Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

CS 420/CS 620 Human-Computer Interaction

Project Part 2: Requirements and Design

October 30, 2012

Due: Thursday November 14, 8:00 pm, by email, single PDF file named "P2_TX" where X is your team number (1, 2 ... 7)

Points: 100

Weight: 12% in CS 420 and 10% in CS 620

Write a document that covers the functionality and the design of your software. Follow the structure below. Remember that your project should focus on interaction and interface details rather than on algorithmic aspects. In the following, **<X/Y>** indicates a specific requirement as follows: **X** for a 3-student team, and **Y** for a 4-student team.

0 **Cover page**: department, university, project title, author(s), instructor, date

1 **Abstract** (between 100 and 150 words) – revised version of your project's abstract written for P-I: Concept

2 **Requirements elicitation** – use *interviews* and/or *questionnaires* to gather requirements from your target users. You can either involve actual users or play the role(s) of these users. Prepare a list with at least **<10/12>** questions and summarize the answers received on about **<2/3>** pages, single spaced, A-4, Times 12, 1-inch all margins (or equivalent).

3 **Use cases and HCI scenarios**

- Create at least *a persona* for your project and write at least *an HCI scenario* for this persona (or personas). The HCI scenario should cover a use case or a combination of use cases (see below).
- Provide a *use case diagram* for your software with at least **<8/10>** use cases. Briefly describe each use case (2-3 lines each).

4 **Functional requirements**

Provide a list of *functional requirements*, organized on three levels:

- Level 1: Functions and features that will be covered in the prototype's interface due in December 2012 and will be fully implemented (from an execution/algorithm point of view).
- Level 2: Functions and features that will be covered in the prototype's interface due in December 2012 but will not be fully implemented.

- Level 3: Functions and features that will not be covered in the above prototype, but would be useful in a possible continuation of the project beyond the time frame of this course.

To describe your system's functionality and features you can use any technique or combination of techniques, including but not limited to: user requirements, system requirements, form-based specifications, or structured English. The idea is to provide a clear and detailed description of what your system is supposed to do, and under what circumstances or constraints.

5 Design

- *High level design*: include *a system-level structural diagram* (e.g., a system context diagram, site map, or an architecture) and a *system-level behavioral diagram* (an activity chart or state-chart) that describe your overall software.

- *Static interface design*: provide at least **<6/8>** *snapshots* of your system's interface, with brief accompanying explanations and descriptions. Focus on the more important aspects of the interface, avoid duplications, and relate the snapshots to the functionality and features presented in Section 3 of this document.
-
- *Alternative designs*: at least **<1/2>** alternative design solutions (snapshots) considered but not used in the project (e.g., different arrangements of the main window(s), different color schemes, different help structures, etc.). Briefly describe the alternatives, indicate what solutions were chosen (you should make references to the previous snapshots or place the alternatives side by side), and briefly explain why the alternatives were not chosen. Consider larger scope, more significant design decisions rather than small design details.

6 Annotated resources/references

- At least **<4/6>** annotated references (between 50 and 80 words each): journal articles, conference papers, and/or books.

7 Contributions of team members

- Be specific about what each of the team members contributed to this document.