

Interactive Prototype (Group)

Due: Tuesday, March 12, 2013

Goals

The goal of this assignment is to learn how to build medium-fidelity prototypes of user interface ideas using an interactive user interface design tool. You will revise your user interface ideas based on the low-fi eval and our feedback. Then, you will use interactive tools to build a medium-fidelity prototype of the design.

Interface Redesign

Use the results of your low-fi prototype tests & teaching staff feedback to design a revised interface. Develop new and/or revised scenarios for your tasks by storyboarding your ideas. The tasks that most of you used in the low-fi assignment should be sufficient for this, but some may have been ***simple or partial tasks*** that did not adequately cover your proposed functionality or your functionality may have changed based on testing or our feedback. Make sure to ***revise those tasks***. If you are changing your tasks, email us to present your new tasks, design ideas, and scenario storyboards for discussion.

Prototyping

You will use a prototyping tool to create an interactive prototype of your application. For most applications, we would like you to use a design tool that targets mobile platforms. We have selected Justinmind Prototyper (use Pro Edition v5– free 30 day trial) and Axure (free 30 day trial), as the preferred tools (Justinmind has iPhone/iPad/Android widgets, rich interactivity support, and export to HTML that makes it easier to test – Axure is more featureful, but might take longer to learn/use). If you have experience with one of these, you should use it. If there is another tool you think would work better for your project, please contact Prof. Landay and discuss it first.

Your prototype should “implement” the three scenarios that you developed for your tasks. You should now be making your design work with the actual target constraints (e.g., size of screen, text size, and built-in controls/widgets) of a real mobile platform (e.g., iPhone, Android, or iPad).

The ***underlying functionality does not have to be fully implemented***. For example, applications requiring large databases of information can instead have a sufficient number of hard-coded data points for supporting the three tasks.

You have a short period of time to complete this prototype, so you should focus on showing only what is essential. You will likely have to make some difficult choices!

Deliverables

1. Prototype

Your prototype must be accessible and/or executable by everyone in the class from your team web site (and should reside on that site). It must be accompanied by a README file that describes the tool that it runs with and operating instructions, including any limitations in the current implementation. **If this is not working on the due date, you will get a zero on this assignment.**

2. Report

You will submit a report of no more than **three pages** of text on your team site (**images free and required**). Make sure the entire report is on our server, not just a link to somewhere else.

Report

The report should follow this outline with separate sections for the top-level items.

1. **Problem and solution overview (1 paragraph)**
2. **Tasks (1/2 page)**
 - 3 representative tasks to test your interface (labeled simple, medium, complex)
 - Note any changes that were made from prior tasks
3. **Revised interface design (1 page plus screenshots)**
 - Changes as a result of low-fi testing and rationale behind the changes (refer to screenshots)
 - Scenarios for 3 tasks
 - Storyboards of scenarios (annotated screenshots, web pages, etc.)
4. **Prototype overview (1 page)**
 - Tools
 - What did you use?
 - How the tool helped
 - How the tool did not help
 - What was left out and why
 - Any wizard of Oz techniques required to make it work
 - Hand-coded features and why required
5. **Prototype screenshots (as many as needed)**

Writing Guidelines / Grading Criteria

Here is the grading criteria for the report and prototype (100 pts total):

Design (40 Points)

- Tasks
 - Do the tasks cover the interesting features of the project?
 - Do the tasks have an appropriate difficulty/complexity specified?
 - Do the tasks altogether form a compelling story for the project?
- Changes
 - Were **appropriate changes** made to address the important problems discovered during the low-fi testing?
 - Is there a clear **rationale for the changes**?
 - Are these changes **well illustrated with screenshots**?
 - Are the three scenarios clear, labeled, and mapped 1 to 1 from the tasks?
- Transition from low-fi to interactive prototype
 - Were some of the **limitations of the low-fi addressed**?
 - Were appropriate **constraints from the final target platform** considered?
 - Were any non-standard interactions described and justified?

Prototype (30 pts)

- Is the prototype **accessible and fully working** for the 3 tasks?
- Can users complete the three tasks with the prototype?
- Were **appropriate tradeoffs** made between functionality and completeness?
- Does the README file summarize these limitations and any other details needed?

Report (30 pts)

- Writing
 - Does the report cover all the topics in the outline?
 - Does the organization follow the outline?
 - Are sub-sections used for easy scanning of important parts?
- Are the **limitations and tradeoffs described and justified** in the report?
- Screenshots and Storyboards
 - Are important figures referenced and placed inline with the text? *
 - Is there a complete set of screenshots in the appendix?
 - Are they clearly annotated?

* Use [Relevance-Enhanced Image Reduction](#) to create effective thumbnail images.