Imports System

Imports BAL

Imports DAL

Imports System.Drawing.Imaging

Imports System.String

Imports System.Printing.PrintQueue

Imports PrinterQueueWatch

Interface IDisplayStudentCalendar

    Function populateCalendar(ByVal selectedmonth As Integer, ByVal selectedyear As Integer)

End Interface

Public Class StudentCalendar

    Implements IDisplayStudentCalendar

    Public currentmonth As Integer

    Public currentyear As Integer

    Public Finalmonth As Integer

    Public FinalYear As Integer

    Dim selectedfolder As String = String.Empty

    Private Property noterow As Integer = -1

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

        REM Populate combobox control with active student name

        Dim names As IPopulateAllNames = New IPopulateNames

        Dim dsStudent As New DataSet

        dsStudent = names.DisplayStudents(False)

        Dim dtStudent As DataTable = dsStudent.Tables("ScheduleFooter")

        ComboBox1.DataSource = dtStudent

        ComboBox1.DisplayMember = "FullName"

        ComboBox1.ValueMember = "FullName"

        ComboBox1.SelectedIndex = 1

        Dim x As Integer

        REM Format DateTimePicker controls

        DateTimePicker2.Value = DateTimePicker1.Value

        DateTimePicker1.CustomFormat = "MMMM, yyyy"

        DateTimePicker2.CustomFormat = "MMMM, yyyy"

        REM Setup Initial display in the Datagrid

        For x = 0 To 4

            Dim dgvRow As New DataGridViewRow

            Dim dgvCell As DataGridViewCell

            dgvCell = New DataGridViewTextBoxCell()

            dgvCell.Value = String.Empty

            dgvRow.Cells.Add(dgvCell)

            DataGridView1.Rows.Add(dgvRow)

            DataGridView1.Rows(x).Cells(0).Style.BackColor = Color.LightGray

        Next

        REM format the size of the Datagrid

        DataGridView1.Height = 550

        Dim dr As DataGridViewRow

        For Each dr In DataGridView1.Rows

            dr.Height = 100

        Next

        Me.ComboBox1.Focus()

    End Sub

    'Setup or Resets the Student Calendar Display for the currently selected month

    Public Sub numberCalendar(ByVal tempdate As String)

        Dim carriage As Integer=0

        Dim date1 As DateTime

        Dim firstDay As String = String.Empty

        Dim newyear As String = String.Empty

        Dim monthnumber As Integer

        Dim currentyear As Integer

        'Convert from a string to a date value in order to figure out the first day, month,and year.

        date1 = Convert.ToDateTime(tempdate)

        firstDay = date1.ToString("dddd")

        newyear = date1.ToString("yyyy")

        monthnumber = date1.Month

        currentyear = Convert.ToInt16(newyear)

        Dim x As Integer = 1

        Dim numberofdays As Integer

        'Return the number of days in the current month

        numberofdays = DateTime.DaysInMonth(currentyear, monthnumber)

        Dim w As Integer = DataGridView1.Rows.Count

        'Remove all the row inthe GridView.

        If DataGridView1.Rows.Count > 0 Then

            For b = 0 To DataGridView1.RowCount - 1

                DataGridView1.Rows.RemoveAt(0)

            Next

        End If

        Dim headername As String = String.Empty

        'Setup each cell in the datagrid and place the date inside each cell

        Do While x < numberofdays

            Dim dgvRow As New DataGridViewRow

            dgvRow.Height = 110

            'Go thorugh each column (Sunday thru Saturday)

            For y = 0 To 6

                'Get thru each headertext (Each day of the week)

                headername = DataGridView1.Columns(y).HeaderText

                Dim dgvCell As DataGridViewCell

                dgvCell = New DataGridViewTextBoxCell()

                'Compare it to the firstday of the month

                If x = 1 And firstDay = headername Then

                    'If firstday match the header name then place the first date into the cell

                    dgvRow.Cells.Add(dgvCell)

                    DataGridView1.DefaultCellStyle.WrapMode = DataGridViewTriState.True

                    dgvCell.Value = x & ControlChars.CrLf & ControlChars.CrLf & ControlChars.CrLf & ControlChars.CrLf

                    DataGridView1.Columns(y).DefaultCellStyle.Font = New Font("Times NewRoman", 10, FontStyle.Regular)

                    x = x + 1

                ElseIf x > 1 Then

                    'Place the current value of x (The current date value) in the cell

                    'If the end of the month is reached then terminate out of the loop

                    dgvRow.Cells.Add(dgvCell)

                    DataGridView1.DefaultCellStyle.WrapMode = DataGridViewTriState.True

                    DataGridView1.Columns(y).DefaultCellStyle.Font = New Font("Times NewRoman", 10, FontStyle.Regular)

                    dgvCell.Value = x & ControlChars.CrLf & ControlChars.CrLf & ControlChars.CrLf & ControlChars.CrLf

                    x = x + 1

                    If x > numberofdays Then Exit For

                Else

                    'Place nothing in the cell until the first date in the month is reached.

                    dgvCell.Value = String.Empty

                    dgvRow.Cells.Add(dgvCell)

                    DataGridView1.DefaultCellStyle.WrapMode = DataGridViewTriState.True

                End If

            Next

            DataGridView1.Rows.Add(dgvRow)

            'Change the Height of the gridview based on the number of weeks in it.

            'Clolor the first column of Sunday, the Total Hours, and Cummulative Hours to the color Gray

            If DataGridView1.Rows.Count < 6 Then

                DataGridView1.Height = 600

                Me.Height = 780

            Else

                'Color the 5th row gray if the height of the gridview is increased

                DataGridView1.Height = 700

                Me.Height = 850

                DataGridView1.Rows(5).Cells(0).Style.BackColor = Color.LightGray

            End If

        Loop

        For x = 0 To 4

            DataGridView1.Rows(x).Cells(0).Style.BackColor = Color.LightGray

        Next

    End Sub

    Private Sub DateTimePicker1\_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DateTimePicker1.ValueChanged

        DisplayStudentCalendar()

    End Sub

    Public Function populateCalendar(ByVal selectedmonth As Integer, ByVal selectedyear As Integer) Implements IDisplayStudentCalendar.populateCalendar

        Dim findScheduledDates As IScheduledHour = New displayProposedDate

        Dim studentnameConversion As INameConversion = New StudentNameconversion

        Dim studentinformation As New StudentCalendar

        Dim Convertname As New Schedule

        Dim x As Integer

        Dim StudentSchedule As New ScheduleConfig

        Dim ds5 As New DataSet

        Dim proposeds As New DataSet

        Dim studentname As String = String.Empty

        Dim currentcalendardate As DateTime

        Dim date3 As Integer = DateTime.DaysInMonth(selectedyear, selectedmonth)

        Dim mergedate1 As String = selectedmonth & "/1/" & selectedyear

        Dim mergedate2 As String = selectedmonth & "/" & date3 & "/" & selectedyear

        'Return prior month

        Dim date1 As DateTime = Convert.ToDateTime(mergedate1).AddDays(-1)

        'Return current month

        Dim date2 As DateTime = Convert.ToDateTime(mergedate2).AddDays(1)

        'Determine the students id number

        Dim studentid As String = String.Empty

        studentname = ComboBox1.SelectedValue.ToString

        studentid = studentnameConversion.convertToId(studentname)

        'Get Proposed Dates

        Dim currentdate As DateTime = Nothing

        Dim finaldate1 As String = String.Empty

        'Get all datasets that are needed in this routine

        ds5 = Convertname.ReturnStudentScheduleinfo(studentid.Trim)

        Dim proposedt As DataTable = ds5.Tables("MainSchedule")

        Dim search2 As Integer = 0

        Dim startindex As Integer = 0

        Dim query2 As String = String.Empty

        Dim Maxrows As Integer = 0

        Maxrows = DataGridView1.Rows.Count

        Dim displaytime As String = String.Empty

        Dim displayhour As String = String.Empty

        For x = 0 To Maxrows - 1 'Rows (Week)

            For y = 0 To 6 'Go through each column(each day within a week)

                displaytime = String.Empty

                Dim dy, getday As String

                displaytime = [String].Empty

                'Return the date that is displayed in  the current cell'

                getday = DataGridView1.Rows(x).Cells(y).Value

                'Check to see if the current cell is a valid date within the current month

                If getday = String.Empty Then

                    currentcalendardate = Nothing

                    'Skip entire iteration and go to the cell

                Else

                    'Determine the day

                    dy = getday

                    'Setup the date in a string format of MM/dd/yyyy

                    Dim edate As String = selectedmonth & "/" & dy.Trim & "/" & selectedyear

                    'Get fulldate of the current month

                    currentcalendardate = Convert.ToDateTime(edate.Trim)

                    findScheduledDates.displayStudentCalendar(proposedt, currentcalendardate, displaytime, x, y)

                End If

                'Display any proposed dates

                'Return all proposed dates that are within the current calendar date

            Next

        Next

        Return Nothing

    End Function

    REM displays the student Calendar

    Public Function DisplayStudentCalendar()

        Dim tempmonth As DateTime

        Dim tempyear As String = String.Empty

        Dim tempdate As String = String.Empty

        Dim monthval As String = String.Empty

        Try

            Dim selectname As String = ComboBox1.SelectedValue.ToString

            If DateTimePicker1.Value = Nothing Then

                MsgBox("Pick a Date!")

                Timer1.Enabled = False

                Timer2.Enabled = False

                Timer3.Enabled = False

                Return Nothing

                Exit Function

            End If

            'Capture dates and convert them in a string date format

            tempmonth = DateTimePicker1.Value

            monthval = DateTimePicker1.Value.ToString("MMMM")

            tempyear = DateTimePicker1.Value.ToString("yyyy")

            Dim year1 As Integer = Convert.ToInt16(tempyear)

            Dim month1 As Integer = tempmonth.Month

            tempdate = month1 & "/1/" & tempyear

            DateTimePicker1.Value = Convert.ToDateTime(tempdate)

            DateTimePicker2.Value = Convert.ToDateTime(tempdate)

            If selectname <> String.Empty Then

                'Format the selected date as a string and pass them to hteir resppected methods

                Label6.Text = tempyear

                Label1.Text = monthval

                'Format Display as a Calenday of the selected Month

                numberCalendar(tempdate.Trim)

                'Place student schedule onto the Calendar Display Grid

                populateCalendar(month1, year1)

            Else

                'Do the following if user does not select a name

                DateTimePicker1.Refresh()

                DateTimePicker1.Value = tempdate

                DateTimePicker2.Value = DateTimePicker1.Value

                Timer1.Enabled = False

                Timer2.Enabled = False

                Timer3.Enabled = False

            End If

        Catch ex As Exception

            DateTimePicker1.Value = Convert.ToDateTime(tempdate)

            DateTimePicker2.Value = DateTimePicker1.Value

            Timer1.Enabled = False

            Timer2.Enabled = False

            Timer3.Enabled = False

            Return Nothing

            Exit Function

        End Try

        Timer3.Enabled = False

        Return Nothing

    End Function

    REM Display a students calendar over a range of more than one months

    Public Function DisplayMultipleCalendars()

        Try

            Dim tempmonth As DateTime

            Dim tempmonth2 As DateTime

            Dim tempyear As String = String.Empty

            Dim tempyear2 As String = String.Empty

            Dim tempdate As String = String.Empty

            Dim tempdate2 As String = String.Empty

            Dim monthval As String = String.Empty

            Dim monthval2 As String = String.Empty

            If ComboBox1.SelectedValue.ToString = String.Empty Then

                Return Nothing

                Exit Function

            End If

            Dim Studentname As String = String.Empty

            Studentname = Me.ComboBox1.SelectedValue.ToString

            Dim CalendarMonth1 As String = String.Empty

            Dim CalendarMonth2 As String = String.Empty

            tempmonth = Me.DateTimePicker1.Value

            tempmonth2 = Me.DateTimePicker2.Value

            monthval = Me.DateTimePicker1.Value.ToString("MMMM")

            tempyear = Me.DateTimePicker1.Value.ToString("yyyy")

            monthval2 = Me.DateTimePicker2.Value.ToString("MMMM")

            tempyear2 = Me.DateTimePicker2.Value.ToString("yyyy")

            Dim year1 As Integer = Convert.ToInt16(tempyear)

            Dim month1 As Integer = tempmonth.Month

            Dim StopMonth As Integer = tempmonth2.Month

            tempdate = month1 & "/1/" & tempyear

            tempdate2 = StopMonth & "/1/" & tempyear2

            Dim Month2 As Date = Me.DateTimePicker2.Value

            CalendarMonth1 = Me.DateTimePicker1.Value

            REM If start date and finish date are equal then print the current screen without activating any timer.

            REM Print only what is displayed. There is no itrastion through the various months because the selected start and finish dates is

            REM The current month selected

            If tempdate = tempdate2 Then

                currentmonth = month1

                 CaptureScreen()

                Timer3.Enabled = False

                MsgBox("The Student Calendar has been saved to " & selectedfolder)

                GroupBox1.Size = New Point(390, 90)

                Return Nothing

                Exit Function

            End If

            numberCalendar(tempdate.Trim)

            Label1.Text = MonthName(month1)

            Label6.Text = year1.ToString()

            If Not CalendarMonth1 = String.Empty And Not Studentname = String.Empty Then

                GroupBox1.Size = New Point(205, 38)

                Me.Label1.Text = monthval

                Me.Label6.Text = tempyear

                Me.populateCalendar(month1, year1)

                currentmonth = month1

                Finalmonth = StopMonth

                FinalYear = tempmonth2.Year

                currentyear = year1

                Timer1.Enabled = True

                Timer2.Enabled = True

                DateTimePicker1.Visible = False

                DateTimePicker2.Visible = False

                Button1.Visible = False

                Label4.Visible = False

                Label5.Visible = False

            Else

                MsgBox("You forgot to choose a name or Date")

            End If

        Catch ex As Exception

        End Try

        Me.Visible = True

        Timer3.Enabled = False

        Return Nothing

    End Function

    REM increase by one month

    REM Display the month

    REM Turn on timer2 to capture image of the month and store it as a file

    Private Sub Timer1\_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.Tick

        Dim tempdate As String

        currentmonth = currentmonth + 1

        REM chjeck for month

        If currentmonth > 12 Then

            currentmonth = 1

            currentyear = currentyear + 1

        End If

        REM check for year

        If currentyear > FinalYear Then

            Timer1.Enabled = False

            Timer2.Enabled = False

            Exit Sub

        End If

        REM compare the month and year to make sure that if the currentmonth goes through a year that the year ism reset

        If (currentmonth <= Finalmonth And currentyear = FinalYear) Or (currentmonth > Finalmonth And currentyear < FinalYear) Then

            GroupBox1.Size = New Point(205, 38)

            tempdate = currentmonth & "/1/" & currentyear

            numberCalendar(tempdate.Trim)

            Label1.Text = MonthName(currentmonth)

            Label6.Text = currentyear.ToString()

            Me.populateCalendar(currentmonth, currentyear)

            Timer1.Enabled = False

            Timer2.Enabled = True

        ElseIf currentmonth > Finalmonth And currentyear = FinalYear Then

REM Disable all timers and redisplay and resize the controls at the top of the form

            Timer1.Enabled = False

            Timer2.Enabled = False

            Timer3.Enabled = False

            DateTimePicker1.Visible = True

            DateTimePicker2.Visible = True

            Button1.Visible = True

            Label4.Visible = True

            Label5.Visible = True

            MsgBox("The Student Calendar has been saved to " & selectedfolder)

            GroupBox1.Size = New Point(390, 90)

        End If

    End Sub

    Private Sub Button1\_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

        DisplayStudentCalendar()

        displayContactProfile()

    End Sub

    Public Function displayContactProfile()

        Dim contactInfo As New ArrayList

        If Me.ComboBox1.Focus() Then

            Dim convertname As INameConversion = New StudentNameconversion

            Dim studentId As String = String.Empty

            Dim studentname As String = String.Empty

            Dim studentContactinfo As IlUsersProfileData = New returnUserProfile

            studentname = ComboBox1.SelectedValue.ToString

            studentId = convertname.convertToId(studentname)

            contactInfo = studentContactinfo.guardian(studentId)

            If contactInfo.Count > 0 Then

                Label2.Text = contactInfo(1) 'Email

                Label3.Text = contactInfo(2) 'Alt. Email

                Label7.Text = contactInfo(3) 'Home Phone

                Label8.Text = contactInfo(4) 'Cell Phone

                Label9.Text = contactInfo(5) 'Work phone

            End If

        End If

        Return Nothing

    End Function

    Private Sub ComboBox1\_SelectedIndexChanged(sender As Object, e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

        Label2.Text = String.Empty

        Label3.Text = String.Empty

        Label7.Text = String.Empty

        Label8.Text = String.Empty

        Label9.Text = String.Empty

        displayContactProfile()

    End Sub

    Private Sub Button3\_Click\_1(sender As System.Object, e As System.EventArgs) Handles Button3.Click

    End Sub

   Capture an image of the current display

    Public Sub CaptureScreen()

        Try

            GroupBox1.Size = New Point(205, 38)

            Dim monthval As String

            monthval = MonthName(currentmonth)

            Dim studentname As String

            studentname = ComboBox1.SelectedValue.ToString

            Dim screenGrab As New Bitmap(Me.Width, Me.Bounds.Height, PixelFormat.Format32bppArgb)

            Dim g As System.Drawing.Graphics = System.Drawing.Graphics.FromImage(screenGrab)

            g.CopyFromScreen(Me.Bounds.X, Me.Bounds.Y, 0, 0, Me.Bounds.Size, CopyPixelOperation.SourceCopy)

            Dim filename As String = selectedfolder & "\" & studentname.Trim() & "\_" & monthval & ".jpg"

            screenGrab.Save(filename, System.Drawing.Imaging.ImageFormat.Jpeg)

        Catch ex As Exception

            MsgBox("You do not have permission to write to this directory.")

            Timer1.Enabled = False

            Timer2.Enabled = False

            Timer3.Enabled = False

        End Try

    End Sub

1. Take a screen shot of the current Calendar being displayed

    2) Turn off Screen capture

    3) Move to the next month

    Private Sub Timer2\_Tick(sender As System.Object, e As System.EventArgs) Handles Timer2.Tick

        CaptureScreen()

        Timer2.Enabled = False

        Timer1.Enabled = True

    End Sub

1. Acquires the file path.
2. Updates that path to the data source.
3. Make a call to alter the display

Private Sub ToolStripButton2\_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripButton2.Click

        If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then

            Dim savePath As IfileStoragePath = New saveStorageModules

            selectedfolder = FolderBrowserDialog1.SelectedPath

            savePath.path(selectedfolder, "student")

            HideControls()

        ElseIf FolderBrowserDialog1.ShowDialog() = DialogResult.Cancel Then

            Exit Sub

        End If

    End Sub

Acquires the selected file location and stores the path into the data source.

Private Sub ToolStripButton1\_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripButton1.Click

        Dim getStoredPath As IfileStoragePath = New returnStorageModules

        Dim savePath As IfileStoragePath = New saveStorageModules

        selectedfolder = getStoredPath.path(selectedfolder, "student")

        If selectedfolder <> String.Empty Then

            savePath.path(selectedfolder, "student")

        End If

        HideControls()

        If ComboBox1.SelectedValue.ToString = String.Empty Then

            Exit Sub

        End If

        If DateTimePicker2.Value < DateTimePicker1.Value Then

            MsgBox("Final date cannot be set before the Start date!")

            Exit Sub

        End If

    End Sub

Initiate the alteration of the display

    Public Sub HideControls()

        'Hide the display controls

        GroupBox1.Size = New Point(205, 38)

        If DateTimePicker1.Value = DateTimePicker2.Value Then

            DisplayStudentCalendar()

            GroupBox1.Size = New Point(205, 38)

            Timer3.Enabled = True

            GroupBox1.Size = New Point(400, 90)

        Else

            DisplayMultipleCalendars()

        End If

    End Sub

Private Sub Timer3\_Tick(sender As System.Object, e As System.EventArgs) Handles Timer3.Tick

        DisplayMultipleCalendars()

    End Sub

End Class

