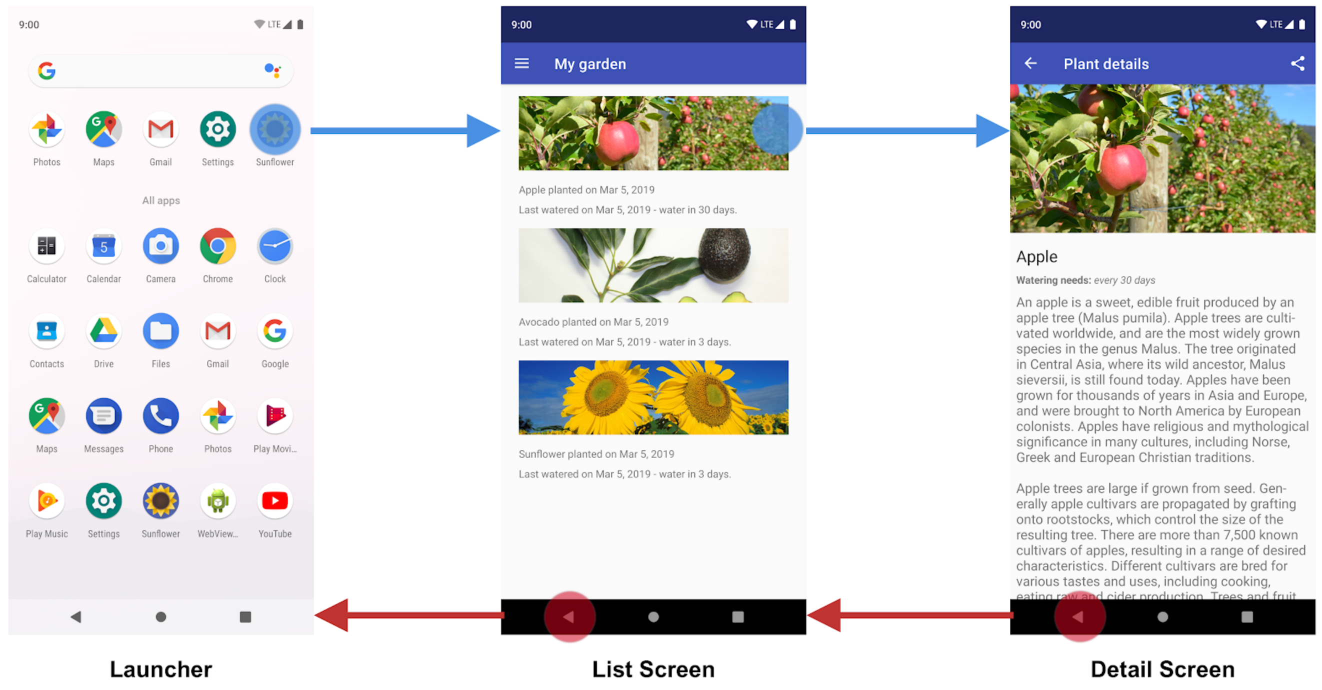
Principles of navigation

Navigation between different screens and apps is a core part of the user experience. The following principles set a baseline for a consistent and intuitive user experience across apps. The Navigation component (/topic/libraries/architecture/navigation) is designed to implement these principles by default, ensuring that users can apply the same heuristics and patterns in navigation as they move between apps.

Note: Even if you aren't using the Navigation component in your project, your app should follow these design principles.

Fixed start destination

Every app you build has a fixed start destination. This is the first screen the user sees when they launch your app from the launcher. This destination is also the last screen the user sees when they return to the launcher after pressing the Back button. Let’s take a look at the Sunflower app (https://github.com/googlesamples/android-sunflower/tree/master/app) as an example.



Apple planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 30 days.

Avocado planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.

Sunflower planted on Mar 5, 2019
Last watered on Mar 5, 2019 - water in 3 days.

Apple

Watering needs: *every 30 days*

An apple is a sweet, edible fruit produced by an apple tree (Malus pumila). Apple trees are cultivated worldwide, and are the most widely grown species in the genus Malus. The tree originated in Central Asia, where its wild ancestor, Malus sieversii, is still found today. Apples have been grown for thousands of years in Asia and Europe, and were brought to North America by European colonists. Apples have religious and mythological significance in many cultures, including Norse, Greek and European Christian traditions.

Apple trees are large if grown from seed. Generally apple cultivars are propagated by grafting onto rootstocks, which control the size of the resulting tree. There are more than 7,500 known cultivars of apples, resulting in a range of desired characteristics. Different cultivars are bred for various tastes and uses, including cooking, eating raw and cider production. Trees and fruit

Figure 1: The List Screen is the Sunflower app's start destination.

When launching the Sunflower app from the launcher, the first screen that a user sees is the **List Screen**, the list of plants in their garden. This is also the last screen they see before exiting the app. If they press the Back button from the list screen, they navigate back to the launcher.

Note: An app might have a one-time setup or series of login screens. These conditional screens (/topic/libraries/architecture/navigation/navigation-conditional) should not be considered start destinations because users see these screens only in certain cases.

Navigation state is represented as a stack of destinations

When your app is first launched, a new task (/guide/components/activities/tasks-and-back-stack) is created for the user, and app displays its start destination. This becomes the base destination of what is known as the *back stack* and is the basis for your app's navigation state. The top of the stack is the current screen, and the previous destinations in the stack represent the history of where you've been. The back stack always has the start destination of the app at the bottom of the stack.

Operations that change the back stack always operate on the top of the stack, either by pushing a new destination onto the top of the stack or popping the top-most destination off the stack. Navigating to a destination pushes that destination on top of the stack.

The Navigation component (/topic/libraries/architecture/navigation) manages all of your back stack ordering for you, though you can also choose to manage the back stack yourself.

Up and Back are identical within your app's task

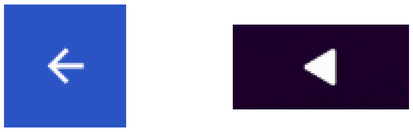


Figure 2: The Up and Back buttons

The Back button appears in the system navigation bar at the bottom of the screen and is used to navigate in reverse-chronological order through the history of screens the user has recently worked with. When you press the Back button, the current destination is popped off the top of the back stack, and you then navigate to the previous destination.

The Up button appears in the app bar (/training/appbar) at the top of the screen. Within your app's task, the Up and Back buttons behave identically.

The Up button never exits your app

If a user is at the app's start destination, then the Up button does not appear, because the Up button never exits the app. The Back button, however, is shown and does exit the app.

When your app is launched using a deep link (/training/app-links/deep-linking) on another app's task, Up transitions users back to your app's task and through a simulated back stack (#deep-link) and not to the app that triggered the deep link. The Back button, however, does take you back to the other app.

Deep linking simulates manual navigation

Whether deep linking (/training/app-links/deep-linking) or manually navigating to a specific destination, you can use the Up button to navigate through destinations back to the start destination.

When deep linking to a destination within your app's task, any existing back stack for your app's task is removed and replaced with the deep-linked back stack.

Using the Sunflower app again as an example, let's assume that the user had previously launched the app from the launcher screen and navigated to the detail screen for an apple. Looking at the Recents screen would indicate that a task exists with the top most screen being the detail screen for the Apple.

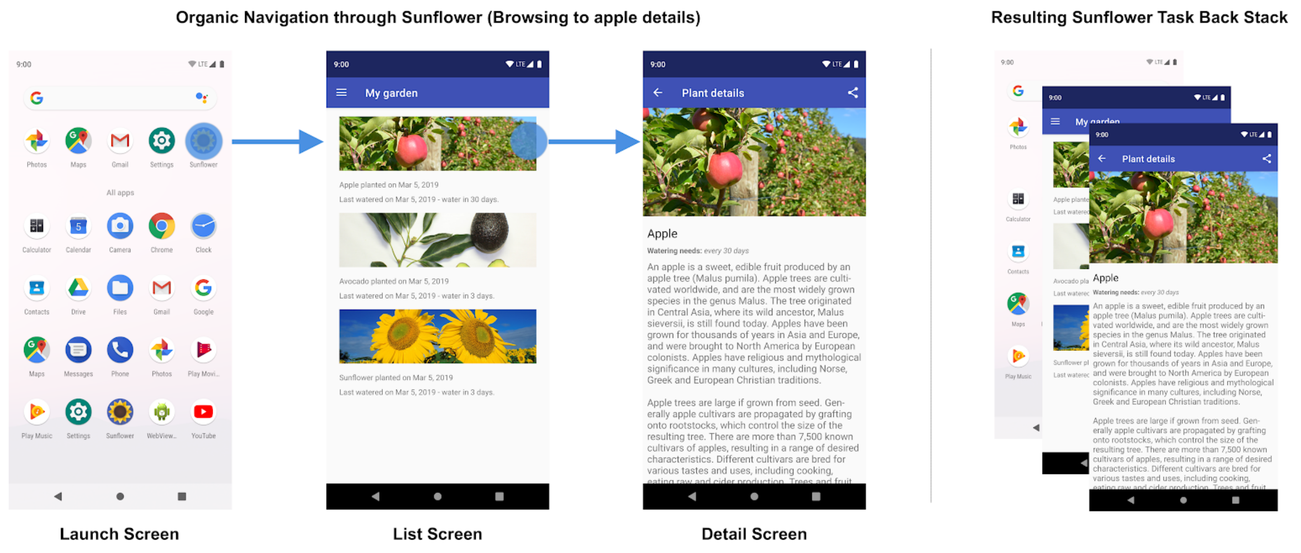
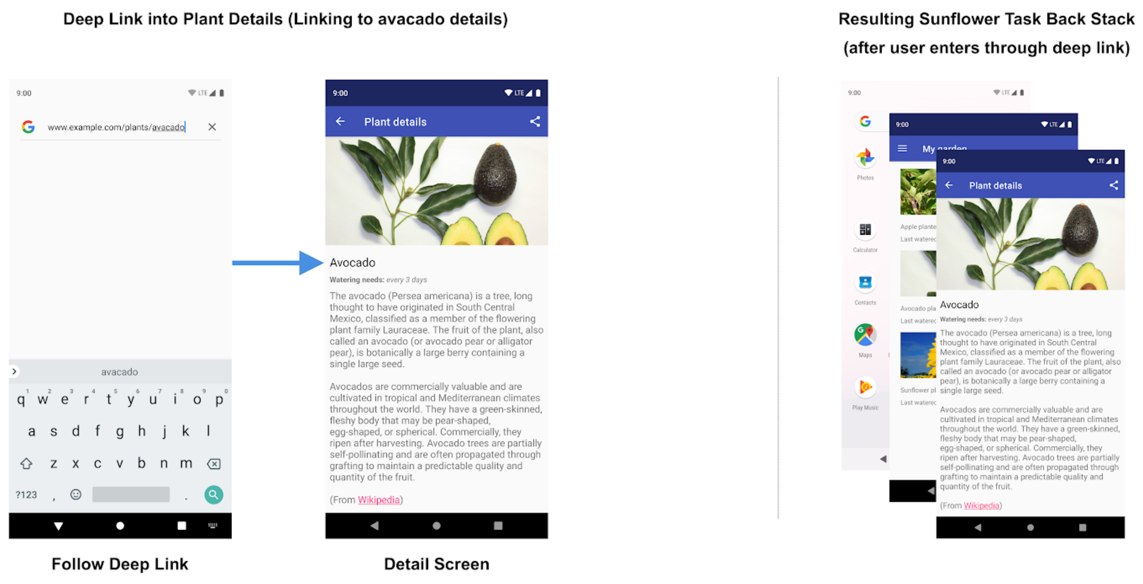

Organic Navigation through Sunflower (Browsing to apple details)
Resulting Sunflower Task Back Stack
My garden
Plant details
Apple
Launch Screen
List Screen
Detail Screen

Figure 3: User navigation through the Sunflower app and the resulting back stack.

At this point, the user can tap the Home button to put the app in the background. Next, let's say this app has a deep link feature that allows users to launch directly into a specific plant detail screen by name. Opening the app via this deep link completely replaces the current Sunflower back stack shown in figure 3 with a new back stack, as shown in figure 4:


Deep Link into Plant Details (Linking to avacado details)
Resulting Sunflower Task Back Stack
(after user enters through deep link)
www.example.com/plants/avacado
avacado
Plant details
My garden
Follow Deep Link
Detail Screen

Avocado

Watering needs: *every 3 days*

The avocado (Persea americana) is a tree, long thought to have originated in South Central Mexico, classified as a member of the flowering plant family Lauraceae. The fruit of the plant, also called an avocado (or avocado pear or alligator pear), is botanically a large berry containing a single large seed.

Avocados are commercially valuable and are cultivated in tropical and Mediterranean climates throughout the world. They have a green-skinned, fleshy body that may be pear-shaped, egg-shaped, or spherical. Commercially, they ripen after harvesting. Avocado trees are partially self-pollinating and are often propagated through grafting to maintain a predictable quality and quantity of the fruit.

(From Wikipedia)

Figure 4: Following a deep link replaces the existing back stack for the Sunflower app.

Notice that the Sunflower back stack is replaced by a *synthetic back stack* with the avocado detail screen at the top. The *My Garden* screen, which is the start destination, was also added to the back stack. This is important because the synthetic back stack must be realistic. It should match a back stack that could have been achieved by organically navigating through the app. The original Sunflower back stack is gone, including the app's knowledge that the user was on the Apple details screen before.

The Navigation component supports deep linking (/topic/libraries/architecture/navigation/navigation-deep-link) and recreates a realistic back stack for you when linking to any destination in your navigation graph.