Program Cover Sheet

|  |
| --- |
| Name: Madison Kell |
| Assignment: 8 |
| List any parts of the assignment that do not work/were not completed: |

|  |
| --- |
| Instructor’s Comments: |
| Grade: |

Program Submission Requirements: (1) all files, zipped and uploaded to Canvas and (2) a completed cover sheet, program execution screenshots and source code printed, **stapled** and turned in during class. Failure to follow the submission requirements will result in points lost on that particular assignment.

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' HOW TO USE DATABASE

' To run the database, I created the tables through

' the Visual Studio IDE. I called a database called

' vehicles, and the file is in my bin/debug folder.

' Therefore, you will have to reattach the database.

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Imports System.Data.SqlClient

'------------------------------------------------------------

'- File Name : Form1.frm -

'- Part of Project: Assign8 -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- File Purpose: -

'- This file contains the main application form where the -

'- user interacts with the gui to compelte their selected action

'- through code.

'------------------------------------------------------------

'- Program Purpose: -

'- The purpose of this program is to manage vehicles and

'- owners that are connected with and to a database.

'- The user can view, add, and delete users and their vehicles

'- that are inside the database.

'------------------------------------------------------------

'- Global Variable Dictionary (alphabetically): -

'- currentUsers - counter of the current number of users in the table

'- DBAdaptOwners - create data adapters so we don't mess stuff up trying to be cute with one adapter

'- DBAdaptVehicles - create data adapters so we don't mess stuff up trying to be cute with one adapter

'- deleted- array to hold the values of the deleted users

'- dsOwners - creating a dataset for owners table

'- dsVehicles - creating a dataset for vehicles table

'- gstrCity- this variable holds the value of the city text field

'- gstrConnString - This is the full connection string

'- gstrDBName- SQL Server database file

'- gstrDBPath- Path to database in executable

'- gstrSavedAddress - this variable holds the value of the address text field

'- gstrSavedNameF - this variable holds the value of the first name text field

'- gstrSavedNameL - this variable holds the value of the last name text field

'- gstrSERVERNAME- Name of the database server

'- strSQLCmdTable- string that holds the sql command that is passed when all data needs to be shown

'- gstrState- this variable holds the value of the state text field

'- gstrZip- this variable holds the value of the zip text field

'- index- index to show the value of the first owner

'- myConn- create a SqlConnection object since we will execute some straight SQL rather than relying on the DBAdapters

'------------------------------------------------------------

Public Class Form1

'index to show the value of the first owner

Dim index As Integer = 1

Const gstrDBName As String = "Assign8Database" 'SQL Server database file

Const gstrSERVERNAME As String = "(localdb)\MSSQLLocalDB" 'Name of the database server

Dim gstrDBPath As String = My.Application.Info.DirectoryPath & "\" & gstrDBName & ".mdf" 'Path to database in executable

Dim gstrConnString As String = "SERVER=" & gstrSERVERNAME & ";DATABASE=" & gstrDBName & ";Integrated Security=SSPI;AttachDbFileName=" & gstrDBPath 'This is the full connection string

'create data adapters so we don't mess stuff up trying to be cute with one adapter

Dim DBAdaptOwners As SqlDataAdapter

Dim DBAdaptVehicles As SqlDataAdapter

'creating a dataset for each table

Dim dsOwners As New DataSet

Dim dsVehicles As New DataSet

'string that holds the sql command that is passed when all data needs to be shown

Dim strSQLCmdTable As String

'counter of the current number of users in the table

Dim currentUsers As Integer

'array to hold the values of the deleted users

Dim deleted(0) As Integer

'create a SqlConnection object since we will execute some

'straight SQL rather than relying on the DBAdapters

Dim myConn As New SqlConnection(gstrConnString)

' listen, i know its bad, but I am not sure how to make this another way.

'These variables hold the value of the text fields

Dim gstrSavedNameF As String

Dim gstrSavedNameL As String

Dim gstrSavedAddress As String

Dim gstrCity As String

Dim gstrState As String

Dim gstrZip As String

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'------------------------------------------------------------

'- Subprogram Name: Form1\_Load -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called on form load. This sub calls a -

'- SQL command to initially populate the gui with data from the

'- tables

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- strSQLCmd - sql command to view all owners

'------------------------------------------------------------

' SQL command to get all from owners table

Dim strSQLCmd As String = "Select \* From Owners"

'loading all of the owners

DBAdaptOwners = New SqlDataAdapter(strSQLCmd, gstrConnString)

DBAdaptOwners.Fill(dsOwners, "Owners")

'show the appropriate buttons and text fields

makeEditable()

'connect data to textfields

loadBindings()

'load the database

loadDB()

'set the number of current users to the number of the rows in the the table

currentUsers = (dsOwners.Tables("Owners").Rows.Count)

End Sub

Private Sub btnFirstRecord\_Click(sender As Object, e As EventArgs) Handles btnFirstRecord.Click

'------------------------------------------------------------

'- Subprogram Name: btnFirstRecord\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- first record button. This shows whatever user is at index 1

'– in the table

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'bind the textfield to whatever owner is at postion 0 in the table (first record)

BindingContext(dsOwners, "Owners").Position = 0

'set index to 1 (first value in the table)

index = 1

'loading the associated vehicles to the user that is shown

strSQLCmdTable = loadingData(index)

'load the database

loadDB()

End Sub

Private Sub btnLastRecord\_Click(sender As Object, e As EventArgs) Handles btnLastRecord.Click

'------------------------------------------------------------

'- Subprogram Name: btnLastRecord\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- last record button. This shows whatever user is at the last

'– index in the table

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'if the index is greater than the current users, do nothing. This

'avoids showing extra data

If (index > currentUsers) Then

'do nothing

ElseIf (index < currentUsers) Then 'if the index is less than the current user

'set index = to whatever the amount of the current users there are

index = currentUsers

'bind the textfield to whatever owner is at postion of the last record

BindingContext(dsOwners, "Owners").Position = (dsOwners.Tables("Owners").Rows.Count - 1)

'if there are any deleted values

If deleted.Length > 1 Then

'loop through all of the deleted

For i As Integer = 0 To deleted.Length - 1

'set the deleted to the current index

If deleted(i) = index Then

'increase index number

index += 1

End If

Next

End If

End If

'loading the associated vehicles to the user that is shown

strSQLCmdTable = loadingData(index)

'load the database

loadDB()

End Sub

Private Sub btnRightArrow\_Click(sender As Object, e As EventArgs) Handles btnRightArrow.Click

'------------------------------------------------------------

'- Subprogram Name: btnRightArrow\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- >> button. This shows whatever user is at the next, scrolling

'– through the indexes in the table

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'if the index is greater than the current users, do nothing. This

'avoids showing extra data

If (index > currentUsers) Then

'do nothing

ElseIf (index < currentUsers) Then 'if the index is less than the current user

'increase the index by 1

index = index + 1

'bind the textfield to whatever owner is at postion of the next record

BindingContext(dsOwners, "Owners").Position = (BindingContext(dsOwners, "Owners").Position + 1)

'if there are any deleted values

If deleted.Length > 1 Then

'loop through all of the deleted

For i As Integer = 0 To deleted.Length - 1

'set the deleted to the current index

If deleted(i) = index Then

'increase index number

index += 1

End If

Next

End If

End If

'loading the associated vehicles to the user that is shown

strSQLCmdTable = loadingData(index)

'load the database

loadDB()

End Sub

Private Sub btnLeftArrow\_Click(sender As Object, e As EventArgs) Handles btnLeftArrow.Click

'------------------------------------------------------------

'- Subprogram Name: btnLeftArrow\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- << button. This shows whatever user is previous, scrolling

'– through the indexes in the table

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'if the index is greater than the current users, do nothing. This

'avoids showing extra data

If (currentUsers < index) Then

'do nothing

ElseIf (index > 1) Then 'if the index greater than 1

'decrease the index by 1

index = index - 1

'bind the textfield to whatever owner is at postion of the previous record

BindingContext(dsOwners, "Owners").Position = (BindingContext(dsOwners, "Owners").Position - 1)

'if there are any deleted values

If deleted.Length > 1 Then

'loop through all of the deleted

For i As Integer = 0 To deleted.Length - 1

'set the deleted to the current index

If deleted(i) = index Then

'increase index number

index -= 1

End If

Next

End If

End If

'loading the associated vehicles to the user that is shown

strSQLCmdTable = loadingData(index)

'load the database

loadDB()

End Sub

Private Sub btnAdd\_Click(sender As Object, e As EventArgs) Handles btnAdd.Click

'------------------------------------------------------------

'- Subprogram Name: btnAdd\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- add button. A new form for the user to add an owner will

'– be shown

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'make the data invisble so the user can add the owenr info peacfully

dgvResults.Visible = False

'set the textfields to the string variabl

gstrSavedNameF = txtFirstName.Text

gstrSavedNameL = txtLastName.Text

gstrSavedAddress = txtStreetAddress.Text

gstrCity = txtCity.Text

gstrState = txtState.Text

gstrZip = txtZip.Text

'load all of the bindings, make the buttons invisible, make the text fields editable

loadBindings()

makeInvisible()

clearTextFields()

showTextFields()

'show the cancel and add buttons

btnCancelAdd.Visible = True

btnSaveAdd.Visible = True

End Sub

Private Sub btnUpdate\_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click

'------------------------------------------------------------

'- Subprogram Name: btnUpdate\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- update button. The existing data selected by the user

'– becomes editable.

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'make the text fields editable

showTextFields()

'show the save and cancel buttons

btnSaveUpdate.Visible = True

btnCancelUpdate.Visible = True

'make the add, delete, update, and arrows invisible

makeInvisible()

End Sub

Private Sub btnDelete\_Click(sender As Object, e As EventArgs) Handles btnDelete.Click

'------------------------------------------------------------

'- Subprogram Name: btnDelete\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- delete button. The existing data selected by the user

'– becomes nonexistent

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- vUserInput - yes no message box

'-strSQLCommand - sql command

'------------------------------------------------------------

'yess no message box to make sure user knows what they are doing

Dim vUserInput = MsgBox("Are you sure you want to delete this user?", vbYesNo, "Hold on!")

'Process User Input Yes,No

Select Case vUserInput

'if the user selects yes

Case vbYes

'create an sql command

Dim strSQLCommand As String

'delete the selected owner with sql, load the information

strSQLCommand = "DELETE FROM Owners WHERE TUID = " & txtIDNumber.Text

DBAdaptOwners = New SqlDataAdapter(strSQLCommand, gstrConnString)

DBAdaptOwners.Fill(dsOwners, "Owners")

'delete the selected vehicle with sql, load the information

strSQLCommand = "DELETE FROM Vehicles WHERE OwnerID= " & txtIDNumber.Text

DBAdaptOwners = New SqlDataAdapter(strSQLCommand, gstrConnString)

DBAdaptOwners.Fill(dsVehicles, "Vehicles")

'clear the owners and relaod the data

dsOwners.Clear()

strSQLCommand = "SELECT \* FROM Owners"

DBAdaptOwners = New SqlDataAdapter(strSQLCommand, gstrConnString)

DBAdaptOwners.Fill(dsOwners, "Owners")

'clear the vehicles and reload the data

dsVehicles.Clear()

strSQLCommand = "SELECT \* FROM Vehicles"

DBAdaptVehicles = New SqlDataAdapter(strSQLCommand, gstrConnString)

DBAdaptVehicles.Fill(dsVehicles, "Vehicles")

'set the binding to the position of the previous value

BindingContext(dsOwners, "Owners").Position = BindingContext(dsOwners, "Owners").Position

'load the owner data

DBAdaptOwners.Fill(dsOwners, "Owners")

'allow both dataset to accept the changed and refresh the datagridview

dsOwners.AcceptChanges()

dsVehicles.AcceptChanges()

dgvResults.Refresh()

'set the index to the 1 adn subtract a user from the current number of users

index = 1

currentUsers -= 1

'load the vehicle data from the owners

loadingData(currentUsers)

'load the database

loadDB()

End Select

End Sub

Private Sub btnCancelUpdate\_Click(sender As Object, e As EventArgs) Handles btnCancelUpdate.Click

'------------------------------------------------------------

'- Subprogram Name: btnCancelUpdate\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- cancel button after selecting the update button

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'load the text fields with the data from the table

loadBindings()

'set the binding to the position of the previous value

BindingContext(dsOwners, "Owners").Position = BindingContext(dsOwners, "Owners").Position

'load the vehicle data

loadingData(index)

'show the appropraite buttons and make text feilds uneditable again

makeEditable()

makeVisible()

'loading the associated vehicles to the user that is shown

strSQLCmdTable = loadingData(index)

'load the database

loadDB()

End Sub

Private Sub btnSaveUpdate\_Click(sender As Object, e As EventArgs) Handles btnSaveUpdate.Click

'------------------------------------------------------------

'- Subprogram Name: btnSaveUpdate\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- save button after updating the user

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'call the update data method which successfully updates the user

updateData()

'show the appropraite buttons and make text feilds uneditable again

makeEditable()

makeVisible()

End Sub

Private Sub btnCancelAdd\_Click(sender As Object, e As EventArgs) Handles btnCancelAdd.Click

'------------------------------------------------------------

'- Subprogram Name: btnCancelAdd\_Click

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- cancel button after clicking the add button –

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'show the vehicles

dgvResults.Visible = True

'set the binding to the position of the previous value and fill the data

BindingContext(dsOwners, "Owners").Position = BindingContext(dsOwners, "Owners").Position

DBAdaptOwners.Fill(dsOwners, "Owners")

'load the vehicle data

loadBindings()

'show the appropraite buttons and make text feilds uneditable again

makeEditable()

makeVisible()

'loading the associated vehicles to the user that is shown

strSQLCmdTable = loadingData(index)

'set the textfields to the string that I saved above

txtFirstName.Text = gstrSavedNameF

txtLastName.Text = gstrSavedNameL

txtStreetAddress.Text = gstrSavedAddress

txtCity.Text = gstrCity

txtState.Text = gstrState

txtZip.Text = gstrZip

End Sub

Private Sub btnSaveAdd\_Click(sender As Object, e As EventArgs) Handles btnSaveAdd.Click

'------------------------------------------------------------

'- Subprogram Name: btnSaveAdd\_Click -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- save button after adding the user

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'increase the id

txtIDNumber.Text = CStr(dsOwners.Tables("Owners").Rows.Count + 1)

'show the vehicles

dgvResults.Visible = True

'call the add user method created

addUser()

'show the appropraite buttons and make text feilds uneditable again

makeEditable()

makeVisible()

End Sub

Sub addUser()

'------------------------------------------------------------

'- Subprogram Name: addUser -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- save add button, this method adds the user to the system

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- newID - hold the value of the newid

'- strSQLCmd - new sql command string

'- DBCmd - new sql command

'------------------------------------------------------------

'hold the value of the newid

Dim newID As Integer

'new sql command string

Dim strSQLCmd As String

'new sql command

Dim DBCmd As SqlCommand = New SqlCommand()

'make the appropriate buttons invidible

makeInvisible()

'increase the current users by 1

currentUsers += 1

'trim and string the id number

newID = Trim(CStr((CInt(txtIDNumber.Text))))

'if there are any deleted values

For i As Integer = 0 To deleted.Length - 1

'if the deleted value is not = to 0

If deleted(i) <> 0 Then

'set the new id to the spot of the previous deleted spot

newID = deleted(i)

'set the array value to 0

deleted(i) = 0

End If

Next

'set the command to get all of the owners

strSQLCmd = "Select \* From Owners"

'open a connection to the db

myConn.Open()

'set the command text to insert the new owner

DBCmd.CommandText = "INSERT INTO Owners (TUID, FirstName, LastName, StreetAddress, City, State, ZipCode)

VALUES ('" & newID & "','" & txtFirstName.Text & "','" & txtLastName.Text & "','" & txtStreetAddress.Text & "','" & txtCity.Text & "','" & txtState.Text & "','" & txtZip.Text & "')"

'set the connection execute the query

DBCmd.Connection = myConn

DBCmd.ExecuteNonQuery()

'increase the index

index = index + 1

'load the vehicles

loadingData(index)

'load all of the owners and relaod the information

DBAdaptOwners = New SqlDataAdapter(strSQLCmd, gstrConnString)

dsOwners.Clear()

DBAdaptOwners.Fill(dsOwners, "Owners")

'accept the changes and refresh the datagridview

dsOwners.AcceptChanges()

dgvResults.Refresh()

'set index back to first user and load their vehicles

index = 1

loadingData(index)

'laod the database

loadDB()

'show confirmation message and close the conenction

MessageBox.Show("User added successfully.", "Congrats!")

myConn.Close()

End Sub

Sub updateData()

'------------------------------------------------------------

'- Subprogram Name: updateData -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- save update button, this method updates the user to the system

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- cmdBuilder -new command builder

'- strSQLCmd - new sql command string

'------------------------------------------------------------

'new command builder

Dim cmdBuilder As SqlCommandBuilder

'new sql commande string

Dim strSQLCmd As String

'selecting all owners with the correct id

strSQLCmd = "Select \* From Owners Where TUID = '" & Trim(txtIDNumber.Text) & "'"

'end the current edit and open the connection

BindingContext(dsOwners, "Owners").EndCurrentEdit()

myConn.Open()

'reload all fo the information

DBAdaptOwners = New SqlDataAdapter(strSQLCmd, gstrConnString)

cmdBuilder = New SqlCommandBuilder(DBAdaptOwners)

DBAdaptOwners.InsertCommand = cmdBuilder.GetInsertCommand

'update the database, close the connection, accept the changes, load the database

DBAdaptOwners.Update(dsOwners, "Owners")

myConn.Close()

dsOwners.AcceptChanges()

loadDB()

End Sub

Sub loadDB()

'------------------------------------------------------------

'- Subprogram Name: loadDB -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the database loads

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- DBConn-

'- strSQLCmdTable - setting sql command to the table information in the loading data sub

'- DBCommand - new sql command

'- myDataSet - new dataset to not mess stuff up

'- DBAdapter - create data adapter so we don't mess stuff up

'------------------------------------------------------------

'create a SqlConnection object since we will execute some

'straight SQL rather than relying on the DBAdapters

Dim DBConn As SqlConnection = New SqlConnection(gstrConnString)

'setting sql command to the table information in the loading data sub

Dim strSQLCmdTable As String = loadingData(index)

'new sql command

Dim DBCommand As New SqlCommand

'new dataset to not mess stuff up

Dim myDataSet As New DataSet

'create data adapter so we don't mess stuff up

Dim DBAdapter As New SqlDataAdapter

'open connection

DBConn.Open()

'loading all of the table informaiton from owners and vehicles and setting the adapte

DBCommand.CommandText = strSQLCmdTable

DBAdapter = New SqlDataAdapter(strSQLCmdTable, DBConn)

'filling the adapter with the vehicles table information

DBAdapter.Fill(myDataSet, "Vehicles")

'settign the data grid view to the information in the vehicles table

dgvResults.DataSource = myDataSet.Tables("Vehicles")

'close connection

DBConn.Close()

End Sub

Sub loadBindings()

'------------------------------------------------------------

'- Subprogram Name: loadBindings -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the text fields need

'– to be populated with the information from the owners table

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Set bindings on the course related fields since we now have some data in

'context and can understand the schema of the table

If txtStreetAddress.DataBindings.Count = 0 Then

txtIDNumber.DataBindings.Add(New Binding("Text", dsOwners, "Owners.TUID"))

txtFirstName.DataBindings.Add(New Binding("Text", dsOwners, "Owners.FirstName"))

txtLastName.DataBindings.Add(New Binding("Text", dsOwners, "Owners.LastName"))

txtStreetAddress.DataBindings.Add(New Binding("Text", dsOwners, "Owners.StreetAddress"))

txtCity.DataBindings.Add(New Binding("Text", dsOwners, "Owners.City"))

txtState.DataBindings.Add(New Binding("Text", dsOwners, "Owners.State"))

txtZip.DataBindings.Add(New Binding("Text", dsOwners, "Owners.ZipCode"))

End If

End Sub

Sub makeEditable()

'------------------------------------------------------------

'- Subprogram Name: makeEditable -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the save,

'– cancel button as well as when the form loads

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'making txt fields not able to be edited

txtFirstName.Enabled = False

txtLastName.Enabled = False

txtCity.Enabled = False

txtState.Enabled = False

txtStreetAddress.Enabled = False

txtZip.Enabled = False

txtIDNumber.Enabled = False

'not showing save or cancel buttons

btnSaveAdd.Visible = False

btnSaveUpdate.Visible = False

btnCancelAdd.Visible = False

btnCancelUpdate.Visible = False

End Sub

Sub showTextFields()

'------------------------------------------------------------

'- Subprogram Name: showTextFields -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user needs to add

'– or update a text field.

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'making text fields editable

txtFirstName.Enabled = True

txtLastName.Enabled = True

txtStreetAddress.Enabled = True

txtCity.Enabled = True

txtState.Enabled = True

txtZip.Enabled = True

End Sub

Sub makeVisible()

'------------------------------------------------------------

'- Subprogram Name: makeVisible -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user loads the porgram,

'– or clicks a save or cancel button. This sub shows the correct

'- buttons

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'making buttons visible

btnDelete.Visible = True

btnAdd.Visible = True

btnUpdate.Visible = True

btnRightArrow.Visible = True

btnLastRecord.Visible = True

btnLeftArrow.Visible = True

btnFirstRecord.Visible = True

End Sub

Sub makeInvisible()

'------------------------------------------------------------

'- Subprogram Name: makeInvisible -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks

'– add or cancel. This sub shows the correct

'- buttons

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'hiding buttons that do not need to be shown in update or add mode

btnDelete.Visible = False

btnAdd.Visible = False

btnUpdate.Visible = False

btnRightArrow.Visible = False

btnLastRecord.Visible = False

btnLeftArrow.Visible = False

btnFirstRecord.Visible = False

End Sub

Sub clearTextFields()

'------------------------------------------------------------

'- Subprogram Name: clearTextFields -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the text fields need

'– to be cleared

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None)

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'clearing all of the text fields

txtFirstName.Text = ""

txtLastName.Text = ""

txtState.Text = ""

txtStreetAddress.Text = ""

txtZip.Text = ""

txtCity.Text = ""

txtIDNumber.Text = ""

End Sub

Function loadingData(id As Integer)

'------------------------------------------------------------

'- Subprogram Name: loadingData -

'------------------------------------------------------------

'- Written By: Madison Kell

'- Written On: 4/3/2022

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the data needs to be

'– loaded from the tables.

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- id As Integer- passing the id allows for correct vehicle data

'- to be shown with the right owner

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'- Returns: -

'- Integer – telling how many records were found -

'------------------------------------------------------------

'sql command to slect the information from the vehicle table

' and join it to the owner infromation by the id and tuid

Dim query = "SELECT v.Make, v.Model, v.Color, v.ModelYear, v.VIN

FROM Vehicles v

JOIN Owners o

ON v.OwnerID = o.TUID

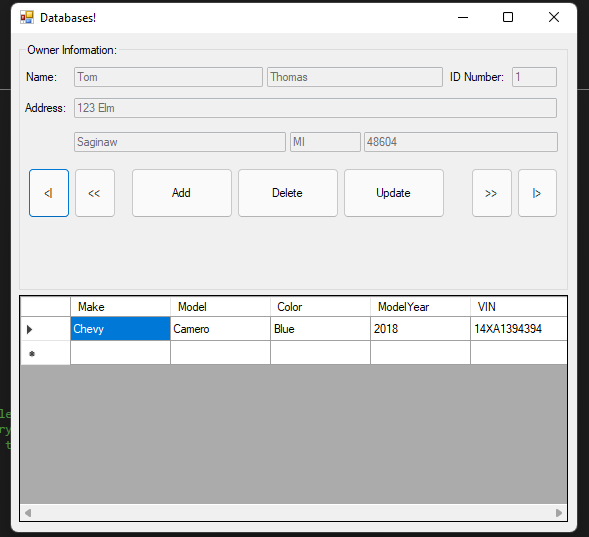
Where v.OwnerID = " & CStr(id)

'reutrnign the query to be called

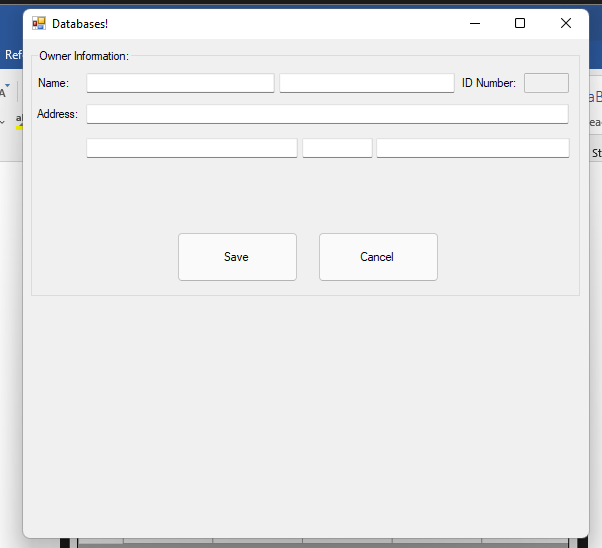
Return query

End Function

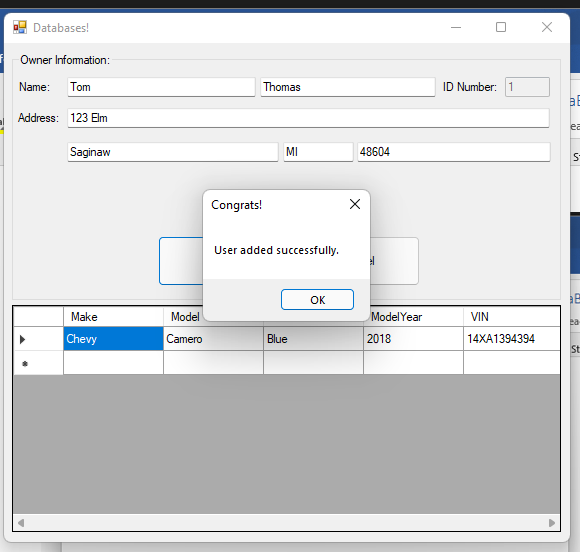
End Class

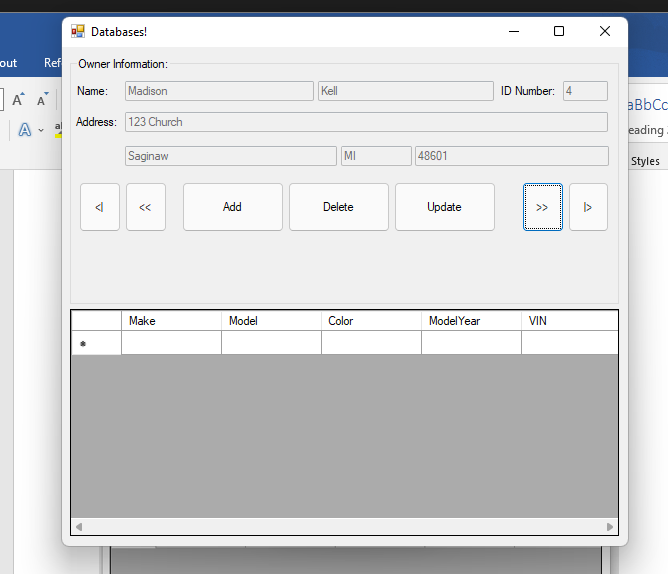


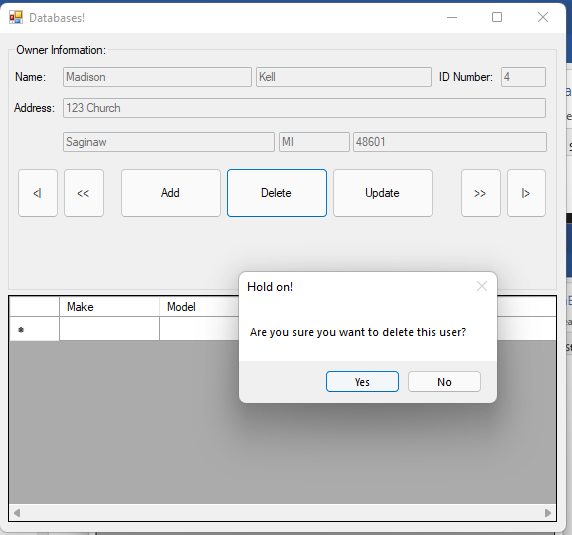
Add



Add – save







Update

