

Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

CS 709B Advanced Software Project Management and Development

March 31, 2009

Project Part III: Design

**Due:** Friday, April 17 @ 5:00 pm, email submission (single PDF file attachment)  
**Points:** 50  
**Weight:** 11% of the course grade

For this part of the project you should provide a *Design Document (DD)* with the following structure (note that **<X/Y>** indicates the number **X** of items required for “teams of one” and the number **Y** of items required for “teams of two”):

**0 Table of contents**

- 1 Abstract:** 100-200 words (you can re-use the abstract you wrote previously – but make changes if updates are needed or if you simply want to improve it).
- 2 Introduction:** a general description (between 400 to 500 words) that re-states the goals of your project and gives a concise account of progress made since the previous report (specification). Indicate changes in the project (if any), refinements, and current status.
- 3 High-level and medium-level design:** present the project in terms of high-level architecture, subsystems, and program units. Given the diversity of projects, there is significant flexibility here. In any case, you should include, with accompanying textual descriptions, the following:
  - At least one *system-level diagram*, e.g., a layered architecture of the system similar, to the one in Fig 17.10 of the book by Arlow and Neustadt (2002), a well explained block diagram, or (for web-based applications only) a site map;
  - The structuring of your software in *program units*. In the case of object-oriented solutions, the classes are examples of such program units, hence a design class diagram with details of attributes, operations, relationships, and multiplicity constraints should be provided (at least **<7/10>** classes are expected). Briefly describe the role of each class as well as the methods included in the classes. In non-object oriented solutions, program units can be modules, functions, procedures, subroutines, etc. Show the organization (hierarchical or not) of these units (at least **<7/12>** units are expected) and provide for each of them: name, description, the higher level unit (e.g., subsystem) to which the program unit belongs, its input, its output, program units called by this unit, its exceptions or interrupts, and any additional comments that could enhance the description of the unit.
  - If database tables are used, for each table indicate its fields (columns) and its primary key(s). For instance, a table containing information on employees may look like the following one (note that the primary key, shown in bold, is SSN):  
| **SSN** | Last Name | First Name | Position | Department | Office | Telephone | Email Address |
- 4 Detailed design:** provide a total **<3/5>** of items of detailed design, as follows:
  - The activity chart of the main “interaction flow” or “work flow” of the system
  - **<2/4>** pseudo-code descriptions, state charts, or activity charts (flow charts)

- 5 **User interface design:** provide at least <7/12> snapshots of the user interface, with accompanying descriptions. In these snapshots, details of the user interface (e.g., panels, toolbars, menus, menu items, buttons, textboxes, etc. for GUI or complete screenshots for text-based interface) should be presented, the format used in reports and statistics should be shown (if applicable), and samples of messages to the user should be provided.
- 6 **References:** provide at least <8/16> references for your project. Very briefly describe how each reference relates to your project (30 -60 words each). You may use references covered in your previous assignments.
- 7 **Glossary updates:** include here at least <3/6> new additions to the project glossary that you wrote for the second part of the project (SRS).
- 8 **Contributions of team members** [<T2> only]