

BSSE23058

Junaina Abdul Aziz

Section A.

Q1. Component based software engineering allows faster delivery. State whether this statement is true or false. Justify your answer. [3]

Q2. While in the process of creating design of a software system, you have identified a list of requirements that are expected to change in the future releases of the system. What strategy would you follow in order to ensure that it is easier to change the system in the future? [3]



$$\begin{array}{r}
 1+2+5+6+5+5+5 \\
 \textcircled{15} \\
 +6 \\
 \hline
 21 \\
 +5 \\
 \hline
 26
 \end{array}$$

Q4. Propose the most appropriate process model that might be used as basis for the development of following systems. Give reasons for your answer. [3+3+3]

- a. A complex real time system whose requirements can be relatively easily identified and are stable.

- b. A website for a local library. Requirements are vague and are likely to change in the future.

- c. An order processing system with a website for a local business. Requirements are unlikely to change in the near future.

Q5. Identify five functional requirements from the following case study. [5]

A software development team is tasked to develop a web-based system for an automobile showroom in Bahrain. The system needs to be user friendly, and easily accessible. The system shall allow users to search for cars of different brands and customer should be able to book a car for purchase. When a car is booked, customer will make the payment and will provide delivery address and contact information. It has been decided by the development team to back up the sales data in an external on monthly basis.

The system shall have an option for customers to register. The System shall store customer's name, address, and debit/credit card information, so that they do not have to enter this information every time that they use the system. Customer data will be kept secured to comply with the Data Protection Act.

The showroom manager manages the orders. The orders status is updated in the system e.g., Booked, Payment done, or Delivered.

The system should be designed keeping in view the load on the system. The system should be capable to handle 500 users at a time without effecting its performance. The system design should consider the future expansion of the system other branches of the showroom.

*Order*  
*Customer*  
*Pizza Type*  
*topping s*  
*Receptionist*

Q6. A pizza shop offers a variety of pizzas, each with a base price, and a selection of toppings. Customers may choose one or more toppings, with the option to double any topping. The system must generate the final bill for a pizza order by combining the base price with the prices of selected toppings. [14]

Additional Information:

- Each pizza type (e.g., Ranch, Pepperoni, and Hawaiian) has a predetermined base price.
- Toppings, such as cheese, mushrooms, onions, peppers, etc., have individual prices.
- Customers can select multiple toppings to customize their pizza order.
- The system should calculate the total price, considering both the base price of the pizza and the prices of selected toppings.

(a) Design a flexible and scalable solution for managing pizza orders, allowing for easy addition of new toppings and customization of pizza orders. Provide a class diagram illustrating the structure of your solution, showcasing relationships between classes, and implement the following methods:

- `getDescription()`: Returns the name of the pizza and its selected toppings.
- `cost()`: Calculates the total price of the pizza order.

- (b) Develop a use case diagram illustrating the interactions between actors and the pizza ordering system. Identify relevant actors and use cases depicting various functionalities such as placing an order, customizing toppings, and calculating the total bill. Utilize include and extend relationships to illustrate optional and mandatory use cases.

... 12m