**1**. **The Various States that an App Can Enter on Flutter Platform**

Flutter apps can transition through several lifecycle states that are essential to understand for effective app management. These states are:

**Inactive**: This state occurs when the app is not currently receiving user input. It might be running in the background, or another application might be active. For instance, this could happen during transitions like incoming phone calls or when a user pulls down the notification panel.

**Paused**: In this state, the app is not visible to the user. It runs in the background and should not perform heavy computations. Background tasks such as playing music or GPS usage can continue in this state.

**Resumed**: The app is visible and interacting with the user. This is the state in which the app is foregrounded and active.

**Detached**: The app's host view is destroyed, meaning the UI is no longer displayed, but the Dart code might still be running in the background. This can occur when the user closes the app or it is terminated by the system.

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

For an efficient and user-friendly Flutter app, especially one like a simple text editor or data entry app, we should focus on managing the following states appropriately:

**Inactive**:

**Why Consider**: Ensures smooth transitions without losing user data or state.

**Actions**: Suspend animations and intensive CPU usage tasks that don’t need to run when the app isn’t receiving input.

**Paused**:

**Why Consider:** Critical for conserving battery life and the device’s performance when the app is not in the foreground.

**Actions:** Suspend tasks that require the UI or that consume unnecessary resources. This is a good time to save all the user’s current state and data.

**Resumed:**

**Why Consider:** Necessary to refresh UI and data to ensure the app appears active and responsive when it’s brought back to the foreground.

**Actions**: Resume any paused operations, refresh user data and check for any updates or changes that might have occurred while the app was inactive or paused.

**Detached:**

**Why Consider:** Essential for handling the end of the app’s lifecycle properly, ensuring user data is saved and sensitive information is secured before the app completely shuts down.

**Actions:** Perform cleanup activities, save all user data permanently to storage, and possibly log session end analytics.