

# CSI2441 Applications Development – Assignment 2 Brief

## Contents

IMPORTANT NOTES: .....	1
Assignment Brief - Rapid Application Development using Asp.Net Code Behind for a Sports Management System (40 Marks) .....	3
Solution Overview .....	3
1 Login Module .....	3
2 Games Management Module (add, edit and delete) .....	4
3 Competitors Management Module (add, edit and delete) .....	4
4 Events Management Module (add, update) .....	5
5 Reports Management Module (search and display) .....	5
Assignment Development .....	6
General Requirements: .....	6
Marking Key .....	6
Submission Requirements .....	8

## IMPORTANT NOTES:

---

- This is an **individual** assignment only and must **not** be completed in collaboration with other students
- You may **not** work with others, acquire code from others, or provide code to others
- Further, you may **not** post any of the assignment tasks below to a code-development community of any kind seeking solutions or advice
- Nor may you copy and paste or otherwise reproduce code provided by external sources and use it as part of your own solutions
- Where it is found that any of these restrictions have been ignored, academic misconduct proceedings may be initiated
- Please read the checklist below and watch the associated video **BEFORE** submitting your assignment

## ACADEMIC INTEGRITY TICK-BEFORE-SUBMIT CHECKLIST

### PLAGIARISM

- ✓ I have not copy and pasted from external sources without appropriate citation
- ✓ My in-text and end-text citations follow APA 7 guidelines
- ✓ I have not used my own or other student's previous assignment work



### COLLUSION

- ✓ I have not worked with any other students on this assignment unless permitted
- ✓ My assignment is not based on or derived from the work of any other students
- ✓ I have not shown or provided other student(s) with my assignment at any point



### CONTRACT CHEATING

- ✓ I have not asked or paid someone to do this assignment for me
- ✓ I have not used any content from a "study notes" or "tutoring" service / website
- ✓ I have not had a friend or family member assist me with this assignment



**IF YOU ARE UNSURE ABOUT ANY OF THE ABOVE, DO NOT SUBMIT YOUR ASSIGNMENT BEFORE SPEAKING WITH YOUR UNIT COORDINATOR OR ECU LEARNING ADVISOR**

[Watch this video before submitting your assignment](#)

## Assignment Brief - Rapid Application Development using Asp.Net Code Behind for a Sports Management System (40 Marks)

In your first assignment, you created a system to input and validate form data for an Olympic scores system for a single Olympic Games. In your second assignment, you will expand on that concept by developing an application that allows authorised users to manipulate data in a backend database system in four (4) ways – Insert, Update, Delete and Search (Read). Your system must be built and delivered as a .NET/C# website using Visual Studio Enterprise 2019 as the development platform. Use a SQL Server instance as the backend database.

*It is strongly recommended that you create the SQL Server database as an object **within** your project's structure/tree. This will make submitting your assignment much simpler.*

### Solution Overview

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

1. Login Module
2. Games Management
3. Competitors Management
4. Events Management
5. Reports management

#### 1 Login Module

---

This module will provide login functionality for two (2) predefined roles (users) - **Administrator** and **Events Manager**

The *Administrator* role must be enabled with access to **all** management modules

The *Events Manager* role must **only** be enabled with access to the *Events* and *Reports* management modules

Both roles (users) must login into the system using username (email address) and password before they can access the pages/management modules they are authorised to use, perform the tasks they are authorised to perform, and then logout from the system

You do not need to develop add/edit/delete functionality for the Login module, rather, just prepopulate the relevant database table with each *UserID*, *UserName*, *UserPassword* and *UserType* provided in the table specification below.

TABLE FIELD	Allowable Values
<b>UserID</b>	1,2
<b>UserEmail</b>	admin@ecu.com event@ecu.com
<b>UserPassword</b>	Admin#1 Event#1
<b>UserType</b>	0 (Admin) 1 (Event Manager)

### Business Rule(s) for Login Module:

- Only a valid combination of username and password should allow access to the system
- Once logged in, the Administrator should be able to access **all** modules, whereas the Event Manager should be able to access the *Events* and *Reports* modules only

## 2 Games Management Module (add, edit and delete)

---

This part of the application allow an authorised role (user) to insert, update and delete games information in the backend database.

Be sure to populate the relevant database table(s) with at least **three (3)** records each for the **three (3)** games *running*, *cycling* and *swimming*.

The *base* specification for this table is as follows:

Game\_ID  
Game\_Code  
Game\_Name  
Game\_DurationInHours  
Game\_Description

### Business Rules for Games Management Module:

- There can be only **one (1)** instance of a game in the system
- The Game\_Code must be **seven (7)** characters in length and a mixture of **four (4)** *uppercase characters* and **three (3)** *numerals*
- Only *Game\_DurationInHours* and *Game\_Description* may be **NULL**

## 3 Competitors Management Module (add, edit and delete)

---

This part of the application allows the authorised role (user) to *add*, *edit* and *delete* competitor information contained within the backend database. You need to create records for at least **25** competitors.

The *base* specification for this table is as follows:

Competitor\_ID  
Competitor\_FName  
Competitor\_Surname  
Competitor\_DOB  
Competitor\_Country  
Competitor\_Bio (*100 words max*)

Be aware that many competitors can participate in one or more games (a many-to-many relationship), so an additional table will be required to resolve this and ensure only one-to-many relationships exist with the database. The base specification for this additional table is as follows:

Game\_ID  
Competitor\_ID

### Business Rules for Competitors Management Module:

- Only *Competitor\_Bio* may be **NULL**
- Ensure that each competitor in the *Competitor* table has participated in at least **one (1)** game

## 4 Events Management Module (add, update)

---

This part of the application allows you to add, update and delete Event information.

The *base* specification for this table is as follows:

Event\_ID  
Game\_ID  
FeatureEvent  
EventVenue  
EventDate  
EventStartTime  
EventEndTime  
EventDescription (*100 words max*)  
WorldRecord

This module will also require an additional table to resolve the above mentioned many-to-many relationship issue. The base specification of this additional table is as follows:

Event\_ID  
Competitor\_ID  
Competitor\_Position (1st,2nd,3rd,4th .... )  
Competitor\_Medal (Gold, Silver, Bronze – otherwise No)

### Business Rules for Events Management Module:

- There can only be one instance of an event in the Event table
- No fields in the Event table may be **NULL**

## 5 Reports Management Module (search and display)

---

Once the various sections of the application are developed that allow adding of competitors, games, and events information, build a reporting module that allow authorised roles (users) to:

- Display the details of a selected event
- List a selected country's Gold, Silver and Bronze medals tally
- Display Gold, Silver and Bronze medals tally for all countries in a ladder-style format
- Show world record(s) achieved by Game (if any)
- Show world record(s) achieved by Country (if any)
- Show all world records achieved overall, including Game, Competitor and Country details

## Assignment Development

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, try to use the presentation and design features available in .Net to build an appealing, easy to navigate web application.

As there is quite a lot to be done in this project, it is suggested that you take advantage of features Visual Studio and .NET provide to build its major elements quickly and avoid coding everything manually. Manual coding should mainly be used when you are connecting modules together or tweaking a component's specific functionality. From a marking perspective however, you will be graded on your application's functionality outcomes rather than the development methodology you employed to build it.

Test your application thoroughly to make sure that it is fully self-contained and portable, and that it will work as expected when your lecturer/tutor downloads the project files submitted to their own Visual Studio Enterprise 2019 environment. With this in mind, it is strongly recommended that you create the required SQL Server database instance as an object within your project's structure/tree.

Assignments are marked as-is and debugging/troubleshooting will not be performed during assignment marking.

Consider using techniques such as pseudocode statements, logic flow diagrams and screen-mock-ups for each module/page prior to coding to guide your development work and spot potential problems ahead of time.

### General Requirements:

- Marks will be allocated according to the marking guide below, so examine it closely and use it as your guide in both development and testing processes
- Data validation can be performed on the *server-side*, *client-side*, or *both* as you see appropriate
- Code commenting should be used to internally explain the various components of your scripts
- Your input validation messaging should be precise and efficient - not 'there was an error' type stuff.

### Marking Key

MARKING KEY		
Note: Marks will be assigned in 0.5 increments based on the extent of an item's implementation ranging from <b>not implemented</b> => [0] to <b>perfect implementation</b> => [full element value]		ELEMENT VALUE
BACKEND DATABASE		
SQL Server database correctly configured		/1
Correctly normalised table implemented with all required fields		/2
All fields set with appropriate data types		/2
Primary and foreign keys correctly applied		/1

LOGIN MODULE		
Required DB table contains the two (2) predefined roles (users) and data stipulated - Administrator and Events Manager		/1
Login/Logout processes function correctly		/1
Administrator role has access to <u>all</u> management modules		/1.5
Events Manager role <u>only</u> has access to Events and Reports management modules		/1.5
Only a valid combination of <i>username</i> and <i>password</i> allow access to the system		/1
GAMES MANAGEMENT MODULE		
A <i>new</i> Game can be <i>inserted</i>		/1
An existing Game can be <i>updated</i>		/1
An existing Game can be <i>deleted</i>		/1
Only <i>one (1)</i> instance of a Game can exist within the system		/1
Game_Code field only allows <i>seven (7)</i> characters comprised of <i>four (4)</i> uppercase characters and <i>three (3)</i> numerals		/1
Only Game_DurationInHours and Game_Description can be left NULL		/1
COMPETITORS MANAGEMENT MODULE		
A <i>new</i> Competitor can be <i>inserted</i>		/1
An existing Competitor can be <i>updated</i>		/1
An existing Competitor can be <i>deleted</i>		/1
Only <i>one (1)</i> instance of a Competitor can exist within the system		/1
Only Competitor_Bio may be <i>NULL</i>		/1
EVENTS MANAGEMENT MODULE		
A <i>new</i> Event can be <i>inserted</i>		/1
An existing event can be <i>updated</i>		/1
Event results can be <i>inserted/updated</i> as required		/1
There can only be <i>one (1)</i> instance of an event in the Event table		/1
No fields in the Event table can be left <i>null</i>		/1
REPORTS MANAGEMENT MODULE		
Displays the details of a selected event		/1
Lists a selected country's Gold, Silver and Bronze medals tally		/1
Displays Gold, Silver and Bronze medals tally for all countries in a ladder-style format		/2
Shows world record(s) achieved by Game (if any)		/1
Shows world record(s) achieved by Country (if any)		/1
Shows all world records achieved overall, including Game, Competitor and Country details		/2

PRESENTATION		
Useful and intuitive application navigation system provided		/2
Basic visual appeal and layout provided		/2
<b>GRAND TOTAL</b>	<b>/40</b>	

## Submission Requirements

You must submit the **complete** and **self-contained** *folder/file* structure required to run your .NET/C# application, including the backed SQL database. The solution must be placed in a single zipped folder named as follows studentID\_surname\_csi2441\_ass2, e.g. [555555\\_brown\\_csi2441\\_ass2.zip](#). This file must be submitted through the appropriate section of Blackboard on or before the due date specified.

Your .NET/C# solution will be run in Visual Studio Enterprise 2019 in its **.sln** project form with the start page being *index.aspx*. It's on this page that your lecturer/tutor will expect to find the Login form. Please note that your solution will **not** be *built* and *published* to a web server.

If your solution, or any specific part of it, does not run for whatever reason, such as:

- a) The file/folders submitted are *incomplete* to **not self-contained**, or
- b) Incompatibility issues caused by using a development environment other than that specified, or
- c) The SQL database was not submitted along with the project files, or
- d) Just straight-up *coding errors*, etc

then your tutor/lecturer will assign marks on a *code read-through* basis only. No troubleshooting will be attempted. Please note that only partial marks are available when a code read-through is required, so please:

- a) Use the development environment specified, Visual Studio Enterprise 2019
- b) Use the language specified, i.e. .NET/C#
- c) Test your application and application components thoroughly before submission
- d) Make sure the database component is embedded and travels with your application's project structure/tree
- e) Make sure you submit your solution in its entirety (complete and self-contained)

END OF ASSIGNMENT BRIEF