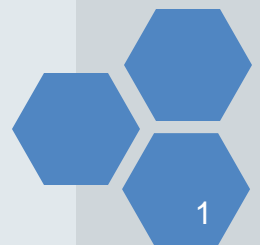


CÔNG NGHỆ JAVA

Nguyễn Hữu Thể

Bài 3: MVC





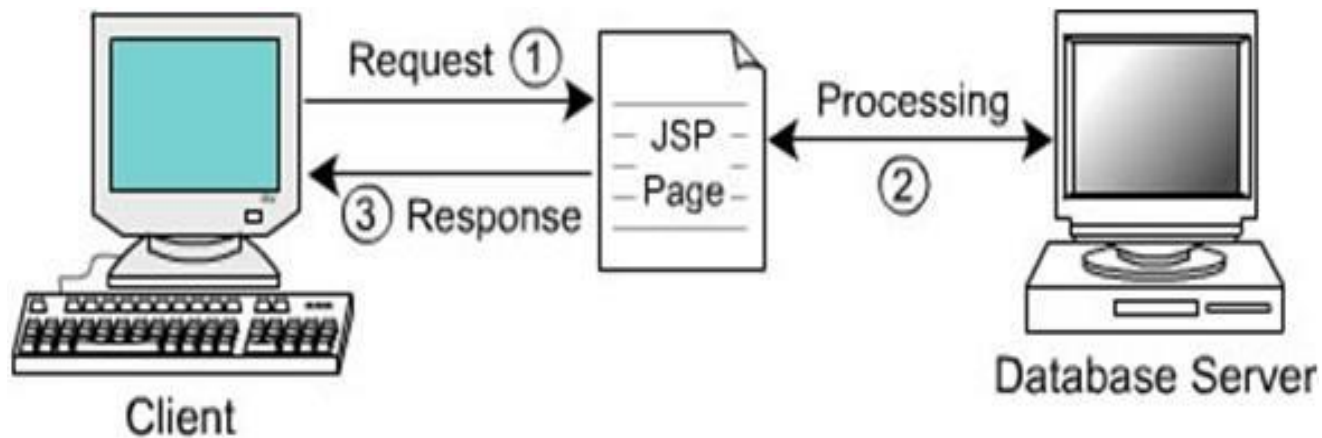
Nội dung

- Mô hình lập trình truyền thống (Mô hình 1)
- Mô hình MVC (Mô hình 2)
- Ứng dụng MVC minh họa



Mô hình 1 (Model 1)

- Khi thiết kế trang JSP, PHP hay ASP, người lập trình thường đan xen các mã html cùng với các mã JSP, PHP, hay ASP.
- Do vậy, có những khó khăn sau có thể gặp phải:
 - Người thiết kế giao diện cũng cần phải biết ngôn ngữ lập trình.
 - Việc bảo trì chúng thường rất khó khăn, vì một phần các mã chương trình lẫn lộn với mã html.
 - Khi có lỗi xảy ra, tìm và định vị lỗi cũng khó khăn.





Model 1 - Ví dụ

- Minh họa kỹ thuật chuyển dữ liệu sang trang trong JSP. Kiểm tra thông tin đăng nhập, với dữ liệu kiểm tra (username = “cnttk3” và password = “cnttk3”)
- JSP xử lý dữ liệu theo Model 1 (2 layer)
- Class Data.java: chứa phương thức kiểm tra đăng nhập checkLogin(,)
- File JSP:
 - Login.jsp: giao diện đăng nhập
 - Process-login.jsp: trang xử lý kết quả đăng nhập, gọi phương thức checkLogin(,) ở class Data, nhận kết quả trả về



Model 1 - Ví dụ: Mã nguồn

- **Data.java**: chứa trong **package** `org.dhcl.basic`

```
package org.dhcl.basic;  
public class Data {  
    //Hàm kiểm tra user và pass là chữ cnttk3  
    public static boolean checkLogin(String user, String pass){  
        if(user.equals("cnttk3") && pass.equals("cnttk3"))  
            return true;  
        else  
            return false;  
    }  
}
```



Model 1 - Ví dụ: Mã nguồn

– Login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<html>
<body>
<form action="process-login.jsp" method="post">
    Username: <input type="text" name="user"><br>
    Password: <input type="password" name="pass"><br>
    <input type="submit" value="Login">
</form>
</body>
</html>
```

The screenshot shows a web browser window with the URL `http://localhost:8080/CDWHG/basic/login4.jsp`. The browser's address bar and search bar are visible. The main content area displays a login form with the following elements:

- A label "Username:" followed by a text input field.
- A label "Password:" followed by a password input field.
- A "Login" button below the password field.

login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<html>
<body>
<form action="process-login.jsp" method="post">
    Username: <input type="text" name="user"><br>
    Password: <input type="password" name="pass"><br>
    <input type="submit" value="Login">
</form>
</body>
</html>
```

process-login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ page import="org.dhcl.basic.*" %>
<%
String user = request.getParameter("user");
String pass = request.getParameter("pass");
if(Data.checkLogin(user, pass))
    out.print("Đăng nhập thành công");
else
    out.print("Đăng nhập không thành công");
%>
```



http://localhost:8080/CDWHG/basic/login4.jsp

http://localhost:8080/CDWHG/ Google

Username:

Password:

Login

http://localhost:8080/CDWHG/basic/process-login4.jsp

http://localhost:8080/CDWHG/ Google

Đăng nhập thành công

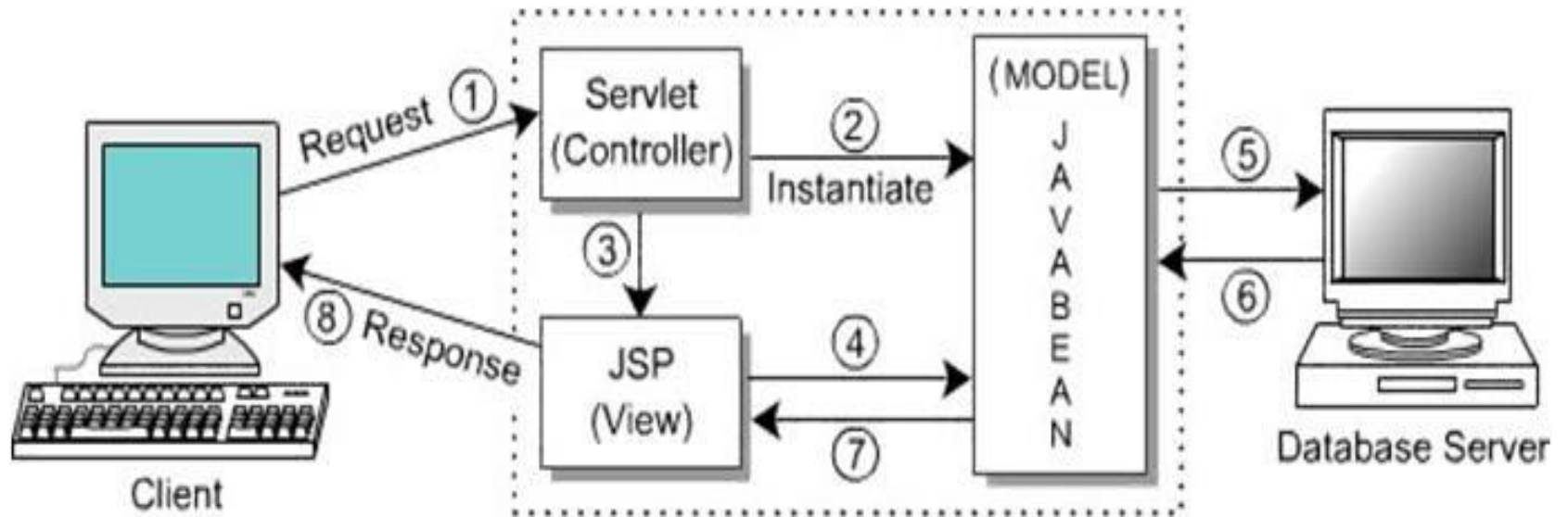


Mô hình MVC (Model 2)

- Mô hình 2 hay mô hình MVC (Model-View-Controller)
- Tương ứng với một trang JSP, tách thành 3 thành phần:
 - Model-View-Controller.
- Các thành phần trên làm việc như sau:
 - Model: là các lớp java có nhiệm vụ:
 - Nhận các yêu cầu từ khung nhìn
 - Thi hành các yêu cầu (tính toán, truy vấn database,...)
 - Trả về các kết quả yêu cầu cho View.
 - View: JSP hiển thị form nhập, các kết quả trả về từ Model.
 - Controller: Đồng bộ hoá giữa Model và View. Tức là với một trang JSP này thì sẽ tương ứng với lớp Java nào để xử lý nó và ngược lại, kết quả sẽ trả về trang JSP nào.



Mô hình MVC (Model 2)





MVC - Ví dụ:

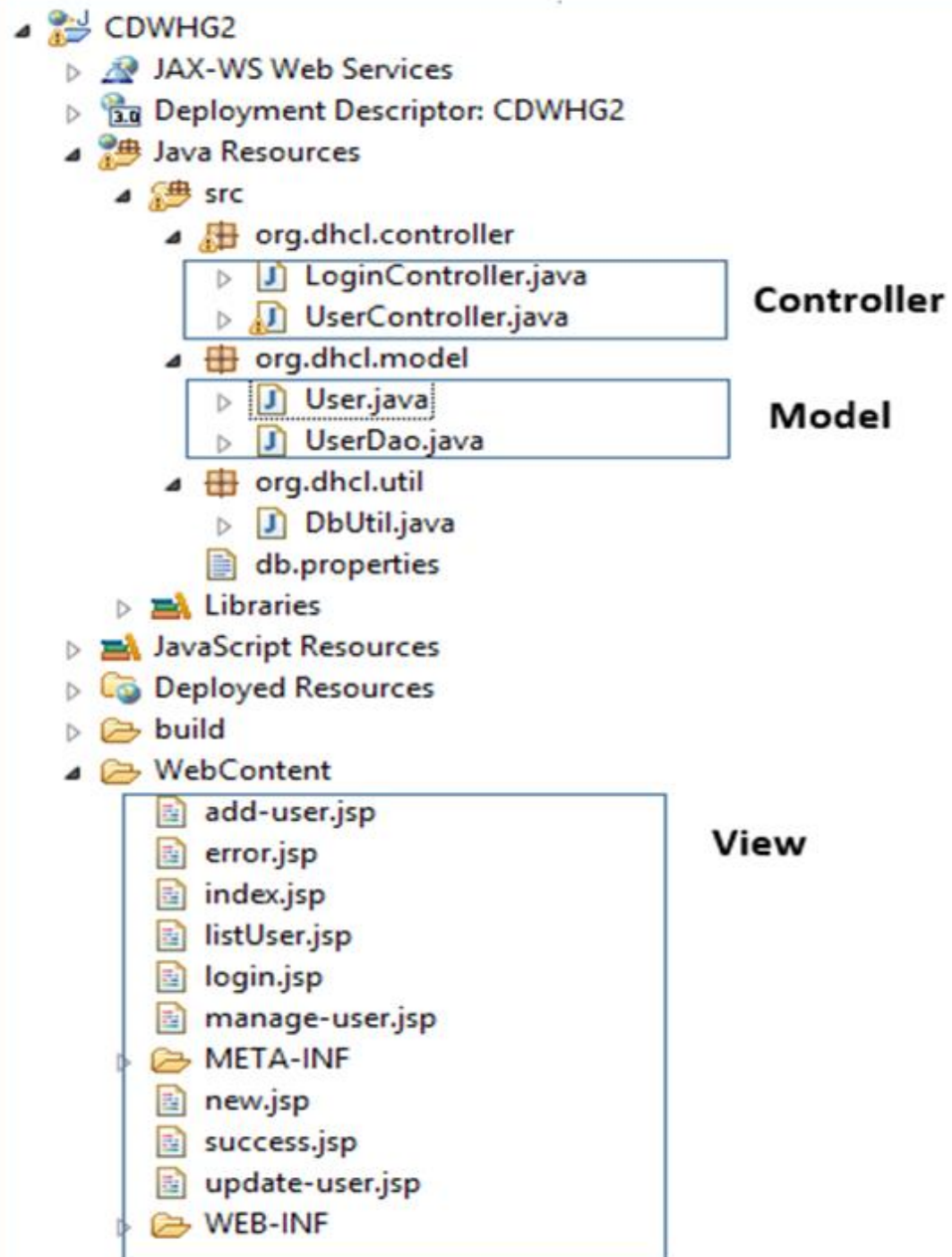
- Ứng dụng quản lý dữ liệu của 1 table Users, bao gồm đăng nhập, quản lý dữ liệu: trình bày dữ liệu, thêm mới, xóa dữ liệu, sửa dữ liệu, xóa dữ liệu.
- Tạo database: Ví dụ: K3MVC, tạo 1 table Users như sau:

```
create database K3MVC;  
use K3MVC;  
create table Users(  
    userid int AUTO_INCREMENT not null primary key,  
    username varchar(30) not null,  
    password varchar(30) not null  
);  
insert into Users (username, password) values  
( 'admin','123456'),  
( 'user1','123456'),  
( 'user2','123456');
```

userid	username	password
1	admin	123456
2	user1	123456
3	user2	123456



Cấu trúc Project





Thiết lập file cấu hình kết nối CSDL

db.properties

```
driver=com.mysql.jdbc.Driver  
url=jdbc:mysql://localhost:3306/k3mvc  
user=root  
password=
```

Hệ quản trị CSDL MySQL:

- *Tên database: “k3mvc”*,
- *Username: root”*,
- *Password: rỗng*

DbUtil.java: class thư viện kết nối CSDL

```
package org.dhcl.util;
import java.io.*;
import java.sql.*;
import java.util.Properties;
public class DbUtil {
    private static Connection connection = null;
    public static Connection getConnection() {
        //Dùng kết nối đến Database, chuỗi kết nối lưu trong file db.properties
        if (connection != null)
            return connection;
        else{
            try{
                Properties pro = new Properties();
                InputStream inputStream =
                    DbUtil.class.getClassLoader().getResourceAsStream("/db.properties");
                pro.load(inputStream);
                String driver = pro.getProperty("driver");
                String url = pro.getProperty("url");
                String user = pro.getProperty("user");
                String password = pro.getProperty("password");
                Class.forName(driver);
                connection = DriverManager.getConnection(url, user, password);
            }catch(ClassNotFoundException e) { e.printStackTrace();
            }catch(SQLException e) { e.printStackTrace();
            }catch(FileNotFoundException e) { e.printStackTrace();
            }catch(IOException e) { e.printStackTrace(); }
        }
        return connection;
    }
}
```



DbUtil.java

```
//Cách kết nối CSDL thứ 2 (Chọn 1 trong 2 cách kết nối)
public static Connection getConnection2() {
    if (connection != null)
        return connection;
    else {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/k3mvc", "root", "");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return connection;
    }
}
```



User.java: Lớp User

```
package org.dhcl.model;
public class User {
    private int userid;
    private String username;
    private String password;

    public int getUserid() {
        return userid;
    }

    public void setUserid(int userid) {
        this.userid = userid;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String
username) {
        this.username = username;
    }
}
```

```
public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

// Constructor có đầy đủ tham số, để khởi tạo dữ
liệu cho class
public User(int userid, String username, String
password) {
    this.userid = userid;
    this.username = username;
    this.password = password;
}

public User() { // Constructor không tham số
    this.userid = 0;
    this.username = "";
    this.password = "";
}
}
```




UserDao.java

```
public class UserDao {
    Connection connection = null;
    public UserDao(){
        connection = DbUtil.getConnection(); //Mỗi lần khởi tạo biến đối tượng
thuộc class UserDao thì khởi tạo luôn kết nối đến database
    }
    public boolean checkLogin(String user, String pass){
        try {
            Statement st = connection.createStatement();
            String sql = "select * from Users where username = '"+user+
                "' and password = '" + pass + "'";
            ResultSet rs = st.executeQuery(sql);
            if(rs.next())
                return true;
            else
                return false;
        } catch (SQLException e) { //Không thể kết nối
            e.printStackTrace();
            return false;
        }
    }
}
```



LoginController

```
public class LoginController extends HttpServlet {  
    protected void doPost(HttpServletRequest request, HttpServletResponse  
    response) throws ServletException, IOException {  
        //Nhận dữ liệu chuyển đến từ trang login.jsp  
        String username = request.getParameter("username");  
        String password = request.getParameter("password");  
        RequestDispatcher view = null;  
        //Kiểm tra user và pass, gọi hàm kiểm tra đăng nhập ở Authenticate.java  
        UserDao u = new UserDao();  
        if(u.checkLogin(username, password)){  
            //Gọi success.jsp nếu user và pass đúng  
            view = request.getRequestDispatcher("success.jsp");  
            User user = new User();  
            user.setUsername(username);  
            user.setPassword(password);  
            request.setAttribute("user_request", user); //Khởi tạo 1 biến  
                //user_request để lưu trữ thông tin user đăng nhập thành công  
                //cho đối tượng request quản lý, giúp triệu gọi thông tin  
                //đăng nhập ở những trang sau.  
        }  
    }  
}
```



LoginController

else *//Gọi error.jsp nếu user và pass sai*

```
view = request.getRequestDispatcher("error.jsp");
```

//Gọi chuyển trang đến success.jsp hoặc error.jsp

```
view.forward(request, response);
```

```
}
```

```
}
```

```
}
```



Login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<html>
<body>
<form action="LoginController" method="post">
    Username: <input type="text" name="username"><br>
    Password: <input type="password" name="password"><br>
    <input type="submit" value="Login">
</form>
</body></html>
```



success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ page import="org.dhcl.model.*" %>
<html>
<body>
Đăng nhập thành công<br>
<%
User user = (User)request.getAttribute("user_request");
out.print("Chào bạn " + user.getUsername());
%>
</body></html>
```

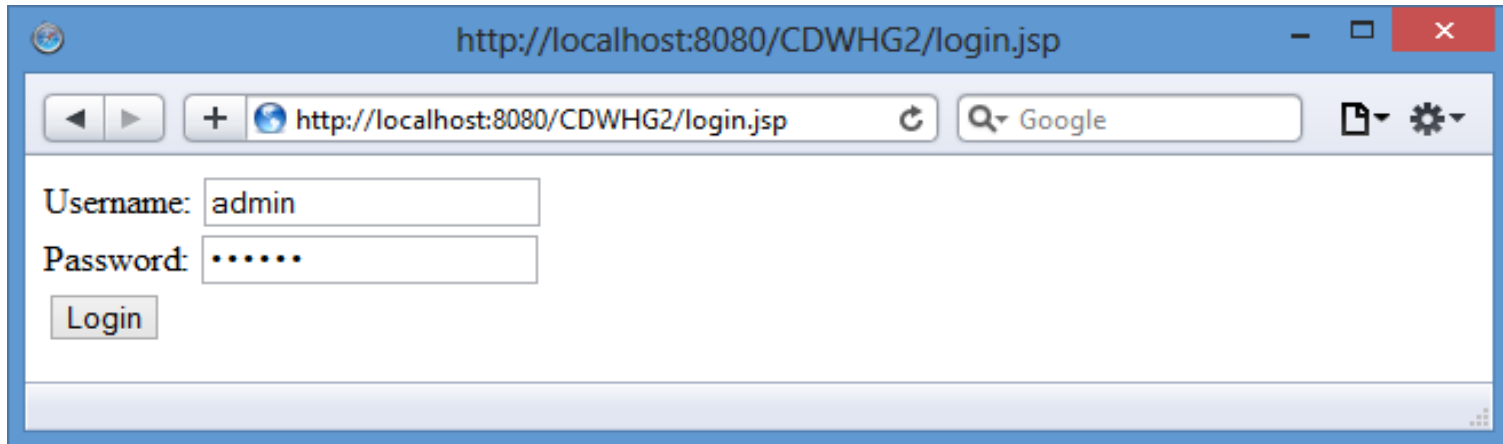
error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<html>
<body>
    Thất bại
</body></html>
```

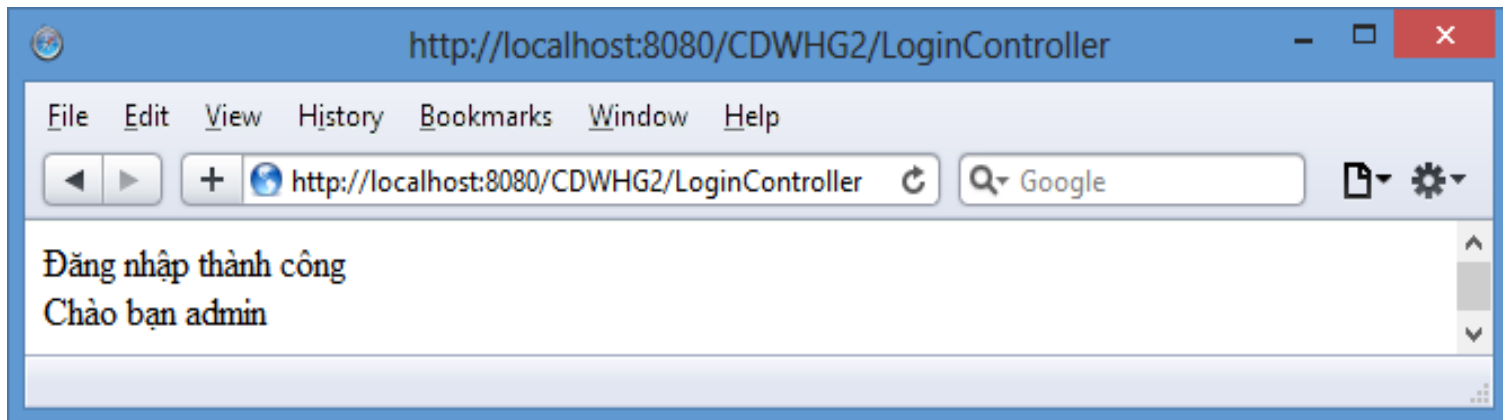


Giao diện thực thi ứng dụng

– Giao diện trang login.jsp



– Thông báo thành công (gọi trang success.jsp bên trong UserController)





– Tiếp tục thực hiện các xử lý cho Users

- Quản lý dữ liệu
- Thêm
- Sửa dữ liệu
- Xóa dữ liệu
- ...



Bổ sung các phương thức xử lý cho các class

UserDao.java

UserController.java

UserDao.java

```
public class UserDao {
    Connection connection = null;
    public UserDao(){
        connection = DbUtil.getConnection();
    }
    public boolean checkLogin(String user, String pass){
        //Kiểm tra thông tin user đăng nhập có tồn tại trong CSDL?
    }
    public void addUser(User user){
        //Thêm dữ liệu vào table Users
    }
    public ResultSet selectUser(){
        //Lấy toàn bộ dữ liệu từ 1 table
    }
    public void deleteUser(int userid){
        //Xóa dữ liệu dựa vào mã số userid của table User
    }
    public void updateUser(User user){
        //Phương thức cập nhật dữ liệu vào table Users, đầu vào là 1 đối
        //tượng Users đã có dữ liệu: userid, username, password.
    }
    public User getUserById(int userid){
        //Phương thức lấy thông tin của 1 user dựa vào mã số userid
        //là đầu vào, trả về 1 đối tượng User
    }
    public List<User> getAllUsers() {
        //Lấy tất cả user
    }
}
```

UserDao.java

```
public class UserDao {  
    //...  
    public boolean checkLogin(String user, String pass){  
        try {  
            Statement st = connection.createStatement();  
            String sql = "select * from Users where username = '"+user+  
                "' and password = '" + pass + "'";  
            ResultSet rs = st.executeQuery(sql);  
            if(rs.next())  
                return true;  
            else  
                return false;  
        } catch (SQLException e) { //Không thể kết nối  
            e.printStackTrace();  
            return false;  
        }  
    }  
    //...  
}
```

UserDao.java

```
public class UserDao {  
    //...  
    //Hàm thêm dữ liệu vào table Users  
    public void addUser(User user){  
        try{  
            Statement st = connection.createStatement();  
            String sql = "insert into Users (username, password)" +  
                " values (" + user.getUsername() + "," + user.getPassword() + ")";  
            st.executeUpdate(sql);  
        }catch(SQLException e){  
            e.printStackTrace();  
        }  
    }  
    //...  
}
```

UserDao.java

```
public class UserDao {  
    //...  
    //Lấy toàn bộ dữ liệu từ 1 table  
    public ResultSet selectUser(){  
        try{  
            Statement st = connection.createStatement();  
            String sql = "select * from Users";  
            ResultSet rs = st.executeQuery(sql);  
            return rs;  
        }catch(SQLException e){  
            e.printStackTrace();  
            return null;  
        }  
    }  
    //...  
}
```

UserDao.java

```
public class UserDao {  
    //...  
    //Phương thức xóa dữ liệu dựa vào mã số userid của table User  
    public void deleteUser(int userid){  
        try{  
            Statement st = connection.createStatement();  
            String sql = "delete from Users where userid = " + userid;  
            st.executeUpdate(sql);  
        }catch(SQLException e){  
            e.printStackTrace();  
        }  
    }  
    //...  
}
```

UserDao.java

```
public class UserDao {  
    //...  
    //Phương thức cập nhật dữ liệu vào table Users, đầu vào là 1 đối  
    //tượng Users đã có dữ liệu: userid, username, password.  
    //Phương thức không trả về dữ liệu  
    public void updateUser(User user){  
        try{  
            Statement st = connection.createStatement();  
            String sql = "update Users set username = '" +  
                user.getUsername() + "', password = '" +  
                user.getPassword() + "' where userid = " +  
                user.getUserid();  
            st.executeUpdate(sql);  
        }catch(SQLException e){  
            e.printStackTrace();  
        }  
    }  
    //...  
}
```

UserDao.java

```
public class UserDao {  
    //...  
    //Phương thức lấy thông tin của 1 user dựa vào mã số userid  
    //là đầu vào, trả về 1 đối tượng User  
    public User getUserById(int userid){  
        User user = new User();  
        try{  
            Statement st = connection.createStatement();  
            String sql = "select * from users where userid = " + userid;  
            ResultSet rs = st.executeQuery(sql);  
            if (rs.next()) {  
                user.setUserid(rs.getInt("userid"));  
                user.setUsername(rs.getString("username"));  
                user.setPassword(rs.getString("password"));  
            }  
            //Cần khai báo biến đối tượng user cục bộ trong hàm  
            //set hay get dữ liệu vào user?  
        }catch(SQLException e){  
            e.printStackTrace();  
        }  
        return user;  
    }  
    //...  
}
```

UserDao.java

```
public class UserDao {  
    //...  
    public List<User> getAllUsers() {  
        List<User> users = new ArrayList<User>();  
        try {  
            Statement statement = connection.createStatement();  
            ResultSet rs = statement.executeQuery("select * from users");  
            while (rs.next()) {  
                User user = new User();  
                user.setUserid(rs.getInt("userid"));  
                user.setUsername(rs.getString("username"));  
                user.setPassword(rs.getString("password"));  
                users.add(user);  
            }  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
        return users;  
    }  
    //...  
}
```


Controller: UserController.java

```
public class UserController extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
    UserDao dao = null;  
    public UserController() {  
        super();  
        dao = new UserDao(); //New UserDao  
    }  
    //Dữ liệu của edit và delete được nhận trong hàm doGet  
    protected void doGet(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {  
        // request action  
        String action = request.getParameter("action");  
        int userid = 0;  
        RequestDispatcher view = null;  
        UserDao u = new UserDao();  
        if(action.equals("all")){  
            request.setAttribute("users", dao.getAllUsers());  
            view = request.getRequestDispatcher("listUser.jsp");  
        }  
    }  
}
```

```
else if(action.equals("insert")){
    view = request.getRequestDispatcher("add-user.jsp");
}
else if(action.equals("edit")){
    userid = Integer.parseInt(request.getParameter("userid"));
    User user = u.getUserById(userid);
    request.setAttribute("userUpdate", user);
    view = request.getRequestDispatcher("update-user.jsp");
}
else if(action.equals("delete")){
    userid = Integer.parseInt(request.getParameter("userid"));
    u.deleteUser(userid);
    view = request.getRequestDispatcher("manage-user.jsp");
}
//Gọi chuyển trang
view.forward(request, response);
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    User user = new User();
    user.setUsername(request.getParameter("username"));
    user.setPassword(request.getParameter("password"));
    String userid = request.getParameter("userid");
    if (userid == null || userid.isEmpty()) {
        dao.addUser(user);
    } else {
        user.setUserId(Integer.parseInt(userid));
        dao.updateUser(user);
    }
    RequestDispatcher view =
        request.getRequestDispatcher("manage-user.jsp");
    view.forward(request, response);
}
}
```



Quản lý dữ liệu: table Users

index.jsp: Lần đầu gọi trang index.jsp => Trang sẽ chuyển sang Controller điều phối với *action=all*

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<html>
<body>
    <jsp:forward page="/UserController?action=all" />
</body>
</html>
```

UserController chuyển sang View: manage-user.jsp

```
<%@ page import="org.dhcl.model.*" %>
<%@ page import="java.sql.*" %>
<html>
<body>
<p><a href="UserController?action=insert">Add User</a></p>
<table border="1">
<tr><td>Userid</td><td>Username</td><td>Password</td><td>Edit</td><td>Delete</td></tr>
<%
 UserDao userDao = new UserDao();
 ResultSet rs = userDao.selectUser(); //selectUser() đã viết trong class UserDao
 while(rs.next()){ %>
<tr>
<td><%= rs.getInt(1) %></td>
<td><%= rs.getString(2) %></td>
<td><%= rs.getString(3) %></td>
<td><a href="UserController?action=edit&userid=
<%=rs.getInt(1)%>">Edit</a></td>
<td><a href="UserController?action=delete&userid=
<%=rs.getInt(1)%>">Delete</a></td>
</tr>
<%} %>
</table>
</body>
</html>
```





Giao diện trang quản lý User: manage-user.jsp

http://localhost:8080/CDWHG2/manage-user.jsp

http://localhost:8080/CDWHG2/manage-user.jsp

Google

Add User

User id	Username	Password	Edit	Delete
1	admin	123456	Edit	Delete
13	user1	123457	Edit	Delete
14	user2	123456	Edit	Delete

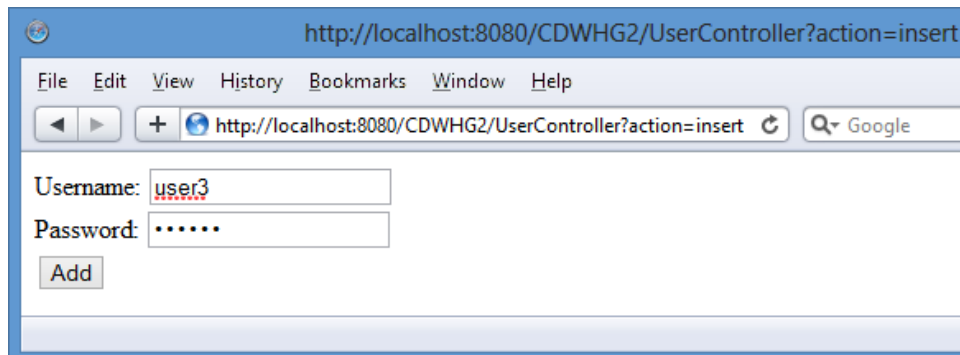
Go to "http://localhost:8080/CDWHG2/UserController?action=insert"



add-user.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<html>
<body>
<form action="UserController" method="post">
    Username: <input type="text" name="username"><br>
    Password: <input type="password" name="password"><br>
    <input type="submit" value="Add">
</form>
</body>
</html>
```

Giao diện trang add-user.jsp được gọi bên trong Controller khi người dùng chọn liên kết Add User từ trang manage-user.jsp





add-user.jsp

Giao diện trang add-user.jsp được gọi bên trong Controller khi người dùng chọn liên kết Add User từ trang manage-user.jsp

http://localhost:8080/CDWHG2/UserController?action=insert

File Edit View History Bookmarks Window Help

http://localhost:8080/CDWHG2/UserController?action=insert Google

Username:

Password:

Add

Sau khi đồng ý, nhấn Add => 1 dòng dữ liệu mới được thêm vào database, Controller gọi View manage-user.jsp trình bày dữ liệu vừa thêm vào.

http://localhost:8080/CDWHG2/UserController

File Edit View History Bookmarks Window Help

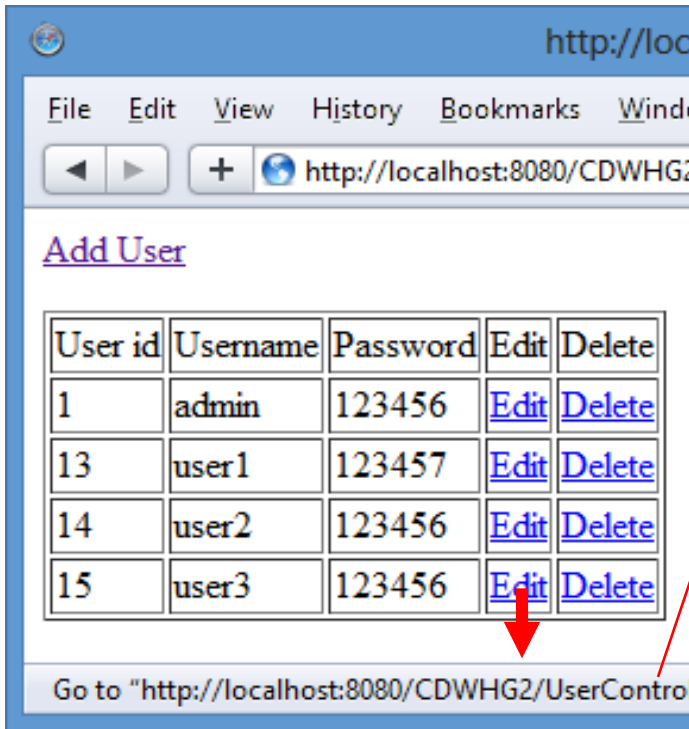
http://localhost:8080/CDWHG2/UserController Google

Add User

User id	Username	Password	Edit	Delete
1	admin	123456	Edit	Delete
13	user1	123457	Edit	Delete
14	user2	123456	Edit	Delete
15	user3	123456	Edit	Delete

Sửa 1 dòng dữ

Sửa 1 dòng dữ liệu, rê ch



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/CDWHG2/UserController?action=edit&userid=15`. The main content area displays a table titled "Add User" with the following data:

User id	Username	Password	Edit	Delete
1	admin	123456	Edit	Delete
13	user1	123457	Edit	Delete
14	user2	123456	Edit	Delete
15	user3	123456	Edit	Delete

A red arrow points from the "Edit" link in the last row of the table to the corresponding code in the adjacent block.

```
public class UserController extends HttpServlet {  
    protected void doGet(...) {  
        int userid = 0;  
        RequestDispatcher view = null;  
        UserDao u = new UserDao();  
        if (action.equals("all")) {  
            request.setAttribute("users", dao.getAllUsers());  
            view = request.getRequestDispatcher("listUser.jsp");  
        } else if (action.equals("insert")) {  
            view = request.getRequestDispatcher("add-  
            user.jsp");  
        } else if (action.equals("edit")) {  
            userid =  
            Integer.parseInt(request.getParameter("userid"));  
            User user = u.getUserById(userid);  
            request.setAttribute("userUpdate", user);  
            view = request.getRequestDispatcher("update-  
            user.jsp");  
        }  
    }  
}
```



Sửa 1 dòng dữ liệu

Chuyển dữ liệu của dòng người dùng muốn sửa, trình bày dữ liệu ra giao diện cập nhật

The screenshot shows a web browser window with the address bar containing the URL: `http://localhost:8080/CDWHG2/UserController?action=edit&userid=15`. The browser's menu bar includes File, Edit, View, History, Bookmarks, Window, and Help. The address bar also shows a search engine (Google) and navigation icons. The main content area displays a form with the following fields:

- Userid:
- Username:
- Password:

Below the form is an button.

Để có được giao diện này, chúng ta phải tạo 1 file **update-user.jsp** để trình bày dữ liệu cần cập nhật



update-user.jsp

```
<%@ page import="org.dhcl.model.*" %>
<%@ page import="java.sql.*" %>
<html>
<body>
<%
//gọi biến userUpdate đã lưu khi xử lý ở Controller
User userUpdate = (User)request.getAttribute("userUpdate");
%>
<form action="UserController" method="post">
    Userid: <input type="text" name="userid"
        readonly value="<%=userUpdate.getUserid() %>"><br>
    Username: <input type="text" name="username"
        value="<%=userUpdate.getUsername() %>"><br>
    Password: <input type="password" name="password"
        value="<%=userUpdate.getPassword() %>"><br>
    <input type="submit" value="Update">
</form>
</body></html>
```



update-user.jsp

Giả sử chúng ta sửa password lại thành 123458, và nhấn nút Update

http://localhost:8080/CDWHG2/UserController?action=edit&userid=15

File Edit View History Bookmarks Window Help

http://localhost:8080/CDWHG2/UserController?action=edit&userid=15 Google

Userid: 15

Username: user3

Password:

Update

Kết quả xử lý như sau:

http://localhost:8080/CDWHG2/UserController

File Edit View History Bookmarks Window Help

http://localhost:8080/CDWHG2/UserController Google

[Add User](#)

User id	Username	Password	Edit	Delete
1	admin	123456	Edit	Delete
13	user1	123457	Edit	Delete
14	user2	123456	Edit	Delete
15	user3	123458	Edit	Delete

```
public class UserController extends HttpServlet {
```

```
//...
```

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

```
    User user = new User();
```

```
    user.setUsername(request.getParameter("username"));
```

```
    user.setPassword(request.getParameter("password"));
```

```
    String userid = request.getParameter("userid");
```

```
    if (userid == null || userid.isEmpty()) {
```

```
        dao.addUser(user);
```

```
    } else {
```

```
        user.setUserid(Integer.parseInt(userid));
```

```
        dao.updateUser(user);
```

```
    }
```

```
    RequestDispatcher  
        request.getRequestDispatcher("forward.jsp")
```

```
        request.getRequestDispatcher("forward.jsp")
```

```
        view.forward(request, response);
```

```
    }
```

```
}
```

```
public class UserDao {
```

```
//...
```

```
public void updateUser(User user){
```

```
    try{
```

```
        Statement st = connection.createStatement();
```

```
        String sql = "update Users set username = " +
```

```
            user.getUsername() + ", password = " +
```

```
            user.getPassword() + " where userid = " +
```

```
            user.getUserid();
```

```
        st.executeUpdate(sql);
```



Xóa dữ

Rê chuột vào liên

User id	Username	Password	Edit	Delete
1	admin	123456	Edit	Delete
13	user1	123457	Edit	Delete
14	user2	123456	Edit	Delete
15	user3	123458	Edit	Delete

Go to "http://localhost:8080/CDWHG2/UserController?action=delete&userid=15"

```
public class UserController extends HttpServlet {
```

```
//...
```

```
protected void doGet(...){
```

```
//...
```

```
if(action.equals("delete")){
```

```
userid = Integer.parseInt(request.getParameter("userid"));
```

```
u.deleteUser(userid);
```

```
view = request.getRequestDispatcher("manage-user.jsp");
```

```
}
```

```
view.forward(request, response);
```

```
}
```

Kết quả:

Controller sẽ điều phối quá trình xử lý và gọi View **manage-user.jsp** trình bày dữ liệu kết quả:

```
public class UserDao {
```

```
//...
```

```
public void deleteUser(int userid){
```

```
try{
```

```
Statement st = connection.createStatement();
```

```
String sql = "delete from Users where userid=" + userid;
```

```
st.executeUpdate(sql);
```

```
}catch(SQLException e){
```

```
e.printStackTrace();
```

```
}
```

```
else if(action.equals("insert")){  
    view = request.getRequestDispatcher("add-user.jsp");  
}  
else if(action.equals("edit")){  
    userid = Integer.parseInt(request.getParameter("userid"));  
    User user = u.getUserById(userid);  
    request.setAttribute("userUpdate", user);  
    view = request.getRequestDispatcher("update-user.jsp");  
}  
else if(action.equals("delete")){  
    userid = Integer.parseInt(request.getParameter("userid"));  
    u.deleteUser(userid);  
    view = request.getRequestDispatcher("manage-user.jsp");  
}  
//Gọi chuyển trang  
view.forward(request, response);  
}
```

UserDao.java

```
public class UserDao {  
    //...  
    //Phương thức xóa dữ liệu dựa vào mã số userid của table User  
    public void deleteUser(int userid){  
        try{  
            Statement st = connection.createStatement();  
            String sql = "delete from Users where userid = " + userid;  
            st.executeUpdate(sql);  
        }catch(SQLException e){  
            e.printStackTrace();  
        }  
    }  
    //...  
}
```