

UI Design – Guidelines and Principles (Part 2)

ITS290F

The Eight Golden Rules of Interface Design (Shneiderman, 1985)

1. Strive for consistency
2. Seek universal usability
3. Offer information feedback
4. Design dialogs to yield closure
5. Prevent errors
6. Permit easy reversal of actions
7. Keep users in control
8. Reduce short-term memory load

1. Strive for consistency

Consistent sequences of actions should be required in similar situations; **identical terminology** should be used in prompts, menus, and help screens; and **consistent color, layout, capitalization, fonts**, and so on, should be employed throughout. Exceptions, such as required confirmation of the delete command or no echoing of passwords, should be comprehensible and limited in number.

2. Seek universal usability

Recognize the **needs of diverse users** and **design for plasticity**, facilitating transformation of content. Novice to expert differences, age ranges, disabilities, international variations, and technological diversity each enrich the spectrum of requirements that guides design. Adding features for **novices**, such as **explanations**, and features for **experts**, such as **shortcuts and faster pacing**, enriches the interface design and improves perceived quality.

3. Offer informative feedback

For every user **action**, there should be an interface **feedback**. For **frequent and minor actions**, the response can be modest, whereas for **infrequent and major actions**, the response should be more substantial. Visual presentation of the objects of interest provides a convenient environment for showing changes explicitly.

4. Design dialogs to yield closure

Sequences of **actions should be organized into groups** with a beginning, middle, and end. **Informative feedback at the completion** of a group of actions gives users the satisfaction of accomplishment, a sense of relief, a signal to drop contingency plans from their minds, and an indicator to **prepare for the next group of actions**. For example, e-commerce websites move users from selecting products to the checkout, **ending with a clear confirmation page** that completes the transaction.

5. Prevent errors

As much as possible, **design the interface so that users cannot make serious errors**; for example, gray out menu items that are not appropriate and do not allow alphabetic characters in numeric entry fields. If users make an error, the interface should **offer simple, constructive, and specific instructions for recovery**. For example, users should not have to retype an entire name-address form if they enter an invalid zip code but rather should be guided to repair only the faulty part. Erroneous actions should leave the interface state unchanged, or the interface should give instructions about restoring the state.

6. Permit easy reversal of actions

As much as possible, **actions should be reversible**. This feature relieves anxiety, since users know that errors can be undone, and encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data-entry task, or a complete group of actions, such as entry of a name-address block.

7. Keep users in control

Experienced users strongly desire the sense that they are in charge of the interface and that the interface responds to their actions. They **don't want surprises or changes in familiar behavior**, and they are annoyed by tedious data-entry sequences, difficulty in obtaining necessary information, and inability to produce their desired result.

8. Reduce short-term memory load

Humans' limited capacity for information processing in short-term memory (the rule of thumb is that people can remember “**seven plus or minus two chunks**” of information) requires that designers **avoid interfaces in which users must remember information from one display and then use that information on another display**. It means that cellphones should not require reentry of phone numbers, website locations should remain visible, and lengthy forms should be compacted to fit a single display.

Principles:

Nielsen and Molich's 10 User Interface Design Guidelines

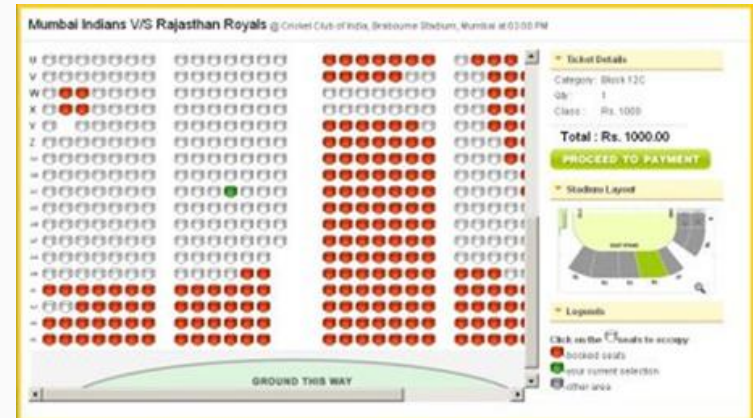
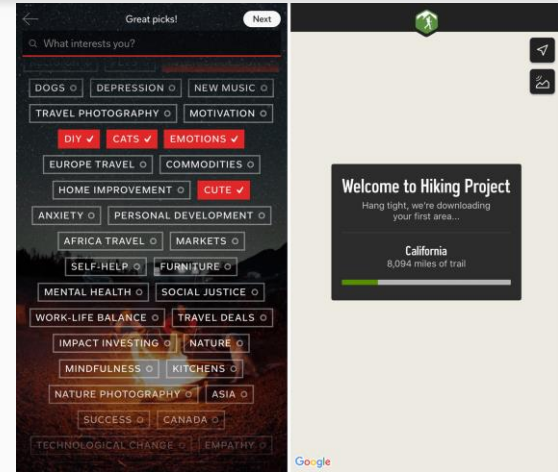
The 10 User Interface Design Guidelines (Nielsen and Molich)

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation


1. Visibility of system status


Users should always be informed of system operations with **easy to understand** and **highly visible status** displayed on the screen within a reasonable amount of time.

- A *change of color* once the user has clicked on a button, or a *progress indicator* when a process needs a little longer to finish. These indicators show that the **system is working**, and **reduce uncertainty** — e.g., users may tap on the same button multiple times because they weren't sure if the first time worked.
- Providing *immediate feedback for interactive events* allows users to **quickly identify the source of errors** and fix them as soon as they were made. In fact, immediate feedback is one of the main benefits of *direct manipulation*, an interaction style in which users can **act directly upon different UI objects**.



1. Visibility of system status: Examples

United Airlines Flight 410		
On-time - departs in 1 hour 41 mins		
PDX		DEN
Departs Portland, today		
Time	Terminal	Gate
11:48 AM	-	E2
Arrives Denver, today		
Scheduled 3:18 PM	Terminal	Gate
3:17 PM	-	B27

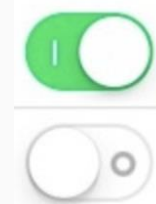
United Airlines Flight 410		
Delayed (21 mins) - arrives in 57 mins		
PDX		DEN
Departs Portland, today		
Scheduled 11:48 AM	Terminal	Gate
11:49 AM	-	E2
Arrives Denver, today		
Scheduled 3:18 PM	Terminal	Gate
3:39 PM	-	B27

Color change to indicate delays

On / Off?



On / Off?



On / Off?

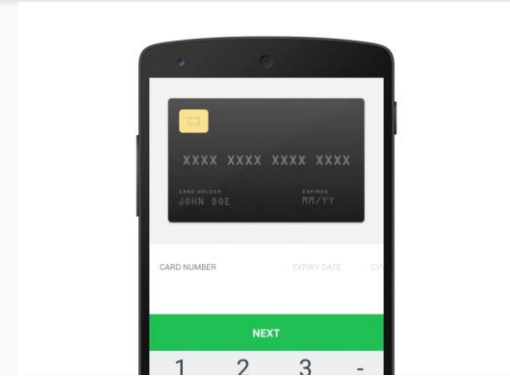


2. Match between system and the real world

Designers should endeavor to mirror the **language and concepts users would find in the real world** based on who their target users are. **Presenting information in logical order** and **piggybacking on user's expectations** derived from their real-world experiences will reduce cognitive strain and make systems easier to use.

Leverage familiarity with real-world objects and activities

- When users transition from the physical world to the digital world, they carry the interpretations **based on their past experiences** with real-world objects.
- Users expect that **UI elements which reflect material objects be similar to those objects from the real world**.



iBooks iPad application using the metaphor of wooden book shelf.

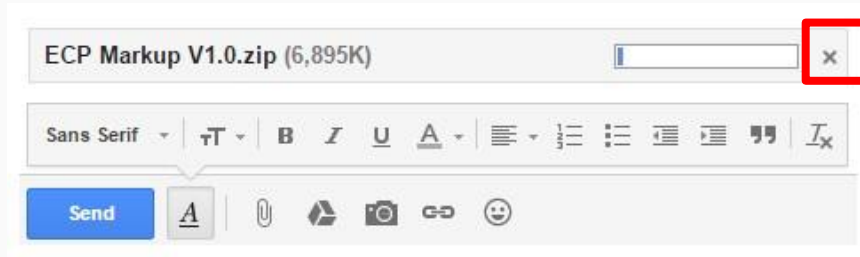


Using real life metaphor in computer application.

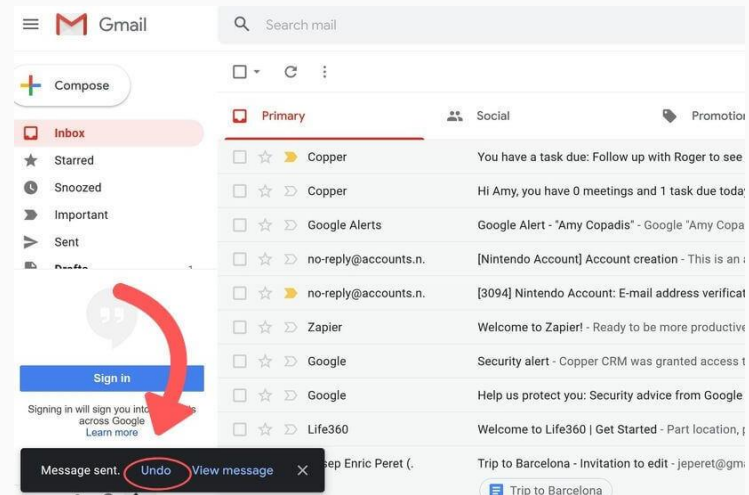
3. User control and freedom

Offer users a digital space where **backward steps are possible**, including **undoing** and **redoing** previous actions.

Users often choose system functions by mistake and will need a clearly marked **emergency exit** to leave the unwanted state without having to go through an extended dialogue. Support **undo** and **redo**.



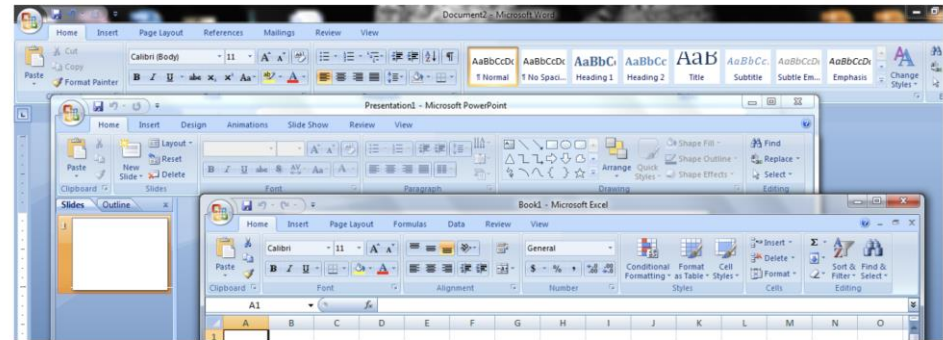
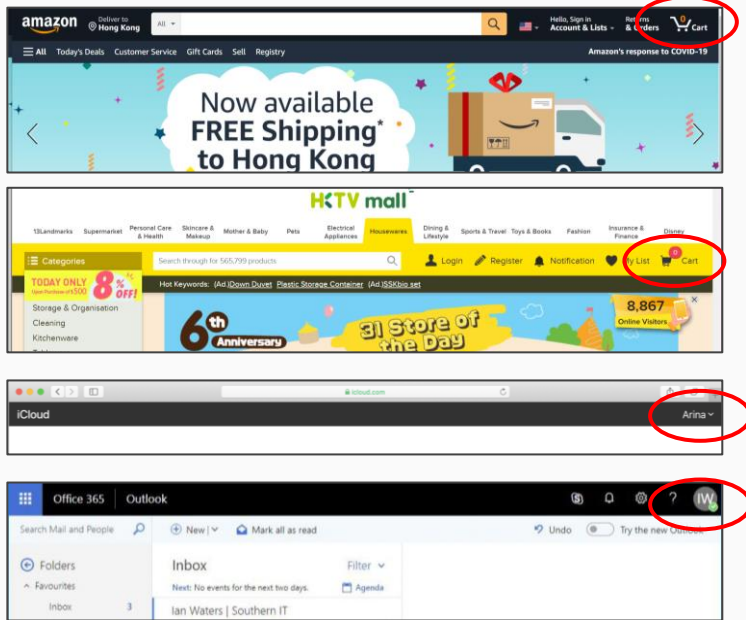
If you attached a large file in gmail by mistakenly, you can cancel it before it is fully uploaded.



You can recall an email in Gmail just after sent.

4. Consistency and standards

Interface designers should ensure that both the **graphic elements and terminology are maintained across similar platforms**. For example, an icon that represents one category or concept should not represent a different concept when used on a different screen.



Microsoft Word, Excel, and PowerPoint all use the same style toolbar with the same primary menu options: Home, Insert, Page Layout... Consistency results in efficiency and perceived intuitiveness

You will always find sign-in, profile details and shopping cart information at the top right corner.

5. Error prevention

Whenever possible, design systems so that **potential errors are kept to a minimum**. Users do not like being called upon to detect and remedy problems, which may on occasion be beyond their level of expertise. **Eliminating** or **flagging** actions that may result in errors are two possible means of achieving error prevention.

From	
Fri, Feb 10 2014 ▼	7:30 PM ▼
<hr/>	
Date occurs in the past	
To	
Fri, Feb 10 2014 ▼	11:30 PM ▼
<hr/>	
Date occurs in the past	

5. Error prevention: Two types of errors

- **Slips occur when users intend to perform one action, but end up doing another (often similar) action.**
 - For example, typing an “i” instead of an “o” counts as a slip; accidentally putting liquid hand soap on one’s toothbrush instead of toothpaste is also a slip.
 - Slips are typically made when users are on autopilot, and when they do not fully devote their attention resources to the task at hand.
- **Mistakes are made when users have goals that are inappropriate for the current problem or task; even if they take the right steps to complete their goals, the steps will result in an error.**
 - For example, if I misunderstood the meaning of the oil-pressure warning light in my car, and thought it was the tire-pressure monitor, no matter how carefully I added air to my tires, it would not fix the issue with my oil pressure. This would be a mistake, since the goal that I was attempting to accomplish was inappropriate for the situation, even though I made no errors in executing my plan.
 - Mistakes are **conscious errors**, and often (though not exclusively) arise when a user has incomplete or incorrect information about the task, and develops a mental model that doesn’t match how the interface actually works.

5. Error prevention: Example - mistake

Scheduling a WordPress post

The diagram illustrates a user's workflow in the WordPress 'Publish' menu. On the left, the 'Publish immediately' option is highlighted with a red box, and its 'Edit' link is circled in red. A large red arrow points to the right, where a date/time selection dialog is shown. This dialog is also highlighted with a red box and contains a date picker set to '02-Feb', '04', '2013' and a time selector set to '@ 09 : 00'. Below the date/time fields are 'OK' and 'Cancel' buttons. At the bottom of the right-hand panel, the 'Publish' button is circled in red, indicating the final action taken by the user.

Left Panel (Initial State):

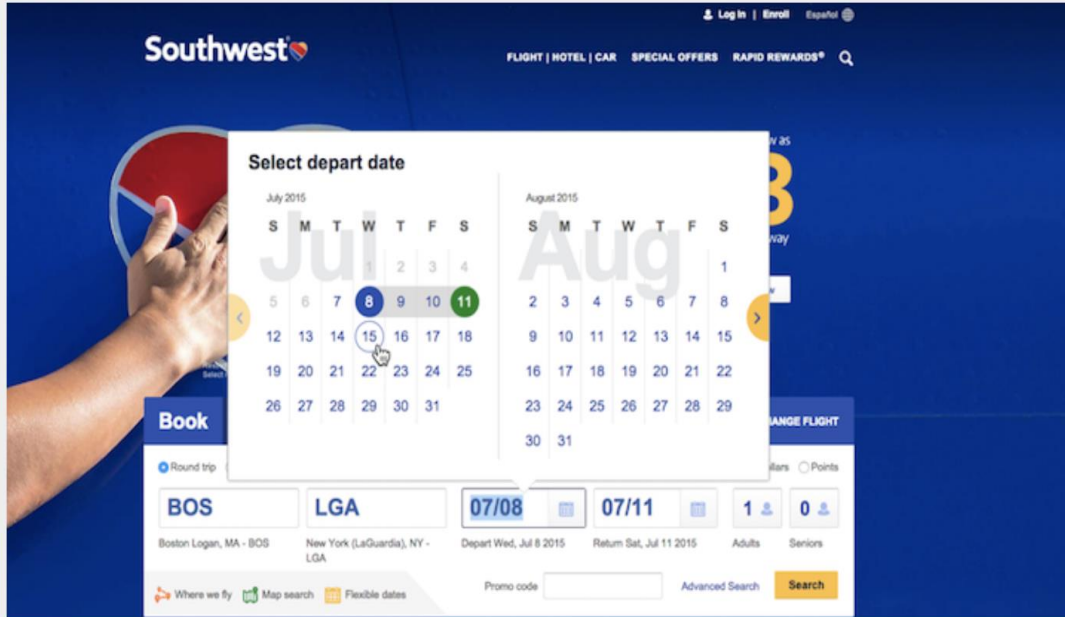
- Buttons: Save Draft, Preview
- Status: Draft [Edit](#)
- Visibility: Public [Edit](#)
- Publish immediately [Edit](#)** (highlighted)
- Publicize: Twitter: @ndunn2 [Edit](#) [Settings](#)
- [Move to Trash](#) **Publish**

Right Panel (After Date/Time Selection):

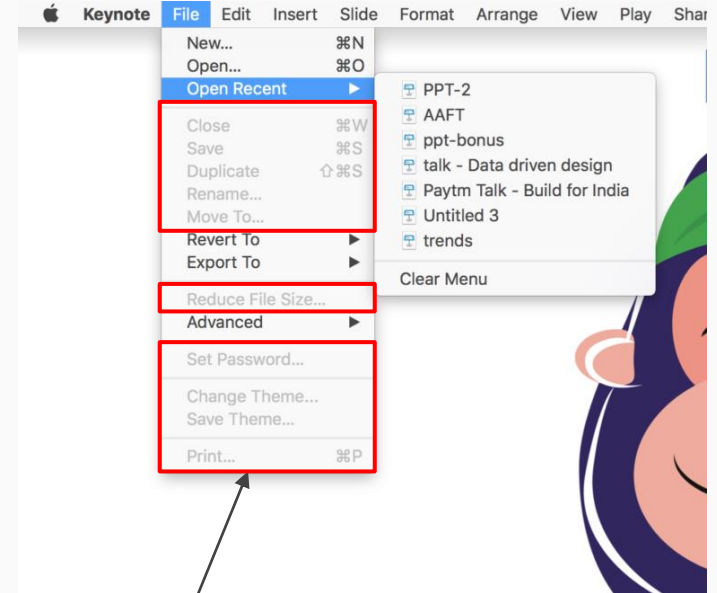
- Buttons: Save Draft, Preview
- Status: Draft [Edit](#)
- Visibility: Public [Edit](#)
- Publish immediately** (highlighted)
 - 02-Feb ▾ 04, 2013 @ 09 : 00
 - OK [Cancel](#)
- Publicize: Twitter: @ndunn2 [Edit](#) [Settings](#)
- [Move to Trash](#) **Publish** (highlighted)

*The user intentionally clicked on the **Publish** button after edit the date/time.*

5. General guidelines for preventing slips - include helpful constraints

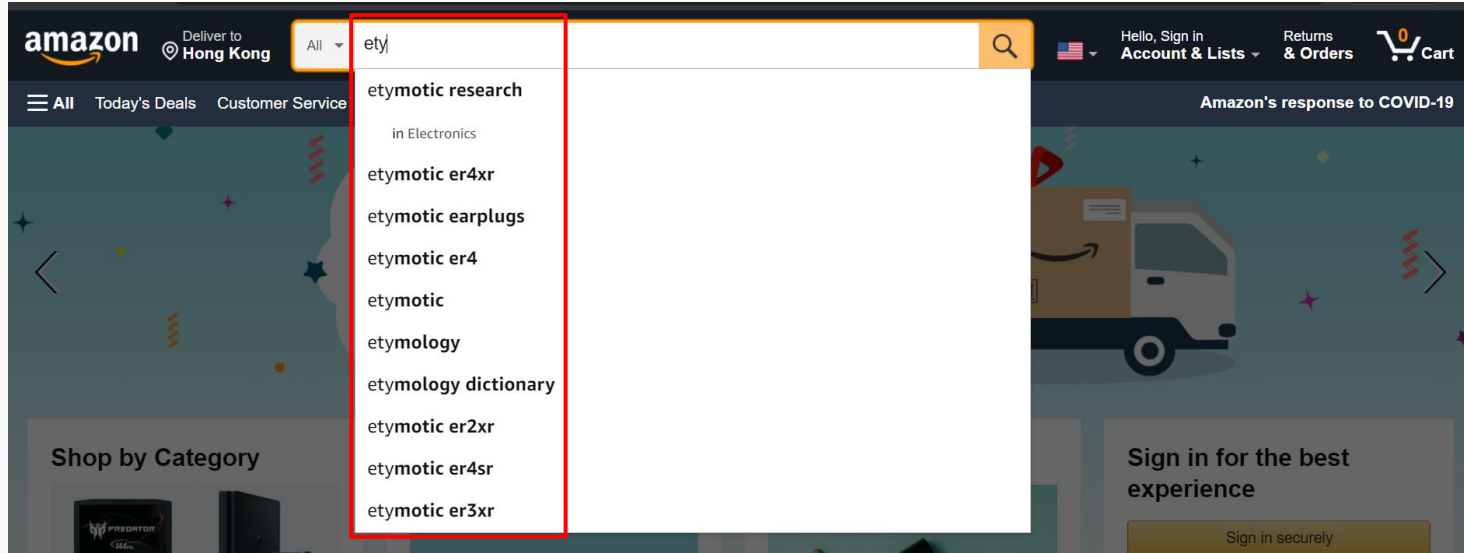


Southwest's calendar widget for picking flight dates uses helpful constraints to prevent users from accidentally setting a nonsensical date range. Even if users attempt to set the return date before the departure date, this widget forces them to pick a departure date first. In addition, it subtly uses color to provide context about which date is about to be changed (in this case, blue for departure), which helps users see which field they are selecting (instead of having to keep that information in their working memory).



Disable menu options that are not applicable

5. General guidelines for preventing slips - offer suggestions



Remembering how to spell Etymotic Research is difficult for users searching for high-quality headphones, and typing is likely to be low accuracy as well. Amazon's clickable search suggestions enable users to type less, thereby making fewer slips or mistakes that would produce no results.

5. General guidelines for preventing slips - choose good defaults

Pre-populates cabin class and travellers fields to what most other users typically select



A flight search form with a dark background. At the top, there are three radio buttons: 'Return' (selected), 'One way', and 'Multi-city'. Below these are four input fields: 'From' (Newcastle (NCL)), 'To' (Country, city or airport), 'Depart' (11/04/2017), and 'Return' (12/04/2017). To the right of these fields is a dropdown menu labeled 'Cabin Class & Travellers' with the text '1 adult, Economy'. Below the input fields are two checkboxes: 'Add nearby airports' and 'Direct flights only'. A green button labeled 'Search flights →' is at the bottom right. A red box highlights the 'Return' radio button and the 'Cabin Class & Travellers' dropdown menu.

Smart default - Pre-select the user's country based on their geolocation data.

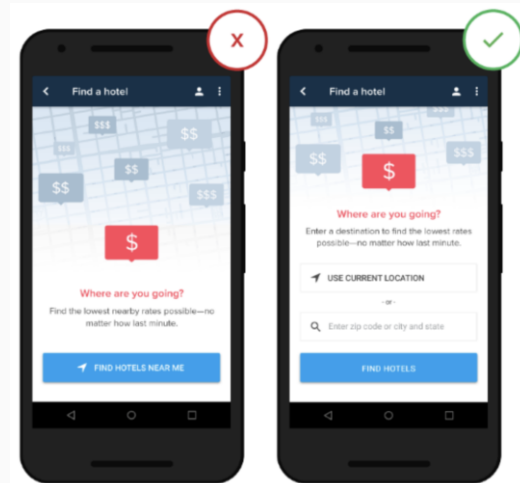
Billing Address

Country *

Australia

First Name *

Last Name *



Make it easy for the user to change default settings

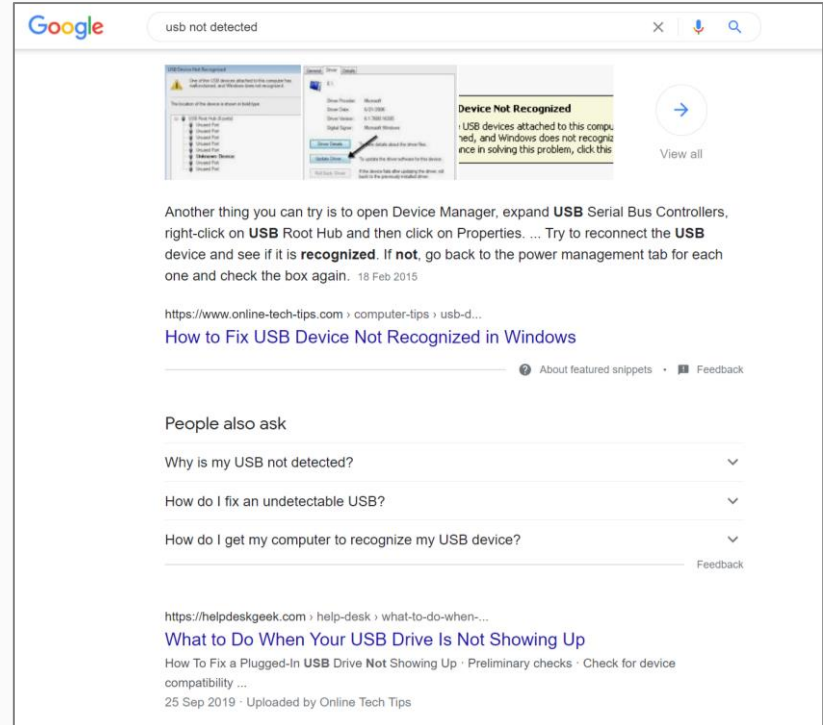
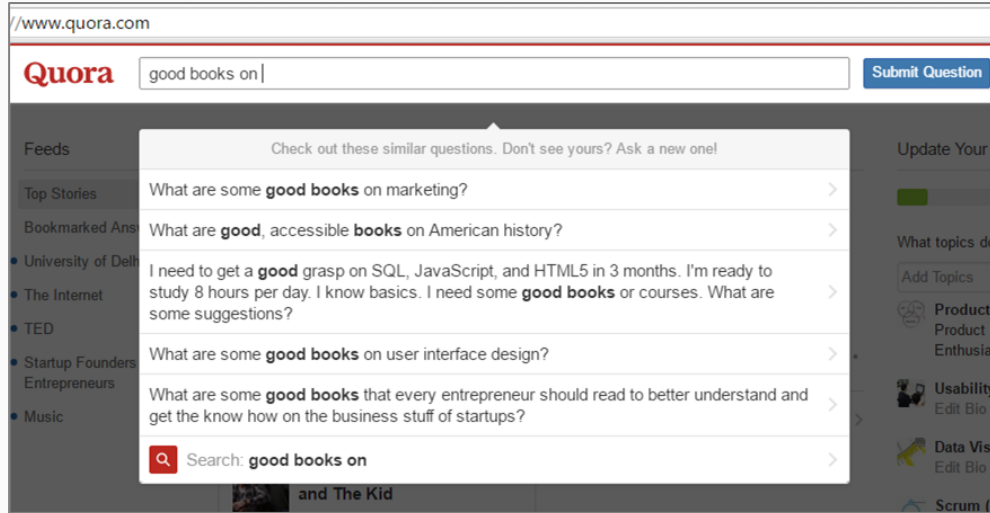
6. Recognition rather than recall

Minimize cognitive load by maintaining task-relevant information within the display while users explore the interface. Human attention is limited and **we are only capable of maintaining around five items in our short-term memory at one time**. Due to the limitations of short-term memory, designers should ensure users can simply employ **recognition instead of recalling information across parts of the dialogue**. Recognizing something is always easier than recall because recognition involves perceiving cues that help us reach into our vast memory and allowing relevant information to surface.

For example, we often find the format of **multiple choice questions easier than short answer questions on a test** because it only requires us to recognize the answer rather than recall it from our memory.

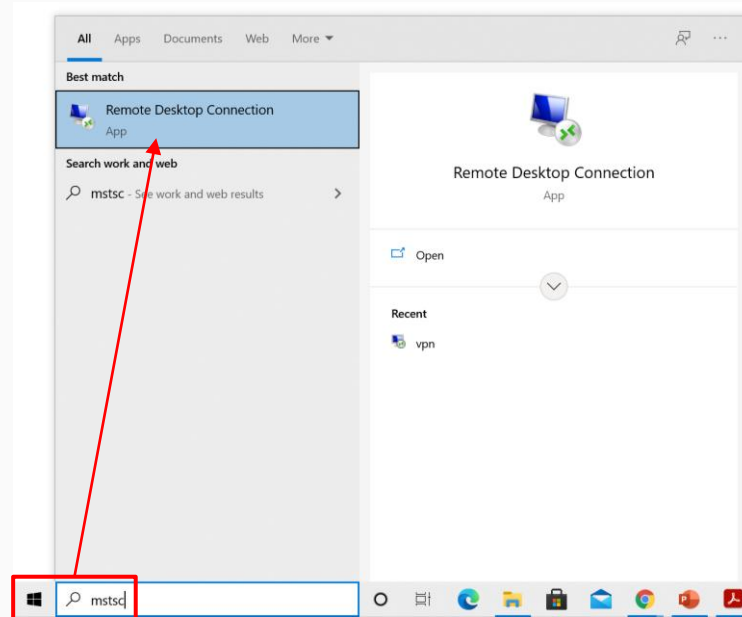
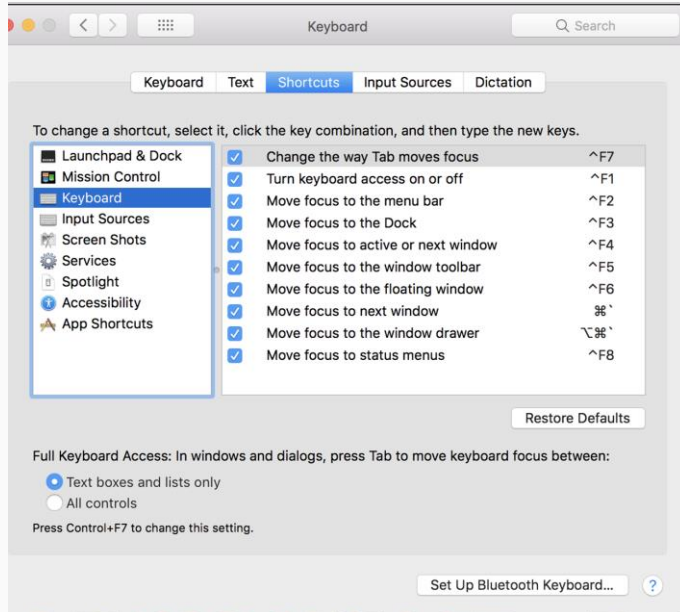
6. Recognition rather than recall: Example

It's always better to suggest the user a set of options than to let him remember and type the whole thing. The goal is to minimize the application of user memory.



7. Flexibility and efficiency of use

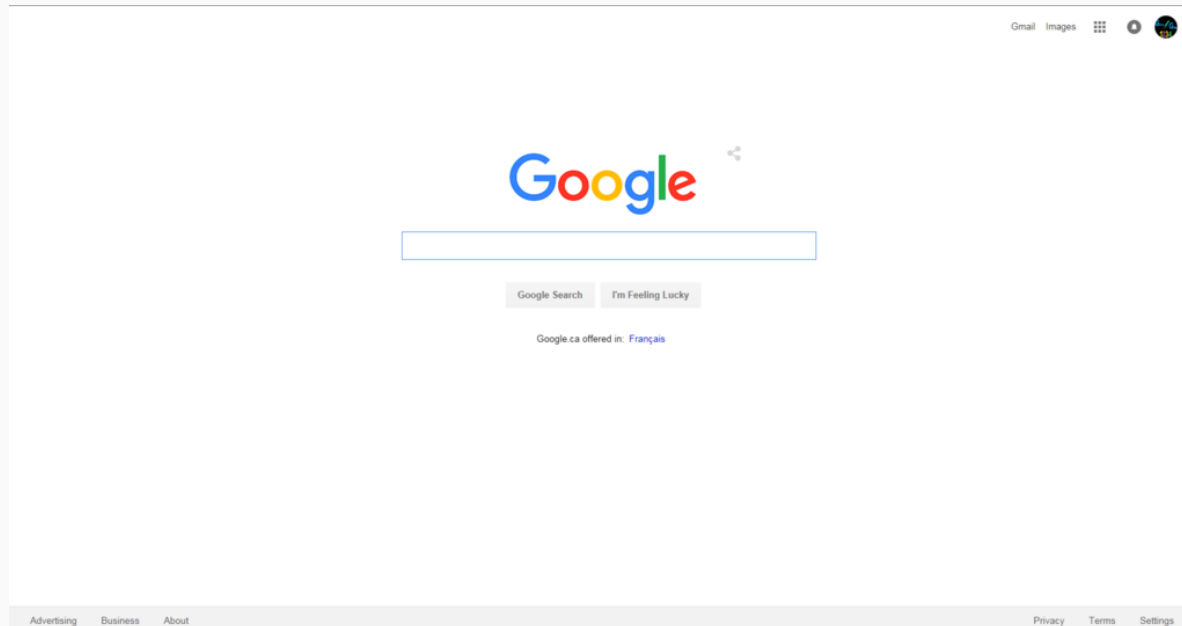
With increased use comes the demand for **less interactions that allow faster navigation**. This can be achieved by using **abbreviations, function keys, hidden commands and macro facilities**. Users should be able to customize or tailor the interface to suit their needs so that **frequent actions can be achieved through more convenient means**.



Save time – Function keys and commands

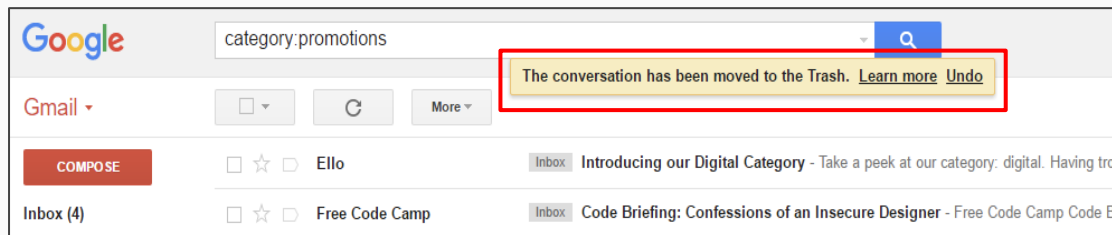
8. Aesthetic and minimalist design


Keep clutter to a minimum. All **unnecessary information competes for the user's limited attentional resources**, which could inhibit user's memory retrieval of relevant information. Therefore, the display must be **reduced to only the necessary components for the current tasks**, whilst providing clearly visible and unambiguous means of navigating to other content.



9. Help users recognize, diagnose, and recover from errors

Designers should assume users are unable to understand technical terminology, therefore, **error messages should almost always be expressed in plain language** to ensure nothing gets lost in translation.



create a new account 

Your Email ✕
Please use a valid email address.

Confirm Email ✕
This does not match the email entered above.

Country ▾

Zip Code OK
SCHENECTADY, NY

Password ✕
Your password must be between 6-16 characters long.

Confirm Password ✕
This does not match the password entered above.

☐ Yes, I agree to the [Terms of Use](#)

10. Help and documentation

Ideally, we want users to **navigate the system without having to resort to documentation**. However, depending on the type of solution, documentation may be necessary. When users require help, ensure it is easily located, specific to the task at hand and worded in a way that will **guide them through the necessary steps towards a solution** to the issue they are facing.

The screenshot shows a web-based wizard titled "Import Users". At the top, a progress bar indicates four steps: "Select File" (green circle with a document icon), "Entity Mapping" (blue circle with a document icon), "Data Mapping" (grey circle with a document icon), and "Summary" (grey circle with a checkmark icon). Below the progress bar, the "Entity Mapping" step is active, showing a description: "Tell us what your excel/csv headers mean". The main area contains four form fields, each with a label and a dropdown menu: "First Name" (Select an Entity), "Email" (Select an Entity), "DOB" (Select an Entity), and "Grade" (Select an Entity). The step indicator "Step 2 of 4" is visible in the top right corner of the form area.



Wizard presents a series of steps or conditions that the user needs to complete in order to accomplish a goal (e.g. use a product).

Reference

<http://www.cs.umd.edu/~ben/goldenrules.html>

<https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb>