
Billing Software Application

Problem Statement: An ABC company requires a billing software for its super market where different types of products are sold to the customers on retail basis. This billing software will help

- a. the cashier to see the product information on entering the product code
- b. calculate the total cost for each product based on the quantity
- c. store the sales details.

Display menu to cashier to perform sales operations.

1. **Make a sale**
2. **Exit**

1. Enter Product Details

Following information should be accepted from the cashier when he/she bills the product chosen by the customer.

Enter the product code: 1001

Enter the quantity: 2

Note:

- i. Quantity should not be lesser than or equal to zero
- ii. Validation should be taken care for all input fields

2. Calculating Total and displaying the bill

When the above details are entered by the cashier, this application should calculate the line total (product_price * quantity), display the following information and save same in Map in Dao class. Product ID can be auto generated with a counter in Dao class.

Product Name:	iPhone
Product Category:	Electronics
Product Description:	Smart Phone
Product Price (Rs):	35000
Quantity:	2
Line Total (Rs):	70000
Proceed with the sale? [Y/N] : Y	
Sales completed and transaction saved	

Classes to be created

com.app.sales.main

```
public class ProductMain {  
    public static void main (  
        // User Interface which will accept the input from user  
        // Create object for service and execute the respective methods}  
    com.app.sales.bean
```

```
public class Product { // to be created with all required properties and methods}
```

```
com.app.sales.service // Service utility classes and interfaces
```

```
interface ProductService {  
    Product getProductDetails (int productCode);  
    boolean insertSalesDetails(Product prod);  
}
```

```
class ProductServiceImpl implements ProductService {  
    Product getProductDetails (int prodCode){ . . . . . }  
    boolean insertSalesDetails(Product prod){ . . . . . }  
}
```

```
com.app.sales.dao // Class to perform Data Accessing logic
```

```
interface ProductDao{  
}
```

Create appropriate user defined exception classes and any other supporting classes required.