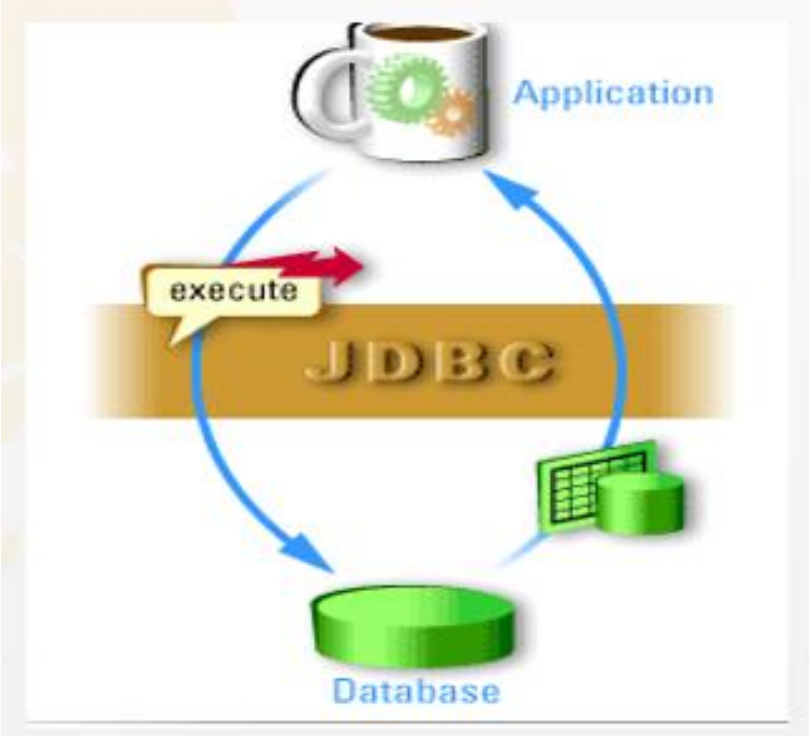
Java EE™

Bài 3
Truy Xuất Dữ Liệu Trong Database

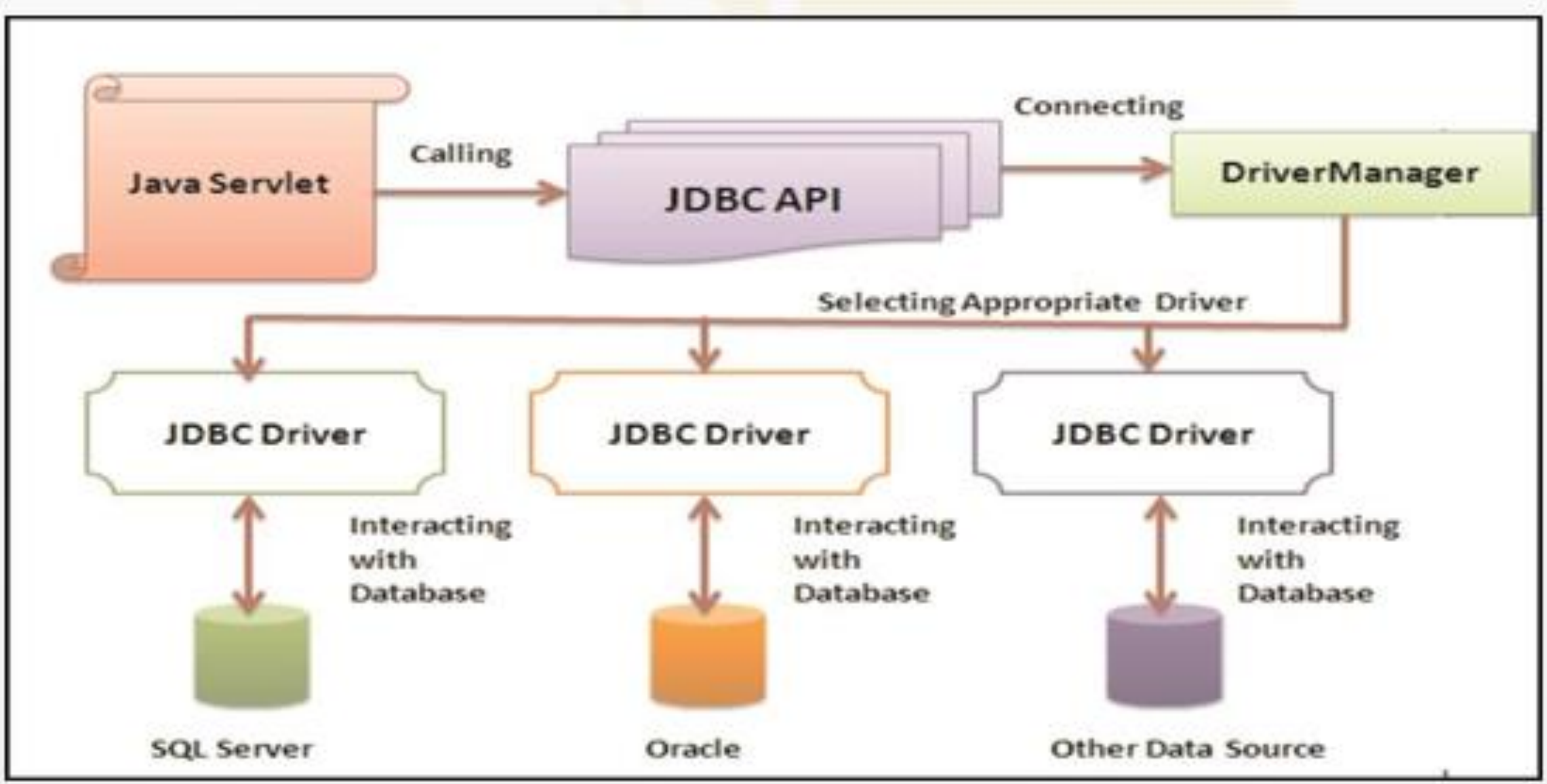
MỤC TIÊU

- Giải thích điều khiển CSDL sử dụng JDBC
- Mô tả JDBC và vai trò của nó
- Giải thích kết nối CSDL sử dụng JDBC
- Hiểu các annotation để tạo các class ánh xạ
- Khai báo và cài đặt các hàm trong DAO để tương tác với CSDL
- Thực hiện được các chức năng CRUD với CSDL

- JDBC cho phép các chương trình Java kết nối với các hệ quản trị CSDL, sử dụng cho các chức năng của ứng dụng.
- Nó là một phần của nền tảng Java SE cho phép ứng dụng web tương tác với một hệ quản trị CSDL.



Kiến trúc JDBC



Các bước tương tác với CSDL bằng JDBC

1. Nạp lớp JDBC driver
2. Thiết lập kết nối với CSDL
3. Tạo và thực thi truy vấn SQL
4. Xử lý kết quả
5. Đóng lại kết nối

Cài đặt hàm để kết nối tới CSDL

- Add gói thư viện sqljdbc4.jar vào project và cài đặt hàm sau:

DBUtility.java

```java
package db;

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

public class DBUtility {
    public static Connection openConnection() {
        Connection con = null;
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            con = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=DB_Product", "sa", "1234$");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return con;
    }

}
```

Khai báo các hàm xử lý trong giao diện DAO

- DAO: Viết tắt của Database Access Object (Đối tượng truy xuất CSDL)

ProductDAO.java

```
1  package dao;
2
3  import java.util.List;
4
5  import entities.Product;
6
7  public interface ProductDAO {
8      public List<Product> getProducts();
9
10     public boolean insertProduct(Product p);
11
12     public boolean updateProduct(Product p);
13
14     public boolean deleteProduct(int proId);
15
16     public Product getProductById(int proId);
17 }
18
```

Cài đặt hàm getProducts() trong lớp thực thi DAO

ProductDAOImpl.java

```
13
14  public class ProductDAOImpl implements ProductDAO{
15      @Override
16      public List<Product> getProducts() {
17          List<Product> list = new ArrayList<Product>();
18
19          Connection con;
20          PreparedStatement pstmt = null;
21          ResultSet rs = null;
22
23          con = DBUtility.openConnection();
24          try {
25              pstmt = con.prepareStatement("select * from Product");
26              rs = pstmt.executeQuery();
27
28              while(rs.next()) {
29                  Product p = new Product();
30                  p.setProId(rs.getInt("ProId"));
31                  p.setProName(rs.getString("ProName"));
32                  p.setProducer(rs.getString("Producer"));
33                  p.setYearMaking(rs.getInt("YearMaking"));
34                  p.setExpireDate(rs.getDate("ExpireDate"));
35                  p.setPrice(rs.getDouble("Price"));
36                  p.setStatus(rs.getBoolean("Status"));
37                  list.add(p);
38              }
39          } catch (SQLException e) {
40              e.printStackTrace();
41          }finally {
42              DBUtility.closeAll(con, pstmt, rs);
43          }
```

Hàm insertProduct() trong lớp thực thi DAO

ProductDAOImpl.java

```
48      @Override
49      public boolean insertProduct(Product p) {
50          boolean bl = false;
51
52          Connection con;
53          PreparedStatement pstmt = null;
54          ResultSet rs = null;
55
56          con = DBUtility.openConnection();
57          try {
58              pstmt = con.prepareStatement("insert into Product values (?,?,?,?,?,?)");
59              pstmt.setString(1, p.getProName());
60              pstmt.setString(2, p.getProducer());
61              pstmt.setInt(3, p.getYearMaking());
62              pstmt.setDate(4, new Date(p.getExpireDate().getTime()));
63              pstmt.setDouble(5, p.getPrice());
64              pstmt.setBoolean(6, p.isStatus());
65
66              int i = pstmt.executeUpdate();
67              if(i>0)
68                  bl = true;
69          } catch (SQLException e) {
70              e.printStackTrace();
71          }finally {
72              DBUtility.closeAll(con, pstmt, rs);
73          }
74
75          return bl;
76      }
```

Hàm updateProduct() trong lớp thực thi DAO

ProductDAOImpl.java

```java
@Override
public boolean updateProduct(Product p) {
    boolean bl = false;

    Connection con;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    con = DBUtility.openConnection();
    try {
        pstmt = con.prepareStatement("update product set ProName=?, Producer=?, YearMaking=?, ExpireDate=?, Price=?, Status=? where ProId=?");
        pstmt.setString(1, p.getProName());
        pstmt.setString(2, p.getProducer());
        pstmt.setInt(3, p.getYearMaking());
        pstmt.setDate(4, new Date(p.getExpireDate().getTime()));
        pstmt.setDouble(5, p.getPrice());
        pstmt.setBoolean(6, p.isStatus());
        pstmt.setInt(7, p.getProId());

        int i = pstmt.executeUpdate();
        if(i>0)
            bl = true;
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        DBUtility.closeAll(con, pstmt, rs);
    }

    return bl;
}
```

Hàm deleteProduct() trong lớp thực thi DAO

ProductDAOImpl.java

```
109
110     @Override
111     public boolean deleteProduct(int proId) {
112         boolean bl = false;
113
114         Connection con;
115         PreparedStatement pstmt = null;
116         ResultSet rs = null;
117
118         con = DBUtility.openConnection();
119         try {
120             pstmt = con.prepareStatement("delete from Product where ProId=?");
121             pstmt.setInt(1, proId);
122
123             int i = pstmt.executeUpdate();
124             if(i>0)
125                 bl = true;
126         } catch (SQLException e) {
127             e.printStackTrace();
128         }finally {
129             DBUtility.closeAll(con, pstmt, rs);
130         }
131
132         return bl;
133     }
```

Hàm deleteProduct() trong lớp thực thi DAO

ProductDAOImpl.java

```
109
110     @Override
111     public boolean deleteProduct(int proId) {
112         boolean bl = false;
113
114         Connection con;
115         PreparedStatement pstmt = null;
116         ResultSet rs = null;
117
118         con = DBUtility.openConnection();
119         try {
120             pstmt = con.prepareStatement("delete from Product where ProId=?");
121             pstmt.setInt(1, proId);
122
123             int i = pstmt.executeUpdate();
124             if(i>0)
125                 bl = true;
126         } catch (SQLException e) {
127             e.printStackTrace();
128         }finally {
129             DBUtility.closeAll(con, pstmt, rs);
130         }
131
132         return bl;
133     }
```

Hàm getProductById() trong lớp thực thi DAO

ProductDAOImpl.java

```
135     @Override
136     public Product getProductById(int proId) {
137         Product p = null;
138
139         Connection con;
140         PreparedStatement pstmt = null;
141         ResultSet rs = null;
142
143         con = DBUtility.openConnection();
144         try {
145             pstmt = con.prepareStatement("select * from Product where ProId=?");
146             pstmt.setInt(1, proId);
147             rs = pstmt.executeQuery();
148
149             if(rs.next()) {
150                 p = new Product();
151                 p.setProId(rs.getInt("ProId"));
152                 p.setProName(rs.getString("ProName"));
153                 p.setProducer(rs.getString("Producer"));
154                 p.setYearMaking(rs.getInt("YearMaking"));
155                 p.setExpireDate(rs.getDate("ExpireDate"));
156                 p.setPrice(rs.getDouble("Price"));
157                 p.setStatus(rs.getBoolean("Status"));
158             }
159         } catch (SQLException e) {
160             e.printStackTrace();
161         }finally {
162             DBUtility.closeAll(con, pstmt, rs);
163         }
164         return p;
165     }
```

Servlet xử lý chức năng load dữ liệu

```
18  @WebServlet("/LoadProducts")
19  public class LoadProducts extends HttpServlet {
20      private static final long serialVersionUID = 1L;
21
22      /**
23       * @see HttpServlet#HttpServlet()
24       */
25      public LoadProducts() {
26          super();
27          // TODO Auto-generated constructor stub
28      }
29      /**
30       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
31       */
32      protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
33          // TODO Auto-generated method stub
34
35          List<Product> list = new ProductDAOImpl().getProducts();
36          request.setAttribute("listP", list);
37
38          request.getRequestDispatcher("listProduct.jsp").forward(request, response);
39      }
40      /**
41       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
42       */
43      protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
44          // TODO Auto-generated method stub
45          doGet(request, response);
46      }
47  }
```

Trang jsp để hiển thị dữ liệu list Products

```
26      <table border="1">
27          <tr>
28              <th>Product Id</th>
29              <th>Product Name</th>
30              <th>Producer</th>
31              <th>Year Making</th>
32              <th>Expire Date</th>
33              <th>Price</th>
34              <th>Status</th>
35              <th>Delete</th>
36              <th>Details</th>
37          </tr>
38
39          <c:forEach items="${listP}" var="p">
40              <tr>
41                  <td>${p.proId}</td>
42                  <td>${p.proName}</td>
43                  <td>${p.producer}</td>
44                  <td>${p.yearMaking}</td>
45                  <td>${p.expireDate}</td>
46                  <td><fmt:formatNumber value="${p.price}"/> </td>
47                  <td>${p.status?"Còn hàng":"Hết hàng"}</td>
48                  <td>
49                      <a href="DeleteProduct?proId=${p.proId}" onclick="return confirm('Sure?')">delete</a>
50                  </td>
51                  <td>
52                      <a href="DetailProduct?proId=${p.proId}">detail</a>
53                  </td>
54              </tr>
55          </c:forEach>
56      </table>
```

Trang jsp insert Product

```
14          <form action="InsertProduct" method="post">
15              <table border="1">
16                  <tr>
17                      <td>Product Name</td>
18                      <td>
19                          <input type="text" name="proName"/>
20                      </td>
21                  </tr>
22                  <tr>
23                      <td>Producer</td>
24                      <td>
25                          <input type="text" name="producer"/>
26                      </td>
27                  </tr>
28                  <tr>
29                      <td>Year making</td>
30                      <td>
31                          <input type="number" name="yearMaking"/>
32                      </td>
33                  </tr>
34                  <tr>
35                      <td>Expire date</td>
36                      <td>
37                          <input type="date" name="expireDate"/>
38                      </td>
39                  </tr>
40                  <tr>
41                      <td>Price</td>
42                      <td>
43                          <input type="number" name="price"/>
44                      </td>
45                  </tr>
```

Servlet xử lý chức năng insert dữ liệu

```
36      protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37          request.setCharacterEncoding("UTF-8");
38
39          String proName = request.getParameter("proName");
40          String producer = request.getParameter("producer");
41          String year = request.getParameter("yearMaking");
42          String date = request.getParameter("expireDate");
43          String price = request.getParameter("price");
44
45          int yearMaking = Integer.parseInt(year);
46          SimpleDateFormat s = new  SimpleDateFormat("yyyy-MM-dd");
47          Date expireDate = null;
48          try {
49              expireDate = s.parse(date);
50          } catch (ParseException e) {
51              e.printStackTrace();
52          }
53          double pr = Double.parseDouble(price);
54
55          Product p = new Product(0, proName, producer, yearMaking, expireDate, pr, true);
56          boolean bl = new ProductDAOImpl().insertProduct(p);
57          if(bl) {
58              request.getRequestDispatcher("index.jsp").forward(request, response);
59          }else {
60              request.setAttribute("err", "Insert failed!");
61              request.getRequestDispatcher("insertProduct.jsp").forward(request, response);
62          }
63      }
```

Servlet xử lý pre update Product

PreUpdate.java

```java
@WebServlet("/PreUpdate")
public class PreUpdate extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public PreUpdate() {
        super();
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        int proId = Integer.parseInt(request.getParameter("proId"));
        Product p = new ProductDAOImpl().getProductById(proId);
        request.setAttribute("p", p);
        request.getRequestDispatcher("updateProduct.jsp").forward(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }

}
```

Trang jsp update Product

```
<form action="UpdateProduct" method="post">
    <table border="1">
        <tr>
            <td>Product Id</td>
            <td>
                <input type="text" name="proId" value="${p.proId}" readonly="true"/>
            </td>
        </tr>
        <tr>
            <td>Product Name</td>
            <td>
                <input type="text" name="proName" value="${p.proName}"/>
            </td>
        </tr>
        <tr>
            <td>Producer</td>
            <td>
                <input type="text" name="producer" value="${p.producer}"/>
            </td>
        </tr>
        <tr>
            <td>Year making</td>
            <td>
                <input type="number" name="yearMaking" value="${p.yearMaking}"/>
            </td>
        </tr>
        <tr>
            <td>Expire date</td>
            <td>
                <input type="date" name="expireDate" value="${p.expireDate}"/>
            </td>
        </tr>
```

Servlet xử lý update Product

```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");

    int proId = Integer.parseInt(request.getParameter("proId"));
    String proName = request.getParameter("proName");
    String producer = request.getParameter("producer");
    String year = request.getParameter("yearMaking");
    String date = request.getParameter("expireDate");
    String price = request.getParameter("price");
    boolean status = Boolean.parseBoolean(request.getParameter("status"));

    int yearMaking = Integer.parseInt(year);
    SimpleDateFormat s = new  SimpleDateFormat("yyyy-MM-dd");
    Date expireDate = null;
    try {
        expireDate = s.parse(date);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    double pr = Double.parseDouble(price);
    Product p = new Product(proId, proName, producer, yearMaking, expireDate, pr, status);
    boolean bl = new ProductDAOImpl().updateProduct(p);
    if(bl) {
        request.setAttribute("p", p);
        request.setAttribute("success", "Update successfully!");
        request.getRequestDispatcher("detailProduct.jsp").forward(request, response);
    }else {
        request.setAttribute("err", "Update failed!");
        request.setAttribute("p", p);
        request.getRequestDispatcher("updateProduct.jsp").forward(request, response);
    }
}
```

Trang jsp hiển thị detail Product

```
20          <h1>DETAIL PRODUCT</h1>
21          <h3 style="color:green">${success}</h3>
22
23          <table border="1">
24              <tr>
25                  <td>Product Id</td>
26                  <td>${p.proId}</td>
27              </tr>
28              <tr>
29                  <td>Product Name</td>
30                  <td>${p.proName}</td>
31              </tr>
32              <tr>
33                  <td>Producer</td>
34                  <td>${p.producer}</td>
35              </tr>
36              <tr>
37                  <td>Year Making</td>
38                  <td>${p.yearMaking}</td>
39              </tr>
40              <tr>
41                  <td>Expire Date</td>
42                  <td><fmt:formatDate value="${p.expireDate}" pattern="dd/MM/yyyy"/> </td>
43              </tr>
44              <tr>
45                  <td>Price</td>
46                  <td><fmt:formatNumber value="${p.price}"></fmt:formatNumber> (VND)</td>
47              </tr>
48              <tr>
49                  <td>Status</td>
50                  <td>${p.status?"In":"Out"}</td>
51              </tr>
```

HỎI ĐÁP

TRẢI NGHIỆM THỰC HÀNH

TRÂN TRỌNG CẢM ƠN!

238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội

0968.27.6996

tuyensinh@bachkhoa-aptech.edu.vn

www.bachkhoa-aptech.edu.vn