

# Architecture Pattern of Blood Bank Management System



**Course Title:** Software Development Project

**Course Number :** CSE 3106

**Submitted to:**

Dr. Amit Kumar Mondal

Assistant Professor

Computer Science and Engineering Discipline

Khulna University ,Khulna

**Submitted By:**

Md. Morslain Biswas

ID:2102009

Md Shohan Hossen

ID:210237

## **Project Title : Blood Bank Management System**

### **Layered Architecture :**

- Layered Architecture for a Blood Bank Management System (LIBSYS) involves structuring the system into distinct layers to manage different aspects efficiently. The Presentation Layer handles the user interface, displaying information related to blood donations, inventory, and donor records. User interactions such as adding or updating data are captured here. The Business Logic Layer processes these inputs, applying rules for managing blood donations, inventory levels, and donor eligibility. It ensures the integrity and consistency of the data and orchestrates business workflows. The Data Access Layer interacts with the database or external systems to retrieve and store data related to donors, inventory, and transactions.

#### **1. Presentation Layer:**

- The Presentation Layer is responsible for presenting the user interface to the user and handling user inputs.
- In the case of a Java Swing application, this layer will consist of the GUI components such as frames, panels, buttons, text fields, etc., along with the corresponding event handling logic.
- This layer should focus solely on the presentation logic and should not contain any business logic or data access code.
- It interacts with the Application Layer to request and receive data, but it does not directly interact with the Business Logic or Data Access Layer.

## **2. Application Layer:**

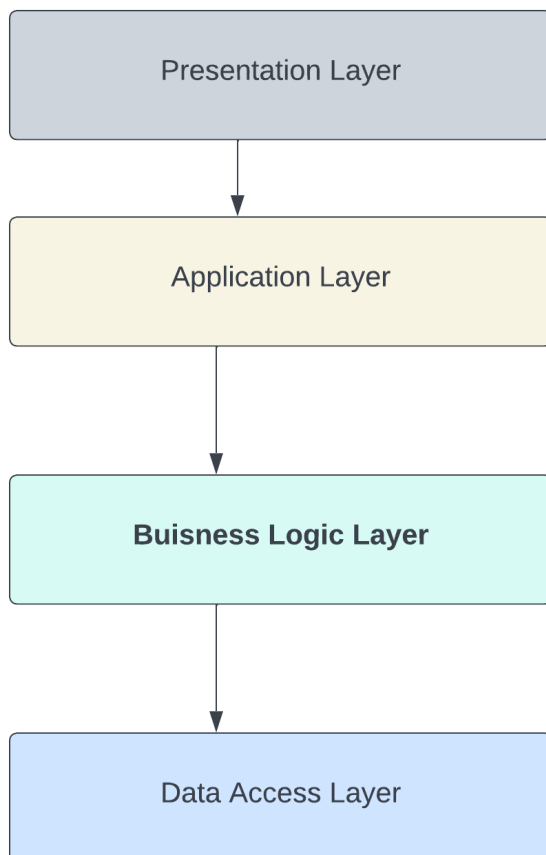
- The Application Layer acts as an intermediary between the Presentation Layer and the Business Logic Layer.
- It receives requests from the Presentation Layer, processes them, and coordinates the flow of data between the Presentation Layer and the Business Logic Layer.
- This layer is responsible for handling user actions, validating inputs, and invoking appropriate methods in the Business Logic Layer to perform business operations.
- It encapsulates the application's use cases and orchestrates the interaction between different components of the system.
- It does not contain any business logic but rather delegates such tasks to the Business Logic Layer.

## **3. Business Logic Layer:**

- The Business Logic Layer contains the core business logic and rules of the application.
- It is responsible for performing operations and computations based on the business requirements.
- This layer encapsulates the business rules and processes, ensuring that they are separated from the presentation and data access concerns.
- It should be independent of any specific user interface or data storage technology, making it reusable and easier to maintain.
- This layer communicates with the Data Access Layer to retrieve or store data but does not contain any direct interaction with the Presentation Layer.

#### 4. Data Access Layer:

- The Data Access Layer is responsible for accessing and manipulating data from external data sources or services.
- In the case of a Blood Management System without a traditional database, this layer might interact with other data sources such as flat files, APIs, or external systems.
- It encapsulates all the data access logic, including querying, updating, and deleting data.
- This layer shields the rest of the application from the complexities of data storage and retrieval, providing a unified interface for accessing data regardless of the underlying storage mechanism.
- It is independent of both the Presentation Layer and the Business Logic Layer, allowing for easier maintenance and scalability.



**Figure : Layered Architecture diagram**

