SCHOOL OF
COMPUTER AND SECURITY SCIENCE

AUSTRALIA ECU EDITH COWAN UNIVERSITY

CSI2441 Applications Development – Assignment Two

Rapid Application Development of a Results Management System

(25 Marks Total)

In your first assignment you created a system to input and validate form data. In your second assignment you will expand on that concept by inputting, search, editing and deleting data via forms and a database system. Your system is to be built and delivered as an ASP.NET(Code behind) website using Visual Studio 2010 – 2018 as your development environment. Your database is to be Microsoft Access, MySQL or SQL Server (Express).

Task

The system you will build has a number of related modules which are listed below;

1. ***Login Module***
2. ***Manage Units***
3. ***Manage Results***
4. ***Manage Reports***

1)Login Module

This module will provide login functionality for the two predefined users, namely, Admin and Results Manager. Admin will be able to access *Manage Units* whereas Result manager will be able to access *Manage Results* and *Manage Reports* modules. Both users will be able to login into the system using username and password, can perform their required tasks and then logout from the system. You don't need to develop add/edit/delete functionality for this module, rather, the UserID, UserName, UserPassword and UserType will be stored directly in the database.

The main fields that describe a login module include;

1. UserID (1 & 2)
2. UserEmail (admin@ecu.com & manager@ecu.com)
3. UserPassword (Admin#1 & Manager#1)
4. UserType (0 for Admin, 1 for Event Manager)

Business Rules

- ✓ The only valid combination of username and password should allow access to the system.
- ✓ Once logged in admin should be able to access *Manage Units* module only whereas Results Manager should be able to access *Manage Results* and *Manage Reports* management modules only.

2)Manage Units:

This part of the application allows admin to add, edit and delete units from a list of units for which scores have been recorded. The three main fields that describe a unit include;

-Unit code
-Unit title
-Unit coordinator
-Unit_outline_pdf_document

For the sake of this assignment we will assume that all units have three assessments, assignment 1, assignment 2 and an exam, worth 20 20 and 60 respectively. In the real world you would define the number of assessments when defining a unit and what each assessment is worth. You are not required to do so in this assignment, but extra marks will be awarded if you can figure out how implement this function.

3)Manage Results:

In this part of the program Results Manager need to be able to add, edit and delete student results for a given unit in a given semester. Some suggested fields that would define the data required for the function include;

-Unit code
-Student id
-Student photo
-Semester
-Year
-Assessment 1 score
-Assessment 2 score
-Exam score

The last three fields are interesting in that if you add up the three assessment scores you should already have the Unit mark, hence why we have not included it in the database. In your assignment you will calculate and display the Unit mark ‘on the fly’ as the result is extracted from the database.

4)View Reports:

This section of the program will allow Results Manager to search for student results according to a number of different criteria;

-By student id
-By unit code
-By semester
-By year
-By a combination of these

Some of these functions will be performed using ‘browse’ options such as drop-down lists of unit codes/titles, or ‘show all’ options. As a part of the output not only does the assessment 1, 2 and exam scores and the unit score need to be shown, but also the Grade which is based on the unit score. For example, if a student has a unit score of 81 then the

word 'HD' should appear next to that score. The Grade is not stored in the database as it a fixed set of values that fit in a known range and can be displayed on screen dynamically according to the value stored for unit score. Reports should be sortable according to student id, unit score or unit code where applicable.

Wherever a set of results is shown on screen it should be accompanied by a 'count' of items on screen – such as 'showing xx results' and an average of the unit scores. For example, if I were to list all the results for csi2441 in 102 I might see the following;

CSI2441 Semester 2 2010

Student Id	Assign 1	Assign 2	Exam	Unit score	Grade	Student Photo	Unit outline
99573784	18	17	30	65	Cr		AP.pdf
88383882	9	20	26	55	Pass		
99393838	19	20	34	73	D		
29484828	20	20	50	90	HD		

Results: 4
Unit avg: 71 D

If I were to show the results of a search by a student id I might see something like the following;

Student ID 99573784

Unit code	Semester	Year	Assign 1	Assign 2	Exam	Unit score	Grade
CSI2441	2	2009	18	17	30	65	Cr
CSG2431	2	2008	10	10	40	60	Cr
CSP2103	1	2007	12	17	51	80	HD
CSI3302	2	2006	15	15	13	43	N

Results: 4
Course avg: 62 Cr

Obviously, if I go into Manage Results and make a change to a student's results, next time I go back into the Reports area their scores will have changed.

In amongst all this data adding and editing we need to validate our input, with the following considered a minimum;

-Assessment scores must be numeric and between 0 – 20
(or the value defined in the database if that is how it has been implemented)
-Exam scores must be numeric and between 0 – 60
(or the value defined in the database if that is how it has been implemented)
-Unit codes cannot be blank and must be 7 characters in length and be in the format aaa1111
(three characters four numbers)
-Unit title cannot be blank
-Student id must be numeric and 8 digits in length

-Year and semester must be numeric and of the correct length
-Validation should be applied for both entering a new record and also editing a record
-When the Delete button for any record is clicked, a prompt should pop up to ask the user 'Are you sure you wish to delete this record' or something similar

How you build your application from a logic and presentation perspective is up to you. It is recommended that you have a Main Menu page with links to each of the sub-sections listed above, each of which in turn may have their own sub-sections if necessary. As well as building a functional and validated data processing system, use the presentation and design features available in .Net to build an appealing, easy to navigate web application.

Test your site thoroughly to make sure that all database connections and data objects work correctly, especially across different machines uses different locations (desktop and my documents for instance) to ensure that no fixed-path issues occur.

Assignment Submission

When you have completed your assignment, you need to load ALL your files into Blackboard on or before the submission date. It is best to zip your site (with your student id and unit code as part of the filename) and then submit that file into Blackboard. Remember, all files will be checked against all others, so do not share your work with others or use other student work as this can lead to issues of collusion and loss of marks.

CSI2441 Assignment 2: Marking Key

Student ID: **Student Name:**

FUNCTIONS

Criteria	Mark Weighting
1) Manage Units (add, edit, delete)	**/ 4**
2) Manage Results (add, edit, delete, correct score management)	**/ 7**
3) View Reports (search and browse options, automatic grade calculation)	**/ 7**
4) Data Validation (validation on blanks, correct scores, data types)	**/ 4**
5) Advanced or Extra Functions (dynamic score management, file upload function for marks spreadsheet, event logging in database, MVC, Entity Framework,etc)	**/ 3**

Overall Summary

Total Mark / 25 **Assignment Grade _____**