CSC584 Enterprise Programming

Chapter 7 – Development of Enterprise Application

MARSHIMA MOHD ROSLI

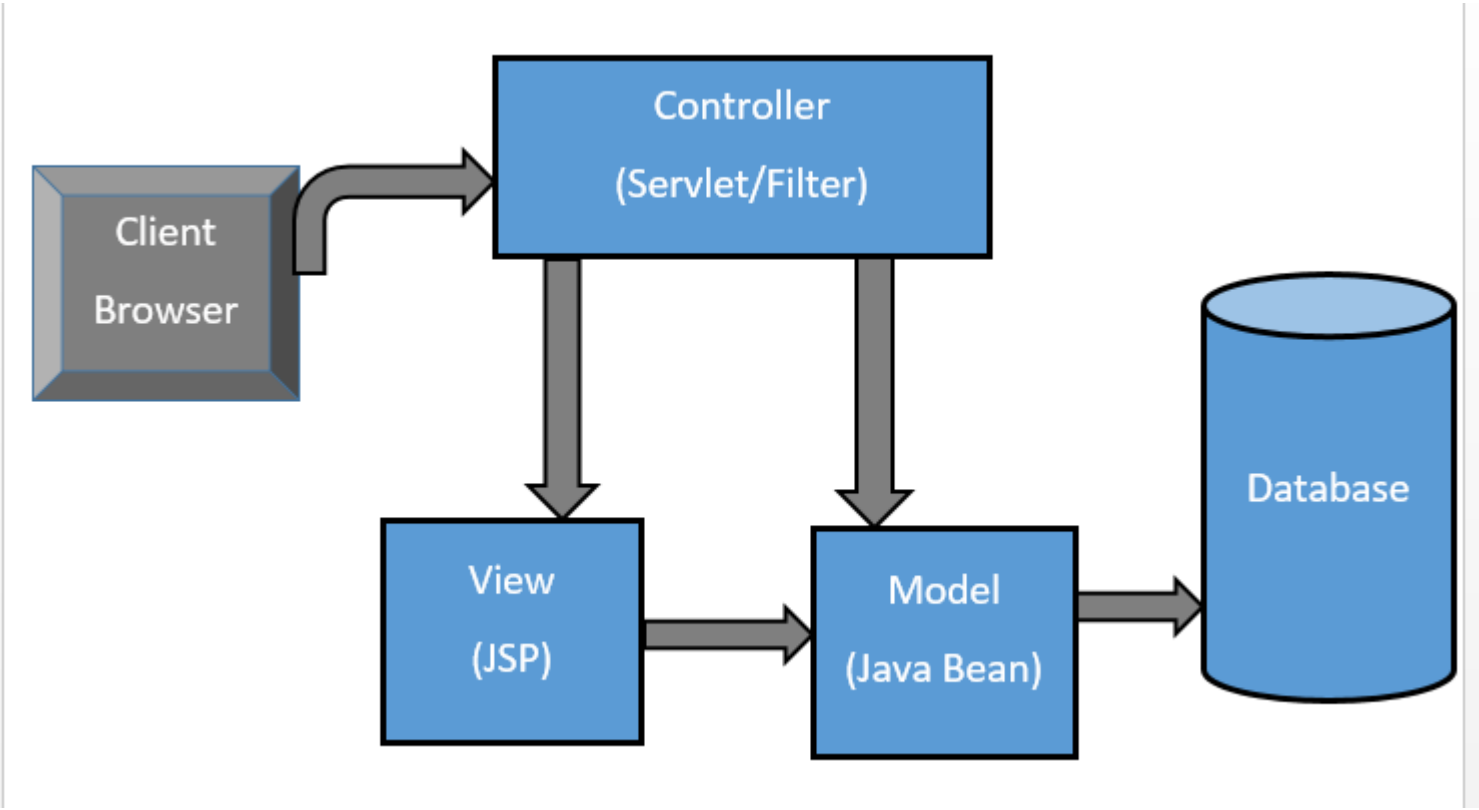
DEPT COMPUTER SCIENCE

Chapter 7 Outline

Development of Enterprise Application

- Choose the Java EE Architecture and Java EE pattern
- Design the Web components - HTML and JSP
- Develop Java Beans and Servlets
- Construct the JDBC connectivity with the enterprise application

Recap….MVC


Controller
(Servlet/Filter)
Client
Browser
View
(JSP)
Model
(Java Bean)
Database

MVC Architecture

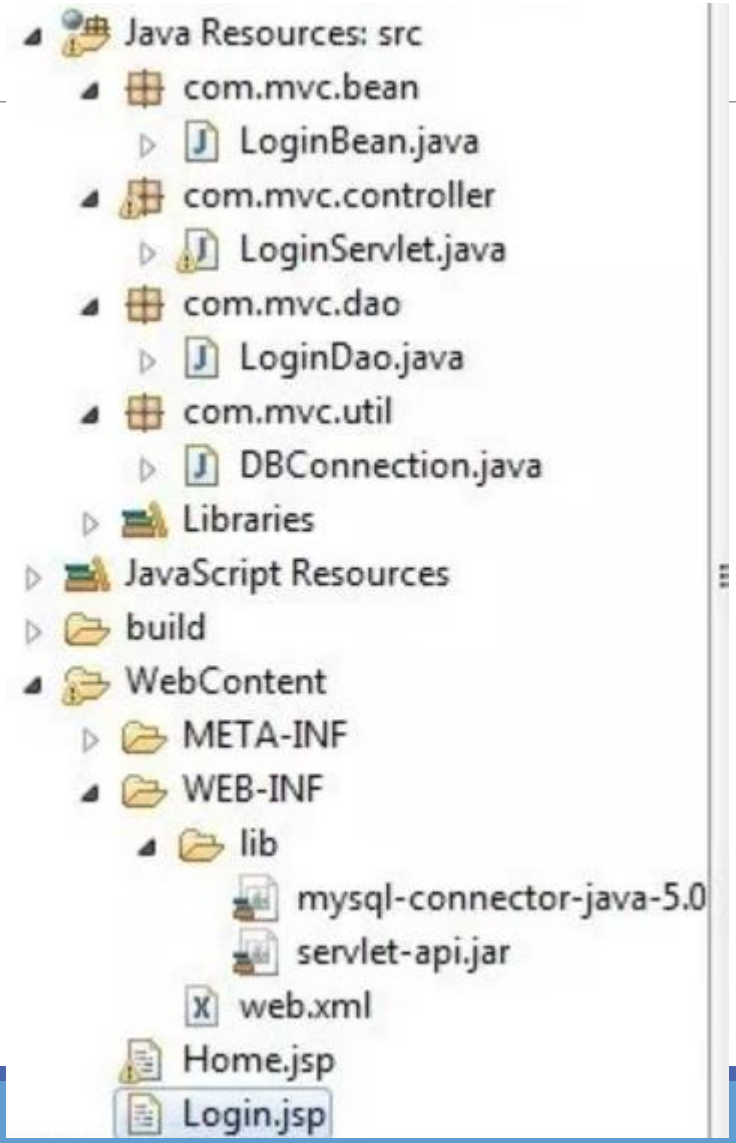
Model View Controller (MVC) is a software design architectural pattern for developing the web application. MVC pattern separates business logic, presentation, and data. MVC pattern has the following three parts —

Model – This level is responsible for all kinds of business logic operations. It has the classes which have the connection with the database.

View – This is the presentation layer which consists of HTML, JSP pages. This layer is responsible for displaying the application's output to the end users.

Controller – This layer handles the HTTP requests and sends to the appropriate **Model** layer for data processing, and once the data are processed and sent back to the controller and then displayed on the **View** layer.

MVC structure



Example MVC with Database

This is how the login page looks in the browser (except for the java logo)

Login JSP

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login</title>
<script>
function validate()
{
 var username = document.form.username.value;
 var password = document.form.password.value;

 if (username==null || username=="")
 {
 alert("Username cannot be blank");
 return false;
 }
 else if(password==null || password=="")
 {
 alert("Password cannot be blank");
 return false;
 }
}
</script>
</head>
<body>
<div style="text-align:center"><h1>Login application in Java using MVC and MySQL </h1> </div>
<br>
<form name="form" action="LoginServlet" method="post" onsubmit="return validate()">
<!-- Do not use table to format fields. As a good practice use CSS -->
<table align="center">
 <tr>
 <td>Username</td>
 <td><input type="text" name="username" /></td>
 </tr>
 <tr>
 <td>Password</td>
 <td><input type="password" name="password" /></td>
 </tr>
 <tr> <!-- refer to the video to understand request.getAttribute() -->
 <td><span style="color:red"><%=(request.getAttribute("errMessage") == null) ? ""
 : request.getAttribute("errMessage")%></span></td>
 </tr>
 <tr>
 <td></td>
 <td><input type="submit" value="Login"></input><input
 type="reset" value="Reset"></input></td>
```

LoginServlet

```java
//LoginServlet.java

package com.mvc.controller;
 import java.io.IOException;
 import javax.servlet.ServletException;
 import javax.servlet.http.HttpServlet;
 import javax.servlet.http.HttpServletRequest;
 import javax.servlet.http.HttpServletResponse;

import com.mvc.bean.LoginBean;
 import com.mvc.dao.LoginDao;

public class LoginServlet extends HttpServlet {

public LoginServlet() {
 }

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletExcept

//Here username and password are the names which I have given in the input box in Login.jsp page. He

String userName = request.getParameter("username");
 String password = request.getParameter("password");

LoginBean loginBean = new LoginBean(); //creating object for LoginBean class, which is a normal java

loginBean.setUserName(userName); //setting the username and password through the loginBean object th
 loginBean.setPassword(password);

LoginDao loginDao = new LoginDao(); //creating object for LoginDao. This class contains main logic o

String userValidate = loginDao.authenticateUser(loginBean); //Calling authenticateUser function

if(userValidate.equals("SUCCESS")) //If function returns success string then user will be rooted to
 {
 request.setAttribute("userName", userName); //with setAttribute() you can define a "key" and value
 request.getRequestDispatcher("/Home.jsp").forward(request, response);//RequestDispatcher is used to
 }
 else
 {
 request.setAttribute("errMessage", userValidate); //If authenticateUser() function returnsother tha
 request.getRequestDispatcher("/Login.jsp").forward(request, response);//forwarding the request
 }
 }

}
```

LoginDao

```java
package com.mvc.dao;

import java.sql.Connection;
 import java.sql.ResultSet;
 import java.sql.SQLException;
 import java.sql.Statement;
 import com.mvc.bean.LoginBean;
 import com.mvc.util.DBConnection;
 public class LoginDao {
 public String authenticateUser(LoginBean loginBean)
 {

String userName = loginBean.getUserName(); //Keeping user entered values in temporary v
 String password = loginBean.getPassword();

Connection con = null;
 Statement statement = null;
 ResultSet resultSet = null;

String userNameDB = "";
 String passwordDB = "";

try
 {
 con = DBConnection.createConnection(); //establishing connection
 statement = con.createStatement(); //Statement is used to write queries. Read more abo
 resultSet = statement.executeQuery("select userName,password from users"); //Here tabl

while(resultSet.next()) // Until next row is present otherwise it return false
 {
  userNameDB = resultSet.getString("userName"); //fetch the values present in database
  passwordDB = resultSet.getString("password");

    if(userName.equals(userNameDB) && password.equals(passwordDB))
    {
       return "SUCCESS"; ////If the user entered values are already present in database,
    }
 }
 catch(SQLException e)
 {
 e.printStackTrace();
 }
 return "Invalid user credentials"; // Just returning appropriate message otherwise
 }
 }
```

LoginBean

```java
//LoginBean.java

package com.mvc.bean;

//As I have already told it contains only setters and getters

public class LoginBean
 {
 private String userName;
 private String password;

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;
 }
 }
```

DBConnection

UNIVERSITI TEKNOLOGI MARA

```
1  //DBConnection.java
2
3  package com.mvc.util;
4
5  import java.sql.Connection;
6  import java.sql.DriverManager;
7
8  public class DBConnection {
9   public static Connection createConnection()
10  {
11  Connection con = null;
12  String url = "jdbc:mysql://localhost:3306/customers"; //MySQL URL and followed by the database name
13  String username = "root"; //MySQL username
14  String password = "root123"; //MySQL password
15
16  try
17  {
18  try
19  {
20  Class.forName("com.mysql.jdbc.Driver"); //loading mysql driver
21  }
22  catch (ClassNotFoundException e)
23  {
24  e.printStackTrace();
25  }
26  con = DriverManager.getConnection(url, username, password); //attempting to connect to MySQL databas
27  System.out.println("Printing connection object "+con);
28  }
29  catch (Exception e)
30  {
31  e.printStackTrace();
32  }
33  return con;
34  }
35 }
36 [/java]
```

Web.xml

UNIVERSITI TEKNOLOGI MARA

```
1  //web.xml
2
3  //this file is known as deployment descriptor. It contains the servlet and other configuraion details
4  <?xml version="1.0" encoding="UTF-8"?>
5  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/java
6   <display-name>LoginMvc</display-name>
7   <welcome-file-list>
8   <welcome-file>Login.jsp</welcome-file>
9   </welcome-file-list>
10  <servlet>
11  <description></description>
12  <display-name>LoginServlet</display-name>
13  <servlet-name>LoginServlet</servlet-name>
14  <servlet-class>com.mvc.controller.LoginServlet</servlet-class>
15  </servlet>
16  <servlet-mapping>
17  <servlet-name>LoginServlet</servlet-name>
18  <url-pattern>/LoginServlet</url-pattern>
19  </servlet-mapping>
20  <servlet>
21  <description></description>
22  <display-name>LogoutServlet</display-name>
23  <servlet-name>LogoutServlet</servlet-name>
24  <servlet-class>com.mvc.controller.LogoutServlet</servlet-class>
25  </servlet>
26  <servlet-mapping>
27  <servlet-name>LogoutServlet</servlet-name>
28  <url-pattern>/LogoutServlet</url-pattern>
29  </servlet-mapping>
30 </web-app>
```

LogoutServlet

UNIVERSITI TEKNOLOGI MARA

```java
//LogoutServlet.java

package com.mvc.controller;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class LogoutServlet extends HttpServlet
{
 private static final long serialVersionUID = 1L;

 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletExcept
 {
 HttpSession session = request.getSession(false); //Fetch session object

if(session!=null) //If session is not null
 {
 session.invalidate(); //removes all session attributes bound to the session
 request.setAttribute("errMessage", "You have logged out successfully");
 RequestDispatcher requestDispatcher = request.getRequestDispatcher("/Login.jsp");
 requestDispatcher.forward(request, response);
 System.out.println("Logged out");
 }
 }
}
```

Exercise PP: JSP and Database

1. Create a JSP page called "bookshop.jsp" which contains a form with 3 checkboxes for three different authors.
2. Write a JSP scriptlet that performs the database query operation. The steps are:
   - Establish a database connection via a `java.sql.Connection` object;
   - Allocate a `java.sql.Statement` object under the `Connection`;
   - Prepare a SQL SELECT string;
   - Execute the SQL SELECT using `executeQuery()` method. The result of query is returned in an object of `java.sql.ResultSet`;
   - Process the `ResultSet` row by row via `ResultSet.next()`;
   - Free resources and close the `Connection`.

Database

Database: **ebookshop**
Table: **books**

id (INT)	title (VARCHAR(50))	author (VARCHAR(50))	price (FLOAT)	qty (INT)
1001	Java for dummies	Tan Ah Teck	11.11	11
1002	More Java for dummies	Tan Ah Teck	22.22	22
1003	More Java for more dummies	Mohammad Ali	33.33	33
1004	A Cup of Java	Kumar	44.44	44
1005	A Teaspoon of Java	Kevin Jones	55.55	55

Web Project Discussions

- Web project
  - Presentation - progress
  - Project Submission =Week 14
    - Presentation
    - User manual
    - Project (netbeans project in a zip file)
  - Test 2 = Week 14 (Lab)