
REDLINE
DESIGN SYSTEM

Redline is a **living-breathing** design system. It's purpose is to be the smallest set of options that allow us to design everything we need.

Principles and Guidelines

The principles, design language, and best practices in this document will allow developers to focus on logic, while allowing the UX Team to focus on improving the user experience, interactions and workflows.

We strive to keep these guidelines top of mind as we make decisions. These principles are prioritized by importance.

> *It's designers, engineers, product owners & the rest of the team sharing the responsibility to build a quality product.*

Diana Mounter, *Design Systems Lead, Github*

Role Based

Support users' needs by only presenting data associated with their role and goal.

Consistent & Predictive

Create familiar experiences by strengthening intuition and applying the same solution to the same problem.

Timely

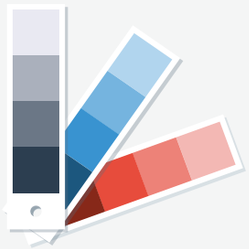
Support current needs for the users by displaying only relevant data as they need it.

Clean & Clear

Build lovable experiences by providing actionable results that create value for the users.

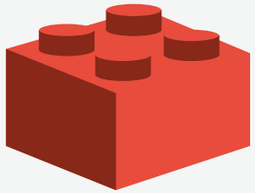
The parts and the whole

The Redline Design System is based on Atomic Design methodology. This modularity allows greater flexibility and consistency, while reducing costs and time to market.



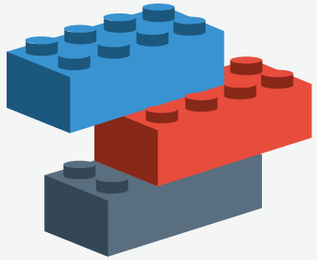
Theme

The theme is the ***basic styling*** of the application. It includes the colors, fonts, icons, and grid structure.



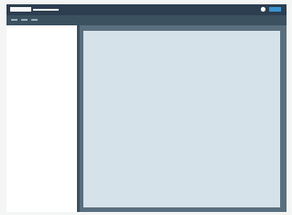
Components

Component are the ***building blocks*** of the application. Our design system utilizes Material Design's components.



Patterns

A pattern is a simple, reusable ***combination of multiple components*** that function together as a single unit.



Templates

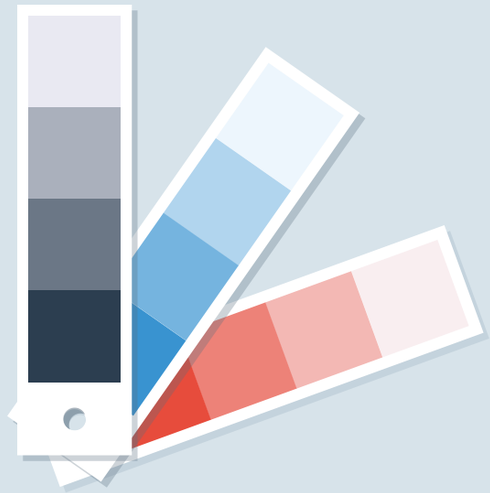
A template defines the ***hierarchy and structure*** of the page. The data and patterns needed dictate which is used.

Solutions

A good design solves a problem. A solution is the ***workflow*** created by combining components, patterns and a template.

http://bradfrost.com/blog/post/atomic-web-design/

https://material.io/guidelines/



Theme

Color Palette

System

.primary
#3993D0
(57, 147, 208, 1)

.secondary
#2C3E50
(44, 62, 80, 1)

.success
#7ABC43
(122, 188, 67, 1)

.caution
#F2B24F
(242, 178, 79, 1)

.danger
#E74C3C
(231, 76, 60, 1)

UI

.bluewood
#334755
(51, 71, 85, 1)

.shuttle
#596E7F
(89, 110, 127, 1)

.nepal
#99AFBF
(153, 175, 191, 1)

.bottecelli
#D7E3EA
(215, 227, 234, 1)

.polar
#EEF6FB
(238, 246, 251, 1)

Neutrals

.black
#000000
(0, 0, 0, 1)

.nevada
#656F76
(101, 111, 118, 1)

.silver
#AAAAAA
(170, 170, 170, 1)

.alto
#E0E0E0
(224, 224, 224, 1)

.alabaster
#FAFAFA
(250, 250, 250, 1)

.white
#FFFFFF
(255, 255, 255, 1)

.transparent
#FFFFFF
(255, 255, 255, 0)

Do not add custom colors to NGEN without consulting the UX Team for approval.

Font and Font Weights

Text is the primary way our users digest data. Help users complete their tasks by creating a clear visual hierarchy of the data.

Roboto *https://fonts.google.com/specimen/Roboto*

Aa Light (300)

Aa Regular (400)

Aa Medium (500)

Aa Bold (700)

Defaults for all text:
color: .secondary;
font-weight: 300;
line-height: 1.5;
margin: 0;
padding: 0;

Typeface styles

Heading Large: 1.5rem (24px)

The quick brown fox jumps over the lazy dog.

Subheads can be made by reducing the size and changing the text color to .nevada.

Heading Medium: 1.25rem (20px)

The quick brown fox jumps over the lazy dog.

Heading Small: 1rem (16px)

The quick brown fox jumps over the lazy dog.

Heading Label: 0.625rem (10px)

The quick brown fox jumps over the lazy dog.

Body: 0.75rem (12px)

The quick brown fox jumps over the lazy dog.

Body Small: 0.625rem (10px)

The quick brown fox jumps over the lazy dog.

Color Contrast *http://webaim.org/resources/contrastchecker/*

The Web Content Accessibility Guidelines (WCAG) recommends a threshold ratio of 4.5:1. Text colors used are Secondary, Nevada, and White.

Aa Aa Aa Bluewood

Aa Aa Aa Shuttle

Aa Aa Aa Nepal

Aa Aa Aa Bottecelli

Aa Aa Aa Polar

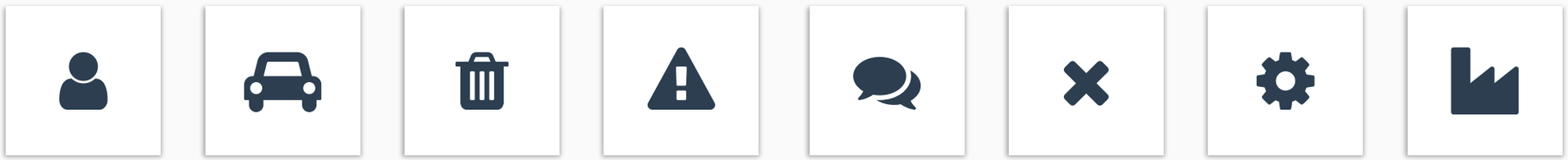
Iconography

Material *https://material.io/icons/*

Material icons are less cartoonish… for this reason, Material is preferred over Font Awesome.

Icons often cause usability problems when they are used without consideration… use a text label and don't rely on a hover for clarification.

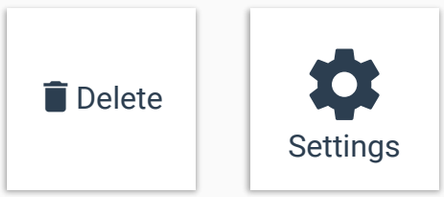
Font Awesome *http://fontawesome.io/icons/*



Font Awesome icons should be used if a suitable Material icon isn't available.

Very few icons are universally recognizable by users. See the UX Team if you need help selecting an icon, or need one custom designed.

Label Placement *http://uxmyths.com/post/715009009/myth-icons-enhance-usability*



Label are placed to the right, or under the icon.

8-Point Grid (Margin and Padding)

Use multiples of 8 to define dimensions, padding, and margin of both block and inline elements. When all of your measurements follow the same rules, you automatically get a more consistent UI. By removing 7 of every 8 spacing options, it allows the developer to eyeball an 8pt increment instead of having to measure each time.

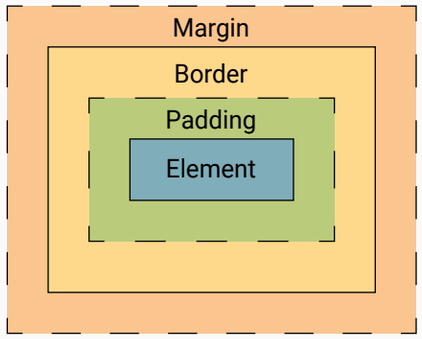
The Box Model

The Box Model is a way to describe an object's dimensions and spacing. It consists of 4 components: border, margin, padding, and the dimensions of the element itself.

Border: the thickness of the stroke around the edges of an element.

Padding: the space between the bounds of an element and its child elements

Margin: the space between the bounds of an element and neighboring objects



Naming

Class	px	rem
none	0	0
x-small	8	.5
small	16	1
medium	24	1.5
large	32	2
x-large	40	2.5
xx-large	48	3

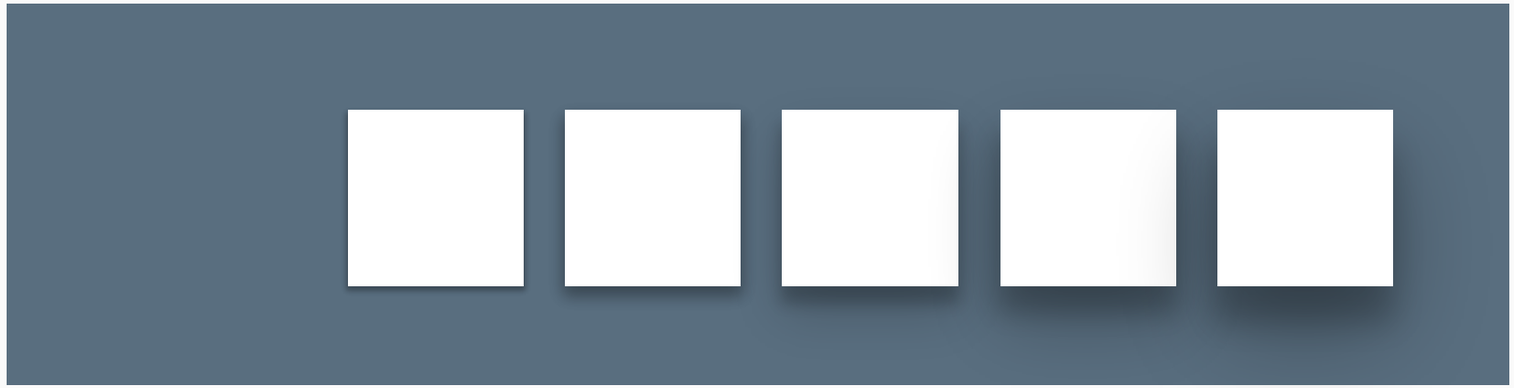
Use these classes to properly size and position the components and patterns.

Do not hard-code margin/padding on components. Do not use numbers that break the 8-point grid.

https://spec.fm/specifics/8-pt-grid

Elevation and Shadows

Elevation provides important visual cues to users, helping them understand what actions are available. The higher an object's elevation, the softer and larger its' shadow becomes.



Elevation should be used to create visual hierarchy. Objects with higher elevations are more prominent and should hold the most important information.

Online version will be more robust with the addition of CSS.

https://material.io/guidelines/material-design/elevation-shadows.html#
https://material-components-web.appspot.com/elevation.html

Motion

The material environment draws inspiration from real-world forces, such as gravity and friction. Incorporating motion improves usability and provides personality by connecting different states and enhancing affordances.

Motion should be used to provide feedback and visibility into the system status.

An animation should never impede the user to interact with the UI.

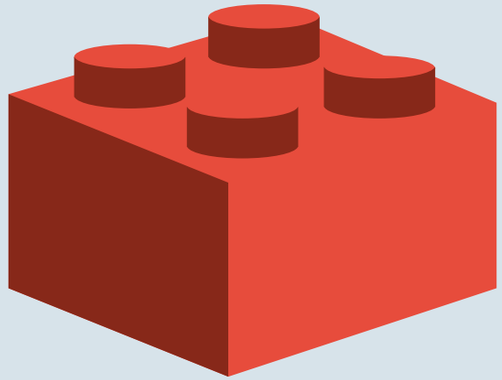
Duration

Online version will be more robust with the addition of CSS examples.

Easing

Online version will be more robust with the addition of CSS examples.

https://material.io/guidelines/motion/material-motion.html#



Components

Badge

A badge passively indicates unread/unseen content.

Badges must be red and can only contain an integer. Badges are used when new/unread information is available for the user (comments, notes, etc). Update the integer as soon as the important content is viewed.

Don't send multiple notifications for the same thing. Badges are designed to be passive, and should not be used for critical information. Badge ≠ Count.

Buttons

Buttons communicate what actions are available to the user.

Raised Buttons

DEFAULT SECONDARY SUCCESS DANGER DISABLED

This button is used to add dimension and emphasize important functions. Raised button representing the primary actions, and should be placed on the right of any secondary buttons.

Don't use icons in buttons. The text and color of the button should reinforce the action being taken, so don't rely on color alone.

Flat Buttons

DEFAULT SECONDARY SUCCESS DANGER DISABLED

This button is used for general functions and reduce the amount of layering on the screen, making it more readable.

Never use more than two secondary buttons. If three or more secondary actions are available, consult the UX Team.

Floating Action Button (FAB)

Very few screens warrant a floating action button. FABs are only allowed to be placed in the bottom-right corner of the stage, and represent the primary application-wide action.

Don't use icons that make the user interact with it to figure out what it does. No screen is permitted multiple FABs.

https://material.io/guidelines/components/buttons.html

Chips

A chip is a small block of supporting data such as a avatar, text or a status. Chips are placed to the right of the data it supports.

Default Primary Success Danger Disabled

Chips should only represent one chunk of data, but can be used in groups. Color can be used to help convey the data (ie. green for a good status, red for bad) but should be used sparingly.

If the data isn't supporting the data directly to its left, a different component is needed. Avoid long, run-on text. Inactive Chips can only be used if the supporting data is inactive.

Deleteable Chips / Avatar Chips

Primary DE Default  Jarrod Murray

Deleteable Chips should only be used when the user added the Chip to the interface. Avatar Chips are acceptable when referring to another NGEN user.

Never allow users to remove an element from the UI without a way to add it back or undo the action.

https://material.io/guidelines/components/chips.html

Text Field

Text Fields allow users to input text and usually appear in forms. Users may enter text, numbers, or mixed-format types of input.

VIN

VIN
1G1YY3D70H5112910

VIN
1G1YY3D70H5112910

VIN
1G1YY3D70H5112910
Error Message

VIN
1G1YY3D70H5112910

Label must be descriptive and short. Use good defaults, and auto-complete when at all possible.

Avoid really long or wrapping labels. Don't repeat section headers with the same label.

Clear Input

VIN
1G1YY3D70H5112910

VIN
1G1YY

Displays only after characters have been entered.

Only use on fields where it makes sense for the user to empty the input field.

Range Inputs

Minimum to Maximum

The minimum is on the left, maximum on right.

Search Input

VIN
1G1YY3D70H5112910

Only use when searching on a specific field.

Don't use for large, app-wide searches… it lacks the affordance of a large-scale search.

https://material.io/guidelines/components/text-fields.html#

Displaying Data

How data is displayed greatly impacts the users ability to accomplish their task. Be mindful of what task the user is trying to accomplish when displaying data.

Vertical

VIN

1G1YY3D70H5112910

The default display of data pairs is vertically. Use short, descriptive labels.

Don't mix displays and text fields because of their similar styling. Avoid clustering too many displays.

Horizontal

VIN	1G1YY3D70H5112910
VIN	1G1YY3D70H5112910

A zebra-striped display can be used if the data is easier to digest/compare. Inline edits are displayed as links.

Table

Header	Left align text	Right align numbers	Resize columns to data
Value	Value	5	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas hendrerit lectus nec.
Value	Value	55	How to truncate if necessary. Lorem ipsum dolor sit amet, consectetur adipiscing elit. …
Value	Value	555	Value

For tabular data only. Use the default Material table but add zebra stripping.

Don't create custom sorts or filters for tables… use the Material defaults when needed.

https://material.io/guidelines/components/data-tables.html

https://material-ui-1dab0.firebaseapp.com/demos/tables/

Date Picker

A control used for selecting a single date.

- Default picker is Inline Container with AutoOK.
- Don't include the word "Date" in the label. The calendar icon and date (when filled in) are adequate affordance.

Floored

Floored

02/18/2017

2017
Fri, Feb 18

February 2017

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

CANCEL

https://material.io/guidelines/components/pickers.html#pickers-date-pickers

Expansion Panels (Accordion)

Expansion Panels allow content to be placed within expandable sections.

Collapsed Secondary Info

Collapsed Secondary Info

Collapsed Secondary Info

Collapsed Secondary Info

Expanded Secondary Info

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nulla lectus, tempor id gravida ve.

OPTIONAL

Collapsed Secondary Info

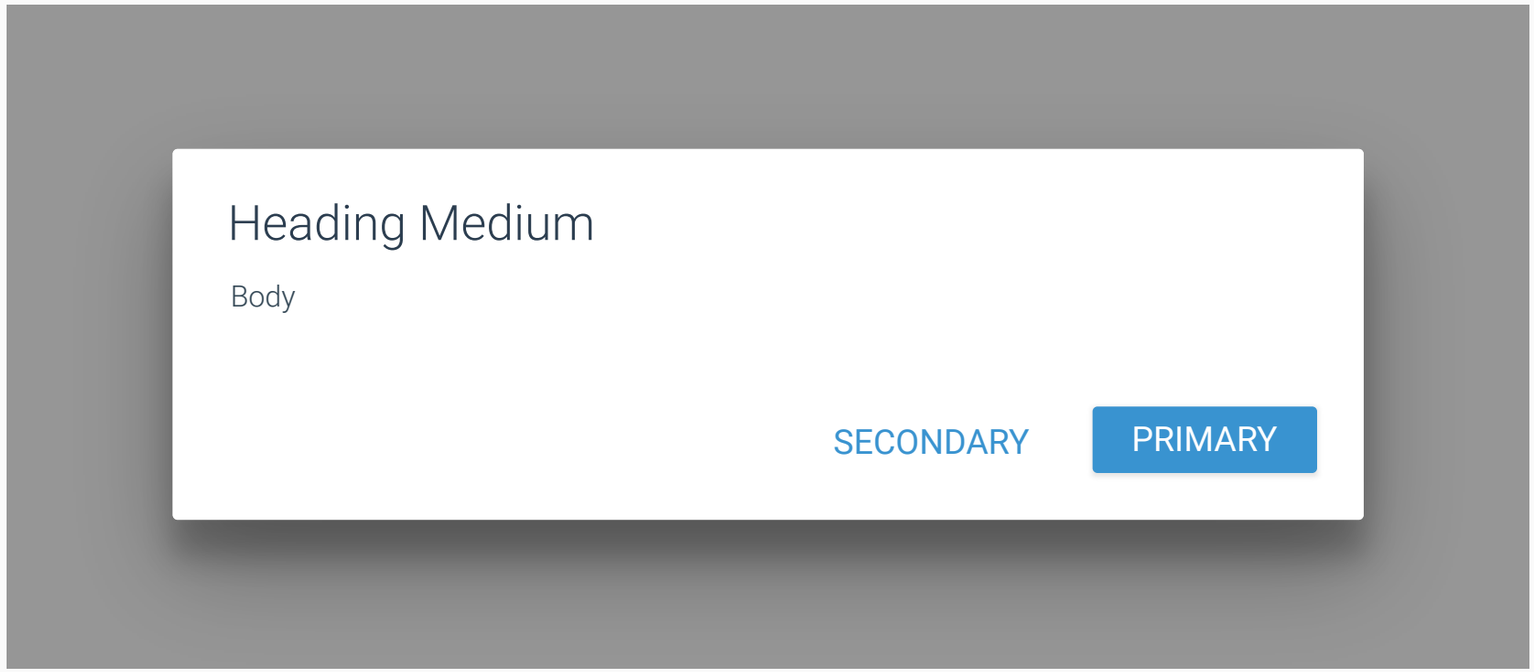
A collapsed panel displays summary information of the data it contains. Expanded panels, like cards, are a blank canvas. They can contain a variety of data.

Never hide pertinent information. Give the user what they need and allow them to get more (if requested).

https://material.io/guidelines/components/expansion-panels.html#

Modal (Dialog)

Modals contain text and UI controls focused on a specific task. They inform users about critical information, require users to make decisions, or involve multiple tasks. Below is the default styling of a modal....more examples can be found in the pattern library.



Use dialogs sparingly because they are interruptive. Not every choice, setting, or detail warrants interruption.

Don't open a modal from within a modal. Avoid scrolling in modals.

https://material.io/guidelines/components/dialogs.html

Notifications

Notifications provide brief feedback about an operation through a message at the bottom of the viewport. Notifications can contain a single action.

Default

Body .white

The default notification should contain a single line of text directly related to the operation performed.

Don’t stack notifications or use icons.

Color and Action

Body .white OPTIONAL ACTION

Success/Danger can be used to convey the result of an operation (example: “Save Successful”). A single action button can be included (if needed to complete the operation).

Notifications can only be Secondary, Success, or Danger… don’t use any other color.

Title

Heading Small .white
Body .white
OPTIONAL ACTION

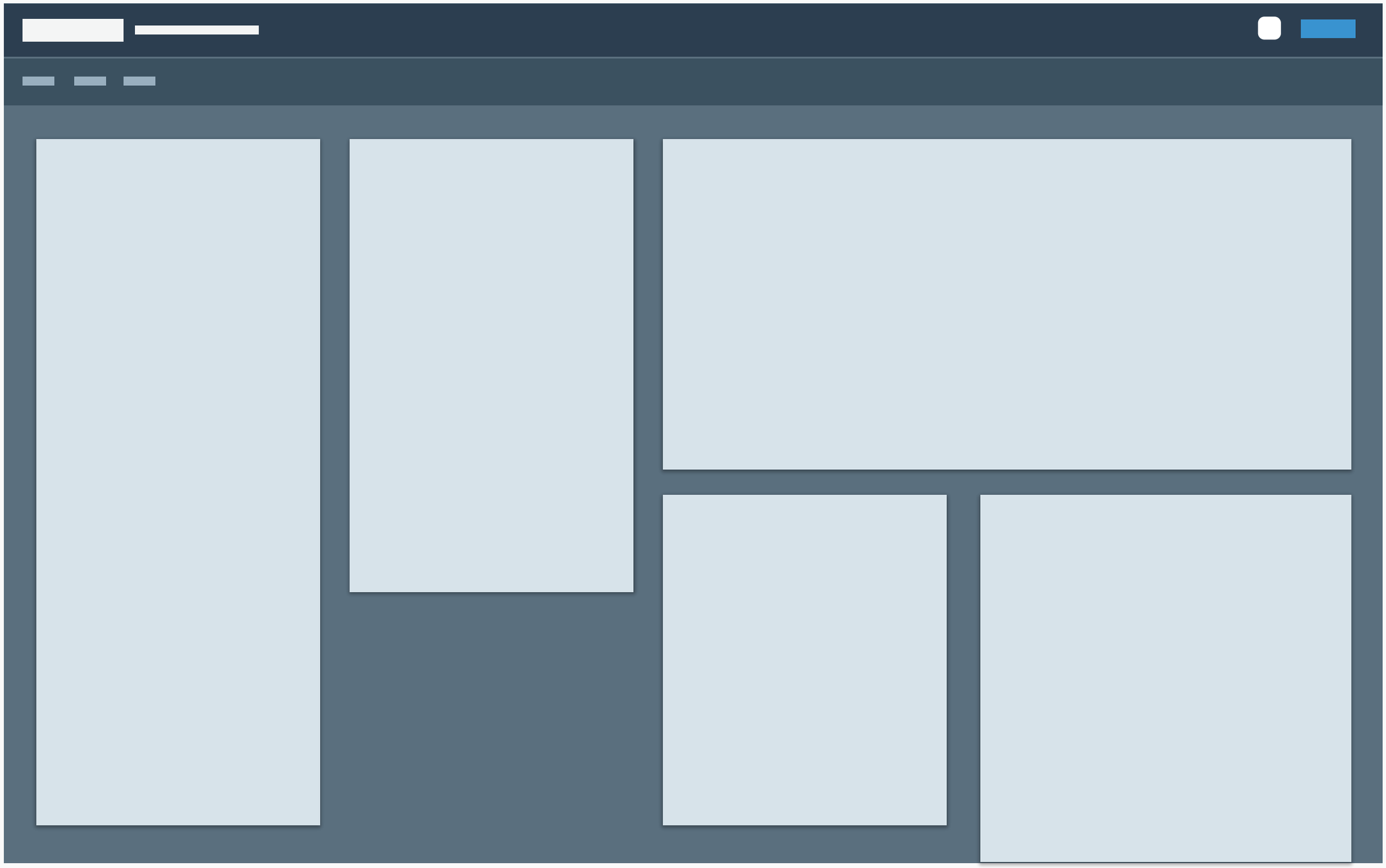
A title can be used for further clarification, when a simple description isn’t enough.

Don’t repeat the title in the body.

https://material.io/guidelines/components/snackbars-toasts.html

Boards

A board is a sheet of material that serves as an entry point to more detailed information. Boards are essentially a div used to group data into logical chunks and to create visual hierarchy on the stage (work area).



Boards are always the same color (.bottecelli) and elevation (2dp). Boards can be tabbed when the data and/or workflow necessitate it. Board layouts are created using Flexbox and should never overflow-x (causing horizontal scrolling).

Boards can not overlap because they are on the same elevation.

https://material.io/guidelines/components/cards.html#

Select

A single-option picklist menu.

Label

– Select –

Label

Option C

Label

Option C

- Option A
- Option B
- Option C
- Option D

Multi-select

A multi-option picklist menu.

Label

Option C, Option D

Label

Option C, Option D, …

- Option A
- Option B
- Option C
- Option D
- Option E
- Option F
- Option G
- Option H

Follow the guidelines set forth in the Text Field section.

Don’t allow the window to overflow the stage… the entire window should be visible.

https://material.io/guidelines/components/text-fields.html#
https://material-components-web.appspot.com/select.html

Selection Controls

Selection controls allow the user to select options.

Checkbox

- [ ] Off
- [x] On
- [ ] Disabled Off
- [x] Disabled On

For selecting multiple options from one set.

Not to be used as a single on/off switch. Use a toggle in such cases.

Radio Buttons

- ○ Off
- ◉ On
- ○ Disabled Off
- ◉ Disabled On

For selecting a single option from a set. Default to the most likely selection (when possible).

Avoid long lists of options. After five options, consider a dropdown.

Toggle

- Off
- On

Use when a single settings is either True/False.

https://material.io/guidelines/components/selection-controls.html#

Tiles

A tile is a sheet of material that serves as an entry point to more detailed information. Below is the default styling of a tile....more examples can be found in the pattern library.

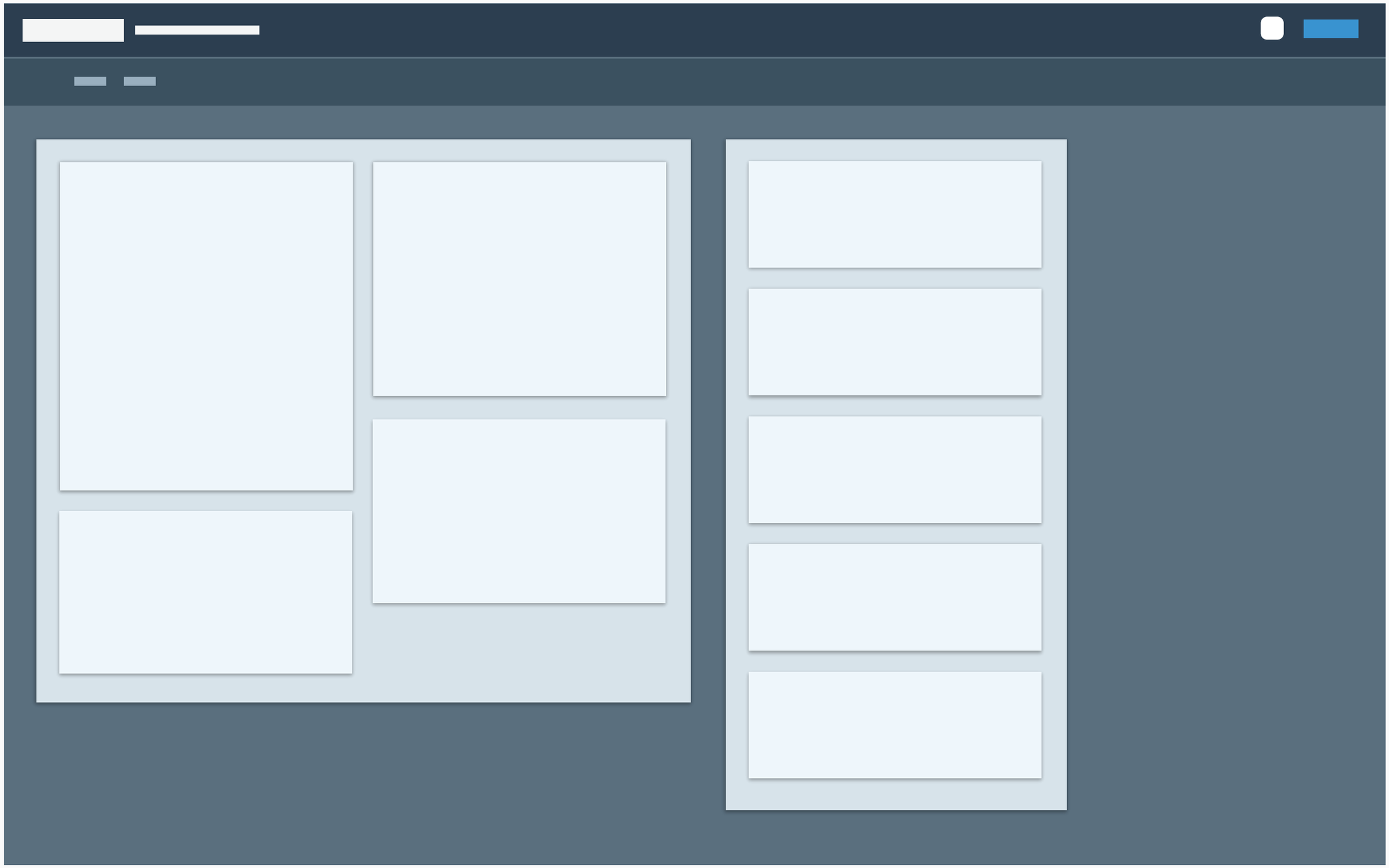
Tiles come in many shapes and sizes. Tiles are composed of different content blocks, which are typically laid out in vertical succession.

Tiles provide context and an entry point to more robust information and views. Don't overload cards with extraneous information or actions. Inline links within text content are strongly discouraged.

https://material.io/guidelines/components/cards.html

Cards

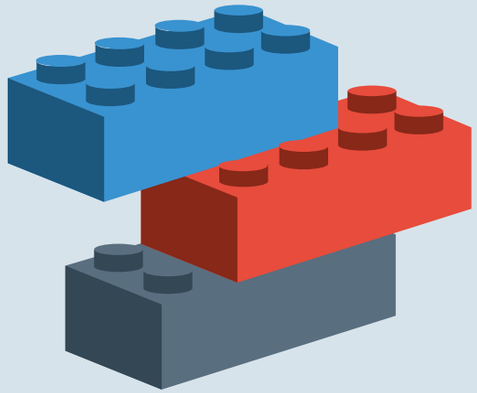
In essense, cards are just nested boards. Cards allow for another level of grouping data into chunks. Cards are only permitted inside a board.



Cards are always the same color (.polar) and elevation (2dp). Cards are never tabbed. Card layouts are created using Flexbox and should never overflow-x (causing horizontal scrolling) the board containing it.

Cards are not to be used as a design element… they are only used to group data within a board.

https://material.io/guidelines/components/cards.html#



Patterns

System Nav

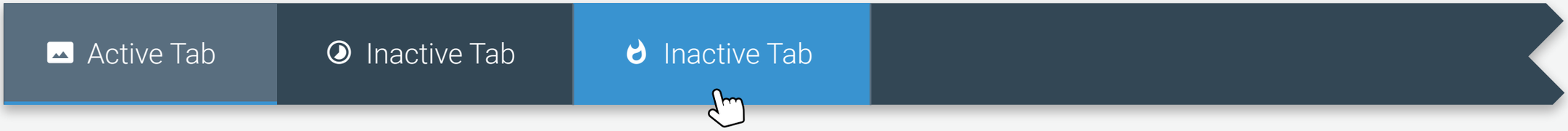
The System Nav displays the current application and user, as well as the ability to switch between NGEN apps. This pattern is required in all NGEN applications.

The System Nav affixes to the top of the viewport and is always accessible.

Do not add any other components (buttons, alerts, etc.) to the System Nav.

App Nav

The App Nav provides navigation within a single app.



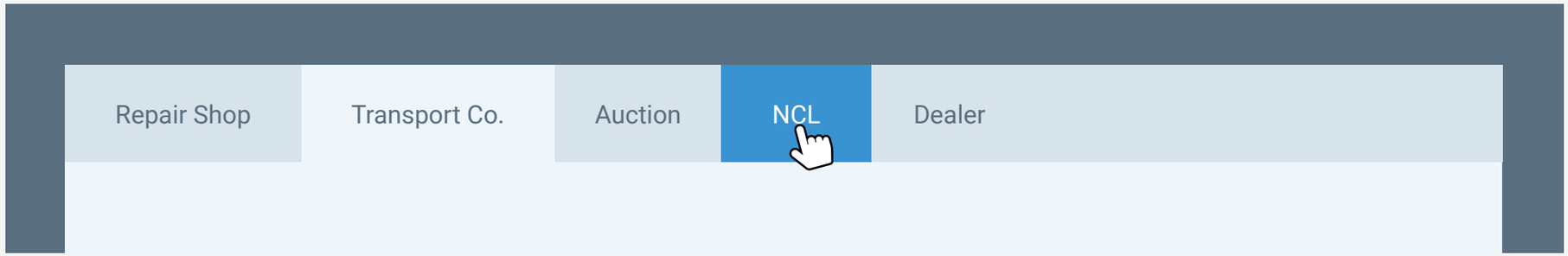
The App Nav is affixed to the System Nav and should always be accessible. The App Nav is not required if/when an app is a single page.

The App Nav is reserved for app-wide navigation and search. No components should be added to it.

Sub Nav

Splitting a panel into horizontal tabs is acceptable when the workflow and data necessitate it.

Text Labels

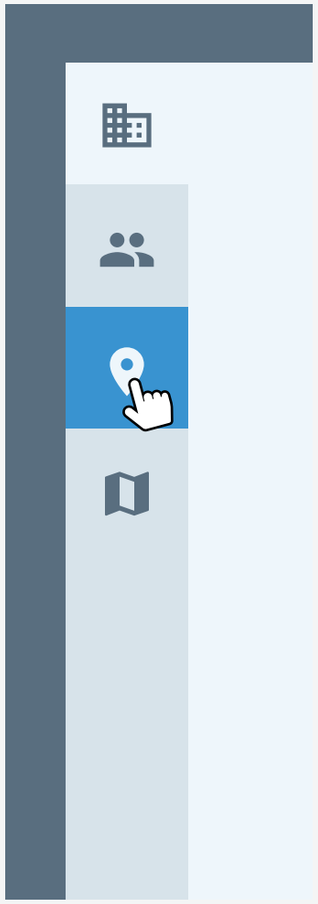


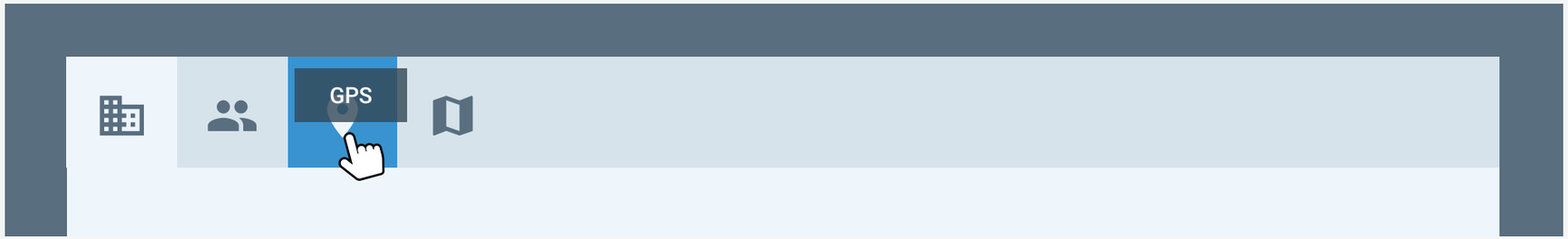
The active tab and panel should be the lightest color, giving the illustion of being higher in elevation.

If you have Information Architecture (IA) questions please consult with UX.

Icon Sub Nav

Splitting a panel with icon tabs is acceptable when the workflow and data necessitate it. Follow the guidelines set in the iconography section for icon best practices.





The active tab and panel should be the lightest color, giving the illustion of being higher in elevation. Icon Sub Navs must include a Tooltip for learnability.

Before simply adding a new tab, consider if the new data can be presented in a better fashion for this workflow. If you have Information Architecture (IA) questions please consult with UX.

Jumbotron

A jumbotron is a full-width board which contains the most pertinent data and actions for any given screen. Jumbotrons are placed directly under the app nav. All data displayed below the jumbotron should directly pertain to it.

New To You Auto Sales 101345 Locked

555-555-5555 | DevTest@nextgearcapital.com

3335 N. Main Ter Ste B | Gainesville, FL 32609-2301

SECONDARY PRIMARY

Finace Type	Dealer Segment	Line(s) of Credit	Total Balance
Core	Key	Wholesale	$321,517.06

Jumbotrons are split into two vertical rows. The upper row contains the “what am I looking at?” and “what actions can I perform here?” information. The lower row holds (Flexbox) columns of data… it should be reserved for the most pertinent data.

The jumbotron is not a catch-all for information. Data displayed inside the jumbotron should not be duplicated in the UI below. The upper row should remain “sticky” to the screen, so users know what they are viewing/taking action on.

Collapsing Jumbotron

New To You Auto Sales 101345 Locked

555-555-5555 | DevTest@nextgearcapital.com

3335 N. Main Ter Ste B | Gainesville, FL 32609-2301

SECONDARY PRIMARY

If there is a lot of data below the jumbotron, the jumbotron can collapse down to only display the upper row. The jumbotron remains visible on the screen at all times. When the user scrolls up, the jumbotron will re-open.

Buttons in the jumbotron are actions performed on a screen level… not on work units below the jumbotron. Never use more than three buttons in the jumbotron without consulting the UX team.

Empty State

An empty state, or zero-data state, notifies users when an item's content can't be shown.

No vehicles

There are currently no vehicles to reconcile

Business info

Please select a business

Great job!

Your queue is empty. Look busy as I find you work.

Think of this empty state as a mini landing page. While minimal in design, a successful empty state will explain a specific feature and then compel the user to take the next step.

Empty states are not to be used for system errors.

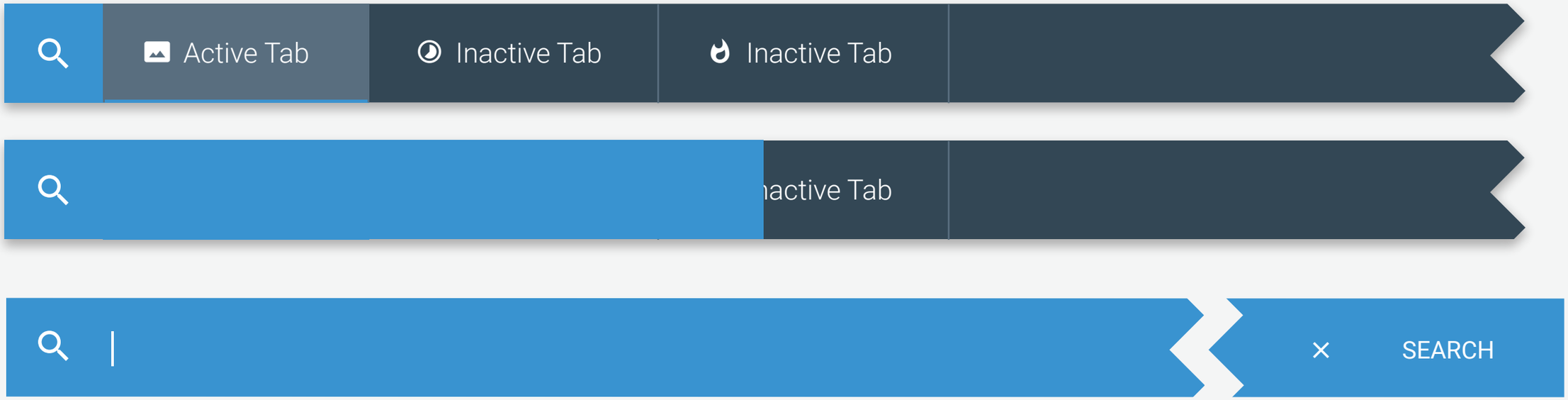
Progress and Loading

Progress and activity indicators are visual indications of an app loading content.

Use Material's Circular indeterminate as the default progress indicator.

Search

A search in the App Nav denotes the search is being performed app-wide (across all tabs).

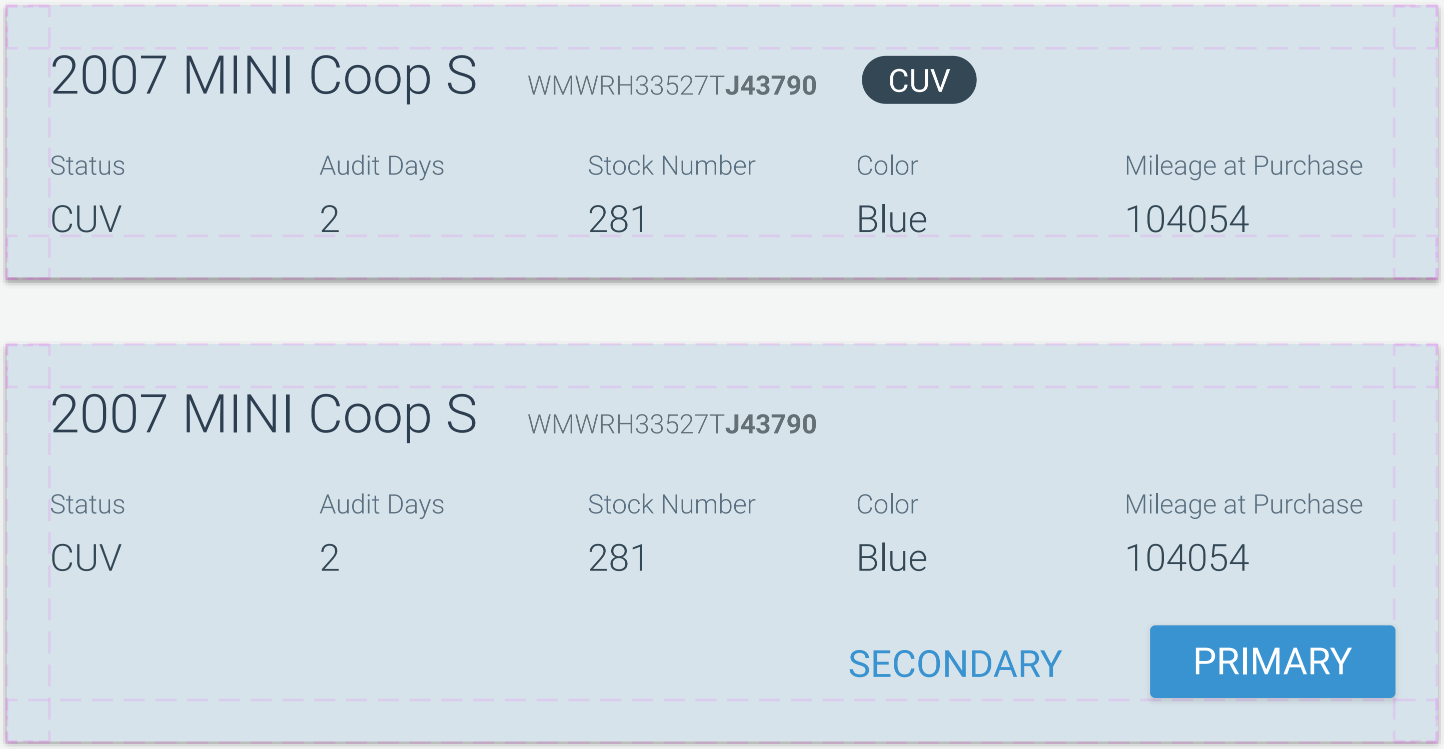


Clicking the Search Box will activate the search bar, which grows across the App Nav.

The App Nav is reserved for app-wide navigation and search. No components should be added to it.

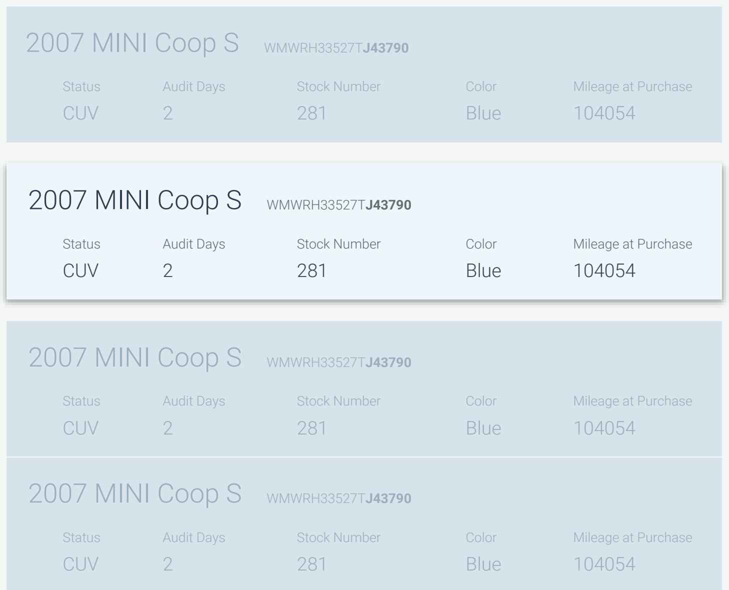
Advanced Tiles

Tiles are a more visually appealing alternative to data grids. Typically, users interact with a tile to get additional information.



Stacking Tiles

Tiles stack vertically on a single sheet of material.



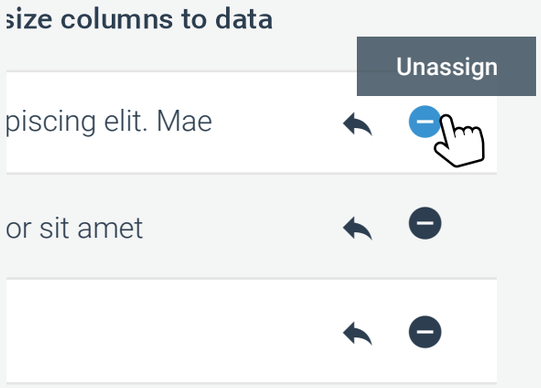
Active Tiles separate from the list with top and bottom margin. It retains the elevation of the list, as the rest of the tiles flatten and gray out.

Cards provide context and an entry point to more robust information and views. Don't overload cards with extraneous information or actions. Inline links within text content are strongly discouraged.

Advanced Tables

Icons/Actions

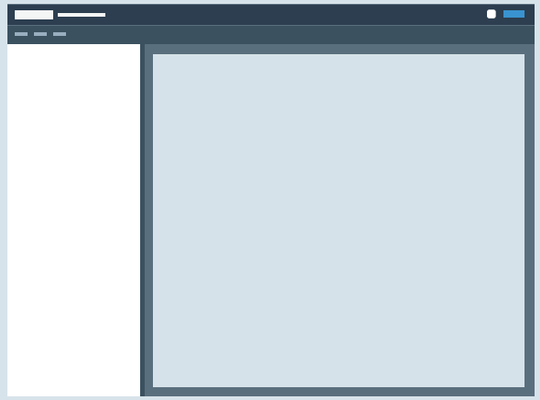
Tiles stack vertically on a single sheet of material, and should be zebra-striped for increased scanability.



All icons must have a hover state, as well as tooltip, that clearly defines what action is taking place.

Don't overload the user by adding too many icons to a row. Be cognizant of how the rows will stack and how the icons will display in a full table view.

https://material.io/guidelines/components/tooltips.html#



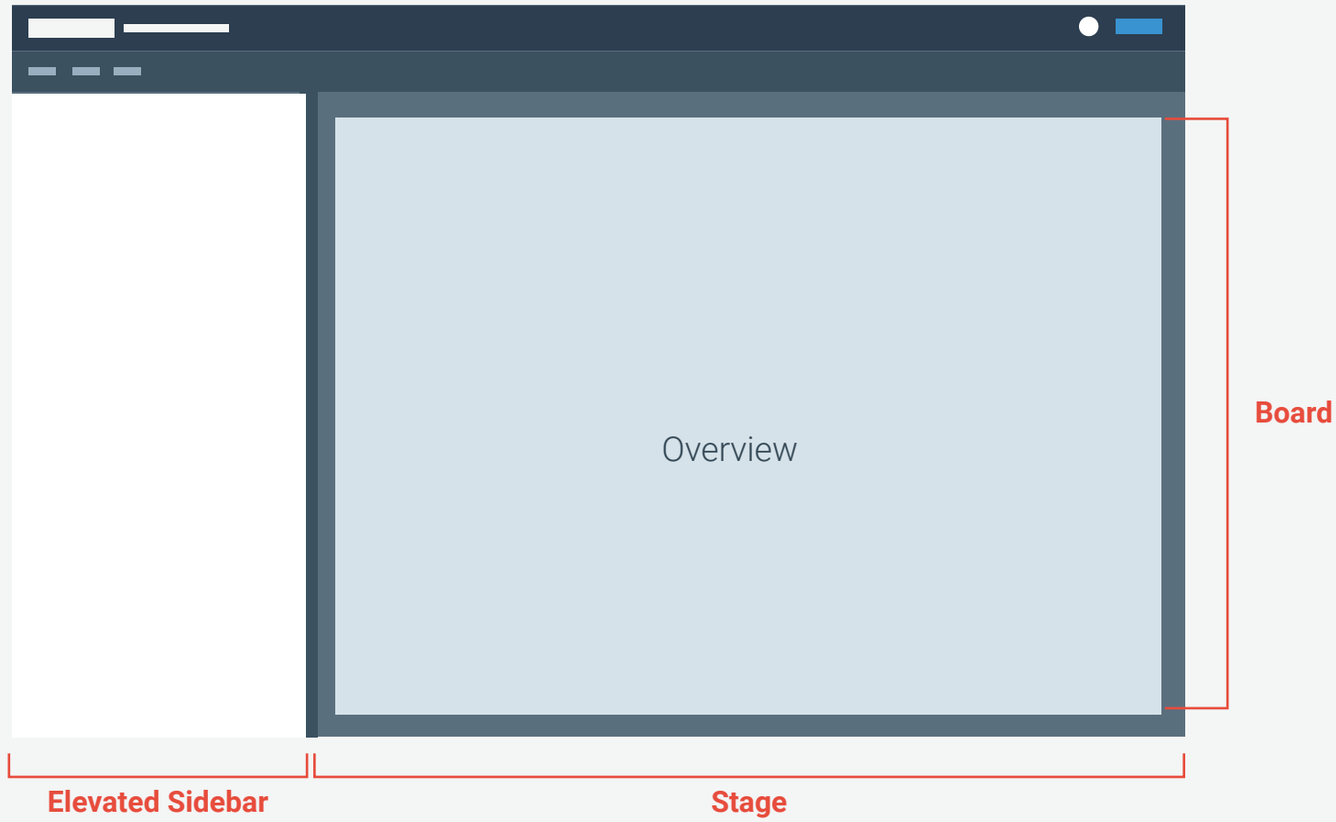
Templates

Overview

A template establishes the hierarchy and structure of the page by defining the size of the stage and number of boards displayed. There are two template types, **Elevated** and **Flat**, the one you use is determined by the content.

Be cognizant of information hierarchy when selecting a template

Do NOT create a new template without consulting with the UX Team.



Elevated

When to Use:
An elevated sidebar is used to establish hierarchy. Components and patterns displayed in the sidebar are considered parent objects. Selecting these components will dynamically display child data on the stage.

The sidebar can be (1/3) **33%**, (1/4) **25%** or (1/6) **16%** of the screen. The one you choose should be based on the size of the parent components.

Make sure the components/patterns follow the Material elevation principles.

Don't use more than 2 columns without consulting with the UX Team.

https://material.io/guidelines/material-design/elevation-shadows.html

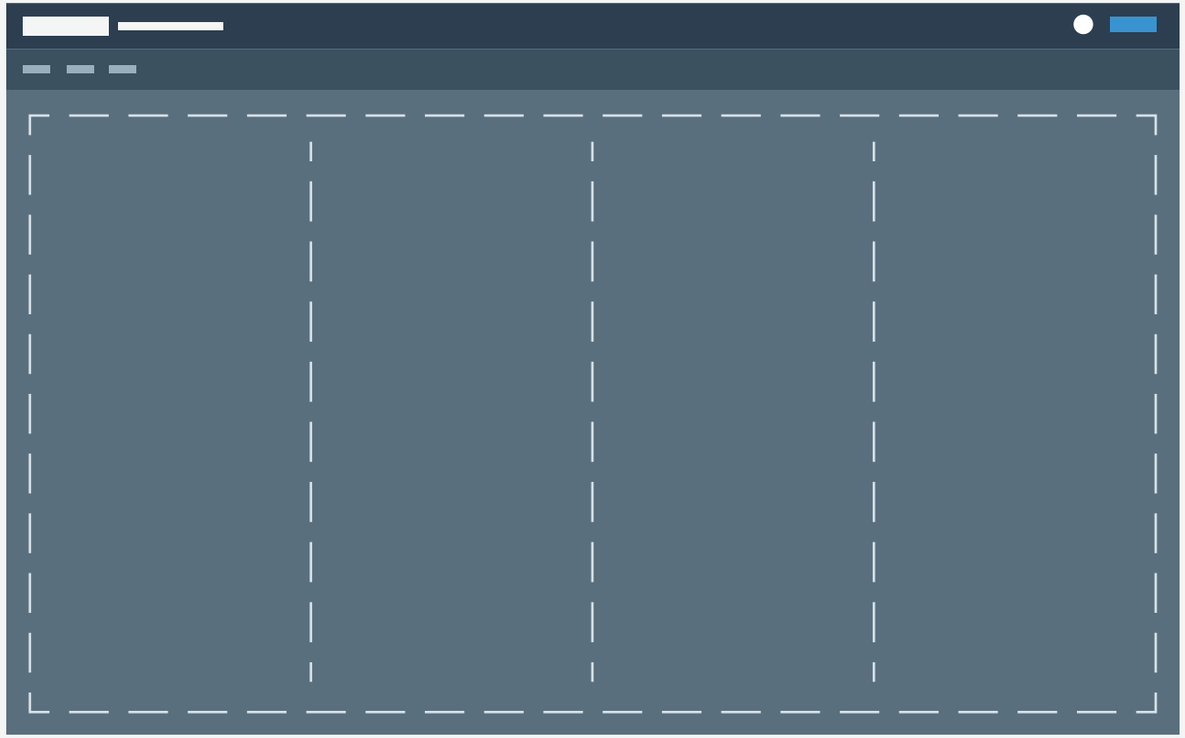
Flat

When to Use:
Use the flat template when the content is non-hierarchical. The data displayed should be considered sibilings to one another.

Rules:
The stage should be no more than 4 columns (before wrapping occurs)
The full stage remains visible in the browser window (no horizontal scrolling)

✓ Nesting flexbox columns within a column is acceptable, space permitting.

✗ NO horizontal scrolling... use flexbox/wrapping.

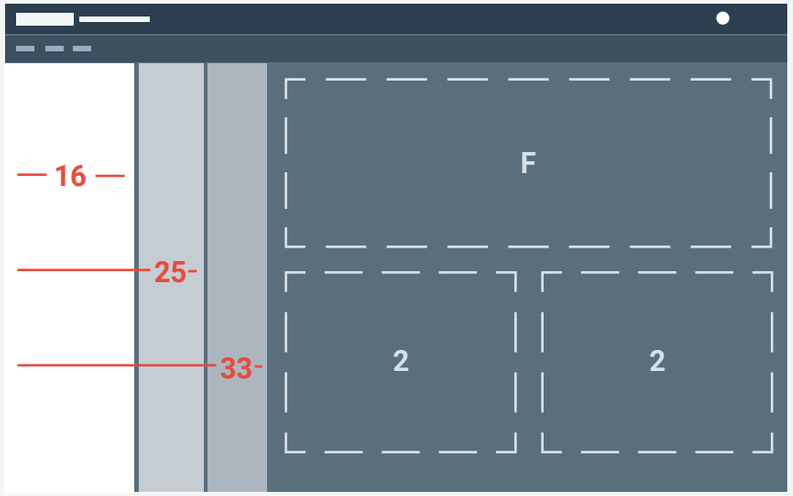


Sibilings
Elevation: 1dp

Naming

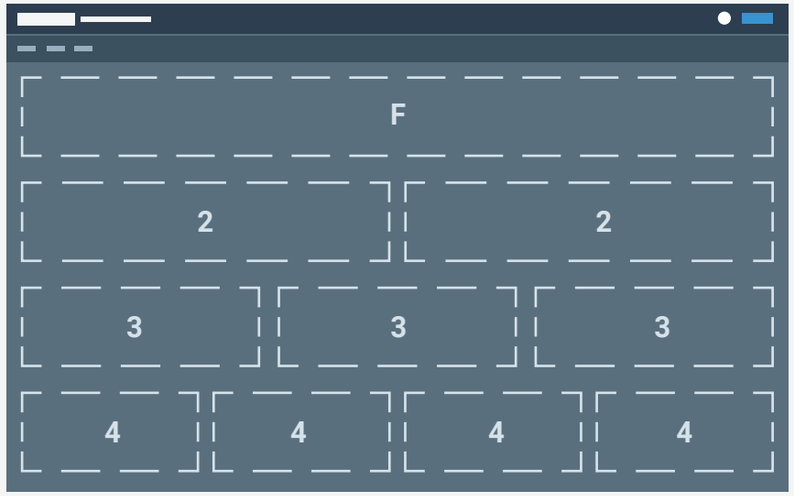
In an effort to make names shorts and descriptive, templates will use the following naming convention: **Sidebar Width** / **Columns**

Elevated



16/F – Sidebar of 16% / Full Width
16/2 – Sidebar of 16% / 2 Columns
25/F – Sidebar of 25% / Full Width
25/2 – Sidebar of 25% / 2 Column
33/F – Sidebar of 33% / Full Width
33/2 – Sidebar of 33% / 2 Column

Flat



F – No Sidebar / Full Width
2 – No Sidebar / 2 Columns
3 – No Sidebar / 3 Columns
4 – No Sidebar / 4 Column