

Building User Interfaces

**Usability Evaluation:
Rapid Methods**

Professor Yuhang Zhao

What we will learn today?

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

What is usability evaluation?

Usability Evaluation

Definition: The assessment of the effectiveness of 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.

We will cover testing-based methods later in the semester.

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.

Our focus today will be on expert-review-based methods.

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**



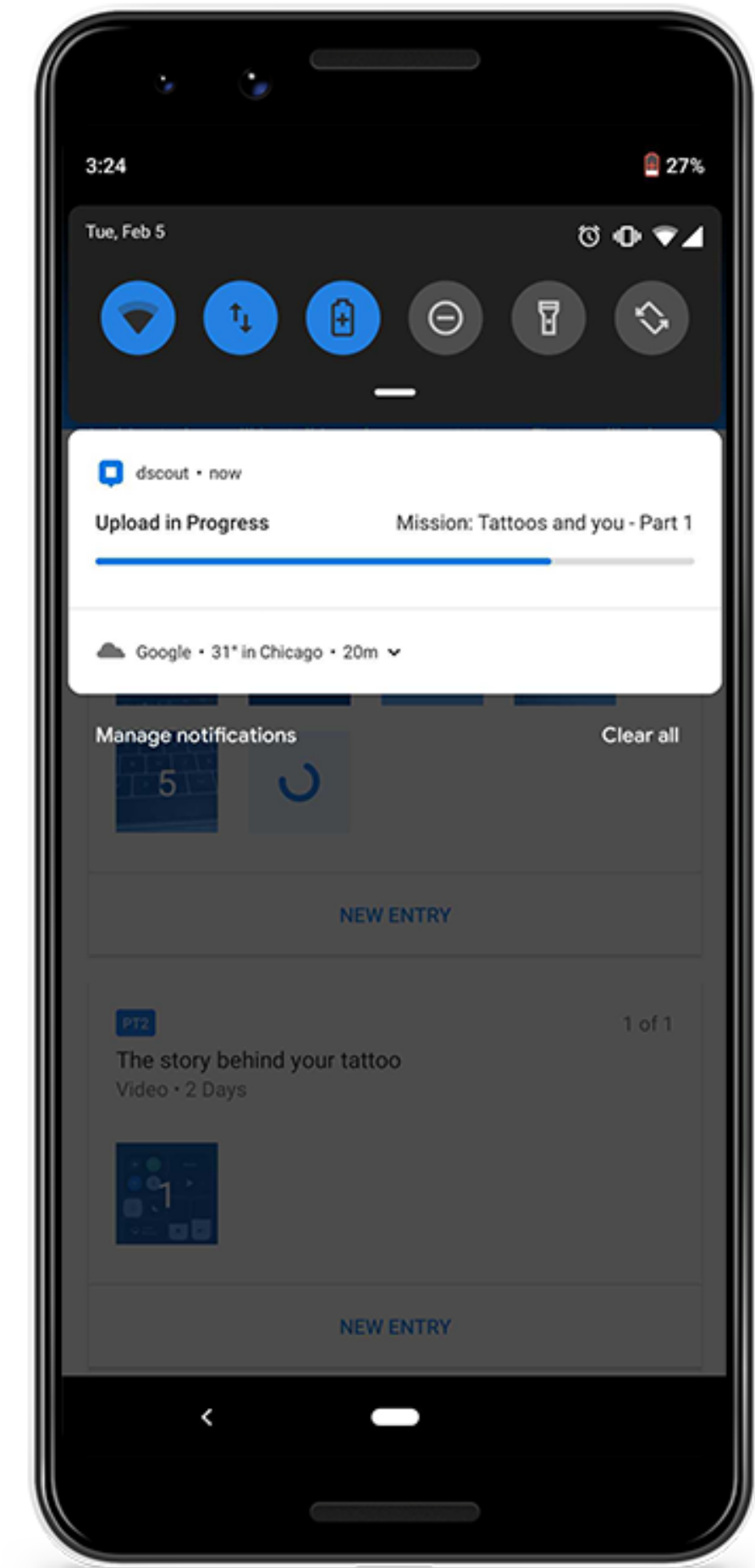
1: Visibility of system status^{3 4}

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

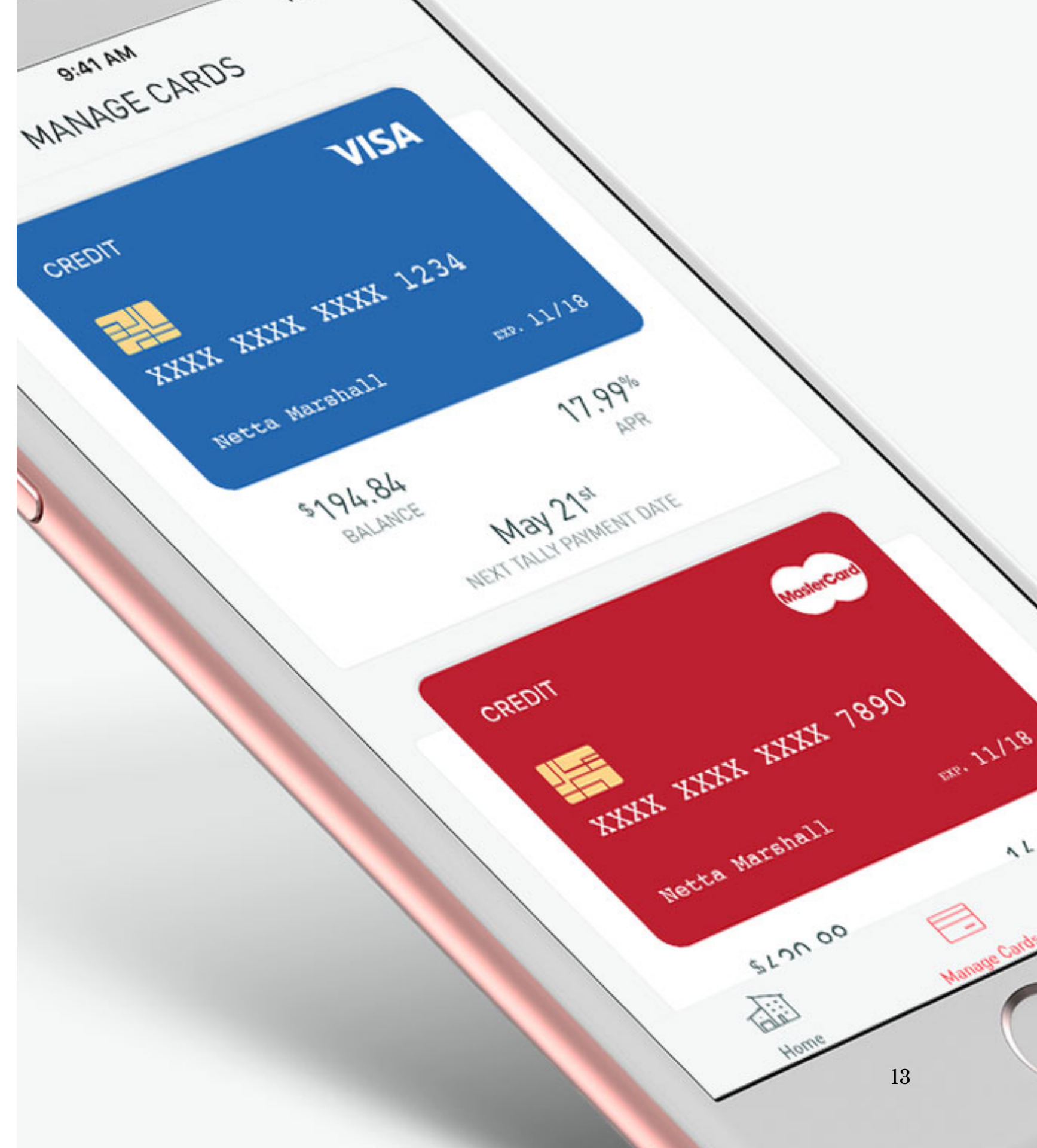


2: Match between system and the real world^{5 6}

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



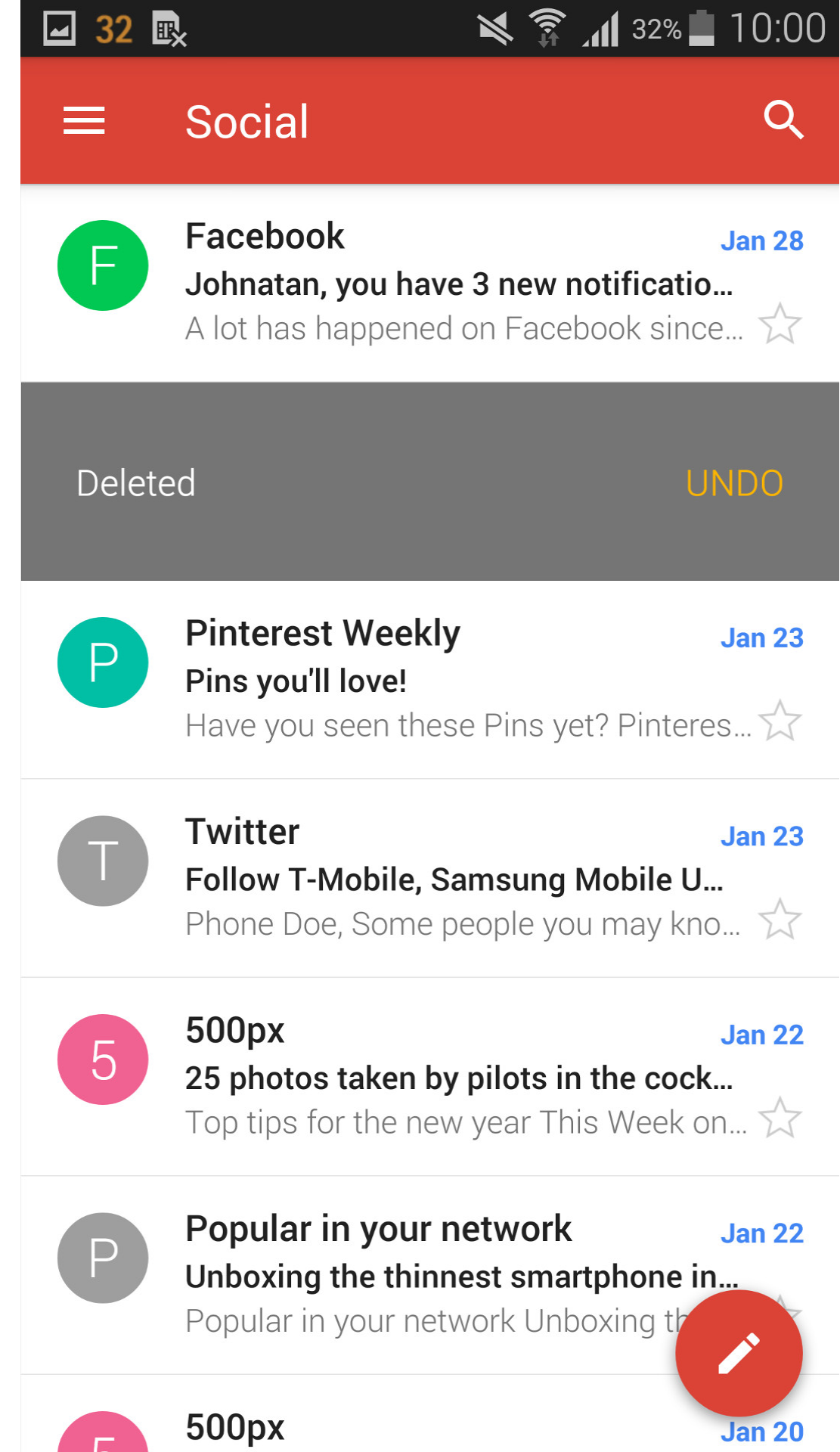
3: User control and freedom^{7 8}

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



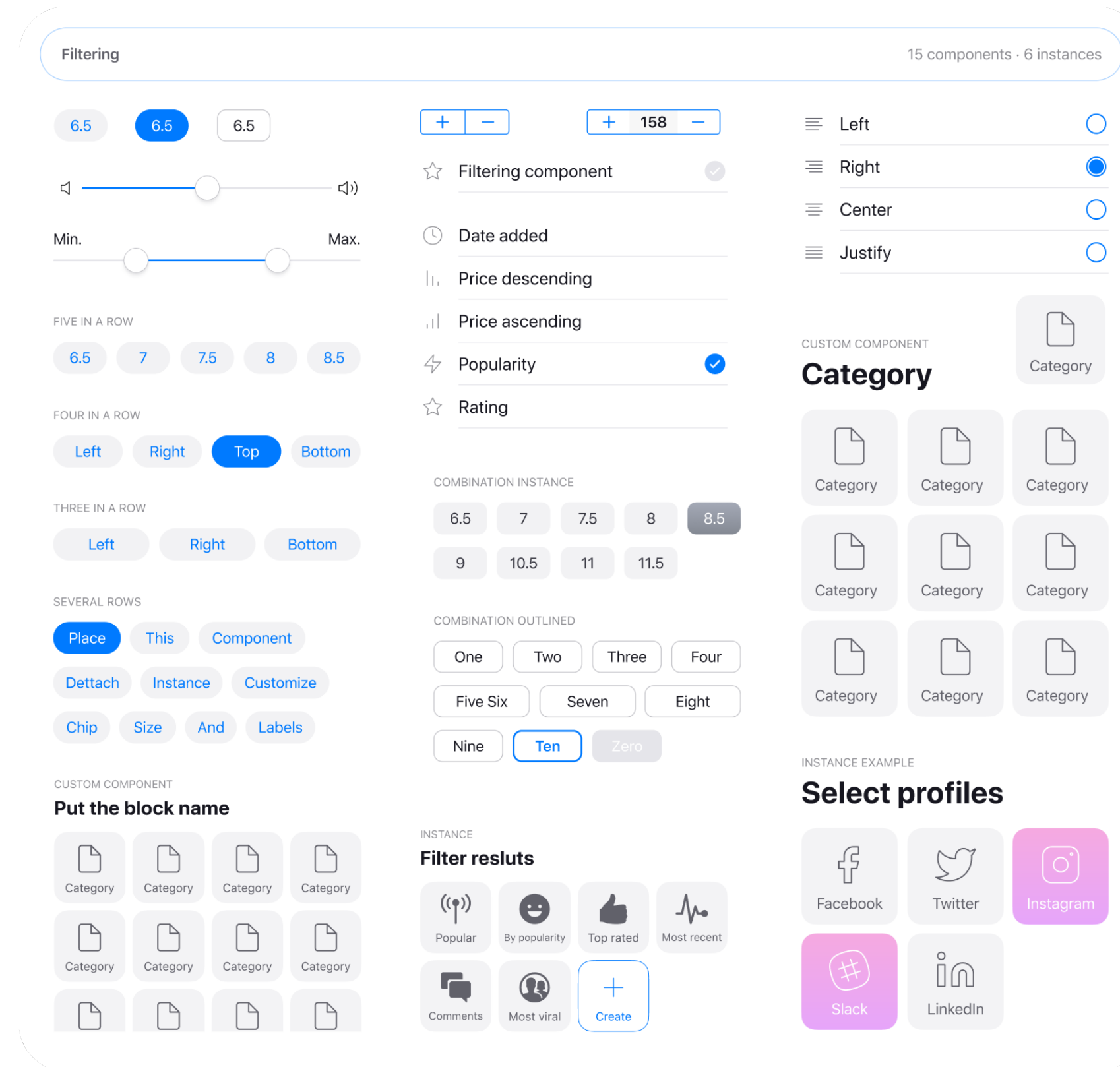
4: Consistency and standards^{9 10}

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.

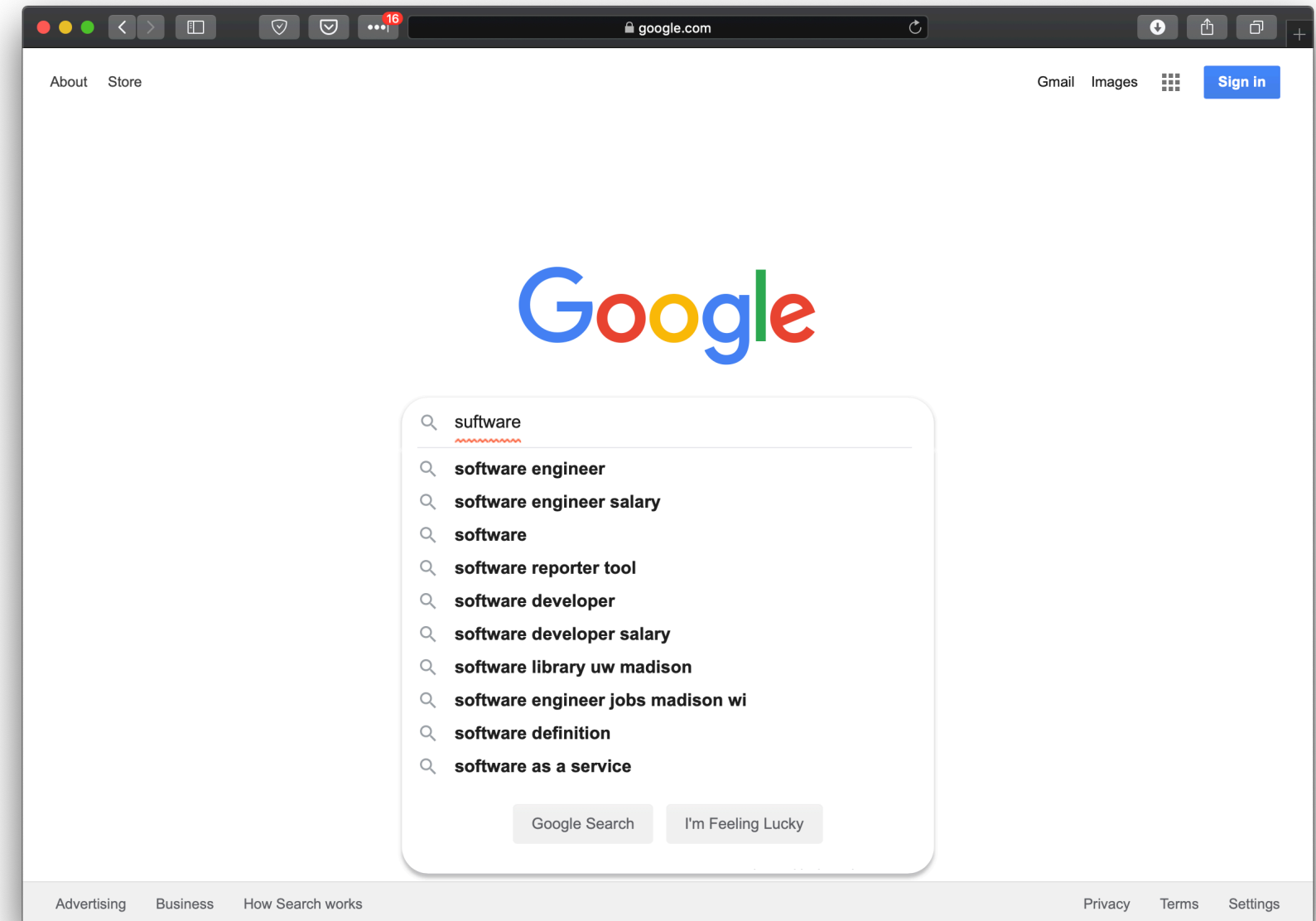
⁹ NN/g: Consistency and standards

¹⁰ Image source



5: Error prevention¹¹

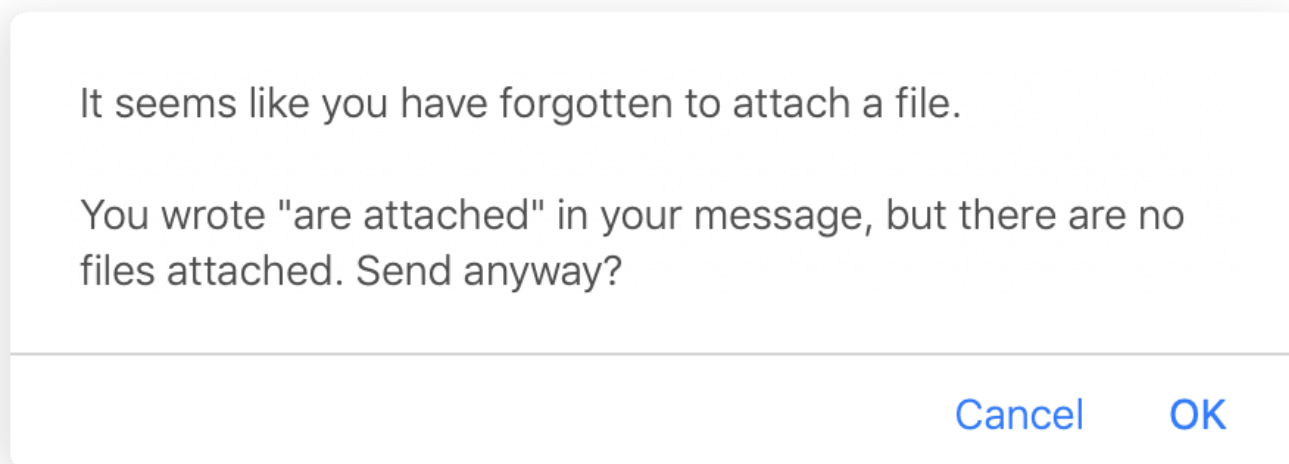
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.



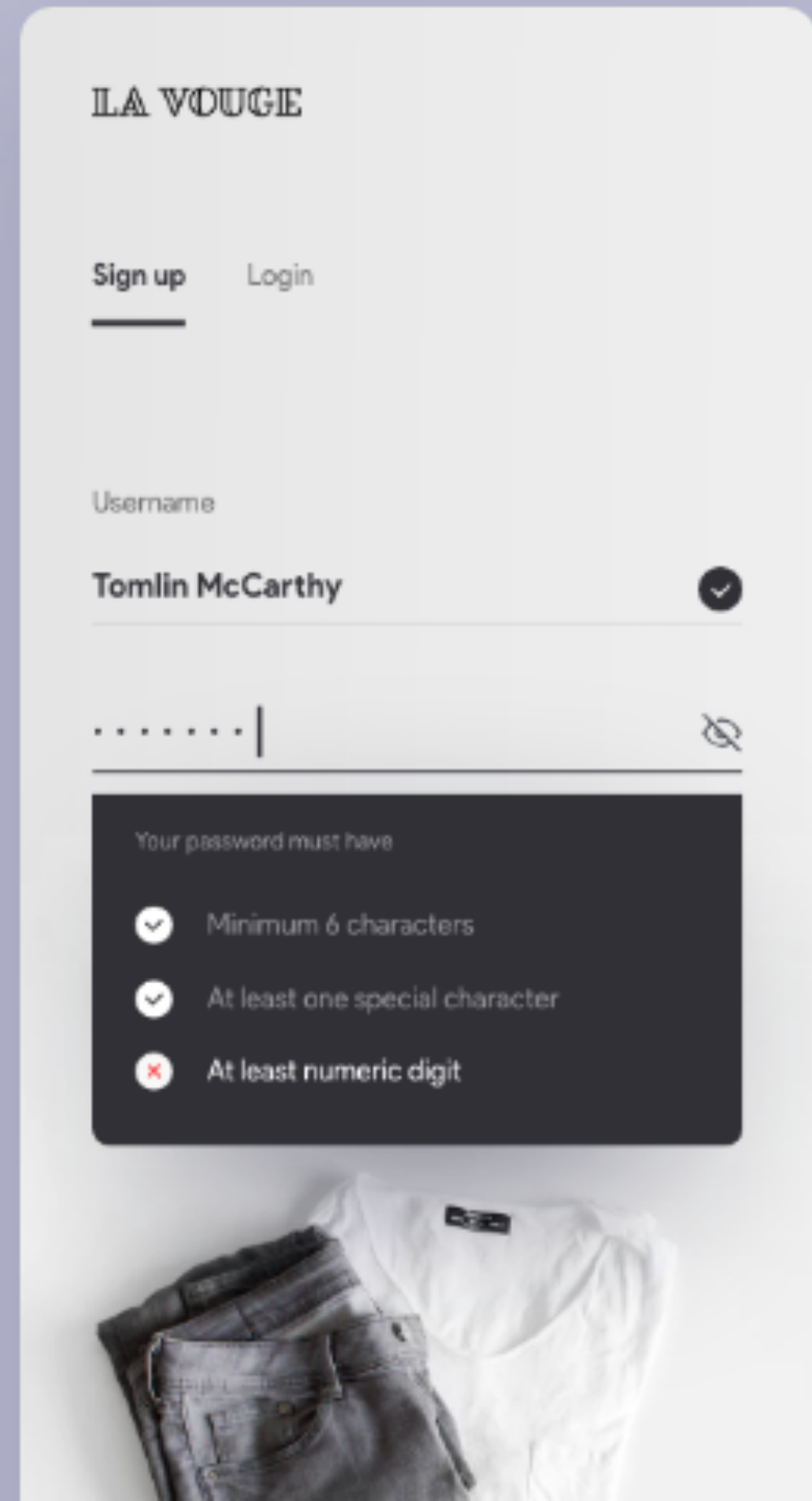
¹¹ NN/g: Error prevention

Examples:¹²

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

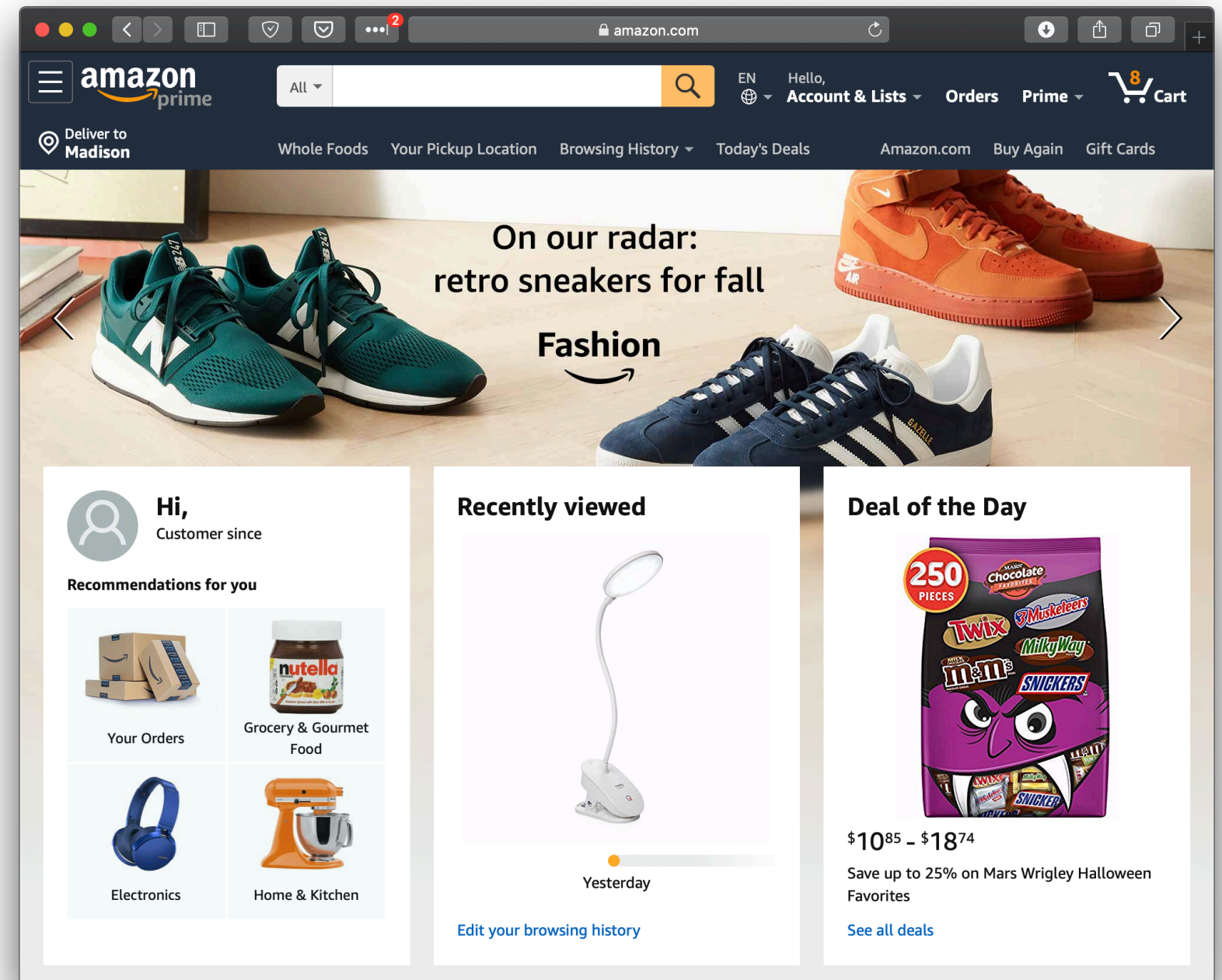


¹² Image source: [Left](#), [Right](#)

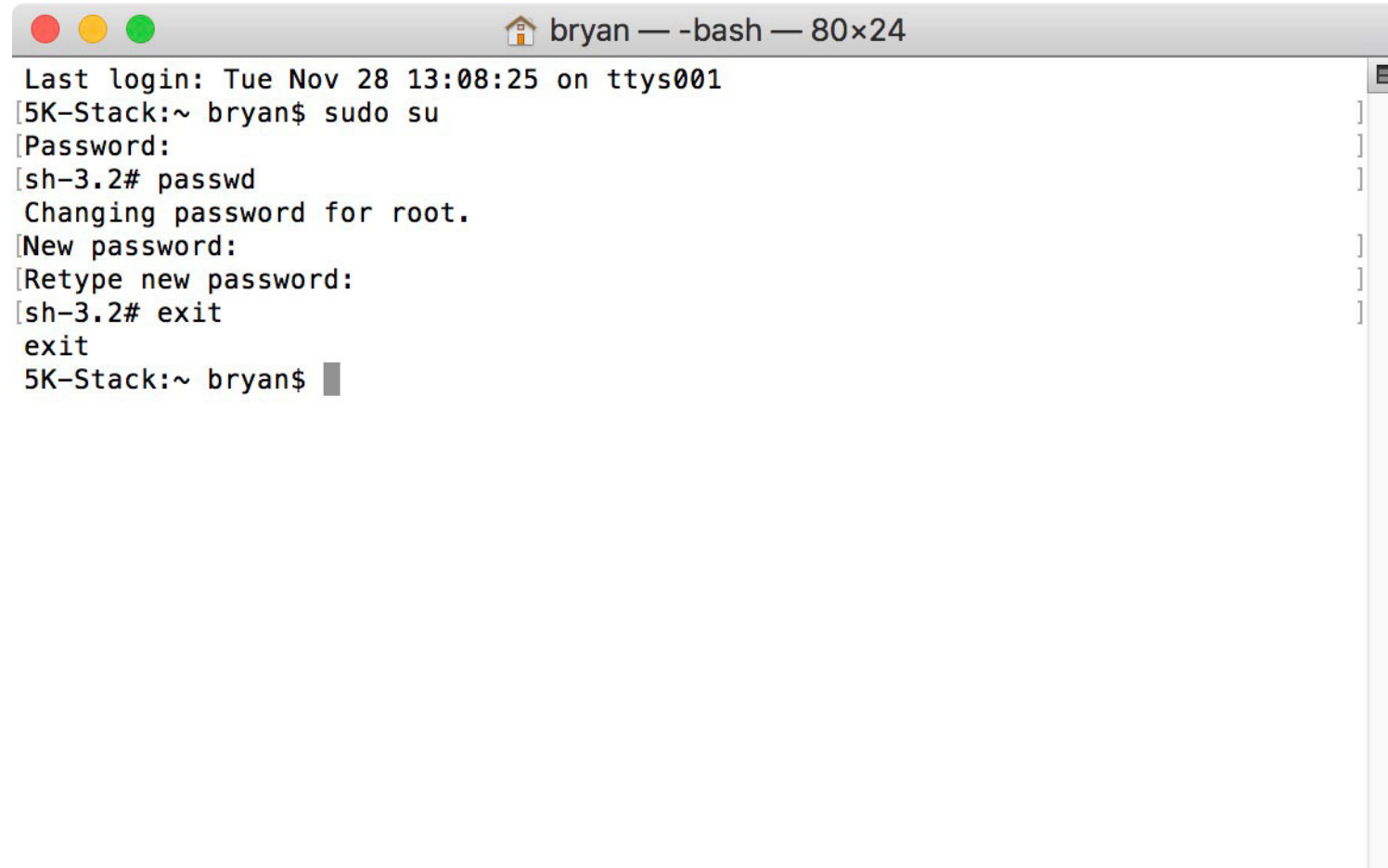


6: Recognition rather than recall¹³

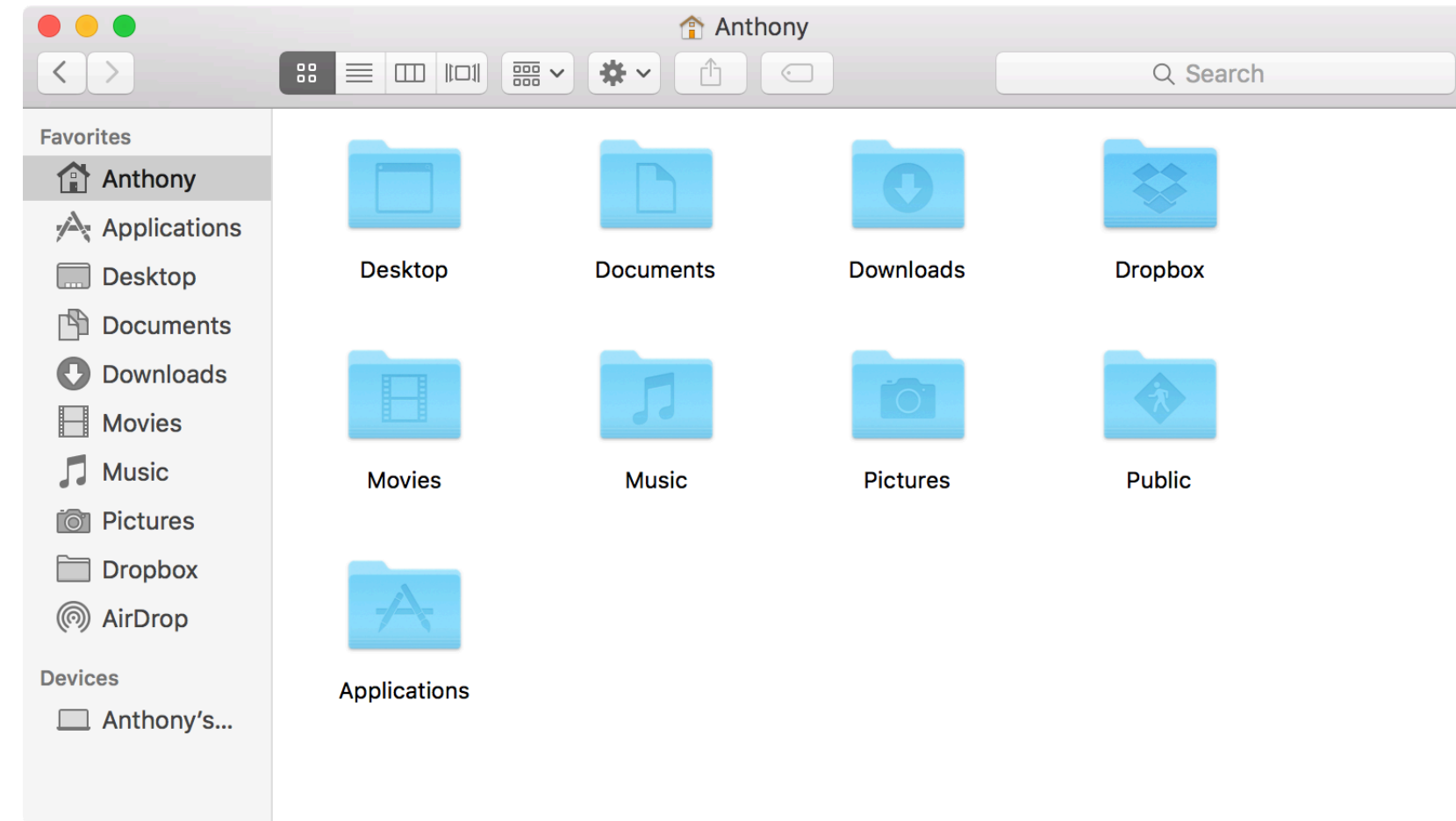
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.



¹³ NN/g: Recognition rather than recall



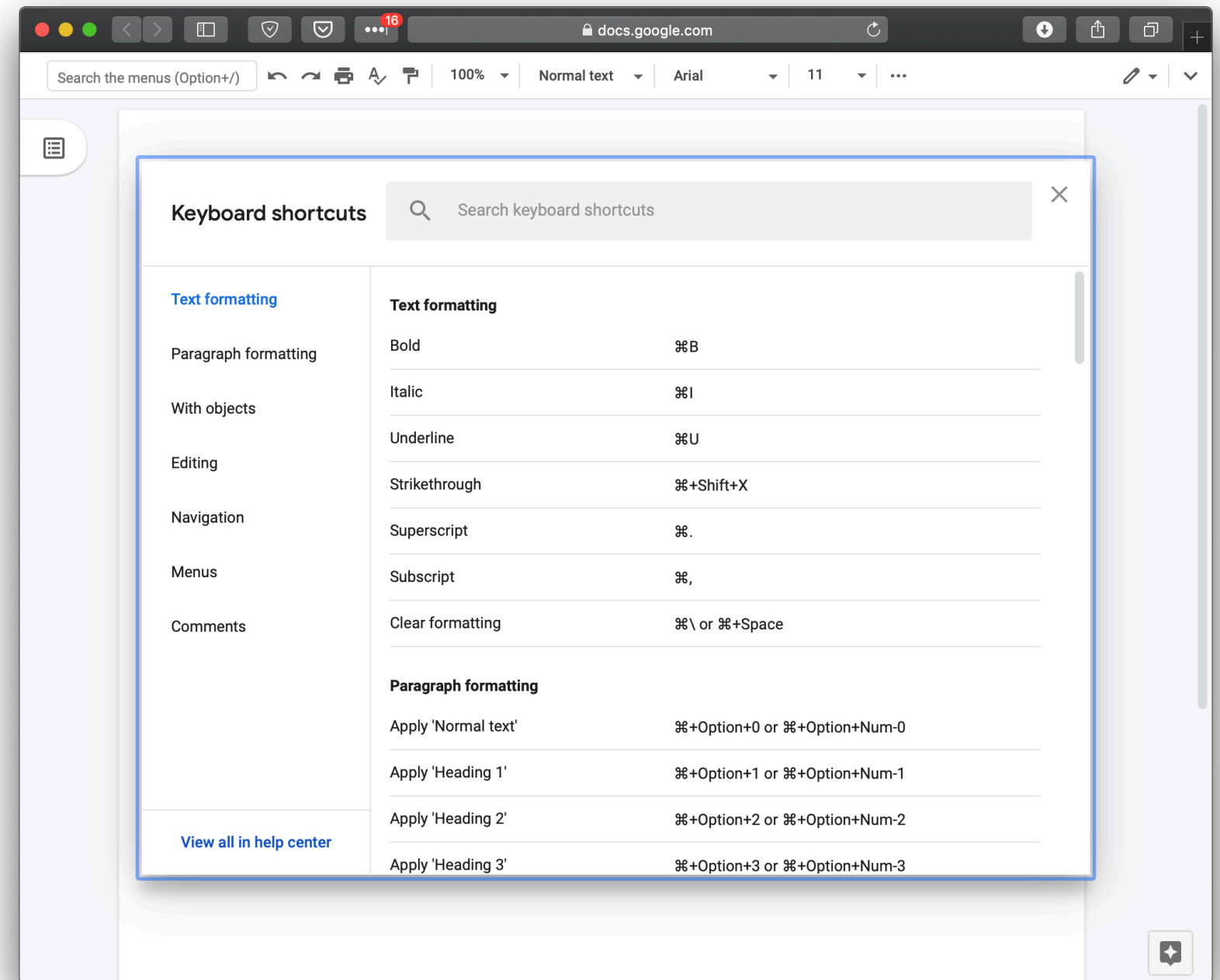
```
bryan — -bash — 80x24
Last login: Tue Nov 28 13:08:25 on ttys001
[5K-Stack:~ bryan$ sudo su
>Password:
[sh-3.2# passwd
Changing password for root.
>New password:
>Retype new password:
[sh-3.2# exit
exit
5K-Stack:~ bryan$
```



¹⁴ Image source: Left, Right

7: Flexibility and efficiency of use¹⁵

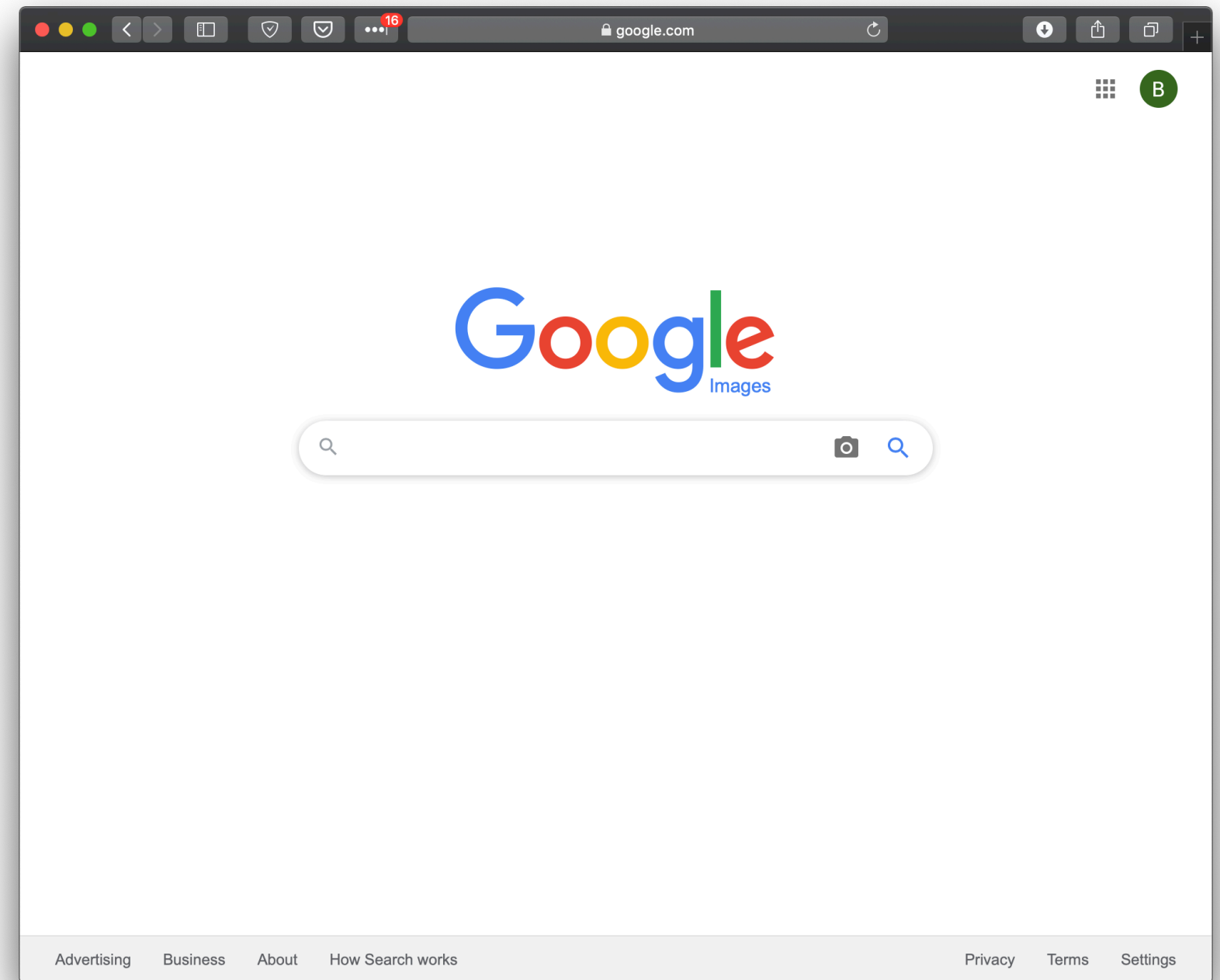
Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.



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.



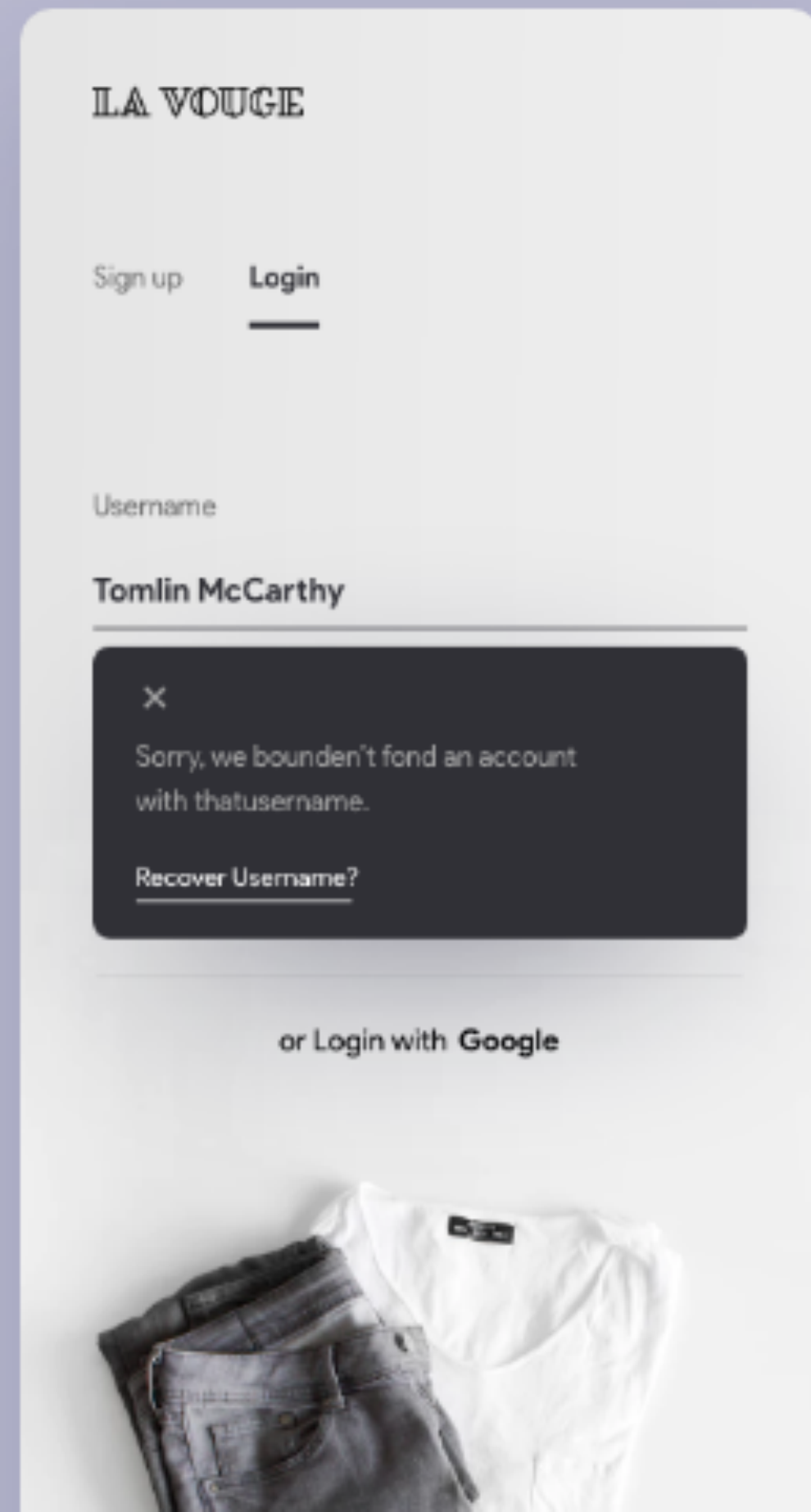
¹⁶ NN/g: Aesthetic and minimalist design

9: Help users recognize, diagnose, and recover from errors^{17 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

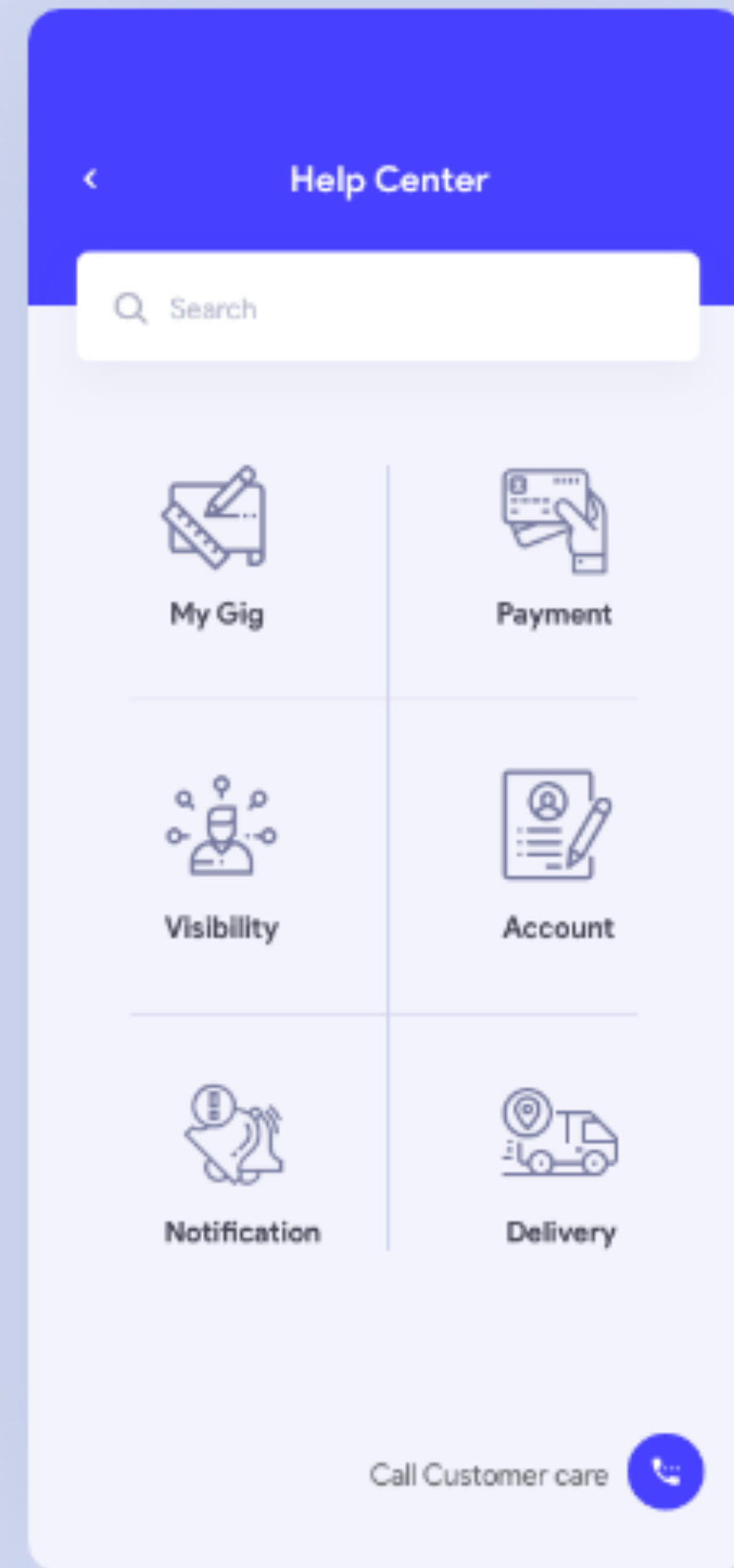


10: Help and documentation^{19 20}

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.

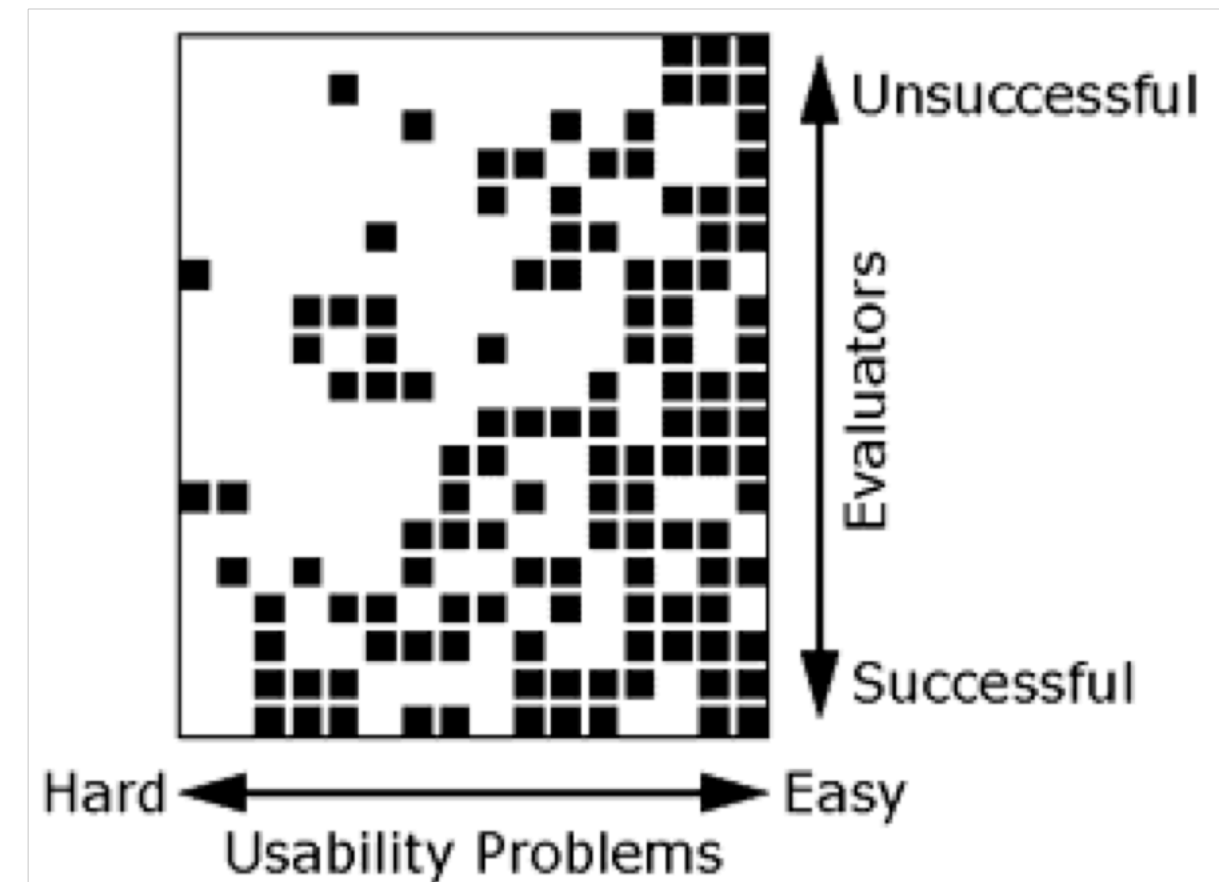
¹⁹ NN/g: [Help and documentation](#)

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.



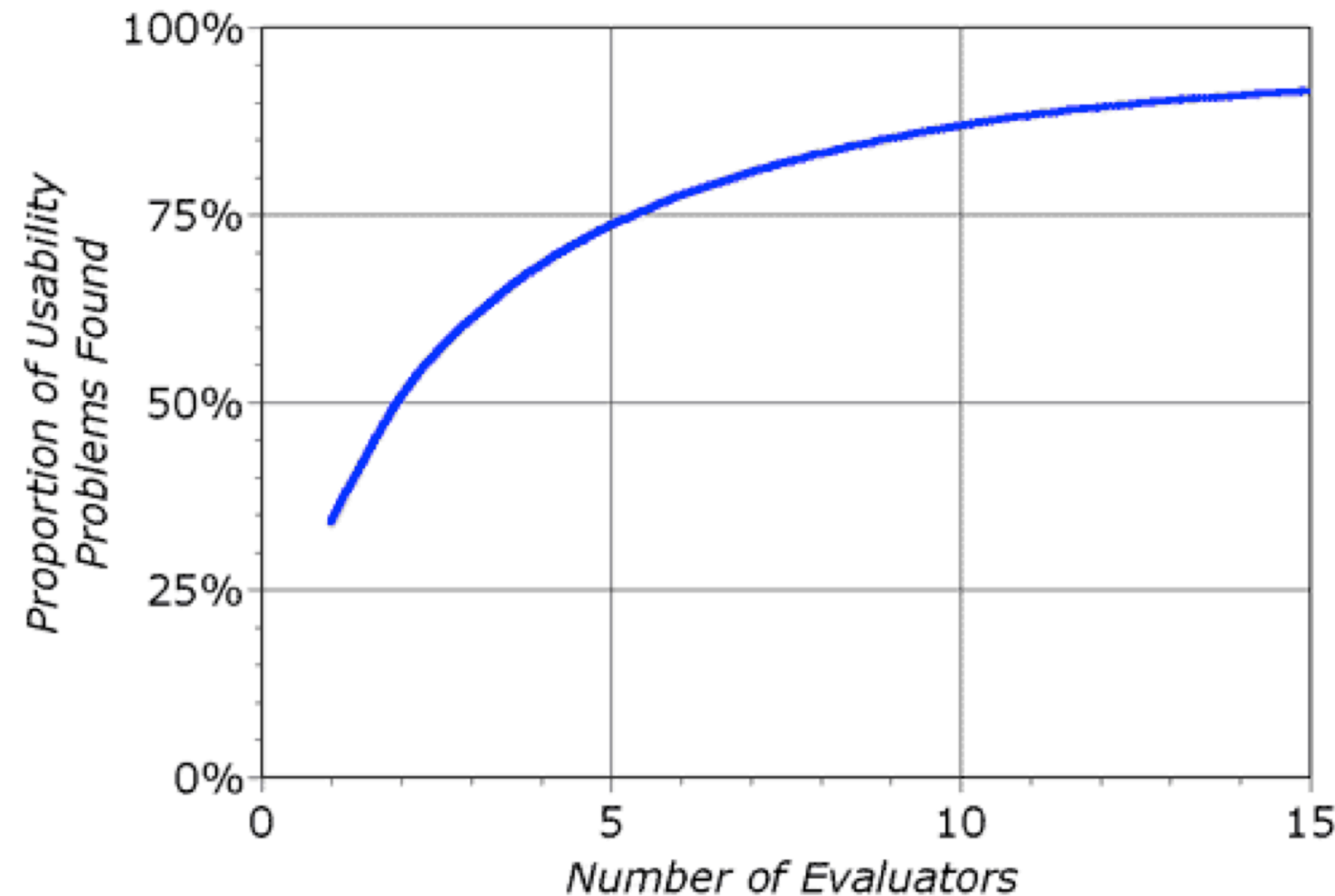
²¹ NN/g: [How to Conduct a Heuristic Evaluation](#)

How many evaluators are needed?

The rule of thumb is 3–5.²²

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

- 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

Evaluator

Browser/OS

Date

Task/Feature

Website/App

0 I don't agree that this is a usability problem at all

1 Cosmetic problem only: need not be fixed unless extra time is available on project

2 Minor usability problem: fixing this should be given low priority

3 Major usability problem: important to fix, so should be given high priority

4 Usability catastrophe: imperative to fix this before product can be released

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

Severity

0 1 2 3 4

Issues Please be specified

Recommendation

2. Match between system and the real world
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

0 1 2 3 4

Issues Please be specified

Recommendation

Credit by Hsin-Jou Lin

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

In-Class Activity

Heuristic Evaluation of **MyFitnessPal**

Design Elements²⁴

- *Add* component
- *Food Dairy* component
- *Nutrition* component

²⁴ UX Collective

10 Usability Heuristics



Visibility

Show system status, tell what's happening



Mapping

Use familiar metaphors & language



Freedom

Provide good defaults & undo



Consistency

Use same interface and language throughout



Error Prevention

Help users avoid making mistakes



Recognition

Make information easy to discover



Flexibility

Make advanced tasks fluid and efficient



Minimalism

Provide only necessary information in an elegant way



Error Recovery

Help users recognize, diagnose and recover from errors



Help

Use proactive and in-place hints to guide users

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?

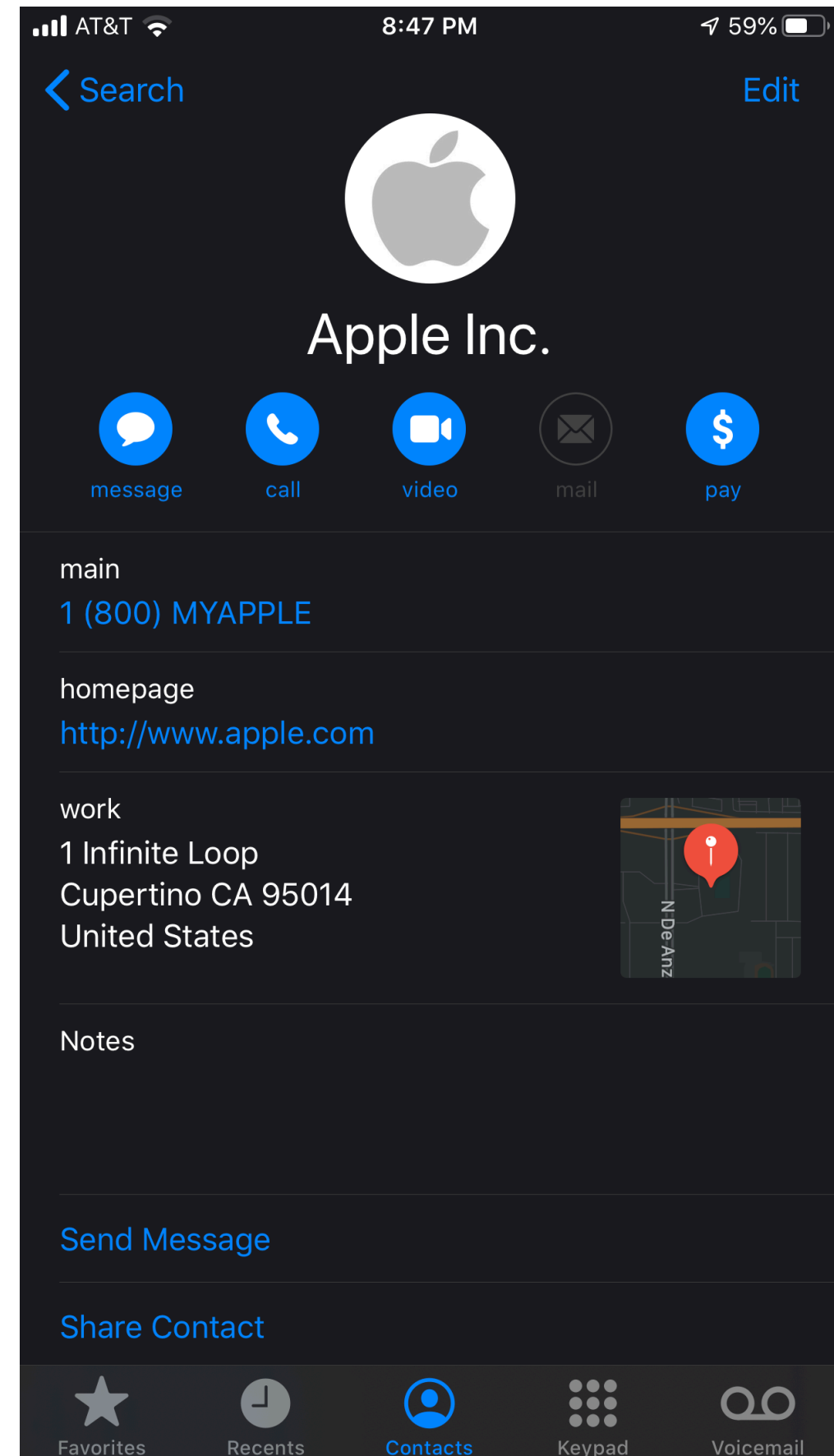
- A prototype
- A user profile
- Set of tasks
- Sequences of actions

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?

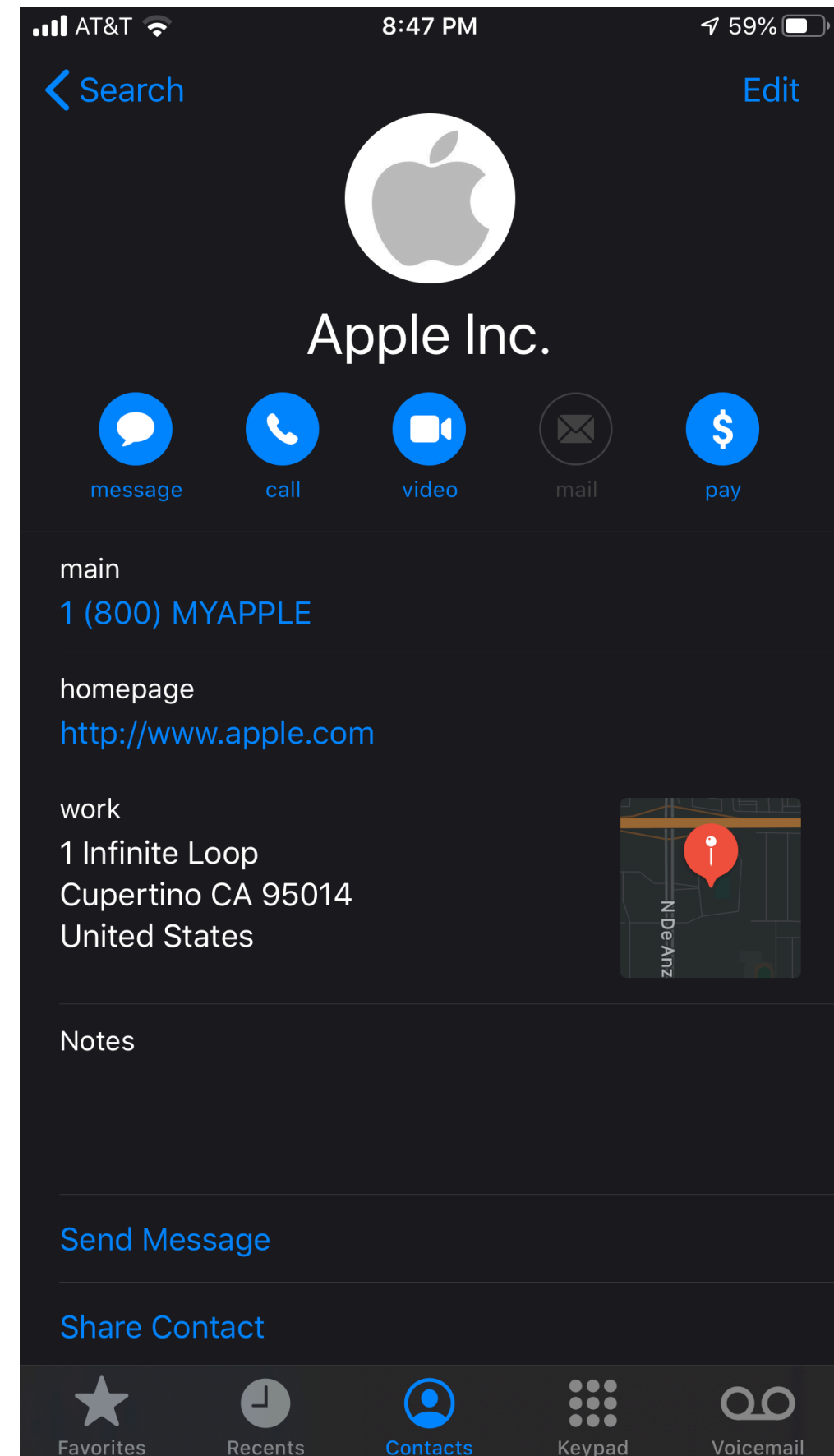
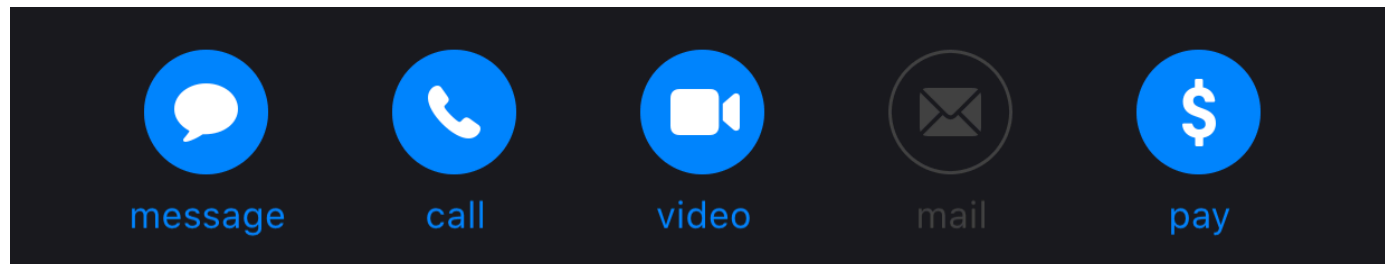
²⁵ Image source



Question 2

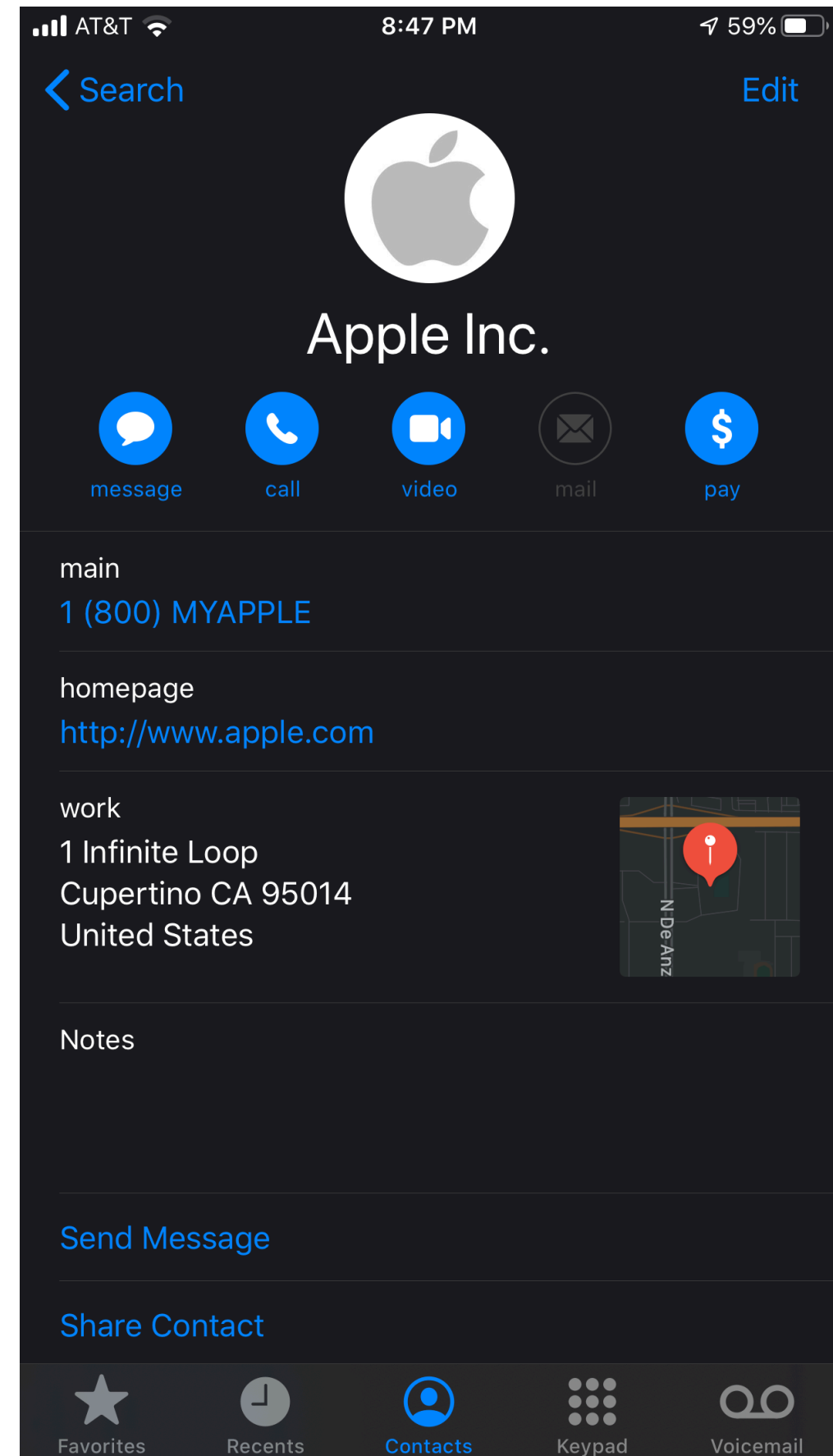
Will the user notice that the correct action is available to them?

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



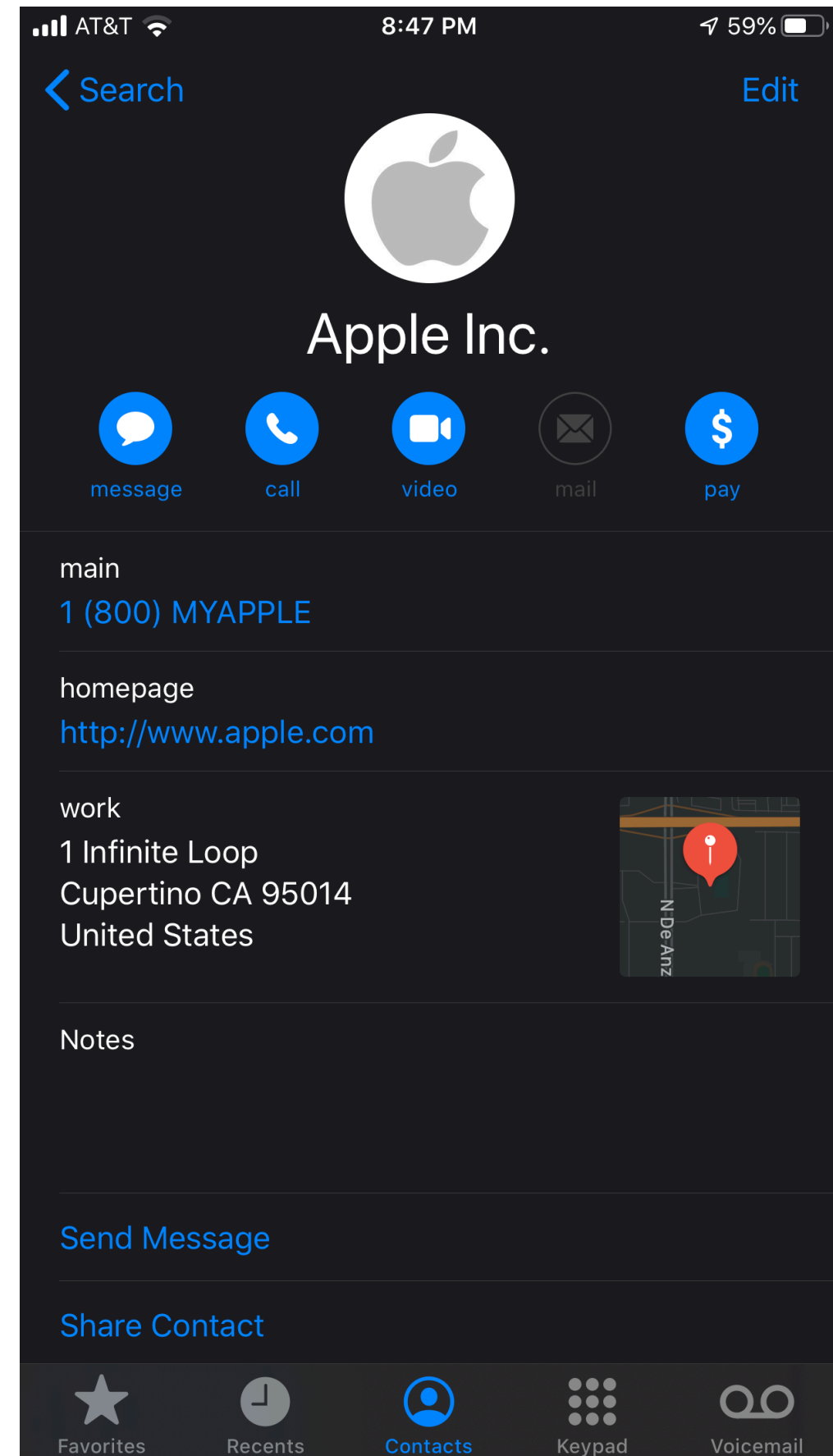
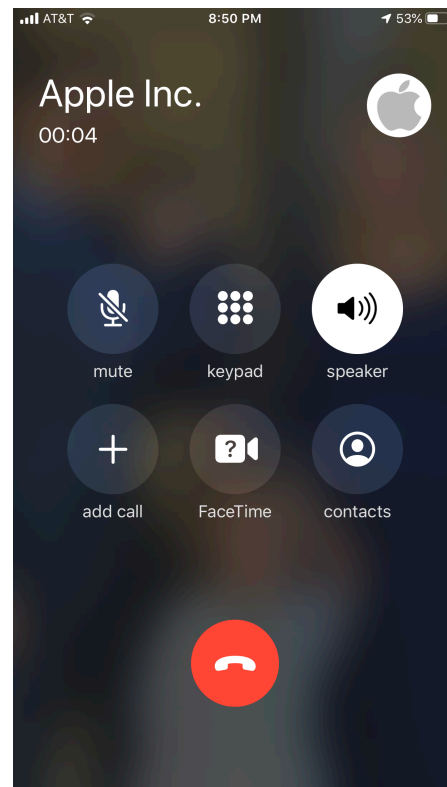
Question 3

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

Const:

- 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

In-Class Activity

Cognitive Walkthrough of **MyFitnessPal**

Tasks

- Add food item
- Review food item nutrients
- Review yesterday's record

Questions

1. Will the user try and achieve the right outcome?
2. Will the user notice that the correct action is available...?
3. Will the user associate the correct action with the outcome they expect...?
4. If the correct action is performed; will the user see that progress is being made towards their intended outcome?

What did we learn today?

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