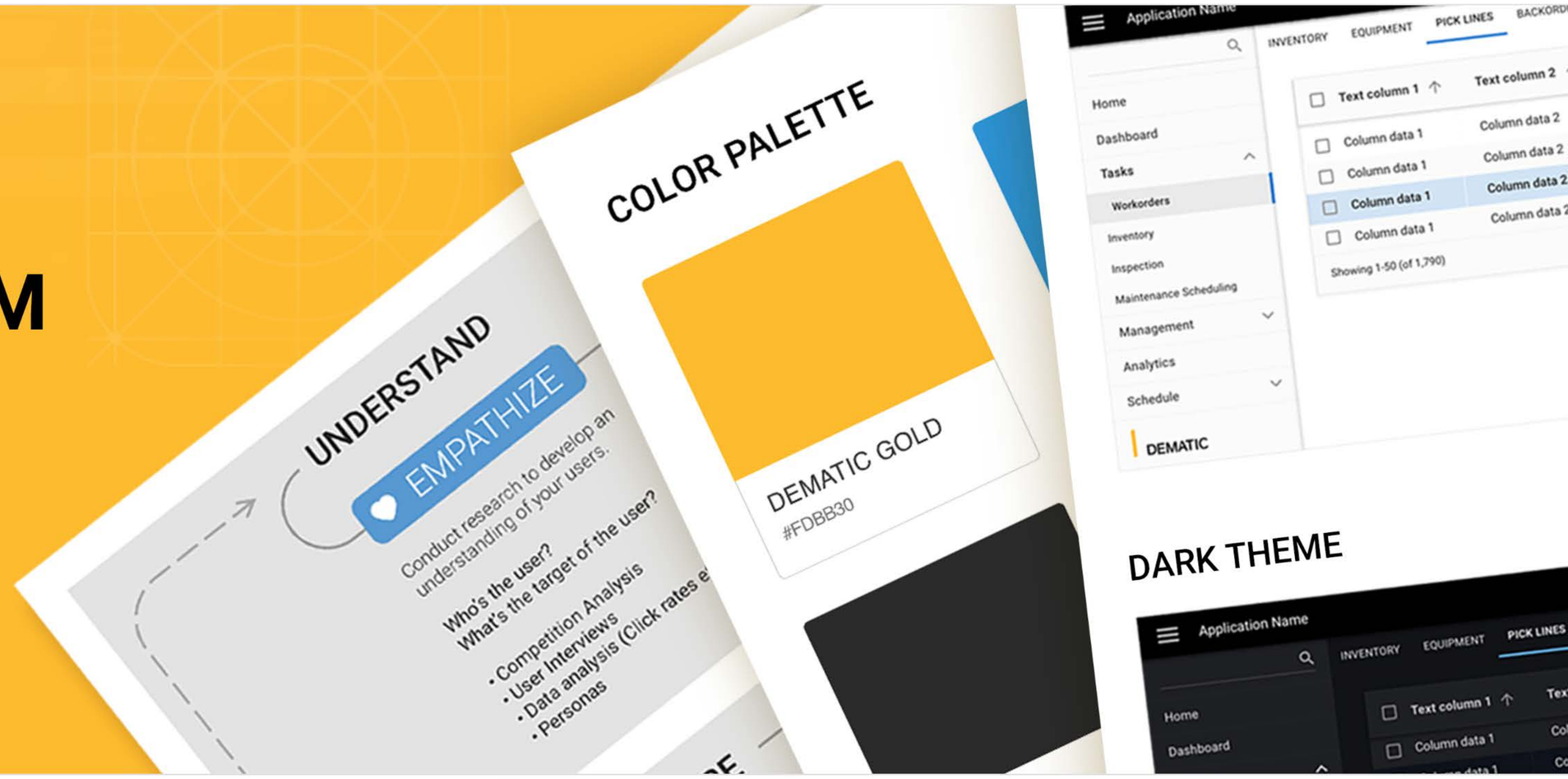
DEMATIC DSM

Updated 10 days ago

Your Single Source for
UX Design and Development



Storybook

A development environment for UI components.

Welcome to the Dematic Design System

What EXACTLY is a design system?

A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build any number of applications.

A design system enables teams to build better products faster by making design reusable—reusability makes scale possible.

It reduces design debt and builds bridges between teams working in concert to bring products to life.

Guiding Principles:

- **It's consistent.** The way components are built and managed follows a predictable pattern.
- **It's self-contained**. Your design system is treated as a standalone dependency.
- **It's reusable**. You've built components so they can be reused in many contexts.
- **It's accessible.** *A*pplications built with your design system are usable by as many people as possible, no matter how they access the web.
- **It's robust.** No matter the product or platform to which your design system is applied, it should perform with grace and minimal bugs.

Goals:

- To **improve customer satisfaction** and loyalty by the ease of use of our products
- To **reduce the learning curve** in using our products
- To **increase agility** in testing, maintaining, and releasing updates
- To provide **a branding and design standard** across all of our products

COMPONENTS
- Accordion
- Action Header 2.0
- AppFrame 2.0
- Button 2.0
- Calendar 2.0
- Card 2.0
- Chart 2.0
- Checkbox 2.0
- Chip 2.0
- Collapse / Expanda...
- Combo Box 2.0
- Dialog 2.0
- Drawer 2.0
- Dropdown 2.0
- Drop Zone 2.0
- Filter 2.0

COMPONENTS

Accordion

An accordion is a vertically stacked list of headers that reveal or hide associated sections of content.

Edited less than a minute ago by Adriene

Design | Code

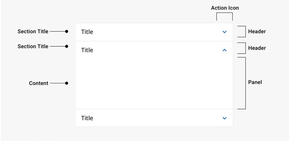
Usage

The accordion component assists in providing large amounts of content in a small space through progressive disclosure. The header gives a high-level overview of the content while a user can decide which sections to read in more detail. Since the detail content is hidden the accordion is best used when the hidden content is not a required to be read.

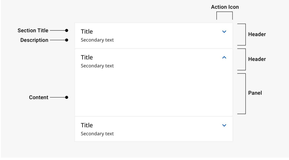
When to use:

- To organize related information.
- To shorten pages when content is not crucial to read in full.
- When space is limited and long content cannot be displayed all at once.

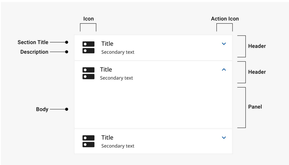
Anatomy



Single Line Accordion



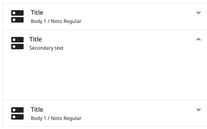
Double Line Accordion



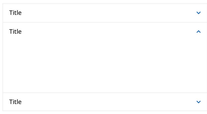
Double Line with Icon

Types

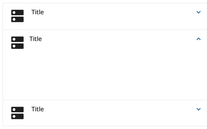
Double Line Header



Double Line Header with Icon



Single Line Header

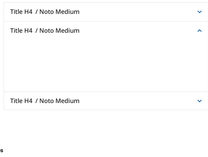


Single Line Header with Icon

Behavior

The accordion component has two main states: **collapsed** and **expanded**. The chevron icon at the end of the accordion indicates which state the accordion is in. The chevron points down to indicate collapsed and up to indicate expanded.

Collapsed State



Expanded State

Additional States

An Accordion can also be set with different state outside the default behavior.

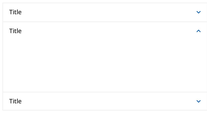
No Collapsing - There will always be one open item. If user clicks a closed accordion row it will open. When they try to collapse a an open accordion row nothing will happen.

Multiple Open Items - Multiple content sections can be open at the same time without one collapsing when the other one is opened

Set Open Items - Items that are specified will be opened initially upon first view.

Disabled State - A Panel is disabled and cannot be changed from its state when clicked.

Themes



Light Theme

Dark Theme

Do and Don't

✓ Do

When the accordion content is longer than the viewport the whole page should vertically scroll.

✗ Don't

Content should not scroll inside of an individual panel. Content should never scroll horizontally in an accordion.

✓ Do

Trigger collapsed and expanded states when clicking on either the header or icon.

✗ Don't

Make the icon the only way to trigger collapsed and expanded states.

COMPONENTS
- Accordion
- Action Header
- AppFrame
  - Design
  - Code
- Button 2.0
- Calendar 2.0
- Card 2.0
- Chart 2.0
- Checkbox 2.0
- Chip 2.0
- Collapse / Expanda...
- Combo Box 2.0
- Dialog 2.0
- Drawer 2.0
- Dropdown 2.0

COMPONENTS

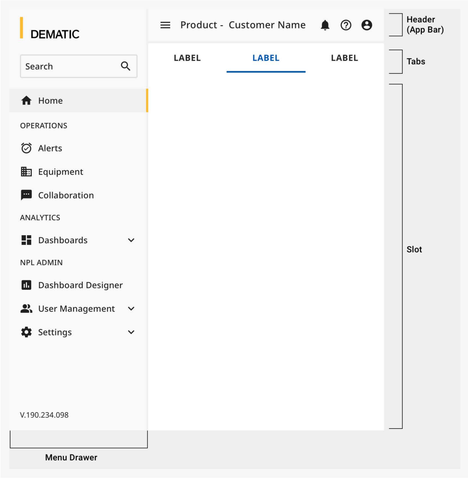
AppFrame

The app frame shell contains the header app bar, tabs, and left navigation drawer. It also provides a slot for other content within the app frame.

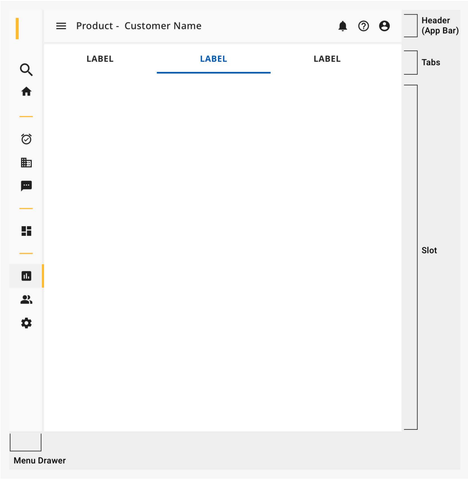
Edited less than a minute ago by Adriene

Design | Code

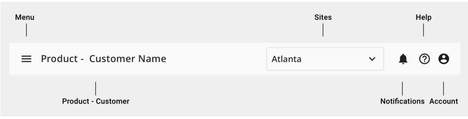
Anatomy



App Frame Expanded Menu Anatomy

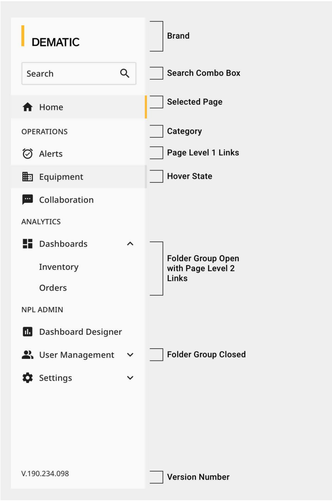


App Frame Condensed Menu Anatomy



App Bar Anatomy

- Menu Icon
- Product name and customer name.
- Site location selector
- Notification icon (Optional)
- Help Icon
- Account icon



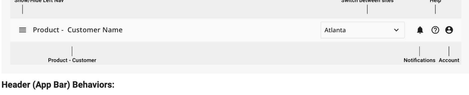
Expanded Menu Drawer Anatomy

- Brand
- Search
- Category Headers (Optional)
- Folder Groups used for relating sub categories in to a collapsible section
- Level 1 Page Links
- Level 2 Page Links
- Application Version Number

The Nav Drawer supports only 2 levels of Page Links. Consider using tabs or dropdown within page to switch to additional levels of content,

Behavior

Header



Header (App Bar) Behaviors:

- Upon selection of Menu icon the menu will expand and collapse between expanded and dense menu displays.
- The site switcher combo box (upon selection) will show all available sites per customer to choose from.
- The Notification icon will change appearance to Alert State, A message will appear there are #of Notifications that

COMPONENTS
- Accordion
- Action Header
- AppFrame
- Button
- Calendar 2.0
- Card 2.0
- Chart 2.0
- Checkbox 2.0
- Chip 2.0
- Collapse / Expanda...
- Combo Box 2.0
- Dialog 2.0
- Drawer 2.0
- Dropdown 2.0
- Drop Zone 2.0
- Filter 2.0

COMPONENTS

Button

Buttons allow users to perform an action or to navigate to another page. They have multiple styles for various needs and are ideal for calling attention to where a user needs to do something in order to move forward in a flow.

Edited less than a minute ago by Adriene

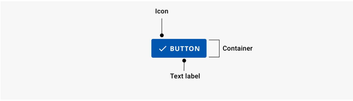
Design | Code | UI String Standards

Usage

Buttons communicate actions that users can take. They are typically placed throughout your UI, in places like:

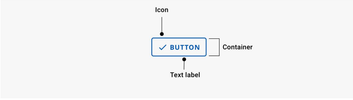
- Dialogs
- Modal windows
- Forms
- Cards
- Toolbars

Anatomy



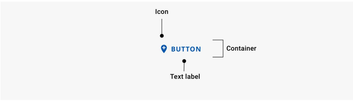
Primary Contained Button Anatomy

- Text label
- Container
- Icon (optional)



Secondary Outlined Button Anatomy

- Text label
- Container
- Icon (optional)

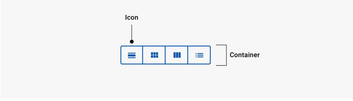


Text Button Anatomy

- Text label
- Container
- Icon (optional)

Toggle Button Group Anatomy

- Text label
- Container
- Icon (optional)

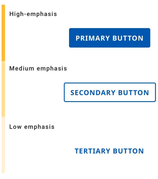


Toggle Icon Only Button Group Anatomy

- Container
- Icon

Emphasis

1. **Contained Buttons** - *(Primary High emphasis)* For the principal call to action on the page. Primary buttons should only appear once on a page.
2. **Outlined buttons** *(Secondary Medium emphasis)* For secondary actions it compliments a primary action or reduces visual noise when there are many actions of equal importance on the page. Secondary buttons should only appear alongside primary buttons. Use no more than two secondary buttons on a page, dialog, or card.
3. **Text buttons** *(Tertiary Low emphasis)* For simple less important actions that is not considered a primary or secondary action. Can be used inline because they are different from content in style and recognizable as buttons alongside content.
4. **Destructive Buttons** Determining which destructive button style to use will depend on the level of emphasis you want to give to the action. Destructive actions that are considered a required or primary step in a workflow should use the primary destructive button style. However, if a destructive action is just one of several actions a user could choose from, then a lower emphasis style like the secondary or tertiary style button may be more appropriate.

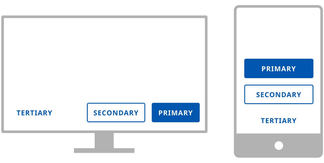


Hierarchy of Button Types

Placement

The Primary Button is placed to the right, to the left is the secondary and tertiary text button link.

In smaller screen sizes the Primary Button will wrap to the top and the Secondary Outline Button and Tertiary Text Buttons will wrap below.



Placement

Types

Primary/Contained Button

Primary Text Only | **Primary Lead Icon** | **Primary Trail Icon**

Secondary/Outlined Button

Secondary Text Only | **Secondary Lead Icon** | **Secondary Trail Icon**

Tertiary/Text Button

Tertiary Text Only | **Tertiary Lead Icon** | **Tertiary Trail Icon**

Destructive Button

Primary Destructive | **Secondary Destructive**

COMPONENTS
- Accordion
- Action Header
- AppFrame
- Button
- Calendar
  - Design
  - Code
- Card 2.0
- Chart 2.0
- Checkbox 2.0
- Chip 2.0
- Collapse / Expanda...
- Combo Box 2.0
- Dialog 2.0
- Drawer 2.0
- Dropdown 2.0

COMPONENTS

Calendar

Calendars allow users to view a day, a week, a work week, or a month of a particular year.

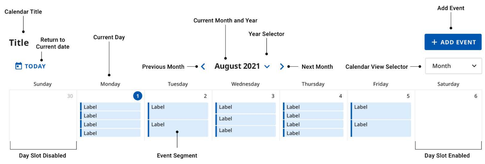
Edited less than a minute ago by Adriene

Design | Code

Usage

Calendars are used to display or schedule new events.

Anatomy

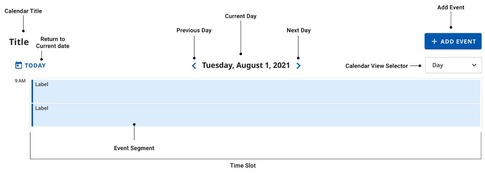


Monthly Calendar Includes:

- **Title:** For the Name of the page user is viewing the calendar on.
- **Today Button:** Used for a one click quick return to the current day
- **Year Selector:** Clicking this displays a popover which allows a user to select a new year than the one currently displayed.
- **Previous Month Arrow:** Changes the view to the month before the currently displayed month.
- **Current Month and Year:** Indicates the current date.
- **Year Selector:** Menu to select a year, upon selection the Months and Days will relate to that year.
- **Next Month Arrow:** Changes the view to the month after the currently displayed month.
- **View Selection:** Allows the user to change the view of the calendar to either Month, Week, Work Week, or Day views.
- **Inactive Date(s):** Indicates a day which is not included in the current month.
- **Event Segment:** Shows an individual event occurring on that particular date. If there are too many events to display for a particular date, the event tile will display 'X More Events' where X is the count of non-displayed events.
- **Day Slot:** Container representing an individual date. Contains a Day Label and if available, an Event.

Weekly Calendar Includes:

- **Title:** For the Name of the page user is viewing the calendar on.
- **Today Button:** Used for a one click quick return to the current day
- **Previous Week Arrow:** Changes the view to the week before the currently displayed week.
- **Next Week Arrow:** Changes the view to the week after the currently displayed week.
- **Time Slot:** Contains the day slots and event segments for the row.
- **View Selection:** Allows the user to change the view of the calendar to either Month, Week, Work Week, or Day views.
- **Week Day Name Label:** Displays the name of the week day.
- **Inactive Date(s):** Indicates a day which is not included in the current month.
- **Day Label:** Shows the numerical date of a day.
- **Current Day:** Indicates the current date.
- **Event Segment:** Shows an individual event occurring on that particular date. If there are too many events to display for a particular date, the event tile will display 'X More Events' where X is the count of non-displayed events.
- **Day Slot:** Container representing an individual date. Contains a Day Label and if available, an Event.

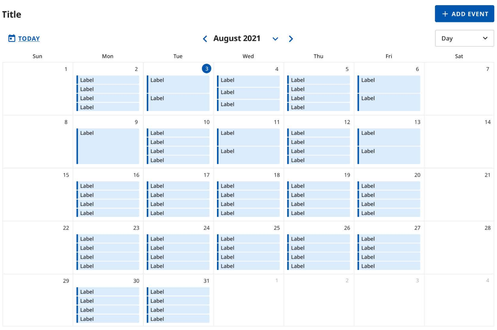


Calendar Day Includes:

- **Title:** For the Name of the page user is viewing the calendar on.
- **Today Button:** Used for a one click quick return to the current day
- **Current Day:** Indicates the current date
- **Previous Arrow:** Changes the view to the day before the currently displayed day.
- **Next Arrow:** Changes the view to the day after the currently displayed day.
- **Add Event:** User can add an event to calendar, upon click drawer will open with controls specified by product needs.
- **Calendar View Selector:** Allows the user to change the view of the calendar to either Month, Week, Work Week, or Day views.
- **Day Slot:** Container space for a day.
- **Event Segment:** Shows an individual event occurring on that particular date.

Types

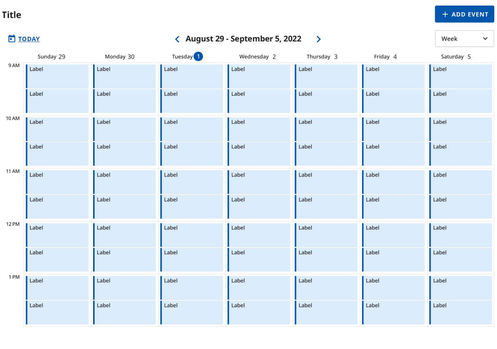
Monthly Calendar



Displays

events included in a calendar month in a grid.

Weekly Calendar



COMPONENTS

- Accordion
- Action Header
- AppFrame
- Button
- Calendar
- Card
- Chart
- Checkbox
- Chip
- Collapse / Expanda...
- Combo Box 2.0
- Dialog 2.0
- Drawer 2.0
- Dropdown 2.0
- Drop Zone 2.0
- Filter 2.0

COMPONENTS

Chip

Chips are compact elements that represent an input, attribute, or action.

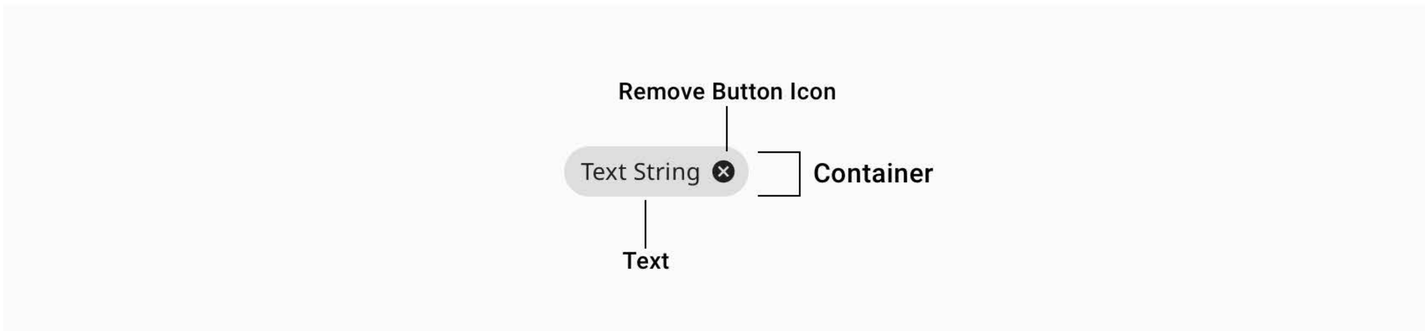
Edited less than a minute ago by Adriene

Design Code

Usage

Chips allow users to enter information, make selections, filter content, or trigger actions. Chips should appear dynamically as a group of multiple interactive elements. Unlike buttons, which should be a consistent and familiar call to action, one that a user expects to appear as the same action in the same general area.

Anatomy



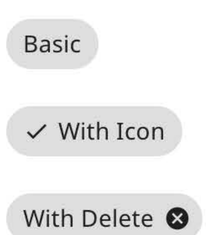
Chip Anatomy:

Container - Chip containers hold all chip elements, and their size is determined by those elements.

Text - Chip text can be an entity name, description, tag, action.

Remove button icon [optional] - Input chips can include a Remove icon.

Types



Button
Calendar
Card
Chart
Checkbox
Chip
Collapse / Expanda...
Combo Box
Dialog
Drawer
Dropdown
Drop Zone
Design
Code
Filter 2.0
Labeled Data 2.0
Menu 2.0

COMPONENTS

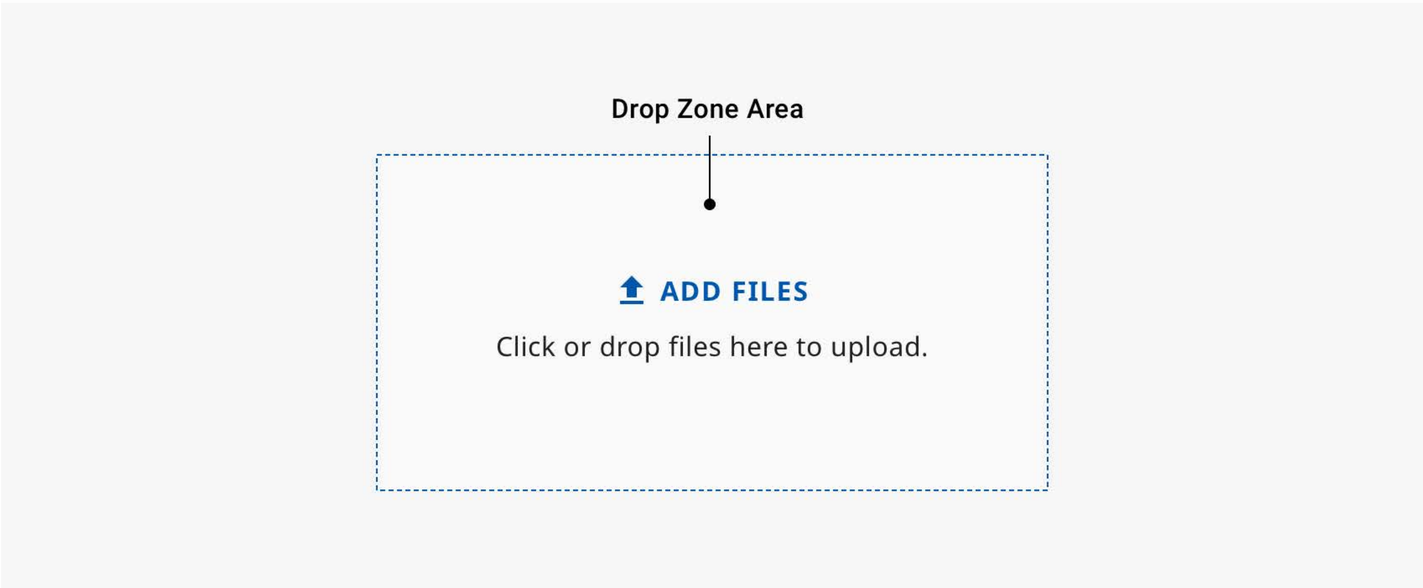
Drop Zone

The drop zone component lets users upload files by dragging and dropping the files into an area on a page, or activating a button.

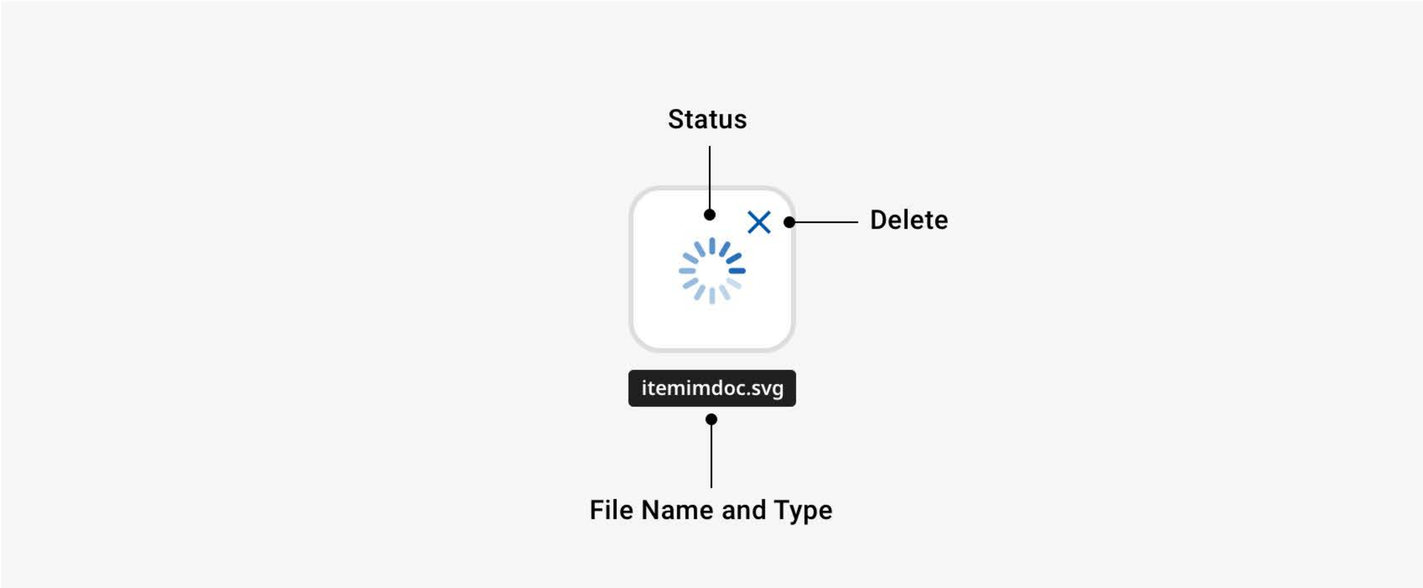
Edited less than a minute ago by Adriene

Design Code

Anatomy

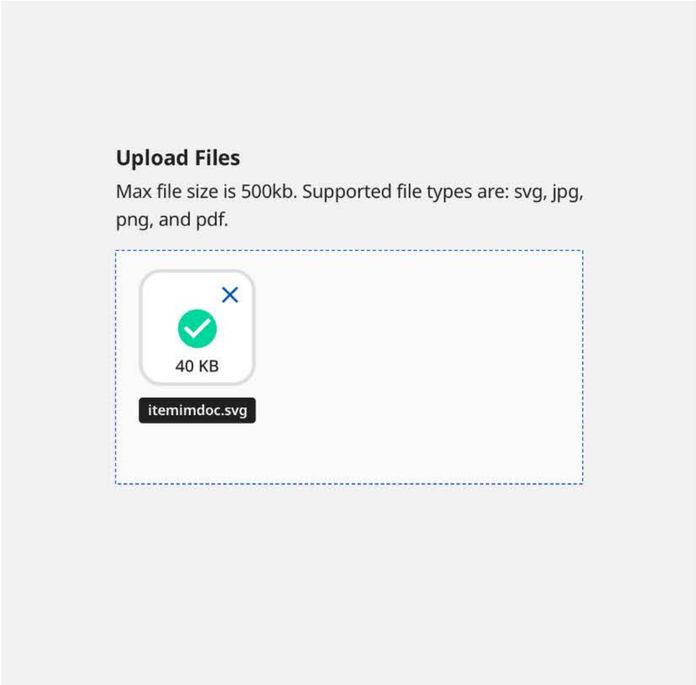


Anatomy



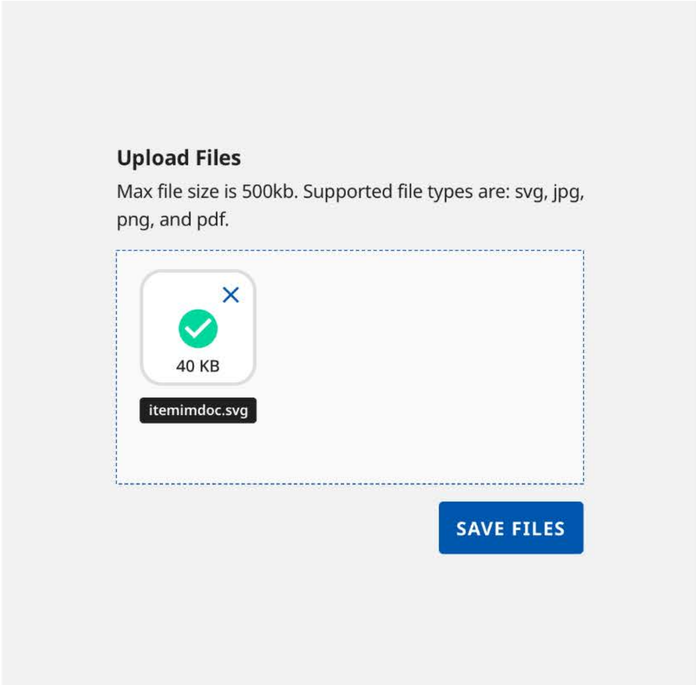
Anatomy DropZone File

Types



Default Upload

A file can be added by drag and dropping within the drop zone area.



Manual Upload

After File is uploaded a Save Button is used to save file to a specific place.