VA Mobile Haptic Feedback Recommendation

# Summary

The haptic feedback, also known as tactile touch experience, is when users interact with their mobile device, the system responds to the user's action. It is an added accessible feature to enhance the onscreen interface experience.

For the sake of this guideline, we will abbreviate it to Haptics. It will cover both iOS and Android (specifically Pixel and Samsung) mobile experience.

[Improving Native App Accessibility via Haptics | by Sheri Byrne-Haber, CPACC](https://sheribyrnehaber.medium.com/improving-native-app-accessibility-via-haptics-66ad190d1e30)

### iOS

Link: <https://developer.apple.com/design/human-interface-guidelines/ios/user-interaction/haptics/>

For iOS, there are several ways we can incorporate haptics into our mobile app:

#### Native experience

* Attach to the native components such as toogle, sliders, or pickers. This ideal situation has already been predetermined by the mobile operating system that allows it to manage the strength and behavior of the feedback. For example, using a toggle component automatically plays a subtle tap.

#### Feedback experience

Use a feedback Generator to play several pre-defined patterns in notification and selected response areas.

#### Custom experience

To consider more control creating your haptic experience, you can create custom patterns that can vary based on content and user input and control the sharpness and intensity.

For iOS operating system 13 or later, custom feedback now includes:

* Transient events are brief and compact experiences that feel like taps or impulses, such as the experience of tapping the Flashlight button on the Home screen
* Continuous events, which feel like sustained vibrations, such as the experience of the laser's effect in a message

### Android

Link: <https://material.io/design/platform-guidance/android-haptics.html#usage>

References: <https://developer.android.com/reference/android/view/HapticFeedbackConstants>

For Android, there are several ways we can incorporate haptics into our mobile app focusing on two major themes:

#### Touch feedback

* Focusing on the touch experience, material design suggests we use the following listed as our feedback model
* Tap and click
* Long press

#### Attention cues

* Alarms and reminders
* Notifications
* Incoming calls
* Error and success states

Additionally, for Android to be similar to iOS, pairing with other audio is beneficial to blind users. But when sound turns off, the experience is also repetitive, using visual elements to guide it. When paired with other features, haptics should be synchronized to create compelling experiences. Finally, the operating system is open-sourced, allowing designers and developers to customize it even further and leaving settings open wide to users.

## General Recommendations

When we consider where we want to implement areas of opportunities for haptics to be successful in our mobile product, we should be mindful of our everyday users' placements and user experience. Things to consider such as:

* For it to be **Complementary** by building a clear relationship between haptic and the other feedbacks helps people understand its use. When the app's visual, auditory, and tactile feedback sync, the user experience is more consistent and can seem more natural.
* Make it **Voluntary** by letting our users have the option to turn on/off haptics if they wish, and the app is still usable when haptic is off.
* Have a **Purposeful** intent by avoiding needless alerts as its announcement is power feedback that should only be used to provide valuable and actionable information. If our users see too many signals, they will ignore where we want them to find them essential when interacting with our app.
* Be **Understated** by considering integrating unnoticeable feedback. We want to ensure the users can get valuable information without being interrupted or taking action. The information does not compete with original content but allows them to check them anytime with a glance.
* Don't exploit the **Intensity** by using the strength of the feedback as an experience. For example, sharpness values would convey precision as a warning or an error or could be to the point as a success.
* *Keep it strictly* ***simple*** *for accessibility. While this benefits deaf and low vision users, it may trigger issues for those with sensory disabilities or PTSD. Focus on simple interactions and start slow.*

## Apps to help with Haptics placement

* Tapticme: <https://apps.apple.com/us/app/tapticme/id1202558790> (Meli’s favorite)
* Haptic Haven: <https://apps.apple.com/us/app/haptic-haven/id1523772947>
* Haptic Kit: <https://apps.apple.com/us/app/haptickit/id1563487692>

## 

# 

# VA Mobile App Recommendations

## 

## VA Mobile app Demo:

| iOS/Android | iOS only | Android only |
| --- | --- | --- |
| * **Light** * **Medium** * **Heavy** * **Rigid** * **Soft** * **Notification success** * **Notification warning** * **Notification error** | * Selection | * Clock tick * Context click * Keyboard press * Keyboard release * Keyboard tap * Long press * Text handle move * Virtual key * Virtual key release * Click effect * Double click effect * Heavy click effect * Tick effect |

## 

## Flow Examples

Areas where haptic flow cycle could be utilized but need to be confirmed by testing with research.

* Play "Heavy" haptic event when completing a long service call that populates a screen's primary purpose. The purpose of the haptic event is to tell the user, "I'm done working; bring your attention back." A maximum of one "Heavy" haptic event is allowed to play per screen automatically.
* A "Heavy" function should eventually follow any "Light" events to indicate the finality of the work.
* Play "Success" when a flow completes as the purpose of the haptic event is to communicate, "you're done. "
* Play "Failure" when flow does not complete as the purpose of the haptic event is to communicate the flow has failed and any user actions have not been committed. The "Failure" event would play when the user arrives on the confirmation (appointments or claims upload) page, but the PNR creation failed; the "Failure" event would play.
* Selection Haptic would be beneficial for situation where selecting text for copy mode.

## Design system references (green indicates recommended first steps)

| **Symbol or experience** | **Classification Reasoning** | **iOS** | **Android** |
| --- | --- | --- | --- |
| **Toggle** | **Settings:**   * **Face ID** * **Notifications** * **Haptics (after added)**   **Letters:** | **“**Heavy” | **“**Heavy” |
| **Alerts** | * Informational: * Warning: “Notification warning” * Error: “Notification error” * Success: “Notification success” | | |
| **Form field buttons** | Areas of ideal usage:  Error state: “Notification error”  Success state | See alerts recommendations | See alerts recommendations |
| **Form fields** | Areas of ideal usage:   * Keyboard Clicking | *Apple does not recommend* | * Keyboard press * Keyboard release * Keyboard tap |
| **Pagination** | Areas of ideal usage:   * **Button** | Light | Light |
| **Bottom Navigation** | Preview list/ shortcut    Response to long presses (accessing extra options menu) | Medium | Medium |
| **Buttons** | Primary/secondary/large card | Light | Light |
| **CTA Buttons (calender etc)** | **Follow buttons** |  |  |
| **Picker list** | Option Scrolling (such as selecting your state from a list) | * Soft | * Soft |