

[CLO 1: To manage the design artifacts and patterns]

Case Study: Blood Bank Management System

Blood bank management system is a system that enables individuals who want to donate blood to help needy. The system targets two types of users: the people who want to donate blood (donors) and recipients who need the donated blood. This system provides various functionalities. It facilitates donor management by recording their physical statistics. It also manages the list of registered donors and recipients. Any registered donor can create a new donation by providing their CNIC. The system checks if the donor is registered and eligible, it creates a new donation record and provides the donor with date and time for the donation. Once the donation is made, it is processed by the system by creating a new blood donation and generating a report. The inventory in the blood bank is responsible for the storage and issuance of blood. Whenever a donation is made, the inventory is updated. By using this system, donors can view their donation records, including when they made donations. The system uses First-In- First-Out stock management, where the bloodstock that is checked into the system first will be the first one given to the recipient when requested. The system can also generate reports for blood donation and blood requests.

Question #1.

- (a) Design a complete class diagram of the blood bank management system. (10 marks)
- (b) For following parts, indicate how your above design is covering the objectives of this system according to General Responsibility Assignment Software Principles? (5 marks)
1. Which class is eligible for information expert according to lastDonationDate, physical Statistics, and registration status?
 2. Which pattern assign responsibility between blood donation and donation request?
 3. Which class acts as a controller class?
 4. How can you minimize dependency between Recipient and BloodBankInventory to get blood?
 5. How can you keep the responsibilities of a class focused and manageable?

[CLO 2: To manage the system artifacts and events]

Question #2. Create a System Sequence Diagram (SSD) for the given use case for Issuances of Sports Equipment. (15 marks)

Actor Action	System Response
1. Member arrives at Sports Desk to request sports equipment and presents identification information	
2. Sports Officer specifies the members credentials	

	<p>3. System verifies the membership</p> <p>4. Credentials, and checks any overdue items, payable fine for the member.</p> <p>5. System verifies that the member is not in the black-list</p>
5. Member requests for specific sports equipment	
6. Sports officer enters details of required equipment	
	<p>7. System checks the availability of required equipment and presents a list of available equipment meeting the criteria</p>
8. Member selects an equipment (or an equipment set) from the equipment list	
9. Sports officer provides the item identifier of the selected item	
	<p>10. System creates a new issuance, logs the item information, issue date, time</p> <p>11. System calculates and presents the return time, and other details</p>
12. Sports officer confirms the details	
	<p>13. System presents a receipt, updates the inventory and status of the equipment.</p>
14. Member takes the equipment and leaves the sports center.	

[CLO 3: To generate code from design artifacts]

Question #3. Generate Code for the given Sequence Diagram. (20 marks)

