CSC 415 Spring 2015

Assignment 4 – Open Source Software: Analysis and Design

Due Date: Tuesday March 31, 2015 by 11:59 p.m.
Grade: 50 points as per the rubric.

Objective:

The purpose of this assignment is for students to better understand how to reason about a problem and design an innovative, computational solution.

Requirements:

Project Design

Analyze the requirements for the project that was proposed and approved in Assignment 3, and develop a design for the system that will guide the implementation.

Using correct UML syntax, formalize and document your design using the following diagrams:

- Detailed **design class** diagram to model the **structure** of the system. Be sure to show the relationships between the classes correctly.
- **Statechart** to model the overall **behavior** of the **system**.
- Detailed **statecharts** to model the **behavior** of the key algorithms.

Also provide mock-ups of some key screens of the **user interface** that utilize the guiding principles of the "Eight Golden Rules".

As you design your project, you must give significant consideration to and demonstrate:

- Modularity and encapsulation of data to facilitate information hiding and reuse,
- Efficiency of the algorithms,
- Appropriateness of the data structures used for the problem at hand.

Use LucidChart or a similar UML-aware tool to draw the diagrams. Export the diagrams as pdf documents. No other format will be accepted. If you have not already done so, you can sign up to join my LucidChart account at https://www.lucidchart.com/e/pulimood.tcnj.edu.

Test Case Design

Apply the insights and knowledge you gained about testing methodologies in Assignment 2 and Chapter 22 in the textbook to design a testing approach for your system development.

- Specify the approaches you will use, i.e. the techniques for unit and integration testing.
- Specify whether you will use debugging and testing tools, and if so, which ones. Use the wiki pages developed for Assignment 1 as a guide.
- Design a set of test cases that will effectively demonstrate successful execution of **all** the functionality and error handling features.

A sample format for the test case design is shown below:

Functionality Tested	Inputs	Expected Output	Actual Output

Deliverables:

1. Submit in Canvas as Word (.doc or .docx) or pdf documents:
   - Detailed design class diagram.
   - Statechart that models the overall behavior of the system.
   - Detailed statecharts that model the behavior of the key algorithms.
   - User interface screen mock-ups.
   - Test Case Design document.
   - Full link to private repository on Github.

   Be sure to label each document and diagram with your name and assignment number.

2. Github private repository wiki updated with
   - Detailed design class diagram.
   - Statechart that models the overall behavior of the system.
   - Detailed statecharts that model the behavior of the key algorithms.
   - User interface screen mock-ups.