Imports BAL

Imports DAL

Imports System

Imports System.Windows

Imports System.Drawing.Imaging

Imports System.Printing.PrintQueue

Public Class OfficeSchedulePrintOut

    Private dateholder As Date

    Private newday As Date

    Private dys As Integer

    Private x1 As Integer

    Private selectedFolder As String = String.Empty

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

        Dim convertApostrophe As New nameOperation

'Label the first row each column with active Clinicians

        Dim clinicianinfo As New Clinicians

        Dim dsclinicians As New DataSet

'Return all active clincians

        dsclinicians = clinicianinfo.GetClinicianInfo(True)

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

        Dim clinicianFirstName As String = String.Empty

        Dim clinicianLastName As String = String.Empty

Place all the clinician that were returned into the dataGridview.

New columns will be added to the gridview according to the number of clinicians that are returned

        Dim row As DataRow

        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("Column1", ClinicianFullName)

        Next

'Setup initial display display with the date from the Home Display Form

        Dim startDay As String = String.Empty

'Display the selected day from the home screen form display

        startDay = DateTimePicker1.Value.ToString("M/dd/yyyy")

        newday = DateTime.ParseExact(startDay, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

        ‘dateholder’ is a global variable that needs to be set to the selected date

        dateholder = newday

        Dim newWidth As Integer

        Dim changeColumnSize As New StoreGridViewColumnWidth

        newWidth = changeColumnSize.ReturnColumnWidth(1, 1)

        TextBox1.Text = newWidth.ToString

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

            DataGridView1.Columns(x).Width = newWidth

        Next

   'Execute the display timer only

        updateDisplay()

        GroupBox1.Visible = True

        GroupBox2.Visible = True

    End Sub

    'Setup the DataGrid Display Layout

    'Return Dataset of all scheduled students and clinicians

    'Return all scheduled students on the given date

    'Go through datset and return each active clinician

    'Go through dataset and return each active clinician

    'Go through each header title in the DataGrid , and match it with

    'the clinician that is returned from the database.

    'If there is a match, iterate through each row in the column of the clinician

    'Iterate through all 24 rows on every column(Clinician)

    'Check to see if Clinician is scheduled Out at that Timeslot'

    'Or Check to see if there is a student name at that Time slot.

    'Check to see if cell is the start of the clinician being scheduled out

    'This is clinicianout variable increment it to see if the clinician is out the entire day

    'Iterate through each student whos start time and end time match within the current time range of the Datagrid Control.

    'Return Student information to be displayed

    'Color the cell accordingly to their status or Attendance value

    'Check to see if the first two hours exsist. If so then color the cell white.

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

    'Clear variables and color time slot as WhiteSmoke if there is nothing at that specific time slot

    '0 Marks the Clinician as not being out for the entire day therefor that column will not be hidden when printed

    'Check to see if the current cell is different from the last cell

    'if so then Display the location of the student

    'Clear Location variable so it will not be displayed in other cells during the iteration process

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

    'Check to see if clinician out has reached the 23 row

    'If Clinician is out the entire day remove the entire clinician column,

    'otherwise if the clinician out only part of the day or not at all then

    'keep column visible.

    Public Function DisplaySetup(ByVal CurrentDate As Date) As Integer

        Dim getReadingLevel As IstudentProfileAttributes = New ReadingLevel

        Dim convertStudentName As INameConversion = New StudentNameconversion

        Dim Clinicianout As Integer = 0

        Dim Student As String = String.Empty

        Dim display As IDisplaySetup = New DisplayModule

        Dim Stat As New Schedule

        Dim ds2 As New DataSet

        Dim ds3 As New DataSet

        Dim ds5 As New DataSet

        Dim ds As New DataSet

        'Setup the DataGrid Display Layout

        DisplayTemplate()

        'Return Dataset of all scheduled students and clinicians

        ds = display.mainDisplaySchedule(CurrentDate)

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

        Dim splitname\_student() As String = Nothing

        Dim studentid As String = String.Empty

        Dim nextname As String = String.Empty

        Dim countid As String = String.Empty

        Dim subject As String = String.Empty

        Dim Location As String = String.Empty

        Dim toggleStudentReadingLevel As Boolean = False

        'Display student reading level

        If CheckBox1.Checked = True Then

            toggleStudentReadingLevel = True

        Else

            toggleStudentReadingLevel = False

        End If

        'Return all scheduled students on the given date

        ds2 = Stat.GetSchedule(CurrentDate, CurrentDate)

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

        Dim b As Integer

        Dim Status As String = String.Empty

        Dim StudentReadingLevel = String.Empty

        Dim Clinician As String = String.Empty

        Dim ClinicianHeader As String = String.Empty

        Dim present As String = String.Empty

Go through and return each active clinician

        For Each rw In dt.Rows

            Clinician = rw("Clinician")

            Dim numberofColumns As Integer = DataGridView1.ColumnCount

            Dim headercount As Integer = 0

'Go through each header title in the GridView control, and match it with

 the clinician that is returned from the database.

 If there is a match, iterate through each row in the current column

            For headercount = 1 To numberofColumns

                ClinicianHeader = DataGridView1.Columns(headercount).HeaderText

                If ClinicianHeader = Clinician 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 24 rows on every column(Clinician)

                    For a = 1 To 24

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

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

                        Student = rw(a).ToString

                        splitname\_student = Student.Split(",")

 'Check to see if Clinician is out'

 'Or Check to see if there is a student name.

                        If Student = "           " & "OUT" Then

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

  Check to see if cell is the start of the clinician being out

                            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

 This is increment to see if the clinician is out the entire day

                            Clinicianout = Clinicianout + 1

                        ElseIf Student <> String.Empty Then

                            studentid = convertStudentName.convertToId(Student)

                            Dim timestamp As String = String.Empty

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

                            Dim time1 As DateTime

                            Dim time2 As DateTime

Iterate through each student whos starttime and endtime match within the current  time range of the Datagrid Control.

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

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

Return Student information to be displayed

                            b = 0

                            Do While b <= foundrow.Length - 1

                                Status = foundrow(b)("status")

                                countid = foundrow(b)("Count")

                                present = foundrow(b)("Attendance")

 Check the Checkbox1 to see if user wants to display the student level number after the last name

                                If toggleStudentReadingLevel = True Then

                                    StudentReadingLevel = getReadingLevel.level(studentid.Trim)

Student = splitname\_student(0).Trim & " " & StudentReadingLevel & ", " & splitname\_student(1).Trim

                                End If

                                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")

                                Next

             Color the cell accordingly to their status and Attendance

                                'Proposed only=Yellow

                                'Absent=PaleVioletRed

                                'Completed=LightSkyBlue

                                'Transfer=LightYellow

                                'Testing=Gray

                                'Meeting=Orange

                                'start hour=AntiqueWhite

                                If present.Trim = "Absent" Then

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

                                ElseIf present.Trim = "Completed" Then

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

                                ElseIf Status.Trim = "Transfer" Then

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

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

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

                                ElseIf present.Trim = "Proposed" And subject.Trim = "Start" Then

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

                                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.LightYellow

                                End If

                                b = b + 1

                            Loop

           'Fill the background color

                        ElseIf Student = String.Empty Then

         'Clear variables and color time slot as empty

                            studentid = String.Empty

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

                  0 Marks the Clinician as not being out for the entire day

                            Clinicianout = 0

                        Else

                        End If

              'Check to see if student is on the start of an hour

              'Check to see if the current cell is different from the last cell

              'if so then display the campus the student is located at

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

                  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

                            Clinicianout = 0

                        Else

                        End If

                    Next

                    Exit For

                Else

                End If

            Next

         Check to see if clinician out has reached the 23rd row

         If Clinician is out the entire day remove the entire clinician column,

         Otherwise if the clinician out only part of the day or not at all then

            'keep column

            If Clinicianout > 22 Then

                DataGridView1.Columns(headercount).Visible = False

            ElseIf Clinicianout < 23 Then

                Clinicianout = 0

                DataGridView1.Columns(headercount).Visible = True

            End If

        Next

        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

        Return DataGridView1.Rows.Count - 1

    End Function

Setup and display all 25 rows labeled with a time slots in the first column of  the Datagrid from 7:30 AM to 7:30 PM

    Public Sub DisplayTemplate()

        DataGridView1.Columns(0).DefaultCellStyle.Font = New Font("Times NewRoman", 8, FontStyle.Regular)

        Dim i As Integer = 0

        Dim timestamp As Array = {"", "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"}

Create each row and place the hour label into column 0 and color it with a  whitesmoke.

        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.White

        Next

        DataGridView1.Columns(0).Width = 54

    End Sub

    'Trigger Print operation

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

        printOfficeSchedule()

    End Sub

    'Display Print Dialog Control

    'Wait for selection and begin timers, Hide the DateTimePicker, and buttons

    'controls at the bottom of the form

    Public Sub printOfficeSchedule()

        'Display Print Dialog Control

        GroupBox1.Visible = True

        GroupBox2.Visible = True

        Dim lastday, startday As String

        'Wait for selection and begin timers, Hide the DateTimePicker, and buttons

        'controls at the bottom of the form

        If PrintDialog1.ShowDialog = DialogResult.OK Then

            startday = DateTimePicker1.Value.ToString("M/dd/yyyy")

            lastday = DateTimePicker2.Value.ToString("M/dd/yyyy")

            newday = DateTime.ParseExact(startday, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

            dateholder = DateTime.ParseExact(lastday, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

            Label7.Visible = True

            Label7.Text = "Date printed: " + Now.ToString("ddd M/d/yyyy   h:mm tt")

            Timer1.Enabled = True

            GroupBox1.Visible = False

            GroupBox2.Visible = False

        Else

            Timer1.Enabled = False

            GroupBox1.Visible = True

            GroupBox2.Visible = True

        End If

    End Sub

'During Screen Grab and print operation pause timer1 (the operation which cycles to the next day)

    'Do a Screen Grab and then

    'Define the printer settings

    Public Sub Printfunction()

        Timer1.Enabled = False

        'Screen capture takes place

        Dim screenGrab As New Bitmap(Me.Bounds.Width, Me.Bounds.Height, PixelFormat.Format32bppArgb)

        Dim g As System.Drawing.Graphics = System.Drawing.Graphics.FromImage(screenGrab)

        g.CopyFromScreen(Me.Bounds.X, Me.Bounds.Y, 0, 0, Me.Bounds.Size, CopyPixelOperation.SourceCopy)

        Dim selectedprinter As String

        For Each PSource As System.Drawing.Printing.PaperSource In PrintForm1.PrinterSettings.PaperSources

            If PSource.Kind = Drawing.Printing.PaperSourceKind.Custom Then

                PrintForm1.PrinterSettings.DefaultPageSettings.PaperSource = PSource

                Exit For

            End If

        Next

        'Print to selected printer

        Dim ps As System.Drawing.Printing.PaperSize

        For ix As Integer = 0 To PrintDialog1.PrinterSettings.PaperSizes.Count - 1

            If PrintDialog1.PrinterSettings.PaperSizes(ix).Kind = Drawing.Printing.PaperKind.Legal Then

                ps = PrintForm1.PrinterSettings.PaperSizes(ix)

                PrintForm1.PrinterSettings.DefaultPageSettings.PaperSize = ps

                Exit For

            End If

        Next

        'Printer settings from the print dialog box

        PrintForm1.PrinterSettings.PrinterName = PrintDialog1.PrinterSettings.PrinterName

        PrintForm1.PrinterSettings.DefaultPageSettings.Landscape = True

        selectedprinter = PrintDialog1.PrinterSettings.PrinterName

        PrintForm1.PrinterSettings.DefaultPageSettings.Margins = New System.Drawing.Printing.Margins(5, 5, 5, 5)

        PrintForm1.PrintAction = Drawing.Printing.PrintAction.PrintToPrinter

        PrintForm1.Print(Me, PowerPacks.Printing.PrintForm.PrintOption.Scrollable)

        Timer1.Enabled = True

    End Sub

  Go through each day within the selected date range and display the calendar day.

    Public Function updateDisplay()

        Dim currentDay As String = String.Empty

        Dim iterateDay As Date

        'Go through each date

        currentDay = dateholder.ToString("M/dd/yyyy")

        iterateDay = DateTime.ParseExact(currentDay, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

        If iterateDay >= newday Then

            'Display the current date

            Label1.Text = dateholder.ToString("dddd, M/d/yyyy")

            'Clear the entire Datagridview by removing all of its rows.

            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

            'Display the next day within the selected date range

            DisplaySetup(iterateDay)

            dateholder = dateholder.AddDays(-1)

            GroupBox1.Visible = False

            GroupBox2.Visible = False

        Else

            Timer1.Enabled = False

            Timer3.Enabled = False

 Start a timer delay to prevent the last day displayed date from showing within the groupboxe controls at the footer of the screen.

            Label7.Visible = False

            GroupBox1.Visible = True

            GroupBox2.Visible = True

        End If

        Return Nothing

    End Function

    'Exit Screen Button

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

        Me.Close()

    End Sub

    'Display Save dialog control. To give the user a place to store all calendar(s)

    Public Sub storedailycalendarSave()

        Dim getStoredPath As IfileStoragePath = New returnStorageModules

        Dim savePath As IfileStoragePath = New saveStorageModules

        selectedFolder = getStoredPath.path(selectedFolder, "office")

        If selectedFolder <> String.Empty Then

            savePath.path(selectedFolder, "office")

        End If

        InitiatePrint()

    End Sub

    Public Sub storedailycalendarSaveAs()

        If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then

            Dim savePath As IfileStoragePath = New saveStorageModules

   'Save new folder location to the data store

            selectedFolder = FolderBrowserDialog1.SelectedPath

            savePath.path(selectedFolder, "office")

            InitiatePrint()

        ElseIf FolderBrowserDialog1.ShowDialog() = DialogResult.Cancel Then

            Exit Sub

        End If

    End Sub

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

        updateDisplay()

        If Timer1.Enabled = True Then

            Printfunction()

        End If

    End Sub

    'Store calendar at the specified location

    'Perform a screen capture

    Private Sub Timer3\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer3.Tick

        updateDisplay()

        saveOfficeSchedule()

    End Sub

    'Save office schedule to data store in a jpg format

    Public Function saveOfficeSchedule()

        Timer3.Enabled = False

        Dim screenGrab As New Bitmap(Me.Bounds.Width, Me.Bounds.Height, PixelFormat.Format32bppArgb)

        Dim g As System.Drawing.Graphics = System.Drawing.Graphics.FromImage(screenGrab)

        g.CopyFromScreen(Me.Bounds.X, Me.Bounds.Y, 0, 0, Me.Bounds.Size, CopyPixelOperation.SourceCopy)

        Dim datea As DateTime = Convert.ToDateTime(Label1.Text.Trim)

        Dim stringdate As String = datea.ToString("MMMM dd, yyyy")

        Dim filename As String = selectedFolder & "\OfficeSchedule\_" & stringdate.Trim & ".jpg"

        screenGrab.Save(filename, System.Drawing.Imaging.ImageFormat.Jpeg)

        'Go to next day

        Timer3.Enabled = True

        Return Nothing

    End Function

    Private Sub DateTimePicker1\_ValueChanged(sender As System.Object, e As System.EventArgs) Handles DateTimePicker1.ValueChanged

        Dim currentDay As String = String.Empty

        currentDay = DateTimePicker1.Value.ToString("M/dd/yyyy")

        DateTimePicker2.Value = DateTimePicker1.Value

        'Display the selected day from the home screen form display

        newday = DateTime.ParseExact(currentDay, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

        'Global variable need to be set to the selected date

        dateholder = newday

        updateDisplay()

        GroupBox1.Visible = True

        GroupBox2.Visible = True

    End Sub

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

        adjustColumnWidth()

    End Sub

    'Adjust the width of the display columns.

    Public Function adjustColumnWidth()

        If TextBox1.Text <> String.Empty Then

        Dim newWidth As Integer

        newWidth = Convert.ToInt16(TextBox1.Text)

        If newWidth < 10 Then

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

                DataGridView1.Columns(x).Width = 80

            Next

            newWidth = 80

        ElseIf newWidth > 150 Then

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

                DataGridView1.Columns(x).Width = 80

            Next

            newWidth = 80

        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(1, 1, newWidth)

            TextBox1.Text = newWidth.ToString

        End If

        Return Nothing

    End Function

    Private Sub CheckBox1\_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles CheckBox1.CheckedChanged

        Dim lastday As String = String.Empty

        'Display the selected day from the home screen form display

        lastday = DateTimePicker1.Value.ToString("M/dd/yyyy")

        newday = DateTime.ParseExact(lastDay, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

        'Global variable need to be set to the selected date

        dateholder = newday

        'Execute the display timer only

        updateDisplay()

        GroupBox1.Visible = True

        GroupBox2.Visible = True

    End Sub

    Private Sub ToolStripButton1\_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripButton1.Click

        storedailycalendarSave()

    End Sub

    Private Sub ToolStripButton2\_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripButton2.Click

        storedailycalendarSaveAs()

    End Sub

    Public Sub InitiatePrint()

        Dim lastDay As String = String.Empty

        Dim startDay As String = String.Empty

        startDay = DateTimePicker1.Value.ToString("M/dd/yyyy")

        lastDay = DateTimePicker2.Value.ToString("M/dd/yyyy")

        newday = DateTime.ParseExact(startDay, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

        dateholder = DateTime.ParseExact(lastDay, "M/dd/yyyy", Globalization.DateTimeFormatInfo.InvariantInfo)

        Label7.Visible = True

        Label7.Text = "Date printed: " + Now.ToString("ddd M/d/yyyy   h:mm tt")

        'Hide the Controls at the bottom of the display

        GroupBox1.Visible = False

        GroupBox2.Visible = False

        Timer3.Enabled = True

    End Sub

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

        storedailycalendarSaveAs()

    End Sub

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

        storedailycalendarSave()

    End Sub

End Class

Retrieve the stored reading level of a student

Public Interface IstudentProfileAttributes

    Function level(ByVal Studentid) As String

End Interface

Public Class ReadingLevel

    Implements IstudentProfileAttributes

    Public Function level(studentId As Object) As String Implements IstudentProfileAttributes.level

        Dim getreadingLevel As New returnStudentData

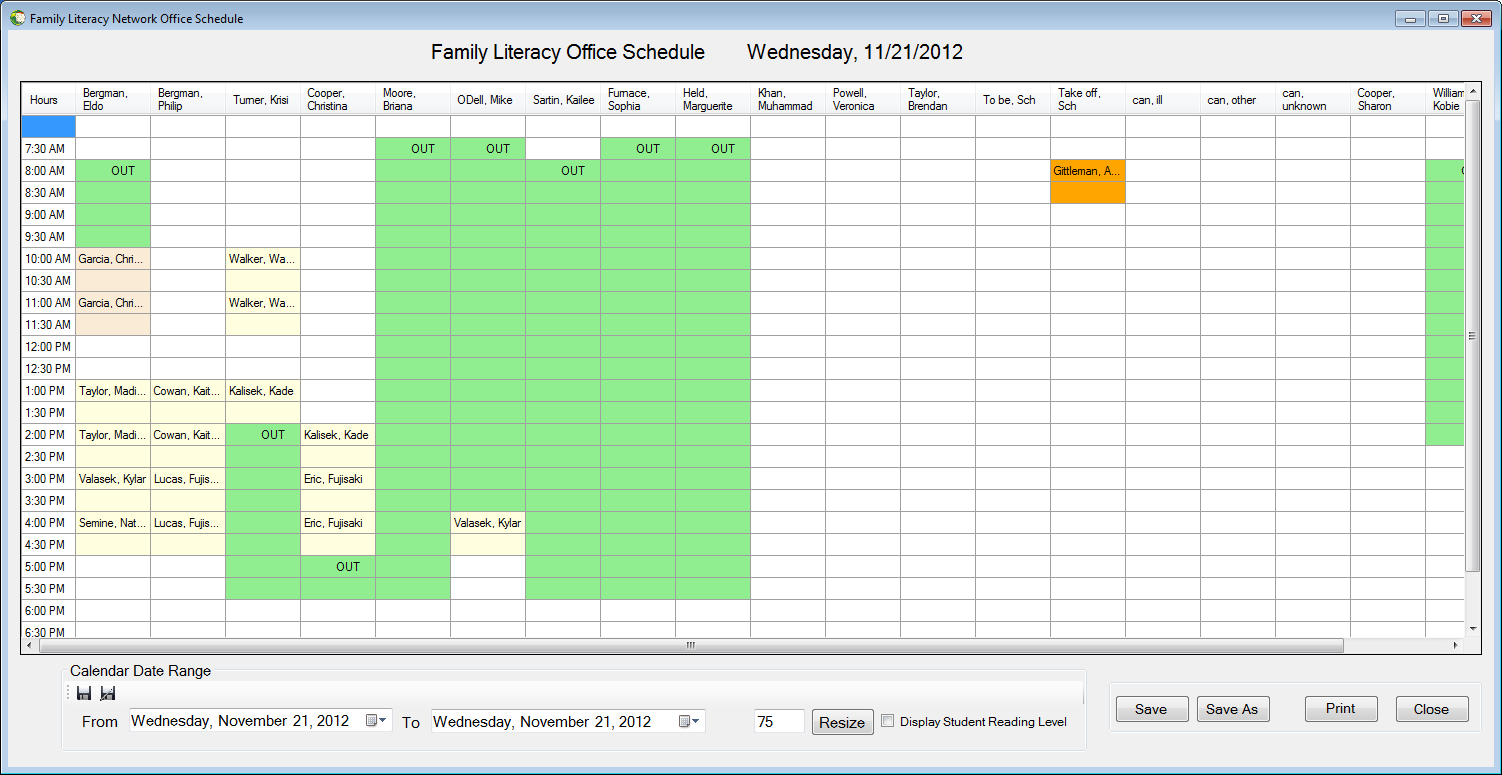
        Dim newlevel As String = String.Empty

        newlevel = getreadingLevel.StudentReadingLevel(studentId)

        Return newlevel

    End Function

End Class



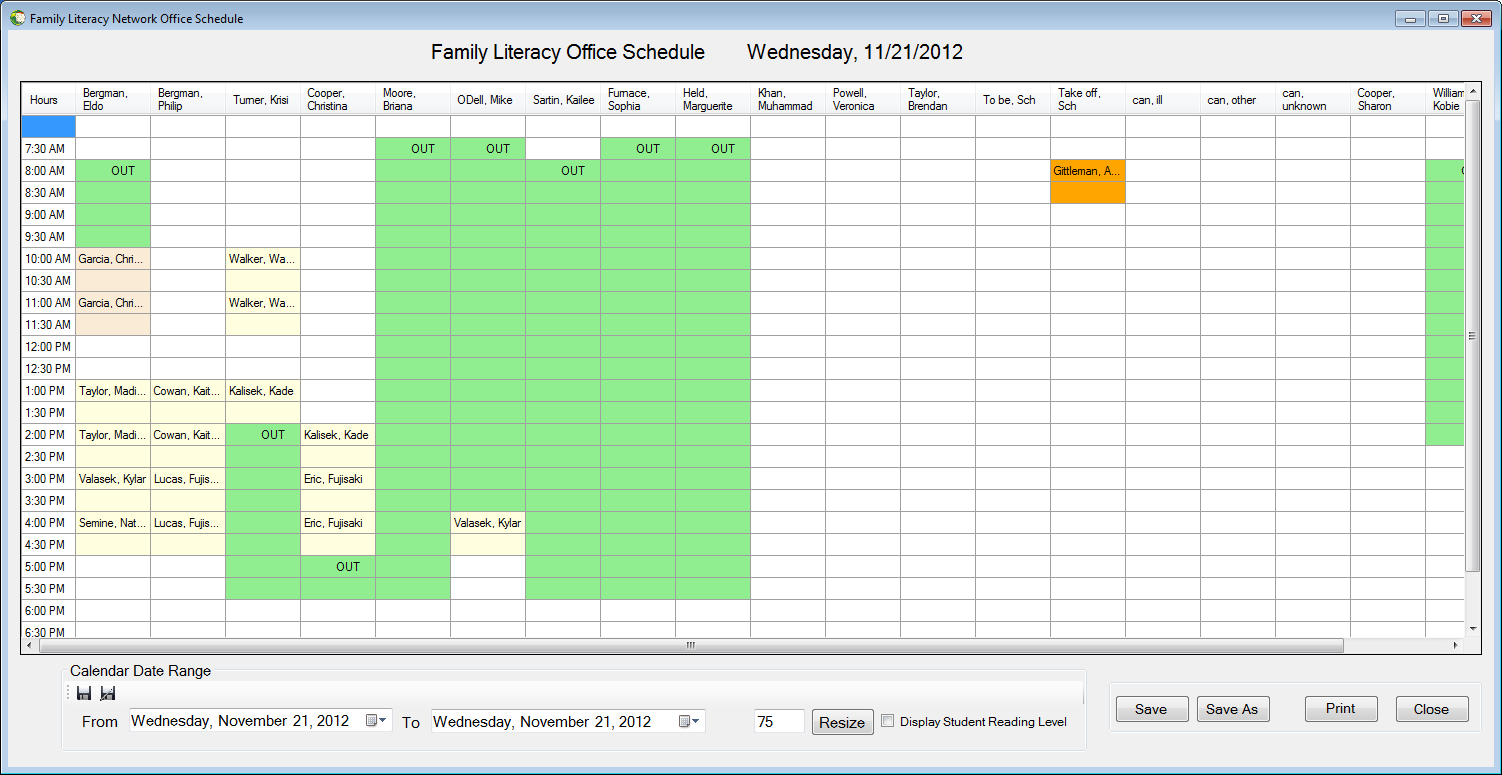
b e

a

c d

This form allows the user to print or save an office schedule of any range of dates that they specify. They can also resize the columns to any given width. The user can also specify whether or not to display the stored reading level of the children.

1. **Main print out display**



This is the viewing area which allows the user to visual see which child is assigned to what clinician.

Green: Clinician is off that particular set of hours.

**Antique White**: Hour 0 of the child

Yellow: Child is scheduled at that time slot to be tutored

Orange: Go to the childs school for testing.

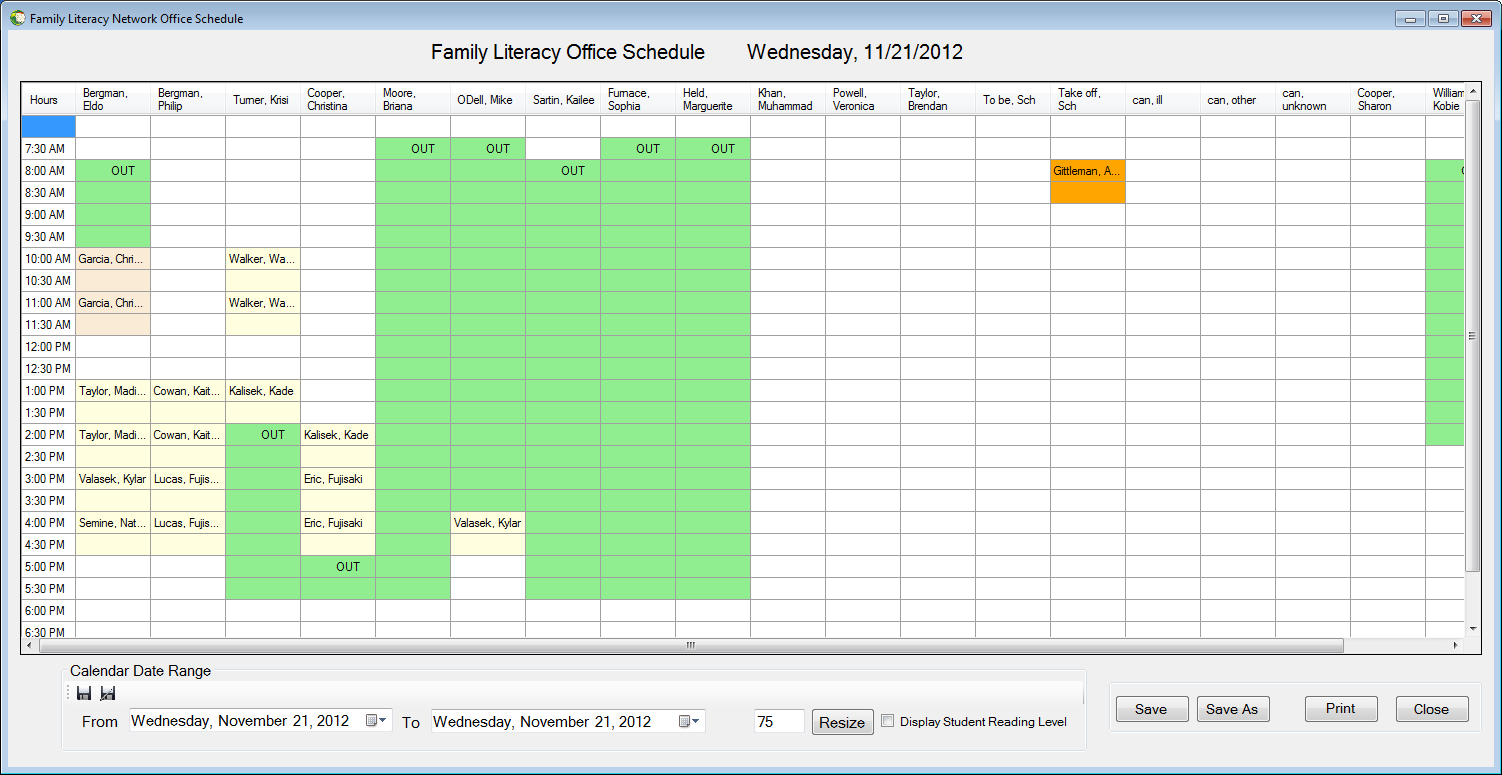
Gray: Child testing

1. **Date Range Panel**

Allow the user to select a range of dates to printed or saved.

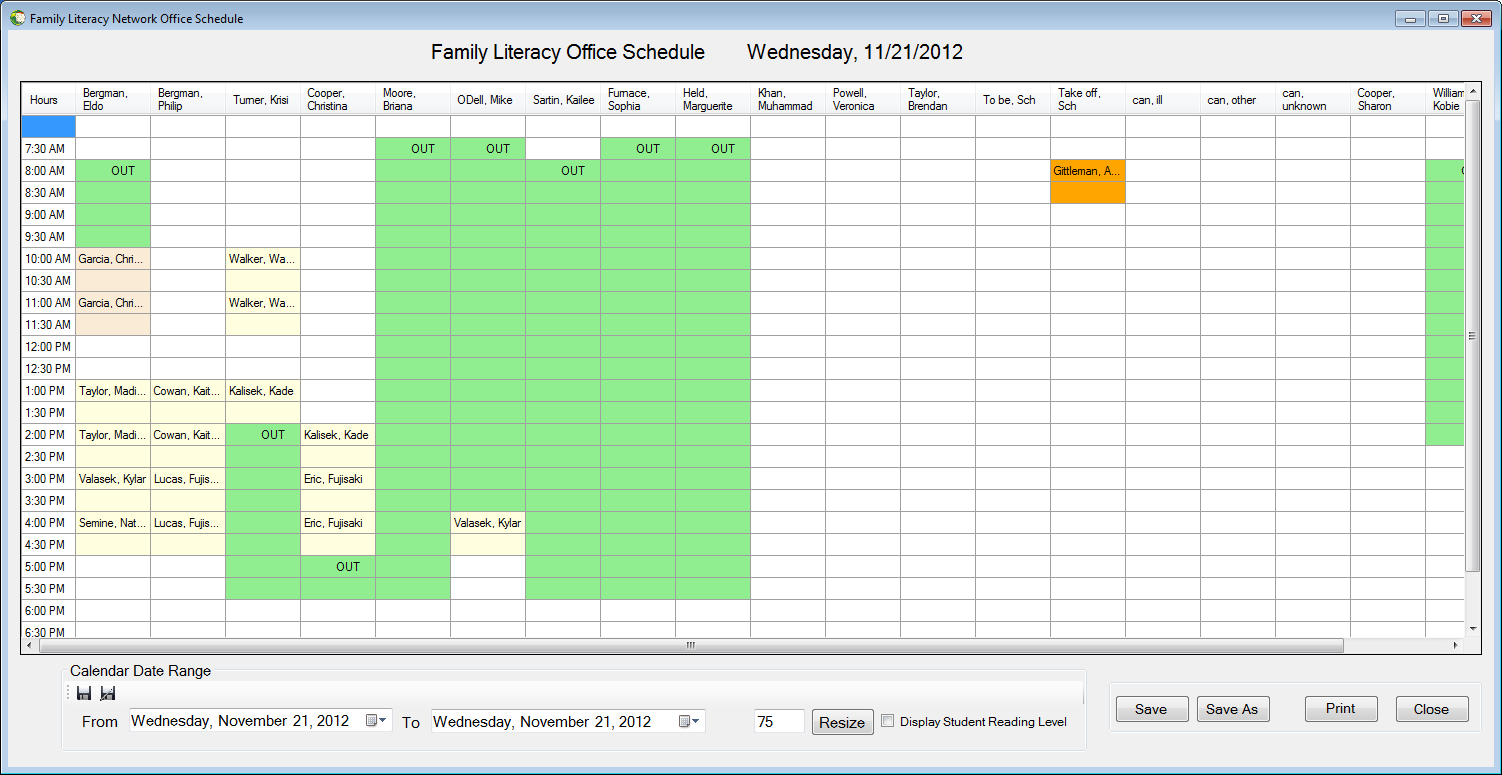
The user must select a starting date and an ending date.

1. **Select a “From” date**



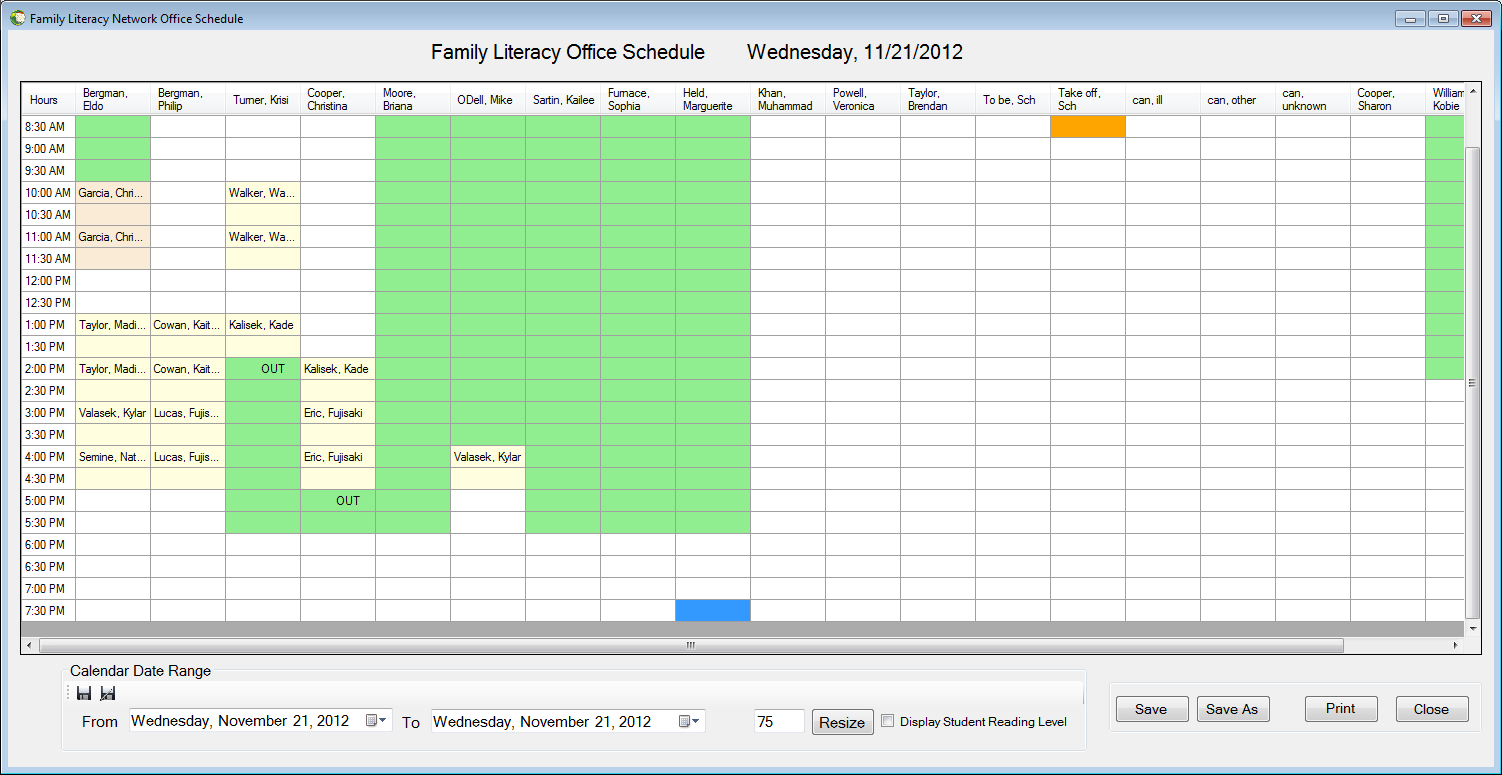
Select a date from which you would like to start printing from.

1. **Select a “To” date**



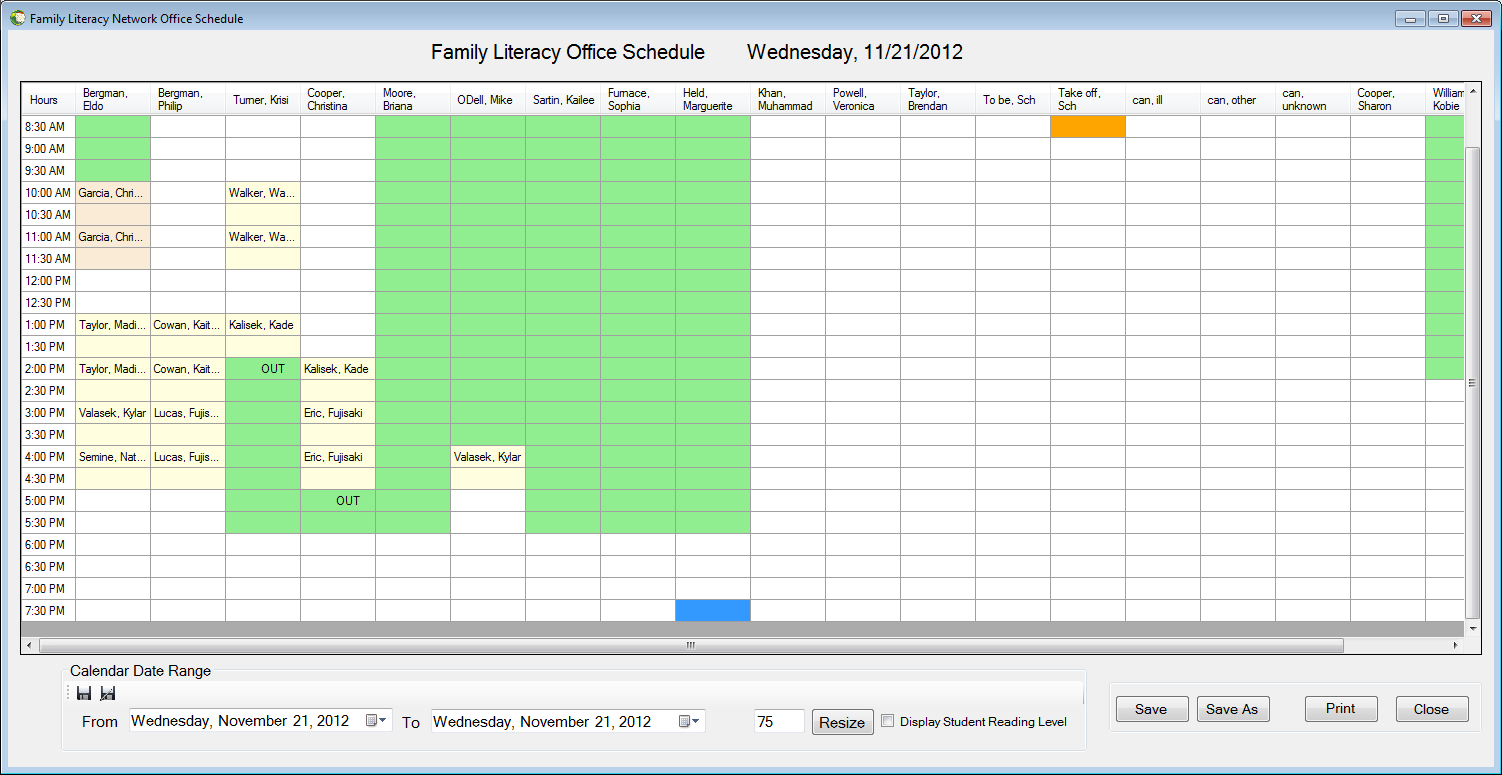
Select a final date from which you would like to stop printing at.

**C) Resize the width of all the columns within the gridview display.**



User can select a value within the range of 10-150 pixels to adjust the width of all the columns in the display grid.

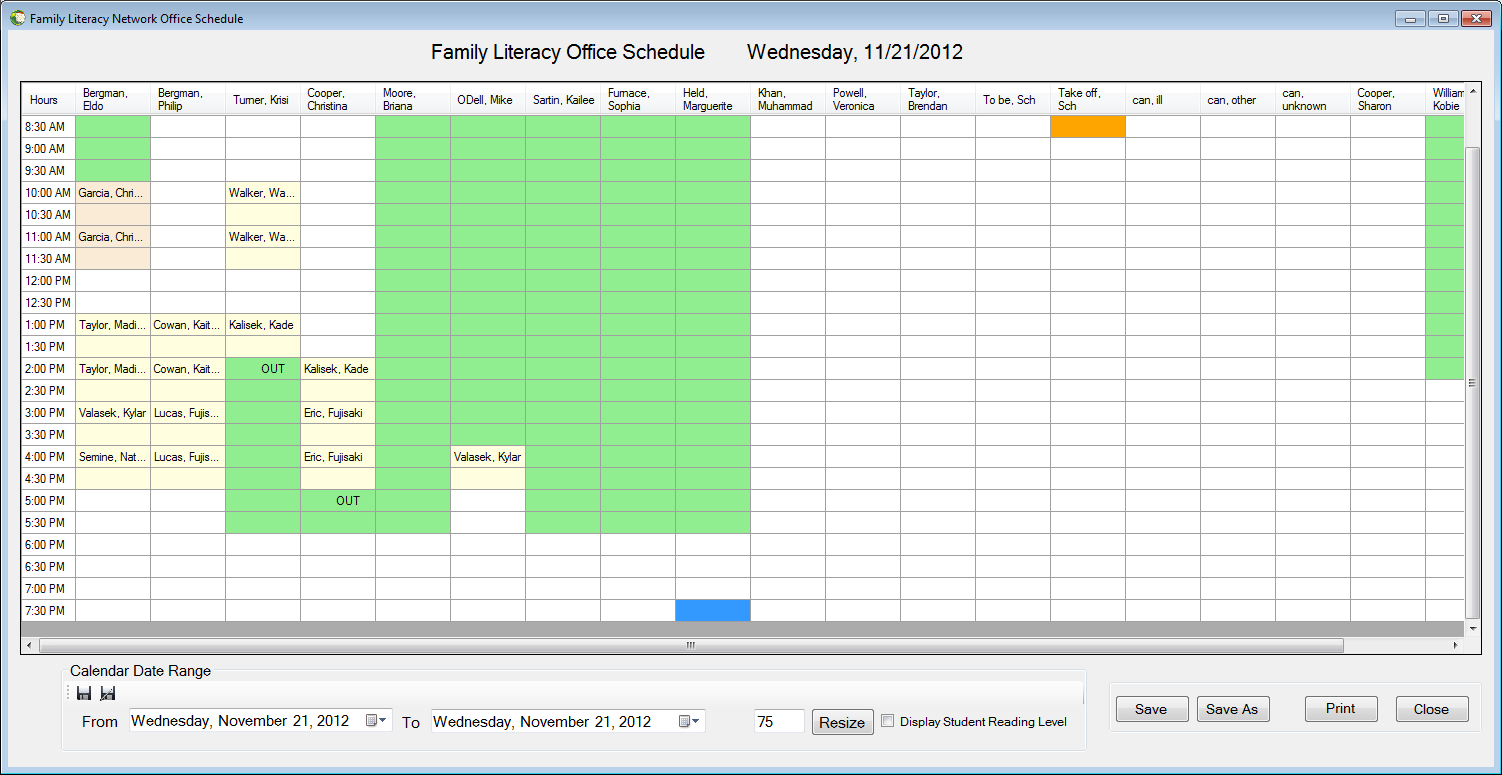
**d)Display Students’ reading level.**



Check the checkbox to display the levels.

Uncheck the checkbox to not display the levels.

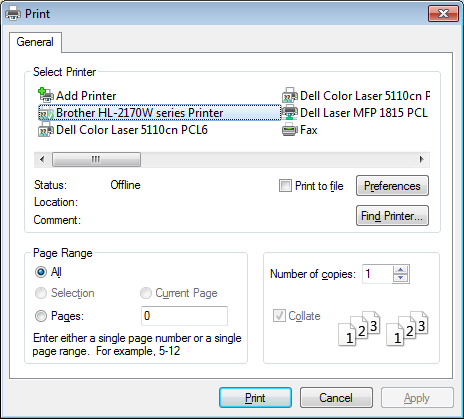
**e) Save/Save-As and Printing area.**



The user has the options of printing to a printer or, saving a digital copy of the office schedule to the hard drive, as well as exiting completely out of the screen.

**1)Save:** Saves the selected date range of office schedules to a predetermined location on the hard drive.

**2)Save As:** Allows the user to change printer options or the location of where the office schedules will be saved on the hard drive.



**Print:** Prints the selected date range of the office schedule to the desired printer

**Close:** Allows the user to exit the office schedule print out screen.