

Building User Interfaces

Expert-based Evaluation

Yixin Hu & Cole Nelson

Please make a copy of today's ICA

What we will learn today?

- What are rapid/expert methods?
- Heuristic evaluation
- In-class activity
- Cognitive walkthrough

Usability Evaluation

Definition: The assessment of the effectiveness, efficiency, and user satisfaction with design solutions.

Types of Usability Evaluation

1. Testing-based methods
2. Expert-review-based methods

Testing-based methods

Definition: Empirical, *i.e.*, based on data, testing with users who represent the target population of design solutions.

Expert-review-based methods

Definition: Also called *usability inspection*, review-based evaluation by experts who follow well-established protocols to inspect the usability of design solutions.

Which methods?

Heuristic evaluation, cognitive walkthrough, pluralistic walkthrough, feature inspection, consistency inspection, standards inspection, formal usability inspection.

We will cover the most commonly use two methods.

Heuristic Evaluation

Heuristic Evaluation

Definition: Developed by Jacob Nielsen, heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").¹ ²

¹ NN/g: [How to conduct a heuristic evaluation](#)

² NN/g: [Video explanations of the 10 heuristics](#)

NN/g

Heuristic
Evaluation



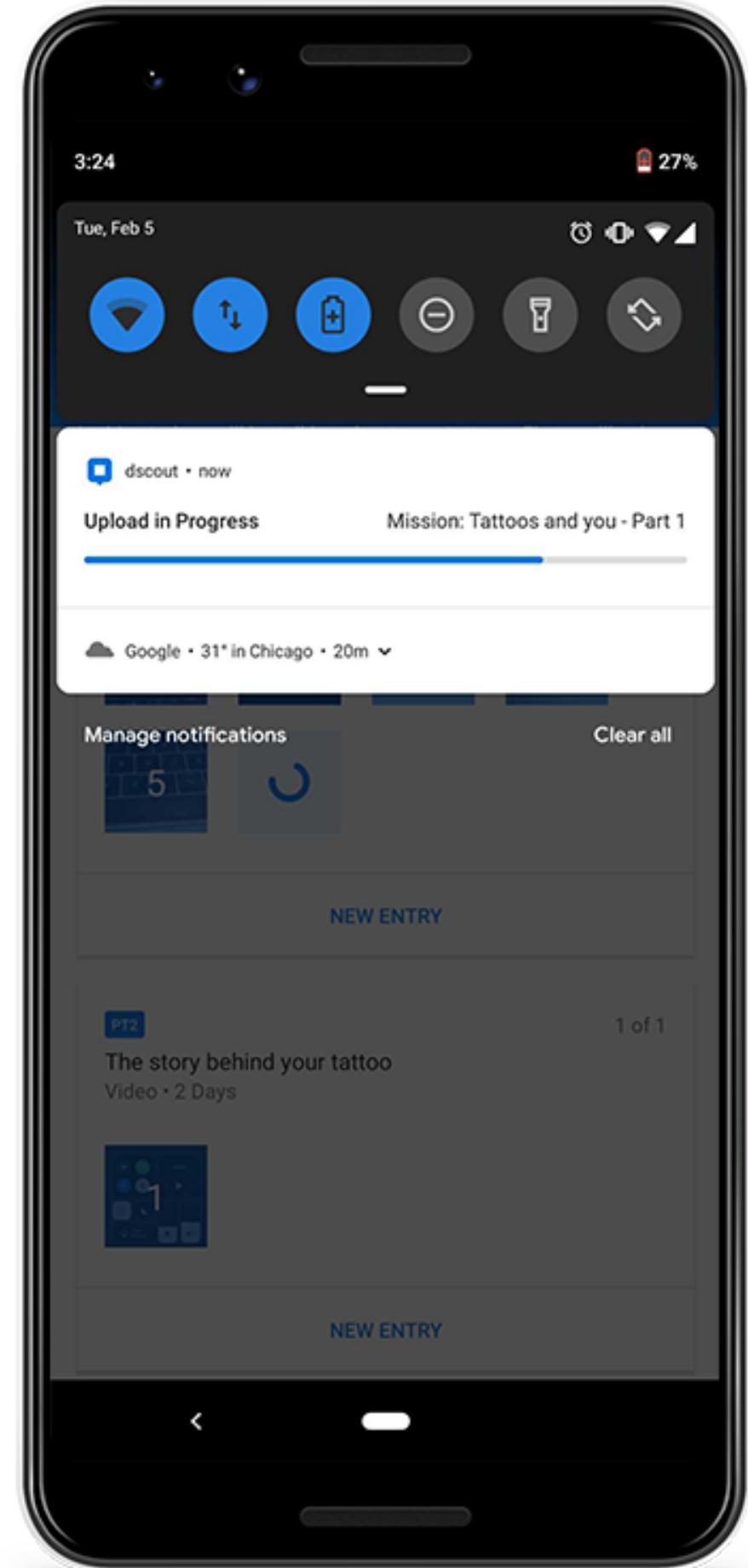
1: Visibility of system status³ ⁴

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

E.g., email clients making a swoosh sound when sending email.

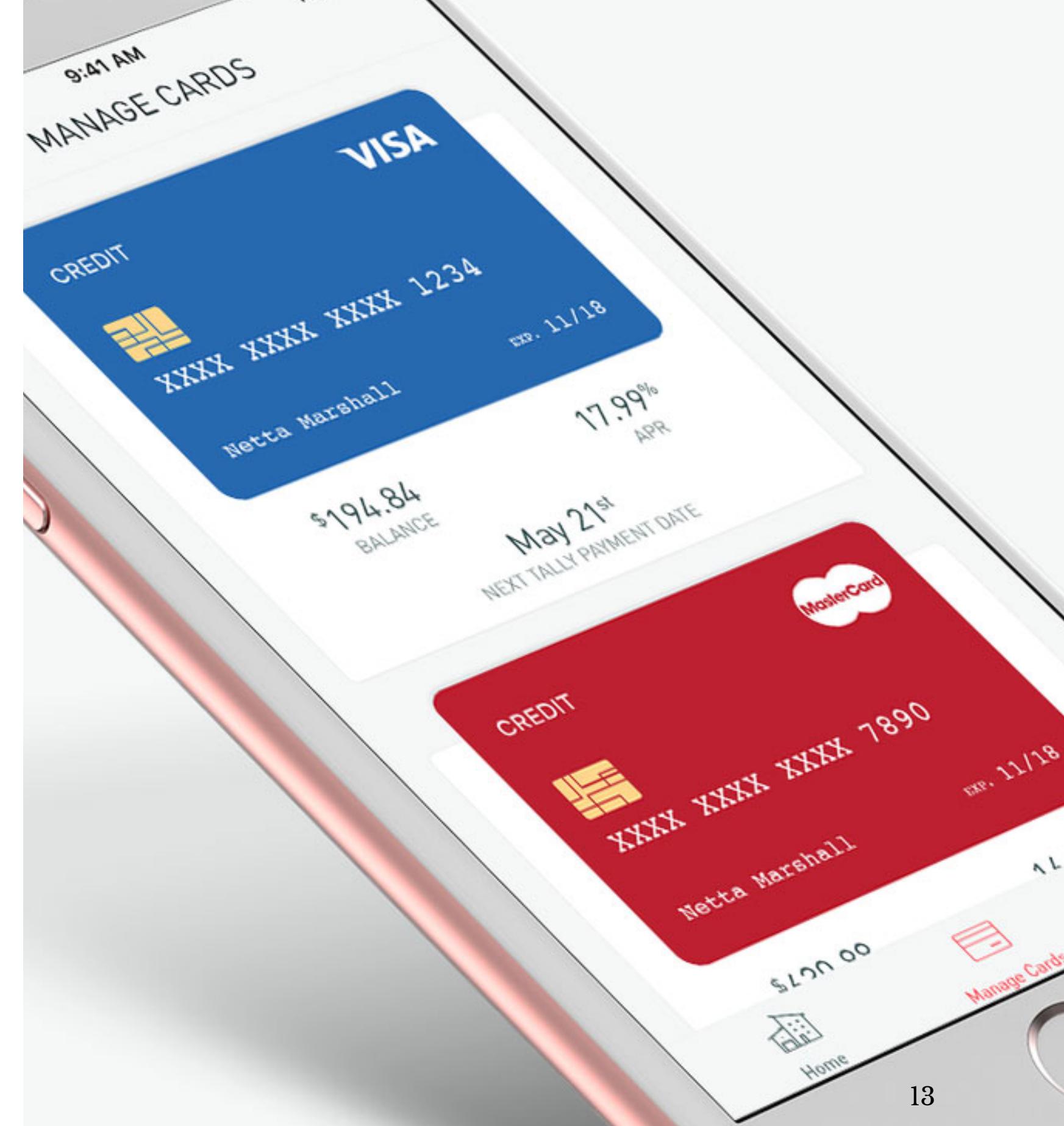
³ NN/g: [Visibility of system status](#)

⁴ [Image source](#)



2: Match between system and the real world⁵ ⁶

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.



⁵ NN/g: Match between system and the real world

⁶ Image source

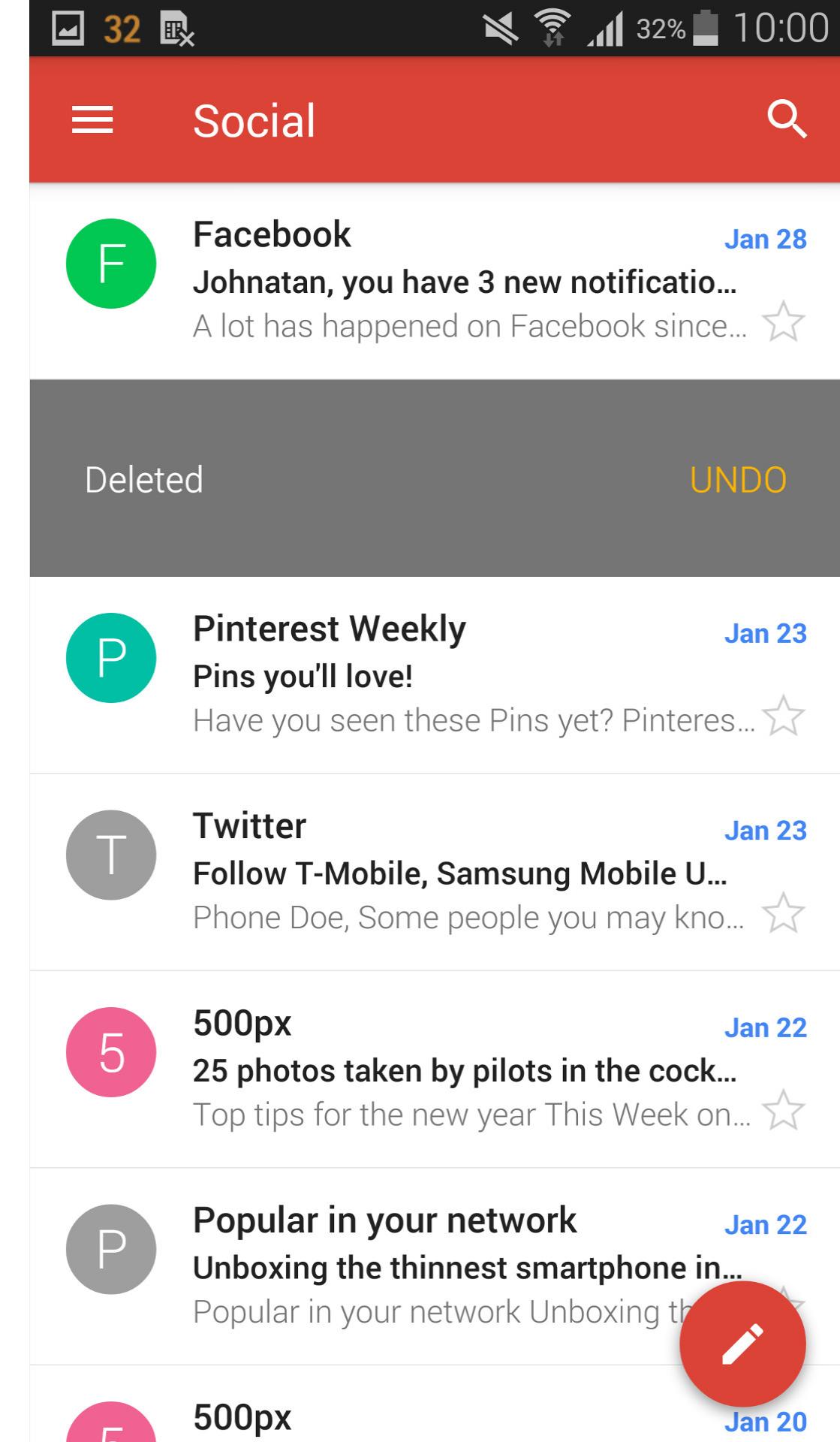
3: User control and freedom^{7 8}

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.

E.g., undo for delete/archive in email clients

⁷ NN/g: User control and freedom

⁸ image source



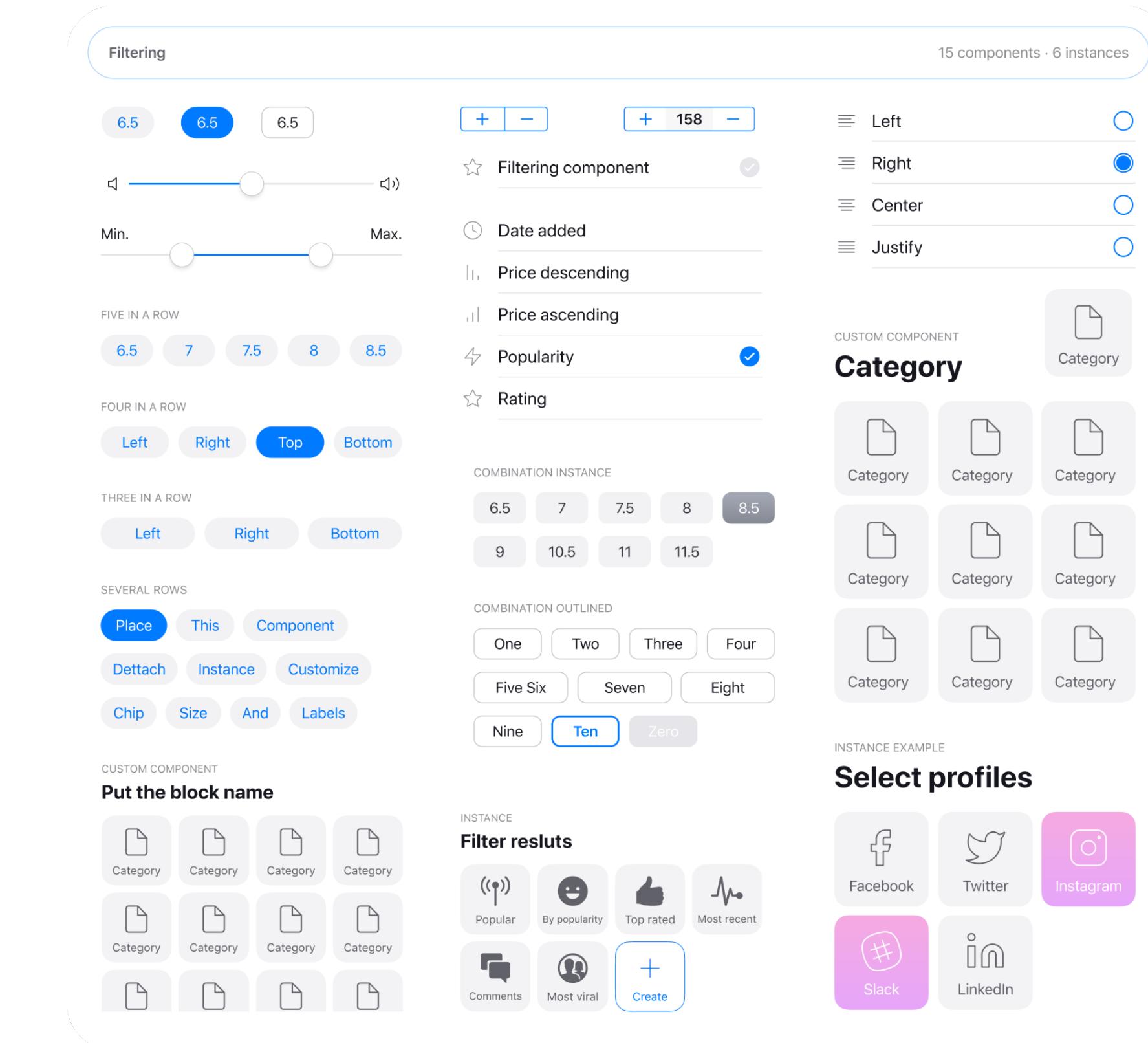
4: Consistency and standards⁹ ¹⁰

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

E.g., component libraries to achieve consistency within an app; platform conventions to achieve consistency across apps.

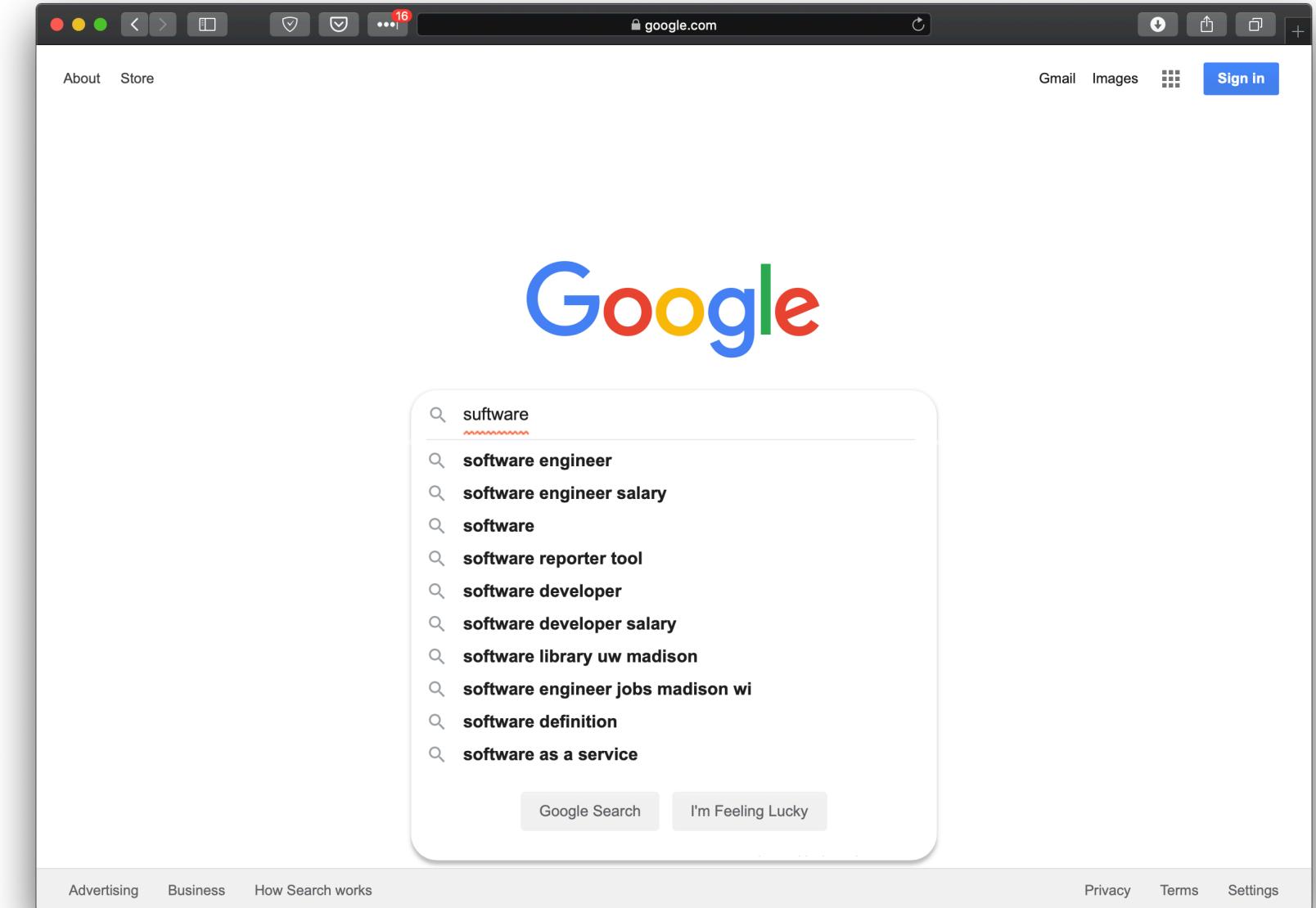
⁹ NN/g: [Consistency and standards](#)

¹⁰ [Image source](#)



5: Error prevention¹¹

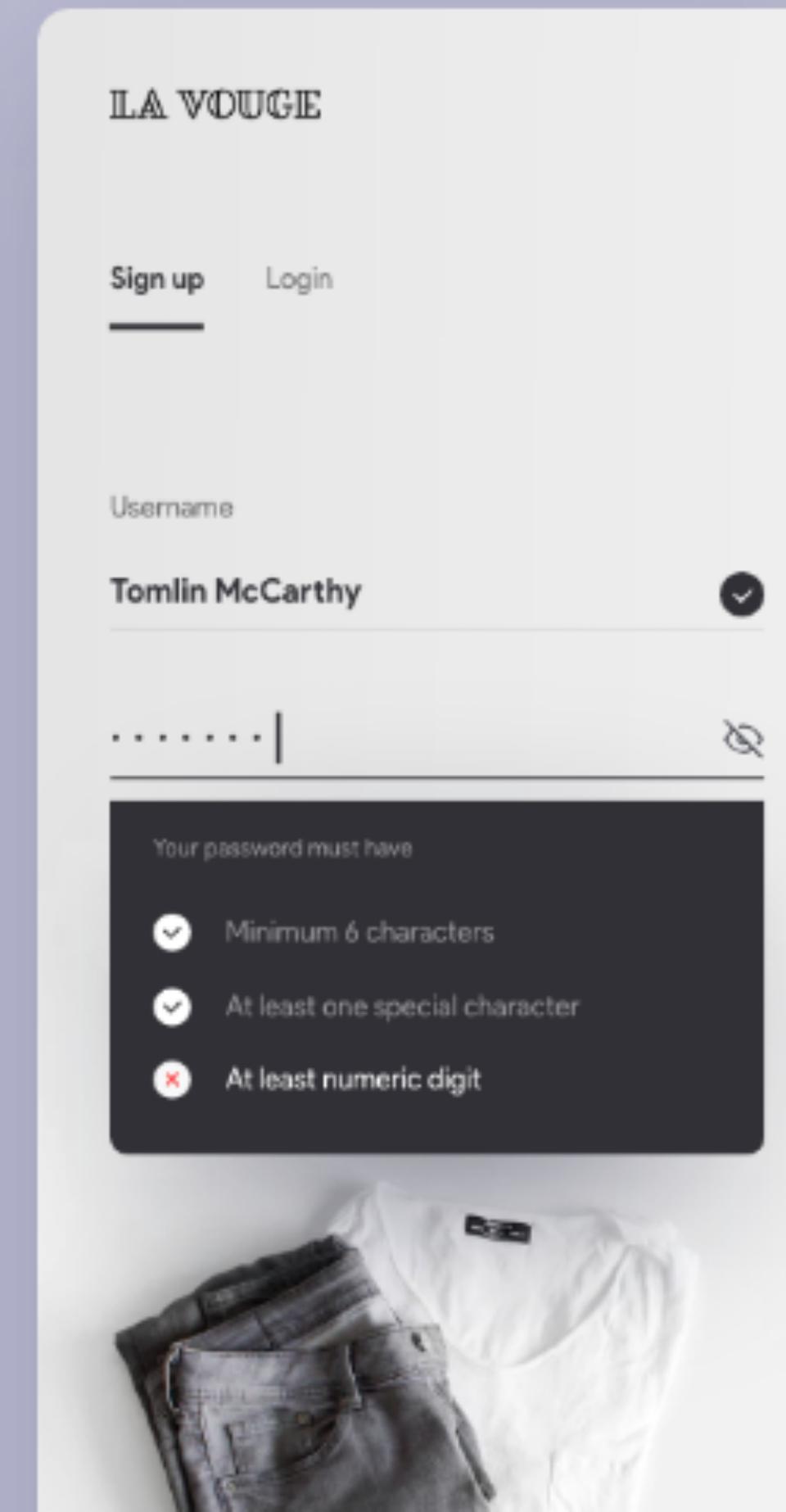
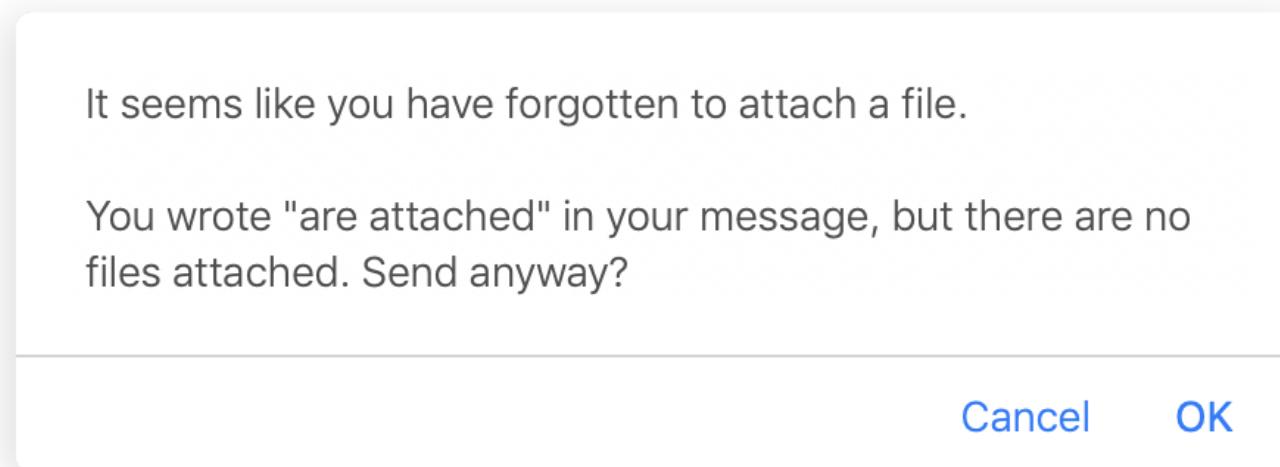
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.



¹¹ NN/g: Error prevention

Examples:¹²

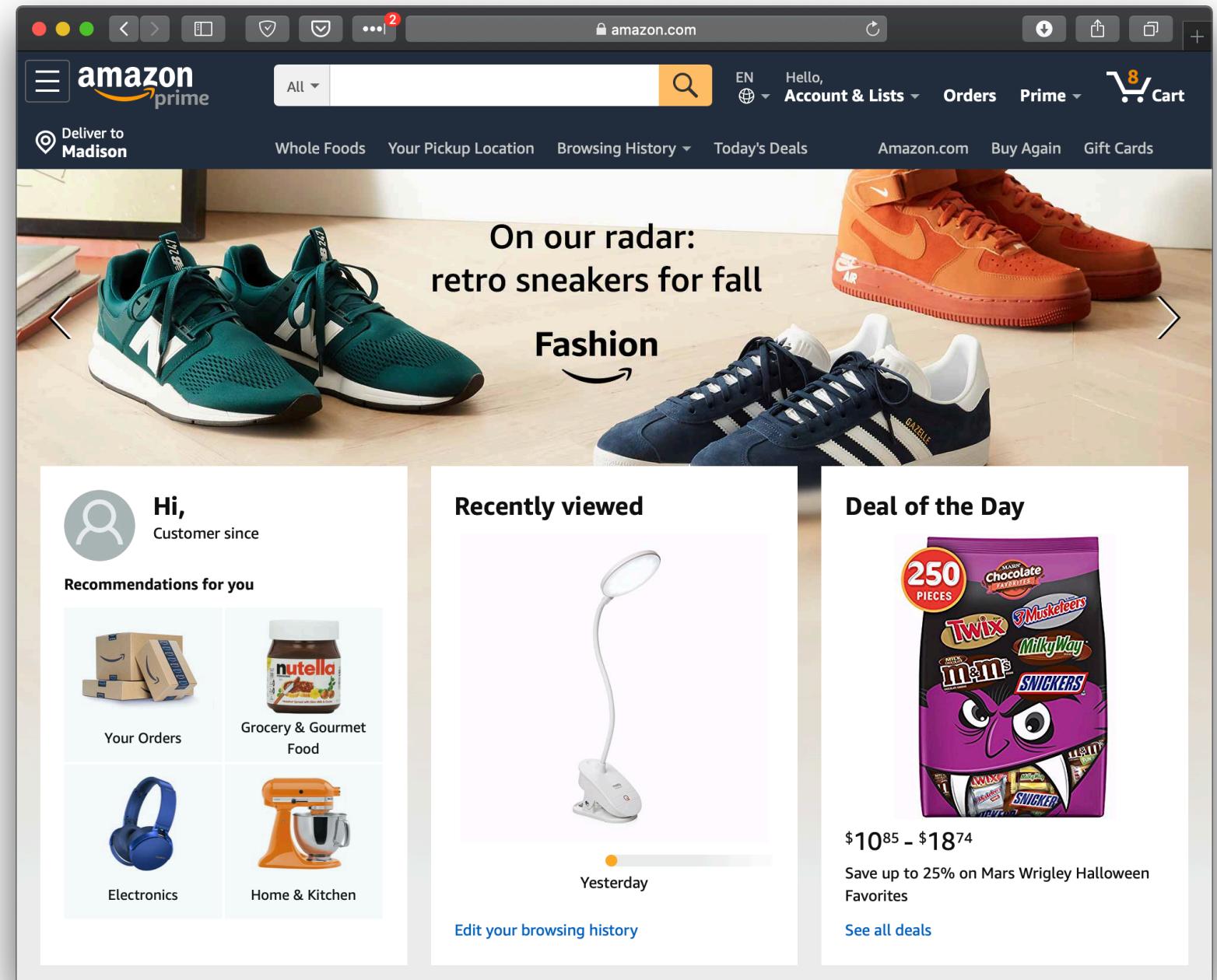
- Autocorrect in search
- Real-time feedback on new user names, password strength, etc.
- Attachment reminders in email clients



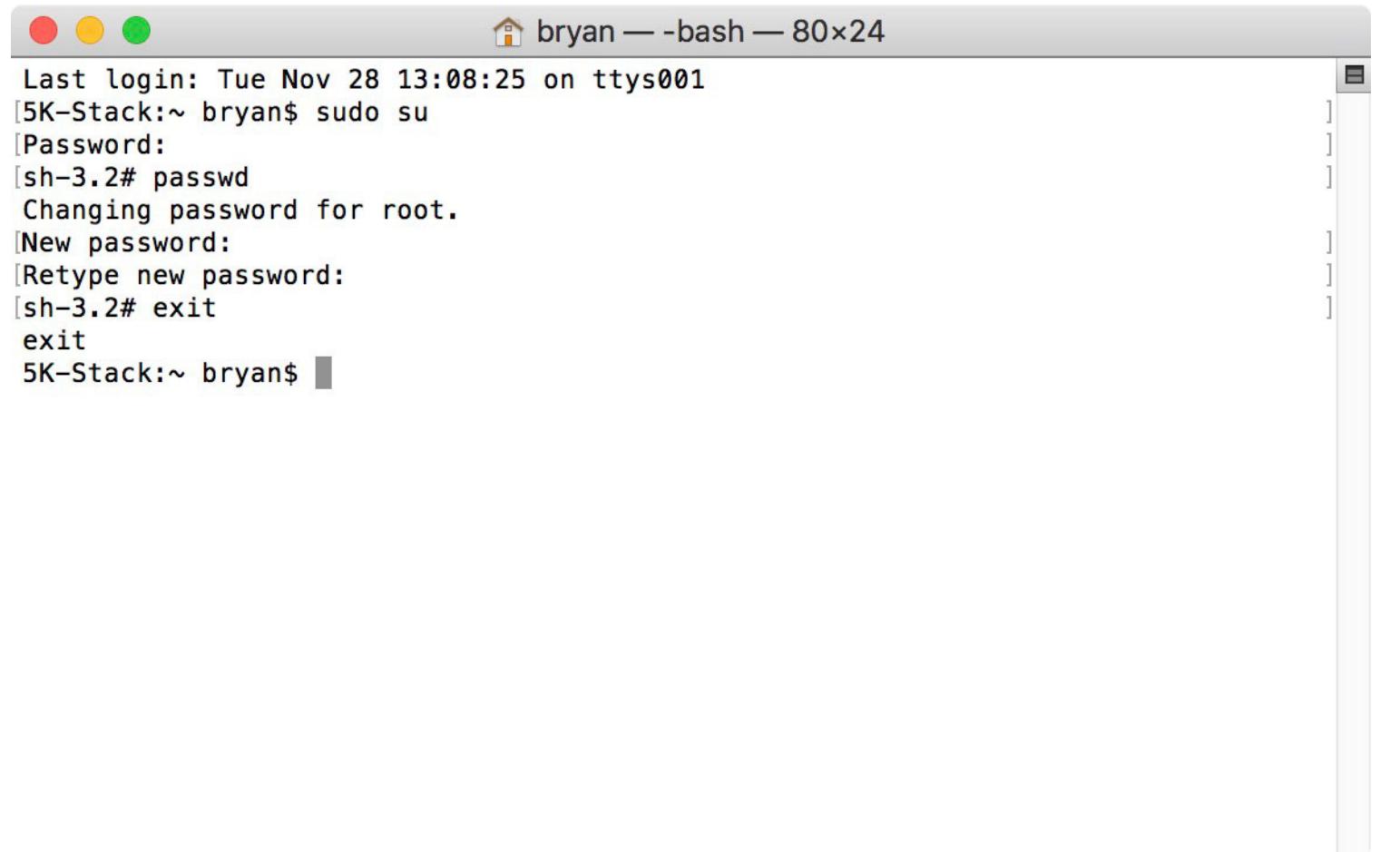
¹² Image source: [Left](#), [Right](#)

6: Recognition rather than recall¹³

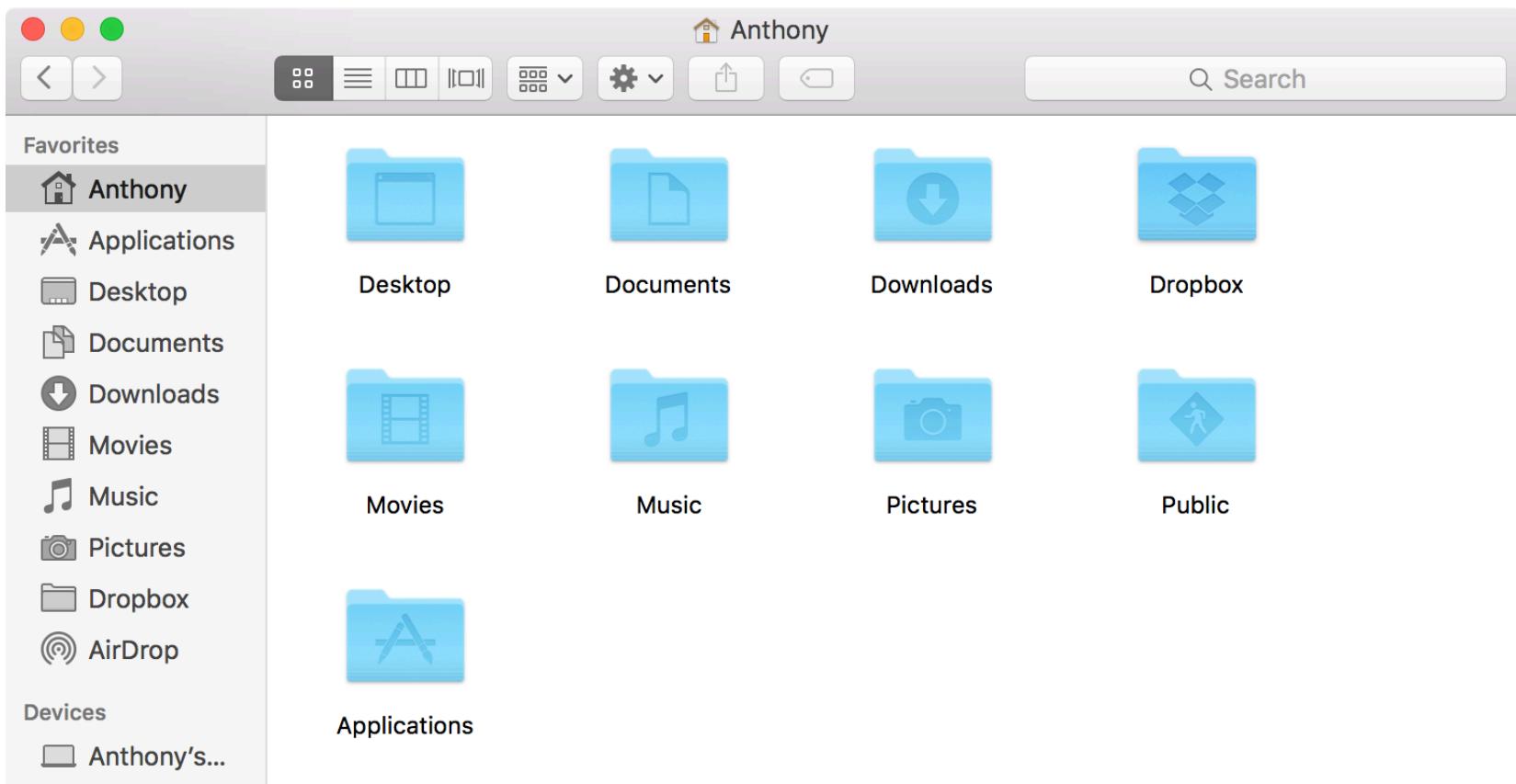
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.



¹³ NN/g: Recognition rather than recall



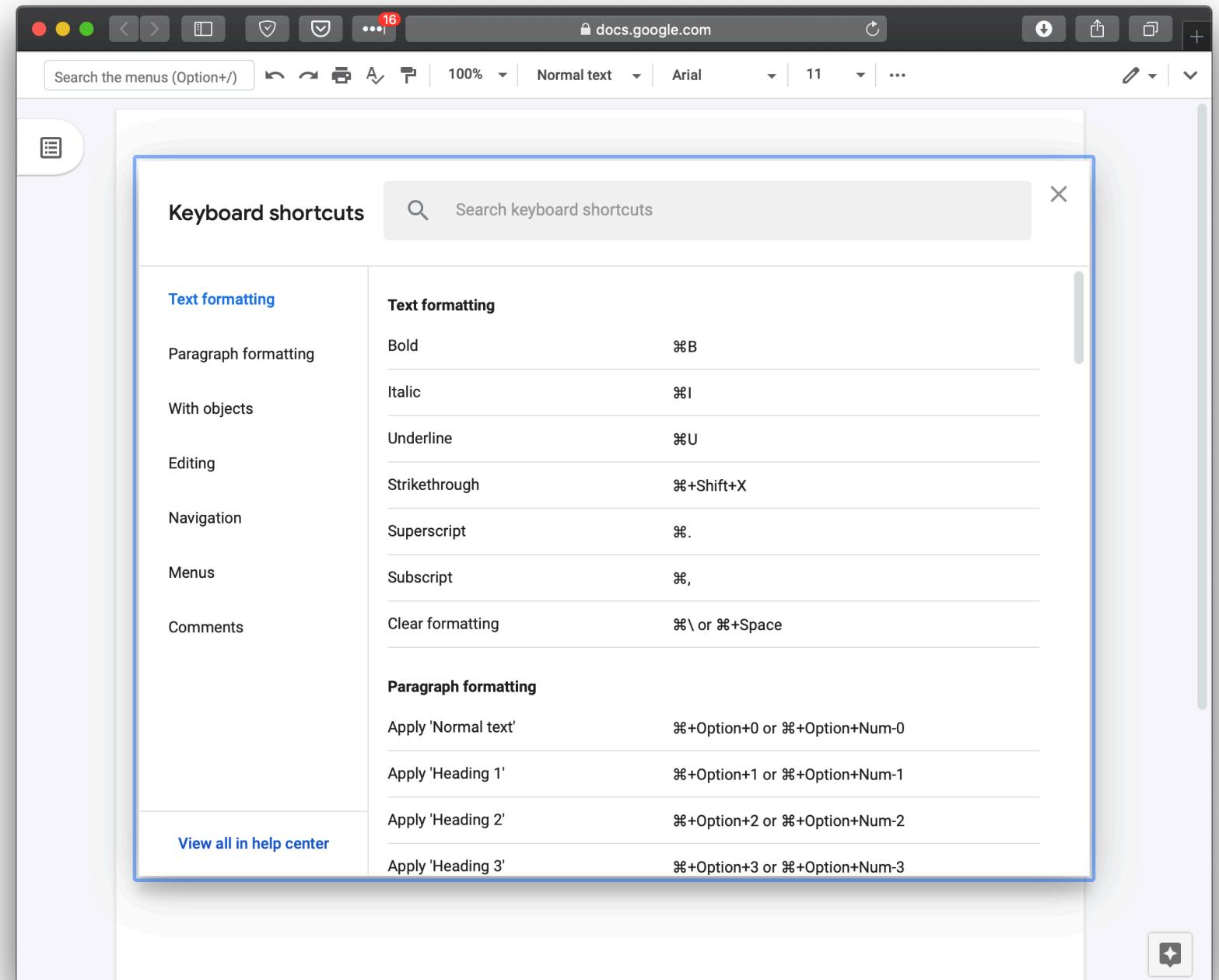
```
Last login: Tue Nov 28 13:08:25 on ttys001
[5K-Stack:~ bryan$ sudo su
>Password:
[sh-3.2# passwd
Changing password for root.
>New password:
Retype new password:
[sh-3.2# exit
exit
5K-Stack:~ bryan$
```



¹⁴ Image source: [Left](#), [Right](#)

7: Flexibility and efficiency of use¹⁵

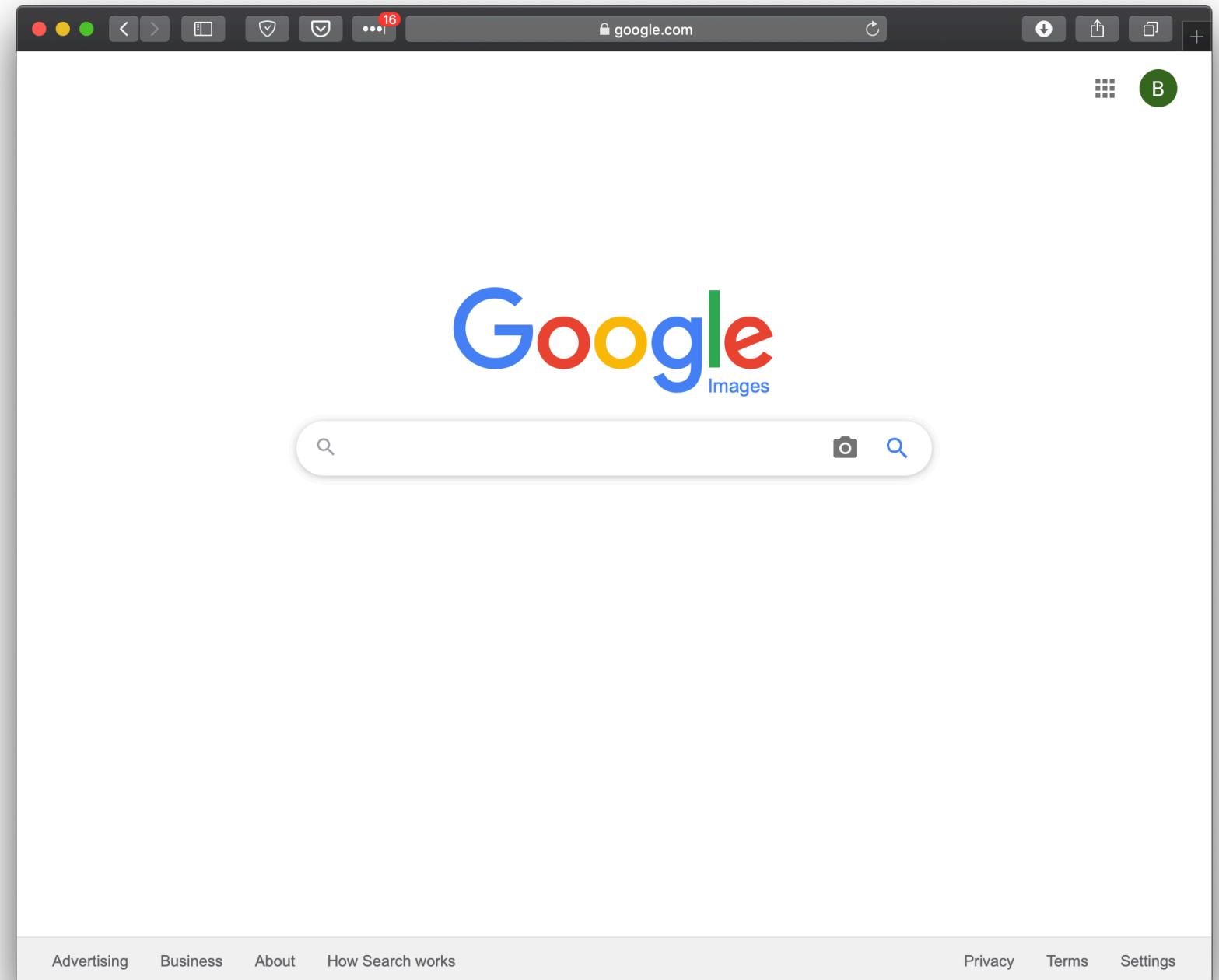
Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.



¹⁵ NN/g: Flexibility and efficiency of use

8: Aesthetic and minimalist design¹⁶

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.



¹⁶ NN/g: Aesthetic and minimalist design



Search the Web...

Login

Home ▾

Reader

Bookmarks

Menu ▾

Notes

You can start adding notes here.

You are not logged in. Please login and try again

Bookmarks

Add a bookmark

Easily store and organize bookmarks for your favorite sites. Import Bookmarks

from chrome, firefox, safari, internet explorer etc.

Google Links

Gmail	Calendar	Photos	YouTube
Maps	Blogger	News	Translate
Drive	Play	Books	Shopping

Google Calendar

Today December 2023 Week Month

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	Dec 1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17

CNN.com - Top Stories

[Hundreds of millions at risk from Chinese shopping app malware](#)
1 month ago

[Russian authorities detain suspect over St. Petersburg cafe blast](#)
1 month ago

[READ: Trump indictment related to hush money payment](#)

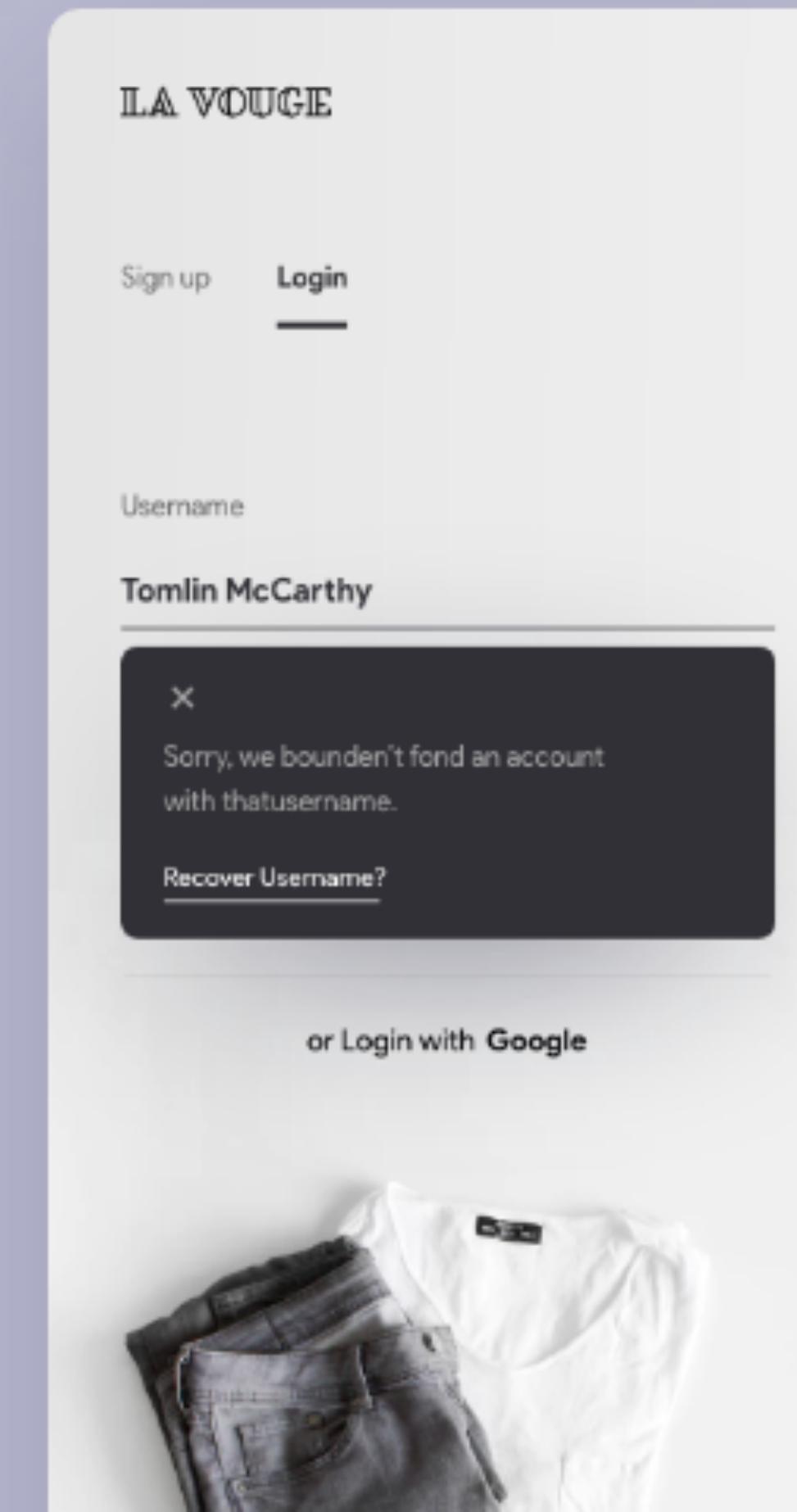
UK news | The Guardian

[Greedflation: corporate profiteering 'significantly' boosted global prices, study shows](#)

Multinationals in particular hiked prices far above rise in costs to deliver an outsize impact on cost of living crisis, report concludes

9: Help users recognize, diagnose, and recover from errors¹⁷ ¹⁸

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

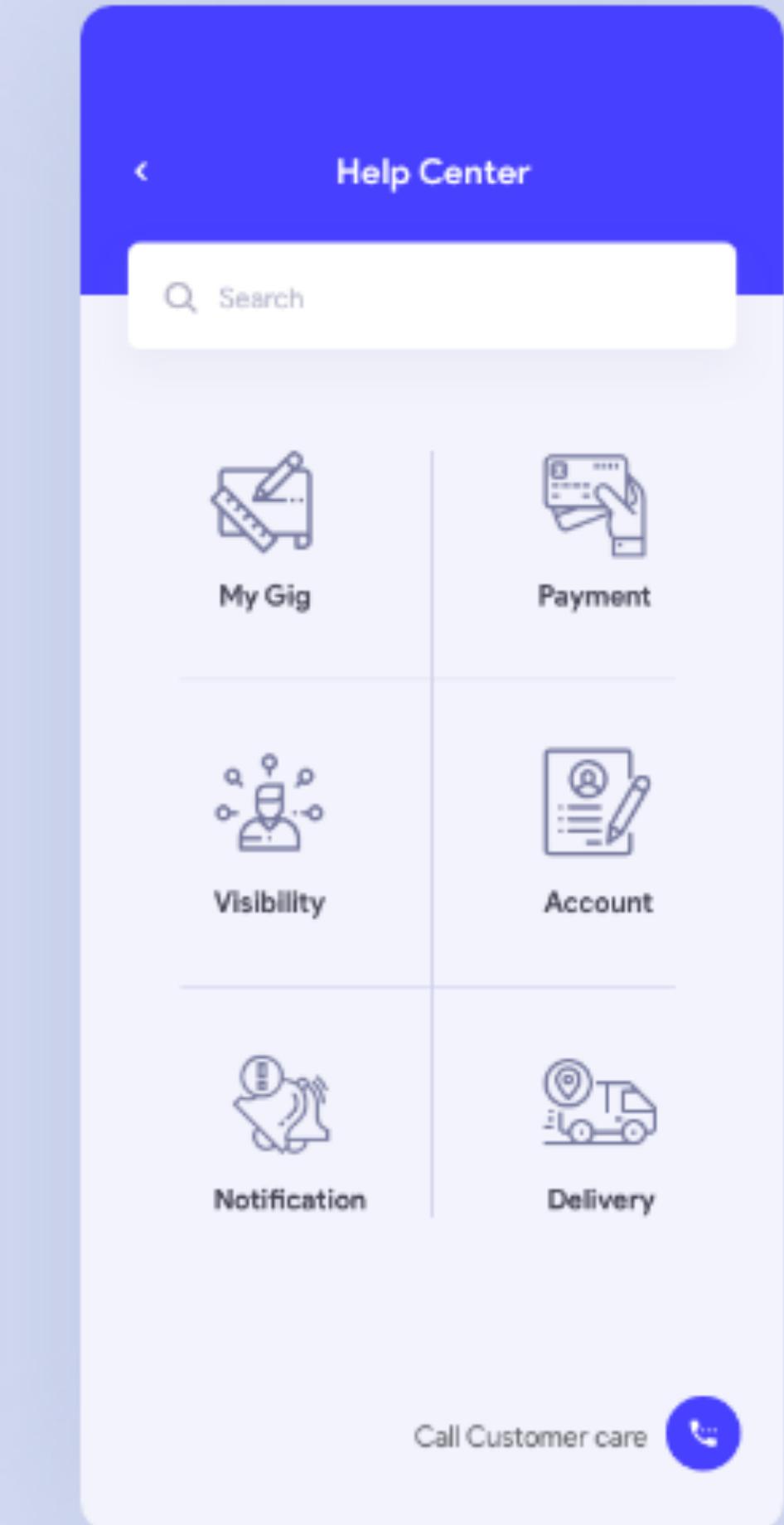


¹⁷ NN/g: Help users recognize, diagnose, and recover from errors

¹⁸ Image source

10: Help and documentation¹⁹ ²⁰

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

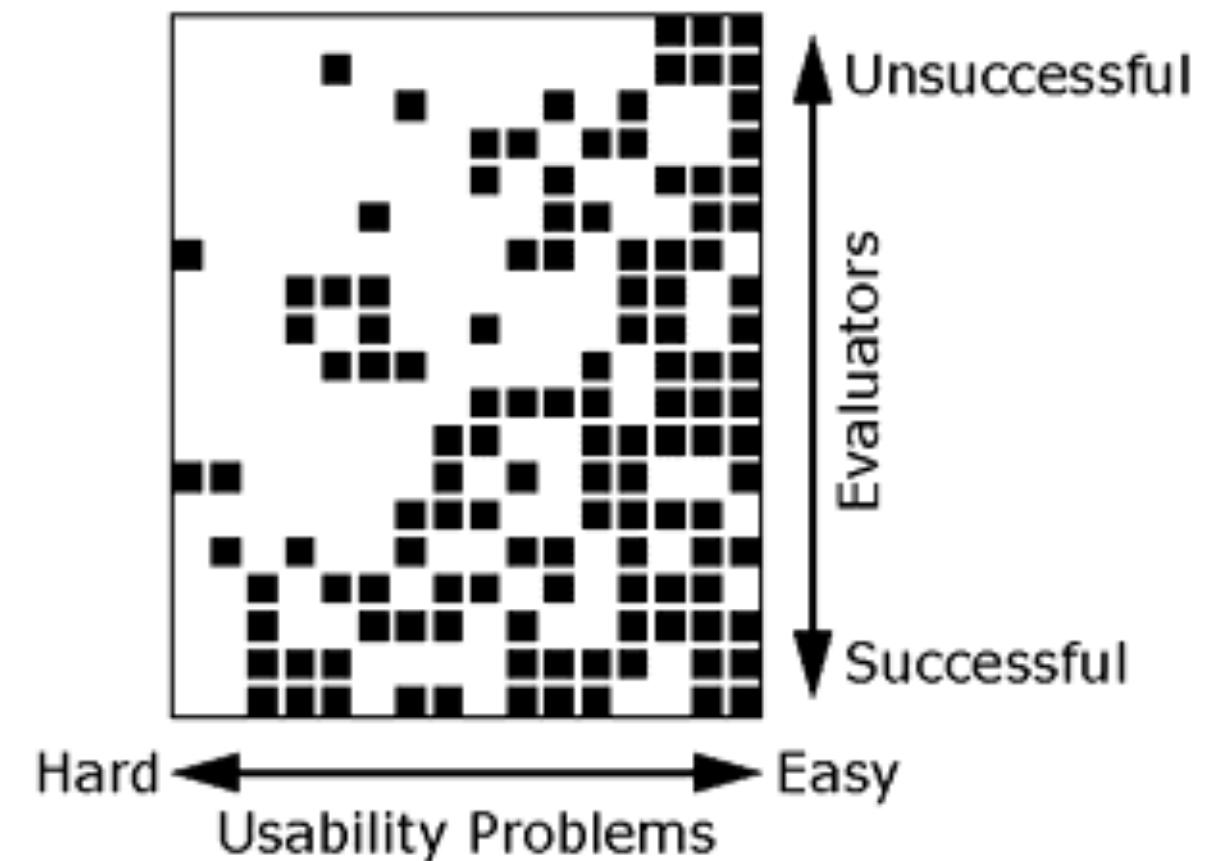


¹⁹ NN/g: [Help and documentation](#)

²⁰ [Image source](#)

Process²¹

1. Identify 3–5 usability experts with domain knowledge; determine the heuristics to use;
2. Each inspector individually reviews as feature/screen/page for each heuristic;
3. Inspectors merge and prioritize their findings, brainstorm solutions, report conclusions.



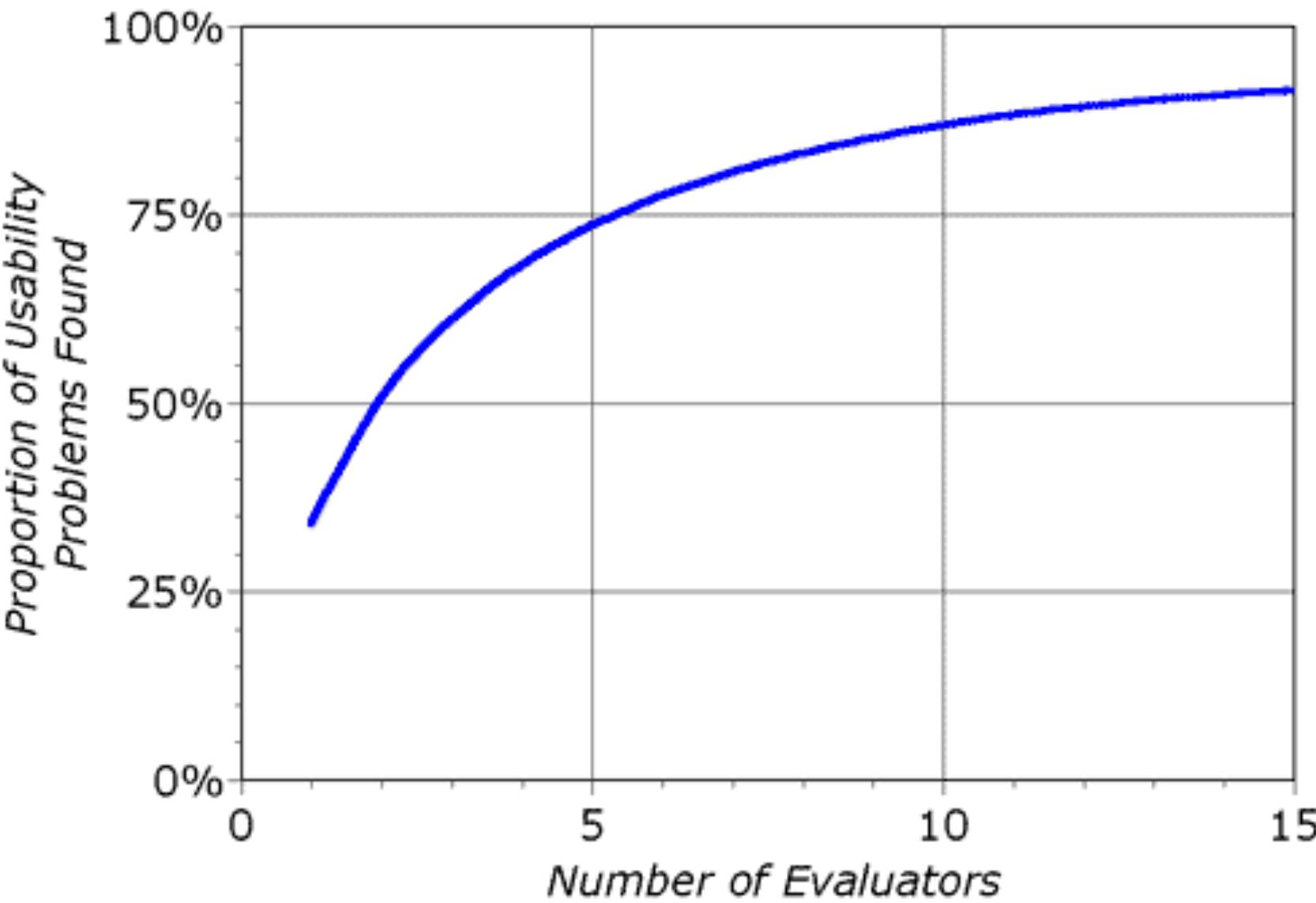
²¹NN/g: [How to Conduct a Heuristic Evaluation](#)

Recap: How many evaluators are needed?

The rule of thumb is 5.²²

$$ProblemsFound(i) = N(1 - (1 - l)^i)$$

- i independent evaluators
- N total number of usability problems in the interface
- l the proportion of all usability problems found by a single evaluator



²² [Image source](#)

Heuristic Evaluation Reporting²³

Definition: A document that highlights the top three to five usability problems and suggested solutions.

²³ A good heuristics evaluation report example

Heuristic evaluation reports usually include:

1. Prototype screen, page, location of the problem
2. Name of heuristic
3. Reason for reporting as negative or positive
4. Scope of problem
5. Severity of problem (high/medium/low)
6. Justification of severity rating
7. Suggestions to fix
8. Possible trade-offs (why the fix might not work)

Severity Ratings

Code	Category	Recommendation
4	Usability catastrophe	<i>Imperative to fix before product can be released</i>
3	Major usability problem	<i>Important to fix, so should be given high priority</i>
2	Minor usability problem	<i>Fixing this should be given low priority</i>
1	Cosmetic problem only	<i>Need not be fixed unless extra time is available on project</i>
0	~	<i>Team does not agree that issue impacts system usability</i>

Heuristic evaluation sheet template

Report the violations you identified in a table and mark them on screenshots.

Determine severity.

Usually, heuristic evaluation sheets are used for this step. We will use a simpler table.

Heuristic Evaluation Sheet		Device	Severity
Evaluator	Date	Browser/OS	0 I don't agree that this is a usability problem at all
Website/App	Task/Feature	Task/Feature	1 Cosmetic problem only: need not be fixed unless extra time is available on project
1. Visibility of system status		Issues Please be specified	2 Minor usability problem: fixing this should be given low priority
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.			3 Major usability problem: important to fix, so should be given high priority
Severity			4 Usability catastrophe: imperative to fix this before product can be released
<input type="radio"/> 0 <input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4			
2. Match between system and the real world		Issues Please be specified	Recommendation
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.			
Severity			
<input type="radio"/> 0 <input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4			

Credit by Hsin-Jou Lin



→ linke

Pros & Cons of Heuristic Evaluation

Pros:

- Inexpensive and intuitive
- Can be used frequently and any time during the design process
- Effective at early stages of design
- Serves as a training tool for designers

Cons:

- Does not capture all aspects of usability
- Does not provide a comprehensive understanding of the interaction
- Might discourage user testing
- May result in false positives

Further Reading on Heuristic Evaluation

- UX Collective article with great examples
- Videos and articles by Jacob Nielsen

ICA E: Expert Evaluation

Perform your own heuristic evaluation!

Cognitive Walkthrough

Cognitive Walkthrough

Definition: Expert review method where a usability specialist assesses the *learnability* and *discoverability* of a design by posing and answering a set of questions.

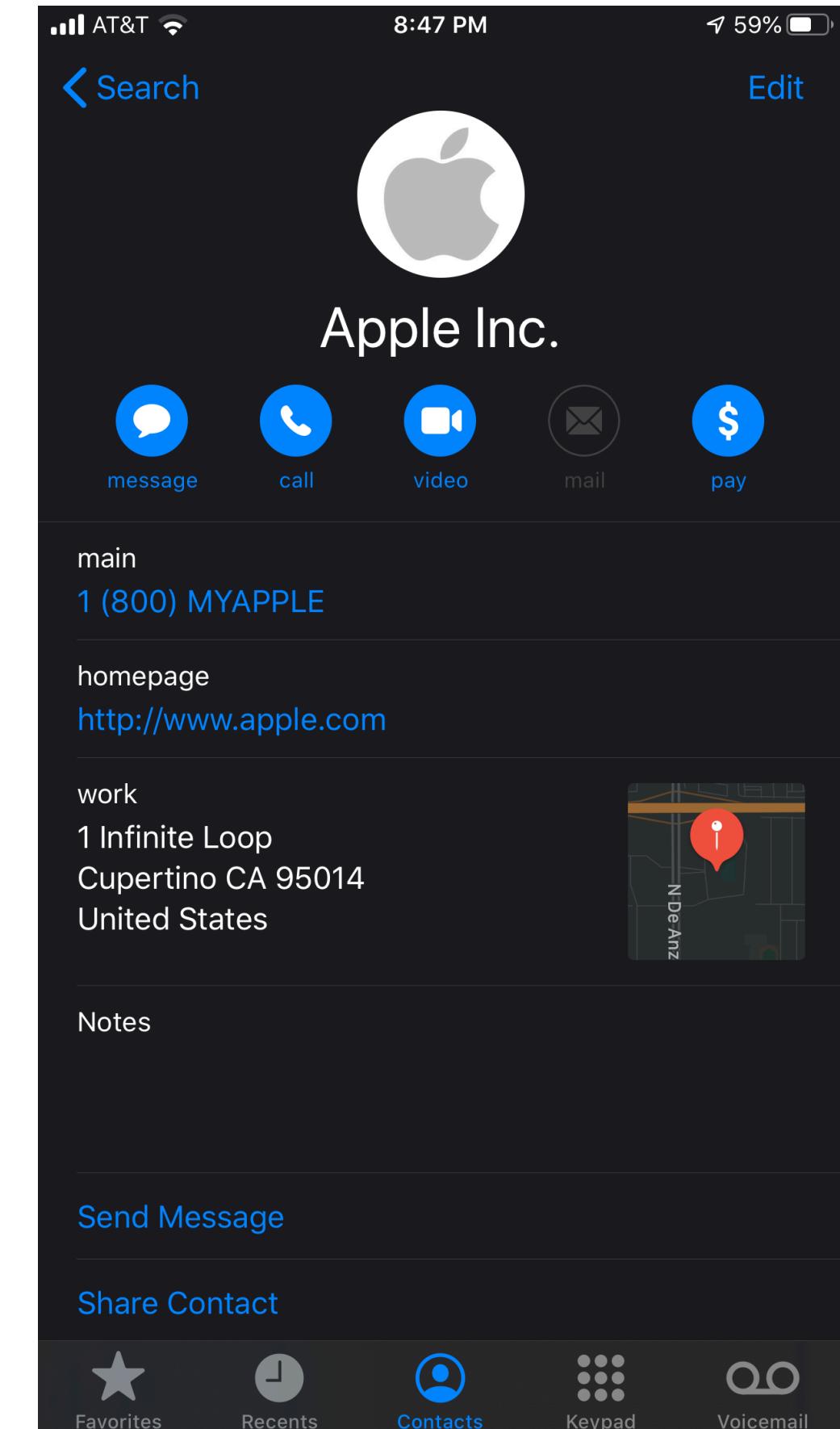
What do we need to perform a cognitive walkthrough?

- A prototype
- A user profile
- Set of tasks
- Sequences of actions

Question 1²⁵

Will the user try and achieve the right outcome?

Does the design support the user's mental model, knowledge, and prior experience to achieve the outcome?

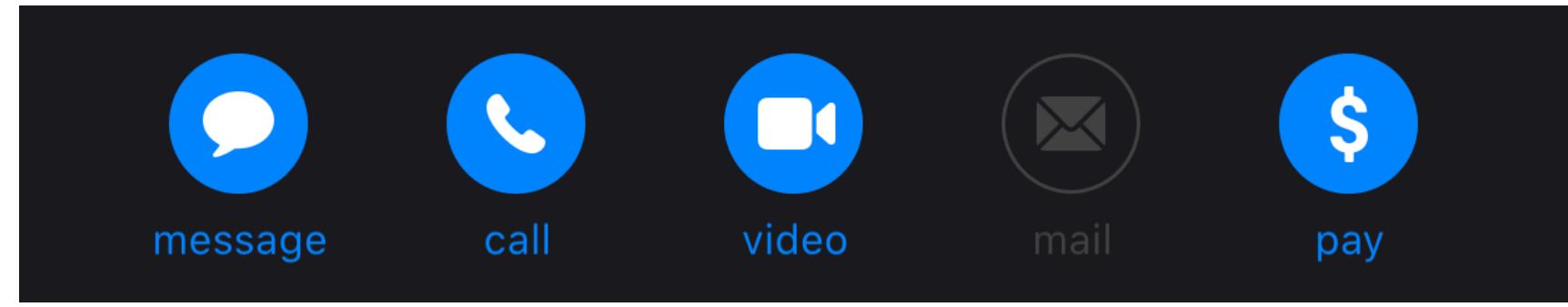


²⁵ [Image source](#)

Question 2

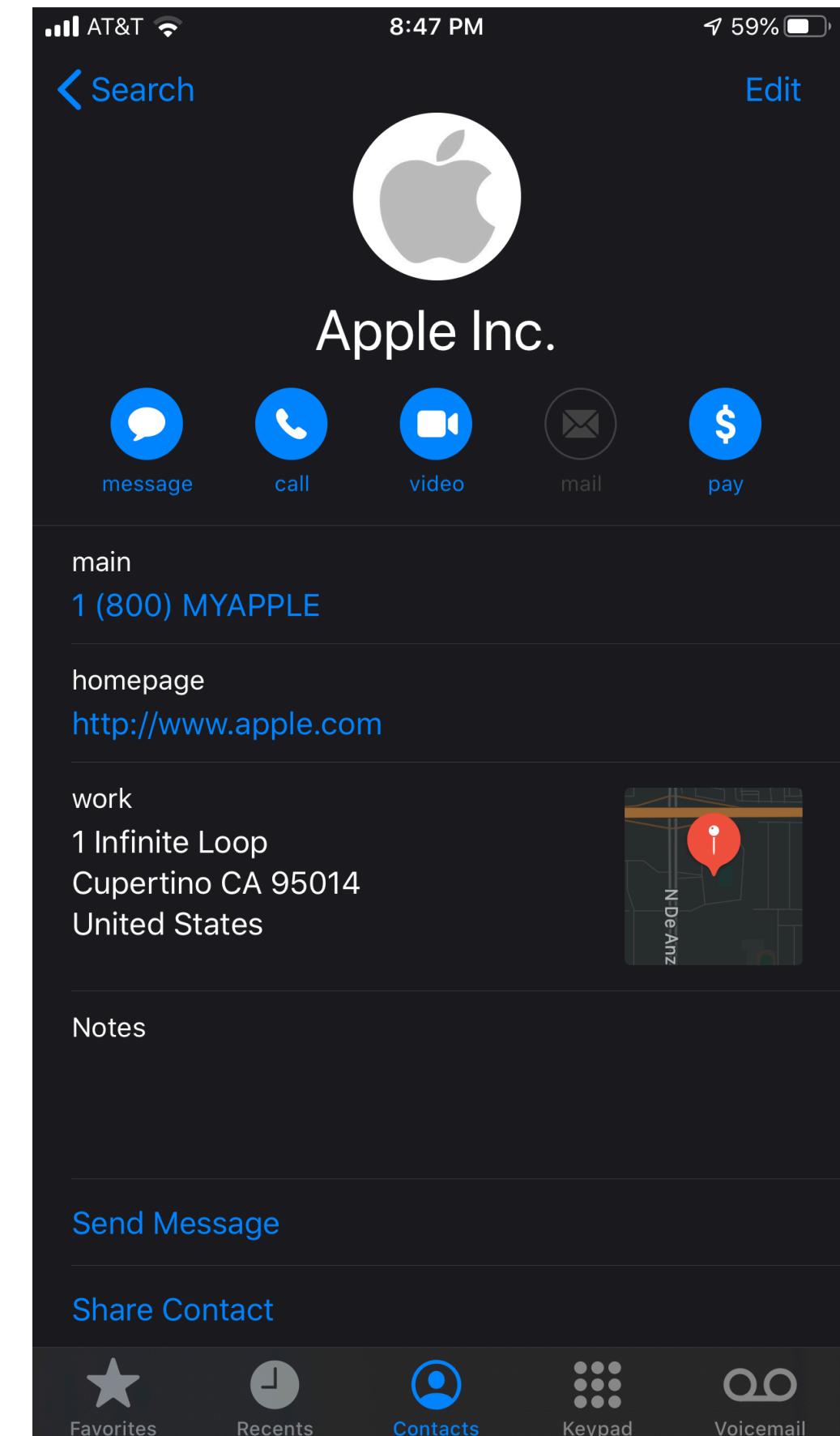
Will the user notice that the correct action is available to them?

Does the visibility, availability, and accessibility of the design element support the action?



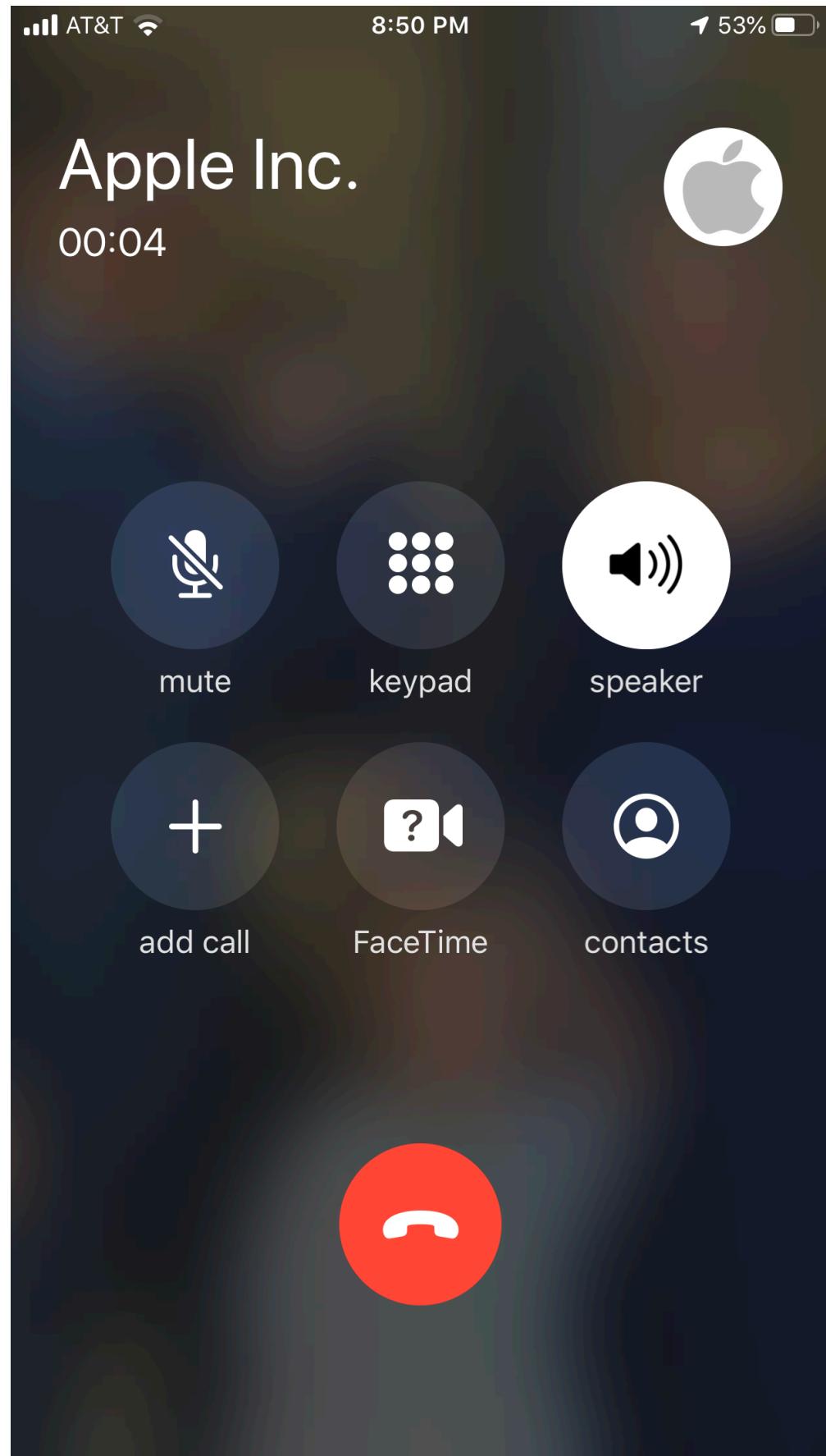
Question 3

Will the user associate the correct action with the outcome they expect to achieve?



Question 4

If the correct action is performed; will the user see that progress is being made towards their intended outcome?



Pros & Cons of Cognitive Walkthrough

Pros:

- Powerful for:
 - Walk-up-and-use interfaces
 - New concepts/forms of interaction
 - Systems designed for various user profiles
- Can be performed frequently and at any stage of the design process

Pros & Cons of Cognitive Walkthrough

Cons:

- Focuses only on discoverability/learnability
- Best when used with usability testing

Further Reading on Cognitive Walkthrough

- UX Collective: Assessing usability with Cognitive Walkthrough
- Usability body of knowledge
- Cognitive Walkthroughs
 - Cognitive walkthrough template

What did we learn today?

- What are rapid/expert methods?
- Heuristic evaluation
- Cognitive walkthrough