

PROG 32758
Enterprise Java Development
ASSIGNMENT #3
Posted: Nov. 17, 2019
Due: Dec. 1, 2019
Weight: 10%

Assignment 3: Student Grades Database

Description

You are to develop a web application for managing lists of student marks for this course. Using MVC platform, this web app. must be coded and implemented entirely using JSP, Expression Language of JSP files, POJOs (Plain Old Java Objects) and HTML files (along with CSS and possibly JavaScript). No plain Servlets!

Accordingly, the expectation is that you will implement the JSPs pages using scriptlet, expression, declaration, and directive tags. You should also limit the use of the **out** implicit object – try to use expression tags embedded in HTML in most places rather than print or println statements.

The Data Layer

You should use **grades.sql** to build your database. It contains one table called “prog32758”. Each row has a student number, first name, last name, and list of marks (4 quizzes out of 20 – each has a actual weight of 2.5%, 3 Assignments out of 50 – each has an actual weight of 10%, a Midterm exam out of 100 and a final exam out of 100 – each has an actual weight of 30%). Any of the marks fields can be null.

All database operations should be done using a **DataAccess** class created by a listener and posted to ServletContext.

The Business Logic Layer

You should create a **Student** object capable of holding an entire database row. This object can be retrieved and then posted to the session for use by all the JSP files. It can also be used as the return value and/or input parameter to DataAccess methods like insertRow, modifyRow and getRow. This object should also contain a method to compute the student’s final grade.

The Web Layer

The front door (**index.html**) should prompt for a student number and redirect to the **intro.jsp** file. The **intro.jsp** file should look up the Student in the database.

If the student does not exist, the user should get two links: **Create a new student** and **back**.

The Create a New Student link should go to a **create.jsp** file that allows the user to fill in a form to create the student record, using the student number they entered on the front door. The form should redirect to **another jsp** that performs the database insertion, issues a report stating exactly what was inserted and offers a link back to the front door.

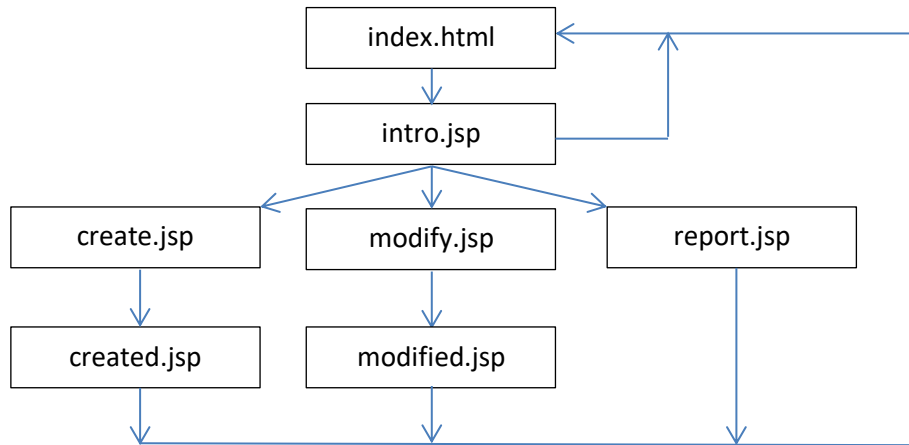
The back link should just go back to **index.html**.

If the student DOES exist, the user should get two different links: **Print a Report** and **Modify Grades**

The Modify Grades link should go to **modify.jsp**. This JSP should offer a form to change each of the grades in the database. Include the existing grades as default values in this form (if the grade is null, leave the default value blank). The form should redirect to another JSP that performs the update, issues a report on exactly what was updated and offers the user a link back to **index.html**.

The Print a Report link should go to **report.jsp**. This JSP should offer a summary of all the student's grades and then a final grade out of 100 along with a letter grade (http://gradecalc.info/ca/on/sheridan/gpa_calc.pl) Use the weighting given in the class plan to calculate the grade. If some of the student's grades are null, you should offer an estimate of their final grade. For example, if they have only done two quizzes, two assignments, and the midterm, you can average their two quizzes and two assignments to get those marks. Then you can calculate a mark out of 70 that does not include the midterm, then scale it up to be out of 100 to get an estimate of their mark so far. At the bottom of the report should be a link back to the front page.

The Flow of the App



Look and Feel

Make all pages have a consistent style **using images and CSS**. It doesn't have to be beautiful, but it should look better than the default styles. You can link all JSP and HTML files to a common style sheet to achieve this. Every page should also have a **common header and footer**. The headers and footers should be contained in separate JSP files and then included using a JSP directive tag.

Documentation

You must comment and indent according to the standards for the course. In a JSP, make a header comment with a description of the file, your name and date (if working with a partner, list the name of the person who was primarily responsible for this file first). Then comment any methods in declaration tags as you would if it were a plain Java file. CSS and HTML files should also have comment headers.

Working with a Partner

If you want to, you can work with a partner for this assignment. If you choose to do so, each of you must decide which of the objects you will each primarily be responsible for (you're splitting at least two POJOs, 1 HTML file, 1 CSS file, 6 full JSPs and 2 smaller JSPs or HTML files to define the headers and footers). Accordingly, each partner should have primary responsibility for their own set of objects and should have written those objects mostly themselves. The split of the work should be roughly 50/50. The two partners might not receive the same mark for the assignment. Additionally, you would need both to submit a short report indicating how you planned to work together and state exactly who did what.

It is strongly suggested that you finish the `DataAccess` and `Business Logic` objects first.

Good luck 😊