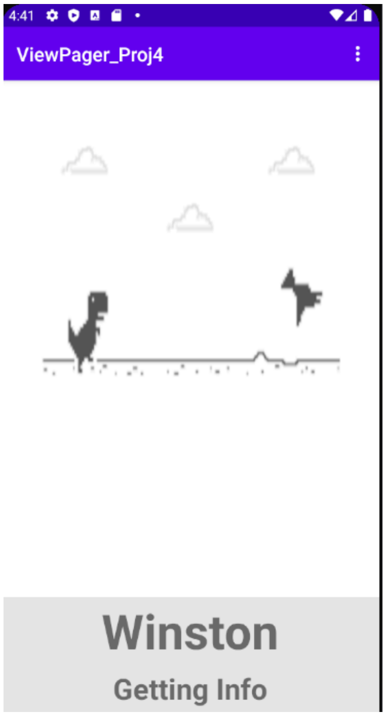
Getting Info

2. If this site cannot be reached (and the Teton Software one cannot) the app should tell the user what the server status code is. This should consist of an appropriate error graphic such as the following.



Server returned 404

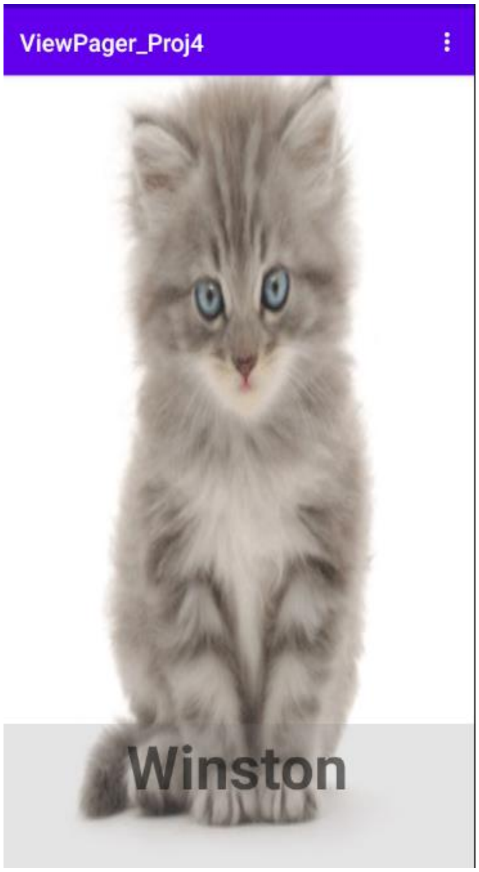
3. Once the app connects and retrieves the JSON it will parse the JSON. The app should then get the first pets name and associated image file from the JSON. The app will then set the first textview to the Pets name and launch a thread to get the Pet image. Note that the second textview indicates that the app is getting information. The UI should look like the following.



Winston

Getting Info

4. Finally, when the image finishes downloading, it will be displayed in the ImageView associated with the currently displayed page. The UI should look like the following.



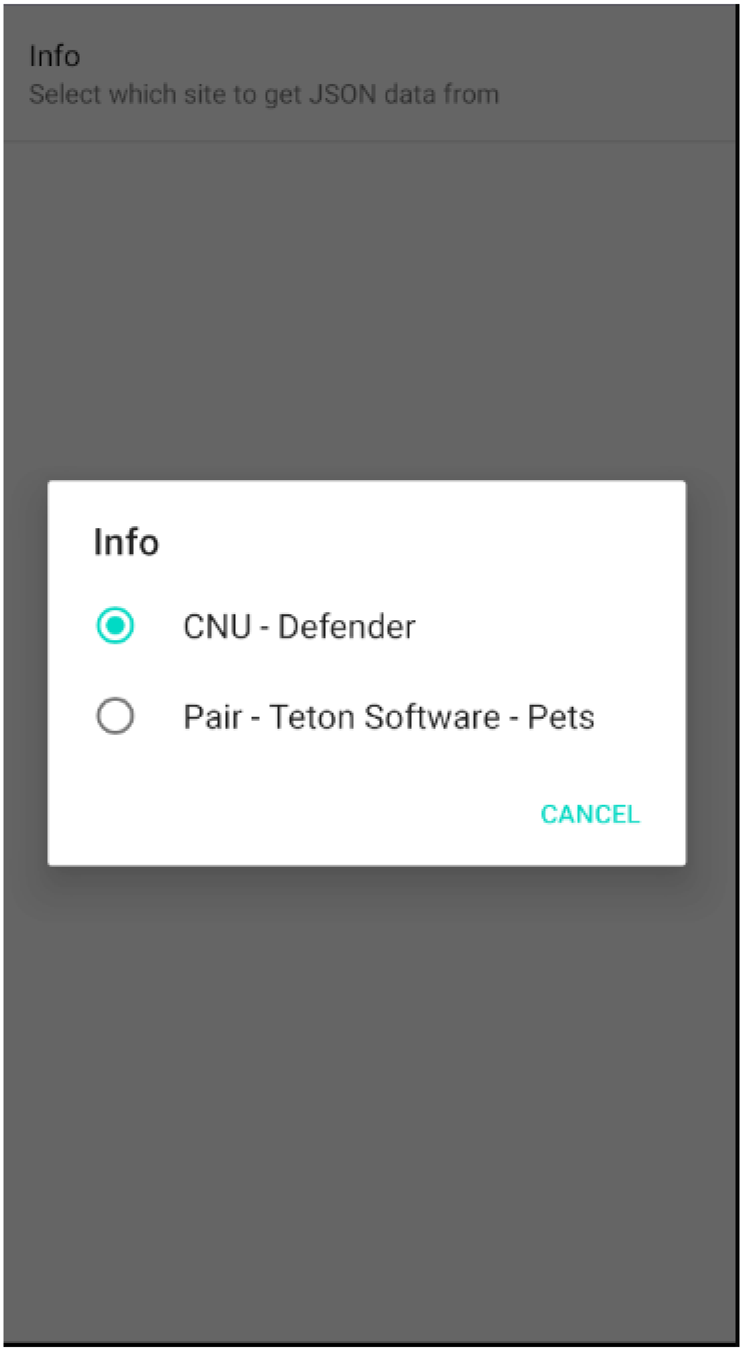
The user should then be able to swipe through the views corresponding to the various pets listed in the JSON file. For each view, the name should be populated from the JSON (downloaded earlier) and a thread should be launched to retrieve the image associated with that name.

Preferences:

If the user clicks the overflow on the appBar (the three vertical dots), and then chooses settings, a setting screen should pop up with a choice of sites to get pet info from. If the user changes the preferred site for info, the app should call

`notifyDataSetChanged() on the current adapter`

to get data from the new site (Effectively starting over on step 1 above).



Info

Select which site to get JSON data from

Info

● CNU - Defender

○ Pair - Teton Software - Pets

Grading

5% gradle files correct

10% Preferences

15% Must use ViewPager2 in XML must have a data adapter that extends RecyclerView.Adapter

10% Must handle case where the bitmap retrieved by an asyncTask is no longer the Bitmap required by the ViewHolder.

15% Correct app behavior on startup

15% Correct thread behavior (first get JSON from the site, then get each image when the user swipes to it)

15% Correct UI

15% Correct behavior when cannot connect

Helpful bits

- Sites where JSON and images are stored

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="JSON_URL_NAME">
        <item>CNU - Defender</item>
        <item>Pair - Teton Software - Pets</item>


    </string-array>
    <string-array name="JSON_URL">
        <item>https://www.pcs.cnu.edu/~kperkins/pets/</item>
        <item>https://www.tetonsoftware.com/pets/</item>

    </string-array>
</resources>to
```

- **How to force UI repopulate and redraw itself**

  call notifyDataSetChanged() on your RecyclerView.Adapter

- **Getting the JSON info**

  The file containing the json data is named pets.json. Download pets.json with this URL.

https://www.pcs.cnu.edu/~kperkins/pets/pets.json

it will return jSON similar to the following (there may be more pets)

```
{
"pets":[
        {
        "name":"Winston",
        "file":"p0.png"
        },
        {
        "name":"Higgens",
p0.png  "file":"p1.png"
        },
        {
        "name":"Broccoli",
        "file":"p2.png"
        }
        ]
}
```