

LISTS

MOCHI

2012 SEPTEMBER 26 | VERSION 03

Design Lead

Esther Leong (esther.leong@palm.com)

Visual Design

Bryson Ahlstrom (bryson.ahlstrom@palm.com)

Liron Damir (liron.damir@palm.com)

Engineering Lead

Jeremy Thomas (jeremy.thomas@palm.com)

Kevin Schaaf (kevin.schaaf@palm.com)

TABLE OF CONTENTS

Summary & Goals	3
Structure	4
List Item Examples	5
Visual Design Example	6
Platform Variations*	7
Scroll to Top*	9
Auto-Scrolling Dividers*	10
Auto-Scrolling Collapsible Dividers*	11
Alpha Jump List	12
Jump List	14
Example Task Flow: Pull Down to Refresh	16
Example Task Flow: Pull Up to Load More	18
Example Task Flow: Search Filter	20
Example Task Flow: Reordering in a Basic List*	22
Example Task Flow: Moving to a List with Collapsible Dividers*	24
Example Task Flow: Multiselect	26
Swipe-able Items	28
Example Task Flow: Swipe-able Items	29
Example Task Flow: Inline Editing	32
Example Task Flow: Expanding Lists	34
Light Collapsible Divider: Structure & States	35
Light Collapsible Divider: Visual Design Example	36
Example Task Flow: Light Collapsible Divider, Default	37
Example Task Flow: Light Collapsible Divider, Custom	38
Heavy Collapsible Divider: Structure	39
Heavy Collapsible Divider: Visual Design Example	40
Heavy Collapsible Divider: Tablet Variation	41
Heavy Collapsible Divider: Basic Functionality	42
Heavy Collapsible Divider: Top & Bottom Pinning	43
Example Task Flow: Searching Collapsible Dividers	44
Example Task Flow: Multi-select with Collapsible Dividers	45
Example Task Flow: Moving an Item Between Lists	46
Navigable Dividers: Visual Design Example	47
Example Task Flow: Navigable Dividers	48
Change History	49

SUMMARY

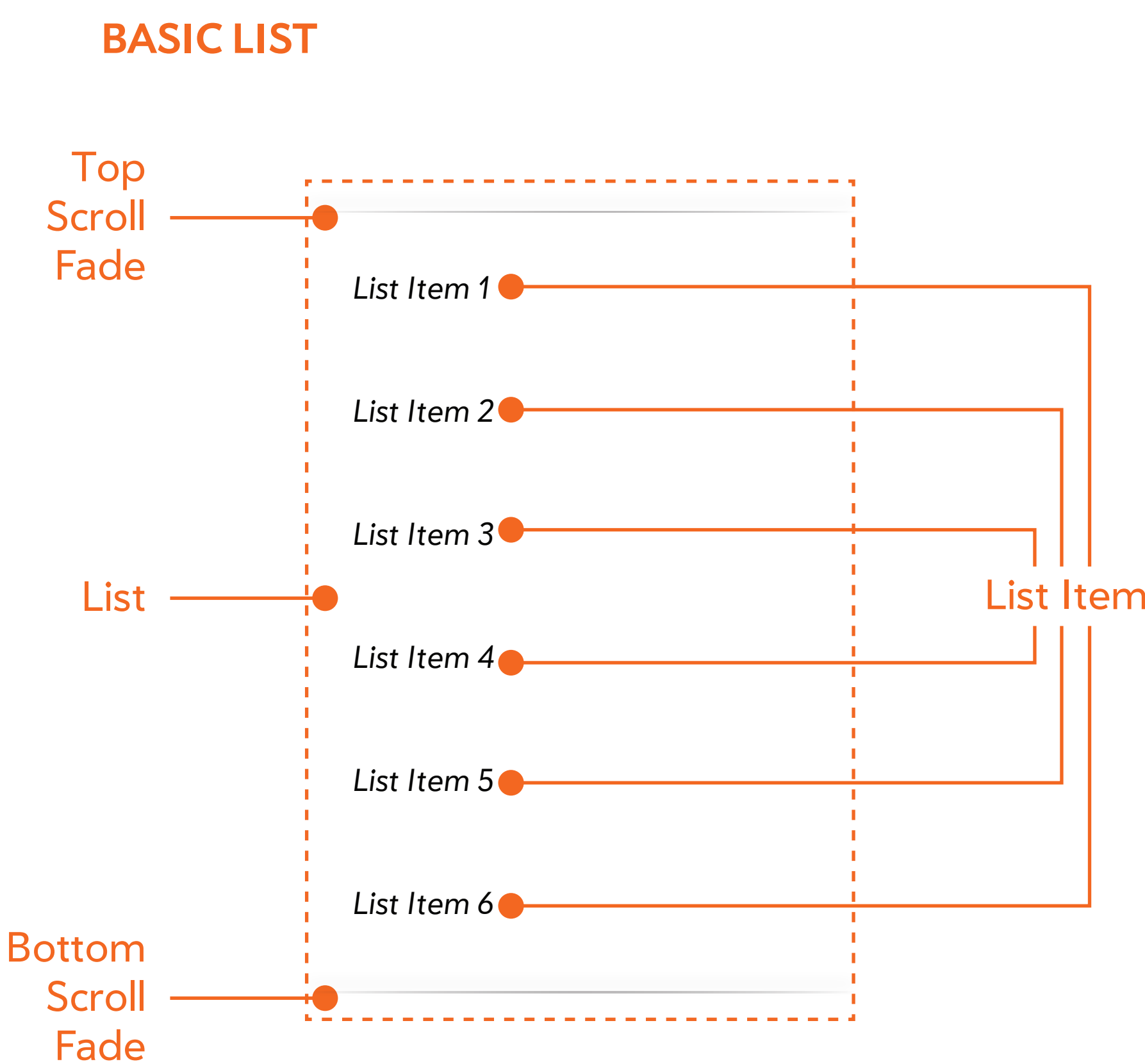
Lists are list items laid out consecutively. The list can have properties as a whole or the individual list item can have its own properties. They can be comprised of the following:

- List Item
- Swipe-able List Item
- Draggable List Item
- Editable List Item
- List
- Reordable List
- Filterable List
- Multi-selectable List
- Pull-down List
- Jump List
- Scroll to Top
- Dividers
- Lazy Loading

GOALS

- To define the various types of lists and their components.
- To show some example task flows illustrating list usage.

STRUCTURE



List

A series of vertical list items; can include scroll fade and dividers. Certain behaviors will be tied to the list itself, such as:

- scrolling (general, incremental, and scroll to top)
- reordering
- multiselect
- filtering
- pull-down (e.g. to refresh)
- jumping (alpha or other category)
- lazy loading

List Item

A single item in a list. Can be single-line or multiple-line. Can contain controls. Can have certain behavior prescribed to it such as the following:

- swipe-able
- draggable
- editable
- expandable
- retaining selected state

Scroll Fade:

Exists at top and/or bottom only when there is content that scrolls in that direction.

Height & Width

Lists have a minimum width and minimum height but no maximum. List items should be constrained to the width of the list they are contained in. Height varies, but there is a recommended max height. Please refer to the redlines for details.

Platform Variations

None. (Alpha-Jump List and Jump List are exceptions.)

LIST ITEM EXAMPLES

SINGLE-LINE

- List Item 1
- List Item 2
- List Item 3

SWIPE-ABLE

- List Item 1
- (Delete)
- List Item 3

INLINE EDIT

- List Item 1
- |
- List Item 3

MULTI-LINE

- List Item 1

12:34 pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...
- List Item 2

Yesterday

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...
- List Item 3

10/07/89

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...

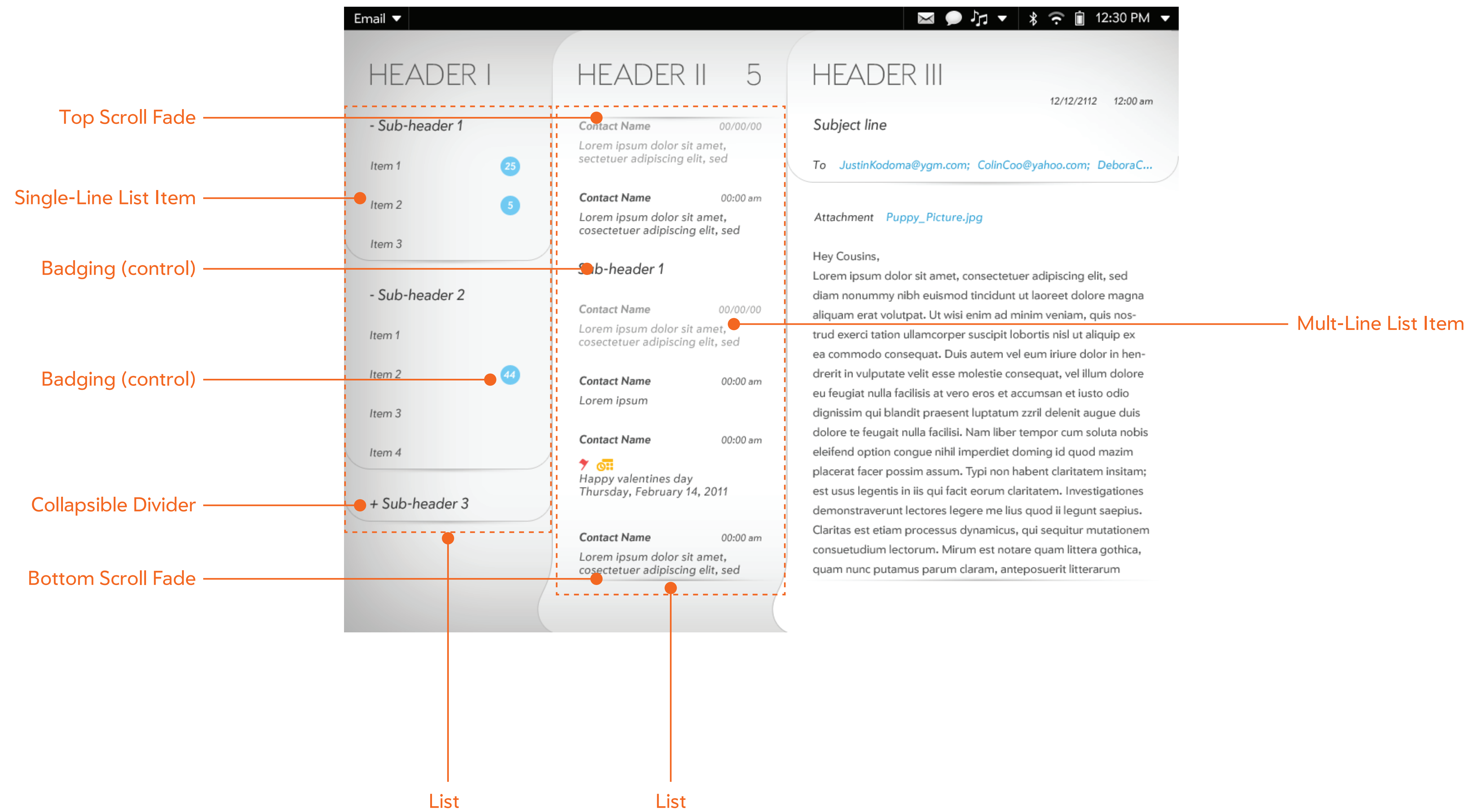
DRAAGABLE

- List Item 1
- List Item 2
- List Item 3

COLLAPSIBLE DIVIDER

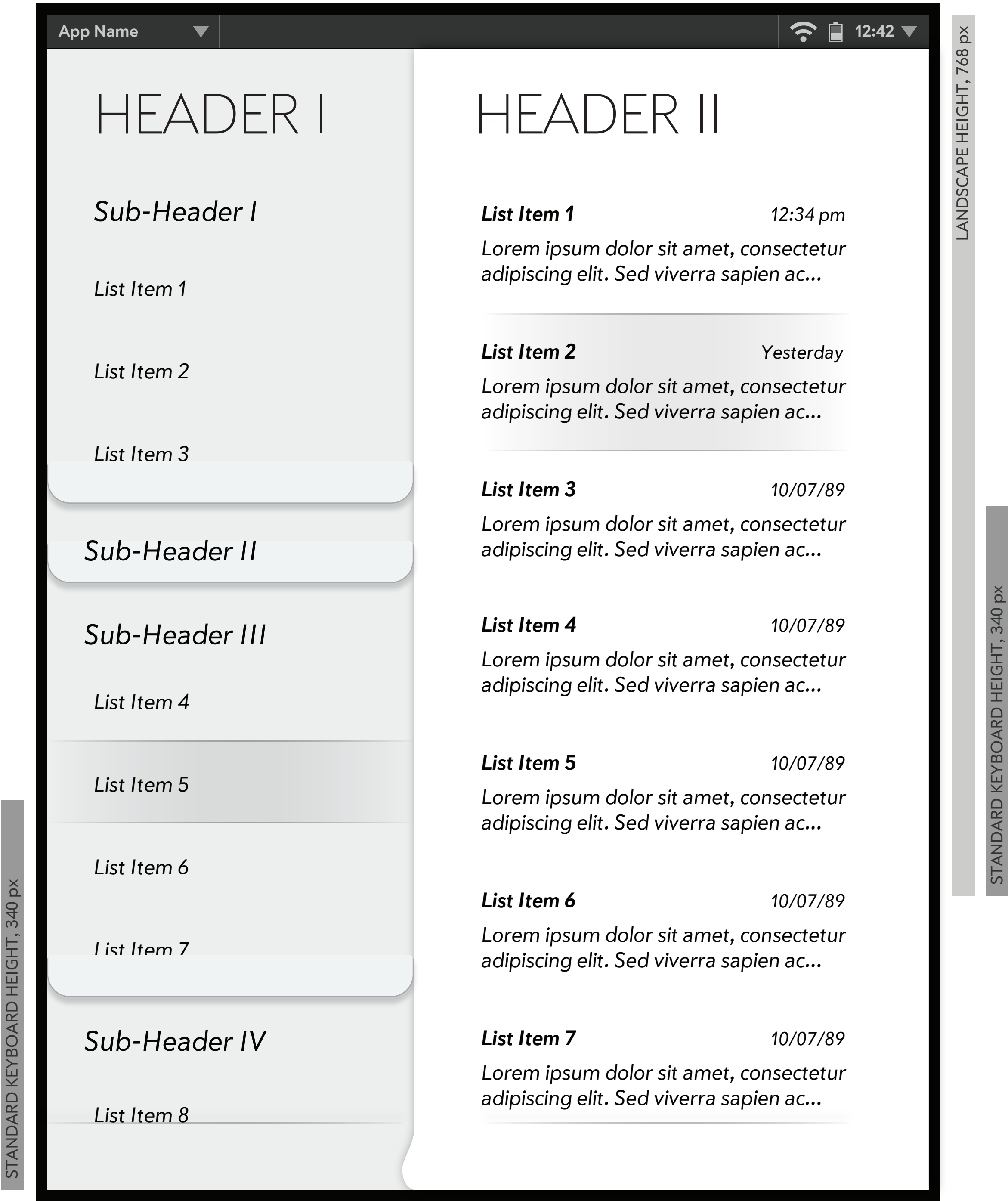
- Sub-Header

VISUAL DESIGN EXAMPLE

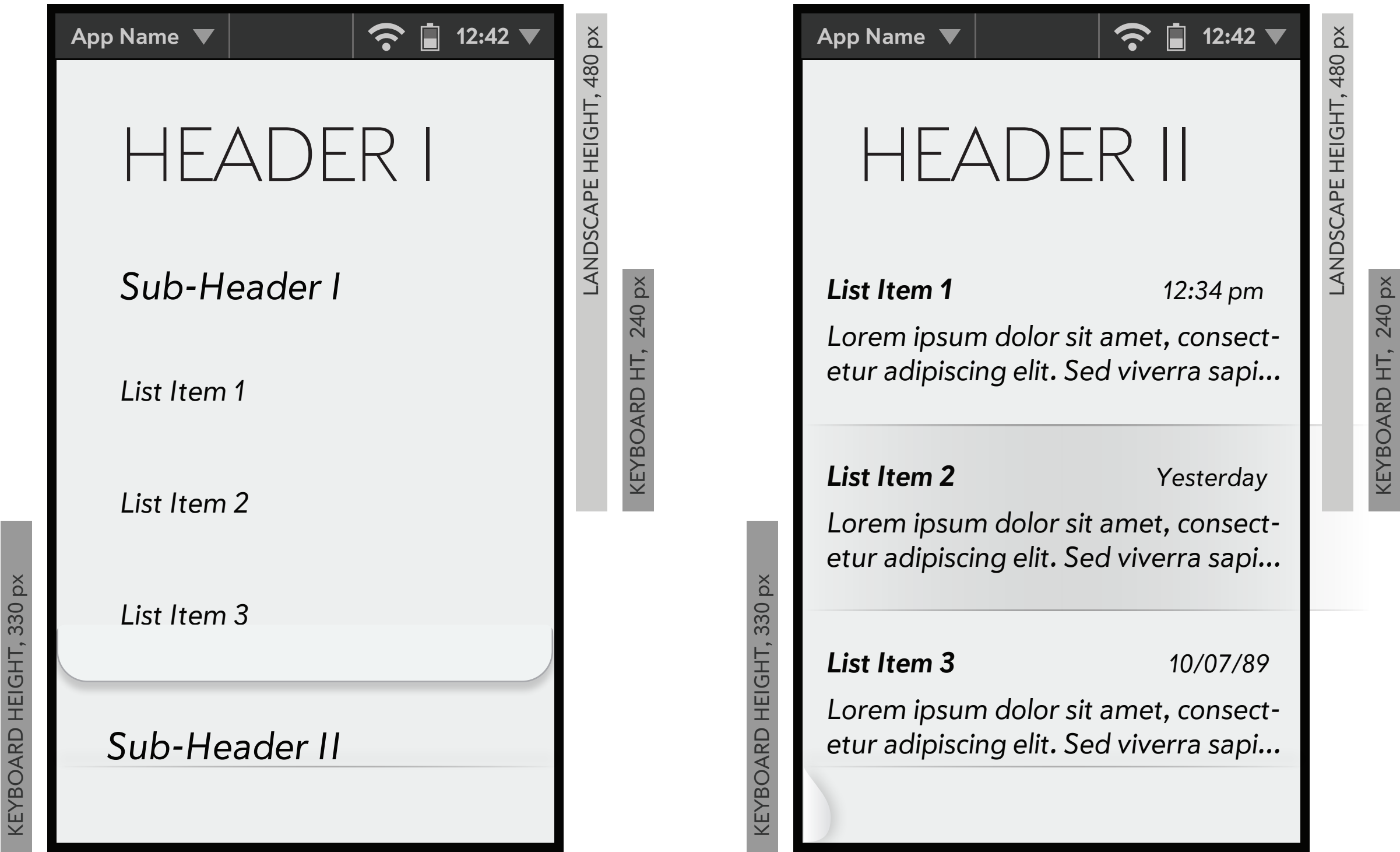


PLATFORM VARIATIONS

TABLET

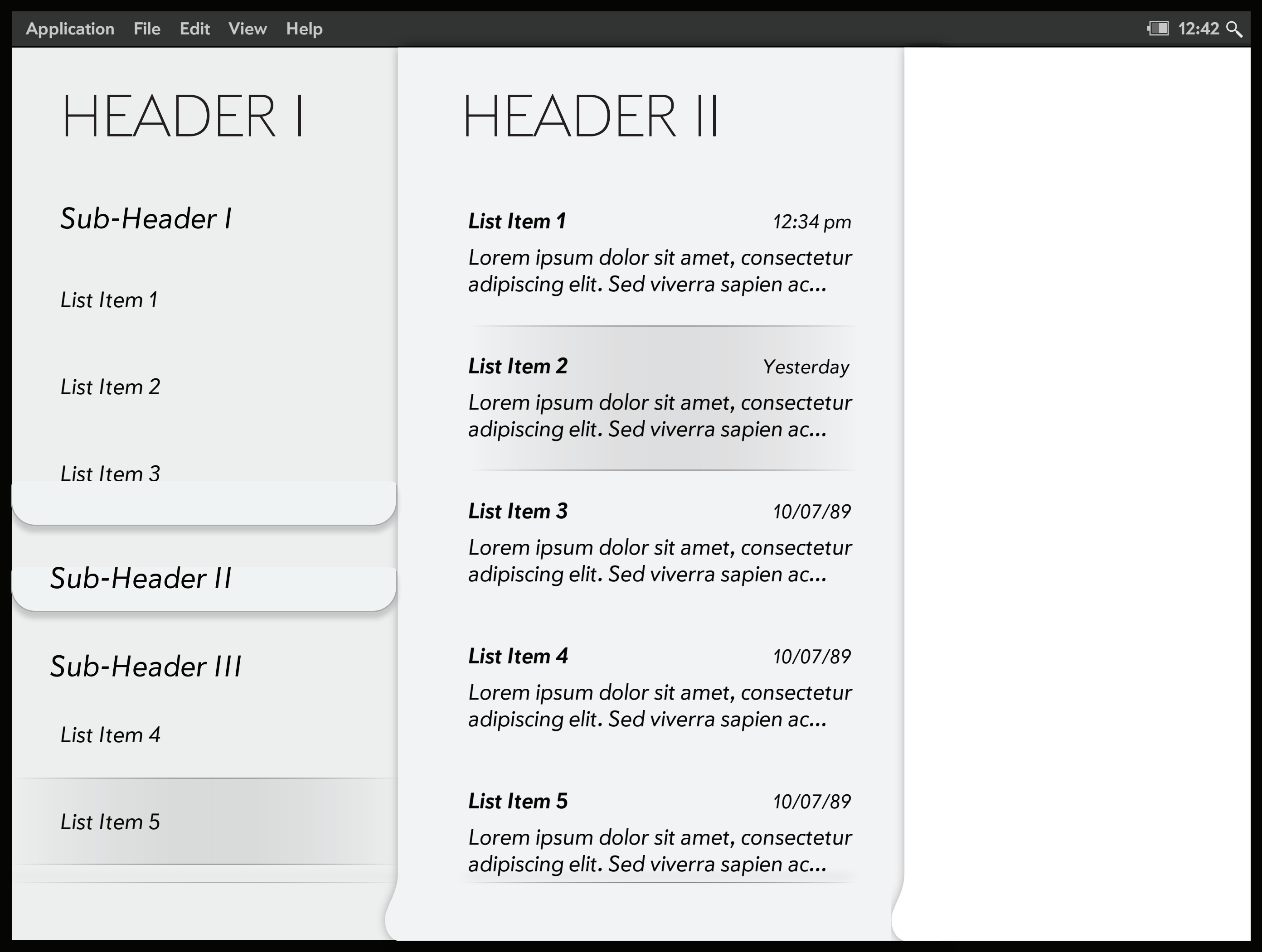


PHONE



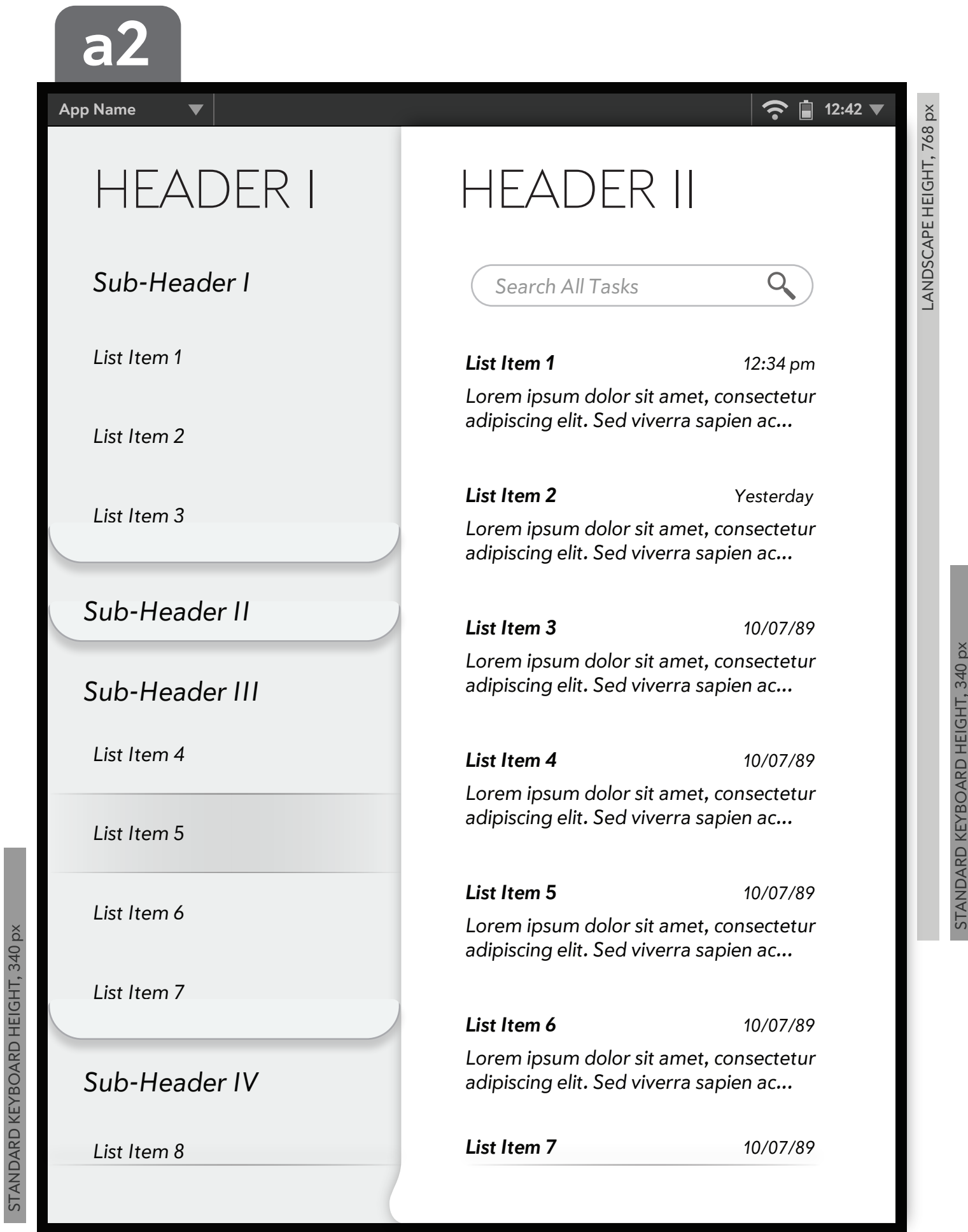
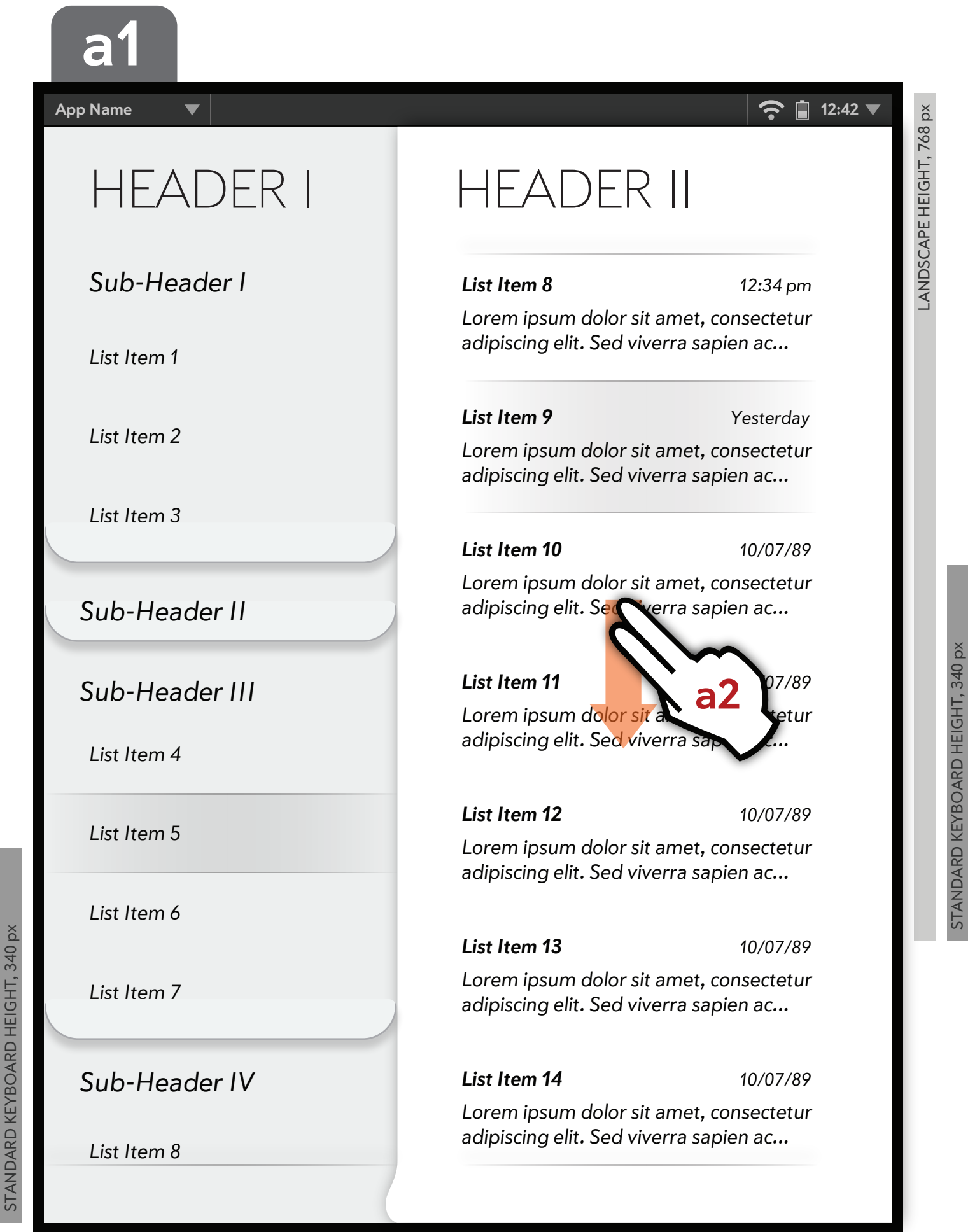
PLATFORM VARIATIONS*

PC



Please refer to documentation on Panels for more information on panel behavior and orientation.

SCROLL TO TOP*



This interaction is tentative. There have been usability concerns on iOS where 2 finger swipes are not consistently recognized.

a1. Two-finger swipe down in a list.

Flow demonstrates a variation of a list on a tablet device and is for illustration purposes only. Please refer to the Visual Design specs for details.

a2. Upon release, scroll to top of list.

This gesture should work in ALL lists.

AUTO-SCROLLING DIVIDERS*



b1. Users can auto-scroll from divider to divider in a list by dragging down or up with two fingers.

Please note that pinning dividers to the top is pending. For visual design, dividers are currently text-only.

b2. Dragging down in the list should scroll the list upwards (as in direct manipulation). The user can drag to scroll through multiple dividers. Upon release, stop scrolling.

b3. Dragging up in the list should scroll the list downwards. The user can drag to scroll through multiple dividers. Upon release, stop scrolling.

This gesture should not work in Alpha Jump Lists or Jump lists.

AUTO-SCROLLING COLLAPSIBLE DIVIDERS*



c1. Users can auto-scroll from divider to divider in a list by dragging down or up with two fingers.

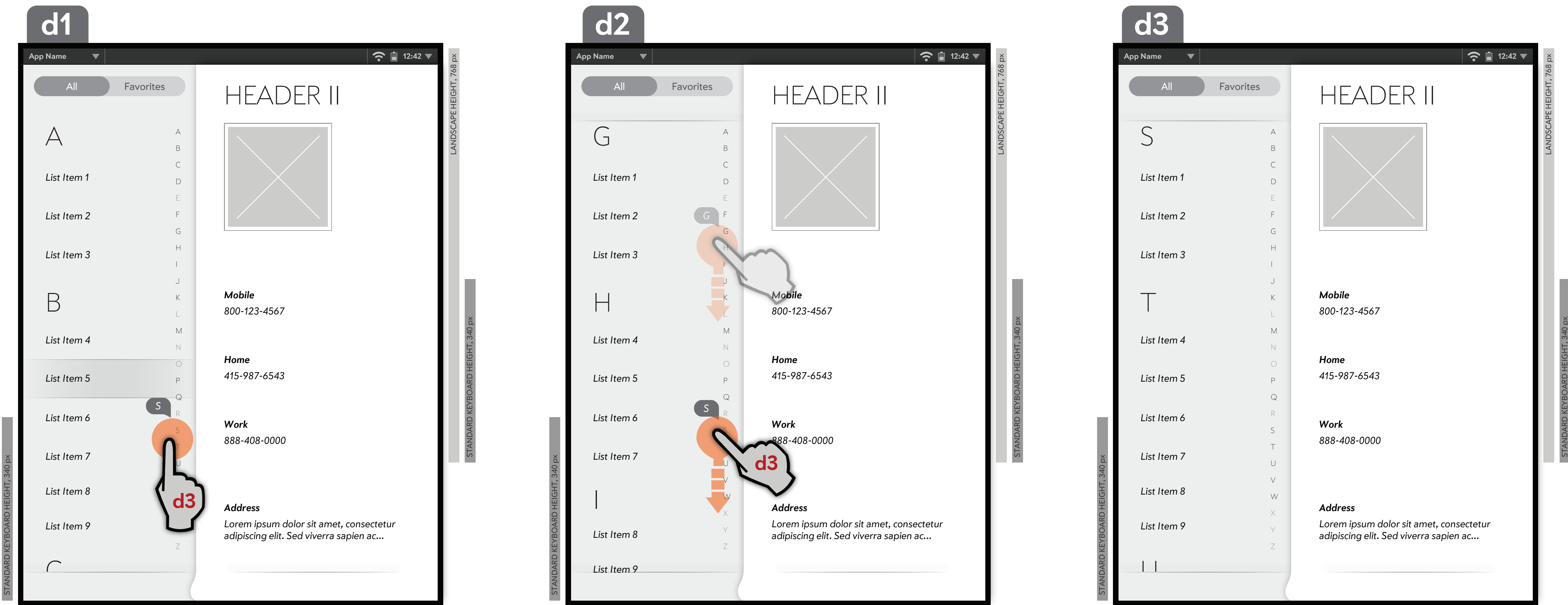
For more information on Collapsible Dividers, please refer to [CollapsibleDividers_Wireframes_2012-08-30.pdf](#).

c2. Dragging down in the list should scroll the list upwards (direct manipulation). The user can drag to scroll through multiple dividers. Upon release, stop scrolling.

c3. Dragging up in the list should shift the list downwards. The user can drag to scroll through multiple dividers. Upon release, stop scrolling.

This gesture should not work in Alpha Jump Lists or Jump lists.

ALPHA-JUMP LIST



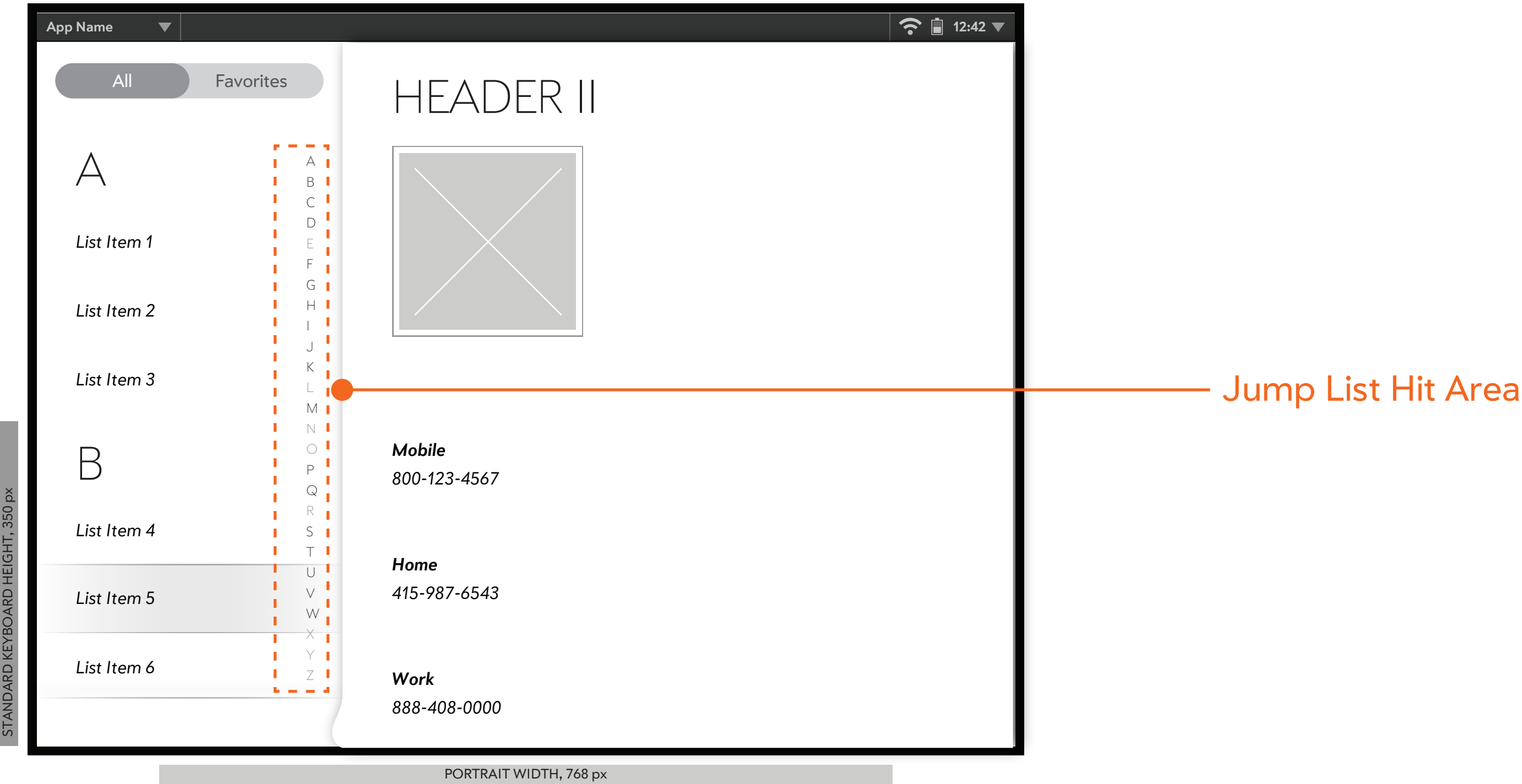
d1. User can tap anywhere on the bar to jump to the corresponding point in the list (**d3**). If no items exist for that letter, go the closest option above. On tap, display tooltip with corresponding letter.

d2. User can also drag along the bar. Display tooltip and update value dynamically until release. Upon release, jump to the corresponding point in list.

d3. Upon release, auto-scroll to corresponding point in list. Please note that 2 finger dragging (**b1-b3**, **c1-c3**) should not work on Alpha-Jump or Jump Lists.

ALPHA JUMP LIST: PLATFORM/ORIENTATION VARIATIONS

TABLET/PC (LANDSCAPE)



Platform Variations

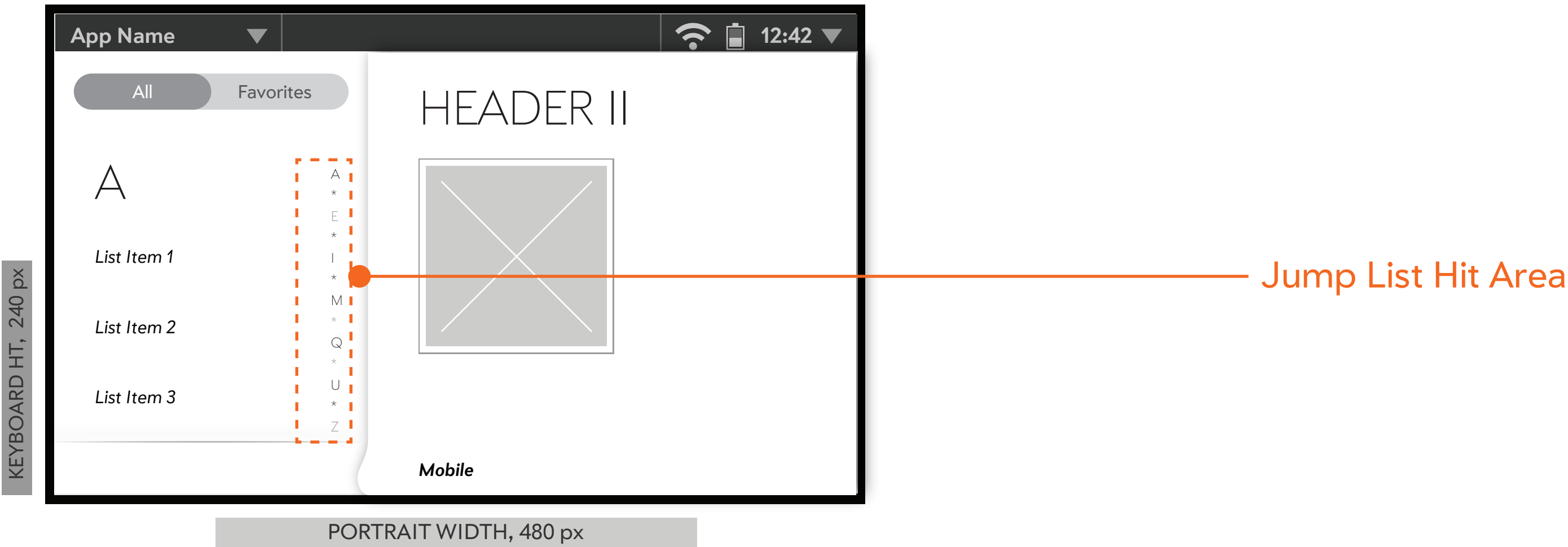
The alpha jump list bar has a fixed width and a maximum height. Leading between letters has a minimum referred to in the redline documentation. Increments are the following:

- Full alphabet
- Every 3rd letter (i.e. A, D, G, J... etc.)
- Every 4th letter (i.e. A, E, I, M... etc.)

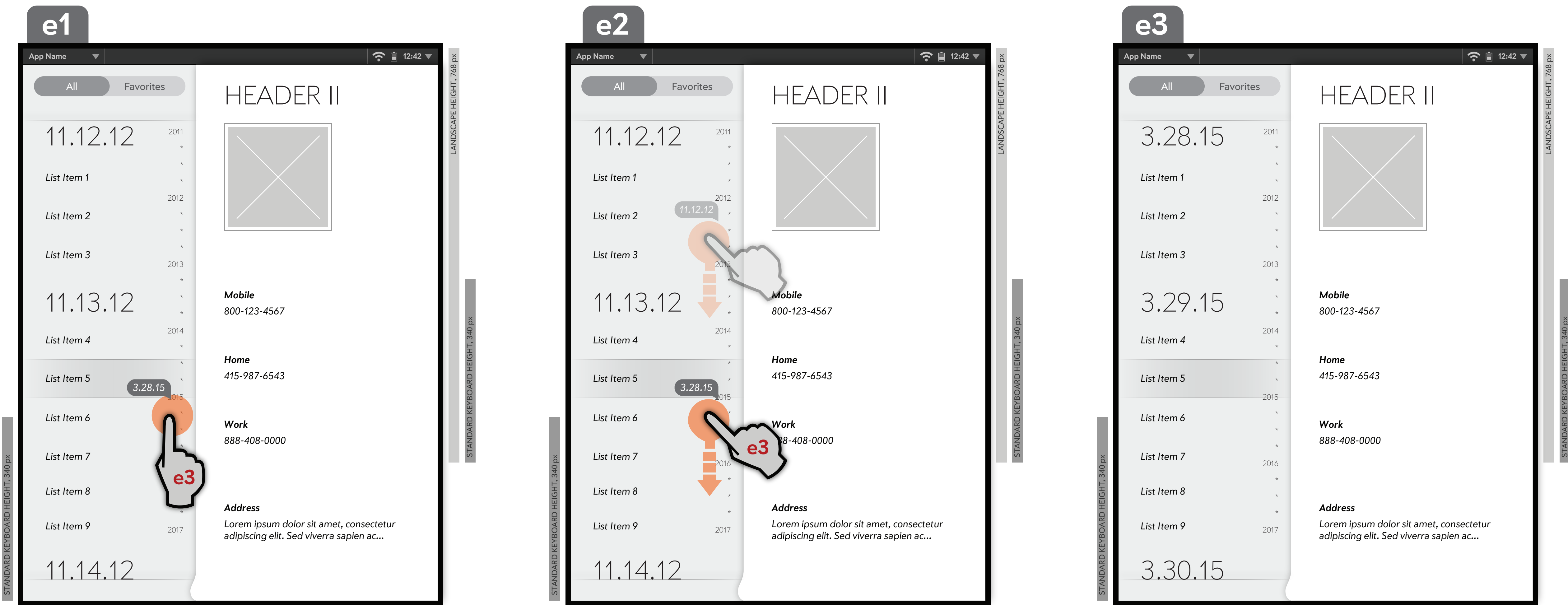
Increments of every 4th letter is the minimum size (see the bottom figure PHONE (LANDSCAPE)).

Please refer to the *AlphaJumpList_Redlines.pdf* for more detailed visual design.

PHONE (LANDSCAPE)



JUMP LIST*



e1. Increments can simply be dots that correspond to dividers in the list. Dates or words are optional.

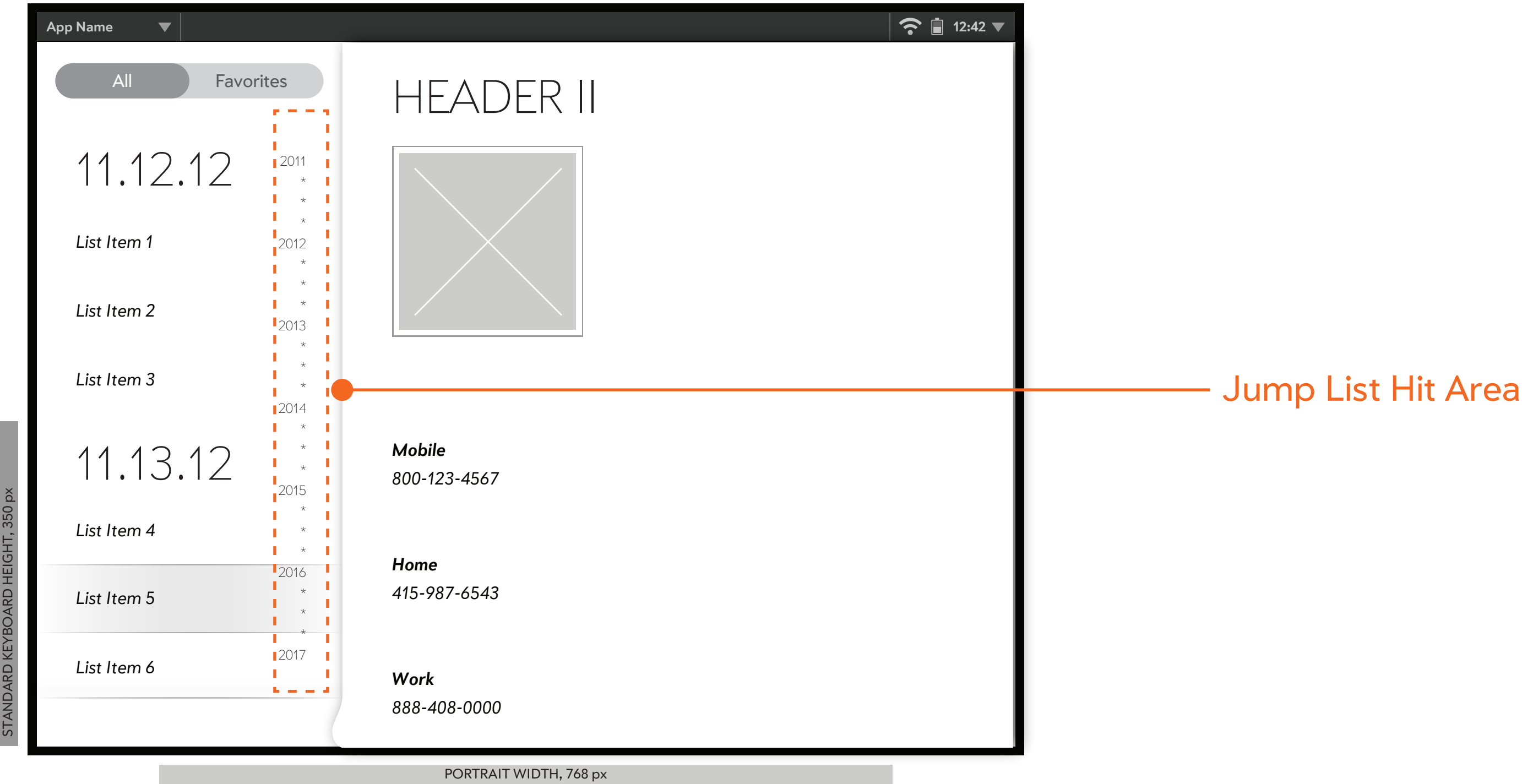
User can tap anywhere on the bar to jump to the corresponding point in the list (**e3**). On tap, display tooltip with corresponding letter.

e2. User can also drag along the bar. Display tooltip and update value dynamically until release. Upon release, jump to the corresponding point in list.

e3. Upon release, auto-scroll to corresponding point in list. Please note that 2 finger dragging (**b1-b3**, **c1-c3**) should not work on Alpha-Jump or Jump Lists.

JUMP LIST: PLATFORM VARIATIONS*

TABLET/PC (LANDSCAPE)



Platform Variations

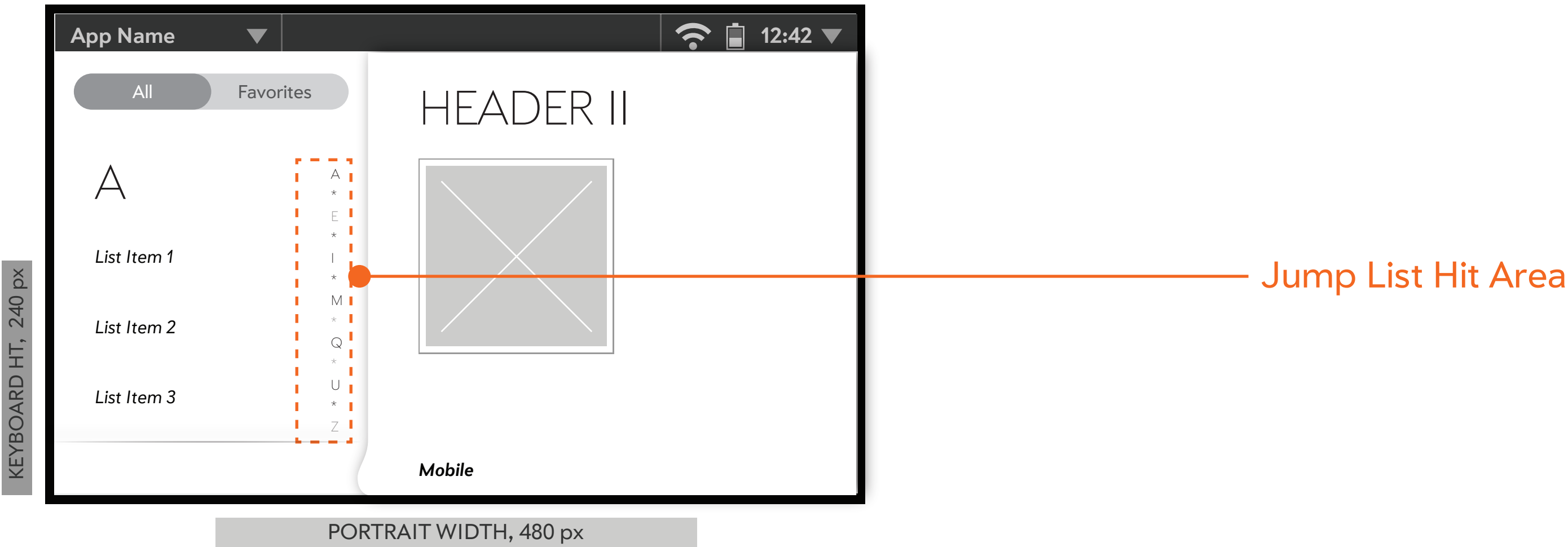
The jump list bar has a fixed width and a maximum height. Leading between increments has a minimum referred to in the redline documentation. Each increment should represent a divider in the list until a minimum size is reached. Increments are the following:

- All
- Every 3rd divider
- Every 4th divider

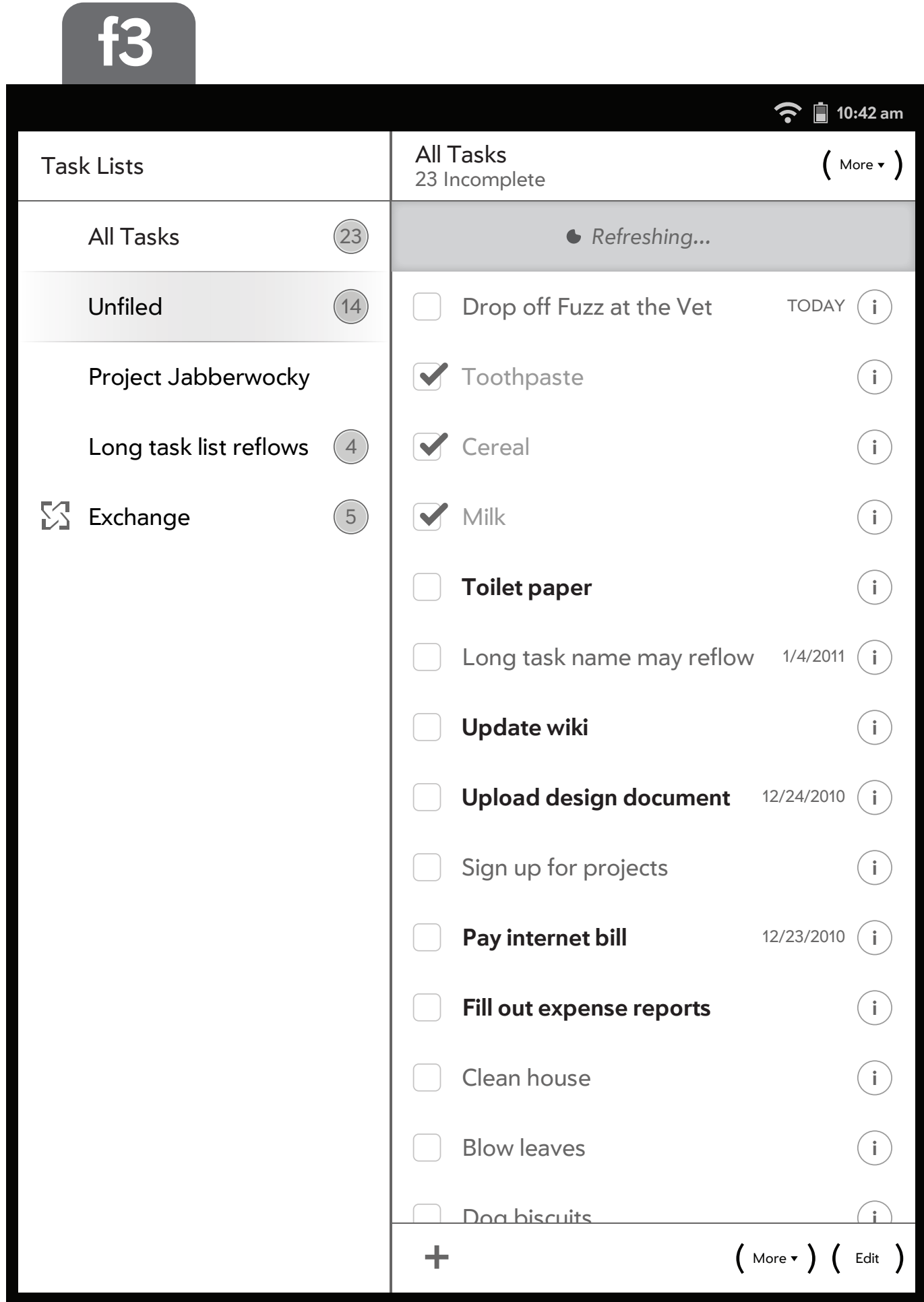
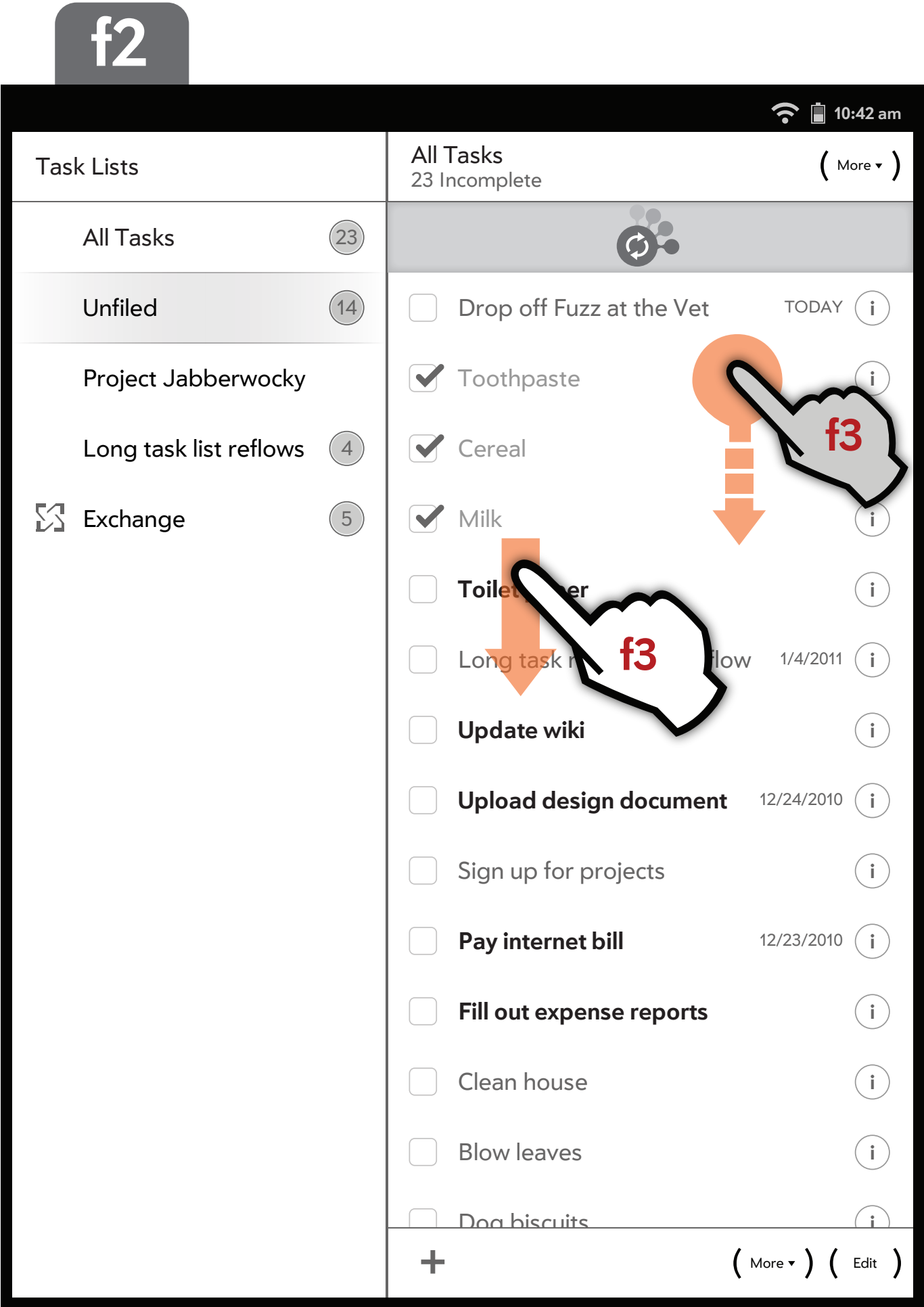
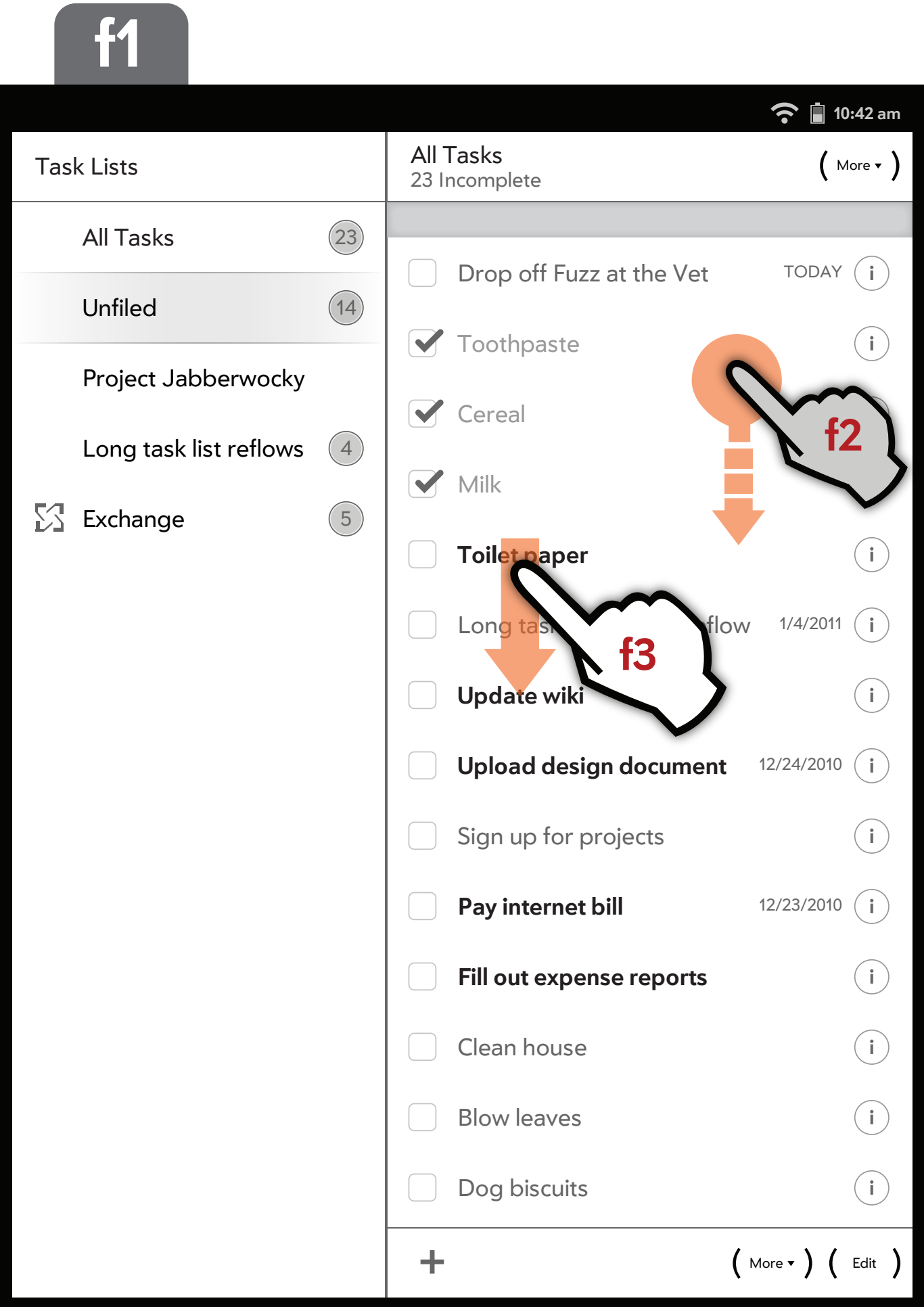
Increments of every 4 is the minimum size (see the bottom figure PHONE (LANDSCAPE)).

Please refer to the *JumpList_Redlines.pdf* for more detailed visual design.

PHONE (LANDSCAPE)



EXAMPLE TASK FLOW: PULL DOWN TO REFRESH



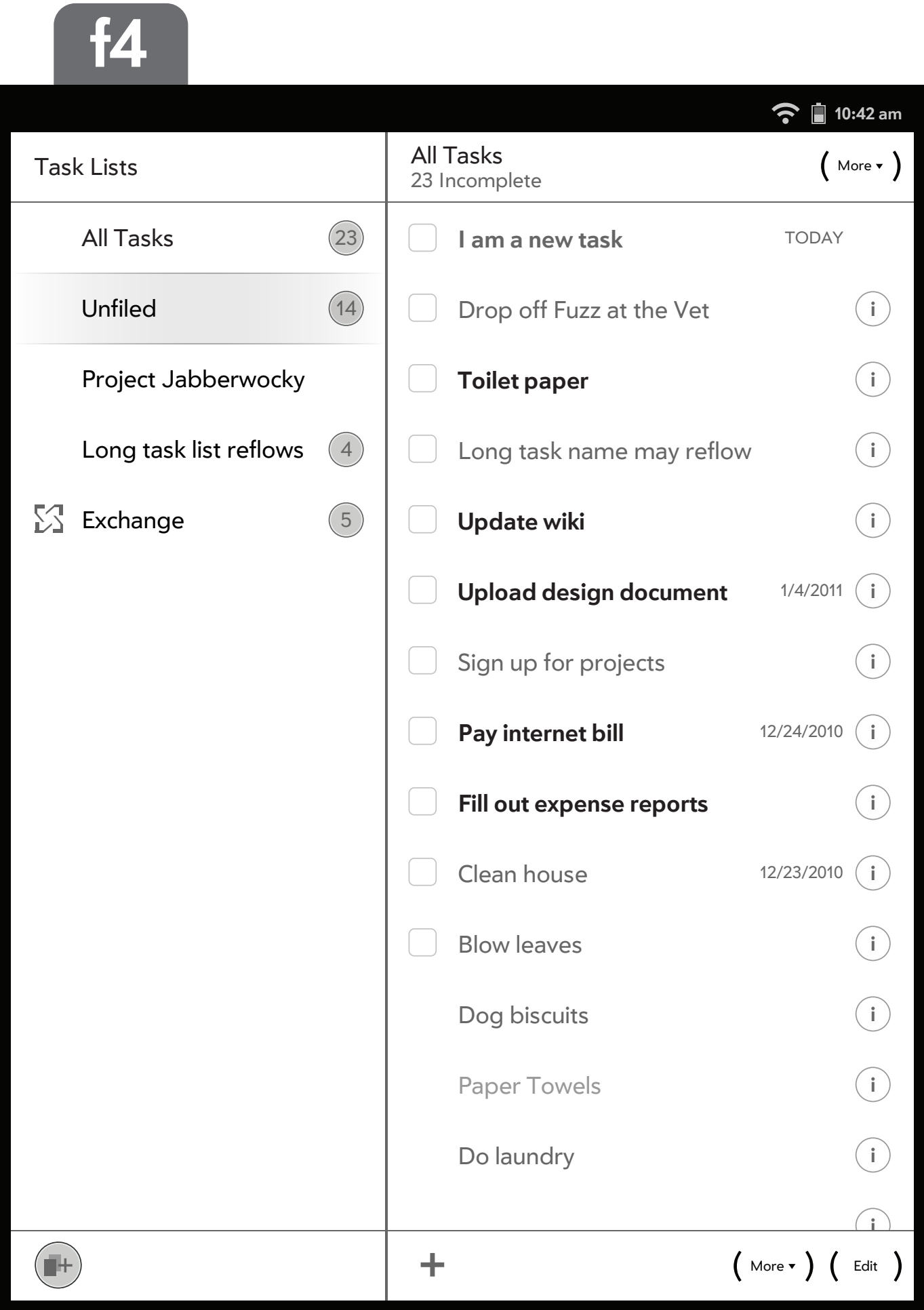
f1. Swipe or drag down from the top of the list.

Flow demonstrates a variation of a list on a tablet device and is for illustration purposes only. Visual design and animation is not represented here.

f2. If the list is pulled to a certain threshold, the “refresh” icon will begin to rotate.

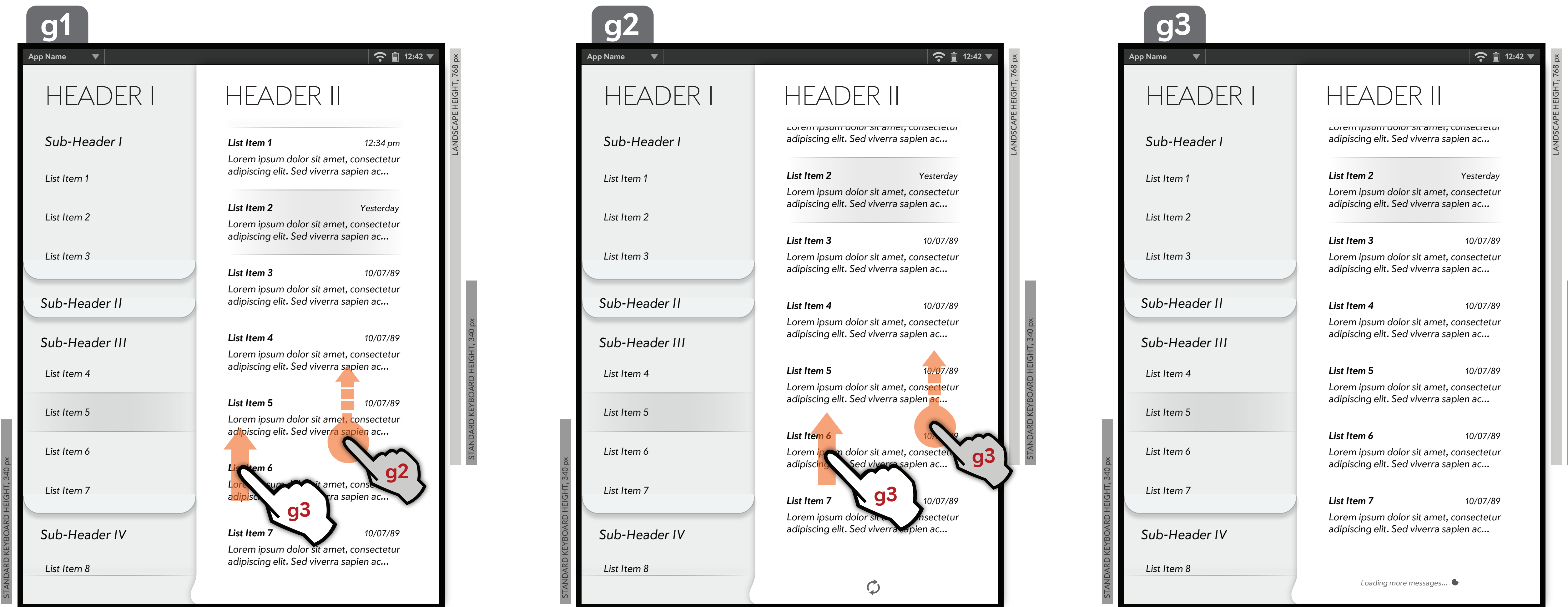
f3. If the user releases while the refresh icon is spinning, display a spinner and re-fresh the scene.

EXAMPLE TASK FLOW: PULL DOWN TO REFRESH (CONT.)



f4. List is refreshed.

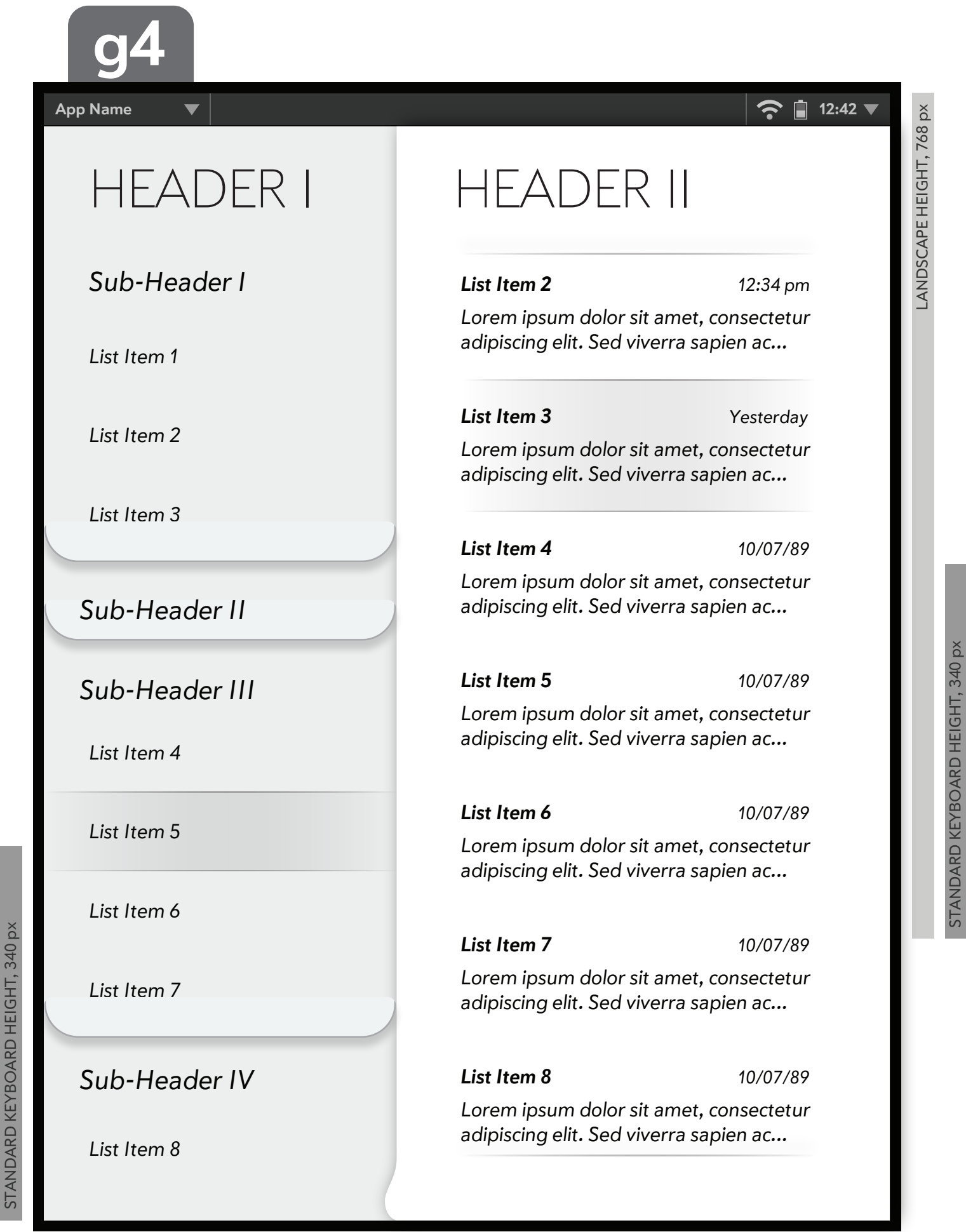
EXAMPLE TASK FLOW: PULL UP TO LOAD MORE (LAZY LOADING)



- g1.** Swipe or drag down from the bottom of the list.
- g2.** If the list is pulled to a certain threshold, an icon will appear.
- g3.** If the user releases while the icon is there, display a spinner and refresh the scene.

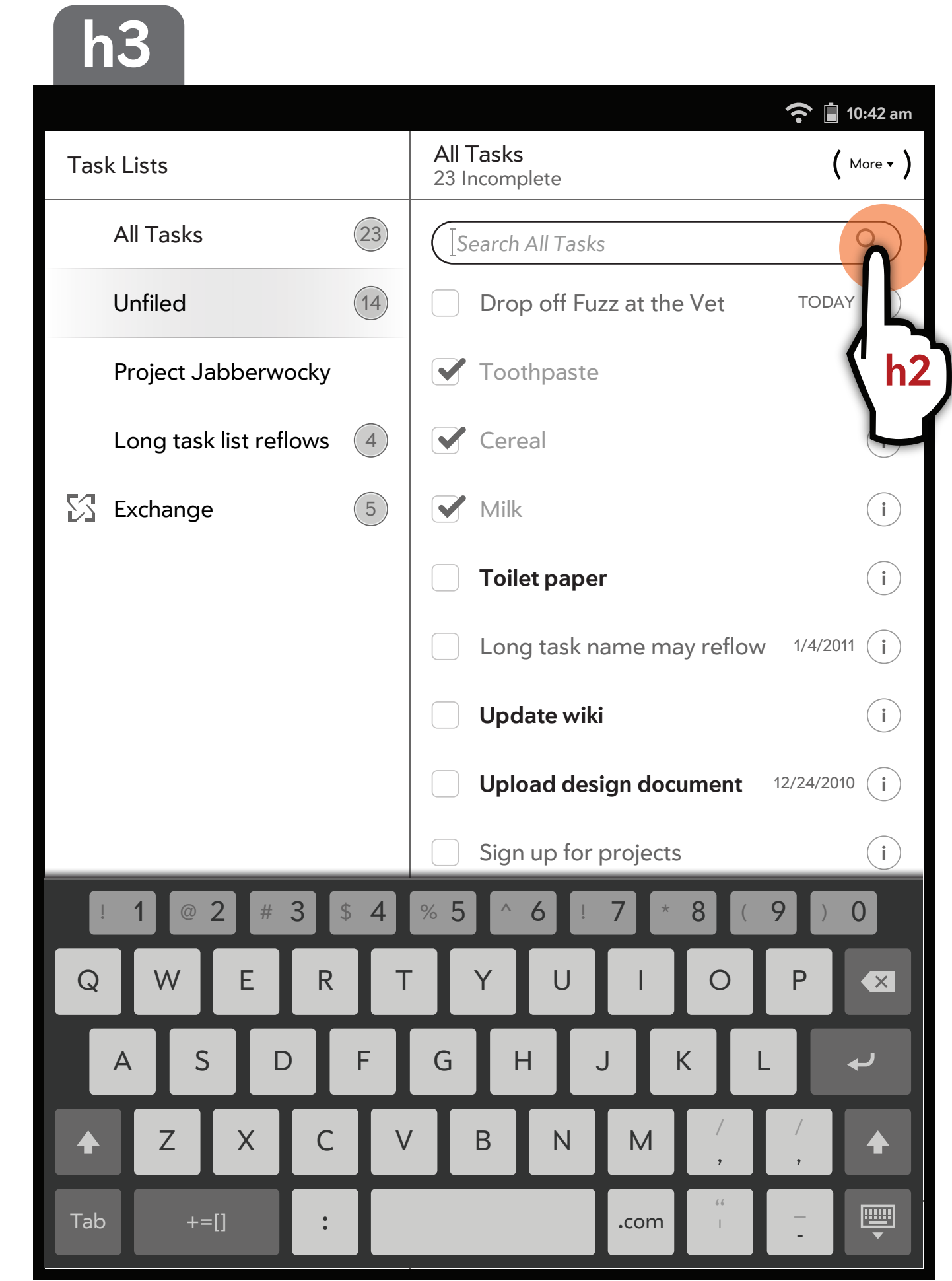
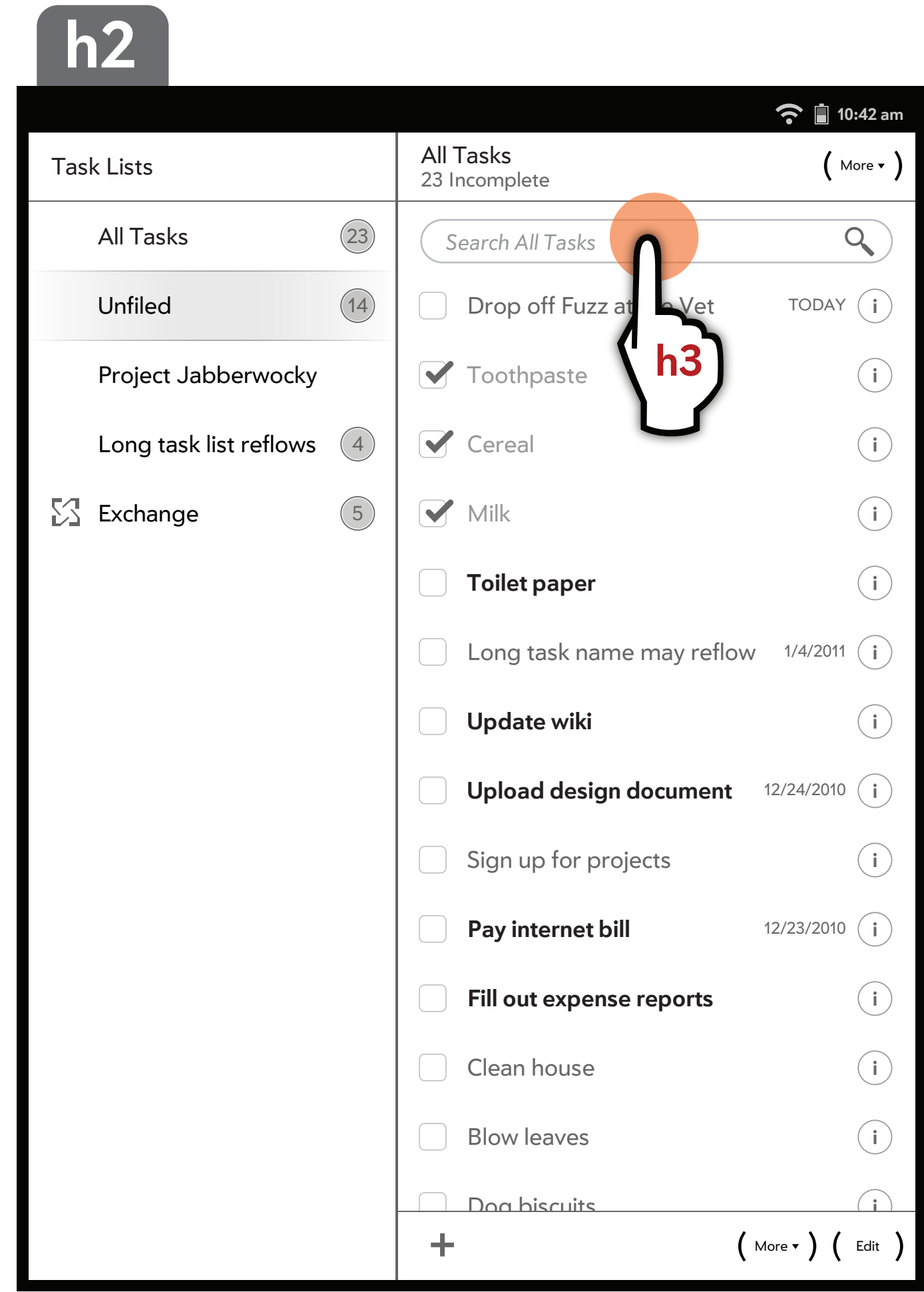
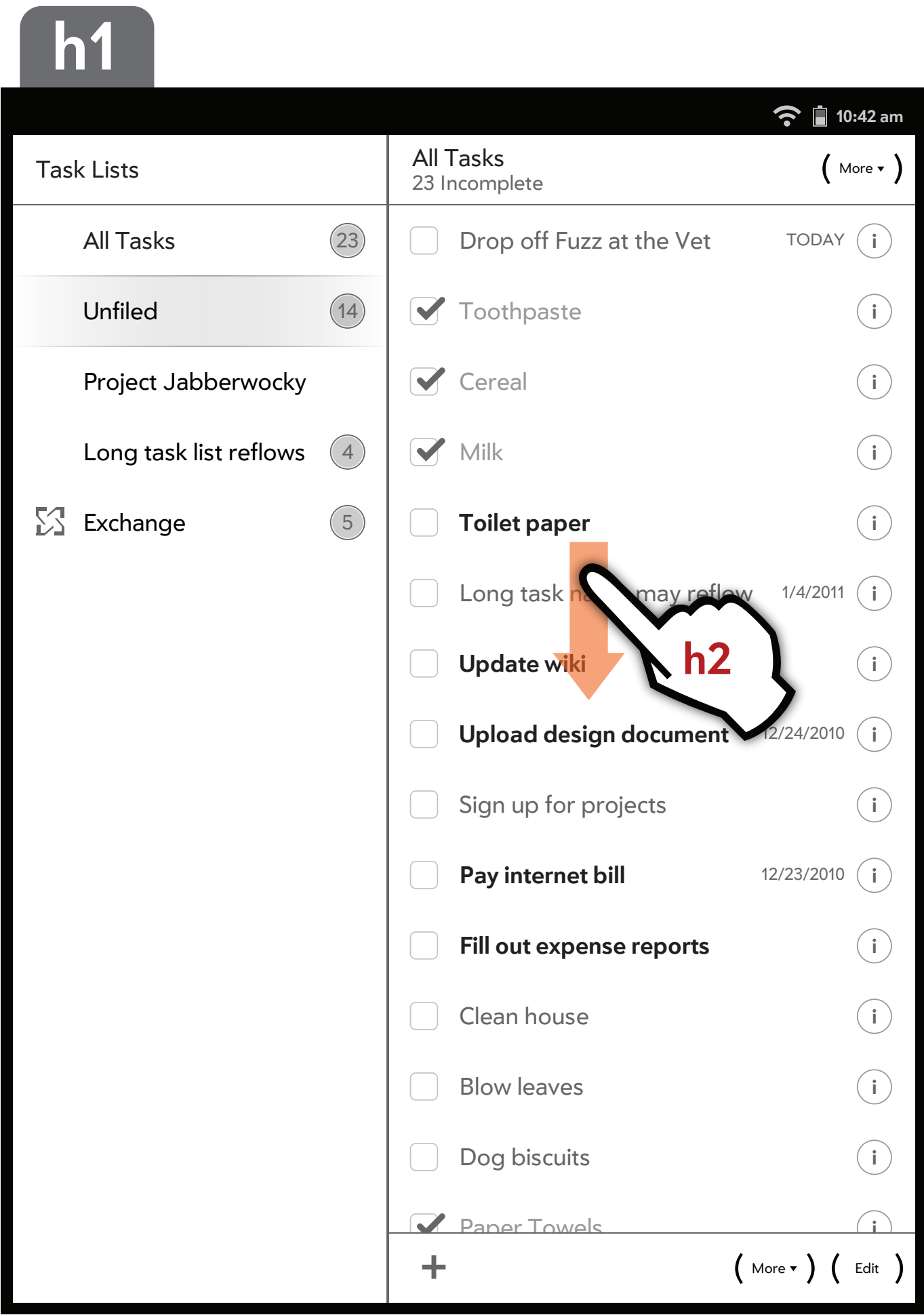
Flow demonstrates a variation of a list on a tablet device and is for illustration purposes only. Visual design and animation is not represented here.

EXAMPLE TASK FLOW: PULL UP TO LOAD MORE (CONT.)



g4. List is updated.

EXAMPLE TASK FLOW: SEARCH FILTER



h1. Overscroll down to reveal hidden Search Input field.

If default is not hidden, skip to h2.

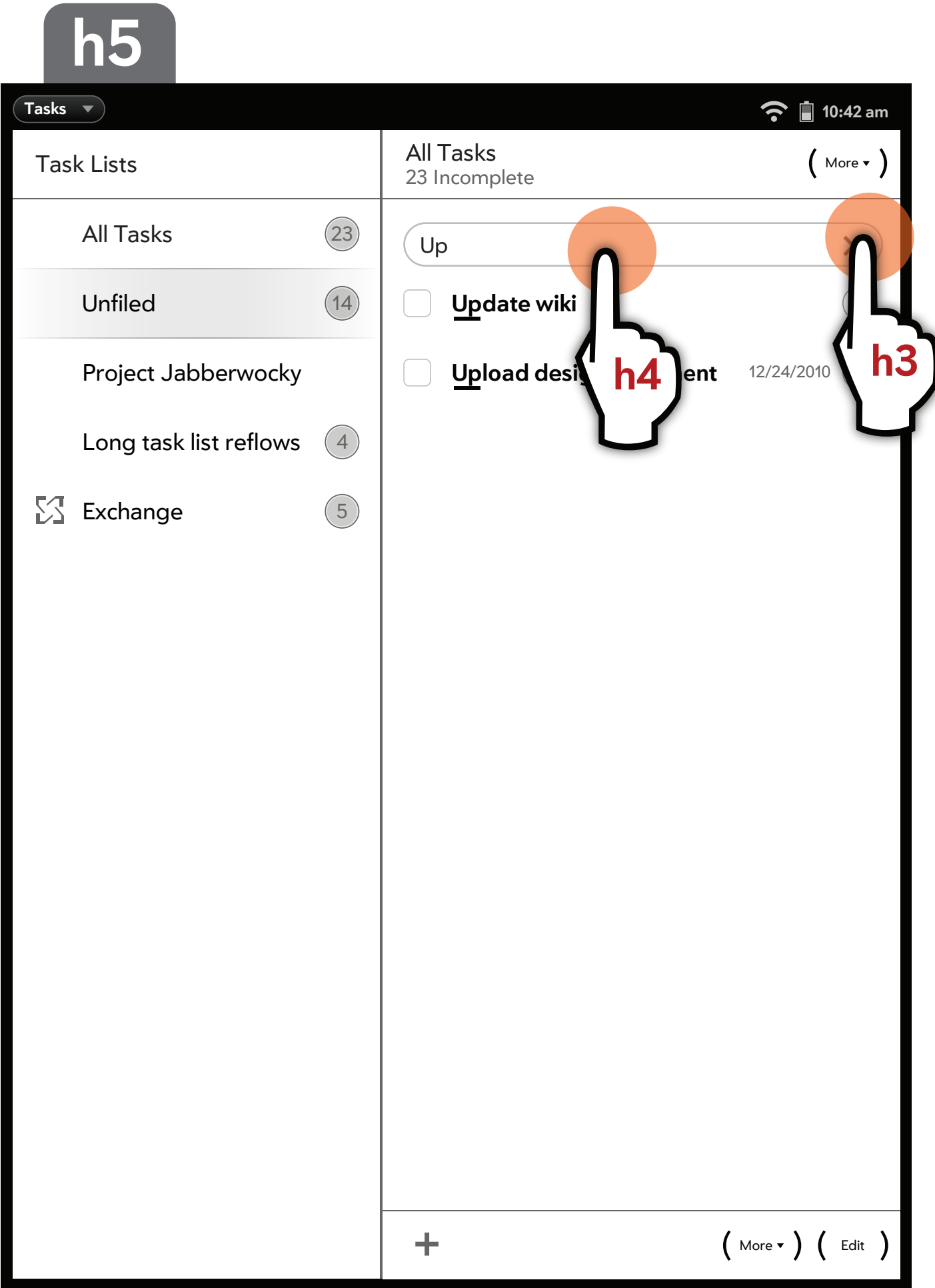
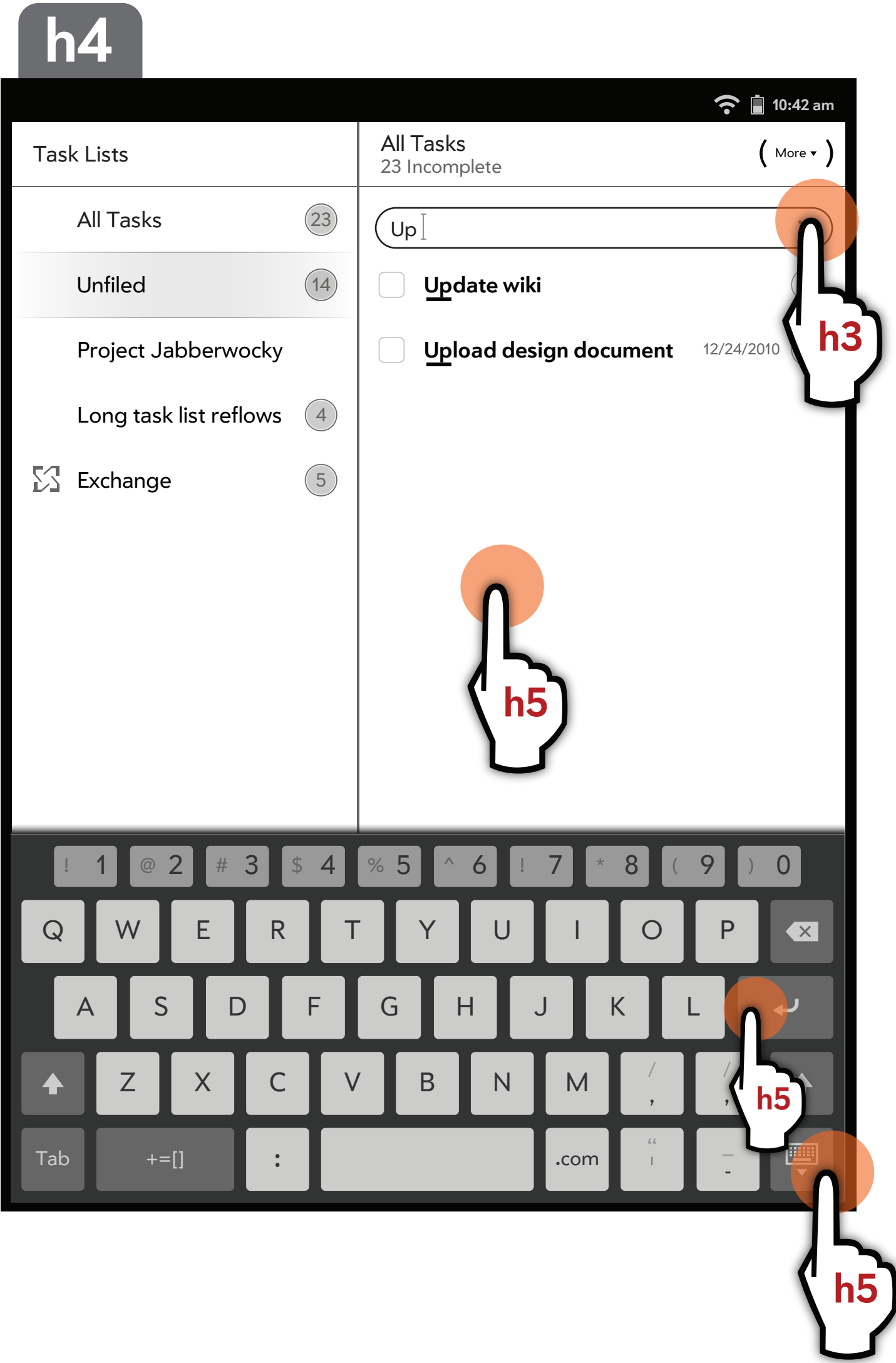
Flow demonstrates an in-list filtered search on a tablet device.

h2. Tapping in the Search Input field focuses the field. Invoke virtual keyboard if it exists.

h3. The field is focused and a cursor appears. Hint text still appears.

- Search mode exits if:
- Keyboard dismissed
 - Tap outside search field or keyboard
 - Tap Search icon

EXAMPLE TASK FLOW: SEARCH FILTER (CONT.)



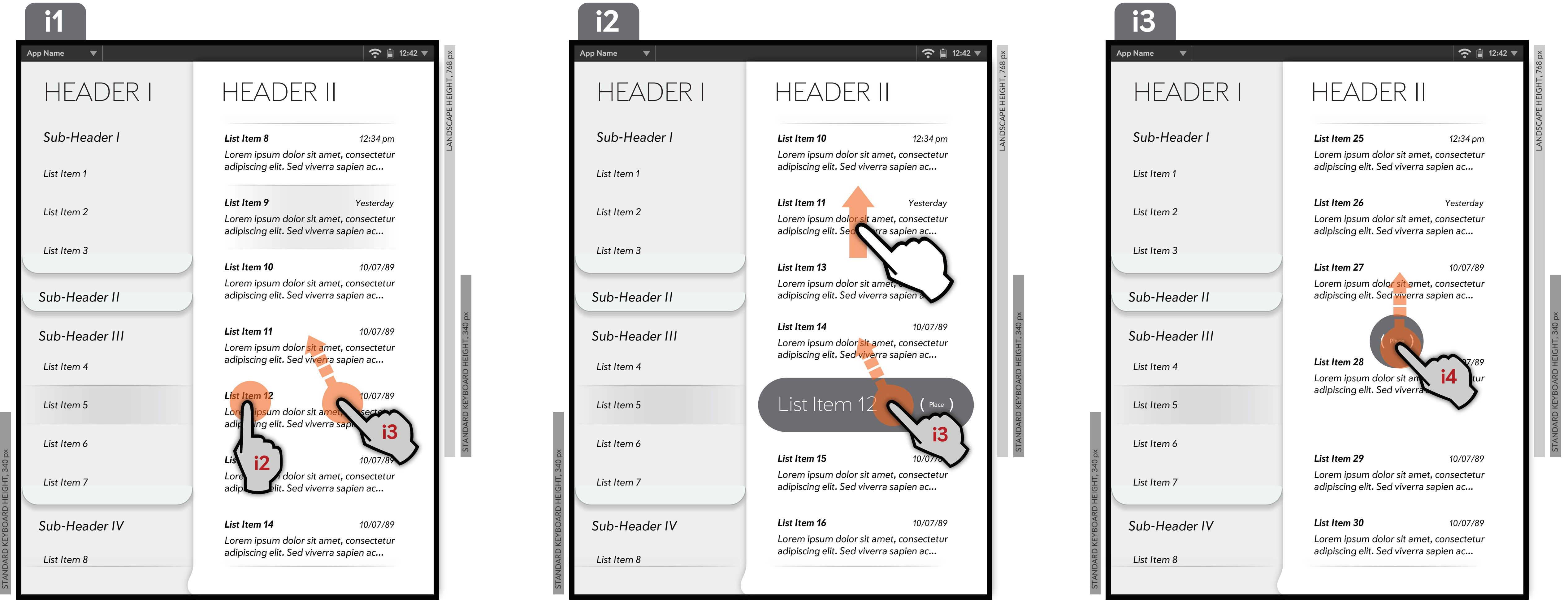
For more details on search, please refer to the corresponding documentation: [Search_2012-08-23.pdf](#)

h4. As text is entered, filtered results appear. Search icon is replaced by a clear icon ("x"). Tapping in an empty area dismisses the keyboard.

To clear the search and show the full list, backspace or tap the clear icon.

h5. Dismissing the keyboard removes focus from the field. Results are still present.

EXAMPLE TASK FLOW: REORDERING IN A BASIC LIST*



i1. User presses and holds a movable item in a grid. If user releases, go to **i2**. If user drags, go to **i3**.

i2. Item animates into a transitory state and anchors into place. Display hint arrows for 3 seconds. User can scroll the list (item remains stationary) until the desired position is reached.

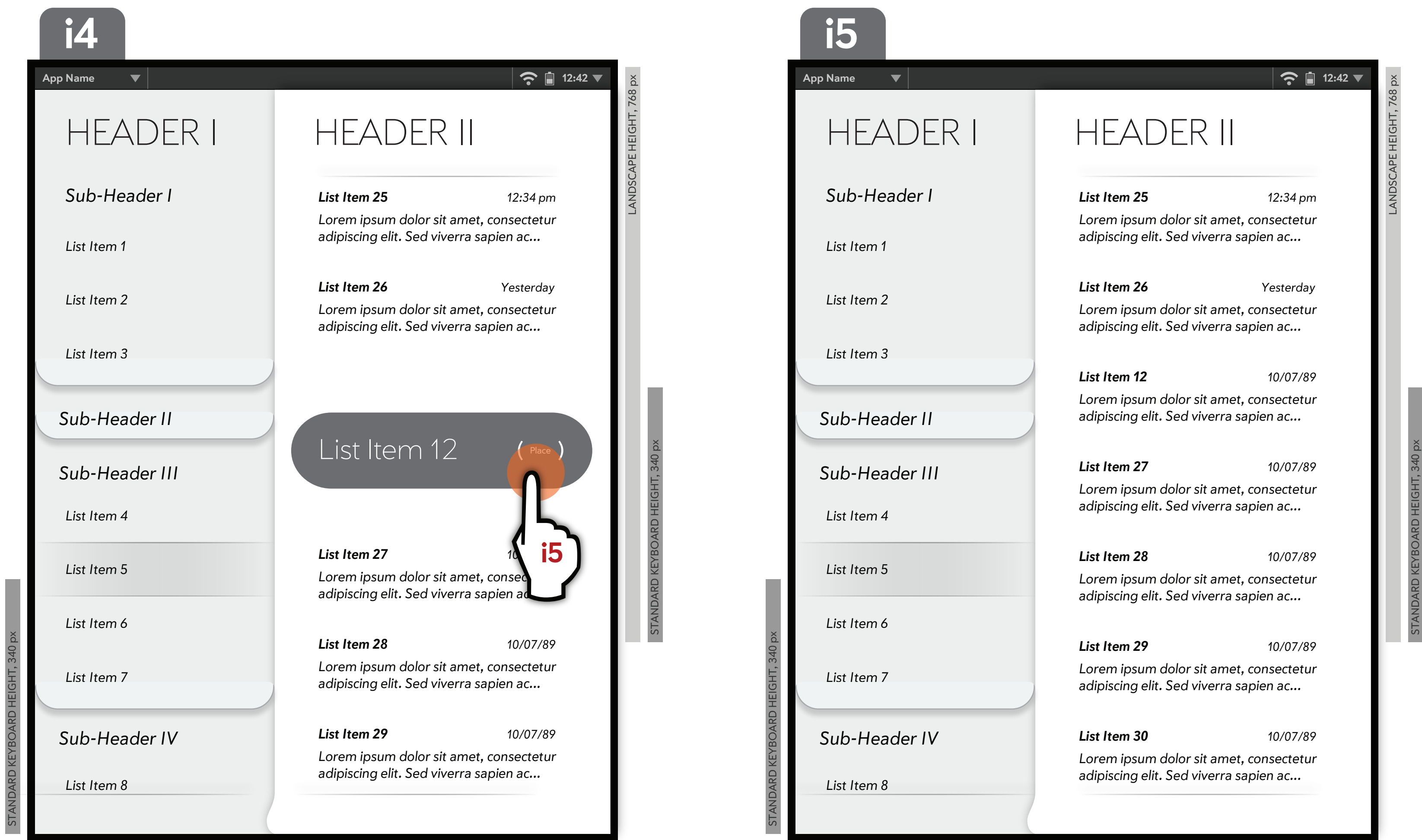
User can also drag the item and move it.

Visual design and animation TBD.

i3. If the user drags on the item itself, the item animates into a little circle as it's moved. Adjacent list items should spread apart to make room for it. Upon release, the item should expand again into its initial transitory state (as in **i2** and **i4**).

Visual design and animation TBD.

EXAMPLE TASK FLOW: REORDERING IN A BASIC LIST (CONT.) *



i4. User taps “Place” button.

i5. Transitory state is exited and item is placed in that position on the list.

EXAMPLE TASK FLOW: MOVING TO A LIST WITH COLLAPSIBLE DIVIDERS*



j1. User presses and holds a movable item in a grid. If user releases, go to j2. If user drags, go to j3.

Visual design and animation TBD.

j2. Item animates into a transitory state and anchors into place.

User can also drag the item and move it.

Visual design and animation TBD.

j3. Once the item is dragged it animates into a smaller object. If the target list contains drawers, etc. the item remains a smaller object. The original list closes behind it.

If the item is over a closed drawer, the target text/object should be highlighted. If it is over list items, adjacent list items should spread apart.

EXAMPLE TASK FLOW: MOVING TO A LIST WITH COLLAPSIBLE DIVS (CONT.)*

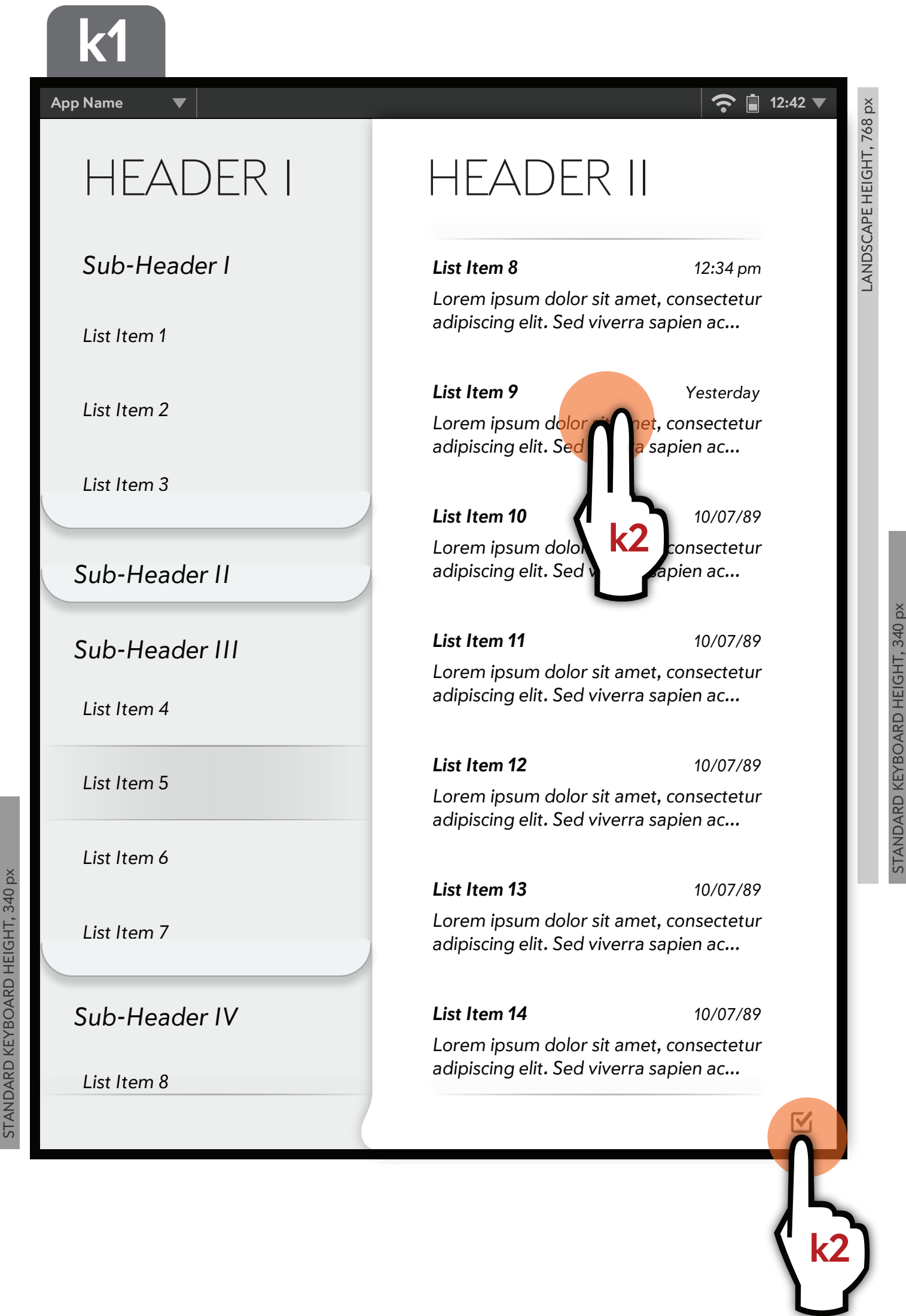


j4. When the item is released, display hint arrows for 3 seconds. User can scroll the list (item remains stationary) until the desired position is reached.

j5. User taps “Place” button.

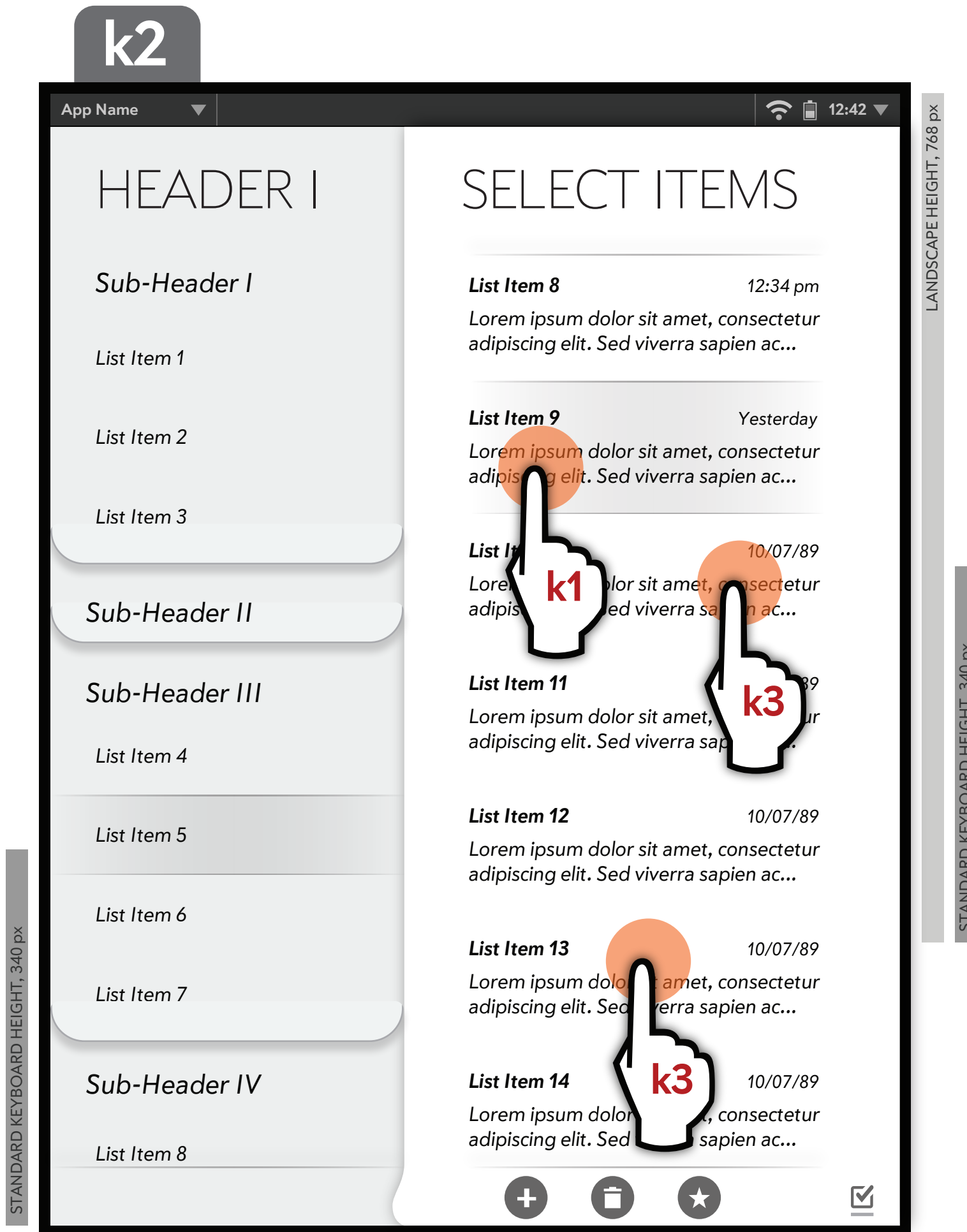
j6. Transitory state is exited and item is placed.

MULTI-SELECT



k1. Multi-select mode can be entered in one of two ways:

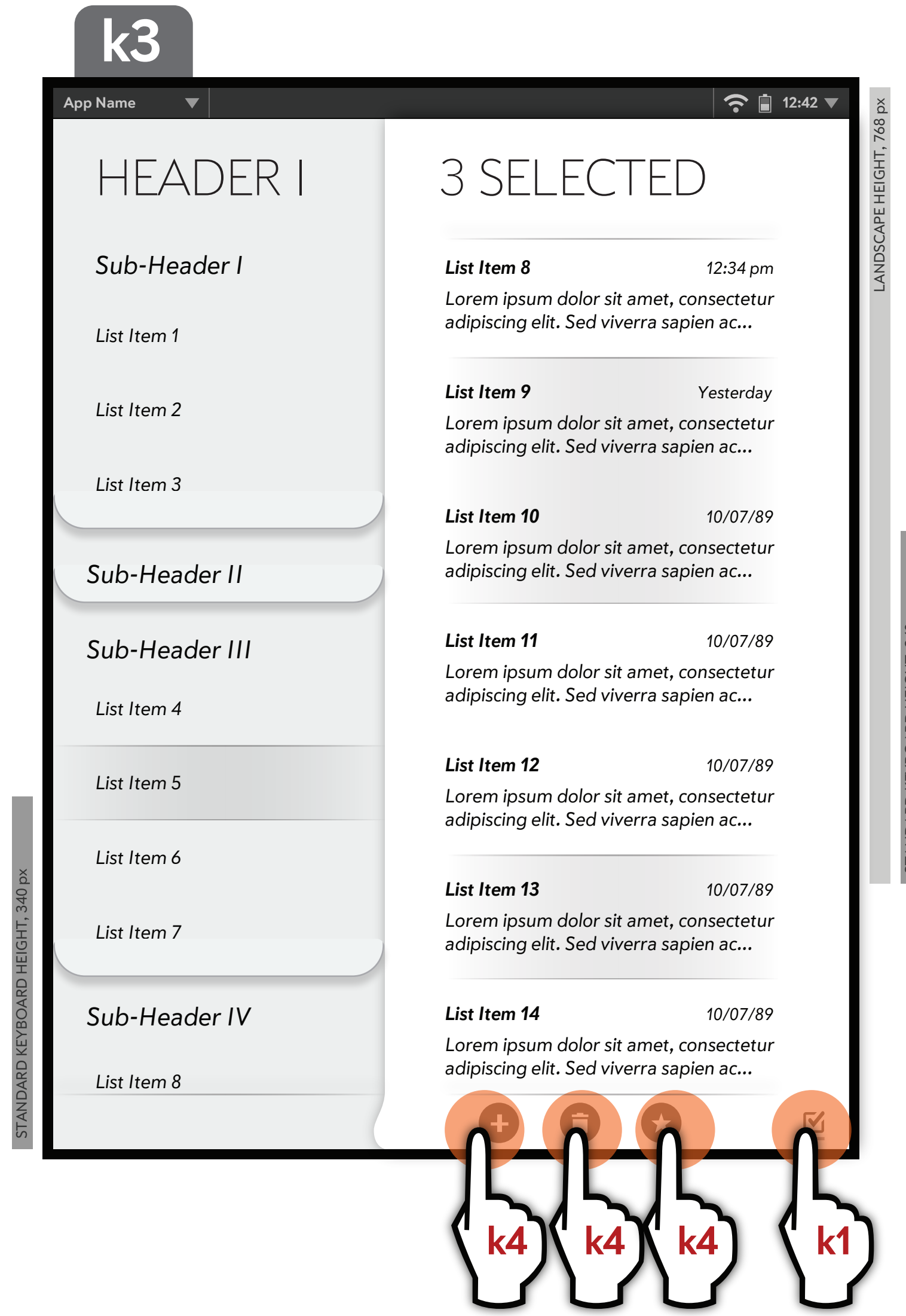
- Tap “multiselect” icon
- Two finger tap any item(s); item(s) will be pre-selected when mode is entered.



k2. List goes into a transitional mode (item is preselected if 2 finger tap is used). “Multiselect” icon should have a highlighted state. Display number of selected items in header.

If user deselects all items, exit multiselect mode.

Visual design TBD.

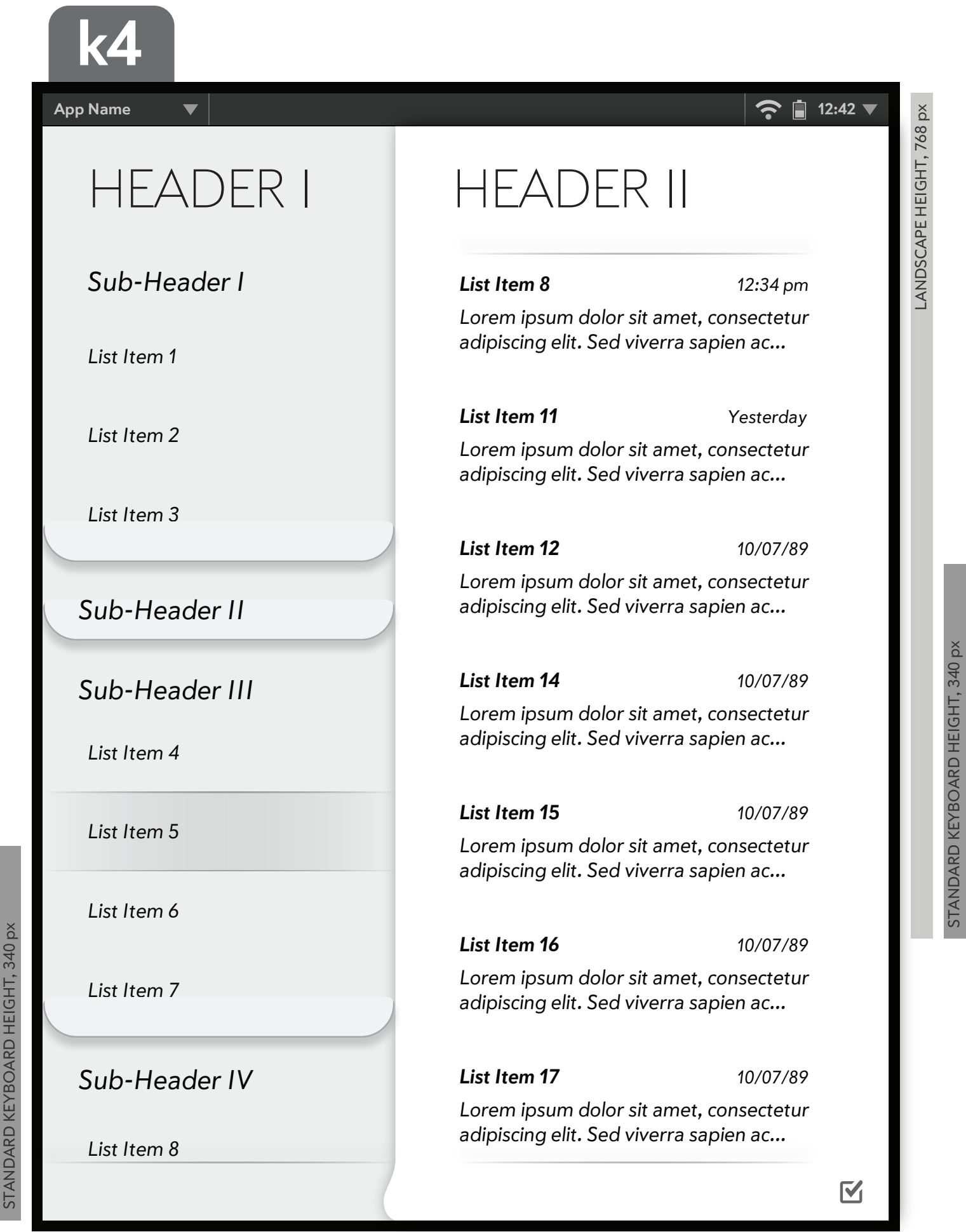


k3. There are three ways to exit multiselect mode:

- Tap the “multiselect” icon (to toggle out of the mode)
- Perform an action on selected items
- Deselect all items

Visual design and animation TBD.

MULTI-SELECT (CONT.)



k4. Mode is exited.

SWIPE-ABLE ITEMS

SINGLE ACTION

(Delete)

List Item 11

10/07/89

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...

(Delete)

MULTIPLE ACTIONS, EXAMPLE 1

Now

Later

List Item 11

10/07/89

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...

MULTIPLE ACTIONS, EXAMPLE 2

List Item 11

10/07/89

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...

Now

Later

DUAL DIRECTION, MULTIPLE ACTIONS

Later

List Item 11

10/07/89

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...

(Delete)

DUAL DIRECTION, MULTIPLE ACTIONS

Now

Later

List Item 11

10/07/89

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed viverra sapien ac...

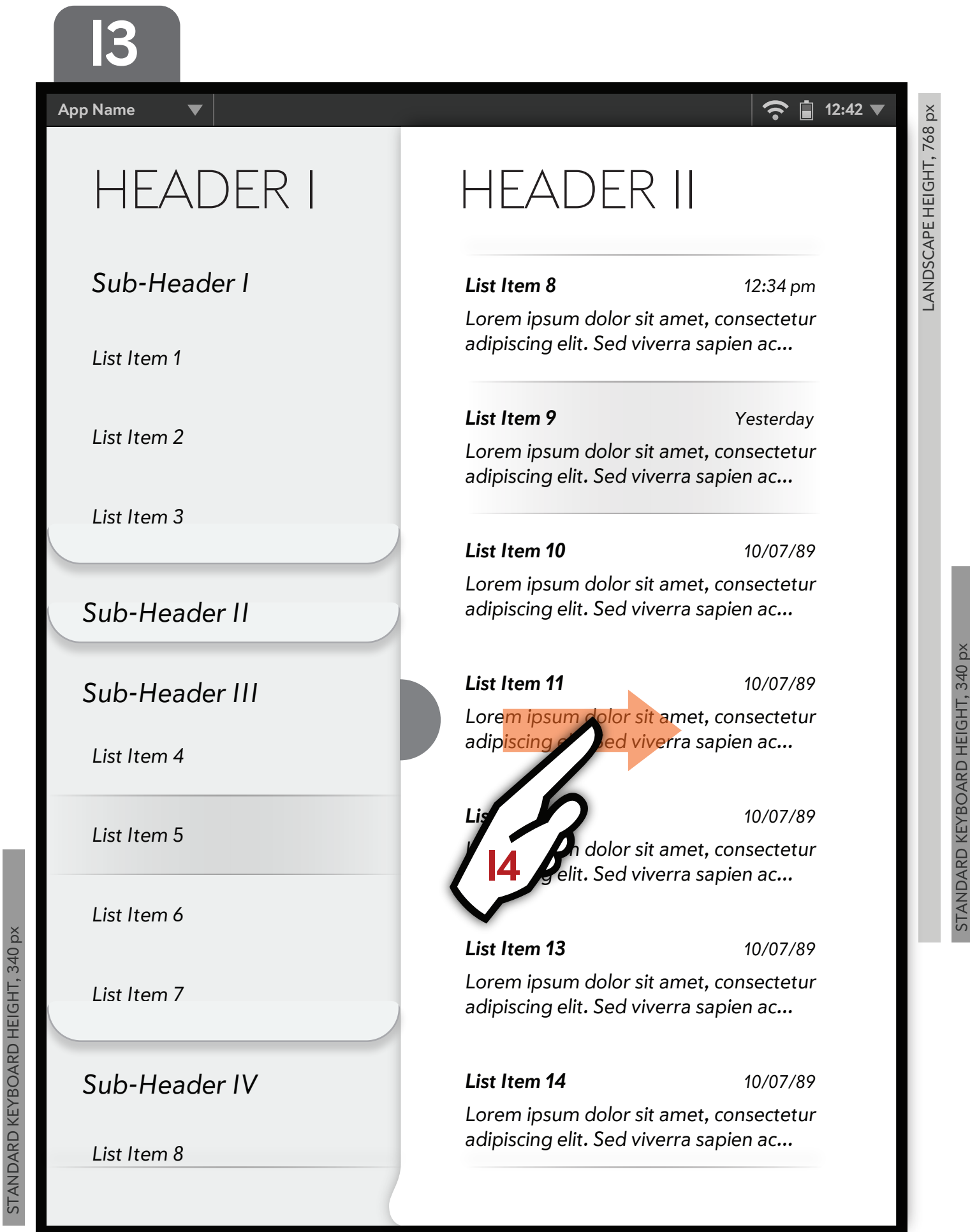
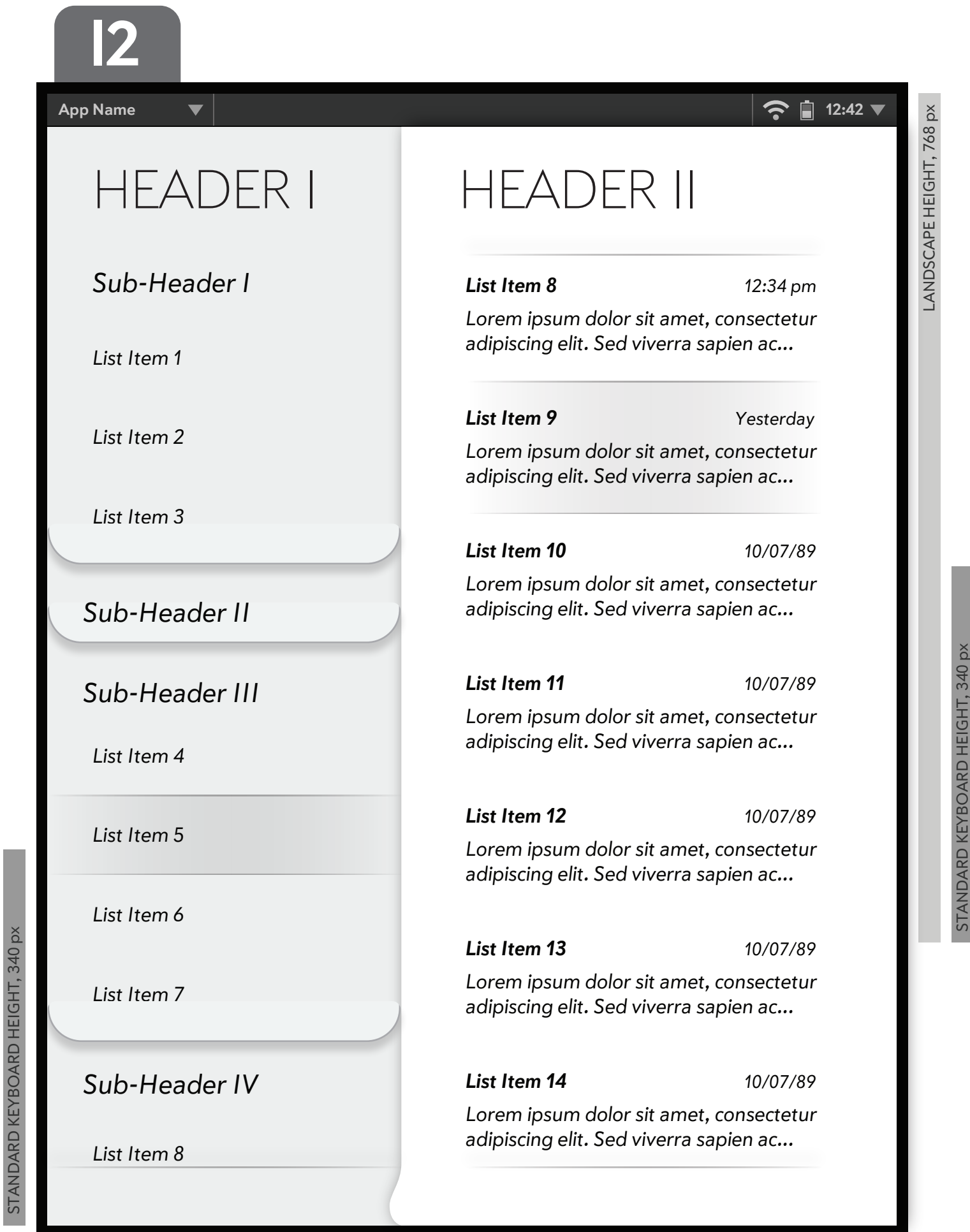
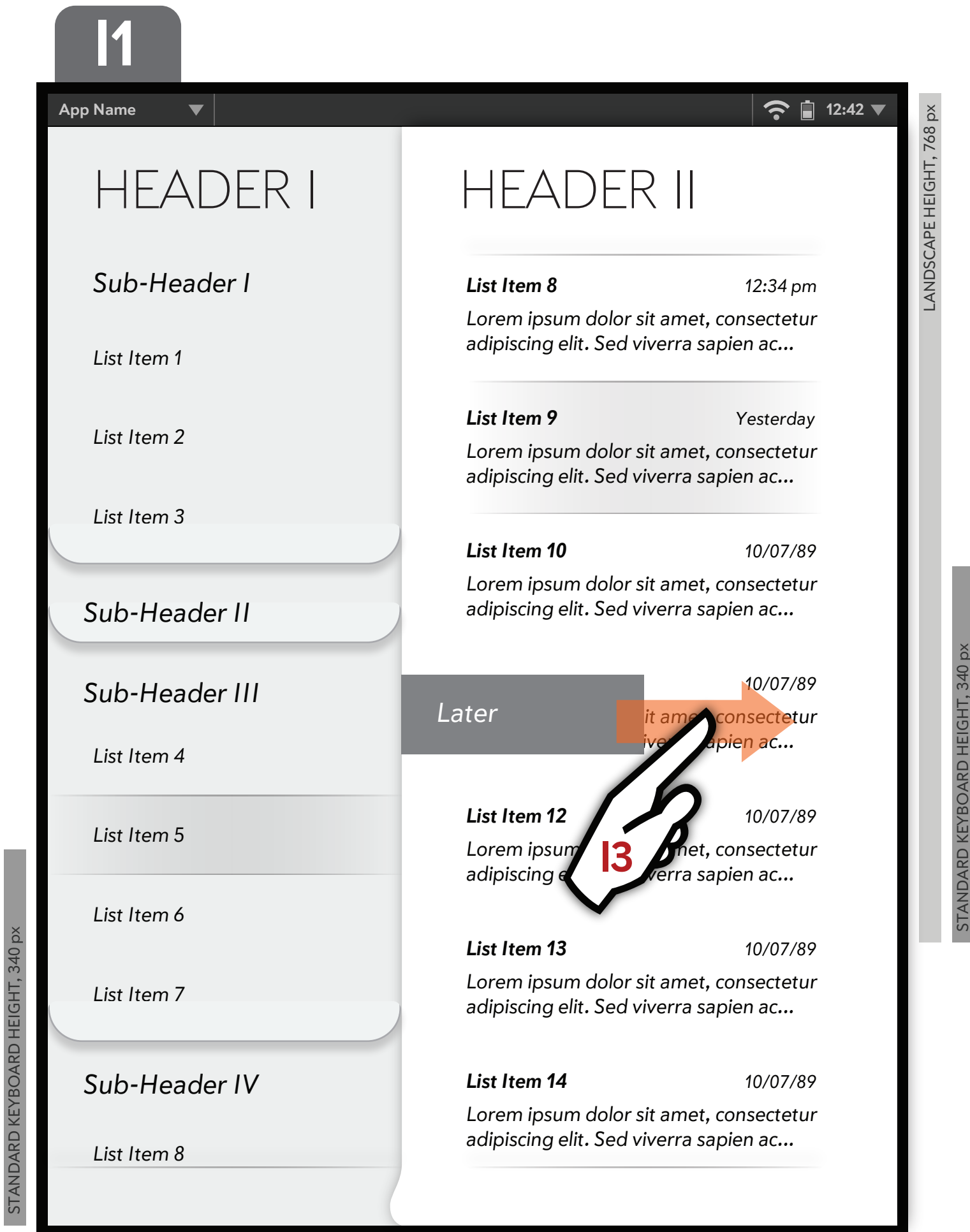
(Delete)

Swipe-able items types:

- single action--swiping left or right will result in the same action
- multiple action--each consecutive swipe has its own action
- dual direction--swiping left performs one action, right performs another

The examples at right illustrate what would happen if the user were to swipe right or left on a swipe-able list item. Please note that when swiped, color blocks sit on top of the original list item. Please refer to the visual design specs for more information.

EXAMPLE TASK FLOW: SWIPE-ABLE ITEM



11. User swipes right on a list item (or drags right to a certain threshold).

12. Display the color strip momentarily then animate to a half circle (n3).

13. Strip recedes to a half circle.
User swipes or drags right again.

EXAMPLE TASK FLOW: SWIPE-ABLE ITEM (CONT.)



- 14. Display the color strip momentarily then animate to a half circle (n5).
- 15. Strip recedes to a half circle marker. User swipes or drags left.
- 16. Item is returned to its original state.

EXAMPLE TASK FLOW: SWIPE-ABLE ITEM (CONT.)



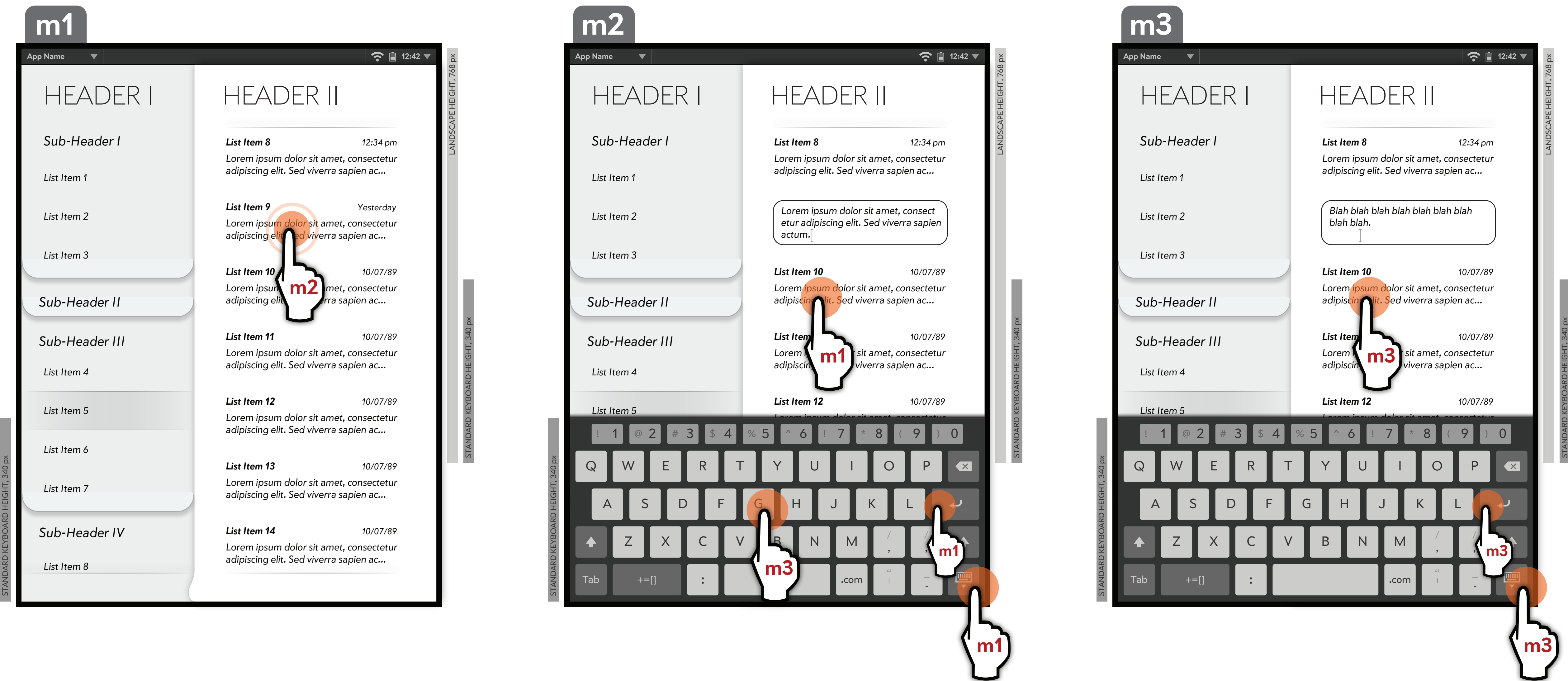
1.7. User swipes or drags left.

1.8. Persistent colored stripe appears with a confirmation button. User taps the button.

1.9. Action is performed (in this case, item is deleted).

Tapping or swiping anywhere outside of the stripe will return the item to its original state.

EXAMPLE TASK FLOW: INLINE EDITING



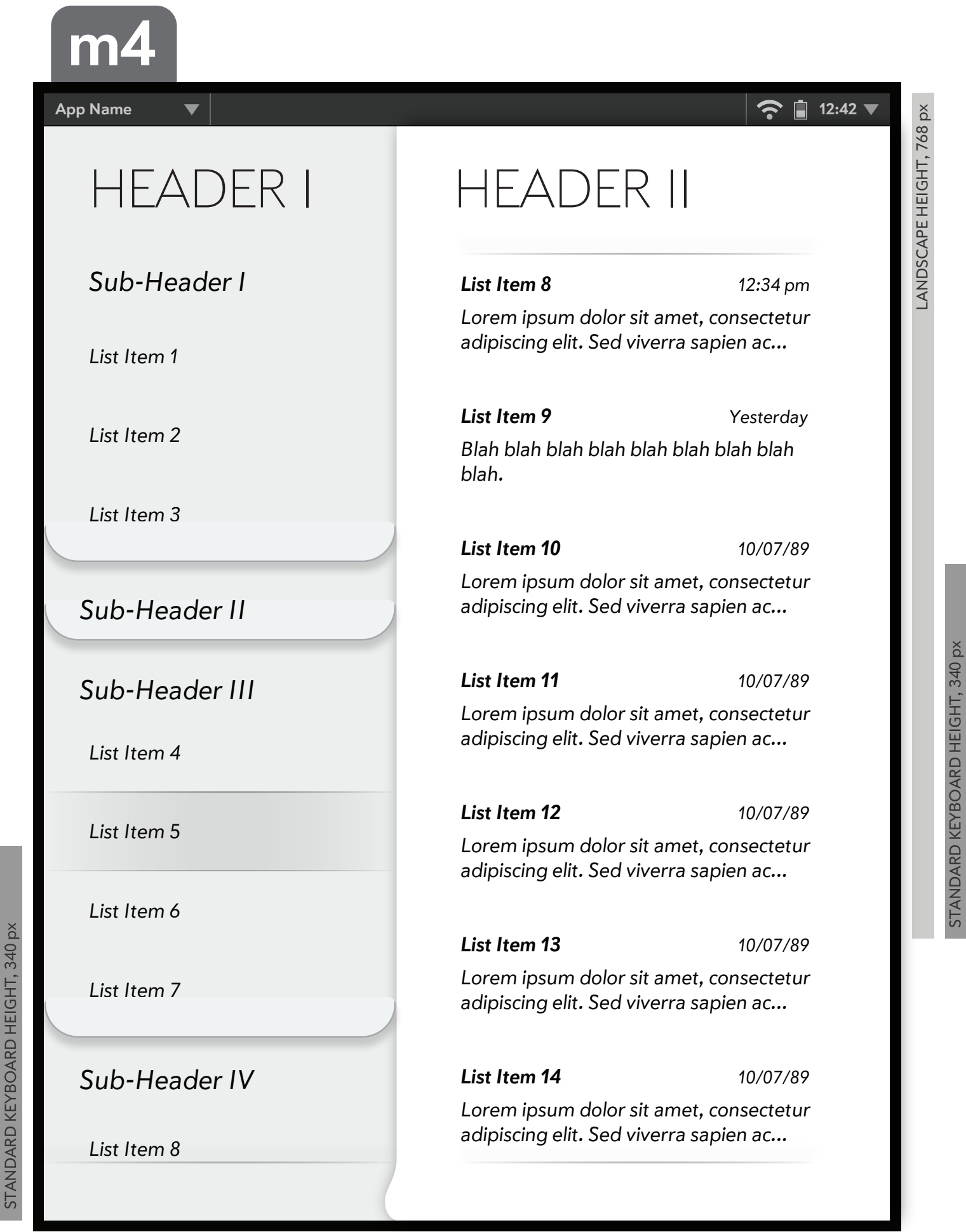
m1. User double taps an editable list item.

m2. Invoke virtual keyboard where that exists and autoscroll the list so that the item being edited is visible.

If user taps out, selects RETURN, or closes the keyboard without editing, do not make any changes.

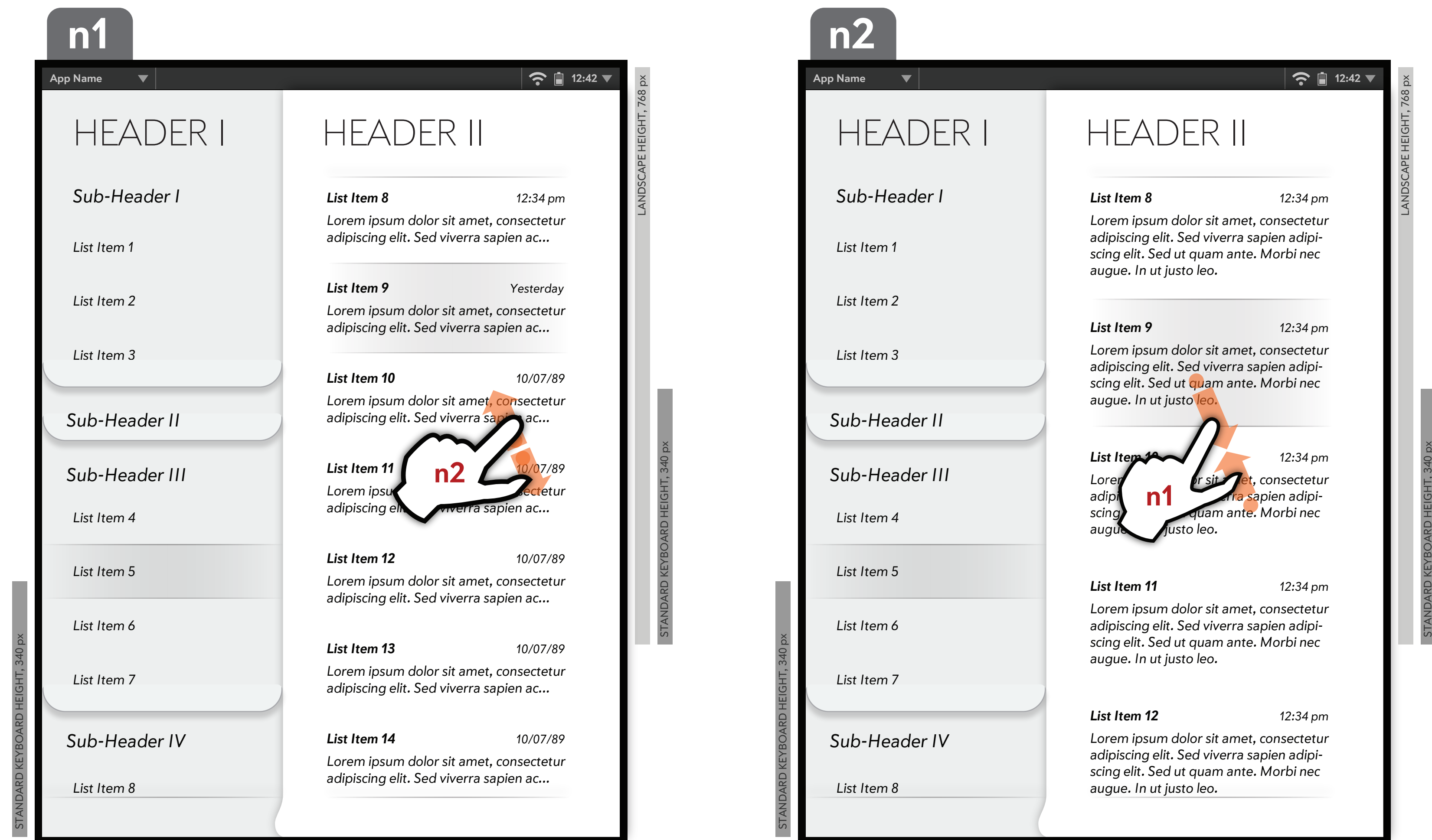
m3. If user has made a change and taps out, selects RETURN, or closes the keyboard without editing, save the changes.

EXAMPLE TASK FLOW: INLINE EDITING (CONT.)



m4. Accept changes and update list item.

EXAMPLE TASK FLOW: EXPANDABLE LIST



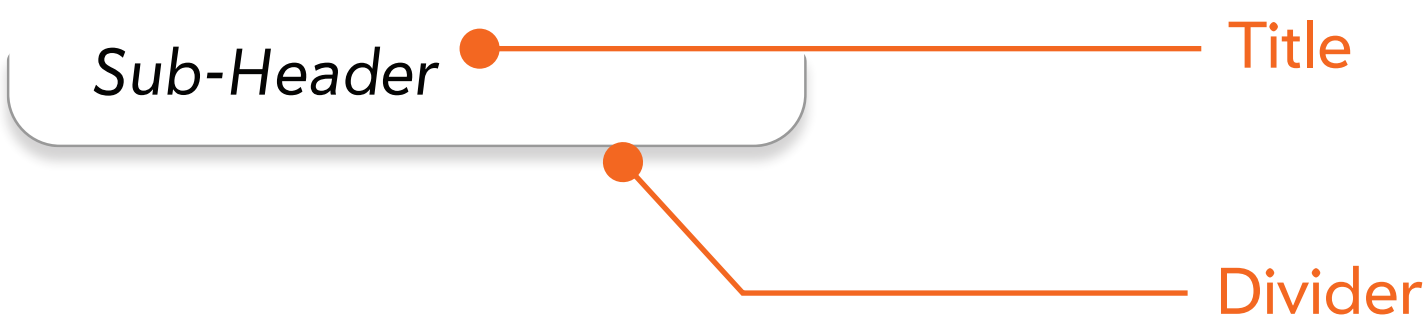
n1. Use two fingers to spread out in an expandable list. This will expand all the list items in a list so that they each show more content.

There are only two states to switch between.

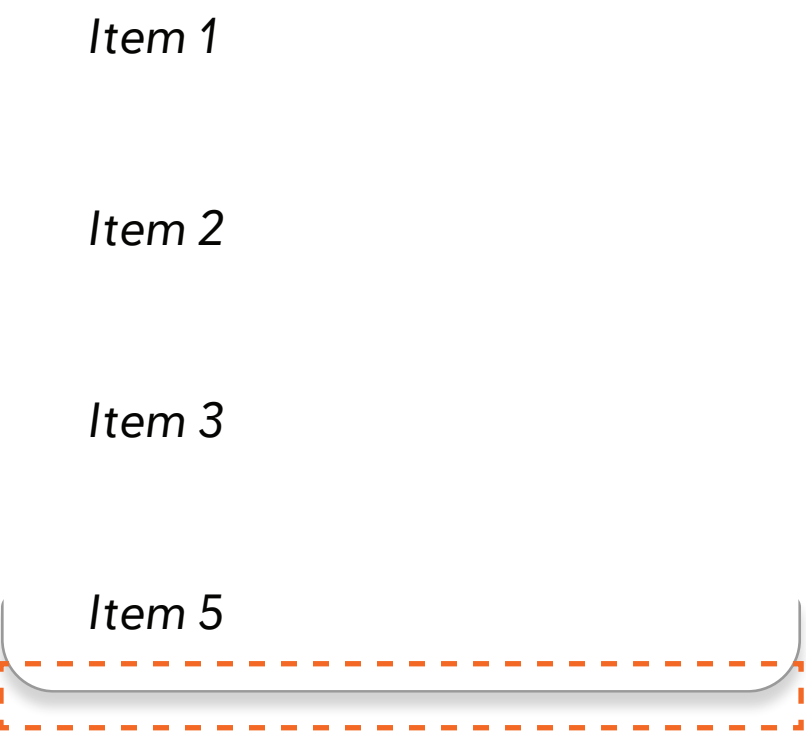
n2. Pinch to shrink all the list items in the list.

LIGHT COLLAPSIBLE DIVIDER: STRUCTURE & STATES

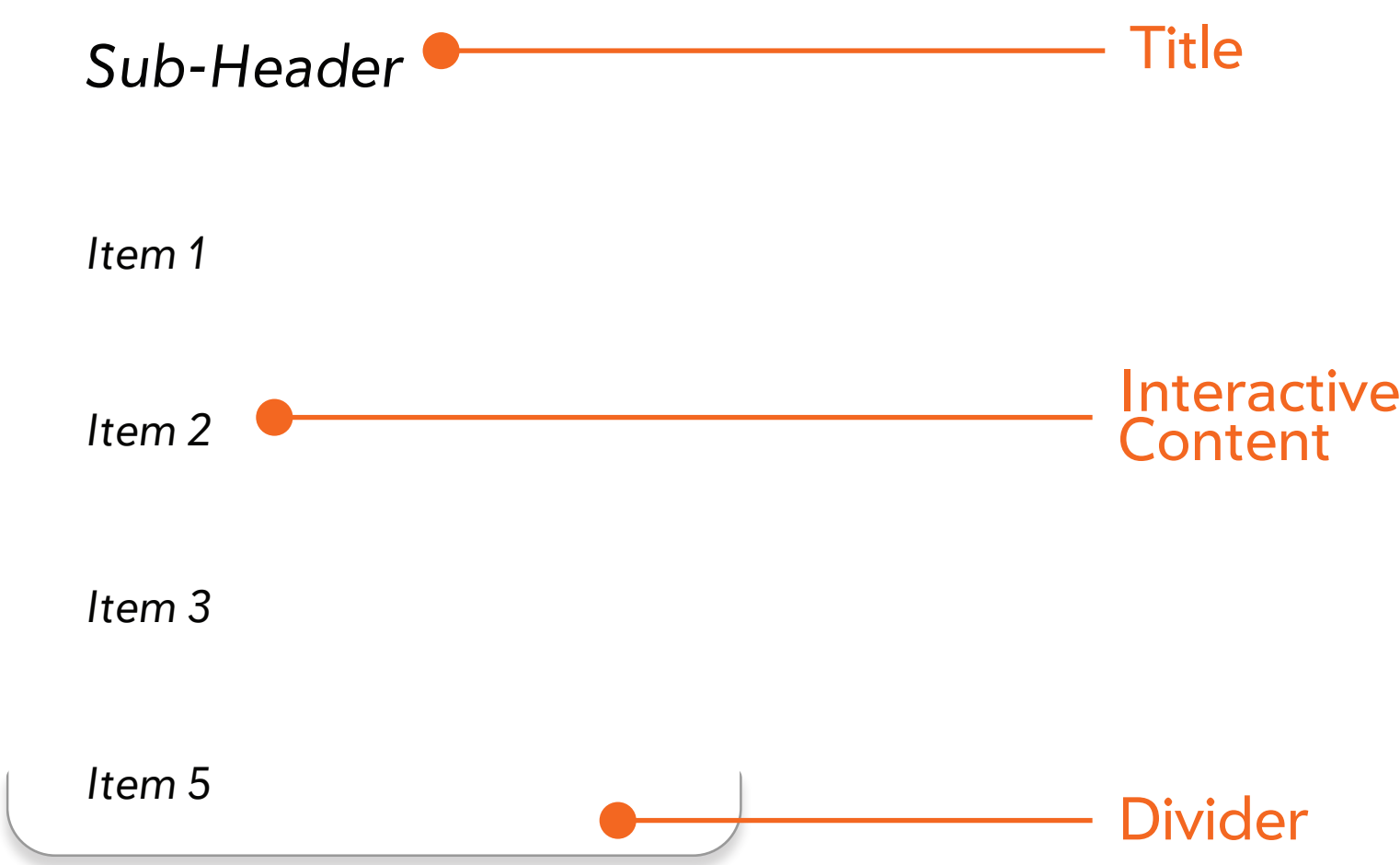
COLLAPSED



HIT AREAS



COLLAPSED



Title

Should always be tappable to toggle between expanded and collapsed states.

Content

Will vary. Can contain lists, grids and controls. Content should not scroll within the container, but the container as a whole can be scrolled (such as a list or grid).

Divider

On tap, will toggle between expanded and collapsed states. The divider is always at the below content when expanded.

Height & Width

The divider has a min-width, but no maximum. Expanded height is unlimited. See visual design redlines for width and height rules across screen sizes.

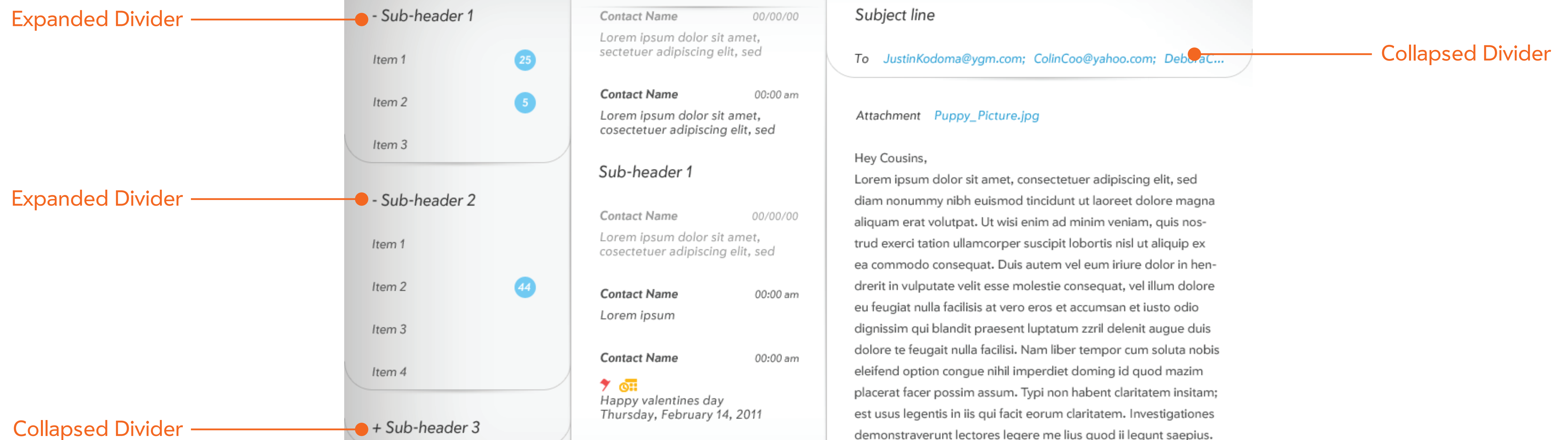
Hit Areas

Include the divider and the title. Please see above.

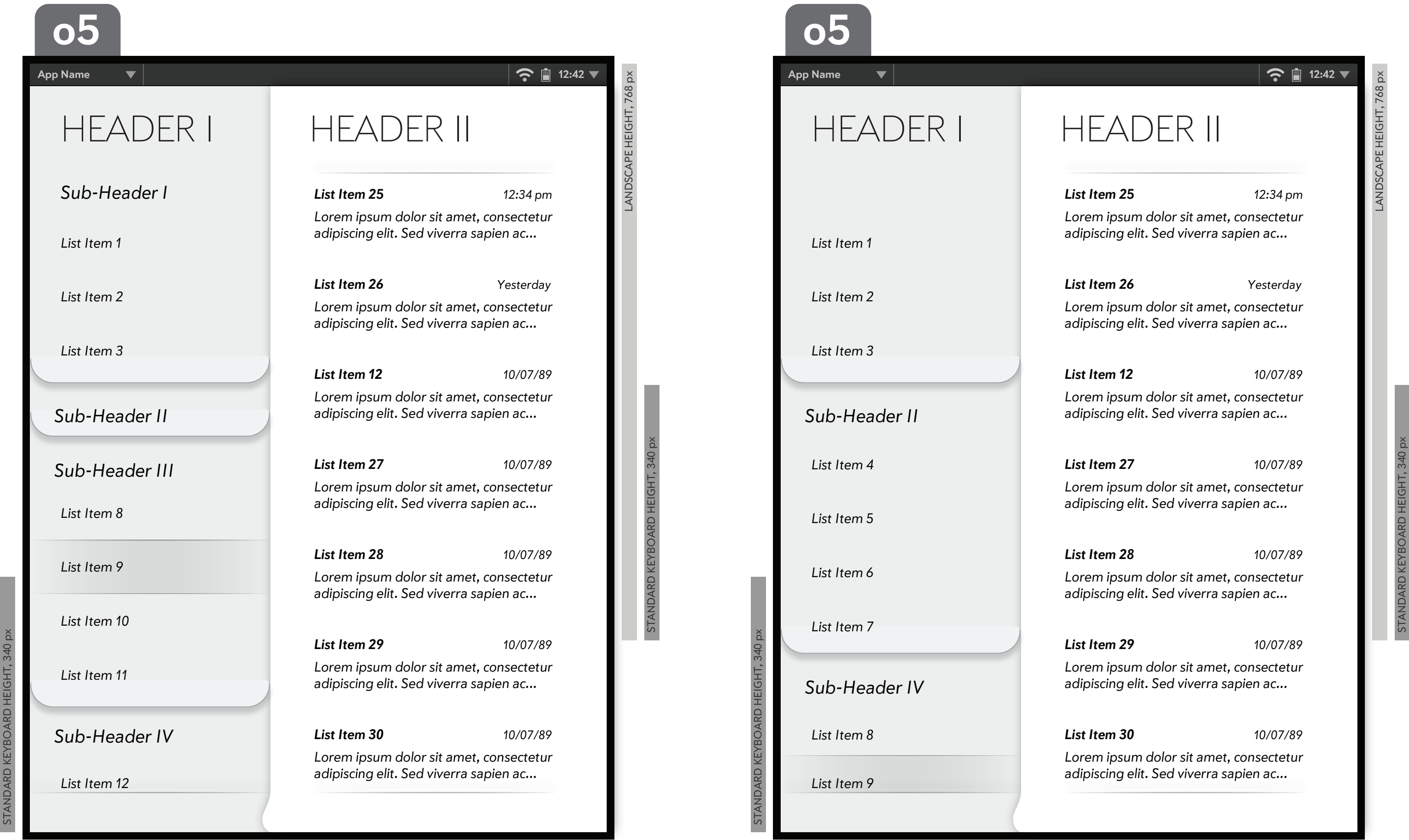
Platform Variations

None.

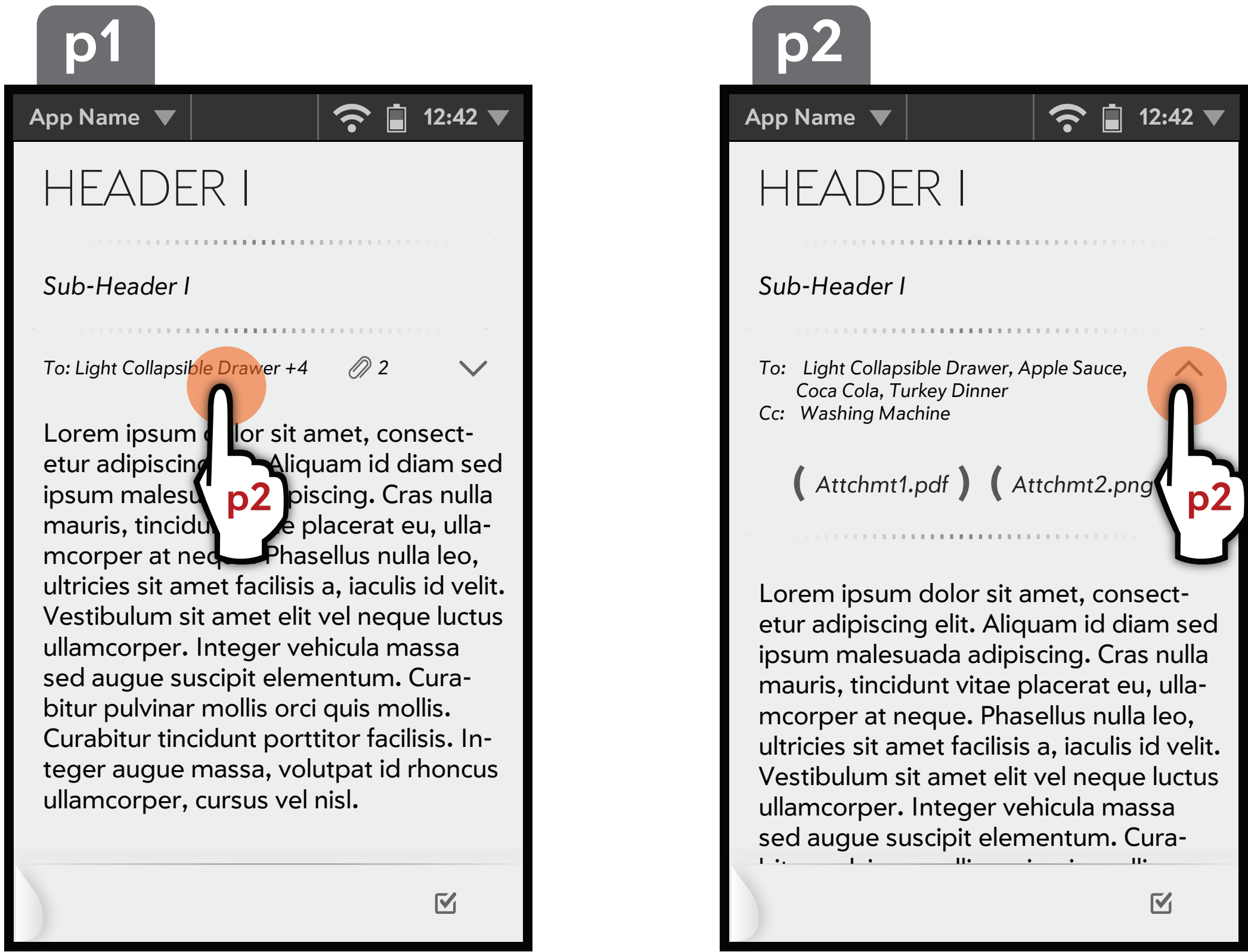
LIGHT COLLAPSIBLE DIVIDER: VISUAL DESIGN EXAMPLE



EXAMPLE TASK FLOW: LIGHT COLLAPSIBLE DIVIDER, DEFAULT



EXAMPLE TASK FLOW: LIGHT COLLAPSIBLE DIVIDER, CUSTOM



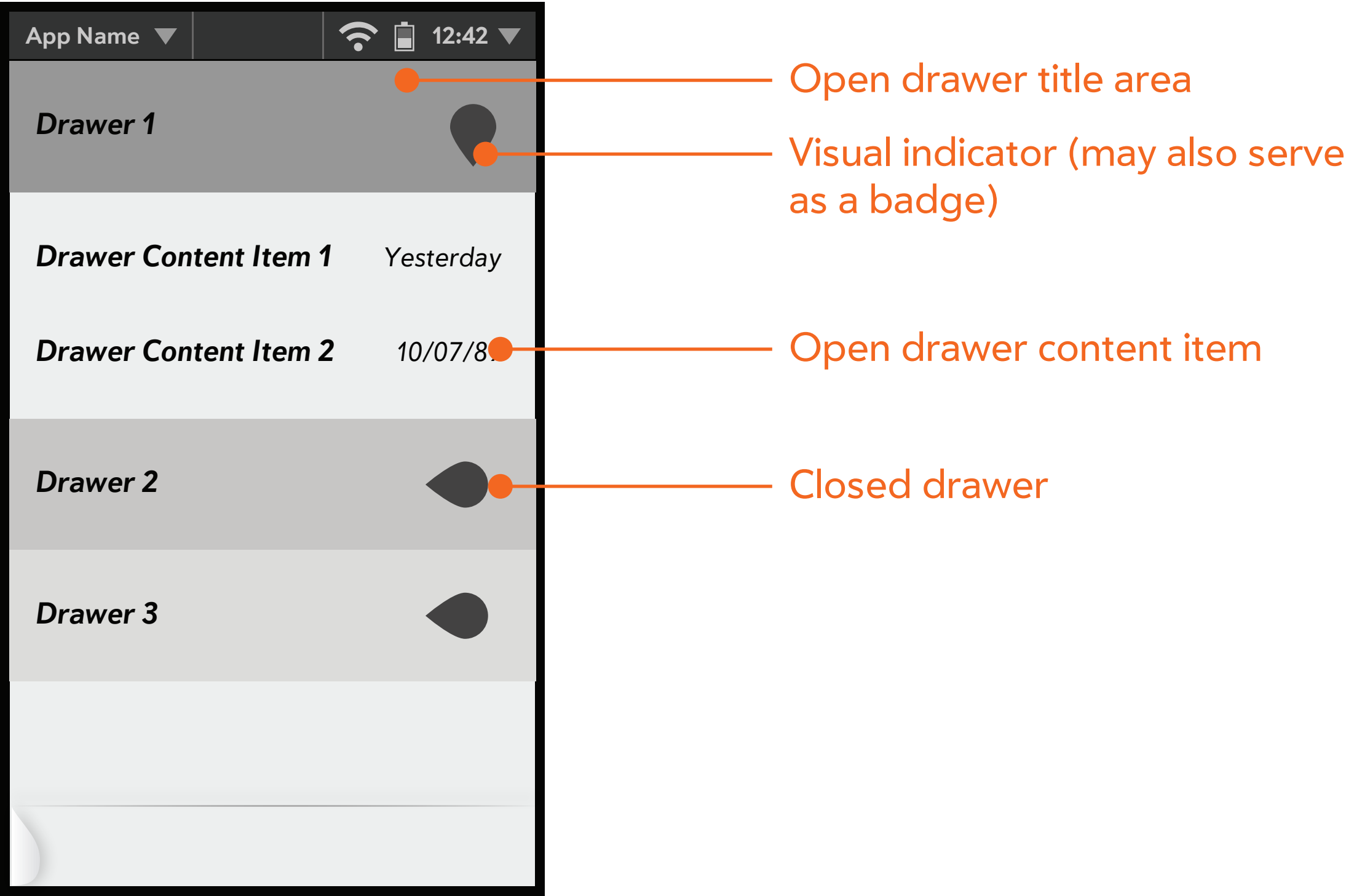
p1. User taps Collapsible Divider.

p2. Expanded content is displayed. Note that the title of the divider has changed.

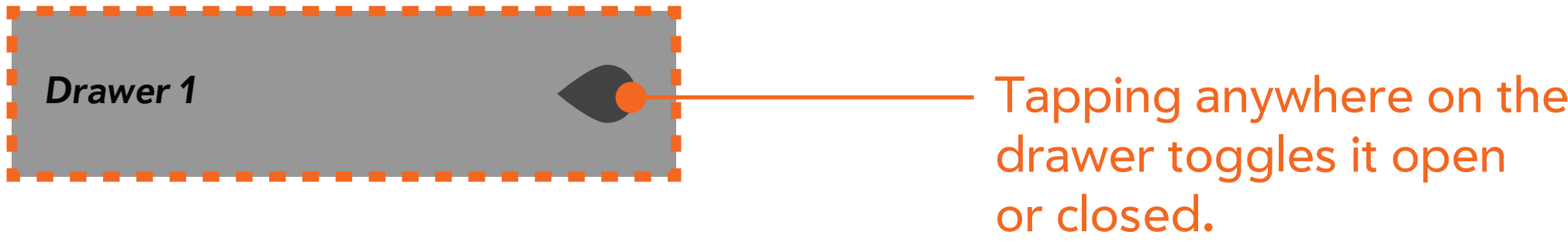
User can collapse the content by tapping on the arrow.

HEAVY COLLAPSIBLE DIVIDER WITH ARROW: STRUCTURE

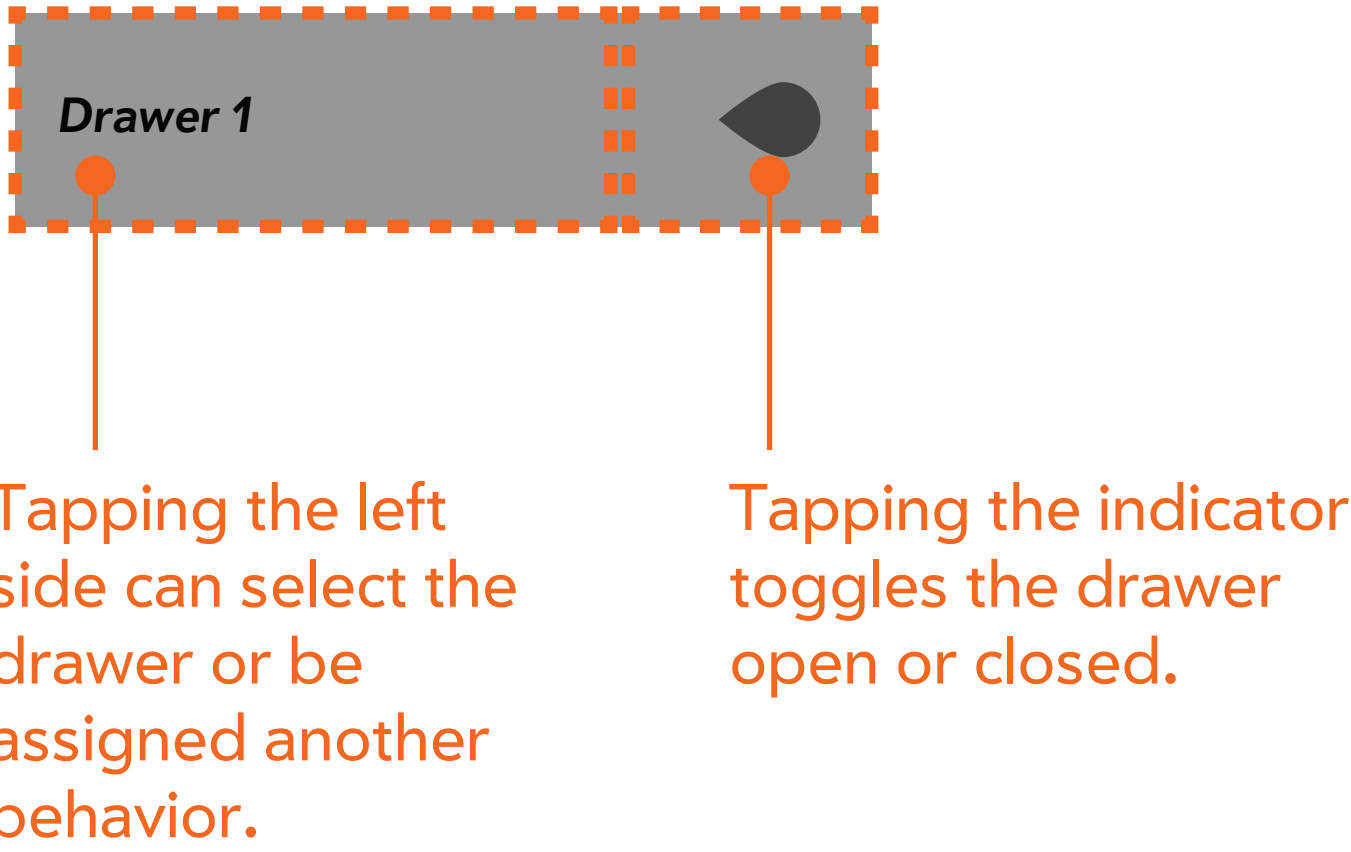
LIST OF DRAWERS



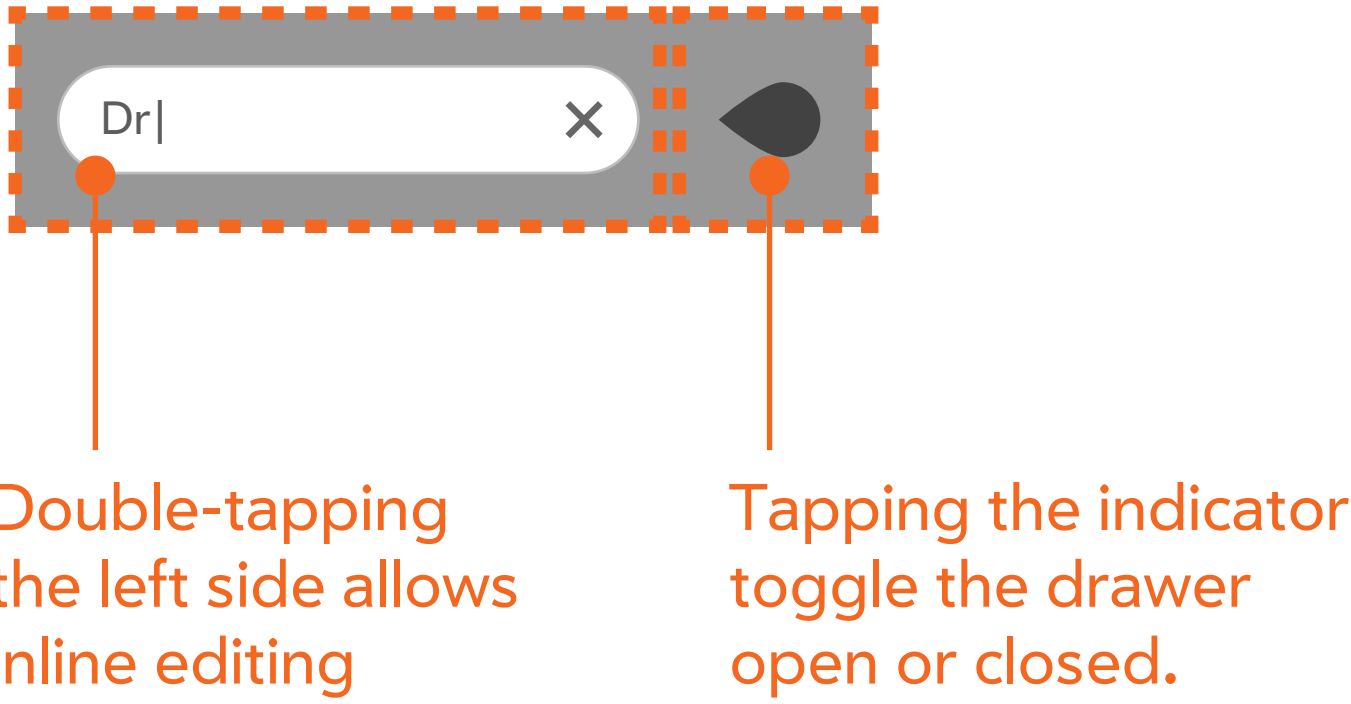
SINGLE HIT AREA



SPLIT HIT AREA

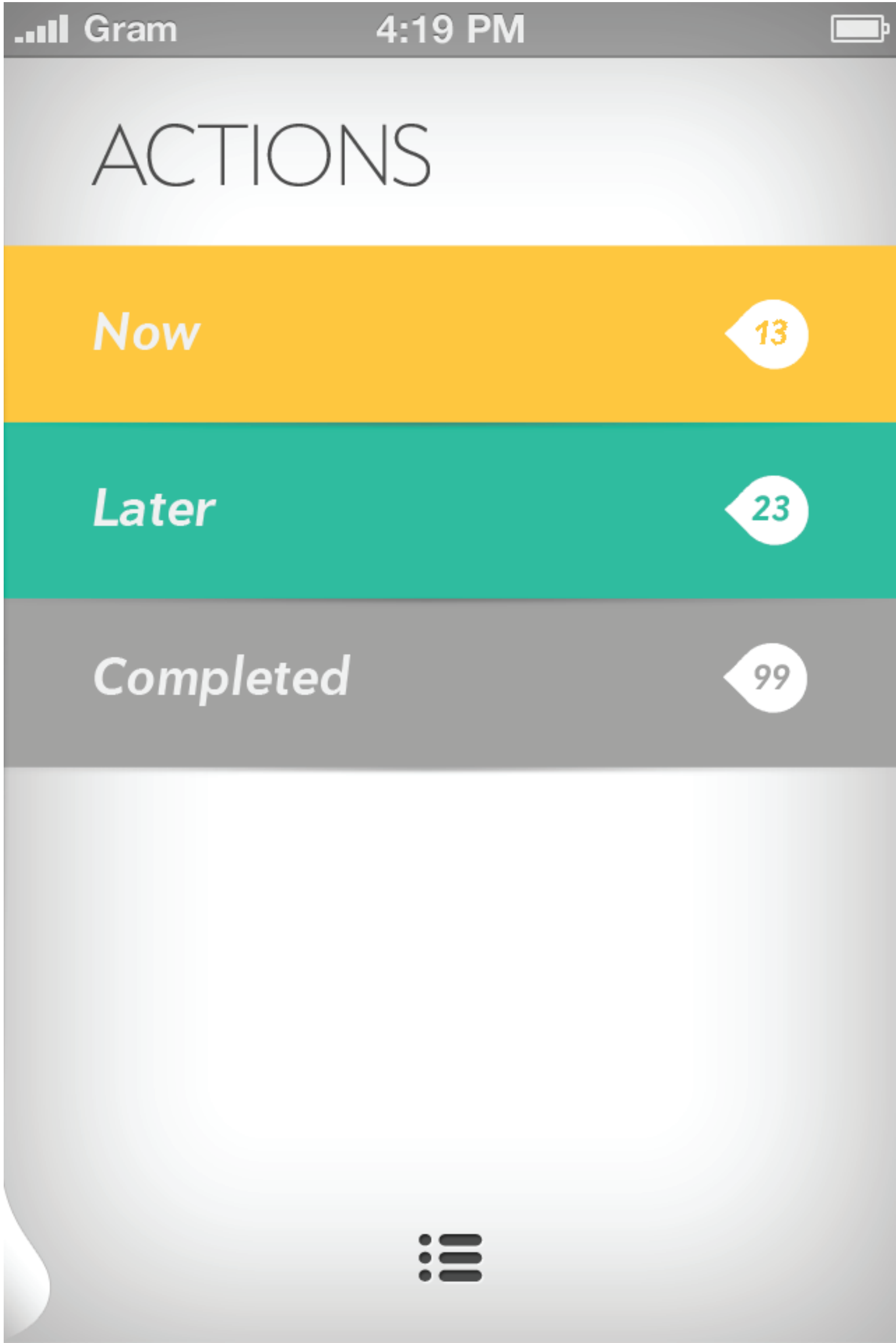


EDITABLE DRAWER ITEM

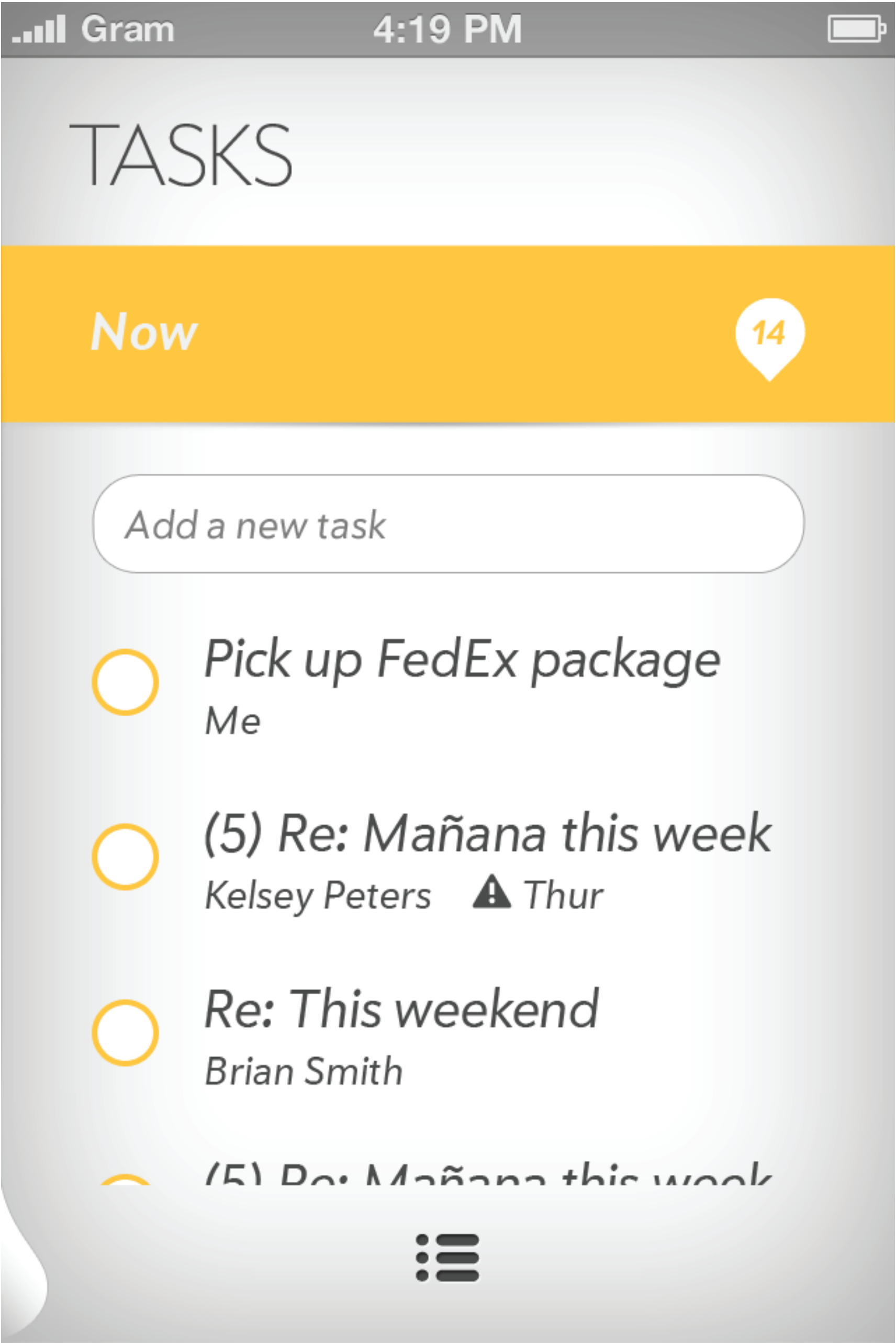


HEAVY COLLAPSIBLE DIVIDER: VISUAL DESIGN EXAMPLE

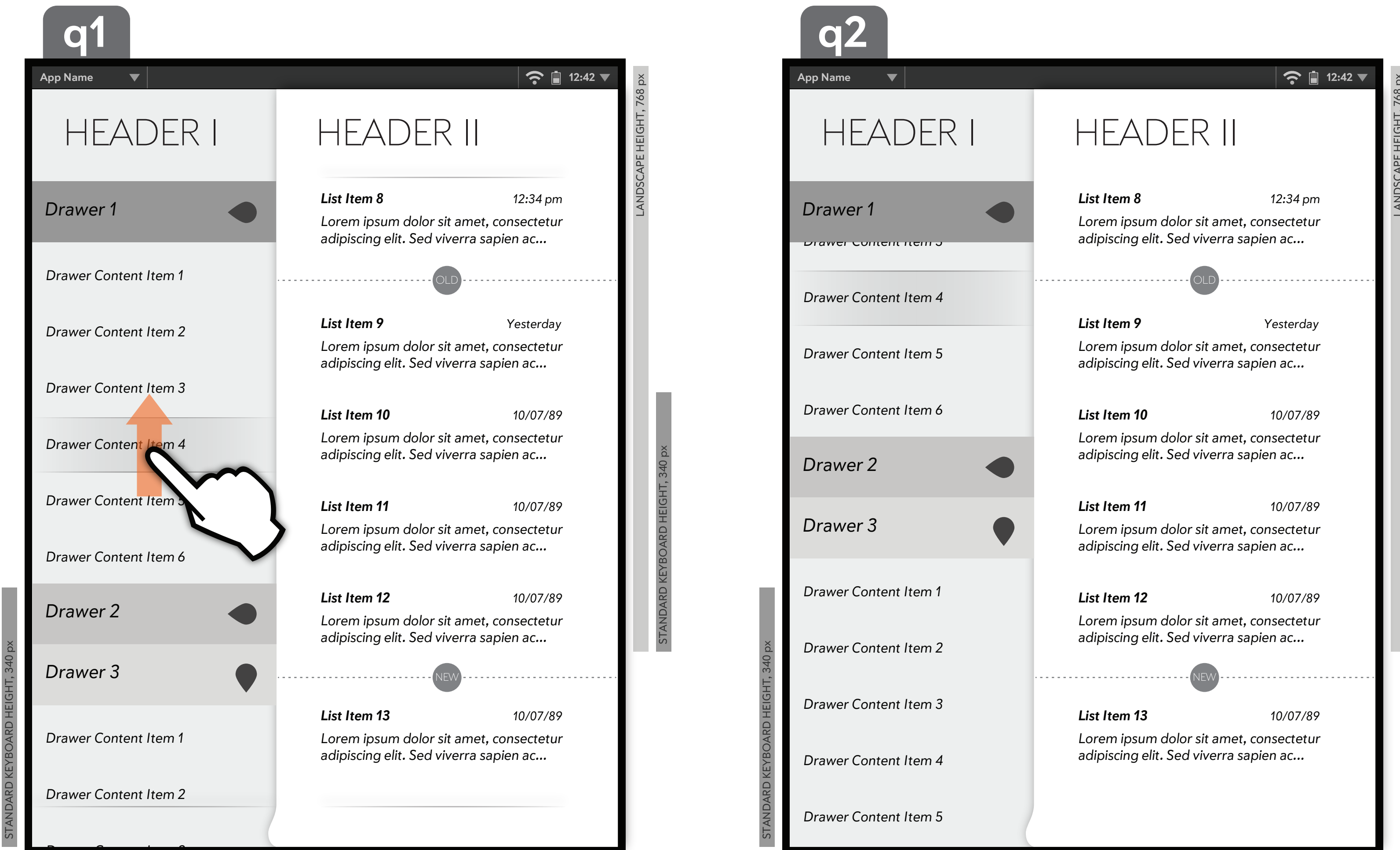
CLOSED



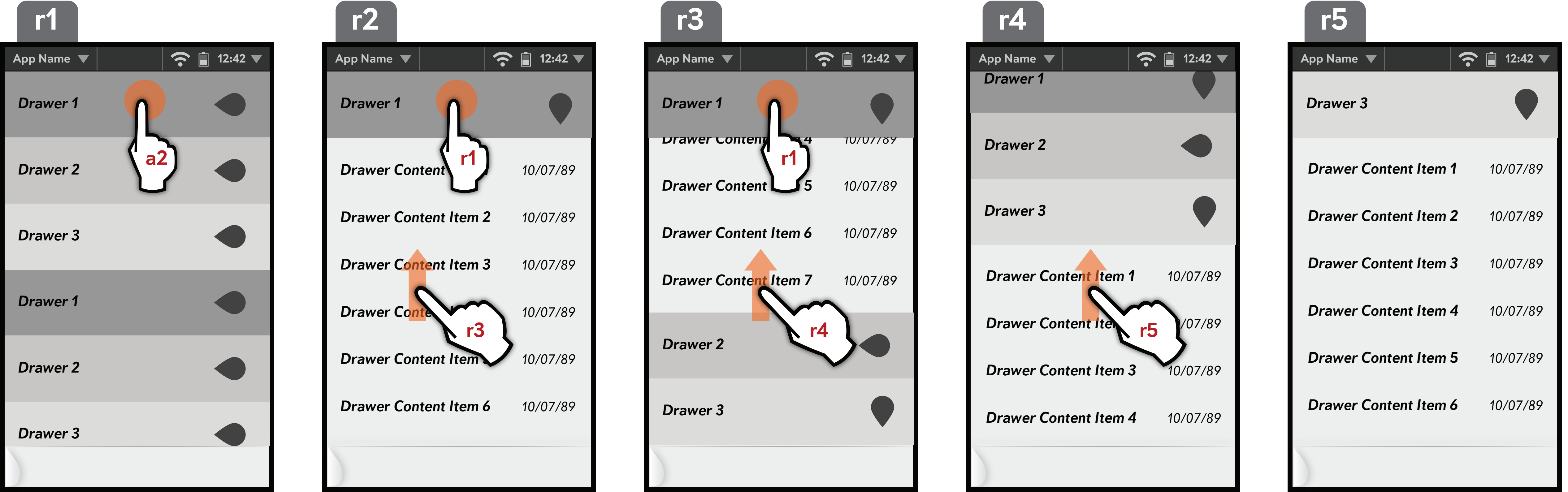
OPEN



HEAVY COLLAPSIBLE DIVIDER: TABLET VARIATION



HEAVY COLLAPSIBLE DIVIDER: BASIC FUNCTIONALITY



r1. User taps the divider to open it. Tear drop indicator rotates clockwise.

Please refer to Structure on p.37 for hit areas.

r2. Expanded divider is indicated with the tearshape pointing downwards.

When user scrolls content, the divider title stays pinned to the top.

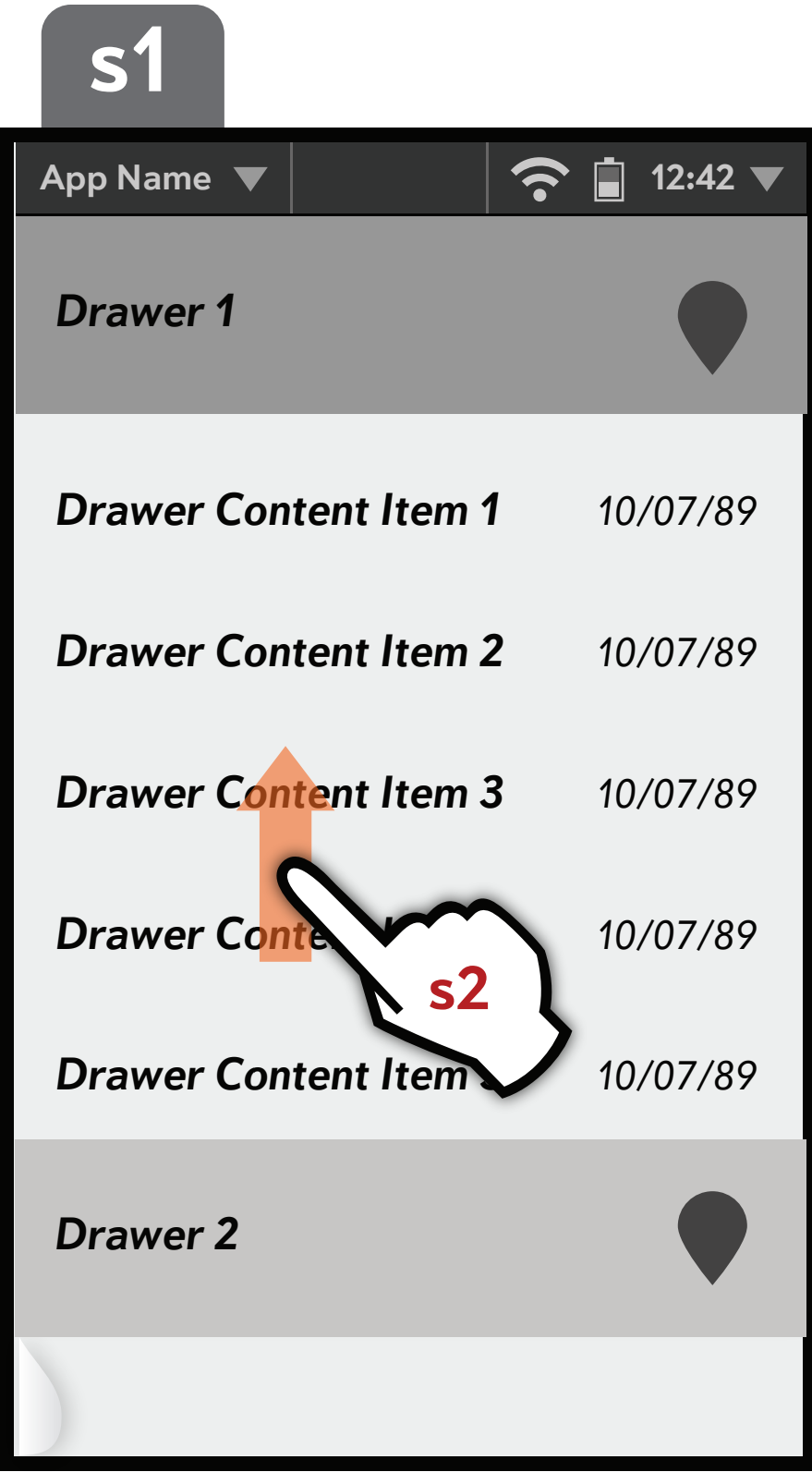
- Allow developer to choose:
- no pinning
 - top pinning (default)
 - bottom pinning
 - top & bottom pinning

r3. Expanded divider's title stays pinned to the top.

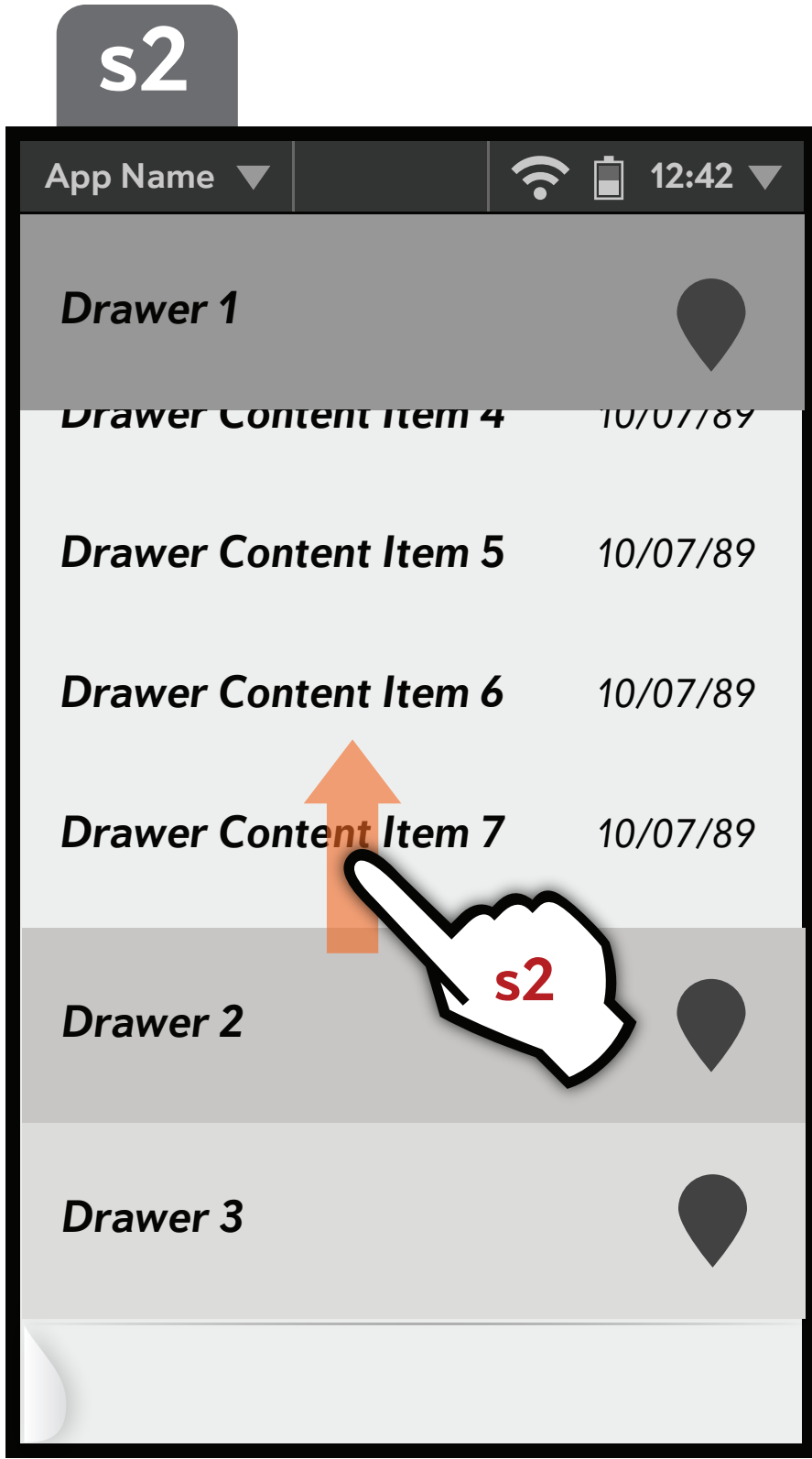
r4. The next expanded divider's title pushes other dividers out of the view area.

r5. Content scrolls until the last visible item hits the bottom of the view area. Title remains pinned to the top.

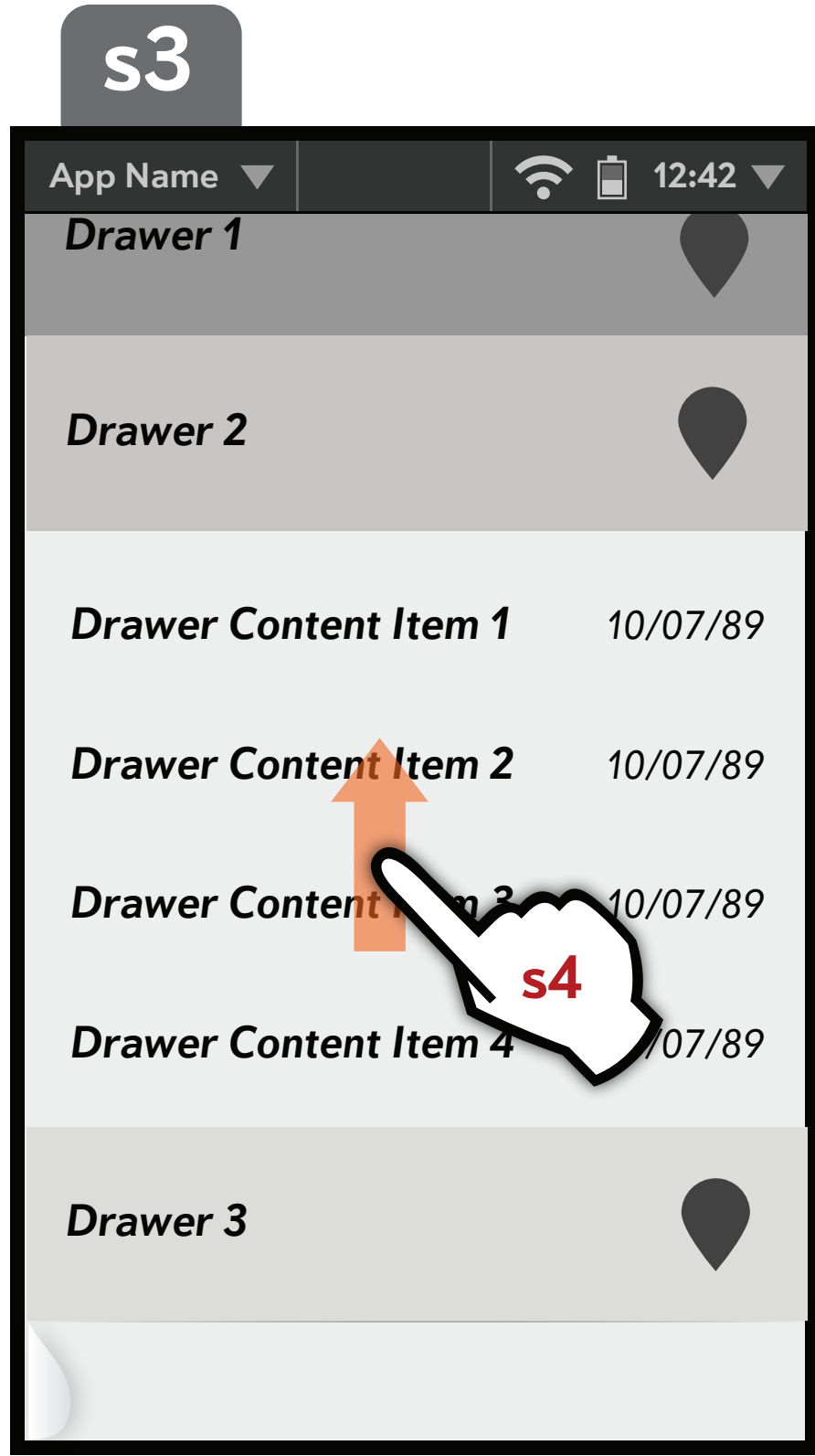
HEAVY COLLAPSIBLE DIVIDER: TOP & BOTTOM PINNING



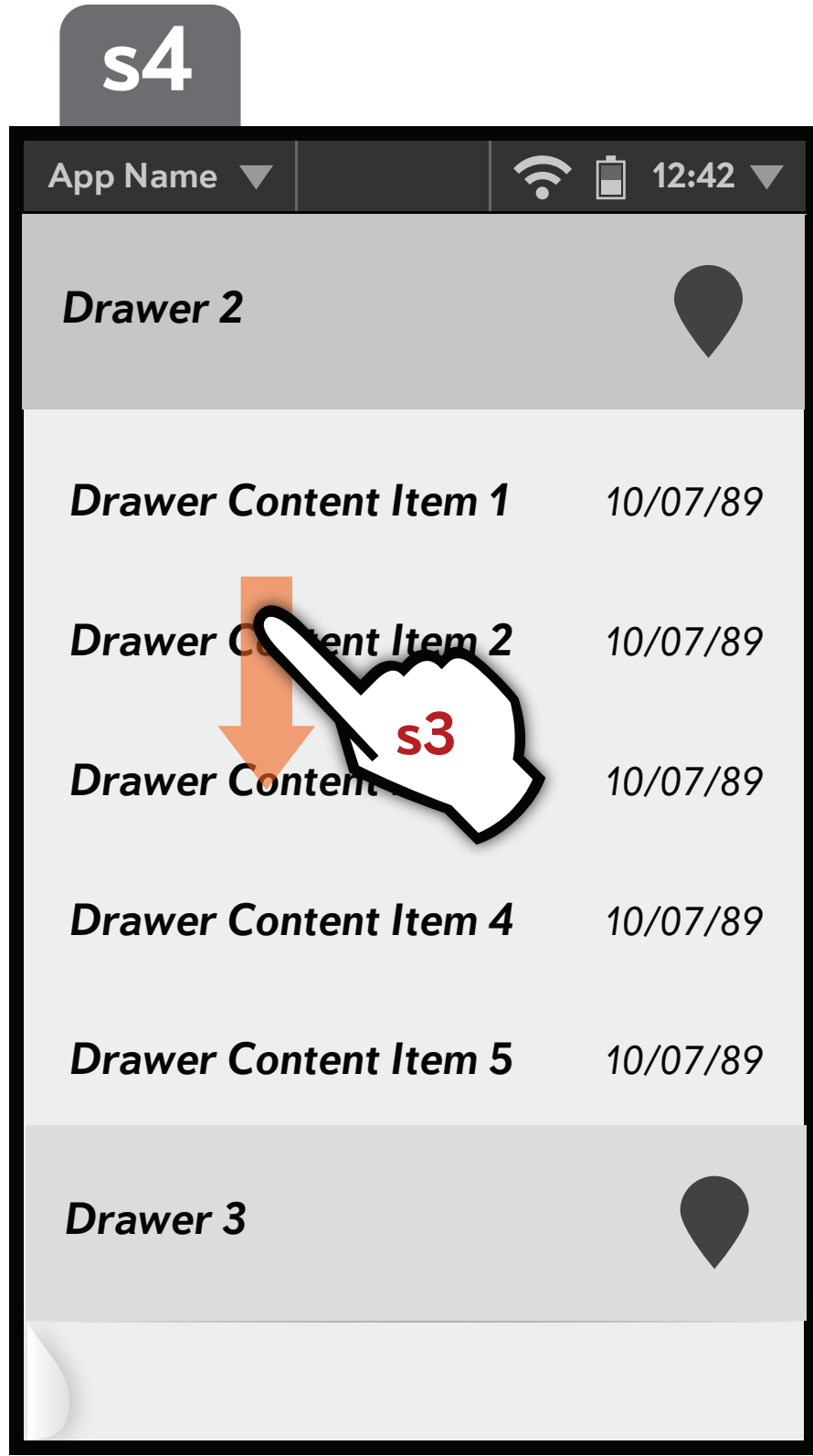
s1. As the user scrolls, the title of currently expanded divider pins to the top of the screen. Title of the next divider pins to the bottom.



s2. When the last item in that section is reached, the next two dividers are pulled onto the screen.



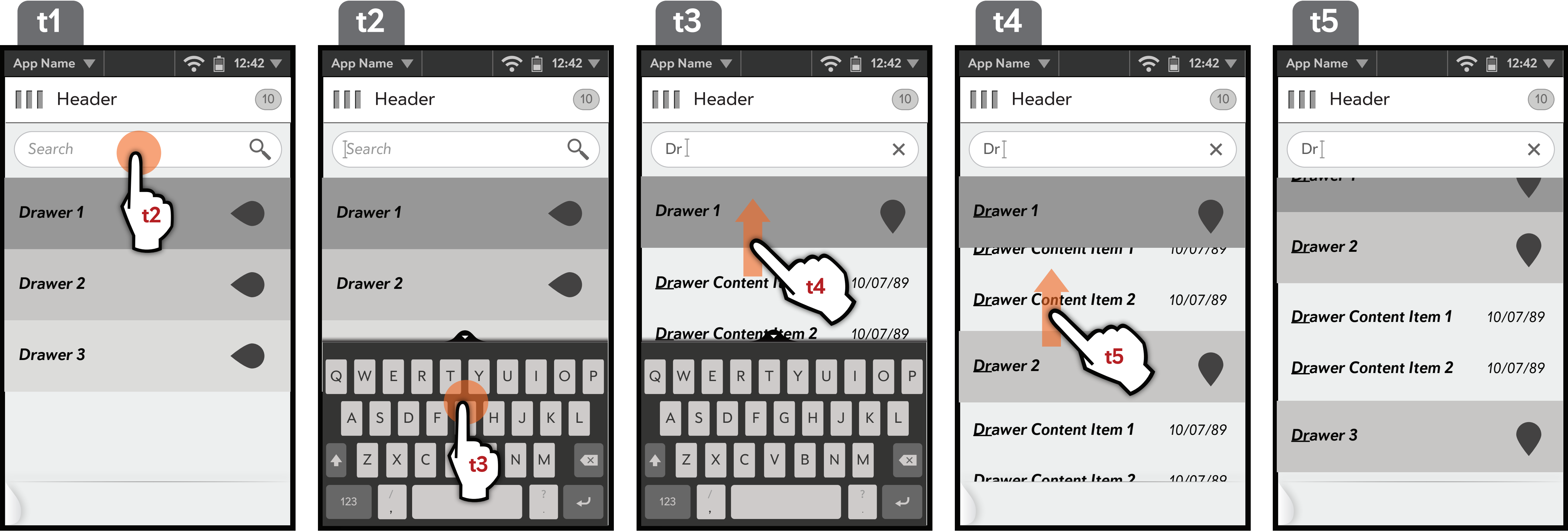
s3. The bottommost divider stays pinned to the top as the middle divider scrolls upwards.



s4. When the middle divider reaches the top divider, it pushes it off the screen. Bottom divider remains pinned to the bottom.

If user scrolls up, behavior is in reverse.

EXAMPLE TASK FLOW: SEARCHING COLLAPSIBLE DIVIDERS



t1. User taps the Search Input field.

t2. Field is focused, and cursor appears. Hint text remains until user begins typing.

For full search behavior please refer to the Search Input documentation.

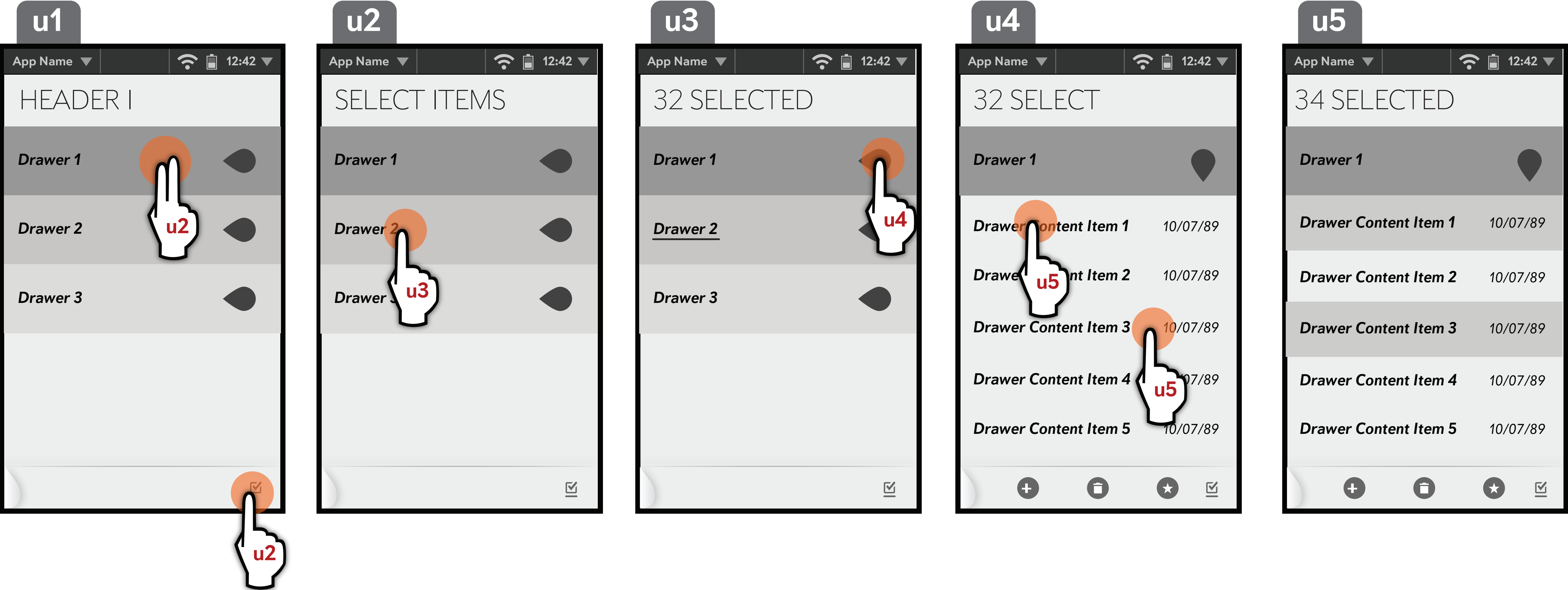
t3. Devs decide whether scrolling will dismiss the keyboard.

t4. Devs decide whether search results are:

- Divider titles--show only collapsed divider results
- Content--automatically expands divders, display relevant content results. Underline matching text in content results only.
- Both--display relevant divider results, automatically expand dividers with matching content results. Underline matching text in titles and content.

t5. User can now make a selection from the filtered results.

EXAMPLE TASK FLOW: MULTI-SELECT WITH COLLAPSIBLE DIVIDERS



u1. User enters the multi select mode by two-finger tap or multi-select button in chrome.

For full Multi-select behavior please refer to the Multi-Select flows on p.26-27.

u2. If the divider has two hit targets (see Structure p.37), tapping the title will select all its content. Divider title should show selected state.

Visual Design TBD.

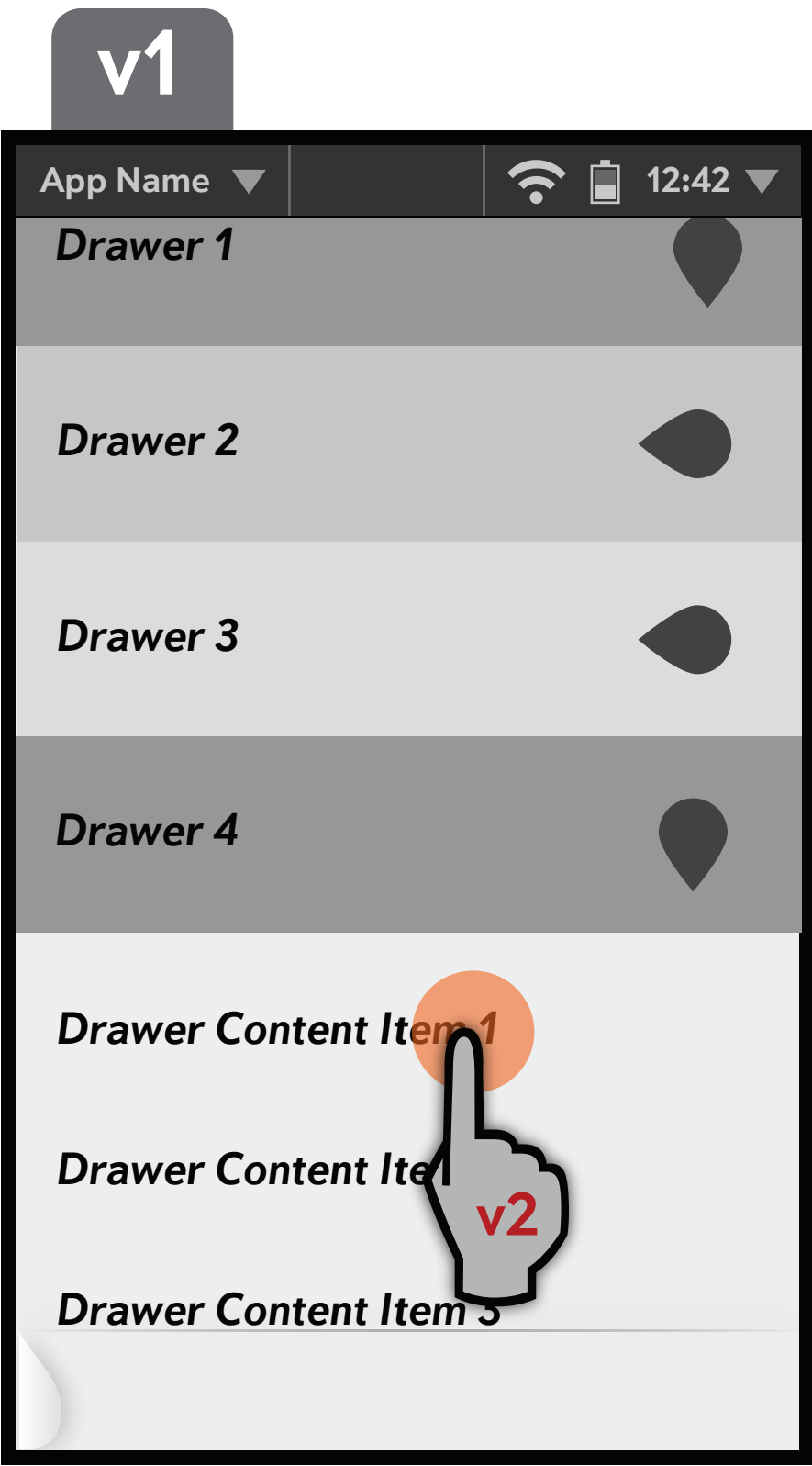
u3. If the user taps on the arrow, expand and display content. (See Structure p.4)

Visual Design TBD.

u4. User can select individual items.

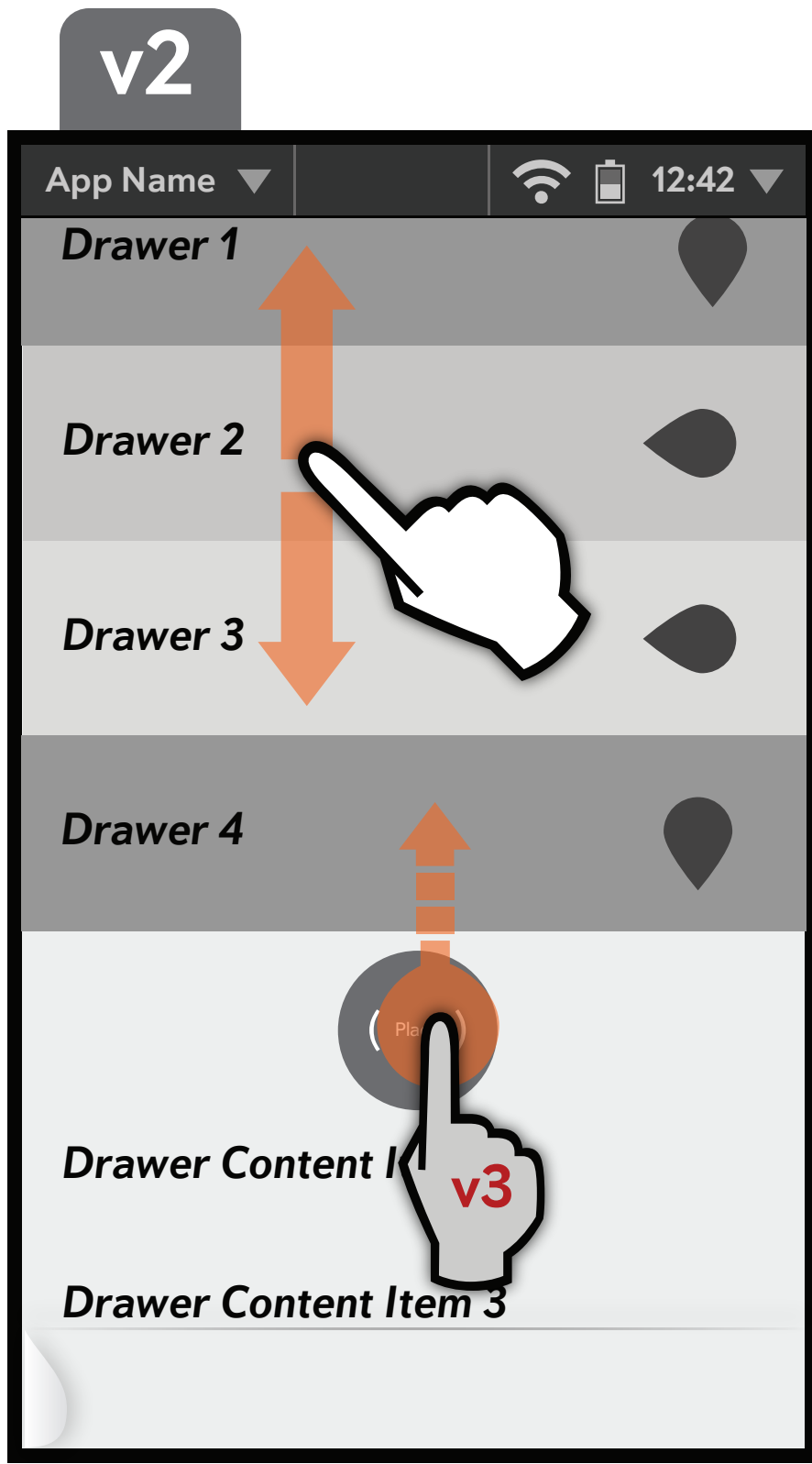
u5. Selected items are highlighted, and users can perform actions to them using the action buttons in the footer

EXAMPLE TASK FLOW: MOVING AN ITEM BETWEEN LISTS



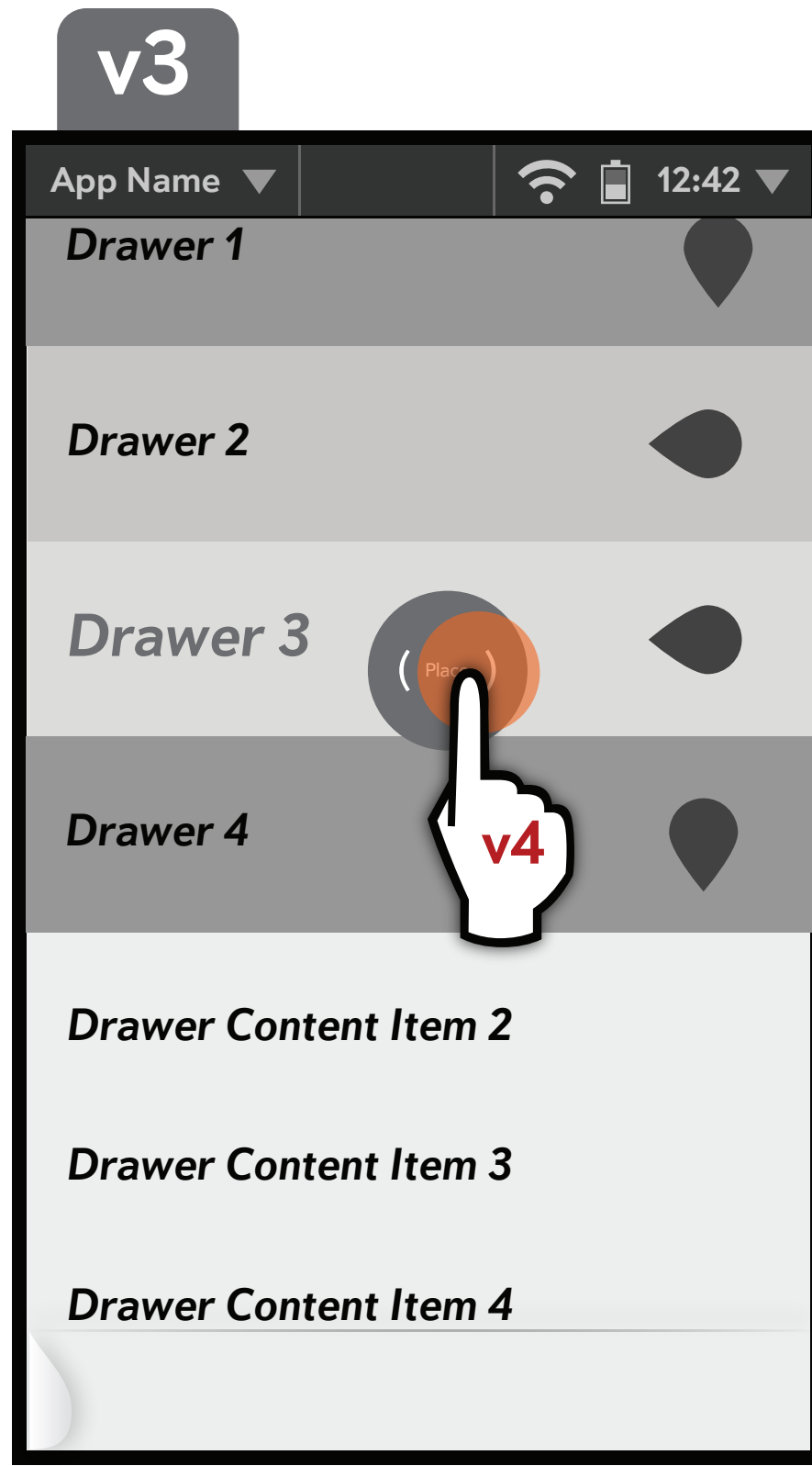
v1. Press and hold to go into drag & drop mode.

Visual Design TBD. For more details on moving and reordering behavior please refer to the Re-ordering and Moving flows on p.22-25)

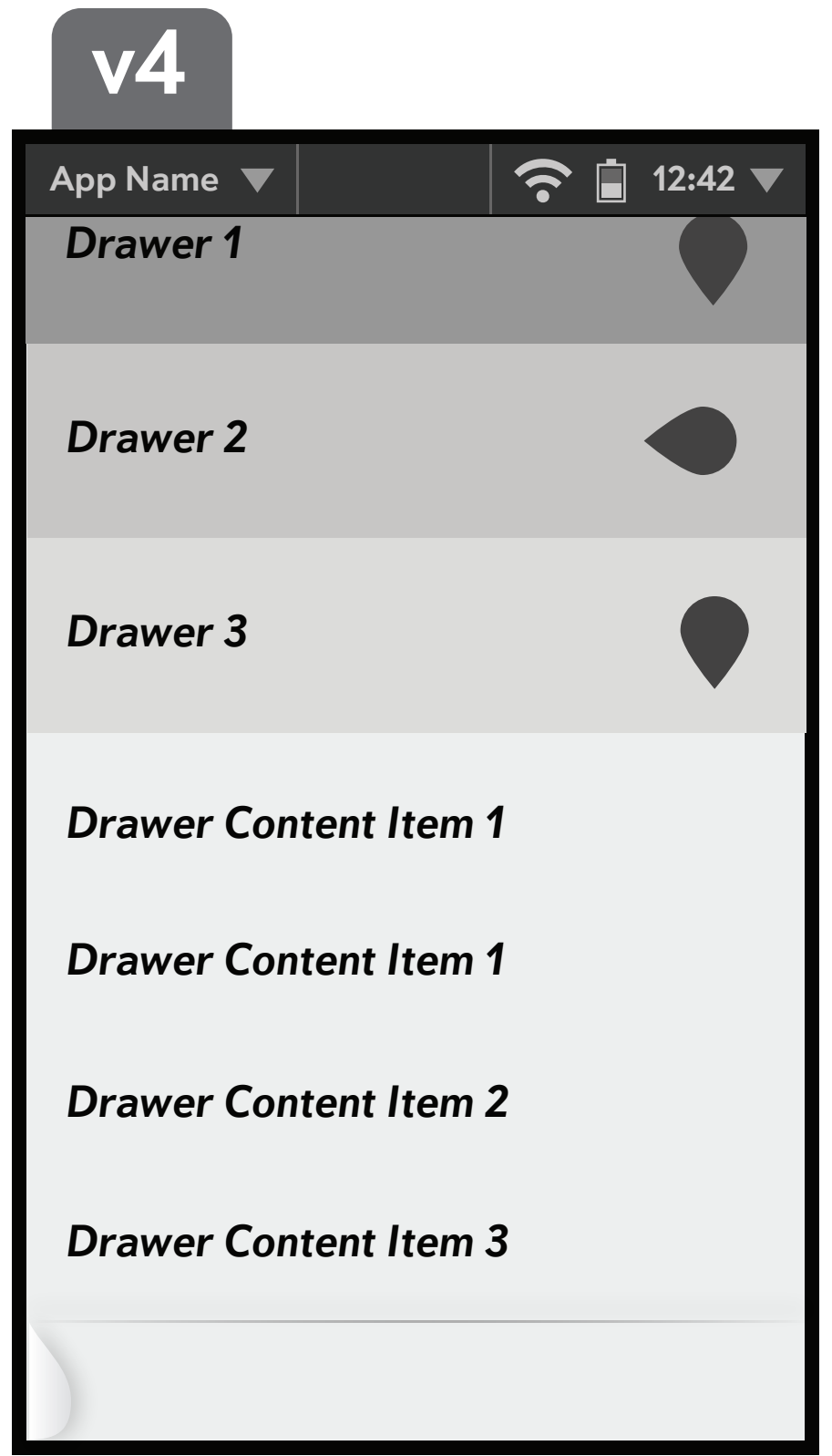


v2. Without lifting the finger, drag the item towards the new location.

User can also release and scroll the list as described in the Lists documentation. Collapsible Dividers can be opened with a tap in this mode (see Structure p.37).

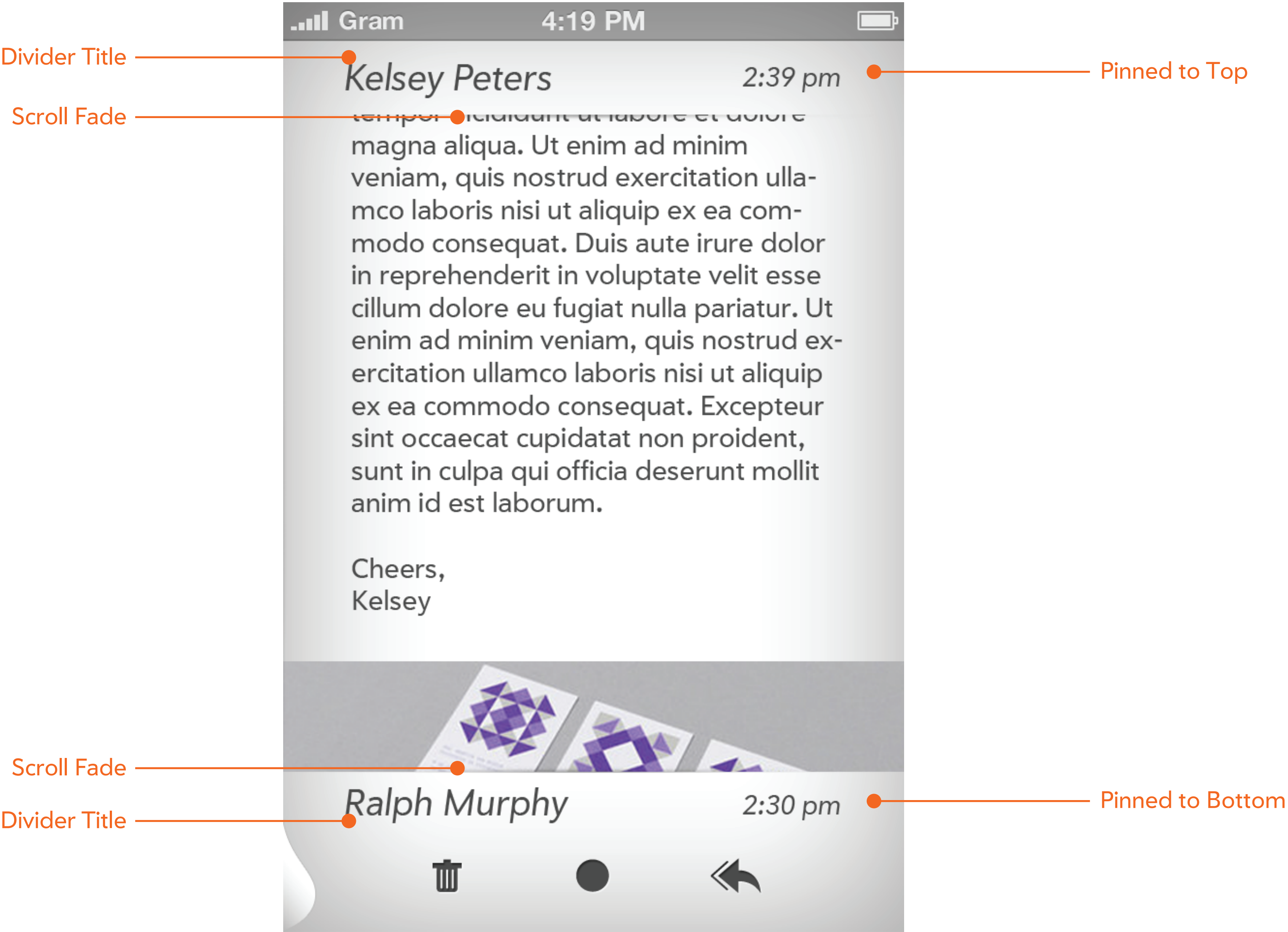


v3. User can drop the item into a drawer by tapping (Place).

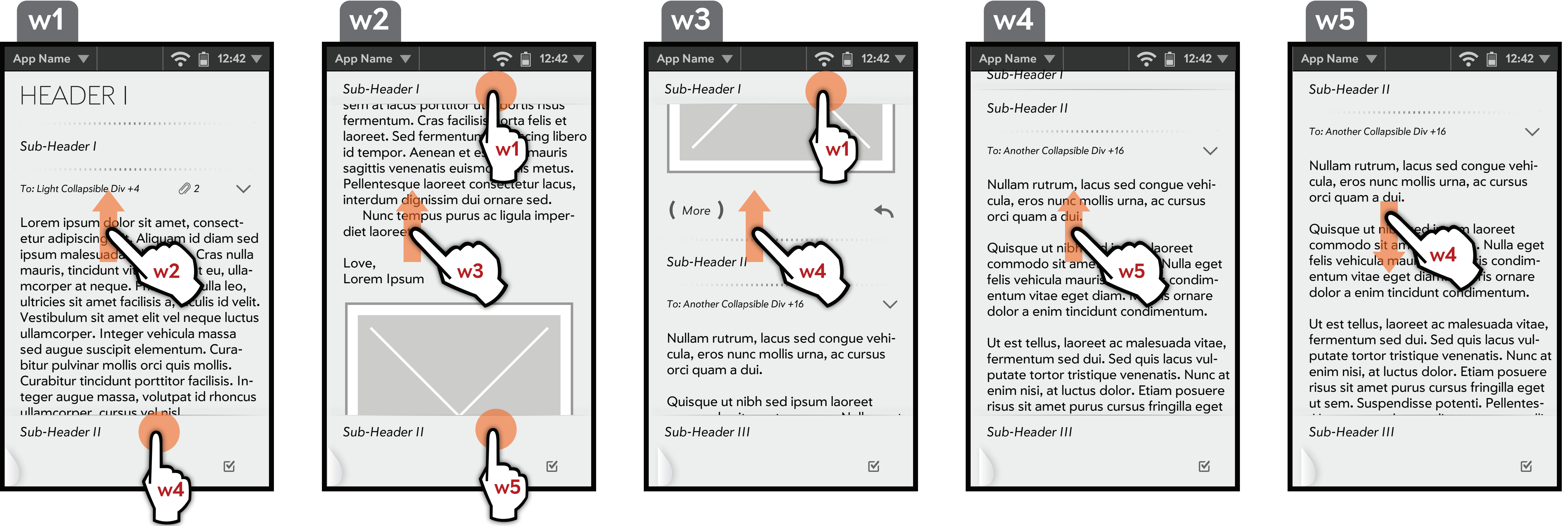


v4. Item is placed as the first item in that drawer.

NAVIGABLE DIVIDERS: VISUAL DESIGN EXAMPLE



NAVIGABLE DIVIDERS



w1. Upon app launch, display 1st list item. Title of the 2nd item is pinned to the bottom.

In this scenario, navigable dividers consist of a selectable sub-header and a scroll fade.

w2. If the user begins to scroll down, the title of the 1st item will pin to the top of the screen.

Tapping the title pinned to the top will go back to the 1st item (**w1**).

Tapping the title pinned to the bottom will go to the next item (**w4**).

w3. When the end of the content of the 1st item is reached, the 2nd item (and its title and content) should begin to scroll upwards, and the 3rd title is pinned to the bottom.

w4. If the user continues to scroll, the title of the 2nd item will push the title of the 1st up off the screen.

w5. Once a title is tapped, it sits at the top of the screen. The user must scroll down before the title is pinned, or up to see the previous header pinned (**w2-4**).

CHANGE HISTORY

9/14/2012

- First Draft

9/19/2012

- Jump List (generic) added
- Moving added
- Multiselect added
- Swipe-able Items added
- Inline Editing added

9/26/2012

- Annotations adjusted for 2-Finger scrolling, Alpha Jump and Jump Lists. Two-finger swipe should work in all lists, but two-finger dragging should not.
- Swiping added to Moving Version B
- Clarifications added for Swipe-able Items.
- Expandable Lists added

10/19/2012

- Reordering and Moving combined into one version
- Multiselect adjusted to be a toggling mode, allowing for a cancel
- Cancel added persistent Swipe-able Item added