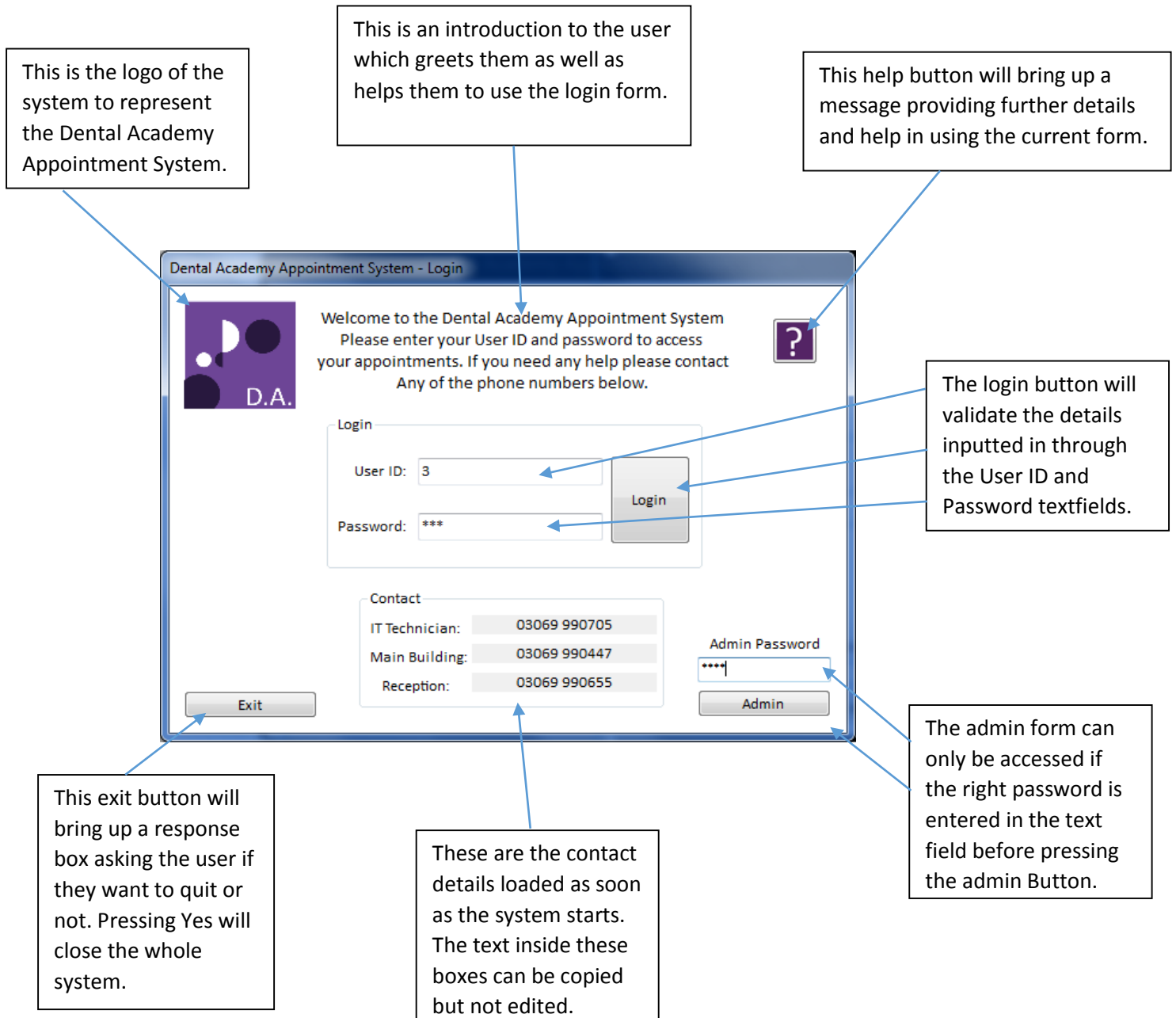


Implemented User Interface

frmLogin



```
Imports System.Data.SqlClient
Public Class frmLogin

    Private Sub frmLogin_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        'On form load I will call in the subroutine in order to connect to the database.'
        Connect()
        'I set the properties of each of the three text boxes so that it can only show a phone number which cannot be manipulated and only copied.'
        txtPhone1.Text = "03069 990705"
        txtPhone1.BorderStyle = BorderStyle.None
        txtPhone1.ReadOnly = True
        txtPhone1.SelectionStart() = 0
        txtPhone1.SelectionLength = txtPhone1.Text.Length
        txtPhone1.ScrollToCaret()
        txtPhone1.Focus()

        txtPhone2.Text = "03069 990447"
        txtPhone2.BorderStyle = BorderStyle.None
        txtPhone2.ReadOnly = True
        txtPhone2.SelectionStart() = 0
        txtPhone2.SelectionLength = txtPhone2.Text.Length
        txtPhone2.ScrollToCaret()
        txtPhone2.Focus()

        txtPhone3.Text = "03069 990655"
        txtPhone3.BorderStyle = BorderStyle.None
        txtPhone3.ReadOnly = True
        txtPhone3.SelectionStart() = 0
        txtPhone3.SelectionLength = txtPhone1.Text.Length
        txtPhone3.ScrollToCaret()
        txtPhone3.Focus()
    End Sub

    Private Sub btnAdmin_Click(sender As System.Object, e As System.EventArgs) Handles btnAdmin.Click
        'If the answer is "dent" then the current form will hide and the admin form will show.'
        If txtAdminPassword.Text = "dent" Then
            Me.Hide()
            frmAdmin.Show()
            'Makes sure that the dataset is removed so that an updated dataset will load when re-entering the form.'
            If ds.Tables.Contains("tblDentists") Then
                ds.Tables.Remove("tblDentists")
            End If
            'clear the textfield so that if another user logs, on they will need to re-enter the password to access the admin form.'
            txtAdminPassword.Clear()
        Else
            'If the answer is incorrect, a message box will be displayed and the input box will close.'
            MessageBox.Show("Incorrect Password")
            txtAdminPassword.Clear()
        End If
    End Sub

    Private Sub btnExit_Click(sender As System.Object, e As System.EventArgs) Handles btnExit.Click
```

```

        'I dim a message box to allow the user to decide incase they accidentally close
the program.'
        Dim response As MsgBoxResult
        response = MsgBox("Do you want to close the program?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Confirm")
        'If the response is yes, the program will close
        If response = MsgBoxResult.Yes Then
            Me.Dispose()
            'If the response is no, the program will remain open.'
        ElseIf response = MsgBoxResult.No Then
            Exit Sub
        End If
    End Sub

    Private Sub btnLogin_Click(sender As System.Object, e As System.EventArgs) Handles
btnLogin.Click
        'This calls the function to validate the login details in the textfields.'
        If ValidateLogin(txtUserID, txtPassword) = True Then
            'the form will hide if the function returns as true.'
            Me.Hide()
            'The menu form will show.'
            frmMenu.Show()
            'This will make sure that the logged in dentists name will load onto the
menu form.'
            DentistID = ds.Tables("tblDentists").Rows(0).Item("DentistID")
            frmMenu.txtDTitle.Text = ds.Tables("tblDentists").Rows(0).Item("DTitle")
            frmMenu.txtDFirstName.Text =
ds.Tables("tblDentists").Rows(0).Item("DFirstName")
            frmMenu.txtDSurname.Text =
ds.Tables("tblDentists").Rows(0).Item("DSurname")
            'clear the textfield so that if another user logs, on they will need to
re-enter the password to access the admin form.'
            txtAdminPassword.Clear()
        End If
    End Sub

    Public Function ValidateLogin(ByVal txtUserID As TextBox, ByVal txtPassword As
TextBox) As Boolean
        AllFieldsValid = True
        'validation to check that the password and username are entered.'
        If String.IsNullOrEmpty(txtUserID.Text) Or
String.IsNullOrEmpty(txtPassword.Text) Then
            MessageBox.Show("Please enter your MemberID and a Password.")
            AllFieldsValid = False
            'Further validation checks that the username is numeric'
        ElseIf Not IsNumeric(txtUserID.Text) Then
            'Validation to check if there are any numeric values in the Username'
            MessageBox.Show("UserID must be numeric...")
            AllFieldsValid = False
        ElseIf txtUserID.Text < 1 Then
            'Validation to see if the text ID is less than 1 or more than 30'
            MessageBox.Show("ID cannot be less than 1.")
            AllFieldsValid = False
        ElseIf txtUserID.Text > 100 Then
            'Validation for textlength on the member ID. The ID must not be larger
than 30 or less than 0.'
            MessageBox.Show("ID cannot be bigger than 100.")
            AllFieldsValid = False
        End If
        'If all fields value is true then the user will be able to log in.'
        If AllFieldsValid = True Then
            'Dims the log member as the User ID text.'

```

```
LogMember = txtUserID.Text
'Selects all from the Dentists table where the Dentist ID matches the ID
text field.'
Sql = "select * from tblDentists where DentistID= " & LogMember
Try
    ds.Clear()
Catch ex As Exception
    'Ignore first time.'
End Try
'Creates a new sql data adapter for the dataset.'
da = New SqlDataAdapter(Sql, dbconn)
'Fills the dataset.'
da.Fill(ds, "tblDentists")
'Makes sure the password is from the user logging on.'
If ds.Tables("tblDentists").Rows.Count > 0 Then
    If ds.Tables("tblDentists").Rows(0).Item("DPassword") =
txtPassword.Text Then
        DentistID = LogMember
        'If this boolean value is true then It will return the function as
true which can be called in the login button.'
        ValidateLogin = True
    Else
        'The value is returned as false therefore the user cannot proceed
to the next form.'
        ValidateLogin = False
        MessageBox.Show("Password is incorrect or doesn't match ID.")
    End If
Else
    'Error message saying the ID doesn't exist.'
    MessageBox.Show("ID does not exist.")
End If
End If
End Function

Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHelp.Click
    'The help button will display a message helping the user with the form.'
    MessageBox.Show("Use your personal ID that has been set to you into the the
text fields. Passwords must be exactly how you set it. If you have any technical
difficulties, please contact the administrations office. Receptionists can access the
admin form by entering the correct password which is written on the counter.")
End Sub
End Class
```

frmAdmin

This image will change depending on which radio button is selected. It will load when an existing user is loaded into the form.

The show button will load the selected user into the combo box and their details will load onto the text fields. The edit button will enable all text fields allowing the user to input updated details. The save button will be enabled allowing validation and updated details to be saved back onto the database.

This help button will bring up a message providing further details and help in using the current form.

Dental Academy Appointment System - Admin

ID: 3 Rosie

Show

STUDENT

Edit Save Cancel Delete

☒ Student ☐ Tutor ☐ Dental Nurse

Title: Miss New Password: ***

First Name: Rosie Retype Password: ***

Surname: Van

Date Of Birth: 12 September 1997

Gender M / F: F

Phone Number: 07564758694

Home Telephone: 07423672672

Log Off

Register New Member

Add

The delete button will delete the existing user off the database.

This button will close the current form and return back to the login form.

This button will disable the functions to edit, save and delete an existing user.

The add button will call in validation to check each text field to see if the data is correctly inputted. This will then add a new row into the dataset.

```
Imports System.Data.SqlClient
Public Class frmAdmin
    'Sets the selected dentist's name as a string.'
    Dim SelecteddentistsName As String
    'Makes the value used to count the dataset rows as an integer.'
    Dim Countrows As Integer

    Private Sub frmAdmin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Checks if the Dentist name table is present and removes it.'
        If ds.Tables.Contains("tblDentistsName") Then
            ds.Tables.Remove("tblDentistsName")
        End If
        'Makes sure that the combo box is reset.'
        cboDentist.SelectedIndex = -1
        'When the form loads the buttons will be set to their default state.'
        btnAdd.Enabled = False
        btnDelete.Enabled = False
        btnEdit.Enabled = False
        btnSave.Enabled = False
        cboDentist.Enabled = True
        btnShow.Enabled = True
        btnRegister.Enabled = True
        lblColour.Visible = False

        'This makes sure that the combo box for the dentists title is disabled.'
        cboDTitle.Enabled = False
        'I make the text fields read-only so that the user cannot enter anything into
them.'
        txtDFirstName.ReadOnly = True
        txtDSurname.ReadOnly = True
        txtDGender.ReadOnly = True
        txtDPhoneNumber.ReadOnly = True
        txtDHomeTelephone.ReadOnly = True
        txtNewPassword.ReadOnly = True
        txtReTypePassword.ReadOnly = True
        DDateOfBirth.Enabled = False

        'This makes sure that the radio buttons cannot be checked.'
        rdoStudent.Enabled = False
        rdoDentalNurse.Enabled = False
        rdoTutor.Enabled = False
        rdoStudent.Checked = True

        'This variable is used to count the rows in a dataset.'
        Dim CountRows As Integer
        'This variable is used loop until I can find the row I'm looking for.'
        Dim i As Integer

        'Defaults the staff image to the student by default.'

        picStaff.Image = My.Resources.picStudent

        'This sql statement will retrieve all the dentist ID's onto the combo box.'
        Sql = "SELECT * FROM tblDentists"
        da = New SqlDataAdapter(Sql, dbconn)
        Try
            'Clears the dataset to make sure that the data is not retrieved twice.'
```

```
        ds.Tables("tblDentists").Clear()
    Catch ex As Exception

End Try
da.Fill(ds, "tblDentistsName")
CountRows = ds.Tables("tblDentistsName").Rows.Count
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1
    cboID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))
cboDentist.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
Next

' This makes sure that the radio buttons will change the label so that the
label text will change the actual staff image.
If rdoDentalNurse.Checked = True Then
    lblColour.Text = "Blue"
ElseIf rdoTutor.Checked = True Then
    lblColour.Text = "Red"
ElseIf rdoStudent.Checked = True Then
    lblColour.Text = "Green"
End If

End Sub

Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLogOff.Click
    'Closes the current form.'
    Me.Close()
    'Returns to the login form.'
    frmLogin.Show()
    'Makes sure the title is reset.'
    cboDTitle.SelectedIndex = -1
    'Clears all textfields.'
    txtDFirstName.Clear()
    txtDSurname.Clear()
    txtDGender.Clear()
    txtDPhoneNumber.Clear()
    txtNewPassword.Clear()
    txtReTypePassword.Clear()
    'Clears all combo boxes.'
    cboDentist.Items.Clear()
    cboID.Items.Clear()

    'Clears the dataset.'
    Try
        If ds.Tables("tblDentists").Rows.Count > 0 Then
            ds.Tables("tblDentists").Clear()
        End If
    Catch ex As Exception
    End Try

End Sub

Private Sub rdoTutor_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rdoTutor.CheckedChanged
    'Sets the image to tutor image when radio button is checked.'
    picStaff.Image = My.Resources.picTutor
End Sub

Private Sub rdoStudent_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rdoStudent.CheckedChanged
    'Sets the image to student image when radio button is checked.'
```

```

        picStaff.Image = My.Resources.picStudent
    End Sub

    Private Sub rdoDentalNurse_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles rdoDentalNurse.CheckedChanged
        'Sets the image to dental nurse image when radio button is checked.'
        picStaff.Image = My.Resources.picDentalNurse
    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
        'Calls in the function to validate the text fields.'
        If ValidateStaff(cboDTitle, txtDFirstName, txtDSurname, txtDGender,
txtDPhoneNumber, txtDHomeTelephone, txtNewPassword, txtReTypePassword) = True Then
            'If the function is returned as true then the sql will run and clear the
dataset.'
            Dim Sql As String = ""
            Try
                ds.Tables("tblDentist").Clear()
            Catch ex As Exception
            End Try

            Try
                ds.Tables("tblNewDentist").Clear()
            Catch ex As Exception
            End Try
            'This statement will create a new row in the dentist table.'
            Sql = "Select * from tblDentists"
            'Creates a new data adapter.'
            da = New SqlDataAdapter(Sql, dbconn)
            'fills the dataset.'
            da.Fill(ds, "tblDentists")
            Dim cb As New SqlCommandBuilder(da)
            Dim dsNewRow As DataRow
            'Creates a new row.'
            dsNewRow = ds.Tables("tblDentists").NewRow()

            'Creates a new row for the user type from the radio buttons.'
            If rdoDentalNurse.Checked = True Then
                dsNewRow.Item("DColour") = "Blue"
            ElseIf rdoStudent.Checked = True Then
                dsNewRow.Item("DColour") = "Green"
            ElseIf rdoTutor.Checked = True Then
                dsNewRow.Item("DColour") = "Red"
            End If
            'Creates a new row for the rest of the dentist details from the
textfields.'
            dsNewRow.Item("DTitle") = cboDTitle.Text
            dsNewRow.Item("DFirstName") = txtDFirstName.Text
            dsNewRow.Item("DSurname") = txtDSurname.Text
            dsNewRow.Item("DDateOfBirth") = DDateOfBirth.Text
            dsNewRow.Item("DGender") = txtDGender.Text
            dsNewRow.Item("DPhoneNumber") = txtDPhoneNumber.Text
            dsNewRow.Item("DHomeTelephone") = txtDHomeTelephone.Text
            dsNewRow.Item("DPassword") = txtNewPassword.Text

            'New row is added.'
            ds.Tables("tblDentists").Rows.Add(dsNewRow)
            'Updates the dataset.'
            da.Update(ds, "tblDentists")
            da = New SqlDataAdapter(Sql, dbconn)
            da.Fill(ds, "tblDentists")

```



```
dbconn.Close()

'This statement will select the registered user.'
Sql = "SELECT TOP 1 * FROM tblDentists ORDER BY tblDentists.DentistID
DESC;"
da = New SqlDataAdapter(Sql, dbconn)
Try
    da.Fill(ds, "tblNewDentist")
Catch ex As Exception
    MessageBox.Show(vbCrLf & ex.Message)
End Try
'A message box will show showing the newly registered user's ID and
Password.'
MessageBox.Show("Your new DentistID is: " &
ds.Tables("tblNewDentist").Rows(0).Item("DentistID") & " Your new Password is: " &
ds.Tables("tblNewDentist").Rows(0).Item("DPassword"))
dbconn.Close()
Connect()

MessageBox.Show("The patient's details have been saved onto the
database.")

'Resets all objects on form to their original states.'
cboDentist.SelectedIndex = -1
btnAdd.Enabled = False
btnDelete.Enabled = False
btnEdit.Enabled = False
btnSave.Enabled = False
cboDentist.Enabled = True
btnShow.Enabled = True
btnRegister.Enabled = True
'Resets combo box.'
cboDTitle.SelectedIndex = -1
'Clears all textfields.'
txtDFirstName.Clear()
txtDSurname.Clear()
txtDGender.Clear()
txtDPhoneNumber.Clear()
txtDHomeTelephone.Clear()
txtNewPassword.Clear()
txtReTypePassword.Clear()
'Makes sure date is set back to the original date.'
DDateOfBirth.Text = "01/01/1998"

'Sets textfields back to original so they can't be edited.'
cboDTitle.Enabled = False
txtDFirstName.ReadOnly = True
txtDSurname.ReadOnly = True
txtDGender.ReadOnly = True
txtDPhoneNumber.ReadOnly = True
txtDHomeTelephone.ReadOnly = True
txtNewPassword.ReadOnly = True
txtReTypePassword.ReadOnly = True
DDateOfBirth.Enabled = False
'Changes the staff image back to the default image for students.'
picStaff.Image = My.Resources.picStudent
'Disabled all radio buttons.'
rdoStudent.Enabled = False
rdoDentalNurse.Enabled = False
rdoTutor.Enabled = False
rdoStudent.Checked = True
'Clears the combo boxes.'
```

```

cboDentist.Items.Clear()
cboID.Items.Clear()

'Statement reloads all dentists onto the combo boxes.'
Sql = "SELECT * FROM tblDentists"
da = New SqlDataAdapter(Sql, dbconn)
Try
    ds.Tables("tblDentistsName").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblDentistsName")
Countrows = ds.Tables("tblDentistsName").Rows.Count
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1

cboID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))

cboDentist.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
Next

End If
End Sub

Public Function ValidateStaff(ByVal cboDTitle As ComboBox, ByVal txtDFirstName As
TextBox, ByVal txtDSurName As TextBox, ByVal txtDGender As TextBox, ByVal
txtDPhoneNumber As TextBox, ByVal txtHomePhone As TextBox, ByVal txtNewPassword As
TextBox, ByVal txtReTypePassword As TextBox) As Boolean
    AllFieldsValid = True
    'Using this line of validation, I can do a visible check on the all the text
boxes I need without having to make separate lines for each visible check.'
    If cboDTitle.Text = "" Or txtDFirstName.Text = "" Or txtDSurName.Text = "" Or
txtDGender.Text = "" Or txtDPhoneNumber.Text = "" Or txtNewPassword.Text = "" Or
txtReTypePassword.Text = "" Then
        MessageBox.Show("Blank box detected. Please fill in all required fields.")
        AllFieldsValid = False
    ElseIf IsNumeric(cboDTitle.Text) Or IsNumeric(txtDFirstName.Text) = True Or
IsNumeric(txtDSurName.Text) Or IsNumeric(txtDGender.Text) = True Then
        MessageBox.Show("First, Surname and Gender must not be numeric.")
        AllFieldsValid = False
        'Declaring the textbox as a date will make sure the date is in the correct
format.'
        'I make sure if the numeric text fields are numeric and if not then a
message box will appear.'
    ElseIf IsNumeric(txtDPhoneNumber.Text) And IsNumeric(txtDHomeTelephone.Text) =
False Then
        MessageBox.Show("Phone numbers and Home number must be numeric.")
        AllFieldsValid = False
        'For these validations I have done a textlength check to make sure that
the fields are valid.'
    ElseIf txtDFirstName.TextLength < 2 Or txtDSurName.TextLength < 2 Then
        MessageBox.Show("First and Surname must be at least 2 characters or
above.")
        AllFieldsValid = False
    ElseIf txtDFirstName.TextLength > 20 Or txtDSurName.TextLength > 20 Then
        MessageBox.Show("First and Surname must be 20 characters less.")
        AllFieldsValid = False
        'Now I am able to validate to make sure that the field can only contain 1
character.'
    ElseIf txtDGender.Text <> "M" And txtDGender.Text <> "F" Then
        MessageBox.Show("Gender must be 1 character. Either 'M' or 'F'.")
        AllFieldsValid = False
    End If
End Function

```

```
'Telephone validated for 11 numbers. If not 11 numbers then error message
will appear.'
ElseIf txtDPhoneNumber.TextLength < 11 Or txtDPhoneNumber.TextLength > 11 Or
txtDHomeTelephone.TextLength < 11 Or txtDHomeTelephone.TextLength > 11 Then
    MessageBox.Show("Telephone numbers has to be 11 numbers long.")
    AllFieldsValid = False
ElseIf txtNewPassword.Text <> txtReTypePassword.Text Then
    MessageBox.Show("Passwords need to be the same in order to validate.")
    AllFieldsValid = False
    Me.Refresh()
End If
If AllFieldsValid = True Then

    'If this boolean value is true then It will return the function as true
    which can be called in the login button.'
    ValidateStaff = True
Else
    'The value is returned as false therefore the user cannot proceed to the
    next form.'
    ValidateStaff = False

End If

End Function

Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHelp.Click
    MessageBox.Show("All details are required to be entered in the textfields. You
can also load existing users by using the list in order to edit any details required.
Select a profile for each member so they can be registered as a student, tutor or
staff.")
End Sub

Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEdit.Click
    'Makes all text fields enabled so details can be edited.'
    cboDTitle.Enabled = True
    txtDFirstName.ReadOnly = False
    txtDSurname.ReadOnly = False
    txtDGender.ReadOnly = False
    txtDPhoneNumber.ReadOnly = False
    txtDHomeTelephone.ReadOnly = False
    txtNewPassword.ReadOnly = False
    txtReTypePassword.ReadOnly = False
    DDateOfBirth.Enabled = True
    'Disables delete button.'
    btnDelete.Enabled = False
    btnSave.Enabled = True
    'Disables the edit button so it cannot be pressed again unless cancelled.'
    btnEdit.Enabled = False
    'Makes the radio buttons checkable again.'
    rdoStudent.Enabled = True
    rdoDentalNurse.Enabled = True
    rdoTutor.Enabled = True

End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    'I dim 'response' as a message result in order to bring up an input box for
    the user to decide if they are certain to proceed.'
```

```

Dim response As MsgBoxResult
'Selected Dentist will be the dentist ID which will be an integer.'
Dim SelectedDentist As Integer
'Dims the comb box for ID as an integer.'
SelectedDentist = cboID.Text
'This is the repsonse used to decide if the user wants to proceed.'
response = MsgBox("Are you sure you want to update?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Confirm")
'If the response is yes then validation is called.'
If response = MsgBoxResult.Yes Then
    If ValidateStaff(cboDTitle, txtDFirstName, txtDSurname, txtDGender,
txtDPhoneNumber, txtDHomeTelephone, txtNewPassword, txtReTypePassword) = True Then
        'This statement selects where the data will be saved at from the
dentist ID.'
        Sql = "SELECT * FROM tblDentists WHERE DentistID = " & SelectedDentist
        da = New SqlDataAdapter(Sql, dbconn)
        Dim cb As New SqlCommandBuilder(da)
        da.Fill(ds, "tblUpdateDentist")

        'All the rows will equal what is entered from the text fields.'
        ds.Tables("tblUpdateDentist").Rows(0).Item("DTitle") = cboDTitle.Text
        ds.Tables("tblUpdateDentist").Rows(0).Item("DFirstName") =
txtDFirstName.Text
        ds.Tables("tblUpdateDentist").Rows(0).Item("DSurname") =
txtDSurname.Text
        ds.Tables("tblUpdateDentist").Rows(0).Item("DDateOfBirth") =
DDateOfBirth.Text
        ds.Tables("tblUpdateDentist").Rows(0).Item("DGender") =
txtDGender.Text
        ds.Tables("tblUpdateDentist").Rows(0).Item("DPhoneNumber") =
txtDPhoneNumber.Text
        ds.Tables("tblUpdateDentist").Rows(0).Item("DHomeTelephone") =
txtDHomeTelephone.Text
        ds.Tables("tblUpdateDentist").Rows(0).Item("DPassword") =
txtNewPassword.Text

        'Make sure the radio buttons will set the colours for the staff.'
        If rdoDentalNurse.Checked = True Then
            lblColour.Text = "Blue"
        ElseIf rdoTutor.Checked = True Then
            lblColour.Text = "Red"
        ElseIf rdoStudent.Checked = True Then
            lblColour.Text = "Green"
        End If
        'The radio buttons will change the label text which is then saved onto
the colour row on the dentist table.'
        ds.Tables("tblUpdateDentist").Rows(0).Item("DColour") = lblColour.Text

        'Updates the dataset.'
        da.Update(ds, "tblUpdateDentist")
        'Message box shows that the details have been saved.'
        MessageBox.Show("Details have been saved and updated")
    Else
        'If any rows are invalid a message box will show.'
        MessageBox.Show("Invalid Row, no changes made.")
    End If
End If

'This is the same code from the cancel button.'
'Resets combo boxes.'
cboDentist.SelectedIndex = -1

```

```
'Sets buttons to default state when the form loads.'  
btnAdd.Enabled = False  
btnDelete.Enabled = False  
btnEdit.Enabled = False  
btnSave.Enabled = False  
cboDentist.Enabled = True  
btnShow.Enabled = True  
btnRegister.Enabled = True  
cboDTitle.SelectedIndex = -1  
  
'Clears all textfields.'  
txtDFirstName.Clear()  
txtDSurname.Clear()  
txtDGender.Clear()  
txtDPhoneNumber.Clear()  
txtDHomeTelephone.Clear()  
txtNewPassword.Clear()  
txtReTypePassword.Clear()  
DDateOfBirth.Text = "01/01/1998"  
  
'Disables combo box.'  
cboDTitle.Enabled = False  
'Makes text fields read only to prevent editing.'  
txtDFirstName.ReadOnly = True  
txtDSurname.ReadOnly = True  
txtDGender.ReadOnly = True  
txtDPhoneNumber.ReadOnly = True  
txtDHomeTelephone.ReadOnly = True  
txtNewPassword.ReadOnly = True  
txtReTypePassword.ReadOnly = True  
DDateOfBirth.Enabled = False  
'Staff image is reset to student.'  
picStaff.Image = My.Resources.picStudent  
rdoStudent.Enabled = False  
rdoDentalNurse.Enabled = False  
rdoTutor.Enabled = False  
rdoStudent.Checked = True  
'Combo boxes are cleared.'  
cboDentist.Items.Clear()  
cboID.Items.Clear()  
  
'Selects all from Dentist table.'  
Sql = "SELECT * FROM tblDentists"  
da = New SqlDataAdapter(Sql, dbconn)  
'Clears all the datasets used on the form.'  
Try  
    ds.Tables("tblDentistsName").Clear()  
Catch ex As Exception  
  
End Try  
  
Try  
    ds.Tables("tblUpdateDentist").Clear()  
Catch ex As Exception  
  
End Try  
da.Fill(ds, "tblDentistsName")  
Countrows = ds.Tables("tblDentistsName").Rows.Count  
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1  
    'Refills comboboxes with data from table.'  
    cboID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))
```

```
cboDentist.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
    Next

End Sub

Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
    'This resets the combo box containing the dentist's name.'
    cboDentist.SelectedIndex = -1
    'All buttons are reset to what they were when the form loaded.'
    btnAdd.Enabled = False
    btnDelete.Enabled = False
    btnEdit.Enabled = False
    btnSave.Enabled = False
    cboDentist.Enabled = True
    btnShow.Enabled = True
    btnRegister.Enabled = True
    cboDTitle.SelectedIndex = -1
    'All text fields are cleared.'
    txtDFirstName.Clear()
    txtDSurname.Clear()
    txtDGender.Clear()
    txtDPhoneNumber.Clear()
    txtDHomeTelephone.Clear()
    txtNewPassword.Clear()
    txtReTypePassword.Clear()
    'Date of birth value is reset to default.'
    DDateOfBirth.Text = "01/01/1998"
    'Disabled combo boxes.'
    cboDTitle.Enabled = False
    'Sets textfields to read only to prevent editing.'
    txtDFirstName.ReadOnly = True
    txtDSurname.ReadOnly = True
    txtDGender.ReadOnly = True
    txtDPhoneNumber.ReadOnly = True
    txtDHomeTelephone.ReadOnly = True
    txtNewPassword.ReadOnly = True
    txtReTypePassword.ReadOnly = True
    'Date of birth field is disabled.'
    DDateOfBirth.Enabled = False
    'Staff image is reset to student.'
    picStaff.Image = My.Resources.picStudent
    'Radio buttons are disabled.'
    rdoStudent.Enabled = False
    rdoDentalNurse.Enabled = False
    rdoTutor.Enabled = False
    rdoStudent.Checked = True

    'Combo boxes are cleared.'
    cboDentist.Items.Clear()
    cboID.Items.Clear()

    'Sequel statement retrieves all rows from dentist table.'
    Sql = "SELECT * FROM tblDentists"
    da = New SqlDataAdapter(Sql, dbconn)
    Try
        ds.Tables("tblDentistsName").Clear()
    Catch ex As Exception

End Try
da.Fill(ds, "tblDentistsName")
```

```
Countrows = ds.Tables("tblDentistsName").Rows.Count
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1
    'ID and Firstnames of all dentists are put into the combo boxes.'
    cboID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))
cboDentist.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
Next
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
    'Makes selected dentist ID as an integer.'
    Dim SelectedDentist As Integer
    'Sets combo box as integer.'
    SelectedDentist = cboID.Text

    'Clears dataset.'
    Try
        ds.Tables("tblDentist").Clear()
    Catch ex As Exception

    End Try
    'Selects the dentist ID from the combo box.'
    Sql = "DELETE FROM tblDentists WHERE DentistID = " & SelectedDentist & ""

    da = New SqlDataAdapter(Sql, dbconn)
    'Deletes the whole row from the selected ID.'
    da.Fill(ds, "tblDentists")

    'Message box shows telling the user that the record is delete.d.'
    MessageBox.Show("Record deleted")

    'This resets the combo box containing the dentist's name.'
    cboDentist.SelectedIndex = -1
    'All buttons are reset to what they were when the form loaded.'
    btnAdd.Enabled = False
    btnDelete.Enabled = False
    btnEdit.Enabled = False
    btnSave.Enabled = False
    cboDentist.Enabled = True
    btnShow.Enabled = True
    btnRegister.Enabled = True
    cboDTitle.SelectedIndex = -1
    'All text fields are cleared.'
    txtDFirstName.Clear()
    txtDSurname.Clear()
    txtDGender.Clear()
    txtDPhoneNumber.Clear()
    txtDHomeTelephone.Clear()
    txtNewPassword.Clear()
    txtReTypePassword.Clear()
    'Date of birth value is reset to default.'
    DDateOfBirth.Text = "01/01/1998"
    'Disabled combo boxes.'
    cboDTitle.Enabled = False
    'Sets textfields to read only to prevent editing.'
    txtDFirstName.ReadOnly = True
    txtDSurname.ReadOnly = True
    txtDGender.ReadOnly = True
    txtDPhoneNumber.ReadOnly = True
    txtDHomeTelephone.ReadOnly = True
    txtNewPassword.ReadOnly = True
```

```
txtReTypePassword.ReadOnly = True
'Date of birth field is disabled.'
DDateOfBirth.Enabled = False
'Staff image is reset to student.'
picStaff.Image = My.Resources.picStudent
'Radio buttons are disabled.'
rdoStudent.Enabled = False
rdoDentalNurse.Enabled = False
rdoTutor.Enabled = False
rdoStudent.Checked = True

'Combo boxes are cleared.'
cboDentist.Items.Clear()
cboID.Items.Clear()

'Sequel statement retrieves all rows from dentist table.'
Sql = "SELECT * FROM tblDentists"
da = New SqlDataAdapter(Sql, dbconn)
Try
    ds.Tables("tblDentistsName").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblDentistsName")
Countrows = ds.Tables("tblDentistsName").Rows.Count
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1
    'ID and Firstnames of all dentists are put into the combo boxes.'
    cboID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))
cboDentist.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
Next

End Sub

Private Sub btnRegister_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRegister.Click
    'Makes sure that combo boxes are reset and disabled.'
    cboDentist.SelectedIndex = -1
    cboDentist.Enabled = False
    'Makes sure that the user can't save it the user as a existing user by
disabling buttons.'
    btnEdit.Enabled = False
    btnSave.Enabled = False
    btnShow.Enabled = False
    'Enabled add button so details can be added onto database.'
    btnAdd.Enabled = True
    'Disables delete button.'
    btnDelete.Enabled = False
    DDateOfBirth.Enabled = True
    'Disables register button to prevent resets.'
    btnRegister.Enabled = False
    'Enables radio buttons and textfields.'
    rdoStudent.Enabled = True
    rdoDentalNurse.Enabled = True
    rdoTutor.Enabled = True
    cboDTitle.Enabled = True
    txtDFirstName.ReadOnly = False
    txtDSurname.ReadOnly = False
    txtDGender.ReadOnly = False
    txtDPhoneNumber.ReadOnly = False
    txtDHomeTelephone.ReadOnly = False
    txtNewPassword.ReadOnly = False
```



```

txtReTypePassword.ReadOnly = False
rdoStudent.Checked = True

End Sub

Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShow.Click
    'Sets the validation as true.'
    AllFieldsValid = True
    'Sets the selected dentist as an integer.'
    Dim SelectedDentist As Integer
    'Validation to make sure that there is data in the combo box.'
    If cboDentist.Text = "" Then
        MessageBox.Show("Please select a member")
        'Makes the fields invalid so the dentist's data doesn't load.'
        AllFieldsValid = False
    End If

    If AllFieldsValid = True Then
        'Edit button is enabled.'
        btnEdit.Enabled = True
        'Makes sure the user cannot create two same users by disabling the add
button.'
        btnAdd.Enabled = False
        btnRegister.Enabled = False
        'Prevents data from being retrieved again.'
        btnShow.Enabled = False
        'Makes sure the user cannot load another user.'
        cboDentist.Enabled = False
        'Deleting the record is available.'
        btnDelete.Enabled = True

        'Sets the combo box as an integer.'
        SelectedDentist = cboID.Text
        Try
            'Clears the dataset.'
            ds.Tables("tblSelectedDentist").Clear()
        Catch ex As Exception

        End Try

        'Selects the selected dentist from combo box.'
        Sql = "SELECT * FROM tblDentists WHERE DentistID = " & SelectedDentist &
""

        da = New SqlDataAdapter(Sql, dbconn)
        da.Fill(ds, "tblSelectedDentist")

        'The dentists data will load into the text fields, labels and combo
boxes.'
        cboDTitle.Text = ds.Tables("tblSelectedDentist").Rows(0).Item("DTitle")
        txtDFirstName.Text =
ds.Tables("tblSelectedDentist").Rows(0).Item("DFirstName")
        txtDSurname.Text =
ds.Tables("tblSelectedDentist").Rows(0).Item("DSurname")
        DDateOfBirth.Text =
ds.Tables("tblSelectedDentist").Rows(0).Item("DDateOfBirth")
        txtDGender.Text = ds.Tables("tblSelectedDentist").Rows(0).Item("DGender")
        txtDPhoneNumber.Text =
ds.Tables("tblSelectedDentist").Rows(0).Item("DPhoneNumber")
        txtDHomeTelephone.Text =
ds.Tables("tblSelectedDentist").Rows(0).Item("DHomeTelephone")
    
```

```
        txtNewPassword.Text =  
ds.Tables("tblSelectedDentist").Rows(0).Item("DPassword")  
        txtReTypePassword.Text =  
ds.Tables("tblSelectedDentist").Rows(0).Item("DPassword")  
        lblColour.Text = ds.Tables("tblSelectedDentist").Rows(0).Item("DColour")  
  
        'The dentist colour will load onto the label text.'  
        Select Case lblColour.Text  
            Case "Green"  
  
                rdoStudent.Checked = True  
            Case "Blue"  
  
                rdoDentalNurse.Checked = True  
            Case "Red"  
  
                rdoTutor.Checked = True  
  
        End Select  
  
    End If  
  
End Sub  
  
Private Sub cboDentist_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cboDentist.SelectedIndexChanged  
    'Set the combo box for dentists name to match their ID's.'  
    SelecteddentistsName = cboDentist.SelectedIndex  
    cboID.SelectedIndex = cboDentist.SelectedIndex  
  
End Sub  
  
Private Sub txtDFirstName_KeyPress(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtDFirstName.KeyPress  
    'Ignore everything but letter keys.'  
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then  
e.Handled = True  
    End Sub  
  
Private Sub txtDSurname_KeyPress(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtDSurname.KeyPress  
    'Ignore everything but letter keys.'  
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then  
e.Handled = True  
    End Sub  
  
Private Sub txtDGender_KeyPress(ByVal sender As Object, ByVal e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtDGender.KeyPress  
    'Ignore everything but letter keys.'  
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then  
e.Handled = True  
    End Sub  
  
Private Sub txtDPhoneNumber_KeyPress(sender As Object, e As  
System.Windows.Forms.KeyPressEventArgs) Handles txtDPhoneNumber.KeyPress  
    'Ignore everything but numeric keys.'  
    If Not Char.IsDigit(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then  
e.Handled = True  
    End Sub
```

Tony Van – 30078829

```
Private Sub txtDHomeTelephone_KeyPress(sender As Object, e As
System.Windows.Forms.KeyPressEventArgs) Handles txtDHomeTelephone.KeyPress
    'Ignore everything but numeric keys.'
    If Not Char.IsDigit(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
        e.Handled = True
    End Sub
End Class
```

frmMenu

The image will change depending on the status of the logged user.

The logged in users details will load here.

These buttons are used to navigate to the Patient List form and the Appointment List form.

This help button will bring up a message providing further details and help in using the current form.

This button will refresh the dataset and load an updated appointment ID list.

Opening the patient ID will load the Patient ID and Treatment ID of the selected booking.

This is a list of all the appointments the logged user has.

This button will close the current form and return back to the login form. While clearing the dataset and details.

The patient's details and booking details will be displayed on the labels and text fields.

This button resets the form, clearing all text fields and refreshing the ID list.

This button completes the appointment and clears it off the booking table.

The screenshot shows a form titled 'frmMenu'. At the top left is a green square logo with the word 'STUDENT' in white. To its right are input fields for 'ID:' (containing '3'), 'Title:' (containing 'Miss'), 'First Name:' (containing 'Rosie'), and 'Surname:' (containing 'Van'). Below these are two buttons: 'Patient List' and 'Appointment List'. To the right of these buttons is a help button with a question mark icon. Below the logo is a 'Refresh List' button and an 'Appointment ID:' dropdown menu (showing '1'). Below that is an 'Open Patient ID' button. To the left of the 'Open Patient ID' button is a list of appointments: '24/06/2016 : 12:30', '13/05/2016 : 14:00', '23/09/2016 : 12:00', and '25/11/2016 : 15:30'. Below this list is a 'Log Off' button. To the right of the 'Open Patient ID' button are input fields for 'Patient ID:' (a dropdown showing '7'), 'Title:' (containing 'Mr'), 'First Name:' (containing 'Bob'), 'Surname:' (containing 'Ros'), 'Date Of Birth:' (containing '23/06/1997'), 'Phone Number:' (containing '12351235135'), and 'Home Telephone:' (containing '02349548582'). Below these are 'Time: 12:30', 'Duration: 60 Minutes', and 'Date: 24/06/2016'. To the right of these fields is a 'Display' button. Below the 'Display' button is a 'Treatments:' label followed by the number '2' and the word 'Polishing'. Below this is a 'Reset' button and a 'Complete' button.

```
Imports System.Data.SqlClient
Public Class frmMenu

    Private Sub frmMenu_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'This variable is used to count the rows in a dataset.'
        Dim CountRows As Integer
        'This variable is used loop until I can find the row I'm looking for.'
        Dim i As Integer
        'This makes sure that the label is invisible. This is used as an indicator for
the image to change to a specific image.'
        Colour.Visible = False
        'Makes sure that the button reset cannot be pressed when the form loads.'
        btnReset.Enabled = False
        'Makes sure that the record cannot be deleted as there would be no information
to delete.'
        btnComplete.Enabled = False
        'Makes sure that the ID is equal the integer in the module, so when the user
logs in, it will be displayed in the textfield.'
        txtID.Text = DentistID
        'This section loads all the data from the selected dentist into the
textfields.'
        txtDTitle.Text = ds.Tables("tblDentists").Rows(0).Item("DTitle")
        txtDFirstName.Text = ds.Tables("tblDentists").Rows(0).Item("DFirstName")
        txtDSurname.Text = ds.Tables("tblDentists").Rows(0).Item("DSurname")
        Colour.Text = ds.Tables("tblDentists").Rows(0).Item("DColour")

        'After label has changed text from the dataset, the text will change what the
picture represents in the case statement.'
        Select Case Colour.Text.ToString
            Case "Green"
                picStaff.Image = My.Resources.picStudent
            Case "Blue"
                picStaff.Image = My.Resources.picDentalNurse
            Case "Red"
                picStaff.Image = My.Resources.picTutor
        End Select

        'This Selects the row from the booking table where the dentist ID matches the
dentist that logs in and loads all their appointments onto a list box and combo box.'
        Sql = "SELECT * FROM tblBooking Where DentistID = " & DentistID
        da = New SqlDataAdapter(Sql, dbconn)
        da.Fill(ds, "tblAppointmentListings")
        CountRows = ds.Tables("tblAppointmentListings").Rows.Count
        For i = 0 To CountRows - 1

            lstPatientsToday.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("Appointme
ntDate") & "      :      " &
(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentTime")))

            cboAppointmentID.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("Appointme
ntID"))
        Next
    End Sub

    Private Sub btnBack_Click(sender As System.Object, e As System.EventArgs) Handles
btnLogOff.Click
        'This will close the form making sure that any details are cleared.'
        Me.Close()
    End Sub
End Class
```

```
'The login form will show again.'
frmLogin.Show()
'Clears the textfields in the login form so that the next user who logs on
will not be able to log into another users account.'
frmLogin.txtUserID.Clear()
frmLogin.txtPassword.Clear()
frmLogin.txtAdminPassword.Clear()
'Makes sure that the dentist details are cleared from the menu form.'
txtID.Clear()
txtDTitle.Clear()
txtDFirstName.Clear()
txtDSurname.Clear()
'Makes sure the dataset is cleared so that a new user can load their details.'
Try
    If ds.Tables("tblDentists").Rows.Count > 0 Then
        ds.Tables("tblDentists").Clear()
    End If
Catch ex As Exception
    'Ignore error if dataset doesn't exist.'
End Try

End Sub

Private Sub btnPatientList_Click(sender As System.Object, e As System.EventArgs)
Handles btnPatientList.Click
    'Hides the current form.'
    Me.Hide()
    'Shows the patient list form.'
    frmPatientList.Show()
End Sub

Private Sub btnAppointmentList_Click(sender As System.Object, e As
System.EventArgs) Handles btnAppointmentList.Click
    'Hides the current form.'
    Me.Hide()
    'Shows the appointment list form.'
    frmAppointmentList.Show()
End Sub

Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHelp.Click
    'Help box to show how to use the current form.'
    MessageBox.Show("Your appointments will have loaded onto the list boxes.
Please select an individual appointment in order to view details and confirm
completion. You can also access the Patient list and Appointment list form.")
End Sub

Private Sub picStaff_Click(sender As System.Object, e As System.EventArgs) Handles
picStaff.Click

End Sub

Private Sub btnRefresh_Click(sender As System.Object, e As System.EventArgs)
Handles btnRefresh.Click
    'This variable is used to count the rows in a dataset.'
    Dim CountRows As Integer
    'This variable is used loop until I can find the row I'm looking for.'
    Dim i As Integer

    'Clears the list box and appointment combobox.'
    lstPatientsToday.Items.Clear()
```

```

cboAppointmentID.Items.Clear()
'Makes sure the combo box is blank.'
cboAppointmentID.SelectedIndex = -1

Try
'Makes sure that if there is any data in the dataset. If there is, it will
be cleared.'
If ds.Tables("tblAppointmentListings").Rows.Count > 0 Then
    ds.Tables("tblAppointmentListings").Clear()
End If
Catch ex As Exception
'Ignore error if dataset doesn't exist.'
End Try
'Sql statement selecting the dentist ID from the booking table which will load
the dentists appointments onto the combo box and list box.'
Sql = "SELECT * FROM tblBooking Where DentistID = " & DentistID
da = New SqlDataAdapter(Sql, dbconn)
da.Fill(ds, "tblAppointmentListings")
CountRows = ds.Tables("tblAppointmentListings").Rows.Count
For i = 0 To CountRows - 1

lstPatientsToday.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentDate") & " : " &
(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentTime")))

cboAppointmentID.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentID"))

Next
End Sub

Private Sub cboAppointmentID_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles cboAppointmentID.SelectedIndexChanged
'IIf a ID is selected from the combo box. the refresh button will be disabled
to prevent refreshing the dataset while it is in use.'
btnRefresh.Enabled = False
End Sub

Private Sub btnDisplay_Click(sender As System.Object, e As System.EventArgs) Handles btnDisplay.Click
'This makes sure that the ID for the appointment and patients are an integer.'
Dim SelectedAppointmentID As Integer
Dim SelectedPatientID As Integer
'Enables the button to display the patients details.'
btnDisplayPatient.Enabled = True
'Makes sure that the combo box values are an integer.'
SelectedPatientID = cboPatientID.SelectedItem
SelectedAppointmentID = cboAppointmentID.SelectedItem
'Disables the display button so you can't retrieve the same details again.'
btnDisplay.Enabled = False
'Validation check to make sure that there is data in the combo boxes.'
AllFieldsValid = True
If cboAppointmentID.Text = "" Then
    MessageBox.Show("Please select a member")
    AllFieldsValid = False
End If

If AllFieldsValid = True Then
'Selects the appointment ID from the booking table and loads data such as
date and time from the booking to the text fields.'
Sql = "SELECT * FROM tblBooking WHERE AppointmentID = " &
SelectedAppointmentID & ""
da = New SqlDataAdapter(Sql, dbconn)

```

```

        da.Fill(ds, "tblSelectedPatientID")

        cboPatientID.Text =
ds.Tables("tblSelectedPatientID").Rows(0).Item("PatientID")
        lblTime.Text =
ds.Tables("tblSelectedPatientID").Rows(0).Item("AppointmentTime")
        lblDate.Text =
ds.Tables("tblSelectedPatientID").Rows(0).Item("AppointmentDate")
        lblDuration.Text =
ds.Tables("tblSelectedPatientID").Rows(0).Item("TreatmentDuration")
        lblTreatmentID.Text =
ds.Tables("tblSelectedPatientID").Rows(0).Item("TreatmentID")

        'Makes sure that the user can't redisplay the same details.'
        btnDisplay.Enabled = False
    End If
End Sub

Private Sub btnDisplayPatient_Click(sender As System.Object, e As
System.EventArgs) Handles btnDisplayPatient.Click
    'This makes sure that the ID for the appointment, patients and treatments.'
    Dim SelectedAppointmentID As Integer
    Dim SelectedPatientID As Integer
    Dim SelectedTreatment As Integer
    'Disables the display patient button.'
    btnDisplayPatient.Enabled = False
    'Makes the comboboxes and label fields for the ID an integer.'
    SelectedPatientID = cboPatientID.Text
    SelectedAppointmentID = cboAppointmentID.SelectedItem
    SelectedTreatment = lblTreatmentID.Text
    'Makes sure the buttons for resetting the fields and deleting the appointments
are enabled.'
    btnReset.Enabled = True
    btnComplete.Enabled = True

    'This sequel statement will select the patients ID from the patients table and
display all their details on the text fields.'
    Sql = "SELECT * FROM tblPatients WHERE PatientID = " & SelectedPatientID
    da = New SqlDataAdapter(Sql, dbconn)
    da.Fill(ds, "tblPatientSelected")

    txtTitle.Text = ds.Tables("tblPatientSelected").Rows(0).Item("Title")
    txtFirstName.Text = ds.Tables("tblPatientSelected").Rows(0).Item("FirstName")
    txtSurname.Text = ds.Tables("tblPatientSelected").Rows(0).Item("Surname")
    txtDateOfBirth.Text =
ds.Tables("tblPatientSelected").Rows(0).Item("DateOfBirth")
    txtPhoneNumber.Text =
ds.Tables("tblPatientSelected").Rows(0).Item("PhoneNumber")
    txtHomeTelephone.Text =
ds.Tables("tblPatientSelected").Rows(0).Item("HomeNumber")

    'This statement will select all the treatments from the ID and displays the
treatment type from that ID.'
    Sql = "SELECT * FROM tblTreatment WHERE TreatmentID = " & SelectedTreatment
    da = New SqlDataAdapter(Sql, dbconn)
    da.Fill(ds, "tblTreatmentSelected")
    lblTreatment.Text =
ds.Tables("tblTreatmentSelected").Rows(0).Item("TreatmentType")

End Sub

```



```
Private Sub btnComplete_Click(sender As System.Object, e As System.EventArgs)
Handles btnComplete.Click
    'Makes the Selected appointment ID and integer.'
    Dim SelectedAppointment As Integer
    SelectedAppointment = cboAppointmentID.Text

    'Clears the dataset.'
    Try
        ds.Tables("tblBooking").Clear()
    Catch ex As Exception

    End Try

    'Statement to delete the row from the booking table where the ID is equal to
the selected appointment.'
    Sql = "DELETE FROM tblBooking WHERE AppointmentID = " & SelectedAppointment &
""

    da = New SqlDataAdapter(Sql, dbconn)

    da.Fill(ds, "tblBooking")

    MessageBox.Show("Record deleted")

    'Same code as reset.'
    Dim CountRows As Integer
    Dim i As Integer

    'Clearing all text fields and combo fields resetting.'
    cboAppointmentID.SelectedIndex = -1
    cboPatientID.SelectedIndex = -1
    cboAppointmentID.Items.Clear()
    cboPatientID.Items.Clear()
    cboPatientID.Text = ""

    txtTitle.Clear()
    txtFirstName.Clear()
    txtSurname.Clear()
    txtPhoneNumber.Clear()
    txtHomeTelephone.Clear()
    txtDateOfBirth.Clear()
    'Makes sure labels are reset to what they were originally.'
    lblTreatment.Text = "Treatment"
    lblTreatmentID.Text = "Treatment ID"
    lblTime.Text = "00:00"
    lblDate.Text = "DD/MM/YYYY"
    lblDuration.Text = "00"

    'Clears the list box.'
    lstPatientsToday.Items.Clear()

    'Clears the appointment listings.'
    If ds.Tables("tblAppointmentListings").Rows.Count > 0 Then
        Try
            ds.Tables("tblAppointmentListings").Clear()
        Catch ex As Exception

        End Try
    End If

    'Clears the tables for the selected patient.'
    If ds.Tables("tblSelectedPatientID").Rows.Count > 0 Then
        Try
```

```

        ds.Tables("tblSelectedPatientID").Clear()
    Catch ex As Exception

    End Try
End If
'Clears the dataset for the patients details.'
If ds.Tables("tblPatientSelected").Rows.Count > 0 Then
    Try
        ds.Tables("tblPatientSelected").Clear()
    Catch ex As Exception

    End Try
End If
'Clears the dataset for the selected treatment.'
If ds.Tables("tblTreatmentSelected").Rows.Count > 0 Then
    Try
        ds.Tables("tblTreatmentSelected").Clear()
    Catch ex As Exception

    End Try
End If
'Statement to reload the details back onto the combo box and list box.'
Sql = "SELECT * FROM tblBooking Where DentistID = " & DentistID
da = New SqlDataAdapter(Sql, dbconn)
da.Fill(ds, "tblAppointmentListings")
CountRows = ds.Tables("tblAppointmentListings").Rows.Count
For i = 0 To CountRows - 1

1stPatientsToday.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentDate") & " : " & (ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentTime")))

cboAppointmentID.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentID"))
Next

'Makes sure that the buttons are reset to their original state.'
btnReset.Enabled = False
btnComplete.Enabled = False
btnDisplay.Enabled = True
cboAppointmentID.SelectedIndex = -1
cboPatientID.SelectedIndex = -1
btnRefresh.Enabled = True
End Sub

Private Sub btnReset_Click(sender As System.Object, e As System.EventArgs) Handles btnReset.Click
    'This variable is used to count the rows in a dataset.'
    Dim CountRows As Integer
    'This variable is used loop until I can find the row I'm looking for.'
    Dim i As Integer

    'Makes sure the combo boxes are blank.'
    cboAppointmentID.SelectedIndex = -1
    cboPatientID.SelectedIndex = -1
    cboAppointmentID.Items.Clear()
    cboPatientID.Items.Clear()
    cboPatientID.Text = ""

    'Clears all text fields for the patients details.'
    txtTitle.Clear()
    txtFirstName.Clear()

```

```
txtSurname.Clear()
txtPhoneNumber.Clear()
txtHomeTelephone.Clear()
txtDateOfBirth.Clear()
'Resets the labels to their original state.'
lblTreatment.Text = "Treatment"
lblTreatmentID.Text = "Treatment ID"
lblTime.Text = "00:00"
lblDate.Text = "DD/MM/YYYY"
lblDuration.Text = "00"

'Clears the list box.'
lstPatientsToday.Items.Clear()

'Clearing all datasets used on the form'
If ds.Tables("tblAppointmentListings").Rows.Count > 0 Then
    Try
        ds.Tables("tblAppointmentListings").Clear()
    Catch ex As Exception

    End Try
End If

If ds.Tables("tblSelectedPatientID").Rows.Count > 0 Then
    Try
        ds.Tables("tblSelectedPatientID").Clear()
    Catch ex As Exception

    End Try
End If

If ds.Tables("tblPatientSelected").Rows.Count > 0 Then
    Try
        ds.Tables("tblPatientSelected").Clear()
    Catch ex As Exception

    End Try
End If

If ds.Tables("tblTreatmentSelected").Rows.Count > 0 Then
    Try
        ds.Tables("tblTreatmentSelected").Clear()
    Catch ex As Exception

    End Try
End If

'Statement to reload the details back into the combo box and list box.'
Sql = "SELECT * FROM tblBooking Where DentistID = " & DentistID
da = New SqlDataAdapter(Sql, dbconn)
da.Fill(ds, "tblAppointmentListings")
CountRows = ds.Tables("tblAppointmentListings").Rows.Count
For i = 0 To CountRows - 1

    lstPatientsToday.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentDate") & " : " & (ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentTime")))

    cboAppointmentID.Items.Add(ds.Tables("tblAppointmentListings").Rows(i).Item("AppointmentID"))
Next
```

```
        'Setting the buttons to their original state.'  
        btnReset.Enabled = False  
        btnComplete.Enabled = False  
        btnDisplay.Enabled = True  
        cboAppointmentID.SelectedIndex = -1  
        cboPatientID.SelectedIndex = -1  
        btnRefresh.Enabled = True  
    End Sub  
End Class
```

frmPatientList

These combo boxes will contain the patient ID and first names of all patients in the patient table.

The show button will display the selected patient's details from the combo box to the text fields.

The save button will validate the text fields and save the data back into the database if validated correctly.

The edit button will enable the text fields once the details are shown.

The cancel button will refresh the form.

This help button will bring up a message providing further details and help in using the current form.

The delete button will delete the existing patient off the database.

This button will close the current form and return back to menu form.

This button will disable the functions to edit, save and delete an existing patient.

The add button will call in validation to check each text field to see if the data is correctly inputted. This will then add a new row into the dataset.

The screenshot shows a form titled 'frmPatientList' with the following fields and controls:

- ID:** A dropdown menu showing '1'.
- First Name:** A text field containing 'Thomas'.
- Title:** A dropdown menu showing 'Mr'.
- Surname:** A text field containing 'Cook'.
- Date Of Birth:** A text field containing '04 December 1976' with a calendar icon.
- Gender M / F:** A text field containing 'M'.
- Phone Number:** A text field containing '07412847188'.
- Home Number:** A text field containing '02346748542'.
- House Number:** A text field containing '23'.
- Street:** A text field containing 'Barnes Rd'.
- City:** A text field containing 'Portsmouth'.
- County:** A text field containing 'Hampshire'.
- Postcode:** A text field containing 'PO4 K4B'.
- Allergies:** A checkbox labeled 'Allergies:' with 'Yes' checked and 'No' unchecked.
- Allergy Information:** A text area containing 'Apples'.
- Buttons:** 'Show', 'Save', 'Cancel', 'Edit', 'Back', 'Register New Member', 'Add', and 'Delete'.
- Help Button:** A button with a question mark icon.

The form also features the University of Portsmouth logo at the bottom left.

```
Imports System.Data.SqlClient
Public Class frmPatientList
    'Sets this variable for dentist's name as a string.'
    Dim SelectedPatientName As String
    'Countrows is set as integer for counting the rows in a dataset.'
    Dim Countrows As Integer
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Clears all fields
        cboTitle.SelectedIndex = -1
        txtFirstName.Clear()
        txtSurname.Clear()
        txtGender.Clear()
        txtPhoneNumber.Clear()
        txtHomeNumber.Clear()
        txtHouseNumber.Clear()
        txtStreet.Clear()
        txtCity.Clear()
        txtCounty.Clear()
        txtPostCode.Clear()
        txtAllergyInfo.Clear()
        'Makes sure check boxes are not checked.'
        chbYes.Checked = False
        chbNo.Checked = False
        'Closes the form.'
        Me.Close()
        'Returns to the menu form.'
        frmMenu.Show()
    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
        'Validation for all text fields are called.'
        'If validation is passed on then the sql will be called.'
        If ValidatePatient(txtFirstName, txtSurname, txtGender, txtPhoneNumber,
txtHomeNumber, txtHouseNumber, txtStreet, txtCity, txtCounty, txtPostCode) = True Then
            Dim Sql As String = ""
            Try
                'Clears dataset.'
                ds.Clear()
            Catch ex As Exception
            End Try
            'Selects all from the patients table.'
            Sql = "Select * from tblPatients"
            da = New SqlDataAdapter(Sql, dbconn)
            da.Fill(ds, "tblPatients")
            Dim cb As New SqlCommandBuilder(da)
            Dim dsNewRow As DataRow
            'Creates a new row for the dataset.'
            dsNewRow = ds.Tables("tblPatients").NewRow()

            'Sets the new row according to the check boxes.'
            If chbYes.Checked = True Then
                dsNewRow.Item("Allergies") = "Y"
            ElseIf chbNo.Checked = True Then
                dsNewRow.Item("Allergies") = "N"
            End If
            'Creates new row for the registered patients details.'
            dsNewRow.Item("Title") = cboTitle.Text
            dsNewRow.Item("FirstName") = txtFirstName.Text
            dsNewRow.Item("Surname") = txtSurname.Text
            dsNewRow.Item("DateOfBirth") = DateOfBirth.Text
        End If
    End Sub
```

```
dsNewRow.Item("Gender") = txtGender.Text
dsNewRow.Item("PhoneNumber") = txtPhoneNumber.Text
dsNewRow.Item("HomeNumber") = txtHomeNumber.Text
dsNewRow.Item("HouseNumber") = txtHouseNumber.Text
dsNewRow.Item("Street") = txtStreet.Text
dsNewRow.Item("City") = txtCity.Text
dsNewRow.Item("County") = txtCounty.Text
dsNewRow.Item("PostCode") = txtPostCode.Text
dsNewRow.Item("AllergyDetails") = txtAllergyInfo.Text
'New row is added to the database.'
ds.Tables("tblPatients").Rows.Add(dsNewRow)
'Data set is updated.'
da.Update(ds, "tblPatients")
da = New SqlDataAdapter(Sql, dbconn)
da.Fill(ds, "tblPatients")
dbconn.Close()
Connect()
```

```
MessageBox.Show("The patient's details have been saved onto the  
database.")
```

```
'Form resets to original state.'  
'Combo boxes are reset.'  
cboPatientList.SelectedIndex = -1  
'Add, delete, edit and save buttons are disabled.'  
btnAdd.Enabled = False  
btnDelete.Enabled = False  
btnEdit.Enabled = False  
btnSave.Enabled = False  
'Combo box is enabled to select the patient.'  
cboPatientList.Enabled = True  
'Show and register buttons are enabled.'  
btnShow.Enabled = True  
btnRegister.Enabled = True  
  
'Combo box is reset.'  
cboTitle.SelectedIndex = -1  
'All text fields are cleared.'  
txtFirstName.Clear()  
txtSurname.Clear()  
txtGender.Clear()  
txtPhoneNumber.Clear()  
txtHomeNumber.Clear()  
txtHouseNumber.Clear()  
txtStreet.Clear()  
txtCity.Clear()  
txtCounty.Clear()  
txtPostCode.Clear()  
txtAllergyInfo.Clear()  
'Date of birth is reset to default.'  
DateOfBirth.Text = "01/01/1998"  
'Both check boxes are unchecked.'  
chbYes.Checked = False  
chbNo.Checked = False  
'Makes sure all combo boxes are disabled apart from the patient list.'  
cboTitle.Enabled = False  
'All text fields are read only to prevent editing.'  
txtFirstName.ReadOnly = True  
txtSurname.ReadOnly = True  
txtGender.ReadOnly = True  
txtPhoneNumber.ReadOnly = True  
txtHomeNumber.ReadOnly = True
```

```

        txtHouseNumber.ReadOnly = True
        txtStreet.ReadOnly = True
        txtCity.ReadOnly = True
        txtCounty.ReadOnly = True
        txtPostCode.ReadOnly = True
        txtAllergyInfo.ReadOnly = True
        DateOfBirth.Enabled = False

        'All combo boxes are cleared.'
        cboPatientList.Items.Clear()
        cboID.Items.Clear()

    End If

    'Sql statment will reload the patients first name and ID back onto the combo
boxes.'
    Sql = "SELECT * FROM tblPatients"
    da = New SqlDataAdapter(Sql, dbconn)
    Try
        ds.Tables("tblPatients").Clear()
    Catch ex As Exception

    End Try
    da.Fill(ds, "tblPatientName")
    Countrows = ds.Tables("tblPatientName").Rows.Count
    For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1
        cboID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))
    Next
    cboPatientList.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
    Next
End Sub

Public Function ValidatePatient(ByVal txtFirstName As TextBox, ByVal txtSurname As
TextBox, ByVal txtGender As TextBox, ByVal txtPhoneNumber As TextBox, ByVal
txtHomeNumber As TextBox, ByVal txtHouseNumber As TextBox, ByVal txtStreet As TextBox,
ByVal txtCity As TextBox, ByVal txtCounty As TextBox, ByVal txtPostCode As TextBox) As
Boolean
    AllFieldsValid = True
    'Using this line of validation, I can do a visible check on the all the text
boxes I need without having to make separeate lines for each visible check.'
    If txtFirstName.Text = "" Or txtSurname.Text = "" Or txtGender.Text = "" Or
txtPhoneNumber.Text = "" Or txtHomeNumber.Text = "" Or txtHouseNumber.Text = "" Or
txtStreet.Text = "" Or txtCity.Text = "" Or txtCounty.Text = "" Or txtPostCode.Text =
"" Or txtAllergyInfo.Text = "" Then
        MessageBox.Show("Blank box detected. Please fill in all required fields.")
        AllFieldsValid = False
    ElseIf IsNumeric(txtFirstName.Text) = True Or IsNumeric(txtSurname.Text) Or
IsNumeric(txtGender.Text) = True Then
        MessageBox.Show("First, Surname and Gender must not be numeric.")
        AllFieldsValid = False
    'Declaring the textbox as a date will make sure the date is in the correct
format.'
    'I make sure if the numeric text fields are numeric and if not then a
message box will appear.'
    ElseIf IsNumeric(txtPhoneNumber.Text) = False Or IsNumeric(txtHomeNumber.Text)
= False Or IsNumeric(txtHouseNumber.Text) = False Then
        MessageBox.Show("Phone numbers and Home number must be numeric.")
        AllFieldsValid = False
    'For these validations I have done a textlength check to make sure that
the fields are valid.'
    ElseIf txtFirstName.TextLength < 2 Or txtSurname.TextLength < 2 Then
        MessageBox.Show("First and Surname must be at least 2 characters or
above.")

```



```
AllFieldsValid = False
ElseIf txtFirstName.TextLength > 20 Or txtSurname.TextLength > 20 Then
    MessageBox.Show("First and Surname must be 20 characters less.")
    AllFieldsValid = False
    'Now I am able to validate to make sure that the field can only contain 1
character.'
ElseIf txtGender.Text <> "M" And txtGender.Text <> "F" Then
    MessageBox.Show("Gender must be 1 character. Either 'M' or 'F'.")
    AllFieldsValid = False
    'Telephone validated for 11 numbers. If not 11 numbers then error message
will appear.'
ElseIf txtPhoneNumber.TextLength < 11 Or txtPhoneNumber.TextLength > 11 Or
txtHomeNumber.TextLength < 11 Or txtHomeNumber.TextLength > 11 Then
    MessageBox.Show("Telephone numbers has to be 11 numbers long.")
    AllFieldsValid = False
ElseIf txtPostCode.TextLength < 6 Or txtPostCode.TextLength > 8 Then
    MessageBox.Show("Post Code must be between 6 to 8 characters.")
    'If the passwords do not match then a error message will appear.'
    AllFieldsValid = False
End If
Me.Refresh()
If AllFieldsValid = True Then
    'If this boolean value is true then It will return the function as true
which can be called in the login button.'
    ValidatePatient = True
Else
    'The value is returned as false therefore the user cannot proceed to the
next form.'
    ValidatePatient = False
End If
End Function

Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHelp.Click
    'A simple messagebox which appears telling the user how to use the current
form.'
    MessageBox.Show("Enter the patient's details onto the text fields, the
validated data will be saved onto the database which will add the new patient onto the
list. The member can then be selected for an appointment.")
End Sub

Private Sub frmPatientList_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'This variable is used to count the rows in a dataset.'
    Dim CountRows As Integer
    'This variable is used loop until I can find the row I'm looking for.'
    Dim i As Integer

    'Sets the combo box text to blank.'
    cboPatientList.SelectedIndex = -1
    'Add, delete, edit and save buttons are disabled.'
    btnAdd.Enabled = False
    btnDelete.Enabled = False
    btnEdit.Enabled = False
    btnSave.Enabled = False
    'The combo box containing the Patients names are loaded.'
    cboPatientList.Enabled = True
    'This button enables the patients details to be shown onto the text fields.'
    btnShow.Enabled = True
    'This allows the user to input data into the text fields.'
    btnRegister.Enabled = True
    'Sets the title to blank.'
```

```
cboTitle.SelectedIndex = -1
'Clears all textfields.'
txtFirstName.Clear()
txtSurname.Clear()
txtGender.Clear()
txtPhoneNumber.Clear()
txtHomeNumber.Clear()
txtHouseNumber.Clear()
txtStreet.Clear()
txtCity.Clear()
txtCounty.Clear()
txtPostCode.Clear()
txtAllergyInfo.Clear()
'Sets date of birth back to default.'
DateOfBirth.Text = "01/01/1998"

'Makes sure the label is invisible.'
lblAllergyCheck.Visible = False
'Disables the combo box for the patient's title.'
cboTitle.Enabled = False
'Makes all text fields read only so they can't be edited.'
txtFirstName.ReadOnly = True
txtSurname.ReadOnly = True
txtGender.ReadOnly = True
txtPhoneNumber.ReadOnly = True
txtHomeNumber.ReadOnly = True
txtHouseNumber.ReadOnly = True
txtStreet.ReadOnly = True
txtCity.ReadOnly = True
txtCounty.ReadOnly = True
txtPostCode.ReadOnly = True
txtAllergyInfo.ReadOnly = True
'Unchecks check boxes.'
chbNo.Enabled = False
chbYes.Enabled = False

'Selects all from the patients table.'
Sql = "SELECT * FROM tblPatients"
da = New SqlDataAdapter(Sql, dbconn)
Try
    ds.Tables("tblPatients").Clear()
Catch ex As Exception

End Try
Try
    ds.Tables("tblPatientName").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblPatientName")
Countrows = ds.Tables("tblPatientName").Rows.Count
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1
    'Retrieves all patients and sets their ID and first names into the combo
    box.'
    cboID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))

cboPatientList.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next
End Sub

Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
```

```
'This variable is used to count the rows in a dataset.'
Dim CountRows As Integer
'This variable is used loop until I can find the row I'm looking for.'
Dim i As Integer

'Sets the combo box text to blank.'
cboPatientList.SelectedIndex = -1
'Add, delete, edit and save buttons are disabled.'
btnAdd.Enabled = False
btnDelete.Enabled = False
btnEdit.Enabled = False
btnSave.Enabled = False
'The combo box containing the Patients names are loaded.'
cboPatientList.Enabled = True
'This button enables the patients details to be shown onto the text fields.'
btnShow.Enabled = True
'This allows the user to input data into the text fields.'
btnRegister.Enabled = True
'Sets the title to blank.'
cboTitle.SelectedIndex = -1
'Clears all textfields.'
txtFirstName.Clear()
txtSurname.Clear()
txtGender.Clear()
txtPhoneNumber.Clear()
txtHomeNumber.Clear()
txtHouseNumber.Clear()
txtStreet.Clear()
txtCity.Clear()
txtCounty.Clear()
txtPostCode.Clear()
txtAllergyInfo.Clear()
'Sets date of birth back to default.'
DateOfBirth.Text = "01/01/1998"

'Makes sure the label is invisible.'
lblAllergyCheck.Visible = False
'Disables the combo box for the patient's title.'
cboTitle.Enabled = False
'Makes all text fields read only so they can't be edited.'
txtFirstName.ReadOnly = True
txtSurname.ReadOnly = True
txtGender.ReadOnly = True
txtPhoneNumber.ReadOnly = True
txtHomeNumber.ReadOnly = True
txtHouseNumber.ReadOnly = True
txtStreet.ReadOnly = True
txtCity.ReadOnly = True
txtCounty.ReadOnly = True
txtPostCode.ReadOnly = True
txtAllergyInfo.ReadOnly = True
'Unchecks check boxes.'
chbNo.Enabled = False
chbYes.Enabled = False

cboPatientList.Items.Clear()
cboID.Items.Clear()

'Selects all from the patients table.'
Sql = "SELECT * FROM tblPatients"
da = New SqlDataAdapter(Sql, dbconn)
```

```
Try
    ds.Tables("tblPatientName").Clear()
Catch ex As Exception

End Try

Try
    ds.Tables("tblPatients").Clear()
Catch ex As Exception
End Try

da.Fill(ds, "tblPatientName")
CountRows = ds.Tables("tblPatientName").Rows.Count
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1
    'Retrieves all patients and sets their ID and first names into the combo
box.'
    cboID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))
cboPatientList.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next
End Sub

Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEdit.Click
    'Enables combo box for title.'
    cboTitle.Enabled = True
    'Turns off read only for text fields so that the user can input their details.'
    txtFirstName.ReadOnly = False
    txtSurname.ReadOnly = False
    txtGender.ReadOnly = False
    txtPhoneNumber.ReadOnly = False
    txtHomeNumber.ReadOnly = False
    txtStreet.ReadOnly = False
    txtCity.ReadOnly = False
    txtCounty.ReadOnly = False
    txtPostCode.ReadOnly = False
    txtHouseNumber.ReadOnly = False
    txtAllergyInfo.ReadOnly = False
    'Enables check boxes.'
    chbNo.Enabled = True
    chbYes.Enabled = True
    DateOfBirth.Enabled = True
    'Disables the delete button so the user can't delete the record while it's
being edited.'
    btnDelete.Enabled = False
    'Save button is enabled to save what the user has edited into the text
fields.'
    btnSave.Enabled = True

End Sub

Private Sub btnRegister_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRegister.Click
    'Makes sure that the user cannot load any profiles when they are making a new
one.'
    'Sets the value of the combo to blank.'
    cboPatientList.SelectedIndex = -1
    'Disables the combo box.'
    cboPatientList.Enabled = False
    'Disables edit, save, show and delete buttons.'
    btnEdit.Enabled = False
    btnSave.Enabled = False
```

```
btnShow.Enabled = False
'Add button is enabled to create a new row from the text fields.'
btnAdd.Enabled = True
btnDelete.Enabled = False
'Date of birth is enabled.'
DateOfBirth.Enabled = True
'Combo box for the title is enabled.'
cboTitle.Enabled = True
'Read only is disabled so all the text fields can be edited again.'
txtFirstName.ReadOnly = False
txtSurname.ReadOnly = False
txtGender.ReadOnly = False
txtPhoneNumber.ReadOnly = False
txtHomeNumber.ReadOnly = False
txtHouseNumber.ReadOnly = False
txtStreet.ReadOnly = False
txtCity.ReadOnly = False
txtCounty.ReadOnly = False
txtPostCode.ReadOnly = False
txtAllergyInfo.ReadOnly = False
'Check boxes are enabled.'
chbNo.Enabled = True
chbYes.Enabled = True
'Makes sure user cannot press the same button again.'
btnRegister.Enabled = False
End Sub

Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShow.Click
'Sets variable to true.'
AllFieldsValid = True
'Sets the selected patient as an integer.'
Dim SelectedPatient As Integer
'Presence check validation for the combo box.'

If cboPatientList.Text = "" Then
    MessageBox.Show("Please select a patient.")
    AllFieldsValid = False
    'Validation is not accepted.'
End If

If AllFieldsValid = True Then
    'Edit, and delete buttons are enabled.'
    btnEdit.Enabled = True
    'Patient list cannot be used to prevent loading another profile.'
    cboPatientList.Enabled = False
    'Add button is disabled when editing an existing profile.'
    btnAdd.Enabled = False
    btnRegister.Enabled = False
    btnShow.Enabled = False
    btnDelete.Enabled = True
    'Sets the combo box as an integer.'
    SelectedPatient = cboID.Text
    Try
        'Clears the dataset.'
        ds.Tables("tblSelectedPatient").Clear()
    Catch ex As Exception

    End Try

    'SQL statement retrieving all the patients details from the selected ID in
    the combo box.'
```

```

        Sql = "SELECT * FROM tblPatients WHERE PatientID = " & SelectedPatient &
""
        da = New SqlDataAdapter(Sql, dbconn)
        da.Fill(ds, "tblSelectedPatient")

        'Details are loaded onto the text fields.'
        cboTitle.Text = ds.Tables("tblSelectedPatient").Rows(0).Item("Title")
        txtFirstName.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("FirstName")
        txtSurname.Text = ds.Tables("tblSelectedPatient").Rows(0).Item("Surname")
        DateOfBirth.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("DateOfBirth")
        txtGender.Text = ds.Tables("tblSelectedPatient").Rows(0).Item("Gender")
        txtPhoneNumber.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("PhoneNumber")
        txtHomeNumber.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("HomeNumber")
        txtHouseNumber.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("HouseNumber")
        txtStreet.Text = ds.Tables("tblSelectedPatient").Rows(0).Item("Street")
        txtCity.Text = ds.Tables("tblSelectedPatient").Rows(0).Item("City")
        txtCounty.Text = ds.Tables("tblSelectedPatient").Rows(0).Item("County")
        txtPostCode.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("PostCode")
        lblAllergyCheck.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("Allergies")
        txtAllergyInfo.Text =
ds.Tables("tblSelectedPatient").Rows(0).Item("AllergyDetails")

        'Checks one of the check boxes according to the value of the label's
text.'
        Select Case lblAllergyCheck.Text

            Case "Y"

                chbYes.Checked = True

            Case "N"

                chbNo.Checked = True

        End Select

    End If

End Sub

Private Sub cboPatientList_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cboPatientList.SelectedIndexChanged
    'Sets the combo boxes to make the Patient's name match their ID.'
    SelectedPatientName = cboPatientList.SelectedIndex
    cboID.SelectedIndex = cboPatientList.SelectedIndex
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
    'Sets the selected patient as an integer.'
    Dim SelectedPatient As Integer
    'The combo box will be set as the integer.'
    SelectedPatient = cboID.Text

    Try

```

```
        'Clears the dataset.'
        ds.Tables("tblPatients").Clear()
    Catch ex As Exception

End Try
'SQL statement to delete the whole row where the PatientID matches the ID in
the combo box.'
Sql = "DELETE FROM tblPatients WHERE PatientID = " & SelectedPatient & ""

da = New SqlDataAdapter(Sql, dbconn)

da.Fill(ds, "tblPatients")

MessageBox.Show("Record deleted")

'This variable is used to count the rows in a dataset.'
Dim CountRows As Integer
'This variable is used loop until I can find the row I'm looking for.'
Dim i As Integer

'Sets the combo box text to blank.'
cboPatientList.SelectedIndex = -1
'Add, delete, edit and save buttons are disabled.'
btnAdd.Enabled = False
btnDelete.Enabled = False
btnEdit.Enabled = False
btnSave.Enabled = False
'The combo box containing the Patients names are loaded.'
cboPatientList.Enabled = True
'This button enables the patients details to be shown onto the text fields.'
btnShow.Enabled = True
'This allows the user to input data into the text fields.'
btnRegister.Enabled = True
'Sets the title to blank.'
cboTitle.SelectedIndex = -1
'Clears all textfields.'
txtFirstName.Clear()
txtSurname.Clear()
txtGender.Clear()
txtPhoneNumber.Clear()
txtHomeNumber.Clear()
txtHouseNumber.Clear()
txtStreet.Clear()
txtCity.Clear()
txtCounty.Clear()
txtPostCode.Clear()
txtAllergyInfo.Clear()
'Sets date of birth back to default.'
DateOfBirth.Text = "01/01/1998"

'Makes sure the label is invisible.'
lblAllergyCheck.Visible = False
'Disables the combo box for the patient's title.'
cboTitle.Enabled = False
'Makes all text fields read only so they can't be edited.'
txtFirstName.ReadOnly = True
txtSurname.ReadOnly = True
txtGender.ReadOnly = True
txtPhoneNumber.ReadOnly = True
txtHomeNumber.ReadOnly = True
txtHouseNumber.ReadOnly = True
txtStreet.ReadOnly = True
```

```

txtCity.ReadOnly = True
txtCounty.ReadOnly = True
txtPostCode.ReadOnly = True
txtAllergyInfo.ReadOnly = True
'Unchecks check boxes.'
chbNo.Enabled = False
chbYes.Enabled = False

cboPatientList.Items.Clear()
cboID.Items.Clear()

'Selects all from the patients table.'
Sql = "SELECT * FROM tblPatients"
da = New SqlDataAdapter(Sql, dbconn)
Try
    ds.Tables("tblPatients").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblPatientName")
CountRows = ds.Tables("tblPatientName").Rows.Count
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1
    'Retrieves all patients and sets their ID and first names into the combo
box.'
    cboID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))

cboPatientList.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next

End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    'I dim 'response' as a message result in order to bring up an input box for
the user to decide if they are certain to proceed.'
    Dim response As MsgBoxResult
    'Sets this variable for dentist's name as a string.'
    Dim SelectedPatient As Integer
    SelectedPatient = cboID.Text
    'This is the repsonse used to decide if the user wants to proceed.'
    response = MsgBox("Are you sure you want to update?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Confirm")
    'If the response is yes then validation is called.'
    If response = MsgBoxResult.Yes Then
        'Validation is called. If it is passed as true, the SQL statement will be
called.'
        If ValidatePatient(txtFirstName, txtSurname, txtGender, txtPhoneNumber,
txtHomeNumber, txtHouseNumber, txtStreet, txtCity, txtCounty, txtPostCode) = True Then
            'SQL statement selected the whole row from the patients table where
the patient ID matches the selected patient from the combo box.'
            Sql = "SELECT * FROM tblpatients WHERE PatientID = " & SelectedPatient
            da = New SqlDataAdapter(Sql, dbconn)
            Dim cb As New SqlCommandBuilder(da)
            da.Fill(ds, "tblUpdatePatient")

            'The selected row is updated with the details from the text fields.'
            ds.Tables("tblUpdatePatient").Rows(0).Item("Title") = cboTitle.Text
            ds.Tables("tblUpdatePatient").Rows(0).Item("FirstName") =
txtFirstName.Text
            ds.Tables("tblUpdatePatient").Rows(0).Item("Surname") =
txtSurname.Text

```



```

        ds.Tables("tblUpdatePatient").Rows(0).Item("DateOfBirth") =
DateOfBirth.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("Gender") = txtGender.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("PhoneNumber") =
txtPhoneNumber.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("HomeNumber") =
txtHomeNumber.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("HouseNumber") =
txtHouseNumber.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("Street") = txtStreet.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("City") = txtCity.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("County") = txtCounty.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("AllergyDetails") =
txtAllergyInfo.Text
        ds.Tables("tblUpdatePatient").Rows(0).Item("Allergies") =
lblAllergyCheck.Text
        'The dataset is updated.'
        da.Update(ds, "tblUpdatePatient")

        'MessageBox shows if the procedure is succesful.'
        MessageBox.Show("Details have been saved and updated")
    Else
        MessageBox.Show("Invalid Row, no changes made.")
    End If
End If

'This variable is used to count the rows in a dataset.'
Dim CountRows As Integer
'This variable is used loop until I can find the row I'm looking for.'
Dim i As Integer

'Sets the combo box text to blank.'
cboPatientList.SelectedIndex = -1
'Add, delete, edit and save buttons are disabled.'
btnAdd.Enabled = False
btnDelete.Enabled = False
btnEdit.Enabled = False
btnSave.Enabled = False
'The combo box containing the Patients names are loaded.'
cboPatientList.Enabled = True
'This button enables the patients details to be shown onto the text fields.'
btnShow.Enabled = True
'This allows the user to input data into the text fields.'
btnRegister.Enabled = True
'Sets the title to blank.'
cboTitle.SelectedIndex = -1
'Clears all textfields.'
txtFirstName.Clear()
txtSurname.Clear()
txtGender.Clear()
txtPhoneNumber.Clear()
txtHomeNumber.Clear()
txtHouseNumber.Clear()
txtStreet.Clear()
txtCity.Clear()
txtCounty.Clear()
txtPostCode.Clear()
txtAllergyInfo.Clear()
'Sets date of birth back to default.'
DateOfBirth.Text = "01/01/1998"

'Makes sure the label is invisible.'

```

```
lblAllergyCheck.Visible = False
'Disables the combo box for the patient's title.'
cboTitle.Enabled = False
'Makes all text fields read only so they can't be edited.'
txtFirstName.ReadOnly = True
txtSurname.ReadOnly = True
txtGender.ReadOnly = True
txtPhoneNumber.ReadOnly = True
txtHomeNumber.ReadOnly = True
txtHouseNumber.ReadOnly = True
txtStreet.ReadOnly = True
txtCity.ReadOnly = True
txtCounty.ReadOnly = True
txtPostCode.ReadOnly = True
txtAllergyInfo.ReadOnly = True
'Unchecks check boxes.'
chbNo.Enabled = False
chbYes.Enabled = False
'Clears combo box of items.'
cboPatientList.Items.Clear()
cboID.Items.Clear()

'Selects all from the patients table.'
Sql = "SELECT * FROM tblPatients"
da = New SqlDataAdapter(Sql, dbconn)
Try
    ds.Tables("tblPatientName").Clear()
Catch ex As Exception

End Try

Try
    ds.Tables("tblPatients").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblPatientName")
CountRows = ds.Tables("tblPatientName").Rows.Count
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1
    'Retrieves all patients and sets their ID and first names into the combo
box.'
    cboID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))

cboPatientList.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next
End Sub

Private Sub txtFirstName_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtFirstName.KeyPress
    'Ignore everything but letter keys.'
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
e.Handled = True
    End Sub

Private Sub txtSurname_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtSurname.KeyPress
    'Ignore everything but letter keys.'
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
e.Handled = True
    End Sub
```

```
Private Sub txtGender_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtGender.KeyPress
    'Ignore everything but letter keys.'
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
e.Handled = True
    End Sub

Private Sub txtPhoneNumber_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtPhoneNumber.KeyPress
    'Ignore everything but numeric characters.'
    If Not Char.IsDigit(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
e.Handled = True
    End Sub

Private Sub txtHomeNumber_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtHomeNumber.KeyPress
    'Ignore everything but numeric characters.'
    If Not Char.IsDigit(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
e.Handled = True
    End Sub

Private Sub txtHouseNumber_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtHouseNumber.KeyPress
    'Ignore everything but numeric characters.'
    If Not Char.IsDigit(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
e.Handled = True
    End Sub

End Class
```

frmAppointmentList

These combo boxes will contain the Appointment ID, Patient ID, Treatments, and Dentist ID's along with their first names.

The show button will display the booking details on the combo boxes and text fields.

The edit button will enable the text fields once the details are shown.

The save button will validate the text fields and save the data back into the database if validated correctly.

The cancel button will refresh the form.

This help button will bring up a message providing further details and help in using the current form.

This button will close the current form and return back to menu form.

This button will disable the functions to edit, save and delete an existing patient.

The add button will call in validation to check each text field to see if the data is correctly inputted. This will then add a new row into the dataset. This will create a new appointment.

This will delete the selected appointment off the booking table.

The screenshot shows a form titled 'frmAppointmentList'. It contains several sections: 'Appointments:' with a dropdown menu showing '1' and a 'Show' button; 'Dentists:' with a dropdown menu showing '3' and a text field showing 'Rosie'; 'Treatments:' with a dropdown menu showing '2' and a text field showing 'Polishing'; 'Patients:' with a dropdown menu showing '7' and a text field showing 'Bob'. Below these are 'Rooms:' with radio buttons for '1 - Fillings', '2 - Polishing', '3 - Dental Care', and '4 - Crowns' (which is selected). To the right of the rooms are 'Date:' (24 June 2016), 'Time:' (12:30), and 'Treatment Duration:' (60). At the bottom are buttons for 'Back', 'Create Appointment', 'Add Appointment', and 'Delete Appointment'. On the right side of the form are buttons for 'Edit', 'Save', 'Cancel', and a help button with a question mark icon. Arrows point from the text boxes to the corresponding UI elements.

```
Imports System.Data.SqlClient
Public Class frmAppointmentList
    'Sets the selected patient as an integer.'
    Dim SelectedPatient As Integer
    'Sets the selected dentist as an integer.'
    Dim SelectedDentist As Integer
    'This variable is used to count the rows in a dataset.'
    Dim CountRows As Integer
    'This variable is used loop until I can find the row I'm looking for.'
    Dim i As Integer

    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'The current form will close.'
        Me.Close()
        'Returns to the menu form.'
        frmMenu.Show()

    End Sub

    Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHelp.Click
        'The help button will display a message helping the user with the form.'
        MessageBox.Show("The appointment list can select each individual existing
appointment in order to be edited. New appointments can be made by assigning a dentist
to a patient and selecting the treatments and other details.")
    End Sub

    Private Sub cboPatients_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cboPatients.SelectedIndexChanged
        'Sets the combo boxes to make the Patient's name match their ID.'
        SelectedPatient = cboPatients.SelectedIndex
        cboPatientID.SelectedIndex = cboPatients.SelectedIndex
    End Sub

    Private Sub cboDentists_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cboDentists.SelectedIndexChanged
        'Sets the combo boxes to make the Dentist's name match their ID.'
        SelectedPatient = cboDentists.SelectedIndex
        cboDentistID.SelectedIndex = cboDentists.SelectedIndex
    End Sub

    Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEdit.Click
        'Enables combo boxes.'
        cboDentists.Enabled = True
        cboPatients.Enabled = True
        cboAppointmentList.Enabled = False
        'Disables show and edit buttons.'
        btnShow.Enabled = False
        btnEdit.Enabled = False
        'Enabled save button.'
        btnSave.Enabled = True
        'Prevents appointments being deleted while editing.'
        btnDeleteAppointment.Enabled = False
        'Makes combo box for treatments enabled.'
        cboTreatments.Enabled = True
        'Makes text fields editable again.'
        txtTime.ReadOnly = False
        'Date enabled.'
        AppointmentDate.Enabled = True
        'All radio buttons are enabled.'
```

```
rdoRoom1.Enabled = True
rdoRoom2.Enabled = True
rdoRoom3.Enabled = True
rdoRoom4.Enabled = True
```

```
End Sub
```

```
Private Sub frmAppointmentList_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'This variable is used to count the rows in a dataset.'
    Dim CountRows As Integer
    'This variable is used loop until I can find the row I'm looking for.'
    Dim i As Integer
    'Combobox for ID's are disabled.'
    cboTreatmentID.Enabled = False
    lblRoomID.Visible = False
    'combo boxes are set to blank.'
    cboAppointmentList.SelectedIndex = -1
    cboDentists.SelectedIndex = -1
    cboPatients.SelectedIndex = -1
    cboTreatments.SelectedIndex = -1
    'Add, delete, edit and save buttons are disabled.'
    btnAddAppointment.Enabled = False
    btnDeleteAppointment.Enabled = False
    btnEdit.Enabled = False
    btnSave.Enabled = False
    'Appointment lists are enabled.'
    cboAppointmentList.Enabled = True
    'The show, cancel and register buttons are enabled.'
    btnShow.Enabled = True
    btnCancel.Enabled = True
    btnRegister.Enabled = True
    'All other combo boxes are disabled.'
    cboDentistID.Enabled = False
    cboPatientID.Enabled = False
    cboPatients.Enabled = False
    cboDentists.Enabled = False
    cboTreatments.Enabled = False
    'Text fields are set to read only to prevent editing.'
    txtTime.ReadOnly = True
    txtTreatmentDuration.ReadOnly = True
    AppointmentDate.Enabled = False
    'All radio buttons are disabled and unchecked.'
    rdoRoom1.Enabled = False
    rdoRoom2.Enabled = False
    rdoRoom3.Enabled = False
    rdoRoom4.Enabled = False
    rdoRoom1.Checked = False
    rdoRoom2.Checked = False
    rdoRoom3.Checked = False
    rdoRoom4.Checked = False

    'All combo boxes and text fields are cleared.'
    cboDentistID.Items.Clear()
    cboPatientID.Items.Clear()
    cboDentists.Items.Clear()
    cboPatients.Items.Clear()
    cboAppointmentList.Items.Clear()
    'Appointment date set to today by default.'
    AppointmentDate.Text = Date.Today
    txtTime.Clear()
    txtTreatmentDuration.Clear()
```

'This statement will load all dentists into the combo box for the dentists ID and first names.'

```
Sql = "SELECT * FROM tblDentists"  
da = New SqlDataAdapter(Sql, dbconn)
```

```
Try  
    'Dataset is cleared first before retrieving the updated list.'  
    ds.Tables("tblDentists").Clear()
```

```
Catch ex As Exception
```

```
End Try
```

```
Try  
    'Dataset is cleared first before retrieving the updated list.'  
    ds.Tables("tblDentistsName").Clear()
```

```
Catch ex As Exception
```

```
End Try
```

```
da.Fill(ds, "tblDentistsName")  
Countrows = ds.Tables("tblDentistsName").Rows.Count  
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1
```

```
cboDentistID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))
```

```
cboDentists.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))  
Next
```

'This SQL statement will load all the patient's ID and first name into the combo boxes.'

```
Sql = "SELECT * FROM tblPatients"  
da = New SqlDataAdapter(Sql, dbconn)
```

```
Try  
    'Dataset is cleared first before retrieving the updated list.'  
    ds.Tables("tblPatients").Clear()
```

```
Catch ex As Exception
```

```
End Try
```

```
Try  
    'Dataset is cleared first before retrieving the updated list.'  
    ds.Tables("tblPatientName").Clear()
```

```
Catch ex As Exception
```

```
End Try
```

```
da.Fill(ds, "tblPatientName")  
Countrows = ds.Tables("tblPatientName").Rows.Count  
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1
```

```
cboPatientID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))
```

```
cboPatients.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))  
Next
```

'This SQL statement will load all the booking appointment ID's into the combo box.'

```
Sql = "SELECT * FROM tblBooking"  
da = New SqlDataAdapter(Sql, dbconn)
```

```
Try  
    'Dataset is cleared first before retrieving the updated list.'
```

```
        ds.Tables("tblBooking").Clear()

    Catch ex As Exception

    End Try

    Try
        'Dataset is cleared first before retrieving the updated list.'
        ds.Tables("tblNewBooking").Clear()
    Catch ex As Exception

    End Try

    da.Fill(ds, "tblNewBooking")
    CountRows = ds.Tables("tblNewBooking").Rows.Count
    For i = 0 To ds.Tables("tblNewBooking").Rows.Count - 1

cboAppointmentList.Items.Add(ds.Tables("tblNewBooking").Rows(i).Item("AppointmentID"))
        Next

    End Sub

    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click

        'This variable is used to count the rows in a dataset.'
        Dim CountRows As Integer
        'This variable is used loop until I can find the row I'm looking for.'
        Dim i As Integer
        'Combobox for ID's are disabled.'
        cboTreatmentID.Enabled = False
        lblRoomID.Visible = False
        'combo boxes are set to blank.'
        cboAppointmentList.SelectedIndex = -1
        cboDentists.SelectedIndex = -1
        cboPatients.SelectedIndex = -1
        cboTreatments.SelectedIndex = -1
        'Add, delete, edit and save buttons are disabled.'
        btnAddAppointment.Enabled = False
        btnDeleteAppointment.Enabled = False
        btnEdit.Enabled = False
        btnSave.Enabled = False
        'Appointment lists are enabled.'
        cboAppointmentList.Enabled = True
        'The show, cancel and register buttons are enabled.'
        btnShow.Enabled = True
        btnCancel.Enabled = True
        btnRegister.Enabled = True
        'All other combo boxes are disabled.'
        cboDentistID.Enabled = False
        cboPatientID.Enabled = False
        cboPatients.Enabled = False
        cboDentists.Enabled = False
        cboTreatments.Enabled = False
        'Text fields are set to read only to prevent editing.'
        txtTime.ReadOnly = True
        txtTreatmentDuration.ReadOnly = True
        AppointmentDate.Enabled = False
        'All radio buttons are disabled and unchecked.'
        rdoRoom1.Enabled = False
```



```
rdoRoom2.Enabled = False
rdoRoom3.Enabled = False
rdoRoom4.Enabled = False
rdoRoom1.Checked = False
rdoRoom2.Checked = False
rdoRoom3.Checked = False
rdoRoom4.Checked = False

'All combo boxes and text fields are cleared.'
cboDentistID.Items.Clear()
cboPatientID.Items.Clear()
cboDentists.Items.Clear()
cboPatients.Items.Clear()
cboAppointmentList.Items.Clear()
'Appointment date set to today by default.'
AppointmentDate.Text = Date.Today
txtTime.Clear()
txtTreatmentDuration.Clear()

'This statement will load all dentists into the combo box for the dentists ID
and first names.'
Sql = "SELECT * FROM tblDentists"
da = New SqlDataAdapter(Sql, dbconn)

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblDentists").Clear()

Catch ex As Exception

End Try
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblDentistsName").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblDentistsName")
CountRows = ds.Tables("tblDentistsName").Rows.Count
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1

cboDentistID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))

cboDentists.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
Next

'This SQL statement will load all the patient's ID and first name into the
combo boxes.'
Sql = "SELECT * FROM tblPatients"
da = New SqlDataAdapter(Sql, dbconn)

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblPatients").Clear()
Catch ex As Exception

End Try

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblPatientName").Clear()
Catch ex As Exception
```

```
End Try

da.Fill(ds, "tblPatientName")
CountRows = ds.Tables("tblPatientName").Rows.Count
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1

cboPatientID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))

cboPatients.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next

'This SQL statement will load all the booking appointment ID's into the combo
box.'
Sql = "SELECT * FROM tblBooking"
da = New SqlDataAdapter(Sql, dbconn)
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblBooking").Clear()

Catch ex As Exception

End Try

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblNewBooking").Clear()
Catch ex As Exception

End Try

da.Fill(ds, "tblNewBooking")
CountRows = ds.Tables("tblNewBooking").Rows.Count
For i = 0 To ds.Tables("tblNewBooking").Rows.Count - 1

cboAppointmentList.Items.Add(ds.Tables("tblNewBooking").Rows(i).Item("AppointmentID"))
Next

End Sub

Private Sub btnRegister_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRegister.Click
    'Sets the combo box to blank.'
    cboAppointmentList.SelectedIndex = -1
    'Combo box for appointment list is disabled.'
    cboAppointmentList.Enabled = False
    'Show, edit and save buttons are disabled.'
    btnShow.Enabled = False
    btnEdit.Enabled = False
    btnSave.Enabled = False
    'Add button is enabled.'
    btnAddAppointment.Enabled = True
    'Delete buttons are disabled.'
    btnDeleteAppointment.Enabled = False
    'The combo boxes and text fields are enable and editable.'
    cboDentists.Enabled = True
    cboPatients.Enabled = True
    cboTreatments.Enabled = True
    txtTime.ReadOnly = False
    txtTreatmentDuration.ReadOnly = True
    AppointmentDate.Enabled = True
```

```
End Sub

Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShow.Click
    'Validation variable used to make sure there is a value in the combo box.'
    AllFieldsValid = True
    'Sets selected appointment as an integer.'
    Dim SelectedAppointment As Integer
    'Validation for presence check for the combo box containing the appointment
    lists.'
    If cboAppointmentList.Text = "" Then
        MessageBox.Show("Please select an appointment")
        'Passes the validation as false.'
        AllFieldsValid = False
    End If
    'When the validation is correct.'
    If AllFieldsValid = True Then
        'Combo box for appointment list is set as an integer.'
        SelectedAppointment = cboAppointmentList.Text

        Try
            'Clears the dataset.'
            ds.Tables("tblSelectedAppointment").Clear()
        Catch ex As Exception

        End Try

        Try
            'Clears the dataset for dentists name.'
            ds.Tables("tblDentistName").Clear()
        Catch ex As Exception

        End Try
        'Makes sure all ID's are set as an integer.'
        Dim ID As Integer
        Dim PatientID As Integer

        'SQL statement selecting all rows where the appointment ID matches the
        value in the combo box.'
        Sql = "SELECT * FROM tblBooking WHERE AppointmentID = " &
SelectedAppointment & ""
        da = New SqlDataAdapter(Sql, dbconn)
        da.Fill(ds, "tblSelectedAppointment")

        'All the appointment details are loaded onto the text fields.'
        AppointmentDate.Text =
ds.Tables("tblSelectedAppointment").Rows(0).Item("AppointmentDate")
        txtTime.Text =
ds.Tables("tblSelectedAppointment").Rows(0).Item("AppointmentTime")
        cboTreatmentID.Text =
ds.Tables("tblSelectedAppointment").Rows(0).Item("TreatmentID")
        txtTreatmentDuration.Text =
ds.Tables("tblSelectedAppointment").Rows(0).Item("TreatmentDuration")
        cboDentistID.Text =
ds.Tables("tblSelectedAppointment").Rows(0).Item("DentistID")
        cboPatientID.Text =
ds.Tables("tblSelectedAppointment").Rows(0).Item("PatientID")
        lblRoomID.Text =
ds.Tables("tblSelectedAppointment").Rows(0).Item("RoomID")
        ID = ds.Tables("tblSelectedAppointment").Rows(0).Item("DentistID")
        PatientID = ds.Tables("tblSelectedAppointment").Rows(0).Item("PatientID")
    End If
End Sub
```

```
text.'
    'This checks one of the radio buttons according to the value of the label
text.'
    Select Case CInt(lblRoomID.Text)
    Case 2
        rdoRoom1.Checked = True
    Case 3
        rdoRoom2.Checked = True
    Case 4
        rdoRoom3.Checked = True
    Case 5
        rdoRoom4.Checked = True
    End Select

    'SQL statement selecting the dentist from the combo box containing the
loaded ID from the selected appointment.'
    Sql = "SELECT * FROM tblDentists WHERE DentistID = " & ID
    da = New SqlDataAdapter(Sql, dbconn)
    da.Fill(ds, "DentistName")
    'Loads first name onto combo box.'
    cboDentists.Text = ds.Tables("DentistName").Rows(0).Item("DFirstName")

    'SQL statement selecting the patient from the combo box containing the
loaded ID from the selected appointment.'
    Sql = "SELECT * FROM tblPatients WHERE PatientID = " & PatientID
    da = New SqlDataAdapter(Sql, dbconn)
    da.Fill(ds, "PatientName")
    'Loads first name onto combo box.'
    cboPatients.Text = ds.Tables("PatientName").Rows(0).Item("FirstName")

    Try
        'Clears the dataset.'
        ds.Tables("DentistName").Clear()
    Catch ex As Exception

    End Try

    Try
        'Clears the dataset.'
        ds.Tables("PatientName").Clear()
    Catch ex As Exception

    End Try

    'Disables the show button and combo box for the appointment list.'
    btnShow.Enabled = False
    cboAppointmentList.Enabled = False
    'Enables edit and delete buttons.'
    'Disables save, add and register buttons.'
    btnEdit.Enabled = True
    btnSave.Enabled = False
    btnDeleteAppointment.Enabled = True
    btnAddAppointment.Enabled = False
    btnRegister.Enabled = False
End If
End Sub

Private Sub cboTreatments_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cboTreatments.SelectedIndexChanged

    'I create a case statement in order to change the text in the treatment
duration field so that it matches the treatment when the treatment type is selected.'
```

```

Select Case cboTreatments.SelectedItem
    'If Fillings is selected the duration will be 30 minutes.'
    Case "Fillings"
        txtTreatmentDuration.Text = "30"
        rdoRoom1.Checked = True
        lblRoomID.Text = "2"
        cboTreatmentID.Text = "1"
        'If Polishing is selected the duration will be 60 minutes.'
    Case "Polishing"
        txtTreatmentDuration.Text = "60"
        rdoRoom2.Checked = True
        lblRoomID.Text = "3"
        cboTreatmentID.Text = "2"
        'If Patient Care is selected the duration will be 60 minutes.'
    Case "Patient Care"
        txtTreatmentDuration.Text = "60"
        rdoRoom3.Checked = True
        lblRoomID.Text = "4"
        cboTreatmentID.Text = "3"
        'If Crowns is selected the duration will be 60 minutes.'
    Case "Crowns"
        txtTreatmentDuration.Text = "60"
        rdoRoom4.Checked = True
        lblRoomID.Text = "5"
        cboTreatmentID.Text = "4"
        'I make sure that the text box is cleared when the cbo field is
empty.'

    Case cboTreatments.SelectedIndex = -1
        txtTreatmentDuration.Clear()

End Select
End Sub

Private Sub btnAddAppointment_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddAppointment.Click
    'This variable is used loop until I can find the row I'm looking for.'
    Dim i As Integer
    'Calls in validation function for the text fields and combo boxes.'
    If ValidateFields(txtTime, cboDentists, cboPatients) = True Then
        Dim Sql As String = ""

        Try
            'Dataset is cleared.'
            ds.Tables("tblBooking").Clear()
        Catch ex As Exception
        End Try

        Try
            'Dataset is cleared.'
            ds.Tables("tblNewBooking").Clear()
        Catch ex As Exception
        End Try

        'SQL to create a new row in the booking table.'
        Sql = "Select * from tblBooking"
        da = New SqlDataAdapter(Sql, dbconn)
        da.Fill(ds, "tblBooking")
        Dim cb As New SqlCommandBuilder(da)
        Dim dsNewRow As DataRow
        'New row in dataset.'
        dsNewRow = ds.Tables("tblBooking").NewRow()
    
```

```
'Each radio button has a set ID.'
If rdoRoom1.Checked = True Then
    dsNewRow.Item("RoomID") = 2
ElseIf rdoRoom2.Checked = True Then
    dsNewRow.Item("RoomID") = 3
ElseIf rdoRoom3.Checked = True Then
    dsNewRow.Item("RoomID") = 4
ElseIf rdoRoom4.Checked = True Then
    dsNewRow.Item("RoomID") = 5
End If

'All details are added as a new row from the text fields.'
dsNewRow.Item("AppointmentDate") = AppointmentDate.Value
dsNewRow.Item("AppointmentTime") = txtTime.Text
dsNewRow.Item("PatientID") = CInt(cboPatientID.Text)
dsNewRow.Item("DentistID") = CInt(cboDentistID.Text)
dsNewRow.Item("TreatmentID") = CInt(cboTreatmentID.Text)
dsNewRow.Item("TreatmentDuration") = CInt(txtTreatmentDuration.Text)

'New row added.'
ds.Tables("tblBooking").Rows.Add(dsNewRow)
'Dataset is updated.'
da.Update(ds, "tblBooking")
'Closes the connection.'
dbconn.Close()
'Message box showing the appointment has been successfully booked.'
MessageBox.Show("Your appointment has been booked onto the system.")

'This variable is used to count the rows in a dataset.'
Dim CountRows As Integer
'This variable is used loop until I can find the row I'm looking for.'
'Combobox for ID's are disabled.'
cboTreatmentID.Enabled = False
lblRoomID.Visible = False
'combo boxes are set to blank.'
cboAppointmentList.SelectedIndex = -1
cboDentists.SelectedIndex = -1
cboPatients.SelectedIndex = -1
cboTreatments.SelectedIndex = -1
'Add, delete, edit and save buttons are disabled.'
btnAddAppointment.Enabled = False
btnDeleteAppointment.Enabled = False
btnEdit.Enabled = False
btnSave.Enabled = False
'Appointment lists are enabled.'
cboAppointmentList.Enabled = True
'The show, cancel and register buttons are enabled.'
btnShow.Enabled = True
btnCancel.Enabled = True
btnRegister.Enabled = True
'All other combo boxes are disabled.'
cboDentistID.Enabled = False
cboPatientID.Enabled = False
cboPatients.Enabled = False
cboDentists.Enabled = False
cboTreatments.Enabled = False
'Text fields are set to read only to prevent editing.'
txtTime.ReadOnly = True
txtTreatmentDuration.ReadOnly = True
AppointmentDate.Enabled = False
'All radio buttons are disabled and unchecked.'
rdoRoom1.Enabled = False
```

```
rdoRoom2.Enabled = False
rdoRoom3.Enabled = False
rdoRoom4.Enabled = False
rdoRoom1.Checked = False
rdoRoom2.Checked = False
rdoRoom3.Checked = False
rdoRoom4.Checked = False

'All combo boxes and text fields are cleared.'
cboDentistID.Items.Clear()
cboPatientID.Items.Clear()
cboDentists.Items.Clear()
cboPatients.Items.Clear()
cboAppointmentList.Items.Clear()
'Appointment date set to today by default.'
AppointmentDate.Text = Date.Today
txtTime.Clear()
txtTreatmentDuration.Clear()

'This statement will load all dentists into the combo box for the dentists
ID and first names.'
Sql = "SELECT * FROM tblDentists"
da = New SqlDataAdapter(Sql, dbconn)

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblDentists").Clear()

Catch ex As Exception

End Try
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblDentistsName").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblDentistsName")
CountRows = ds.Tables("tblDentistsName").Rows.Count
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1

cboDentistID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))

cboDentists.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
Next

'This SQL statement will load all the patient's ID and first name into the
combo boxes.'
Sql = "SELECT * FROM tblPatients"
da = New SqlDataAdapter(Sql, dbconn)

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblPatients").Clear()
Catch ex As Exception

End Try

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblPatientName").Clear()
Catch ex As Exception
```

```
End Try

da.Fill(ds, "tblPatientName")
CountRows = ds.Tables("tblPatientName").Rows.Count
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1

cboPatientID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))

cboPatients.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next

'This SQL statement will load all the booking appointment ID's into the
combo box.'
Sql = "SELECT * FROM tblBooking"
da = New SqlDataAdapter(Sql, dbconn)
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblBooking").Clear()

Catch ex As Exception

End Try

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblNewBooking").Clear()
Catch ex As Exception

End Try

da.Fill(ds, "tblNewBooking")
CountRows = ds.Tables("tblNewBooking").Rows.Count
For i = 0 To ds.Tables("tblNewBooking").Rows.Count - 1

cboAppointmentList.Items.Add(ds.Tables("tblNewBooking").Rows(i).Item("AppointmentID"))
Next

End If

End Sub
Public Function ValidateFields(ByVal txtTime As TextBox, ByVal cboDentists As
ComboBox, ByVal cboPatients As ComboBox) As Boolean
    AllFieldsValid = True
    'Using this line of validation, I can do a visible check on the all the text
boxes I need without having to make separate lines for each visible check.'
    If txtTime.Text = "" Or txtTreatmentDuration.Text = "" Or cboDentists.Text =
"" Or cboPatients.Text = "" Then
        MessageBox.Show("Blank box detected. Please fill in all required fields.")
        AllFieldsValid = False
    End If
    If AllFieldsValid = True Then

        'If this boolean value is true then It will return the function as true
which can be called in the login button.'
        ValidateFields = True
    Else
        'The value is returned as false therefore the user cannot proceed to the
next form.'
        ValidateFields = False
    End If
End Function
```



```

Private Sub cboTreatmentID_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cboTreatmentID.SelectedIndexChanged
    'Sets the treatment text to different treatments depending on what the ID is
    set to.'
    If cboTreatmentID.Text = "1" Then
        cboTreatments.Text = "Fillings"
    ElseIf cboTreatmentID.Text = "2" Then
        cboTreatments.Text = "Polishing"
    ElseIf cboTreatmentID.Text = "3" Then
        cboTreatments.Text = "Patient Care"
    ElseIf cboTreatmentID.Text = "4" Then
        cboTreatments.Text = "Crowns"
    End If
End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    'I dim 'response' as a message result in order to bring up an input box for
    the user to decide if they are certain to proceed.'
    Dim response As MsgBoxResult
    'Sets the selected appointment as an integer.'
    Dim SelectedAppointment As Integer
    'Sets the combo box as an integer.'
    SelectedAppointment = cboAppointmentList.Text
    'This is the response used to decide if the user wants to proceed.'
    response = MsgBox("Are you sure you want to update?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Confirm")
    'If the response is yes then validation is called.'
    If response = MsgBoxResult.Yes Then
        If ValidateFields(txtTime, cboDentists, cboPatients) = True Then
            'SQL statement selecting the row from the appointment ID in the combo
            box.'
            Sql = "SELECT * FROM tblBooking WHERE AppointmentID = " &
SelectedAppointment
            da = New SqlDataAdapter(Sql, dbconn)
            'Dataset is filled with updated data.'
            da.Fill(ds, "tblAppointmentUpdate")

            Dim cb As New SqlCommandBuilder(da)
            'Updated data is saved back into the row from the textfields.'
            ds.Tables("tblAppointmentUpdate").Rows(0).Item("AppointmentDate") =
AppointmentDate.Value
            ds.Tables("tblAppointmentUpdate").Rows(0).Item("AppointmentTime") =
txtTime.Text
            ds.Tables("tblAppointmentUpdate").Rows(0).Item("PatientID") =
cboPatientID.Text
            ds.Tables("tblAppointmentUpdate").Rows(0).Item("DentistID") =
cboDentistID.Text
            ds.Tables("tblAppointmentUpdate").Rows(0).Item("RoomID") =
lblRoomID.Text
            ds.Tables("tblAppointmentUpdate").Rows(0).Item("TreatmentID") =
cboTreatmentID.Text
            ds.Tables("tblAppointmentUpdate").Rows(0).Item("TreatmentDuration") =
txtTreatmentDuration.Text
            'Dataset is updated.'
            da.Update(ds, "tblAppointmentUpdate")
            'Message boxes to show if the update was successful or not.'
            MessageBox.Show("Details have been saved and updated")
        Else
            MessageBox.Show("Invalid Row, no changes made.")
        End If
    End If
End Sub

```

```
        End If
    End If

    'This variable is used to count the rows in a dataset.'
    Dim CountRows As Integer
    'This variable is used loop until I can find the row I'm looking for.'
    Dim i As Integer
    'Combobox for ID's are disabled.'
    cboTreatmentID.Enabled = False
    lblRoomID.Visible = False
    'combo boxes are set to blank.'
    cboAppointmentList.SelectedIndex = -1
    cboDentists.SelectedIndex = -1
    cboPatients.SelectedIndex = -1
    cboTreatments.SelectedIndex = -1
    'Add, delete, edit and save buttons are disabled.'
    btnAddAppointment.Enabled = False
    btnDeleteAppointment.Enabled = False
    btnEdit.Enabled = False
    btnSave.Enabled = False
    'Appointment lists are enabled.'
    cboAppointmentList.Enabled = True
    'The show, cancel and register buttons are enabled.'
    btnShow.Enabled = True
    btnCancel.Enabled = True
    btnRegister.Enabled = True
    'All other combo boxes are disabled.'
    cboDentistID.Enabled = False
    cboPatientID.Enabled = False
    cboPatients.Enabled = False
    cboDentists.Enabled = False
    cboTreatments.Enabled = False
    'Text fields are set to read only to prevent editing.'
    txtTime.ReadOnly = True
    txtTreatmentDuration.ReadOnly = True
    AppointmentDate.Enabled = False
    'All radio buttons are disabled and unchecked.'
    rdoRoom1.Enabled = False
    rdoRoom2.Enabled = False
    rdoRoom3.Enabled = False
    rdoRoom4.Enabled = False
    rdoRoom1.Checked = False
    rdoRoom2.Checked = False
    rdoRoom3.Checked = False
    rdoRoom4.Checked = False

    'All combo boxes and text fields are cleared.'
    cboDentistID.Items.Clear()
    cboPatientID.Items.Clear()
    cboDentists.Items.Clear()
    cboPatients.Items.Clear()
    cboAppointmentList.Items.Clear()
    'Appointment date set to today by default.'
    AppointmentDate.Text = Date.Today
    txtTime.Clear()
    txtTreatmentDuration.Clear()

    'This statement will load all dentists into the combo box for the dentists ID
    and first names.'
    Sql = "SELECT * FROM tblDentists"
    da = New SqlDataAdapter(Sql, dbconn)
```

```
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblDentists").Clear()

Catch ex As Exception

End Try
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblDentistsName").Clear()
Catch ex As Exception

End Try
da.Fill(ds, "tblDentistsName")
CountRows = ds.Tables("tblDentistsName").Rows.Count
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1

cboDentistID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))

cboDentists.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))
Next

    'This SQL statement will load all the patient's ID and first name into the
    combo boxes.'
    Sql = "SELECT * FROM tblPatients"
    da = New SqlDataAdapter(Sql, dbconn)

    Try
        'Dataset is cleared first before retrieving the updated list.'
        ds.Tables("tblPatients").Clear()
    Catch ex As Exception

    End Try

    Try
        'Dataset is cleared first before retrieving the updated list.'
        ds.Tables("tblPatientName").Clear()
    Catch ex As Exception

    End Try

    da.Fill(ds, "tblPatientName")
    CountRows = ds.Tables("tblPatientName").Rows.Count
    For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1

cboPatientID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))

cboPatients.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next

    'This SQL statement will load all the booking appointment ID's into the combo
    box.'
    Sql = "SELECT * FROM tblBooking"
    da = New SqlDataAdapter(Sql, dbconn)
    Try
        'Dataset is cleared first before retrieving the updated list.'
        ds.Tables("tblBooking").Clear()

    Catch ex As Exception

    End Try
```

```
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblNewBooking").Clear()
Catch ex As Exception

End Try

da.Fill(ds, "tblNewBooking")
CountRows = ds.Tables("tblNewBooking").Rows.Count
For i = 0 To ds.Tables("tblNewBooking").Rows.Count - 1

cboAppointmentList.Items.Add(ds.Tables("tblNewBooking").Rows(i).Item("AppointmentID"))
Next

End Sub

Private Sub btnDeleteAppointment_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeleteAppointment.Click
    'Sets the appointment as an integer.'
    Dim SelectedAppointment As Integer
    'Sets the combo box as an integer.'
    SelectedAppointment = cboAppointmentList.Text

Try
    'Clears the dataset.'
    ds.Tables("tblBooking").Clear()
Catch ex As Exception

End Try

'Selects the whole row where the appointment ID matches the combo box.'
Sql = "DELETE FROM tblBooking WHERE AppointmentID = " & SelectedAppointment &
""

da = New SqlDataAdapter(Sql, dbconn)
'Record is deleted.'
da.Fill(ds, "tblBooking")

MessageBox.Show("Record deleted")

'This variable is used to count the rows in a dataset.'
Dim CountRows As Integer
'This variable is used loop until I can find the row I'm looking for.'
Dim i As Integer
'Combobox for ID's are disabled.'
cboTreatmentID.Enabled = False
lblRoomID.Visible = False
'combo boxes are set to blank.'
cboAppointmentList.SelectedIndex = -1
cboDentists.SelectedIndex = -1
cboPatients.SelectedIndex = -1
cboTreatments.SelectedIndex = -1
'Add, delete, edit and save buttons are disabled.'
btnAddAppointment.Enabled = False
btnDeleteAppointment.Enabled = False
btnEdit.Enabled = False
btnSave.Enabled = False
'Appointment lists are enabled.'
cboAppointmentList.Enabled = True
```

```
'The show, cancel and register buttons are enabled.'  
btnShow.Enabled = True  
btnCancel.Enabled = True  
btnRegister.Enabled = True  
'All other combo boxes are disabled.'  
cboDentistID.Enabled = False  
cboPatientID.Enabled = False  
cboPatients.Enabled = False  
cboDentists.Enabled = False  
cboTreatments.Enabled = False  
'Text fields are set to read only to prevent editing.'  
txtTime.ReadOnly = True  
txtTreatmentDuration.ReadOnly = True  
AppointmentDate.Enabled = False  
'All radio buttons are disabled and unchecked.'  
rdoRoom1.Enabled = False  
rdoRoom2.Enabled = False  
rdoRoom3.Enabled = False  
rdoRoom4.Enabled = False  
rdoRoom1.Checked = False  
rdoRoom2.Checked = False  
rdoRoom3.Checked = False  
rdoRoom4.Checked = False  
  
'All combo boxes and text fields are cleared.'  
cboDentistID.Items.Clear()  
cboPatientID.Items.Clear()  
cboDentists.Items.Clear()  
cboPatients.Items.Clear()  
cboAppointmentList.Items.Clear()  
'Appointment date set to today by default.'  
AppointmentDate.Text = Date.Today  
txtTime.Clear()  
txtTreatmentDuration.Clear()  
  
'This statement will load all dentists into the combo box for the dentists ID  
and first names.'  
Sql = "SELECT * FROM tblDentists"  
da = New SqlDataAdapter(Sql, dbconn)  
  
Try  
    'Dataset is cleared first before retrieving the updated list.'  
    ds.Tables("tblDentists").Clear()  
  
Catch ex As Exception  
  
End Try  
Try  
    'Dataset is cleared first before retrieving the updated list.'  
    ds.Tables("tblDentistsName").Clear()  
  
Catch ex As Exception  
  
End Try  
da.Fill(ds, "tblDentistsName")  
CountRows = ds.Tables("tblDentistsName").Rows.Count  
For i = 0 To ds.Tables("tblDentistsName").Rows.Count - 1  
  
cboDentistID.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DentistID"))  
  
cboDentists.Items.Add(ds.Tables("tblDentistsName").Rows(i).Item("DFirstName"))  
Next
```

```
'This SQL statement will load all the patient's ID and first name into the
combo boxes.'
Sql = "SELECT * FROM tblPatients"
da = New SqlDataAdapter(Sql, dbconn)

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblPatients").Clear()
Catch ex As Exception

End Try

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblPatientName").Clear()
Catch ex As Exception

End Try

da.Fill(ds, "tblPatientName")
CountRows = ds.Tables("tblPatientName").Rows.Count
For i = 0 To ds.Tables("tblPatientName").Rows.Count - 1

cboPatientID.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("PatientID"))
cboPatients.Items.Add(ds.Tables("tblPatientName").Rows(i).Item("FirstName"))
Next

'This SQL statement will load all the booking appointment ID's into the combo
box.'
Sql = "SELECT * FROM tblBooking"
da = New SqlDataAdapter(Sql, dbconn)
Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblBooking").Clear()

Catch ex As Exception

End Try

Try
    'Dataset is cleared first before retrieving the updated list.'
    ds.Tables("tblNewBooking").Clear()
Catch ex As Exception

End Try

da.Fill(ds, "tblNewBooking")
CountRows = ds.Tables("tblNewBooking").Rows.Count
For i = 0 To ds.Tables("tblNewBooking").Rows.Count - 1

cboAppointmentList.Items.Add(ds.Tables("tblNewBooking").Rows(i).Item("AppointmentID"))
Next
End Sub
End Class
```