INB271 / INN271 "*The Web*"

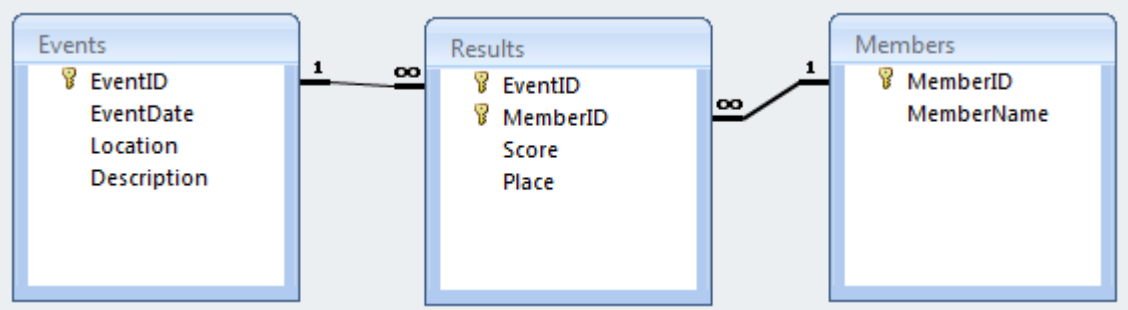
Assignment 2 – ASP.NET Web Implementation

Worth: 30%
Due: 28th May 2010
Teams: can be done individually or in pairs

Synopsis: This assignment will require you to implement part of a simple data-driven web application using ASP.NET.

Restrictions: The ASP.NET pages must be implemented in a declarative style using data bound and data source controls. Absolutely no VB.NET or C# can be written – you should remove all code behind files. Your web site should work in any modern browser - it should not rely on any technologies that require browser plug-ins (e.g. you cannot use Java Applets, Flash or SilverLight, etc).

Overview: This assignment continues the development of your club web site introduced in assignment 1. You will be able to reuse some of what you developed for assignment one, but some parts will need to be re-implemented and the functionality may need to change in some areas. For this assignment you have been provided with a Microsoft Access database which you must use.



You may not add any additional tables or alter any of the existing columns or indexes. You may however add additional columns to the existing tables as required to meet the specific needs of your club's competitions. You should expand the **`Members`** table with additional columns to record other pieces of information about each member (e.g. age, address, etc) as entered on your registration page from assignment 1. The **`Events`** table will contain information about your club's past competition events. The **`Results`** table is used to record which club members competed in those events and their results. The Score column is used to record the number of points, goals, runs, hands, sets etc scored or time taken, distance travelled etc (depending on the nature of the event). The Place column indicates whether that competitor came 1st, 2nd, 3rd, etc.
You cannot create any other tables and can only implement the following web pages, so your web site may be quite different to the one you designed for assignment one. Feel free to change it as much or as little as you wish or need to. You should endeavour to use a wide variety of ASP.NET web controls however you should always choose the control that is most appropriate for the task from a user's perspective.

Requirements:

1. As previously, you need to maintain a common look and feel across all pages in the web site. This is achieved via a common page layout and development of a style sheet used across the site. For this assignment, the common page layout must be achieved via the use of ASP.NET Master page(s). The **`Master page`** will define a page layout which includes at least a header section and a section for a navigation menu of some form down the left side of the screen for navigation to each of the other top level pages. A **`CSS file`** must be included in the master page and `div` elements used to achieve page layout. Your home (**`Default.aspx`**) page will need to be re-implemented to use this new master page. Your home page should provide general information about your club.

2. You need to re-implement your HTML registration page as an ASP.NET web form called **`RegisterMember.aspx`** that inserts data into the Members table. This insertion must be done entirely using data bound and data source controls (i.e. you may not write any C# or VB code to do the insertion). The data bound control should default to Insert Mode and should not support browsing, updating or deleting of Member information. . (The MemberID should not be provided by the user, but should be automatically generated by the Microsoft Access database). ASP.NET Validation controls should be used (instead of JavaScript) to perform client side validation for all fields as appropriate. After a user has successfully registered they should be automatically redirected to a simple page that indicates that their registration was successful.

3. You must implement a new web form called **`ListEvents.aspx`** which lists all of the events in the Events table. The EventID should not be displayed on any page. Dates should be displayed in short form (i.e. without the time of day). Each event entry displayed should include a hyperlink that if selected will take the user to a new page which lists all the results for that event.

4. You must implement a web form called **`ListResults.aspx`** which displays information about just one event and lists all results for that event. The event is specified via a query string parameter, e.g. **`ListResults.aspx?EventID=42`**. There should not be links to this web form from any of the main navigation menus; users should only come to this web form by way of the ListEvents page (so that a EventID is always specified). The top of the page should show all information about the event (from the Events table) and the bottom of the page should show a list of results for that event. The MemberID should not be displayed on any of the pages. The Member's name (from the Members table) should be shown instead. The results should by default be ordered by place (descending) but should be paginated and support sorting by other columns as selected by the end-user.

Administrator Pages: You are also required to implement a separate set of web pages to be used by club administrators to enter events and record results. The menus displayed to normal users should not include links to any of these admin pages. Administrators will need to manually navigate directly to the record events page in order to access this functionality. All of the admin pages should be protected by ASP.NET forms security. Any attempt to navigate to any of these pages by an unauthenticated user should result in them being automatically redirected to the Login.aspx page. Normal users should not be required to login to access any of the normal web pages.

5. You must implement a page called **`RecordEvents.aspx`**. The top of the page contains a form for entering information about new events. (The EventID should not be provided by the user, but should be automatically generated by the Microsoft Access database). ASP.NET Validation controls should be used (instead of Javascript) to perform client side validation for all fields as appropriate. The bottom of the page should display a list of all previously recorded events (set the `ViewState` property of this data bound control to `false` to cause it to be automatically refreshed when a new record is inserted via the form at the top of the page. Each event entry listed at the bottom of the page should include a hyperlink that will take the administrator to a new web form for recording results for that event. The EventID is passed via a query string parameter, e.g. **`RecordResults.aspx?EventID=4`**.

6. You must implement a new web form called **`RecordResults.aspx`**. The top of the page contains a form for entering results for competitors in that event (i.e. inserting into the Results table). The EventID is not entered by the user, but rather taken from the Query String. Administrators should only come to this web form by way of the RecordEvents page (so that a EventID is always specified). The member should be selected from a drop down list that is populated with the names of all club members. The bottom of the page displays a list of previously entered results for this event. Member names should be displayed rather than MemberIDs.

7. The **`Login.aspx`** page should include an ASP.NET login control. Authentication should be performed using ASP.NET Membership Services. You should use the ASP.NET Web Site administration tool to configure `Forms` authentication and to create a user with username "admin" and password "secret!".

Best Practice

When you are "finished", carefully scrutinize each page from a user's perspective. Is each field appropriately formatted (e.g. currency values)? We want to make sure that none of the "*implementation details" leaks* through into the *user experience*. For example, we create `ID` columns in our database to be used as a foreign key, but we don't want our users to ever have to see those IDs and we certainly never want them to have to key in such an ID. The exception is for IDs such as student numbers or social security numbers which are well know to the end users. We also need to be careful with auto generated label names – don't simply use the default label derived from the database column name. At the very least we would want to add spaces between the words to make it more *user friendly*. When a user finishes filling in their registration details – what would they expect the "submit" button to be labelled? I.e., don't just go with what is auto-generated, instead consider "*What would make most sense to the user?*" Reconsider the design principles from assignment 1; have I made good use of fonts, colours, alignment, whitespace, etc. Is it visually appealing and professionally presented?

Your assignment will be accessed not only on functional correctness, but also on adherence to best practises that support maintainability. This includes naming **everything** (especially control id's) with meaningful names – don't just accept the default names that Visual Studio generates (eg `GridView1`, `SqlDataSource2`). You should also manually delete auto-generated code that is not required. For example, we do not require UpdateCommands or EditTemplates for the registration page as it only needs to support insertion. Less code is better code – don't make the maintainer have to read and comprehend code that is not even used!

What to Submit

A single ZIP file containing your entire ASP.NET web site folder. Your tutor should be able to unzip your file, load it straight into Visual Studio 2008, browse all of your source code and successfully execute your home page (which should be set as the default start up page).
A `Readme.txt` file should be added to the web project that lists the names and student numbers of the students in the group and briefly describes any known omissions or bugs in your code.

How to Submit

Assignments must be submitted electronically via the Online Assessment System (OAS):
http://www.scitech.qut.edu.au/study/current/oas/