

# Mobile Interaction

## GUIs on Small Displays

# Lectures

#	Date	Topic
1	4.4.	Introduction to Mobile Interaction, Mobile Device Platforms
2	11.4.	Mobile Input and Output Technologies
3	18.4.	Text Input
4	25.4.	Touch Screen Gestures
5	2.5.	Touch Screen Interaction
	9.5.	no class (Christi Himmelfahrt)
6	16.5.	Visualization for Small Displays
	23.5.	no class (Pfingstwoche)
7	30.5.	Design, Prototyping, Evaluation
8	6.6.	Mobile Communication
9	13.6.	Location and Context
10	20.6.	Linking Mobile Devices to the World
11	27.6.	Sensor-Based Mobile Interaction
12	4.7.	Camera-Based Mobile Interaction
13	11.7.	Application Areas

# Preview

- Rapid serial visual presentation
- Glanceable displays
- Layout and navigation for small displays

# RAPID SERIAL VISUAL PRESENTATION (RSVP)

# Rapid Serial Visual Presentation (RSVP)

- Small displays have only little space for showing text
- Idea: Present text serially at a single visual location
  - One or more words at a time
  - Reduces need for scrolling
  - Reduces amount of eye movement during reading
  - Automatic serial presentation reduces controllability → frustration
- Example

Reading

200 wpm (0.3 spw)

1 sec pause at end of sentence

Reading

300 wpm (0.2 spw)

Reading

600 wpm (0.1 spw)

# Reading Performance Metrics

- Reading speed
  - Words per minute (wpm)
  - 250 wpm is typical
- Comprehension
  - Percentage of correctly answered questions
  - Multiple choice questions
- Reading experience
  - Subjective task load (physical, mental, temporal, effort, frustration)
  - "Ease" of the reading process
  - Depends on prior knowledge
- Eye movements
  - Observe differences depending on presentation formats

# Rapid Serial Visual Presentation (RSVP)

- First experiments 1970 (paper rolls, film strips)
- Reading comprehension tests (Juola et al., 1982)
  - Comparing (1) text on a page with (2) 200–700 wpm RSVP
  - Exposure times to text equal in both conditions
  - Participants were informed about exposure time before trial
- Results
  - Comprehension: RSVP equal to or slightly below page mode
  - Other studies found better comprehension for page mode than for RSVP
  - Comprehension is better if there is a pause between sentences

# Rapid Serial Visual Presentation (RSVP)

- Experiment to find optimum number of characters and rate (Cocklin et al., 1984)
  - Widths: 5.9–19.6 characters
  - Rates: 200–800 wpm
  - 72 participants
- Results
  - Comprehension lower for faster rates
  - Optimal width: 12 characters (2–3 words), best reading comprehension



# RSVP User Performance Parameters

- Adapting speed to word length, shape, or frequency
- Taking linguistic structure into account ("idea units")
  - Noun phrase: The old tree was struck by lightning.
  - Verb phrase: John gave Mary a book.
  - Adjective noun: The small tree blooms.
  - Prepositional phrases: She walked up the stairs.
- Offering user interaction
  - Control speed, go back one sentence, etc.
- Learning from user behavior
  - Tracking eye movements, adapting speed

# Study on Mobile Text Presentation Methods

## Scrolling



Arrow keys scroll text vertically line byline.

## Paging



Text divided into pages that fit the screen. Arrow keys to switch pages.

## Leading



A single line of text scrolling horizontally pixel by pixel.

## RSVP



One or a few words at a fixed location on the screen.

Progress bars show location in text.

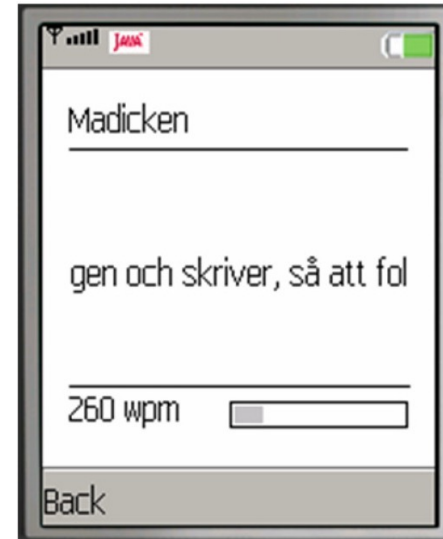
# Study on Mobile Text Presentation Methods

Dynamic formats

Leading and RSVP

- Controlled with joystick / numeric keypad
- Presentation speed (up/down)
- Backward/forward (left/right)
- Start/stop (joystick press)
- Initial speed: 250 wpm

Leading



A single line of text scrolling horizontally pixel by pixel.

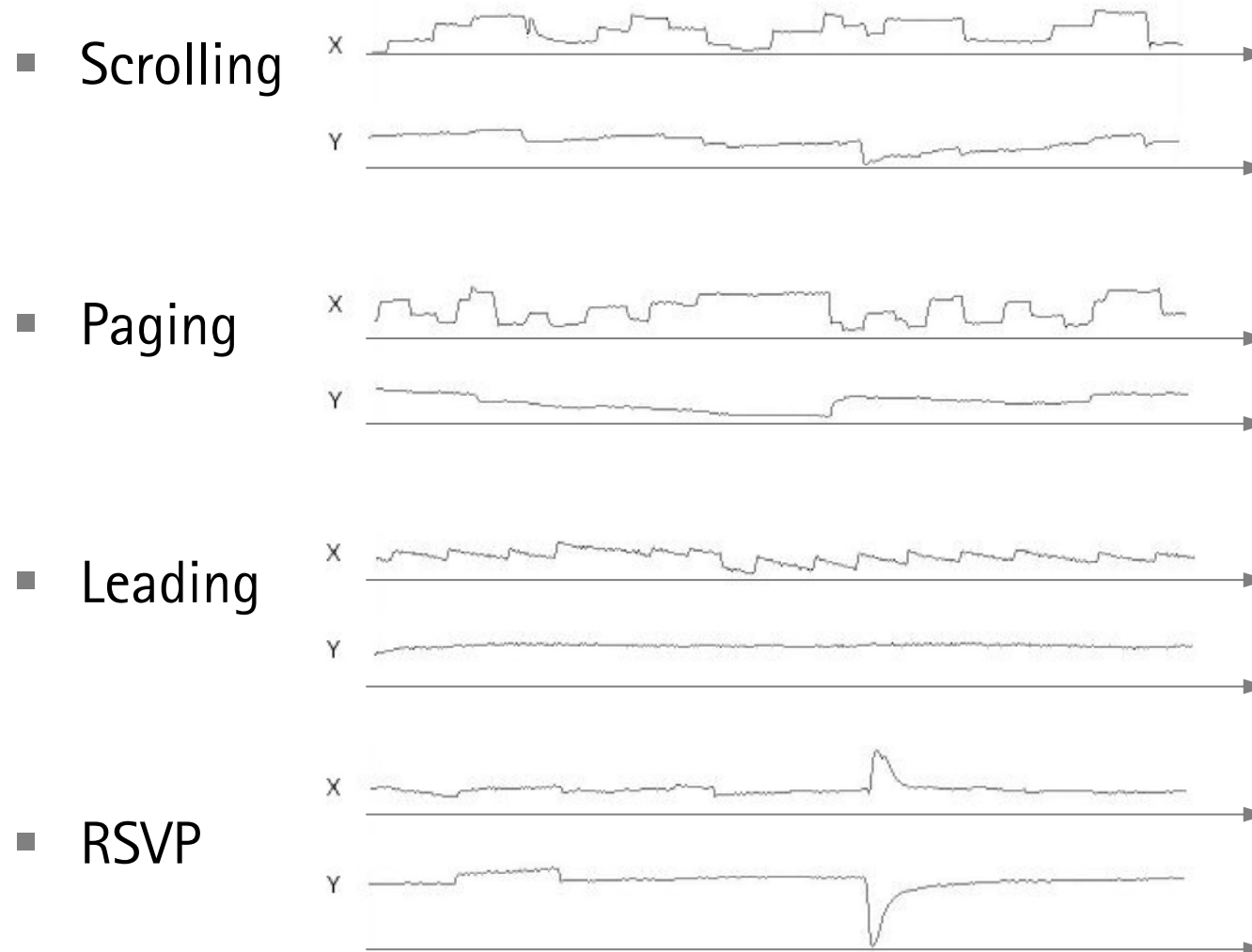
RSVP



One or a few words at a fixed location on the screen.

Progress bars show location in text.

# Eye Movement Traces while Reading (10 sec)



eye tracking glasses, chin rest

Öquist, Lundin. [Eye movement study of reading text on a mobile phone using paging, scrolling, leading, and RSVP](#). MUM 2007.

# Results

- Reading speed
  - Scrolling: 178.1 wpm (sd = 60.1)
  - Paging: 217.7 wpm (sd = 70.7)
  - Leading: 195.2 wpm (sd = 56.4)
  - RSVP: 135.4 wpm (sd = 44.3)
- Comprehension scores
  - 87.5% or better for all conditions, no significant differences
- Subjective task load
  - Mental demand for leading higher than for paging
  - Temporal demand for leading and RSVP higher than for scrolling and paging
- Eye movements
  - For RSVP fewest saccades, regressions, and eye blinks

Öquist, Lundin. Eye movement study of reading text on a mobile phone using paging, scrolling, leading, and RSVP. MUM 2007.

# Many RSVP Readers for Mobile Devices

- Example: Spritz, single-word RSVP
- Optimal recognition point
  - Highlighting of a character that the eyes should focus on
  - Choose point so as to minimize eye movement
- Different word durations
  - Adaptation by log-length
- Pauses between sentences
- Benedetto et al., 2015
  - Impairs literal comprehension
  - Increases visual fatigue



Figure 1. RSVP alignment of words



Figure 2. Spritz alignment of words

Source: <http://www.spritzinc.com/blog/>

Benedetto et al.: Rapid serial visual presentation in reading: The case of Spritz, 2015. <https://doi.org/10.1016/j.chb.2014.12.043>

# GLANCEABLE DISPLAYS

# Glanceable Display

- Glance: Brief (~5 s) sessions in which individuals check activity levels with no further interaction
  - 70% of the usage of physical activity trackers driven by glances
- Glanceable display
  - Output in a location that users will frequently see
  - Live wallpaper, smartwatch face
  - Simple, stylized, aesthetic, or abstracted visualization
  - Quick and easy to pick up
  - Perceivable at the periphery of one's attention
- Awareness display
- Problem: Privacy risk

Consolvo et al. lowers or a Robot Army? Encouraging Awareness & Activity with Personal, Mobile Displays. UbiComp 2008. <https://doi.org/10.1145/1409635.1409644>



# UbiFit Glanceable Display: Garden Representing Physical Activity



at the beginning  
of the week



after one cardio  
workout



a full garden  
with variety



cardio



strength training



flexibility training



walk



“other”



primary goal met



alternate goal met



Consolvo et al. lowers or a Robot Army? Encouraging Awareness & Activity with Personal, Mobile Displays. UbiComp 2008. <https://doi.org/10.1145/1409635.1409644>

# UbiFit System

- Glanceable display + mobile app + activity tracker
- Glanceable display as smartphone background screen
  - Updated when a new activity is detected
  - Mobile awareness display
  - Subtle reminder whenever the phone is used
- Goal: Improving awareness of daily life
  - Here: self-monitoring of physical activity
- Glanceable awareness display

Stylized, aesthetic representation of physical activities and goal attainment to keep the individual focused on the act of self-monitoring and her commitment to fitness.

Consolvo et al. lowers or a Robot Army? Encouraging Awareness & Activity with Personal, Mobile Displays. UbiComp 2008. <https://doi.org/10.1145/1409635.1409644>

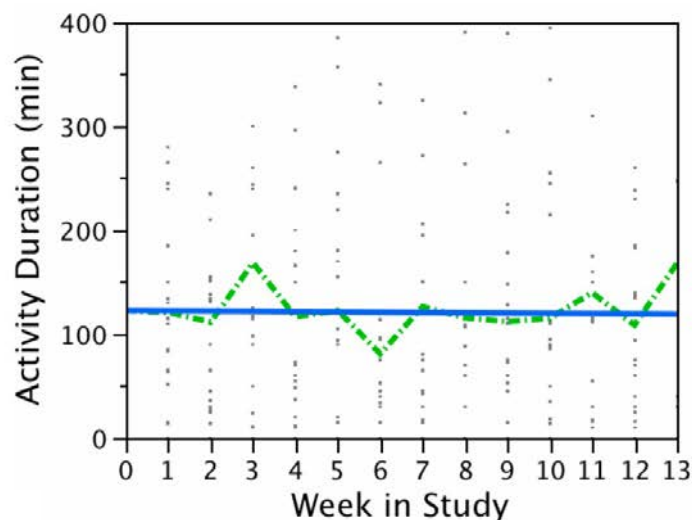
# UbiFit Field Study

- 28 participants
- 3 between-subjects conditions
  - Full system: glanceable display + mobile app + activity tracker
  - No activity tracker: glanceable display + mobile app
  - No glanceable display: mobile app + activity tracker
- 3 months in day-to-day life (over the winter holiday season)

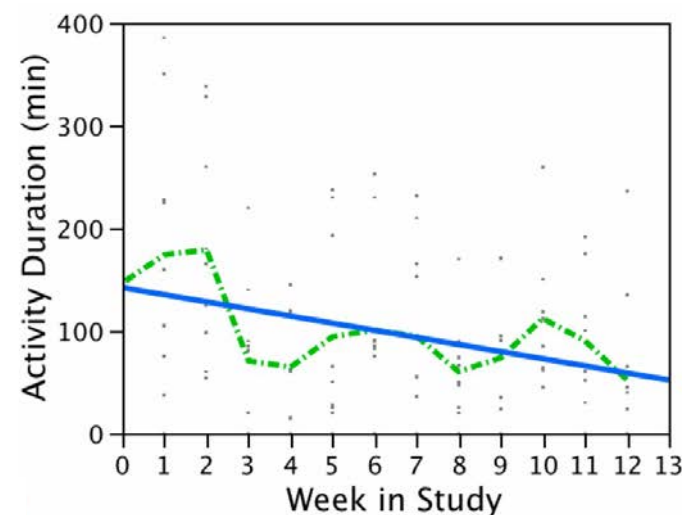
Consolvo et al. lowers or a Robot Army? Encouraging Awareness & Activity with Personal, Mobile Displays. UbiComp 2008. <https://doi.org/10.1145/1409635.1409644>

# UbiFit Field Study: Results

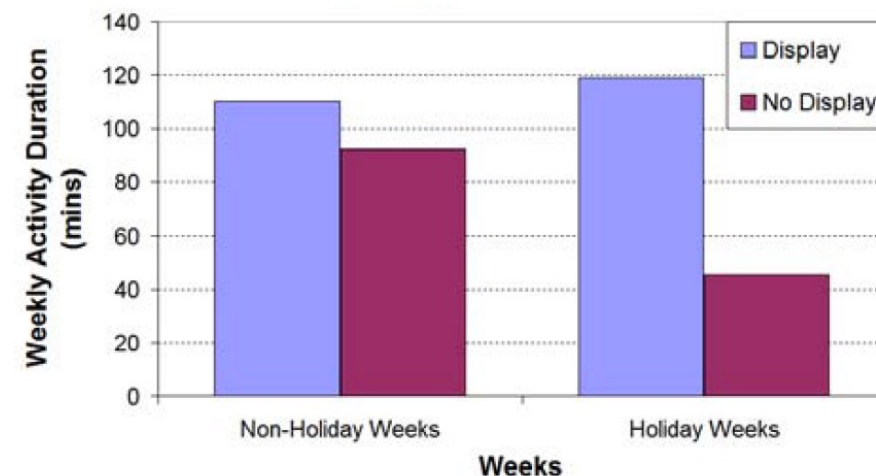
- With display: Participants maintained physical activity level
- Without display: Activity level dropped significantly



with glanceable display



without glanceable display



Consolvo et al. lowers or a Robot Army? Encouraging Awareness & Activity with Personal, Mobile Displays. UbiComp 2008. <https://doi.org/10.1145/1409635.1409644>

# Glanceable Displays for Physical Activity Tracking with Smartwatches



TickTock: activity over the past hour



Normly: one's goal completion in comparison to others with a similar goal



Gardy: a simplified variant of UbiFit's Garden



Goal completion: represents daily progress

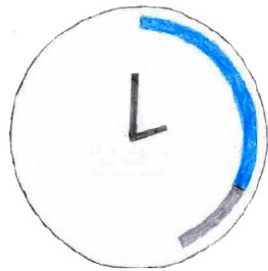
Gouveia et al. Exploring the design space of glanceable feedback for physical activity trackers. UbiComp 2016. <https://doi.org/10.1145/2971648.2971754>



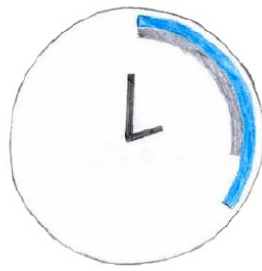
# Design Variations



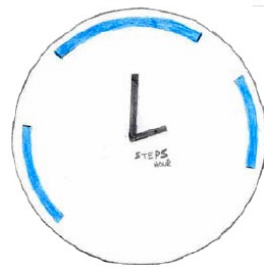
1 - Gardy



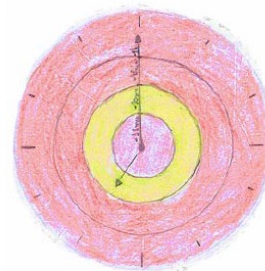
2 - Catchup



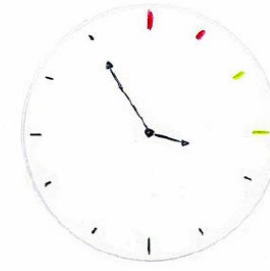
3 - Normly



4 - TickTock



5 - Move



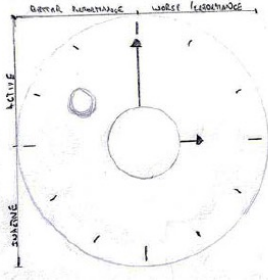
6 - ActiveHours



7 - PastPerform



8 - Geotivity



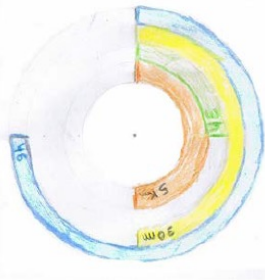
9 - BalanceYou



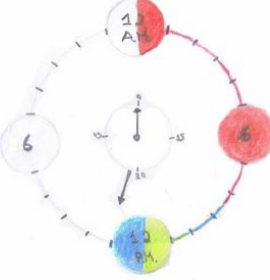
10 - PA Scale



11 - CrowdWalk



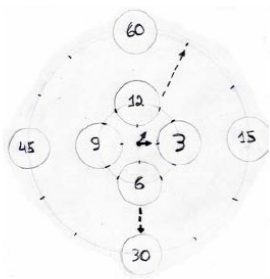
12 - SocialWalk



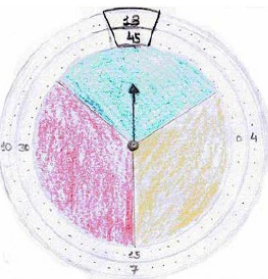
13 - Sections



14 - Predicto



15 - JustaWatch



16 - Rotations



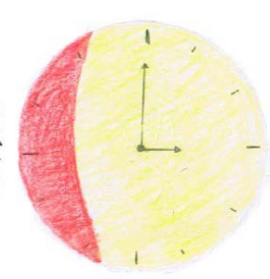
17 - Locals



18 - Scarcition



19 - DistantYou



20 - DayBalance



21 - Meanfull

Gouveia et al. Exploring the design space of glanceable feedback for physical activity trackers. UbiComp 2016. <https://doi.org/10.1145/2971648.2971754>

# Glanceable Displays for Physical Activity Tracking with Smartwatches

- Glance: Brief ( $\sim 5$  s) sessions in which individuals check activity levels with no further interaction
  - 70% of the usage of physical activity trackers driven by glances
- How to best design glanceable feedback?
- Design qualities
  - Being abstract
  - Integrating with activities
  - Comparison to targets and norms
  - Being actionable
  - Leading to checking habits: novelty, scarcity
  - Proxy to further engagement

Gouveia et al. Exploring the design space of glanceable feedback for physical activity trackers. UbiComp 2016. <https://doi.org/10.1145/2971648.2971754>

# Glanceable Displays for Physical Activity Tracking with Smartwatches

- Between-subjects study: 12 participants, 28 days



*Normly*



*TickTock*



*Goal Comp.*



*Gardy*

- ~100 watch interactions per day
- Normly demotivating if too far off
- Gardy was least preferred

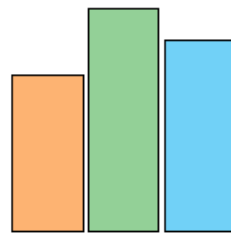
<i>Usage sessions</i>	122 (SD: 99)	110 (SD: 81)	108 (SD: 69)	86 (SD: 60)
<i>Step count</i>	5460 (SD: 4528)	5150 (SD: 4543)	5340 (SD: 4528)	3760 (SD: 3511)

Gouveia et al. Exploring the design space of glanceable feedback for physical activity trackers. UbiComp 2016. <https://doi.org/10.1145/2971648.2971754>

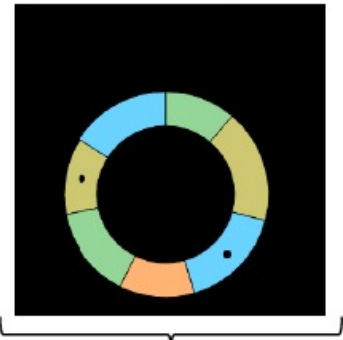


# Glanceable Data Visualizations

- Compare required glancing time for 3 visualizations
- Types: Bar, Donut, Radial
- Data size: 7, 12, 24
- 24 participants

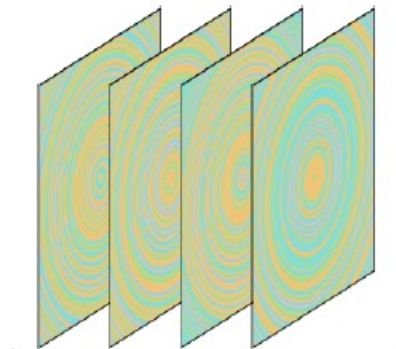


## Stimulus Exposure



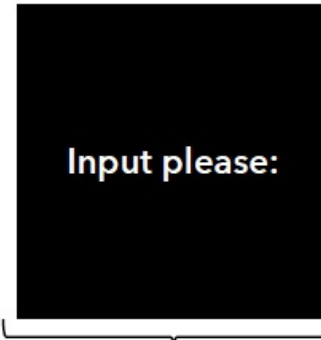
Time as per staircase

## Intervening Images



4 × 20 ms = 80 ms

## Participant Input



Wait until participant presses ☐ or ☐

## Feedback

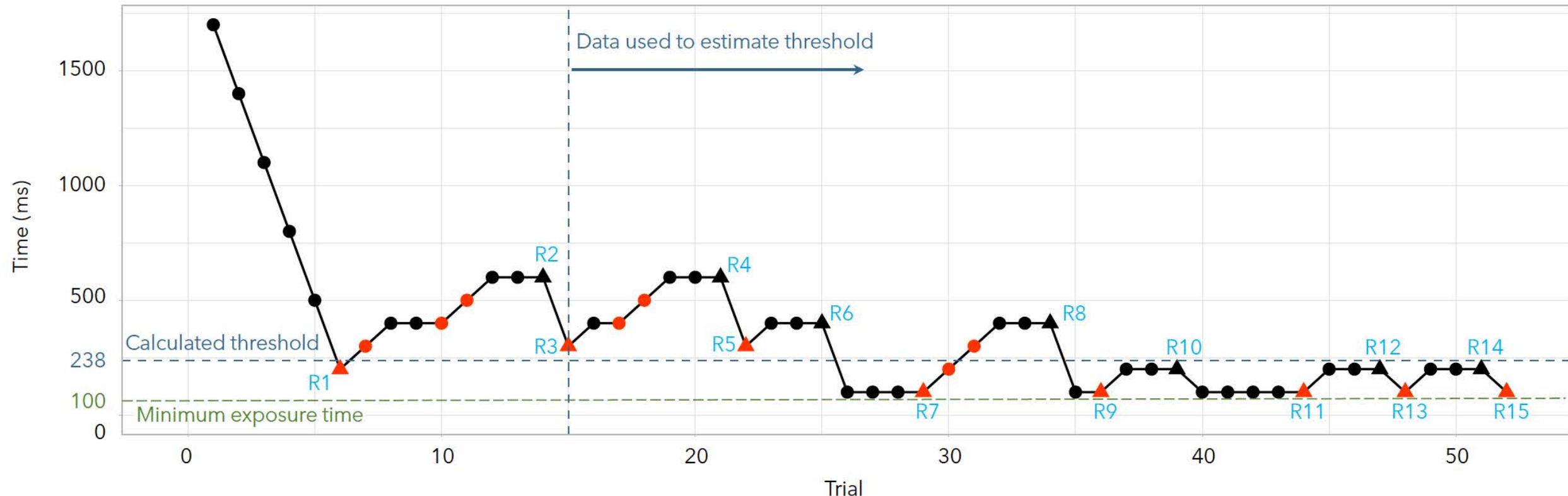


1000 ms

While et al. Glanceable Data Visualizations for Older Adults: Establishing Thresholds and Examining Disparities Between Age Groups. CHI 2024. <https://doi.org/10.1145/3613904.3642776>

# Determine Time Threshold using Staircase Procedure

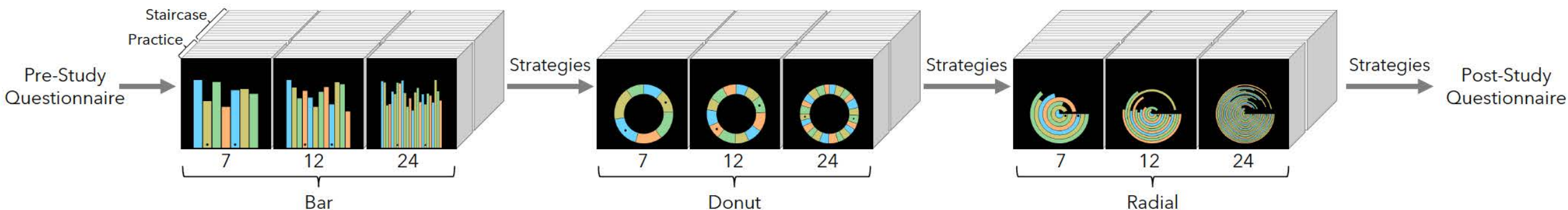
- 3 correct: down 300 ms, 1 wrong: up 100 ms
- threshold = mean(R3..R15)



While et al. Glanceable Data Visualizations for Older Adults: Establishing Thresholds and Examining Disparities Between Age Groups. CHI 2024. <https://doi.org/10.1145/3613904.3642776>

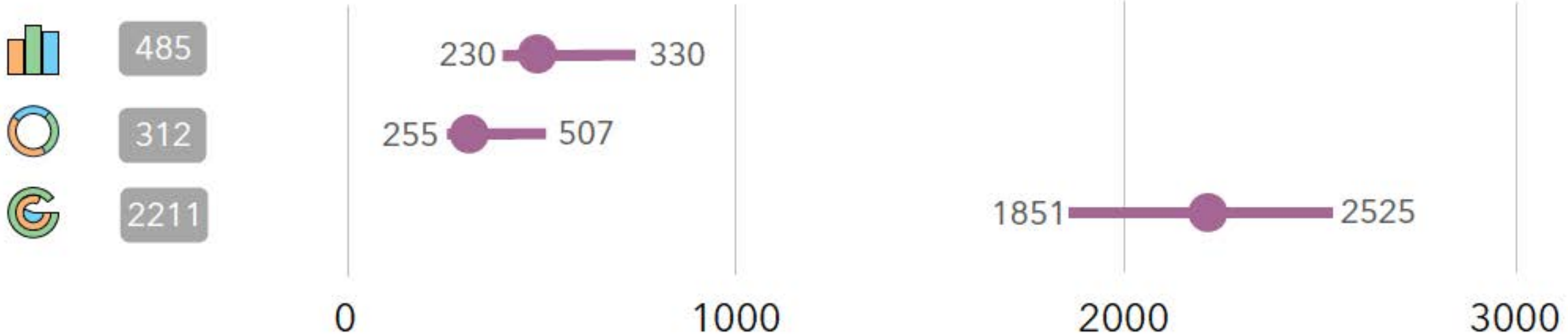
# Glanceable Data Visualizations

order of chart types and data sizes counterbalanced



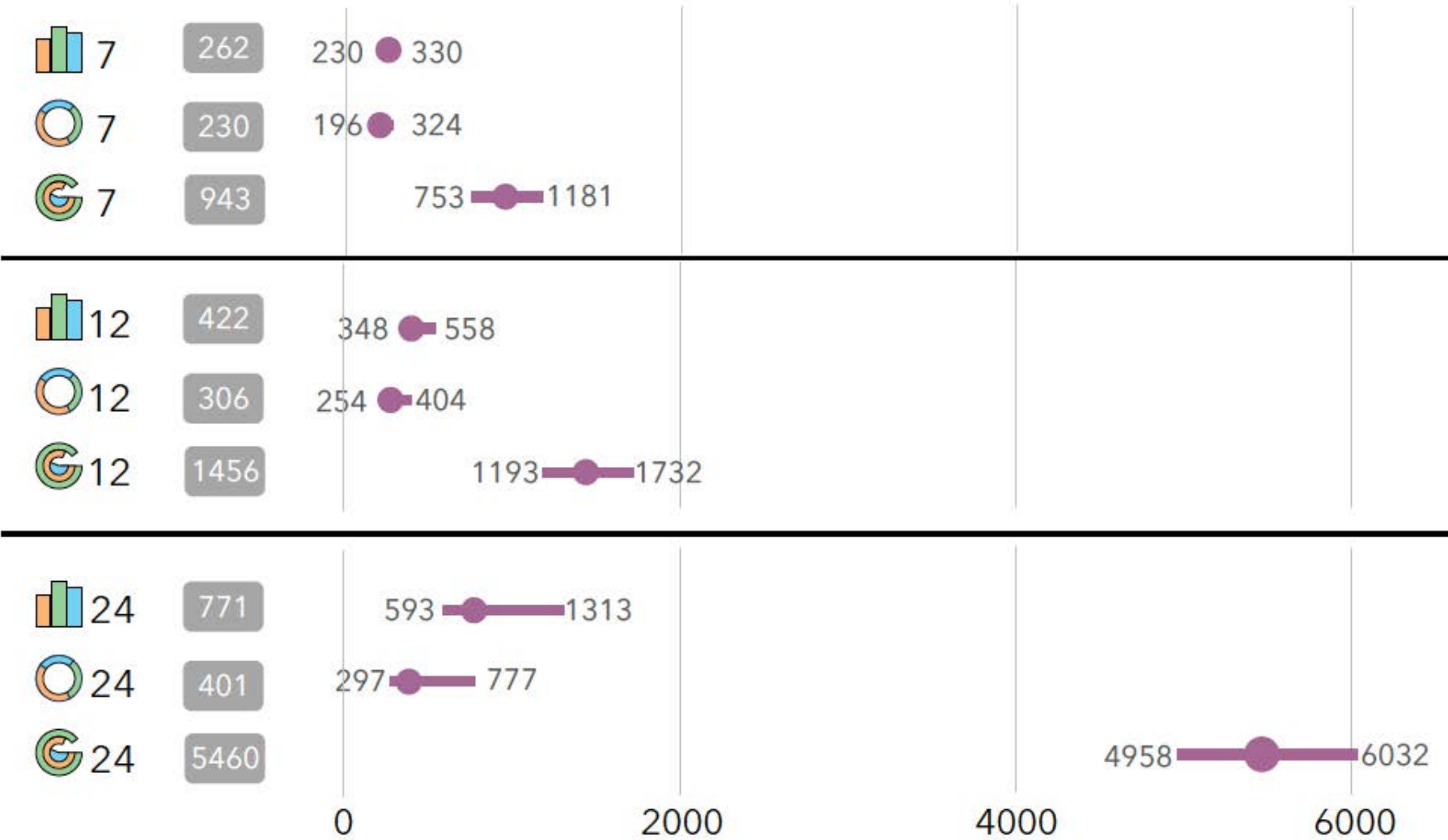
While et al. Glanceable Data Visualizations for Older Adults: Establishing Thresholds and Examining Disparities Between Age Groups. CHI 2024. <https://doi.org/10.1145/3613904.3642776>

# Glanceable Data Visualizations

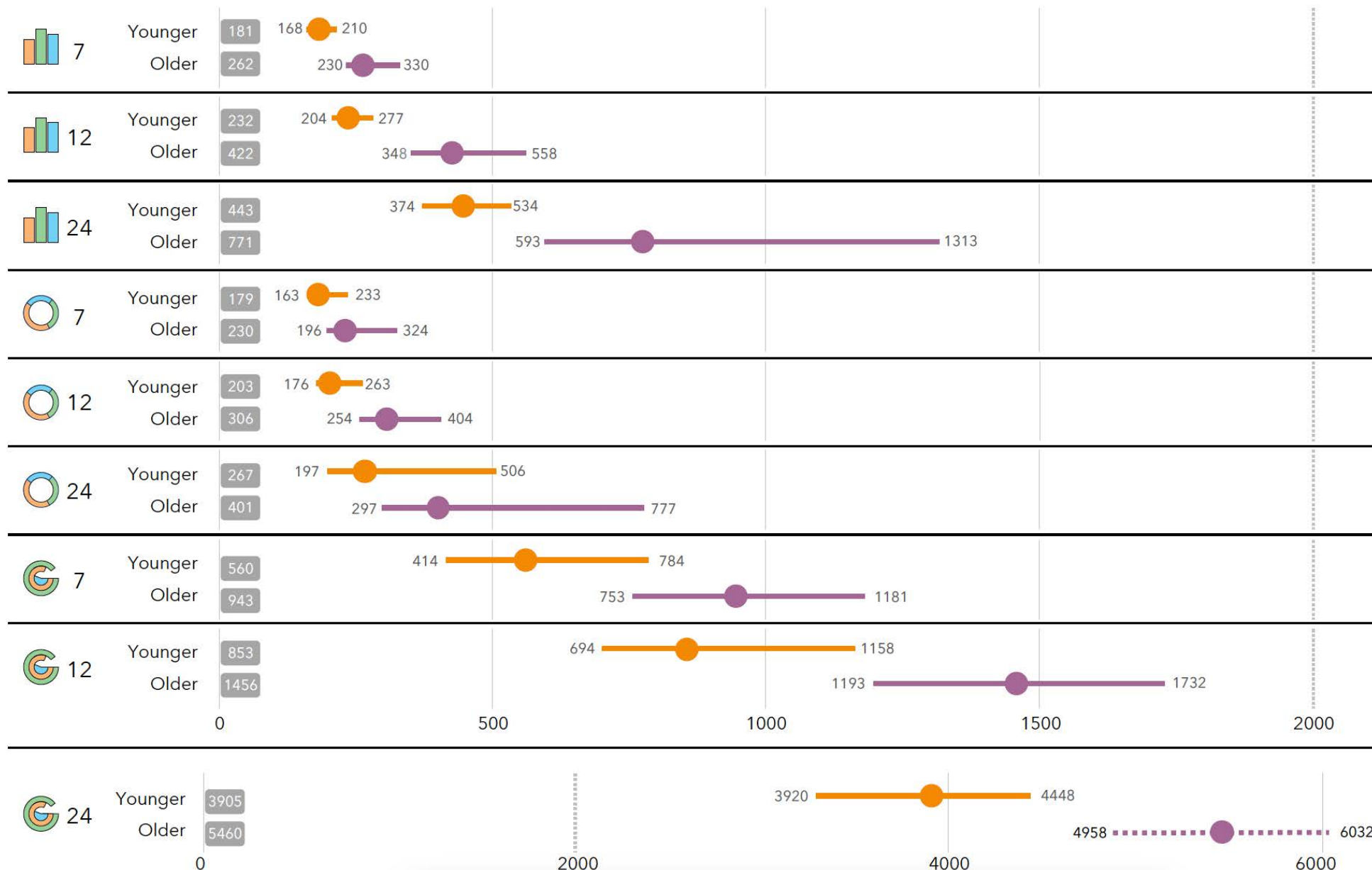


While et al. Glanceable Data Visualizations for Older Adults: Establishing Thresholds and Examining Disparities Between Age Groups. CHI 2024. <https://doi.org/10.1145/3613904.3642776>

# Glanceable Data Visualizations

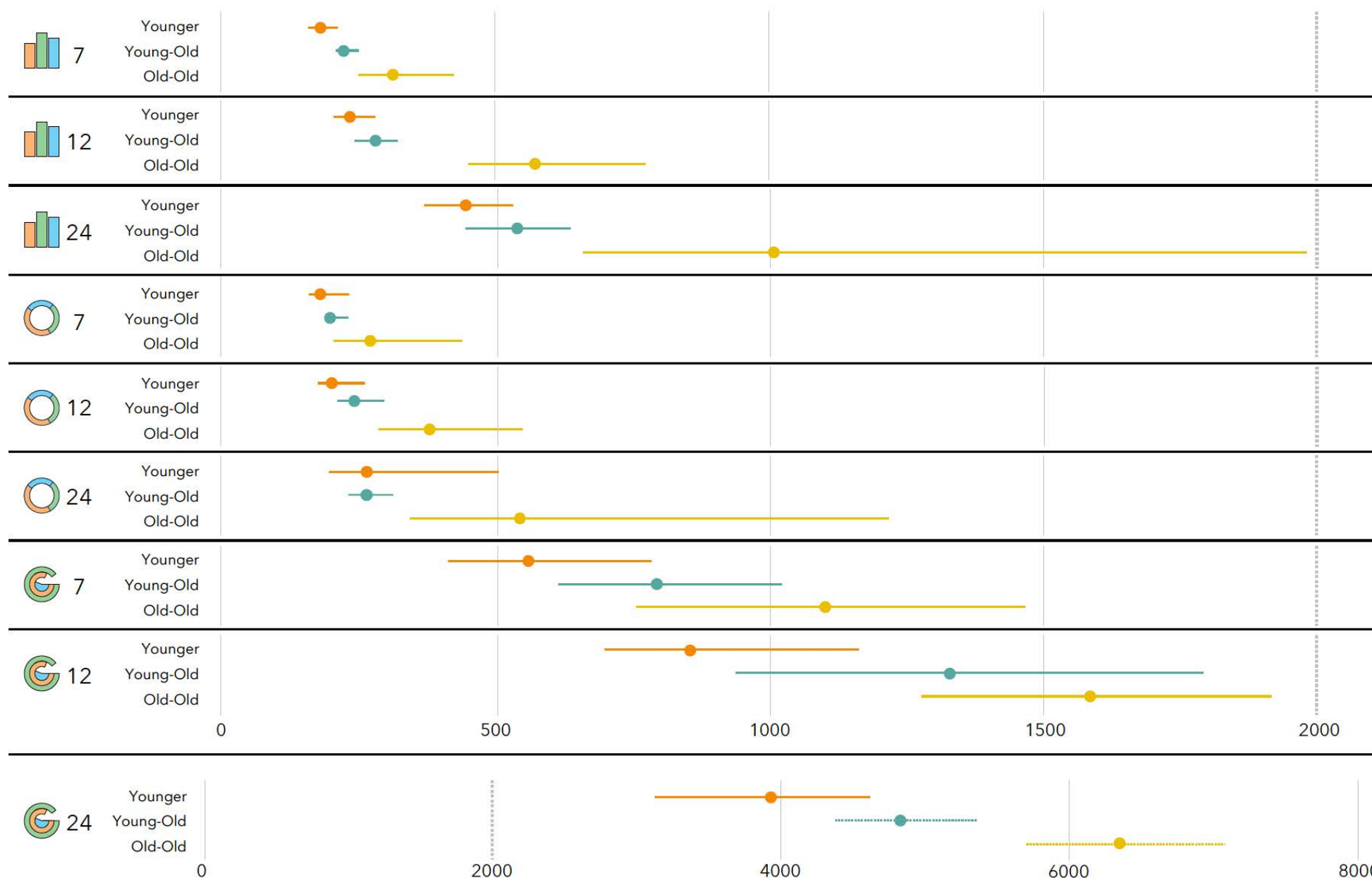


While et al. Glanceable Data Visualizations for Older Adults: Establishing Thresholds and Examining Disparities Between Age Groups. CHI 2024. <https://doi.org/10.1145/3613904.3642776>



age:  
< 65  
≥ 65

While et al. Glanceable Data Visualizations for Older Adults: Establishing Thresholds and Examining Disparities Between Age Groups. CHI 2024. <https://doi.org/10.1145/3613904.3642776>



age:  
 < 65  
 65-74  
 ≥ 75

While et al. Glanceable Data Visualizations for Older Adults: Establishing Thresholds and Examining Disparities Between Age Groups. CHI 2024. <https://doi.org/10.1145/3613904.3642776>

# LAYOUT AND NAVIGATION FOR SMALL DISPLAYS



# Content on Small Displays

- Limited amount of space
  - Distribute content across multiple screens
  - Support navigation and search to access content
  - Establish navigation components and conventions
- Optimize graph of linked content
  - Short navigation path for frequently accessed content
- Aspects of screen design
  - Layout
  - Screen linking and navigation patterns
  - Typography
  - Writing style

# Layout Guidelines

- Avoid startup splash screens, banners, unmotivated content
  - Leads to delay, additional perceptual load
- Provide a visual hierarchy
  - More important items are larger, higher, brighter
  - Position, size, shape, contrast, color, form
- Use the Gestalt perception capabilities
  - Closure, continuity, figure and ground, proximity, similarity, symmetry
  - Example: Group related items, use whitespace between them
- Provide landmarks for orientation
  - Persistent and consistent structure of system and apps, e.g., system bars

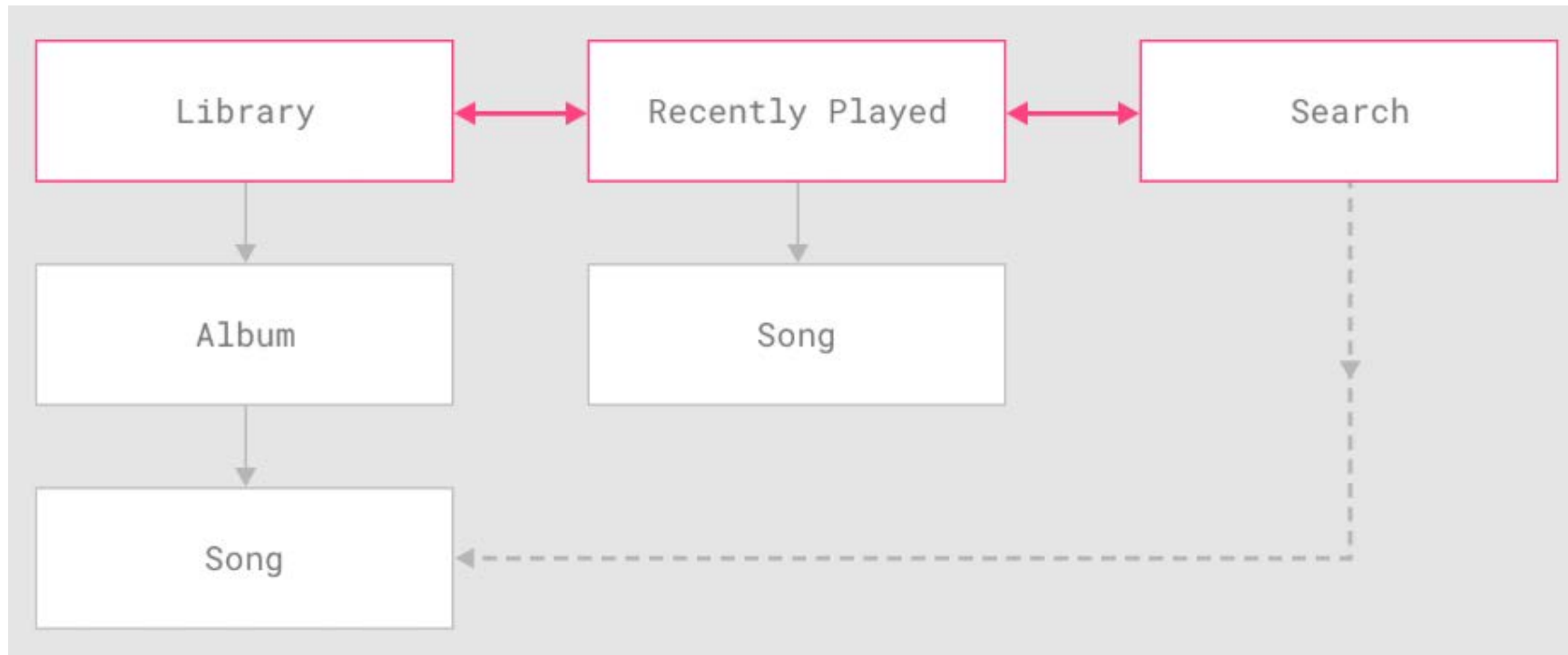
# Supporting Navigation and Orientation

- Navigation: Moving between screens of an app
  - How to get there?
- Orientation: Avoid users being lost in content hierarchy
  - Where am I?
- Give spatial cues
  - Current position in app / on page?
  - Where is goal of my activity?
  - Next step towards goal?
  - Has goal been achieved?
  - How to go back / exit?
- Use real-world spatial knowledge and skills
  - Paths, edges, nodes, landmarks, districts

# Navigation Support in Android

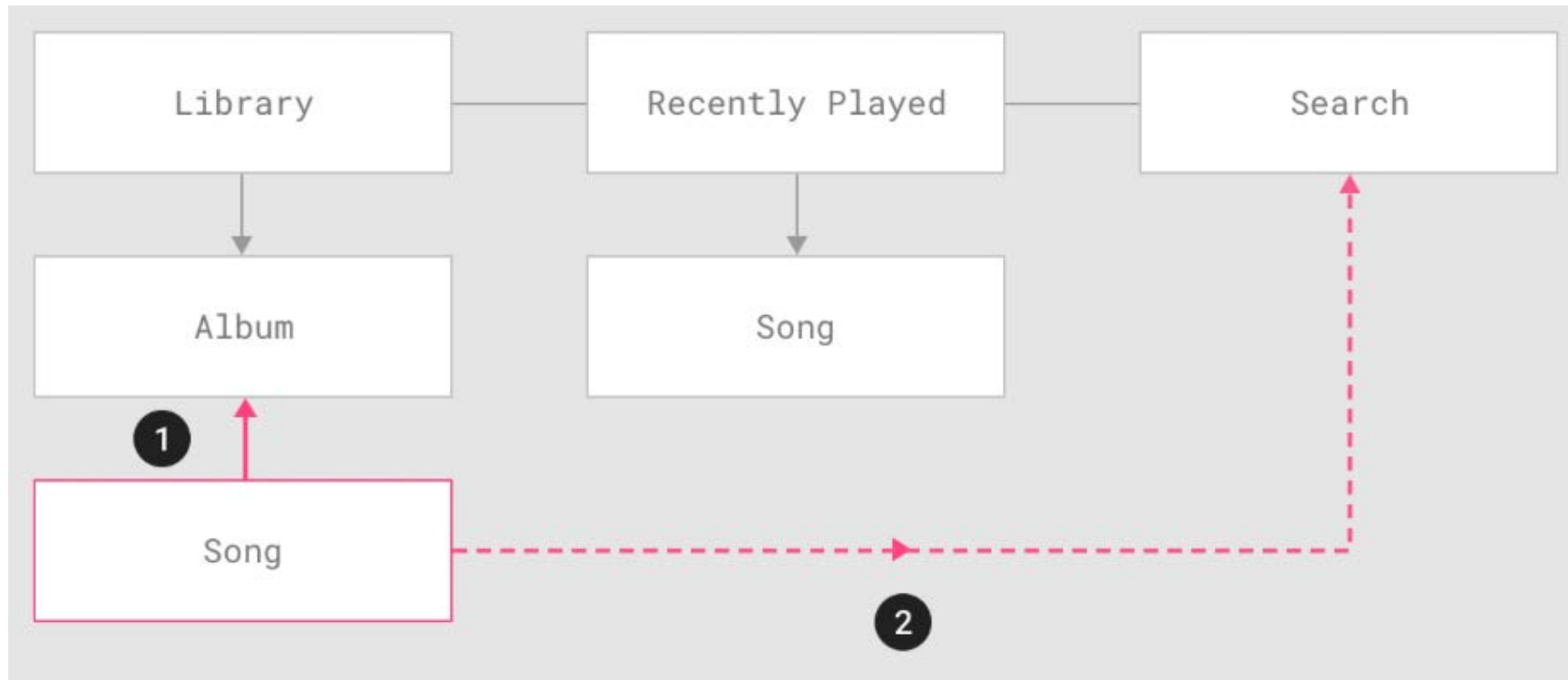
- Android provides GUI elements and conventions that support navigation and orientation
- Restore state of visited destinations to support orientation, task resumption, and recall
- Clearly indicate the navigation hierarchy
- Navigation directions
  - **Lateral:** Moving between destinations at the same level, initial state of app should provide access to all destinations at top level
  - **Forward:** Moving downward in an app's hierarchy, forward in a task, forward across apps
  - **Reverse:** Moving back hierarchically or chronologically

# Navigation Directions: Lateral





# Navigation Directions: Reverse

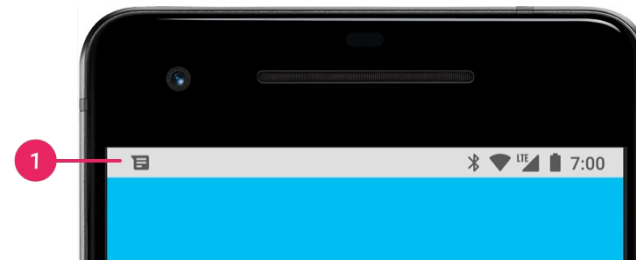


# Navigation Support in Android: System Bars

Persistent, available throughout the system

## 1. Status Bar

- Pending notifications
- Status (time, battery level, signal strength)
- Bezel-swipe down for notifications, details



## 2. Navigation Bar

- In lieu of hardware keys
- Controls: Back, Home, Recent Apps

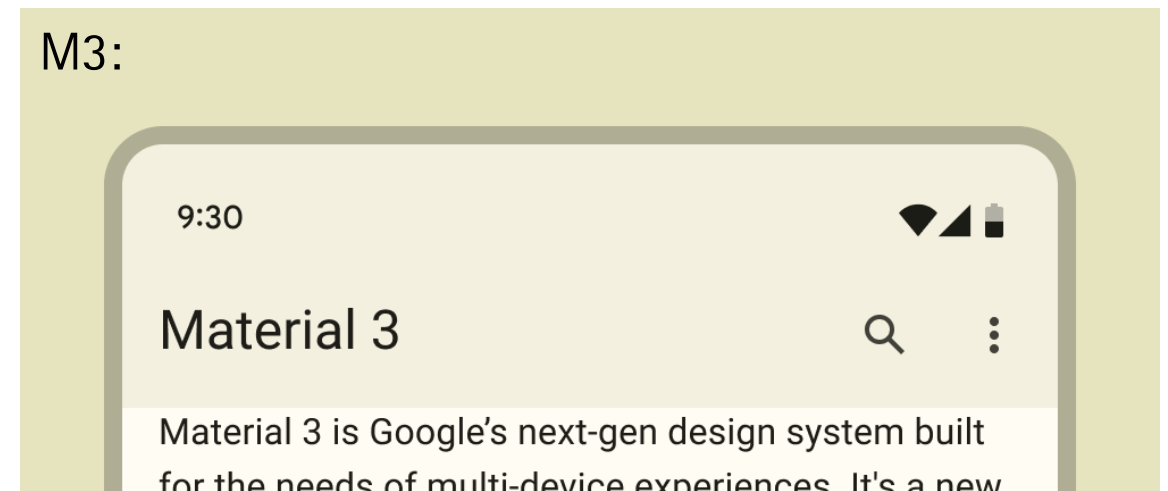
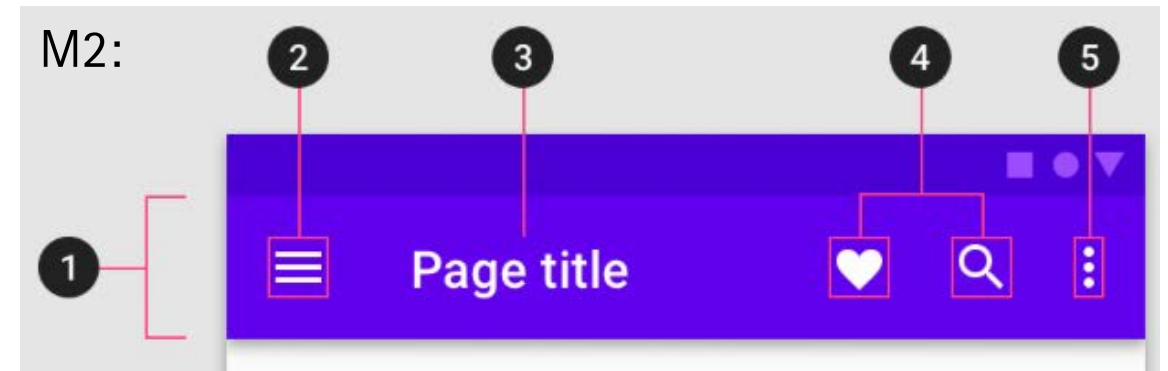


<https://developer.android.com/design/handhelds/>



# Navigation Support in Android: Top App Bar

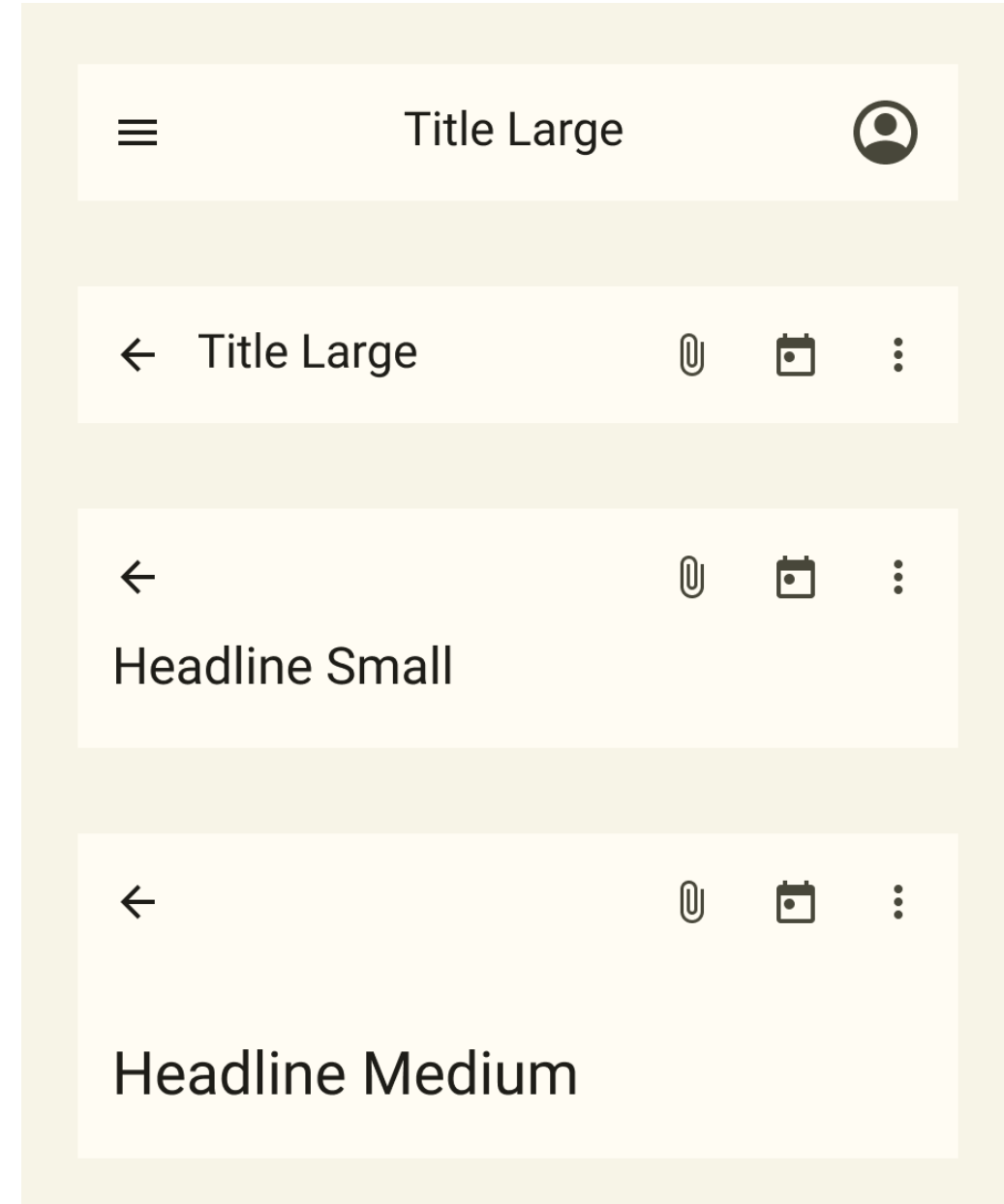
- Information and actions related to current screen
- Available throughout the app: consistence, guidance
- Contextual actions bars: actions for selected items
  1. Container (default height)
  2. Navigation icon for view switching
  3. Headline to give the screen an identity
  4. Action items for most important actions
  5. Overflow menu for rarely used actions



<https://material.io/design/components/app-bars-top.html>  
<https://m3.material.io/components/top-app-bar/overview>

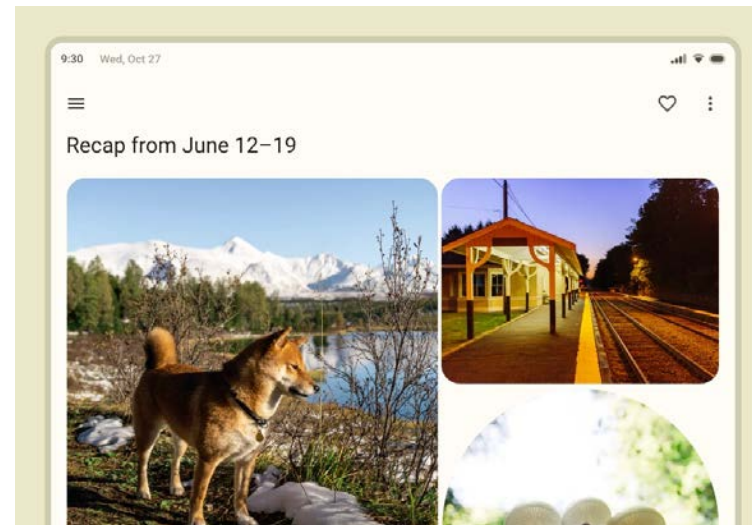
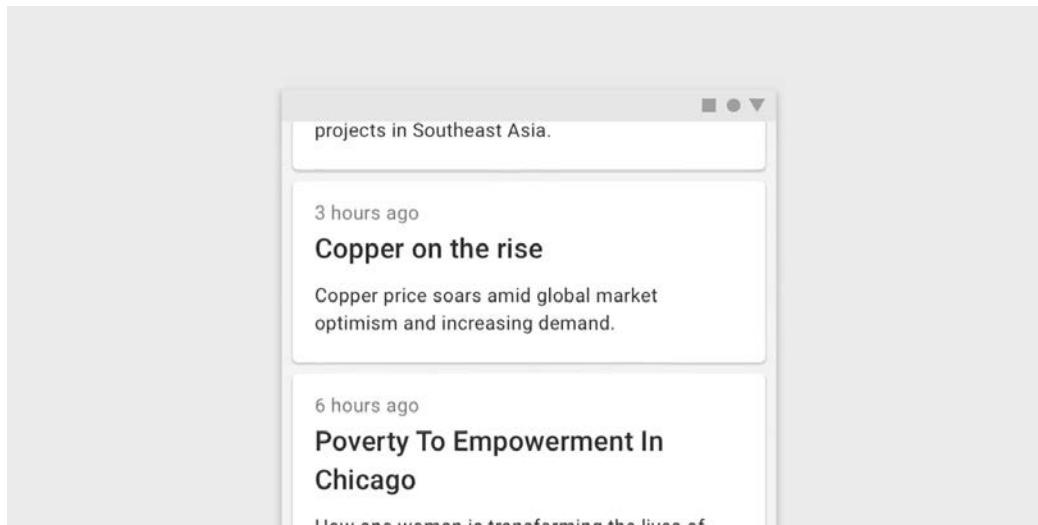
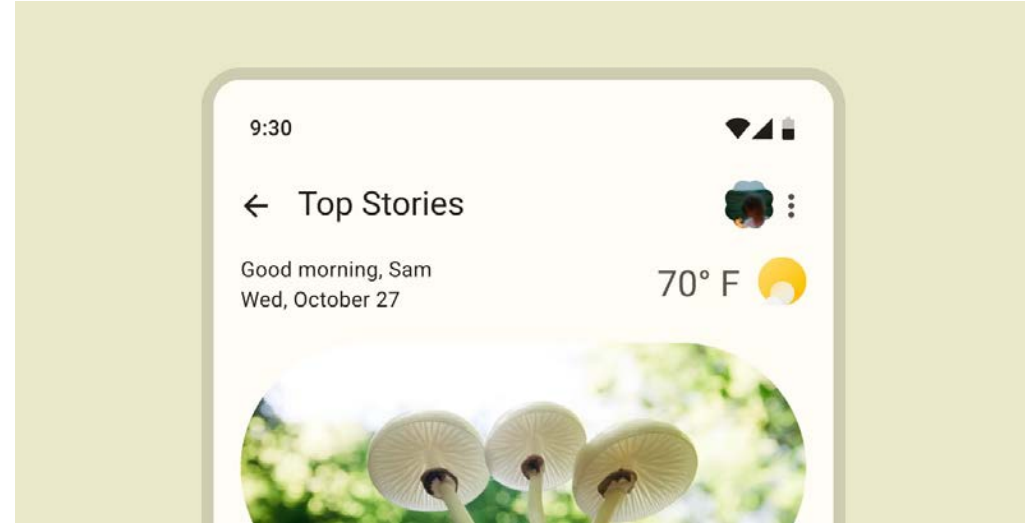
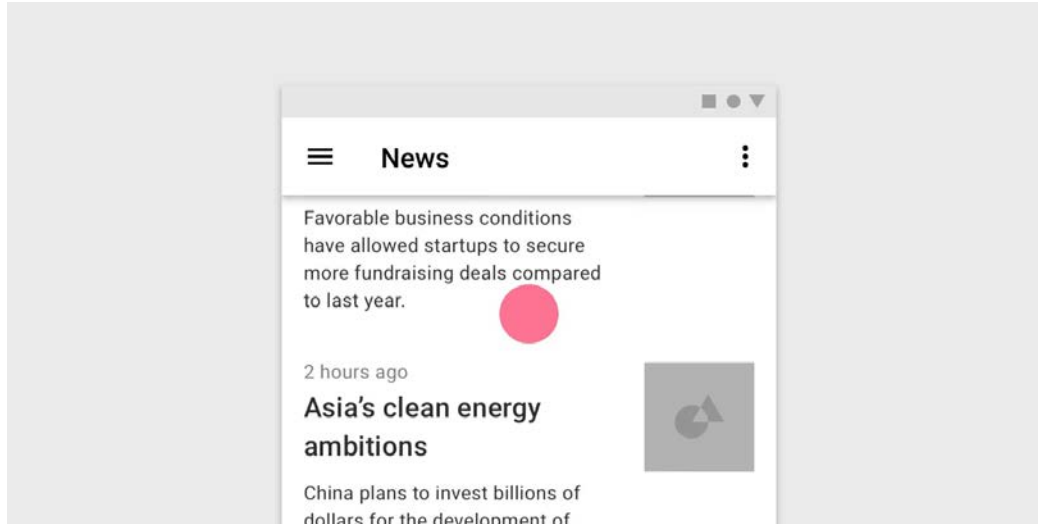
# Navigation Support in Android: Top App Bar

- Information and actions related to current screen
- Available throughout the app: consistence, guidance
- Contextual actions bars: actions for selected items
  1. Container (default height)
  2. Navigation icon for view switching
  3. Headline to give the screen an identity
  4. Action items for most important actions
  5. Overflow menu for rarely used actions



<https://material.io/design/components/app-bars-top.html>  
<https://m3.material.io/components/top-app-bar/overview>

# Top App Bar Behavior: Remain or Scroll



<https://material.io/design/components/app-bars-top.html>  
<https://m3.material.io/components/top-app-bar/overview>

# Action Buttons

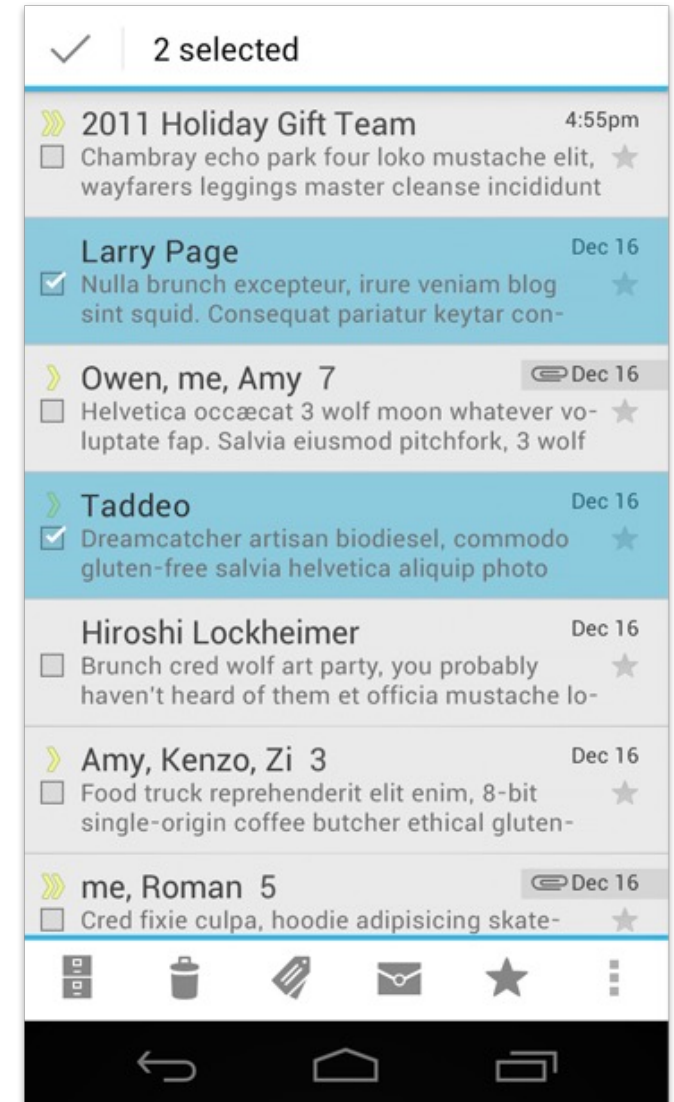
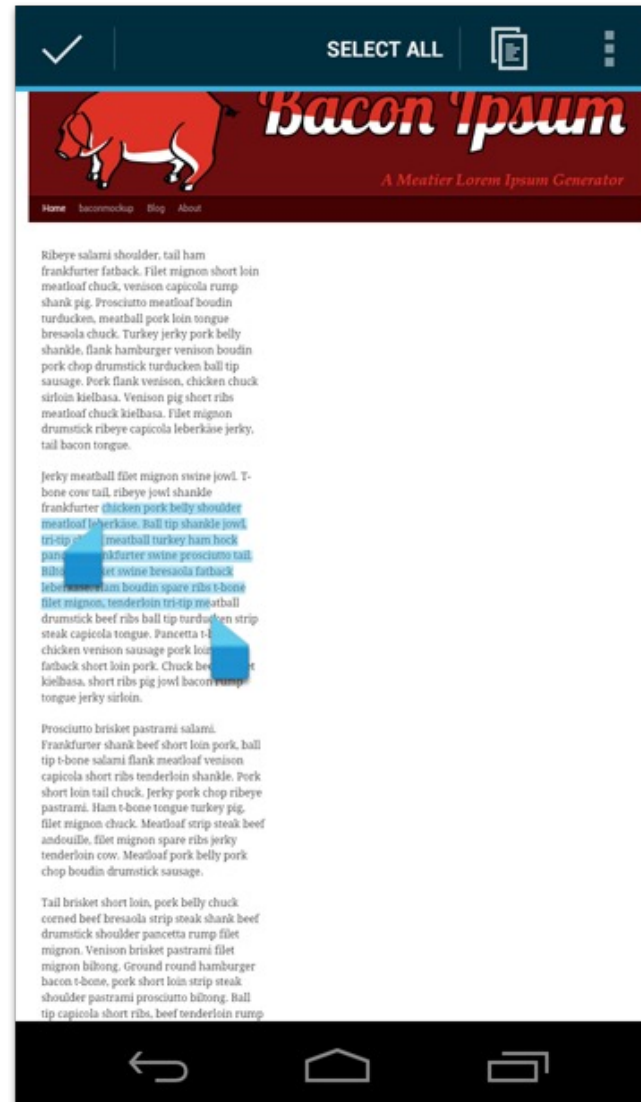
- Action buttons linked to app's most important activities
  - Order by frequency of use
  - Minimize the number of interaction steps
  - Hide unavailable actions
- Prioritizing actions
  - "FIT scheme": Move to action bar if F or I or T; otherwise overflow menu
  - Frequent: Action used  $\geq 70\%$  of the time, or several times in a row
  - Important: Action should be discovered ("selling point")
  - Typical: First-class action in similar apps
- Predefined icons for common actions



<http://developer.android.com/design/patterns/actionbar.html>

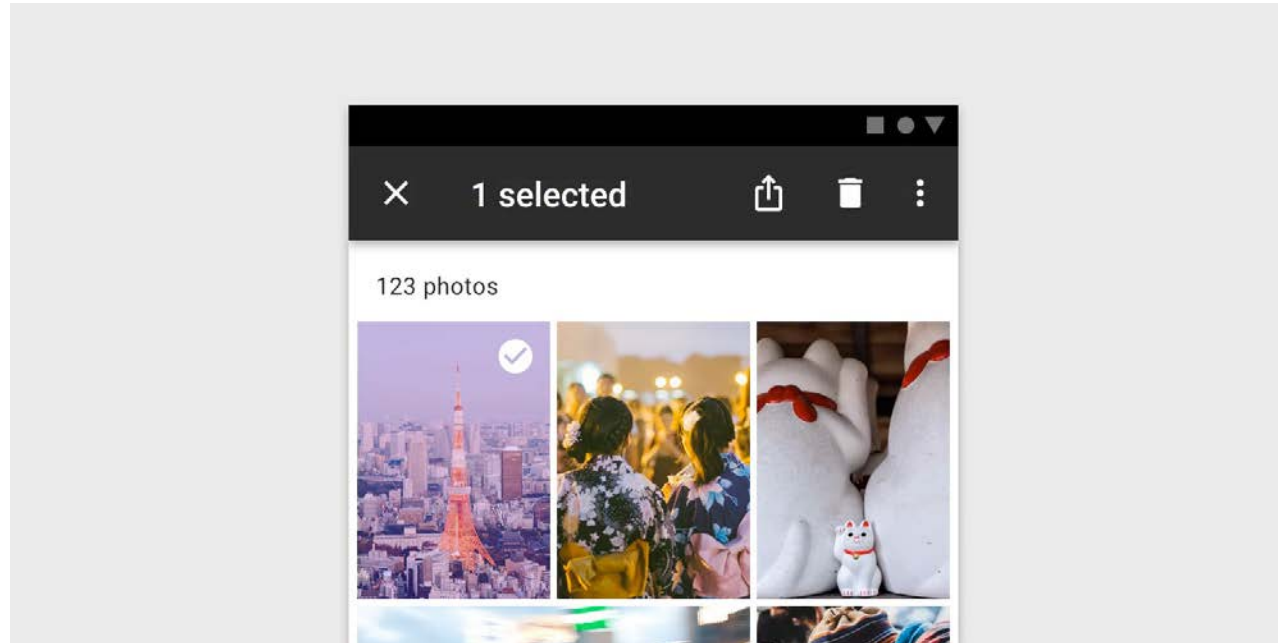
# Contextual Action Bar

- Temporary action bar
- Overlays app's action bar during a sub-task
- For acting on selected data
- Dismiss with ✓ or ↶



<http://developer.android.com/design/patterns/actionbar.html>

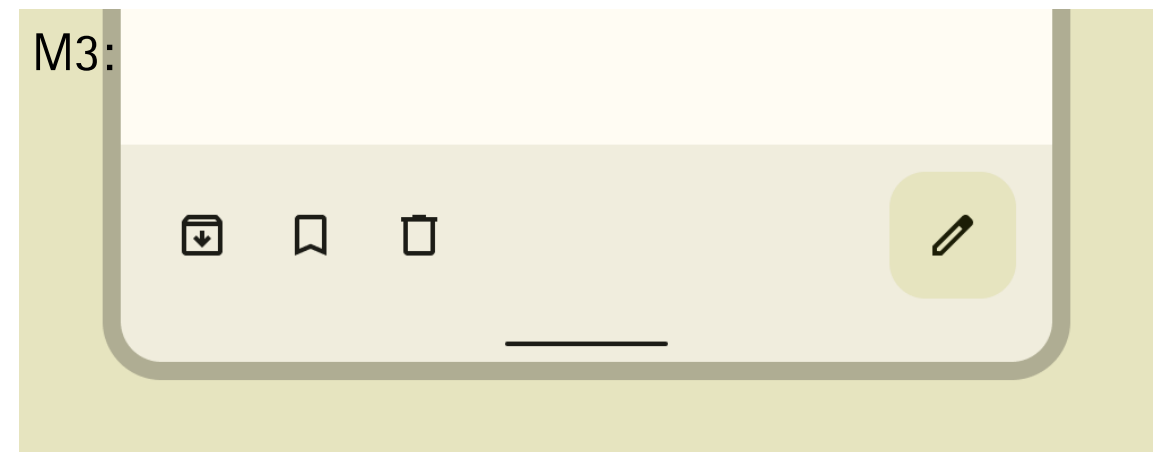
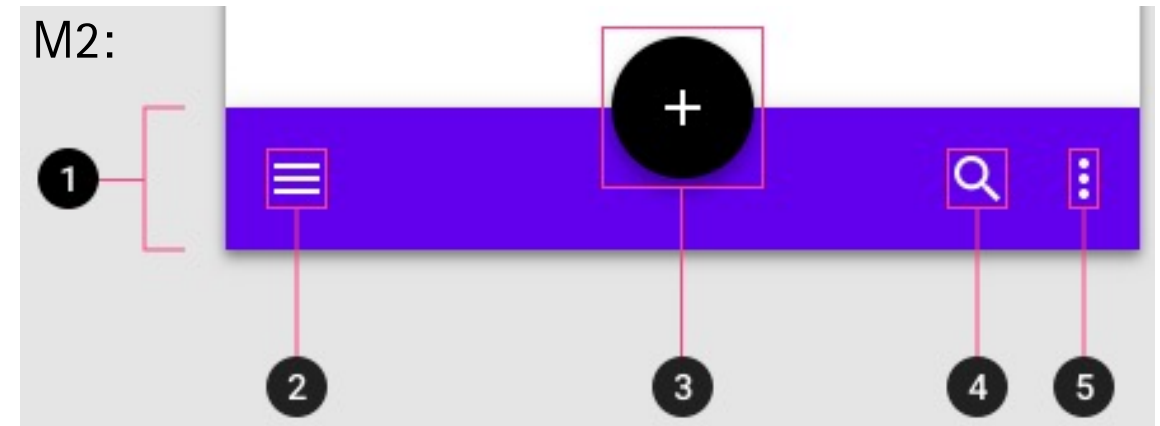
# Top App Bar Behavior: Contextual Action Bar



1. Close button
2. Contextual title (e.g., number of selected items)
3. Action buttons for contextual actions
4. Overflow menu for rarely used contextual actions

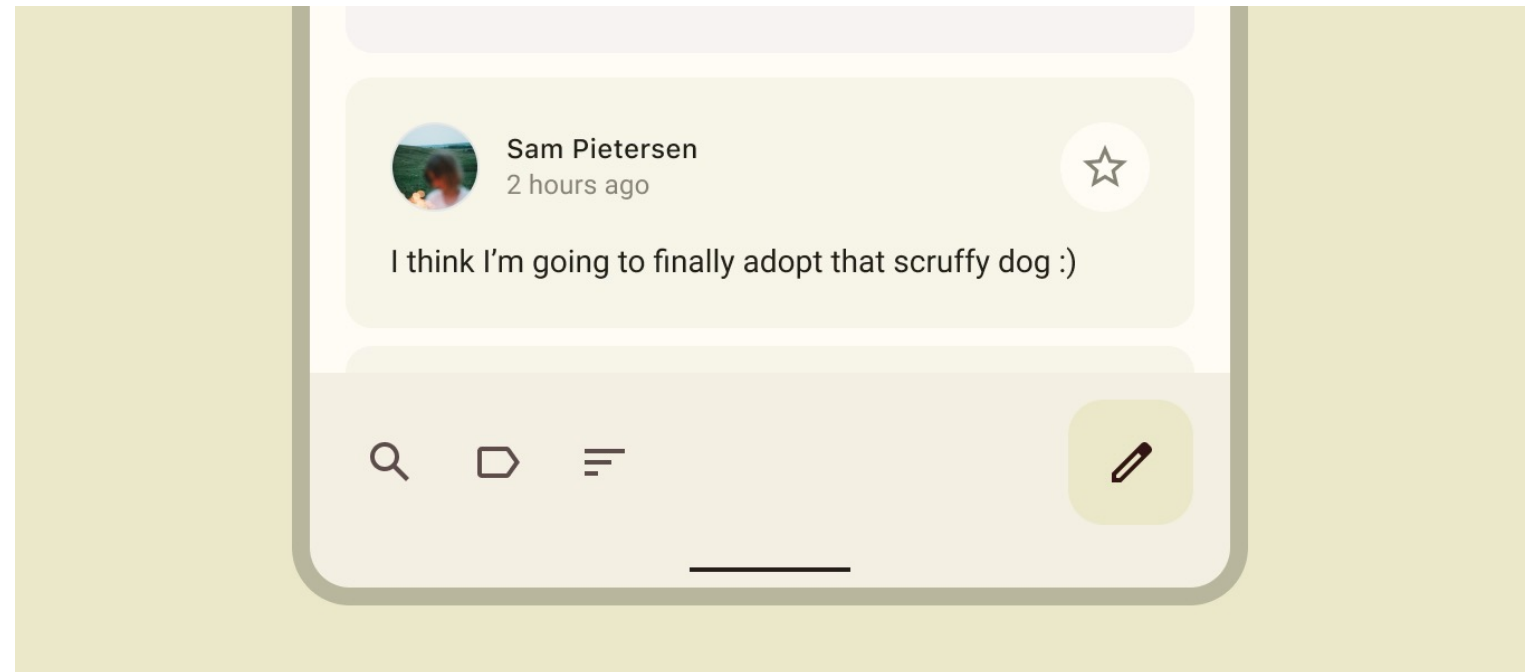
# Navigation Support in Android: Bottom App Bar

- Important actions related to current screen
- Floating action button for primary action
- Ergonomically easy to reach for thumb, convenient access
  1. Bottom app bar
  2. Navigation icon for view switching
  3. Floating action button (FAB)
  4. Action buttons for most important actions or destinations
  5. Overflow menu for rarely used actions (M2)



# Navigation Support in Android: Bottom App Bar

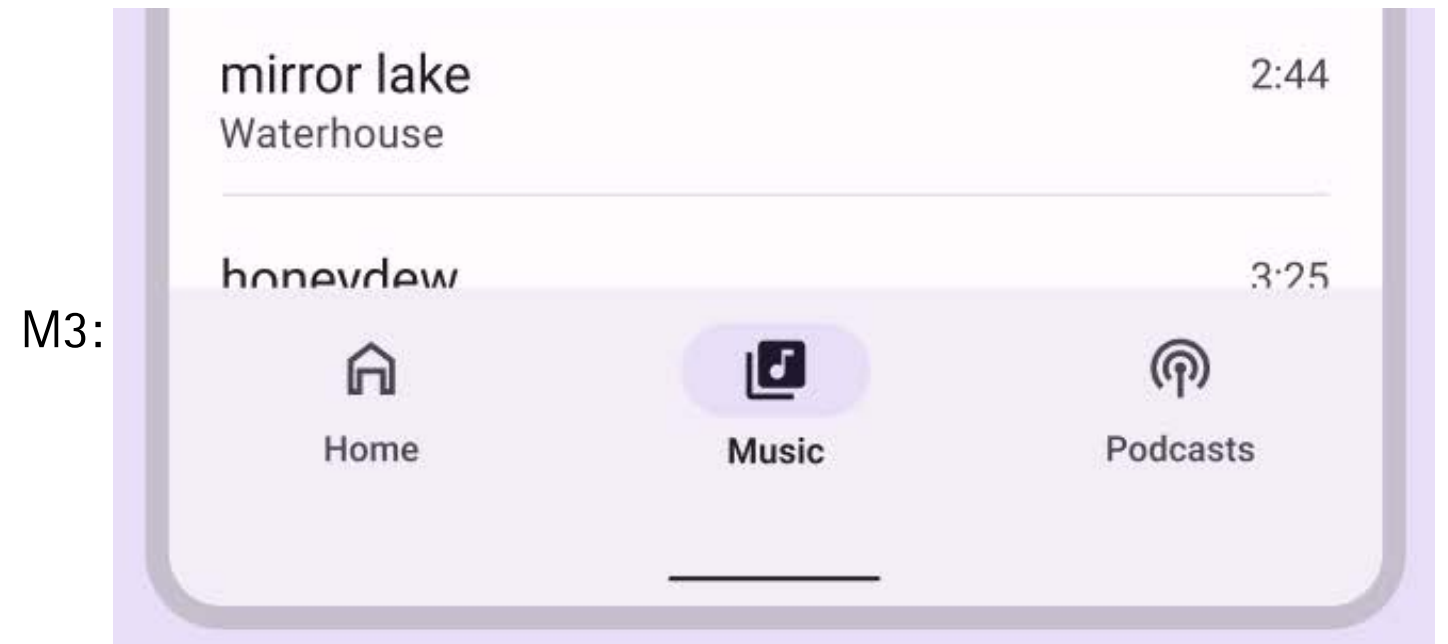
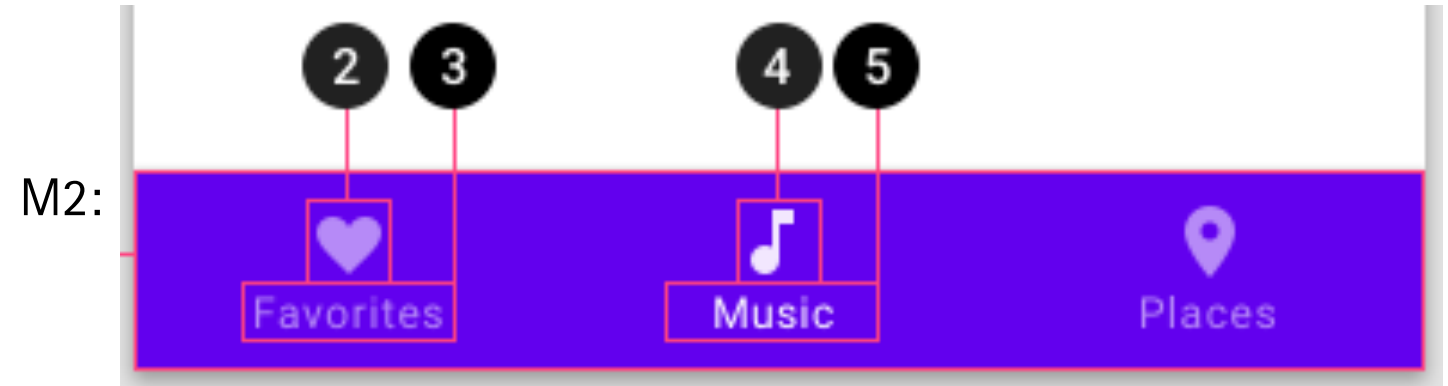
- Destructive actions should provide undo snackbar
- Actions should be divided consistently across top and bottom app bars
  - top: navigation, overflow menu
  - actions like search in a consistent location





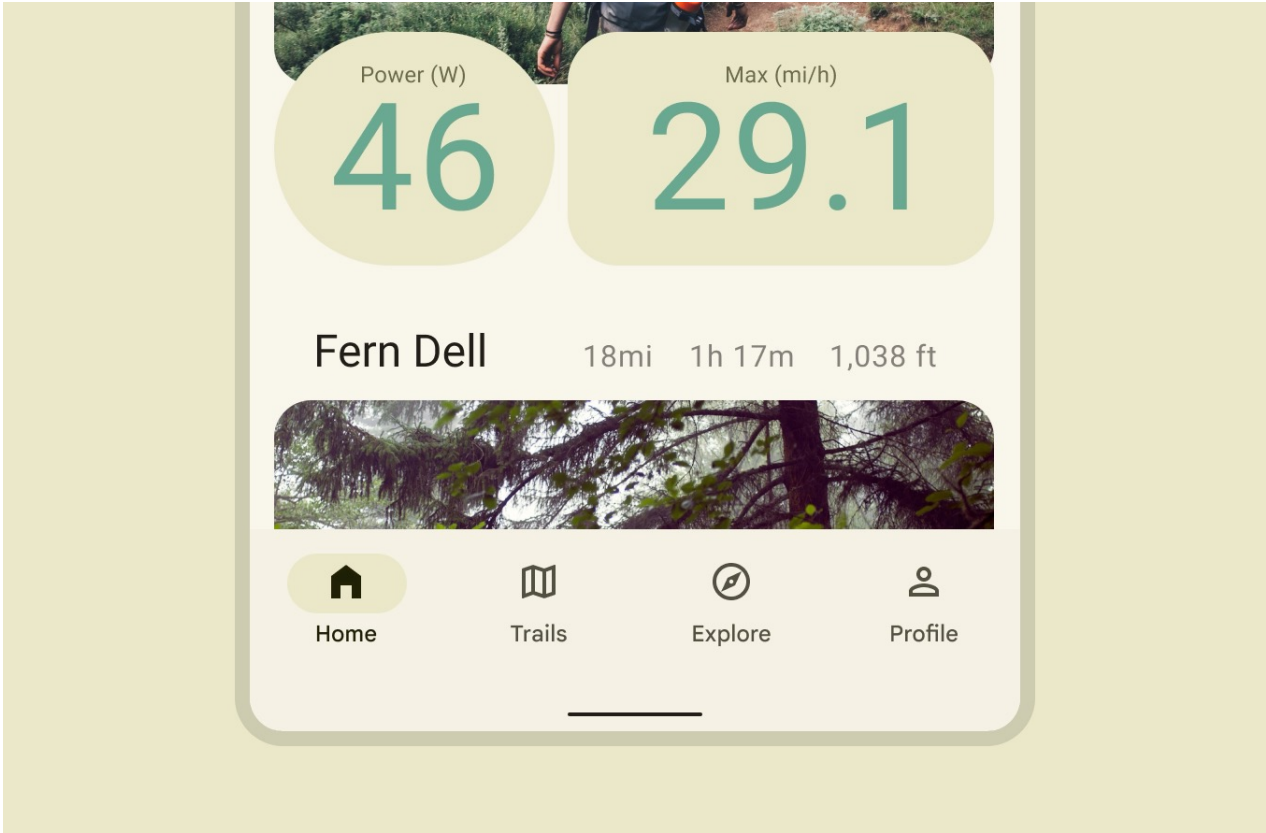
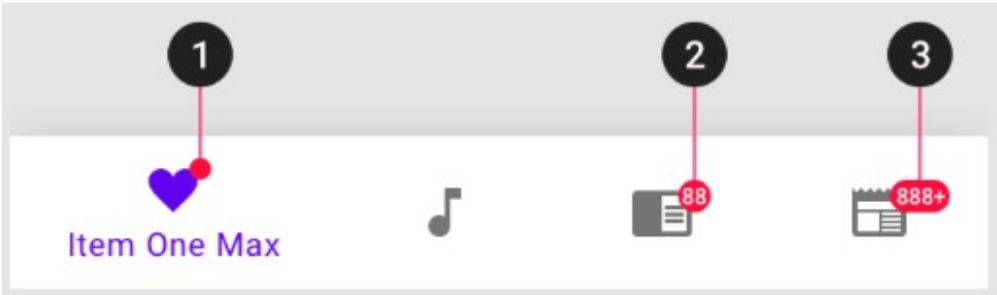
# Navigation Support in Android: Navigation Bar

- Quick navigation to 3-5 destinations within the app
  - Top-level destinations
- Consistently available throughout the app
- Ergonomically easy to reach
  1. Bottom navigation bar
  2. Inactive icon
  3. Inactive text label
  4. Active icon
  5. Active text label



# Navigation Bar Behavior

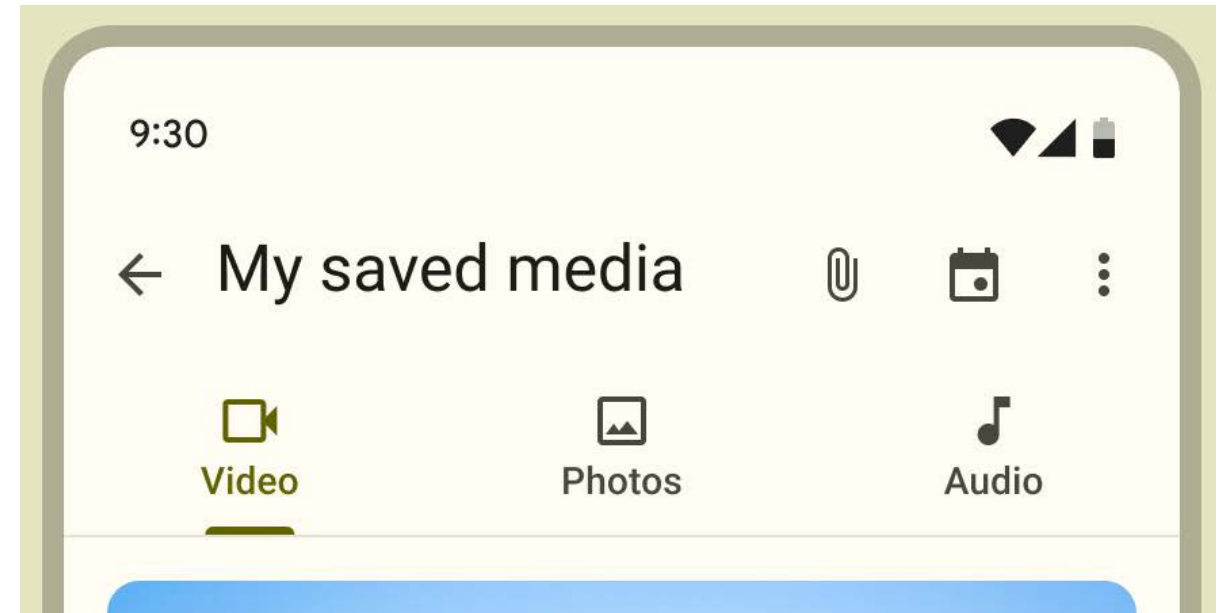
- Tapping a visited destination: restore previous scroll position or reset scroll position
- Tapping the current destination: scroll to top, refresh
- Back button does not navigate between destinations
- Navigation icons may show state



<https://material.io/design/components/app-bars-top.html>  
<https://m3.material.io/components/navigation-bar/overview>

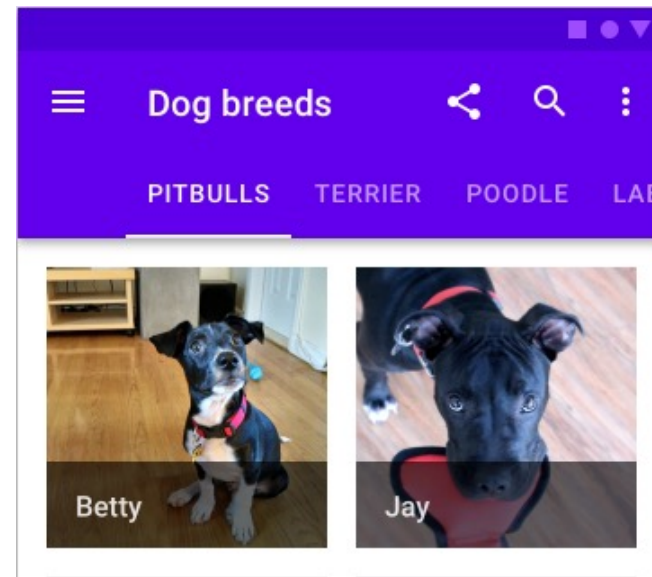
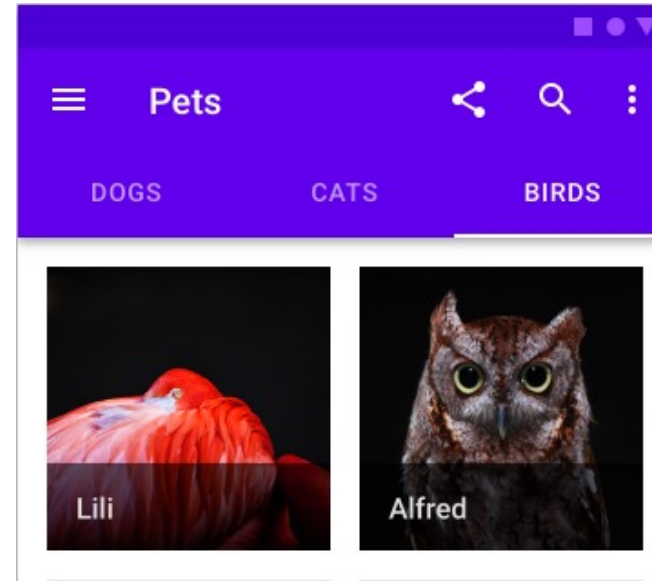
# Navigation Support in Android: Tabs

- Navigate between groups of content
  - Genres of music, kinds of documents, etc.
- Each tab at same level of hierarchy, same importance
- Short tab labels, optional icons
- Visibility
  - Underlining indicates active tab
- Affordance
  - Navigate by tapping tab or swiping content



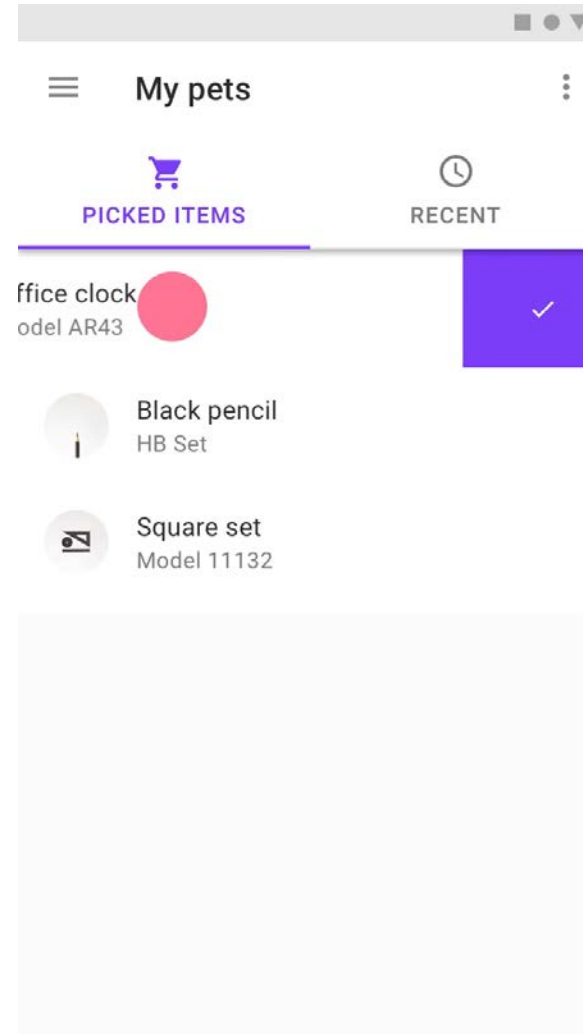
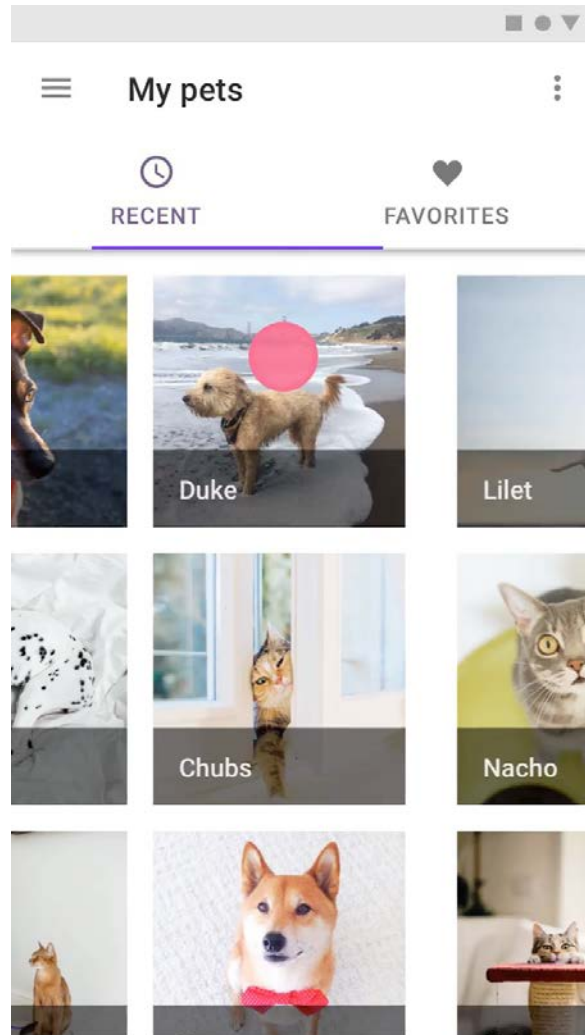
# Tab Types

- Fixed tabs
  - Switching between a (small) fixed number of categories
  - All tabs visible at once
  
- Scrollable tabs
  - Arbitrary number of categories
  - Scrollable, some tabs off-screen
  
- Vertical scrolling
  - Tab bar may be fixed
  - or disappear/reappear



<https://material.io/design/components/tabs.html>

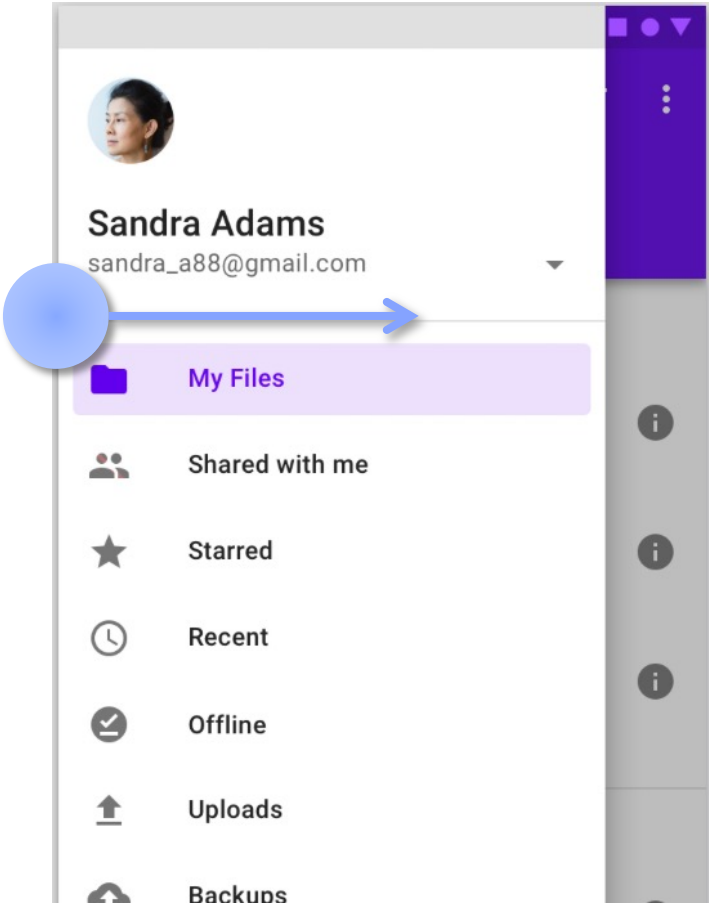
# Tabs and Swiping Horizontally: Pitfalls



<https://material.io/design/components/tabs.html>

# Navigation Drawer

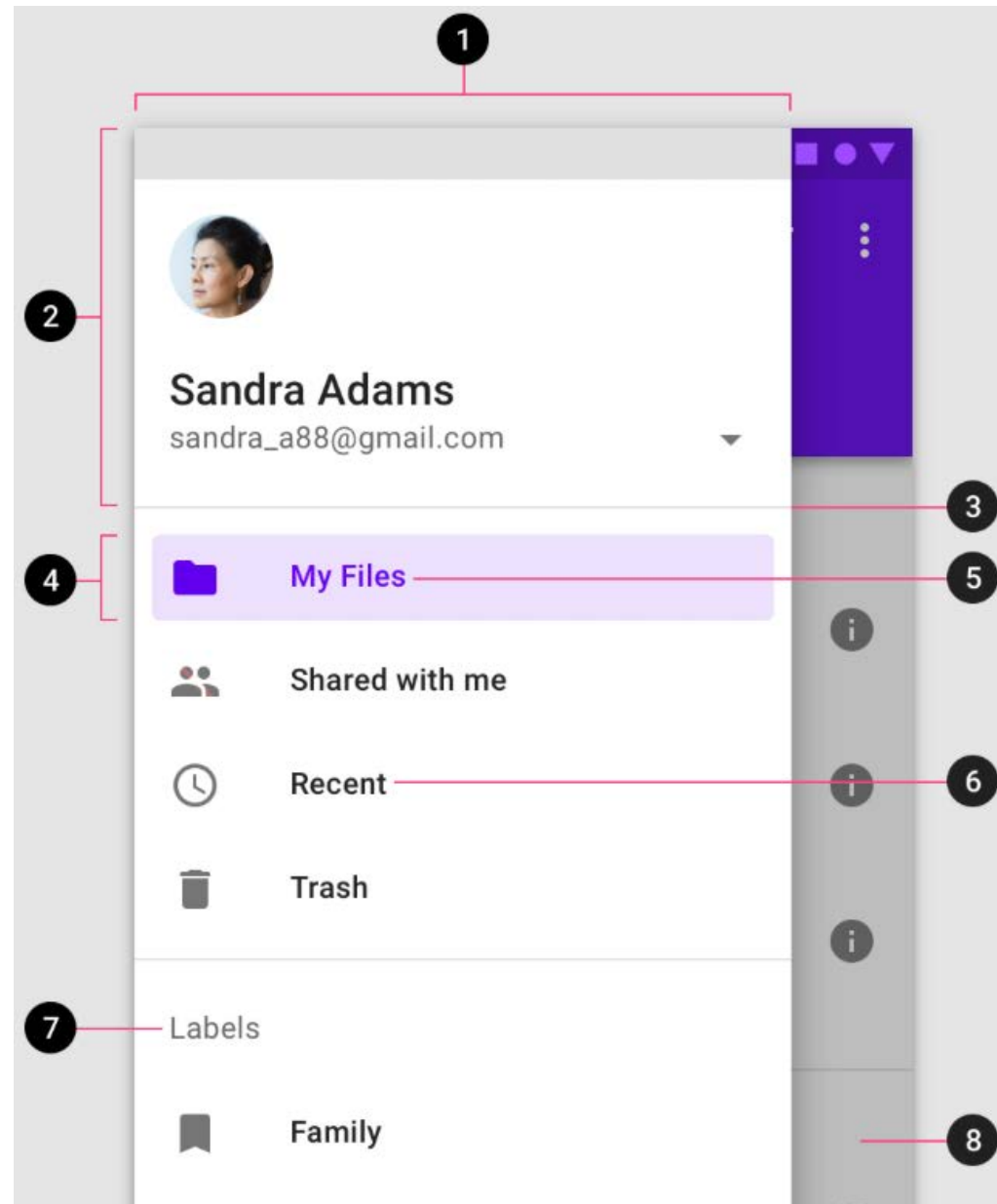
- Menu of all navigation destinations
  - Frequent destinations first
  - Related destinations grouped
- Quick random access to destinations
- Bezel-swipe from left (or bottom) edge or navigation menu icon
- Suitable if
  - Many top-level views
  - Deep navigation hierarchy
- Avoid combining with bottom navigation bar



<https://material.io/design/components/navigation-drawer.html>

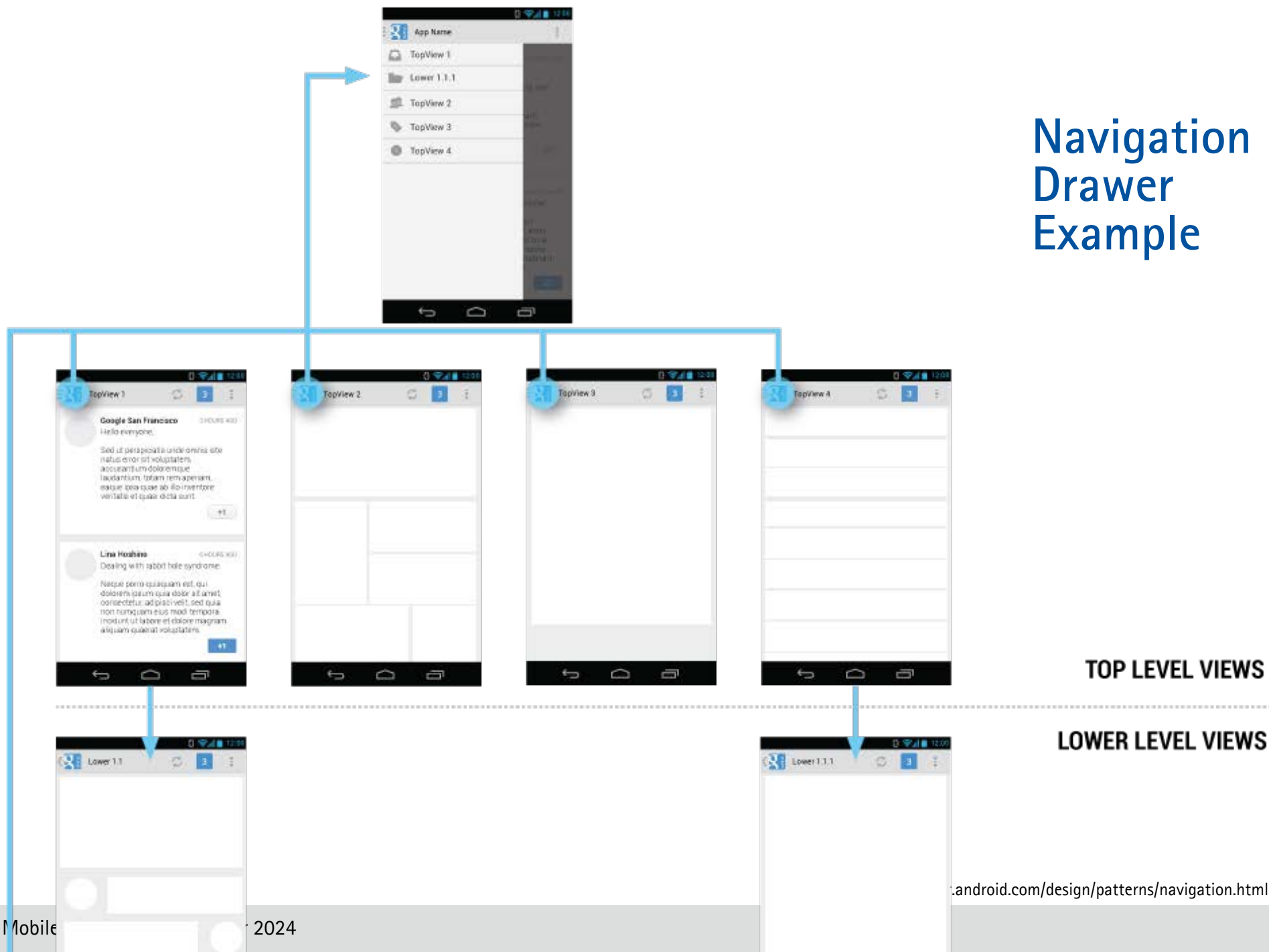
# Navigation Drawer

1. Navigation drawer
2. Header
3. Divider
4. Active text overlay
5. Active text
6. Inactive text
7. Subtitle



<https://material.io/design/components/navigation-drawer.html>

# Navigation Drawer Example

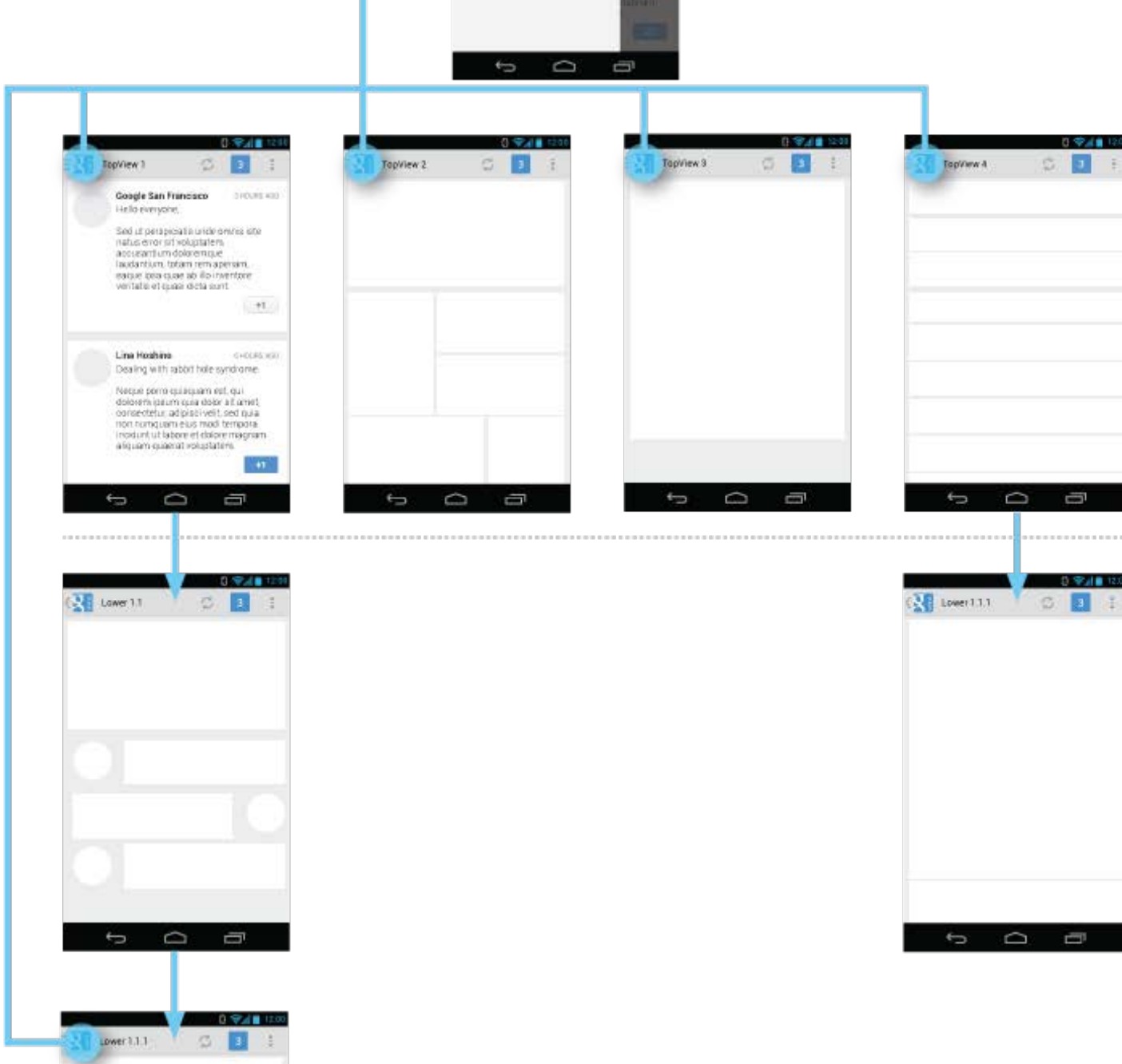




# Navigation Drawer Example

TOP LEVEL VIEWS

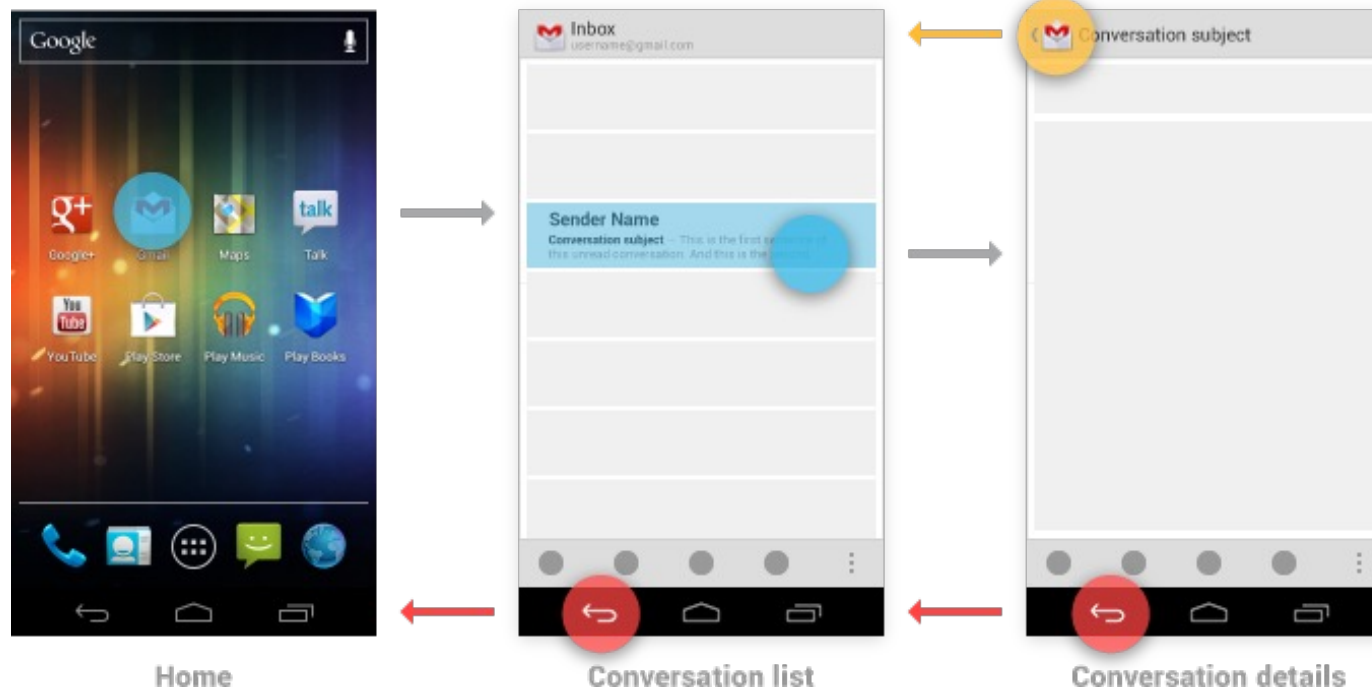
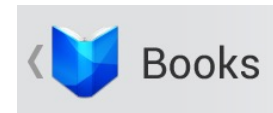
LOWER LEVEL VIEWS



<http://developer.android.com/design/patterns/navigation.html>

# Back and Up Buttons

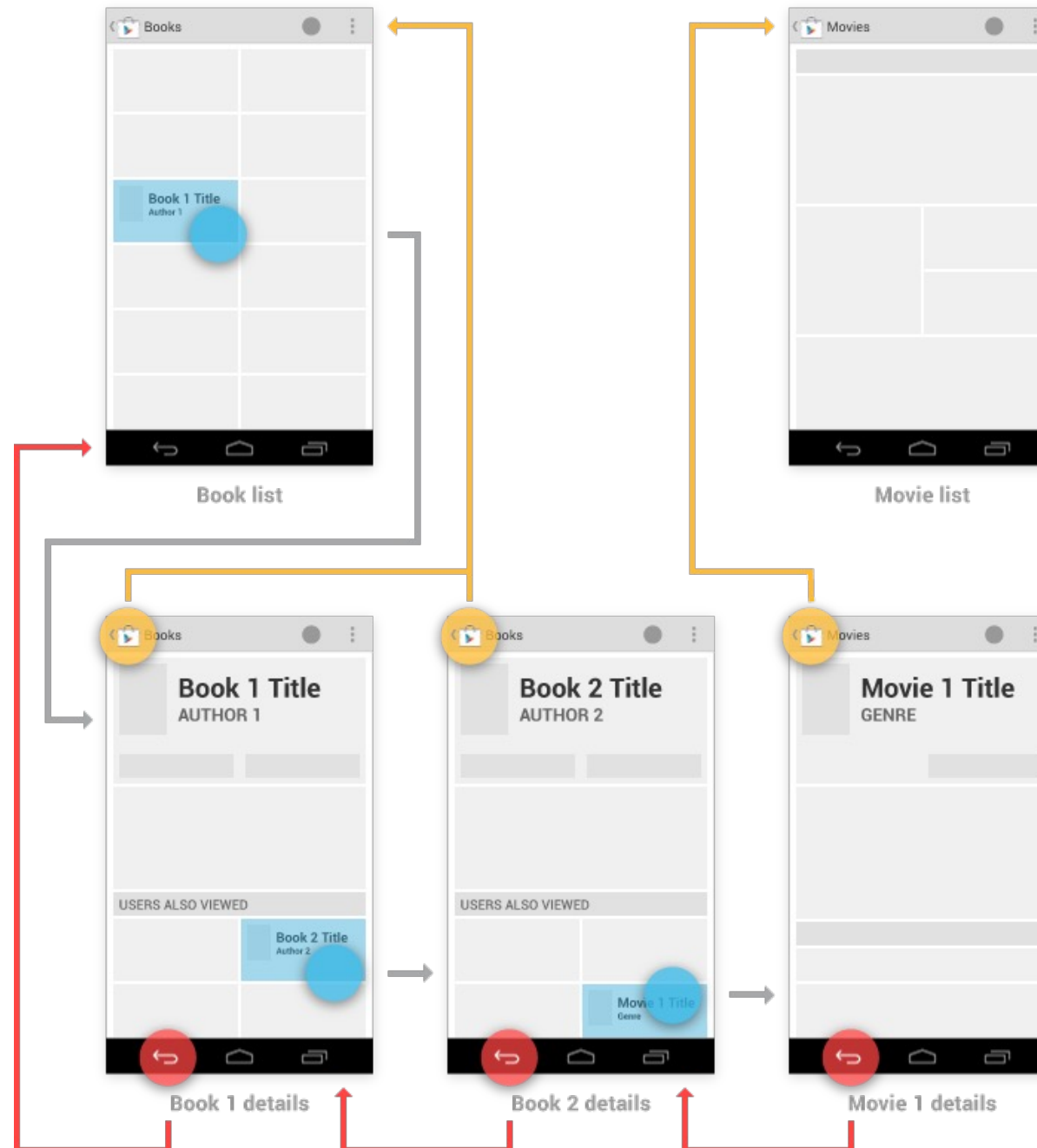
- Back button: Back in chronological order
- Up button: Up in app hierarchy
  - Cannot "fall off" the app using Up



<http://developer.android.com/design/patterns/navigation.html>

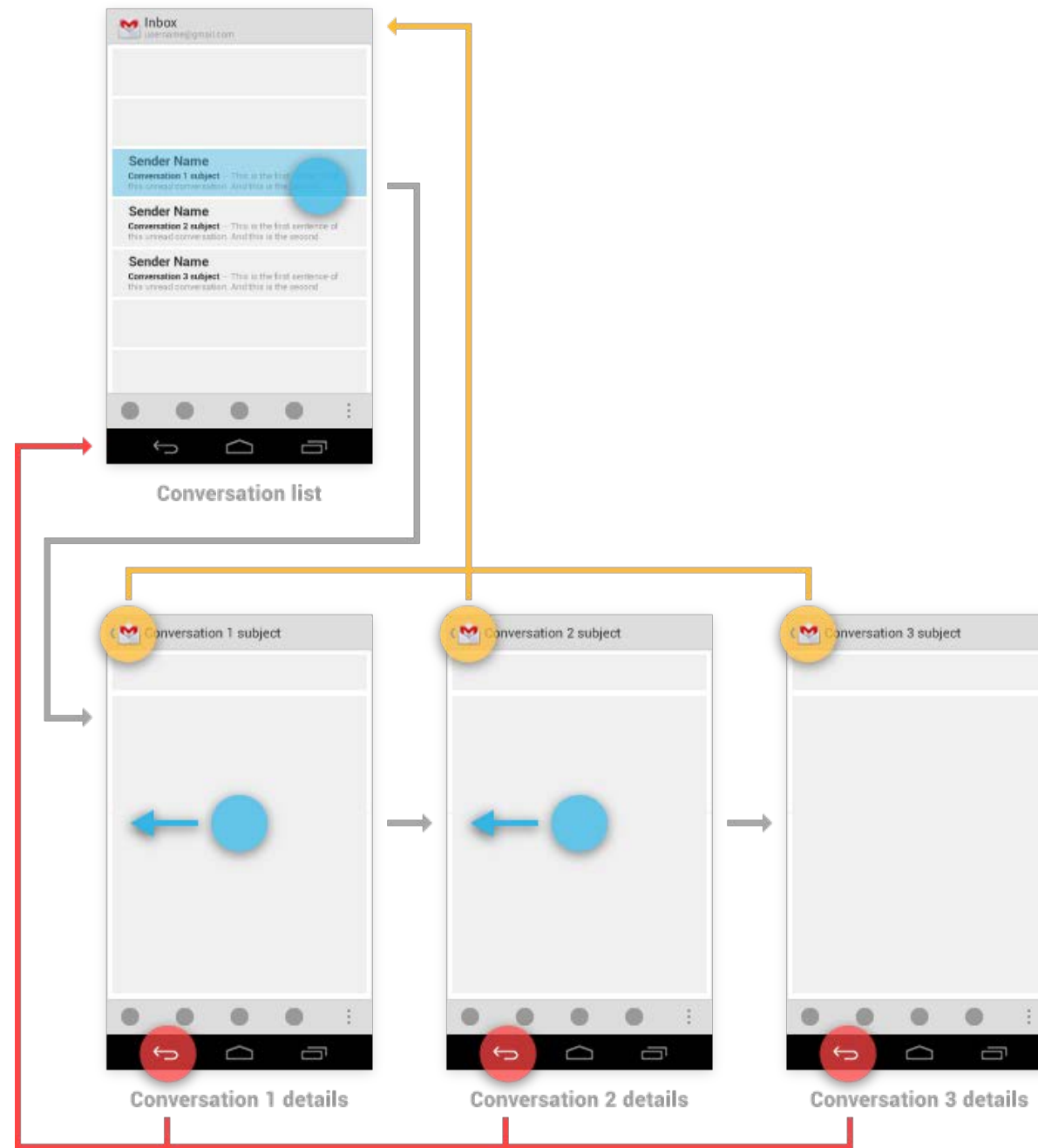
# Back and Up

- Back: Back in chronological order
- Up: Up in app hierarchy



# Back and Up

- Horizontal swiping does not add to the Back history



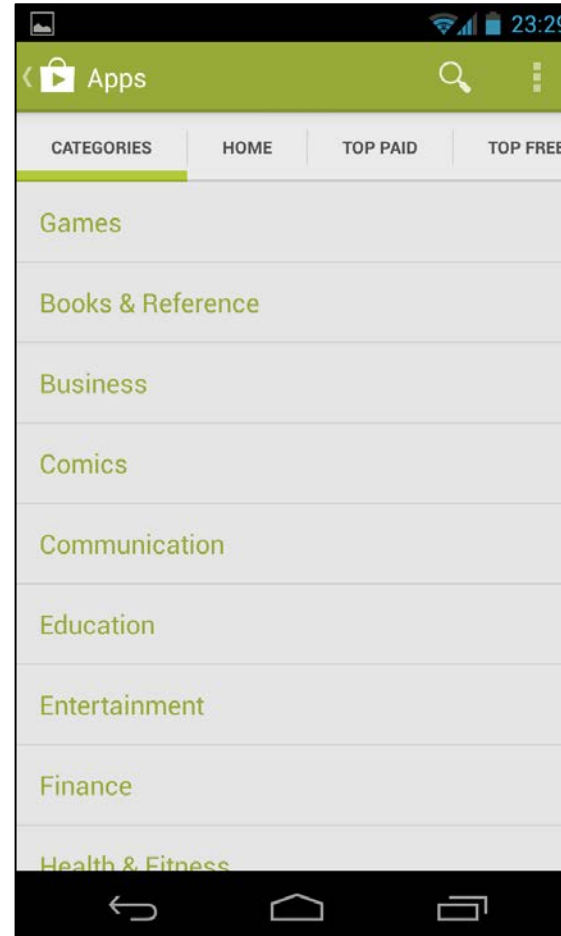
# Implementing Navigation in Android

- Implementing navigation
  - Lateral: navigation components
  - Forward: content (overview to detail, links), buttons, search
  - Reverse: back button, upward-arrow
- Navigation components support navigation in different ways

Component	Use for	Number of destinations
navigation drawer	top-level destinations	5+
navigation bar	top-level destinations	3-5
tabs	any level of hierarchy	2+

# Application Structure

- Top level views
  - Avoid navigation-only views

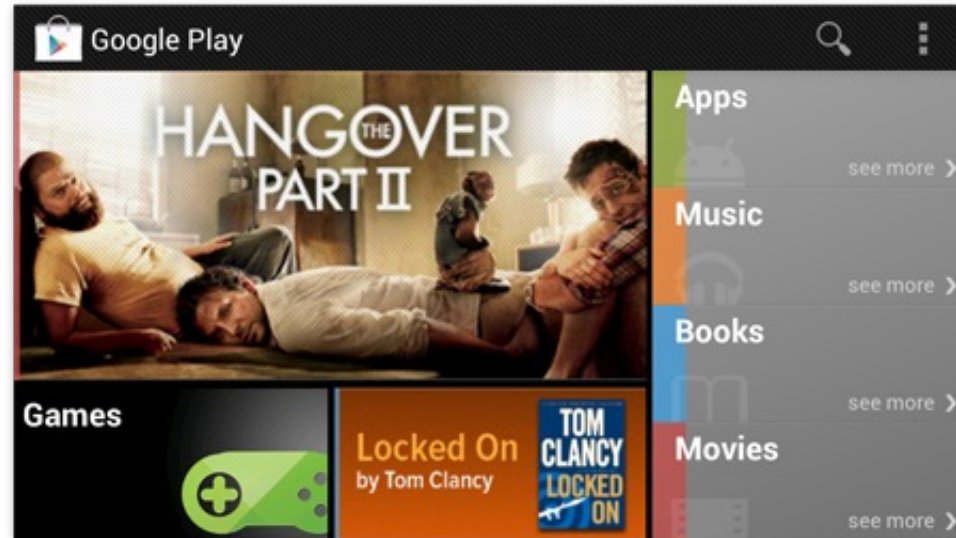


Avoid navigation-only views

<http://developer.android.com/design/patterns/app-structure.html>

# Application Structure

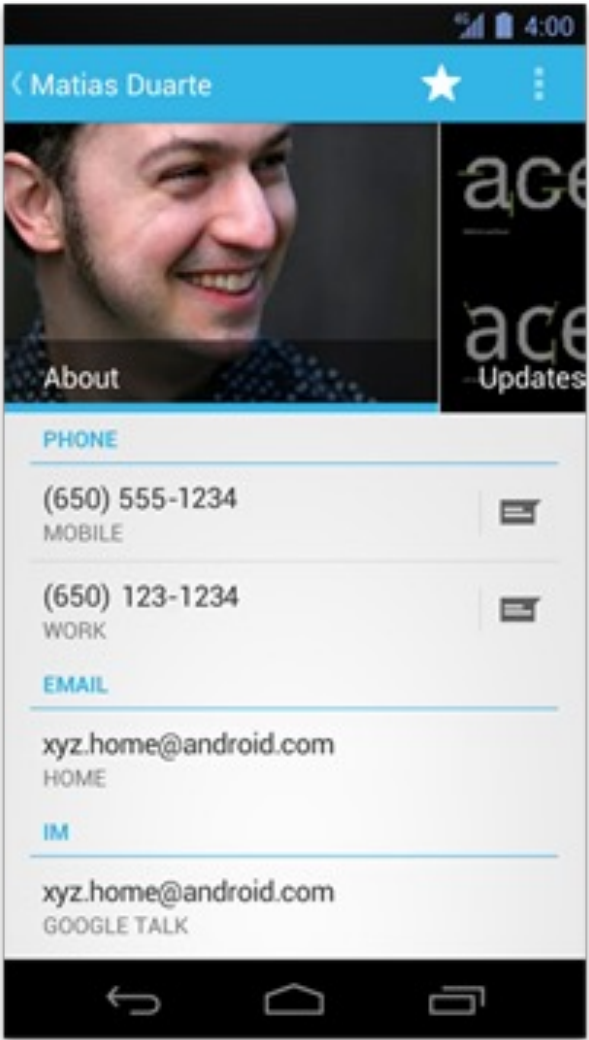
- Top level views
  - Show content on top level
- Detail and edit views
  - Content, structure, interaction



Top level has navigation and content

# Detail Views

- Consider activities users want to perform in the detail view
- Example: People detail view
  - Purpose: access communication options
  - Requirement: efficient scanning, quick access of phone numbers and email addresses, initiation of communication
  - Solution aspect: Split menu items combine calling and messaging

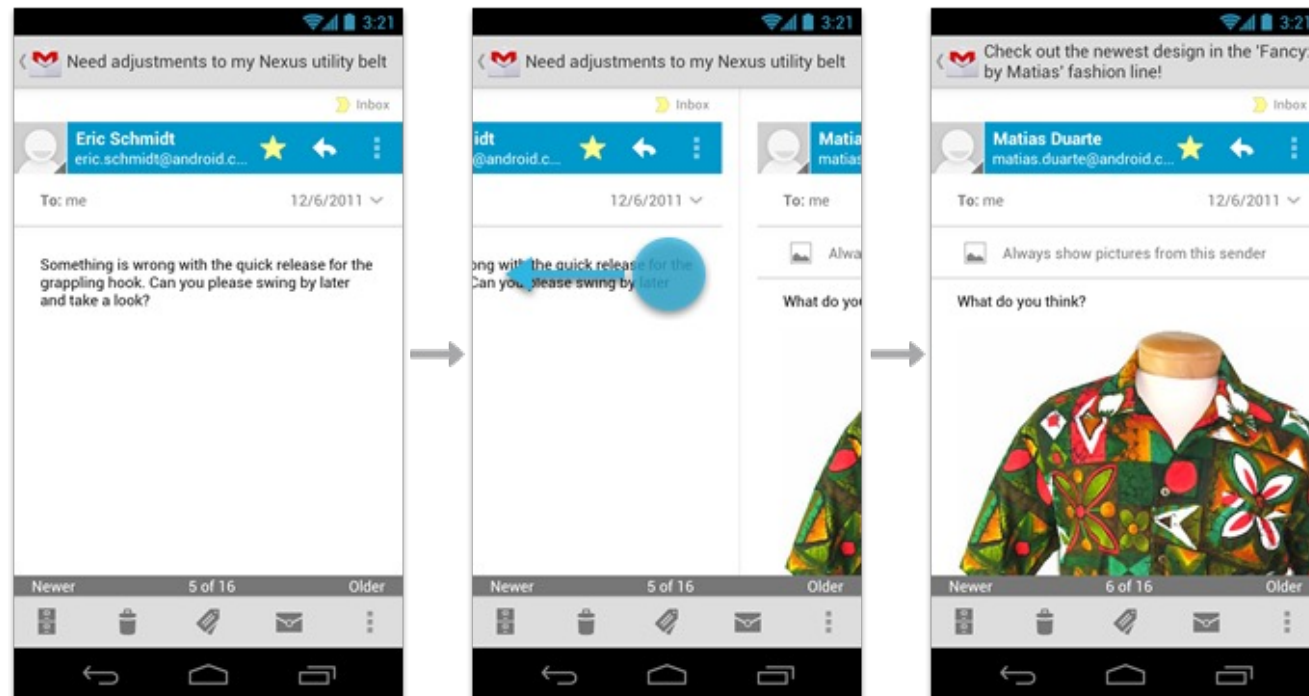


<http://developer.android.com/design/patterns/app-structure.html>



# Efficient Navigation between Detail Views

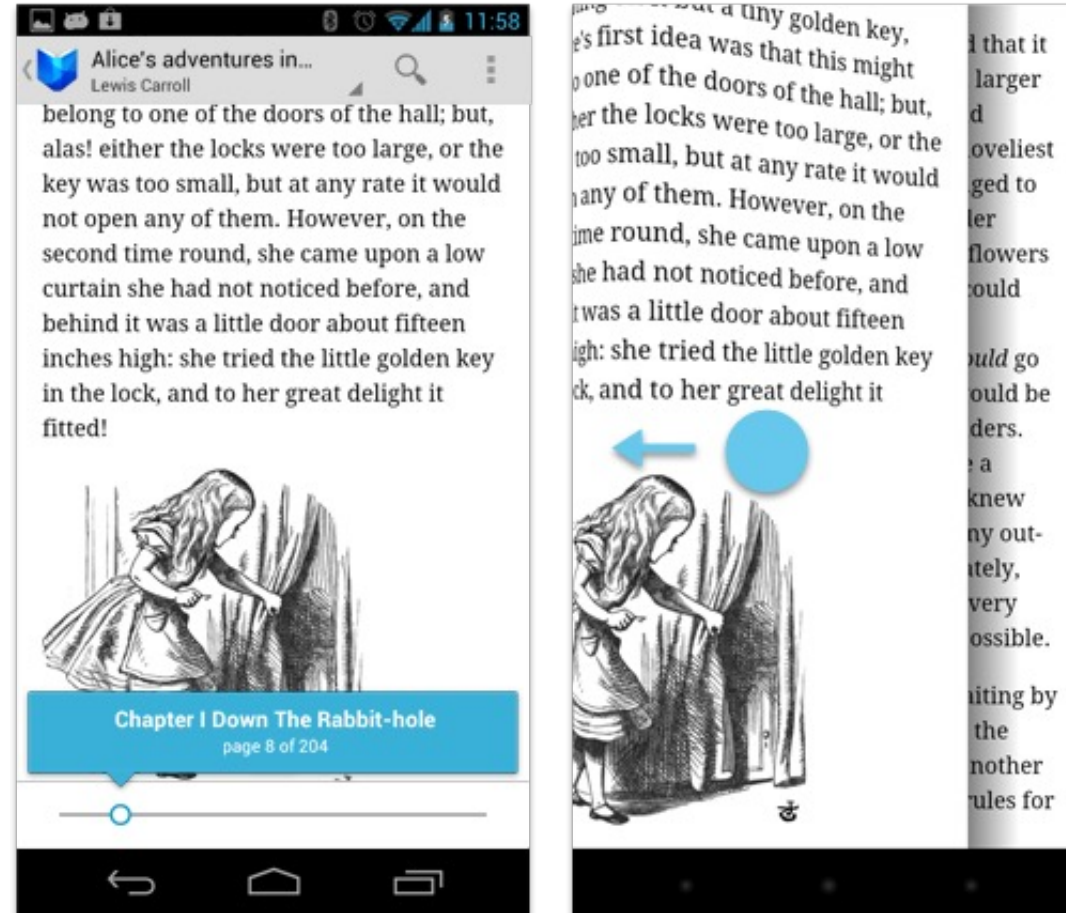
- Avoid need to jump back and forth between overview and detail
  - Example: Horizontal swipe between emails, rather than going back to list each time



<http://developer.android.com/design/patterns/app-structure.html>

## Detail Views: "Lights-Out Mode"

- Full screen:  
Immersive content  
without distractions
- Fade away  
UI elements
- Let user enjoy  
the content
- Touch brings back  
UI elements



<http://developer.android.com/design/patterns/app-structure.html>

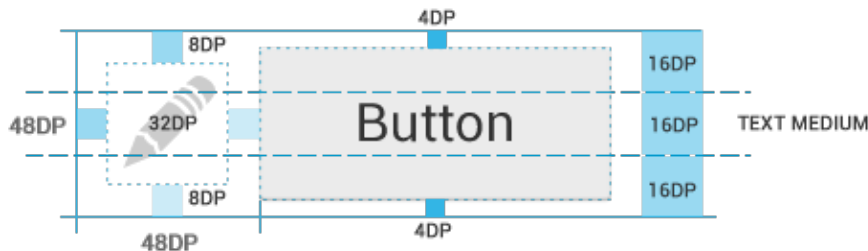
# Metrics and Grids (Android)

## ■ Touchable UI components: 48 dp grid



- $48 \text{ [pixels]} / 160 \text{ [pixels/inch]} = 0.3 \text{ inch} = 7.62 \text{ mm}$
- Reliable touch targets for fingers
- Compromise between information density and targetability
- Margins between UI components: 8 dp
  - $8 \text{ [pixels]} / 160 \text{ [pixels/inch]} = 0.05 \text{ inch} = 1.27 \text{ mm}$
  - Whitespace is an important design element → Gestalt laws

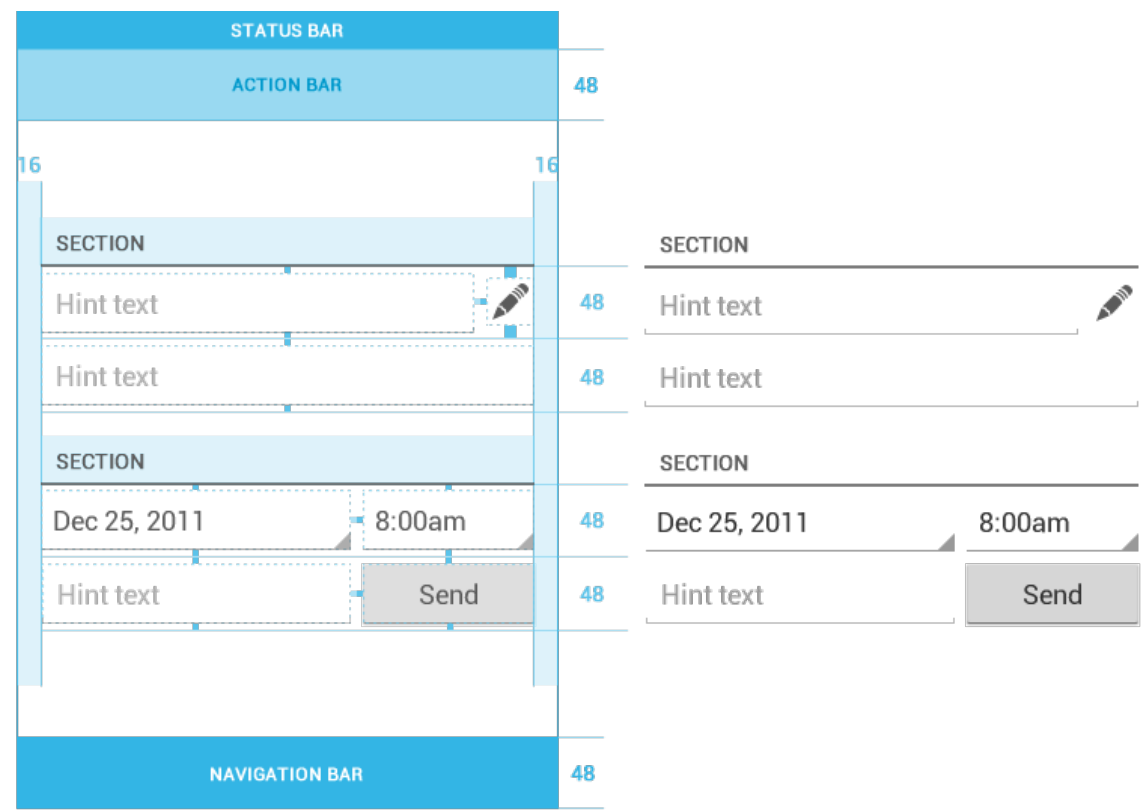
- dpi: dots per inch
- dp: density-independent pixel (assuming a 160 dpi screen)



<http://developer.android.com/design/get-started/ui-overview.html>

# Metrics and Grids (Android)

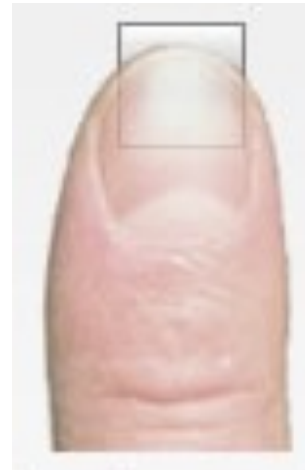
- Touchable UI components: 48 dp (7.62 mm)
- Margins between UI components: 8 dp (1.27 mm)



<http://developer.android.com/design/get-started/ui-overview.html>

# Target Sizes for One-Handed Thumb Use

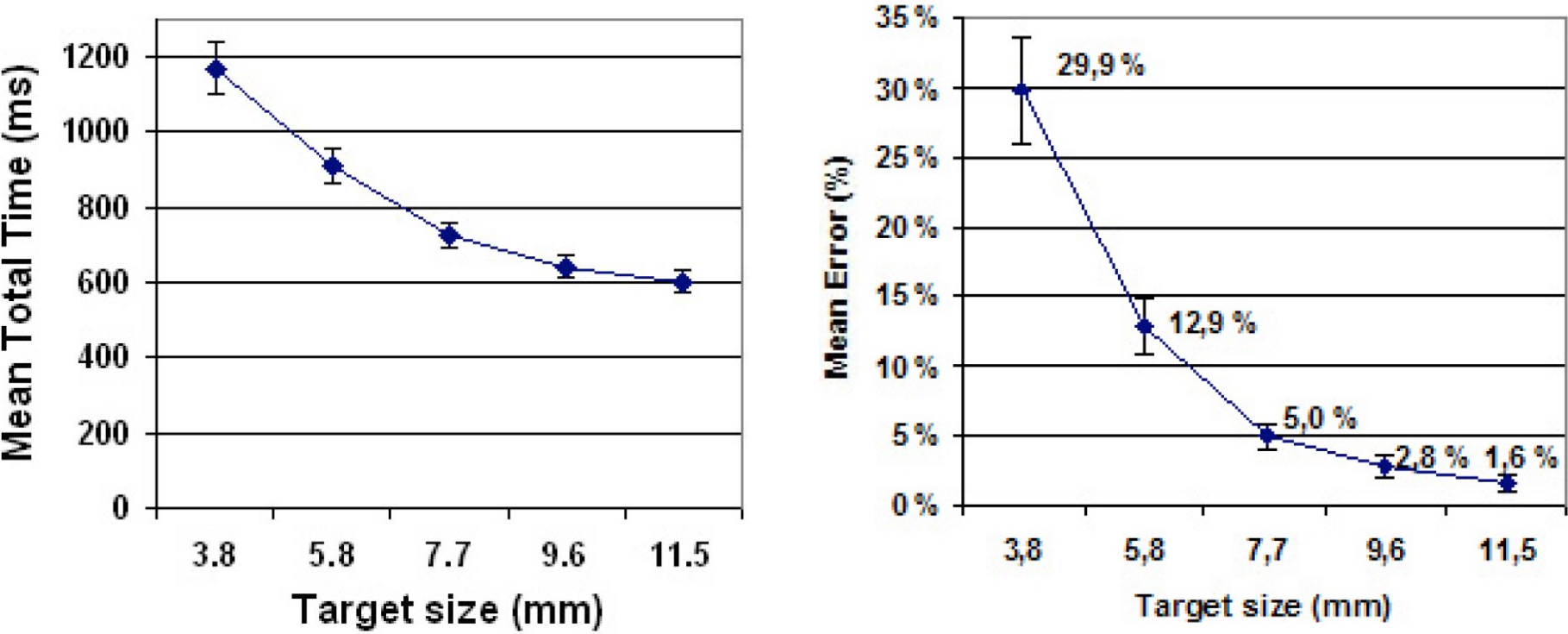
- Recommended target sizes
  - 9.2 mm for discrete pointing tasks (e.g., buttons)
  - 9.6 mm for serial tasks (e.g., text entry)
  - Based on performance and preference results of user studies
- Error rate
  - No differences for targets  $\geq 9.6$  mm (discrete) and  $\geq 7.7$  mm (serial)
  - Android recommends: 7.62 mm (48 dp)



Parhi, Karlson, Bederson. Target size study for one-handed thumb use on small touchscreen devices. MobileHCI 2006

# Target Sizes for One-Handed Thumb Use

- Make targets as small as possible, keep error rate acceptable



Parhi, Karlson, Bederson. Target size study for one-handed thumb use on small touchscreen devices. MobileHCI 2006

# Typography (Android)

- Font family: Roboto
- Font sizes help organize layouts

Text Size Micro 12sp

Text Size Small 14sp

Text Size Medium 18sp

Text Size Large 22sp

- sp: scale-independent pixels, 1 pixel on 160 dpi screen with global text scale set to 100%
- Font colors
  - Not too many colors
  - Consistent use

## Roboto Regular

ABCDEFGHIJKLMN  
OPQRSTUVWXYZ  
abcdefghijklmn  
opqrstuvwxyz

## Roboto Bold

**ABCDEFGHIJKLMN**  
**OPQRSTUVWXYZ**  
**abcdefghijklmn**  
**opqrstuvwxyz**

<http://developer.android.com/design/get-started/ui-overview.html>

# Writing Style

- Brief
  - Concise, precise, clear
- Simple
  - No technical jargon, short words, common words
- Friendly
  - Talk to the reader ("you"), conversational
- Most important information first
  - First two words should convey the most important information
- Only what is necessary, no more
- Avoid repetition



# Writing Style: Examples

## ■ Brief

- "Read the instructions that came with your phone."
- "Your phone is contacting Google. This can take up to 5 minutes."

## ■ Wordy

- "Consult the documentation that came with your phone for further instructions"
- "Your phone needs to communicate with Nokia servers to sign in to your account. This may take up to five minutes."

# Writing Style: Examples

- Simple
  - "GPS – Let apps use satellites to pinpoint your location."
  - "MyApp isn't responding. Do you want to close it?"
- Technical jargon
  - "Use GPS satellites – When locating, accurate to street level."
  - "Sorry! – Activity MyAppActivity (in application MyApp) is not responding."

# Writing Style: Examples

- Most important first
  - "To finish setup using Wi-Fi, touch Next."
- Most important last
  - "Touch Next to complete setup using a Wi-Fi connection."