

## Interactive Medium-fi Prototype #1 (Group)

***Due: Friday, October 31, 2014 (talk due in Studio / write-up due at 6 PM)***

### 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 Marvel, InVision, and proto.io. If there is another tool (e.g., Justinmind Prototyper – which can also support Google Glass) you think would work better for your project due to capabilities or expertise on your team, please contact your CA and 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, iPad, or watch, or Google Glass). Many of the limitations and tradeoffs you made for the low-fidelity prototype should be addressed by this medium fidelity prototype.

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. Focus on UI, not underlying implementation. 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 a web 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 **2 pages** of text (**images free and required**).

### 3. Presentation

One member of your team will present your project in class during a nine-minute presentation. See the grading guidelines for information on how to structure your talk. Practice in advance! You must **make the slides available for download on your (eventual) web site**. Look at the final presentations from the [UW version of this class in 2013](#) to see what good slides look like

## 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)

- o 3 representative tasks to test your interface (labeled simple, medium, complex)
- o Note any changes that were made from prior tasks

### 3. Revised interface design (3/4 page plus screenshots)

- o Changes as a result of low-fi testing and rationale behind the changes (refer to screenshots)
- o Scenarios for 3 tasks
  - Storyboards of scenarios (annotated screenshots, web pages, etc.)

### 4. Prototype overview (1/2 page)

- o Tools
  - What did you use?
  - How the tool helped
  - How the tool did not help
- o Limitations/tradeoffs of the current prototype (What was left out and why)
- o Any wizard of Oz techniques required to make it work
- o 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?
  - Are the tasks complete, real tasks rather than features or sub-tasks?
  - 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 between functionality and completeness described and justified** in the report?
- Are the tools limitations & weaknesses described? Wizard of Oz techniques?
- 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.

## Presentation Guidelines

The presentation grading will be broken into two components: the individual grade of the presenter and a group grade for the presentation of the medium-fi prototype and design changes. Note that you should use images liberally and try to keep the text on the slides brief (and use large fonts – **no less than 20 pt anywhere**). The grades for each of these components are explained in more detail below.

### Presenter's grades (NAME: \_\_\_\_\_)

- Suggested Organization
  - \_\_\_ Overview of talk (1 slide) – don't read this, ***tell it like a story***
  - \_\_\_ 3 representative tasks noting key changes (1-3 slides)
  - \_\_\_ Revised UI design & rationale (1-3 slides – mainly images w/ captions)
  - \_\_\_ 3 scenarios shown carrying out each task w medium-fi prototype (use screen shots and/or live demo)
  - \_\_\_ Tools Used (what worked, what didn't work, WoZ techniques) (1-3 slides)
  - \_\_\_ Summary of talk (1 slide)
- Presentation
  - \_\_\_ Use slides. Ensure that the presentation shows appropriate preparation, and that visual aids are effective, properly prepared, and properly employed. Make sure that people at the back of the room can see your slides.
  - \_\_\_ Cover the required scope within the 9 minute time period (there will be 1 extra minute for questions). **Practice & time your presentation in advance. We will cut you off if you go over and you will not be able to gain points for missed material.**
  - \_\_\_ Ensure the presenter makes eye contact and projects well.

### Group grade (GROUP NAME: \_\_\_\_\_)

- Representative Tasks & Scenarios
  - \_\_\_ Did they provide coverage of the functionality?
  - \_\_\_ Where the tasks too easy or too hard?
  - \_\_\_ Where the tasks real, complete tasks or fragmented?
- Medium-fi Prototype
  - \_\_\_ Was the interface novel and creative?
  - \_\_\_ Was it appropriate for the supported tasks?
  - \_\_\_ Did UI changes follow from sound reasoning/data from low-fi testing?
  - \_\_\_ Did interface fit the target platform's constraints?
- Tools
  - \_\_\_ Were appropriate tools used & explained?
  - \_\_\_ Were tradeoffs to using the tool discussed?
  - \_\_\_ If Wizard of Oz techniques used, were they clearly explained?