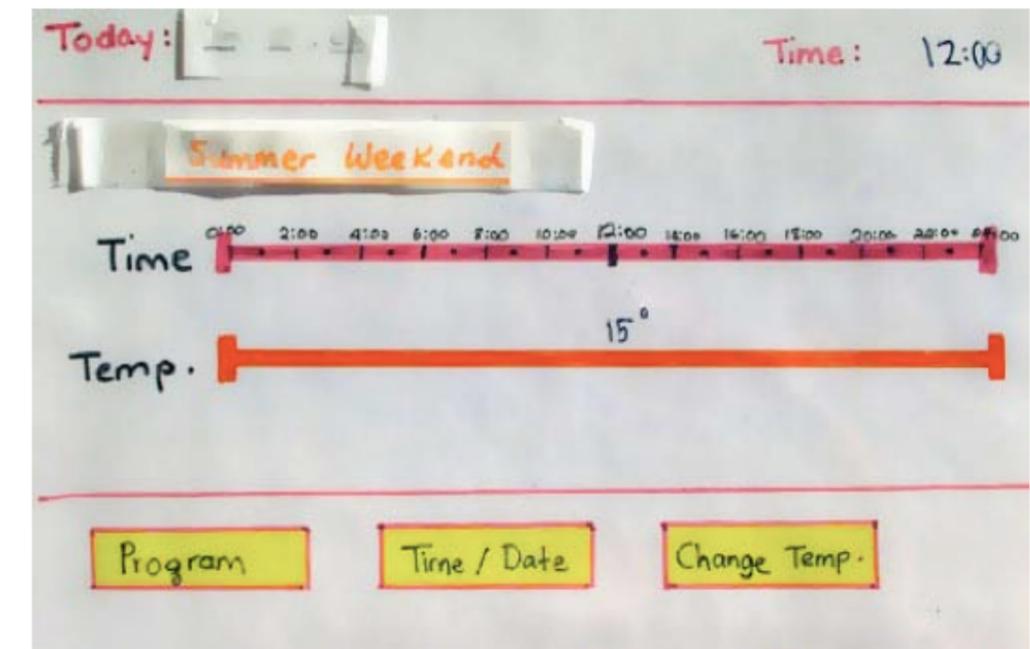
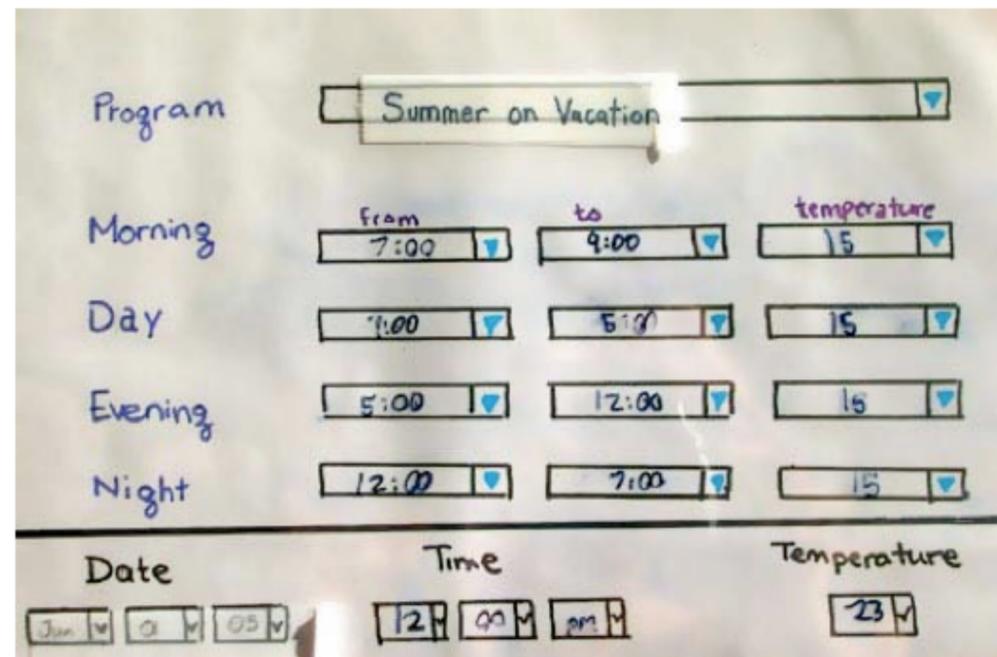
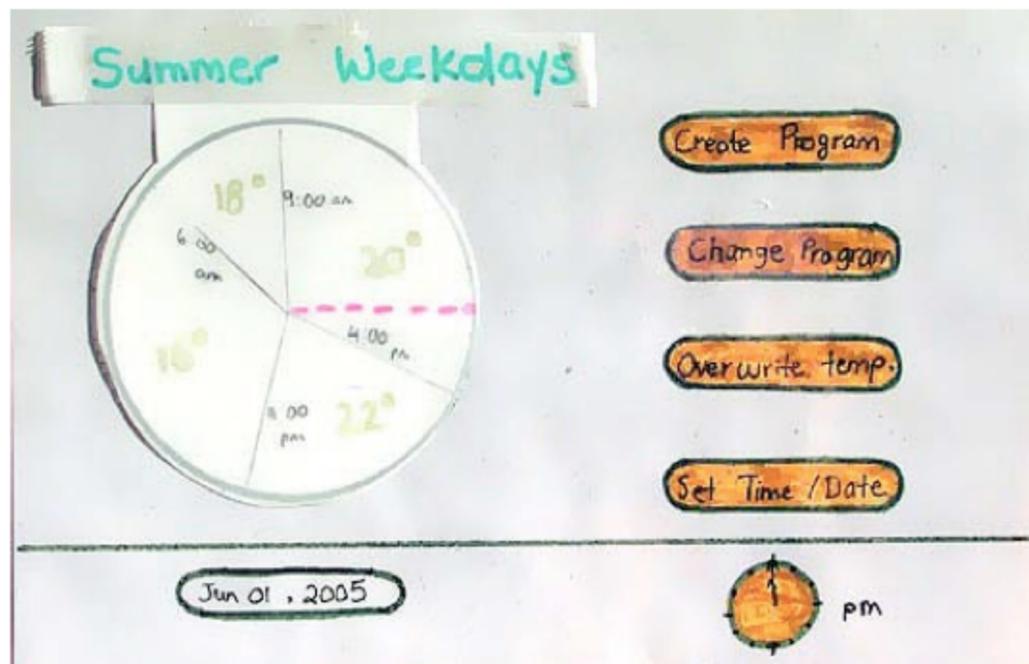


# CS122 Class 13: Prototyping



Three different paper prototypes for the same idea

<http://courses.csail.mit.edu/6.831/2014/readings/L10-prototyping/#paper-prototypes>

# Class 13 agenda

- ZC
- Lecture: From design goals to designs: prototyping!
- Break
- Studio: Paper prototypes (wireframes)
- Midsemester feedback form

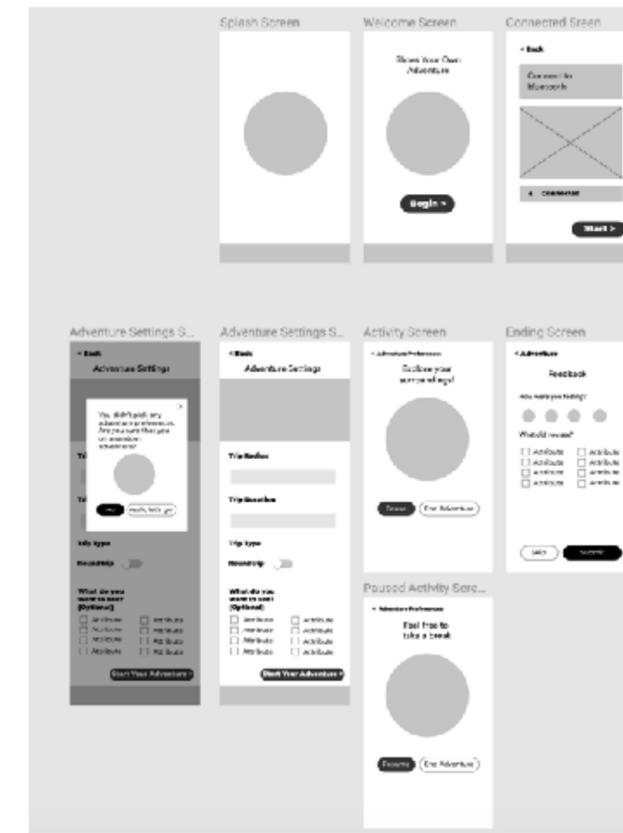
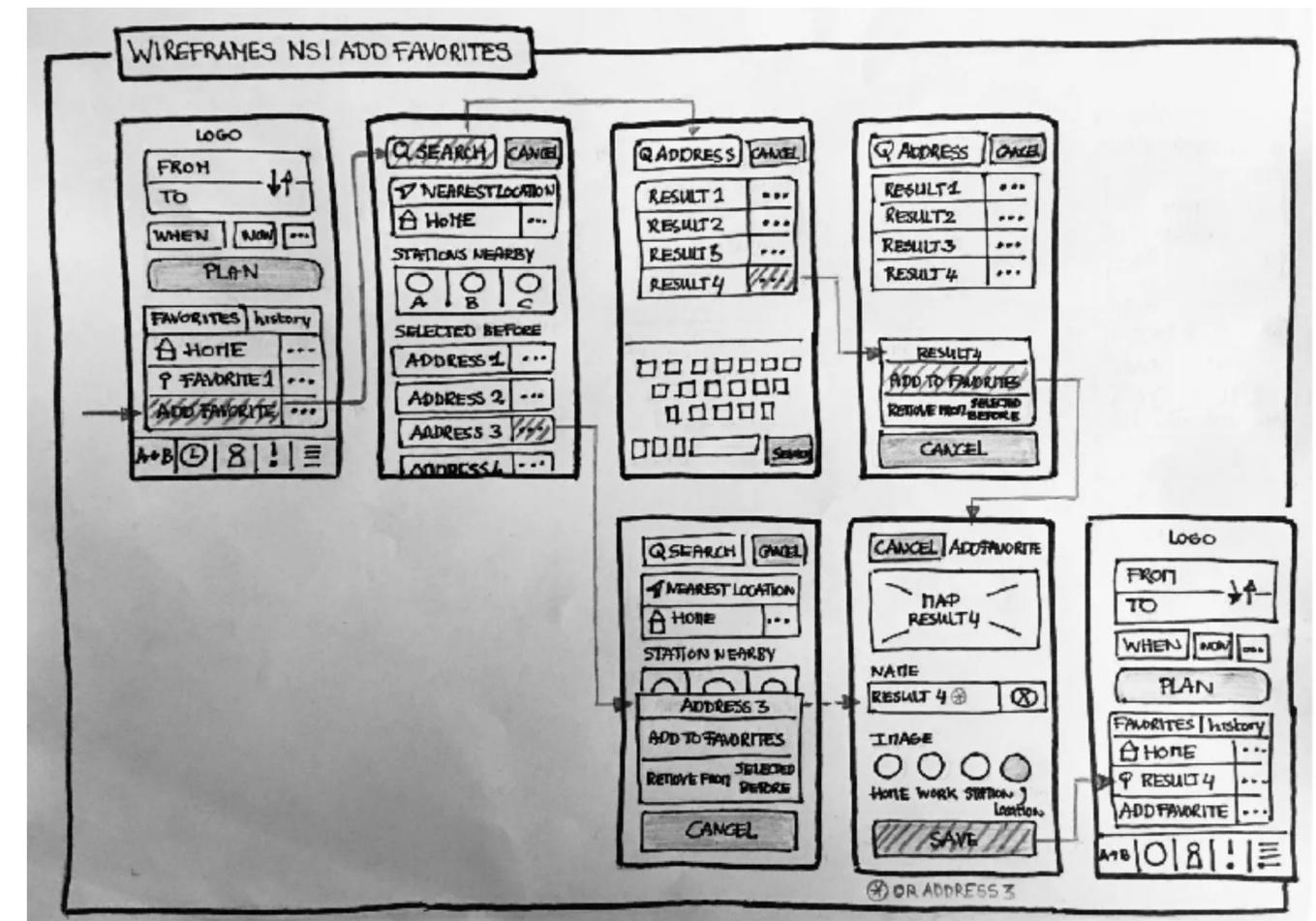
# Prototypes

# From design goals to design

- Prototypes are instantiations of ideas that can be informed by your user needs and design principles
- Prototypes force you to make design decisions
- This class we'll be making low fidelity *paper prototypes*. Benefits of paper:
  - Faster to build
  - Easier to change
  - Focuses attention on big picture rather than nitpicking UI details

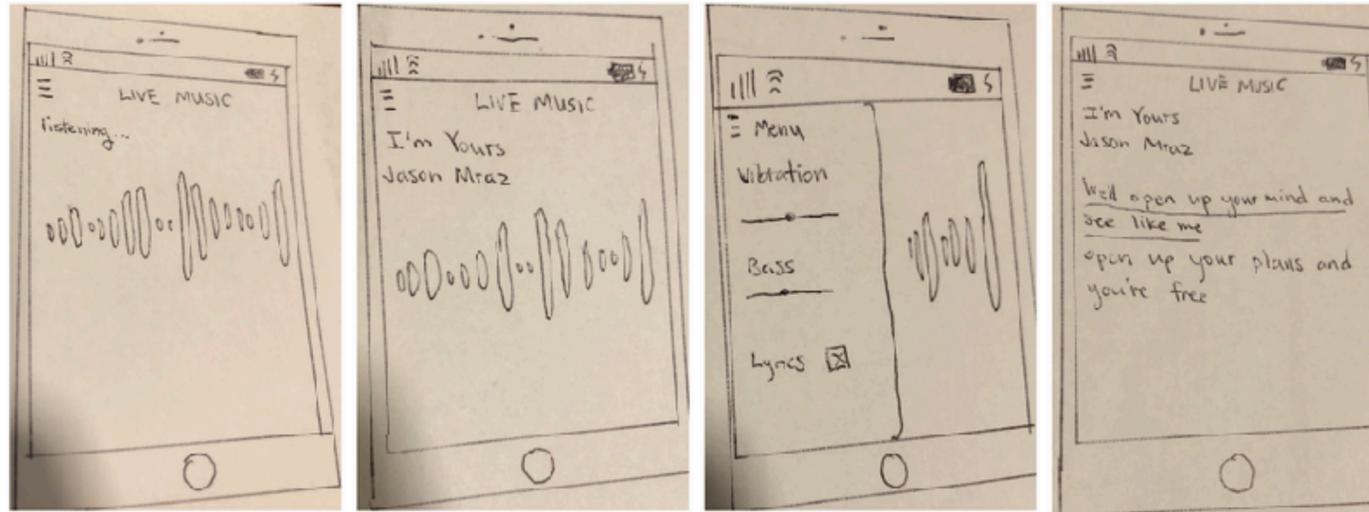
# Wireframe

- A wireframe is an **architectural representation** of a user interface like a screen, window, dialog box or web page.
- Wireframes visually depict the basic layout of an app or website. They typically lack style, color, and graphics as their purpose is to test **functionality** and determine **priority of content**.

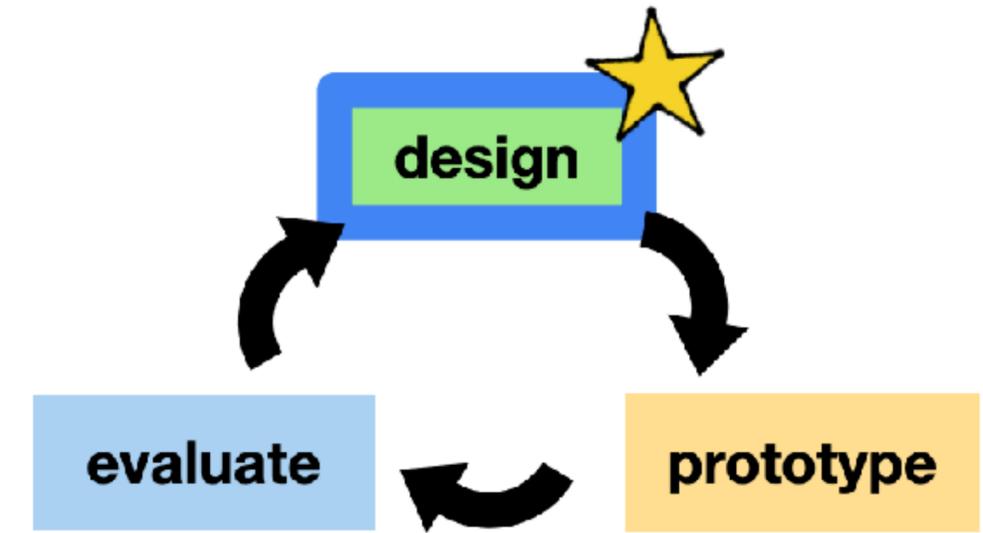
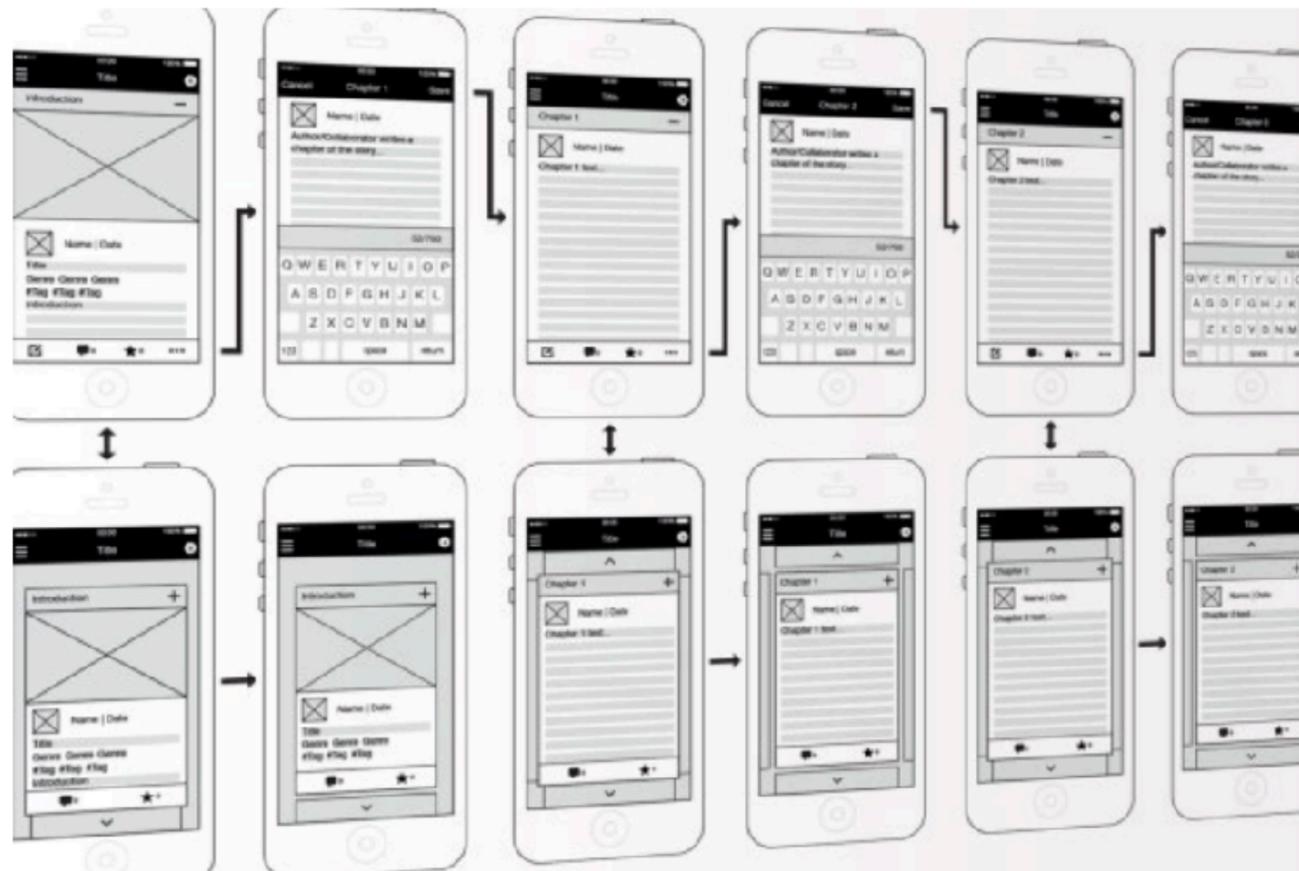


# Sketch vs wireframe

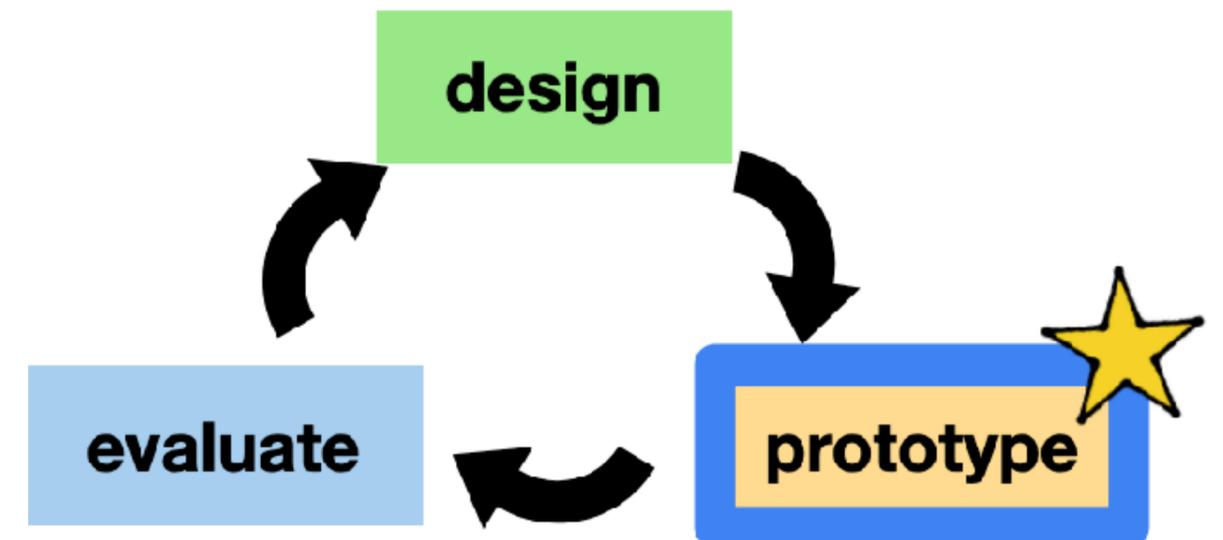
Sketch



Wireframe

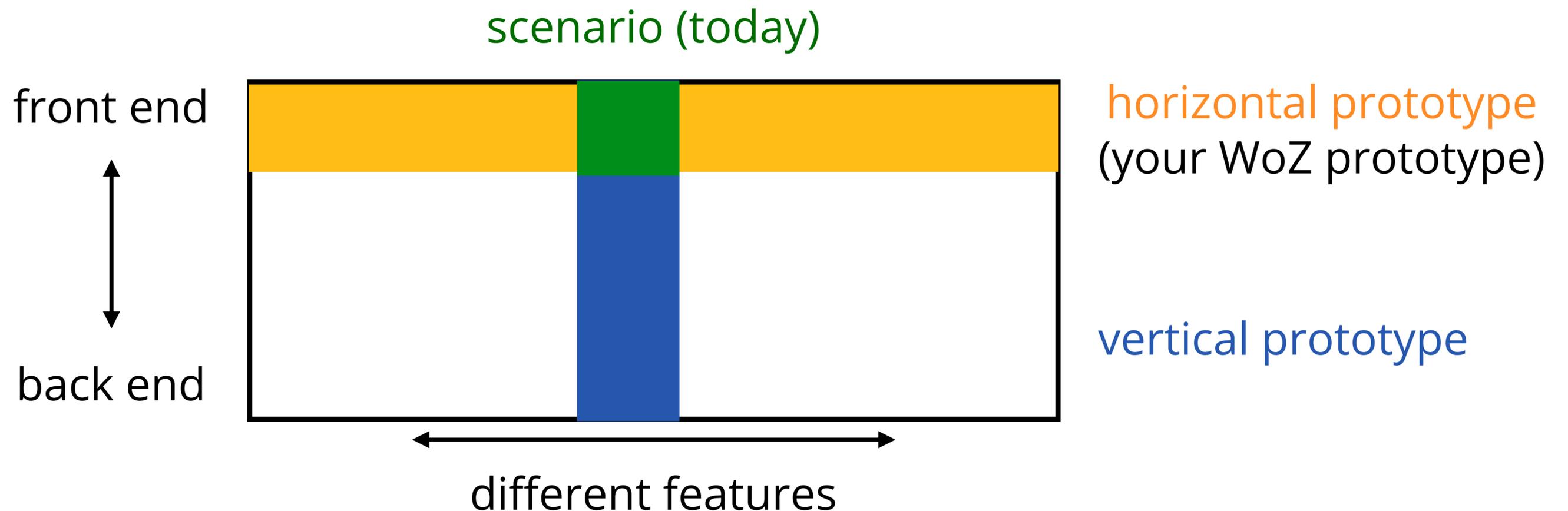


A wireframe fleshes out more details than a sketch, such as specific UI elements



# Prototype fidelity is multidimensional

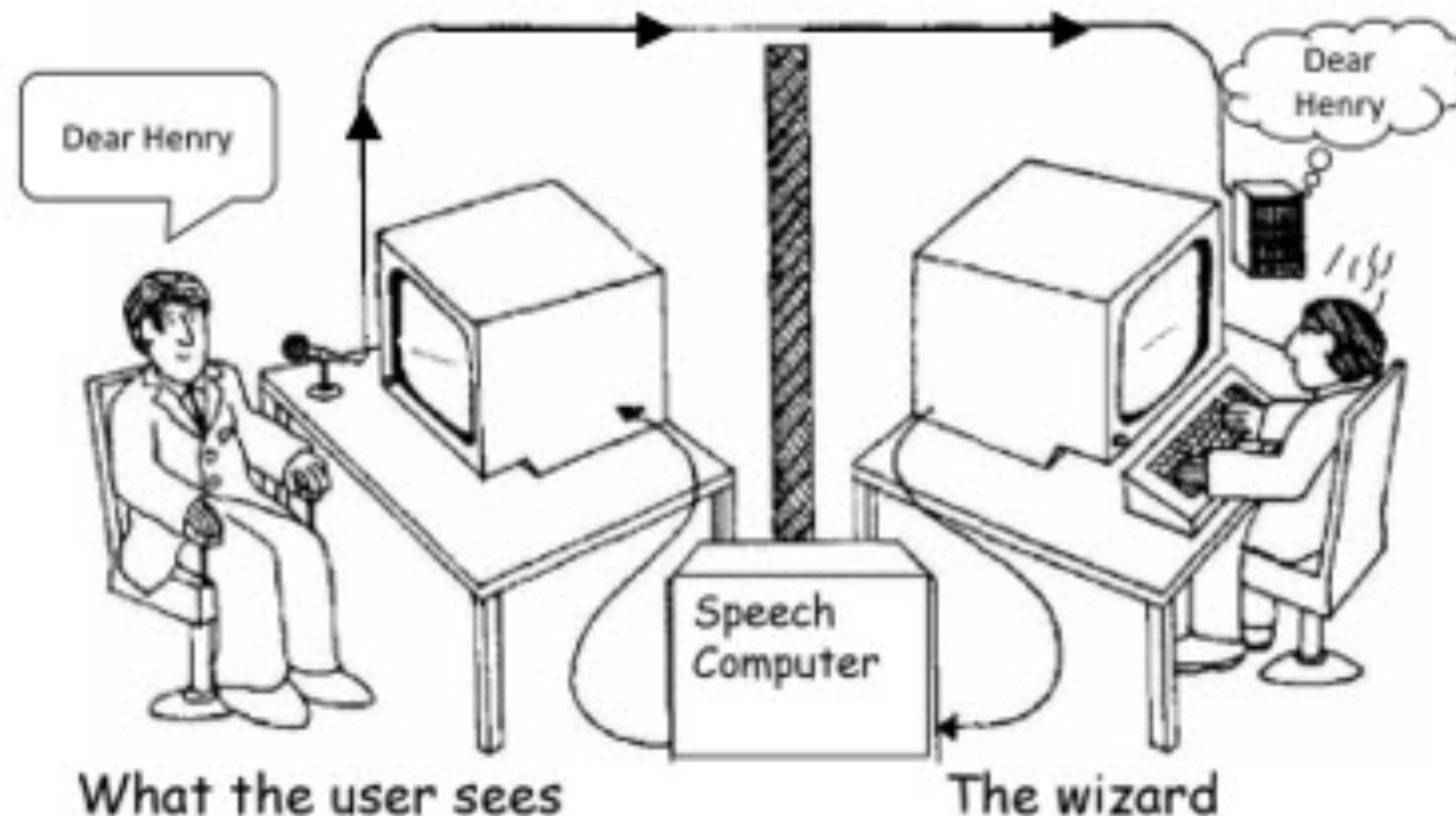
- Breadth (horizontal) - how many features does your prototype cover?
- Depth (vertical) - degree of functionality (how responsive to user inputs is your prototype? Or does it just have "canned" options)



# The Wizard of Oz technique

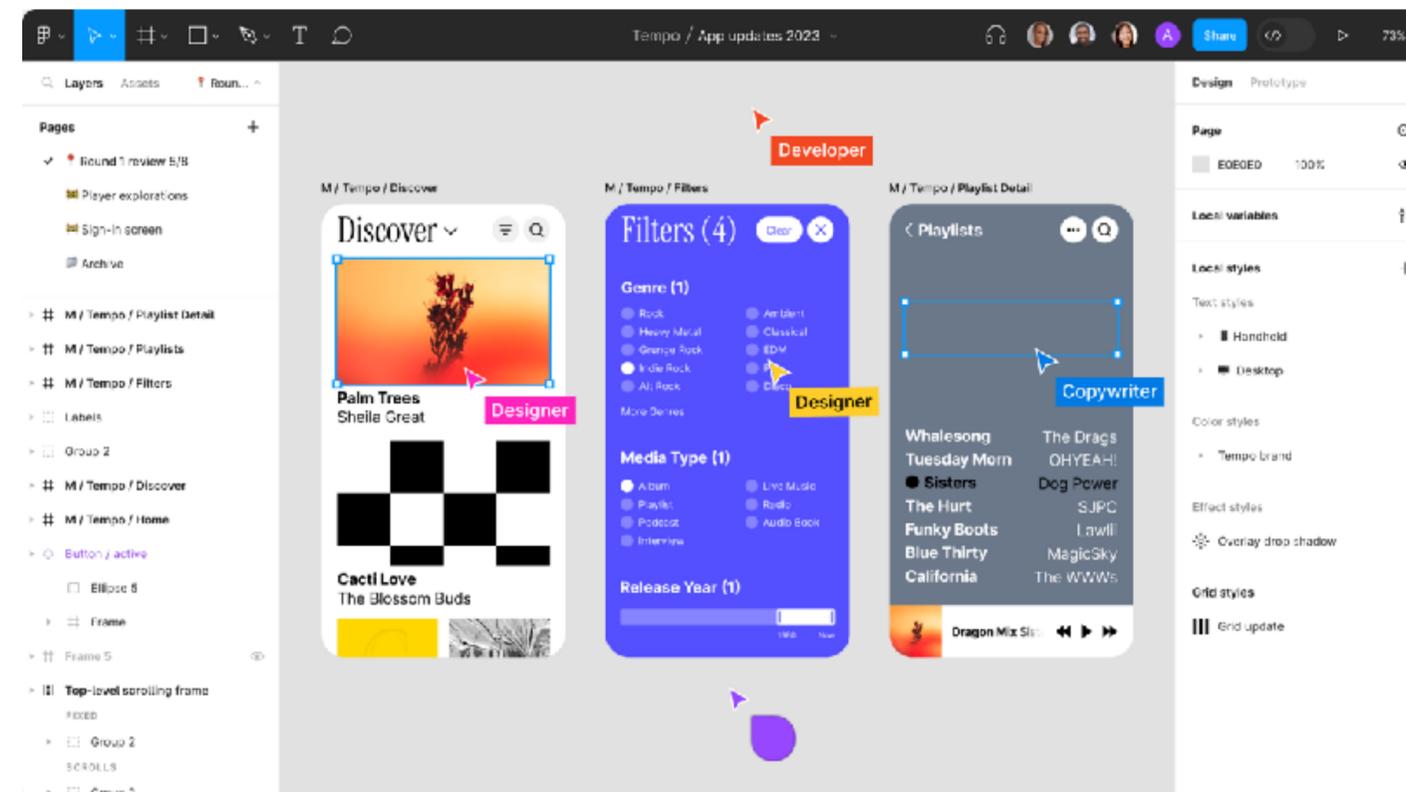
- A setup where subjects interact with a computer system they believe to be autonomous, but the system is actually operated by an (unseen) human being

*Wizard of Oz testing – The listening type writer IBM 1984*



# Your project timeline

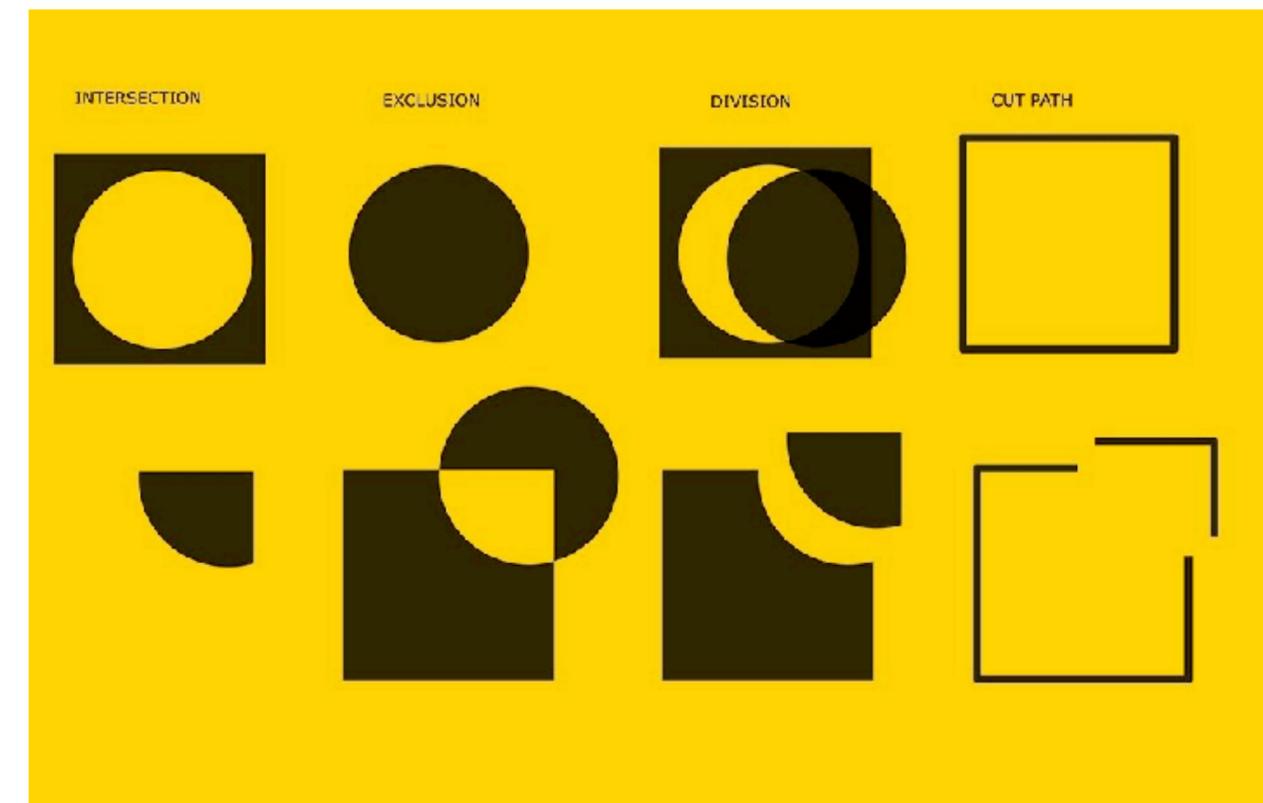
- Today: get started on paper prototyping your main **scenario**
- March 25 (after spring break): evaluate and test paper prototypes in class
- April 8: evaluate and test a Figma Wizard of Oz breadth prototype in class



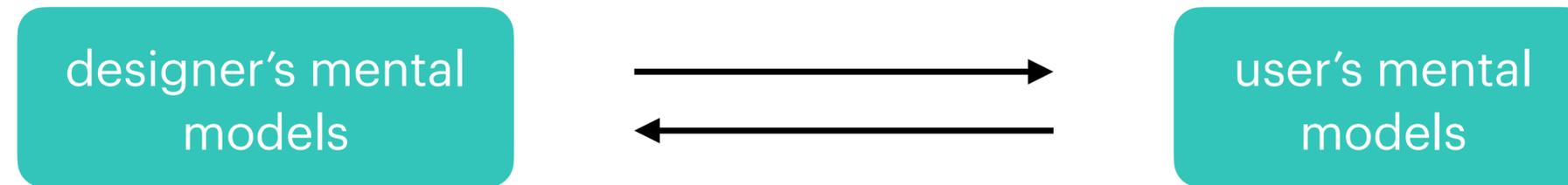
**Going back to mental  
models...**

# Mental models

- An internal representation of external reality that influences our behavior (Kenneth Craik, 1943)
- Different ways of thinking based on the constraints of the tool
- As we use new tools, we develop new mental models
- Ex: Procreate (mental model closer to drawing) vs Inkscape (new vector and boolean mental model)



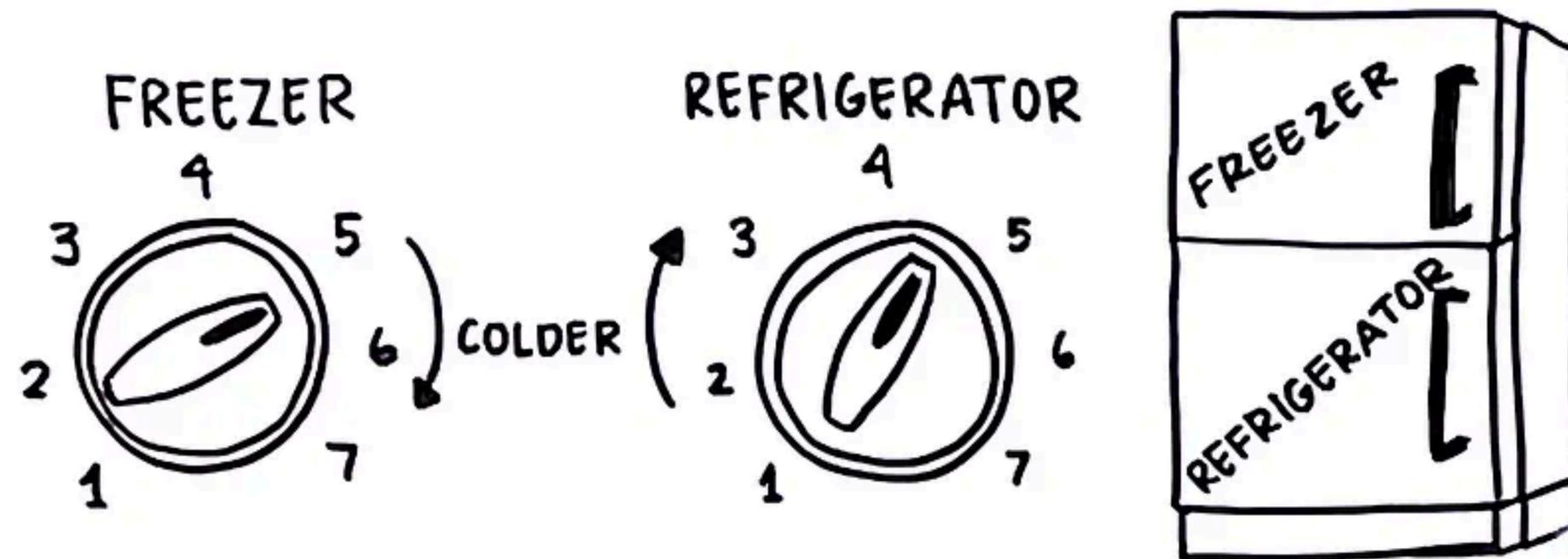
# Conceptual model



- As designers, we want to minimize the gap between a designer's mental model and the user's mental model
- That's the purpose of an *interface*: to communicate our design intent without the need for written instructions/documentation
- "Walk up and use" cases

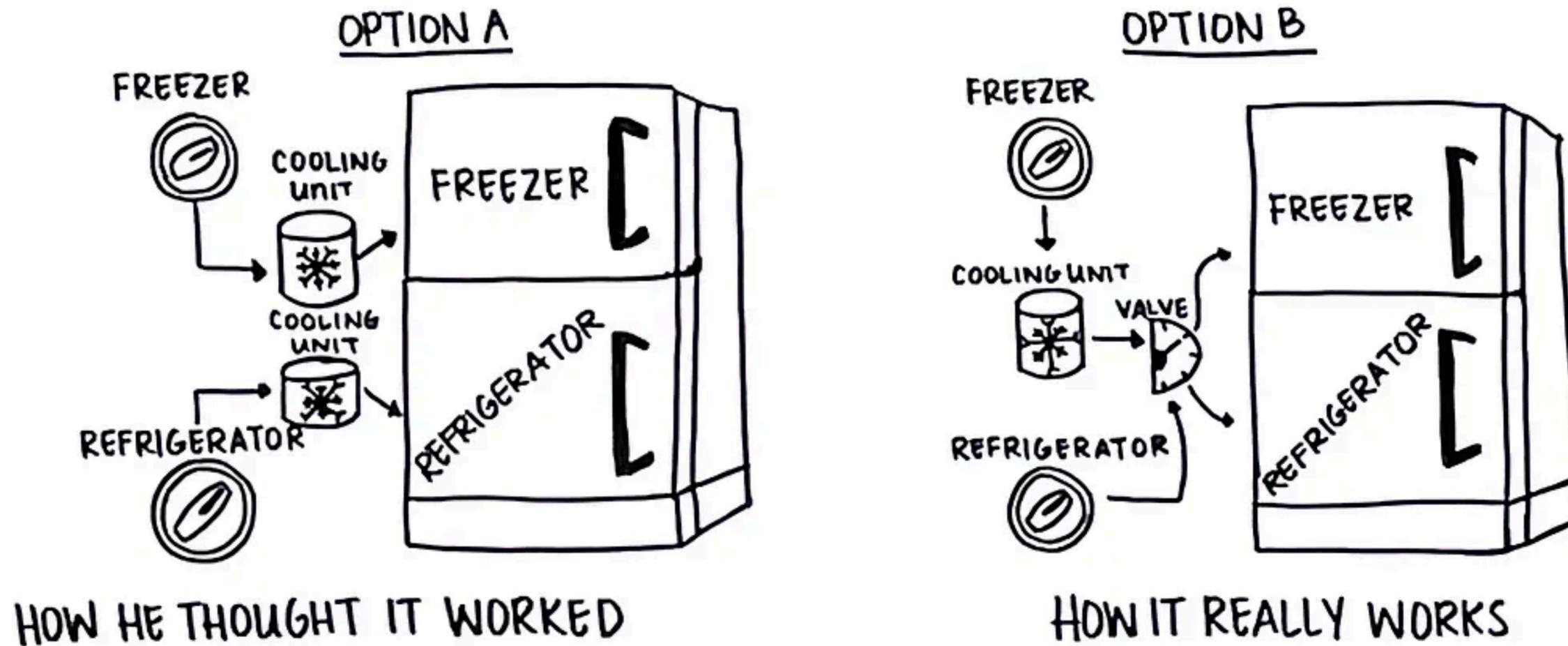
# Conceptual model example

- In *The Design of Everyday Things*, Don Norman gives a fridge example. He wanted to make his freezer colder, so he turned the freezer dial. But it made his refrigerator colder too.



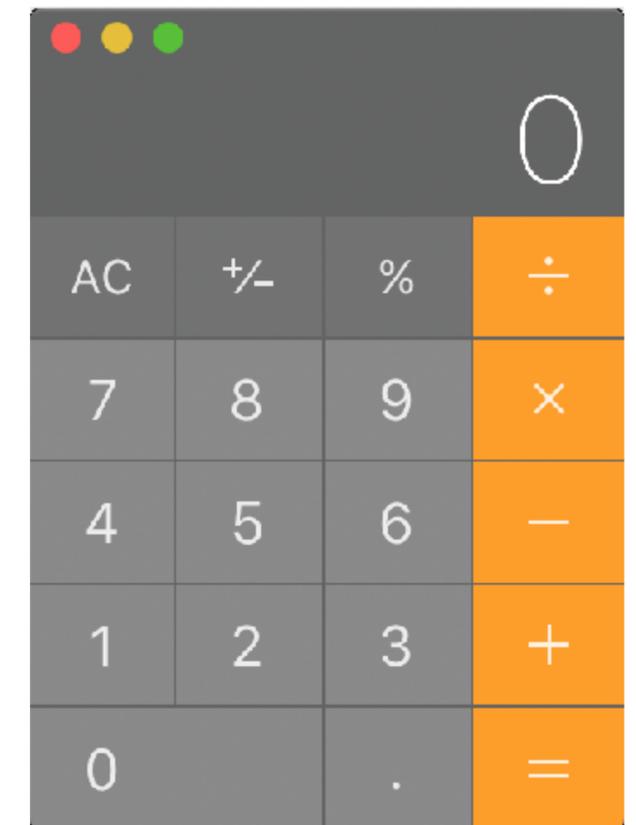
# Conceptual model example

- In reality, the controls were shared: a difference in the designer's versus user's mental models.



# Skeumorphism

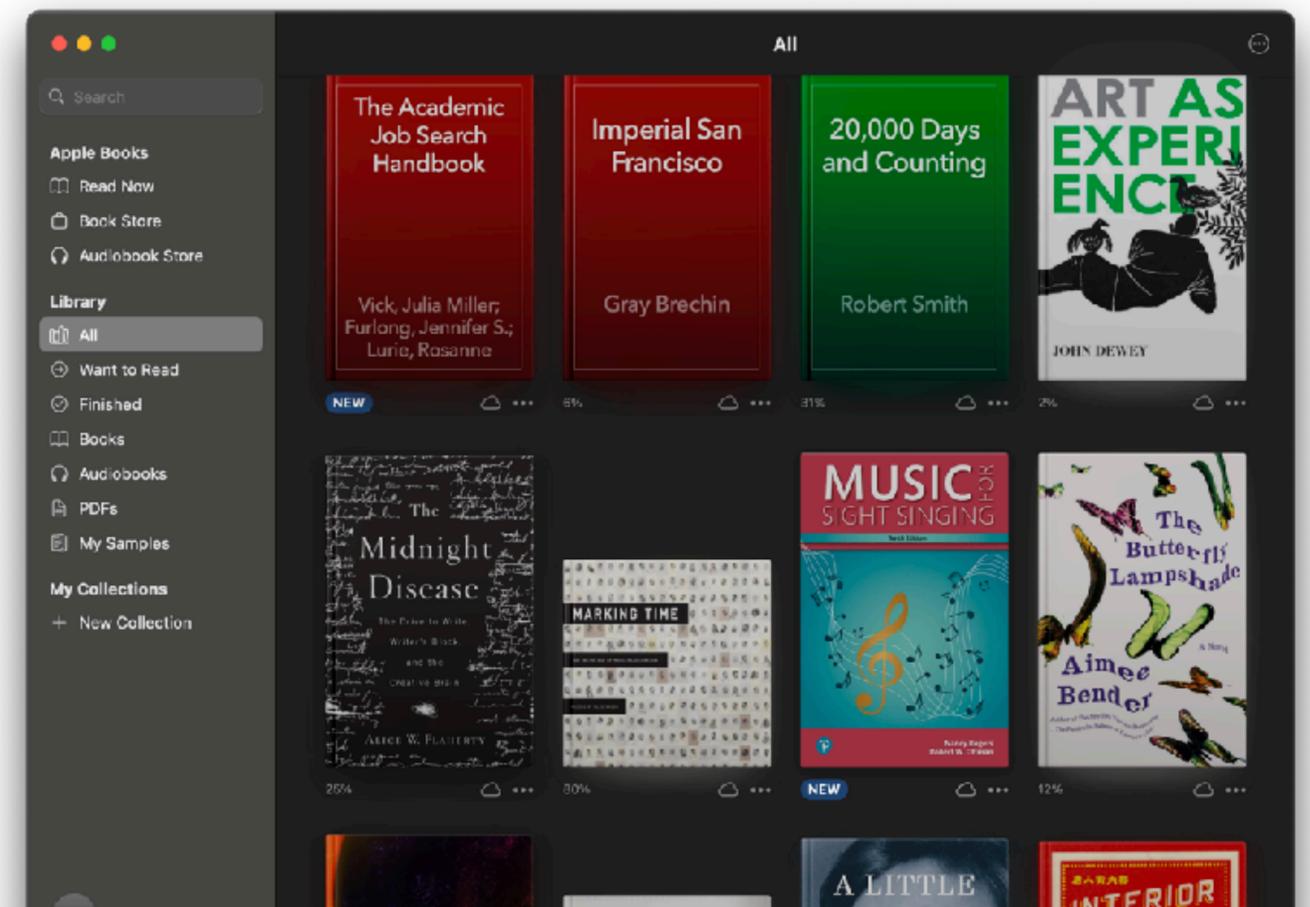
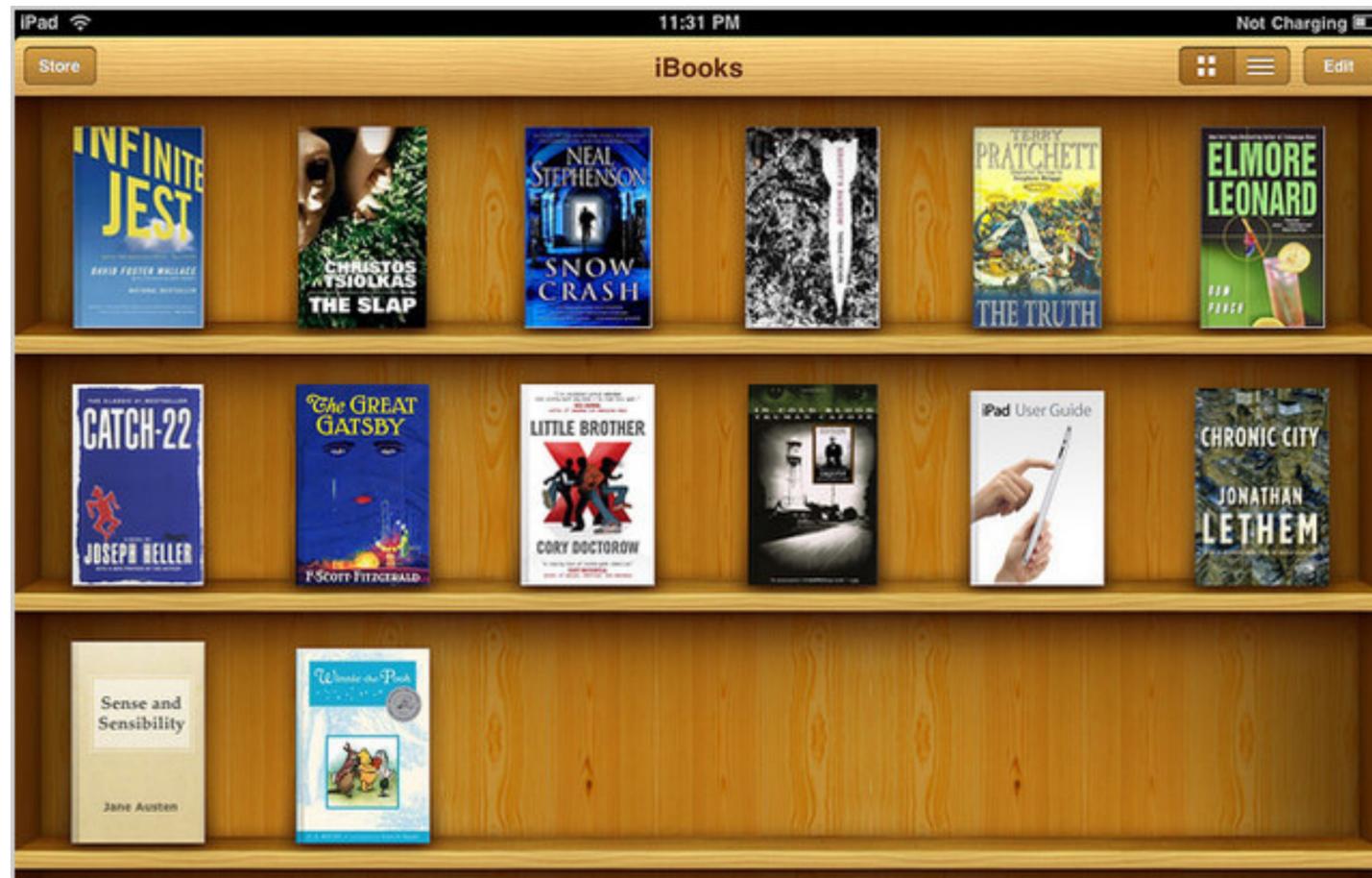
- Using metaphors is a design strategy to help leverage the familiar, but risks in resulting in naive conceptual models
- Skeumorphism is a kind of metaphor where digital GUIs (graphical user interfaces) were made to resemble analog equivalents



# Skeuomorphism

- Pros
- Reuse learned knowledge
- Repurpose conceptual models
- Clearly shows affordances

- Cons
- Limits what users can do
- Implies capabilities that may not exist



# **3 Design Principles for good design**

**1. Make controls visible**

**2. Make mappings clear**

**3. Provide feedback**

# Make controls visible



Poor visibility (how do you set an alarm?)

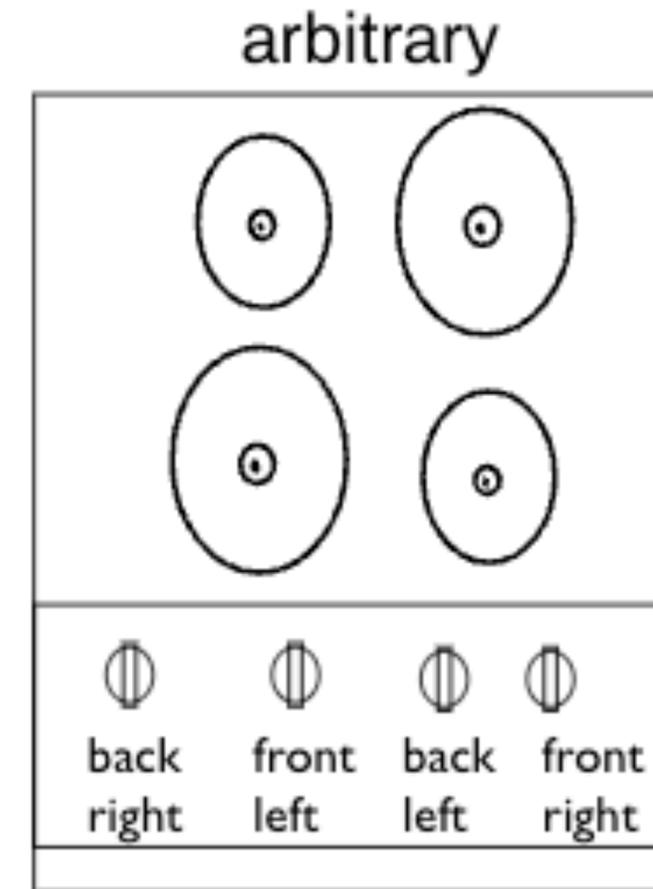


Too much visibility

# Make mappings clear



Good mapping for car seat controls



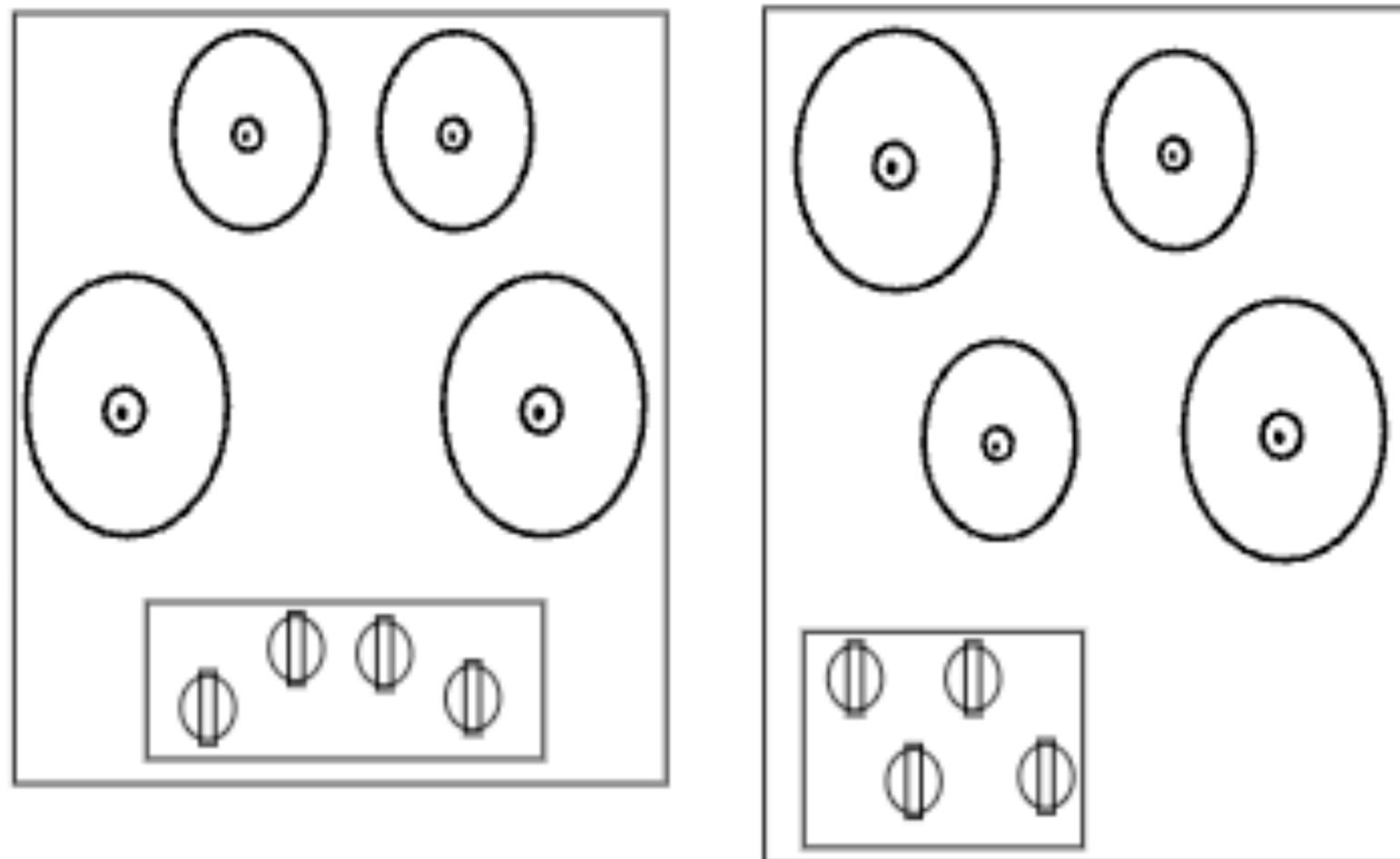
24 possibilities!

Requires:  
visible labels  
memory

Bad mapping for stove top

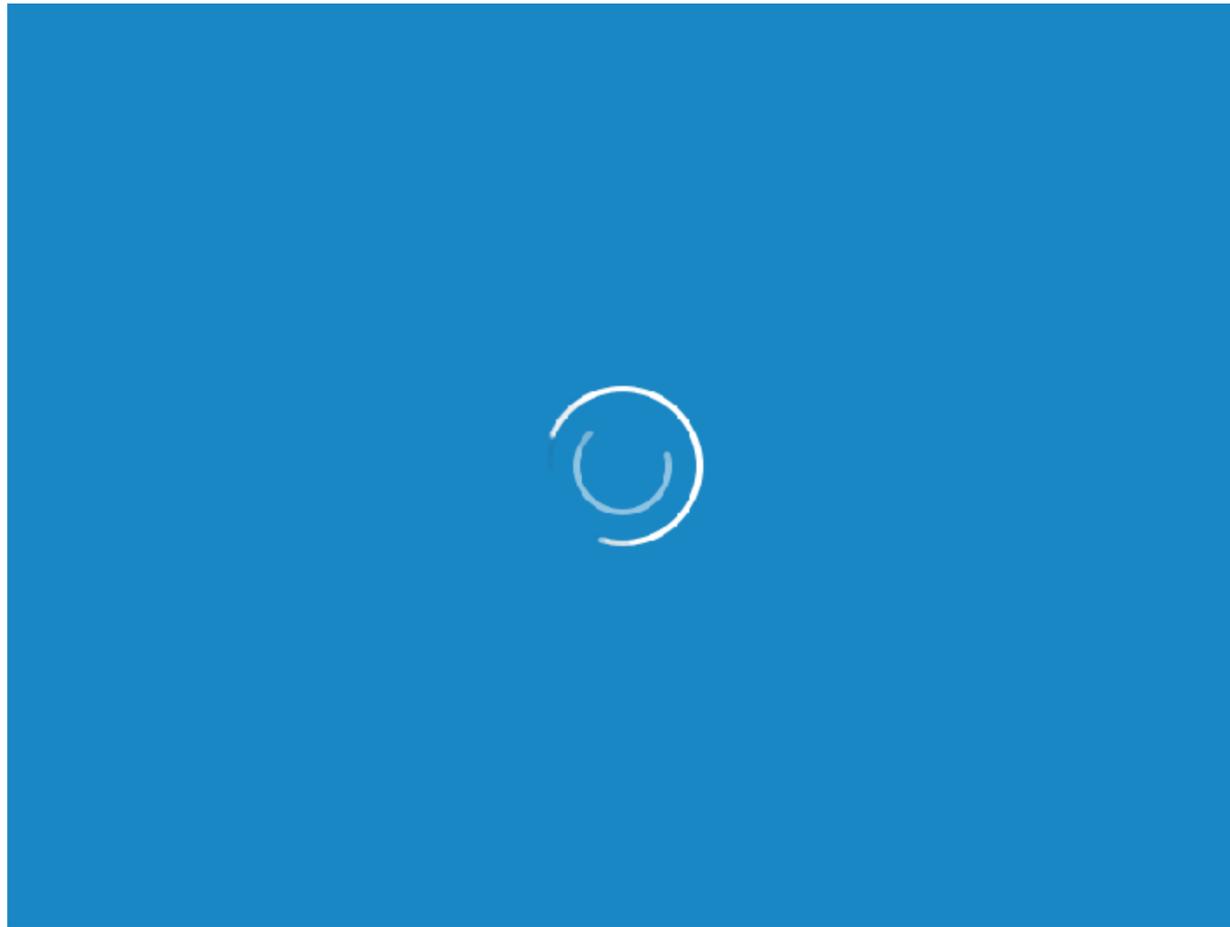
# Make mappings clear

full mapping

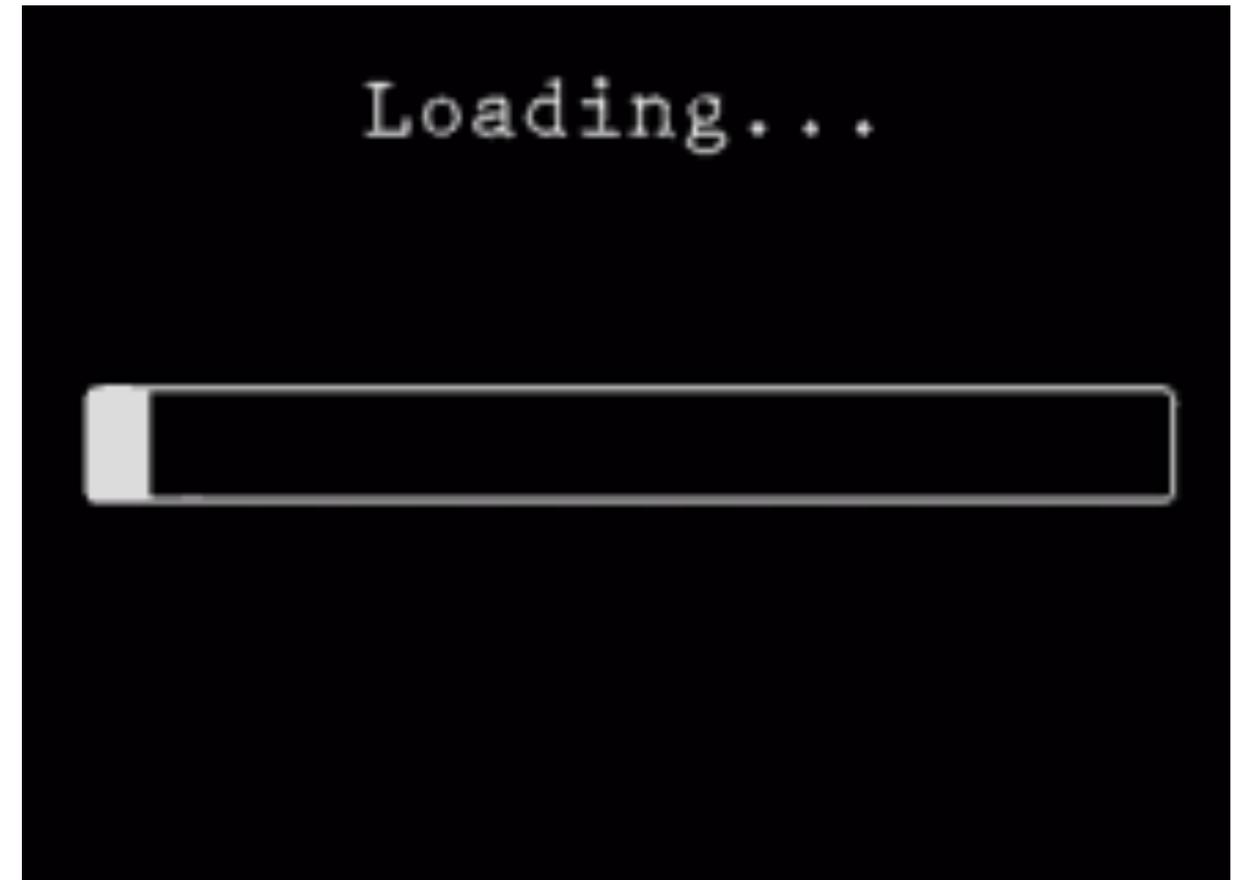


Better: controls placed in the same orientation as burners

# Provide feedback



Bad: when will it be done??



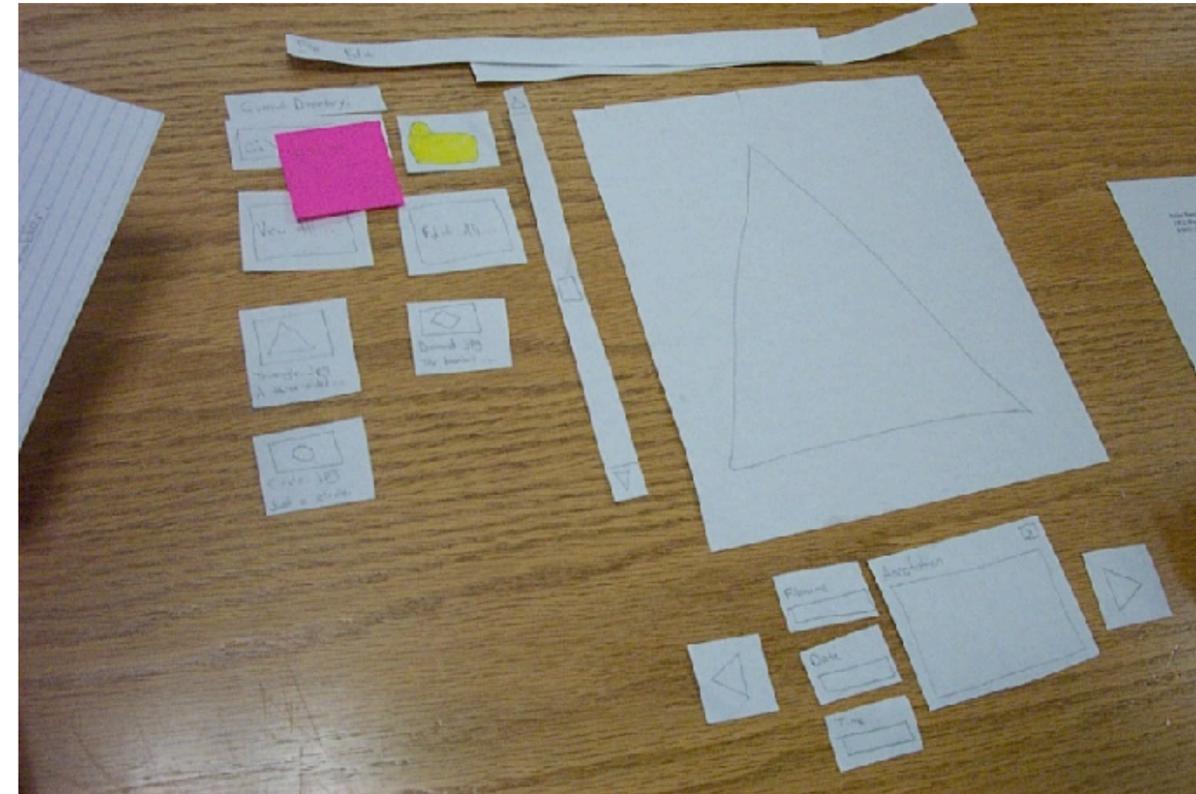
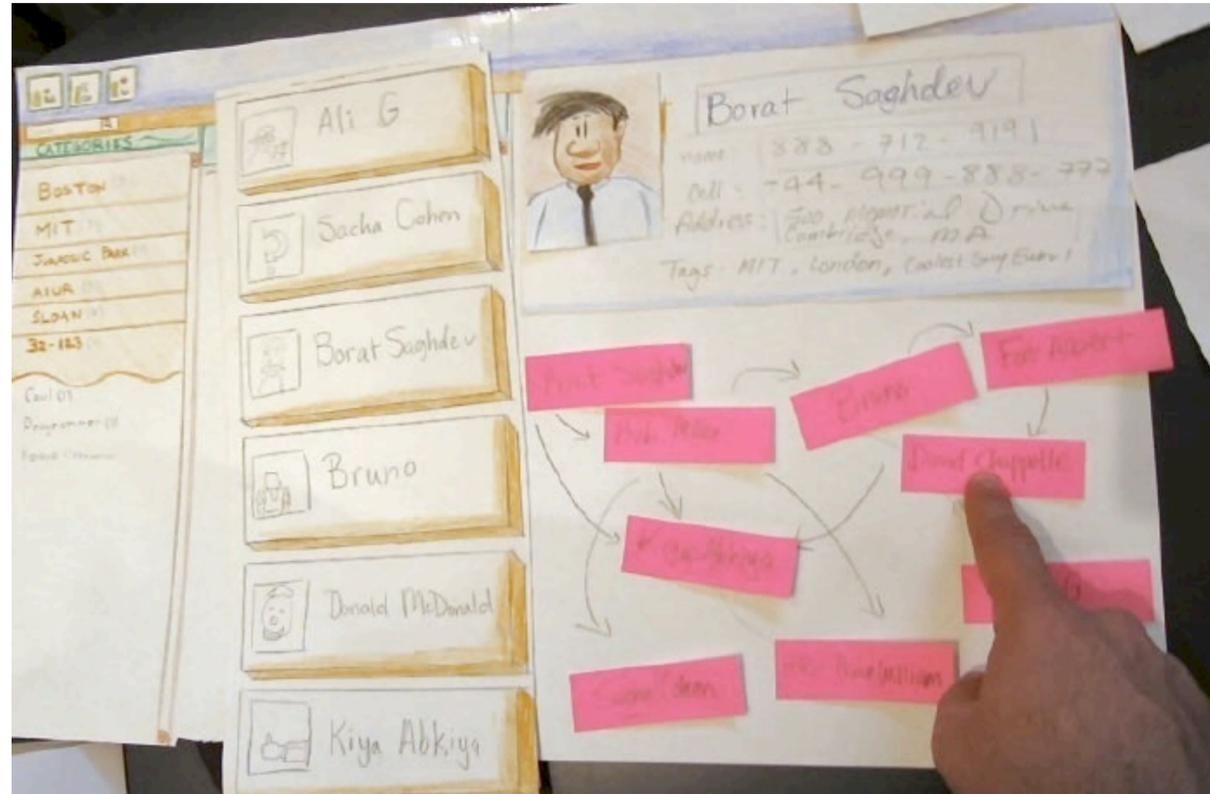
Better: some relative information

# Summary

- Prototypes force you to make design decisions quickly and cheaply. “Fail fast, fail often” before you sink in engineering time
- Much of “good” design stems from matching your and the user’s mental models. To do so,
  - **Make affordances** and controls **visible** but not over-exposed
  - Provide clear **mappings** between controls (inputs) and outputs
  - Provide clear and timely **feedback** in response to user input

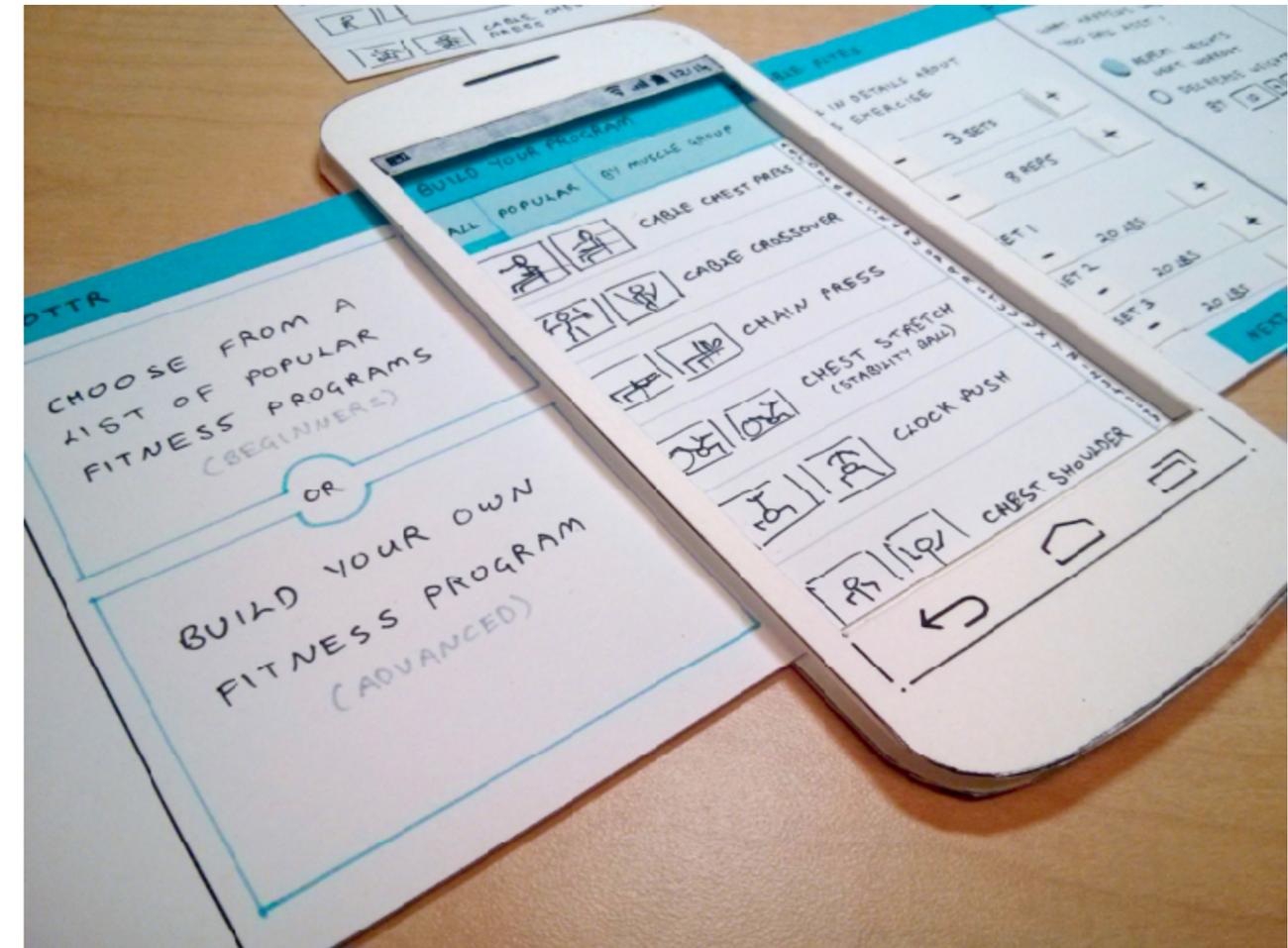
**Your first paper prototype**

# Paper is easier than code



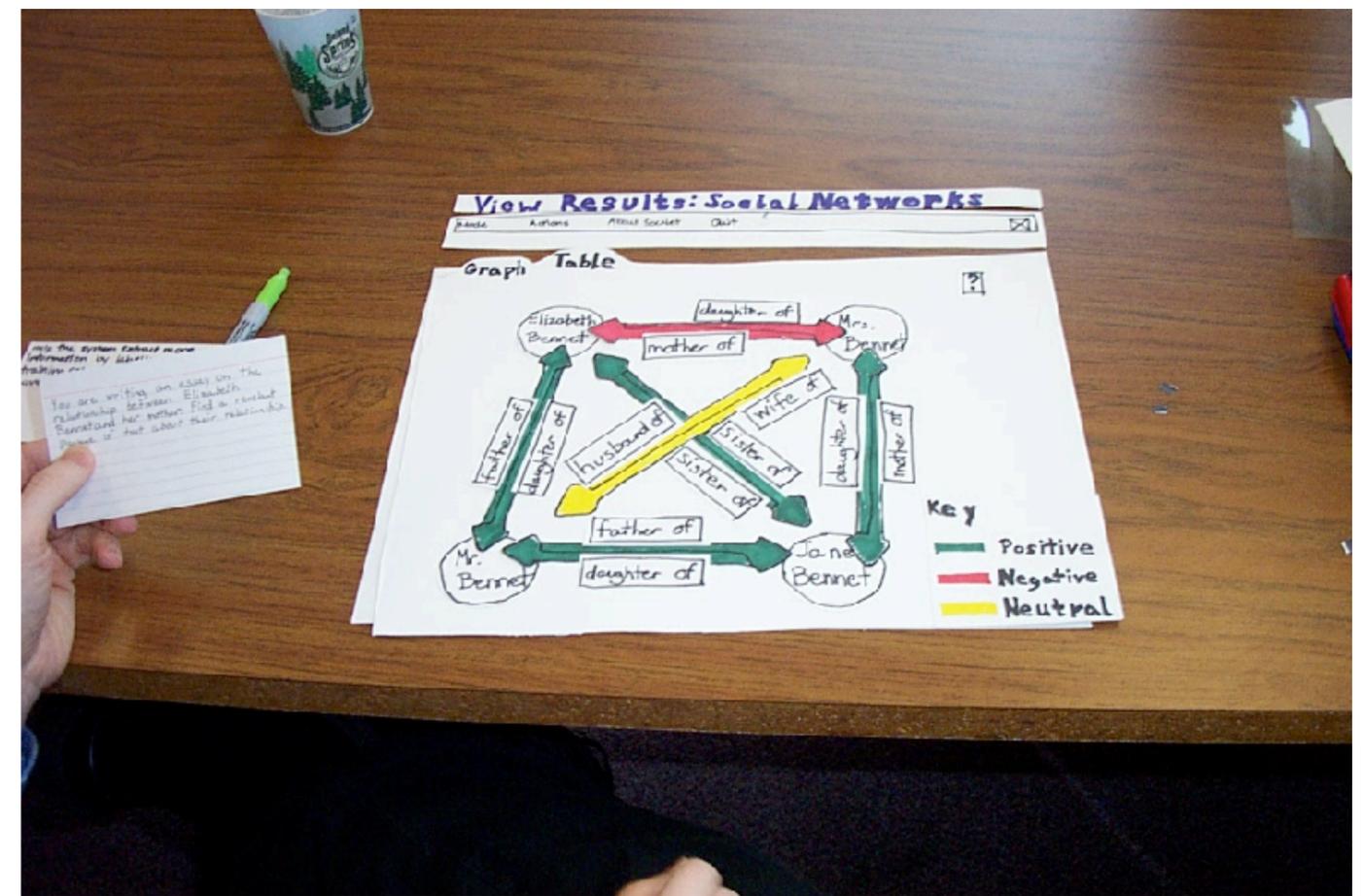
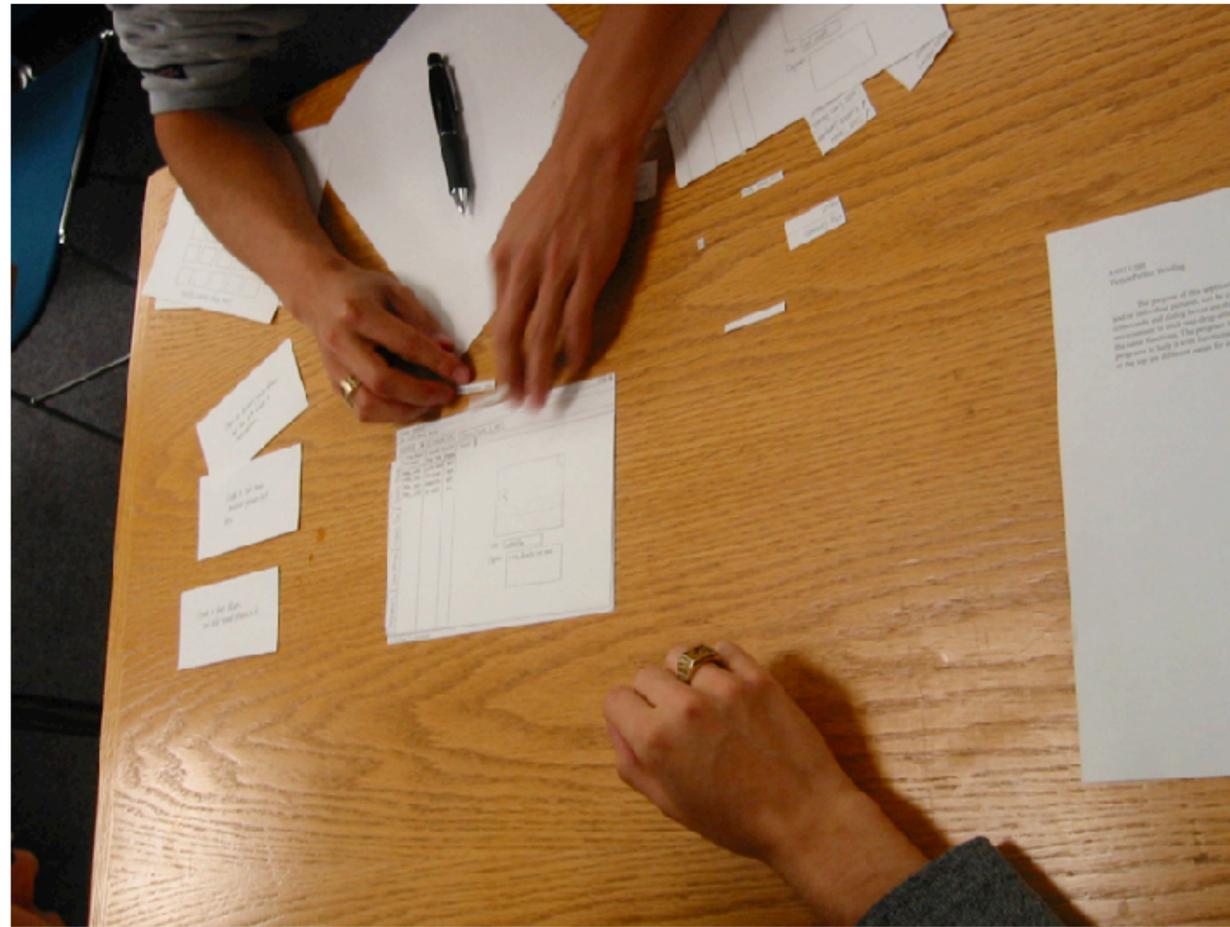
- Making with paper is faster and cheaper than implementing with code
- Use post-its or cut out smaller sheets of paper to replace UI elements

# Other ways of interactivity



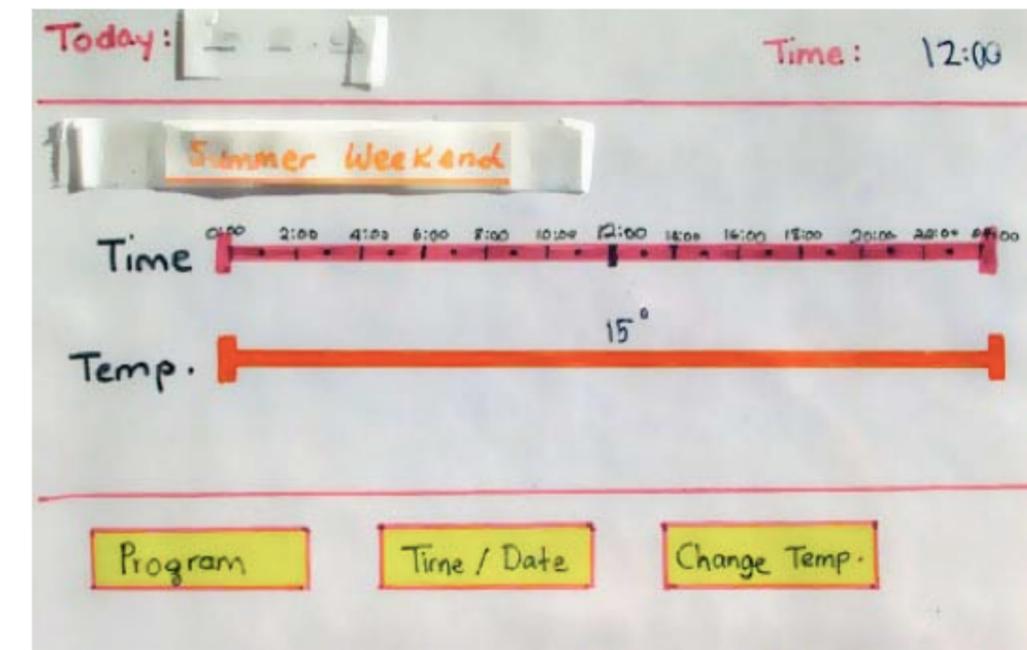
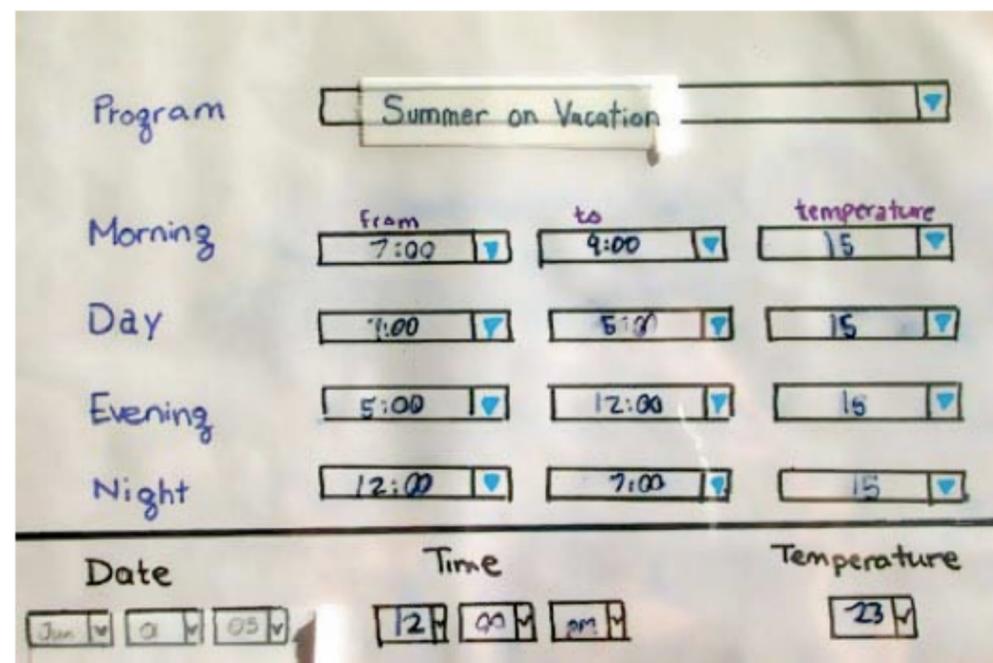
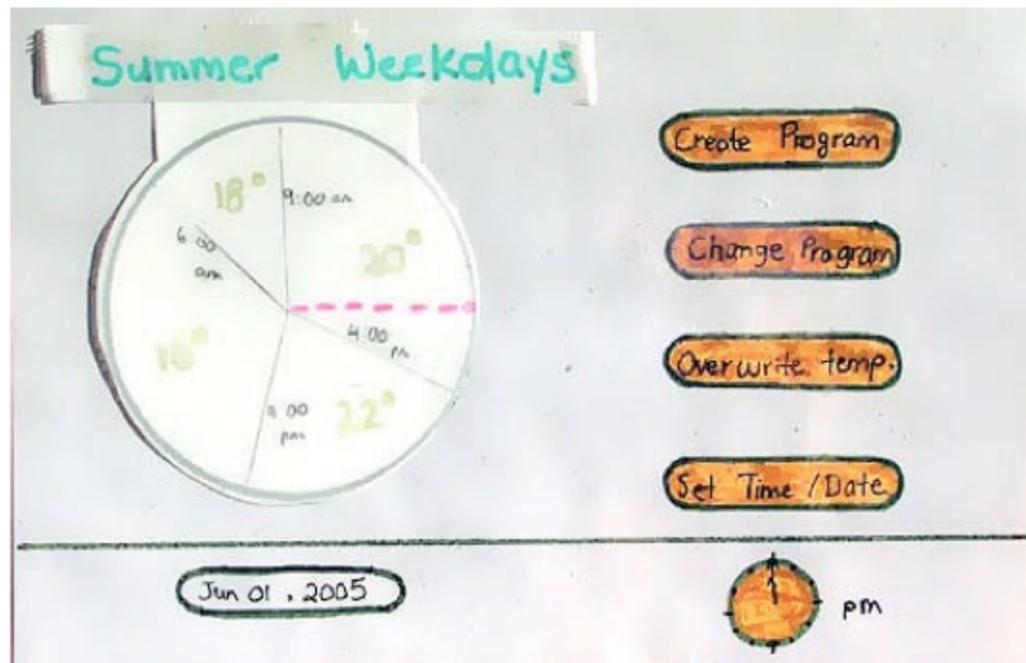
- Make a “frame” (like a phone or laptop screen) and replace screen content inside

# Size matters



- Bigger prototypes are easier to manipulate and switch out screens
- If you're making a web app, aim for a full 8.5x11 sheet of paper (a laptop screen size)
- Write big and dark: markers are better than pencil

# Multiple alternatives generate better feedback



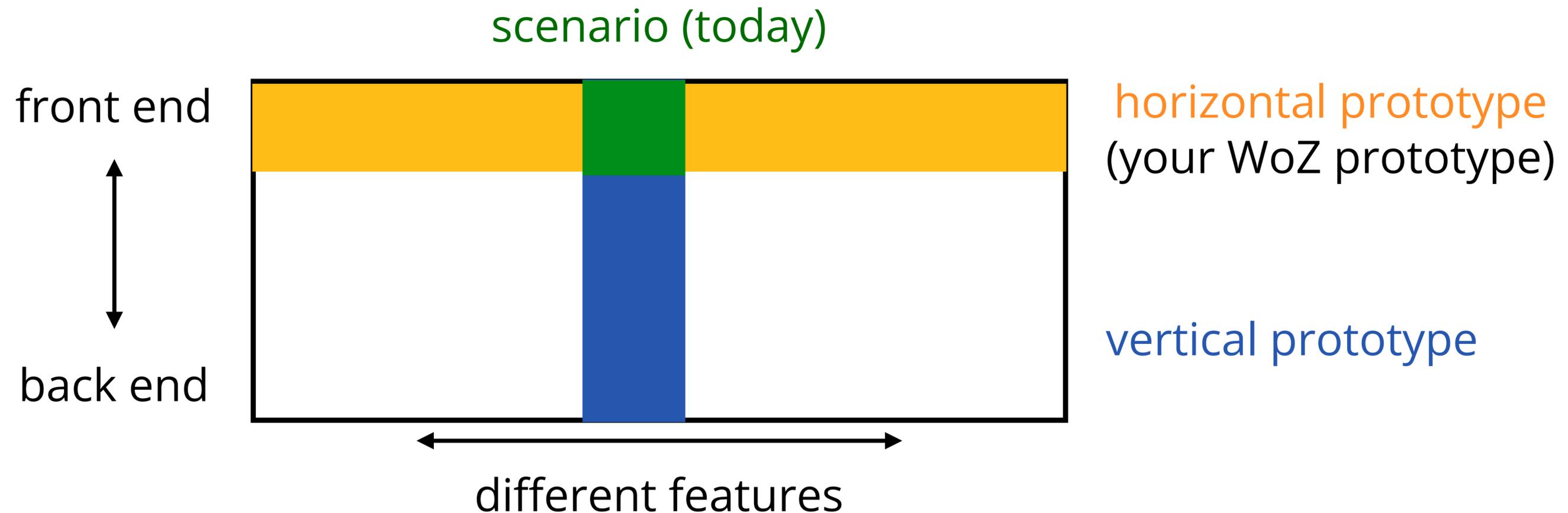
- When a design is presented with others, people tend to be **more ready to criticize** and offer problems, which is exactly what you want in the early stages of design.

# What can and can't you learn?

- **Functionality:** Does your tool do what it needs to do? Is it missing features?
  - **Conceptual model:** Do users understand it?
  - **Navigation & task flow:** Can users find their way around?
  - **Terminology:** Do users understand symbols or how to interact with UI elements?
  - **Screen content:** Are you displaying the right information at the right time?
- Can't learn...
    - Look & feel
    - Response time
    - Exploration << deliberation: users don't explore as much with paper prototypes

# Step 1: Pick your scenario

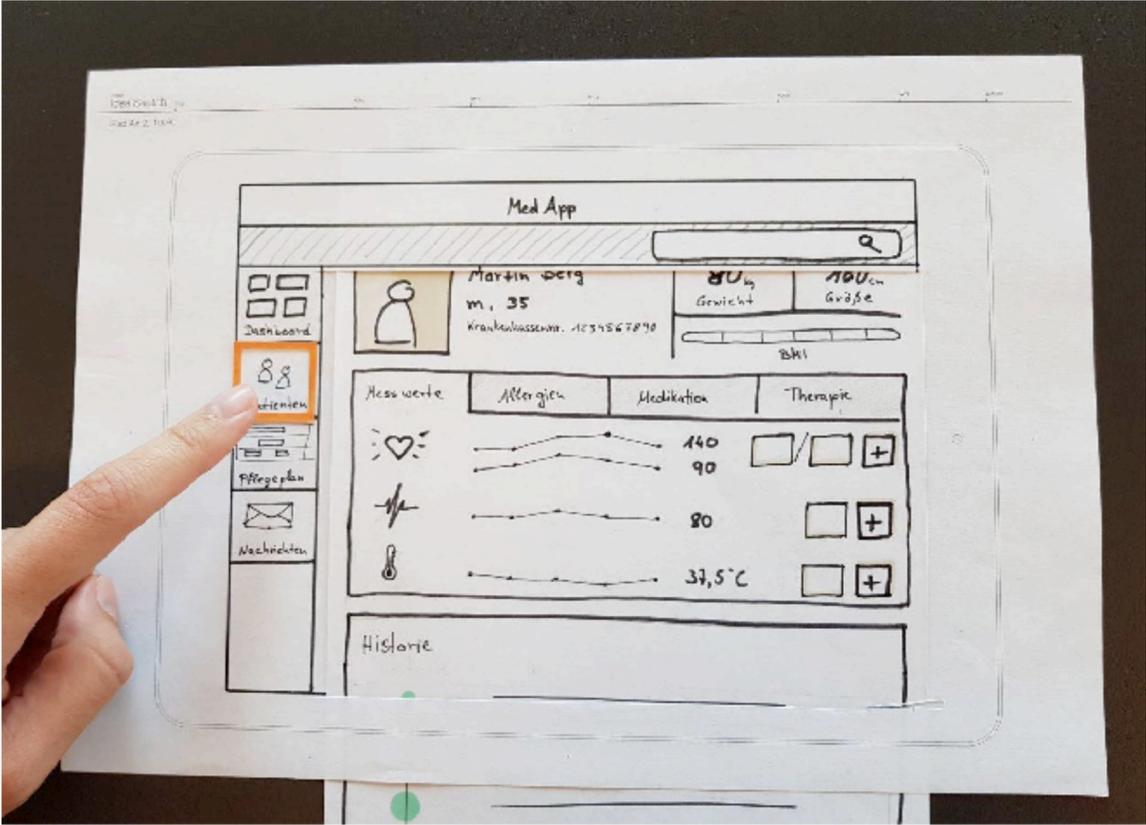
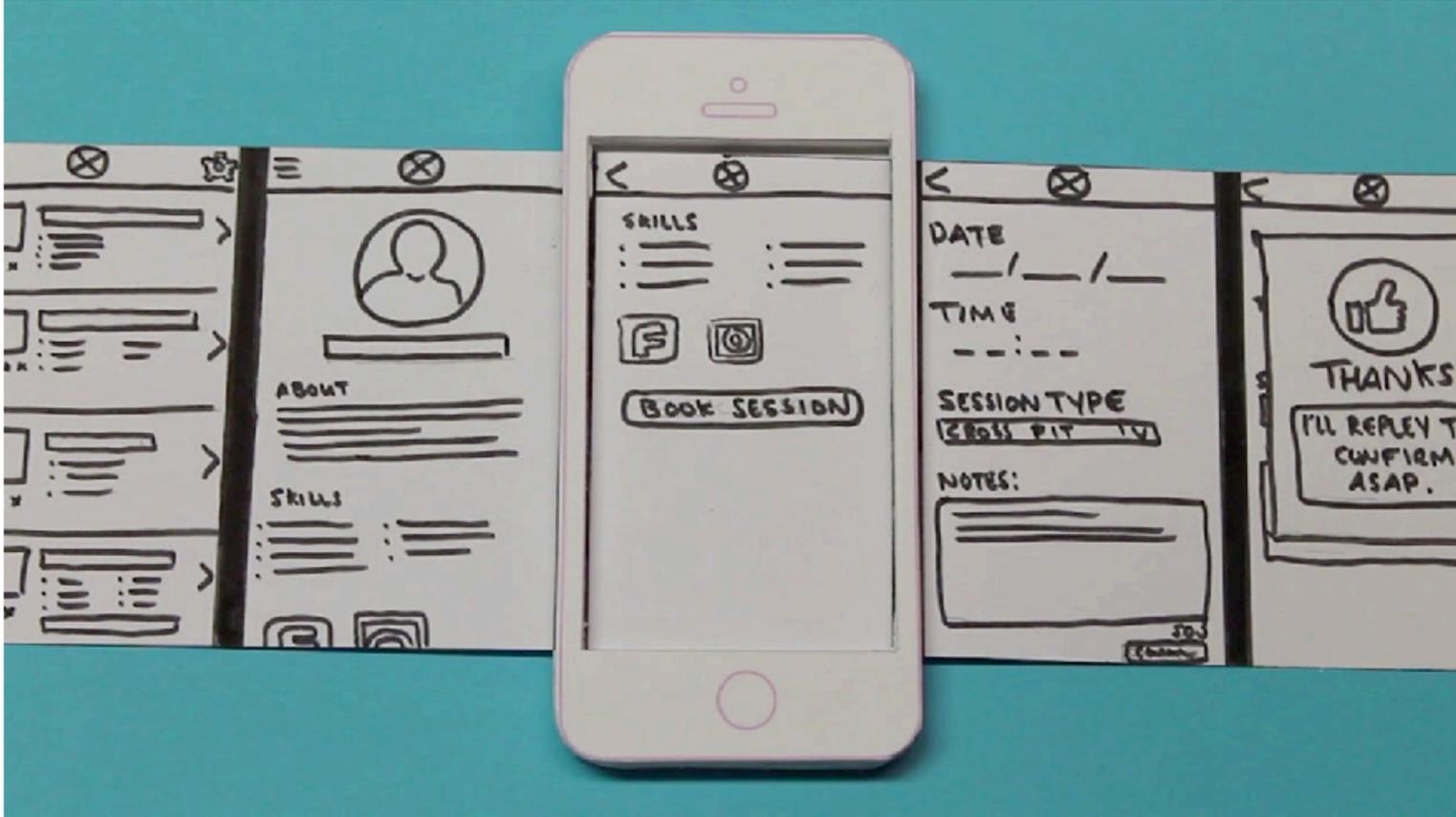
- 2 min: As a group, decide what the most important scenario for your tool is (what do users want to do, and how will your tool accomplish it?)



# Step 2: Everyone individually make a paper prototype

- Everyone should make their own first stab at a wireframe to accomplish this task scenario (more ideas generates better feedback!)
  - What elements are on the screen? How does the user interact with them?  
**(visible affordances)**
  - What options are available to the user? What parameters are important?  
**(mappings)**
  - How does interacting with the elements change the UI? **(feedback/flow)**
  - How does the user know when the scenario is done? **(feedback)**
- Your next assignment (after needfinding interviews) is to flesh out these prototypes

# Note: A prototype is not a sketch



# Midsemester feedback

- <https://forms.gle/bWJYu6c7AcKtvDhX9>



# Class 13 recap

- TODOs:
  - **Weds**
    - ZC from Ben
    - Needfinding milestone due 11am before class
    - Next milestone (task analysis & video prototype) released
  - **Week after spring break**
    - I will be away at a conference in Luxembourg; your TAs will lead class again!