Upgrading the Dropdown

What is a dropdown menu?

Thinking about different users

What Spotify does well

How can we make it better?

Prototyping my revised design

Reflection

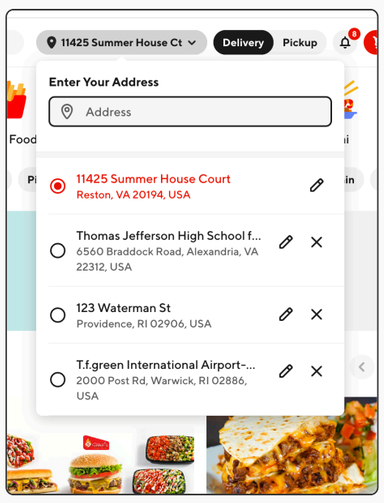
Accessible Components Project

Join me as I take a deep dive into a classic UI component—the dropdown menu! I'll analyze how it's designed across three different applications, evaluating its strengths and weaknesses in learnability, memorability, efficiency, and accessibility. Using these insights, I'll redesign the component to create a more seamless and inclusive experience for all users.

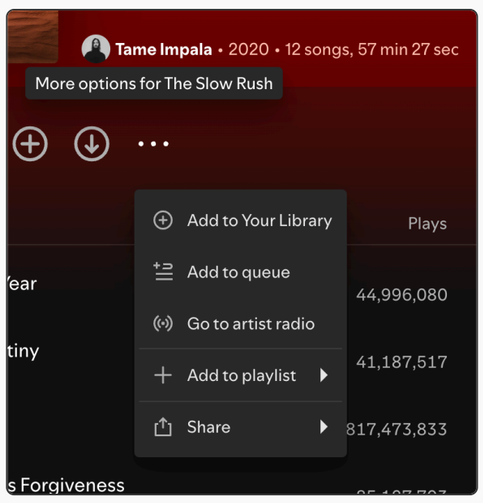
What is a dropdown menu?

A dropdown menu is an interactive UI component that reveals a list of options when clicked or hovered over. It's extremely common in web design in any context where you need to organize multiple choices without overwhelming the screen. Your favorite website might use a dropdown menu for navigation menus, settings, filters, and form selections. Dropdowns improve organization by reducing clutter and consolidating information, making them essential for modern, clean, intuitive interfaces.

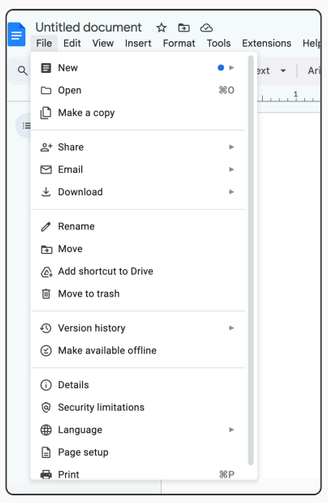
Here are some examples:



Doordash website



Spotify app on Mac



Google Docs website

Thinking about different users

For this project, I interacted with the dropdowns while trying to keep different users in mind. As a user, I'm most comfortable navigating my laptop with a mousepad but I also tried using my keyboard and Mac's screenreader, VoiceOver.

	Doordash	Spotify	Google Docs
Mouse & Touchpad	- Highlights component on hover - Highlights options on hover - Menu appears on release rather than click	- On hover: displays label and highlights - Highlights options on hover - Menu appears on release rather than click	- Highlights component on hover - Highlights options on hover - Displays submenus on hover - Menu appears on click rather than release
Keyboard	- Tab to access component and continue traversing options - Enter to show the menu and select an option - Esc/enter to leave and reenter menu	- Tab to access component - Esc/enter to leave and reenter - Up/down arrows to traverse options (circularly)	- Left/right arrows to navigate if you're already in the toolbar - Characters to select some options (e.g. "A" key for "Align & Indent"). - Esc key to deselect a tab - Up/down arrows to traverse options (circularly)
Touch	N/A	N/A	N/A
Visuals	- Bold borders on current selection - Option icons are very clear - Dot indicator for address is redundant since the chosen address is colored red - Several states and components within the menu itself (e.g. search bar) - Automatically closes the menu once you select an option	- Icon is initially somewhat ambiguous - Submenus indicated with arrow (expands to the right) - Grays out unavailable options - Option icons are effective - Automatically closes the menu once you select an option - No latency for submenu to pop up	- Grays out unavailable options - Navigate long menus with scroll bar - Submenus indicated with arrow (expands to the right) - Option icons are effective - Keyboard shortcut annotations for some options - Options are subdivided into categories with lines - Automatically closes the menu once you select an option - Poor latency for submenu to pop up

General observations from interacting with the dropdowns

	Doordash	Spotify	Google Docs
Default/Enabled	- Oval shaped component - Light gray fill - "Down" icon to indicate dropdown feature	- Small icon, no outline - Light gray	- Only text, no outline/border/ fill
Hover	- Highlights component, dark gray	- Color change to white - Displays component label	- Highlights component with a box, light gray
Focused	- Black border around component	- Displays component label - Blue & white outline around icon	N/A
Selected	- Returns to light gray fill - White menu with drop shadow - Display selection with red text - Display search bar component	- Grays out unavailable options - Menu appears on release rather than click	- Highlights component with a box, light gray - White menu with drop shadow
Hover over option	- Highlights option, light gray	- Highlights option, light gray - Displays submenus (if available)	- Highlights option, light gray - Displays submenus (if available)
Selected option	- Automatically closes the menu on click selection	- Automatically closes the menu on click selection	- Automatically closes the menu on click selection

Comparing output between dropdowns in various states

Upgrading the Dropdown

What is a dropdown menu?

Thinking about different users

What Spotify does well

How can we make it better?

Prototyping my revised design

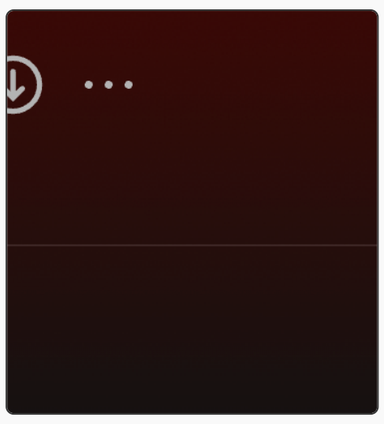
Reflection

	Doordash	Spotify	Google Docs
Flow, commands, and output	1. Tab through other toolbar components to get to icon 2. On focus: reads *"[current choice] menu pop up, button"* 3. Ctrl-opt-space to see component options: reads *"menu 5 items"* 4. Reads search bar component label. 5. Tab to navigate menu items 6. Reads which option is currently selected	1. Tab through other toolbar components to get to icon 2. On focus, reads component label: *"More options for [album], menu pop up collapsed, button"* 3. Ctrl-opt-space to see component options: reads *"menu 5 items"* 4. Ctrl-opt-left/right to navigate menu items 5. Reads menu items on selection. Also reads *"item [number] / 5"* 6. Ctrl-opt-shift-up to exit menu focus group	1. On click, reads component label: *"[menu name], expanded, submenu, (1 of 8)"* 2. Up/down to navigate menu items 3. Reads blurb for each menu item – jumbled and unclear (e.g. "*New n right pointing black pointer, submenu" for "File > New"*)
Issues and inconveniences	- Doesn't read component label on focus - Doesn't clearly state that you are in a search bar or expected to type - Options are read in a long-winded /unclear/jumbled way (reads entire address) - Doesn't tell you how to exit from any focus groups or navigate between options	- Automatically opens submenus but very unclear how to leave that focus group and return to the main menu options	- Tab doesn't work for navigation with an open text editor - Doesn't tell you how to exit from any focus groups or navigate between options or open submenus

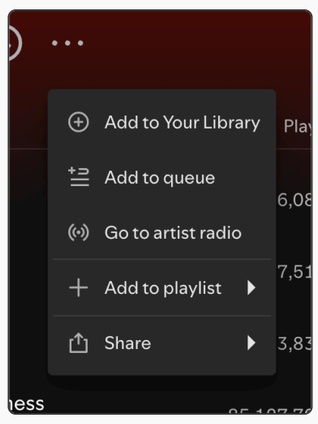
Evaluating dropdown accessibility using VoiceOver

What Spotify does well

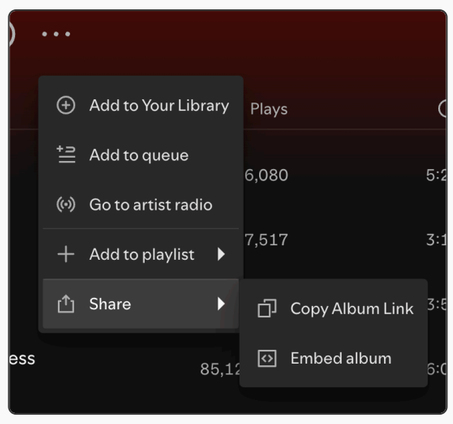
For now, let's focus on one of my FAVORITE applications. What is Spotify's take on the classic component? In this example, Spotify provides a list of additional options for a given album:



Before - Default state (unclicked)



Before - Selected state (clicked)



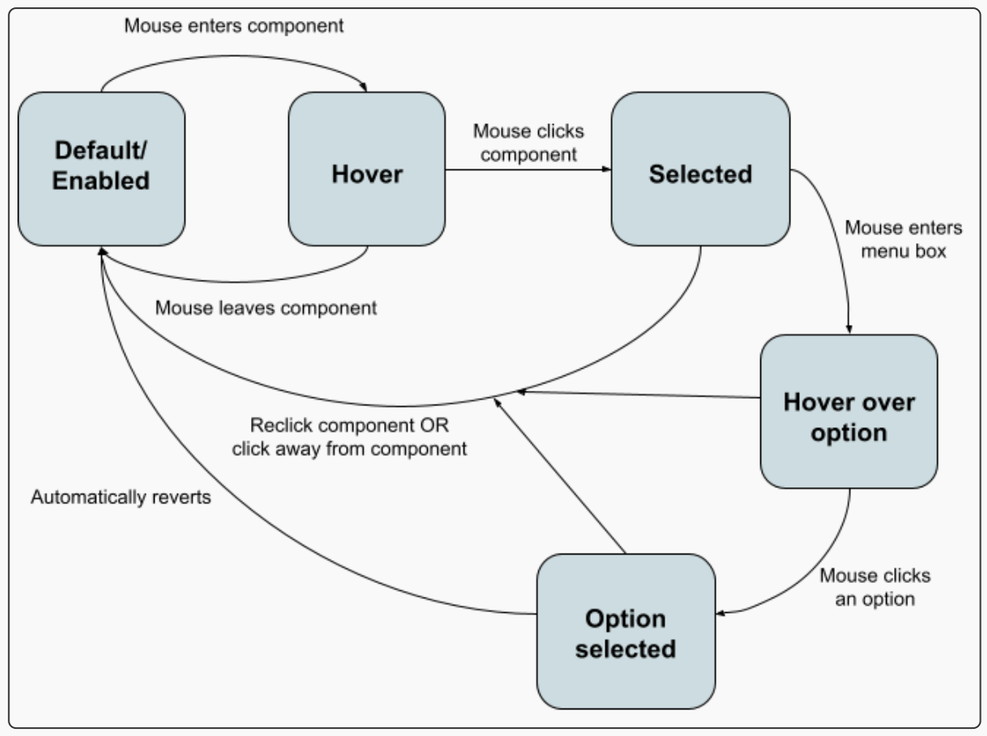
Before - Option hover state

Here's what I like about their execution:

- Spotify indicates that the Options icon is interactive by highlighting it when your mouse hovers over it.
- The location on the page is intuitive.
- It appears on a menu bar BELOW the album but ABOVE the songs. This is also convenient when navigating the page with the tab key.
- The icons for each option are explanatory and effective.
- Organizing options into submenus makes it efficient with space on the screen. No need to scroll!
- Spotify highlights options and opens submenus on hover as well.
- You can use up/down arrows to navigate the menu options which is also intuitive. When you reach the bottom, the focus order just circles back to the top!

How can we make it better?

Here are diagrams summarizing the state changes for the current dropdown menu design



State model for mouse user

Upgrading the Dropdown

What is a dropdown menu?

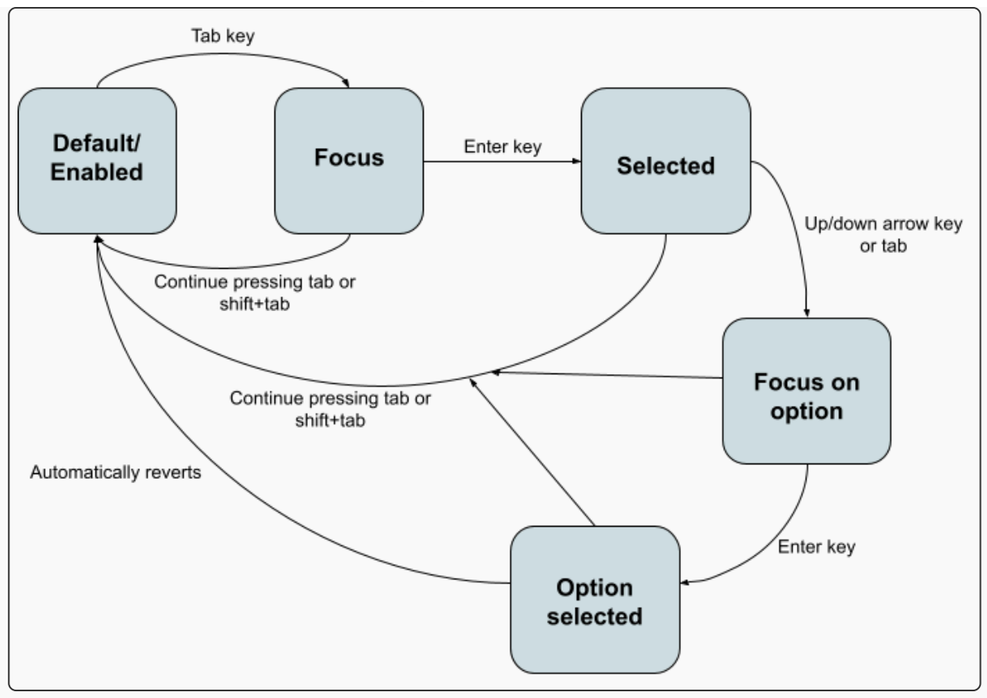
Thinking about different users

What Spotify does well

How can we make it better?

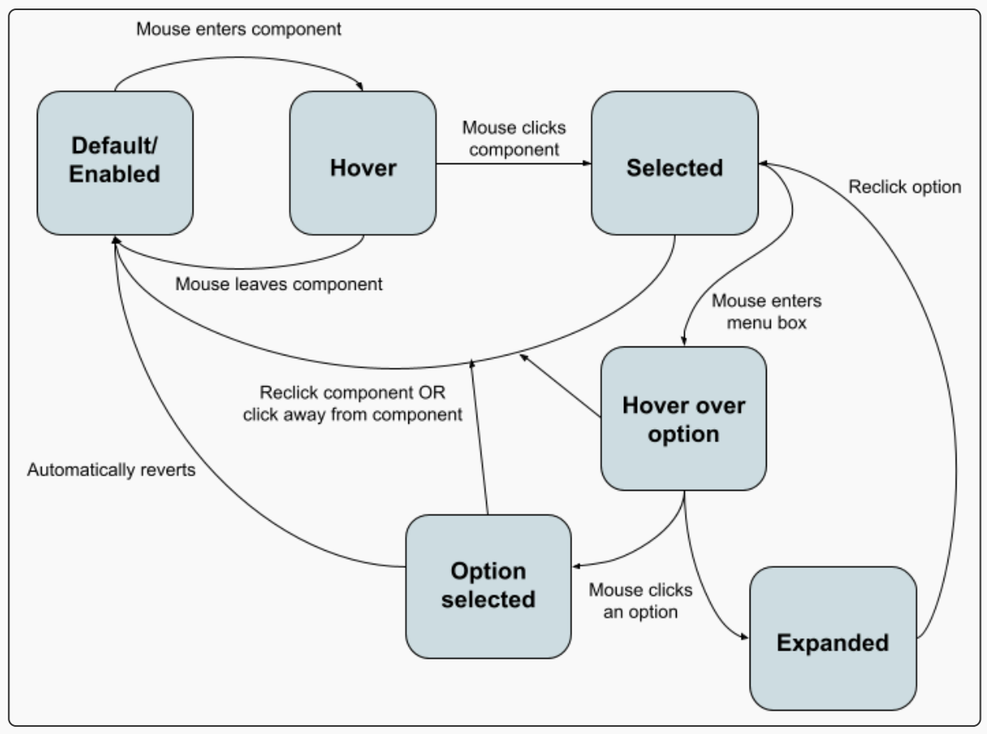
Prototyping my revised design

Reflection

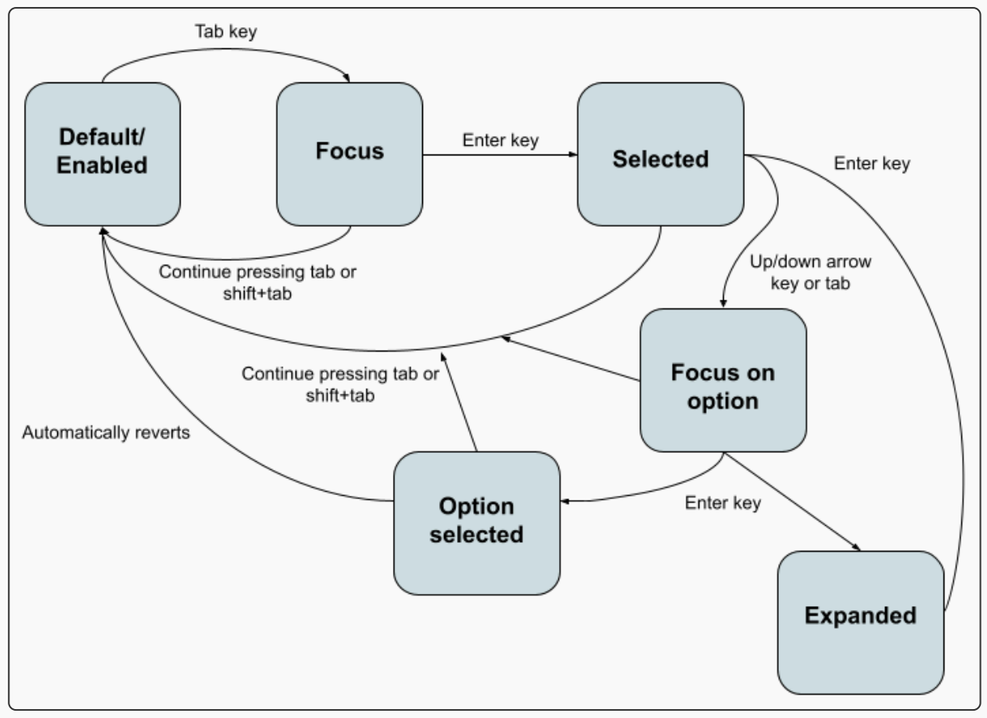


State model for keyboard user

There's definitely room for improvement here! Let's make this more screenreader-friendly.

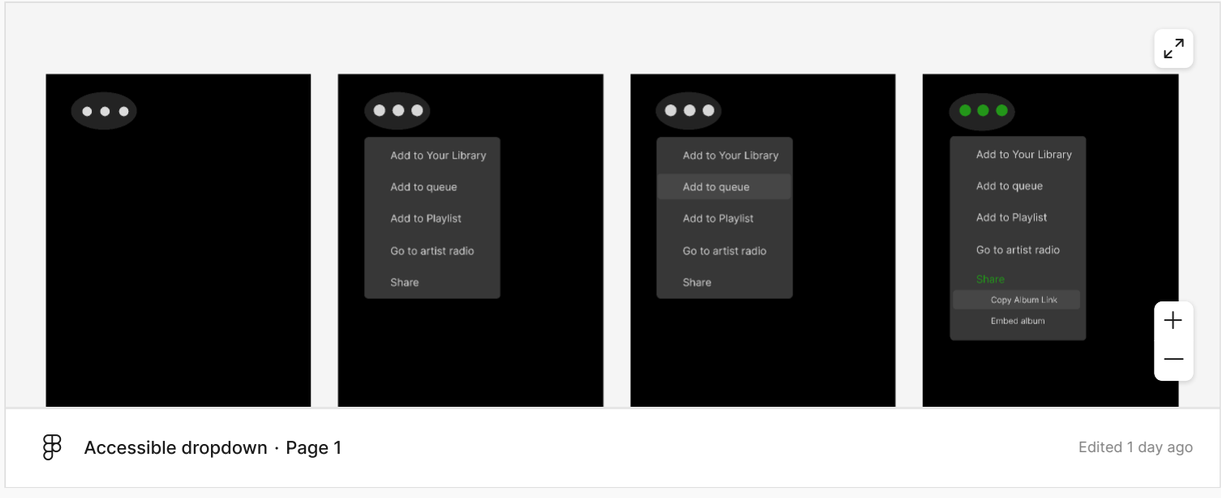


Revised state model for mouse user



Revised state model for keyboard user

Prototyping my revised design



Redesigning Spotify's Dropdown

As we can see, the current design is already functional, but small tweaks can make a big difference in usability for all. Here's how I improved it:

Clearer Interaction & Feedback

- The **menu icon turns green when active**, providing stronger visual confirmation than the subtle gray-to-white transition in the original design.

Upgrading the Dropdown

What is a dropdown menu?

Thinking about different users

What Spotify does well

How can we make it better?

Prototyping my revised design

Reflection

Improved Discoverability & Ease of Use

- **Icons enlarge on hover**, signaling interactivity and making it clear that an action can be performed.
- The menu icon's **dots are bigger and better spaced**, improving click accuracy and making it harder to miss.
- A **subtle outline around the icon** separates the pop-up menu from the background, enhancing contrast and visibility.

Better Layout & Organization

- **More spacing between options** reduces accidental selections, making the menu easier to navigate with a mouse.
- Grouping related actions into expandable submenus (accordion-style) keeps the interface clean and structured, reducing cognitive load.
- **Frequently used options are now placed near the top**, making navigation more intuitive.

Enhanced Accessibility for Screen Reader Users

- The menu is now **structured into logical sections**, preventing an overwhelming list of options.
- **Users can control what they hear** and skip past submenus when navigating with a keyboard—accordion menus only expand on mouse click or Enter key.

Reflection

The dropdown menus I observed did well in terms of logically abstracting away extensive information in the menu options. Both Google Docs and Spotify used submenus that extended out to the right for additional options. I also liked how Doordash used an outline around the component to show that you can interact with it. I incorporated these things into my new design. In terms of accessibility for visually-impaired users, I also tried to improve contrast by using the accent colors already created by Spotify.

I also tried to make the menu more straightforward to navigate with a keyboard and screen reader. With the current focus order laid out by Spotify, left/right arrow keys move between options. However, this makes it unintuitive to move from a submenu back into the main menu. With my new design, you have to explicitly open the submenu so that it opens below the label. This way you can use the Enter key to open it and use the up/down arrows to navigate the submenu. This change accounts for a mismatch when using Spotify as a visually-impaired person who wants to navigate the app using VoiceOver.

When using Google Docs, the screen reader reads menu items in a jumbled and unclear way (e.g., "New n right pointing black pointer, submenu"). This negatively impacts users with visual or cognitive impairments, as they might find it difficult to understand the options or navigate effectively. The lack of clarity could make it difficult to complete tasks, such as creating a new document or accessing submenus.

Initially, I marked Spotify's low latency as an objectively positive feature when interacting with the menu. However, after using VoiceOver, I found that it can also make navigation difficult for some users. If the screen reader responds too quickly to every minor interaction (hovering, clicking, or keystrokes), it can produce a flood of auditory feedback. I often found that screen readers read out multiple items at once, making it hard to keep up or distinguish between them.

When creating components, mouse users seem to be the most commonly prioritized. Components usually respond to hover states, right-click menus, and drag-and-drop interactions, which are not easy to access with other input methods. I found that using a keyboard (and screen reader) was less intuitive and inconvenient, requiring more complex keystrokes. While all three applications worked relatively will with the mouse, Google Docs performed notably low with the keyboard and exemplified how important it is to fill in those gaps in order to be inclusive of all.