User Experience

Mobile Style Guide

12.14.15

Table of Contents

Anatomy of an App

Basic Building Blocks for Releasing a Successful Product

All of our applications use a common set of components to make up all the various pages found inside an app.

The components we have defined can be assigned into one of three broad categories:

NAVIGATION

Informs the user where they currently are within the app, what other areas they have access to, and what they will find when they get there.

CONTENT & MESSAGING

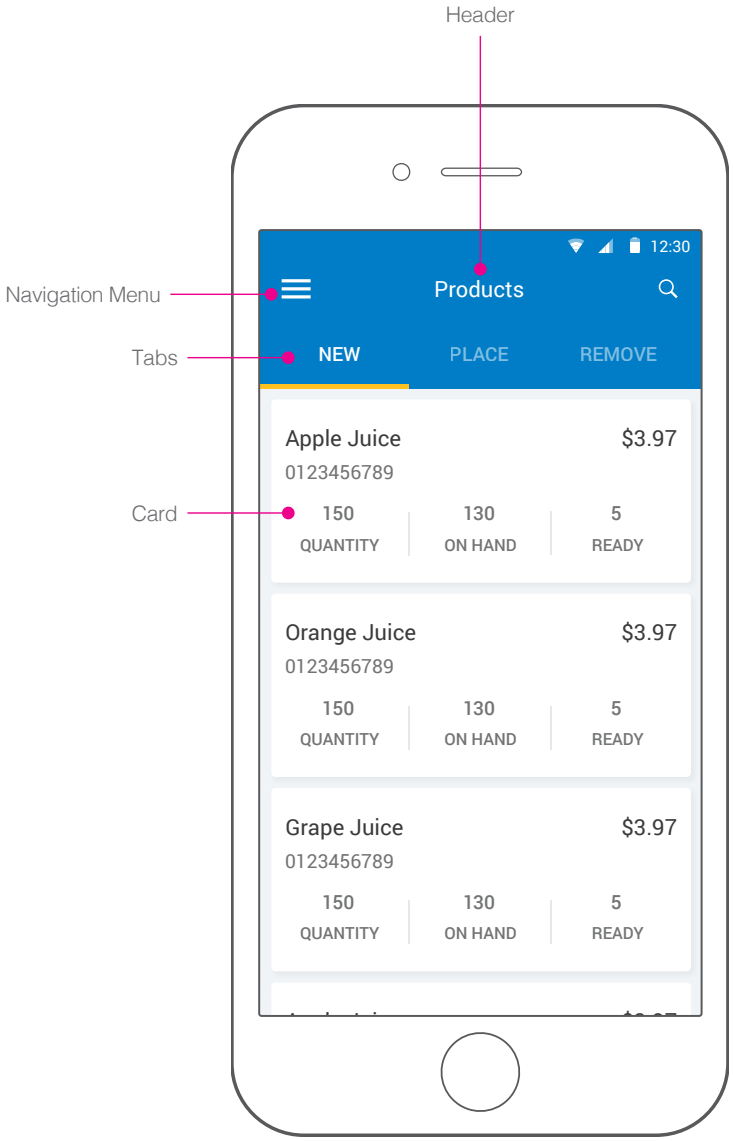
Formatted information that allows the user to accomplish tasks efficiently and accurately.

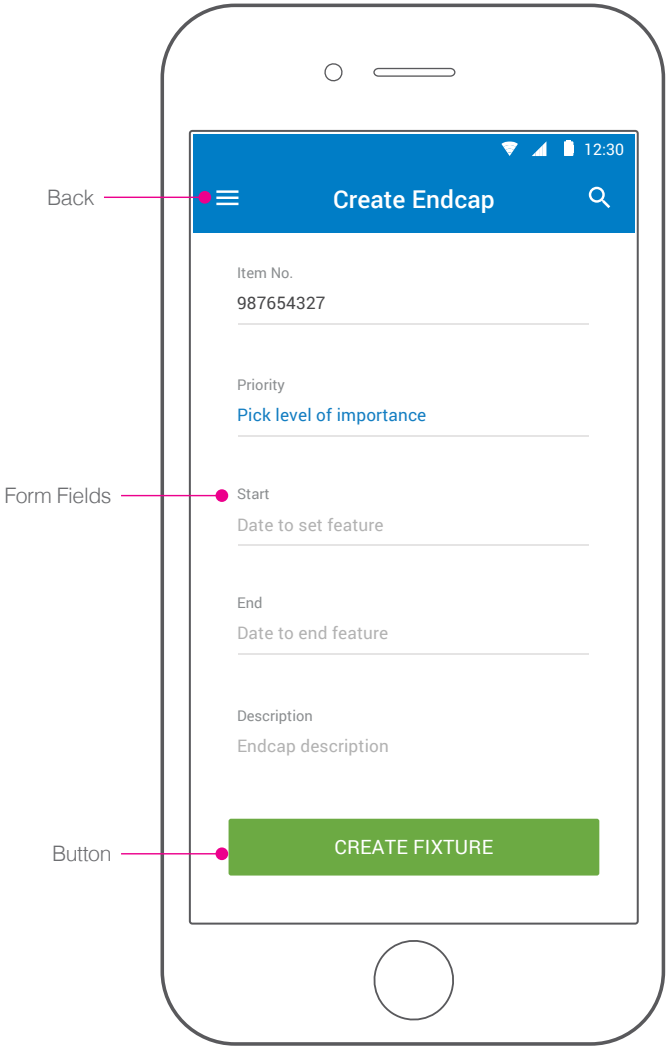
INTERACTION

Allows the user to input information into our systems with form fields and buttons.

This icon shares helpful external references.

This icon identifies key call outs.


Header
Navigation Menu
Tabs
Card
12:30
Products
NEW
PLACE
REMOVE
Apple Juice
$3.97
0123456789
150
QUANTITY
130
ON HAND
5
READY
Orange Juice
$3.97
0123456789
150
QUANTITY
130
ON HAND
5
READY
Grape Juice
$3.97
0123456789
150
QUANTITY
130
ON HAND
5
READY


Back
Form Fields
Button
12:30
Create Endcap
Item No.
987654327
Priority
Pick level of importance
Start
Date to set feature
End
Date to end feature
Description
Endcap description
CREATE FIXTURE

Style

A Comprehensive Visual System

Style guidelines are a set of design standards to ensure consistency over every instance of associate engagement. They unify every application, website, and terminal interface to belong to a cohesive sentence.

By applying these standards, your project will become more familiar and usable to the end-user, thus making the project more successful overall. These standards will also decrease your development time by providing you solutions to common design and interaction problems.

We want this document to be your trusted go-to resource.

1.0 Style

Brand Identity

The DNA of our Brand

All of our applications incorporate elements from our brand to express a singular representation across many platforms and devices. There are three configurations of the Walmart wordmark and Spark that can be used within an app:

WORDMARK WITH SPARK

Useful in load pages and headers (Fig. 1).

ONLY SPARK ICON

A decorative element to represent the Walmart brand (Fig. 2).

STACKED LOGO

An accent piece, useful in areas such as the bottom of a page (Fig. 3).

Use only authorized artwork from: www.walmartbrandcenter.com

FIG. 1 Wordmark with spark

FIG. 2 Spark icon

FIG. 3 Stacked logo

Brand Identity

Best Practice

DO

- ✓ Keep the minimum size of brand assets to 100px wide.
- ✓ Maintain clear space around other elements. There should be at least 10px of padding between other elements.

DO NOT

- ✖ Stretch the logo or distort the proportions of the logo and wordmark.
- ✖ Add or modify brand assets.
- ✖ Add effects, such as coloring, drop shadow or glow.
- ✖ Use brand assets that include the tagline.

Use only authorized artwork from: www.walmartbrandcenter.com

FIG. 1 Keep brand identity assets proportional and in the correct color

Brand Identity

Variation

Use a horizontal treatment with departments and services to help communicate the relationship between the department and the brand (Fig. 1).

FIG. 1 Logo with department

Colors

A Powerful Tool that Influences our Emotions and How We Perceive Information

A color palette is critical in creating a consistent brand image. In establishing consistency you can also highlight connections in related content to aid users in making faster, better informed decisions. Reserve certain colors for branding to provide a sense of place, and other colors for task-related elements to avoid reducing the colors' effectiveness. Your users will quickly learn to look for these visual cues and enjoy a better experience.

Contrast is the art of using color to implement visual hierarchies that draw in the eye. It is important that an item contrasts enough with its background to provide sufficient visibility, in hue and saturation. If there is insufficient contrast text can be impossible to read, and items in smaller areas are harder to identify, reducing effectiveness.

Over time basic accessibility standards have been established by W3 and released as Web Content Accessibility Guidelines (WCAG). WCAG 2.0 requires a contrast ratio of

4.5:1 for normal text (14 point bold or larger) and 3:1 for large text (18 point or larger) for at least an AA rating.

See the Walmart Brand Style Guide: http://www.walmartbrandcenter.com

Walmart Dark Blue
Pantone 287
C100/M68/Y0/K20
R0/G76/B145
HEX #004c91

Walmart Medium Blue
Pantone 285
C100/M40/Y0/K0
R0/G125/B198
HEX #007dc6

Walmart Light Blue
Pantone 284
C50/M13/Y0/K0
R120/G185/B231
HEX #78b9e7

Walmart Orange
Pantone 166
C0/M68/Y100/K00
R244/G115/B33
HEX #f47321

Walmart Yellow
Pantone 1235
C0/M25/Y95/K0
R255/G194/B32
HEX #ffc220

Fresh Product Dark Green
Pantone 364
C70/M10/Y100/K32
R54/G124/B43
HEX #367c2b

Fresh Product Light Green
Pantone 368
C58/M0/Y100/K00
R118/G192/B67
HEX #76c043

Walmart Red
Pantone 485
C0/M100/Y100/K0
R230/G43/B30
HEX e62b1e

FIG. 1 Core Walmart brand colors

Colors

Best Practice

DO

- ✓ Stick with simple palettes such as a primary and three accent colors.
- ✓ Take advantage of white space.
- ✓ Use an opacity of 26% for dark text on light backgrounds for hints and disabled text.
- ✓ Use an opacity of 30% for white text on dark backgrounds for hints and disabled text.
- ✓ Use primary colors for large areas of color.
- ✓ Use accent colors for buttons and other actionable items.

DO NOT

- ✖ Use dark backgrounds with light text.
- ✖ Use too many bright colors at once; this creates a negative effect.
- ✖ Use low-contrast text and images on low-value backgrounds.
- ✖ Use color combinations that are difficult for the visually impaired.
- ✖ Use accent colors for body copy.
- ✖ Interchange primary and accent colors.

There is good contrast between this grey (#394449) and the white background.

There is too little contrast between this grey (#CCCCCC) and the white background.

FIG. 1 Use proper contrast with text

FIG. 2 Do not use shades of a single color for contrast

Colors

Variations

Over time basic accessibility standards have been established by W3 and released as Web Content Accessibility Guidelines (WCAG). WCAG 2.0 requires a contrast ratio of 4.5:1 for normal text (14 point bold or larger) and 3:1 for large text (18 point or larger) for at least an AA rating.

- Text colors used in examples are white (#FFFFFF) and black (#000000) according to an AA rating.
- Acceptable text use may include grey for body text of #4f595d or darker. Lighter greys are reserved for disabled states.
- Tints - Adding white to a pure hue.
- Shades - Adding black to a pure hue.

Official Site:http://www.w3.org/WAI/intro/wcag

Test contrast ratios: http://leaverou.github.io/contrast-ratio

	Primary Palette Best for headers, footer			Secondary Palette Best for icons, accents				Grey Palette Best for text
Darker Shades	#022f56	#064c75	#486f8a	#90451c	#987320	#21491c	#48722d	#881b16	#282828
	#043e72	#0d659c	#6294b7	#c15c26	#cb9a2d	#2d6226	#60993b	#b62420	#3b3b3b
	#064680	#0e71b0	#6ea7cd	#d9672b	#e4ad32	#326e2a	#6caa43	#cd2823	#595959
Core Hues	Walmart Dark Blue #004c91	Walmart Medium Blue #007dc6	Walmart Light Blue #78b9e7	Walmart Orange #f47321	Walmart Yellow #ffc220	Fresh Product Green #367c2b	Fresh Product Light Green #76c043	Walmart Red #e62b1e	Walmart Dark Grey #737373
	#1f5f9a	#248bc9	#88c1e7	#f38142	#fec748	#4c8844	#86c45c	#e6433b	#828282
	#3671a5	#3898cf	#95c7ea	#f48f54	#fecd59	#609558	#92cb6d	#e9564f	#a1a1a1
	#6895bb	#69b2db	#afd5f0	#f48f54	#fed97f	#87af82	#aed891	#ee817a	#c1c1c1
Lighter Tints	#9ab7d2	#9bcbe6	#cae3f4	#fac7a8	#fee6a9	#b0caab	#c8e6b4	#f3aaa6	#e0e0e0

FIG. 1 Color palette

Iconography

Shortcuts with Symbols

Icons are used to transmit familiar pieces of information in a small amount of space. They are a shortcut that benefits the user by providing meaningful information at a glance. For example, instead of having to write, “Sales are up by 0.37 percent,” a developer can program the app to display “↑ 0.37%.”

For common pieces of functionality, use the standard Walmart font library. Icons should be responsive to what the user is doing and have different states for common interactions.

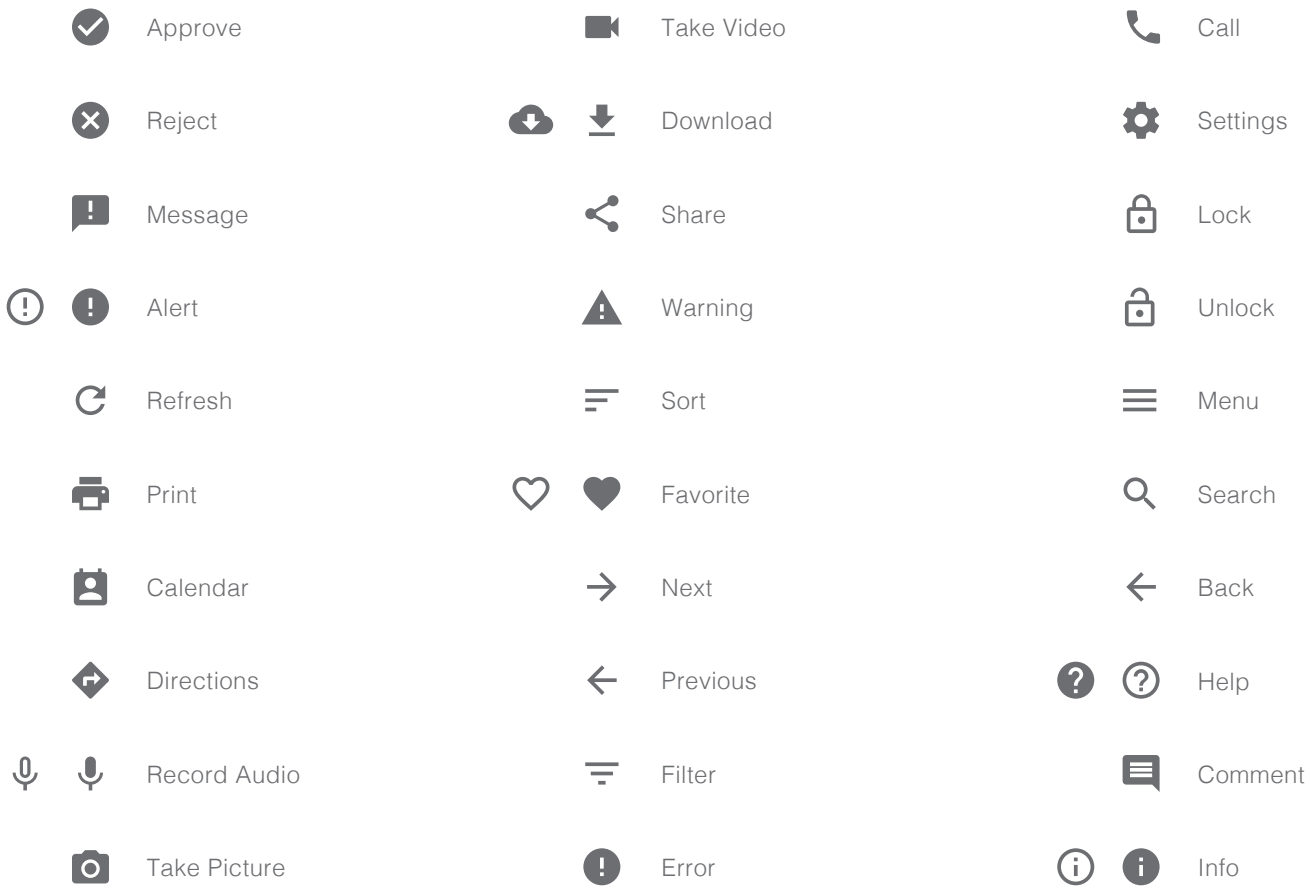


Fig. 1 Icon library

Iconography

Best Practices

DO

✓ Maintain a consistent size and color for icons within an app.

DO NOT

✖ Rely on an icon to be the only way to trigger an action. An actionable icon should be embedded inside a button.

✖ Add gradients, embossing or other distracting design elements to an icon.

✖ Use an image asset, such as png or jpg, for an icon. Icons should always be fonts or vector graphics.

Fig. 1 Keep icons simple and monotne

Typography

Making Content Readable

Through size, weight and color, fonts can direct the user's attention to the most critical pieces of information within an app. Walmart's digital font is Roboto. It is used across all mobile apps, websites and any other digital interface. The implementation of this font, along with careful attention to this style guide, will create a consistent experience across multiple platforms.

Use different weights and styles sparingly. It is important that an interface not have too many different styles or sizes on a single page. Use the different weights to establish a primary focus, secondary supplemental information and calls to action.

In order to create pages that are easy-to-read, line height has been set at 1.5. Line height should only apply to elements that wrap, including body, subhead, and other smaller type styles. Different font styles should be separated by a line break.

Font size can be determined by pixels or ems on a page. Pixels are fixed, and ems is relative. We will be staying with pixels for now.

Font color can be used to emphasize status or hierarchy. However, the more that color is used, the less effective it becomes. Use text color in an app very sparingly.

Download Roboto font from https://www.google.com/design/spec/resources/roboto-noto-fonts.html

Application Title	**Medium 41px/ 2.93em**
Subhead / H1	Regular 32px/ 2.29em
Form Field Entry	Regular 31px/ 2.21em
Drop Down Menu / H2	Light 29px/ 2.07em
Drop Down Menu 2	**Medium 29px/ 2.07em**
Primary Button	**Bold 27px/ 1.93em**
Secondary Button	Regular 27px/ 1.93em
Form Field Title	Regular 24px/ 1.71em
H3	Regular 16px/ 1.14em
Body	Regular 14px/ 1em
Minimum size	Regular 10px/ 0.741em

Fig. 1 Font guidelines

Typography

Best Practice

DO

- ✓ Keep line length between 60 and 80 characters.
- ✓ Use bold and heavy weight fonts on only one element per page.

DO NOT

- ✖ Use more than three weights and/or four sizes of type in a single design.
- ✖ Have more than two text colors on a page.

✔

Pellentesque habitant morbi tristique senectus et netus etmalesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eulibero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

Pellentesque habitant morbi tristique senectus et netus etmalesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante.

Fig. 1 Keep line length between 60-80 characters

✖

Pellentesque habitant morbi tristique senectus et netus etmalesuada fames ac turpis egestas. **Vestibulum** tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eulibero sit amet quam egestas semper. Aenean ultricies **mi vitae est**. Mauris placerat eleifend leo.

Fig. 1 Maintain uniform sizes and weights

Imagery

Every Picture Tells a Story

Photography and illustrations are a powerful medium for expressing our values and making human connections. We want our images to look natural and compelling. Care should be taken to avoid stock photography syndrome. Images of people should reflect our personality traits and be simple, real, positive and caring. Our internal material should take care to show inspirational professionalism and be informative. Reserve the use of photography for when telling a story with a human focus, telling an emotional story or referring to specific products. Use illustrations when trying to convey ideas in a way more abstract than photographs can clearly express. When using illustrations try to use clean, flat graphics and avoid elaborate realistic creations or low-quality clip art.

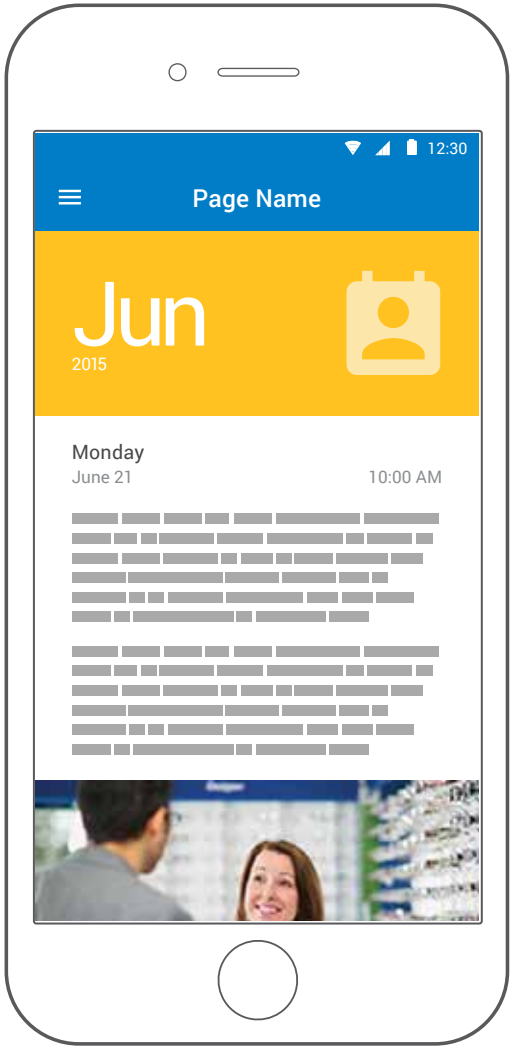


Fig. 1 Mobile example

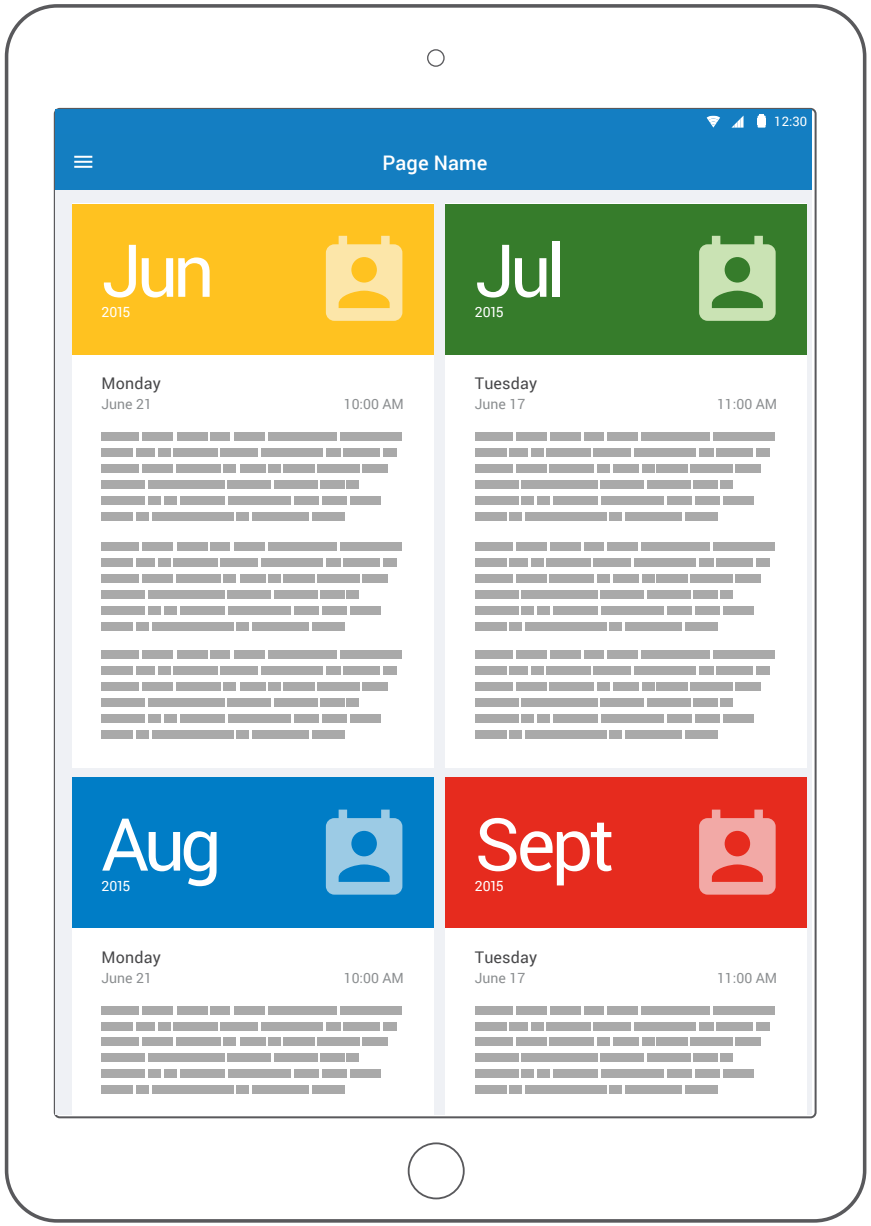


Fig. 2 Tablet example

Imagery

Best Practice

DO

- ✓ Use clean and well-lit product images.
- ✓ Reflect a positive emotional message with human photographs.
- ✓ Always reflect the brand, if possible.
- ✓ Display the diversity of the workforce when possible.
- ✓ Always think of quality from best display to lowest.
- ✓ Follow the visual flow of an image when adding text (where are things pointing?).

DO NOT

- ✖ Use cheap clipart or generic stock photos.
- ✖ Waste page real estate; consider the impact size of your message.
- ✖ Use highly feathered edges or bad cutouts.
- ✖ Use boxes around photographs or drop shadows.
- ✖ Display pixelated or distorted graphics.
- ✖ Overload a picture with text. If it includes more characters than a tweet you are doing it wrong.

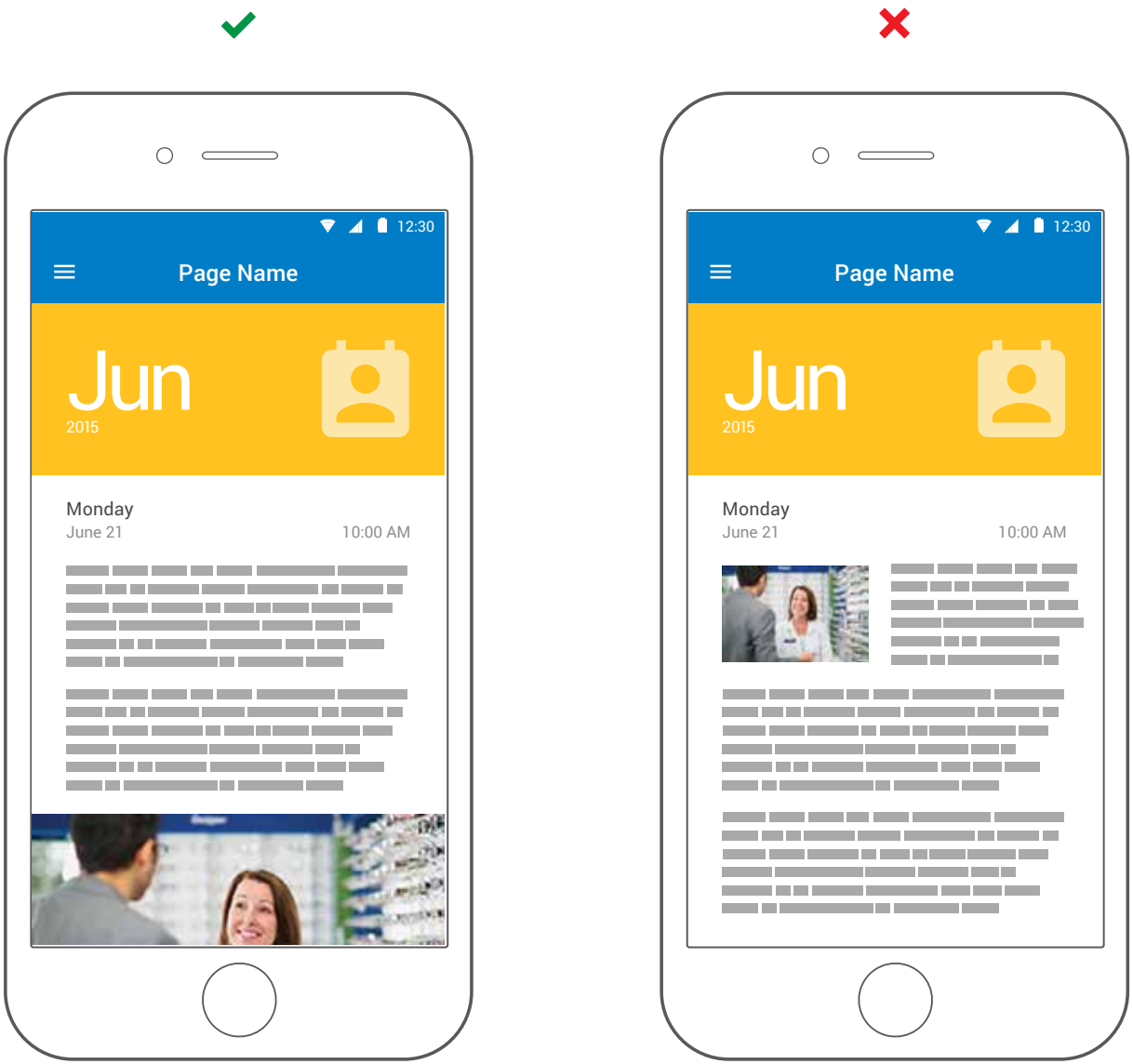


Fig. 1 Consider the impact of your visual. Larger visuals help break up text

Imagery

Variations

When displaying graphics we want to present a clean visual. Avoid elements that take away from the message. Avoid gradient and color overlays, except in limited cases for functionality, and then, use only if the image's message is not damaged by doing so. If you use gradients, make them long to avoid barring. Use at least 3x the average navigational bar height with the colored edge closer to the darkest region, by about 3/10ths, to encourage a natural look and avoid a sharp edge. Use a darker gradient for darker images, lighter for lighter images ranging from 60% – 20%. When using text overlays, do not place text over faces or busy images with a clear focus. Ensure that there is sufficient contrast. This does not necessarily mean to use black or white text, but use complementary colors, positioning and sizing in comparison with the focus. Putting text in a soft opacity box or circle is acceptable, but again make sure you are not obscuring important messaging elements of the image. Do not overlay text on images smaller than 70px.

FIG. 1 Consider the impact of your visual. Larger visuals help break up text

Layouts

Page Content is only as Good as its Layout.

It is important to be mindful of the size, space and arrangement of content on a page. Visual hierarchy of content can help a user quickly execute their goals, making for a successful app. Following a grid system helps reinforce this by enforcing alignment, making it easier to determine content placement and provides a professional appearance regardless of device.

2.0 Layouts

Device and Screen Size

Breaking Down Platform Specs

In a world of fragmented devices, each with their own screen sizes and resolutions, it is important to recognize different devices and how they display graphics.

TERMINOLOGY TO RECOGNIZE:

- Page size is the diagonal measurement of the actual page in inches.
- Resolution is how many pixels a display can show at a time, in order of width x height.
- Density is the number of pixels compressed into a space to create a sharper image.
- Pixels represent a basic graphical element and is an absolute form of measurement.
- Ems represents a basic unit of measurement based off typography and is a relative form of measurement. Ems are also applied to padding and margins.

Fig. 1 Mobile example

Fig. 2 Tablet example

Device and Screen Size

Best Practice

DO

- ✓ Know your customer and their needs. Phones are used differently from tablets.
- ✓ Optimize layouts for multiple page sizes.
- ✓ Keep interactions and graphics lightweight.

DO NOT

- ✖ Make small images larger (stretching does not increase resolution, only pixelates) so always start with a large copy and size down.
- ✖ Copy platform-specific controls.

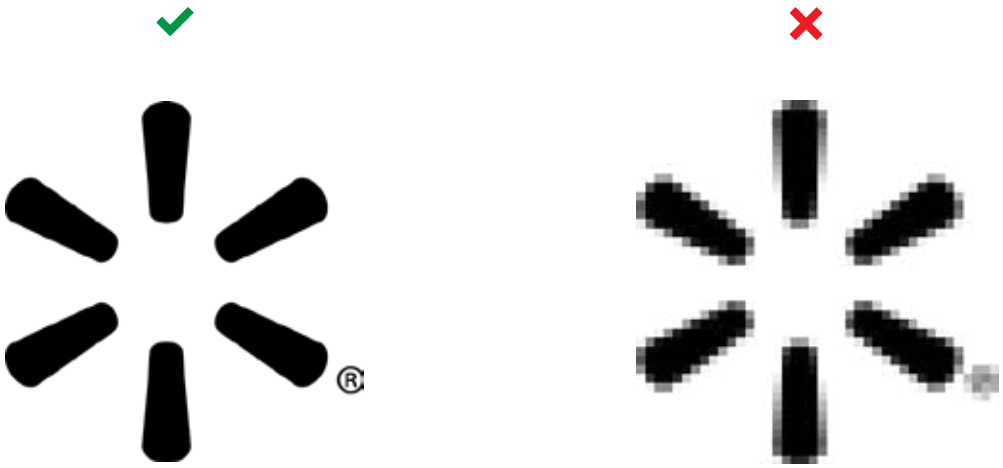


Fig. 1 Optimize page assets based on screen size

Device and Screen Size

iOS Devices

iOS DEVICES

Instead of pixel-based measurements, iOS draws its coordinates in distances called points. The purpose is to create a consistent scale, independent of device, to specify the size and position of views and rendered content. These points are then multiplied by a scale factor. If the screen has a lower pixel resolution this rendering process is resized (downscaled).

Example: An icon for a standard page would be named “icon.png” at 72x72 px. For a retina iPad it would be 144 x 144 px, named “icon@2x.png.”

- Always remember one point does not have to match to one pixel.
- A point is a typographic term that measures a physical length.
- 1 px is equal to .75 pt.
- Retina pages contain a higher pixel density.
- “Rendered pixels” refers to the full number of pixels in a given area.

- iPhone 6 Resolutions: http://iphone6plusresolution.com
- Guide to iPhone Resolutions: http://www.paintcodeapp.com/news/ultimate-guide-to-iphone-resolutions
- iOS Developers: https://developer.apple.com/library/ios/documentation/2DDrawing/Conceptual/DrawingPrintingiOS/GraphicsDrawingOverviewGraphicsDrawingOverview.html

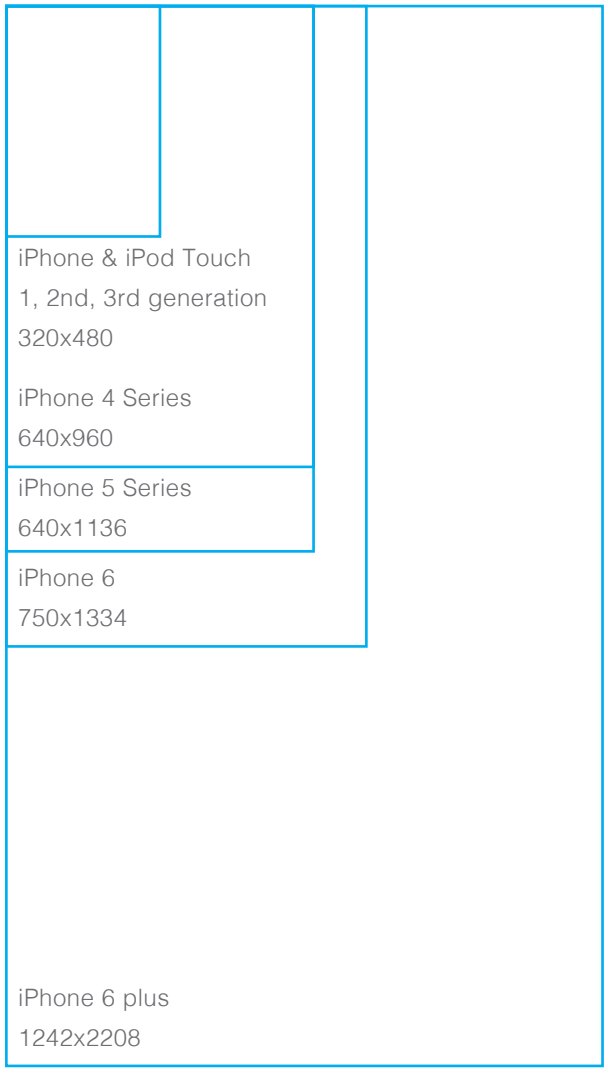


Fig. 1 iPhone screen dimensions

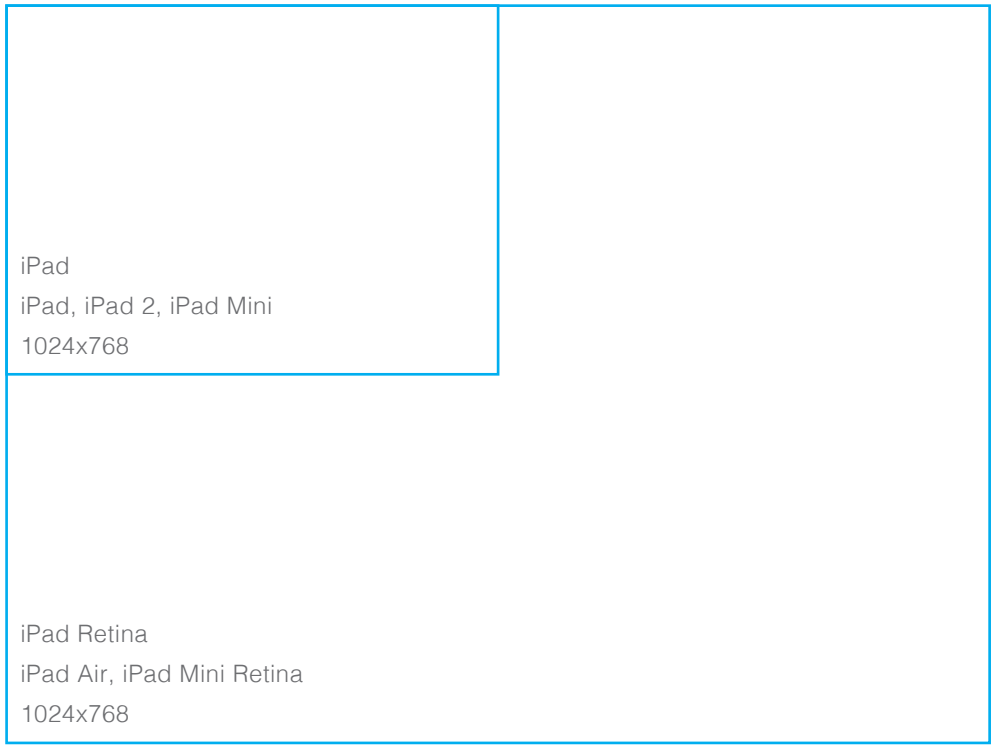


Fig. 2 iPad screen dimensions

Device and Screen Size

Android Devices

Android has adapted a different method to address image quality than iOS, partly in response to having less control over their hardware. Android refers to pixel density in terms of DPI (dots per inch). Android recognizes six general DPI densities: Low, Medium, High, Extra-High, Extra-Extra-High and Extra-Extra-Extra-High.

NOTE: In MDPI pages, physical pixels are identical to DPIs, but always remember one DPI does not equal one pixel. (“DPI” was originally a print term for the number of dots on a printed inch. The more dots, the higher detail.)

- See for dpi conversions at: http://androidpixels.net
- Android Developers: https://developer.android.com/about/dashboards/index.html

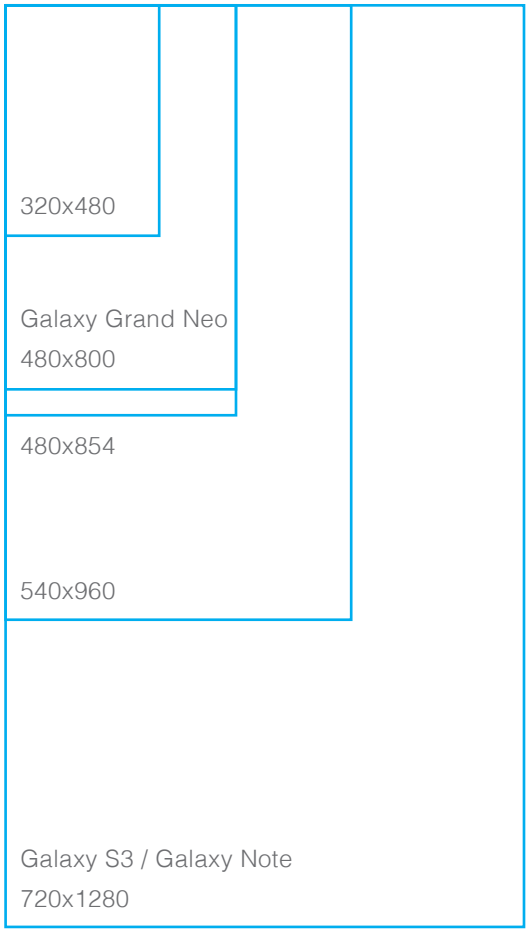


Fig. 1 Android screen dimensions

Fig. 2 Android screen dimensions

Responsive Page Elements

Ensure The Fluidity of your Content to Fit Across Multiple Devices

Responsive design adapts content based on the size and capabilities of a device's page according to a layout grid. It takes advantage of breakpoints (the target points at which your content responds to changes in the page width and height) rather than following a device's set page size. When laying out content it is important to start small and scale up to ensure your content works on mobile first.

LAYOUT ESSENTIALS CHECKLIST:

- Focus – What element commands the most attention? Are those the elements you want to do so?
- Flow – Western culture reads in a left to right, top to bottom style. Does the content support this?
- Grouping – Do related elements appear to belong together?
- Resize Ability – Have you used relative sizes for elements so they can stretch and shrink to fit different pages better?
- Simplicity – Is there visual clutter that may distract users from their goals?
- Consistency – Are elements treated the same across apps for easier recognition?
- Bite Size – Is text simple and to the point?

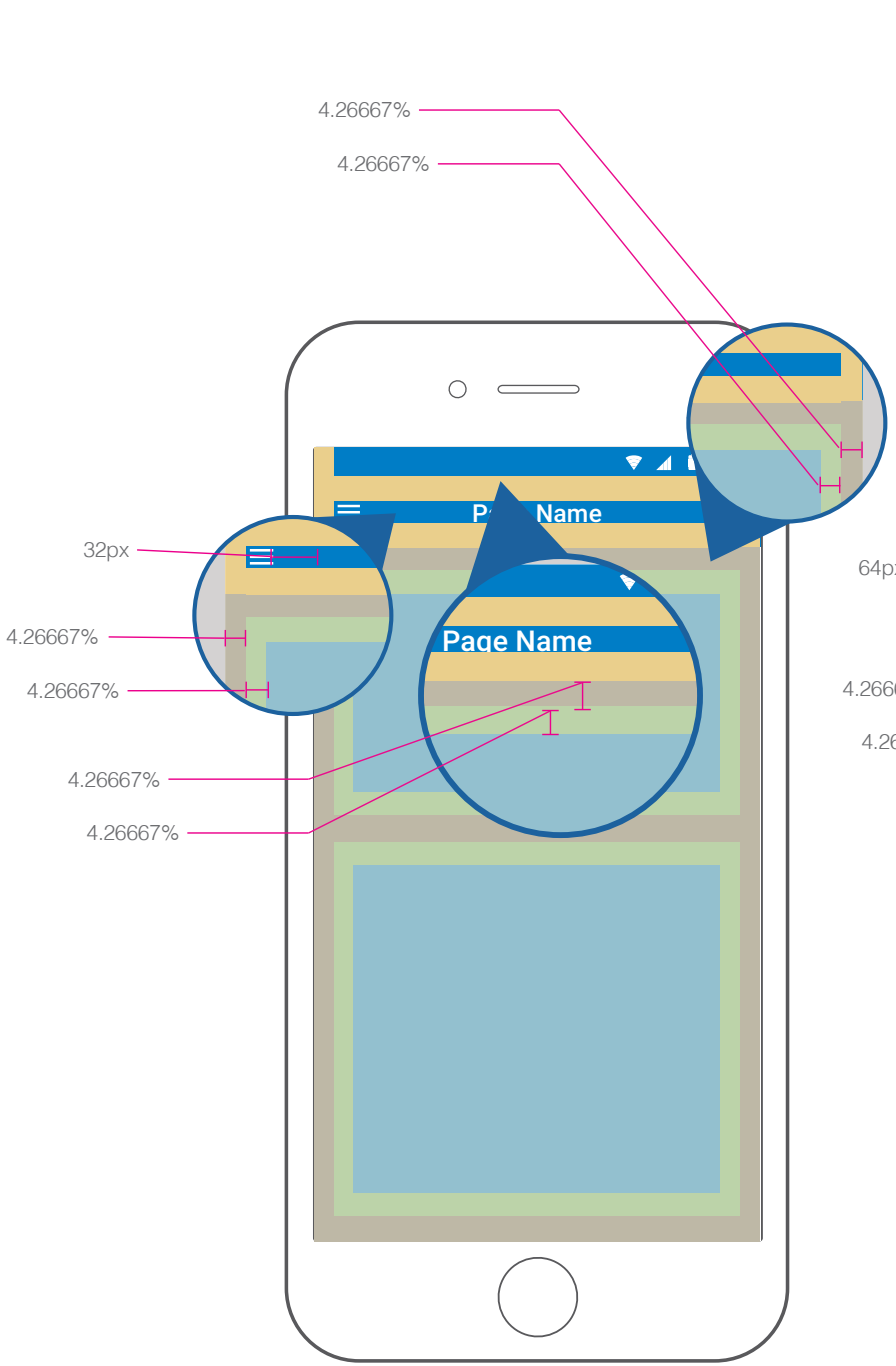


Fig. 1 Mobile example

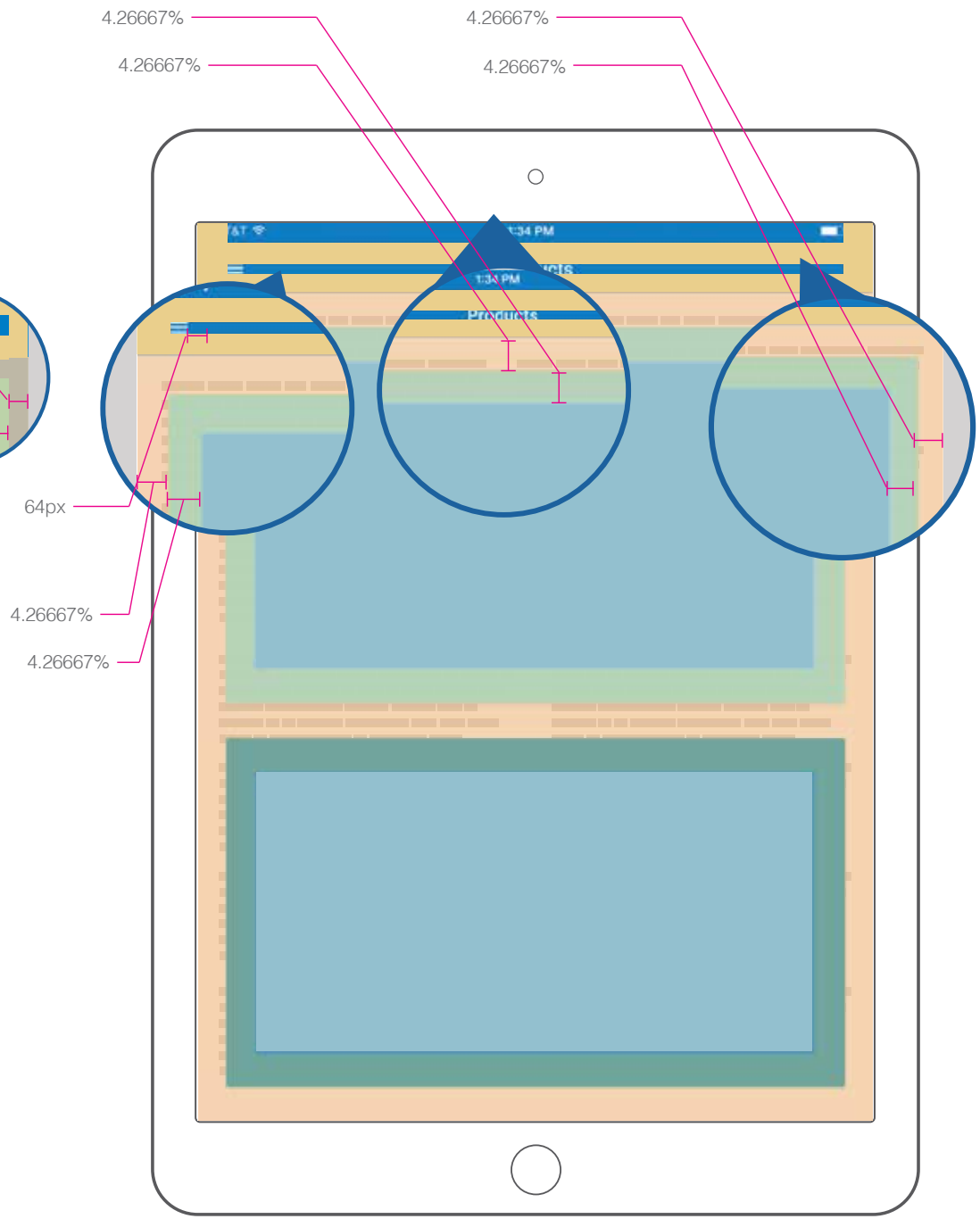


Fig. 2 Tablet example

Responsive Page Elements

Best Practice

DO

- ✓ Create a flow map to understand the layout of your product and how users will interact with it.
- ✓ Keep calls to action front and center.
- ✓ Make it easy to get to the Home Page.
- ✓ Make sure images and videos have some elasticity.
- ✓ Remember, the rule of thumb is to have fonts at 14 pt on mobile.

DO NOT

- ✖ Place critical content in the bottom left corner.
- ✖ Force users to scroll horizontally.
- ✖ Give everything to everyone.
- ✖ Require interaction to see important elements.

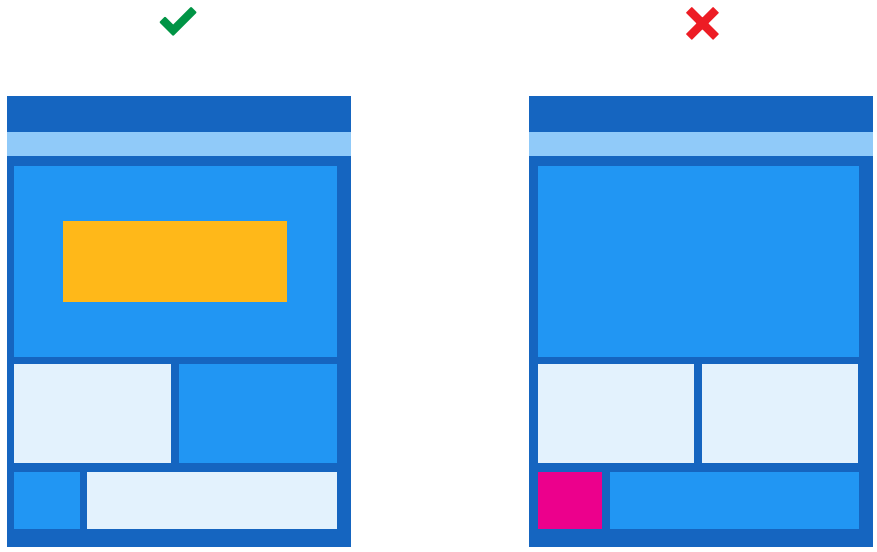


Fig. 1 Avoid cluttered content layout

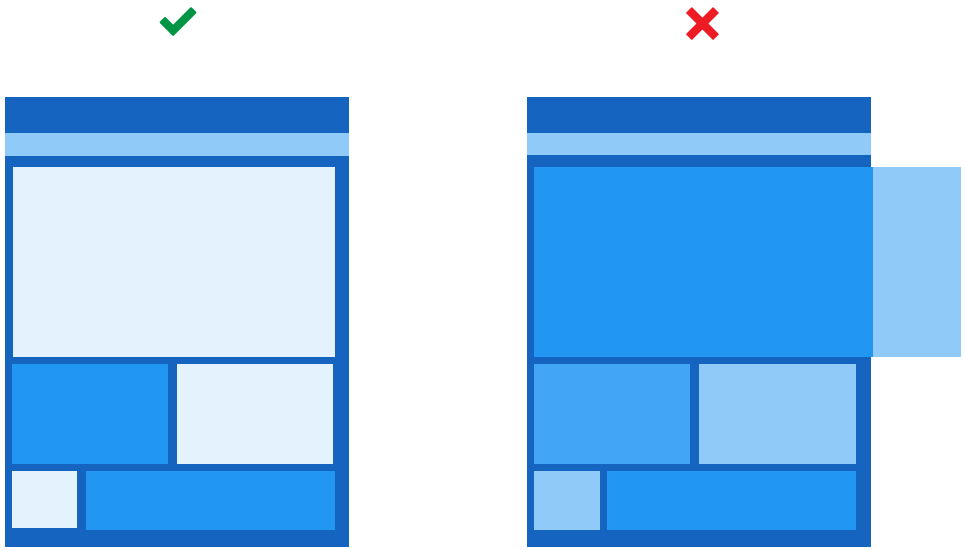


Fig. 2 Do not force users to scroll horizontally

Responsive Page Elements

Variations

Responsive templates work as blueprints for buckets of content. Shown in **(Fig. 1)** are several template layout styles to use as resources. They can contain the universal elements such as the header, footer, navigation, and any necessary content in a consistent and aligned fashion. Each template can be custom-modified to accommodate specific needs.

- Code Snippets for Mobile Web: https://developers.google.com/web/fundamentals/resources/samples/?hl=en
- More Responsive Patterns: https://bradfrost.github.io/this-is-responsive/patterns.html

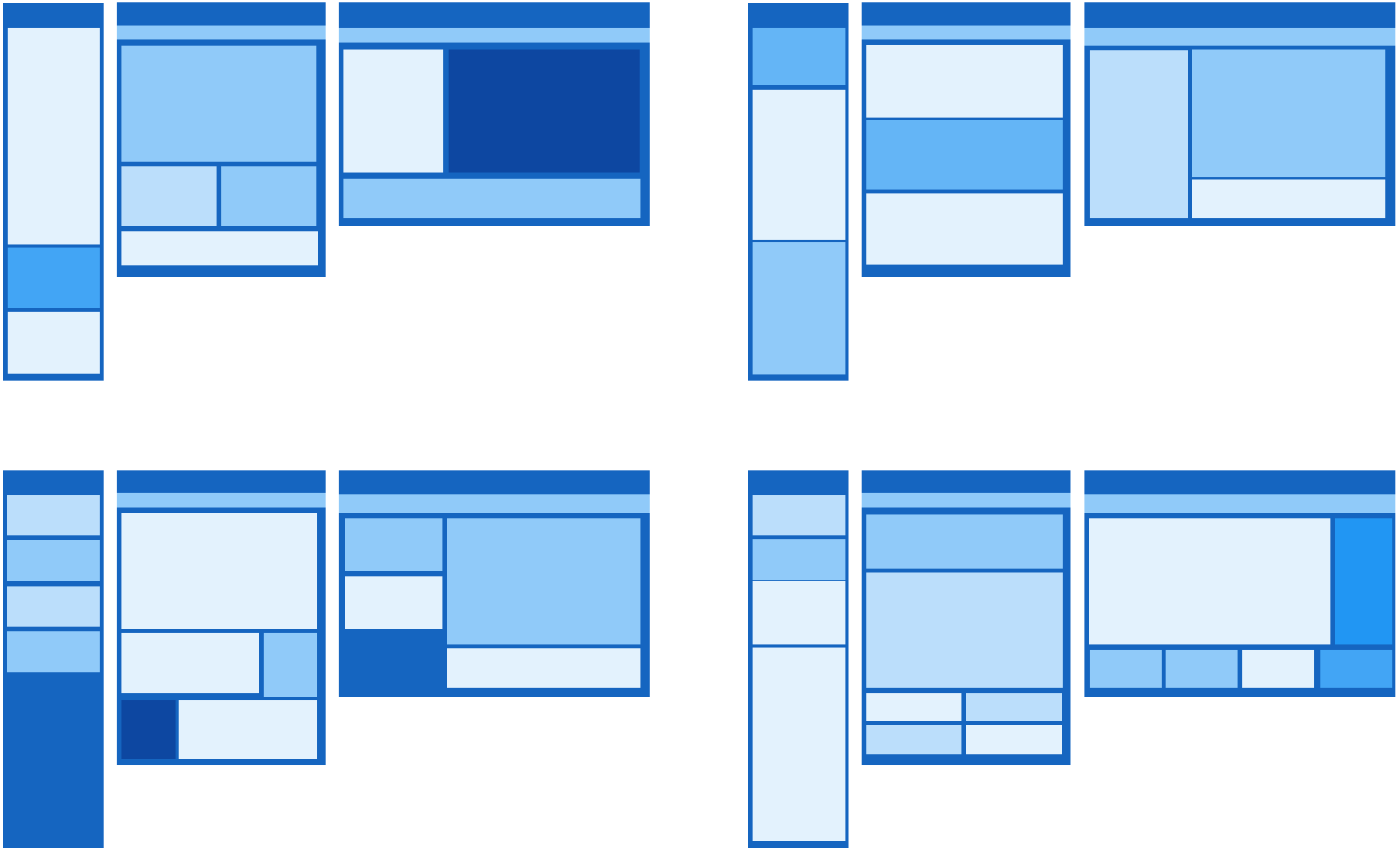


Fig. 1 Examples of basic responsive layouts across devices

Responsive Grids

Defining Dimensions for Responsive Content

Layout grids consist of a series of rows and columns that help establish an organizational framework. A standard full responsive grid view contains 12 columns with a total width of 100%. Portrait mobile phones typically contain 4 columns, tablets 8 columns.

CORE ELEMENTS TO CONSIDER WHEN DESIGNING FOR RESPONSIVE:

- Media Queries – Code that checks the media type declaration and media feature expressions. This can reveal the height, width, resolution and aspect ratio of a device.
- Fluid Grids – The responsive layout coded to various breakpoints rather than to fixed pixels.
- Flexible Content – Even if the layout is responsive, images and video need to scale as well.

Fluid grids are not responsive in and of themselves. You cannot infinitely compress them. Rather the number of columns vary according to need. An adaptive grid focuses on establishing consistent margin and gutter widths.

More on media queries: https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

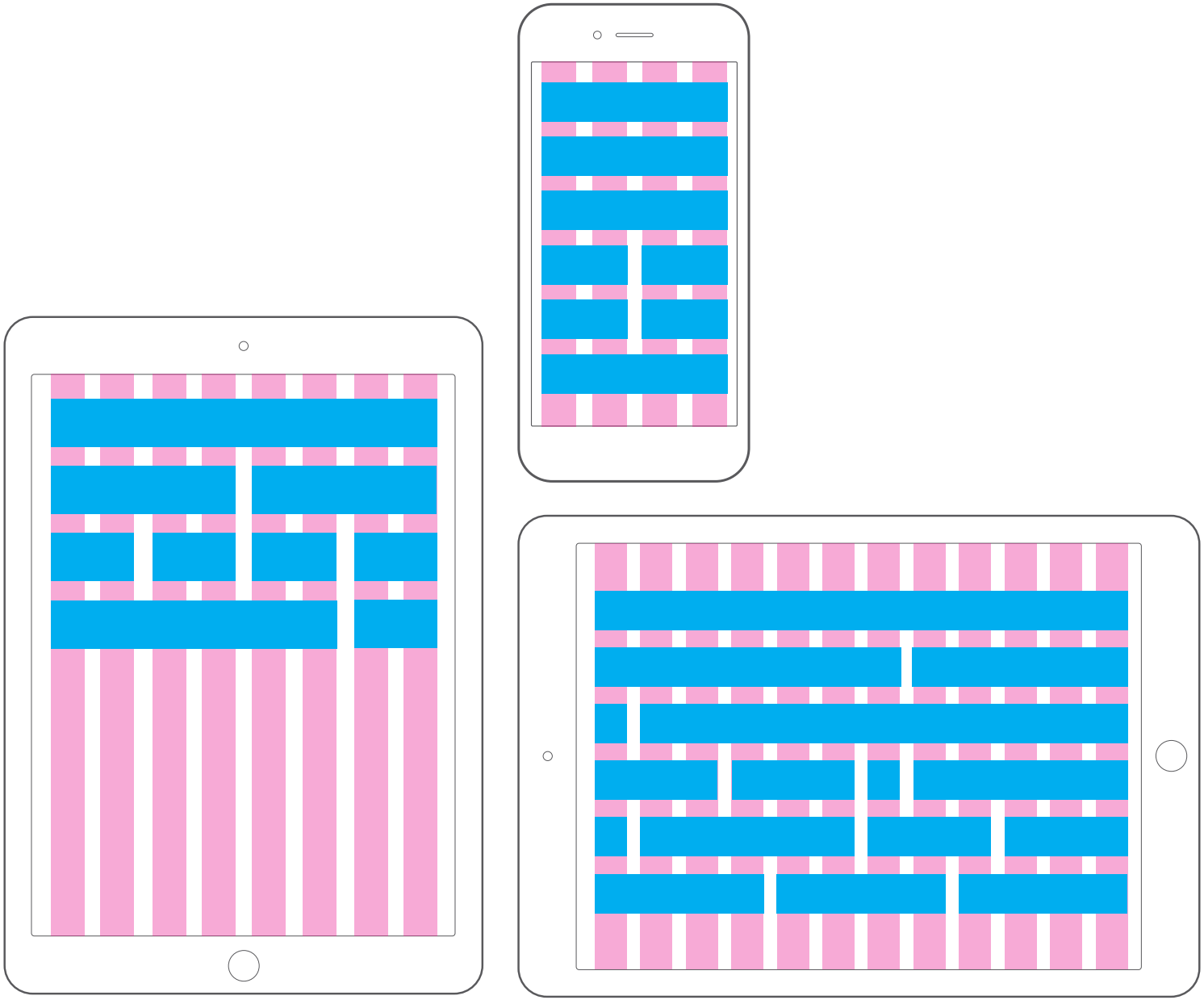


Fig. 1 Examples utilizing columns

Responsive Grids

Baseline Grids

In the previous chapter you laid content areas out in large blocks according to responsive columns. You still need to establish your grid lines. The horizontal grid (AKA, a baseline grid) refers to the line that your typography will rest on. Once that is established, we can start on the details of determining the margins and padding of the type and icon elements. An example of a simple default grid is an 8 px square baseline with type aligning to a 4 px baseline. This can apply to all mobile, tablet and desktop designs.

Vertical height determines the height of the content, and horizontal margins determine the spacing. Once these are established there is the inner padding, also referred to as key lines. When addressing responsive grids and typography, many professionals use EMs. Pixels and EMs are both units of measurement for web elements. However pixels are absolute and EMs are relative. One can calculate EMs by dividing the desired PX value from the inherited PX value. For simplicity's sake, we will be staying with pixels until further updates.

More resources on PX to EM conversions: http://pxtoem.com

Fig. 1 Baseline grid featuring text

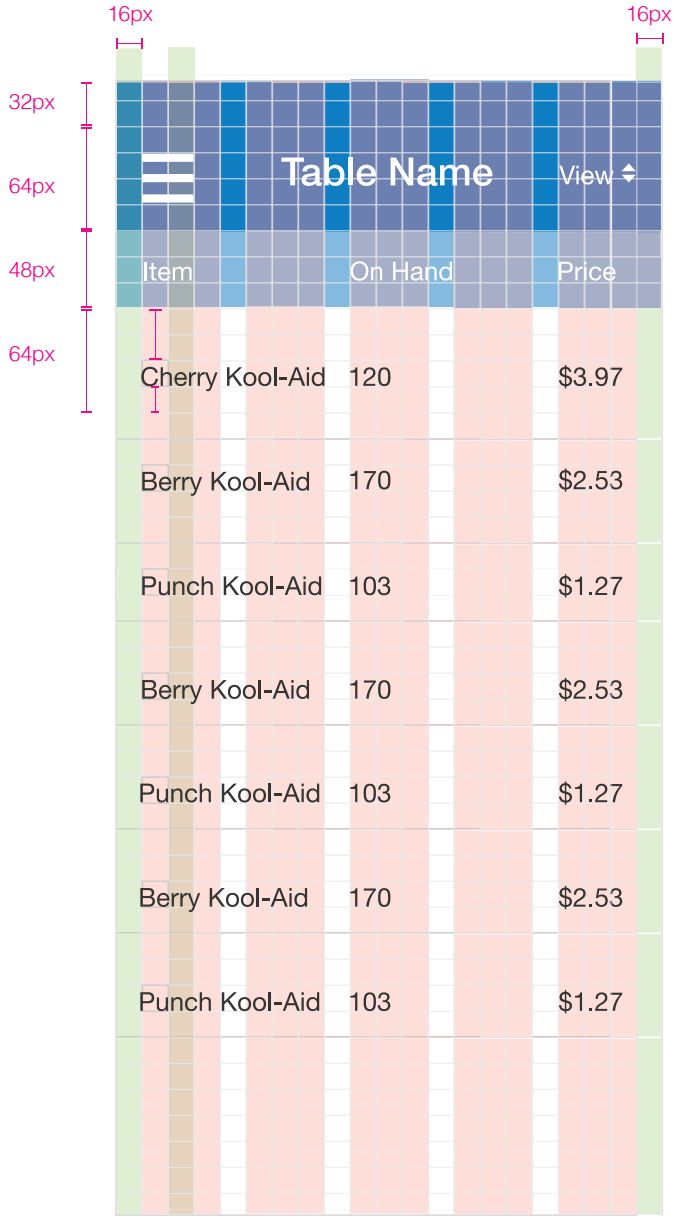


Fig. 2 Baseline grid featuring margins and vertical spacing

Responsive Grids

Breakpoints

Breakpoints are the target points at which your content responds to changes in the page width and height. Do not design breakpoints based on a specific device, but react to the page size with a percent fill for best elasticity. In addition to major breakpoints, add some padding and implement custom breakpoints to help content feel more natural.

STANDARD BREAKPOINTS INCLUDE

- 480, 600, 840, 960, 1280, 1440, and 1600 px

STANDARD MARGINS AND GUTTERS INCLUDE

- 16 pixels wide

These are “optimal examples,” and not all situations may apply.

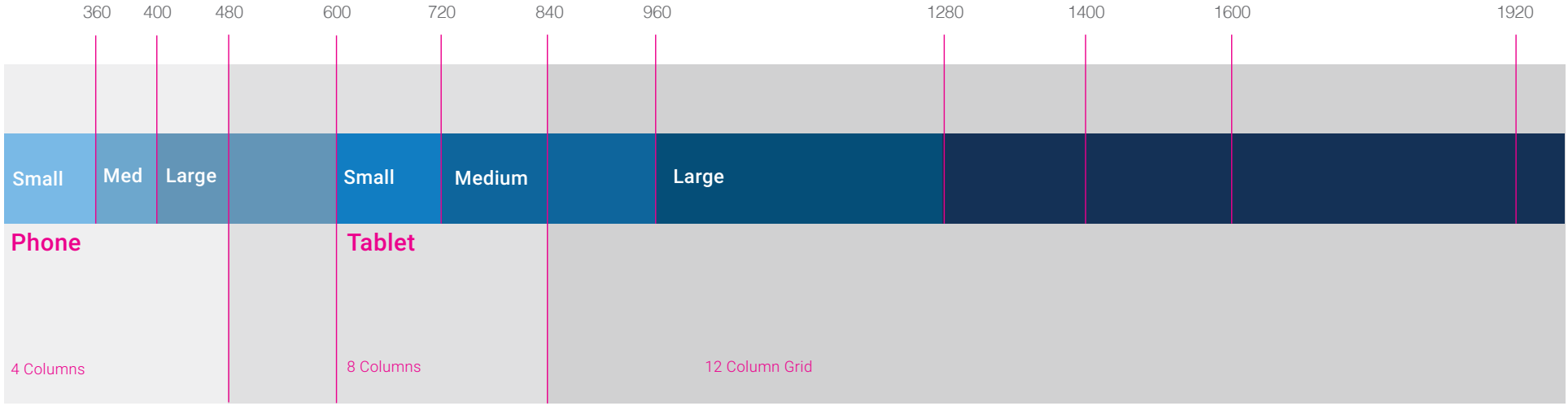


Fig. 1 Scale of different breakpoints

Navigation

Creating Intuitive Navigation

The best navigation makes an application feel seamless. It requires little effort for the user to understand where in the app they can go, and what they will find when they get there. Navigation is most often noticed by the user when it does not meet their expectations.

As app developers, it is our job to create a navigation experience that most closely resembles the mental model the end-user has established for the task at hand.

3.0 Navigation

Header

Tells the User where They are and What They can do

The header indicates to the user where they are, and offers a space for high-level functionality.

Every page has a header and title **(Fig. 1)**. If the page is considered an internal page, a back button exists as well.

The header can also include a space for actions, such as search and sort. An options menu can also drop down to reveal additional actions if needed.

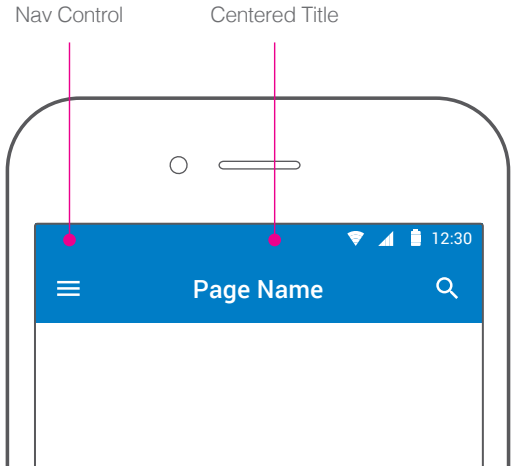


Fig. 1 Mobile best example

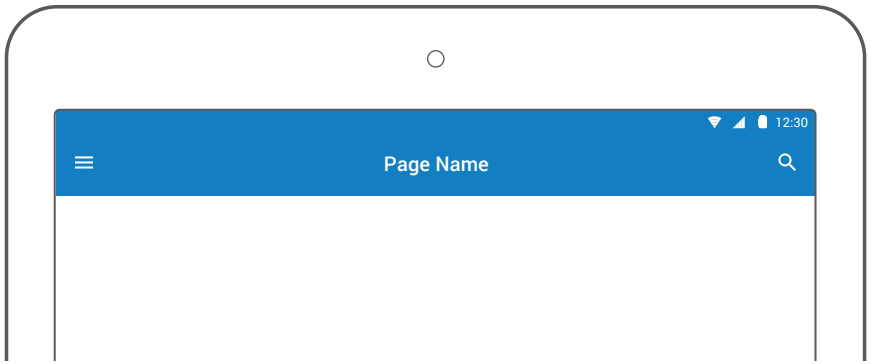


Fig. 2 Tablet best example

Header

Best Practice

DO

- ✓ Limit the character count of page names to ensure they fit on smaller devices.
- ✓ Use text labels for app-unique functionality.

DO NOT

- ✖ Change the height of the header on different pages.
- ✖ Have a back button that is disabled or broken. The back Have a back button that is disabled or broken. The back button should always return the user to the last state they were in.

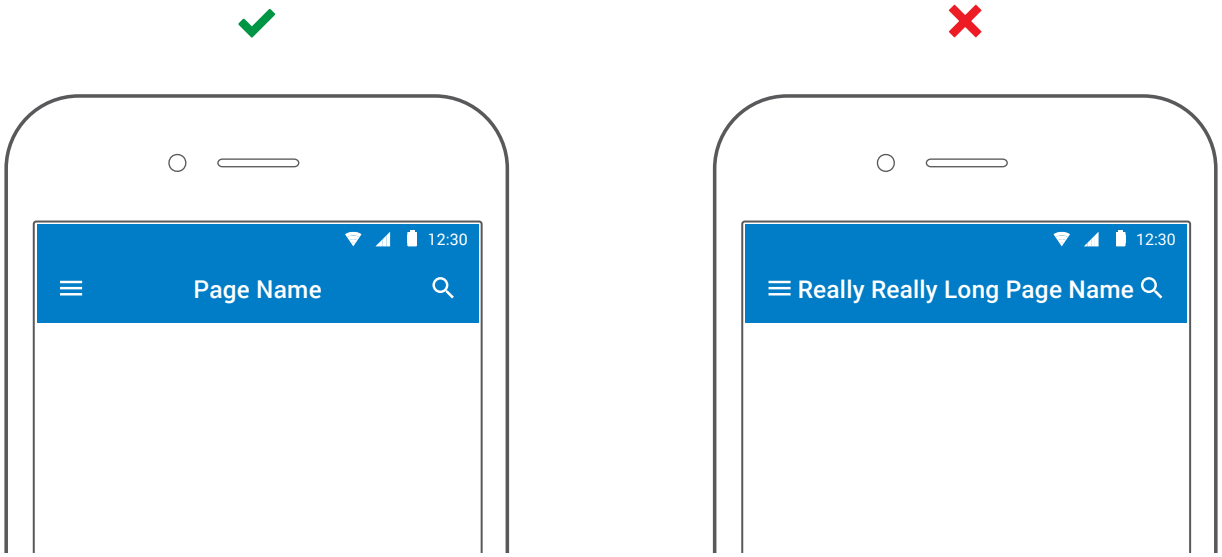


Fig. 1 Keep page names to a minimum of 20 characters to prevent overcrowding

Header

Variations

The header can also include a space for two actions, such as search and refresh (Fig. 1). If more than two actions are needed in the header, use a context menu instead (Fig. 2).

Actions that are necessary, but might not be used every time the user accesses the page should be placed inside a context menu. Example actions include: Print, Edit, Clear, Cut, Copy and Paste.

Context menu: A context menu is a space reserved for additional functionality within an app. On desktop applications, the context menu is normally accessed with a right-mouse click. On mobile, the vertical ellipses icon represents the context menu.

Fig. 1 Mobile Context Menu

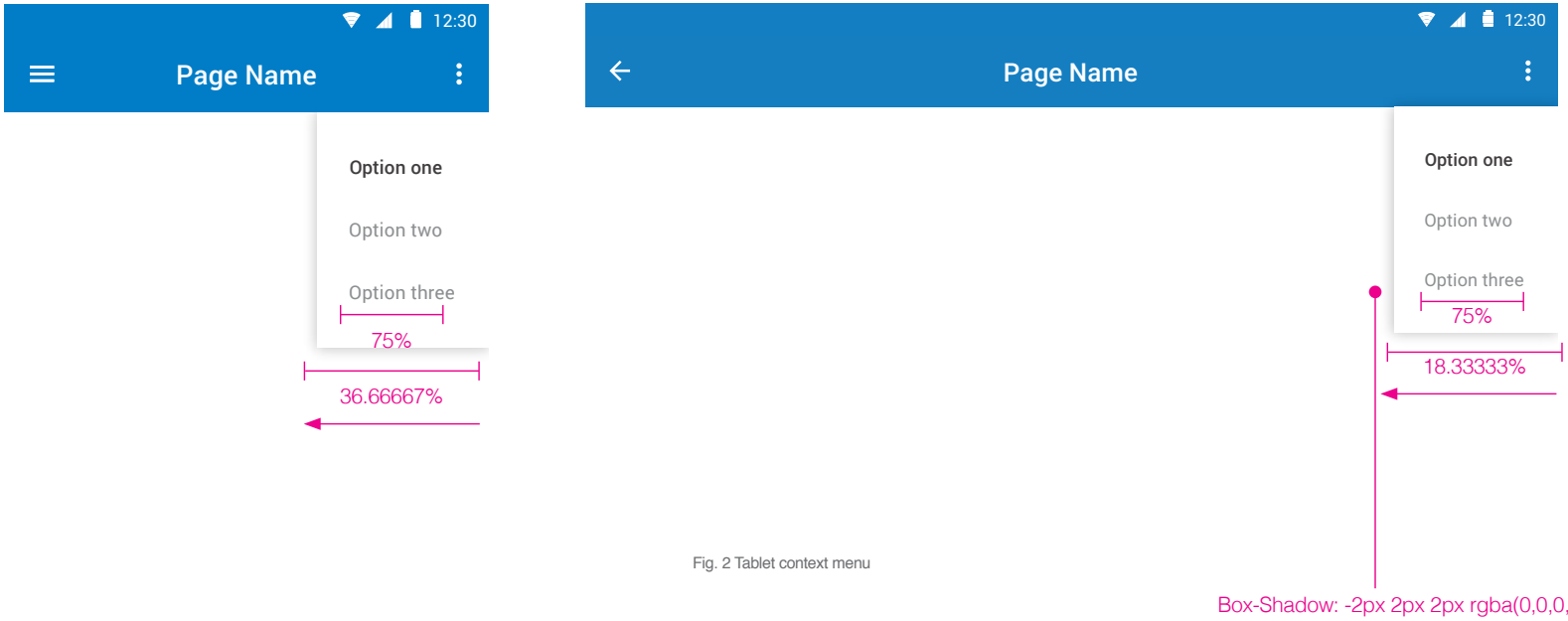


Fig. 2 Tablet context menu

Navigation Menus

How a User Moves Through Your App

Also known as “The Hamburger Menu,” this navigation pattern is commonly seen across websites and mobile applications. Tapping the menu icon causes a drawer of links to slide into view. Tapping the menu icon again, or any area outside of the drawer, will cause the menu to slide back out of view. The navigation drawer always begins in the closed position when an app is opened.

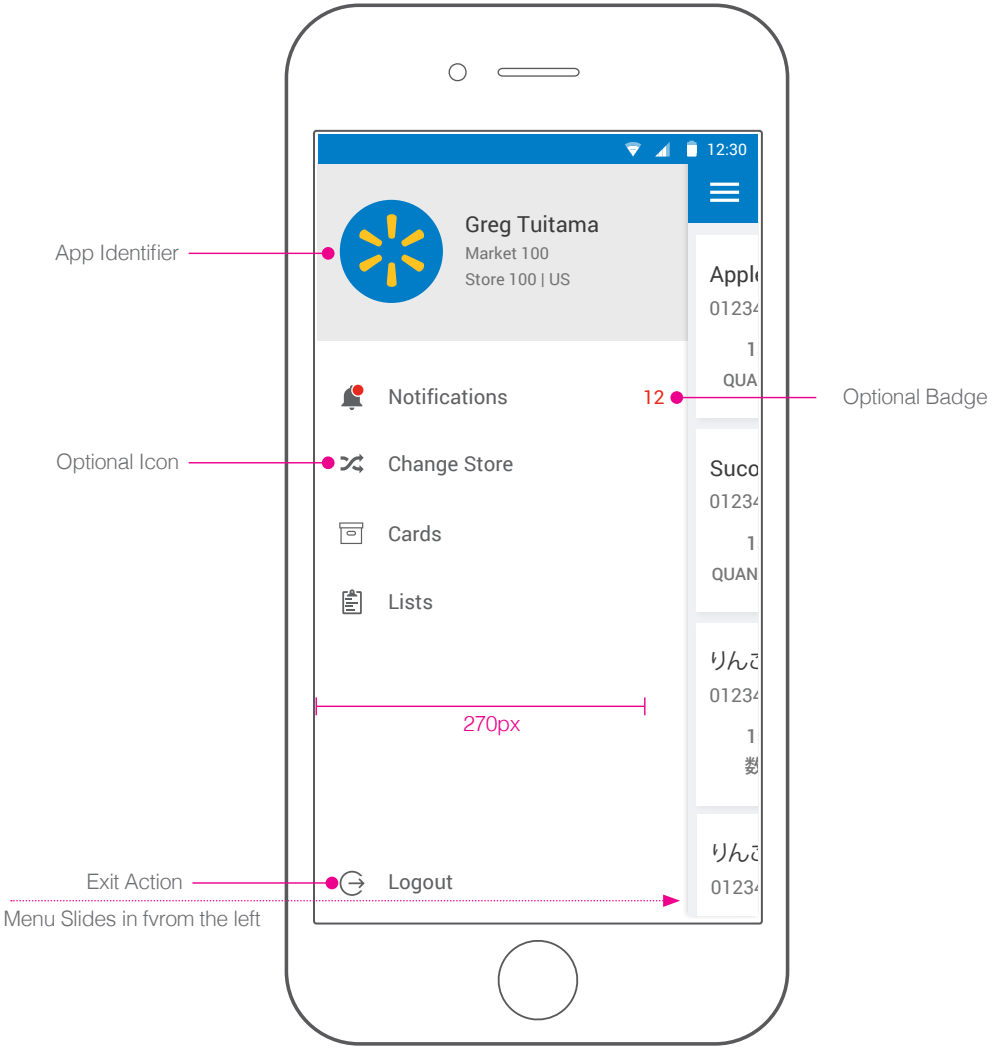


Fig. 1 Mobile example

Navigation Menus

Best Practice

DO

- ✓ Include notification badges to direct user's attention when needed.
- ✓ Show the current nav item the user is currently in.
- ✓ Make it easier to tap menu items with a large hit target.

DO NOT

- ✖ Combine or mix navigation styles.
- ✖ Put checkboxes, radio dials or other buttons inside the nav menu.
- ✖ Use only icons. Menu items must have a label.

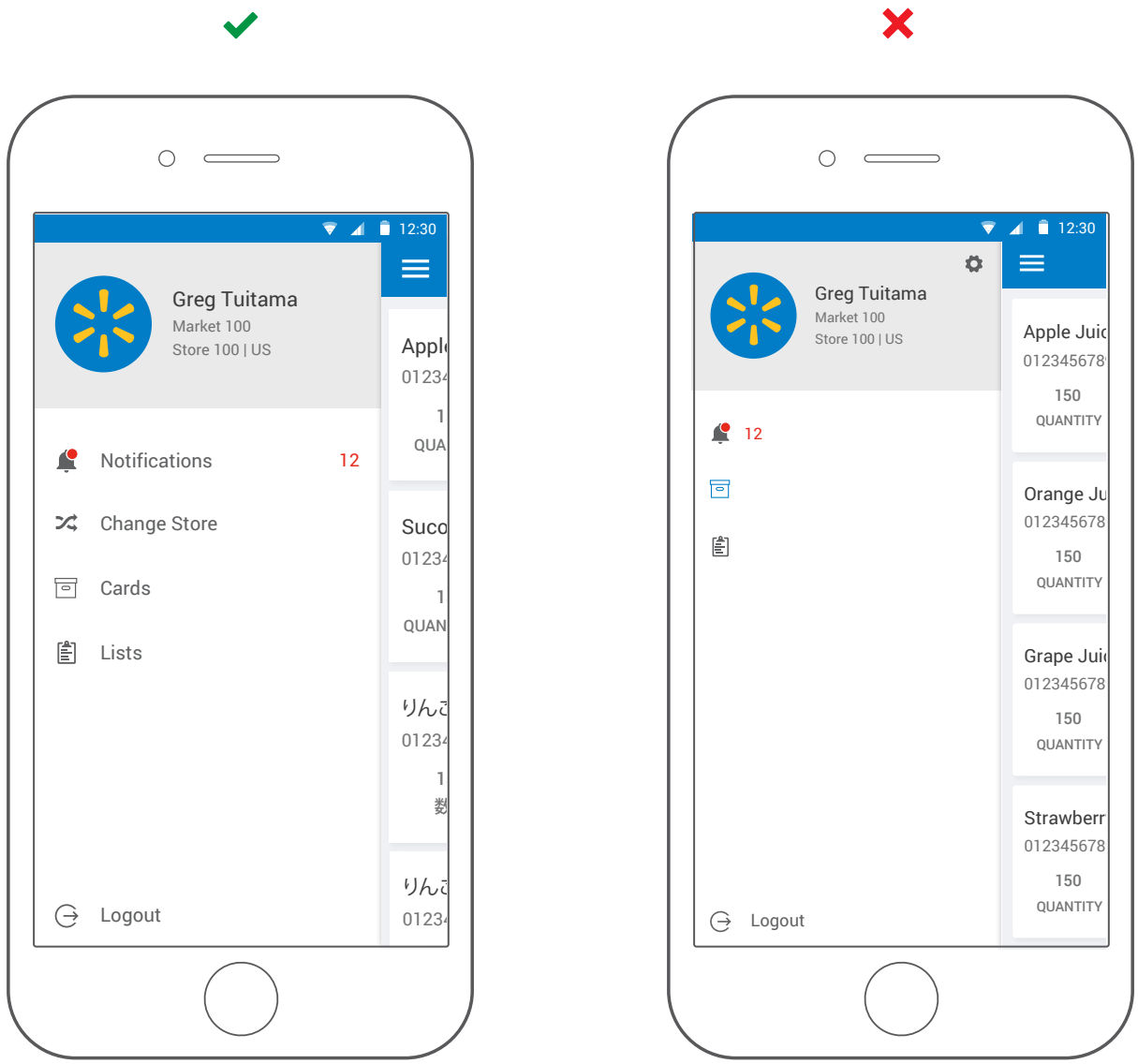


Fig. 1 Use text labels for menu items

Navigation Menus

Variations

The Nav-bar Apps that are centered around three to five pieces of core functionality might be best suited for a nav-bar. Nav-bars are a fixed, single row at the bottom of the user's viewport, with content scrolling behind them. Each button on the nav-bar represents a section the user can visit, with a label and icon.

The persistence of the nav-bar makes it easy for users to switch between different pages, making it a good choice if the user is likely to repeat many actions within your app. However, the persistence also means that valuable page real estate is lost. Consider if the benefits of the nav-bar outweigh the negatives.

A nav-bar is not the same as a tab-bar. While a nav takes the user to different pages, a tab is a series of pre-defined filters applied to a single set of content.

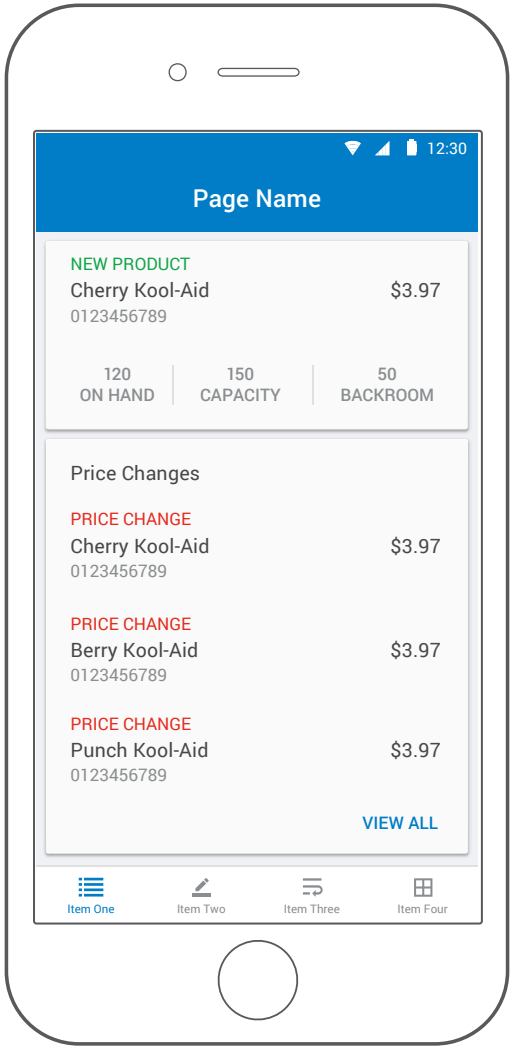


Fig. 1 Navigation bar

Back Button

Creating a Consistent Experience

Whether or not to use the navigation menu button or the back button is dependent upon whether a page is considered “primary” or “child.” A primary page is any page that is directly accessed via the menu navigation. Pages that are not part of the main menu structure, which can only be accessed through primary pages, are considered “children” pages. Primary pages use the navigation menu icon to move between them, and children pages use the back button to always return to a parent page before moving to a different page (Fig. 1).

The back button is placed on the left-side of the header, making it easy for the user to access at any time.

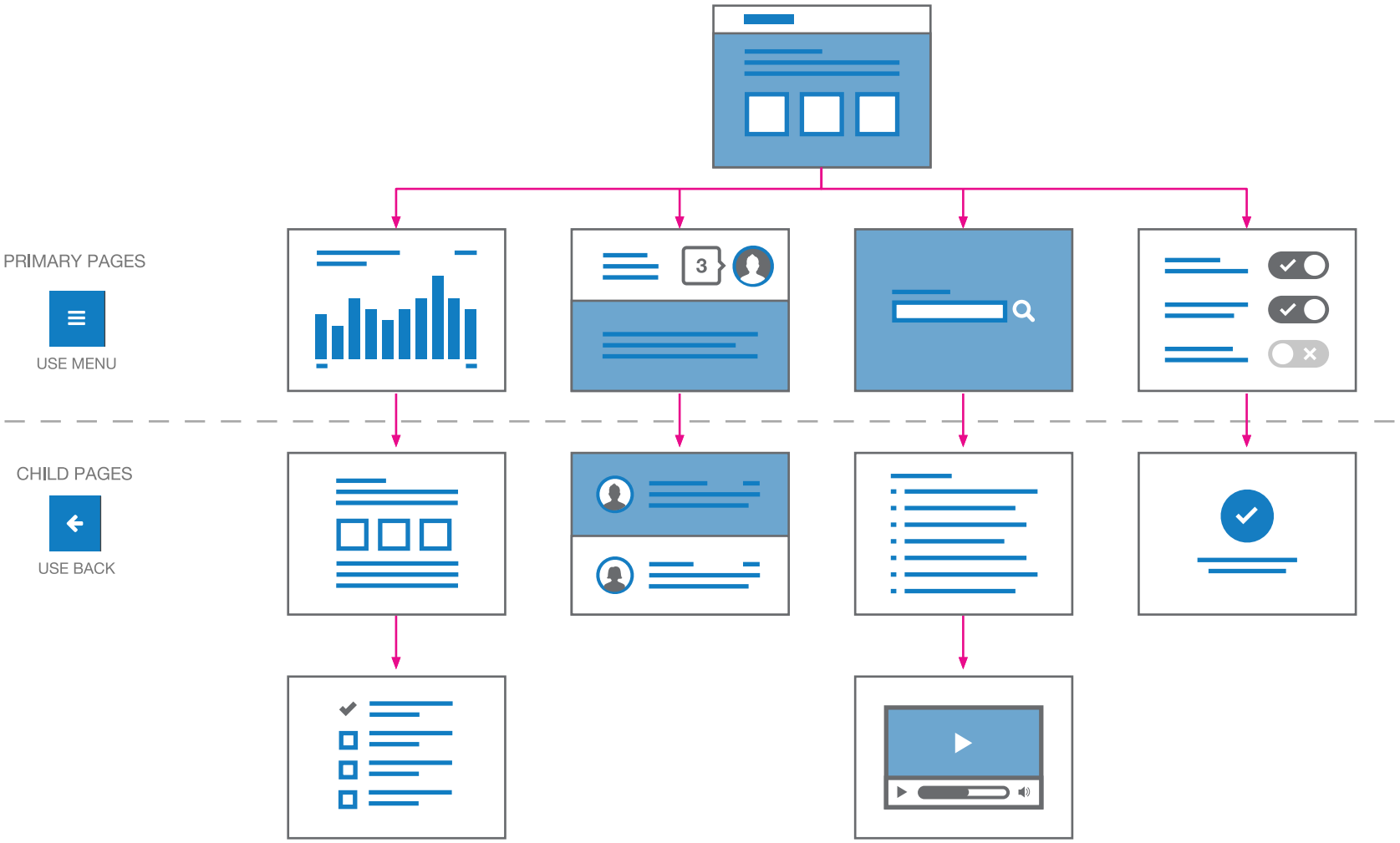


Fig. 1 App site map

Back Button

Best Practice

DO

✓ Program the back button to take the user back to the previous page they were viewing, even if it is not the direct parent of the child page they are currently viewing.

DO NOT

✖ Add back buttons within your content.

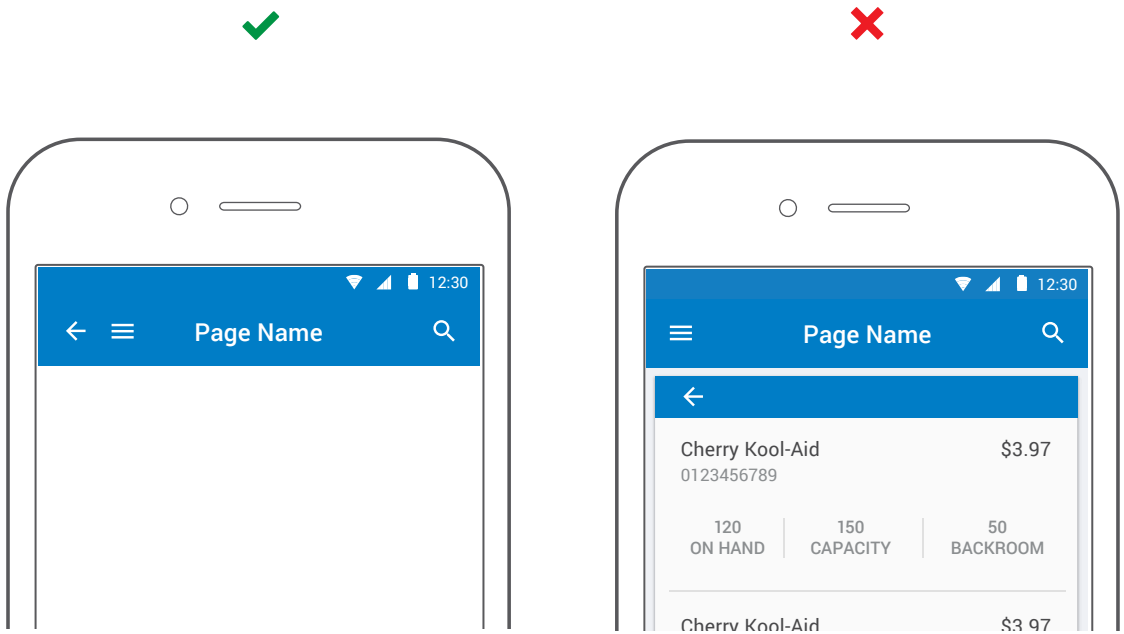


Fig. 1 Back buttons should only exist within the header

Back Button

Variations

In some instances it might be necessary to include both the back button and menu icon in the header. In these cases, limit right hand actions to a single.

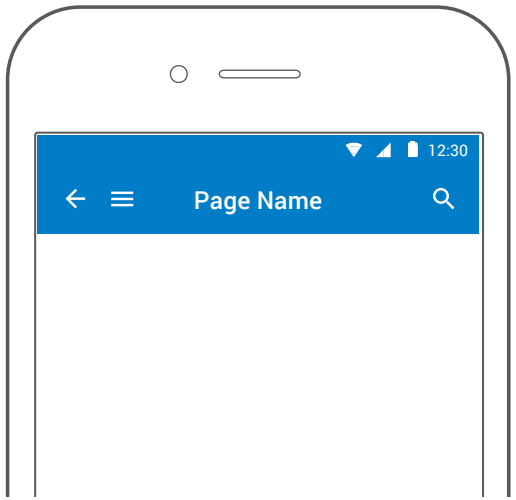


Fig. 1 Back buttons should only exist within the header

Tabs

Pre-set Filters

Tabs are a group of pre-filtered controls that are fixed to the header. If the user needs to frequently switch between different views or functionality related to the same piece of content, tabs can be a good solution.

Each tab has a text label that succinctly describes what the user will find if they tap the tab. When a user taps a tab, the viewport updates to reveal the content found within that tab.

On larger page devices, tabs by default are left-alignedv to the side of the page.

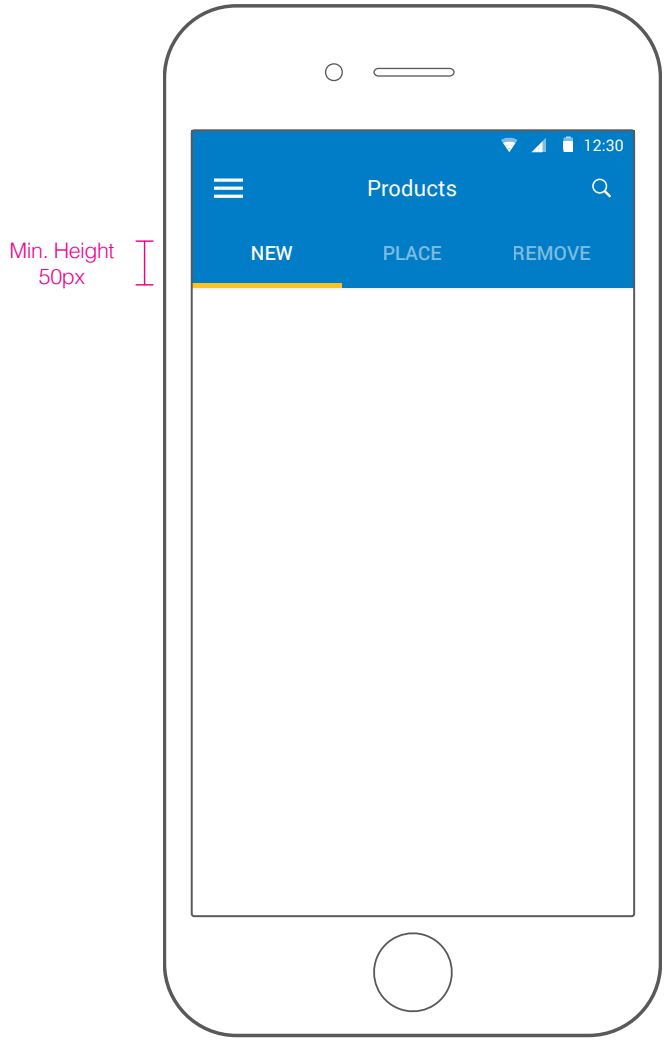


Fig. 1 Mobile example

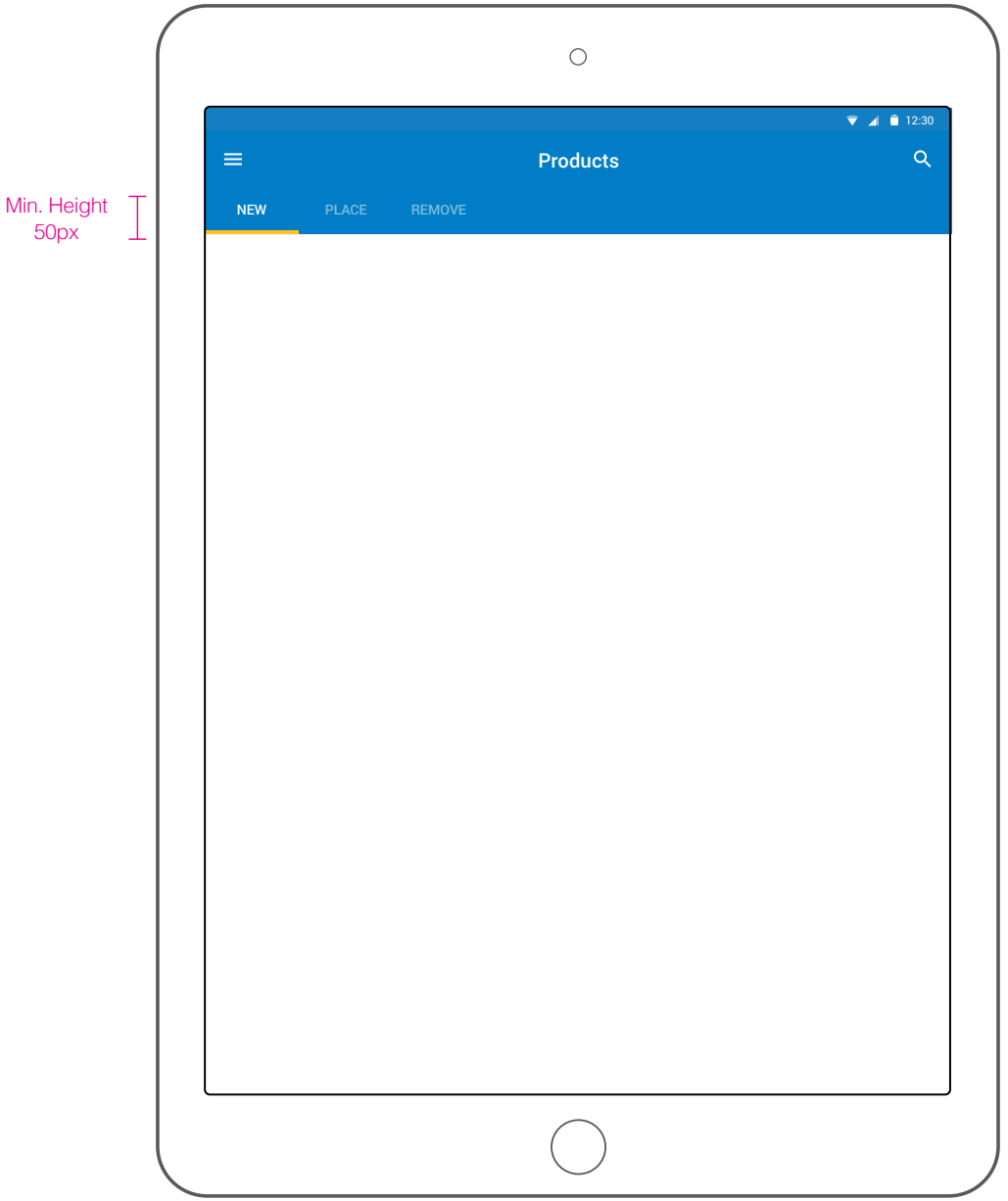


Fig. 2 Tablet example

Tabs

Best Practice

DO

- ✓ Ensure each tab in a set has a fixed length.
- ✓ Connect tabs to match a user's mental model of how they approach the content.
- ✓ Always fix the tab bar to the user's viewport.
- ✓ Accommodate a long text label by increasing the height of the tab container.

DO NOT

- ✖ Use just icons in tab labels. All tabs require text labels.
- ✖ Use tabs to switch between unrelated pieces of content or functionality.
- ✖ Use swipes to navigate between tabs.
- ✖ Nest tabs or create a secondary level of tabs.

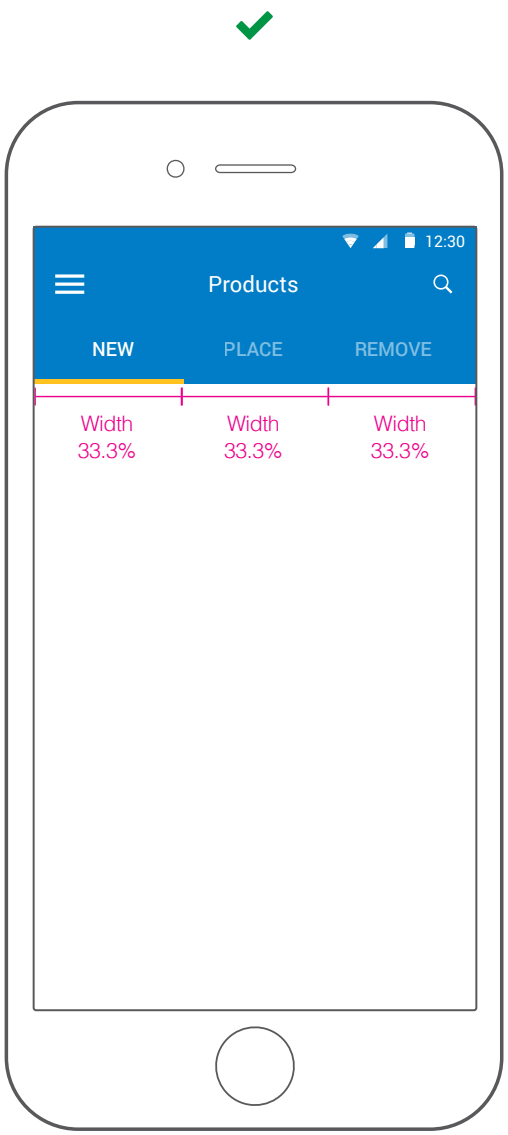


FIG. 1 Tabs should have a fixed, equal length

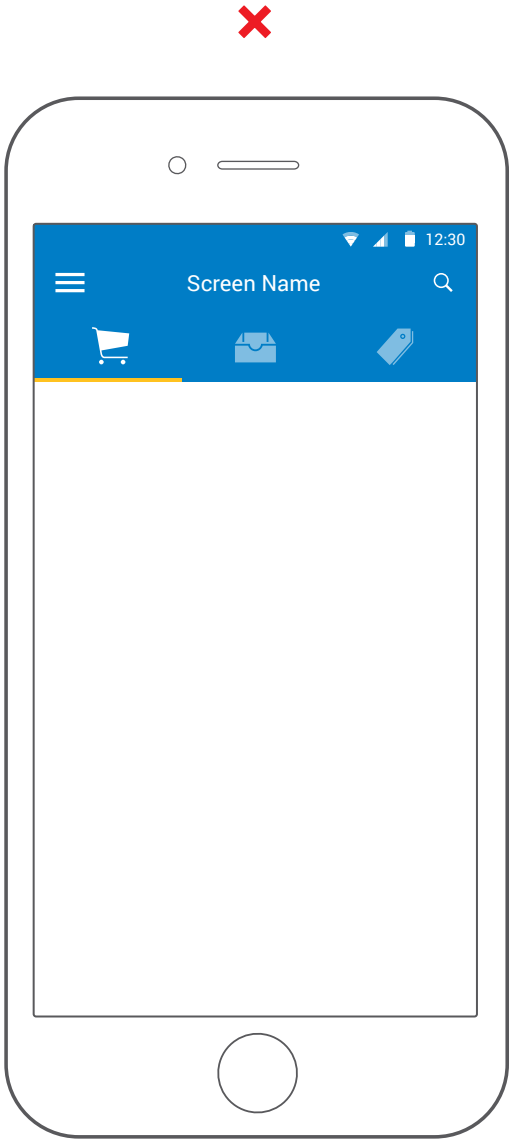


FIG. 2 Tabs must include text labels

Tabs

Variations

Tabs can scroll to the left when the length of the tab-bar is longer than the width of the viewport (Fig. 2).

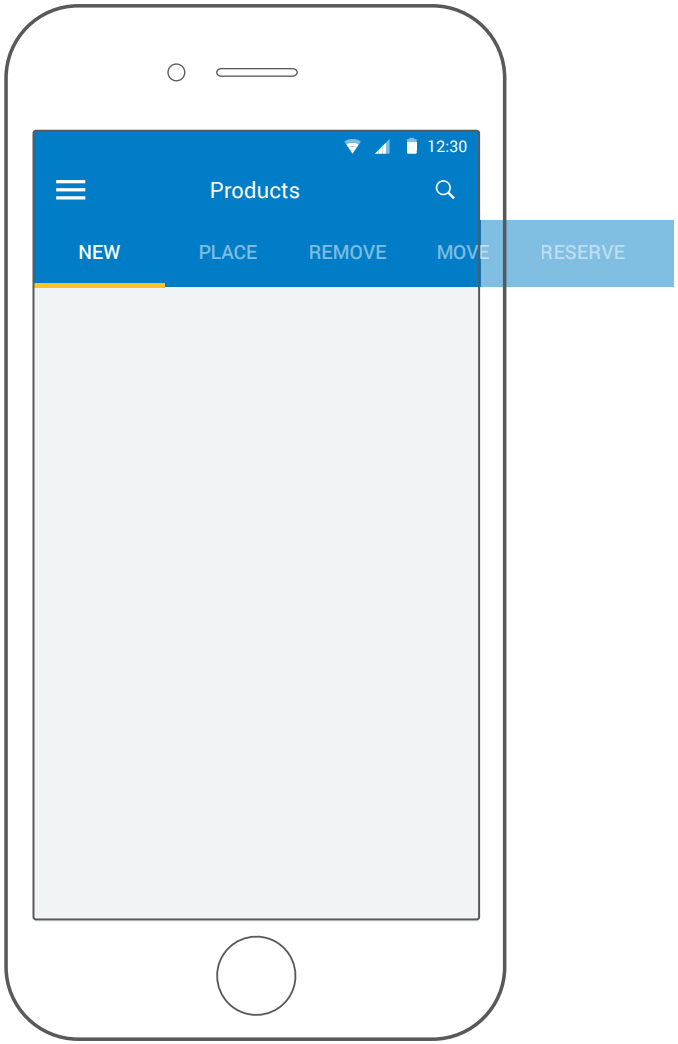
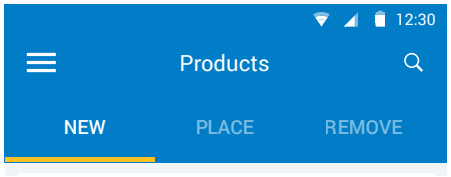
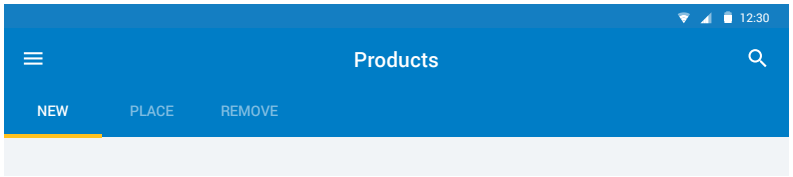


FIG. 1 Horizontal scroll





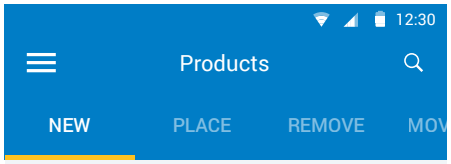


Fig. 2 Over flow tabs

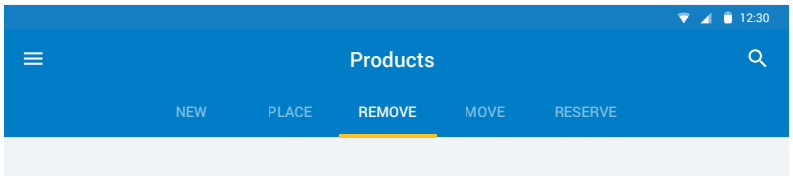


Fig. 3 Centering a tab to its container is also an option

Interactions

The Illusion of Touching Tangible Elements

Interaction is the art form of communicating to users whether or not they are accomplishing their goals. Feedback encourages users, providing a sense of security that they are making a worthwhile investment of their time, and building trust and a rapport with your program. This means to complement the experience, not complicate it.

Even if the user commits an error, as long as there is quick communication on how it happened and an easy way to undo it is immediately provided, you can still support a good experience.

4.0 Interactions

Touch and Gesture

Reducing Steps to Complete Actions and Eliminating Button Clutter

A key to creating an optimal product is understanding and predicting user behavior. Think in terms of time, dimension and animation to answer these questions: How long does an action that is triggered by the user take to complete? Is the space sufficiently sized and spaced for interaction? How is that visualized? Are the mechanics intuitive?

- Note the more visual clues you remove, the higher the experience complexity becomes.
- The mobile example displayed here is optimal reach for a right-handed grip.

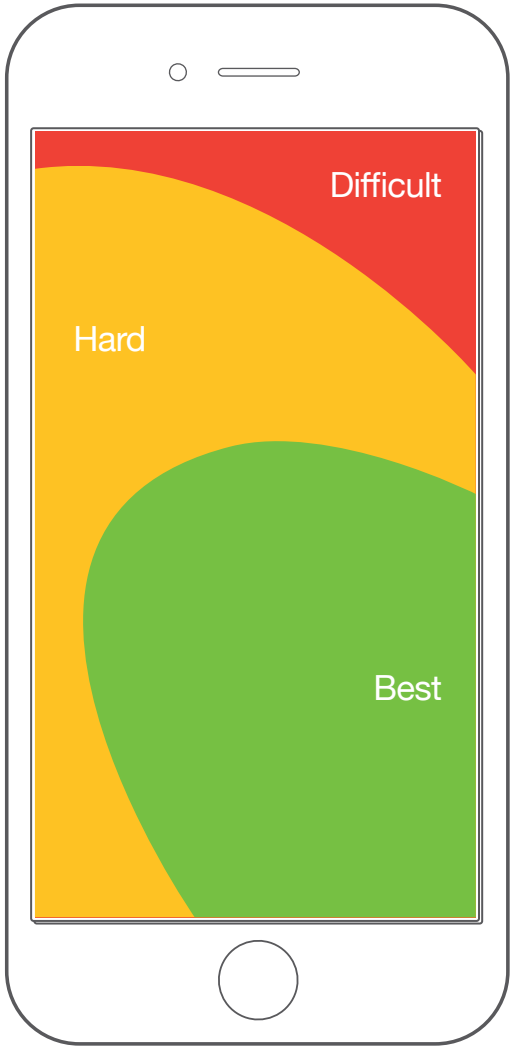


Fig. 1 Optimal reach mobile (best example)

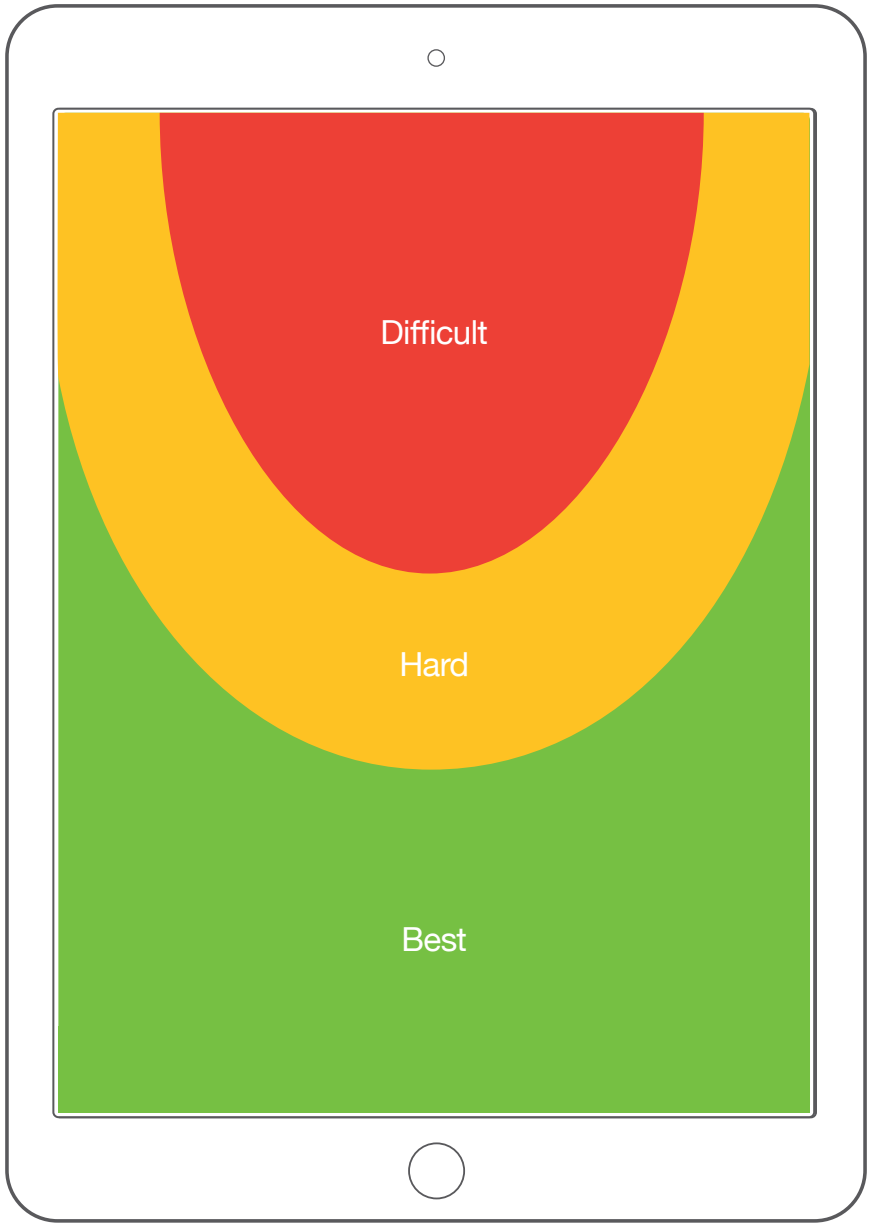


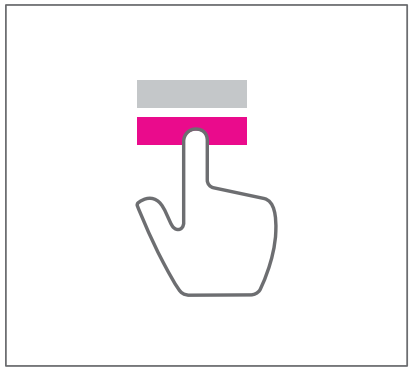
Fig. 2 Optimal reach tablet (best example)

Touch and Gesture

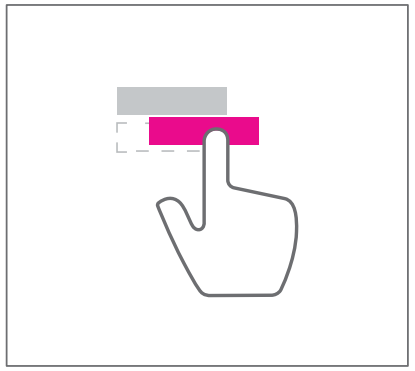
Variation - Visual Confirmation

Providing visual feedback of interactive elements helps build user confidence in executing functions. Elements can fulfill this by changing color, size or moving. A gesture may have multiple results based off of the context; visual confirmation can provide context clues on what to expect. Examples of how to do this are displayed on the right. Feedback alerts do not have to be pink.

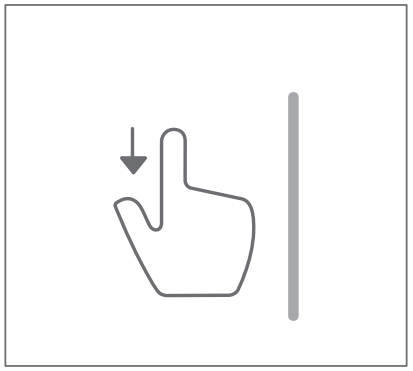
Make sure all interactive controls have text labels, tooltips, or placeholder text to indicate their purpose.



Color to Indicated Selection



Show Contextual Clues



Indicate Scroll

Fig. 1 Visual feedback

Touch and Gesture

Variation - Touch Library

Touch gestures are the actions a user's fingers perform on the page **(Fig. 1)**. It is important to establish consistency for the user based on actions that come naturally. The simpler the action, the better, as users often mistakenly tap twice, use knuckles and accidently tap with multiple fingers when interacting with their pages.

Note the more visual clues you remove, the higher the experience complexity becomes.

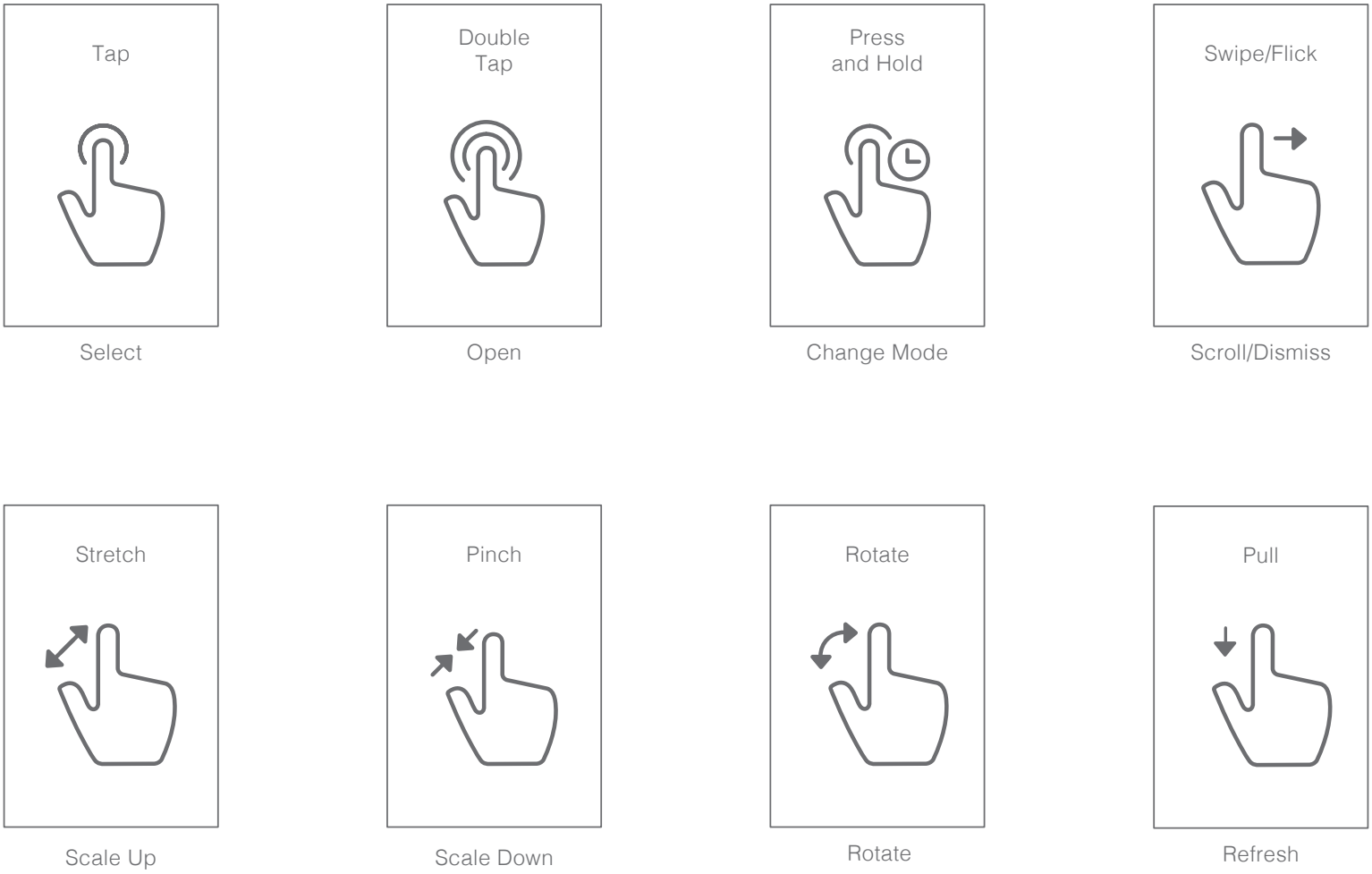


Fig. 1 Touch library

Touch and Gesture

Best Practice

DO

- ✓ Always include easy touch targets and optimal interaction spaces.
- ✓ Maintain an awareness of how the user will be holding the device.
- ✓ Ensure access to the right tools at the right time.
- ✓ Make tasks obvious and in context to the task at hand.
- ✓ Always keep consistency in mind.
- ✓ Keep interactions reversible.

DO NOT

- ✖ Rely on compound gestures such as the double tap or press and hold.
- ✖ Include timed interactions.
- ✖ Include elements in the center of a page where interactions will cover content.
- ✖ Overlap touch areas.

Fig. 1 Hover states do not exist on mobile. Use visual Clues

Touch and Gesture

Variation - Touch Targets

Touch target areas are the items and the space around them that support touch manipulation. Touch areas are able to sense the central point of a user's touch. Care must always be taken to ensure there is a large enough touch area around items, spacing between items and visual confirmation clues to actually use. This helps take finger size and human error into account. This is important for critical actions such as saving or deleting files.

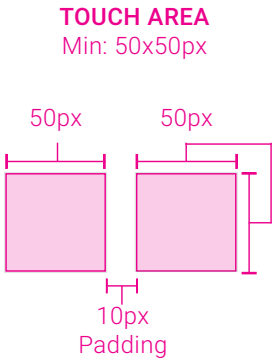


Fig. 1 When touch is critical (i.e.; save and delete)

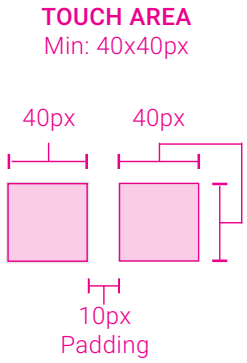


Fig. 2 Acceptable average

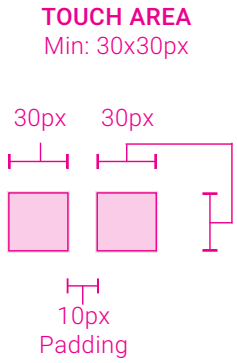


Fig. 3 Minimal must only if not a critical function

Form Fields

How Information Enters the System

Users input information into systems through forms. Forms must be easy to understand, use and modify.

A form field requires a form title and “ghost” text that can provide additional support **(Fig. 1)**.

When the user taps anywhere inside the border of the field box, the box becomes “active” to help the user understand what field they are modifying **(Fig. 2)**. If the user changes focus to another element, or taps “enter” on their on-page keyboard, the text field becomes read-only **(Fig. 3)**.

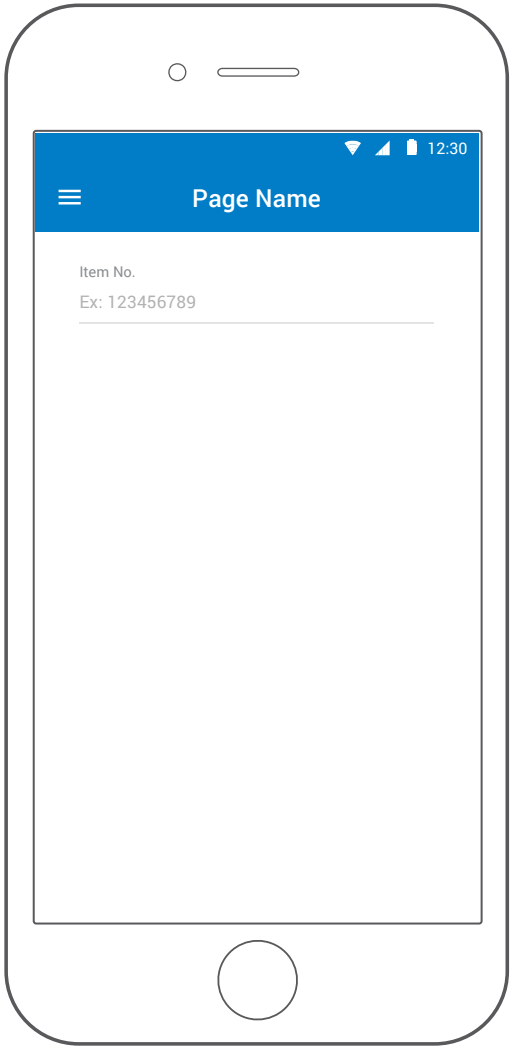


Fig. 1 Default

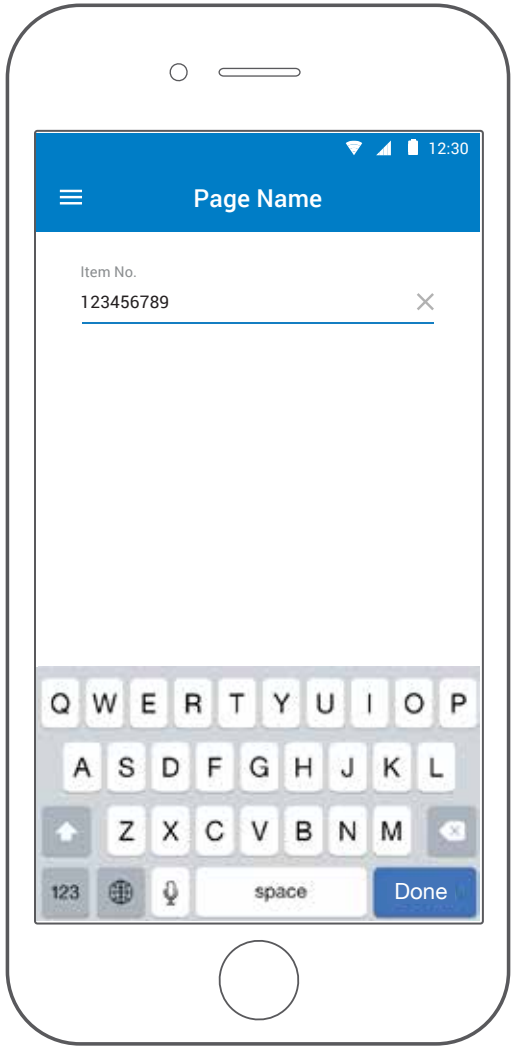


Fig. 2 On focus

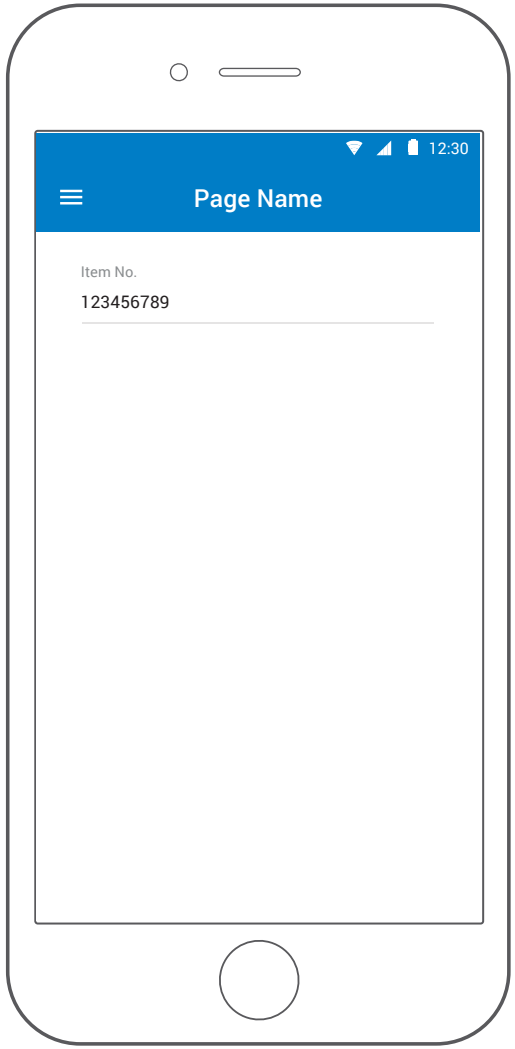


Fig. 3 Off focus

Form Fields

Best Practice

DO

- ✓ Give enough room for users to be able to select the field with a tap. A field box should have a height no shorter than 75px.
- ✓ Group related fields together for long forms.
- ✓ Program keyboard listeners properly to allow them to move between form fields with the “tab” or “next” key.

DO NOT

- ✖ Use only hint text to label a form field. Form fields always need a label outside of the field for maximum clarity.
- ✖ Mark fields as “required.” Only indicate which fields are optional.

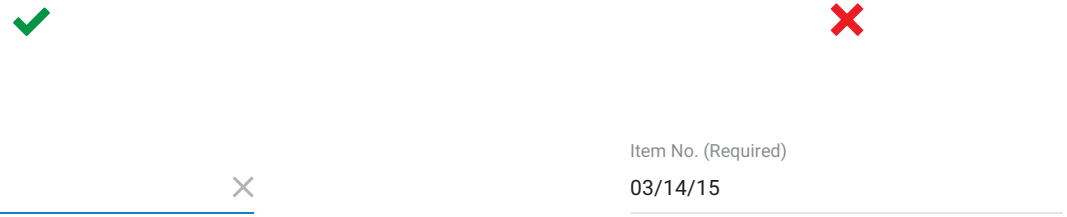


Fig. 1 Only note when a field is optional for the user to fill out. Never indicate a field is “required”

Form Fields

Variations

A form field with a limited number of options to choose from should use a dropdown menu. If the field requires specific formatting, include the formatting characters to help the user input information in the correct syntax.

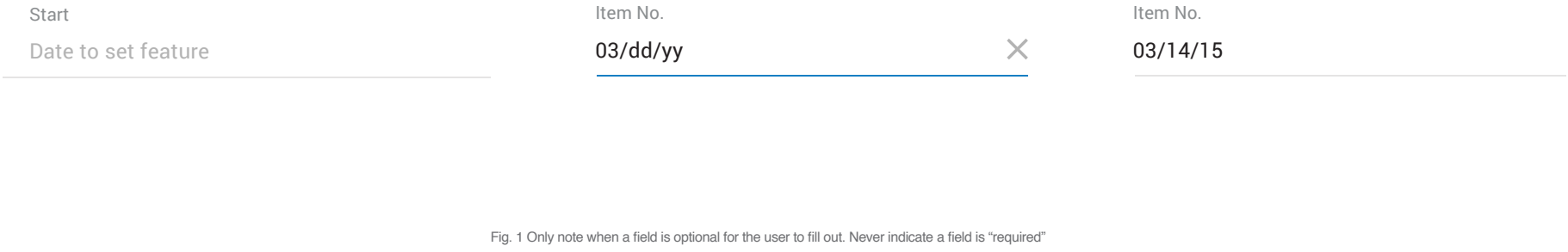


Fig. 1 Only note when a field is optional for the user to fill out. Never indicate a field is “required”

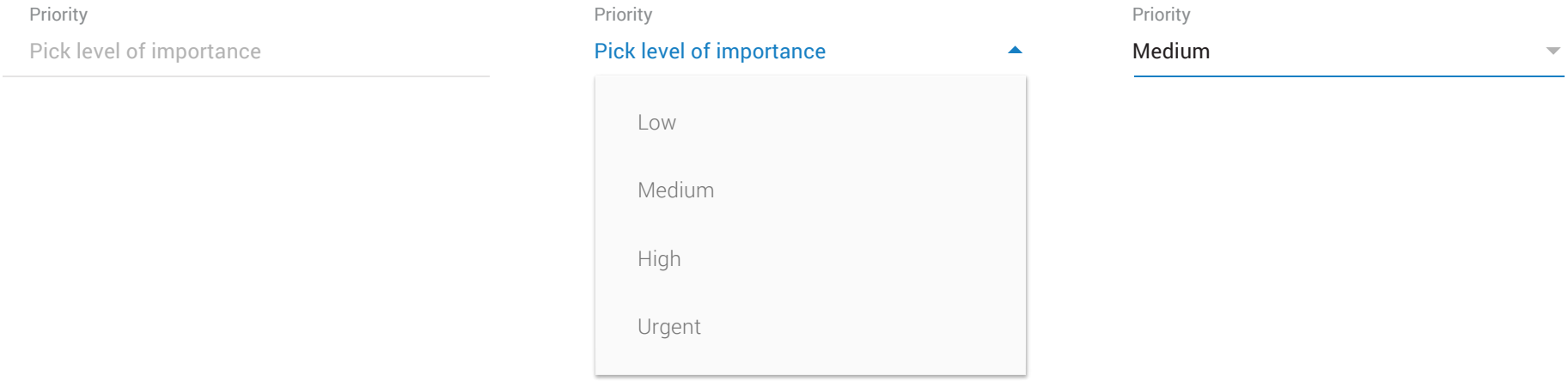


Fig. 2 Default field

Fig. 3 On focus

Fig. 4 Off focus

Buttons

A Graphic Element that Initiates an Immediate Action

Buttons must clearly communicate what action a user should expect from them. There should only be one primary action on a page followed by secondary or tertiary action. If there is a positive action it should always be the primary action, then negative, than neutral – in that order. All descriptions should be short and descriptive in all caps. Button elements include the type, state and style. Buttons can be stacked or arranged side by side, but care must be given to their padding to ensure the touch areas do not overlap.

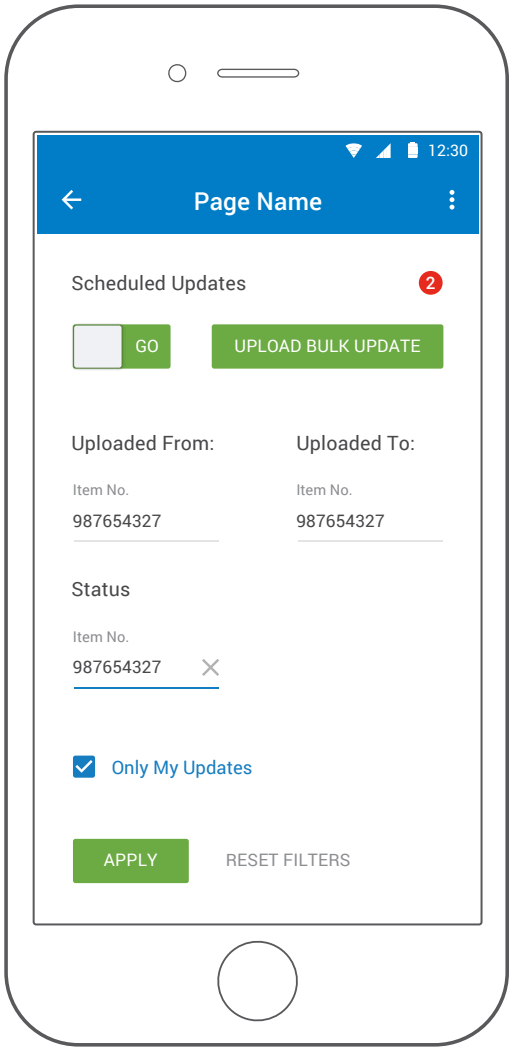


Fig. 1 Mobile example

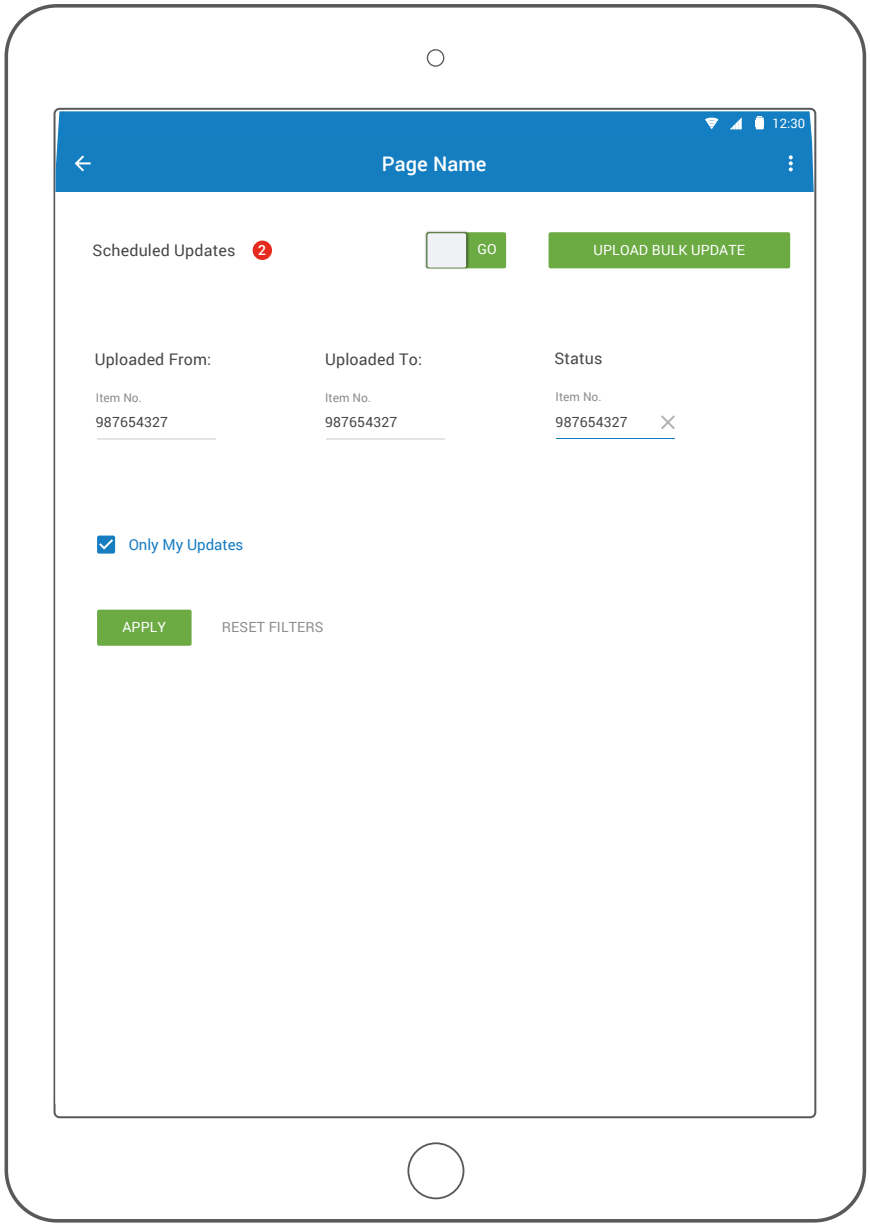


Fig. 2 Tablet example

Buttons

Best Practice

DO

- ✓ Ensure buttons are no smaller than 84 px by 44 px with at least 10 px internal and 10px external padding.
- ✓ Only use icons in specific cases such as app bars and action buttons.
- ✓ Ensure button text is always all caps.
- ✓ Make sure button text is concise and describes the action it performs.
- ✓ Color positive actions a cool color such as green or blue.
- ✓ Remember positive buttons should have a cool color with white text and the highest contrast.
- ✓ Color negative actions warm colors such as orange and red with white text.
- ✓ Remember negative actions should only be solid with white text when there is no positive action.
- ✓ Note that all text-only buttons are acceptable in dialog boxes, but primary action must have a heavier weight.
- ✓ Restrict use to one to two buttons for a user at a time if at all possible.

DO NOT

- ✖ Fill a neutral or negative next to a solid positive.
- ✖ Use a light grey for negative buttons, or the user will mistake it as disabled.
- ✖ Use a button when a link will work.
- ✖ Misshape buttons.
- ✖ Change button text while user in still in the page.

ⓘ Find jQuery animations here http://www.jqueryscript.net/demo/Form-Submit-Buttons-with-Built-in-Loading-Indicators-Ladda

Fig. 1 Positive buttons should be the loudest

Fig. 2 Button labels must be simple and concise

Buttons

Variations - Defining How to Display a Button

TYPES

There are three main functions buttons are used for:

Positive – Creates a change to complete a task, such as save and download.

Negative – Deletes or resets information.

Neutral – Cancels or takes no action to change the main page state.

Positive buttons should have the highest contrast and be a cool color such as green with white text. Negative colors should be warm colors such as orange or red.
Do not use a soft grey for neutral or users will think it is disabled. (Fig. 2)

STYLES

These types are performed with three button styles: solid, outlined and text-only.

Choosing the button style depends on its function, emphasis and location. When we want the user to commit a primary action rely on the solid button. Outlined buttons are for secondary actions with less emphasis. Only use plain text buttons for minor actions such as clearing text or in dialog buttons (in which case the primary action carries a heavier font weight). Show 50% opacity for disabled buttons. It is acceptable to use text-only buttons in dialog boxes.

STATES

Buttons have four states: normal, pressed, loading and disabled.

Show full color for normal, 50% opacity for pressed and grey for disabled. When showing load state, do not stretch or expand the button area.

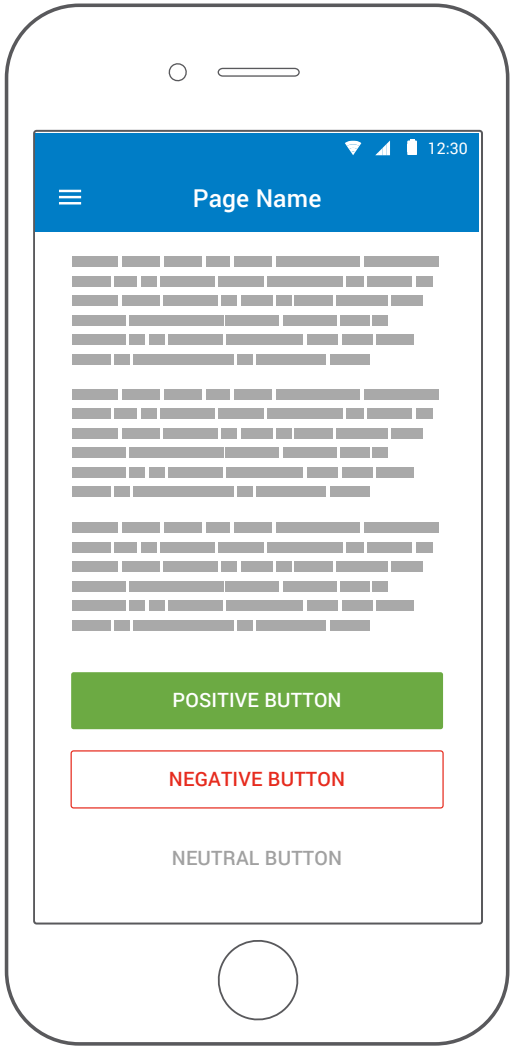


Fig. 1 Examples of button types and styles

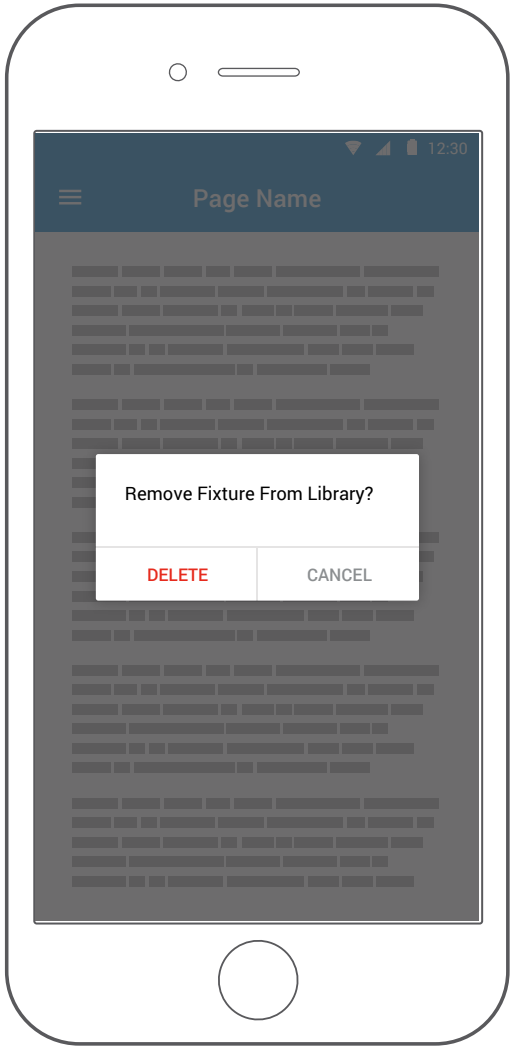


Fig. 2 Example of a dialog box

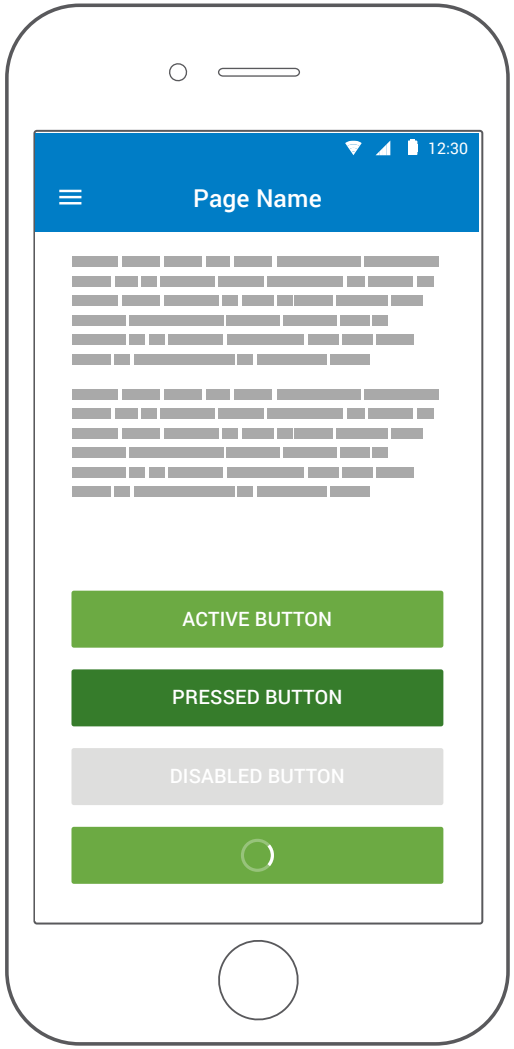


Fig. 3 Examples of different button states

Buttons

Button Guidelines

After determining the function and style of your buttons it is important to ensure that the button meets space requirements. They need to be large for easy tap ability and clear to read. DO NOT overlap text, padding or touch areas. Round edges when presented as solids or with outlines by a 2 px radius. Always present button text in all caps. In cases of two or more buttons they can be stacked vertically or horizontally. In most cases, such as a modal window, they will most likely need to be vertically organized. Again, dialog buttons do not require a background color, only the appropriate coloring and weight.

Icon-only buttons are only for limited cases such as search in navigation.

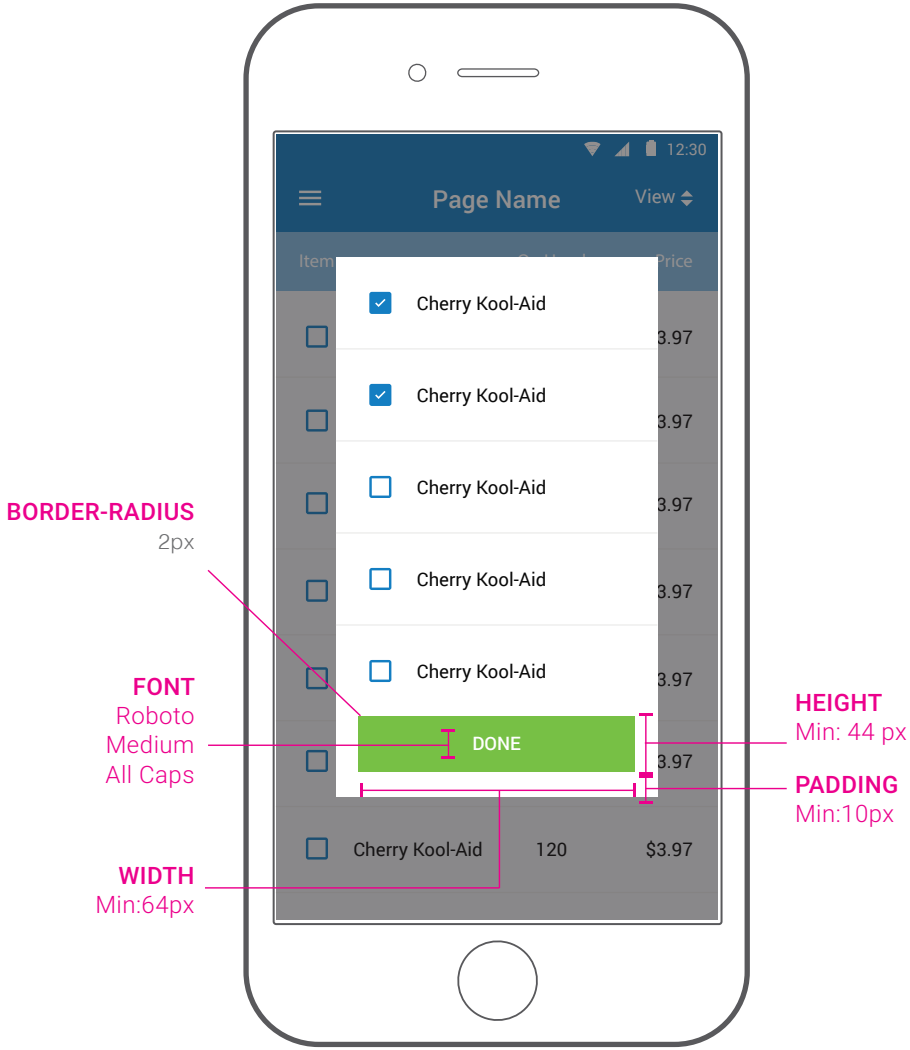


Fig. 1 Button Breakdown

These graphics are not true to px size.

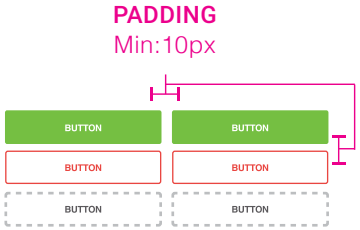


Fig. 2 All buttons, stacked vertically
or horizontally must have 10 px padding

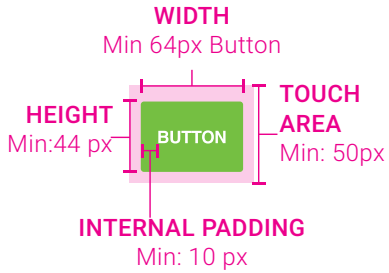


Fig. 3 The minimal acceptable button dimensions

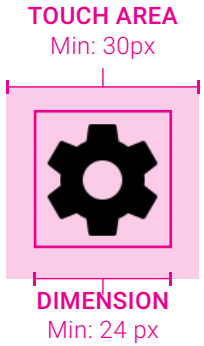


Fig. 4 Icons can be smaller than 30px but must have at
least a 30px touch area to meet the minimum.

Progress Indicator

Assurance that Everything is Working

Tasks that are running in the background should be represented on the interface with a progress indicator. By giving the user acknowledgement that the app is working on their request, the user feels confident that everything is working properly, and they will be able to complete their work.

A progress indicator has a graphic visualizing the percentage of work completed and a short message indicating what action is taking place (Fig. 1).

If the amount of work completed cannot be measured, use a spinning “loading” icon instead (Fig. 2). For tasks that are processing-intensive, an optional “cancel” action can be added if the user wishes to back out of the action.

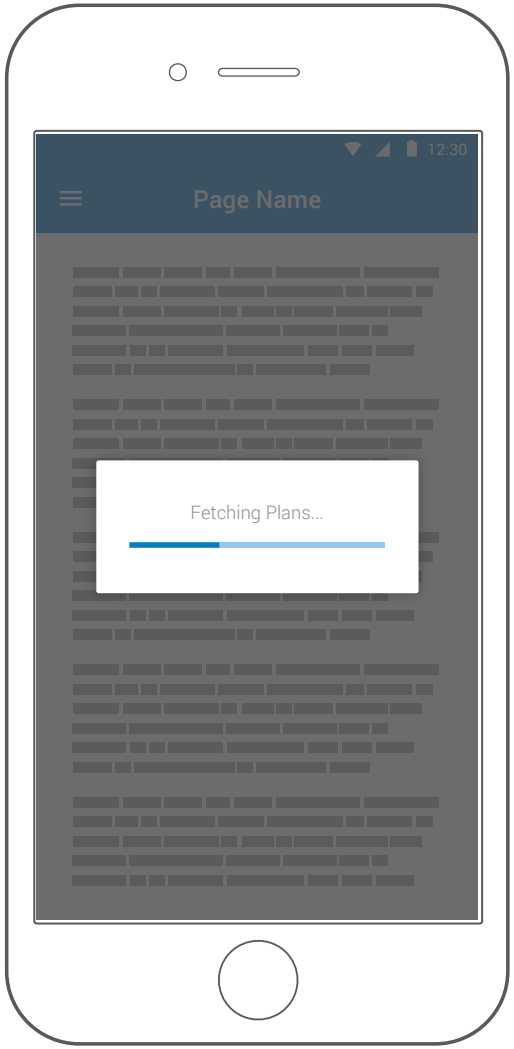


Fig. 1 Finite loading

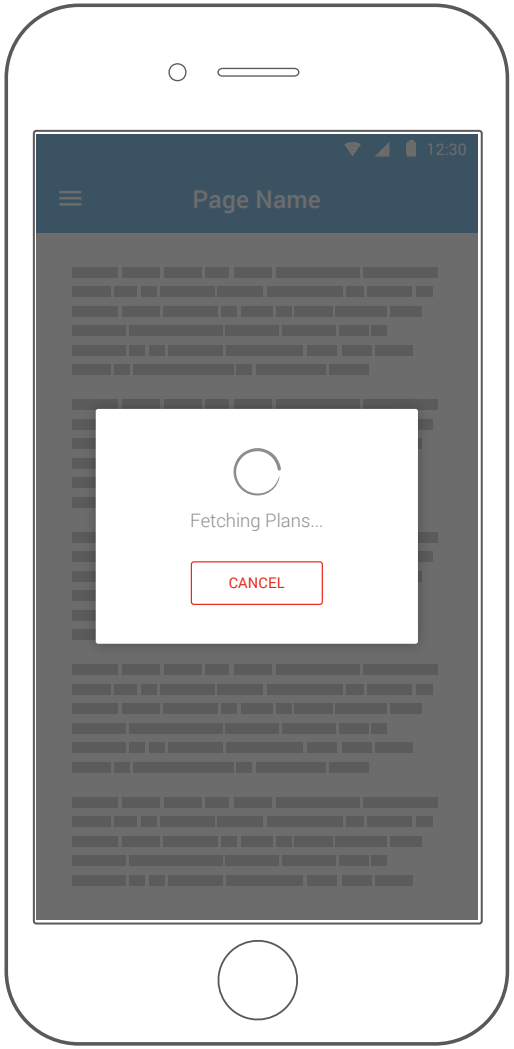


Fig. 2 Infinite loader

Progress Indicator

Best Practice

DO

✓ Include time-out functions that will exit out of the progress indicator after it exceeds a set amount of time.

DO NOT

✖ Show the percentage of work left or completed.

✖ Be overly technical in the description.

✖ Let the load bar decrease in value.

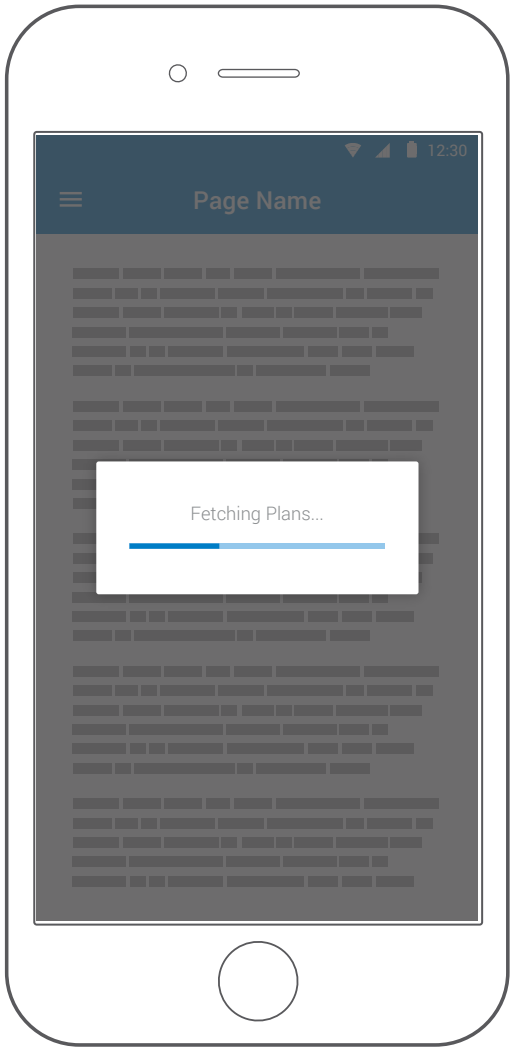


Fig. 1 Embed progress indicator into the content it is loading.

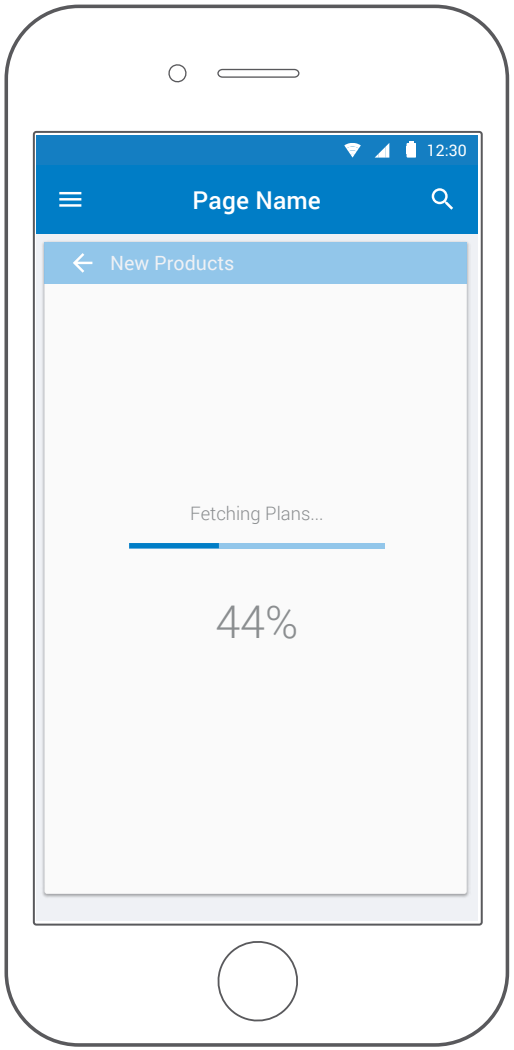


Fig. 2 Do not show a percentage amount of work remaining or completed

Progress Indicator

Variation

When possible, the interface surrounding the loading content should still be in view **(Fig. 3)**. This allows the user to continue viewing other content while other tasks are working in the background.

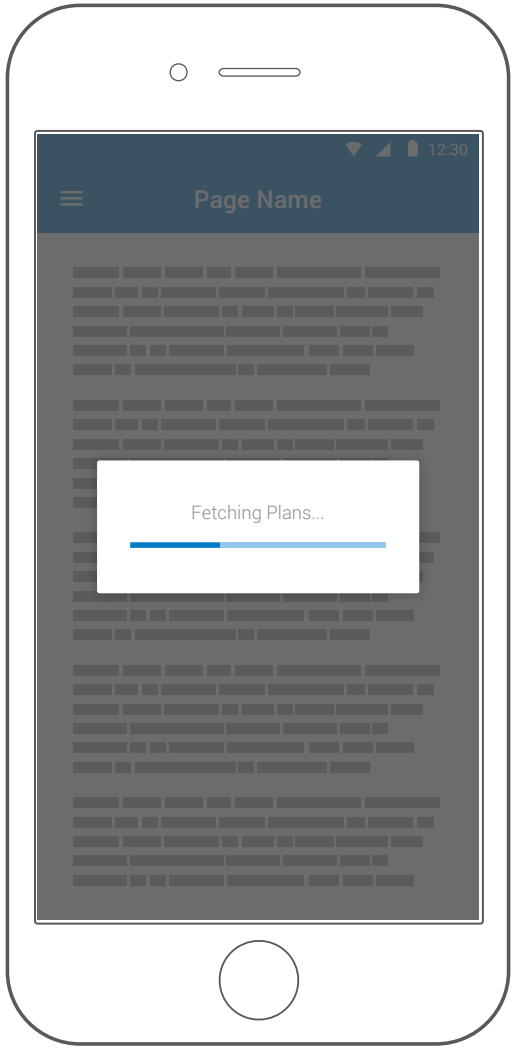


Fig. 3 Progress indicator inside content

Radio Buttons

Selecting a Single Item from Multiple Options

Radio buttons are a round graphical element used when there is a list of two or more mutually exclusive items. A radio button only allows ONE choice among a mutually exclusive selection. Do not confuse this with a checkbox, which allows any number of options to be selected among an inclusive selection.

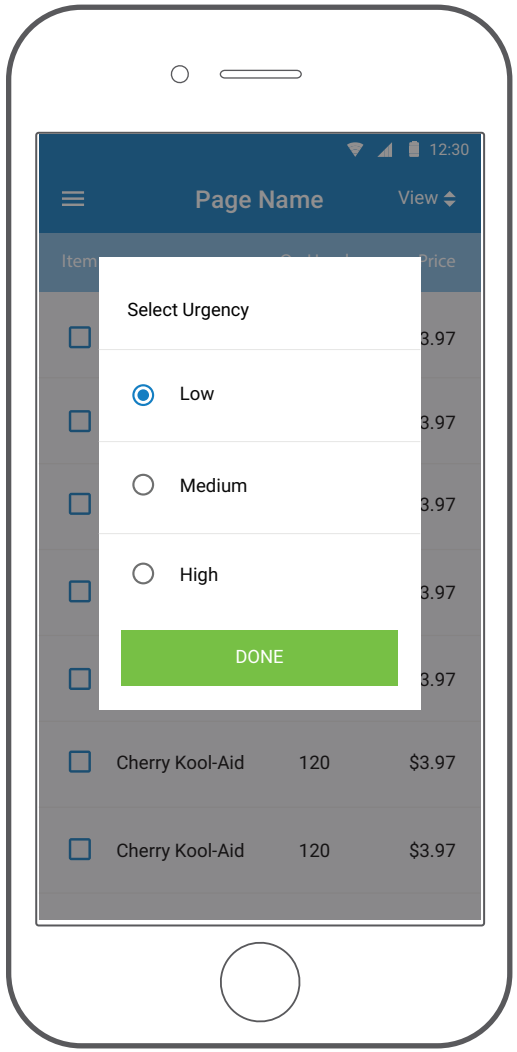


Fig. 1 Best example

Radio Buttons

Best Practice

DO

- ✓ Always use labels to inform the user the active state.
- ✓ Always use short text, such as “On” and “Off.”
- ✓ Always have active state highlighted and titled.

DO NOT

- ✖ Just rely on toggle color to indicate status.

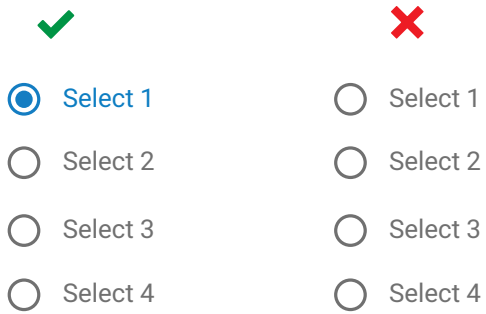


Fig. 1 Always autofill a radio with the common answer to hint the user

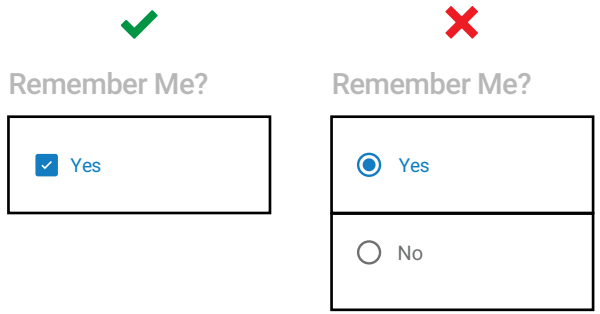


Fig. 2 Use a checkbox for two mutually exclusive items

Radio Buttons

Variation

The radio button is named after the old radios that used buttons to select preset stations and only allowed one button to be depressed and set at a time.

STATES

Selected - Displays a filled in circle
Normal - Shows non selected state
Disabled - Greyed out to show no interaction is enabled

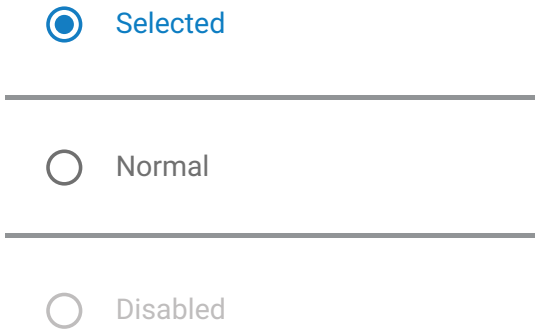


Fig. 1 Different states

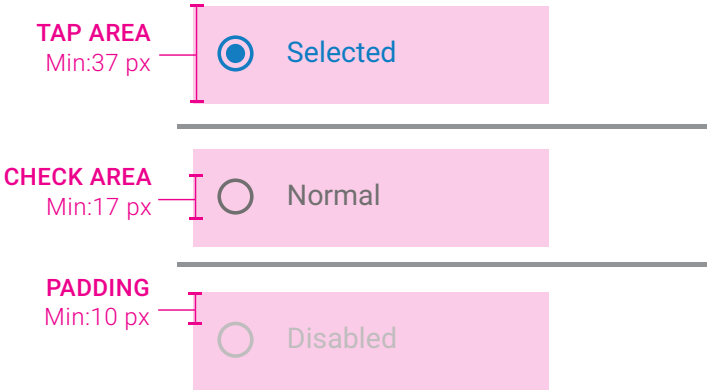


Fig. 2 Padding and tap area

Checkboxes

Selecting Multiple Options from a List

Checkboxes are a square, graphical element that allows users to select one or more options in response to a question. Checkboxes are often confused with radio buttons in cases of bad UX. However, while a radio button only allows ONE choice from a mutually exclusive selection, checkboxes allow any number of options to be selected among an inclusive selection. Checkboxes may also be used for a single on and off option such as “Remember Me” logins; however, do not use them as action settings – that is what a toggle button is for.

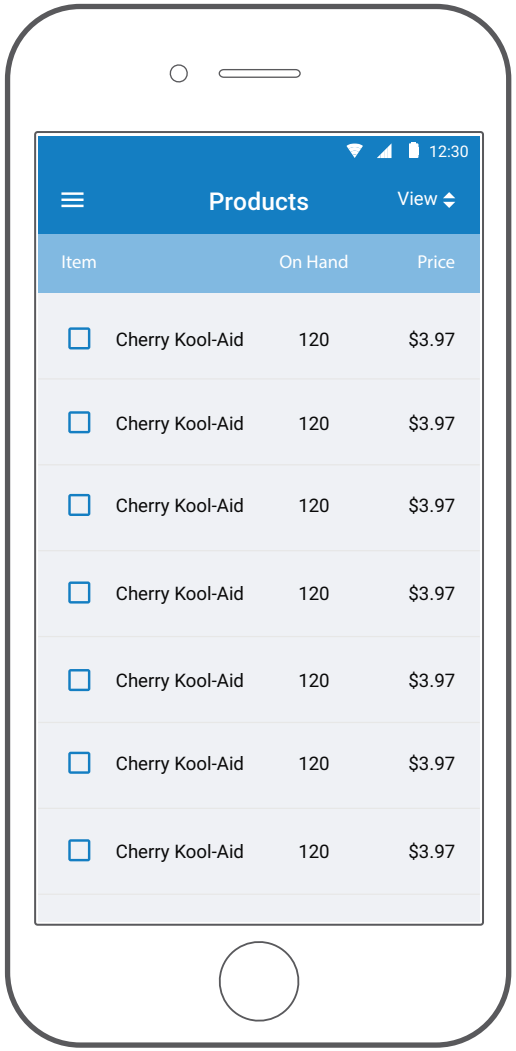


Fig. 1 Checkboxes in a table

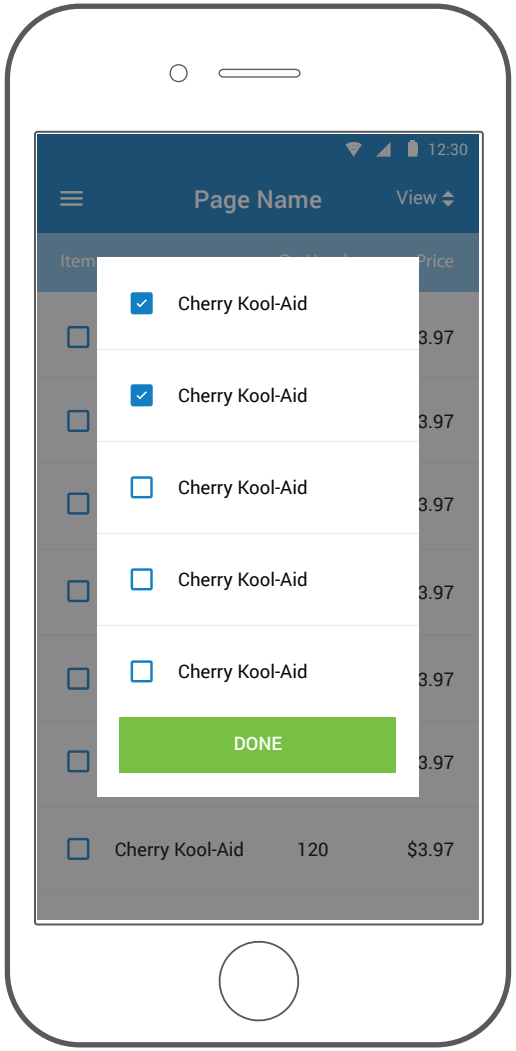


Fig. 2 Checkboxes in a dialog box

Check Boxes

Best Practice

DO

- ✓ Align checkboxes vertically if more than three.
- ✓ Visually show choices as groups, using subheads to break up different groups.
- ✓ Limit text content to two lines.
- ✓ Word text content as a true statement, so the absence of a check mark makes it false, for clarity.
- ✓ Always use labels.

DO NOT

- ✖ Use the checkbox to toggle options on or off. Use a toggle.
- ✖ Use the indeterminate state to represent a third state. This is used to indicate that an option is set for some, but not all, sub-choices. Use a drop menu or radio buttons instead.
- ✖ Use more than 10 checkboxes in a group.
- ✖ Use checkboxes to perform commands.

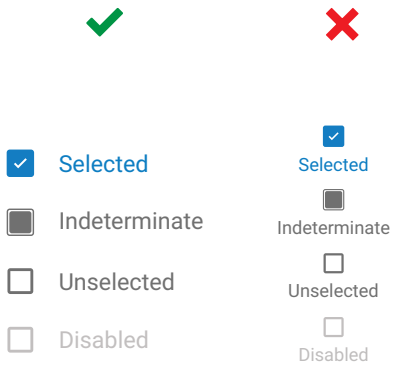


Fig. 1 Place labels to the right for readability

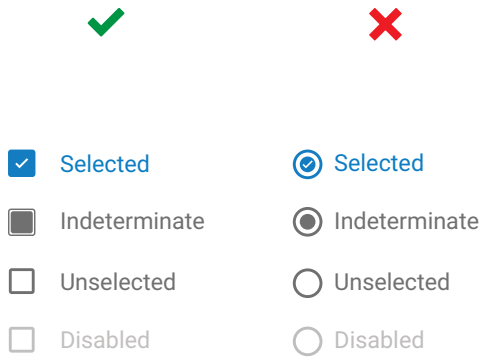


Fig. 2 Use standard checkbox format

Check Boxes

Variations

STATES

Check boxes exist in three states, unselected, intermediate, selected and disabled.

Make sure to always include a title to the left of a box, and make it tappable as well. Allow for sufficient padding around the box and include a backdrop to further pad the tappable area (Fig.2).

- The indeterminate state is for when there are a group of sub-choices that have selected and unselected states.
- Use radios when the choice is an “or.” Use checkboxes when choice is an “and.”

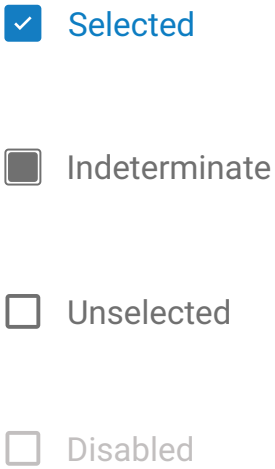


Fig. 1 Different states

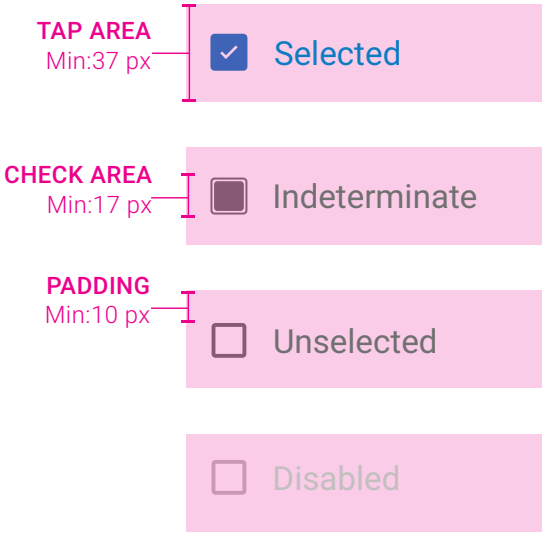


Fig. 2 Padding and tap area

Toggle Switch

Clear Indication Between Two States

Toggle switches mimic physical switches that allow users to turn things “On” or “Off.” Toggle switches can be very efficient in terms of space and control. This is for binary actions that are effective immediately. There may be times when the choice between a checkbox and a toggle switch is confusing. If multiple items can be selected, or multiple steps need to be taken, then use a checkbox. Switches are usually seen in places such as settings.

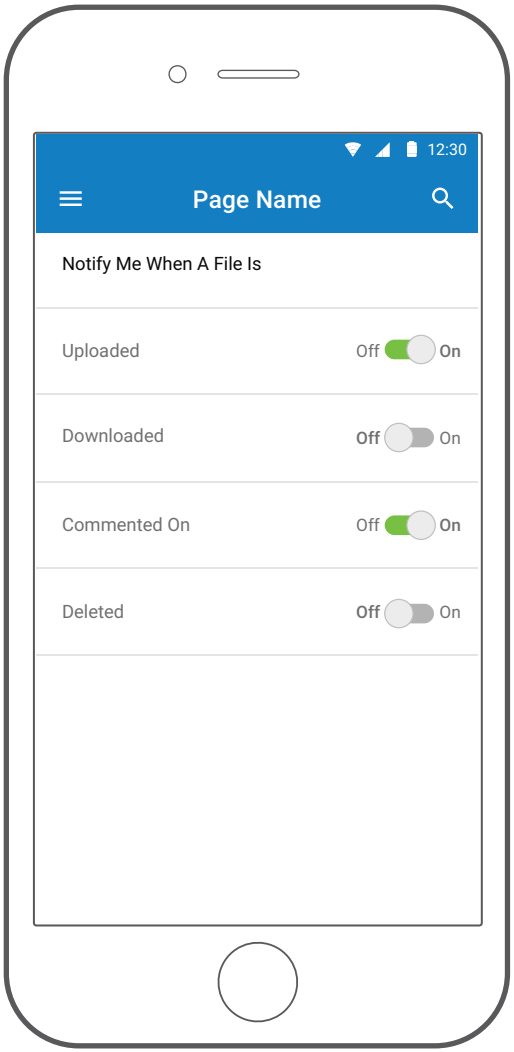


Fig. 1 Best example

Toggle Switch

Best Practice

DO

- ✓ Always use labels to inform the user of the active state.
- ✓ Always use short text, such as “On” and “Off.”
- ✓ Always have active state highlighted and titled.

DO NOT

- ✖ Just rely on toggle color to indicate status.

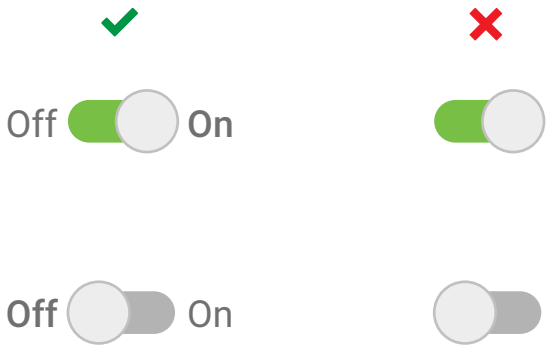


Fig. 1 Label toggle switches

Toggle Switch

Variation

STATES

Selected - Shows a true selection

Indeterminate - Used to show selection of a group

Unselected - Shows a false selection

Disabled - User is unable to select, may be influenced by other action

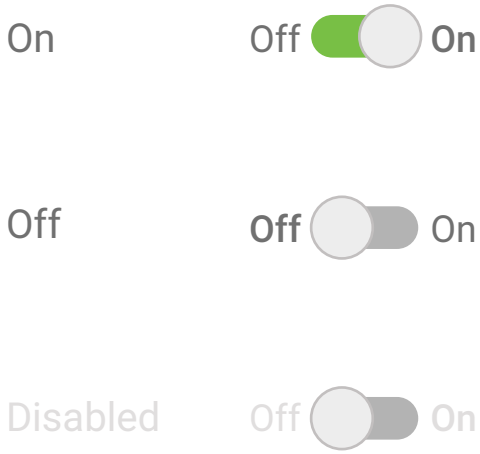


Fig. 1 Different states

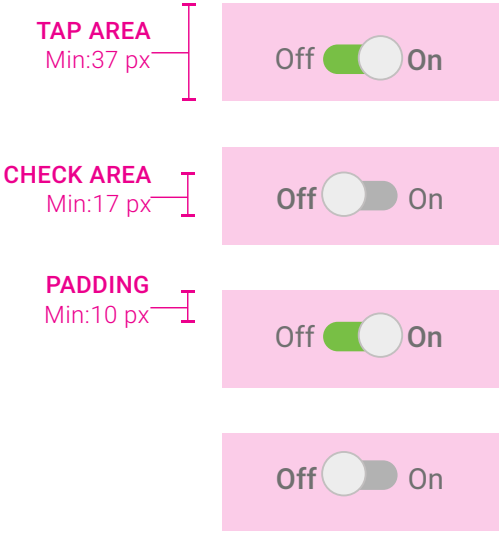


Fig. 2 Padding and tap area

Validation and Errors

Form Feedback in Real-time

Feedback that lets the user know when they have successfully entered information into a field, and where they need to make corrections, helps the user input information into the system and complete tasks.

A combination of color, iconography and messaging directs the user's attention and provides help with correcting the error (Fig. 1).

Ideally validation can happen in real-time; after a user makes an input, a background task ensures that what the user just entered is valid. If the input is valid, providing visual feedback is helpful for the user (Fig. 2).

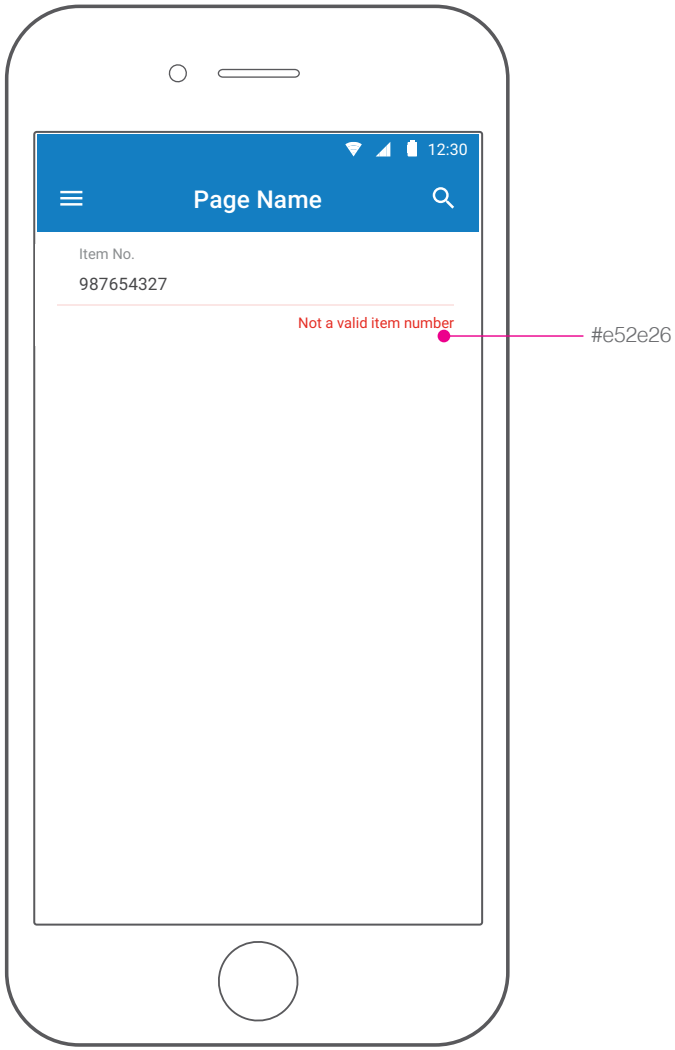


Fig. 1 Error state example

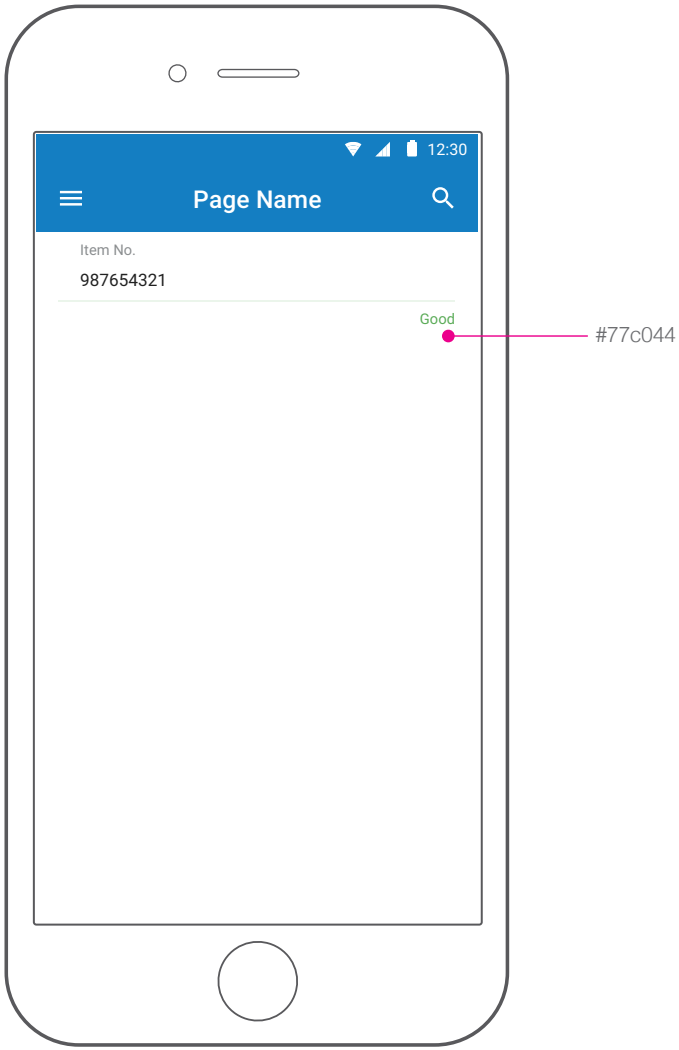


Fig. 2 Validation example

Validation and Errors

Best Practice

DO

✓ Use clear, direct messaging when writing error states.

DO NOT

✖ Rely on just color and iconography to show error states. Users need messaging.

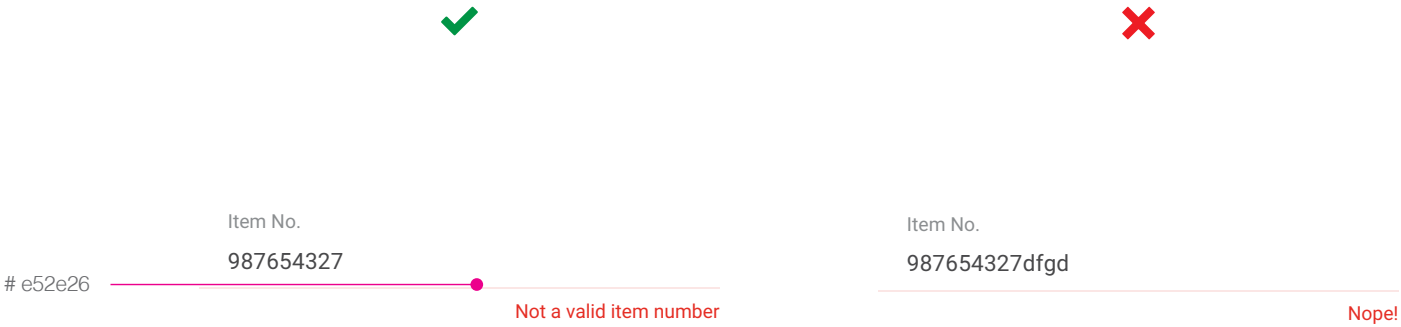


Fig. 1 Be descriptive in error messaging

Validation and Errors

Variation

If asynchronous validation is not possible, then error states will occur after the user makes a submission action. A top-level notification alerts the user that their submission failed, and to correct their errors and retry (Fig. 1).

Conversely, a successful form submission should have a confirmation message (Fig. 2).

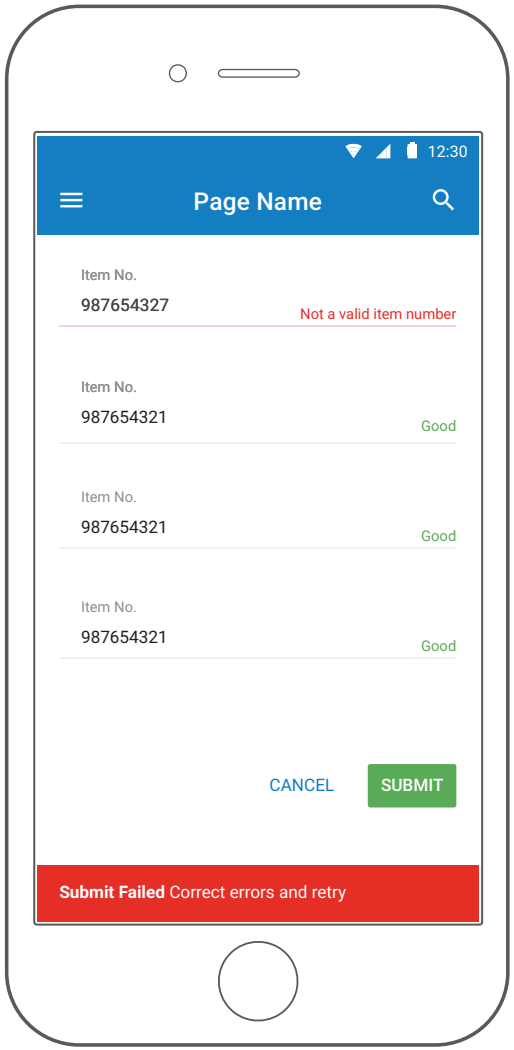


Fig. 1 Error mesage

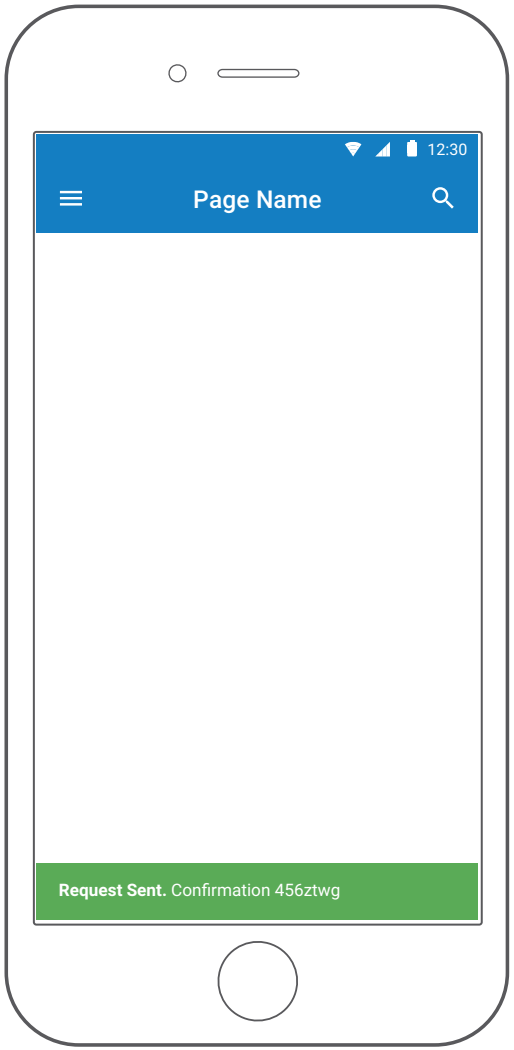


Fig. 2 Success message

Search

Find Exactly What You are Looking for

Search is a powerful and versatile way for users to find the content they need to perform their job.

Selecting the search button animates the search input in the header, while simultaneously bringing up the on-page keyboard **(Fig. 2)**. If the user taps the “back” button, they will return back to the normal header. The user can also select from recent search queries (optional).

After the user has entered their query, they are taken to the search results page **(Fig. 3)**. An optional tab bar can help the user refine their results.

The search bar can also include an option to search by barcode scan. Tapping the barcode icon in the search bar takes the user to the barcode scanner process, then returns them back to search results after a successful scan.

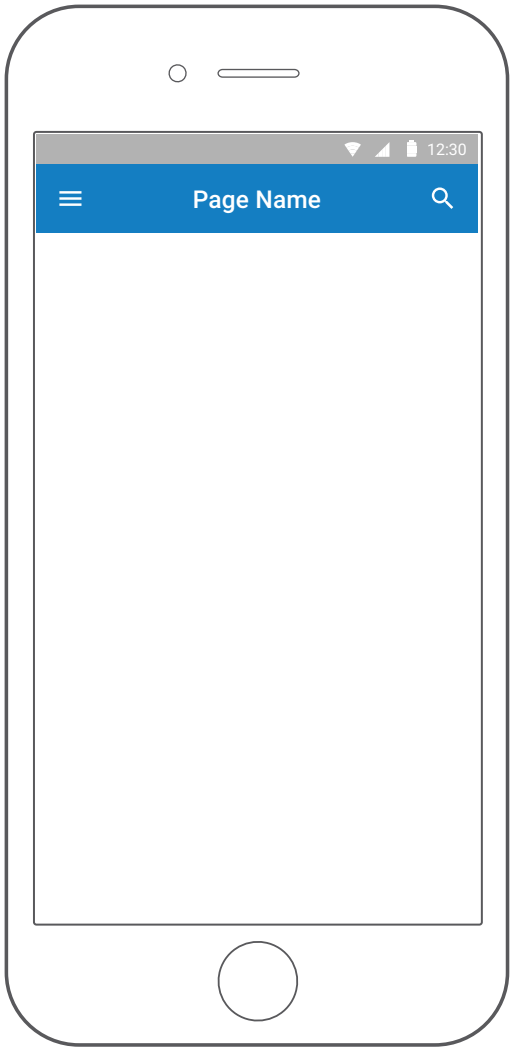


Fig. 1 Header with search icon

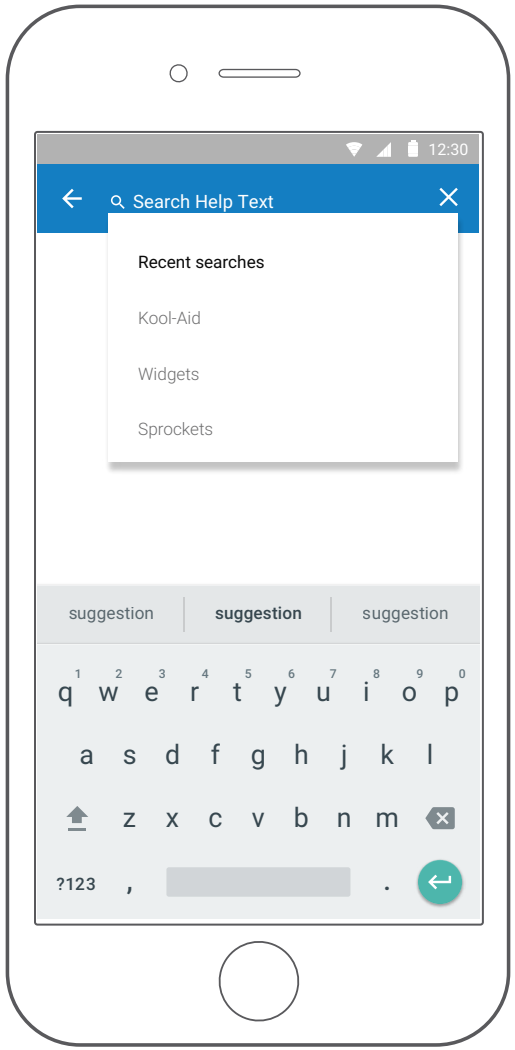


Fig. 2 On select

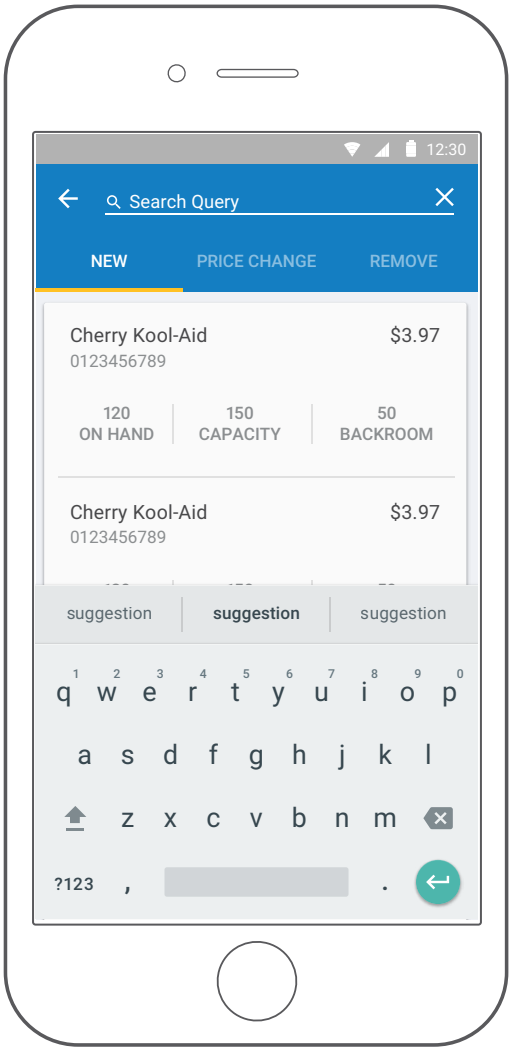


Fig. 3 Results

Search

Best Practice

DO

✓ Provide helpful hints on what keywords the user can use for their query.

DO NOT

✖ Put too much information into the results page. Only include enough metadata to help the user reach their destination.

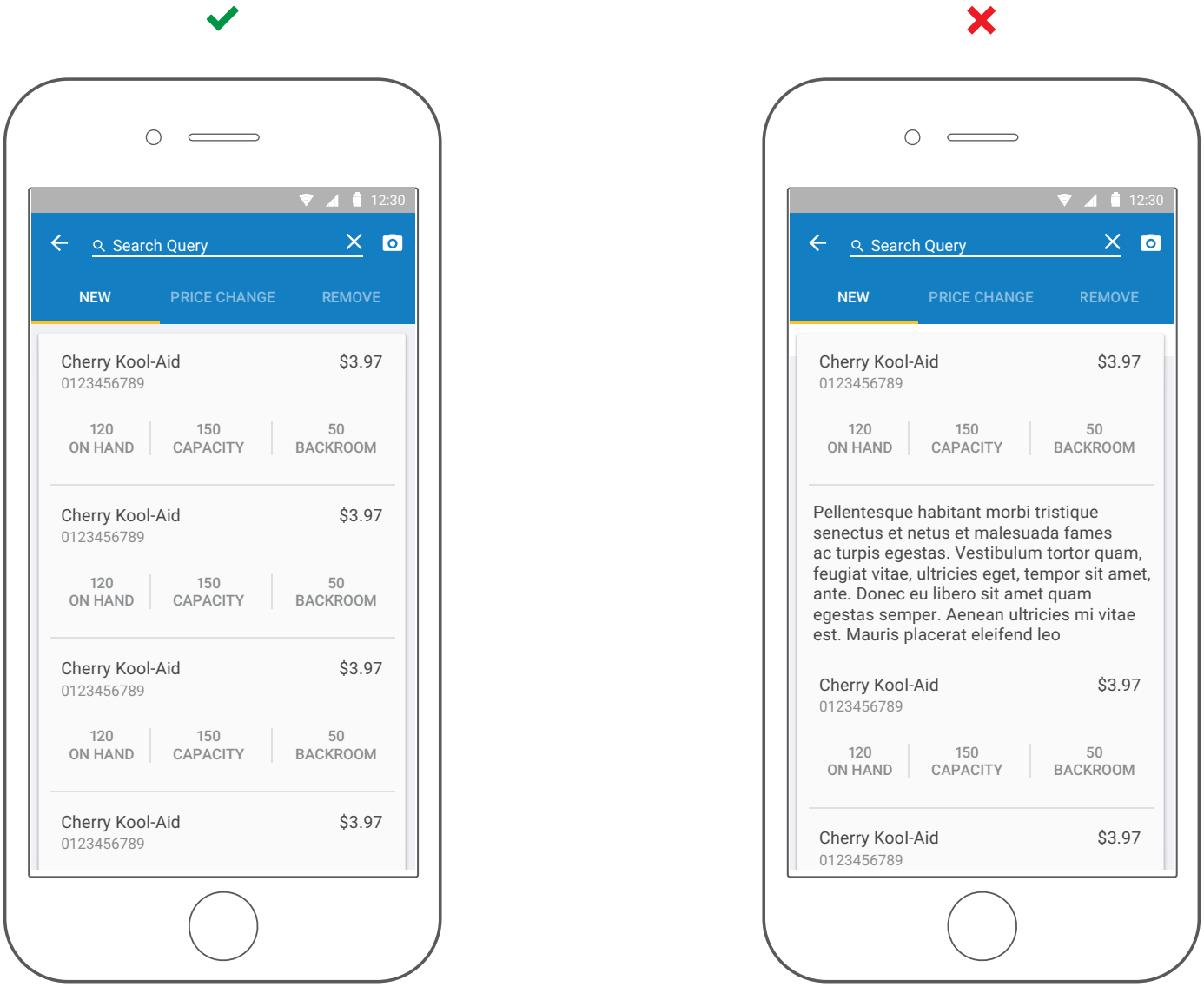


Fig. 1 Search results should have just enough information to get the user to where they need to go

Search

Variations

If the primary action of your app is to find a specific piece of information out of a large set of data, a search nav-bar might be a good option.

The search bar is placed inside the header, with an optional tab-bar below to further refine results.

The search nav-bar could be a good solution if your app serves a single function (Example: store locator, item lookup, etc.).

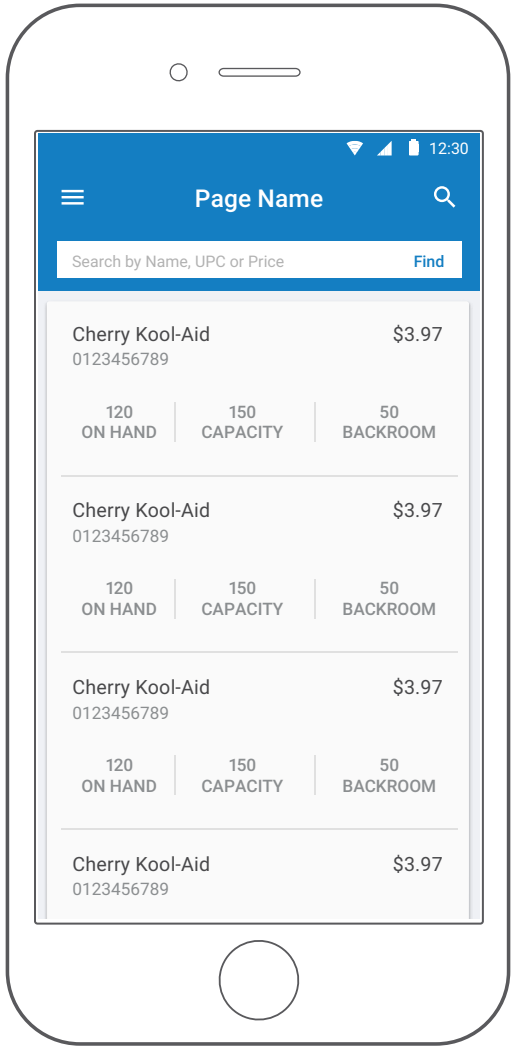


Fig. 1 Search bar

Export

Print, Share, and Download Data

Information needs to be portable. Everyday within Walmart, associates take portions of data from multiple sources and combine them to gain new insights. All data in an app should be easily extractable, shareable and readable for other applications.

Pages that include any sort of information (Example: tables, lists, data visualization) should have the “send to” option within the context menu (Fig. 1). When a user taps “send to,” the user will have several options to choose from:

Print: Creates a printable version of the current page and opens the system print dialog.

Share: Opens a dialog to allow the user to send page via email (or the wire).

Export: Converts the information on page into a common format (.csv, .doc, .pdf).

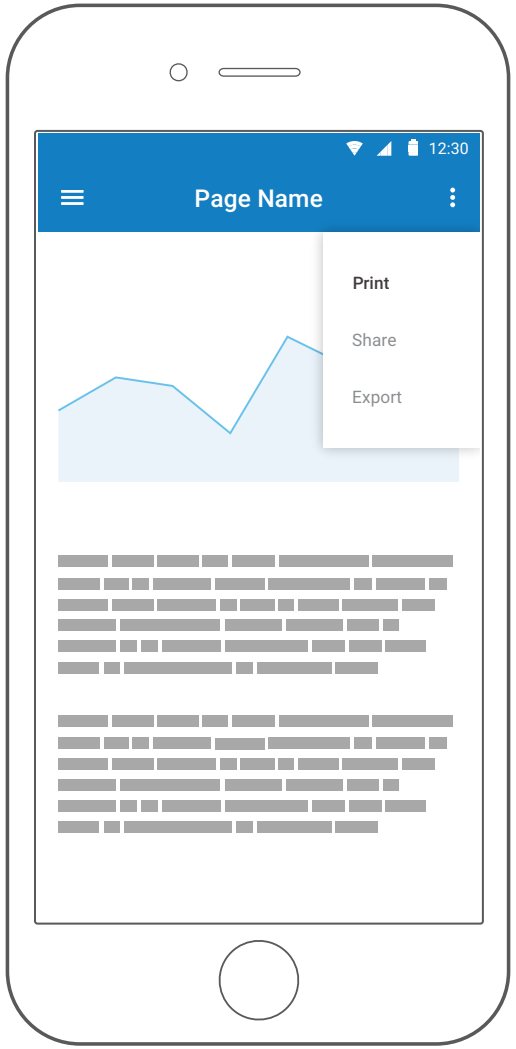


Fig. 1 Mobile example

Export

Best Practice

DO

- ✓ Create a separate print style sheet that prints only the relevant data.
- ✓ Label exports with a filename that explains the data type, the app and the date it was created.
- ✓ Get permission from project stakeholders that data within the app can be shared and exported.

DO NOT

- ✖ Allow the user to print, share or export files that are beyond reasonable limits for mobile devices.

Fig. 1 Create a sperate stylesheet for print pages that optimizes the layout for a single page

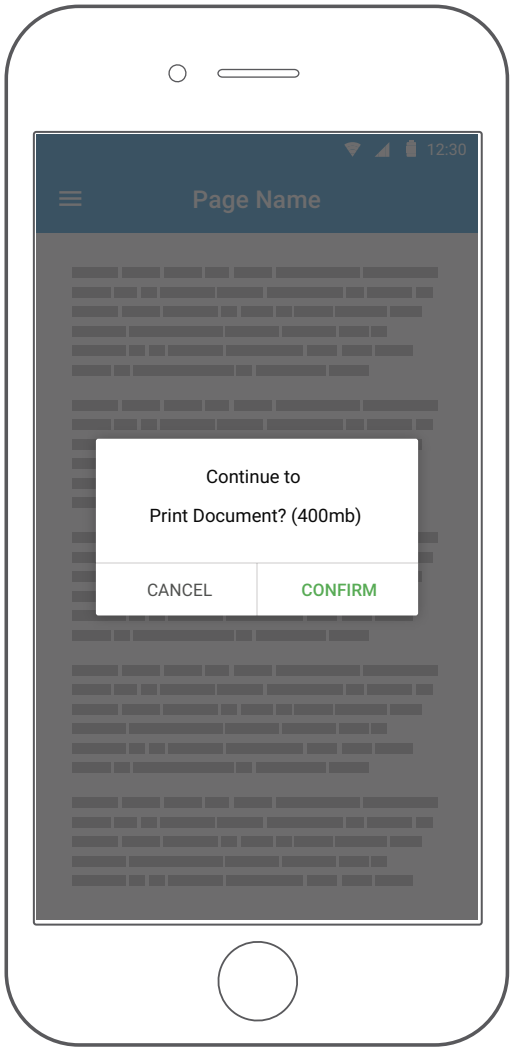


Fig. 2 Don't allow the user to print, share, or download a file that is too large.

Export

Variations

If printing, sharing or exporting data is a common action the user will take, move the action into the page as a series of buttons (Fig. 1).

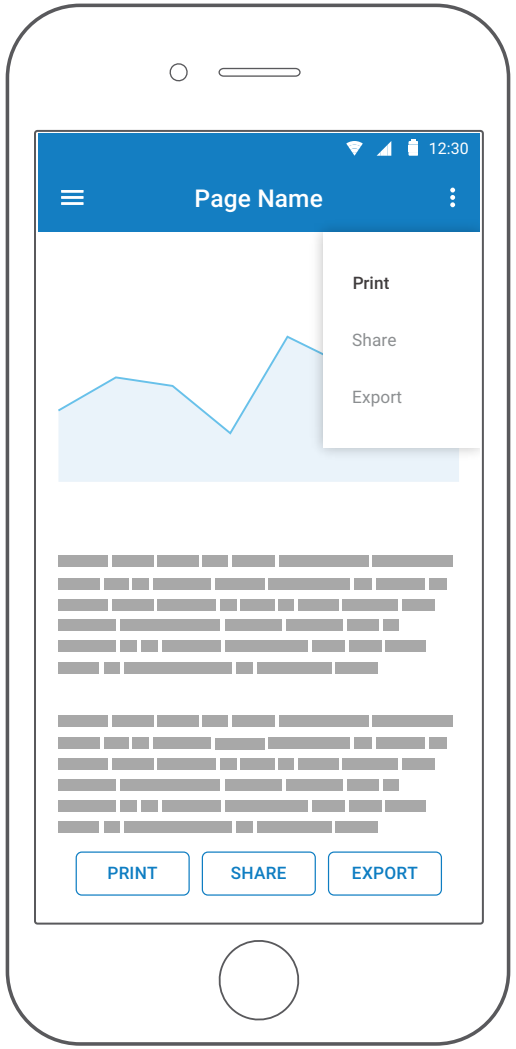


Fig. 1 Export options as in-page buttons

Content

The Primary Vehicle to Deliver Information

Content makes up the bulk of our apps. It gives the user the crucial information they need to do their job efficiently and correctly. Content can be shaped into many forms, including tables for organized data, cards for quick summaries and accordions for structured text blocks. As app builders, it is our job to shape content into the most valuable format for our associates.

It is important that our content be as clear and straightforward as possible. All of our associates work in demanding positions that require their attention in multiple areas at once. By ensuring our content is actionable, we allow our associates to make the right decisions.

5.0 Content

Accordions

Sectioned Content

Content can become unwieldy on small page devices. The limited width causes content to condense down, forcing the user to scroll up and down to find the information relevant to their needs. Accordions solve this problem by wrapping content into different “blocks” that expand and collapse when a user taps them.

Each accordion block includes a title that summarizes the content that it contains, and a directional icon that lets the user know that the element will expand when selected.

If the accordion content is enumerated content, such as a list, table or alerts, an optional badge can be added to provide additional context to the status of the content inside.

When a user taps one accordion, any accordions that are open should close automatically.

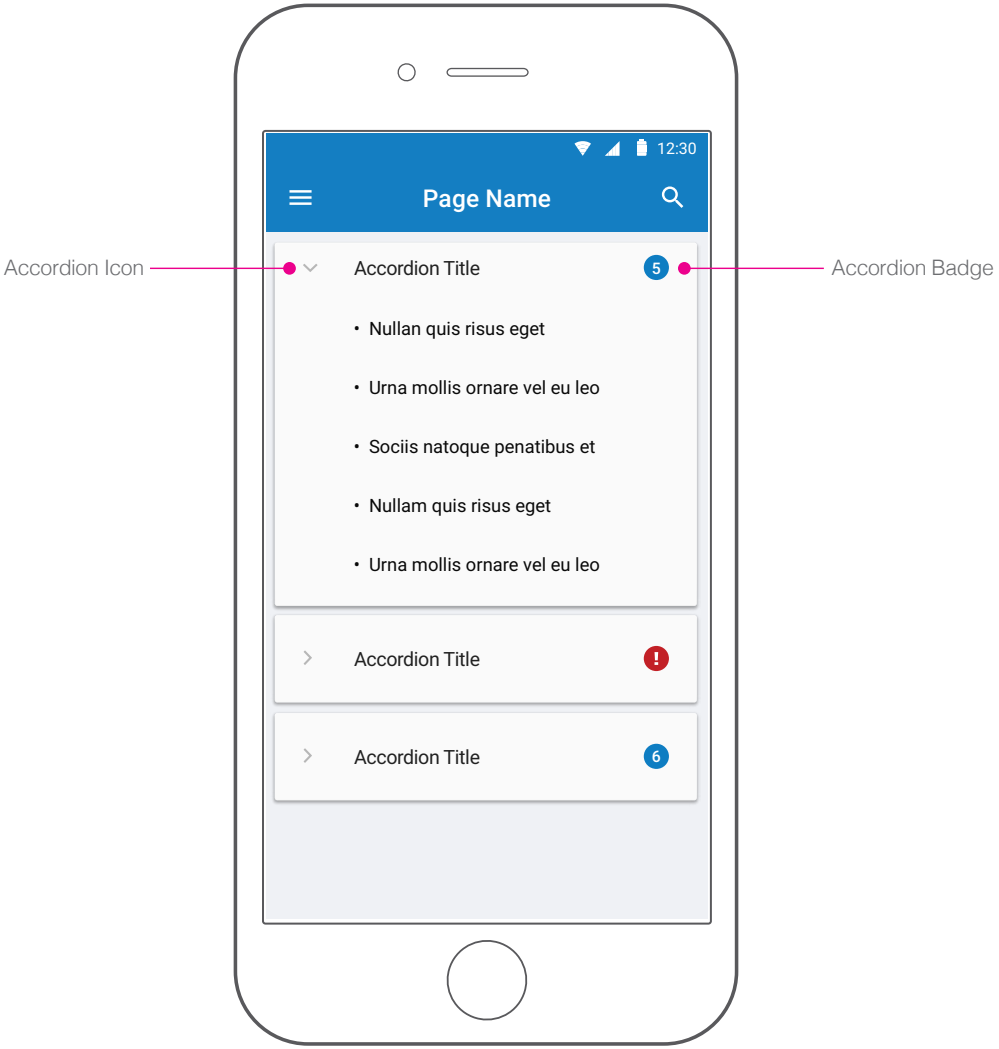


Fig. 1 Mobile example

Accordions

Best Practice

DO

✓ Use accordions to section content off and make finding it easier.

DO NOT

✖ Nest accordions inside other accordions.

✖ Put too much information into an accordion.

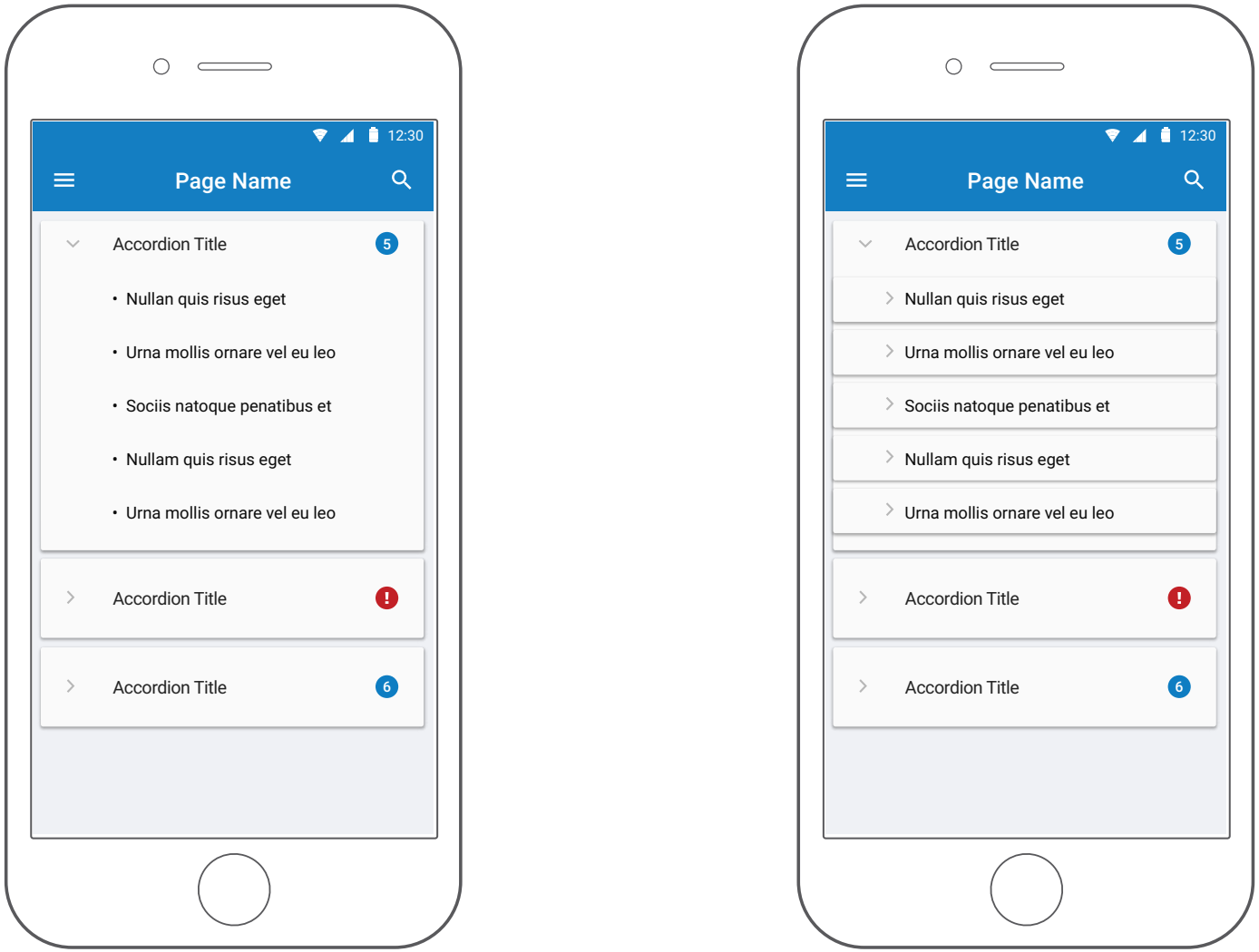


Fig. 1 Do not nest accordions within other accordions

Tables

Organized Content

Data that is organized into a table format is easy for the user to read and sort.

It is extremely important that table data be streamlined to the bare essentials when designing for mobile. If the data can be displayed as a card or visualization, use that instead.

Selecting the header of a number column reorganizes the data with the highest number in the first row position, with the following data appearing in descending order (Fig. 1). Selecting the column again inverts the data to ascending order. Columns with non-numerical data will instead sort alphabetically.

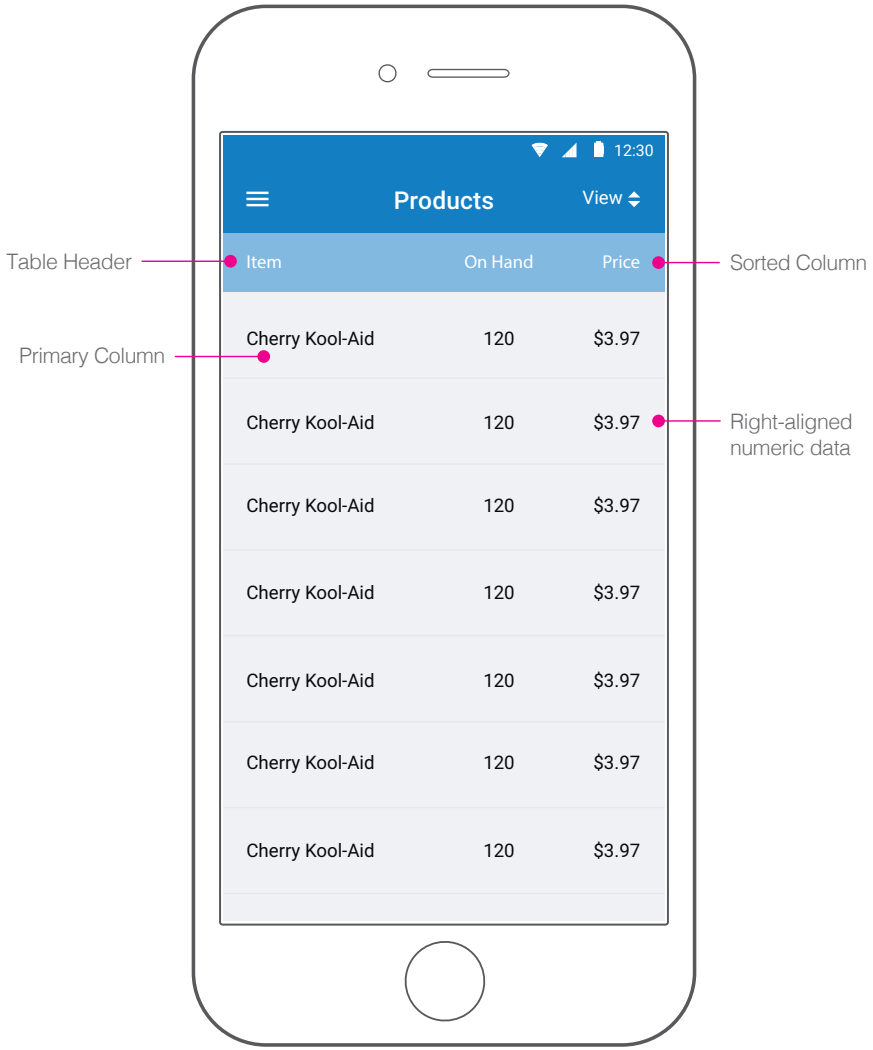


Fig. 1 Layout with multiple cards

Tables

Best Practice

DO

- ✓ Use colors in tables sparingly.
- ✓ Consider if the data can instead be displayed as a card or data visualization.

DO NOT

- ✖ Cramp too many table cells into the view. The more white space and padding an element has, the cleaner it will look.

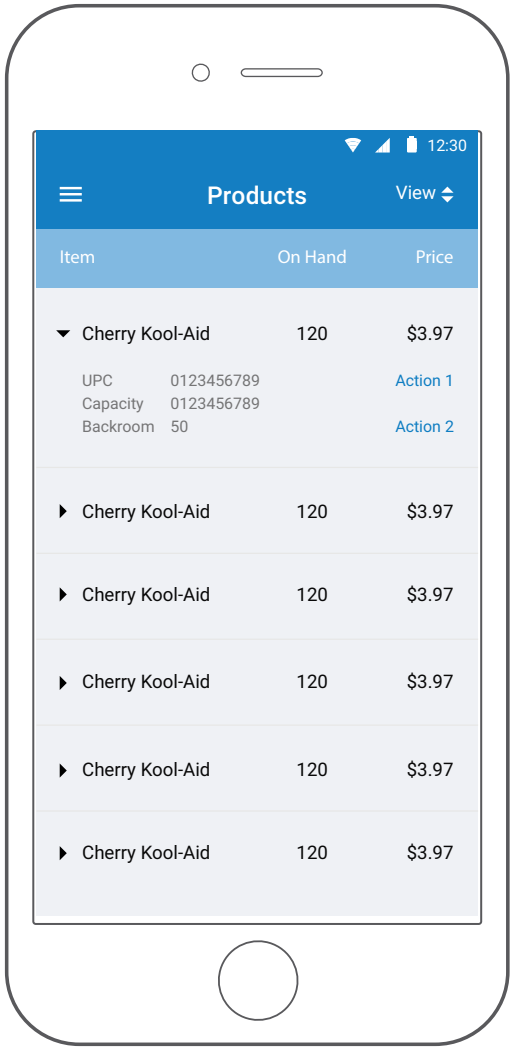


Fig. 1 Layout with multiple cards

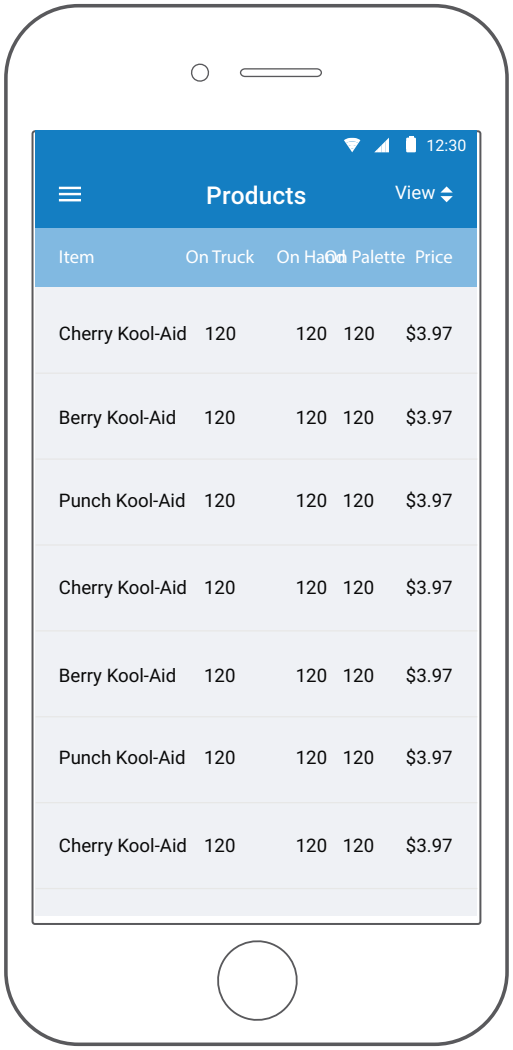


Fig. 2 Layout of cards as a list

Tables

Variations

A table row can have a checkbox that will allow the user to perform an action across multiple rows at once (Fig. 1). Alternatively, a column can be created exclusively for a singular action (Fig. 2). Only use this option when the user is expected to perform a single action during the scenario.

Visual elements, such as stats, badges and form fields can be inserted into tables as well (Fig. 3).

Data that is hierarchal (or is considered supplemental) can be hidden behind an expandable accordion (Fig. 4). When the user taps a row, it expands to reveal additional content and/or actions. Tapping the row again, or another row, collapses the row back to its default state.

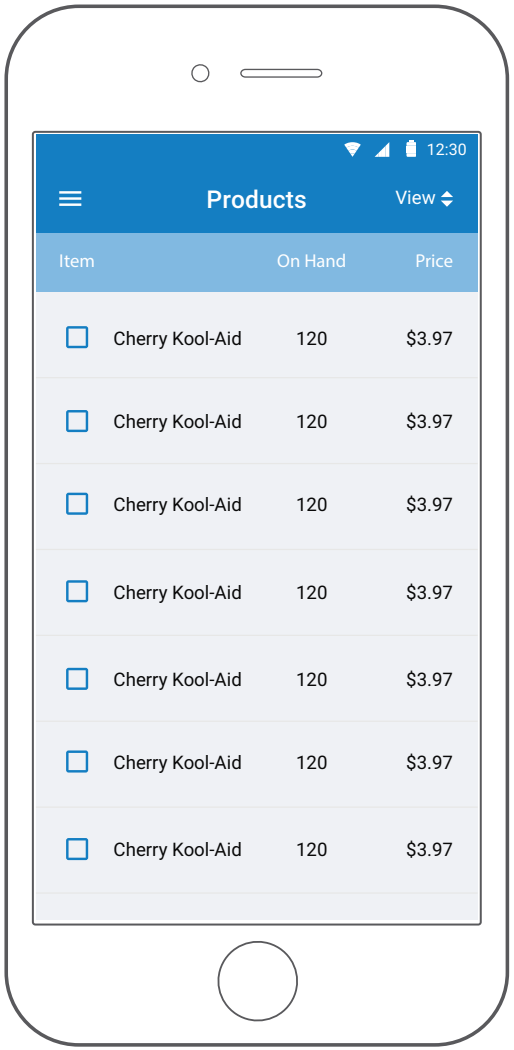


FIG. 1 Table with Checkboxes

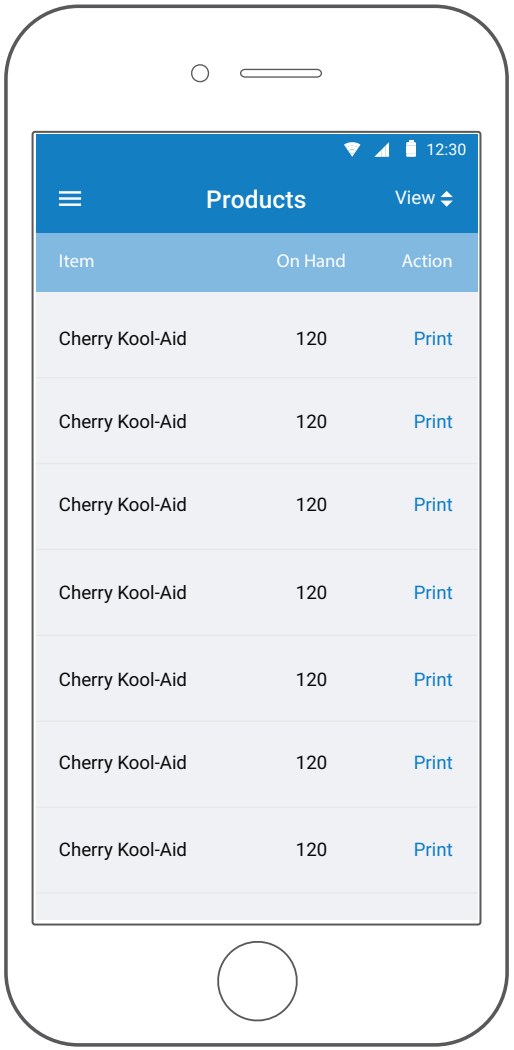


Fig. 2 Table with actions

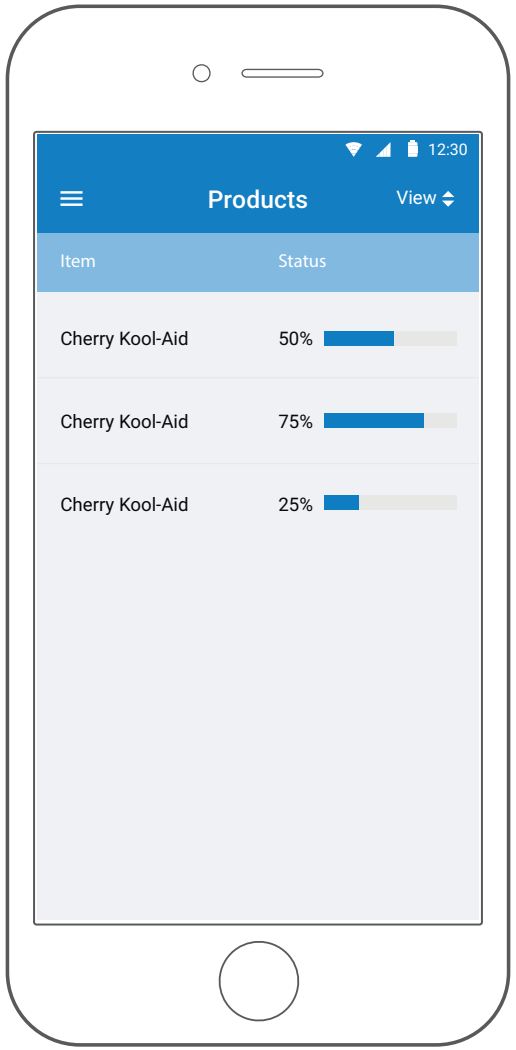


Fig. 3 Table with data visualization

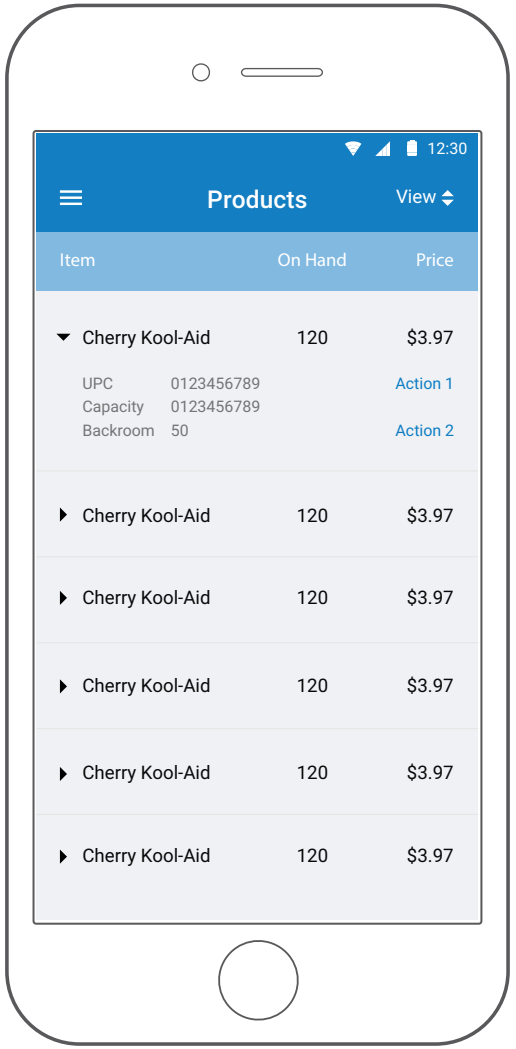


Fig. 4 Tables as accordions

Tables

Responsive with Visibility Filter (Phone)

For tables with many columns of data, a filter that can toggle visibility of a specific column on and off allows maximum flexibility.

EXAMPLE

Media queries determine the number of columns that can fit into view (Fig. 1). The remaining columns are automatically set to “unchecked” in the view menu (Fig. 2).

By selecting the “view” button, the user has the ability to re-configure which columns are rendered visible (Fig. 2). A scroll bar will appear to allow the user to pan across columns if the user selects more columns than the viewport can accommodate (Fig. 3). The leftmost column is regarded as a “primary” column that always stays locked to the viewport and can never be set to hidden by the user.

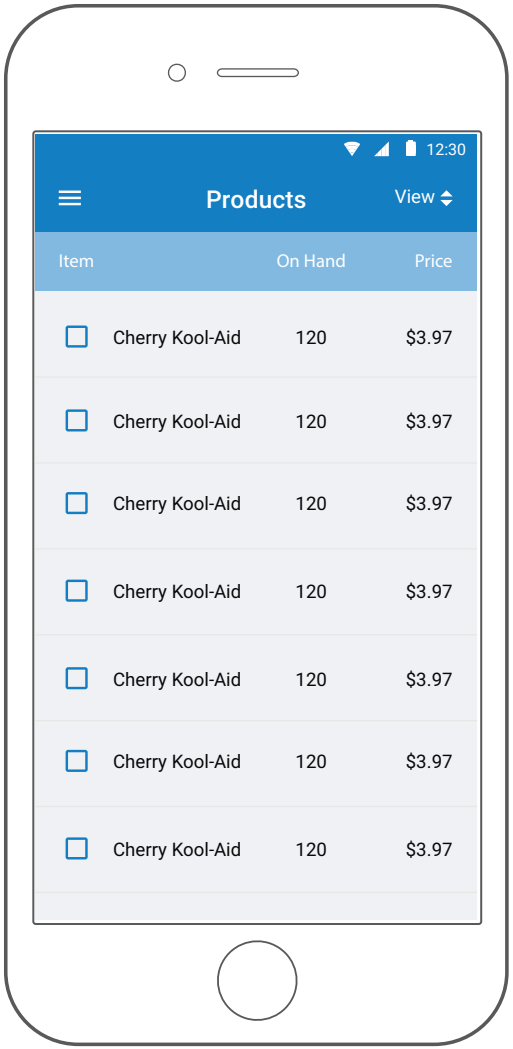


Fig. 1 Default table view

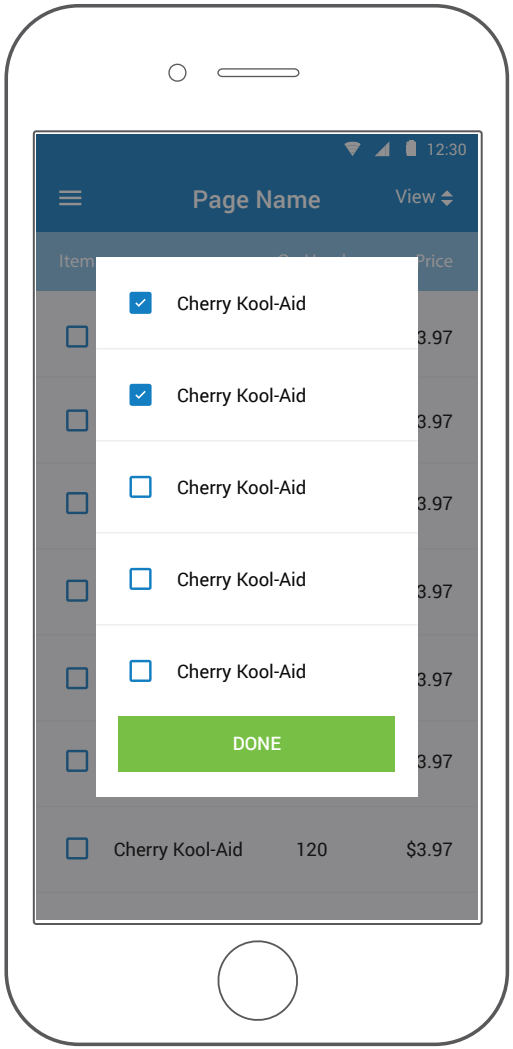


Fig. 2 View box

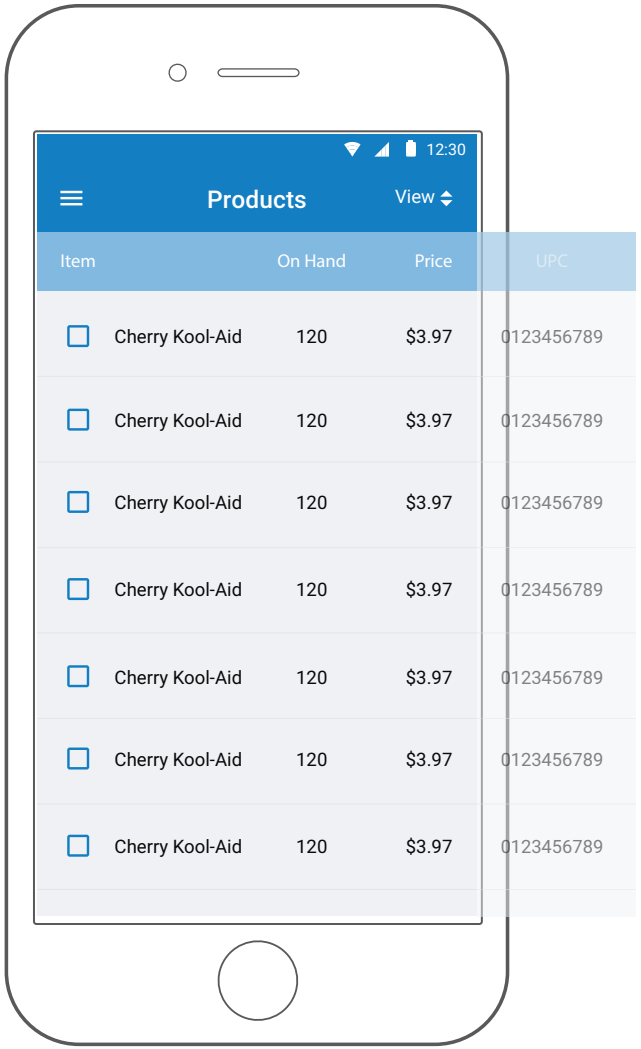


Fig. 3 Table with added columns

Tables

Responsive with Visibility Filter (Tablet)

Devices with a larger screen size have the ability to Devices with a larger page size have the ability to accommodate more columns in view **(Fig. 1)**. Tapping the “view” button triggers a popover menu that shows which columns are set to visible **(Fig. 2)**. When a user checks or unchecks a box, the column action happens immediately.

The behavior for having more columns than the page width can accommodate is the same as for mobile: Columns overflow the page left to right, with the primary column always in a “locked” position.

Related links: http://gergeo.se/RWD-Table-Patterns/#demo

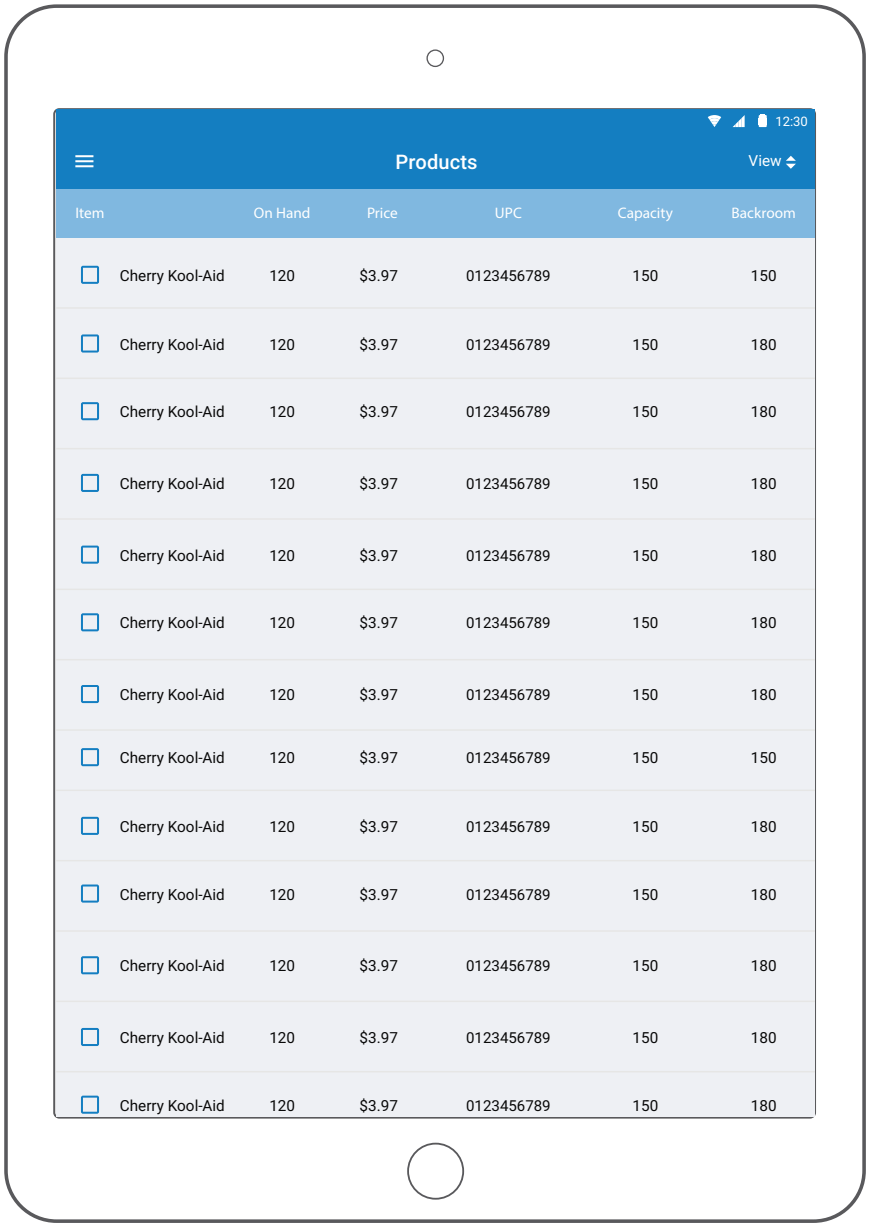


Fig. 1 Tablet table example

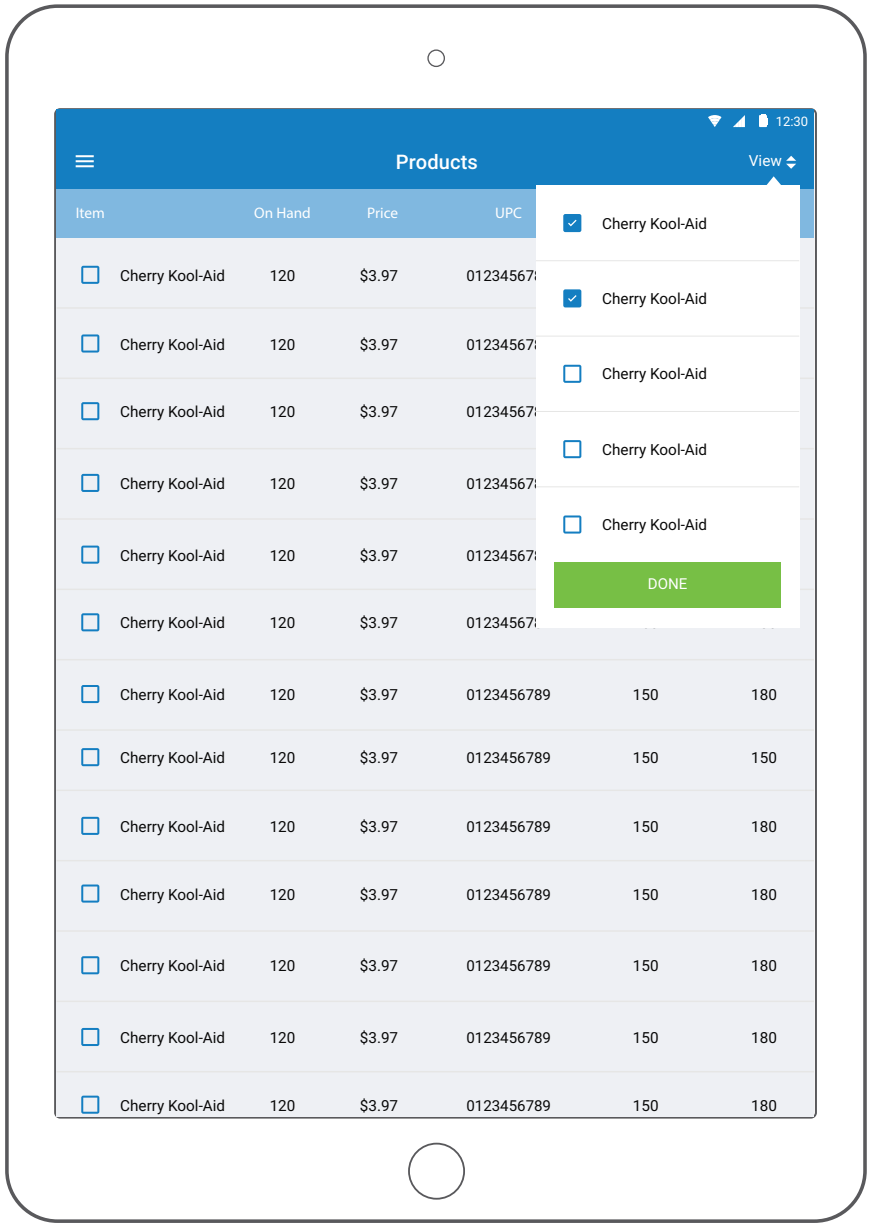


Fig. 2 Tablet table menu example

Cards

Snapshot Content

Cards are modular pieces of content broken down into individualized components. They allow the user to get a wide variety of aggregated content in an easy-to-read manner. The height of a card is variable, but the width on all cards remains constant (insert dimensions here).

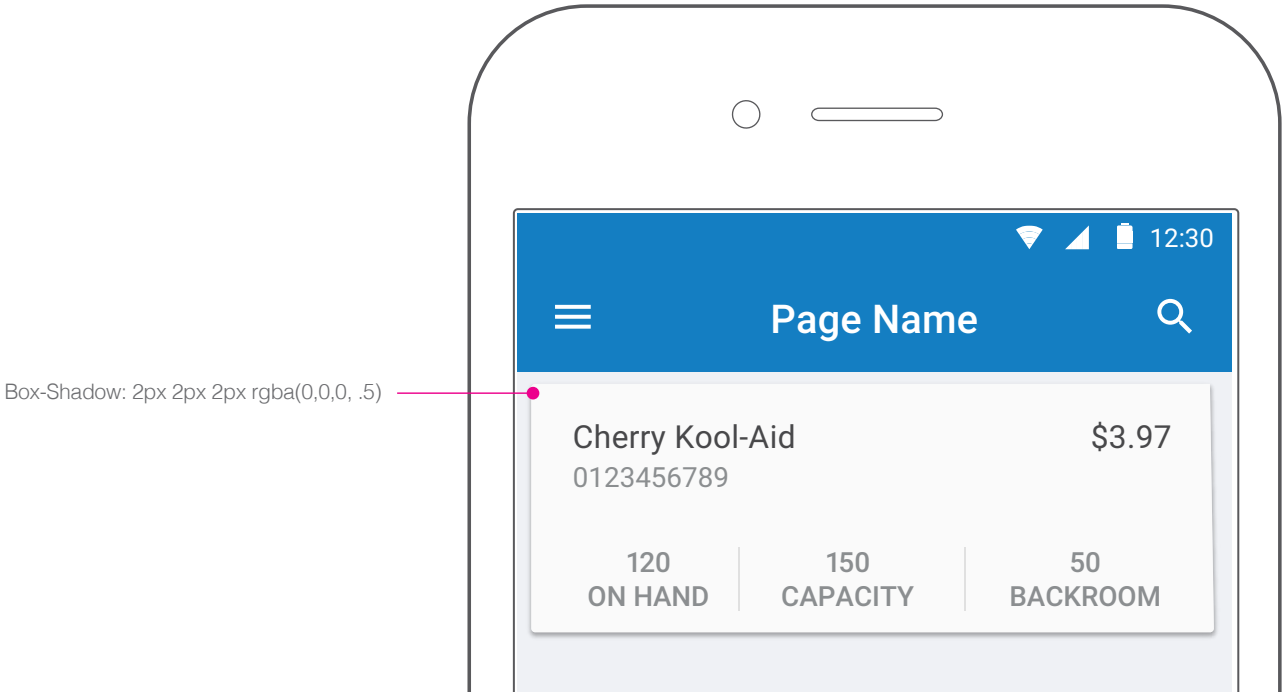


Fig. 1 Mobile example

Cards

Best Practice

When adding cards to a layout, space cards with different content types evenly (16px) (Fig. 1).
Cards that are part of a set should have no spacing between them, and a header for the entire set.

DO

- ✓ Work with stakeholders to provide the most useful card content to the end user.
- ✓ Use images and icons to convey information.
- ✓ Write clear actions that explain where the user will go when tapped.

DO NOT

- ✖ Create a card height greater than 80% of viewport.
- ✖ Have in-line actions inside card content.
- ✖ Include more than two actions inside a card.

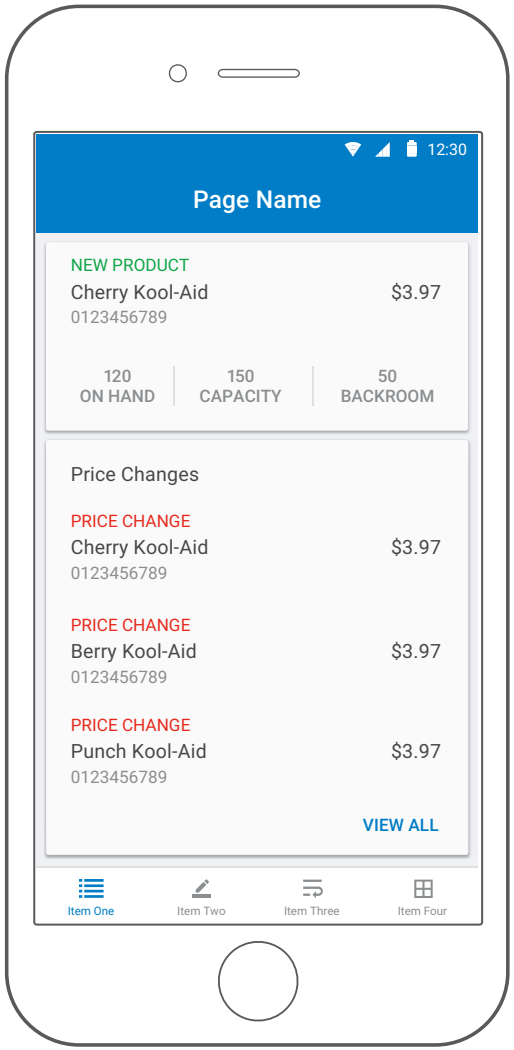


Fig. 1 Layout with multiple cards

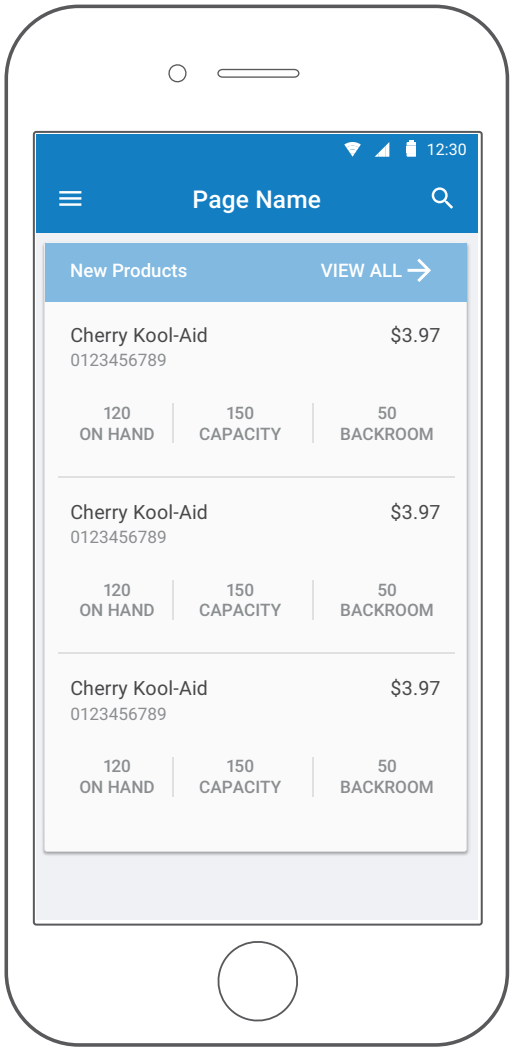


Fig. 2 Layout of cards as a list

Cards

Variations

Buttons can also be added to a card to take action (Fig. 1), or to take the user to a detail page to see more information related to the card. There should never be more than two buttons on a card.

There are multiple variations of the card layout to support different types of content. Elements can be arranged into a list inside a card, or an image/stat can be embedded into the card (Fig. 2).

Cards should tell a succinct story. Consider which elements of a card are the most important for a user, and include those with a larger font and weight. If required, a card heading can be added to the card to give additional context (Fig. 3).

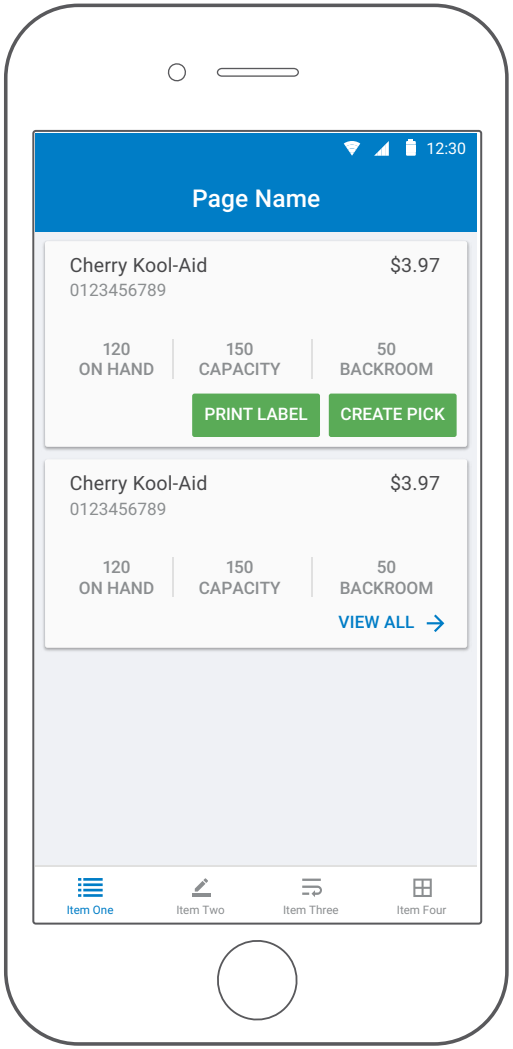


Fig. 1 Cards with actions

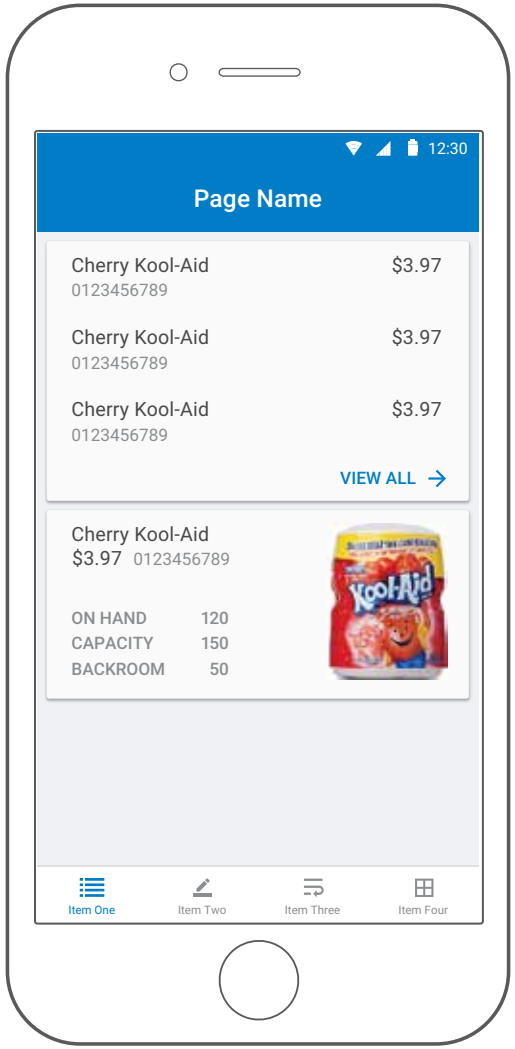


Fig. 2 Cards with images

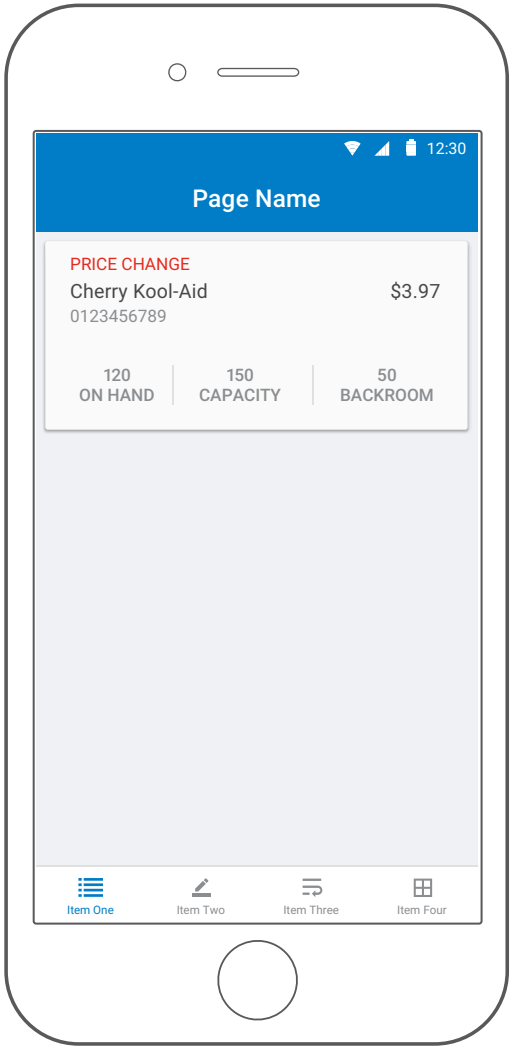


Fig. 3 Card with subject heading

Images & Videos

Delivering Image and Video Content

Studies have discovered that our brains not only process images faster, but they also retain images at a greater rate. Images also encourage greater interaction rates. This is a great opportunity, but make sure your content is relevant and contains energy. You must also be able to deliver an optimized format on multiple devices. Avoid wasted bandwidth, or this will become one of your apps biggest obstacles. Slow connections are still a major consideration, even as pages become more powerful.

Related links: https://html.spec.whatwg.org/multipage/embedded-content html#embedded-content

Related links: http://www.smashingmagazine.com/2014/05/responsive-images-done-right-guide-picture-srcset

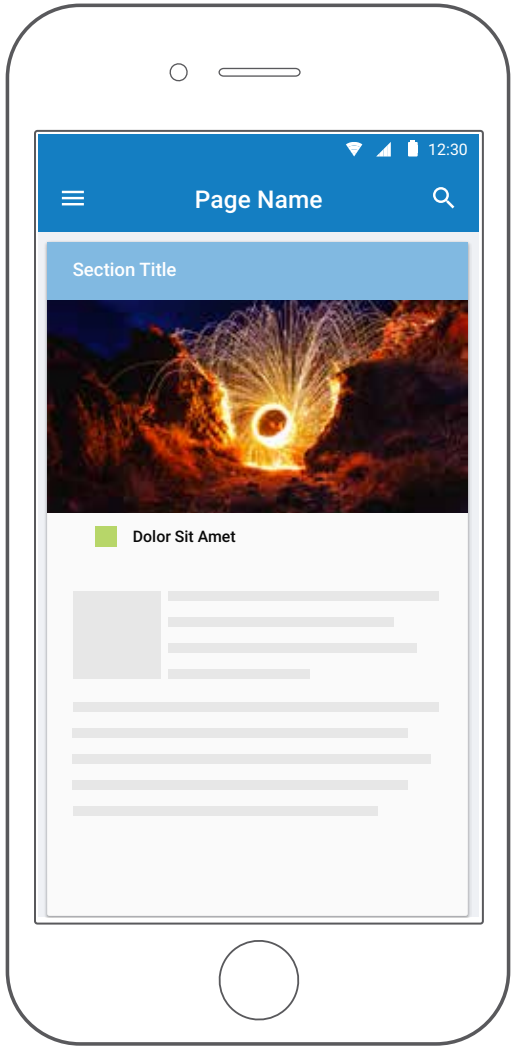


Fig. 1 Mobile example

Images and Videos

Best Practice

DO

- ✓ Use vector-based images and font icons, if possible.
- ✓ Provide automated output of your image markup.
- ✓ Put your best content forward.
- ✓ Maintain a consistent aspect ratio so your content does not letterbox or distort.
- ✓ Use high-quality video stills so users are able to see something if your video does not auto-start.
- ✓ Always use landscape view for video.
- ✓ Remove unnecessary metadata.

DO NOT

- ✖ Waste page real estate; consider the impact size of your message.
- ✖ Expect auto-start video playback.
- ✖ Use Flash on mobile.

Fig. 1 Use correct ratios

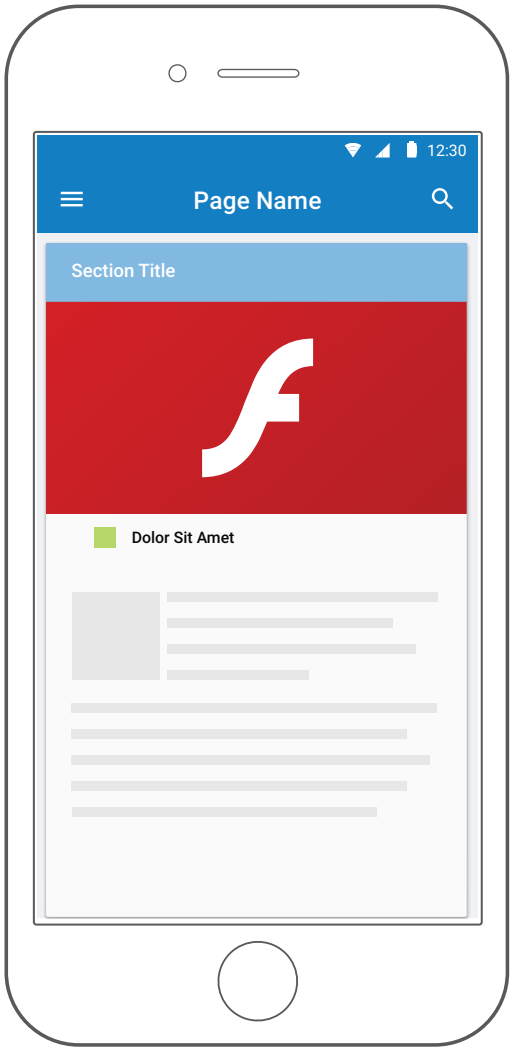


Fig. 2 Do not use flash

Images and Videos

Variations

There are a few ways to address this problem. SVG vector-based images and font icons help considerably. Responsive images were made flexible with img `{max-width: 100%;}`, but this caused jumpiness with reflow. An alternative is to use a responsive embed, setting a wrapper around the image and using a “padded-bottom” to preserve aspect ratios. Another is to create a media query to serve different CSS content to different devices. Avoid double serving resolution images, especially as this is a disservice to non-retina devices.

Video content can be handled in a similar fashion to images (Example: creating a container and setting the width or padding – another container would allow more precise sizing for the video player itself). Always note a video’s ratio to avoid cutting it off. In HTML you may have to adjust the iframe. If the content is not sensitive, a simple solution is to take advantage of server-side options that specialize in video, such as YouTube.
If your app is referring to outside content (such as inventory), and not all graphics are available, it is best to ensure a placeholder in case such graphics becomes available in the future. Implementing placeholders also protect the structure and consistency of the overall page.

Fig. 1 Always have a placeholder ready.

Fig. 2 Make sure you have permission to use 3rd party platforms

Messaging

Helping Users with Critical Information

Messages should be used whenever the end-user needs to take additional actions to complete a task or there is new information they should be made aware of immediately. Messages are often context-driven; an external event occurs, and the user is made aware of the event through a notification, alert or dialog. A well-built application will incorporate messaging throughout to help the user succeed in their goals.

6.0 Messaging

System Notifications

Messaging Outside of your Application

System notifications give the user the ability to receive information as it happens in real-time, outside of the application.

Notifications display a message and timestamp, along with an app icon to show which app the notification came from (Fig. 1).

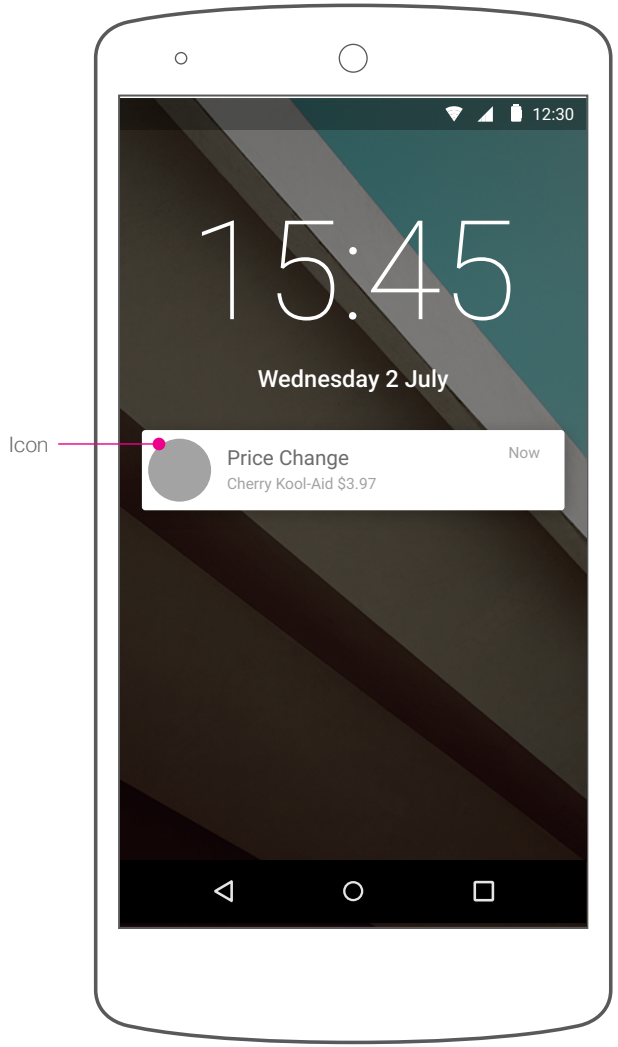


Fig. 1 Android lock screen notification

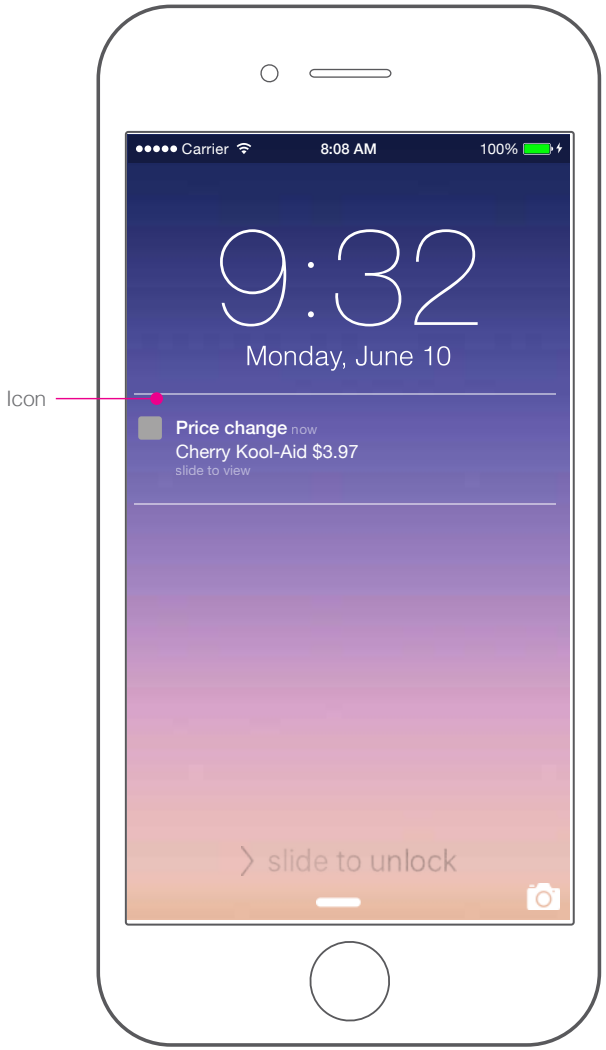


Fig. 2 iOS lock screen notification

System Notifications

Best Practice

DO

- ✓ Set rules that will make the notification expire after a certain amount of time or a set action.
- ✓ Keep your notification message short. iOS limits messages to 110 characters in lock page, 62 in banner.
- ✓ Let users control the type and frequency of notifications on your settings page.

DO NOT

- ✖ Include your app name in the message.
- ✖ Repeat notifications.

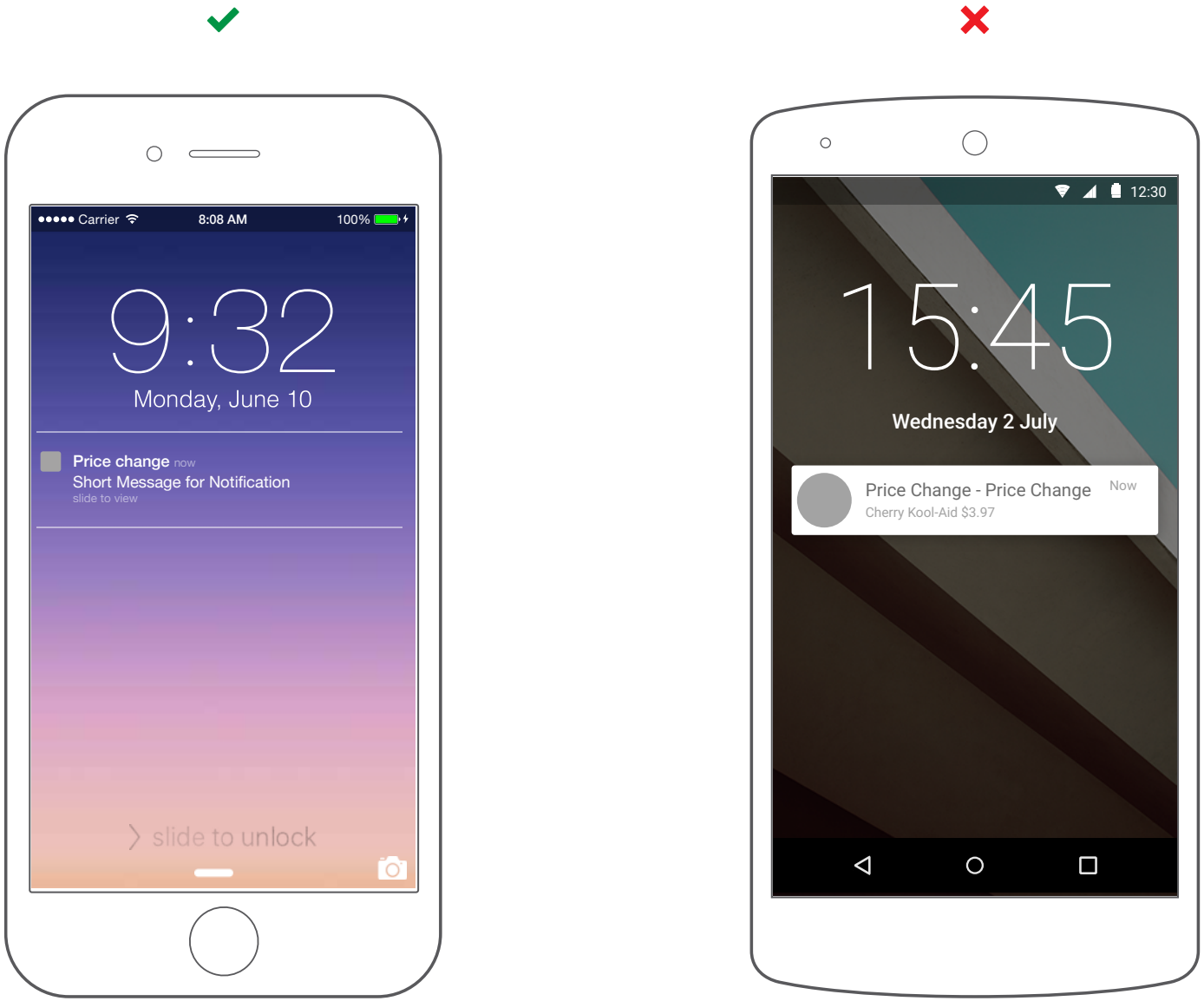


Fig. 1 Do not include the app name in the notification

System Notifications

Variations

While push notifications rely on a server-side connection to broadcast a message, apps can also use “local” notifications to trigger at a specific time. For example, an app can specify a meeting reminder 10 minutes before it is supposed to occur **(Fig. 1)**. Use local notifications whenever possible, since local notifications are less dependent upon a strong device signal.

Different operating systems dictate specific guidelines on how notifications should be handled. It is important to stay up-to-date on these requirements.

iOS: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/NotificationCenter.html

Android: http://developer.android.com/guide/topics/ui/notifiers/notifications.html

Fig. 1 Local notifications are time based and do not rely on a server connection to publish the notification

Alerts

Putting Important Information in Front of the User

Alerts appear inside the app to give the user important information that has occurred since they last used the app.

An alert box is a container that displays all alerts for the user. The alert box will typically be part of the “Front Page” experience, but it could be found deeper within an app’s hierarchy if necessary.

The user has several options with alerts:

- The alert can be linked to the subject, allowing the user to take a shortcut to where they need to go to take action.
- A single alert can be dismissed so that a user never sees the alert again.
- If the user wants to dismiss several alerts at once, they can tap “dismiss all” from the alert box heading.

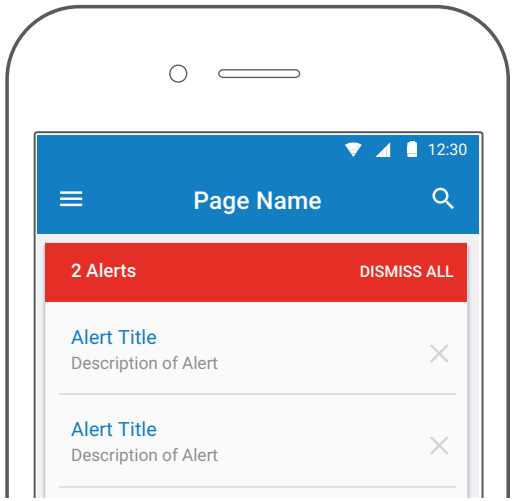


Fig. 1 Mobile example

Alerts

Best Practice

DO

- ✓ Use alerts sparingly. Users will ignore alerts if they are not useful.
- ✓ Remember that alerts are used as a shortcut; they are not a primary path to a destination.
- ✓ Include logic that will make the alert expire if it is no longer relevant to the user.

DO NOT

- ✖ Duplicate alerts. If the user has not dismissed the alert, they should not receive the same alert twice. Consider updating the alert instead.

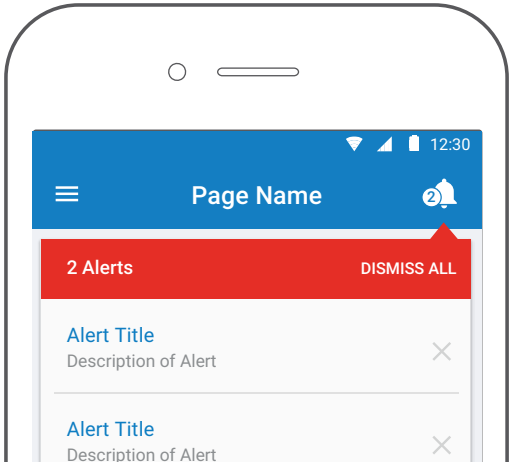


Fig. 1 Set alerts to expire after a certain amount of time

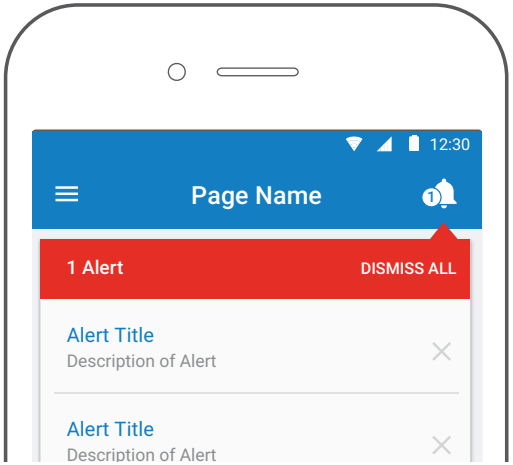


Fig. 2 The user should not receive multiple alerts with the same message

Alerts

Variations

An app where alerts frequently occur could have an alert icon in the header (Fig. 1). This approach maintains the visibility of alerts while taking up less space.

If an alert requires a simple action, the action can be embedded into the alert itself (Fig. 2).

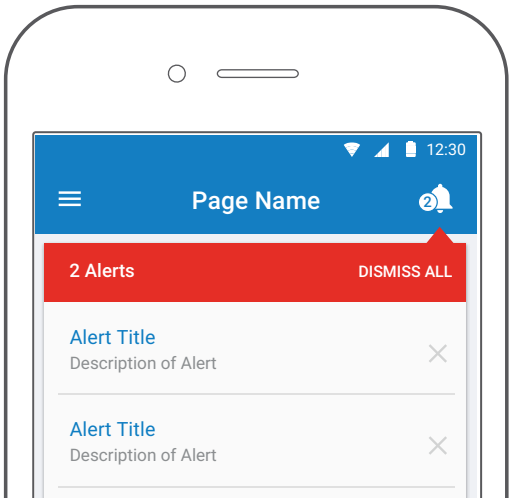


Fig. 1 Alerts within header

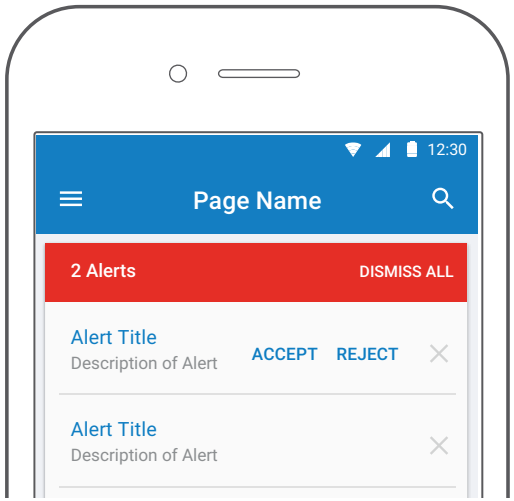


Fig. 2 Alerts with action

Dialog Boxes

Critical Messaging

A dialog box is any sort of message that appears on top of the page to inform the user of critical information. The dialog box can also include actions for the user to take or a link to visit a page to get additional details on the message.

The bare essentials of a dialog box consist of a short message and an action the user can take related to the message (Fig. 1). When useful, a dialog title can be added to the message to give additional support on what the dialog box is referring to (Fig. 2).

Actions that the user can perform related to the message always occupy the bottom of the dialog box. The rightmost position of the dialog box is considered the primary spot.

Actions that would be the most natural choice for the user to take should occupy this spot. Actions that are considered “destructive” (Example: delete, not save), should always occupy the leftmost position.

On tablets, the dialog boxes are aligned both vertically and horizontally to the user’s device (Fig. 3).

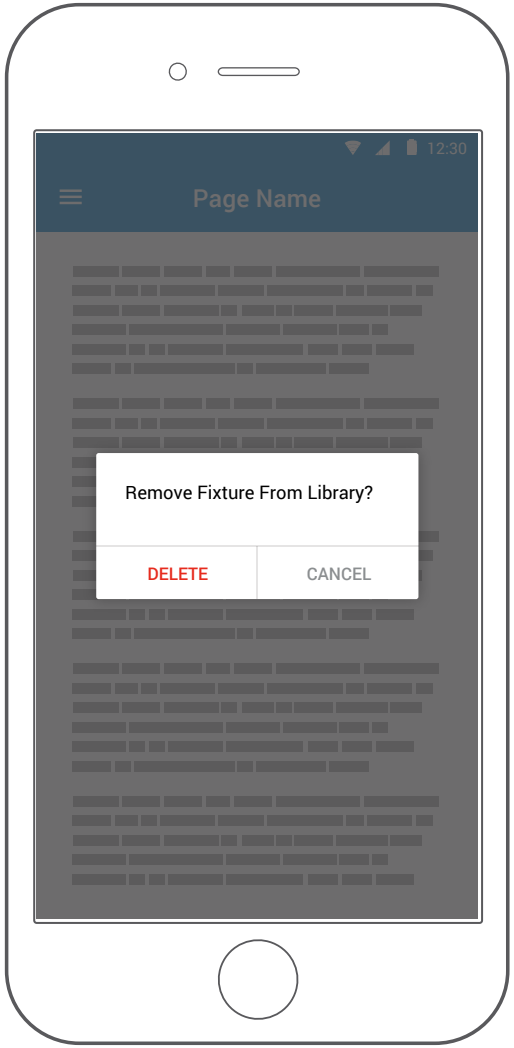


Fig. 1 Mobile example

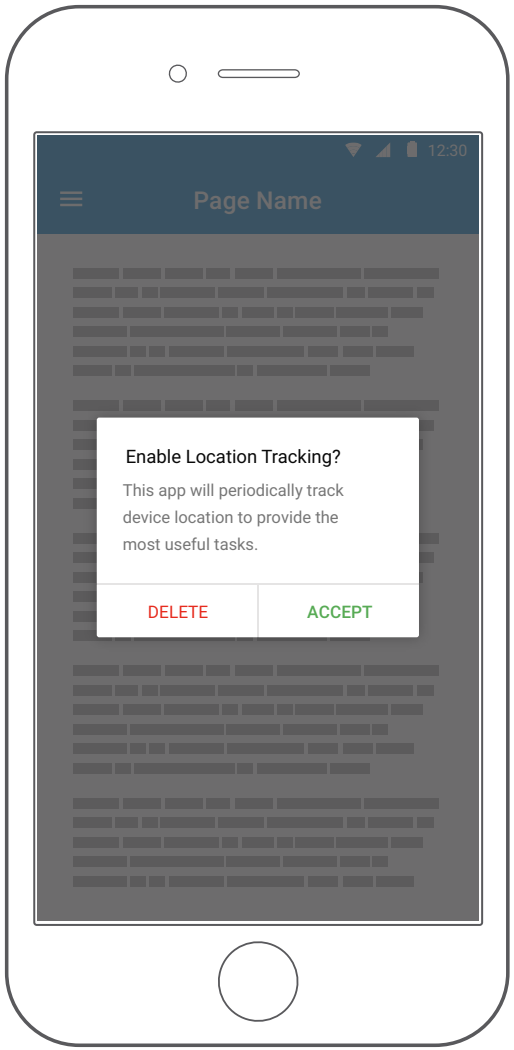


Fig. 2 Dialogue with message

Fig. 3 Tablet example

Dialog Boxes

Best Practice

DO

- ✓ Keep dialog boxes as simple as possible.
- ✓ Try making a dialog title a question. Questions are clear and easy-to-understand.
- ✓ Capitalize every word in a dialog title, and do not end with a period.
- ✓ Try to limit actions to two choices. It is easier for users to understand their options when there only have two to choose from.
- ✓ Label actions as verbs (Example: Agree/Disagree, Reply/Ignore).

DO NOT

- ✖ Have a single-word title. Titles such as “Error” or “Warning” add little to telling the user what they need to know.
- ✖ Use exclamation points or other non-neutral language.
- ✖ Be ambiguous in action labels. Users must know with certainty what will happen when they tap an action.
- ✖ Create a dialog box with just a title. A dialog box can have a message with no title, but it cannot have a title without a message.
- ✖ Allow dialogs to scroll.
- ✖ Have multiple actions or steps occur with dialogs. Anything that requires more than one user action should happen on its own page.

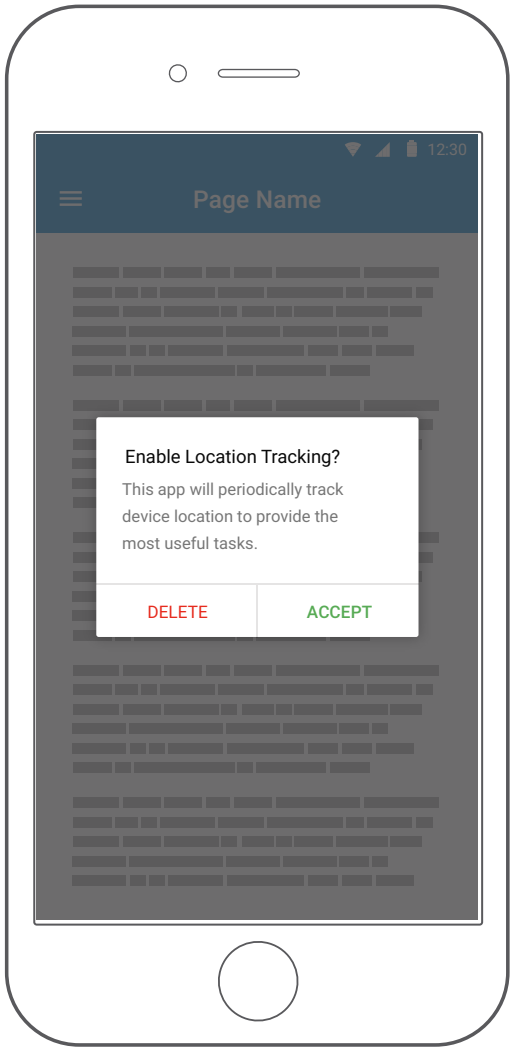


Fig. 1 Finite loading

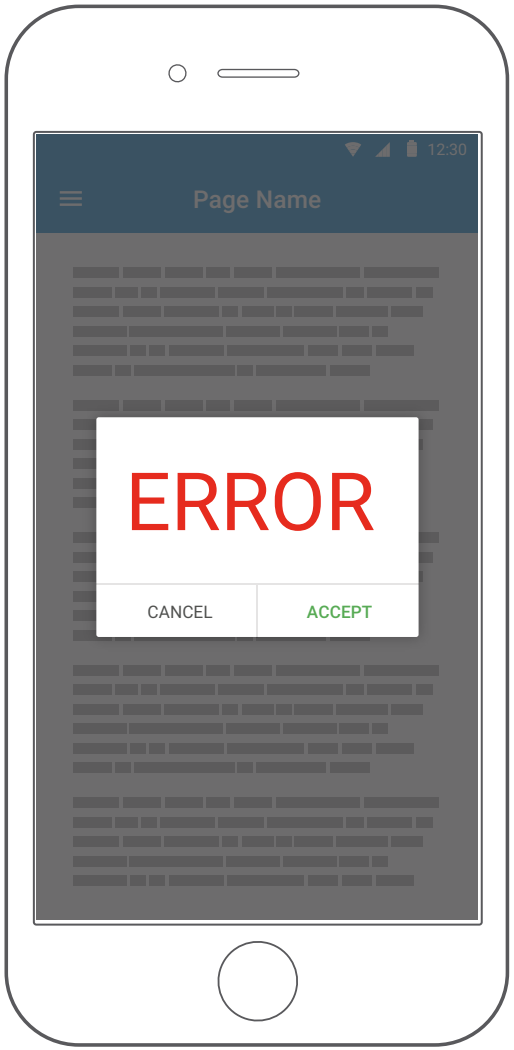


Fig. 2 Infinite loader

Dialog Boxes

Variations

A dialog box can also include more advanced actions, such as radio buttons or checkboxes (Fig. 1, 2). When using advanced functionality, keep the options limited to a select few.

dialog boxes with more than three options can stack vertically, with the default option being in the highest position (Fig. 3). A dialog box can also include an input field, and should include an example to provide additional clarity.

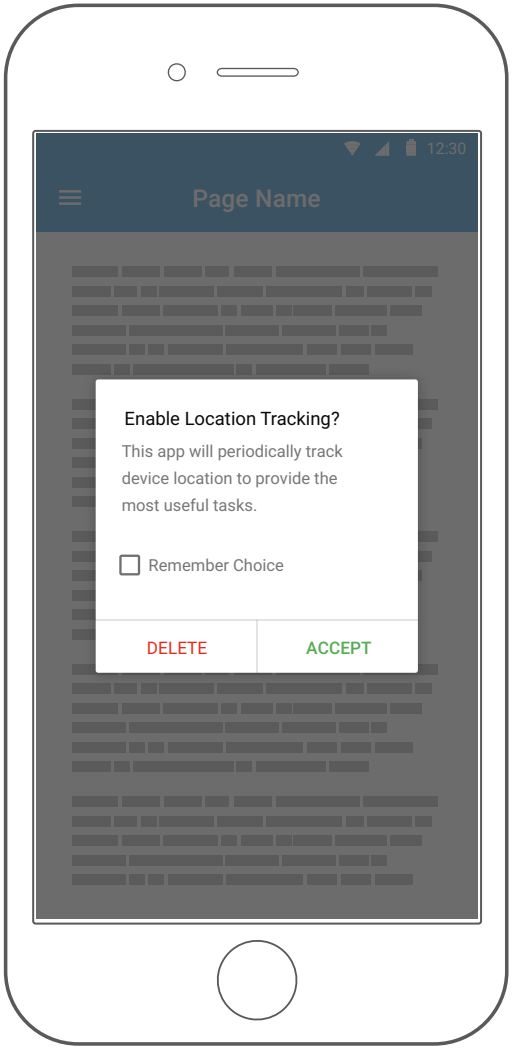


Fig. 1 Dialog with checkbox

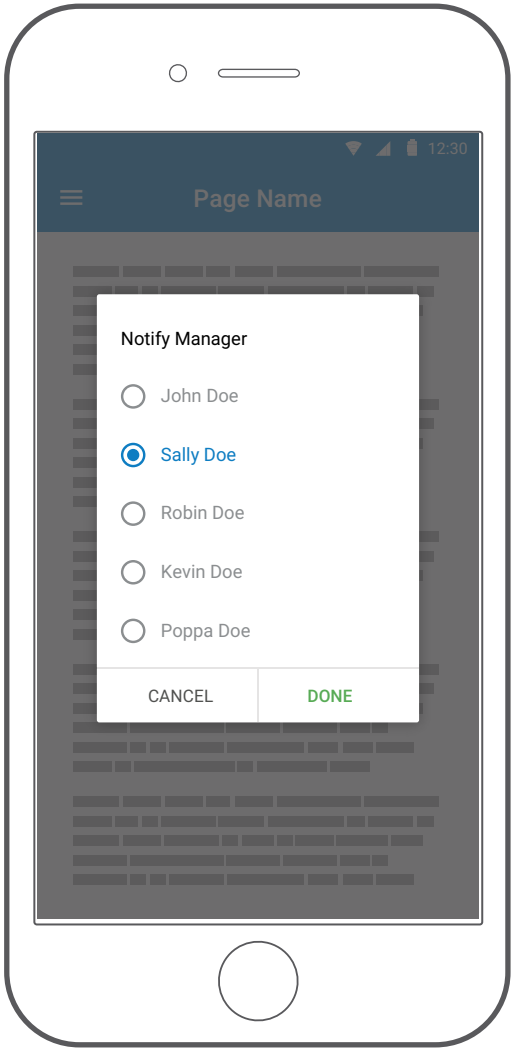


Fig. 2 Dialog with radio buttons

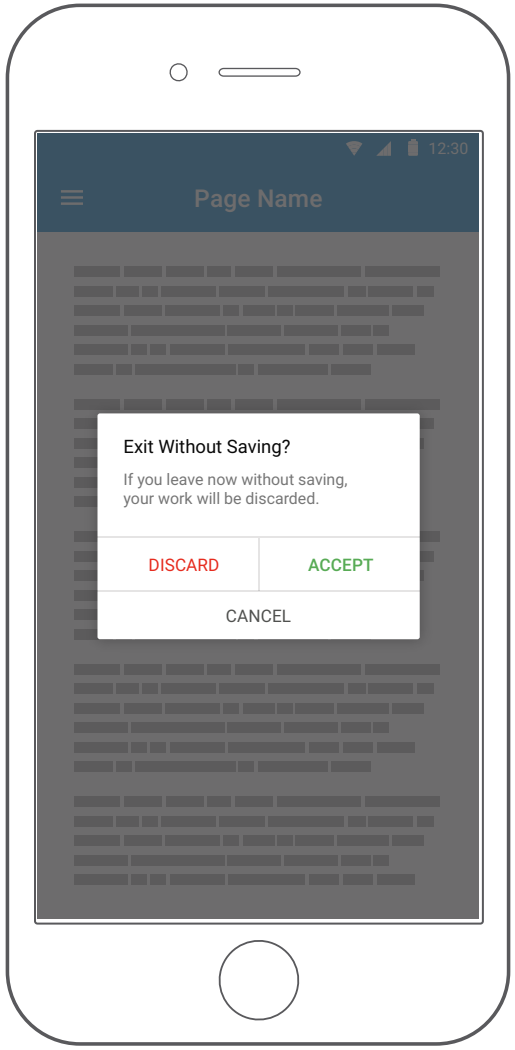


Fig. 3 Dialog with three options

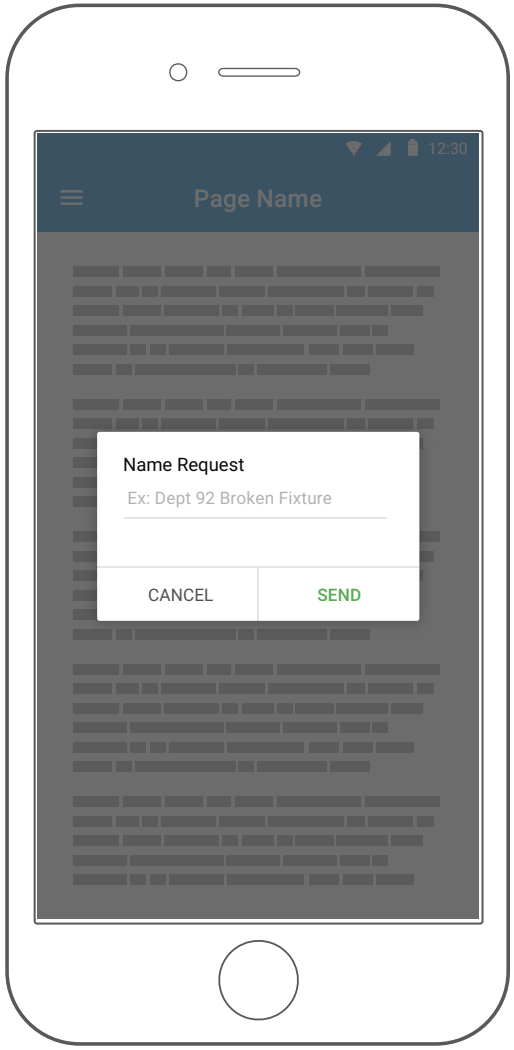


Fig. 4 Dialog with form field

Tooltips

Relevant Information on an As-needed Basis

A tooltip is a short message that gives the user additional information about an element on the page. The user can view the tooltip by interacting with the element, for example by tapping on a data point in a visualization or on a cell inside a table.

A tooltip has a container with a directional arrow and a short message inside the container (Fig. 1). Tapping an element will trigger the toolTip to animate into view. Making any interaction outside of the tooltip will cause the toolTip to disappear.

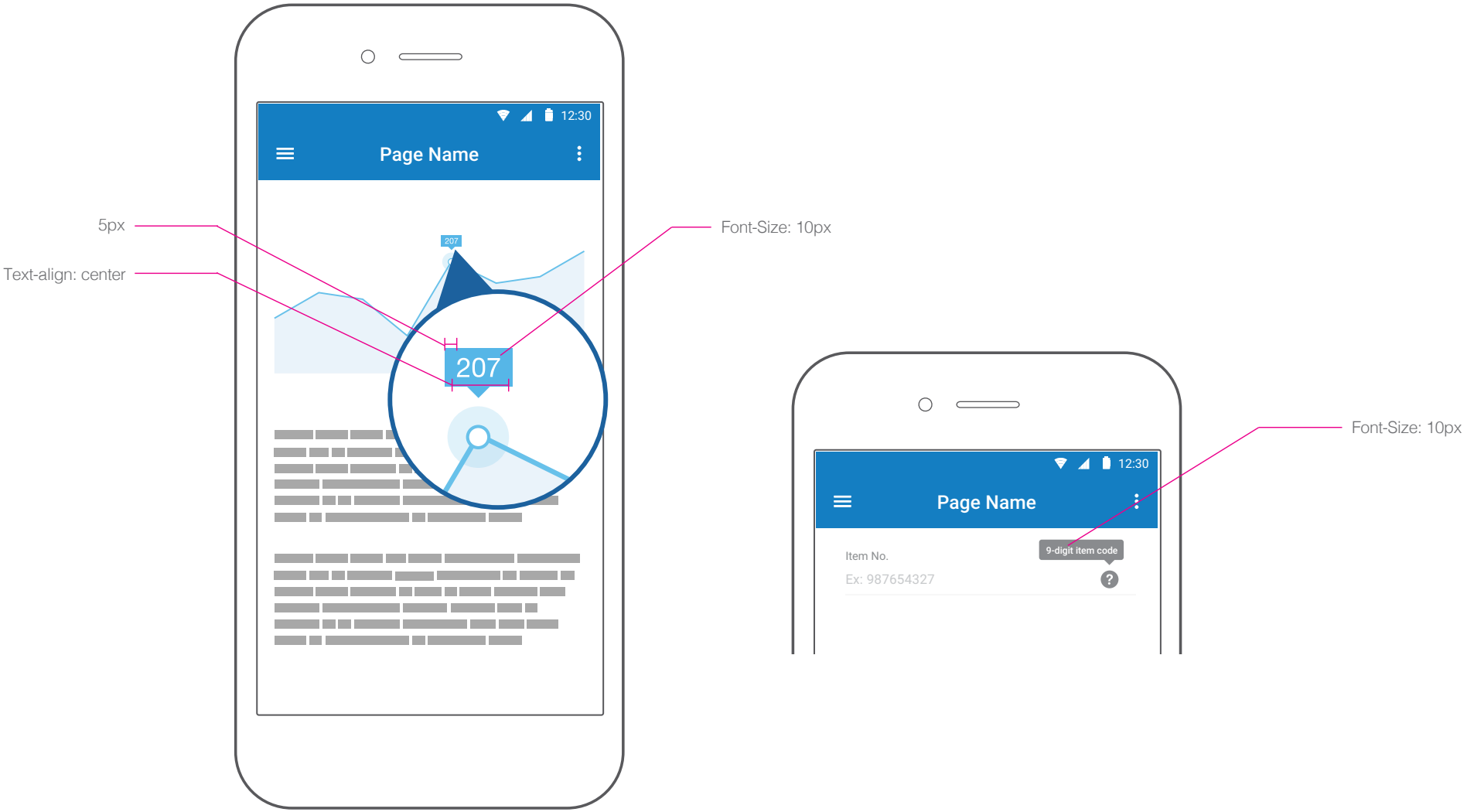


Fig. 1 Mobile example

Tooltips

Best Practice

DO

- ✓ Position tooltips in such a way as to prevent them from going off the page (Fig. 1).
- ✓ Use a consistent tooltip color that provides a lot of contrast to the page body.

DO NOT

- ✖ Include links or actions inside tooltips. Consider using an accordion instead.
- ✖ Exceed more than two lines of text or truncate text.

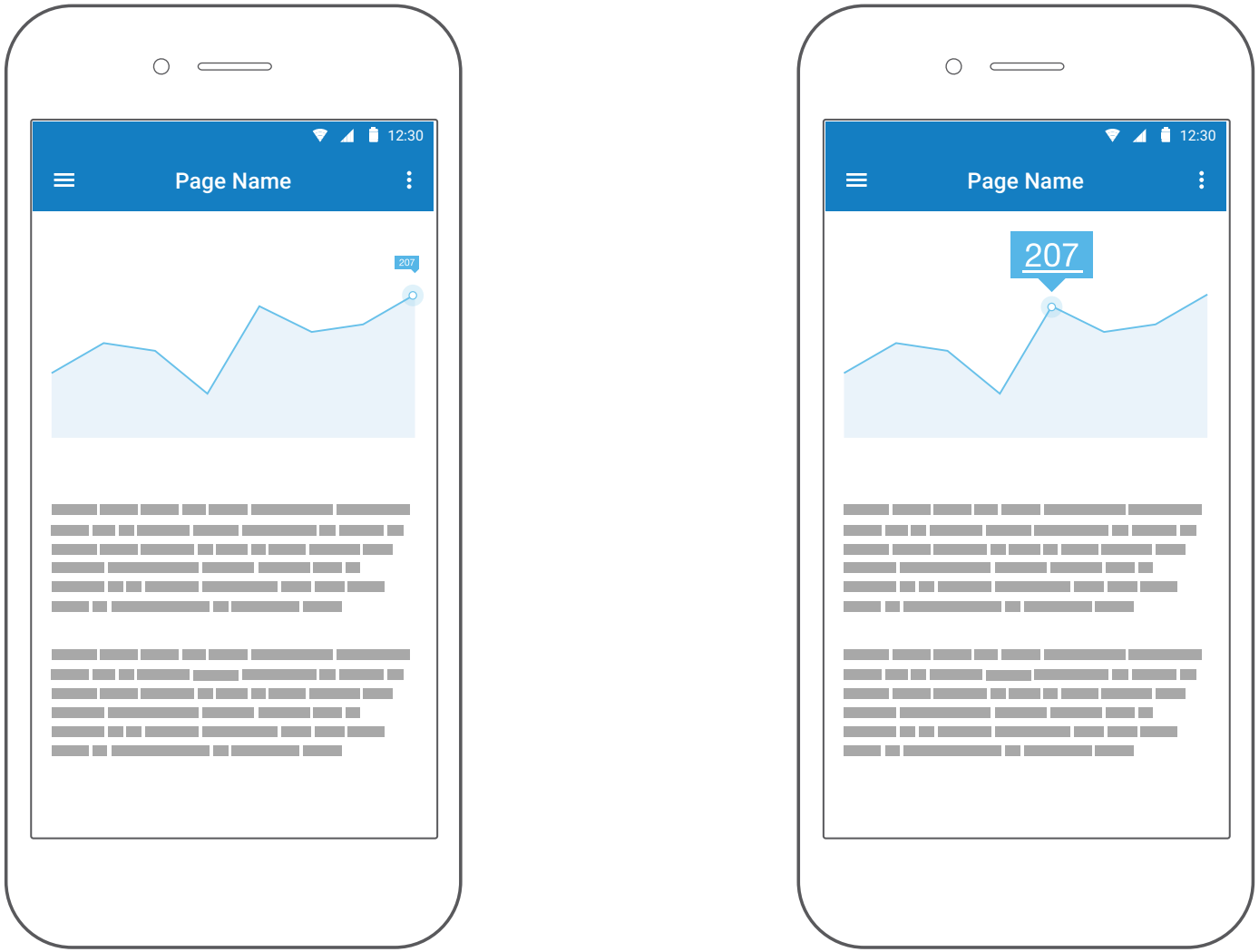


Fig. 1 Do not make a tooltip a link

Tooltips

Variations

Tooltips can include iconography, although it should be very simple.

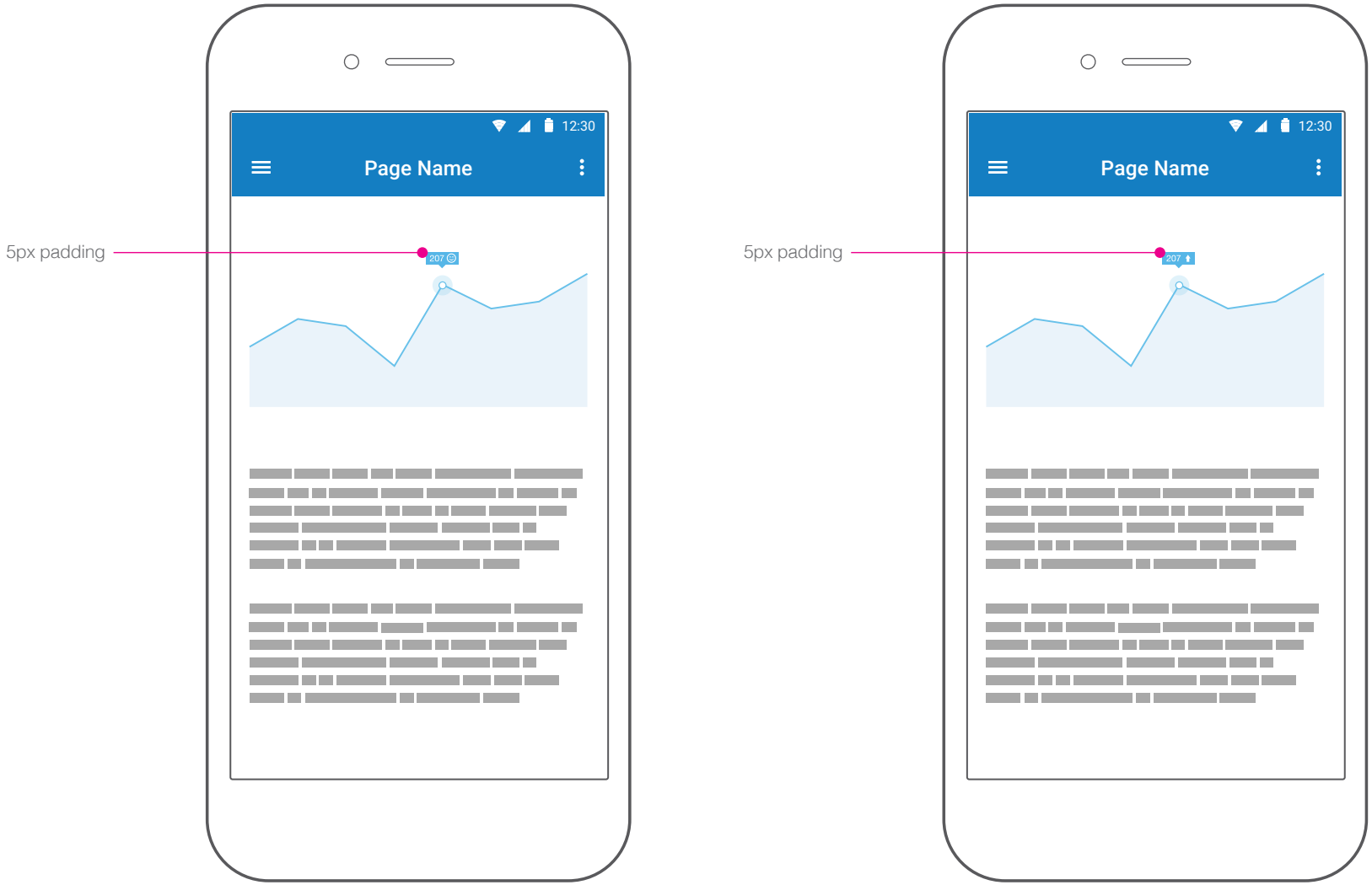


Fig. 1 Iconography in tooltips should be simple and understandable

Data Visualization

Communicating Large Amount of Data through Clean, Concise Images

Due to sheer amounts of data and the minimal real estate available on a mobile page, spreadsheets are hard to visualize and comprehend on most devices. Printing all that data is little help either as the information still needs to be processed, just now in physical format.

The best data visualizations consolidate data into a simple way that exposes important data. With touch technology we are able to take it a step further. With a tap we are able to allow users to drill down into charts for more details, and interact to change what is seen and how it is processed.

- Determine what type of data you are trying to communicate.
- Use a visual to convey that information in the simplest way possible.
- When displaying information on touch, designate an area on page that will update accordingly.
- Know how your audience will interpret your data.
- Remove all but what is critical to informing the user.

7.0 Data Visualization

Line Graphs

Line Segments Connecting Data Points to Highlight Relationships over Time using Continuous Data

Ideal for showing trends over time, line graphs are very flexible in how they can be applied, both visually and informatively.

They are also good for tracking relationships between two values for correlating trends. (But only if the axis follows the same scales.)

A line graph may fill the width of a phone page as well as a tablet. On the phone it can be reduced to its most fundamental elements to allow for more detail when there is real estate (such as on the tablet). Conditions may be included to allow the user to interact and drill down or adjust data in the line graph.

Key information must always be labeled next to the corresponding line or via key. Always try to include a zero baseline for best results.

Fig. 1 Mobile example

Fig. 2 Tablet example

Line Graphs

Best Practice

DO

- ✓ Use zero as the baseline, unless impossible.
- ✓ Show changes over a period of time for more than one group.
- ✓ Use one color per data set. Accent colors may be used to highlight significant data points.
- ✓ Label lines directly, if possible, versus using a legend.
- ✓ Equally spaced intervals in a line graph.

DO NOT

- ✖ Show more than four lines in a graph.
- ✖ Plot data points to cover more than 2/3rds on the y axis of the graph.

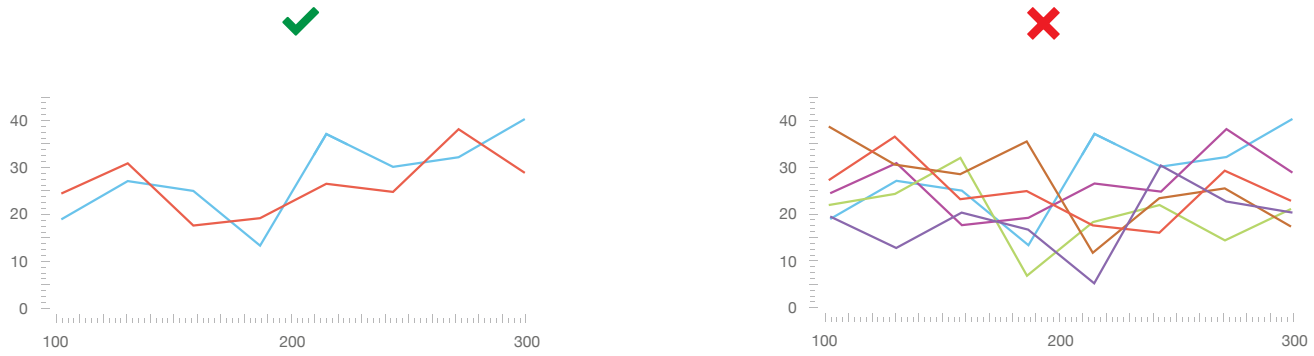


Fig. 1 Too many lines make the graph difficult to read

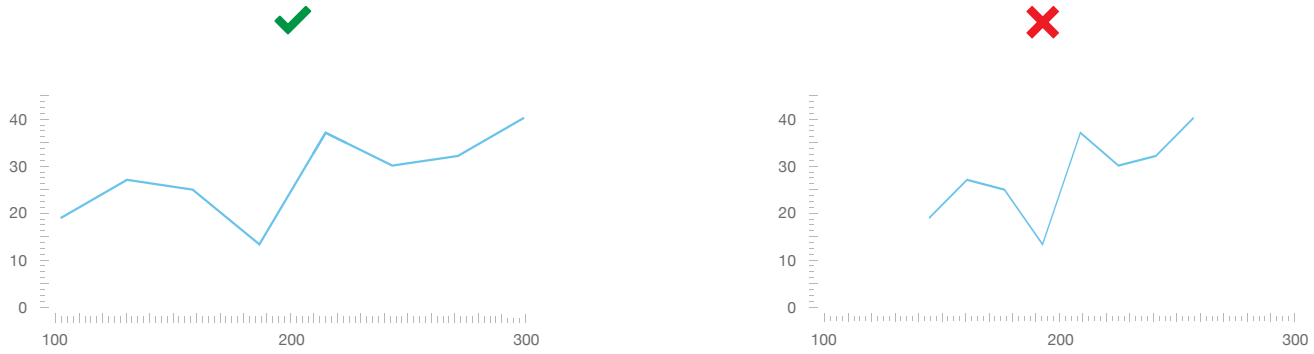


Fig. 2 Horizontal axis should be scaled to the start and end data points

Line Graphs

Variations

It is recommended to show a more detailed line graph if there is more page real estate, such as for desktop or tablet. When the user is on a mobile phone, hide less critical details and show a more simplified version.

A sparkline is the most simplified version on a line graph, useful for areas in-line with text, such as on a table. All visual elements must be hidden. Highlighting the first and last data points is recommended.

Various graph names include: Spark Graph, Sparkline, Point to Point Graph

A spark line graph is a line graph in its simplest form. All visual elements must be hidden except the data series, and highlighting the first and last data points is recommended.

FIG. 1 Sparkline

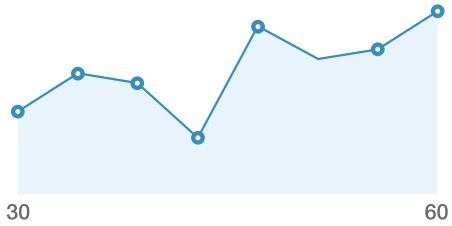


Fig. 2 Simplified line graph with filled space

Fig. 3 Simplified line graph with filled space (Color Reversal)

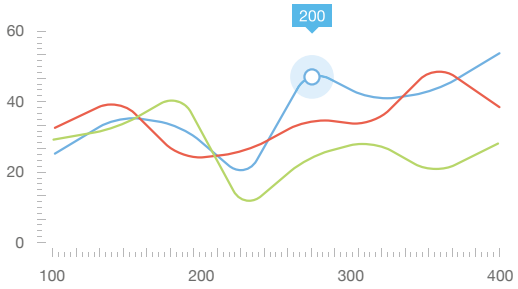


Fig. 4 Detailed graph with tooltip and rounded points

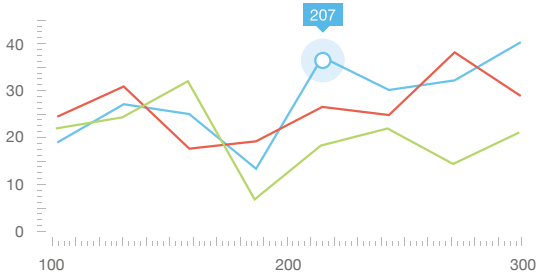


Fig. 5 Detailed graph with tooltip

Information Title 

Fig. 6 Title

Bar Graphs

Ideal for Showing Comparisons and Trends

Flexible, bar graphs may be represented vertically or horizontally, with the axis for a specific category and another for the set value being compared. Bars can be grouped in clusters of more than one or stacked to show cumulative effect. Unlike the pie graph, a bar graph may fill the width of a phone page as well as a tablet. On the phone it can be reduced to its most fundamental elements of bars and sum data (conditions can be included that the user can interact with to see more such as tabs (Example: day, week, month) that may be layered back in with tablet. Other acceptable interactions may include the ability to tap a bar to highlight and display a figure, a number popup or popup displaying drill-down information.

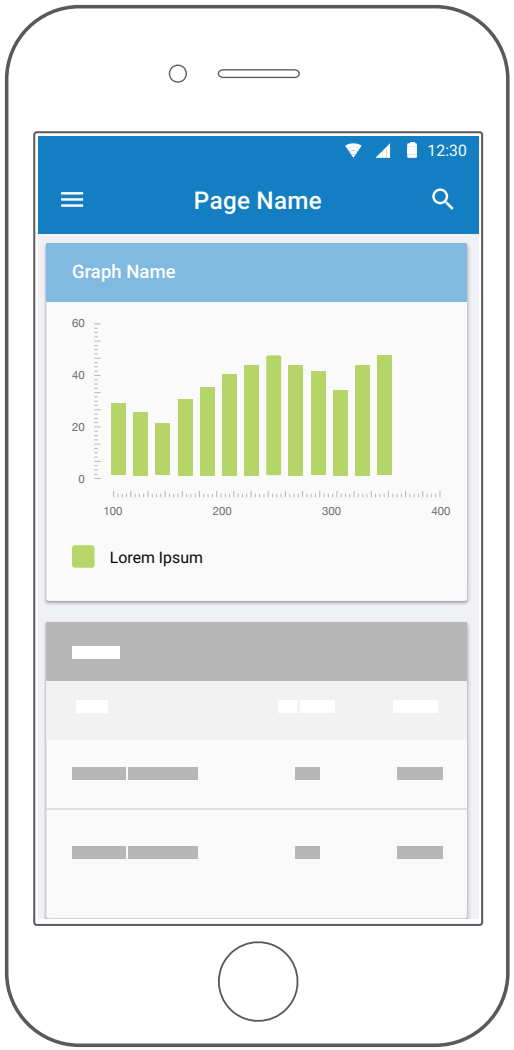


Fig. 1 Mobile example

Fig. 2 Table example

Bar Graphs

Best Practice

DO

- ✓ Compare values of different categories (Example: the year's yields of various items) or compare parts of a whole (Example: percent distribution of movie sales across various genres) to show change over time (Example: net monthly earnings over a year).
- ✓ Space bars equally and consistently. Space equal to 50% of the bar width is recommended, except for histograms.
- ✓ Use one color per data set. Accent colors may be used to highlight significant data points.

DO NOT

- ✖ Start a value above zero.
- ✖ Overlap bars. Set spacing width to 50-150% width of bars.
- ✖ Use pattern fills and borders, except for highlights.

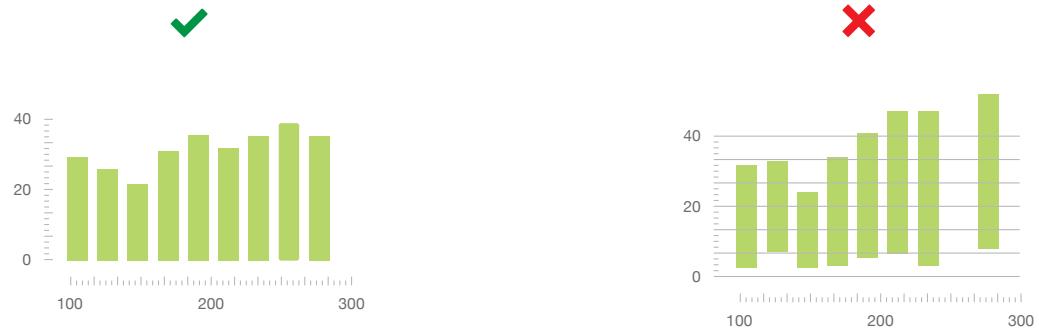


Fig. 1 Start all your bars from zero

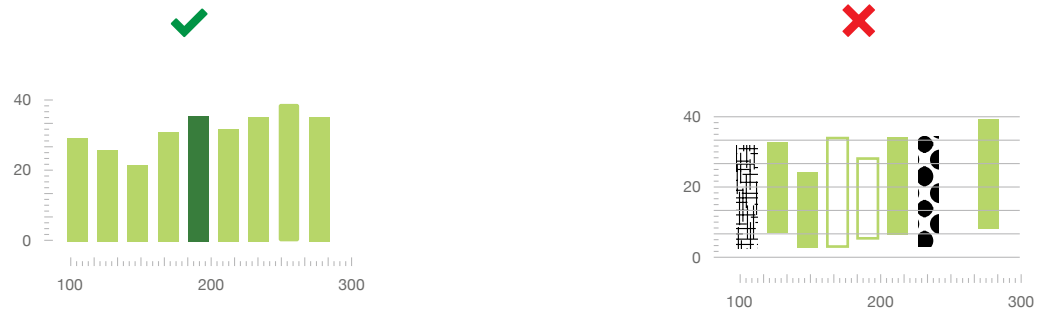


Fig. 2 Use color consistently; only use a second color for a data set if as an accent

Bar Graphs

Variations

Bar graphs can be displayed in a variety of ways. One of such is the using horizontal orientations when charting different categories, especially those with long labels. Vertical orientations are best used for chronological data or when negative values are involved.

Grouped bar graphs display the different subgroups of main categories as separate bars with their own coloring.

Stacked bar graphs also displays parts of data, but stacks groups in vertical segments. The height represents the combined results of a group.

Various graph names include: Horizontal bar charts, grouped bar charts, stacked bar charts, column chart

- A histogram also displays information with a series of rectangular bars, but groups data into ranges rather than categories and does not use gaps. Histograms are used to display univariate data sets in a way that show's the data's frequency distribution. Examples include weight, height, how much time, ect.
- Try using horizontal orientations when charting different categories, especially those with long labels. Vertical orientations are best used for chronological data or when negative values are involved.

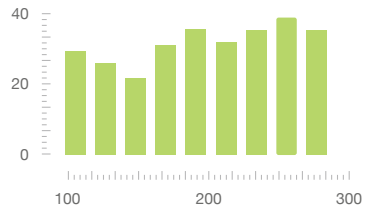


Fig. 1 Standard bar graph

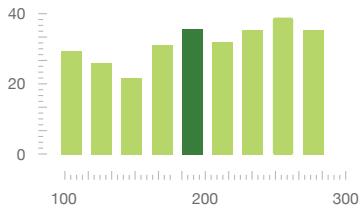


Fig. 2 Standard bar graph with accent color

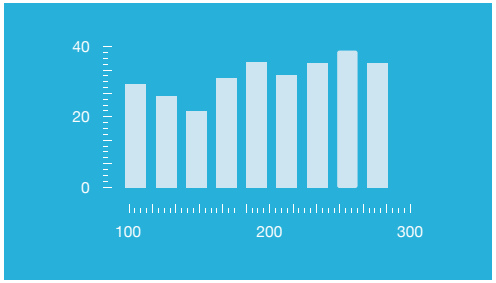


Fig. 3 Alternate color treatment (color reversal)

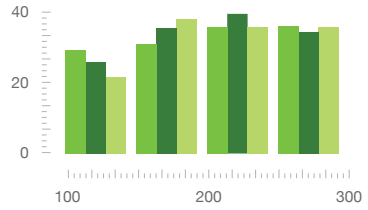


Fig. 4 Standard graph with three data groups

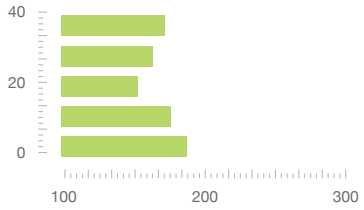


Fig. 5 Vertical data graph

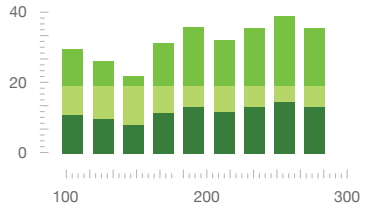


Fig. 6 Stacked bar graph

Pie Charts

Ideal for Visualizing the Proportions of a Complete Data Set

It is critical to keep your pie charts succinct in mobile. The pie chart can be displayed without a frame or on a card. Users can interact with pie charts to display deeper levels of information. Slices can be tapped to “explode outwards” separately from the rest of the circle to highlight specific data points. If the pie chart is a donut then a specific number, icon or title can be displayed in the center to support the data being displayed.

For tablets and desktop do not stretch a pie chart to fill the entire width, but arrange it with the legend to the side or with a card displaying more information next to it to fill the space.

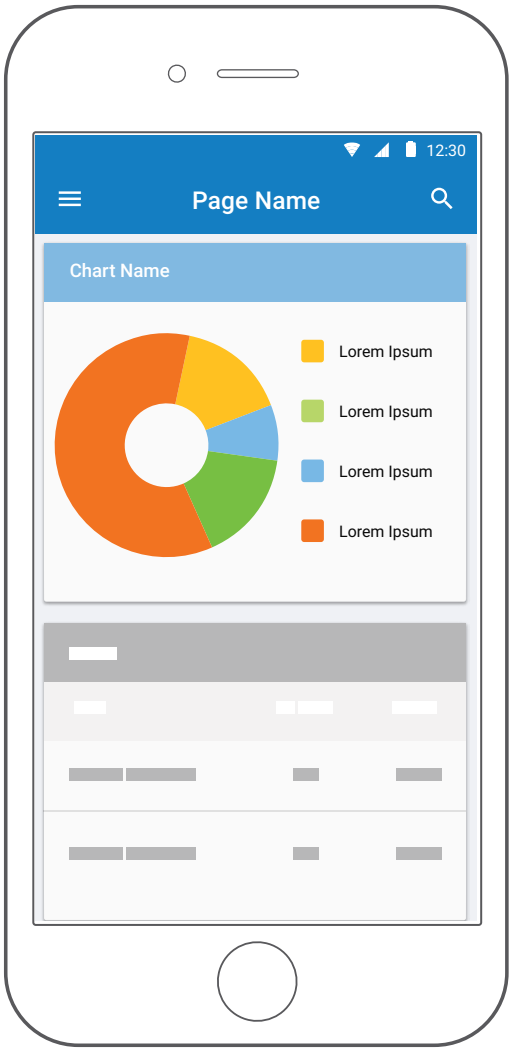


Fig. 1 Mobile example

Fig. 2 Tablet example

Pie Charts

Best Practice

DO

- ✓ Have the information make up a meaningful whole. The pie chart is to show part-whole relationships.
- ✓ Make sure the data adds up to 100%.

DO NOT

- ✖ Use the chart if there is overlapping data between parts.
- ✖ Create pie charts to compare data sets to each other.
- ✖ Include more than seven slices. Four is the recommended max. If absolutely necessary, group smallest slices into one “Other” grouping.
- ✖ Use to show comparisons over a period of time.
- ✖ Create a 3D chart – it visually distorts the perspective of the information

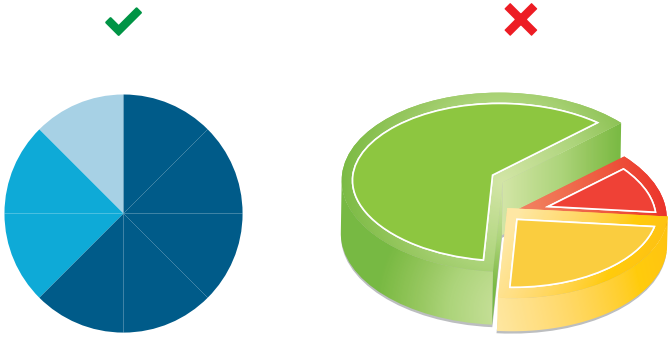


Fig. 1 3D models skew perspective

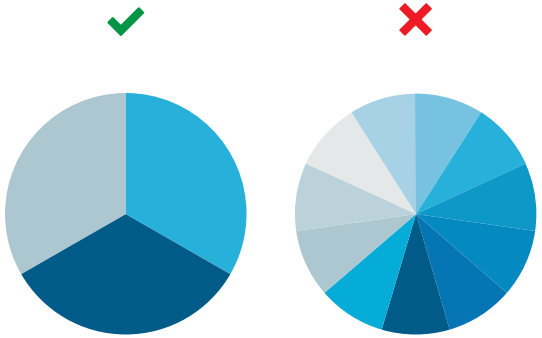


Fig. 2 A maximum for 4 slices is recommended, but no more than 7

Pie Charts

Variations

PIE CHART VARIATIONS

Exploded pie chart, donut chart, and multi-level pie chart (ring or sunburst chart) pie chart, radial pie chart and polar area chart. (The polar chart is also known as a radar chart, web chart, spider chart and star chart).

LEGEND VARIATIONS

If a simple percentage radial, the total and title may be displayed next to the chart. If not, then display the legend with a bullet or tab list. Legend must be consistent with that of any other graphs used. Do not use icons in legends.

Fig. 1 Exploded pie chart

Fig. 2 Donut pie chart

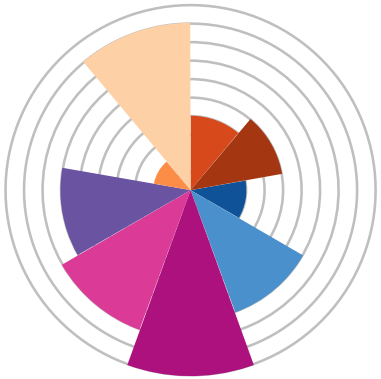


Fig. 3 Sunburst ring chart

Fig. 4 Concentric/radial pie chart

Scatter Plot Graph

For Showing the Relationship Between Two Sets of Data

Unlike other charts, scatter charts are good for displaying trends, clusters and stories that can be interpreted in different ways. These are ideal for mobile phone views for their minimal real estate demands.

Scatter graphs are a simple set of data points representing the relationship between two variables plotted along an x and y axis.

WHEN CREATING A SCATTER GRAPH IDENTIFY YOUR VARIABLES:

Independent Variable – The data that can be controlled or changed
Dependent Variable - The data controlled by an outside factor

One example would be the association between the size of a display along an x axis and its retail price along the y axis, the trend showing the correlation as possibly positive or negative.

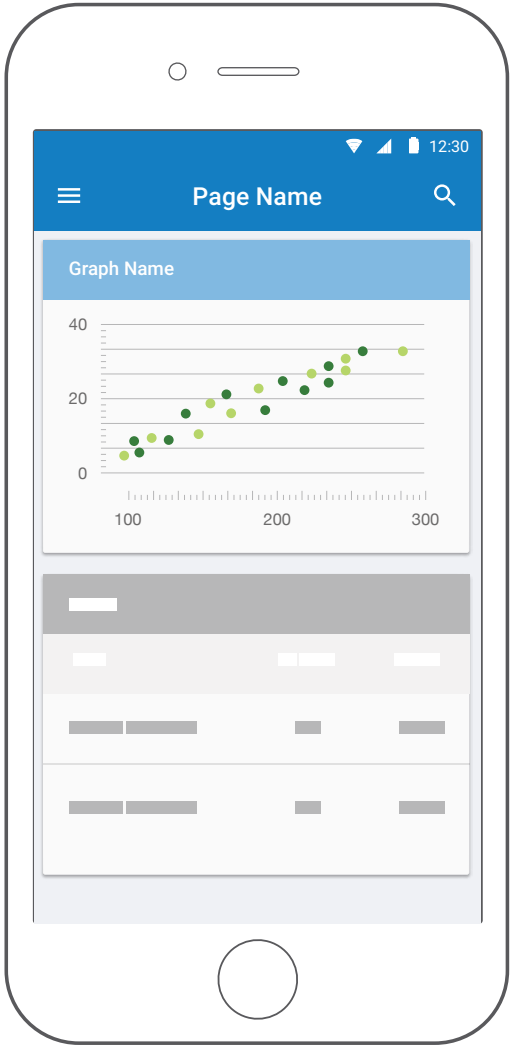


Fig. 1 Mobile example

Fig. 2 Tablet example

Scatter Plot Graph

Best Practice

DO

- ✓ Start Y axis with zero as the baseline.
- ✓ Alter dot size and color to communicate variables.
- ✓ Use trend lines to help show correlation in variables to highlight trends.

DO NOT

- ✖ Use more than two trend lines.
- ✖ Use when data may need to be rounded up or is vague.

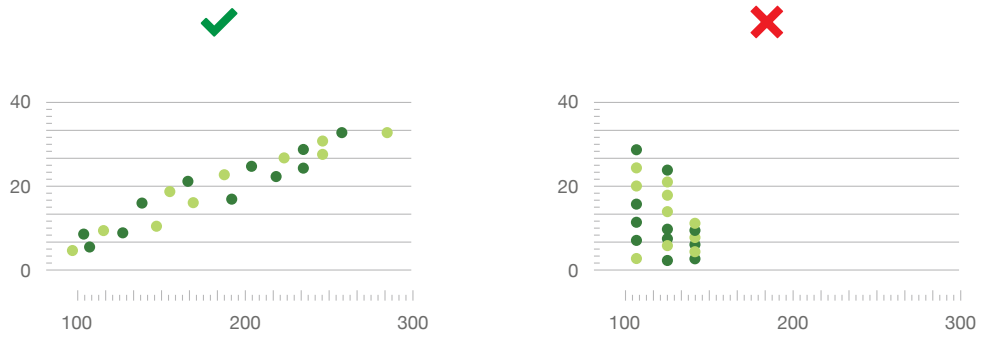


Fig. 1 Too few values along one axis can cause overplotting

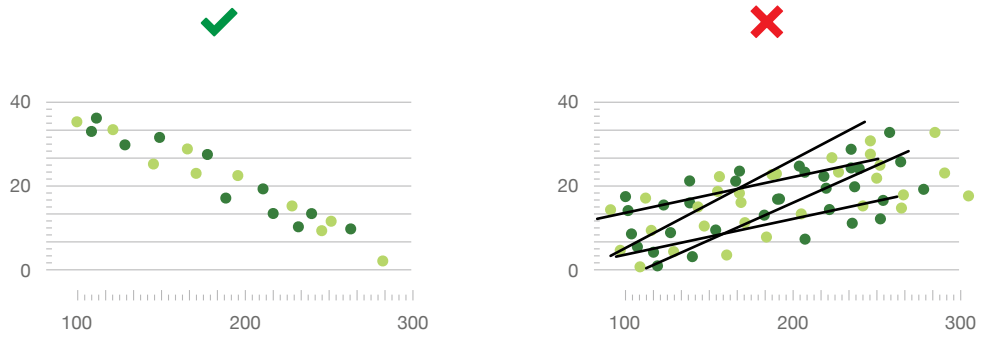


Fig. 2 Avoid using more than one trend line

Scatter Plot Graph

Variations

Scatter graphs can have many different trend styles based from the data they are mapping. Trends can arch upwards (positive correlation), downwards (negative correlation, or even have a cluster of data outline of a main trend outliers).

Various graph names include: Scatter point, scattergram, scatter diagram, scatter graph.

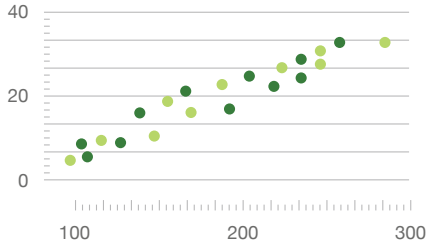


Fig. 1 Positive correlation

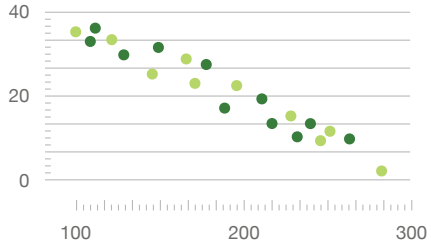


Fig. 2 Negative correlation

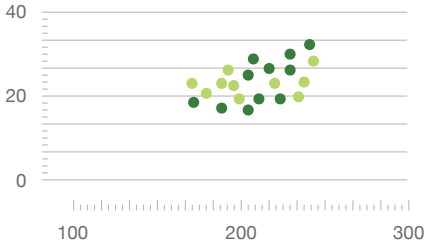


Fig. 3 Null

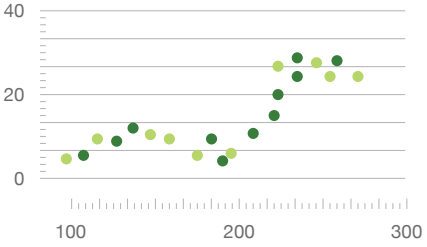


Fig. 4 Curvilinear

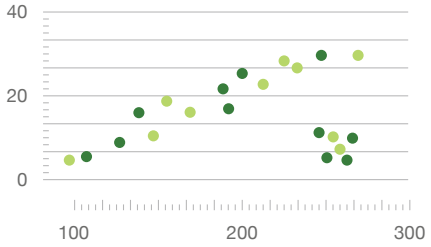


Fig. 5 Outliers

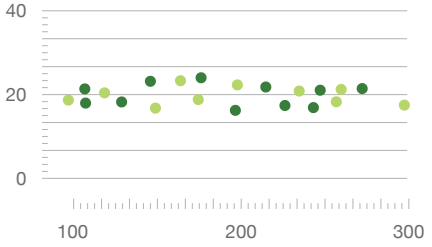


Fig. 6 Linear

Choropleth

Visualizing Geographical Trends

It is important to be careful when using choropleth. Hues and colors should be easily identifiable. It is important to remember the human eye is limited in the range it can distinguish, as well as disabilities such as color blindness.

A map generator that works with excel and javascript http://patrickgarvin.com/maps/USmap.htm

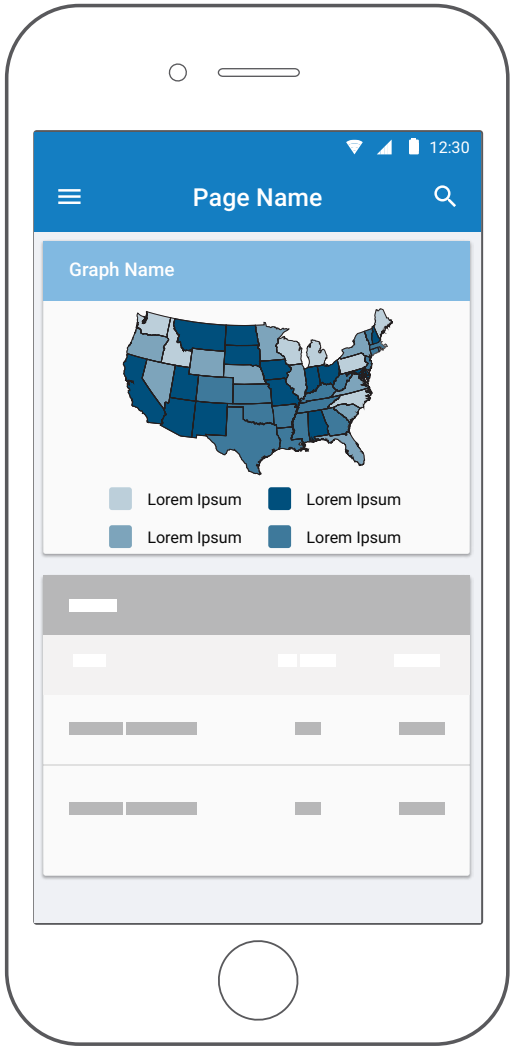


Fig. 1 Mobile example

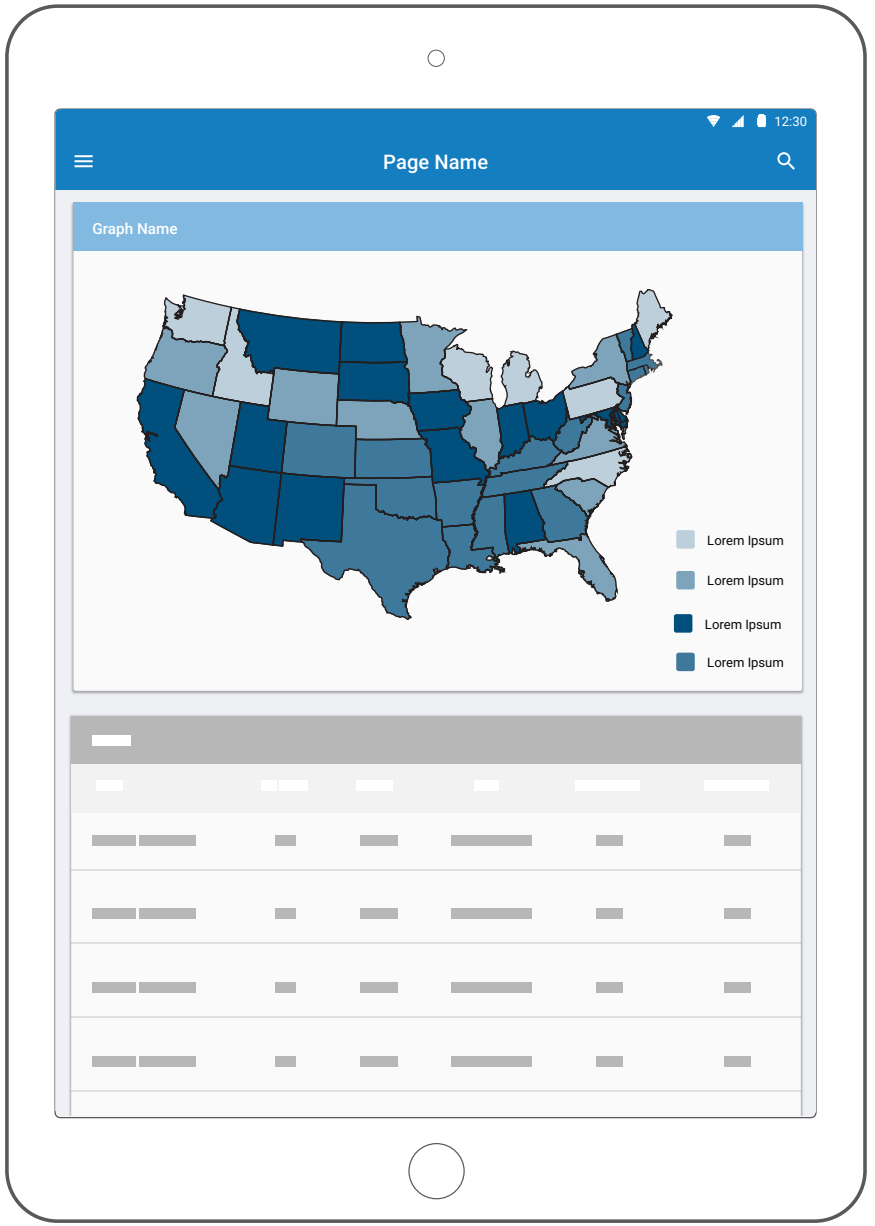


Fig. 2 Tablet example

Choropleth

Best Practice

DO

- ✓ Increase legend values from left to right, top to bottom.
- ✓ Consider a smaller date set if the level of variation is too great within the listed data items of a set.

DO NOT

- ✖ Use a random range of colors for quantities.
- ✖ Use Geographic data that is continuous in nature.
- ✖ Use to show variability within a region.

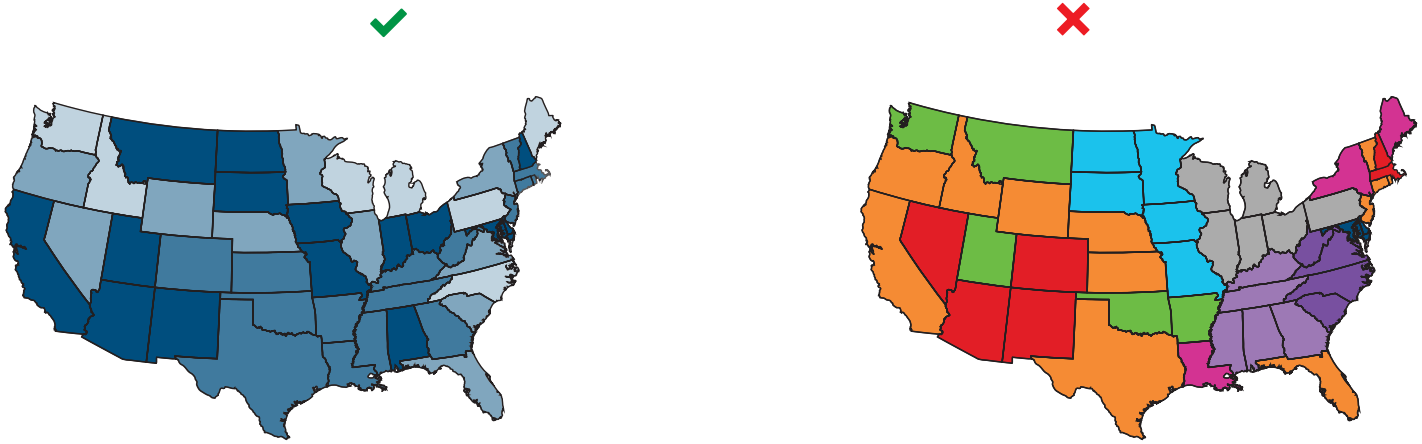


Fig. 1 Avoid using too many bright and unrelated colors

Choropleth

Variations

Various graph names include: Heat maps, area or shaded mapping, enumeration mapping

This is not the same thing as an isoline graph. An isoline graph maps continuous lines joining points of the same value, such as an elevation map.

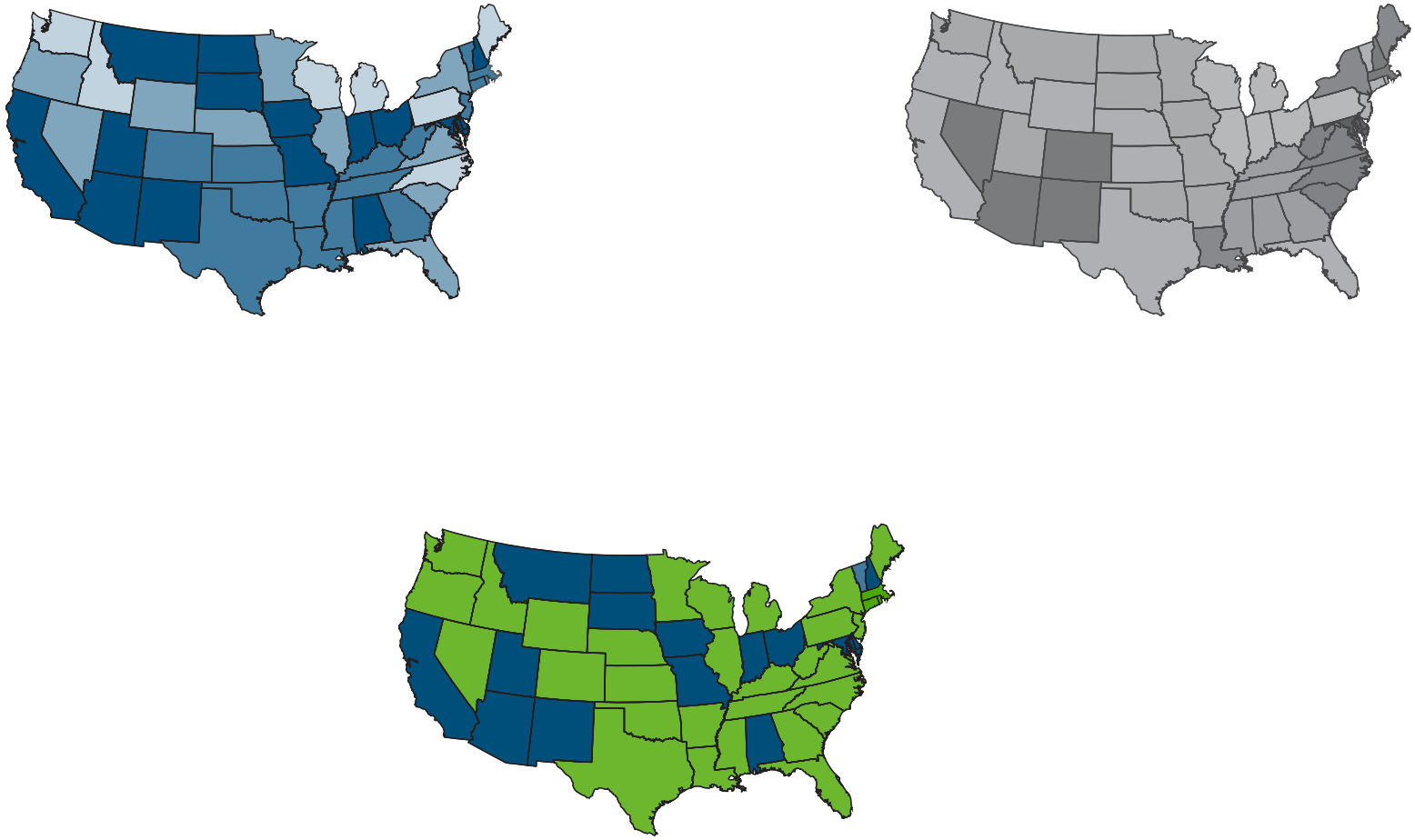


Fig. 1 Different examples of color progressions for choropleth

Dashboard

Data Visualization

Each graph has its own strengths and weaknesses that can come together to create a powerful interface that can be used to quickly assess and take action on. Dashboards should organize information to support meaningful use in an aesthetically pleasing and consistent way for quick and accurate use. A dashboard's makeup can vary by the triggers that compel a user to access it. Some best practices include:

- Choosing metrics that contribute to the user's needs. Focus is critical.
- Take advantage of white space; clutter distracts the eye.
- Keep it visual. Dashboards are meant to be read quickly. Text-based reports are not fast or easily read, and are often difficult to access.

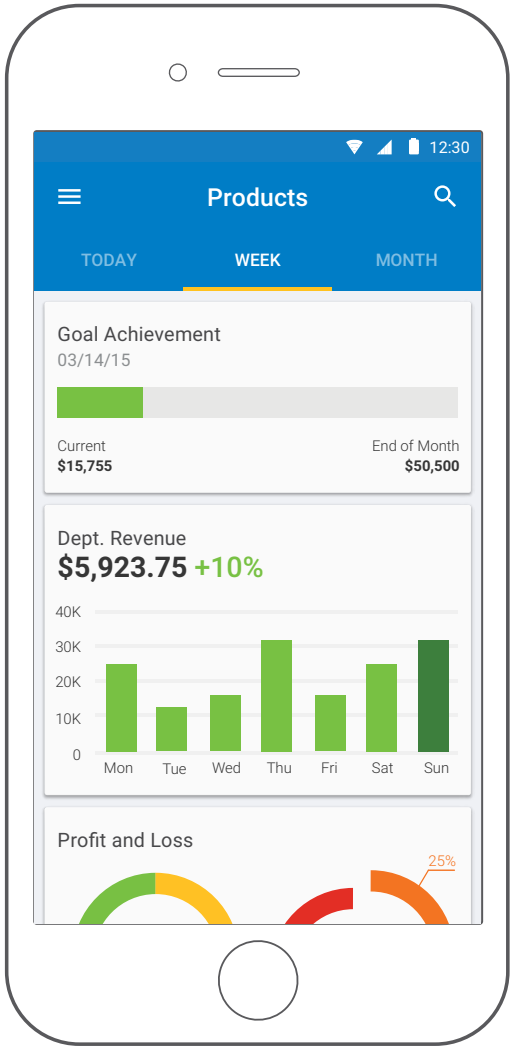


Fig. 1 Mobile example

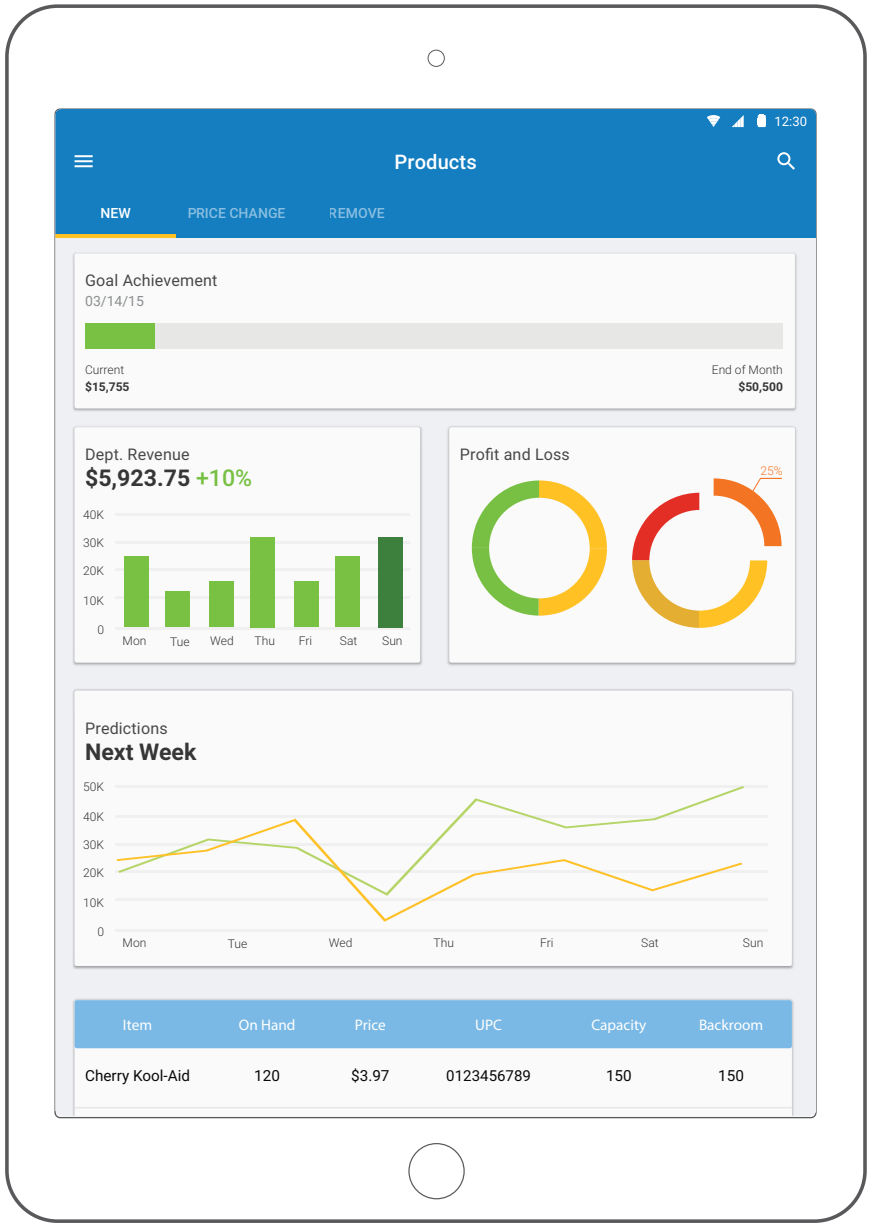


Fig. 2 Tablet example

References

BRAND IDENTITY

Branding

Official Online Walmart Brand Center: http://www.walmartbrandcenter.com

Color

Official Online Walmart Brand Center: http://www.walmartbrandcenter.com

Web Content Accessibility Guidelines: http://www.w3.org/WAI/intro/wcag

Test contrast ratios: http://leaverou.github.io/contrast-ratio

Typography

Translating px to ems, ems to px: http://pxtoem.com

Roboto Font Family Download: https://www.google.com/fonts/specimen/Roboto

Iconography

Material Design Font Family Download: https://www.google.com/design/icons

LAYOUTS

Devices and pages

iOS

iPhone 6 Resolutions: http://iphone6plusresolution.com

Guide to iPhone Resolutions: http://www.paintcodeapp.com/news/ultimate-guide-to-iphone-resolutions

iOS Developers: https://developer.apple.com/library/ios documentation/2DDrawing/Conceptual/DrawingPrintingiOS/GraphicsDrawingOverview/GraphicsDrawingOverview.html

Android

Android Convert It: http://androidpixels.net

Android Developers: https://developer.android.com/about/dashboards/index.html

Responsive Page Elements

Code Snippets for Mobile Web: https://developers.google.com/web/fundamentals/resources/samples/?hl=en

Responsive Patterns: https://bradfrost.github.io/this-is-responsive/patterns.html

Responsive Grids

Media Queries: https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

Calculating columns: gridulator.com

Managing Columns, Gutters, and Margins: guideguide.me

Px to Em Conversion: pxtoem.com

INTERACTIONS

Buttons

jQuery animations here: http://www.jqueryscript.net/demo/Form-Submit-Buttons-with-Built-in-Loading-Indicators-Ladda

CONTENT

Tables

Related links: http://gergeo.se/RWD-Table-Patterns/#demo

Cards

More Relating to Cards: www.website.com/resources

Images & Videos

More On Delivering Image Content: http://www.smashingmagazine.com/2014/05responsive-images-done-right-guide-picture-srcset/

https://html.spec.whatwg.org/multipage/embedded-content.html#embedded-content

MESSAGING

Notifications

iOS: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual

Mobile: HIG/NotificationCenter.html#//apple_ref/doc/uid/TP40006556-CH39-SW1

Android: http://developer.android.com/guide/topics/ui/notifiers/notifications.html

DATA VISUALIZATION

Chlorograph

A map generator that works with excel and JavaScript: http://patrickgarvin.com/maps/USmap.htm

Index

L

M

N

P

R

S

T

V

Walmart Retail Technology

Questions or Inquiries
Store Systems Mobile Development
STORESYS52@email.wal-mart.com