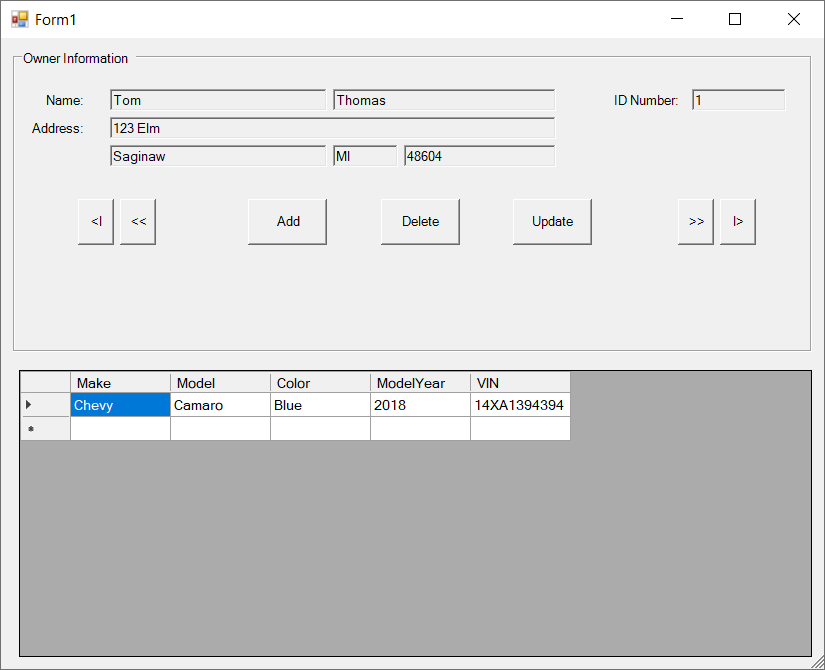
Program Cover Sheet

|  |
| --- |
| Name: Nathan VanSnepson |
| Assignment: Assignment 8 |
| List any parts of the assignment that do not work/were not completed: (None) |

|  |
| --- |
| 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.



Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

‘Database and tables are generated by the program!!!!!!!!!!!

‘All you have to do is run the program

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

'- File Name : Main.vb -

'- Part of Project: Assignment8 -

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

' Written By: Nathan VanSnepson -

' Written On: April 4, 2022 -

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

'- File Purpose: -

'- -

'- This file contains the main method for the entire project-

'- It handles creating a database with 2 tables then loading-

'- Form1 for user interaction. -

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

'- Program Purpose: -

'- -

'- The purpose of this program is to view the relationship -

'- between Owners and Vehicles and to be able to add new -

'- owners. -

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

'- Global Variable Dictionary (alphabetically): -

'-\*strCONNECTION - Connection String for database -

'-\*strDBPATH - File path to the database file -

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

Imports System.Data.SqlClient

Module Main

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

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

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

Const strDBNAME As String = "VehicleOwners" 'Name of the database

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

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

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

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

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

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

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

Dim strDBPATH As String = My.Application.Info.DirectoryPath & "\" & strDBNAME & ".mdf" 'Path of the database file

'Connection String for database

Dim strCONNECTION As String = "SERVER=" & strServerName & ";DATABASE=" & strDBNAME &

";Integrated Security=SSPI;AttachDbFileName=" & strDBPATH

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

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

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

Sub main()

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

'- Subprogram Name: main -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to initialize the -

'- program by calling a method to create the database then -

'- displaying a form for the user to interact with the -

'- tables. -

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

'- Parameter Dictionary (in parameter order): -

'- (None) -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'Calls method to begin creating database

connectDatabases()

''Show the main dialog

Form1.ShowDialog()

'Prompts to delete the database

If MessageBox.Show("Do you want to physically delete the database?", "",

MessageBoxButtons.YesNo) = DialogResult.Yes Then

DeleteDatabase(strServerName, strDBNAME)

End If

End Sub

#Region "Create and Connect Database"

Private Sub connectDatabases()

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

'- Subprogram Name: connectDatabases -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to call the methods to -

'- create the database, delete all records from the tables, -

'- then populate the tables -

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

'- Parameter Dictionary (in parameter order): -

'- (None) -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'If the file does the exist then create the database

If Not (System.IO.File.Exists(strDBNAME & ".mdf")) Then

CreateDatabase(strServerName, strDBNAME, strDBPATH, strCONNECTION)

End If

'Make sure all tables are cleaned out each time we run this

CleanOutTables(strCONNECTION, "Owners")

CleanOutTables(strCONNECTION, "Vehicles")

'Put some data into the tables

PopulateOwnersTable(strCONNECTION)

PopulateVehiclesTable(strCONNECTION)

End Sub

Private Sub CreateDatabase(ByVal strSERVERNAME As String, ByVal strDBNAME As String,

ByVal strDBPATH As String, ByVal strCONNECTION As String)

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

'- Subprogram Name: CreateDatabase -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to create the database -

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

'- Parameter Dictionary (in parameter order): -

'-\*strSERVERNAME - Name of the databaseserver -

'-\*strDBNAME - Name of the database -

'-\*strDBPATH - Path to the database -

'-\*strCONNECTION - Connection string for database -

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

'- Local Variable Dictionary (alphabetically): -

'-\*DBConn - SQLConnection used to connect database -

'-\*DBCmd - SqlCommand for running sql code -

'-\*strSQLCmd - The sql command to run -

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

'Let's build a SQL Server database from scratch

Dim DBConn As SqlConnection

Dim DBCmd As SqlCommand

Dim strSQLCmd As String

'All we need to do initially is just point at the server

DBConn = New SqlConnection("Server=" & strSERVERNAME)

'Let's write a SQL DDL Command to build the database

'There are a lot of other parameters but we can let them default

'All we need are these three

strSQLCmd = "CREATE DATABASE " & strDBNAME & " On " &

"(NAME = '" & strDBNAME & "', " &

"FILENAME = '" & strDBPATH & "')"

DBCmd = New SqlCommand(strSQLCmd, DBConn)

Try

'Open the connection and try running the command

DBConn.Open()

DBCmd.ExecuteNonQuery()

'MessageBox.Show("Database was successfully created", "",

'MessageBoxButtons.OK, MessageBoxIcon.Information)

Catch ex As Exception

'If we can't build the database, we are dead in the water so bail...

MessageBox.Show(ex.ToString())

MessageBox.Show("Cannot build database! Closing program down...")

End

End Try

'We are currently pointing at the [MASTER] database, so we

'need to close the connection and reopen it pointing at the

'Registration database...

If (DBConn.State = ConnectionState.Open) Then

DBConn.Close()

End If

'Now we need to use the full connection string with the Integrated

'Security line, et cetera

DBConn = New SqlConnection(strCONNECTION)

DBConn.Open()

'Build the Tables one at a time

'Build the Student Table by writing the SQL DDL Command

DBCmd.CommandText = "CREATE TABLE Owners (" &

"[TUID] INT NOT NULL, " &

"[FirstName] NVARCHAR(50), " &

"[LastName] NVARCHAR(50), " &

"[StreetAddress] NVARCHAR(50), " &

"[City] NVARCHAR(50), " &

"[State] NVARCHAR(50), " &

"[ZipCode] NVARCHAR(50), " &

"PRIMARY KEY CLUSTERED ([TUID] ASC)" &

")"

DBCmd.Connection = DBConn

Try

DBCmd.ExecuteNonQuery()

'MessageBox.Show("Created Owners Table")

Catch Ex As Exception

MessageBox.Show("Clients Table Already Exists")

End Try

'Build the Courses Table

DBCmd.CommandText = "CREATE TABLE Vehicles (" &

"[TUID] INT NOT NULL, " &

"[OwnerID] INT NULL," &

"[Make] NVARCHAR(50)," &

"[Model] NVARCHAR(50)," &

"[Color] NVARCHAR(50)," &

"[ModelYear] INT," &

"[VIN] NVARCHAR(50)," &

"PRIMARY KEY CLUSTERED ([TUID] ASC)" &

")"

DBCmd.Connection = DBConn

Try

DBCmd.ExecuteNonQuery()

'MessageBox.Show("Created Vehicles Table")

Catch Ex As Exception

MessageBox.Show("Courses Table Already Exists")

End Try

'We can check to see if we're open before trying to

'issue a connection close

If (DBConn.State = ConnectionState.Open) Then

DBConn.Close()

End If

End Sub

Sub CleanOutTables(ByVal strConn As String, ByVal strTableName As String)

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

'- Subprogram Name: CleanOutTables -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Delete all records -

'- from a table -

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

'- Parameter Dictionary (in parameter order): -

'-\*strConn - Connection string to database -

'-\*strTableName - Name of the table in the database -

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

'- Local Variable Dictionary (alphabetically): -

'-\*DBConn - SQLConnection used to connect database -

'-\*DBCmd - SqlCommand for running sql code -

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

Dim DBConn As SqlConnection

Dim DBCmd As SqlCommand = New SqlCommand()

'Now try to open up a connection to the database

DBConn = New SqlConnection(strConn)

DBConn.Open()

'Use SQL DML to zap the contents of the table

DBCmd.CommandText = "DELETE FROM " & strTableName

DBCmd.Connection = DBConn

DBCmd.ExecuteNonQuery()

'Close the DBConnection

DBConn.Close()

End Sub

Sub PopulateOwnersTable(ByVal strConn As String)

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

'- Subprogram Name: PopulateOwnersTable -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to populate the Owners -

'- table -

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

'- Parameter Dictionary (in parameter order): -

'-\*strConn - Connection string to database -

'-\*strTableName - Name of the table in the database -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

InsertOwner(strConn, "1", "Tom", "Thomas", "123 Elm", "Saginaw", "MI", "48604")

InsertOwner(strConn, "2", "Jane", "Jones", "456 Pine", "Saginaw", "MI", "48605")

InsertOwner(strConn, "3", "Bob", "Fredericks", "789 Maple", "Birch Run", "MI", "48415")

InsertOwner(strConn, "4", "John", "Doe", "7645 Center", "Bay City", "MI", "48732")

End Sub

Sub PopulateVehiclesTable(ByVal strConn As String)

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

'- Subprogram Name: PopulateVehiclesTable -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to populate the -

'- Vehicles table -

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

'- Parameter Dictionary (in parameter order): -

'-\*strConn - Connection string to database -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

InsertVehicle(strConn, "1", "1", "Chevy", "Camaro", "Blue", "2018", "14XA1394394")

InsertVehicle(strConn, "2", "2", "Ford", "F-150", "Red", "2017", "2A7764747236")

InsertVehicle(strConn, "3", "2", "Dodge", "Dart", "Red", "2017", "45B6D7667")

InsertVehicle(strConn, "4", "3", "Kia", "Soul", "Green", "2013", "1A1467464484")

InsertVehicle(strConn, "5", "3", "Dodge", "Viper", "Yello", "2014", "48J764E7633")

End Sub

Private Sub InsertOwner(ByVal strConn, ByVal strTUID, ByVal strFirstName, ByVal strLastName, ByVal strStreetAddress, ByVal strCity,

ByVal strState, ByVal strZipCode)

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

'- Subprogram Name: InsertOwner -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Insert a record into-

'- the Owners table -

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

'- Parameter Dictionary (in parameter order): -

'-\*strConn - Connection string to database -

'-\*strTUID - OwnerID -

'-\*strFirstName - Owners first name -

'-\*strLastName - Onwers Last name -

'-\*strStreetAddress - Onwers Street Address -

'-\*strCity - Owners City -

'-\*strState - Owners State -

'-\*strZipCode - Owners ZipCode -

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

'- Local Variable Dictionary (alphabetically): -

'-\*DBConn - SQLConnection used to connect database -

'-\*DBCmd - SqlCommand for running sql code -

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

Dim DBConn As SqlConnection

Dim DBCmd As SqlCommand = New SqlCommand()

'Now try to open up a connection to the database

DBConn = New SqlConnection(strConn)

DBConn.Open()

'Insert to Owners Table

DBCmd.CommandText = "INSERT INTO Owners (TUID, FirstName, LastName,

StreetAddress, City, State, ZipCode) " &

"VALUES (" & strTUID & ", '" & strFirstName & "','" & strLastName & "','" & strStreetAddress & "',

'" & strCity & "', '" & strState & "', '" & strZipCode & "')"

DBCmd.Connection = DBConn

DBCmd.ExecuteNonQuery()

'Close the database connection

DBConn.Close()

End Sub

Private Sub InsertVehicle(ByVal strConn, ByVal strTUID, ByVal strOwnerID, ByVal strMake, ByVal strModel, ByVal strColor,

ByVal strModelYear, ByVal strVIN)

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

'- Subprogram Name: InsertVehicle -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Insert a record into-

'- the Vehicles table -

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

'- Parameter Dictionary (in parameter order): -

'-\*strConn - Connection string to database -

'-\*strTUID - VehicleID -

'-\*strOwnerID - Owners ID -

'-\*strMake - Vehicles Make -

'-\*strModel - Vehicles Model -

'-\*strColor - Vehicles Color -

'-\*strModelYear - Vehicles Model Year -

'-\*strVIN - Vehicles VIN -

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

'- Local Variable Dictionary (alphabetically): -

'-\*DBConn - SQLConnection used to connect database -

'-\*DBCmd - SqlCommand for running sql code -

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

Dim DBConn As SqlConnection

Dim DBCmd As SqlCommand = New SqlCommand()

'Now try to open up a connection to the database

DBConn = New SqlConnection(strConn)

DBConn.Open()

'Insert into Vehicles table

DBCmd.CommandText = "INSERT INTO Vehicles (TUID, OwnerID, Make, Model, Color, ModelYear, VIN) " &

"VALUES (" & strTUID & ", " & strOwnerID & ",'" & strMake & "','" & strModel & "',

'" & strColor & "', " & strModelYear & ", '" & strVIN & "')"

DBCmd.Connection = DBConn

DBCmd.ExecuteNonQuery()

'Close the database connection

DBConn.Close()

End Sub

Private Sub DeleteDatabase(ByVal strSERVERNAME As String, ByVal strDBNAME As String)

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

'- Subprogram Name: DeleteDatabase -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Delete a database -

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

'- Parameter Dictionary (in parameter order): -

'-\*strSERVERNAME - Name of the server -

'-\*strDBNAME - Name of the database -

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

'- Local Variable Dictionary (alphabetically): -

'-\*DBConn - SQLConnection used to connect database -

'-\*DBCommand - SqlCommand for running sql code -

'-\*strSQLCmd - String containing SQL Code -

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

'This routine shows how to delete a database completely

'from code. It does not consider deleting the data from

'the tables nor dropping the tables -- it just zaps the

'database completely

Dim DBConn As SqlConnection

Dim DBCommand As SqlCommand

Dim strSQLCmd As String

'We need to point back at the [Master] database itself

DBConn = New SqlConnection("Server=" & strSERVERNAME)

'Try to force single ownership of the database so that we have the

'permissions to delete it

strSQLCmd = "ALTER DATABASE [" & strDBNAME & "] SET " &

"SINGLE\_USER WITH ROLLBACK IMMEDIATE"

DBCommand = New SqlCommand(strSQLCmd, DBConn)

Try

DBConn.Open()

DBCommand.ExecuteNonQuery()

'MessageBox.Show("Database set for exclusive use", "",

'MessageBoxButtons.OK, MessageBoxIcon.Information)

Catch ex As Exception

MessageBox.Show(ex.ToString())

End Try

If (DBConn.State = ConnectionState.Open) Then

DBConn.Close()

End If

'Now drop the database

strSQLCmd = "DROP DATABASE " & strDBNAME

DBCommand = New SqlCommand(strSQLCmd, DBConn)

Try

DBConn.Open()

DBCommand.ExecuteNonQuery()

'MessageBox.Show("Database has been deleted", "", MessageBoxButtons.OK,

'MessageBoxIcon.Information)

Catch ex As Exception

MessageBox.Show(ex.ToString())

End Try

'Close connection to database

If (DBConn.State = ConnectionState.Open) Then

DBConn.Close()

End If

End Sub

#End Region

End Module

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

'- File Name : Form1.frm -

'- Part of Project: Assignment8 -

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

' Written By: Nathan VanSnepson -

' Written On: April 4, 2022 -

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

'- File Purpose: -

'- -

'- This file contains the form logic to allow a user to -

'- interact with a database -

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

'- Global Variable Dictionary (alphabetically): -

'-\*DBAdaptOwner - Used for Insert, Update, and Select on the-

'- owners table -

'-\*DBAdaptVehicle - Used for Insert, Update, and Select on -

'- the Vehicles table -

'-\*cmdBuilder - used to create insert statements for Owners -

'- table -

'-\*dsOwners - DataSet to hold results from Select query -

'-\*dsVehicles - DataSet to hold results from Select query -

'-\*intPrevOwner - Holds the value of the previous Owner -

'-\*strButtonClicked - Which button was clicked -

'-\*strDBPATH - Path of the database file -

'-\*strCONNECTION - Connection string to database -

'-\*myConn - Used for SQL code -

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

Imports System.Data.SqlClient

Public Class Form1

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

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

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

Const strDBNAME As String = "VehicleOwners" 'Name of the database

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

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

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

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

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

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

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

'SQLDataAdapters to run queries such as Insert, Update, and Select

Dim DBAdaptOwner As SqlDataAdapter

Dim DBAdaptVehicle As SqlDataAdapter

'Path to the database

Dim strDBPATH As String = My.Application.Info.DirectoryPath & "\" & strDBNAME & ".mdf"

'Connection string to connect to databse

Dim strCONNECTION As String = "SERVER=" & strServerName & ";DATABASE=" & strDBNAME &

";Integrated Security=SSPI;AttachDbFileName=" & strDBPATH

'DataSets to hold data from queries

Dim dsOwners As New DataSet

Dim dsVehicles As New DataSet

'Used to hold the location of the prev owner when adding a new record

Dim intPrevOwner As Integer

'String to hold which button was held ("ADD" OR "UPDATE")

Dim strButtonClicked As String

'SQLCommandBuilder to build Update command

Dim cmdBuilder As SqlCommandBuilder

'We'll also create a SqlConnection object since we will execute some

'straight SQL rather than relying on the DBAdapters...

Dim myConn As New SqlConnection(strCONNECTION)

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

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

'- Subprogram Name: Form1\_Load -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to get data from the -

'- owners and vehicles tables and populating the controls -

'- with the information. -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'-\*strSQLCmd - String containing SQL code -

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

Dim strSQLCmd As String

'Select All from Owners

strSQLCmd = "SELECT \* FROM Owners"

DBAdaptOwner = New SqlDataAdapter(strSQLCmd, strCONNECTION)

cmdBuilder = New SqlCommandBuilder(DBAdaptOwner)

DBAdaptOwner.InsertCommand = cmdBuilder.GetInsertCommand

DBAdaptOwner.DeleteCommand = cmdBuilder.GetDeleteCommand

dsOwners.Clear()

DBAdaptOwner.Fill(dsOwners, "Owners")

'Hide panel

pnlSaveRecord.Visible = False

'Set Text inputs to readonly

SetReadOnly(True)

'Set Bindings

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

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

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

txtAddress.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"))

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

'Sets the first Vehicle record

LoadVehicleRecord(CType(txtIDNumber.Text, Integer))

End Sub

Private Sub LoadVehicleRecord(ByVal intOwnerID)

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

'- Subprogram Name: LoadVehicleRecord -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to return data from -

'- the Vehicles table based on an OnwersID -

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

'- Parameter Dictionary (in parameter order): -

'-\*intOwnerID - TUID of the Owner -

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

'- Local Variable Dictionary (alphabetically): -

'-\*strSQLCmd - String containing SQL code -

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

Dim strSQLCmd As String

'Load vehicles because it does not change while program runs

strSQLCmd = "SELECT Make, Model, Color, ModelYear, VIN FROM Vehicles WHERE OwnerID = '" & intOwnerID & "'"

DBAdaptVehicle = New SqlDataAdapter(strSQLCmd, strCONNECTION)

dsVehicles.Clear()

DBAdaptVehicle.Fill(dsVehicles, "Vehicles")

'Updates the dataGridViews datasource

dgvVehicles.DataSource = dsVehicles.Tables("Vehicles")

End Sub

Private Sub SetReadOnly(ByVal blnValue As Boolean)

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

'- Subprogram Name: SetReadOnly -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to update the readonly -

'- property of textbox controls -

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

'- Parameter Dictionary (in parameter order): -

'-\*blnValue - Value to set readonly to -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

txtNameFirst.ReadOnly = blnValue

txtNameLast.ReadOnly = blnValue

txtAddress.ReadOnly = blnValue

txtCity.ReadOnly = blnValue

txtState.ReadOnly = blnValue

txtZipCode.ReadOnly = blnValue

End Sub

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

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

'- Subprogram Name: btnNextRec\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to go to the next -

'- record -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'Set the BindingConetext value to the next position

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

"Owners").Position + 1)

'Load Vehicle records for the Owner

LoadVehicleRecord(CType(txtIDNumber.Text, Integer))

End Sub

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

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

'- Subprogram Name: btnLastRec\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to go to the last -

'- record -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'Goes to the last record

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

"Owners").Count - 1)

'Load Vehicle records for the Owner

LoadVehicleRecord(CType(txtIDNumber.Text, Integer))

End Sub

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

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

'- Subprogram Name: btnPreviousRec\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to go to the previous -

'- record -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'Goes to the previous record

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

"Owners").Position - 1)

'Load Vehicle records for the Owner

LoadVehicleRecord(CType(txtIDNumber.Text, Integer))

End Sub

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

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

'- Subprogram Name: btnFirstRec\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to go to the first -

'- record -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'Goes to the first record

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

'Load Vehicle records for the Owner

LoadVehicleRecord(CType(txtIDNumber.Text, Integer))

End Sub

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

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

'- Subprogram Name: btnAdd\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Add an Owner record -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'Sets the previously displayed Owner

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

'Sets the button that was clicked to be Add

strButtonClicked = "Add"

'Ends the Current Edit and Adds a new record into dsOwners

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

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

'Set textboxes readonly to false

SetReadOnly(False)

'Set default values for textboxes

txtIDNumber.Text = CType(dsOwners.Tables("Owners").Compute("Max(TUID)", ""), Integer) + 1

txtNameFirst.Text = ""

txtNameLast.Text = ""

txtAddress.Text = ""

txtCity.Text = ""

txtState.Text = ""

txtZipCode.Text = ""

'Update dataGridViews dataSource

dgvVehicles.DataSource = Nothing

'Displays the Save and Cancel panel

pnlRecordButtons.Visible = False

pnlSaveRecord.Visible = True

End Sub

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

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

'- Subprogram Name: btnCancel\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Cancel Add or Update-

'- Operation -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'If the Add button was clicked remove the new record from dsOwners

If strButtonClicked = "Add" Then

BindingContext(dsOwners, "Owners").RemoveAt(dsOwners.Tables("Owners").Rows.Count)

End If

'Updates the binding source to the prvious Owner

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

LoadVehicleRecord(CType(txtIDNumber.Text, Integer))

'Sets textboxes to be readonly

SetReadOnly(True)

'Hides the save and cancel panel

pnlRecordButtons.Visible = True

pnlSaveRecord.Visible = False

'Update button clicked to be empty string

strButtonClicked = ""

End Sub

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

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

'- Subprogram Name: btnSave\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Save from the Add -

'- or Update Operation -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'If Add was clicked then insert record into Owner

If strButtonClicked = "Add" Then

InsertOwner(strCONNECTION, txtIDNumber.Text, txtNameFirst.Text, txtNameLast.Text,

txtAddress.Text, txtCity.Text, txtState.Text, txtZipCode.Text)

'If update was selected update the current user

ElseIf strButtonClicked = "Update" Then

'Stop any current edits.

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

Dim DBConn As SqlConnection

Dim DBCmd As SqlCommand = New SqlCommand()

'Now try to open up a connection to the database

DBConn = New SqlConnection(strCONNECTION)

DBConn.Open()

'Updates Owners Table

DBCmd.CommandText = "Update Owners " &

"Set FirstName = '" & txtIDNumber.Text & "', LastName = '" & txtNameLast.Text & "', " &

"StreetAddress = '" & txtAddress.Text & "', City = '" & txtCity.Text & "'," &

"State = '" & txtState.Text & "', ZipCode = '" & txtZipCode.Text & "'" &

"Where TUID = '" & txtIDNumber.Text & "'"

DBCmd.Connection = DBConn

DBCmd.ExecuteNonQuery()

'Update the dataset to correspond with database.

dsOwners.AcceptChanges()

End If

'Sets the button clicked to an empty String

strButtonClicked = ""

'Hides the Save and Cancel Panel

pnlRecordButtons.Visible = True

pnlSaveRecord.Visible = False

'Sets textboxes to be readonly

SetReadOnly(True)

'Loads Vehicle records with new OwnerID

LoadVehicleRecord(txtIDNumber.Text)

End Sub

Private Sub InsertOwner(ByVal strConn, ByVal strTUID, ByVal strFirstName, ByVal strLastName, ByVal strStreetAddress, ByVal strCity,

ByVal strState, ByVal strZipCode)

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

'- Subprogram Name: InsertOwner -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Insert a record into-

'- the Owners table -

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

'- Parameter Dictionary (in parameter order): -

'-\*strConn - Connection string to database -

'-\*strTUID - OwnerID -

'-\*strFirstName - Owners first name -

'-\*strLastName - Onwers Last name -

'-\*strStreetAddress - Onwers Street Address -

'-\*strCity - Owners City -

'-\*strState - Owners State -

'-\*strZipCode - Owners ZipCode -

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

'- Local Variable Dictionary (alphabetically): -

'-\*DBConn - SQLConnection used to connect database -

'-\*DBCmd - SqlCommand for running sql code -

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

Dim DBConn As SqlConnection

Dim DBCmd As SqlCommand = New SqlCommand()

'Now try to open up a connection to the database

DBConn = New SqlConnection(strConn)

DBConn.Open()

'Insert to Owners Table

DBCmd.CommandText = "INSERT INTO Owners (TUID, FirstName, LastName,

StreetAddress, City, State, ZipCode) " &

"VALUES (" & strTUID & ", '" & strFirstName & "','" & strLastName & "','" & strStreetAddress & "',

'" & strCity & "', '" & strState & "', '" & strZipCode & "')"

DBCmd.Connection = DBConn

DBCmd.ExecuteNonQuery()

'Close the database connection

DBConn.Close()

End Sub

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

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

'- Subprogram Name: btnUpdate\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Update an Owner -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'Sets textboxes readonly to be false

SetReadOnly(False)

'Sets button clicked to Update

strButtonClicked = "Update"

'Displays the Save and Cancel Panel

pnlRecordButtons.Visible = False

pnlSaveRecord.Visible = True

End Sub

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

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

'- Subprogram Name: btnDelete\_Click -

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

'- Written By: Nathan VanSnepson -

'- Written On: April 4, 2022 -

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

'- Subprogram Purpose: -

'- The purpose of this subprogram is to Delete and Owner -

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

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

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

'- Local Variable Dictionary (alphabetically): -

'-\*DBConn - SQL Connection to database -

'-\*DBCmd - SQL Command to run on database -

'-\*strSQLCmd - SQL Code -

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

Dim DBConn As SqlConnection

Dim DBCmd As SqlCommand = New SqlCommand()

Dim strSQLCmd As String

'Now try to open up a connection to the database

DBConn = New SqlConnection(strCONNECTION)

DBConn.Open()

'Confirm to delete the recorrd

If MessageBox.Show("Are you sure you want to delete this record?", "",

MessageBoxButtons.YesNo) = DialogResult.Yes Then

'Use SQL DML to zap the contents of the table

DBCmd.CommandText = "DELETE FROM Vehicles Where OwnerID = '" & txtIDNumber.Text & "'"

DBCmd.Connection = DBConn

DBCmd.ExecuteNonQuery()

'Use SQL DML to zap the contents of the table

DBCmd.CommandText = "DELETE FROM Owners Where TUID = '" & txtIDNumber.Text & "'"

DBCmd.Connection = DBConn

DBCmd.ExecuteNonQuery()

'Updates Owners

strSQLCmd = "SELECT \* FROM Owners"

DBAdaptOwner = New SqlDataAdapter(strSQLCmd, strCONNECTION)

dsOwners.Clear()

DBAdaptOwner.Fill(dsOwners, "Owners")

End If

'Closes the connection

DBConn.Close()

End Sub

End Class