Interactive High-fi Prototype

Checkpoint: Start of Thu/Fri studio (Nov 30/Dec 1)
Final: Start of Thu/Fri studio (Dec 7-8)

Goals
Learn how to build high-fidelity, interactive prototypes with code. Understand the tradeoffs and platform constraints compared to low-fi and med-fi prototyping.

Assignment Overview
At the checkpoint in studio, you will present your high-fi prototype progress and plans. By the final due date, your prototype should operate to feel realistic to a usability test participant. You will:

1. **Use the results of the heuristic evaluation to revise your UI.** First, fix all heuristic violations severity level 3 or 4. If your team disagrees with the evaluators and wishes to disregard a violation, justify why. Also address any other level 1 and 2 violations that are easy to fix. If necessary, you can develop new task flows, but Slack your CA first to discuss such changes. Larger revisions to the design should be iterated on in your med-fi tool first while smaller changes can go straight into code in your high-fi.

2. **Develop your prototype to be sufficient for a usability test.** The underlying functionality does not have to work completely, but a participant should be able to complete all your tasks. Unlike your med-fi prototype, we want your high-fi to look and feel like a real app. Simulating a realistic experience is more important than backend computation/scalability.

Checkpoint Presentation Guidelines
One team member who has not yet presented will present the checkpoint in studio. For teams of 3, someone will present a second time and the higher of their 2 grades will be counted. There will be 10 minutes for the presentation and 6 minutes for questions and feedback. The time spent on the demo will not count toward the 10 minute limit; however, it should be brief (90 sec. max).

Required Content
1. Project title & value prop
2. Team
3. Problem/solution overview
4. Talk outline
5. Heuristic evaluation results
   a. High-level summary of results
   b. Additional details on severity 3 and 4 violations
6. All UI and product revisions
   a. Before and after shots
   b. Rationale for changes
c. Progress towards usability goals made by the revised designs

7. Prototype implementation status – **we expect at least 1 task to be implemented**
   a. Framework/tools being used
   b. Implemented features
   c. Unimplemented features & plans to finish
   d. Wizard of Oz techniques
   e. Hard-coded aspects
   f. Issues/Questions: Anything you are unsure of how to do?

8. Demo of prototype so far (live or recorded)

**Final Prototype Expectations**
See the [class resources document](https://hci.stanford.edu/courses/cs147/2023/au/) for helpful tips, tricks, and tools. Prototype should:

- Implement the task flows that you devised
- Properly account for the size, resolution, colors, standard widgets/controls, and other attributes of your target platform
- Apply good and consistent visual design principles
- Be more functional than your med-fi in Assignment 6
- Simulate the real experience; however, underlying functionality does not have to be fully implemented. For example, applications requiring large databases of information or live social networks can instead have a sufficient amount of hard-coded data
- Focus on developing only what is essential to simulate a realistic user experience

**README Guidelines**
Brief document accompanying the final prototype. Provide helpful context for anyone (including those outside of Stanford) to understand the purpose/use cases of the prototype. Then describe installation requirements, operating instructions, and any limitations in the implementation.

**Deliverables**
Upload deliverables to a subdirectory titled “Assignment 8” in your team’s Google Drive folder. You should also add relevant links to your website.

1. **Checkpoint presentation**
   Google Slides deck presented by 1 team member during studio week 8. At least one of your tasks should be implemented by this point.

2. **Final prototype**
   Must be accessible from your website and ideally executable without installing any additional software. If this is not possible for your group, make arrangements with your CA before your last studio session.
3. README file
   PDF linked on website. Google Doc in your team’s Google Drive folder.

Examples
**Note: this assignment spec has undergone some changes, but much of the quality of the work in these examples still stands**

GardenShare: on website
Dishcovery: Checkpoint Presentation, README, Demo Video
Off: Checkpoint Presentation, README
Fluantly: Checkpoint Presentation, README
Sprout: Checkpoint Presentation, README, Demo Video
Reclaim: Checkpoint Presentation, README, Demo Video
Pebble: Checkpoint Presentation, README, Demo Video

Grading Criteria
Grading will be broken into 3 components: 1) the individual grade of the presenter, 2) a group grade for the checkpoint presentation content, and 3) a group grade for the final prototype.

Individual Presenter Checkpoint Grade (100 pts)
   ___ Well-designed slides; visual aids are aesthetic and effective
   ___ Cover required scope (pre-demo) within 10 minutes
   ___ Engages with the audience and isn’t reading from a script
   ___ Projects voice well and communicates clearly

Group Checkpoint Presentation Grade (100 pts)
Heuristic evaluation results (20)
   ___ Summarizes high-level results and key takeaways
   ___ Details which major violations were addressed, which were not, and why

Revised UI design (30)
   ___ Changes are explained clearly and follow from sound reasoning and HE results
   ___ Changes make the design better
   ___ Interface is novel/different from other known products

Prototype (50)
   ___ Outlines tools/framework, Wizard of Oz items, and hard-coded aspects
   ___ Implemented task is visually appealing and follows standards for the platform
   ___ Clear what has been implemented; sufficient plan to implement what is missing
   ___ On track to be done by the end of the quarter

CS 147 Autumn 2023 website
https://hci.stanford.edu/courses/cs147/2023/au/
Final Prototype Grade (100 pts)

Prototype (90)

- Implemented fixes from heuristic evaluation results outlined in A8 checkpoint
- User can easily accomplish the tasks
- Limitations of the med-fi are addressed; feels realistic for a usability test
- Appropriate trade-offs made between functionality and design completeness
- Constraints and styles from the target platform are considered
- Prototype exhibits strong visual design
- Executable with no additional software required (except by permission)

README (10)

- Includes necessary details and context for prototype setup and operation
- Outlines limitations, Wizard of Oz items, and hard-coded aspects