**~~SOAP 2 – Extending the Service~~**

~~In this lab we want to create a data access class to provide Get, GetAll and Delete operations. We want to avoid hardcoding Student details like in the previous lab.~~

**~~Step 1: Create a Student class~~**

~~Create a class for a Student, complete with constructor, getters and setters:~~

Text

Description automatically generated with low confidence

**~~Step 2: Create a Data Access class~~**

~~Next we want to create a Data Access class~~.



**Step 3: Edit StudentDetailsEndpoint so it uses the StudentDao**

Edit the ‘StudentDetailsEndpoint’ class so that it uses the new data access class, and then run the service and test by trying to access different students. (We had this part hardcoded in the previous lab).



~~Now, rerun the server and use Wizdler to test the the service by inputting different values for ID.~~

~~The result of asking for student with id=3:~~

Text

Description automatically generated

**Step 4: Edit the student-details.xsd**

~~To create a SOAP service to get~~ **~~all student details~~**~~, we first need to change the student-details.xsd to include sections for GetAllStudentDetailsRequest and GetAllStudentDetailsResponse.~~

~~Add the following to the student-details.xsd file:~~

Graphical user interface, text, application, email

Description automatically generated

~~Note that the GetAllStudentDetailsResponse has the addition of the maxOccurs=”unbounded” as in this case, all the student’s details will be returned.~~

~~You should see that new Java classes have been made for both the request and the response.~~

**~~Step 5: Edit the~~** **~~StudentDetailsEndpoint~~**

~~We need to add a method to ‘StudentDetailsEndpoint’ to accommodate the new request/response.~~

~~Add the following method to StudentDetailsEndpoint~~:

Graphical user interface, text, application

Description automatically generated

**~~Step 6: Run the application and test for GetAllStudentDetails.~~**

Graphical user interface, text, application

Description automatically generated

Result:

Text

Description automatically generated

**Exercise 1:**

Get the Delete operation working (Delete one student by id). Remember the steps

1. ~~Edit the student-details.xsd to include a request and response for the delete operation. The delete should return an integer variable called status, 1 for success, 0 for failure.~~
2. ~~Add a new method in the StudentDetailsEndpoint that uses the data access class to delete a student.~~
3. Test the service.

(Note: the operation should return a status of 1 for success and 0 for failure)

When finished, you should have all 3 operations working:

Graphical user interface, text, application, chat or text message

Description automatically generated

**Exercise 2:**

~~Using Wizdler, send a bogus id to GetStudentDetailsRequest to see the what a fault response looks like.~~

A screenshot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

~~Also, take time to note the structure of the SOAP response (Envelope, Header and Body)~~

**Exercise 3:**

~~Take time to go through the WSDL file and see if you can understand all the parts that are needed to make a SOAP service.~~

