Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

CS 425 Software Engineering

Project Part 4:

Implementation, Integration and Testing

December 5, 2013

Due: Demos on Monday December 9, or Tuesday December 10, or Wednesday December 11 Deliverables on Thursday, December 12, 2013 before 5:00 pm (by email).

Points: 100

Weight: 17% of the course grade

A Deliverables of Part 4 of the Project

This final part of the project involves submitting the complete, commented source code of your software (by email, as a zipped file, before 5:00 pm, Thursday December 12) AND demonstrating your project to the instructor on Monday December 09, or Tuesday December 10, or Wednesday December 11 (see course website for details of the schedule). All the members of the team must be present at the demo. Deliverables:

A₁ – *Introduction* & *overview*. The *introduction* document (200 to 400 words) should concisely indicate changes and additions, if any, of your project's specification and design. The *overview file* should provide a <u>summary</u> (1 or 2 pages) of your source code, indicating all program units and modules included in the code. Both *introduction* and *overview* files should be submitted in a zip file that should also contain the entire source code indicated next (A₂).

 A_2 -Source code. The documented source code of your program should be submitted by email in the zip file mentioned above.

A₃ -*Test cases*. Test cases will be illustrated by demonstrating the functionality of your software to the instructor and/or the grader. At the time your *demo* is scheduled you should be prepared with the program "ready to run" and have a sequence of tests prepared. The instructor will try additional uses (tests) of the program and will ask you questions regarding the project.

B Grading of Part 3 of the Project

The final part of the project will be assessed using to the following criteria: completeness (roughly, this means implementation of level 1 requirements), correctness, and robustness of the software; programming style used, including code documentation; quality of the user interface; and quality of the demonstration, including answers to questions.