

Interactive High-fi Prototype

Checkpoint: Start of Thu/Fri Week 9 studio (Nov 20-21)

Final: Start of Thu/Fri Week 10 studio (Dec 4-5)

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, 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 major 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 independently 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.

Expected Content

1. Project title & [value prop](#)
2. Team
3. Problem/solution overview
4. Presentation outline
5. Heuristic evaluation results
 - a. High-level summary of results: total violations, severity breakdown, most common violations, total number of UI revisions needed to be made based on the major severity 3 & 4 violations as well as any severity 1 & 2 violations deemed necessary

- i. Consider communicating your results through a visual representation of these statistics and reflecting on how these high-level trends informed your approach to addressing the violations
 - b. Discuss and justify changes to address all severity 3 and 4 violations
 - i. Violations should be clearly connected to the changes made in part 6. Consider using visuals to explain specific issues.
6. All major UI and product revisions
 - a. Before and after shots
 - b. Rationale for changes
 - c. Progress towards *each* usability goal made by the revised designs. Discuss the impact of the progress made on design values, project next steps, and/or overall success metrics.
7. Prototype implementation status – *we expect at least 1 task to be implemented*
 - a. Framework/tools being used and what each is being used for
 - b. (if applicable) How AI was used in implementation and visual design: explain exactly where AI was used with justification of why
 - c. Implemented features
 - d. Unimplemented features & ***plans to finish with an estimated timeline***
 - e. Wizard of Oz techniques (& justify why Wizard of Oz was necessary)
 - f. Hard-coded aspects (& justify why hard coding was necessary)
 - g. Issues/Questions: Anything you are unsure of how to do?
 - h. Discussion on plan for finishing
 - i. Consider including a discussion of how your team will prioritize the remaining work and who will be responsible for what
8. Demo of prototype so far (live or recorded)

Final Prototype Expectations

See the [class resources document](#) for helpful tips, tricks, and tools. If you are using **React Native**, see [this document](#). If you are implementing a native **iOS app**, see [these setup instructions](#).

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 ***far 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, live social networks, or complex AI can instead have a sufficient amount of hard-coded data/wizard of oz behavior
- 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. The following components should be included in your README:

1. How to set up the prototype for testing
2. Context for the prototype: when is the app used? What should a user be able to accomplish? Be sure to include the **tasks** represented in your hi-fi prototype
3. Summary of limitations
4. Summary of accessibility considerations & limitations

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. The demo videos attached below are the teams' finalized demo videos submitted for A10. Your demo is not expected to be at that level at this point, but the videos demonstrate the fidelity we are looking for.*

superbloom (au25): [Checkpoint Presentation](#), [README](#), [Demo Video](#)

FitCast (au23): [Checkpoint Presentation](#), [README](#), [Demo Video](#)

Coco (wi23): [Checkpoint Presentation](#), [README](#), [Demo Video](#)

Arbor (au23) : [Checkpoint Presentation](#), [README](#), [Demo Video](#)

Nurture (wi22): [Checkpoint Presentation](#), [README](#), [Demo Video](#)

Sprout (wi21): [Checkpoint Presentation](#), [README](#), [Demo Video](#)

Reclaim (wi21): [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 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 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 so far; sufficient plan to implement what is missing
- On track to be done by the end of the quarter

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 and interaction design

README (10)

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