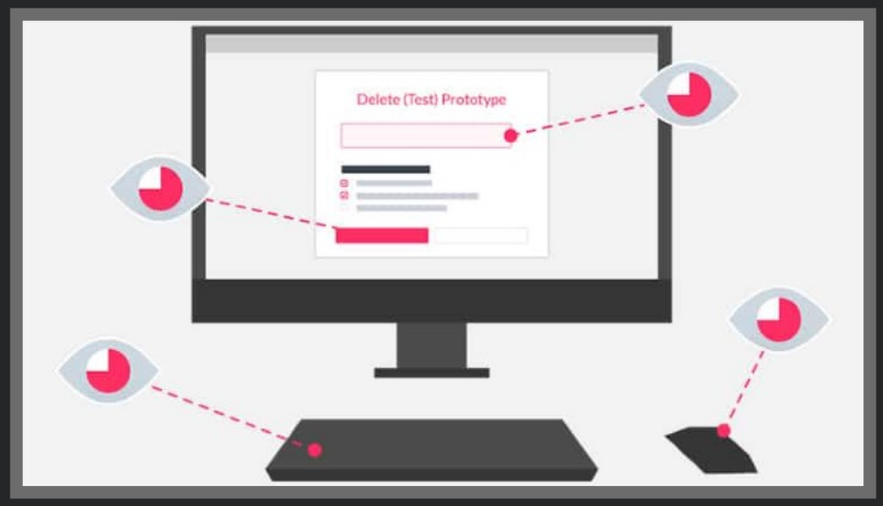
# SWE 432 -Web Application Development

Spring 2023

George Mason University

Dr. Kevin Moran

# *Week 12:* User Centered Design



# Administrivia

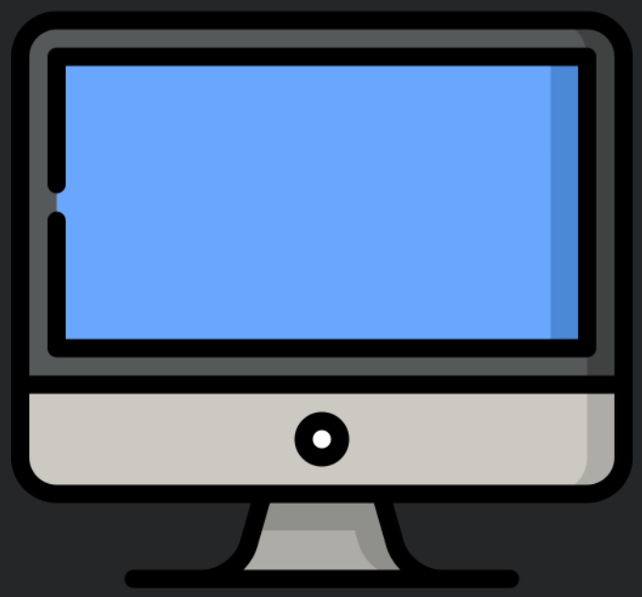
- *Midterm Exam Grades -* Sorry for delay, these will be posted tonight!
- *HW Assignment 3* - Due Thursday at midnight.
- *HW Assignment 4 -* Assignment out Thursday, due in two weeks (April 27th) ***at midnight***.
  - Extra Credit Opportunity!

# Class Overview

- ***Part 1:*** Introduction to User-centered design
  - Quick Lecture
  - Heuristic Evaluation Activity
- *Part 2:* Sketching and Prototyping
  - Quick Lecture
  - Hands-on with Heuristic Evaluation and a Prototyping Tool

# User-Centered Design

# Web Apps are Ubiquitous



“Good Design” is incredibly important

"Good Design" is incredibly important

… and is centered on ***usability***

# What is Usability?

Ease of Use

Productivity

Learnability

Efficiency

Retainability

User Satisfaction

Effectiveness

# Usable or Unusable?

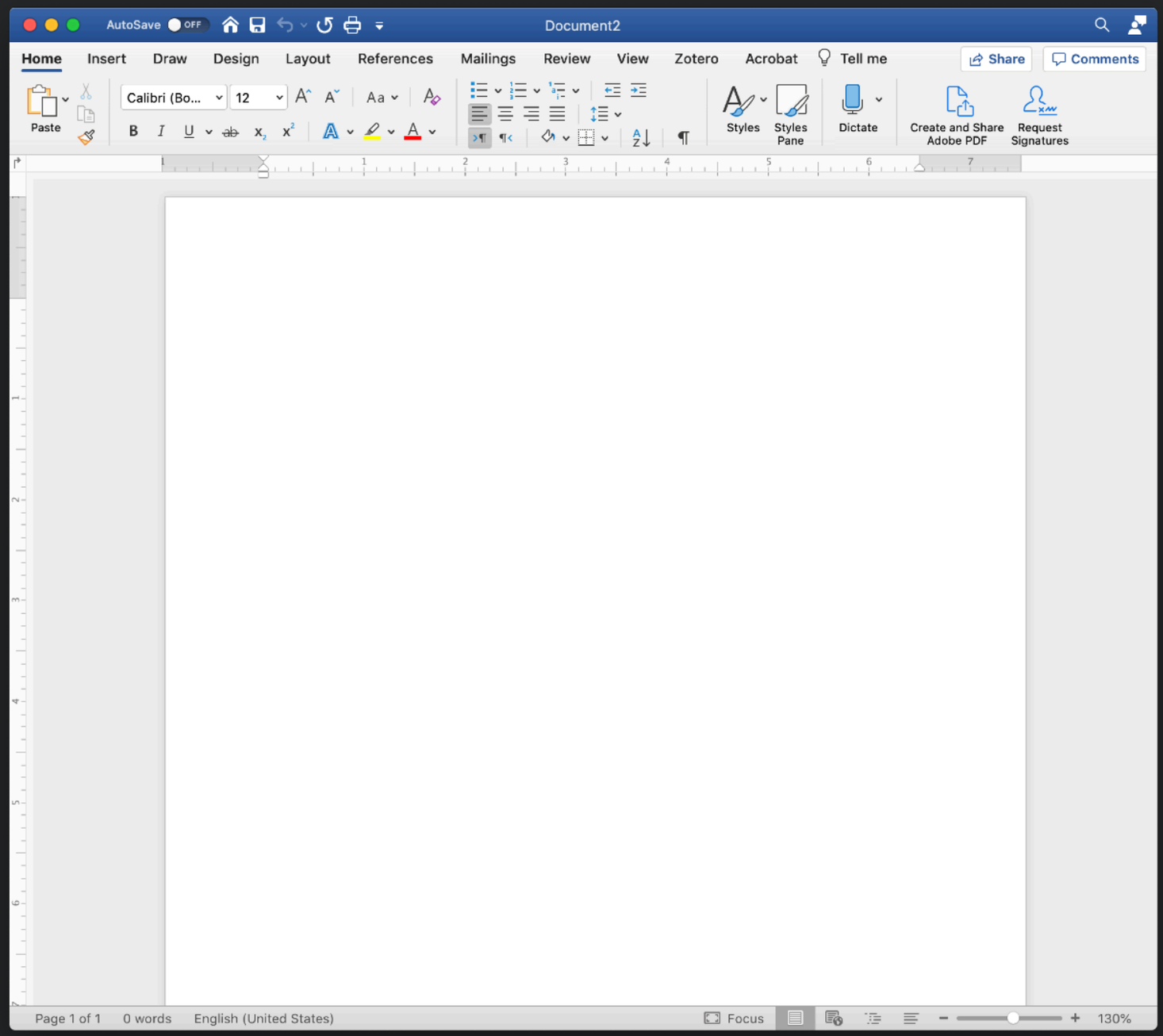
A Teapot

# Usable or Unusable?

A Door

# Usable or Unusable?

A Word Processor



# Usability

- A property of the relationship between
  - humans with goal-driven tasks
    - an artifact
- The speed and success with which the goals can be accomplished (task ***performance***)

# Needfinding

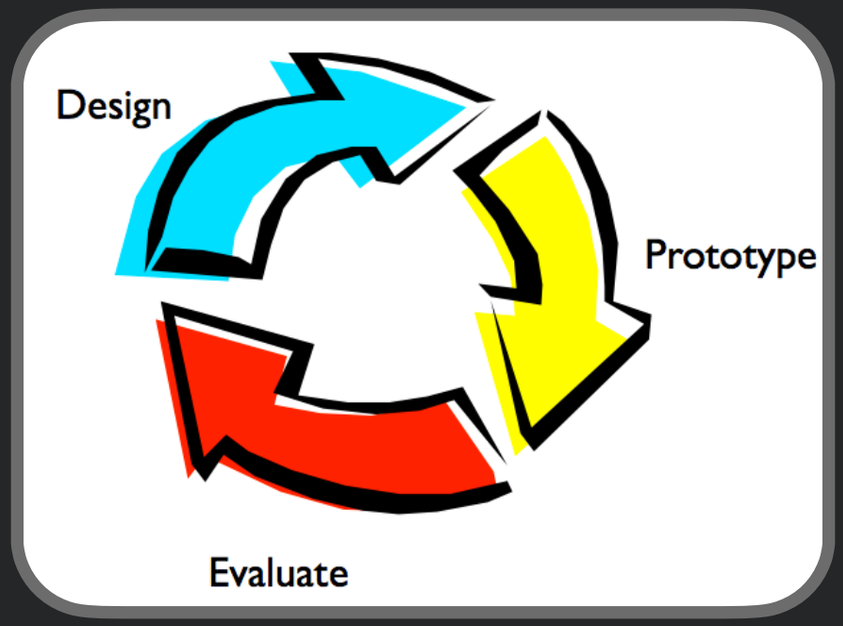
- Given an existing artifact and humans doing a set of tasks, determine ***goals*** and identify ***usability issues*** that decrease task performance

# User-Centered Design

- Given humans with goals and tasks, design an ***artifact*** that helps to accomplish these tasks

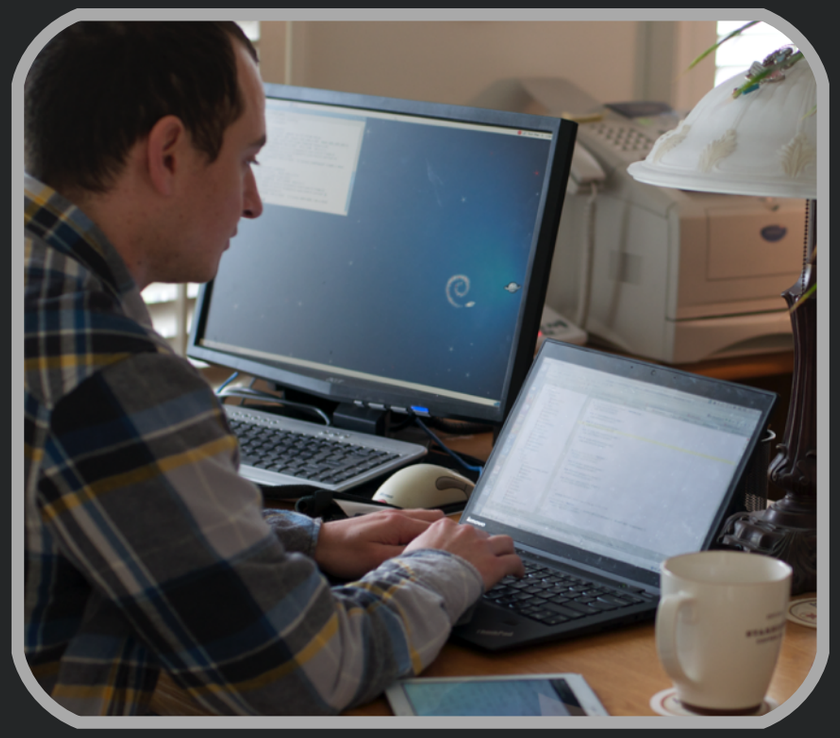
# **Iterative** User-Centered Design

- Given humans with goals and tasks, ***re***design an ***existing*** artifact that helps to accomplish these tasks ***faster*** and more ***successfully***



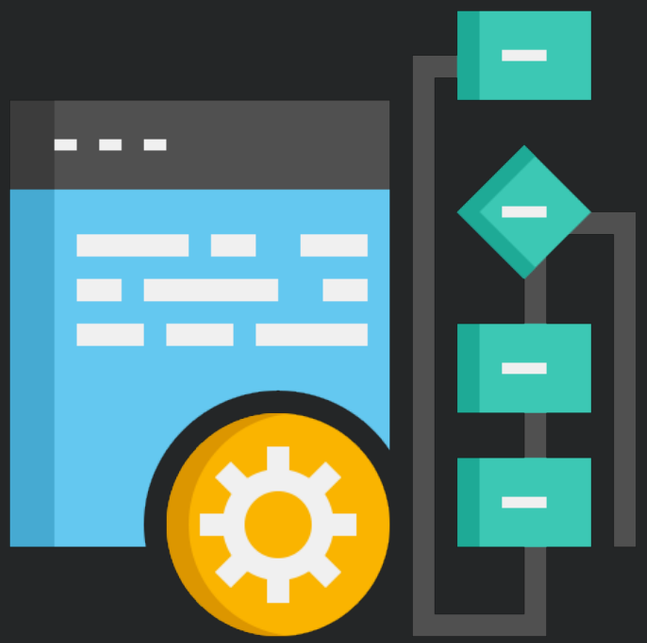
# **Empirical:** Usability **Study**

- Given humans with goals and tasks an artifact, ***observe humans*** to identify usability issues that decrease task performance



- ***"Ground Truth"***

# **Analytical:** Usability **Principles**



- Given humans with goals and tasks and an artifact, ***<u>assess for conformance to UI principles</u>*** to identify usability issues that decrease task performance

- ***Approximation of "ground truth"***

# Why Study Usability?

*"The results show that in today's applications, an average of **48% of the code** is devoted to the user interface portion."*

*"The average time spent on the user interface portion is 45% during the design phase, 50% during the implementation phase, and 37% during the maintenance phase."*

– Myers & Rosson, CHI'92

# Why Study Usability?

## Life-Threatening Errors

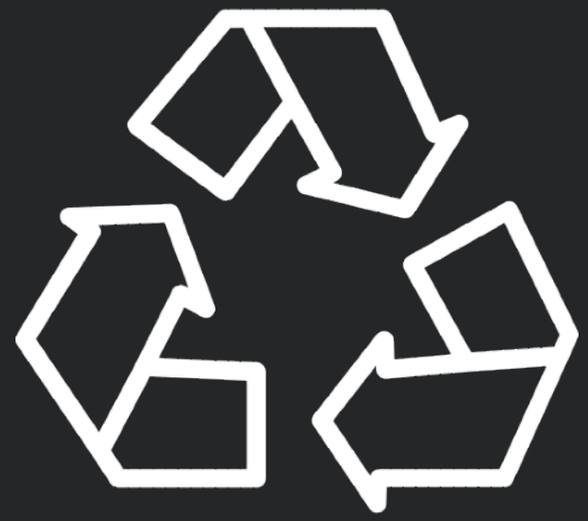
- 1995 American Airlines jet crashed into canyon wall, killing all aboard
- On approach to Rozo airport in Colombia
- Pilot skipped some of the approach procedures
- Pilot typed in “R” and system completed full name of airport to Romeo
- Guidance system executed turn at low altitude to head for Romeo airport
- 9 seconds later plane struck canyon wall
- Is the pilot to blame?
- *http://en.wikipedia.org/wiki/American_Airlines_Flight_965*

# Iterative Model of User-Centered Design

***Observation***

(Re)Define the Problem

Understand User Needs



***Idea Generation***

Brainstorm
what to build

***Prototype/ Implementation***

Build

***Test***

Evaluate what
you have built

# Heuristic Evaluation

# Heuristic Evaluation (Analytical)

- "***Discount*** usability engineering methods" - Jakob Nielsen

- Involves a small team of evaluators to evaluate an interface based on recognized usability principles

- Heuristics – "rules of thumb"

Adapted from slides by Bonnie John and Jennifer Mankoff

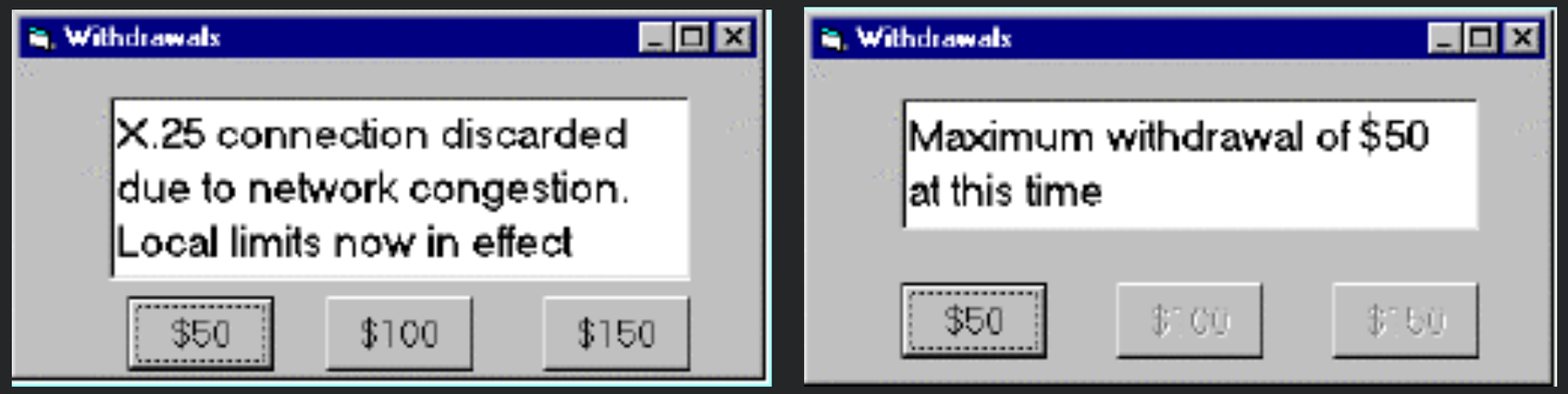
# Heuristic Evaluation

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 vs. 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

# **H1:** Visibility of System Status


- ***What input has been received*** - Does the interface above say what the search input was?
- ***What processing it is currently doing*** - Does it say what it is currently doing?
- ***What the results of processing are*** - Does it give descriptive results?
- Feedback allows user to monitor progress towards solution of their task, allows the closure of tasks and reduces user anxiety (*Lavery et al*)

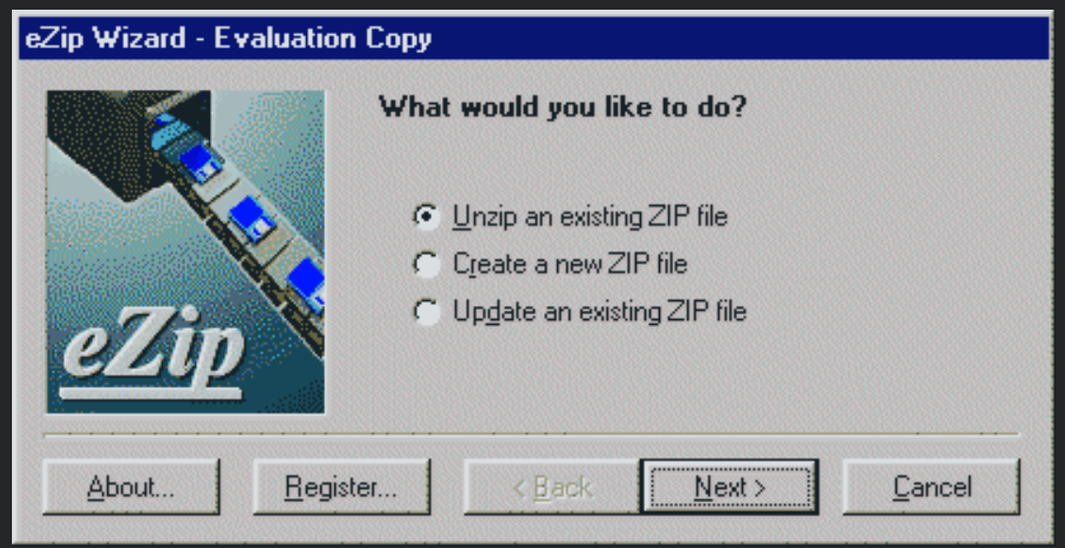
# H2: Match Between System & Real World




- Speak the users' language
- Follow real world conventions

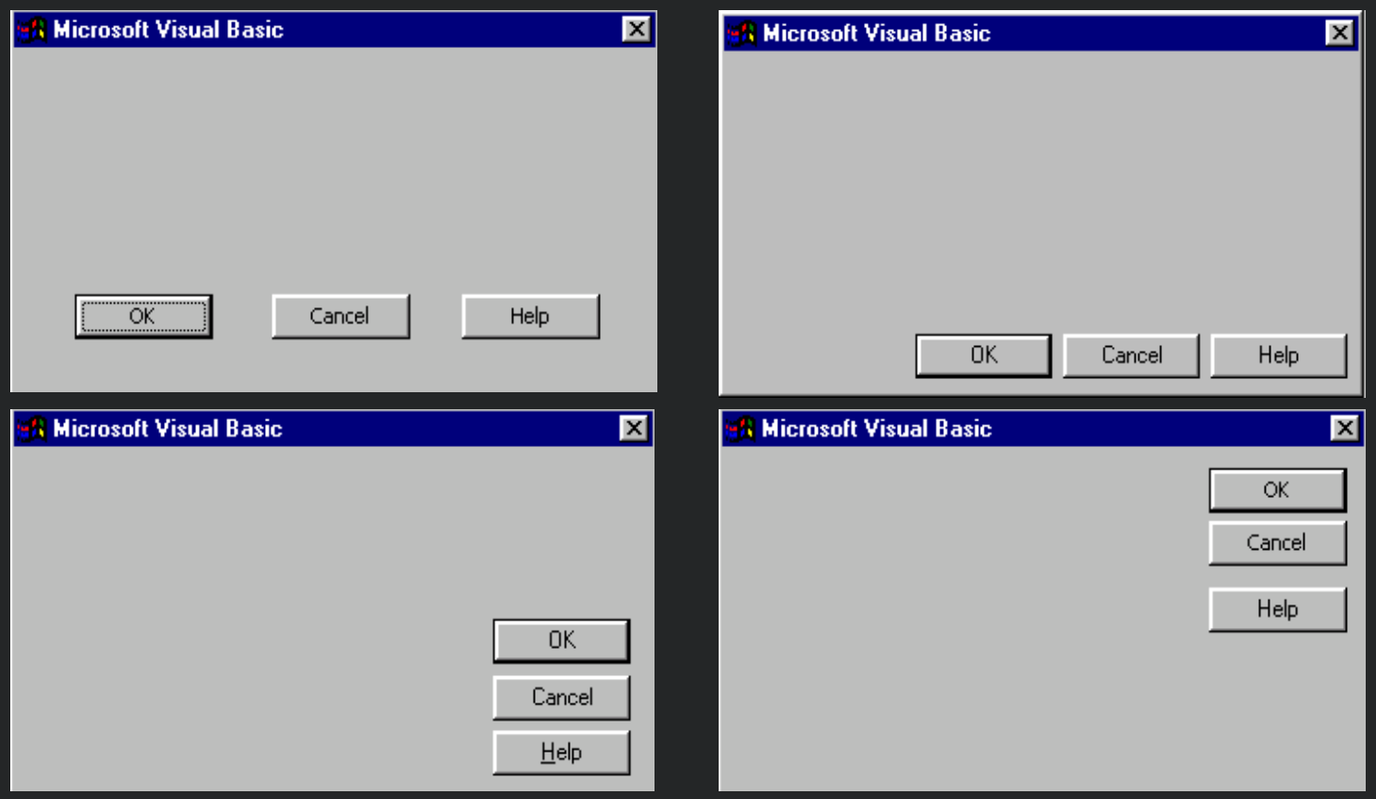
# H2: Match Between System & Real World


# H3: User Control & Freedom




- “Exits” for mistaken choices, undo, redo
- Don’t force down fixed paths

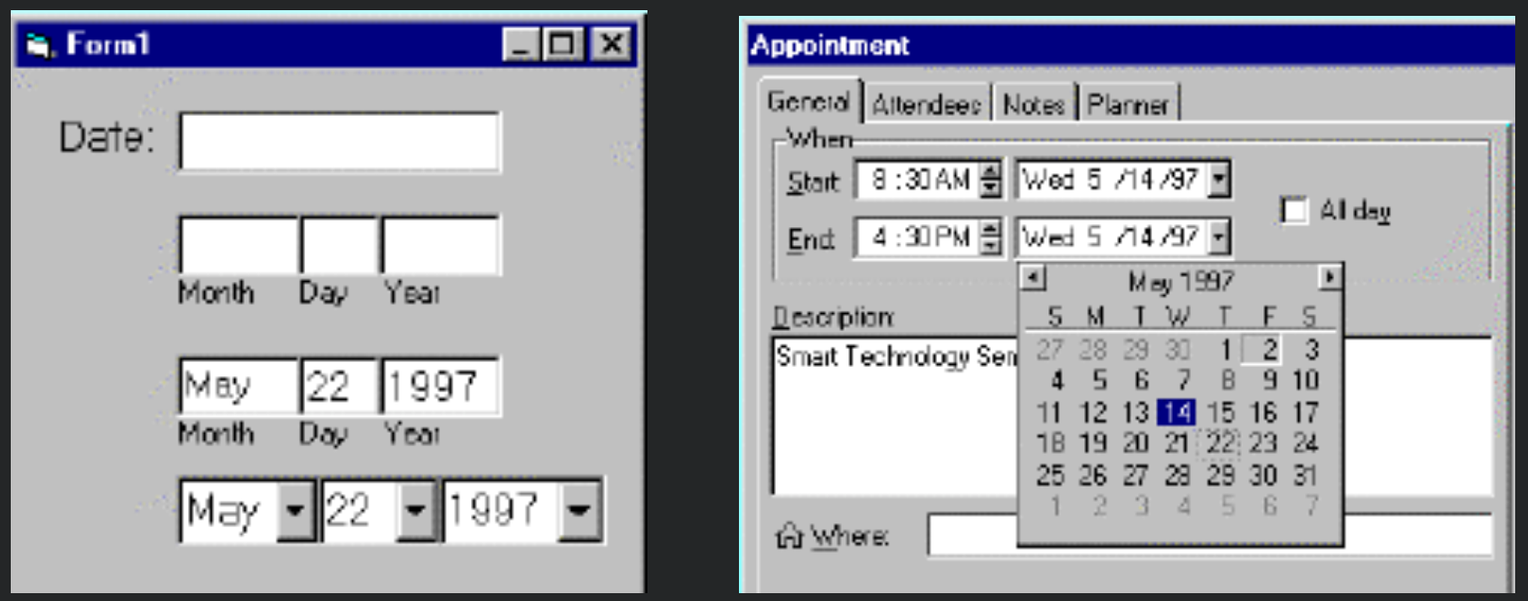
# H4: Consistency & Standards




- Same words, situations, actions, should mean the same thing in ***similar*** situations; same things look the same, be located in the same place.
- Different things should be different

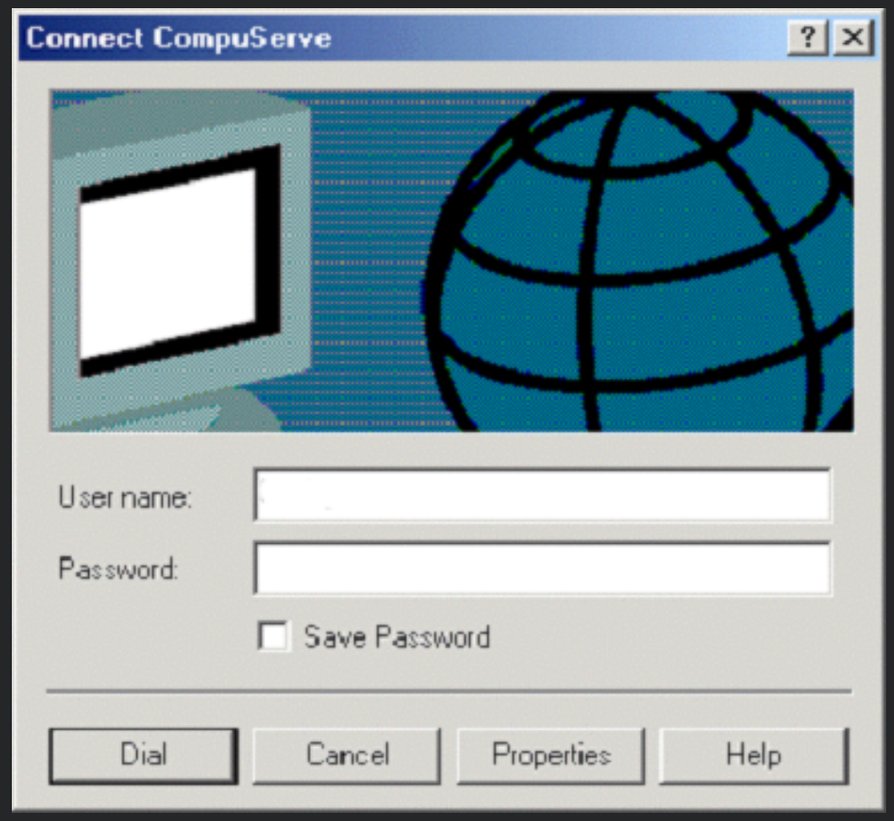
# H4: Consistency & Standards

# H5: Error Prevention



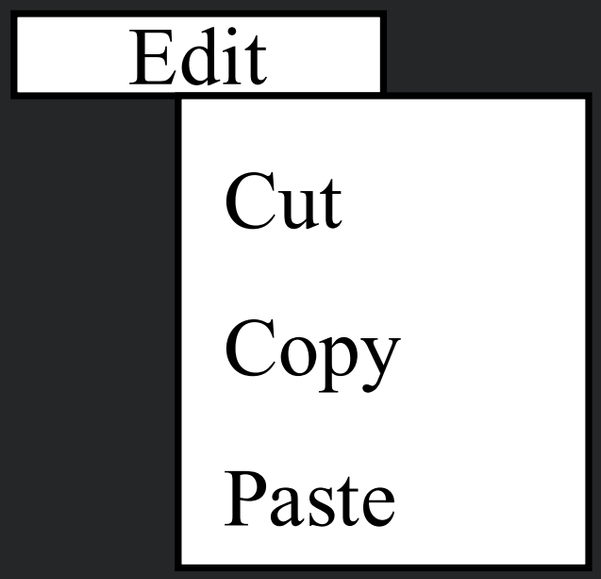

- Careful design which prevents a problem from occurring in the first place

# **H6:** Recognition Not Recall



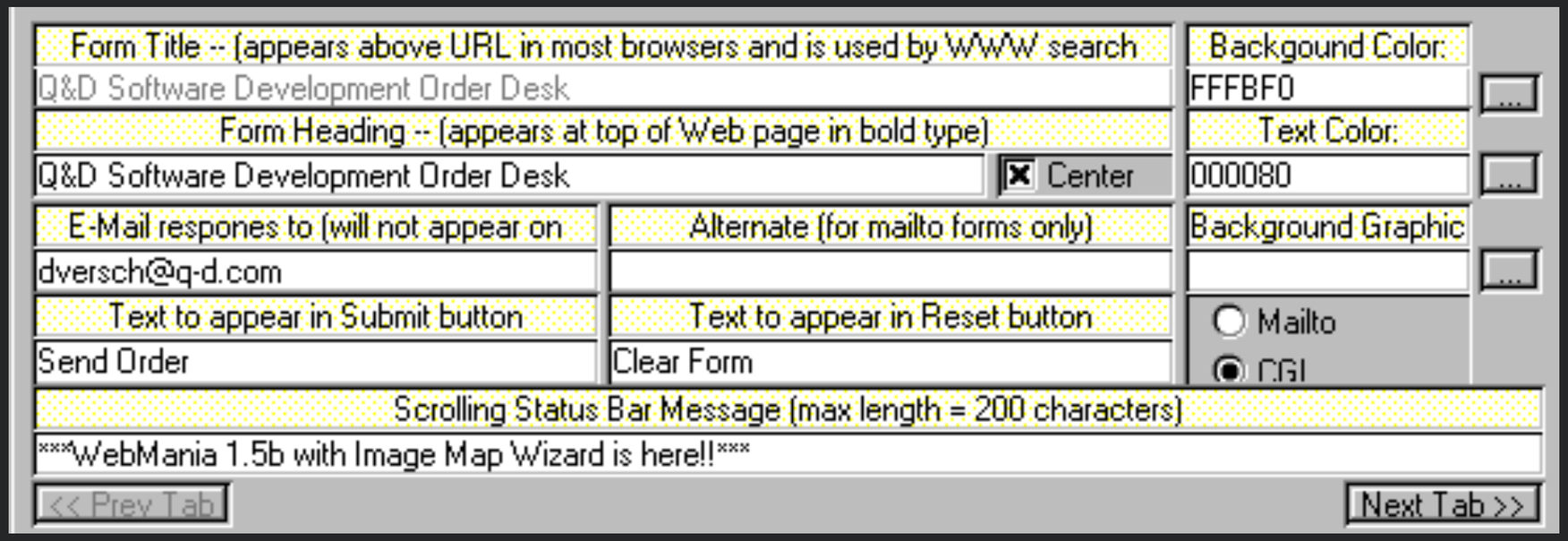

- Make objects, actions and options visible or easily retrievable

# H7: Flexibility & Efficiency of Use




- Accelerators for experts (e.g., gestures, kb shortcuts)
- Allow users to tailor frequent actions (e.g., macros)

# H8: Aesthetic & Minimalist Design



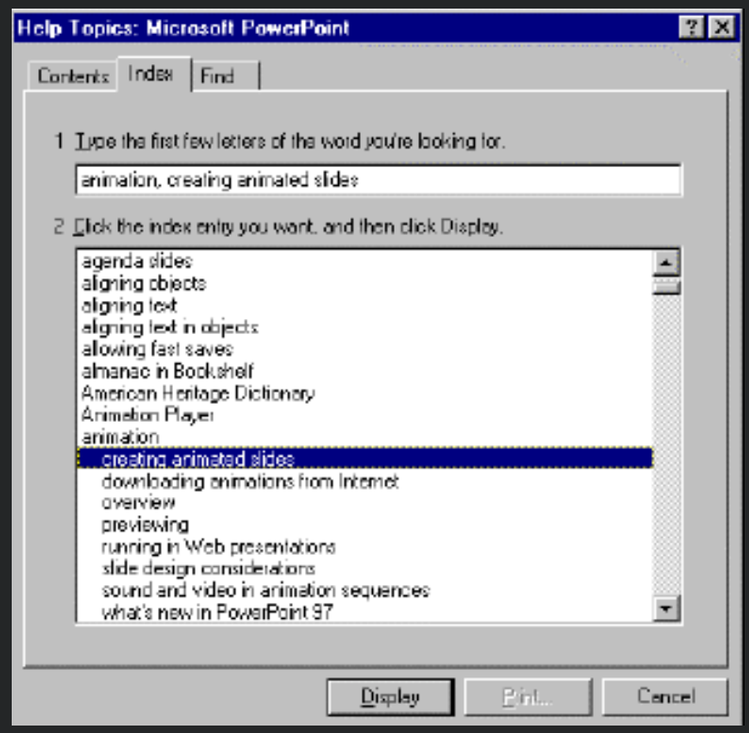

- Interfaces should not contain irrelevant or rarely needed information

# H9: RDR from Errors

Help Users *Recognize*, *Diagnose*, and *Recover* from Errors


- Error messages in language user will understand
- Precisely indicate the problem
- Constructively suggest a solution

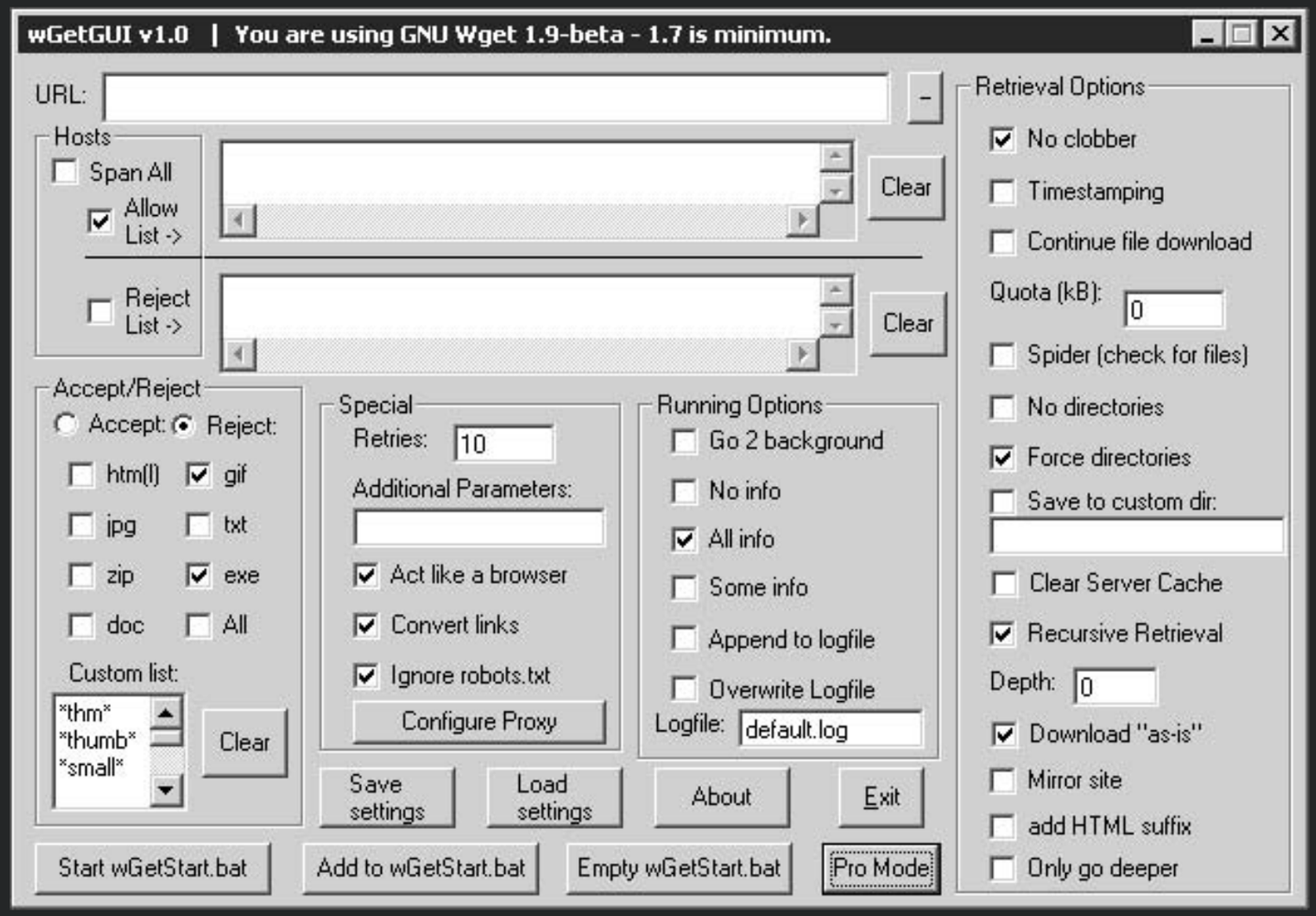
# H10: Help & Documentation




- Easy to search
- Focused on the user's task
- List concrete steps to carry out
- Always available

# Example

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 vs. 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



# Advantages of Heuristic Evaluation

- “Discount usability engineering” - Intimidation low
- Don’t need to identify tasks, activities
- Can identify some fairly obvious fixes
- Can expose problems user testing doesn’t expose
- Provides a language for justifying usability recommendations

# Disadvantages of Heuristic Evaluation

- Un-validated
- Do not employ real users
- Can be error prone
- Better to use usability experts
- Problems unconnected with tasks
- Heuristics may be hard to apply to new technology

# Using Heuristic Evaluation

- Can be used informally to identify issues in a website
- Can be used as a more formal usability inspection method
- Evaluators each first separately identify issues
- Issues then combined from each evaluator

# Ways to Use Heuristic Evaluation

- Early in design process to catch major issues
- When time or resources are not available for empirical usability evaluation

# SWE 432 - Web Application Development

George Mason University


Instructor:
Dr. Kevin Moran

Teaching Assistant:
Oyindamola Oluyemo


Class will start in:

10:00

# Iterative Model of User-Centered Design

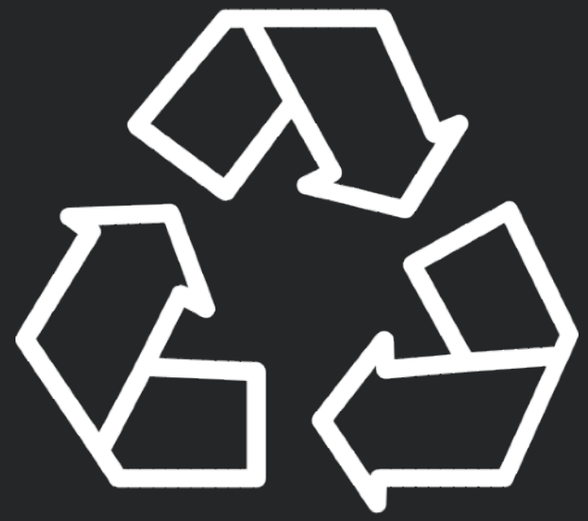
***Observation***

(Re)Define the Problem

Understand User Needs

***Idea Generation***

Brainstorm
what to build



***Prototype***

Build

***Test***

Evaluate what
you have built

# Iterative Model of User-Centered Design

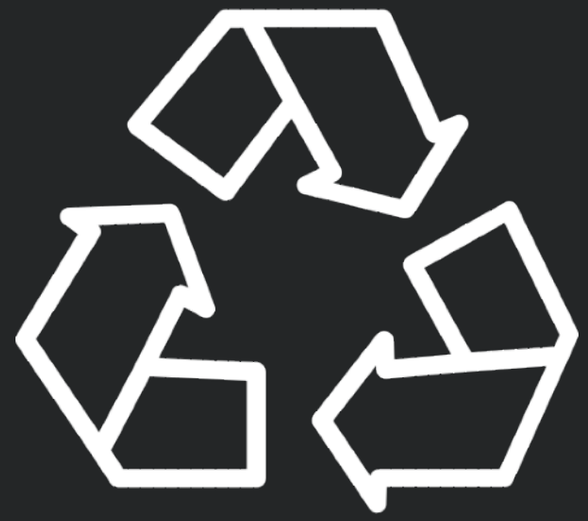
***Observation***

(Re)Define the Problem

Understand User Needs

***Idea Generation***

Brainstorm
what to build



***Prototype***

Build

***Test***

Evaluate what
you have built

# Sketching & Storyboards



Based on slides by Bonnie John and Jennifer Mankoff

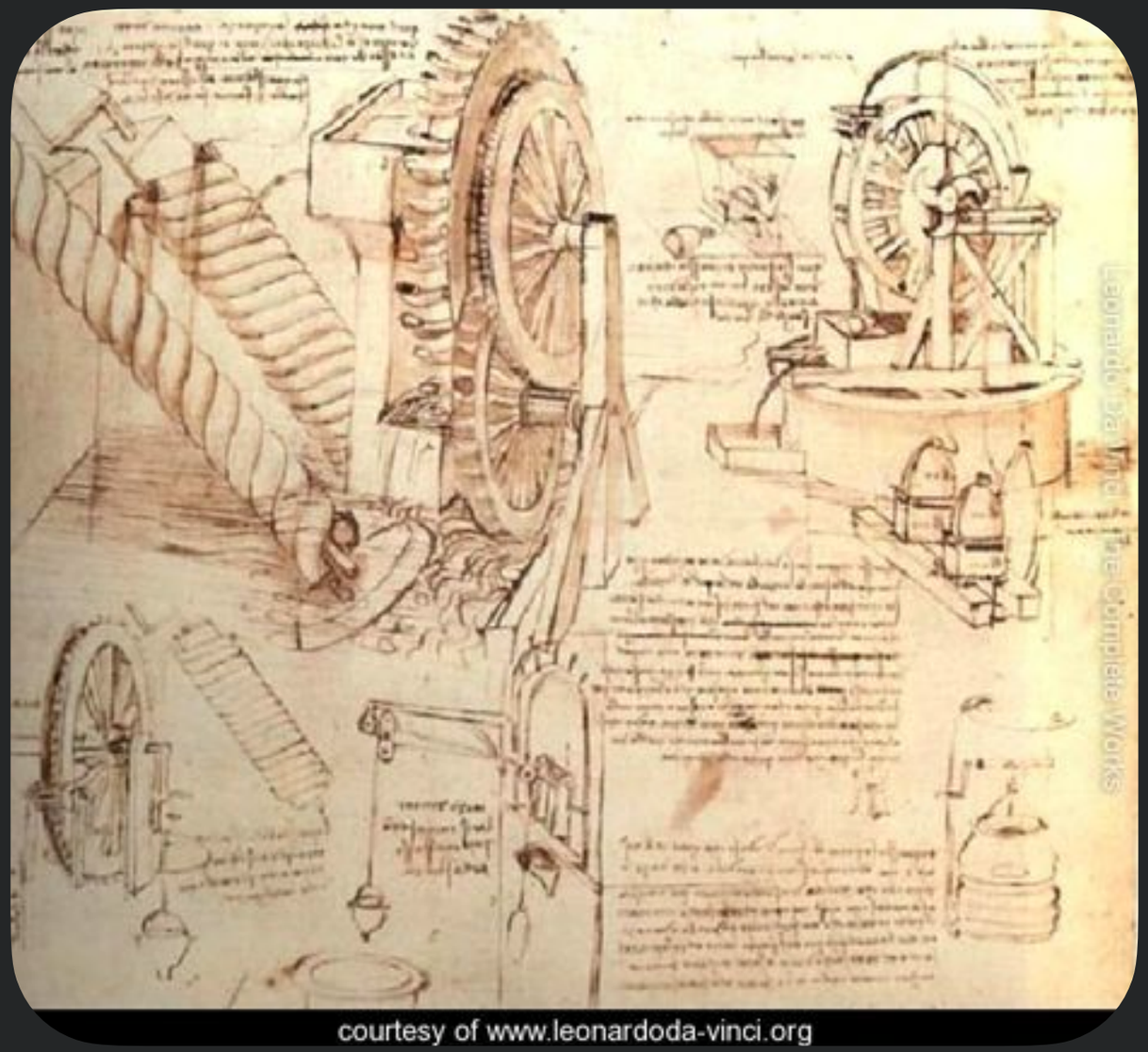
# How do You Brainstorm?

# What is a Sketch?

*"A conversation between the sketcher or designer and the artifact"*

# Why Sketch?

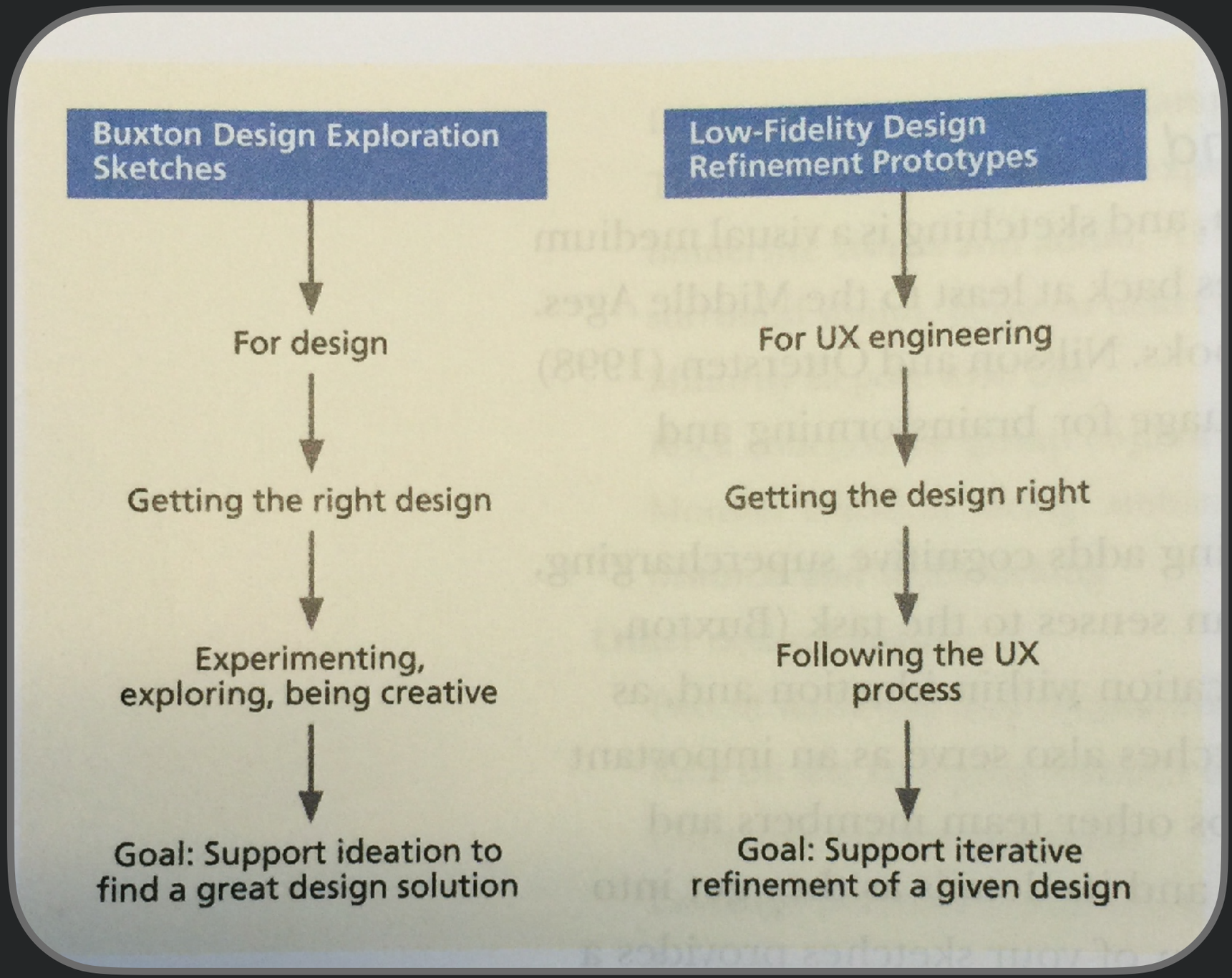
- Sketching offers ***visual*** medium for exploration, offering cognitive scaffolding to externalize cognition



# Being Creative with Sketches

- How do you come up with a great idea?
  - Generate lots of ideas
  - Work through ideas through externalization in sketch
  - Critique the ideas
  - Refine them to make them better
- Sketching offers a low-cost medium for working with early ideas ***before*** committing to one
- Design is process of creation & ***exploration***

# Sketching vs. Prototyping



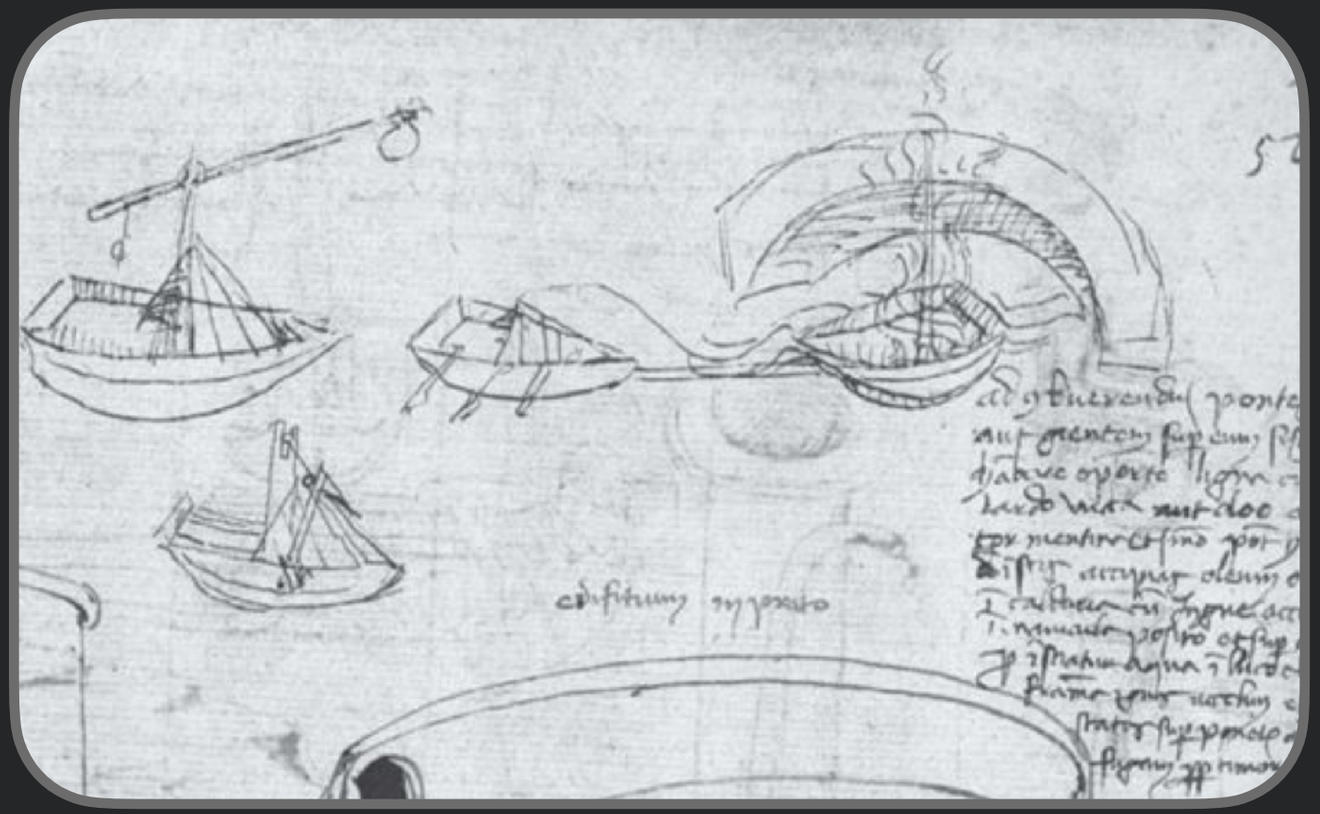
# Physical Sketches

- Production tools for sketching:
  - whiteboards, blackboards, cork boards, flip chart easels
  - post it notes
  - duct tape, scotch tape, push pins, staples
  - marking pens, crayons, spray paint
  - scissors, hobby knives, foam core board
  - duct tape
  - bits of cloth, rubber

# The Space Remembers

- Covering walls, whiteboards, etc. w/ materials is extremely useful
- Provides fast access for revisiting and remixing old ideas
- Facilitates group discussion of designs

# Sketches are Sketchy



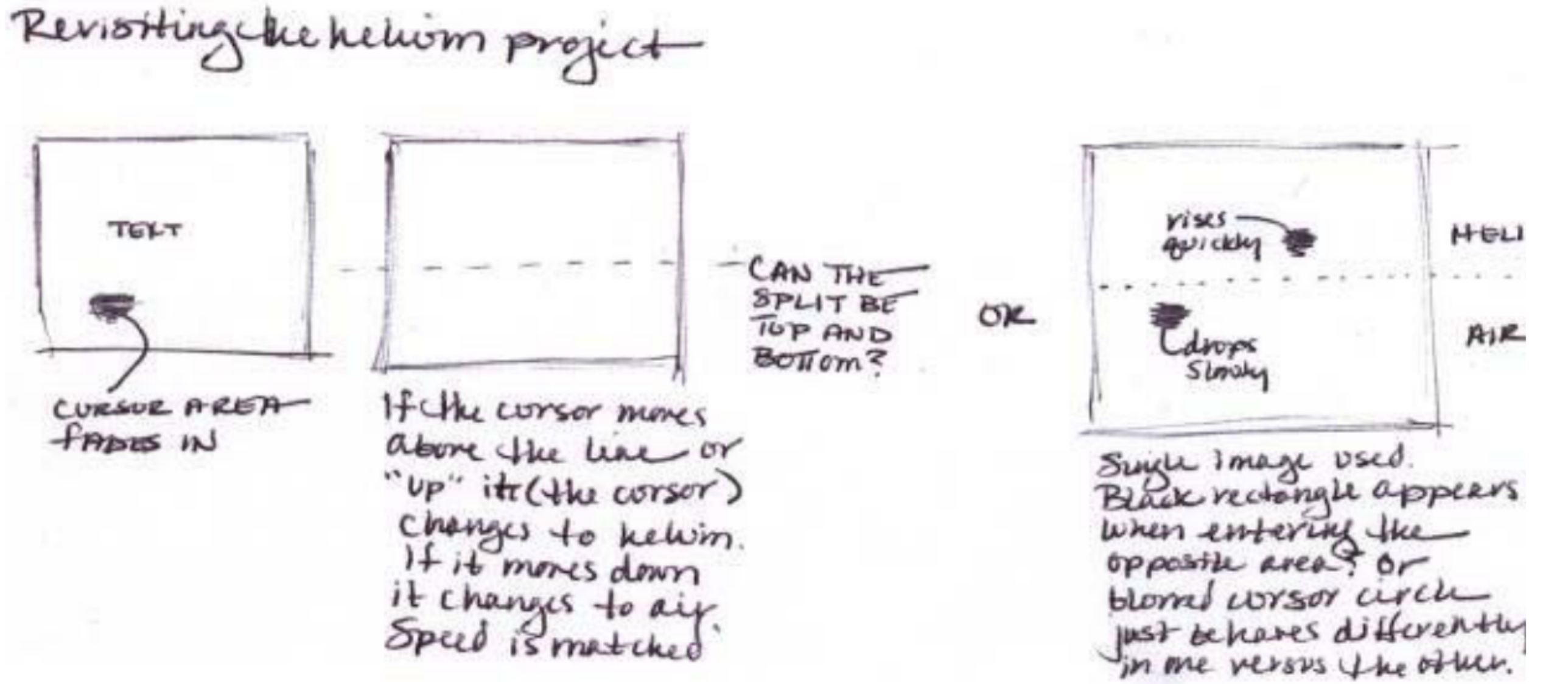
- Not mechanically correct and perfectly straight lines
- ***Freehand***, open gestures
- Strokes may miss connections
- Resolution & detail **low** enough to suggest is concept
- Deliberately ***ambiguous*** & abstract, leaving “holes” for imagination

# Rules for Sketching

- ***Everyone*** can sketch; you do not have to be artistic
- Most ideas conveyed more effectively with sketch than words.
- Sketches are ***quick*** and inexpensive to create; do not inhibit early exploration
- Sketches are ***disposable***; no investment in sketch itself
- Sketches are ***timely***; made in-the-moment, just-in-time
- Sketches are ***plentiful***; entertain large # of ideas w/ multiple sketches of each

# Sketches Include Annotations

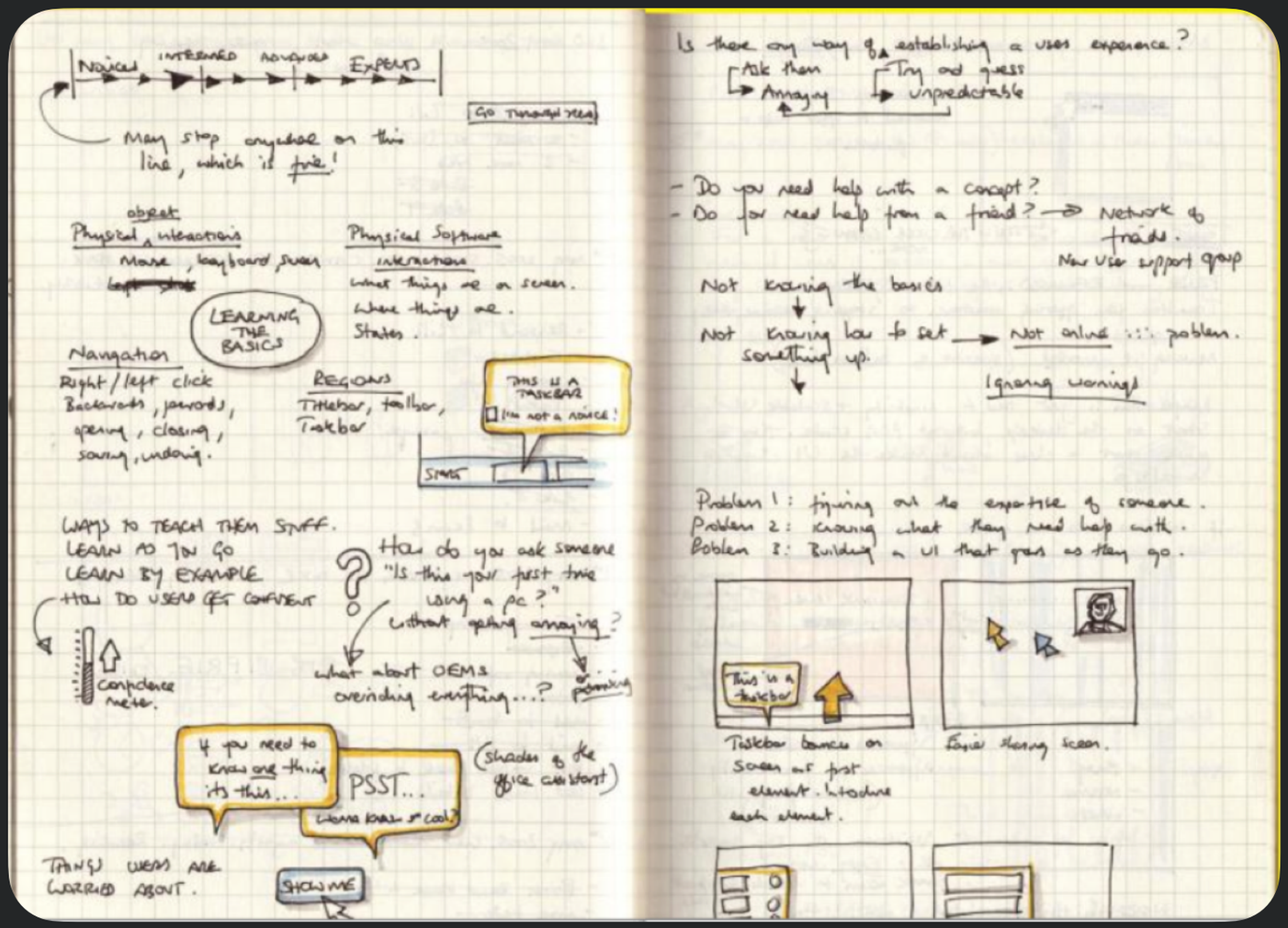
- Annotations explain what is going on in each part of sketch & how



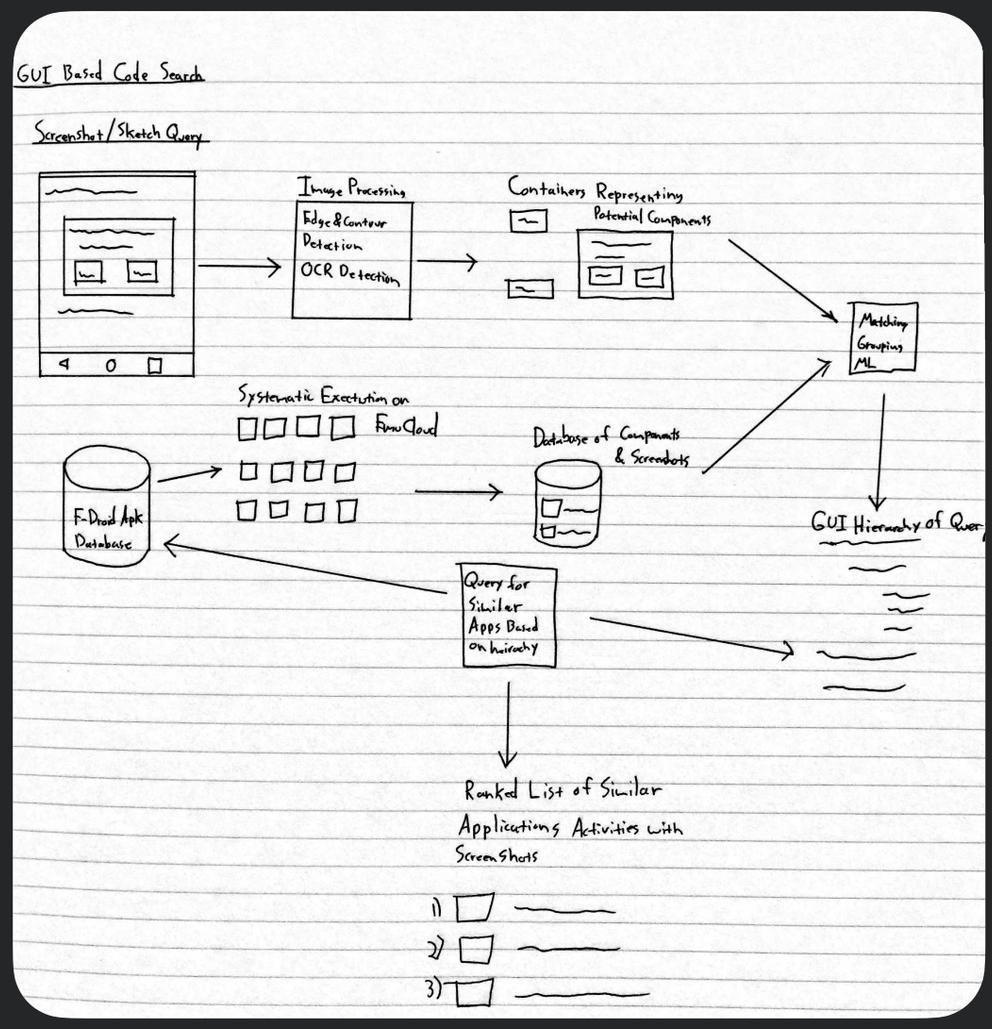



*Myers et al. (2008). How Designers Design and Program Interactive Behaviors. VL/HCC 2008.*

# Sketches are Part of Design Exploration



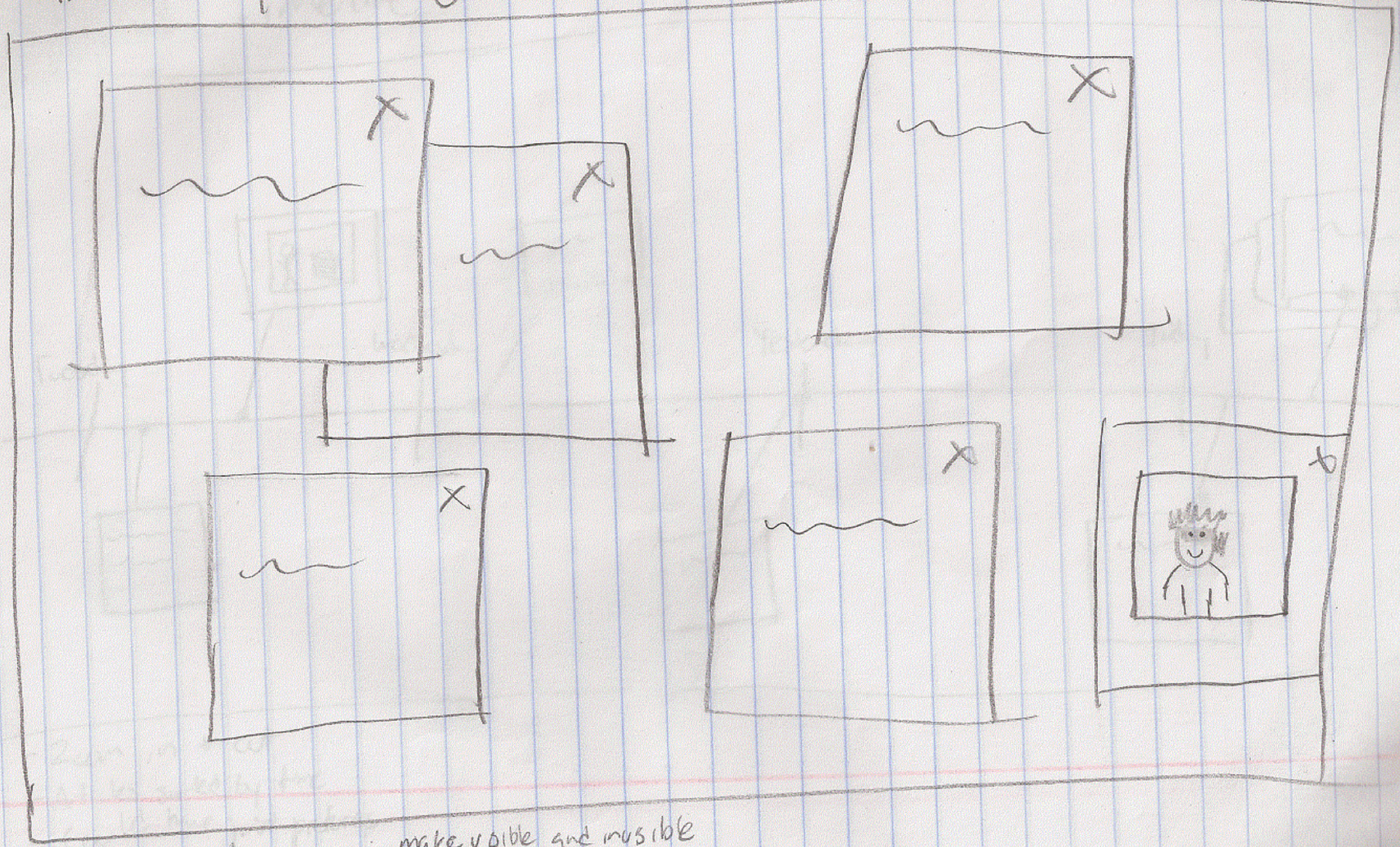

*B. Buxton. Sketching User Experiences.*




*K. Moran, ReDraw Project Sketch*

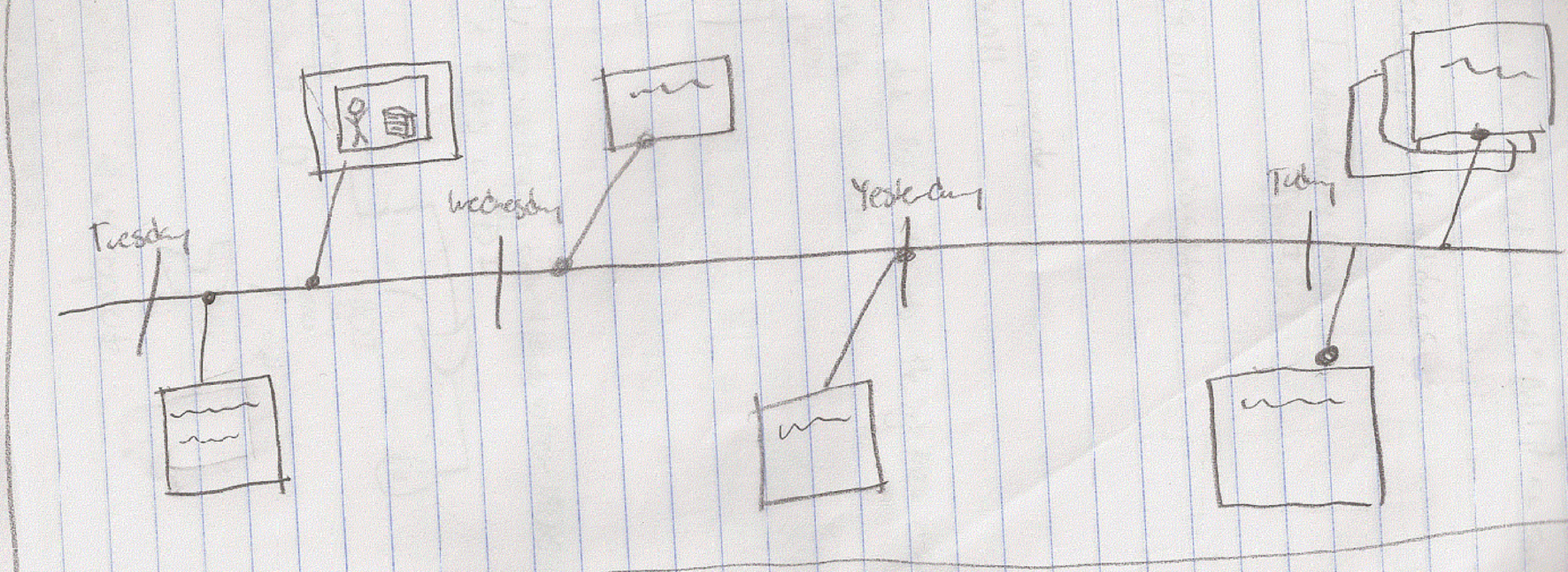
# Sketching Example: News Viewer

Article Layout through moveable windows (DADA) - drag and drop articles



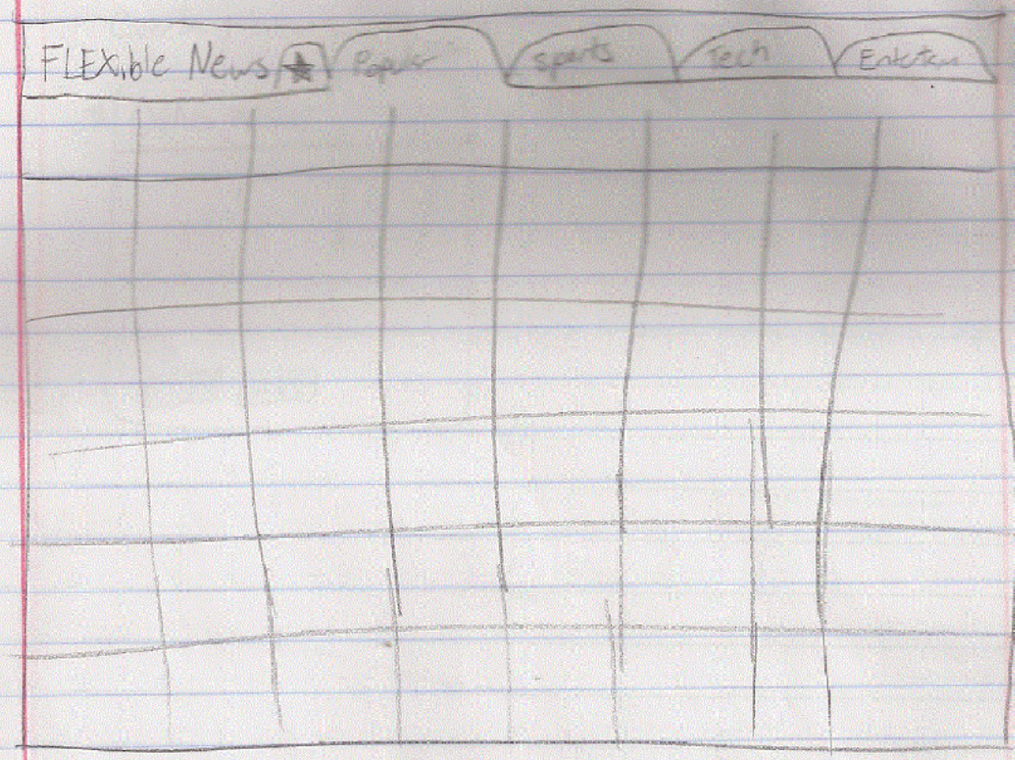
- Moveable windows
- closeable
- Layered by importance
- make visible and invisible

# News Timeline

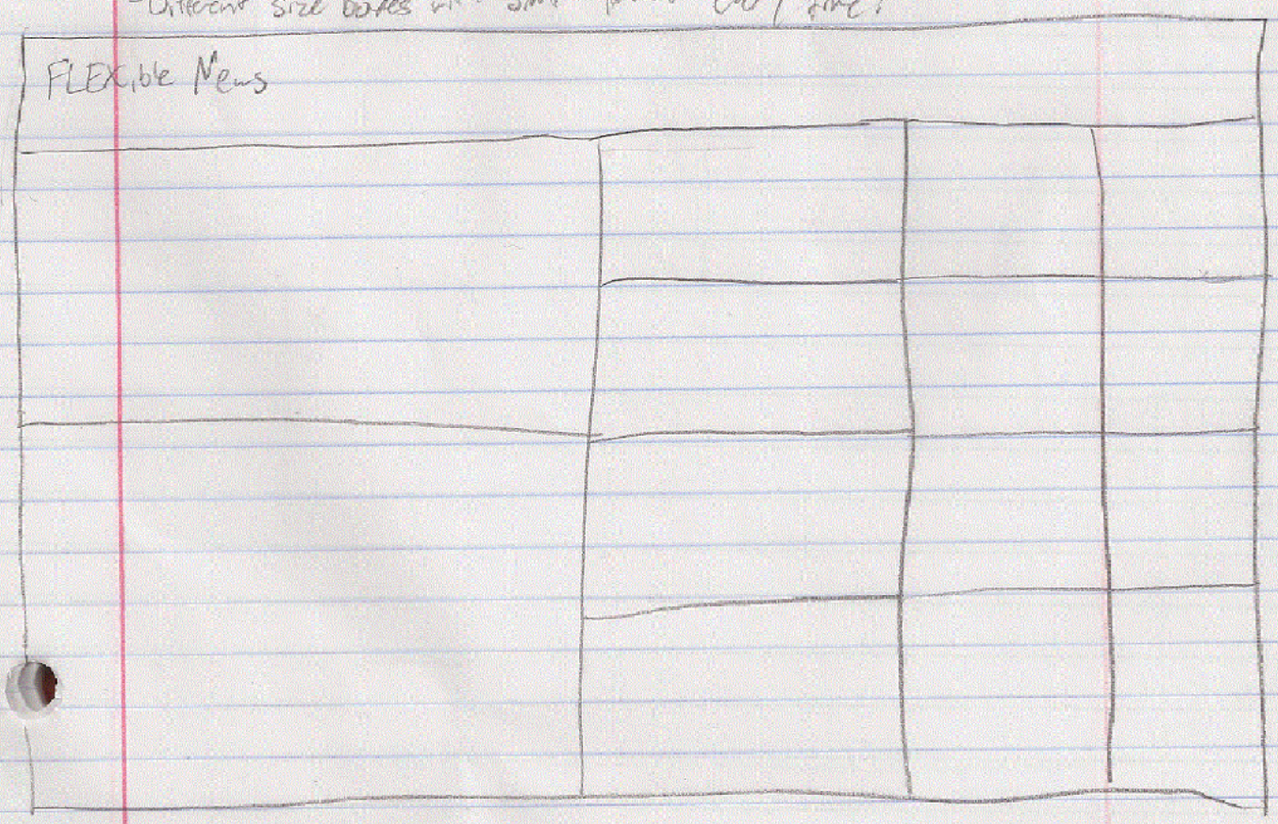



- Zoom in & out
- Articles sorted by time
- Could have just pictures

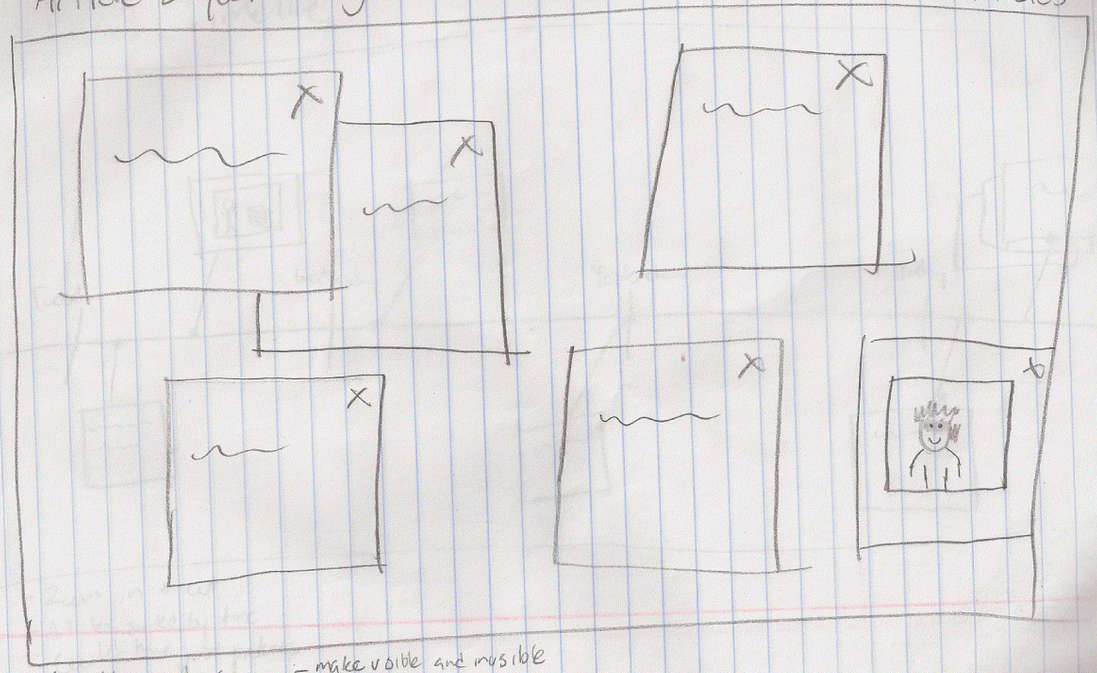
# UID Wireframe




- Even boxes?
- Different size boxes with simple format every time?

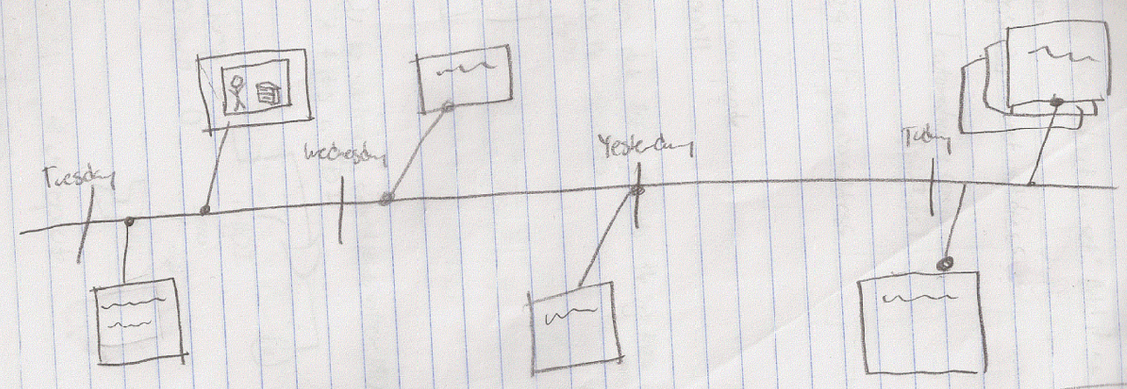


Article Layout through moveable windows (DADA) - drag and drop articles



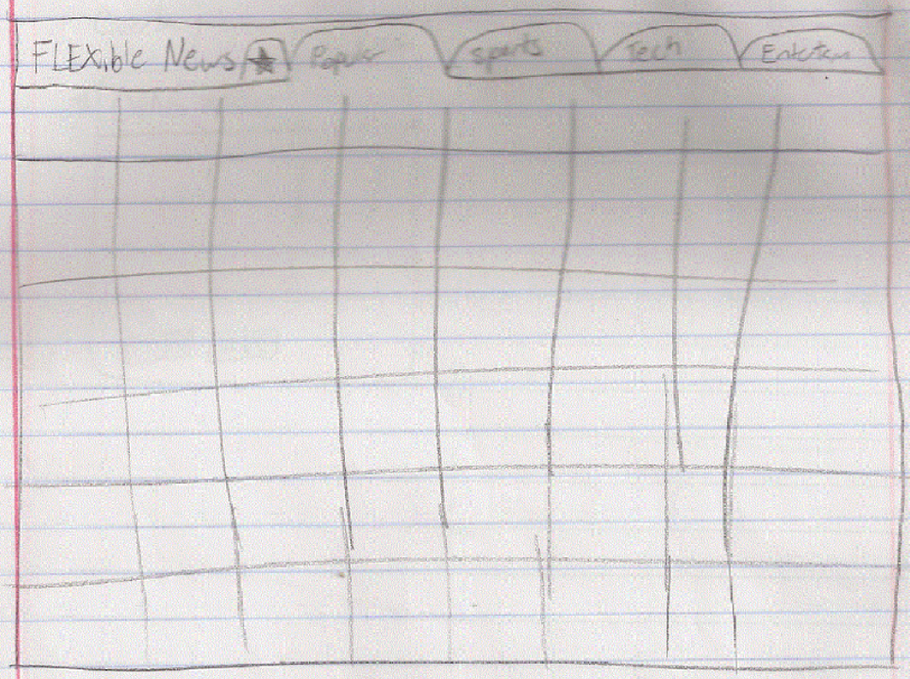
- Moveable windows
- closeable
- Layered by importance
- make visible and invisible

News Timeline

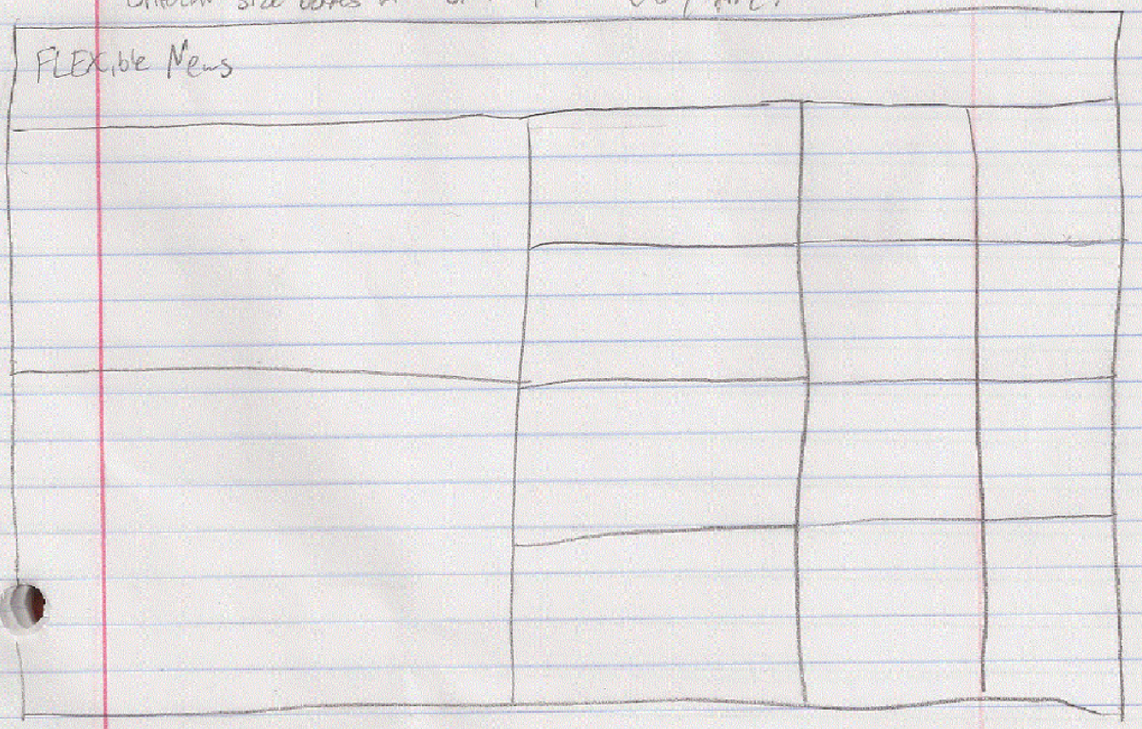



- Zoom in & out
- Articles sorted by time
- Could have just pictures

UID Wireframe

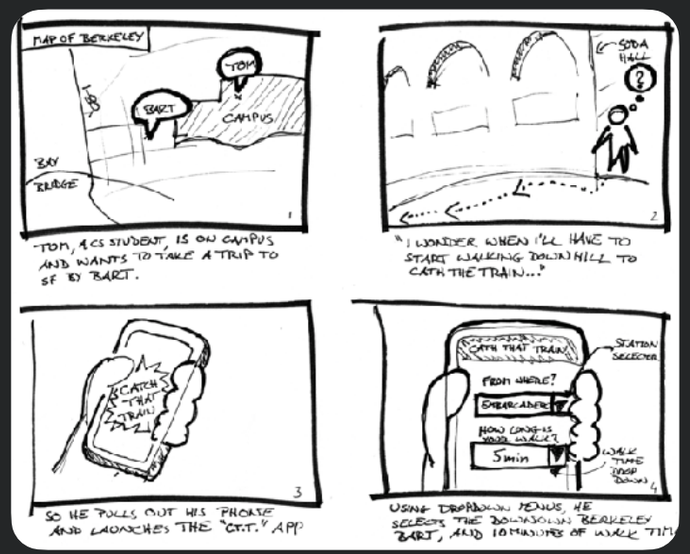


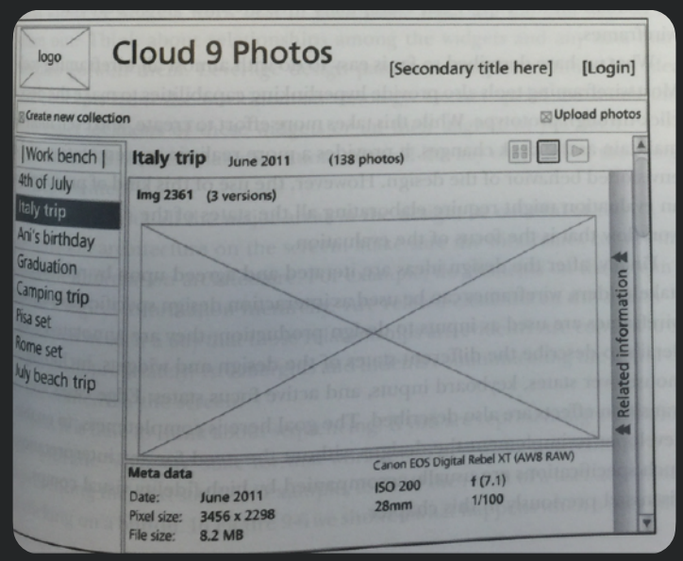

- Even boxes?
- Different size boxes with similar format every time?

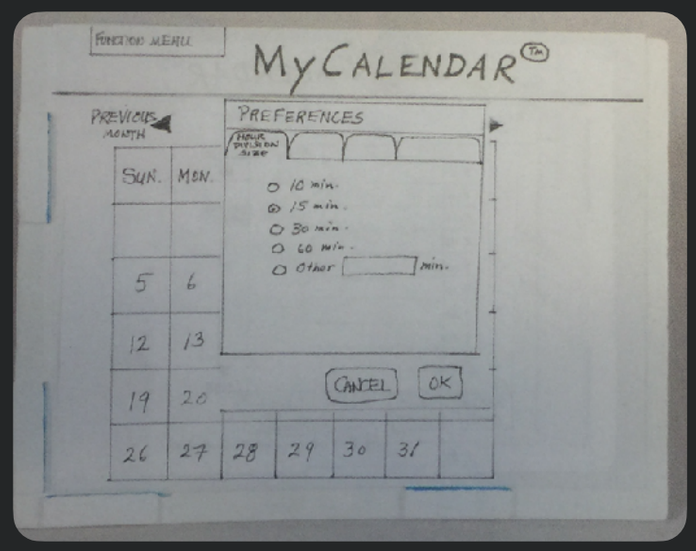


# Storyboards

# Fidelity of Sketches & Mockups




Storyboard — Wireframe — Prototype

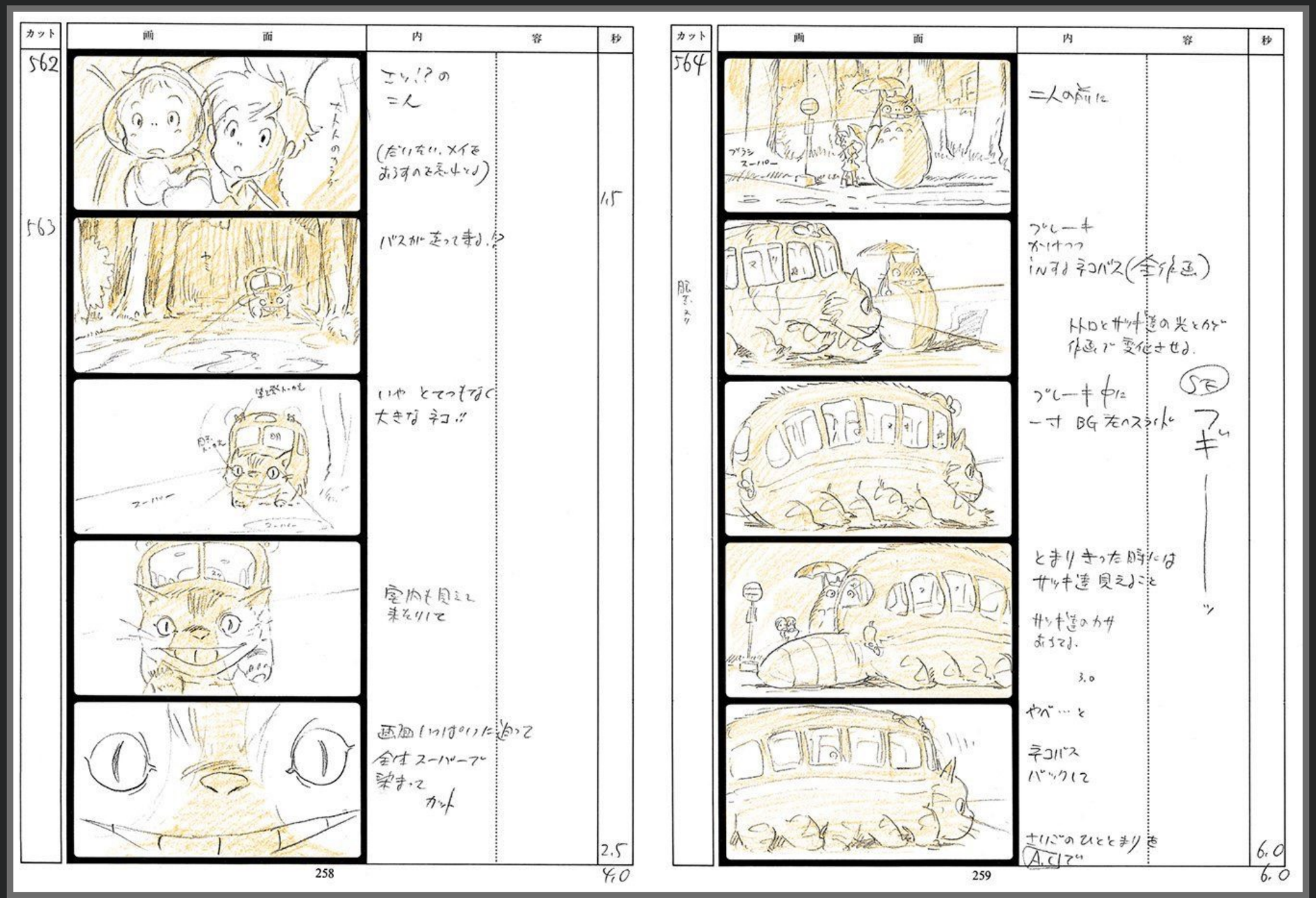
low ← → high

Fidelity

(many details left unspecified)

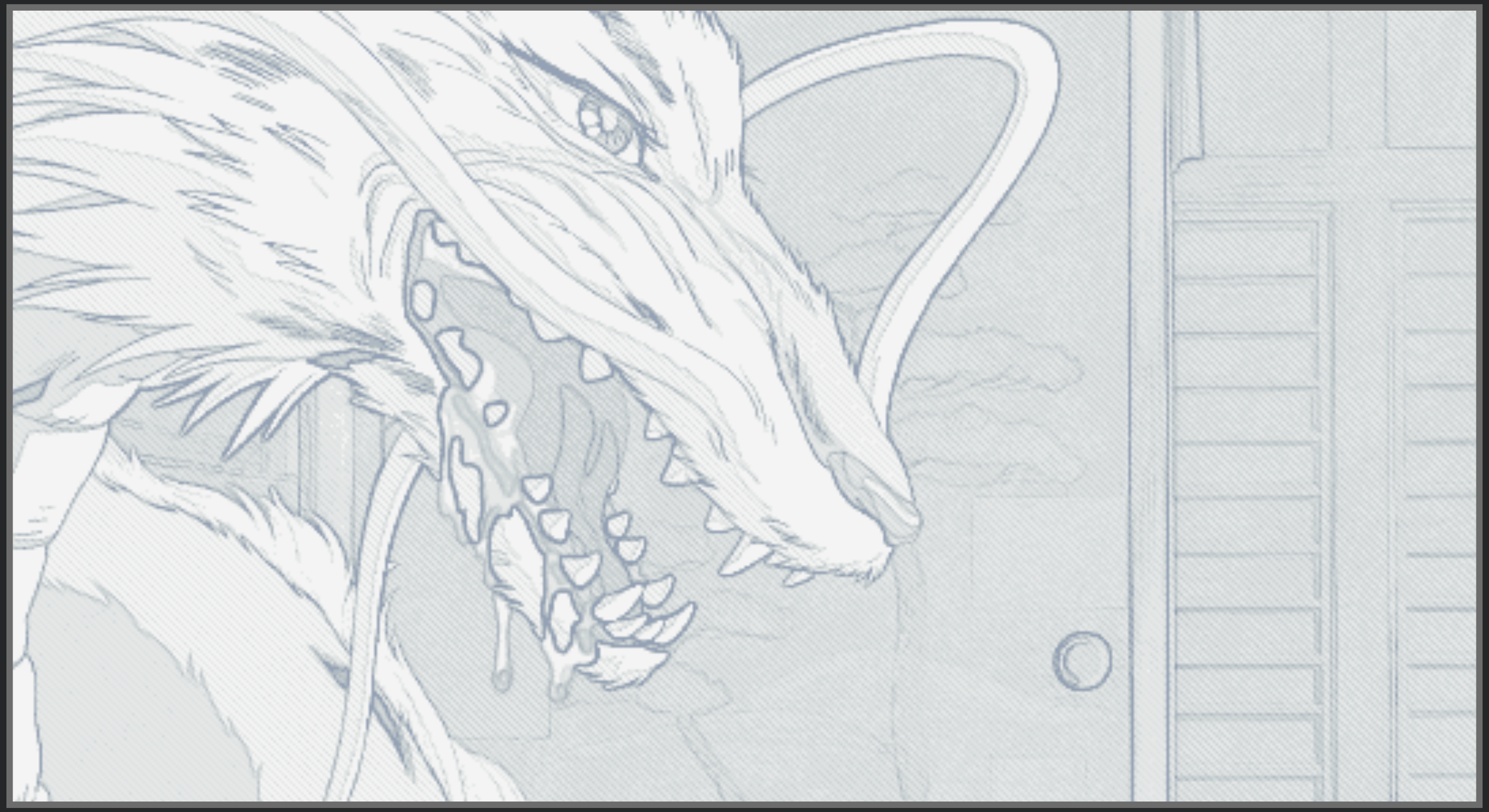
(more polished & detailed)

# Classic StoryBoards




Storyboard from Studio Ghibli: *“My Neighbor Totoro”*

# Classic Storyboards



Credit Studio Ghibli: *“Spirited Away”*

# Storyboards for UI Design

- Sequence of visual “frames” illustrating ***interplay*** between user & envisioned system
- Explains how app fits into a larger ***context*** through a single scenario / story
- Bring design to ***life*** in graphical clips - freeze frame sketches of user interactions
- “Comic-book” style ***illustration*** of a scenario, with actors, screens, interaction, & dialog

# Crafting a Storyboard

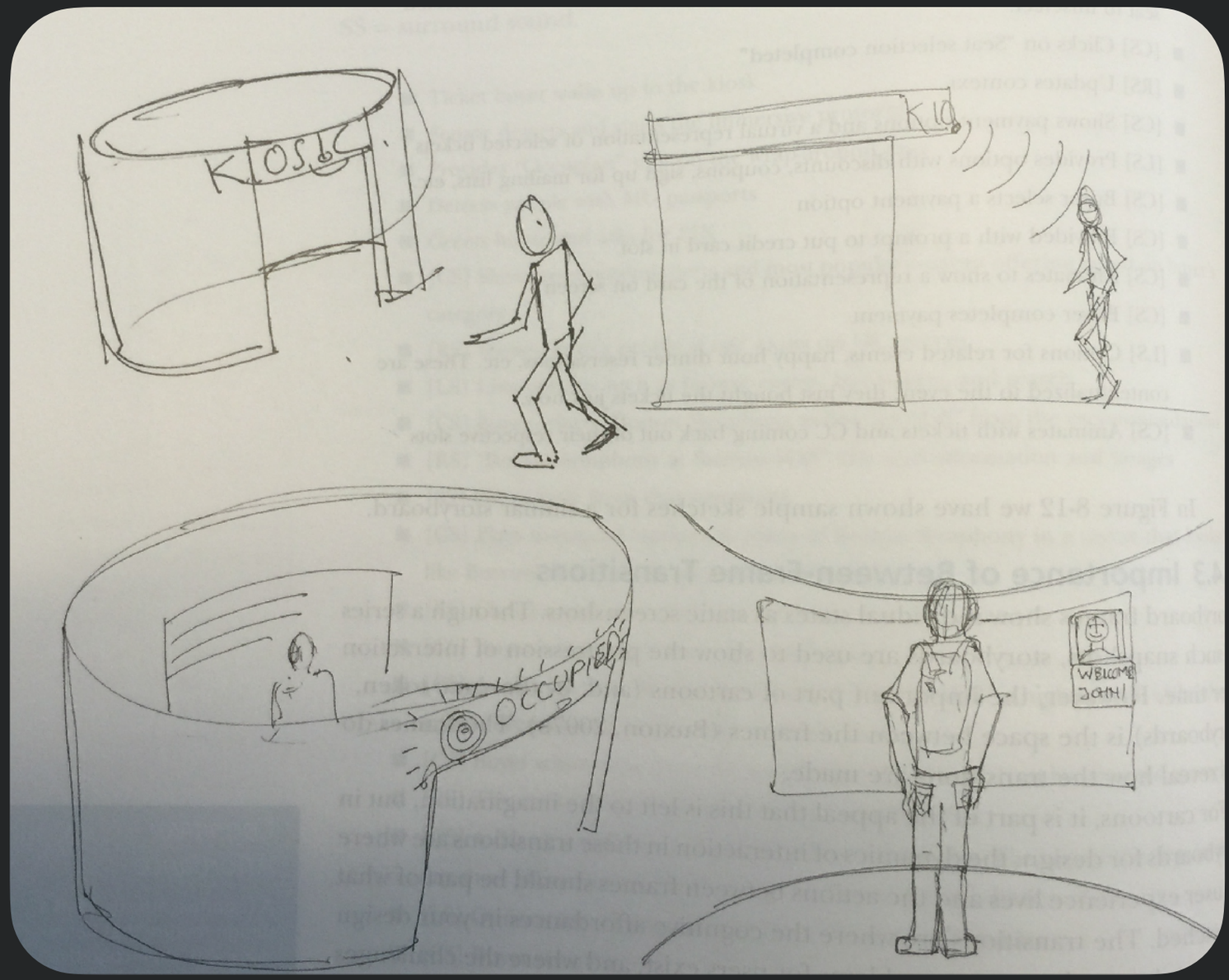
- Set the stage:
  - Who? What Where? Why? When?
- Show key interactions with application
- Show consequences of taking actions
- May also think about errors

# Example Elements of a UI Storyboard

- Hand-sketched pictures annotated with a few words
- Sketch of user activity before or after interacting w/ system
- Sketches of devices & screens
- Connections with system (e.g., database connection)
- Physical user actions
- Cognitive user action in “thought balloons”

# Example: Ticket Kiosk

Ticket buyer walks up to the kiosk



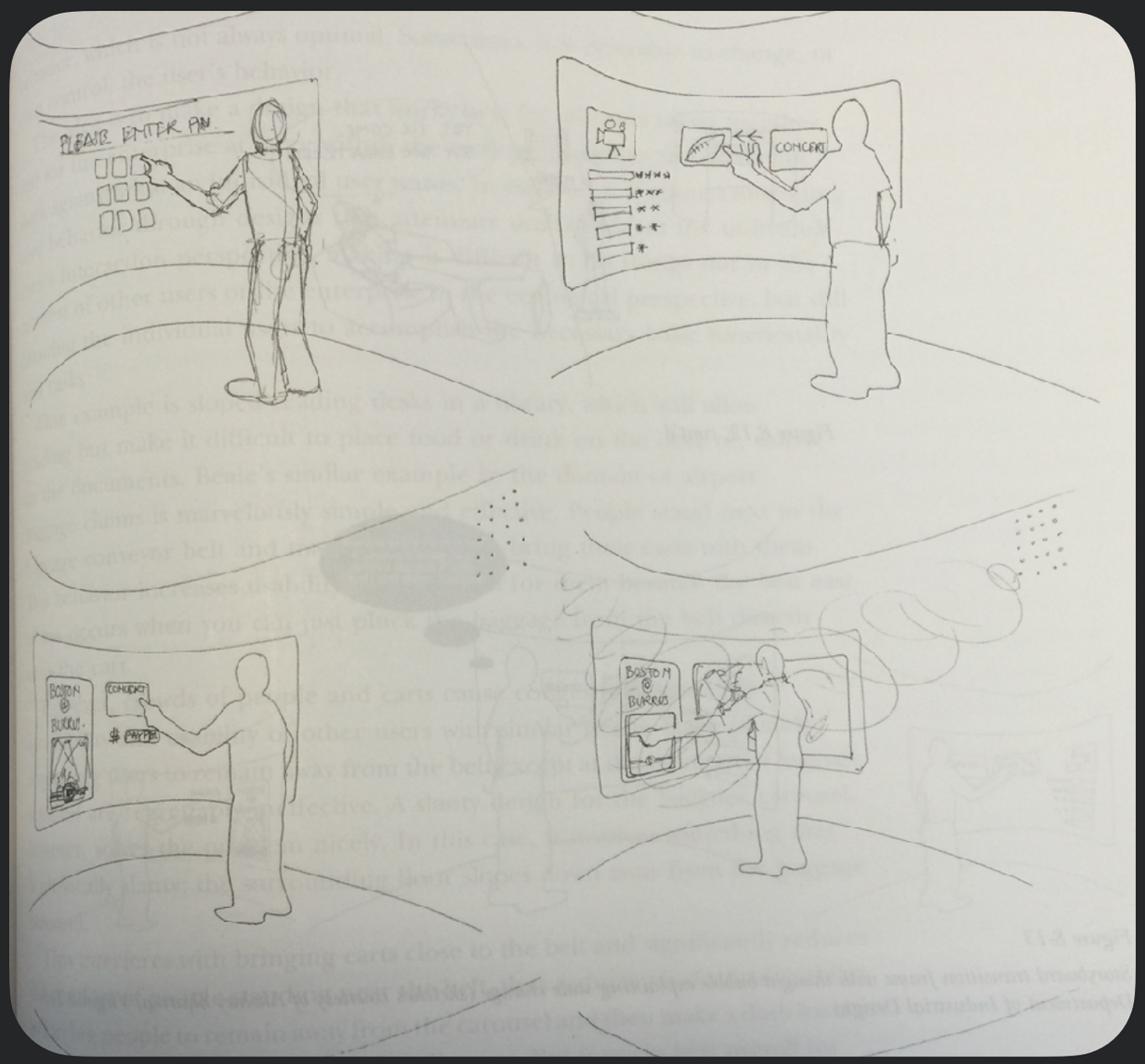

Sensor detects user & starts immersive process

Displays “Occupied” sign on wraparound case

Detects people with ID card

# Example: Ticket Kiosk

Greets buyer and asks for PIN




Shows recommendations & most popular categories

Buyer selects “Boston symphony at Burruss Hall”

Plays music from symphony, shows date & time picker

# Frame Transitions

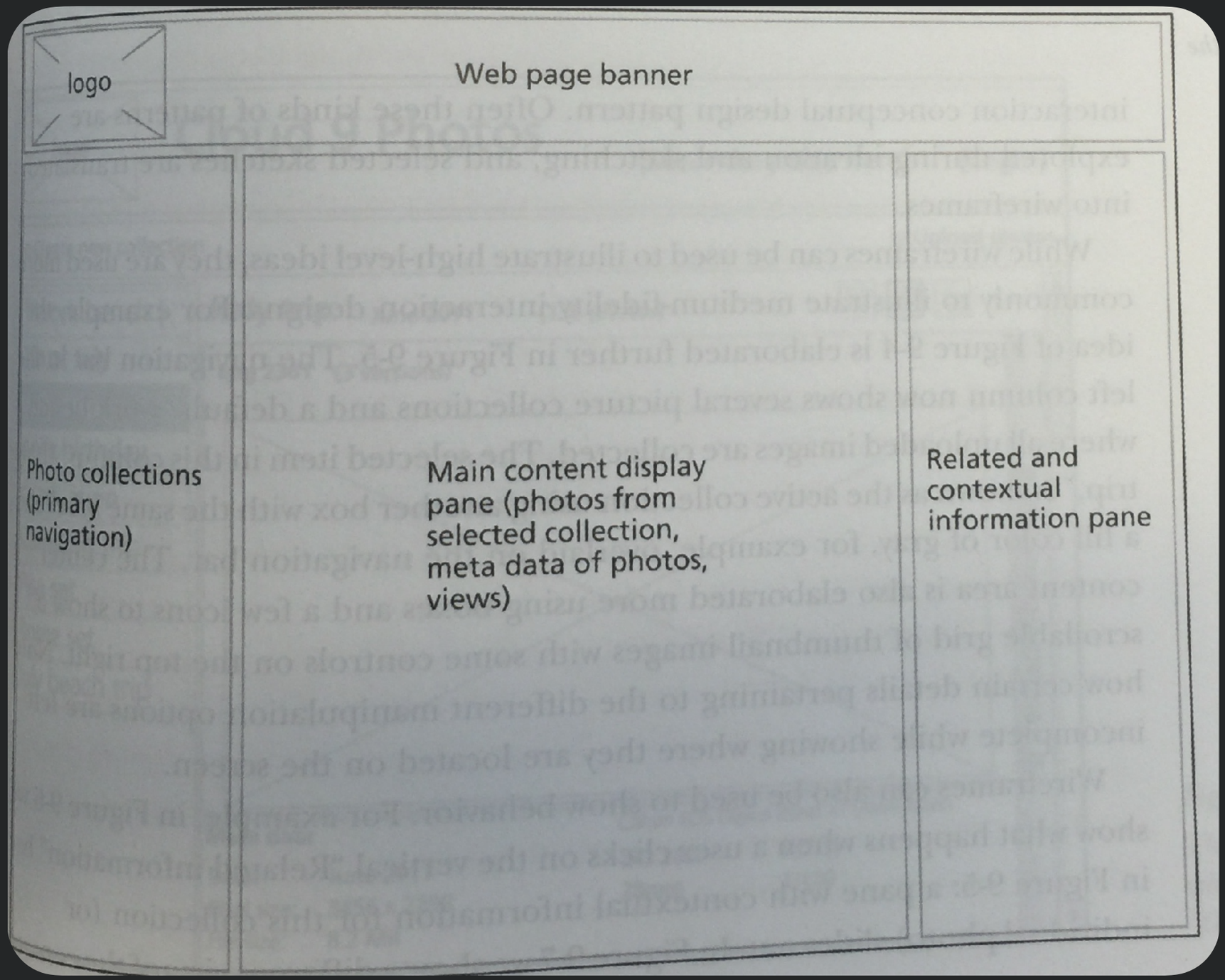
- Transitions between frames particularly important
- What users think, how users choose actions
- Many problems can occur here (e.g., gulfs of execution & evaluation) - we will talk more in a future class!
- Useful to think about how these work, can add thought bubbles to describe

# Wireframes & Design Critiques

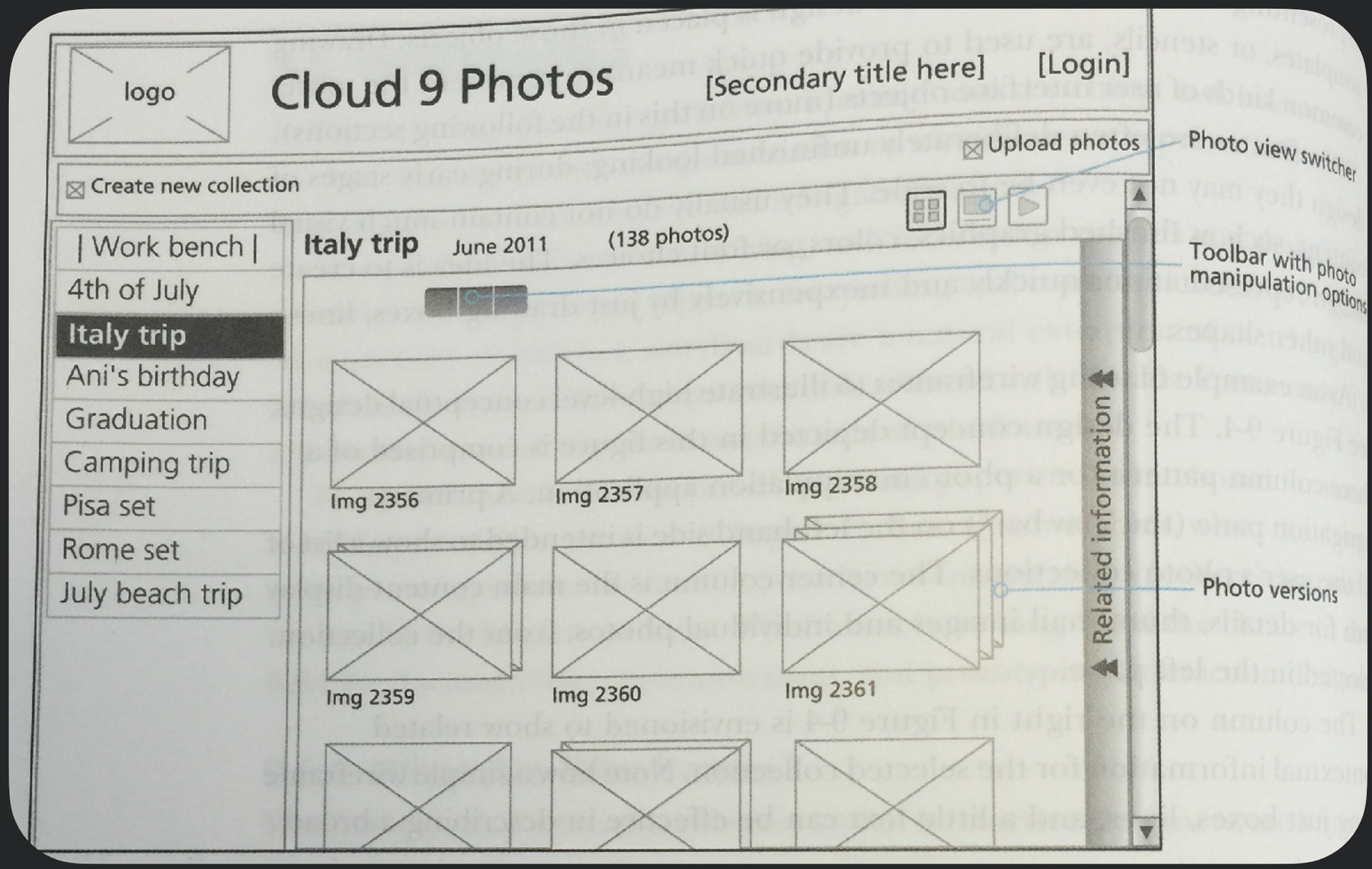
# Wireframes

- Lines & outlines (“wireframes”) of boxes & other shapes
- Capturing emerging interaction designs
- Schematic designs to define screen content & visual flow
- Illustrate approximate visual layout, behavior, transitions emerging from task flows
- Deliberate unfinished: do not contain finished graphics, colors, or fonts

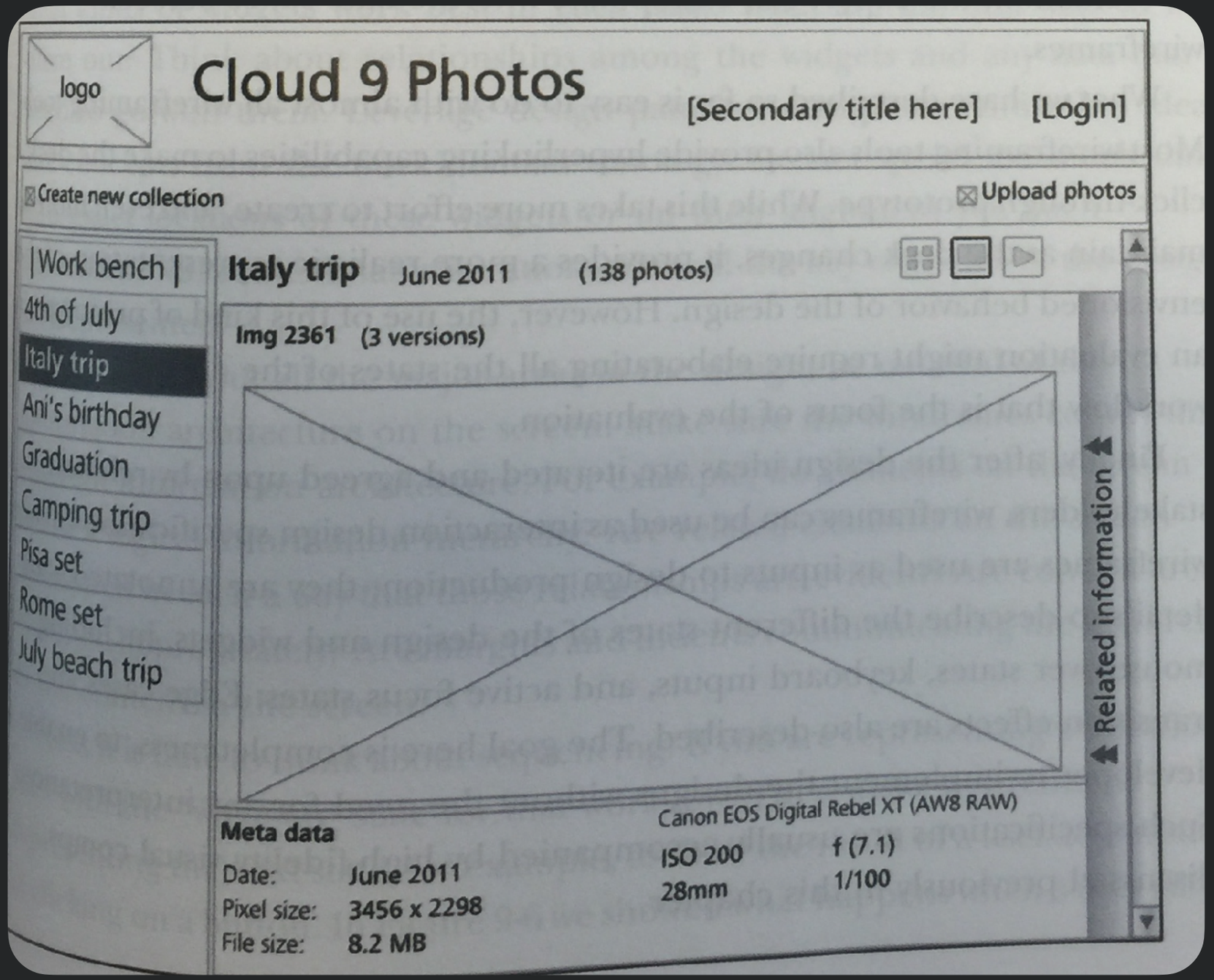
# Example


logo
Web page banner
Photo collections (primary navigation)
Main content display pane (photos from selected collection, meta data of photos, views)
Related and contextual information pane

# Example


logo
Cloud 9 Photos
[Secondary title here]
[Login]
Upload photos
Create new collection
Photo view switcher
| Work bench |
4th of July
Italy trip
Ani's birthday
Graduation
Camping trip
Pisa set
Rome set
July beach trip
Italy trip
June 2011
(138 photos)
Toolbar with photo manipulation options
Related information
Img 2356
Img 2357
Img 2358
Img 2359
Img 2360
Img 2361
Photo versions

# Example


logo
Cloud 9 Photos
[Secondary title here]
[Login]
Create new collection
Upload photos
| Work bench |
4th of July
Italy trip
Ani's birthday
Graduation
Camping trip
Pisa set
Rome set
July beach trip
Italy trip
June 2011
(138 photos)
Img 2361
(3 versions)
Related information
Meta data
Date:
June 2011
Pixel size:
3456 x 2298
File size:
8.2 MB
Canon EOS Digital Rebel XT (AW8 RAW)
ISO 200
f (7.1)
28mm
1/100

# Wireframes

- Can be used to step through a particular scenario
- Focus on key screens rather than every screen
- Tools can help
  - Can be made clickable
  - Can use stencils & templates; copy & edit similar screens

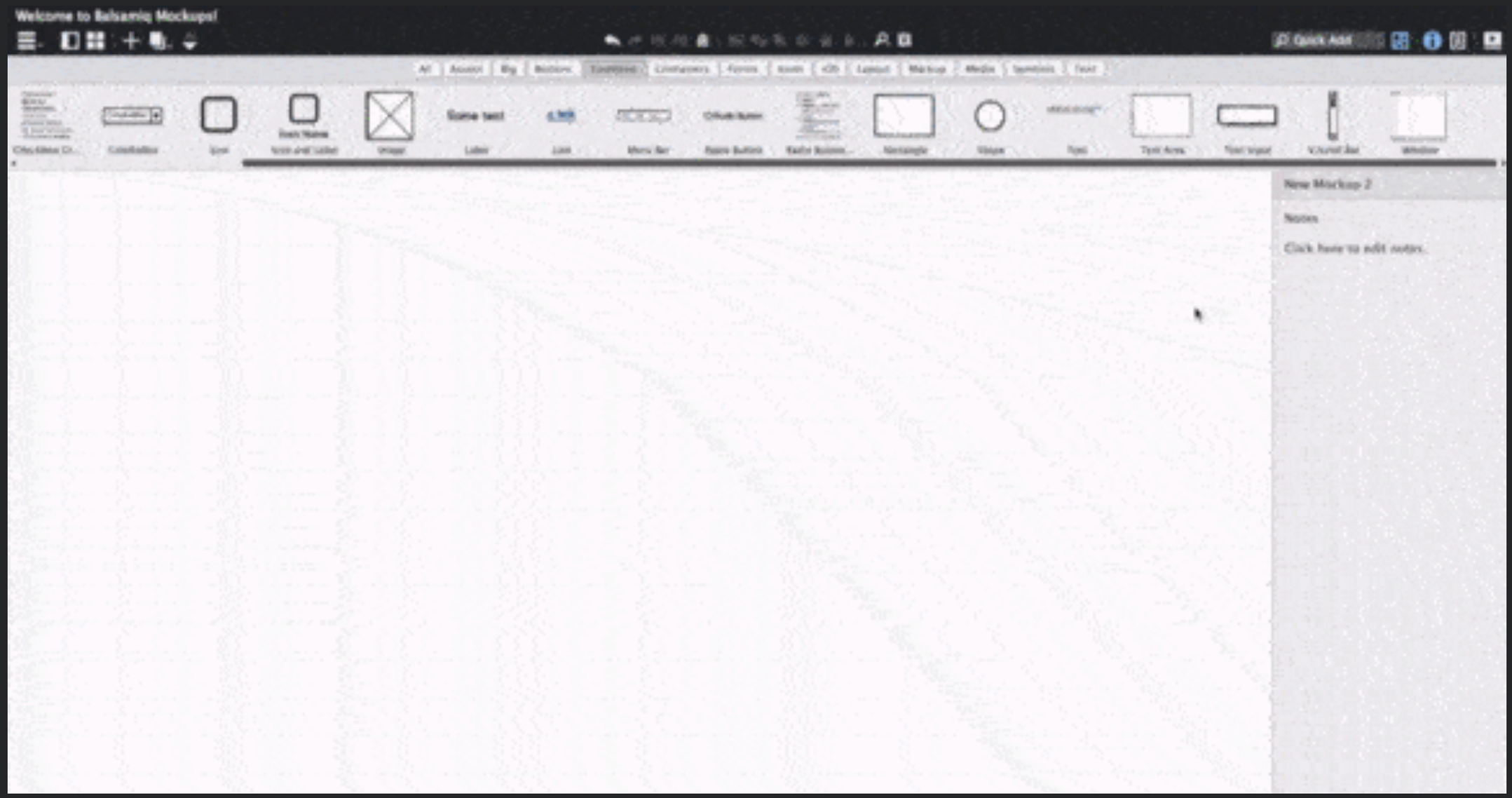
# Creating a Wireframe - (1)

- What are the key interactions needed to support design?
- What widgets support these interactions?
- What are the best ways to lay them out?
- How do these relate to conceptual design & user's mental model?

# Creating a Wireframe - (2)

- What are all of the items: toolbars, scrollbars, windows, …?
- Are there too many widgets on the screen?
- What happens when data is larger than available space? Will entire page scroll, or individual panel?
- How much detail of items to show?

# Example Tool - Balsamiq

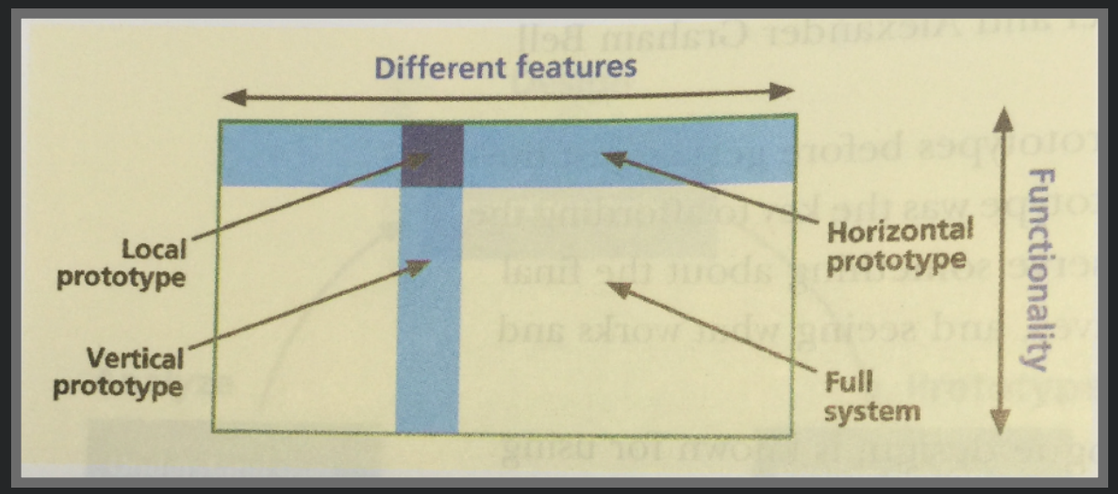


# Prototyping

# Prototyping

- How do you know your system design is right before you invest the time to build it?
- Answer: prototyping!
  - Evaluation performed ***before*** investing resources in building finished product
  - Early version of system constructed much ***faster*** & with less expense used to evaluate & ***refine*** design ideas

# Types of Prototypes




- Which details do you leave out?
- **Horizontal**: ***broad*** in features, less depth
  - Explore overall concept of app, but not specific workflows
- **Vertical**: lots of ***depth***, but only for a few features
  - Enables testing limited range of features w/ realistic user evals
- **T**: most of UI realized at low depth, few parts realized in depth
  - Combination of vertical & horizontal
- **Local**: focused prototype on ***specific*** interaction detail

# Interactivity of Prototypes

- Scripted, click through prototypes
  - Prototype w/ ***clickable*** links to move between screens
  - Live action storyboard of screens
    - Simulates real ***task flow***, but w/ static content
- Fully-implemented prototypes
  - Usually ***expensive*** to implement actual system
  - But can build key piece of system first to evaluate

# Wizard of Oz

- Goal: ***simulate*** actual system w/ out building it
  - Want user to interact ***as if*** they were interacting w/ real system
    - Helps explore how users would interact w/ novel interaction if it were to exist
- Example: natural command line (Good et al 1984)
  - Users typed in commands to interact w/ computer
  - Commands intercepted by hidden human who interpreted commands & executed them

# In Class Activity

# Group activity

- In groups of 2/3:
  - *Part 1: Apply Heuristics to a website (e.g., Word, Twitter)*
    - Work individually to identify at least 1 usability issue
    - For each issue, identify the heuristic, identify the functionality in the application, and summarize how the heuristic is violated in a few sentences
    - Use Online Google Document shared on Ed
  - *Part 2: Design an improved version of the site/app you chose*
    - Start with a specific set of user needs identified
    - Create Wireframe design of a new system that addresses the users' needs
    - Build a series of at least 2 wireframe "pages" supporting one scenario for the app.
    - Use draw.io folder shared on Ed.

# Acknowledgements


- Slides adapted from Dr. Thomas Latoza's SWE 432 course