CS 425 Software Engineering – Project Part 3: Design

Released Nov 4, 2021; due Tuesday Nov 16, 2021 at 11:59 pm; points: 100

Weight 14% of course grade

File naming convention: P3_T**.pdf, where T** is your team number.

Deliverables

Note: In the following **<T4>** denotes a team of four students and **<T5>** denotes a team of five. Also, the notation **<X/Y>** means **X** applies to **<T4>** and **Y** to **<T5>**.

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

0. **Cover Page:** Should contain the following items:
   - Department, university
   - Project title
   - Team number [also, optional, Team name]
   - Team members
   - Instructors
   - External advisor(s)
   - Date
1. **Table of contents**
2. **Abstract:** A revised version of your project's abstract (100 to 125 words).
3. **Introduction**: A concise description (between 250 to 600 words) that re-states the goals of your project and gives a brief account of progress made since the previous report (specification). Also, indicate and briefly explain significant changes and updates made to your project's requirements.
4. **High-level and medium-level design**: Present the project in terms of high-level architecture, subsystems, and program units (modules). 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., your system's context model (see Chapter 5 of the CS 425 textbook) or your system's architectural pattern (see Chapter 6 of the textbook).
   - 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 **<10/12>**classes are expected). Briefly

describe the role of each class and indicate its main attributes and methods (in total, at least **<20/25>** methods should be described).
    - 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 **<16/20>**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 (if any), and any additional comments that could enhance the description of the unit.
- Describe the main *data structures* that will be used in the project. 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 ID# – the employee's ID #) (see below)

Example database table

ID#	Last Name	First Name	Position	Department	Office	Telephone	Email

5. **Detailed design:** Include at least 4 **<T4>** or at least 5 **<T5>** state charts, activity diagrams, and/or sequence diagrams describing in details non-trivial components of your system's behavior (operations/functions). Use at least 2 types of detailed design notations (from the above 3: state charts, activity diagrams, sequence diagrams).
6. **Initial hardware design – only if applicable**
    - Provide a high-level diagram showing the organization of the hardware components of your system (that is, main components and their connections).
    - Include a list of potential components with brief descriptions of their roles. Include several snapshots (photos, figures, or diagrams) of components likely to be used. Indicate the sources used for snapshots/figures.
7. **User interface design:** Provide at least 10 **<T4>** or at least 12 **<T5>** snapshots of the user interface, with accompanying descriptions. In these snapshots, the main user interface components (e.g., panels, toolbars, menus, menu items, buttons, data entry boxes, etc.) should be presented with details, and the format used in outputting results, messages, reports, and/or statistics should be shown.
8. **Version control and software management system**. For your project, create a public repository in GitHub, and include a link to it. Please note that, if necessary, the teaching team will look at the activity in the repository to decide on certain aspects of the grading.
9. **Contributions of team members**. Provide an estimate on how much time each team member worked on this project part (P-3) and indicate on what specific activities. For estimated time use multiples of half hours.

Grading Note that both the technical content and the presentation style (including quality of writing and document formatting) of your design document will be graded.