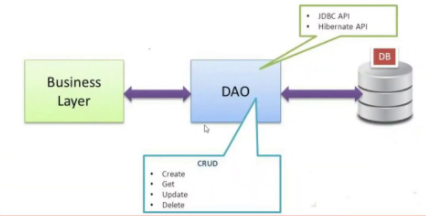
Here we will have to our application in professional manure.

**What does we will have to follow?**

1. Layer Architecture
   1. Presentation Layer
   2. Logic Layer
   3. DAO layer

**Why does we will have to use Layered Architecture?**

It is a object/interface, which is used to access data from database of data storage. WHY WE USE **DAO:**To abstract the retrieval of data from a data resource such as a database. The concept is to "separate a data resource's client interface from its data access mechanism.



DAO stands for Data Access Object. DAO [Design Pattern](https://www.journaldev.com/1827/java-design-patterns-example-tutorial) is used to separate the data persistence logic in a separate layer. This way, the service remains completely in dark about how the low-level operations to access the database is done. This is known as the principle of Separation of Logic.

**DAO Design Pattern**

With DAO design pattern, we have following components on which our design depends:

* The model which is transferred from one layer to the other.
* The [interfaces](https://www.journaldev.com/1601/interface-in-java) which provides a flexible design.
* The interface implementation which is a concrete implementation of the persistence logic.

**Implementing DAO pattern**

With above mentioned components, let’s try to implement the DAO pattern. We will use 3 components here:

Scenario 🡪 We have to develop an application which can maintain Product information.

We will have to save only Product id and Product Name only.

1. The **Product** model which is transferred from one layer to the other.
2. The **ProductDao** interface that provides a flexible design and API to implement.
3. **ProductDaoImpl** concrete class that is an implementation of the **ProductDaoImpl** interface.

s