

Department of Computer Science and Engineering

College of Engineering

University of Nevada, Reno

CS 709

Advanced Topics in Computer Science [Software Engineering]

Project Part II: Design

October 31, 2017

Due:	Monday November 13, 11:59 pm
Points:	100
Weight in course grade:	12%

Note: In the following <T1> denotes a single-student project while <T2> denotes a 2-student team project.

For this part of the project you should provide a **Design Document (DD)** with the following structure:

0 Table of contents

- 1 Abstract:** 100-150 words (you can re-use the abstract you wrote previously – but make changes if updates are needed, or simply try to improve it).
- 2 Introduction:** a general description (between 300 to 500 words) that re-states the goals of your project and gives a concise account of progress made since the previous report. 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 block diagram, a context diagram, an architectural pattern (style) [see (Sommerville, 2015)] or a website 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 8 classes are expected for <T1> and at least 14 classes for <T2>). 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, and subroutines. Show the organization (hierarchical or not) of these units (at least 10 units are expected for <T1> and at least 16 units for <T2>) 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.

- 4 **Data design:** provide details of the data structures that will be used in the project. If database tables will be 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 |

- 5 **Detailed design:** provide a total of at least 4 <T1> or 8 <T2> items of detailed design, as follows:
- The activity chart or state chart of the main “interaction flow” or “work flow” of the system
 - At least 3 <T1> or at least 7 <T2> pseudo-code descriptions, state charts, or activity charts/flow charts.
- 6 **User interface design:** provide at least 5 <T1> or at least 8 <T2> 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.
- 7 **References:** provide at least 2 <T1> or at least 4 <T2> new references for your project. For each new reference, provide a brief description of its content (30- 60 words each).
- 8 **Glossary updates:** include at least 4 <T1> or at least 8 <T2> new additions to the project glossary that you created for the deliverable of the project.