CSC540 Project Description, Fall 2007

The project involves the design, implementation in Java, and usability testing of a graphical user interface that supports information visualization. This is an individual project worth 40% of the course grade. Each student will be assigned a different application. The applications can be classified as search tools or monitoring tools. An example of a search application for this project is a program to allow a user to explore data about colleges to help her identify colleges to which she would like to apply. Other examples include finding apartments or finding restaurants. An example of a monitoring application is a program that allows a user to monitor data about farm animal diseases so that he can detect possible epidemics, their origin, and geographic areas likely to be affected if it spreads. Other examples include monitoring crime, financial markets, earthquake-volcanic activity, medical data (e.g., in an intensive care unit), industrial processes (e.g., a brewery), weather station, and sales at different stores in a chain. All applications should use static and dynamic information visualization techniques to support the user’s tasks. (This was covered in lecture but not in the textbook.) You are responsible for researching the requirements in more detail, including identifying the target user profiles, inputting realistic data, understanding problems the users would face, identifying tasks and goals, and proposing a design that meets these requirements.

The project is divided into several deliverables as shown in the following schedule. More details about each deliverable are provided below the schedule.

<table>
<thead>
<tr>
<th>Due Date (due at beginning of class)</th>
<th>Deliverables (summarized description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct. 23 (demo may continue on Oct. 25)</td>
<td>Part I: Written Report, First Implementation (all static information presentations), Demo</td>
</tr>
<tr>
<td>Nov. 27 (usability testing may continue Nov. 29)</td>
<td>Part IIa: Final Implementation (Part I with dynamic info viz operations added), Written Usability Test Design</td>
</tr>
<tr>
<td>Dec. 4 (in-class demo may continue Dec. 6)</td>
<td>Part IIb: Final Written Report, Demo</td>
</tr>
</tbody>
</table>

Description of Deliverables (with percent of course grade):
- Part I written report (10%): (See below for general instructions for all written deliverables.) Section 1: describe the intended users, the problem(s) that this product solves for them, their tasks/goals, and the data. (Include a table showing all data values that are hard-coded into the program.) Section 2: give a persona and scenario, illustrated with actual screen shots of the implemented program, to show what the program does. This should illustrate every type of static data presentation but do not show every minor change to every screen. Section 3: Describe the dynamic information visualization operations to be added to the implementation in Part II. Include images created using PowerPoint (as a low-fidelity prototyping tool) to illustrate the dynamic features in use.
- Part I implementation and demo (10%): (See below for details on how you should implement the program and how to submit it for grading!!) This version of the program should provide all the static presentations of data to go in the final program and will be graded on its usability. Your written report and the demo will be the main source of
information for grading it, although the instructor should be able to run the program again later in case questions arise.

- Part IIa written usability test design (5%): You will need to submit a written description of the usability test design and install your Part IIa final implementation in the Bryan 330 lab by the beginning of class on Nov. 27 so that you can test your program’s usability in class (Nov. 27 and 29); everyone must play the role of a user in your classmates’ usability tests. You will be given instructions on usability test design after we cover that material in class. (The usability test results will go in the Part IIb written report.)

- Part IIa final implementation (10%): This version will be graded on the usability of the final product, which should include at least one of these dynamic information visualization operations described in the information visualization lectures: filter, relate (brush), or extract (copy/drag).

- Part IIb written report (5%): This includes the usability test results and an updated version of Part I Section 3 of the written report illustrating the dynamic features in the actual final version of the program; replace the PowerPoint illustrations with real screen shots. As in Part I this report and the final in-class demo will be the main source of information used to grade the Part IIa final implementation, although the instructor should be able to run the program again later.

Implementation Details: The program (Part I and Part II) must be implemented as a Java Applet, and it must run on the PCs in the 330 Bryan Lab. (All demos, usability testing, and grading will be done there.) Therefore, it is recommended that you do most of your development in that lab to avoid last-minute problems due to differences in environments!! Your programs should use the standard Java class framework (SWING, etc.); the only other code that you may use that you did not write yourself is an open-source package called JFreeChart. (We will go over in class some example Java programs that use SWING and JFreeChart.) To keep your program independent of the file system, your program should not do any file input/output; this implies that the data must be hard coded in your program. Although your program uses hard-coded knowledge about the data, its information visualization should work in principle on any data that fits the data definition. Each application’s hard-coded dataset should have at least 50 records and at least 5 attributes; the attributes should include at least one nominal, one ordinal, and one quantitative attribute, and a spatial and/or temporal attribute. All the search applications should have a spatial attribute; all the monitoring applications should have time-series data; some of the monitoring applications such as animal disease, crime, earthquake, and weather, also require spatial data. (Because of copyright restrictions, spatial data does not need to be presented on a high-fidelity map; your program may use a lower-fidelity map created by the program.) You will be given detailed instructions later on how to submit the Java source and executable files for Part I and Part II.

Written deliverables: Each written deliverable (including diagrams) should be computer-edited and printed with pages stapled together. Written deliverables should meet professional standards in terms of readability, grammar, spelling, format, etc. Each written deliverable should include a title page giving the name of the student to receive credit, the date, and the name of the deliverable.