Maintenance

Contents

List of procedures

frmAdmin

Private Sub btnalgorithm_Click

This is a subroutine that calls "thesortingAlgortithm()", which is a subroutine that generates the appointments. This is triggered by the user clicking on btnalgorithm.
It is used to generate appointments

```
Private Sub btnalgorithm_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnalgorithm.Click
     'calls the sorting algortithm which is stored in mod1
     Call TheSortingAlogorithm()
End Sub
```

Private Sub btnBack_Click

This is a subroutine that is triggered by clicking on btnBack and it closes the current form and opens frmStart
This is used to leave the form and go back to the prior one.

```
Private Sub btnBack_Click(sender As System.Object, e As System.EventArgs) Handles
    btnBack.Click
        'opens form start and close the admin form
        frmStart.Show()
        Me.Close()
    End Sub
```

Private Sub btnDay_Click

This is a subroutine that is triggered by clicking on btnDay and it opens frmdaysettings
This is used to go to the day settings form

```
    Private Sub btnDay_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnDay.Click
        'opnes frm daysettings
        frmDaySettings.Show()
    End Sub
```

Private Sub btnImport_Click

This is a subroutine that is triggered by clicking btnImport and it tries to call the 4 different import subroutines stored in mod1. If any fail due to problems with the csv then it will report an error message and end the subroutine. If it fails due to an error with the data then it will be handled by the having passed back stopimport as true at which point the subroutine will end.
This is used to import student staff and lesson information from csv's to dat

```
Private Sub btnImport_Click(sender As System.Object, e As System.EventArgs)
Handles btnImport.Click
     'tries to inport staff. if it failes then due to safeguards in Importstaff()
     it must be that there is an error
```

```
        'with the csv so it reports that and quits the subroutine
        Try
            Call ImportStaff()
        Catch
            MsgBox("system cannot find staff.csv", , "Error")
            Exit Sub
        End Try
    'if there was no problems with the staff csv but there were isues with the
    data then an error message will have been sent
        'and the varialbe stopimport will be set to true meaning that the subroutine
    should stop.
        If stopimport = True Then
            stopimport = False
            Exit Sub
        End If
    'tries to inport students. if it failes then due to safeguards in
    Importstudents() it must be  that there is an error
        'with the csv so it reports that and quits the subroutine
        Try
            Call ImportStudents()
        Catch
            MsgBox("system cannot find student.csv", , "Error")
            Exit Sub
        End Try
    'if there was no problems with the student csv but there were isues with the
    data then an error message will have been sent
        'and the varialbe stopimport will be set to true meaning that the subroutine
    should stop.
        If stopimport = True Then
            stopimport = False
            Exit Sub
        End If
        'tries to inport student lesson information. if it failes then due to
    safeguards in ImportLessonsStudent() it must be  that there is an error
        'with the csv so it reports that and quits the subroutine
        Try
            Call importLessonsStudent()
        Catch
            MsgBox("system cannot find studentclasses.csv", , "Error")
            Exit Sub
        End Try
    'if there was no problems with the studentclass csv but there were isues
    with the data then an error message will have been sent
        'and the varialbe stopimport will be set to true meaning that the subroutine
    should stop.
        If stopimport = True Then
            stopimport = False
            Exit Sub
        End If
        'tries to inport staff lesson information. if it failes then due to
    safeguards in ImportLessonStaff() it must be  that there is an error
        'with the csv so it reports that and quits the subroutine
        Try
            Call importLessonStaff()
        Catch
            MsgBox("system cannot find classslots.csv", , "Error")
            Exit Sub
        End Try
    'if there was no problems with the classslots csv but there were isues with
    the data then an error message will have been sent
        'and the varialbe stopimport will be set to true meaning that the subroutine
        should stop.
        If stopimport = True Then
            stopimport = False
            Exit Sub
        End If
End Sub
```

Private Sub btnResent_Click

This is a subroutine that is triggerd by the clicking of btnresent which is a typo and should have been btnreset. It using a loop from 0 to the number of records stored in each dat file overwrites every field and record with nothing.
This is used to reset the dat files between parents evenings

```
   Private Sub btnResent_Click(sender As System.Object, e As System.EventArgs)
   Handles btnResent.Click
        ' for each dat file it goes through and writes over every record wiht empty
space

        'overwrites staff.dat
        For counter As Integer = 1 To Nstaff
            Staff = Nothing
            PutStaff(Staff, counter)
        Next

        'overwrites staffav.dat
        For counter As Integer = 1 To NStaffAv
            StaffAv = Nothing
            PutStaffAv(StaffAv, counter)
        Next

        'overwrites student.dat
        For counter As Integer = 1 To Nstudents
            student = Nothing
            PutStudent(student, counter)
        Next

        'overwrites studentav.dat
        For counter As Integer = 1 To NStudAv
            StudAv = Nothing
            PutStudAv(StudAv, counter)
        Next

        'overwrites day.dat
        For counter As Integer = 1 To NDay
            Day = Nothing
            Putday(Day, counter)
        Next

        'overwrites lesson.dat
        For counter As Integer = 1 To Nlesson
            Lesson = Nothing
            Putlesson(Lesson, counter)
        Next

        'overwrites appointment.dat
        For counter As Integer = 1 To NAppointment
            Appointment = Nothing
            Putappointment(Appointment, counter)
        Next
    End Sub
```

frmAvailability

Private Sub btnExit_Click

This subroutine is triggered by clicking on btnexit and it closes the current form and opens frmstart

This is used to leave the availability form and go back to the last form

```
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
    'closes the availability form and opens the start form
    frmStart.Show()
    Me.Close()
End Sub
```

Private Sub chklst_ItemCheck

This subroutine is triggered by the user checking the checkbox on the side of an item in the list. It checks weather the user is a student or a staff member and whether it has been checked or not. It then works out the time as a value from 0 to 277. It then finds the record and sets corrects the block and availability.
This is used to handle the changing of availbablitiy of blocks by students and teachers

```
Private Sub chklst_ItemCheck(ByVal sender As Object, ByVal e As
System.Windows.Forms.ItemCheckEventArgs) Handles chklstavailability.ItemCheck
        'handles the chanaging of availablitity for a blcok
        Dim parts() As String = Split(chklstavailability.SelectedItem, " ")
        'checks if the availability is turned on or off
        If chklstavailability.GetItemChecked(chklstavailability.SelectedIndex) =
False Then
            'availability turned on
            If usertype = 1 Then
                'student
                If Appointmentlength = 5 Then
                    'appointment length 5
                    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)
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 1 to  their block number
                        StudAv.available = True
                        StudAv.Block += 1
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                    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)
                        'finds relevant stud av records that are in the second half
hour of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 2 to  their block number
                        StudAv.available = True
                        StudAv.Block += 2
```

```
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                Else
                    'appointment length 10

                    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) + 4) Step 2
                        'finds relevant stud av records that are in the first third
of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 1 to  their block number
                        StudAv.available = True
                        StudAv.Block += 1
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                    For counter As Integer = (parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
                        'finds relevant stud av records that are in the second
third of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 2 to  their block number
                        StudAv.available = True
                        StudAv.Block += 2
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                    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
                        'finds relevant stud av records that are in the last third
of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 4 to  their block number
                        StudAv.available = True
                        StudAv.Block += 4
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                End If
            Else
                'staff
                'appointment length 5
                If Appointmentlength = 5 Then
                    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)
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
```

```
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 1 to  their block number
                        StaffAv.Available = True
                        StaffAv.Block += 1
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                    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))
                        'finds relevant stud av records that are in the second half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 2 to  their block number
                        StaffAv.Available = True
                        StaffAv.Block += 2
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                Else
                    'appointment length 10

                    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) + 4) Step 2
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 1 to  their block number
                        StaffAv.Available = True
                        StaffAv.Block += 1
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                    For counter As Integer = (parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
                        'finds relevant stud av records that are in the second half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 2 to  their block number
                        StaffAv.Available = True
                        StaffAv.Block += 2
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                    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
```

```
                        'finds relevant stud av records that are in the third half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets them to available and addes 4 to  their block number
                        StaffAv.Available = True
                        StaffAv.Block += 4
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                End If
            End If
        Else
            'box is unchecked
            If usertype = 1 Then
                'student
                If Appointmentlength = 5 Then
                    'appointment lenght 5
                    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)
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 1
                        StudAv.Block -= 1
                        If StudAv.Block = 0 Then
                            StudAv.available = False
                        End If
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                    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)
                        'finds relevant stud av records that are in the second half
hour of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 2
                        StudAv.Block -= 2
                        If StudAv.Block = 0 Then
                            StudAv.available = False
                        End If
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                Else
                    'appointment length 10
                    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) + 4) Step 2
```

```
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 1
                        StudAv.Block -= 1
                        If StudAv.Block = 0 Then
                            StudAv.available = False
                        End If
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                    For counter As Integer = (parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
                        'finds relevant stud av records that are in the second half
hour of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 2
                        StudAv.Block -= 2
                        If StudAv.Block = 0 Then
                            StudAv.available = False
                        End If
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                    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
                        'finds relevant stud av records that are in the third half
hour of the block
                        For counter1 As Integer = 1 To NStudAv
                            GetStudAV(counter1)
                            If StudAv.StudNo = student.StudNO And
StudAv.Appointment = counter And StudAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 4
                        StudAv.Block -= 4
                        If StudAv.Block = 0 Then
                            StudAv.available = False
                        End If
                        PutStudAv(StudAv, StudAv.studAVNO)
                    Next
                End If
            Else
                'staff
                If Appointmentlength = 5 Then
                    'appointment length 5
                    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)
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To Nstaff
```

```
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 1
                        StaffAv.Block -= 1
                        If StaffAv.Block = 0 Then
                            StaffAv.Available = False
                        End If
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                    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)
                        'finds relevant stud av records that are in the second half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 2
                        StaffAv.Block -= 2
                        If StaffAv.Block = 0 Then
                            StaffAv.Available = False
                        End If
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                Else
                    'appointment length 10
                    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) + 4) Step 2
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 1
                        StaffAv.Block -= 1
                        If StaffAv.Block = 0 Then
                            StaffAv.Available = False
                        End If
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                    For counter As Integer = (parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((parts(2)) -
(((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
                        'finds relevant stud av records that are in the second half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
```

```
                        'sets there availability to false if block number is 0
after decreasing it by 2
                        StaffAv.Block -= 2
                        If StaffAv.Block = 0 Then
                            StaffAv.Available = False
                        End If
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                    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
                        'finds relevant stud av records that are in the first half
hour of the block
                        For counter1 As Integer = 1 To NStaffAv
                            GetStaffAV(counter1)
                            If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Appointment = counter And StaffAv.DayNO = parts(1) Then
                                Exit For
                            End If
                        Next
                        'sets there availability to false if block number is 0
after decreasing it by 4
                        StaffAv.Block -= 4
                        If StaffAv.Block = 0 Then
                            StaffAv.Available = False
                        End If
                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
                    Next
                End If
            End If
        End If
    End Sub
```

Private Sub frmAvailability_Load

This is a subroutine that is triggered by frmavailability loading. It populates a label that greets the user and adds a personal touch. It then adds an item for each block to the checked list box. This is used to greet the user and set up the form

```
Private Sub frmAvailability_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'hides the start form
        frmStart.Hide()
        'checks weather the user is a student and files in the name correctly
        If usertype = 1 Then
            lblname.Text = "welcome " + student.Forename.Trim + " " +
student.Surname.Trim
        ElseIf usertype = 2 Then
            lblname.Text = "welcome " + Staff.Forename.Trim + " " +
Staff.Surname.Trim
        End If
        'populates the checked list box containing appointment blocks

        If Appointmentlength = 5 Then
            'appointmetn length is 5 minuets
            For counter As Integer = 1 To NDay
                'start and finish times for each day are retrived
                Day = GetDay(counter)
                For counter2 As Integer = 0 To (Day.finish - Day.Start) \ 6 - 2
                  'for each appintment an item is added contianinng the appointment
                  number changed into a 24 hour time
                  chklstavailability.Items.Add("Day " + counter.ToString + " " +
                  militarytime(Day.Start + (counter2 * 6)) + " to " +
                  militarytime(Day.Start + 12 + (counter2 * 6)))
                Next
```

```
            Next
        Else
            'appointment lent is 10 minuetes
            For counter As Integer = 1 To NDay
                'start and finish times for each day are retrived
                Day = GetDay(counter)
                For counter2 As Integer = 0 To (Day.finish - Day.Start) \ 6 - 4
                  'for each appintment an item is added contianinng the appointment
                  number changed into a 24 hour time
                  chklstavailability.Items.Add("Day " + counter.ToString + " " +
                  militarytime(Day.Start + (counter2 * 6)) + " to " +
                  militarytime(Day.Start + 18 + (counter2 * 6)))
                Next
            Next
        End If
    End Sub
```

frmDaySettings

Private Sub btnBack_Click

this is a subroutine that is triggered by clicking on btnback. it closes the currnet form
this is used to exit the form.

```
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'closes the form
        Me.Close()
    End Sub
```

Private Sub cmbDay_SelectedIndexChanged

This subroutine is triggerd by selecting a value in cmbday. It then calls populatestartenddaysettings.
This is used to trigger the effects of changing the day selected

```
    Private Sub cmbDay_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmbDay.SelectedIndexChanged
        'calls the subrootine that will populate cmbstart and cmbend
        Call populateStartEndDaySettings()
    End Sub
```

Private Sub cmbEnd_SelectedIndexChanged

This subroutine is triggerd by selecting a value in cmbend. It sets the time as the new end time for that day then clears the tiem cmb boxes and calls populatestartenddaysettings to repopulate them but updated. It then selects the current start and end times
This is used to inact any changes made to the end time of a day.

```
    Private Sub cmbEnd_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmbEnd.SelectedIndexChanged
        Dim hours As Integer = (cmbEnd.SelectedItem) \ 100
        Dim minuets As Integer = ((cmbEnd.SelectedItem) - (((cmbEnd.SelectedItem) \
100) * 100)) / 5
        'eliminates accidental loops
        If change1 = True Then
```

```
            change1 = False
            Exit Sub
        End If
        'sets the end time for the day selected
        Day.finish = (hours * 12) + minuets
        Day.DayNO = cmbDay.SelectedItem
        Putday(Day, Day.DayNO)
        'clears cmbend and cmbstart
        cmbEnd.Items.Clear()
        cmbStart.Items.Clear()
        'calls the subroutine to populate cmbstart and cmbend
        Call populateStartEndDaySettings()
        change = True
        'selscts the start time in cmbstart
        cmbStart.SelectedIndex = Day.Start / 6
        change1 = True
        'selects the end time in cmbend
        If Appointmentlength = 5 Then
            cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 2
        Else
            cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 3
        End If
    End Sub
```

Private Sub cmbNdays_SelectedIndexChanged

This subroutine is triggered by changing the selected item in cmbNdays. It clears cmbstart end and day, then it resets the day.dat file then it calls secondhalf to populate the day altering comboboxes

It is used to populate generate the day records and incase it changed again it resets all the day information

```
   Private Sub cmbNdays_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmbNdays.SelectedIndexChanged
        'records the number of dayz
        NDay = cmbNdays.SelectedItem

        'clears the secondary comboboxes
        cmbStart.Items.Clear()
        cmbEnd.Items.Clear()
        cmbDay.Items.Clear()
        'resets the day settings
        For counter As Integer = 1 To NDay
            Day.DayNO = counter
            Day.finish = 288
            Day.Start = 0
            Putday(Day, Day.DayNO)
        Next
        'calls the subroutine second half which decided weather to complete the
other repacutions of changing the number of days
        Call secondhalf()


    End Sub
```

Private Sub cmbStart_SelectedIndexChanged

This subroutine is triggerd by selecting a value in cmbstart. It sets the time as the new start time for that day then clears the tiem cmb boxes and calls populatestartenddaysettings to repopulate them but updated. It then selects the current start and end times

This is used to inact any changes made to the start time of a day.

```
    Private Sub cmbStart_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cmbStart.SelectedIndexChanged
        Dim hours As Integer = (cmbStart.SelectedItem) \ 100
        Dim minuets As Integer = ((cmbStart.SelectedItem) -
(((cmbStart.SelectedItem) \ 100) * 100)) / 5

        'eliminates any accidental loops
        If change = True Then
            change = False
            Exit Sub
        End If
        'sets the start time for the day selected
        Day.Start = (hours * 12) + minuets
        Day.DayNO = cmbDay.SelectedItem
        Putday(Day, Day.DayNO)
        'clears the cmbstart andc cmbend
        cmbEnd.Items.Clear()
        cmbStart.Items.Clear()
        'calls the subroutine to populate cmbstart and cmbend
        Call populateStartEndDaySettings()
        change = True
        'selects the start time in cmbstart
        cmbStart.SelectedIndex = Day.Start / 6
        change1 = True
        'selects the end time in cmbend
        If Appointmentlength = 5 Then
            cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 2
        Else
            cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 3
        End If
    End Sub
```

Private Sub frmDaySettings_Load

This subroutine is triggerd by frmdaysettings loading. I then populates cmbndays with 1 to 255 days.
It is used to set up the basic parts of frmday settings

```
    Private Sub frmDaySettings_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'populates the dropdown combobox with numbers 1 to 255
        For counter As Integer = 1 To 255
            cmbNdays.Items.Add(counter)
        Next
    End Sub
```

Private Sub rad10min_CheckedChanged

This subroutine is triggered by checking the 10 min radio button. Incase it has been done once day information has been input it resets the day records and calls second half to start populatingthe day settings combo boxes that decide times.
It is used to populate generate the day records and incase it changed again it resets all the day information

```
    Private Sub rad10min_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rad10min.CheckedChanged
        'records the new value of appintment length and checks if its ready to
start hte second half
        Appointmentlength = 10

        'clears the secondary comboboxes
```

```
        cmbStart.Items.Clear()
        cmbEnd.Items.Clear()
        cmbDay.Items.Clear()
        'resets the day settings
        For counter As Integer = 1 To NDay
            Day.DayNO = counter
            Day.finish = 288
            Day.Start = 0
            Putday(Day, Day.DayNO)
        Next
        'calls the subroutine second half which decided weather to complete the
other repacutions of changing the appointment length
        Call secondhalf()
    End Sub
```

Private Sub rad5min_CheckedChanged

This subroutine is triggered by checking the 5 min radio button. Incase it has been done once day information has been input it resets the day records and calls second half to start populatingthe day settings combo boxes that decide times.
It is used to populate generate the day records and incase it changed again it resets all the day information

```
    Private Sub rad5min_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rad5min.CheckedChanged
        'checks if the the 5 min radio button is checked and if it is sets the
appointmetn length to 5 mins
        'otherwise it sets the appointment length to 10 mins
        If rad5min.Checked = True Then
            Appointmentlength = 5
        Else
            Appointmentlength = 10
        End If

        'clears the secondary comboboxes
        cmbStart.Items.Clear()
        cmbEnd.Items.Clear()
        cmbDay.Items.Clear()
        'resets the day settings
        For counter As Integer = 1 To NDay
            Day.DayNO = counter
            Day.finish = 288
            Day.Start = 0
            Putday(Day, Day.DayNO)
        Next

        'calls the subroutine second half which decided weather to complete the
other repacutions of changing the appointment length
        Call secondhalf()
    End Sub
```

Public Sub Secondhalf

This subroutine is triggered by changing the appointmentlength or the number of days. It checks if there is an appointment length set and if there is a number days set. If there is it genereates the day records and makes the cmbday , cmbstart and cmbend combo boxes visible and populates cmbdy.
It used to populate the day file and allow for the editing of the time for days.

```
    Public Sub secondhalf()
```

```
        'sub to check if the appointment length and day numbers have been chosen.
if so it populates the
        'day.dat file
        If NDay <> -1 And Appointmentlength <> -1 Then
            For counter As Integer = 1 To NDay
                Day.DayNO = counter
                Day.Start = 0
                Day.finish = 288
                Putday(Day, counter)
            Next
        Else
            Exit Sub
        End If

        cmbDay.Visible = True
        cmbStart.Visible = True
        cmbEnd.Visible = True

        'populates the day selector combo box with each of the days
        cmbDay.Items.Clear()
        For counter = 1 To NDay
            cmbDay.Items.Add(counter)
        Next
    End Sub
```

frmStart

Private Sub btnadmin_Click

This subroutine is triggered by clicking btnadmin. It closes the current form and opens up frm admin
It is used to go to the admin form

```
    Private Sub btnadmin_Click(sender As System.Object, e As System.EventArgs)
Handles btnadmin.Click
        'opens up the admin form and closes the start form
        frmAdmin.Show()
        Me.Hide()
    End Sub
```

Private Sub btnAvailability_Click

This subroutine is triggered by clicking btnavailability. It checks If the users user type Is 1 or 2 if so it opesn up from availability.
It is used to go to the availability form

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

Private Sub btnExit_Click

This subroutine is triggered by clicking btnexit. It closes the current form.
It is used to close the start form and close the program.

```
    Private Sub btnExit_Click(sender As System.Object, e As System.EventArgs)
Handles btnExit.Click
        'closes the form
        Me.Close()
    End Sub
```

Private Sub frmStart_Load

This subroutine is triggered by the loading of frmstart. It retrives the username and sets that to user. It then works out the number of each type of record.it checks the username against the staff and student records to find a match. If the user is a student their user type is set to 1 if it is staff it is set to 2. If there is no staff.dat file then if the user is in the format of a staff member they are sent to the admin form. If they did not match anything then they are told exactly that and advised to see an admin if it is wrong
It is used to set up the start form and work out who is logging on and give them the access they are permited.

```
Private Sub frmStart_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'splits up the user data to seperate out the users username
        Dim parts() As String = Split(My.User.Name, "\")
        user = parts(1)

        'finds the length of each dat file and if they dont exist sets it to -1
        Try
            Nstaff = FileLen("staff.dat") / Len(Staff)
        Catch
            Nstaff = -1
        End Try
        Try
            NStaffAv = FileLen("staffav.dat") / Len(StaffAv)
        Catch
            NStaffAv = -1
        End Try
        Try
            Nstudents = FileLen("student.dat") / Len(student)
        Catch
            Nstudents = -1
        End Try
        Try
            NStudAv = FileLen("studav.dat") / Len(StudAv)
        Catch
            NStudAv = -1
        End Try
        Try
            NAppointment = FileLen("appointments.dat") / Len(Appointment)
        Catch
            NAppointment = -1
        End Try
        Try
            NDay = FileLen("day.dat") / Len(Day)
        Catch
            NDay = -1
        End Try
```

```
        Try
            Nlesson = FileLen("lesson.dat") / Len(Lesson)
        Catch
            Nlesson = -1
        End Try



        'checks if the user is a student or a staff member
        If Len(user) = 6 Then
            'user is in the form of a students so the list of students is checked
            For counter As Integer = 1 To Nstudents
                student = GetStudent(counter)
                'checks if the student has been found
                If user = student.StudID Then
                    'records that the user is a student
                    usertype = 1
                    Exit For
                End If
            Next
        Else
            'user is not in the form of a student so the list of staff is checked
            For counter As Integer = 1 To Nstaff
                Staff = GetStaff(counter)
                'checks if the user has been found
                If user.ToUpper = Staff.staffID.ToUpper Then
                    'records that the user is a member of staff
                    usertype = 2
                    'checks if the user is an admin
                    If Staff.admin = True Then
                        'makes the admin button visible so that the admin form may
be accessed
                        btnAvailability.Text = "Your Availability"
                        btnadmin.Visible = True
                    End If
                    Exit For
                End If
            Next
        End If
        'checks if the system doesnt have any staff. if so then there arnt any
admins and it checks if the user is in the format of a staff member
        'if so then it sends the user to the admin file to set up the system

        If FileLen("staff.dat") = 0 And IsNumeric(user) = False And Len(user) = 3
Then
            frmAdmin.Show()
            Me.Close()
        End If

        'if the usesr has been given a usertype of 0 then he is not in the system
so access to the other forms is blocked of by making the buttons
        'invisible and it then sends an error message sayting they arent in the
system and advising them to check with an admisistrator if they
        'feel it is wrong
        If usertype = 0 Then
            btnAvailability.Visible = False
            MsgBox("Your username is not recognised by the system. If this is an
error please contact the it technicians.", , "ERROR")

        End If


    End Sub
```

mod1

Public Function Getappointmentrec

This function is triggered by being called. It has the parameter recNo which is an integer and stores the record number that the user is looking for. The function getappoinmentrec is in the structure of an appointments rec. it opens the appioment.dat file and goes along a the record number's worth of record lengths and reads in the record then.
This is used to retrieve an appointment record by record number

```
    'retreives a appointment record
    Public Function Getappointmentrec(ByVal RecNo As Integer) As AppointmentsRec
        'function for getting data from Appointments dat file
        Dim Filenum As Integer = FreeFile()
        Getappointmentrec = Nothing
        'opens the appointments dat file
        FileOpen(Filenum, "Appointments.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Appointment))
        'gets data
        FileGet(Filenum, Getappointmentrec, RecNo)
        'closes file
        FileClose(Filenum)
    End Function
```

Public Function GetDay

This function is triggered by being called. It has the parameter recNo which is an integer and stores the record number that the user is looking for. The function getday is in the structure of a day rec. it opens the day.dat file and goes along the record number's worth of record lengths and reads in the record then.
This is used to retrieve a day record by record number

```
    'retrives a day record
    Public Function GetDay(ByVal RecNo As Integer) As DayRec
        'function for getting data from day dat file
        Dim Filenum As Integer = FreeFile()
        GetDay = Nothing
        'opens the day dat file
        FileOpen(Filenum, "Day.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Day))
        'gets data
        FileGet(Filenum, GetDay, RecNo)
        'closes file
        FileClose(Filenum)
    End Function
```

Public Function Getlesson

This function is triggered by being called. It has the parameter recNo which is an integer and stores the record number that the user is looking for. The function getlesson is in the structure of an lesson rec. it opens the lesson.dat file and goes along the record number's worth of record lengths and reads in the record then.
This is used to retrieve a lesson record by record number

```
    'retrives a lesson record
    Public Function Getlesson(ByVal RecNo As Integer) As LessonRec
        'function for getting data from lesson dat file
        Dim Filenum As Integer = FreeFile()
        Getlesson = Nothing
        'opens the lesson dat file
        FileOpen(Filenum, "Lesson.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Lesson))
        'gets data
        FileGet(Filenum, Getlesson, RecNo)
        'closes file
        FileClose(Filenum)
    End Function
```

Public Function GetStaff

This function is triggered by being called. It has the parameter recNo which is an integer and stores the record number that the user is looking for. The function getstaff is in the structure of an staff rec. it opens the staff.dat file and goes along the record number's worth of record lengths and reads in the record then.
This is used to retrieve a staff record by record number

```
    'retrives a staff record
    Public Function GetStaff(ByVal RecNo As Integer) As StaffRec
        'function for getting data from staff dat file
        Dim Filenum As Integer = FreeFile()
        GetStaff = Nothing
        'opens the staff dat file
        FileOpen(Filenum, "Staff.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Staff))
        'gets data
        FileGet(Filenum, GetStaff, RecNo)
        'closes file
        FileClose(Filenum)
    End Function
```

Public Function GetStaffAV

This function is triggered by being called. It has the parameter recNo which is an integer and stores the record number that the user is looking for. The function getstaffav is in the structure of an staffav rec. it opens the staffav.dat file and goes along the record number's worth of record lengths and reads in the record then.
This is used to retrieve a staffav record by record number

```
    'retrives a staff availiability record
    Public Function GetStaffAV(ByVal RecNo As Integer) As StaffAvRec
        'function for getting data from staffav dat file
        Dim Filenum As Integer = FreeFile()
        GetStaffAV = Nothing
        'opens the staffav dat file
        FileOpen(Filenum, "StaffAV.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(StaffAv))
        'gets data
        FileGet(Filenum, GetStaffAV, RecNo)
        'closes file
        FileClose(Filenum)
    End Function
```

Public Function GetStudAV

This function is triggered by being called. It has the parameter recNo which is an integer and stores the record number that the user is looking for. The function getstudav is in the structure of an studav rec. it opens the studav.dat file and goes along the record number's worth of record lengths and reads in the record then.
This is used to retrieve a studav record by record number

```
    'retrives a student availability record
    Public Function GetStudAV(ByVal RecNo As Integer) As StudAvRec
        'function for getting data from studAV dat file
        Dim Filenum As Integer = FreeFile()
        GetStudAV = Nothing
        'opens the studav dat file
        FileOpen(Filenum, "StudAV.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(StudAv))
        'gets data
        FileGet(Filenum, GetStudAV, RecNo)
        'closes file
        FileClose(Filenum)
    End Function
```

Public Function GetStudent

This function is triggered by being called. It has the parameter recNo which is an integer and stores the record number that the user is looking for. The function getstudent is in the structure of an student rec. it opens the student.dat file and goes along the record number's worth of record lengths and reads in the record then.
This is used to retrieve a student record by record number

```
    ' retrives a student record
    Public Function GetStudent(ByVal RecNo As Integer) As StudRec
        'function for getting data from student dat file
        Dim Filenum As Integer = FreeFile()
        GetStudent = Nothing
        'opens the student dat file
        FileOpen(Filenum, "Student.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(student))
        'gets data
        FileGet(Filenum, GetStudent, RecNo)
        'closes file
        FileClose(Filenum)
    End Function
```

Public Sub importLessonsStudent

this is a subroutine that is triggerd by being called. It opens up the visual basic text reader and reads in the csv one line at a time. It then splits up the line by the commas and puts it in to the array of strings current row. It then for each row creates a lesson record with the lesson number the current line the staff number the lesson code and the student number the studno.
It imports lesson information from csv to dat file witht eh staff information a lesson code.

```
Public Sub importLessonsStudent()
        'opens microsoft file reader and sets the file to be read as tutor.csv
```

```
        Dim TextFileReader As New
Microsoft.VisualBasic.FileIO.TextFieldParser("studentclass.csv")
        TextFileReader.TextFieldType = FileIO.FieldType.Delimited
        TextFileReader.SetDelimiters(",")

        Dim CurrentRow As String()
        Dim OnRec As Integer = 0
        Dim FileNum As Integer = FreeFile()

        'opens the file
        FileOpen(FileNum, "lesson.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Lesson))
        While Not TextFileReader.EndOfData
            Try
                CurrentRow = TextFileReader.ReadFields()
                If Not CurrentRow Is Nothing Then
                    OnRec = OnRec + 1
                    'puts data into file structure staff
                    With Lesson
                        Try
                            .LessonNO = OnRec
                            .StaffNO = CurrentRow(1)
                            .StudNO = CurrentRow(0)
                        Catch
                            'this is trigered if there was a problem with inputing
to the lesson structure
                            'it stops the importing and telles the suer what has
happened
                            stopimport = True
                            MsgBox("error with studentclass.csv")
                            Exit Sub
                        End Try
                    End With
                    'puts data in the lesson file structure into the lesson dat
file
                    FilePut(FileNum, Lesson, OnRec)
                End If
            Catch ex As _
            Microsoft.VisualBasic.FileIO.MalformedLineException
                'error in text sends error message and ends try
                MsgBox("Line " & ex.Message & "is not valid and will be skipped.")
            End Try
        End While
        Nlesson = OnRec
        'sends message box notifying admin that student part of lessons have been
imported and how many have been imported
        MsgBox("Student half imported")
        FileClose(FileNum)
        TextFileReader.Dispose()
    End Sub
```

Public Sub importLessonStaff

this is a subroutine that is triggerd by being called. It opens up the visual basic text reader and reads in the csv one line at a time. It then splits up the line by the commas and puts it in to the array of strings current row. It then for each row finds an lesson records with a lesson code the same as that row and then replaces the records value with the staffNO.

It reads a the classslots csv file and finds and inserts the teacher for each lesson.

```
Public Sub importLessonStaff()
        'opens microsoft file reader and sets the file to be read as classslots.csv
```

```
        Dim TextFileReader As New
Microsoft.VisualBasic.FileIO.TextFieldParser("classSlots.csv")
        TextFileReader.TextFieldType = FileIO.FieldType.Delimited
        TextFileReader.SetDelimiters(",")

        Dim lastlesson As Integer = -1
        Dim currentrow As String() = Nothing
        Dim onrec As Integer = 0

        OnRec = 0

        While Not TextFileReader.EndOfData
            Try
                currentrow = TextFileReader.ReadFields()
                If (Not currentrow Is Nothing) And (currentrow(0) <>
lastlesson.ToString) Then
                    onrec = onrec + 1
                    lastlesson = currentrow(0)
                    For counter As Integer = 1 To Nlesson
                        Getlesson(counter)
                        Try
                            If Lesson.StaffNO = currentrow(0) Then
                                Lesson.StaffNO = currentrow(3)
                            End If
                        Catch
                            'this is trigered if there was a problem with inputing
to the lesson structure
                            'it stops the importing and telles the suer what has
happened
                            stopimport = True
                            MsgBox("error with classslots.csv")
                            Exit Sub
                        End Try
                        Putlesson(Lesson, Lesson.LessonNO)
                    Next
                End If
            Catch ex As  _
                    Microsoft.VisualBasic.FileIO.MalformedLineException
                'error in text sends error message and ends try
                MsgBox("Line " & ex.Message & "is not valid and will be skipped.")
            End Try
        End While
        'sends message box notifying admin that the staff side of lessons have been
imported
        MsgBox("staff half imported")
        TextFileReader.Dispose()

    End Sub
```

Public Sub ImportStaff

this is a subroutine that is triggerd by being called. It opens up the visual basic text reader and reads in the tutorcsv one line at a time. It then splits up the line by the commas and puts it in to the array of strings current row. It then for each row creates a staff record with the staff number the current line and the staff id and other pieces of information is from one of the different parts of the row.
It imports staff information from csv to dat file.

```
'imports staff into thier dat file
    Public Sub ImportStaff()
        'opens microsoft file reader and sets the file to be read as tutor.csv
        Dim TextFileReader As New
Microsoft.VisualBasic.FileIO.TextFieldParser("tutor.csv")
```

```
        TextFileReader.TextFieldType = FileIO.FieldType.Delimited
        TextFileReader.SetDelimiters(",")

        Dim CurrentRow As String()
        Dim OnRec As Integer = 0
        Dim FileNum As Integer = FreeFile()

        'opens the file
        FileOpen(FileNum, "Staff.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Staff))
        Dim parts() As String
        While Not TextFileReader.EndOfData
            Try
                CurrentRow = TextFileReader.ReadFields()
                If Not CurrentRow Is Nothing Then
                    OnRec = OnRec + 1
                    'puts data into file structure staff
                    With Staff
                        Try
                            .StaffNO = CurrentRow(0)
                            'forename and surname are saved in the same field on
the parent file so need to be broken up
                            parts = Split(CurrentRow(1), " ")
                            .Surname = parts(1)
                            .Forename = parts(0)
                            .staffID = CurrentRow(2)
                            If CurrentRow(3) = 0 Then
                                .admin = False
                            Else : .admin = True
                            End If
                        Catch
                            'this is trigered if there was a problem with inputing
to the staff structure
                            'it stops the importing and telles the suer what has
happened
                            stopimport = True
                            MsgBox("error with staff.csv")
                            Exit Sub
                        End Try
                    End With
                    'puts data in file structure staff into the staff dat file
                    FilePut(FileNum, Staff, OnRec)
                End If
            Catch ex As _
            Microsoft.VisualBasic.FileIO.MalformedLineException
                'error in text sends error message and ends try
                MsgBox("Line " & ex.Message & "is not valid and will be skipped.")
            End Try
        End While
        Nstaff = OnRec
        'sends message box notifying student that staff have been imported and how
many have been
        MsgBox(Nstaff & " Staff imported")
        FileClose(FileNum)
        TextFileReader.Dispose()
    End Sub
```

Public Sub ImportStudents

this is a subroutine that is triggerd by being called. It opens up the visual basic text reader and reads in the studentscsv one line at a time. It then splits up the line by the commas and puts it in to the array of strings current row. It then for each row creates a student record with the staff number the current line and the

student id and other pieces of information is from one of the different parts of the row.
It imports student information from csv to dat file.

```
'reading csv files and creating dat files

    'imports the students into thier dat file
    Public Sub ImportStudents()
        'opnes up file reader and sets it to read students.csv the file in which
the student data is stored
        Dim TextFileReader As New
Microsoft.VisualBasic.FileIO.TextFieldParser("students.csv")
        TextFileReader.TextFieldType = FileIO.FieldType.Delimited
        TextFileReader.SetDelimiters(",")

        Dim CurrentRow As String()
        Dim OnRec As Integer = 0
        Dim FileNum As Integer = FreeFile()
        'opens file
        FileOpen(FileNum, "Student.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(student))

        While Not TextFileReader.EndOfData
            Try
                CurrentRow = TextFileReader.ReadFields()
                If Not CurrentRow Is Nothing Then
                    OnRec = OnRec + 1
                    'puts data into the studet structure
                    With student
                        Try

                            .StudNO = CurrentRow(0)
                            .StudID = CurrentRow(1)
                            .Surname = CurrentRow(2)
                            .Forename = CurrentRow(3)
                            .Year = CurrentRow(4)
                        Catch
                            'this is trigered if there was a problem with inputing
to the student structure
                            'it stops the importing and telles the suer what has
happened
                            stopimport = True
                            MsgBox("error with student.csv")
                            Exit Sub
                        End Try

                    End With
                    'puts data from the student structure into the student dat file
                    FilePut(FileNum, student, OnRec)
                End If
            Catch ex As _
                Microsoft.VisualBasic.FileIO.MalformedLineException
                'if error then error message is sent and try ends
                MsgBox("Line " & ex.Message & "is not valid and will be skipped.")
            End Try
        End While
        'message box is sent saying that the students are imported and how many
        Nstudents = OnRec
        MsgBox(NStudents & " students imported")
        'file is closed
        FileClose(FileNum)
        TextFileReader.Dispose()
    End Sub
```

Public Function militarytime

This is a function that is triggered by being called. It has a perameter timeNO which is an appointment time as a number from 0 to 287. It works out how many 12's go into time number and that is the number of hours. It then fills out that string with 0's if needed to make it 2 characters. For the minuets it does the same except it uses the remmander after divideing by 12 and multiplies it by 5. The string " hours" and "minuets" are then joined together and returned
It converts a number into a time in 24 hour time.

```
    'fucntion that changes a number from 0 to 287 into its coresponidng 24 hour
clock time
    Public Function militarytime(ByVal timeNO As Integer) As String
        Dim hours As String
        Dim minuets As String

        'works out how many hours there are
        hours = (timeNO \ 12).ToString
        'puts in the place filler zeroes to keep it 2 characters
        If Len(hours) = 1 Then
            hours = "0" + hours
        ElseIf Len(hours) = 0 Then
            hours = "00"
        End If
        'works out how many minuets remain not counting the hours
        minuets = (timeNO - ((timeNO \ 12) * 12))
        minuets = minuets * 5
        'puts in the place filling zeroes if need to keep it to 2 characters
        If Len(minuets) = 1 Then
            minuets = "0" + minuets
        ElseIf Len(minuets) = 0 Then
            minuets = "00"
        End If
        'puts the 2 halves to gether to be returned
        militarytime = hours + minuets
    End Function
```

Public Sub populateStartEndDaySettings

This is a subroutine that is triggered by being called. It checks the appointment length and gets the dayrecord and populates cmbstart from 0 to day.finish less 12 if appoiment length is 5 or 18 if 10. It then populates cmbstart from cmb.start +12 or 18 depending on length to 288. For each time it is passed through military time to give a time the user can comprehend.
It populates cmbstart and cmbend on frmdaysettings.

```
Public Sub populateStartEndDaySettings()
        'populates cmbstart with the times at the required appiontment length apart
        Day = GetDay(frmDaySettings.cmbDay.SelectedItem)
        If Appointmentlength = 5 Then
            'populates cmbstart with 5 min appointments slots
            For counter As Integer = 0 To (Day.finish - 12) Step 6
                'each slot contained between the beginin and the finsish time is
converted into 24hour
                'military style time
                frmDaySettings.cmbStart.Items.Add(militarytime(counter))
            Next

            'populates the cmbEnd list witht the available times
            For counter As Integer = (Day.Start + 12) To 288 Step 6
```

```
                'each slot contained between the beginin and the finsish time is
converted into 24hour
                'military style time
                frmDaySettings.cmbEnd.Items.Add(militarytime(counter))
            Next
        Else
            'populates cmbstart with 10 min appointments slots
            For counter As Integer = 0 To (Day.finish - 18) Step 6
                'each slot contained between the beginin and the finsish time is
converted into 24hour
                'military style time
                frmDaySettings.cmbStart.Items.Add(militarytime(counter))
            Next

            'populates the cmbEnd list witht the available times
            For counter As Integer = (Day.Start + 18) To 288 Step 6
                'each slot contained between the beginin and the finsish time is
converted into 24hour
                'military style time
                frmDaySettings.cmbEnd.Items.Add(militarytime(counter))
            Next

        End If

    End Sub
```

Public Sub Putappointment

This subroutine is triggered by being called. It has the perameter editedappointment and recNO. It opens the dat file goes along to the right record and overwrites the old record with editedappointment. It then closes the file.
It puts writes an edited record saving the changes.

```
    'overwrites an appointment onto the appointment dat file
    Public Sub Putappointment(ByVal Editedappointment As AppointmentsRec, ByVal
RecNo As Integer)
        'sub for putting data into the appointment dat file
        Dim Filenum As Integer = FreeFile()
        'opens appointment dat file
        FileOpen(Filenum, "appointments.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Appointment))
        'puts data into appointment dat file
        FilePut(Filenum, Editedappointment, RecNo)
        'closes the appointment dat file
        FileClose(Filenum)
    End Sub
```

Public Sub Putday

This subroutine is triggered by being called. It has the perameter editedday and recNO. It opens the dat file goes along to the right record and overwrites the old record with editedday. It then closes the file.
It puts writes an edited record saving the changes.

```
    'overwrites an Day onto the day dat file
    Public Sub Putday(ByVal Editedday As DayRec, ByVal RecNo As Integer)
        'sub for putting data into the day dat file
        Dim Filenum As Integer = FreeFile()
        'opens day dat file
        FileOpen(Filenum, "day.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Day))
```

```
        'puts data into day dat file
        FilePut(Filenum, Editedday, RecNo)
        'closes day dat file
        FileClose(Filenum)
    End Sub
```

Public Sub Putlesson

This subroutine is triggered by being called. It has the perameter editedlesson and recNO. It opens the dat file goes along to the right record and overwrites the old record with editedlesson. It then closes the file.
It puts writes an edited record saving the changes.

```
    'overwrites an lesson onto the lesson dat file
    Public Sub Putlesson(ByVal Editedlesson As LessonRec, ByVal RecNo As Integer)
        'sub for putting data into the lesson dat file
        Dim Filenum As Integer = FreeFile()
        'opens lesson dat file
        FileOpen(Filenum, "Lesson.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Lesson))
        'puts data into lesson dat file
        FilePut(Filenum, Editedlesson, RecNo)
        'closes lesson dat file
        FileClose(Filenum)
    End Sub
```

Public Sub PutStaff

This subroutine is triggered by being called. It has the perameter editedstaff and recNO. It opens the dat file goes along to the right record and overwrites the old record with editedstaff. It then closes the file.
It puts writes an edited record saving the changes.

```
    'overwrites an staff onto the staff dat file
    Public Sub PutStaff(ByVal EditedStaff As StaffRec, ByVal RecNo As Integer)
        'sub for putting data into the staff dat file
        Dim Filenum As Integer = FreeFile()
        'opens staff dat file
        FileOpen(Filenum, "Staff.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(Staff))
        'puts data into staff dat file
        FilePut(Filenum, EditedStaff, RecNo)
        'closes staff dat file
        FileClose(Filenum)
    End Sub
```

Public Sub PutStaffAv

This subroutine is triggered by being called. It has the perameter editedstaffav and recNO. It opens the dat file goes along to the right record and overwrites the old record with editedstaffav. It then closes the file.
It puts writes an edited record saving the changes.

```
    'overwrites an staffAv onto the staffAv dat file
    Public Sub PutStaffAv(ByVal EditedStaffAv As StaffAvRec, ByVal RecNo As
Integer)
        'sub for putting data into the staffAv dat file
        Dim Filenum As Integer = FreeFile()
```

```
        'opens staffAv dat file
        FileOpen(Filenum, "StaffAv.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(StaffAv))
        'puts data into staffAv dat file
        FilePut(Filenum, EditedStaffAv, RecNo)
        'closes staffAv dat file
        FileClose(Filenum)
    End Sub
```

Public Sub PutStudAv

This subroutine is triggered by being called. It has the perameter editedstudav and recNO. It opens the dat file goes along to the right record and overwrites the old record with editedstudav. It then closes the file.
It puts writes an edited record saving the changes.

```
    'overwrites an studavrec onto the studav dat file
    Public Sub PutStudAv(ByVal EditedStudAv As StudAvRec, ByVal RecNo As Integer)
        'sub for putting data into the studAv dat file
        Dim Filenum As Integer = FreeFile()
        'opens studAv dat file
        FileOpen(Filenum, "StudAv.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(StudAv))
        'puts data into studAv dat file
        FilePut(Filenum, EditedStudAv, RecNo)
        'closes studAv dat file
        FileClose(Filenum)
    End Sub
```

Public Sub PutStudent

This subroutine is triggered by being called. It has the perameter editedstudent and recNO. It opens the dat file goes along to the right record and overwrites the old record with editedstudent. It then closes the file.
It puts writes an edited record saving the changes.

```
    'overwrites an student onto the student dat file
    Public Sub PutStudent(ByVal EditedStudent As StudRec, ByVal RecNo As Integer)
        'sub for putting data into the student dat file
        Dim Filenum As Integer = FreeFile()
        'opens student dat file
        FileOpen(Filenum, "Student.dat", OpenMode.Random, OpenAccess.Default,
OpenShare.Default, Len(student))
        'puts data into student dat file
        FilePut(Filenum, EditedStudent, RecNo)
        'closes student dat file
        FileClose(Filenum)
    End Sub
```

Public Sub SendEmails2

This subroutine is triggerd by being called. It has 8 parametres. Fromaddress which is the address it is being sent from subject which is the subject of the email. Body which is the main text, username and password which are for logging onto the account the emails are being sent from. Recipant the recivers email. Server and port are optional and will likely not used. It links to an smtpsever in this case gmail and then constructs and sends the email from the parametres.

Its sends the email that has been designed in send email part 1

```
Public Sub SendEmails2(ByVal FromAddress As String, _
                       ByVal Subject As String, _
                          ByVal Body As String, _
                            ByVal UserName As String, _
                            ByVal Password As String, _
                            ByVal recipient As String, _
                            Optional ByVal Server As String = "smtp.gmail.com", _
                            Optional ByVal Port As Integer = 587)

        Dim Email As New MailMessage()

        'trys to send the email
        Try
            Dim SMTPServer As New SmtpClient
            'fills in the senders email adress from the fromaddress parameter
            Email.From = New MailAddress(FromAddress)
            'puts in the recipent for the mail
            For Each Recipient As String In Recipients
                Email.To.Add(Recipient)
            Next
            'adds subject body and server, host and such information
            Email.Subject = Subject
            Email.Body = Body
            SMTPServer.Host = Server
            SMTPServer.Port = Port
            SMTPServer.Credentials = New System.Net.NetworkCredential(UserName,
Password)
            SMTPServer.EnableSsl = True
            'sends it
            SMTPServer.Send(Email)
            'clears it
            Email.Dispose()
            'notificaltion if smtp failed
        Catch ex As SmtpException
            Email.Dispose()
            MsgBox("Sending Email Failed. Smtp Error.")
            'notification if portnuimber owas wrong
        Catch ex As ArgumentOutOfRangeException
            Email.Dispose()
            MsgBox("Sending Email Failed. Check Port Number.")
            'notification if portnunber is wrong
        Catch Ex As InvalidOperationException
            Email.Dispose()
            MsgBox("Sending Email Failed. Check Port Number.")
        End Try
    End Sub
```

Public Sub sendemailspart1

This subroutine is triggerd by being called. I goes through every student and staff member and finds all there appointments. It then puts the time and the names of the other part of each appointment and the time into a variable used to contain the body of an email. Once it has all f the appoitments compiled into the body string it calls the second email function.

It finds all the appionments for every one and in passes them on to be emailed.

```
public sub sendemailspart1()
        Dim subject As String = "Consultation evening appointments"
        Dim body As String = ""
        Dim username As String = "sim.bellows@gmail.com"
        Dim password As String = "l09m3e?!"
        Dim recipient As String
```

```
        'loop that cycles through each student so each gets an email
        For counter1 As Integer = 1 To Nstudents
            'loads current students details
            student = GetStudent(counter1)
            'generates the students school email
            recipient = student.StudID + "@WMSF.ac.uk"
            'puts the initial greeting for the email and clears the old message
            body = "Dear " & student.Forename & " " & student.Surname & vbNewLine &
vbNewLine
            'finds each appointment of the student
            For counter2 As Integer = 1 To NAppointment
                If Appointment.studNO = student.StudNO Then
                    'gets the name of the member of staff the appointment is with
                    Staff = GetStaff(Appointment.StaffNO)
                    'puts in the details of the appiontment into the email
                    body = body & Staff.Forename & " " & Staff.Surname & "" &
militarytime(Appointment.start) & " day " & Appointment.day & vbNewLine
                End If
            Next
            'puts the sign off of the email into the text
            body = body & vbNewLine & "thank you very much" & vbNewLine & "simon
bellows" & vbNewLine & vbNewLine & "deputy head"
            'calls the routine to send the email
            Call SendEmails2("sim.bellows@gmail.com", subject, body, username,
password, recipient)
        Next
        'loop that cycles through each staff member so each gets an email
        For counter1 As Integer = 1 To Nstaff
            'loads current staff members details
            Staff = GetStaff(counter1)
            'genereates the staff school email
            recipient = Staff.staffID + "@WMSF.ac.uk"
            'puts the intitial greeting for the email and clears the old message
            body = "Dear " & Staff.Forename & " " & Staff.Surname & vbNewLine &
vbNewLine
            'finds each appoointment of the staff member
            For counter2 As Integer = 1 To NAppointment
                If Appointment.StaffNO = Staff.StaffNO Then
                    'gets the name of the student the appointmetn is with
                    student = GetStudent(Appointment.studNO)
                    'puts in the details of the appointment into the email
                    body = body & student.Forename & " " & student.Surname & "" &
militarytime(Appointment.start) & " day " & Appointment.day & vbNewLine
                End If
            Next
            'puts the sign off of the email into the message
            body = body & vbNewLine & "thank you very much" & vbNewLine & "simon
bellows" & vbNewLine & vbNewLine & "deputy head"
            'calls the routine to send the email
            Call SendEmails2("sim.bellows@gmail.com", subject, body, username,
password, recipient)
        Next

    end sub
```

Public Sub TheSortingAlogorithm

This subroutine is triggerd by being called. For each student it finds all the lessons they have. It then loads up information on the first teacher. Its finds the first time they are mutualy available and generates that appointment. It changes block and availability information for the students other availability

records so that no appointment can be outside a single block can be used. Then it repeats with the next lesson.

It is a subroutine that is used to divvy out appointments for students.

```
 Public Sub TheSortingAlogorithm()
        Dim OnAppointment As Integer = 0
        Dim lowerbound As Integer = 0

        'for loop goes through every student
        For counter1 As Integer = 0 To Nstudents
            'gets studnet information
            student = GetStudent(counter1)
            'finds all the lessons with the student and retives the teacher
            For counter2 As Integer = 0 To Nlesson
                Lesson = Getlesson(counter2)
                If Lesson.StudNO = student.StudNO Then
                    Staff = GetStaff(counter2)
                    'for each students availablity it looks for avaliable spots
that are also available for the teacher
                    For counter3 As Integer = 0 To NStudAv
                        StudAv = GetStudAV(counter3)
                        If StudAv.StudNo = student.StudNO And StudAv.available =
True Then
                            For counter4 As Integer = 0 To NStaffAv
                                StaffAv = GetStaffAV(counter4)
                                If StaffAv.StaffNO = Staff.StaffNO And
StaffAv.Available = True Then
                                    'an appointment slot has been found for which
both the studen and staff memver are available for
                                    'the record for the appointment is populated
                                    Appointment.AppointmentNO = OnAppointment
                                    Appointment.studNO = StudAv.StudNo
                                    Appointment.StaffNO = StaffAv.StaffNO
                                    Appointment.day = StudAv.DayNO
                                    Appointment.StaffNO = StudAv.Appointment
                                    'handels block values for the student so as to
make sure that the values are only within one block.
                                    If Appointmentlength = 5 Then
                                        For counter5 As Integer = 0 To NStudAv
                                            studav2 = GetStudAV(counter5)
                                            'checks if teh block value of the
current spot is 21 because if it is and the other is 20 then they are
                                            'not in the same block and it needs to
be set as unavailable
                                            If StudAv.Block = 21 And studav2.Block
= 20 Then
                                                studav2.Block = 0
                                                studav2.available = False
                                                PutStudAv(studav2,
studav2.studAVNO)
                                            '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
                                            ElseIf StudAv.Block = 20 And
studav2.Block = 21 Then
                                                studav2.Block = 0
                                                studav2.available = False
                                                PutStudAv(studav2,
studav2.studAVNO)
                                            End If
                                        Next

                                    Else
                                        'handles 10 min cases
                                        For counter5 As Integer = 0 To NStudAv
```

```
                                        studav2 = GetStudAV(counter5)
                                        'checks if teh block value of the
current spot is 20 because if it is and the other is 22 or 23 then they are
                                        'not in the same block and it needs to
be set as unavailable
                                        If StudAv.Block = 20 And (studav2.Block
= 22 Or studav2.Block = 23) Then
                                            studav2.Block = 0
                                            studav2.available = False
                                            PutStudAv(studav2,
studav2.studAVNO)
                                            'checks if teh block value of the
current spot is 21 because if it is and the other is 23 then they are
                                            'not in the same block and it needs
to be set as unavailable
                                        ElseIf StudAv.Block = 21 And
studav2.Block = 23 Then
                                            studav2.Block = 0
                                            studav2.available = False
                                            PutStudAv(studav2,
studav2.studAVNO)
                                            'checks if teh block value of the
current spot is 23 because if it is and the other is 20 or 21 then they are
                                            'not in the same block and it needs
to be set as unavailable
                                        ElseIf StudAv.Block = 23 And
(studav2.Block = 20 Or studav2.Block = 21) Then
                                            studav2.Block = 0
                                            studav2.available = False
                                            PutStudAv(studav2,
studav2.studAVNO)
                                            'checks if teh block value of the
current spot is 22 because if it is and the other is 20 then they are
                                            'not in the same block and it needs
to be set as unavailable
                                        ElseIf StudAv.Block = 22 And
studav2.Block = 20 Then
                                            studav2.Block = 0
                                            studav2.available = False
                                            PutStudAv(studav2,
studav2.studAVNO)
                                        End If
                                    Next
                                End If
                                'sets the staffav availablitity
                                StaffAv.Available = False
                                PutStaffAv(StaffAv, StaffAv.staffAVNO)

                                'sets the stud blocks and availablitity for
appointment blocks
                                'cycles through each studavrecord
                                For counter5 As Integer = 0 To NStudAv
                                    studav2 = GetStudAV(counter5)
                                    'for appointmetn length 5
                                    If Appointmentlength = 5 Then
                                        'works out the first appointment of the
half hour
                                        lowerbound = (studav2.Appointment \ 6)
* 6
                                        'checks if the studav is not already
been set in the earlier check and if it is for the right student
                                        If studav2.StudNo = student.StudNO And
StudAv.Block <> 21 Or 20 Then
                                            Select Case studav2.Appointment
                                                'too early
                                                Case Is <= (lowerbound - 7)
                                                    studav2.available = False
                                                    studav2.Block = 0
```

```
                                                    'too late
                                                Case Is >= (lowerbound + 12)
                                                    studav2.available = False
                                                    studav2.Block = 0
                                                    'half hour early
                                                Case (lowerbound - 6) To
(lowerbound - 1)
                                                    studav2.Block = 20
                                                    'half hour later
                                                Case (lowerbound + 6) To
(lowerbound + 11)
                                                    studav2.Block = 21
                                                    'appointmetn before
                                                Case (StudAv.Appointment - 1)
                                                    studav2.available = False
                                                    studav2.Block = 0
                                                    'appointment after
                                                Case (StudAv.Appointment + 1)
                                                    studav2.available = False
                                                    studav2.Block = 0
                                                    'the appointmetn in
question
                                                Case StudAv.Appointment
                                                    studav2.Block = 0
                                                    studav2.available = False
                                            End Select


                                        End If
                                    Else
                                        'works out the first appointment of the
half hour
                                        lowerbound = (studav2.Appointment \ 6)
* 6
                                        'checks if the studav is not already
been set in the earlier check and if it is for the right student
                                        If studav2.StudNo = student.StudNO And
StudAv.Block <> 21 Or 20 Then
                                            Select Case studav2.Appointment
                                                'too early
                                                Case Is <= (lowerbound - 13)
                                                    studav2.available = False
                                                    studav2.Block = 0
                                                    'too late
                                                Case Is >= (lowerbound + 18)
                                                    studav2.available = False
                                                    studav2.Block = 0
                                                    'half hour early
                                                Case (lowerbound - 6) To
(lowerbound - 1)
                                                    studav2.Block = 21
                                                    'half hour later
                                                Case (lowerbound + 6) To
(lowerbound + 11)
                                                    studav2.Block = 22
                                                    'hour early
                                                Case (lowerbound - 12) To
(lowerbound - 7)
                                                    studav2.Block = 20
                                                    'hour later
                                                Case (lowerbound + 12) To
(lowerbound + 17)
                                                    studav2.Block = 23
                                                    'appointmetn before
                                                Case (StudAv.Appointment - 1)
                                                    studav2.available = False
                                                    studav2.Block = 0
                                                    'appointment after
```

```
                                                    Case (StudAv.Appointment + 1)
                                                        studav2.available = False
                                                        studav2.Block = 0
                                                        'the appointmetn in
question
                                                    Case StudAv.Appointment
                                                        studav2.Block = 0
                                                        studav2.available = False
                                                End Select
                                            End If
                                        End If
                                    Next
                                End If
                            Next
                        End If
                    Next
                End If
            Next
        Next



    End Sub
```

List of variables

Global variables

```
Public Appointment As AppointmentsRec = Nothing
  mod1
```

This variable is used to store appointment records when they are being manipulated by the program

```
Public Appointmentlength As Integer = -1
  mod1
```

This variable is used to store the appointment length of the parents evening.

```
Public Day As DayRec = Nothing
  mod1
```

This variable is used to store day records when they are being manipulated by the program

```
Public Lesson As LessonRec = Nothing
  mod1
```

This variable is used to store lesson records when they are being manipulated by the program

```
Public NAppointment As Integer = -1
  mod1
```

This variable is used to store the number of appointment records in the appointment.dat file

```
Public NDay As Integer = -1
  mod1
```

This variable is used to store the number of day records in the day.dat file

```
Public Nlesson As Integer = -1
  mod1
```

This variable is used to store the number of lesson records in the lesson.dat file

```
Public Nstaff As Integer = -1
  mod1
```

This variable is used to store the number of staff records in the staff.dat file

```
Public NStaffAv As Integer = -1
```
mod1

This variable is used to store the number of staff availability records in the staffav.dat file

```
Public NStudAv As Integer = -1
```
mod1

This variable is used to store the number of student availability records in the studav.dat file

```
Public Nstudents As Integer = -1
  mod1
```

This variable is used to store the number of student records in the students.dat file

```
Public Staff As StaffRec = Nothing
  mod1
```

This variable is used to store staff records when they are being manipulated by the program

```
Public StaffAv As StaffAvRec = Nothing
  mod1
```

This variable is used to store staff availability records when they are being manipulated by the program

```
Public stopimport As Boolean = False
  mod1
```

this variable is used to store whether or not to stop an import due to errors in the data being inputed

```
Public StudAv As StudAvRec = Nothing
  mod1
```

This variable is used to store student availability records when they are being manipulated by the program

```
Public studav2 As StudAvRec = Nothing
  mod1
```

This variable is used to store student availability records when they are being manipulated by the program when 2 student availability records are being compared

```
Public student As StudRec = Nothing
  mod1
```

This variable is used to store student records when they are being manipulated by the program

```
Public user As String
  mod1
```

This variable is used to store the string that the user used as his network username

```
Public usertype As Byte = 0
  mod1
```

This variable is used to store the user type of the user, 0 is not on the system 1 is a student and 2 is a student.

Module-level variables

frmDaySettings

```
Public change As Boolean = False
```

This variable is used to store whether or not the change was done by the user or by a piece of code triggering it self

```
Public change1 As Boolean = False
```

This variable is used to store whether or not the change was done by the user or by a piece of code triggering it self

mod1.AppointmentsRec

```
Public AppointmentNO As Byte
Public day As integer
Public StaffNO As integer
Public start As integer
Public studNO As integer
```

This set of variables is saved as a structure called appointments rec and is the format of an appointment record.

mod1.DayRec

```
Public DayNO As Byte
Public finish As Integer
Public Start As Integer
```

This set of variables is saved as a structure called dayrec and is the format of a day record.

mod1.LessonRec

```
Public LessonNO As Integer
Public StaffNO As Short
Public StudNO As Short
```

This set of variables is saved as a structure called lessonrec and is the format of a lesson record.

mod1.StaffAvRec

```
Public Appointment As Integer
Public Available As Boolean
Public Block As Byte
Public DayNO As Byte
Public staffAVNO As integer
Public StaffNO As Byte
```

This set of variables is saved as a structure called staffavrec and is the format of a staff availablility record.

mod1.StaffRec

```
Public admin As Boolean
Public Forename As String
Public staffID As String
Public StaffNO As Byte
Public Surname As String
```

This set of variables is saved as a structure called staffrec and is the format of a staff record.

mod1.StudAvRec

```
Public Appointment As Integer
Public available As Boolean
Public Block As Byte
Public DayNO As Byte
Public studAVNO As integer
Public StudNo As Short
```

This set of variables is saved as a structure called studavrec and is the format of a student avability record.

mod1.StudRec

```
Public Forename As String
Public StudID As String
Public StudNO As Short
Public Surname As String
Public Year As Byte
```

This set of variables is saved as a structure called studrec and is the format of a student record.

Data type summary

Variables and parameters

Type	As Type	$%&!#@^	Implicit	Total
Boolean	6			6
Byte	10			10
Object	1			1

Other	47			47
Short	4			4
String, $	22			22
String(), $()	7			7
Structure	15			15
integer	84			84
Total	196	0	0	196

Summary	**Total**	**%**
Numeric	104	53%
String	22	11%
Array	7	4%
Other	63	32%
Total	**196**	**100%**

Glossary

As Type: Variable declared with regular 'As Datatype' clause.
$%&!#@^: Variable declared with type character.
Implicit: Variable declared with no explicit datatype. Compiler decides type.

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

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

```
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.
             EventArgs) Handles btnalgorithm.Click
00120            'calls the sorting algortithm which is stored in mod1
00121            Call TheSortingAlogorithm()
00122        End Sub
00123    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              'appointmetn 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 contianinng 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 minuetes
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 contianinng 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 availablitity 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
```

```
        )) - (((parts(2)) \ 100) * 100)) / 5) + 5)
00050       'finds relevant stud av records that are in the first half
             hour of the block
00051       For counter1 As Integer = 1 To NStudAv
00052           GetStudAV(counter1)
00053           If StudAv.StudNo = student.StudNO And StudAv.Appointment
                 = counter And StudAv.DayNO = parts(1) Then
00054               Exit For
00055           End If
00056       Next
00057       'sets them to available and addes 1 to  their block number
00058       StudAv.available = True
00059       StudAv.Block += 1
00060       PutStudAv(StudAv, StudAv.studAVNO)
00061   Next
00062   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)
00063       'finds relevant stud av records that are in the second half
             hour of the block
00064       For counter1 As Integer = 1 To NStudAv
00065           GetStudAV(counter1)
00066           If StudAv.StudNo = student.StudNO And StudAv.Appointment
                 = counter And StudAv.DayNO = parts(1) Then
00067               Exit For
00068           End If
00069       Next
00070       'sets them to available and addes 2 to  their block number
00071       StudAv.available = True
00072       StudAv.Block += 2
00073       PutStudAv(StudAv, StudAv.studAVNO)
00074   Next
00075 Else
00076   'appointment length 10
00077
00078   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) + 4) Step 2
00079       'finds relevant stud av records that are in the first third
             of the block
00080       For counter1 As Integer = 1 To NStudAv
00081           GetStudAV(counter1)
00082           If StudAv.StudNo = student.StudNO And StudAv.Appointment
                 = counter And StudAv.DayNO = parts(1) Then
00083               Exit For
00084           End If
00085       Next
00086       'sets them to available and addes 1 to  their block number
00087       StudAv.available = True
00088       StudAv.Block += 1
00089       PutStudAv(StudAv, StudAv.studAVNO)
00090   Next
00091   For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
         parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((
         parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
00092       'finds relevant stud av records that are in the second third
             of the block
00093       For counter1 As Integer = 1 To NStudAv
00094           GetStudAV(counter1)
00095           If StudAv.StudNo = student.StudNO And StudAv.Appointment
                 = 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)) - (((
                          parts(2)) \ 100) * 100)) / 5) To ((parts(2) \ 100) + (((parts(2
                          )) - (((parts(2)) \ 100) * 100)) / 5) + 4) Step 2
00152                        'finds relevant stud av records that are in the first half
                              hour of the block
00153                        For counter1 As Integer = 1 To NStaffAv
00154                            GetStaffAV(counter1)
00155                            If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
                                  Appointment = counter And StaffAv.DayNO = parts(1) Then
00156                                Exit For
00157                            End If
00158                        Next
00159                        'sets them to available and addes 1 to  their block number
00160                        StaffAv.Available = True
00161                        StaffAv.Block += 1
00162                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
00163                    Next
00164                    For counter As Integer = (parts(2) \ 100) + (((parts(2)) - (((
                          parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((
                          parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
00165                        'finds relevant stud av records that are in the second half
                              hour of the block
00166                        For counter1 As Integer = 1 To NStaffAv
00167                            GetStaffAV(counter1)
00168                            If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
                                  Appointment = counter And StaffAv.DayNO = parts(1) Then
00169                                Exit For
00170                            End If
00171                        Next
00172                        'sets them to available and addes 2 to  their block number
00173                        StaffAv.Available = True
00174                        StaffAv.Block += 2
00175                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
00176                    Next
00177                    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
00178                        'finds relevant stud av records that are in the third half
                              hour of the block
00179                        For counter1 As Integer = 1 To NStaffAv
00180                            GetStaffAV(counter1)
00181                            If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
                                  Appointment = counter And StaffAv.DayNO = parts(1) Then
00182                                Exit For
00183                            End If
00184                        Next
00185                        'sets them to available and addes 4 to  their block number
00186                        StaffAv.Available = True
00187                        StaffAv.Block += 4
00188                        PutStaffAv(StaffAv, StaffAv.staffAVNO)
00189                    Next
00190                End If
00191            End If
00192        Else
00193            'box is unchecked
```

```
00194                If usertype = 1 Then
00195                    'student
00196                    If Appointmentlength = 5 Then
00197                        'appointment lenght 5
00198                        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)
00199                            'finds relevant stud av records that are in the first half
                                 hour of the block
00200                            For counter1 As Integer = 1 To NStudAv
00201                                GetStudAV(counter1)
00202                                If StudAv.StudNo = student.StudNO And StudAv.Appointment
                                     = counter And StudAv.DayNO = parts(1) Then
00203                                    Exit For
00204                                End If
00205                            Next
00206                            'sets there availability to false if block number is 0 after
                                 decreasing it by 1
00207                            StudAv.Block -= 1
00208                            If StudAv.Block = 0 Then
00209                                StudAv.available = False
00210                            End If
00211                            PutStudAv(StudAv, StudAv.studAVNO)
00212                        Next
00213                        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)
00214                            'finds relevant stud av records that are in the second half
                                 hour of the block
00215                            For counter1 As Integer = 1 To NStudAv
00216                                GetStudAV(counter1)
00217                                If StudAv.StudNo = student.StudNO And StudAv.Appointment
                                     = counter And StudAv.DayNO = parts(1) Then
00218                                    Exit For
00219                                End If
00220                            Next
00221                            'sets there availability to false if block number is 0 after
                                 decreasing it by 2
00222                            StudAv.Block -= 2
00223                            If StudAv.Block = 0 Then
00224                                StudAv.available = False
00225                            End If
00226                            PutStudAv(StudAv, StudAv.studAVNO)
00227                        Next
00228                    Else
00229                        'appointment length 10
00230                        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) + 4) Step 2
00231                            'finds relevant stud av records that are in the first half
                                 hour of the block
00232                            For counter1 As Integer = 1 To NStudAv
00233                                GetStudAV(counter1)
00234                                If StudAv.StudNo = student.StudNO And StudAv.Appointment
                                     = counter And StudAv.DayNO = parts(1) Then
00235                                    Exit For
00236                                End If
00237                            Next
00238                            'sets there availability to false if block number is 0 after
                                 decreasing it by 1
00239                            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)) - (((
                                     parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + (((
                                     parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
00246                                    'finds relevant stud av records that are in the second half
                                         hour of the block
00247                                    For counter1 As Integer = 1 To NStudAv
00248                                        GetStudAV(counter1)
00249                                        If StudAv.StudNo = student.StudNO And StudAv.Appointment
                                             = counter And StudAv.DayNO = parts(1) Then
00250                                            Exit For
00251                                        End If
00252                                    Next
00253                                    'sets there availability to false if block number is 0 after
                                         decreasing it by 2
00254                                    StudAv.Block -= 2
00255                                    If StudAv.Block = 0 Then
00256                                        StudAv.available = False
00257                                    End If
00258                                    PutStudAv(StudAv, StudAv.studAVNO)
00259                                Next
00260                                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
00261                                    'finds relevant stud av records that are in the third half
                                         hour of the block
00262                                    For counter1 As Integer = 1 To NStudAv
00263                                        GetStudAV(counter1)
00264                                        If StudAv.StudNo = student.StudNO And StudAv.Appointment
                                             = counter And StudAv.DayNO = parts(1) Then
00265                                            Exit For
00266                                        End If
00267                                    Next
00268                                    'sets there availability to false if block number is 0 after
                                         decreasing it by 4
00269                                    StudAv.Block -= 4
00270                                    If StudAv.Block = 0 Then
00271                                        StudAv.available = False
00272                                    End If
00273                                    PutStudAv(StudAv, StudAv.studAVNO)
00274                                Next
00275                            End If
00276                        Else
00277                            'staff
00278                            If Appointmentlength = 5 Then
00279                                'appointment length 5
00280                                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)
00281                                    'finds relevant stud av records that are in the first half
                                         hour of the block
00282                                    For counter1 As Integer = 1 To Nstaff
00283                                        GetStaffAV(counter1)
00284                                        If StaffAv.StaffNO = Staff.StaffNO And StaffAv.
                                             Appointment = counter And StaffAv.DayNO = parts(1) Then
00285                                            Exit For
00286                                        End If
```

```
00287                    Next
00288                    'sets there availability to false if block number is 0 after decreasing it by 1
00289                    StaffAv.Block -= 1
00290                    If StaffAv.Block = 0 Then
00291                        StaffAv.Available = False
00292                    End If
00293                    PutStaffAv(StaffAv, StaffAv.staffAVNO)
00294                Next
00295                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)
00296                    'finds relevant stud av records that are in the second half hour of the block
00297                    For counter1 As Integer = 1 To NStaffAv
00298                        GetStaffAV(counter1)
00299                        If StaffAv.StaffNO = Staff.StaffNO And StaffAv. Appointment = counter And StaffAv.DayNO = parts(1) Then
00300                            Exit For
00301                        End If
00302                    Next
00303                    'sets there availability to false if block number is 0 after decreasing it by 2
00304                    StaffAv.Block -= 2
00305                    If StaffAv.Block = 0 Then
00306                        StaffAv.Available = False
00307                    End If
00308                    PutStaffAv(StaffAv, StaffAv.staffAVNO)
00309                Next
00310            Else
00311                'appointment length 10
00312                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) + 4) Step 2
00313                    'finds relevant stud av records that are in the first half hour of the block
00314                    For counter1 As Integer = 1 To NStaffAv
00315                        GetStaffAV(counter1)
00316                        If StaffAv.StaffNO = Staff.StaffNO And StaffAv. Appointment = counter And StaffAv.DayNO = parts(1) Then
00317                            Exit For
00318                        End If
00319                    Next
00320                    'sets there availability to false if block number is 0 after decreasing it by 1
00321                    StaffAv.Block -= 1
00322                    If StaffAv.Block = 0 Then
00323                        StaffAv.Available = False
00324                    End If
00325                    PutStaffAv(StaffAv, StaffAv.staffAVNO)
00326                Next
00327                For counter As Integer = (parts(2) \ 100) + (((parts(2)) - ((( parts(2)) \ 100) * 100)) / 5) + 6 To ((parts(2) \ 100) + ((( parts(2)) - (((parts(2)) \ 100) * 100)) / 5) + 10) Step 2
00328                    'finds relevant stud av records that are in the second half hour of the block
00329                    For counter1 As Integer = 1 To NStaffAv
00330                        GetStaffAV(counter1)
00331                        If StaffAv.StaffNO = Staff.StaffNO And StaffAv. Appointment = counter And StaffAv.DayNO = parts(1) Then
00332                            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
            System.EventArgs) Handles rad5min.CheckedChanged
00006          'checks if the the 5 min radio button is checked and if it is sets the
                appointmetn length to 5 mins
00007          'otherwise it sets the appointment length to 10 mins
00008          If rad5min.Checked = True Then
00009              Appointmentlength = 5
00010          Else
00011              Appointmentlength = 10
00012          End If
00013
00014          'clears the secondary comboboxes
00015          cmbStart.Items.Clear()
00016          cmbEnd.Items.Clear()
00017          cmbDay.Items.Clear()
00018          'resets the day settings
00019          For counter As Integer = 1 To NDay
00020              Day.DayNO = counter
00021              Day.finish = 288
00022              Day.Start = 0
00023              Putday(Day, Day.DayNO)
00024          Next
00025
00026          'calls the subroutine second half which decided weather to complete the
                other repacutions of changing the appointment length
00027          Call secondhalf()
00028      End Sub
00029
00030      Private Sub frmDaySettings_Load(ByVal sender As System.Object, ByVal e As System
            .EventArgs) Handles MyBase.Load
00031          'populates the dropdown combobox with numbers 1 to 255
00032          For counter As Integer = 1 To 255
00033              cmbNdays.Items.Add(counter)
00034          Next
00035      End Sub
00036
00037      Private Sub cmbNdays_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
            As System.EventArgs) Handles cmbNdays.SelectedIndexChanged
00038          'records the number of dayz
00039          NDay = cmbNdays.SelectedItem
00040
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
                other repacutions of changing the number of days
00053          Call secondhalf()
00054
00055
```

```
00056       End Sub
00057
00058       Public Sub secondhalf()
00059           'sub to check if the appointment length and day numbers have been chosen. if so it populates the
00060           'day.dat file
00061           If NDay <> -1 And Appointmentlength <> -1 Then
00062               For counter As Integer = 1 To NDay
00063                   Day.DayNO = counter
00064                   Day.Start = 0
00065                   Day.finish = 288
00066                   Putday(Day, counter)
00067               Next
00068           Else
00069               Exit Sub
00070           End If
00071
00072           cmbDay.Visible = True
00073           cmbStart.Visible = True
00074           cmbEnd.Visible = True
00075
00076           'populates the day selector combo box with each of the days
00077           cmbDay.Items.Clear()
00078           For counter = 1 To NDay
00079               cmbDay.Items.Add(counter)
00080           Next
00081       End Sub
00082
00083       Private Sub rad10min_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles rad10min.CheckedChanged
00084           'records the new value of appintment length and checks if its ready to start hte second half
00085           Appointmentlength = 10
00086
00087           'clears the secondary comboboxes
00088           cmbStart.Items.Clear()
00089           cmbEnd.Items.Clear()
00090           cmbDay.Items.Clear()
00091           'resets the day settings
00092           For counter As Integer = 1 To NDay
00093               Day.DayNO = counter
00094               Day.finish = 288
00095               Day.Start = 0
00096               Putday(Day, Day.DayNO)
00097           Next
00098           'calls the subroutine second half which decided weather to complete the other repacutions of changing the appointment length
00099           Call secondhalf()
00100       End Sub
00101
00102       Private Sub cmbDay_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmbDay.SelectedIndexChanged
00103           'calls the subrootine that will populate cmbstart and cmbend
00104           Call populateStartEndDaySettings()
00105       End Sub
00106
00107       Private Sub cmbStart_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmbStart.SelectedIndexChanged
00108           Dim hours As Integer = (cmbStart.SelectedItem) \ 100
00109           Dim minuets As Integer = ((cmbStart.SelectedItem) - (((cmbStart.SelectedItem) \ 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          change1 = 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.
          EventArgs) Handles btnBack.Click
00138          'closes the form
00139          Me.Close()
00140      End Sub
00141
00142      Private Sub cmbEnd_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
          As System.EventArgs) Handles cmbEnd.SelectedIndexChanged
00143          Dim hours As Integer = (cmbEnd.SelectedItem) \ 100
00144          Dim minuets As Integer = ((cmbEnd.SelectedItem) - (((cmbEnd.SelectedItem) \
           100) * 100)) / 5
00145          'eliminates accidental loops
00146          If change1 = True Then
00147              change1 = False
00148              Exit Sub
00149          End If
00150          'sets the end time for the day selected
00151          Day.finish = (hours * 12) + minuets
00152          Day.DayNO = cmbDay.SelectedItem
00153          Putday(Day, Day.DayNO)
00154          'clears cmbend and cmbstart
00155          cmbEnd.Items.Clear()
00156          cmbStart.Items.Clear()
00157          'calls the subroutine to populate cmbstart and cmbend
00158          Call populateStartEndDaySettings()
00159          change = True
00160          'selscts the start time in cmbstart
00161          cmbStart.SelectedIndex = Day.Start / 6
00162          change1 = True
00163          'selects the end time in cmbend
00164          If Appointmentlength = 5 Then
00165              cmbEnd.SelectedIndex = (Day.finish - Day.Start) \ 6 - 2
00166          Else
00167              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 be accessed
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 admins and it checks if the user is in the format of a staff member
00079        'if so then it sends the user to the admin file to set up the system
00080
00081        If FileLen("staff.dat" ) = 0 And IsNumeric(user) = False And Len(user) = 3 Then
00082            frmAdmin.Show()
00083            Me.Close()
00084        End If
00085
00086        'if the usesr has been given a usertype of 0 then he is not in the system so access to the other forms is blocked of by making the buttons
00087        'invisible and it then sends an error message sayting they arent in the system and advising them to check with an admisistrator if they
00088        'feel it is wrong
00089        If usertype = 0 Then
00090            btnAvailability.Visible = False
00091            MsgBox("Your username is not recognised by the system. If this is an error please contact the it technicians." , , "ERROR" )
00092
00093        End If
00094
00095
00096    End Sub
00097
00098    Private Sub btnAvailability_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAvailability.Click
00099        'though the button should only be vissible if hte user is in the system if first checks if they are tne sends them to the availablility form
00100        If usertype = 1 Or usertype = 2 Then
00101            frmAvailability.Show()
00102        End If
00103    End Sub
00104
00105    Private Sub btnExit_Click(sender As System.Object, e As System.EventArgs) Handles btnExit.Click
00106        'closes the form
00107        Me.Close()
00108    End Sub
00109
00110    Private Sub btnadmin_Click(sender As System.Object, e As System.EventArgs) Handles btnadmin.Click
00111        '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 availabity 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 stucture
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
```

```
            student data is stored
00123       Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
             "students.csv" )
00124       TextFileReader.TextFieldType = FileIO.FieldType.Delimited
00125       TextFileReader.SetDelimiters("," )
00126
00127       Dim CurrentRow As String()
00128       Dim OnRec As Integer = 0
00129       Dim FileNum As Integer = FreeFile()
00130       'opens file
00131       FileOpen(FileNum, "Student.dat" , OpenMode.Random, OpenAccess.Default,
             OpenShare.Default, Len(student))
00132
00133       While Not TextFileReader.EndOfData
00134           Try
00135               CurrentRow = TextFileReader.ReadFields()
00136               If Not CurrentRow Is Nothing Then
00137                   OnRec = OnRec + 1
00138                   'puts data into the studet structure
00139                   With student
00140                       Try
00141
00142                           .StudNO = CurrentRow(0)
00143                           .StudID = CurrentRow(1)
00144                           .Surname = CurrentRow(2)
00145                           .Forename = CurrentRow(3)
00146                           .Year = CurrentRow(4)
00147                       Catch
00148                           'this is trigered if there was a problem with inputing
                             to the student structure
00149                           'it stops the importing and telles the suer what has
                             happened
00150                           stopimport = True
00151                           MsgBox("error with student.csv" )
00152                           Exit Sub
00153                       End Try
00154
00155                   End With
00156                   'puts data from the student structure into the student dat file
00157                   FilePut(FileNum, student, OnRec)
00158               End If
00159           Catch ex As _
00160                   Microsoft.VisualBasic.FileIO.MalformedLineException
00161               'if error then error message is sent and try ends
00162               MsgBox("Line "  & ex.Message & "is not valid and will be skipped." )
00163           End Try
00164       End While
00165       'message box is sent saying that the students are imported and how many
00166       Nstudents = OnRec
00167       MsgBox(NStudents & " students imported" )
00168       'file is closed
00169       FileClose(FileNum)
00170       TextFileReader.Dispose()
00171   End Sub
00172
00173   'imports staff into thier dat file
00174   Public Sub ImportStaff()
00175       'opens microsoft file reader and sets the file to be read as tutor.csv
00176       Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
             "tutor.csv" )
00177       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 trigered if there was a problem with inputing
                                  to the staff structure
00207                                'it stops the importing and telles the suer 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,
               OpenShare.Default, Len(Lesson))
00241     While Not TextFileReader.EndOfData
00242         Try
00243             CurrentRow = TextFileReader.ReadFields()
00244             If Not CurrentRow Is Nothing Then
00245                 OnRec = OnRec + 1
00246                 'puts data into file structure staff
00247                 With Lesson
00248                     Try
00249                         .LessonNO = OnRec
00250                         .StaffNO = CurrentRow(1)
00251                         .StudNO = CurrentRow(0)
00252                     Catch
00253                         'this is trigered if there was a problem with inputing
                           to the lesson structure
00254                         'it stops the importing and telles the suer what has
                           happened
00255                         stopimport = True
00256                         MsgBox("error with studentclass.csv" )
00257                         Exit Sub
00258                     End Try
00259                 End With
00260                 'puts data in the lesson file structure into the lesson dat file
00261                 FilePut(FileNum, Lesson, OnRec)
00262             End If
00263         Catch ex As _
00264             Microsoft.VisualBasic.FileIO.MalformedLineException
00265             'error in text sends error message and ends try
00266             MsgBox("Line "  & ex.Message & "is not valid and will be skipped." )
00267         End Try
00268     End While
00269     Nlesson = OnRec
00270     'sends message box notifying admin that student part of lessons have been
           imported and how many have been imported
00271     MsgBox("Student half imported" )
00272     FileClose(FileNum)
00273     TextFileReader.Dispose()
00274 End Sub
00275
00276 Public Sub importLessonStaff()
00277     'opens microsoft file reader and sets the file to be read as classslots.csv
00278     Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
           "classSlots.csv" )
00279     TextFileReader.TextFieldType = FileIO.FieldType.Delimited
00280     TextFileReader.SetDelimiters("," )
00281
00282     Dim lastlesson As Integer = -1
00283     Dim currentrow As String() = Nothing
00284     Dim onrec As Integer = 0
00285
00286     OnRec = 0
00287
00288     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 trigered if there was a problem with inputing
                              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(Filenum)
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 Filenum As Integer = FreeFile()
00351         GetStudAV = Nothing
00352         'opens the studav dat file
00353         FileOpen(Filenum, "StudAV.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(StudAv))
00354         'gets data
00355         FileGet(Filenum, GetStudAV, RecNo)
00356         'closes file
00357         FileClose(Filenum)
00358     End Function
00359 
00360     'retrives a staff availiability record
00361     Public Function GetStaffAV(ByVal RecNo As Integer) As StaffAvRec
00362         'function for getting data from staffav dat file
00363         Dim Filenum As Integer = FreeFile()
00364         GetStaffAV = Nothing
00365         'opens the staffav dat file
00366         FileOpen(Filenum, "StaffAV.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(StaffAv))
00367         'gets data
00368         FileGet(Filenum, GetStaffAV, RecNo)
00369         'closes file
00370         FileClose(Filenum)
00371     End Function
00372 
00373     'retrives a day record
00374     Public Function GetDay(ByVal RecNo As Integer) As DayRec
00375         'function for getting data from day dat file
00376         Dim Filenum As Integer = FreeFile()
00377         GetDay = Nothing
00378         'opens the day dat file
00379         FileOpen(Filenum, "Day.dat" , OpenMode.Random, OpenAccess.Default, OpenShare
              .Default, Len(Day))
00380         'gets data
00381         FileGet(Filenum, GetDay, RecNo)
00382         'closes file
00383         FileClose(Filenum)
00384     End Function
00385 
00386     'retrives a lesson record
00387     Public Function Getlesson(ByVal RecNo As Integer) As LessonRec
00388         'function for getting data from lesson dat file
00389         Dim Filenum As Integer = FreeFile()
00390         Getlesson = Nothing
00391         'opens the lesson dat file
00392         FileOpen(Filenum, "Lesson.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(Lesson))
00393         'gets data
00394         FileGet(Filenum, Getlesson, RecNo)
00395         'closes file
00396         FileClose(Filenum)
00397     End Function
00398 
00399     'retreives 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,
              OpenShare.Default, Len(Appointment))
00406        'gets data
00407        FileGet(Filenum, Getappointmentrec, RecNo)
00408        'closes file
00409        FileClose(Filenum)
00410    End Function
00411
00412    'overwrites an student onto the student dat file
00413    Public Sub PutStudent(ByVal EditedStudent As StudRec, ByVal RecNo As Integer)
00414        'sub for putting data into the student dat file
00415        Dim Filenum As Integer = FreeFile()
00416        'opens student dat file
00417        FileOpen(Filenum, "Student.dat" , OpenMode.Random, OpenAccess.Default,
              OpenShare.Default, Len(student))
00418        'puts data into student dat file
00419        FilePut(Filenum, EditedStudent, RecNo)
00420        'closes student dat file
00421        FileClose(Filenum)
00422    End Sub
00423
00424    'overwrites an staff onto the staff dat file
00425    Public Sub PutStaff(ByVal EditedStaff As StaffRec, ByVal RecNo As Integer)
00426        'sub for putting data into the staff dat file
00427        Dim Filenum As Integer = FreeFile()
00428        'opens staff dat file
00429        FileOpen(Filenum, "Staff.dat" , OpenMode.Random, OpenAccess.Default,
              OpenShare.Default, Len(Staff))
00430        'puts data into staff dat file
00431        FilePut(Filenum, EditedStaff, RecNo)
00432        'closes staff dat file
00433        FileClose(Filenum)
00434    End Sub
00435
00436    'overwrites an studavrec onto the studav dat file
00437    Public Sub PutStudAv(ByVal EditedStudAv As StudAvRec, ByVal RecNo As Integer)
00438        'sub for putting data into the studAv dat file
00439        Dim Filenum As Integer = FreeFile()
00440        'opens studAv dat file
00441        FileOpen(Filenum, "StudAv.dat" , OpenMode.Random, OpenAccess.Default,
              OpenShare.Default, Len(StudAv))
00442        'puts data into studAv dat file
00443        FilePut(Filenum, EditedStudAv, RecNo)
00444        'closes studAv dat file
00445        FileClose(Filenum)
00446    End Sub
00447
00448    'overwrites an staffAv onto the staffAv dat file
00449    Public Sub PutStaffAv(ByVal EditedStaffAv As StaffAvRec, ByVal RecNo As Integer)
00450        'sub for putting data into the staffAv dat file
00451        Dim Filenum As Integer = FreeFile()
00452        'opens staffAv dat file
00453        FileOpen(Filenum, "StaffAv.dat" , OpenMode.Random, OpenAccess.Default,
              OpenShare.Default, Len(StaffAv))
00454        'puts data into staffAv dat file
00455        FilePut(Filenum, EditedStaffAv, RecNo)
```

```
00456         'closes staffAv dat file
00457         FileClose(Filenum)
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 Filenum As Integer = FreeFile()
00464         'opens day dat file
00465         FileOpen(Filenum, "day.dat" , OpenMode.Random, OpenAccess.Default, OpenShare
               .Default, Len(Day))
00466         'puts data into day dat file
00467         FilePut(Filenum, Editedday, RecNo)
00468         'closes day dat file
00469         FileClose(Filenum)
00470     End Sub
00471
00472     '------------------------------------------------------------------------
          'overwrites an lesson onto the lesson dat file
00473     Public Sub Putlesson(ByVal Editedlesson As LessonRec, ByVal RecNo As Integer)
00474         'sub for putting data into the lesson dat file
00475         Dim Filenum As Integer = FreeFile()
00476         'opens lesson dat file
00477         FileOpen(Filenum, "Lesson.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(Lesson))
00478         'puts data into lesson dat file
00479         FilePut(Filenum, Editedlesson, RecNo)
00480         'closes lesson dat file
00481         FileClose(Filenum)
00482     End Sub
00483
00484     '------------------------------------------------------------------------
          'overwrites an appointment onto the appointment dat file
00485     Public Sub Putappointment(ByVal Editedappointment As AppointmentsRec, ByVal
           RecNo As Integer)
00486         'sub for putting data into the appointment dat file
00487         Dim Filenum As Integer = FreeFile()
00488         'opens appointment dat file
00489         FileOpen(Filenum, "appointments.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(Appointment))
00490         'puts data into appointment dat file
00491         FilePut(Filenum, Editedappointment, RecNo)
00492         'closes the appointment dat file
00493         FileClose(Filenum)
00494     End Sub
00495
00496     '------------------------------------------------------------------------
          Public Sub populateStartEndDaySettings()
00497         'populates cmbstart with the times at the required appiontment length apart
00498         Day = GetDay(frmDaySettings.cmbDay.SelectedItem)
00499         If Appointmentlength = 5 Then
00500             'populates cmbstart with 5 min appointments slots
00501             For counter As Integer = 0 To (Day.finish - 12) Step 6
00502                 'each slot contained between the beginin and the finsish time is
                       converted into 24hour
00503                 'military style time
00504                 frmDaySettings.cmbStart.Items.Add(militarytime(counter))
00505             Next
00506
00507             'populates the cmbEnd list witht the available times
00508             For counter As Integer = (Day.Start + 12) To 288 Step 6
00509                 'each slot contained between the beginin and the finsish time is
                       converted into 24hour
00510                 '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
                       converted into 24hour
00517                 'military style time
00518                 frmDaySettings.cmbStart.Items.Add(militarytime(counter))
00519             Next
00520
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
                       converted into 24hour
00524                 'military style time
00525                 frmDaySettings.cmbEnd.Items.Add(militarytime(counter))
00526             Next
00527
00528         End If
00529
00530     End Sub
00531     'fucntion that changes a number from 0 to 287 into its coresponidng 24 hour
          clock time
00532     Public Function militarytime(ByVal timeNO As Integer) As String
00533         Dim hours As String
00534         Dim minuets As String
00535
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 TheSortingAlogorithm()
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 avaliable 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
                                              the other is 20 then they are
00589                                            '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
00595                                            'not in the same block and it needs
                                              to be set as unavailable
00596                                            ElseIf StudAv.Block = 20 And studav2.
                                              Block = 21 Then
00597                                                studav2.Block = 0
00598                                                studav2.available = False
00599                                                PutStudAv(studav2,
                                                  studav2.studAVNO)
00600                                            End If
00601                                        Next
00602
00603                                    Else
00604                                        'handles 10 min cases
00605                                        For counter5 As Integer = 0 To NStudAv
00606                                            studav2 = GetStudAV(counter5)
00607                                            'checks if teh block value of the
                                              current spot is 20 because if it is
                                              and
```

```
                    the other is 22 or 23 then they are
00608               'not in the same block and it needs to
                    be set as unavailable
00609               If StudAv.Block = 20 And (studav2.Block
                    = 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
                        current spot is 21 because if it
                        is
                        and the other is 23 then they are
00614                   'not in the same block and it needs
                        to be set as unavailable
00615               ElseIf StudAv.Block = 21 And studav2.
                    Block = 23 Then
00616                   studav2.Block = 0
00617                   studav2.available = False
00618                   PutStudAv(studav2,
                        studav2.studAVNO)
00619                   'checks if teh block value of the
                        current spot is 23 because if it
                        is
                        and the other is 20 or 21 then
                        they
                        are
00620                   'not in the same block and it needs
                        to be set as unavailable
00621               ElseIf StudAv.Block = 23 And (studav2.
                    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
                        current spot is 22 because if it
                        is
                        and the other is 20 then they are
00626                   'not in the same block and it needs
                        to be set as unavailable
00627               ElseIf StudAv.Block = 22 And studav2.
                    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 availablitity
00635       StaffAv.Available = False
00636       PutStaffAv(StaffAv, StaffAv.staffAVNO)
00637
00638       'sets the stud blocks and availablitity for
            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
```

```
00644                    'works out the first appointment of the half hour
00645                    lowerbound = (studav2.Appointment \ 6) * 6
00646                    'checks if the studav is not already been set in the earlier check and if it is for the right student
00647                    If studav2.StudNo = student.StudNO And StudAv.Block <> 21 Or 20 Then
00648                        Select Case studav2.Appointment
00649                            'too early
00650                        Case Is <= (lowerbound - 7)
00651                            studav2.available = False
00652                            studav2.Block = 0
00653                            'too late
00654                        Case Is >= (lowerbound + 12)
00655                            studav2.available = False
00656                            studav2.Block = 0
00657                            'half hour early
00658                        Case (lowerbound - 6) To (lowerbound - 1)
00659                            studav2.Block = 20
00660                            'half hour later
00661                        Case (lowerbound + 6) To (lowerbound + 11)
00662                            studav2.Block = 21
00663                            'appointmetn before
00664                        Case (StudAv.Appointment - 1)
00665                            studav2.available = False
00666                            studav2.Block = 0
00667                            'appointment after
00668                        Case (StudAv.Appointment + 1)
00669                            studav2.available = False
00670                            studav2.Block = 0
00671                            'the appointmetn in question
00672                        Case StudAv.Appointment
00673                            studav2.Block = 0
00674                            studav2.available = False
00675                        End Select
00676
00677
00678                    End If
00679                Else
00680                    'works out the first appointment of the half hour
00681                    lowerbound = (studav2.Appointment \ 6) * 6
00682                    'checks if the studav is not already been set in the earlier check and if it is for the right student
00683                    If studav2.StudNo = student.StudNO And StudAv.Block <> 21 Or 20 Then
00684                        Select Case studav2.Appointment
00685                            'too early
00686                        Case Is <= (lowerbound - 13)
```

```
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                                                'appointmetn 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 appointmetn 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 = "l09m3e?!"
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
                  & vbNewLine
00749            'finds each appointment of the student
00750            For counter2 As Integer = 1 To NAppointment
00751                If Appointment.studNO = student.StudNO Then
00752                    'gets the name of the member of staff the appointment is with
00753                    Staff = GetStaff(Appointment.StaffNO)
00754                    'puts in the details of the appiontment into the email
00755                    body = body & Staff.Forename & " "  & Staff.Surname & ""  &
                          militarytime(Appointment.start) & " day "  & Appointment.day &
                          vbNewLine
00756                End If
00757            Next
00758            'puts the sign off of the email into the text
00759            body = body & vbNewLine & "thank you very much"  & vbNewLine & "simon
                  bellows"  & vbNewLine & vbNewLine & "deputy head"
00760            'calls the routine to send the email
00761            Call SendEmails2("sim.bellows@gmail.com" , subject, body, username,
                  password, recipient)
00762        Next
00763        'loop that cycles through each staff member so each gets an email
00764        For counter1 As Integer = 1 To Nstaff
00765            'loads current staff members details
00766            Staff = GetStaff(counter1)
00767            'genereates the staff school email
00768            recipient = Staff.staffID + "@WMSF.ac.uk"
00769            'puts the intitial greeting for the email and clears the old message
00770            body = "Dear "  & Staff.Forename & " "  & Staff.Surname & vbNewLine &
                  vbNewLine
00771            'finds each appoointment of the staff member
00772            For counter2 As Integer = 1 To NAppointment
00773                If Appointment.StaffNO = Staff.StaffNO Then
00774                    'gets the name of the student the appointmetn is with
00775                    student = GetStudent(Appointment.studNO)
00776                    'puts in the details of the appointment into the email
00777                    body = body & student.Forename & " "  & student.Surname & ""  &
                          militarytime(Appointment.start) & " day "  & Appointment.day &
                          vbNewLine
00778                End If
00779            Next
00780            'puts the sign off of the email into the message
00781            body = body & vbNewLine & "thank you very much"  & vbNewLine & "simon
                  bellows"  & vbNewLine & vbNewLine & "deputy head"
00782            'calls the routine to send the email
00783            Call SendEmails2("sim.bellows@gmail.com" , subject, body, username,
                  password, recipient)
00784        Next
00785
00786    end sub
00787    Public Sub SendEmails2(ByVal FromAddress As String, _
00788                           ByVal Subject As String, _
00789                              ByVal Body As String, _
00790                                ByVal UserName As String, _
00791                                ByVal Password As String, _
00792                                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 recipent 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,
                  Password)
00813             SMTPServer.EnableSsl = True
00814             'sends it
00815             SMTPServer.Send(Email)
00816             'clears it
00817             Email.Dispose()
00818             'notificaltion if smtp failed
00819         Catch ex As SmtpException
00820             Email.Dispose()
00821             MsgBox("Sending Email Failed. Smtp Error." )
00822             'notification if portnuimber owas wrong
00823         Catch ex As ArgumentOutOfRangeException
00824             Email.Dispose()
00825             MsgBox("Sending Email Failed. Check Port Number." )
00826             'notification if portnunber is wrong
00827         Catch Ex As InvalidOperationException
00828             Email.Dispose()
00829             MsgBox("Sending Email Failed. Check Port Number." )
00830         End Try
00831     End Sub
00832 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"
       CurrentProfile="(Default)" UseMySettingsClassName="true">
00003   <Profiles>
00004     <Profile Name="(Default)" />
00005   </Profiles>
00006   <Settings />
00007 </SettingsFile>
00008
```