CSE 120: Software Engineering Course Fall 2022

Software Capstone Project – Lab 04: Implementation Phase Total Points: 200

Purpose of Assignment

The Software Engineering course covers techniques for dealing with the complexity of software systems. We will focus on the technology of software engineering for the individual and small team, rather than business or management issues. This software capstone project aims to provide an opportunity for students to analyze, design, develop, and deploy a software product that is sponsored by an industry partner. By the end of this lab, you will be able to define the main steps of developing a software project by using a code revision and collaboration tools for **implementation phase**.

Direction

This phase of project aims to implement the following:

1- A database/datastore for your project.

2- Code implementation of your software product based on selected IDE and selected platform from *Lab 1*. In order to complete this task, you may use existing components or start from scratch.

a. All team members must use GitHub as a code version tools to track their code modifications, as well as it will be used for collaboration between all members of your group to complete this phase. The code revision tools will also be used for presenting progress reports in the next four weeks. Complete the following tutorial if you do not have experience with GitHub: https://guides.github.com/activities/hello-world/. You may also want to view a video tutorial on GitHub. Here are a couple of good examples:

- https://www.youtube.com/watch?v=SWYqp7iY_Tc
- https://www.youtube.com/watch?v=RGOj5yH7evk

And below is a very good 3-part series on git:

- https://www.youtube.com/watch?v=kmGsHjQ2wsY
- https://www.youtube.com/watch?v=BF2OHMM86Ik
- https://www.youtube.com/watch?v=NXaEImbo-n8

b. Each defined prototype from *Lab 3* must be implemented in this phase. There is still room for innovation and modifications. However, revising all previous phases are required if you have any fundamental change. For instance, if you decide to make substantial modifications on scenarios from *Lab 2*.

c. Object-Oriented Programming (OOP) is not required for implementation phase, but highly recommended. You may find more detail on OOP in the textbook and the recommended textbook. A simple tutorial for C# OOP can be found here and another tutorial for Java OOP can be found here.

Each item you found as a *thing* from *Lab 2* can be considered as an *object*. In order to find the relationship between objects, you may use *state diagram* and *scenarios* from *Lab 2* and the *class diagram* from *Lab 3*.

Lab #04 Assignment Rubric (200 points):

Note: All phases have the standard 1-week grace period past the due date without penalty for finalizing your work if necessary.

Lab04.1. DB Report: Due 14-OCT (40 points):

1- A **summary** of project progress report (6 points) and a URL to your **private** GitHub repository (4 points)

2- The first implementation of database and its key functions (insert, delete, select, etc.). Submit the script of database design (script to create a DB) and implement DB functions. In case your project does not involve implementing a database (from scratch) then follow the instructions below (30 points):

- If your project is provided with the database by the client, then present and describe the schema, with some examples of queries you would use to create the tables and insert the data. Then focus on the queries you will perform against the database.
- If your project does not require a database, but uses data in a different format (text, csv, etc.), provide a description of the data, and focus on the operations you will be performing on the provided data, and any data that your project will generate and maintain.
- If your project does not require a database, then provide a description of the log or input data that will be stored and maintained, and any operations that will be performed against that data.

Lab04.2. Code Report: Due 28-OCT (40 points):

1- A **summary** of project progress report (10 points)

2- **The first code** revision review on GitHub. Submit the URL of your repository and a one to two page that describe the interface of your classes (the main functions of your code). (30 points)

Lab04.3. Code Report: Due 18-NOV (40 points):

1- A **summary** of project progress report (10 points)

2- **The second** code revision review on GitHub, Submit the URL of your repository, a GitHub *compare change* (comparing commits across time) function that describes the difference between previous report and the current report, and a one to two page that describe the interface of your classes (the main functions of your code). (30 points)

Lab04.4. Final Implementation Phase Report: Due 02-DEC (80 points):

1- A **summary** of project progress report for implementation phase (20 points)

2- **Comprehensive** code revision review. Submit a GitHub *compare change* (comparing commits across time) function that describes the difference between the first report, the previous report and the current report (50 points) including the database/datastore and its functionality (10 points)