Mobile Application Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

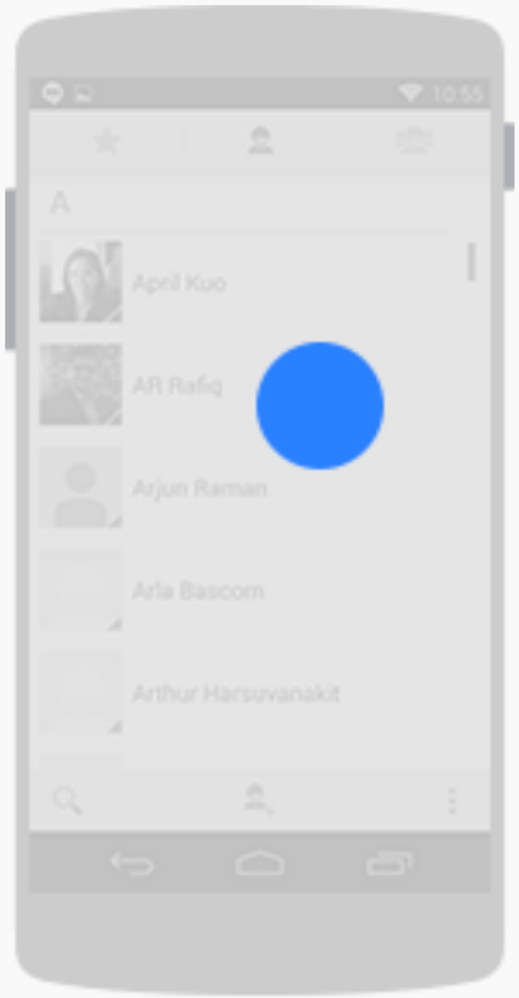
eLearning support unit

Application UI Structure

android

Agenda

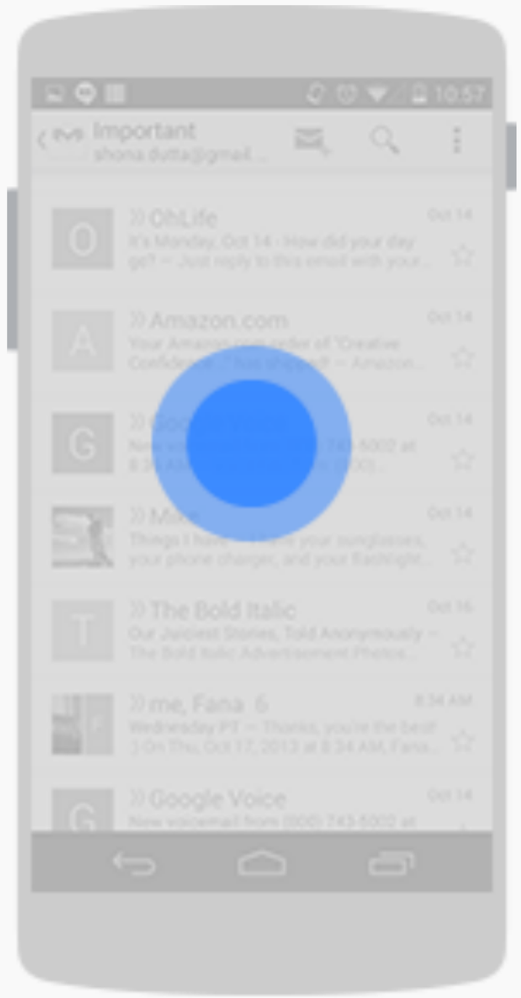
- Gestures
- Navigation Structure
- Top Level Content
- Top Level Switching
- Categories
- Details



Touch

Triggers the default functionality for a given item.

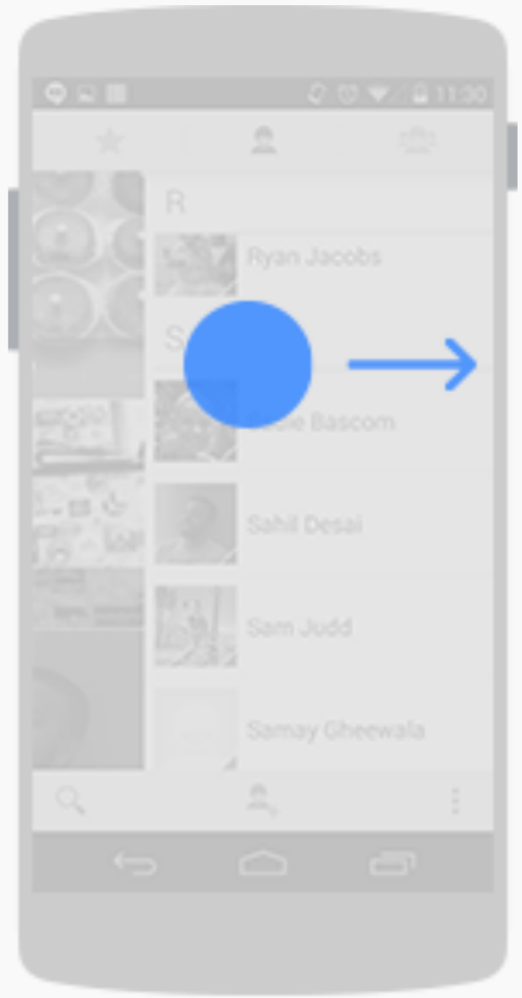
Action
Press, lift



Long press

Enters data selection mode. Allows you to select one or more items in a view and act upon the data using a contextual action bar. Avoid using long press for showing contextual menus.

Action
Press, wait, lift



Swipe or drag

Scrolls overflowing content, or navigates between views in the same hierarchy. Swipes are quick and affect the screen even after the finger is picked up. Drags are slower and more precise, and the screen stops responding when the finger is picked up.

Action
Press, move, lift

Long press drag

Rearranges data within a view, or moves data into a container (e.g. folders on Home Screen).

Action
Long press, move, lift

Double touch

Scales up a standard amount around the target with each repeated gesture until reaching maximum scale. For nested views, scales up the smallest targetable view, or returns it to its original scale. Also used as a secondary gesture for text selection.

Action
Two touches in quick succession

Double touch drag

Scales content by pushing away or pulling closer, centered around gesture.

Action
A single touch followed in quick succession by a drag up or down:

- Dragging up decreases content scale
- Dragging down increases content scale
- Reversing drag direction reverses scaling.

Pinch open

Zooms into content.

Action
2-finger press, move outwards, lift

Pinch close

Zooms out of content.

Action
2-finger press, move inwards, lift

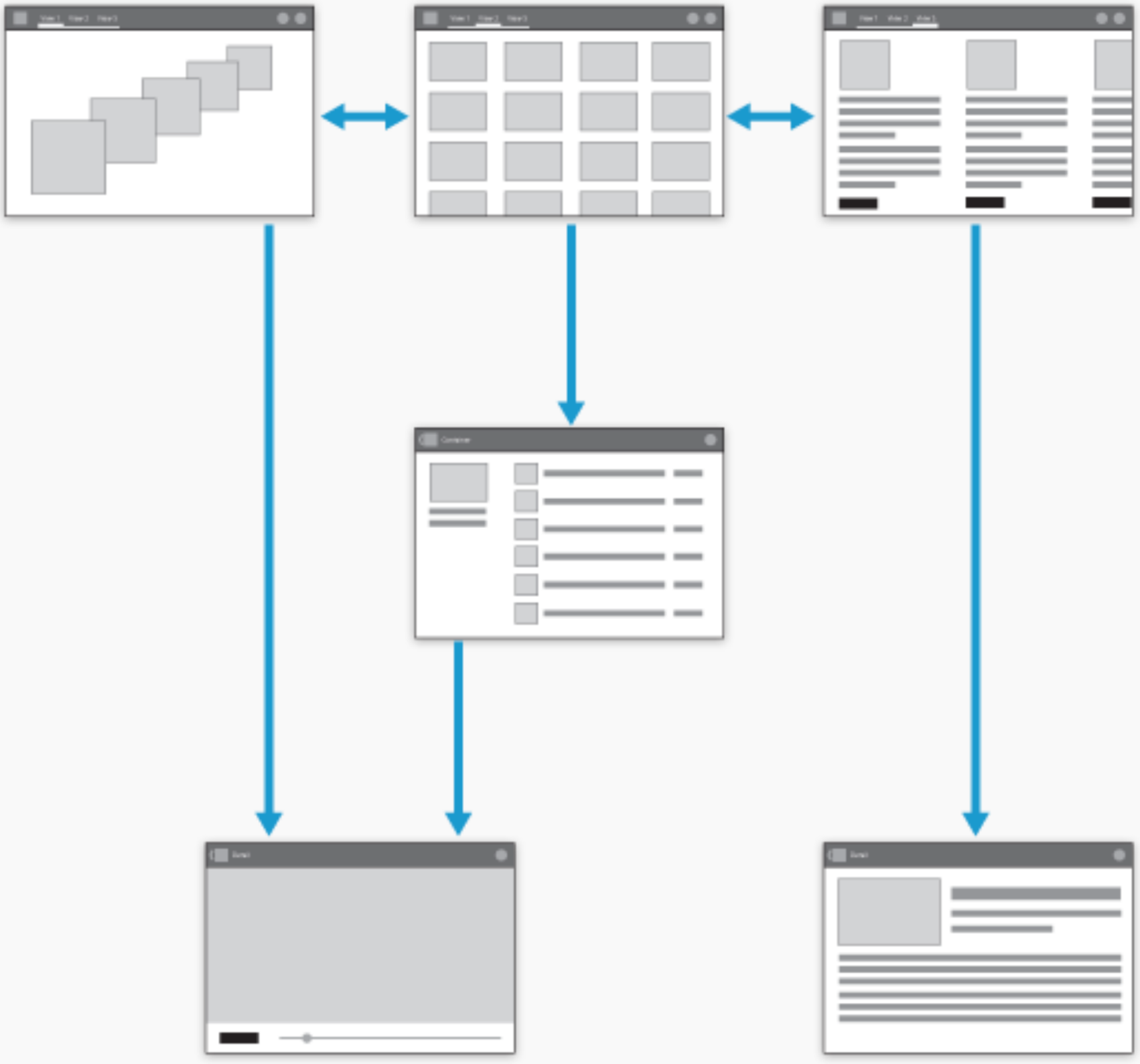
Example App Styles

App structure depends largely on the content and tasks you want to surface for your users.

- *Calculator or Camera*: single focused activity handled from a single screen

- *Phone*: main purpose is to switch between different activities without deeper navigation

- *Gmail:* combine a broad set of data views with deep navigation



Top level views

The top level of the app typically consists of the different views that your app supports. The views either show different representations of the same data or expose an altogether different functional facet of your app.

Category views

Category views allow you to drill deeper into your data.

Detail/edit view

The detail/edit view is where you consume or create data.

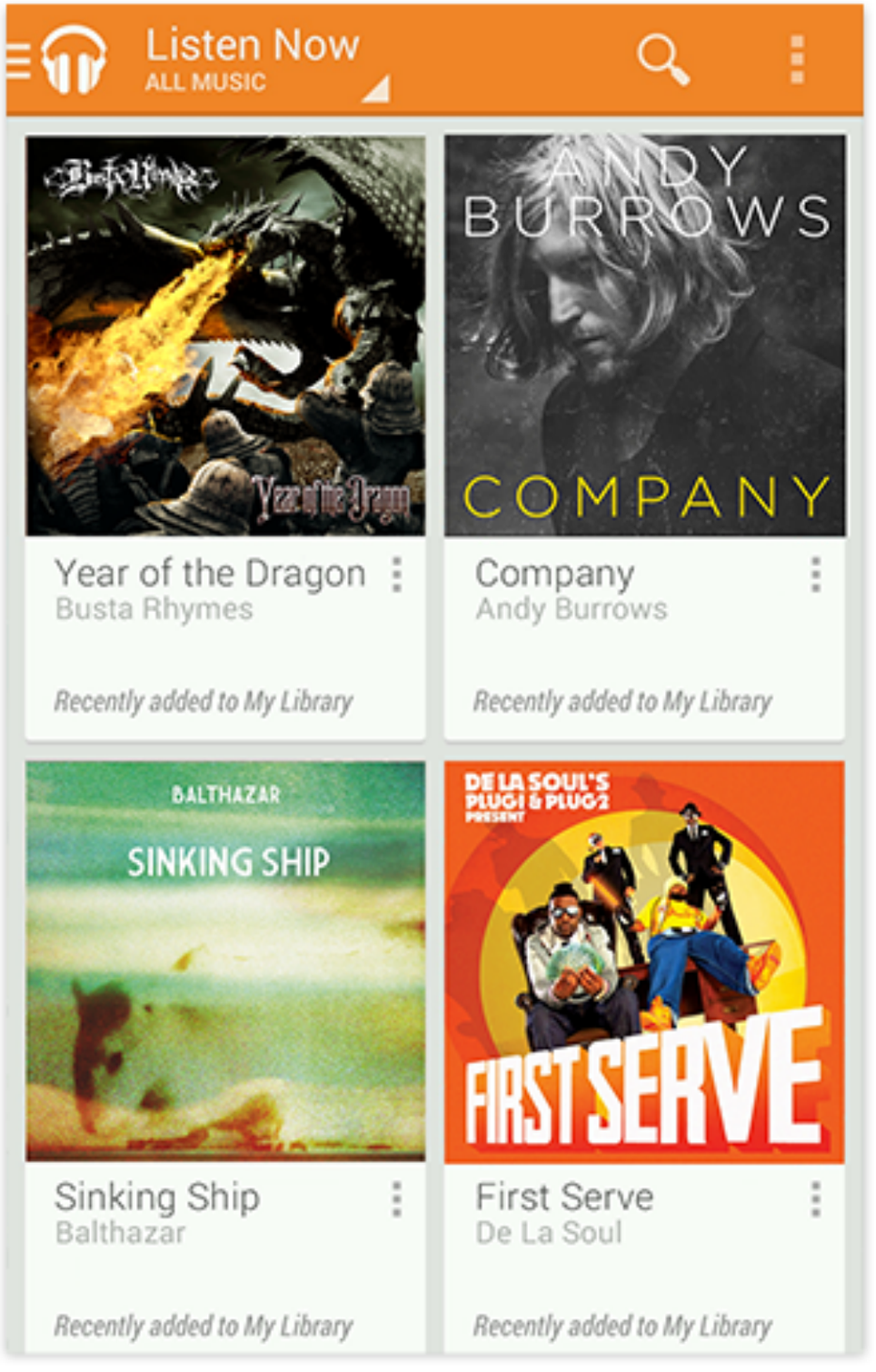
App Structure : Considerations

- Top Level Content
- Top Level Switching
- Categories
- Details

Top Level : Content

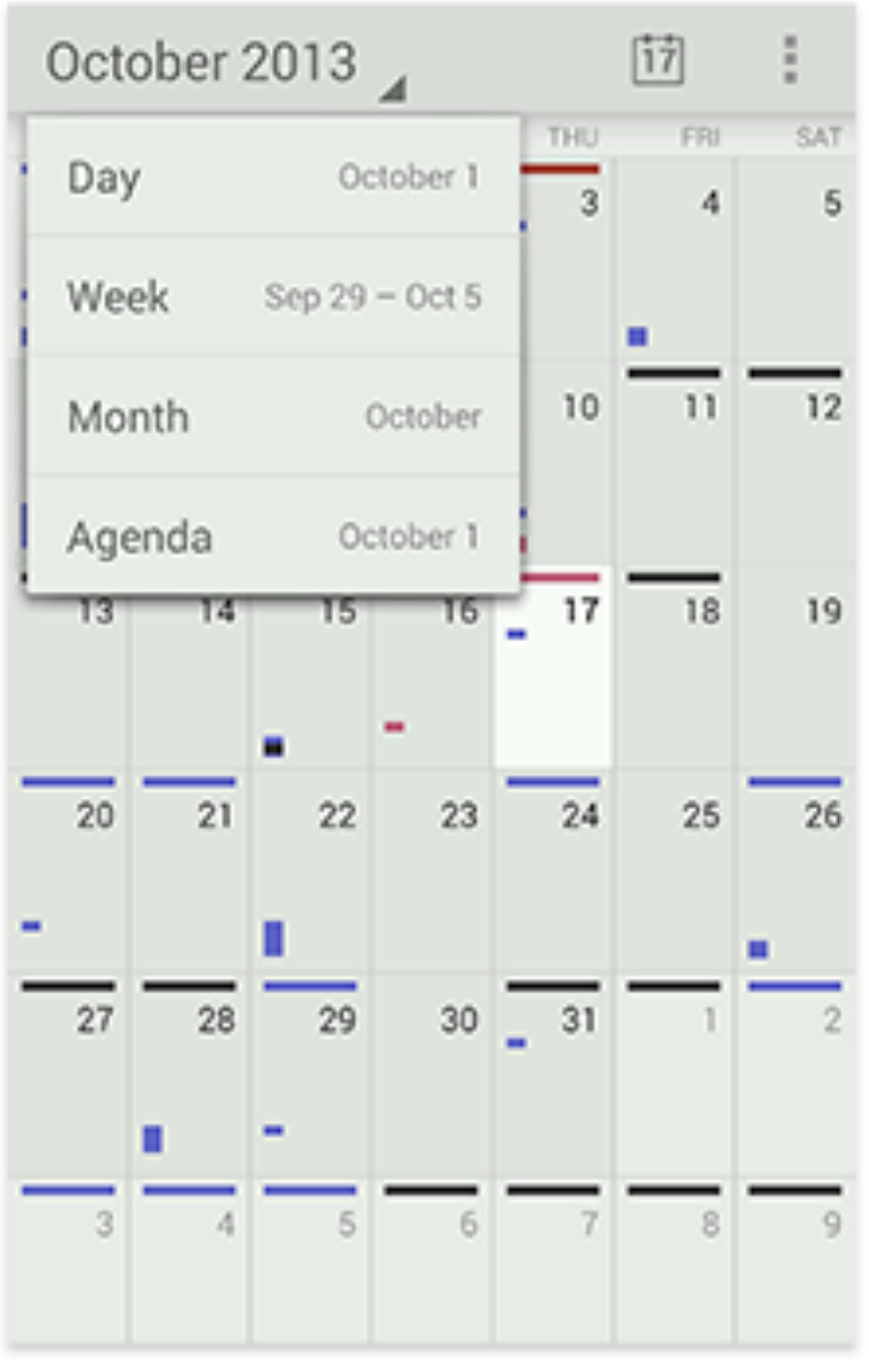
"What are my typical users most likely going to want to do in my app?"

- **Put content forward**
  - Focus on the content
  - Avoid navigation-only screens
  - Get to the core content of your app right away
  - Visually engaging content display and appropriate for the data type and screen size



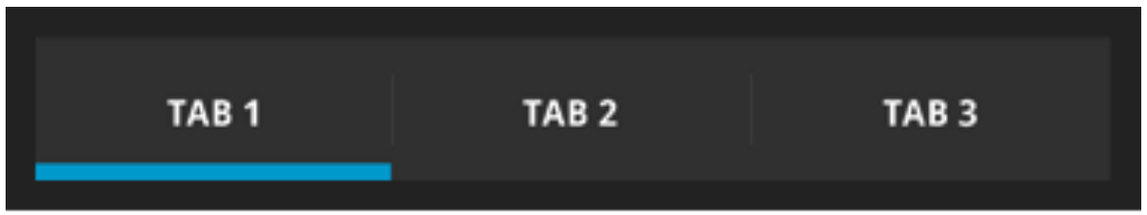
Top Level: Navigation

- Display action bars to provide consistent navigation…
- In the action bar:
  - Display your app's icon or title.
  - If multiple views, make it easy for the user to navigate between views by adding view controls.
  - If app allows content creation, make the content accessible right from the top level.
  - If content searchable, include the Search action in the action bar



Top Level Switching: Fixed Tabs

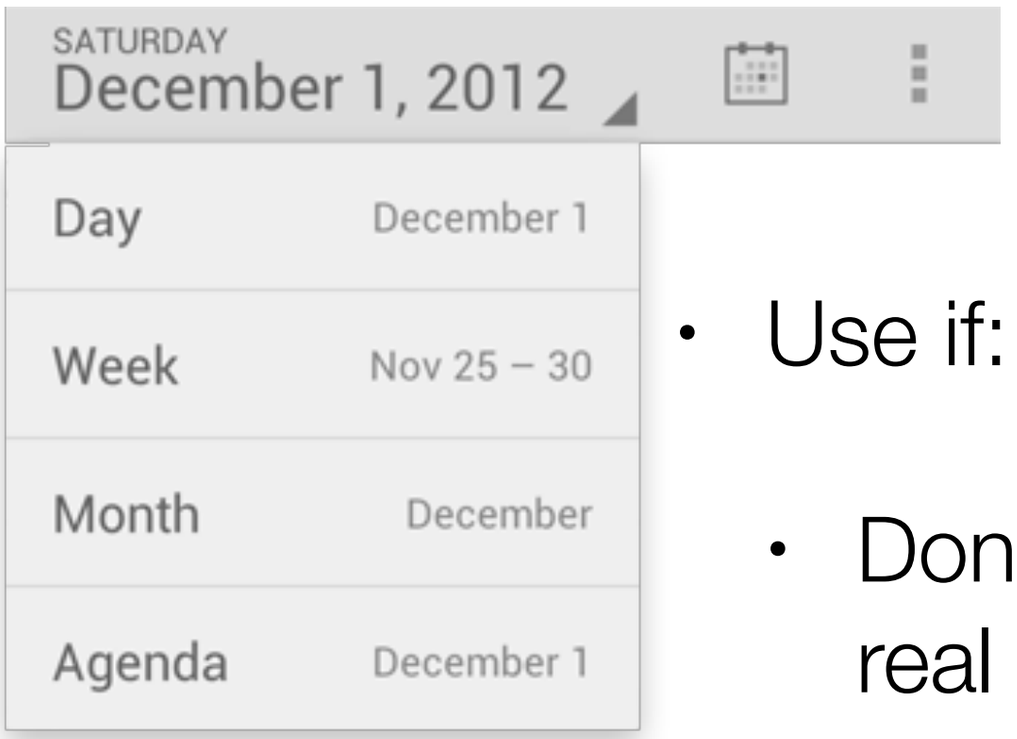
- Fixed tabs display top-level views concurrently and make it easy to explore and switch between them.
- They are always visible on the screen, and can't be moved out of the way like scrollable tabs.
- Fixed tabs should always allow the user to navigate between the views by swiping left or right on the content area.



- Use if:
  - You expect your app's users to switch views frequently.
  - You have a limited number of up to three top-level views.
  - You want the user to be highly aware of the alternate views.

Top Level Switching: Spinners

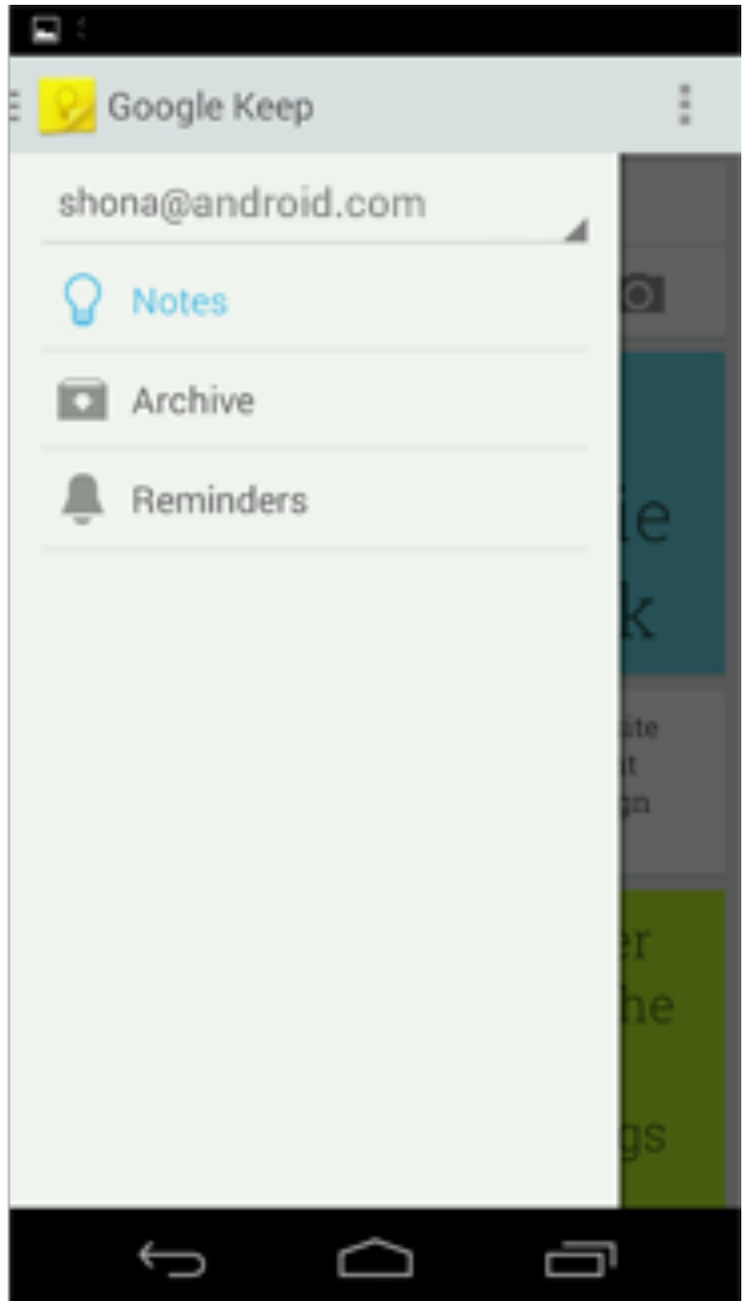
- A drop-down menu that allows users to switch between views



- Use if:
  - Don't want to give up the vertical screen real estate for a dedicated tab bar.
  - The user is switching between views of the same data set (for example: calendar events viewed by day, week, or month)

Top Level Switching: Navigation Drawer

- A slide-out menu that allows users to switch between views
- Can hold a large number of items and is accessible from anywhere in the app.
- Show the app's top-level views, but can also provide navigation to lower-level screens.
- Use if:
  - Don't want to give up the vertical screen real estate for a dedicated tab bar.
  - Have a large number of top-level views.
  - Provide direct access to screens on lower levels.
  - Provide quick navigation to views which don't have direct relationships between each other.



Categories

- The purpose of a deep, data-driven app is to navigate through organizational categories to the detail level.

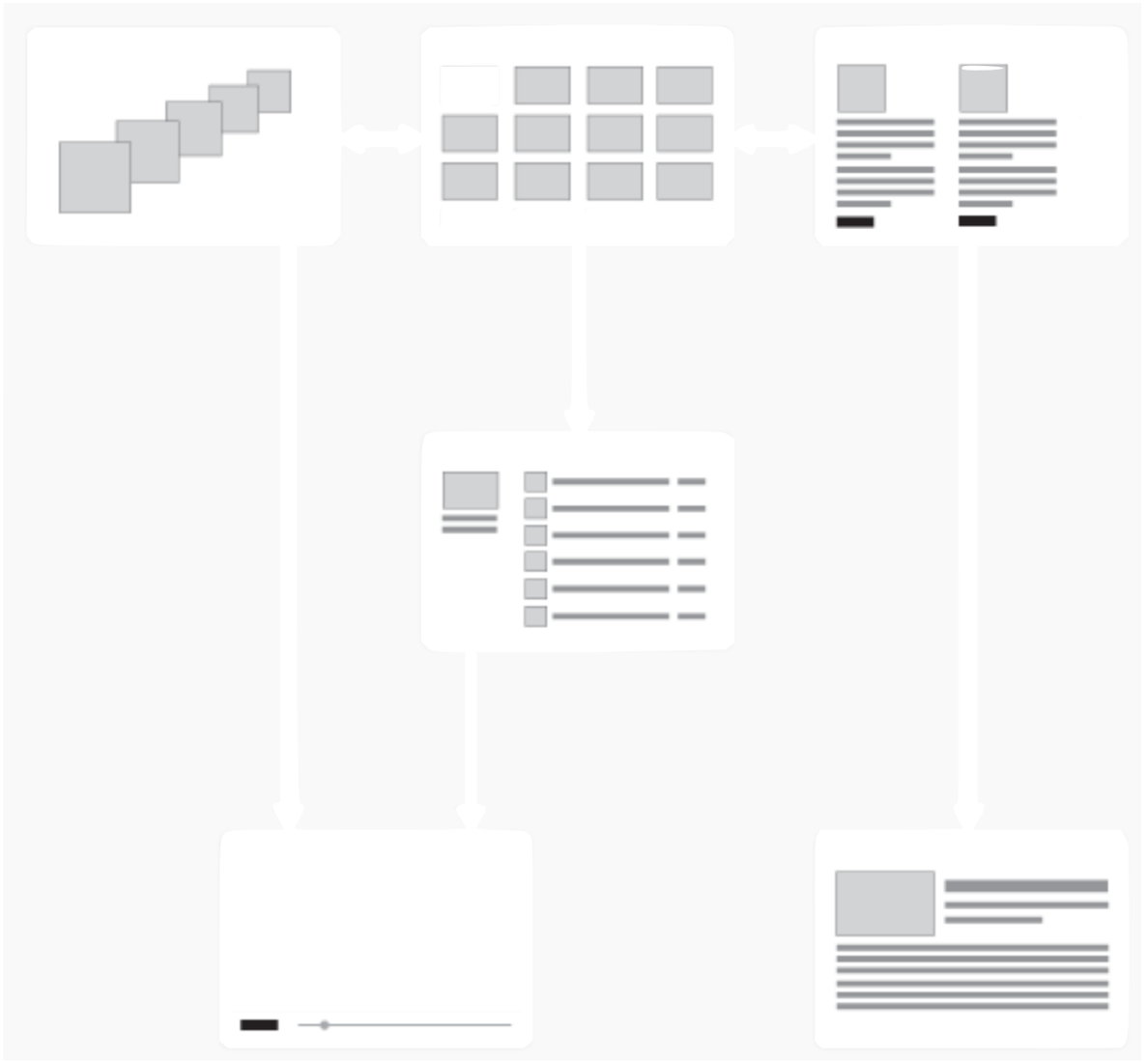
- Minimize perceived navigation effort by keeping app shallow.

- Cut down on the perception of onerous navigation.

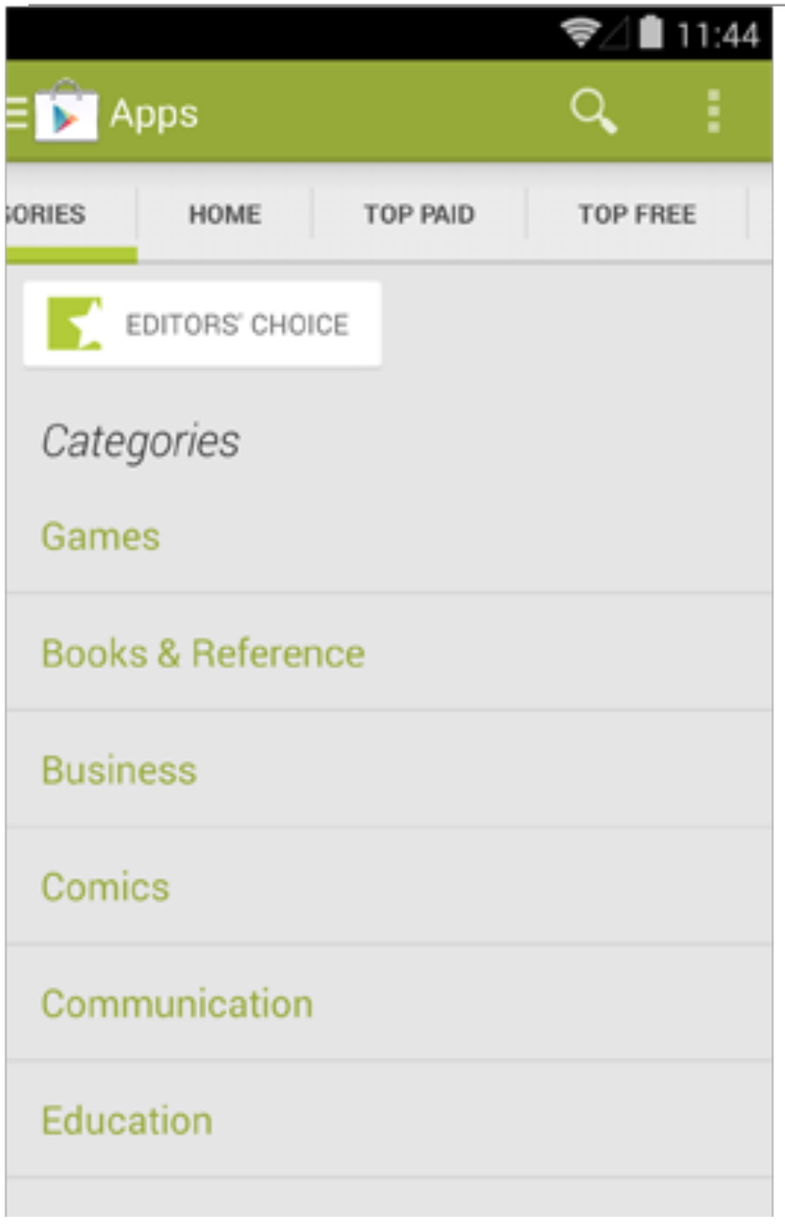
  - Use tabs to combine category selection and data display

  - Allow cutting through hierarchies

  - Acting upon multiple data items



Categories: Use tabs to combine category selection and data

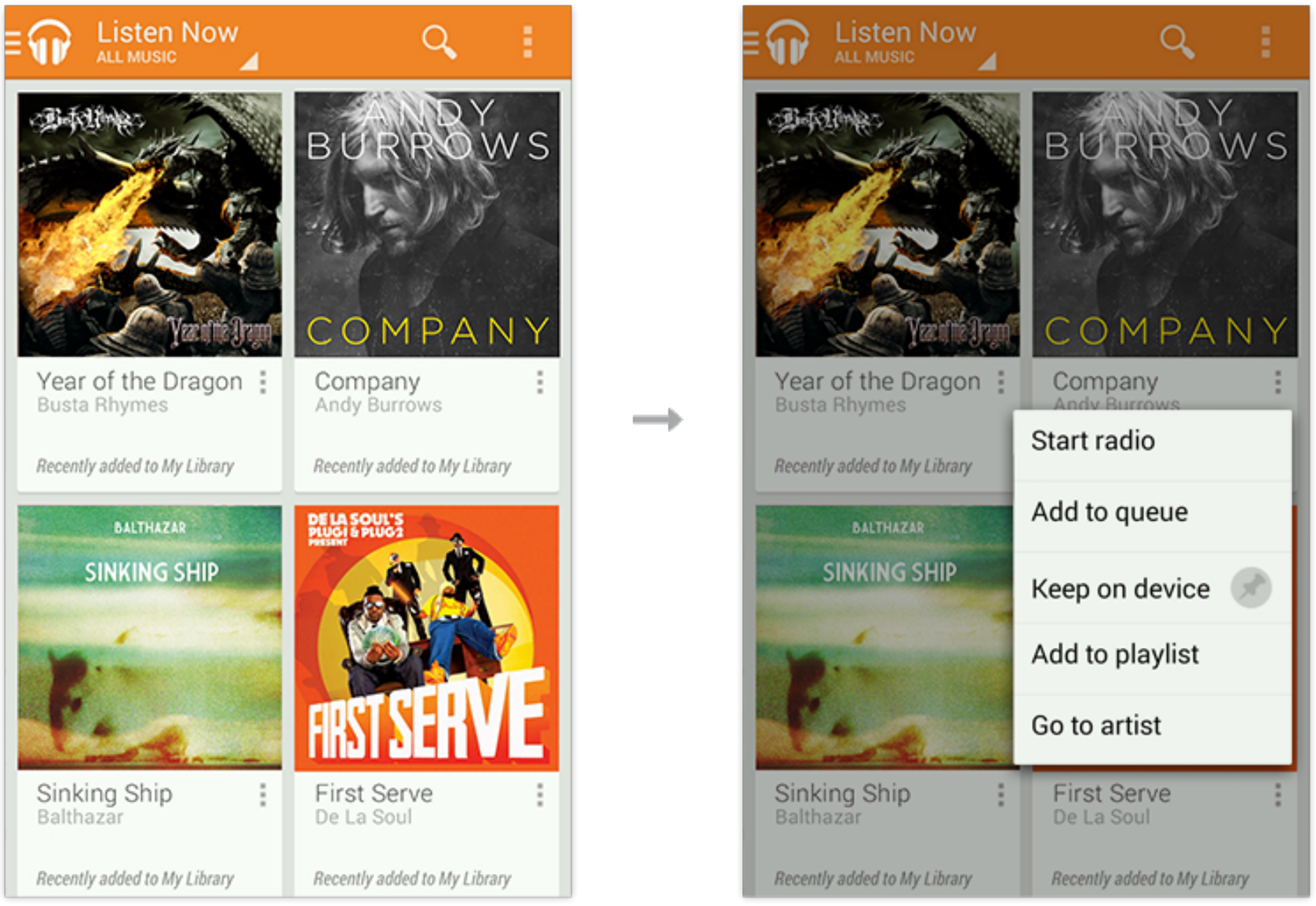


- Can work if the categories are familiar or the number of categories is small.
- Removes a level of hierarchy and data remains at the center of the user's attention.
- Navigating laterally between data-rich categories is more akin to a casual browsing experience than to an explicit navigation step.

Categories: Allow cutting through hierarchies

- Take advantage of shortcuts that allow people to reach their goals quicker.

- Display prominent actions directly on list view items using drop-downs or split list items.



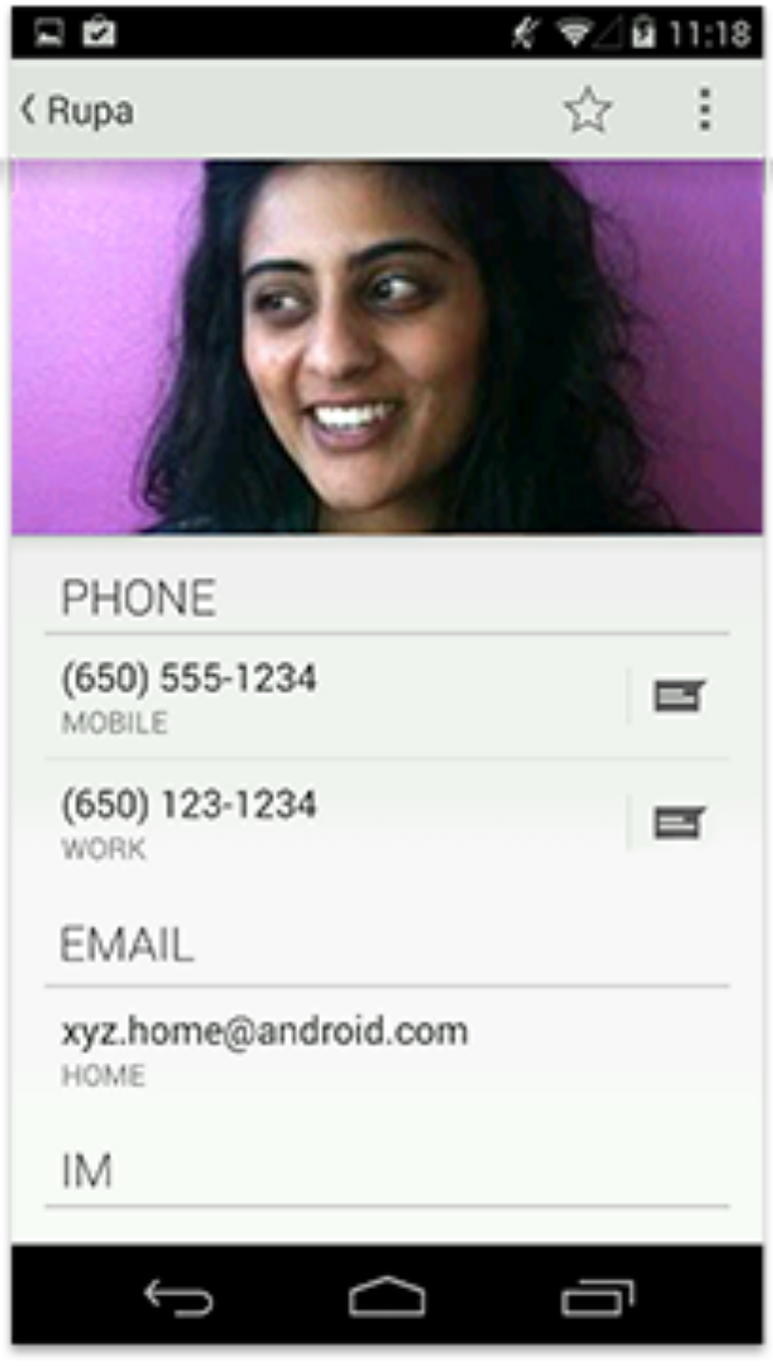
- Lets people invoke actions on data without having to navigate all the way down the hierarchy.

Categories: Acting upon multiple data items

- There are often good reasons to act on collections of data as well.
    - E.g: allow deletion of an item in a detail view, also allow deletion of multiple items in the category view.
- Analyze which detail view actions are applicable to collections of items.
    - Then use multi-select to allow application of those actions to multiple items in a category view.

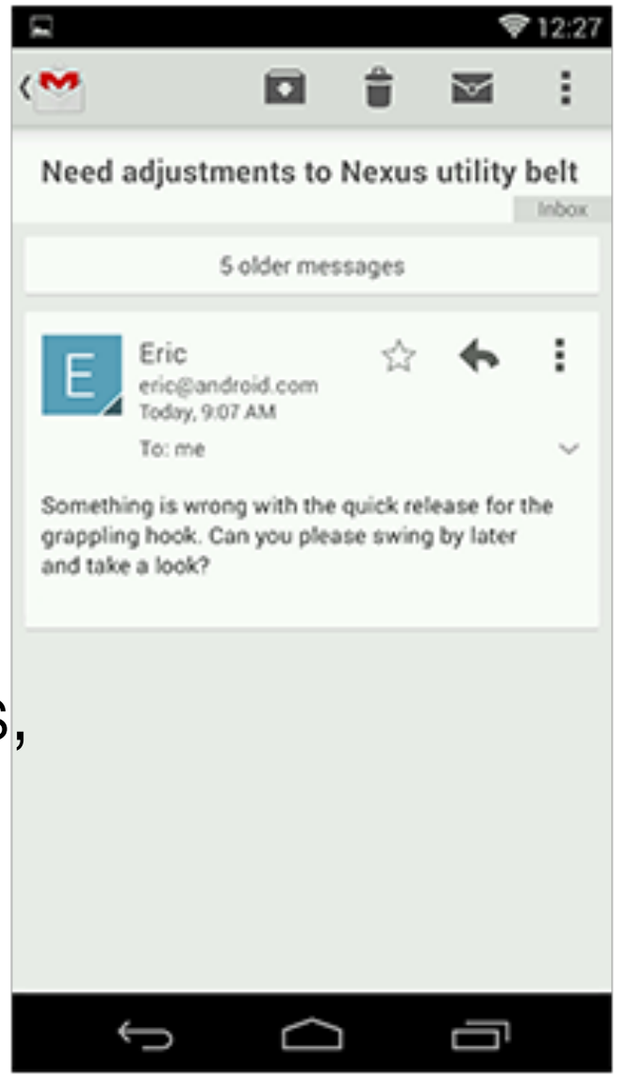
Details : Layout

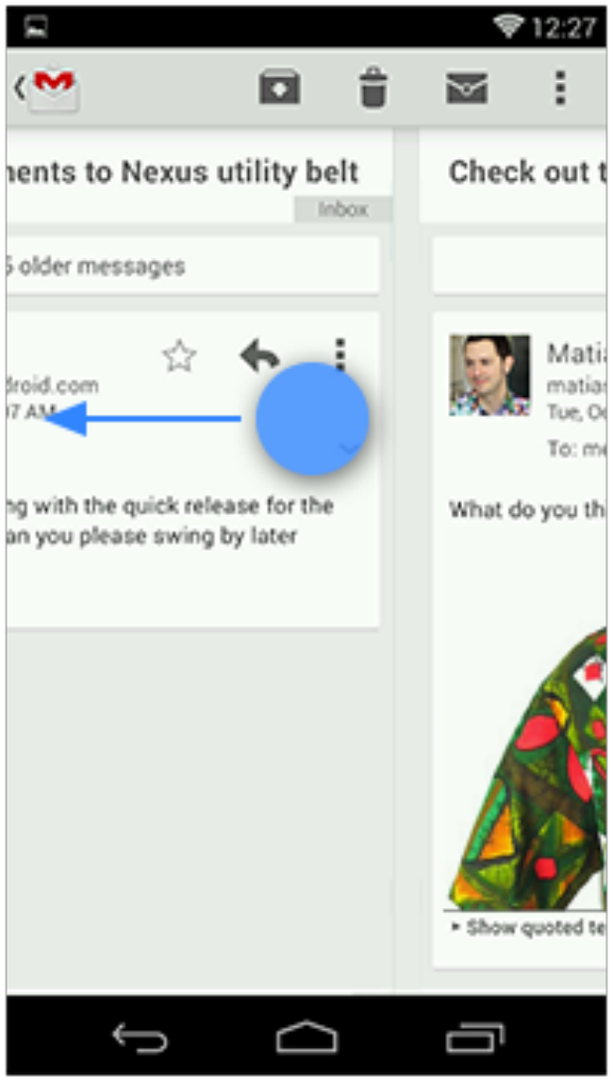
- Allows user to view and act on data.
- Depends on the data type being displayed
- Consider the activities people will perform in the detail view and arrange the layout accordingly.

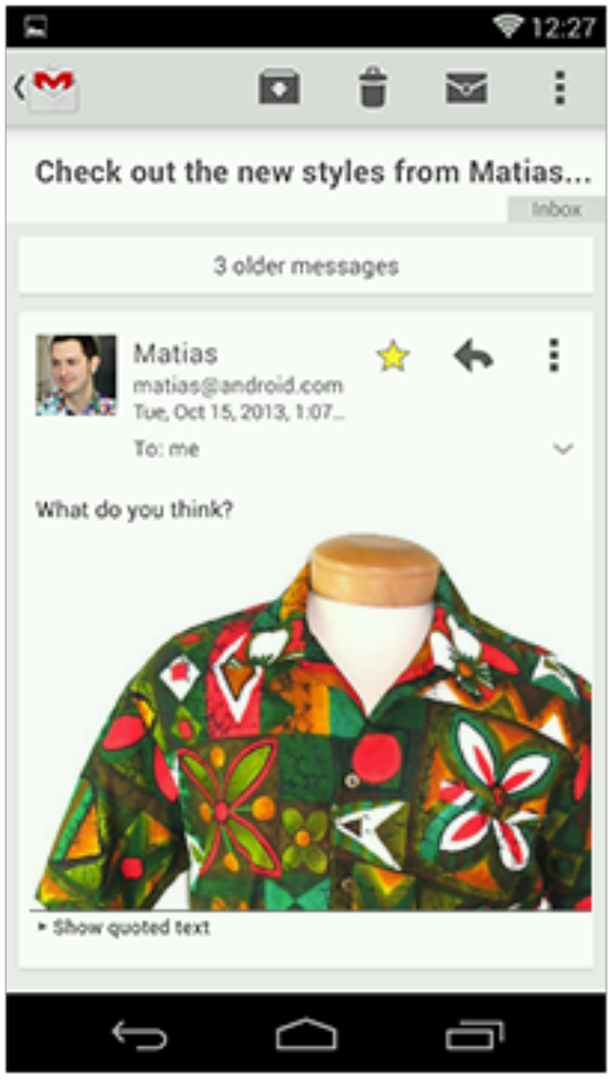


Details: Navigation

- Make navigation between detail views efficient
- Consider allowing user to navigate between items from within the detail view.
- Use swipe views or other techniques, such as thumbnail view controls, to achieve this.







App Structure Checklist

- Find ways to display useful content on your start screen.
- Use action bars to provide consistent navigation.
- Keep your hierarchies shallow by using horizontal navigation and shortcuts.
- Use multi-select to allow the user to act on collections of data.
- Allow for quick navigation between detail items with swipe views.

Much of the material in this slide deck is adapted from http://developer.android.com/