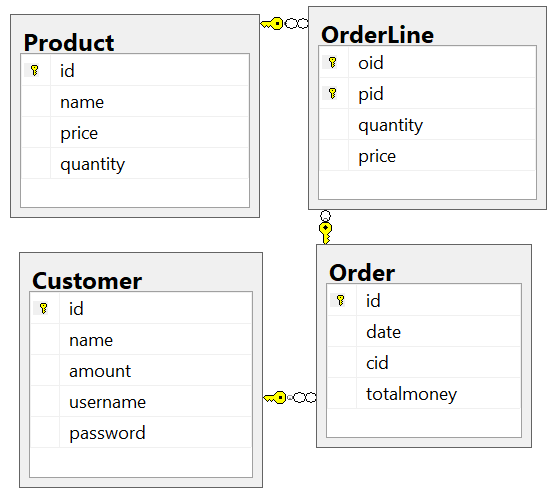
Database



**Các Model:**

|  |
| --- |
| Product.java |
| public class Product {  private int id;  private String name;  private double price;  private int quantity; |
| Customer.java |
| public class Customer {  private int id;  private String name;  private double amount;  private String username,password; |
| Order.java |
| public class Order {  private int id;  private String date;  private int cusid;  private double totalmoney; |
| OrderDetail.java |
| public class OrderDetail {  private int oid,pid;  private int quantity;  private double price; |
| Item.java |
| public class Item {  private Product product;  private int quantity;  private double price; |
| Cart.java |
| public class Cart {  private List<Item> items;  public Cart() {  items=new ArrayList<>();  }  public List<Item> getItems() {  return items;  }  public int getQuantityById(int id){  return getItemById(id).getQuantity();  }  private Item getItemById(int id){  for(Item i:items){  if(i.getProduct().getId()==id)  return i;  }  return null;  }  public void addItem(Item t){  if(getItemById(t.getProduct().getId())!=null){  Item m=getItemById(t.getProduct().getId());  m.setQuantity(m.getQuantity()+t.getQuantity());  }else  items.add(t);  }  public void removeItem(int id){  if(getItemById(id)!=null){  items.remove(getItemById(id));  }  }  public double getTotalMoney(){  double t=0;  for(Item i:items)  t+=(i.getQuantity()\*i.getPrice());  return t;  }  } |

**Phần dal**

|  |
| --- |
| CustomerDAO.java |
| public class CustomerDAO extends DBContext{  public Customer getAcount(String user,String pass){  String sql="select \* from customer "  + "where username=? and password=?";  try{  PreparedStatement st=con.prepareStatement(sql);  st.setString(1, user);  st.setString(2, pass);  ResultSet rs=st.executeQuery();  if(rs.next()){  return new Customer(rs.getInt("id"),  rs.getString("name"),  rs.getDouble("amount"),  rs.getString("username"),  rs.getString("password"));  }  }catch(SQLException e){  System.out.println(e);  }  return null;  }  } |
| ProductDAO.java |
| public class ProductDAO extends DBContext{  public List<Product> getAll(){  List<Product> list=new ArrayList<>();  String sql="select \* from Product";  try{  PreparedStatement st=con.prepareStatement(sql);  ResultSet rs=st.executeQuery();  while(rs.next()){  Product p=new Product(rs.getInt("id"),  rs.getString("name"),  rs.getDouble("price"),  rs.getInt("quantity"));  list.add(p);  }  }catch(SQLException e){  System.out.println(e);  }  return list;  }  public Product getProductByID(int id){  String sql="select \* from Product where id=?";  try{  PreparedStatement st=con.prepareStatement(sql);  st.setInt(1, id);  ResultSet rs=st.executeQuery();  if(rs.next()){  return new Product(id,  rs.getString("name"),  rs.getDouble("price"),  rs.getInt("quantity"));  }  }catch(SQLException e){  System.out.println(e);  }  return null;  }  } |

**Servlet**

|  |
| --- |
| ShopServlet.java |
| protected void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  ProductDAO pdb=new ProductDAO();  List<Product> prods=pdb.getAll();  Cookie[] arr = request.getCookies();  String txt = "";  if(arr!=null){  for (Cookie o : arr) {  if (o.getName().equals("cart")) {  txt = txt + o.getValue();  }  }  }  Cart cart=new Cart(txt,prods);  List<Item> list=cart.getItems();  int n;  if(list!=null)  n=list.size();  else  n=0;  //request.setAttribute("cart", cart);  request.setAttribute("size", n);  request.setAttribute("data", prods);  request.getRequestDispatcher("myshop.jsp").forward(request, response);  } |

**Phần jsp**

|  |
| --- |
| myshop.jsp |
| <style>  #bag{  text-align: right;  margin-right: 30px;  margin-top: 30px;  }  table{  border-collapse: collapse;  }  .ta{  text-align: right  }  a{  text-decoration: none;  }  </style> |
| <c:set var="size" value="${requestScope.size}"/>  Hello ${sessionScope.acount.name}  <p id="bag">  <img src="images/icon.png" width="50" height="50"/>  <a href="show">  Mybag (${size}) items  </a>  </p>  <h1>The list of products</h1>  <form name="f" action="" method="post">  Enter the number of items to by:  <input style="text-align: center" type="number" name="num" value="1"/>  <hr/>  <table border="1px" width="40%">  <tr>  <th>ID</th>  <th>Name</th>  <th>Quantity</th>  <th>Price</th>  <th>Action</th>  </tr>  <c:forEach items="${requestScope.data}" var="p">  <c:set var="id" value="${p.id}"/>  <tr>  <td>${id}</td>  <td>${p.name}</td>  <td class="ta">${p.quantity}</td>  <td class="ta">$<fmt:formatNumber pattern="##.#" value="${(p.price\*1.2)}"/></td>  <td style="text-align: center">    <input type="button" onclick="buy('${id}')" value="Buy item"/>  </td>  </tr>  </c:forEach>  </table>  </form> |
| <script type="text/javascript">  function buy(id){  var m=document.f.num.value;  document.f.action="buy?id="+id+"&num="+m;  document.f.submit();  }  </script> |

**Servlet**

|  |
| --- |
| BuyServlet.java |
| protected void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  ProductDAO pdb=new ProductDAO();  List<Product> prods=pdb.getAll();  Cookie arr[] = request.getCookies();  String txt = "";  if(arr!=null){  for (Cookie o : arr) {  if (o.getName().equals("cart")) {  txt = txt + o.getValue();  o.setMaxAge(0);  response.addCookie(o);  }  }  }  String num=request.getParameter("num");  String id=request.getParameter("id");  if (txt.isEmpty()) {  txt = id+":"+num;  } else {  txt = txt + "," + id+":"+num;  }  Cart cart=new Cart(txt,prods);  Cookie c = new Cookie("cart", txt);  c.setMaxAge(60 \* 60 \* 24);  response.addCookie(c);  List<Item> list=cart.getItems();  request.setAttribute("size", list.size());  request.setAttribute("data", prods);  request.getRequestDispatcher("shop").forward(request, response);  } |
| ShowServlet.java |
| protected void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  ProductDAO pdb=new ProductDAO();  List<Product> prods=pdb.getAll();  Cookie arr[] = request.getCookies();  String txt = "";  for (Cookie o : arr) {  if (o.getName().equals("cart")) {  txt = txt + o.getValue();  }  }  Cart cart=new Cart(txt,prods);  request.setAttribute("cart", cart);  request.getRequestDispatcher("mycart.jsp").forward(request, response);  } |

**Phần jsp**

|  |
| --- |
| mycart.java |
| <style>  a{  text-decoration: none;  color: chocolate;  font-size: 22px;  font-weight: bold;  }  table{  border-collapse: collapse;  }  .ta{  text-align: right  }  </style> |
| <h1>Shopping cart online</h1>  <table border="1px" width="40%">  <tr>  <th>No</th>  <th>Name</th>  <th>quantity</th>  <th>price</th>  <th>Total money</th>  <th>action</th>  </tr>  <c:set var="o" value="${requestScope.cart}"/>  <c:set var="t" value="0"/>  <c:forEach items="${o.items}" var="i">  <c:set var="t" value="${t+1}"/>  <tr>  <td>${t}</td>  <td>${i.product.name}</td>    <td style="text-align: center">  <button ><a href="process?num=-1&id=${i.product.id}">-</a></button>  <input type="text" readonly value="${i.quantity}"/>  <button><a href="process?num=1&id=${i.product.id}">+</a></button>  </td>  <td class="ta">  $<fmt:formatNumber pattern="##.#" value="${i.price}"/>  </td>  <td class="ta">  $<fmt:formatNumber pattern="##.#" value="${i.quantity\*i.price}"/>  </td>  <td style="text-align: center">  <form action="process" method="post">  <input type="hidden" name="id" value="${i.product.id}"/>  <input type="submit" value="Return item"/>  </form>  </td>  </tr>  </c:forEach>  </table>  <h3>Total money:$${o.getTotalMoney()}</h3>  <p></p>  <form action="checkout" method="post">  <input type="submit" value="Check out"/>  </form>  <hr/>  <a href="shop">CLICK ME CONTINUE SHOPPING</a> |

**Servlet**

|  |
| --- |
| ProcessServlet.java |
| protected void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  ProductDAO pdb=new ProductDAO();  List<Product> prods=pdb.getAll();  Cookie arr[] = request.getCookies();  String txt = "";  for (Cookie o : arr) {  if (o.getName().equals("cart")) {  txt = txt + o.getValue();  o.setMaxAge(0);  response.addCookie(o);  }  }  Cart cart=new Cart(txt, list);  String num\_raw=request.getParameter("num");  String id\_raw=request.getParameter("id");  int id,num=0;  try{  id=Integer.parseInt(id\_raw);  Product p=d.getProductById(id);  int numStore=p.getQuantity();  num=Integer.parseInt(num\_raw);  if(num==-1&&(cart.getQuantityById(id)<=1)){  cart.removeItem(id);  }else{  if((num==1) && cart.getQuantityById(id)>=numStore){  num=0;  }  double price=p.getPrice()\*2;  Item t=new Item(p, num, price);  cart.addItem(t);  }  }catch(NumberFormatException e){  }  List<Item> items=cart.getItems();  txt="";  if(items.size()>0){  txt=items.get(0).getProduct().getId()+":"+items.get(0).getQuantity();  for (int i = 1; i < items.size(); i++) {  txt=txt+","+items.get(i).getProduct().getId()+  ":"+items.get(i).getQuantity();  }  }  Cookie c=new Cookie("cart", txt);  c.setMaxAge(24\*60\*60);  response.addCookie(c);  request.setAttribute("cart", cart);  request.getRequestDispatcher("mycart.jsp").forward(request, response); } |
| protected void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  ProductDAO pdb=new ProductDAO();  List<Product> prods=pdb.getAll();  Cookie[] arr = request.getCookies();  System.out.println(Arrays.toString(arr));  String txt = "";  for (Cookie o : arr) {  if (o.getName().equals("cart")) {  txt = txt + o.getValue();  o.setMaxAge(0);  response.addCookie(o);  } }  String id = request.getParameter("id");  String ids[] = txt.split(",");  String txtOutPut = "";  for (int i = 0; i < ids.length; i++) {  String[] s=ids[i].split(":");  if (!s[0].equals(id)) {  if (txtOutPut.isEmpty()) {  txtOutPut = ids[i];  } else {  txtOutPut = txtOutPut + "," + ids[i];  }  }  }  if (!txtOutPut.isEmpty()) {  Cookie c = new Cookie("cart", txtOutPut);  c.setMaxAge(60 \* 60 \* 24);  response.addCookie(c);  }  Cart cart=new Cart(txtOutPut,prods);  request.setAttribute("cart", cart);  request.getRequestDispatcher("mycart.jsp").forward(request, response);  } |

**Module login**

|  |
| --- |
| Login.jsp |
| <h1>Login form</h1>  <h3 style="color: red">${requestScope.error}</h3>  <form action="login" method="get">  <table>  <tr>  <td>Enter username:</td>  <td><input type="text" name="user"/></td>  </tr>  <tr>  <td>Enter password:</td>  <td><input type="password" name="pass"/></td>  </tr>  <tr>  <td>  </td>  <td><input type="submit" value="Login" /></td>  </tr>  </table>  </form> |
| LoginServlet.java |
| protected void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  String u=request.getParameter("user");  String p=request.getParameter("pass");  CustomerDAO ad=new CustomerDAO();  Customer a=ad.getAcount(u, p);  if(a==null){  String er="username: "+u+" and password: "+p+" don't exsited!";  request.setAttribute("error", er);  request.getRequestDispatcher("login.jsp").forward(request, response);  }else{  HttpSession session=request.getSession(true);  session.setAttribute("acount", a);  request.getRequestDispatcher("shop").forward(request, response);  }  } |

**Tiếp phần dal**

|  |
| --- |
| OrderDAO.java |
| public class OrderDAO extends DBContext{  public void addOrder(Customer u, Cart cart) {  LocalDate curDate = java.time.LocalDate.now();  String date = curDate.toString();  try{  //add vào bảng Order  String sql="insert into [order] values(?,?,?)";  PreparedStatement st=con.prepareStatement(sql);  st.setString(1, date);  st.setInt(2, u.getId());  st.setDouble(3, cart.getTotalMoney());  st.executeUpdate();  //lấy ra id của Order vừa add  String sql1="select top 1 id from [Order] order by id desc";  PreparedStatement st1=con.prepareStatement(sql1);  ResultSet rs=st1.executeQuery();  //add vào bảng OrderLine  if(rs.next()){  int oid=rs.getInt(1);  for(Item i:cart.getItems()){  String sql2="insert into [orderline] values(?,?,?,?)";  PreparedStatement st2=con.prepareStatement(sql2);  st2.setInt(1, oid);  st2.setInt(2, i.getProduct().getId());  st2.setInt(3, i.getQuantity());  st2.setDouble(4, i.getPrice());  st2.executeUpdate();  }  }  //update so luong trong bang Product sau khi ban  String sql3 = "update Product set quantity = quantity - ? "  + "where id = ?";  PreparedStatement st3=con.prepareStatement(sql3);  for(Item i:cart.getItems()){  st3.setInt(1, i.getQuantity());  st3.setInt(2, i.getProduct().getId());  st3.executeUpdate();  }  }catch(SQLException e){  System.out.println(e);  }  }  } |

**Servlet**

|  |
| --- |
| CheckoutServlet.java |
| protected void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  ProductDAO pdb=new ProductDAO();  List<Product> prods=pdb.getAll();  Cookie arr[] = request.getCookies();  String txt = "";  for (Cookie o : arr) {  if (o.getName().equals("cart")) {  txt = txt + o.getValue();  }  }  Cart cart=new Cart(txt,prods);  HttpSession session =request.getSession(true);  Customer acount=null;  Object a=session.getAttribute("acount");  if(a!=null){  acount=(Customer)a;  OrderDAO odb=new OrderDAO();  odb.addOrder(acount, cart);  Cookie c = new Cookie("cart", "");  c.setMaxAge(0);  response.addCookie(c);  request.setAttribute("size", 0);  request.setAttribute("data", prods);  request.getRequestDispatcher("myshop.jsp").forward(request, response);  }else{  response.sendRedirect("login.jsp");  }  } |