

CS-639 — Interaction Design Studio

Redesign with Intelligence | In-
Class Exercise

Professor Bilge Mutlu

Today's Exercise

Design two versions of the same interface — side by side.

- **Left:** Traditional design (user controls everything, static structure)
- **Right:** Intelligence as core element (the design itself is shaped by intelligence)

This is not about adding P-tags to a finished design. It's about designing an interface that would be

Step 1: Choose Your Domain

Pick a domain — your A1 app, a class exercise from Weeks 2–6, or a simple example (to-do list, email, calendar, music player).

Pick one key task with 3–5 steps (onboarding, search, checkout, creating content, planning).

If you use your A1 domain, this is a separate exercise — not part of A1.

Step 1b: Choose One Material Property

Pick the property that could most transform your task:

Property	What It Enables
Adaptivity	System learns and changes over time
Initiative	System proposes actions proactively
Inference	System understands implicit intent
Delegation	User states goals, system handles steps
Collaboration	Human and AI refine together
Contextuality	System adapts to situation

Design with **one** property. Depth over breadth.

Step 2: Design Both Versions

Traditional version (left side):

Sketch 2–4 screens applying your Week 1–6 fundamentals — hierarchy, grid, flow, navigation. This should be solid work, not a throwaway.

Intelligent version (right side):

Redesign the same task with your property as a core element. Ask:

- What screens change, appear, or disappear?
- Where does the system act instead of the user?
- Where does control transfer between user and system?

The right side should be a design that **wouldn't make sense without intelligence.**

What Good Looks Like

Strong submissions:

- Two visibly different designs
 - not the same screens with/without labels
- Specific, concrete intelligence behavior (not "AI helps here")
- Clear indication of who does what at each step
- Fundamentals still strong on both sides

Common pitfalls:

- Right side is just left side with annotations
- Vague intelligence ("system suggests something useful")
- Intelligence that doesn't connect to the chosen property
- Trying to use all six properties instead of going deep on one

Example: To-Do List with Delegation

Traditional (left):

- User opens app → sees list
- User taps "+" → types task
- User sets due date manually
- User assigns priority manually
- User checks off tasks one by one

5 explicit user actions per task

With delegation as core (right):

- User opens app → sees today's priorities (system-organized)
- User says "Plan my week" → system drafts a schedule
- System shows proposed plan → user adjusts
- Completed tasks auto-archive; recurring tasks auto-regenerate
- "Add groceries" → system infers items from past lists

User states goals; system handles steps

Get Started

1. **5 min:** Choose domain, task, and property
2. **15 min:** Sketch traditional version (left)
3. **15 min:** Sketch intelligent version (right)
4. **Remaining:** Annotate differences

Due Friday morning on Canvas — photo of both versions side-by-side. Friday in class: critique.

We'll explore: "Is your right side a different design, or the same design with annotations?"