Programming 3 Assignment 6

Assignment Overview

- The specifications associated with this assignment will further explore LINQ. *LINQ-to-SQL Server* will be used to retrieve Student and Registration records from the BITCollege database. Data from the database will be displayed to the user using a BindingSource object bound to Windows controls including the DataGridView control. Masked controls will be used to appropriately display the data. End users will be able to perform transactions and verify the results of those transactions.
- Work associated with this assignment will reinforce the student's learning in the following areas:
    - Develop *LINQ-to-SQL Server* queries
    - Work with the DataGridView control
    - Work with overloaded constructors
    - Use WCF Web Services in a Windows project

Evaluation

- The evaluation of this assignment will be in the form of a Code Review and Run Time Demonstration.
- During the Code Review evaluation will be based on:
    - Comments
    - Documentation
    - Adherence to Standards – and Pattern Best Practices
    - Coding choices (i.e. refactoring, no repeated code, readability etc…)
    - Implementation of Requirements
    - Verbal discussion of code
- During the Run-Time Demonstration evaluation will be based on:
    - A variety of run-time tests

Specifications

Given Open Source: MaskedLabel

- An Open Source solution has been created by **EWSoftware** (maintained by the Code Project) which provides a label with masking capabilities.
- Add the MaskedLabel Project to the current solution. **Ensure the MaskedLabel project folder is in your solution directory**.
- Perform an initial build of the MaskedLabel project.

Programming 3 Assignment 6

Given Project: BITCollegeWindows

- Add the given BITCollegeWindows project to the assignment solution. **Ensure the BITCollegeWindows project folder is in your solution directory**.
- The BITCollegeWindows project contains classes to act as a starting point for the Windows College application
- Review the given content of the BITCollegeWindows Project as it pertains to this as well as upcoming assignments

References

- Add a reference to the BITCollege_FL MVC project.
- Add a reference to the Utility project – **Note**: The explicit reference to Utility may not be needed as it may be added implicitly when the BITCollege_FL reference is created.
- Add a reference to the MaskedLabel project.
- Add a **Service Reference** to the BITCollegeService WCF Web Service.
- Use the Nuget Package Manager to ensure that Entity Framework has been incorporated into the BITCollegeWindows project.

Database Connection

- Copy the ***<connectionStrings>*** block (including the tags) from the Web.Config file of the BITCollege_FL MVC project to the App.Config file of the BITCollegeWindows Project.
- Place the copied ***<connectionStrings>*** block just below the ***<configSections>*** block.

ConstructorData

- Within the ConstructorData class create two Auto-Implemented properties based on the following Models
  - Student
  - Registration
- These Auto-Implemented properties will be used to pass Student-specific and Registration-specific data among the forms in the BITCollegeWindows Project.
- Build the BITCollegeWindows Project.

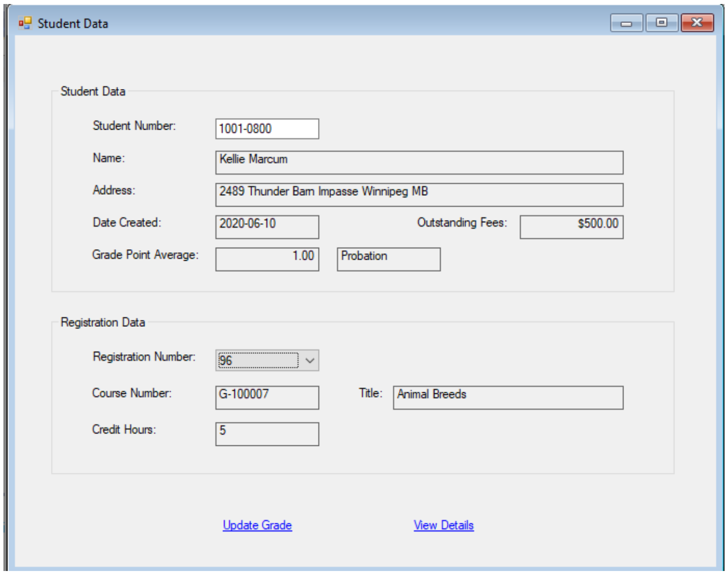
Programming 3 Assignment 6

DataSource

- Define an Object Data Source for the BITCollegeWindows Project by selecting the BITCollege_FL.Models as the Data Object.

Windows Form: StudentData

- Use the DataSource objects defined above to design the given StudentData form as shown below:



Student Data

Student Data

Student Number: 1001-0800

Name: Kellie Marcum

Address: 2489 Thunder Barn Impasse Winnipeg MB

Date Created: 2020-06-10

Outstanding Fees: $500.00

Grade Point Average: 1.00 Probation

Registration Data

Registration Number: 96

Course Number: G-100007

Title: Animal Breeds

Credit Hours: 5

Figure 1

Design

- Ensure that the generated BindingNavigator is removed from the form.
- For the Student Number field, use a MaskedTextBox control.
    - Set the Mask property of the MaskedTextBox control such that a hyphen appears between the first four and last four digits of the Student Number as shown above.
- For the Date Created, Outstanding Fees and Grade Point Average fields use labels.
    - Modify the Advanced data bindings such that the formatting of the data matches the format shown above.
    - Ensure data is properly aligned in these controls.
- For the Registration Number field use a ComboBox control.
    - Ensure the data items in the ComboBox are not editable.
    - Ensure the binding for the ComboBox is correct.

- For all other fields use Label controls.
    - In the Registration Data groupbox, obtain the Course Number, Title and Credit Hours via the Course navigation property in the Registration model.
- The label beside the Grade Point Average field is the GradePointState Description.
    - Obtain the GradePointState Description via the navigation property in the Student model.
- Remove any predefined Text values from all of the controls.
- Ensure the given Link Labels are initially disabled.
- Ensure the Student Number MaskedTextBox has initial focus.

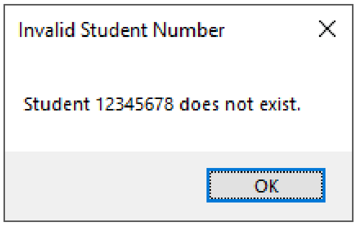
Script

- Some given code exists to allow for navigation between the Student form and some of the other forms used in the Windows Application.
    - Review the given code such that you understand the navigation techniques and variables used.

studentNumberMaskedTextBox_Leave Event

- Instantiate an object of your BITCollege_FLContext class.
- Ensure the user has completed the requirements for the Mask.
- Define a *LINQ-to-SQL Server* query selecting data from the Students table whose StudentNumber matches the value in the MaskedTextBox.
- If no records were retrieved:
    - Disable the Link Labels at the bottom of the form
    - Set focus back to the MaskedTextBox control.
    - Use the following syntax to clear the Student BindingSource object such that any previous results do not remain on the form:
        - ***studentBindingSource.DataSource = typeof(Student);***
    - Use the following syntax to clear the Registration BindingSource object such that any previous results do not remain on the form.
        - ***registrationBindingSource.DataSource = typeof(Registration);***

  - Display a MessageBox, as shown below, to the end user indicating that the student number entered does not exist.



Invalid Student Number

Student 12345678 does not exist.

Figure 2

- If a student record was retrieved:
  - Set the DataSource property of the BindingSource object representing the Student controls to the result set of the Student query.
  - Define a *LINQ-to-SQL Server* query selecting all Registrations in which the StudentId corresponds to the record represented by the StudentNumber value in the MaskedTextBox.
  - If no Registration records were retrieved:
    - Disable the link labels at the bottom of the form.
    - Clear the Registration BindingSource object such that any previous results do not remain on the form (See the syntax above).
  - If Registration record(s) were retrieved:
    - Set the DataSource property of the BindingSource object representing the Registration controls to the result set of the Registration query.
    - Enable the link labels at the bottom of the form.

Update Grade and View Details Link Labels Link Click

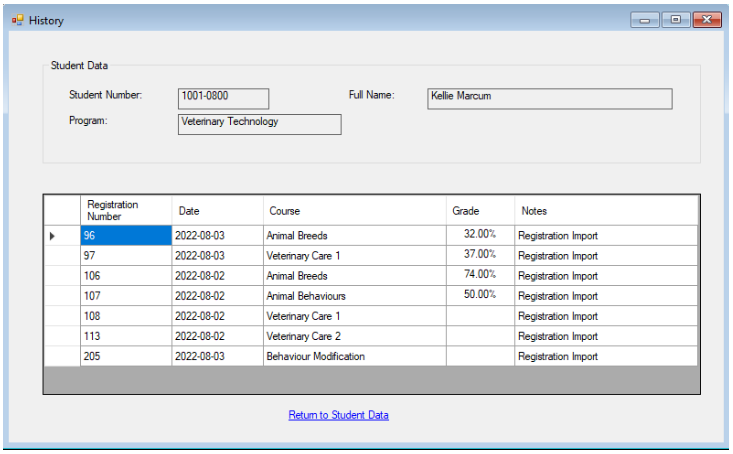
- For the Update Grade and View Details Link Labels LinkClick events, ensure the constructorData object is populated with the current Student record and current Registration record prior to constructing the Grading and History forms.
  - **Note**: Since both events require the same code, this is an excellent opportunity for refactoring to avoid repetitive code.

Programming 3 Assignment 6

Windows Form: History

Design

- Use the DataSource objects defined earlier to modify the design of the given History form as shown below:



History

Student Data

Student Number: 1001-0800

Full Name: Kellie Marcum

Program: Veterinary Technology

	Registration Number	Date	Course	Grade	Notes
▸	96	2022-08-03	Animal Breeds	32.00%	Registration Import
	97	2022-08-03	Veterinary Care 1	37.00%	Registration Import
	106	2022-08-02	Animal Breeds	74.00%	Registration Import
	107	2022-08-02	Animal Behaviours	50.00%	Registration Import
	108	2022-08-02	Veterinary Care 1		Registration Import
	113	2022-08-02	Veterinary Care 2		Registration Import
	205	2022-08-03	Behaviour Modification		Registration Import

Figure 3

- For the Student Number field use a MaskedLabel control.
    - Set the Student Number mask such that a hyphen appears between the first four and last four digits of the Student Number.
- For the Full Name and Program (Description) fields use Label controls.
    - Obtain the Academic Program Description via the navigation property in the Student model.
- The DataGridView will display information from the Registrations and Courses tables
    - Obtain the DataGridView from the Registration model in the Data Source.
    - Customize the DataGridView columns such that they display as shown above.
    - For the Date field use any date format that suppresses the timestamp.
    - For the Grade field use the format **P2**.
    - Set the Width property of the Course column to be 200px.
    - Set the Width property of the Grade column to be 50px and center the data.
    - Set the Width property of the Notes column to be 200px.
- Ensure that the generated BindingNavigator is removed from the form.

Script

- **<u>NOTE</u>**: Some given code exists to allow for navigation between this form and the StudentData form.

  - Review the given code such that you understand the navigation techniques and variables used.
- Instantiate an object of your BITCollege_FLContext class for use throughout this form.

History Constructor

- Following the given code, provide additional code to:
  - Populate the History form's constructorData object with the corresponding data received in the constructor.
  - Populate the upper controls with the corresponding data received in the constructor.

History_Load

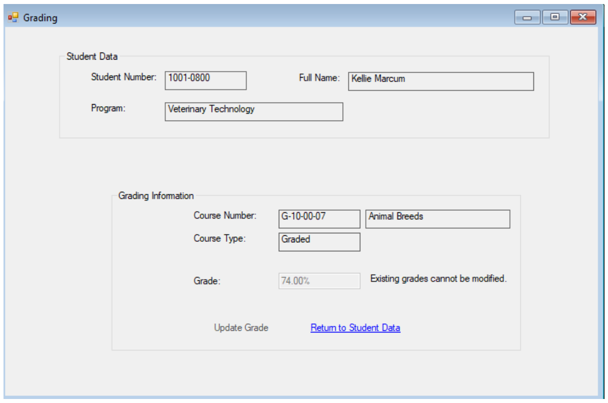
- Define a *LINQ-to-SQL Server* query selecting data from the Registrations and Courses tables whose StudentId corresponds to Student passed to this form.
  - **<u>Note</u>**: This **<u>must</u>** be achieved using a join operation.
  - Set the DataSource property of the BindingSource object representing the DataGridView control to the result set of this query.
- Ensure proper exception handling is used in this routine such that if an exception occurs, a MessageBox is shown to the end user providing details to the user as to the exception that has taken place.

Programming 3 Assignment 6

Windows Form: Grading

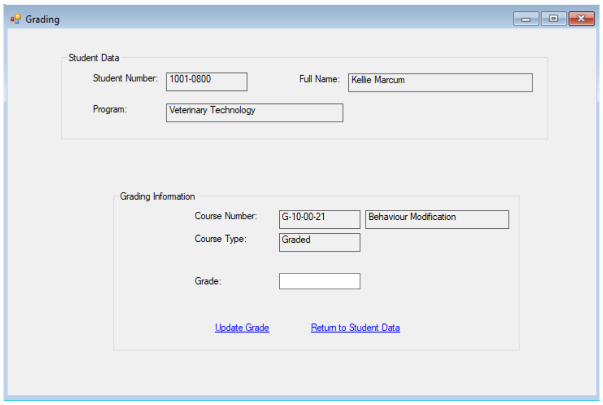
Design

- Use the DataSource objects defined earlier to modify the design of the given Grading form as shown below:



Student Number: 1001-0800
Full Name: Kellie Marcum
Program: Veterinary Technology
Course Number: G-10-00-07 Animal Breeds
Course Type: Graded
Grade: 74.00%
Existing grades cannot be modified.

Figure 4 Grading form when Grade is NOT editable



Student Number: 1001-0800
Full Name: Kellie Marcum
Program: Veterinary Technology
Course Number: G-10-00-21 Behaviour Modification
Course Type: Graded
Grade:

Figure 5 Grading form when Grade IS editable

- Use a MaskedLabel for the Student Number field.
  - Set the Student Number mask such that a hyphen appears between the first four and last four digits of the Student Number.
- For the Full Name and Program (Program) fields use Label controls.
  - Obtain the Academic Program Description via the navigation property in the Student model.
- For the fields in the lower GroupBox, use fields from the **Registration** entity classes in the DataSource.
  - Use a MaskedLabel for the Course Number field.
    - The Course Number mask will be determined through code later in the assignment.
  - Use Labels for the Course Title and Course Type fields
  - Use a TextBox for the Grade field.
    - Modify the DataBindings(Advanced) property by assigning a custom format of **P2** to the Grade TextBox.
  - Initially set the “Existing grades cannot…” label as invisible.
  - Initially set the Update LinkLabel as disabled.

- Ensure that the generated BindingNavigator is removed from the form.

Script

- **<u>NOTE</u>**: Some given code exists to allow for navigation between this form and the StudentData form.
    - Review the given code such that you understand the navigation techniques and variables used.

Grading Constructor

- Following the given code, provide additional code to:
    - Populate the Grading form's constructorData object with the corresponding data received in the constructor.
    - Populate the upper controls with the corresponding data received in the constructor.
    - Populate the lower controls with the corresponding data received in the constructor.

Grading_Load

- Use the given CourseFormat method within the Utility project to set the Course Number MaskedLabel mask.
    - **<u>Hint</u>**: Make use of the navigation properties available through the constructorData properties to send the appropriate argument to the Utility Project method.
- Ensure that controls are initially enabled and visible based on the following criteria:
    - If a grade has previously been entered:
        - Disable the Grade Textbox.
        - Disable the Update LinkLabel.
        - Make visible the label indicating that grading is not possible.
    - If no grade has been previously entered:
        - Enable the Grade Textbox.
        - Enable the Update LinkLabel.
        - Make invisible the label indicating that grading is not possible.

Update Grade LinkLabel LinkClicked

Discussion

- When the user enters a Grade (e.g. 0.50), the TextBox (if formatted correctly – see above) will correctly display the Grade as 50.00%. However, the entered grade must be updated as 0.50 according to the rules placed on the field when the model was designed. Therefore, the percent formatting must be stripped and the Grade value must be divided by 100 (getting it back to the original 0.50 value) before capturing that value for database update.
    - **Hint:** Removing the percent formatting can be done using a helper method found in the Utility project's Numeric class.

Requirement

- Define a String which will contain the value from the TextBox without the percent formatting.
- Use given functionality in the Utility project's Numeric class to ensure that the above value is numeric.
    - If non-numeric:
        - Display an appropriate MessageBox to the end user and do not proceed with the update.
    - If numeric:
        - Divide the above value by 100 and ensure this value is within the 0 - 1 range of numeric values.
        - If not within the range, display an appropriate MessageBox to the end user and do not proceed with the update.
            - Ensure the Messagebox explains that the grade must be entered as a decimal value such that it appropriately displays when formatted as a percent.
        - If the data is within the proper range of values, use the Client Endpoint of the WCF Web Service (created in an earlier assignment) to update the Grade, the Student's GradePointAverage and corresponding Grade Point State.
- Disable the Grade TextBox so that no further Grade modification can be made. This will also give the user a visual cue that the update has completed.

Programming 3 Assignment 6

Windows Form: StudentData

- The Student form must redisplay the data that had been initially selected when returning from the History and Grading forms.

Constructor

- Modify the overloaded constructor as follows:
    - Set the constructorData instance variable to the value of the corresponding argument.
    - Set the Student Number MaskedTextBox value using the Student property of the constructor argument.
    - Call the MaskedTextBox_Leave event passing null for each of the event arguments.

studentNumberMaskedTextBox_Leave(null, null);

MaskedTextBox Leave Event

- Modify the MaskedTextBox_Leave event as follows:
    - Following the code that populates the BindingSource object associated with the Registration controls with the results of the Registration query:
        - If the Registration object of ConstructorData is not null
            - Set the RegistrationNumber Combobox's text property to the value of the Registration Number returned to this form.

Test

- Test your solution to ensure it meets all of the requirements as defined above.

Backup

- Backup your Solution.
- Create a backup of your database and its corresponding log file.

Programming 3 Assignment 6

Prior to Evaluation

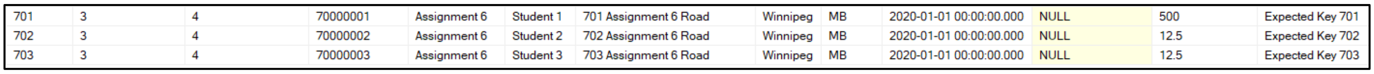
- Specific (sanitized) test data will be used for the evaluation of this assignment.
- In the Assignment 06 Files/SQL Server Script folder you will find an SQL Server script. Use this script in the instructions that follow:

1. Ensure you have created a backup of your database.

2. Modify the script called ***A6TestData.sql*** found in the Assignment 6 Files\SQL Server Scripts folder by replacing the [DATABASENAME] placeholder with the name of your BitCollege_FLContext database.

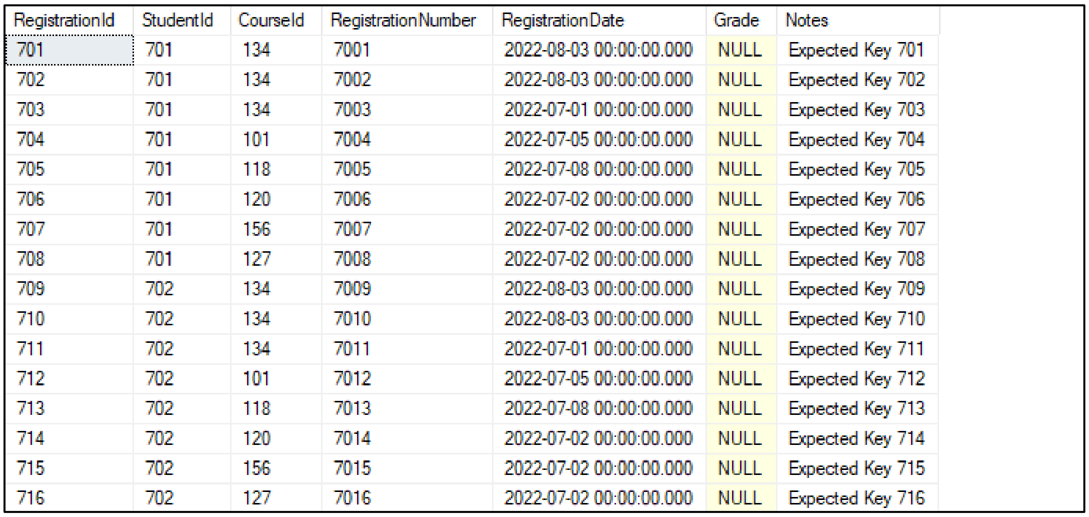
3. Run the script.

Three records should be inserted into the Students table starting with the primary key value of 701.



701	3	4	70000001	Assignment 6	Student 1	701 Assignment 6 Road	Winnipeg	MB	2020-01-01 00:00:00.000	NULL	500	Expected Key 701
702	3	4	70000002	Assignment 6	Student 2	702 Assignment 6 Road	Winnipeg	MB	2020-01-01 00:00:00.000	NULL	12.5	Expected Key 702
703	3	4	70000003	Assignment 6	Student 3	703 Assignment 6 Road	Winnipeg	MB	2020-01-01 00:00:00.000	NULL	12.5	Expected Key 703

Sixteen records should be inserted into the Registrations table starting with the primary key value of 701.



RegistrationId	StudentId	CourseId	RegistrationNumber	RegistrationDate	Grade	Notes
701	701	134	7001	2022-08-03 00:00:00.000	NULL	Expected Key 701
702	701	134	7002	2022-08-03 00:00:00.000	NULL	Expected Key 702
703	701	134	7003	2022-07-01 00:00:00.000	NULL	Expected Key 703
704	701	101	7004	2022-07-05 00:00:00.000	NULL	Expected Key 704
705	701	118	7005	2022-07-08 00:00:00.000	NULL	Expected Key 705
706	701	120	7006	2022-07-02 00:00:00.000	NULL	Expected Key 706
707	701	156	7007	2022-07-02 00:00:00.000	NULL	Expected Key 707
708	701	127	7008	2022-07-02 00:00:00.000	NULL	Expected Key 708
709	702	134	7009	2022-08-03 00:00:00.000	NULL	Expected Key 709
710	702	134	7010	2022-08-03 00:00:00.000	NULL	Expected Key 710
711	702	134	7011	2022-07-01 00:00:00.000	NULL	Expected Key 711
712	702	101	7012	2022-07-05 00:00:00.000	NULL	Expected Key 712
713	702	118	7013	2022-07-08 00:00:00.000	NULL	Expected Key 713
714	702	120	7014	2022-07-02 00:00:00.000	NULL	Expected Key 714
715	702	156	7015	2022-07-02 00:00:00.000	NULL	Expected Key 715
716	702	127	7016	2022-07-02 00:00:00.000	NULL	Expected Key 716

4. **Note**: If any testing based on the records inserted above takes place prior to the evaluation:
    a. Delete the newly inserted Student records from the database.

```
DELETE FROM [dbo].[Students]
      WHERE StudentId > 700
```

b. This will cause a cascade delete of related Registration records.

c. Re-run the ***A6TestData.sql*** script.

Evaluation: In Person/Hybrid Delivery

- This assignment must be evaluated on or before the due date/time specified in Learn.
    - The assignment is due at the beginning of class on the due date specified. Your instructor will evaluate the assignment on the students' machine in a random order. Remote evaluations will be based on a first come, first served basis.