2021

Project 3-D_YK

YAW KOOSONO

Course: CIS232-40 .NET Programming II

Instructor: Bob Desilets Spring 2021 [v2 02/17/21]

Program specifications

This project displays all open incidents of a technician. Users need to select a technician from dropdown and software will display email and phone number of the technician along with all open incident assigned with the selected technician.

It is an offline desktop software which needs OS WIN 2007 and higher also need OLEDB driver installed to run.

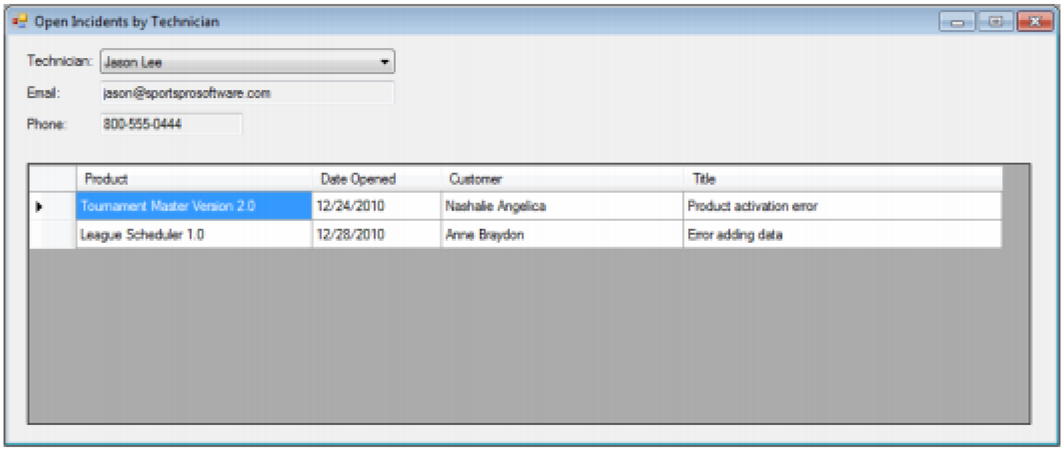
Users need to store database file (TechSupport.mdb) into C:\Bob location of same PC where the software has installed.

Hardware: Computer/Laptop, Mouse, Keyboard

Project 3-D: Display open incidents by technician

For this project, you'll enhance the SportsPro application by adding a form that displays the open incidents for a selected technician. To do that, you'll use two object data sources. (*Required reading: chapters 6, 7, and 9.*)

The Open Incidents by Technician form



Technician: Jason Lee

Email: jason@sportsprosoftware.com

Phone: 800-555-0444

	Product	Date Opened	Customer	Title
▸	Tournament Master Version 2.0	12/24/2010	Nashalie Angelica	Product activation error
	League Scheduler 1.0	12/28/2010	Anne Braydon	Error adding data

SportsPro project item

Name	Description
frmTechnicianIncidents	A form that lets the user display the open incidents for a technician.

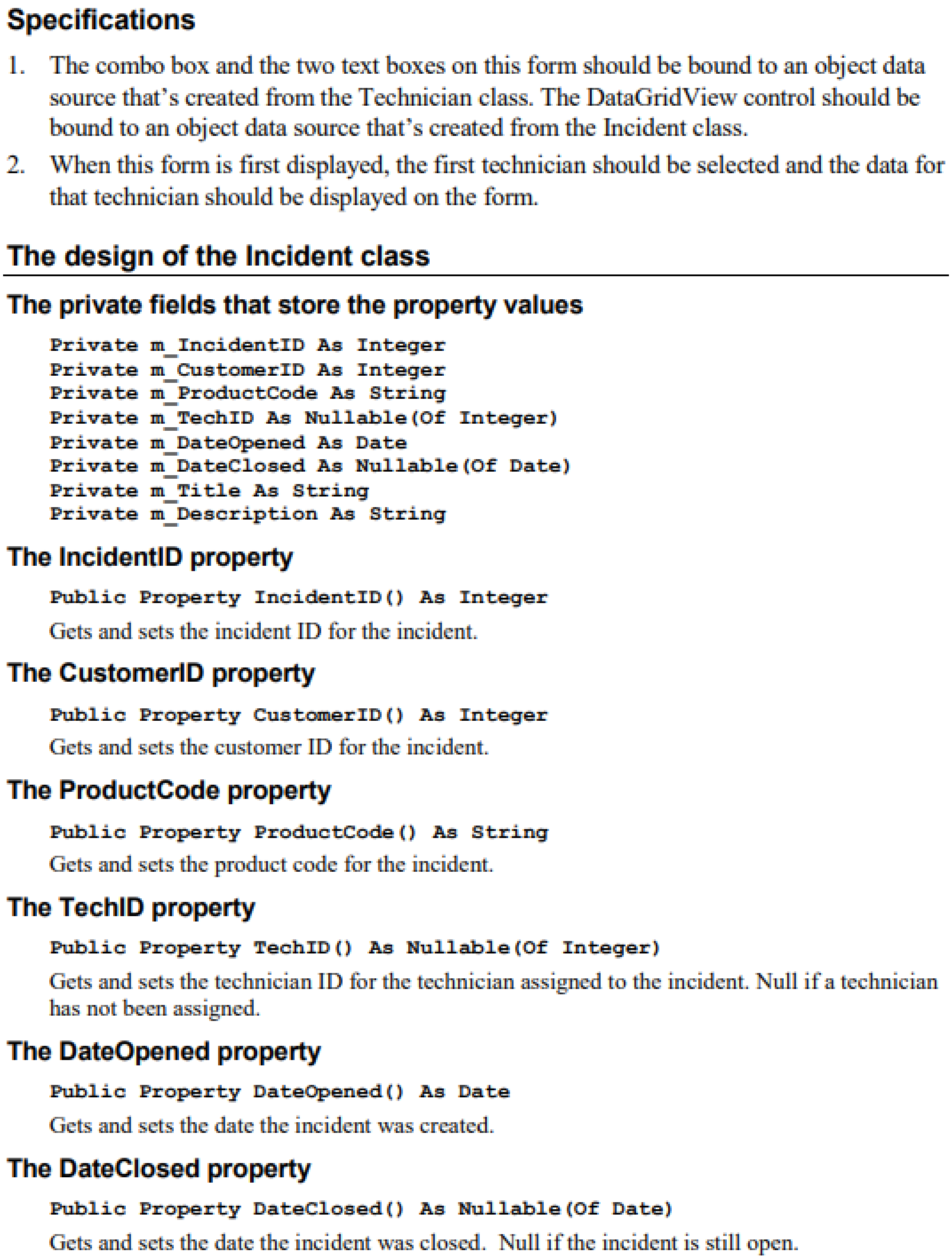
TechSupportData project items

Name	Description
Technician	A business class that represents a single technician.
Incident	A business class that represents a single incident.
TechSupportDB	A database class that contains a method that returns a connection object for the TechSupport database.
TechnicianDB	A database class that contains methods for working with the Technicians table in the TechSupport database.
IncidentDB	A database class that contains methods for working with the Incidents table in the TechSupport database.
ProductDB	A database class that contains methods for working with the Products table in the TechSupport database.
CustomerDB	A database class that contains methods for working with the Customers table in the TechSupport database.

Operation

- The Open Incidents by Technician form should be displayed when the user chooses the Incidents→Display Open Incidents by Technician command from the menu on the main form.
- To display the open incidents for a technician, the user selects the technician from the combo box. In addition to the incidents, contact information for the selected technician is displayed on the form.

Table of Contents



Specifications

1. The combo box and the two text boxes on this form should be bound to an object data source that's created from the Technician class. The DataGridView control should be bound to an object data source that's created from the Incident class.
2. When this form is first displayed, the first technician should be selected and the data for that technician should be displayed on the form.

The design of the Incident class

The private fields that store the property values

```
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
```

The IncidentID property

```
Public Property IncidentID() As Integer
```

Gets and sets the incident ID for the incident.

The CustomerID property

```
Public Property CustomerID() As Integer
```

Gets and sets the customer ID for the incident.

The ProductCode property

```
Public Property ProductCode() As String
```

Gets and sets the product code for the incident.

The TechID property

```
Public Property TechID() As Nullable(Of Integer)
```

Gets and sets the technician ID for the technician assigned to the incident. Null if a technician has not been assigned.

The DateOpened property

```
Public Property DateOpened() As Date
```

Gets and sets the date the incident was created.

The DateClosed property

```
Public Property DateClosed() As Nullable(Of Date)
```

Gets and sets the date the incident was closed. Null if the incident is still open.

..........51

The Title property

```
Public Property Title() As String
```

Gets and sets the title for the incident.

The Description property

```
Public Property Description() As String
```

Gets and sets the description for the incident.

The CustomerName property

```
Public ReadOnly Property CustomerName() As String
```

Gets the name for the customer associated with the incident using the GetCustomerName method of the CustomerDB class.

The ProductName property

```
Public ReadOnly Property ProductName() As String
```

Gets the name for the product associated with the incident using the GetProductName method of the ProductDB class.

The design of the Technician class

The private fields that store the property values

```
    Private m_TechID As Integer
    Private m_Name As String
    Private m_Email As String
    Private m_Phone As String
```

The TechID property

```
Public Property TechID() As Integer
```

Gets and sets the ID for the technician.

The Name property

```
Public Property Name() As String
```

Gets and sets the name for the technician.

The Email property

```
Public Property Email() As String
```

Gets and sets the email address for the technician.

The Phone property

```
Public Property Phone() As String
```

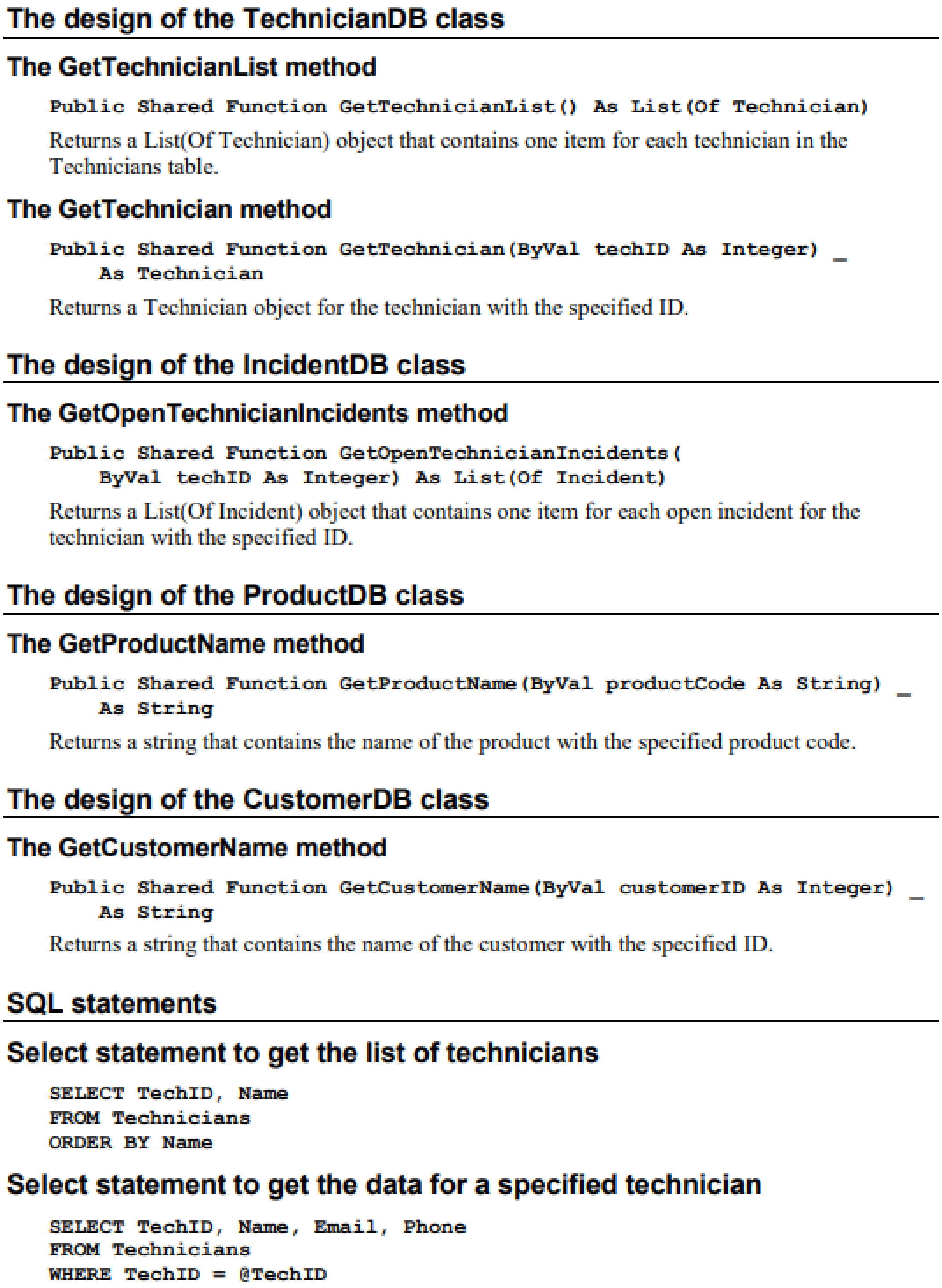
Gets and sets the phone number for the technician.

The design of the TechSupportDB class

The GetConnection method

```
Public Shared Function GetConnection() As SqlConnection
```

Returns a SqlConnection object that establishes a connection to the TechSupport database.



The design of the TechnicianDB class

The GetTechnicianList method

```
Public Shared Function GetTechnicianList() As List(Of Technician)
```

Returns a List(Of Technician) object that contains one item for each technician in the Technicians table.

The GetTechnician method

```
Public Shared Function GetTechnician(ByVal techID As Integer) _
    As Technician
```

Returns a Technician object for the technician with the specified ID.

The design of the IncidentDB class

The GetOpenTechnicianIncidents method

```
Public Shared Function GetOpenTechnicianIncidents(
    ByVal techID As Integer) As List(Of Incident)
```

Returns a List(Of Incident) object that contains one item for each open incident for the technician with the specified ID.

The design of the ProductDB class

The GetProductName method

```
Public Shared Function GetProductName(ByVal productCode As String) _
    As String
```

Returns a string that contains the name of the product with the specified product code.

The design of the CustomerDB class

The GetCustomerName method

```
Public Shared Function GetCustomerName(ByVal customerID As Integer) _
    As String
```

Returns a string that contains the name of the customer with the specified ID.

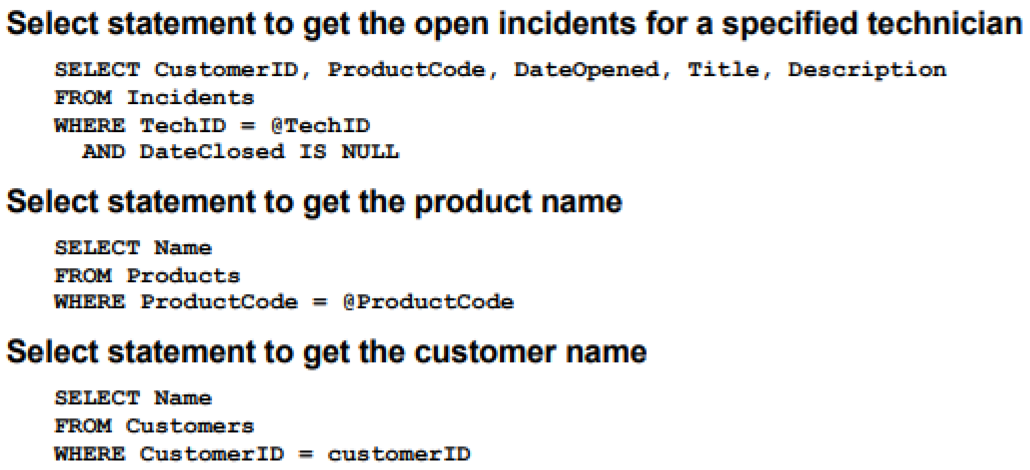
SQL statements

Select statement to get the list of technicians

```
SELECT TechID, Name
FROM Technicians
ORDER BY Name
```

Select statement to get the data for a specified technician

```
SELECT TechID, Name, Email, Phone
FROM Technicians
WHERE TechID = @TechID
```



Select statement to get the open incidents for a specified technician

```
SELECT CustomerID, ProductCode, DateOpened, Title, Description
FROM Incidents
WHERE TechID = @TechID
  AND DateClosed IS NULL
```

Select statement to get the product name

```
SELECT Name
FROM Products
WHERE ProductCode = @ProductCode
```

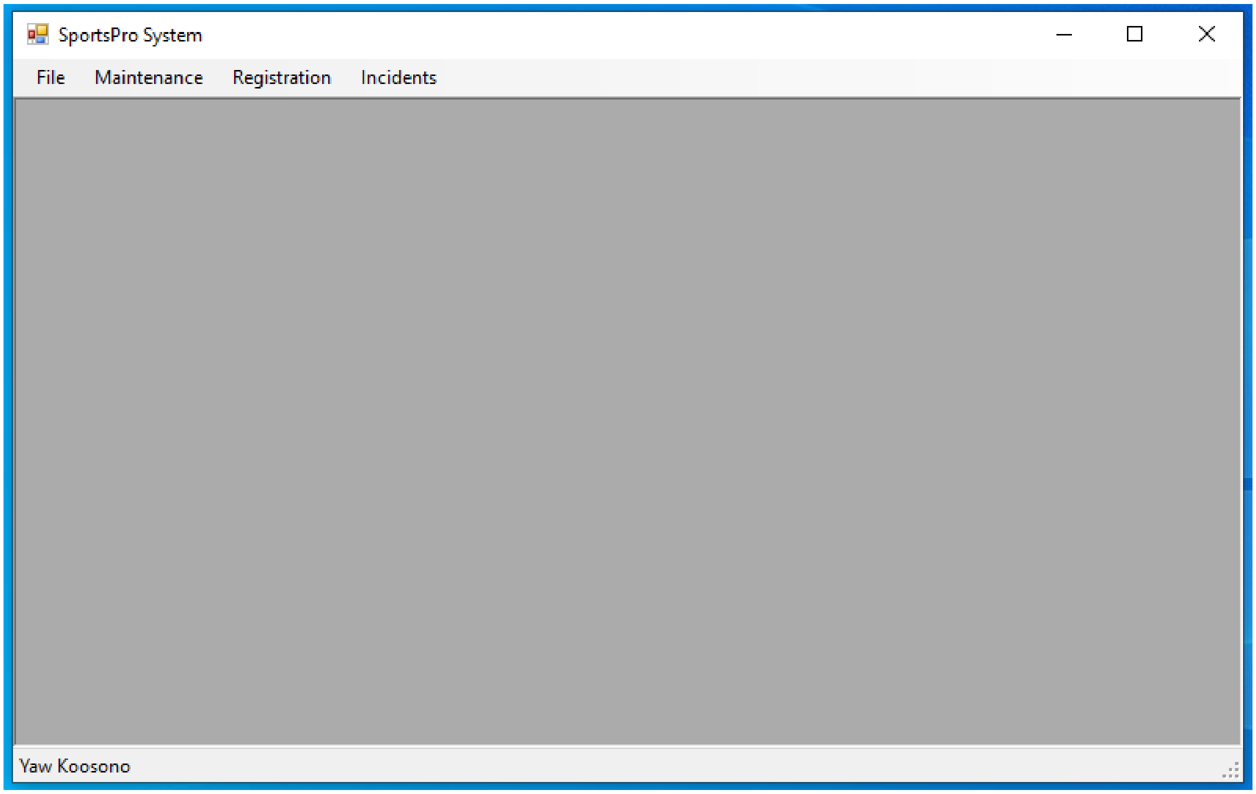
Select statement to get the customer name

```
SELECT Name
FROM Customers
WHERE CustomerID = customerID
```

...............54

A walk through of the application..54

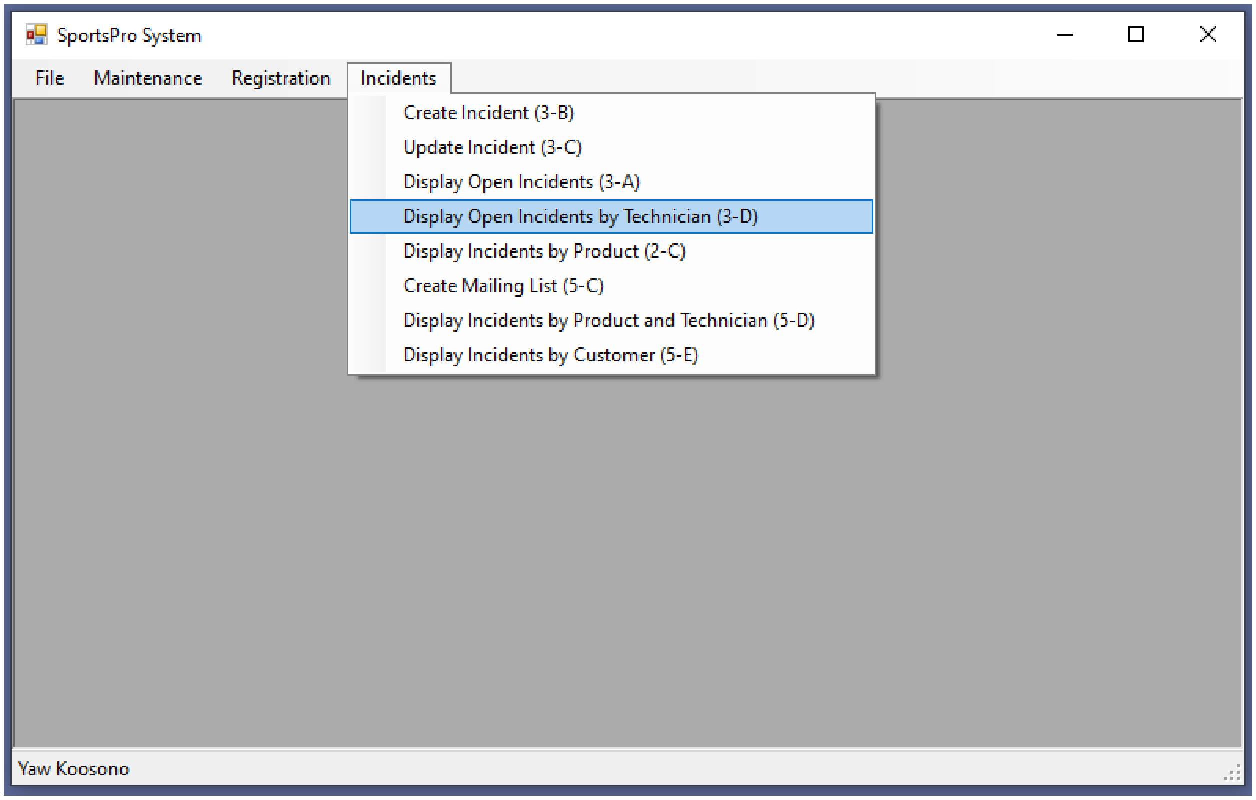
Main form display when user run the application. All sub forms can navigate through main form. 54



....55

Fig-1: Main form when program start...55

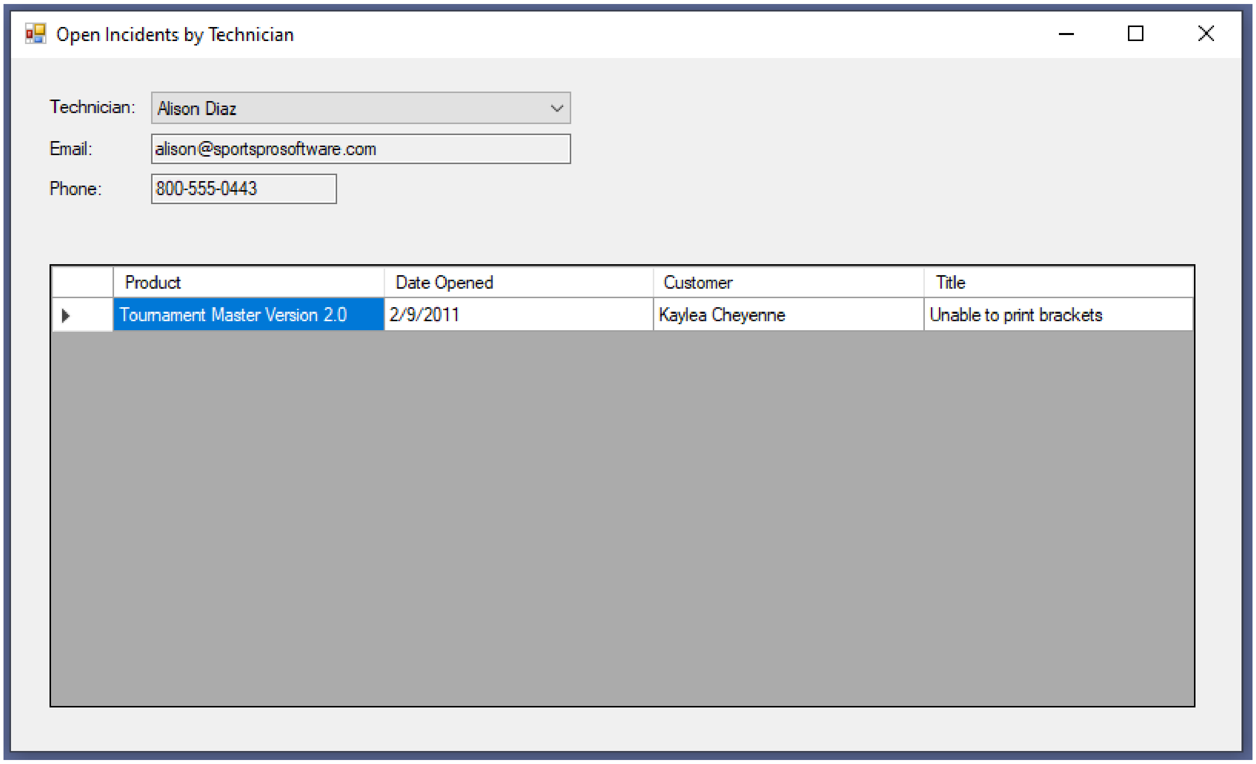
Displaying the navigation menu for Display open incidents by technician form. 55


SportsPro System
File
Maintenance
Registration
Incidents
Create Incident (3-B)
Update Incident (3-C)
Display Open Incidents (3-A)
Display Open Incidents by Technician (3-D)
Display Incidents by Product (2-C)
Create Mailing List (5-C)
Display Incidents by Product and Technician (5-D)
Display Incidents by Customer (5-E)
Yaw Koosono

....56

Fig-2: Main form with Display open Incidents by technician navigation..............................56

After open the form showing first selected technician with linked open incidents. 56


Open Incidents by Technician

Technician: Alison Diaz

Email: alison@sportsprosoftware.com

Phone: 800-555-0443

Product	Date Opened	Customer	Title
Tournament Master Version 2.0	2/9/2011	Kaylea Cheyenne	Unable to print brackets

....57

Fig-3: Display open incidents by technician form 57

Source Code..57

Classes of TechSupportData class library 57

Code of TechSupportDB Class (for partial 3D)..57

Imports System.Data.OLEDB 57

Public Class TechSupportDB..57

Public Shared Function GetConnection() As OleDbConnection 57

Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Bob\TechSupport.mdb;Persist Security Info=True" 57

Return New OleDbConnection(connString)...58

End Function..58

End Class..58

Code of Technician Class (for 3D) 58

Public Class Technician...58

Private m_TechID As Integer..58

Private m_Name As String 58

Private m_Email As String...58

Private m_Phone As String 58

Public Property TechID() As Integer 58
Get....58
Return m_TechID 58
End Get 58
Set(ByVal value As Integer)....58
m_TechID = value....58
End Set....58
End Property 58
Public Property Name() As String 58
Get....58
Return m_Name 58
End Get 58
Set(ByVal value As String) 58
m_Name = value....58
End Set....58
End Property 58
Public Property Email() As String....58
Get....58
Return m_Email....58
End Get 58
Set(ByVal value As String) 58
m_Email = value 58
End Set....58
End Property 58
Public Property Phone() As String....58
Get....58
Return m_Phone....59
End Get 59
Set(ByVal value As String) 59
m_Phone = value 59
End Set....59
End Property 59
End Class....59

Code of Incident Class .... **(for 3D)** 59

Imports System.Data.OLEDB 59

Public Class Incident 59

Private m_IncidentID As Integer 59

Private m_CustomerID As Integer 59

Private m_ProductCode As String 59

Private m_TechID As Nullable(Of Integer) 59

Private m_DateOpened As Date 59

Private m_DateClosed As Nullable(Of Date) 59

Private m_Title As String 59

Private m_Description As String 59

Public Sub New() 59

End Sub...59

Public Property IncidentID() As Integer 59

Get..59

Return m_IncidentID 59

End Get 59

Set(ByVal value As Integer)..59

m_IncidentID = value..59

End Set..59

End Property 59

Public Property CustomerID() As Integer 59

Get..59

Return m_CustomerID 59

End Get 59

Set(ByVal value As Integer)..59

m_CustomerID = value..59

End Set..60

End Property 60

Public Property ProductCode() As String 60

Get..60

Return m_ProductCode 60

End Get 60

Set(value As String) 60
m_ProductCode = value 60
End Set..60
End Property 60
Public Property TechID() As Nullable(Of Integer)..60
Get..60
If m_TechID.HasValue Then 60
Return CInt(m_TechID)...60
Else 60
Return Nothing..60
End If 60
End Get 60
Set(value As Nullable(Of Integer))..60
m_TechID = value..60
End Set..60
End Property 60
Public Property DateOpened() As Date 60
Get..60
Return m_DateOpened..60
End Get 60
Set(value As Date)...60
m_DateOpened = value 60
End Set..60
End Property 60
Public Property DateClosed() As Nullable(Of Date)..60
Get..60
If m_DateClosed.HasValue Then 60
Return CDate(m_DateClosed) 60
Else 60
Return Nothing..61
End If 61
End Get 61
Set(value As Nullable(Of Date)) 61

m_DateClosed = value 61
End Set 61
End Property 61
Public Property Title() As String 61
Get 61
Return m_Title 61
End Get 61
Set(value As String) 61
m_Title = value 61
End Set 61
End Property 61
Public Property Description() As String 61
Get 61
Return m_Description 61
End Get 61
Set(value As String) 61
m_Description = value 61
End Set 61
End Property 61
Public ReadOnly Property CustomerName() As String 61
Get 61
Dim name As String = "" 61
If m_CustomerID <> 0 Then 61
Try 61
name = CustomerDB.GetCustomerName(m_CustomerID) 61
Catch ex As Exception 61
Throw ex 61
End Try 61
End If 61
Return name 61
End Get 61
End Property 62
Public ReadOnly Property TechName() As String 62

Get 62
Dim name As String = "" 62
If m_TechID.HasValue And m_TechID <> 0 Then 62
Try 62
name = TechnicianDB.GetTechnicianName(CInt(m_TechID)) 62
Catch ex As Exception 62
Throw ex 62
End Try 62
End If 62
Return name 62
End Get 62
End Property 62
Public ReadOnly Property ProductName() As String 62
Get 62
Dim name As String = "" 62
If m_ProductCode <> "" Then 62
Try 62
name = ProductDB.GetProductName(m_ProductCode) 62
Catch ex As Exception 62
Throw ex 62
End Try 62
End If 62
Return name 62
End Get 62
End Property 62
End Class 62
Code of Customer Class (for partial 3D) 62
Public Class Customer 62
Private m_CustomerID As Integer 62
Private m_Name As String 62
Public Property CustomerID() As Integer 62
Get 62
Return m_CustomerID 63

End Get 63
Set(ByVal value As Integer) 63
m_CustomerID = value 63
End Set 63
End Property 63
Public Property Name() As String 63
Get 63
Return m_Name 63
End Get 63
Set(ByVal value As String) 63
m_Name = value 63
End Set 63
End Property 63
End Class 63
Code of Product Class (for partial 3D) 63
Public Class Product 63
Private m_ProductCode As String 63
Private m_Name As String 63
Public Property ProductCode() As String 63
Get 63
Return m_ProductCode 63
End Get 63
Set(ByVal value As String) 63
m_ProductCode = value 63
End Set 63
End Property 63
Public Property Name() As String 63
Get 63
Return m_Name 63
End Get 63
Set(ByVal value As String) 63
m_Name = value 63
End Set 63

End Property 63
End Class 63
Code of TechnicianDB Class (for 3D) 64
Imports System.Data.OleDb 64
Public Class TechnicianDB 64
Public Shared Function GetTechnicianName(ByVal p_TechID As Integer) As String.............64
Dim name As String 64
Dim conn As OleDbConnection = TechSupportDB.GetConnection......................................64
Dim selectQuery As String = "SELECT Name FROM Technicians WHERE TechID = " & p_TechID 64
Dim selectCmd As New OleDbCommand(selectQuery, conn)...64
Try 64
conn.Open()..64
name = selectCmd.ExecuteScalar.ToString 64
Catch ex As Exception 64
Throw ex 64
Finally..64
conn.Close()..64
End Try..64
Return name 64
End Function 64
Public Shared Function GetTechnicianList() As List(Of Technician) 64
Dim technicianList As New List(Of Technician)..64
Dim conn As OleDbConnection = TechSupportDB.GetConnection......................................64
Dim selectQuery As String = "SELECT TechID, Name From Technicians Order By Name".....64
Dim selectCmd As New OleDbCommand(selectQuery, conn)...64
Try 64
conn.Open()..64
Dim reader As OleDbDataReader = selectCmd.ExecuteReader..64
Dim technician As Technician...64
Do While reader.Read 64
technician = New Technician 64
technician.TechID = CInt(reader("TechID")) 64
technician.Name = reader("Name").ToString 64

technicianList.Add(technician) 65
Loop 65
reader.Close() 65
Catch ex As Exception 65
Throw ex 65
Finally 65
conn.Close() 65
End Try 65
Return technicianList 65
End Function 65
Public Shared Function GetTechnician(ByVal techID As Integer) As Technician 65
Dim technician As Technician = Nothing 65
Dim conn As OleDbConnection = TechSupportDB.GetConnection 65
Dim selectQuery As String = "SELECT TechID, Name, Email, Phone FROM Technicians WHERE TechID = @TechID" 65
Dim selectCmd As New OleDbCommand(selectQuery, conn) 65
selectCmd.Parameters.AddWithValue("@TechID", techID) 65
Try 65
conn.Open() 65
Dim reader As OleDbDataReader = selectCmd.ExecuteReader 65
If reader.Read Then 65
Technician = New Technician 65
Technician.TechID = CInt(reader("TechID")) 65
technician.Name = reader("Name").ToString 65
technician.Email = reader("Email").ToString 65
technician.Phone = reader("Phone").ToString 65
End If 65
reader.Close() 65
Catch ex As Exception 65
Throw ex 65
Finally 65
conn.Close() 65
End Try 65
Return technician 65

End Function 66
End Class 66
Code of IncidentDB Class (for 3D) 66
Imports System.Data.OleDb 66
Public Class IncidentDB 66
Public Shared Function GetOpenIncidents() As List(Of Incident) 66
Dim incidentList As New List(Of Incident) 66
Dim connection As OleDbConnection = TechSupportDB.GetConnection 66
Dim selectQuery = "SELECT CustomerID, ProductCode, TechID, DateOpened, Title " &.......66
"FROM Incidents " & 66
"WHERE DateClosed IS NULL" 66
Dim selectCmd As New OleDbCommand(selectQuery, connection) 66
Try 66
connection.Open() 66
Dim reader As OleDbDataReader = selectCmd.ExecuteReader 66
Dim incident As Incident..66
Do While reader.Read 66
incident = New Incident 66
incident.CustomerID = CInt(reader("CustomerID"))..66
incident.ProductCode = reader("ProductCode").ToString 66
If IsDBNull(reader("TechID")) Then 66
incident.TechID = Nothing 66
Else 66
incident.TechID = CInt(reader("TechID")) 66
End If 66
incident.DateOpened = CDate(reader("DateOpened")) 66
incident.Title = reader("Title").ToString 66
incidentList.Add(incident) 66
Loop 66
reader.Close() 66
Catch ex As Exception 66
Throw ex 66
Finally..66

connection.Close() 66

End Try 67

Return incidentList 67

End Function 67

Public Shared Sub AddIncident(ByVal p_Incident As Incident) 67

Dim connection As OleDbConnection = TechSupportDB.GetConnection 67

Dim insertQuery = "INSERT INTO Incidents " & 67

"(CustomerID, ProductCode, DateOpened, Title, Description) " & 67

"VALUES (@CustomerID, @ProductCode, @DateOpened, @Title, @Description)" 67

Dim insertCmd As New OleDbCommand(insertQuery, connection) 67

Try 67

insertCmd.Parameters.AddWithValue("@CustomerID", p_Incident.CustomerID) 67

insertCmd.Parameters.AddWithValue("@ProductCode", p_Incident.ProductCode) 67

insertCmd.Parameters.AddWithValue("@DateOpened", CDate(DateTime.Today)) 67

insertCmd.Parameters.AddWithValue("@Title", p_Incident.Title) 67

insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description) 67

connection.Open() 67

insertCmd.ExecuteNonQuery() 67

Catch ex As Exception 67

Throw ex 67

Finally 67

connection.Close() 67

End Try 67

End Sub 67

Public Shared Function GetIncident(ByVal p_IncidentID As Integer) As Incident 67

Dim incident As Incident = Nothing 67

Dim connection As OleDbConnection = TechSupportDB.GetConnection 67

Dim selectQuery = "SELECT IncidentID, CustomerID, ProductCode, TechID, " & 67

"DateOpened, DateClosed, Title, Description " & 67

"From Incidents Where IncidentID = @IncidentID" 67

Dim selectCmd As New OleDbCommand(selectQuery, connection) 67

selectCmd.Parameters.AddWithValue("@IncidentID", p_IncidentID) 67

Try 68

connection.Open() 68
Dim reader As OleDbDataReader = selectCmd.ExecuteReader 68
If reader.Read Then 68
incident = New Incident 68
incident.IncidentID = CInt(reader("IncidentID")) 68
incident.CustomerID = CInt(reader("CustomerID")) 68
incident.ProductCode = reader("ProductCode").ToString 68
If IsDBNull(reader("TechID")) Then 68
incident.TechID = Nothing 68
Else 68
incident.TechID = CInt(reader("TechID")) 68
End If 68
incident.DateOpened = CDate(reader("DateOpened")) 68
If IsDBNull(reader("DateClosed")) Then 68
incident.DateClosed = Nothing 68
Else 68
incident.DateClosed = CDate(reader("DateClosed")) 68
End If 68
incident.Title = reader("Title").ToString 68
incident.Description = reader("Description").ToString 68
End If 68
reader.Close() 68
Catch ex As Exception 68
Throw ex 68
Finally 68
connection.Close() 68
End Try 68
Return incident 68
End Function 68
Public Shared Function UpdateIncident(ByVal p_Incident As Incident, ... 68
ByVal p_Description As String) As Boolean 68
Dim isUpdated As Boolean = False 68
Dim connection As OleDbConnection = TechSupportDB.GetConnection 68

Dim insertQuery = "UPDATE Incidents SET Description = @NewDescription " & 69

"WHERE IncidentID = @IncidentID " & 69

"And Description = @Description " &................69

"And DateClosed Is NULL" 69

Dim insertCmd As New OleDbCommand(insertQuery, connection) 69

Try 69

insertCmd.Parameters.AddWithValue("@NewDescription", p_Description) 69

insertCmd.Parameters.AddWithValue("@IncidentID", p_Incident.IncidentID) 69

insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description) 69

connection.Open() 69

If insertCmd.ExecuteNonQuery() > 0 Then 69

isUpdated = True 69

End If 69

Catch ex As Exception 69

Throw ex 69

Finally................69

connection.Close() 69

End Try................69

Return isUpdated................69

End Function 69

Public Shared Function CloseIncident(ByVal p_Incident As Incident) As Boolean 69

Dim isUpdated As Boolean = False 69

Dim connection As OleDbConnection = TechSupportDB.GetConnection 69

Dim insertQuery = "UPDATE Incidents SET DateClosed = @DateClosed " & 69

"WHERE IncidentID = @IncidentID " & 69

"And Description = @Description " &................69

"And DateClosed Is NULL" 69

Dim insertCmd As New OleDbCommand(insertQuery, connection) 69

Try 70

insertCmd.Parameters.AddWithValue("@DateClosed", CDate(DateTime.Today)) 70

insertCmd.Parameters.AddWithValue("@IncidentID", p_Incident.IncidentID) 70

insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description) 70

connection.Open() 70

If insertCmd.ExecuteNonQuery() > 0 Then 70

isUpdated = True 70

End If 70

Catch ex As Exception 70

Throw ex 70

Finally 70

connection.Close() 70

End Try 70

Return isUpdated 70

End Function 70

Public Shared Function GetOpenTechnicianIncidents(ByVal techID As Integer) As List(Of Incident) 70

Dim incidentList As New List(Of Incident) 70

Dim connection As OleDbConnection = TechSupportDB.GetConnection 70

Dim selectQuery = "SELECT CustomerID, ProductCode, TechID, DateOpened, Title " &....... 70

"FROM Incidents " & 70

"WHERE TechID = @TechID AND DateClosed IS NULL" 70

Dim selectCmd As New OleDbCommand(selectQuery, connection) 70

selectCmd.Parameters.AddWithValue("@TechID", techID) 70

Try 70

connection.Open() 70

Dim reader As OleDbDataReader = selectCmd.ExecuteReader 70

Dim incident As Incident 70

Do While reader.Read 70

incident = New Incident 70

incident.CustomerID = CInt(reader("CustomerID")) 70

incident.ProductCode = reader("ProductCode").ToString 70

incident.TechID = CInt(reader("TechID")) 70

incident.DateOpened = CDate(reader("DateOpened")) 70

incident.Title = reader("Title").ToString 71

incidentList.Add(incident) 71

Loop 71

reader.Close() 71

Catch ex As Exception 71

Throw ex 71
Finally...71
connection.Close() 71
End Try...71
Return incidentList 71
End Function 71
End Class 71
Code of CustomerDB Class (for partial 3D)...71
Imports System.Data.OleDb 71
Public Class CustomerDB 71
Public Shared Function GetCustomerName(ByVal p_CustomerID As Integer) As String 71
Dim name As String 71
Dim conn As OleDbConnection = TechSupportDB.GetConnection 71
Dim selectQuery As String = "SELECT Name FROM Customers WHERE CustomerID = " & p_CustomerID 71
Dim selectCmd As New OleDbCommand(selectQuery, conn)...71
Try 71
conn.Open()..71
name = selectCmd.ExecuteScalar.ToString 71
Catch ex As Exception 71
Throw ex 71
Finally...71
conn.Close()...71
End Try...71
Return name 71
End Function 71
Public Shared Function GetCustomerList() As List(Of Customer)..71
Dim customerList As New List(Of Customer) 71
Dim conn As OleDbConnection = TechSupportDB.GetConnection 71
Dim selectQuery As String = "SELECT CustomerID, Name FROM Customers ORDER BY Name" 72
Dim selectCmd As New OleDbCommand(selectQuery, conn)...72
Try 72
conn.Open()..72

Dim reader As OleDbDataReader = selectCmd.ExecuteReader 72
Dim customer As Customer 72
Do While reader.Read 72
customer = New Customer 72
customer.CustomerID = CInt(reader("CustomerID"))...72
customer.Name = reader("Name").ToString...72
customerList.Add(customer) 72
Loop...72
reader.Close() 72
Catch ex As Exception 72
Throw ex 72
Finally...72
conn.Close()..72
End Try...72
Return customerList 72
End Function 72
End Class 72
Code of ProductDB Class (for partial 3D)...72
Imports System.Data.OleDb 72
Public Class ProductDB...72
Public Shared Function GetProductName(ByVal p_ProductCode As String) As String 72
Dim name As String 72
Dim conn As OleDbConnection = TechSupportDB.GetConnection 72
Dim selectQuery As String = "SELECT Name FROM Products WHERE ProductCode = @ProductCode" 72
Dim selectCmd As New OleDbCommand(selectQuery, conn)..72
selectCmd.Parameters.AddWithValue("@ProductCode", p_ProductCode) 73
Try 73
conn.Open()..73
name = selectCmd.ExecuteScalar.ToString 73
Catch ex As Exception 73
Throw ex 73
Finally...73
conn.Close()..73

End Try..73
Return name 73
End Function 73
Public Shared Function GetProductList() As List(Of Product) 73
Dim productList As New List(Of Product) 73
Dim conn As OleDbConnection = TechSupportDB.GetConnection.......................................73
Dim selectQuery As String = "SELECT ProductCode, Name FROM Products ORDER BY Name" 73
Dim selectCmd As New OleDbCommand(selectQuery, conn)...73
Try 73
conn.Open()...73
Dim reader As OleDbDataReader = selectCmd.ExecuteReader...73
Dim product As Product..73
Do While reader.Read 73
product = New Product 73
product.ProductCode = reader("ProductCode").ToString 73
product.Name = reader("Name").ToString 73
productList.Add(product)...73
Loop..73
reader.Close() 73
Catch ex As Exception...73
Throw ex 73
Finally..73
conn.Close()...73
End Try...73
Return productList 74
End Function 74
End Class...74
Code of RegistrationDB Class 74
Imports System.Data.OleDb 74
Public Class RegistrationDB..74
Public Shared Function ProductRegistered(ByVal p_CustomerID As Integer, ...74
ByVal p_ProductCode As String) As Boolean...74
Dim isRegistered As Boolean = False 74

Dim connection As OleDbConnection = TechSupportDB.GetConnection 74
Dim selectQuery As String = "SELECT Count(*) FROM Registrations " &............................74
"WHERE CustomerID = @CustomerID AND ProductCode = @ProductCode" 74
Dim selectCmd As New OleDbCommand(selectQuery, connection) 74
Try 74
selectCmd.Parameters.AddWithValue("@CustomerID", p_CustomerID) 74
selectCmd.Parameters.AddWithValue("@ProductCode", p_ProductCode) 74
connection.Open() 74
Dim count As Integer = selectCmd.ExecuteScalar 74
If count > 0 Then 74
isRegistered = True 74
End If 74
Catch ex As Exception 74
Throw ex 74
Finally...74
connection.Close() 74
End Try...74
Return isRegistered..74
End Function 75
End Class 75
Classes of SportsPro project 75
Code of frmTechnicianIncidents_YK Class 75
Imports TechSupportData 75
Public Class frmTechnicianIncidents_YK...75
Private techList As List(Of Technician) 75
Private Sub frmTechnicianIncidents_YK_Load(sender As Object, e As EventArgs) Handles MyBase.Load 75
Try 75
techList = TechnicianDB.GetTechnicianList..75
technicianComboBox.DataSource = techList...75
Catch ex As Exception 75
MessageBox.Show(ex.Message, ex.GetType.ToString) 75
End Try...75
End Sub 75

Private Sub technicianComboBox_SelectedIndexChanged(sender As Object, e As EventArgs) Handles technicianComboBox.SelectedIndexChanged 75

Try 75

If technicianComboBox.SelectedIndex >= 0 Then 75

Dim techId = technicianComboBox.SelectedValue 75

Dim technician As Technician = TechnicianDB.GetTechnician(techId)................................75

TechnicianBindingSource.Clear() 75

TechnicianBindingSource.Add(technician) 75

Dim incidentList As List(Of Incident) = IncidentDB.GetOpenTechnicianIncidents(techId)......75

IncidentBindingSource.DataSource = incidentList 75

End If 75

Catch ex As Exception...75

MessageBox.Show(ex.Message, ex.GetType.ToString) 75

End Try...75

End Sub..75

End Class 75

Added code of frmMain_YK Class only for Project 3-D 75

Private Sub DisplayOpenIncidentsByTechnician3DToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles DisplayOpenIncidentsByTechnician3DToolStripMenuItem.Click 76

Dim frmTechIncidents As New frmTechnicianIncidents_YK...76

frmTechIncidents.Show() 76

End Sub...76

Specifications

1. The combo box and the two text boxes on this form should be bound to an object data source that's created from the Technician class. The DataGridView control should be bound to an object data source that's created from the Incident class.
2. When this form is first displayed, the first technician should be selected and the data for that technician should be displayed on the form.

The design of the Incident class

The private fields that store the property values

```
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
```

The IncidentID property

```
Public Property IncidentID() As Integer
```

Gets and sets the incident ID for the incident.

The CustomerID property

```
Public Property CustomerID() As Integer
```

Gets and sets the customer ID for the incident.

The ProductCode property

```
Public Property ProductCode() As String
```

Gets and sets the product code for the incident.

The TechID property

```
Public Property TechID() As Nullable(Of Integer)
```

Gets and sets the technician ID for the technician assigned to the incident. Null if a technician has not been assigned.

The DateOpened property

```
Public Property DateOpened() As Date
```

Gets and sets the date the incident was created.

The DateClosed property

```
Public Property DateClosed() As Nullable(Of Date)
```

Gets and sets the date the incident was closed. Null if the incident is still open.

The Title property

```
Public Property Title() As String
```

Gets and sets the title for the incident.

The Description property

```
Public Property Description() As String
```

Gets and sets the description for the incident.

The CustomerName property

```
Public ReadOnly Property CustomerName() As String
```

Gets the name for the customer associated with the incident using the GetCustomerName method of the CustomerDB class.

The ProductName property

```
Public ReadOnly Property ProductName() As String
```

Gets the name for the product associated with the incident using the GetProductName method of the ProductDB class.

The design of the Technician class

The private fields that store the property values

```
    Private m_TechID As Integer
    Private m_Name As String
    Private m_Email As String
    Private m_Phone As String
```

The TechID property

```
Public Property TechID() As Integer
```

Gets and sets the ID for the technician.

The Name property

```
Public Property Name() As String
```

Gets and sets the name for the technician.

The Email property

```
Public Property Email() As String
```

Gets and sets the email address for the technician.

The Phone property

```
Public Property Phone() As String
```

Gets and sets the phone number for the technician.

The design of the TechSupportDB class

The GetConnection method

```
Public Shared Function GetConnection() As SqlConnection
```

Returns a SqlConnection object that establishes a connection to the TechSupport database.

The design of the TechnicianDB class

The GetTechnicianList method

```
Public Shared Function GetTechnicianList() As List(Of Technician)
```

Returns a List(Of Technician) object that contains one item for each technician in the Technicians table.

The GetTechnician method

```
Public Shared Function GetTechnician(ByVal techID As Integer) _
    As Technician
```

Returns a Technician object for the technician with the specified ID.

The design of the IncidentDB class

The GetOpenTechnicianIncidents method

```
Public Shared Function GetOpenTechnicianIncidents(
    ByVal techID As Integer) As List(Of Incident)
```

Returns a List(Of Incident) object that contains one item for each open incident for the technician with the specified ID.

The design of the ProductDB class

The GetProductName method

```
Public Shared Function GetProductName(ByVal productCode As String) _
    As String
```

Returns a string that contains the name of the product with the specified product code.

The design of the CustomerDB class

The GetCustomerName method

```
Public Shared Function GetCustomerName(ByVal customerID As Integer) _
    As String
```

Returns a string that contains the name of the customer with the specified ID.

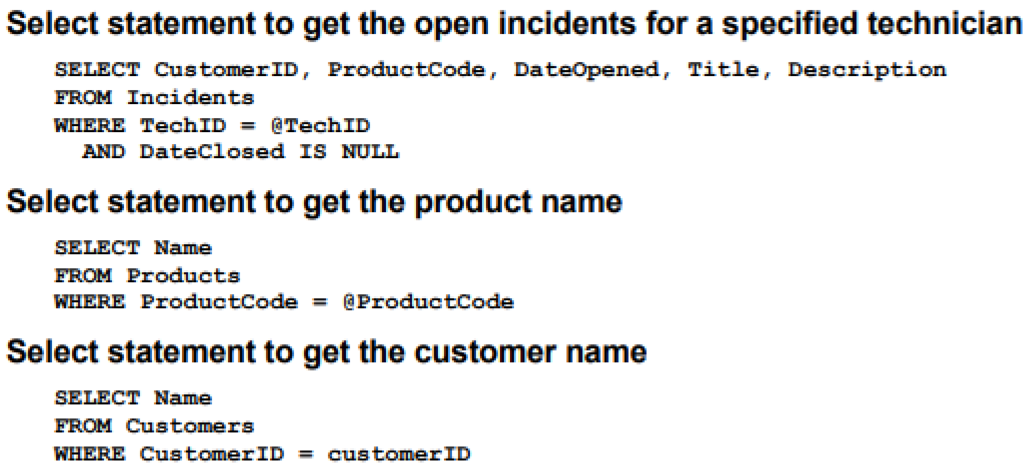
SQL statements

Select statement to get the list of technicians

```
SELECT TechID, Name
FROM Technicians
ORDER BY Name
```

Select statement to get the data for a specified technician

```
SELECT TechID, Name, Email, Phone
FROM Technicians
WHERE TechID = @TechID
```



Select statement to get the open incidents for a specified technician

```
SELECT CustomerID, ProductCode, DateOpened, Title, Description
FROM Incidents
WHERE TechID = @TechID
  AND DateClosed IS NULL
```

Select statement to get the product name

```
SELECT Name
FROM Products
WHERE ProductCode = @ProductCode
```

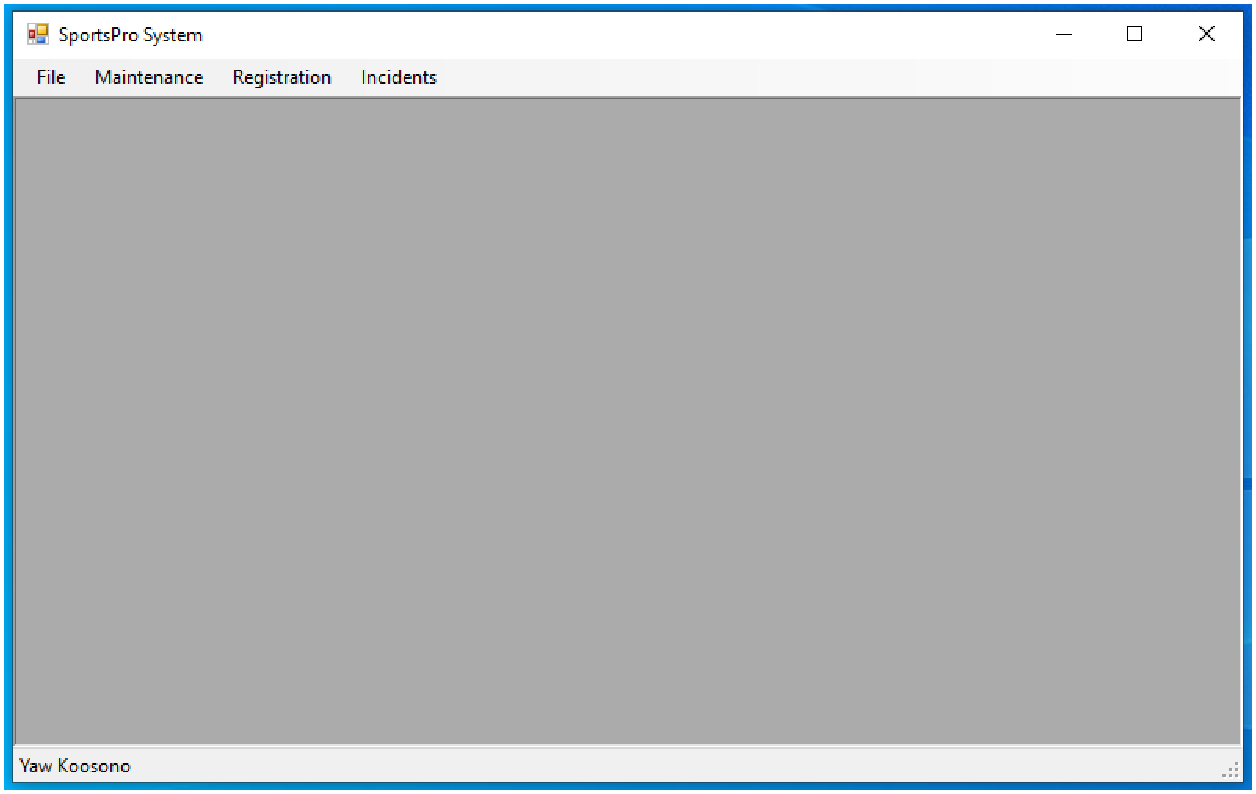
Select statement to get the customer name

```
SELECT Name
FROM Customers
WHERE CustomerID = customerID
```

...............54

A walk through of the application 54

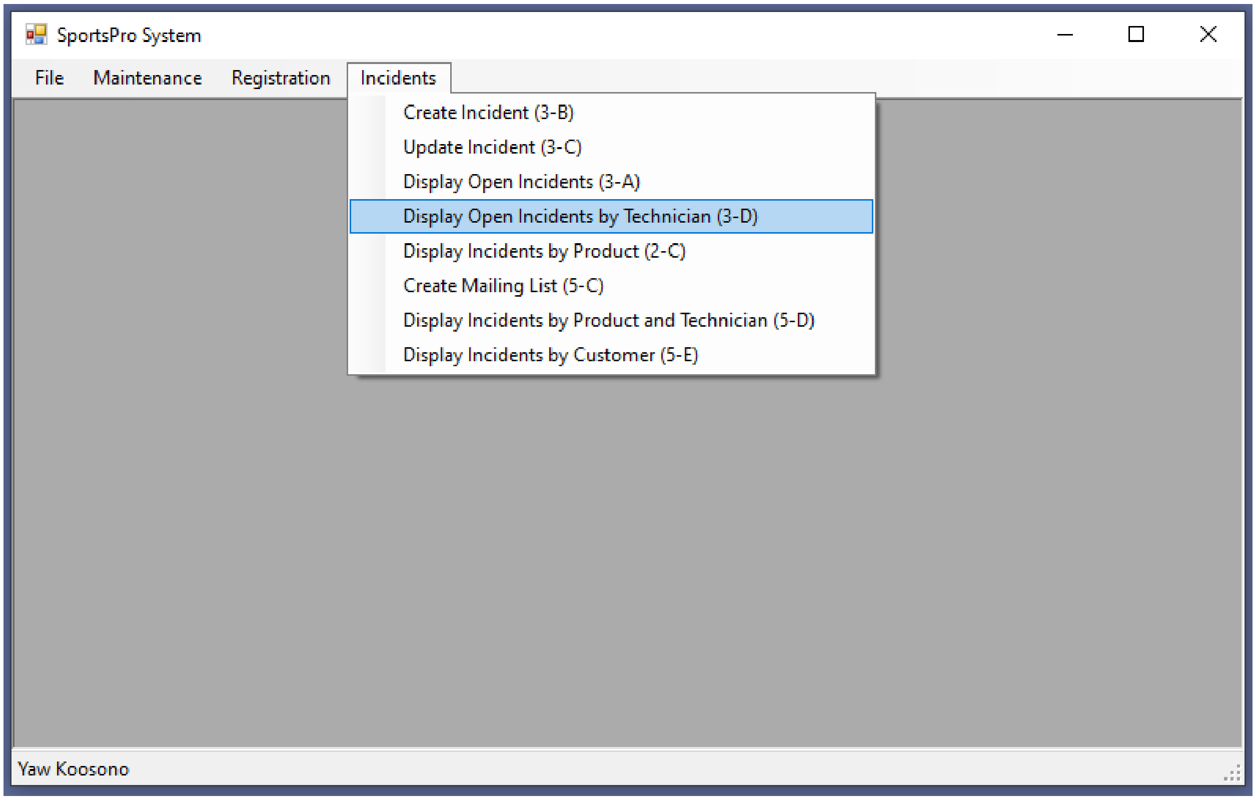
Main form display when user run the application. All sub forms can navigate through main form. 54



....55

Fig-1: Main form when program start...55

Displaying the navigation menu for Display open incidents by technician form. 55

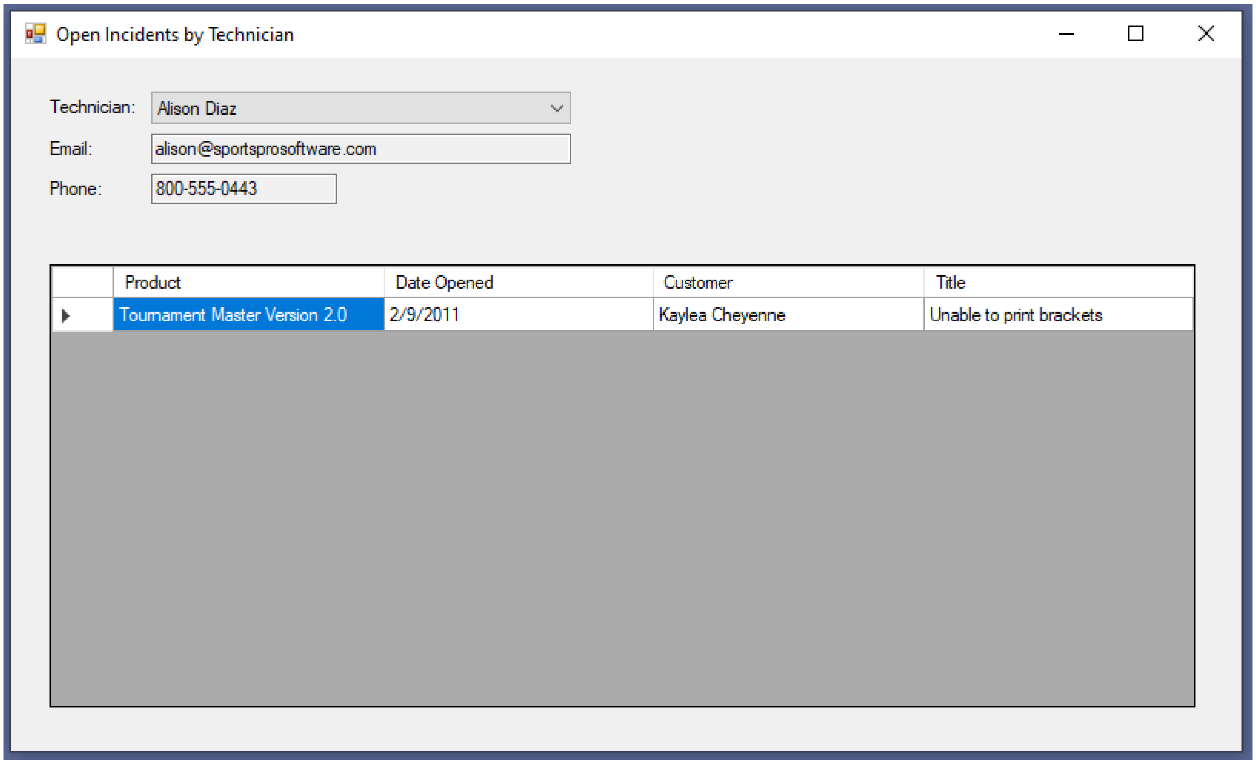


SportsPro System
File
Maintenance
Registration
Incidents
Create Incident (3-B)
Update Incident (3-C)
Display Open Incidents (3-A)
Display Open Incidents by Technician (3-D)
Display Incidents by Product (2-C)
Create Mailing List (5-C)
Display Incidents by Product and Technician (5-D)
Display Incidents by Customer (5-E)
Yaw Koosono

....56

Fig-2: Main form with Display open Incidents by technician navigation..............................56

After open the form showing first selected technician with linked open incidents. 56



Open Incidents by Technician

Technician: Alison Diaz

Email: alison@sportsprosoftware.com

Phone: 800-555-0443

	Product	Date Opened	Customer	Title
▶	Tournament Master Version 2.0	2/9/2011	Kaylea Cheyenne	Unable to print brackets

....57

Fig-3: Display open incidents by technician form 57

Source Code..57

Classes of TechSupportData class library 57

Code of TechSupportDB Class (for partial 3D)..................................57

Imports System.Data.OLEDB 57

Public Class TechSupportDB..57

Public Shared Function GetConnection() As OleDbConnection 57

Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Bob\TechSupport.mdb;Persist Security Info=True" 57

Return New OleDbConnection(connString)..58

End Function 58

End Class...58

Code of Technician Class (for 3D) 58

Public Class Technician...58

Private m_TechID As Integer...58

Private m_Name As String 58

Private m_Email As String...58

Private m_Phone As String 58

Public Property TechID() As Integer 58
Get 58
Return m_TechID 58
End Get 58
Set(ByVal value As Integer) 58
m_TechID = value 58
End Set 58
End Property 58
Public Property Name() As String 58
Get 58
Return m_Name 58
End Get 58
Set(ByVal value As String) 58
m_Name = value 58
End Set 58
End Property 58
Public Property Email() As String 58
Get 58
Return m_Email 58
End Get 58
Set(ByVal value As String) 58
m_Email = value 58
End Set 58
End Property 58
Public Property Phone() As String 58
Get 58
Return m_Phone 59
End Get 59
Set(ByVal value As String) 59
m_Phone = value 59
End Set 59
End Property 59
End Class 59

Code of Incident Class .... **(for 3D)** 59
Imports System.Data.OLEDB 59
Public Class Incident 59
Private m_IncidentID As Integer 59
Private m_CustomerID As Integer 59
Private m_ProductCode As String 59
Private m_TechID As Nullable(Of Integer) 59
Private m_DateOpened As Date 59
Private m_DateClosed As Nullable(Of Date) 59
Private m_Title As String 59
Private m_Description As String 59
Public Sub New() 59
End Sub 59
Public Property IncidentID() As Integer 59
Get 59
Return m_IncidentID 59
End Get 59
Set(ByVal value As Integer) 59
m_IncidentID = value 59
End Set 59
End Property 59
Public Property CustomerID() As Integer 59
Get 59
Return m_CustomerID 59
End Get 59
Set(ByVal value As Integer) 59
m_CustomerID = value 59
End Set 60
End Property 60
Public Property ProductCode() As String 60
Get 60
Return m_ProductCode 60
End Get 60

Set(value As String) 60
m_ProductCode = value 60
End Set 60
End Property 60
Public Property TechID() As Nullable(Of Integer) 60
Get 60
If m_TechID.HasValue Then 60
Return CInt(m_TechID) 60
Else 60
Return Nothing 60
End If 60
End Get 60
Set(value As Nullable(Of Integer)) 60
m_TechID = value 60
End Set 60
End Property 60
Public Property DateOpened() As Date 60
Get 60
Return m_DateOpened 60
End Get 60
Set(value As Date) 60
m_DateOpened = value 60
End Set 60
End Property 60
Public Property DateClosed() As Nullable(Of Date) 60
Get 60
If m_DateClosed.HasValue Then 60
Return CDate(m_DateClosed) 60
Else 60
Return Nothing 61
End If 61
End Get 61
Set(value As Nullable(Of Date)) 61

m_DateClosed = value 61
End Set..61
End Property 61
Public Property Title() As String 61
Get..61
Return m_Title 61
End Get 61
Set(value As String) 61
m_Title = value 61
End Set..61
End Property 61
Public Property Description() As String..61
Get..61
Return m_Description 61
End Get 61
Set(value As String) 61
m_Description = value 61
End Set..61
End Property 61
Public ReadOnly Property CustomerName() As String..61
Get..61
Dim name As String = "" 61
If m_CustomerID <> 0 Then..61
Try 61
name = CustomerDB.GetCustomerName(m_CustomerID) 61
Catch ex As Exception..61
Throw ex 61
End Try..61
End If 61
Return name 61
End Get 61
End Property 62
Public ReadOnly Property TechName() As String 62

Get 62
Dim name As String = "" 62
If m_TechID.HasValue And m_TechID <> 0 Then 62
Try 62
name = TechnicianDB.GetTechnicianName(CInt(m_TechID)) 62
Catch ex As Exception 62
Throw ex 62
End Try 62
End If 62
Return name 62
End Get 62
End Property 62
Public ReadOnly Property ProductName() As String 62
Get 62
Dim name As String = "" 62
If m_ProductCode <> "" Then 62
Try 62
name = ProductDB.GetProductName(m_ProductCode) 62
Catch ex As Exception 62
Throw ex 62
End Try 62
End If 62
Return name 62
End Get 62
End Property 62
End Class 62
Code of Customer Class (for partial 3D) 62
Public Class Customer 62
Private m_CustomerID As Integer 62
Private m_Name As String 62
Public Property CustomerID() As Integer 62
Get 62
Return m_CustomerID 63

End Get 63
Set(ByVal value As Integer)..63
m_CustomerID = value..63
End Set...63
End Property 63
Public Property Name() As String 63
Get..63
Return m_Name 63
End Get 63
Set(ByVal value As String) 63
m_Name = value...63
End Set...63
End Property 63
End Class..63
Code of Product Class (for partial 3D) 63
Public Class Product 63
Private m_ProductCode As String 63
Private m_Name As String 63
Public Property ProductCode() As String 63
Get..63
Return m_ProductCode 63
End Get 63
Set(ByVal value As String) 63
m_ProductCode = value...63
End Set...63
End Property 63
Public Property Name() As String 63
Get..63
Return m_Name 63
End Get 63
Set(ByVal value As String) 63
m_Name = value...63
End Set...63

End Property 63
End Class 63
Code of TechnicianDB Class (for 3D) 64
Imports System.Data.OleDb 64
Public Class TechnicianDB 64
Public Shared Function GetTechnicianName(ByVal p_TechID As Integer) As String.............64
Dim name As String 64
Dim conn As OleDbConnection = TechSupportDB.GetConnection......................................64
Dim selectQuery As String = "SELECT Name FROM Technicians WHERE TechID = " & p_TechID 64
Dim selectCmd As New OleDbCommand(selectQuery, conn)...64
Try 64
conn.Open() 64
name = selectCmd.ExecuteScalar.ToString 64
Catch ex As Exception 64
Throw ex 64
Finally..64
conn.Close() 64
End Try...64
Return name 64
End Function 64
Public Shared Function GetTechnicianList() As List(Of Technician) 64
Dim technicianList As New List(Of Technician)...64
Dim conn As OleDbConnection = TechSupportDB.GetConnection......................................64
Dim selectQuery As String = "SELECT TechID, Name From Technicians Order By Name".....64
Dim selectCmd As New OleDbCommand(selectQuery, conn)...64
Try 64
conn.Open() 64
Dim reader As OleDbDataReader = selectCmd.ExecuteReader..64
Dim technician As Technician..64
Do While reader.Read 64
technician = New Technician 64
technician.TechID = CInt(reader("TechID")) 64
technician.Name = reader("Name").ToString 64

technicianList.Add(technician) 65

Loop 65

reader.Close() 65

Catch ex As Exception 65

Throw ex 65

Finally 65

conn.Close() 65

End Try 65

Return technicianList 65

End Function 65

Public Shared Function GetTechnician(ByVal techID As Integer) As Technician 65

Dim technician As Technician = Nothing 65

Dim conn As OleDbConnection = TechSupportDB.GetConnection 65

Dim selectQuery As String = "SELECT TechID, Name, Email, Phone FROM Technicians WHERE TechID = @TechID" 65

Dim selectCmd As New OleDbCommand(selectQuery, conn) 65

selectCmd.Parameters.AddWithValue("@TechID", techID) 65

Try 65

conn.Open() 65

Dim reader As OleDbDataReader = selectCmd.ExecuteReader 65

If reader.Read Then 65

Technician = New Technician 65

Technician.TechID = CInt(reader("TechID")) 65

technician.Name = reader("Name").ToString 65

technician.Email = reader("Email").ToString 65

technician.Phone = reader("Phone").ToString 65

End If 65

reader.Close() 65

Catch ex As Exception 65

Throw ex 65

Finally 65

conn.Close() 65

End Try 65

Return technician 65

End Function 66
End Class 66
Code of IncidentDB Class (for 3D) 66
Imports System.Data.OleDb 66
Public Class IncidentDB 66
Public Shared Function GetOpenIncidents() As List(Of Incident) 66
Dim incidentList As New List(Of Incident) 66
Dim connection As OleDbConnection = TechSupportDB.GetConnection 66
Dim selectQuery = "SELECT CustomerID, ProductCode, TechID, DateOpened, Title " &.......66
"FROM Incidents " & 66
"WHERE DateClosed IS NULL" 66
Dim selectCmd As New OleDbCommand(selectQuery, connection) 66
Try 66
connection.Open() 66
Dim reader As OleDbDataReader = selectCmd.ExecuteReader 66
Dim incident As Incident..66
Do While reader.Read 66
incident = New Incident 66
incident.CustomerID = CInt(reader("CustomerID"))..66
incident.ProductCode = reader("ProductCode").ToString 66
If IsDBNull(reader("TechID")) Then 66
incident.TechID = Nothing 66
Else 66
incident.TechID = CInt(reader("TechID")) 66
End If 66
incident.DateOpened = CDate(reader("DateOpened")) 66
incident.Title = reader("Title").ToString 66
incidentList.Add(incident) 66
Loop 66
reader.Close() 66
Catch ex As Exception 66
Throw ex 66
Finally 66

connection.Close() 66

End Try 67

Return incidentList 67

End Function 67

Public Shared Sub AddIncident(ByVal p_Incident As Incident) 67

Dim connection As OleDbConnection = TechSupportDB.GetConnection 67

Dim insertQuery = "INSERT INTO Incidents " & 67

"(CustomerID, ProductCode, DateOpened, Title, Description) " & 67

"VALUES (@CustomerID, @ProductCode, @DateOpened, @Title, @Description)" 67

Dim insertCmd As New OleDbCommand(insertQuery, connection) 67

Try 67

insertCmd.Parameters.AddWithValue("@CustomerID", p_Incident.CustomerID) 67

insertCmd.Parameters.AddWithValue("@ProductCode", p_Incident.ProductCode) 67

insertCmd.Parameters.AddWithValue("@DateOpened", CDate(DateTime.Today)) 67

insertCmd.Parameters.AddWithValue("@Title", p_Incident.Title) 67

insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description) 67

connection.Open() 67

insertCmd.ExecuteNonQuery() 67

Catch ex As Exception 67

Throw ex 67

Finally 67

connection.Close() 67

End Try 67

End Sub 67

Public Shared Function GetIncident(ByVal p_IncidentID As Integer) As Incident 67

Dim incident As Incident = Nothing 67

Dim connection As OleDbConnection = TechSupportDB.GetConnection 67

Dim selectQuery = "SELECT IncidentID, CustomerID, ProductCode, TechID, " & 67

"DateOpened, DateClosed, Title, Description " & 67

"From Incidents Where IncidentID = @IncidentID" 67

Dim selectCmd As New OleDbCommand(selectQuery, connection) 67

selectCmd.Parameters.AddWithValue("@IncidentID", p_IncidentID) 67

Try 68

connection.Open() 68
Dim reader As OleDbDataReader = selectCmd.ExecuteReader 68
If reader.Read Then 68
incident = New Incident 68
incident.IncidentID = CInt(reader("IncidentID")) 68
incident.CustomerID = CInt(reader("CustomerID")) 68
incident.ProductCode = reader("ProductCode").ToString 68
If IsDBNull(reader("TechID")) Then 68
incident.TechID = Nothing 68
Else 68
incident.TechID = CInt(reader("TechID")) 68
End If 68
incident.DateOpened = CDate(reader("DateOpened")) 68
If IsDBNull(reader("DateClosed")) Then 68
incident.DateClosed = Nothing 68
Else 68
incident.DateClosed = CDate(reader("DateClosed")) 68
End If 68
incident.Title = reader("Title").ToString 68
incident.Description = reader("Description").ToString 68
End If 68
reader.Close() 68
Catch ex As Exception 68
Throw ex 68
Finally 68
connection.Close() 68
End Try 68
Return incident 68
End Function 68
Public Shared Function UpdateIncident(ByVal p_Incident As Incident, ... 68
ByVal p_Description As String) As Boolean 68
Dim isUpdated As Boolean = False 68
Dim connection As OleDbConnection = TechSupportDB.GetConnection 68

Dim insertQuery = "UPDATE Incidents SET Description = @NewDescription " & 69

"WHERE IncidentID = @IncidentID " & 69

"And Description = @Description " & 69

"And DateClosed Is NULL" 69

Dim insertCmd As New OleDbCommand(insertQuery, connection) 69

Try 69

insertCmd.Parameters.AddWithValue("@NewDescription", p_Description) 69

insertCmd.Parameters.AddWithValue("@IncidentID", p_Incident.IncidentID) 69

insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description) 69

connection.Open() 69

If insertCmd.ExecuteNonQuery() > 0 Then 69

isUpdated = True 69

End If 69

Catch ex As Exception 69

Throw ex 69

Finally 69

connection.Close() 69

End Try 69

Return isUpdated 69

End Function 69

Public Shared Function CloseIncident(ByVal p_Incident As Incident) As Boolean 69

Dim isUpdated As Boolean = False 69

Dim connection As OleDbConnection = TechSupportDB.GetConnection 69

Dim insertQuery = "UPDATE Incidents SET DateClosed = @DateClosed " & 69

"WHERE IncidentID = @IncidentID " & 69

"And Description = @Description " & 69

"And DateClosed Is NULL" 69

Dim insertCmd As New OleDbCommand(insertQuery, connection) 69

Try 70

insertCmd.Parameters.AddWithValue("@DateClosed", CDate(DateTime.Today)) 70

insertCmd.Parameters.AddWithValue("@IncidentID", p_Incident.IncidentID) 70

insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description) 70

connection.Open() 70

If insertCmd.ExecuteNonQuery() > 0 Then 70
isUpdated = True 70
End If 70
Catch ex As Exception 70
Throw ex 70
Finally 70
connection.Close() 70
End Try 70
Return isUpdated 70
End Function 70
Public Shared Function GetOpenTechnicianIncidents(ByVal techID As Integer) As List(Of Incident) 70
Dim incidentList As New List(Of Incident) 70
Dim connection As OleDbConnection = TechSupportDB.GetConnection 70
Dim selectQuery = "SELECT CustomerID, ProductCode, TechID, DateOpened, Title " &.......70
"FROM Incidents " & 70
"WHERE TechID = @TechID AND DateClosed IS NULL" 70
Dim selectCmd As New OleDbCommand(selectQuery, connection) 70
selectCmd.Parameters.AddWithValue("@TechID", techID) 70
Try 70
connection.Open() 70
Dim reader As OleDbDataReader = selectCmd.ExecuteReader 70
Dim incident As Incident 70
Do While reader.Read 70
incident = New Incident 70
incident.CustomerID = CInt(reader("CustomerID")) 70
incident.ProductCode = reader("ProductCode").ToString 70
incident.TechID = CInt(reader("TechID")) 70
incident.DateOpened = CDate(reader("DateOpened")) 70
incident.Title = reader("Title").ToString 71
incidentList.Add(incident) 71
Loop 71
reader.Close() 71
Catch ex As Exception 71

Throw ex 71
Finally..71
connection.Close() 71
End Try...71
Return incidentList 71
End Function 71
End Class 71
Code of CustomerDB Class (for partial 3D)...71
Imports System.Data.OleDb 71
Public Class CustomerDB 71
Public Shared Function GetCustomerName(ByVal p_CustomerID As Integer) As String 71
Dim name As String 71
Dim conn As OleDbConnection = TechSupportDB.GetConnection 71
Dim selectQuery As String = "SELECT Name FROM Customers WHERE CustomerID = " & p_CustomerID 71
Dim selectCmd As New OleDbCommand(selectQuery, conn)...............................71
Try 71
conn.Open()..71
name = selectCmd.ExecuteScalar.ToString 71
Catch ex As Exception 71
Throw ex 71
Finally..71
conn.Close()..71
End Try...71
Return name 71
End Function 71
Public Shared Function GetCustomerList() As List(Of Customer)........................71
Dim customerList As New List(Of Customer) 71
Dim conn As OleDbConnection = TechSupportDB.GetConnection 71
Dim selectQuery As String = "SELECT CustomerID, Name FROM Customers ORDER BY Name" 72
Dim selectCmd As New OleDbCommand(selectQuery, conn)...............................72
Try 72
conn.Open()..72

Dim reader As OleDbDataReader = selectCmd.ExecuteReader 72
Dim customer As Customer 72
Do While reader.Read 72
customer = New Customer 72
customer.CustomerID = CInt(reader("CustomerID")) 72
customer.Name = reader("Name").ToString 72
customerList.Add(customer) 72
Loop 72
reader.Close() 72
Catch ex As Exception 72
Throw ex 72
Finally 72
conn.Close() 72
End Try 72
Return customerList 72
End Function 72
End Class 72
Code of ProductDB Class (for partial 3D) 72
Imports System.Data.OleDb 72
Public Class ProductDB 72
Public Shared Function GetProductName(ByVal p_ProductCode As String) As String 72
Dim name As String 72
Dim conn As OleDbConnection = TechSupportDB.GetConnection 72
Dim selectQuery As String = "SELECT Name FROM Products WHERE ProductCode = @ProductCode" 72
Dim selectCmd As New OleDbCommand(selectQuery, conn) 72
selectCmd.Parameters.AddWithValue("@ProductCode", p_ProductCode) 73
Try 73
conn.Open() 73
name = selectCmd.ExecuteScalar.ToString 73
Catch ex As Exception 73
Throw ex 73
Finally 73
conn.Close() 73

End Try......73
Return name......73
End Function......73
Public Shared Function GetProductList() As List(Of Product)......73
Dim productList As New List(Of Product)......73
Dim conn As OleDbConnection = TechSupportDB.GetConnection......73
Dim selectQuery As String = "SELECT ProductCode, Name FROM Products ORDER BY Name" 73
Dim selectCmd As New OleDbCommand(selectQuery, conn)......73
Try......73
conn.Open()......73
Dim reader As OleDbDataReader = selectCmd.ExecuteReader......73
Dim product As Product......73
Do While reader.Read......73
product = New Product......73
product.ProductCode = reader("ProductCode").ToString......73
product.Name = reader("Name").ToString......73
productList.Add(product)......73
Loop......73
reader.Close()......73
Catch ex As Exception......73
Throw ex......73
Finally......73
conn.Close()......73
End Try......73
Return productList......74
End Function......74
End Class......74
Code of RegistrationDB Class......74
Imports System.Data.OleDb......74
Public Class RegistrationDB......74
Public Shared Function ProductRegistered(ByVal p_CustomerID As Integer, ...74
ByVal p_ProductCode As String) As Boolean......74
Dim isRegistered As Boolean = False......74

Dim connection As OleDbConnection = TechSupportDB.GetConnection 74
Dim selectQuery As String = "SELECT Count(*) FROM Registrations " &............................74
"WHERE CustomerID = @CustomerID AND ProductCode = @ProductCode" 74
Dim selectCmd As New OleDbCommand(selectQuery, connection)....................................74
Try 74
selectCmd.Parameters.AddWithValue("@CustomerID", p_CustomerID) 74
selectCmd.Parameters.AddWithValue("@ProductCode", p_ProductCode) 74
connection.Open() 74
Dim count As Integer = selectCmd.ExecuteScalar...74
If count > 0 Then 74
isRegistered = True 74
End If 74
Catch ex As Exception...74
Throw ex 74
Finally...74
connection.Close() 74
End Try...74
Return isRegistered...74
End Function 75
End Class...75
Classes of SportsPro project 75
Code of frmTechnicianIncidents_YK Class 75
Imports TechSupportData 75
Public Class frmTechnicianIncidents_YK..75
Private techList As List(Of Technician) 75
Private Sub frmTechnicianIncidents_YK_Load(sender As Object, e As EventArgs) Handles MyBase.Load...75
Try 75
techList = TechnicianDB.GetTechnicianList..75
technicianComboBox.DataSource = techList..75
Catch ex As Exception...75
MessageBox.Show(ex.Message, ex.GetType.ToString) 75
End Try...75
End Sub 75

Private Sub technicianComboBox_SelectedIndexChanged(sender As Object, e As EventArgs) Handles technicianComboBox.SelectedIndexChanged 75

Try 75

If technicianComboBox.SelectedIndex >= 0 Then 75

Dim techId = technicianComboBox.SelectedValue 75

Dim technician As Technician = TechnicianDB.GetTechnician(techId) 75

TechnicianBindingSource.Clear() 75

TechnicianBindingSource.Add(technician) 75

Dim incidentList As List(Of Incident) = IncidentDB.GetOpenTechnicianIncidents(techId)......75

IncidentBindingSource.DataSource = incidentList 75

End If 75

Catch ex As Exception 75

MessageBox.Show(ex.Message, ex.GetType.ToString) 75

End Try 75

End Sub 75

End Class 75

Added code of frmMain_YK Class only for Project 3-D 75

Private Sub DisplayOpenIncidentsByTechnician3DToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles DisplayOpenIncidentsByTechnician3DToolStripMenuItem.Click 76

Dim frmTechIncidents As New frmTechnicianIncidents_YK 76

frmTechIncidents.Show() 76

End Sub 76

Specifications

1. The combo box and the two text boxes on this form should be bound to an object data source that's created from the Technician class. The DataGridView control should be bound to an object data source that's created from the Incident class.
2. When this form is first displayed, the first technician should be selected and the data for that technician should be displayed on the form.

The design of the Incident class

The private fields that store the property values

```
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
```

The IncidentID property

```
Public Property IncidentID() As Integer
```

Gets and sets the incident ID for the incident.

The CustomerID property

```
Public Property CustomerID() As Integer
```

Gets and sets the customer ID for the incident.

The ProductCode property

```
Public Property ProductCode() As String
```

Gets and sets the product code for the incident.

The TechID property

```
Public Property TechID() As Nullable(Of Integer)
```

Gets and sets the technician ID for the technician assigned to the incident. Null if a technician has not been assigned.

The DateOpened property

```
Public Property DateOpened() As Date
```

Gets and sets the date the incident was created.

The DateClosed property

```
Public Property DateClosed() As Nullable(Of Date)
```

Gets and sets the date the incident was closed. Null if the incident is still open.

The Title property

```
Public Property Title() As String
```

Gets and sets the title for the incident.

The Description property

```
Public Property Description() As String
```

Gets and sets the description for the incident.

The CustomerName property

```
Public ReadOnly Property CustomerName() As String
```

Gets the name for the customer associated with the incident using the GetCustomerName method of the CustomerDB class.

The ProductName property

```
Public ReadOnly Property ProductName() As String
```

Gets the name for the product associated with the incident using the GetProductName method of the ProductDB class.

The design of the Technician class

The private fields that store the property values

```
    Private m_TechID As Integer
    Private m_Name As String
    Private m_Email As String
    Private m_Phone As String
```

The TechID property

```
Public Property TechID() As Integer
```

Gets and sets the ID for the technician.

The Name property

```
Public Property Name() As String
```

Gets and sets the name for the technician.

The Email property

```
Public Property Email() As String
```

Gets and sets the email address for the technician.

The Phone property

```
Public Property Phone() As String
```

Gets and sets the phone number for the technician.

The design of the TechSupportDB class

The GetConnection method

```
Public Shared Function GetConnection() As SqlConnection
```

Returns a SqlConnection object that establishes a connection to the TechSupport database.

The design of the TechnicianDB class

The GetTechnicianList method

```
Public Shared Function GetTechnicianList() As List(Of Technician)
```

Returns a List(Of Technician) object that contains one item for each technician in the Technicians table.

The GetTechnician method

```
Public Shared Function GetTechnician(ByVal techID As Integer) _
    As Technician
```

Returns a Technician object for the technician with the specified ID.

The design of the IncidentDB class

The GetOpenTechnicianIncidents method

```
Public Shared Function GetOpenTechnicianIncidents(
    ByVal techID As Integer) As List(Of Incident)
```

Returns a List(Of Incident) object that contains one item for each open incident for the technician with the specified ID.

The design of the ProductDB class

The GetProductName method

```
Public Shared Function GetProductName(ByVal productCode As String) _
    As String
```

Returns a string that contains the name of the product with the specified product code.

The design of the CustomerDB class

The GetCustomerName method

```
Public Shared Function GetCustomerName(ByVal customerID As Integer) _
    As String
```

Returns a string that contains the name of the customer with the specified ID.

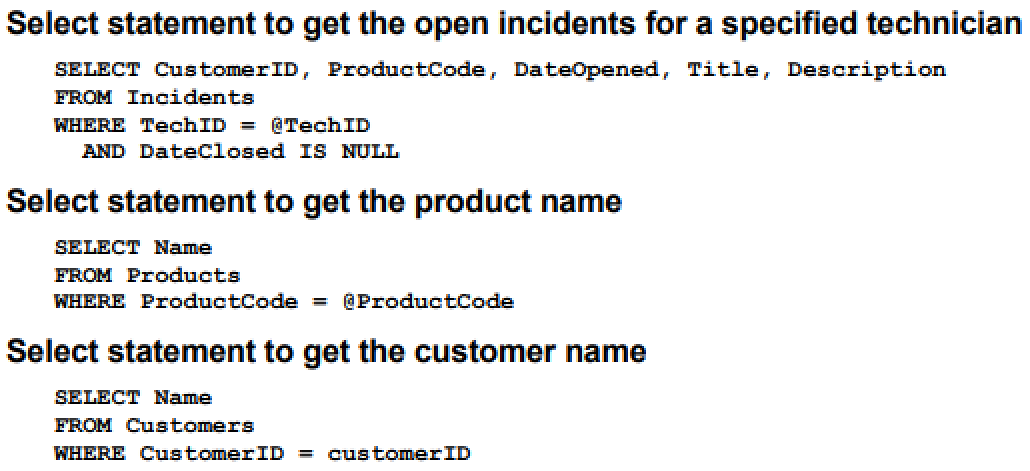
SQL statements

Select statement to get the list of technicians

```
SELECT TechID, Name
FROM Technicians
ORDER BY Name
```

Select statement to get the data for a specified technician

```
SELECT TechID, Name, Email, Phone
FROM Technicians
WHERE TechID = @TechID
```



Select statement to get the open incidents for a specified technician

```
SELECT CustomerID, ProductCode, DateOpened, Title, Description
FROM Incidents
WHERE TechID = @TechID
  AND DateClosed IS NULL
```

Select statement to get the product name

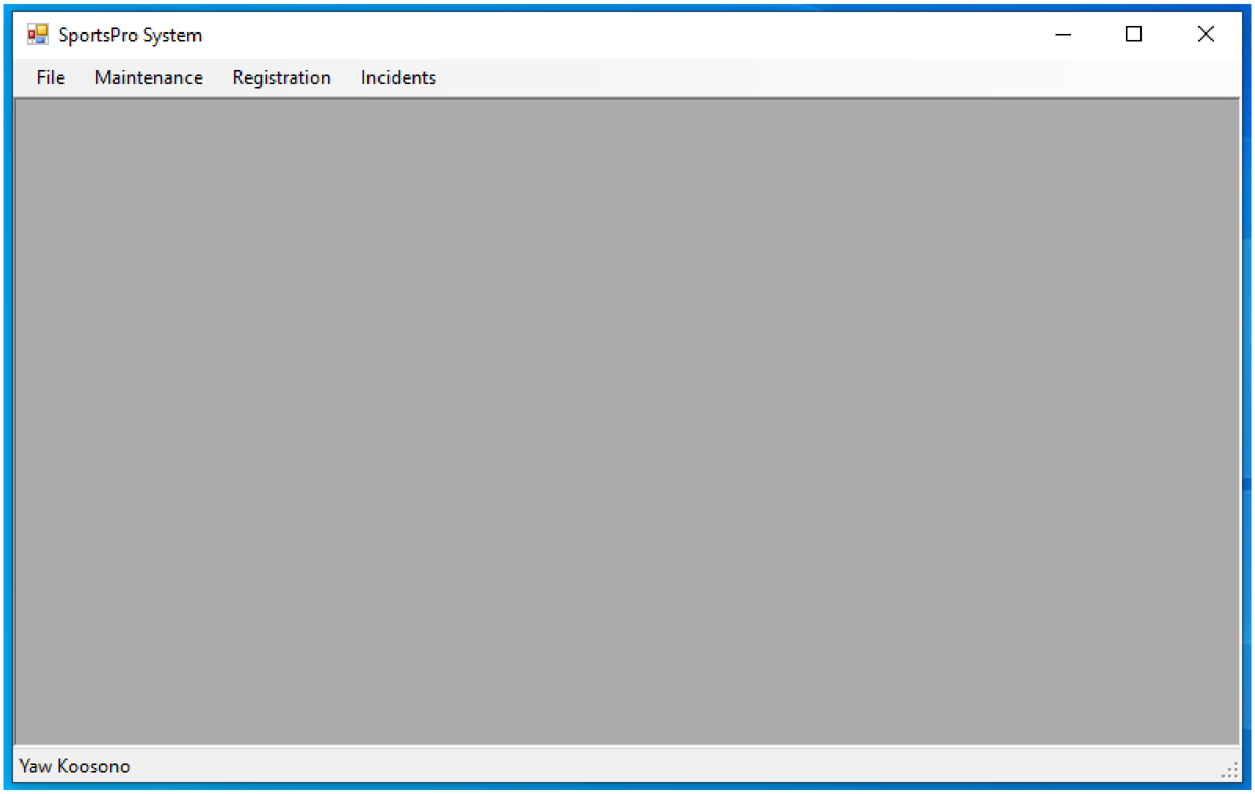
```
SELECT Name
FROM Products
WHERE ProductCode = @ProductCode
```

Select statement to get the customer name

```
SELECT Name
FROM Customers
WHERE CustomerID = customerID
```

A walk through of the application

Main form display when user run the application. All sub forms can navigate through main form.



Yaw Koosono

Fig-1: Main form when program start

Displaying the navigation menu for Display open incidents by technician form.

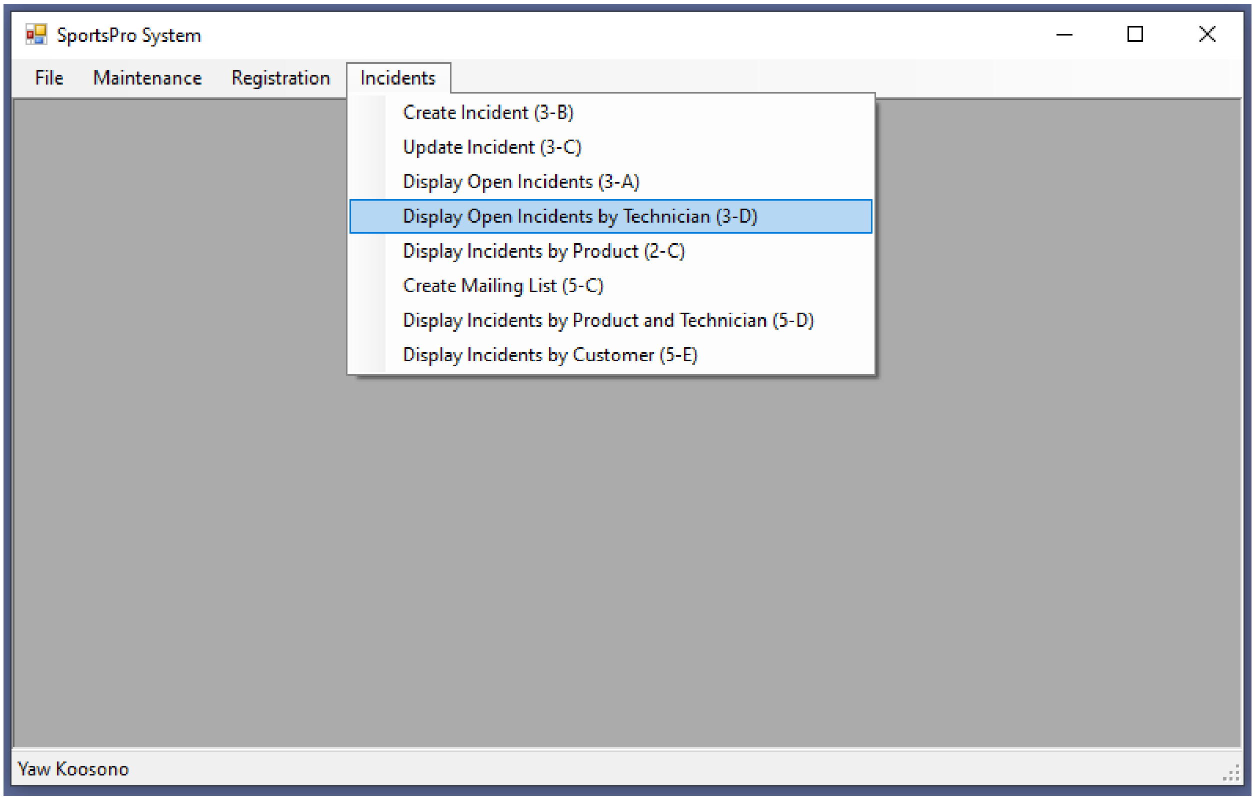
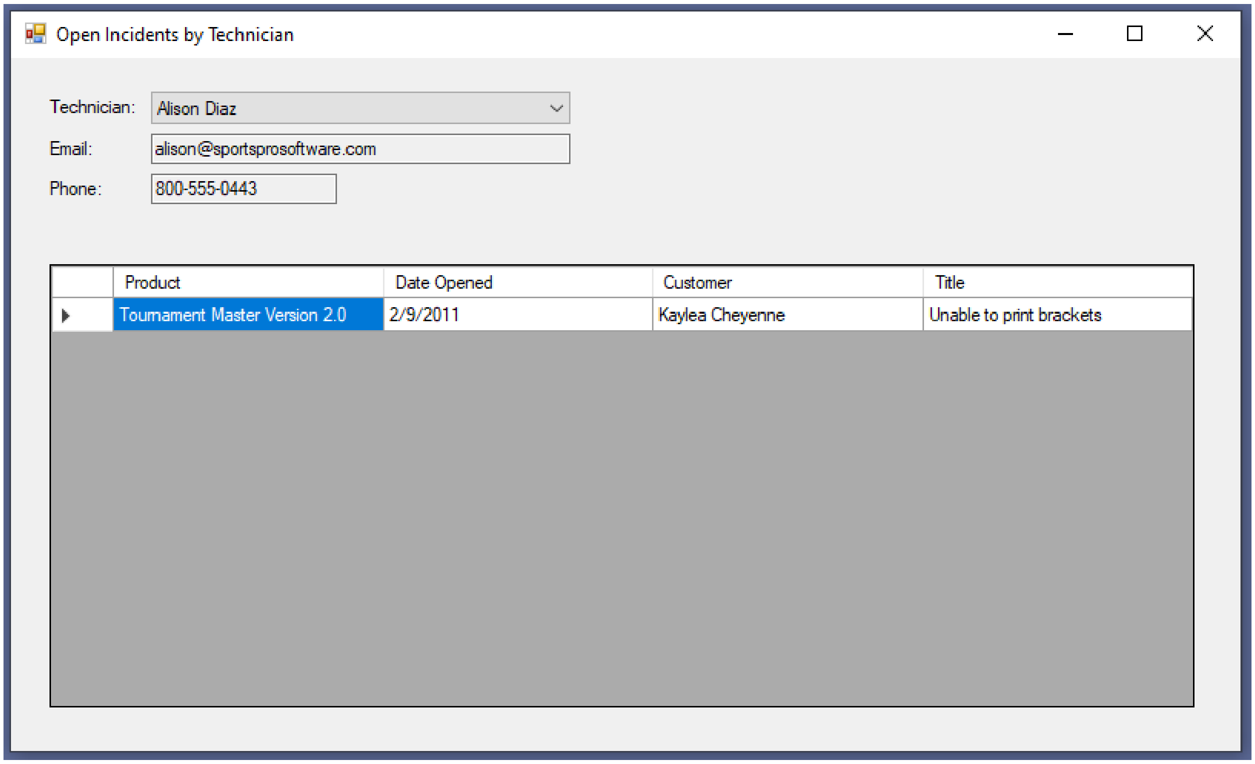

SportsPro System
File
Maintenance
Registration
Incidents
Create Incident (3-B)
Update Incident (3-C)
Display Open Incidents (3-A)
Display Open Incidents by Technician (3-D)
Display Incidents by Product (2-C)
Create Mailing List (5-C)
Display Incidents by Product and Technician (5-D)
Display Incidents by Customer (5-E)
Yaw Koosono

Fig-2: Main form with Display open Incidents by technician navigation

After open the form showing first selected technician with linked open incidents.



Technician: Alison Diaz

Email: alison@sportsprosoftware.com

Phone: 800-555-0443

Product	Date Opened	Customer	Title
Tournament Master Version 2.0	2/9/2011	Kaylea Cheyenne	Unable to print brackets

Fig-3: Display open incidents by technician form

Source Code

Classes of TechSupportData class library

Code of TechSupportDB Class (for partial 3D)

```
Imports System.Data.OLEDB
Public Class TechSupportDB
    Public Shared Function GetConnection() As OleDbConnection
        Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Bob\TechSupport.mdb;Persist Security Info=True"
        Return New OleDbConnection(connString)
    End Function
End Class
```

Code of Technician Class (for 3D)

```
Public Class Technician
    Private m_TechID As Integer
    Private m_Name As String
    Private m_Email As String
    Private m_Phone As String
    Public Property TechID() As Integer
        Get
            Return m_TechID
        End Get
        Set(ByVal value As Integer)
            m_TechID = value
        End Set
    End Property

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

    Public Property Email() As String
        Get
            Return m_Email
        End Get
        Set(ByVal value As String)
            m_Email = value
        End Set
    End Property

    Public Property Phone() As String
        Get
```

```
            Return m_Phone
        End Get
        Set(ByVal value As String)
            m_Phone = value
        End Set
    End Property
End Class
```

Code of Incident Class (for 3D)

```
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

    Public Sub New()

    End Sub

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

    Public Property CustomerID() As Integer
        Get
            Return m_CustomerID
        End Get
        Set(ByVal 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 CDate(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 TechName() As String
        Get
            Dim name As String = ""

            If m_TechID.HasValue And m_TechID <> 0 Then
                Try
                    name = TechnicianDB.GetTechnicianName(CInt(m_TechID))
                Catch ex As Exception
                    Throw ex
                End Try
            End If
            Return name
        End Get
    End Property

    Public ReadOnly Property ProductName() As String
        Get
            Dim name As String = ""

            If m_ProductCode <> "" Then
                Try
                    name = ProductDB.GetProductName(m_ProductCode)
                Catch ex As Exception
                    Throw ex
                End Try
            End If
            Return name
        End Get
    End Property
End Class
```

Code of Customer Class (for partial 3D)

```
Public Class Customer
    Private m_CustomerID As Integer
    Private m_Name As String
    Public Property CustomerID() As Integer
        Get
```

```
            Return m_CustomerID
        End Get
        Set(ByVal value As Integer)
            m_CustomerID = value
        End Set
    End Property

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

Code of Product Class (for partial 3D)

```
Public Class Product
    Private m_ProductCode As String
    Private m_Name As String
    Public Property ProductCode() As String
        Get
            Return m_ProductCode
        End Get
        Set(ByVal value As String)
            m_ProductCode = value
        End Set
    End Property

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

Code of TechnicianDB Class (for 3D)

```
Imports System.Data.OleDb
Public Class TechnicianDB
    Public Shared Function GetTechnicianName(ByVal p_TechID As Integer) As String
        Dim name As String
        Dim conn As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery As String = "SELECT Name FROM Technicians WHERE TechID = " &
p_TechID
        Dim selectCmd As New OleDbCommand(selectQuery, conn)

        Try
            conn.Open()
            name = selectCmd.ExecuteScalar.ToString
        Catch ex As Exception
            Throw ex
        Finally
            conn.Close()
        End Try
        Return name
    End Function

    Public Shared Function GetTechnicianList() As List(Of Technician)
        Dim technicianList As New List(Of Technician)
        Dim conn As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery As String = "SELECT TechID, Name From Technicians Order By Name"
        Dim selectCmd As New OleDbCommand(selectQuery, conn)

        Try
            conn.Open()
            Dim reader As OleDbDataReader = selectCmd.ExecuteReader
            Dim technician As Technician

            Do While reader.Read
                technician = New Technician
                technician.TechID = CInt(reader("TechID"))
                technician.Name = reader("Name").ToString
                technicianList.Add(technician)
            Loop
            reader.Close()
        Catch ex As Exception
            Throw ex
        Finally
            conn.Close()
        End Try
        Return technicianList
    End Function

    Public Shared Function GetTechnician(ByVal techID As Integer) As Technician
        Dim technician As Technician = Nothing
        Dim conn As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery As String = "SELECT TechID, Name, Email, Phone FROM Technicians
WHERE TechID = @TechID"
        Dim selectCmd As New OleDbCommand(selectQuery, conn)
        selectCmd.Parameters.AddWithValue("@TechID", techID)

        Try
            conn.Open()
            Dim reader As OleDbDataReader = selectCmd.ExecuteReader

            If reader.Read Then
                Technician = New Technician
                Technician.TechID = CInt(reader("TechID"))
                technician.Name = reader("Name").ToString
                technician.Email = reader("Email").ToString
                technician.Phone = reader("Phone").ToString
            End If
            reader.Close()
        Catch ex As Exception
            Throw ex
        Finally
            conn.Close()
        End Try

        Return technician
    End Function
End Class
```

Code of IncidentDB Class (for 3D)

```
Imports System.Data.OleDb
Public Class IncidentDB
    Public Shared Function GetOpenIncidents() As List(Of Incident)
        Dim incidentList As New List(Of Incident)
        Dim connection As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery = "SELECT CustomerID, ProductCode, TechID, DateOpened, Title " &
                            "FROM Incidents " &
                            "WHERE DateClosed IS NULL"
        Dim selectCmd As New OleDbCommand(selectQuery, connection)

        Try
            connection.Open()
            Dim reader As OleDbDataReader = selectCmd.ExecuteReader
            Dim incident As Incident

            Do While reader.Read
                incident = New Incident
                incident.CustomerID = CInt(reader("CustomerID"))
                incident.ProductCode = reader("ProductCode").ToString

                If IsDBNull(reader("TechID")) Then
                    incident.TechID = Nothing
                Else
                    incident.TechID = CInt(reader("TechID"))
                End If
                incident.DateOpened = CDate(reader("DateOpened"))
                incident.Title = reader("Title").ToString
                incidentList.Add(incident)
            Loop
            reader.Close()
        Catch ex As Exception
            Throw ex
        Finally
            connection.Close()
        End Try
        Return incidentList
    End Function

    Public Shared Sub AddIncident(ByVal p_Incident As Incident)
        Dim connection As OleDbConnection = TechSupportDB.GetConnection

        Dim insertQuery = "INSERT INTO Incidents " &
                            "(CustomerID, ProductCode, DateOpened, Title, Description) " &
                            "VALUES (@CustomerID, @ProductCode, @DateOpened, @Title,
@Description)"

        Dim insertCmd As New OleDbCommand(insertQuery, connection)

        Try
            insertCmd.Parameters.AddWithValue("@CustomerID", p_Incident.CustomerID)
            insertCmd.Parameters.AddWithValue("@ProductCode", p_Incident.ProductCode)
            insertCmd.Parameters.AddWithValue("@DateOpened", CDate(DateTime.Today))
            insertCmd.Parameters.AddWithValue("@Title", p_Incident.Title)
            insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description)
            connection.Open()
            insertCmd.ExecuteNonQuery()
        Catch ex As Exception
            Throw ex
        Finally
            connection.Close()
        End Try
    End Sub

    Public Shared Function GetIncident(ByVal p_IncidentID As Integer) As Incident
        Dim incident As Incident = Nothing
        Dim connection As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery = "SELECT IncidentID, CustomerID, ProductCode, TechID, " &
                          "DateOpened, DateClosed, Title, Description " &
                          "From Incidents Where IncidentID = @IncidentID"
        Dim selectCmd As New OleDbCommand(selectQuery, connection)
        selectCmd.Parameters.AddWithValue("@IncidentID", p_IncidentID)
    Try
        connection.Open()
        Dim reader As OleDbDataReader = selectCmd.ExecuteReader

        If reader.Read Then
            incident = New Incident
            incident.IncidentID = CInt(reader("IncidentID"))
            incident.CustomerID = CInt(reader("CustomerID"))
            incident.ProductCode = reader("ProductCode").ToString

            If IsDBNull(reader("TechID")) Then
                incident.TechID = Nothing
            Else
                incident.TechID = CInt(reader("TechID"))
            End If
            incident.DateOpened = CDate(reader("DateOpened"))
            If IsDBNull(reader("DateClosed")) Then
                incident.DateClosed = Nothing
            Else
                incident.DateClosed = CDate(reader("DateClosed"))
            End If
            incident.Title = reader("Title").ToString
            incident.Description = reader("Description").ToString
        End If

        reader.Close()
    Catch ex As Exception
        Throw ex
    Finally
        connection.Close()
    End Try
    Return incident
End Function

Public Shared Function UpdateIncident(ByVal p_Incident As Incident,
                             ByVal p_Description As String) As Boolean

    Dim isUpdated As Boolean = False
    Dim connection As OleDbConnection = TechSupportDB.GetConnection
    Dim insertQuery = "UPDATE Incidents SET Description = @NewDescription " &
                      "WHERE IncidentID = @IncidentID " &
                      "And Description = @Description " &
                      "And DateClosed Is NULL"


    Dim insertCmd As New OleDbCommand(insertQuery, connection)

    Try
        insertCmd.Parameters.AddWithValue("@NewDescription", p_Description)
        insertCmd.Parameters.AddWithValue("@IncidentID", p_Incident.IncidentID)
        insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description)

        connection.Open()
        If insertCmd.ExecuteNonQuery() > 0 Then
            isUpdated = True
        End If
    Catch ex As Exception
        Throw ex
    Finally
        connection.Close()
    End Try
    Return isUpdated
End Function

Public Shared Function CloseIncident(ByVal p_Incident As Incident) As Boolean

    Dim isUpdated As Boolean = False
    Dim connection As OleDbConnection = TechSupportDB.GetConnection

    Dim insertQuery = "UPDATE Incidents SET DateClosed = @DateClosed " &
                      "WHERE IncidentID = @IncidentID " &
                      "And Description = @Description " &
                      "And DateClosed Is NULL"


    Dim insertCmd As New OleDbCommand(insertQuery, connection)
```

```
        Try
            insertCmd.Parameters.AddWithValue("@DateClosed", CDate(DateTime.Today))
            insertCmd.Parameters.AddWithValue("@IncidentID", p_Incident.IncidentID)
            insertCmd.Parameters.AddWithValue("@Description", p_Incident.Description)

            connection.Open()
            If insertCmd.ExecuteNonQuery() > 0 Then
                isUpdated = True
            End If
        Catch ex As Exception
            Throw ex
        Finally
            connection.Close()
        End Try
        Return isUpdated
    End Function

    Public Shared Function GetOpenTechnicianIncidents(ByVal techID As Integer) As List(Of
Incident)
        Dim incidentList As New List(Of Incident)
        Dim connection As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery = "SELECT CustomerID, ProductCode, TechID, DateOpened, Title " &
                            "FROM Incidents " &
                            "WHERE TechID = @TechID AND DateClosed IS NULL"
        Dim selectCmd As New OleDbCommand(selectQuery, connection)
        selectCmd.Parameters.AddWithValue("@TechID", techID)

        Try
            connection.Open()
            Dim reader As OleDbDataReader = selectCmd.ExecuteReader
            Dim incident As Incident

            Do While reader.Read
                incident = New Incident
                incident.CustomerID = CInt(reader("CustomerID"))
                incident.ProductCode = reader("ProductCode").ToString
                incident.TechID = CInt(reader("TechID"))
                incident.DateOpened = CDate(reader("DateOpened"))
                incident.Title = reader("Title").ToString
                incidentList.Add(incident)
            Loop
            reader.Close()
        Catch ex As Exception
            Throw ex
        Finally
            connection.Close()
        End Try
        Return incidentList
    End Function
End Class
```

Code of CustomerDB Class (for partial 3D)

```
Imports System.Data.OleDb
Public Class CustomerDB
    Public Shared Function GetCustomerName(ByVal p_CustomerID As Integer) As String
        Dim name As String
        Dim conn As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery As String = "SELECT Name FROM Customers WHERE CustomerID = " &
p_CustomerID
        Dim selectCmd As New OleDbCommand(selectQuery, conn)

        Try
            conn.Open()
            name = selectCmd.ExecuteScalar.ToString
        Catch ex As Exception
            Throw ex
        Finally
            conn.Close()
        End Try
        Return name
    End Function

    Public Shared Function GetCustomerList() As List(Of Customer)
        Dim customerList As New List(Of Customer)
        Dim conn As OleDbConnection = TechSupportDB.GetConnection
        Dim selectQuery As String = "SELECT CustomerID, Name FROM Customers ORDER BY Name"
        Dim selectCmd As New OleDbCommand(selectQuery, conn)

        Try
            conn.Open()
            Dim reader As OleDbDataReader = selectCmd.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 Exception
            Throw ex
        Finally
            conn.Close()
        End Try
        Return customerList
    End Function
End Class
```

Code of ProductDB Class (for partial 3D)

```
Imports System.Data.OleDb

Public Class ProductDB
    Public Shared Function GetProductName(ByVal p_ProductCode As String) As String
        Dim name As String
        Dim conn As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery As String = "SELECT Name FROM Products WHERE ProductCode = @ProductCode"
        Dim selectCmd As New OleDbCommand(selectQuery, conn)
        selectCmd.Parameters.AddWithValue("@ProductCode", p_ProductCode)

        Try
            conn.Open()
            name = selectCmd.ExecuteScalar.ToString
        Catch ex As Exception
            Throw ex
        Finally
            conn.Close()
        End Try
        Return name
    End Function

    Public Shared Function GetProductList() As List(Of Product)
        Dim productList As New List(Of Product)
        Dim conn As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery As String = "SELECT ProductCode, Name FROM Products ORDER BY
Name"
        Dim selectCmd As New OleDbCommand(selectQuery, conn)

        Try
            conn.Open()
            Dim reader As OleDbDataReader = selectCmd.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 Exception
            Throw ex
        Finally
            conn.Close()
        End Try
        Return productList
    End Function
End Class
```

Code of RegistrationDB Class

```
Imports System.Data.OleDb

Public Class RegistrationDB

    Public Shared Function ProductRegistered(ByVal p_CustomerID As Integer,
                      ByVal p_ProductCode As String) As Boolean

        Dim isRegistered As Boolean = False
        Dim connection As OleDbConnection = TechSupportDB.GetConnection

        Dim selectQuery As String = "SELECT Count(*) FROM Registrations " &
            "WHERE CustomerID = @CustomerID AND ProductCode = @ProductCode"

        Dim selectCmd As New OleDbCommand(selectQuery, connection)

        Try
            selectCmd.Parameters.AddWithValue("@CustomerID", p_CustomerID)
            selectCmd.Parameters.AddWithValue("@ProductCode", p_ProductCode)
            connection.Open()

            Dim count As Integer = selectCmd.ExecuteScalar

            If count > 0 Then
                isRegistered = True
            End If

        Catch ex As Exception
            Throw ex
        Finally
            connection.Close()
        End Try

        Return isRegistered
    End Function
End Class
```

Classes of SportsPro project

Code of frmTechnicianIncidents_YK Class

```
Imports TechSupportData
Public Class frmTechnicianIncidents_YK
    Private techList As List(Of Technician)
    Private Sub frmTechnicianIncidents_YK_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Try
            techList = TechnicianDB.GetTechnicianList
            technicianComboBox.DataSource = techList
        Catch ex As Exception
            MessageBox.Show(ex.Message, ex.GetType.ToString)
        End Try
    End Sub

    Private Sub technicianComboBox_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles technicianComboBox.SelectedIndexChanged
        Try
            If technicianComboBox.SelectedIndex >= 0 Then
                Dim techId = technicianComboBox.SelectedValue
                Dim technician As Technician = TechnicianDB.GetTechnician(techId)
                TechnicianBindingSource.Clear()
                TechnicianBindingSource.Add(technician)

                Dim incidentList As List(Of Incident) =
IncidentDB.GetOpenTechnicianIncidents(techId)
                IncidentBindingSource.DataSource = incidentList

            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message, ex.GetType.ToString)
        End Try
    End Sub
End Class
```

Added code of frmMain_YK Class only for Project 3-D

```
Private Sub DisplayOpenIncidentsByTechnician3DToolStripMenuItem_Click(sender As Object, e
As EventArgs) Handles DisplayOpenIncidentsByTechnician3DToolStripMenuItem.Click
    Dim frmTechIncidents As New frmTechnicianIncidents_YK

    frmTechIncidents.Show()
End Sub
```