

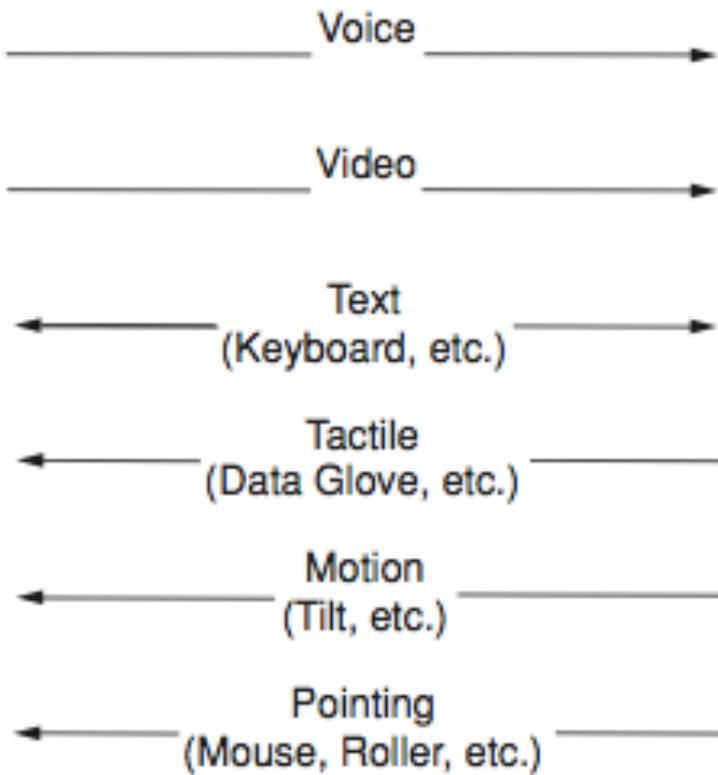


Ingeniería de Software en la
Práctica

UI Android



UI - Interaccion





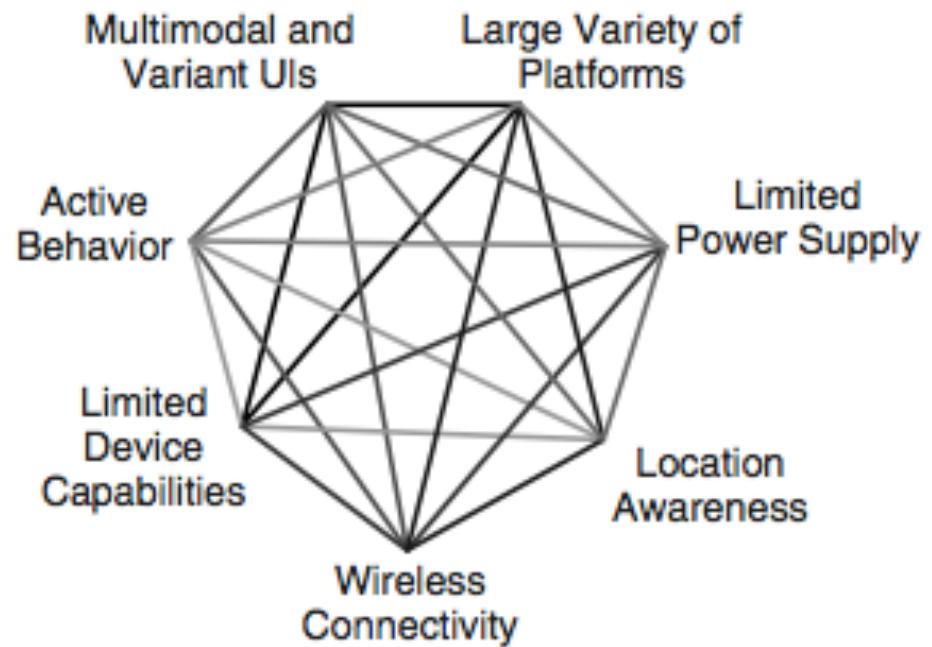
Consideraciones Básicas

FACTOR HUMANO

- Look & Feel de la aplicación (cuán agradable es la interfaz para los usuarios)
- Pendiente de curva de aprendizaje
- Consecuencias de Salud por uso de UI



UI – Nuevas Dimensiones





Usuario Móvil

- El usuario móvil transita, al menos ocasionalmente, entre ubicaciones conocidas y desconocidas
- Generalmente no está enfocado en la tarea computacional
- *Suele requerir grandes grados de inmediatez y tiempo de respuesta del sistema*
- *El usuario suele cambiar de tareas frecuentemente y/o abruptamente*
- Requiere acceso al sistema desde cualquier lado, y en cualquier momento



Atributos de calidad de UI

- Intuitividad
- Consistencia
- Robustez
- Ayuda no intrusiva
- Confiabilidad
- Performance
- Consideracion para usuarios expertos
- Learnability



Consideraciones extra

- Ciclos transaccionales cortos
- Interfaces de usuario multi canal
- Intermitencia en la conectividad
- Foco no exclusivo
- Expectativas del usuario móvil



Atributos de Calidad de Uis Moviles

- Tiempo de Respuesta y Acceso
- Interfaces limpias y eficientes
- Consistencia de la Interfaz entre dispositivos y resoluciones
- Limitaciones del sistema sensorial humano
- Adaptabilidad Posicional
- Priorizacion de elementos de la UI



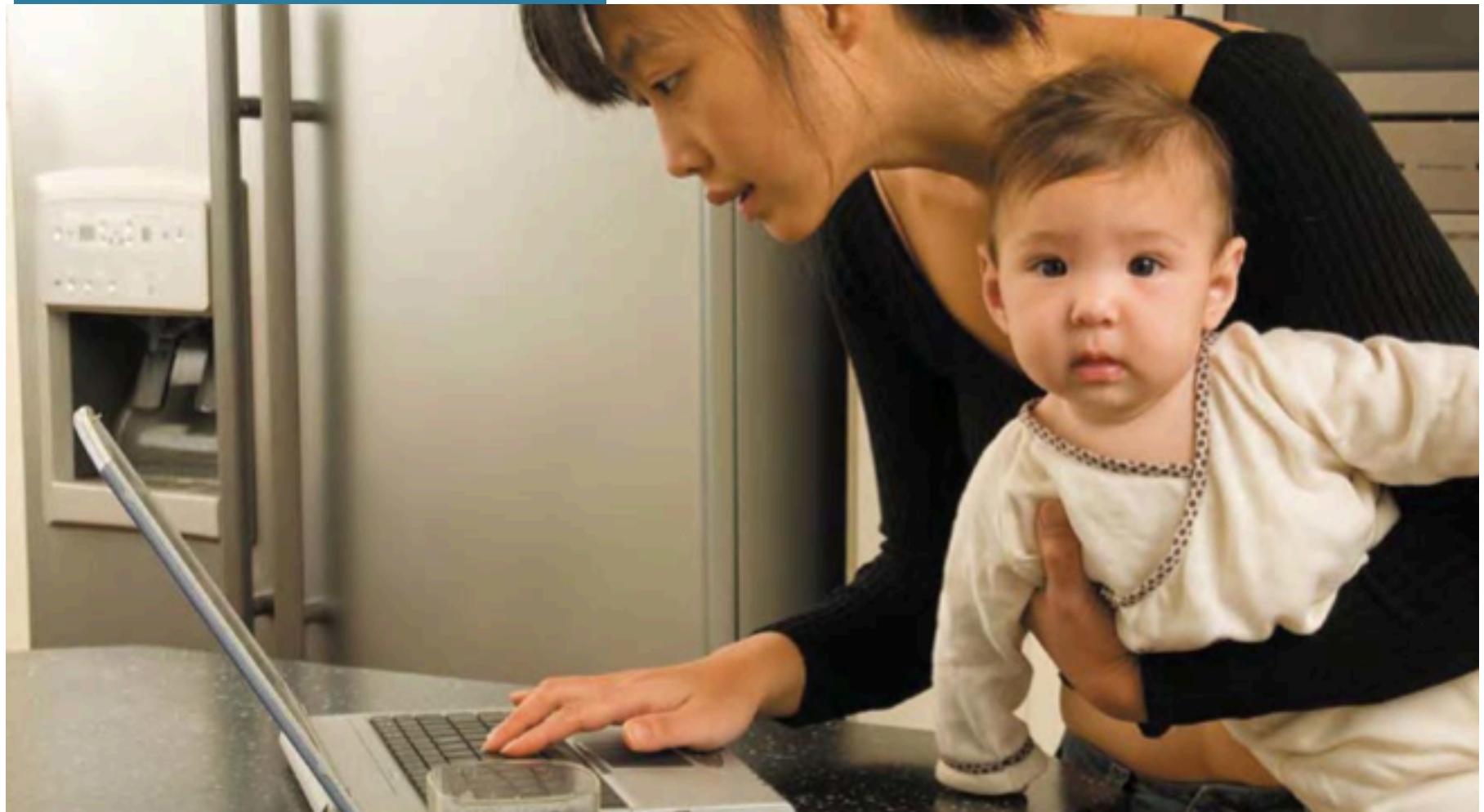
Como empezar?

ENTENDER AL USUARIO!





...en su ambiente
habitual





Tipos de usuario

- Early adopters
- Mainstream
- Expertos





Mainstreamer: que quiere?

- Interesados en terminar la tarea cuanto antes
- Facilidad de control
- Resultados confiables
- En una palabra: SIMPLICIDAD





Como simplificar?

4 estrategias

- Eliminar
- Organizar
- Esconder
- Reemplazar





Simplificando...

ELIMINAR



ORGANIZAR

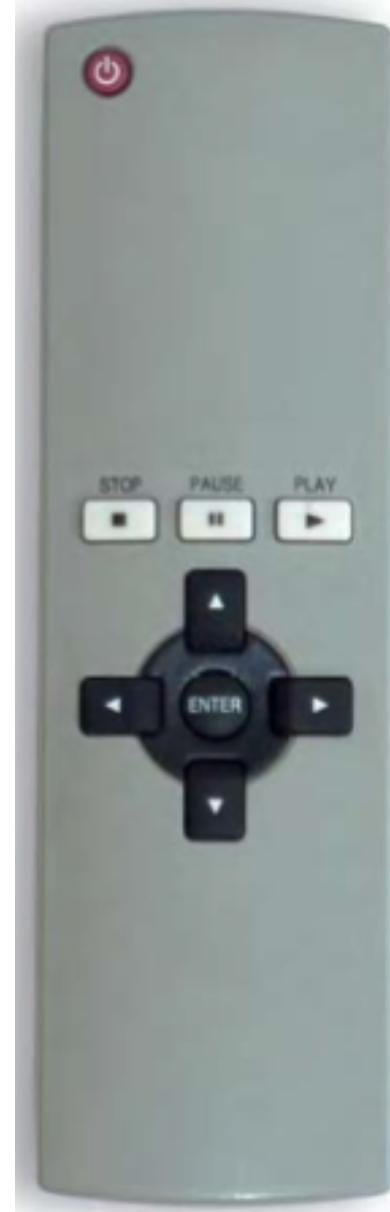




Simplificando...

ESCONDER

REEMPLAZAR





Patrones de Diseño UI

- Patron de diseño: “solución general reusable a un problema recurrente en el diseño de software”
- Las aplicaciones fáciles de usar están diseñadas para tener un look “familiar” hacia el usuario
- La idea de los patrones de UI es que a partir del conocimiento del desarrollador de qué es lo esperado por parte del usuario, poder armar una UI que se sienta "familiar" y a la vez sea original.

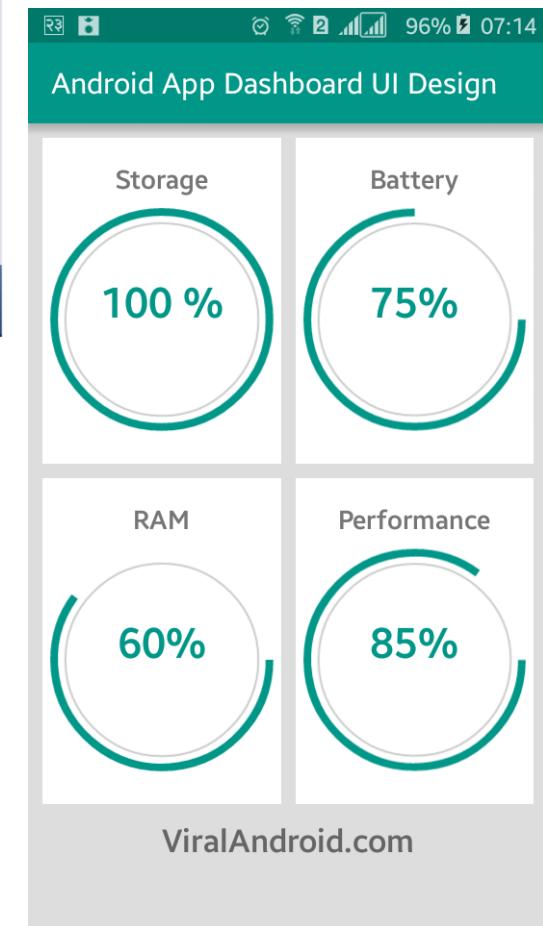


Algunos Patrones

- Dashboard
- Action Bar
- Search Bar
- Quick Actions
- Action Drawer



Dashboard





Dashboard

- Página inicial de la aplicación. Una intro rápida, revela capacidades y presenta nuevo contenido proactivamente.
- Pantalla completa.
- Tipos de organización:
 - Contenidos
 - Categorías
 - Cuentas



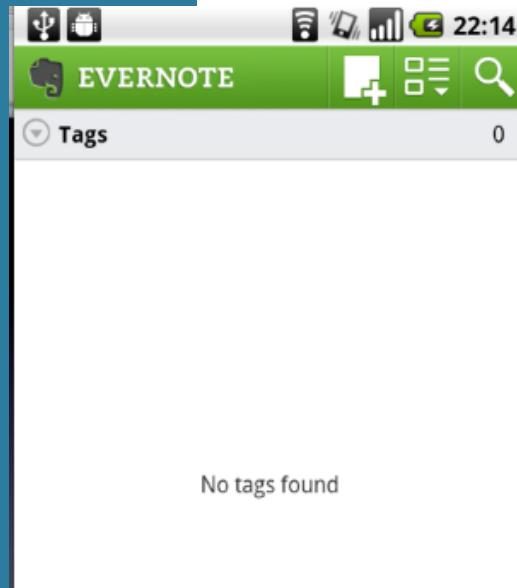
Dashboard

■ **Problema:** Una navegabilidad clara, simple y accesible para la funcionalidad principal es algo vital en las aplicaciones móviles. Las mismas se usan por períodos cortos de tiempo, usualmente para lograr un objetivo simple y único (actualización en red social, enviar mensaje, tomar una foto, etc.)

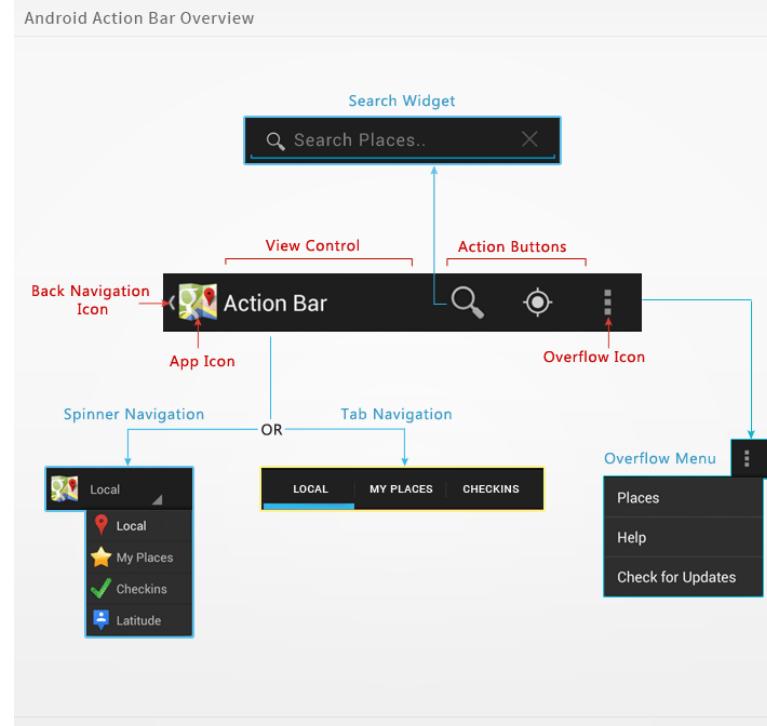
■ **Solución:** La página inicial de la aplicación debería ser **visualmente clara** y brindar **fácil acceso a las tareas principales** que el usuario desea realizar con la aplicación



Action Bar



- Zona dedicada en parte superior de la pantalla para permitir navegación y operaciones frecuentes.
- Reemplaza barra de título
- Útil para operaciones como: buscar, actualizar, y Componer (nuevo)
- Puede permitir un acceso rápido a Dashboard, o pantalla de inicio



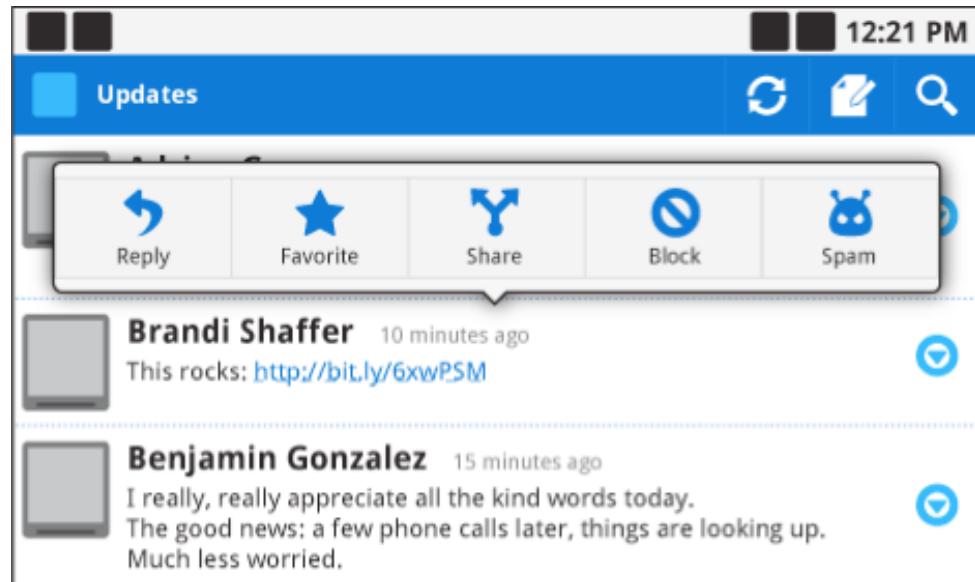


Action Bar

- **Problema:** En aplicaciones móviles el espacio de pantalla es muy limitado. Generalmente una aplicación tiene múltiples acciones disponibles para el usuario en una misma pantalla. Crear botones para cada acción puede consumir mucho espacio. Más allá de esto, los botones también pueden causar problemas de consistencia entre pantallas y aplicaciones
- **Solución:** Action Bar es una barra con el nombre de la aplicación, en la parte superior de la pantalla con acciones relevantes y acceso rápido a la pantalla de inicio.



Quick Actions Menu





Quick Actions Menu

- **Problema:** Los usuarios pueden ejecutar más de una acción sobre objetos en pantalla. Presentar estas acciones en las pequeñas pantallas touch suele ser complejo. Menús del tipo pop-up suelen hacer que el usuario pierda el contexto de su acción.

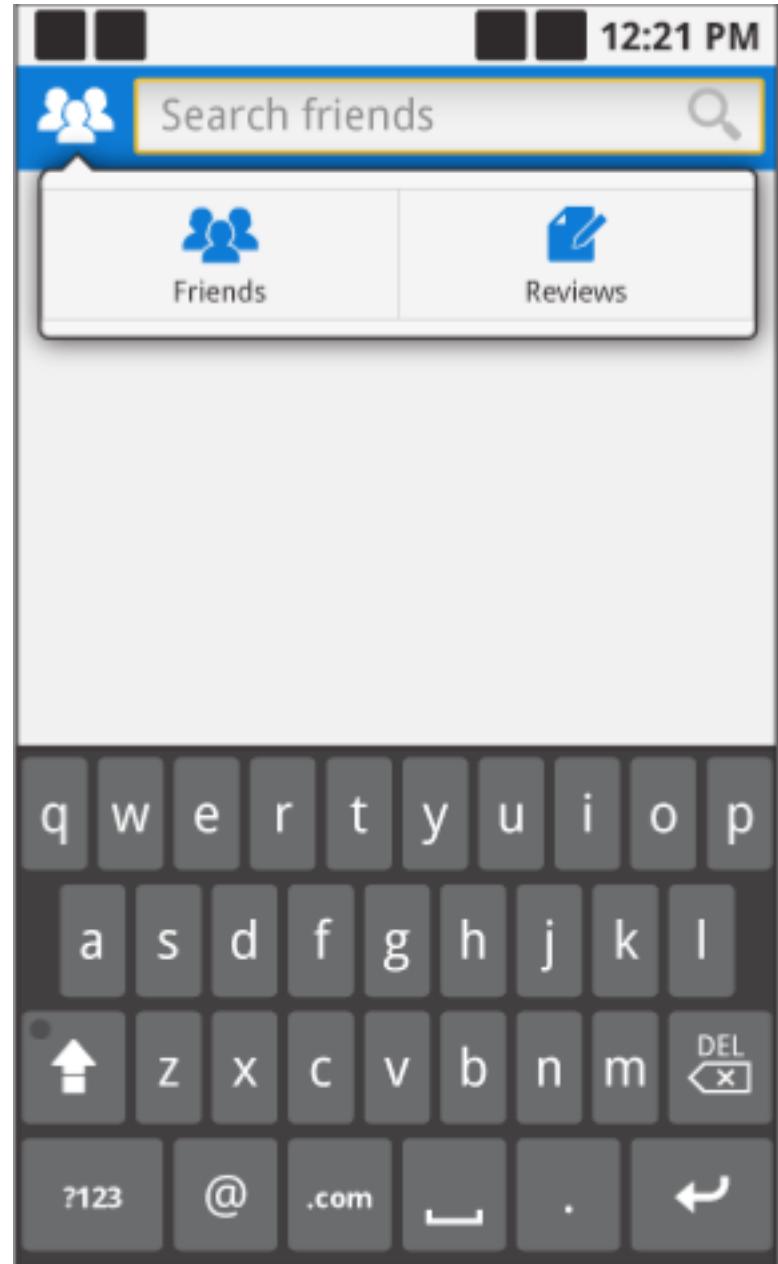


Quick Actions Menu

- **Solución:**
- El menú quick action brinda una forma unificada de presentar acciones ejecutables sobre un objeto. Consiste de 3 componentes:
 - “**Click Target**”: El objeto que el usuario identifica como objetivo para la interacción. Presionar este objeto, abre el menú pop-up.
 - “**Pop-up Action Container**”: Contenedor de las acciones, que debe indicar sobre qué objeto son aplicables (el click target), y NO debe esconder o cubrir dicho objeto.
 - “**Action Icons**”: Representan las acciones disponibles. Solo deben presentarse las acciones aplicables al ‘click target’ . En caso de ser necesario, el contenedor debe mostrar un scrollbar.



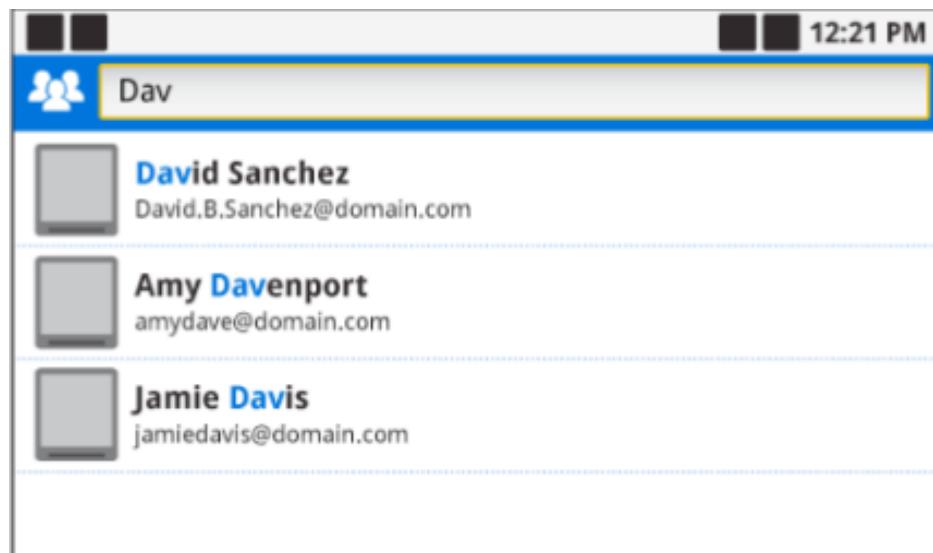
Search Bar





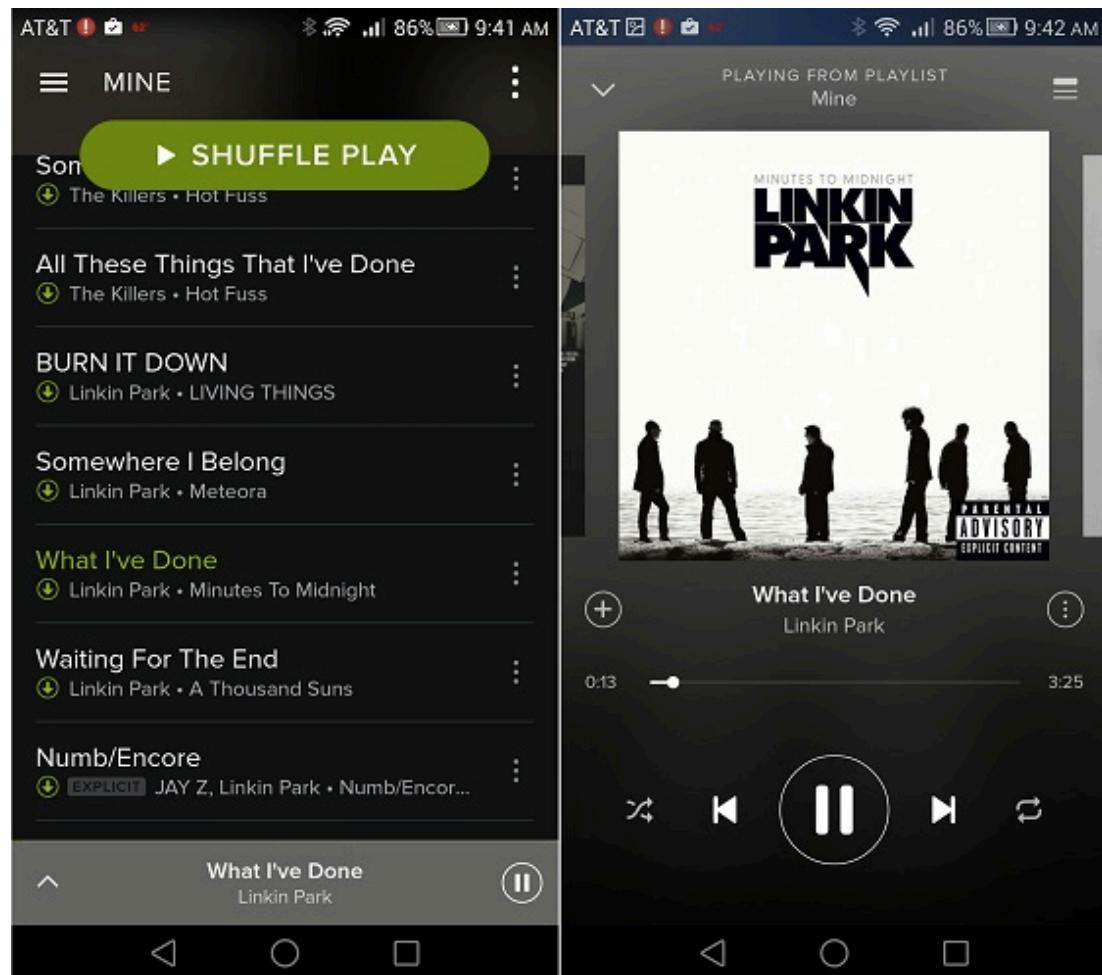
Search Bar

- **Problema:** ¿Cómo buscar información dentro de una aplicación?
- **Solución:** Crear un control pop-in de búsqueda anclado a la parte superior de la pantalla, que reemplace al Action Bar (si está presente).
 - El control debe soportar sugerencias





Action Drawer





Action Drawer

■ **Problema:** Algunas aplicaciones necesitan brindar al usuario una acción común independiente de donde se encuentre el mismo dentro de la aplicación. Por ejemplo, los reproductores de música y sus controles de ejecución. Intentar diseñar una solución consistente para cada pantalla puede ser imposible, o sumamente complejo.

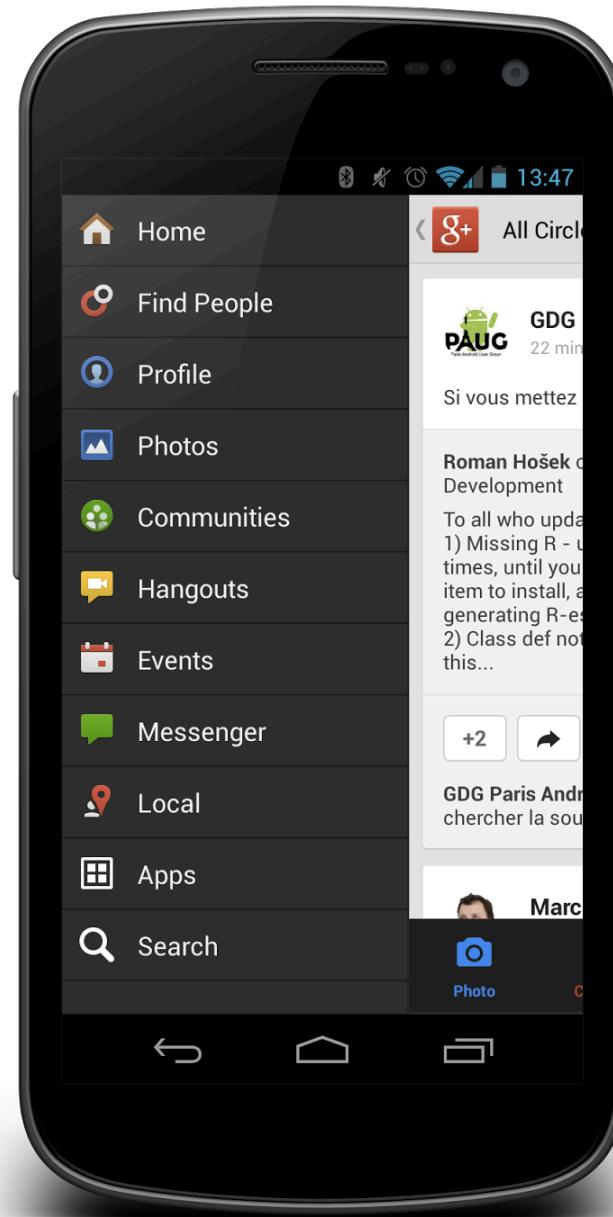


Action Drawer

■ **Solución:** Action drawer es un ‘gabinete’ que puede ser abierto a partir de un control especial (drawer) que ocupa la pantalla completa y brinda las acciones requeridas por el usuario. Luego de utilizado el drawer, el usuario se encuentra en el mismo contexto previo al uso del mismo. Este paradigma es muy sencillo de entender por el usuario si es implementado de manera visualmente correcta. El action drawer debería ocupar prácticamente toda la pantalla, pero mantener ciertas pistas visuales que indiquen que está por encima de la interfaz de usuario normal.



Navigation Drawer





Navigation Drawer

- **Problema:** Las aplicaciones necesitan brindar al usuario una forma rápida de navegar dentro de la aplicación y que esta forma sea independiente de donde se encuentre el mismo dentro de la aplicación. La navegación debe ser rápida, intuitiva y fácil de usar
- **Solución:** Incluir un punto único de acceso al “mapa” de la aplicación, que sea independiente de lo que el usuario esté haciendo.



Una app con patrones

Dashboard

App
Dashboard

- Unique detail
- Search
- Common action 1
- Common action 2

Intents

Activity

Action bar

App home

Search

Common action 1

Common action 2



Material Design

- Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices. To use material design in your Android apps, follow the guidelines described in the material design specification and use the new components and functionality available in Android 5.0 (API level 21).
- You can also use the Design Support library to integrate additional material design components and patterns into your applications to complement the components available in the Android framework. The Design Support library is compatible with Android 2.3 (API level 9) and above.
- <http://developer.android.com/design/index.html>



Material Theme

- The new material theme provides:
 - System widgets that let you set their color palette
 - Touch feedback animations for the system widgets
 - Activity transition animations

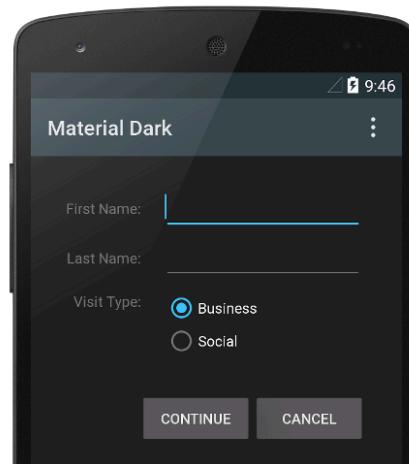


Figure 1. Dark material theme

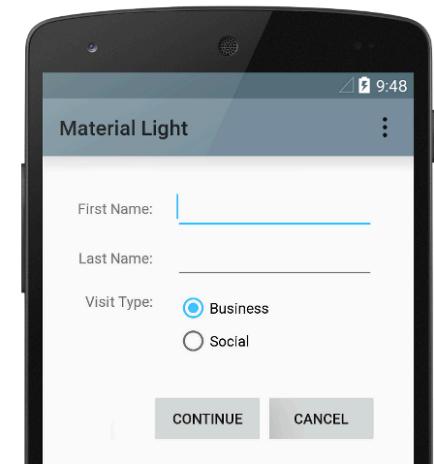


Figure 2. Light material theme



Colores y su importancia

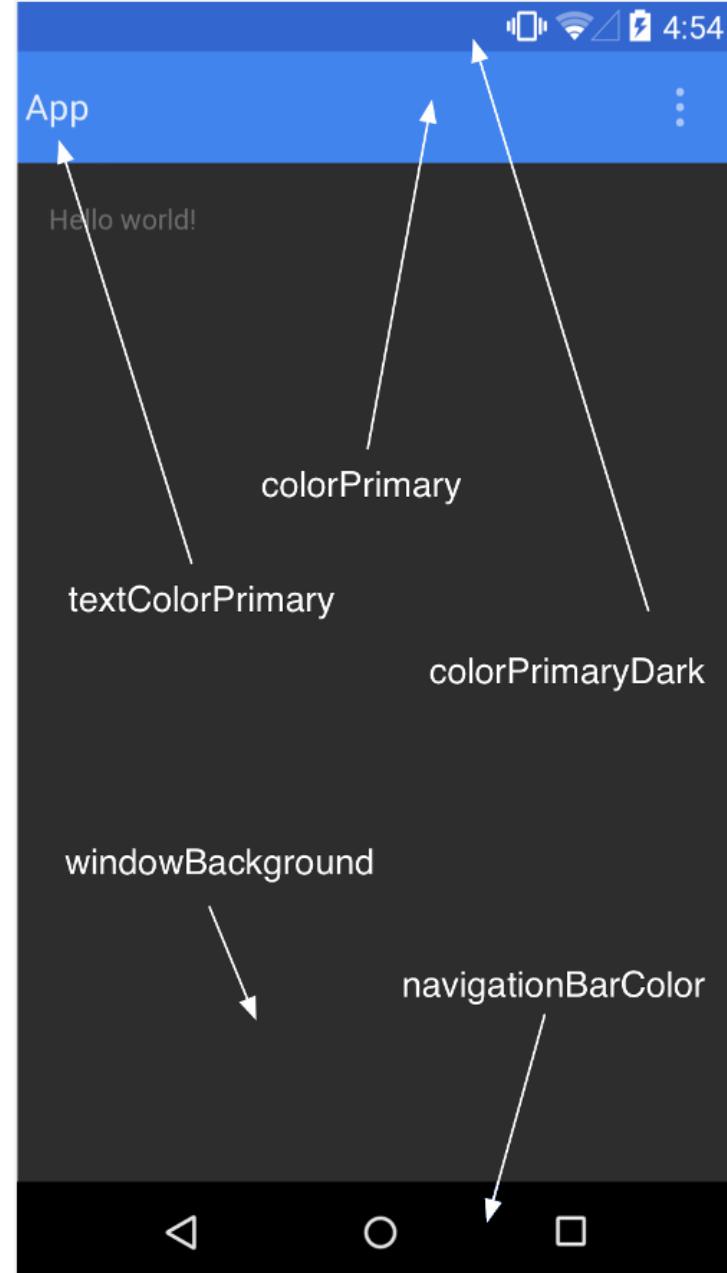
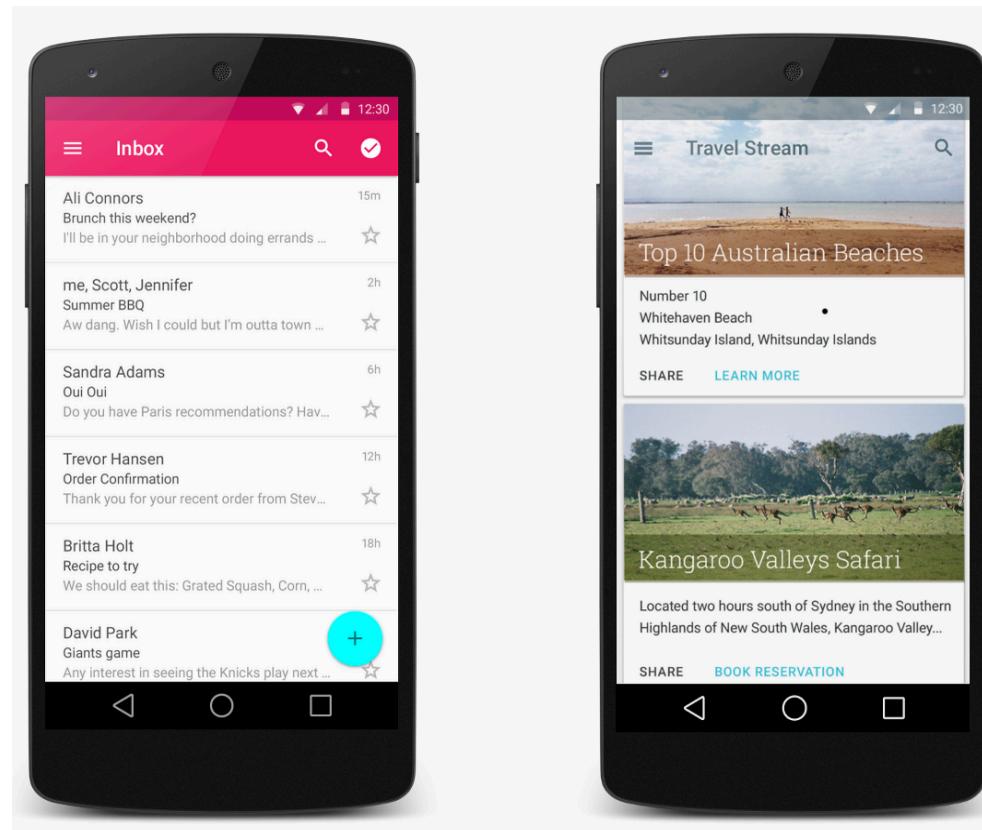


Figure 3. Customizing the material theme.



Listas y tarjetas

- The new RecyclerView widget is a more pluggable version of ListView that supports different layout types and provides performance improvements.
- The new CardView widget lets you display important pieces of information inside cards that have a consistent look and feel.

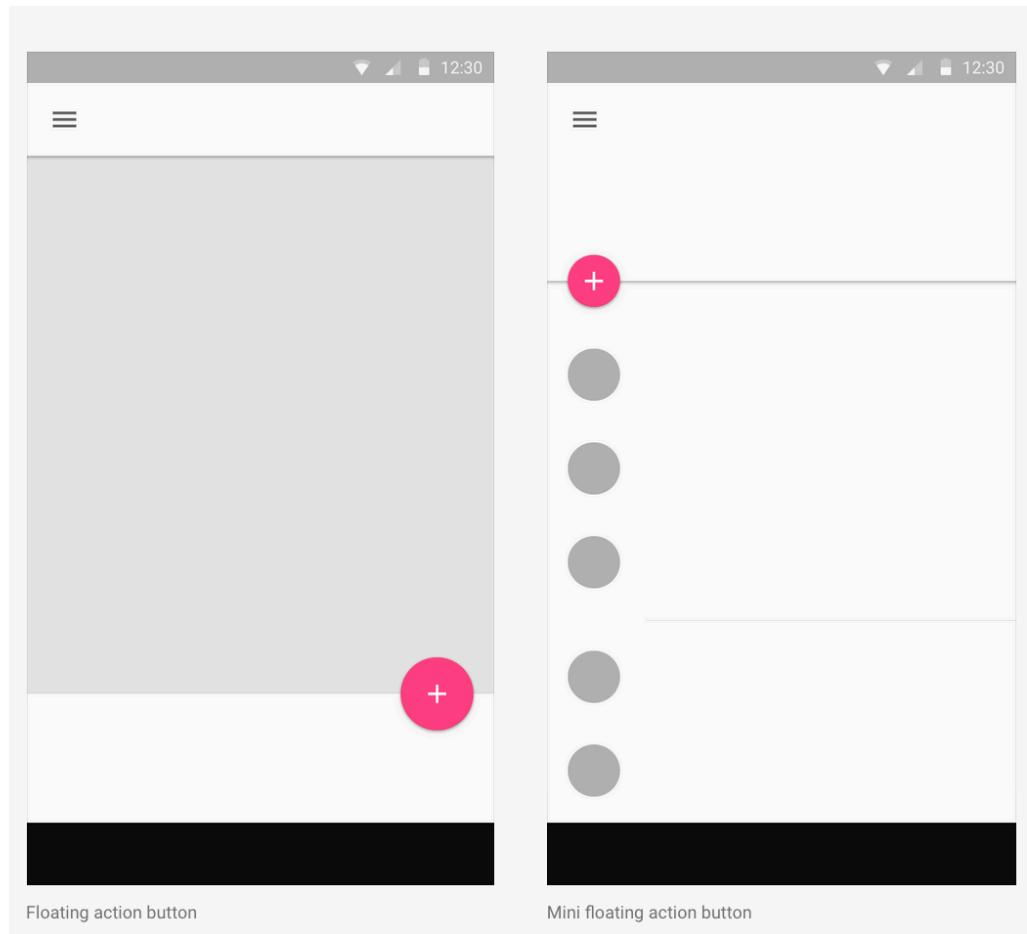




Floating Action Button

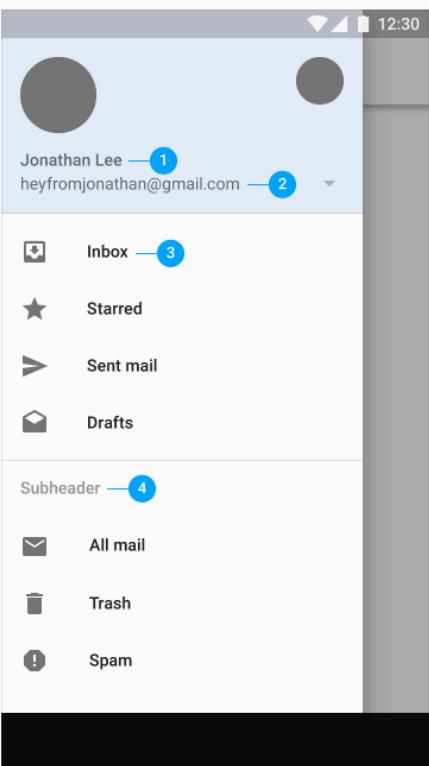
■ **Floating Action Button:** Acción “promovida” dentro de la aplicación.

■ Se usa en dos tamaños:



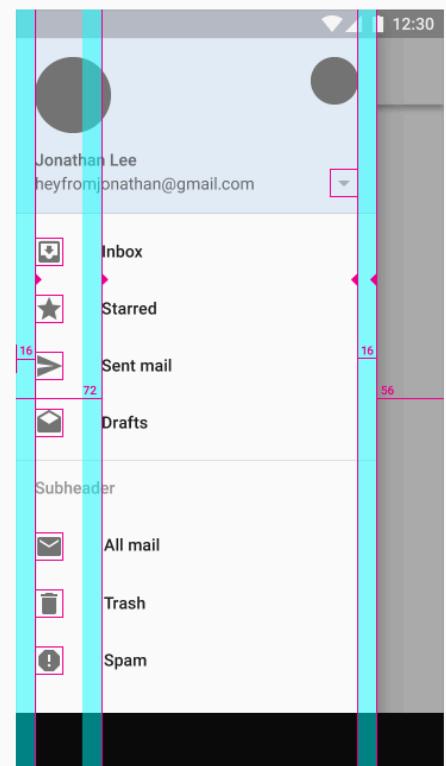


Navigation Drawer



Typography

- Name: Roboto Medium, 14sp, #FFFFFF
- Email address: Roboto Regular, 14sp, #FFFFFF
- List item: Roboto Medium, 14sp, 87% #000000
- Subheader: Roboto Medium, 14sp, 54% #000000. It aligns to the 16dp keyline.



Keylines and margins

- Icons align at screen left and right margins: 16dp

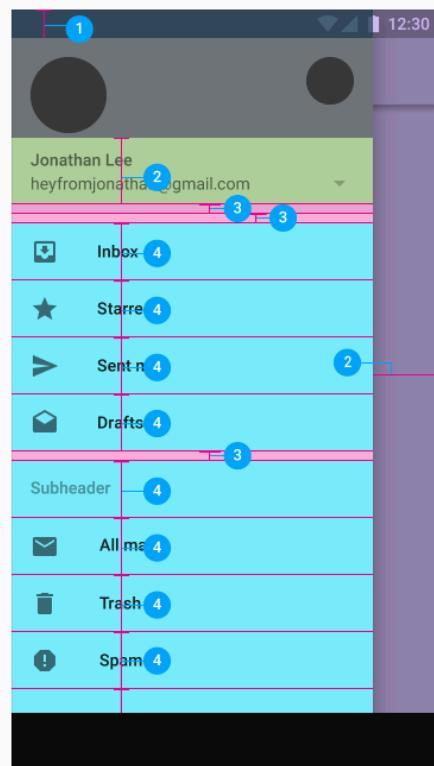
Icon values: 54% #000000

Content associated with an icon or avatar left margin: 72dp

Side nav width: Equal to the screen width minus the height of the action bar. In the example shown above, the nav drawer is 56dp from the right edge of the screen.

Maximum width: The maximum width of the nav drawer is 280dp on mobile and 320dp on tablet. This is calculated by multiplying the standard increment by five (the standard increment is 56dp on mobile and 64dp on tablet).

Use 16dp horizontal margins on mobile and 24dp on tablet



Vertical spacing

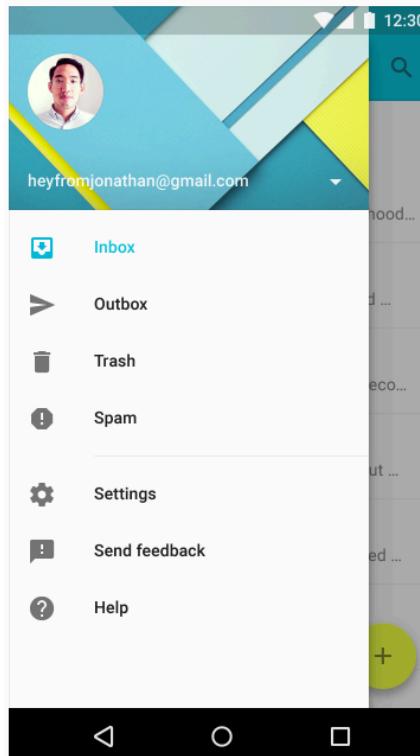
- Status bar: 24dp
- Subtitle: 56dp
- Space between content areas: 8dp
- Subtitles and list items: 48dp

Add 8dp padding at the top and bottom of every list grouping. One exception is at the top of a list with a subheader, because subheaders contain their own padding.

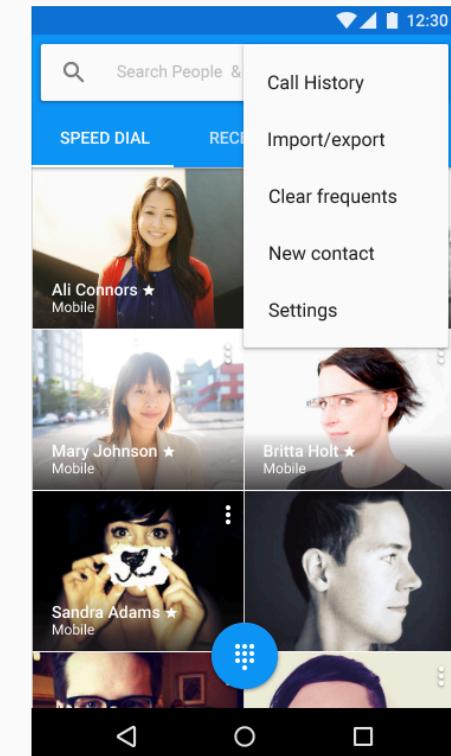


Settings

- All of an app's settings should be reached via the "Settings" label. Do not use synonyms such as "Options" or "Preferences."
- Side navigation: If side navigation such as a navigation drawer exists, include "Settings" below all other items (except Help & Feedback).
- Apps without navigation: Place Settings in the toolbar menu below all other items (except Help & Feedback).



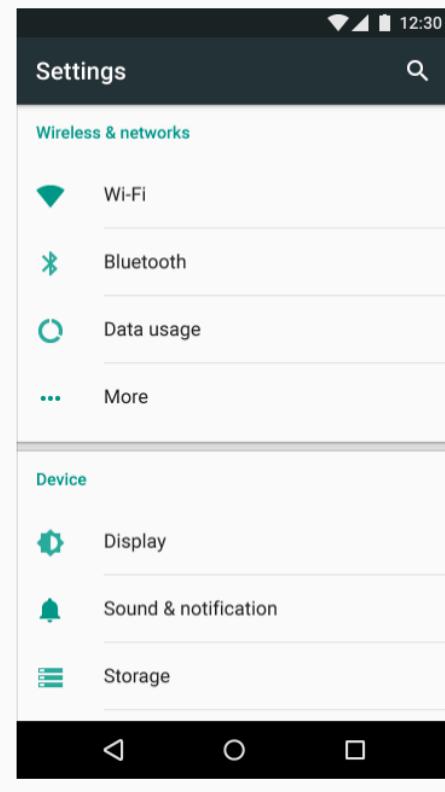
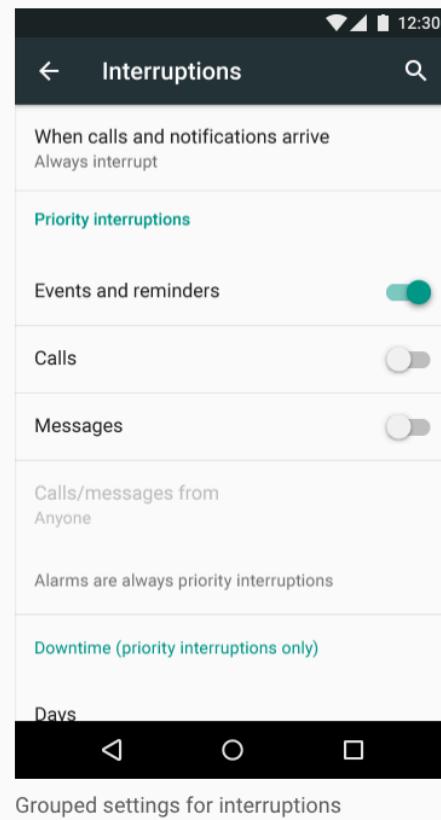
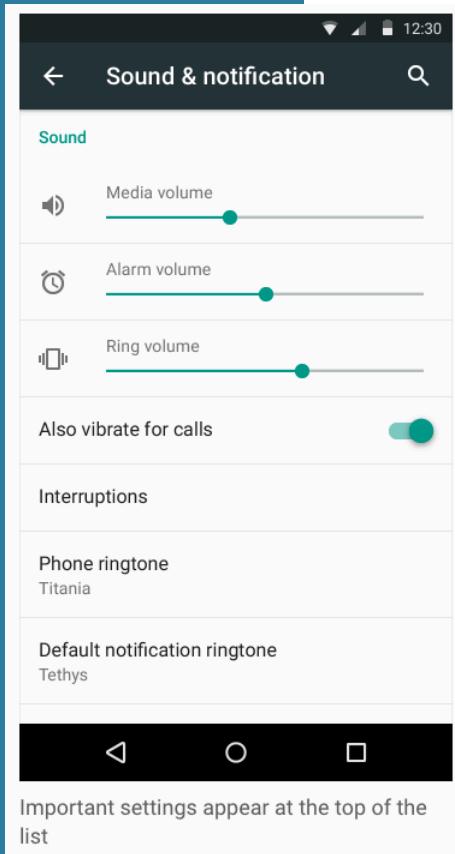
Settings in the navigation drawer



Settings in the toolbar menu



Settings





Notifications

The screenshot displays two collapsed notifications side-by-side. The top notification, labeled 'M collapsed notification', is for 'Justin Rhyss' at 6:00 PM, with the message 'Do you want to go see a movie tonight?'. The bottom notification, labeled 'N collapsed notification', is for 'Messenger' at 'now', with the message 'Do you want to go see a movie tonight?' and includes the app icon and name. Both notifications feature a circular profile picture of a person with glasses and a blue speech bubble icon.

M collapsed notification

Justin Rhyss 6:00 PM
Do you want to go see a movie tonight?

N collapsed notification

Messenger • now ▾
Justin Rhyss
Do you want to go see a movie tonight?

Collapsed notification on Android M (above) and Android N (below)

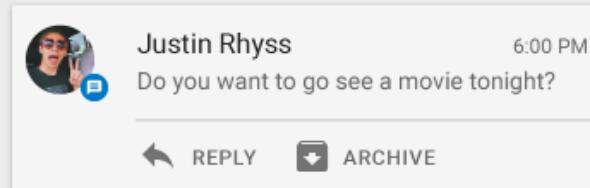
Collapsed notification on Android M (top)
The avatar and icon are on the left and the timestamp is on the right.

Collapsed notification on Android N (bottom)
The app icon, app name, timestamp, and expansion indicator icon appear on the top left. The avatar is on the right.

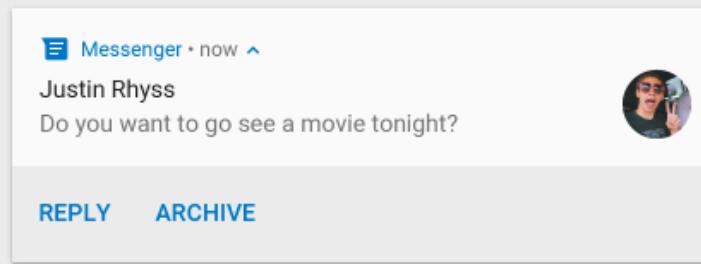


Notifications - Actions

M expanded notification



N expanded notification



Expanded notification on Android M (above) and Android N (below)



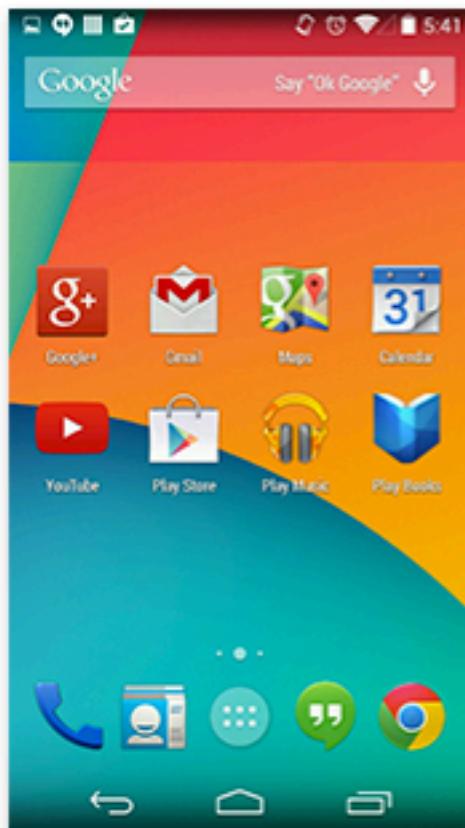
Navigation with Back and Up

Up vs. Back

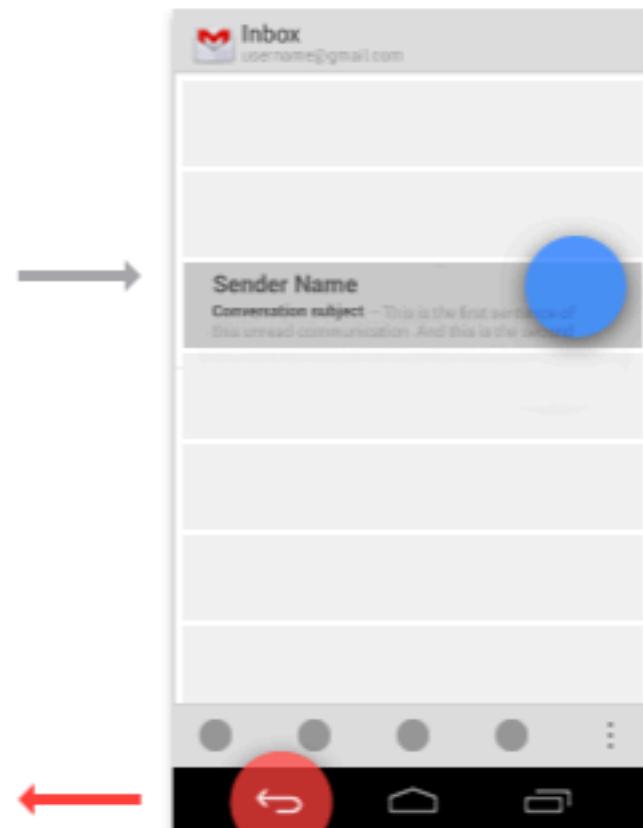
- The Up button is used to navigate within an app based on the hierarchical relationships between screens. For instance, if screen A displays a list of items, and selecting an item leads to screen B (which presents that item in more detail), then screen B should offer an Up button that returns to screen A.
- If a screen is the topmost one in an app (that is, the app's home), it should not present an Up button.
- The system Back button is used to navigate, in reverse chronological order, through the history of screens the user has recently worked with. It is generally based on the temporal relationships between screens, rather than the app's hierarchy.
- When the previously viewed screen is also the hierarchical parent of the current screen, pressing the Back button has the same result as pressing an Up button—this is a common occurrence. However, unlike the Up button, which ensures the user remains within your app, the Back button can return the user to the Home screen, or even to a different app.
- The Back button also supports a few behaviors not directly tied to screen-to-screen navigation:
 - Dismisses floating windows (dialogs, popups)
 - Dismisses contextual action bars, and removes the highlight from the selected items
 - Hides the onscreen keyboard (IME)



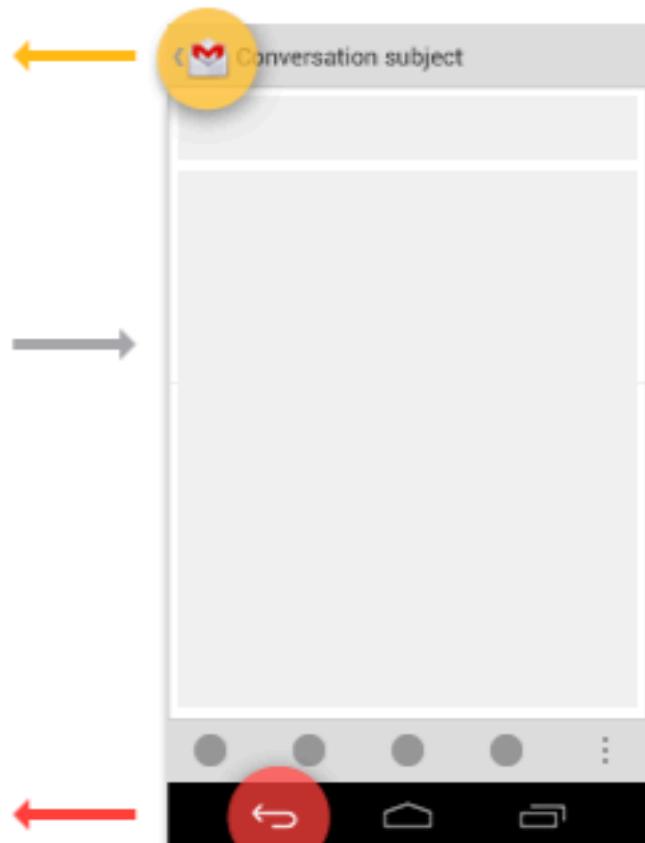
Navigation with Back and Up



Home



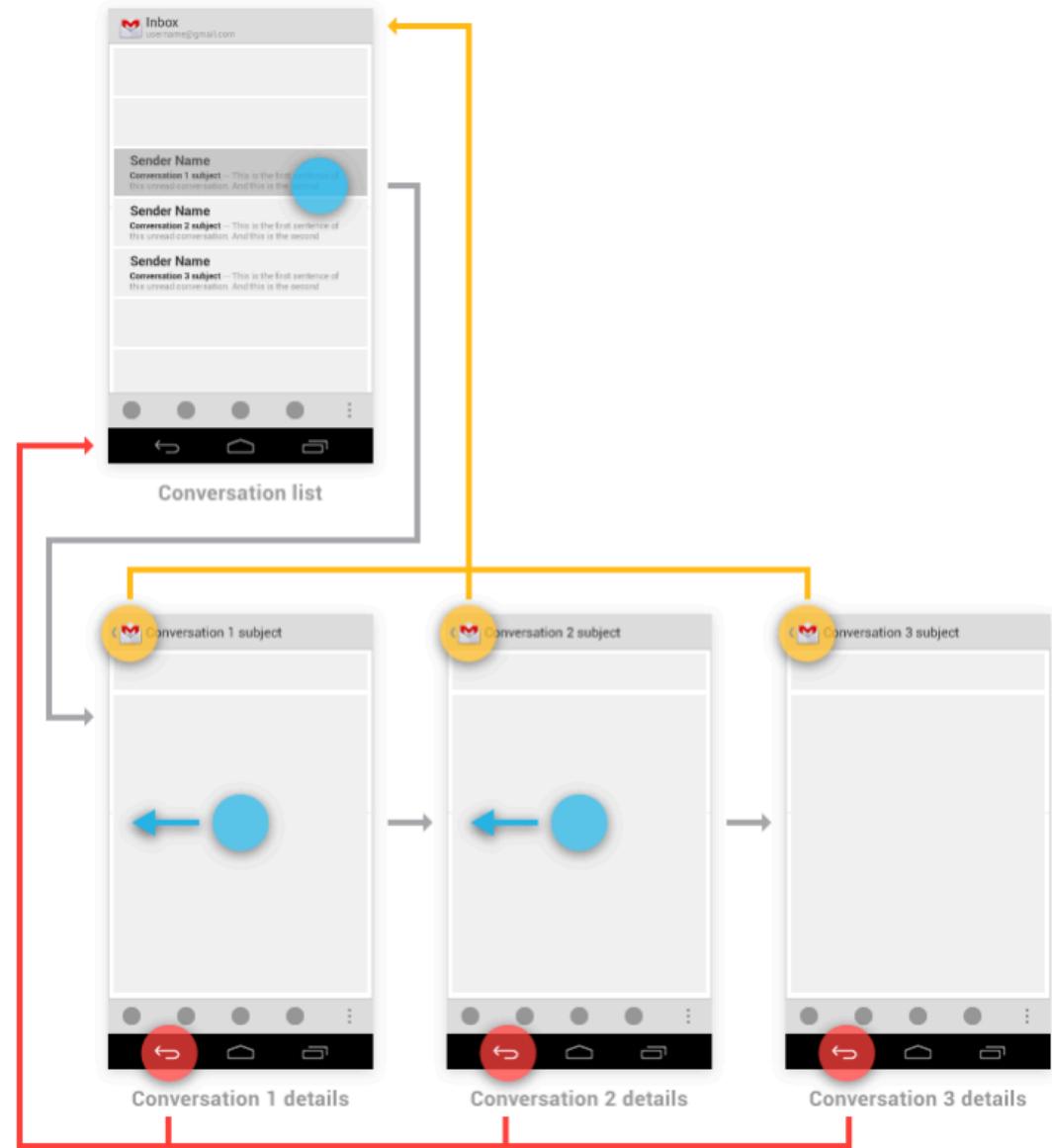
Conversation list



Conversation details

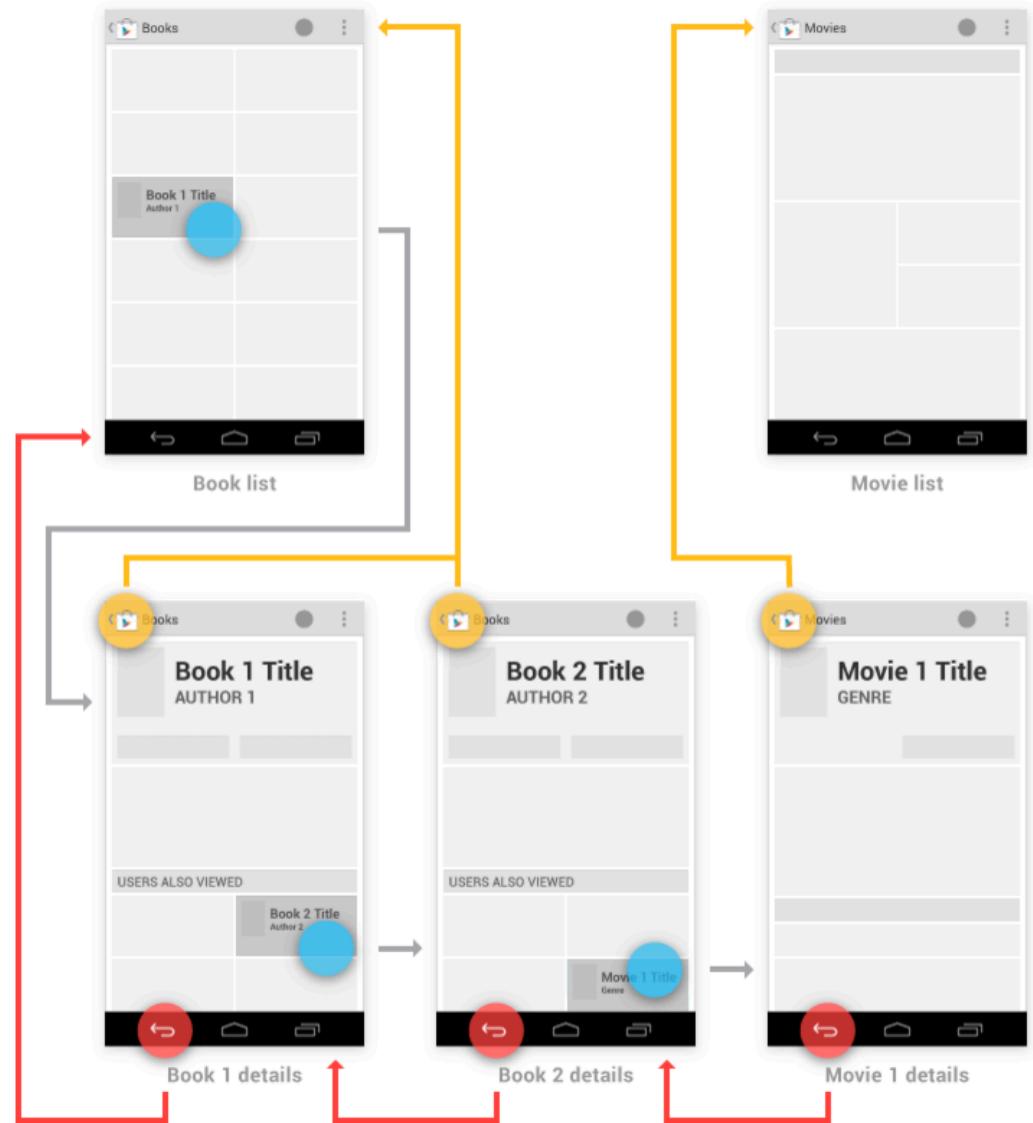


Navigation with Back and Up



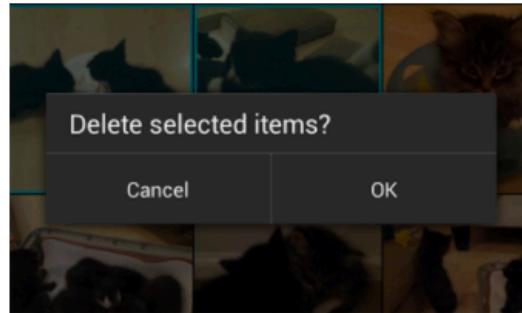


Navigation with Back and Up

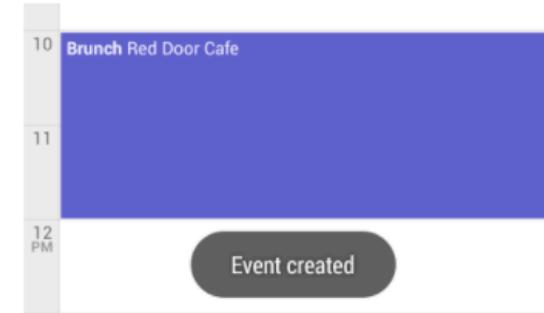




Confirming & Acknowledging



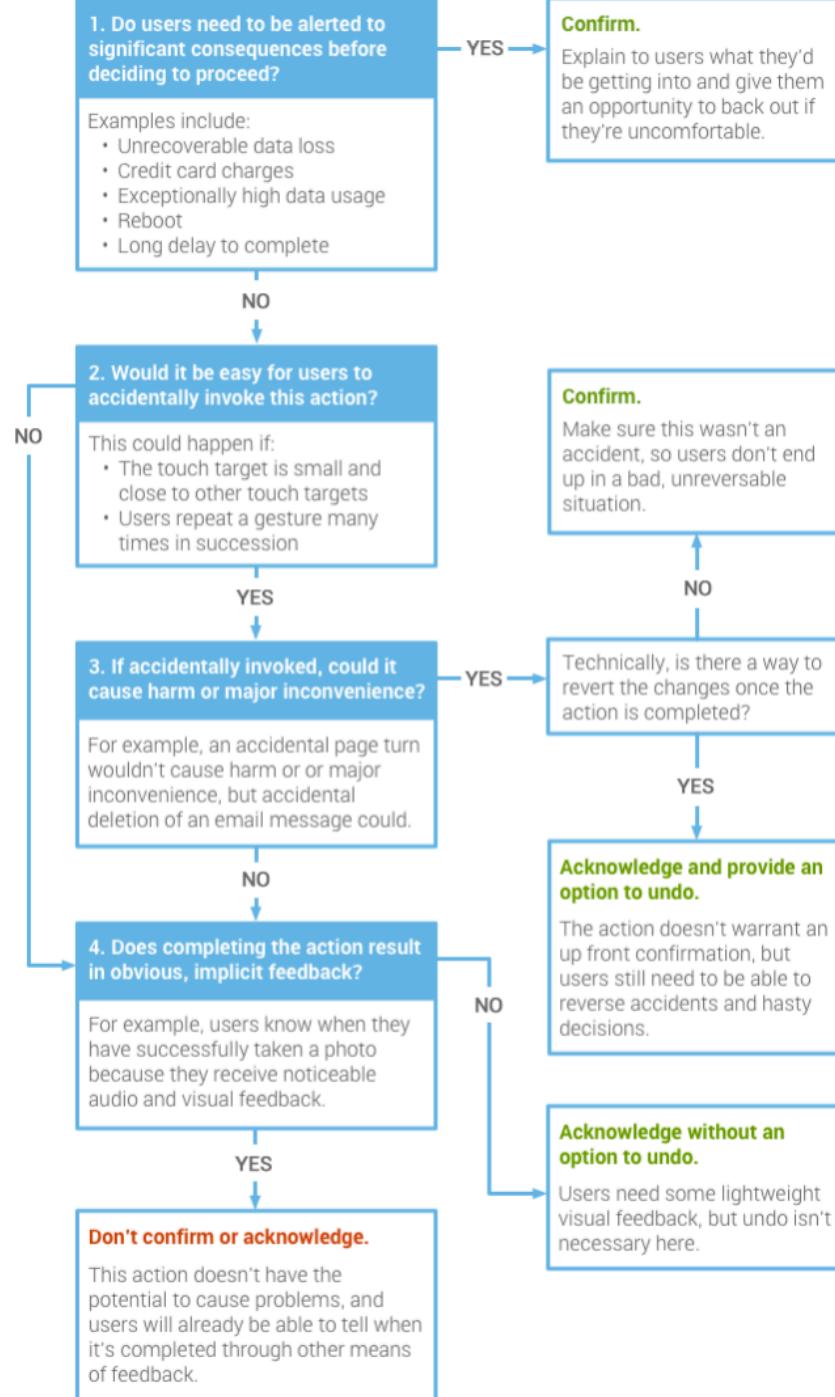
Confirming is asking the user to verify that they truly want to proceed with an action they just invoked. In some cases, the confirmation is presented along with a warning or critical information related to the action that they need to consider.



Acknowledging is displaying text to let the user know that the action they just invoked has been completed. This removes uncertainty about implicit operations that the system is taking. In some cases, the acknowledgment is presented along with an option to undo the action.



Confirming & Acknowledging





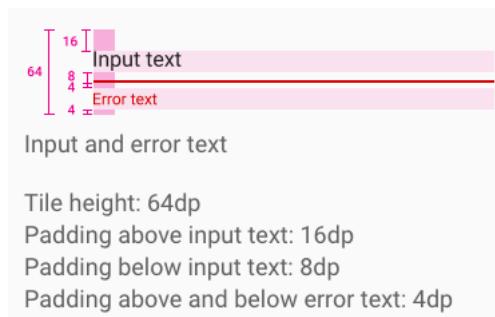
Errors – text field

The screenshot shows a mobile application interface titled "Application". Under the heading "Personal info", there are two text input fields: "First name" and "Last name". Below these are two more fields: "Birthday" (MM/DD/YYYY) and "Social Security" (### - ## - ####). At the bottom, there are fields for "Residence address" (Address, City, State), "ZIP Code", and "Country". A black navigation bar at the bottom contains icons for back, home, and recent apps.

Example of a text field input

This screenshot is similar to the first one, showing the "Personal info" screen. However, the "First name" field now has a red underline and the error message "First name is required" displayed below it. The "Last name" field contains the value "Connors". The "Birthday" and "Social Security" fields remain the same. The "Residence address" section is also present. The black navigation bar is at the bottom.

Example of a text field input with error text





Errors with floating text label

The screenshot shows the 'Create an event' screen with floating text labels. The labels are positioned above their respective input fields. The labels are: 'Event name', 'Location', 'From', 'To', 'Description', and 'Attendees'. The input fields are: 'Event name' (empty), 'Location' (empty), 'From' (Fri, Oct 10 2014, 4:30 PM), 'To' (Fri, Oct 10 2014, 5:30 PM), 'Description' (empty), and 'Attendees' (empty). The bottom of the screen features a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

Example of floating text label with input text

The screenshot shows the 'Create an event' screen with floating text labels. The labels are: 'Event name', 'Location', 'From', 'To', 'Description', and 'Attendees'. The input fields contain errors: 'Event name' has 'Mel's birthday celebration', 'Location' has '500' (with red underline and 'Incomplete address' message), 'From' has 'Fri, Feb 10 2014, 4:30 PM' (with red underline and 'Date occurs in the past' message), 'To' has 'Fri, Feb 10 2014, 5:30 PM' (with red underline and 'Date occurs in the past' message), 'Description' has 'Come help celebrate Mel's birthday with dinner, drinks and putt-putt.', and 'Attendees' is empty. The bottom of the screen features a black navigation bar with three white icons: a triangle pointing left, a circle, and a square.

Example of floating text label with error text



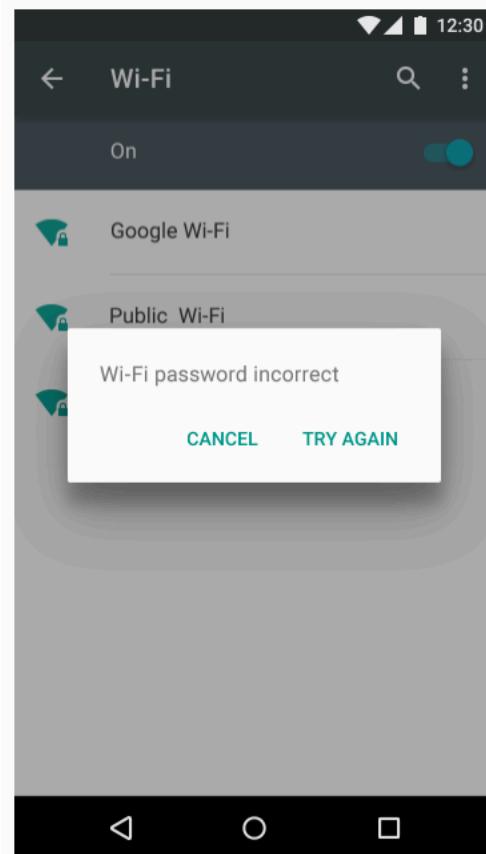
Specs for floating text label

Tile height: 80dp
Padding above label text: 8dp
Padding above and below input text: 8dp
Padding above and below error text: 4dp

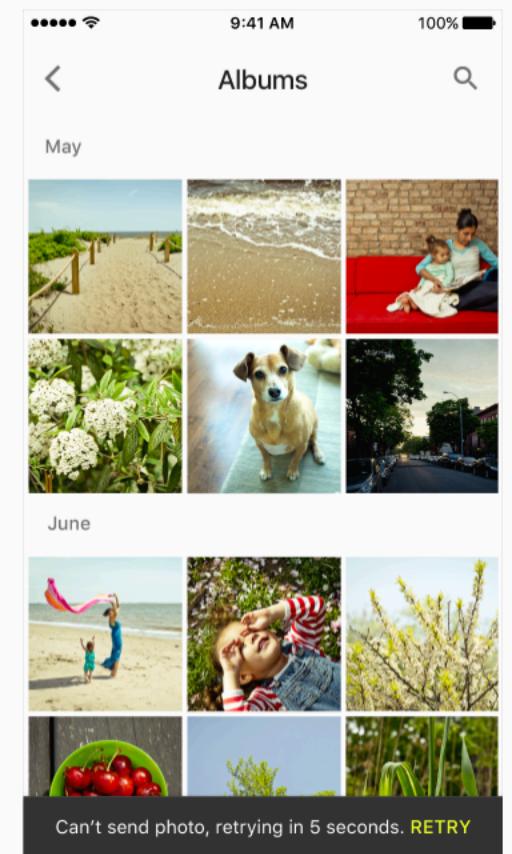


App Errors

General app error



Alert dialog: app feedback about an error that is blocking normal operation

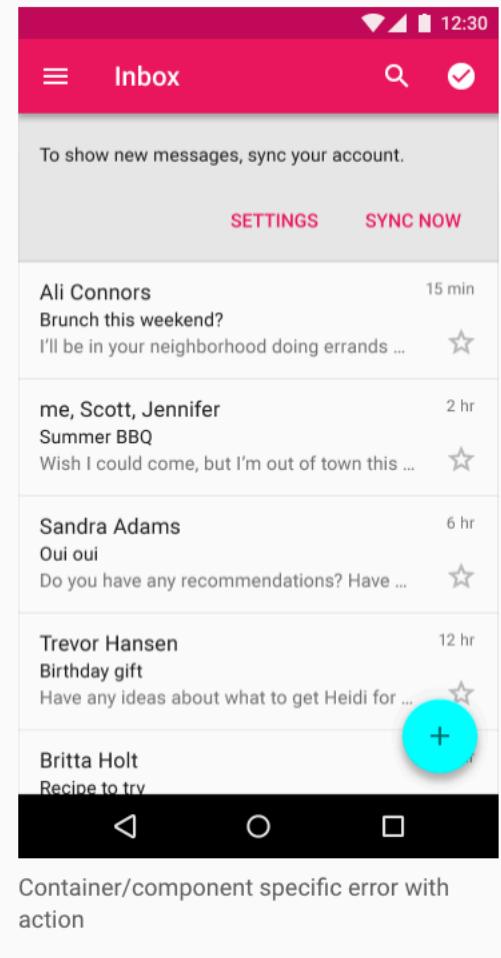
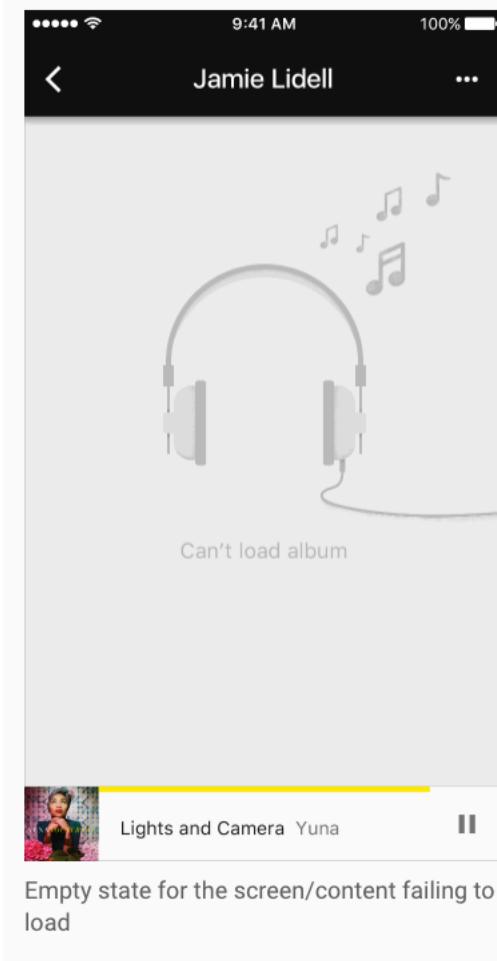


The snackbar contains app feedback about a peripheral error. Snackbars are transient. Don't use them for critical, persistent, or bulk errors.



App Errors

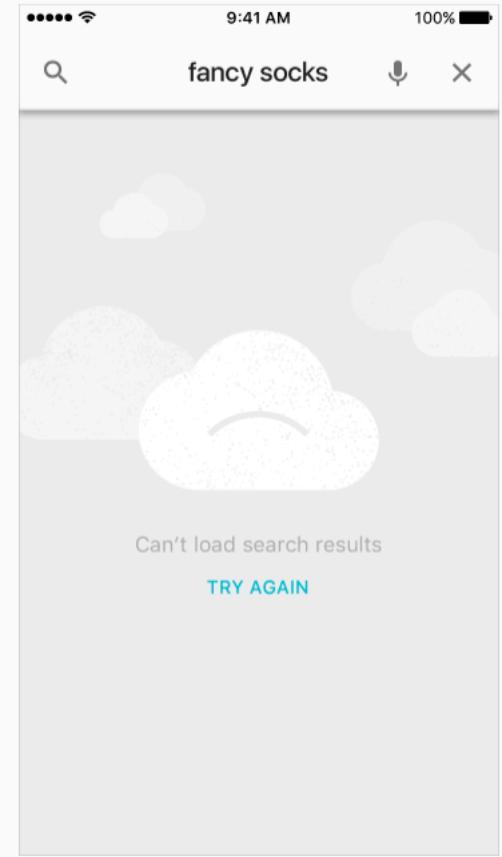
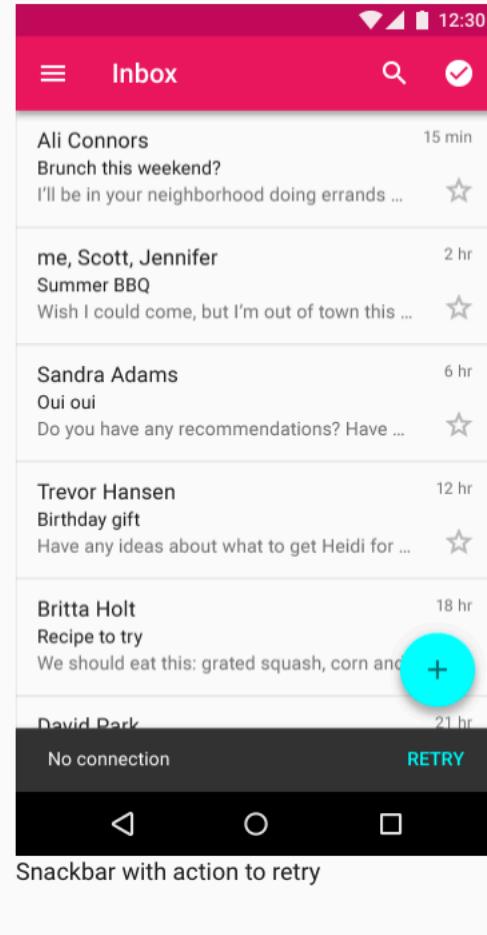
Sync error/failure to load





App Errors

Connectivity



Empty state for a screen that is only available online



Material Design

MATERIAL DESIGN	Components	Patterns
Material design	Bottom navigation Bottom sheets Buttons Buttons: Floating Action Cards Chips Data tables Dialogs Dividers Expansion panels Grid lists Lists Lists: Controls Menus Pickers Progress & activity Selection controls Sliders Snackbars & toasts Steppers Subheaders Tabs Text fields Toolbars Tooltips Widgets	Confirmation & acknowledgements Data formats Empty states Errors Fingerprint Gestures Help & feedback Launch screens Loading images Navigation Navigation drawer Navigational transitions Notifications Permissions Scrolling techniques Search Selection Settings Swipe to refresh
Motion		
Style		
Layout		
		Platforms
		Platform adaptation Android
		Resources
		Color palettes Devices Layout templates Roboto & Noto fonts Sticker sheets & icons Shadows
		Growth & communications
		Introduction Onboarding Feature discovery Gesture education
		Usability
		Accessibility Bidirectionality



Android Support Library

- When developing apps that support multiple API versions, you may want a standard way to provide newer features on earlier versions of Android or gracefully fall back to equivalent functionality. Rather than building code to handle earlier versions of the platform, you can leverage these libraries to provide that compatibility layer. In addition, the Support Libraries provide additional convenience classes and features not available in the standard Framework API for easier development and support across more devices.
- Originally a single binary library for apps, the Android Support Library has evolved into a suite of libraries for app development.
- **Caution:** Starting with Support Library release 24.2.0 (August 2016), the minimum supported API level across most support libraries has increased to Android 2.3 (API level 9) for most library packages.
- <https://developer.android.com/topic/libraries/support-library/index.html>