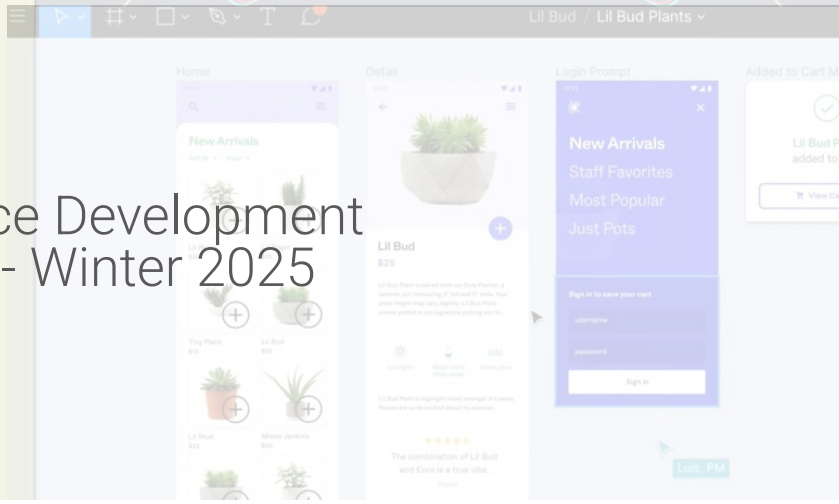
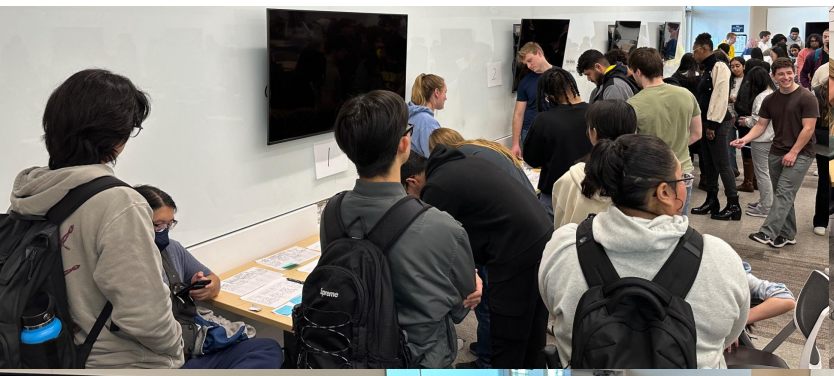
Heuristic Evaluation

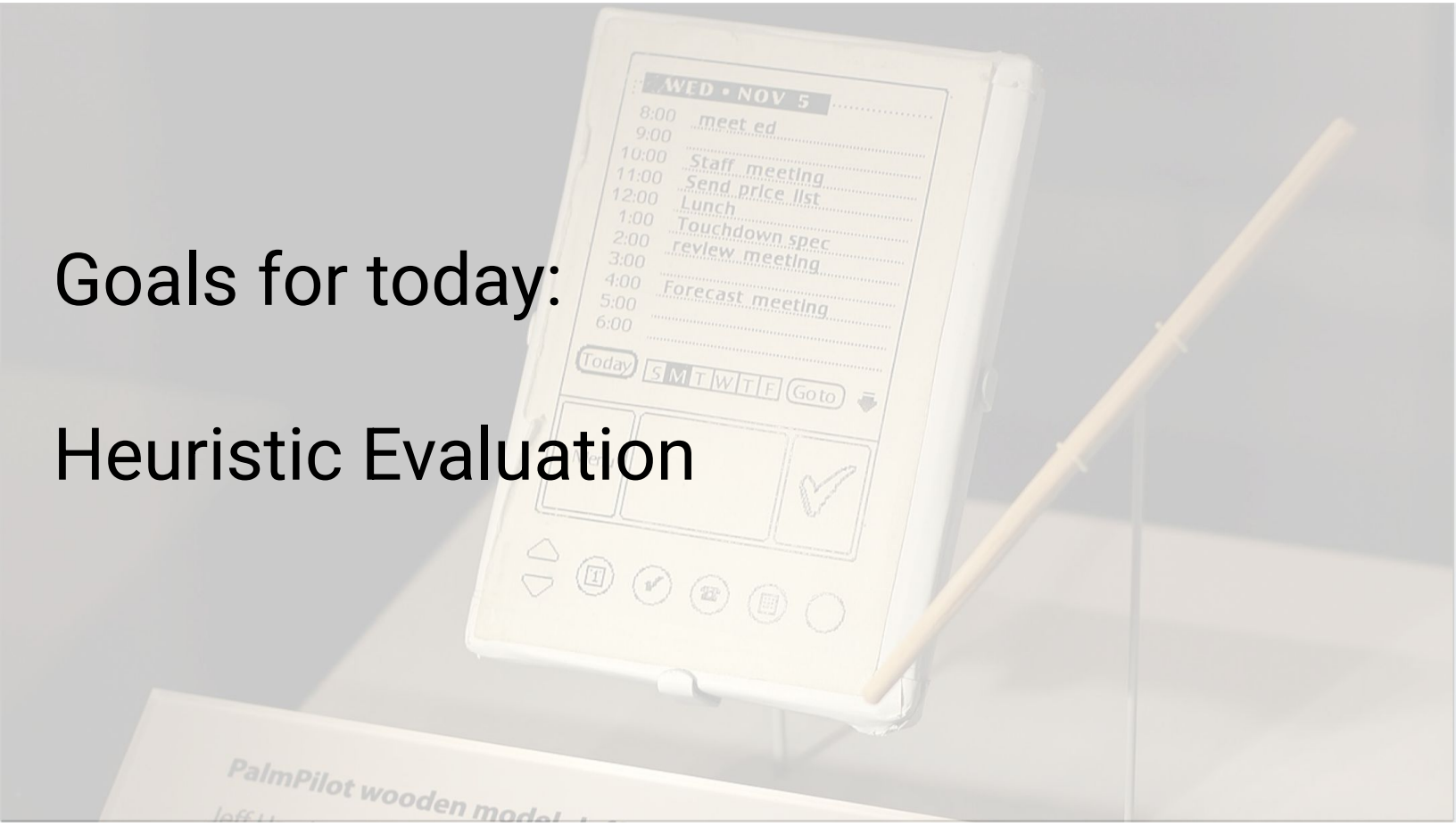
User Interface Development
EECS 493 - Winter 2025



Current items

1. Storyboard Bake-off next Thursday – required attendance
   a. Sign up for a slot!!
   b. Spreadsheet for signing up is in the spec





Heuristic Evaluation

A Method for you to evaluate an interface or a prototype before talking to real users.

4 Phases of Human-Centered Design

- **Need finding** → Conduct research about customers who will use the product being developed/improved
- **Ideation**→ Generate potential solutions
- **Prototyping** → Build a mockup of each potential solution
- **Testing** → Have sample of individuals from target population use prototype in order to test design suitability

Usability Inspection Techniques

These are methods you can do by yourself or in your team (team is better)

- Cognitive walk-throughs
- Heuristic evaluations

These are “discount usability” methods

These are *not user testing*

We’ll continue with user testing after the break

Heuristic Evaluation

- For each screen/page/state
  - Inspect various dialog elements
  - Compare with list of usability principles
- Usability principles
  - Nielsen's "heuristics"
  - Apple UI Guidelines
  - Universal design principles
  - Etc

Heuristic Evaluation vs. User Testing

- Heuristic Evaluation is much faster than user testing
  - 1-2 hours each evaluator vs. days- weeks
- Heuristic Evaluation does not require interpreting users' actions
- User testing is more accurate and realistic (by definition)
  - Takes into account actual users and tasks
  - Heuristic Evaluation may miss problems
- Good to alternate between Heuristic Evaluation and user testing
  - Find different problems
  - Don't waste participants

Heuristic Evaluation vs. User Testing

- Heuristic Evaluation is much faster than user testing
  - 1-2 hours each evaluator vs. days- weeks
- Heuristic Evaluation does not require interpreting users' actions
- User testing is more accurate and realistic (by definition)
  - Takes into account actual users and tasks
  - Heuristic Evaluation may miss problems
- Good to alternate between Heuristic Evaluation and user testing
  - Find different problems
  - **Don't waste participants**

Nielsen's Usability Heuristics

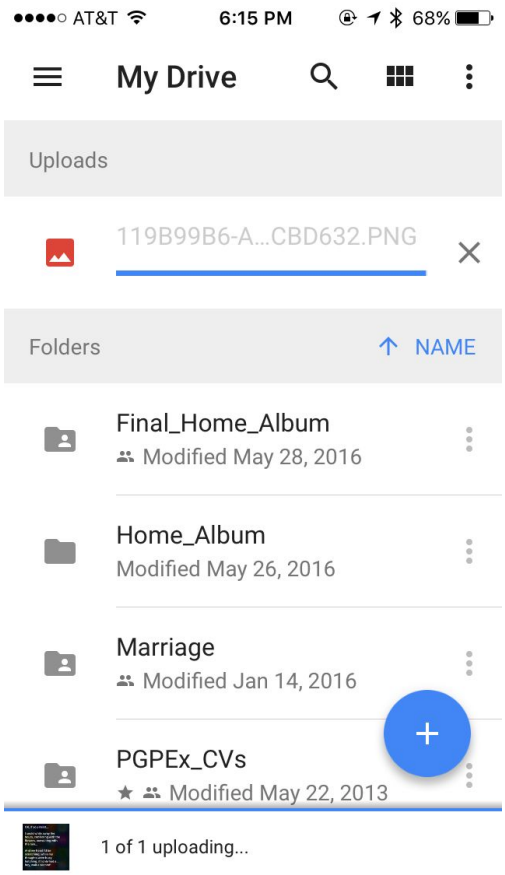
1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover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

#1 Visibility of system status

- The user should know what's going on inside the system.
- We need to give feedback of their action within a reasonable time.
- This feedback is normally associated with points of action and can be provided using a color change, loader, time-left graphics, etc.

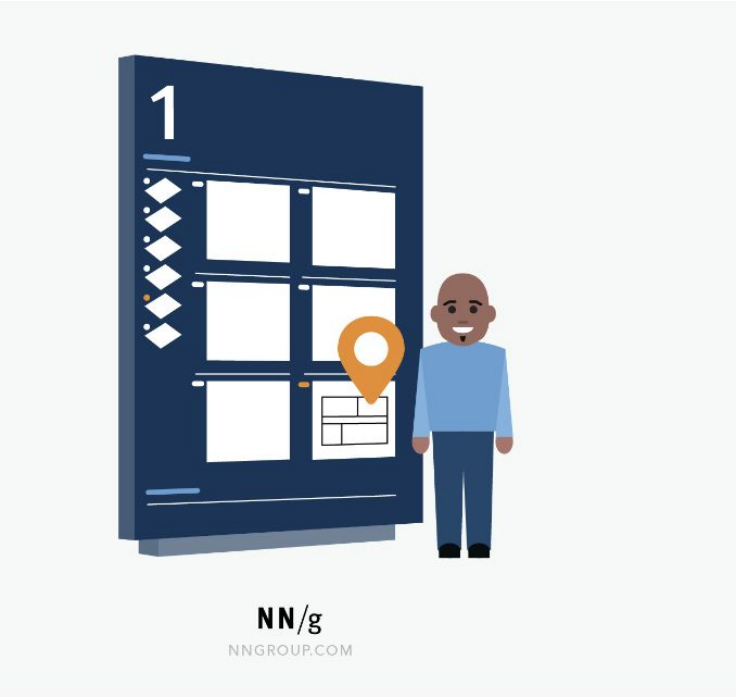
#1 Visibility of system status

Google drive showing the status of a document upload



#1 Visibility of system status

"You Are Here" indicators on mall maps have to show people where they currently are, to help them understand where to go next.

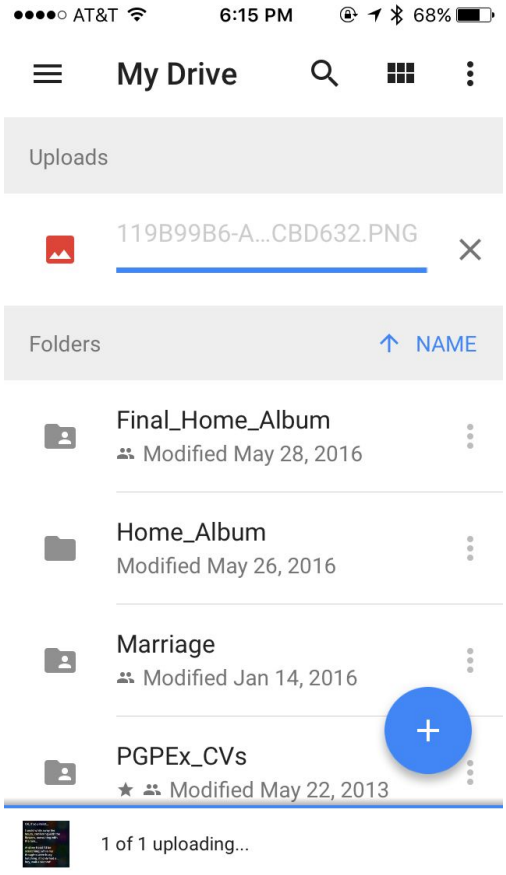


#1 Visibility of system status

Which Gulf?

Gulf of execution

Gulf of evaluation



#2 Match between system and the real world

- The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.
- Is there something on your application that a user may not understand?
- Ensure users can understand meaning without having to go look up a word’s definition.
- User research will help you uncover your users' familiar terminology, as well as their mental models around important concepts.

#2 Match between system and the real world

When stovetop controls match the layout of heating elements, users can quickly understand which control maps to which heating element.

#2 Match between system and the real world

“Yes, I want Neil to teach me how to grow my business”

#3 User control and freedom

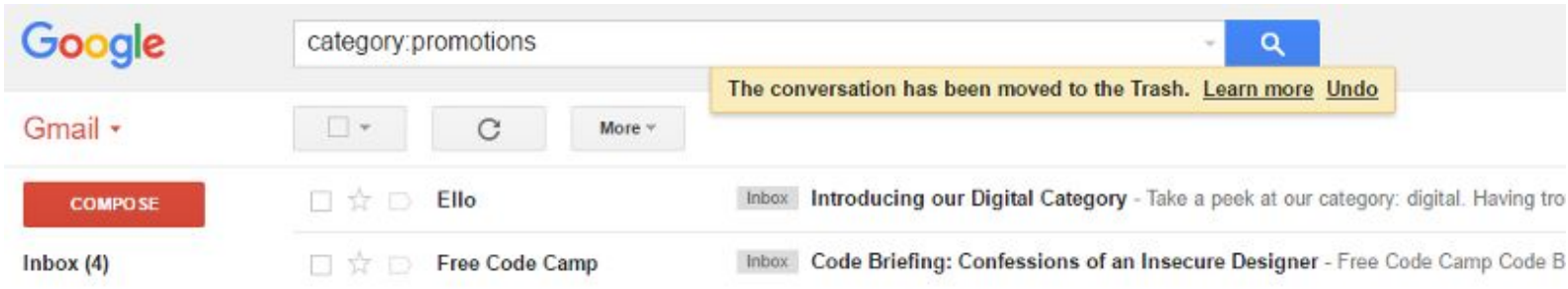
- Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.
- When it's easy for people to back out of a process or undo an action, it fosters a sense of freedom and confidence. Exits allow users to remain in control of the system and avoid getting stuck and feeling frustrated.

#3 User control and freedom

- Support “Undo” and “Redo”
- Let people “exit” form mistaken choices

#3 User control and freedom

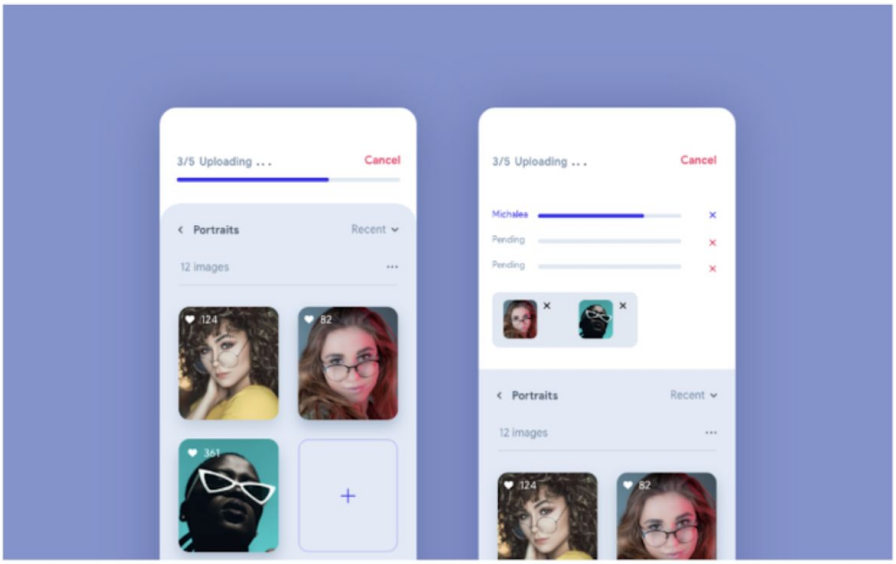
“Undo” button in gmail after sending a message



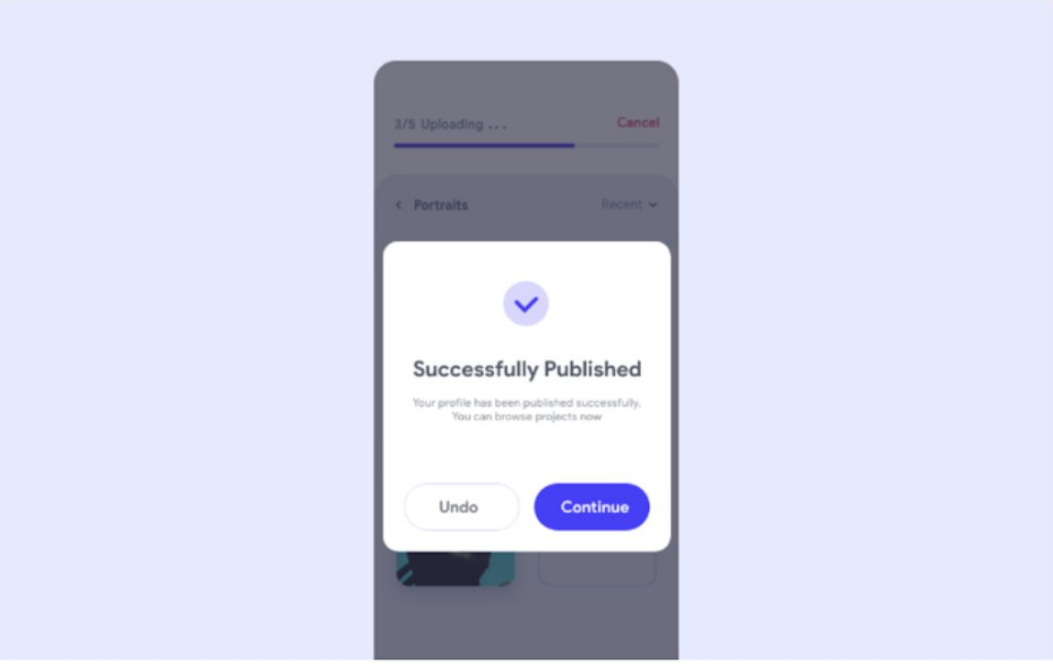
Lecture 13 Survey 1

www.yellkey.com/tell ⇒ https://forms.gle/KRokgPdi4TPLuKzm9

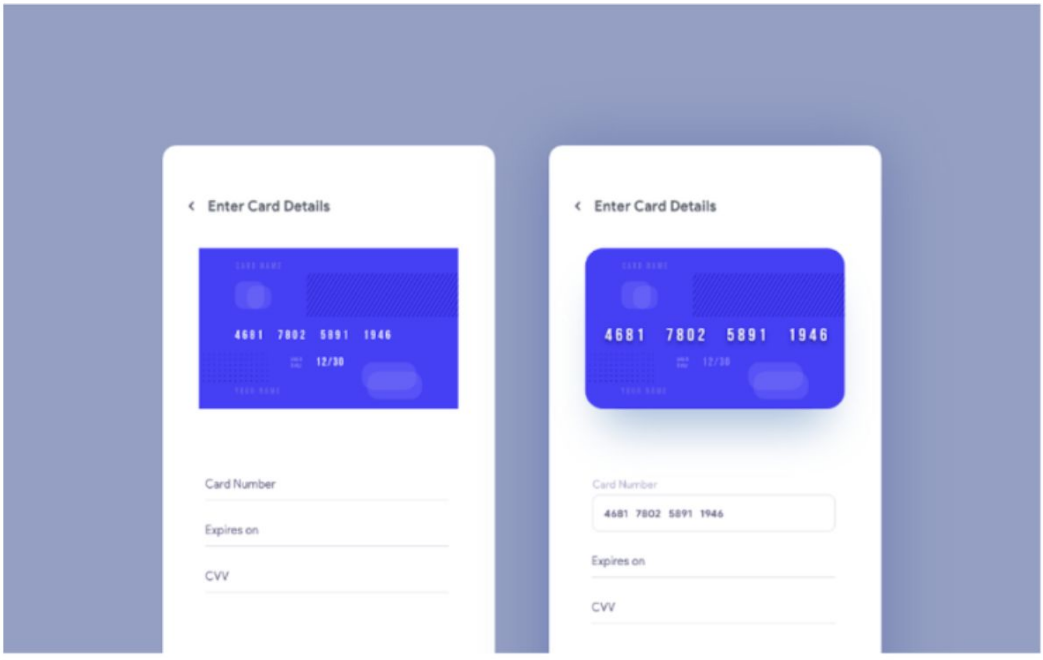
In this UI, the progress bar on top is showing the user 3 out of 5 images are uploaded to make sure the user is well aware of the progress and can wait until hesitation until the process is completed.



- ◯ Visibility of system status
- ◯ Match between system and the real world
- ◯ User control and freedom



- ◯ Visibility of system status
- ◯ Match between system and the real world
- ◯ User control and freedom



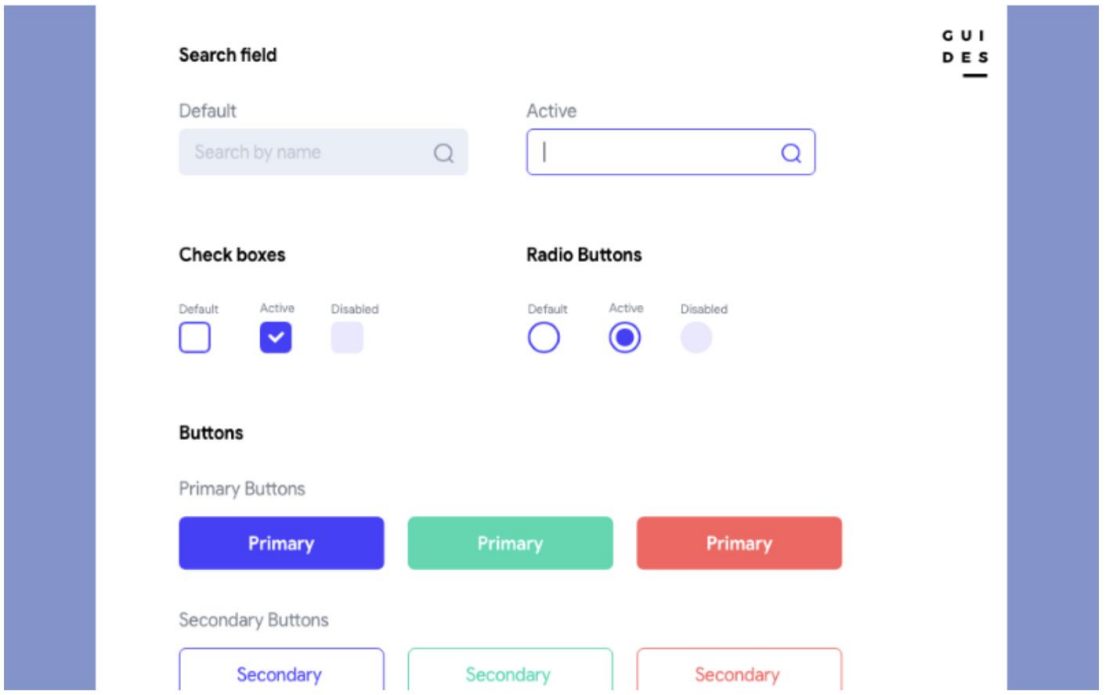
- ◯ Visibility of system status
- ◯ Match between system and the real world
- ◯ User control and freedom

#4 Consistency and standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.
- A Submit button in one page should look the same across the site on any page.
- If we show the data in a particular table format on one page, it should look the same the next time data is being shown in tabular format.
- If we are choosing popup as a solution to a problem, then use the same solution on all the similar situations. Let them used to it.

#4 Consistency and standards

User the same design language across a product.

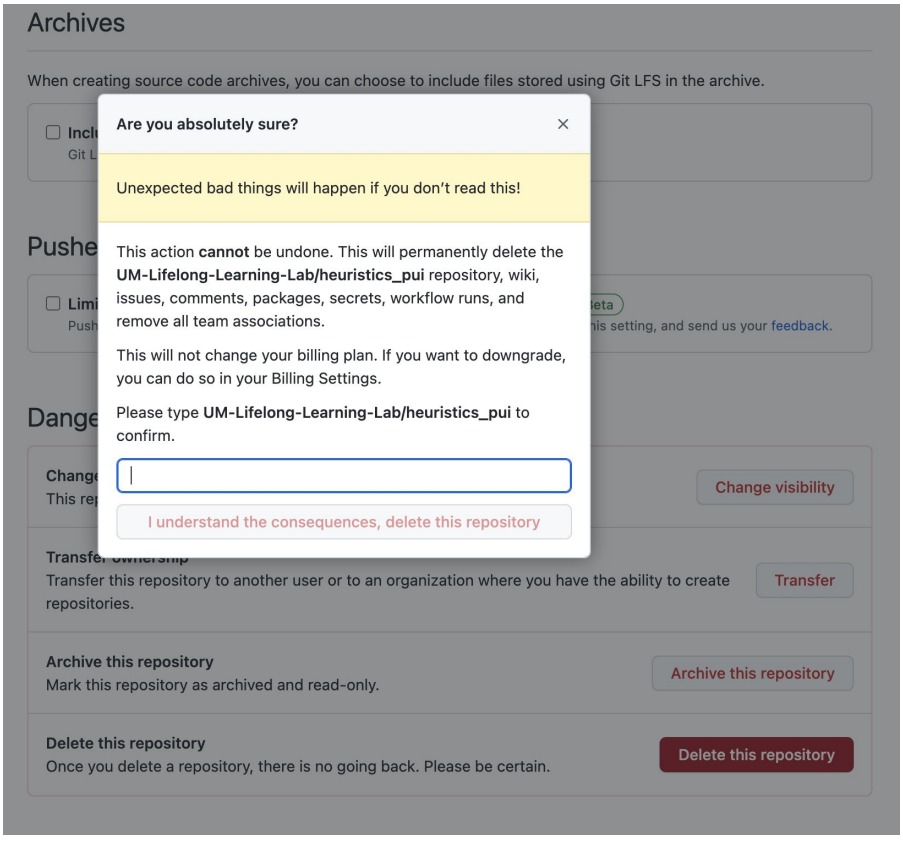


#5 Error prevention

- Good error messages are important, but the best designs carefully prevent problems 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.
- How many times did your outlook remind you that there is no attachment in the email while you mentioned that something is attached?
- Outlook intuitively scans the email for such keywords and alerts the user before sending.
- Double-checking before a deletion

#5 Error prevention

Github double checks with users before they delete a project.



#5 Error prevention

Stop errors before they occur

Enter birthdate: [ ] (bad)

ERROR: Birthdate must be entered in MM/DD/YYYY format

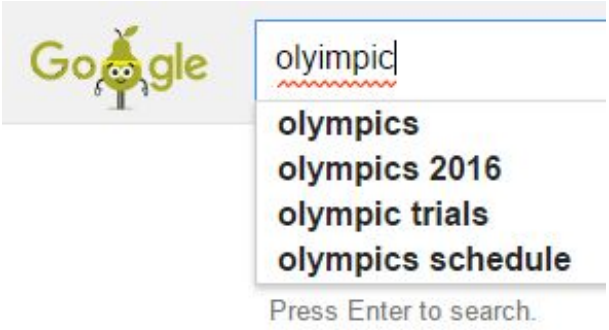
Enter birthdate (MM/DD/YYYY): [ ] (better)

Enter birthdate: [ ] / [ ] / [ ] (best)

Month Day Year

#5 Error prevention

Auto-correct of spelling



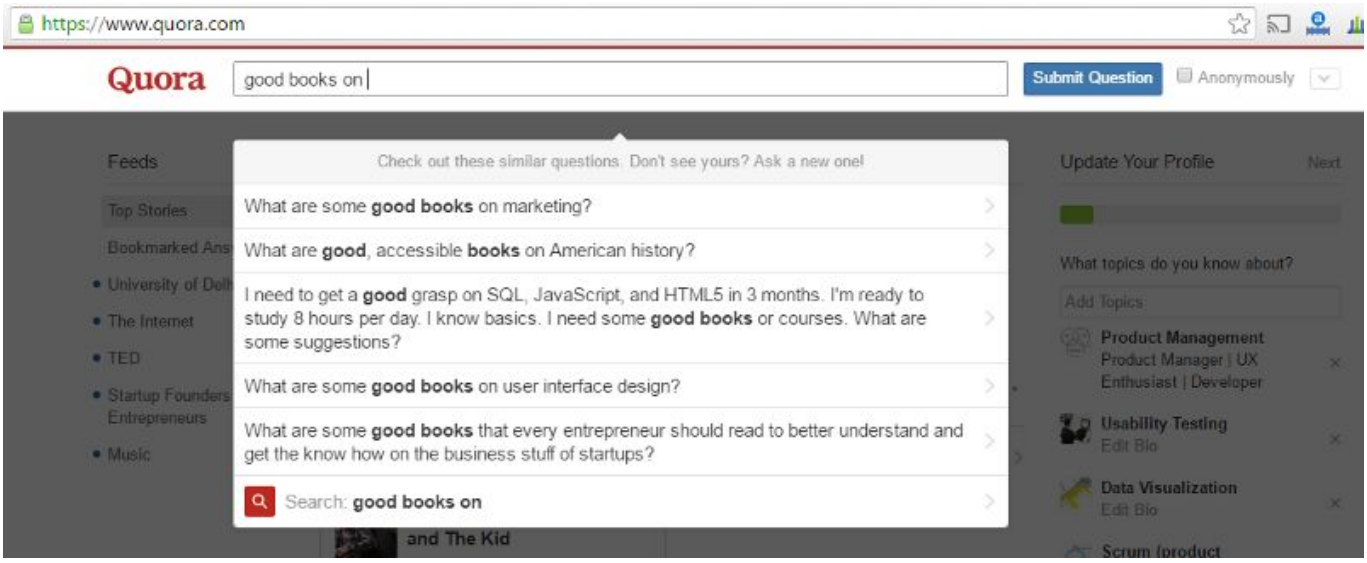
Other examples (password validation), a signal that the password doesn't meet certain rules

#6 Recognition rather than recall

- Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.
- It’s always better to suggest the user a set of options than to let them remember and type the whole thing.

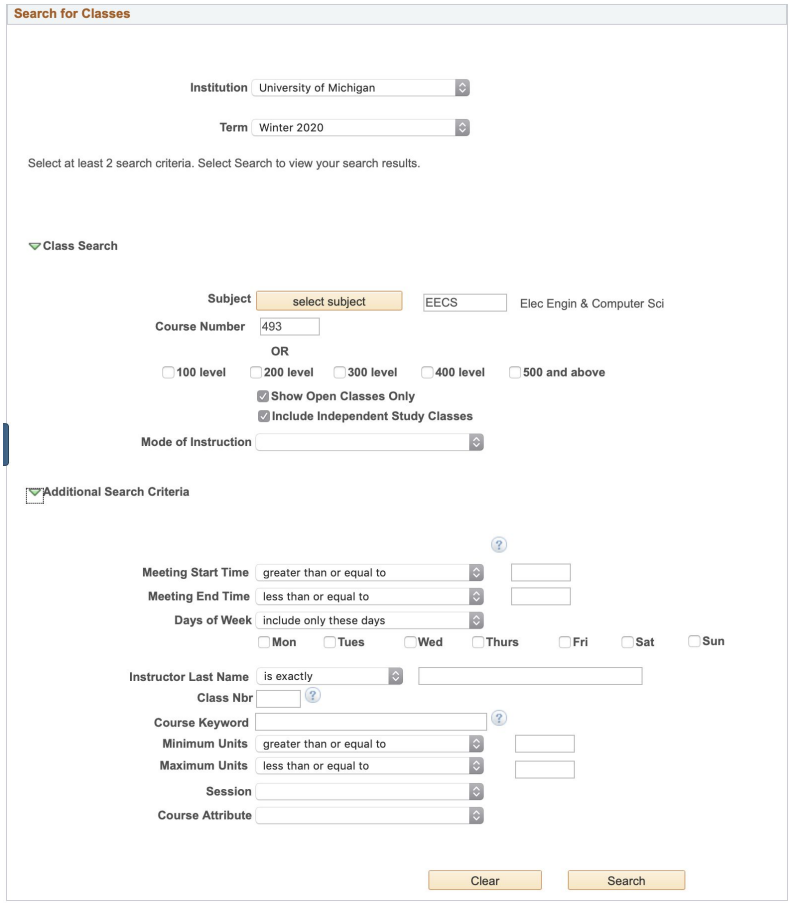
#6 Recognition rather than recall

Autocompletion in search



#6 Recognition rather than recall

Bad example, users have to memorize information

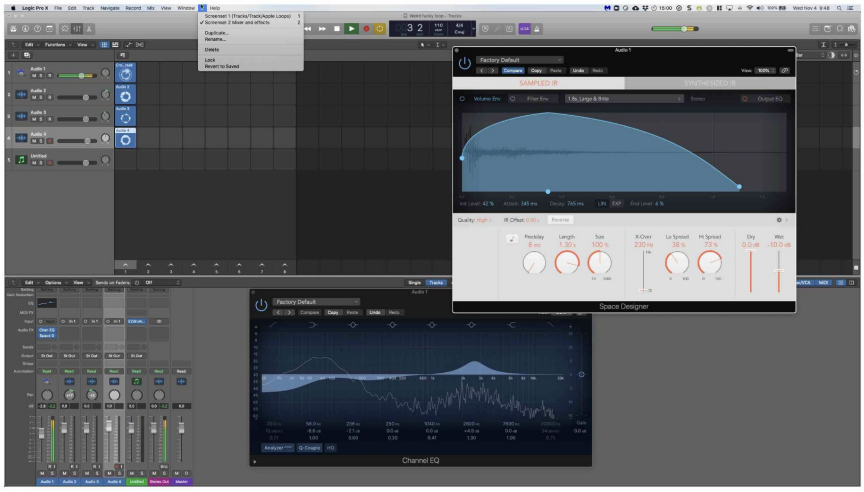


#7 Flexibility and efficiency of use

- Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- Example: when installing a software, it asks if the user wants to go ahead with default installation or custom installation
- Shortcuts

#7 Flexibility and efficiency of use

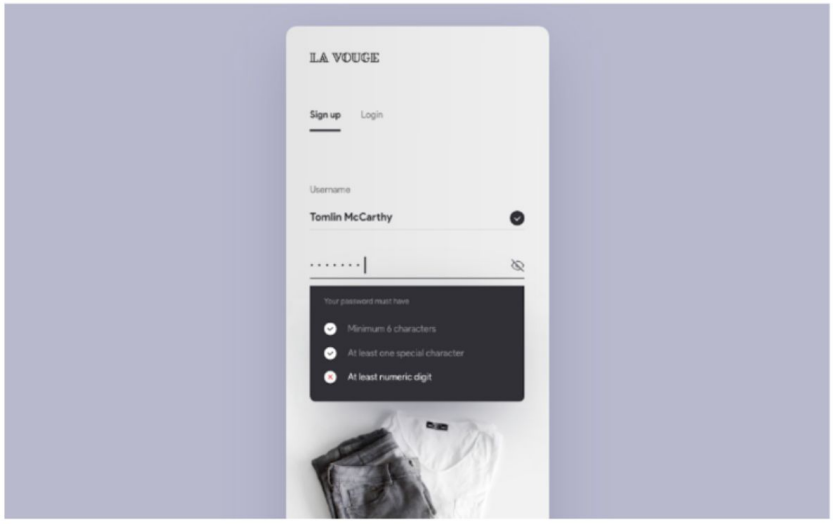
Logic Pro X allows users to set up their windows into arrangements that can be recreated with a single keystroke or through the menu. This capability allows power users to flexibly (and efficiently) customize their display throughout the various subtasks involved in recording or mixing music.



Lecture 13 Survey 2

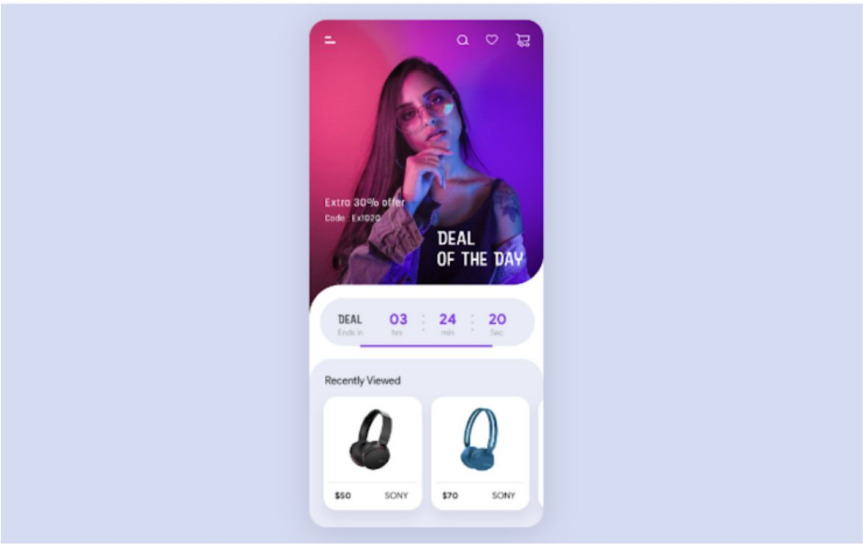
www.yellkey.com/remove ⇒ https://forms.gle/jbrH8Pw62rEiQ43d7

In this UI, it shows the password standards which a user needs to follow while setting up the password, so that the user can complete the task successfully in their first attemp.



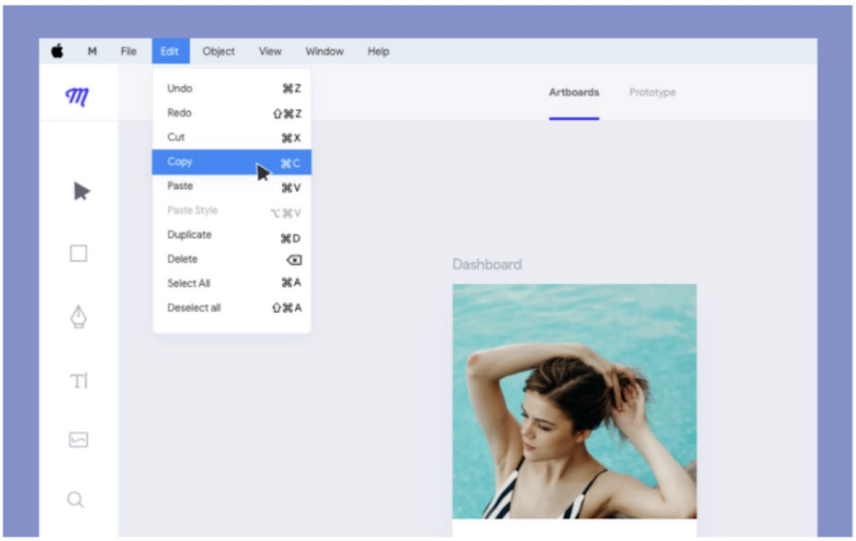
- ◯ Error Prevention
- ◯ Recognition rather than recall
- ◯ User control and freedom
- ◯ Flexibility and efficiency of use

In this UI, the app keeps the user's recently viewed item in the home screen that will remind the user about their last searchers and items needed to buy.



- ◯ Visibility of system status
- ◯ Recognition rather than recall
- ◯ User control and freedom
- ◯ Error prevention

In this app, when the user wants to copy-paste an item, they can either go to the menu and select copy, or they can also do command + C as a short cut.

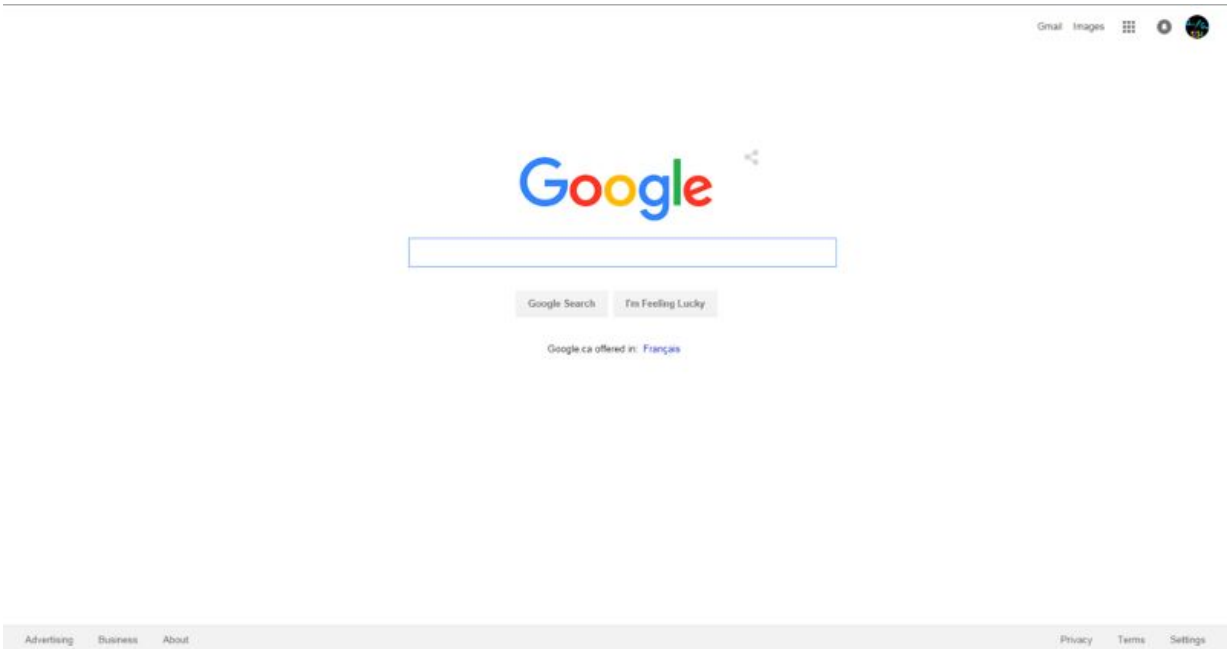


- ◯ Flexibility and efficiency of use
- ◯ Match between system and the real world
- ◯ User control and freedom
- ◯ Error prevention

#8 Aesthetic and minimalist design

- Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.
- Is every information displayed on interface necessary and useful?

#8 Aesthetic and minimalist design

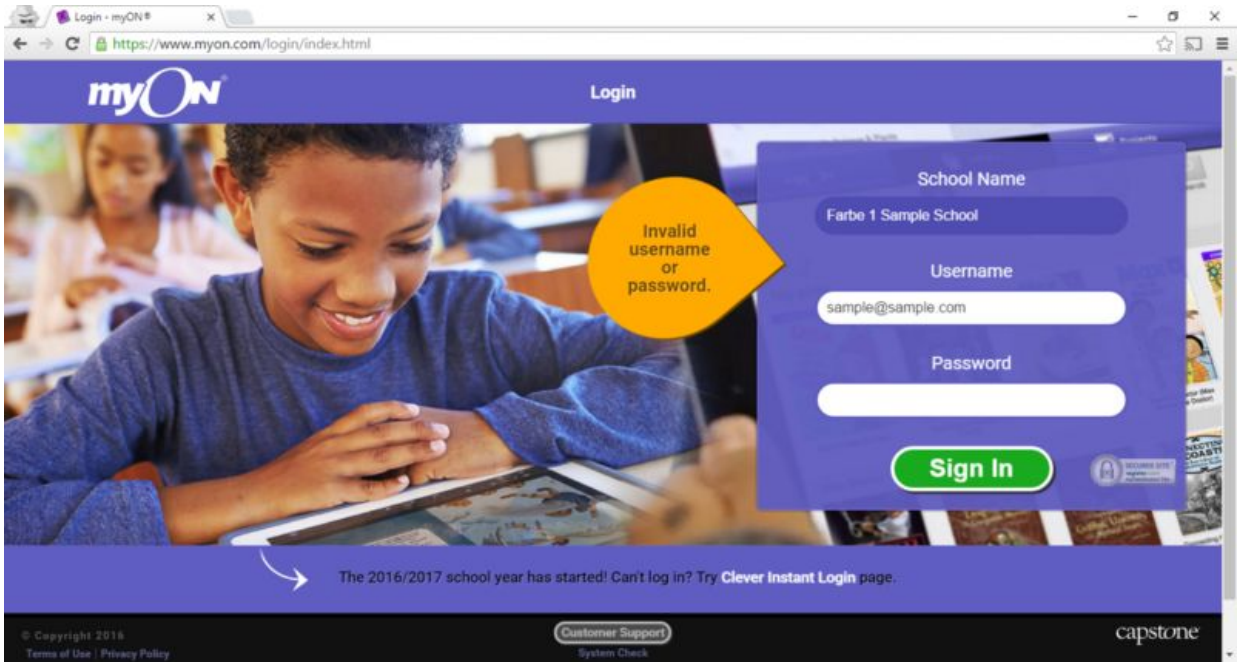


#9 Help users recognize, diagnose, and recover from errors

- Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.
- Use traditional error message visuals, like bold, red text.
- Tell users what went wrong in language they will understand — avoid technical jargon.
- Offer users a solution, like a shortcut that can solve the error immediately.

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

Bad error message, what does this mean?

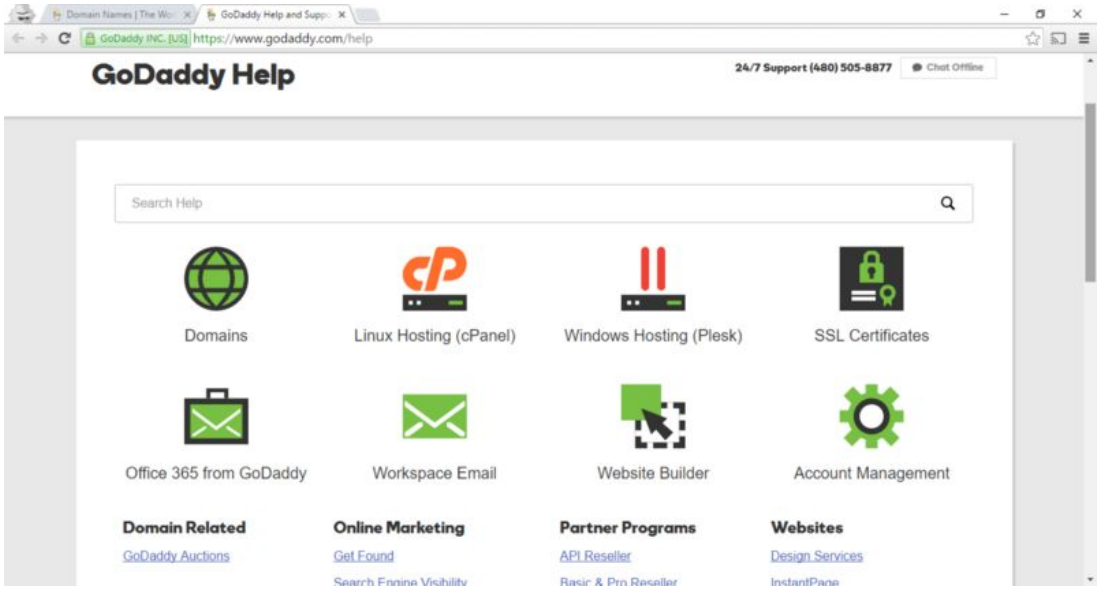


#10 Help and documentation

- If a user has reached this step, something is not right with the interface. A great user interface lets the user navigate through its features without any documentation or training.
- But if there is any user who could not make it out, adequate help should be provided within the product.
- Ensure that the help documentation is easy to search.
- Whenever possible, present the documentation in context right at the moment that the user requires it.
- List concrete steps to be carried out.

#10 Help and documentation

Categorize different areas in a proper way so that the user can find a solution more easily.



Advantages and Disadvantages of Heuristic Evaluation

- Fast, Cheap
  - Can be scaled up or down depending on the needs/resources
- Tends to find the most severe issues
  - Good for use in early phases of rapid iterative design
- Find issues related to basic UI properties, e.g., learnability, discoverability
- Good for use in early phases of rapid iterative design

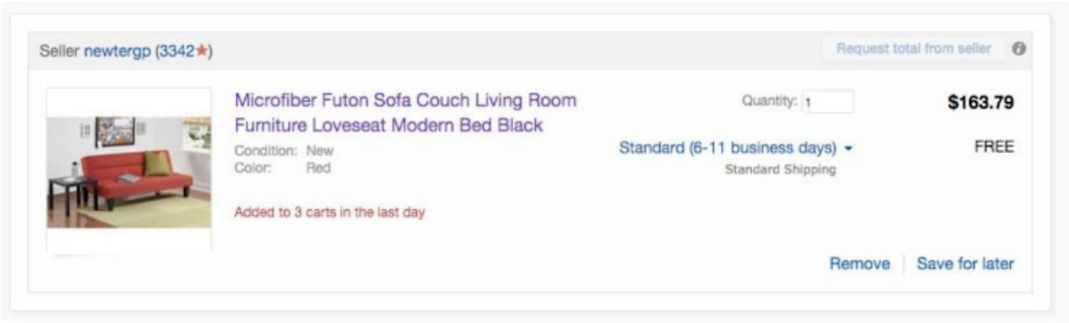
Advantages and **Disadvantages** of Heuristic Evaluation

- False positives
  - Can find issues that are unlikely to arise
- Need people who understand the heuristics
  - E.g., you
- Hard to separate out issues that affect different users

Lecture 13 Survey 3

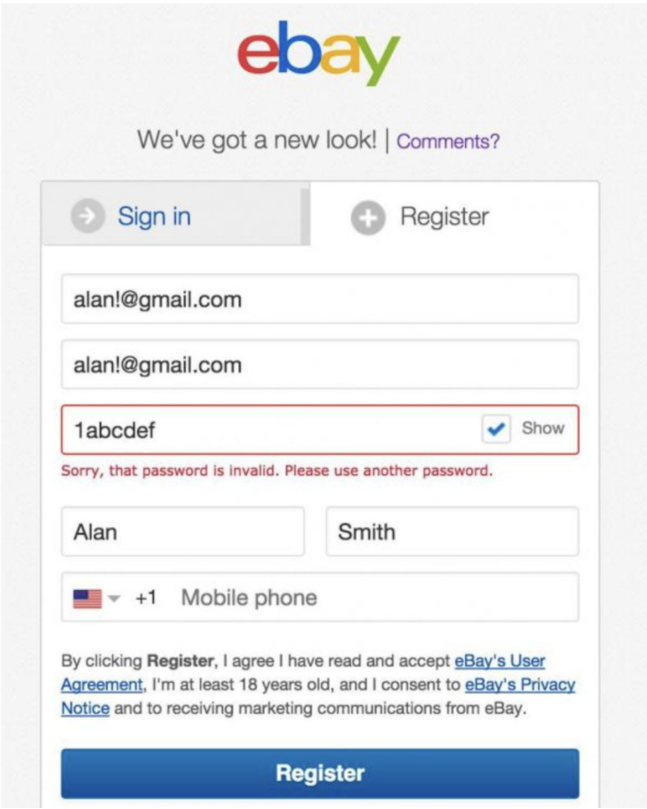
www.yellkey.com/white ⇒ https://forms.gle/Sgixzi9e24hwcKDeA

Here is a screenshot from Ebay (www.ebay.com). The user added an item to the cart. The problem with this interface is that it does not allow the user to change the attributes of the item that is in their shopping cart. Please describe which heuristic rule does this violate? *



- ◯ Consistency and standards
- ◯ Error prevention
- ◯ User control and freedom
- ◯ Flexibility and efficiency of use

Here is a screenshot from Ebay (www.ebay.com). The user Alan is creating an account. After entering the password, the system sends an error message "Sorry, that password is invalid..." Please describe which heuristic rule does this violate? *



Nielsen’s Usability Heuristics

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

Questions?

Nielsen’s Usability Heuristics

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

Questions?
Assignment 4
Evaluate your prototype