**Code Review Inputs from Mphasis**

1. Is the app responsive, will it work both in Landscape and Portrait mode?
2. How to handle/parse an API endpoint in the below scenarios:
   1. End point returning null
   2. Key missing
   3. Additional keys/objects
   4. Null values
   5. Data type mismatch
   6. Multiple or alternate keys from same attribute
   7. End point did not return anything
   8. End point does not exist
   9. N/w failure or slow internet encountered during the call
   10. Server/End point not responding
3. Recycler View - Where there are multiple lists on the screen if one of the API fails what will happen to the other?
4. Think about edge case scenarios or all boundary conditions while implementing
5. When you add or edit is it possible to push this data to API endpoint?
6. When data from API rendered in the UI, how exactly the refresh is considered (is it pulled from the API again)
7. Let’s say we call an endpoint, at the time of response, we get an IoException (let’s say this is a business exception) and you want to show the message “Please try later” — how will you handle this?
8. Use getFirst and getLast do not use index to get the data
9. Top bar and bottom nav should be consistent
10. How is the call to the API and UI rendering during the following:
    1. Search - new data, already searched data
    2. Adding new data
    3. Updating or deletion
11. While making call all the time. Live API call how is it happening. Are you going to network connector at that time? How are you making sure Network adaptor is called only once, but during internal calls it should be taken care without calling Network Adaptor multiple times (refer the above scenarios)
12. When Network adapter calls an api endpoint and retrieves the data if the same API end point is called again will it load the data from the cache or it will call the end point again?
13. What are the different scenarios considered from a unit testing coverage standpoint?
14. How do you get started with Unit Testing and how will you make sure the test coverage is > 70%?
15. How is mocking done from fake repository or from hard coded data?
16. Make sure the data entity and the data are mapped properly
17. Use Dagger Hilt for dependency injection so you will not pass the Dao as a param to the constructor
18. Timber logs - What exactly are timber logs, how to make use of it, where to use it and what should be part of it other than errors like info, warn, debug and trace logs
19. In model make all the data attribute nullable
20. Does the app have localization? If so, what are all the things considered and will it affect the rendering of elements in the layout?
21. Have an action bar/Top Nav bar and bottom nav consistent in all the screens
22. If there is a back button in the top bar what will happen to the state of the app when users go back and forth using the back and forward button.
    1. Example: How exactly the Refresh will be considered to display the fragments of the page. Or the Refresh will not happen. Example: in the weather app Homepage has temperature - 32 degrees. And you go to page 2 and then press the back button and come back to home page. In this scenario the actual temperature may have changed to 31 degrees, so will the page refresh with the actual temperature of 31 degrees or the refresh will not happen, and the temperature remains at 32 degrees.
23. Put all the logic in View Model not in Activity
24. Parsing. Can you show how the JSON parsing is done? What are all the validations considered when passing
25. Always write code in Kotlin-ish way like using Scope functions instead of if else