

Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

CS 791 Graduate Topics in Computer Science [Software Engineering]

## Project Part 2: Design

October 30, 2019

Due:	Friday November 15, 11:59 pm; by email to the instructor, a pdf file named <a href="#">P3_YourLastName.pdf</a>
Points:	100
Weight in course grade:	10%

**Note:** The notation  $\langle X/Y \rangle$  means  $X$  applies to a team of 1 and  $Y$  applied to a team of 2 students.

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

### 0 Table of contents

- 1 Abstract:** 100-130 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 250 to 400 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 such as in Sommerville (2015), an architectural diagram, or 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  $\langle 8/12 \rangle$  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  $\langle 15/25 \rangle$  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.
- 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 <2/4> items of detailed design, as follows:

- The activity chart or state chart of the main “work flow” of the system
- At least <1/3> other activity charts, state charts, or pseudocode descriptions.

**6 User interface design:** provide at least <6/10> 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.