Imports DAL

Imports System.IO

Imports System.Net

Imports System.Globalization

 Reschedule a student from one hour to the next

Public Interface IReschedule

    Function AutoSelectClinician(ByVal StudentFullName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts)

Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts)

Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal StartTime As String, ByVal Endtime As String, ByVal timeintervals As ArrayList) As List(Of AutoSelectConflicts)

End Interface

Public Class ManuallyRescheduling

    Implements IReschedule

    Public Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts) Implements IReschedule.ClinicianAvailability

        Dim cliniciannameConversion As INameConversion = New ClinicianNameConversion

        Dim studentnameConversion As INameConversion = New StudentNameconversion

        Dim student\_already\_schedulein\_timeslot As IScheduleConflicts = New SelfConflict

        Dim student\_conflict\_with\_another\_student As IScheduleConflicts = New StudentConflict

        Dim student\_conflict\_with\_clinician As IScheduleConflicts = New ClinicianConflict

        Dim stored\_Conflict As New List(Of AutoSelectConflicts)

        Dim conflictFlag2 As Boolean = False

        Dim conflictFlag3 As Boolean = False

        Dim conflictFlag4 As Boolean = False

        Dim conflictFlag As Boolean = False

        Dim clinicianId As String = String.Empty

        Dim Studentid As String = String.Empty

        Dim timeIn As String = String.Empty

        Dim timeOut As String = String.Empty

        Dim tm1 As DateTime

        Dim tm2 As DateTime

        Dim conflictType As String = String.Empty

        Dim Conflict As New DataSet

        Dim currentDate As Date

        clinicianId = cliniciannameConversion.convertToId(ClinicianName)

        Studentid = studentnameConversion.convertToId(StudentFullName)

        tm1 = timeintervals(0) 'Set the Starttime to 'TimeIn'

        tm2 = timeintervals(timeintervals.Count - 1) 'Set Final time to timeOut

        timeIn = tm1.ToString("h:mm tt")

        timeOut = tm2.ToString("h:mm tt")

        currentDate = Convert.ToDateTime(DateStart).ToLongDateString

        'Check to see if the Student is scheduled at the same time range. If so then set the 'reflexive variable to TRUE

        conflictFlag4 = student\_already\_schedulein\_timeslot.conflict(StudentFullName.Trim, ClinicianName.Trim, DateStart.Trim, DateStart.Trim, timeIn.Trim, timeOut.Trim, stored\_Conflict)

        'Check to see if the student has a conflict with another student with the particular clinician being off on the currentdate and the specific time slot within this iteration of dt1

        conflictFlag2 = student\_conflict\_with\_clinician.ConflictwithClinician(clinicianId.Trim, currentDate, timeintervals, conflictFlag)

        'Check to see if there is a conflict with another student at the specific time slot

        conflictFlag3 = student\_conflict\_with\_another\_student.ConflictWithAnotherStudent(ClinicianName.Trim, DateStart.Trim, DateStart.Trim, timeIn.Trim, timeOut.Trim, stored\_Conflict) 'Check to see if there is a Student Scheduled at this Specific time slot

        If conflictFlag3 = True Or conflictFlag4 = True Then

            conflictFlag3 = False

            conflictFlag4 = False

        ElseIf conflictFlag2 = True Then

            stored\_Conflict.Add(New AutoSelectConflicts(StudentFullName, ClinicianName.Trim, DateStart.Trim, timeIn, timeOut, True, "clinician"))

            conflictFlag2 = False

        End If

        Return stored\_Conflict

    End Function

    Function AutoSelectClinician(ByVal StudentFullName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts) Implements IReschedule.AutoSelectClinician

        Return Nothing

    End Function

    Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal StartTime As String, ByVal Endtime As String, ByVal timeintervals As ArrayList) As List(Of AutoSelectConflicts) Implements IReschedule.ClinicianAvailability

        Return Nothing

    End Function

End Class

Public Class ManuallyReschedulingTime

    Implements IReschedule

    Public Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal StartTime As String, ByVal Endtime As String, ByVal timeintervals As ArrayList) As List(Of AutoSelectConflicts) Implements IReschedule.ClinicianAvailability

        Dim cliniciannameConversion As INameConversion = New ClinicianNameConversion

        Dim studentnameConversion As INameConversion = New StudentNameconversion

        Dim studentAlreadyScheduleinTimeslot As IScheduleConflicts = New SelfConflict

Dim studentConflictWithAnotherStudent As IScheduleConflicts = New StudentConflict

 Dim studentConflictWithClinician As IScheduleConflicts = New ClinicianConflict

        Dim stored\_Conflict As New List(Of AutoSelectConflicts)

        Dim conflictFlag2 As Boolean = False

        Dim conflictFlag3 As Boolean = False

        Dim conflictFlag4 As Boolean = False

        Dim conflictFlag As Boolean = False

        Dim clinicianId As String = String.Empty

        Dim Studentid As String = String.Empty

        Dim timeIn As String = String.Empty

        Dim timeOut As String = String.Empty

        Dim tm1 As DateTime

        Dim tm2 As DateTime

        Dim conflictType As String = String.Empty

        Dim Conflict As New DataSet

        Dim currentDate As Date

        clinicianId = cliniciannameConversion.convertToId(ClinicianName)

        Studentid = studentnameConversion.convertToId(StudentFullName)

        tm1 = timeintervals(0) 'Set the Starttime to 'TimeIn'

        tm2 = timeintervals(timeintervals.Count - 1) 'Set Final time to timeOut

        timeIn = tm1.ToString("h:mm tt")

        timeOut = tm2.ToString("h:mm tt")

        currentDate = Convert.ToDateTime(DateStart).ToLongDateString

        'Check to see if the Student is scheduled at the same time range. If so then set the 'reflexive variable to TRUE

        conflictFlag4 = studentAlreadyScheduleinTimeslot.conflict(StudentFullName.Trim, ClinicianName.Trim, DateStart.Trim, DateStart.Trim, timeIn.Trim, timeOut.Trim, stored\_Conflict)

        'Check to see if the student has a conflict with another student with the particular clinician being off on the currentdate and the specific time slot within this iteration of dt1

        conflictFlag2 = studentConflictWithClinician.ConflictwithClinician(clinicianId.Trim, currentDate, timeintervals, conflictFlag)

        'Check to see if there is a conflict with another student at the specific time slot

        conflictFlag3 = studentConflictWithAnotherStudent.ConflictWithAnotherStudent(ClinicianName.Trim, DateStart.Trim, DateStart.Trim, timeIn.Trim, timeOut.Trim, stored\_Conflict) 'Check to see if there is a Student Scheduled at this Specific time slot

        If conflictFlag3 = True Or conflictFlag4 = True Then

            conflictFlag3 = False

            conflictFlag4 = False

        ElseIf conflictFlag2 = True Then

            'Store the conflict

            stored\_Conflict.Add(New AutoSelectConflicts(StudentFullName, ClinicianName, DateStart.Trim, timeIn, timeOut, True, "clinician"))

            conflictFlag2 = False

        End If

        Return stored\_Conflict

    End Function

    Function AutoSelectClinician(ByVal StudentFullName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts) Implements IReschedule.AutoSelectClinician

        Return Nothing

    End Function

    Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts) Implements IReschedule.ClinicianAvailability

        Return Nothing

    End Function

End Class

Public Class AutomateRescheduling

    Inherits Clinicians

    Implements IReschedule

    Dim intervals As IEvaluateDateTimeIntervals = New datetimeIntervalConversion

    Function AutoSelectClinician(ByVal StudentFullName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts) Implements IReschedule.AutoSelectClinician

        Dim cliniciannameConversion As INameConversion = New ClinicianNameConversion

        Dim studentnameConversion As INameConversion = New StudentNameconversion

        Dim studentAlreadyScheduleInTimeslot As IScheduleConflicts = New SelfConflict

 Dim studentConflictWithAnotherStudent As IScheduleConflicts = New StudentConflict

Dim studentConflictWithClinician As IScheduleConflicts = New ClinicianConflict

        Dim tutor As String = String.Empty

        Dim conflictFlag2 As Boolean

        Dim conflictFlag3 As Boolean

        Dim conflictFlag4 As Boolean

        Dim conflictFlag As Boolean

        Dim studentid As String = String.Empty

        Dim clinicianId As String = String.Empty

        Dim dateavailability As Boolean = True

        Dim timeIn As String = String.Empty

        Dim timeOut As String = String.Empty

        Dim tm1, tm2 As DateTime

        Dim conflictType As String = String.Empty

        Dim Conflict As New DataSet

        Dim scheduleresults As New Schedule

        '  Dim clinicianScheduleResults As New Clinicians

        Dim currentDate As Date

        Dim dsa As New DataSet

        Dim clinicianData As New DataSet

        Dim autoSelect As Boolean = True

        Dim maindisplayidnumber As New idgenerator

        Dim stored\_Conflict As New List(Of AutoSelectConflicts)

        Dim idnumber As Integer = 0

        Dim conflictLabel As String = String.Empty

        Dim convDate As String = String.Empty

        Dim d1 As String = String.Empty

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

        clinicianData = GetClinicianInfo(True) 'Return all  Clinician that are active

        studentid = studentnameConversion.convertToId(StudentFullName) 'get student Id number

        Dim dt1 As DataTable = clinicianData.Tables("Clinician")

        Dim rw As DataRow

        For Each rw In dt1.Rows 'Iterate through all active Clinicians

            clinicianId = rw("ClinicianId")

            tutor = cliniciannameConversion.convertname(clinicianId)

            autoSelect = rw("AutoSelect")

            If autoSelect = True Then

                conflictFlag = 0 'Reset the flag

                tm1 = timeintervals(0) 'Set the Starttime to 'TimeIn'

                tm2 = timeintervals(timeintervals.Count - 1) 'Set Final time to timeOut

                timeIn = tm1.ToString("h:mm tt")

                timeOut = tm2.ToString("h:mm tt")

                currentDate = Convert.ToDateTime(DateStart).ToLongDateString

                'Check to see if student is already scheduled at this time

                conflictFlag4 = studentAlreadyScheduleInTimeslot.conflict(StudentFullName, tutor, DateStart.Trim, DateStart.Trim, timeIn.Trim, timeOut.Trim, stored\_Conflict)

                'Check to see if the clinician is scheduled to be off

                conflictFlag2 = studentConflictWithClinician.ConflictwithClinician(clinicianId, currentDate, timeintervals, conflictFlag)

                'Check to see if the student has a conflict with another student with the particular clinician within this iteration

                conflictFlag3 = studentConflictWithAnotherStudent.ConflictWithAnotherStudent(tutor, DateStart.Trim, DateStart.Trim, timeIn.Trim, timeOut.Trim, stored\_Conflict) 'Check to see if there is a Student Scheduled at this Specific time slot

                If conflictFlag4 = True Then

                    dateavailability = False

                    conflictFlag4 = False

                    Exit For

                ElseIf conflictFlag2 = True Or conflictFlag3 = True Then

                    dateavailability = False

                    conflictFlag2 = False

                    conflictFlag3 = False

                Else

                    dateavailability = True

Exit For

                End If

            End If

        Next

        If dateavailability = False Then

            conflictLabel = "NothingAvailable"

            stored\_Conflict.Add(New AutoSelectConflicts(StudentFullName, tutor, DateStart, timeIn, timeOut, True, conflictLabel))

            dateavailability = False

        End If

        Return stored\_Conflict

    End Function

Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal PriorStartTime As String, ByVal PriorEndTime As String, ByVal StartTime As String, ByVal Endtime As String, ByVal priorappointment As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal excuse As String, ByVal location As String, ByVal subject As String, ByVal Processor As String, ByVal noteValue As String, ByVal xfrhour As String, ByVal x1 As Boolean, ByVal x2 As Boolean) As List(Of AutoSelectConflicts) Implements IReschedule.ClinicianAvailability

        Return Nothing

    End Function

    Function ClinicianAvailability(ByVal StudentFullName As String, ByVal ClinicianName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal StartTime As String, ByVal Endtime As String, ByVal timeintervals As ArrayList) As List(Of AutoSelectConflicts) Implements IReschedule.ClinicianAvailability

        Return Nothing

    End Function

End Class