ASSIGNMENT -2   
CWID : 50380788

1.

System Analysis is the process of figuring out exactly "what" the software needs to do. This involves talking to people, gathering user needs, identifying the problems faced, understanding the existing practices, and defining the requirements based on this information.

Whereas, System Design is the process that is done after the system analysis, figuring out "how" to build it. This includes activities such as creating a blueprint of the overall structure, making decisions about the architecture to follow, breaking down the system into smaller components, and designing how each part interacts through interfaces.

2.

The SDLC is basically a well-defined structure that acts as a checklist in the development of software systems in the industry. It ensures that the software is of high quality, meets the requirements, mitigates risks, and includes detailed recorded documentation. Not only is the product important, but the process also accommodates troubleshooting and adherence to standards.

Essentially, it is a framework that addresses every aspect involved in building software both technical and commercial aspect (Time and Money).

There are many approaches or methodologies to the SDLC, such as Waterfall, Agile, and Object-Oriented Analysis.

3.

Six core processes required in every SDLC are:

* Identify the problem or need and obtain approval

First, you need to figure out what problem you're trying to solve or what need you're addressing with your software. Here, we talk to the people who will be using the software to understand their needs. Once we have a clear idea, we get the green light from management in charge to move forward with the project.

* Plan and monitor the project

Second, having a solid plan to guide the project and checking where we are.

Here, we create a timeline, outline resources, and key milestones using project management tools to keep track of progress and make sure everything stays on schedule and within budget.

* Discover and understand the details

Thirdly, we dig deeper into the specifics of each aspect of the software and its functions.

Here, we document everything, including the functionality of what the software must do and non-functional requirements such as performance and security.

* Design the system components that solve the problem

Fourthly, with the requirements in hand, we can start designing the system. By creating models, interfaces, and how different parts of the system will interact. We break the system into smaller chunks and design each one in detail.

* Build, test, and integrate system components

Fifthly, we start building the system. We write the code, test each component individually to make sure it works correctly, then integrate all the components and test the complete system to ensure everything works together as expected.

* Complete system tests and then deploy the solution

Finally, we test and deploy it on many devices, changing environments to check the readiness before delivery or making it public to the users.

We perform a series of various tests to check all the functional and non-functional aspects, and once everything is good to go, we deploy it in the production environment and provide training and support to the users.

*(4th question answer below)*

**4.**

|  |  |  |
| --- | --- | --- |
| Figure | Category | Reason |
| **a. System Vision Document (Figure 1-8)** | Tradeshow System Overall | *As it deals with overall objective* |
| **b. Work Breakdown Structure (Figure 1-9)** | Tradeshow System Overall | *It elaborates entire A-Z process* |
| **c. List of Use Cases (Figure 1-11)** | Tradeshow System Overall | *It clearly meant for both the subsystems (Product and Supplier)* |
| **d. List of Domain Classes (Figure 1-12)** | Tradeshow System Overall | *It clearly meant for both the subsystems (Product and Supplier)* |
| **e. Domain Class Diagram (Figure 1-13)** | Tradeshow System Overall | *It clearly meant for both the subsystems (Product and Supplier)* |
| **f. Use Case Diagram (Figure 1-14)** | Supplier Subsystem | *It deals with only purchasing agent and the manager* |
| **g. Activity Diagram (Figure 1-15)** | Supplier Subsystem | *The supplier info and contact info in the tradeshow belongs to the Supplier Subsystem* |
| **h. Initial Screen Layout (Figure 1-16)** | Tradeshow System Overall | *It addresses both the product and the supplier subsystem* |
| **i. Database Design (Figure 1-17)** | Supplier Subsystem | *Purely the data is related to the supplier is seen in the figure, however in reality it is for overall Tradeshow system.* |
| **j. Component Diagram (Figure 1-18)** | Tradeshow System Overall | *It shows the browser, internet, entire trade show system and the database* |
| **k. Design Class Diagram (Figure 1-19)** | Supplier Subsystem | *Only includes information of the suppliers* |
| **l. Software Architecture (Figure 1-20)** | Supplier Subsystem | *The view and domain belong to the supplier* |
| **m. Code for Supplier View Class (Figure 1-21)** | Supplier Subsystem | *Code addresses the suppliers* |
| **n. Screen Capture (Figure 1-23)** | Tradeshow System Overall | *It clearly needs information of the product and supplier.* |