Imports DAL

Imports System.IO

Imports System.Net

Imports System.Globalization

Scans for Scheduling Conflicts between clinicians scheduled out or other students scheduled at the exact same day and time.

Also, the routine checks for any conflict with oneself.

Public Interface IScheduleConflicts

    Function ConflictWithAnotherStudent(ByVal Clinicianfullname As String, ByVal stamp1 As String, ByVal stamp2 As String, ByVal Timein As String, ByVal Timeout As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean

    Function conflict(ByVal Studentfullname As String, ByVal ClinicianName As String, ByVal Date1 As String, ByVal Date2 As String, ByVal STime As String, ByVal FTime As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean

    Function ConflictwithClinician(ByVal clinician As String, ByVal currentdate As Date, ByVal timeintervals As ArrayList, ByVal conflictFlag As Boolean) As Boolean

End Interface

Public Class StudentConflict

    Inherits ScheduleConfig

    Implements IScheduleConflicts

    Dim intervals As IEvaluateDateTimeIntervals = New datetimeIntervalConversion

Public Function ConflictWithAnotherStudent(ByVal clinicianFullname As String, ByVal stamp1 As String, ByVal stamp2 As String, ByVal Timein As String, ByVal Timeout As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean Implements IScheduleConflicts.ConflictWithAnotherStudent

        Dim cliniciannameConversion As INameConversion = New ClinicianNameConversion

        Dim studentNameConversion As INameConversion = New StudentNameconversion

        Dim conflictFlag = False

Clinicianfullname variable can be the id or fullname of the Clinician

        Dim datestamp1, datestamp2 As DateTime

        Dim dsconflict As New DataSet

        Dim clinicianId As String = String.Empty

        Dim PossibleconflictDates As DataTable

Check to see if the clinician full name is bieng passed or the clinician's identification number

Return clincians id if fullname was passed otherwise the clinician's fullname has been passed as the  identification number

        clinicianId = cliniciannameConversion.convertToId(clinicianFullname)

        Dim scheduleinfo As New Schedule

        Dim time1 As DateTime

        Dim time2 As DateTime

        time1 = Convert.ToDateTime(Timein.Trim)

        time2 = Convert.ToDateTime(Timeout.Trim)

        Timein = time1.ToString("h:mm tt")

        Timeout = time2.ToString("h:mm tt")

        datestamp1 = Convert.ToDateTime(stamp1)

        datestamp2 = Convert.ToDateTime(stamp2)

        Dim studentName As String = String.Empty

        Dim conflictTable As DataTable

        conflictTable = Scheduleconflict()

Get the current clinicians entire schedule and match it against the  desired time slots

        PossibleconflictDates = scheduleinfo.ReturnClinicianScheduleinfo(datestamp1, datestamp1, clinicianId.Trim)

        Dim t1 As DateTime

        Dim t2 As DateTime

        Dim rwCount As Integer

        Dim x As Integer

        rwCount = PossibleconflictDates.Rows.Count

'Get total number of dates where the student is scheduled at the  same time

        Dim collectionOfTimeSlots As String = String.Empty

        Dim Sdate As String

        Dim Ct1, Ct2, Ctr As TimeSpan

        Dim dtime1, dtime2, dtime3 As DateTime

        Dim StudentId As String = String.Empty

        Dim clinicianname As String

        Dim row As DataRow

   Check to see if there is one or more conflicts

        If rwCount > 0 Then

            For Each row In PossibleconflictDates.Rows

   'Record each conflict into their respective variables

                Dim Tempdate As DateTime

                StudentId = row("StudentID").ToString.Trim

                Tempdate = row("Date")

                Sdate = Tempdate.ToString("M/d/yyyy")

                t1 = row("TimeIn")

                t2 = row("TimeOut")

                clinicianname = row("ClinicianSignature")

                studentName = studentNameConversion.convertName(StudentId.Trim)

          Convert the time to a string type with the format 'h:mm tt' "

                Dim timeconv1 As String = t1.ToString("h:mm tt")

                Dim timeconv2 As String = t2.ToString("h:mm tt")

                'Clean up clincicians number of any white spaces'

                Dim timeresults As New ArrayList

 Return all the time intervals and store them as an arraylist

                timeresults = intervals.timeIntervals(Timein.Trim, Timeout.Trim)

                Dim max As Integer = timeresults.Count – 1

 'Get the total number of time intervals

                Dim tr As String

                For x = 0 To max

                    tr = timeresults(x)

     'Convert the timeintervals from a string to an DateTime datatype.

      so they can be used in the following conditional statements.

                    collectionOfTimeSlots = timeresults(x)

                    dtime2 = DateTime.ParseExact(timeconv1, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                    Ct1 = dtime2.TimeOfDay

                    dtime3 = DateTime.ParseExact(timeconv2, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                    Ct2 = dtime3.TimeOfDay

                    dtime1 = Date.ParseExact(collectionOfTimeSlots, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                    Ctr = dtime1.TimeOfDay

If the intervals that the user chose fall between the returned  intervals in the database

Then conditional statement is true otherwise ignore

                    If Ctr > Ct1 And Ctr < Ct2 Then

 Add a new record to the dataset in memory that will be binded to the interface

 Store student schedule information into the generic collection

                        storeconflict.Add(New AutoSelectConflicts(studentName, clinicianname.Trim(), Sdate, timeconv1, timeconv2, True, "student"))

                        conflictFlag = True

                        Exit For 'Exit Loop and go to next student

                    Else

                    End If

                Next

            Next

        Else

        End If

        Return conflictFlag  'Return the datset with conflicts

    End Function

Function conflict(ByVal Studentfullname As String, ByVal ClinicianName As String, ByVal Date1 As String, ByVal Date2 As String, ByVal STime As String, ByVal FTime As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean Implements IScheduleConflicts.conflict

        Return Nothing

    End Function

Function ConflictwithClinician(ByVal clinician As String, ByVal currentdate As Date, ByVal timeintervals As ArrayList, ByVal conflictFlag As Boolean) As Boolean Implements IScheduleConflicts.ConflictwithClinician

        Return Nothing

    End Function

End Class

Public Class SelfConflict

    Inherits ScheduleConfig

    Implements IScheduleConflicts

    Dim intervals As IEvaluateDateTimeIntervals = New datetimeIntervalConversion

    Dim studentnameConversion As INameConversion = New StudentNameconversion

 This function checks to see if a student is in conflict with themself

    'Return TRUE or FALSE

    'Otherwise it will return an empty dataset

Public Function conflict(ByVal Studentfullname As String, ByVal ClinicianName As String, ByVal Date1 As String, ByVal Date2 As String, ByVal STime As String, ByVal FTime As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean Implements IScheduleConflicts.conflict

        Dim conflictFlag As Boolean

        Dim Ct1, Ct2, Ctr As TimeSpan

        Dim ds As DataSet

        Dim dt As DataTable

        Dim studentid As String = String.Empty

        Dim returnSelf As New Schedule

        Dim collectionOfTimeSlots As String = String.Empty

        Dim t1 As DateTime

        Dim t2 As DateTime

        'Return student id number

        studentid = studentnameConversion.convertToId(Studentfullname.Trim)

        ds = returnSelf.ReturnStudentSchedule(studentid.Trim, Date1.Trim)

        dt = ds.Tables("MainSchedule")

        Dim dtime1, dtime2, dtime3 As DateTime

        Dim row As DataRow

        Dim targetStudent = -1

        For Each row In dt.Rows

            Dim tempdate As DateTime

            Dim Student, Sdate As String

            Student = row("StudentID")

            tempdate = row("Date")

            t1 = row("TimeIn")

            t2 = row("Timeout")

            'Return the start date of the student

            Sdate = tempdate.ToString("M/d/yyyy")

            Dim timeconv1 As String = t1.ToString("h:mm tt")

            Dim timeconv2 As String = t2.ToString("h:mm tt")

            Dim timeresults As New ArrayList

            timeresults = intervals.timeIntervals(STime.Trim, FTime.Trim)

            Dim max As Integer = timeresults.Count - 1

            Dim tr As String = String.Empty

            For x = 0 To max

                collectionOfTimeSlots = timeresults(x)

                dtime2 = DateTime.ParseExact(timeconv1, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                Ct1 = dtime2.TimeOfDay

                dtime3 = DateTime.ParseExact(timeconv2, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                Ct2 = dtime3.TimeOfDay

                dtime1 = Date.ParseExact(collectionOfTimeSlots, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                Ctr = dtime1.TimeOfDay

 Check for conflict within time slot.

     If the desired time falls within the conflict of when the student is scheduled then flag the conflict

                If Ctr > Ct1 And Ctr < Ct2 Then

                    storeconflict.Add(New AutoSelectConflicts(Studentfullname.Trim, ClinicianName.Trim, Sdate.Trim, timeconv1.Trim, timeconv2.Trim, True, "self"))

                    conflictFlag = True

                    Return conflictFlag

                    Exit For

                End If

            Next

        Next

        Return conflictFlag

    End Function

Public Function ConflictWithAnotherStudent(ByVal Clinicianfullname As String, ByVal stamp1 As String, ByVal stamp2 As String, ByVal Timein As String, ByVal Timeout As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean Implements IScheduleConflicts.ConflictWithAnotherStudent

        Return Nothing

    End Function

Function ConflictwithClinician(ByVal clinician As String, ByVal currentdate As Date, ByVal timeintervals As ArrayList, ByVal conflictFlag As Boolean) As Boolean Implements IScheduleConflicts.ConflictwithClinician

        Return Nothing

    End Function

End Class

Public Class ClinicianConflict

    Inherits Clinicians

    Implements IScheduleConflicts

Public Function ConflictWithAnotherStudent(ByVal Clinicianfullname As String, ByVal stamp1 As String, ByVal stamp2 As String, ByVal Timein As String, ByVal Timeout As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean Implements IScheduleConflicts.ConflictWithAnotherStudent

        Return Nothing

    End Function

Function conflict(ByVal Studentfullname As String, ByVal ClinicianName As String, ByVal Date1 As String, ByVal Date2 As String, ByVal STime As String, ByVal FTime As String, ByVal storeconflict As List(Of AutoSelectConflicts)) As Boolean Implements IScheduleConflicts.conflict

        Return Nothing

    End Function

Function ConflictwithClinician(ByVal clinicianId As String, ByVal currentdate As Date, ByVal timeintervals As ArrayList, ByVal conflictFlag As Boolean) As Boolean Implements IScheduleConflicts.ConflictwithClinician

        Dim t1, t2 As DateTime

        Dim ClinicianName As New ArrayList

        Dim ClinicianFullName As New ArrayList

        Dim storeConflict As New List(Of AutoSelectConflicts)

        Dim idnumber As Integer = 0

        Dim convertname As New Schedule

        Dim dTime1, dTime2, dTime3 As DateTime

        Dim Conflict As New DataSet

        clinicianScheduleResults As New Clinicians

        Dim dsa As New DataSet

        dsa = clinicianOutSchedule(currentdate, currentdate)

 'Return all Clinician that are out at the desired Time and Date range

Check to see if the current Clinician in the iteration is scheduleed  to be 'OUT' at this iterations particular date and time

        Dim dt As DataTable = dsa.Tables("Clinician\_DailyOutSchedule")

        'Set the current interval to DateStop and DateStart

        Dim foundrow() As DataRow = dt.Select("ClinicianId='" & clinicianId.Trim & "' AND Date='" & currentdate & "'")

'Search for the current Clinicianin the Dailyout dataset

Return the row containing the Date timein and Timeout values of the  specific Clinician

        Dim tm As TimeSpan

        Dim tIn As TimeSpan

        Dim tOut As TimeSpan

        Dim rw1 As DataRow

        Dim collectionOfTimeSlots As String = String.Empty

        Dim sTime As String = String.Empty

        Dim fTime As String = String.Empty

        For Each rw1 In foundrow

            t1 = rw1("TimeIn")

            t2 = rw1("TimeOut")

            sTime = t1.ToString("h:mm tt")

            fTime = t2.ToString("h:mm tt")

            Dim y As Integer

            'iterate through each stored 30 min time slot

            For y = 0 To timeintervals.Count - 1

                collectionOfTimeSlots = timeintervals(y)

                dTime2 = DateTime.ParseExact(sTime, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                tIn = dtime2.TimeOfDay

                dTime3 = DateTime.ParseExact(fTime, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                tOut = dtime3.TimeOfDay

                dTime1 = Date.ParseExact(collectionOfTimeSlots, "h:mm tt", Globalization.DateTimeFormatInfo.InvariantInfo)

                tm = dTime1.TimeOfDay

                'Check to see if Clinician is out at this timeslot.

                If tm > tIn And tm < tOut Then

'conflict is flagged as true if the time interval is in between timein and timeout (Conflict=TRUE)

                    conflictFlag = True

                    Return conflictFlag

                    Exit Function

                Else

                    conflictFlag = False

                End If

            Next

        Next

        Return conflictFlag

    End Function

End Class