My Weather App

Implementations (Frontend):

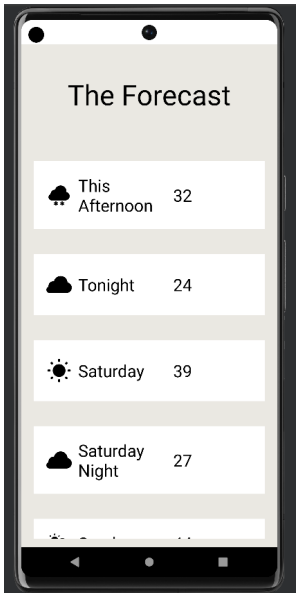
1. Should be implemented as closely as possible to the provided screen. Margins, text sizes, and shades of colors don't have to be precise. ✅

2. Should consist of a textview on top and a recyclerview on the remainder of the screen.

   Every forecast should have a corresponding:

   - icon
   - time of day
   - temperature

   If the Api fails to call then display the failure in the textView ✅



Messages are also displayed on forecast loading and error.

Considerations:

1. Made two tweaks to get the code working

- `String.valueOf(forecasts.get(position).temperature)` instead of `forecasts.get(position).temperature` in RecyclerAdapter.java since `setText` takes string value, not int.
- Changed `holder.weatherIcon.setImageIcon(Icon.createWithContentUri(forecasts.get(position).weatherIcon))` to `holder.weatherIcon.setImageResource(Integer.valueOf(forecasts.get(position).weatherIcon))` since the URI creation was causing an issue.

Implementation (Backend):

When Screen #1's forecast button is pressed the application should initialize the forecast activity, which will call the national weather api for the forecast of a given location. ✅

Code found in: MainActivity.java

Calling the weather api is a two step process.

- The first step is to send a location request. The response will be two grid coordinates. ✅(Pre-written code in Forecast View Model)
- The second step will be to send a forecast request with the previously given grid coordinates. The response will be the finale forecast for the next couple of days.✅(Code found in: Forecast View Model, specifically the `getWeatherProperties() function`)
- After receiving the weekly forecasts load them into a recyclerView.✅(Code found in: ForecastActivity.java)

Final Questions

1. How long did the assessment take to complete?
   The assessment took around 2 days to complete. However, I was working on other things as well and did not complete it in a single stretch of time.
2. Was there any task that took significantly longer than the others, and if so why?
   There were things that took longer than expected. The UI designing took really long because I mostly work with templates that dont require too much focus on shape, colour, etc. Writing the entire design from scratch was relatively new for me and hence took longer.
   The other thing that I struggled with for too long was what I needed to pass in this function: `Icon.createWithContentUri(forecasts.get(position).weatherIcon).` I went through almost everything to fix it and had to finally change the predefined function as mentioned above.
   
3. Were there any tasks that you initially did not know how to complete? And if so, how did you approach them and take the steps to learn how to complete them?

I think that most of the things that I implemented were new for me since I had worked with Android around 4 years back and didn't remember much. But since there was already a structure to the code, it was relatively easy to follow. I didn't have much issues with the backend since it was just writing java code, something that I have done fairly often. For the frontend, I took one part at a time and worked on it. For example, first I focused on changing the colour of the button, then adding an icon to the left of the text, then rounding the edges, etc. I referenced a lot of the documentation and stackOverflow for help too.

4. Was this assessment feasible for the time given and relatively easy to follow? Any points of feedback?

   I think the assessment was feasible for the time given and relatively easy to follow. Specially since everything was well structured and explained.