CPSC1517 Project Specs

Adventure Works Database

# Overview

This document outlines the marking guide and general requirements for the lab. Each student is expected to select one scenario from the catalogue of scenarios developed for the Adventure Works database. Every student in the class is required to select a different scenario, so that the details of each student’s lab are unique. For that scenario, the student must complete three fully-functional forms:

* Form A – Single item Create/Read/Update/Delete.
* Form B – GridView Lookup with Code-Behind.
* Form C – GridView Lookup with ObjectDataSource controls.

Your instructor will provide information on

* Where to access the catalogue of scenarios
* Where to obtain the database
* Any additional information on submitting/demonstrating the lab

# The Deliverables

A detailed marking guide is provided at the end of this document. For this lab, the student must demonstrate their working Visual Studio solution in **three** parts (listed below). Each demonstration of the student’s scenario must be presented to the instructor **in-class** in order to gain the marks for that part.

* **Part 1: 10 %** – **Due: First class of week 12**
  + Initial solution with documentation and navigation
* **Part 2: 15 %** – **Due: Second class of week 14**
  + Form A
* **Part 3: 10 %**  – **Due: Second class of week 15**
  + Forms B and C

Your instructor will provide specific dates on each deliverable. ***Late labs or failure to demonstrate in-class will receive a mark of zero for each affected part.***

# The Visual Studio Solution

Your solution must be a client-server solution consisting of multiple projects as outlined in the course. The Presentation Layer portion is to be an ASP.NET 4.5 website. The BLL and DAL portions of the system are to be placed in a separate class library project. Likewise, the entities of the system must be in their own separate class library project.

# The Adventure Works Database

The database supplied for this lab is an SQL Server database named “AdventureWorks\_CPSC1517”. This database is an enhanced version of the sample Adventure Works database provided by Microsoft. The enhancements consist primarily of a complete set of stored procedures for performing CRUD operations for each of the tables.

The following is a sample of the connection string that may be used for the Presentation Layer.

<connectionStrings>

<add name="AWContext" connectionString="Data Source=.;Initial Catalog=AdventureWorks\_CPSC1517;Integrated Security=True"

providerName="System.Data.SqlClient" />

</connectionStrings>

# The Forms

There are three web forms for the core functionality of this project (described below). You must name the forms **FormA.aspx**, **FormB.aspx**, and **FormC.aspx**. In addition, you must have a **Default.aspx** web form that acts as the lab documentation for your selected scenario. All of these forms must share the same Master Page, which must provide navigation among all the pages in your web site.

## Master Page

Your master page must contain the following elements:

* **Site Navigation** – Links to all the pages in the web site (use the Menu control).
* **Scenario Title** – The number and name of the scenario (e.g.: G1 – Reservations by Group).
* **Student Name** – Your first and last name.

## Default Page

Your default (home) page must contain the following elements (each with their own heading):

* **Form Description** – A brief description of each form in the solution, identifying the name of the form and its purpose, along with any unique constraints or characteristics of the form’s behaviour.
* **Known Bugs** – A bulleted list of all the known bugs and incomplete portions of the lab.
* **Entity Relationship Diagram** - The ERD diagram (from the lab selection) of your selected scenario.
* **Class Diagrams** – Class Diagram images of the BLL and Entities, showing the full method/property signatures of each class.
* **Stored Procedures** – A bulleted list of all the stored procedures used in your project.

## Form A – Single Item CRUD

This form is similar to the cumulative results of concepts demonstrated in the Data Access CRUD topics and exercises. In this form, the student must provide a means to

* Lookup and display a single row of data from a table
* Insert new rows of data into a table
* Update an existing row of data in a table
* Delete (or mark as inactive/not current) a row of data in a table

Depending on the scenario and the table, the Delete functionality may not be a physical removal of a row of data. This is because some tables have triggers or other constraints that prevent the removal of existing rows. In these cases where a row cannot be deleted (excluding foreign key constraints preventing removal), the table will have some means of flagging the row as being inactive or not current. For example, in the Employee table, a trigger prevents any rows from being deleted. However, that table has a column called “CurrentFlag” which can be set to indicate that the person is not currently employed. For advice on how to deal with these situations, please consult your instructor.

## Form B – Search & Display with a GridView

This form is similar to the concepts demonstrated in the Data Access GridView topic and exercise. In this form, the student must display multiple rows of data using a customized GridView that shows the first **five** columns/properties in the selected scenario. Note that all column headers **must** present user-friendly text (as opposed to the default title-cased property names of the entity). The form must perform some kind of lookup or search functionality.

## Form C – ObjectDataSource Search & Display with a GridView

This form is similar to the concepts demonstrated in the Data Bound Controls topic and exercise. In this form, the student must use a GridView to display multiple rows of data. All fields of the displayed entity must be used (except for rowguid and modifieddate) unless otherwise specification in the Important Information section.

For the GridView control, all foreign key information must be displayed/editiable via a DropDownList that displays appropriate information to the user while representing the value for the foreign key. Additionally, the GridView must enable paging if there are more than 50 rows of existing data in the database.

The controls in this form ***must*** be populated using ObjectDataSource controls only (no code-behind can be used to populate the forms).

# Marking Guides

Each deliverable part is cumulative, building on previous deliverable parts of the solution. For each deliverable, any and all supporting functionality of previous deliverables must function correctly in order to gain marks for the demonstrated deliverable.

### Part 1 – Scenario Number: \_\_\_\_\_\_\_\_

This portion of the lab consists of the structured solution, documentation, and functioning code for required entities and DAL layer.

Your instructor is free to make changes and/or additions to this lab and its marking guide. **You** are responsible to make sure that you are up-to-date with all the changes/additions made by your instructor.

|  |  |
| --- | --- |
| ✓ | Item |
| Lab Documentation (Web Form) | |
|  | Requested Lab documentation placed as the home page for the site (Default.aspx) |
|  | List of known bugs & incomplete portions of lab |
|  | Entity Relationship Diagram of selected scenario |
|  | Class diagram file of Entities in class library as well as exported image(s) on the Default page |
| Project Architecture & Code Quality | |
|  | Client-Server architecture (three projects inside a single solution) |
|  | Master Page with functioning site-master based navigation, Scenario Title and student’s name |
|  | Appropriate Entity classes coded for all the required tables |
|  | DAL class with appropriate code for all the required tables |
| Form A – Single Item CRUD | |
|  | Blank form with title of **selected option** in the student’s scenario. |
| Form B – GridView Lookup with Code-Behind | |
|  | Blank form with title of **selected option** in the student’s scenario. |
| Form C – GridView/FormView CRUD (must *not* use code-behind to populate controls) | |
|  | Blank form with title of **selected option** in the student’s scenario. |

### Part 2 – Scenario Number: \_\_\_\_\_\_\_\_

This portion of the lab consists of a functioning Form A, demonstrating Single Item CRUD supported by code-behind on the form and the BLL.

Your instructor is free to make changes and/or additions to this lab and its marking guide. **You** are responsible to make sure that you are up-to-date with all the changes/additions made by your instructor.

|  |  |
| --- | --- |
| ✓ | Item |
| Lab Documentation (Web Form) | |
|  | List of known bugs & incomplete portions of lab |
| Project Architecture & Code Quality | |
|  | Proper & consistent use of exceptions and exception handling |
|  | Separate BLL classes coded for all the required tables |
|  | BLL method(s) required for all form queries |
|  | BLL method required for Add |
|  | BLL method required for Update |
|  | BLL method(s) required for Delete |
| Form A – Single Item CRUD | |
|  | Form correctly populates DropDownList controls on Page\_Load event |
|  | Form correctly looks up and displays a single item |
|  | Form does ***not*** use DataSource controls to fill/populate data-bound controls |
|  | Form uses disabled (non-editable) controls for all data that is ***not*** related to Add/Update functionality |
|  | Form correctly uses Validation controls for client-side validation of Add/Update functionality |
|  | Form correctly adds data to a database |
|  | Form correctly gives up-to-date feedback on whether the add was successful or not |
|  | Form correctly updates data in the database |
|  | Form correctly gives up-to-date feedback on whether the update was successful or not |
|  | Form correctly performs Delete of data (or equivalent) in the database |
|  | Form correctly gives up-to-date feedback on whether the attempted delete (as applicable) is successful |

### Part 3 – Scenario Number: \_\_\_\_\_\_\_\_

This portion of the lab consists of the functioning Forms B and C and the related BLL components.

Your instructor is free to make changes and/or additions to this lab and its marking guide. **You** are responsible to make sure that you are up-to-date with all the changes/additions made by your instructor.

|  |  |  |
| --- | --- | --- |
| ✓ | | Item |
| Lab Documentation (Web Form) | | |
|  | List of known bugs & incomplete portions of lab | |
| Project Architecture & Code Quality | | |
|  | | Proper & consistent use of exceptions and exception handling |
|  | | Separate BLL classes coded for all the required tables |
| Form B – GridView Lookup with Code-Behind | | |
|  | | Form correctly handles Page\_Load event to populate DropDownList controls |
|  | | Form correctly performs Lookup and binds data to a GridView control |
|  | | GridView control uses column headers with meaningful names (not auto-generated) |
|  | | Form does ***not*** use DataSource controls to fill/populate data-bound controls |
| Form C – GridView/FormView CRUD (must *not* use code-behind to populate controls) | | |
|  | | BLL methods set up for Data Bound controls (DataObject and DataObjectMethod attributes) |
|  | | Form uses DataSource control(s) to supply data to the GridView |
|  | | GridView correctly displays data in the form |
|  | | GridView control uses column headers with meaningful names (not auto-generated) |
|  | | GridView control uses template columns with DropDownList control |