

Option Compare Database

Option Explicit

Private strBookMark As Variant ' BookMark for Current Record.

Private Sub binCancelled\_AfterUpdate()

' Error Trapping System Initiated.

If glbHandleErrors Then On Error GoTo Proc\_Err

' Dimension Variables.

Dim dbs As DAO.Database

Dim rsSQL As DAO.Recordset

Dim sqlStr As String

Dim strMessage As String, Response

' Validate that 1) User really wants to mark this Person as Cancelled Record, because if so GDPR Anonymisation Routine is instigated. In

' addition, 2) binGDPR is Checked as True, Then if Persons End Date is Null, automatically add members end date

If Me.Dirty Then Me.Dirty = False ' Save any potential record changes prior to carrying out this sub routine (avoids write conflicts with form).

' THIS IS THE PART 1 Option of the Validation Routine.

If binCancelled = True And Not IsNull(pkPersonID) Then

' Prompt GDPR Validation (Double Confirmation Routine).

' Initialise Message String.

' Confirmation Message 1.

strMessage = "Marking as Cancelled Record will, trigger GDPR Protocol." & vbCrLf \_

& "Personal Details relating to Member and Emergency Contact, will be Anonymised."  
& vbCrLf \_

& "Select NO to Abandon Change or YES to Continue"

```
Response = MsgBox(strMessage, vbYesNo)
```

```
If Response = vbNo Then
```

```
    ' Abandon Changes.
```

```
    Me.Undo
```

```
    Me.binCancelled.SetFocus
```

```
    Me.binCancelled = False
```

```
    Me.binCancelled.Enabled = True
```

```
    Exit Sub
```

```
Else
```

```
    ' Confirmation Message 2. Ask User to Re-Confirm GDPR Anonymisation Protocol.
```

```
    strMessage = "Please confirm you wish to Anonymise, Member and Emergency Contact  
data. " & vbCrLf _
```

```
        & "Select YES to Confirm or NO to Abandon Changes"
```

```
Response = MsgBox(strMessage, vbYesNo)
```

```
If Response = vbNo Then
```

```
    ' Abandon Changes.
```

```
    Me.Undo
```

```
    Me.binCancelled.SetFocus
```

```
    Me.binCancelled = False
```

```
    Me.binCancelled.Enabled = True
```

```
    Exit Sub
```

```
End If
```

```
End If
```

```
Else
```

```
    Exit Sub
```

```
End If
```

' GDPR Anonymisation Routine. Two Parts A) Member Data/Emergency Contact Data and B)  
Updating Finish Date for all relevant records in Status Sub-Form.

' A) Generate Member/Emergency Contact Personal Information RecordSet for individual Person.

```
sqlStr = "SELECT tblPerson.pkPersonID, tblPerson.fkTitleID, tblPerson.txtFirstName,  
tblPerson.txtMiddleName, tblPerson.txtSurname, " _
```

```
& "tblPerson.txtNameNo, tblPerson.txtAddress1, tblPerson.txtAddress2,  
tblPerson.txtParishWard, tblPerson.txtCityTown, " _
```

```
& "tblPerson.fkCountyID, tblPerson.txtHomeTelNo, tblPerson.txtMobileNo,  
tblPerson.txtEMail, tblPerson.binGDPR FROM tblPerson " _
```

```
& "WHERE tblPerson.pkPersonID = " & Val(Me![pkPersonID]) & ";"
```

```
Set dbs = CurrentDb
```

```
Set rsSQL = dbs.OpenRecordset(sqlStr)
```

```
rsSQL.MoveFirst ' Moves to First Record in RecordSet
```

```
Do While Not rsSQL.BOF And Not rsSQL.EOF
```

```
' Anonymise Member/Non Member Person Data
```

```
rsSQL.Edit
```

```
' Member Details.
```

```
rsSQL!fkTitleID = Null
```

```
rsSQL!txtFirstName = "*****"
```

```
rsSQL!txtMiddleName = IIf(IsNull(Me.txtMiddleName), Null, "*****")
```

```
rsSQL!txtSurname = IIf(IsNull(Me.txtSurname), Null, "*****")
```

```
rsSQL!txtNameNo = IIf(IsNull(Me.txtNameNo), Null, "*****")
```

```
rsSQL!txtAddress1 = IIf(IsNull(Me.txtAddress1), Null, "*****")
```

```
rsSQL!txtAddress2 = IIf(IsNull(Me.txtAddress2), Null, "*****")
```

```
rsSQL!txtParishWard = IIf(IsNull(Me.txtParishWard), Null, "*****")
```

```
rsSQL!txtCityTown = IIf(IsNull(Me.txtCityTown), Null, "*****")
```

```
rsSQL!fkCountyID = Null
```

```
rsSQL!txtHomeTelNo = If(IsNull(Me.txtHomeTelNo), Null, "*****")
```

```
rsSQL!txtMobileNo = If(IsNull(Me.txtMobileNo), Null, "*****")
```

```
rsSQL!txtEMail = If(IsNull(Me.txtEMail), Null, "*****")
```

```
' Emergency Contact Details.
```

```
' ADD LATER
```

```
' Upgrade GDPR Field.
```

```
rsSQL!binGDPR = True
```

```
binGDPR.Enabled = False ' Set GDPR Box to False
```

```
' Upgrade Membership Cancelled Enabled Field.
```

```
binCancelled.Enabled = False
```

```
rsSQL.Update
```

```
rsSQL.MoveNext ' Moves to next record.
```

```
Loop
```

```
' Recordset/Database CleanUp.
```

```
rsSQL.Close
```

```
Set rsSQL = Nothing
```

```
' THIS IS THE PART 2 Option of Validation Routine
```

```
' Generate Status Info Sub Form Recordset.
```

```
sqlStr = "SELECT tblStatus.fkPersonID, tblStatus.dtmSFinish FROM tblPerson " _
```

```
& "INNER JOIN tblStatus ON tblPerson.pkPersonID = tblStatus.fkPersonID " _
```

```
& "WHERE tblPerson.pkPersonID = " & Val(Me![pkPersonID]) & " AND (tblStatus.dtmSFinish)  
Is Null" & " ;"
```

```
Set rsSQL = dbs.OpenRecordset(sqlStr)
```

rsSQL.MoveFirst ' Moves to First Record in Record Set

If Err.Number = 3021 Then GoTo Proc\_CleanUp

Do While Not rsSQL.BOF And Not rsSQL.EOF

' Update the dtmSFinish field with Today's Date as Default (Can be overwritten with Actual Date).

rsSQL.Edit

rsSQL!dtmSFinish = Date

rsSQL.Update

rsSQL.MoveNext ' Moves to next record.

Loop

Proc\_CleanUp:

rsSQL.Close

dbs.Close

Set dbs = Nothing

Set rsSQL = Nothing

' Clear Any Errors

Err.Clear

' Exit Sub-Routine

Exit Sub

Proc\_Err:

' Generates Error Message

If Err.Number = 3021 Then Resume Next ' Checks to see if No Current Record is thrown

If Err.Number = 2105 Then Resume Next

```
MsgBox "Error: (" & Err.Number & ") " & Err.Description & ". E-Mail Error Number and Description  
to rberry425@gmail.com", vbCritical
```

```
GoTo Proc_CleanUp
```

```
End Sub
```

```
Private Sub binCancelled_Click() ' Error Trapping System Initiated.
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
' Automatically Updates the Comments Box.
```

```
If binCancelled Then
```

```
    memMiscComment = If(IsNull(memMiscComment), "Duplicate Record - Membership Cancelled or  
Member No Longer Active", memMiscComment & ". Duplicate Record - Membership Cancelled") '   
Membership Cancelled Box Checked
```

```
    binCancelled.Enabled = False
```

```
Else
```

```
    memMiscComment = "" ' Membership Cancelled Box NOT Checked
```

```
End If
```

```
ExitHere: ' Any Clean Up Code
```

```
    Err.Clear
```

```
Exit Sub
```

```
ErrorHandler: ' ERROR HANDLING ROUTINE.
```

```
If Err.Number <> 0 Then
```

```
    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
    Resume ExitHere
```

```
End If
```

```
End Sub
```

```
Private Sub binGDPR_AfterUpdate()
```

' Error Trapping System Initiated.

If glbHandleErrors Then On Error GoTo Proc\_Err

' Dimension Variables.

Dim dbs As DAO.Database

Dim rsSQL As DAO.Recordset

Dim sqlStr As String

Dim strMessage As String, Response

' Validate that 1) User really wants to mark this Person as GDPR Compliant, because if so GDPR Anonymisation Routine is instigated. In

' addition, 2) binGDPR is Checked as True, Then if Members End Date is Null, automatically add members end-date to fields that are null.

If Me.Dirty Then Me.Dirty = False ' Save any potential record changes prior to carrying out this sub routine (avoids write conflicts with form).

' THIS IS THE PART 1 Option of the Validation Routine.

If binGDPR = True And Not IsNull(pkPersonID) Then

    ' Prompt GDPR Validation (Double Confirmation Routine).

        ' Initialise Message String.

        ' Confirmation Message 1.

        strMessage = "Marking Member as GDPR Compliant will, trigger GDPR Protocol." & vbCrLf \_  
                    & "Personal Details relating to Member and Emergency Contact, will be Anonymised."  
        & vbCrLf \_  
            & "Select NO to Abandon Change or YES to Continue"

Response = MsgBox(strMessage, vbYesNo)

If Response = vbNo Then

    ' Abandon Changes.

    Me.Undo

```

        Me.binGDPR.SetFocus

        Me.binGDPR = False

        Me.binGDPR.Enabled = True

        Exit Sub

    Else

        ' Confirmation Message 2. Ask User to Re-Confirm GDPR Anonymisation Protocol.

        strMessage = "Please confirm you wish to Anonymise, Member and Emergency Contact
data. " & vbCrLf _

            & "Select YES to Confirm or NO to Abandon Changes"

        Response = MsgBox(strMessage, vbYesNo)

        If Response = vbNo Then

            ' Abandon Changes.

            Me.Undo

            Me.binGDPR.SetFocus

            Me.binGDPR = False

            Me.binGDPR.Enabled = True

            Exit Sub

        End If

    End If

Else

    Exit Sub

End If

' GDPR Anonymisation Routine. Two Parts A) Member Data/Emergency Contact Data and B)
Updating Finish Date for all relevant records in Status Sub-Form.

' A) Generate Member/Emergency Contact Personal Information RecordSet for individual Person.

sqlStr = "SELECT tblPerson.pkPersonID, tblPerson.fkTitleID, tblPerson.txtFirstName,
tblPerson.txtMiddleName, tblPerson.txtSurname, " _

    & "tblPerson.txtNameNo, tblPerson.txtAddress1, tblPerson.txtAddress2,
tblPerson.txtParishWard, tblPerson.txtCityTown, " _

```



& "tblPerson.fkCountyID, tblPerson.txtHomeTelNo, tblPerson.txtMobileNo,  
tblPerson.txtEMail, tblPerson.fkECTitleID, " \_

& "tblPerson.txtECFirstName, tblPerson.txtECMiddleName, tblPerson.txtECSurname,  
tblPerson.txtECHomeTelNo, " \_

& "tblPerson.txtECEMail, tblPerson.fkECID, tblPerson.binGDPR FROM tblPerson " \_

& "WHERE tblPerson.pkPersonID = " & Val(Me![pkPersonID]) & ";"

Set dbs = CurrentDb

Set rsSQL = dbs.OpenRecordset(sqlStr)

rsSQL.MoveFirst ' Moves to First Record in RecordSet

Do While Not rsSQL.BOF And Not rsSQL.EOF

' Anonymise Member/Non Member Person Data

rsSQL.Edit

' Member Details.

rsSQL!fkTitleID = Null

rsSQL!txtFirstName = "\*\*\*\*\*"

rsSQL!txtMiddleName = IIf(IsNull(Me.txtMiddleName), Null, "\*\*\*\*\*")

rsSQL!txtSurname = IIf(IsNull(Me.txtSurname), Null, "\*\*\*\*\*")

rsSQL!txtNameNo = IIf(IsNull(Me.txtNameNo), Null, "\*\*\*\*\*")

rsSQL!txtAddress1 = IIf(IsNull(Me.txtAddress1), Null, "\*\*\*\*\*")

rsSQL!txtAddress2 = IIf(IsNull(Me.txtAddress2), Null, "\*\*\*\*\*")

rsSQL!txtParishWard = IIf(IsNull(Me.txtParishWard), Null, "\*\*\*\*\*")

rsSQL!txtCityTown = IIf(IsNull(Me.txtCityTown), Null, "\*\*\*\*\*")

rsSQL!fkCountyID = Null

rsSQL!txtHomeTelNo = IIf(IsNull(Me.txtHomeTelNo), Null, "\*\*\*\*\*")

rsSQL!txtMobileNo = IIf(IsNull(Me.txtMobileNo), Null, "\*\*\*\*\*")

rsSQL!txtEMail = IIf(IsNull(Me.txtEMail), Null, "\*\*\*\*\*")

' Emergency Contact Details.

rsSQL!fkECTitleID = Null

rsSQL!txtECFirstName = If(IsNull(Me.txtECFirstName), Null, "\*\*\*\*\*")

rsSQL!txtECMiddleName = If(IsNull(Me.txtECMiddleName), Null, "\*\*\*\*\*")

rsSQL!txtECSurname = If(IsNull(Me.txtECSurname), Null, "\*\*\*\*\*")

rsSQL!txtECHomeTelNo = If(IsNull(Me.txtECHomeTelNo), Null, "\*\*\*\*\*")

rsSQL!txtECEMail = If(IsNull(Me.txtECEMail), Null, "\*\*\*\*\*")

rsSQL!fkECID = Null

' Upgrade GDPR Field.

rsSQL!binGDPR = True

binGDPR.Enabled = False ' Set GDPR Box to False

' Upgrade Membership Cancelled Enabled Field.

binCancelled.Enabled = False

rsSQL.Update

rsSQL.MoveNext ' Moves to next record.

Loop

' Recordset/Database CleanUp.

rsSQL.Close

Set rsSQL = Nothing

' THIS IS THE PART 2 Option of Validation Routine

' Generate Status Info Sub Form Recordset.

sqlStr = "SELECT tblStatus.fkPersonID, tblStatus.dtmSFinish FROM tblPerson " \_

& "INNER JOIN tblStatus ON tblPerson.pkPersonID = tblStatus.fkPersonID " \_

& "WHERE tblPerson.pkPersonID = " & Val(Me![pkPersonID]) & " AND (tblStatus.dtmSFinish)  
Is Null" & " ;"

```
Set rsSQL = dbs.OpenRecordset(sqlStr)
```

```
rsSQL.MoveFirst ' Moves to First Record in Record Set
```

```
If Err.Number = 3021 Then GoTo Proc_CleanUp
```

```
Do While Not rsSQL.BOF And Not rsSQL.EOF
```

```
    ' Update the dtmSFinish field with Today's Date as Default (Can be overwritten with Actual Date).
```

```
        rsSQL.Edit
```

```
        rsSQL!dtmSFinish = Date
```

```
        rsSQL.Update
```

```
        rsSQL.MoveNext ' Moves to next record.
```

```
Loop
```

```
Proc_CleanUp:
```

```
    rsSQL.Close
```

```
    dbs.Close
```

```
    Set dbs = Nothing
```

```
    Set rsSQL = Nothing
```

```
' Clear Any Errors
```

```
    Err.Clear
```

```
' Exit Sub-Routine
```

```
    Exit Sub
```

```
Proc_Err:
```

```
' Generates Error Message
```

```
    If Err.Number = 3021 Then Resume Next ' Checks to see if No Current Record is thrown
```

```
If Err.Number = 2105 Then Resume Next
```

```
MsgBox "Error: (" & Err.Number & ") " & Err.Description & ". E-Mail Error Number and Description  
to rberry425@gmail.com", vbCritical
```

```
GoTo Proc_CleanUp
```

```
End Sub
```

```
Private Sub butAllocatedLeaner_Click()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
Call modTracking("frmAllocated", 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks  
into Stated Form.
```

```
Call modOpenForm("frmAllocated", "dtmDateFrom", 7, 10, "") ' Open designated form and move  
to designated control on form opened.
```

```
ExitHere: ' Any Clean Up Code
```

```
Err.Clear
```

```
Exit Sub
```

```
ErrHandler: ' ERROR HANDLING ROUTINE.
```

```
If Err.Number <> 0 Then
```

```
Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
Resume ExitHere
```

```
End If
```

```
End Sub
```

```
Private Sub butAssetRegister_Click()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
Call modTracking("frmAssetRegisterTracker", 6, Environ("ComputerName"), Environ("UserName"),  
"") ' Tracks into Stated Form.
```

Call modOpenForm("frmAssetRegisterTracker", "dtmDateFrom", 7, 10, "") ' Open designated form and move to designated control on form opened.

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"), Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub butCaseNotes\_Click()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

Call modTracking("frmCaseNotes", 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks into Stated Form.

Call modOpenForm("frmCaseNotes", "fkPersonID", 7, 10, "") ' Open designated form and move to designated control on form opened.

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"), Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub butFormPlans\_Click()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

    Call modTracking("frmPlanReview", 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks into Stated Form.

    Call modOpenForm("frmPlanReview", "fkPlanCodeID", 7, 10, "") ' Open designated form and move to designated control on form opened.

ExitHere: ' Any Clean Up Code

    Err.Clear

Exit Sub

ErrHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"), Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub butHealth\_Click()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

    Call modTracking("frmHealth", 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks into Stated Form.

    Call modOpenForm("frmHealth", "fkPersonID", 7, 10, "") ' Open designated form and move to designated control on form opened.

ExitHere: ' Any Clean Up Code

    Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub butJobDescription\_Click()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

    Call modTracking("frmJob", 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks into  
Stated Form.

    Call modOpenForm("frmJob", "txtJobTitle", 8, 10, "") ' Open designated form and move to  
designated control on form opened.ExitHere: ' Any Clean Up Code

ExitHere: ' Any Clean Up Code

    Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub butPhoto\_Click()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

If Forms!frmLoginScreen!numSecurityLevel >= 8 Then ' Checks Security Clearance

    Call browseFiles("frmPersonnel", "", "txtPhoto", glbstrImageDocumentsPath, "Select JPG File",  
    "\*.jpg", "SELECT PERSONS PHOTO", 1)

End If

ExitHere: ' Any Clean Up Code

    Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
        Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub butSecurity\_Click()

    If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

        Call modTracking("frmLoginSecurity", 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks into Stated Form.

        Call modOpenForm("frmLoginSecurity", "txtSearch", 9, 10, "") ' Open designated form and move to designated control on form opened.

ExitHere: ' Any Clean Up Code

    Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
        Environ("ComputerName"), "", glbHandleErrors)



```

        Resume ExitHere
    End If
End Sub

Private Sub butTitle_Click()
    If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

    Call modTracking("frmTitles", 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks
into Stated Form.

    Call modOpenForm("frmTitles", "Title", 8, 10, "") ' Open designated form and move to designated
control on form opened.

ExitHere: ' Any Clean Up Code

    Err.Clear

    Exit Sub

ErrHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If
End Sub

Private Sub cmbAddNew_Click()
    If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' HOUSE-KEEPING WHICHS LOADS NEW RECORD & TRACKS RECORD ADDITION.
If Forms!frmLoginScreen!numSecurityLevel >= 8 Then ' Checks Security Clearance

    Call addNewRecord("frmPersonnel", "fkTitleID") ' Add New Record Function

    Me!frmStatusSubForm.Form.AllowAdditions = True ' Set Allow Additions Default Setting.

    Me.fkTitleID.SetFocus ' Moves focus to Title Field.

```

Call modTracking(Me.Name, 1, Environ("ComputerName"), Environ("UserName"), "") ' Tracks  
Addition of Record to WLP System.

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub cmdChangePassword\_Click()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Allow User to Change their own password.

RunCommand acCmdUserAndGroupAccounts

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub cmbPostCode\_AfterUpdate()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Updates address fields from an attached postcode table.

Dim ctrl As Control

Set ctrl = Me.ActiveControl

Me.[txtAddress1] = ctrl.Column(1) ' Address Line 1

Me.[txtParishWard] = ctrl.Column(2) ' Address Line 2

Me.[txtCityTown] = ctrl.Column(3) ' Address Line 3

Me.[fkCountyID] = 94 ' Address Line 4

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub cmdClose\_Click()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Tracks into Stated Form.

Call modTracking("frmDoctor", 6, Environ("ComputerName"), Environ("UserName"), "")

' Call Sub-Procedure that opens form requested.

Call modOpenForm("frmDoctor", "cmbGPSurgery", 7, 10, "2")

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub Combo140\_AfterUpdate()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Updates address fields from an attached postcode table.

Dim ctrl As Control

Set ctrl = Me.ActiveControl

Me.[txtAddress1] = ctrl.Column(1) ' Address Line 1

Me.[txtParishWard] = ctrl.Column(5) ' Ward/Parish

Me.[txtCityTown] = ctrl.Column(3) ' City/Town

Me.[fkCountyID] = 94 ' Cheshire

Me.Refresh

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub cmdGP\_Click()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Tracks into Stated Form.

    Call modTracking("frmDoctor", 6, Environ("ComputerName"), Environ("UserName"), "")

' Call Sub-Procedure that opens form requested.

    Call modOpenForm("frmDoctor", "cmbGPSurgery", 8, 10, "2")

ExitHere: ' Any Clean Up Code

    Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub Command145\_Click()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Open designated form and move to designated control on form opened.

Call modOpenForm("frmLoginSecurity", "txtSearch", 7, 10, "")

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub Command214\_Click()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Tracks into Stated Form.

Call modTracking("frmPlanReview", 6, Environ("ComputerName"), Environ("UserName"), "")

' Call Sub-Procedure that opens form requested.

Call modOpenForm("frmPlanReview", "dtmDateFrom", 7, 10, "2")

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

```
    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
    Resume ExitHere
```

```
End If
```

```
End Sub
```

```
Private Sub Command319_Click()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
If Forms!frmLoginScreen!numSecurityLevel >= 8 Then ' Checks Security Clearance
```

```
    Call modOpenForm("frmPrintMenu", "cmbCategory", 7, 10, 2) ' Open designated form and move  
to designated control on form opened. Also Set Background Colour.
```

```
End If
```

```
ExitHere: ' Any Clean Up Code
```

```
Err.Clear
```

```
Exit Sub
```

```
ErrHandler: ' ERROR HANDLING ROUTINE.
```

```
If Err.Number <> 0 Then
```

```
    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
    Resume ExitHere
```

```
End If
```

```
End Sub
```

```
Private Sub dtmDOB_AfterUpdate()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
' Checks Date of Birth
```

```
If dtmDOB >= Date Then
```

```
    MsgBox "Check Date of Birth"
```

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub fkECID\_DblClick(Cancel As Integer)

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Call Sub-Procedure that opens form requested.

Call modOpenForm("frmRelationship", "txtRelationship", 8, 10, "")

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub



```

Private Sub fkECTitleID_DblClick(Cancel As Integer)
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Call Sub-Procedure that opens form requested.
    Call modOpenForm("frmTitles", "Title", 8, 10, "")

ExitHere: ' Any Clean Up Code
    Err.Clear
Exit Sub

ErrHandler: ' ERROR HANDLING ROUTINE.
    If Err.Number <> 0 Then
        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),
Environ("ComputerName"), "", glbHandleErrors)
        Resume ExitHere
    End If
End Sub

```

```

Private Sub fkTitleID_DblClick(Cancel As Integer)
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Call Sub-Procedure that opens form requested.
    Call modOpenForm("frmTitles", "txtSearch", 8, 10, "1")

ExitHere: ' Any Clean Up Code
    Err.Clear
Exit Sub

ErrHandler: ' ERROR HANDLING ROUTINE.
    If Err.Number <> 0 Then

```

```
Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
Resume ExitHere
```

```
End If
```

```
End Sub
```

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
Dim strMessage As String, Response
```

```
If Me.NewRecord = True Then ' Validates if New Record
```

```
' Validate that Surname has been entered.
```

```
If IsNull(txtSurname) Then
```

```
' Initialise Message String.
```

```
strMessage = "Cannot Save Record Without Surname." & vbCrLf _  
    & "Select YES to return to Surname Field." & vbCrLf _  
    & "Select NO to Abandon Change(s). Return to Previous Record."
```

```
Response = MsgBox(strMessage, vbYesNo)
```

```
If Response = vbNo Then
```

```
Cancel = True ' Cancel Record Save/Update
```

```
Me.Undo
```

```
Me.Bookmark = strBookMark ' Return to Previous Record
```

```
Me.Refresh
```

```
Me.txtSearch.SetFocus
```

```
GoTo ExitHere:
```

```
Else
```

```
Me.txtSurname.SetFocus
```

```

        GoTo ExitHere:

    End If

End If

' Asks User to Save Record or Not!

Response = MsgBox("Save New Record?", vbYesNo)

If Response = vbNo Then

    Cancel = True

    Me.Undo

End If

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),
Environ("ComputerName"), "", glbHandleErrors)

    Resume ExitHere

End If

End Sub

Private Sub Form_Close()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' SET PERSISTENT CONNECTION (CLOSE) TO WLP SYSTEM DATABASES.

OpenAllDatabases False

ExitHere: ' Any Clean Up Code

```

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
    Environ("ComputerName"), "", glbHandleErrors)

    Resume ExitHere

End If

End Sub

Private Sub Form\_Current()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' SET VISIBLE PROPERTY FOR FOLLOWING FIELDS ON CURRENT RECORD

    Me.lblConfirmEMail.Visible = False

    Me.lblCopyPaste.Visible = False

    Me.txtConfirmEMail.Visible = False

' CHECKS WHETHER CANCELLED BOXES IS CHECKED AND IF SO SETS ENABLED PROPERTY TO FALSE.

' Enable/Disable Controls below.

If Me.binCancelled = True Then

    Me.binCancelled.Enabled = False

Else

    Me.binCancelled.Enabled = True

End If

If Me.binGDPR = True Then

    Me.binGDPR.Enabled = False

Else

    Me.binGDPR.Enabled = True

End If

' CHECK FOR NO PHOTO IMAGE DISPLAYED.

If IsNull(Me.txtPhoto) Then

Me.txtPhoto = glbstrImageDocumentsPath & "NoPhoto.jpg"

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub Form\_KeyDown(KeyCode As Integer, Shift As Integer)

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

Dim intCtrlDown As Integer

intCtrlDown = (Shift And acCtrlMask) > 0

' Assesses whether Ctrl & V is enabled or disabled. Relates to Members e mail and confrimatory E  
Mail Control Fields

If KeyCode = vbKeyV And (Screen.ActiveControl.Name = "txtConfirmEMail" Or  
Screen.ActiveControl.Name = "txtEMail") Then

If intCtrlDown Then

KeyCode = 0

End If

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub Form\_Load()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Dimension DAO Objects and local variables.

Dim dbs As DAO.Database

Dim rs As DAO.Recordset

' Initialise Dimensioned Variables

Set dbs = CurrentDb

Set rs = Me.RecordsetClone

' INITIAL PERSONS RECORD LOADED DEPENDENT ON LOGIN DETAILS

rs.FindFirst "pkPersonID=" & Forms!frmLoginScreen!fkID

Me.Bookmark = rs.Bookmark

Me.txtSearch.SetFocus

Me.txtSearch.Value = Forms!frmLoginScreen!fkID ' Populate txtSearch combo field when form  
initially loaded.

' THIS SECTION SETS WHICH TABS A USER CAN ACCESS.

' Initial default position of all tabs on WLP System.

```
Forms!frmPersonnel.TabCtPersonnel.Pages("pgStaff").Visible = False
Forms!frmPersonnel.TabCtPersonnel.Pages("pgVolunteer").Visible = False
Forms!frmPersonnel.TabCtPersonnel.Pages("pgHumanResource").Visible = False
Forms!frmPersonnel.TabCtPersonnel.Pages("pgLearner").Visible = False
Forms!frmPersonnel.TabCtPersonnel.Pages("pgAttendance").Visible = False
Forms!frmPersonnel.TabCtPersonnel.Pages("pgAdmin").Visible = False
```

' Set Tabs Visibility Property dependent on Security Clearance.

Select Case Forms!frmLoginScreen!numSecurityLevel

Case 1 To 2 ' Support Staff

```
Forms!frmPersonnel.TabCtPersonnel.Pages("pgContact").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgPersonal").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgLearner").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgLearner1").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgAttendance").Visible = True
```

Case Is = 5 'Supervisors

```
Forms!frmPersonnel.TabCtPersonnel.Pages("pgContact").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgPersonal").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgLearner1").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgStaff").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgHumanResource").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgLearner").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgAttendance").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgAdmin").Visible = True
```

Case 8 To 10 ' Senior Management Team and Developer

```
Forms!frmPersonnel.TabCtPersonnel.Pages("pgContact").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgPersonal").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgLearner1").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgStaff").Visible = True
```

```
Forms!frmPersonnel.TabCtPersonnel.Pages("pgVolunteer").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgHumanResource").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgLearner").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgAttendance").Visible = True
Forms!frmPersonnel.TabCtPersonnel.Pages("pgAdmin").Visible = True
```

End Select

' FURTHER HOUSEKEEPING ROUTINES

Call modTracking(Me.Name, 6, Environ("ComputerName"), Environ("UserName"), "") ' Tracks Entry into Form

Call subCtrlLockUnlock(Form.Name, "", "", "UL", 8, 10, 1) ' Call Control Locking Sub-Procedure on for form

Call txtSearchRowSource(Form.Name, "", "", "txtSearch", 1) ' Set Row Source for combo control txtSearch

Call MaximiseScreen ' Maximise the Form

OpenAllDatabases True ' Set Persistent Connection (Open) to WLP System Databases.

Me.TabCtPersonnel = 0 ' Set Active Tab Page to Contact Tab

If binCancelled Then

    binCancelled.Enabled = False

Else

    binCancelled.Enabled = True

End If

If binGDPR Then

    binGDPR.Enabled = False

Else

    binGDPR.Enabled = True

End If



ExitHere: ' Any Clean Up Code

rs.Close

dbs.Close

Set rs = Nothing

Set dbs = Nothing

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub frmStatusSubForm\_Enter()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Checks to see Status Record Already exists for designated Person

Dim dbs As DAO.Database ' Dimension Database

Dim rs As DAO.Recordset ' Dimesion Recordset

Dim strSQL As String ' Dimension SQL Statement

Dim strMessage As String ' Dimension Message String.

' Initialise recordset and variable

Set dbs = CurrentDb ' Initialise a reference to the current database

strSQL = "SELECT tblStatus.pkStatusID, tblStatus.fkPersonID FROM tblStatus " \_

& "WHERE NZ(tblStatus.fkPersonID,0) = " & Nz([Forms]![frmPersonnel]![pkPersonID], 0) & ";"

```
Set rs = dbs.OpenRecordset(strSQL) ' Initialise Recordset to determine if Available.
```

```
' Check to see if Surname Has been entered if not then return to Surname field
```

```
    If IsNull(Me.txtSurname) Then
```

```
        MsgBox "Enter Surname to Continue"
```

```
        Me.txtSurname.SetFocus
```

```
        GoTo ExitHere:
```

```
    End If
```

```
If rs.RecordCount = 1 Then ' Set Status of Record Additions
```

```
    Me!frmStatusSubForm.Form.AllowAdditions = False
```

```
Else
```

```
    Me!frmStatusSubForm.Form.AllowAdditions = True
```

```
End If
```

```
' Set level of User Input Control to Record dependent on field completion
```

```
    If Not IsNull(Me!frmStatusSubForm.Form!txtStatusDesc) Then ' Descriptor Field
```

```
        Me!frmStatusSubForm.Form!txtStatusDesc.Locked = True
```

```
    End If
```

```
    If Not IsNull(Me!frmStatusSubForm.Form!dtmSStart) Then ' Start Date
```

```
        Me!frmStatusSubForm.Form!dtmSStart.Locked = True
```

```
    End If
```

```
    If Not IsNull(Me!frmStatusSubForm.Form!dtmSFinish) Then ' Start Date
```

```
        Me!frmStatusSubForm.Form!dtmSFinish.Locked = True
```

```
    End If
```

```
ExitHere: ' Any Error Clean Up Code
```

```
rs.Close
```

```
dbs.Close
```

Set rs = Nothing

Set dbs = Nothing

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
    Environ("ComputerName"), "", glbHandleErrors)

    Resume ExitHere

End If

End Sub

Private Sub frmStatusSubForm\_Exit(Cancel As Integer)

' Error Trapping System Initiated.

    On Error GoTo Proc\_Err

If Me.Dirty Then

' Check to see if Records in Sub Form have the necessary Status and Start Dates completed for this  
Person.

' Dimension DAO Objects and local variables.

    Dim wks As DAO.Workspace

    Dim dbs As DAO.Database

    Dim rs As DAO.Recordset

    Dim strSQL As String

    Dim numRecordCount As Long ' Marker Tags Record

' Initialise Workspace, Database, QueryDef, Recordset and Ordinary Variables

    Set wks = DBEngine.Workspaces(0)

    Set dbs = CurrentDb

```
strSQL = "SELECT tblStatus.fkPersonID, tblStatus.txtStatusDesc, tblStatus.dtmSStart FROM  
tblStatus WHERE " _
```

```
& "[tblStatus.fkPersonID] = " & Val(Nz(Me![pkPersonID], 0)) & ";"
```

```
Set rs = dbs.OpenRecordset(strSQL)
```

```
rs.MoveLast ' Go to Last Record in RecordSet.
```

```
' Checks to see if there are any Status Records
```

```
If rs.RecordCount = 0 Then
```

```
    MsgBox "Enter Status Details"
```

```
    GoTo Proc_CleanUp
```

```
Else
```

```
' Validate that the Status Records have a Status and Start Date Associated with Them.
```

```
If Not IsNull(pkPersonID) Then
```

```
' Generate Status Info Sub Form Recordset.
```

```
    rs.MoveFirst ' Moves to First Record in RecordSet
```

```
Do While Not rs.BOF And Not rs.EOF
```

```
' Check to see if Status or Start Date Date for WLP Client is Null
```

```
    If IsNull(rs!txtStatusDesc) Or IsNull(rs!dtmSStart) Then
```

```
        MsgBox "Either Status or Start Date missing from Status Info Sub Form Records"
```

```
    ' Clean Up Database
```

```
        GoTo Proc_CleanUp
```

```
    Else
```

```
        rs.MoveNext ' Moves to next record.
```

```
    End If
```

```
Loop
```

```
End If
```

```
End If
```

Proc\_CleanUp:

rs.Close  
dbs.Close  
wks.Close

Set rs = Nothing  
Set wks = Nothing  
Set dbs = Nothing  
Err.Clear ' Clear Any Errors  
Exit Sub ' Exit Sub-Routine

Proc\_Err:

' Generates Error Message

If Err.Number = 94 Then Resume Next ' Checks to see if No Current Record is thrown

If Err.Number = 3021 Then Resume Next ' Checks to see if No Current Record is thrown

MsgBox "Error: (" & Err.Number & ") " & Err.Description & ". E-Mail Error Number and Description  
to rberry425@gmail.com", vbCritical

GoTo Proc\_CleanUp

End If

End Sub

Private Sub frmSubEContacts\_Enter()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Check to see if Surname Has been entered if not then return to Surname field

If IsNull(Me.txtSurname) Then

MsgBox "Enter Surname to Continue"

Me.txtSurname.SetFocus

End If

ExitHere: ' Any Error Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub frmSubLearner\_Enter()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Checks to see if this is a Learners Record.

If Not Forms("frmPersonnel").Form("frmStatusSubForm").Form("txtStatusDesc") = "Learner" Then

Forms!frmPersonnel.TabCtPersonnel = 0 ' Moves to Contact Tab

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

```
Private Sub frmSubLearnerTab_Enter()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
' Checks to see if this is a Learners Record.
```

```
    If Not Forms("frmPersonnel").Form("frmStatusSubForm").Form("txtStatusDesc") = "Learner" Then
```

```
'        Forms!frmPersonnel.TabCtPersonnel = 0 ' Moves to Contact Tab
```

```
    End If
```

```
ExitHere: ' Any Clean Up Code
```

```
    Err.Clear
```

```
    Exit Sub
```

```
ErrHandler: ' ERROR HANDLING ROUTINE.
```

```
    If Err.Number <> 0 Then
```

```
        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
        Resume ExitHere
```

```
    End If
```

```
End Sub
```

```
Private Sub frmVolunteerSubForm_Enter()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
' Checks to see if this is a Volunteer Record.
```

```
    If Not Forms("frmPersonnel").Form("frmStatusSubForm").Form("txtStatusDesc") = "Volunteer"  
Then
```

```
        Forms!frmPersonnel.TabCtPersonnel = 0 ' Moves to Contact Tab
```

```
    End If
```

```
ExitHere: ' Any Clean Up Code
```

```
    Err.Clear
```

```
    Exit Sub
```

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub Label150\_DblClick(Cancel As Integer)

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Call Sub-Procedure that opens form requested.

    Call modOpenForm("frmPostCode", "txtSearch", 7, 10, "")

ExitHere: ' Any Clean Up Code

    Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub numHearAbout\_DblClick(Cancel As Integer)

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Call Sub-Procedure that opens form requested.

    Call modOpenForm("frmHeardAbout", "txtSearch", 7, 10, "1")



ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub Text148\_DblClick(Cancel As Integer)

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Call Sub-Procedure that opens form requested.

Call modOpenForm("frmPostCode", "txtSearch", 7, 10, "")

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

```
Private Sub pgContact_Click()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
' Forms("frmPersonnel")("frmSubLearner").Form("fkPersonID") =  
Forms("frmPersonnel").Form("pkPersonID")
```

```
ExitHere: ' Any Clean Up Code
```

```
Err.Clear
```

```
Exit Sub
```

```
ErrHandler: ' ERROR HANDLING ROUTINE.
```

```
If Err.Number <> 0 Then
```

```
Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
Resume ExitHere
```

```
End If
```

```
End Sub
```

```
Private Sub pgStaff_Click()
```

```
If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling
```

```
' Checks to see if this is a Staff Record.
```

```
If Not Forms("frmPersonnel").Form("frmStatusSubForm").Form("txtStatusDesc") Like "Staff*" Then
```

```
Forms!frmPersonnel.TabCtPersonnel = 0 ' Moves to Contact Tab
```

```
End If
```

```
ExitHere: ' Any Clean Up Code
```

```
Err.Clear
```

```
Exit Sub
```

```
ErrHandler: ' ERROR HANDLING ROUTINE.
```

```

If Err.Number <> 0 Then

    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),
Environ("ComputerName"), "", glbHandleErrors)

    Resume ExitHere

End If

End Sub

```

```

Private Sub TabCtPersonnel_Change()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

```

```

    Dim dbs As DAO.Database ' Dimension Database
    Dim rs As DAO.Recordset ' Dimesion Recordset
    Dim strSQL As String ' Dimension SQL Statement

```

```

    Set dbs = CurrentDb ' Initialise a reference to the current database

```

```

    strSQL = "SELECT tblStatus.pkStatusID, tblStatus.fkPersonID FROM tblStatus " _
        & "WHERE NZ(tblStatus.fkPersonID,0) = " & Nz([Forms]![frmPersonnel]![pkPersonID], 0) & ";"

```

```

    Set rs = dbs.OpenRecordset(strSQL) ' Initialise Recordset to determine if Available.

```

```

' 1. Checks to ensure Surname field has been completed.

```

```

    If IsNull(Me.txtSurname) Then

        MsgBox "Complete Surname field to Access Other Tabs"

        Me.txtSurname.SetFocus ' Returns focus to Surname Field on Contact Tab

        Me.TabCtPersonnel = 0 ' Moves to Contact Tab

        GoTo ExitHere:

    End If

```

```

' 2. Checks to see that Status Sub Form exists.

```

```

    If rs.RecordCount = 0 Then

```

```

        MsgBox "Complete Status and Start Date fields of Status Info Sub Form to Access Other Tabs"

        Me.TabCtPersonnel = 0 ' Moves to Contact Tab

        Me!frmStatusSubForm.Form.txtStatusDesc.SetFocus ' Returns focus to first field of Status Info
Sub Form

    Else

        ' Checks to see that BOTH Type and Start Date fields are complete.

        If IsNull(Me!frmStatusSubForm.Form.txtStatusDesc) Or
IsNull(Me!frmStatusSubForm.Form.dtmSStart) Then

            MsgBox "Complete Status and Start Date fields of Status Info Sub Form to Access Other Tabs"

            Me.TabCtPersonnel = 0 ' Moves to Contact Tab

            If IsNull(Me!frmStatusSubForm.Form.txtStatusDesc) Then

                Me!frmStatusSubForm.Form.txtStatusDesc.SetFocus

            Else

                Me!frmStatusSubForm.Form.dtmSStart.SetFocus

            End If

        End If

    End If

End If

```

```

ExitHere: ' Any Error Clean Up Code

```

```

    rs.Close

    dbs.Close

    Set rs = Nothing

    Set dbs = Nothing

    Err.Clear

    Exit Sub

```

```

ErrorHandler: ' ERROR HANDLING ROUTINE.

```

```

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),
Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

```

End Sub

Private Sub Text282\_Enter()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Checks to see if this is a Staff Record.

If Not Forms("frmPersonnel").Form("frmStatusSubForm").Form("txtStatusDesc") Like "Staff\*" Then

Forms!frmPersonnel.TabCtPersonnel = 0 ' Moves to Contact Tab

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"), Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub txtConfirmEMail\_Exit(Cancel As Integer)

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Validate e mail address against confirmatory e mail

If LTrim(Me.txtEMail) = LTrim(Me.txtConfirmEMail) Then

Me.lblConfirmEMail.Visible = False

```

        Me.txtGender.SetFocus

        Me.txtConfirmEMail.Visible = False

        ' Set Visible Instruction Label for Confirmatory E Mail Address

        Me.lblCopyPaste.Visible = False

        Exit Sub

    Else

        MsgBox ("E Mail Addresses Don't Match")

        Me.txtGender.SetFocus

    End If

ExitHere: ' Any Clean Up Code

    Err.Clear

    Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

    If Err.Number <> 0 Then

        Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),
        Environ("ComputerName"), "", glbHandleErrors)

        Resume ExitHere

    End If

End Sub

Private Sub txtEMail_AfterUpdate()

    If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

    ' Make unbound confirmatory e mail label and text box visible and set focus to text box.

    Me.lblConfirmEMail.Visible = True ' Label for Confirmatory E Mail Control

    Me.txtConfirmEMail.Visible = True ' Confirmatory E Mail Address control (Set Visible)

    Me.txtConfirmEMail.SetFocus

    Me.txtConfirmEMail = Null ' Set Confirmatory EMail box to null

```

' Set Visible Instruction Label for Confirmatory E Mail Address

Me.IblCopyPaste.Visible = True

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub txtRefID\_Enter()

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Checks to see if have adequate security level to access the HR Tab.

If Forms!frmLoginScreen!numSecurityLevel < 8 Then

Forms!frmPersonnel.TabCtPersonnel = 0 ' Moves to Contact Tab

End If

ExitHere: ' Any Clean Up Code

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub txtSearch\_AfterUpdate()

If glbHandleErrors Then On Error GoTo ErrHandler ' Set Error Handling

' Dimension DAO Objects and local variables.

Dim dbs As DAO.Database

Dim rs As DAO.Recordset

Dim strSQL As String

' Initialise Dimensioned Variables

Set dbs = CurrentDb

Set rs = Me.RecordsetClone

' GOTO RECORD SELECTED BY COMBO FIELD TXTSEARCH

rs.FindFirst "pkPersonID=" & Forms!frmPersonnel!txtSearch.Column(0)

Me.Bookmark = rs.Bookmark

' CHECKS TO SEE IF STATUS RECORD ALREADY EXISTS FOR THIS RECORD

strSQL = "SELECT tblStatus.pkStatusID, tblStatus.fkPersonID FROM tblStatus " \_  
& "WHERE NZ(tblStatus.fkPersonID,0)= " & [Forms]![frmPersonnel]![pkPersonID] & ";"

Set rs = dbs.OpenRecordset(strSQL) ' Initialise Recordset to determine if Available.

' Set Status of Record Additions

If rs.RecordCount = 1 Then

Forms("frmPersonnel").frmStatusSubForm.Form.AllowAdditions = False

Else

Forms("frmPersonnel").frmStatusSubForm.Form.AllowAdditions = True

End If



' Call Control Locking Sub-Procedure for form.

Call subCtrlLockUnlock("frmPersonnel", "frmStatusSubForm", "", "UL", 8, 10, 2)

ExitHere: ' Any Clean Up Code

Set dbs = Nothing

Set rs = Nothing

Err.Clear

Exit Sub

ErrorHandler: ' ERROR HANDLING ROUTINE.

If Err.Number <> 0 Then

Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)

Resume ExitHere

End If

End Sub

Private Sub txtSearch\_Exit(Cancel As Integer)

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

If glbHandleErrors Then On Error GoTo ErrorHandler ' Set Error Handling

' Dimension DAO Objects and local variables.

Dim dbs As DAO.Database

Dim rs As DAO.Recordset

' Initialise Dimensioned Variables

Set dbs = CurrentDb

Set rs = Me.RecordsetClone

' POPULATE CURRENT RECORD POSITION INTO STRBOOKMARK (In Declaration Section)

rs.FindFirst "pkPersonID=" & Forms!frmPersonnel!txtSearch.Column(0)

```
strBookMark = rs.Bookmark
```

```
If IsEmpty(strBookMark) Then MsgBox " Bookmark is Empty"
```

```
Me.Refresh
```

```
ExitHere: ' Any Clean Up Code
```

```
rs.Close
```

```
dbs.Close
```

```
Set rs = Nothing
```

```
Set dbs = Nothing
```

```
Err.Clear ' Clear Any Errors
```

```
Exit Sub ' Exit Sub-Routine
```

```
ErrorHandler: ' ERROR HANDLING ROUTINE.
```

```
If Err.Number <> 0 Then
```

```
    Call LogError(Err.Number, Err.Description, Forms!frmLoginScreen!fkID, Environ("UserName"),  
Environ("ComputerName"), "", glbHandleErrors)
```

```
    Resume ExitHere
```

```
End If
```

```
End Sub
```