



PROTOTYPING: LOW TO HIGH FIDELITY PROTOTYPING

CPSC 544 FUNDAMENTALS IN DESIGNING INTERACTIVE COMPUTATION
TECHNOLOGY FOR PEOPLE

© 2023 Karon MacLean – University of British Columbia

Material created by Joanna McGrenere, Karon MacLean, Leila Aflatoony, Jessica Dawson & Heather O'Brien

COMING UP

This week (Oct 23)

- Sun 10/22:
 - Researcher Journal #9
- Tues 10/25
 - Researcher Journal #10
 - Team deliverable: Finalized Conceptual Model & Storyboard (Report)

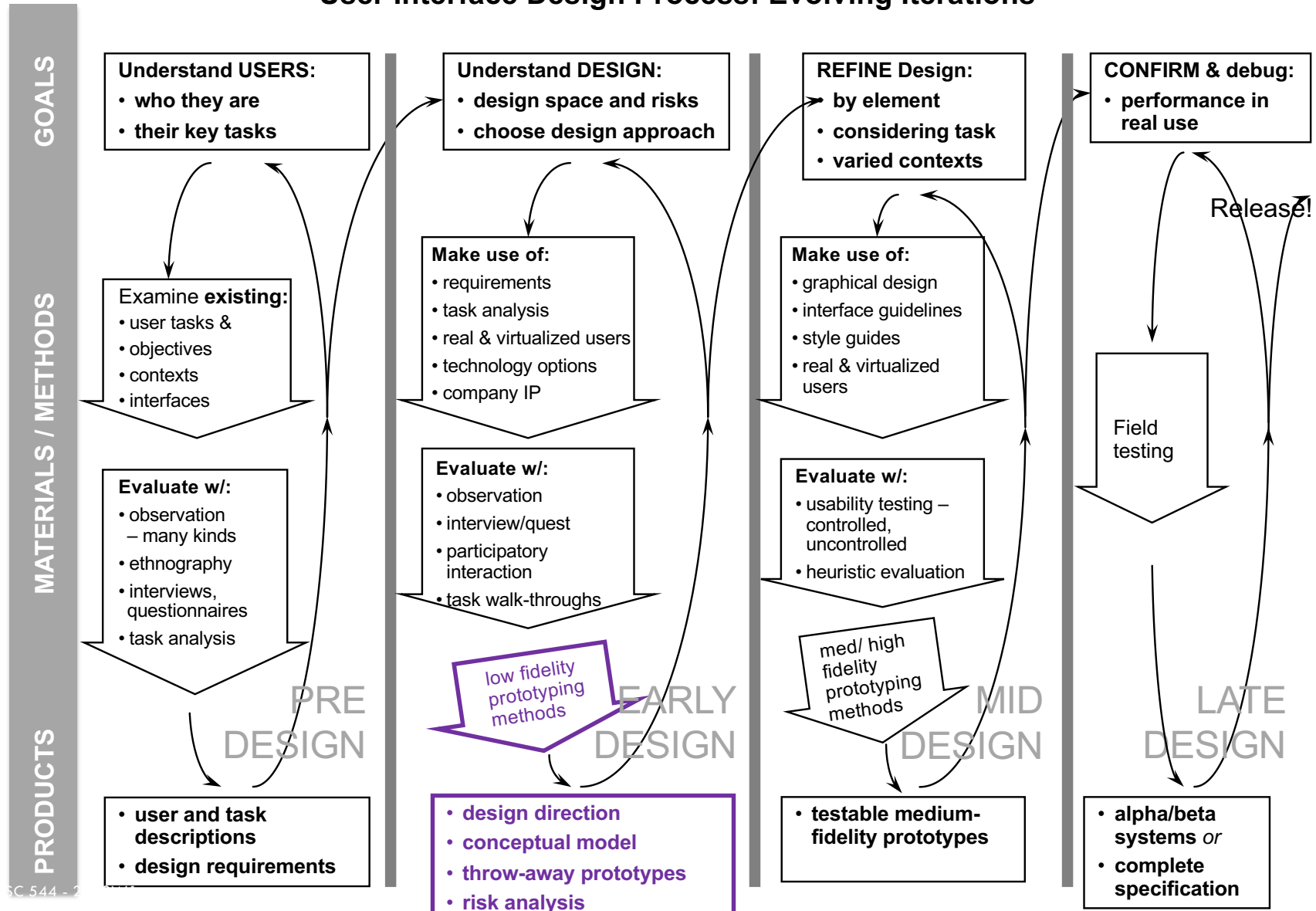
Coming Up

- Sun 10/29 – Tues 10/31:
 - Researcher Journals #11, 12; mid-course survey
- Mon 11/06: Low-Fi Prototype Presentations
- Wed 11/08: In-class: cog walkthrough of lo-fi prototypes
- Midterm break – no class or deliverables all week

LEARNING GOALS

- Understand different types of prototyping, purpose and characteristics of each
- List dimensions of prototyping fidelity and explain how these dimensions may vary
- Explain how these dimensions might differ in low to med to high fidelity prototypes, and give examples of when/why you may use each type
- Make strategic choices about prototyping tools given your goals and constraints; be able to justify your choice

User Interface Design Process: Evolving Iterations



SC 544 - 2

K MacLean - derived from version by Saul Greenberg (U Calgary)

FROM CONCEPTUAL TO CONCRETE

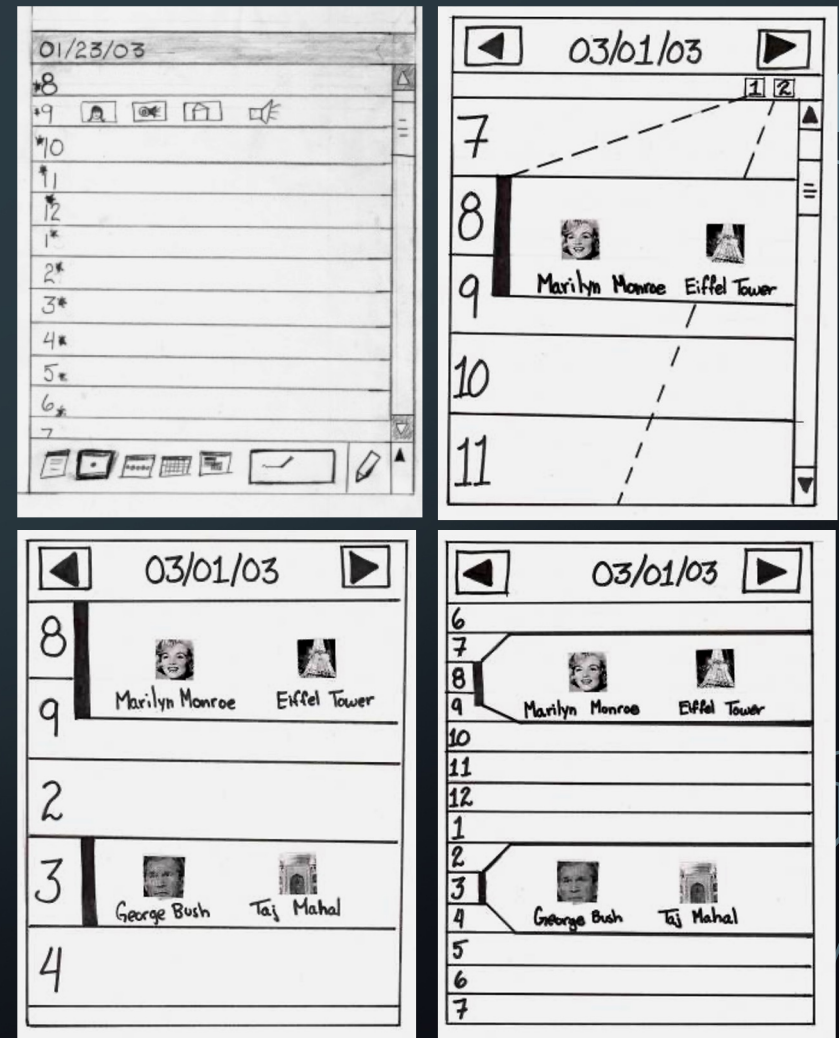
- Interface design goal → communicate conceptual model
- Challenge:
 - Designer's conceptual model is communicated via system image:
 - Interface, appearance, instructions, system behavior through interaction
 - If system image does not make model clear and consistent:
 - user's mental model will be inconsistent with conceptual model

How do we start designing the interface?
Prototyping!

WHAT IS A PROTOTYPE?

- Representation of conceptual design for users to interact with
 - and designers & other stakeholders
- Prototypes take many forms:
 - Cardboard, foam, software, video, clay, paper, hidden people, website, sketches, scripts, index cards, etc.
- To begin to build ... Need:
 - Users and tasks to build your prototype around
 - Requirements
 - Goals: questions your prototype(s) need to answer

4 designs: image-enhanced planner



WHY PROTOTYPE?

Many different kinds of goals and questions possible

- Communication: discuss ideas with stakeholders
 - “Where’s the ON button?”
- Develop requirements and/or specifications
 - “Uh-oh, here’s something we forgot.”
- Learning and problem solving
 - “Hey, that will work!”
- Evaluate interface effectiveness for communicating conceptual model
 - “Whoops, users didn’t understand that.”
- Further develop conceptual and physical design
 - “That’s way too heavy”
- Save time and money
 - Don’t waste time coding/building the wrong thing

NOT JUST FOR SOFTWARE: HANDHELD “UNIVERSAL REMOTE CONTROL”

Conceptual Prototypes



WHEN TO USE DIFFERENT TYPES OF PROTOTYPES?

early design

Choose a representation; Rough out interface style; Task walkthrough & redesign

Fine tune interface, screen design
Heuristic evaluation and redesign

Usability testing and redesign
Limited field testing

Alpha/Beta tests

- Low fidelity prototypes

- Medium fidelity prototypes

- High fidelity prototypes

- Working systems

late design

APPROACHES TO PROTOTYPE/ PRODUCT INTEGRATION

- Throw-away
 - prototype only serves to elicit user reaction
 - creating prototype must be rapid, otherwise too expensive
- Incremental
 - product built as separate components (modules)
 - each component prototyped and tested, then added to the final system
- Evolutionary
 - prototype altered to incorporate design changes
 - eventually becomes the final product

APPROACHES TO 'SCOPING' PROTOTYPE FUNCTIONALITY

- **Vertical prototype**

- includes in-depth functionality for only a few selected features
- key design ideas can be tested in depth

- **Horizontal prototype**

- surface layers only: includes the entire user interface with no underlying functionality
- a simulation; no real work can be performed

Common strategy at low, med fi stage:
do both -- horizontal plus a slice of vertical with limited focus.



LOW FIDELITY PROTOTYPES

- Meant to be rough, quick to build, easy to throw away
- Purposes
 - proof of concept(s)
 - rough (but flexible) interface design
 - facilitate communication with users early on
 - useful for generating and narrowing requirements

BENEFITS OF LOW FIDELITY PROTOTYPES

- Cheap/easy to make
- Try out and explore multiple conceptual models
- Lack of polish less intimidating to users (this is surprisingly important)
- More willingness to criticize
- Inspires more creative feedback
- Avoids nitpicky feedback
- Reduces effort invested by design team
- So easier to make changes, start over

IDEO SURGICAL TOOL PROTOTYPE

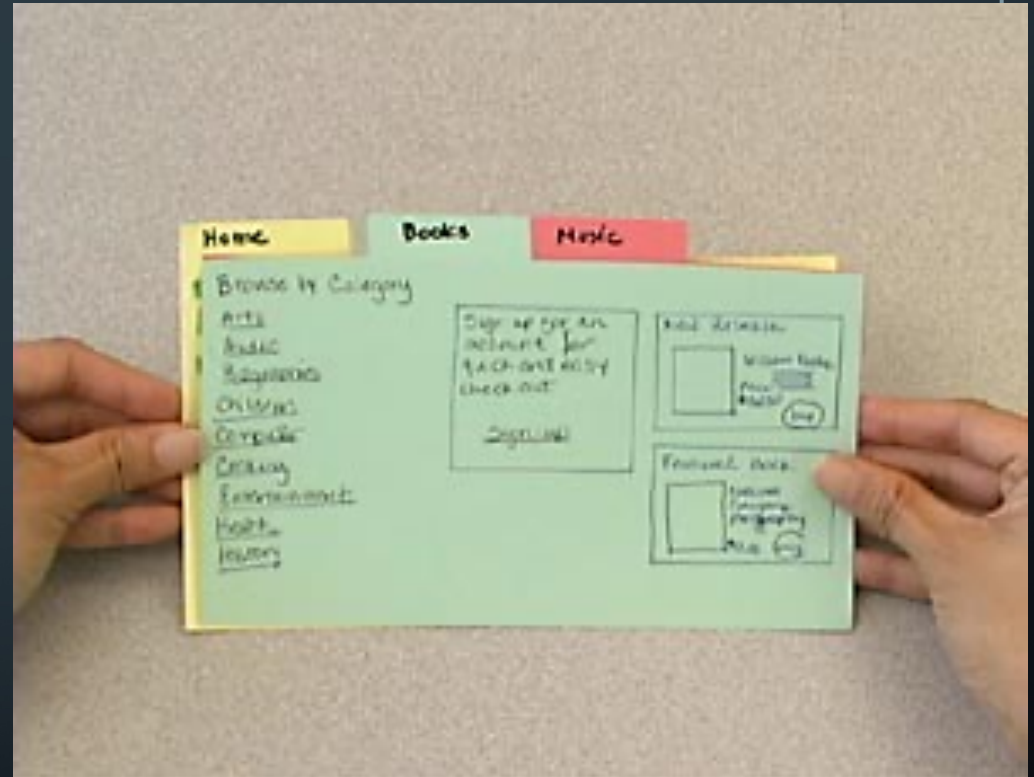


LO-FI PROTOTYPING METHODS

PAPER | SKETCHING | STORY BOARDING | SCRIPTED SIMULATIONS | WIZARD OF OZ

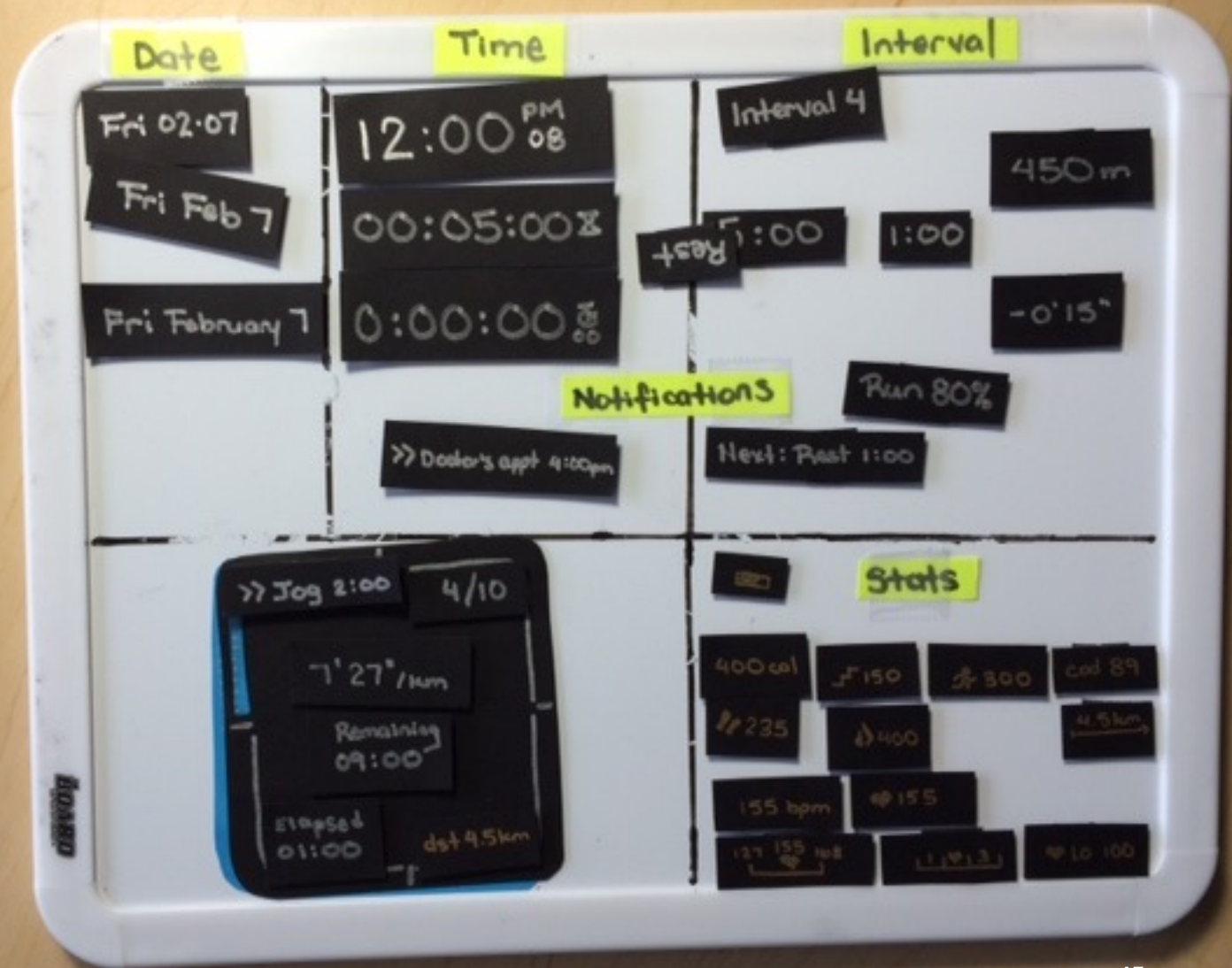
“PAPER” PROTOTYPING

- Common low fidelity technique
- Popular in industry . . .
- Despite prevalence of ‘mockup’ software tools because easy to:
 - build
 - alter on the fly
 - show
 - stick on wall & compare
 - discuss



Low-fi prototyping does not need to be paper

Haptok prototype
(Tam & MacLean):
the power of magnetic tape!



PAPER PROTOTYPING VS. SKETCHING

For different intentions,
use different sketching
approaches

Buxton, B. (2007). Sketching user experiences:
Getting the design right and the right
design. **Chapter 13-17**. *Sketching
interaction* (pp.135-155), Morgan Kaufmann
Publishers Inc.

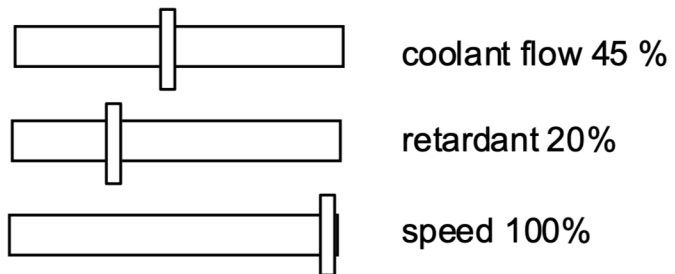
<u>SKETCH</u>		<u>PROTOTYPE</u>
EVOCATIVE	→	DIDACTIC
SUGGEST	→	DESCRIBE
EXPLORE	→	REFINE
QUESTION	→	ANSWER
PROPOSE	→	TEST
PROVOKE	→	RESOLVE
TENTATIVE	→	SPECIFIC
NONCOMMITTAL	→	DEPICTION

DIGITAL STORYBOARDS

- Draw each storyboard scene on computer
 - use wire framing/mockup software (e.g., balsamiq)
 - or painting/drawing packages (e.g., photoshop)
- A very thin horizontal prototype
- Does not capture the interaction “feel”

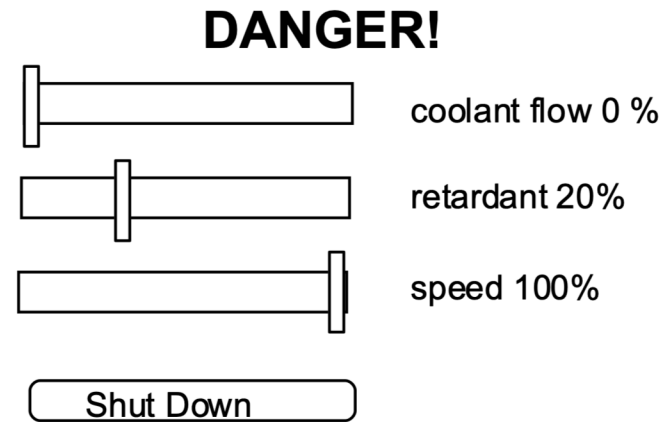
Elements aren't active:
They are like a paper
prototype, but on-screen

Control panel for pump 2



next
drawing
→
(for shut
down
condition)

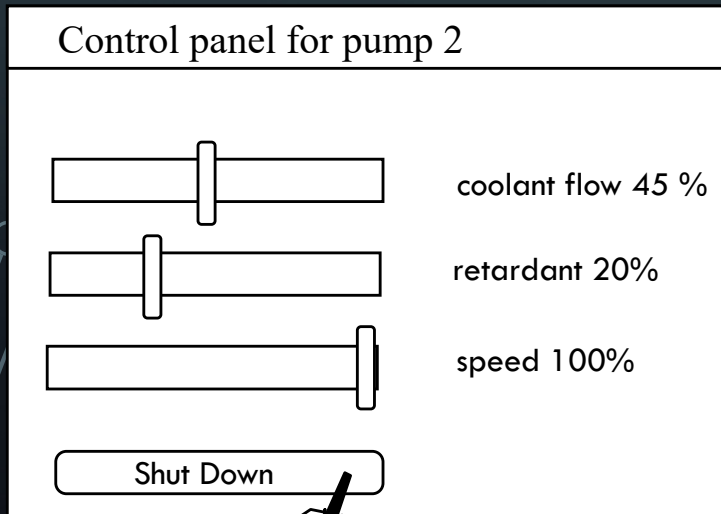
Control panel for pump 2



TECHNIQUE: SCRIPTED SIMULATIONS & SLIDE SHOWS

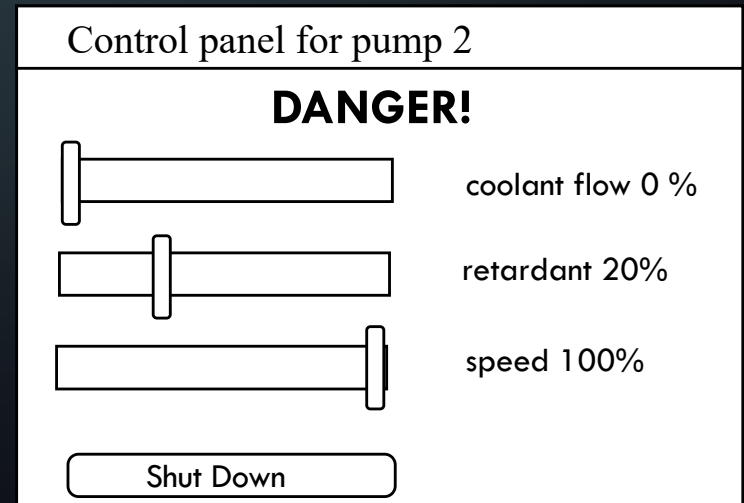
moving towards med-fi elements
can be active – but still only
narrow functionality

- Encode the storyboard on the computer
 - scene transition activated by simple user inputs (i.e. clickable regions)
 - a simple horizontal and/or vertical prototype
 - supports 'limited' branching
- User given a very tight script/task to follow
 - appears to behave as a real system, but script deviations blow the simulation



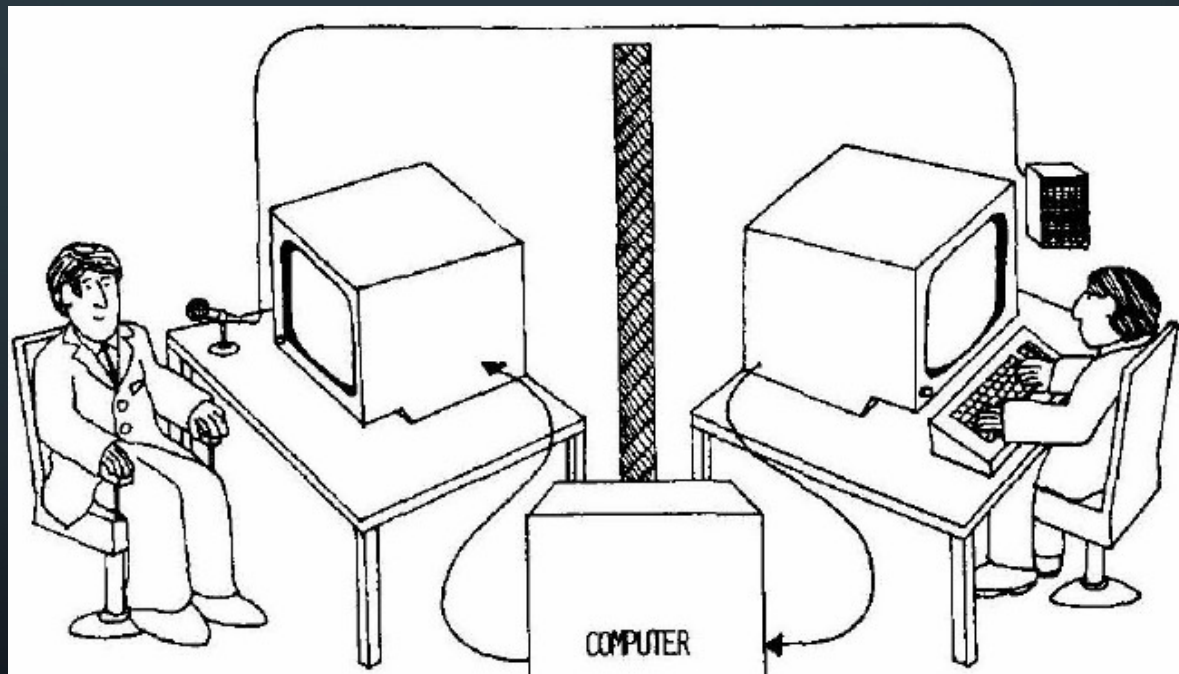
Next drawing →

On mouse, press
over button
(appears to be
active)



WIZARD OF OZ

- A totally different method of testing a system that does not exist
 - The voice editor, by IBM (1984)



WIZARD OF OZ (“WOZ”)

- Human simulates system’s intelligence & interacts with user
 - “Pay no attention to the man behind the curtain!”
- User uses computer as expected
- “Wizard” (sometimes hidden):
 - interprets subject’s input according to a preset algorithm
 - makes computer/screen behave in appropriate manner
- Good for:
 - adding simulated and complex vertical functionality
 - testing futuristic ideas
- Cons?

MEDIUM-FIDELITY PROTOTYPES

- Requires prototyping with a computer
- Engaging for end users
- Simulate some but not all features of the interface (interactive)
- Can test more subtle design issues
- Pitfalls
 - User's reactions often "in the small"
 - Blinds people to major representational flaws because of a tendency to focus on more minor details
 - Users reluctant to change/challenge designer

DIFFERENCE BETWEEN “LOW” AND “MEDIUM” ?

- Less clear than it used to be...
- Many powerful tools that...
 - make it very easy (a low-fi trait) to generate mockups
 - look real and are at least somewhat interactive (usually a “medium fidelity” trait)
 - e.g.: balsamiq, axure – low or medium; usually not high

MANY DIMENSIONS OF “FIDELITY”

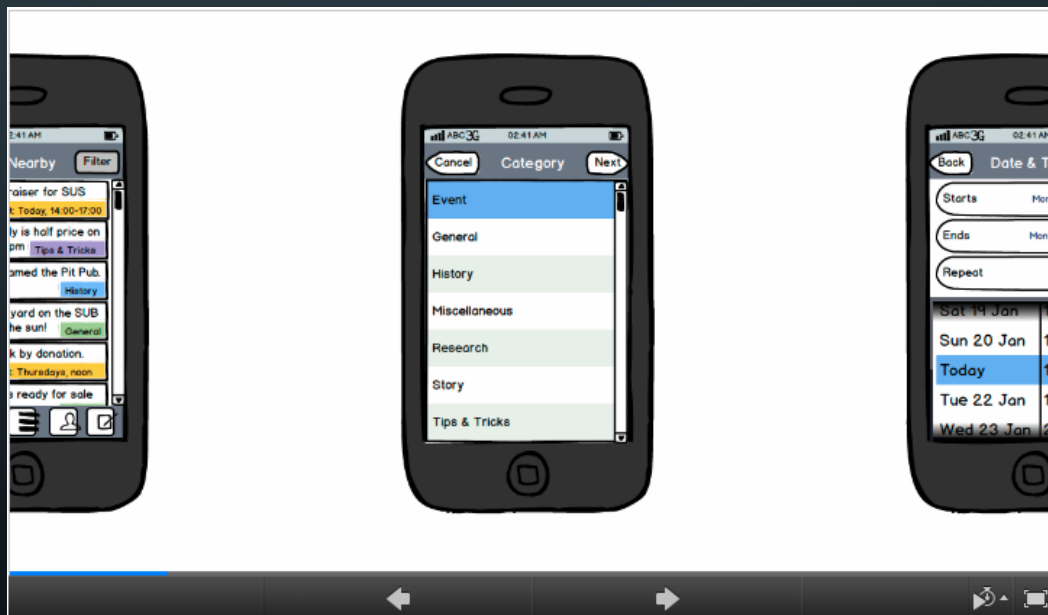
- What are ways a prototype can be ‘true to life’?
 - Visual realism: how real it looks. polish, graphic imagery
 - Physical realism: shape and form for 3D objects; feel
 - Scope: how many functions included; horizontal vs. vertical
 - Functionality: what actually works? e.g. web app: links live?
 - Data: operates on real vs faked data
 - Autonomy: operates alone vs requires “supervision”
 - Platform: interim vs final implementation

IMPORTANT LESSONS:

- It is **COMPLICATED** (slow, expensive) to prototype multiple dimensions at once.
→ so don't. Instead: modularity of prototyping.
- Each prototyping tool has strengths and weaknesses
 - May be better (more efficient and capable) for some of these prototyping dimensions than others.
→ you may need multiple tools throughout your design's life cycle.

BALSAMIQ: LOW TO MEDIUM

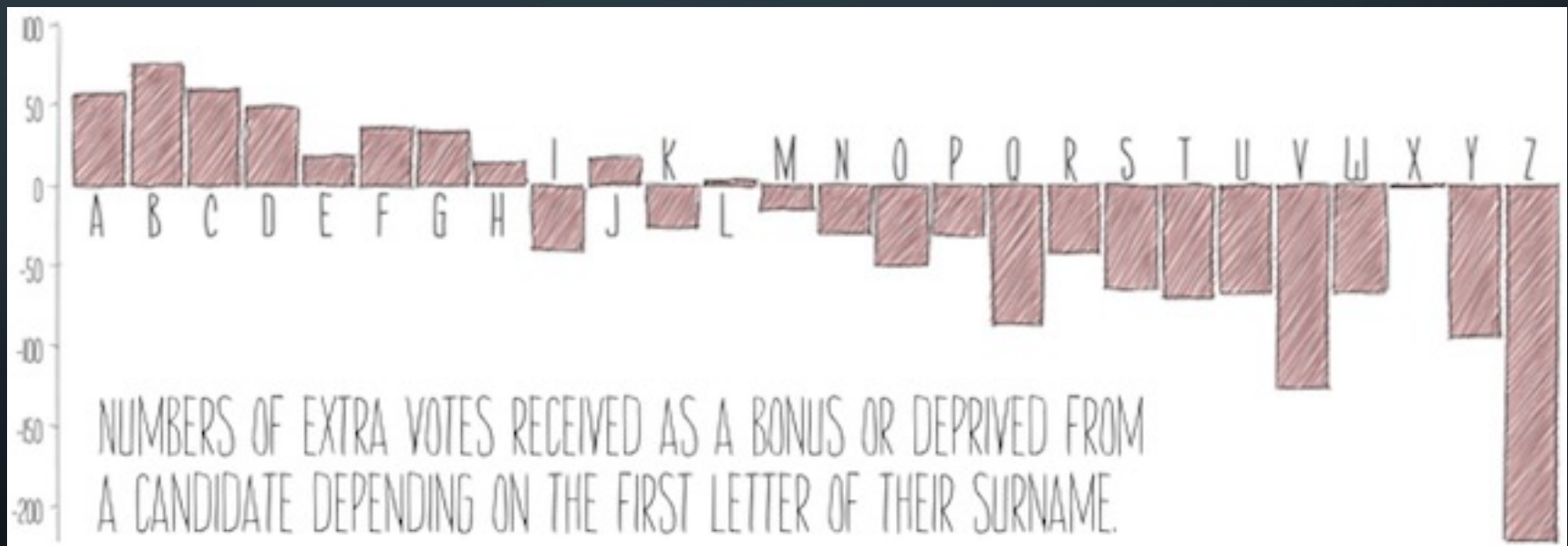
- Quickly mock up images and hyperlinked interactivity.
- But - real functionality difficult.
- <https://balsamiq.com/> (good ~1 min intro video)



MAKE MEDIUM-FIDELITY MOCKUPS LOOK LOW-FI

- this graphic is generated from code (processing).

<http://www.gicentre.org/handy/>



DIFFERENCE BETWEEN MED & HIGH-FIDELITY PROTOTYPES

- Increasing in completeness and detail:
 - More aspects being prototyped at same time
 - Higher degree of functionality
 - Higher degree of polish...
- Fidelity is a spectrum
- Not always a firm line between low/med or med/hi

THE SITUATION TODAY FOR PROTOTYPING TOOLS (VS. DEVELOPING ON FINAL PLATFORM)

- For simple prototyping.
 - Figma, Balsamiq, Axure, HTML, Powerpoint
- Advanced UIs still require (scripting) language + libraries
 - HTML + javascript
 - Tool Command Language/Tool Kit (TCL/TK)
 - Python
 - Processing (Java based, but way more accessible; good for sketching, no good for larger code projects)
 - still a need for more powerful programming languages
 - *Available tools change every year*

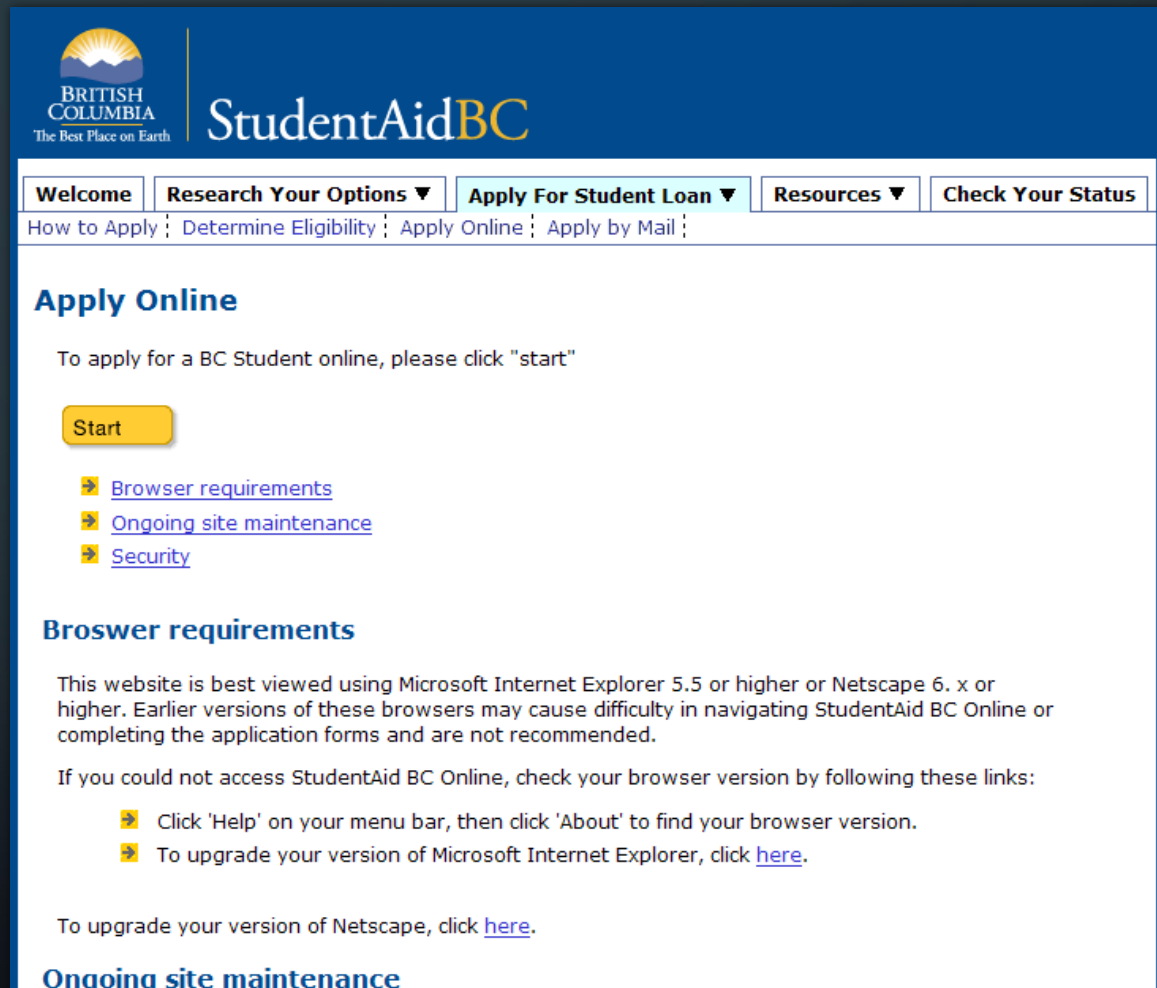
FUNCTIONALITY CHECKLIST

- How **realistic** does the prototype need to be? How important is **appearance**?
- Is the scope **horizontal** or **vertical**?
- How **functional** is it? what has to work vs. what can be faked?
- Is the data **real** or **faked**?
- Can it be used **alone**? or is use **supervised**?
- Is the platform **final** or **interim**?

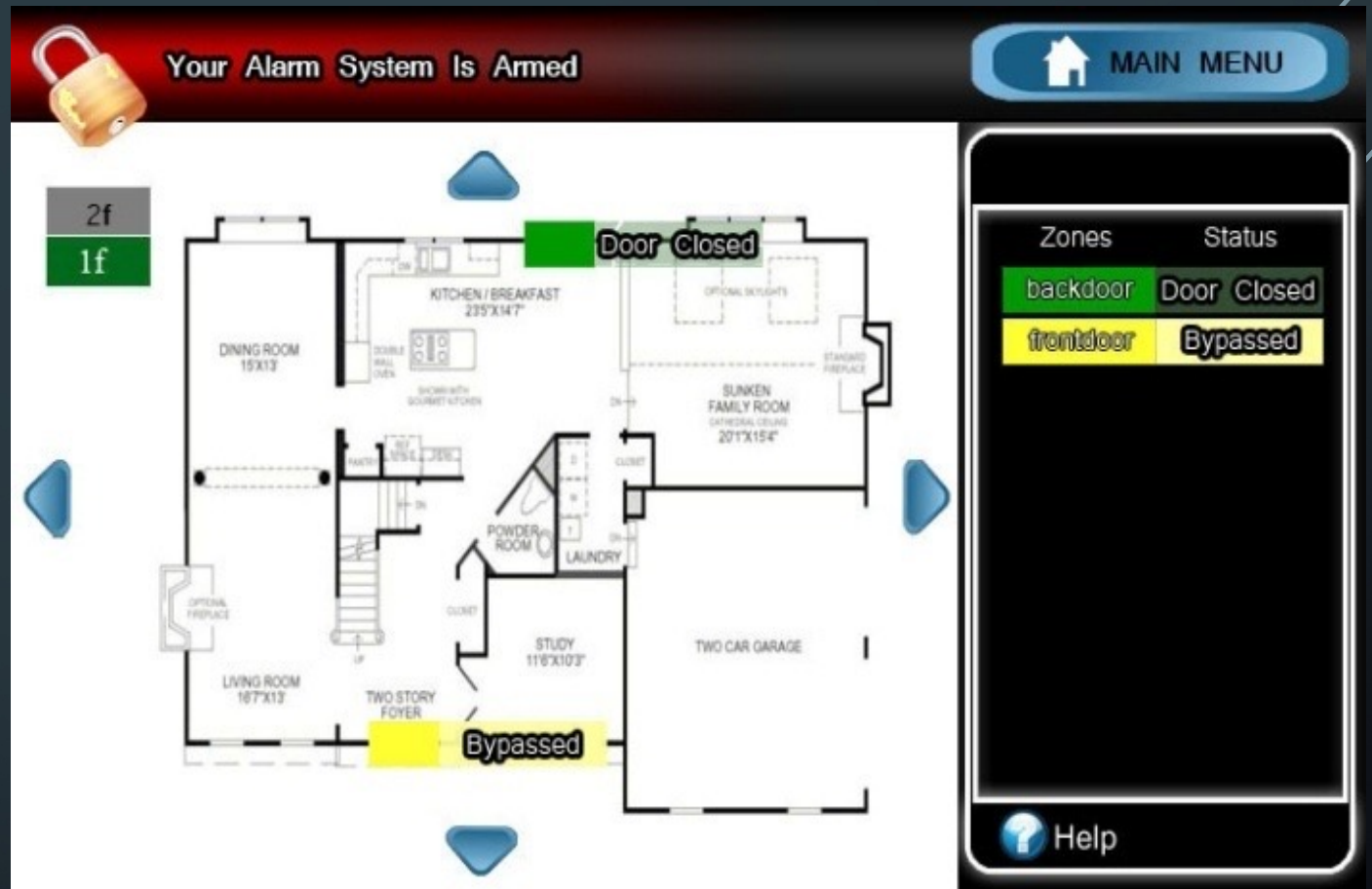
UBC STUDENT AID – HTML PAST CPSC344 PROJECT

HTML:

- final platform didn't need to be glitzy
- easy to copy existing text, look and feel
- then alter everything



HOME ALARM SYSTEM



flash:

- product for the home
- needed to gauge reactions to having it in one's house
- imagery + graphic resolution critical

E-READER & NOTE-TAKING TOOL

Hybrid View:

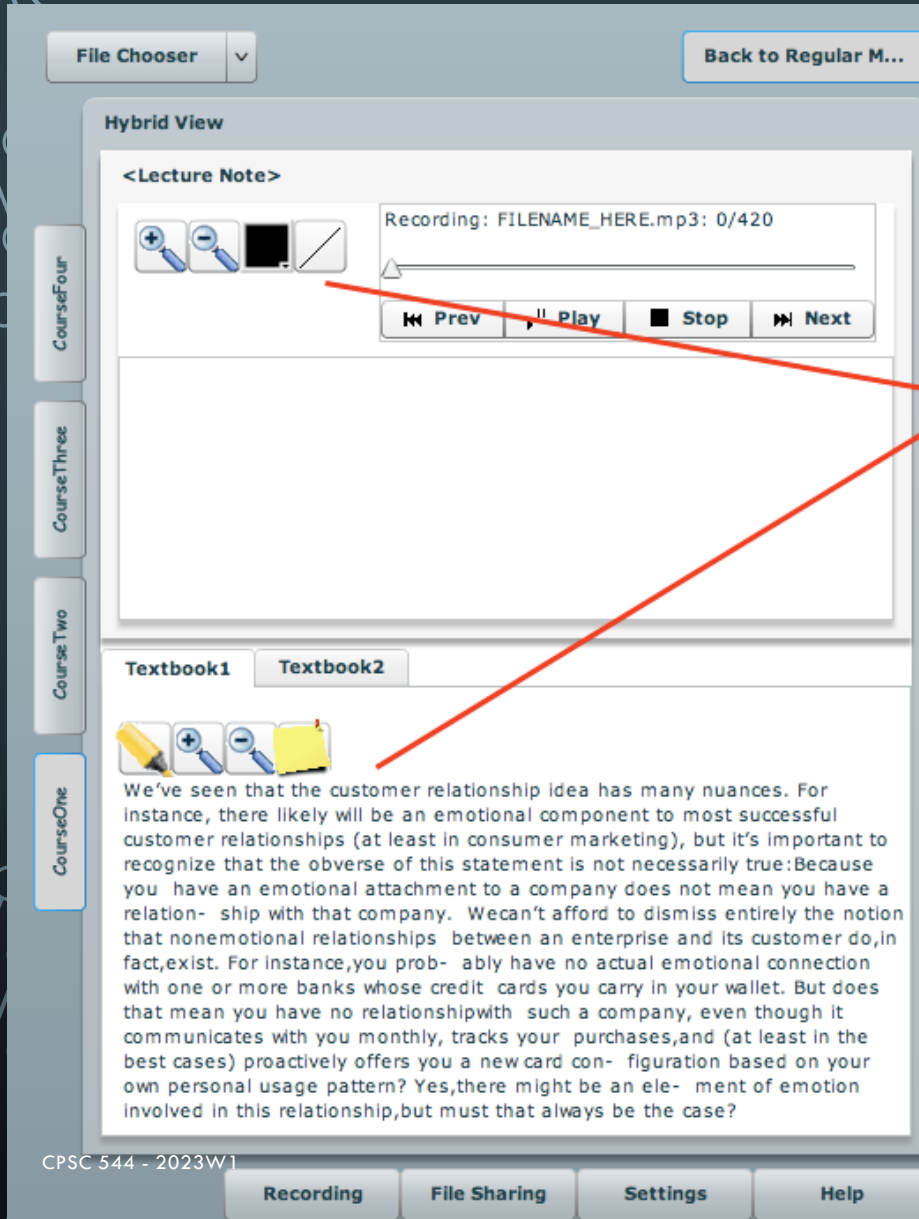
Split views for displaying two files simultaneously.

References: Can make hyperlink references between content

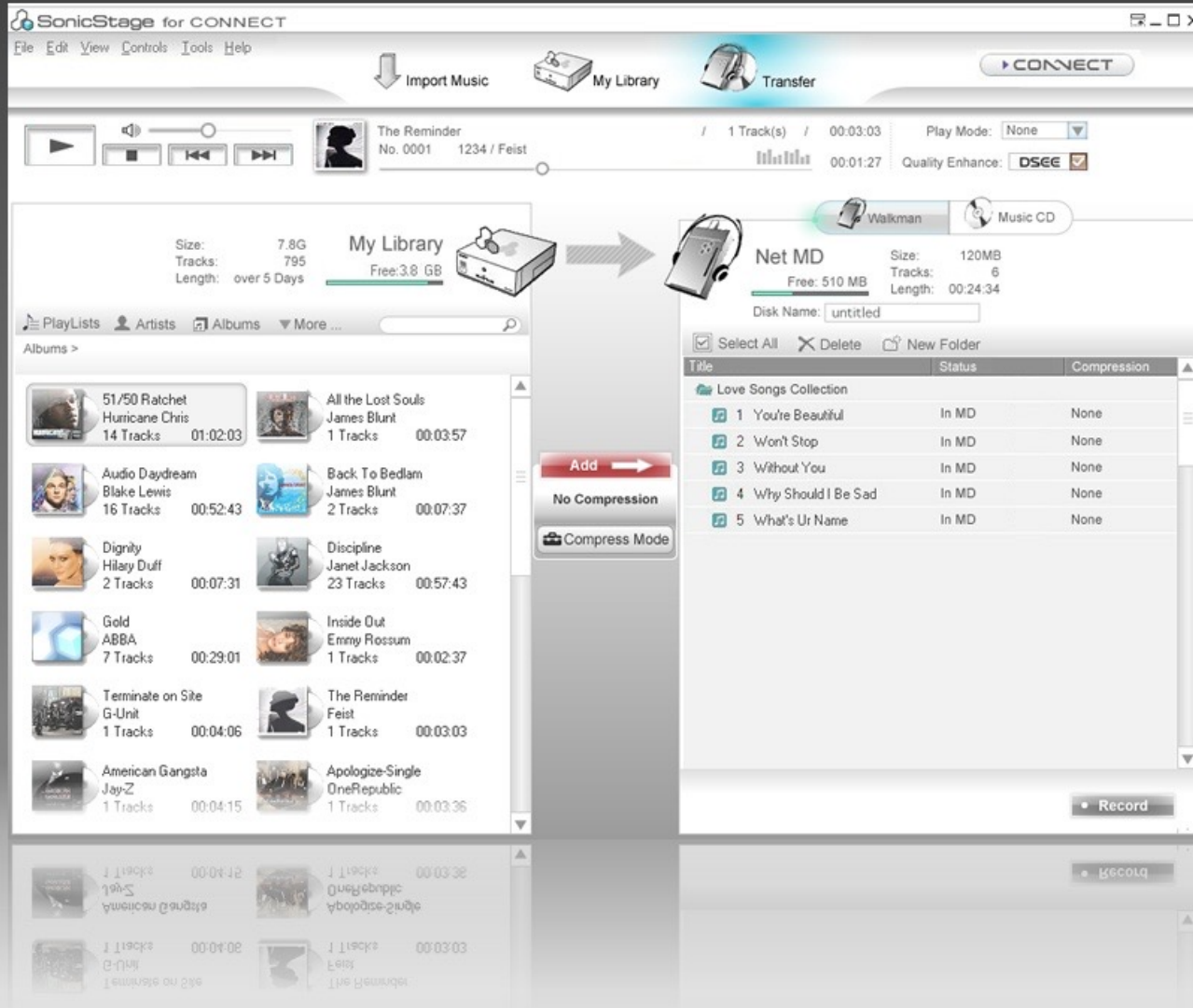
All controls are preserved

Flex:

- needed to test how well the concept worked for actually taking notes in lecture
- highly functional
- detailed vertical



SONIC STAGE MUSIC SYNCHRONIZATION TOOL



flash with
imported
photoshop images

Features scanned,
hand-drawn
sketches

Aesthetics were
important

HOW DO YOU KNOW WHEN YOU HAVE – OR NEED – A HIGH-FI PROTOTYPE?

- Scope is complete (horizontal and vertical)
- Prototype can be tested in just about every way:
 - performance
 - subjective and cognitive analysis
 - more realistic scenarios
 - in field
- Feels like time to switch to final development platform
- Design is becoming rigid and finalized

SUMMARY

Low Fidelity

- cheap
- easy to build lots
- facilitate communication
- gross design (layout)
- market requirements
- proof-of-concept
- limited error checking
- hard to get to code
- facilitator driven
- limited functionality

High Fidelity

- full functionality
- interactive
- user-driven
- exploration and testing
- final look & feel
- provides specs
- marketing tool
- expensive
- time consuming
- inefficient proof-of-concept
- poor for req'mts gathering
- hard to discard