Department of Computer Science and Engineering
College of Engineering
University of Nevada, Reno

CS 426 - Senior Projects in Computer Science

Project Part 2: Specification & Analysis

February 20, 2016

Due: Friday, March 4, 2016 @ 8:00 pm. Single PDF file named P1_T##.pdf or P1_LastName.pdf, sent by email to the instructor at dascalus@cse.unr.edu

Points: 100

Weight in course grade: **Team option:** 8% **Individual work option:** 11%

Note: The project in this class is usually a team project, but a student has the option to work individually (not in a team) on his or her own software-intensive project. In this case --*individual work option*-- the requirements for the course will consist of project Parts 1, 2, 3, 4, 5, Project Poster, Test, and Class Participation, indicated in the course syllabus presented in class on January 19, 2016.

In the following **<T1>** denotes individual work option (one student), **<T3>** denotes a team of three students, and **<T4>** denotes a team of four students.

A. Team Option

1. Cover page

- Department, University
- Project title
- [Team name,] team members
- Instructor
- External advisor(s) and their affiliation(s)
- Date

2. **Abstract** (100 to 150 words). Concisely and clearly indicate what your project is about, why it is important, and what exactly you intend to design and implement. You may reuse most of your Project concept abstract, but try to do your best to refine and improve it.

3. Requirements

3.1 Requirements workshop summary. Organize a requirements identification workshop consisting of a *requirements brainstorming session*, followed by a *requirements analysis* session (see Section 3.7.4 of the textbook). In the brainstorming session simply record, without discussing, all ideas and requirements-related observations by project members. In the analysis session discuss all the ideas and observations recorded, identify all key requirements, and prioritize them. After the workshop, your team should further revise the requirements and finalize their priority order. For this subsection write *a summary of the brainstorming and analysis session* (200 to 500 words), highlighting the most important findings of this requirement identification workshop. Indicate when and where you met for this workshop, who attended, in what roles, and how long the meeting took.

3.2 **Software requirements**. Specify as completely as possible your project's functional and non-functional requirements. The functional requirements should be prioritized on 3 levels: Level 1 should contain software requirements that your team will implement by the end of April 2016; Level 2 should contain requirements that you will try to implement by then; and Level 3 should consist of requirements that would be good for your project, but will likely not be implemented by the end of the semester.

3.3 **Hardware requirements** [*if applicable*]. Indicate all functional and non-functional requirements for the hardware components of your system, as complete of possible. Prioritization of requirements is optional, but in any case you should indicate all requirements that you plan to implement by end of April 2016.

4 Use Case Modeling

4.1 **Use case diagram.** At least 7 use cases for **<T1>,** 12 for **<T3>**, and 16 for **<T4>** should be included in this diagram. Note also that at least two advanced use case modeling elements should be part of the diagram (actor generalization, use case generalization, inclusion, or extension – see Chapter 5 of the textbook).

4.2 **Detailed use cases**. Provide concise text descriptions (2 to 4 lines each) for all the above use cases, as well as detailed templates for 2 **<T1>**, 4 **<T3>,** or 5 **<T4>** use cases.

4.3 **Requirement traceability matrix**. Include a table with the detailed mappings between use cases and functional requirements.

5 Glossary of terms

Include definitions for at least 10 **<T1>**, 20 **<T3>,** or 26 **<T4>** terms from the problem domain.

6 Analysis

Provide the analysis model of your software application - include and briefly describe *the diagram showing the analysis packages* representing your software. Include at least 1 analysis package for **<T1>**, 3 analysis packages for **<T3>,** and 4 analysis packages for **<T4>.** These packages should depict the high-level structure of your software.

For each package, include *its corresponding analysis class diagram* – or, if (part of) your solution is not object-oriented, its corresponding *hierarchical module diagram.* Then detail each class (or module) in terms of CRC (class/module name, responsibilities, and collaborators) OR names, attributes, and operations.

In total there should be at least 10 analysis classes (or modules) for **<T1>**, 15 for **<T3>** and 20 for **<T4>.** Note that if (part of) your solution is not object-oriented, then (that part of) the analysis model should contain procedural modules rather than classes.

7 References

Provide references for *a problem-domain book* and at least 2 related *scientific articles* for **<T1>,** 4 for **<T3>,** and 6 for **<T4>**. The articles should be relevant *journal papers, conference papers*, and/or online *technical reports*.

8 Contributions of Team Members

Indicate how much time each team member worked on this project part (P2), and on which specific activities.

B. Individual Work Option

You are asked to complete the following items:

1 – Cover Page
2 – Abstract
3.2 – Software requirements
3.3 – Hardware requirements (if applicable)
4.1 – Use case diagram
4.2 – Detailed use cases
5 – Glossary of terms
6 – Analysis
7 – References

as described in Section A above.