Lists

Best Practice: How to Implement Lists

There are different types of lists that are helpful for different cases:

- A **list inlay** allows the user to view the contents of all items in a list, from one screen, by simply clicking on the item.
- **Archive lists** consolidate data sets into broad categories, listed by the day, week, month or year in which the information items were uploaded or edited.
- A **scrolling list** makes all options available on the page, window or panel, with items further down the list accessed by scrolling.

The following steps will help you implement lists successfully:

1 Before doing anything else, you must **gather all of the options**, arrange them into broader categories if necessary (e.g., for archive lists these would be days, weeks, months or years), and assign logical labels to these options.

2 **Place the options in a logical order.** For example, the list may be ordered according to the date files were last modified—as is the case with some file management applications—or they may simply be arranged alphabetically. Whichever method you adopt, ensure it is both *instantly apparent* and the *best order* for the *job*. For example, arranging emails alphabetically would not be ideal for users, as they typically need to distinguish new messages from ones they have already opened.

[Continued on next page]

[Continued from previous page]

3 **Make the means of scrolling clearly apparent** but unobtrusive when using a scrolling list design pattern. Allow the user to use the keyboard to move through the options.

4 **Provide the user with shortcuts** to jump to frequently selected options, such as by typing in the first letter of the item or the automatic placing of these items at the top of the list after a number of uses.

5 **Display or open the contents as close to the list as possible** and without the need for any movement between different regions of the user interface, such as tabs, windows or distinct panels. With the implementation of a list inlay, the contents should be displayed directly below the selected item and the same gesture should allow the user to close the contents—returning the list back to its original state. Some designs allow the users to open items by clicking on the general region occupied by the option label, and they can then close each by clicking on the general region occupied by the contents. This has the benefit of reducing the level of precision required to open, close, and switch between items *quickly*.

6 When a selection has been made, **ensure that the contents open automatically** and are instantly visible so the user does not have to find them.

7 **Provide informative feedback** confirming the user's selection by changing the color of the chosen option.

In general, you must ensure the category labels appear ‘clickable’ so the user knows which elements can and cannot be interacted with. For example, using a dotted line underneath the titled categories or an east-facing arrow can give the impression of ‘interactability’. Additionally, it can be useful to change the color of category label text when the users have successfully made their selection. This has two major benefits: firstly, the user knows his/her click has hit the target; secondly, thanks to the change in the color of previously selected options, the user is aware of categories he/she has already checked when searching for an item, thereby saving time and effort.

Some items in a list are more important to a user than others. Reflecting these personal differences in a rigid user interface design is hard. Therefore, in websites and applications where

[Continued on next page]

[Continued from previous page]

users would benefit from having certain dates more prominent, you should provide them with the ability to personalize the display. For example, allowing the users to place a star next to a specific category or establish their own dashboard, containing important dates, allows them to get the design working for them, rather than having to work around the design.