BAHRIA UNIVERSITY (KARACHI CAMPUS)

Software Design & Architecture (SEN-221)

SOLUTION-ASSIGNMENT # 3 – Spring 2023

Problem-Based Learning (PBL)

Based on: CLO-4

Class: **BSE-4A/B** Submission Deadline: **10th Jun 23**

Course Instructor: **ENGR. MAJID KALEEM** Max Marks: **08**

INSTRUCTIONS:

You have to complete this assignment in a group; maximum of three (03) students. This assignment is based on the following PBL attributes:

a) The problem should involve **real world scenarios**.

b) The problem should require students to make **reasoned decisions** and to defend them (investigation and critical analysis).

c) If used for a **group activity**, the problem should involve collaboration and group discussions.

- Present your understanding in the class and upload your findings on LMS as an assignment in **.DOCX** format.
- Assignment will be done in a group; however, each member *must* upload it on LMS individually.

Scenario:

Online collaborative applications are software tools or platforms that enable people to work together on shared tasks or projects over the internet, regardless of their physical location. These applications typically provide a range of features and functionalities to support collaboration, such as real-time communication tools, document sharing and editing capabilities, project management tools, and version control systems.

Examples of online collaborative applications include:

1. ***Project management tools*** such as Trello, Asana, and Basecamp that allow teams to manage and track project progress, assign tasks, and collaborate on project-related documents.

2. ***Communication tools*** such as Slack, Microsoft Teams, and Zoom that enable real-time communication and collaboration through text, voice, and video chat.
3. ***Document sharing and editing tools*** such as Google Docs, Dropbox, and OneDrive that allow multiple users to collaborate on documents simultaneously, track changes, and provide feedback.
4. ***Code collaboration tools*** such as GitHub, Bitbucket, and GitLab that enable developers to collaborate on code repositories, manage code versions, and track changes.
5. ***Online collaborative applications*** have become increasingly popular in recent years, as more and more people work remotely or in geographically distributed teams. These applications can help increase productivity, reduce communication barriers, and improve teamwork by providing a centralized platform for collaboration.

Suppose you have designed an online collaborative application. Now want to make sure that the architecture for the application you have selected is correct. For this purpose, you want to apply any of the architecture evaluation techniques.

List down the names and merits & demerits of various *(at least 5)* architecture evaluation techniques and what do you think which architecture evaluation technique is well suited for this type of software application?

SOLUTION:

Answer similar to the following would be acceptable:

There are several architecture evaluation techniques that can be used to assess the correctness of the architecture for an online collaborative application. Some of the most commonly used techniques are:

1. ***Scenario-based evaluation:*** This technique involves creating hypothetical scenarios and testing how well the architecture supports them. The scenarios can be based on user requirements or potential use cases. The main advantage of this technique is that it can help identify potential issues early on in the design process. However, it can be time-consuming to create and evaluate the scenarios, and there is a risk of overlooking important scenarios.

2. ***Formal verification:*** This technique involves using mathematical proofs to verify the correctness of the architecture. This technique is highly reliable and can provide a high

level of confidence in the correctness of the architecture. However, it can be complex and time-consuming to apply, and may require specialized expertise.

3. ***Expert review:*** This technique involves having experts in the relevant areas review the architecture and provide feedback. This technique can be effective in identifying potential issues and providing recommendations for improvement. However, the quality of the review depends on the expertise of the reviewers, and there is a risk of overlooking important issues.

4. ***Prototyping:*** This technique involves building a prototype of the system and testing it with users. This technique can be effective in identifying usability and functionality issues. However, it can be time-consuming and expensive to build a prototype.

5. ***Quality attribute analysis:*** This technique involves evaluating the architecture against specific quality attributes such as performance, security, and scalability. This technique can help identify potential issues related to these attributes and ensure that the architecture meets the desired quality standards. However, it can be challenging to define the appropriate quality attributes and their associated metrics.

In the case of an online collaborative application, a ***scenario-based evaluation*** and quality attribute analysis could be well-suited techniques. Scenario-based evaluation can help assess the effectiveness of the architecture in supporting collaboration among users, while quality attribute analysis can help ensure that the architecture meets the desired performance, security, and scalability requirements. It is also important to consider the expertise and resources available for each technique before making a final decision. Ultimately, a combination of multiple techniques may provide the most comprehensive evaluation of the architecture.

---Good Luck!---