Introduction to Software Engineering

PROJECT ASSIGNMENT 4 (PA4)

Due Date: **11/12/2024**

Total score: 80 points

This is a project assignment which will be submitted by each team.
You have the following documents to be done and submitted.

a – Revise SAD (15 points)

Your team is asked to revise the Software Architecture Document (SAD) that you submitted in PA3. Your team needs to update Sections 1-4 and fill the content for Sections 5 and 6, based on the feedback for PA3 from the TA's and more information you have about your project now.

For Section 5, draw a deployment diagram(s) using UML and describe briefly each node. Note that the deployment diagram in UML consists of nodes that run your system's components and sub-systems and links among them. For example, if your system runs only on Android devices, without servers, then the diagram has only one node. If your system is a Web-based application, it typically has to run on one or more Web servers, desktop-based Web clients, and mobile Web clients, so the deployment diagram should have three nodes connecting each other via Internet (either HTTP or HTTPS protocols).

For Section 6, provide structures for folders that store your code and supporting files. For example, you have a Web server, provide a folder structure for source code and supporting files on your Web server. Let's just focus on the main folders only. The folder structure can look like a folder structure on Windows.

b – UI prototype (15 points)

Your team needs to sketch the user interface for the system your team is designing. You can use tools such as https://moqups.com, https://www.figma.com, and https://www.visily.ai to design the UI. You need to sketch the UI for the main screen and at least 3 screens for the key scenarios. The key scenarios must follow what you describe in the use-case specifications.

For each screen, you need to describe its purpose, what it shows and how users can use it. To do so, you can copy all screens to a document and provide the description below each screen.

c – Working software (10 points)

As requested in your project planning, each team needs to produce a working software version at the end of each iteration or sprint. In this PA, the TAs will grade the working version of your software. The main flow of at least one main use-case is implemented and working.

d – Test plan and test cases (35 points)

Software testing is an indispensable part of software development. In this step, you are asked to prepare a test plan, design test cases, execute test cases, summarize and report test results. All templates of these documents are provided on Moodle.

In the test plan, you should have several sections as mentioned in the Software Testing lecture. Fill all of the sections specified in the test plan template. In the test plan, you need to list the features needed for testing.

Test cases: You are required to perform functional testing, which is one of the techniques used to test your application.

Select three use-cases from the application to write test cases for testing. You have to write **at least 5 test cases** for each use-case. It means that at least 15 test cases in total should be created.

Test report. Execute the test cases designed at least once. Generate a test report describing what use-cases are tested, on what environments (browsers, devices, etc.) are tested, and all defects found (if any).

e – Weekly report (5 points)

Prepare weekly reports for the weeks starting from the last reported week until the time of submission. Fill out the template with all information required.

Submission Guidelines

You must copy all documents to the directory named **PA4-Group[GroupId]** and compress the whole directory to the zip/rar file named PA4-Group[GroupId].zip/.rar.
For example: **PA4-Group01.zip**

Grading Criteria

using the information your team has at this time. Criteria include:

- English writing: easy to understand, few errors.
- Well formatted texts, diagrams, charts.
- Level of completeness of the documents, based on the information available at this time.
- The software architecture must capture key components necessary for the system.
- Class diagrams must include key classes, attributes, operations, and relationships.
- The UI design must match the use-cases modeled in the previous assignments.
- The UI has sufficient elements (labels, controls, etc.) for users to complete their tasks.