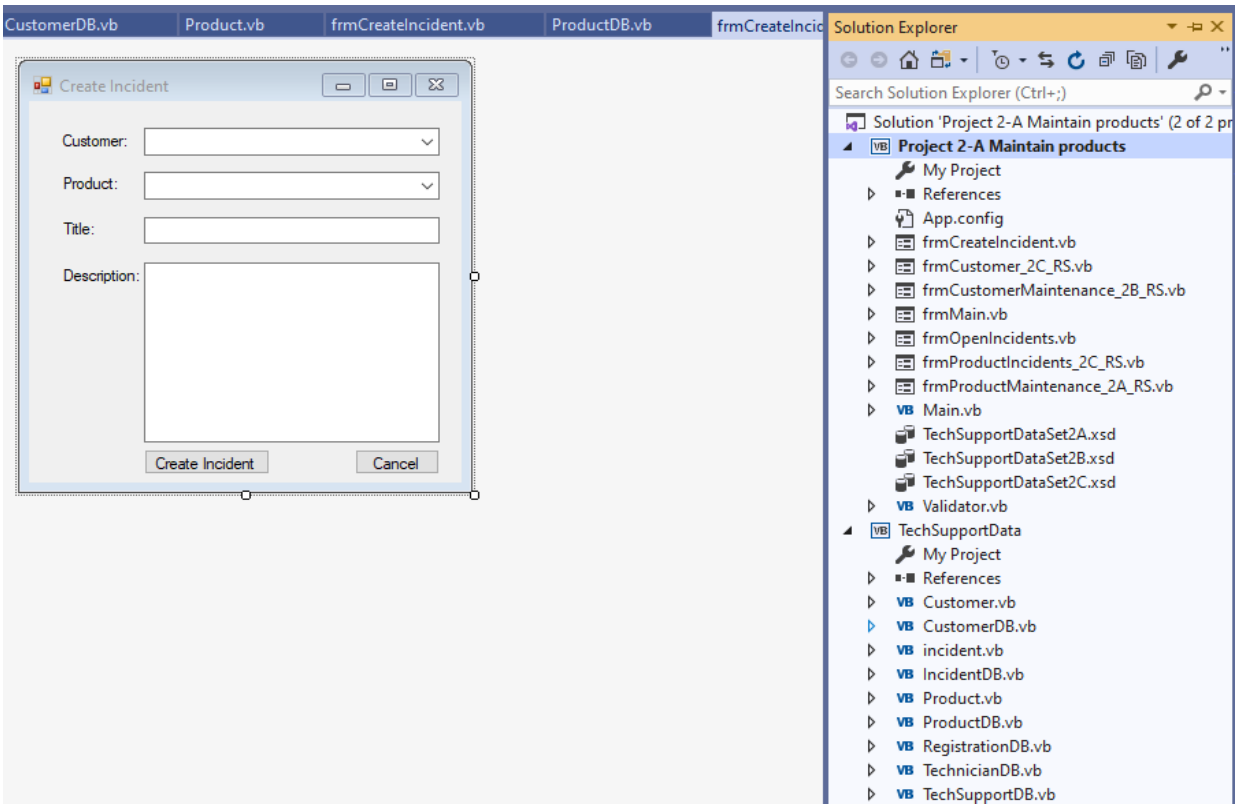
Raymond Sarfo

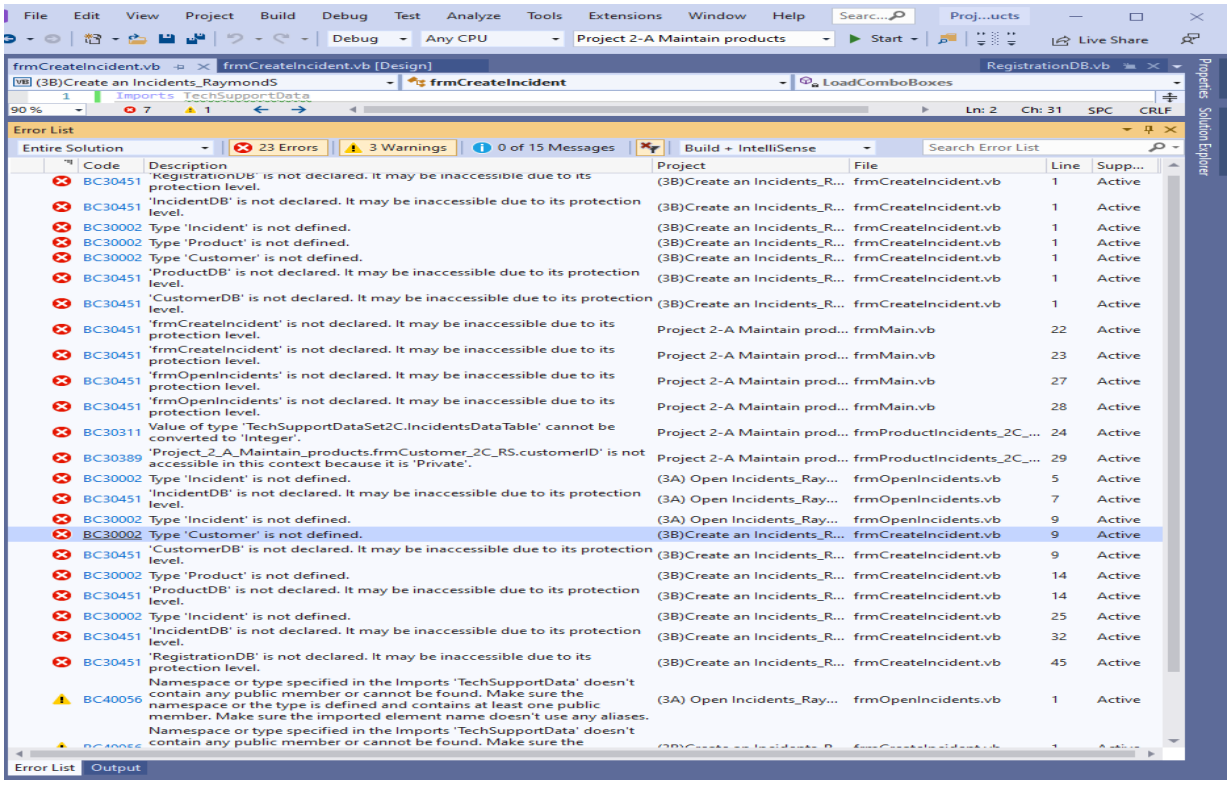
3B Create incident

3/26/20

Contents

FrmCreateIncident






```
Imports TechSupportData
Public Class frmCreateIncident
    Private Sub frmCreateIncident_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Me.LoadComboBoxes()
    End Sub

    Private Sub LoadComboBoxes()
        Try
            Dim customerList As List(Of Customer) = CustomerDB.GetCustomerList
            cboCustomers.DataSource = customerList
            cboCustomers.DisplayMember = "Name"
            cboCustomers.ValueMember = "CustomerID"

            Dim productList As List(Of Product) = ProductDB.GetProductList
            cboProducts.DataSource = productList
            cboProducts.DisplayMember = "Name"
            cboProducts.ValueMember = "ProductCode"
        Catch ex As Exception
            MessageBox.Show(ex.Message, ex.GetType.ToString)
        End Try
    End Sub

    Private Sub btnCreateIncident_Click(sender As Object, e As EventArgs) Handles
btnCreateIncident.Click
        If IsValidData() Then
            Dim incident As New Incident
            incident.CustomerID = CInt(cboCustomers.SelectedValue)
            incident.ProductCode = cboProducts.SelectedValue.ToString
            incident.DateOpened = DateTime.Today
            incident.Title = txtTitle.Text
            incident.Description = txtDescription.Text
```

```
            Try
                IncidentDB.AddIncident(incident)
                MessageBox.Show("The incident has been created.", "Incident Created",
                                MessageBoxButtons.OK, MessageBoxIcon.Information)
                Me.Close()
            Catch ex As Exception
                MessageBox.Show(ex.Message, ex.GetType.ToString)
            End Try
        End If
    End Sub

    Private Function IsValidData() As Boolean
        Dim customerID As Integer = CInt(cboCustomers.SelectedValue)
        Dim productCode As String = cboProducts.SelectedValue.ToString
        If RegistrationDB.ProductRegistered(customerID, productCode) Then
            If Validator.IsPresent(txtTitle, "Title") AndAlso
                    Validator.IsPresent(txtDescription, "Description") Then
                Return True
            Else
                Return False
            End If
        Else
            MessageBox.Show("This product is not registered to the customer you
selected.",
                            "Product Not Registered")
            cboCustomers.Select()
            Return False
        End If
    End Function

    Private Sub btnCancel_Click(sender As Object, e As EventArgs) Handles
btnCancel.Click
        Me.Close()
    End Sub
End Class
```

Validator Class

```
Public Class Validator
    Public Shared Function IsPresent(ByVal textBox As TextBox, ByVal name As String) As
Boolean
        If textBox.Text = "" Then
            MessageBox.Show(name & " is a required field.", "Entry Error")
            textBox.Select()
            Return True
        Else
            Return False
        End If
    End Function
    Public Shared Function IsInt32(ByVal textBox As TextBox, ByVal name As String) As
Boolean
        Try
            Convert.ToInt32(textBox.Text)
            Return True
        Catch ex As FormatException
            MessageBox.Show(name & "must be an integer value.", "Entry Error")
            textBox.Select()
            textBox.SelectAll()
```

```
            Return False
        End Try
    End Function
End Class
```

TechSupportData

Incident class

```
Imports System.Data.OleDb
Public Class incident
    Private m_IncidentID As Integer
    Private m_CustomerID As Integer
    Private m_ProductCode As String
    Private m_TechID As Nullable(Of Integer)
    Private m_DateOpened As Date
    Private m_DateClosed As Nullable(Of Date)
    Private m_Title As String
    Private m_Description As String
    Private m_CustomerName As String
    Private m_TechnicianName As String

    Public Sub New()

    End Sub
    Public Property IncidentID() As Integer
        Get
            Return m_IncidentID
        End Get
        Set(value As Integer)
            m_IncidentID = value
        End Set
    End Property

    Public Property CustomerID() As Integer
        Get
            Return m_CustomerID
        End Get
        Set(value As Integer)
            m_CustomerID = value
        End Set
    End Property

    Public Property ProductCode() As String
        Get
            Return m_ProductCode
        End Get
        Set(value As String)
            m_ProductCode = value
        End Set
    End Property
    Public Property TechID() As Nullable(Of Integer)
        Get
            If m_TechID.HasValue Then
                Return CInt(m_TechID)
            Else
```

```
            Return Nothing
        End If
    End Get
    Set(value As Nullable(Of Integer))
        m_TechID = value
    End Set
End Property
Public Property DateOpened() As Date
    Get
        Return m_DateOpened
    End Get
    Set(value As Date)
        m_DateOpened = value
    End Set
End Property
Public Property DateClosed() As Nullable(Of Date)
    Get
        If m_DateClosed.HasValue Then
            Return (m_DateClosed)
        Else
            Return Nothing
        End If
    End Get
    Set(value As Nullable(Of Date))
        m_DateClosed = value
    End Set
End Property
Public Property Title() As String
    Get
        Return m_Title
    End Get
    Set(value As String)
        m_Title = value
    End Set
End Property
Public Property Description() As String
    Get
        Return m_Description

    End Get
    Set(value As String)
        m_Description = value
    End Set
End Property
Public ReadOnly Property CustomerName() As String
    Get
        Dim name As String = ""
        If m_CustomerID <> 0 Then
            Try
                name = CustomerDB.GetCustomerName(m_CustomerID)
            Catch ex As Exception
                Throw ex
            End Try
        End If
        Return name
    End Get
End Property
Public ReadOnly Property TechnicianName() As String
```

```
        Get
            Dim name As String = ""
            If m_TechID.HasValue Then
                Try
                    name = TechnicianDB.GetTechnicianName(CInt(m_TechID))
                Catch ex As Exception
                    Throw ex
                End Try
            End If
            Return name
        End Get
    End Property

End Class
```

Customer class

```
Public Class Customer
    Private m_CustomerID As Integer
    Private m_Name As String

    Public Sub New()

    End Sub
    Public Property CustomerID() As Integer
        Get
            Return m_CustomerID
        End Get
        Set(value As Integer)
            m_CustomerID = value
        End Set
    End Property

    Public Property Name() As String
        Get
            Return m_Name
        End Get
        Set(value As String)
            m_Name = value
        End Set
    End Property
End Class
```

Product class

```
Public Class Product
    Private m_ProductCode As String
    Private m_Name As String

    Public Sub New()

    End Sub
    Public Property ProductCode() As String
        Get
            Return m_ProductCode
```

```
        End Get
        Set(value As String)
            m_ProductCode = value
        End Set
    End Property
    Public Property Name() As String
        Get
            Return m_Name
        End Get
        Set(value As String)
            m_Name = value
        End Set
    End Property
End Class
```

TechSupportDB

```
Imports System.Data.OleDb
Public Class TechSupportDB
    Public Shared Function GetConnection() As OleDbConnection
        Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;
        Data Source = C:\Bob\TechSupport.ACCDB"
        Return New OleDbConnection(connectionString)
    End Function

End Class
```

IncidentDB

```
Imports System.Data.OleDb
Public Class IncidentDB
    Public Shared Sub AddIncident(ByVal incident As incident)
        Dim connection As OleDbConnection = TechSupportDB.GetConnection
        Dim insertStatement As String =
            "INSERT Into Incidents" &
            "(CustomerID, ProductCode, DateOpened, Title, Description) " &
            "VALUES (@CustomerID, @ProductCode, @DateOpened, @Title, @Description)"
        Dim insertCommand As New OleDbCommand(insertStatement, connection)
        insertCommand.Parameters.AddWithValue("@CustomerID", incident.CustomerID)
        insertCommand.Parameters.AddWithValue("@productCode", incident.ProductCode)
        insertCommand.Parameters.AddWithValue("@DateOpened", incident.DateOpened)
        insertCommand.Parameters.AddWithValue("@Title", incident.Title)
        insertCommand.Parameters.AddWithValue("@Description", incident.Description)
        Try
            connection.Open()
            insertCommand.ExecuteNonQuery()
        Catch ex As OleDbException
            Throw ex
        Finally
            connection.Close()
        End Try
    End Sub
End Class
```

CustomerDB

```
Imports System.Data.OleDb
Public Class CustomerDB
   Public Shared Function GetCustomerList() As List(Of Customer)
        Dim customerList As New List(Of Customer)
        Dim connection As OleDbConnection = TechSupportDB.GetConnection
        Dim selectStatement As String =
            "SELECT CustomerID, Name FROM Customers " &
            "ORDER BY Name "
        Dim selectCommand As New OleDbCommand(selectStatement, connection)
        Try
            connection.Open()
            Dim reader As OleDbDataReader = selectCommand.ExecuteReader
            Dim customer As Customer
            Do While reader.Read
                customer = New Customer
                customer.CustomerID = CInt(reader("CustomerID"))
                customer.Name = reader("Name").ToString
                customerList.Add(customer)
            Loop
            reader.Close()
        Catch ex As OleDbException
            Throw ex
        Finally
            connection.Close()
        End Try
        Return customerList
    End Function
End Class
```

ProductDB

```
Imports System.Data.OleDb
Public Class ProductDB
    Public Shared Function GetProductList() As List(Of Product)
        Dim productList As New List(Of Product)
        Dim connection As OleDbConnection = TechSupportDB.GetConnection
        Dim selectStatement As String =
            "SELECT ProductCode, Name FROM Products " &
            "ORDER BY Name "
        Dim selectCommand As New OleDbCommand(selectStatement, connection)
        Try
            connection.Open()
            Dim reader As OleDbDataReader = selectCommand.ExecuteReader
            Dim product As Product
            Do While reader.Read
                product = New Product
                product.ProductCode = reader("ProductCode").ToString
                product.Name = reader("Name").ToString
                productList.Add(product)
            Loop
            reader.Close()
        Catch ex As OleDbException
            Throw ex
        Finally
            connection.Close()
```

```
        End Try
        Return productList
    End Function
End Class
```

RegistrationDB

```
Imports System.Data.OleDb
Public Class RegistrationDB
    Public Shared Function ProductRegistered(ByVal customerID As Integer, ByVal
productCode As String) As Boolean
        Dim count As Integer = 0
        Dim connection As OleDbConnection = TechSupportDB.GetConnection
        Dim selectStatement As String =
            "SELECT COUNT(*) FROM Registrations " &
            "WHERE CustomerID = @CustomerID AND ProductCode = @ProductCode"
        Dim selectCommand As New OleDbCommand(selectStatement, connection)
        selectCommand.Parameters.AddWithValue("@CustomerID", customerID)
        selectCommand.Parameters.AddWithValue("@ProductCode", productCode)
        Try
            connection.Open()
            count = CInt(selectCommand.ExecuteScalar)
        Catch ex As OleDbException
            Throw ex
        Finally
            connection.Close()
        End Try
        If count > 0 Then
            Return True
        Else
            Return False
        End If
    End Function
End Class
```