```java
package com.pms.main;

import java.util.Iterator;
import java.util.List;
import java.util.Scanner;

import com.pms.bean.Product;
import com.pms.service.OrdersService;
import com.pms.service.ProductService;

public class App {

public static void main(String[] args) {

int choice;
String con="";
Scanner sc = new Scanner(System.in);
int pid;
String pname;
float price;
String result;
ProductService ps = new ProductService();
OrdersService os = new OrdersService();
do {
System.out.println("1:Add, 2 : Delete, 3 : Update 4: Retrieve 5 : Search
Product 6 : Place Order 7 : Order Details");
System.out.println("Plz enter your choice");
choice = sc.nextInt();
switch (choice) {
case 1:     System.out.println("Plz enter the pid");
pid = sc.nextInt();
System.out.println("Plz enter the pname");
pname = sc.next();
System.out.println("Plz enter the price");
price = sc.nextFloat();
Product pp1 = new Product(pid, pname, price);
result = ps.storeProduct(pp1);
System.out.println(result);
break;
case 2:System.out.println("Plz enter the pid");
pid = sc.nextInt();
result = ps.deleteProduct(pid);
System.out.println(result);
break;
case 3:System.out.println("Plz enter the pid");
pid = sc.nextInt();
System.out.println("Plz enter the price");
price = sc.nextFloat();
Product pp2 = new Product();
pp2.setPid(pid);
pp2.setPrice(price);
result = ps.updateProduct(pp2);
System.out.println(result);
break;
case 4: List<Product> listOfProduct = ps.findAllProduct();
for(Product p : listOfProduct) {
```

```
System.out.println(p); // it will call toString method
}
break;
case 5:System.out.println("Plz enter product id");
 pid = sc.nextInt();
 result = ps.findProduct(pid);
 System.out.println(result);
 break;
case 6:     List<Product> listOfProduct1 = ps.findAllProduct();
for(Product p : listOfProduct1) {
System.out.println(p); // it will call toString method
}
System.out.println("Please enter product id which you want to place the
order");
pid = sc.nextInt();
result = os.placeOrder(pid);
System.out.println(result);
break;
case 7 :System.out.println("All Order details are ");
List<Object[]> ll = os.getAllOderDetails();
Iterator<Object[]> li = ll.iterator();
while(li.hasNext()) {
Object obj[] = (Object[])li.next();
System.out.println("Product name "+obj[0]+" Order Date "+obj[1]);
}
break;
default:System.out.println("Wrong choice");
break;
}
System.out.println("Do you want to continue?(y/n)");
con = sc.next();

} while (con.equalsIgnoreCase("y"));

System.out.println("Thank you!");

}
package com.pms.bean;

public class Product {

private int pid;
private String pname;
private float price;

public Product() {
super();
// TODO Auto-generated constructor stub
}

public Product(int pid, String pname, float price) {
super();
this.pid = pid;
this.pname = pname;
this.price = price;
```

```
}

@Override
public String toString() {
return "Product [pid=" + pid + ", pname=" + pname + ", price=" + price +
"]";
}

public int getPid() {
return pid;
}

public void setPid(int pid) {
this.pid = pid;
}

public String getPname() {
return pname;
}

public void setPname(String pname) {
this.pname = pname;
}

public float getPrice() {
return price;
}

public void setPrice(float price) {
this.price = price;
}
```

```
package com.pms.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import com.pms.bean.Product;
import com.pms.resource.DbResource;

public class ProductDao {

public int storeProduct(Product product) {
try {
//                      Class.forName("com.mysql.cj.jdbc.Driver");
//                      Connection con  =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb_phase2", "root",
"Un$ammu#2982");
```

```
Connection con = DbResource.getDbConnection();
PreparedStatement pstmt = con.prepareStatement("insert into product values(?,?,?)");
pstmt.setInt(1, product.getPid());
pstmt.setString(2, product.getPname());
pstmt.setFloat(3, product.getPrice());
return pstmt.executeUpdate();
} catch (Exception e) {
System.err.println("Product insert exception"+e);
return 0;
}
}

public int deleteProduct(int pid) {
try {
//                      Class.forName("com.mysql.cj.jdbc.Driver");
//                      Connection con  =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb_phase2", "root",
"Un$ammu#2982");
Connection con = DbResource.getDbConnection();
PreparedStatement pstmt = con.prepareStatement("delete from product where pid = ?");
pstmt.setInt(1, pid);
return pstmt.executeUpdate();
} catch (Exception e) {
System.err.println("Product delete exception"+e);
return 0;
}
}

public int updateProduct(Product product) {
try {
//                      Class.forName("com.mysql.cj.jdbc.Driver");
//                      Connection con  =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb_phase2", "root",
"Un$ammu#2982");
Connection con = DbResource.getDbConnection();
PreparedStatement pstmt = con.prepareStatement("update product set price  =? where pid = ? ");
pstmt.setInt(2, product.getPid());
pstmt.setFloat(1, product.getPrice());
return pstmt.executeUpdate();
} catch (Exception e) {
System.err.println("Product update exception"+e);
return 0;
}
}

public Product findProduct(int pid) {
try {
//                      Class.forName("com.mysql.cj.jdbc.Driver");
```

```
//                          Connection con  =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb_phase2", "root",
"Un$ammu#2982");
Connection con = DbResource.getDbConnection();
PreparedStatement pstmt = con.prepareStatement("select * from product where pid=?");
pstmt.setInt(1, pid);
ResultSet rs = pstmt.executeQuery();
if(rs.next()) {
Product  p = new Product();                         // converting query into product object.
p.setPid(rs.getInt(1));
p.setPname(rs.getString(2));
p.setPrice(rs.getFloat(3));
return p;
}
} catch (Exception e) {
System.err.println("Search product by id"+e);
}
return null;
}

public List<Product> retrieveProduct() {
List<Product> listOfProduct = new ArrayList<>();
try {
//                          Class.forName("com.mysql.cj.jdbc.Driver");
//                          Connection con  =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb_phase2", "root",
"Un$ammu#2982");
Connection con = DbResource.getDbConnection();
PreparedStatement pstmt = con.prepareStatement("select * from product");
ResultSet rs = pstmt.executeQuery();
while(rs.next()) {
Product  p = new Product();                         // converting query into product object.
p.setPid(rs.getInt(1));
p.setPname(rs.getString(2));
p.setPrice(rs.getFloat(3));
listOfProduct.add(p);
}
} catch (Exception e) {
System.err.println("Search product by id"+e);
}
return listOfProduct;
}
}
```

```
package com.pms.service;

import java.util.Iterator;
```

```java
import java.util.List;

import com.pms.bean.Product;
import com.pms.dao.ProductDao;

public class ProductService {

ProductDao pd = new ProductDao();

public String storeProduct Product product) {
if(product.getPrice()<1000) {
return "Price must be > 1000";                    // simple or complex
condition on demand
}else if(pd.storeProduct(product)>0) {
return "Product details stored successfully";
}else {
return "Product didn't store";
}


}


public String deleteProduct(int pid) {

if(pd.deleteProduct(pid)>0) {
return "Product information deleted successfully";
}else {
return "Product not present";
}


}


public String updateProduct(Product product) {

if(pd.updateProduct(product)>0) {
return "Product information updated successfully";
}else {
return "Product not present";
}
}


public void retrieveProduct() {
//condition
pd.retrieveProduct();


}


public String findProduct int id) {
Product p = pd.findProduct(id);
if(p==null) {
return "Product not present";
}else {
return p.toString(); // toString method display product details in string
format.
}
```

```
}

public List<Product> findAllProduct() {
List<Product> listOfProduct = pd.retrieveProduct();
Iterator<Product> li = listOfProduct.iterator();

while(li.hasNext()) {
Product p = li.next();
float discount =0.10f*p.getPrice(); // 6500
p.setPrice(p.getPrice()-discount);
}

return listOfProduct;
}
}
```

```
package com.pms.resource;

import java.sql.Connection;
import java.sql.DriverManager;

public class DbResource {

static Connection con;
// it load only once
static {
System.out.println("This block loaded only once");
try {
Class.forName("com.mysql.cj.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb_phase2",
"root", "Un$ammu#2982");

} catch (Exception e) {
System.err.println("Db Connection error loaded only once "+e);

}
}


public static Connection getDbConnection() {
try {
return con;
} catch (Exception e) {
System.err.println("Db Connection error "+e);
return null;
}
}

public static void closeConnection() {
try {
con.close();
} catch (Exception e) {

}
}
```