

HUMAN COMPUTER INTERFACE (CSC440)

PRACTICAL GUIDE

Project teams may consist of 4-5 members. Ensure the names and matric number are prominently displayed and that the document is well-organized.

Here are some guidelines to help you develop your project proposal:

Your project must have a substantial user interface. A system that is simply static information or administers a form is not enough.

The user interface must be interactive. A system that simply displays a page of text or sequences through a series of pages would not be acceptable

The user interface should not exist already as a commercial product (you should design something novel).

Ideally, you should have access to three people who are representative of the end users of your system.

What to include

For every team project assignment you will need to include the followings:

1. Problem: describe the problem you chose
2. Target users: who will use this application?
3. Solution: Describe a possible solution to the problem, i.e., the interface that you envision, and how it will address the problem. You aren't absolutely committed to your solution, since you may find after building and evaluating some prototypes that a wholly different solution will work better.
4. Roles and Tasks.

You should indicate the roles each team member played on the project. The roles are *project manager, business analyst, architect, designer, etc.* For each team member, please include not only the title of the role, but also the tasks they accomplished. If you need to add roles to reflect the work you are doing, that's fine.

STAGES OF THE PROJECT

1. Task Analysis

In this team assignment, you will start the design of your term project by doing the following:

- **User analysis.** Identify the characteristics of your user population, as we discussed in lecture. If you have multiple user classes (types of stakeholders), identify each one. Create one or more personas for your primary stakeholders.
- **Task analysis.** 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 6 tasks. If you can't find that many tasks in your problem, try drilling down to more specific tasks, and consider exceptional and emergency tasks. Write "Task Scenarios" (for different representative users and contexts). At this stage you should be focused on the abstract steps of each task, and should not be thinking about what your interface will look like yet.
- **Usability requirements.** Draft usability requirements that make sense for your application, using your Essential Use Cases as standardized tasks.

2. Conceptual Design

you will continue the design of your term project, by converting your task scenarios and essential use cases into a conceptual design.

- **Use Cases.** Expand each of your Task Scenarios into a Use Case, including functions, links, objects, and constraints. Abstract these into a Content Diagram.
- **Metaphors.** Make a list of possible interaction metaphors for your interface. For each of your task scenarios list at least two options for interaction metaphors that encompass some or all of the task, why you think they are appropriate, and some of the implications of your choice. At this stage you should still be focused on the abstract steps of each task, including user input and system output actions, and should not be thinking about the details of your interface's appearance yet.

3. Design Sketches

continue the design of your team project by exploring possible design options, and sketching what your interface will look like.

Take a little time now to brainstorm a variety of different interface designs, taking into account your interaction metaphors. Then choose one that seems the most promising to focus on. When you draw your sketches, don't get bogged down in details like wording, icon appearance, or layout. Keep things simple. Focus on the 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 low-level details is pointless if big things have to change in the next design iteration.

- **Preliminary interface design.** A preliminary design consists of one or more sketched windows or dialog boxes, along with the menus and controls that the user manipulates.
- **Storyboards.** For each of your Use Cases, describe how your preliminary interface would be used to perform the task. Use rough sketches to illustrate how the interface would look at important points in the task.

4. Paper Prototyping

In this team assignment, you will do your first implementation of your term project, which will be a paper prototype. Your paper prototype should be able to handle at least 3 of your Concrete Use Cases.

5. Computer Prototyping

Here you will do the first computer-based implementation of your project.

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. Your prototype need not be pixel-for-pixel identical to your final implementation, however.
- **Medium fidelity in feel.** This prototype will run on a desktop computer with a mouse and a keyboard. Also, your prototype may not support some advanced interactions with high fidelity, such as drag & drop. That's OK. You can simulate these interactions with a little animation, or at least with a popup that describes in English what would happen.
- **Medium fidelity in breadth.** Your prototype should be able to handle at least the 3 scenarios you described in your task analysis. In addition, your prototype should include every major screen or dialog you expect to have in your final implementation.
- **Low fidelity in depth.** Don't implement any backend. Where system responses are needed, make them canned (i.e., always the same) or random. Write minimal code.

Startup instructions. Specify the platform and browser requirements for your prototype. Give any special instructions for starting it up.

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

Window resizing.

Platform independence. Focus on Windows for now.

6. Final Report and Presentation

Your team will give a 15 minute (max) presentation of your project. This talk should include the following:

- i. **Problem.** (2 min) What user problem are you trying to solve? Who are the users? What are their tasks?
- ii. **Demonstration.** (6 min) Demonstrate your design and implementation via a live demo of your system, working through one sample task. Discuss major design decisions. Run on YOUR computer to minimize compatibility issues. You should test with the projector before class starts.
- iii. **Evaluation.** (5 min) Discuss the major findings. This will be followed by a few minutes of Q&A during which time the next team will set up their computer.
- iv. **Summarize.** (1 min) What are the big take away? What did you learn? What would you have done differently?

We are on a very tight schedule, so if you are still talking at 10 minutes you will be cut off. There is no time to recover from technical difficulties, so if your demo is not working you will have to skip it.

7. Reflection:

Discuss what you learned over the course of the iterative design process. If you did it again, what would you have done differently? Focus in this part not on the specific design decisions of your project (which you already discussed in the Design section), but instead on the meta-level decisions about your design process: what features to prototype, what prototype techniques to use, and how to evaluate the results.

Important: Your grade is based as much on process as product. Provide evidence of the design alternatives you considered at each point in the process, and rationale for your design decisions. Your report should be 8 -10 pages long.