

Consultation evening project v1 {Solution}	2
Consultation evening project v1 {Project}	2
frmAdmin.vb {ProjectItem}	2
frmAdmin {Class}	2
btnAlgorithm_Click {Function}	4
btnBack_Click {Function}	2
btnDay_Click {Function}	2
btnImport_Click {Function}	2
btnResent_Click {Function}	3
frmAvailability.vb {ProjectItem}	5
frmAvailability {Class}	5
btnExit_Click {Function}	12
chkIst_ItemCheck {Function}	5
frmAvailability_Load {Function}	5
frmDaySettings.vb {ProjectItem}	13
frmDaySettings {Class}	13
btnBack_Click {Function}	15
cmbDay_SelectedIndexChanged {Function}	14
cmbEnd_SelectedIndexChanged {Function}	15
cmbNdays_SelectedIndexChanged {Function}	13
cmbStart_SelectedIndexChanged {Function}	14
frmDaySettings_Load {Function}	13
rad10min_CheckedChanged {Function}	14
rad5min_CheckedChanged {Function}	13
secondhalf {Function}	14
frmStart.vb {ProjectItem}	17
frmStart {Class}	17
btnadmin_Click {Function}	18
btnAvailability_Click {Function}	18
btnExit_Click {Function}	18
frmStart_Load {Function}	17
mod1.vb {ProjectItem}	20
mod1 {Module}	20
AppointmentsRec {Struct}	20
DayRec {Struct}	20
Getappointmentrec {Function}	26
GetDay {Function}	26
Getlesson {Function}	26
GetStaff {Function}	25
GetStaffAV {Function}	26
GetStudAV {Function}	26
GetStudent {Function}	25
importLessonsStudent {Function}	23
importLessonStaff {Function}	24
ImportStaff {Function}	22
ImportStudents {Function}	21
LessonRec {Struct}	21
militarytime {Function}	29
populateStartEndDaySettings {Function}	28
Putappointment {Function}	28
Putday {Function}	28
Putlesson {Function}	28
PutStaff {Function}	27
PutStaffAv {Function}	27
PutStudAv {Function}	27
PutStudent {Function}	27
SendEmails2 {Function}	34
sendemailspart1 {Function}	33
StaffAvRec {Struct}	20
StaffRec {Struct}	20
StudAvRec {Struct}	20
StudRec {Struct}	20
TheSortingAlgorithm {Function}	29
My Project {PhysicalFolder}	37
Settings.settings {ProjectItem}	37

```

00001 Public Class frmAdmin
00002
00003 Private Sub btnBack_Click(sender As System.Object, e As System.EventArgs)
00004     Handles btnBack.Click
00005         'opens form start and close the admin form
00006         frmStart.Show()
00007         Me.Close()
00008 End Sub
00009
00010 Private Sub btnDay_Click(ByVal sender As System.Object, ByVal e As System.
00011     EventArgs) Handles btnDay.Click
00012     'opnes frm daysettings
00013     frmDaySettings.Show()
00014 End Sub
00015
00016 Private Sub btnImport_Click(sender As System.Object, e As System.EventArgs)
00017     Handles btnImport.Click
00018     'tries to inport staff. if it failes then due to safeguards in Importstaff()
00019     it must be that there is an error
00020     'with the csv so it reports that and quits the subroutine
00021 Try
00022     Call ImportStaff()
00023 Catch
00024     MsgBox("system cannot find staff.csv" , , "Error" )
00025     Exit Sub
00026 End Try
00027 'if there was no problems with the staff csv but there were issues with the
00028 data then an error message will have been sent
00029 'and the varialbe stopimport will be set to true meaning that the subroutine
00030 should stop.
00031 If stopimport = True Then
00032     stopimport = False
00033     Exit Sub
00034 End If
00035 'tries to inport students. if it failes then due to safeguards in
00036 Importstudents() it must be that there is an error
00037 'with the csv so it reports that and quits the subroutine
00038 Try
00039     Call ImportStudents()
00040 Catch
00041     MsgBox("system cannot find student.csv" , , "Error" )
00042     Exit Sub
00043 End Try
00044 'if there was no problems with the student csv but there were issues with the
00045 data then an error message will have been sent
00046 'and the varialbe stopimport will be set to true meaning that the subroutine
00047 should stop.
00048 If stopimport = True Then
00049     stopimport = False
00050     Exit Sub
00051 End If
00052 'tries to inport student lesson information. if it failes then due to
00053 safeguards in ImportLessonsStudent() it must be that there is an error
00054 'with the csv so it reports that and quits the subroutine
00055 Try
00056     Call importLessonsStudent()
00057 Catch
00058     MsgBox("system cannot find studentclasses.csv" , , "Error" )
00059     Exit Sub
00060 End Try
00061 'if there was no problems with the studentclass csv but there were issues

```

```

00052         with the data then an error message will have been sent
00053         'and the varialbe stopimport will be set to true meaning that the subroutine
00054             should stop.
00055     If stopimport = True Then
00056         stopimport = False
00057         Exit Sub
00058     End If
00059     'tries to inport staff lesson information. if it failes then due to
00060     safeguards in ImportLessonStaff() it must be that there is an error
00061     'with the csv so it reports that and quits the subroutine
00062     Try
00063         Call importLessonStaff()
00064     Catch
00065         MsgBox("system cannot find classslots.csv" , , "Error" )
00066         Exit Sub
00067     End Try
00068     'if there was no problems with the classslots csv but there were issues with
00069     the data then an error message will have been sent
00070     'and the varialbe stopimport will be set to true meaning that the subroutine
00071     should stop.
00072     If stopimport = True Then
00073         stopimport = False
00074         Exit Sub
00075     End If
00076 End Sub
00077
00078 Private Sub btnResent_Click(sender As System.Object, e As System.EventArgs)
00079     Handles btnResent.Click
00080     ' for each dat file it goes through and writes over every record wiht empty
00081     space
00082
00083     'overwrites staff.dat
00084     For counter As Integer = 1 To Nstaff
00085         Staff = Nothing
00086         PutStaff(Staff, counter)
00087     Next
00088
00089     'overwrites staffav.dat
00090     For counter As Integer = 1 To NStaffAv
00091         StaffAv = Nothing
00092         PutStaffAv(StaffAv, counter)
00093     Next
00094
00095     'overwrites student.dat
00096     For counter As Integer = 1 To Nstudents
00097         student = Nothing
00098         PutStudent(student, counter)
00099     Next
00100
00101     'overwrites studentav.dat
00102     For counter As Integer = 1 To NStudAv
00103         StudAv = Nothing
00104         PutStudAv(StudAv, counter)
00105     Next
00106
00107     'overwrites day.dat
00108     For counter As Integer = 1 To NDay
00109         Day = Nothing
00110         Putday(Day, counter)
00111     Next

```

```
00106      'overwrites lesson.dat
00107      For counter As Integer = 1 To Nlesson
00108          Lesson = Nothing
00109          Putlesson(Lesson, counter)
00110      Next
00111
00112      'overwrites appointment.dat
00113      For counter As Integer = 1 To NAppointment
00114          Appointment = Nothing
00115          Putappointment(Appointment, counter)
00116      Next
00117  End Sub
00118
00119  Private Sub btnalgorithm_Click(ByVal sender As System.Object, ByVal e As System.
00120      EventArgs) Handles btnalgorithm.Click
00121      'calls the sorting algortithm which is stored in mod1
00122      Call TheSortingAlogorithm()
00123  End Sub
End Class
```

```

00001 Public Class frmAvailability
00002
00003
00004
00005 Private Sub frmAvailability_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
00006     'hides the start form
00007     frmStart.Hide()
00008     'checks weather the user is a student and files in the name correctly
00009     If usertype = 1 Then
00010         lblname.Text = "welcome " + student.Forename.Trim + " " + student.
            Surname.Trim
00011     ElseIf usertype = 2 Then
00012         lblname.Text = "welcome " + Staff.Forename.Trim + " " + Staff.Surname.
            Trim
00013     End If
00014     'populates the checked list box containing appointment blocks
00015
00016     If Appointmentlength = 5 Then
00017         'appointment length is 5 minuets
00018         For counter As Integer = 1 To NDay
00019             'start and finish times for each day are retrived
00020             Day = GetDay(counter)
00021             For counter2 As Integer = 0 To (Day.finish - Day.Start) \ 6 - 2
00022                 'for each appintment an item is added contianing the
                    appointment number changed into a 24 hour time
00023                 chklstavailability.Items.Add("Day " + counter.ToString + " " +
                    militarytime(Day.Start + (counter2 * 6)) + " to " +
                    militarytime(Day.Start + 12 + (counter2 * 6)))
00024             Next
00025         Next
00026     Else
00027         'appointment lent is 10 minuets
00028         For counter As Integer = 1 To NDay
00029             'start and finish times for each day are retrived
00030             Day = GetDay(counter)
00031             For counter2 As Integer = 0 To (Day.finish - Day.Start) \ 6 - 4
00032                 'for each appintment an item is added contianing the
                    appointment number changed into a 24 hour time
00033                 chklstavailability.Items.Add("Day " + counter.ToString + " " +
                    militarytime(Day.Start + (counter2 * 6)) + " to " +
                    militarytime(Day.Start + 18 + (counter2 * 6)))
00034             Next
00035         Next
00036     End If
00037 End Sub
00038
00039 Private Sub chklst_ItemCheck(ByVal sender As Object, ByVal e As System.Windows.
    Forms.ItemCheckEventArgs) Handles chklstavailability.ItemCheck
00040     'handles the chanaging of availablity for a blcok
00041     Dim parts() As String = Split(chklstavailability.SelectedItem, " ")
00042     'checks if the availability is turned on or off
00043     If chklstavailability.GetItemChecked(chklstavailability.SelectedIndex) =
        False Then
00044         'availability turned on
00045         If usertype = 1 Then
00046             'student
00047             If Appointmentlength = 5 Then
00048                 'appointment length 5
00049                 For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
                    parts(2)) \ 100) * 100)) / 5) To ((parts(2) \ 100) + (((parts(2

```

```

00050    )) - (((parts(2)) \ 100) * 100) / 5) + 5)
00051    'finds relevant stud av records that are in the first half
00052    hour of the block
00053    For counter1 As Integer = 1 To NStudAv
00054        GetStudAV(counter1)
00055        If StudAv.StudNo = student.StudNO And StudAv.Appointment
00056            = counter And StudAv.DayNO = parts(1) Then
00057            Exit For
00058        End If
00059    Next
00060    'sets them to available and addes 1 to their block number
00061    StudAv.available = True
00062    StudAv.Block += 1
00063    PutStudAv(StudAv, StudAv.studAVNO)
00064    Next
00065    For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00066        parts(2)) \ 100) * 100) / 5) + 6 To ((parts(2) \ 100) + (((
00067        parts(2)) - (((parts(3)) \ 100) * 100) / 5) + 11)
00068    'finds relevant stud av records that are in the second half
00069    hour of the block
00070    For counter1 As Integer = 1 To NStudAv
00071        GetStudAV(counter1)
00072        If StudAv.StudNo = student.StudNO And StudAv.Appointment
00073            = counter And StudAv.DayNO = parts(1) Then
00074            Exit For
00075        End If
00076    Next
00077    'sets them to available and addes 2 to their block number
00078    StudAv.available = True
00079    StudAv.Block += 2
00080    PutStudAv(StudAv, StudAv.studAVNO)
00081    Next
00082    Else
00083        'appointment length 10
00084        For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00085            parts(2)) \ 100) * 100) / 5) To ((parts(2) \ 100) + (((parts(2
00086            )) - (((parts(2)) \ 100) * 100) / 5) + 4) Step 2
00087        'finds relevant stud av records that are in the first third
00088        of the block
00089        For counter1 As Integer = 1 To NStudAv
00090            GetStudAV(counter1)
00091            If StudAv.StudNo = student.StudNO And StudAv.Appointment
00092                = counter And StudAv.DayNO = parts(1) Then
00093                Exit For
00094            End If
00095        Next
00096        'sets them to available and addes 1 to their block number
00097        StudAv.available = True
00098        StudAv.Block += 1
00099        PutStudAv(StudAv, StudAv.studAVNO)
00100    Next
00101    For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00102        parts(2)) \ 100) * 100) / 5) + 6 To ((parts(2) \ 100) + (((
00103        parts(2)) - (((parts(2)) \ 100) * 100) / 5) + 10) Step 2
00104    'finds relevant stud av records that are in the second third
00105    of the block
00106    For counter1 As Integer = 1 To NStudAv
00107        GetStudAV(counter1)
00108        If StudAv.StudNo = student.StudNO And StudAv.Appointment
00109            = counter And StudAv.DayNO = parts(1) Then

```

```

00096         Exit For
00097     End If
00098 Next
00099     'sets them to available and addes 2 to their block number
00100     StudAv.available = True
00101     StudAv.Block += 2
00102     PutStudAv(StudAv, StudAv.studAVNO)
00103 Next
00104 For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
    parts(2)) \ 100) * 100)) / 5) + 12 To ((parts(2) \ 100) + (((
    parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 16) Step 2
00105     'finds relevant stud av records that are in the last third
    of the block
00106     For counter1 As Integer = 1 To NStudAv
00107         GetStudAV(counter1)
00108         If StudAv.StudNo = student.StudNO And StudAv.Appointment
            = counter And StudAv.DayNO = parts(1) Then
00109             Exit For
00110         End If
00111     Next
00112     'sets them to available and addes 4 to their block number
00113     StudAv.available = True
00114     StudAv.Block += 4
00115     PutStudAv(StudAv, StudAv.studAVNO)
00116 Next
00117 End If
00118 Else
00119     'staff
00120     'appointment length 5
00121     If Appointmentlength = 5 Then
00122         For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
            parts(2)) \ 100) * 100)) / 5) To ((parts(2) \ 100) + (((parts(2)
            )) - (((parts(2)) \ 100) * 100)) / 5) + 5)
00123             'finds relevant stud av records that are in the first half
            hour of the block
00124             For counter1 As Integer = 1 To NStaffAv
00125                 GetStaffAV(counter1)
00126                 If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
                    Appointment = counter And StaffAv.DayNO = parts(1) Then
00127                     Exit For
00128                 End If
00129             Next
00130             'sets them to available and addes 1 to their block number
00131             StaffAv.Available = True
00132             StaffAv.Block += 1
00133             PutStaffAv(StaffAv, StaffAv.staffAVNO)
00134         Next
00135         For counter As Integer = ((parts(2) \ 100) + (((parts(2)) - (((
            parts(2)) \ 100) * 100)) / 5) + 6) To ((parts(2) \ 100) + (((
            parts(2)) - (((parts(3)) \ 100) * 100)) / 5) + 11))
00136             'finds relevant stud av records that are in the second half
            hour of the block
00137             For counter1 As Integer = 1 To NStaffAv
00138                 GetStaffAV(counter1)
00139                 If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
                    Appointment = counter And StaffAv.DayNO = parts(1) Then
00140                     Exit For
00141                 End If
00142             Next
00143             'sets them to available and addes 2 to their block number
00144             StaffAv.Available = True

```

```

00145         StaffAv.Block += 2
00146         PutStaffAv(StaffAv, StaffAv.staffAVNO)
00147     Next
00148 Else
00149     'appointment length 10
00150
00151     For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00152         parts(2)) \ 100) * 100)) / 5) To ((parts(2) \ 100) + (((
00153         parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 4) Step 2
00154         'finds relevant stud av records that are in the first half
00155         hour of the block
00156         For counter1 As Integer = 1 To NStaffAv
00157             GetStaffAV(counter1)
00158             If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
00159                 Appointment = counter And StaffAv.DayNO = parts(1) Then
00160                 Exit For
00161             End If
00162         Next
00163         'sets them to available and addes 1 to their block number
00164         StaffAv.Available = True
00165         StaffAv.Block += 1
00166         PutStaffAv(StaffAv, StaffAv.staffAVNO)
00167     Next
00168     For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00169         parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((
00170         parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
00171         'finds relevant stud av records that are in the second half
00172         hour of the block
00173         For counter1 As Integer = 1 To NStaffAv
00174             GetStaffAV(counter1)
00175             If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
00176                 Appointment = counter And StaffAv.DayNO = parts(1) Then
00177                 Exit For
00178             End If
00179         Next
00180         'sets them to available and addes 2 to their block number
00181         StaffAv.Available = True
00182         StaffAv.Block += 2
00183         PutStaffAv(StaffAv, StaffAv.staffAVNO)
00184     Next
00185     For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00186         parts(2)) \ 100) * 100)) / 5) + 12 To ((parts(2) \ 100) + (((
00187         parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 16) Step 2
00188         'finds relevant stud av records that are in the third half
00189         hour of the block
00190         For counter1 As Integer = 1 To NStaffAv
00191             GetStaffAV(counter1)
00192             If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
00193                 Appointment = counter And StaffAv.DayNO = parts(1) Then
00194                 Exit For
00195             End If
00196         Next
00197         'sets them to available and addes 4 to their block number
00198         StaffAv.Available = True
00199         StaffAv.Block += 4
00200         PutStaffAv(StaffAv, StaffAv.staffAVNO)
00201     Next
00202 End If
00203 Else
00204     'box is unchecked

```



```

00194 If usertype = 1 Then
00195     'student
00196     If Appointmentlength = 5 Then
00197         'appointment length 5
00198         For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00199             parts(2)) \ 100) * 100)) / 5) To ((parts(2) \ 100) + (((parts(2
00200             )) - (((parts(2)) \ 100) * 100)) / 5) + 5)
00201             'finds relevant stud av records that are in the first half
00202             hour of the block
00203             For counter1 As Integer = 1 To NStudAv
00204                 GetStudAV(counter1)
00205                 If StudAv.StudNo = student.StudNO And StudAv.Appointment
00206                     = counter And StudAv.DayNO = parts(1) Then
00207                     Exit For
00208                 End If
00209             Next
00210             'sets there availability to false if block number is 0 after
00211             decreasing it by 1
00212             StudAv.Block -= 1
00213             If StudAv.Block = 0 Then
00214                 StudAv.available = False
00215             End If
00216             PutStudAv(StudAv, StudAv.studAVNO)
00217         Next
00218         For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00219             parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((
00220             parts(2)) - (((parts(3)) \ 100) * 100)) / 5) + 11)
00221             'finds relevant stud av records that are in the second half
00222             hour of the block
00223             For counter1 As Integer = 1 To NStudAv
00224                 GetStudAV(counter1)
00225                 If StudAv.StudNo = student.StudNO And StudAv.Appointment
00226                     = counter And StudAv.DayNO = parts(1) Then
00227                     Exit For
00228                 End If
00229             Next
00230             'sets there availability to false if block number is 0 after
00231             decreasing it by 2
00232             StudAv.Block -= 2
00233             If StudAv.Block = 0 Then
00234                 StudAv.available = False
00235             End If
00236             PutStudAv(StudAv, StudAv.studAVNO)
00237         Next
00238     Else
00239         'appointment length 10
00240         For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00241             parts(2)) \ 100) * 100)) / 5) To ((parts(2) \ 100) + (((parts(2
00242             )) - (((parts(2)) \ 100) * 100)) / 5) + 4) Step 2
00243             'finds relevant stud av records that are in the first half
00244             hour of the block
00245             For counter1 As Integer = 1 To NStudAv
00246                 GetStudAV(counter1)
00247                 If StudAv.StudNo = student.StudNO And StudAv.Appointment
00248                     = counter And StudAv.DayNO = parts(1) Then
00249                     Exit For
00250                 End If
00251             Next
00252             'sets there availability to false if block number is 0 after
00253             decreasing it by 1
00254             StudAv.Block -= 1

```

```

00240     If StudAv.Block = 0 Then
00241         StudAv.available = False
00242     End If
00243     PutStudAv(StudAv, StudAv.studAVNO)
00244 Next
00245 For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00246     parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((
00247     parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
00248     'finds relevant stud av records that are in the second half
00249     hour of the block
00250     For counter1 As Integer = 1 To NStudAv
00251         GetStudAV(counter1)
00252         If StudAv.StudNo = student.StudNO And StudAv.Appointment
00253             = counter And StudAv.DayNO = parts(1) Then
00254             Exit For
00255         End If
00256     Next
00257     'sets there availability to false if block number is 0 after
00258     decreasing it by 2
00259     StudAv.Block -= 2
00260     If StudAv.Block = 0 Then
00261         StudAv.available = False
00262     End If
00263     PutStudAv(StudAv, StudAv.studAVNO)
00264 Next
00265 For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00266     parts(2)) \ 100) * 100)) / 5) + 12 To ((parts(2) \ 100) + (((
00267     parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 16) Step 2
00268     'finds relevant stud av records that are in the third half
00269     hour of the block
00270     For counter1 As Integer = 1 To NStudAv
00271         GetStudAV(counter1)
00272         If StudAv.StudNo = student.StudNO And StudAv.Appointment
00273             = counter And StudAv.DayNO = parts(1) Then
00274             Exit For
00275         End If
00276     Next
00277     'sets there availability to false if block number is 0 after
00278     decreasing it by 4
00279     StudAv.Block -= 4
00280     If StudAv.Block = 0 Then
00281         StudAv.available = False
00282     End If
00283     PutStudAv(StudAv, StudAv.studAVNO)
00284 Next
00285 End If
00286 Else
00287     'staff
00288     If Appointmentlength = 5 Then
00289         'appointment length 5
00290         For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
00291             parts(2)) \ 100) * 100)) / 5) To ((parts(2) \ 100) + (((parts(2)
00292             )) - (((parts(2)) \ 100) * 100)) / 5) + 5)
00293             'finds relevant stud av records that are in the first half
00294             hour of the block
00295             For counter1 As Integer = 1 To Nstaff
00296                 GetStaffAV(counter1)
00297                 If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
00298                     Appointment = counter And StaffAv.DayNO = parts(1) Then
00299                     Exit For
00300                 End If

```

```

00287     Next
00288     'sets there availability to false if block number is 0 after
00289     decreasing it by 1
00290     StaffAv.Block -= 1
00291     If StaffAv.Block = 0 Then
00292         StaffAv.Available = False
00293     End If
00294     PutStaffAv(StaffAv, StaffAv.staffAVNO)
00295 Next
00296 For counter As Integer = (parts(2) \ 100) + (((parts(2)) - ((
00297     parts(2)) \ 100) * 100)) / 5 + 6 To ((parts(2) \ 100) + (((
00298     parts(2)) - (((parts(3)) \ 100) * 100)) / 5) + 11)
00299     'finds relevant stud av records that are in the second half
00300     hour of the block
00301     For counter1 As Integer = 1 To NStaffAv
00302         GetStaffAV(counter1)
00303         If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
00304             Appointment = counter And StaffAv.DayNO = parts(1) Then
00305             Exit For
00306         End If
00307     Next
00308     'sets there availability to false if block number is 0 after
00309     decreasing it by 2
00310     StaffAv.Block -= 2
00311     If StaffAv.Block = 0 Then
00312         StaffAv.Available = False
00313     End If
00314     PutStaffAv(StaffAv, StaffAv.staffAVNO)
00315 Next
00316 Else
00317     'appointment length 10
00318     For counter As Integer = (parts(2) \ 100) + (((parts(2)) - ((
00319         parts(2)) \ 100) * 100)) / 5 To ((parts(2) \ 100) + (((parts(2
00320         )) - (((parts(2)) \ 100) * 100)) / 5) + 4) Step 2
00321     'finds relevant stud av records that are in the first half
00322     hour of the block
00323     For counter1 As Integer = 1 To NStaffAv
00324         GetStaffAV(counter1)
00325         If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
00326             Appointment = counter And StaffAv.DayNO = parts(1) Then
00327             Exit For
00328         End If
00329     Next
00330     'sets there availability to false if block number is 0 after
00331     decreasing it by 1
00332     StaffAv.Block -= 1
00333     If StaffAv.Block = 0 Then
00334         StaffAv.Available = False
00335     End If
00336     PutStaffAv(StaffAv, StaffAv.staffAVNO)
00337 Next
00338 For counter As Integer = (parts(2) \ 100) + (((parts(2)) - ((
00339     parts(2)) \ 100) * 100)) / 5 + 6 To ((parts(2) \ 100) + (((
00340     parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
00341     'finds relevant stud av records that are in the second half
00342     hour of the block
00343     For counter1 As Integer = 1 To NStaffAv
00344         GetStaffAV(counter1)
00345         If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
00346             Appointment = counter And StaffAv.DayNO = parts(1) Then
00347             Exit For

```

```

00333     End If
00334 Next
00335     'sets there availability to false if block number is 0 after
        decreasing it by 2
00336     StaffAv.Block -= 2
00337     If StaffAv.Block = 0 Then
00338         StaffAv.Available = False
00339     End If
00340     PutStaffAv(StaffAv, StaffAv.staffAVNO)
00341 Next
00342 For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
        parts(2)) \ 100) * 100)) / 5) + 12 To ((parts(2) \ 100) + (((
        parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 16) Step 2
00343     'finds relevant stud av records that are in the first half
        hour of the block
00344     For counter1 As Integer = 1 To NStaffAv
00345         GetStaffAV(counter1)
00346         If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
            Appointment = counter And StaffAv.DayNO = parts(1) Then
00347             Exit For
00348         End If
00349     Next
00350     'sets there availability to false if block number is 0 after
        decreasing it by 4
00351     StaffAv.Block -= 4
00352     If StaffAv.Block = 0 Then
00353         StaffAv.Available = False
00354     End If
00355     PutStaffAv(StaffAv, StaffAv.staffAVNO)
00356 Next
00357 End If
00358 End If
00359 End If
00360 End Sub
00361
00362 Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles btnExit.Click
00363     'closes the availability form and opens the start form
00364     frmStart.Show()
00365     Me.Close()
00366 End Sub
00367 End Class

```

```

00001 Public Class frmDaySettings
00002     'used in the time selecting subroutines
00003     Public change As Boolean = False
00004     Public change1 As Boolean = False
00005     Private Sub rad5min_CheckedChanged(ByVal sender As System.Object, ByVal e As
00006         System.EventArgs) Handles rad5min.CheckedChanged
00007         'checks if the the 5 min radio button is checked and if it is sets the
00008         'appointment length to 5 mins
00009         'otherwise it sets the appointment length to 10 mins
00010         If rad5min.Checked = True Then
00011             Appointmentlength = 5
00012         Else
00013             Appointmentlength = 10
00014         End If
00015         'clears the secondary comboboxes
00016         cmbStart.Items.Clear()
00017         cmbEnd.Items.Clear()
00018         cmbDay.Items.Clear()
00019         'resets the day settings
00020         For counter As Integer = 1 To NDay
00021             Day.DayNO = counter
00022             Day.finish = 288
00023             Day.Start = 0
00024             Putday(Day, Day.DayNO)
00025         Next
00026         'calls the subroutine second half which decided weather to complete the
00027         'other repacutions of changing the appointment length
00028         Call secondhalf()
00029     End Sub
00030     Private Sub frmDaySettings_Load(ByVal sender As System.Object, ByVal e As System
00031         .EventArgs) Handles MyBase.Load
00032         'populates the dropdown combobox with numbers 1 to 255
00033         For counter As Integer = 1 To 255
00034             cmbNdays.Items.Add(counter)
00035         Next
00036     End Sub
00037     Private Sub cmbNdays_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
00038         As System.EventArgs) Handles cmbNdays.SelectedIndexChanged
00039         'records the number of dayz
00040         NDay = cmbNdays.SelectedItem
00041         'clears the secondary comboboxes
00042         cmbStart.Items.Clear()
00043         cmbEnd.Items.Clear()
00044         cmbDay.Items.Clear()
00045         'resets the day settings
00046         For counter As Integer = 1 To NDay
00047             Day.DayNO = counter
00048             Day.finish = 288
00049             Day.Start = 0
00050             Putday(Day, Day.DayNO)
00051         Next
00052         'calls the subroutine second half which decided weather to complete the
00053         'other repacutions of changing the number of days
00054         Call secondhalf()
00055

```

```

00056 End Sub
00057
00058 Public Sub secondhalf()
00059     'sub to check if the appointment length and day numbers have been chosen. if
00060     'so it populates the
00061     'day.dat file
00062     If NDay <> -1 And Appointmentlength <> -1 Then
00063         For counter As Integer = 1 To NDay
00064             Day.DayNO = counter
00065             Day.Start = 0
00066             Day.finish = 288
00067             Putday(Day, counter)
00068         Next
00069     Else
00070         Exit Sub
00071     End If
00072
00073     cmbDay.Visible = True
00074     cmbStart.Visible = True
00075     cmbEnd.Visible = True
00076
00077     'populates the day selector combo box with each of the days
00078     cmbDay.Items.Clear()
00079     For counter = 1 To NDay
00080         cmbDay.Items.Add(counter)
00081     Next
00082 End Sub
00083
00084 Private Sub rad10min_CheckedChanged(ByVal sender As System.Object, ByVal e As
00085     System.EventArgs) Handles rad10min.CheckedChanged
00086     'records the new value of appintment length and checks if its ready to start
00087     'hte second half
00088     Appointmentlength = 10
00089
00090     'clears the secondary comboboxes
00091     cmbStart.Items.Clear()
00092     cmbEnd.Items.Clear()
00093     cmbDay.Items.Clear()
00094     'resets the day settings
00095     For counter As Integer = 1 To NDay
00096         Day.DayNO = counter
00097         Day.finish = 288
00098         Day.Start = 0
00099         Putday(Day, Day.DayNO)
00100     Next
00101     'calls the subroutine second half which decided weather to complete the
00102     'other repacutions of changing the appointment length
00103     Call secondhalf()
00104 End Sub
00105
00106 Private Sub cmbDay_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
00107     As System.EventArgs) Handles cmbDay.SelectedIndexChanged
00108     'calls the subrooutine that will populate cmbstart and cmbend
00109     Call populateStartEndDaySettings()
00110 End Sub
00111
00112 Private Sub cmbStart_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
00113     As System.EventArgs) Handles cmbStart.SelectedIndexChanged
00114     Dim hours As Integer = (cmbStart.SelectedItem) \ 100
00115     Dim minuets As Integer = ((cmbStart.SelectedItem) - ((cmbStart.SelectedItem
00116         ) \ 100) * 100)) / 5

```

```

00110
00111     'eliminates any accidental loops
00112     If change = True Then
00113         change = False
00114         Exit Sub
00115     End If
00116     'sets the start time for the day selected
00117     Day.Start = (hours * 12) + minuets
00118     Day.DayNO = cmbDay.SelectedItem
00119     Putday(Day, Day.DayNO)
00120     'clears the cmbstart andc cmbend
00121     cmbEnd.Items.Clear()
00122     cmbStart.Items.Clear()
00123     'calls the subroutine to populate cmbstart and cmbend
00124     Call populateStartEndDaySettings()
00125     change = True
00126     'selects the start time in cmbstart
00127     cmbStart.SelectedIndex = Day.Start / 6
00128     changel = True
00129     'selects the end time in cmbend
00130     If Appointmentlength = 5 Then
00131         cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 2
00132     Else
00133         cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 3
00134     End If
00135 End Sub
00136
00137 Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
00138     EventArgs) Handles btnBack.Click
00139     'closes the form
00140     Me.Close()
00141 End Sub
00142
00143 Private Sub cmbEnd_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
00144     As System.EventArgs) Handles cmbEnd.SelectedIndexChanged
00145     Dim hours As Integer = (cmbEnd.SelectedItem) \ 100
00146     Dim minuets As Integer = (((cmbEnd.SelectedItem) - (((cmbEnd.SelectedItem) \
00147         100) * 100)) / 5
00148     'eliminates accidental loops
00149     If changel = True Then
00150         changel = False
00151         Exit Sub
00152     End If
00153     'sets the end time for the day selected
00154     Day.finish = (hours * 12) + minuets
00155     Day.DayNO = cmbDay.SelectedItem
00156     Putday(Day, Day.DayNO)
00157     'clears cmbend and cmbstart
00158     cmbEnd.Items.Clear()
00159     cmbStart.Items.Clear()
00160     'calls the subroutine to populate cmbstart and cmbend
00161     Call populateStartEndDaySettings()
00162     change = True
00163     'selsects the start time in cmbstart
00164     cmbStart.SelectedIndex = Day.Start / 6
00165     changel = True
00166     'selects the end time in cmbend
00167     If Appointmentlength = 5 Then
00168         cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 2
00169     Else
00170         cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 3

```

```
00168 |      End If
00169 |    End Sub
00170 |  End Class
```



```

00001 Public Class frmStart
00002
00003 Private Sub frmStart_Load(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles MyBase.Load
00004
00005     'splits up the user data to seperate out the users username
00006     Dim parts() As String = Split(My.User.Name, "\" )
00007     user = parts(1)
00008
00009     'finds the length of each dat file and if they dont exist sets it to -1
00010     Try
00011         Nstaff = FileLen("staff.dat" ) / Len(Staff)
00012     Catch
00013         Nstaff = -1
00014     End Try
00015     Try
00016         NStaffAv = FileLen("staffav.dat" ) / Len(StaffAv)
00017     Catch
00018         NStaffAv = -1
00019     End Try
00020     Try
00021         Nstudents = FileLen("student.dat" ) / Len(student)
00022     Catch
00023         Nstudents = -1
00024     End Try
00025     Try
00026         NStudAv = FileLen("studav.dat" ) / Len(StudAv)
00027     Catch
00028         NStudAv = -1
00029     End Try
00030     Try
00031         NAppointment = FileLen("appointments.dat" ) / Len(Appointment)
00032     Catch
00033         NAppointment = -1
00034     End Try
00035     Try
00036         NDay = FileLen("day.dat" ) / Len(Day)
00037     Catch
00038         NDay = -1
00039     End Try
00040     Try
00041         Nlesson = FileLen("lesson.dat" ) / Len(Lesson)
00042     Catch
00043         Nlesson = -1
00044     End Try
00045
00046
00047
00048     'checks if the user is a student or a staff member
00049     If Len(user) = 6 Then
00050         'user is in the form of a students so the list of students is checked
00051         For counter As Integer = 1 To Nstudents
00052             student = GetStudent(counter)
00053             'checks if the student has been found
00054             If user = student.StudID Then
00055                 'records that the user is a student
00056                 usertype = 1
00057                 Exit For
00058             End If
00059         Next
00060     Else

```

```

00061         'user is not in the form of a student so the list of staff is checked
00062         For counter As Integer = 1 To Nstaff
00063             Staff = GetStaff(counter)
00064             'checks if the user has been found
00065             If user.ToUpper = Staff.staffID.ToUpper Then
00066                 'records that the user is a member of staff
00067                 usertype = 2
00068                 'checks if the user is an admin
00069                 If Staff.admin = True Then
00070                     'makes the admin button visible so that the admin form may
00071                     btnAvailability.Text = "Your Availability"
00072                     btnadmin.Visible = True
00073                 End If
00074                 Exit For
00075             End If
00076         Next
00077     End If
00078     'checks if the system doesnt have any staff. if so then there arnt any
00079     'admins and it checks if the user is in the format of a staff member
00080     'if so then it sends the user to the admin file to set up the system
00081     If FileLen("staff.dat" ) = 0 And IsNumeric(user) = False And Len(user) = 3
00082     Then
00083         frmAdmin.Show()
00084         Me.Close()
00085     End If
00086     'if the usesr has been given a usertype of 0 then he is not in the system so
00087     'access to the other forms is blocked of by making the buttons
00088     'invisible and it then sends an error message sayting they arent in the
00089     'system and advising them to check with an admisistrator if they
00090     'feel it is wrong
00091     If usertype = 0 Then
00092         btnAvailability.Visible = False
00093         MsgBox("Your username is not recognised by the system. If this is an
00094         error please contact the it technicians." , , "ERROR" )
00095     End If
00096 End Sub

00098 Private Sub btnAvailability_Click(ByVal sender As System.Object, ByVal e As
00099 System.EventArgs) Handles btnAvailability.Click
00100     'though the button should only be vissible if hte user is in the system if
00101     'first checks if they are tne sends them to the availablility form
00102     If usertype = 1 Or usertype = 2 Then
00103         frmAvailability.Show()
00104     End If
00105 End Sub

00106 Private Sub btnExit_Click(sender As System.Object, e As System.EventArgs)
00107 Handles btnExit.Click
00108     'closes the form
00109     Me.Close()
00110 End Sub

00111 Private Sub btnadmin_Click(sender As System.Object, e As System.EventArgs)
00112 Handles btnadmin.Click
00113     'opens up the admin form and closes the start form

```

```
00112 |         frmAdmin.Show()  
00113 |         Me.Hide()  
00114 |     End Sub  
00115 | End Class
```

```
00001 Module mod1
00002     'structures
00003
00004     'Student structure
00005
00006     Public Structure StudRec
00007         Public StudNO As Short
00008         <VBFixedString(6)> Public StudID As String
00009         <VBFixedString(20)> Public Forename As String
00010         <VBFixedString(20)> Public Surname As String
00011         Public Year As Byte
00012     End Structure
00013
00014     'staff structure
00015
00016     Public Structure StaffRec
00017         Public StaffNO As Byte
00018         <VBFixedString(3)> Public staffID As String
00019         <VBFixedString(20)> Public Forename As String
00020         <VBFixedString(20)> Public Surname As String
00021         Public admin As Boolean
00022     End Structure
00023
00024     'Student availability structure
00025
00026     Public Structure StudAvRec
00027         public studAVNO as integer
00028         Public Appointment As Integer
00029         Public StudNo As Short
00030         Public DayNO As Byte
00031         Public Block As Byte
00032         Public available As Boolean
00033     End Structure
00034
00035     'staff availability structure
00036
00037     Public Structure StaffAvRec
00038         public staffAVNO as integer
00039         Public Appointment As Integer
00040         Public StaffNO As Byte
00041         Public DayNO As Byte
00042         Public Block As Byte
00043         Public Available As Boolean
00044     End Structure
00045
00046     'day structure
00047
00048     Public Structure DayRec
00049         Public DayNO As Byte
00050         Public Start As Integer
00051         Public finish As Integer
00052     End Structure
00053
00054     'appointments structure
00055
00056     Public Structure AppointmentsRec
00057         Public studNO As integer
00058         Public StaffNO As integer
00059         public start as integer
00060         public day as integer
00061         Public AppointmentNO As Byte
```

```

00062 End Structure
00063
00064 'Lesson structure
00065
00066 Public Structure LessonRec
00067     Public LessonNO As Integer
00068     Public StudNO As Short
00069     Public StaffNO As Short
00070 End Structure
00071
00072 'variables
00073
00074
00075 'general variables
00076 'contains the users username
00077 Public user As String
00078 Public usertype As Byte = 0
00079 Public stopimport As Boolean = False
00080
00081 'student variables
00082
00083 Public student As StudRec = Nothing
00084 Public Nstudents As Integer = -1
00085
00086 'staff variables
00087
00088 Public Staff As StaffRec = Nothing
00089 Public Nstaff As Integer = -1
00090
00091 'student availability variables
00092
00093 Public StudAv As StudAvRec = Nothing
00094 Public studav2 As StudAvRec = Nothing
00095 Public NStudAv As Integer = -1
00096
00097 'staff availablitiy variables
00098
00099 Public StaffAv As StaffAvRec = Nothing
00100 Public NStaffAv As Integer = -1
00101
00102 'Day variables
00103
00104 Public Appointmentlength As Integer = -1
00105 Public Day As DayRec = Nothing
00106 Public NDay As Integer = -1
00107
00108 'Appointment variables
00109
00110 Public Appointment As AppointmentsRec = Nothing
00111 Public NAppointment As Integer = -1
00112
00113 'Lesson variables
00114
00115 Public Lesson As LessonRec = Nothing
00116 Public Nlesson As Integer = -1
00117
00118 'reading csv files and creating dat files
00119
00120 'imports the students into thier dat file
00121 Public Sub ImportStudents()
00122     'opnes up file reader and sets it to read students.csv the file in which the

```

```

00123         student data is stored
00124     Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
00125         "students.csv" )
00126
00127     TextFileReader.TextFieldType = FileIO.FieldType.Delimited
00128     TextFileReader.SetDelimiters(",")
00129
00130     Dim CurrentRow As String()
00131     Dim OnRec As Integer = 0
00132     Dim FileNum As Integer = FreeFile()
00133     'opens file
00134     FileOpen(FileNum, "Student.dat" , OpenMode.Random, OpenAccess.Default,
00135         OpenShare.Default, Len(student))
00136
00137     While Not TextFileReader.EndOfData
00138     Try
00139         CurrentRow = TextFileReader.ReadFields()
00140         If Not CurrentRow Is Nothing Then
00141             OnRec = OnRec + 1
00142             'puts data into the studet structure
00143             With student
00144             Try
00145                 .StudNO = CurrentRow(0)
00146                 .StudID = CurrentRow(1)
00147                 .Surname = CurrentRow(2)
00148                 .Forename = CurrentRow(3)
00149                 .Year = CurrentRow(4)
00150             Catch
00151                 'this is trigered if there was a problem with inputing
00152                 'to the student structure
00153                 'it stops the importing and telles the suer what has
00154                 happened
00155                 stopimport = True
00156                 MsgBox("error with student.csv" )
00157                 Exit Sub
00158             End Try
00159             End With
00160             'puts data from the student structure into the student dat file
00161             FilePut(FileNum, student, OnRec)
00162         End If
00163     Catch ex As _
00164         Microsoft.VisualBasic.FileIO.MalformedLineException
00165         'if error then error message is sent and try ends
00166         MsgBox("Line " & ex.Message & "is not valid and will be skipped." )
00167     End Try
00168 End While
00169
00170 'message box is sent saying that the students are imported and how many
00171 Nstudents = OnRec
00172 MsgBox(Nstudents & " students imported" )
00173 'file is closed
00174 FileClose(FileNum)
00175 TextFileReader.Dispose()
00176 End Sub
00177
00178 'imports staff into thier dat file
00179 Public Sub ImportStaff()
00180     'opens microsoft file reader and sets the file to be read as tutor.csv
00181     Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
00182         "tutor.csv" )
00183     TextFileReader.TextFieldType = FileIO.FieldType.Delimited

```

```

00178 TextFileReader.SetDelimiters(",")
00179
00180 Dim CurrentRow As String()
00181 Dim OnRec As Integer = 0
00182 Dim FileNum As Integer = FreeFile()
00183
00184 'opens the file
00185 FileOpen(FileNum, "Staff.dat", OpenMode.Random, OpenAccess.Default,
    OpenShare.Default, Len(Staff))
00186 Dim parts() As String
00187 While Not TextFileReader.EndOfData
00188     Try
00189         CurrentRow = TextFileReader.ReadFields()
00190         If Not CurrentRow Is Nothing Then
00191             OnRec = OnRec + 1
00192             'puts data into file structure staff
00193             With Staff
00194                 Try
00195                     .StaffNO = CurrentRow(0)
00196                     'forename and surname are saved in the same field on the
                        'parent file so need to be broken up
00197                     parts = Split(CurrentRow(1), " ")
00198                     .Surname = parts(1)
00199                     .Forename = parts(0)
00200                     .staffID = CurrentRow(2)
00201                     If CurrentRow(3) = 0 Then
00202                         .admin = False
00203                     Else :.admin = True
00204                     End If
00205                 Catch
00206                     'this is triggered if there was a problem with inputting
                        'to the staff structure
00207                     'it stops the importing and tells the user what has
                        'happened
00208                     stopimport = True
00209                     MsgBox("error with staff.csv")
00210                     Exit Sub
00211                 End Try
00212             End With
00213             'puts data in file structure staff into the staff dat file
00214             FilePut(FileNum, Staff, OnRec)
00215         End If
00216     Catch ex As _
00217         Microsoft.VisualBasic.FileIO.MalformedLineException
00218         'error in text sends error message and ends try
00219         MsgBox("Line " & ex.Message & "is not valid and will be skipped.")
00220     End Try
00221 End While
00222 Nstaff = OnRec
00223 'sends message box notifying student that staff have been imported and how
    many have been
00224 MsgBox(Nstaff & " Staff imported")
00225 FileClose(FileNum)
00226 TextFileReader.Dispose()
00227 End Sub
00228
00229 Public Sub importLessonsStudent()
00230     'opens microsoft file reader and sets the file to be read as tutor.csv
00231     Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
        "studentclass.csv")
00232     TextFileReader.TextFieldType = FileIO.FieldType.Delimited

```

```

00233     TextFileReader.SetDelimiters(",")
00234
00235     Dim CurrentRow As String()
00236     Dim OnRec As Integer = 0
00237     Dim FileNum As Integer = FreeFile()
00238
00239     'opens the file
00240     FileOpen(FileNum, "lesson.dat", OpenMode.Random, OpenAccess.Default,
00241         OpenShare.Default, Len(Lesson))
00242     While Not TextFileReader.EndOfData
00243     Try
00244         CurrentRow = TextFileReader.ReadFields()
00245         If Not CurrentRow Is Nothing Then
00246             OnRec = OnRec + 1
00247             'puts data into file structure staff
00248             With Lesson
00249             Try
00250                 .LessonNO = OnRec
00251                 .StaffNO = CurrentRow(1)
00252                 .StudNO = CurrentRow(0)
00253             Catch
00254                 'this is triggered if there was a problem with inputting
00255                 'to the lesson structure
00256                 'it stops the importing and tells the user what has
00257                 'happened
00258                 stopimport = True
00259                 MsgBox("error with studentclass.csv")
00260                 Exit Sub
00261             End Try
00262             End With
00263             'puts data in the lesson file structure into the lesson dat file
00264             FilePut(FileNum, Lesson, OnRec)
00265         End If
00266     Catch ex As _
00267         Microsoft.VisualBasic.FileIO.MalformedLineException
00268         'error in text sends error message and ends try
00269         MsgBox("Line " & ex.Message & "is not valid and will be skipped.")
00270     End Try
00271 End While
00272 Nlesson = OnRec
00273 'sends message box notifying admin that student part of lessons have been
00274 imported and how many have been imported
00275 MsgBox("Student half imported")
00276 FileClose(FileNum)
00277 TextFileReader.Dispose()
00278 End Sub
00279
00280 Public Sub importLessonStaff()
00281     'opens microsoft file reader and sets the file to be read as classslots.csv
00282     Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
00283         "classSlots.csv")
00284     TextFileReader.TextFieldType = FileIO.FieldType.Delimited
00285     TextFileReader.SetDelimiters(",")
00286
00287     Dim lastlesson As Integer = -1
00288     Dim currentrow As String() = Nothing
00289     Dim onrec As Integer = 0
00290
00291     OnRec = 0
00292
00293     While Not TextFileReader.EndOfData

```



```

00289     Try
00290         currentrow = TextFileReader.ReadFields()
00291         If (Not currentrow Is Nothing) And (currentrow(0) <> lastlesson.
            ToString) Then
00292             onrec = onrec + 1
00293             lastlesson = currentrow(0)
00294             For counter As Integer = 1 To Nlesson
00295                 Getlesson(counter)
00296                 Try
00297                     If Lesson.StaffNO = currentrow(0) Then
00298                         Lesson.StaffNO = currentrow(3)
00299                     End If
00300                 Catch
00301                     'this is triggered if there was a problem with inputting
                        to the lesson structure
00302                     'it stops the importing and telles the suer what has
                        happened
00303                     stopimport = True
00304                     MsgBox("error with classslots.csv" )
00305                     Exit Sub
00306                 End Try
00307                 Putlesson(Lesson, Lesson.LessonNO)
00308             Next
00309         End If
00310     Catch ex As _
00311         Microsoft.VisualBasic.FileIO.MalformedLineException
00312         'error in text sends error message and ends try
00313         MsgBox("Line " & ex.Message & "is not valid and will be skipped." )
00314     End Try
00315 End While
00316 'sends message box notifying admin that the staff side of lessons have been
    imported
00317 MsgBox("staff half imported" )
00318 TextFileReader.Dispose()
00319
00320 End Sub
00321 ' retrives a student record
00322 Public Function GetStudent(ByVal RecNo As Integer) As StudRec
00323     'function for getting data from student dat file
00324     Dim Filenum As Integer = FreeFile()
00325     GetStudent = Nothing
00326     'opens the student dat file
00327     FileOpen(Filenum, "Student.dat" , OpenMode.Random, OpenAccess.Default,
        OpenShare.Default, Len(student))
00328     'gets data
00329     FileGet(Filenum, GetStudent, RecNo)
00330     'closes file
00331     FileClose(Filenum)
00332 End Function
00333
00334 'retrives a staff record
00335 Public Function GetStaff(ByVal RecNo As Integer) As StaffRec
00336     'function for getting data from staff dat file
00337     Dim Filenum As Integer = FreeFile()
00338     GetStaff = Nothing
00339     'opens the staff dat file
00340     FileOpen(Filenum, "Staff.dat" , OpenMode.Random, OpenAccess.Default,
        OpenShare.Default, Len(Staff))
00341     'gets data
00342     FileGet(Filenum, GetStaff, RecNo)
00343     'closes file

```

```

00344     FileClose(Filename)
00345 End Function
00346
00347 'retrives a student availability record
00348 Public Function GetStudAV(ByVal RecNo As Integer) As StudAvRec
00349     'function for getting data from studAV dat file
00350     Dim Filename As Integer = FreeFile()
00351     GetStudAV = Nothing
00352     'opens the studav dat file
00353     FileOpen(Filename, "StudAV.dat" , OpenMode.Random, OpenAccess.Default,
00354         OpenShare.Default, Len(StudAv))
00355     'gets data
00356     FileGet(Filename, GetStudAV, RecNo)
00357     'closes file
00358     FileClose(Filename)
00359 End Function
00360
00361 'retrives a staff availability record
00362 Public Function GetStaffAV(ByVal RecNo As Integer) As StaffAvRec
00363     'function for getting data from staffav dat file
00364     Dim Filename As Integer = FreeFile()
00365     GetStaffAV = Nothing
00366     'opens the staffav dat file
00367     FileOpen(Filename, "StaffAV.dat" , OpenMode.Random, OpenAccess.Default,
00368         OpenShare.Default, Len(StaffAv))
00369     'gets data
00370     FileGet(Filename, GetStaffAV, RecNo)
00371     'closes file
00372     FileClose(Filename)
00373 End Function
00374
00375 'retrives a day record
00376 Public Function GetDay(ByVal RecNo As Integer) As DayRec
00377     'function for getting data from day dat file
00378     Dim Filename As Integer = FreeFile()
00379     GetDay = Nothing
00380     'opens the day dat file
00381     FileOpen(Filename, "Day.dat" , OpenMode.Random, OpenAccess.Default, OpenShare
00382         .Default, Len(Day))
00383     'gets data
00384     FileGet(Filename, GetDay, RecNo)
00385     'closes file
00386     FileClose(Filename)
00387 End Function
00388
00389 'retrives a lesson record
00390 Public Function GetLesson(ByVal RecNo As Integer) As LessonRec
00391     'function for getting data from lesson dat file
00392     Dim Filename As Integer = FreeFile()
00393     GetLesson = Nothing
00394     'opens the lesson dat file
00395     FileOpen(Filename, "Lesson.dat" , OpenMode.Random, OpenAccess.Default,
00396         OpenShare.Default, Len(Lesson))
00397     'gets data
00398     FileGet(Filename, GetLesson, RecNo)
00399     'closes file
00400     FileClose(Filename)
00401 End Function
00402
00403 'retrives a appointment record

```

```

00400 Public Function Getappointmentrec(ByVal RecNo As Integer) As AppointmentsRec
00401     'function for getting data from Appointments dat file
00402     Dim Filenum As Integer = FreeFile()
00403     Getappointmentrec = Nothing
00404     'opens the appointments dat file
00405     FileOpen(Filenum, "Appointments.dat" , OpenMode.Random, OpenAccess.Default,
00406         OpenShare.Default, Len(Appointment))
00407     'gets data
00408     FileGet(Filenum, Getappointmentrec, RecNo)
00409     'closes file
00410     FileClose(Filenum)
00411 End Function
00412
00413 'overwrites an student onto the student dat file
00414 Public Sub PutStudent(ByVal EditedStudent As StudRec, ByVal RecNo As Integer)
00415     'sub for putting data into the student dat file
00416     Dim Filenum As Integer = FreeFile()
00417     'opens student dat file
00418     FileOpen(Filenum, "Student.dat" , OpenMode.Random, OpenAccess.Default,
00419         OpenShare.Default, Len(student))
00420     'puts data into student dat file
00421     FilePut(Filenum, EditedStudent, RecNo)
00422     'closes student dat file
00423     FileClose(Filenum)
00424 End Sub
00425
00426 'overwrites an staff onto the staff dat file
00427 Public Sub PutStaff(ByVal EditedStaff As StaffRec, ByVal RecNo As Integer)
00428     'sub for putting data into the staff dat file
00429     Dim Filenum As Integer = FreeFile()
00430     'opens staff dat file
00431     FileOpen(Filenum, "Staff.dat" , OpenMode.Random, OpenAccess.Default,
00432         OpenShare.Default, Len(Staff))
00433     'puts data into staff dat file
00434     FilePut(Filenum, EditedStaff, RecNo)
00435     'closes staff dat file
00436     FileClose(Filenum)
00437 End Sub
00438
00439 'overwrites an studavrec onto the studav dat file
00440 Public Sub PutStudAv(ByVal EditedStudAv As StudAvRec, ByVal RecNo As Integer)
00441     'sub for putting data into the studAv dat file
00442     Dim Filenum As Integer = FreeFile()
00443     'opens studAv dat file
00444     FileOpen(Filenum, "StudAv.dat" , OpenMode.Random, OpenAccess.Default,
00445         OpenShare.Default, Len(StudAv))
00446     'puts data into studAv dat file
00447     FilePut(Filenum, EditedStudAv, RecNo)
00448     'closes studAv dat file
00449     FileClose(Filenum)
00450 End Sub
00451
00452 'overwrites an staffAv onto the staffAv dat file
00453 Public Sub PutStaffAv(ByVal EditedStaffAv As StaffAvRec, ByVal RecNo As Integer)
00454     'sub for putting data into the staffAv dat file
00455     Dim Filenum As Integer = FreeFile()
00456     'opens staffAv dat file
00457     FileOpen(Filenum, "StaffAv.dat" , OpenMode.Random, OpenAccess.Default,
00458         OpenShare.Default, Len(StaffAv))
00459     'puts data into staffAv dat file
00460     FilePut(Filenum, EditedStaffAv, RecNo)

```

```

00456         'closes staffAv dat file
00457         FileClose(Filename)
00458     End Sub
00459
00460     'overwrites an Day onto the day dat file
00461     Public Sub Putday(ByVal Editedday As DayRec, ByVal RecNo As Integer)
00462         'sub for putting data into the day dat file
00463         Dim Filename As Integer = FreeFile()
00464         'opens day dat file
00465         FileOpen(Filename, "day.dat" , OpenMode.Random, OpenAccess.Default, OpenShare
00466             .Default, Len(Day))
00467         'puts data into day dat file
00468         FilePut(Filename, Editedday, RecNo)
00469         'closes day dat file
00470         FileClose(Filename)
00471     End Sub
00472
00473     'overwrites an lesson onto the lesson dat file
00474     Public Sub Putlesson(ByVal Editedlesson As LessonRec, ByVal RecNo As Integer)
00475         'sub for putting data into the lesson dat file
00476         Dim Filename As Integer = FreeFile()
00477         'opens lesson dat file
00478         FileOpen(Filename, "Lesson.dat" , OpenMode.Random, OpenAccess.Default,
00479             OpenShare.Default, Len(Lesson))
00480         'puts data into lesson dat file
00481         FilePut(Filename, Editedlesson, RecNo)
00482         'closes lesson dat file
00483         FileClose(Filename)
00484     End Sub
00485
00486     'overwrites an appointment onto the appointment dat file
00487     Public Sub Putappointment(ByVal Editedappointment As AppointmentsRec, ByVal
00488         RecNo As Integer)
00489         'sub for putting data into the appointment dat file
00490         Dim Filename As Integer = FreeFile()
00491         'opens appointment dat file
00492         FileOpen(Filename, "appointments.dat" , OpenMode.Random, OpenAccess.Default,
00493             OpenShare.Default, Len(Appointment))
00494         'puts data into appointment dat file
00495         FilePut(Filename, Editedappointment, RecNo)
00496         'closes the appointment dat file
00497         FileClose(Filename)
00498     End Sub
00499
00500     Public Sub populateStartEndDaySettings()
00501         'populates cmbstart with the times at the required appiontment length apart
00502         Day = GetDay(frmDaySettings.cmbDay.SelectedItem)
00503         If Appointmentlength = 5 Then
00504             'populates cmbstart with 5 min appointments slots
00505             For counter As Integer = 0 To (Day.finish - 12) Step 6
00506                 'each slot contained between the beginin and the finsish time is
00507                 converted into 24hour
00508                 'military style time
00509                 frmDaySettings.cmbStart.Items.Add(militarytime(counter))
00510             Next
00511
00512             'populates the cmbEnd list witht the available times
00513             For counter As Integer = (Day.Start + 12) To 288 Step 6
00514                 'each slot contained between the beginin and the finsish time is
00515                 converted into 24hour
00516                 'military style time

```

```

00511         frmDaySettings.cmbEnd.Items.Add(militarytime(counter))
00512     Next
00513 Else
00514     'populates cmbstart with 10 min appointments slots
00515     For counter As Integer = 0 To (Day.finish - 18) Step 6
00516         'each slot contained between the beginin and the finsish time is
00517         converted into 24hour
00518         'military style time
00519         frmDaySettings.cmbStart.Items.Add(militarytime(counter))
00520     Next
00521     'populates the cmbEnd list witht the available times
00522     For counter As Integer = (Day.Start + 18) To 288 Step 6
00523         'each slot contained between the beginin and the finsish time is
00524         converted into 24hour
00525         'military style time
00526         frmDaySettings.cmbEnd.Items.Add(militarytime(counter))
00527     Next
00528 End If
00529
00530 End Sub
00531 'fucntion that changes a number from 0 to 287 into its corespondng 24 hour
00532 clock time
00533 Public Function militarytime(ByVal timeNO As Integer) As String
00534     Dim hours As String
00535     Dim minuets As String
00536     'works out how many hours there are
00537     hours = (timeNO \ 12).ToString
00538     'puts in the place filler zeroes to keep it 2 characters
00539     If Len(hours) = 1 Then
00540         hours = "0" + hours
00541     ElseIf Len(hours) = 0 Then
00542         hours = "00"
00543     End If
00544     'works out how many minuets remain not counting the hours
00545     minuets = (timeNO - ((timeNO \ 12) * 12))
00546     minuets = minuets * 5
00547     'puts in the place filling zeroes if need to keep it to 2 characters
00548     If Len(minuets) = 1 Then
00549         minuets = "0" + minuets
00550     ElseIf Len(minuets) = 0 Then
00551         minuets = "00"
00552     End If
00553     'puts the 2 halves to gether to be returned
00554     militarytime = hours + minuets
00555 End Function
00556
00557 Public Sub TheSortingAlgorithm()
00558     Dim OnAppointment As Integer = 0
00559     Dim lowerbound As Integer = 0
00560
00561     'for loop goes through every student
00562     For counter1 As Integer = 0 To Nstudents
00563         'gets studnet information
00564         student = GetStudent(counter1)
00565         'finds all the lessons with the student and retives the teacher
00566         For counter2 As Integer = 0 To Nlesson
00567             Lesson = Getlesson(counter2)
00568             If Lesson.StudNO = student.StudNO Then

```

```

00569 Staff = GetStaff(counter2)
00570 'for each students availablity it looks for available spots that
                                'are also available for the teacher
00571 For counter3 As Integer = 0 To NStudAv
00572     StudAv = GetStudAV(counter3)
00573     If StudAv.StudNo = student.StudNO And StudAv.available =
                                True Then
00574         For counter4 As Integer = 0 To NStaffAv
00575             StaffAv = GetStaffAV(counter4)
00576             If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
                                Available = True Then
00577                 'an appointment slot has been found for which
                                both the studen and staff memver are available
                                for
00578                 'the record for the appointment is populated
00579                 Appointment.AppointmentNO = OnAppointment
00580                 Appointment.studNO = StudAv.StudNo
00581                 Appointment.StaffNO = StaffAv.StaffNO
00582                 Appointment.day = StudAv.DayNO
00583                 Appointment.StaffNO = StudAv.Appointment
00584                 'handels block values for the student so as to
                                make sure that the values are only within one
                                block.
00585                 If Appointmentlength = 5 Then
00586                     For counter5 As Integer = 0 To NStudAv
00587                         studav2 = GetStudAV(counter5)
00588                         'checks if teh block value of the
                                current spot is 21 because if it is
                                and
00589                         the other is 20 then they are
                                'not in the same block and it needs to
                                be set as unavailable
00590                         If StudAv.Block = 21 And studav2.Block
                                =
                                20 Then
00591                             studav2.Block = 0
00592                             studav2.available = False
00593                             PutStudAv(studav2,
                                studav2.studAVNO)
00594                             'checks if teh block value of the
                                current spot is 20 because if it
                                is
                                and the other is 21 then they are
                                'not in the same block and it needs
                                to be set as unavailable
00595                             ElseIf StudAv.Block = 20 And studav2.
                                Block = 21 Then
00596                                 studav2.Block = 0
00597                                 studav2.available = False
00598                                 PutStudAv(studav2,
                                studav2.studAVNO)
00599                             End If
00600                         Next
00601                     Else
00602                         'handles 10 min cases
00603                         For counter5 As Integer = 0 To NStudAv
00604                             studav2 = GetStudAV(counter5)
00605                             'checks if teh block value of the
                                current spot is 20 because if it is
                                and

```

```

00608 the other is 22 or 23 then they are
'not in the same block and it needs to  P
be set as unavailable
00609 If StudAv.Block = 20 And (studav2.Block  P
= 22 Or studav2.Block = 23) Then
00610 studav2.Block = 0
00611 studav2.available = False
00612 PutStudAv(studav2,
studav2.studAVNO)
00613 'checks if teh block value of the  P
current spot is 21 because if it  P
is
and the other is 23 then they are
'not in the same block and it needs  P
to be set as unavailable
00615 ElseIf StudAv.Block = 21 And studav2.  P
Block = 23 Then
00616 studav2.Block = 0
00617 studav2.available = False
00618 PutStudAv(studav2,
studav2.studAVNO)
00619 'checks if teh block value of the  P
current spot is 23 because if it  P
is
and the other is 20 or 21 then  P
they
are
'not in the same block and it needs  P
to be set as unavailable
00621 ElseIf StudAv.Block = 23 And (studav2.  P
Block = 20 Or studav2.Block = 21) Then
00622 studav2.Block = 0
00623 studav2.available = False
00624 PutStudAv(studav2,
studav2.studAVNO)
00625 'checks if teh block value of the  P
current spot is 22 because if it  P
is
and the other is 20 then they are
'not in the same block and it needs  P
to be set as unavailable
00627 ElseIf StudAv.Block = 22 And studav2.  P
Block = 20 Then
00628 studav2.Block = 0
00629 studav2.available = False
00630 PutStudAv(studav2,
studav2.studAVNO)
00631 End If
00632 Next
00633 End If
00634 'sets the staffav availablility
00635 StaffAv.Available = False
00636 PutStaffAv(StaffAv, StaffAv.staffAVNO)
00637
00638 'sets the stud blocks and availablility for  P
appointment blocks
00639 'cycles through each studavrecord
00640 For counter5 As Integer = 0 To NStudAv
00641 studav2 = GetStudAV(counter5)
00642 'for appointmetn length 5
00643 If Appointmentlength = 5 Then

```

32 of 39

mod1.vb

15:27:00


```

00687 studav2.available = False
00688 studav2.Block = 0
00689 'too late
00690 } Case Is >= (lowerbound + 18)
00691 studav2.available = False
00692 studav2.Block = 0
00693 'half hour early
00694 } Case (lowerbound - 6) To
(lowerbound
- 1)
00695 studav2.Block = 21
00696 'half hour later
00697 } Case (lowerbound + 6) To
(lowerbound
+ 11)
00698 studav2.Block = 22
00699 'hour early
00700 } Case (lowerbound - 12) To (
lowerbound - 7)
00701 studav2.Block = 20
00702 'hour later
00703 } Case (lowerbound + 12) To (
lowerbound + 17)
00704 studav2.Block = 23
00705 'appointment before
00706 Case (StudAv.Appointment - 1)
00707 studav2.available = False
00708 studav2.Block = 0
00709 'appointment after
00710 Case (StudAv.Appointment + 1)
00711 studav2.available = False
00712 studav2.Block = 0
00713 'the appointment in question
00714 Case StudAv.Appointment
00715 studav2.Block = 0
00716 studav2.available = False
00717 End Select
00718 End If
00719 End If
00720 Next
00721 End If
00722 Next
00723 End If
00724 Next
00725 End If
00726 Next
00727 Next
00728
00729
00730
00731
00732 End Sub
00733 public sub sendemailspart1()
00734 Dim subject As String = "Consultation evening appointments"
00735 Dim body As String = ""
00736 Dim username As String = "sim.bellows@gmail.com"
00737 Dim password As String = "109m3e?!"
00738 Dim recipient As String
00739
00740
00741 'loop that cycles through each student so each gets an email

```

```

00742 For counter1 As Integer = 1 To Nstudents
00743     'loads current students details
00744     student = GetStudent(counter1)
00745     'generates the students school email
00746     recipient = student.StudID + "@WMSF.ac.uk"
00747     'puts the initial greeting for the email and clears the old message
00748     body = "Dear " & student.Forename & " " & student.Surname & vbNewLine
00749         & vbNewLine
00750     'finds each appointment of the student
00751     For counter2 As Integer = 1 To NAppointment
00752         If Appointment.studNO = student.StudNO Then
00753             'gets the name of the member of staff the appointment is with
00754             Staff = GetStaff(Appointment.StaffNO)
00755             'puts in the details of the appointment into the email
00756             body = body & Staff.Forename & " " & Staff.Surname & " " &
00757                 militarytime(Appointment.start) & " day " & Appointment.day &
00758                 vbNewLine
00759         End If
00760     Next
00761     'puts the sign off of the email into the text
00762     body = body & vbNewLine & "thank you very much" & vbNewLine & "simon
00763         bellows" & vbNewLine & vbNewLine & "deputy head"
00764     'calls the routine to send the email
00765     Call SendEmails2("sim.bellows@gmail.com" , subject, body, username,
00766         password, recipient)
00767 Next
00768 'loop that cycles through each staff member so each gets an email
00769 For counter1 As Integer = 1 To Nstaff
00770     'loads current staff members details
00771     Staff = GetStaff(counter1)
00772     'generates the staff school email
00773     recipient = Staff.staffID + "@WMSF.ac.uk"
00774     'puts the intitial greeting for the email and clears the old message
00775     body = "Dear " & Staff.Forename & " " & Staff.Surname & vbNewLine &
00776         vbNewLine
00777     'finds each appoointment of the staff member
00778     For counter2 As Integer = 1 To NAppointment
00779         If Appointment.StaffNO = Staff.StaffNO Then
00780             'gets the name of the student the apppointmetn is with
00781             student = GetStudent(Appointment.studNO)
00782             'puts in the details of the appointment into the email
00783             body = body & student.Forename & " " & student.Surname & " " &
00784                 militarytime(Appointment.start) & " day " & Appointment.day &
00785                 vbNewLine
00786         End If
00787     Next
00788     'puts the sign off of the email into the message
00789     body = body & vbNewLine & "thank you very much" & vbNewLine & "simon
00790         bellows" & vbNewLine & vbNewLine & "deputy head"
00791     'calls the routine to send the email
00792     Call SendEmails2("sim.bellows@gmail.com" , subject, body, username,
00793         password, recipient)
00794 Next
00795 end sub
00796 Public Sub SendEmails2(ByVal FromAddress As String, _
00797     ByVal Subject As String, _
00798     ByVal Body As String, _
00799     ByVal UserName As String, _
00800     ByVal Password As String, _
00801     ByVal recipient As String, _

```

```
00793         Optional ByVal Server As String = "smtp.gmail.com" , _
00794         Optional ByVal Port As Integer = 587)
00795
00796     Dim Email As New MailMessage()
00797
00798     'trys to send the email
00799     Try
00800         Dim SMTPServer As New SmtpClient
00801         'fills in the senders email adress from the fromaddress parameter
00802         Email.From = New MailAddress(FromAddress)
00803         'puts in the recipient for the mail
00804         For Each Recipient As String In Recipients
00805             Email.To.Add(Recipient)
00806         Next
00807         'adds subject body and server, host and such information
00808         Email.Subject = Subject
00809         Email.Body = Body
00810         SMTPServer.Host = Server
00811         SMTPServer.Port = Port
00812         SMTPServer.Credentials = New System.Net.NetworkCredential(UserName,
00813             Password)
00814         SMTPServer.EnableSsl = True
00815         'sends it
00816         SMTPServer.Send(Email)
00817         'clears it
00818         Email.Dispose()
00819         'notification if smtp failed
00820     Catch ex As SmtpException
00821         Email.Dispose()
00822         MsgBox("Sending Email Failed. Smtip Error." )
00823         'notification if portnuimber owas wrong
00824     Catch ex As ArgumentOutOfRangeException
00825         Email.Dispose()
00826         MsgBox("Sending Email Failed. Check Port Number." )
00827         'notification if portnunber is wrong
00828     Catch Ex As InvalidOperationException
00829         Email.Dispose()
00830         MsgBox("Sending Email Failed. Check Port Number." )
00831     End Try
00832 End Sub
End Module
```

```
00001 Imports System
00002 Imports System.Reflection
00003 Imports System.Runtime.InteropServices
00004
00005 ' General Information about an assembly is controlled through the following
00006 ' set of attributes. Change these attribute values to modify the information
00007 ' associated with an assembly.
00008
00009 ' Review the values of the assembly attributes
00010
00011 <Assembly:AssemblyTitle("Consultation evening project v1" )>
00012 <Assembly:AssemblyDescription("")>
00013 <Assembly:AssemblyCompany("Home" )>
00014 <Assembly:AssemblyProduct("Consultation evening project v1" )>
00015 <Assembly:AssemblyCopyright("Copyright © Home 2013" )>
00016 <Assembly:AssemblyTrademark("")>
00017
00018 <Assembly:ComVisible(False)>
00019
00020 'The following GUID is for the ID of the typelib if this project is exposed to COM
00021 <Assembly:Guid("3808a6f4-30cc-4466-883e-d765ea8a54dd" )>
00022
00023 ' Version information for an assembly consists of the following four values:
00024 '
00025 '     Major Version
00026 '     Minor Version
00027 '     Build Number
00028 '     Revision
00029 '
00030 ' You can specify all the values or you can default the Build and Revision Numbers
00031 ' by using the '*' as shown below:
00032 ' <Assembly: AssemblyVersion("1.0.*")>
00033
00034 <Assembly:AssemblyVersion("1.0.0.0" )>
00035 <Assembly:AssemblyFileVersion("1.0.0.0" )>
```

```
00001 <?xml version='1.0' encoding='utf-8'?>
00002 <SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
00003     CurrentProfile="(Default)" UseMySettingsClassName="true">
00004     <Profiles>
00005         <Profile Name="(Default)" />
00006     </Profiles>
00007     <Settings />
00008 </SettingsFile>
```

—
_, 22-25, 34, 35

A

Add, 5, 13, 14, 28, 29, 35
admin, 18, 20, 23
Appointment, 4, 6-12, 17, 20, 21, 27, 28, 30, 32-34
Appointmentlength, 5, 7, 9, 10, 13-15, 21, 28, 30, 31
AppointmentNO, 20, 30
AppointmentsRec, 20, 21, 27, 28
ArgumentOutOfRangeException, 35
AssemblyCompany, 36
AssemblyCopyright, 36
AssemblyDescription, 36
AssemblyFileVersion, 36
AssemblyProduct, 36
AssemblyTitle, 36
AssemblyTrademark, 36
AssemblyVersion, 36
Available, 7, 8, 11, 12, 20, 30, 31
available, 6, 7, 9, 10, 20, 30-33

B

Block, 6-12, 20, 30-33
body, 33, 34
Body, 34, 35
btnadmin, 18
btnadmin_Click, 18
btnalgorithm, 4
btnalgorithm_Click, 4
btnAvailability, 18
btnAvailability_Click, 18
btnBack, 2, 15
btnBack_Click, 2, 15
btnDay, 2
btnDay_Click, 2
btnExit, 12, 18
btnExit_Click, 12, 18
btnImport, 2
btnImport_Click, 2
btnResent, 3
btnResent_Click, 3

C

change, 13, 15
change1, 13, 15
Checked, 13
CheckedChanged, 13, 14
chklst_ItemCheck, 5
chklstavailability, 5
Clear, 13-15
Click, 2-4, 12, 15, 18
Close, 2, 12, 15, 18
cmbDay, 13-15, 28
cmbDay_SelectedIndexChanged, 14
cmbEnd, 13-15, 29
cmbEnd_SelectedIndexChanged, 15
cmbNdays, 13

cmbNdays_SelectedIndexChanged, 13
cmbStart, 13-15, 28, 29
cmbStart_SelectedIndexChanged, 14
ComVisible, 36
counter, 3-14, 17, 18, 25, 28, 29
counter1, 6-12, 29, 34
counter2, 5, 29, 30, 34
counter3, 30
counter4, 30
counter5, 30, 31
Credentials, 35
currentrow, 24, 25
CurrentRow, 22-24

D

Day, 3, 5, 13-15, 17, 21, 26, 28, 29
day, 20, 30, 34
DayNO, 6-15, 20, 30
DayRec, 20, 21, 26, 28
Delimited, 22-24
Dispose, 22-25, 35

E

e, 2-5, 12-15, 17, 18
Editedappointment, 28
Editedday, 28
Editedlesson, 28
EditedStaff, 27
EditedStaffAv, 27
EditedStudAv, 27
EditedStudent, 27
Email, 35
EnableSsl, 35
EndOfData, 22-24
EventArgs, 2-5, 12-15, 17, 18
Ex, 35
ex, 22-25, 35

F

FieldType, 22-24
FileClose, 22-28
FileGet, 25-27
FileIO, 22-25
FileLen, 17, 18
Filenum, 25-28
FileNum, 22-24
FileOpen, 22-28
FilePut, 22-24, 27, 28
finish, 5, 13-15, 20, 28, 29
Forename, 5, 20, 22, 23, 34
Forms, 5
FreeFile, 22-28
frmAdmin, 2, 18, 19
frmAvailability, 5, 18
frmAvailability_Load, 5
frmDaySettings, 2, 13, 28, 29
frmDaySettings_Load, 13
frmStart, 2, 5, 12, 17
frmStart_Load, 17
From, 35
FromAddress, 34, 35

G

Getappointmentrec, 27
GetDay, 5, 26, 28
GetItemChecked, 5
Getlesson, 25, 26, 29
GetStaff, 18, 25, 30, 34
GetStaffAV, 7, 8, 10-12, 26, 30
GetStudAV, 6, 7, 9, 10, 26, 30, 31
GetStudent, 17, 25, 29, 34
Guid, 36

H

Hide, 5, 19
Host, 35
hours, 14, 15, 29

I

importLessonsStudent, 2, 23
importLessonStaff, 3, 24
ImportStaff, 2, 22
ImportStudents, 2, 21
InteropServices, 36
InvalidOperationException, 35
IsNumeric, 18
ItemCheck, 5
ItemCheckEventArgs, 5
Items, 5, 13-15, 28, 29

L

lastlesson, 24, 25
lname, 5
Len, 17, 18, 22-29
Lesson, 4, 17, 21, 24-26, 28, 29
LessonNO, 21, 24, 25
LessonRec, 21, 26, 28
Load, 5, 13, 17
lowerbound, 29, 32, 33

M

MailAddress, 35
MailMessage, 35
MalformedLineException, 22-25
Message, 22-25
Microsoft, 22-25
militarytime, 5, 28, 29, 34
minuets, 14, 15, 29
mod1, 20
MsgBox, 2, 3, 18, 22-25, 35
My, 17

N

Name, 17
NAppointment, 4, 17, 21, 34
NDay, 3, 5, 13, 14, 17, 21
Net, 35
NetworkCredential, 35
Nlesson, 4, 17, 21, 24, 25, 29
Nstaff, 3, 10, 17, 18, 21, 23, 34
NStaffAv, 3, 7, 8, 11, 12, 17, 21, 30
NStudAv, 3, 6, 7, 9, 10, 17, 21, 30, 31
Nstudents, 3, 17, 21, 22, 29, 34

NStudents, 22

O

OnAppointment, 29, 30

OnRec, 22-24

onrec, 24, 25

OpenAccess, 22-28

OpenMode, 22-28

OpenShare, 22-28

P

parts, 5-12, 17, 23

Password, 34, 35

password, 33, 34

populateStartEndDaySettings, 14, 15, 28

Port, 35

Putappointment, 4, 28

Putday, 3, 13-15, 28

Putlesson, 4, 25, 28

PutStaff, 3, 27

PutStaffAv, 3, 7, 8, 11, 12, 27, 31

PutStudAv, 3, 6, 7, 9, 10, 27, 30, 31

PutStudent, 3, 27

R

rad10min, 14

rad10min_CheckedChanged, 14

rad5min, 13

rad5min_CheckedChanged, 13

Random, 22-28

ReadFields, 22-25

Recipient, 35

recipient, 33, 34

Recipients, 35

RecNo, 25-28

Reflection, 36

Runtime, 36

S

secondhalf, 13, 14

SelectedIndex, 5, 15

SelectedIndexChanged, 13-15

SelectedItem, 5, 13-15, 28

Send, 35

SendEmails2, 34

sendemailspart1, 33

sender, 2-5, 12-15, 17, 18

Server, 35

SetDelimiters, 22-24

Show, 2, 12, 18, 19

SmtpClient, 35

SmtpException, 35

SMTPServer, 35

Split, 5, 17, 23

Staff, 3, 5, 7, 8, 10-12, 17, 18, 21, 23, 25, 27, 30, 34

StaffAv, 3, 7, 8, 10-12, 17, 21, 26, 27, 30, 31

staffAVNO, 7, 8, 11, 12, 20, 31

StaffAvRec, 20, 21, 26, 27

staffID, 18, 20, 23, 34

StaffNO, 7, 8, 10-12, 20, 21, 23-25, 30, 34

StaffRec, 20, 21, 25, 27

Start, 5, 13-15, 20, 28, 29

start, 20, 34

stopimport, 2, 3, 21-25

StudAv, 3, 6, 7, 9, 10, 17, 21, 26, 27, 30-33

studav2, 21, 30-33

studAVNO, 6, 7, 9, 10, 20, 30, 31

StudAvRec, 20, 21, 26, 27

student, 3, 5-7, 9, 10, 17, 21, 22, 25, 27, 29, 30, 32, 34

StudID, 17, 20, 22, 34

StudNo, 6, 7, 9, 10, 20, 30, 32

StudNO, 6, 7, 9, 10, 20-22, 24, 29, 30, 32, 34

studNO, 20, 30, 34

StudRec, 20, 21, 25, 27

subject, 33, 34

Subject, 34, 35

Surname, 5, 20, 22, 23, 34

System, 2-5, 12-15, 17, 18, 35, 36

T

Text, 5, 18

TextFieldParser, 22-24

TextFieldType, 22-24

TextFileReader, 22-25

TheSortingAlgorithm, 4, 29

timeNO, 29

ToString, 5, 25, 29

ToUpper, 18

Trim, 5

U

user, 17, 18, 21

User, 17

UserName, 34, 35

username, 33, 34

usertype, 5, 9, 17, 18, 21

V

VBFixedString, 20

vbNewLine, 34

Visible, 14, 18

VisualBasic, 22-25

W

Windows, 5

Y

Year, 20, 22