CMSC434: Introduction to Human-Computer Interaction
SAMPLE SOLUTIONS FOR EXAM #1

This solution set is by no means complete and extensive. This document's purpose is more to give you a general sense of the answers. Most of the answers required creativity, so it would be very difficult to come up with a superset of all of the answers. If you exhibited knowledge and came up with reasonable solutions, you most likely received credit.

If you have any questions about the grading of your exam, or would like a more extensive answer, please contact Adam (adamp@cs.umd.edu).

1. Describe three populations of users with special needs. For each of these populations, suggest three ways current interfaces could be improved to better serve them. **(18 points)**

Grading: 6 points total for each population, 2 points per way.

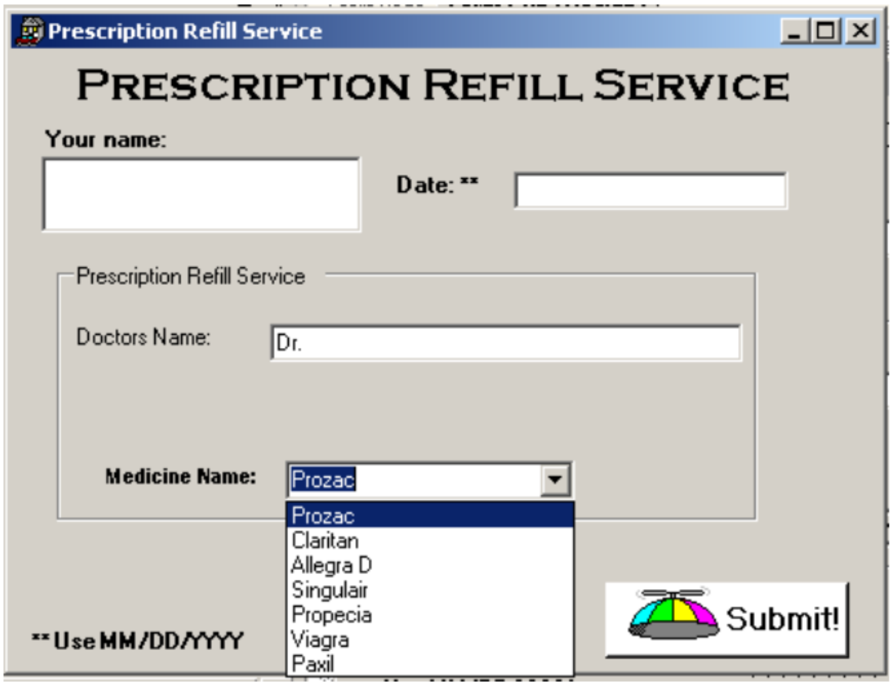
Here are some possible answers. Refer to Chapter 1 of the text for a complete discussion of user groups in need of universal usability.

Disabled users – Have audio that reads text back to them, Braille on keyboards or output, have keyboard prompts that can perform same tasks as direct manipulation, have text associated with any audio

Elderly user – enlarge font size and contrast, reduce the amount of typing, reduce short time memory load by remembering their passwords and other information, enlarge width of buttons and controls so they are more easily clicked

Children – colorful, attractive interface, minimize the syntax so its as easy to learn as possible, prevent errors and provide reversibility to encourage exploration of system, enlarge width of buttons and controls so they are more easily clicked, simpler vocabulary.

2. Examine the following interface:



a) Describe five things wrong with the interface on the left. **(10 points)**

Grading: Any reasonable answer would receive credit (2 points each mistake).

Some mistakes in the above interface are:
Text box size are not uniformed or aligned. The size of the box seems to have very little to do with the intended content. The list of medicines in the combo box has no apparent organization. The picture on the submit button only provides distraction, has seemingly nothing to do with the interface. The date box does not prevent errors – it has a label describing valid input that is extremely far away, and should make use of a calender, automatic entry of today's date, or drop down boxes for month, date, year.

b) Name two of the Eight Golden Rules that have been violated by this interface. Give a brief explanation of those rules. **(6 points)**

Grading: Each golden rule listed was worth three points. You only needed to describe what the rule was, not how it related to the interface.

1. *Strive for consistency.* This rule is the most frequently violated one, but following it can be tricky because there are many forms of consistency. Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent color, layout, capitalization, fonts, and so on should be employed throughout. Exceptions, such as required confirmation of the delete command or no echoing of passwords, should be comprehensible and limited in number.

2. *Cater to universal usability.* Recognize the needs of diverse users and design for *plasticity*, facilitating transformation of content. Novice-expert differences, age ranges, disabilities, and technology diversity each enrich the spectrum of requirements that guides design. Adding features for novices, such as explanations, and features for experts, such as short cuts and faster pacing, can enrich the interface design and improve perceived system quality.
3. *Offer informative feedback.* For every user action, there should be system feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial. Visual presentation of the objects of interest provides a convenient environment for showing changes explicitly (see the discussion of direct manipulation in Chapter 6).
4. *Design dialogs to yield closure.* Sequences of actions should be organized into groups with a beginning, middle, and end. Informative feedback at the completion of a group of actions gives operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans from their minds, and a signal to prepare for the next group of actions. For example, e-commerce Web sites move users from selecting products to checkout, ending with a clear confirmation page that completes the transaction.
5. *Prevent errors.* As much as possible, design the system such that users cannot make a serious error; for example, gray out menu items that are not appropriate and do not allow alphabetic characters in numeric entry fields (see Section 2.4.5). If users make an error, the system should detect the error and offer simple, constructive, and specific instructions for recovery. For example, users should not have to retype an entire name-address form if they enter an invalid zip code, but rather should be guided to repair only the faulty part. Erroneous actions should leave the system state unchanged, or the system should give instructions about restoring the state.
6. *Permit easy reversal of actions.* As much as possible, actions should be reversible. This feature relieves anxiety, since the user knows that errors can be undone, thus encouraging exploration of unfamiliar options. The units of reversibility may be a single action, a data-entry task, or a complete group of actions, such as entry of a name and address block.
7. *Support internal locus of control.* Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Surprising system actions, tedious sequences of data entries, inability to obtain or difficulty in obtaining necessary information, and inability to produce the action desired all build anxiety and dissatisfaction. Gaines (1981) captured part of this principle with his rule *avoid acausality* and his encouragement to make users the *initiators* of actions rather than the *responders* to actions.
8. *Reduce short-term memory load.* The limitation of human information processing in short-term memory (the rule of thumb is that humans can remember "seven plus or minus two chunks" of information) requires that displays be kept simple, multiple-page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions. Where appropriate, online access to command-syntax forms, abbreviations, codes, and other information should be provided.

3. a) Describe the four principles of direct manipulation. **(4 points)**

Grading: 1 point for each principle.

1. Continuous representation of the objects and actions of interest with meaningful visual metaphors
2. Physical actions or presses of labeled buttons, instead of complex syntax
3. Rapid incremental reversible actions
4. Effects on objects of interest is visible immediately

b) Name two ways you could update the interface in Question 2 to support these principles. Draw a sketch of your redesign in your answer book. **(10 points)**

Grading: If you provide an example that met any of these principles, you would receive credit. Due to the fact that the greatest weaknesses in the interface were the date input and selection of medicine, I expected you addressed these issues in your design.

4. A new start-up, DTUI Inc., is designing a system for hospitals that will allow receptionists to check in people at the hospital faster. They wish to develop an interface that meets both the receptionist and patient's needs. Describe in detail a design methodology of 4-8 stages to facilitate proper design of such a system. Be sure to include early stage data gathering, design processes, user testing, and final stage issues.

Write your answer in the form of a management plan for this project. For each stage, indicate the number of weeks that should be allocated. **(30 points)**

If you provided a plan that contained each of the steps presented in the LUCID model, you would receive credit. As long as the number of wee

Envision: Align the agendas of all stakeholders with organizational strategy and the need for "extreme usability," and develop a clear, shared product vision, embodied in a concept sketch.

Discovery: Study users to determine high-level user requirements, terminology, and mental models.

Design Foundation: Develop a conceptual design and create a key screen prototype to convey the visual style. Usability test the design, revise, repeat.

Design Detail: Flesh out the high-level design into complete specifications.

Build: Support the production process through review and late-stage change management.

Release: Develop a roll-out plan to support for users transition to the new product; conduct final usability test, document lessons learned.

5. Name three expert review methods. For each of them, offer a brief description and how it could be applied to the interface described in Question 4. **(12 points)**

Grading: Here is a list of the expert review methods listed in Chapter 4. By listing three methods, offering a brief description of the method, and offering an idea of how it could be applied to the DTUI interface, you would receive full credit.

Heuristic evaluation The expert reviewers critique an interface to determine conformance with a short list of design heuristics such as the eight golden rules (Chapter 2). It makes an enormous difference if the experts are familiar with the rules and are able to interpret and apply them.

- *Guidelines review* The interface is checked for conformance with the organizational or other guidelines document. Because guidelines documents may contain a thousand items, it may take the expert reviewers some time to master the guidelines, and days or weeks to review a large interface.
- *Consistency inspection* The experts verify consistency across a family of interfaces, checking for consistency of terminology, fonts, color schemes, layout, input and output formats, and so on within the interface as well as in the training materials and online help. Software tools can help automate the process, as well as produce concordances of words and abbreviations.
- *Cognitive walkthrough* The experts simulate users walking through the interface to carry out typical tasks. High-frequency tasks are a starting point, but rare critical tasks, such as error recovery, also should be walked through. Some form of simulating the day in the life of the user should be part of expert-review process. Cognitive walkthroughs were developed for interfaces that can be learned by exploratory browsing (Wharton et al., 1994), but they are useful even for interfaces that require substantial training. An expert might try the walkthrough privately and explore, but then there also would be a group meeting with designers, users, or managers to conduct the walkthrough and to provoke a discussion. Extensions to cover web site navigation incorporate richer descriptions of users and their goals plus linguistic analysis programs to estimate the similarity of link labels and destinations (Blackmon et al., 2002).
- *Formal usability inspection* The experts hold courtroom-style meeting, with a moderator or judge, to present the interface and to discuss its merits and weaknesses. Design-team members may rebut the evidence about problems in an adversarial format. Formal usability inspections can be educational experiences for novice designers and managers, but they may take longer to prepare and more personnel to carry out than do other types of review.

6. a) Describe the notion of *user-interface independence* in two sentences or less. **(5 points)**

user-interface independence: the decoupling of the user-interface design from the complexities of programming.

b) Briefly describe how the designers, user manual writers, and software engineers have benefits that stem from this notion. Be sure you have an explicit description for each of these three populations. **(5 points)**

This decoupling allows the designers to lay out sequences of displays in just a few hours, to make revisions in minutes, and to support the expert-review and usability-testing processes. The programming needed to complete the underlying system can be applied once the user-interface design has been stabilized. The user-interface prototypes can serve as specifications from which writers create user manuals, and from which software engineers build the system using other tools. The latter are required to produce a system that works just like the prototype.