The various states that an app can enter on your platform of choice

So, for android app here are the various states an App Can Enter:

1. Created:

   - The app is initialized and is in the process of starting up.

   - This state is linked with the `onCreate()` lifecycle callback function

2. Started:

   - The app becomes visible to the user, that is it is shown on the mobile screen, but may not be interactable yet.

   - This state is linked with the `onStart()` lifecycle callback.

3. Resumed:

   - The app is in the foreground and can be actively used by the user.

   - This state is linked with the `onResume()` lifecycle callback.

4. Paused:

   - The app is still in the foreground but not actively interactable (when a system overlay or another app is in the foreground maybe when you get a call).

   - This state is linked with the `onPause()` lifecycle callback.

5. Stopped:

   - The app is no longer visible to the user and has been paused or moved to the background. Basically switching to another app

   - This state is linked with the `onStop()` lifecycle callback.

6. Destroyed:

   - The app is in the process of being terminated and all resources should be released.

   - This state is linked with the `onDestroy()` lifecycle callback.

The various states that you must consider for your app, why you must consider it, and what must happen in each state.

1. Created State:

   - Why Consider It: The app is initializing; developer should set up your app's resources and state.

   - What Must Happen:

     - Initialize components such as databases, network connections, or UI elements.

    - Restore state from a previous session if necessary.

2. Started State:
   - Why Consider It: The app becomes visible to the user, but it's not yet in the foreground.
   - What Must Happen:
     - Prepare the UI for interaction.
     - Check and apply necessary changes before transitioning to the resumed state.

3. Resumed State:
   - Why Consider It: The app is fully in the foreground and interactable for the user
   - What Must Happen:
     - Resume animations, media playback, and other ongoing tasks.
     - Check for updates and reflect changes in the UI.

4. Paused State:
   - Why Consider It: The app is temporarily in the background but might return to the foreground soon.
   - What Must Happen:
     - Save app state and unsaved data.
     - Pause ongoing tasks and animations to conserve resources.

5. Stopped State:
   - Why Consider It: The app is in the background and not visible to the user.
   - What Must Happen:
     - Release resources that aren't needed while the app is in the background.
     - Save persistent data and state.

6. Destroyed State:
   - Why Consider It: The app is being terminated and should clean up properly.
   - What Must Happen:
     - Release all resources, including closing database connections and network sockets.
     - Save any persistent data as necessary.