User Interface Design Assignment

Each GR section adds to the final report.

GR1: Project Proposal and Analysis

The heart of this course is a semester-long project, in which you will design, implement, and evaluate a user interface. User interface design is an iterative process, so you will build your UI not just once, but three times, as successively higher-fidelity and more complete prototypes. In order to have time for these iterations, we need to get started on the project as early as possible.

Each project group should consist of **6 - 8 Students**.

Choosing a Project

Your Project may be a web, desktop, or mobile interface. If you choose to do a mobile application, note that it must at least be possible to simulate your project on the web or the desktop, since one of your prototypes will be such a simulation that you have to give to your classmates to evaluate.

Analysis

After you choose a project, you will start your design by doing the following:

User analysis. [10 Marks]

Identify the characteristics of your user population, as we discussed in the lecture (review chapter10). If you have multiple user classes, identify each one.

Task analysis. [10 Marks]

Determine the tasks of the problem you've chosen, analyze their characteristics, and answer the general questions about tasks we asked in lecture. Think about other questions you should ask that might be relevant to your particular domain. You should find and analyze at least 3 high-level tasks. For example, in a recipe site, the most central, interesting tasks might be editing a recipe, finding a recipe and using a recipe (to actually cook). If you can't find 3 interesting tasks, then your problem may be too small to serve as a good project, and you should rethink it. Every task should have a goal and subtasks. Some tasks may also need preconditions, exceptions, time constraints, and frequency of use.

Domain analysis. [10 Marks]

Determine the important entities and relationships of your problem domain, and show them in a diagram (a problem object model or entity-relationship diagram), use your knowledge form System development course units. Include multiplicities where important. Include text that defines entities or relations that aren't obvious. Use cases can also be used.

To gather information for the user and task analysis, you must talk with at least 3 representative users who face the problem you are tackling (at least 1 from each user class, if you have multiple user classes). If possible, observe them dealing with the problem in their real work environment. When you write up your analysis, you must give us evidence that you

interviewed and observed people, but don't provide a narrative of these sessions. Instead, offer your conclusions, and justify them when you can by referring to observations. For example, "grocery shoppers may be distracted by children; one mother was repeatedly harassed by her son to buy some candy." Also, don't identify the users you interviewed by name, unless you get their permission to do so.

What to Hand In
For this assignment your project should include the following parts:

Group members
A list of your group members.
Problem statement
Briefly state the problem(s) that your project will seek to solve. Take the user's point of view. Consider what the user's goals are, and what obstacles lie in the way.

GR1 Analysis
Write up your user analysis, task analysis, and domain analysis clearly, concisely, and completely.

GR2: Designs
In this group assignment, you will start designing your semester project.

Designs
Scenario. Write a scenario that involves all three of the tasks you identified in GR1. Where your task descriptions in task analysis were abstract, your scenario should be concrete, complete with imaginary users' names and imaginary details.

Storyboard designs. Generate **three different preliminary designs** for your user interface. Explain each design and include a storyboard showing how it works for your scenario. The storyboard should combine words with sketches showing how the interface would look over the course of the scenario. After the storyboard, you should have an analysis that considers the design's good and bad points for learnability, visibility, efficiency, and error prevention.

Your designs will be judged on how well you've described them, how well you analyze them, and how diverse they are. Three designs that differ only in small ways will not receive much credit.

Take time to brainstorm a variety of different interface designs, sketching them by hand on paper or a whiteboard. You should play with many more than three designs, but we only require you to record three

When you draw your sketches, don't get bogged down in details like wording, graphical appearance, or layout. Keep things simple. Focus on the conceptual model you're trying to communicate to the user, and think about your task analysis: what the user needs to do and how they can do it. Putting too much time into designing low-level details is pointless if big things have to change on the next design iteration.

Hand-drawn sketches are preferred or use a mockup software. There are a number of ways to get hand-drawn sketches into your web page. You can draw on paper and use a scanner to convert it to electronic form. Make sure your sketches are readable, and crop them and size them appropriately so.

What to Hand In

Update your group's Project report so that it contains a section **GR2 Designs**, with subsections for:

i. Scenario **[10 Marks]**
ii. Designs (Storyboards) **[10 Marks]**

GR3: Computer Prototyping. [20 Marks]

In this group assignment, you will do the first computer-based implementation of your term project.

You may want to use a prototyping tool for this assignment -- such as an HTML editor if you're building a web application or any other, or a GUI builder if you're writing a desktop application. You don't necessarily have to throw this prototype away, so you can choose a tool that will produce code you can use in your final implementation. But you shouldn't get too attached to this prototype either, and be prepared to make radical changes or throw it away if evaluation reveals serious usability problems.

Prototype building.

Your computer prototype should be:

High fidelity in look.

Use this prototype to explore the graphic design of your final implementation. Lay out screens as you want them to appear in your final implementation. Make choices about colors, fonts, alignment, icons, and white space.

Medium fidelity in feel.

This prototype must run on a desktop computer with a mouse and a keyboard. For most projects, the web or desktop is the target setting, so your prototype will approach high fidelity in feel.

Low fidelity in breadth.

Your prototype should include every feature needed by your scenario from GR2.

Low fidelity in depth.

You can leave out most of your backend. Where system responses are needed, make them canned (i.e., always the same) or random. Consider using static images (pixel-model output that you created in a drawing program) in places where the final implementation would have to draw on the fly (stroke-model or component-model). **Use realistic data** in your canned displays, however -- in particular, data of a realistic scale. If you're building (say) an MP3 player and your prototype displays only three songs in the user's library, that's pretty unrealistic, and won't adequately test your UI design choices. Your domain analysis from GR1 should help you recall what was realistic.

Here are some issues you should not worry about in this prototype:

Window resizing.

When a window is resized, its layout has to adjust to respond. Don't worry about this for now. Determine a good default size for your windows and design a good layout for that size (using either automatic layout or even absolute positioning). Your final implementation probably should support window resizing, depending on your application, but you should decide how to make your interface's default look as good as possible, before worrying about variation.

Platform independence.

Even if your final implementation will run on multiple platforms -- different browsers, different operating systems -- your prototype doesn't need to look good or work well everywhere. Focus on one platform for now.

After you have finished your prototype, you will distributed to at least three of your classmates groups, who will do heuristic evaluations of it for assignment HW2 and give their reports back at the same time of handing in the coursework. Since your evaluators must be able to view and interact with your prototype, this puts some constraints on how you implement your prototype. It must run on a conventional desktop computer with a mouse, keyboard, and screen, running at least one of the common platforms at MUK (Windows, Linux). The prototype you give to your evaluators can be any of the following:

- **HTML.**

All three platforms have web browsers that support modern HTML and JavaScript (Firefox, Chrome, Safari, Opera, and Internet Explorer). You can require evaluators to use a particular web browser to ensure the correct appearance and operation of your prototype, as long as the browser is commonly installed on at least one of the three platforms.

- **Java JAR file or Flash SWF file.**

All three platforms can run Java or Flash.

- **Windows, Linux, or Mac executable. Find a way of transferring the code project to them**

You can assume that your evaluators can find the appropriate platform if necessary.

If you want to hand in your prototype in a format not listed here, check with the course staff first.

What to Hand In

Add a section called **GR3 Computer Prototype** to your project report. This section should provide instructions for accessing and starting up your computer prototype, both for your classmates who will be evaluating your prototype and for the staff who will be grading it.

- Specify the platform and software requirements for your prototype.
- Give brief, step-by-step instructions for starting up your prototype. For web sites, a hyperlink to the site is sufficient. For JAR files or executables, say how to start the program.
- Design a simple user guide for your prototype.
- Describe which parts of the prototype are shallow (incompletely implemented or canned), so that your evaluators know what should work and what shouldn't.
- Your prototype must remain frozen (no changes) and accessible at this location for two weeks after the due date.
- Your prototype should be downloadable as a single file. Package multiple files into a ZIP archive for easy downloading or handing in.
- Provide screen short while presenting the computer prototype.

Prototype Evaluation

a) Perform a Usability testing (Use any of the Usability testing techniques e.g. Laboratory testing etc.) on the prototype in GR3 to evaluate your final design. **[10 Marks]**
b) Using the DECIDE framework, evaluate another groups Prototype and develop a report. **[20 Marks]**

Hand in all the deliverables as one report

DEADLINE: 6th NOVEMBER 2018

Submit your reports to: mugramo@yahoo.com

Good Luck!!