Homework 4 (HW4) -- Part I of Project  
CSC540, UNCG, Fall 2003

Due date and late policies: see course web page.

Purpose of assignment:
1. Gain experience in user-centered design methods, by doing a hierarchical task analysis.
2. Gain experience using a useful design prototyping tool, storyboards.
3. Design and begin implementing (about 50%) of the final program (to be completed for HW6).
4. Get feedback on how to improve the external (user-visible) design of your proposed final program.

What to do to get credit for this assignment:
1. Sign up for a user interface project from the list provided by the instructor. (Only one person can work on each project.)
2. Checklist of what to turn in (paper deliverables should be fastened together using a binder, giant clip, or envelope). Each item is described in more detail in 3-5.
   - cover sheet (CSC540 HW4, your name, and signed Academic Integrity statement)
   - hierarchical task analysis (HTA)
   - storyboard diagrams and written explanation of external design
   - evolutionary prototype. Submit the Java applet's files (.html, .java, .class, .readme files) both on paper and in the course folder so that it can be run by the instructor as you did for HW1.
3. Hierarchical task analysis (3 points): Do an HTA diagram and two plans for a manual system that could be used for a similar purpose to your proposed system. Use the same format as in HW3.
4. Storyboard diagrams and written explanation (6 points): This is a written and illustrated description of the complete, proposed, external (user-visible) design of your final program that you will implement for HW6. You should organize the illustrations and give a written description of the flow from one screen to another that makes it easy for the client to visualize interaction with the final program. (Suggestion: number each illustration for reference in your written description.) Draw illustrations showing everything that your finished program in HW6 will do and what all of its screens will look like. Illustrations should be accompanied by written descriptions explaining features that are shown. To create illustrations, you can use screen shots for parts of the program that you have already implemented for HW4 (use the Prnt Scrn key on a PC to copy the screen and paste it into a document). For unimplemented parts, you can draw the illustrations neatly by hand or use a software drawing tool (such as in Word).
5. Evolutionary prototype (6 points): Implement a Java applet that demonstrates all of the main screen designs of the proposed final program (HW6) and that demonstrates at least that its simple widgets (such as buttons, text entry, menus, choice/radio boxes) are working. (Do not implement any of the design in HTML.) For HW4 it is not necessary to implement the more complex features (for example, slider-controlled data visualization) or much of the application logic (the "backend"); you will add those features for HW6. It is recommended that you plan the internal design (software architecture and classes) of the final program (HW6) now and implement HW4 so that you can add functionality for HW6 to HW4 without having to revise or reorganize much code.
   - Note on code reuse: Do not use code added by a Java development toolkit or from any other source (except the classes that come with Java) such as the Internet, a book, or another person. If you use code that you did not author yourself, it may be considered plagiarism. In doubt about whether you can reuse any code, get permission from the instructor in advance. You may discuss debugging problems with other students or get help with debugging, but you may not use other students' code (and should not let anyone copy your code).
   - Note on "back-end" requirements for the project (both HW4 and HW6): You only need to implement enough of the backend application logic to demonstrate your interface in HW4 and HW6. For example, if a real product could use a back-end database, you could just simulate that for your project with an array of objects; then "queries" to the "back-end database" could be simulated by a simple function that loops through the array and tests the values of attributes of each object. Remember that the purpose of the project is to learn how to design the interface for usability, not to deliver a fully-implemented commercial product!