Project #3 – Access Database display front-end

DUE DATE: 2/25/20 – End of Day

Description:

Your sales department received your excel front-end application with great applause. Your IT department, however, does not feel it meets long-term needs of the organization and has defined new requirements for the project after your successful demonstration. Your IT department's resident DBA has converted the Excel file into a standard Access database, would like you to change the application to utilize that instead and permit the user to edit the fields.

Skills Used: interfaces, polymorphism, unit testing, form-based database applications using data-bound controls

User Experience:

1. At Startup, the user is presented with a form displaying no field data and record number zero
2. The user uses File > Open to browse for the access database file containing the information
3. The screen populates immediately with the first record in the database if any records exist
4. The user may click Preview or Next buttons to populate the fields with the appropriate record from the database
5. User cannot reduce the record number below zero
6. User cannot increase the record number beyond the maximum number of rows in the database
7. If the access data contains zero rows, then the row number must remain at zero
8. If the user attempts to enter empty text or whitespace into any field they edit, an error message is displayed and the user will not be able to save their changes to the database
9. The user may click on the 'Save' button to persist the changed form fields to the access database

Project Instructions

1. Clone or download the repository for this project to your local machine. If you have troubles retrieving the project from the repository URL, please contact the instructor for assistance.

   GITHUB PROJECT URL: https://github.com/gottjl01/cs203-project3-cs
   GITHUB REPOSITORY URL: https://github.com/gottjl01/cs203-project3-cs.git

2. You will need an internet connection to restore the nuget packages in this project.
3. Do not modify any files except the AccessPersister.cs and the newly added method in the frmMain.cs class
4. The AccessPersister.cs must be implemented in the following manner:
   a. The Default Constructor: This constructor must initialize the persister's local noDatabase variable to true and all methods must act as though the database is empty and no throw an exception
   b. The Single Parameter Constructor: This constructor accepts the file path to an access database that is used by the persister. If the supplied file does not exist an exception is thrown. The file path is stored in a local variable.
   c. GetRow() Method: This method must use OleDbConnection and OleDbCommand objects using a SQL statement (`SELECT * FROM [tblEmployees] where [ID] = @ID`) to retrieve the specific entry required.
   d. CountRows() Method: This method uses a SQL statement (`SELECT count(*) FROM [tblEmployees]`) to count the number of entries in the source table
   e. Dispose() Method: This method must set the IsDisposed variable to true once called. Once this variable is set to true, further calls to the other methods will throw ObjectDisposedExceptions
   f. GetData() Method: This method uses a SQL statement (`SELECT * FROM [tblEmployees]`) to load all data from the database table as a DataTable and returns it back to the caller

g. UpdateRow() Method: This method is a new method added to the IPersistData interface due to the requirement that the application be able to modify content in the database. It must make use of the OleDbConnection, OleDbCommand and a SQL statement (`UPDATE [tblEmployees] SET [Company] = @Company,[Last Name] = @LastName,[First Name] = @FirstName,[E-mail Address] = @EmailAddress,[Job Title] = @JobTitle,[Business Phone] = @BusinessPhone WHERE [ID] = @ID`) to make modifications to the database.

5. The frmMain.cs must be implemented in the following manner:
   a. `TxtFirstname_Validating` () Method: If the text of the control being modified is an empty string or whitespace, it must display a specific message ("`Value must be non-whitespace and non-empty`") and set the offending control as the source of the error using the ErrorProvider's SetError() method. Also, it must set the ErrorProvider's message to string.empty when the field is non-whitespace.
   NOTE: Despite the name of this method, it actually handles validation for all of the form's textboxes.
6. You must make all unit tests pass with the test explorer. Do not modify any of the unit tests
7. If you receive 'Provider not registered' errors when connecting to the access database, please download and install the following package from Microsoft: https://www.microsoft.com/en-us/download/details.aspx?id=54920

Code Documentation

Code comments are meant to provide a brief explanation in areas where the code is less self-explanatory. I ask that you use common sense and critically think about the places a reader of your code might need some guidance. In this project, I have provided many comments in the areas that are pre-implemented that you have been instructed not to modify. However, any method, field or property that you create or modify must have comments added using the standard IDE behavior in the Visual Studio IDE by typing three slashes (///) above a method, property or field. This will cause the IDE to automatically create the commenting structure that you will need to fill in. All parameters and return information must be completed.

Submission

Zip your assignment, including all source, project and solution file(s) and submit the archive through Blackboard by the due date. Late projects will be accepted with a one-week grace period, but no later barring extenuating circumstances, which must be communicated and approved by the professor.

Grading

Your grade is determined by the following rubic:

Scoring Rubic	
Assignment Task	**Points**
Unit Tests	40
Proper User Experience Behavior	5
Class, Method and Property Documentation	5
TOTAL	**50**

Extra Credit

You may earn additional credit on this project by implementing the same project using VB.NET. You can find the online github repos of the VB.NET version of these projects at the following URLs:

GITHUB PROJECT URL: https://github.com/gottjl01/cs203-project3-vb
GITHUB REPOSITORY URL: https://github.com/gottjl01/cs203-project3-vb.git

Please include the VB.NET solution with your normal project solution. It is best to place the two solutions in different folders and ZIP them up together for the Blackboard submission.

NOTE: The purpose of the extra credit is to encourage you to learn how to convert C# code into VB.NET code using the .NET Framework. DO NOT utilize any online or downloaded tools to automate this process. If your project is detected as having used such tools, you will receive zero extra credit, lose extra credit earned on any other projects and become ineligible for future extra credit on subsequent projects.