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
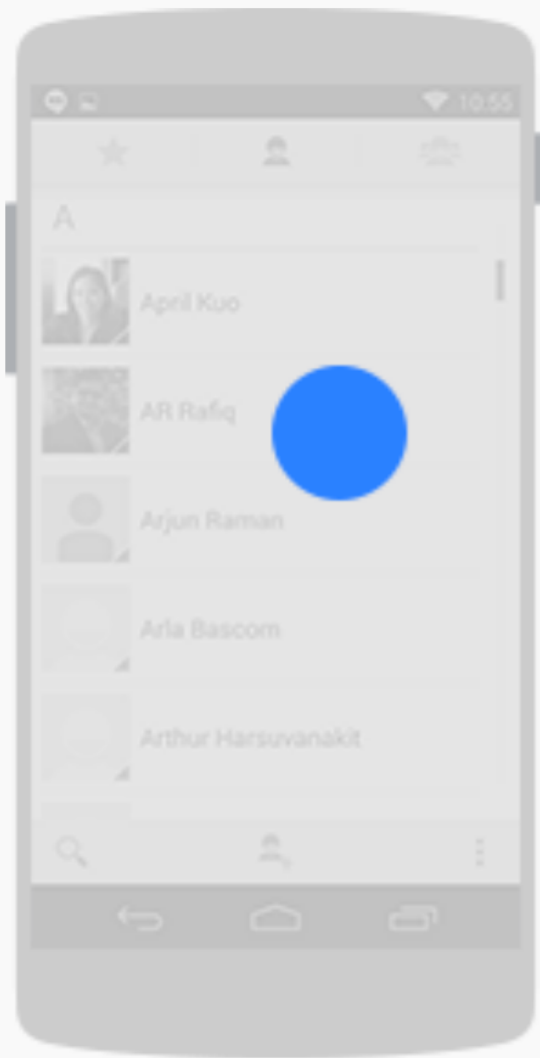
android

Agenda

- Gestures

- App Structure

- Top Level Content

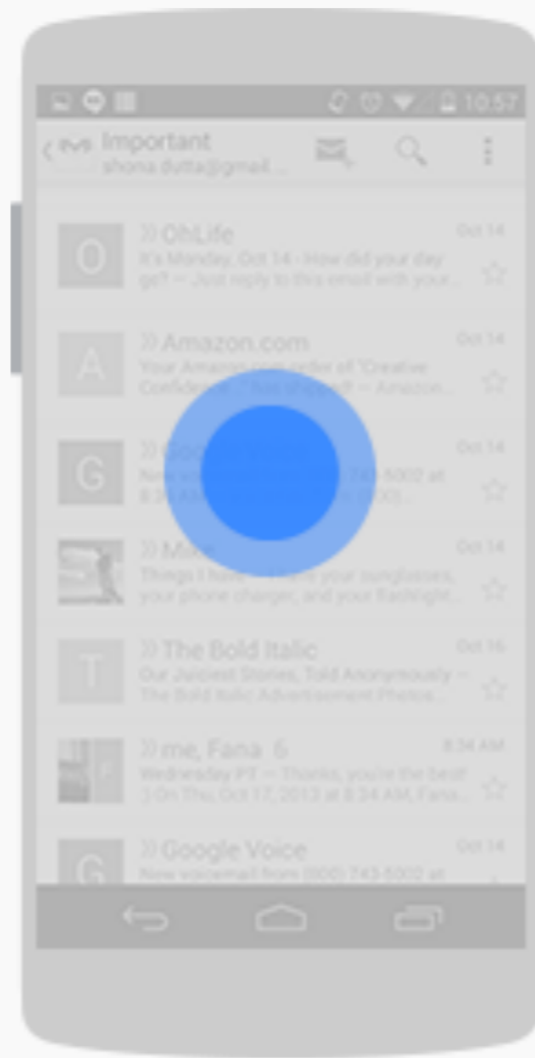- Top Level Switching

- Categories

- Details

Touch

Triggers the default functionality for a given item.
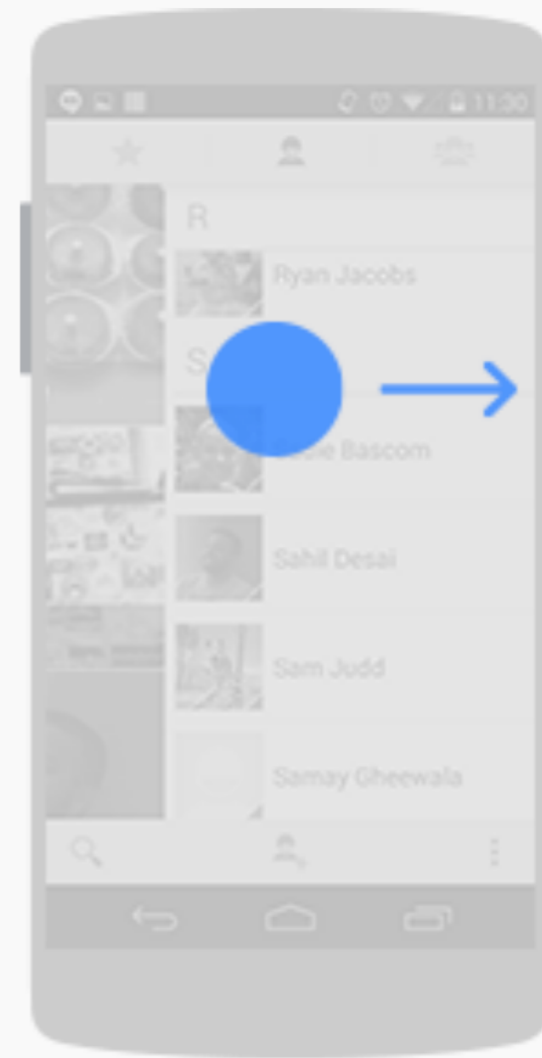
👆 **Action**
Press, lift

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.
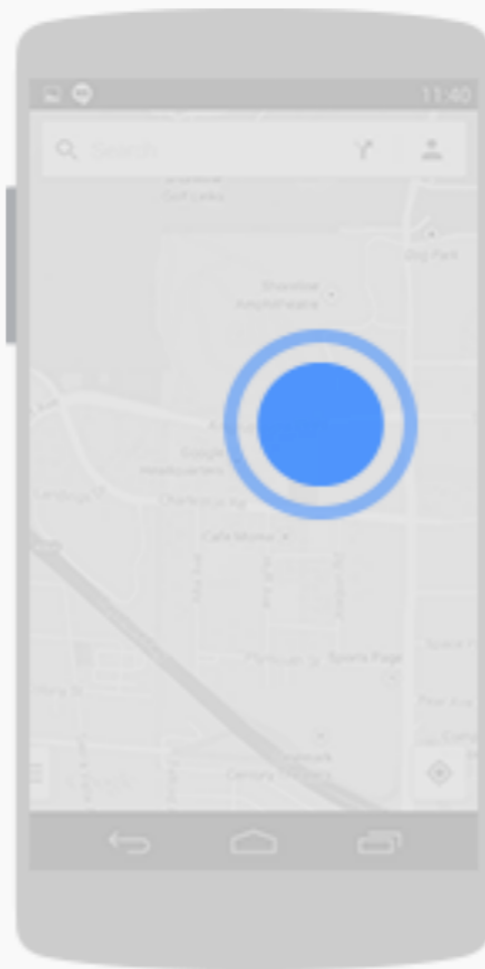
👆 **Action**
Press, move, lift

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

App Structure

- Apps come in many varieties that address very different needs. For example:

  - Apps such as Calculator or Camera that are built around a single focused activity handled from a single screen

  - Apps such as Phone whose main purpose is to switch between different activities without deeper navigation

  - Apps such as Gmail or the Play Store that combine a broad set of data views with deep navigation

- Your app's structure depends largely on the content and tasks you want to surface for your users.
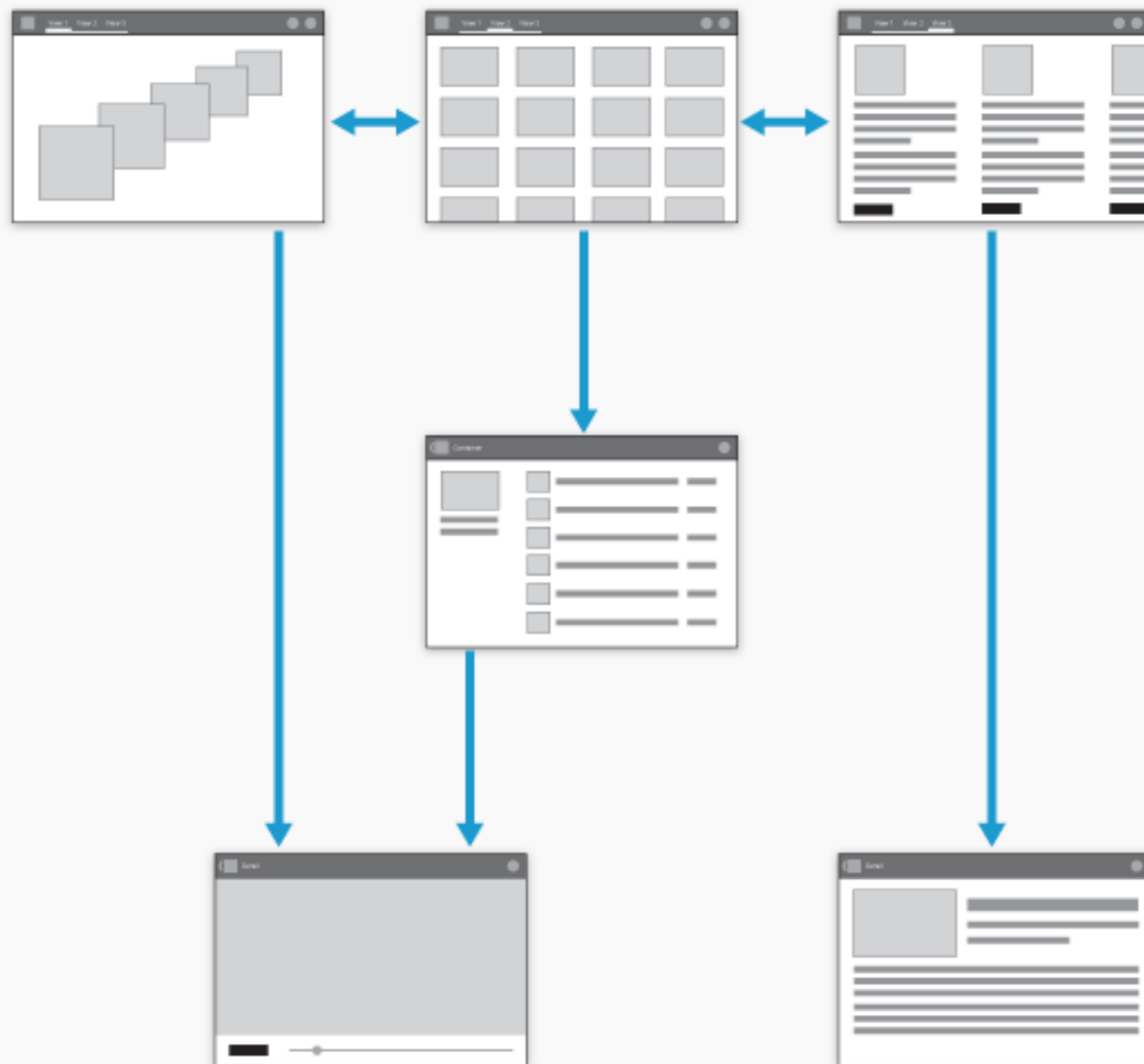
Top level views

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

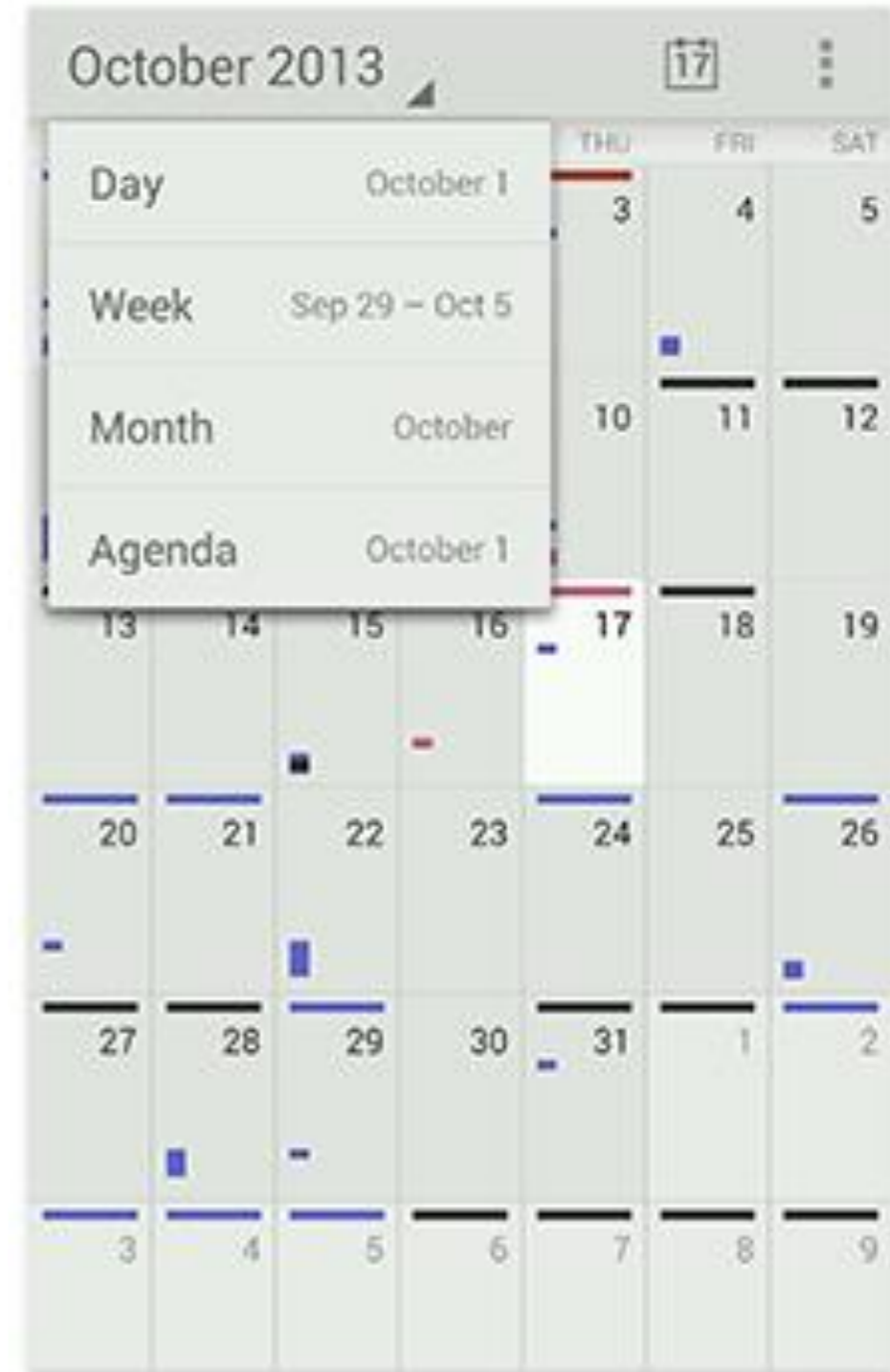- Top Level Switching

- Categories

- Details

Top Level : Content

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

- **Put content forward**

  - Many apps focus on the content display.

  - Avoid navigation-only screens and instead let people get to the meat of your app right away by making content the centerpiece of your start screen.

  - Choose layouts that are visually engaging and appropriate for the data type and screen size
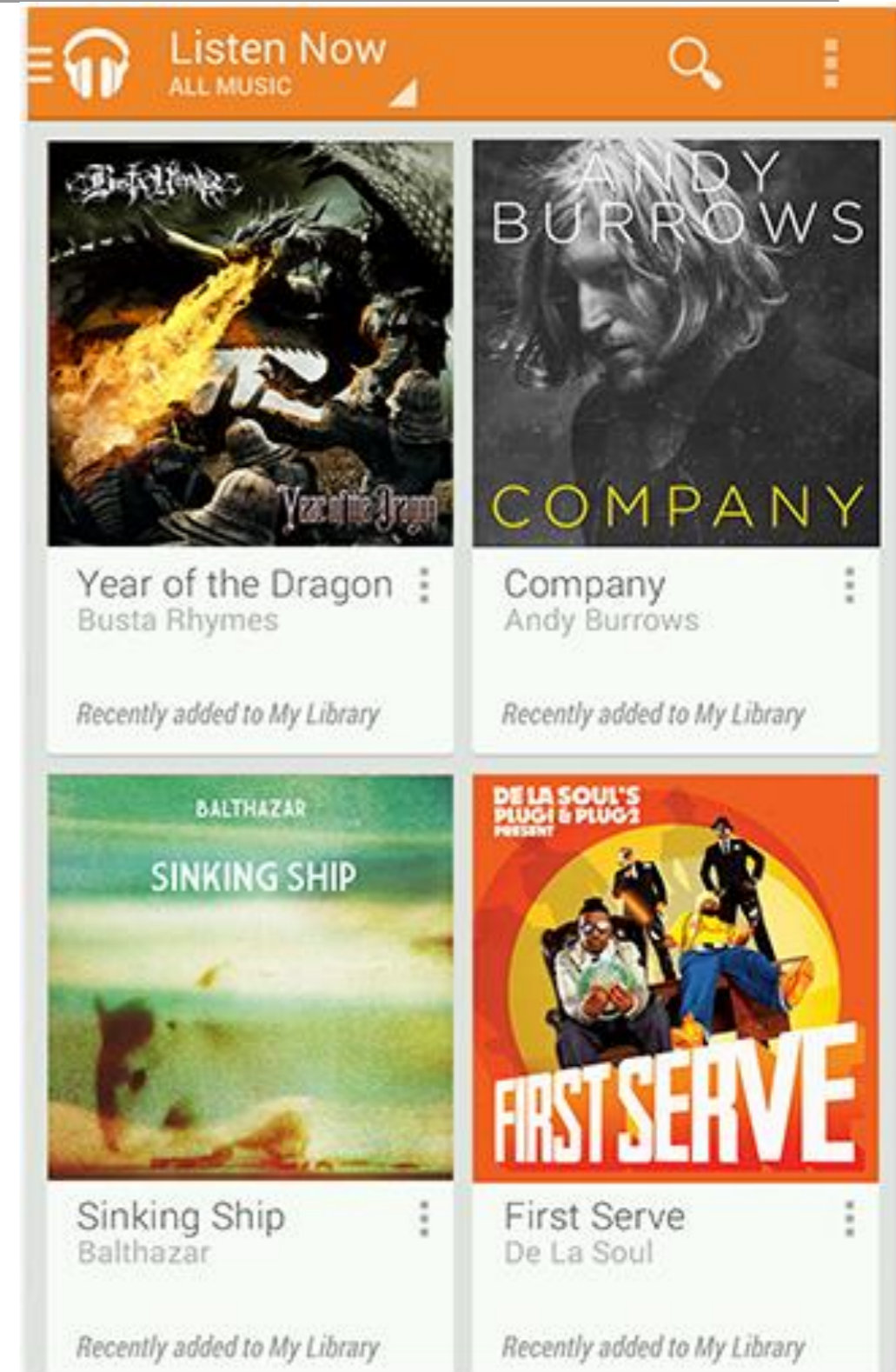
Top Level: Navigation

- All screens in your app should display action bars to provide consistent navigation and surface important actions.

- At the top level, special considerations apply to the action bar:

  - Use the action bar to display your app's icon or title.

  - If your top level consists of multiple views, make sure that it's easy for the user to navigate between them by adding view controls to your action bar.

  - If your app allows people to create content, consider making the content accessible right from the top level.

  - if your content is searchable, include the Search action in the action bar so people can cut through the navigation hierarchy

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.
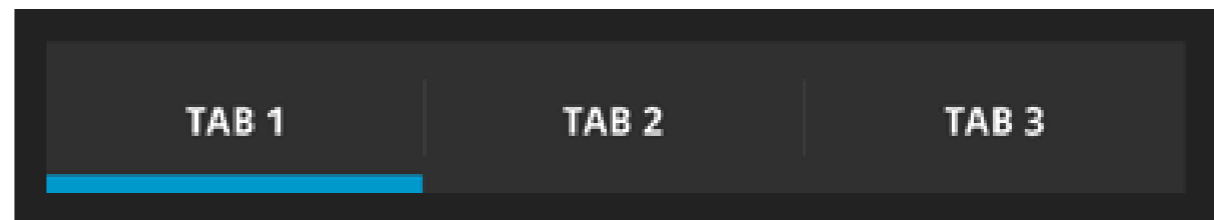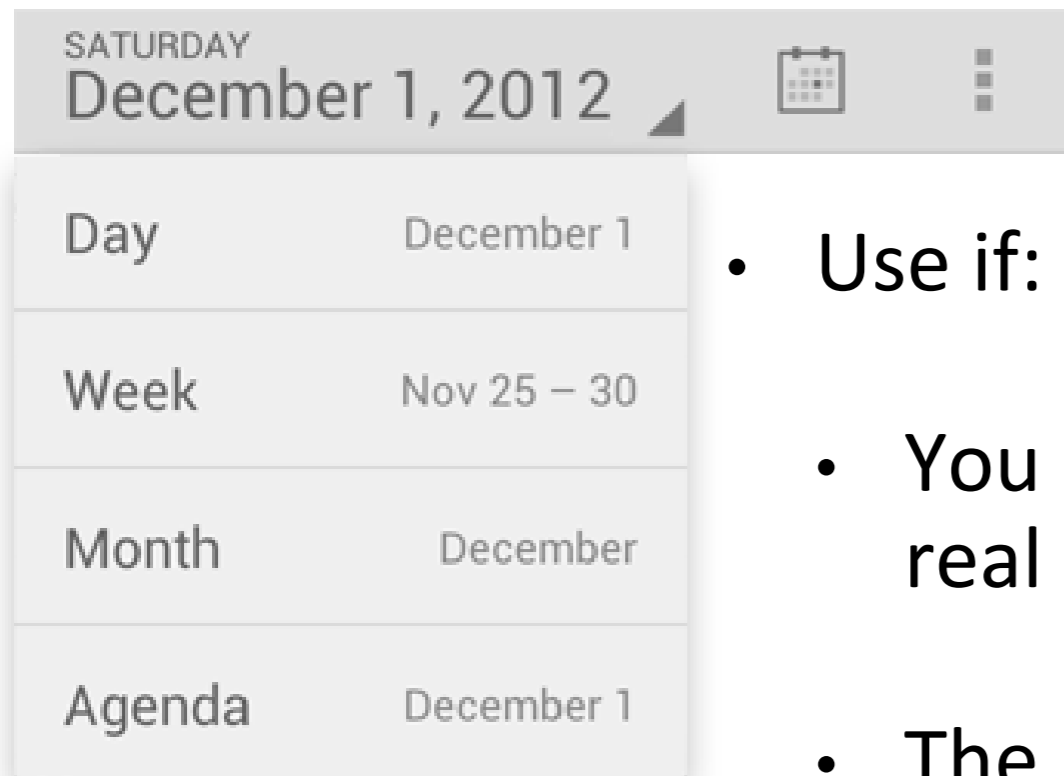

- Use if:

  - 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 spinner is a drop-down menu that allows users to switch between views of your app.

SATURDAY
December 1, 2012 ◢ 📅 ⋮

Day	December 1
Week	Nov 25 – 30
Month	December
Agenda	December 1

- Use if:

  - You 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) or data sets of the same type (such as content for two different accounts).

Top Level Switching: Navigation Drawer

- A navigation drawer is a slide-out menu that allows users to switch between views of your app. It can hold a large number of items and is accessible from anywhere in your app. Navigation drawers show your app's top-level views, but can also provide navigation to lower-level screens. This makes them particularly suitable for complex apps.

- Use if:

  - You don't want to give up the vertical screen real estate for a dedicated tab bar.

  - You have a large number of top-level views.

  - You want to provide direct access to screens on lower levels.

  - You want to provide quick navigation to views which don't have direct relationships between each other.

 - You have particularly deep navigation branches.
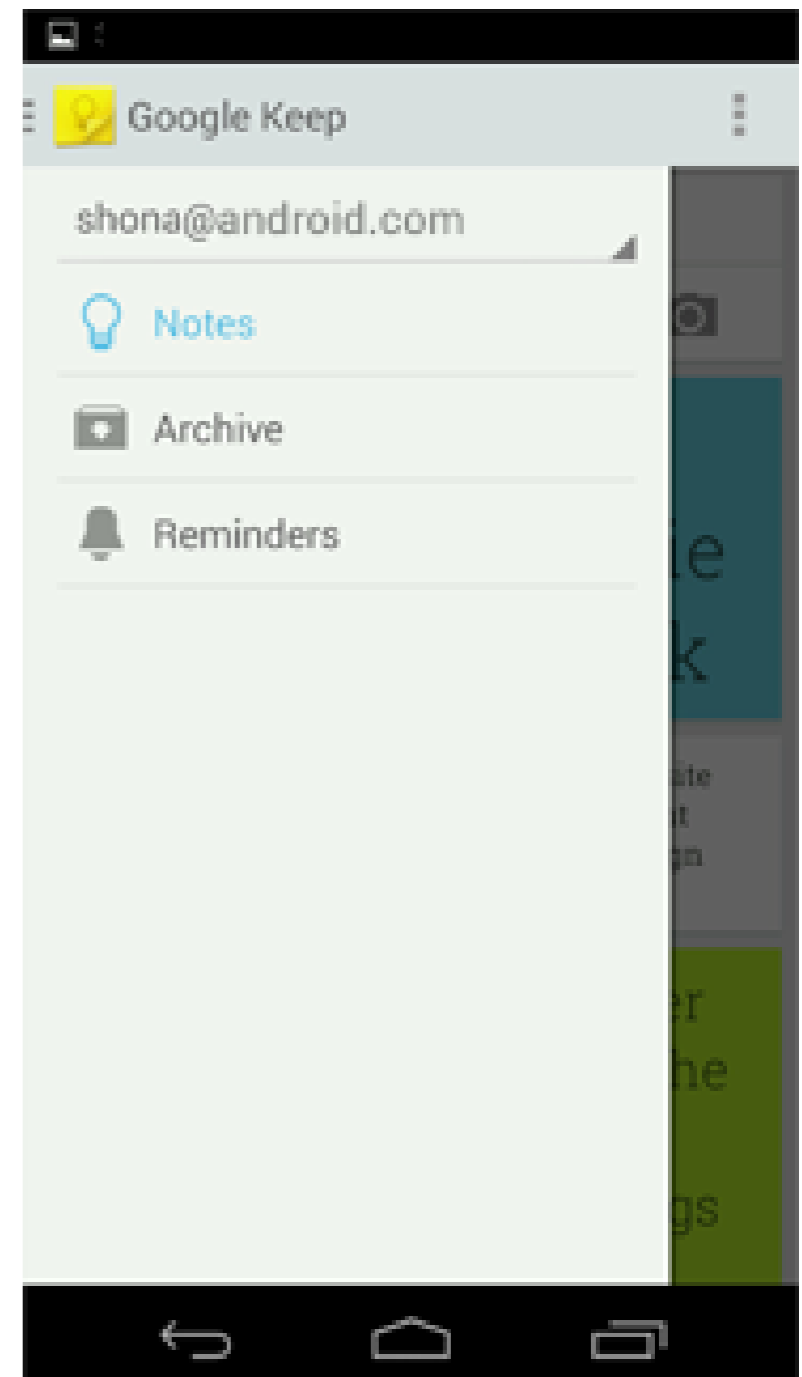
Categories

- Generally, the purpose of a deep, data-driven app is to navigate through organizational categories to the detail level, where data can be viewed and managed.

- Minimize perceived navigation effort by keeping your apps shallow.

- Even though the number of vertical navigation steps from the top level down to the detail views is typically dictated by the structure of your app's content, there are several ways you can 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
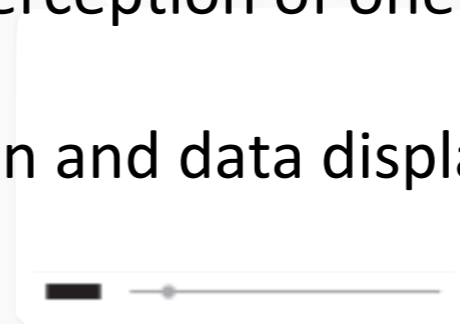
# Categories: Use tabs to combine category selection and data
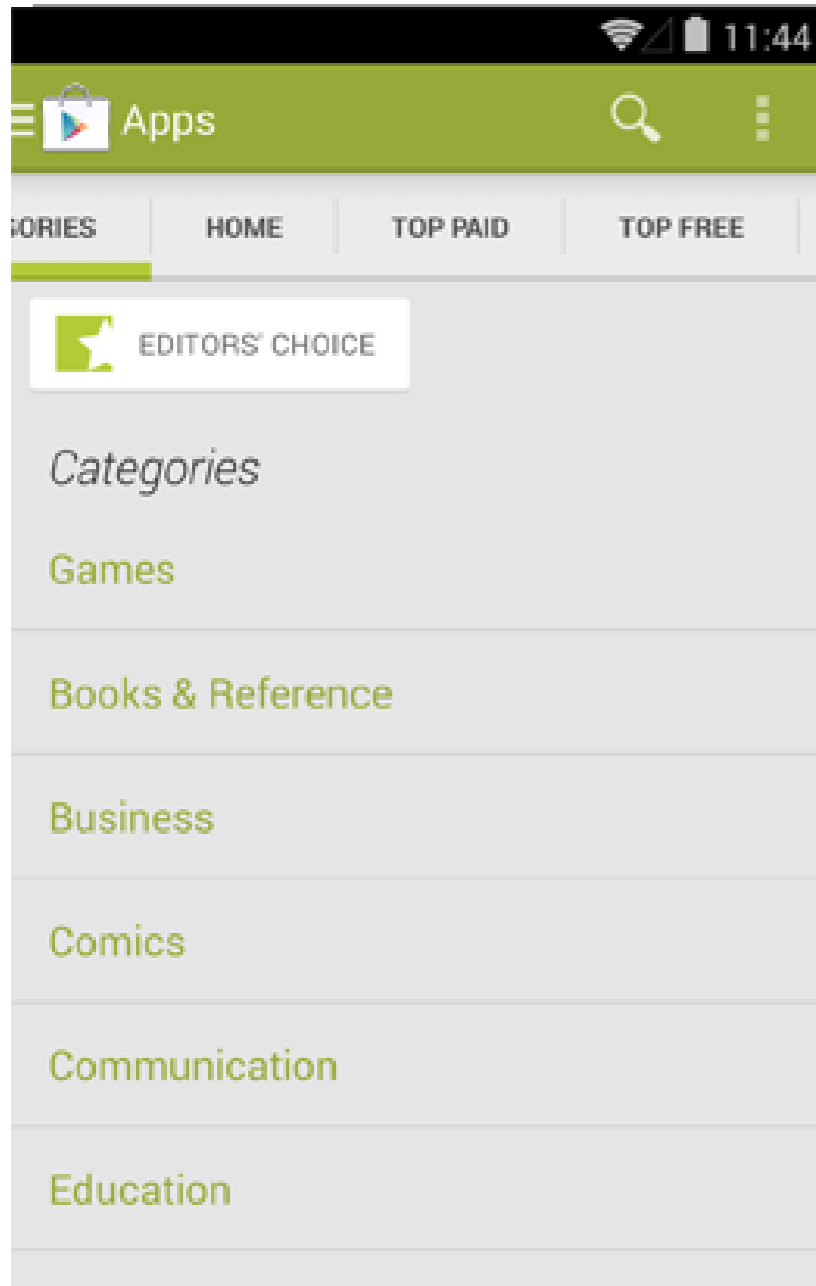
- This can be successful if the categories are familiar or the number of categories is small.

- It has the advantage that a level of hierarchy is removed 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.

- To allow top-level invocation of actions for a data item from within list or grid views, display prominent actions directly on list view items using drop-downs or split list items.
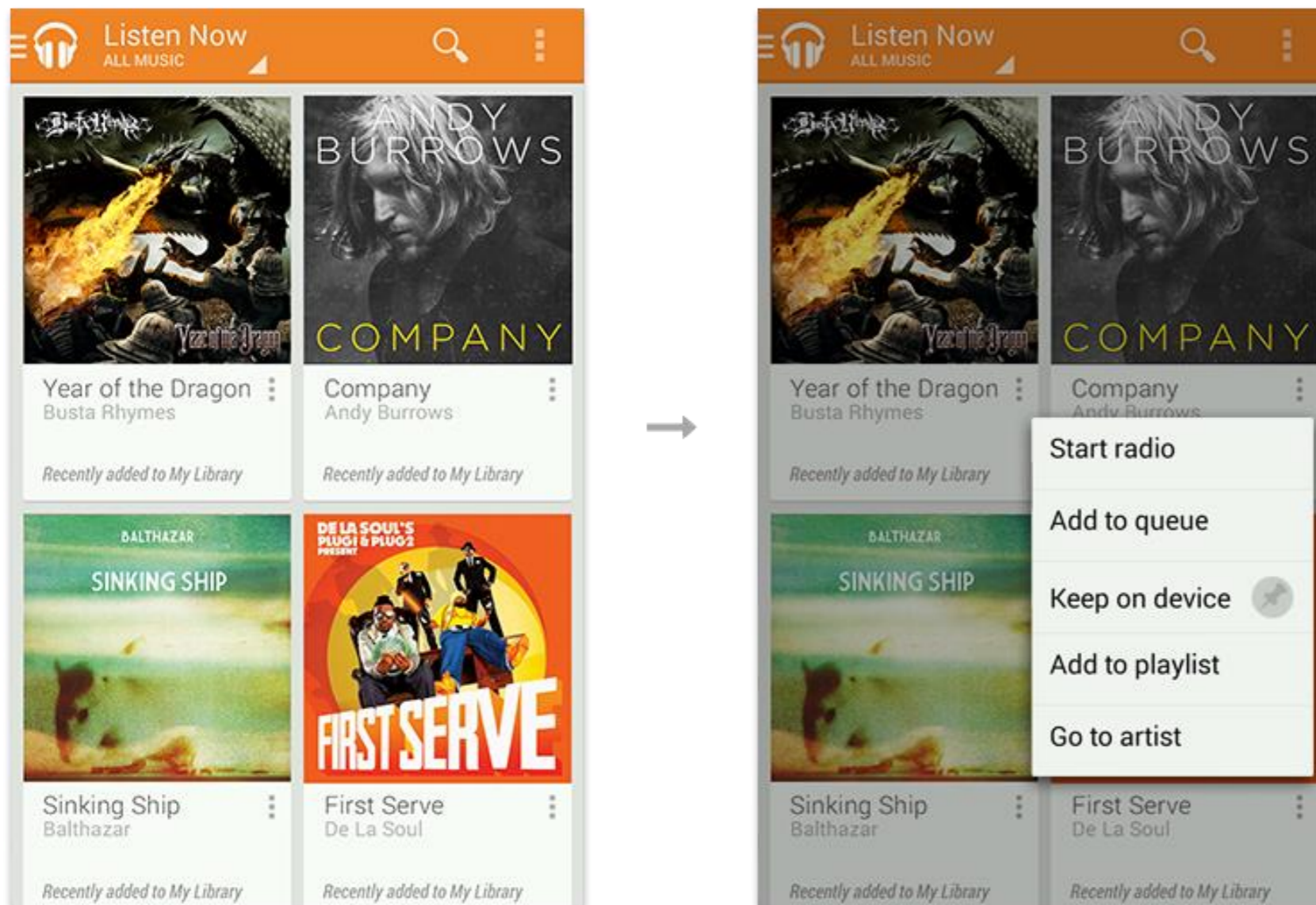
- This lets people invoke actions on data without having to navigate all the way down the hierarchy.

Categories: Acting upon multiple data items

- Even though category views mostly serve to guide people to content detail, keep in mind that there are often good reasons to act on collections of data as well.

- For example, if you allow people to delete an item in a detail view, you should also allow them to delete 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

- The detail view allows you to view and act on your data.

- The layout of the detail view depends on the data type being displayed, and therefore differs widely among apps.

- Consider the activities people will perform in the detail view and arrange the layout accordingly.

Details: Navigation

- Make navigation between detail views efficient

- If your users are likely to want to look at multiple items in sequence, allow them to navigate between items from within the detail view.

- Use swipe views or other techniques, such as thumbnail view controls, to achieve this.
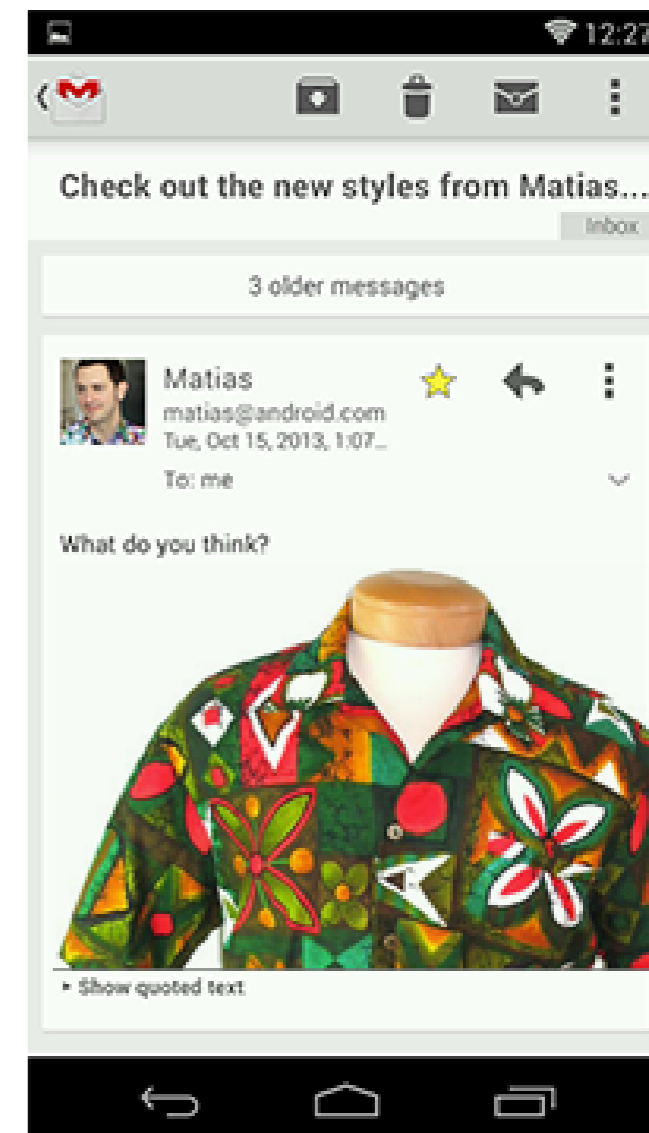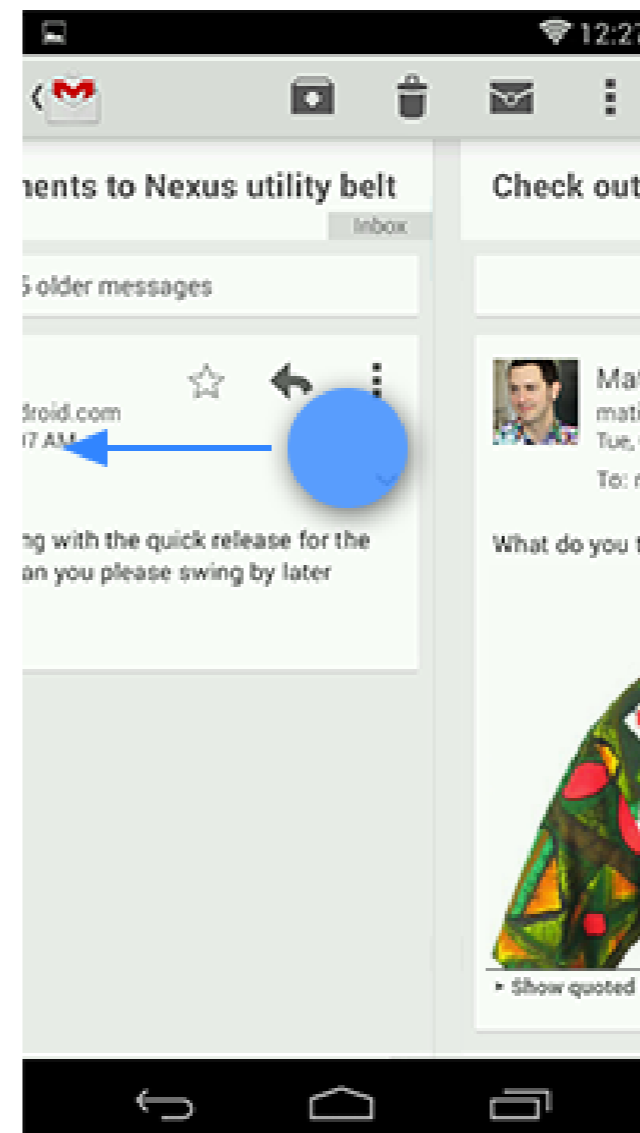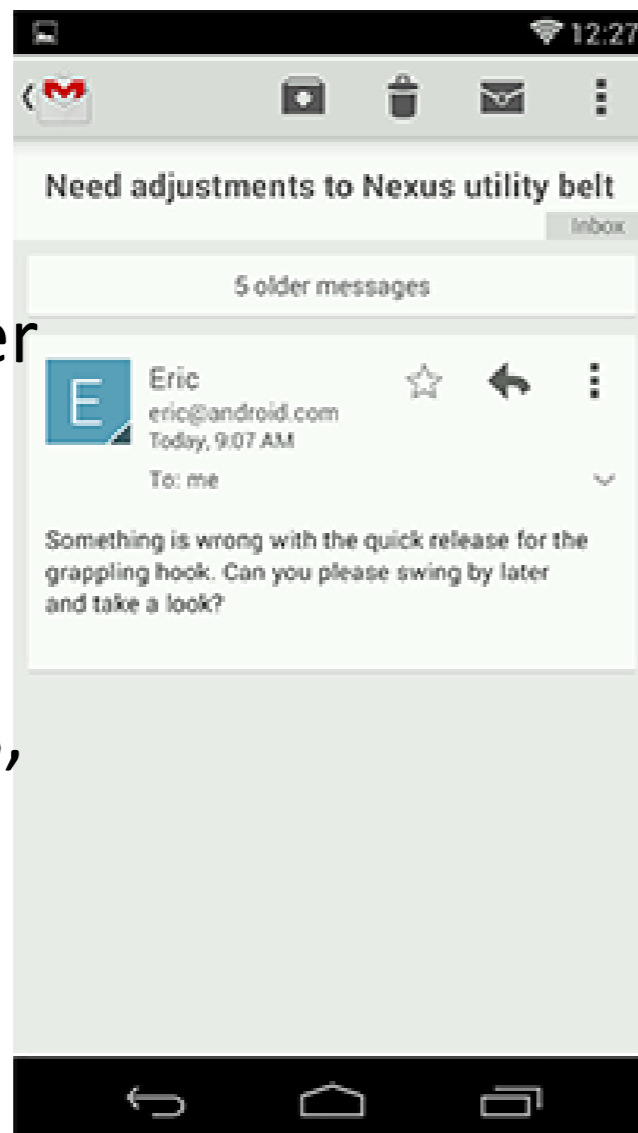
App Structure Checklist

- 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.

Brilliant UX Resource

- https://developer.android.com/training/best-ux.html

Best Practices for Interaction and Engagement

These classes teach you how to engage and retain your users by implementing the best interaction patterns for Android. For instance, to help users quickly discover content in your app, your app should match their expectations for user interaction on Android. And to keep your users coming back, you should take advantage of platform capabilities that reveal and open your content without requiring users to go through the app launcher.

Designing Effective Navigation

How to plan your app's screen hierarchy and forms of navigation so users can effectively and intuitively traverse your app content using various navigation patterns.

Planning Screens and Their Relationships

Planning for Multiple Touchscreen Sizes

Providing Descendant and Lateral Navigation

Providing Ancestral and Temporal Navigation

Putting it All Together: Wireframing the Example App

Implementing Effective Navigation

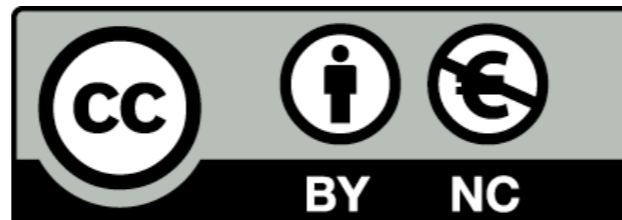How to implement various navigation patterns such as swipe views, a navigation drawer, and up navigation.

Creating Swipe Views with Tabs

Creating a Navigation Drawer

Providing Up Navigation

Providing Proper Back Navigation

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

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit