Imports System

Imports System.Globalization

Imports System.Data

Imports System.Data.SqlClient

Imports System.IO

Imports System.Configuration

Imports DAL

'DataSet that are used to prepare the data to be displayed and bound to the calling User interfaces.

Public Interface IDisplaySetup

    Function mainDisplaySchedule(ByVal DateDefinition As Date) As DataSet

End Interface

Public Class DisplayModule

    Inherits ScheduleTemplate

    Implements IDisplaySetup

    Dim scheduleResults As New Schedule

    Dim clinicianinfo As New Clinicians

    Dim intervals As IEvaluateDateTimeIntervals = New datetimeIntervalConversion

    Public Function mainDisplaySchedule(ByVal DateDefinition As Date) As DataSet Implements IDisplaySetup.mainDisplaySchedule

        Dim findApostrophe As IparseUserName = New parsefullName

        Dim parseApostrophe As New nameOperation

        'Convert date definition to a string

        Dim statusid As New ArrayList

        Dim ds, ds1, ds2 As New DataSet

        'Return the template of the schedule for the display. This is the dataset that will store the Schedule.

        '

        ds = ScheduleTemplate()

        Dim dt1 As DataTable = ds.Tables("ScheduleDisplayScreen")

        Dim y As Integer

        Dim t1, t2 As String

        Dim Clinician As String

        Dim subject As String = Nothing

        Dim slot As DateTime

        Dim timeintervals As ArrayList

        Dim name As New ArrayList

        Dim Studentname As New ArrayList

        Dim studentinfo As New returnStudentData

        Dim studentfullName As String = Nothing

        Dim clinicianFullName As String = String.Empty

        'Return all clinician off days for the current date

        Dim ds4 As New DataSet

        ds4 = clinicianinfo.clinicianOutSchedule(DateDefinition, DateDefinition)

        Dim dt4 As DataTable = ds4.Tables("Clinician\_DailyOutSchedule")

        'Return maximum number of active Clinicians

        Dim ds8 As New DataSet

        Dim dt As DataTable

        ds2 = studentinfo.GetStudentInfo

        ds8 = scheduleResults.GetSchedule(DateDefinition, DateDefinition)

        dt = ds8.Tables("MainSchedule")

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

        Dim dt3 As DataTable = clinicianinfo.ClinicianProfile

        Dim dr1 As DataRow

        Dim hour As DateTime

        Dim returntime As New ArrayList

        Dim dsclinicians As New DataSet

        'Return id numbers of all Clinicians who are set as active

        dsclinicians = clinicianinfo.GetClinicianInfo(True)

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

        Dim cln, cfn As String

        Dim row As DataRow

        For Each row In dtClinicians.Rows

            REM Get Clinicians full name associated with the ID

            REM cfn = Clinicians First name and cln=Clinicians Last name

            REM sepearate with a comma

            dr1 = dt1.NewRow()

            cln = row("LastName")

            cfn = row("FirstName")

Dim query As String = "LastName='" & cln.Trim & "' AND FirstName='" & cfn.Trim & "'"

            ClinicianFullName = cln.Trim & ", " & cfn.Trim

            Clinician = parseApostrophe.executeName(ClinicianfullName, 0)

            dr1("Clinician") = Clinician.Trim

            Dim clid As String = String.Empty

            Dim id As String = Nothing

            Dim foundClinician() As DataRow = dt3.Select(query)

            Dim a As Integer

            Dim cid As String = Nothing

            REM Search for Clincian id number

            For a = 0 To foundClinician.Length - 1

                clid = foundClinician(a)("ClinicianID")

                cid = clid.Trim

                Dim time1 As String

                Dim time2 As String

                Dim stmp As DateTime

                Dim ftmp As DateTime

                Dim index As Integer

                Dim b As Integer

Search for the Clinicians time schedule from the table called MainSchedule

          Dim foundrows() As DataRow = dt.Select("ClinicianID='" & cid.Trim & "'")

                Dim Status As String = String.Empty

                For b = 0 To foundrows.Length - 1

                    'Convert Time to string

                    stmp = foundrows(b)("TimeIn")

                    ftmp = foundrows(b)("TimeOut")

                    time1 = stmp.ToString("hh:mm tt")

                    time2 = ftmp.ToString("hh:mm tt")

                    Status = foundrows(b)("Status")

                    index = foundrows(b)("Count")

                    REM Determine the classroom information and campus of the Student

                    Dim ds9 As New DataSet

                    ds9 = scheduleResults.GetClassroomData(index)

                    Dim dt9 As DataTable = ds9.Tables("Classroom")

                    Dim Location As String = String.Empty

                    Dim Classrow As DataRow

                    Dim x As Integer = dt9.Rows.Count - 1

                    For Each Classrow In dt9.Rows

                        Location = Classrow("Campus")

                    Next

                    REM Return the Students identification number

                    Dim stid As String = Convert.ToString(foundrows(b)("StudentID"))

Dim foundname() As DataRow = dt2.Select("StudentID='" & stid.Trim & "'")

                    Dim fn, ln As String

                    studentfullName = String.Empty

Find the student using their identification number and determine his name and place him in the respective time slot.

                    For c = 0 To foundname.Length - 1

                        id = Convert.ToString(foundname(c)("StudentID"))

                        ln = foundname(c)("Last Name")

                        fn = foundname(c)("First Name")

                        ln = parseApostrophe.executeName(ln, 2)

                        fn = parseApostrophe.executeName(fn, 2)

                        studentfullName = ln.Trim & ", " & fn.Trim

Return all time 30 min time intervals between the time the student is scheduled to  start and finish

                        returntime = intervals.timeIntervals(time1, time2)

                        Dim maxtime As Integer = returntime.Count - 1

Determine if the student is has 1)Rescheduled/Transfer, 2)Canceled, or 3)Proposed

                        dr1("Status") = Status.Trim

Iterate through each half hour segment and check for a match. If there is a match store the studentsfull name.

                        For y = 0 To maxtime - 1

                            hour = returntime(y)

                            Select Case hour

                                Case #7:30:00 AM#

                                    dr1("7:30 AM") = studentfullName

                                Case #8:00:00 AM#

                                    dr1("8:00 AM") = studentfullName

                                Case #8:30:00 AM#

                                    dr1("8:30 AM") = studentfullName

                                Case #9:00:00 AM#

                                    dr1("9:00 AM") = studentfullName

                                Case #9:30:00 AM#

                                    dr1("9:30 AM") = studentfullName

                                Case #10:00:00 AM#

                                    dr1("10:00 AM") = studentfullName

                                Case #10:30:00 AM#

                                    dr1("10:30 AM") = studentfullName

                                Case #11:00:00 AM#

                                    dr1("11:00 AM") = studentfullName

                                Case #11:30:00 AM#

                                    dr1("11:30 AM") = studentfullName

                                Case #12:00:00 PM#

                                    dr1("12:00 PM") = studentfullName

                                Case #12:30:00 PM#

                                    dr1("12:30 PM") = studentfullName

                                Case #1:00:00 PM#

                                    dr1("1:00 PM") = studentfullName

                                Case #1:30:00 PM#

                                    dr1("1:30 PM") = studentfullName

                                Case #2:00:00 PM#

                                    dr1("2:00 PM") = studentfullName

                                Case #2:30:00 PM#

                                    dr1("2:30 PM") = studentfullName

                                Case #3:00:00 PM#

                                    dr1("3:00 PM") = studentfullName

                                Case #3:30:00 PM#

                                    dr1("3:30 PM") = studentfullName

                                Case #4:00:00 PM#

                                    dr1("4:00 PM") = studentfullName

                                Case #4:30:00 PM#

                                    dr1("4:30 PM") = studentfullName

                                Case #5:00:00 PM#

                                    dr1("5:00 PM") = studentfullName

                                Case #5:30:00 PM#

                                    dr1("5:30 PM") = studentfullName

                                Case #6:00:00 PM#

                                    dr1("6:00 PM") = studentfullName

                                Case #6:30:00 PM#

                                    dr1("6:30 PM") = studentfullName

                                Case #7:00:00 PM#

                                    dr1("7:00 PM") = studentfullName

                                Case #7:30:00 PM#

                                    dr1("7:30 PM") = studentfullName

                            End Select

                        Next

                    Next

                Next

            Next

            'Place all days when the Clinician will be out into the table

            Dim foundrow() As DataRow = dt4.Select("Clinicianid='" & clid.Trim & "'") 'Locate the clinician within the clinicianOut table

            Dim tm1, tm2 As DateTime

For y1 = 0 To foundrow.Count – 1

'Iterate through the number of times the current clinician is returned

                tm1 = foundrow(y1)("TimeIn")

                tm2 = foundrow(y1)("TimeOut")

                t1 = tm1.ToString("h:mm tt")

                t2 = tm2.ToString("h:mm tt")

                timeintervals = intervals.timeIntervals(t1, t2) 'Return all time intervals within the time range found on the current row of the specific clinician

                Dim max As Integer = timeintervals.Count - 1 'Set final interval

                through the entire timeinterval range less 1.(Less 1 due to display range.)

                    slot = timeintervals(z1)

                    'Locate the time slot in which the will be out in each iteration of each 30min slot within the timeinterval range

'OUT will be placed in the time slot that is currently being located.

                    Select Case slot

                        Case #7:30:00 AM#

                            dr1("7:30 AM") = "           " & "OUT"

                        Case #8:00:00 AM#

                            dr1("8:00 AM") = "           " & "OUT"

                        Case #8:30:00 AM#

                            dr1("8:30 AM") = "           " & "OUT"

                        Case #9:00:00 AM#

                            dr1("9:00 AM") = "           " & "OUT"

                        Case #9:30:00 AM#

                            dr1("9:30 AM") = "           " & "OUT"

                        Case #10:00:00 AM#

                            dr1("10:00 AM") = "           " & "OUT"

                        Case #10:30:00 AM#

                            dr1("10:30 AM") = "           " & "OUT"

                        Case #11:00:00 AM#

                            dr1("11:00 AM") = "           " & "OUT"

                        Case #11:30:00 AM#

                            dr1("11:30 AM") = "           " & "OUT"

                        Case #12:00:00 PM#

                            dr1("12:00 PM") = "           " & "OUT"

                        Case #12:30:00 PM#

                            dr1("12:30 PM") = "           " & "OUT"

                        Case #1:00:00 PM#

                            dr1("1:00 PM") = "           " & "OUT"

                        Case #1:30:00 PM#

                            dr1("1:30 PM") = "           " & "OUT"

                        Case #2:00:00 PM#

                            dr1("2:00 PM") = "           " & "OUT"

                        Case #2:30:00 PM#

                            dr1("2:30 PM") = "           " & "OUT"

                        Case #3:00:00 PM#

                            dr1("3:00 PM") = "           " & "OUT"

                        Case #3:30:00 PM#

                            dr1("3:30 PM") = "           " & "OUT"

                        Case #4:00:00 PM#

                            dr1("4:00 PM") = "           " & "OUT"

                        Case #4:30:00 PM#

                            dr1("4:30 PM") = "           " & "OUT"

                        Case #5:00:00 PM#

                            dr1("5:00 PM") = "           " & "OUT"

                        Case #5:30:00 PM#

                            dr1("5:30 PM") = "           " & "OUT"

                        Case #6:00:00 PM#

                            dr1("6:00 PM") = "           " & "OUT"

                        Case #6:30:00 PM#

                            dr1("6:30 PM") = "           " & "OUT"

                        Case #7:00:00 PM#

                            dr1("7:00 PM") = "           " & "OUT"

                        Case #7:30:00 PM#

                            dr1("7:30 PM") = "           " & "OUT"

                    End Select

                Next

            Next

            dt1.Rows.Add(dr1) 'add the row information to the dataset

        Next 'Return to get next clinician

        Return ds

    End Function

End Class

Public Class ScheduleConfig

    Inherits ScheduleTemplate

    Dim scheduleResults As New Schedule

    Dim clinicianinfo As New Clinicians

    Public Overloads Function StudentSpecificSchedule(ByVal studentid As String) As DataSet

        Dim ds, ds1 As New DataSet

        Dim date1 As Date

        Dim StartTime As DateTime

        Dim EndTime As DateTime

        ds1 = StudentSchedule()

        ds = scheduleResults.ReturnStudentScheduleinfo(studentid)

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

        Dim dt1 As DataTable = ds1.Tables("Clinicianinfo")

        Dim row As DataRow

        Dim row1 As DataRow

        For Each row In dt.Rows

            date1 = row("Date")

            StartTime = row("TimeIn")

            EndTime = row("TimeOut")

            row1 = dt1.NewRow()

            row1("Date") = date1.ToString("dddd,  M/dd/yyyy")

            row1("Timein") = StartTime.ToString("h:mm tt")

            row1("Timeout") = EndTime.ToString("h:mm tt")

            row1("Clinician") = row("ClinicianSignature")

            dt1.Rows.Add(row1)

        Next

        Return ds1

    End Function

    Public Overloads Function StudentSpecificSchedule(ByVal studentid As String, ByVal index As Integer, ByVal StartDate As Date, ByVal EndDate As Date, ByVal Clinician As String) As DataSet

        Dim ds, ds1 As New DataSet

        Dim date1 As Date

        Dim StartTime As DateTime

        Dim EndTime As DateTime

        ds1 = StudentSchedule()

        If index = 0 Then 'Search by Date

            ds = scheduleResults.ReturnStudentScheduleinfo(Clinician)

        ElseIf index = 1 Then 'Search by clinician

        End If

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

        Dim dt1 As DataTable = ds1.Tables("Clinicianinfo")

        Dim row As DataRow

        Dim row1 As DataRow

        For Each row In dt.Rows

            date1 = row("Date")

            StartTime = row("TimeIn")

            EndTime = row("TimeOut")

            row1 = dt1.NewRow()

            row1("Date") = date1.ToString("dddd,  M/dd/yyyy")

            row1("Timein") = StartTime.ToString("h:mm tt")

            row1("Timeout") = EndTime.ToString("h:mm tt")

            row1("Clinician") = row("ClinicianSignature")

            dt1.Rows.Add(row1)

        Next

        Return ds1

    End Function

    Public Overloads Function ClinicianSchedule(ByVal Clinician As String, ByVal date1 As Date, ByVal date2 As Date) As DataSet

        Dim convertStudentName As INameConversion = New StudentNameconversion

        Dim dsClinicianScheduleTable As DataSet

        Dim ClinicianScheduleTable As DataTable

        dsClinicianScheduleTable = ClinicianStudentSchedule()

        If Clinician <> String.Empty Then

            ClinicianScheduleTable = dsClinicianScheduleTable.Tables("ClinicianSchedule")

            Dim StudentFullName As String

            Dim Student As New ArrayList

            Dim id As String

            Dim schdate As DateTime

            Dim t1, t2 As DateTime

            Dim ds As New DataSet

            ds = scheduleResults.ViewSchedule(Clinician.Trim, date1, date2)

            Dim dt As DataTable

            Dim studentdata As New returnStudentData

            dt = ds.Tables("MainSchedule")

            Dim finalrow As Integer = dt.Rows.Count

            Dim x As Integer = -1

            Dim dr1 As DataRow

            Dim dr = dt.Rows

            For x = 0 To finalrow - 1

                dr1 = ClinicianScheduleTable.NewRow()

                id = dr(x)("StudentID")

                schdate = dr(x)("Date")

                t1 = dr(x)("TimeIn")

                t2 = dr(x)("Timeout")

                StudentFullName = convertStudentName.convertName(id)

                dr1("Clinician") = dr(x)("ClinicianSignature")

                dr1("Student") = StudentFullName

                dr1("Scheduled Date") = schdate.ToString("dddd, M/dd/yyyy")

                dr1("Start") = t1.ToString("h:mm tt")

                dr1("Finish") = t2.ToString("h:mm tt")

                ClinicianScheduleTable.Rows.Add(dr1)

            Next

        End If

        Return dsClinicianScheduleTable

    End Function

    Public Overloads Function ClinicianSchedule(ByVal Clinician As String) As DataSet

        Dim convertStudentName As INameConversion = New StudentNameconversion

        Dim dsClinicianScheduleTable As DataSet

        Dim ClinicianScheduleTable As DataTable

        dsClinicianScheduleTable = ClinicianStudentSchedule()

        If Clinician <> String.Empty Then

            ClinicianScheduleTable = dsClinicianScheduleTable.Tables("ClinicianSchedule")

            Dim StudentFullName As String

            Dim Student As New ArrayList

            Dim id As String

            Dim schdate As DateTime = Nothing

            Dim t1 As DateTime

            Dim t2 As DateTime

            Dim ds As New DataSet

            Dim Classroominfo As New Schedule

            ds = scheduleResults.ViewSchedule(Clinician.Trim)

            Dim dt As DataTable

            Dim studentdata As New returnStudentData

            dt = ds.Tables("MainSchedule")

            Dim finalrow As Integer = dt.Rows.Count

            Dim x As Integer = -1

            Dim dr1 As DataRow

            Dim dr As DataRow

            Dim dr2 As DataRow

            Dim ds2 As New DataSet

            Dim dt2 As DataTable = Nothing

            Dim count As Integer = 0

            For Each dr In dt.Rows

                dr1 = ClinicianScheduleTable.NewRow()

                id = dr("StudentID")

                count = dr("Count")

                ds2 = Classroominfo.GetClassroomData(count)

                dt2 = ds2.Tables("ClassRoom")

                schdate = dr("Date")

                t1 = Convert.ToDateTime(dr("Timein"))

                t2 = Convert.ToDateTime(dr("Timein"))

                'Student = studentdata.GetStudentInfo(id.Trim)

                StudentFullName = convertStudentName.convertName(id)

                dr1("Student") = StudentFullName

                dr1("Scheduled Date") = schdate.ToString("dddd, M/dd/yyyy")

                dr1("Start") = t1.ToString("h:mm tt")

                dr1("Finish") = t2.ToString("h:mm tt")

                'Search the Classroom table for corresponding count id that matches the student

                If dt2.Rows.Count > 0 Then

                    For Each dr2 In dt2.Rows

                        dr1("Campus") = dr2("Campus")

                        dr1("Subject") = dr2("Subject")

                    Next

                End If

                ClinicianScheduleTable.Rows.Add(dr1)

            Next

        End If

        Return dsClinicianScheduleTable

    End Function

    Public Function clinicianOutSchedule(ByVal Clinicainid As String, ByVal d1 As String, ByVal d2 As String) As DataTable

        'Go search for clinician name and place it into the table

        Dim ds As New DataSet

        Dim dsout As DataSet

        Dim date1, date2 As DateTime

        date1 = Convert.ToDateTime(d1)

        date2 = Convert.ToDateTime(d2)

        dsout = clinicianinfo.clinicianOutSchedule(Clinicainid, date1, date2)

        Dim dtout As DataTable = dsout.Tables("Clinician\_DailyOutSchedule")

        Dim dsClinicianScheduleTable As DataSet

        Dim dtClinicianSchedule As DataTable

        dsClinicianScheduleTable = OffSchedule()

        Dim dv As New DataView(dtout)

        dv.Sort = "Date,TimeIn ASC"

        dtClinicianSchedule = dsClinicianScheduleTable.Tables("Clinicianoff")

        Dim dr1 As DataRow

        Dim dr = dtout.Rows

        Dim offdate As Date

        Dim offtime1 As DateTime

        Dim offtime2 As DateTime

        For Each rowView As DataRowView In dv

            Dim row As DataRow = rowView.Row

            dr1 = dtClinicianSchedule.NewRow()

            offdate = row("Date")

            offtime1 = row("TimeIn")

            offtime2 = row("TimeOut")

            dr1("Date") = offdate.ToString("dddd,  M/dd/yyyy")

            dr1("From") = offtime1.ToString("h:mm tt")

            dr1("To") = offtime2.ToString("h:mm tt")

            dtClinicianSchedule.Rows.Add(dr1)

        Next

        Return dtClinicianSchedule

    End Function

    Public Overloads Function ReturnallStudentDailySchedule(ByVal Studentid As String, ByVal StartDate As DateTime, ByVal EndDate As DateTime) As DataSet

        Dim ds, ds1 As New DataSet

        Dim convertStudentName As INameConversion = New StudentNameconversion

        Dim schedule As New Schedule

        Dim studentname, Status As String

        Dim date1 As Date

        Dim StartTime As DateTime

        Dim EndTime As DateTime

        ds1 = ClinicianStudentSchedule()

        ds = schedule.ReturnStudentScheduleinfo(Studentid.Trim, StartDate, EndDate)

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

        Dim dt1 As DataTable = ds1.Tables("ClinicianSchedule")

        Dim row As DataRow

        Dim row1 As DataRow

        For Each row In dt.Rows

            date1 = row("Date")

            StartTime = row("TimeIn")

            EndTime = row("TimeOut")

            Studentid = row("Studentid")

            Status = row("Status")

            studentname = convertStudentName.convertName(Studentid)

            row1 = dt1.NewRow()

            row1("Clinician") = row("ClinicianSignature")

            row1("Student") = studentname

            row1("Scheduled Date") = date1.ToString("dddd M/dd/yyyy")

            row1("Start") = StartTime.ToString("h:mm tt")

            row1("Finish") = EndTime.ToString("h:mm tt")

            row1("Status") = Status.Trim

            dt1.Rows.Add(row1)

        Next

        Return ds1

    End Function

    Public Overloads Function DisplayStudentData() As DataSet

        Dim Firstname As String = Nothing

        Dim Lastname As String = Nothing

        Dim DateofBirth As Date

        Dim dob As String = Nothing

        Dim Gender As String = [String].Empty

        Dim initialInquiry As Date

        Dim ii As String = [String].Empty

        Dim rd As String = [String].Empty

        Dim reportdiscussion As Date

        Dim Assessment As Date

        Dim assess As String = [String].Empty

        Dim SchoolDist As String = [String].Empty

        Dim school As String = [String].Empty

        Dim tutoringstartdate As Date = Nothing

        Dim tutoringstopdate As Date = Nothing

        Dim tutstart As String = [String].Empty

        Dim tutstop As String = [String].Empty

        Dim activestudent As String = [String].Empty

        Try

            Dim Data As New returnStudentData

            Dim ds As New DataSet

            Dim ds1 As New DataSet

            ds = Data.GetStudentInfo()

            ds1 = StudentProfile()

            Dim table1 As DataTable = ds.Tables("StudentProfile")

            Dim table2 As DataTable = ds1.Tables("StudentProfileData")

            Dim dr As DataRow

            Dim dr1 As DataRow

            Dim x As Integer = table1.Rows.Count - 1

            For Each dr In table1.Rows

                Firstname = dr("First Name").ToString

                Lastname = dr("Last Name").ToString

                Gender = dr("Gender").ToString

                SchoolDist = dr("District Zone").ToString

                school = dr("School Attending").ToString

                If IsDBNull(dr("DOB")) = True Then

                    DateofBirth = Nothing

                    dob = ""

                Else

                    DateofBirth = dr("DOB")

                    dob = DateofBirth.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Initial Inquiry Date")) = True Then

                    initialInquiry = Nothing

                    ii = [String].Empty

                Else

                    initialInquiry = dr("Initial Inquiry date")

                    ii = initialInquiry.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Assessment Date")) = True Then

                    Assessment = Nothing

                    assess = [String].Empty

                Else

                    Assessment = dr("Assessment Date")

                    assess = Assessment.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Report Discussion Date")) = True Then

                    rd = Nothing

                    rd = [String].Empty

                Else

                    reportdiscussion = dr("Report Discussion Date")

                    rd = reportdiscussion.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Tutoring Start Date")) = True Then

                    tutoringstartdate = Nothing

                    tutstart = ""

                Else

                    tutoringstartdate = dr("Tutoring Start Date")

                    tutstart = tutoringstartdate.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Tutoring Stop Date")) = True Then

                    tutoringstopdate = Nothing

                    tutstop = ""

                Else

                    tutoringstopdate = dr("Tutoring Stop Date")

                    tutstop = tutoringstopdate.ToString("M/dd/yyyy")

                End If

                activestudent = dr("Active")

                dr1 = table2.NewRow()

                dr1("FirstName") = Firstname

                dr1("LastName") = Lastname

                dr1("DOB") = dob

                dr1("Gender") = Gender

                dr1("Initial Inquiry") = ii

                dr1("Assessment") = assess

                dr1("Report Discussion") = rd

                dr1("School Dist") = SchoolDist

                dr1("School") = school

                dr1("Tutoring Start") = tutstart

                dr1("Tutoring Stop") = tutstop

                If activestudent = True Then

                    dr1("Active") = "YES"

                ElseIf activestudent = False Then

                    dr1("Active") = "NO"

                End If

                table2.Rows.Add(dr1)

            Next

            Return ds1

        Catch ex As Exception

            Throw ex

        End Try

    End Function

    Public Overloads Function DisplayStudentData(ByVal StudentFullname As String, ByVal Key As Integer) As DataSet

        Dim convertStudentName As INameConversion = New StudentNameconversion

        Dim Firstname As String = String.Empty

        Dim Lastname As String = String.Empty

        Dim DateofBirth As Date

        Dim dob As String = String.Empty

        Dim Gender As String = String.Empty

        Dim Initialinquiry As Date = Nothing

        Dim ReportDiscussion As Date = Nothing

        Dim Assessment As Date = Nothing

        Dim assess As String = String.Empty

        Dim ii As String = [String].Empty

        Dim rd As String = [String].Empty

        Dim SchoolDist As String = [String].Empty

        Dim school As String = [String].Empty

        Dim tutoringstartdate As Date = Nothing

        Dim tutoringstopdate As Date = Nothing

        Dim tutstart As String = [String].Empty

        Dim tutstop As String = [String].Empty

        Dim Studentid As String = [String].Empty

        Dim splitName() As String = Nothing

        Dim ActiveStudent As String = [String].Empty

        Try

            Dim Data As New returnStudentData

            Dim ds As New DataSet

            Dim ds1 As New DataSet

            Studentid = convertStudentName.convertToId(StudentFullname)

            Dim lname As String

            splitName = StudentFullname.Split(",")

            lname = splitName(0).Trim

            ds = Data.GetStudentInfo(lname, Key)

            ds1 = StudentProfile()

            Dim table1 As DataTable = ds.Tables("StudentProfile")

            Dim table2 As DataTable = ds1.Tables("StudentProfileData")

            Dim dr As DataRow

            Dim dr1 As DataRow

            For Each dr In table1.Rows

                Firstname = dr("First Name").ToString

                Lastname = dr("Last Name").ToString

                Gender = dr("Gender").ToString

                SchoolDist = dr("District Zone").ToString

                school = dr("School Attending").ToString

                If IsDBNull(dr("DOB")) = True Then

                    DateofBirth = Nothing

                    dob = ""

                Else

                    DateofBirth = dr("DOB")

                    dob = DateofBirth.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Initial Inquiry Date")) = True Then

                    Initialinquiry = Nothing

                    ii = [String].Empty

                Else

                    Initialinquiry = dr("Initial Inquiry Date")

                    ii = Initialinquiry.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Report Discussion Date")) = True Then

                    ReportDiscussion = Nothing

                    rd = [String].Empty

                Else

                    ReportDiscussion = dr("Report Discussion Date")

                    rd = ReportDiscussion.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Assessment Date")) = True Then

                    Assessment = Nothing

                    assess = [String].Empty

                Else

                    Assessment = dr("Assessment Date")

                    assess = Assessment.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Tutoring Start Date")) = True Then

                    tutoringstartdate = Nothing

                    tutstart = ""

                Else

                    tutoringstartdate = dr("Tutoring Start Date")

                    tutstart = tutoringstartdate.ToString("M/dd/yyyy")

                End If

                If IsDBNull(dr("Tutoring Stop Date")) = True Then

                    tutoringstopdate = Nothing

                    tutstop = ""

                Else

                    tutoringstopdate = dr("Tutoring Stop Date")

                    tutstop = tutoringstopdate.ToString("M/dd/yyyy")

                End If

                ActiveStudent = dr("Active")

                dr1 = table2.NewRow()

                dr1("FirstName") = Firstname

                dr1("LastName") = Lastname

                dr1("DOB") = dob

                dr1("Initial Inquiry") = ii

                dr1("Report Discussion") = rd

                dr1("Gender") = Gender

                dr1("Assessment") = assess

                dr1("School Dist") = SchoolDist

                dr1("School") = school

                dr1("Tutoring Start") = tutstart

                dr1("Tutoring Stop") = tutstop

                If ActiveStudent = True Then

                    dr1("Active") = "YES"

                ElseIf ActiveStudent = False Then

                    dr1("Active") = "NO"

                End If

                table2.Rows.Add(dr1)

            Next

            Return ds1

        Catch ex As Exception

            Throw ex

        End Try

    End Function

End Class