Proton Technologies Mobile Dev Test

Time Limit - 3 Hours

Please return your attempt of this task 3 hours from receiving it.

Overview

This exercise is meant to test your coding skills as well as how you approach the app development process in general. Your task is to take the provided sample app, get it running and improve it based on the description in this document, as well as your own expectation of code and app quality. How you choose to approach this task is up to you and you should make your own judgements about what's most important to fix or implement first and take a best guess if unsure about something.

You should approach this app with the attitude that you and other developers may have to expand and maintain it for an unknown number of years and will likely outlive you working on it.

Since the time is limited, you are not expected to fix everything, so if you identify something as taking a long time to fix, just leave it.

If you identify some things you would like to do but run out of time for, feel free to submit some notes along with your task submission.

Task

The purpose of this app is to retrieve basic weather information from a RESTful API (already in the source code) and present that information in master and detail screens. Basic data should be downloaded automatically at app launch and images downloaded on request by the user. It would also be good if the app is able to fall back on a cache in case the API is unavailable.

Master Screen

The master screen should have "Forecast" in the navigation bar. Below the navigation bar should be a pair of tabs with the options "Upcoming" and "Hottest". Below these buttons should be a table view displaying the relevant forecasts and in the correct order. - Upcoming: All upcoming days returned by the API in order of day - Hottest: Any days that have less than a 50% chance of rain, ordered hottest to coldest

The rows should be in the format "Day {day}: {description}" where day and description are keys returned by the API. If the image for a particular day is downloaded (initiated by the user in the detail screen), then the row in the master view should distinguish this visually by changing the row's color or using a thumbnail.

Detail Screen

The navigation bar title should be in the form "Day {day}". Below the navigation bar the weather data should be laid out in a consistent manor as you see fit using appropriate units and values. Below the text data you should display the image (once downloaded). A "Download Image" button should be aligned to the bottom of the screen, and should disappear upon successful download.