Imports DAL

Imports System.IO

Imports System.Net

'Check to see the next available clinician

'Checks to see if the clinician is scheduled out

'Removes a Student from the Schedule

Public Interface ISchedule

    Function AutoSelectClinician(ByVal StudentFullName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal location As String, ByVal subject As String, ByVal Processor As String) As List(Of AutoSelectConflicts)

    Function ManuallySelectAClinician(ByVal studentFullName As String, ByVal clinicianName As String, ByVal StartDate As String, ByVal EndDate As String, ByVal StartTime As String, ByVal EndTime As String, ByVal timeIntervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal location As String, ByVal subject As String, ByVal Processor As String) As List(Of AutoSelectConflicts)

    Function RemoveSchedule(ByVal Student As Array, ByVal selecteddays As Array)

End Interface

'Removes a clinician off days

Public Interface IscheduleClinician

Function deleteClinicianOffdays(ByVal clinician As Array, ByVal selecteddays As Array)

End Interface

Public Class Scheduling

    Inherits CommitChanges

    Implements ISchedule

    Dim cliniciannameConversion As INameConversion = New ClinicianNameConversion

    Dim intervals As IEvaluateDateTimeIntervals = New datetimeIntervalConversion

    Dim clinicianName As INameConversion = New ClinicianNameConversion

    Dim studentName As INameConversion = New StudentNameconversion

Function AutoSelectClinician(ByVal StudentFullName As String, ByVal DateStart As String, ByVal DateStop As String, ByVal timeintervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal location As String, ByVal subject As String, ByVal Processor As String) As List(Of AutoSelectConflicts) Implements ISchedule.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 Cdate1, Cdate2, currentDate As Date

        Dim dsActiveclinicians As New DataSet

        Dim autoSelect As Boolean = True

        Dim mainDisplayIdnum As Iidnumbers = New GenerateMainDisplayIdnum

        Dim storeConflict 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

        Cdate1 = Convert.ToDateTime(DateStart)

        Cdate2 = Convert.ToDateTime(DateStop)

        dsActiveclinicians = clinicianScheduleResults.GetClinicianInfo(True)

'Return all Clinician that are active

        Studentid = studentnameConversion.convertToId(StudentFullName.Trim)

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

        Dim rw As DataRow

        Dim w As Integer

'Increment to the next date in the the interval range chosen by the user

        For w = 0 To dateintervals.Count 1

 'Iterate through each active Clinicians

'Check clinician active status.Execute code if clinician is active otherwise iterate to the next Clinician

            For Each rw In dt1.Rows

'Check to see if clinician is set for auto selection

                autoSelect = rw("AutoSelect")

                If autoSelect = True Then

                    conflictFlag = 0 'Reset the flag

                    clinicianId = rw("ClinicianId")

                    tutor = cliniciannameConversion.convertName(clinicianId.Trim)

                    d1 = dateintervals(w)

                    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(d1).ToLongDateString

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

                    conflictFlag4 = studentAlreadyScheduleInTimeslot.conflict(StudentFullName.Trim, tutor.Trim, d1.Trim, d1.Trim, timeIn.Trim, timeOut.Trim, storeConflict)

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

                    conflictFlag2 = studentConflictWithClinician.ConflictwithClinician(clinicianId.Trim, 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.Trim, d1.Trim, d1.Trim, timeIn.Trim, timeOut.Trim, storeConflict)

'Check to see if there is a Student Scheduled at this Specific time slot

                    If conflictFlag4 = True Then

                        dateavailability = False

                        conflictFlag4 = False

                        'Go to next Date

                        Exit For

                    ElseIf conflictFlag2 = True Then

                        dateAvailability = False

                        conflictFlag2 = False

                        'Go to next Clinician

                    ElseIf conflictFlag3 = True Then

                        dateAvailability = False

                        conflictFlag3 = False

'Go to next Clinician

                    Else

'Save student to the data source

                        dateAvailability = True

'Generate row key for the maindisplaySchedule

                        idnumber = mainDisplayIdnum.generateId

Cleanup the processors name by adjusting an accent mark if one exsist.

                        Dim replaceAccentMark As New nameOperation

                        Processor = replaceAccentMark.executeName(Processor, 1)

                        tutor = replaceAccentMark.executeName(tutor, 1)

                        Submitstudent(idnumber, Studentid.Trim, clinicianId.Trim, tutor.Trim, timeIn.Trim, timeOut.Trim, currentDate, Status.Trim, requesteddate, Howrequestwasmade, location, subject, Processor)

                        'Proceed to the next Date

                        Exit For

                    End If

                End If

            Next

            If dateavailability = False Then

                conflictLabel = "NothingAvailable"

                storeConflict.Add(New AutoSelectConflicts(StudentFullName, tutor, dateintervals(w), timeIn, timeOut, True, conflictLabel))

                dateavailability = False

            End If

        Next

        Return storeConflict

    End Function

Public Function ManuallySelectAClinician(ByVal studentFullName As String, ByVal clinicianName As String, ByVal StartDate As String, ByVal EndDate As String, ByVal StartTime As String, ByVal EndTime As String, ByVal timeIntervals As ArrayList, ByVal dateintervals As ArrayList, ByVal Status As String, ByVal requesteddate As String, ByVal Howrequestwasmade As String, ByVal location As String, ByVal subject As String, ByVal Processor As String) As List(Of AutoSelectConflicts) Implements ISchedule.ManuallySelectAClinician

       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 mainDisplayIdnum As Iidnumbers = New GenerateMainDisplayIdnum

        Dim storeConflict As New List(Of AutoSelectConflicts)

        Dim AutoSelect As Boolean = True

        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 scheduleresults As New Schedule

        Dim currentDate As Date

        Dim idnumber As Integer = 0

        Dim d1 As String

        clinicianId = cliniciannameConversion.convertToId(clinicianName)

        Studentid = studentnameConversion.convertToId(studentFullName)

        Dim w As Integer

        For w = 0 To dateintervals.Count - 1

'Increment to the next date in the the interval range chosen by the user

            conflictFlag = False 'Reset the flag

            d1 = dateintervals(w)

            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(d1)

'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, d1.Trim, d1.Trim, timeIn.Trim, timeOut.Trim, storeConflict)

'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)

            conflictFlag3 = studentConflictWithAnotherStudent.ConflictWithAnotherStudent(clinicianName.Trim, d1.Trim, d1.Trim, timeIn.Trim, timeOut.Trim, storeConflict)

'Check to see if there is a Student Scheduled at this Specific time slot

            If conflictFlag3 = True Then

                conflictFlag3 = False

            ElseIf conflictFlag4 = True Then

                conflictFlag4 = False

            ElseIf conflictFlag2 = True Then

'Record conflict with clinician being scheduled out

                storeConflict.Add(New AutoSelectConflicts(studentFullName.Trim, clinicianName.Trim, dateintervals(w), timeIn.Trim, timeOut.Trim, True, "clinician"))

                conflictFlag2 = False

            Else

'Save student to the data source

'Get primaryKey for record

                idnumber = mainDisplayIdnum.generateId

Cleanup the processors name by adjusting an accent mark if one exsist.

Dim replaceAccentMark As New nameOperation

                Processor = replaceAccentMark.executeName(Processor, 1)

                clinicianName = replaceAccentMark.executeName(clinicianName, 1)

                Submitstudent(idnumber, Studentid.Trim, clinicianId.Trim, clinicianName.Trim, timeIn.Trim, timeOut.Trim, currentDate, Status.Trim, requesteddate, Howrequestwasmade, location, subject, Processor)

            End If

        Next

        Return storeConflict

    End Function

Public Function RemoveSchedule(ByVal removeScheduleAppointment As Array, ByVal selecteddays As Array) Implements ISchedule.RemoveSchedule

        Dim tempdate As Date

        Dim totaltimeintervals As ArrayList

        Dim allStudents As Integer = UBound(removeScheduleAppointment)

        Dim studentid As String = String.Empty

        Dim clinicianid As String = String.Empty

        Dim Firstname As String = String.Empty

        Dim Lastname As String = String.Empty

        For y = 0 To allStudents

            Dim dateintervals As ArrayList

            dateintervals = intervals.dateIntervals(removeScheduleAppointment(y, 1), removeScheduleAppointment(y, 1), selecteddays)

            totaltimeintervals = intervals.timeIntervals(removeScheduleAppointment(y, 2), removeScheduleAppointment(y, 3))

            For x = 0 To dateintervals.Count - 1

                Dim Studentfullname As String = removeScheduleAppointment(y, 0)

                Dim clinicianfullName As String = removeScheduleAppointment(y, 4)

                studentid = studentName.convertToId(Studentfullname)

                If clinicianfullName.Trim = "TRI" Then

                    clinicianid = "018c"

                Else

                    clinicianid = clinicianName.convertToId(clinicianfullName)

                End If

                tempdate = FormatDateTime(dateintervals(x), DateFormat.LongDate)

                RemoveStudent(studentid.Trim, removeScheduleAppointment(y, 2), removeScheduleAppointment(y, 3), tempdate, clinicianid.Trim)

            Next

        Next

        Return Nothing

    End Function

End Class