CZ3002 Advanced Software Engineering

Assignment 2, AY2020-21 Semester 2

Gupta Jay (U1822549K)
Li Wei-Ting (U1823750J)
Tai Wei Shen Wilson (U1920566A)

School of Computer Science & Engineering
Nanyang Technological University, Singapore

Team Structure

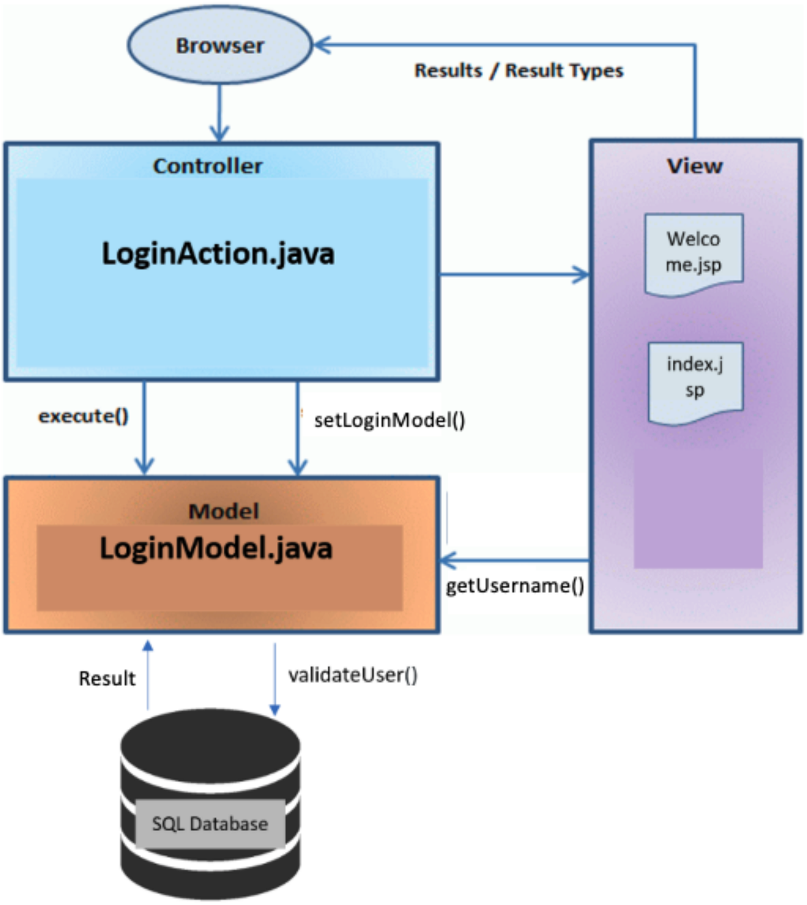
Name	Contribution
Gupta Jay	Struts Setup, JSP Pages, Documentation
Li Wei-Ting	MySQL Database, Documentation
Tai Wei Shen Wilson	UI/UX of JSP Pages, Documentation

Introduction on MVC

The MVC is an architectural design pattern that consists of 3 main components: Model, View, and Controller. This pattern is commonly used for developing modern user interfaces as it provides separation of concern that leads to high maintainability and extensibility. Every object can have multiple presentations with separate interactions.

A model holds the data and conducts computational operations to prepare the data from data sources. View refers to the user interface with the data gathered from the model. Controller alters both models and views by accepting users' inputs.

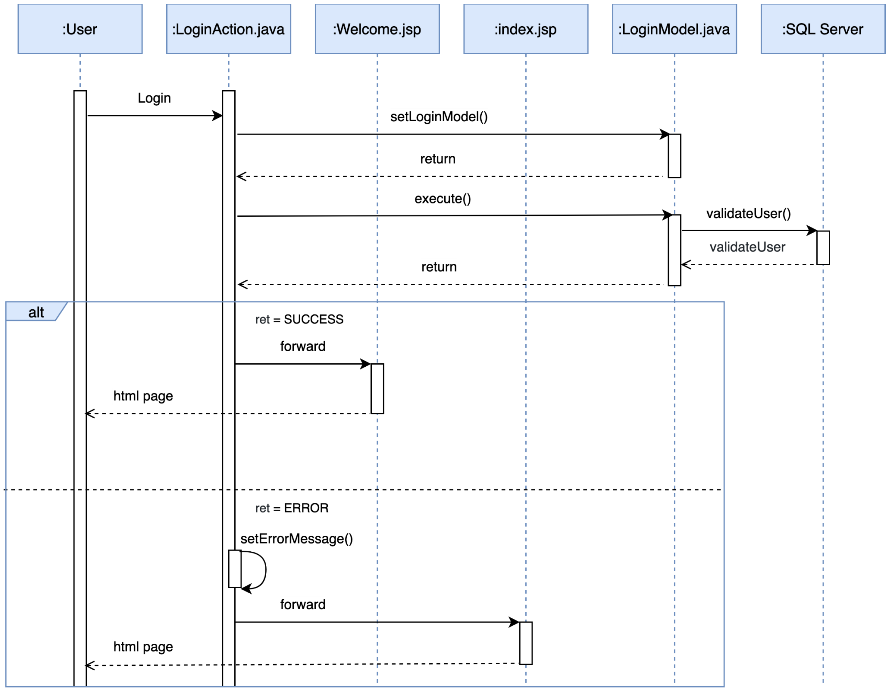
Architectural Diagram



Our login page utilises the MVC architecture. All the JSPs, `index.jsp` and `Welcome.jsp` are the views that display information to the user through the browser. `LoginAction.java` is the controller that accepts the input from the user and sets the LoginModel with the login details provided by the user. It then calls functions from the model to verify the details and decides which view to show depending on whether the login authentication is successful. `LoginModel.java` is the model class that handles data such as the username and password. `LoginModel.java` interacts with the SQL database to determine whether the login information provided is found in the database.

Once the credentials are verified, the `Welcome.jsp` view fetches the authenticated username from the model and displays it in a greeting message to the user. In the case of incorrect credentials, the user is directed back to the login view (`index.jsp`) with an error message.

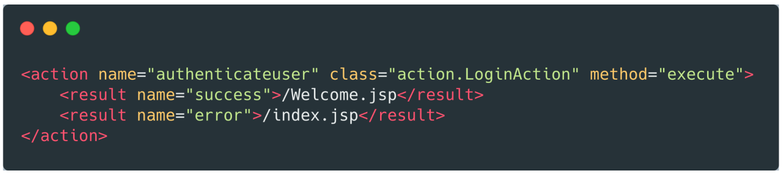
Execution Flow



1) A request is sent to the server for the URL *http://localhost:8080/login-struts/index.action*. A web container will receive the request.
2) With the `web.xml` file, all requests are routed via `StrutsPrepareAndExecuteFilter`, to our application.
3) Controller gets the information for the action, `LoginAction` from the ActionMapper.
4) The user enters their username and password using the form provided.
5) The action creates and sets the `LoginModel` object with the values submitted by the user and the framework calls the Action's `execute` method.
6) The `execute` method validates the credentials, using the `validateUser` method which queries the MySQL database. The `execute` method returns 'success' or returns 'error'.
7) In the case of an error, an error message is set by the Action class with the `setErrorMessage()` method.
8) The framework checks the action mapping for page loading, loading `Welcome.jsp` page on successful login and `index.jsp` page on an error.
9) While the page `Welcome.jsp` is being processed, the `<s:property value = "loginModel.username"/>` tag calls the `loginModel` function to return the username to display on the web-page.
10) A pure HTML response is sent back to the browser.

Dynamic Binding

Dynamic Binding occurs when methods being called upon their objects, are determined during the runtime. In this concept, the compiler does not resolve the binding at compile time but at run time. This can be seen through 'Wiring-By-Configuration', where components and wired together are runtime according to the configuration file, `struts.xml`.



```
<action name="authenticateuser" class="action.LoginAction" method="execute">
    <result name="success">/Welcome.jsp</result>
    <result name="error">/index.jsp</result>
</action>
```

The class loader uses Java reflection to instantiate objects. There is no need to re-compile the entire system, every time a change occurs.

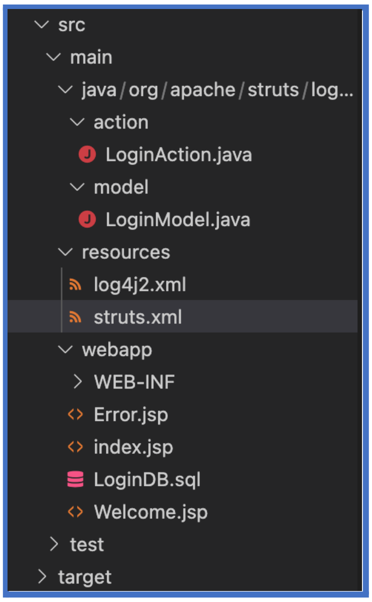
The struts.xml file promotes maintainability since it is easy to handle changes in the application using the `struts.xml` file. Let us suppose, the development team decides to let go of password-based verification, and implement Two-Factor Authentication (2FA) instead. After the change in the code is finished, we can simply update the `struts.xml` file to switch to a different authentication class, all during the runtime.

Code

Organization

The code is primarily divided in 3 segments: Model (`LoginModel.jsp`), View (`index.jsp, Welcome.jsp`), and Controller (`LoginAction.jsp`).

For the Struts 2 Framework, a `struts.xml` file is made for configuration management. It maps all the actions to different routes as part of the wire-by-configuration feature.



```
∨ src
  ∨ main
    ∨ java / org / apache / struts / log...
      ∨ action
        LoginAction.java
      ∨ model
        LoginModel.java
    ∨ resources
      log4j2.xml
      struts.xml
    ∨ webapp
      > WEB-INF
      Error.jsp
      index.jsp
      LoginDB.sql
      Welcome.jsp
  > test
> target
```

Installation

Make sure you have the following dependencies available:

- Linux, Mac OS X, or Windows.
- JAVA SE Development Kit 8
- MySQL Community Server 8.0.23 *(user: root, password: password)*
- Apache Maven 3.8.1

Run the following steps to set up your environment:

1. To set up the MySQL Database, turn on the database server. Run `mysql -u root -p` command. Run the SQL queries given in the SQL script *LoginDB.sql*.
2. Run `mvn clean package`' from the project's root folder.
3. To start the application, run `mvn jetty:run`
4. In a web browser go to *http://localhost:8080/login-struts/index.action*.

User Manual

The application features a Login feature, where the user is required to enter their credentials, consisting of a username and password, to successfully log in.

In a case of a successful login, the user is directed to a welcome page and greeted with a 'Successful Login' message. Otherwise, the user is redirected back to the home page with an 'Unsuccessful Login' message.

As per the predefined SQL queries, these credentials will result in a successful login:

- Username: `jaygupta`
- Password: `password`

References

[1] The Apache Software Foundation. 2017. "Hello World Using Struts 2." Struts. https://struts.apache.org/getting-started/hello-world-using-struts2.html#how-the-code-works.

[2] NTU. 2021. "Solution to Tutorial 8." In *Tutorial 8 Software Maintenance*, 5. Singapore: NTU.

[3] tutorialspoint. . "Struts 2 - Database Access." tutorialspoint. https://www.tutorialspoint.com/struts_2/struts_database_access.htm.