```java
package client;

import model.Product;
import repos.ProductRepo;
import service.ProductService;

import java.util.Scanner;

public class Client {
    public static void main(String[] args) {

        int id;

        // object to access methods of ProductRepo
        ProductRepo productRepo = new ProductRepo();
        Scanner sc = new Scanner(System.in);

//        // Purchase product
//        for(Product p: productRepo.getALlProducts()) {
//            System.out.println(p);
//        }
//        System.out.print("\nEnter id of the product you want to buy: ");
//        id = sc.nextInt();
//        System.out.print("\nEnter quantity: ");
//        int quantity = sc.nextInt();
//        double bill_amt = ProductService.getBillAmount(id, quantity);
//        System.out.println("Bill amount: " + bill_amt);
//        System.out.println("Have you paid the bill (yes/no)? :");
//        sc.nextLine();
//        String bill_payment = sc.nextLine();
//        if(bill_payment.equalsIgnoreCase("yes")) {
//            if(ProductService.purchase(id, quantity)) {
//                System.out.println("Purchase Complete.");
//            }
//            else {
//                System.out.println("Purchase could not be processed.");
//            }
//        }

        // SEARCH BY ID

        System.out.println("Enter product id to search: ");
        id = sc.nextInt();

        Product product = productRepo.getProductById(id);

        if(product == null) {
            System.out.println("No such product found.");
        }
        else {
            System.out.println(product);
        }


        // ADD PRODUCT
        product = new Product();
        System.out.println("Enter product id, title, price, quantity to save: ");
        product.setId(sc.nextInt());
        sc.nextLine();
        product.setTitle(sc.nextLine());
        product.setPrice(sc.nextDouble());
        product.setQuantity(sc.nextInt());

        System.out.println(product);
    }
}

```

```java
package model;

public class Product {
    private int id;
    private String title;
    private double price;
    private int quantity;

    @Override
    public String toString() {
        return "Product{" +
                "id=" + id +
                ", title='" + title + '\'' +
                ", price=" + price +
                ", quantity=" + quantity +
                '}';
    }

    public Product() {
    }

    public Product(int id, String title, double price, int quantity) {
        this.id = id;
        this.title = title;
        this.price = price;
        this.quantity = quantity;
    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
}
```

```java
package repos;

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

// Singleton Class
public class DBUtil {

    private static Connection connection = null;
    // Private constructor for singleton class
    private DBUtil() {}

    public static Connection getConnection() throws SQLException {
        if(connection == null) {
//          String username = "<your-username>";
            String username = "<your-username>";
            String password = "<your-password>";
            String url = "jdbc:mysql://localhost:3306/<your-database>";
            connection = DriverManager.getConnection(url, username, password);
        }
        return connection;
    }
}
```

```java
package repos;

import model.Product;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class ProductRepo{
    private Product convertRowToProduct(ResultSet resultSet) throws SQLException {
        Product product = new Product();
        product.setId(resultSet.getInt("id"));
        product.setTitle(resultSet.getString("title"));
        product.setPrice(resultSet.getDouble("price"));
        product.setQuantity(resultSet.getInt("quantity"));
        return product;
    }
    public List<Product> getALlProducts() {
        List<Product> products = new ArrayList<>();
        try(Connection connection = DBUtil.getConnection();
            PreparedStatement pst = connection.prepareStatement(ProductQueries.GET_ALL_PRODUCTS)) {
            ResultSet resultSet = pst.executeQuery();
            while(resultSet.next()) {
                Product product = convertRowToProduct(resultSet);
                products.add(product);
            }
            resultSet.close();
        }
        catch (SQLException ex){
            System.out.println("Problem:" + ex.getMessage());
        }

        return products;
    }
    public Product getProductById(int productId) {
        Product product = null;
        try(Connection connection = DBUtil.getConnection();
            PreparedStatement pst = connection.prepareStatement(ProductQueries.GET_PRODUCT_BY_ID)) {

            System.out.println("flag1");
            // replacing ? in the statement
            pst.setInt(1, productId);

            System.out.println("flag2");
            ResultSet resultSet = pst.executeQuery();

            System.out.println("flag3");
            while(resultSet.next()) {
                product = convertRowToProduct(resultSet);
            }
            resultSet.close();
        } catch (SQLException e) {
            System.out.println("Problem in getting products. " + e.getMessage());
        }
        return product;
    }
    public boolean save(Product product) {
        try(Connection connection = DBUtil.getConnection();
            PreparedStatement pst = connection.prepareStatement(
ProductQueries.ADD_PRODUCT)) {

            pst.setInt(1, product.getId());
            pst.setString(2, product.getTitle());
            pst.setDouble(3, product.getPrice());
            pst.setInt(4, product.getQuantity());

            // executeUpdate() returns number of rows affected.
            return pst.executeUpdate() == 1;

        } catch (SQLException e) {
            throw new RuntimeException("Problem in getting products.");
        }
    }
    public boolean updateProductPrice(int productId, double newPrice) {
        try(Connection connection = DBUtil.getConnection();
            PreparedStatement pst = connection.prepareStatement(
ProductQueries.UPDATE_PRICE)) {
            pst.setDouble(1, newPrice);
            pst.setInt(2, productId);
            // executeUpdate() returns number of rows affected.
            return pst.executeUpdate() == 1;
        } catch (SQLException e) {
            throw new RuntimeException("Problem in getting products.");
        }
    }
    public boolean updateProductQuantity(int productId, int newQuantity) {
        try(Connection connection = DBUtil.getConnection();
            PreparedStatement pst = connection.prepareStatement(
ProductQueries.UPDATE_QUANTITY)) {
            pst.setDouble(1, newQuantity);
            pst.setInt(2, productId);
            // executeUpdate() returns number of rows affected.
            return pst.executeUpdate() == 1;
        } catch (SQLException e) {
            throw new RuntimeException("Problem in getting products.");
        }
    }
    public boolean remove(int productId) {
        try(Connection connection = DBUtil.getConnection();
            PreparedStatement pst = connection.prepareStatement(
ProductQueries.REMOVE_PRODUCT)) {
            pst.setInt(1, productId);
            // executeUpdate() returns number of rows affected.
            return pst.executeUpdate() == 1;
        } catch (SQLException e) {
            throw new RuntimeException("Problem in getting products.");
        }
    }
}

```

```java
package repos;

public class ProductQueries {
   // add new product query
   public static final String ADD_PRODUCT = "insert into products values (?,?,?,?)";
   // read all products
   public static final String GET_ALL_PRODUCTS = "select * from products";
   // read single product
   public static final String GET_PRODUCT_BY_ID = "select * from products where id = ?";
   // update price
   public static final String UPDATE_PRICE = "update products set price = ? where id = ?";
   // update quantity
   public static final String UPDATE_QUANTITY = "update products set quantity = ? where id = ?";
   // remove product
   public static final String REMOVE_PRODUCT = "delete from products where id = ?";
}
```

```java
package service;

import model.Product;
import repos.DBUtil;
import repos.ProductRepo;

public class ProductService {

    private static boolean checkQuantity(int id, int quantity){
        ProductRepo productRepo = new ProductRepo();
        Product product = productRepo.getProductById(id);
        return product.getQuantity() <= quantity;
    }


    public static double getBillAmount(int id, int quantity) {
        // getting the requested product
        ProductRepo productRepo = new ProductRepo();
        Product product = productRepo.getProductById(id);

        // return -1 if quantity in stock is not enough
//        if(!checkQuantity(id, quantity))
//            return -1;

        return product.getPrice() * quantity;
    }

    public static boolean purchase(int id, int quantity) {
        ProductRepo productRepo = new ProductRepo();
        Product product = productRepo.getProductById(id);
//        if(!checkQuantity(id, quantity))
//            return false;

        return productRepo.updateProductQuantity(id, (product.getQuantity() -
quantity));
    }



}
```

```java
package service;

public class CountriesService {
}

```

```
package service;

public class EmployeesService {
}

```