Imports BAL

Imports DAL

Imports System.ComponentModel

Public Class HomeDisplay

    Implements IMessageFilter

    Public Sub New()

        InitializeComponent()

        Application.AddMessageFilter(Me)

    End Sub

    'Sign-in timer

    Public Function PreFilterMessage(ByRef m As Message) As Boolean Implements IMessageFilter.PreFilterMessage

Retrigger timer on keyboard and mouse for 30 minutes

        If (m.Msg >= &H100 And m.Msg <= &H109) Or (m.Msg >= &H200 And m.Msg <= &H20E) And Button11.Enabled = False Then

            Timer2.Stop()

            Timer2.Start()

        End If

        Return Nothing

    End Function

    'Setup initial display

    Private Sub Display\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        HomeScreen(Today)

        Dim newWidth As Integer

        Dim names As IPopulateAllNames = New IPopulateNames

        Dim Displaydata As New ScheduleConfig

        Dim ds As New DataSet

        ds = Displaydata.DisplayStudentData

        Dim dsStudent As New DataSet

        dsStudent = names.DisplayStudents(True)

        Dim dtStudent As DataTable = dsStudent.Tables("StudentList")

        ComboBox1.DataSource = dtStudent

        ComboBox1.DisplayMember = "FullName"

        ComboBox1.ValueMember = "FullName"

        ComboBox1.SelectedIndex = 0

        Dim changeColumnSize As New StoreGridViewColumnWidth

        newWidth = changeColumnSize.ReturnColumnWidth(0, 1)

        ToolStripTextBox1.Text = newWidth.ToString

        For x = 1 To DataGridView1.Columns.Count - 1

            DataGridView1.Columns(x).Width = newWidth

        Next

        With Me.DataGridView1

            .SelectionMode = DataGridViewSelectionMode.RowHeaderSelect

            .MultiSelect = False

        End With

        'Refresh Display Screen

        Timer1.Enabled = True

        logOffUser()

        ToolStripMenuItem1.Enabled = False

        AddRemoveAppointmentToolStripMenuItem.Enabled = False

        DailyScheduleToolStripMenuItem.Enabled = False

    End Sub

    Public Function HomeScreen(ByVal Startupdate As Date)

        Dim clinicianinfo As New Clinicians

        Dim convertApostrophe As New nameOperation

        Dim dsclinicians As New DataSet

Inactive clinicans are clinician who have left the institute and will not be coming back in the future.

Active clinicians are those clinicians who are currently working or are not working for an extended peiod of time.

Return all active Clinicians. Passing TRUE value looks for all active clinicians and returns all active clinicians.

If one were to choose false it would return all inactive  clinicians

        dsclinicians = clinicianinfo.GetClinicianInfo(True)

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

        Dim clinicianFirstName As String = String.Empty

        Dim clinicianLastName As String = String.Empty

        Dim row As DataRow

Populate columnheaders of the DataGridView control with all the Clinicians that have returned as active

        For Each row In dtClinicians.Rows

            clinicianLastName = row("LastName")

            clinicianFirstName = row("FirstName")

            Dim clinicianFullName As String = clinicianLastName.Trim & ", " & clinicianFirstName.Trim

            clinicianFullName = convertApostrophe.executeName(clinicianFullName, 0)

            DataGridView1.Columns.Add(clinicianFullName, clinicianFullName)

        Next

Gererate DataGrid Display Layout, and Populate it

        DisplaySetup(Startupdate)

        DataGridView1.SelectionMode = DataGridViewSelectionMode.CellSelect

        DataGridView1.AllowDrop = True

        DataGridView1.AllowUserToResizeColumns = True

        Return Nothing

    End Function

    Public Function RemoveColumns()

        Me.DataGridView1.Columns.Clear()

        Return Nothing

    End Function

    Private Sub MonthCalendar1\_DateChanged(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DateRangeEventArgs) Handles MonthCalendar1.DateChanged

        Dim Schedule\_Date As Date

        Schedule\_Date = MonthCalendar1.SelectionStart

        Removerows(Schedule\_Date)

    End Sub

Setup the column that list all 25 time intervals between 7:30 AM to 6:00 PM

Dataset ds will store all the students in their respective time slots.

Dataset ds will store all the students in their respective time slots.

Get Selected Calendar value from the CalendarControl

The line below is necessary in order to attain other attributes associated with each student such as classroom, and Campus information

Return each clincian

Iterate through each Clinician listed in each header column of the DataGrid

Check to see if the current Clinician matches the Clinician name Listed in the Header Column of the DataGrid control

Iterate through all the rows/labeled time intervals

Store current time calue from the first column of the GridView Control

Get the first student name at this time interval that is stored in the dataset

Check to see if clinician is off at the particular time interval by testing for an empty string.

Otherwise convert the students name to an thier identification number and place him in the respective cell in the Gridview

Get student id

Return additional information of the student

Check proposed (not rescheduled) hours that are hour 1 or 2

Check rescheduled hours that are hour 1 or 2

Determine the Color which corresponds to the students Status  Reschedule/ Transfer, Proposed,

Also, determine the students Attendance, No Show/Absent then set  the respective cell color

Also check if there will be a meeting, or Testing

Reset the student id and location

Check for the first two hours of the current student. If the  first two hours are found the flag it inthevariable called start hour

Check to see if the first two hours are triggered. If so then  color the cell white.

Check to see if this is a different student from the last  iteration.

Otherwise the location will show up in every cell/timeslot that corresponds to the student in the current iteration

rather than once.

Also check to see if student name appears on every hour that is scheduled.

Store the student name so the current student in the iteration  only appears once.

Public Sub DisplaySetup(ByVal CurrentDate As Date)

        Dim display As IDisplaySetup = New DisplayModule

        Dim convertname As INameConversion = New StudentNameconversion

        Dim Student As String = String.Empty

        Dim Stat As New Schedule

Setup the column that list all 25 time intervals Between 7:30 AM to 6:00 PM

        DisplayTemplate()

        Dim ds As New DataSet

The GridView control will mirror the dataset.

Dataset ds will store all the students in their respective time  slots.

        ds = display.mainDisplaySchedule(CurrentDate)

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

        Dim ds2 As New DataSet

        Dim ds3 As New DataSet

        Dim ds5 As New DataSet

        Dim studentid As String

        Dim nextname As String = Nothing

        Dim location As String = Nothing

Get Selected Calendar value from the CalendarControl

        Dim startdate As DateTime = MonthCalendar1.SelectionStart

        Dim FinalDate As DateTime = MonthCalendar1.SelectionStart

The line below is necessary in order to attain other attributes associated with each student such as classroom, and Campus information

        ds2 = Stat.GetSchedule(startdate, FinalDate)

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

        Dim b As Integer

        Dim Status As String = String.Empty

        Dim Subject As String = String.Empty

        Dim present As String = String.Empty

        Dim Clinician As String = String.Empty

        Dim ClinicianHeader As String = String.Empty

        Dim countid As String = String.Empty

Return each clincian

        For Each rw In dt.Rows

            Clinician = rw("Clinician")

            Dim numberofColumns As Integer = DataGridView1.ColumnCount

            Dim headercount As Integer = 0

Iterate through each Clinician listed in each header column of the DataGrid

            For headercount = 1 To numberofColumns - 1

                ClinicianHeader = DataGridView1.Columns(headercount).HeaderText

Check to see if the current Clinician matches the Clinician name Listed in the Header Column of the DataGrid control

                If ClinicianHeader.Trim = Clinician.Trim Then

                    Dim a As Integer = 0

                    Dim timein As String = String.Empty

                    Dim t1 As DateTime = Nothing

                    Dim query As String = String.Empty

                    Dim c As Integer = 0

iterate through all the rows/labeled time intervals

                    For a = 1 To 24

Store current time calue from the first column of the GridView  Control

                        timein = DataGridView1.Rows(a).Cells(0).Value

                        t1 = Convert.ToDateTime(timein.Trim).ToShortTimeString

Get the first student name at this time interval that is stored  in the dataset

                        Student = rw(a).ToString

Check to see if clinician is off at the particular time interval by testing for an empty string.

Otherwise convert the students name to an thier identification  number and place him in the respective cell in the Gridview

                        If Student = "           " & "OUT" Then

                            location = String.Empty

                            DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Green

                            If c = 0 Then

                                DataGridView1.Rows(a).Cells(headercount).Value = "           " & "OUT"

                                DataGridView1.Columns(headercount).DefaultCellStyle.Font = New Font("Times NewRoman", 8, FontStyle.Regular)

                            End If

                            c = c + 1

                        ElseIf Student.Trim <> String.Empty Then

Get student id

                            studentid = convertname.convertToId(Student.Trim)

                            Dim timestamp As String = String.Empty

                            timestamp = Convert.ToDateTime("1900-01-01 " & timein)

                            Dim time1 As DateTime

                            Dim time2 As DateTime

                            query = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

Return additional information of the student

                            Dim foundrow() As DataRow = dt2.Select(query)

                            b = 0

                            Do While b <= foundrow.Length - 1

                                Status = foundrow(b)("status")

                                countid = foundrow(b)("Count")

                                present = foundrow(b)("Attendance")

                                time1 = foundrow(b)("TimeIn")

                                time2 = foundrow(b)("TimeOut")

                                ds5 = Stat.GetClassroomData(countid.Trim)

                                Dim dt5 As DataTable = ds5.Tables("Classroom")

                                Dim subjectrow As DataRow

                                For Each subjectrow In dt5.Rows

                                    Subject = subjectrow("Subject")

                                    location = subjectrow("Campus")

                                Next

'Check for the first two hours of the current student. If the  first two hours are found the flag it inthevariable called start hour

                                Dim starthr As String = String.Empty

                                Dim gethr As String = String.Empty

'Check Proposed(not rescheduled) hours that are hour 1 or 2

                                Dim query2 = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

                                Dim d As Integer = 0

'Check rescheduled hours that are hour 1 or 2

                                Dim transferquery = "Studentid='" & studentid.Trim & "' AND Timein <='" & timestamp & "' AND TimeOut >='" & timestamp & "'"

Determine the Color which corresponds to the students Status Reschedule/Transfer, Proposed,

Also, Determine the students Attendance,No Show/Absent then set the respective cell color

Also, check if there will be a meeting, or Testing

                                REM Proposed=Yellow

                                REM Abset=Red

                                REM Completed=Blue

                                REM Transfer=Azure

                                REM Testing=Gray

                                REM Meeting=Orange

1 or Hour 2 = White regardless of other color conditions that may coincide

                                If present.Trim = "Absent" And (timestamp >= time1 Or timestamp <= time2) Then

                                    DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Red

                                ElseIf present.Trim = "Completed" Then

                                    DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Blue

                                ElseIf Status.Trim = "Transfer" Then

                                    DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Azure

                                ElseIf Status.Trim = "Proposed" And Subject.Trim = "Testing" Then

                                    DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Gray

                                ElseIf Status.Trim = "Proposed" And Subject.Trim = "Meeting" Then

                                    DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Orange

                                ElseIf present.Trim = "Proposed" And (timestamp >= time1 Or timestamp <= time2) Then

                                    DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.Yellow

                                End If

                                b = b + 1

                            Loop

                        ElseIf Student = String.Empty Then

Reset the studentid and location

                            studentid = [String].Empty

                            location = [String].Empty

                            DataGridView1.Rows(a).Cells(headercount).Style.BackColor = Color.AntiqueWhite

                        Else

                        End If

Check to see if this is a different student from the last  iteration.

Otherwise the location will show up in every cell/timeslot that corresponds to the student in the current iteration

rather than once.

Also, check to see if student name appears on every hour that is scheduled.

                        If nextname <> Student Or nextname = Student And b = 2 Then

                            If location = "Northwest" Or location = "NorthWest" Then

                                location = "NW="

                            ElseIf location.Trim = "Main" Then

                                location = "M="

                            Else

                                location = [String].Empty

                            End If

Concatenate the Location with the student name

                            DataGridView1.Rows(a).Cells(headercount).Value = location & Student

                            DataGridView1.Columns(headercount).DefaultCellStyle.Font = New Font("Times NewRoman", 8, FontStyle.Regular)

Store the student name so the current student in the iteration only appears once.

                            nextname = Student

                        End If

                    Next

                    Exit For

                Else

                End If

            Next

        Next

        DataGridView1.AllowDrop = True

        Dim x As Integer = 0

        Dim y As Integer = 0

        y = DataGridView1.ColumnCount - 1

        For x = 0 To y

            DataGridView1.Columns(x).SortMode = DataGridViewColumnSortMode.NotSortable

        Next

    End Sub

This is a Templat which populates the all 26 rows in the Firstcolumn of the DataGrid Control with all the time values between 7:30 AM to 6:00 PM

Set the Font size and Color

Add each row into the DataGrid Control throug interation

Public Sub DisplayTemplate()

This is a Template which Populates the all 26 rows in the Firstcolumn of the DataGrid Control with all the time values Between  7:30 AM to 6:00 PM

Set the Font size and Color

        DataGridView1.Columns(0).DefaultCellStyle.Font = New Font("Times NewRoman", 10, FontStyle.Regular)

        Dim i As Integer

        Dim timestamp As Array = {"Hour", "7:30 AM", "8:00 AM", "8:30 AM", "9:00 AM", "9:30 AM", "10:00 AM", "10:30 AM", "11:00 AM", "11:30 AM", "12:00 PM", "12:30 PM",

                                  "1:00 PM", "1:30 PM", "2:00 PM", "2:30 PM", "3:00 PM", "3:30 PM", "4:00 PM", "4:30 PM", "5:00 PM", "5:30 PM", "6:00 PM",

                                 "6:30 PM", "7:00 PM", "7:30 PM"}

Add each row into the Data Grid Control through iteration

        For i = 0 To 25

            Dim dgvRow As New DataGridViewRow

            Dim dgvCell As DataGridViewCell

            dgvCell = New DataGridViewTextBoxCell()

            dgvCell.Value = timestamp(i)

            dgvRow.Cells.Add(dgvCell)

            DataGridView1.Rows.Add(dgvRow)

            DataGridView1.Rows(i).Cells(0).Style.BackColor = Color.WhiteSmoke

        Next

    End Sub

Remove every Row in the GridView Control

    Public Sub Removerows(ByVal Schedule\_Date As Date)

        REM Remove every Row in the GridView Control

        Dim w As Integer = DataGridView1.Rows.Count

        If DataGridView1.Rows.Count > 0 Then

            For b = 0 To DataGridView1.RowCount - 1

                DataGridView1.Rows.RemoveAt(0)

            Next

        End If

        DisplaySetup(Schedule\_Date)

    End Sub

    Private Sub ClinicianManagerToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ClinicianManagerToolStripMenuItem.Click

        ClinicianConsole.Show()

        ClinicianConsole.Focus()

    End Sub

    Private Sub CloseToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CloseToolStripMenuItem.Click

        Me.Close()

    End Sub

    Private Sub StudentManagerToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StudentManagerToolStripMenuItem1.Click

        StudentManager.Show()

        StudentManager.Focus()

    End Sub

    Private Sub StudentManagerToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

        RescheduleDailyDisplay.Show()

        RescheduleDailyDisplay.Focus()

    End Sub

    Private Sub Timer1\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

        Dim scheduleDate As Date

        scheduleDate = MonthCalendar1.SelectionStart

        Removerows(scheduleDate)

    End Sub

    Private Sub DailyScheduleToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DailyScheduleToolStripMenuItem.Click

        PrintofficeSchedules()

    End Sub

    'Load up print daily office schedule form

    Public Sub PrintofficeSchedules()

        OfficeSchedulePrintOut.Show()

  OfficeSchedulePrintOut.DateTimePicker1.Value = MonthCalendar1.SelectionStart()

  OfficeSchedulePrintOut.DateTimePicker2.Value = MonthCalendar1.SelectionStart()

        OfficeSchedulePrintOut.Focus()

    End Sub

    'Allows operator the ability to adjust width of display  columns in the gridview control

Private Sub ToolStripButton4\_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripButton4.Click

        If ToolStripTextBox1.Text = String.Empty Then

            Exit Sub

        End If

        Dim newWidth As Integer

        newWidth = Convert.ToInt16(ToolStripTextBox1.Text)

        If newWidth < 10 Then

            For x = 1 To DataGridView1.Columns.Count - 1

                DataGridView1.Columns(x).Width = 100

            Next

            newWidth = 100

            Exit Sub

        ElseIf newWidth > 150 Then

            For x = 1 To DataGridView1.Columns.Count - 1

                DataGridView1.Columns(x).Width = 100

            Next

            newWidth = 100

        Else

            For x = 1 To DataGridView1.Columns.Count - 1

                DataGridView1.Columns(x).Width = newWidth

            Next

        End If

        Dim changeColumnSize As New StoreGridViewColumnWidth

        newWidth = changeColumnSize.SaveColumnWidth(0, 1, newWidth)

        ToolStripTextBox1.Text = newWidth.ToString

    End Sub

    Private Sub Button4\_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click

        AddStudentTrigger()

    End Sub

    'Open NewStudent.vb FORM

    Public Sub AddStudentTrigger()

    If ComboBox1.SelectedText = String.Empty And ComboBox1.SelectedIndex <= 0 Then

            Dim firstName As String = String.Empty

            Dim lastName As String = String.Empty

            Dim studentFullname As String = String.Empty

            NewStudent.Show()

            NewStudent.Focus()

            Dim splitName As String() = Nothing

            studentFullname = ComboBox1.Text

            If studentFullname.IndexOf(",") > -1 Then

                If studentFullname <> String.Empty Then

                    splitName = studentFullname.Split(", ")

                    firstName = splitName(1)

                    lastName = splitName(0)

                End If

                NewStudent.TextBox1.Text = firstName.Trim

                NewStudent.TextBox2.Text = lastName.Trim

            End If

        End If

    End Sub

    Private Sub Button3\_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click

        EditUser()

    End Sub

'Lookup student data and place then into their respective  fields of the EditStudentProfile.vb FORM

    Public Function EditUser()

        If ComboBox1.SelectedIndex > 0 Then

            Dim studentData As IstudentAttributesCollection = New userAttributesCollection

            Dim studentId As String = String.Empty

            Dim studentFullname As String = String.Empty

            Dim splitName As String() = Nothing

            Dim parseApostrophe As New nameOperation

            Dim SchoolDistrict As String = String.Empty

            Dim School As String = String.Empty

            Dim AssessmentDate As String = String.Empty

            Dim RptDiscussiondate As String = String.Empty

            Dim InitialInquiry As String = String.Empty

            Dim TutorStart As String = String.Empty

            Dim tutorStop As String = String.Empty

            Dim DOB As String

            Dim convertstudentname As INameConversion = New StudentNameconversion

            Dim web As String = String.Empty

            Dim activeStudent As String = String.Empty

            Dim dgvRow As New DataGridViewRow

            Dim convertname As New Schedule

            EditStudentProfile.Show()

            studentFullname = ComboBox1.SelectedValue

            studentId = convertstudentname.convertToId(studentFullname)

            Dim studentCollection As New ArrayList

            studentCollection = studentData.StudentInfo(studentId)

            splitName = studentFullname.Split(", ")

            studentLastname = splitName(0)

            studentFirstname = splitName(1)

            studentId = convertstudentname.convertToId(studentFullname.Trim)

            studentFirstname = parseApostrophe.executeName(studentFirstname, 2)

            studentLastname = parseApostrophe.executeName(studentLastname, 2)

            DOB = studentCollection(2)

            Dim Gender As String = studentCollection(3)

            InitialInquiry = studentCollection(6)

            AssessmentDate = studentCollection(7)

            RptDiscussiondate = studentCollection(8)

            TutorStart = studentCollection(9)

            tutorStop = studentCollection(10)

            SchoolDistrict = studentCollection(4)

            School = studentCollection(5)

            activeStudent = studentCollection(11)

            School = parseApostrophe.executeName(School, 2)

            SchoolDistrict = parseApostrophe.executeName(SchoolDistrict, 2)

    'Appends a 0 to the beginning of the date if it is missing

            Dim iilength As Integer = 0

            iilength = InitialInquiry.Length

            If iilength = 9 Then

                InitialInquiry = "0" & InitialInquiry

            End If

            EditStudentProfile.TextBox3.Text = SchoolDistrict.Trim

            EditStudentProfile.TextBox4.Text = School.Trim

            EditStudentProfile.ComboBox1.SelectedItem = Gender.Trim

            EditStudentProfile.TextBox1.Text = studentFirstname.Trim

            EditStudentProfile.TextBox2.Text = studentLastname.Trim

            EditStudentProfile.MaskedTextBox1.Text = DOB.Trim

            EditStudentProfile.MaskedTextBox2.Text = InitialInquiry

            EditStudentProfile.MaskedTextBox3.Text = AssessmentDate.Trim

            EditStudentProfile.MaskedTextBox4.Text = RptDiscussiondate.Trim

            EditStudentProfile.MaskedTextBox5.Text = TutorStart.Trim

            EditStudentProfile.MaskedTextBox6.Text = tutorStop.Trim

            EditStudentProfile.displayGuardians(studentFirstname, studentLastname)

            If activeStudent = "True" Then

                EditStudentProfile.CheckBox1.Checked = True

            ElseIf activeStudent = "False" Then

                EditStudentProfile.CheckBox1.Checked = False

            End If

            If Gender.Trim = "Male" Then

                EditStudentProfile.ComboBox1.SelectedIndex = 0

            Else

                EditStudentProfile.ComboBox1.SelectedIndex = 1

            End If

            EditStudentProfile.Label29.Text = studentId.Trim

            Dim readingLevel As String = String.Empty

            Dim returnLevel As New returnStudentData

            readingLevel = returnLevel.StudentReadingLevel(studentId.Trim)

            EditStudentProfile.TextBox7.Text = readingLevel.Trim

            Dim studentInfo As IstudentAttributesDatasets = New userProfileAttributes

            Dim schooltype As Boolean

            Dim dt As DataTable

            dt = studentInfo.RetrieveStudentSchool(studentId.Trim)

            Dim row As DataRow

            For Each row In dt.Rows

                schooltype = row("Prv\_Pub")

            Next

            EditStudentProfile.TextBox3.Visible = True

            EditStudentProfile.ComboBox2.Visible = False

            EditStudentProfile.TextBox3.Text = SchoolDistrict.Trim

            EditStudentProfile.RadioButton1.Checked = True

            EditStudentProfile.RadioButton2.Checked = False

            EditStudentProfile.Label7.Text = "School District"

            EditStudentProfile.TextBox4.Text = School.Trim

            Dim dv As New DataView

            dv = studentInfo.RetrieveNotes(studentId.Trim)

            EditStudentProfile.DataGridView1.DataSource = dv

            EditStudentProfile.DataGridView1.Columns(0).Visible = False

            EditStudentProfile.DataGridView1.Columns(1).Visible = False

            EditStudentProfile.DataGridView1.Columns(3).Width = 150

            EditStudentProfile.DataGridView1.Columns(4).Width = 400

            EditStudentProfile.TextBox5.ReadOnly = True

            EditStudentProfile.TextBox6.ReadOnly = True

        End If

        Return Nothing

    End Function

    'Refresh student combobox Listing

    Public Function Reset()

        Dim names As IPopulateAllNames = New IPopulateNames

        Dim ds As New DataSet

        Dim dsStudent As New DataSet

        dsStudent = names.DisplayStudents(True)

        Dim dtStudent As DataTable = dsStudent.Tables("StudentList")

        ComboBox1.DataSource = dtStudent

        ComboBox1.DisplayMember = "FullName"

        ComboBox1.ValueMember = "FullName"

        ComboBox1.SelectedIndex = 0

        Return Nothing

    End Function

    Private Sub ComboBox1\_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

        ComboBox1.Focus()

    End Sub

    Private Sub Button10\_Click(sender As System.Object, e As System.EventArgs) Handles Button10.Click

        ExportSchedule.Show()

    End Sub

    Private Sub Button9\_Click(sender As System.Object, e As System.EventArgs) Handles Button9.Click

        ClinicianCalendar.Show()

    End Sub

    Private Sub Button8\_Click(sender As System.Object, e As System.EventArgs) Handles Button8.Click

        StudentCalendar.Show()

    End Sub

    Private Sub Button7\_Click(sender As System.Object, e As System.EventArgs) Handles Button7.Click

        OfficeSchedulePrintOut.Show()

    End Sub

    Private Sub Button5\_Click(sender As System.Object, e As System.EventArgs) Handles Button5.Click

        StudentManager.Show()

    End Sub

    Private Sub Button6\_Click(sender As System.Object, e As System.EventArgs) Handles Button6.Click

        ClinicianConsole.Show()

    End Sub

    Private Sub Button2\_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click

        SchedulingConsole.Show()

    End Sub

    Private Sub Timer2\_Tick(sender As System.Object, e As System.EventArgs) Handles Timer2.Tick

        Timer2.Stop()

        logOffUser()

    End Sub

    Private Sub Button11\_Click(sender As System.Object, e As System.EventArgs) Handles Button11.Click

        SignIn.Show()

        SignIn.Focus()

    End Sub

    Private Sub Button12\_Click(sender As System.Object, e As System.EventArgs) Handles Button12.Click

        logOffUser()

    End Sub

    'Sign out the user and disable controls

    Public Sub logOffUser()

        Me.Timer2.Enabled = False

        GroupBox4.Enabled = False

        GroupBox5.Enabled = False

        ComboBox1.Enabled = False

        Button3.Enabled = False

        Button4.Enabled = False

        Button11.Enabled = True

        Button12.Enabled = False

        SchedulingConsole.Close()

        RescheduleDailyDisplay.Close()

        StudentManager.Close()

        StudentParentAsociation.Close()

        NewStudent.Close()

        ClinicianConsole.Close()

        EditStudentProfile.Close()

        EditGuardianProfile.Close()

        EditClinicianProfile.Close()

        NewClinician.Close()

        NewGuradian.Close()

        StudentCalendar.Button5.Enabled = False

        Label4.Text = "None"

    End Sub

    Private Sub ClinicianScheduleToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs) Handles ClinicianScheduleToolStripMenuItem.Click

        ClinicianCalendar.Show()

        ClinicianCalendar.Focus()

    End Sub

    Private Sub ScheduleDetailsToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs) Handles ScheduleDetailsToolStripMenuItem.Click

        ExportSchedule.Show()

        ExportSchedule.Focus()

    End Sub

    Private Sub StudentScheduleToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs) Handles StudentScheduleToolStripMenuItem.Click

        StudentCalendar.Show()

        StudentCalendar.Focus()

    End Sub

    Private Sub AddRemoveAppointmentToolStripMenuItem\_Click(sender As System.Object, e As System.EventArgs) Handles AddRemoveAppointmentToolStripMenuItem.Click

        SchedulingConsole.Show()

        SchedulingConsole.Focus()

    End Sub

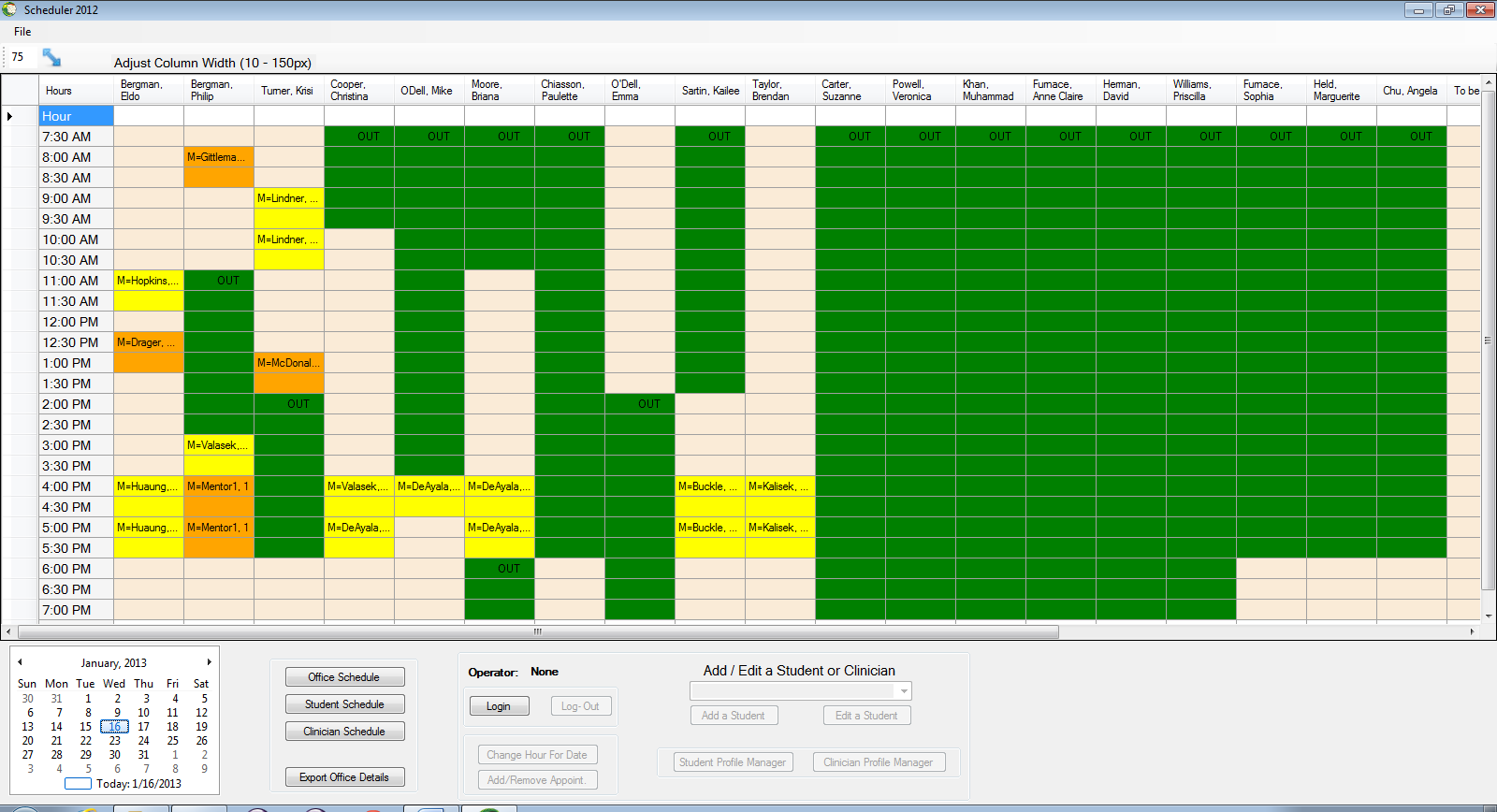
    Private Sub ToolStripMenuItem1\_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripMenuItem1.Click

        RescheduleDailyDisplay.Show()

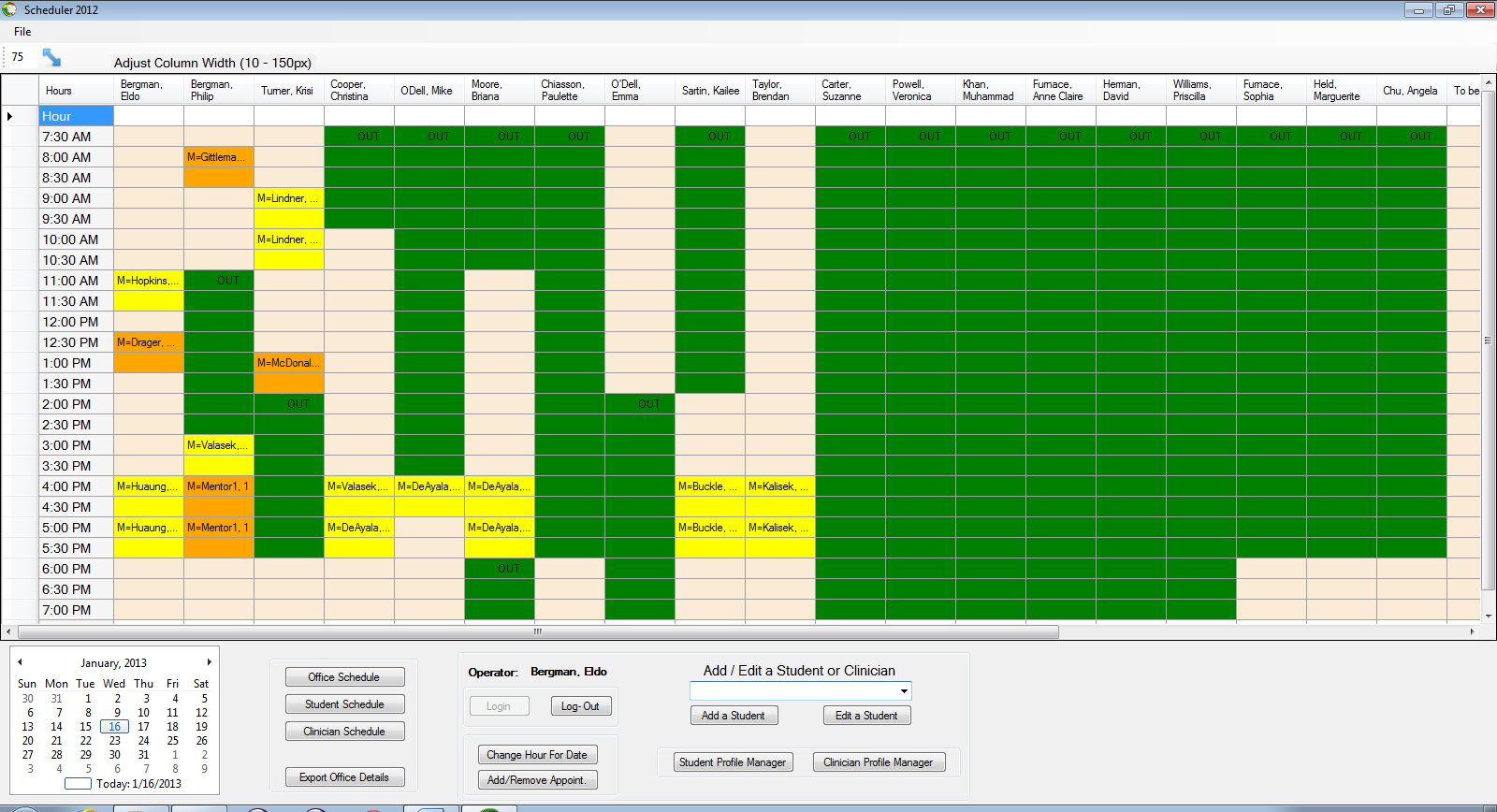
        RescheduleDailyDisplay.Focus()

    End Sub

End Class



Home Display with user signed out.



Home Display with user signed in