Print/Save Office Schedule (PrintScheduleForms.vb)

Imports BAL

Imports DAL

Imports System

Imports System.Windows

Imports System.Drawing.Imaging

Imports System.Printing.PrintQueue

Imports PrinterQueueWatch

Public Class PrintScheduleForms

'Necessary declaration for timer 2

Dim dateholder As Date

Dim newday As Date

Dim dys As Integer

'Necessary declaration for timer3

Dim selectedFolder As String = String.Empty

'Initial Form Load

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

'Label the first row each column with active Clincians

Dim clinicianinfo As New Clinicians

Dim dsclinicians As New DataSet

'Return all active clincians

dsclinicians = clinicianinfo.GetClinicianInfo(True)

Dim dtClinicians As DataTable = dsclinicians.Tables("Clinician")

'Place all the clinician that were return into the DataGrid,

'The columns will grow according to the number of clincians that are returned

Dim row As DataRow

For Each row In dtClinicians.Rows

Dim ClinicianName As String = row("LastName").Trim & ", " & row("FirstName").Trim

DataGridView1.Columns.Add("Column1", ClinicianName)

Next

'Setup initial display display with the date from the Home Display Form

DateTimePicker1.Value = HomeDisplay.MonthCalendar1.SelectionStart

DateTimePicker2.Value = HomeDisplay.MonthCalendar1.SelectionStart

'Display the selected day from the home screen form display

newday = Convert.ToDateTime(DateTimePicker1.Value)

'Global variable need to be set to the selected date

dateholder = newday

'Execute the display timer only

Timer1.Enabled = True

End Sub

'Setup the DataGrid Display Layout

'Return Dataset of all scheduled students and clinicians

'Return all scheduled students on the given date

'Go through datset and return each active clinician

'Go through dataset and return each active clinician

'Go through each header title in the DataGrid , and match it with

'the clinician that is returned from the database.

'If there is a match, iterate through each row in the column of the clinician

'Iterate through all 24 rows on every column(Clinician)

'Check to see if Clinician is scheduled Out at that Timeslot'

'Or Check to see if there is a student name at that Time slot.

'Check to see if cell is the start of the clinician being scheduled out

'This is clinicianout variable increment it to see if the clinician is out the entire day

'Iterate through each student whos starttime and endtime match within the current time range of the Datagrid Control.

'Return Student information to be displayed

'Check Proposed hours that are hour 1 or 2

'Check rescheduled hours that are hour 1 or 2

'Color the cell accordingly to their status or Attendance value

'Check to see if the first two hours exsist. If so then color the cell white.

'Check for the first two hours of the current student. If the first two hours are found the flag it inthevariable called start hour

'Clear variables and color time slot as WhiteSmoke if there is nothing at that specific time slot

'0 Marks the Clinician as not being out for the entire day therefor that column will not be hidden whne printed

'Check to see if student is on the start of an hour.

'Check to see if the current cell is different from the last cell

'if so then Display the location of the student

'Clear Location variable so it will not be displayed in other cells during the iteration process

'Store the student name so the current student name in the iteration only appears once.

'Check to see if clinicianout has reached the 23 row

'If Clinician is out the entire day remove the entire clinician column,

'otherwise if the clinician out only part of the day or not at all then

'keep column visible.

Public Function DisplaySetup(ByVal CurrentDate As Date) As Integer

Try

Dim Clinicianout As Integer = 0

Dim Student As String = String.Empty

Dim Display As New ScheduleConfig

Dim Stat As New Schedule

Dim ds2 As New DataSet

Dim ds3 As New DataSet

Dim ds5 As New DataSet

Dim ds As New DataSet

'Setup the DataGrid Display Layout

DisplayTemplate()

'Return Dataset of all scheduled students and clinicians

ds = Display.SetupScheduleColRows(CurrentDate)

Dim dt As DataTable = ds.Tables("Schedule1")

Dim splitname\_student() As String = Nothing

Dim studentid As String = String.Empty

Dim nextname As String = String.Empty

Dim countid As String = String.Empty

Dim subject As String = String.Empty

Dim Location As String = String.Empty

'Return all scheduled students on the given date

ds2 = Stat.GetSchedule(CurrentDate, CurrentDate)

Dim dt2 As DataTable = ds2.Tables("MainSchedule")

Dim b As Integer

Dim Status As String = String.Empty

Dim Clinician As String = String.Empty

Dim ClinicianHeader As String = String.Empty

Dim present As String = String.Empty

' Go through and return each active clinician

For Each rw In dt.Rows

Clinician = rw("Clinician")

Dim numberofColumns As Integer = DataGridView1.ColumnCount

Dim headercount As Integer = 0

'Go through each header title in the GridView control, and match it with

'the clinician that is returned from the database.

'If there is a match, iterate through each row in the column

For headercount = 1 To numberofColumns

ClinicianHeader = DataGridView1.Columns(headercount).HeaderText

If ClinicianHeader = Clinician Then

Dim a As Integer = 0

Dim timein As String = String.Empty

Dim t1 As DateTime = Nothing

Dim query As String = String.Empty

Dim c As Integer = 0

'Iterate through all 24 rows on every column(Clinician)

For a = 1 To 24

timein = DataGridView1.Rows(a).Cells(0).Value

t1 = Convert.ToDateTime(timein.Trim).ToShortTimeString

Student = rw(a).ToString

'Check to see if Clinician is out'

'Or Check to see if there is a student name.

If Student = " " & "OUT" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Green

Location = String.Empty

'Check to see if cell is the start of the clinician being out

If c = 0 Then

DataGridView1.Rows(a).Cells(headercount).Value = " " & "OUT"

DataGridView1.Columns(headercount).DefaultCellStyle.Font = New Font("Times NewRoman", 8, FontStyle.Regular)

End If

c = c + 1

'This is increment to see if the clinician is out the entire day

Clinicianout = Clinicianout + 1

ElseIf Student <> String.Empty Then

splitname\_student = Student.Split(",")

studentid = Stat.ReturnStudentInfo(splitname\_student(1).Trim, splitname\_student(0).Trim)

Dim timestamp As String = Nothing

timestamp = Convert.ToDateTime("1900-01-01 " & timein)

Dim time1 As DateTime

Dim time2 As DateTime

'Iterate through each student whos starttime and endtime match within the current time range of the Datagrid Control.

query = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

Dim foundrow() As DataRow = dt2.Select(query)

'Return Student information to be displayed

b = 0

Do While b <= foundrow.Length - 1

Status = foundrow(b)("status")

countid = foundrow(b)("Count")

present = foundrow(b)("Attendance")

ds5 = Stat.GetClassroomData(countid.Trim)

Dim dt5 As DataTable = ds5.Tables("Classroom")

Dim subjectrow As DataRow

For Each subjectrow In dt5.Rows

subject = subjectrow("Subject")

Location = subjectrow("Campus")

Next

'Check for the first two hours of the current student. If the first two hours are found the flag it inthevariable called start hour

Dim starthr As String = String.Empty

Dim gethr As String = String.Empty

'Check Proposed(not rescheduled) hours that are hour 1 or 2

Dim dt3 As DataTable

Dim query2 = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

ds3 = Stat.ReturnBillingDetailsinfo(studentid, CurrentDate, CurrentDate)

dt3 = ds3.Tables("BillingDetails")

Dim foundrow\_starthour() As DataRow = dt3.Select(query2)

Dim d As Integer = 0

Do While d <= foundrow\_starthour.Length - 1

gethr = foundrow\_starthour(d)("HourNoIn")

If gethr = 1 Or gethr = 2 Then

starthr = gethr

End If

d = d + 1

Loop

'Check rescheduled hours that are hour 1 or 2

Dim transferquery = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

Dim dt4 As DataTable

dt4 = Stat.ReturnStudentTransfer(studentid.Trim, CurrentDate)

Dim foundrow\_transferstarthour() As DataRow = dt4.Select(transferquery)

Dim e As Integer = 0

Do While e <= foundrow\_transferstarthour.Length - 1

gethr = foundrow\_transferstarthour(e)("TransferHrIn")

If gethr = 1 Or gethr = 2 Then

starthr = gethr

End If

e = e + 1

Loop

'Color the cell accordingly to their status and Attendance

If present.Trim = "Absent" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Red

ElseIf present.Trim = "Completed" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Blue

ElseIf Status.Trim = "Transfer" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Yellow

ElseIf Status.Trim = "Proposed" And subject.Trim = "Testing" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Gray

ElseIf Status.Trim = "Proposed" And subject.Trim = "Meeting" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Orange

ElseIf present.Trim = "Proposed" And (timestamp >= time1 Or timestamp <= time2) Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Yellow

End If

'Check to see if the first two hours are triggered. If so then color the cell white.

If starthr = "1" Or starthr = "2" Then

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.White

starthr = String.Empty

End If

b = b + 1

Loop

ElseIf Student = String.Empty Then

'Clear variables and color time slot as empty

studentid = String.Empty

Location = String.Empty

DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.AntiqueWhite

'0 Marks the Clinician as not being out for the entire day

Clinicianout = 0

Else

End If

'Check to see if student is on the start of an hour

'Check to see if the current cell is different from the last cell

'if so then Display the location of the student

If nextname <> Student Or nextname = Student And b = 2 Then

If Location.Trim = "NorthWest" Then

Location = "NW="

ElseIf Location.Trim = "Main" Then

Location = "M="

Else

'Clear Location variable so it will not be displayed in other cells

Location = String.Empty

End If

REM Concatenate the Location with the student name

DataGridView1.Rows(a).Cells(headercount).Value = Location & Student

DataGridView1.Columns(headercount).DefaultCellStyle.Font = New Font("Times NewRoman", 8, FontStyle.Regular)

REM store the student name so the current student in the iteration only appears once.

nextname = Student

Clinicianout = 0

Else

End If

Next

Exit For

Else

End If

Next

'Check to see if clinicianout has reached the 23 row

'If Clinician is out the entire day remove the entire clinician column,

'otherwise if the clinician out only part of the day or not at all then

'keep column

If Clinicianout > 22 Then

DataGridView1.Columns(headercount).Visible = False

ElseIf Clinicianout < 23 Then

Clinicianout = 0

DataGridView1.Columns(headercount).Visible = True

End If

Next

Catch ex As Exception

End Try

Return DataGridView1.Rows.Count - 1

End Function

'Setup and display all 25 rows labeled with a time slots in the first column of the Datagrid 7:30 AM to 7:30 PM

Public Sub DisplayTemplate()

DataGridView1.Columns(0).DefaultCellStyle.Font = New Font("Times NewRoman", 10, FontStyle.Regular)

Dim i As Integer = 0

Dim timestamp As Array = {"", "7:30 AM", "8:00 AM", "8:30 AM", "9:00 AM", "9:30 AM",

"10:00 AM", "10:30 AM", "11:00 AM", "11:30 AM", "12:00 PM", "12:30 PM",

"1:00 PM", "1:30 PM", "2:00 PM", "2:30 PM", "3:00 PM", "3:30 PM",

"4:00 PM", "4:30 PM", "5:00 PM", "5:30 PM", "6:00 PM",

"6:30 PM", "7:00 PM", "7:30 PM"}

'Create each row and place the hour label into column 0 and color it with a whitesmoke

For i = 0 To 25

Dim dgvRow As New DataGridViewRow

Dim dgvCell As DataGridViewCell

dgvCell = New DataGridViewTextBoxCell()

dgvCell.Value = timestamp(i)

dgvRow.Cells.Add(dgvCell)

DataGridView1.Rows.Add(dgvRow)

DataGridView1.Rows(i).Cells(0).Style.BackColor = Color.WhiteSmoke

Next

End Sub

'Print Button

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

RunDisplay()

End Sub

'Display Print Dialog Control

'Wait for selection and begin timers, Hide the DateTimePicker, and buttons

'controls at the bottom of the form

Public Sub RunDisplay()

'Display Print Dialog Control

GroupBox1.Visible = True

GroupBox2.Visible = True

newday = Convert.ToDateTime(DateTimePicker2.Value)

Dim date1 As Date = DateTimePicker1.Value

'Wait for selection and begin timers, Hide the DateTimePicker, and buttons

'controls at the bottom of the form

If PrintDialog1.ShowDialog = DialogResult.OK Then

dateholder = Convert.ToDateTime(date1)

Timer1.Enabled = True

Timer2.Enabled = True

GroupBox1.Visible = False

GroupBox2.Visible = False

Else

End If

End Sub

'Do a Screen Grab and then

'Define the printer settings

'Print the current Calendar displayed in the DataGrid to the selected Printer.

'Uses VisualBasic PowerPack

Public Sub Printfunction()

Dim selectedprinter As String

For Each PSource As System.Drawing.Printing.PaperSource In PrintForm1.PrinterSettings.PaperSources

If PSource.Kind = Drawing.Printing.PaperSourceKind.Custom Then

PrintForm1.PrinterSettings.DefaultPageSettings.PaperSource = PSource

Exit For

End If

Next

'Print to selected printer

Dim ps As System.Drawing.Printing.PaperSize

For ix As Integer = 0 To PrintDialog1.PrinterSettings.PaperSizes.Count - 1

If PrintDialog1.PrinterSettings.PaperSizes(ix).Kind = Drawing.Printing.PaperKind.Legal Then

ps = PrintForm1.PrinterSettings.PaperSizes(ix)

PrintForm1.PrinterSettings.DefaultPageSettings.PaperSize = ps

End If

Next

'Printer settings from the print dialog box

PrintForm1.PrinterSettings.PrinterName = PrintDialog1.PrinterSettings.PrinterName

PrintForm1.PrinterSettings.DefaultPageSettings.Landscape = True

selectedprinter = PrintDialog1.PrinterSettings.PrinterName

PrintForm1.PrinterSettings.DefaultPageSettings.Margins = New System.Drawing.Printing.Margins(5, 5, 5, 5)

PrintForm1.PrintAction = Drawing.Printing.PrintAction.PrintToPrinter

PrintForm1.Print(Me, PowerPacks.Printing.PrintForm.PrintOption.Scrollable)

'Screen Grab

Dim screenGrab As New Bitmap(Me.Bounds.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 datea As DateTime = Label1.Text.Trim

Dim stringdate As String = datea.ToString("MMMM dd, yyyy")

End Sub

'Go through each day and display the calendar day.

'Remove all rows from the DataGrid

Public Function updateDisplay()

'Iterate through each day

Dim Iterateday As Date = dateholder

If Iterateday < newday Or Iterateday = newday Then

Label1.Text = dateholder.ToString("dddd, M/dd/yyyy")

'Remove all rows from the datagrid

Dim w As Integer = DataGridView1.Rows.Count

If DataGridView1.Rows.Count > 0 Then

For b = 0 To DataGridView1.RowCount - 1

DataGridView1.Rows.RemoveAt(0)

Next

End If

'Go to the next day within the selected date range

Dim rowcount As Integer = 0

rowcount = DisplaySetup(Iterateday)

dateholder = dateholder.AddDays(+1)

Else

Timer1.Enabled = False

Timer2.Enabled = False

GroupBox1.Visible = True

GroupBox2.Visible = True

End If

Return Nothing

End Function

'Exit Screen Button

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Me.Close()

End Sub

'Save Calendar Button

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

storedailycalendar()

End Sub

'Display Save dialog control. To give the user a place to store all calendar(s

Public Sub storedailycalendar()

If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then

selectedFolder = FolderBrowserDialog1.SelectedPath

dateholder = Convert.ToDateTime(DateTimePicker1.Value)

newday = Convert.ToDateTime(DateTimePicker2.Value)

'Hide the Controls at the bottom of the display

GroupBox1.Visible = False

GroupBox2.Visible = False

Timer1.Enabled = True

Timer3.Enabled = True

ElseIf FolderBrowserDialog1.ShowDialog() = DialogResult.Cancel Then

Exit Sub

End If

End Sub

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

updateDisplay()

End Sub

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

Printfunction()

End Sub

'Store calendar at the specified location

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

Dim screenGrab As New Bitmap(Me.Bounds.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 datea As DateTime = Convert.ToDateTime(Label1.Text.Trim)

Dim stringdate As String = datea.ToString("MMMM dd, yyyy")

Dim filename As String = selectedFolder & "\OfficeSchedule\_" & stringdate.Trim & ".jpg"

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

Timer3.Enabled = False

'Go to next day

Timer1.Enabled = True

Timer3.Enabled = True

End Sub

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

DateTimePicker2.Value = DateTimePicker1.Value

'Display the selected day from the home screen form display

newday = Convert.ToDateTime(DateTimePicker1.Value)

'Global variable need to be set to the selected date

dateholder = newday

'Execute the display timer only

Timer1.Enabled = True

End Sub

End Class