

We will develop small console base application

## Product-management-system

Database

Login -> table → emailid, password, typeofuser -> column in table

PK

## Java side

In Java for Login table we need to create **JavaBean class** ie **Login**

Class Login

**Emailed**

**Password**

**Typeofuser** are variable with setter and getter method.

Table = JavaBean class

All column = variable names

**DAO : Data Access Object** : inside dao package we need to create more than one class base upon our requirement and in each class we need to write more than one method which contains jdbc logic to insert, delete, update and retrieve.

LoginDao -> signIn and signUp

ProductDao → storeProduct, deleteProduct, updateProduct, retrieveProduct

Now we will create utility package which is responsible to create Menu driven option and take value through keyboard.

`com.pms.utility`

`class MenuOption`

now we will create main class inside below

`com.pms.main`

with name `App`

**resource layer**

this layer or package contains set of classes which provide resource details like database connection.

**Service layer** : this layer contains business logic.

If we write business logic and jdbc ie database login in one class.

If future if we want to do any changes in business logic it will effect to database logic and vice-versa.

**App** -→ **Main** class part of com.pms.main package.

Main method

We call utility methods

-----→

**MenuOption** -→ LoginMenuOption part of com.pms.utility package.

We call service layer methods

-----→

**LoginService** -→ signIn and signUp method part of com.pms.service package

This class contains service logic those logic execute before all dao method or after dao method or no service logic.

**LoginDao** -→ signUp and signIn method with jdbc code interact with database.

-----→

**DbConnection** -→ getConnection() method. this method provide connection details.

In Database create product table

**Product**

pid → PK

pname

price

**bean class** -> generally link table which contains property setter, getter and constructor.

**dao class** -> it contains pure database logic code. Like insert, delete, update and retrieve etc. using JDBC.

**service class** -> it contains pure business logic. Those logic can execute before calling dao method or after dao method or without conditions.

**resource class** -> provide resource details like database connection

**utility class** -> in this class we are written menu driven options.

**Interact with console**

Then we can interact with browser

Then we can interact with angular

**main class** -> entry of the application in core java

Generally dao and service layer not responsible to interact with input as well output device.

In DAO layer, service layer, resource layer don't use System.out.println() to display the message. Because System.out.println always display the message on console not on browser.