Android : UI Design

Éric Languénou

2012-2013

UNIVERSITÉ DE NANTES

This presentation has been made thanks to:
`http://developer.android.com/design/get-started/creative-vision.html`

Sommaire

UNIVERSITÉ DE NANTES

Good design

UNIVERSITÉ DE NANTES

Goals:

- Enchant me
  - Good design = good business
- Simplify my life
  - Users should grasp the most important features at first use;
  - consider file management and syncing
  - decompose in simple tasks
  - users like to feel in control
- Make me amazing
  - Allow applications to group into new workflows through multitasking, notifications, and sharing across apps.

Enchant Me

UNIVERSITÉ DE NANTES

- surprising ways
  - A beautiful surface,
  - carefully-placed animations,
  - well-timed sound effect
- metaphors for buttons and menus
  - Allow users to directly touch and manipulate objects
  - it reduces the cognitive effort needed to perform a task
- Let me make mine
  - users love to add personal touches
  - consider fun, optional customizations that don't hinder primary tasks
- Get to know me
  - Learn peoples' preferences over time.
  - place previous choices within easy reach

Simplify My Life

UNIVERSITÉ DE NANTES

- Keep it brief
  - short phrases with simple words
- Pictures are faster than words
  - use pictures to explain ideas
  - pictures get attention and can be much more efficient than words.
- Decide for me but let me have the final say
  - use best guess and act (rather than asking first)
  - avoid many choices and decisions (make people unhappy)
  - allow for 'undo'.

Simplify My Life(2)

UNIVERSITÉ DE NANTES

- Only show what I need when I need it
  - don't show too much at once
  - Break tasks and information into small, digestible chunks
  - Hide options that aren't essential (teach people as they go)
- I should always know where I am
  - Make places in your app look distinct and use transitions to show relationships among screens
  - Provide feedback on tasks in progress.
- Never lose my stuff
  - let users access the stuff they've created from anywhere
  - Remember settings, personal touches, and creations across phones, tablets, and computers. (easier upgrading)

Simplify My Life(3)

UNIVERSITÉ DE NANTES

- If it looks the same, it should act the same
  - functional differences are made visually distinct;
  - Avoid modes.
- Only interrupt me if it's important

Make Me Amazing

UNIVERSITÉ DE NANTES

- Give me tricks that work everywhere
  - let users figure things out for themselves
  - use consistency through visual patterns and muscle memory from other Android apps
- It's not my fault
  - Be gentle in your prompt
  - users want to feel smart
  - give clear recovery instructions (not the technical details)
  - fix behind the scene if possible

Make Me Amazing(2)

UNIVERSITÉ DE NANTES

- Sprinkle encouragement
  - Break complex tasks into smaller steps (easily accomplished)
  - feedback on actions (even subtle)
- Do the heavy lifting for me
  - Make novices feel like experts
  - provide shortcuts (even for complicated actions)
- Make important things fast
  - hierarchy in actions
  - make important actions easy to find and fast to use (shutter, play/pause)

UI Overview

UNIVERSITÉ DE NANTES

Recent UI shifts in android :

Home screen

All Apps Screen

Recents Screen

Home Screen

UNIVERSITÉ DE NANTES

Home screen

- customizable space that houses :
  - app shortcuts,
  - folders
  - and widgets.
- Navigate between different home screen panels
- The Favorites Tray at the bottom
  - shows most important shortcuts and folders
  - regardless of which panel is currently showing.
- The button at the center of the Favorites Tray shows the *All apps screen*.

All Apps Screen

UNIVERSITÉ DE NANTES

Home screen

- shows the entire set of apps and widgets
- Users can drag an app or widget icon from the All Apps screen toward any Home screen.

Recents Screen

Home screen

- easy way to switch between recently used applications
- provides a clear navigation path
- accessible by the Recents button at the right side of the navigation bar displays

System Bars

UNIVERSITÉ DE NANTES

System Bars(2)

UNIVERSITÉ DE NANTES

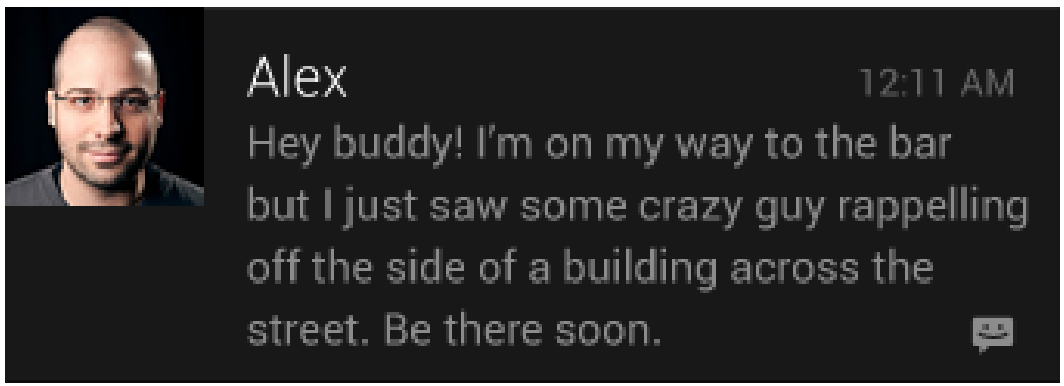
- screen areas dedicated to the display of
  - notifications,
  - communication of device status,
  - device navigation
- displayed concurrently with your app
- can be temporarily hiden

System Bars (3)

UNIVERSITÉ DE NANTES

1. Status Bar
   - Displays pending notifications on the left and
   - status, such as time, battery level, or signal strength, on the right.
   - Swipe down from the status bar to show notification details.
2. Navigation Bar
   - since Android 4.0,
   - present only on devices that don't have the traditional hardware keys
   - provides for device navigation controls Back, Home, and Recents
3. Combined Bar
   - the status and navigation bars are combined into a single bar at the bottom of the screen

Notifications



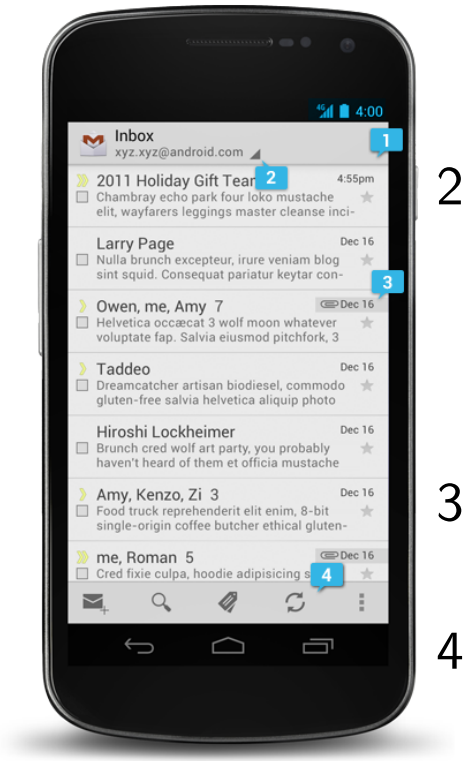
Notifications

Notifications

UNIVERSITÉ DE NANTES

- brief messages that users can access at any time from the status bar.
- provide updates, reminders, or information that's important, but not critical enough to warrant interrupting the user
- access by swiping down on the status bar
- Touching a notification opens the associated app.
- Notifications can be expanded
- When collapsed, notifications have a one-line title and a one-line message.
- Swiping a notification right or left removes it from the notification drawer.

Typical App UI



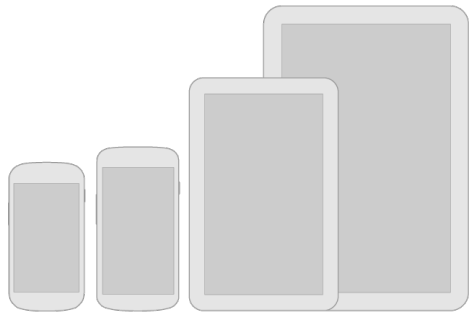
1. Main Action Bar
   - is the command and control center for your app
   - contains elements for navigating hierarchy and views
   - surfaces the most important actions.
2. View Control
   - switch between the different views of your app
   - Views consist of
     - different arrangements of your data
     - different functional aspects of your app
3. Content Area
   - Displays the content of your app
4. Split Action Bar
   - additional bars located below the main action bar or at the bottom of the screen
   - important actions that won't fit in the main bar to the bottom

Sommaire

UNIVERSITÉ DE NANTES

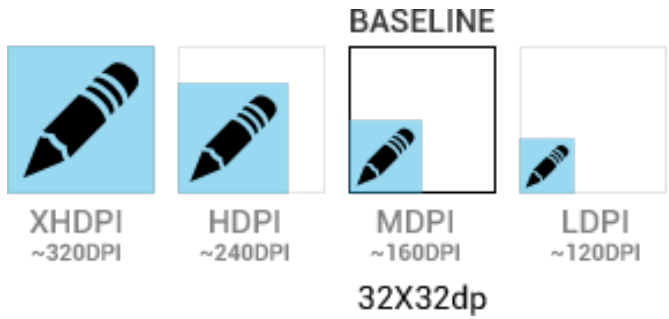
Devices and Displays

UNIVERSITÉ DE NANTES



- Be flexible
  - accommodate various heights and widths.
- Optimize layout
  - On larger devices, take advantage of extra screen
  - Create compound views that combine multiple views
    - to reveal more content
    - and ease navigation.
- Assets for all
  - Provide resources for different screen densities (DPI)

Devices and Displays

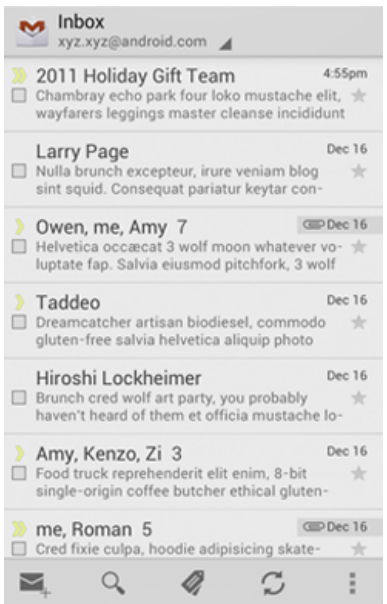


- Strategies while designing for multiple screens?
  - work in the base standard (normal size and MDPI) and scale it up or down for the other buckets.
  - start with the device with the largest screen size, and then scale down and figure out the UI compromises you'll need to make on smaller screens.

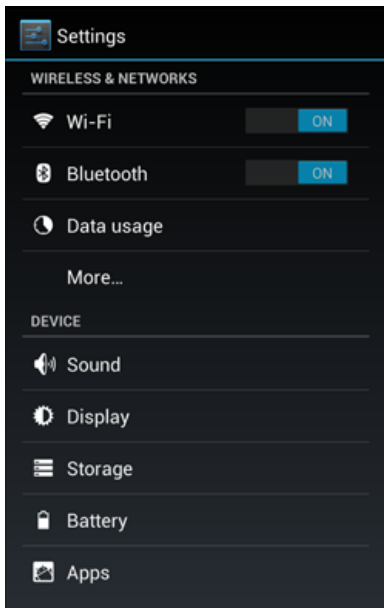
Themes

UNIVERSITÉ DE NANTES

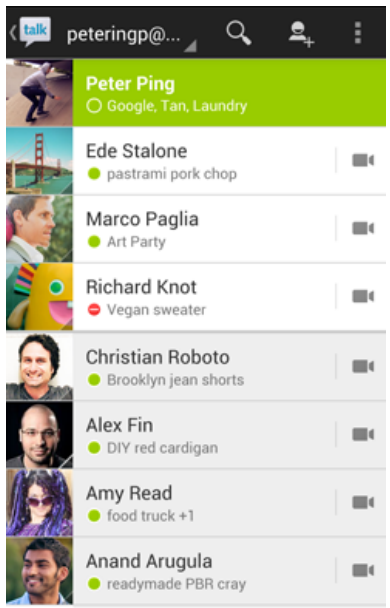
- mechanism for applying a consistent style to an app or activity
- such as color, height, padding and font size
- Android provides three system themes that you can choose from when building apps (ICS)
- use android system style as a starting point



Gmail in Holo light



Settings in Holo Dark



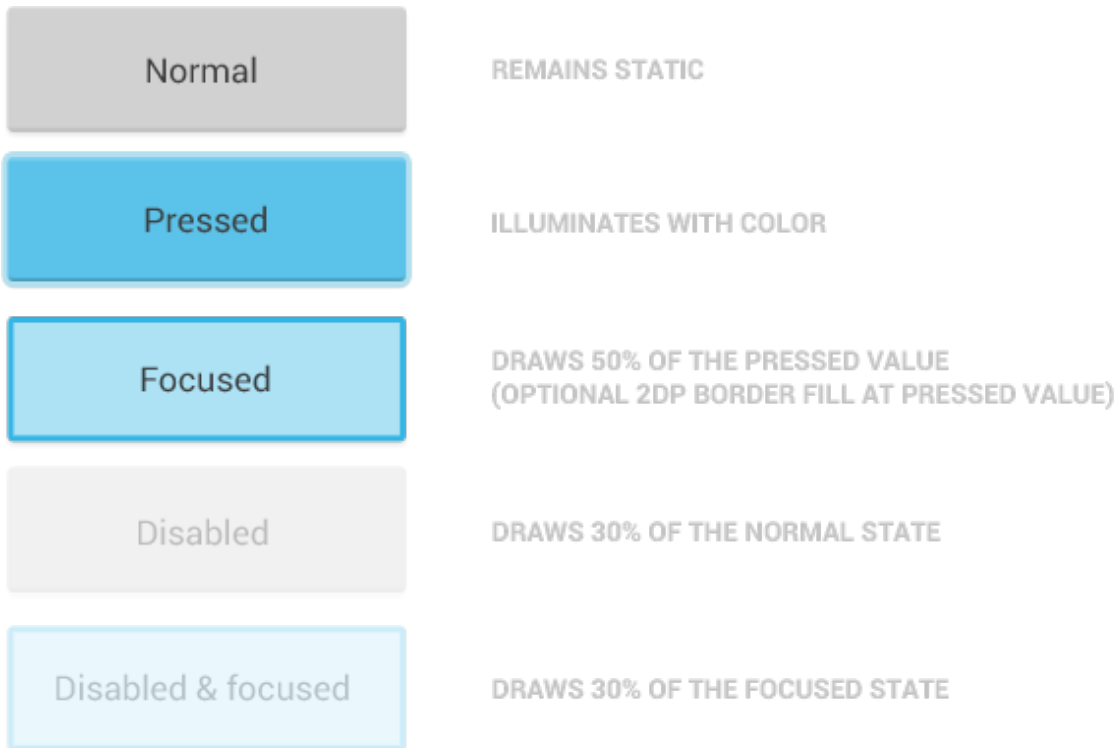
Talk in Holo Light with dark action Bar

Touch Feedback

UNIVERSITÉ DE NANTES

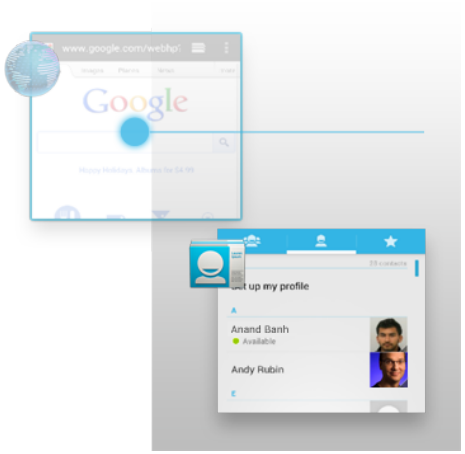
- Use color and illumination
  - to respond to touches,
  - reinforce the resulting behaviors of gestures, and
  - indicate what actions are enabled and disabled.
- Always provide a visual response.
- To let the user know which object was touched and that your app is "listening".

States



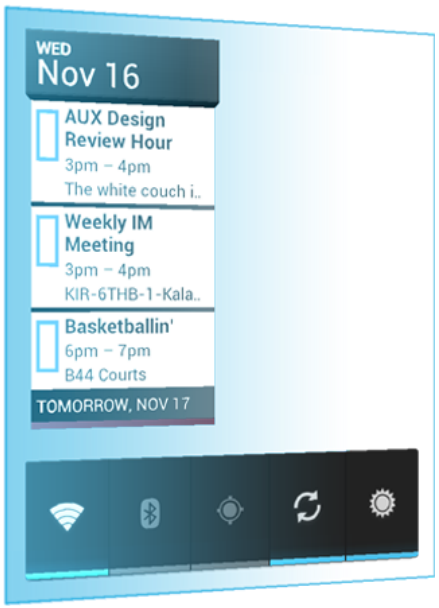
Most of Android's UI elements have touch-feedback built in, including states that indicate whether touching the element will have any effect.

Communication



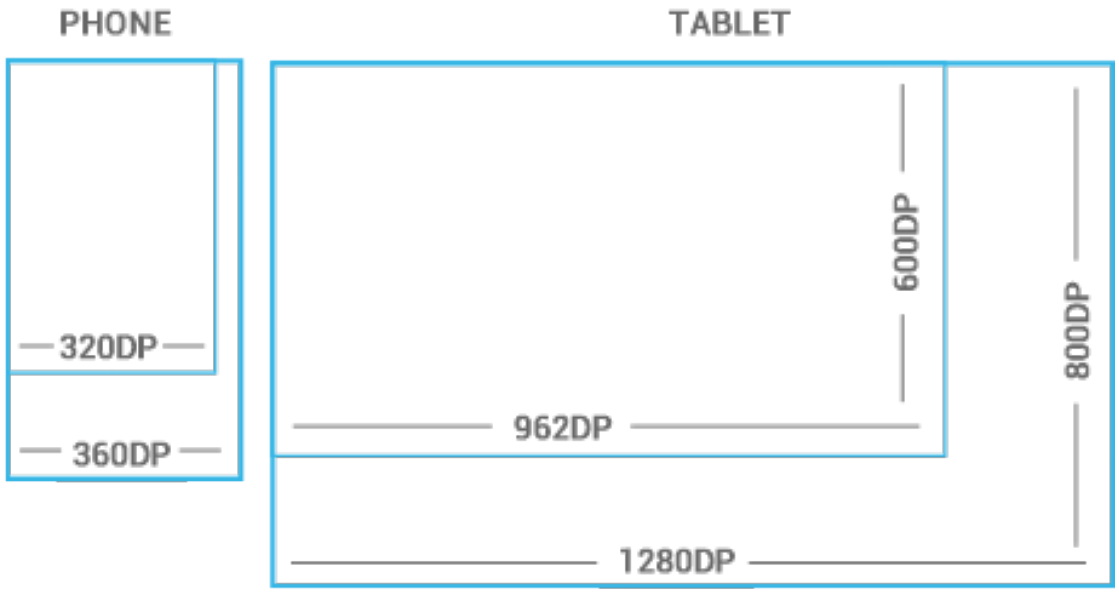
- help users understand what the outcome of the operation will be;
- Example, in Recents, when you start swiping a thumbnail left or right, it starts to dim. This helps the user understand that swiping will cause the item to be removed.

Boundaries



- When users try to scroll past the upper or lower limit of a scrollable area, communicate the boundary with a visual cue.
- For example, if a user attempts to scroll past the first home screen panel, the screen content tilts to the right to indicate that further navigation in this direction is not possible;
- already implemented in most of Android Widget;

Metrics and Grids



- Devices vary not only in physical size, but also in screen density (DPI)
- Design for multiple screens,
  - think of each device as falling into a particular size bucket and density bucket

Themes

UNIVERSITÉ DE NANTES

- The **size buckets** are
  - handset (smaller than 600dp)
  - tablet (larger than or equal 600dp)
- The **density buckets** are :
  - LDPI
  - MDPI
  - HDPI
  - XHDPI

48dp Rhythm

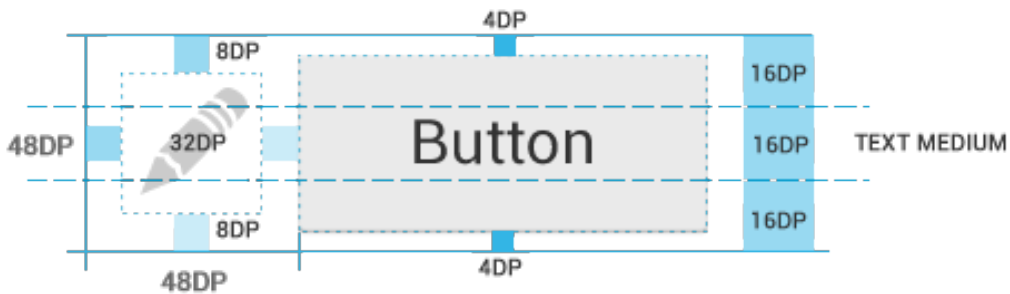
UNIVERSITÉ DE NANTES

Touchable UI components are generally laid out along 48dp units.

Why 48dp ?

UNIVERSITÉ DE NANTES

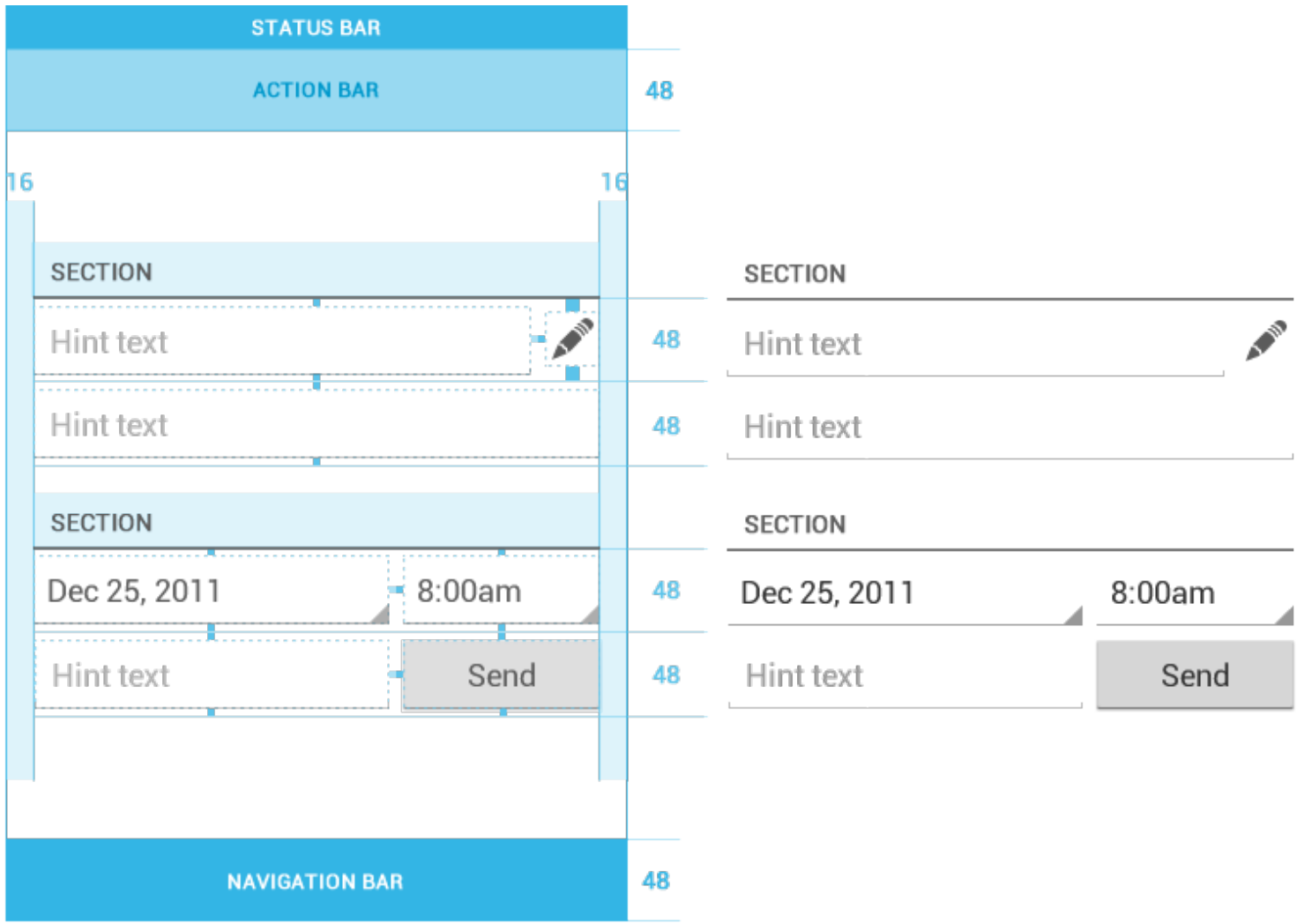
- 48dp translate to a physical size of about 9mm
- recommended target sizes (7-10 mm) for touchscreen objects
- Users can target them with their fingers.
- at least 48dp high and wide, can guarantee that:
  - target never be smaller than the minimum recommended (regardless of the screen).
  - good compromise between overall information density, and targetability of UI elements.



Spacing

UNIVERSITÉ DE NANTES

Spacing between each UI element is 8dp.

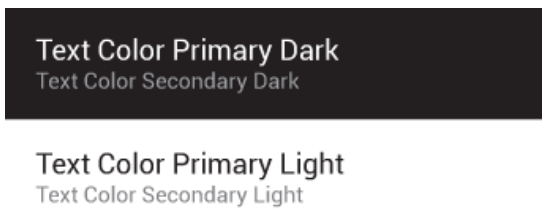


Typography

- Android design language relies on traditional typographic tools such as scale, space, rhythm, and alignment with an underlying grid;
- new type family named Roboto, created specifically for the requirements of UI and high-resolution screens.
- TextView framework supports regular, bold, italic, and bold italic weights by default.

Type colors and Scale

UNIVERSITÉ DE NANTES



- Following default color styles: *textColorPrimary* and *textColorSecondary*.
- For light themes use *textColorPrimaryInverse* and *textColorSecondaryInverse*.

Type colors and Scale(2)

UNIVERSITÉ DE NANTES

- Too many different sizes in the same UI can be messy.
- Users can select a system-wide scaling factor for text in the Settings app
- Type should be specified in scale-independent pixels (sp) wherever possible.
- The Android framework uses the following limited set of type sizes:

Color

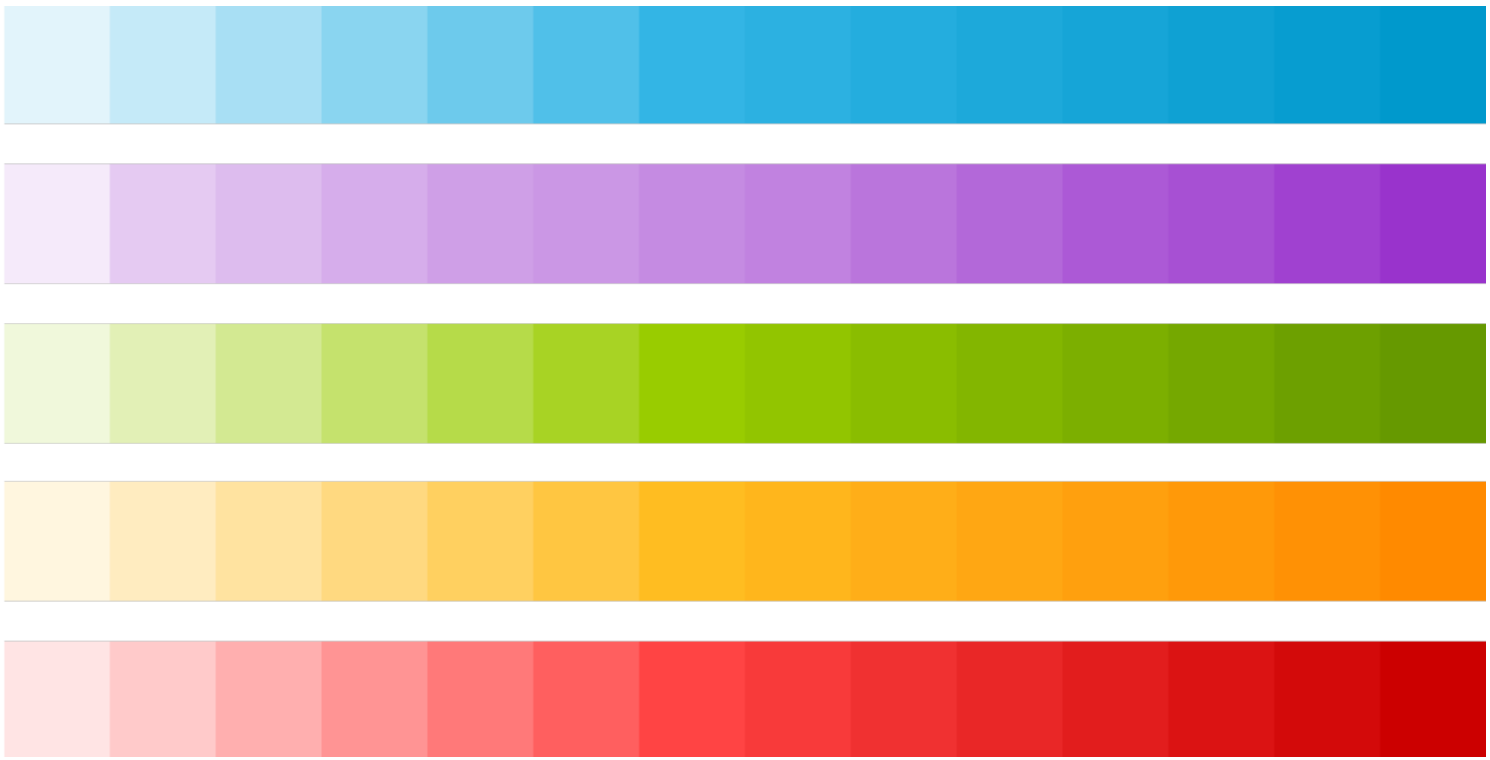
UNIVERSITÉ DE NANTES

- Use color primarily for emphasis.
- Choose colors that fit with your brand and provide good contrast between visual components.
- Note that red and green may be indistinguishable to color-blind users.

Palette

UNIVERSITÉ DE NANTES

- Blue is the standard accent color in Android's color palette.
- Each color has a corresponding darker shade that can be used as a complement when needed.



Iconography

An icon is a graphic that

- takes up a small portion of screen real estate and
- provides a quick, intuitive representation of an action, a status, or an app.

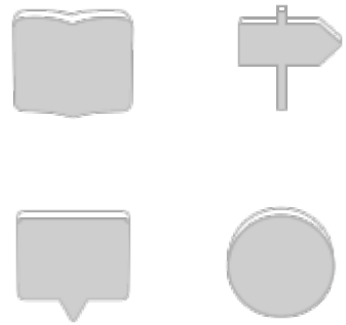
Launcher

Sizes & scale

- Launcher icons on a mobile device must be 48x48 dp.
- Launcher icons for display on Google Play (Google store) must be 512x512 pixels.

Proportions

- Full asset, 48x48 dp



Style

- Use a distinct silhouette.
- Three-dimensional, front view, with a slight perspective as if viewed from above,
- so that users perceive some depth.

Action Bar

UNIVERSITÉ DE NANTES

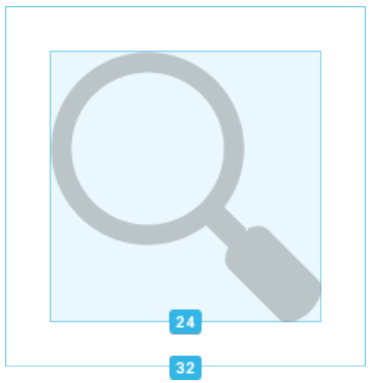





Action Bar

UNIVERSITÉ DE NANTES

- Action bar icons are graphic buttons (most important actions in your App)
- simple metaphor for a simple concept
- downloadable pre-defined set of icons
- size is 32dpx32dp
- Full asset, 32x32 dp, Optical square, 24x24 dp
- Style : Pictographic, flat, not too detailed, with smooth curves or sharp shapes.



Contextual Icon

UNIVERSITÉ DE NANTES

- within the body of your App
- Small icons should be 16x16 dp.
- Full asset, 16x16 dp, Optical square, 12x12 dp
- Style : Neutral, flat, and simple. Filled shapes are easier to see than thin strokes.

Notification Icon

UNIVERSITÉ DE NANTES

- 24x24 dp.
- Full asset, 24x24 dp, Optical square, 22x22 dp
- Style : Keep the style flat and simple, using the same single, visual metaphor as your launcher icon.
- Colors : Notification icons must be entirely white (and the system may scale down and/or darken the icons).

Writing Style

UNIVERSITÉ DE NANTES

Choosing words:

1. Keep it brief: Be concise, simple and precise. Start with a 30 character limit..
2. Keep it simple : Pretend you're speaking to someone who's smart and competent, but doesn't know technical jargon and may not speak English very well. Use short words, active verbs, and common nouns.
3. Be friendly : Use contractions. Talk directly to the reader using second person ("you")

Writing Style

UNIVERSITÉ DE NANTES

4. Put the most important thing first : The first two words (around 11 characters, including spaces) should include at least a taste of the most important information in the string. If they don't, start over.
5. Describe only what's necessary, and no more : Don't try to explain subtle differences. They will be lost on most users.
6. Avoid repetition : If a significant term gets repeated within a screen or block of text, find a way to use it just once.

See examples on `http://developer.android.com/design/style/writing.html`

Sommaire

UNIVERSITÉ DE NANTES

Pattern

UNIVERSITÉ DE NANTES

New

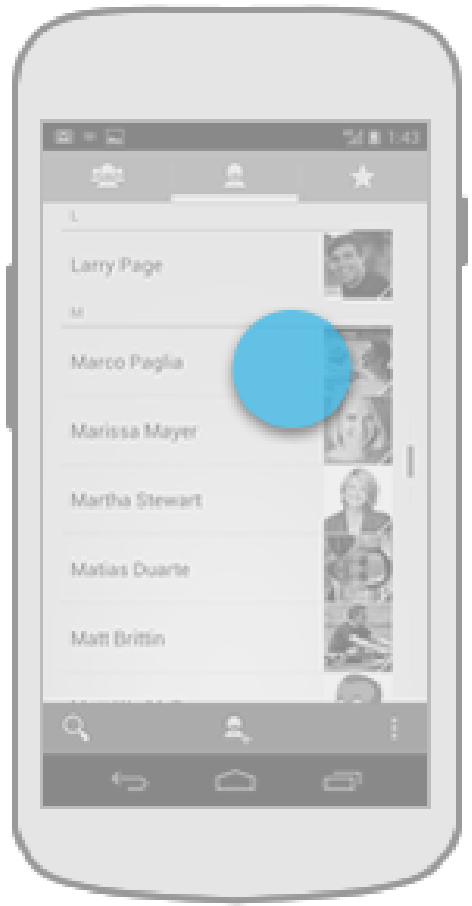
UNIVERSITÉ DE NANTES

- Jelly Bean - Android 4.1
  - Notifications
  - Resizable Application Widgets
  - Accessibility
- Ice Cream Sandwich - Android 4.0
  - Navigation bar
  - Action bar
  - Multi-pane layouts
  - Selection

Gesture

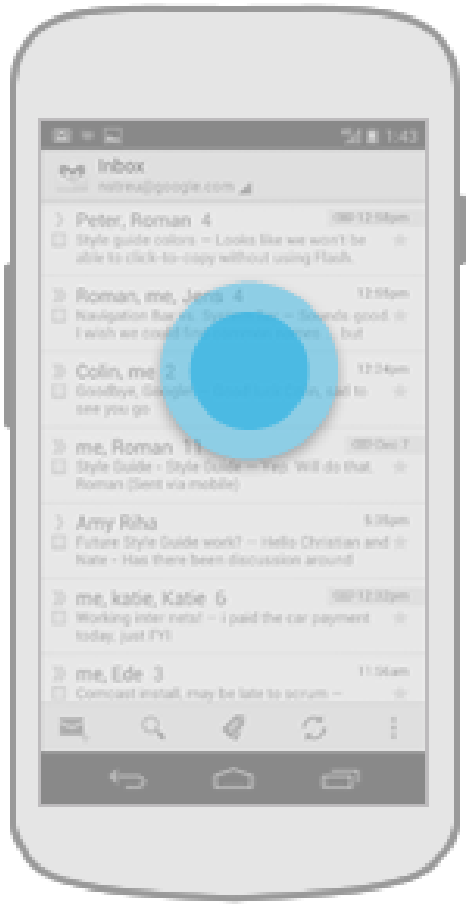
UNIVERSITÉ DE NANTES

Gestures allow users to interact with your app by manipulating the screen objects you provide.



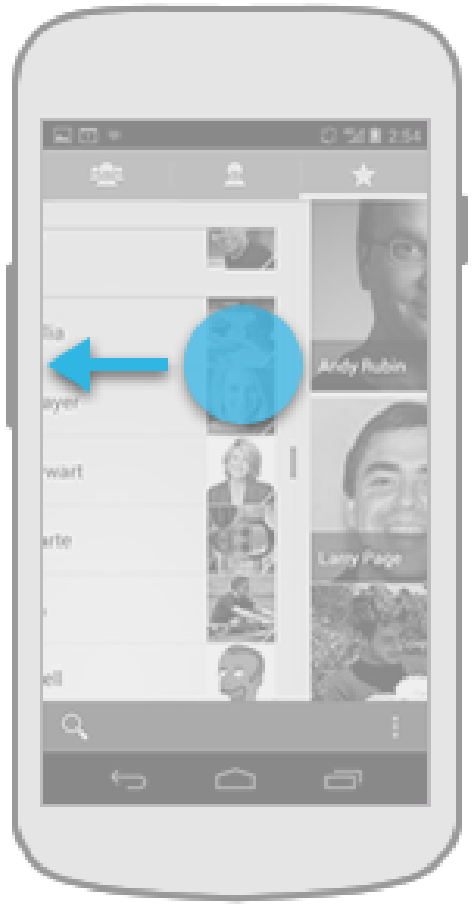
- Touch
  - What : Triggers the default functionality for a given item.
  - Action : Press, liftLauncher icons on a mobile device must be 48x48 dp.

Gesture



- Long press
  - What : 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

Gesture



- Swipe
  - What : Scrolls overflowing content, or navigates between views in the same hierarchy.
  - Action : Press, move, lift

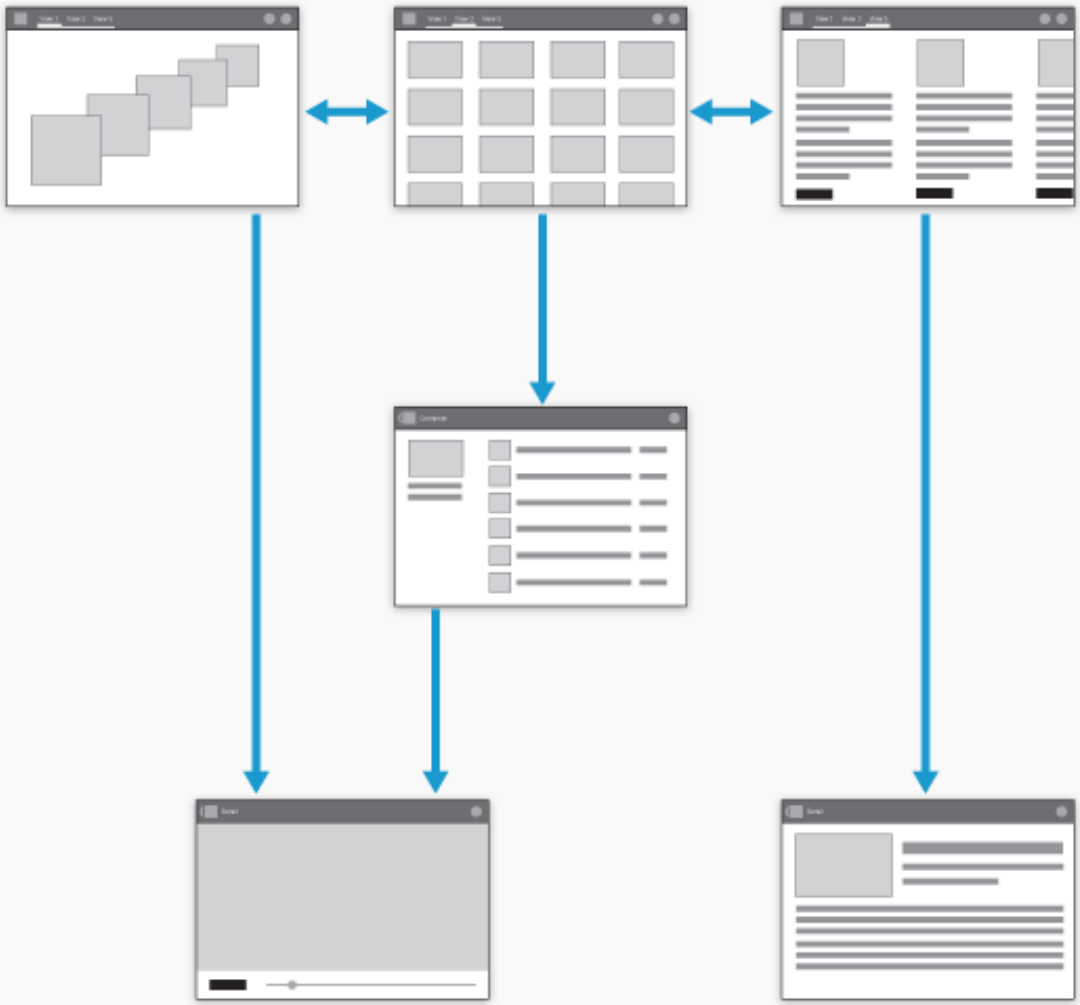
Gesture

UNIVERSITÉ DE NANTES

More gesture on `http://developer.android.com/design/patterns/gestures.html`

General Structure

UNIVERSITÉ DE NANTES



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.

Top Level

UNIVERSITÉ DE NANTES

- start screen requires special attention
- equally rewarding experience for new and frequent visitors alike
- "What are my typical users most likely going to want to do in my app?" gives the structure of your start screen.
- put content forward
  - making content the centerpiece of your start screen
  - avoid navigation-only screen for start screen
  - choose an appealing layout appropriate for the data type and screen size.

Top Level(2)

UNIVERSITÉ DE NANTES

- Set up action bars for navigation and actions
- At the top level, special considerations apply to the action bar:
  - display your app's icon or title.
  - if multiple views (i.e multiple accounts) provide navigation through the action bar
  - if content creation, consider making the content accessible right from the top level.
  - searchable content : include the Search action in the action bar

Top Level(3)

UNIVERSITÉ DE NANTES

- Create an identity for your app
  - Creating an identity for your app goes beyond the action bar.
  - Your app communicates its identity through its data, the way that data is arranged, and how people interact with it
  - try to create unique layouts that showcase your data and go beyond the monotony of simple list views.

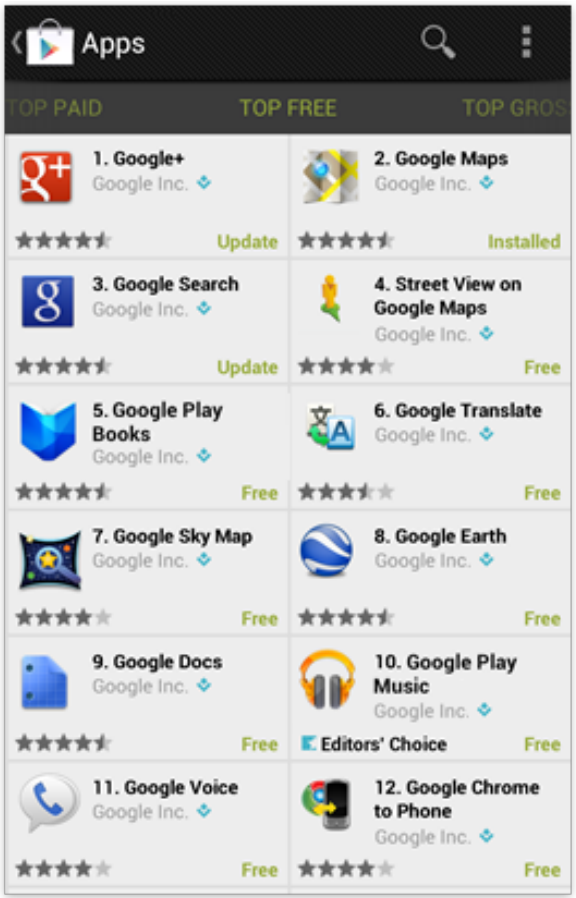
Categories

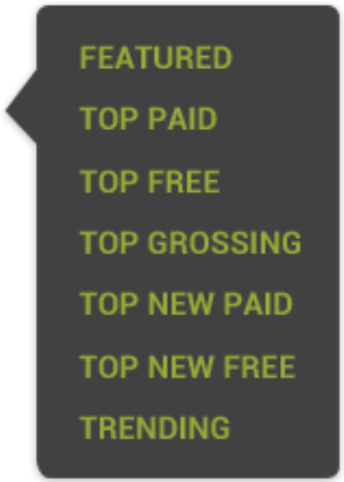
UNIVERSITÉ DE NANTES

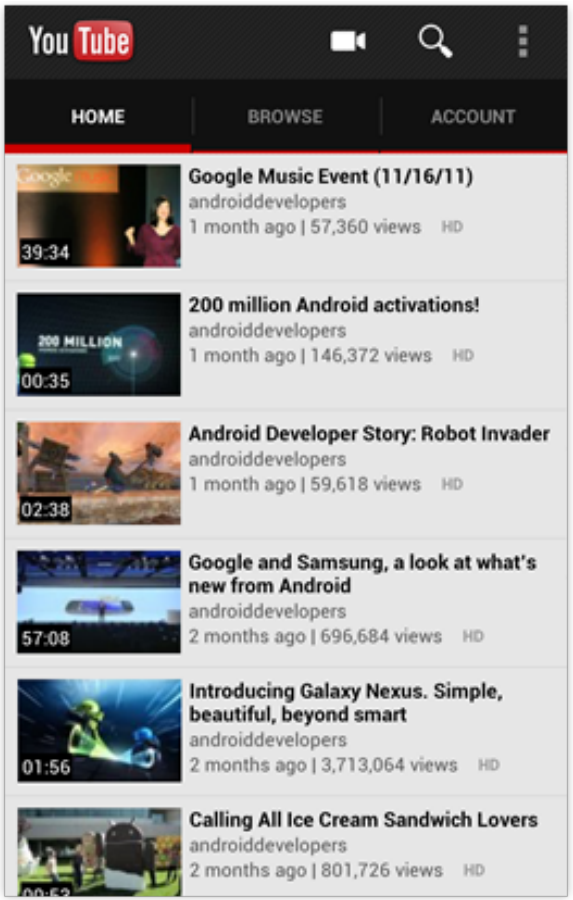
- Generally, the purpose of a deep, data-driven app is to navigate through organizational categories to the detail level, where data can be viewed and managed.
- It's important to minimize perceived navigation effort.

Use of Tabs

UNIVERSITÉ DE NANTES







Use of Tabs

UNIVERSITÉ DE NANTES

Use tabs to combine category selection and data display

- if categories are familiar or the number of categories is small
- a level of hierarchy is removed and data remains at the center of the user's attention
- this is not perceived as a navigation step by the user

Fixed Tab or Scrollable Tab

UNIVERSITÉ DE NANTES

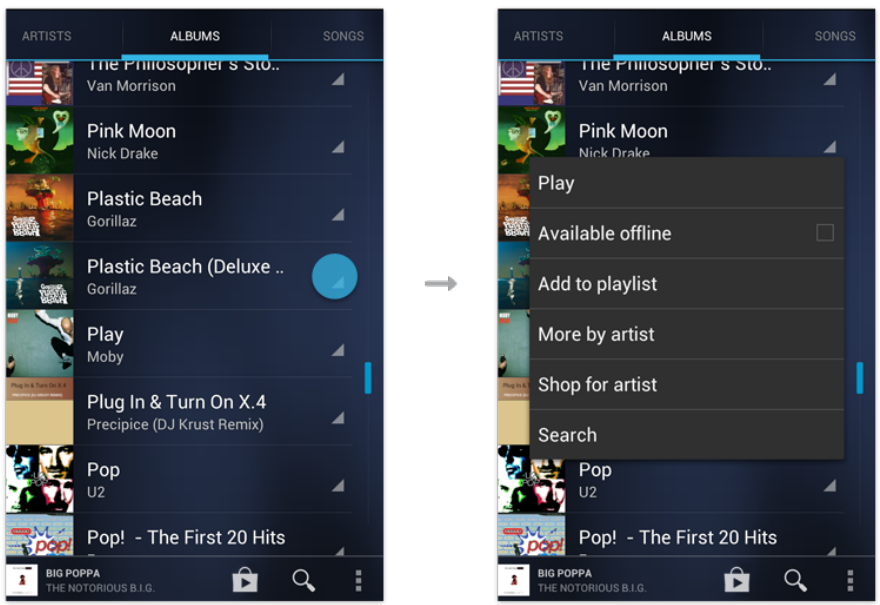
Use tabs to combine category selection and data display

- for categories familiar, predictable or closely related :
  - use scrolling tabs (where not all items are in view simultaneously).
  - Keep the number of scrolling tabs at a manageable level to minimize navigational effort.
  - Rule of thumb: no more than 5-7 tabs.
- otherwise
  - favor fixed tabs
  - all categories are in view at the same time.

Allow cutting through hierarchies

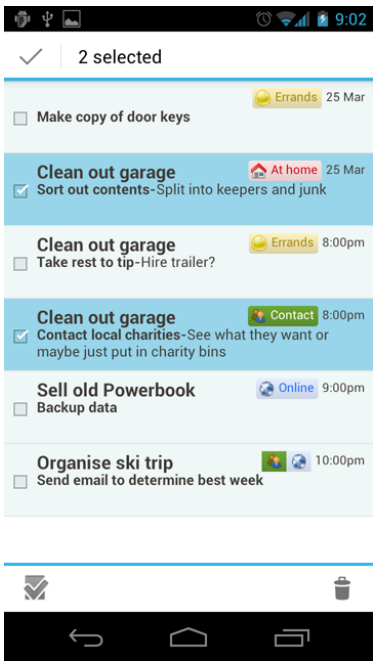
UNIVERSITÉ DE NANTES

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



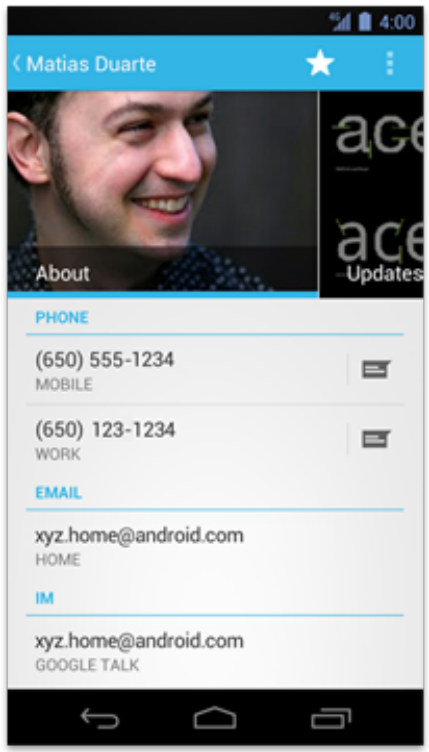
Acting upon multiple data items

UNIVERSITÉ DE NANTES



- 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.
- 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

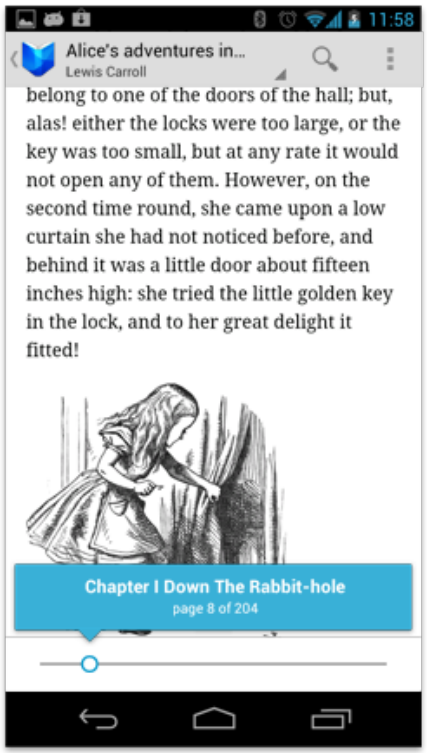
Details View

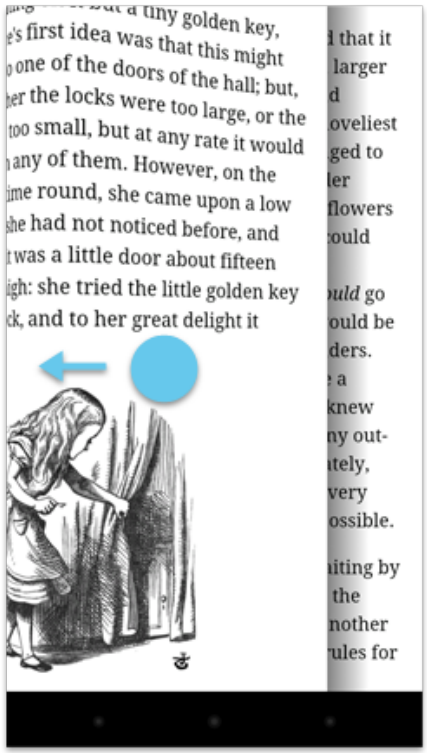


The purpose of the People app's detail view is to surface communication options. The list view allows for efficient scanning and quick access. Split items are used to combine calling and messaging into one compact line item.

- 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,

Lights-out mode

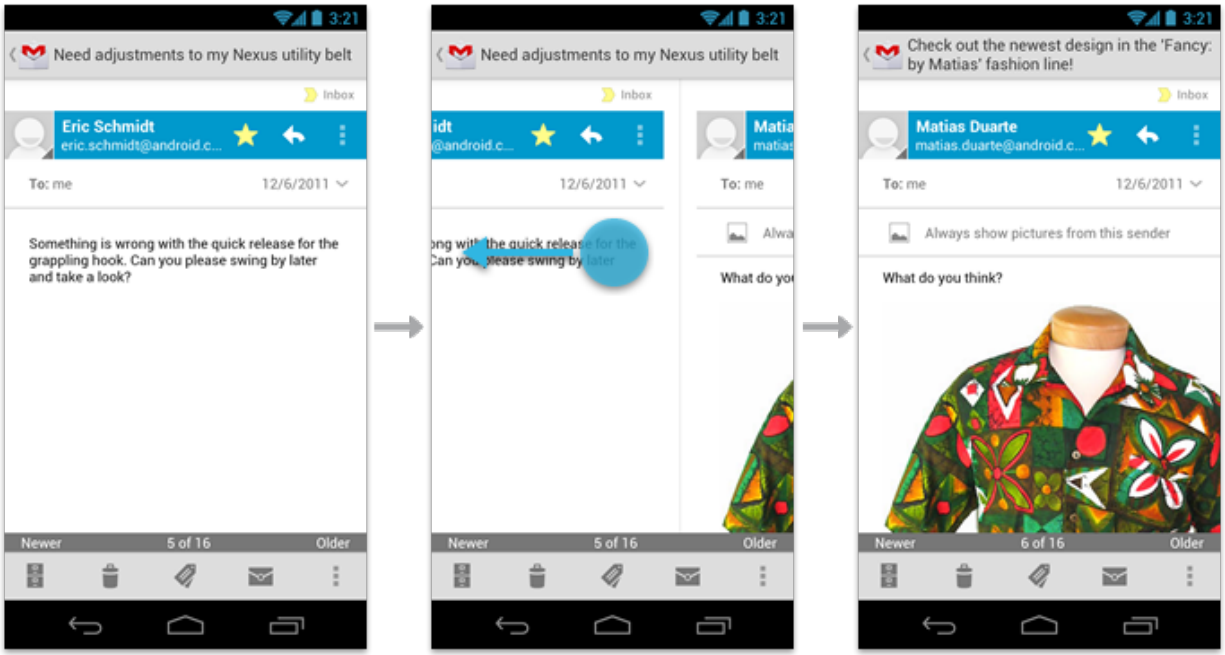




- some apps need a full screen without distractions
- non intrusive menu/buttons disappears after a while
- a touch on the screen reset them visible

Make navigation between detail views efficient

UNIVERSITÉ DE NANTES



- look at multiple items in sequence,
- allow the user to navigate between items from within the detail view.
- Use swipe views or other techniques, such as thumbnail view controls.

App Structure CheckList

UNIVERSITÉ DE NANTES

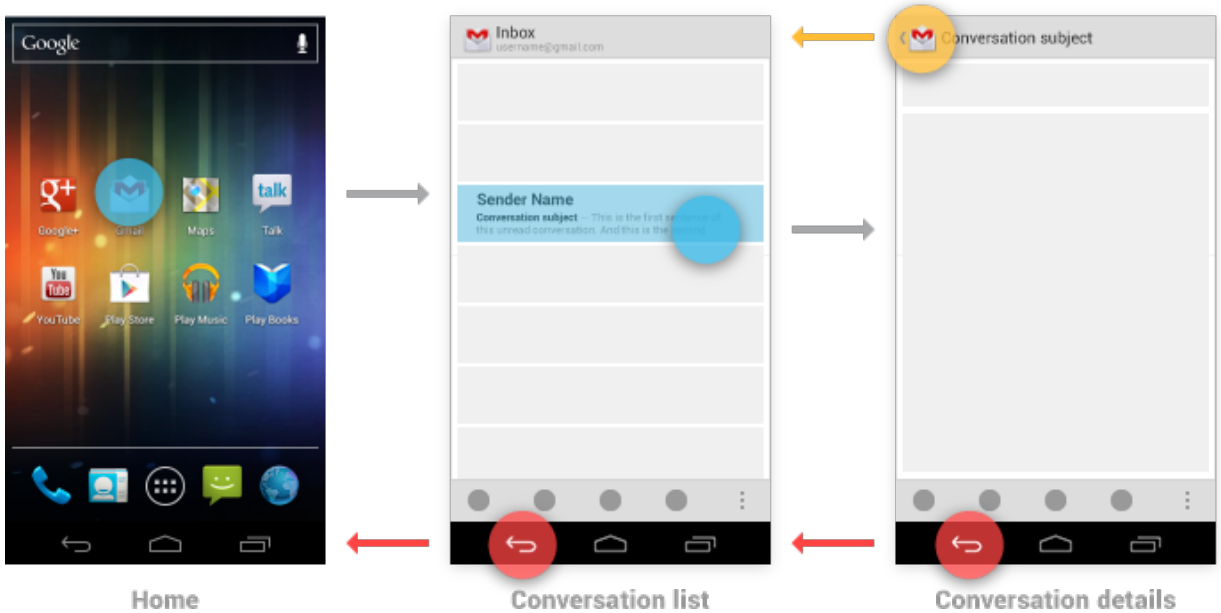
- Find ways to display useful content on your start screen.
- Use action bars to provide consistent navigation.
- Keep your hierarchies shallow by using horizontal navigation and shortcuts.
- Use multi-select to allow the user to act on collections of data.
- Allow for quick navigation between detail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

Navigation with Back and Up

UNIVERSITÉ DE NANTES

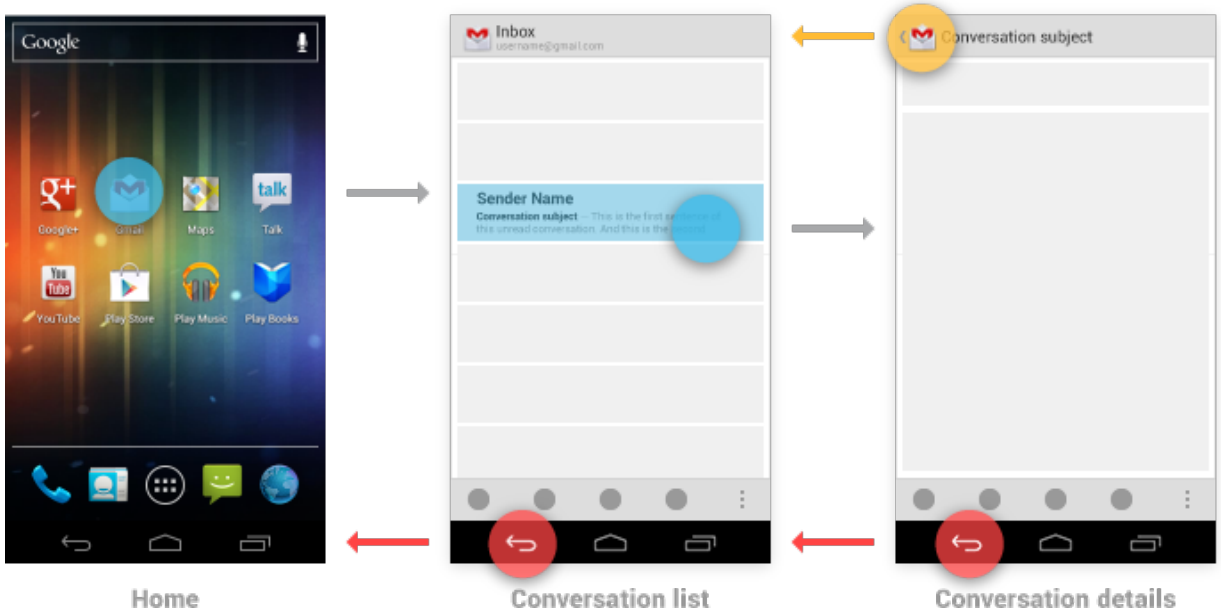
- Consistent navigation is an essential component of the overall user experience
- Inconsistent navigation frustrates the user
- Android 3.0 introduced significant changes to the global navigation behavior
- Android 2.3 and earlier relied upon the system Back button for supporting navigation within an app.
- With action bars in Android 3.0, a second navigation mechanism appeared: the Up button, consisting of the app icon and a left-point caret.

Up vs. Back



- Up button is used to navigate within an app based on the hierarchical relationships between screens
- Start screen should not present an Up button.
- Back button is used to navigate, in reverse chronological order, through the history of screens

Up vs. Back (2)



- based on the temporal relationships between screens, rather than the app's hierarchy
- the Back button can return the user to the Home screen, or even to a different app
- The Back button also supports
  - Dismisses floating windows (dialogs, popups)
  - Dismisses contextual action bars, and
  - removes the highlight from the selected items
  - Hides the onscreen keyboard (IME)

Changing view within a screen

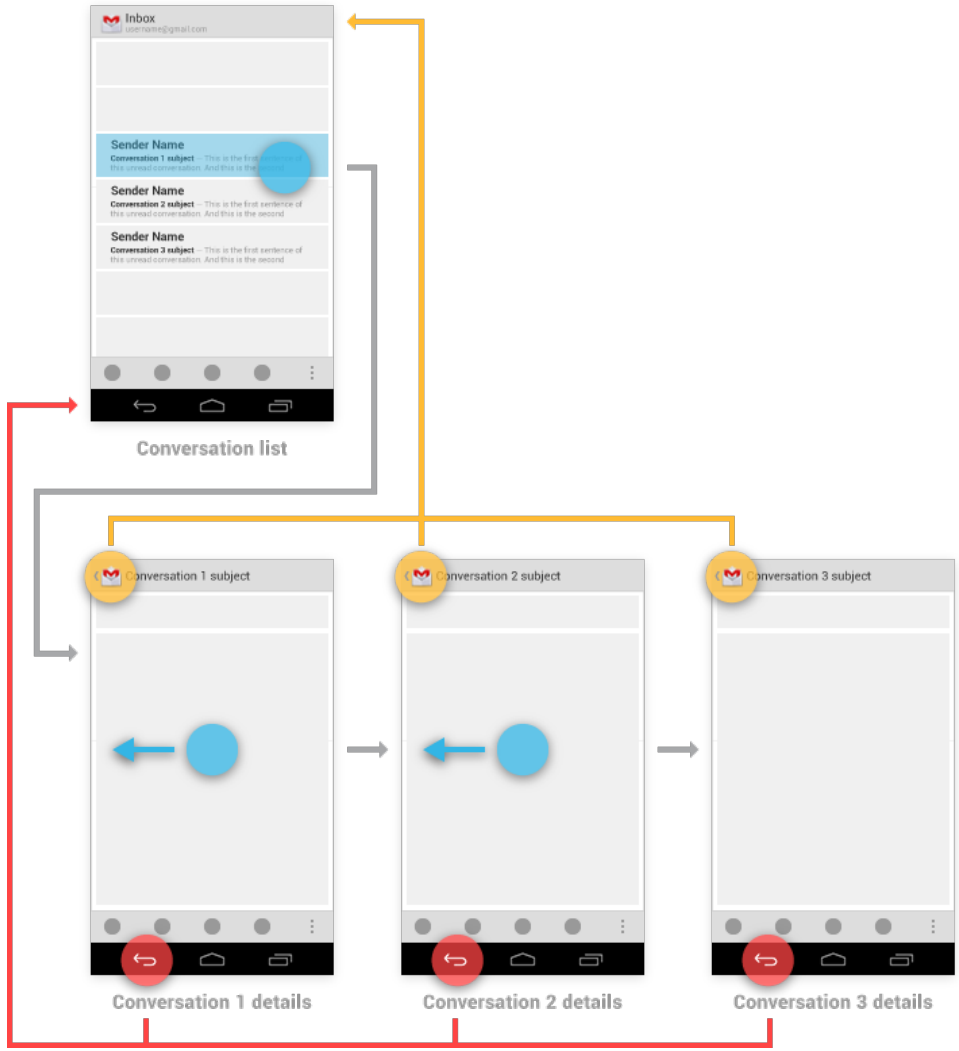
UNIVERSITÉ DE NANTES

- Changing view options for a screen does not change the behavior of Up or Back:
- the screen is still in the same place within the app's hierarchy,
- no new navigation history is created.
- Examples of such view changes are:
  - Switching views using tabs and/or left-and-right swipes
  - Switching views using a dropdown (aka collapsed tabs)
  - Filtering a list
  - Sorting a list
  - Changing display characteristics (such as zooming)

Navigating between sibling screens

UNIVERSITÉ DE NANTES

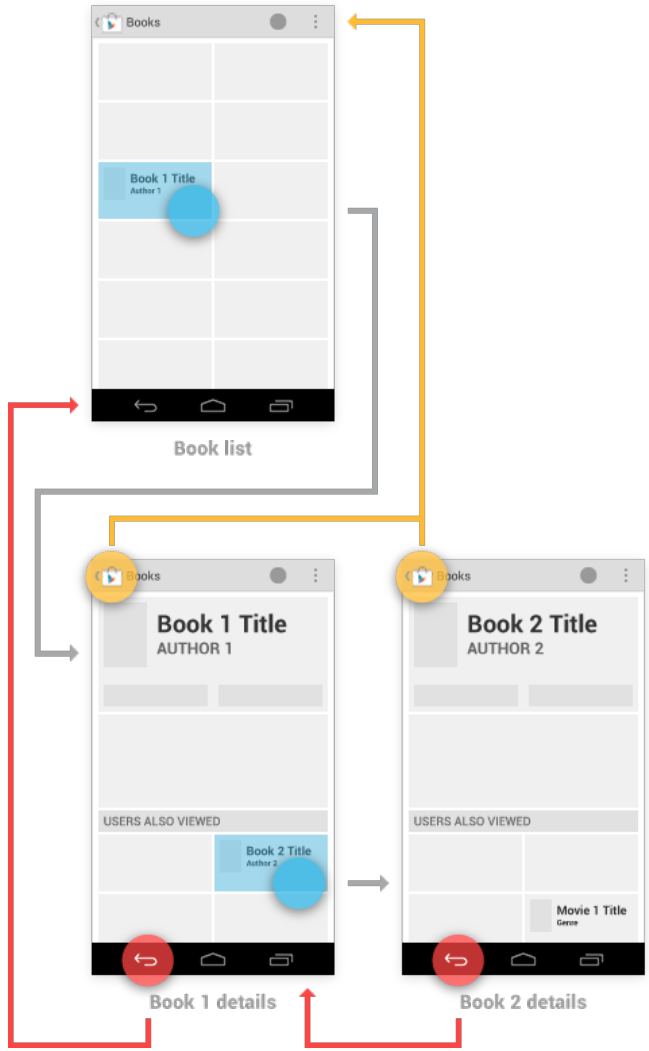
List View



Navigating between sibling screens(2)

UNIVERSITÉ DE NANTES

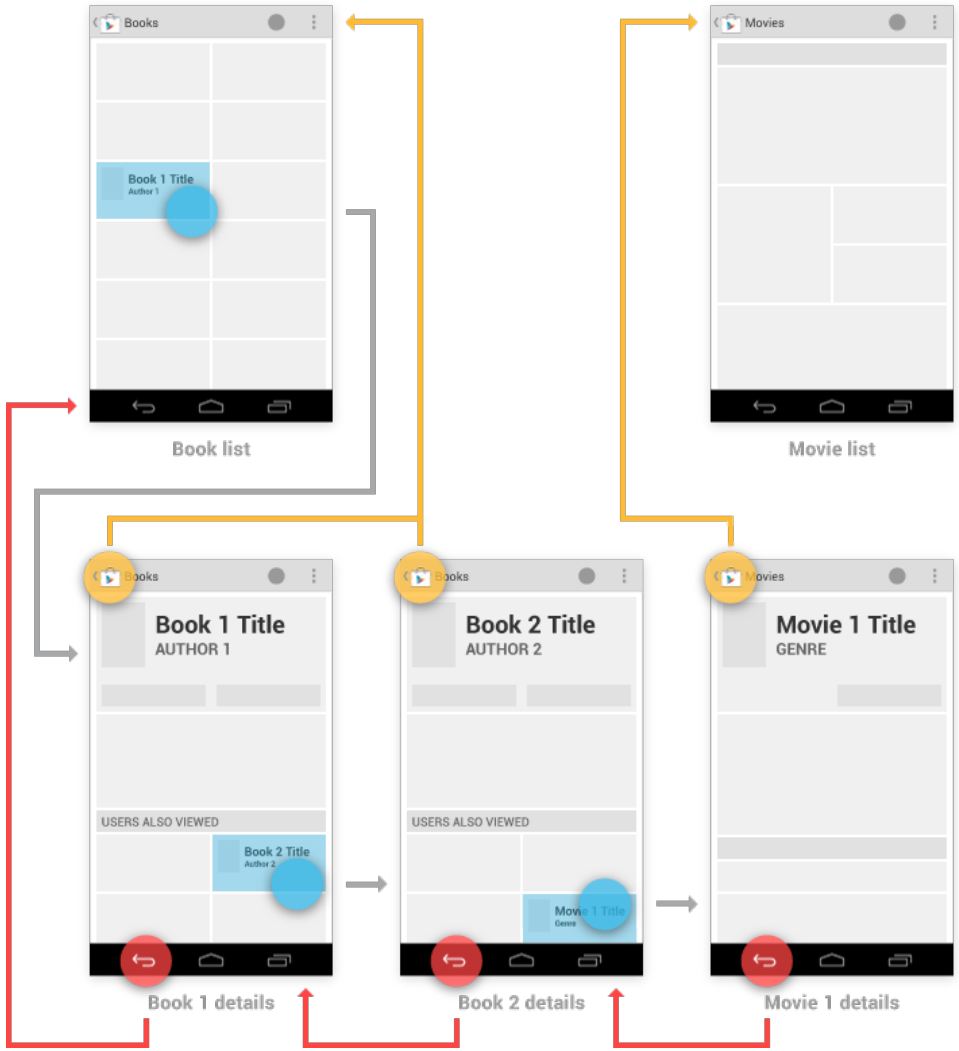
Grid View



Navigating between sibling screens(3)

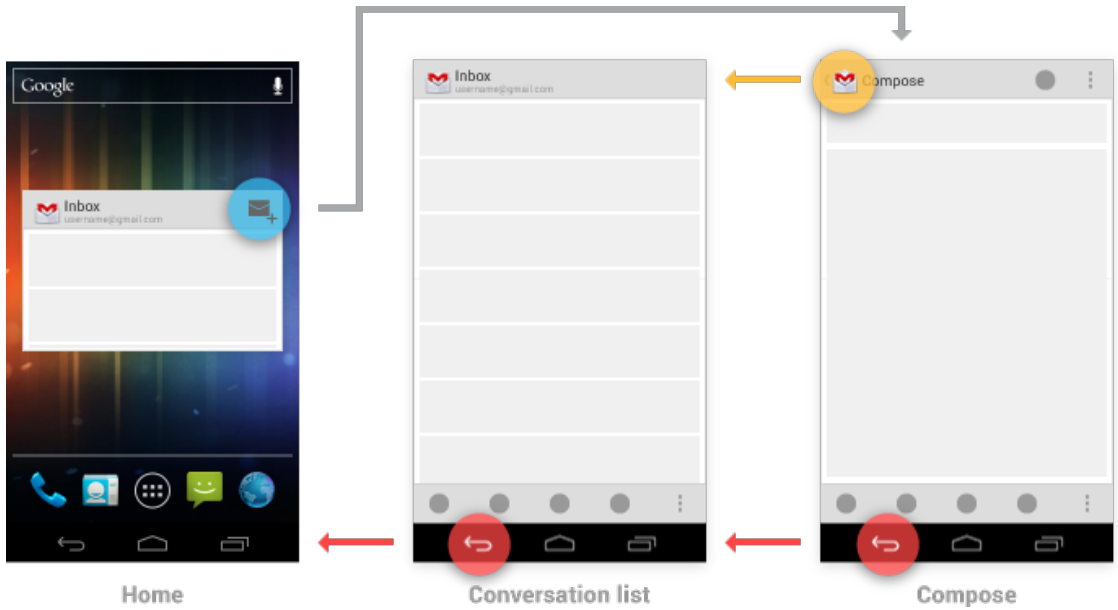
UNIVERSITÉ DE NANTES

Even smarter



Navigation : Widgets and Notifications

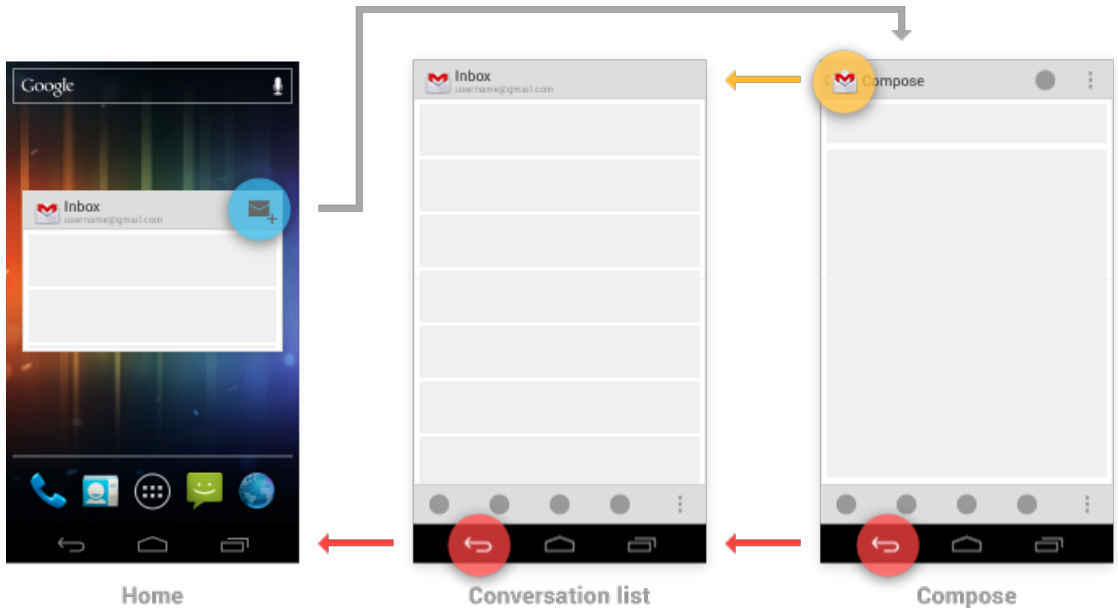
UNIVERSITÉ DE NANTES



- You can use Home screen widgets or notifications to help your users navigate directly to screens deep within your app's hierarchy
- widget and notification can bypass the start screen

Navigation : Widgets and Notifications (2)

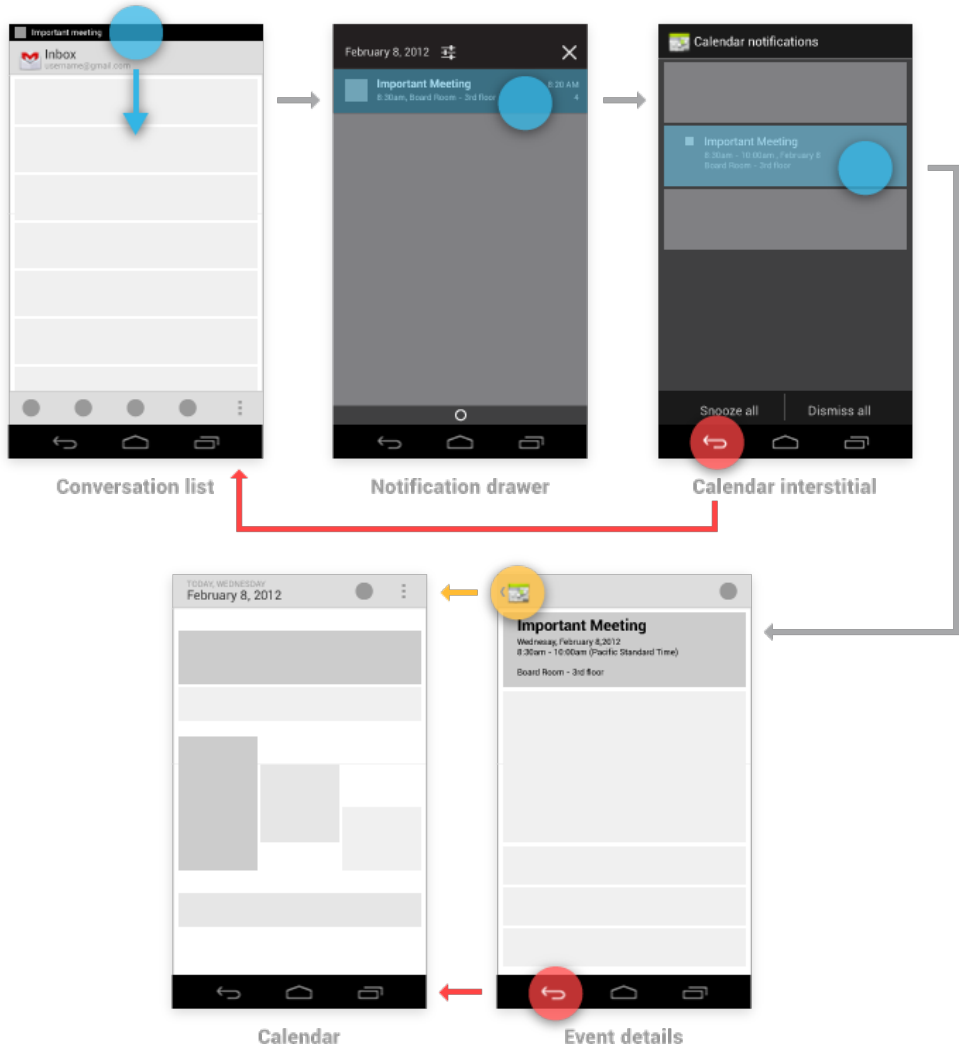
UNIVERSITÉ DE NANTES



- For both of these cases, handle the Up button as follows:
  - If the destination screen is typically reached from one particular screen within your app, Up should navigate to that screen.
  - Otherwise, Up should navigate to the topmost ("Home") screen of your app.

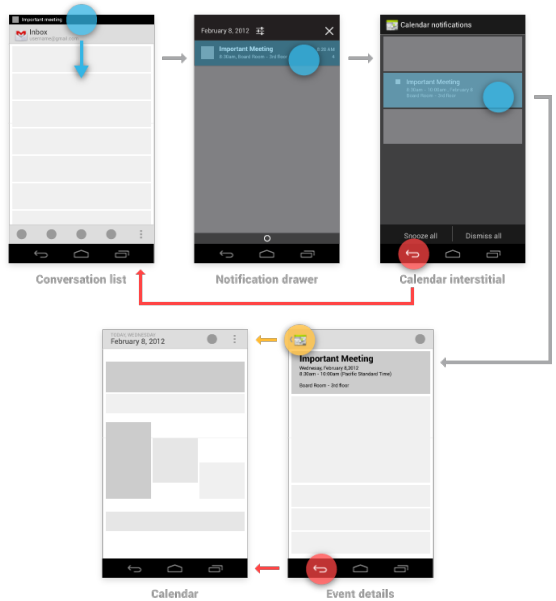
Indirect Notifications

UNIVERSITÉ DE NANTES



Indirect Notifications(2)

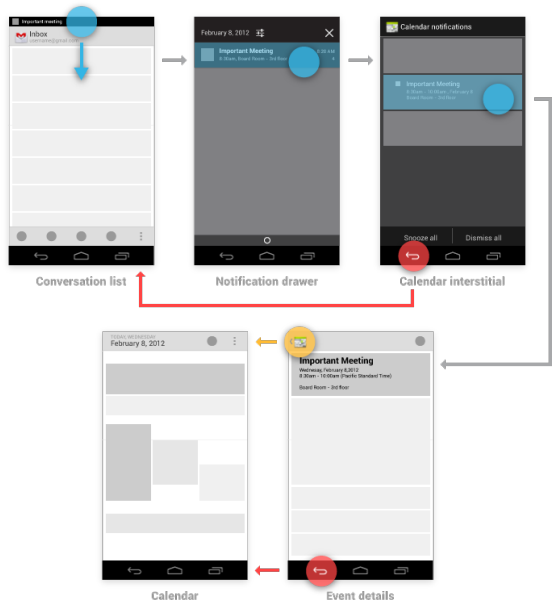
UNIVERSITÉ DE NANTES



- needs to present information about multiple events simultaneously
  - it can use a single notification that directs the user to an **interstitial screen**
  - This screen summarizes these events, and provides paths for the user to dive deeply into the app
- Notifications of this style are called indirect notifications.

Indirect Notifications (3)

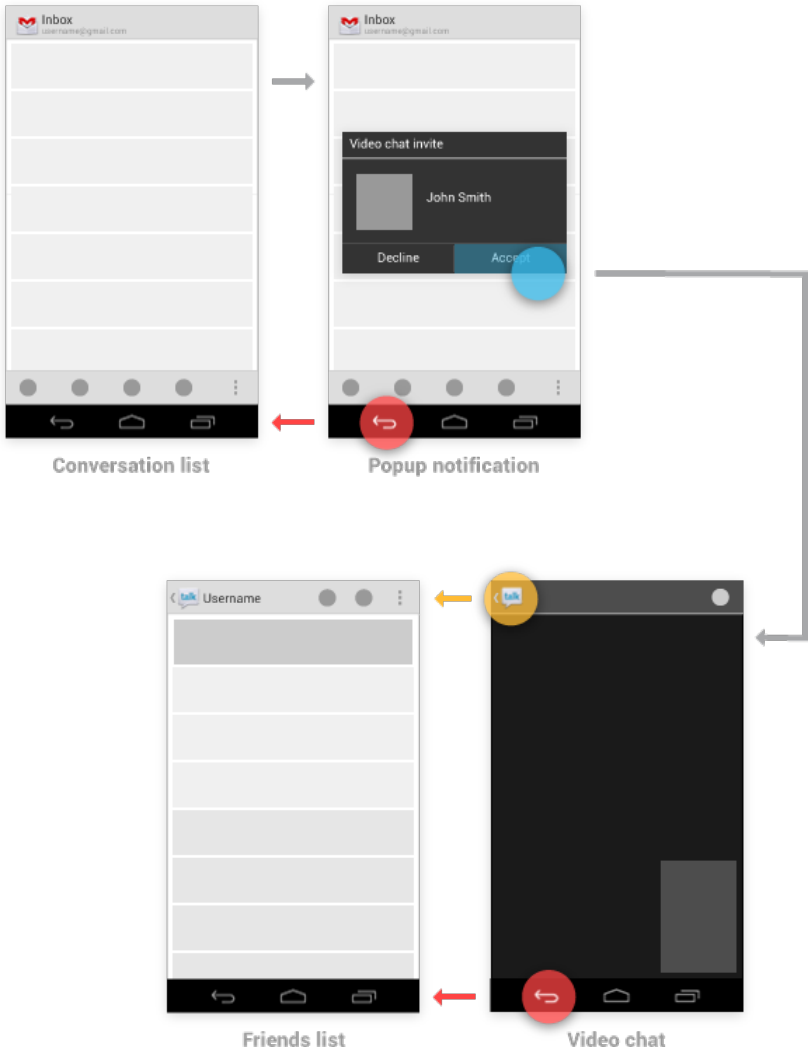
UNIVERSITÉ DE NANTES



- pressing Back from an indirect notification's interstitial screen returns the user to the point the notification was triggered from
- once into the app from its interstitial screen : navigating within the app rather than returning to the interstitial

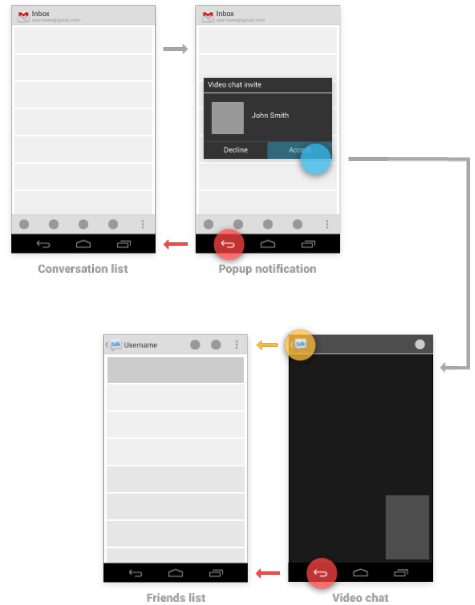
Pop-up Notifications

UNIVERSITÉ DE NANTES



Pop-up Notifications

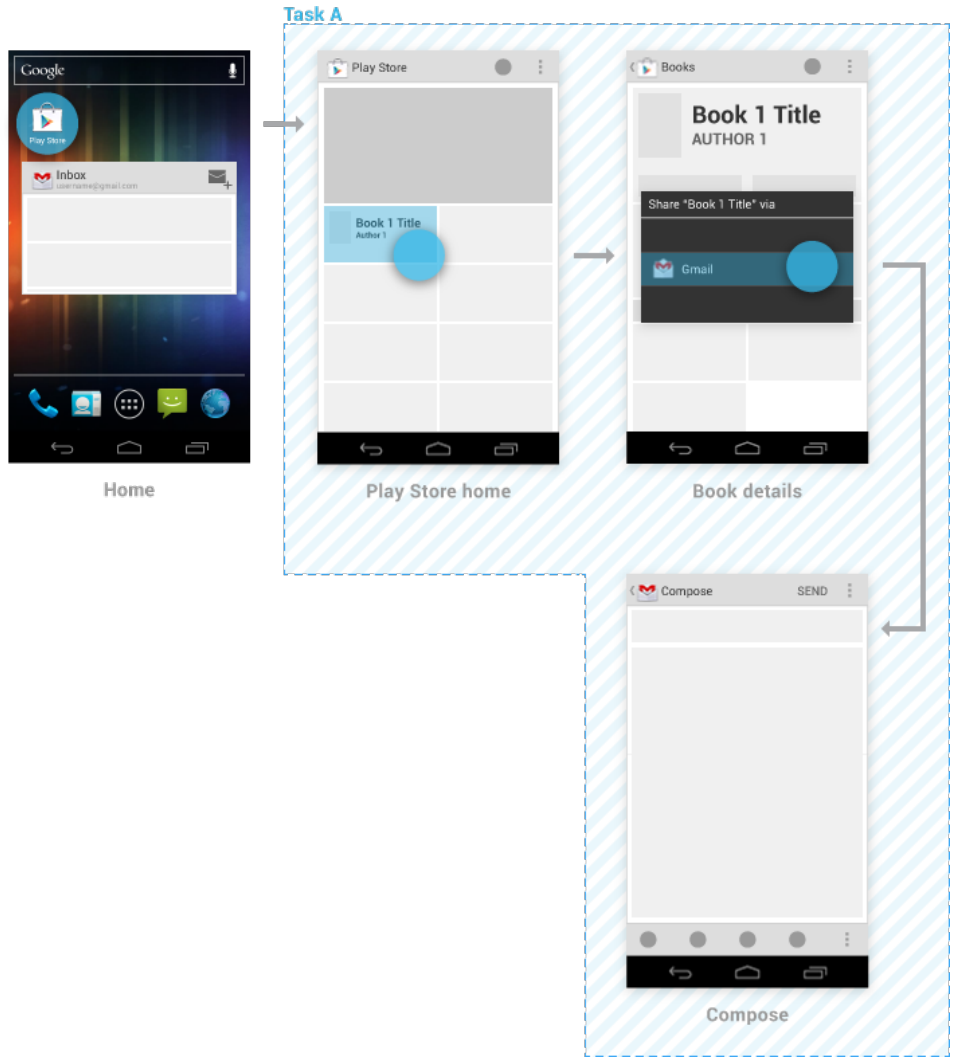
UNIVERSITÉ DE NANTES



- reserved for occasions where a timely response is required and the interruption of the user's context is necessary
- Pop-up notifications appear directly in front of the user (bypass the notification drawer, instead)
- pop-up notification navigation = indirect notification's interstitial screen navigation

Navigation Between Apps

UNIVERSITÉ DE NANTES



Navigation Between Apps

UNIVERSITÉ DE NANTES

- Consider how one app allows users to share content by using another app
- launching the Play Store app from Home begins new Task A.
- After touching a promoted book to see its details, the user remains in the same task

Navigation Between Apps (2)

UNIVERSITÉ DE NANTES

- Triggering the Share action prompts the user with a dialog listing each of the reactive activities (intentResolver).
- i.e Gmail's compose activity is added as a continuation of Task A
- From the compose activity, sending the message or touching the Back button returns the user to the book details activity.
- Subsequent touches of Back : typically apps navigation

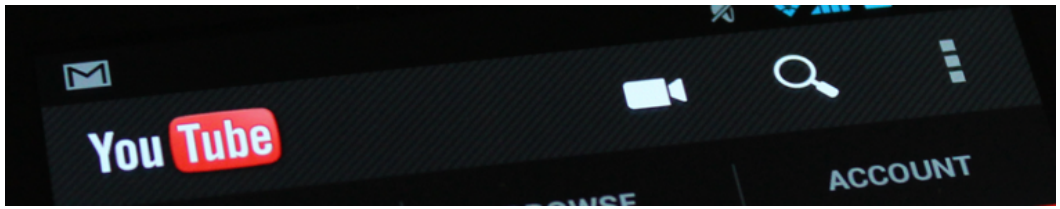
Navigation Between Apps (3)

UNIVERSITÉ DE NANTES

- touching Up from the compose activity, indicates a desire to remain within Gmail
- Gmail's conversation list activity appears (Task B is created for it).
- new task are always rooted to home (touching Back from the conversation list returns home).

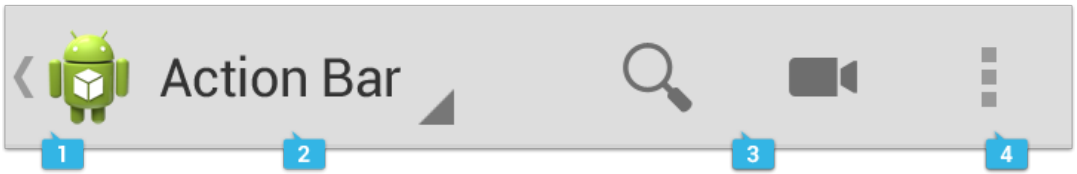
Action Bar

UNIVERSITÉ DE NANTES



- dedicated piece of real estate at the top of each screen, generally persistent throughout the app
- It provides several key functions:
  - Makes important actions prominent and accessible in a predictable way (such as New or Search).
  - Supports consistent navigation and view switching within apps.
  - Reduces clutter by providing an action overflow for rarely used actions.
  - Provides a dedicated space for giving your app an identity.

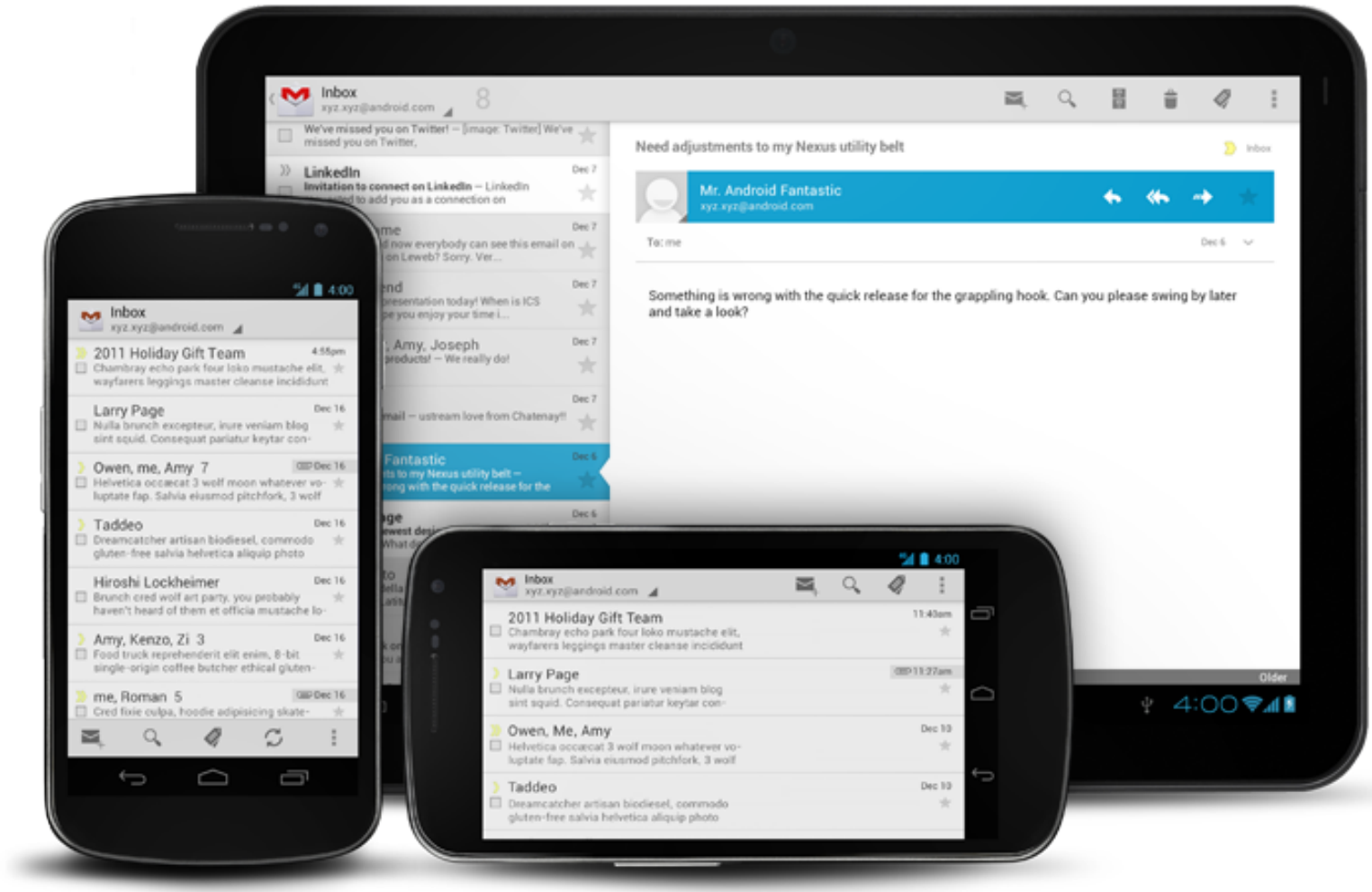
General Organisation



- split into four different areas
  1. App icon (indicates the up bracket if not at start screen)
  2. View control (switch views or non interactive content)
  3. Action buttons (show most important actions of the app)
  4. Action overflow (contains the actions that don't fit in the actions buttons)

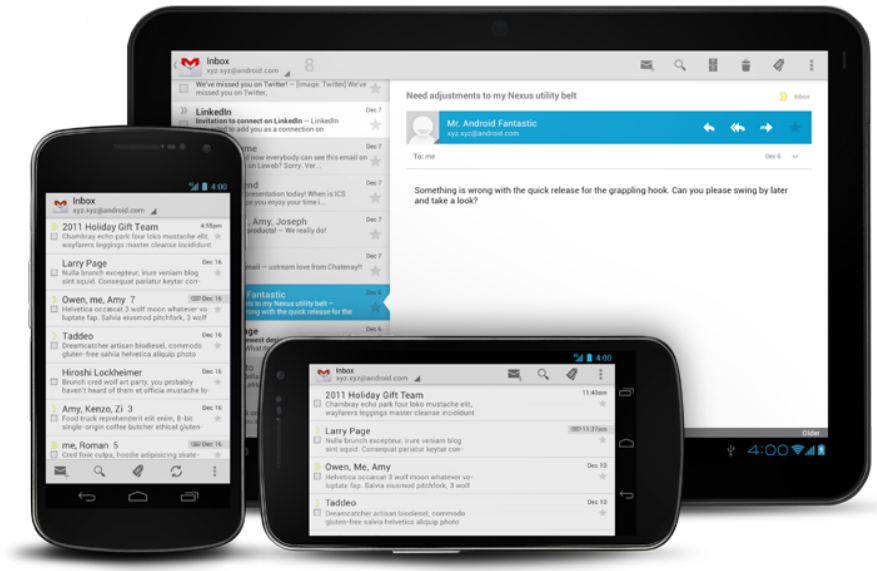
Rotation and Different Screen Sizes

UNIVERSITÉ DE NANTES



Rotation and Different Screen Sizes(2)

UNIVERSITÉ DE NANTES

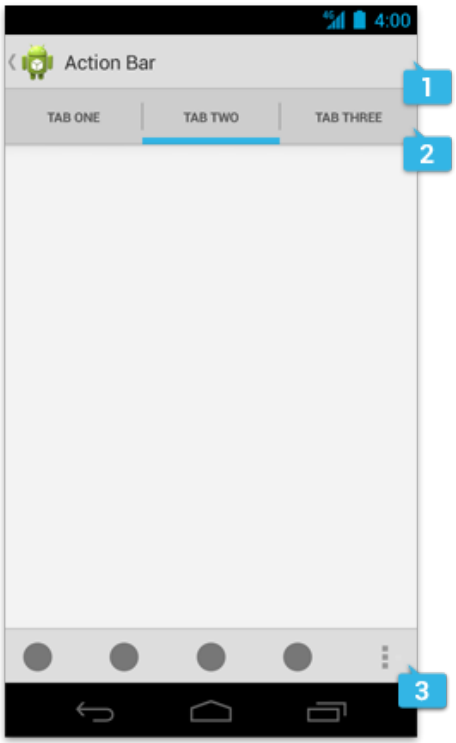


- adjust to screen rotation on different screen sizes.
- using split action bars,
- which allow you to distribute action bar content across multiple bars
  - located below the main action bar or
  - at the bottom of the screen.

Split Action Bars and Locations

UNIVERSITÉ DE NANTES

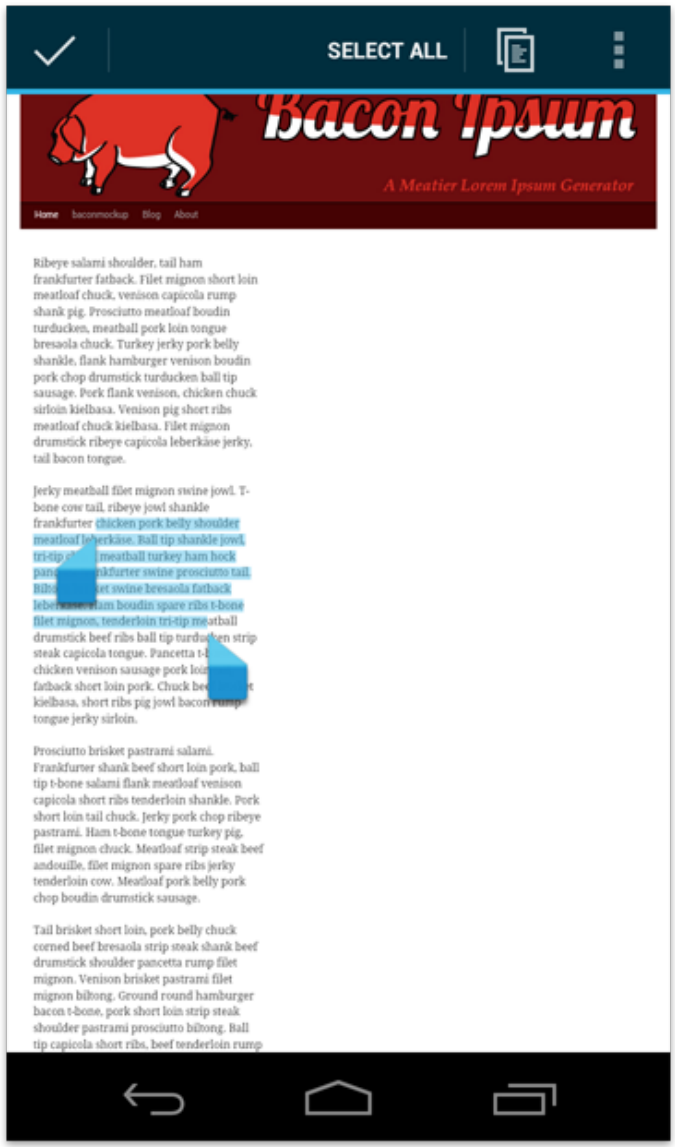
When splitting up content across multiple action bars, you generally have three possible locations for action bar content:

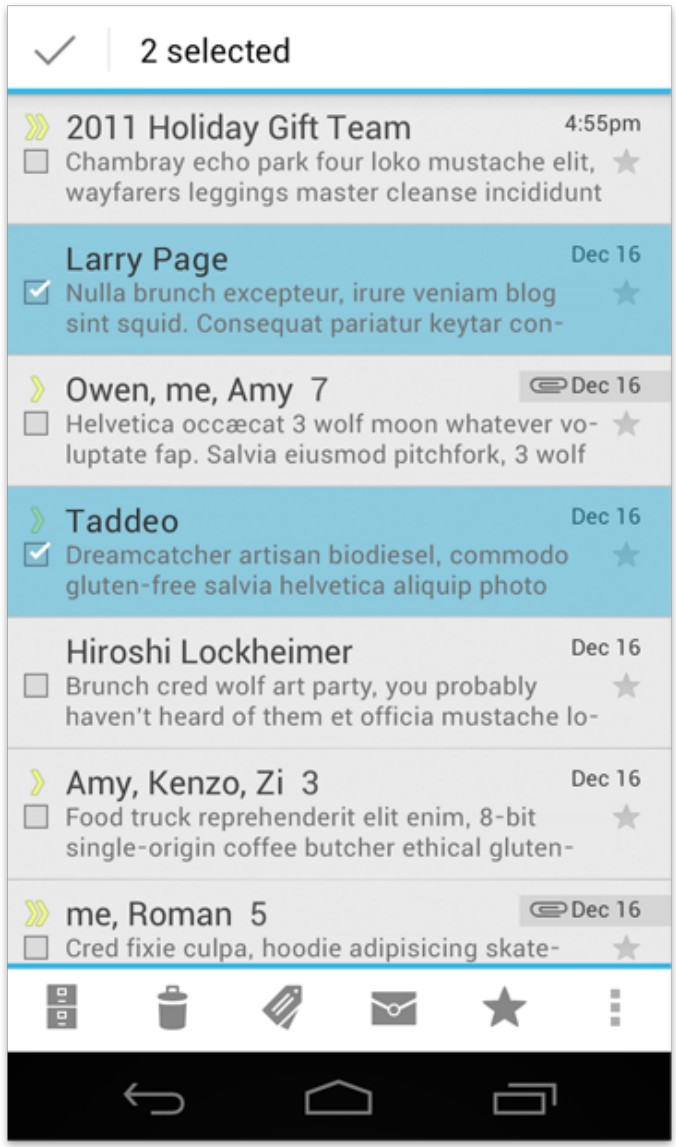


1. Main action bar (the main action bar contains the up caret, at a minimum.)
2. Top bar (to switch between the views your app provides, use tabs or a spinner in the top bar)
3. Bottom bar (To display actions and, if necessary, the action overflow, use the bottom bar.)

Contextual Action Bars

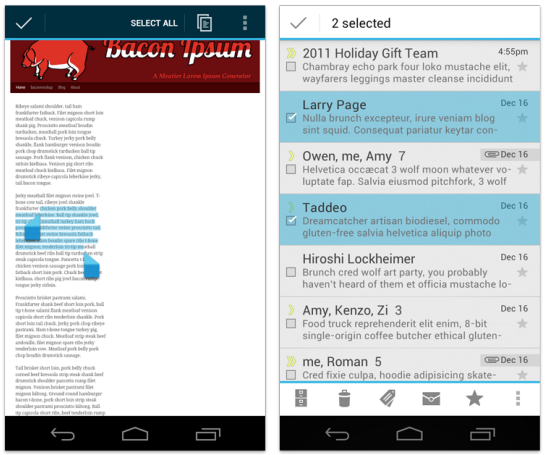
UNIVERSITÉ DE NANTES





Contextual Action Bars(2)

UNIVERSITÉ DE NANTES



A contextual action bar (CAB) is a temporary action bar that

- overlays the app's action bar for the duration of a particular sub-task
- are most typically used for tasks that involve acting on selected data or text.

CAB(3)

UNIVERSITÉ DE NANTES

the Selection CAB appears after a long press on a selectable data item triggers selection mode

- Select additional elements by touching them.
- Trigger an action from the CAB that applies to all selected data items (The CAB then automatically dismisses itself.)
- Dismiss the CAB via the navigation bar's Back button or the CAB's checkmark button (This removes the CAB along with all selection highlights.)

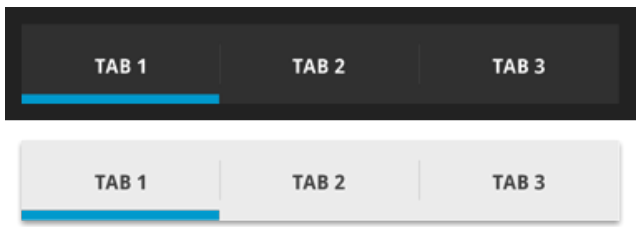
View Controls

- different views, the action bar has three different controls to allow users to switch between them:
  1. tabs (fixed tabs and scrollable tabs),
  2. spinners, and
  3. drawers.
- use tabs if users is expected to switch views frequently
- the user should be aware of the alternate views

Fixed Tabs

UNIVERSITÉ DE NANTES

- always visible on the screen
- Fixed tabs in the main action bar can move to the top bar when the screen orientation changes.
- support quick changes between two or three app views
- triggered by swiping left or right on the content area.



Scrollable Tabs

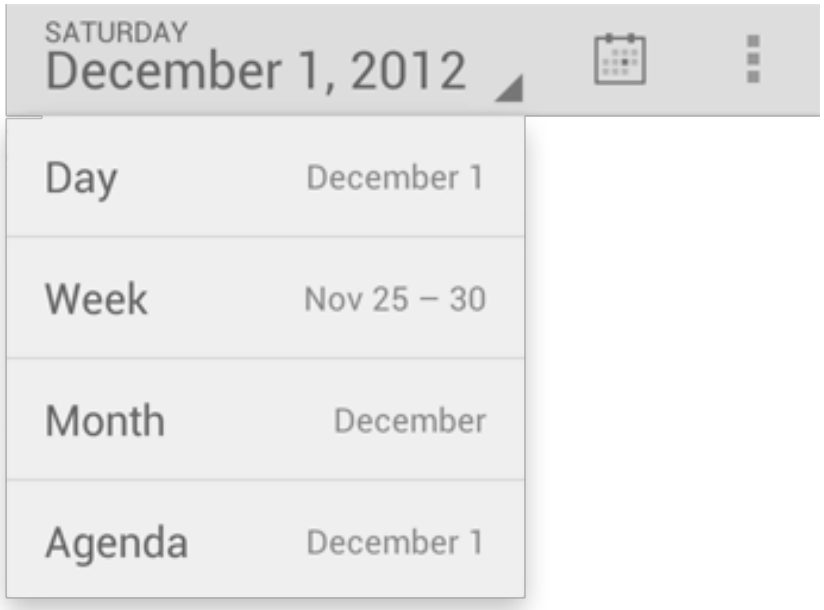
UNIVERSITÉ DE NANTES

- Scrollable tabs can themselves be scrolled horizontally to bring more tabs into view
- take up the entire width of the bar, with the currently active view item in the center (need to live in a dedicated bar).
- use if
  - large number of views or number of views is dynamic
  - triggered by swiping left or right on the content area.

Spinners

UNIVERSITÉ DE NANTES

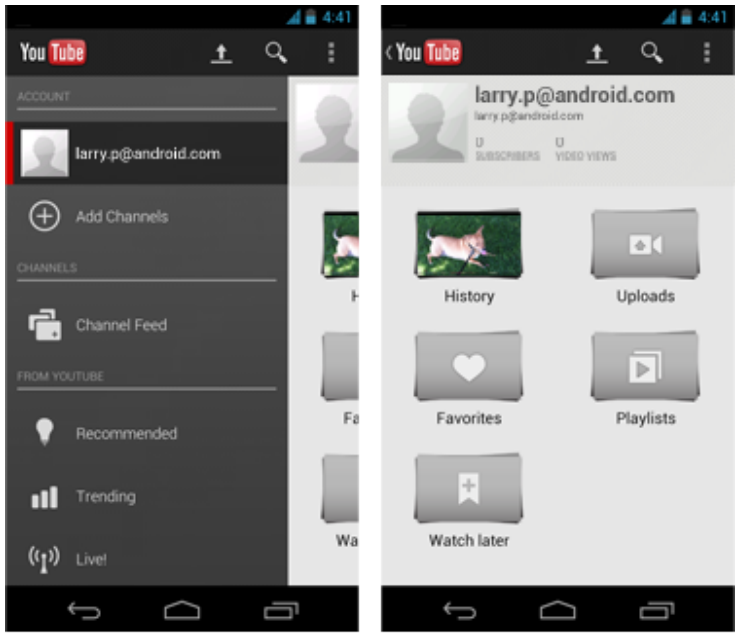
- A spinner is a drop-down menu that allows users to switch between views of your app.
- use if
  - don't want a dedicated tab bar to allow switching views.
  - there is consistency between views (same data set or data set of same type)



Drawers

UNIVERSITÉ DE NANTES

- A drawer is a slide-out menu that allows users to switch between views of your app
- can be opened by touching the action bar's app icon (so only suitable for the start screen)
- can be revealed by an edge swipe from the left of the screen
- use if:
  - don't want a dedicated tab bar to allow switching views.
  - want direct navigation to a number of views which don't have direct relationships between each other.



Action Buttons

UNIVERSITÉ DE NANTES

- Depending on available screen real estate, the system shows your most important actions as action buttons and moves the rest to the action overflow
- buttons will get used most often, and order them accordingly
- If an action is unavailable in the current context, hide it. Do not show it as disabled. (mode?)

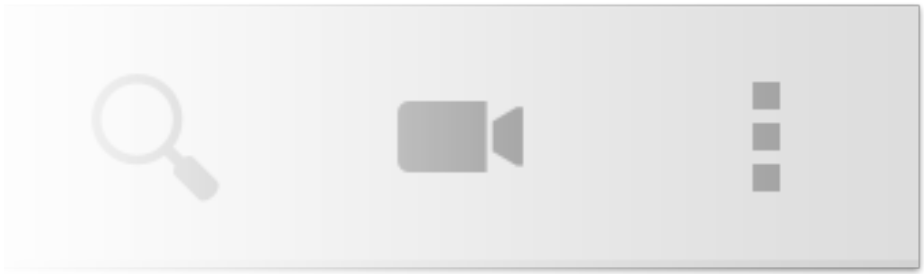
the FIT scheme

UNIVERSITÉ DE NANTES

- F is for Frequent
  - use this action at least 7 out of 10 times they visit the screen?
  - typically use it several times in a row?
  - an extra step would be annoying
- I is for Important
  - the action should be discovered
  - need to be effortless when it's needed
- T is for Typical
  - presented as a first class action in similar apps
  - given the context, would people be surprised if it were buried in the action overflow?
- If either F, I, or T apply, then it's appropriate for the action bar. Otherwise, it belongs in the action overflow.
- there exists android's pre-defined glyphs

Action Overflow

UNIVERSITÉ DE NANTES

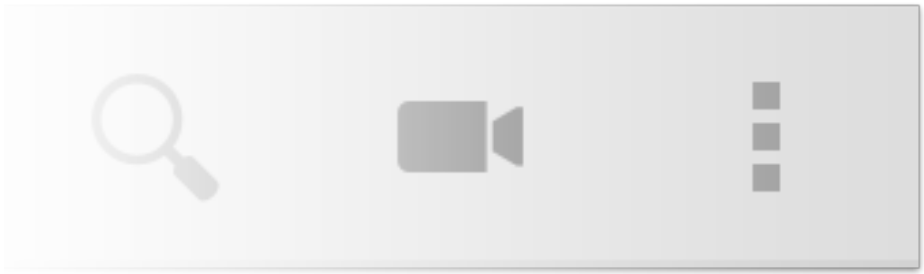


On the right side

- provides access to your app's less frequently used actions
- appears on phones that have no menu hardware keys.
- Phones with menu keys display the action overflow when the user presses the key.
- Action bar capacity is controlled by the following rules:

How many actions?

UNIVERSITÉ DE NANTES



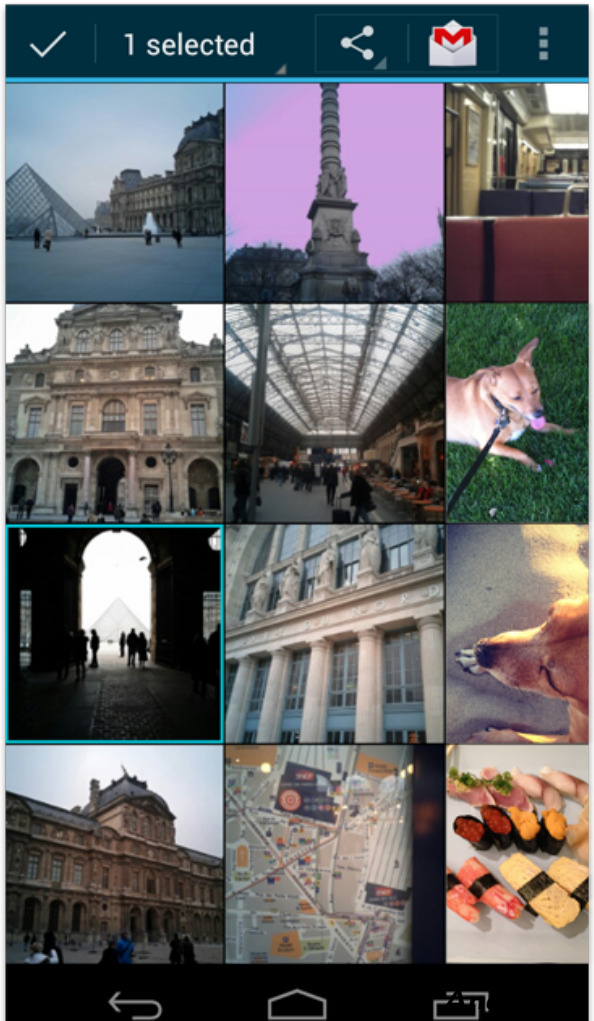
How many actions will fit in the main action bar?

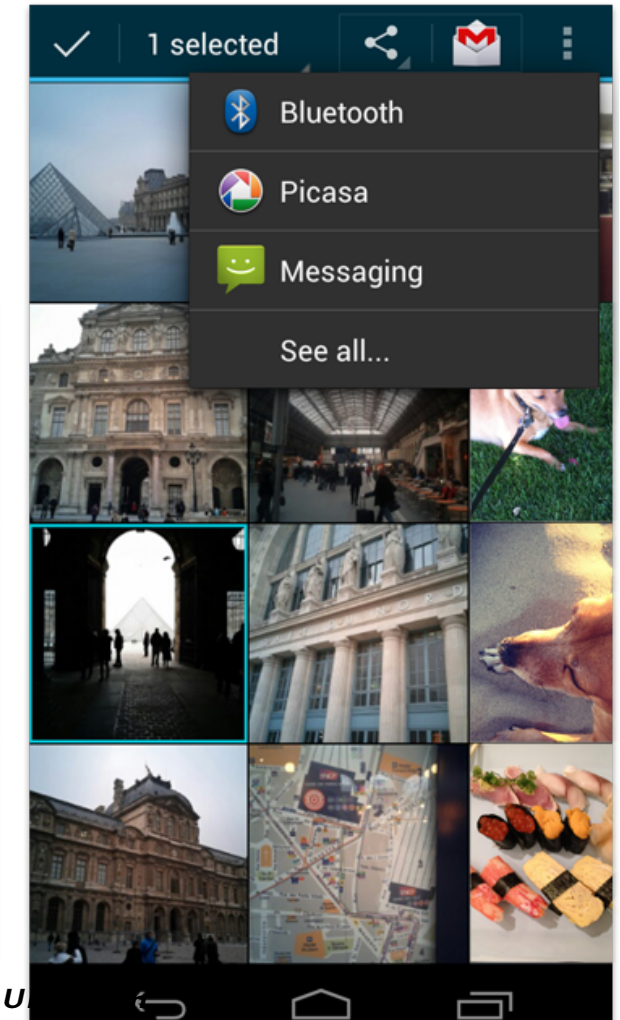
- Action buttons in the main action bar may not occupy more than 50 % of the bar's width.
- Action buttons on bottom action bars can use the entire width.
- The screen width in density-independent pixels (dp) determine the number of items that will fit in the main action bar:
  - smaller than 360 dp = 2 icons
  - 360-499 dp = 3 icons
  - 500-599 dp = 4 icons
  - 600 dp and larger = 5 icons

Sharing data

UNIVERSITÉ DE NANTES

- use a share action provider in your action bar
- The share action provider displays the most recently used sharing service next to a spinner button that contains other sharing options.

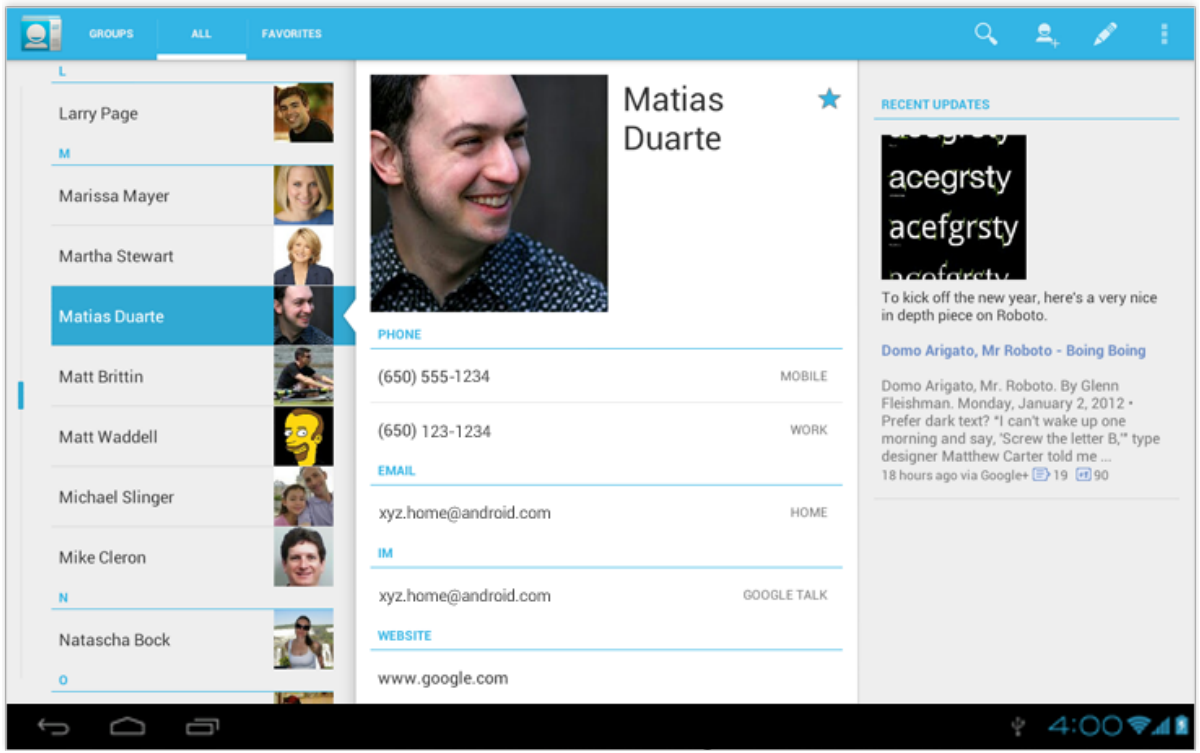




Multi-pane Layouts

UNIVERSITÉ DE NANTES

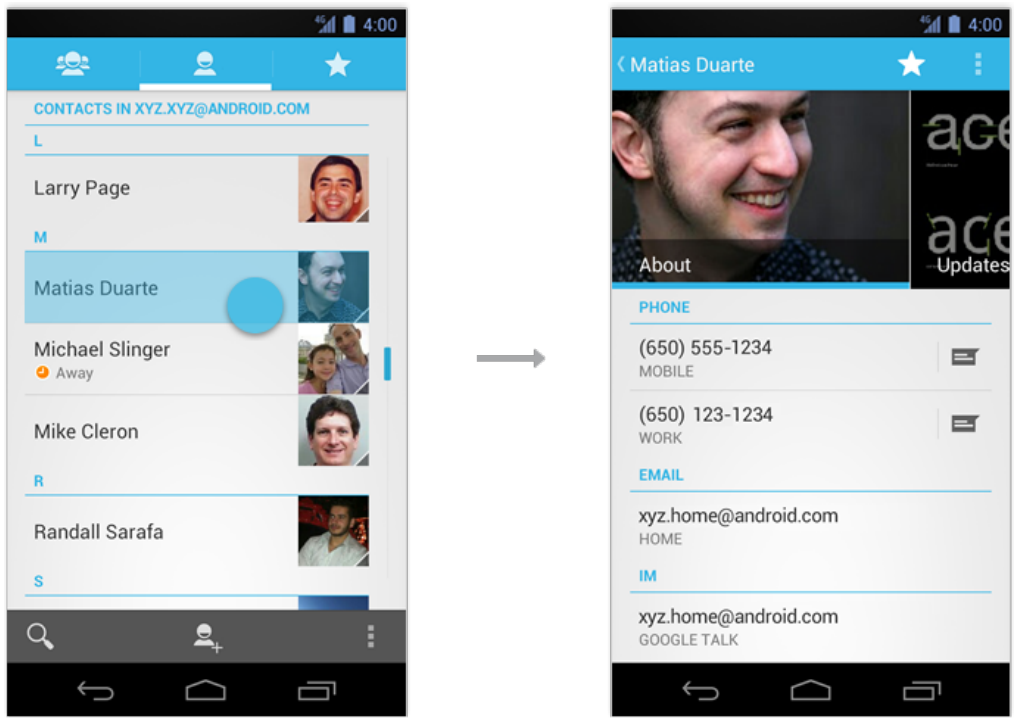
- Android devices come in many different screen sizes and types
- your app should provide a balanced and aesthetically pleasing layout by adjusting its content to varying screen sizes and orientations
- Panels allow you to combine multiple views into one compound view
- when a lot of horizontal screen real estate is available and
- by splitting them up when less space is available.



Combining Multiple Views Into One

UNIVERSITÉ DE NANTES

On small devices typically, a list view opens a detail view.



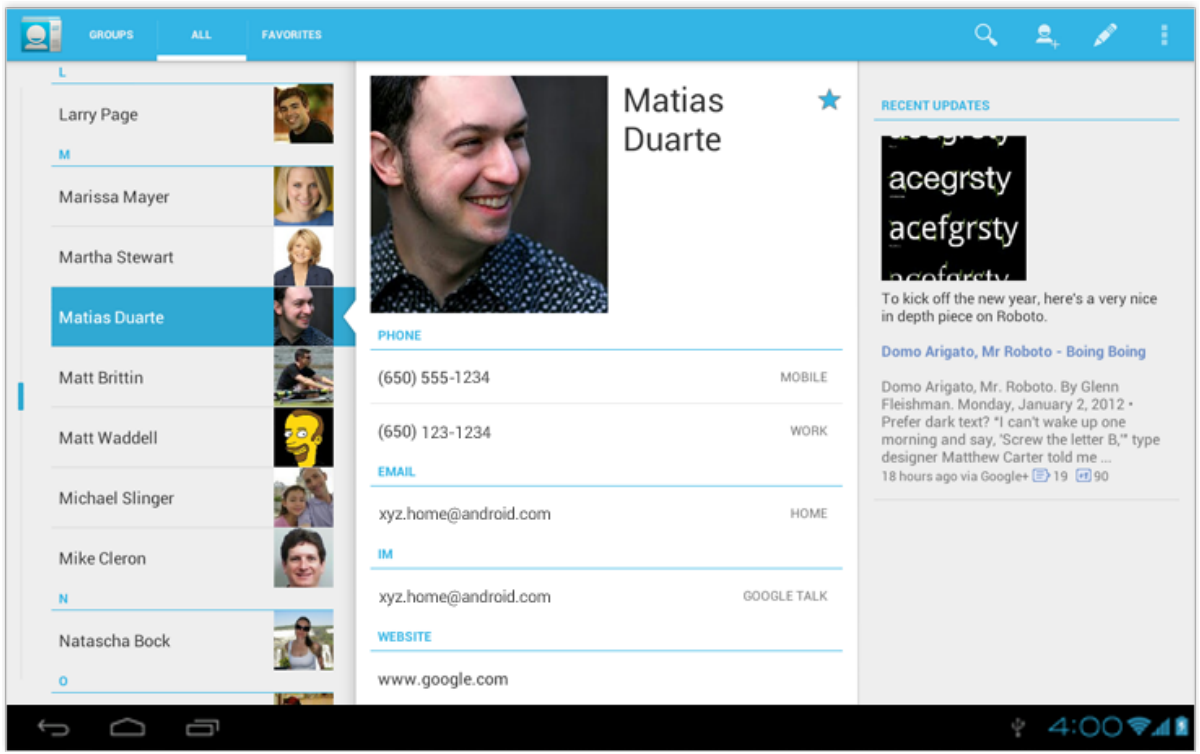
List view Detail view

Small Screen : 2 views

Combining Multiple Views Into One (2)

UNIVERSITÉ DE NANTES

- On tablets you can use panels to combine these two views into a single one.
- selection on the left (keep the item hi-lighted) and detailed view on the right

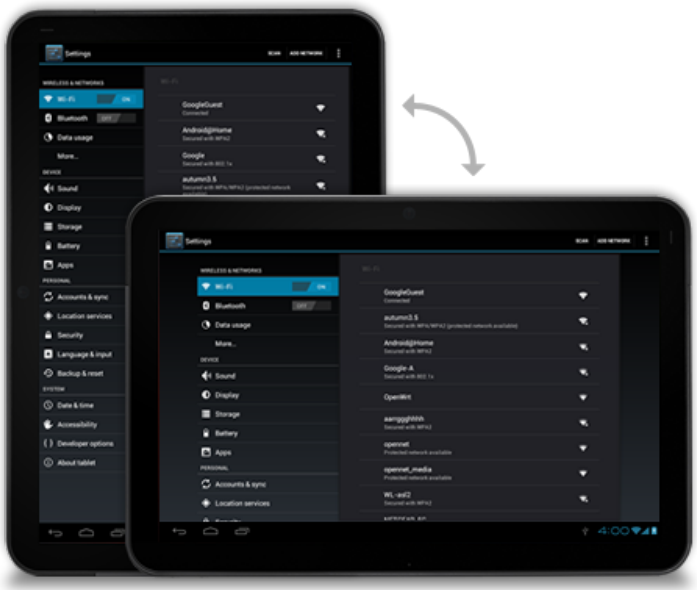


Small Screen : 2 views

Compound Views and Orientation Changes

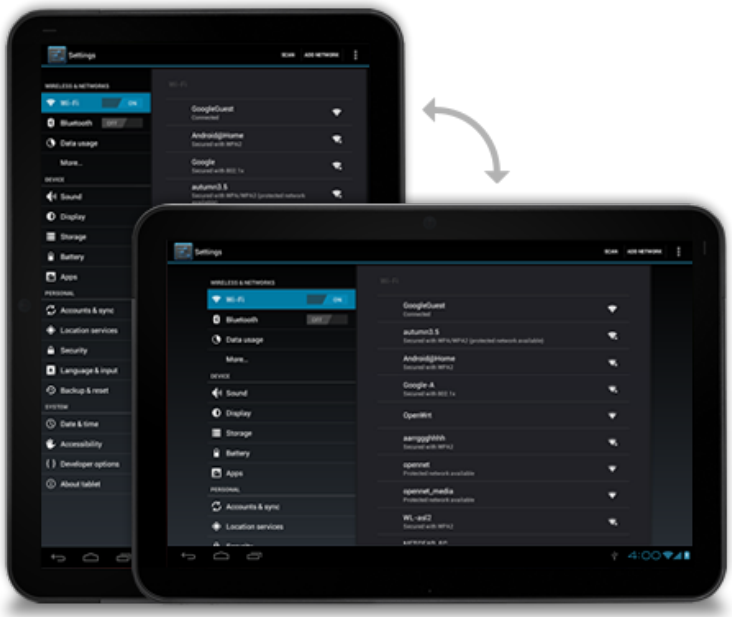
UNIVERSITÉ DE NANTES

- Screens should strive to have the same functionality regardless of orientation.
- If you use a compound view in one orientation, try not to split it up when the user rotates the screen.
- There are several techniques you can use to adjust the layout after orientation change while keeping functional parity intact.



Stretch/compress

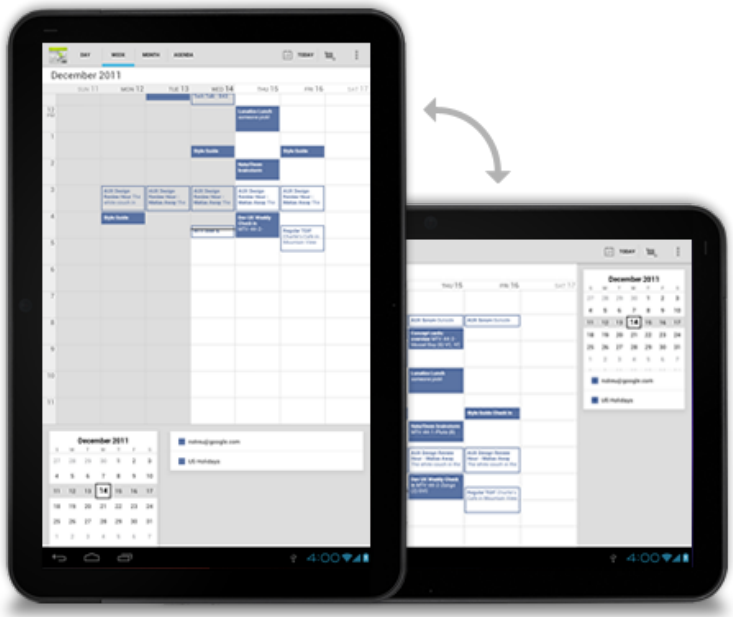
UNIVERSITÉ DE NANTES



- Adjust the column width of your left pane to achieve a balanced layout in both orientations.

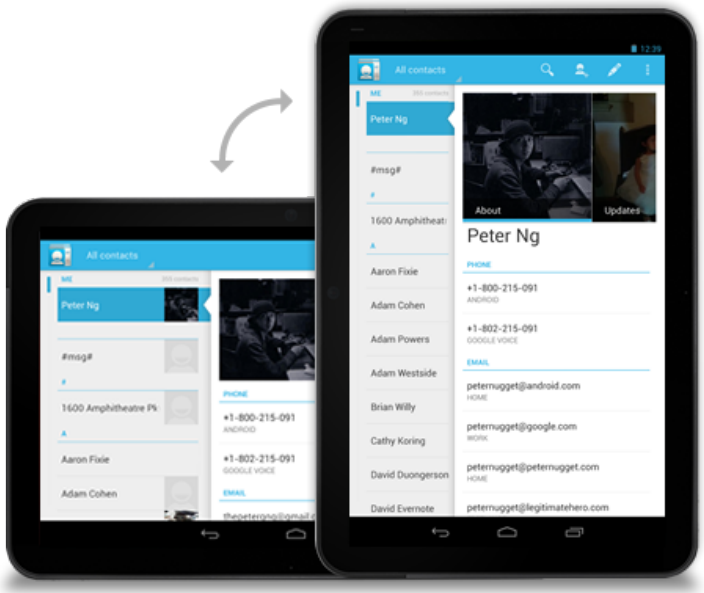
Stack

UNIVERSITÉ DE NANTES



- Rearrange the panels on your screen to match the orientation.

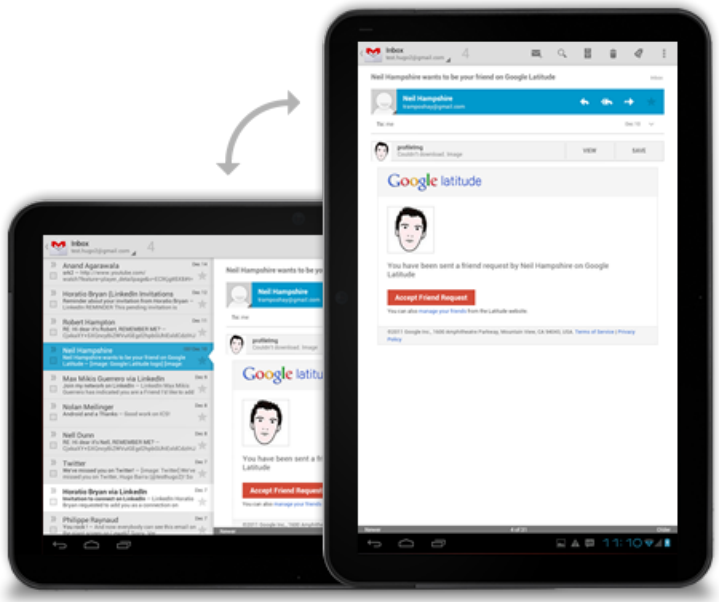
Expand/Collapse



- When the device rotates, collapse the left pane view to only show the most important information.

Show/Hide

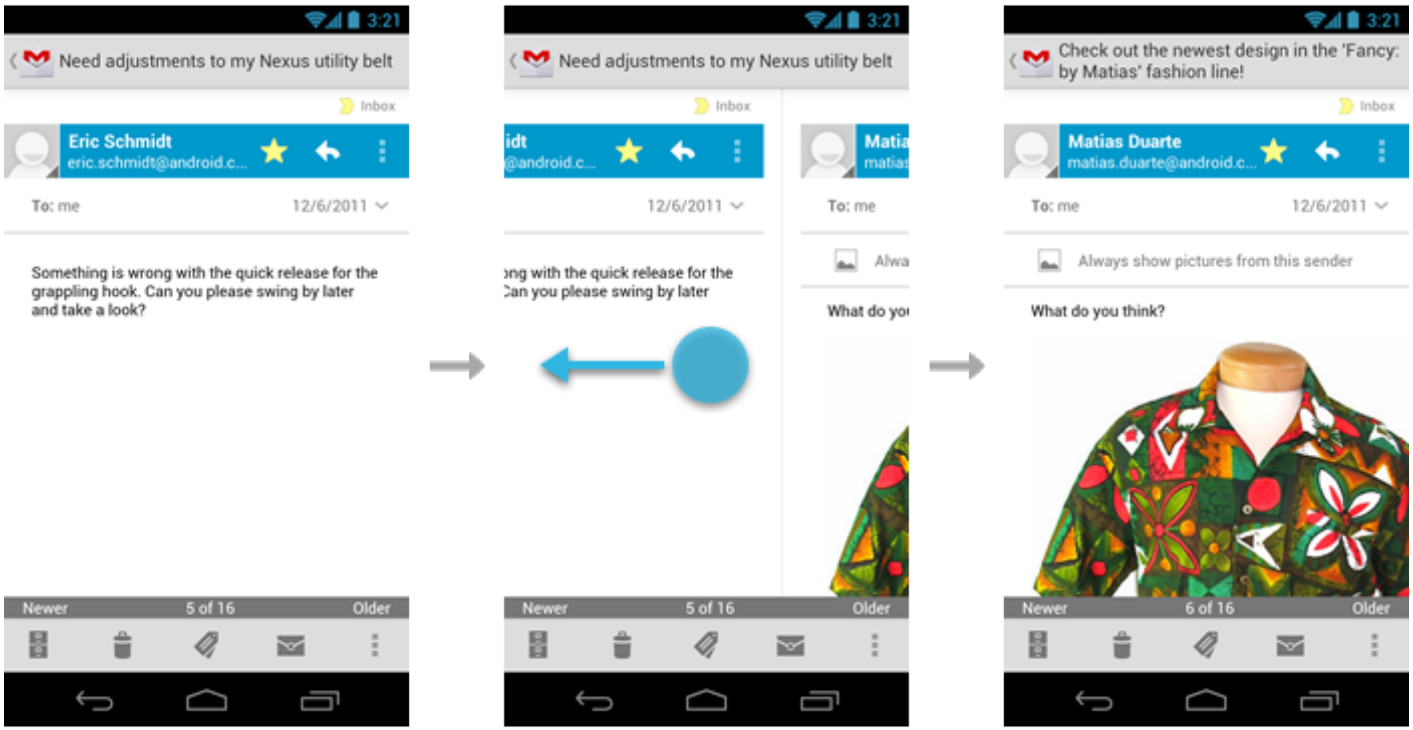
UNIVERSITÉ DE NANTES



- If your screen cannot accommodate the compound view on rotation
  - show the right pane in full screen view on rotation to portrait.
  - Use the Up icon in action bar to show the parent screen.

Swipe Views

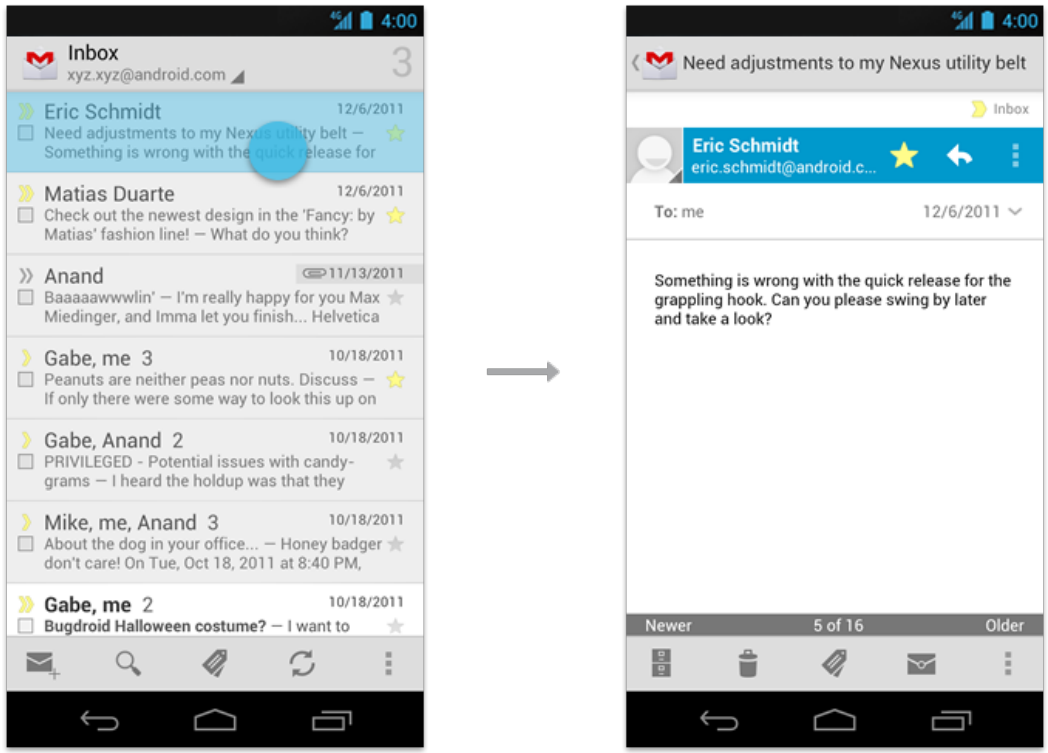
UNIVERSITÉ DE NANTES



- Apps are generally built in a hierarchical fashion,
- horizontal navigation can make access to related data items faster and more enjoyable.

Swipe Views(1)

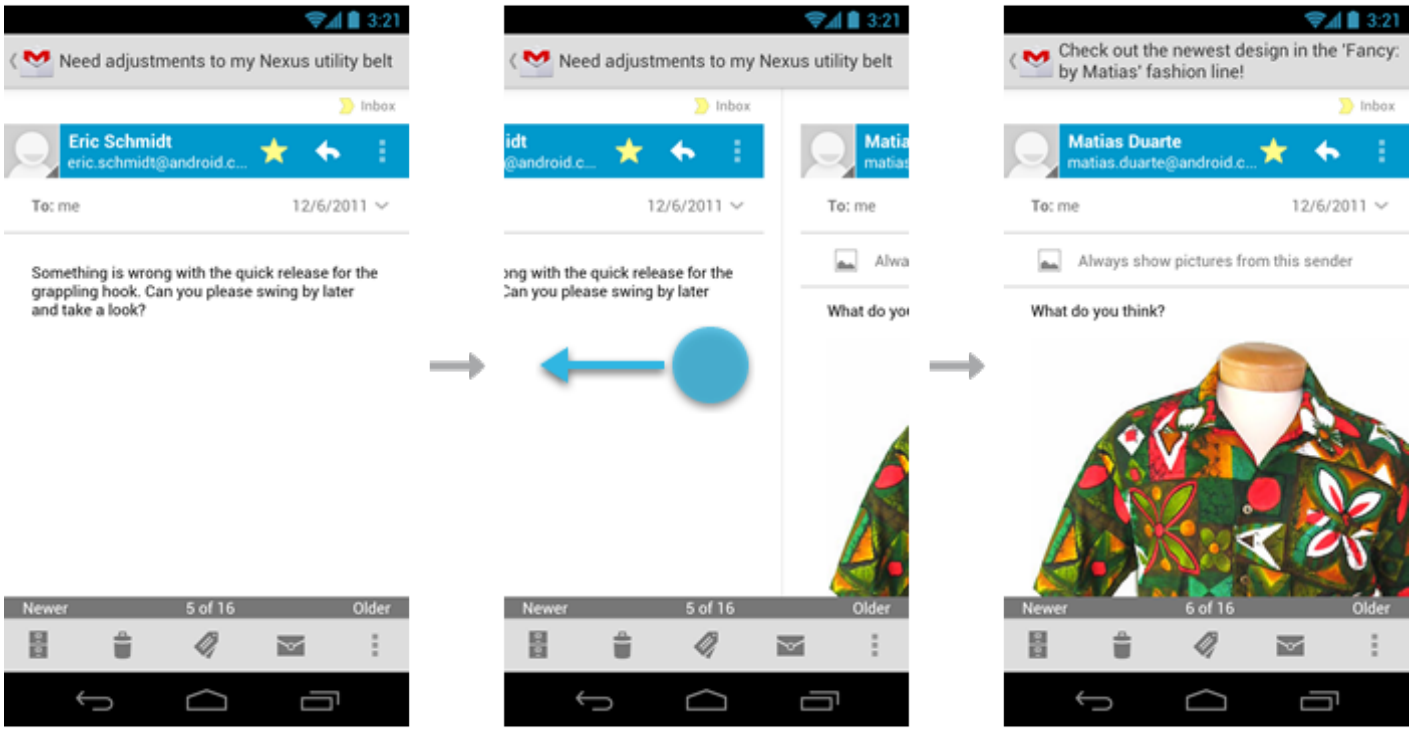
UNIVERSITÉ DE NANTES



Traditional, without swipe

- requires the user to jump back and forth between the list and the detail view, aka "pogo-sticking".

Swipe Views(2)

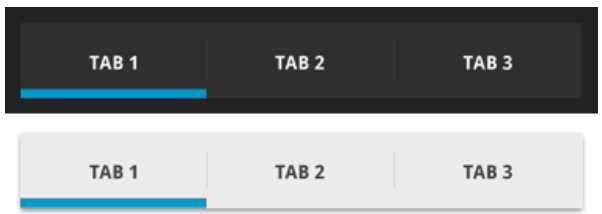


With swipe

- In cases where users will want to view multiple detail items in succession, avoid pogo-sticking by using the swipe gesture to navigate to the next/previous detail view.

Swiping Between Tabs

UNIVERSITÉ DE NANTES



- If your app uses action bar tabs, use swipe to navigate between the different views.

Not Presented

UNIVERSITÉ DE NANTES

- Selection
- Confirming & Acknowledging
- Notifications
- Widgets
- Settings

End

UNIVERSITÉ DE NANTES

END.