Imports System

Imports System.Data

Imports System.Data.SqlClient

Imports System.IO

Imports System.Configuration

'Fetch a users specific attribute by returning it as a datatable or an array collection

Public Interface IstudentAttributesDatasets

    Function RetrieveNotes(ByVal Studentid As String) As DataView

    Function RetrieveNotes(ByVal Studentid As String, ByVal dateid As String, ByVal noteheader As String) As DataView

    Function CheckIfNoteExsist(ByVal Studentid As String, ByVal Date1 As Date, ByVal noteheader As String) As Boolean

    Function RetrieveStudentSchool(ByVal Studentid As String) As DataTable

    Function StudentInfo(ByVal studentId As String) As DataTable

    Function Studentinfo(ByVal fn As String, ByVal ln As String) As String

End Interface

Public Interface IguardianAttributes

    Function GuardianInfo(ByVal firstname As String, lastname As String) As String

End Interface

Public Interface IstudentAttributesCollection

    Function StudentInfo(ByVal studentId As String) As ArrayList

    Function StudentInfo() As Array

End Interface

Public Interface IclinicianAttributes

    Function getContactInformation(ByVal clinicianId As String) As ArrayList

End Interface

Public Class userProfileAttributes

    Implements IstudentAttributesDatasets

    Dim connectionString As Object

    Public Sub New()

        Try

            connectionString = ConfigurationManager.ConnectionStrings("Familyliteracy").ConnectionString

        Catch ex As SqlException

            MsgBox("Cannot connect to Datasbase!")

        End Try

    End Sub

    Public Function RetrieveNotes(ByVal Studentid As String) As DataView Implements IstudentAttributesDatasets.RetrieveNotes

        Dim index As Integer

        index = Convert.ToInt16(Studentid)

        Dim query As String = "Select \* From StudentNote Where Studentid='" & Studentid & "'"

        Dim conn As New SqlConnection(connectionString)

        Dim cmd As New SqlCommand(query, conn)

        Dim da As New SqlDataAdapter(cmd)

        Dim ds As New DataSet

        conn.Open()

        da.Fill(ds, "StudentNote")

        conn.Close()

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

        Dim dv As New DataView(dt)

        dv.Sort = "RemarkDate DESC"

        Return dv

    End Function

Public Function RetrieveNotes(ByVal Studentid As String, ByVal dateid As String, ByVal noteheader As String) As DataView Implements IstudentAttributesDatasets.RetrieveNotes

        Dim index As Integer

        index = Convert.ToInt16(Studentid)

        Dim cnvdate As Date

        cnvdate = Convert.ToDateTime(dateid)

        Dim query As String = "Select \* From StudentNote Where Studentid='" & index & "' And RemarkDate='" & cnvdate & "' AND RemarkHeader='" & noteheader & "'"

        Dim conn As New SqlConnection(connectionString)

        Dim cmd As New SqlCommand(query, conn)

        Dim da As New SqlDataAdapter(cmd)

        Dim ds As New DataSet

        conn.Open()

        da.Fill(ds, "StudentNote")

        conn.Close()

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

        Dim dv As New DataView(dt)

        dv.Sort = "RemarkDate DESC"

        Return dv

    End Function

    Public Function CheckIfNoteExsist(ByVal Studentid As String, ByVal Date1 As Date, ByVal noteheader As String) As Boolean Implements IstudentAttributesDatasets.CheckIfNoteExsist

        Dim index As Integer

        index = Convert.ToInt16(Studentid)

        Dim query1 As String = "SELECT \* FROM Studentnote where Studentid='" & index & "' AND RemarkHeader='" & noteheader & "' AND RemarkDate='" & Date1 & "'"

        Dim conn1 As New SqlConnection(connectionString)

        Dim cmd1 As New SqlCommand(query1, conn1)

        Dim da1 As New SqlDataAdapter(cmd1)

        Dim ds1 As New DataSet

        conn1.Open()

        da1.Fill(ds1, "StudentNote")

        conn1.Close()

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

        Dim num As Integer = 0

        num = dt1.Rows.Count

        If num > 0 Then

            Return True

        Else

            Return False

        End If

    End Function

Public Function RetrieveStudentSchool(ByVal Studentid As String) As DataTable Implements IstudentAttributesDatasets.RetrieveStudentSchool

        Dim ds As New DataSet

        Dim index As Integer

        If Studentid <> String.Empty Then

            index = Convert.ToInt16(Studentid)

            Dim query As String = "Select \* From StudentSchool Where Studentid='" & index & "'"

            Dim conn As New SqlConnection(connectionString)

            Dim cmd As New SqlCommand(query, conn)

            Dim da As New SqlDataAdapter(cmd)

            conn.Open()

            da.Fill(ds, "StudentSchool")

            conn.Close()

        End If

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

        Return dt

    End Function

Public Function Studentinfo(ByVal fn As String, ByVal ln As String) As String Implements IstudentAttributesDatasets.StudentInfo

        Dim student As String = String.Empty

        If fn <> String.Empty And ln <> String.Empty Then

            Dim query As String = "SELECT \* FROM StudentProfile where [First Name]='" & fn & "'" & " AND [Last Name]='" & ln & "'"

            Dim conn As New SqlConnection(connectionString)

            Dim cmd As New SqlCommand(query, conn)

            Dim da As New SqlDataAdapter(cmd)

            Dim ds As New DataSet

            conn.Open()

            da.Fill(ds, "StudentProfile")

            conn.Close()

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

            Dim rw As DataRow

            For Each rw In dt.Rows

                student = rw("StudentId")

            Next

        Else

            student = String.Empty

        End If

        student = Convert.ToString(student)

        Return student

    End Function

Public Function StudentInfo(ByVal studentId As String) As DataTable Implements IstudentAttributesDatasets.StudentInfo

        Dim index As Integer

        index = Convert.ToInt16(studentId)

        Dim dt As DataTable

        Dim query As String = "SELECT \* FROM StudentProfile Where StudentId='" & index & "' ORDER BY [Last Name] ASC, [First Name] ASC"

        Dim conn As New SqlConnection(connectionString)

        Dim cmd As New SqlCommand(query, conn)

        Dim da As New SqlDataAdapter(cmd)

        Dim ds As New DataSet

        conn.Open()

        da.Fill(ds, "StudentProfile")

        conn.Close()

        dt = ds.Tables("StudentProfile")

        Return dt

    End Function

End Class

Public Class userAttributesCollection

    Implements IstudentAttributesCollection

    Dim connectionString As Object

    Public Sub New()

        Try

            connectionString = ConfigurationManager.ConnectionStrings("Familyliteracy").ConnectionString

        Catch ex As Exception

            MsgBox("Cannot connect to Datasbase!")

        End Try

    End Sub

    Public Function StudentInfo() As Array Implements IstudentAttributesCollection.StudentInfo

        Dim query As String = "SELECT \* FROM StudentProfile ORDER BY [Last Name]"

        Dim conn As New SqlConnection(connectionString)

        Dim cmd As New SqlCommand(query, conn)

        Dim da As New SqlDataAdapter(cmd)

        Dim ds As New DataSet

        conn.Open()

        da.Fill(ds, "StudentProfile")

        conn.Close()

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

        Dim x As Integer = 0

        Dim rw As DataRow

        Dim max As Integer = dt.Rows.Count - 1

        Dim student\_attributes(max, 7) As String

        For Each rw In dt.Rows

            student\_attributes(x, 0) = rw("First Name").ToString

            student\_attributes(x, 1) = rw("Last Name").ToString

            student\_attributes(x, 2) = rw("DOB").ToString

            student\_attributes(x, 3) = rw("Gender").ToString

            student\_attributes(x, 4) = rw("District Zone").ToString

            student\_attributes(x, 5) = rw("School Attending").ToString

            student\_attributes(x, 6) = rw("Active").ToString

            x = x + 1

        Next

        Return student\_attributes

    End Function

Public Function StudentInfo(ByVal studentId As String) As ArrayList Implements IstudentAttributesCollection.StudentInfo

        Dim index As Integer

        index = Convert.ToInt16(studentId)

        Dim query As String = "SELECT \* FROM StudentProfile WHERE StudentId='" & index & "' ORDER BY [Last Name]"

        Dim conn As New SqlConnection(connectionString)

        Dim cmd As New SqlCommand(query, conn)

        Dim da As New SqlDataAdapter(cmd)

        Dim ds As New DataSet

        conn.Open()

        da.Fill(ds, "StudentProfile")

        conn.Close()

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

        Dim x As Integer = 0

        Dim rw As DataRow

        Dim max As Integer = dt.Rows.Count - 1

        Dim student As New ArrayList

        Dim dateofBirth As String = String.Empty

        Dim initialInquiry As String = String.Empty

        Dim assessment As String = String.Empty

        Dim reportDiscussion As String = String.Empty

        Dim tutorStartDate As String = String.Empty

        Dim tutorStopDate As String = String.Empty

        For Each rw In dt.Rows

            student.Add(rw("First Name").ToString)

            student.Add(rw("Last Name").ToString)

            If (IsDate(rw("DOB"))) Then

                dateofBirth = DateTime.Parse(rw("DOB")).ToString("MM/dd/yyyy")

            End If

            student.Add(dateofBirth)

            student.Add(rw("Gender").ToString)

            student.Add(rw("District Zone").ToString)

            student.Add(rw("School Attending").ToString)

            If (IsDate(rw("Initial Inquiry Date"))) Then

initialInquiry = DateTime.Parse(rw("Initial Inquiry Date")).ToString("MM/dd/yyyy")

            End If

            student.Add(initialInquiry)

            If (IsDate(rw("Assessment Date"))) Then

                assessment = DateTime.Parse(rw("Assessment Date")).ToString("MM/dd/yyyy")

            End If

            student.Add(assessment)

            If (IsDate(rw("Report Discussion Date"))) Then

reportDiscussion = DateTime.Parse(rw("Report Discussion Date")).ToString("MM/dd/yyyy")

            End If

student.Add(reportDiscussion)

            If (IsDate(rw("Tutoring Start Date"))) Then

tutorStartDate = DateTime.Parse(rw("Tutoring Start Date")).ToString("MM/dd/yyyy")

            End If

            student.Add(tutorStartDate)

            If (IsDate(rw("Tutoring Stop Date"))) Then

tutorStopDate = DateTime.Parse(rw("Tutoring Stop Date")).ToString("MM/dd/yyyy")

            End If

            student.Add(tutorStopDate)

            student.Add(rw("Active").ToString)

        Next

        Return student

    End Function

End Class

Public Class clinicianInfo

    Inherits Clinicians

    Implements IclinicianAttributes

Public Function getContactInformation(clinicianId As String) As ArrayList Implements IclinicianAttributes.getContactInformation

        Dim dtcontactinfo As DataTable

        Dim contactinfo As New ArrayList

        dtcontactinfo = ClinicianProfile(clinicianId, False)

        Dim row As DataRow

        For Each row In dtcontactinfo.Rows

            contactinfo.Add(row("Email"))

            contactinfo.Add(row("Phone"))

            contactinfo.Add(row("cellular"))

            contactinfo.Add(row("Alt Phone"))

        Next

        Return contactinfo

    End Function

End Class

Public Class guardianInfo

    Implements IguardianAttributes

    Dim connectionString As Object

    Public Sub New()

        Try

            connectionString = ConfigurationManager.ConnectionStrings("Familyliteracy").ConnectionString

        Catch ex As Exception

            MsgBox("Cannot connect to Datasbase!")

        End Try

    End Sub

    Public Function GuardianInfo(ByVal firstname As String, lastname As String) As String Implements IguardianAttributes.GuardianInfo

        Dim Guardian As String = String.Empty

        Dim query As String = "SELECT \* FROM GuardianProfile where [First Name]='" & firstname & "'" & " AND [Last Name]='" & lastname & "'"

        Dim conn As New SqlConnection(connectionString)

        Dim cmd As New SqlCommand(query, conn)

        Dim da As New SqlDataAdapter(cmd)

        Dim ds As New DataSet

        conn.Open()

        da.Fill(ds, "GuardianProfile")

        conn.Close()

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

        Dim rw As DataRow

        For Each rw In dt.Rows

            Guardian = Convert.ToString(rw("GuardianId"))

        Next

        Return Guardian

    End Function

End Class