



LUXCE

UI Style and Interaction Guide for Mission Planning



Version 1.0
Created on May 19, 2021

Revision History

Version	Release Date	Revised By	Description
1.0	June 4, 2021	GTRI	Initial release of UI Style and Interaction Guide
-	-	-	-
-	-	-	-
-	-	-	-



Table of Contents

Design System Overview

- Material-UI 05
- Colors 06
- Contrast 07
- Typography 08
- Graphics and Icons 09
- Creating Depth 10
- Imagery 11

Responsive Design

- Responsive Grid 13
- Breakpoint Definition 14
- Fluid Content 15
- Tips and Tricks 16

Component Specifications

- Summary 18
- Button 19
- Date/Time Picker 20



01

Design System Overview

Material-UI

What is Material-UI

The Core Mission Planning software teams are using the React JavaScript library to build the front-end mission planning user interfaces. Developed by Google in 2014, Material-UI is a CSS framework built for React that gives access to a large collection of complete, reusable components built to support fully responsive websites. It utilizes a balance of flat and skeuomorphic design principles to provide a well rounded, flexible, and user centric design language.

According to it's designer, Matías Duarte; "unlike real paper, our digital material can expand and reform intelligently. Material has physical surfaces and edges. Seams and shadows provide meaning about what you can touch." Material-UI was chosen due to its ease of integration with React, support for responsive design, and ease of styling to match the look & feel desired by the users of the Core Mission Planning interface.

More information on Material-UI can be found on the [Material-UI website](#).



How LUXCE is Utilizing Material-UI

Material-UI serves three important roles for the LUXCE team. First and foremost, it provides a comprehensive framework that can be used as a ground truth for the Core Mission Planning interface. It provides important tools (e.g., a responsive grid) that allows multiple teams to standardize the design and development of screens for the interface. Secondly, Material-UI provides a robust component library that serves as the backbone of the LUXCE component specifications. The vast majority of our components are simply updated version of those that already exist in the Material-UI component library. This strategy was implemented to help with consistency for the designers and ease of implementation for the developers. Finally, Material-UI provides an ever-expanding, versatile icon set that is utilized throughout many components and within the interface itself. Utilizing this existing Material-UI iconography wherever possible helps streamline the entire design and development process.



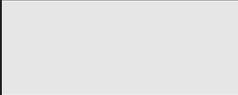
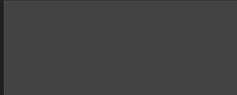
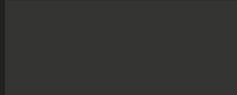
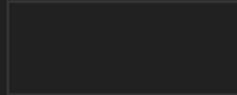
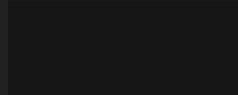
Colors

Guidelines for Using Color

Colors for the Core Mission Planning user interface were selected to provide acceptable contrast between foreground content and the display background and are used in accordance with guidance on color coding and color meanings derived from MIL-STD-1472H. Color shall not be used as a primary coding technique per MIL-STD-1472H. Color shall only be used as a **redundant coding technique**; that is, the coding of information shall not be represented by color alone.

The swatches below outline the major colors used in the Core Mission Planning user interface (*defined by both a hex value as well as RGB values*) and each color has been given a name that will be used to identify it throughout the rest of this document.

The primary accent color in the Core Mission Planning user interface is PrimaryBlue. This is the “active color” for the interface, and should be used to draw the user’s attention to focused tasks, active selections, and primary controls within the interface. PrimaryBlue should not be used for hover states, but rather for active or focused states. Other saturated colors are generally only used in the interface to indicate alerts (*success, warning, error*) or as visual cues to indicate mission-specific information within the interface. Other colors may be utilized when necessary within the interface outside of this core palette but **must be denoted** in the specification for that specific component or page.

 White #ffffff R: 255 G: 255 B: 255	 Graye5 #e5e5e5 R: 229 G: 229 B: 229	 Graybd #bdbdbd R: 189 G: 189 B: 189	 Gray42 #424242 R: 66 G: 66 B: 66	 Gray33 #333333 R: 51 G: 51 B: 51	 Gray22 #222222 R: 34 G: 34 B: 34	 Gray17 #171717 R: 23 G: 23 B: 23
 Black #000000 R: 0 G: 0 B: 0	 PrimaryBlue #0979cc R: 9 G: 121 B: 204	 DarkBlue #0064b7 R: 0 G: 100 B: 183	 SuccessGreen #0064b7 R: 0 G: 100 B: 183	 ErrorRed #0064b7 R: 0 G: 100 B: 183	 WarningAmber #0064b7 R: 0 G: 100 B: 183	



Contrast

Ensuring Proper Contrast Ratios

The contrast ratio between foreground elements and background elements must be sufficient for users to be able to read the content in a variety of lighting conditions on a wide range of screens. While best practice has traditionally meant striving to meet the MIL-STD-1472H guidelines for contrast, these standards can be hard to achieve when working with a complex user interface in a responsive environment. Dogmatic adherence to MIL-STD-1472H can actually lead to a lack of hierarchy and clutter up page designs which drives down usability. When meeting MIL-STD-1472H guidelines is not feasible, the Web Content Accessibility Guidelines (WCAG) provide a slightly more flexible framework while still ensuring proper usability.

WCAG recommends a **minimum contrast ratio** of 4.5:1 for normal text and 3:1 for large text and graphics, with preferred contrast ratios of 7:1 for normal text and 4.5:1 for large text. (*Large text is defined as 24 px, or 18.66 px bold.*) The matrix below shows the contrast ratio for all of the major colors found in the Core Mission Planning user interface. Values that appear in **red** denote color combinations that do not pass WCAG and values that appear in **amber** denote those that only pass for large text.

Additionally, there are a variety of websites (such as [Contrast Ratio Tool](#) or [WebAIM Contrast Checker](#)) that provide useful contrast checking tools for designers.

	White	Graye5	Graybd	Gray42	Gray33	Gray22	Gray17	Black	PrimaryBlue	DarkBlue
White		1.25	1.87	10.04	12.63	15.9	17.92	21	4.54	5.98
Graye5	1.25		1.49	7.97	10.02	12.63	14.23	16.67	3.6	4.75
Graybd	1.87	1.49		5.34	6.72	8.46	9.54	11.17	2.42	3.18
Gray42	10.04	7.97	5.34		1.25	1.58	1.78	2.08	2.21	1.67
Gray33	12.63	10.02	6.72	1.25		1.25	1.41	1.66	2.77	2.1
Gray22	15.9	12.63	8.46	1.58	1.25		1.12	1.31	3.49	2.65
Gray17	17.92	14.23	9.54	1.78	1.41	1.12		1.17	3.94	3.0
Black	21	16.67	11.17	2.08	1.66	1.31	1.17		4.61	3.5
PrimaryBlue	4.54	3.6	2.42	2.21	2.77	3.49	3.94	4.61		1.31
DarkBlue	5.98	4.75	3.18	1.67	2.1	2.65	3.0	3.5	1.31	



Typography

Guidelines for Using Typography

Most of the text in the UI should be plain (without attributes like bolding, Italics, underlining, etc.). When text attributes are used, consider the following guidelines:

- Bold text can be used to make text easier to parse. It should be used sparingly to draw attention to text that users must read.
- Excessive use of bold text lessens its impact.
- Italic text should be used sparingly to emphasize specific words.
- Italic text is also used in the UI design to denote undefined values or incomplete information.
- Text that is both bold and italic should be avoided and underlined text should be avoided except for hyperlinks.

Fonts

The primary font used in the interface is Roboto, a general purpose sans-serif font. This font is utilized for the majority of control labels as well as body copy. Chakra Petch is a more stylized, angular font used for some headings and for accent purposes. Both fonts are free, open source, and available for download from Google Fonts.

Roboto Chakra Petch

Font Hierarchy Table

Heading 1 •---- Chakra Petch Bold @ 2.5rem (40pt)

Heading 2 •----- Chakra Petch Bold @ 1.25rem (20pt)

Heading 3 •----- Roboto Medium @ 1.25rem (20pt)

Heading 4 •----- Roboto Medium @ 1.125rem (18pt)

Heading 5 •----- Chakra Petch Regular @ 1.0625rem (17pt)

Body Text Regular •----- Roboto Regular @ 1rem (16pt)

Body Text Small •----- Roboto Regular @ 0.875rem (14pt)

Body Text Extra Small •----- Roboto Regular @ 0.75rem (12pt)

Button Label Large •----- Roboto Medium @ 0.9375rem (15pt)

Button Label Medium •----- Roboto Medium @ 0.875rem (14pt)

Button Label Small •----- Roboto Medium @ 0.8125rem (13pt)

Fonts are specified in rem units. One rem equals the font size of the body text of an HTML page, in this case 16pt. This allows fonts to be scaled for varying screen sizes at the same proportions. Further information can be [found here](#).



Graphics and Icons

Guidelines for Using Graphics and Icons

Vector graphics and icons can quickly convey meaningful information to users at strategic points within the interface and are often helpful in reducing the amount of text necessary to convey information. The majority of graphics and icons used in the interface should be semantic (*to convey meaning and identify actions, either alone or in conjunction with text*); decorative icons should be used sparingly.

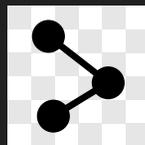
Icons should convey the intended meaning as clearly as possible through visual association with the thing they represent, either literally (*e.g., a house icon for “Home”*) or through established associations and software conventions (*e.g., a trash can icon for “Delete”*). Tooltips can be used to reinforce icon meanings. Purely abstract icons **should be avoided**.

Sources for Existing Icons

Where possible, icons from the official [Material Icons library](#) will be used (*the “Filled” set is recommended, but the “Outlined” set can be used to supplement*). Additional icons can be sourced from the community-generated [Material Design Icons library](#).

Creating Custom Icons

In situations where no suitable icon exists in either of the libraries linked above, custom icons in a compatible style can be created from scratch or by modifying existing icons. Custom icons should be tested with a representative user population to ensure that the intended meaning is conveyed. Custom icons should adhere to the following Material-UI guidelines:



- Custom icons should be scaled for a 24x24 pixel viewport
- Custom icons should have transparent backgrounds
- Custom icons should be provided in an .SVG format



Creating Depth

Guidelines for Using Transparency

Black, white, and various shades of gray are used throughout the interface in keeping with the overall “dark theme” aesthetic of the user interface. Generally speaking, lighter text will be easier to read, but this must be balanced against the need to create hierarchy and make different types of content distinguishable. For example, making all text white would maximize contrast but would make it more difficult to differentiate between the label and the input text in a text field. Transparency is used extensively, particularly for white text and other white elements.

Some user interface components, such as text inputs and drop downs, use transparency so that they can be displayed over a variety of backgrounds while still maintaining hierarchy (see table below). To ensure sufficient contrast, user interface components using transparency should typically not be used on background colors lighter than Gray33. Semi-transparent black boxes can be used behind content to group related controls and provide additional contrast to components like text fields and selects that use transparency (see page 15, “Tips and Tricks”).

Definition

Typical Uses

White at 100% opacity	Top-level information (titles, headings)
White at 60% opacity	Secondary information (control labels)
White at 30% opacity	Disabled items
White at 15% opacity	Hovered component backgrounds
White at 10% opacity	Component backgrounds

Guidelines for Using Elevation

Elevation, or the visual layering of interface components on the z-axis, provides important interaction cues in the Material-UI framework. It should only be used to indicate when a piece of content sits “above” another. Surfaces positioned in front of other surfaces typically focus the user’s attention (as in a dialog window) or control the content behind it (as in navigation and tab bars). For example, using a drop shadow on the Top-Level Navigation Bar indicates to the user that the components on that bar control the underlying content, and that when the user expands the bar, the expanded panel will appear on top of other content. Drop shadows are one method for providing cues about elevation, direction of movement, and the edges of surfaces. Drop shadows **should be avoided** on interface components that are not 100% opaque.



Imagery

Guidelines for Using Images

Photographic images may be used as accents in the interface to provide visual interest and theming appropriate to the intended user population. Only **high quality, professional images** (free from compression artifacts) should be used. Use of accent images should be limited to simple, high-level pages (such as the login screen or the home page) and should not distract from the primary focus of the interface.

Several screens within the design (such as the Login Page and Homepage Dashboard) utilize background images with a dark overlay (Black at 70% opacity) on top of them. Utilizing a dark overlay on top of an image should only be used in specific scenarios when it is necessary to help with the legibility of content that appears over top of that image.

Responsive Design Considerations

It is important to remember that the images you select will be viewed at a variety of sizes across a variety of devices. When selecting an image, the first thing you should consider is resolution. Selecting a high resolution image will ensure that it does not appear “fuzzy” on newer screens and high density displays. In addition, it is vital to view the interface on a variety of devices during testing while paying close attention to image fidelity.

The second important consideration for image selection is cropping. In a responsive environment it is likely that any image you select will appear not only at a variety of sizes but also in both landscape and portrait orientations depending on the user’s device. Therefore, it is important when selecting images to be cognizant of how they will look in both orientations and communicate with the development team to ensure images are embedded in a way that best preserves their intended purpose.

Sourcing and Attributing Images

Any images used in the interface should be appropriately licensed for use in the application, with preference to images in the public domain, United States government creative works, or Creative Commons images with no known copyright restrictions. Images requiring attribution should be properly attributed. Use of images with additional restrictions (e.g., no modification allowed or non-commercial use only) should be avoided.

Two of the best sources comprised of high quality images with appropriate licenses are the [U.S. Department of Defense](#) photo gallery and the [Wikimedia Commons](#) image repository.



02

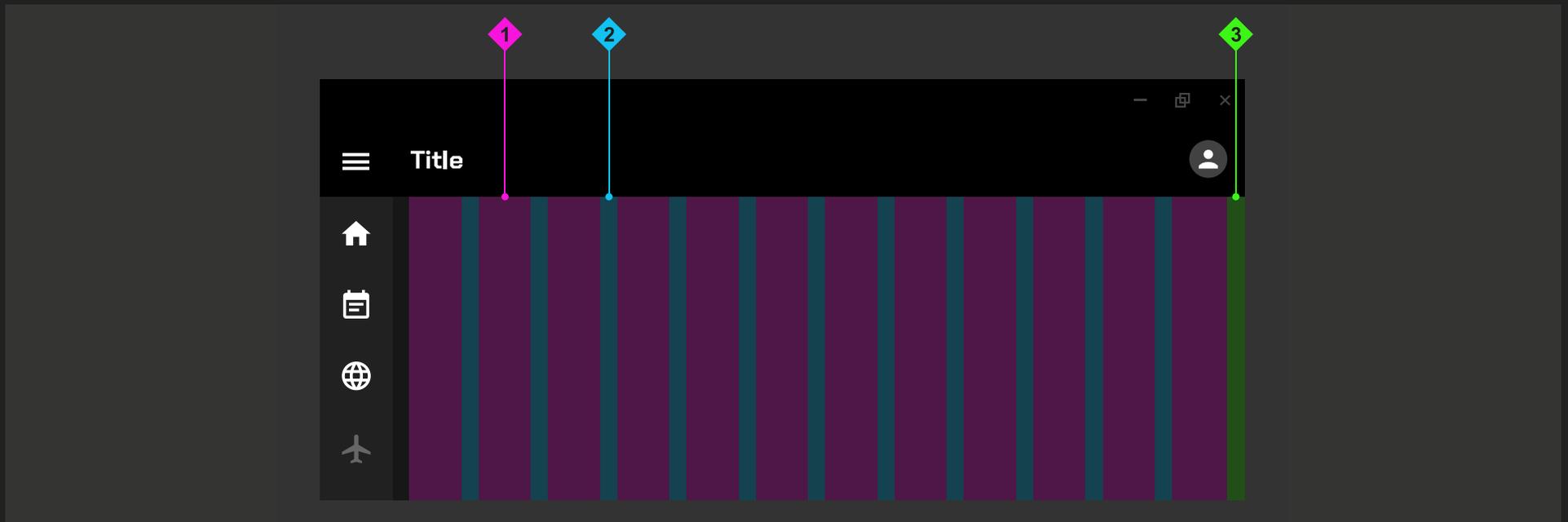
Responsive Design

Responsive Grid

Basic Overview and Grid Structure

Because the Core Mission Planning interface is designed as a web-based UI that may be used on a variety of devices, the UI design must be scalable to accommodate different display sizes, resolutions, and orientations. This is achieved through a design approach known as Responsive Design. This approach structures content on a flexible grid made up of columns, gutters, and margins (see the image below) that allow content to reflow seamlessly across different screen sizes. **Columns** 1 subdivide the content area and guide the layout of content. **Gutters** 2 are the space between columns. **Margins** 3 are the space between content and the left and right edges of the screen.

Material-UI uses a 12-column fluid grid layout, where the screen is divided into 12 columns with widths defined using percentages, and gutter and margin widths are fixed at each breakpoint. As the size of the interface changes (based on display hardware, display orientation, or window resizing), the twelve columns (and the content) will expand or contract to adapt to the screen size, while the fixed margins and gutters ensure consistent spacing between interface components. At four breakpoints (defined by Material-UI and based on common screen resolutions), the content may adjust dramatically (for example, by reorganizing into a more vertical layout) to display correctly at a different screen size.



Breakpoint Definition

Overview and Definition

Material-UI recommends five breakpoints based on common screen widths. Because small mobile devices (*i.e.*, *smartphones*) are out of scope for the Core Mission Planning interface, the *xsmall* breakpoint will not be considered in the design. Our responsive grid will support screen widths as narrow as 600 pixels, which accommodates tablets in the vertical (*portrait*) orientation.

For the Core Mission Planning interface, the margin width remains the same across all breakpoints. Gutters decrease in width at smaller breakpoints but are always defined as a multiple of 8 pixels per Material-UI guidance. The number of columns does not change at any breakpoint, but the page content may redistribute to ensure that the content remains usable.

Name	Breakpoint Range	Margins	Gutters
xlarge (xl)	1920 pixels+	24 pixels	24 pixels
large (lg)	1280 pixels - 1919 pixels	24 pixels	24 pixels
medium (md)	960 pixels - 1279 pixels	24 pixels	16 pixels
small (sm)	600 pixels - 959 pixels	24 pixels	8 pixels
xsmall (xs)	1 pixel - 599 pixels (<i>not applicable</i>)	N/A	N/A

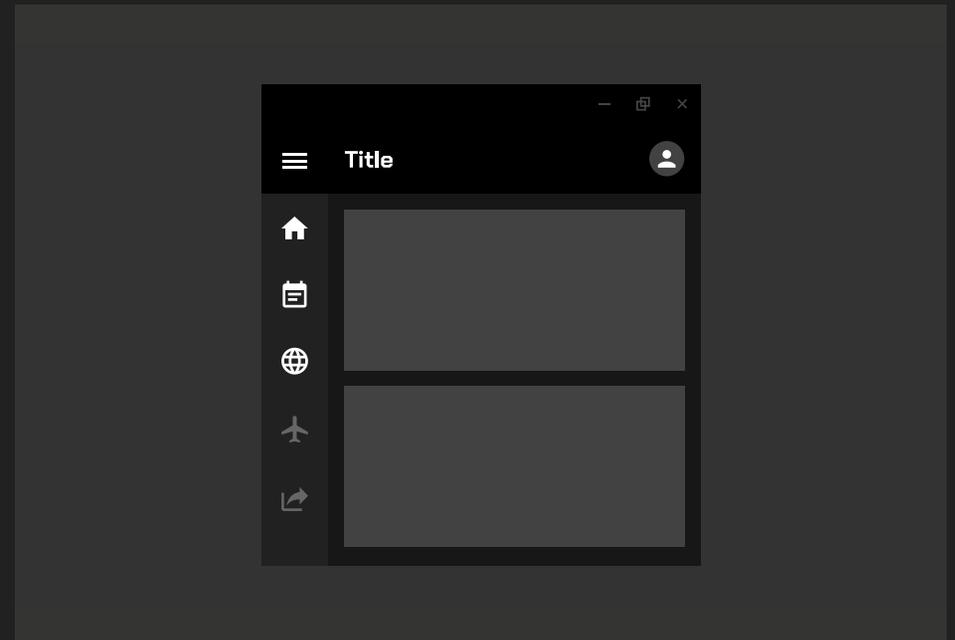


Fluid Content

Implications for Content Layout

As columns expand and contract across breakpoints, page content will also need to expand and contract. Often this means that a component may actually be wider at the smallest breakpoint than at the largest (*based on the number of columns it spans at each respective breakpoint*). This concept is illustrated in the images below. In the first image, four components each span 25% of the screen at the xlarge breakpoint and their approximate width would be 480 pixels (*25% of 1920 pixels*). In the second image the four components each span 100% of the screen in four rows at the small breakpoint and their approximate width would be 800 pixels (*100% of 800 pixels*).

It may also be necessary to set minimum widths for certain components to ensure that their content remains legible as they contract. When using minimum widths, the layout of page content may readjust within a breakpoint. For example, if three components occupy a 1920 pixel screen width at a 50% / 25% / 25% split and the screen width decreases to 1300 pixels (*still within the large breakpoint but small enough to violate the minimum width set for the first component*) the content layout will adjust to two rows with the first component spanning 100% of the first row and the other two components each spanning 50% of a second row.



Tips and Tricks

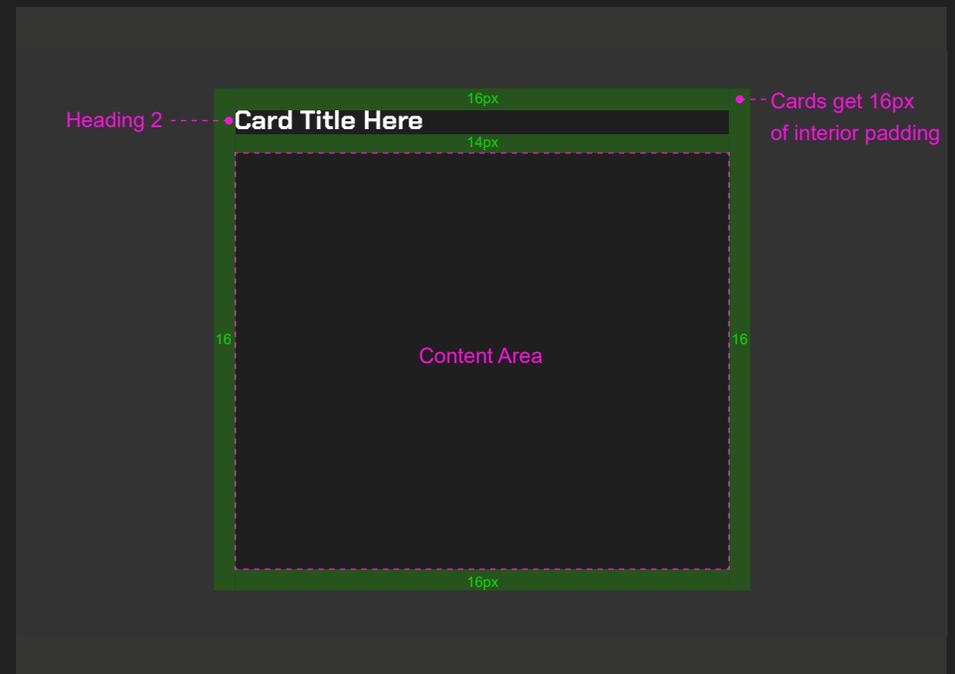
Designing on a Responsive Grid

Responsive design requires flexibility. Unique content will require a variety of layout strategies for implementation across screen sizes. Below are some basic guidelines and best practices for designing responsive layouts. There's no uniform solution for laying out content but following these guidelines will help achieve a usable result.

- Follow the grid. Components can span multiple grid columns and gutters, but should always start and end on column boundaries.
- Don't view content as having a set width. Content will need to expand and contract along with the browser window.
- Consider all breakpoints when laying out content. Understand how content will move and react as screen sizes decrease or increase. Considering only a very small size or a very large size may result in a layout that doesn't work at the opposite end of the range of breakpoints.
- Use cards to help control the user experience. Cards help keep the content within them organized as desired while the card itself remains aligned to the grid.
- Use minimum widths to help control the user experience. Setting minimum widths for text fields and select controls with long labels or inputs ensures that they do not get cut off for users. It's preferable to have extra wide controls than to compromise usability by truncating labels or typical inputs values.
- Embrace vertical scrolling. As screen sizes decrease, it often makes sense to allow components to span the entire screen width (*within the margins*). This will inevitably push some content farther down the screen, requiring scrolling. Vertical scrolling is a common and expected interaction in web interfaces (*particularly on smaller displays like tablets*). Horizontal scrolling should be avoided as much as possible.

Using Cards to Control Content

To provide additional structure to some interface pages, Material-UI card components are used to group information. Using cards allows for additional spacing between content groups and allows more flexibility in the distribution of components within the cards. Each card aligns with the 12-column grid but the content within each card uses the card as its frame of reference. This allows greater flexibility and uniformity for content layout within each card and allows the use of minimum and maximum widths to prevent content from shrinking or expanding past the point of legibility.



03

Component Specifications

Summary

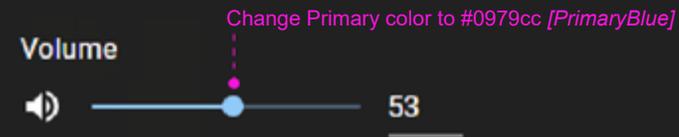
Component Specification Structure

Components in the Core Mission Planning UI will be based on Material-UI, an open-source user interface library that implements Google's Material Design. Visually, the components will be based on Material-UI's dark theme with some custom appearance changes to create a consistent theme for the mission planning UI. Extensive documentation of Material-UI components can be found on the [Material-UI website](#).

Visual specifications for the current set of available components are provided in the following sections. Each specification graphic shows the original appearance of the Material-UI component, the notional appearance of the component with the custom styling applied, magenta annotations describing any visual customizations made to the original components, green annotations denoting margins and spacing, occasional yellow annotations with any supplemental behavioral information about the component, and a [reference link](#) to the Material Component.

When utilizing an existing Material-UI component is not possible, custom components that provide capabilities and interactions not supported by the Material-UI components will be required. The specifications for these components will be nearly identical to those based on the Material-UI components except the Core Mission Planning visual styling will be applied directly to the custom component with no need for an original appearance graphic or reference link.

Original Material-UI Component

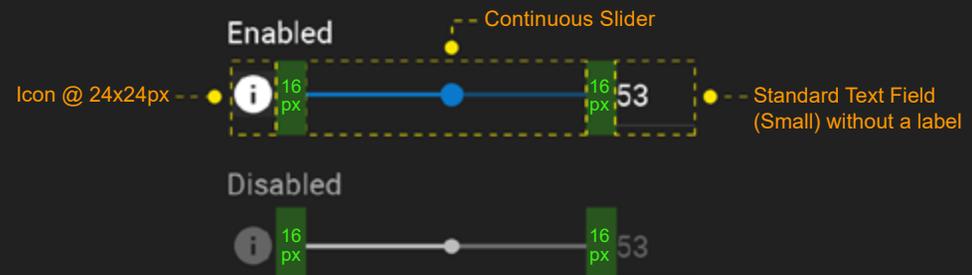


Decrease opacity to 60% -- ● Volume



Decrease opacity to 30% -- ●

Core Mission Planning Styled Version



Button

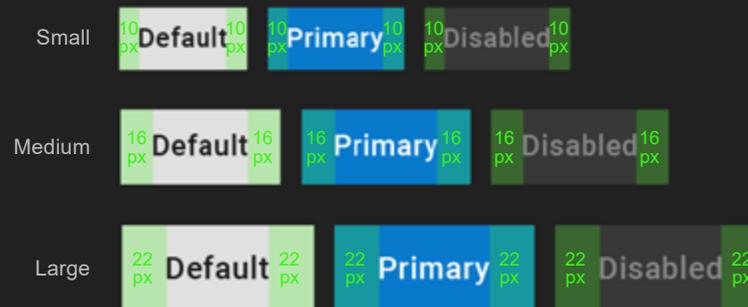
Component Summary

Buttons allow users to immediately execute actions with a single input (e.g., a click or a touch). The label of the button should clearly convey the action that will occur when the user activates the button. The visual style of buttons creates a hierarchy of salience; contained buttons are the most salient, followed by outlined buttons and text buttons. More salient button styles should be used for higher priority actions. For button styles that offer a range of sizes, the medium size should be used in most situations. The large size should only be used in situations with limited content (e.g., a login screen), and should be paired only with other appropriately sized controls (i.e., don't use a large button next to a medium button). The Primary button style should be used in most situations, however the Default button style is acceptable for use over imagery.

Original Material UI Component



Core Mission Planning Styled Version

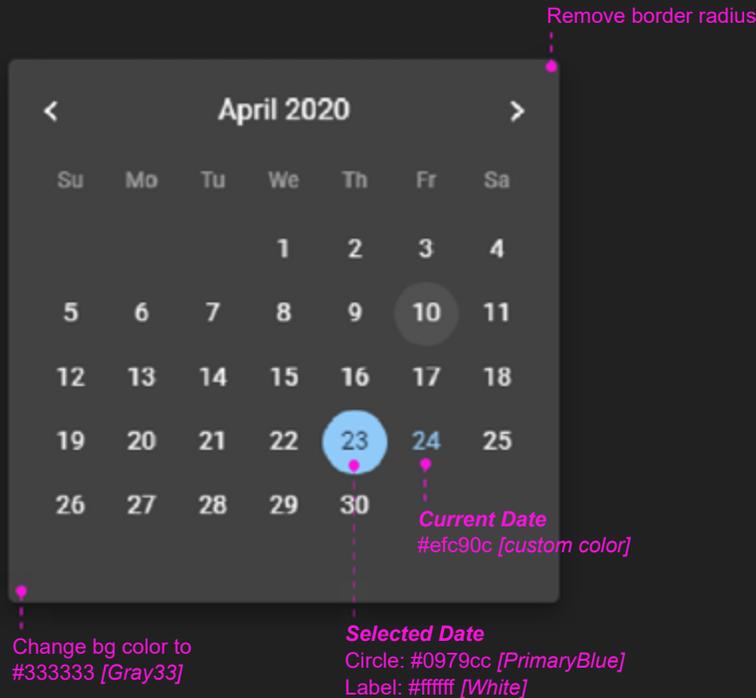


Date/Time Picker

Component Summary

Date and time fields provide support for keyboard entry of date and time information (*with format validation*) as well as graphical selection with pickers. Pickers can be displayed inline (*appearing as a temporary overlay below the field*) or as a dialog.

Original Material UI Component



Core Mission Planning Styled Version

