infoEmployes.jsp

```jsp
    <h1>Liste Employes</h1><br>
<table border="1">
    <%
        ListEmployes listEmployes =
        (ListEmployes)request.getAttribute("listEmployes");
    %>
    <thead>
    <tr>
        <th>#</th>
        <th>Nom</th>
        <th>Age</th>
        <th>Fonction</th>
        <th>Salaire</th>
        <th>Service</th>
    </tr>
    </thead>
    <tbody>
        <% for (Employe employe : listEmployes.getEmployes()) { %>
            <tr>
                <td><%= employe.getEmpNo() %></td>
                <td><%= employe.getNom() %></td>
                <td><%= employe.getAge() %></td>
                <td><%= employe.getFonction() %></td>
                <td><%= employe.getSalaire() %></td>
                <td><%= employe.getService().getDname() %></td>
            </tr>
        <% } %>
    </tbody>
</table><br> <br>
```

saisieEmploye.jsp

```jsp
<head>
    <meta charset="UTF-8">
    <title>Saisie Employé</title>
</head>
<body>
<form action="TraitementEmploye" method="post">
    <%!
    ServiceDao serviceDao = new ServiceDao();
     List<Service> services =
             serviceDao.getListeServices().getServices();
    %>
    <h1>Ajouter Employe</h1>
    <label for="nom">Nom :</label>
    <input type="text" name="nom" id="nom" required>
    <br><br>
    <label for="age">Age :</label>
    <input type="text" name="age" id="age" required>
    <br><br>
    <label for="fonction">Fonction :</label>
    <input type="text" name="fonction" id="fonction" required>
    <br><br>
    <label for="salaire">Salaire :</label>
    <input type="text" name="salaire" id="salaire" required>
    <br><br>
    <label for="service">Service :</label>
    <select name="service" id="service" required>
       <% for (Service service : services) { %>
            <option value="<%= service.getDeptNo() %>">
            <%= service.getDname() %></option>
        <% } %>
    </select>

    <br><br>
    <input type="submit" value="Ajouter">
</form>  </body>    </html>
```

TraitementEmploye.java

```java
public class TraitementEmploye extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String nom = request.getParameter("nom");
        int age = Integer.parseInt(request.getParameter("age"));
        String fonction = request.getParameter("fonction");
        double salaire = Double.parseDouble(request.getParameter("salaire"));

        Employe employe = new Employe();
        employe.setNom(nom);
        employe.setAge(age);
        employe.setFonction(fonction);
        employe.setSalaire(salaire);

        int deptNo = Integer.parseInt(request.getParameter("service"));

        ServiceDao serviceDao = new ServiceDao();
        Service service = serviceDao.getServicesByID(deptNo);

        employe.setService(service);

        EmployeDao employeDao = new EmployeDao();
        Employe emp = employeDao.enregistrerEmploye(employe);

        ListEmployes listEmployes = employeDao.getListeEmployes();

        request.setAttribute("listEmployes", listEmployes);

        request.getRequestDispatcher("infoEmployes.jsp").forward(request, response);
    }
}
```

ServiceDao.java

```java
package ma.dao;

import ma.model.Service;

public class ServiceDao implements IServiceDao {

    private static final String SELECT_SERVICE_BY_NAME
    = "SELECT * FROM dept WHERE DNAME = ?";

    public Service getService(String name) {
        Service service = null;

        try (Connection connection = DataSource.getConnection();
          PreparedStatement preparedStatement
          = connection.prepareStatement(SELECT_SERVICE_BY_NAME)) {

            preparedStatement.setString(1, name);
            try (ResultSet resultSet = preparedStatement.executeQuery()) {
                if (resultSet.next()) {
                    service = new Service();
                    service.setDeptNo(resultSet.getInt("DEPTNO"));
                    service.setDname(resultSet.getString("DNAME"));
                    service.setLoc(resultSet.getString("LOC"));
                }
            }

        } catch (SQLException e) {
```

ServiceDao.java

```java
    private static final String SELECT_SERVICE_BY_ID =
            "SELECT * FROM dept WHERE DEPTNO = ?";
    @Override
    public Service getServicesByID(int id) {
        Service service = null;

        try (Connection connection = DataSource.getConnection();
             PreparedStatement preparedStatement =
                     connection.prepareStatement(SELECT_SERVICE_BY_ID)) {

            preparedStatement.setInt(1, id);
            try (ResultSet resultSet = preparedStatement.executeQuery()) {
                if (resultSet.next()) {
                    service = new Service();
                    service.setDeptNo(resultSet.getInt("DEPTNO"));
                    service.setDname(resultSet.getString("DNAME"));
                    service.setLoc(resultSet.getString("LOC"));
                }
            }

        } catch (SQLException e) {
            e.printStackTrace(); }
```

ServiceDao.java

```java
    private static final String SELECT_ALL_SERVICES
        = "SELECT * FROM dept";
    @Override
    public ListServices getListeServices() {
        ListServices listServices = new ListServices();
        List<Service> services = new ArrayList<>();

        try (Connection connection = DataSource.getConnection();
             PreparedStatement preparedStatement =
                     connection.prepareStatement(SELECT_ALL_SERVICES);
             ResultSet resultSet = preparedStatement.executeQuery()) {

            while (resultSet.next()) {
                Service service = new Service();
                service.setDeptNo(resultSet.getInt("DEPTNO"));
                service.setDname(resultSet.getString("DNAME"));
                service.setLoc(resultSet.getString("LOC"));
                services.add(service);
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }

        listServices.setServices(services);
        return listServices;
    }
```

EmployeDao.java

```java
    private static final String SELECT_EMPLOYES_WITH_SERVICES =
     "SELECT e.*, d.DNAME, d.LOC FROM emp e JOIN dept d ON e.DEPTNO = d.DEPTNO";

    public ListEmployes getListeEmployes() {
        ListEmployes listEmployes = new ListEmployes();
        List<Employe> employes = new ArrayList<>();
        try (Connection connection = DataSource.getConnection();
             PreparedStatement preparedStatement = connection
                     .prepareStatement(SELECT_EMPLOYES_WITH_SERVICES);
             ResultSet resultSet = preparedStatement.executeQuery()) {
            while (resultSet.next()) {
                Employe employe = new Employe();
                employe.setEmpNo(resultSet.getInt("EMPNO"));
                employe.setNom(resultSet.getString("ENAME"));
                employe.setAge(resultSet.getInt("AGE"));
                employe.setFonction(resultSet.getString("JOB"));
                employe.setSalaire(resultSet.getDouble("SAL"));

                Service service = new Service();
                service.setDeptNo(resultSet.getInt("DEPTNO"));
                service.setDname(resultSet.getString("DNAME"));
                service.setLoc(resultSet.getString("LOC"));
                employe.setService(service);

                employes.add(employe);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }

        listEmployes.setEmployes(employes);
        return listEmployes;   }
}
```

EmployeDao.java

```java
package ma.dao;

import ma.model.Employe;

public class EmployeDao implements IEmployeDao {

    public EmployeDao() {
        }
    private static final String INSERT_EMPLOYE = "INSERT INTO emp "
            + "(ENAME, AGE, JOB, SAL, DEPTNO) VALUES (?, ?, ?, ?, ?)";

    @Override
    public Employe enregistrerEmploye(Employe employe) {
        try (Connection connection = DataSource.getConnection();
             PreparedStatement preparedStatement = connection
                     .prepareStatement(INSERT_EMPLOYE)) {

            preparedStatement.setString(1, employe.getNom());
            preparedStatement.setInt(2, employe.getAge());
            preparedStatement.setString(3, employe.getFonction());
            preparedStatement.setDouble(4, employe.getSalaire());

            preparedStatement.setInt(5, employe.getService().getDeptNo());

            preparedStatement.executeUpdate();

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return employe;
    }
```

ListEmployes.java

```java
package ma.model;

import java.util.List;

public class ListEmployes {

    private List<Employe> employes;

    public ListEmployes() {
        super();
    }

    public List<Employe> getEmployes() {
        return employes;
    }

    public void setEmployes(List<Employe> employes) {
        this.employes = employes;
    }
```

Employe.java

```java
package ma.model;

public class Employe {
    private int empNo;
    private String nom;
    private int age;
    private String fonction;
    private double salaire;
    private Service service;

    public Employe() {
        super();
    }

    //getters and setters
```

produits.jsp

```
<%
    ProduitModel model = (ProduitModel) request.getAttribute("model");
    List<Produit> produits = model.getProduits();
    ProduitDaoImpl produitDao = new ProduitDaoImpl();
    int nombreDeProduits = produitDao.countProduits();
%>
  <h1>Liste des produits</h1> <br>
  <p>Nombre total de produits : <%= nombreDeProduits %></p><br>

  <table border="1">
    <thead>
      <tr>
        <th>ID</th>
        <th>Désignation</th>
        <th>Prix</th>
        <th>Quantité</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
```

Produit.java

```
package metier.entities;

import java.io.Serializable;

public class Produit implements Serializable{

    private long id;
    private String designation;
    private double prix;
    private int quantite;

    //getters setters
```

editProduit.jsp

```
<body><%
    Produit produit = (Produit) request.getAttribute("produit");
    if (produit != null) {
%>

<form method="post">

    <label><strong>Désignation</strong></label><br>
    <input type="text" name="designation" value="<%= produit.getDesignation() %>"><br>
    <label><strong>Prix</strong></label><br>
    <input type="text" name="prix" value="<%= produit.getPrix() %>"><br><br>
    <label><strong>Quantité</strong></label><br>
    <input type="text" name="quantite" value="<%= produit.getQuantite() %>"><br><br>
    <button type="submit">Valider</button>
</form><%  } %></body>
```

saisieProduit.jsp

```
<body>
  <form action="save.do" method="post">
      <h1>Ajouter Produit</h1> <br>
      <label><strong>Désignation</strong></label><br><br>
       <input type="text" name="designation"><br><br>
       <label><strong>Prix</strong></label><br><br>
       <input type="text" name="prix"><br><br>
       <label><strong>Quantité</strong></label><br><br>
        <input type="text" name="quantite"><br><br>
       <button type="submit">Valider</button>
```

Controleur.java

```
public class Controleur extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private IProduitDao produitdao;

    public void init() throws ServletException{
        produitdao = new ProduitDaoImpl();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
            String path = request.getServletPath();

            if(path.equals("/index.do")) {

                ProduitModel model = new ProduitModel();
                List<Produit> produits = produitdao.getProduits();
                model.setProduits(produits);
                request.setAttribute("model", model);

                request.getRequestDispatcher("produits.jsp").forward(request, response);
            }
            else if(path.equals("/saisie.do")) {

                request.getRequestDispatcher("saisieProduit.jsp").forward(request, response);
            }

            else if(path.equals("/save.do") && (request.getMethod().equals("POST"))
                String designation = request.getParameter("designation");
                double prix = Double.parseDouble(request.getParameter("prix"));
                int quantite = Integer.parseInt(request.getParameter("quantite"));

                Produit p = produitdao.Save(new Produit(designation,prix,quantite))
                request.setAttribute("produit", p);
                response.sendRedirect("index.do");}
```

MODEL packagecom.model;
public class ClientCompte implements java.io.Serializable{
privateint identifiant; private String login;
private String password; private String libelle;
privatedouble solde; //Constructeur getters and setters ...}
INTERFACE DAO com.dao; //import ...
public interface IGestionCompteDAO {
ClientCompteLogin(String login, String password) throwsSQLException;}

Controleur.java

```
        else if (path.equals("/delete.do")) {
            long id = Long.parseLong(request.getParameter("id"));
            produitdao.Delete(id);
            response.sendRedirect("index.do");
        }

        else if (path.equals("/edit.do")) {
            if (request.getMethod().equals("GET")) {

                long id = Long.parseLong(request.getParameter("id"));
                Produit p = produitdao.getProduit(id);
                request.setAttribute("produit", p);
                request.getRequestDispatcher("editProduit.jsp").forward(request, response)
            } else if (request.getMethod().equals("POST")) {

                long id = Long.parseLong(request.getParameter("id"));
                String designation = request.getParameter("designation");
                double prix = Double.parseDouble(request.getParameter("prix"));
                int quantite = Integer.parseInt(request.getParameter("quantite"));

                Produit p = new Produit(designation, prix, quantite);
                p.setId(id);
                produitdao.Update(p);
                response.sendRedirect("index.do");
            }
        } else {
            response.sendError(Response.SC_NOT_FOUND);
        }}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    doGet(request, response);
}}
```

connection

```
public class SingletonConnection {

    private static Connection connection;
    static {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");   // Charger l
            connection = DriverManager.getConnection
```

connection suite

```
            ("jdbc:mysql://localhost:3306/db_produits", "root", "");
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
}
public static Connection getConnection() {
    return connection;
```

ProduitDaoImpl.java

```
    public Produit getProduit(long id) {
        Connection connection = SingletonConnection.getConnection();
        Produit produit = null;

        try { PreparedStatement ps =
                    connection.prepareStatement
                    ("SELECT * FROM PRODUITS WHERE ID = ?");
            ps.setLong(1, id);
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                produit = new Produit();
                produit.setId(rs.getLong("ID"));
                produit.setDesignation(rs.getString("DESIGNATION"));
                produit.setPrix(rs.getDouble("PRIX"));
                produit.setQuantite(rs.getInt("QUANTITE"));
            }
            rs.close();
            ps.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return produit;
    }
public Produit Save(Produit p) {
    Connection connection = SingletonConnection.getConnection();
    try {
        PreparedStatement ps =
                connection.prepareStatement
                ("INSERT INTO PRODUITS "
                    + "(DESIGNATION , PRIX , QUANTITE ) VALUES (? , ? , ?)");

        ps.setString(1,p.getDesignation());
        ps.setDouble(2,p.getPrix());
        ps.setInt(3, p.getQuantite());

        ps.executeUpdate();

        PreparedStatement ps2 =
                connection.prepareStatement
                ("SELECT MAX(ID) AS MAX_ID FROM PRODUITS");
        ResultSet rs = ps2.executeQuery();
        if(rs.next()) {
            p.setId(rs.getLong("MAX_ID"));
        }
        ps.close();

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return p;
```

Tp6 Exo2 :

```
public class GestionCompteDAO implements IGestionCompteDAO {
    public ClientCompte Login(String login, String password)
        throws SQLException {
    Connection connection = null;
    try {
        connection = ConnexionDB.getConnection();
        String query = "SELECT identifiant, libelle, "
            + "solde FROM comptes WHERE login=? AND password=?";

        try (PreparedStatement preparedStatement
                = connection.prepareStatement(query)) {
            preparedStatement.setString(1, login);
            preparedStatement.setString(2, password);

            try (ResultSet resultSet
                    = preparedStatement.executeQuery()) {
                if (resultSet.next()) {
                    ClientCompte clientCompte = new ClientCompte();
                    clientCompte.setIdentifiant(resultSet.getInt("identifiant"));
                    clientCompte.setLibelle(resultSet.getString("libelle"));
                    clientCompte.setSolde(resultSet.getDouble("solde"));
                    return clientCompte;
                }
            }}} catch (SQLException e) {
        e.printStackTrace();}
    return null; }  }
```

loginForm.jsp

```
<%@ page language="java"contentType="text/html;charset=UTF-8"pageEncoding="UTF-8"%><!DOCTYPE html>
<html><head><meta charset="UTF-8"><title>Formulairede Login</title></head><body>
<h2>Bienvenuesur l'application Bankonet</h2>
<form action="login.do"method="post">
<label for="login">Login :</label><input type="text" id="login"name="login"required>
<label for="password">Password :</label><input type="password"
id="password"name="password"required>
<input type="submit" value="Se Connecter">
```

compteDetails.jsp

```
<!DOCTYPE html><html><head><metacharset="UTF-8"><title>Détailsducompte</title></head>
<body><h2>Détailsducompte</h2>
<%sessionClientCompteclientCompte = (ClientCompte) session.getAttribute("clientCompte");
if (clientCompte != null) {%>
<table border="1">
<tr><th>Identifiant</th>
<td><%= clientCompte.getIdentifiant() %></td></tr>
<tr><th>Libellé</th>
<td><%= clientCompte.getLibelle() %></td></tr>
<tr><th>Solde</th>
<td><%= clientCompte.getSolde() %></td></tr>
</table><br><br>
<form action="logout.do"method="post"> <input type="submit" value="Déconnexion"></form>
<% }else { response.sendRedirect("erreur.jsp"); } %></body></html>
```

erreur.jsp

```
<body>
<h2>${message}</h2></body>
```

packagecom.controller; //import ...

```
public class CompteController extends HttpServlet {
private static final long serialVersionUID = 1L;
private IGestionCompteDAO gestionCompteDAO;
public void init(){
gestionCompteDAO = new GestionCompteDAO();}
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
String path = request.getServletPath();
if (path.equals("/login.do")) {
String login = request.getParameter("login");
String password = request.getParameter("password");
try { ClientCompte clientCompte = gestionCompteDAO.Login(login, password);
if (clientCompte != null) {
HttpSession session = request.getSession();
session.setAttribute("clientCompte", clientCompte);
response.sendRedirect("compteDetails.jsp");
} else {
request.setAttribute("message", "Client inconnu !");
request.getRequestDispatcher("erreur.jsp").forward(request, response);}}
catch (SQLException e) {
e.printStackTrace();
request.setAttribute("message", "Erreur lors de la récupération du compte");
request.getRequestDispatcher("erreur.jsp").forward(request, response);}}
else if (path.equals("/logout.do")) {
HttpSession session = request.getSession();
if(session != null) {
session.invalidate();
}
response.sendRedirect("loginForm.jsp"); } } }
```

ProduitDaoImpl.java

```
public int countProduits() {
    int count = 0;
    Connection connection = SingletonConnection.getC
    try {    PreparedStatement ps
        = connection.prepareStatement
        ("SELECT COUNT(*) AS COUNT FROM PRODUITS");
        ResultSet rs = ps.executeQuery();

        if (rs.next()) {
            count = rs.getInt("COUNT");
        }
        rs.close();
        ps.close();
    } catch (SQLException e) {
        e.printStackTrace();  }
    return count;
```

DaoImpl.java

```
public Produit Update(Produit p) {
    Connection connection = SingletonConnection.getConnection();

    try { PreparedStatement ps =
            connection.prepareStatement
            ("UPDATE PRODUITS SET DESIGNATION = ?, "
                + "PRIX = ?, QUANTITE = ? WHERE ID = ?");

        ps.setString(1,p.getDesignation());
        ps.setDouble(2,p.getPrix());
        ps.setInt(3, p.getQuantite());
        ps.setLong(4, p.getId());
        ps.executeUpdate();
        PreparedStatement ps2 = connection.prepareStatement
                ("SELECT MAX(ID) AS MAX_ID FROM PRODUITS");
        ResultSet rs = ps2.executeQuery();
        if(rs.next()) {
            p.setId(rs.getLong("MAX_ID"));
        }
        ps.close();
```

mpl.java

```
public void Delete(long id) {
    Connection connection = SingletonConnection.getConnection();
    try {
        PreparedStatement ps =
                connection.prepareStatement
                ("DELETE FROM PRODUITS WHERE ID = ?");
        ps.setLong(1, id);
        ps.executeUpdate();
        ps.close();
    } catch (SQLException e) {
        e.printStackTrace();
```