Sidebar Apps: Design Guidelines

Table of Contents

Resources

Garden
Garden Archive (for viewing component UX that hasn’t been moved to new Garden page yet)
Garden sticker sheet Sketch file - contains Sketch components for common Garden components

When you design an app for the Apps Sidebar, there are several things to take into consideration due to the amount of space available. Below are some general guidelines to keep in mind, and some UI/UX recommendations around basic, common Garden components that you might be using. In general, try to keep apps as compact and as simple as possible.

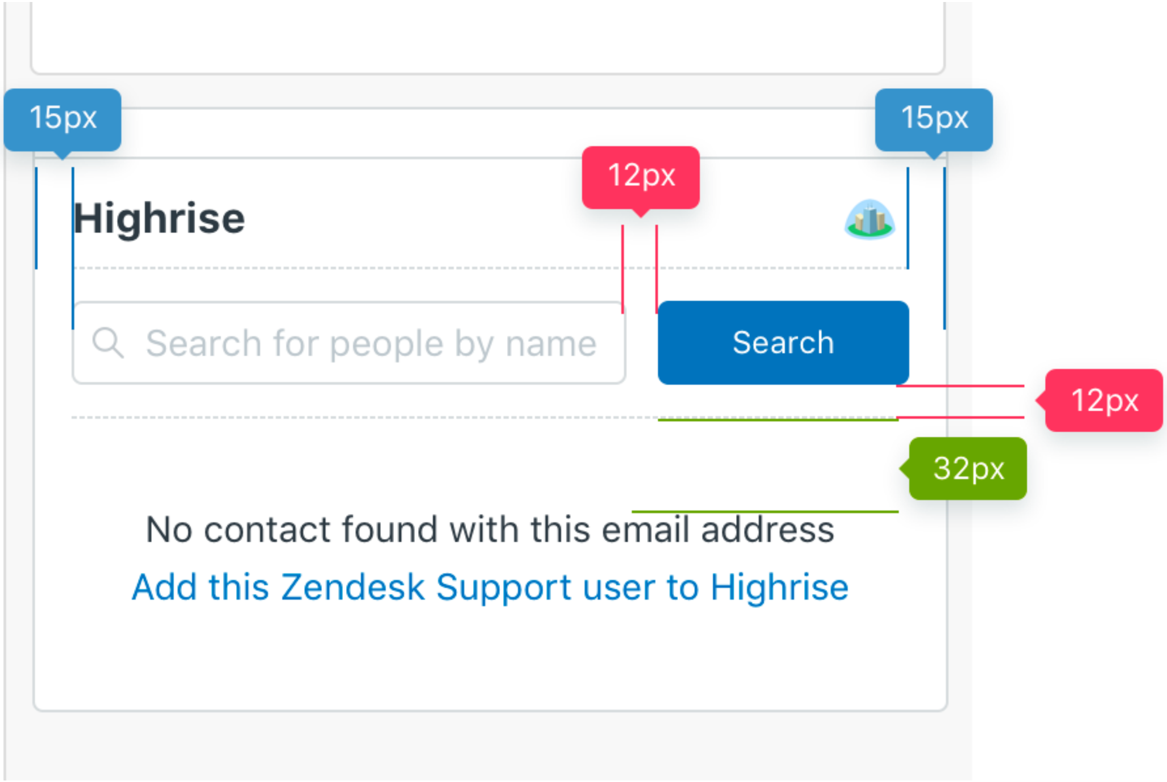
General Guidelines

Use the Base-4 system

Detailed doc

Use base-4 when determining padding between components. The only exception is that the `left-padding` and `right-padding` for the app container should be 15px (old base-5 system), otherwise things could cut off due to the iframe.

Common increments of padding are: 4px, 8px, 12px, 16px, 24px, 32px, 40px. You will notice in the input field components in Sketch that the padding between label + field is always 8px. If you follow the common patterns in the Sketch files, you'll be fine.

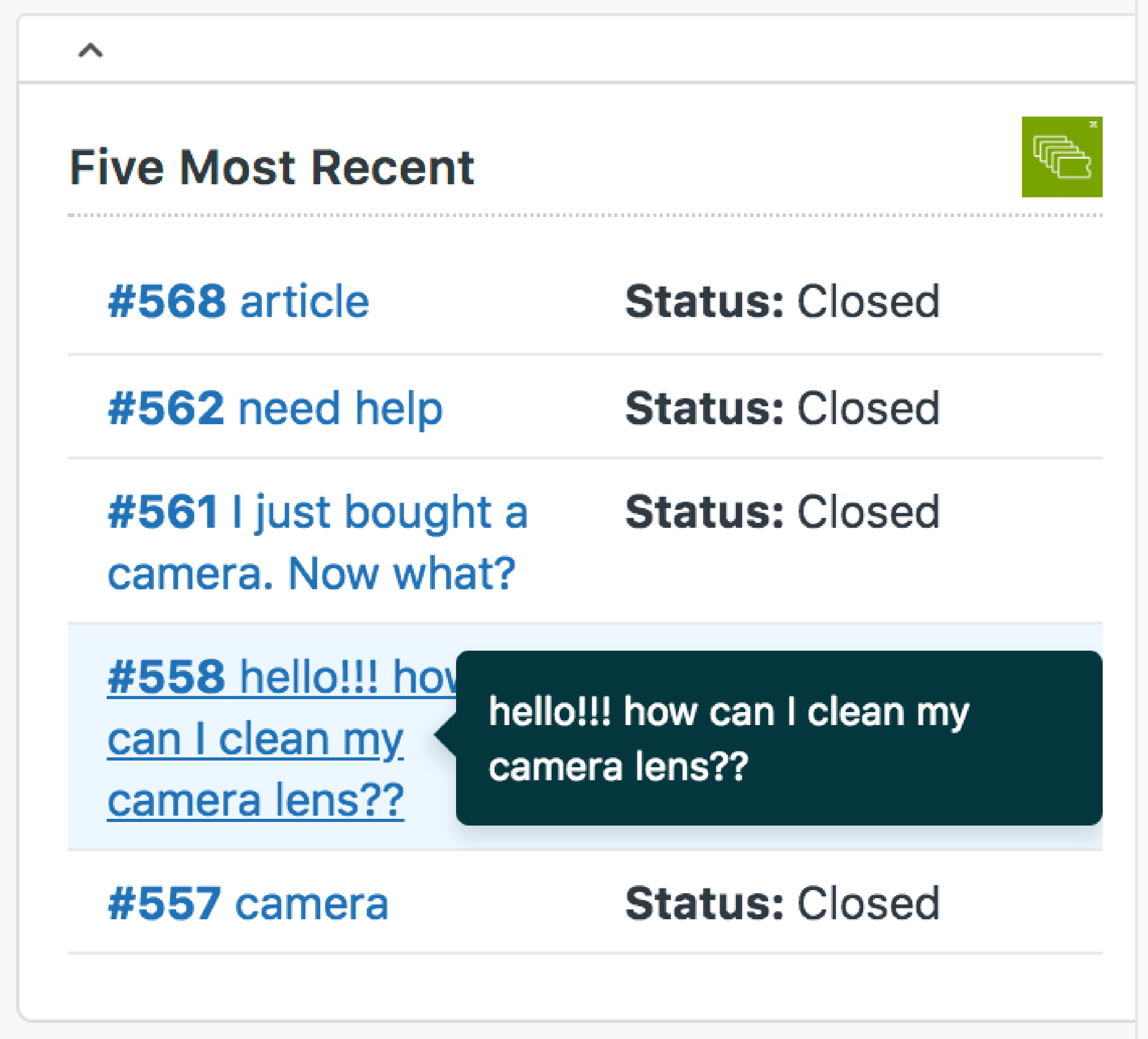

15px
15px
12px
Highrise
Search for people by name
Search
12px
32px
No contact found with this email address
Add this Zendesk Support user to Highrise

Accessibility

Garden components are designed to be accessible and if you stick to using Garden components, you should be fine on this front. However, there are some cases where the lack of space might contribute to inaccessibility.

Truncation and Tooltips

If you run into issues where there's not enough space for a line of text and you need to truncate with ellipses (eg. displaying 2-columns of data), you should always make sure that you include a tooltip on hover that will display the full string.



Five Most Recent

#568 article Status: Closed

#562 need help Status: Closed

#561 I just bought a camera. Now what? Status: Closed

#558 hello!!! how can I clean my camera lens??

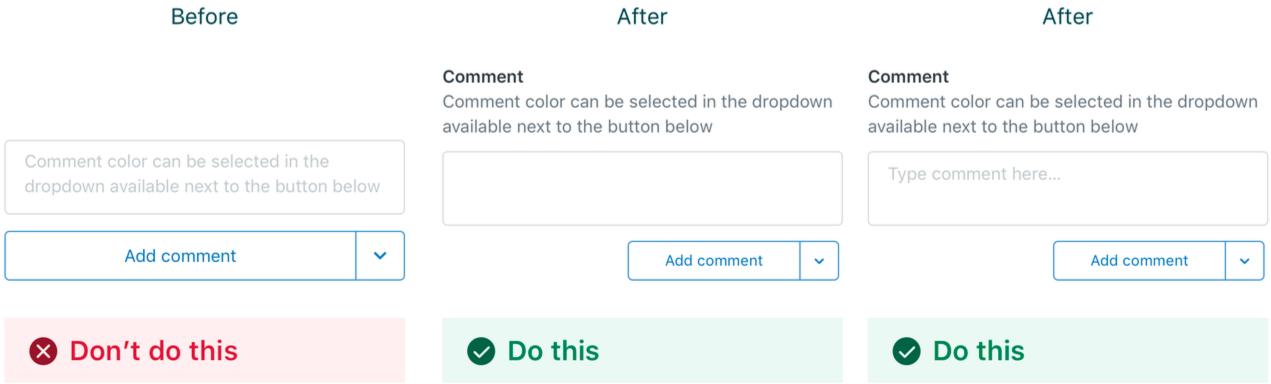
hello!!! how can I clean my camera lens??

#557 camera Status: Closed

Five Most Recent App. Strings shown above aren't long enough to truncate with (...) yet, but at least a user can view the entire ticket title if they need to with tooltips. These tooltips include logic to flip the correct way without cutting off due to the app iframe.

Text

Do not only use placeholder text to describe instructions in a field without a label since this is not accessible. If it's instructional, use hint text instead. As long as there is a label describing the field and optional hint text, it's fine to have minimal placeholder text. Do not use any full stops in hint text unless there is more than one sentence.



Comment color can be selected in the dropdown available next to the button below

Comment

Comment color can be selected in the dropdown available next to the button below

Comment

Comment color can be selected in the dropdown available next to the button below

Type comment here...

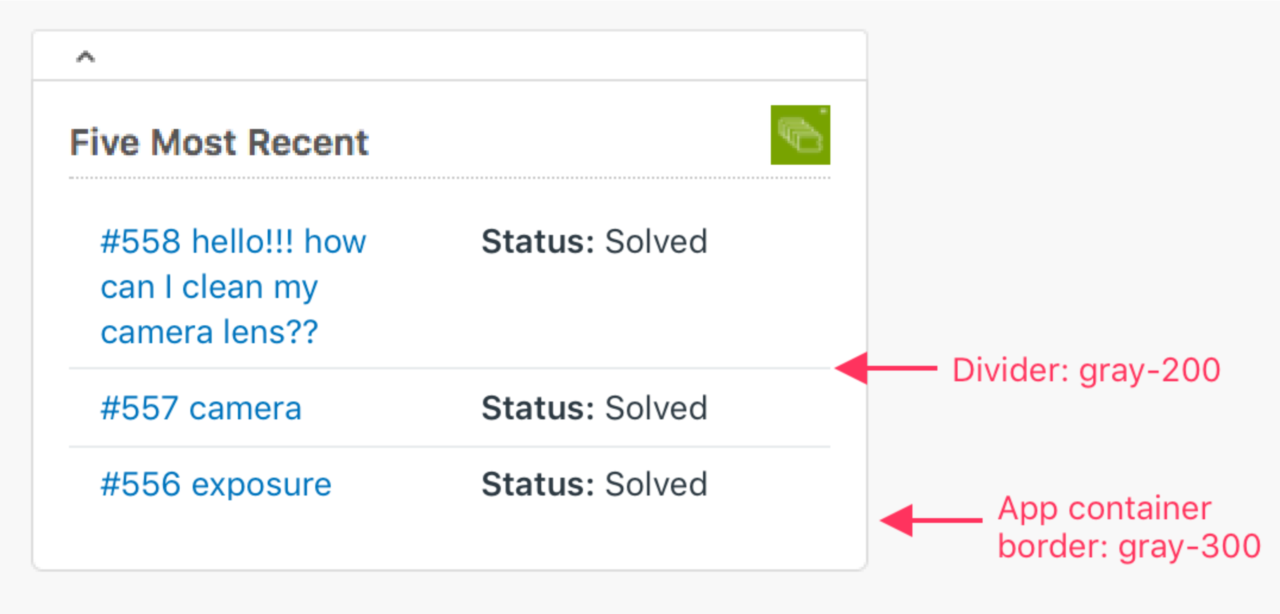
Having “Type comment here...” as placeholder text is fine because there is a clear “Comment” label and additional instructions in the hint text.

Disabled components

Avoid using any disabled components anywhere unless necessary for some reason. However, it’s OK to show disabled buttons in certain cases if it means we can avoid having buttons constantly appear and disappear for the end-user (this could be confusing).

Borders, Containers, and Dividers

Outer container borders should be `gray-300`. Any dividers in between content/data should be `gray-200`.



Five Most Recent

#558 hello!!! how can I clean my camera lens?? **Status:** Solved

#557 camera **Status:** Solved

#556 exposure **Status:** Solved

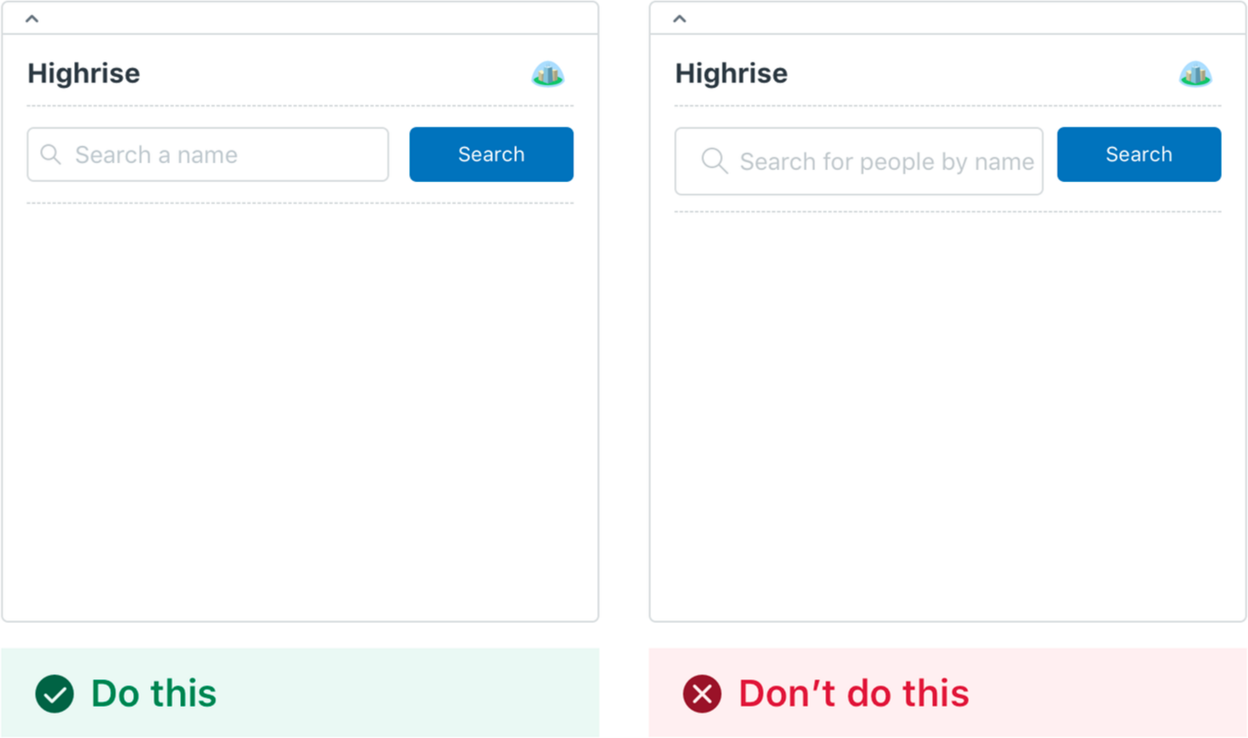
Divider: gray-200

App container border: gray-300

Buttons and Input Fields

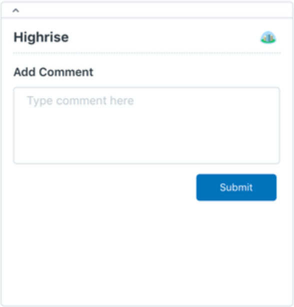
Because there isn't that much space available in the app container and we don't want to take up excessive vertical height, the recommendation is to use the `sm` (small) class buttons from Garden. This also means you should use `sm` input fields to match. A good rule of thumb is that if you place a button next to an input field, their heights should match.

The exception of using the `default` size buttons and input fields would be in a modal.



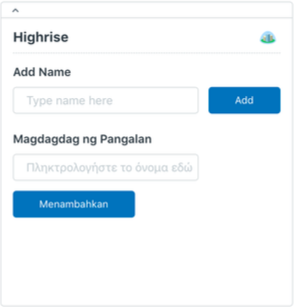
Use the same size class of buttons and input fields throughout the app. Also avoid using long placeholder text that could cut off. Note: The Garden search bar component doesn’t include an explicit “Search” button; it’s just shown for demo purposes above.

In this case, we’ve used a `Search` button next to an input field, which is not ideal for localization because it will probably wrap in a foreign language. However, it’s not uncommon to see a button below an input field so it still works. It would have been better to design it that way from the beginning since you can see that the placeholder text in English just barely fits inside the input field; this will definitely cut off in a foreign language.



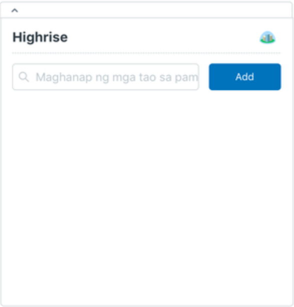
Do this

Buttons can be placed below an input field if there isn't enough horizontal space.



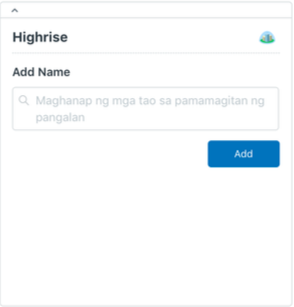
Do this sometimes

If necessary, you can place a button next to an input field, but you might run into wrapping issues because of width.



Don't do this

Keep placeholder text as short as possible for localization, as it could cut off.

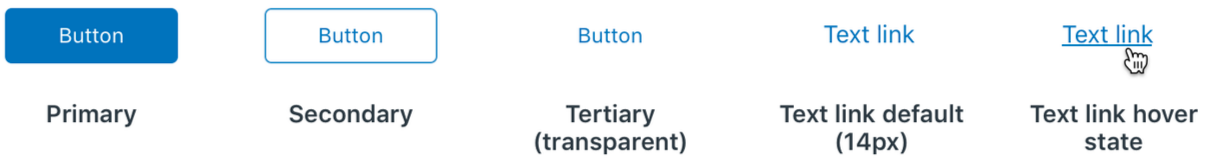


Don't do this

We never want placeholder text or input fields to wrap to a second line for fields that are meant to fit single lines of text.

Primary and secondary buttons

When should you use a Primary (`solid`) button vs a secondary (`outline`) button? In general there should just be 1-2 primary action buttons at a time; the rest should be secondary or tertiary.



Primary buttons

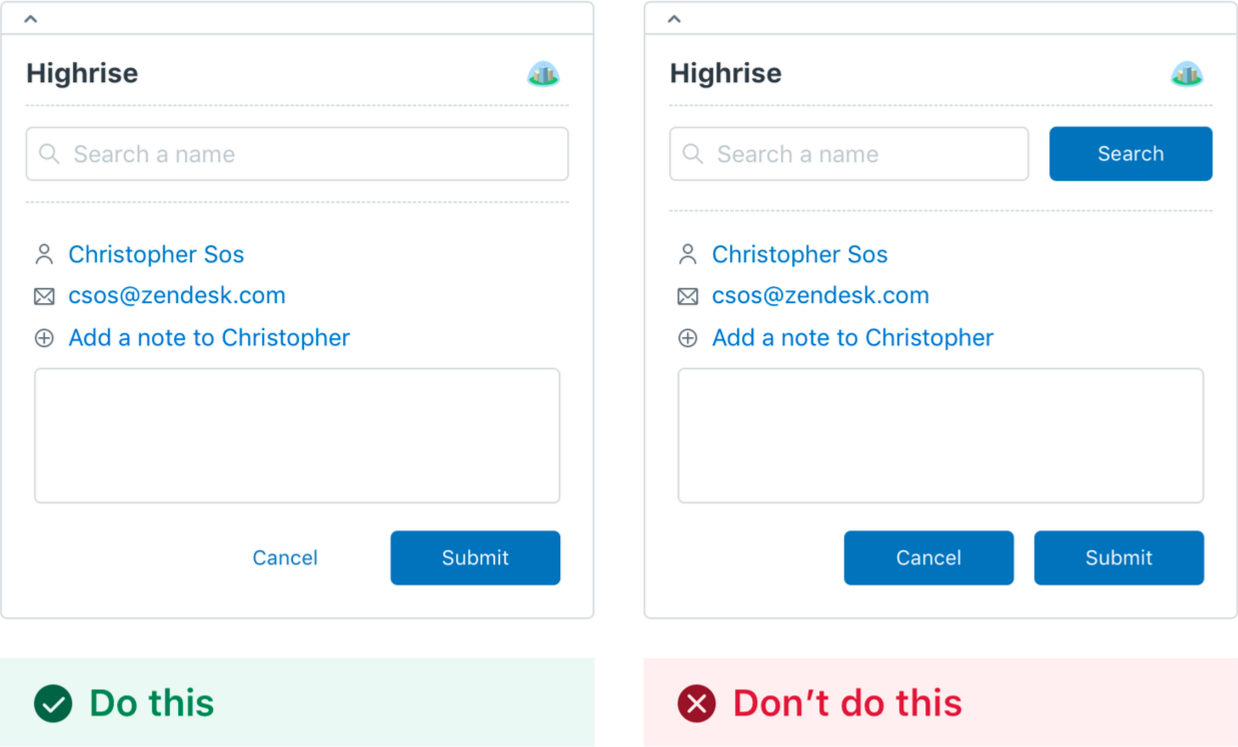
Primary buttons are used for any main actions or confirmation actions such as `Submit`, `Send`, and `Search`. It is the main action the user should take. There shouldn't be too many primary buttons occurring in the same page, or in this case, the app container, because using the visual weight across all buttons might hinder the user from quickly seeing what the main action is.

Secondary buttons

Secondary buttons are used for other actions that are not as important as the main action. It can be used when you want the main action to keep the strongest visual weight on the page.

Tertiary buttons
We use tertiary buttons for all `Cancel` buttons, as well as for any actions that are of even lesser importance than secondary buttons.

Text links
These are not buttons, but you can use text links for data that links out to external browsers, tables of data, and in cases where using buttons don't make sense. This includes actions such as `Show more`, `Show less`. Text links should be underlined on hover.

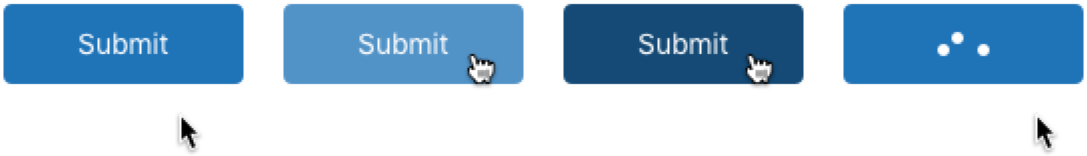


Only the main action(s) should be primary. Avoid too many primary buttons on the same screen. The "Search" button is in the image for demo purposes only; the Garden search bar doesn't include a button.

In the example above, you can see that we use the Tertiary button for the `Cancel` action. In this case, it is also OK to use the primary button for both `Search` and `Submit` because they are the main actions for each use case - `Search` for entering a query in the search bar, and `Submit` for adding a note.

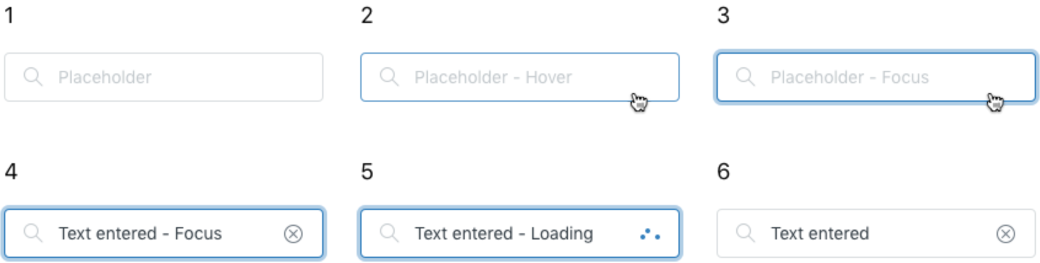
Loading states

Buttons should use the 3-dot loader from Garden, which replaces the text inside the button when loading.



Button flow of when loading should occur for buttons.

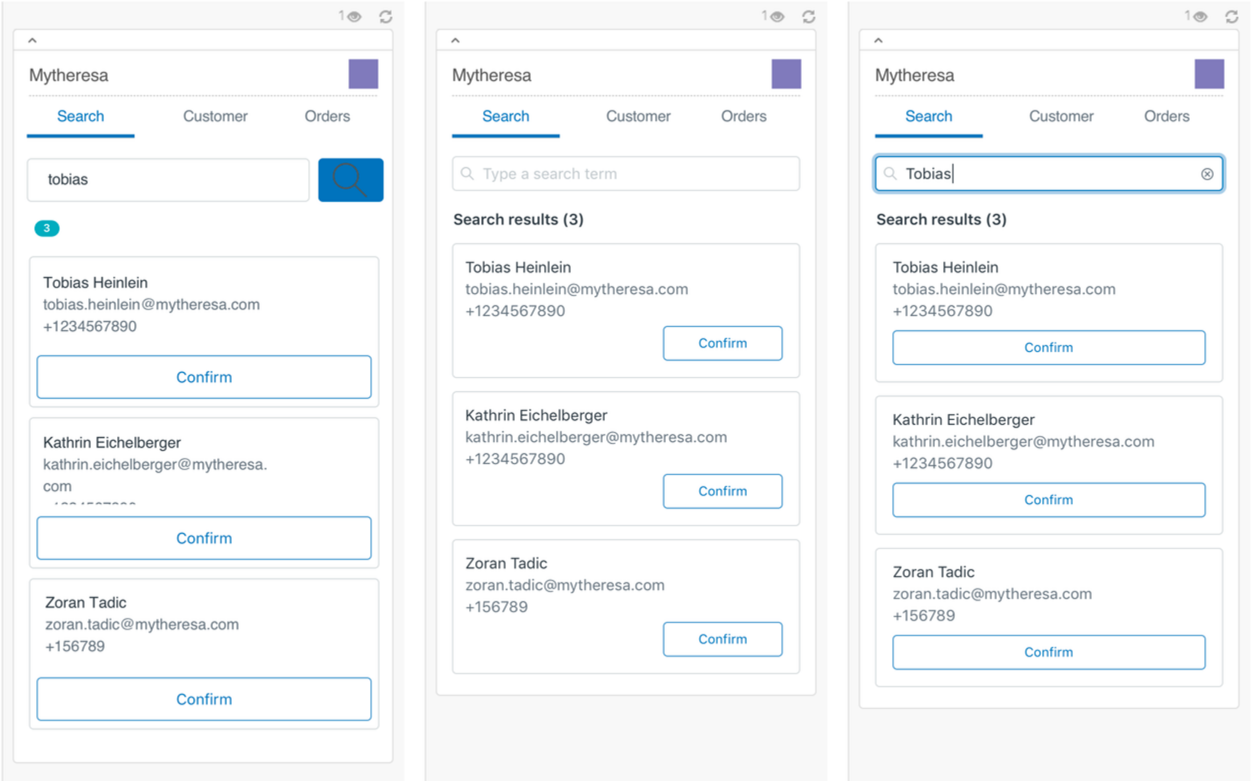
Search bars from Garden do not have an explicit “Search” button and the 3-dot loader appears on the right-hand side of the input field when loading search results.



An example of when loaders kick in for search.

Placement

In Zendesk, we normally place the primary button on the right-hand side (to the right side of the container), and `Cancel` buttons to the left of the primary button. If there is only a single button, it can span the entire app width, but it’s up to your discretion whether it is necessary.



Original

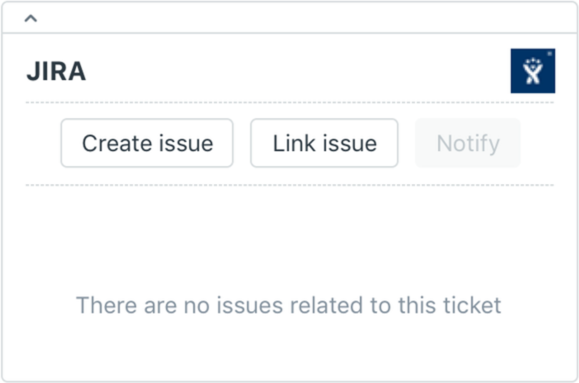
Small button anchored to right

Small button spanning width of app

You can see the amount of vertical space saved by using "small" size buttons and fields. In this case, it's fine to have the button span the entire container, otherwise there's a lot of empty space on the left. The difference in vertical height saved when compared to the implementation in the middle image is minimal.

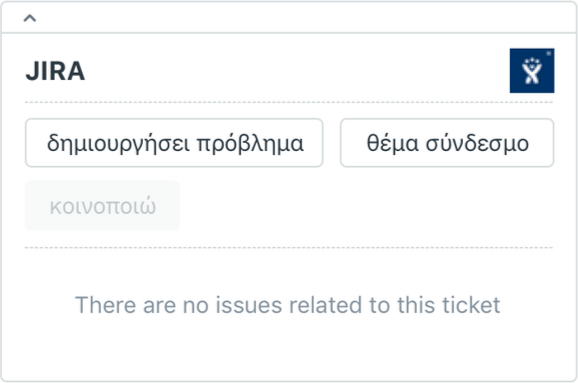
Button wrapping

Try to avoid using too many buttons together in a row because this will result in wrapping for localization, which could disrupt the rest of your app content below. It doesn't mean you can't do this, but just be aware that multiple buttons in a row buttons might wrap.



There are no issues related to this ticket

English language



There are no issues related to this ticket

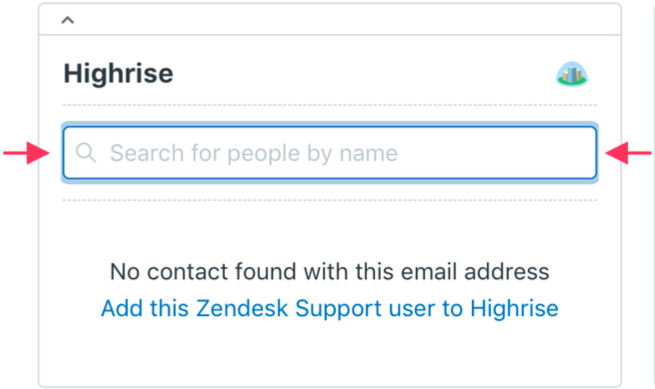
Greek language

Note: This app is old so the button colors are wrong. We have a "Muted" button in Garden if you really need a neutral color, but we should use the default blue-600 whenever possible.

Likewise, for long strings, consider using a text link instead of a button because a button should never wrap to accomodate more than one line of text.

Iframe issues

Sometimes the input field focus state will cut off from the left and right due to iframe issues, if this is the case, then increase the left and right padding by a few pixels until it no longer cuts off.

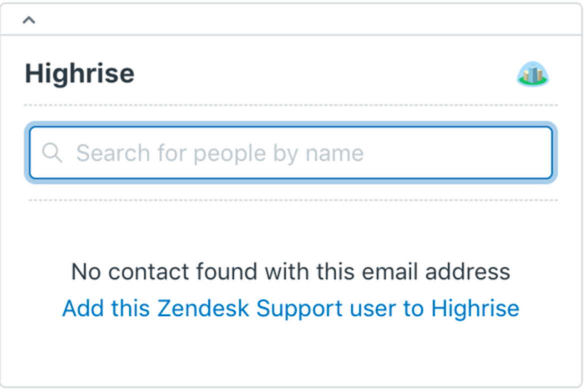


Highrise

Search for people by name

No contact found with this email address

Add this Zendesk Support user to Highrise



Highrise

Search for people by name

No contact found with this email address

Add this Zendesk Support user to Highrise

✗ Don’t do this

Sometimes the app iframe will cut off the focus state for input fields. Don’t leave it like this :(

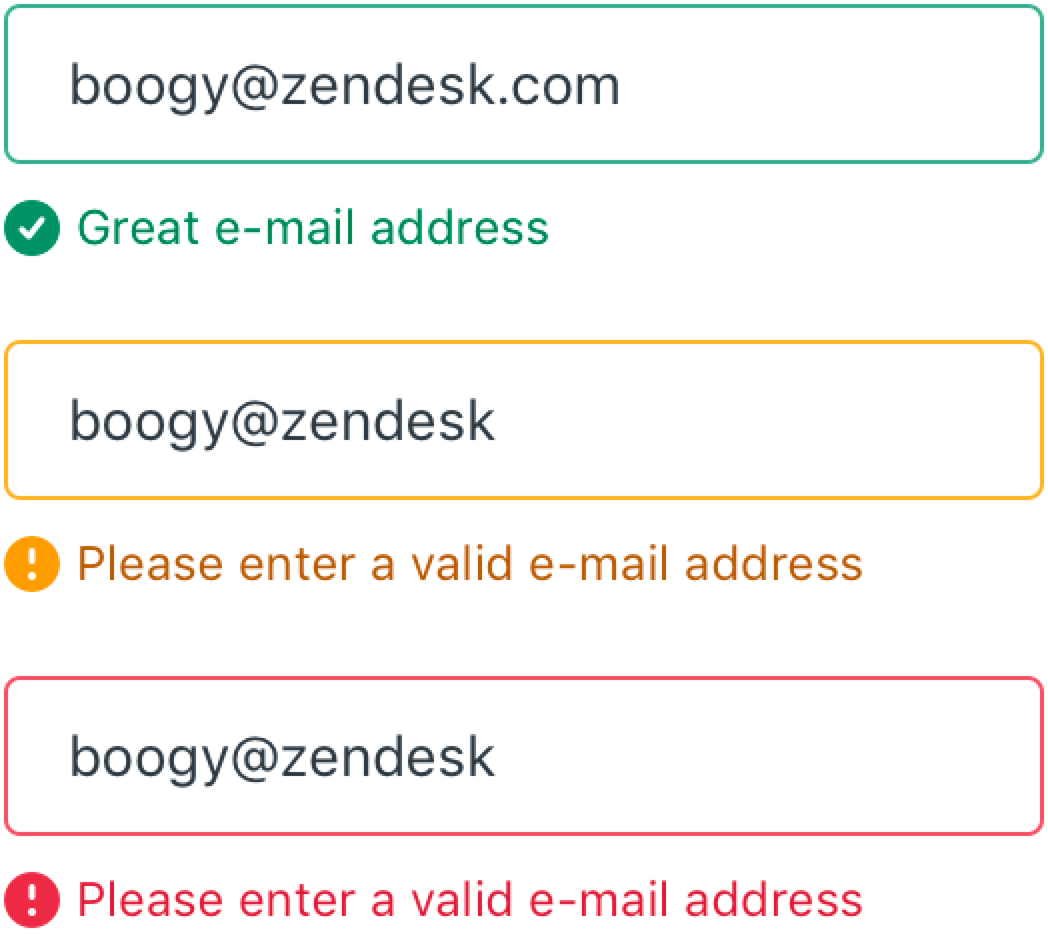
✓ Do this

To fix this, you’ll have to increase the left/right padding evenly so the focus state is no longer cut off.

Make sure other content below it such as dividers and other input fields follow the same padding so you don’t end up with misaligned elements.

Errors

There’s a major difference between form validations and warnings. Form **validations enforce** a set of rules and won’t allow the user to proceed, while **warnings alert** the user about possible problems but will allow them to proceed.



boogy@zendesk.com

✓ Great e-mail address

boogy@zendesk

! Please enter a valid e-mail address

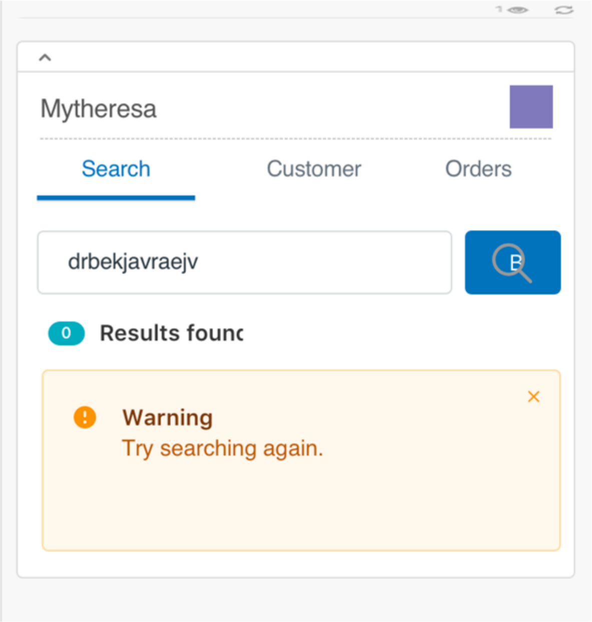
boogy@zendesk

! Please enter a valid e-mail address

Examples of input field validations. It’s rare that we need to use this for app design.

The Garden `Well` component has quite a lot of padding; we should avoid using error messages unless absolutely necessary.

Before

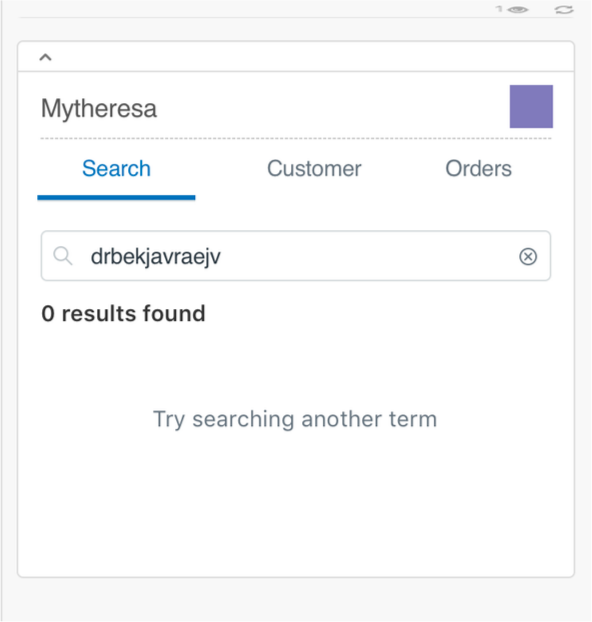


drbekjavraejv

0 Results founc

Warning
Try searching again.

After



drbekjavraejv

0 results found

Try searching another term

✕ Don't do this

Warning and Error wells are for extremely important or fatal notifications. They also have a lot of padding so try to avoid using them in a small app container.

✓ Do this

For something like no search results, we can simply use a label for no results and some helpful hint text.

Empty states

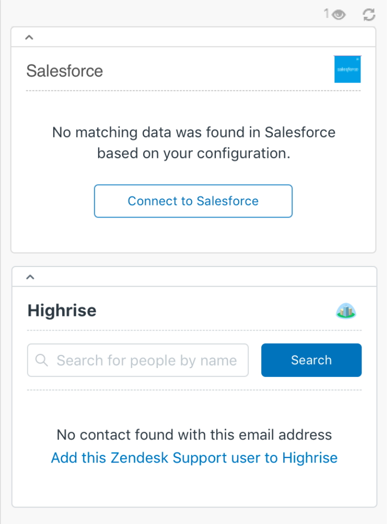
Sometimes apps have empty states if there's no data. The best approach to this is to use plain text and simple strings to describe the empty state, and provide an action for the user to get that data if possible.

Icons, Colors and Typography

Original Garden doc; summary below:

Utility Icons (16x16px/12x12px)

The most flexible collection that is widely used in toolbars, buttons, or visual supplements to text labels. **16px icons** are considered the default size to use within Zendesk products (toolbars, icons w/ text, buttons) where as **12px** are used for more compacted designs (essential cards). The default weight for icons **is stroke, not fill**. 12px icons may be used @2x (24px) as well.



Salesforce

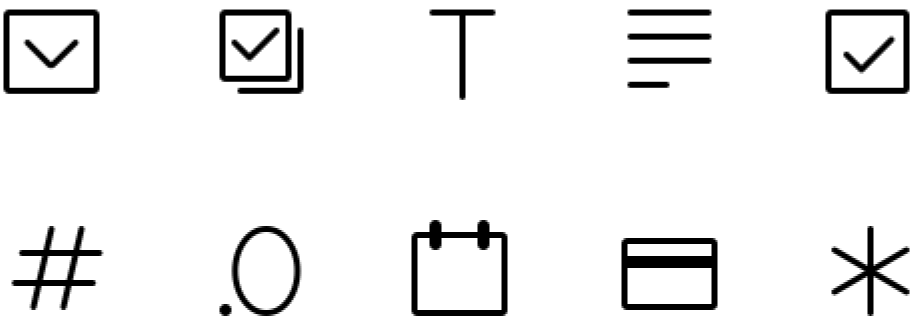
No matching data was found in Salesforce based on your configuration.

Highrise

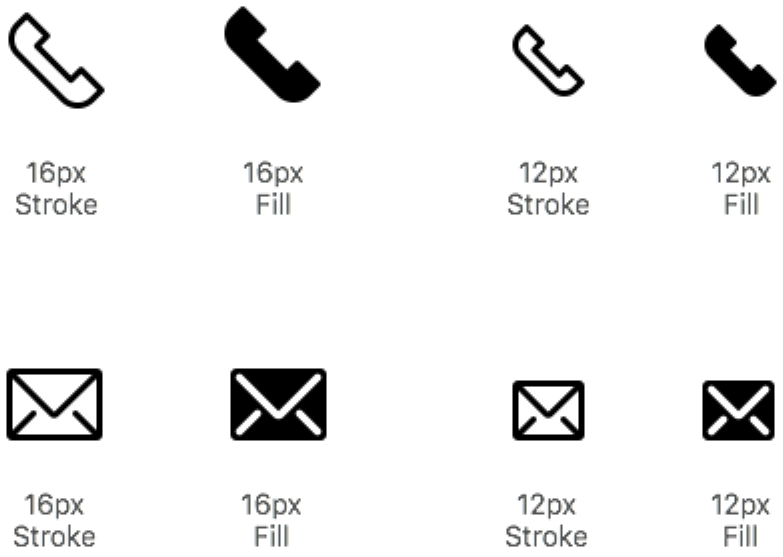
No contact found with this email address

Add this Zendesk Support user to Highrise

Empty state examples. The Highrise app uses a text link because the string is too long to fit in a button.



Both sets come in two different styles as well: **stroke and fill.** These styles are meant either show an action with the icon (ie: selected vs unselected or normal vs push state) or to help strengthen the visibility of an icon.

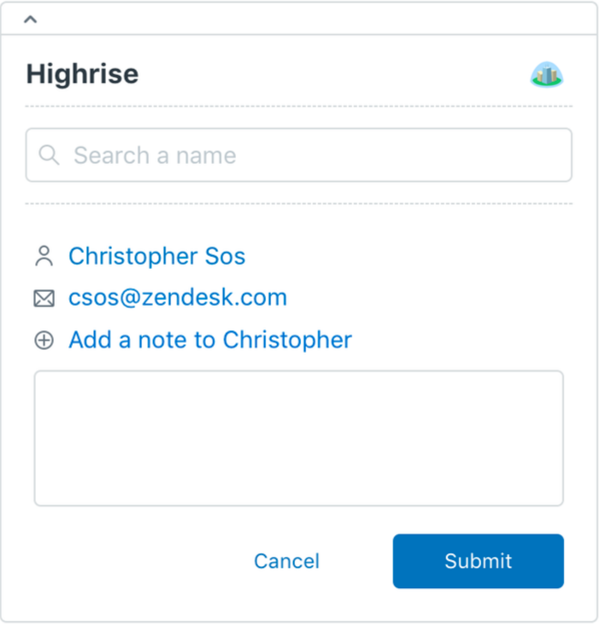


Avoid resizing, but if you must, do it by @1x, @2x, and @3x. Use Scale (`cmd` + `K`) in Sketch. This will properly scale the shapes and strokes.

Icons in Apps

Use the appropriate 12px or 16px `stroke` icons in your app designs; you will rarely need to use larger-sized icons. Try to avoid multiple icon sizes in a single app and keep icon size consistent.

If icons are meant to be actionable/clickable, then use the Garden `iconButton` component. These should have a default and hover state (refer to Color and Typography table below). If they are just for visual purposes, then use the icon as-is with a default of `gray-600`.



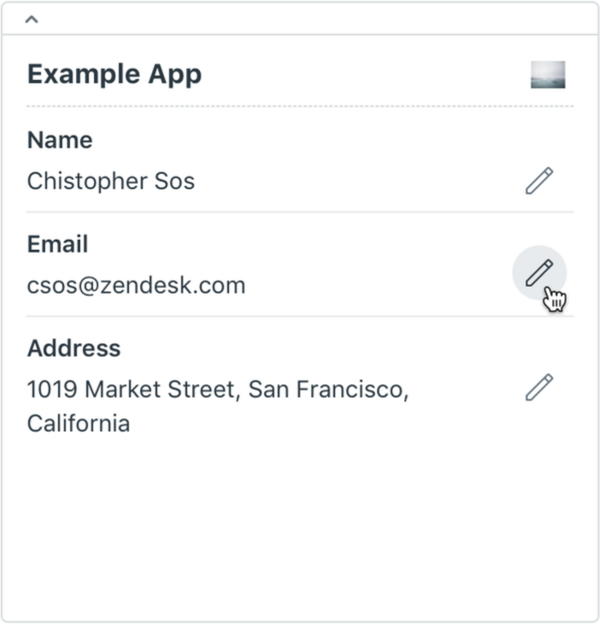
Highrise

Search a name

Christopher Sos

csos@zendesk.com

Add a note to Christopher



Example App

Name
Chistopher Sos

Email
csos@zendesk.com

Address
1019 Market Street, San Francisco,
California

The icons in this app aren't actionable/clickable so we keep them at gray-600

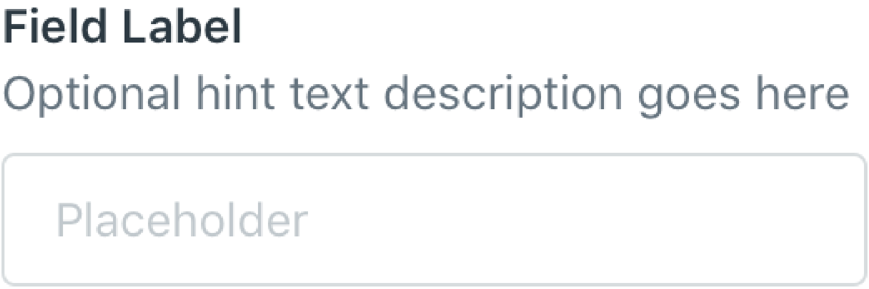
The icons in this app are actionable/clickable so we keep use the Garden iconButton component. These icons should have a default and hover/active state.

Colors and Typography

If you follow and use Garden component guidelines, you should be fine here. All text should be using the font-family SF font/system font at 14px with a 20px line-height. Any bolding should use `semi-bold`, not `bold`. Below is a quick overview:

Component	**Color**
Typography	SF Font 14 px; `line-height`: 20px
Labels (semi-bold)	`gray-800`
Hint text (You can also use this for empty state text if necessary)	`gray-600`
Base text	`gray-800`
Placeholder text	`gray-400`
Buttons (primary/secondary/tertiary out-	`blue-600`

of-the-box from Garden)	
Text links	`blue-600`
Search icon inside input field	Default/inactive - `gray-400` Hover/active/focus - `gray-600`
All other icons IF the icon is actionable/clickable, then you can add a hover/active/focus state; if not, just keep it at the default color	Default - `gray-600` Hover/active/focus `gray-700`
Errors	Default Garden `error` states (looks red)
Warning	Default Garden `warning` states (looks yellow)

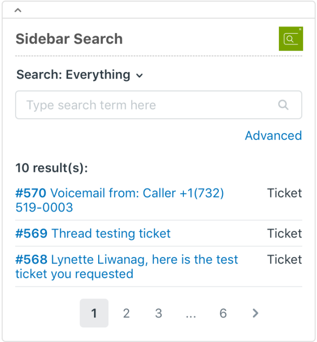


Field Label

Optional hint text description goes here

Placeholder

Label, hint text, and placeholder text example



Sidebar Search

Search: Everything

Type search term here

Advanced

10 result(s):

#570 Voicemail from: Caller +1(732) 519-0003 Ticket

#569 Thread testing ticket Ticket

#568 Lynette Liwanag, here is the test ticket you requested Ticket

"Ticket" is a base text example; ticket names are a text link example

Lists

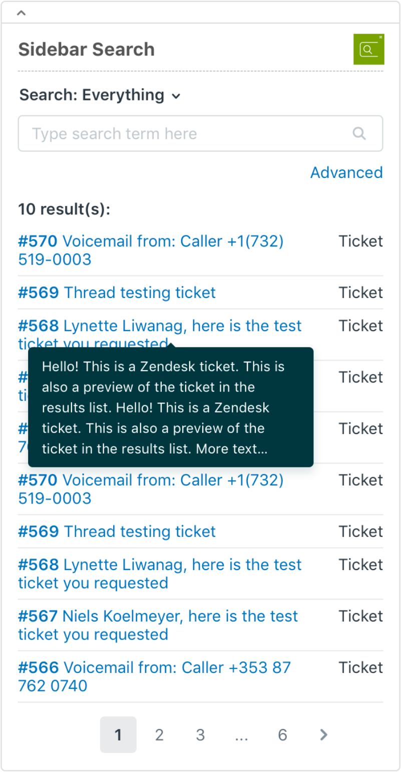
Oftentimes apps contain lists of data. Below are some guidelines for displaying this data in an optimal way.

Columns and Rows

It's **not** recommended to use more than **two columns** for data due to the app container width, and because of localization issues that might occur. You can technically have as many rows as you want but to cap the app height, you should paginate (see below).

Paginate for long lists

If lists can get long, it’s recommended to paginate so that users can sort through the data easily, and so the app doesn’t become massively long.



Sidebar Search

Search: Everything

Type search term here

Advanced

10 result(s):

#570 Voicemail from: Caller +1(732) 519-0003	Ticket
#569 Thread testing ticket	Ticket
#568 Lynette Liwanag, here is the test ticket you requested	Ticket
# ti	Ticket
# 7	Ticket
#570 Voicemail from: Caller +1(732) 519-0003	Ticket
#569 Thread testing ticket	Ticket
#568 Lynette Liwanag, here is the test ticket you requested	Ticket
#567 Niels Koelmeyer, here is the test ticket you requested	Ticket
#566 Voicemail from: Caller +353 87 762 0740	Ticket

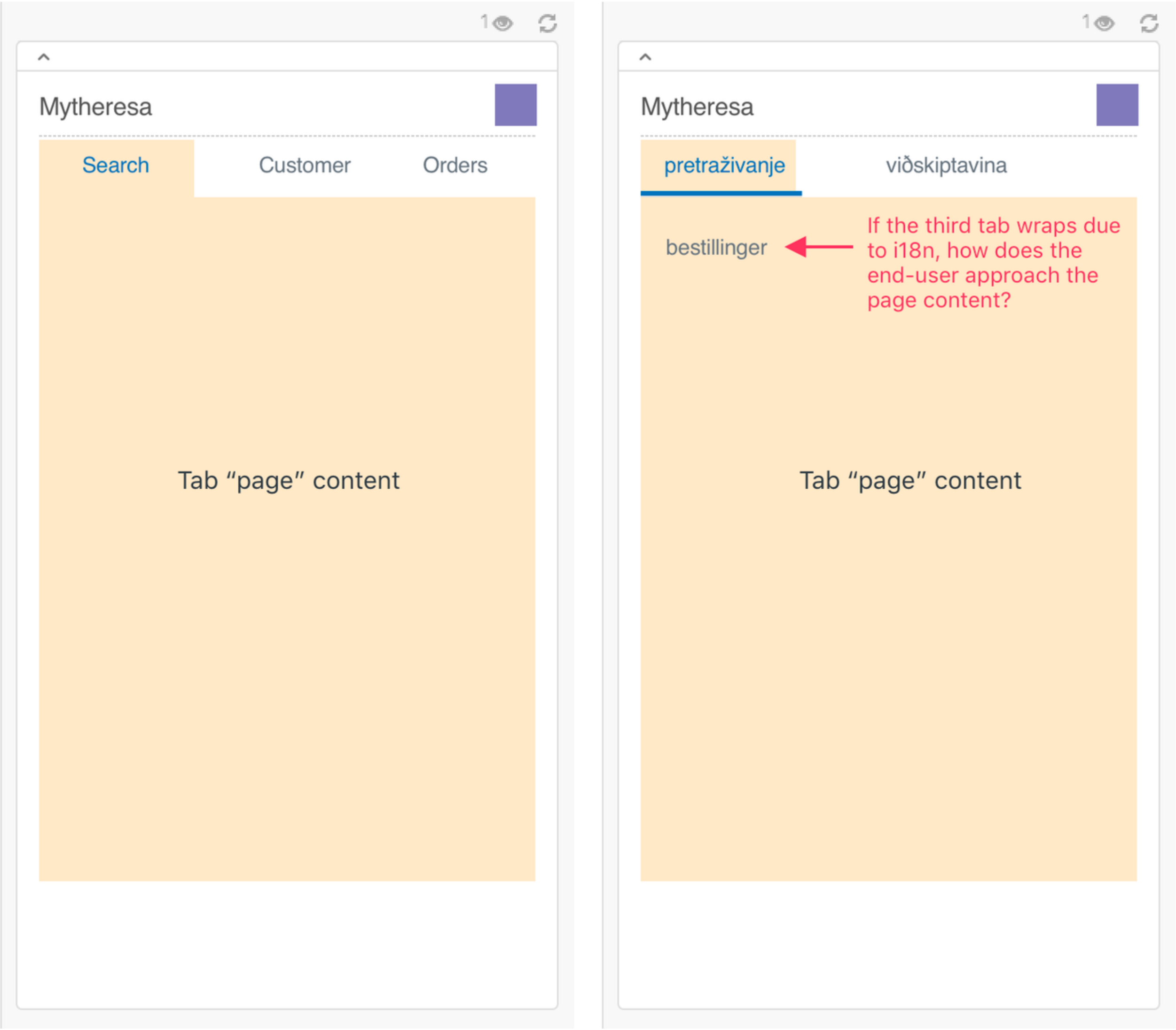
Hello! This is a Zendesk ticket. This is also a preview of the ticket in the results list. Hello! This is a Zendesk ticket. This is also a preview of the ticket in the results list. More text...

Example of data list that contains two columns, multiple rows, pagination, and text links.

Tabs

It’s not recommended to use tabs in app containers due to lack of space width-wise. Tabs also often cause wrapping issues for localization when used in width-restricted spaces.

If a tab wraps, we run into issues with the relationship between the content being displayed and the active/inactive tabs.

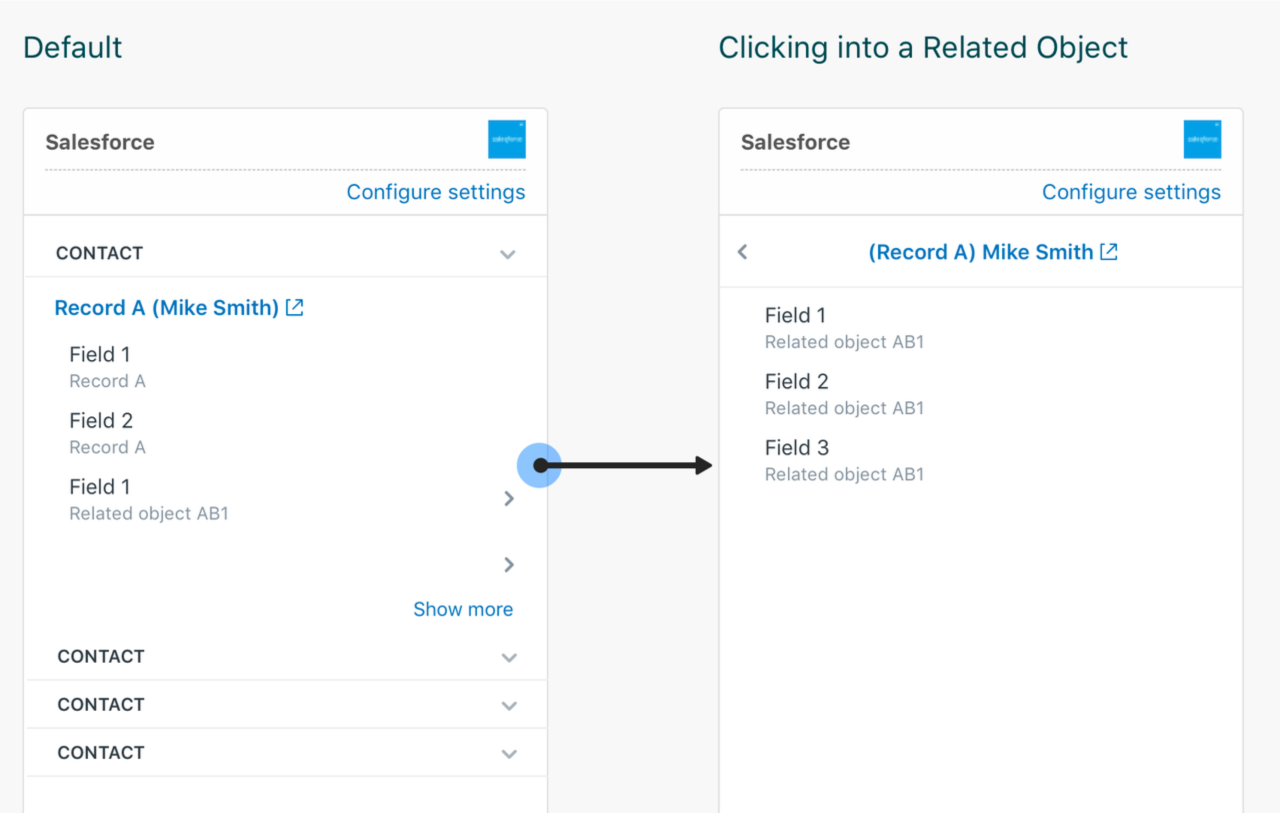


If the third tab wraps due to i18n, how does the end-user approach the page content?

If tabs fit using the English language and the app doesn’t need to be localized, then it could work fine.

Alternative solutions

There is not really a better alternative to replacing tabs in apps currently, but you could look into using buttons or text links to open a completely new screen in the app container and provide a `Back` button to return to the previous screen.



Default

Clicking into a Related Object

Field 1
Record A

Field 2
Record A

Field 1
Related object AB1

Field 1
Related object AB1

Field 2
Related object AB1

Field 3
Related object AB1

Another option could be using buttons to open the information in a modal if it makes sense for what you're trying to do, eg. you need a large space to input a lot of information.

Other

There will be other components that you might need to use that Garden doesn't provide or that we don't have any set patterns around. The best thing to do is to reach out to me via Slack if you have any questions. Provide screenshots and as much context as possible and I can try to give advice on the best approach so that it fits in with our patterns.