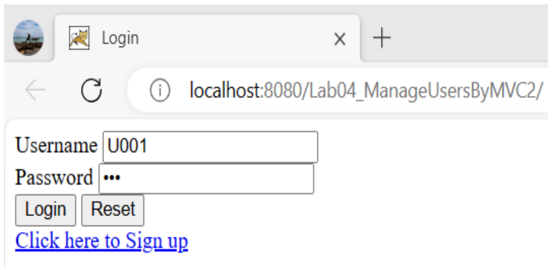
Building a Java Web Application

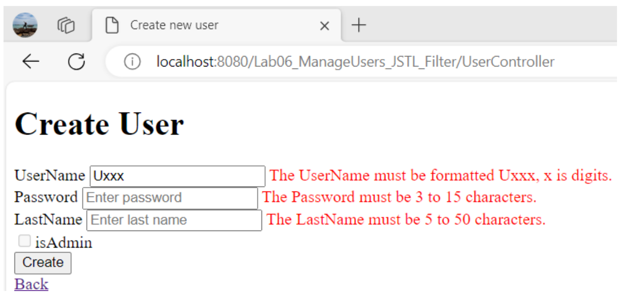
Using MVC2, JSP, JSTL & Filter

Requirements

Create a Java web application (using HTML, Servlets, JSP, EL, JSTL, Filter, and JDBC) by MVC2 Model to perform functions as manage users.

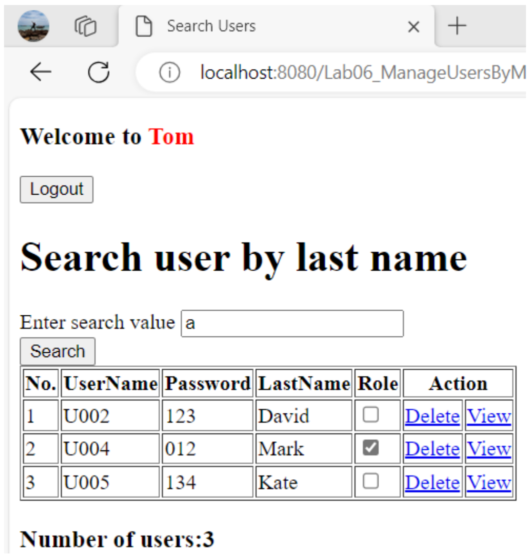


Username U001
Password •••
Login Reset
Click here to Sign up



Create User

UserName Uxxx The UserName must be formatted Uxxx, x is digits.
Password Enter password The Password must be 3 to 15 characters.
LastName Enter last name The LastName must be 5 to 50 characters.
☐ isAdmin
Create
Back



Welcome to Tom

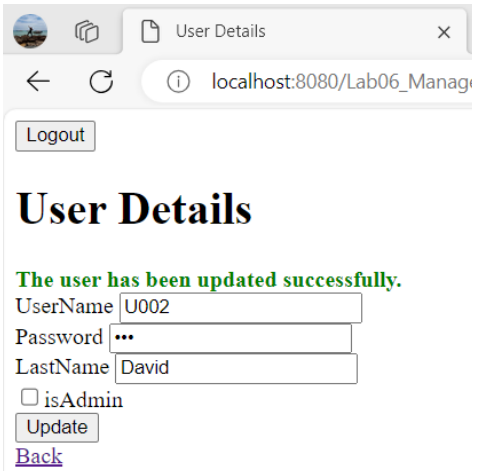
Logout

Search user by last name

Enter search value a
Search

No.	UserName	Password	LastName	Role	Action	
1	U002	123	David	☐	Delete	View
2	U004	012	Mark	☑	Delete	View
3	U005	134	Kate	☐	Delete	View

Number of users:3



Logout

User Details

The user has been updated successfully.
UserName U002
Password •••
LastName David
☐ isAdmin
Update
Back

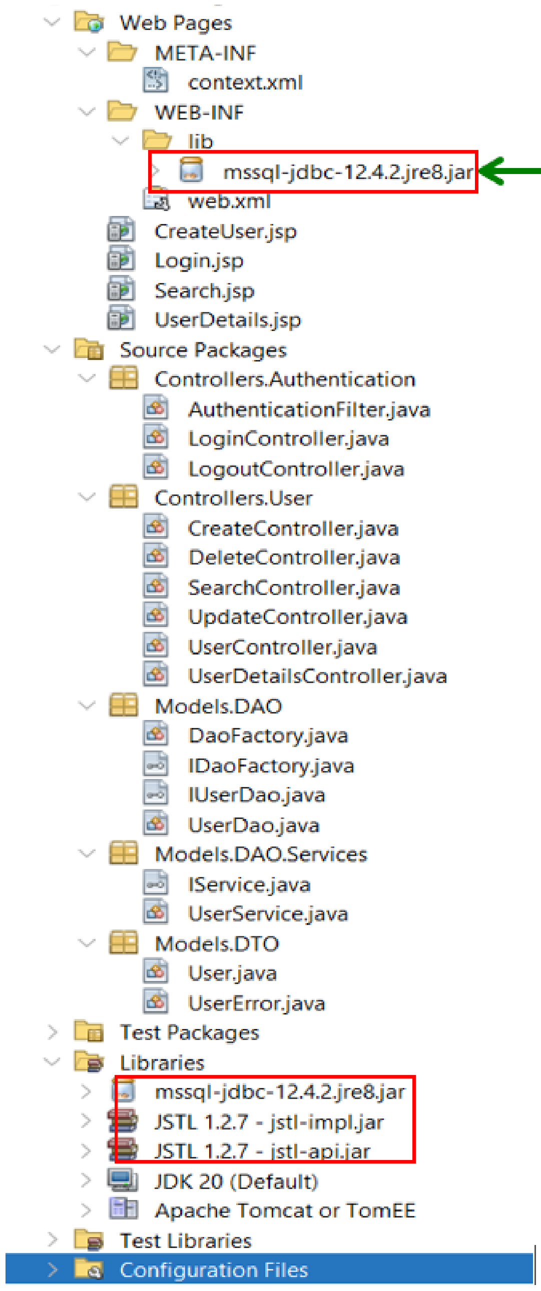
Lab Objectives

In this lab, you will be:

- Create Java web project by MVC2 Model
- Create the web pages by HTML, and JSP (JSTL & EL)
- Apply passing data from HTML to Servlet by GET and POST method
- Apply Filter and Session for the Authentication
- Apply Error handling in the web app

- Apply design pattern: Repository and Factory pattern
- Apply RequestDispatcher to pass data between Servlets
- Performs CRUD functions by SqlJDBC driver and SQL Server
- Create a dynamic connection to the SQL Server
- Run the project and test the functions of the project

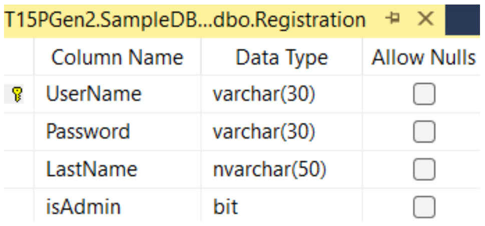
Step 01. Create a Java Web App named **Lab07ManageUsers_JSTL_Filter** is structured as follows:



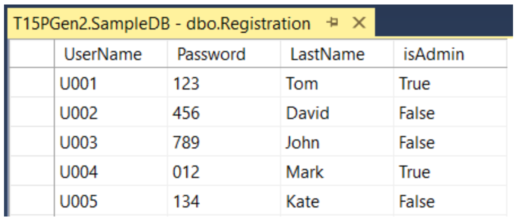
```
Web Pages
  META-INF
    context.xml
  WEB-INF
    lib
      mssql-jdbc-12.4.2.jre8.jar
    web.xml
  CreateUser.jsp
  Login.jsp
  Search.jsp
  UserDetails.jsp
Source Packages
  Controllers.Authentication
    AuthenticationFilter.java
    LoginController.java
    LogoutController.java
  Controllers.User
    CreateController.java
    DeleteController.java
    SearchController.java
    UpdateController.java
    UserController.java
    UserDetailsController.java
  Models.DAO
    DaoFactory.java
    IDaoFactory.java
    IUserDao.java
    UserDao.java
  Models.DAO.Services
    IService.java
    UserService.java
  Models.DTO
    User.java
    UserError.java
Test Packages
Libraries
  mssql-jdbc-12.4.2.jre8.jar
  JSTL 1.2.7 - jstl-impl.jar
  JSTL 1.2.7 - jstl-api.jar
  JDK 20 (Default)
  Apache Tomcat or TomEE
Test Libraries
Configuration Files
```

- On the project, right-click on Libraries | Add JAR/Folder then select **sqljdbc(sqljdbc42.jar)** driver that matches the SQL Server version on the machine.

-Open SQL Server Management Studio and create a database named **SampleDB** has a table named **Registration** as follows:

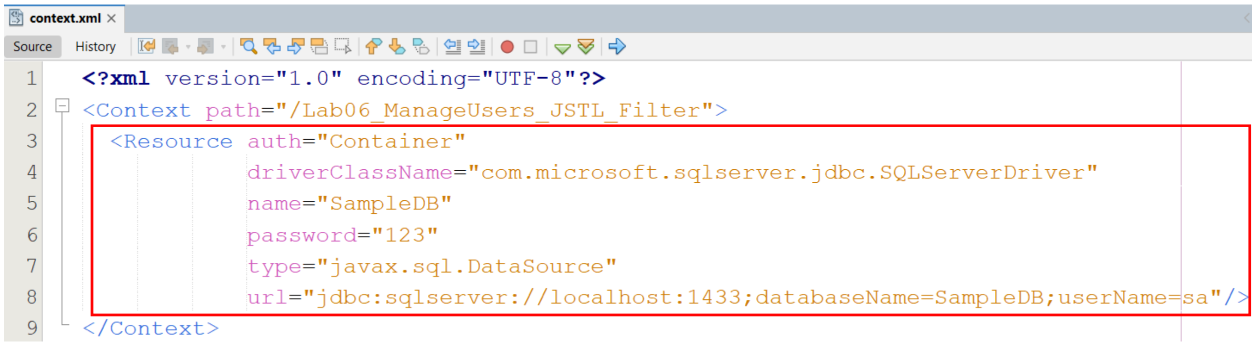

T15PGen2.SampleDB...dbo.Registration

Column Name	Data Type	Allow Nulls
UserName	varchar(30)	☐
Password	varchar(30)	☐
LastName	nvarchar(50)	☐
isAdmin	bit	☐


T15PGen2.SampleDB - dbo.Registration

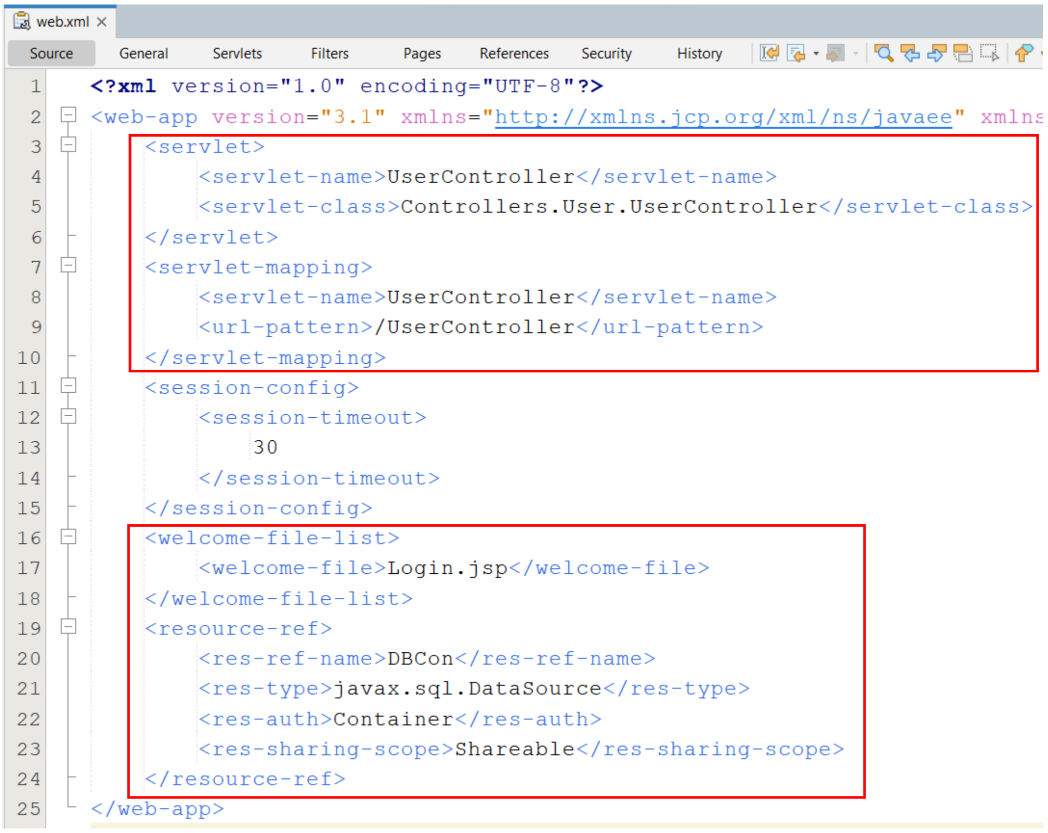
UserName	Password	LastName	isAdmin
U001	123	Tom	True
U002	456	David	False
U003	789	John	False
U004	012	Mark	True
U005	134	Kate	False

-Open the **Context.xml** in the META-INF folder and config for the connection string as follows:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/Lab06_ManageUsers_JSTL_Filter">
    <Resource auth="Container"
              driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
              name="SampleDB"
              password="123"
              type="javax.sql.DataSource"
              url="jdbc:sqlserver://localhost:1433;databaseName=SampleDB;userName=sa"/>
</Context>
```

-Open the **web.xml** in the WEB-INF folder then update as follows:

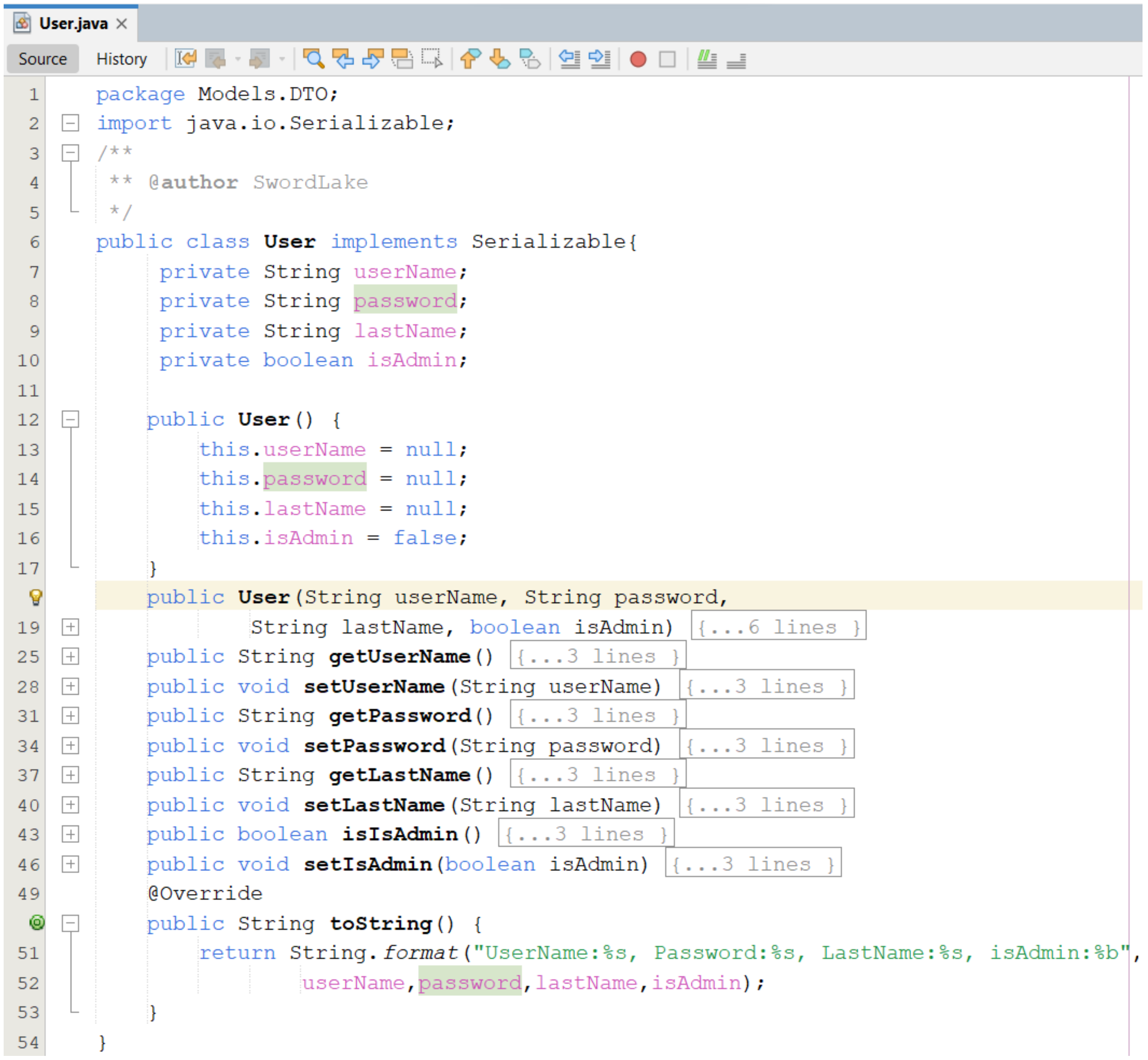


```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns
    <servlet>
        <servlet-name>UserController</servlet-name>
        <servlet-class>Controllers.User.UserController</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>UserController</servlet-name>
        <url-pattern>/UserController</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>Login.jsp</welcome-file>
    </welcome-file-list>
    <resource-ref>
        <res-ref-name>DBCon</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
        <res-sharing-scope>Shareable</res-sharing-scope>
    </resource-ref>
</web-app>
```

Step 02. Write codes for the classes in the **Models.DTO, Models.DAO** and **Models.DAO.Services** packages are as follows:

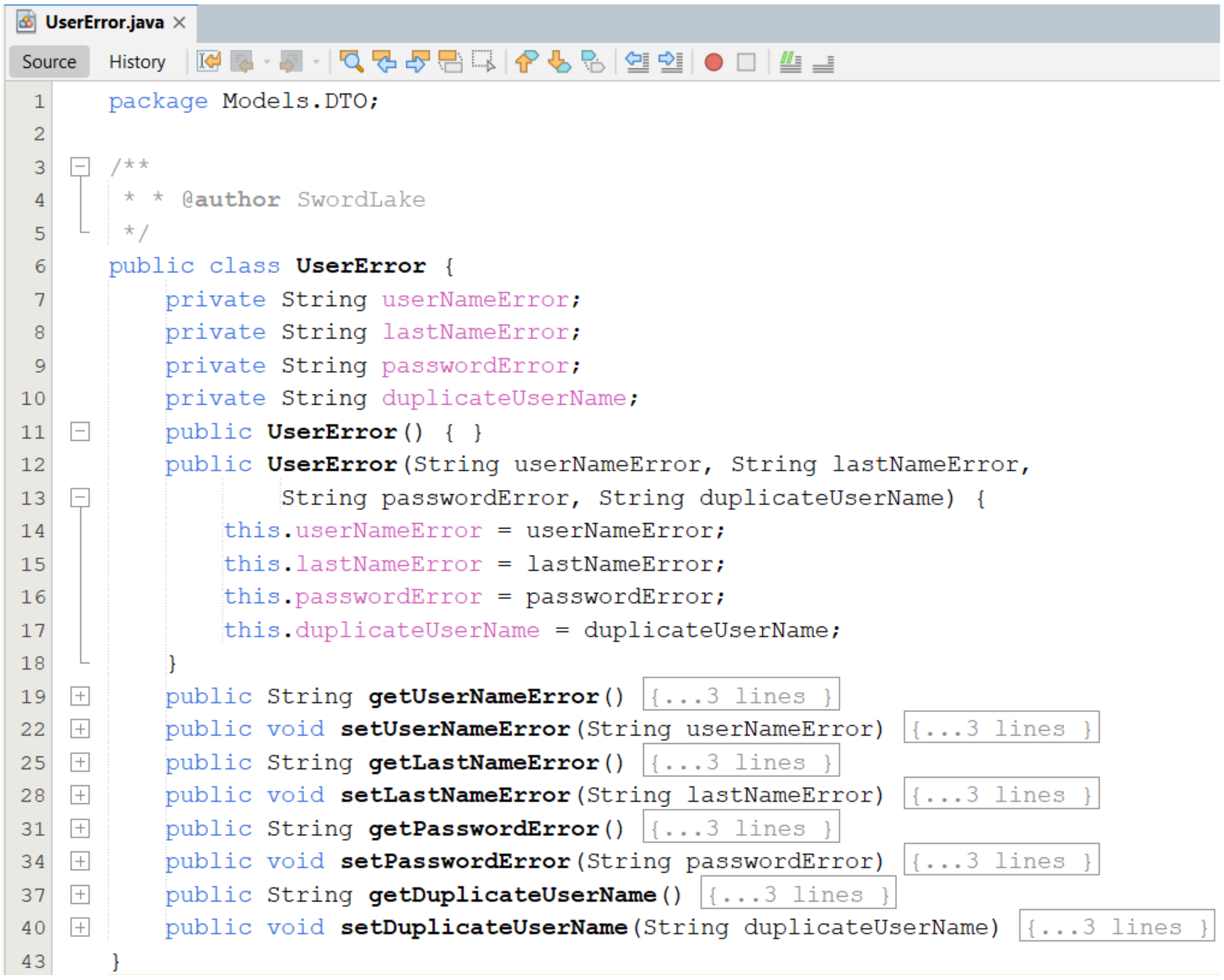
2.1. Models.DTO package

- **User.java**



```
package Models.DTO;
import java.io.Serializable;
/**
 ** @author SwordLake
 */
public class User implements Serializable{
     private String userName;
     private String password;
     private String lastName;
     private boolean isAdmin;

    public User() {
        this.userName = null;
        this.password = null;
        this.lastName = null;
        this.isAdmin = false;
    }
    public User(String userName, String password,
            String lastName, boolean isAdmin) {...6 lines }
    public String getUserName() {...3 lines }
    public void setUserName(String userName) {...3 lines }
    public String getPassword() {...3 lines }
    public void setPassword(String password) {...3 lines }
    public String getLastName() {...3 lines }
    public void setLastName(String lastName) {...3 lines }
    public boolean isIsAdmin() {...3 lines }
    public void setIsAdmin(boolean isAdmin) {...3 lines }
    @Override
    public String toString() {
        return String.format("UserName:%s, Password:%s, LastName:%s, isAdmin:%b",
                userName,password,lastName,isAdmin);
    }
}
```

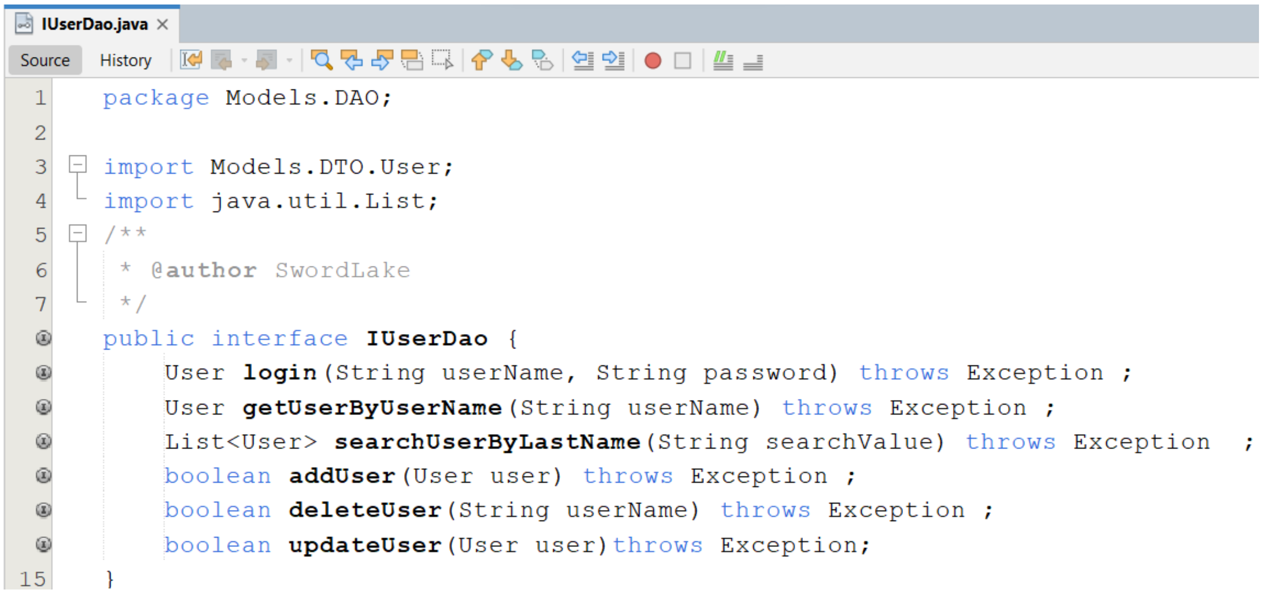
- **UserError.java**


UserError.java

```
package Models.DTO;

/**
 * * @author SwordLake
 */
public class UserError {
    private String userNameError;
    private String lastNameError;
    private String passwordError;
    private String duplicateUserName;
    public UserError() { }
    public UserError(String userNameError, String lastNameError,
            String passwordError, String duplicateUserName) {
        this.userNameError = userNameError;
        this.lastNameError = lastNameError;
        this.passwordError = passwordError;
        this.duplicateUserName = duplicateUserName;
    }
    public String getUserNameError() {...3 lines }
    public void setUserNameError(String userNameError) {...3 lines }
    public String getLastNameError() {...3 lines }
    public void setLastNameError(String lastNameError) {...3 lines }
    public String getPasswordError() {...3 lines }
    public void setPasswordError(String passwordError) {...3 lines }
    public String getDuplicateUserName() {...3 lines }
    public void setDuplicateUserName(String duplicateUserName) {...3 lines }
}
```

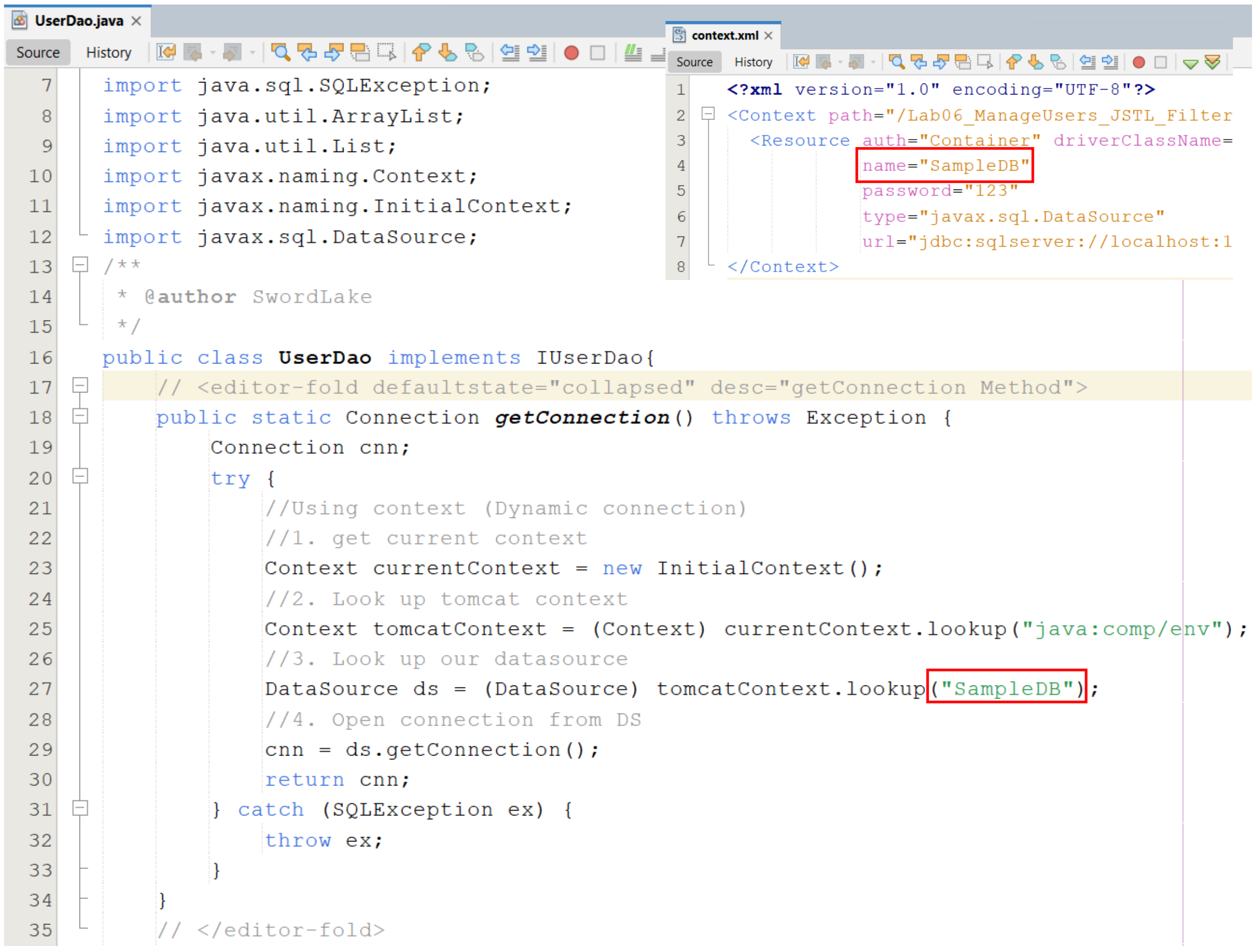
2.2. **Models.DAO** package

- **IUserDao.java**


IUserDao.java

```
package Models.DAO;

import Models.DTO.User;
import java.util.List;
/**
 * @author SwordLake
 */
public interface IUserDao {
    User login(String userName, String password) throws Exception ;
    User getUserByUserName(String userName) throws Exception ;
    List<User> searchUserByLastName(String searchValue) throws Exception  ;
    boolean addUser(User user) throws Exception ;
    boolean deleteUser(String userName) throws Exception ;
    boolean updateUser(User user)throws Exception;
}
```

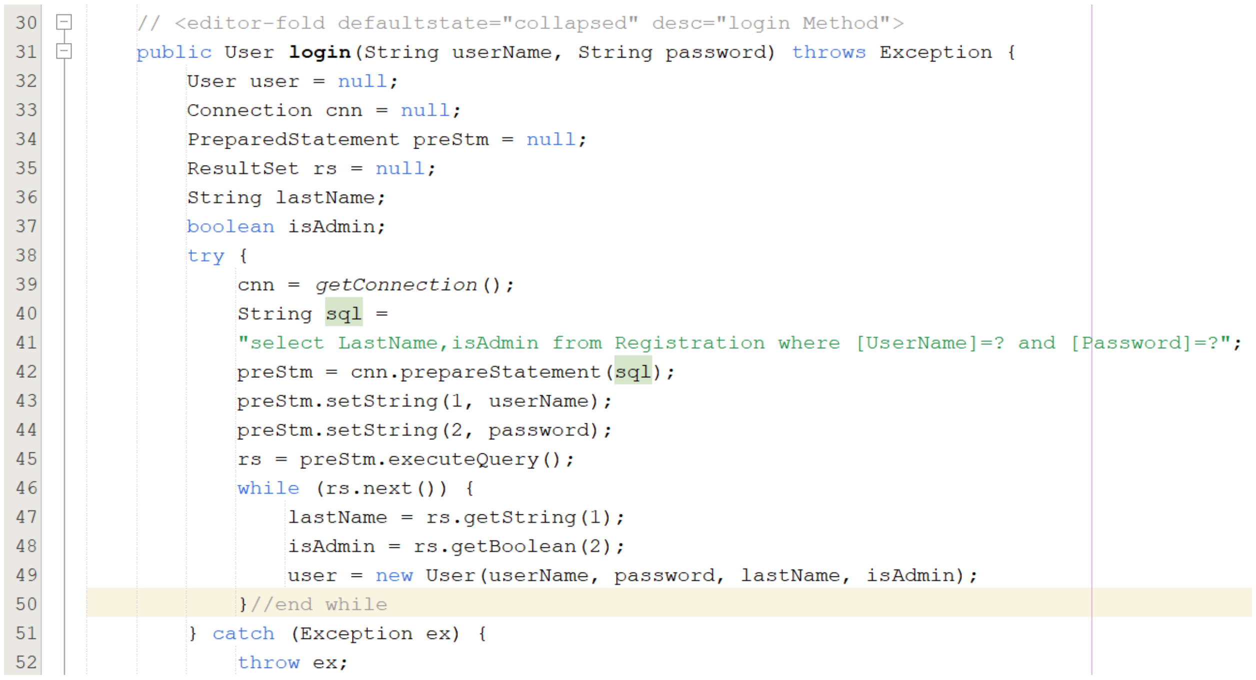
- **UserDao.java**


UserDao.java × Source History

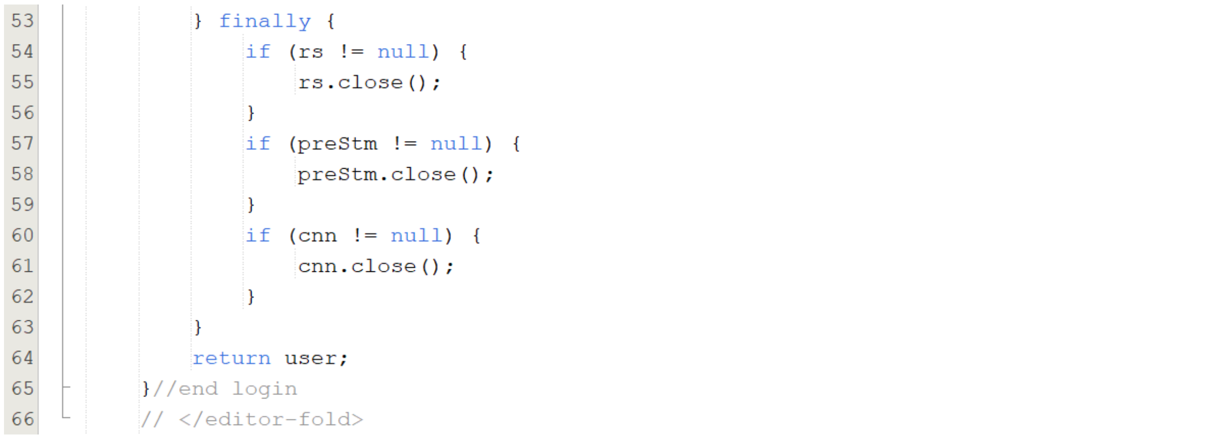
```java
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
/**
 * @author SwordLake
 */
public class UserDao implements IUserDao{
    // <editor-fold defaultstate="collapsed" desc="getConnection Method">
    public static Connection getConnection() throws Exception {
        Connection cnn;
        try {
            //Using context (Dynamic connection)
            //1. get current context
            Context currentContext = new InitialContext();
            //2. Look up tomcat context
            Context tomcatContext = (Context) currentContext.lookup("java:comp/env");
            //3. Look up our datasource
            DataSource ds = (DataSource) tomcatContext.lookup("SampleDB");
            //4. Open connection from DS
            cnn = ds.getConnection();
            return cnn;
        } catch (SQLException ex) {
            throw ex;
        }
    }
    // </editor-fold>
```

context.xml × Source History

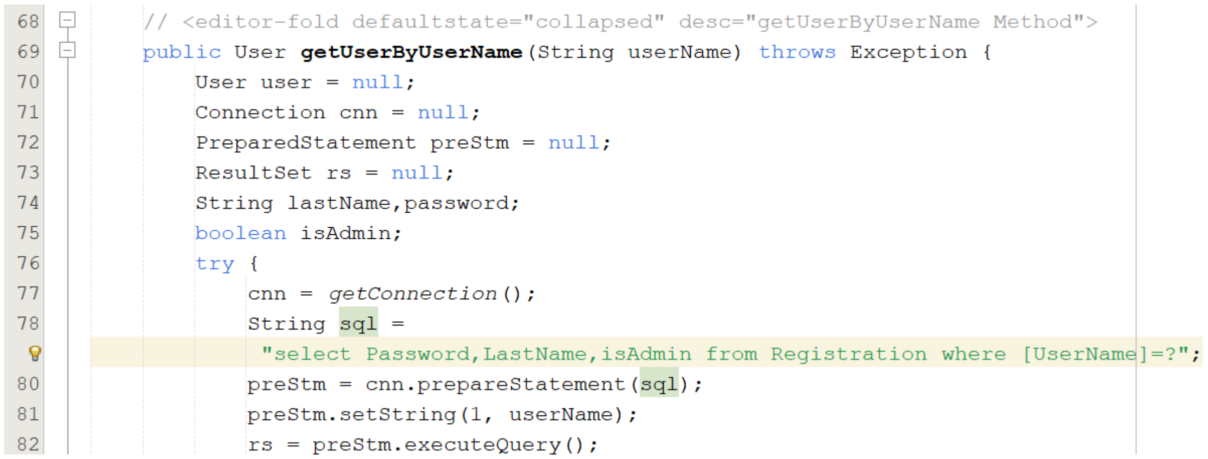
```xml
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/Lab06_ManageUsers_JSTL_Filter
  <Resource auth="Container" driverClassName=
            name="SampleDB"
            password="123"
            type="javax.sql.DataSource"
            url="jdbc:sqlserver://localhost:1
</Context>
```



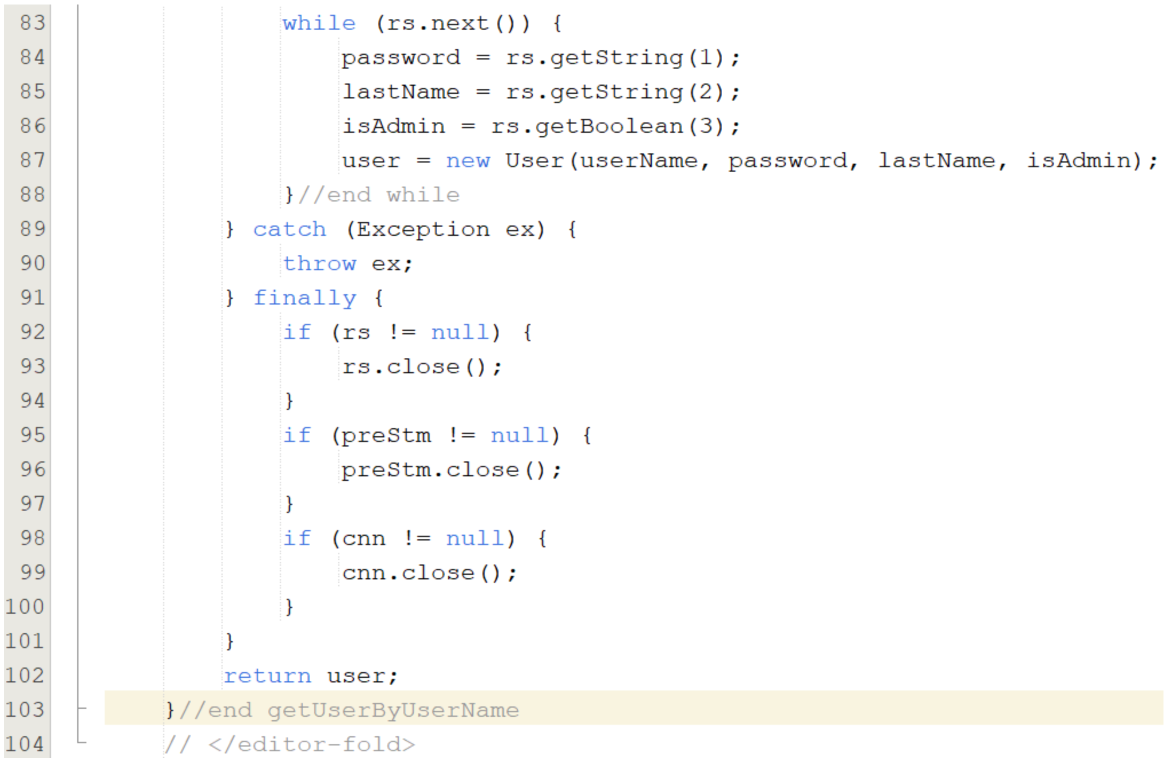
```java
    // <editor-fold defaultstate="collapsed" desc="login Method">
    public User login(String userName, String password) throws Exception {
        User user = null;
        Connection cnn = null;
        PreparedStatement preStm = null;
        ResultSet rs = null;
        String lastName;
        boolean isAdmin;
        try {
            cnn = getConnection();
            String sql =
            "select LastName,isAdmin from Registration where [UserName]=? and [Password]=?";
            preStm = cnn.prepareStatement(sql);
            preStm.setString(1, userName);
            preStm.setString(2, password);
            rs = preStm.executeQuery();
            while (rs.next()) {
                lastName = rs.getString(1);
                isAdmin = rs.getBoolean(2);
                user = new User(userName, password, lastName, isAdmin);
            }//end while
        } catch (Exception ex) {
            throw ex;
```



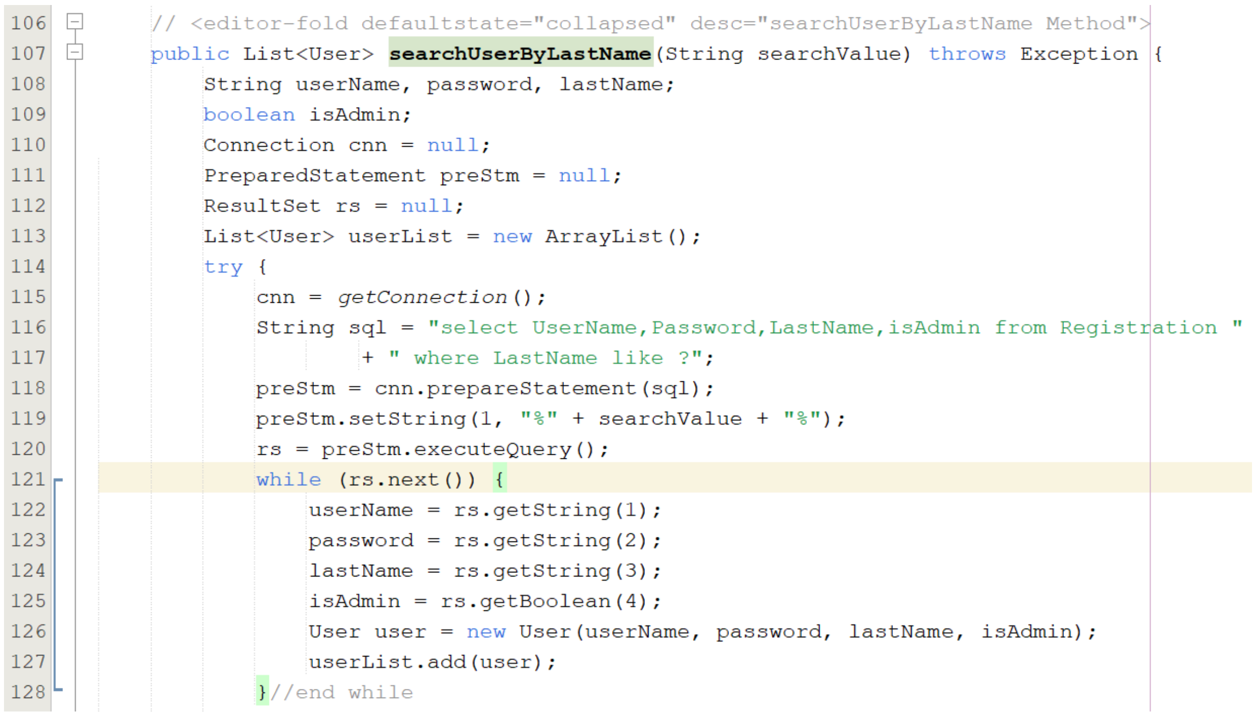
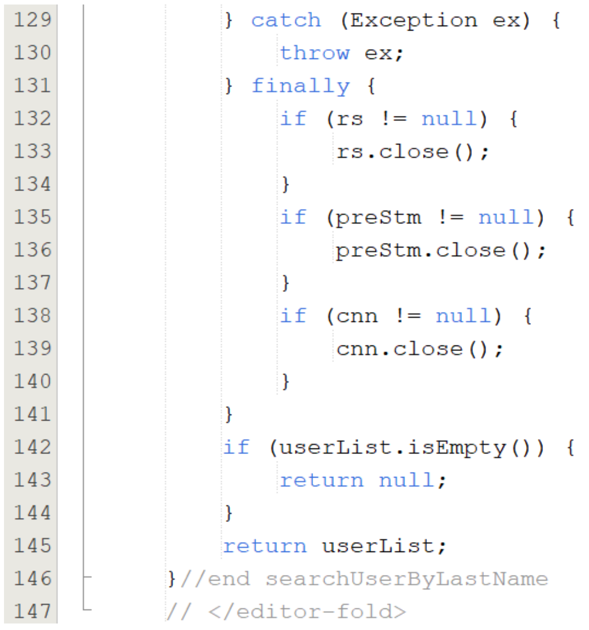
```java
        } finally {
            if (rs != null) {
                rs.close();
            }
            if (preStm != null) {
                preStm.close();
            }
            if (cnn != null) {
                cnn.close();
            }
        }
        return user;
    }//end login
    // </editor-fold>
```



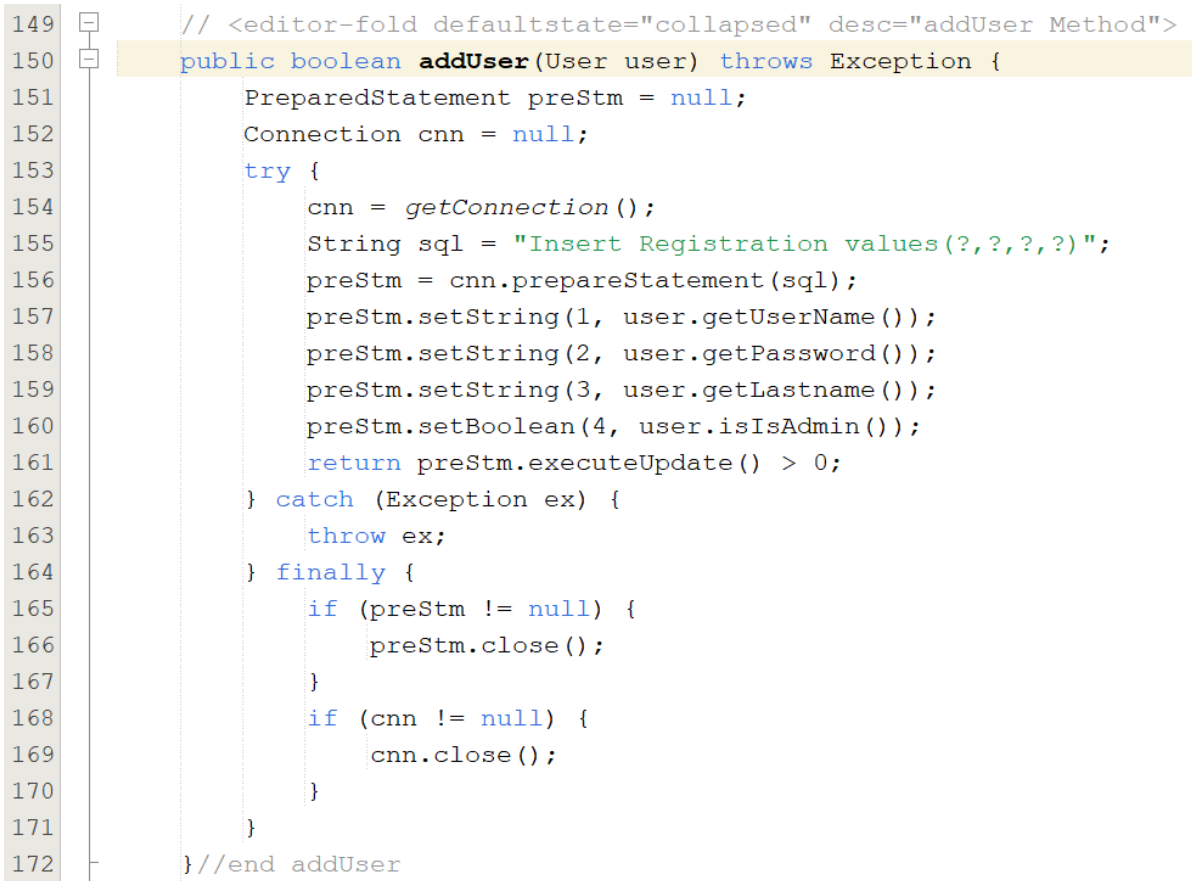
```java
    // <editor-fold defaultstate="collapsed" desc="getUserByUserName Method">
    public User getUserByUserName(String userName) throws Exception {
        User user = null;
        Connection cnn = null;
        PreparedStatement preStm = null;
        ResultSet rs = null;
        String lastName,password;
        boolean isAdmin;
        try {
            cnn = getConnection();
            String sql =
             "select Password,LastName,isAdmin from Registration where [UserName]=?";
            preStm = cnn.prepareStatement(sql);
            preStm.setString(1, userName);
            rs = preStm.executeQuery();
```



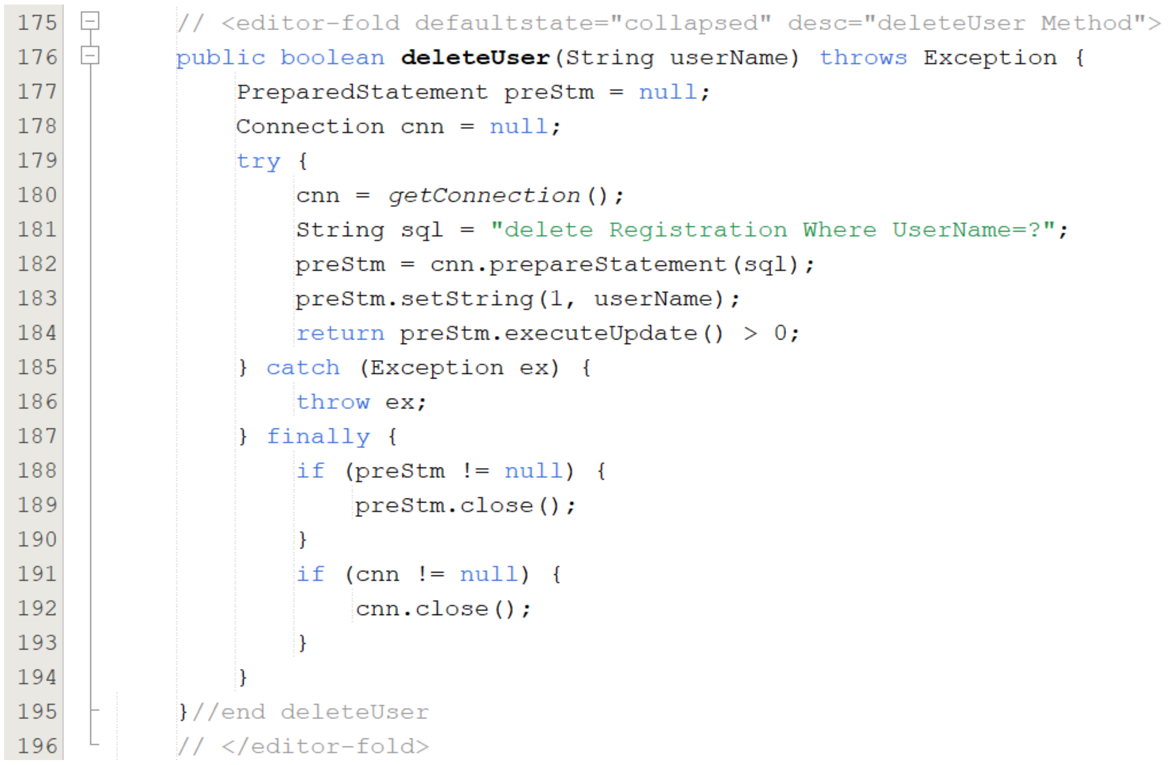
```java
            while (rs.next()) {
                password = rs.getString(1);
                lastName = rs.getString(2);
                isAdmin = rs.getBoolean(3);
                user = new User(userName, password, lastName, isAdmin);
            }//end while
        } catch (Exception ex) {
            throw ex;
        } finally {
            if (rs != null) {
                rs.close();
            }
            if (preStm != null) {
                preStm.close();
            }
            if (cnn != null) {
                cnn.close();
            }
        }
        return user;
    }//end getUserByUserName
    // </editor-fold>
```

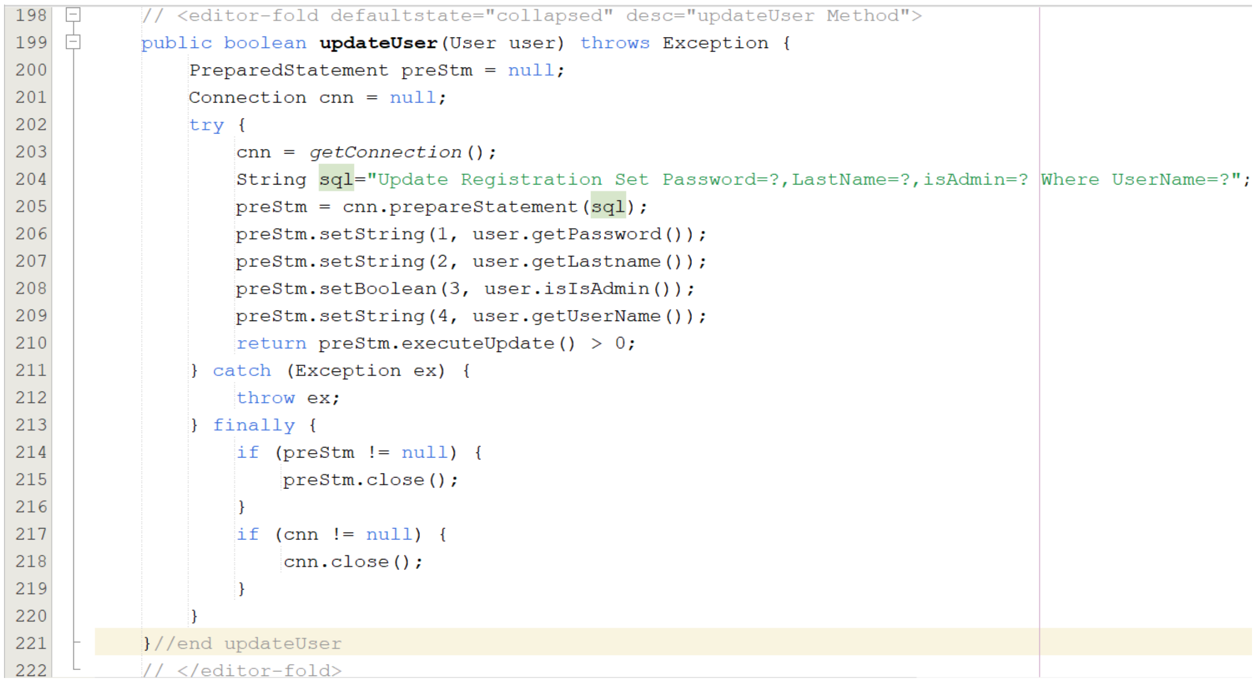
```java
    // <editor-fold defaultstate="collapsed" desc="searchUserByLastName Method">
    public List<User> searchUserByLastName(String searchValue) throws Exception {
        String userName, password, lastName;
        boolean isAdmin;
        Connection cnn = null;
        PreparedStatement preStm = null;
        ResultSet rs = null;
        List<User> userList = new ArrayList();
        try {
            cnn = getConnection();
            String sql = "select UserName,Password,LastName,isAdmin from Registration "
                    + " where LastName like ?";
            preStm = cnn.prepareStatement(sql);
            preStm.setString(1, "%" + searchValue + "%");
            rs = preStm.executeQuery();
            while (rs.next()) {
                userName = rs.getString(1);
                password = rs.getString(2);
                lastName = rs.getString(3);
                isAdmin = rs.getBoolean(4);
                User user = new User(userName, password, lastName, isAdmin);
                userList.add(user);
            }//end while
        } catch (Exception ex) {
            throw ex;
        } finally {
            if (rs != null) {
                rs.close();
            }
            if (preStm != null) {
                preStm.close();
            }
            if (cnn != null) {
                cnn.close();
            }
        }
        if (userList.isEmpty()) {
            return null;
        }
        return userList;
    }//end searchUserByLastName
    // </editor-fold>
```



```java
    // <editor-fold defaultstate="collapsed" desc="addUser Method">
    public boolean addUser(User user) throws Exception {
        PreparedStatement preStm = null;
        Connection cnn = null;
        try {
            cnn = getConnection();
            String sql = "Insert Registration values(?,?,?,?)";
            preStm = cnn.prepareStatement(sql);
            preStm.setString(1, user.getUserName());
            preStm.setString(2, user.getPassword());
            preStm.setString(3, user.getLastname());
            preStm.setBoolean(4, user.isIsAdmin());
            return preStm.executeUpdate() > 0;
        } catch (Exception ex) {
            throw ex;
        } finally {
            if (preStm != null) {
                preStm.close();
            }
            if (cnn != null) {
                cnn.close();
            }
        }
    }//end addUser
```

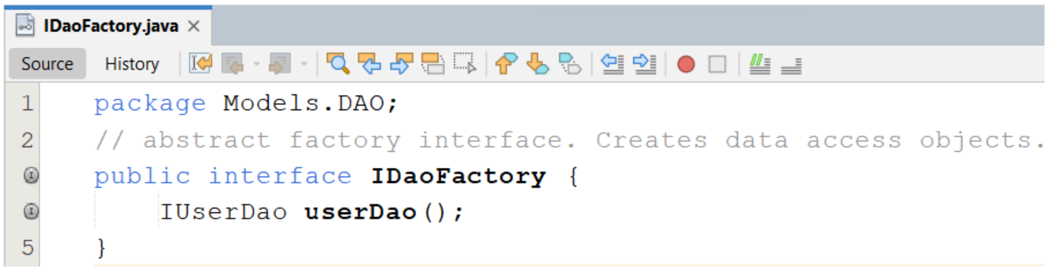


```java
    // <editor-fold defaultstate="collapsed" desc="deleteUser Method">
    public boolean deleteUser(String userName) throws Exception {
        PreparedStatement preStm = null;
        Connection cnn = null;
        try {
            cnn = getConnection();
            String sql = "delete Registration Where UserName=?";
            preStm = cnn.prepareStatement(sql);
            preStm.setString(1, userName);
            return preStm.executeUpdate() > 0;
        } catch (Exception ex) {
            throw ex;
        } finally {
            if (preStm != null) {
                preStm.close();
            }
            if (cnn != null) {
                cnn.close();
            }
        }
    }//end deleteUser
    // </editor-fold>
```



```java
    // <editor-fold defaultstate="collapsed" desc="updateUser Method">
    public boolean updateUser(User user) throws Exception {
        PreparedStatement preStm = null;
        Connection cnn = null;
        try {
            cnn = getConnection();
            String sql="Update Registration Set Password=?,LastName=?,isAdmin=? Where UserName=?";
            preStm = cnn.prepareStatement(sql);
            preStm.setString(1, user.getPassword());
            preStm.setString(2, user.getLastname());
            preStm.setBoolean(3, user.isIsAdmin());
            preStm.setString(4, user.getUserName());
            return preStm.executeUpdate() > 0;
        } catch (Exception ex) {
            throw ex;
        } finally {
            if (preStm != null) {
                preStm.close();
            }
            if (cnn != null) {
                cnn.close();
            }
        }
    }//end updateUser
    // </editor-fold>
```

- **IDaoFactory.java**

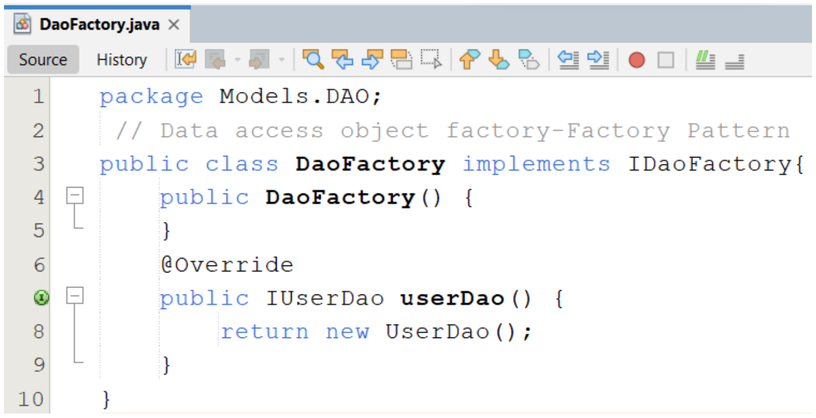


IDaoFactory.java

Source History

```java
package Models.DAO;
// abstract factory interface. Creates data access objects.
public interface IDaoFactory {
    IUserDao userDao();
}
```

- **DaoFactory.java**



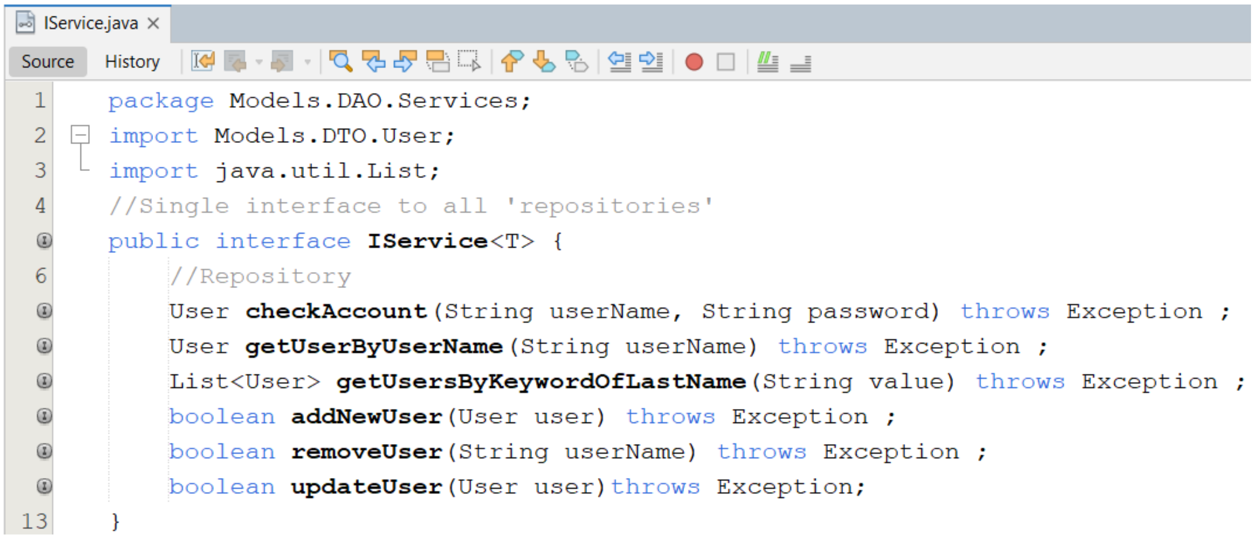
DaoFactory.java

Source History

```java
package Models.DAO;
 // Data access object factory-Factory Pattern
public class DaoFactory implements IDaoFactory{
    public DaoFactory() {
    }
    @Override
    public IUserDao userDao() {
        return new UserDao();
    }
}
```

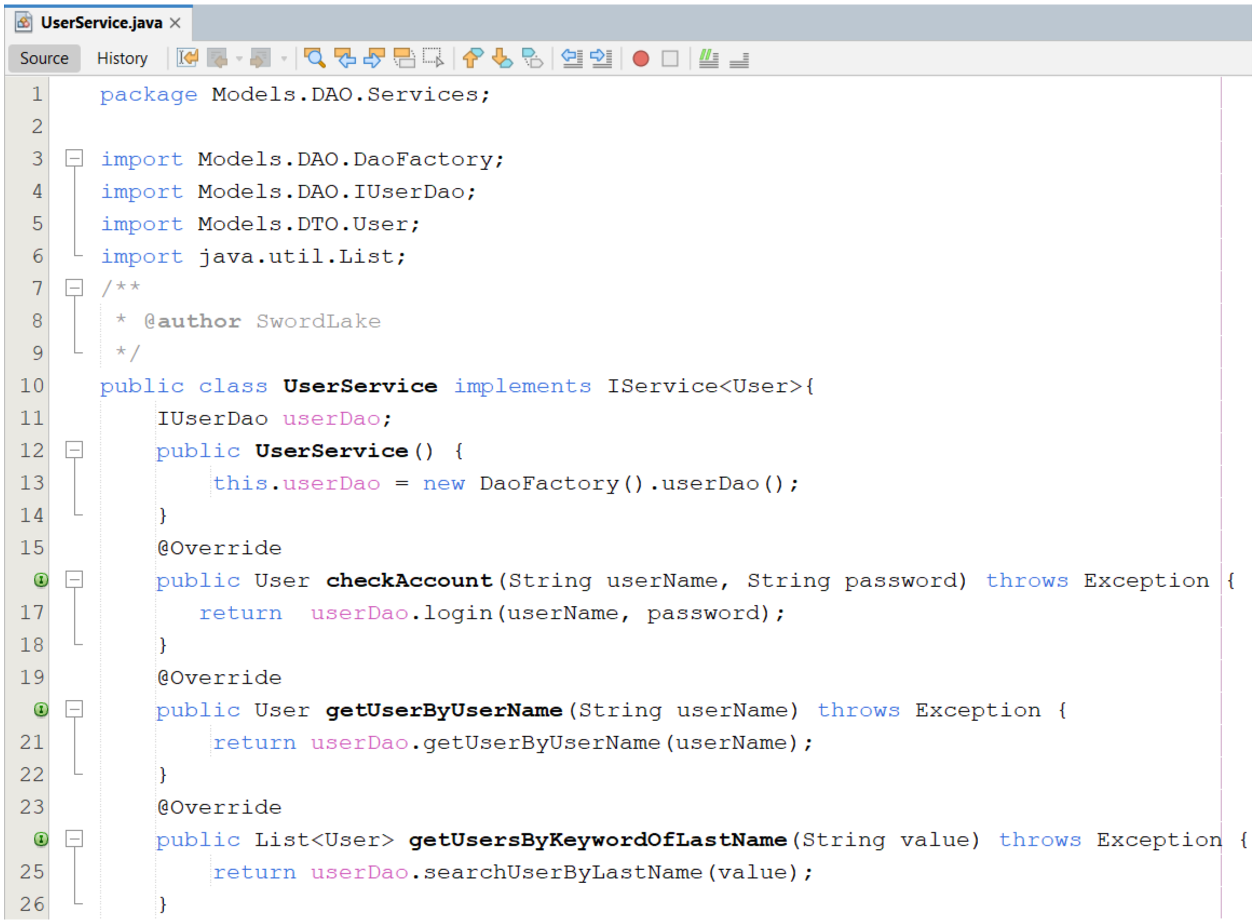
2.3. Models. DAO.Services package

- **IService.java**

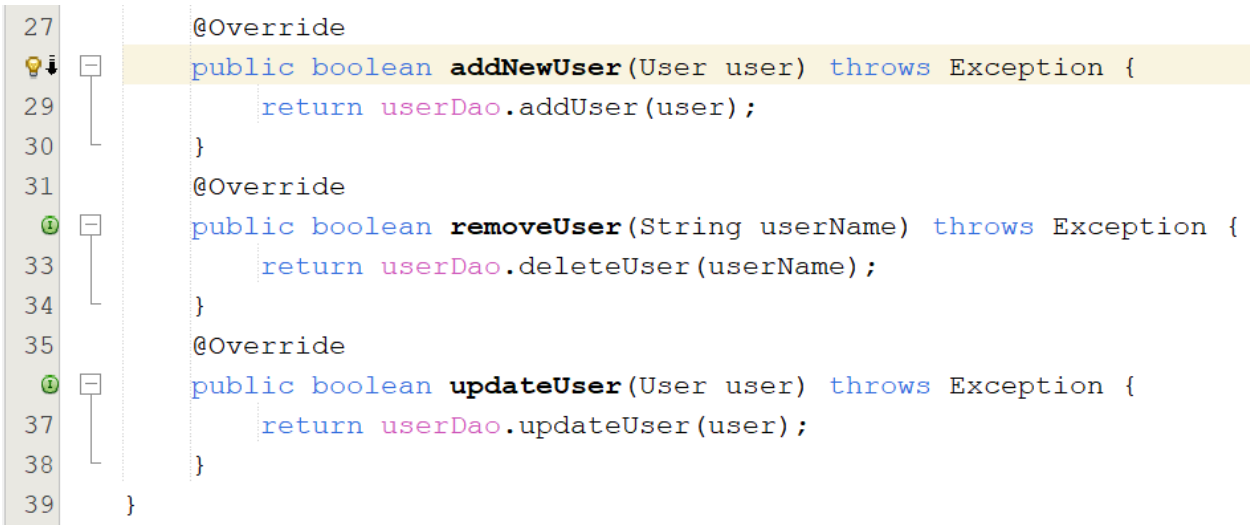


```
package Models.DAO.Services;
import Models.DTO.User;
import java.util.List;
//Single interface to all 'repositories'
public interface IService<T> {
    //Repository
    User checkAccount(String userName, String password) throws Exception ;
    User getUserByUserName(String userName) throws Exception ;
    List<User> getUsersByKeywordOfLastName(String value) throws Exception ;
    boolean addNewUser(User user) throws Exception ;
    boolean removeUser(String userName) throws Exception ;
    boolean updateUser(User user)throws Exception;
}
```

- **UserService.java**



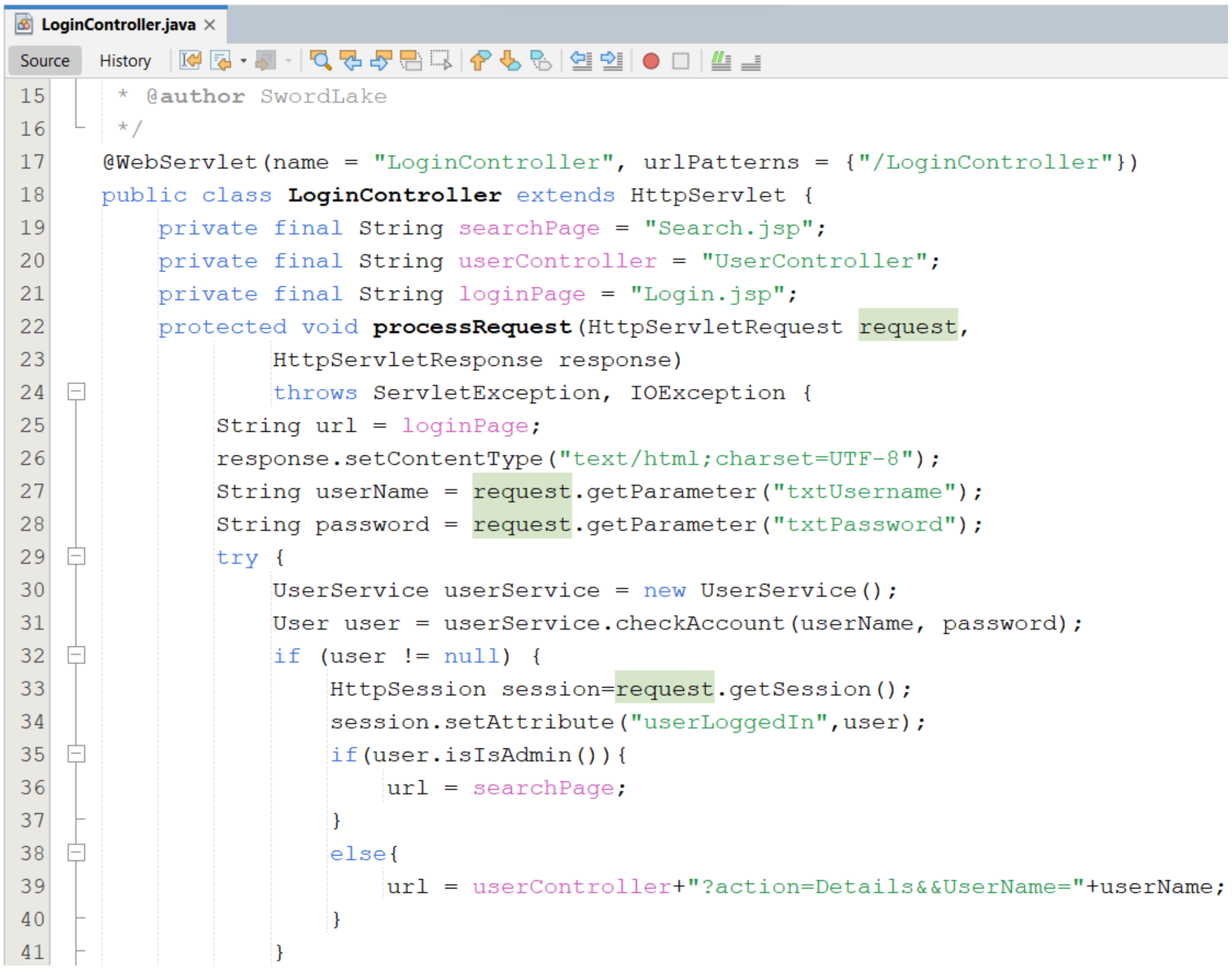
```
package Models.DAO.Services;

import Models.DAO.DaoFactory;
import Models.DAO.IUserDao;
import Models.DTO.User;
import java.util.List;
/**
 * @author SwordLake
 */
public class UserService implements IService<User>{
    IUserDao userDao;
    public UserService()  {
        this.userDao = new DaoFactory().userDao();
    }
    @Override
    public User checkAccount(String userName, String password) throws Exception {
       return  userDao.login(userName, password);
    }
    @Override
    public User getUserByUserName(String userName) throws Exception {
        return userDao.getUserByUserName(userName);
    }
    @Override
    public List<User> getUsersByKeywordOfLastName(String value) throws Exception {
        return userDao.searchUserByLastName(value);
    }
```



```
    @Override
    public boolean addNewUser(User user) throws Exception {
        return userDao.addUser(user);
    }
    @Override
    public boolean removeUser(String userName) throws Exception {
        return userDao.deleteUser(userName);
    }
    @Override
    public boolean updateUser(User user) throws Exception {
        return userDao.updateUser(user);
    }
}
```

<u>Step 03</u>. Write codes for the classes in the **Controllers.Authentication** package is as follows:

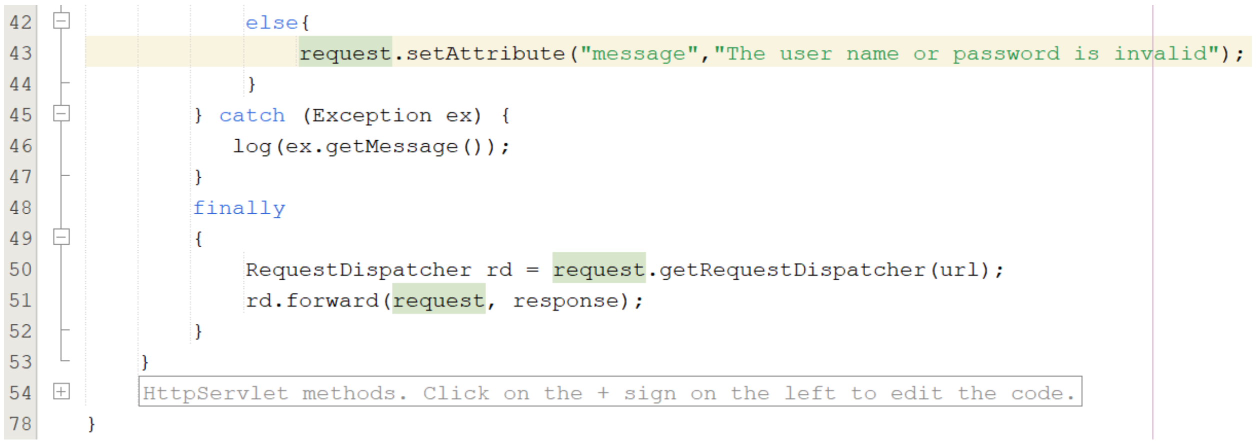
- **LoginController.java**



LoginController.java

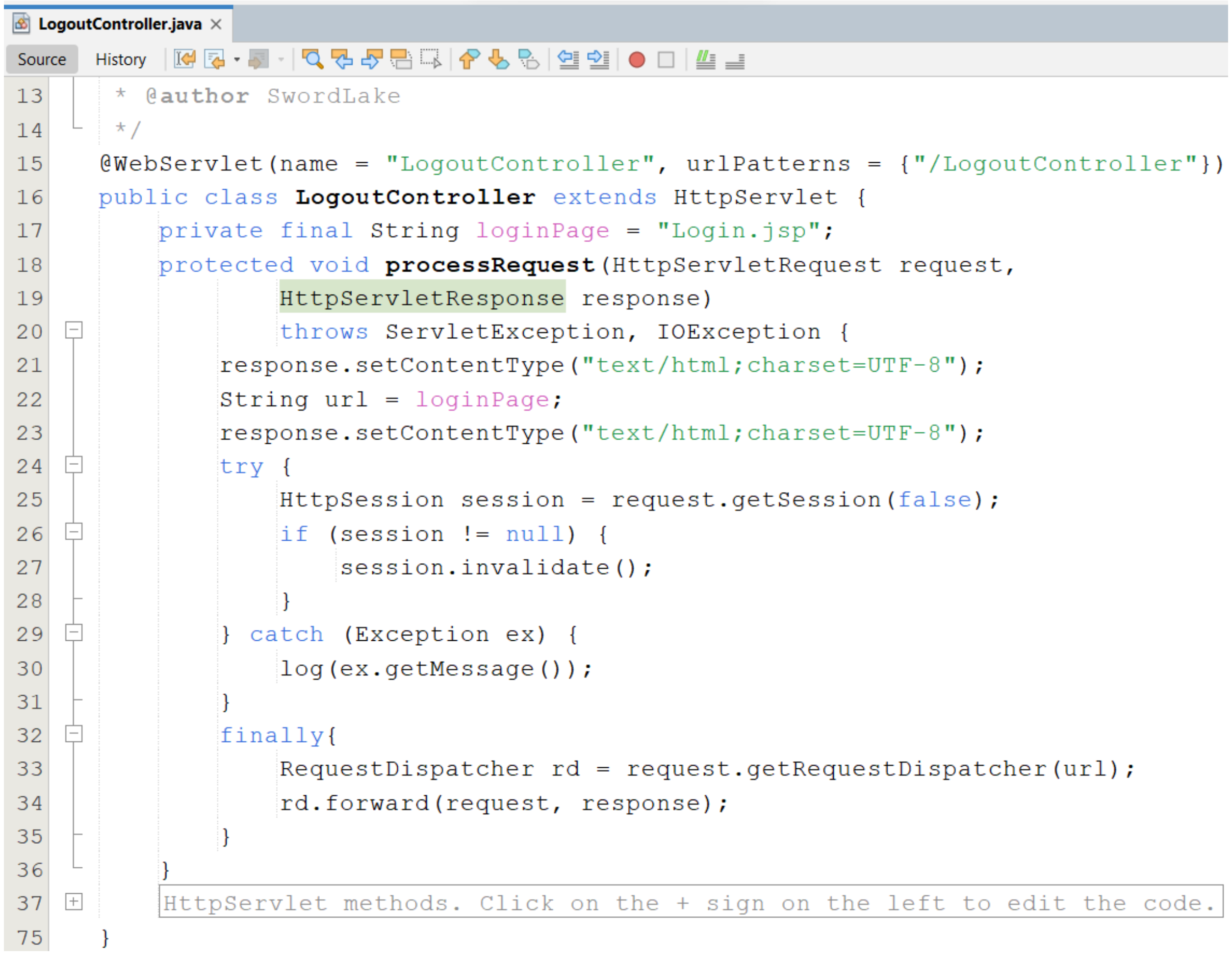
Source History

```
 * @author SwordLake
 */
@WebServlet(name = "LoginController", urlPatterns = {"/LoginController"})
public class LoginController extends HttpServlet {
    private final String searchPage = "Search.jsp";
    private final String userController = "UserController";
    private final String loginPage = "Login.jsp";
    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        String url = loginPage;
        response.setContentType("text/html;charset=UTF-8");
        String userName = request.getParameter("txtUsername");
        String password = request.getParameter("txtPassword");
        try {
            UserService userService = new UserService();
            User user = userService.checkAccount(userName, password);
            if (user != null) {
                HttpSession session=request.getSession();
                session.setAttribute("userLoggedIn",user);
                if(user.isIsAdmin()){
                    url = searchPage;
                }
                else{
                    url = userController+"?action=Details&&UserName="+userName;
                }
            }
```



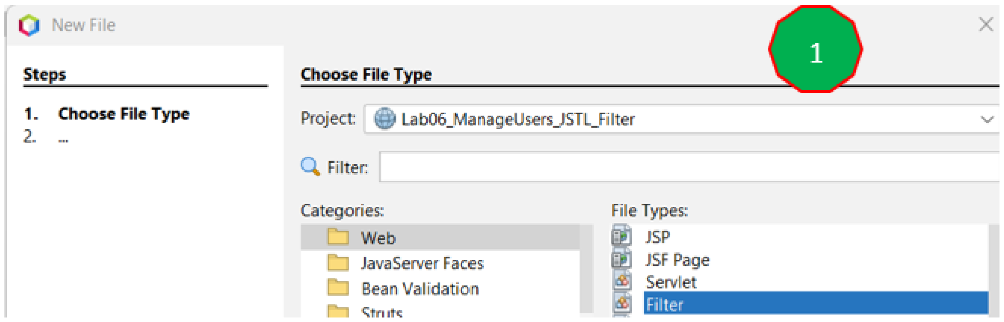
```
                else{
                    request.setAttribute("message","The user name or password is invalid");
                }
            } catch (Exception ex) {
                log(ex.getMessage());
            }
            finally
            {
                RequestDispatcher rd = request.getRequestDispatcher(url);
                rd.forward(request, response);
            }
        }
        HttpServlet methods. Click on the + sign on the left to edit the code.
    }
```

- **LogoutController.java**



```
 * @author SwordLake
 */
@WebServlet(name = "LogoutController", urlPatterns = {"/LogoutController"})
public class LogoutController extends HttpServlet {
    private final String loginPage = "Login.jsp";
    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String url = loginPage;
        response.setContentType("text/html;charset=UTF-8");
        try {
            HttpSession session = request.getSession(false);
            if (session != null) {
                session.invalidate();
            }
        } catch (Exception ex) {
            log(ex.getMessage());
        }
        finally{
            RequestDispatcher rd = request.getRequestDispatcher(url);
            rd.forward(request, response);
        }
    }
    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

-Right-click on the Controllers.Authentication package then add a filter controller named **AuthenticationFilter.java** as follows:



New File

Steps

1. Choose File Type
2. ...

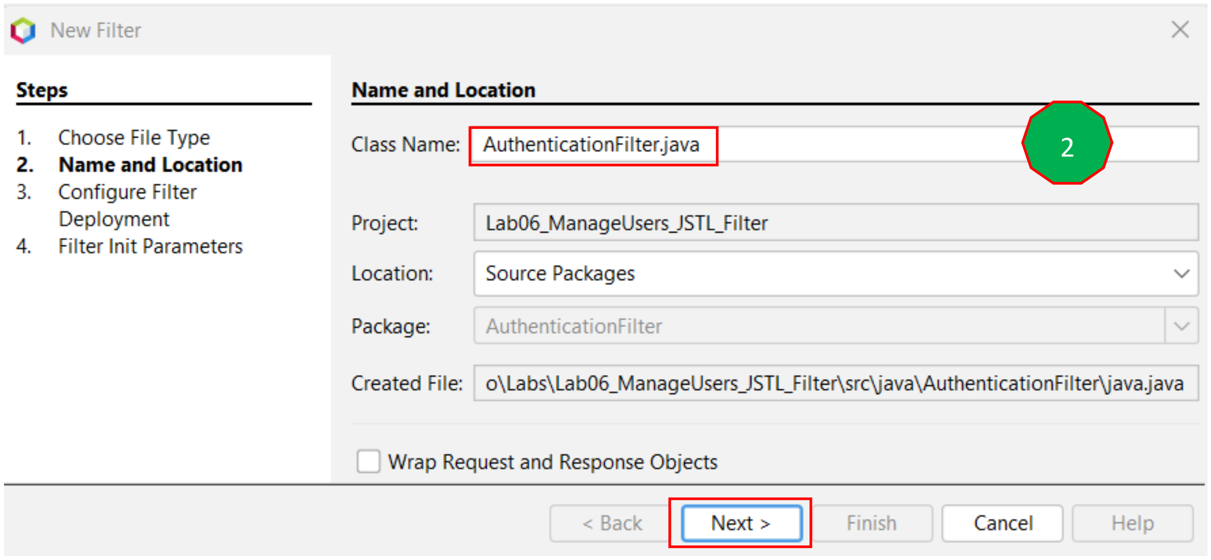
Choose File Type

Project: Lab06_ManageUsers_JSTL_Filter

Filter:

Categories: Web, JavaServer Faces, Bean Validation, Struts

File Types: JSP, JSF Page, Servlet, Filter



New Filter

Steps

1. Choose File Type
2. **Name and Location**
3. Configure Filter Deployment
4. Filter Init Parameters

Name and Location

Class Name: AuthenticationFilter.java

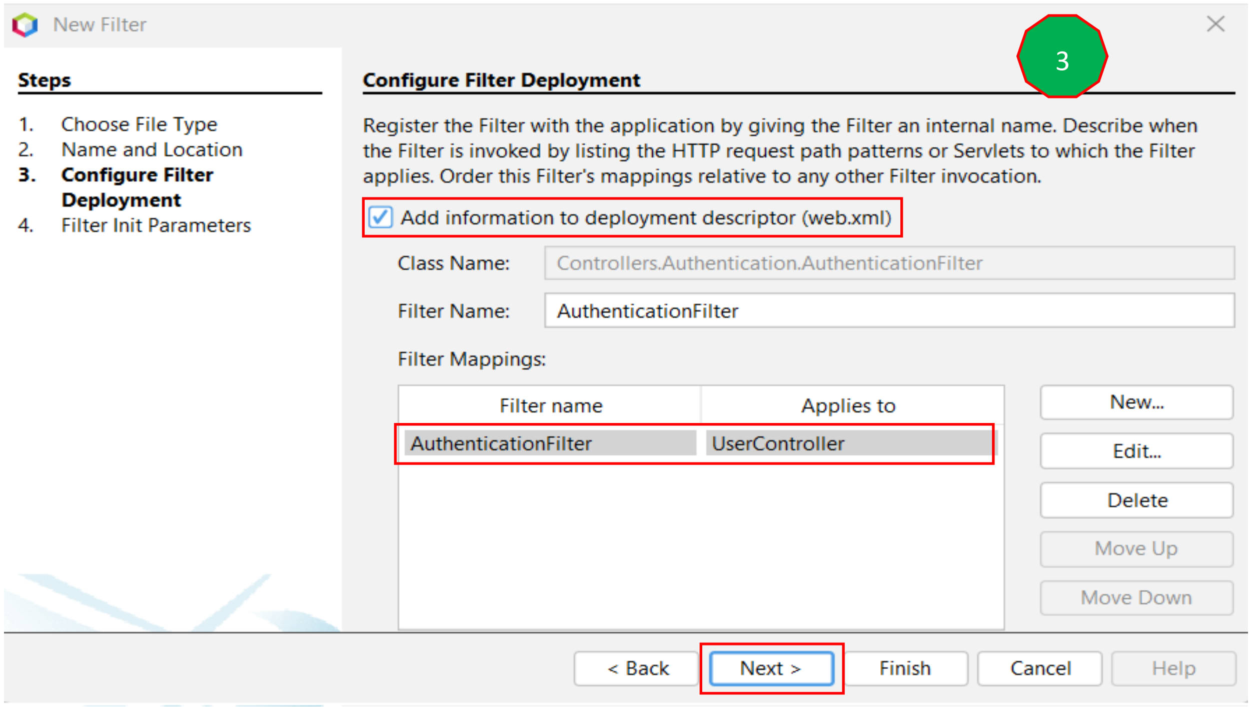
Project: Lab06_ManageUsers_JSTL_Filter

Location: Source Packages

Package: AuthenticationFilter

Created File: o\Labs\Lab06_ManageUsers_JSTL_Filter\src\java\AuthenticationFilter\java.java

☐ Wrap Request and Response Objects



New Filter

Steps

1. Choose File Type
2. Name and Location
3. **Configure Filter Deployment**
4. Filter Init Parameters

Configure Filter Deployment

Register the Filter with the application by giving the Filter an internal name. Describe when the Filter is invoked by listing the HTTP request path patterns or Servlets to which the Filter applies. Order this Filter's mappings relative to any other Filter invocation.

☑ Add information to deployment descriptor (web.xml)

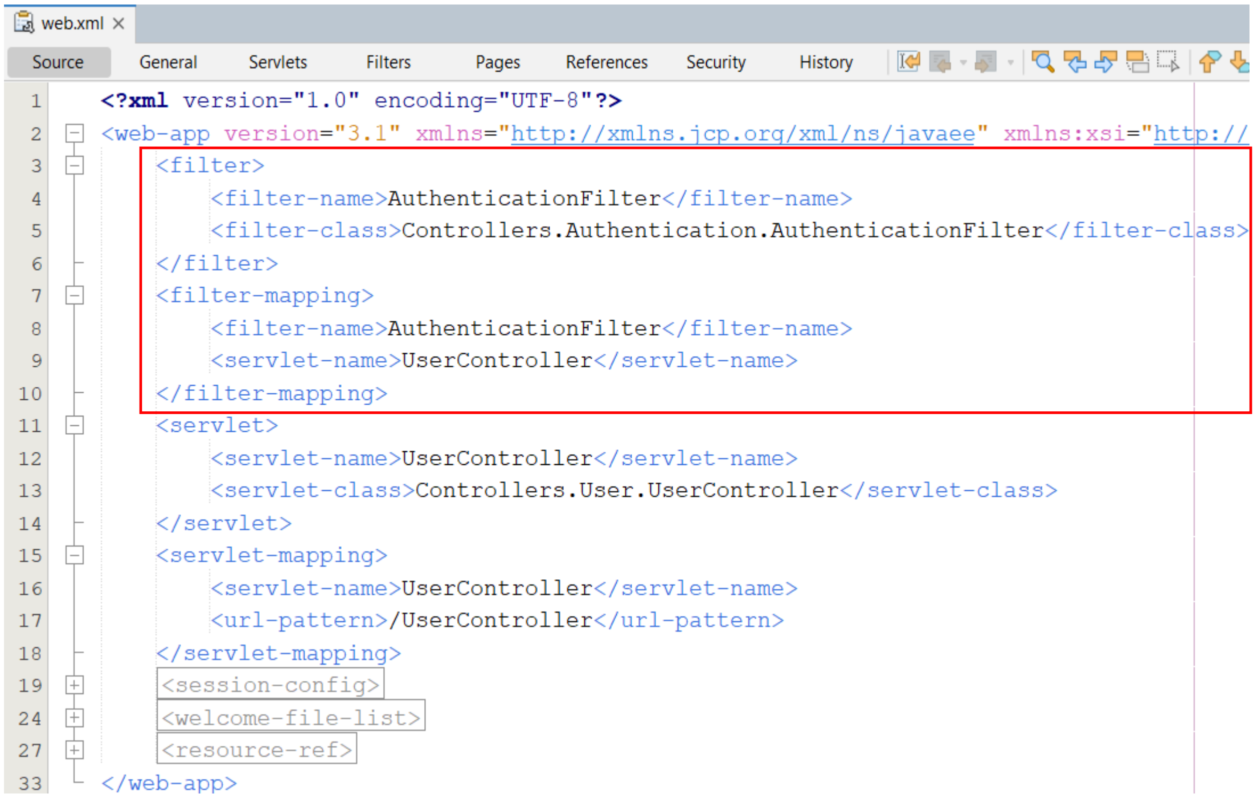
Class Name: Controllers.Authentication.AuthenticationFilter

Filter Name: AuthenticationFilter

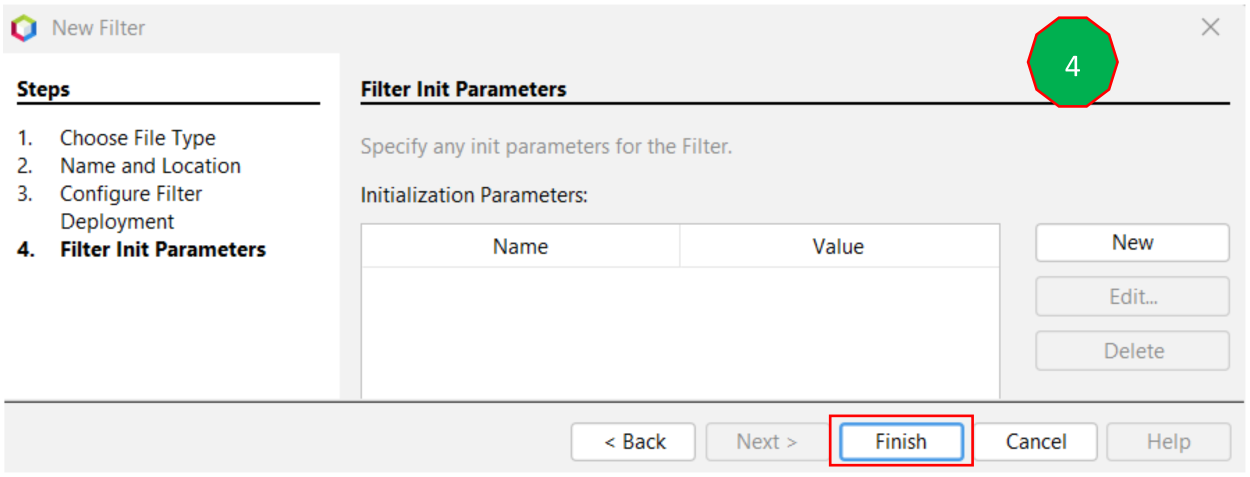
Filter Mappings:

Filter name	Applies to
AuthenticationFilter	UserController

-Open the **web.xml** in the WEB-INF folder to view configuration information of the **AuthenticationFilter** and update if not available as follows:



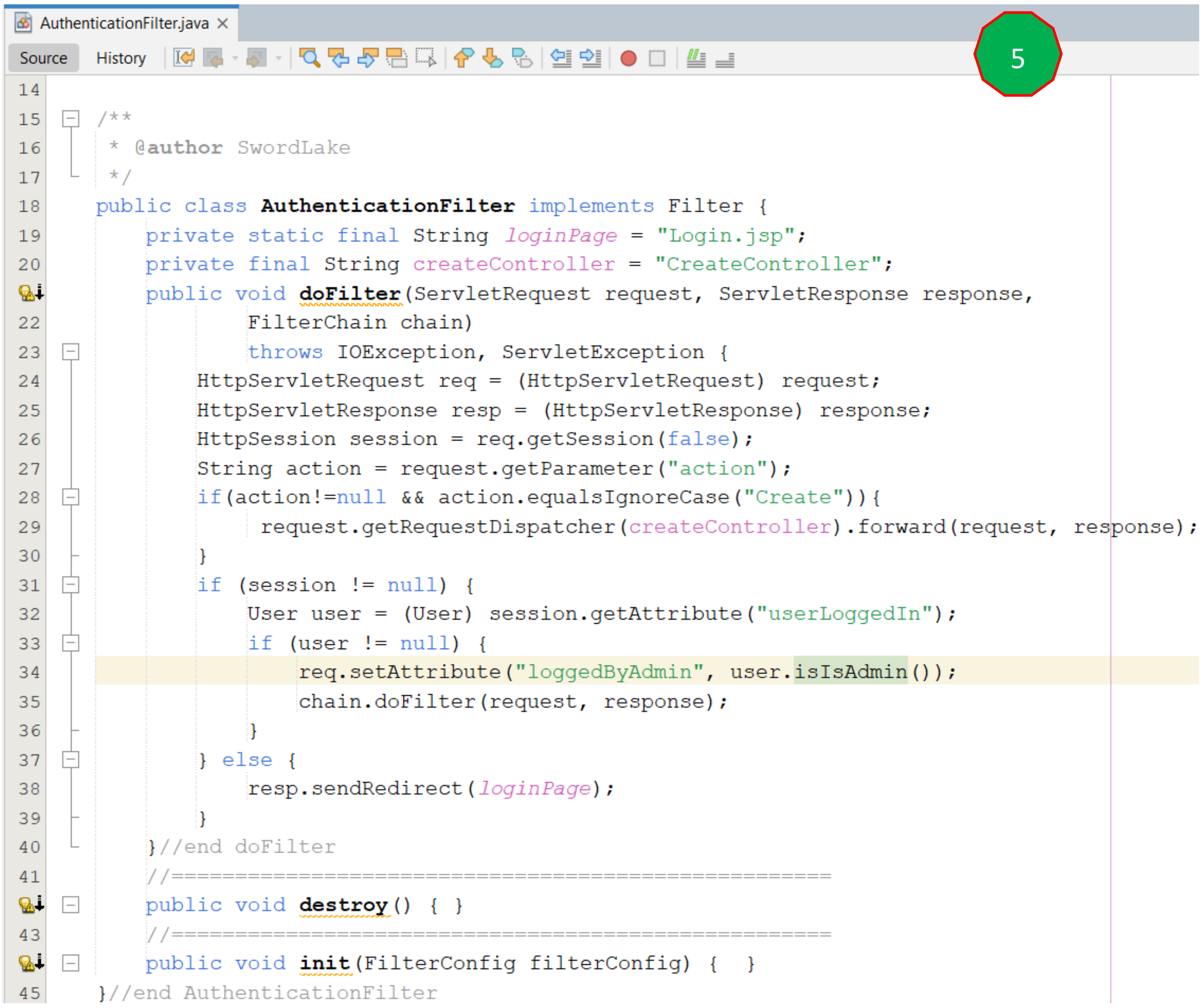
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://
    <filter>
        <filter-name>AuthenticationFilter</filter-name>
        <filter-class>Controllers.Authentication.AuthenticationFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>AuthenticationFilter</filter-name>
        <servlet-name>UserController</servlet-name>
    </filter-mapping>
    <servlet>
        <servlet-name>UserController</servlet-name>
        <servlet-class>Controllers.User.UserController</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>UserController</servlet-name>
        <url-pattern>/UserController</url-pattern>
    </servlet-mapping>
    <session-config>
    <welcome-file-list>
    <resource-ref>
</web-app>
```



Specify any init parameters for the Filter.

Initialization Parameters:

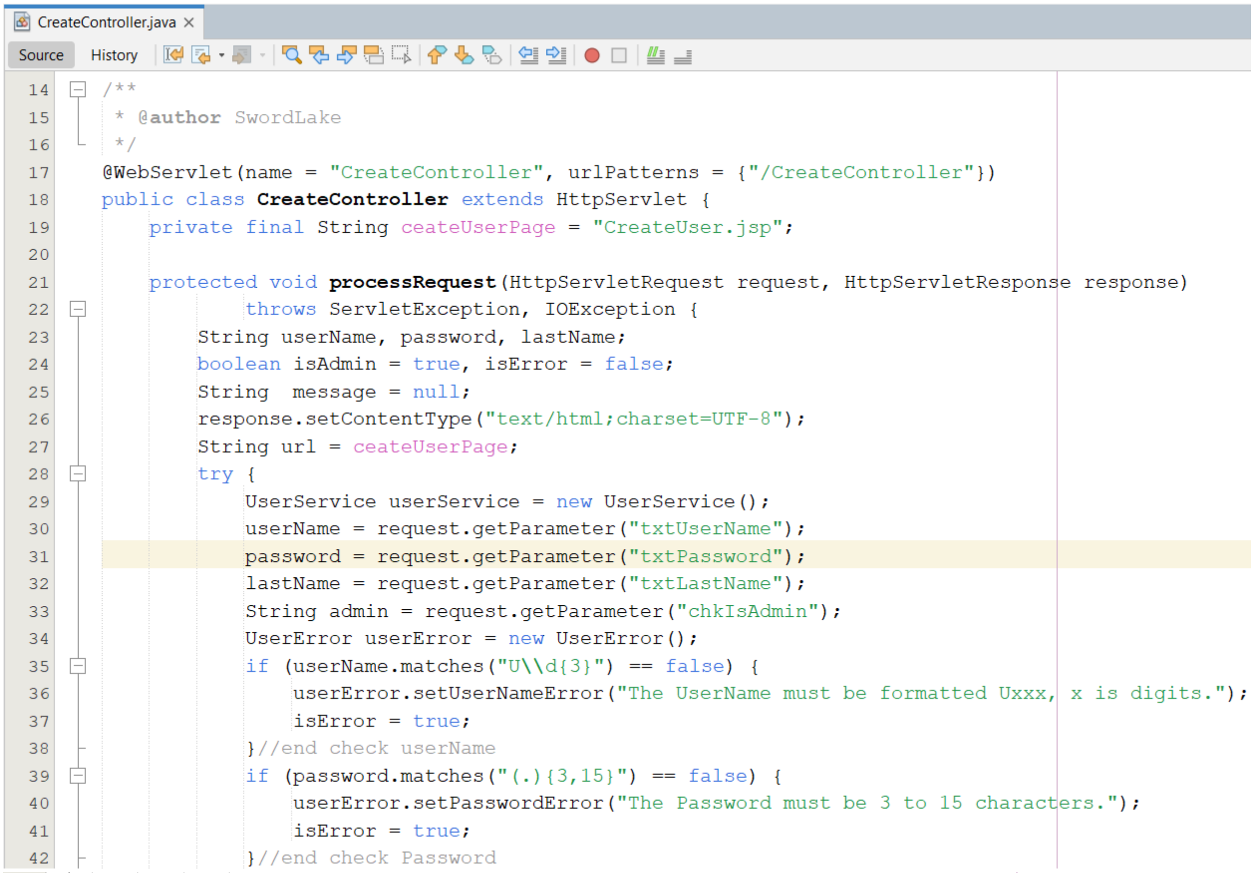
- Write codes for **AuthenticationFilter.java** as follows:



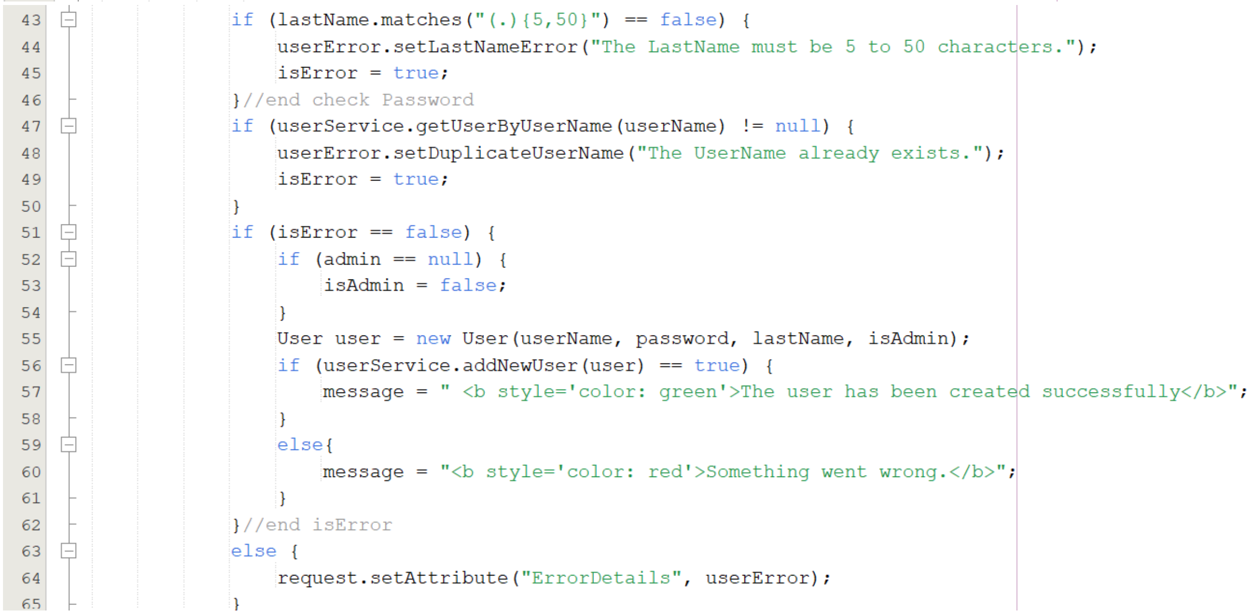
```java
/**
 * @author SwordLake
 */
public class AuthenticationFilter implements Filter {
    private static final String loginPage = "Login.jsp";
    private final String createController = "CreateController";
    public void doFilter(ServletRequest request, ServletResponse response,
            FilterChain chain)
            throws IOException, ServletException {
        HttpServletRequest req = (HttpServletRequest) request;
        HttpServletResponse resp = (HttpServletResponse) response;
        HttpSession session = req.getSession(false);
        String action = request.getParameter("action");
        if(action!=null && action.equalsIgnoreCase("Create")){
             request.getRequestDispatcher(createController).forward(request, response);
        }
        if (session != null) {
            User user = (User) session.getAttribute("userLoggedIn");
            if (user != null) {
                req.setAttribute("loggedByAdmin", user.isIsAdmin());
                chain.doFilter(request, response);
            }
        } else {
            resp.sendRedirect(loginPage);
        }
    }//end doFilter
    //====================================================
    public void destroy() { }
    //====================================================
    public void init(FilterConfig filterConfig) {   }
}//end AuthenticationFilter
```

Step 04. Write codes for the classes in the **Controllers.User** package is as follows:

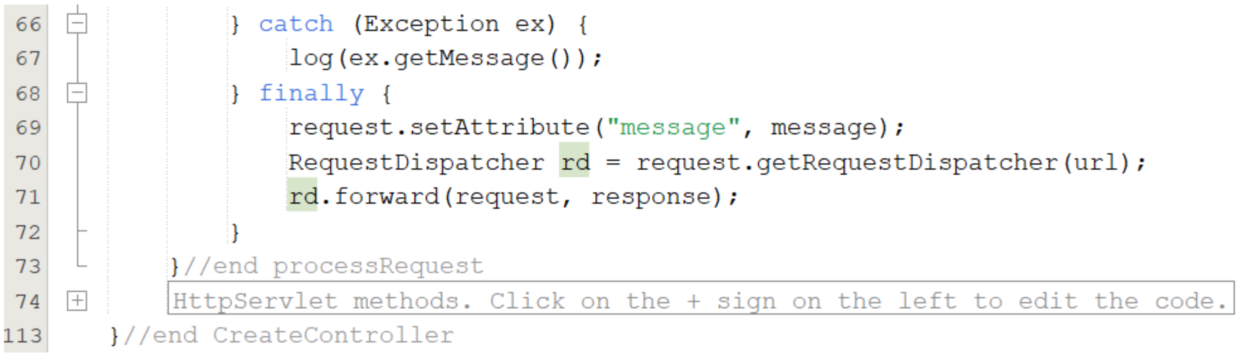
- **CreateController.java**



```java
/**
 * @author SwordLake
 */
@WebServlet(name = "CreateController", urlPatterns = {"/CreateController"})
public class CreateController extends HttpServlet {
    private final String ceateUserPage = "CreateUser.jsp";

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String userName, password, lastName;
        boolean isAdmin = true, isError = false;
        String  message = null;
        response.setContentType("text/html;charset=UTF-8");
        String url = ceateUserPage;
        try {
            UserService userService = new UserService();
            userName = request.getParameter("txtUserName");
            password = request.getParameter("txtPassword");
            lastName = request.getParameter("txtLastName");
            String admin = request.getParameter("chkIsAdmin");
            UserError userError = new UserError();
            if (userName.matches("U\\d{3}") == false) {
                userError.setUserNameError("The UserName must be formatted Uxxx, x is digits.");
                isError = true;
            }//end check userName
            if (password.matches("(.){3,15}") == false) {
                userError.setPasswordError("The Password must be 3 to 15 characters.");
                isError = true;
            }//end check Password
```

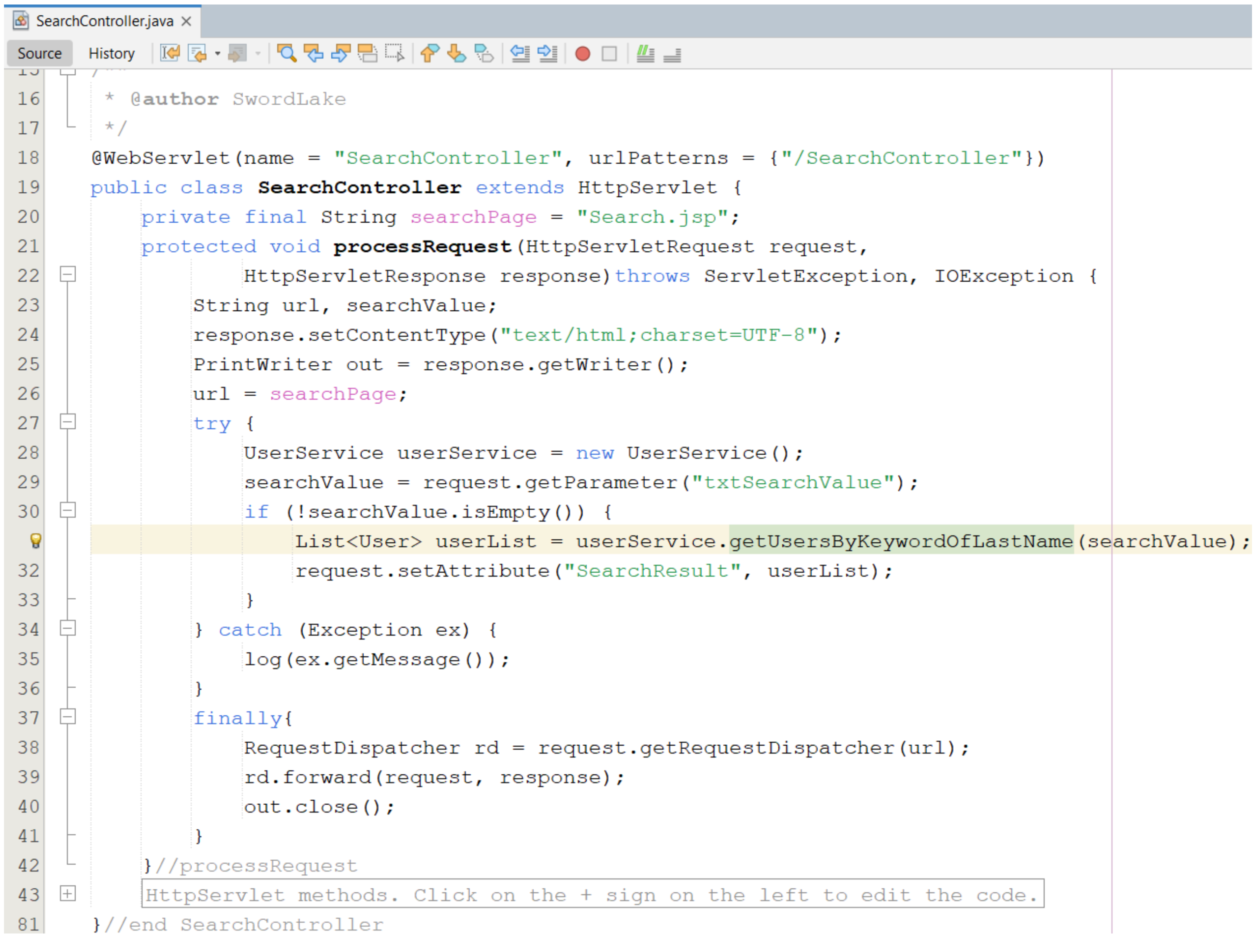


```java
            if (lastName.matches("(.){5,50}") == false) {
                userError.setLastNameError("The LastName must be 5 to 50 characters.");
                isError = true;
            }//end check Password
            if (userService.getUserByUserName(userName) != null) {
                userError.setDuplicateUserName("The UserName already exists.");
                isError = true;
            }
            if (isError == false) {
                if (admin == null) {
                    isAdmin = false;
                }
                User user = new User(userName, password, lastName, isAdmin);
                if (userService.addNewUser(user) == true) {
                    message = " <b style='color: green'>The user has been created successfully</b>";
                }
                else{
                    message = "<b style='color: red'>Something went wrong.</b>";
                }
            }//end isError
            else {
                request.setAttribute("ErrorDetails", userError);
            }
```



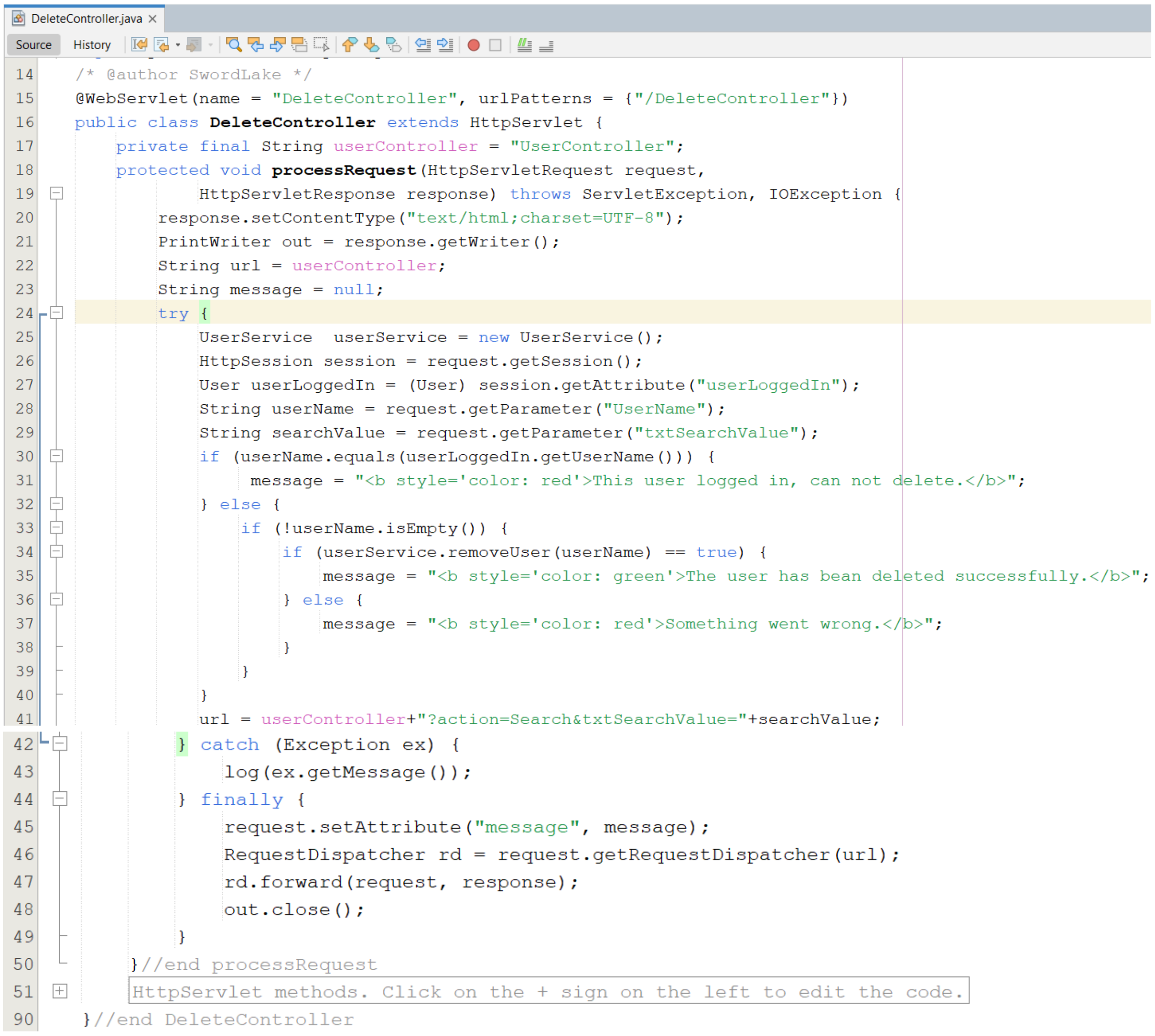
```
            } catch (Exception ex) {
                log(ex.getMessage());
            } finally {
                request.setAttribute("message", message);
                RequestDispatcher rd = request.getRequestDispatcher(url);
                rd.forward(request, response);
            }
        }//end processRequest
        HttpServlet methods. Click on the + sign on the left to edit the code.
    }//end CreateController
```

- **SearchController.java**



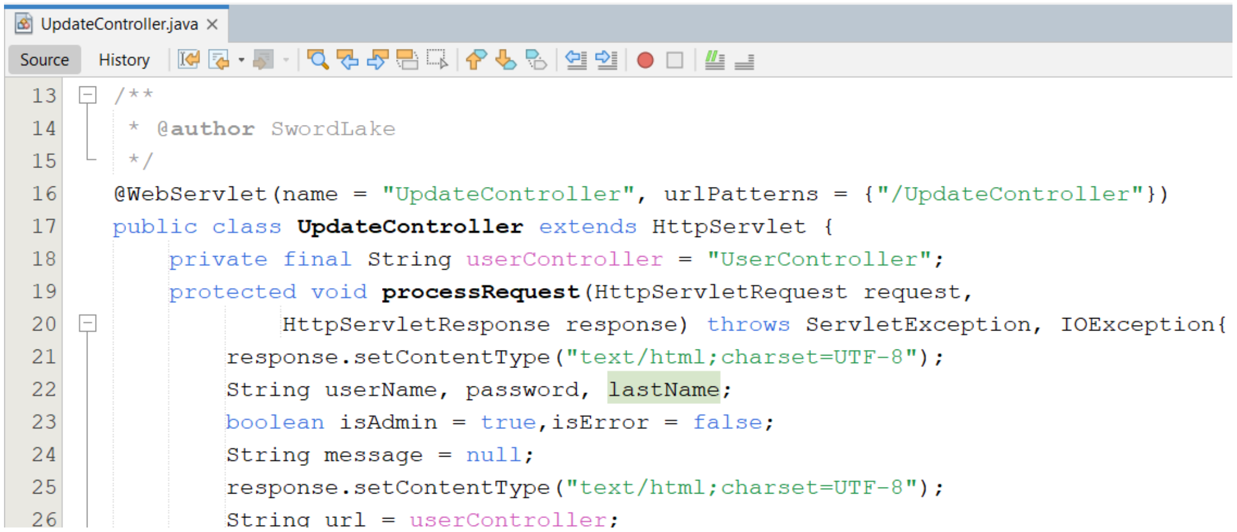
```
     * @author SwordLake
     */
    @WebServlet(name = "SearchController", urlPatterns = {"/SearchController"})
    public class SearchController extends HttpServlet {
        private final String searchPage = "Search.jsp";
        protected void processRequest(HttpServletRequest request,
                HttpServletResponse response)throws ServletException, IOException {
            String url, searchValue;
            response.setContentType("text/html;charset=UTF-8");
            PrintWriter out = response.getWriter();
            url = searchPage;
            try {
                UserService userService = new UserService();
                searchValue = request.getParameter("txtSearchValue");
                if (!searchValue.isEmpty()) {
                    List<User> userList = userService.getUsersByKeywordOfLastName(searchValue);
                    request.setAttribute("SearchResult", userList);
                }
            } catch (Exception ex) {
                log(ex.getMessage());
            }
            finally{
                RequestDispatcher rd = request.getRequestDispatcher(url);
                rd.forward(request, response);
                out.close();
            }
        }//processRequest
        HttpServlet methods. Click on the + sign on the left to edit the code.
    }//end SearchController
```

- **DeleteController.java**

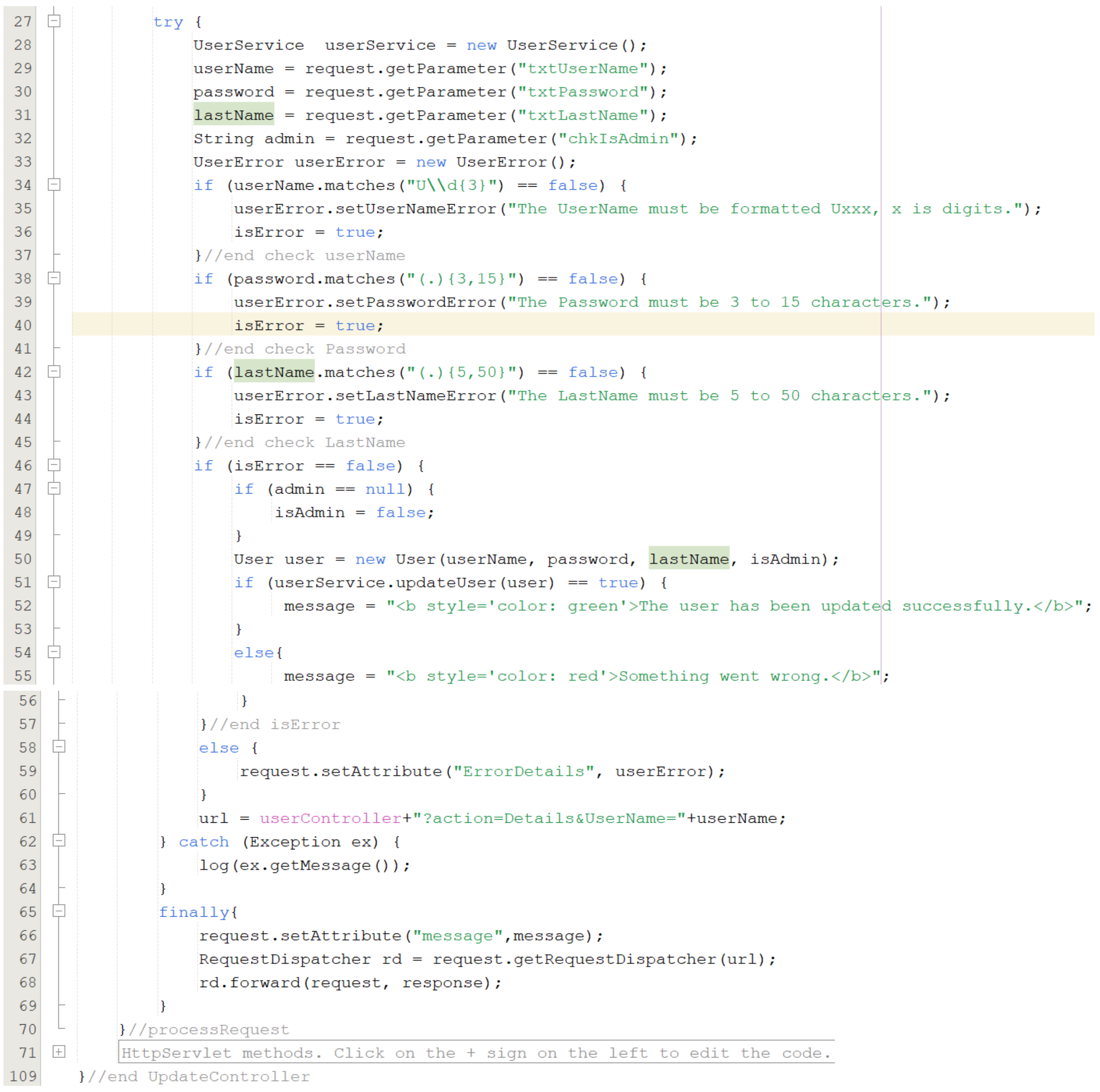


```java
/* @author SwordLake */
@WebServlet(name = "DeleteController", urlPatterns = {"/DeleteController"})
public class DeleteController extends HttpServlet {
    private final String userController = "UserController";
    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String url = userController;
        String message = null;
        try {
            UserService  userService = new UserService();
            HttpSession session = request.getSession();
            User userLoggedIn = (User) session.getAttribute("userLoggedIn");
            String userName = request.getParameter("UserName");
            String searchValue = request.getParameter("txtSearchValue");
            if (userName.equals(userLoggedIn.getUserName())) {
                message = "<b style='color: red'>This user logged in, can not delete.</b>";
            } else {
                if (!userName.isEmpty()) {
                    if (userService.removeUser(userName) == true) {
                        message = "<b style='color: green'>The user has bean deleted successfully.</b>";
                    } else {
                        message = "<b style='color: red'>Something went wrong.</b>";
                    }
                }
            }
            url = userController+"?action=Search&txtSearchValue="+searchValue;
        } catch (Exception ex) {
            log(ex.getMessage());
        } finally {
            request.setAttribute("message", message);
            RequestDispatcher rd = request.getRequestDispatcher(url);
            rd.forward(request, response);
            out.close();
        }
    }//end processRequest
    HttpServlet methods. Click on the + sign on the left to edit the code.
}//end DeleteController
```

- **UpdateController.java**

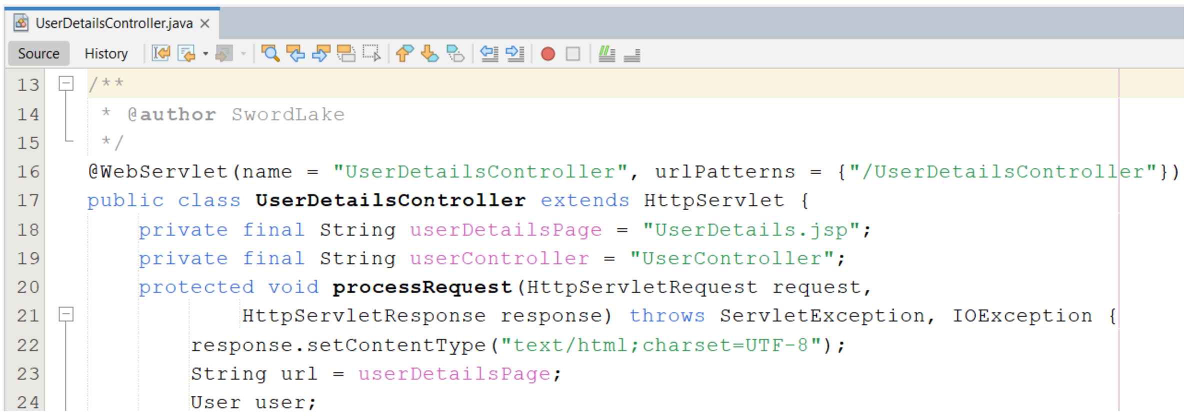


```java
/**
 * @author SwordLake
 */
@WebServlet(name = "UpdateController", urlPatterns = {"/UpdateController"})
public class UpdateController extends HttpServlet {
    private final String userController = "UserController";
    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        String userName, password, lastName;
        boolean isAdmin = true,isError = false;
        String message = null;
        response.setContentType("text/html;charset=UTF-8");
        String url = userController;
```

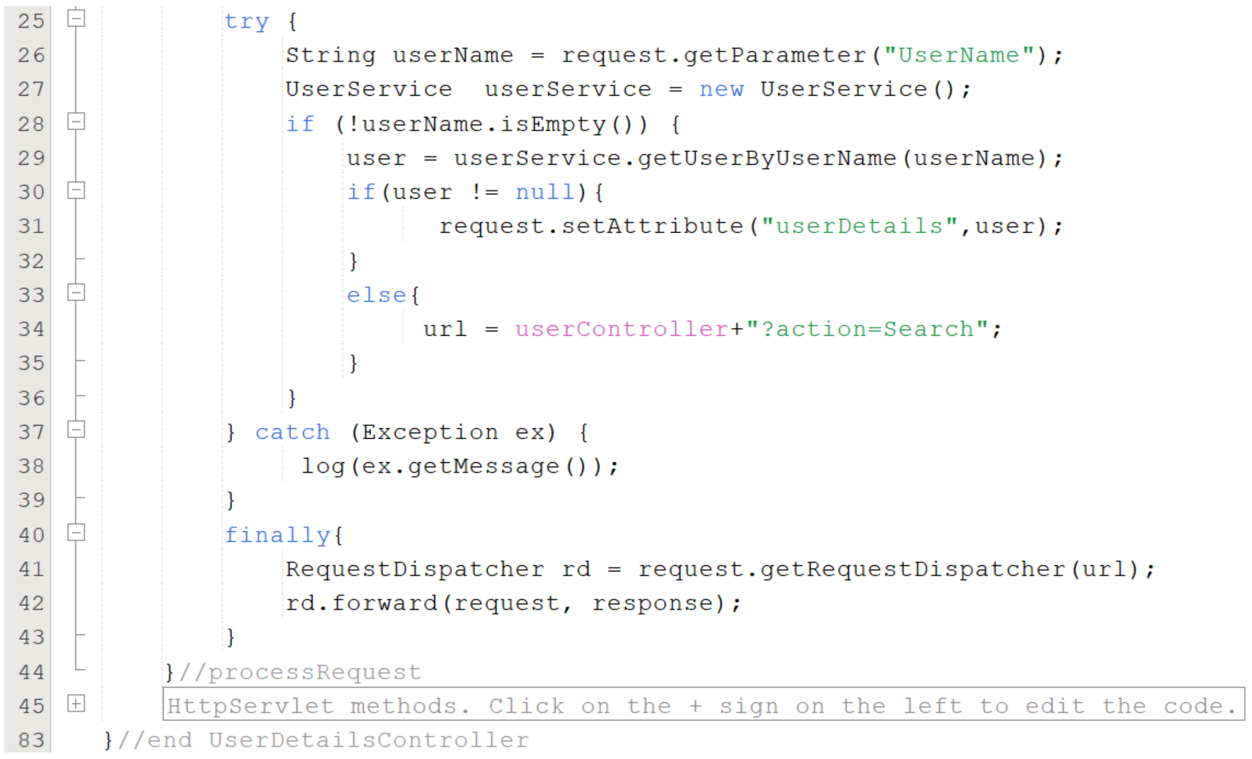


```java
        try {
            UserService  userService = new UserService();
            userName = request.getParameter("txtUserName");
            password = request.getParameter("txtPassword");
            lastName = request.getParameter("txtLastName");
            String admin = request.getParameter("chkIsAdmin");
            UserError userError = new UserError();
            if (userName.matches("U\\d{3}") == false) {
                userError.setUserNameError("The UserName must be formatted Uxxx, x is digits.");
                isError = true;
            }//end check userName
            if (password.matches("(.){3,15}") == false) {
                userError.setPasswordError("The Password must be 3 to 15 characters.");
                isError = true;
            }//end check Password
            if (lastName.matches("(.){5,50}") == false) {
                userError.setLastNameError("The LastName must be 5 to 50 characters.");
                isError = true;
            }//end check LastName
            if (isError == false) {
                if (admin == null) {
                    isAdmin = false;
                }
                User user = new User(userName, password, lastName, isAdmin);
                if (userService.updateUser(user) == true) {
                     message = "<b style='color: green'>The user has been updated successfully.</b>";
                }
                else{
                     message = "<b style='color: red'>Something went wrong.</b>";
                }
            }//end isError
            else {
                request.setAttribute("ErrorDetails", userError);
            }
            url = userController+"?action=Details&UserName="+userName;
        } catch (Exception ex) {
            log(ex.getMessage());
        }
        finally{
            request.setAttribute("message",message);
            RequestDispatcher rd = request.getRequestDispatcher(url);
            rd.forward(request, response);
        }
    }//processRequest
    HttpServlet methods. Click on the + sign on the left to edit the code.
}//end UpdateController
```

- **UserDetailsController.java**

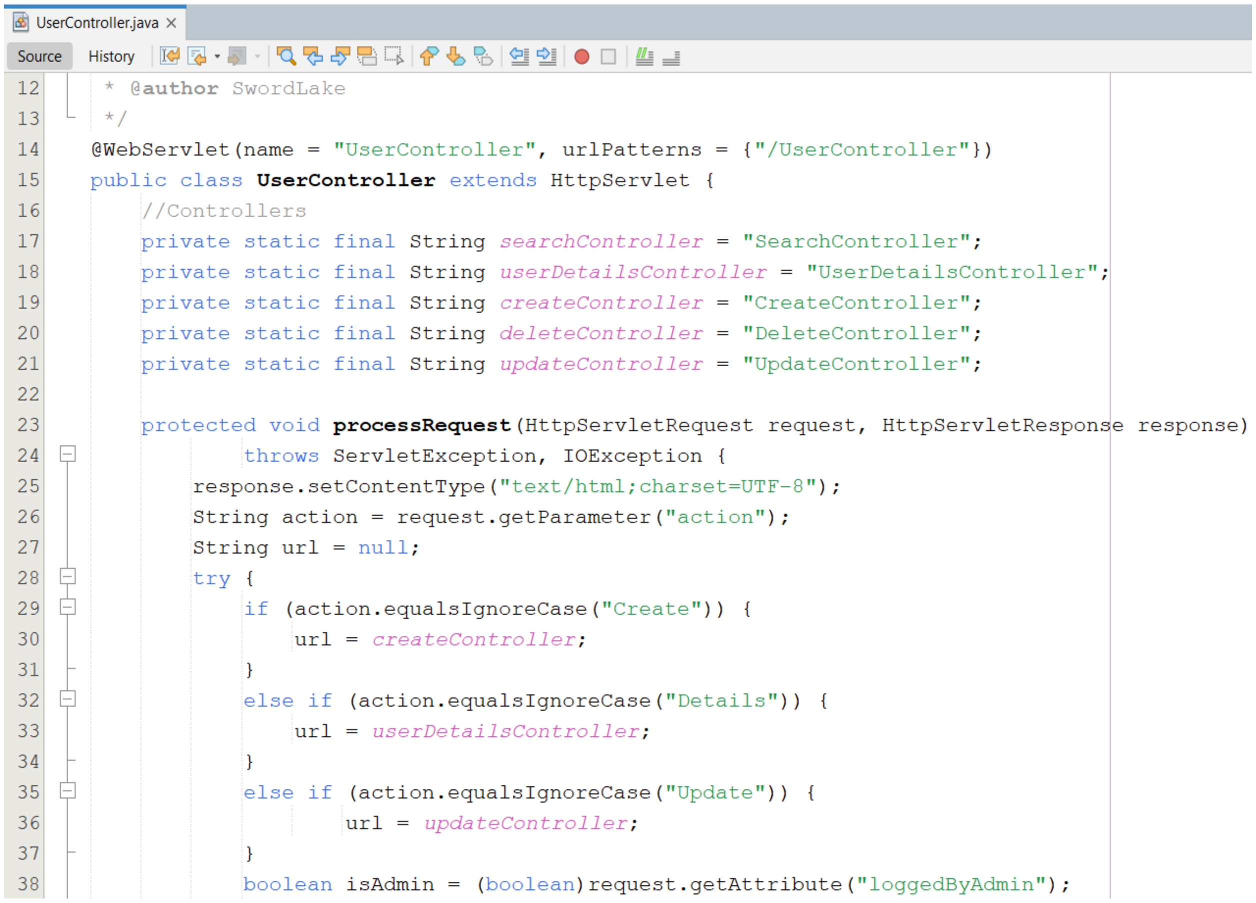


```java
/**
 * @author SwordLake
 */
@WebServlet(name = "UserDetailsController", urlPatterns = {"/UserDetailsController"})
public class UserDetailsController extends HttpServlet {
    private final String userDetailsPage = "UserDetails.jsp";
    private final String userController = "UserController";
    protected void processRequest(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String url = userDetailsPage;
        User user;
```

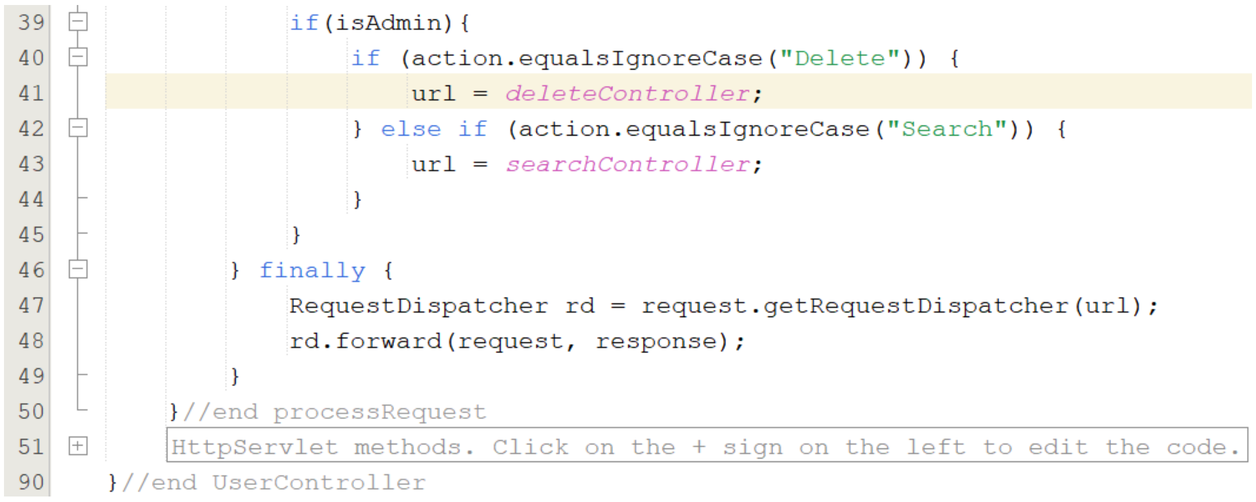


```
        try {
            String userName = request.getParameter("UserName");
            UserService  userService = new UserService();
            if (!userName.isEmpty()) {
                user = userService.getUserByUserName(userName);
                if(user != null){
                      request.setAttribute("userDetails",user);
                }
                else{
                     url = userController+"?action=Search";
                }
            }
        } catch (Exception ex) {
             log(ex.getMessage());
        }
        finally{
            RequestDispatcher rd = request.getRequestDispatcher(url);
            rd.forward(request, response);
        }
    }//processRequest
    HttpServlet methods. Click on the + sign on the left to edit the code.
}//end UserDetailsController
```

- **UserController.java**



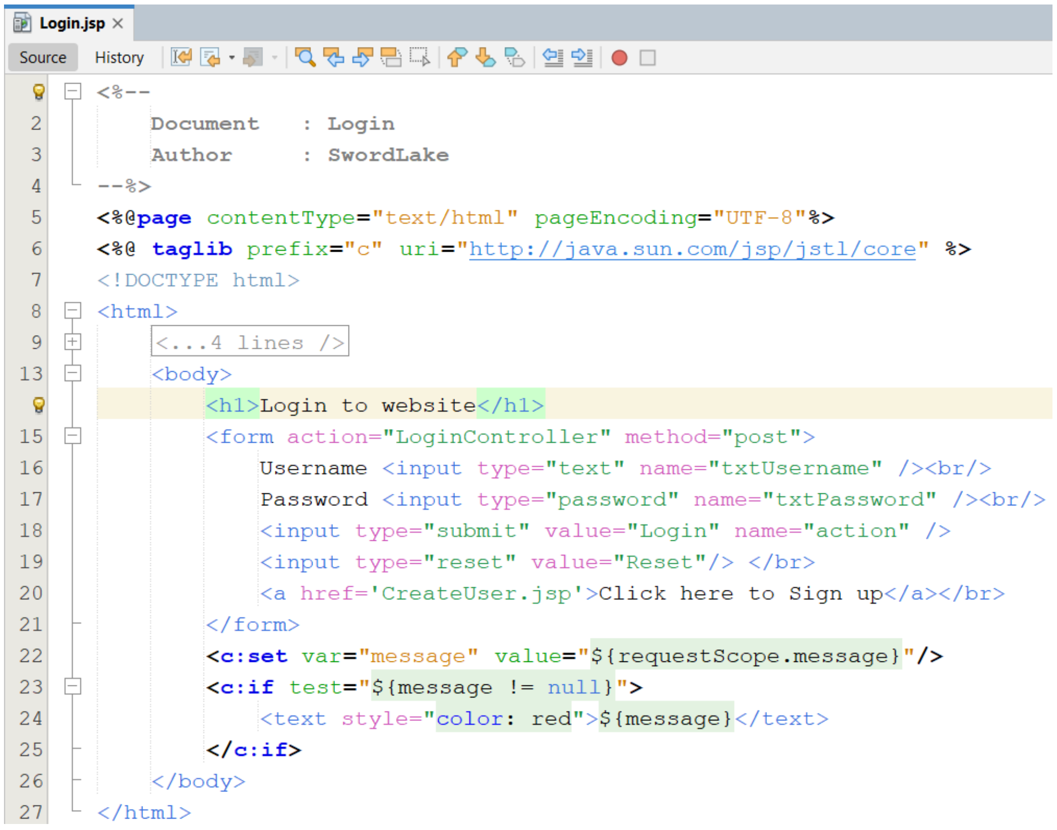
```
 * @author SwordLake
 */
@WebServlet(name = "UserController", urlPatterns = {"/UserController"})
public class UserController extends HttpServlet {
    //Controllers
    private static final String searchController = "SearchController";
    private static final String userDetailsController = "UserDetailsController";
    private static final String createController = "CreateController";
    private static final String deleteController = "DeleteController";
    private static final String updateController = "UpdateController";

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String action = request.getParameter("action");
        String url = null;
        try {
            if (action.equalsIgnoreCase("Create")) {
                url = createController;
            }
            else if (action.equalsIgnoreCase("Details")) {
                url = userDetailsController;
            }
            else if (action.equalsIgnoreCase("Update")) {
                    url = updateController;
            }
            boolean isAdmin = (boolean)request.getAttribute("loggedByAdmin");
```



```
                if(isAdmin){
                    if (action.equalsIgnoreCase("Delete")) {
                        url = deleteController;
                    } else if (action.equalsIgnoreCase("Search")) {
                        url = searchController;
                    }
                }
            } finally {
                RequestDispatcher rd = request.getRequestDispatcher(url);
                rd.forward(request, response);
            }
        }//end processRequest
        HttpServlet methods. Click on the + sign on the left to edit the code.
    }//end UserController
```

Step 05. Write codes for the web pages as follows:

- **Login.jsp page**

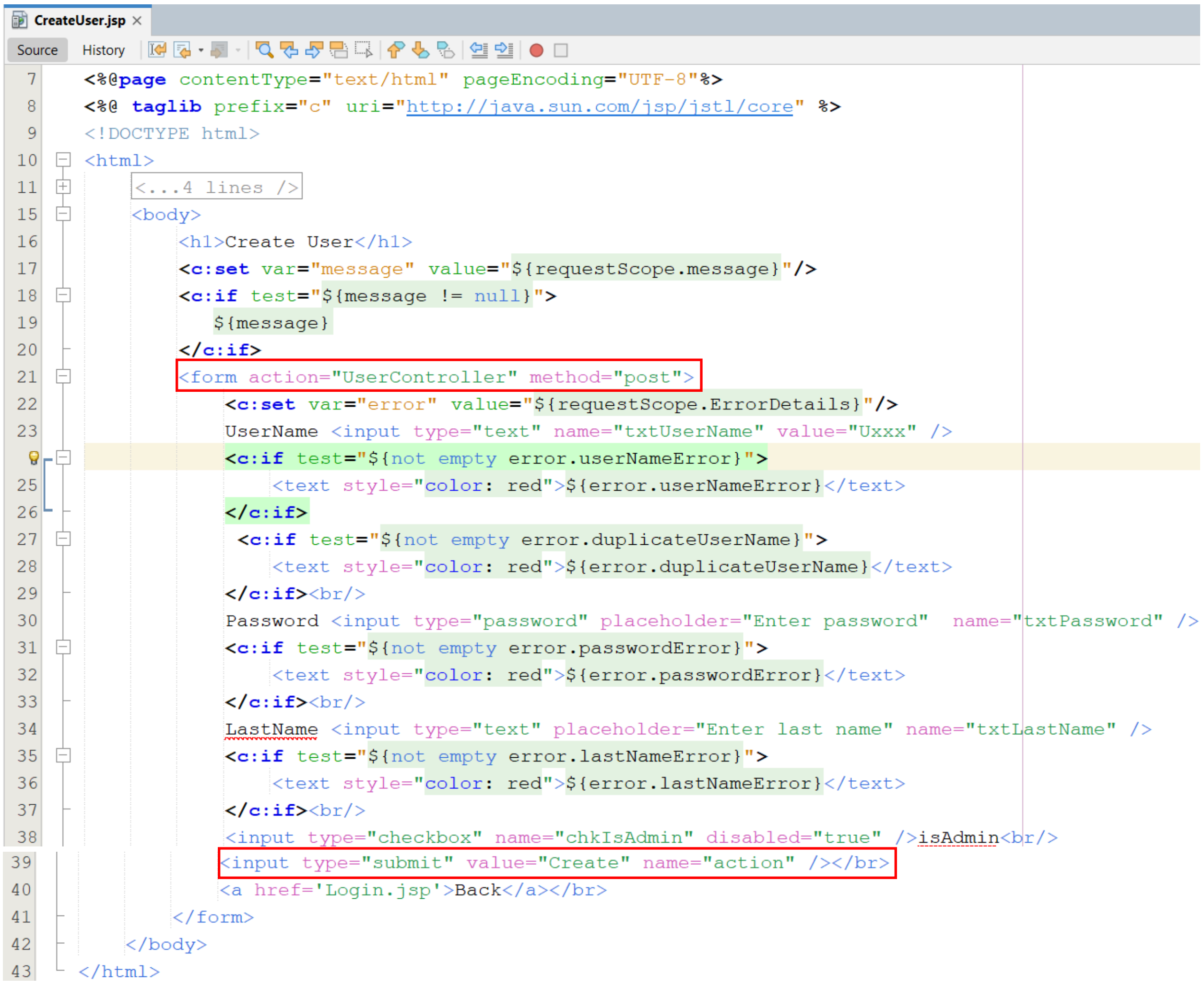


Login.jsp ×

Source History

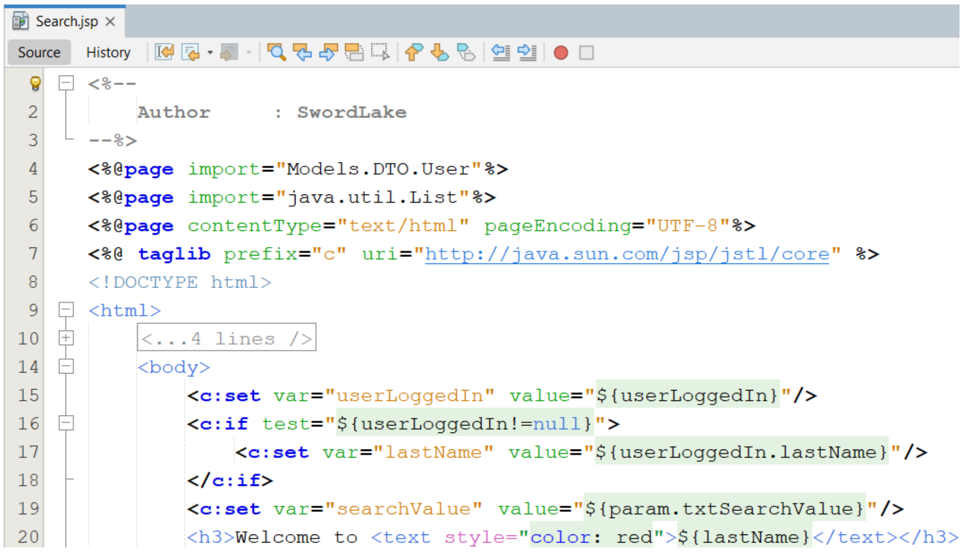
```
<%--
    Document   : Login
    Author     : SwordLake
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <...4 lines />
    <body>
        <h1>Login to website</h1>
        <form action="LoginController" method="post">
            Username <input type="text" name="txtUsername" /><br/>
            Password <input type="password" name="txtPassword" /><br/>
            <input type="submit" value="Login" name="action" />
            <input type="reset" value="Reset"/> </br>
            <a href='CreateUser.jsp'>Click here to Sign up</a></br>
        </form>
        <c:set var="message" value="${requestScope.message}"/>
        <c:if test="${message != null}">
            <text style="color: red">${message}</text>
        </c:if>
    </body>
</html>
```

- **CreateUser.jsp page**

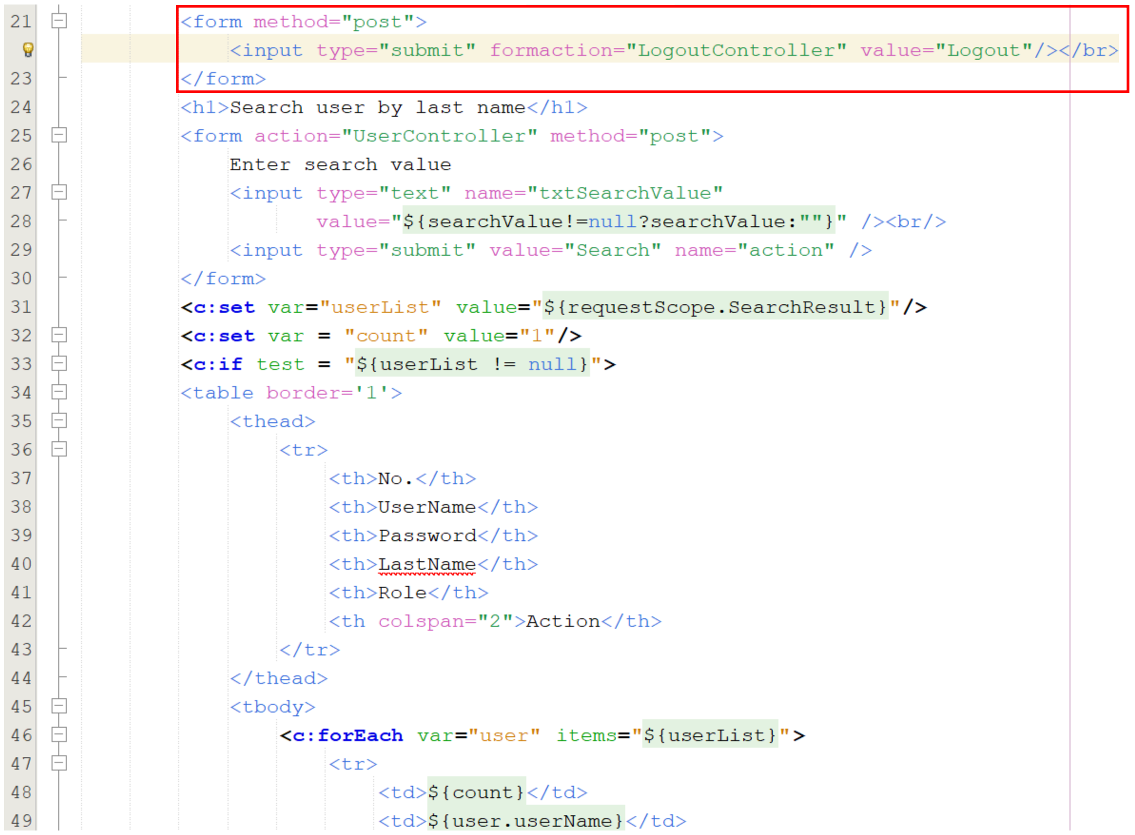


```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <...4 lines />
    <body>
        <h1>Create User</h1>
        <c:set var="message" value="${requestScope.message}"/>
        <c:if test="${message != null}">
           ${message}
        </c:if>
        <form action="UserController" method="post">
            <c:set var="error" value="${requestScope.ErrorDetails}"/>
            UserName <input type="text" name="txtUserName" value="Uxxx" />
            <c:if test="${not empty error.userNameError}">
                <text style="color: red">${error.userNameError}</text>
            </c:if>
             <c:if test="${not empty error.duplicateUserName}">
                <text style="color: red">${error.duplicateUserName}</text>
            </c:if><br/>
            Password <input type="password" placeholder="Enter password"  name="txtPassword" />
            <c:if test="${not empty error.passwordError}">
                <text style="color: red">${error.passwordError}</text>
            </c:if><br/>
            LastName <input type="text" placeholder="Enter last name" name="txtLastName" />
            <c:if test="${not empty error.lastNameError}">
                <text style="color: red">${error.lastNameError}</text>
            </c:if><br/>
            <input type="checkbox" name="chkIsAdmin" disabled="true" />isAdmin<br/>
            <input type="submit" value="Create" name="action" /></br>
            <a href='Login.jsp'>Back</a></br>
        </form>
    </body>
</html>
```

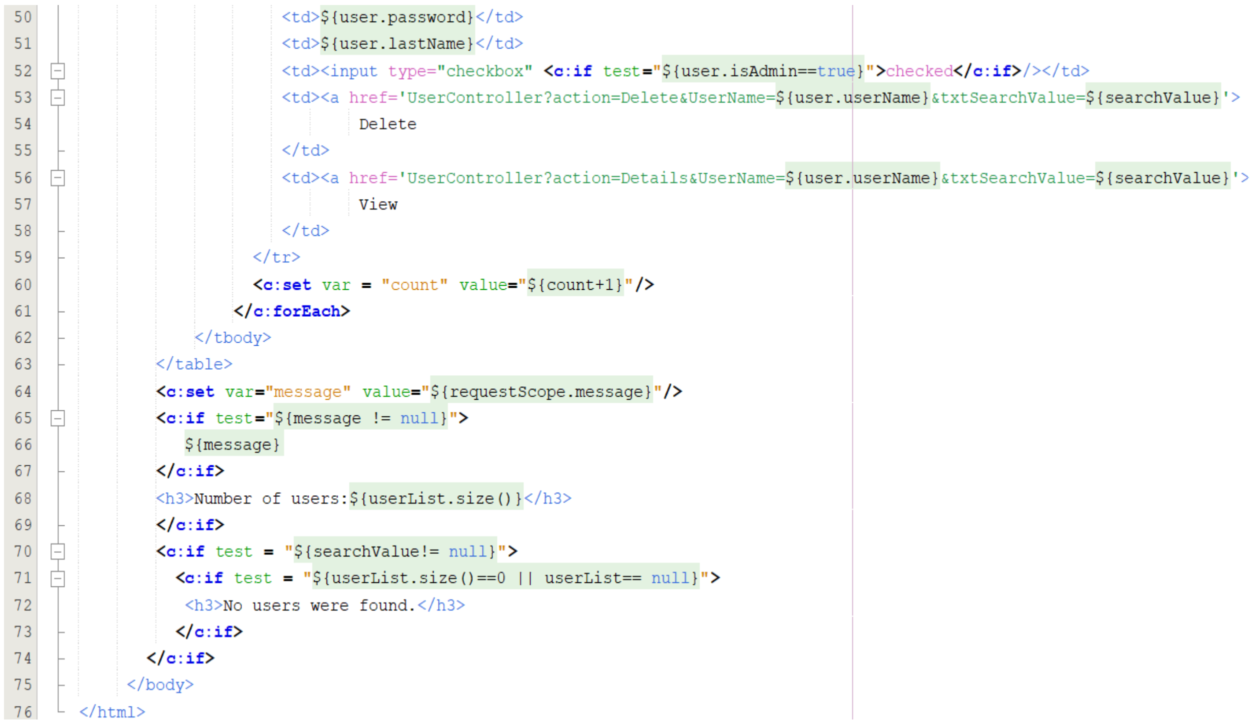
- **Search.jsp page**



```
<%--
    Author     : SwordLake
--%>
<%@page import="Models.DTO.User"%>
<%@page import="java.util.List"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <...4 lines />
    <body>
        <c:set var="userLoggedIn" value="${userLoggedIn}"/>
        <c:if test="${userLoggedIn!=null}">
            <c:set var="lastName" value="${userLoggedIn.lastName}"/>
        </c:if>
        <c:set var="searchValue" value="${param.txtSearchValue}"/>
        <h3>Welcome to <text style="color: red">${lastName}</text></h3>
```

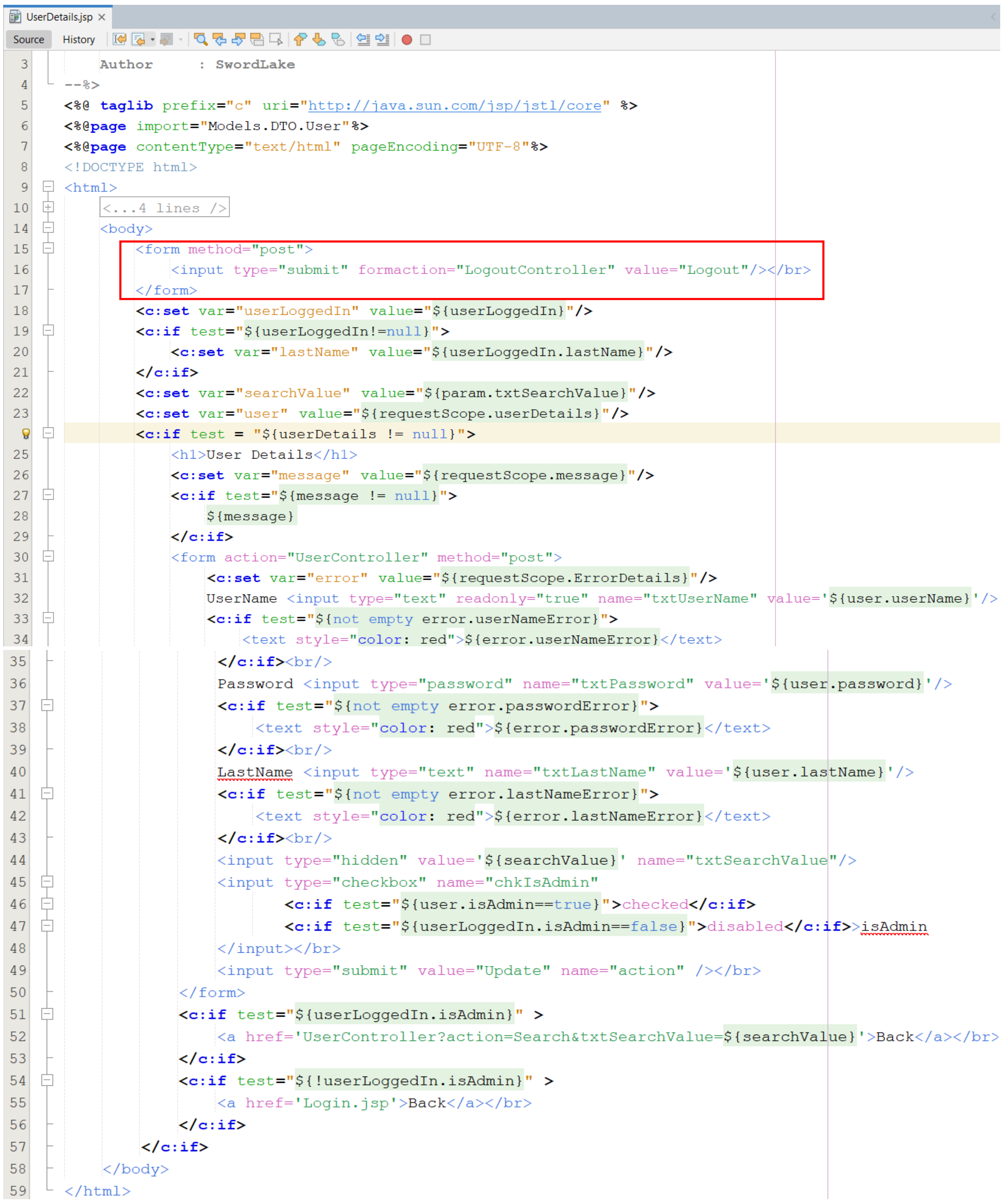


```
<form method="post">
    <input type="submit" formaction="LogoutController" value="Logout"/></br>
</form>
<h1>Search user by last name</h1>
<form action="UserController" method="post">
    Enter search value
    <input type="text" name="txtSearchValue"
           value="${searchValue!=null?searchValue:""}" /><br/>
    <input type="submit" value="Search" name="action" />
</form>
<c:set var="userList" value="${requestScope.SearchResult}"/>
<c:set var = "count" value="1"/>
<c:if test = "${userList != null}">
<table border='1'>
    <thead>
        <tr>
            <th>No.</th>
            <th>UserName</th>
            <th>Password</th>
            <th>LastName</th>
            <th>Role</th>
            <th colspan="2">Action</th>
        </tr>
    </thead>
    <tbody>
        <c:forEach var="user" items="${userList}">
            <tr>
                <td>${count}</td>
                <td>${user.userName}</td>
```



```
                     <td>${user.password}</td>
                     <td>${user.lastName}</td>
                     <td><input type="checkbox" <c:if test="${user.isAdmin==true}">checked</c:if>/></td>
                     <td><a href='UserController?action=Delete&UserName=${user.userName}&txtSearchValue=${searchValue}'>
                             Delete
                     </td>
                     <td><a href='UserController?action=Details&UserName=${user.userName}&txtSearchValue=${searchValue}'>
                             View
                     </td>
                  </tr>
                  <c:set var = "count" value="${count+1}"/>
                </c:forEach>
            </tbody>
        </table>
        <c:set var="message" value="${requestScope.message}"/>
        <c:if test="${message != null}">
           ${message}
        </c:if>
        <h3>Number of users:${userList.size()}</h3>
        </c:if>
        <c:if test = "${searchValue!= null}">
          <c:if test = "${userList.size()==0 || userList== null}">
           <h3>No users were found.</h3>
          </c:if>
       </c:if>
     </body>
</html>
```

- **UserDetails.jsp page**



```
        Author      : SwordLake
    --%>
    <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
    <%@page import="Models.DTO.User"%>
    <%@page contentType="text/html" pageEncoding="UTF-8"%>
    <!DOCTYPE html>
    <html>
        <...4 lines />
        <body>
            <form method="post">
                <input type="submit" formaction="LogoutController" value="Logout"/></br>
            </form>
            <c:set var="userLoggedIn" value="${userLoggedIn}"/>
            <c:if test="${userLoggedIn!=null}">
                <c:set var="lastName" value="${userLoggedIn.lastName}"/>
            </c:if>
            <c:set var="searchValue" value="${param.txtSearchValue}"/>
            <c:set var="user" value="${requestScope.userDetails}"/>
            <c:if test = "${userDetails != null}">
                <h1>User Details</h1>
                <c:set var="message" value="${requestScope.message}"/>
                <c:if test="${message != null}">
                    ${message}
                </c:if>
                <form action="UserController" method="post">
                    <c:set var="error" value="${requestScope.ErrorDetails}"/>
                    UserName <input type="text" readonly="true" name="txtUserName" value='${user.userName}'/>
                    <c:if test="${not empty error.userNameError}">
                        <text style="color: red">${error.userNameError}</text>
                    </c:if><br/>
                    Password <input type="password" name="txtPassword" value='${user.password}'/>
                    <c:if test="${not empty error.passwordError}">
                        <text style="color: red">${error.passwordError}</text>
                    </c:if><br/>
                    LastName <input type="text" name="txtLastName" value='${user.lastName}'/>
                    <c:if test="${not empty error.lastNameError}">
                        <text style="color: red">${error.lastNameError}</text>
                    </c:if><br/>
                    <input type="hidden" value='${searchValue}' name="txtSearchValue"/>
                    <input type="checkbox" name="chkIsAdmin"
                           <c:if test="${user.isAdmin==true}">checked</c:if>
                           <c:if test="${userLoggedIn.isAdmin==false}">disabled</c:if>>isAdmin
                    </input></br>
                    <input type="submit" value="Update" name="action" /></br>
                </form>
                <c:if test="${userLoggedIn.isAdmin}" >
                    <a href='UserController?action=Search&txtSearchValue=${searchValue}'>Back</a></br>
                </c:if>
                <c:if test="${!userLoggedIn.isAdmin}" >
                    <a href='Login.jsp'>Back</a></br>
                </c:if>
            </c:if>
        </body>
    </html>
```

Step 06. Right-click on **Lab07_ManageUsers_JSTL_Filter** project, select Run to run the web app then test all functions.