DT228/3 Software Engineering III Lab 5 (Week 8)

Eclipse Setup for a Dynamic Web Project

In this lab, we will setup an environment for developing a web application that will implement the design patterns and their collaboration as discussed in class. This lab is concerned with setting up the database and web project using Eclipse and installing the initial codebase that we will build on in next week's lab.

NB: Check the name of any files downloaded from webcourses – remove any bracketed numbers in the file names if you see them e.g. Command*(1)*.java -> Command.java.

1 Set up a Folder Structure

- On your personal drive (or laptop), create a folder for your Eclipse lab work (e.g. *SE3**Java*).
- Within that folder create a subfolder called *Libs* to hold any libraries (jar files) that we need (e.g. *SE3**Java**Libs*).
- Create another subfolder under *SE3**Java* called *Sql* to hold any sql scripts that we need (e.g. *SE3**Java**Sql*).
- Create another sub folder under *SE3**Java* called *Workspace* to hold your Eclipse projects (e.g. *SE3**Java**Workspace*).

2 MySQL Database & JDBC driver jar file

- As previously, we will be using the MySQL database that is distributed with the *xampp* software distribution package on the lab machines.
- Download the *mysql-connector-java-5.1.xx-bin.jar* file from the *Lab Materials* folder on webcourses (or retrieve it from previous lab) and place it in your *Libs* folder (created in step 1 above).

3 Using Eclipse to Set Up your Sample Database

- Start Eclipse (ensure it is the Java **Enterprise** Edition). Choose the workspace folder you created earlier when starting Eclipse (e.g. SE3\Java\Workspace).
- Close the welcome tab if that is what you see.
- Ensure you still have your MySQL connection configured (use the *Database Development Perspective* as before). If your connection is not configured then recreate it as per the previous lab, it should only take a few minutes.

Note: Remember we are using Eclipse for two distinct purposes. We are using it as a *database development tool* which is very much separated from what we will do next which is to *develop a Dynamic Web Application*.

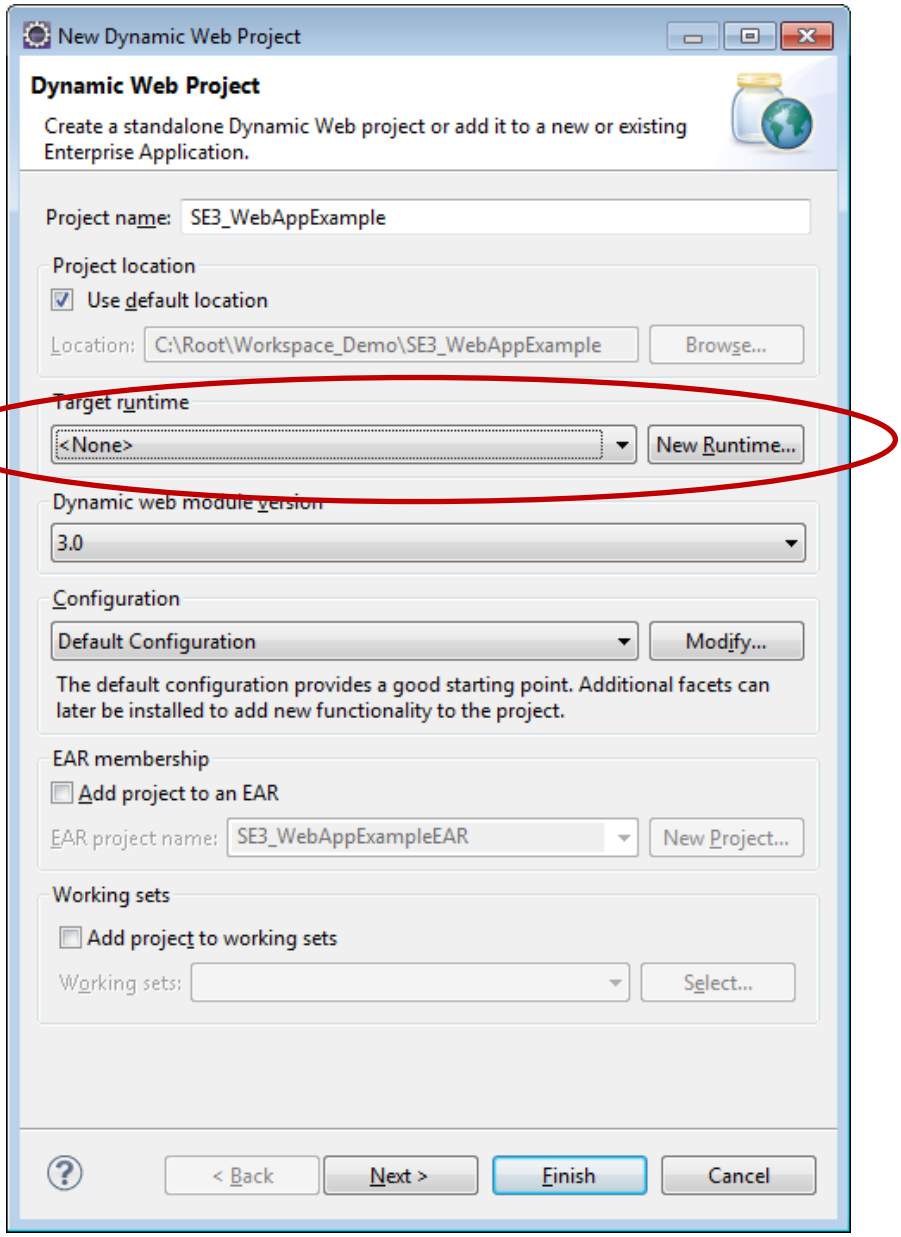
Part II – Creating a Dynamic Web Project

Our web application will use *Java Enterprise Edition (JEE)* technology (JSP, Servlets and plain Java). This means we need a *JEE container* to run on our server. *Tomcat* is both a web server and *JEE servlet container*. The Xampp distribution contains Tomcat and we can configure and use this within Eclipse.

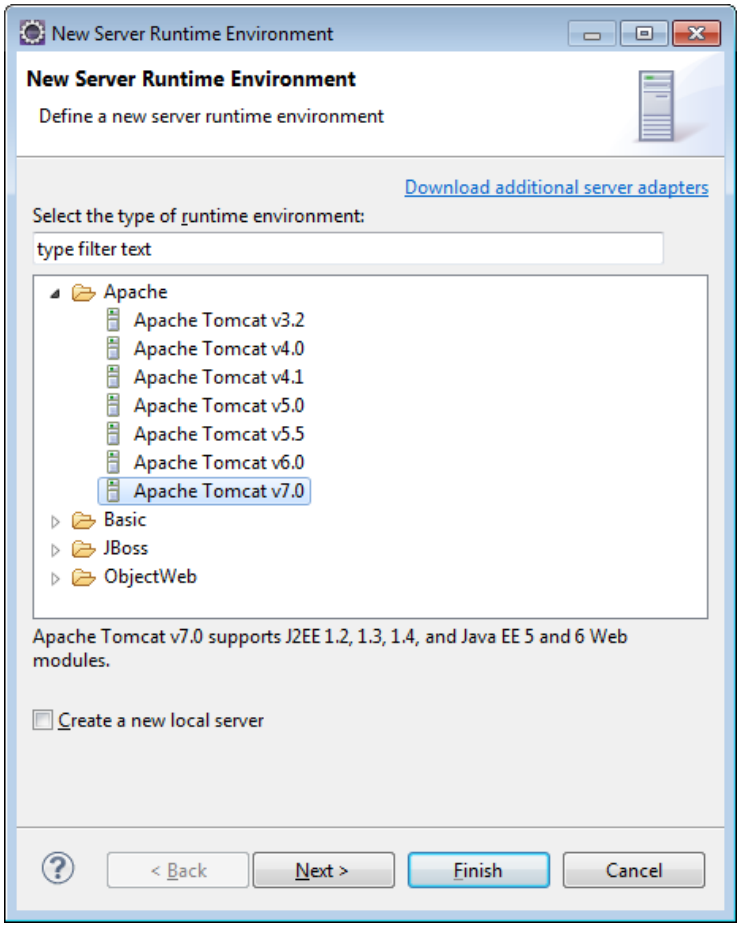
1 Create the project

- In Eclipse, switch to the Java EE perspective.
- Create a new ***Dynamic Web Project*** in Eclipse called *SE3_WebAppExample (File->New-> Dynamic Web Project)*. You should see the following screen:

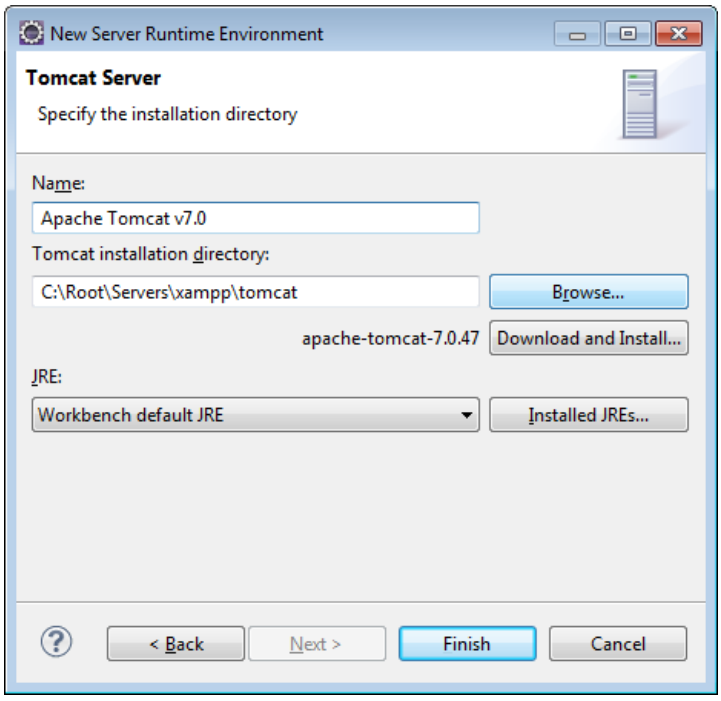
Note: Ensure it is a Dynamic Web Project and not a Java Project.



- If there is no "Apache Tomcat" *Target Runtime,* choose *New Runtime…* you should see the following:



- Select *Apache Tomcat v7.0* and click *Next>*. You should see the following:

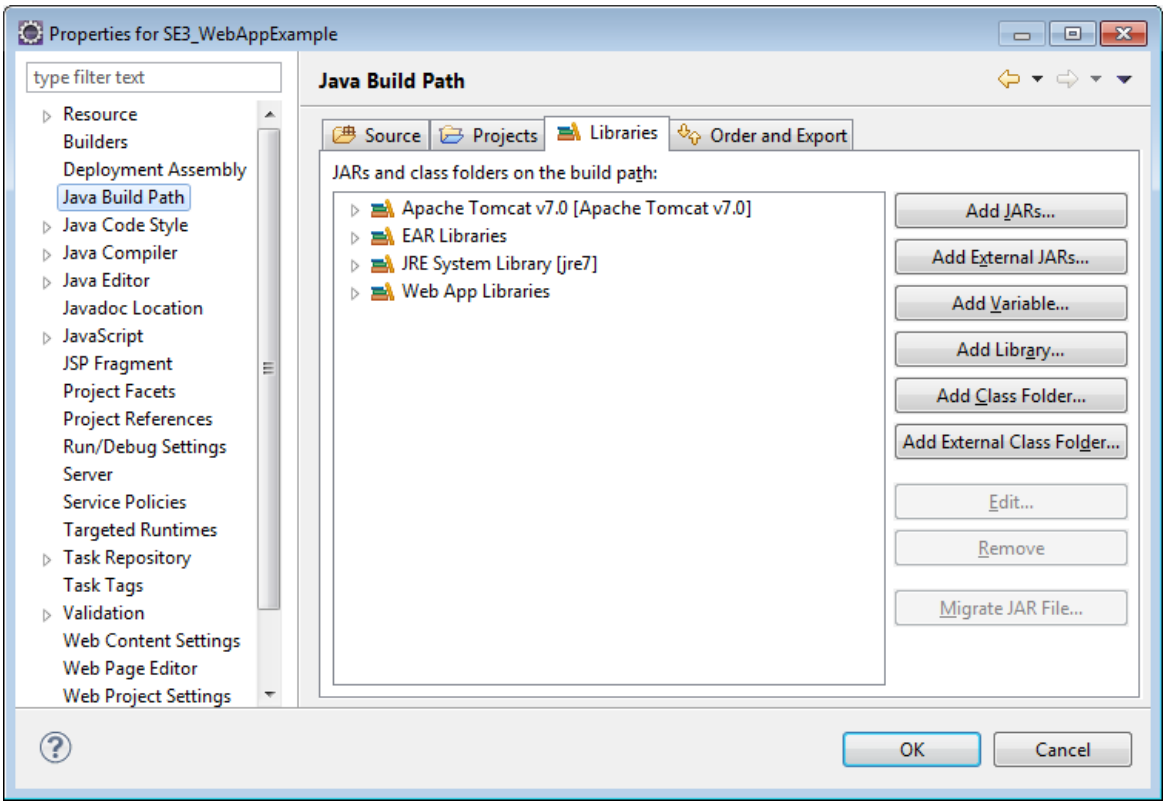


- Using the *Browse* button, navigate to the *tomcat* folder under the xampp distribution (probably *d:\xampp\tomcat* on the lab machines*)*.
- Click *Finish* to return to the previous screen and then click *Finish* again to create the project.

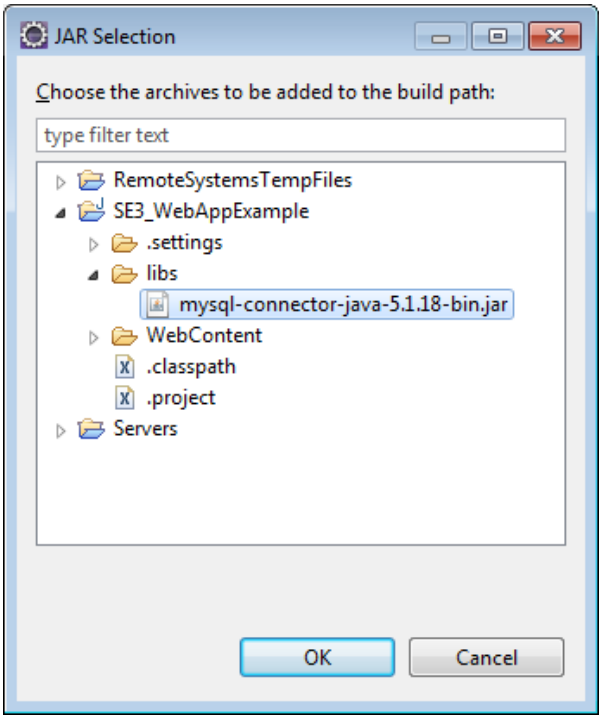
2 Add the MySQL JDBC driver to your project Build Path as follows

Note: Here we are adding the MySQL JDBC implementation jar file to our project classpath so that our Java code can utilise it to access the MySQL database.

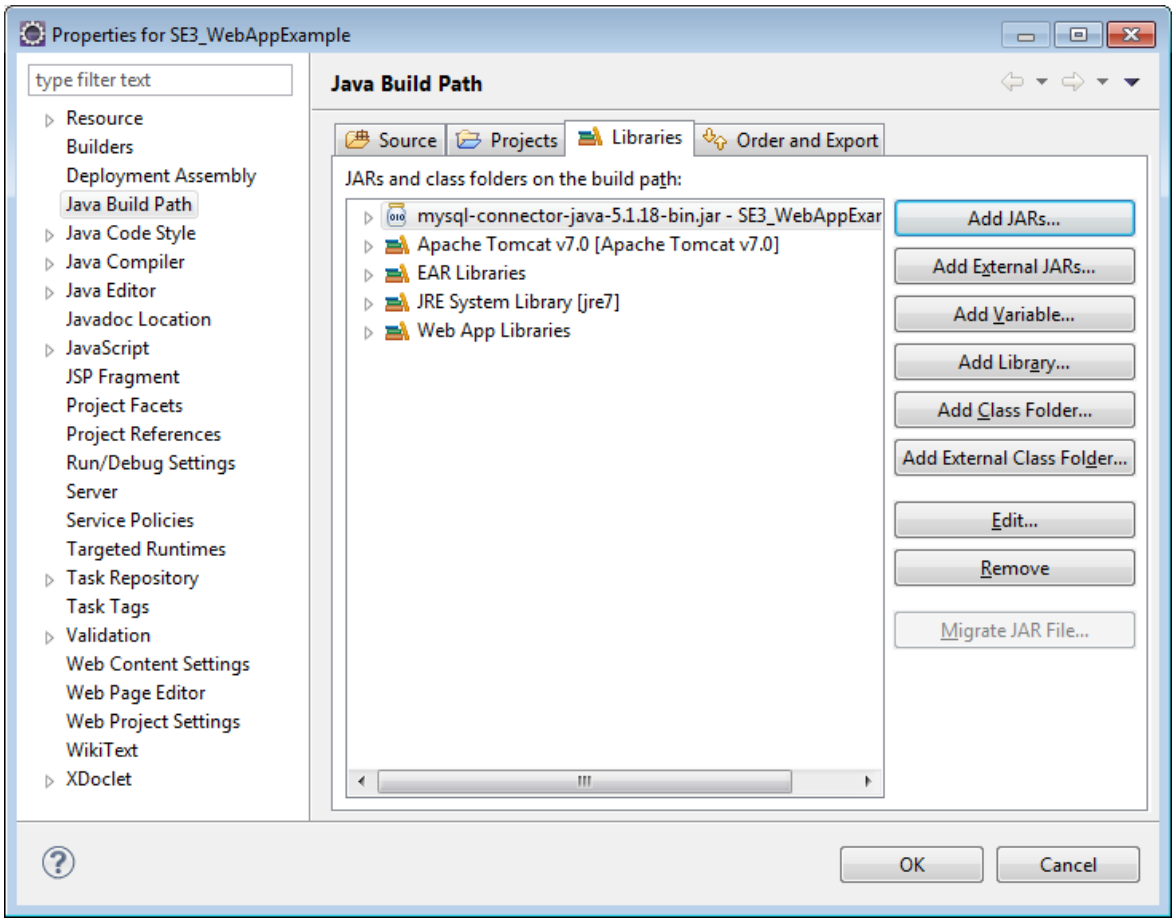
- Add a folder to your project called *libs* (right-click on the project in the Project Explorer in Eclipse and choose *New -> Folder*).
- Copy the MySQL JDBC jar file that you downloaded earlier to the new project folder (you can drag the file from windows file explorer into the Eclipse project explorer).
- Click on the project in Eclipse and from the menu bar go to *Project -> Properties* and select *Java Build Path* on the left. Select the *Libraries* tab on the right – you should see something like the screenshot below.



- Click *Add JARs…* and navigate to the MySQL JDBC jar file as below:

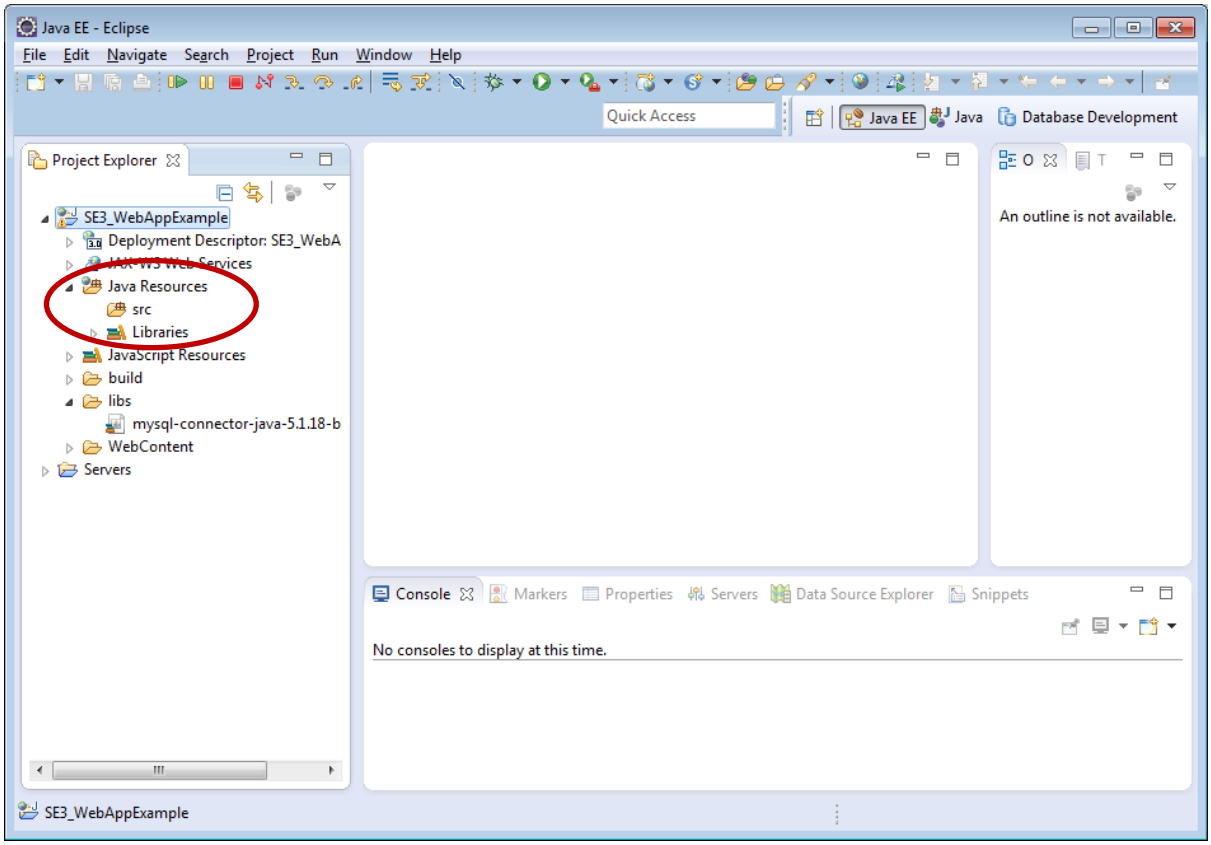


- Choose the jar file and click OK.
- You should now see the jar file on the right under the *Libraries* tab as below.
- Click *OK* to finish.

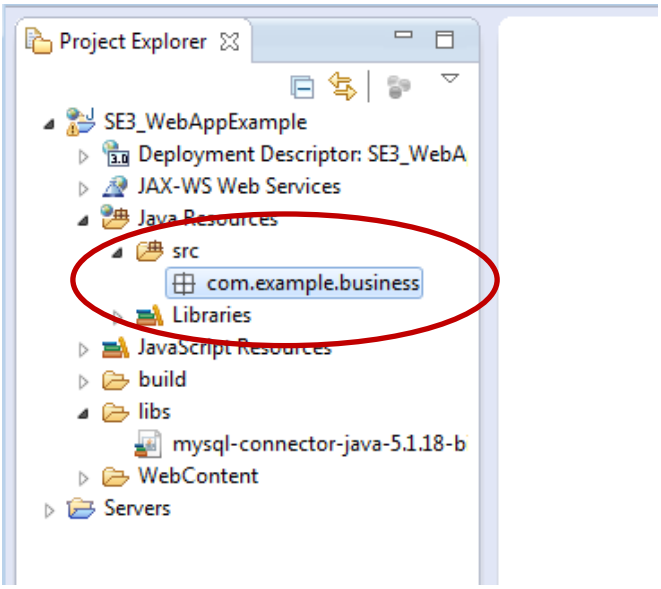


3 Add a Package Structure

- Expand the Java Resources folder in the Eclipse Project Explorer so that you see the *src* folder that was automatically created – this will hold our java source code:



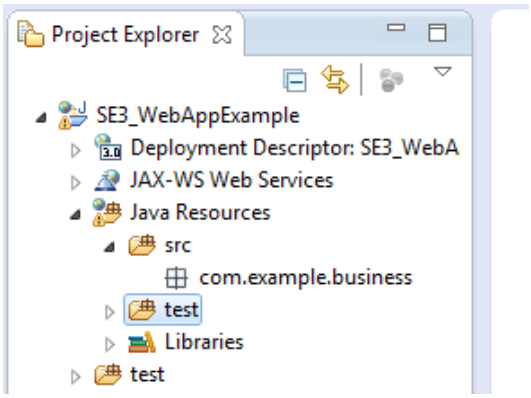
- Right-click on the *src* folder and choose *New -> Package*. Name it *com.example.business*.
- You should see the following:



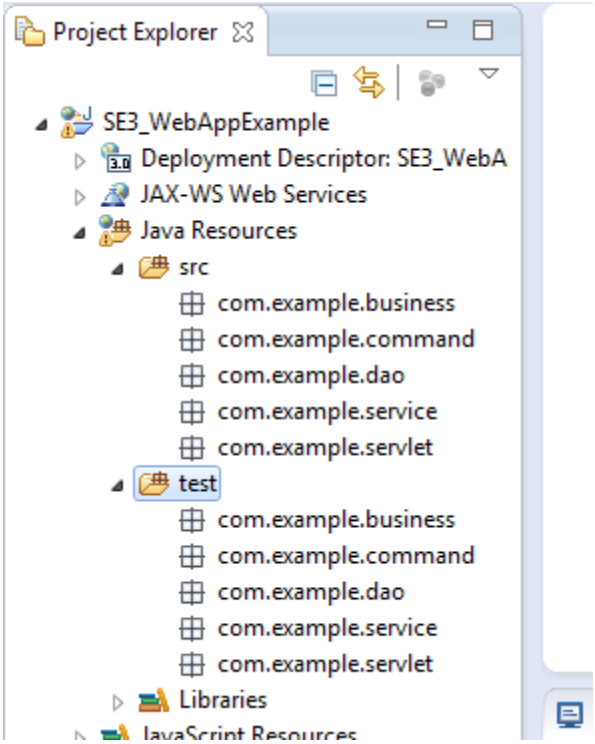
- Also create the following packages:

- com.example.dao
- com.example.service
- com.example.command
- com.example.servlet

- Create another source folder alongside our current *src* folder (right-click on the project in the Project Explorer and choose *New -> Source Folder)* and name it *test* – you should now have the following:



- Replicate the package structure that you created in the *src* folder so you have the following:



4 Test Database Connection

Note: Here we will just execute some java code to connect to the database and retrieve some data – this is not part of our application, just a checkpoint to see our project is set up correctly.

- Download the *JDBCExample.java* source file from webcourses.
- Copy it into the root of the *test* source folder (this will create a default package).
- Open it in the editor and right-click -> Run As -> Java Application.
- Have a look at the output in the console view.

PART III - Download the Initial Codebase

1 The DAO Layer

- Download the *Dao* and *UserDao* source files (.java files) from the webcourses lab materials folder – copy them to the *com.example.dao* package in the *src* folder of your dynamic web project.
- Create a new package called *com.example.exceptions* in your *src* folder.
- Download the *DaoException* source file from the webcourses lab materials folder – copy it to the *com.example.exceptions* package in the *src* folder of your dynamic web project.

2 The Service Layer

- Download the *UserService* source file from the webcourses lab materials folder – copy it to the *com.example.service* package in the *src* folder of your dynamic web project.

3 The Command Layer

- Download the *CommandFactory, Command* and *LoginUserCommand* source files from the webcourses lab materials folder – copy them to the *com.example.command* package in the *src* folder of your dynamic web project.
- Download the *CommandCreationException* source file from the webcourses lab materials folder – copy it to the *com.example.exceptions* package in the *src* folder of your dynamic web project.

4 Domain Objects

- Download the *User* source file from the webcourses lab materials folder – copy it to the *com.example.business* package in the *src* folder of your dynamic web project.

5 FrontController

- Download the *FrontController* source file from the webcourses lab materials folder – copy it to the *com.example.servlet* package in the *src* folder of your dynamic web project.

6 The View Layer

- Download the *login.html* source file from the webcourses lab materials folder – copy it to the *WebContent* folder in your dynamic web project.
- Download the *loginSuccess, loginFailure and errorPage* jsp files from the webcourses lab materials folder – copy them to the *WebContent* folder in your dynamic web project.
- You may see some error reporting in the JSP files initially, ignore these for now.

7 Some Additional Java Libraries

- Download the following java libraries from the lab materials folder on webcourses – copy them into the *libs* folder in your project. These libraries allow us to use special tags inside our JSPs that provide certain code execution that we would otherwise have to write java code for:

  jstl-api-1.2.jar
  jstl-impl-1.2.jar

- As this is a web application that runs inside the Tomcat servlet container, any libraries that we use in our project need to be deployed to Tomcat. With this in mind, copy the above two jar files and **also the mysql jar file** to the following folder:

  <<xampp_folder>>\tomcat\lib

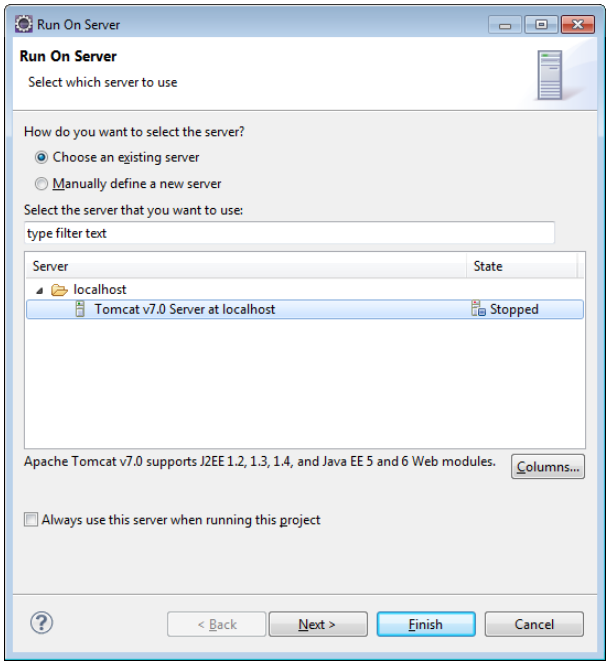
Part IV – Testing the Application

1 Set Up the Database

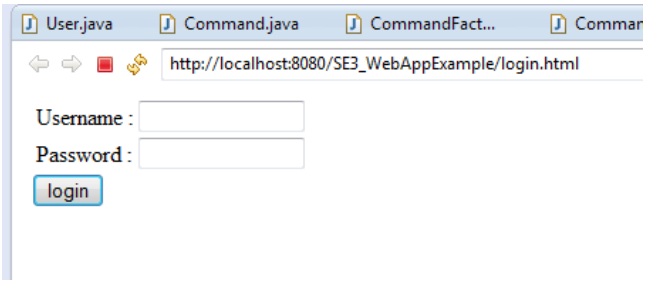
- Add a *USER* table to the database – download the *user_table.sql* from the webcourses lab material folder – copy it to your sql folder in your project.
- Ensure the MySQL rdbms is running (use the Command prompt or Xampp control panel).
- In Eclipse, connect to your test database using the *Database Development* perspective.
- Execute the *user_table.sql (as done in the previous lab)* – this should create a USER table and populate it with some sample data. Ensure the table has been created and the data inserted.

2 Run the Web App

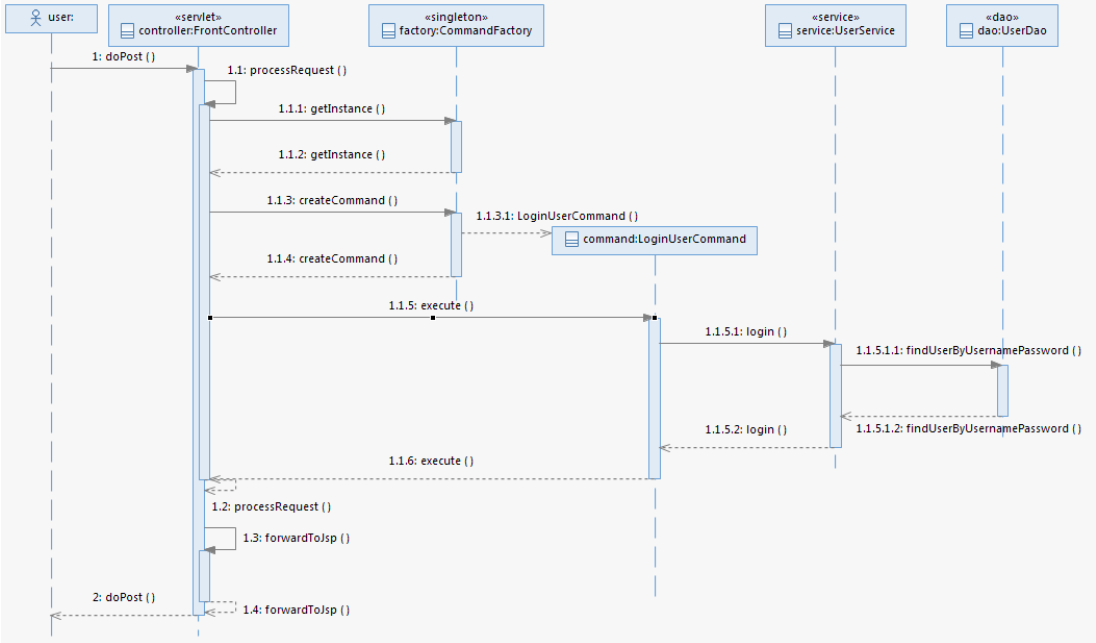
- Right-click on the *login.html* file in the Project Explorer and choose *Run As -> Run on Server.* You should see the following:



- Choose *Finish*.
- Eclipse should start tomcat and open a new browser tab and display the login page:



- Test out the application…
- Recall our web app design:



- Study the source code to see how our design has been implemented.