**FIle javaswingvippro**

**Client:**   
package rmi;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

import java.util.List;

import java.util.Scanner;

public class Client extends JFrame {

private InterfacePro productManager;

private Scanner scanner;

public Client(InterfacePro productManager) {

this.productManager = productManager;

initComponents();

scanner = new Scanner(System.in);

}

private void initComponents() {

setTitle("Product Management Client");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 300);

setLocationRelativeTo(null);

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(6, 1));

JLabel titleLabel = new JLabel("MENU");

titleLabel.setHorizontalAlignment(JLabel.CENTER);

panel.add(titleLabel);

JButton actionButton = new JButton("Chọn chức năng");

actionButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

showOptionDialog();

}

});

panel.add(actionButton);

getContentPane().add(panel);

}

private void showOptionDialog() {

String[] options = {"Hiển thị danh sách những em gái múp", "Tìm kiếm sản phẩm theo ID", "Thêm sản phẩm mới",

"Cập nhật thông tin sản phẩm", "Xóa sản phẩm", "Thoát"};

int response = JOptionPane.showOptionDialog(this,

"Chọn chức năng", "MENU", JOptionPane.DEFAULT\_OPTION,

JOptionPane.PLAIN\_MESSAGE, null, options, options[0]);

switch (response) {

case 0:

displayAllProducts();

break;

case 1:

searchProductById();

break;

case 2:

addNewProduct();

break;

case 3:

updateProduct();

break;

case 4:

deleteProduct();

break;

case 5:

System.exit(0);

break;

default:

break;

}

}

private void displayAllProducts() {

try {

List<Product> allProducts = productManager.getAllProducts();

StringBuilder productList = new StringBuilder("Danh sách sản phẩm:\n");

for (Product product : allProducts) {

productList.append(product.toString()).append("\n");

}

JOptionPane.showMessageDialog(this, productList.toString(), "Danh sách sản phẩm", JOptionPane.INFORMATION\_MESSAGE);

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Lỗi khi hiển thị danh sách sản phẩm: " + e.getMessage(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

private void searchProductById() {

try {

int productId = Integer.parseInt(JOptionPane.showInputDialog(this, "Nhập ID sản phẩm cần tìm:", "Tìm kiếm sản phẩm theo ID", JOptionPane.QUESTION\_MESSAGE));

Product product = productManager.getProductById(productId);

if (product != null) {

JOptionPane.showMessageDialog(this, "Sản phẩm có ID " + productId + ":\n" + product.toString(), "Kết quả tìm kiếm", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(this, "Không tìm thấy sản phẩm có ID " + productId + ".", "Kết quả tìm kiếm", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, "Vui lòng nhập số nguyên cho ID sản phẩm.", "Lỗi", JOptionPane.ERROR\_MESSAGE);

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Lỗi khi tìm kiếm sản phẩm: " + e.getMessage(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

private void addNewProduct() {

try {

int id = Integer.parseInt(JOptionPane.showInputDialog(this, "Nhập ID sản phẩm mới:", "Thêm sản phẩm mới", JOptionPane.QUESTION\_MESSAGE));

String name = JOptionPane.showInputDialog(this, "Nhập tên sản phẩm mới:", "Thêm sản phẩm mới", JOptionPane.QUESTION\_MESSAGE);

String price = JOptionPane.showInputDialog(this, "Nhập giá sản phẩm mới:", "Thêm sản phẩm mới", JOptionPane.QUESTION\_MESSAGE);

String quantity = JOptionPane.showInputDialog(this, "Nhập số lượng sản phẩm mới:", "Thêm sản phẩm mới", JOptionPane.QUESTION\_MESSAGE);

String ngaysx = JOptionPane.showInputDialog(this, "Nhập ngày sản xuất mới (yyyy-MM-dd):", "Thêm sản phẩm mới", JOptionPane.QUESTION\_MESSAGE);

String ngayhethan = JOptionPane.showInputDialog(this, "Nhập ngày hết hạn mới (yyyy-MM-dd):", "Thêm sản phẩm mới", JOptionPane.QUESTION\_MESSAGE);

Product newProduct = new Product(id, name, price, quantity, ngaysx, ngayhethan);

productManager.addProduct(newProduct);

JOptionPane.showMessageDialog(this, "Sản phẩm mới đã được thêm vào.", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, "Vui lòng nhập số nguyên cho ID sản phẩm.", "Lỗi", JOptionPane.ERROR\_MESSAGE);

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Lỗi khi thêm sản phẩm mới: " + e.getMessage(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

private void updateProduct() {

try {

int id = Integer.parseInt(JOptionPane.showInputDialog(this, "Nhập ID sản phẩm cần cập nhật:", "Cập nhật thông tin sản phẩm", JOptionPane.QUESTION\_MESSAGE));

Product productToUpdate = productManager.getProductById(id);

if (productToUpdate != null) {

String name = JOptionPane.showInputDialog(this, "Nhập tên mới cho sản phẩm:", "Cập nhật thông tin sản phẩm", JOptionPane.QUESTION\_MESSAGE);

String price = JOptionPane.showInputDialog(this, "Nhập giá mới cho sản phẩm:", "Cập nhật thông tin sản phẩm", JOptionPane.QUESTION\_MESSAGE);

String quantity = JOptionPane.showInputDialog(this, "Nhập số lượng mới cho sản phẩm:", "Cập nhật thông tin sản phẩm", JOptionPane.QUESTION\_MESSAGE);

String ngaysx = JOptionPane.showInputDialog(this, "Nhập ngày sản xuất mới (yyyy-MM-dd):", "Cập nhật thông tin sản phẩm", JOptionPane.QUESTION\_MESSAGE);

String ngayhethan = JOptionPane.showInputDialog(this, "Nhập ngày hết hạn mới (yyyy-MM-dd):", "Cập nhật thông tin sản phẩm", JOptionPane.QUESTION\_MESSAGE);

productToUpdate.setName(name);

productToUpdate.setPrice(price);

productToUpdate.setQuantity(quantity);

productToUpdate.setNgaysx(ngaysx);

productToUpdate.setNgayhethan(ngayhethan);

productManager.updateProduct(productToUpdate);

JOptionPane.showMessageDialog(this, "Thông tin sản phẩm đã được cập nhật.", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(this, "Không tìm thấy sản phẩm có ID " + id + ".", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, "Vui lòng nhập số nguyên cho ID sản phẩm.", "Lỗi", JOptionPane.ERROR\_MESSAGE);

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Lỗi khi cập nhật sản phẩm: " + e.getMessage(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

private void deleteProduct() {

try {

int id = Integer.parseInt(JOptionPane.showInputDialog(this, "Nhập ID sản phẩm cần xóa:", "Xóa sản phẩm", JOptionPane.QUESTION\_MESSAGE));

Product productToDelete = productManager.getProductById(id);

if (productToDelete != null) {

productManager.deleteProduct(id);

JOptionPane.showMessageDialog(this, "Sản phẩm có ID " + id + " đã được xóa.", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(this, "Không tìm thấy sản phẩm có ID " + id + ".", "Thông báo", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, "Vui lòng nhập số nguyên cho ID sản phẩm.", "Lỗi", JOptionPane.ERROR\_MESSAGE);

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Lỗi khi xóa sản phẩm: " + e.getMessage(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

public static void main(String[] args) {

try {

Registry registry = LocateRegistry.getRegistry("localhost", 1099);

InterfacePro productManager = (InterfacePro) registry.lookup("ProductManager");

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new Client(productManager).setVisible(true);

}

});

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "Client exception: " + e.toString(), "Lỗi", JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

}

}

**Implement:**

package rmi;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class Implement extends UnicastRemoteObject implements InterfacePro {

private static final long serialVersionUID = 1L;

private Connection connection;

public Implement() throws RemoteException {

super();

// Khởi tạo kết nối JDBC ở đây

try {

// Thiết lập kết nối JDBC

String url = "jdbc:mysql://127.0.0.1:3306/product\_management";

String username = "root";

String password = "1234";

connection = DriverManager.getConnection(url, username, password);

} catch (SQLException e) {

e.printStackTrace();

}

}

public List<Product> getAllProducts() throws RemoteException {

List<Product> products = new ArrayList<>();

try {

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("SELECT \* FROM product");

while (resultSet.next()) {

Product product = new Product();

product.setId(resultSet.getInt("id"));

product.setName(resultSet.getString("name"));

product.setPrice(resultSet.getString("price"));

product.setQuantity(resultSet.getString("quantity"));

product.setNgaysx(resultSet.getString("ngaysx")); // Thêm ngaysx vào đây

product.setNgayhethan(resultSet.getString("ngayhethan")); // Thêm ngayhethan vào đây

products.add(product);

}

} catch (SQLException e) {

e.printStackTrace();

}

return products;

}

public Product getProductById(int id) throws RemoteException {

Product product = null;

try {

PreparedStatement statement = connection.prepareStatement("SELECT \* FROM product WHERE id = ?");

statement.setInt(1, id);

ResultSet resultSet = statement.executeQuery();

if (resultSet.next()) {

product = new Product();

product.setId(resultSet.getInt("id"));

product.setName(resultSet.getString("name"));

product.setPrice(resultSet.getString("price"));

product.setQuantity(resultSet.getString("quantity"));

product.setNgaysx(resultSet.getString("ngaysx")); // Thêm ngaysx vào đây

product.setNgayhethan(resultSet.getString("ngayhethan")); // Thêm ngayhethan vào đây

}

} catch (SQLException e) {

e.printStackTrace();

}

return product;

}

public void addProduct(Product product) throws RemoteException {

try {

PreparedStatement statement = connection.prepareStatement("INSERT INTO product(id, name, price, quantity, ngaysx, ngayhethan) VALUES (?, ?, ?, ?, ?, ?)");

statement.setInt(1, product.getId());

statement.setString(2, product.getName());

statement.setString(3, product.getPrice());

statement.setString(4, product.getQuantity());

statement.setString(5, product.getNgaysx()); // Thêm ngaysx vào đây

statement.setString(6, product.getNgayhethan()); // Thêm ngayhethan vào đây

statement.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

public void updateProduct(Product product) throws RemoteException {

try {

PreparedStatement statement = connection.prepareStatement("UPDATE product SET name = ?, price = ?, quantity = ?, ngaysx = ?, ngayhethan = ? WHERE id = ?");

statement.setString(1, product.getName());

statement.setString(2, product.getPrice());

statement.setString(3, product.getQuantity());

statement.setString(4, product.getNgaysx()); // Thêm ngaysx vào đây

statement.setString(5, product.getNgayhethan()); // Thêm ngayhethan vào đây

statement.setInt(6, product.getId());

statement.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

public void deleteProduct(int id) throws RemoteException {

try {

PreparedStatement statement = connection.prepareStatement("DELETE FROM product WHERE id = ?");

statement.setInt(1, id);

statement.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

**InterfacePro:**

package rmi;

import java.rmi.Remote;

import java.rmi.RemoteException;

import java.util.List;

public interface InterfacePro extends Remote {

List<Product> getAllProducts() throws RemoteException;

Product getProductById(int id) throws RemoteException;

void addProduct(Product product) throws RemoteException;

void updateProduct(Product product) throws RemoteException;

void deleteProduct(int id) throws RemoteException;

}

**Product:**

package rmi;

import java.io.Serializable;

public class Product implements Serializable {

private static final long serialVersionUID = 1L;

private int id;

private String name;

private String price;

private String quantity;

private String ngaysx;

private String ngayhethan;

public Product() {

// Constructor mặc định

}

public Product(int id, String name, String price, String quantity, String ngaysx, String ngayhethan) {

this.id = id;

this.name = name;

this.price = price;

this.quantity = quantity;

this.ngaysx = ngaysx;

this.ngayhethan = ngayhethan;

}

// Getter và setter cho id

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

// Getter và setter cho name

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

// Getter và setter cho price

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

public String getNgaysx() {

return ngaysx;

}

public void setNgaysx(String ngaysx) {

this.ngaysx = ngaysx;

}

public String getNgayhethan() {

return ngayhethan;

}

public void setNgayhethan(String ngayhethan) {

this.ngayhethan = ngayhethan;

}

// Getter và setter cho quantity

public String getQuantity() {

return quantity;

}

public void setQuantity(String quantity) {

this.quantity = quantity;

}

@Override

public String toString() {

return "Product{" +

"id=" + id +

", name='" + name + '\'' +

", price=" + price +

", quantity=" + quantity +

", ngaysx='" + ngaysx + '\'' +

", ngayhethan='" + ngayhethan + '\'' +

'}';

}

}

**Server:**package rmi;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

public class Server {

public static void main(String[] args) {

try {

Implement productManager = new Implement();

Registry registry = LocateRegistry.createRegistry(1099);

registry.rebind("ProductManager", productManager);

System.out.println("Server started.");

} catch (Exception e) {

System.err.println("Server exception: " + e.toString());

e.printStackTrace();

}

}

}