# *Web Programming IV (420-C40-HR)*

# *Lab 12 – Using Layers for Data Access*

Date assigned: Thursday March 30, 2017

Date due: **Thursday, March 30, 2017, 4:00 p.m.**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Work with layers of abstraction in an application
* Perform updates using those layers

Lab Set-Up

1. Create a folder **YourUserName\_C40L12** in your H:\420-C40\Labs folder.
2. In this lab you can continue to work on the application from last week or you can create a new one. The choice is up to you..
3. If you use last week’s lab, use the Master page in the creation of all new pages. Make sure to design the pages nicely.

To do:

1. Create a blank Web Application project and add an App\_Code folder to your project. Note: All classes can go directly in this folder or you can create two subfolders, one for BLL and the other DB. I would NOT do this as I think this might be part of the referencing problem we are having at times.
2. Inside the App\_Code folder create a class called Faculty. The class has all the attributes from the IU\_FACULTY table; that is, facultyId, facName, roomId, phoneNum and deptId. You should also add attributes for deptName which is a string and roomLoc which is a string. Add gets and sets for all the attributes. Add a constructor which accepts all the IU\_faculty table fields.
3. Also create a class called Department which contains the attributes deptId and deptName matching the table iu\_Department.
4. Lastly create a class called Location which contains the attributes roomId (an int) and roomLoc (a string) matching the iu\_Location table.
5. Make sure that you include (using) System.ComponentModel and System.Data in all the classes.
6. In the App\_Code folder add a class called FacultyDB. In this class, add a method called selectAllFaculty which returns a DataSet containing all the facultyId and facName sorted by facName. (select facultyId, name from iu\_faculty order by name).
7. FacultyDB class add a method called selectFacultyById which is passed a facultyId and returns a dataset object of all the columns from the table matching that facultyid (DO NOT RETURN DEPARTMENT NAME OR ROOM LOCATION. Return only the fields from the iu\_faculty page. Note: this dataset should only have one row.
8. Create a DepartmentDB class which has two similar methods. The first returns a dataset of all the department id and department names. The second returns a STRING of the department name for the specified ID. Call the methods selectAllDept and selectDeptById respectively.
9. Create a LocationDB class which has two similar methods. The first method (selectAllLocation) returns all the location information in a dataset with TWO columns. The first column is the roomId and the second column is a combination of the building and roomNo in the format building-roomno (Ghandi-101), This combining of fields can be done in the select statement. The second method (selectLocationById) returns a STRING of the building-roomno for the specified roomId.
10. Back in the Faculty class, create a method called getFaculty which calls the method selectAllFaculty and receives the dataset. It then calls the **private** method convertFacList to convert the dataset into a list of faculty that only contains the facultyId and facName.
11. Also in the Faculty class create a method called getFacultyById which calls the selectFacultyById method in the DB class and receives a dataset of the matching faculty. It then calls a private method convertFac which returns a Faculty object (singular) matching the data in the dataset.
12. Add similar methods to the Department and Location classes to return all and a selected values. Note: there is no need for a conversion function when returning the string of a specific department name or room location.
13. Now use the System.ComponentModel to make the get methods of the Faculty, Location and Department classes to be Select methods [DataObjectMethod(DataObjectMethodType.Select, true)]. Do the same for the getAll methods, but make the last parameter false.
14. Add a page called faculty.aspx. Add an ObjectDataSource to the page and connect the select to the Faculty getAllFaculty method. Add a drop down list to the page which displays the Faculty Name and has the faculty ID as the data value.
15. Add two more object data sources and two more drop down lists to the page to make sure the department and location are being displayed.
16. Add another object data source to the page where the select statement uses the getFacultyById method. Add a detailsview or listview which uses this object data source.
17. ~~Add an event handler to the code behind such that, when the selection changes in the drop down list, the getFacultyById method is called and the details of the faculty member are displayed on the page (this will include the department number and room id).~~
18. ~~Add a databind to the onload or prerender method to make sure that the detailsview displays the first record in the dropdown when the page is initially loaded.~~
19. Back to the Faculty class. Add a method called getFacultyDetailExpanded which receives a faculty id and returns a faculty object with the department name (deptName) and room location (roomLoc) with the human readable values of the department and location. To do this do the following in the getFacultyDetailExpanded:
    1. First calls the getFacultyById method to get a Faculty object for the matching id.
    2. Uses the deptId from the faculty object to call the selectDepartmentById method. The returned value is assigned to the deptName attribute of the Faculty object.
    3. Uses the roomId from the faculty object to call the selectLocationById method. The returned value is assigned to the roomLoc attribute of the Faculty Object.
    4. Returns the Faculty object (with the string values).
20. Declare this method as a DataObjectMethod of type select as well.
21. Add another (5th) object data source to the page and have it use the getFacultyDetailExpanded method.
22. Add another listview or details view to the page which is connected to the above ODS.
23. Add code to the selection changed method for the drop down above to call this new getFaculty method and fill the new view. Format the detailsview so that the department ID and roomID are not displayed but the text values are.

BONUS: If you get this done, try adding an update feature to the second details view. This will require a new method in the FacultyDB class and Faculty class. In the update, make the department number and roomId both drop downs (use an Edit Template). In the Faculty class update method, grab the faculty ID and RoomID from the dropdowns and pass them to the update method in the database.

**To submit**

When you have completed the lab exercise, create a single zip file called YourUserName\_C40\_L12.zip. The zip file must contain all of the parts of the lab in the folder you created at the beginning of the lab. Copy the file to the Moodle page for the course.