Imports BAL

Imports DAL

Imports System.ComponentModel

Public Class SchedulingConsole

    'Load all default values

    'Populate student Names that are active (only)

    'Setup Single date range area

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

        'Display single date range setup (datetimepicker1 only)

        Dim names As IPopulateAllNames = New IPopulateNames

        Dim todaysdate As Date

        todaysdate = Today

        CheckBox3.Checked = False

        DateTimePicker2.Visible = False

        GroupBox2.Visible = False

        Label4.Visible = False

        GroupBox4.Location = New Point(37, 100)

        GroupBox3.Location = New Point(256, 100)

        Label2.Location = New Point(31, 165)

        Label12.Location = New Point(198, 165)

        GroupBox1.Size = New Point(563, 250)

        Button1.Location = New Point(340, 185)

        Button2.Location = New Point(448, 185)

        ComboBox6.Location = New Point(34, 185)

        ComboBox7.Location = New Point(201, 185)

        ComboBox6.SelectedIndex = 0

        ComboBox7.SelectedIndex = 0

       Fill the active Students inside of ComboBox1

        Dim dsStudent As New DataSet

        dsStudent = names.DisplayStudents(False)

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

        ComboBox1.DataSource = dtStudent

        ComboBox1.DisplayMember = "FullName"

        ComboBox1.ValueMember = "FullName"

        MaskedTextBox1.Text = todaysdate.ToString("MM/dd/yyyy")

        REM Fill the Clinician Combobox

        Dim ds2 As New DataSet

        ds2 = names.DisplayClinician(True)

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

        ComboBox2.DataSource = dt2

        ComboBox2.DisplayMember = "clinicianFullName"

        ComboBox2.ValueMember = "clinicianFullName"

        ThismonthDaterange()

        Dim totalDays As Integer

        totalDays = Date.DaysInMonth(Now.Year, Now.Month)

        DateTimePicker1.Value = Today

        DateTimePicker2.Value = Today.AddDays(totalDays)

        ComboBox1.Focus()

    End Sub

Preset the datetimepicker date range of the beginning and last  day of the current month for the Display Console.

    Public Sub ThismonthDaterange()

        Dim month1 As Integer

        Dim year1 As Integer

        month1 = Date.Today.Month

        year1 = Date.Today.Year

        Dim startofthismonthrange As String

        Dim endofthismonthrange As String

        startofthismonthrange = month1 & "/1/" & year1

        Dim lastday As Integer

        lastday = Date.DaysInMonth(year1, month1)

        startofthismonthrange = month1 & "/1/" & year1

        endofthismonthrange = month1 & "/" & lastday & "/" & year1

        DateTimePicker3.Value = startofthismonthrange

        DateTimePicker4.Value = endofthismonthrange

    End Sub

    'Set the date range for next month in the datetimepickers

    Private Function NextmonthDaterange() As ArrayList

        Dim month1 As Integer

        Dim year1 As Integer

        month1 = Date.Today.Month

        year1 = Date.Today.Year

        If month1 = 12 Then

            year1 = year1 + 1

            month1 = 0

        End If

        'Increase month number by one month

        month1 = month1 + 1

        Dim startofnextmonthrange As String

        Dim endofnextmonthrange As String

        startofnextmonthrange = month1 & "/1/" & year1

        Dim lastday As Integer

        lastday = Date.DaysInMonth(year1, month1)

        startofnextmonthrange = month1 & "/1/" & year1

        endofnextmonthrange = month1 & "/" & lastday & "/" & year1

        Dim nextmonthsrange As New ArrayList

        nextmonthsrange.Add(startofnextmonthrange)

        nextmonthsrange.Add(endofnextmonthrange)

        Return Nextmonthsrange

    End Function

    'Display students schedule within select datetimepicker date range

    Private Sub DisplayStudentSchedule()

        Dim convertStudentName As INameConversion = New StudentNameconversion

        Dim schedule As IDisplaySchedule = New ReschedulingDisplay

        Dim studentid As String

        Dim studentname As String = ComboBox1.SelectedValue.ToString

        'If there is no name then exit routine

        If studentname = String.Empty Then

            Exit Sub

        End If

        Dim interpretname As New Schedule

        Dim StudentSchedule As New ScheduleConfig

'Get beginning and final selected dates from the datetimepicker and convert their data types

        Dim Date1 As String = DateTimePicker3.Text

        Dim Date2 As String = DateTimePicker4.Text

        Dim dt1 As DateTime

        Dim dt2 As DateTime

        dt1 = CDate(Date1.Trim)

        dt2 = CDate(Date2.Trim)

        If Date1 <> String.Empty Or Date2 <> String.Empty Then

            'Get student id number

            studentid = convertStudentName.convertToId(studentname)

Clear out the schedule Display from a previously  populated section

            Dim allrows As Integer = DataGridView1.RowCount

            If DataGridView1.RowCount > 0 Then

                For x = 0 To allrows - 1

                    DataGridView1.Rows.RemoveAt(0)

                Next

            End If

  Return dates to be displayed

            Dim dv As DataView

            dv = schedule.mainDisplay(studentid.Trim, dt1, dt2)

            DataGridView1.DataSource = dv

        Else

        End If

Display and set the width of the columns to be displayed in the datagrid from the dataview

        DataGridView1.Sort(DataGridView1.Columns(2), ListSortDirection.Ascending)

        DataGridView1.Columns(1).ReadOnly = True

        DataGridView1.Columns(2).ReadOnly = True

        DataGridView1.Columns(3).ReadOnly = True

        DataGridView1.Columns(4).ReadOnly = True

        DataGridView1.Columns(5).ReadOnly = True

        DataGridView1.Columns(6).ReadOnly = True

        DataGridView1.Columns(7).ReadOnly = True

        DataGridView1.Columns(8).ReadOnly = True

        DataGridView1.Columns(9).ReadOnly = True

        DataGridView1.Columns(0).Width = 30

        DataGridView1.Columns(1).Width = 55

        DataGridView1.Columns(2).Width = 150

        DataGridView1.Columns(3).Width = 60

        DataGridView1.Columns(4).Width = 60

        DataGridView1.Columns(5).Width = 140

        DataGridView1.Columns(6).Width = 60

        DataGridView1.Columns(7).Width = 65

        DataGridView1.Columns(8).Width = 50

        DataGridView1.Columns(9).Width = 130

        DataGridView1.Columns(1).Visible = False

        'Display Color for each date based on status

        GridViewColorcode()

    End Sub

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

         DataEntry()

    End Sub

    Public Function CheckforselectedDays(startdate As String, enddate As String, ByVal dayofweek As Array)

        Dim trigger As Boolean = False

        Dim d1 As Date

        Dim d2 As Date

        Dim theday As String

        d1 = Convert.ToDateTime(startdate)

        d2 = Convert.ToDateTime(enddate)

        While d1 <= d2

            For x = 0 To dayofweek.Length - 1

                theday = d2.ToString("dddd")

                If theday = dayofweek(x) Then

                    trigger = True

                End If

            Next

            d2 = d2.AddDays(-1)

        End While

        Return trigger

    End Function

'Determines an available location for the desired time slot.

'Store all the entries Student, date, time(Single or, multiple date), todays date, how it was request3ed the date,clinician, Campus, Subject, with validation

'Also processes [AUTO SELECT] or the manual selection of  Clinicians.

    Public Sub DataEntry()

        Dim intervals As IEvaluateDateTimeIntervals = New datetimeIntervalConversion

        Dim autoAssignClinician As ISchedule = New Scheduling

        Dim manualAssignClinician As ISchedule = New Scheduling

        Dim level As IstudentProfileAttributes = New ReadingLevel

        Dim convertStudentName As INameConversion = New StudentNameconversion

        Dim parseApostrophe As New nameOperation

        Dim dateintervals As New ArrayList

        Dim datestamp As New ArrayList

        Dim timeintervals As New ArrayList

        Dim StudentFullName As String

        Dim Processor As String = String.Empty

        Dim StartDate As String = String.Empty

        Dim EndDate As String = String.Empty

        Dim StartTime As String = String.Empty

        Dim EndTime As String = String.Empty

        Dim SchClinician As String = String.Empty

        Dim PlaceClinician As New ArrayList

        Dim status As String = String.Empty

        Dim requestedDate As String = String.Empty

        Dim MeansofRequest As String = String.Empty

        Dim foundClinician As New List(Of AutoSelectConflicts)

        Dim Location As String = String.Empty

        Dim subject As String = String.Empty

        Dim Priorappointment As String = String.Empty

        Dim p As AutoSelectConflicts

        Dim readingLevel As String = String.Empty

        Dim studentid As String = String.Empty

        Dim Conflict As New DataSet

"unassigned" is a place holder for a day of the week

        Dim week() As String = {"unassigned", "unassigned", "unassigned", "unassigned", "unassigned", "unassigned", "unassigned"}

        If Label14.Text = String.Empty Then

            MsgBox("There is no clinician that is signed in. Please close this screen and sign in.")

            Exit Sub

        End If

        StartDate = DateTimePicker1.Value

    Check to see if the datetimepicker end date is visible

        If CheckBox3.Checked = True Then

            EndDate = DateTimePicker2.Value

        Else

            EndDate = DateTimePicker1.Value

        End If

        'Store all neccessary values

        StartTime = ComboBox3.SelectedItem

        EndTime = ComboBox4.SelectedItem

        SchClinician = ComboBox2.SelectedValue

        StudentFullName = ComboBox1.SelectedValue.ToString

        StudentFullName = parseApostrophe.executeName(StudentFullName, 1)

        requestedDate = MaskedTextBox1.Text

        MeansofRequest = ComboBox5.SelectedItem

        Processor = Label14.Text

Validation check for the how the requested schedule was made.

        If MeansofRequest = String.Empty Then

            MsgBox("You must select a mode of Request")

            Exit Sub

        End If

        Location = ComboBox6.SelectedItem

        subject = ComboBox7.SelectedItem

  Check to see if your choose a block no larger than 1 hour.

        Dim time1 As String = String.Empty

        Dim time2 As String = String.Empty

        Dim totalminutes As Integer = 0

        Dim t1, t2 As DateTime

        Dim ts As TimeSpan

        time1 = ComboBox3.SelectedItem

        time2 = ComboBox4.SelectedItem

        t1 = Convert.ToDateTime(time1)

        t2 = Convert.ToDateTime(time2)

        ts = (t2 - t1)

        totalminutes = ts.TotalMinutes

This was a validation check if the user assigned a student for more than 60 minutes

        'Must be in one hour time slots

        '   If totalminutes > 60 Or totalminutes = 0 Then

        'MsgBox("You can only schedule 1 hr. blocks at a time.")

        '   Exit Sub

        '   End If

        If StudentFullName = String.Empty Then

            MsgBox("You must choose a Student")

        ElseIf ComboBox3.SelectedItem = "" Then

            MsgBox("You use pick a Start Time")

        ElseIf ComboBox4.SelectedItem = "" Then

            MsgBox("You use pick a Finish Time")

 Automate the selection of the next available Clinician then send the result to the display

        ElseIf SchClinician = "AUTO SELECT" Then

            If CheckBox3.Checked = True Then

                fweek = Storedayofweek(week)

                Dim chkdays As Boolean

                chkdays = CheckforselectedDays(StartDate.Trim, EndDate.Trim, week)

                If chkdays = False Then

                    MsgBox("You must select the day(s) of the week")

                    Exit Sub

                End If

            Else

                Dim days As DateTime = Nothing

                Dim daynum As Integer = 0

                days = Convert.ToDateTime(StartDate)

                daynum = days.DayOfWeek

                week(daynum) = days.ToString("dddd").Trim

            End If

            studentid = convertStudentName.convertToId(StudentFullName)

  Determine an available location for the desired time slot.

  Return time and date interval within a date range.

            timeintervals = intervals.timeIntervals(StartTime.Trim, EndTime.Trim)

            dateintervals = intervals.dateIntervals(StartDate.Trim, EndDate.Trim, week)

            readingLevel = level.level(studentid)

Check to see if selected dtudent is already scheduled at the selected date and time

Determine any conflicts within the selected scheduled dates, and store values within the Mainschedule Table within the  database

            foundClinician = autoAssignClinician.AutoSelectClinician(StudentFullName.Trim, StartDate.Trim, EndDate.Trim, timeintervals, dateintervals, "Proposed", requestedDate, MeansofRequest, Location, subject, Processor.Trim)

    Validate check for conflicts for the automated scheduling option.

            Dim person As Integer

            person = foundClinician.Count

            Dim conflictType As String = String.Empty

            Dim conflictDate As String = String.Empty

            Dim conflictTimeIn As String = String.Empty

            Dim conflictTimeOut As String = String.Empty

            Dim tutor As String = String.Empty

            'There are no conflicts then stored values

            If person > 0 Then

                For i = 0 To person - 1

                    p = CType(foundClinician(i), AutoSelectConflicts)

                    tutor = p.D\_Clinician

                    conflictType = p.ConflictType

                    conflictDate = p.ScheduledDate

                    conflictTimeIn = p.DestinationTimeIn

                    conflictTimeOut = p.DestinationTimeout

                    If conflictType = "NothingAvailable" Then

                        MsgBox("There is no available openings on " & p.ScheduledDate & ".")

                    ElseIf conflictType = "self" Then

                        MsgBox(StudentFullName & " is already scheduled on " & p.ScheduledDate & " from " & conflictTimeIn & " to " & conflictTimeOut & ".")

                    End If

                Next

            Else

                MsgBox(StudentFullName & " has been added to the schedule.")

            End If

        Else

'Manually selection option of a clinician

            If CheckBox3.Checked = True Then

                week = Storedayofweek(week)

            Else

                'Determine the day of the week Sun - Sat

                Dim days As DateTime = Nothing

                Dim daynum As Integer = 0

                days = Convert.ToDateTime(StartDate)

                daynum = days.DayOfWeek

                week(daynum) = days.ToString("dddd").Trim

            End If

Determine any conflicts within the selected scheduled dates,and store values within the Mainschedule Table within the  database.

            Dim clinicianoff As Boolean = False

Return time and date interval within a date range.

          timeintervals = intervals.timeIntervals(StartTime.Trim, EndTime.Trim)

          dateintervals = intervals.dateIntervals(StartDate.Trim, EndDate.Trim, week)

            foundClinician = manualAssignClinician.ManuallySelectAClinician(StudentFullName.Trim, SchClinician, StartDate.Trim, EndDate.Trim, StartTime, EndTime.Trim, timeintervals, dateintervals, "Proposed", requestedDate, MeansofRequest, Location, subject, Processor)

            'Validate for conflicts for manual scheduling

            Dim person As Integer

            person = foundClinician.Count

            Dim conflictType As String = String.Empty

            Dim conflictDate As String = String.Empty

            Dim conflictTimeIn As String = String.Empty

            Dim conflictTimeOut As String = String.Empty

            Dim conflictwithstudent As String = String.Empty

            Dim tutor As String = String.Empty

            'There are no conflicts found therefore store values

            If person > 0 Then

                For i = 0 To person - 1

                    p = CType(foundClinician(i), AutoSelectConflicts)

                    tutor = p.D\_Clinician

                    conflictType = p.ConflictType

                    conflictDate = p.ScheduledDate

                    conflictTimeIn = p.DestinationTimeIn

                    conflictTimeOut = p.DestinationTimeout

                    conflictwithstudent = p.ConflictType

                    If conflictType = "student" Then

                        MsgBox("There is no available opening on " & p.ScheduledDate & " from " & p.DestinationTimeIn & " to " & p.DestinationTimeout & "'")

                    ElseIf conflictType = "self" Then

                        MsgBox(StudentFullName & " is already scheduled on " & p.ScheduledDate & " from " & conflictTimeIn & " to " & conflictTimeOut & ".")

                    ElseIf conflictType = "clinician" Then

                        MsgBox(SchClinician & " is scheduled to be off " & p.ScheduledDate & " from " & conflictTimeIn & " to " & conflictTimeOut & ".")

                    End If

                Next

            Else

                MsgBox(StudentFullName & " has been added to the schedule.")

            End If

        End If

        DisplayStudentSchedule()

        HomeDisplay.Show()

        HomeDisplay.Removerows(HomeDisplay.MonthCalendar1.SelectionStart)

        foundClinician.Clear()

    End Sub

    'Store the days of the week Sun - Sat

    Public Function Storedayofweek(ByVal dayofweek As Array) As Array

        If Sunday.Checked = True Then

            dayofweek(0) = "Sunday"

        Else

            dayofweek(0) = "Unassigned"

        End If

        If Monday.Checked = True Then

            dayofweek(1) = "Monday"

        Else

            dayofweek(1) = "Unassigned"

        End If

        If Tuesday.Checked = True Then

            dayofweek(2) = "Tuesday"

        Else

            dayofweek(2) = "Unassigned"

        End If

        If Wednesday.Checked = True Then

            dayofweek(3) = "Wednesday"

        Else

            dayofweek(3) = "Unassigned"

        End If

        If Thursday.Checked = True Then

            dayofweek(4) = "Thursday"

        Else

            dayofweek(4) = "Unassigned"

        End If

        If Friday.Checked = True Then

            dayofweek(5) = "Friday"

        Else

            dayofweek(5) = "Unassigned"

        End If

        If Saturday.Checked = True Then

            dayofweek(6) = "Saturday"

        Else

            dayofweek(6) = "Unassigned"

        End If

        Return dayofweek

    End Function

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

        HomeDisplay.Show()

        HomeDisplay.Removerows(HomeDisplay.MonthCalendar1.SelectionStart)

        Me.Close()

    End Sub

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

        HomeDisplay.Show()

        HomeDisplay.Removerows(HomeDisplay.MonthCalendar1.SelectionStart)

        HomeDisplay.Focus()

        Me.Close()

    End Sub

    'Display a students schedule

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

        RefreshDisplay()

    End Sub

    Public Sub RefreshDisplay()

        Try

            Dim studentinfo As New ScheduleConfig

            Dim Studentfullname As String = Nothing

            Studentfullname = ComboBox1.SelectedValue.ToString

            If Studentfullname = String.Empty Then

                MsgBox("You Must Select a Students name")

                Exit Sub

            End If

            DisplayStudentSchedule()

        Catch ex As Exception

            Throw ex

        End Try

    End Sub

Permanently removes a single, or multiple scheduled dates  from the datasource. Only remove Proposed and Canceled dates.

    Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

        Dim convertName As INameConversion = New StudentNameconversion

        Dim removeDates As ISchedule = New Scheduling

        Dim daysofweek() As String = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}

        Dim row As DataGridViewRow = Nothing

        Dim totalTime As New ArrayList

        Dim appointment As String = String.Empty

        Dim StartTime As String = String.Empty

        Dim EndTime As String = String.Empty

        Dim SchClinician As String = String.Empty

        Dim StudentName As String = String.Empty

        Dim splitname\_student() As String = Nothing

        Dim studentid As String = String.Empty

        Dim hourno As String = [String].Empty

        Dim maxrows As Integer = DataGridView1.Rows.Count

        Dim counter As Integer = -1

        Dim deleteDate(maxrows, 4) As String

        StudentName = ComboBox1.SelectedValue.ToString

        Dim status As String = Nothing

        'Scan each row for a checkbox that is set to true

        Dim requesteddate As String = String.Empty

        Dim Meansofrequest As String = String.Empty

        Dim EntryDate As String = String.Empty

        Dim Processor As String = String.Empty

        Dim Campus As String = String.Empty

        Dim Subject As String = String.Empty

        Dim enteredby As String = String.Empty

        Meansofrequest = ComboBox5.SelectedItem

        EntryDate = MaskedTextBox1.Text

        Processor = Label14.Text

        'Get all checked dates, and store then into an array

        For i = 0 To DataGridView1.RowCount - 1

            'Capture column values from the selected row

            appointment = DataGridView1.Rows(i).Cells(2).Value

            StartTime = DataGridView1.Rows(i).Cells(3).Value

            EndTime = DataGridView1.Rows(i).Cells(4).Value

            SchClinician = DataGridView1.Rows(i).Cells(5).Value

            status = DataGridView1.Rows(i).Cells(6).Value

            Campus = DataGridView1.Rows(i).Cells(7).Value.ToString

            Subject = DataGridView1.Rows(i).Cells(8).Value.ToString

            'Get student id number

            studentid = convertName.convertToId(StudentName)

If the checkbox is set to true then store each column item into a two dimensional array. First element of the array represents the row index, and the  second number represents the attribute of the column

            If DataGridView1.Rows(i).Cells(0).Value = True Then

                status = DataGridView1.Rows(i).Cells(6).Value

                If status.Trim = "Proposed" Then

Increment counter that represents the row index by 1

                    counter = counter + 1

 Store each label into its respective array location

                    DeleteDate(counter, 0) = StudentName.Trim

                    DeleteDate(counter, 1) = appointment.Trim

                    DeleteDate(counter, 2) = StartTime.Trim

                    DeleteDate(counter, 3) = EndTime.Trim

                    DeleteDate(counter, 4) = SchClinician.Trim

                End If

            End If

        Next

Delete the respective intervals by passing the array of  selected row items and the days of the week

        removeDates.RemoveSchedule(deleteDate, daysofweek)

        'Refresh the updated gridview control.

        DisplayStudentSchedule()

        'Refresh Home screen

        HomeDisplay.Removerows(HomeDisplay.MonthCalendar1.SelectionStart)

    End Sub

When checkbox is checked it Selects or Deselects all rows  (dates) in the datagrid control.

    Private Sub CheckBox1\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged

        Dim trigger As Boolean = False

        For i = 0 To DataGridView1.RowCount - 1

            If CheckBox1.Checked = True Then

                DataGridView1.Rows(i).Cells(0).Value = True

                trigger = True

            ElseIf CheckBox1.Checked = False Then

                DataGridView1.Rows(i).Cells(0).Value = False

                trigger = False

            End If

        Next

    End Sub

   Color rows (Scheduled dates) according to their status

    Public Sub GridViewColorcode()

        Dim Status As String

        Dim i As Integer

        Dim numofrows As Integer

        numofrows = DataGridView1.RowCount - 1 'Check for empty Rows

        If numofrows > -1 Then

            For i = 0 To numofrows

                Status = DataGridView1.Rows(i).Cells(6).Value

                If Status.Trim = "Canceled" Then

                    DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.LightGray

                ElseIf Status.Trim = "Proposed" Then

                    DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.Yellow

                ElseIf Status.Trim = "Transfer" Then

                    DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.Azure

                ElseIf Status.Trim = "No Show" Then

                    DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.Red

                ElseIf Status.Trim = "Completed" Then

                    DataGridView1.Rows(i).DefaultCellStyle.BackColor = Color.Green

                End If

            Next

        End If

    End Sub

Checkbox control display multiple date range (two datetimepicker controls) or single date range (one date time picker control)

Display the date range by making the  “End date Datetimepicker” control visible. Relocate control within this area of the screen.

Or Display only a single datetimepicker control.

    Private Sub CheckBox3\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox3.CheckedChanged

        Dim groupBoxlocal As New GroupBox()

        'Single date entry

        If CheckBox3.Checked = False Then

            DateTimePicker2.Visible = False

            GroupBox2.Visible = False

            Label4.Visible = False

            GroupBox4.Location = New Point(37, 100)

            GroupBox3.Location = New Point(256, 100)

            GroupBox1.Size = New Point(563, 260)

            Label2.Location = New Point(31, 165)

            Label12.Location = New Point(198, 165)

            Button1.Location = New Point(340, 185)

            Button2.Visible = True

            ComboBox6.Location = New Point(34, 185)

            ComboBox7.Location = New Point(201, 185)

            ComboBox6.SelectedIndex = 0

            ComboBox7.SelectedIndex = 0

            'Date range entry

        Else

            Label4.Visible = True

            DateTimePicker2.Visible = True

            GroupBox2.Visible = True

            GroupBox4.Location = New Point(37, 150)

            GroupBox3.Location = New Point(256, 150)

            Label2.Location = New Point(39, 215)

            Label12.Location = New Point(198, 215)

            ComboBox6.Location = New Point(34, 235)

            ComboBox7.Location = New Point(201, 235)

            Button1.Location = New Point(340, 230)

            Button2.Visible = False

            GroupBox1.Size = New Point(563, 296)

            ComboBox6.SelectedIndex = 0

            ComboBox7.SelectedIndex = 0

        End If

    End Sub

When the combobox3 start time selector is selected then increment the combobox4 stop stime selector by 60 minutes.

    Private Sub ComboBox3\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox3.SelectedIndexChanged

        Dim time1 As String = String.Empty

        Dim t1, t2 As DateTime

        time1 = ComboBox3.SelectedItem

        t1 = Convert.ToDateTime(time1)

        t2 = t1.AddHours(1)

        ComboBox4.SelectedItem = t2.ToString("h:mm tt")

    End Sub

    Private Sub Button7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click

        ThismonthDaterange()

        RefreshDisplay()

    End Sub

    Private Sub Button6\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click

        Dim monthrange As New ArrayList

        monthrange = NextmonthDaterange()

        DateTimePicker3.Value = monthrange(0)

        DateTimePicker4.Value = monthrange(1)

        RefreshDisplay()

    End Sub

    Private Sub DataGridView1\_CellValueChanged(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellValueChanged

        Dim trigger As Boolean = False

        For i = 0 To DataGridView1.RowCount - 1

  If DirectCast(DataGridView1.Rows(i).Cells(0), DataGridViewCheckBoxCell).Value = True Then

                trigger = True

            End If

        Next

    End Sub

    Private Sub DataGridView1\_CurrentCellDirtyStateChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles DataGridView1.CurrentCellDirtyStateChanged

        If DataGridView1.IsCurrentCellDirty Then

            DataGridView1.CommitEdit(DataGridViewDataErrorContexts.Commit)

        End If

    End Sub

    Private Sub Button8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click

        RescheduleDailyDisplay.Show()

        RescheduleDailyDisplay.Focus()

    End Sub

 Private Sub DataGridView1\_CellContentClick(sender As System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

    End Sub

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

        Me.Close()

    End Sub

    Private Sub Button5\_Click(sender As System.Object, e As System.EventArgs)

        Dim getstudentId As INameConversion = New StudentNameconversion

        Dim studentId As String = String.Empty

        Dim studentname As String = String.Empty

        If ComboBox1.SelectedIndex > 0 Then

            DisplayNotes.Show()

            studentname = ComboBox1.SelectedValue

            studentId = getstudentId.convertToId(studentname)

            DisplayNotes.PopulateGrid(studentId)

        End If

    End Sub

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

        Dim totalDays As Integer

     totalDays = Date.DaysInMonth(DateTimePicker1.Value.Year, DateTimePicker1.Value.Month)

    DateTimePicker2.Value = DateTimePicker1.Value.AddDays(totalDays)

    End Sub

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

        Displaynotes()

    End Sub

    Public Function Displaynotes()

        Dim returnnotes As IstudentAttributesDatasets = New userProfileAttributes

        Dim convertStudentId As INameConversion = New StudentNameconversion

        Dim dv As New DataView

        Dim studentName As String = Nothing

        Dim studentId As String = String.Empty

        If ComboBox1.SelectedIndex > 0 Then

            studentName = ComboBox1.SelectedValue

            studentId = convertStudentId.convertToId(studentName)

            dv = returnnotes.RetrieveNotes(studentId.Trim)

            DataGridView4.DataSource = dv

            DataGridView4.Columns(0).Visible = False

            DataGridView4.Columns(1).Visible = False

            DataGridView4.Columns(3).Width = 150

            DataGridView4.Columns(4).Width = 500

            DataGridView4.Columns(2).SortMode = SortOrder.Descending

        End If

        Return Nothing

    End Function

    Private Sub DateTimePicker3\_ValueChanged(sender As System.Object, e As System.EventArgs) Handles DateTimePicker3.ValueChanged

        DateTimePicker4.Value = DateTimePicker3.Value

    End Sub

    Private Sub DataGridView4\_CellContentClick(sender As System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView4.CellContentClick

        studentProfileUpdate()

    End Sub

    Public Sub studentProfileUpdate()

       Dim studentData As IstudentAttributesCollection = New userAttributesCollection

        Dim convertName As INameConversion = New StudentNameconversion

        Dim studentAttributes As New ArrayList

        Dim parseApostrophe As New nameOperation

        Dim Lastname As String = String.Empty

        Dim Firstname As String = String.Empty

        Dim SchoolDistrict As String = String.Empty

        Dim School As String = String.Empty

        Dim assessmentDate As String = String.Empty

        Dim rptDiscussiondate As String = String.Empty

        Dim initialInquiry As String = String.Empty

        Dim tutorStart As String = String.Empty

        Dim tutorStop As String = String.Empty

        Dim DOB As String = String.Empty

        Dim studentLastName As String = String.Empty

        Dim studentfirstName As String = String.Empty

        Dim studentfullname As String = String.Empty

        Dim gender As String = String.Empty

        Dim studentid As String = String.Empty

        Dim web As String = String.Empty

        Dim activeStudent As String = String.Empty

        studentfullname = ComboBox1.SelectedValue

        EditStudentProfile.Show()

        EditStudentProfile.Focus()

        studentid = convertName.convertToId(studentfullname.Trim)

        studentAttributes = studentData.StudentInfo(studentid.Trim)

        Firstname = studentAttributes(0)

        Lastname = studentAttributes(1)

        SchoolDistrict = studentAttributes(4)

        School = studentAttributes(5)

        gender = studentAttributes(3)

        DOB = studentAttributes(2)

        initialInquiry = studentAttributes(6)

        assessmentDate = studentAttributes(7)

        rptDiscussiondate = studentAttributes(8)

        tutorStart = studentAttributes(9)

        tutorStop = studentAttributes(10)

        activeStudent = studentAttributes(11)

        studentfirstName = parseApostrophe.executeName(Firstname, 2)

        studentLastName = parseApostrophe.executeName(Lastname, 2)

        'Append a ‘0’ to the beginning of the date string

        Dim initialInquiryLength As Integer = 0

        initialInquiryLength = initialInquiry.Length

        If initialInquiryLength = 9 Then

            initialInquiry = "0" & initialInquiry

        End If

        EditStudentProfile.TextBox3.Text = SchoolDistrict.Trim

        EditStudentProfile.TextBox4.Text = School.Trim

        EditStudentProfile.ComboBox1.SelectedItem = gender.Trim

        EditStudentProfile.TextBox1.Text = studentfirstName.Trim

        EditStudentProfile.TextBox2.Text = studentLastName.Trim

        DOB = DateScan(DOB)

        EditStudentProfile.MaskedTextBox1.Text = DOB.Trim

        AssessmentDate = DateScan(AssessmentDate)

        EditStudentProfile.MaskedTextBox2.Text = initialInquiry

        EditStudentProfile.MaskedTextBox3.Text = assessmentDate.Trim

        rptDiscussiondate = DateScan(rptDiscussiondate)

        EditStudentProfile.MaskedTextBox4.Text = rptDiscussiondate.Trim

        TutorStart = DateScan(TutorStart)

        EditStudentProfile.MaskedTextBox5.Text = TutorStart.Trim

        tutorStop = DateScan(tutorStop)

        EditStudentProfile.MaskedTextBox6.Text = tutorStop.Trim

        If activeStudent = "True" Then

            EditStudentProfile.CheckBox1.Checked = True

        ElseIf activeStudent = "False" Then

            EditStudentProfile.CheckBox1.Checked = False

        End If

        If gender.Trim = "Male" Then

            EditStudentProfile.ComboBox1.SelectedIndex = 0

        Else

            EditStudentProfile.ComboBox1.SelectedIndex = 1

        End If

        EditStudentProfile.Label29.Text = studentid

        Dim readingLevel As String = String.Empty

        Dim returnLevel As New returnStudentData

        readingLevel = returnLevel.StudentReadingLevel(studentid.Trim)

        EditStudentProfile.TextBox7.Text = readingLevel.Trim

        Dim studentInfo As IstudentAttributesDatasets = New userProfileAttributes

        Dim schooltype As Boolean

        Dim dt As DataTable

        dt = studentInfo.RetrieveStudentSchool(studentid.Trim)

        Dim row As DataRow

        For Each row In dt.Rows

            schooltype = row("Prv\_Pub")

        Next

        If schooltype = False Then

            EditStudentProfile.TextBox3.Visible = True

            EditStudentProfile.ComboBox2.Visible = False

            EditStudentProfile.TextBox3.Text = SchoolDistrict.Trim

            EditStudentProfile.RadioButton1.Checked = True

            EditStudentProfile.RadioButton2.Checked = False

            EditStudentProfile.Label7.Text = "School District"

        ElseIf schooltype = True Then

            EditStudentProfile.TextBox3.Visible = False

            EditStudentProfile.ComboBox2.Visible = True

            EditStudentProfile.ComboBox2.SelectedItem = SchoolDistrict.Trim

            EditStudentProfile.RadioButton2.Checked = True

            EditStudentProfile.RadioButton1.Checked = False

            EditStudentProfile.Label7.Text = "Type of Private School"

        End If

        EditStudentProfile.TextBox4.Text = School.Trim

        Dim dv As New DataView

        dv = studentInfo.RetrieveNotes(studentid.Trim)

        EditStudentProfile.DataGridView1.DataSource = dv

        EditStudentProfile.DataGridView1.Columns(0).Visible = False

        EditStudentProfile.DataGridView1.Columns(1).Visible = False

        EditStudentProfile.DataGridView1.Columns(3).Width = 150

        EditStudentProfile.DataGridView1.Columns(4).Width = 400

        EditStudentProfile.TextBox5.ReadOnly = True

        EditStudentProfile.TextBox6.ReadOnly = True

    End Sub

    Public Function DateScan(ByVal Value As String) As String

        If Value = String.Empty Then

            Return Value

            Exit Function

        End If

        If Value.Chars(1) = "/" Then

            Value = "0" & Value

        End If

        Return Value.Trim

    End Function

    Private Sub Button2\_Click\_1(sender As System.Object, e As System.EventArgs) Handles Button2.Click

        OpenStudentcalendar()

    End Sub

    Public Function OpenStudentcalendar()

        StudentCalendar.Show()

        StudentCalendar.Focus()

      StudentCalendar.ComboBox1.SelectedValue = DirectCast(Me.ComboBox1.Text.Trim, String)

        DateTimePicker1.Value = Me.DateTimePicker1.Value

        StudentCalendar.StartUp()

        Return Nothing

    End Function

    Private Sub DataGridView4\_DoubleClick(sender As Object, e As System.EventArgs) Handles DataGridView4.DoubleClick

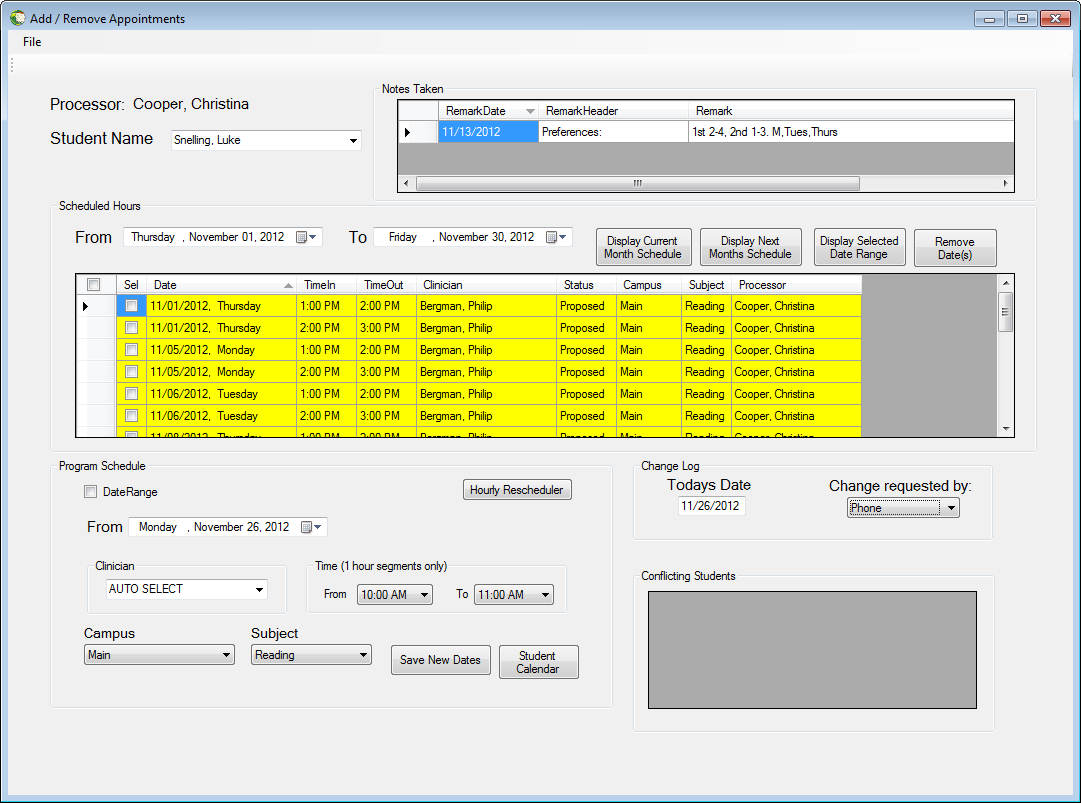
        studentProfileUpdate()

    End Sub

End Class

a

b c f e f g h i j



K l m n o p q r s t u v

This screen allows the user to schedule a student on a single date or multiple dates at a particular time slot. Also, it allows the user to assign a student to a particular clinician; the user can set the computer to assign a student to a clinician. Dates can be removed from a student’s schedule. Reminder notes about a student are displayed. The reason for the scheduled change and how it was requested was made is displayed. Another feature is whenever a student is scheduled the program will inform the user if there is a conflict with another student or a clinician who is scheduled out.