Computer Science and Engineering Discipline

KHULNA UNIVERSITY , KHULNA

Submitted To

Amit Kumar Mondal

Associate Professor

Computer Science and Engineering Discipline

Khulna University,Khulna

Submitted by

Name:Arnob Chakrobortty

Student ID:210205

Name:Samia Khanom Asa

Student ID:210222

Project Title: Inventory Management System

Decision: Used Layered Architecture for Inventory Management System Development

Reasoning: By organizing the Inventory Management Systems functionalities into these distinct layers we ensure modularity, scalability and maintainability. The presentation layer interacts directly with the user, providing a user-friendly interface for managing inventory. Business logic layer encapsulates the core logic of the application. Separating data access concerns from business logic improves scalability and maintainability. The infrastructure layer abstracts away common concerns that are not specific to the business logic or data access layers.

Layered Architecture Details:

Presentation Layer:

- Responsibility: This layer is responsible for presenting information to the user and gathering user input. It interacts directly with the user interface.
- Functionalities:
    - Displaying user interfaces for various tasks such as adding items, updating quantities, and generating reports.
    - Accepting user input and passing it to the business logic layer for processing.

Business logic Layer:

- Responsibility: This layer contains the core logic and rules of the application. It acts as an intermediary between the presentation layer and the data access layer.
- Functionalities: Validating input data from the presentation layer.
    - Executing business rules and enforcing constraints (e.g., minimum stock levels, pricing rules).

Data Access Layer:

- Responsibility: This layer is responsible for interacting with the data storage mechanism.
- Functionalities: Executing CRUD operations (Create, Read, Update, Delete) on inventory data.

Infrastructure Layer:

- Responsibility: This layer provides supporting infrastructure and services required by the other layers.
- Functionalities:
    - Managing configuration settings.
    - Logging and error handling.
    - Caching data for performance optimization.

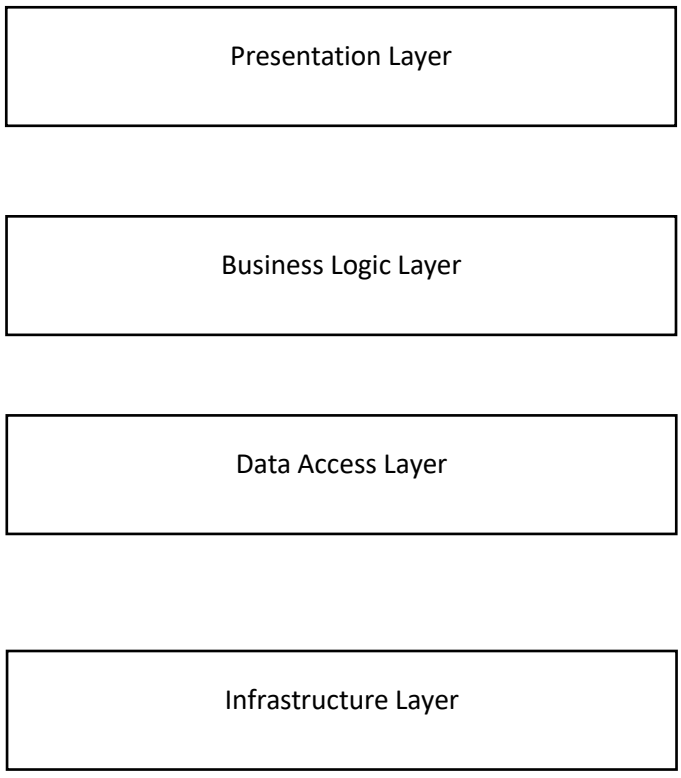

Figure: Diagram of Layered Architecture Pattern.