**Test project description:**

# Step-1:

I downloaded test.zip file and sync the gradle. It took a while to finish it.

# Step-2:

WeatherForcast activity created.

Button added to ProfileActivity Page to open WeatherForecast activity when the user click the button. The button text added to ProfileActivity is “Weather Forcasting”

# Step-3:

Added user permission to AndroidManifest.xml.

<uses-permission android:name="android.permission.INTERNET" />

# Step-4:

Layout created and widgets are added to the layout

First I add linear layout with vertical orientation.

Then Imageview add to the layout

Then next to Image view progress bar

Then add four TextView for current temperature, min temperature, max temperature and vu rating.

All configurating are configured

# Step-5:

Inner class ForecastQuery created that extend from AsyncTask<String, Integer, String>. Also create 5 variable one bitmap variable for image and 4 string for temperature and UV rating.

# Step-6:

I checked both the weather and UV rating URL on browser. Weather URL return xml formatted data and UV rating URL return Json formatted data.

# Step-7:

The function protected String doInBackground(String …args) implemented from asyncTask.

I copied the code from GitHub branch “week6\_xml” and pasted to doInBackground(String …args) function.

I changed the starting tags that to “temperature”. Each time, in while loop check if the current tag’s name is “temperature”.

If starting tags is “temperature”, then get the value, min, and max parameters and save the text the variable created in step-5.

# Step-8:

call publishProgress() with 25, 50, 75 as the parameters to show the progress of retrieving the data.

Parse the xml and get icon String and set to String variable called ‘iconName.

# Step-9:

First check the image is exist in the file or not. then if the image exist in the file, it load from the file otherways it download from the web.

Because of there is not clear information about loading UV rating I just did after finishing temprature loading.

# Step-10:

Progress bar update on each progress.

# Step-11:

After excuting all athe loading finally set the result to the textViews and ImageView.

The progress bar set to invisible

# Step-12

I tested and it worked us expected

Thank you!

# 