
InQuira UX Patterns

December 2014

Written by Theresa Wilkinson

Table of Contents

Interactions.....	3
Overview	3
Invitations.....	4
Hover Help	7
Tooltip Invitations.....	10
Navigation.....	15
Overview	15
Why Should We Care About Navigation if We Have Search?	15
Global Navigation	18
Fly-Out Menu	19
Breadcrumbs	21
Persistent (Sticky) Menu.....	24
Page Up Control	26
Search.....	28
Users of Search	28
Why Should We Care About Navigation if We Have Search?	29
Action Links	31
Alternate Row Color.....	34
Application Search.....	35
Display Quantity	37
Facet Filtering	39
Input Prompt	43
Pagination.....	44
Primary & Secondary Actions	47
Progressive Disclosure.....	50
(Search) Results List	53
Search Filters	55
UI Controls.....	57
Accordions.....	57
Carousel	59
Disclosure Arrows.....	62
Hyperlinks.....	64
List Builder	66
Module Tabs	69
Sidebar Toggle	71
Standard Icons	73
Tables.....	75
Appendix A	78
User Experience Overview.....	78

Interactions

Overview

The idea behind an invitation is using as many avenues as possible to get the user to discover the feature through an interaction like a mouse-over or hover.

Invitations are good when:

- You want to lead a user to discover a new feature.
- You want to introduce the user to a new way to interact.
- The actual interaction engagement you are driving towards is the next logical event – hover leads to a click or a click leads to another click.

Accessibility

Since invitations are about visual communication they are, by default, not accessible by users with visual impairments. To make the invitation accessible:

- Set the focus on the item that was changed. This enables assistive technologies to detect where a change happened. If the user is typing or needs the focus on another area, then this solution cannot be used.
- **Provide ALT tags that preview what will happen when the user engages an interaction.** For example, if you can close a panel and provide a way to communicate where the panel is collapsed to, you might consider putting text in the ALT tag for the image that tells the user what will happen when the click/activate the close icon and how they can get the panel open again.

Capitalization

Element	Capitalization Style	Examples
Content	Sentence case: Capitalize first word of each sentence Not bold Use ending punctuation	Members may have coverage that provides reimbursement for the replacement of keys, key fobs, or locksmith services already performed. This coverage may originate from

		RSA Basic or Plus coverage, EXABLO coverage, or On Your Side Rewards.
Hover Help	<p>Sentence case: Capitalize first word of each sentence</p> <p>Not bold</p> <p>Use ending punctuation</p>	Some customers have additional lockout coverage under the Extra Coverage endorsement (EXABLO) Emergency Lockout up to \$150.00.
Input Prompts	<p>Sentence case</p> <p>Gray font color</p> <p>Italics</p> <p>No ending punctuation</p>	<p><i>Enter keywords</i></p> <p><i>Add a category name</i></p>
Tooltip Invitations	<p>On a button or icon:</p> <p>Use initial cap</p> <p>Not bold</p> <p>No ending punctuation</p> <p>On a statement (text):</p> <p>Sentence case</p> <p>Not bold</p> <p>No ending punctuation</p>	<p>Limit to</p> <p>Click to expand field</p> <p>Search</p> <p>View fewer companies</p> <p>Hover to display screenshot</p>

Invitations

What Problem Does This Solve?

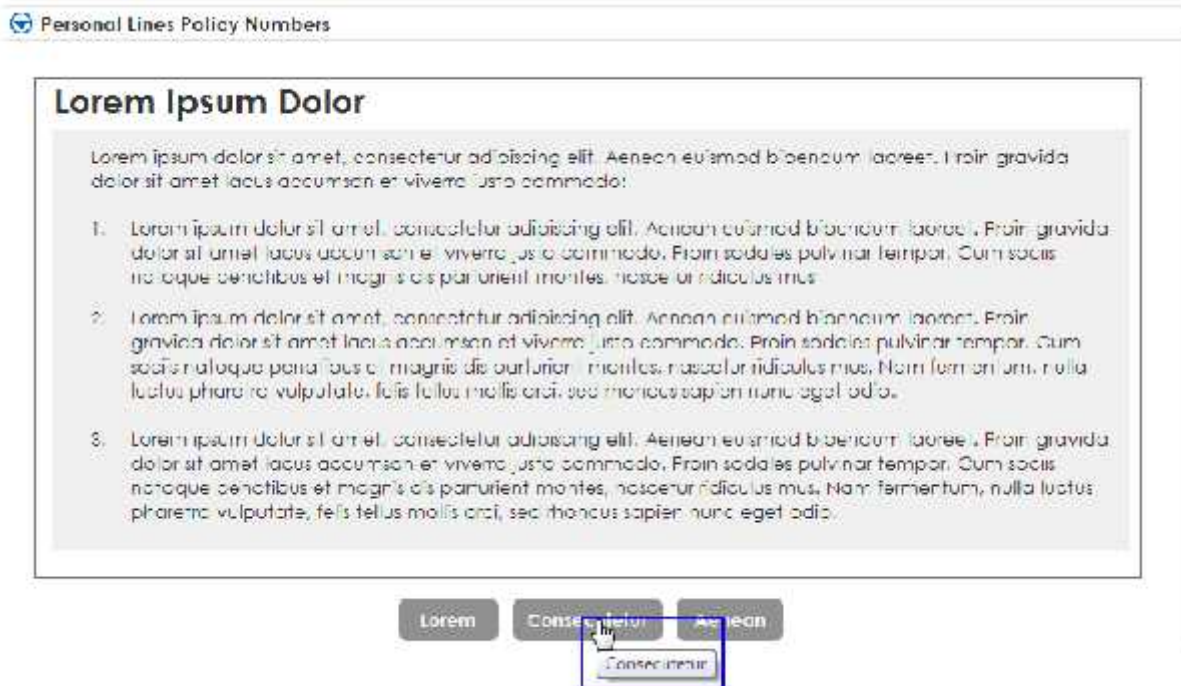
User needs a cue that an object can be interacted with.

Solution

- Change the cursor to indicate the operation. For example, to indicate a drag, change the cursor to a hand or arrow.
- Change the cursor when the user moves over the object with which they will interact.
- When the operation is initiated with a click or drag.

Interaction – Hand Cursor

- User hovers over button. Cursor changes to the hand cursor. Tooltip displays.
- User clicks button.
- Carousel advances to next page.



Interaction – Arrow Cursor

- User wants to view more alerts. Cursor changes to the arrow cursor.
- User clicks pagination.
- Next page of alert results displays.

Welcome



Interaction – Edit Cursor

- User wants to perform a search. User hovers over desired field. Cursor changes to edit cursor.
- User clicks into field and enters search term.
- User clicks Search button. (Search) Results List displays.



Rationale

- The idea behind an invitation is using as many avenues as possible to get the user to discover the feature through an interaction like a mouse-over or hover.

Invitations are good when:

- You want to lead a user to discover a new feature.
- You want to introduce the user to a new way to interact.
- The actual interaction engagement you are driving towards is the next logical event – hover leads to a click or a click leads to another click.

When to Use This Pattern

- The operation is initiated with hovering over target (dotted underline).
- The interaction will be an operation you perform on the object.
- The operation is not easily previewed.
- It is not necessary to preview the operation.
- You want as lightweight of an invitation as possible.

Accessibility

Since invitations are about visual communication they, by default, are not accessible by users with visual impairments. To make the invitation accessible:

- Set the focus on the item that was changed. This enables assistive technologies to detect where a change happened. If the user is typing or needs the focus on another area, then this solution cannot be used.
- **Provide ALT tags that preview what will happen when the user engages an interaction.** For example, if you can close a panel and provide a way to communicate where the panel is collapsed to, you might consider putting text in the ALT tag for the image that tells the user what will happen when the click/activate the close icon and how they can get the panel open again.

See also:

- [Carousel](#)
- [Pagination](#)
- [Standard Icons](#)
- [Tooltip Invitations](#)

Hover Help

What Problem Does this Solve?

More information about an item exists, but it does not make sense to clutter the page with more than just primary information or have the user navigate away from the item.

Solution

- A word or phrase requiring extra information can be displayed as a link with a dotted underline.
- Provide hover help if users can benefit from extra information related to a word or phrase.
- This information may be a definition, example, or both, depending on the word or term.
- Users can skip over this extra information if they don't need it, and actively seek it if they want more details.

Example

Help Text

If the claim is taken in ClaimCenter and is filed from a lead, take a claim on the policy and policies. If vehicles are on more than one policy, Take a claim on each policy involved.

Use this workflow when... Some customers have additional lockout coverage under the Extra Coverage endorsement (EXABLO) Emergency Lockout up to \$150.00.

Centralized Fire and Theft claims area handles key replacements and reimbursement. In some cases Agera is able to cut keys when service is dispatched; additional fees may apply. Commercial and Specialty Auto (Titan) customers **may only submit reimbursements by mail**.

Screenshot



Interaction

- User hovers over an underlined word. Cursor changes to an arrow or hand cursor.
- Additional information displays in a hover.

Rationale

There are two key principles at work here:

1. First and most important is that showing the additional data in context of the higher-level items maintains the context so users can consider it easily within that broader context (without mentally leaving the higher context that a full pop-up window or navigation to a new page would cause).
2. The second aspect is the triggering based on hover. Hovering is less committing than selecting. It's like a tentative I-am-curious-about-this-item-but-not-committed-to-it activity, like browsing the items in a shopping aisle versus picking an item up off the shelf. As such, showing additional information, even if you have more to show upon selection, is a nice experience. Also, seasoned computer users will be familiar with the idea of tooltips and hovering in this way, so if you can provide them, they are usually appreciated.

By the same token, the information displayed needs to be easily consumable. The user wants to touch their toe into the water before jumping in, so don't overload hover help—display just enough of that extra information to provide value to further help them decide if they want to take the plunge.

When to Use This Pattern

- You are displaying items for which you have additional information beyond what is displayed, such as a chart, overview table, etc.
- You have additional information that would be useful for users to have easy access to without losing their context.
- The additional information is consumable in a small window format.

- The interface recognizes that a user is hovering (such as with a mouse, stylus, finger, or other pointing device) so it knows when to show the additional information and for what.

Use hover help for:

- Defining words
- Providing examples like screenshots
- Content normally found in parentheses
- Additional information that is not crucial, but may be helpful and ensures consistency of use

See also:

- [Invitations](#)
- [Tooltip Invitations](#)

Tooltip Invitations

A message that appears when a cursor is positioned over an icon, image, hyperlink, or other element in a graphical user interface.

What Problem Does This Solve?

- User needs a cue about what will happen if they click the mouse on the object.
- User does not understand an object and it is not described in the UI.

Solution

- When the invitation to interact with an object would benefit from a written description, show a tooltip when the mouse cursor hovers over the target area.
- When an object or unlabeled control would benefit from a written description, show a tooltip when the mouse cursor hovers over the target area.
- The tooltip copy should be a short verbal phrase that calls the user to action, such as “Click to expand” or “Search”.

- Default delay on tooltip is 250ms when the mouse hovers of the interaction area.
- Keep the tooltip visible the whole time the user has the mouse over the interaction area and remove it promptly when the mouse leaves the interaction area.
- Display tips only as the result of the user interaction—never display them on their own.
- Tooltips may also provide more detail for [labeled buttons](#) if doing so is helpful. Don't just repeat or give a wordy restatement of what is already in the label.
- Always place the tooltip on the side of the object, even if that requires separation between the pointer and the tip.
- Avoid covering any object that the user is likely to view or interact with.

Examples

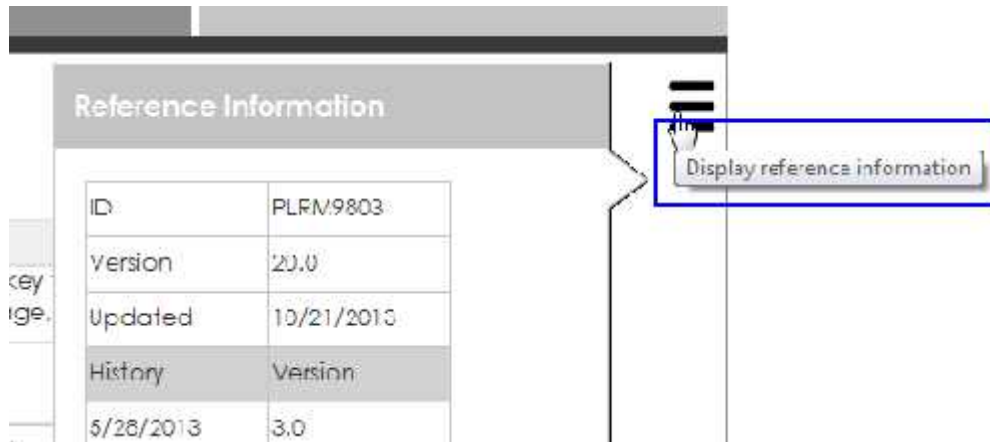
Tooltip on Accordion



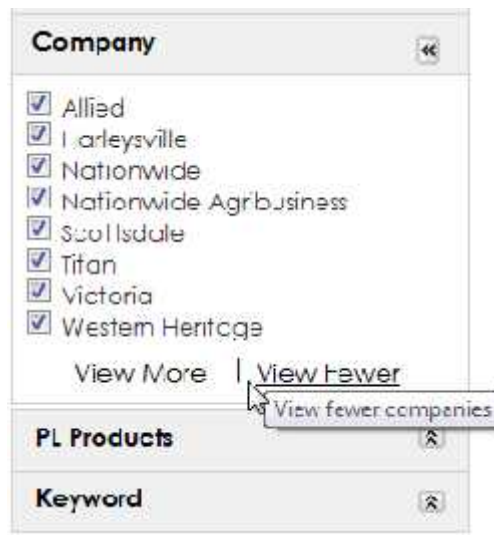
Tooltip on Button



Tooltip on Icon



Tooltip on Action Link



Labels

Element	Capitalization Style	Examples
Tooltips	<ul style="list-style-type: none"> On a button or icon: Use initial cap Not bold No ending punctuation 	<ul style="list-style-type: none"> Click to collapse Hide sidebar Display reference information

	On a statement (text) or Action link: Use sentence capitalization Not bold No ending punctuation	View fewer companies View more companies Hover to display screenshot
--	--	--

Suggestions

- Keep the tooltip text concise. Tooltips are perfect for short sentences and sentence fragments.
- Create helpful, supplemental tooltip text. Tooltip text must be informative. Don't make it obvious or just repeat what is already on the screen.
- Because tooltip text isn't always visible, it should be supplemental info that users don't have to read.
- Communicate important info using self-explanatory control labels or in-place supplemental text.
- Don't use a tooltip to display text already visible in the UI.
- Don't put interactive controls inside the tooltip.
- Don't change the offset values.
- Don't change the target relative positioning.

Interaction

- User mouse-overs an icon.
- Tooltip displays.

Rationale

- The idea behind an invitation is using as many avenues as possible to get the user to discover the feature through an interaction like a mouse-over or hover.
- Use tooltips sparingly. Tooltips are an interruption. A tooltip can be as distracting as a pop-up, so don't use them unless they add significant value.
- Use a tooltip to reveal more info about a control before asking the user to perform an action.

Invitations are good when:

- You want to lead a user to discover a new feature.
- You want to introduce the user to a new way to interact.
- The actual interaction engagement you are driving towards is the next logical event – hover leads to a click or a click leads to another click.

Accessibility

Since invitations are about visual communication they by default are not accessible by users with visual impairments. To make the invitation accessible:

- Set the focus on the item that was changed. This enables assistive technologies to detect where a change happened. If the user is typing or needs the focus on another area, then this solution cannot be used.
- **Provide ALT tags that preview what will happen when the user engages an interaction.** For example, if you can close a panel and provide a way to communicate where the panel is collapsed to, you might consider putting text in the ALT tag for the image that tells the user what will happen when the click/activate the close icon and how they can get the panel open again.

When to Use This Pattern

- You want to invite the user to click or interact with the object being hovered over.
- You want to make it clear that something will happen when the user clicks on the object being hovered over.
- The user is interacting directly with the object.
- You need a text description to make it explicit what will happen when the user interacts with the object.
- You want to entice the user to interact with the feature.
- If the control does not have a label, use a tooltip.
- Toolbar controls and command buttons with graphic labels need tooltips.

See also:

- [Accordions](#)
- [Disclosure Arrows](#)
- [Pagination](#)
- [Primary & Secondary Actions](#)
- [Sidebar Toggle](#)
- [Standard Icons](#)

Navigation

Navigation means using the available options and contextual clues to locate content. Buttons, tabs, tables of contents, menus, links, and indexes are common navigation options online. In print, readers navigate with the help of tables of content, indices, and page numbers.

Search is the act of looking for specific content by entering a query in a search engine or application. Users form queries with keywords or search terms, and the search engine displays the query results based on its index and understanding of the request.

Findability is a term for the ease with which information contained on a website or application can be discovered and retrieved through searching or browsing.

Overview

As Hansel and Gretel fairy tales suggest, getting lost is often a bad thing. It is associated with confusion, frustration, anger and fear. In response to this danger, tools have been developed to prevent users from getting lost. Navigation tools are used to chart the user's course, determine their position, and find a way back.

Why Should We Care About Navigation if We Have Search?

Here are five reasons why websites and applications cannot expect users to rely on search only:

1. Search Requires Knowledge of the Search Space

In order to formulate a good search query, users need to know fairly well what they are searching for. They need to understand the search space and put in the right keywords. Sometimes that is easy — for instance, most Help Desk CSRs can quickly come up with a reasonably good query. But when a novice CSR is trying to use InQuira, he is likely to need some guidance in the search space and calls the Help Desk.

Category landing pages (Services, PL FNOL) often have navigation that helps the user quickly get a mental model of the search space: what subcategories are available and what attributes are relevant for that particular category.

2. Search Increases [Memory Load](#)

Even when users are familiar with the search space, **search requires them to recall information from their memory**. To come up with a meaningful query, a user needs to think about attributes that are relevant for his goal and incorporate them in the query.

Navigation replaces the recall with recognition: instead of forcing users to come up with a complex query, they can recall only a minimum and then use recognition to augment their query with relevant terms.

If search was the only tool available, you would have to come up with your own constraints and then translate them into terms that are likely to be relevant on this application.

Thus, you attempt to retrieve from memory query terms matching that goal. Translating the user goal into a search query that is understandable by the site is no easy task (and in fact people often fail at it — see below) and **imposes a higher cognitive load** than simply selecting the right link from the page.

3. Search Has Higher Interaction Cost than Browsing

Search forces users to work more because not only must they come up with a query, but they also must type it. Typing has high interaction cost; it is error prone and time consuming even with a full keyboard (and even more so on a touch screen). The interaction cost of search is greater than that of browsing. True, some of that interaction cost is mitigated by auto-suggestions,

but auto-suggestions usually work well when the query is fairly common and simple.

4. Search Often Works Poorly

Even if users are able to come up with a reasonable search string, chances are that the results returned will often be irrelevant. Big search engines have behind them complex algorithms and a huge database to search in; even if the query is not perfectly formulated, chances are that some relevant results will still be found. That is often not the case on specialized sites or applications.

5. Users Have Poor Search Skills and Don't Know How Search Works

A while back Jakob Nielsen called users' search skills "incompetent". It is still true. We keep seeing this over and over again: **people have no understanding of what makes a good search query**. Their search mental model is corrupted by the big search engines and **they expect search to work in the same way on every site**. Some sites attempt to do some parsing of the search string and map it onto their attribute space, but most of the time that mapping is very poor resulting in poor search results.

Site search is vital and can save the day for those users who have well defined goals and a good understanding of the information space in which they are searching. However, navigation serves important functions:

1. It shows users what they can find in the application.
2. Teaches them about the structure of the search space.

Using the navigation categories is often faster and easier for users, than generating a good search query. Plus, many times site search does not work well or requires users to have a good understanding of its limitations.

References:

- [Search and You May Find](#)
- [Search Is Not Enough: Synergy Between Navigation and Search](#)

See also:

- [Breadcrumbs](#)

- [Fly-Out Menu](#)
- [Global Navigation](#)
- [Navigation](#)
- [Page Up Control](#)
- [Persistent \(Sticky\) Menu](#)
- [Search](#)
- [User Working Memory](#)

Global Navigation

What Problem Does This Solve?

- Allows the user to locate content necessary to accomplish a task.
- Indicates the user's location.
- Allows the user to alternate between search and browse functionality.

Solution

- A horizontal bar contains navigational elements linking to the different site areas. It spans the full width of the browser window.
- The same structure (order) of the navigational elements should be maintained from page to page, so that the user can relate the navigation of the different pages to each other.
- The current selected area should visually stand out compared to the unselected areas.

Example



Interaction

- User rolls over items in main navigation.
- Selected area highlights and fly-out menu displays.

Rationale

- Users are accustomed to looking at the header area for navigation.
- Navigation provides a clear visual indication of what content can be found and places data in context.

When to Use This Pattern

On all screens within InQuira.

See also:

- [Breadcrumbs](#)
- [Navigation](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Fly-Out Menu

A menu that branches off some other visual element.

What Problem Does This Solve?

Users need to have direct access to sub-navigation but the amount of screen real estate for navigation is limited

Solution

- Combine horizontal navigation with a sub-menu that flies-out when the users hovers over the main menu item
- On mouse-over, show the menu.
- The menu always directly moves up, down or right in such a way that it does not cover the other menu items.
- Due to the amount of sub-navigation, at least four levels should be displayed.
- The flying-out should be instant or very quick otherwise, users cannot switch quickly.
- Navigation should be combined with breadcrumbs.

Issues with Fly-Out Menus

- Difficult to select if you leave the area. Make it wider than it appears so that it doesn't disappear if you are moving the mouse just a couple of pixels outside of the menu.
- Fly-out with submenus are not a good idea. Really hard to select - tiny targets (Fitts law).
- You must make it clear that it is a menu, for example change the cursor to indicate the operation by changing the cursor to a hand or arrow to indicate a click.
- Change the cursor when the user moves over the object with which they will interact.
- Use a delay before hiding the fly-out menu again, e.g., 250ms.
- Only use horizontal menus with vertical fly-out menus. The vertical area in which the mouse-pointer needs to be move is too small, i.e., typically 12-14px, and causes the menu to be hidden.

Example



Interaction

- User rolls over items in main navigation.
- Selected area highlights and menu displays.

Rationale

If executed well, with the mentioned points addressed, this menu provides very quick access to sections in the application without cluttering up the screen with navigation. In particular, when users 'get it' it may help them a lot. On the other hand, for one-off interactions, it may cause usability issues.

When to Use This Pattern

- For experienced users who want quick access to menu items.
- Typically there is a main page and all clickable items are grouped under a label.
- Space for navigation is limited.
- Menu items should be found always in the same spot.
- The number of mouse-clicks should be minimized.
- No reloads of pages.
- On all screens within InQuira.

See also:

- [Breadcrumbs](#)
- [Hyperlinks](#)
- [Navigation](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Breadcrumbs

A breadcrumb is a type of secondary navigation scheme that reveals the user's location in a web application. The term comes from the Hansel and Gretel fairy tale in which two little children drop breadcrumbs to form a trail back to their home. Just like in the tale, breadcrumbs in applications offer users a way to trace the path back to their original landing point.

What Problem Does This Solve?

The user needs to be able to navigate up (towards the root page) and have an understanding of where she is in relation to the rest of the application.

Solution

- Show the hierarchical path from the top level to the current page and make each step clickable.
- Displaying a horizontal list of labels that reflect the location of the current page in the hierarchy of the application provides context and helps the user understand where in the application she is and enables her to easily navigate up in the site hierarchy.
- The labels should match the page titles function as hyperlinks to that page and be divided with delimiters such as greater-than signs (>) having the current page as the last label.

Example



Labels

- Where possible, labels should match the title of their corresponding page.
- Use the rules of title capitalization for labels in the breadcrumb.
- Separate each label with a greater-than sign (>) or equivalent delimiter.
- Include the title of the current page as the last label in the breadcrumb.
- Do not use the label "Home" for the topmost page. Instead use the specific name for the topmost page (e.g., "InQuira" rather than "Home".)

Hyperlinks

- [Hyperlink all labels](#) in the breadcrumb except the last one, which corresponds to the title of the current page.

- Never hyperlink the greater-than sign (>) or the spaces that separate the labels.
- The hyperlink should be styled the same regardless of whether it has been visited or not.

Other Considerations

- Never display a breadcrumb on the site's topmost page.
- The breadcrumb sometimes corresponds to the user's recent browsing history, but is not equivalent to it.
- Dynamically include the titles of user-generated content pages as appropriate.

Rationale

- Breadcrumbs provide context relative to the rest of the site.
- Breadcrumbs provide a way to navigate up the site hierarchy.
- The site's main navigation does not always show how a page is related to other pages. A site can have multiple navigation schemes, or a two-level tab navigation that hides the actual complexity of the site.
- When the user performs a search, he does not use the main navigation but jumps directly to a page, so he can lose sense of location easily.
- On websites or applications with three levels or more, users need a fallback navigation method if other navigation methods (menu, search) fail.
- The users need to know where they are in a hierarchical structure and navigate back to higher levels in the hierarchy.

When to Use This Pattern

- On all sites or applications with three levels or more.
- On all screens within InQuira, except the main screen.

See also:

- [Hyperlinks](#)
- [Navigation](#)

- [Why Should We Care About Navigation if We Have Search?](#)

Persistent (Sticky) Menu

Sticky, or fixed, navigation is basically a website menu that is locked into place so that it does not disappear as the user scrolls down the page; in other words, it is accessible from anywhere on the website without having to scroll.

A [usability study](#) found:

Sticky Menus Are 22% Quicker to Navigate

The data from the study indicated that participants were able to find what they were looking for quicker when they didn't have to scroll back to the top of the page. According to this data, sticky navigation could cut 36 seconds off of a five-minute visit to a website.

100% of Test Participants Preferred Sticky Menus Without Knowing Why

Problem

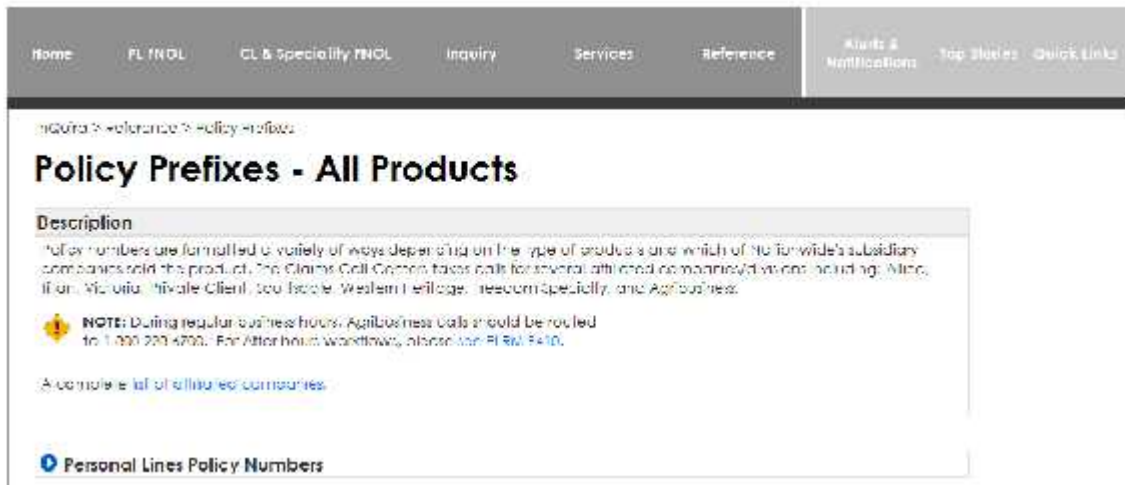
As the user scrolls down the page, she cannot see or use the navigation unless she scrolls all the way back to the top.

Solution

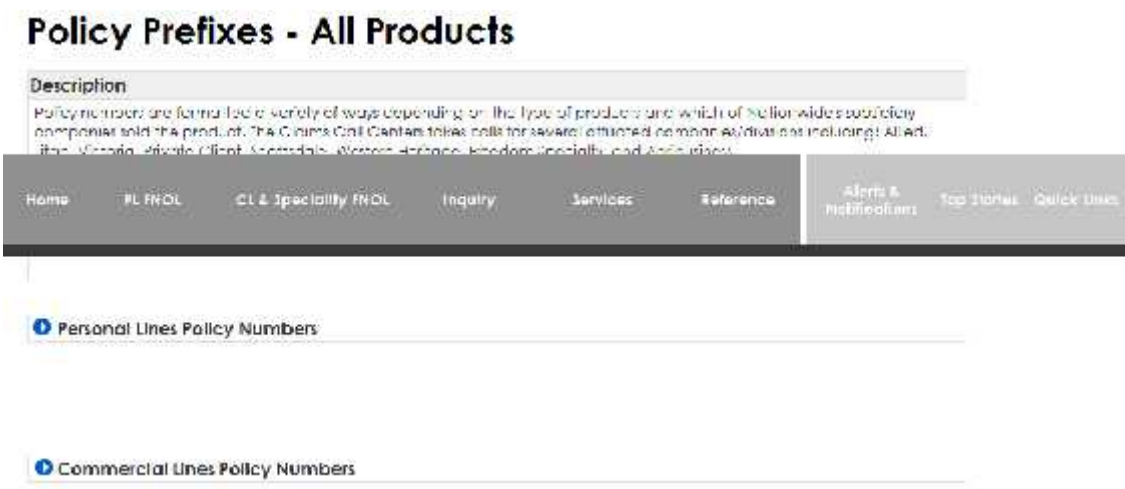
- As the user scrolls down the page, the persistent navigation bar stays with her so that she does not need to scroll all the way back to the top of the page.
- Place an always-visible navigation at a fixed position on the top of the page.

Examples

Before Scroll



After Scroll



Interaction

- User scrolls down the page.
- As user scrolls, the navigation stays in view on the page.

Rationale

This control allows the navigation to always be accessible from anywhere in the application without scrolling.

When to Use This Pattern

On applications with long pages.

See also:

- [Global Navigation](#)
- [Navigation](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Page Up Control

What Problem Does This Solve?

User wants to quickly get to the top of a long page.

Solution

- Add a page up control that scrolls down the page and is visible as the user progresses.
- When clicked, it takes the user to to the top of a long page quickly.

Example

Code	Loss Cause	Description
01	Fire	Any other than those caused by lightning, explosion or wood-burning stoves. Fires caused by appliances, auxiliary heaters, candles, cooling, electrical systems, fireplaces, machinery, fires, smoking, etc.
02	Lightning	Damage caused by lightning, including fires caused by lightning and damage to structures or contents, property resulting from direct or indirect lightning strikes. Also includes power surges from lightning.
03	Controlled Fires (not from Wood-Burning Stoves)	Smoke from controlled fires, such as candles, smoking or oil fires, heaters. Smoke damage from other fires, or from the insured premises, but does not include fires, including smoke damage from wood-burning stoves.
04	Wind/Water Damage (Wind Driven Rain)	Damage caused by wind. Includes water damage from storms such as tropical storms, hurricanes and typhoons (except when reduced to less than 25% for water damage other than from wind-driven rain over 100 mph (157 km/h)).
05	All Other Physical Damage	Any loss that does not fit one of the other loss causes.
06	Vandalism and Malicious Mischief	Damage to property and contents, including damage by tenants, unless excluded. Vandalism and Malicious Mischief. Some exclusions include: theft through postholes, holes, building windows.



Interaction

- User scrolls down long page.
- Page up control is in view as user scrolls down page.
- User clicks page up control and goes to the top of the page.

Rationale

This control allows the user to get to the top of the page quickly.

When to Use This Pattern

With all long pages in InQuira.

See also:

- [Global Navigation](#)
- [Navigation](#)
- [Persistent \(Sticky\) Menu](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Search

Navigation means using the available options and contextual clues to locate content. Buttons, tabs, tables of contents, menus, links, and indexes are common navigation options online. In print, readers navigate with the help of tables of content, indices, and page numbers.

Search is the act of looking for specific content by entering a query in a search engine or application. Users form queries with keywords or search terms, and the search engine displays the query results based on its index and understanding of the request.

Findability is a term for the ease with which information contained on a website or application can be discovered and retrieved through searching or browsing.

Users of Search

Search is one of the most important user interface elements in any large application. Usability studies have classified the following search users:

- **Search-dominant users** will usually go straight for the search button when they enter a website or application. They are not interested in looking around; they are task-focused and want to find specific information as fast as possible.
- **Link-dominant users** prefer to follow the links around a site. Even when they want to find specific information, they will initially try to get to it by following promising links from the home page. Only when they get hopelessly lost will link-dominant users admit defeat and use a search command.
- **Mixed-behavior users** switch between search and link-following, depending on what seems most promising to them at any given time but do not have an inherent preference.

Despite the primacy of search, the website or application still needs to be grounded in a **strong sense of structure and navigation support**: all pages must make it clear where they fit in the larger scheme of the site.

- First, there is obviously a need to support those users who don't like search or who belong to the mixed-behavior group.

- Second, **users** who get to a page through search still **need structure to understand the nature of the page relative to the rest of the site**.
- They also need navigation to move around the site in the neighborhood of the page they found by searching: it is rare that a single page holds all the answers or even that the search found the most relevant page, so users need to see related pages.

Search should be easily available from **every single page** in the application. Search-dominant users will often click on a search button right on the home page, but other users may move around until they become lost. Once that happens, you don't want them to have to search for the search, so it should be right there on the page. This means *any* page, since you can't predict when users will give up navigating and look for the search button.

Why Should We Care About Navigation if We Have Search?

Here are five reasons why websites and applications cannot expect users to rely on search only.

1. Search Requires Knowledge of the Search Space

In order to formulate a good search query, users need to know fairly well what they are searching for. They need to understand the search space and put in the right keywords. Sometimes that is easy — for instance, most Help Desk CSRs can quickly come up with a reasonably good query. But when a novice CSR is trying to use InQuira, he is likely to need some guidance in the search space and calls the Help Desk.

Category landing pages (Services, PL FNOL) often have navigation that helps the user quickly get a mental model of the search space: what subcategories are available and what attributes are relevant for that particular category.

2. Search Increases [Memory Load](#)

Even when users are familiar with the search space, search requires them to **recall** information from their memory. To come up with a meaningful query, a user needs to think about attributes that are relevant for his goal and incorporate them in the query.

Navigation replaces the recall with recognition: instead of forcing users to come up with a complex query, they can recall only a minimum and then use recognition to augment their query with relevant terms.

If search was the only tool available, you would have to come up with your own constraints and then translate them into terms that are likely to be relevant on this application.

Thus, you attempt to retrieve from memory query terms matching that goal. Translating the user goal into a search query that is understandable by the site is no easy task (and in fact people often fail at it — see below) and **imposes a higher cognitive load** than simply selecting the right link from the page.

3. Search Has Higher Interaction Cost than Browsing

Search forces users to work more because not only must they come up with a query, but they also must type it. Typing has high interaction cost; it is error prone and time consuming even with a full keyboard (and even more so on a touch screen). The interaction cost of search is greater than that of browsing. True, some of that interaction cost is mitigated by auto-suggestions, but auto-suggestions usually work well when the query is fairly common and simple.

4. Search Often Works Poorly

Even if users are able to come up with a reasonable search string, chances are that the results returned will often be irrelevant. Big search engines have behind them complex algorithms and a huge database to search in; even if the query is not perfectly formulated, chances are that some relevant results will still be found. That is often not the case on specialized sites or applications.

5. Users Have Poor Search Skills and Don't Know How Search Works

A while back Jakob Nielsen called users' search skills "incompetent". It is still true. We keep seeing this over and over again: **people have no understanding of what makes a good search query**. Their search mental model is corrupted by the big search engines and **they expect search to work in the same way on every site**. Some sites attempt to do some parsing of

the search string and map it onto their attribute space, but most of the time that mapping is very poor resulting in poor search results.

Site search is vital and can save the day for those users who have well defined goals and a good understanding of the information space in which they are searching. However, navigation serves important functions:

1. It shows users what they can find in the application.
2. Teaches them about the structure of the search space.

Using the navigation categories is often faster and easier for users, than generating a good search query. Plus, many times site search does not work well or requires users to have a good understanding of its limitations.

References

- [Search and You May Find](#)
- [Search Is Not Enough: Synergy Between Navigation and Search](#)

See also:

- [Global Navigation](#)
- [Navigation](#)
- [User Working Memory](#)

Action Links

What Problem Does This Solve?

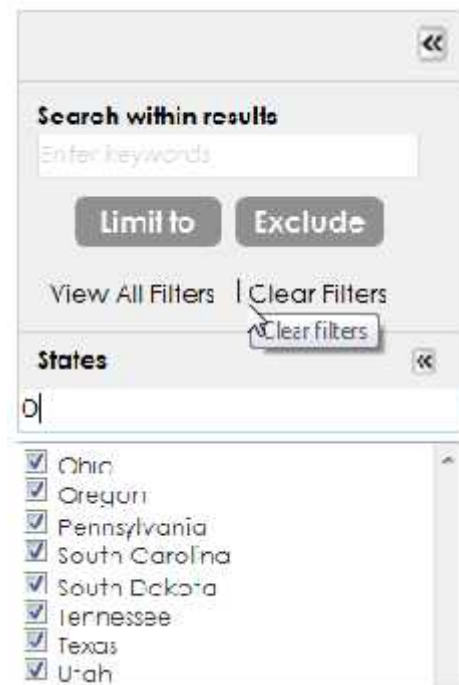
- You have several commands that you need to display together or want to indicate command precedence visually.
- If you're strapped for space, use links instead of buttons.

Solution

- Use a link instead of a button.
- Focus action link labels on helping users make the right decision.
- Use positive phrasing.
- Don't use group labels for multiple action links.

- Using verbs in labels on action links makes the screens more usable because the labels contain all of the information the user needs to be able to make a decision. **All the information is contained in the action link labels.**
- Use [wording that describes the action](#) that the action link represents.
- Use three or fewer words in the label. If the label needs further explanation, provide the details in a [tooltip](#).
- Always present action links in a set of two or more. Logically, there is no reason to ask a question that has only one answer.

Example



Labels

Element	Capitalization Style	Examples
Action Links & Hyperlinks	Title case Not bold No period (.) No colon (:)	View All Filters View More Clear Filters

Interaction

- In the facet filter, user clicks the View All Filters link.
- System displays all filters in sidebar.

Rationale

- Use links instead of buttons to minimize visual noise, conserve screen real estate, or to contrast with buttons to indicate [importance or precedence](#).
- Link commands can also conserve space, both horizontally and vertically, so if you're strapped for space, they can save room.
- One thing to be careful of is that many people perceive links as purely navigational or at least assume they're mostly harmless. If you use them for commands, be sure that the commands are mostly harmless, are easily undoable, or if you do nothing else, at least prompt for confirmation on such commands.
- Since users perceive links as purely navigational, add a [tooltip](#) to the link describing its action, like "Open all filters".
- When it comes to command precedence, using links is a great way to visually subdue certain commands so that standard button commands stand out more. Because links are text and tend to blend into their surroundings (more so for textual contexts, obviously), you can use them to mute certain commands while simultaneously emphasizing others, which is good for communicating the [Primary Action](#).

When to Use This Pattern

- You have several commands that you need to display together or want to indicate command precedence visually.
- If you're strapped for space, use links instead of buttons.

See also:

- [Facet Filtering](#)
- [Hyperlinks](#)
- [Primary & Secondary Actions](#)
- [\(Search\) Results List](#)

- [Search](#)
- [Tooltip Invitations](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Alternate Row Color

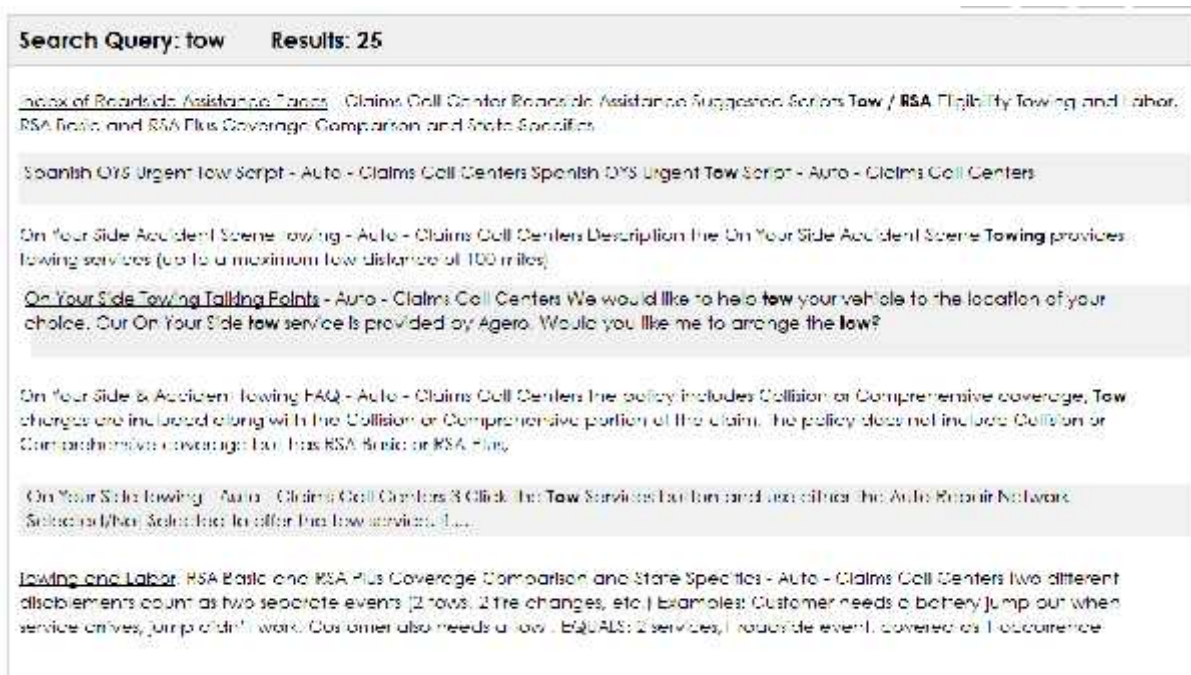
What Problem Does This Solve?

The user needs to visually separate similar looking and overcrowded rows in a table in order to match the values in each row.

Solution

A different shade is used as a background color for every second row.

Example



Interaction

No interactions.

Rationale

Shading every second row increases the ability of users to distinguish between overcrowded data in multiple rows and columns.

When to Use This Pattern

- There are several columns in a table.
- The content of each column is often of similar nature.
- Use when there are multiple rows.
- The height of each row varies making it difficult to judge row separation.

See also:

- [\(Search\) Results List](#)
- [Search](#)
- [Tables](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Application Search

Search is a field that accepts typed-in text in order to look up something or launch a search for related information. All search controls should be able to use keyboard inputs, delete/backspace, the up and down arrows, and the Enter and Return keys. Using the up and down arrows moves you to beginning or end of the text string.

What Problem Does This Solve?

Users need to find a specific item in a large collection of items.

Solution

Provide users with an easy-to-use search mechanism.

Example

Search on InQuira Screen



Quick Search on Every Screen



Interaction

- User clicks into Search box.
- User enters keyword and clicks Search button.
- (Search) Results List displays.

Rationale

- In many cases, users have a sense of what they are looking for, or at least a sense of what to ask for. Users also in many cases are thinking of what they are looking for in ways that do not match with the way you have organized your content.
- The [facet filtering](#) functions give the users control of many aspects of the search. This requires additional knowledge of the data that is available and it is only for experienced users.

- Many users have problems using the AND/OR operators so it is easier to offer the type of match as an option (all terms, any term, etc.).

When to Use This Pattern

- Facet filtering is intended for intermediate to expert users that need full control over the search engine so that they can be more successful in achieving their goals.
- Such users typically have more than superficial knowledge about the items they are searching for. They may know the type of items and are set to search for a specific item they know or may not know exists.

See also:

- [Facet Filtering](#)
- [Primary & Secondary Actions](#)
- [\(Search\) Results List](#)
- [Search](#)
- [Users of Search](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Display Quantity

What Problem Does This Solve?

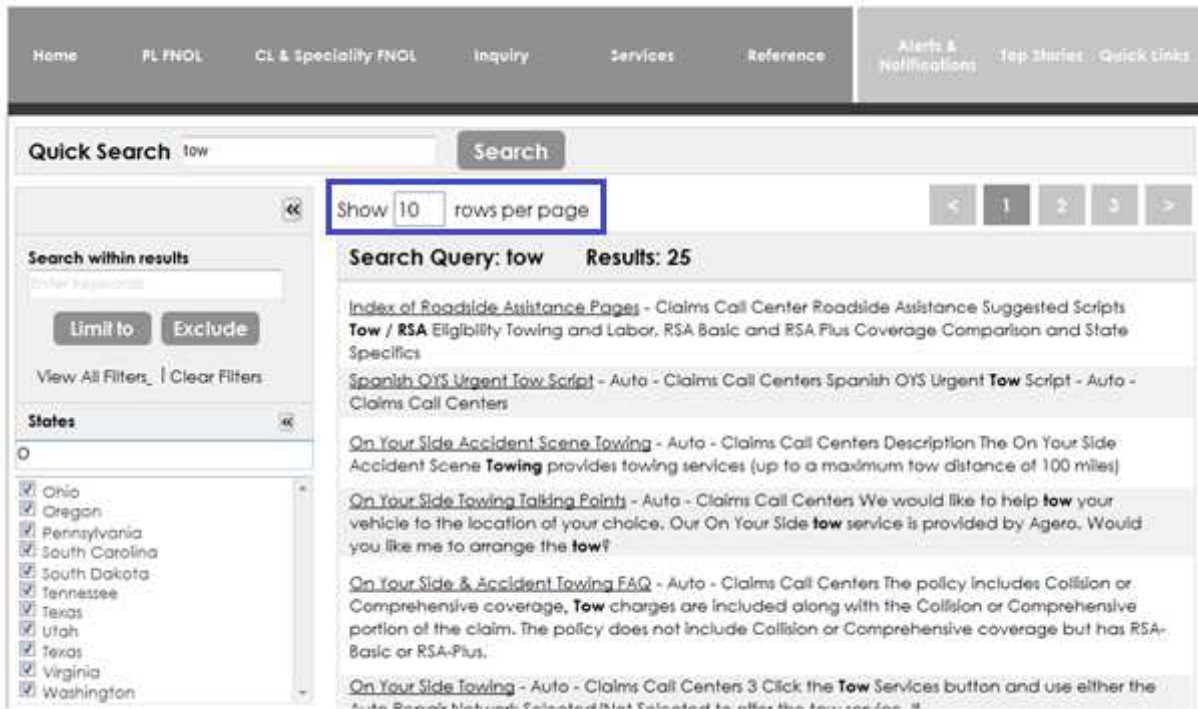
- When there are a substantial number of search results pages it can be cumbersome to navigate through them.
- You need to ask the user for input, usually one-line text, a number or a choice from a drop-down list.

Solution

- Add user-settable view preference mechanism.
- Order your input fields (be it a text box, radio buttons, checkboxes, or drop-down list) as if they are part of a sentence.
- Write a sentence and let the user fill in the blanks of the sentence by selecting or filling out input fields that are in place of words.

- When the input field is not inserted at the end or the beginning of the sentence, it is important for the general readability and understandability of the interface that the input fields do not take up more space than the height of one-line text.
- Input text boxes and drop-down lists therefore work the best for this sort of pattern.

Example



Labels

Element	Capitalization Style	Examples
Drop-Down List Items	Title case Not bold No period (.) No colon (:)	1 10
Static, Supplemental, UI text	Sentence case: Capitalize first word of each sentence Not bold Use ending punctuation	Show n rows per page

Interaction

- User enters "tow" into the Search field.
- User clicks Search button.
- (Search) Results List display.
- User selects 10 from the Show drop-down list.
- Ten rows of search results display on the page.

Rationale

- This pattern helps make the interface self-explanatory.
- Seeing the input or "blanks," in the context of a verbal description helps the user understand what's going on and what he's being asked to do.

When to Use This Pattern

- You need to ask the user for input, usually one-line text, a number or a choice from a drop-down list.
- You tried to write it out as a set of labels/control pairs, but the labels' typical declarative style (such as Name and Address) isn't clear enough for users to understand what is being requested.
- You can verbally describe the action to be taken once everything's filled out, in an active voice sentence or phrase.
- On (search) results lists.

See also:

- [\(Search\) Results List](#)
- [Search](#)
- [Users of Search](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Facet Filtering

The user wants to find a specific item in a large set of information, too large for a person to easily sift through without being able to narrow it down. The information shares a set of common categories that they could be filtered by.

The categories are easily understandable by the users and are something they would likely want to filter based on.

Thus, a filter analyzes a large set of content and excludes any objects that don't meet certain criteria. Facet filtering provides multiple filters to analyze a large set of information.

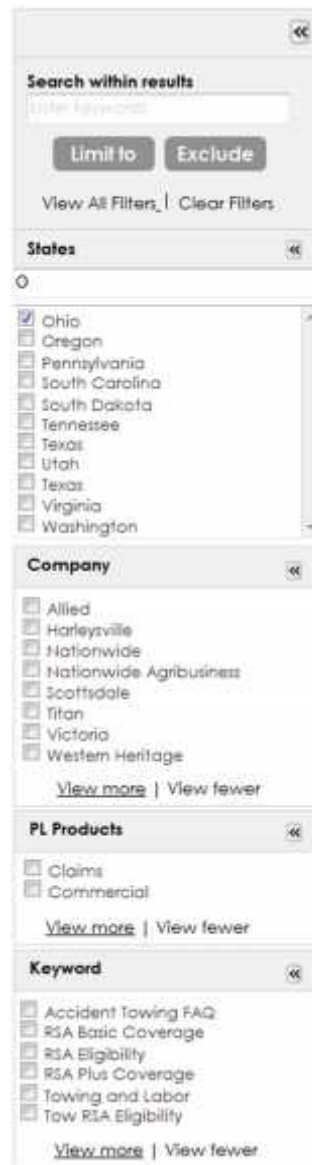
What Problem Does This Solve?

- User wants to refine the search results based on categories.
- User wants to narrow down the results shown by specific categories.

Solution

- Allow user to refine search results in real time using multiple filters in any order until the desired items are found.
- Help users find what they want by providing them with various categories that they can use to filter a [result set](#).
- Check boxes allow users indicate parallel selections of multiple filter values, limiting the scope of search results to those that match them. Provide undo for filter selections. Use the deselect-to-undo paradigm successfully.
- Always make all filters easily available. It is perfectly acceptable to collapse filters to just a label, providing a single link like "View All Filters", or to display previously selected filtering options in a unique way.

Example



Interaction

- User clicks into the States field and enters "O".
- States beginning with "O" display. User selects Ohio.
- The (Search) Results List updates to only display results pertaining to Ohio.

Rationale

- This pattern is more to find something than to analyze data, and it is often used as a way to navigate a catalog of information.
- This is sort of the inverse of the common “Advanced Search” screen, and it is better because it doesn’t require the user to know or specify these details up front.
- It is encouraging and usually optional because users are not confronted with a big, complex form they have to fill out in order to start exploring the data and, if implemented well, **it actually encourages more exploration by fast updating based on progressive facet selection.**

When to Use This Pattern

- There is a large set of information, too large for the user to easily sift through without being able to narrow it down.
- The information shares a set of common categories that they could be filtered by.
- The categories are easily understandable by the users and are something they would likely want to filter based on.
- The users don’t prefer to navigate this kind of information in another way.

See also:

- [Action Links](#)
- [Disclosure Arrows](#)
- [Primary & Secondary Actions](#)
- [Progressive Disclosure](#)
- [Sidebar Toggle](#)
- [Standard Icons](#)
- [Tooltip Invitations](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Input Prompt

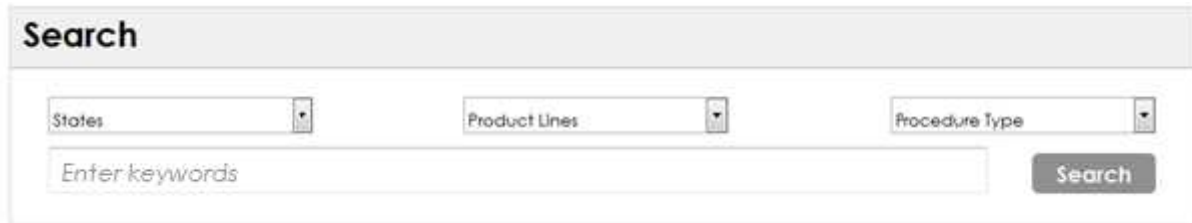
What Problem Does This Solve?

- The user needs to enter data into the system.
- The user doesn't know what to put into fields on forms.

Solution

- Use when the label of an input field does not fully explain what should be filled into it.
- Use in combination with a control label to further explain what kind of input is needed.

Example



The image shows a search interface. At the top, there is a header labeled "Search". Below this, there are three dropdown menus: "States", "Product Lines", and "Procedure Type". Below the dropdowns is a text input field with the placeholder text "Enter keywords" and a "Search" button to its right.

Labels

Element	Capitalization Style	Examples
Input Prompt	Sentence case Gray font color Italics No ending punctuation	<i>Enter keywords</i> <i>Add a category name</i>

Interaction

- User clicks in field containing input prompt.
- Input prompt disappears.
- User enters keywords and clicks the primary action button.

Rationale

- When a user fills out a form, it is most often with the purpose of filling out as quickly as possible to get on with completing their task.
- This is why users often just scan through fields and labels without giving the labels much of a glance.
- By using input prompts, immediate attention is drawn to what the user needs to fill in.

When to Use This Pattern

Use when the user is interacting with the application and she wants to easily and quickly fill out a form.

See also:

- [Primary & Secondary Actions](#)

Pagination

- When there is too much data (or too long a list of items) to display on a single page, break the list of items into a sequence of pages.
- Provide previous and next links to enable people to navigate through the paginated data, as well as, links to the first and last pages in the set.

What Problem Does This Solve?

The user needs to view a potentially large set of data that will not be easy to display on a single page.

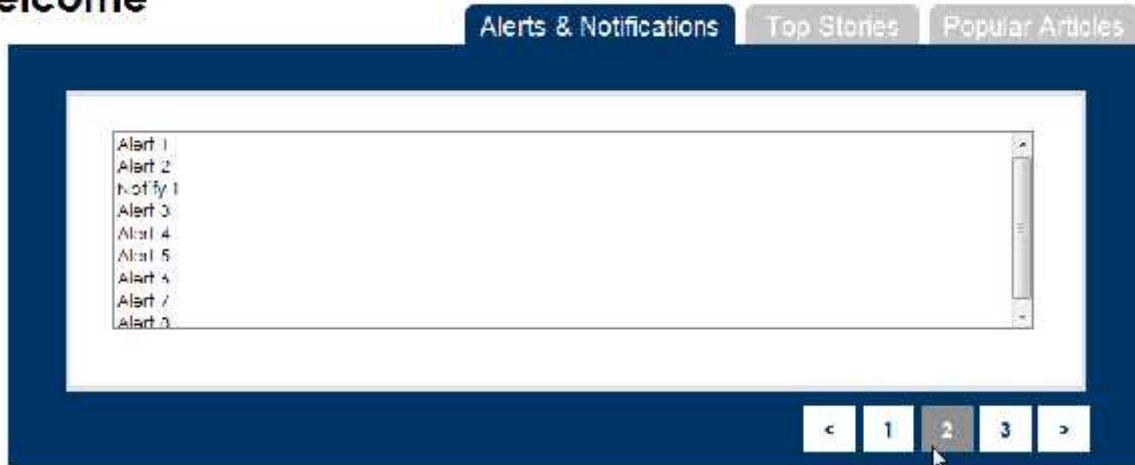
Solution

- Break the list of items into a sequence of pages.
- Provide links to access the *previous* and *next* pages of information.
- Provide information about what type of object the user is browsing.
- Provide large click targets. Small arrows are not a good idea -- really hard to select - tiny targets (Fitts law).

Example

Paginator 1 – Large Click Targets on Dark Background

Welcome



Paginator 2 – Carousel with Large Click Targets

Personal Lines Pollay Numbers

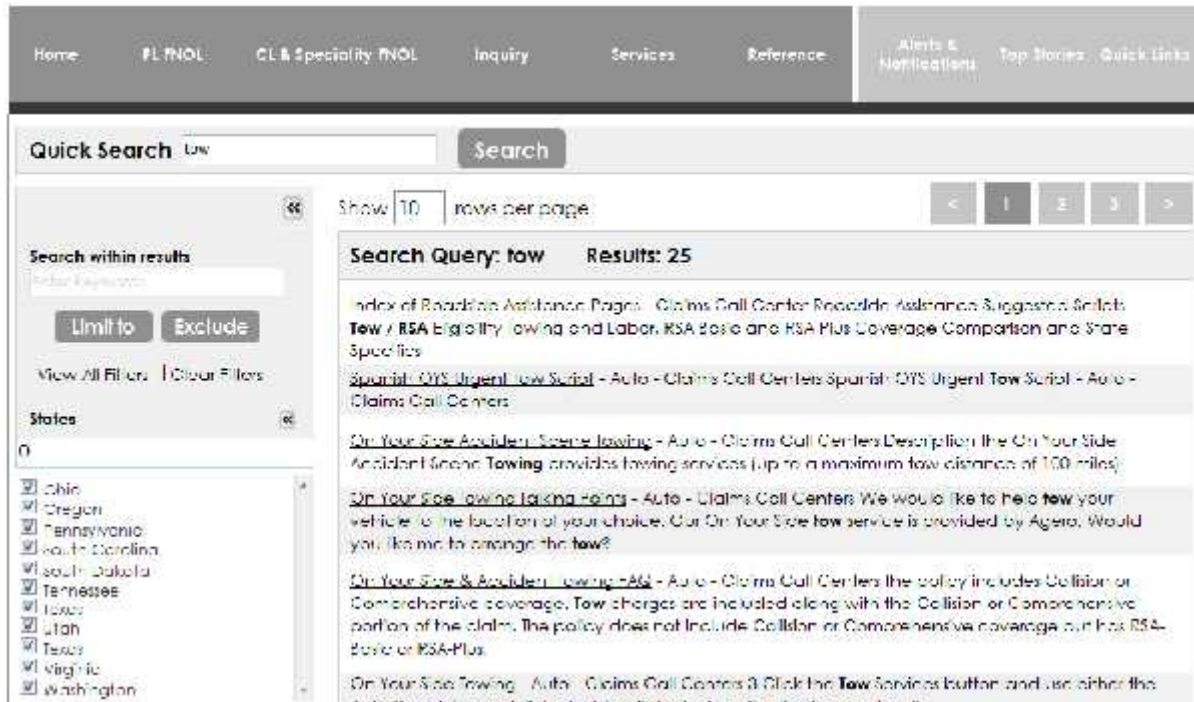
Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla foctus pharetra vulputate, felis tellus mollis orci, sed rhoncus sapien nunc eget odio.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla foctus pharetra vulputate, felis tellus mollis orci, sed rhoncus sapien nunc eget odio.



Paginator 3 – Large Click Targets on Light Background



Interaction

- User enters a keyword into the search field and clicks Search button.
- (Search) Results List displays.
- User clicks 2 and the second page of results display.
- User clicks 1 and the previous page of results display.
- User clicks > and the second page of results display.

Rationale

- The controls are visible at all times during paging (even when disabled).
- This prevents distraction to the user that would occur when removing unavailable controls during paging.

When to Use This Pattern

There is more than one page of data.

See also:

- [Carousel](#)

Primary & Secondary Actions

- Match the visual presentation of actions to their importance to minimize the risk for potential errors and help people complete a form as quickly as possible.
- Primary actions are actions such as Search, Add, and Delete. They enable form completion. Therefore, they are the most important action on the form.
- Secondary actions are actions such as Cancel or Reset. They are less important or alternate actions, which often have negative consequences when used accidentally. For example, clicking Reset erases all the data the user has entered.

What Problem Does This Solve?

Commands that look the same can be hard for users to correctly and quickly choose between, thus slowing down task completion.

Solution

- In general, the navigational path out of a form should be limited to explicit primary and secondary actions.
- A button or [action link](#) represents an action that users can initiate.
- Using verbs in labels on buttons and links makes the screens more usable because the labels contain all of the information the user needs to be able to make a decision. **All the information is contained in the button labels.**
- The primary action should be included in the title/header text, message text and on the primary action button.

Primary Action

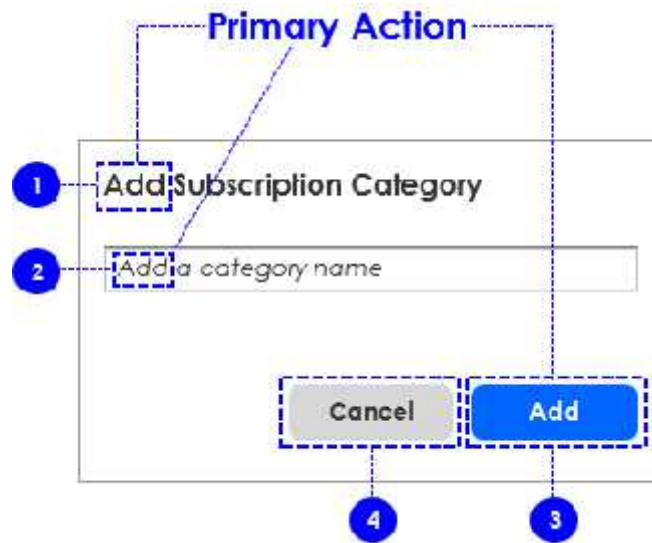
- If there is an overall goal that a view is addressing, then the primary action will most likely be whatever enables users to accomplish that goal.

- Add visual weight to primary action elements.

Secondary Action

- Avoid secondary actions if possible.
- Reduce visual weight on the secondary actions.
- Ensure a clear visual distinction between primary and secondary actions.

Example



1	Title/header
2	Input prompt
3	Primary action button
4	Secondary action button

Labels

The primary action should be included in the title/header text, message/static text and on the primary action button.

Element	Capitalization Style	Examples
Buttons	Title case Bold	Search Limit to Exclude Cancel
Message, Screen Header (Title)	Title case Bold No period (.) No colon (:)	Search Region & State Codes Add Categories
Input Prompt	Sentence case Gray font color Italics No ending punctuation	<i>Enter keywords</i> <i>Add a category name</i>

Interaction

Primary Action

- User clicks into the field and enters a category name.
- User clicks the Add (primary action) button.
- The Subscription screen displays with the category name of "Tow".

Secondary Action

- User clicks into field and enters a category name.
- User clicks the Cancel (secondary action) button.
- No information is saved.

Rationale

- Visual distinction between actions helps users make "good" choices.
- Having an obvious target action that makes sense for most people most of the time both reinforces a sense of the goal that people are working towards and makes it easy to take action to accomplish that goal.
- Especially when there are multiple possible commands within a given view, visually emphasizing the primary action can really help people quickly move on with accomplishing their goal and also can increase the likelihood that they will choose the right action for a given context,

which could mean lower customer service costs and higher customer satisfaction.

When to Use This Pattern

- All form actions are equal. Therefore, the visual presentation of the action should match their importance to make it easier to complete a form.
- There is a known primary action that users normally want to do or that you want to guide them towards.

See also:

- [Action Links](#)

Progressive Disclosure

- With a progressive disclosure control, users can show or hide additional information including data, options, or commands.
- Progressive disclosure promotes simplicity by focusing on the essential, yet revealing additional detail as needed.

What Problem Does this Solve?

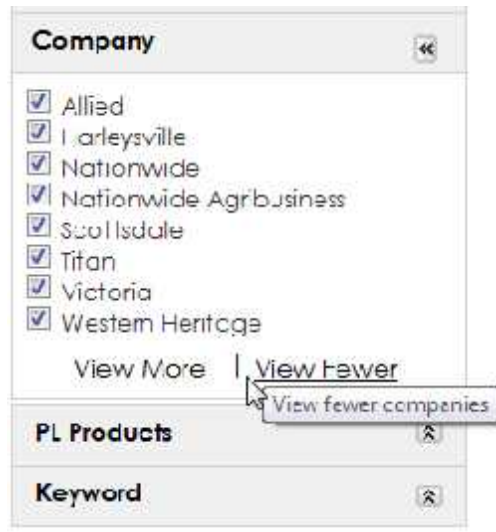
Information that is unnecessary for a given task at a specific point in time can distract users.

Solution

- Initially, only show users information and elements that are necessary, and only show potentially needed information as needed.
- All of the information that the user may need to see during the complete process may seem overwhelming if shown initially.
- Simplify a UI by focusing on the essential, yet revealing additional detail as needed.
- Simplify a UI's appearance by reducing the perception of clutter.

Sample

View Fewer



View More



Labels

Element	Capitalization Style	Examples
Action Links & Hyperlinks	Title case Not bold No period (.) No colon (:)	View Fewer View All Filters Clear Filters

Interaction

View Fewer

- In the company filter, user clicks the View Fewer link.
- System displays only three companies in the filter.

View More

- In the company filter, user clicks the View More link.
- System displays eight companies in the filter.

Rationale

- In order to keep the user focused on a given step or task, it can be useful to only show information and UI elements related to only the current task.
- Long forms and lists of steps may seem daunting to a user. Even if it will not be necessary to fill out all of an overwhelmingly long form, a user will likely see the length of the form or the number of UI elements before coming to the understanding that only relevant portions must be completed.

When to Use This Pattern

- Information that will be provided to users at a given step may differ based on responses or interactions during previous steps.
- All of the information that the user may need to see during the complete process may seem overwhelming if shown initially.

See also:

- [Accordions](#)
- [Action Links](#)
- [Disclosure Arrows](#)

(Search) Results List

What Problem Does This Solve?

Search results can be confusing.

Solution

- Offer search results in comprehensible formats, allowing users to modify these formats to suit their needs.
- While you want to provide users with the ability to reformat search results, you need to try and anticipate a good default way to present them, based on what you know about your users and the type of information.
- As part of the search results, show the user the search terms that were previously provided by the user in an editable search field along with a search button. **Allow the user to modify the search at any time, without having to return to a different search page.**
- A good default will be to sort the results based on whatever relevance algorithms or rules the underlying search engine has been programmed to use.
- Because you won't know much else about how the user may want to see the results, you should offer easy-to-use formatting features to the user. These include allowing the user to sort (e.g., by date, relevance, and other appropriate attributes), as well as, filter based on attributes of the results.
- A [common type of filtering](#) allows users to filter out results based on categories.
- Many searches result in large numbers of results, so allow the user to see those results via a [display quantity](#).
- Also include a pattern such as [paging](#) to break large numbers of search results into manageable chunks of results that users can view one page at a time.

Example



Interactions

- User clicks into Search box.
- User enters keyword and clicks Search button.
- (Search) Results List displays.

Rationale

- Users will have been searching for something because they aren't sure exactly where it is.
- They aren't even sure that the search terms they used were the best ones.
- When they get search results they are likely going to quickly scan the results, potentially looking through many results before they find just the right match.
- Therefore, your goal is to provide search results in a default format that are very easy to visually scan, while also providing a way for users to reformat the results to their liking (e.g., sort, filter, etc.).

When to Use This Pattern

- Users have used search, either a Search Box or Advanced Search.
- Now the users need to process the results, most likely by choosing one or more items for further investigation.

See also:

- [Action Links](#)
- [Alternate Row Color](#)
- [Application Search](#)
- [Display Quantity](#)
- [Facet Filters](#)
- [Pagination](#)
- [Primary & Secondary Actions](#)
- [Sidebar Toggle](#)
- [Search](#)
- [Search Filters](#)
- [Tables](#)
- [Tooltip Invitations](#)
- [Why Should We Care About Navigation if We Have Search?](#)

Search Filters

What Problem Does This Solve?

The user wants to narrow down the results shown by specific categories.

Solution

- Provide drop-down inputs that present the categories by which the user can filter the dataset.
- Once the user selects a category and clicks to filter, the same dataset is shown but only with rows that belong to the selected category.

Example



The screenshot shows a search interface titled "Search". It features three filter dropdown menus: "States", "Product Lines", and "Procedure Type". The "Procedure Type" dropdown is open, showing a list of options: "Procedure Type", "Procedure type", "Frank", "Bulk Bill" (highlighted in blue), "OSU - AND", and "OSU - UC Tech". Below the filters is a text input field labeled "Procedures" and a search button.

Labels

Element	Capitalization Style	Examples
Drop-Down List Items	Title case No period (.) No colon (:)	States Products Procedure Type Bulk Bill

Interaction

- User clicks on Procedure Type filter. Items display in the list.
- User clicks Bulk Bill.
- User enters Procedures into field and clicks Search button.
- All procedures related to Bulk Bill display in the (Search) Results List.

Rationale

- Adding filters allows the user to reduce the number of items displayed and pinpoint the appropriate dataset.

When to Use This Pattern

The dataset that is too large to show on one page.

See also:

- [Application Search](#)
- [Pagination](#)
- [Primary & Secondary Actions](#)
- [Search](#)
- [Tables](#)

- [Why Should We Care About Navigation if We Have Search?](#)

UI Controls

A user interface control is an element of a graphical user interface (GUI) such as a button, menu, list box, text window and dialog box.

Accordions

- With a progressive disclosure control, users can show or hide additional information including data, options, or commands.
- Progressive disclosure promotes simplicity by focusing on the essential, yet revealing additional detail as needed.

What Problem Does This Solve?

When there are more items than can comfortably fit into a limited space or when the number of items, if displayed all at once, would overwhelm the user, then the question is how to give the user access to all of the items in digestible chunks and without requiring scrolling, which can remove the user from the context or page position they may prefer.

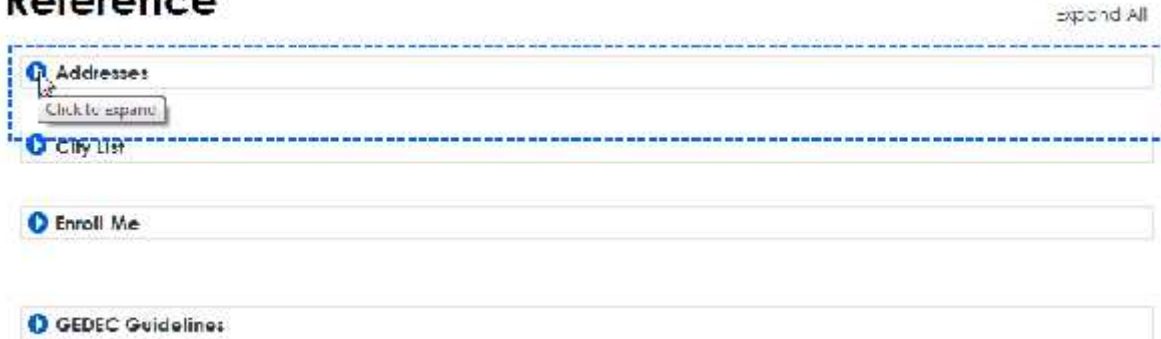
Solution

- Each section has a panel, which upon clicking can be expanded vertically to show its subsections.
- Highlight the current panel so the user can distinguish open headers from closed panel headers.
- Add [tooltips](#) on the [expand/collapse arrows](#) that open and close the panels, like "Click to expand".

Example

Accordion Closed

Reference



Accordion Opened

Reference



Interaction

- User hovers over arrow. Tooltip "Click to expand" displays. User clicks arrow or panel to open it.
- Clicking again on arrow or panel closes it.

Rationale

- When there is too much information/data on a page, the user can easily become overwhelmed and lost.
- Accordions break up information into smaller, logical chunks.

When to Use This Pattern

Use to compress a large amount of information into a limited space.

Accessibility

- For keyboard users, an accordion usually ends up behaving either as a tree view or as a tab view.
- Add keyboard navigation to the accordion with tabbing between labels and down/up arrows switching between panels.

See also:

- [Carousel](#)
- [Disclosure Arrows](#)
- [Progressive Disclosure](#)
- [Standard Icons](#)
- [Tooltip Invitations](#)

Carousel

- [Usability studies](#) show that carousel [paginators](#) that are clear and with visible label [navigation](#) are more usable.
- Descriptive labels are more informative and describe the information contained on the page.
- Labeling each slide tells users what they'll get, thus helping users to find information faster.
- Users are more likely to click on something that looks informative to them.
- The label navigation is also larger and placed in a more visible location than navigation arrows (Tiny Targets – Fitt's Law). It's not only easier to see, but users know what to expect when they see it.

What Problem Does This Solve?

The user needs to browse through a set of items and possibly select one of them.

Solution

- Because the carousel only shows one item at a time, you only need a small space to show many items.
- Show the items (3 – 8) in a circular fashion so that one item can be selected at a time.
- If the user wants to view the rest of the items on the list, she must click one of the button labels.
- The user can in this way browse the list of items back and forth.

Example





Interaction

- User clicks on a label button on the carousel.
- The next screen displays.

Rationale

- The carousel only takes up small space on the screen, so it allows you to let the user scroll through many list items without scrolling up or down.
- As the carousel is circular, the start of the list will be shown after the user has reached the end. This behavior gives the user a great opportunity to browse through all list items.

When to Use This Pattern

- Use when you have a large set of items to show, but want to let the user concentrate his or her attention only on a select few items at a time
- Use when you do not have enough space to show all items at once.

See also:

- [Accordions](#)

- [Navigation](#)
- [Pagination](#)
- [Progressive Disclosure](#)

Disclosure Arrows

What Problem Does This Solve?

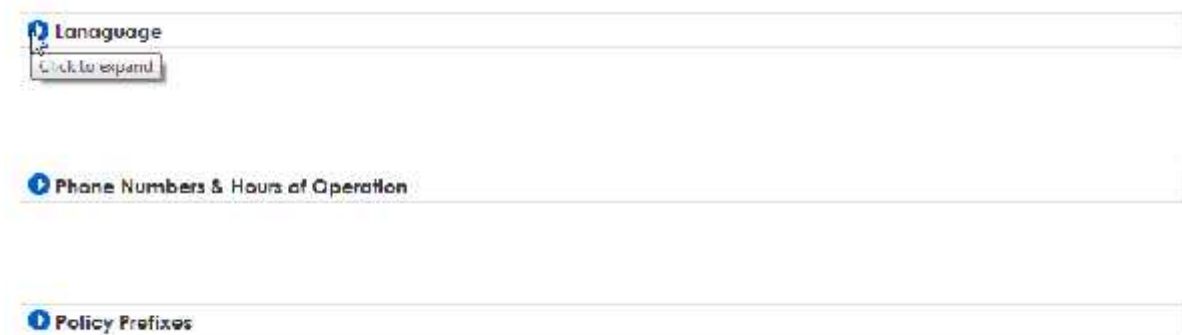
When there are more items than can comfortably fit into a limited space or when the number of items, if displayed all at once, would overwhelm the user, then the question is how to give the user access to all of the items in digestible chunks and without requiring scrolling, which can remove the user from the context or page position they may prefer.

Solution

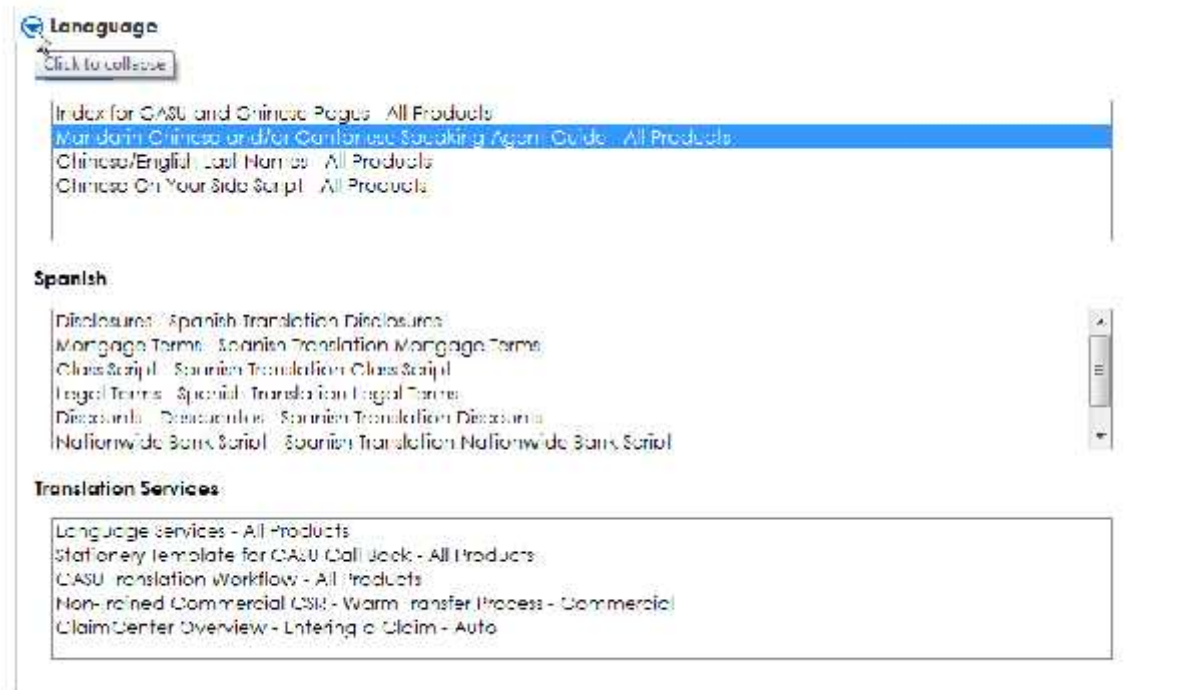
- Each section has a panel, which upon clicking can be expanded vertically to show its subsections.
- Highlight the current panel so the user can distinguish open headers from closed panel headers.
- Add [tooltips](#) on the [expand/collapse arrows](#) that open and close the panels, like “Click to expand”.

Example

Accordion Closed



Accordion Opened



Interaction

- User hovers over arrow. Tooltip “Click to expand” displays. User clicks arrow or panel to open it.
- Clicking again on arrow or panel closes it.

Rationale

- When there is too much information/data on a page, the user can easily become overwhelmed and lost.
- Accordions break up in information into smaller, logical chunks.

When to Use This Pattern

Use to compress a large amount of information into a limited space.

Accessibility

- For keyboard users, an accordion usually ends up behaving either as a tree view or as a tab view.
- Add keyboard navigation to the accordion with tabbing between labels and down/up arrows switching between panels.

See also:

- [Accordions](#)
- [Progressive Disclosure](#)
- [Standard Icons](#)
- [Tooltip Invitations](#)

Hyperlinks

What Problem Does This Solve?

User wants to navigate to another page.

Solution

- Use a hyperlink when hypertext is the right interaction pattern—that is, text that responds when clicked and navigates the user to more information about, or related to, the text that was clicked.
- Keep individual hyperlinks far enough apart from one another that the user clicks them accurately.
- Only use the disabled state for a hyperlink if that state is temporary—such as other system processes are occurring—or if the hyperlink can be enabled by user action.
- When navigating to an external website, put the domain name inside the [tooltip](#) and style it with a secondary font color. Adding the domain name to the tooltip lets users know that they're about to navigate to an external site so they won't be surprised when they click the link. It's enough to just show the top-level domain.
- Keep the link text concise. If you want to provide additional information, put it inside the link's tooltip.

Example

Roadside Assistance Reimbursement Workflow

Expand All

Description

If a Towing/RSA service has already been completed on a member's vehicle, they can pay for the services covered by Roadside Assistance out of pocket and request reimbursement by phone, mail, email or fax.

Guidelines

- No claim needs to be filed for reimbursement except for reimbursement of boat tows. (Claims Services will file the claim and assign to adjuster for handling once the bill is received) **UPDATE!**
- **Note:** If a customer indicates they are calling in regarding reimbursement for towing charges related to an existing NW claim, please refer them to their claim adjuster.
- Agero does not handle reimbursement requests for lock outs, out-of-network towing, trip interruption, etc. Reimbursement requests are handled by Nationwide.
- Commercial and Specialty Auto (Titan) customers **may only submit reimbursements by email, mail or fax, not phone.**
- Questions about coverages and limits can be answered using [Roadside Assistance Reference](#).

Handling Procedures

Labels

Element	Capitalization Style	Examples
Action Links & Hyperlinks	Title case Not bold No period (.) No colon (:)	Roadside Assistance Reference

Interaction

- User clicks the hyperlink.
- User navigates to another page in the application.

Rationale

In either form, a hyperlink is a piece of text that a user can click to open a web page in a browser, or to navigate to another page—or to another section within the same page.

When to Use This Pattern

- Hyperlinks have multiple uses, as [navigation](#), as paths to more information, and as access points to features and functionality.
- Hyperlinks are primarily used for navigation.
- Hyperlinks are clearly distinguished from regular text so that users can easily recognize the linked text as a link.
- Consistent treatment of hyperlinks helps ensure that users understand the difference between regular text and hyperlinks.

See also:

- [Action Links](#)
- [Breadcrumbs](#)
- [Navigation](#)
- [Tooltip Invitations](#)

List Builder

What Problem Does This Solve?

The users need to build and manage a list of items.

Solution

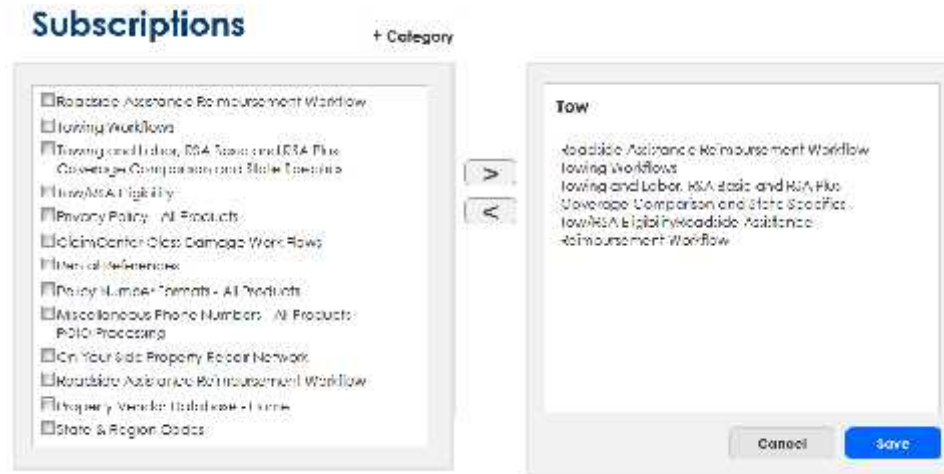
- Show both the “source” list and the “destination” list on the same page.
- Let users move items between them via buttons.

Example

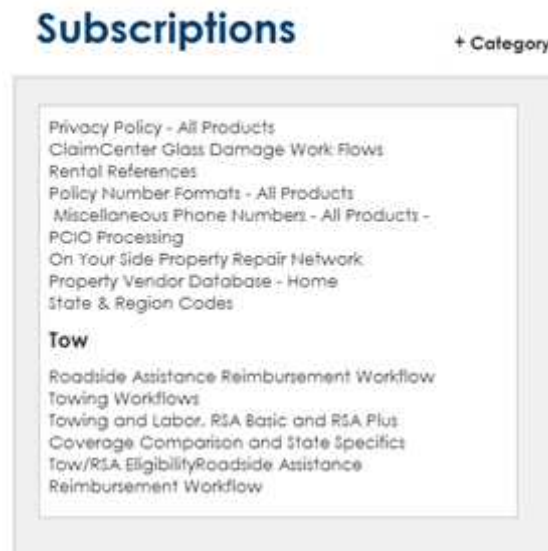
Add Subscription Category



Tow with PLRMs



Subscriptions With Category



Interaction

- User clicks the +Category button.
- A screen displays allowing user to create a category name. User enters "Tow" and clicks the Add button.
- The subscription builder displays allowing user to add subscriptions to the newly-created category. User clicks the checkboxes next to the corresponding PLRMs and then clicks the the arrow button.
- Selected PLRMs now display under Tow. User clicks the Save button.

Rationale

- This pattern displays both lists on the same page for the user to see.
- It also prevents the user from having to scroll down a long list of items.

When to Use This Pattern

- The user wants to create a list of items by choosing them from another list.
- The list may be long – too long to easily show as a set of checkboxes.

See also:

- [Primary & Secondary Actions](#)
- [Standard Icons](#)

Module Tabs

- Using a single-line row of links formatted as tabs that are immediately stacked above the content is a solution to provide access to content if limited space is available and page refreshes are not desired.
- The active pane needs to be visualized and category titles should be short and less than ten.
- Module tabs can be used if the content does not need to be viewed in context with each other. Using tabs builds on a real-world metaphor of file folders.

What Problem Does This Solve?

The user needs to navigate through one or more stacked panes of content without refreshing the page.

Solution

- Present a single-line row of links immediately above the stacked panes of content they will control.
- Never stack multiple rows of tabs to control a single piece of content.
- Separate the links with the vertical bar character "|" or through an equivalent graphic.
- Always show one tab as selected with its content visible
- Indicate the selected tab by highlighting the background of the "cell" around it.
- The content should be visually connected to the tab it is associated with and be visually bounded (usually by a box.)
- Only one content area may be visible at a time.

Maintaining the Metaphor

- Tabs should appear in the same location as the user clicks from one tab to another.
- Wherever possible, do not refresh the entire page upon selection of a new tab.
- Selecting a tab will not affect other parts of the page.
- Selecting a tab will not navigate to a different page or perform an action (beyond switching the visible content).

Example

Services



Interaction

- User clicks on the Auto Repair Network tab.
- PLRM numbers for that category display.

Rationale

- Tabs provide context. They give visual indication of a user's location within a body of information.

- Tabs build on a real world metaphor. The selected state is reinforced with the file folder tab metaphor of a folder physically in front of the others in the set.
- Tabs provide [navigation](#). They provide the ability to navigate alternate content views.

When to Use This Pattern

- There are two or more sections.
- The category titles are relatively short and predictable.
- The entire width of the page is needed for content.
- The categories belong to a single topic.
- Divide a large dataset into manageable chunks.

Sidebar Toggle

With a [progressive disclosure control](#), users can show or hide additional information including data, options, or commands. [Progressive disclosure](#) promotes simplicity by focusing on the essential, yet revealing additional detail as needed.

What Problem Does This Solve?

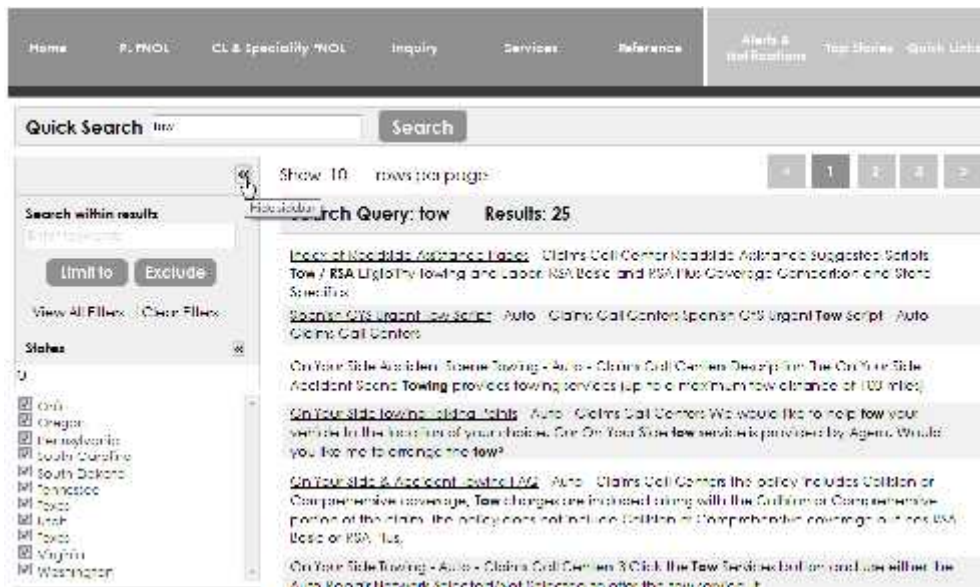
User needs more room in the main page area to view information.

Solution

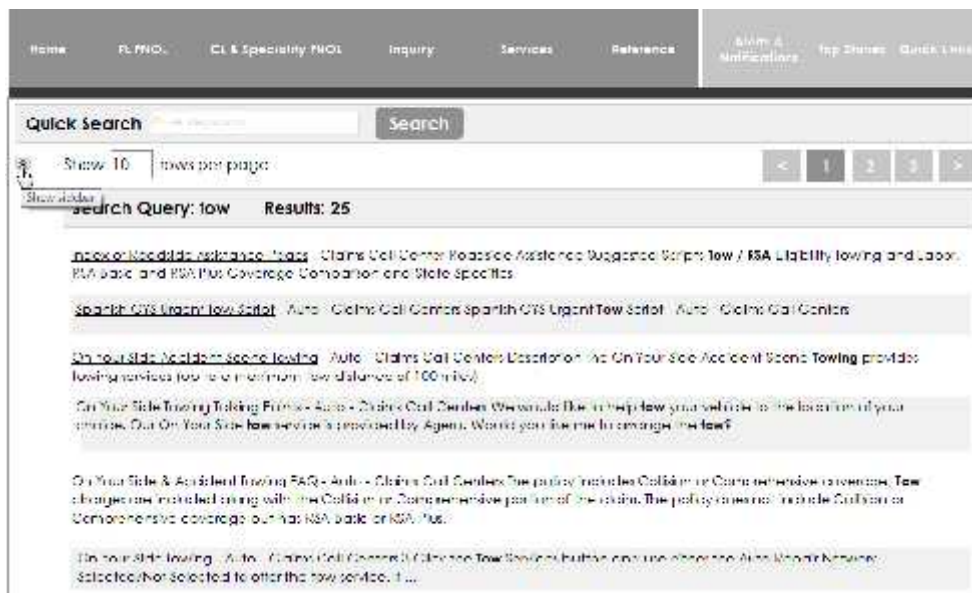
- By default, the sidebar is open.
- Use an arrow to allow the user to hide and then show the faceted filtering sidebar.
- Add [tooltips](#) on the [expand/collapse arrows](#) that open and close the panels, like “Show sidebar” and “Hide sidebar”.

Example

Sidebar Open



Sidebar Closed



Interaction

- By default, the sidebar is open.

- User hovers over arrow. Tooltip “Hide sidebar” displays. User clicks arrow to close sidebar.
- Clicking again on arrow opens it.

Rationale

- As a rule, people like to adjust their environment to suit their needs and tastes, especially when they are going to spend a lot of time there.
- Using the sidebar toggle allows people to do this, and in doing so, they can take better advantage of their spatial memory to more effectively and comfortably use the application.

Accessibility

Provide ALT tags that preview what will happen when the user engages an interaction. For example, if you can close a panel and provide a way to communicate where the panel is collapsed to, you might consider putting text in the ALT tag for the image that tells the user what will happen when the click/activate the close icon and how they can get the panel open again.

When to Use This Pattern

On the (Search) Results List.














See also:

- [Accordions](#)
- [Disclosure Arrows](#)
- [Progressive Disclosure](#)
- [Tooltip Invitations](#)

Standard Icons

- Icons provide a visual cue to pay more attention to the information.
- Use Icons to convey warnings/alerts such as Caution, Stop, Go. Be sure to use them consistently.

Task	Icon	Suggested Tooltip
Alert, Warning		“Important information”

Close	 	"Click to open" "Click to expand"
Confirmation		"Confirmation" or "Success"
Open	 	"Click to collapse" "Click to close"
Error, Warning		"Warning" or "Error"
Reference		"Display reference information"
Updated Information		"Updated information"
Multi Selection Lists		
Add		"Add"
Remove		"Remove"
Pagination		
The number appears selected to indicate what page of results the user is currently viewing.		
Numbers represent the pages in the results list.		
Next		"Next"
Previous		"Previous"

Tables

Left-Justified Text is Easier to Read

Left aligned text is easier to read than centered text for paragraphs. This is because when you center your text, the starting place of each line changes. This forces your users to work harder to find where each line begins to continue reading.

Without a straight left edge, there is no consistent place where users can move their eyes to when they complete each line. However, when your paragraph text is left aligned, a straight left edge appears. Users can read each line by simply moving their eyes to the left edge each time. This makes your paragraphs faster and easier to read because the user's eyes don't have to work as hard to find where the line starts each time.

What Problem Does This Solve?

- User needs to browse and analyze a large amount of structured data.
- User needs to read and scan structured data.
- User needs to compare structured data.

Solution

- Create a table to display structured data so that users can read, scan, compare, browse, and analyze the data.
- Provide a descriptive title. Display title at top of the table.
- Focus should be on the information not the table design.
- Use [alternating row colors](#) to visually separate similar looking and overcrowded rows in a table.
- Don't remove UI elements that are useful. Keep elements that support structure, organization, and readability.
- Allow sorting.
- Provide [filtering options](#).
- Use [pagination](#) when results don't fit on one page.

Example

Search Query: tow Results: 25

[Index of Roadside Assistance Tolls - Claims Call Center Roadside Assistance Suggested Scripts **Tow / RSA** Eligibility Towing and Labor, RSA Basic and RSA Plus Coverage Comparison and State Specifics](#) - Auto - Claims Call Centers

[Spanish OYS Urgent Tow Script - Auto - Claims Call Centers Spanish OYS Urgent **Tow** Script - Auto - Claims Call Centers](#)

[On Your Side Accident Scene Towing - Auto - Claims Call Centers Description The On Your Side Accident Scene **Towing** provides towing services \(up to a maximum tow distance of 100 miles\)](#)

[On Your Side Towing Telling Points - Auto - Claims Call Centers We would like to help **tow** your vehicle to the location of your choice. Our On Your Side **tow** service is provided by Agero. Would you like me to arrange the **tow**?](#)

[On Your Side & Accident Towing FAQ - Auto - Claims Call Centers The policy includes Collision or Comprehensive coverage. **Tow** charges are included along with the Collision or Comprehensive portion of the claim. The policy does not include Collision or Comprehensive coverage but has RSA Basic or RSA Plus.](#)

[On Your Side Towing - Auto - Claims Call Centers & Click the **Tow** Services button and use either the Auto Repair Network Selection \(for Selection\) to offer the **tow** services. \[...\]](#)

[Towing and Labor: RSA Basic and RSA Plus Coverage Comparison and State Specifics - Auto - Claims Call Centers Two different disbursements count as two separate events \(2 tows, 2 tire changes, etc.\) Example: Customer needs a battery jump out when service arrives, jump didn't work. Customer also needs a tow. \(BQOAS: 2 services, 1 roadside event\). Covered as 1 occurrence.](#)

State	Time Zone
Alabama - AL	Central Standard Time (CST)
Alaska - AK	Alaska Standard Time (AKST)
	Aleutian Islands Hawaii-Aleutian Standard Time (HST)
Arizona - AZ	Mountain Standard Time (MST)
	The Navajo Nation uses Daylight Savings Time (DST), the rest of the state does not.
Arkansas - AR	Central Standard Time (CST)
California - CA	Pacific Standard Time (PST)
Colorado - CO	Mountain Standard Time (MST)
Connecticut - CT	Eastern Standard Time (EST)
Delaware - DE	Eastern Standard Time (EST)
Florida - FL	Most of state: Eastern Standard Time (EST)
	West of the Apalachicola River: Central Standard Time (CST)
Georgia - GA	Eastern Standard Time (EST)
Hawaii - HI	Hawaii-Aleutian Standard Time (HST)
	Hawaii does not use Daylight Savings Time (DST).

Interaction

- User enters a search term in the search box.
- (Search) Results List displays in table.
- User clicks “Index of Roadside Assistance Pages” [hyperlink](#).
- The RSA index page displays.

Rationale

- Tables provide structure to data.
- A well-designed table makes it easy to scan information and find items from a large dataset.
- Users are familiar with data tables.
- Tables allow easy comparison between items.

When to Use This Pattern

There is tabular, mostly text-based data that user needs to browse through and analyze

See also:

- [Alternating Row Colors](#)
- [Disclosure Arrows](#)
- [Display Quantity](#)
- [Facet Filtering](#)
- [Search Filters](#)
- [Pagination](#)
- [\(Search\) Results List](#)

Appendix A

User Experience Overview

In IBM's write-up on User-Centered Design, there is a section titled "[Cost Justifying Ease of Use](#)," which states: "**For developers and manufacturers, the advantages of creating usable products far outweigh the costs. The rule of thumb: Every dollar invested in ease of use returns \$10 to \$100.**"

The write-up then goes on to say that reports have shown that it is far more economical to consider user needs in the early stages of design than it is to solve them later. For example, in *Software Engineering: A Practitioner's Approach*, author Robert Pressman shows that for **every dollar spent to resolve a problem during product design, \$10 would be spent on the same problem during development and \$100 or more if the problem had to be solved after the product's release.**

Simply stated, it is far less expensive to prevent a problem occurring in the first place, than to fix it later.

User Working Memory and Long-Term Memory

Working memory, the part of our brain that consciously processes information, dominates everything we do in terms of learning. Working memory can only hold 4-5 bits of information at one time and information in working memory lasts only around ten seconds.

Long-term memory appears to have an unlimited capacity. Information in long-term memory is stored in schemas, which are mental structures we use to organize and structure knowledge. Schemas incorporate multiple elements of information into a single element with a specific function. And existing knowledge in long-term memory is brought into working memory to help us understand the world.

Cognitive Load

Sometimes learning involves great effort. That's because working memory is quite vulnerable to overload, which occurs as we study increasingly complex subjects and perform increasingly complex tasks. As user experience designers, we have to watch out for **cognitive load**, which refers to the total amount of mental activity imposed on working memory in any one instant.

What causes too much demand on working memory? More information than the user can process. But high cognitive load is also strongly influenced by the number of elements in working memory that interact with each other. Often, complex learning is based on interacting elements that must be processed simultaneously.

Minimizing Cognitive Load

Many top usability guidelines—from chunking content to optimizing response times—are aimed at minimizing cognitive load. In addition to these basics, there are three more tips for minimizing cognitive load:

1. **Avoid visual clutter** - Redundant links and meaningless typography flourishes that slow users down.
2. **Build on existing mental models** - People already have mental models about how applications work, based on their past experiences using

other applications. When you use labels and layouts that they've encountered in other applications, you reduce the amount of learning they need to do.

3. **Offload tasks** - Look for anything in your design that requires users to read text, remember information, or make a decision. Then look for alternatives: Can you show a picture, re-display previously entered information, or set a smart default? You won't be able to shift all tasks away from users, but every task you eliminate leaves more mental resources for the decisions that truly are essential.