

UNIVERSITY OF  
BIRMINGHAM

**University of Birmingham Research Archive**

**e-theses repository**

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

## **Appendix to**

**Context Awareness for Wearable Computers**

**By**

**Huw William Bristow**

A thesis submitted to  
The University of Birmingham  
for the degree of  
DOCTOR OF PHILOSOPHY  
School of Electronic, Electrical &  
Computer Engineering  
The University of Birmingham

## Appendix A Context Software Code

### A.1 Form SSW

The screenshot shows a Windows-style application window titled "SSW PC". The interface is divided into several sections:

- GPS Coordinates Data:** Latitude (5227.2000), Longitude (00156.9210), and a URL (http://www.eee.bham.ac.uk).
- Time/Date/Day:** Time (18:30:04), Date (Monday 17 February 2003), and a WeekDay dropdown.
- Building:** A text field containing "The School of Electronic and Electrical Engineering".
- Acc Data:** X Axis Data (0.562) and Y Axis Data (1.125). Below this is a stick figure icon.
- Object:** An empty text field for "Object Name".
- Global Movement:** Movement status (Moving), Speed (020.0 Knots), and Direction (000.0 degrees).
- Possible Content:** A list box containing files like BED.jpg, BEDSIT.htm, BEDSITHOM.htm, BEDSTD.htm, BEDSTDHOM.ht, B0.jpg, B02.jpg, B02SITEEE.pdf, and B02STDEEE.htm.
- Comm 1 GPS:** A text field with the string "\$GPRMC,183004,A,5227.2000,N,00156.9210,W,020.0,000." and a "Female Voice" label.
- Comm 2 Acc:** A text field with a string of "X" characters.
- Comm 3 Obj:** An empty text field.

Figure 1 - SSW Form

Dim check As Boolean 'used on the menu to show when the raw data is being shown

```

Dim testmodecheck As Boolean 'check mode allows the user to choose locations
Public GPSBuildingOld As String 'the building that the user was last at

Public UserPosition As String 'the users body position
Public UserPositionOld As String 'the users old body position
Public Sound As String 'the name of the sound to be played
Public Pic As String 'the picture of the body position to be displayed
Public XDataAverage As Double 'the average X value of acceleration
Public YDataAverage As Double 'the average Y value of acceleration

Public GPSLongitudeStart As Integer 'the position in the buffer where the longitude starts
Public GPSLatitudeStart As Integer 'the position in the buffer where the latitude starts
Public GPSStartCode As Integer 'position of "$GPRMC" in buffer
Public GPSTime As Integer 'the time from the GPS
Public GPSURLOld As String 'the last URL sent the the web browser
Public GPSURLDatabaseEnd As Integer '
Public GPSURLDatabaseLength As Integer '
Public GPSURLNewDatabase As String

Public ObjectID As String 'binary ID of the object
Public ObjectBuffer As String 'input from object
Public ObjectIDOld As String 'last object

Public Path As String 'the path of the appication of folder where files are stored
Public redirect As String 'the URL the web browser is redirected to

Public accCount As Integer 'which body position in test mode
Public gpsCount As Integer 'which location in test mode
Public objectCount As Integer 'which object in test mode
Public testmode As Boolean 'testmode on or off

Public number As Integer
Public CamImageNum As Integer

Public Voice As SpeechLib.SpVoice 'delcares the voice

'Note limitation: user must be on line at all times, else timming errors occur

Public Sub Form_Load()

'Path = "c:\program files\LBpTO\"
'Path = App.Path
Path = "c:\program files\moreconimg\"

```

```

Call CommSet
MSComm2.CommPort = GPSComm           'Defines Comm port as port 2 for GPS
MSComm2.Settings = "4800,N,8,1"     'Sets comm port settings
On Error GoTo CommError
MSComm2.PortOpen = True              'Opens comm port

MSComm1.CommPort = ACCComm           'Defines Comm port as port 4 for ACC
MSComm1.Settings = "4800,N,8,1"     'Sets comm port settings
On Error GoTo CommError
MSComm1.PortOpen = True              'Opens comm port

MSComm3.CommPort = OBJComm           'Defines Comm port as port 4 for ACC
MSComm3.Settings = "2400,N,8,1"     'Sets comm port settings
On Error GoTo CommError
MSComm3.PortOpen = True              'Opens comm port

GPSBuffer.Text = "$GPRMC,183004,A,5227.2000,N,00156.9210,W,000.0,000.0,141100,004.1,W*68" 'elec eng

If (MSComm1.PortOpen = True) Then

    TmrAcc.Enabled = True
    TmrAcc.Interval = 2005

    TmrHistory.Enabled = True
    TmrHistory.Interval = 10000

    LocationNumber = 0 'Sets the location history to start at the begining of the array

    If number <> 1 Then
        GPSURLNew.Text = Path & "map.htm"

        'mainIexplorer = Shell("C:\Program Files\Internet Explorer\IEXPLORE.EXE", 3)
        frminternet.Show

        GPSBuildingOld = "Hello"
        UserPositionOld = "Hello"
        ObjectIDOld = "Hello"

        CamImageNum = 1 'sets the images to 1
        'frmInternet.actiVideoX1.ConnectVideo 'connects to webcam

    accCount = 0
    gpsCount = 0

```

```

objectCount = 5
testmode = False

PlayFirstTime = 1 'variible used in making the appointment notify only play the first time it is called

Set Voice = New SpeechLib.SpVoice 'set a voice
Voice.Rate = "0" 'sets the speech rate
Set Voice.Voice = Voice.GetVoices("gender=female").Item(0) 'sets a female voice

    Voice.AlertBoundary = SVEPhoneme 'Let words be interrupted
End If
Exit Sub

CommError:
MsgBox ("You have selected an invalid comm port please try again")
SSW.TmrAcc.Enabled = False
frmcomm.Show
End Sub

Private Sub Form_Unload(Cancel As Integer)
Unload frmcomm
Unload frmspeech
Unload frminternet
End
End Sub

Private Sub muabout_Click()

frmabout.Show

End Sub

Private Sub mucomm_Click()
frmcomm.Show
End Sub

Private Sub muhelp_Click()

HelpIEexplorer = Shell("C:\Program Files\Internet Explorer\IEXPLORE.EXE", 3) 'Activates Internet Explorer
SendKeys "^o" & "C:\Documents and Settings\huw\My Documents\My Webs\WECA PC Help\WECA PChelp.htm" & "{Enter}"

```

```
End Sub
```

```
Private Sub muEmail_Click()  
    frmEMail.MAPISession1.SignOn  
  
    frmEMail.MAPIMessages1.SessionID = frmEMail.MAPISession1.SessionID  
  
    frmEMail.MAPIMessages1.Compose  
    frmEMail.MAPIMessages1.RecipDisplayName = "h.w.bristow@eee-fs7.bham.ac.uk"  
    frmEMail.MAPIMessages1.MsgSubject = "Huw is on the move..."  
    frmEMail.MAPIMessages1.MsgNoteText = "This is a test"  
  
    frmEMail.MAPIMessages1.Send  
    frmEMail.MAPISession1.SignOff  
End Sub
```

```
Private Sub muHide_Click()  
    Me.Hide  
End Sub
```

```
End Sub
```

```
Private Sub murawdata_Click()  
'allows a user to see the raw data that is coming in from the comm ports  
'very useful when debugging the system
```

```
    If check = False Then  
        GPSBuffer.Visible = True  
        GPSCaption.Visible = True  
        AccBuffer.Visible = True  
        AccCaption.Visible = True  
        ObjBuffer.Visible = True  
        lbObjCap.Visible = True  
        murawdata.Checked = True  
        lbGPSComm.Visible = True  
        lbAccComm.Visible = True  
        lbObjComm.Visible = True  
        SSW.Height = 6885  
  
        check = True  
    ElseIf check = True Then  
        GPSBuffer.Visible = False  
        GPSCaption.Visible = False  
        AccBuffer.Visible = False
```

```
    AccCaption.Visible = False
    ObjBuffer.Visible = False
    lbObjCap.Visible = False
    murawdata.Checked = False
    lbGPSComm.Visible = False
    lbAccComm.Visible = False
    lbObjComm.Visible = False
    SSW.Height = 5850

    check = False
End If

End Sub

Private Sub muspeech_Click()
    frmSpeech.Show
End Sub

Private Sub muTestMode_Click()
    'turns test mode on and off

    If testmodecheck = True Then
        testmode = True
        testmodecheck = False
        muTestMode.Checked = True
    ElseIf testmodecheck = False Then
        testmode = False
        testmodecheck = True
        muTestMode.Checked = False
    End If

End Sub

End Sub

Private Sub muWebCamConnect_Click()
    frmInternet.actiVideoX1.ConnectVideo 'connect to camera
End Sub

Private Sub muWebCamDisconnect_Click()
    frmInternet.actiVideoX1.DisconnectVideo 'Disconnects video
End Sub

Private Sub muWebCamShow_Click()
```







```

ProgressMonitor2.Visible = True
ProgressMonitor2.Caption = "No GPS Signal"
On Error GoTo HistoryError
GPSLatitudeNew.Text = LatitudeHistory(LocationNumber - 1)
GPSLongitudeNew.Text = LongitudeHistory(LocationNumber - 1)
Call DayTime
End If
GoTo ObjectInput

```

```

HistoryError: 'when first turned on there is no previous location therefore if the
'the gps signal is not present use EEE as the default location
GPSLatitudeNew.Text = 5227.2
GPSLongitudeNew.Text = 156.921
Call DayTime

```

```

ObjectInput:
'If objectCount = 0 Then
ObjectBuffer = MSComm3.Input 'get input from object
'ElseIf objectCount = 1 Then
'ObjectBuffer = "OBJECT,t" 'coffee cup
'ElseIf objectCount = 2 Then
'ObjectBuffer = "OBJECT,v" 'Desktop PC
'ElseIf objectCount = 3 Then
'ObjectBuffer = "OBJECT,x" 'Book
'ElseIf objectCount = 4 Then
'ObjectBuffer = "OBJECT,y" 'Diary
'ElseIf objectCount = 5 Then
'ObjectBuffer = "OBJECT,z" 'TV
'ElseIf objectCount = 6 Then
'ObjectBuffer = "OBJECT,u" 'Book2
'ElseIf objectCount = 7 Then
'ObjectBuffer = "OBJECT,~" 'Mode of transport
'ObjectBuffer = "OBJECT,3" 'Wallett
'ObjectBuffer = "OBJECT,2" 'Food
'ObjectBuffer = "OBJECT,7" 'Toilet
'ElseIf objectCount = 8 Then
'ObjectBuffer = "OBJECT,p" 'transmitting but not known objectNothing
'End If
'ObjectBuffer = "OBJECT,K" 'signal not being transmitted
'ObjectBuffer = ""

If ObjectBuffer <> "" Then 'if object is present
Call objectcalc
progressmonitor4.Visible = False
ElseIf ObjectBuffer = "" Then

```

```

        progressmonitor4.Visible = True
        progressmonitor4.Caption = "No Object Signal"
        ObjectID = "1111"
    End If

    'if the user has moved location or bobdy position or the object has changed
    If UserPosition <> "" And GPSLatitudeNew.Text <> "" And GPSLongitudeNew.Text <> "" Then
        Call Databasecall 'recheck the data base
    End If

        Call ToDoTest 'checks what current and future activities are
    Then
        Call PlaySound(vbNullString, 0) 'stop playing sound

        Call ExecuteAction 'executes action

        Call SSW.MyGPSIExplorer 'sends url to web browser, loads pictures and outputs speech

    End If

    Debug.Print "Timer Complete"
End Sub

Public Sub MyGPSIExplorer()
    'MsgBox ("Got to here")

    'Sends URL extracted from database and identified by GPS coordinates to IExplorer
    'SendKeys "^o" & GPSURLNew
    'SendKeys "{Enter}"

    frminternet.WebBrowser.Navigate GPSURLNew 'tells the web browser where to go
    frminternet.comBrowser.Text = GPSURLNew 'adds the address to the address bar
    frminternet.comBrowser.AddItem (frminternet.comBrowser.Text)
    frminternet.SetFocus 'sets the browser in focus

    ProgressMonitor2.Caption = ""

    If UserPosition = "Error" Then
        lbuser.FontSize = 12
        lbuser.Caption = "Error User Position can not be found"
        PicUser.Visible = False
    End If
End Sub

```

```

        UserPositionOld = UserPosition

Else

    If UserPositionOld <> UserPosition Or GPSBuildingOld <> GPSBuilding.Text Then 'if user location or body position has
changed
        SpeechCase = UserPosition 'set speechcase to user position
            Call OutputSpeech
        End If

        If ToDoLocation <> BuildingID And ToDoLocation <> "UNN" Then
            SpeechCase = "NotInLocation" 'if the user is not in the right location for appointment give warning
        Call OutputSpeech
        End If

PicUser.Visible = True
PicUser.Picture = LoadPicture(Path & "Pictures\" & Pic)

End If
frminternet.lbBuilding.Caption = GPSBuilding.Text
frminternet.imgBPosition.Picture = LoadPicture(Path & "Pictures\" & Pic)

frminternet.actiVideoX1.ConnectVideo 'connect to camera
startt = Timer

Do Until Timer > startt + 3
    frminternet.On.FillColor = &HFF&
    DoEvents
Loop
frminternet.On.FillColor = &HC0C0C0

frminternet.actiVideoX1.BMPFile = Path & "Pictures\CamImages\Image" & CamImageNum & ".jpg"
frminternet.actiVideoX1.CaptureFrame ffJPEG, 100
'frmInternet.actiVideoX1.DisconnectVideo 'disconnects webcam
Debug.Print "Image Taken"
CamImageNum = CamImageNum + 1
frminternet.actiVideoX1.DisconnectVideo 'disconnect camera

```

```

GPSBuildingOld = GPSBuilding.Text
UserPositionOld = UserPosition
ObjectIDOld = ObjectID

    'add location, object and BodyPosition with a timestamp to the history
    Call LocationHistory
    Call ObjectHistory
    Call BodyPositionHistory
    Debug.Print "mygpsixpolorer complete"
End Sub

Private Sub TmrHistory_Timer()
    Call AutoHistoryOutput
End Sub

```

## A.2 Form frmInternet

```

Dim OptionsOpen As Boolean
Public TaskChanged As Boolean

Private Sub butBack_Click() 'back button
    On Error GoTo nodocinhistory 'if there is not doc in history stay where you are
    WebBrowser.GoBack

    Exit Sub

nodocinhistory:
    WebBrowser.Navigate (comBrowser.Text)

End Sub

Private Sub butForward_Click() 'forward button
    On Error GoTo nodocinhistory 'if there is not doc in history stay where you are
    WebBrowser.GoForward

    Exit Sub

```

```
nodocinhistory:
    WebBrowser.Navigate (comBrowser.Text)
End Sub

Private Sub butOptions_Click() 'shows the content choices
    If OptionsOpen = True Then
        WebBrowser.Width = 485.827
        OptionsOpen = False
        lstOptions.Visible = False
        lbContentChoice.Visible = False
    ElseIf OptionsOpen = False Then
        WebBrowser.Width = 277.953
        lstOptions.Visible = True
        OptionsOpen = True
        lbContentChoice.Visible = True
    End If
End Sub

Private Sub comBrowser_Change()
    butForward.Enabled = True
    butBack.Enabled = True
End Sub

Private Sub comBrowser_Click()
    WebBrowser.Navigate (comBrowser.Text)
End Sub

Private Sub comBrowser_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        comBrowser_Click
    End If
End Sub

Private Sub Form_Load()
    WebBrowser.Navigate "http://www.eee.bham.ac.uk"
    comBrowser.Text = "http://www.eee.bham.ac.uk"
    comBrowser.AddItem (comBrowser.Text)
    comBrowser.AddItem ("http://www.sport.bham.ac.uk")
    comBrowser.AddItem ("http://www.is.bham.ac.uk")
    frminternet.Show
End Sub
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    Call PlaySound(vbNullString, 0) 'stop any sound
    frminternet.actiVideoX1.DisconnectVideo 'disconnects webcam
    MsgBox ("we cam disconnected")
    Unload frmcomm
    Unload frmspeech
    Unload frminternet
    Unload SSW
End
End Sub
```

```
Private Sub lbCurrentTask_Change() 'when the task changes update the content list
If TaskOtherInfo <> "NONE" And ToDoLocation = BuildingID Then 'with task specific info,if any and tell the user
    strspeakspeech = "Your task has content associated with it, use the content choice to view it"
    Call SpeakSpeech(strspeakspeech)
    TaskChanged = True
End If
Call ContentSearch
End Sub
```

```
Private Sub lstOptions_Click()
'when user selects different content, get filename and display
    If InStr(1, lstOptions.List(lstOptions.ListIndex), "Best Option c:\") <> 0 Then
        SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 13, (Len(lstOptions.List(lstOptions.ListIndex)) - 11))
    ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Best Option http://") <> 0 Then
        SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 13, (Len(lstOptions.List(lstOptions.ListIndex)) - 11))
    ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Best Option about:blank") <> 0 Then
        SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 13, (Len(lstOptions.List(lstOptions.ListIndex)) - 11))
    ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Default c:\") <> 0 Then
        SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 9, (Len(lstOptions.List(lstOptions.ListIndex)) - 7))
    ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Default http://") <> 0 Then
        SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 9, (Len(lstOptions.List(lstOptions.ListIndex)) - 7))
    ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Task Content") <> 0 Then
        SSW.GPSURLNew = SSW.Path & "pictures\" & Mid(lstOptions.List(lstOptions.ListIndex), 14,
(Len(lstOptions.List(lstOptions.ListIndex)) - 7))
    Else
        SSW.GPSURLNew = SSW.Path & "pictures\" & Mid(lstOptions.List(lstOptions.ListIndex), 5,
(Len(lstOptions.List(lstOptions.ListIndex)) - 3))
    End If
```

```
WebBrowser.Navigate SSW.GPSURLNew
comBrowser.Text = SSW.GPSURLNew
comBrowser.AddItem (comBrowser.Text)
End Sub

Private Sub muHistory_Click()
    frmLocationHistory.Show
End Sub

Private Sub muMain_Click()
    SSW.Show
End Sub

Private Sub MuShowWebCam_Click()
    frminternet.Show
End Sub

Private Sub muToDoList_Click()
    frmToDo.Show
End Sub

Public Sub WebBrowser_TitleChange(ByVal Text As String)
    Call AddressHistory
End Sub
```

### A.3 Module modCommSetup

```
Public ACCComm As String
Public GPSComm As String
Public OBJComm As String

Public Sub CommSet()
    Dim commsettings As String
    Dim ACCStart As Integer
    Dim GPSStart As Integer
    Dim linetemp As String
```



```

Open (SSW.Path & "commset.ini") For Input As #1 'loads in commset.txt
  Do While Not EOF(1) 'until EOF - end of file
    Line Input #1, linetemp 'reads eachline and adds it to text string
    commsettings = commsettings + linetemp
  Loop
Close #1

ACCStart = InStr(1, commsettings, "ACCComm=") 'Looks for save comm settings
ACCComm = Mid(commsettings, (ACCStart + 8), 1)
GPSStart = InStr(1, commsettings, "GPSComm=")
GPSComm = Mid(commsettings, (GPSStart + 8), 1)
OBJStart = InStr(1, commsettings, "OBJComm=")
OBJComm = Mid(commsettings, (OBJStart + 8), 1)

End Sub

```

#### A.4 Module modBodyPosition

```

Public Sub AccCalculate()

Dim XDataChr As String
Dim YDataChr As String
Dim XData As Double
Dim YData As Double
Dim XdataArray(100) As Double
Dim YdataArray(100) As Double
Dim XDataSum As Double
Dim YDataSum As Double
Dim Counter As Integer
Dim Length As Integer

Length = Len(SSW.AccBuffer.Text)
Counter = 4
  For i = 0 To 100
    DoEvents
      Xdatachrstart = InStr(1, SSW.AccBuffer.Text, "X") 'looks for X
      If Xdatachrstart <> 0 Then 'if there is an X then continues
        XDataChr = Mid(SSW.AccBuffer.Text, (Xdatachrstart + 1), 1) 'gets the X value
        YDataChrStart = InStr(1, SSW.AccBuffer.Text, "Y") 'looks for Y
        If YDataChrStart <> 0 Then 'if there is a Y then continue
          YDataChr = Mid(SSW.AccBuffer.Text, (YDataChrStart + 1), 1) ' gets the Y value

```

```

        XData = (Asc(XDataChr) - 128) / 12.5 'Converts into double
        YData = (Asc(YDataChr) - 128) / 12.5 'converts into double
        XdataArray(i) = XData 'puts X value into array location i
        YdataArray(i) = YData 'puts Y value into array location i
        SSW.AccBuffer.Text = Mid(SSW.AccBuffer.Text, Xdatachrstart + 4, (Length - Counter)) 'removes data
already added to array

        Counter = Counter + 4 'adds the four characters that hae been anaylesed to the counter
        'Debug.Print "Counter"; Counter, "Len"; Length, "Sum"; (Length - Counter), "i"; i 'debug tool

    'Else
    '    SSW.lbAccProgress.Visible = True
    '    SSW.lbAccProgress.Caption = "Error in Acceleromter Signal"
End If

'Else
'    SSW.lbAccProgress.Visible = True
'    SSW.lbAccProgress.Caption = "Error in Acceleromter Signal"
End If

Next

For Z = 0 To 100
    XDataSum = XDataSum + (XdataArray(Z) ^ 2)
    YDataSum = YdataArray(Z) ^ 2)
    'Debug.Print Z
Next
    SSW.XDataAverage = Sqr(XDataSum / 100)
    SSW.YDataAverage = Sqr(YDataSum / 100)

Call AccDecide
Debug.Print "AccCalcuate Comp"

End Sub
Private Sub AccDecide()

'this sub decides wether the user is standing, sitting or walking
SSW.txtXdata.Text = SSW.XDataAverage
SSW.txtYdata.Text = SSW.YDataAverage

    If SSW.XDataAverage > 1.5 Then 'when the users foot hits the ground as the walk the acc is high, therefore the average is
higher.
        SSW.UserPosition = "Walking"
    ElseIf SSW.XDataAverage < 0.6 And SSW.YDataAverage >= 0.9 Then 'if gravity is acting on the y axis i.e. across the leg the
user is sitting

```

```

        SSW.UserPosition = "Sitting"
    ElseIf SSW.XDataAverage >= 1.1 And SSW.YDataAverage < 0.8 Then 'if gravity is acting on the x axis i.e. down the leg, user
is standing
        SSW.UserPosition = "Standing"
    End If

Debug.Print "AccDecide Comp"
End Sub

```

## A.5 Module modLocation

```

'Note only works in Noth Hemisphere
Public Sub GPSCalculate()
    On Error GoTo errorlocation
    SSW.GPSStartCode = InStr(1, SSW.GPSBuffer.Text, "$GPRMC") 'Checks message is correct message
    If SSW.GPSStartCode <> 0 Then
        SSW.GPSLatitudeStart = InStr(1, SSW.GPSBuffer.Text, "A,") 'Looks for start of Latitude
        If SSW.GPSLatitudeStart <> 0 Then 'If no latitude goes no futher
            SSW.GPSLatitudeNew.Text = Mid(SSW.GPSBuffer.Text, (SSW.GPSLatitudeStart + 2), 9) 'Take Latitude from buffer
            SSW.GPSLongitudeStart = InStr(1, SSW.GPSBuffer.Text, "N,") 'Looks of londitude start
            If SSW.GPSLongitudeStart <> 0 Then 'If no Longitude goes no futher
                SSW.GPSLongitudeNew.Text = Mid(SSW.GPSBuffer.Text, SSW.GPSLongitudeStart + 2, 10) 'Takes Longitude from
buffer
            Else 'There is a GPS signal but it is corrupted
                SSW.ProgressMonitor2.Visible = True
                SSW.ProgressMonitor2.Caption = "GPS Signal is corrupted"
            End If
        Else 'There is a GPS signal but it is corrupted
            SSW.ProgressMonitor2.Visible = True
            SSW.ProgressMonitor2.Caption = "GPS Signal is corrupted"
            'therefore use te last know location
            SSW.GPSLatitudeNew.Text = LatitudeHistory(LocationNumber - 1)
            SSW.GPSLongitudeNew.Text = LongitudeHistory(LocationNumber - 1)
        End If

    Else 'There is a GPS signal but it is corrupted
        SSW.ProgressMonitor2.Visible = True
        SSW.ProgressMonitor2.Caption = "GPS Signal is corrupted"
        'therefore use the last know location
        SSW.GPSLatitudeNew.Text = LatitudeHistory(LocationNumber - 1)
        SSW.GPSLongitudeNew.Text = LongitudeHistory(LocationNumber - 1)
    End If

```

```

End If
Exit Sub
errorlocation:

'MsgBox ("ERROR")
SSW.GPSLatitudeNew.Text = 5227.2
SSW.GPSLongitudeNew.Text = 156.921
End Sub

```

## A.6 Module modDayTime

```

Public CurrentTime As Date
Public WarningTime As Date

Public Sub DayTime()

Dim todaydate As String

    CurrentTime = Format(Now, "hh:mm") 'looks up current system time and formats to show SHORT TIME
    frminternet.lbTime = CurrentTime 'Shows the time on mail screen
    WarningTime = DateAdd("n", 15, CurrentTime) 'calculates the time 15 mins from current time

'Buffer$ = "$GPRMC,183004,A,5227.2000,N,00156.9210,W,000.0,000.0,141200,004.1,W*68" 'elec eng

TimeStart = InStr(1, SSW.GPSBuffer.Text, "GPRMC,") 'Looks for start of Latitude
If TimeStart <> 0 Then 'If no latitude goes no futher
    FullTime = Mid(SSW.GPSBuffer.Text, (TimeStart + 6), 6) 'Take Time from buffer
    hours = Mid(FullTime, 1, 2)
    minutes = Mid(FullTime, 3, 2)
    Seconds = Mid(FullTime, 5, 2)
    SSW.Time.Text = hours & ":" & minutes & ":" & Seconds

    fulldate = Mid(SSW.GPSBuffer.Text, (TimeStart + 52), 6) 'Take Date from buffer
    Amonth = Mid(fulldate, 3, 2)
    wmonth = MonthName(Amonth)
    wyear = Mid(fulldate, 5, 2)
    today = Mid(fulldate, 1, 2)
    todaydate = today & " " & wmonth & " " & wyear
    Whichday = Weekday(todaydate, 0) 'decides which day i.e. Monday Tuesday...
    WhatDay = WeekdayName(Whichday) 'Decides if a Whichday returns a number between 1 and 7
    SSW.Date.Text = WhatDay & " " & today & " " & wmonth & " 20" & wyear

```

```

Else 'There is a GPS signal but it is corrupted
    SSW.ProgressMonitor3.Visible = True
    SSW.ProgressMonitor3.Caption = "Time Signal is corrupted"
End If

If (Whichday = 6 Or Whichday = 7) Then 'If a weeks day then display weekdayif not weekend
SSW.WeekDayEnd.Text = "WeekEnd"
Else: SSW.WeekDayEnd.Text = "WeekDay"
End If

```

```
End Sub
```

## A.7 Module modMovement

```

Public Sub GlobalMovement()

'Buffer$ = "$GPRMC,183004,A,5227.2000,N,00156.9210,W,000.0,000.0,141200,004.1,W*68" 'elec eng

header = InStr(1, SSW.GPSBuffer.Text, "GPRMC,") 'Looks for start of Latitude
If header <> 0 Then 'If no latitude goes no futher
    MovementStart = InStr(1, SSW.GPSBuffer.Text, "W,")
    SpeedValue = Mid(SSW.GPSBuffer.Text, (MovementStart + 2), 5) 'Take speed from buffer
    SSW.txtSpeed.Text = SpeedValue & " Knots"
    SSW.txtDirection.Text = (Mid(SSW.GPSBuffer.Text, (MovementStart + 8), 5)) & " degrees"

    If (SpeedValue > "0010.0") Then
        SSW.movement.Text = "Moving"
    Else: SSW.movement.Text = "Not Moving"
    End If

Else 'There is a GPS signal but it is corrupted
    SSW.ProgressMonitor3.Visible = True
    SSW.ProgressMonitor3.Caption = "Time Signal is corrupted"
End If

End Sub

```

## A.8 Module modObject

```

Public CodedObjectID As String
Public ObjectTransmitting As Boolean

Public Sub objectcalc()

    objectstartcode = InStr(1, SSW.ObjectBuffer, "OBJECT") ' looks for the prefix OBJECT,
    If objectstartcode <> 0 Then
        CodedObjectID = Mid(SSW.ObjectBuffer, (objectstartcode + 7), 1)
        Call DecToBin ' call procedure to decode object id (converts the decimal value of the ASCII
        'character to the binary code)
        Debug.Print "Object ID "; SSW.ObjectID
        If ObjectTransmitting = False Then 'since the decoder latches need to know if transmission is
        'actually going on or if it has just stored the last transmission
        SSW.ObjectID = "0000" 'if not transmitting then user in not interacting with object
        'even though the object code is still coming through, therefore set objct code to zero
        End If

    Else
        SSW.progressmonitor4.Visible = True
        SSW.progressmonitor4.Caption = "Object Signal Error"
    End If

End Sub

Public Sub DecToBin()
'Modified from 'Modified by David Cowan from KB article #109260'
'Declare variables...
'lCount(thats L) -- used to loop...
'lConvert -- the decimal to be converted...
'sBinary -- the final answer, in binary format...
'iMaxpower -- maximum length of Binary digits...
Dim lCount As Long, lConvert As Long, sBinary As String
Dim iMaxpower As Integer

txtDecimal = Asc(CodedObjectID)
sBinary = "" 'Build the desired binary number in this string...
lConvert = Val(txtDecimal) 'Convert decimal string in txtDecimal to long integer...

' Here is the heart of the conversion from decimal to binary...

```

```

For lCount = 7 To 0 Step -1
  If lConvert And (2 ^ lCount) Then ' Use the logical "AND" operator...
    sBinary = sBinary + "1"
  Else
    sBinary = sBinary + "0"
  End If
Next

SSW.ObjectID = Mid(sBinary, 5, 4) 'The fourleast significant bits of the binary code corresponding to PIC pins RB0 to RB3
ObjectTransmitting = Mid(sBinary, 4, 1) 'The fith significant bit of the binary code, telling the program if the object is still
present

Debug.Print "Object Transmitting "; ObjectTransmitting

End Sub

```

## A.9 Module modDataBase

```

Public otherinfo As String
Public Action As String
Public BuildingID As String

Public MailList As String
Public Attach As String
Public BodyPositionID As String
Public ObjecttxtID As String
Public DefaultLocationURL As String

Public Sub Databasecall()

  Dim db As Database
  Dim rst As Recordset 'Set of records that meet query
  Dim rstUserPos As Recordset
  Dim rstObject As Recordset

  Dim lat
  Dim lng
  Dim qsql
  Dim num
  Dim User
  Dim object

  Dim UserPosID As String

```

```

lat = SSW.GPSLatitudeNew.Text
lng = SSW.GPSLongitudeNew.Text
User = SSW.UserPosition
object = SSW.ObjectID

qsql = "SELECT * FROM Location WHERE LatitudeTop > " & lat _
& " AND LatitudeBottom < " & lat & " AND LongitudeTop > " _
& lng & " AND LongitudeBottom < " & lng

'Checks Location with database
Set db = OpenDatabase(SSW.Path & "context.mdb")
Set rst = db.OpenRecordset(qsql, dbOpenDynaset)

If rst.RecordCount <> 0 Then

    rst.MoveLast
    num = rst.RecordCount
    rst.MoveFirst

    For l = 1 To num
        SSW.GPSBuilding.Text = rst![Building]
        DefaultLocationURL = rst![Action]
        'otherinfo = rst![otherinfo]
        BuildingID = rst![BuildingID]
        MailList = rst("MailList") 'gets mail list for building
        rst.MoveNext
        Debug.Print "Counter l"; l
    Next

Else

BuildingID = "UNN"
SSW.GPSBuilding.Text = "An Unknown Building"
Action = "Display"
DefaultLocationURL = SSW.Path & "map.htm"
End If

'Checks BodyPosition with the table for the location found above

qsqlUser = "SELECT * FROM BodyPosition WHERE BodyPosition ='" & User & "'"

Set rstUserPos = db.OpenRecordset(qsqlUser, dbOpenDynaset)

If rstUserPos.RecordCount <> 0 Then

```



```

rstUserPos.MoveLast
num = rstUserPos.RecordCount
rstUserPos.MoveFirst

    For j = 1 To num
        SSW.Pic = rstUserPos![Picture]          'gets picture of user sitting, standing etc
        Action = rstUserPos![Action] 'get action
        BodyPositionID = rstUserPos("BodyPositionID") 'get bodyposition ID
        otherinfo = rstUserPos("otherinfo")
        rstUserPos.MoveNext
        Debug.Print "Counter j"; j
    Next

Else

End If

'Checks Object with the table for the location and body position found above
If object <> "" Then 'if the user isn't interacting with a object then keeps action from before
    sqlobject = "SELECT * FROM Object WHERE ObjectID =" & object & "" 'selects object from database for the given objectID

    Set rstObject = db.OpenRecordset(sqlobject, dbOpenDynaset) 'sets recordset for object

If rstObject.RecordCount <> 0 Then 'checks there are records

    rstObject.MoveLast          'finds the number of records (should only be one)
    num = rstObject.RecordCount
    rstObject.MoveFirst

    For k = 1 To num
        SSW.txtObjectName = rstObject("ObjectName") 'retrieves object name
        ObjecttxtID = rstObject("ObjecttxtID")
        Action = rstObject("Action") 'retrieves action
        Attach = rstObject("OtherInfo")
        otherinfo = rstObject("OtherInfo")
        If rstObject("MailList") <> "NONE" Then 'if there is a different
            MailList = rstObject("MailList") 'mailing list for the object
        End If 'then update mailing list
        rstObject.MoveNext

        Debug.Print "Counter k"; k
    Next

    frminternet.picObject.Visible = True
    frminternet.picObject.Picture = LoadPicture(SSW.Path & "Pictures\" & ObjecttxtID & ".jpg")

```

```
Else
SSW.txtObjectName.Text = ""
ObjecttxtID = ""
frminternet.picObject.Visible = False
End If
End If

db.Close 'closes database
```

```
Debug.Print "Database Comp"
End Sub
```

## A.10 Module modToDo

```
Public CurrentTask As String
Public NextTask As String
Public RecordNumber As Integer
Public ToDoLocation As String
Public PlayFirstTime As Integer
Public TaskOtherInfo As String
```

```
Public Sub ToDo()
```

```
Dim dbToDo As Database
Dim rstToDo As Recordset
Dim rstTask As Recordset
```

```
qsql = "SELECT * FROM ToDo ORDER BY StartTime"
qsqlNonTime = "SELECT * FROM Task"
```

```
'creates to do list
Set dbToDo = OpenDatabase(SSW.Path & "context.mdb")
Set rstToDo = dbToDo.OpenRecordset(qsql, dbOpenDynaset)
```

```

If rstToDo.RecordCount <> 0 Then

    rstToDo.MoveLast
    num = rstToDo.RecordCount
    rstToDo.MoveFirst

    For i = 1 To num
        Item = rstToDo("Task")
        If Item = CurrentTask Then
            frmToDo.txtTask.Font.Bold = True
        Else
            frmToDo.txtTask.Font.Bold = False
        End If
        frmToDo.txtTask.Text = frmToDo.txtTask.Text & Item & vbCrLf
        Item = rstToDo("StartTime")
        frmToDo.txtStartTime.Text = frmToDo.txtStartTime.Text & Item & vbCrLf
        Item = rstToDo("EndTime")
        frmToDo.txtEndTime.Text = frmToDo.txtEndTime.Text & Item & vbCrLf

        Item = rstToDo("Location")
        frmToDo.txtLocation.Text = frmToDo.txtLocation.Text & Item & vbCrLf

        rstToDo.MoveNext
    Next

Else

    MsgBox ("Nothing To do today")
End If

Set rstTask = dbToDo.OpenRecordset(qsqlNonTime, dbOpenDynaset)

If rstTask.RecordCount <> 0 Then
    rstTask.MoveLast
    num2 = rstTask.RecordCount
    rstTask.MoveFirst

    For i = 1 To num2
        frmToDo.txtTaskNonTime = frmToDo.txtTaskNonTime & rstTask("Task") & vbCrLf
        rstTask.MoveNext
    Next
Else
    MsgBox ("No Tasks Today")
End If

```

```
dbToDo.Close

End Sub

Public Sub ToDoTest()
'checks the todo list and tells the user what the current and next activities are

    Dim dbToDo As Database
    Dim rstToDoTest As Recordset
    Dim rstToDoWarning As Recordset

    qsqlToDoTest = "SELECT * FROM ToDo WHERE StartTime <= #" & CurrentTime & "# And EndTime >= #" & CurrentTime & "#"
'checks what time user should be doing at this time on their todo list
    Set dbToDo = OpenDatabase(SSW.Path & "context.mdb")
    Set rstToDoTest = dbToDo.OpenRecordset(qsqlToDoTest, dbOpenDynaset)

    If rstToDoTest.RecordCount <> 0 Then

        rstToDoTest.MoveLast
        num = rstToDoTest.RecordCount
        rstToDoTest.MoveFirst

        For i = 1 To num
            ToDoLocation = rstToDoTest("Location")
            TaskOtherInfo = rstToDoTest("otherinfo")
            frminternet.lbCurrentTask.Caption = ("Your current appointment is " & rstToDoTest("Task"))
            CurrentTask = rstToDoTest("Task")

            rstToDoTest.MoveNext
        Next

    Else

        TaskOtherInfo = "NONE"
        frminternet.lbCurrentTask.Caption = "Your Time is free"

        ToDoLocation = "UNN"
    End If
End Sub
```

```

    qsqlToDoWarning = "SELECT * FROM ToDo WHERE StartTime <= #" & WarningTime & "# And StartTime > #" & CurrentTime & "# And
EndTime > #" & CurrentTime & "# ORDER BY StartTime"

'checks what time user should be doing in 15 mins time on their todo list
Set dbToDo = OpenDatabase(SSW.Path & "context.mdb")
Set rstToDoWarning = dbToDo.OpenRecordset(qsqlToDoWarning, dbOpenDynaset)

If rstToDoWarning.RecordCount <> 0 Then

    rstToDoWarning.MoveLast
    num = rstToDoWarning.RecordCount
    rstToDoWarning.MoveFirst
    Debug.Print "Number of records", num
    'If num > 1 Then 'And num <> 2 Then
    ' rstToDoWarning.MoveNext
    'End If

        If (DateDiff("n", CurrentTime, rstToDoWarning("StartTime"))) = 1 Then
            minnumber = " min"
        Else
            minnumber = " mins"
        End If

        frminternet.lbNextTask.Caption = _
        ("Your next appointment is " & rstToDoWarning("Task") _
        & ", in ") & (DateDiff("n", CurrentTime, rstToDoWarning("StartTime"))) _
        & minnumber

        If (DateDiff("n", CurrentTime, rstToDoWarning("StartTime"))) = 15 And PlayFirstTime = 1 Then

            'Call PlaySound(SSW.Path & "sounds\notify.wav", 0)
            strspeakspeech = "Your next appointment is in 15 minutes."
            Call SpeakSpeech(strspeakspeech)
            PlayFirstTime = 2
        End If

        NextTask = rstToDoWarning("Task")

Else

End If
dbToDo.Close
If CurrentTask = NextTask Then

```

```

        frminternet.lbNextTask.Visible = False
        PlayFirstTime = 1

    Else
        frminternet.lbNextTask.Visible = True

    End If

    Debug.Print "ToDoTest Complete"
End Sub

Public Sub OutPutToDo()
ToDoFile = "pictures\" & ObjecttxtID & BodyPositionID & BuildingID & ".htm"
Open (SSW.Path & ToDoFile) For Output As #6 'loads in commset.txt
    Dim dbToDo As Database
    Dim rstToDoOutput As Recordset
    Dim rstTaskOutput As Recordset

    qsqltodooutput = "SELECT * FROM ToDo ORDER BY StartTime"

    Print #6, "<span style='font-family:Arial'>"
    Print #6, "<table border=1 cellspacing=0 cellpadding=0 style='border-collapse:collapse;"
Print #6, "border:none;mso-border-alt:solid windowtext .5pt;mso-padding-alt:0cm 5.4pt 0cm 5.4pt'>"
Print #6, "<tr>"
Print #6, "<td width=284 valign=top style='width:213.05pt;border:solid windowtext .5pt;"
Print #6, "padding:0cm 5.4pt 0cm 5.4pt'>"
Print #6, "<p><B> Calendar " & Format(Now, "d/m/yy") & "</p></B>"
'creates to do list
Set dbToDo = OpenDatabase(SSW.Path & "context.mdb")
Set rstToDoOutput = dbToDo.OpenRecordset(qsqltodooutput, dbOpenDynaset)

If rstToDoOutput.RecordCount <> 0 Then

    rstToDoOutput.MoveLast
    num = rstToDoOutput.RecordCount
    rstToDoOutput.MoveFirst

    For i = 1 To num

        Location = rstToDoOutput("Location")
        If Location = BuildingID Then

```

```

formatting = "<span style='color:red'><B>"
Print #6, formatting
task = rstToDoOutput("Task")
If task = CurrentTask Then
    formatting = "<span style='font-size:16.0pt;Font -Family: Arial;color:blue '>"
    Print #6, formatting
End If

StartTime = Format(rstToDoOutput("StartTime"), "hh:mm")
EndTime = Format(rstToDoOutput("EndTime"), "hh:mm")
Print #6, task & " " & StartTime & "-" & EndTime & " " & Location _
& "</p></B></span><span style='font-size:12.0pt;Font -Family: Arial;color:black '>"

Else

task = rstToDoOutput("Task")
If task = CurrentTask Then
    formatting = "<span style='font-size:16.0pt;Font -Family: Arial;color:blue'>"
    Print #6, formatting
End If
StartTime = Format(rstToDoOutput("StartTime"), "hh:mm")
EndTime = Format(rstToDoOutput("EndTime"), "hh:mm")
Print #6, task & " " & StartTime & "-" & EndTime & " " & Location _
& "</p><span style='font-size:12.0pt;Font -Family: Arial; color:black '>"

End If

rstToDoOutput.MoveNext
Next

Else

MsgBox ("Nothing To do today")
End If

Print #6, "<span style='font-size:12.0pt;Font -Family: Arial;color:black '>"
Print #6, "</td>"
Print #6, "<td width=284 valign=top style='width:213.05pt;border:solid windowtext .5pt;"
Print #6, "border-left:none;mso-border-left-alt:solid windowtext .5pt;padding:0cm 5.4pt 0cm 5.4pt'>"
Print #6, "<p><B> Tasks </p></B>"

qsqNonTime = "SELECT * FROM Task"

```

```

Set rstTaskOutput = dbToDo.OpenRecordset(qsqlNonTime, dbOpenDynaset)

If rstTaskOutput.RecordCount <> 0 Then
    rstTaskOutput.MoveLast
    num2 = rstTaskOutput.RecordCount
    rstTaskOutput.MoveFirst

    For i = 1 To num2
        Print #6, "<p>" & rstTaskOutput("Task") & "</p>"
        rstTaskOutput.MoveNext
    Next
Else
    MsgBox ("No Tasks Today")
End If

dbToDo.Close

Print #6, "</td></tr></table>"
Print #6, "<p><span style='color:red'><B> Shown in Red are appointments at your current location. </span></B>"
Print #6, "<span style='color:blue'><B>Shown in Blue is your current appointment.</span></B>"
Close #6
End Sub

```

## A.11 Module modSound

```

'API Declarations in your Module
Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal lpszSoundName As String, ByVal uFlags As Long) As Long

'Constants
'Play sound synchronously (default)
Const SND_SYNC = &H0

'Play sound asynchronously
Const SND_ASYNC = &H1

'Loop the sound until the function sndPlaySound is called again
Const SND_LOOP = &H8

'If sound is playing, dont play it and return false

```



```

    Const SND_NOSTOP = &H10

' registry sounds
Public Const elDefault = ".Default"
Public Const elGPF = "AppGPFault"
Public Const elClose = "Close"
Public Const elEmptyRecycleBin = "EmptyRecycleBin"
Public Const elMailBeep = "MailBeep"
Public Const elMaximize = "Maximize"
Public Const elMenuCommand = "MenuCommand"
Public Const elMenuPopUp = "MenuPopup"
Public Const elMinimize = "Minimize"
Public Const elOpen = "Open"
Public Const elRestoreDown = "RestoreDown"
Public Const elRestoreUp = "RestoreUp"
Public Const elSystemAsterisk = "SystemAsterisk"
Public Const elSystemExclamation = "SystemExclamation"
Public Const elSystemExit = "SystemExit"
Public Const elSystemHand = "SystemHand"
Public Const elSystemQuestion = "SystemQuestion"
Public Const temStart = "SystemStart"

'The PlaySound function
Function PlaySound(FileName As String, Optional Mode As Integer) As Integer

    Dim SoundMode As Integer

    'Check if the file does exist
    'If Dir(FileName) = "" Then
    '    MsgBox "File not found", vbInformation
    '    Exit Function
    'End If

    Select Case Mode

        Case Is = 0
            SoundMode = SND_ASYNC

        Case Is = 1
            SoundMode = SND_SYNC

        Case Is = 2
            SoundMode = SND_LOOP Or SND_ASYNC

        Case Is = 3

```

```

        SoundMode = SND_NOSTOP Or SND_ASYNC

    Case Else
        MsgBox "Invalid Mode received", vbInformation
        Exit Function

    End Select

    'Call the sndPlaySound API function
    PlaySound = sndPlaySound(ByVal FileName, SoundMode)

End Function

```

## A.12 Module modAction

```

Public Attachment As String
Public SpeechCase As String
Public music As String
Public OtherInfoUsed As Boolean

Public Sub ExecuteAction()
    OtherInfoUsed = False
    If ObjecttxtID = "TODO" Then
        Call OutPutToDo
    End If
    If ObjecttxtID = "BED" Then
        Call PlaySound(SSW.Path & "sounds\complicated.wav", 0)
        MsgBox ("Play music")
    End If

    Select Case Action
        Case "mailto:" 'if the action is to mail a message
            Select Case BodyPositionID
                Case "SIT"
                    Attachment = (SSW.Path & "docs\Meeting1Data.txt") ' attaches a doc if needed

                    Call SimpleSend 'call the procedure to send the email

                    Call ContentSearch

                Case "STD"

```

```
        Call SimpleSend
        Call ContentSearch
    Case "WLK" 'if walking then blank screen, output will be of some other method
        SSW.GPSURLNew.Text = "about:blank"
        Call ContentSearch
    End Select

Case "Display" 'if display is the action
    Select Case BodyPositionID ' selects the body position
        Case "SIT" 'if sitting then show the most detailed page associated with building
            If otherinfo = "NONE" Then

                Call ContentSearch
            Else
                SSW.GPSURLNew.Text = otherinfo
                Debug.Print "Other info used!"
                OtherInfoUsed = True
                Call ContentSearch
            End If

        Case "STD" 'if standing then find page that has been adapted for standing
            If otherinfo = "NONE" Then

                Call ContentSearch
            Else
                SSW.GPSURLNew.Text = otherinfo
                Debug.Print "Other info used!"
                OtherInfoUsed = True
                Call ContentSearch
            End If

        Case "WLK" 'if walking then blank screen, output will be of some other method
            SSW.GPSURLNew.Text = "about:blank"
            Call ContentSearch
        End Select
    End Select

    Debug.Print "Action Complete"

End Sub
```

### A.13 Module modSearch

```

Dim LocationWeight As Integer
Dim ObjectWeight As Integer
Dim BodyPositionWeight As Integer
Dim Length As Integer
Dim SearchObject As Boolean
Dim Weight As Integer
Dim Target As Integer
Dim TargetReached As Boolean

Public Sub ContentSearch()
'set weights
LocationWeight = 4
BodyPositionWeight = 3
ObjectWeight = 1
Length = 14
SearchObject = True
Weight = 0

SSW.File1.Path = SSW.Path & "pictures\"
Target = 8 'set the target level
TargetReached = False
If ObjecttxtID = "" Then SearchObject = False 'if there is not object then
Debug.Print "start" 'don't search for the object in the content
frminternet.lstOptions.Clear 'clear the list
For FileCounterA = 0 To 3
    num = SSW.File1.ListCount

    For FileCounter = 0 To (num - 1)

        num2 = InStr(1, SSW.File1.List(FileCounter), BuildingID) 'looks for building id in file name
        num3 = InStr(1, SSW.File1.List(FileCounter), BodyPositionID) 'looks for BP in filename
        If SearchObject = True Then 'if an object is present then look for object id in filename
            num4 = InStr(1, SSW.File1.List(FileCounter), ObjecttxtID)
        Else 'if an object not present then check length of filename
            num4 = 1 'if it only has Building ID and body poss (i.e 10 letters) then reduce the object weighting
            'this makes content that is associated with the correct location and BP adn no object
            'higher weighted than content with Correct location and BP and any object
            If Len(SSW.File1.List(FileCounter)) > 10 Then ObjectWeight = -1
        End If
        If num2 <> 0 Then
            Weight = Weight + LocationWeight 'if location found in content
        
```

```

End If                                     'add location weight
If num3 <> 0 Then
    Weight = Weight + BodyPositionWeight 'if bodypos found in content
End If                                     'add bodypos weight
If num4 <> 0 Then
    Weight = Weight + ObjectWeight 'if object found in content
    ObjectWeight = 1
End If                                     ' add object weight
Debug.Print Weight
Select Case Weight
Case 8 'ideal option where content includes location, body position and object
    SSW.GPSURLNew = SSW.Path & "pictures\" & (SSW.File1.List(FileCounter))
    TargetReached = True
Case Target 'add other content option to list
    frminternet.lstOptions.AddItem ("W=" & Weight & " " & SSW.File1.List(FileCounter))
End Select

Weight = 0 'reset the weight for next pass
Next

Target = Target - 1 'reduce target

Next

If OtherInfoUsed = False And SSW.UserPosition <> "Walking" Then
    If TargetReached = False Then 'if the ideal does not exist then use the default location url
        SSW.GPSURLNew = DefaultLocationURL
        frminternet.lstOptions.AddItem ("Default " & (SSW.GPSURLNew)) 'add default at bottom of list
    Else
        frminternet.lstOptions.AddItem ("Best Option " & (SSW.GPSURLNew)), 0 'add best option to top of list
    End If

Else 'if content specific to object then put that as best
    frminternet.lstOptions.AddItem ("Best Option " & (SSW.GPSURLNew)), 0
    frminternet.lstOptions.AddItem ("Default " & (DefaultLocationURL))
End If

'when the task changes or the context changes and there is task dependent
'information update list
If frminternet.TaskChanged = True Or TaskOtherInfo <> "NONE" Then
    frminternet.lstOptions.AddItem ("Task Content " & TaskOtherInfo), _
    (frminternet.lstOptions.ListCount)
    frminternet.TaskChanged = False
End If
Debug.Print "Default used"
End Sub

```

## A.14 Module modSimpleSend

```

Dim AttachNum As Integer

Public Sub SimpleSend()
AttachNum = 2
sendagain:
    frmEMail.MAPISession1.SignOn

    frmEMail.MAPIMessages1.SessionID = frmEMail.MAPISession1.SessionID

    frmEMail.MAPIMessages1.Compose
    frmEMail.MAPIMessages1.RecipDisplayName = MailList
    frmEMail.MAPIMessages1.MsgSubject = SSW.txtObjectName.Text
    frmEMail.MAPIMessages1.MsgNoteText = "I am in " & SSW.GPSBuilding & " Please Join Me!" & _
vbCrLf & " This message is part of an experiment please just store it, thank you" 'body of message tells
'recipient where user is and asks them to join them

If Attach = "ATTACH" Then

frmEMail.MAPIMessages1.AttachmentPathName = Attachment
    strspeakspeech = "Your code word is hours and staff! An e mail has just been sent with your questionnaire answers"
    Call SpeakSpeech(strspeakspeech)
End If

    frmEMail.MAPIMessages1.Send
'attaches the second and third meeting data and sends it

frmEMail.MAPISession1.SignOff

If AttachNum < 4 And Attach = "ATTACH" Then
    If AttachNum = 2 Then
        Attachment = (SSW.Path & "docs\Meeting2Data.txt")
        AttachNum = 3

        GoTo sendagain
    
```

```

ElseIf AttachNum = 3 Then
    Attachment = (SSW.Path & "docs\Meeting3Data.txt")
    AttachNum = 4

    GoTo sendagain
End If

End If

If ObjecttxtID = "COF" Then
    strspeakspeech = "Your code word is meet. I repeat your code word is meet. An e mail has just been sent to tell your friends
you are having coffee"
    Call SpeakSpeech(strspeakspeech)
End If

End Sub

```

## A.15 Module modSpeech

```

Public strspeakspeech As String

Public Sub female()

    Set SSW.Voice.Voice = SSW.Voice.GetVoices("gender=female").Item(0)
    SSW.lbVoice.Caption = "Female Voice"

End Sub

Public Sub male()

    Set SSW.Voice.Voice = SSW.Voice.GetVoices("gender=male").Item(0)
    SSW.lbVoice.Caption = "Male Voice"

End Sub

Public Sub OutputSpeech()
Dim speech As String
Dim SpeechWarning As String
    speech = ""
    SpeechWarning = ""

    Select Case SpeechCase
    Case "Walking"
        speech = "You are " + SSW.UserPosition + " near " + SSW.GPSBuilding.Text + _

```

```

" If you require more detailed information please stand still or sit down. "
Case "Standing"
speech = "You are " + SSW.UserPosition + " near " + SSW.GPSBuilding.Text + _
" Some information about this location is being shown to you. If you require more detailed information please sit down."
Case "Sitting"
speech = ""
Case "NotInLocation"
SpeechWarning = "Be advised you are not in location for you current appointment."
End Select

'Call the Speak method with the text from above. We use the
SVSFlagsAsync flag to speak asynchronously and return immediately
from this call.

If speech <> "" Then
    SSW.Voice.Speak speech, SVSFPurgeBeforeSpeak Or SVSFlagsAsync 'if the users position or the building they are
near changes stop the speech and remove it from buffer
    SSW.Voice.Speak SpeechWarning, SVSFlagsAsync 'if the users position or the building they are near changes stop
the speech and remove it from buffer

Else
    SSW.Voice.Speak speech, SVSFlagsAsync
    SSW.Voice.Speak SpeechWarning, SVSFlagsAsync
End If
End Sub

Function SpeakSpeech(speech As String)

If strspeakspeech <> "" Then
    SSW.Voice.Speak strspeakspeech, SVSFPurgeBeforeSpeak Or SVSFlagsAsync 'if the users position or the building they are near
changes stop the speech and remove it from buffer
Else
    SSW.Voice.Speak strspeakspeech, SVSFlagsAsync
End If

End Function

```

## A.16 Module modHistory

```

Public LocationNumber As Integer
Public LocationHistoryAry(600) As String
Public LongitudeHistory(600) As String
Public LatitudeHistory(600) As String

```



```

Public ObjectNumber As Integer
Public ObjectHistoryAry(600) As String

Public BodyPositionNumber As Integer
Public BodyPositionHistoryAry(600) As String

Public AddressNumber As Integer
Public AddressHistoryAry(600) As String

Public Sub LocationHistory()
    'adds the users current location to an array
    LocationHistoryAry(LocationNumber) = SSW.GPSBuilding.Text & " " & CurrentTime
    LongitudeHistory(LocationNumber) = SSW.GPSLongitudeNew.Text
    LatitudeHistory(LocationNumber) = SSW.GPSLatitudeNew.Text
    frmLocationHistory.txtLocationHistory.Text = frmLocationHistory.txtLocationHistory.Text & LocationHistoryAry(LocationNumber)
    & vbCrLf
    LocationNumber = LocationNumber + 1
End Sub

Public Sub ObjectHistory()
    'adds the users current object to an array
    ObjectHistoryAry(ObjectNumber) = SSW.txtObjectName.Text & " " & CurrentTime
    frmLocationHistory.txtObjectHistory.Text = frmLocationHistory.txtObjectHistory.Text & ObjectHistoryAry(ObjectNumber) & vbCrLf
    ObjectNumber = ObjectNumber + 1
End Sub

Public Sub BodyPositionHistory()
    'adds the users current body position to an array
    BodyPositionHistoryAry(BodyPositionNumber) = SSW.UserPosition & " " & CurrentTime
    frmLocationHistory.txtBodyPositionHistory.Text = frmLocationHistory.txtBodyPositionHistory.Text & BodyPositionHistoryAry(BodyPositionNumber) & vbCrLf
    BodyPositionNumber = BodyPositionNumber + 1
End Sub

Public Sub RecallHistory()
    'read the locations, body positions and object out of the arrays and displays them
    For m = 0 To (LocationNumber - 1)

```

```

        frmLocationHistory.txtBodyPositionHistory.Text          =          frmLocationHistory.txtBodyPositionHistory.Text          &
BodyPositionHistoryAry(m) & vbCrLf
        frmLocationHistory.txtObjectHistory.Text = frmLocationHistory.txtObjectHistory.Text & ObjectHistoryAry(m) & vbCrLf
        frmLocationHistory.txtLocationHistory.Text = frmLocationHistory.txtLocationHistory.Text & LocationHistoryAry(m) & vbCrLf
    Next
End Sub

Public Sub OutputHistory()
'outputs the location, body position and objects to a file
On Error GoTo usercancel

'opens up the save file box, with default ext as .txt
frmLocationHistory.cldOutputHistory.CancelError = True
frmLocationHistory.cldOutputHistory.InitDir = SSW.Path
frmLocationHistory.cldOutputHistory.Filter = "txt|*.txt"
frmLocationHistory.cldOutputHistory.DefaultExt = "txt"
frmLocationHistory.cldOutputHistory.ShowSave

'opens the file and saves from the history arrays the history
Open (frmLocationHistory.cldOutputHistory.FileName) For Output As #4

    For n = 0 To (LocationNumber - 1)
        Print #4, "," & (LocationHistoryAry(n) & "," & BodyPositionHistoryAry(n) & "," & ObjectHistoryAry(n))
    Next
Close #4
n = 0
MsgBox ("General History Saved")
Exit Sub

usercancel:

Debug.Print "user cancelled"
End Sub

Public Sub OutputAddressHistory()
'outputs all the address used by the user to a file
On Error GoTo usercancel

'opens up the save file box, with default ext as .txt
frmLocationHistory.cldOutputHistory.CancelError = True
frmLocationHistory.cldOutputHistory.InitDir = SSW.Path
frmLocationHistory.cldOutputHistory.Filter = "txt|*.txt"
frmLocationHistory.cldOutputHistory.DefaultExt = "txt"
frmLocationHistory.cldOutputHistory.ShowSave

```

```

'opens the file and saves from the combrowser the address history
Open (frmLocationHistory.cldOutputHistory.FileName) For Output As #4

    For n = 0 To (AddressNumber - 1)
        Print #4, "," & AddressHistoryAry(n)

    Next
Close #4
n = 0
MsgBox ("Address History Saved")
Exit Sub

usercancel:

Debug.Print "user cancelled"

End Sub

Public Sub AddressHistory()
'puts the addresses used by the user in an array
AddressHistoryAry(AddressNumber) = frminternet.WebBrowser.LocationURL
Debug.Print AddressHistoryAry(AddressNumber)
AddressNumber = AddressNumber + 1

End Sub

Public Sub AutoHistoryOutput()

Open (SSW.Path & "docs\autohistory.txt") For Output As #9

    For n = 0 To (LocationNumber - 1)
        Print #9, "," & (LocationHistoryAry(n) & "," & BodyPositionHistoryAry(n) & "," & ObjectHistoryAry(n))

    Next
Close #9
n = 0
Open (SSW.Path & "docs\autoaddress.txt") For Output As #8

    For n = 0 To (AddressNumber - 1)
        Print #8, "," & AddressHistoryAry(n)

    Next
Close #8
n = 0

```

```
End Sub
```

### A.17 Form frmAbout



Figure 2 - Form About

```
Private Sub cmdreturn_Click()  
    Unload frmabout  
End Sub
```

### A.18 Form frmComm

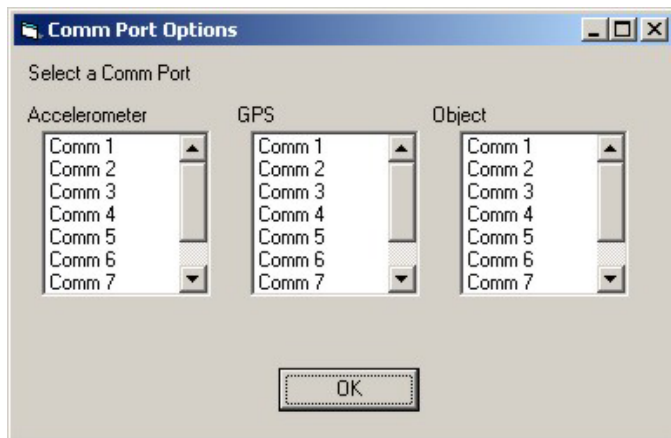


Figure 3 - Form Comm

```

Private Sub Command1_Click()
    Dim commsettings As String

    If lbacc.Caption = "" Or lbgps.Caption = "" Then
        MsgBox ("You have not selected two comm ports please try again")

    ElseIf lbacc.Caption = lbgps.Caption Then
        MsgBox ("You have selected both devices on the same comm port, select again using different comm ports for each device")
    Else

        AccCommNum = Mid(lbacc.Caption, 6, 1)
        GPSCommNum = Mid(lbgps.Caption, 6, 1)
        OBJCommNum = Mid(lbObj.Caption, 6, 1)

        commsettings = "ACCComm=" & AccCommNum & "GPSComm=" & GPSCommNum & "OBJComm=" & OBJCommNum

        Open (SSW.Path & "commset.ini") For Output As #3

            Write #3, commsettings

        Close #3

        SSW.lbAccComm = lbacc.Caption
        SSW.lbGPSComm = lbgps.Caption
    
```

```
SSW.lbObjComm = lObj.Caption

If SSW.MSComm1.PortOpen = True Then
    SSW.MSComm1.PortOpen = False
End If
If SSW.MSComm2.PortOpen = True Then
    SSW.MSComm2.PortOpen = False
End If
If SSW.MSComm3.PortOpen = True Then
    SSW.MSComm3.PortOpen = False
End If

SSW.number = 1

Me.Hide
Call SSW.Form_Load

End If

End Sub

Private Sub Form_Load()
    For X = 1 To 9
        lstACCComm.AddItem ("Comm " & X)
        lstGPSComm.AddItem ("Comm " & X)
        lstObjComm.AddItem ("Comm " & X)
    Next X
End Sub

Private Sub lstACCComm_Click()
    lbacc.Caption = lstACCComm.List(lstACCComm.ListIndex)
End Sub

Private Sub lstGPSComm_Click()
    lbgps.Caption = lstGPSComm.List(lstGPSComm.ListIndex)
End Sub

Private Sub lstObjComm_Click()
    lbObj.Caption = lstObjComm.List(lstObjComm.ListIndex)
End Sub
```

### A.19 Form frmInternet



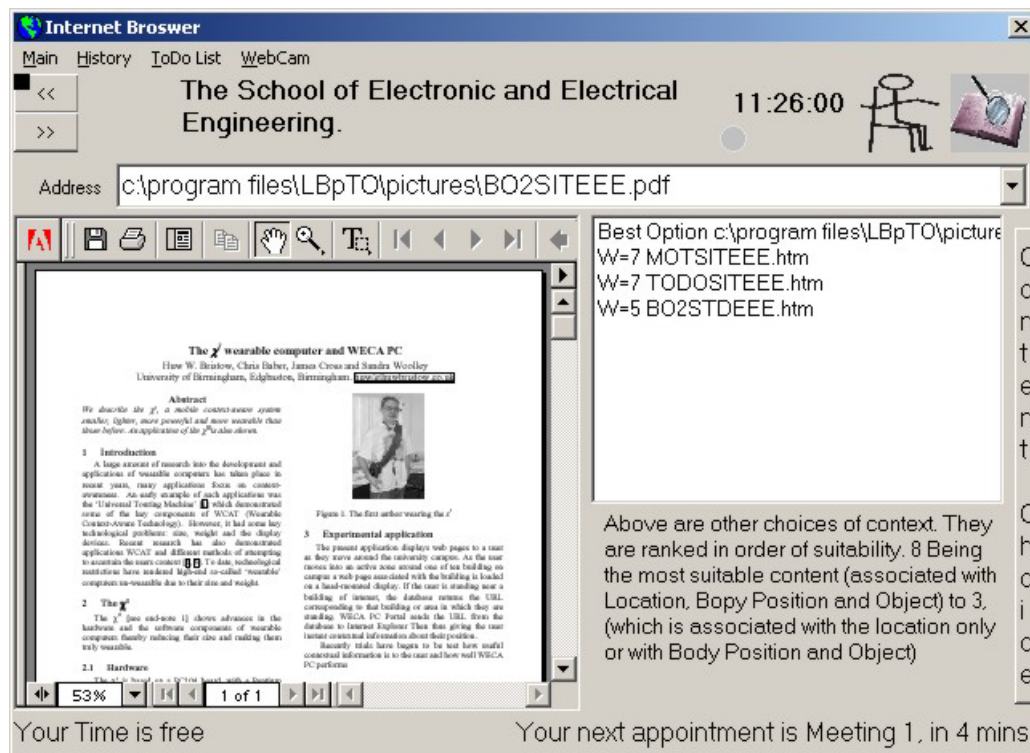


Figure 4 - Form Internet with and without the content choice

```
Dim OptionsOpen As Boolean
Public TaskChanged As Boolean
```

```
Private Sub butBack_Click() 'back button
    On Error GoTo nodocinhistory 'if there is not doc in history stay where you are
    WebBrowser.GoBack
```

```
Exit Sub
```

```
nodocinhistory:
    WebBrowser.Navigate (comBrowser.Text)
```



```
End Sub

Private Sub butForward_Click() 'forward button
    On Error GoTo nodocinhistory 'if there is not doc in history stay where you are
    WebBrowser.GoForward

    Exit Sub

nodocinhistory:
    WebBrowser.Navigate (comBrowser.Text)
End Sub

Private Sub butOptions_Click() 'shows the content choices
    If OptionsOpen = True Then
        WebBrowser.Width = 485.827
        OptionsOpen = False
        lstOptions.Visible = False
        lbContentChoice.Visible = False
    ElseIf OptionsOpen = False Then
        WebBrowser.Width = 277.953
        lstOptions.Visible = True
        OptionsOpen = True
        lbContentChoice.Visible = True
    End If
End Sub

Private Sub comBrowser_Change()
    butForward.Enabled = True
    butBack.Enabled = True
End Sub

Private Sub comBrowser_Click()
    WebBrowser.Navigate (comBrowser.Text)
End Sub

Private Sub comBrowser_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        comBrowser_Click
    End If
End Sub
```

```

Private Sub Form_Load()
    WebBrowser.Navigate "http://www.eee.bham.ac.uk"
    comBrowser.Text = "http://www.eee.bham.ac.uk"
    comBrowser.AddItem (comBrowser.Text)
    comBrowser.AddItem ("http://www.sport.bham.ac.uk")
    comBrowser.AddItem ("http://www.is.bham.ac.uk")
    frminternet.Show
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Call PlaySound(vbNullString, 0) 'stop any sound
    frminternet.actiVideoX1.DisconnectVideo 'disconnects webcam
    'MsgBox ("we cam disconnected")
    Unload frmcomm
    Unload frmspeech
    Unload frminternet
    Unload SSW
End Sub

Private Sub lbCurrentTask_Change() 'when the task changes update the content list
If TaskOtherInfo <> "NONE" And ToDoLocation = BuildingID Then 'with task specific info,if any and tell the user
    strspeakspeech = "Your task has content associated with it, use the content choice to view it"
    Call SpeakSpeech(strspeakspeech)
    TaskChanged = True
End If
Call ContentSearch
End Sub

Private Sub lstOptions_Click()
'when user selects different content, get filename and display
If InStr(1, lstOptions.List(lstOptions.ListIndex), "Best Option c:\") <> 0 Then
SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 13, (Len(lstOptions.List(lstOptions.ListIndex)) - 11))
ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Best Option http://") <> 0 Then
SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 13, (Len(lstOptions.List(lstOptions.ListIndex)) - 11))
ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Best Option about:blank") <> 0 Then
SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 13, (Len(lstOptions.List(lstOptions.ListIndex)) - 11))
ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Default c:\") <> 0 Then

```

```

SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 9, (Len(lstOptions.List(lstOptions.ListIndex)) - 7))
ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Default http://") <> 0 Then
SSW.GPSURLNew = Mid(lstOptions.List(lstOptions.ListIndex), 9, (Len(lstOptions.List(lstOptions.ListIndex)) - 7))
ElseIf InStr(1, lstOptions.List(lstOptions.ListIndex), "Task Content") <> 0 Then
SSW.GPSURLNew = SSW.Path & "pictures\" & Mid(lstOptions.List(lstOptions.ListIndex), 14,
(Len(lstOptions.List(lstOptions.ListIndex)) - 7))
Else
SSW.GPSURLNew = SSW.Path & "pictures\" & Mid(lstOptions.List(lstOptions.ListIndex), 5,
(Len(lstOptions.List(lstOptions.ListIndex)) - 3))
End If
WebBrowser.Navigate SSW.GPSURLNew
comBrowser.Text = SSW.GPSURLNew
comBrowser.AddItem (comBrowser.Text)
End Sub

Private Sub muHistory_Click()
frmLocationHistory.Show
End Sub

Private Sub muMain_Click()
SSW.Show
End Sub

Private Sub MuShowWebCam_Click()
frminternet.Show
End Sub

Private Sub muToDoList_Click()
frmToDo.Show
End Sub

Public Sub WebBrowser_TitleChange(ByVal Text As String)
Call AddressHistory
End Sub

```

## A.20 Form frmLocaitonHistory

Location	Body Position	Object
The School of Electronic and Electrical Engineering. 11:25:00	Sitting 11:25:00	11:25:00
The School of Electronic and Electrical Engineering. 11:25:00	Sitting 11:25:00	Book2 11:25:00

Buttons: Output History To File, Output Address History

Figure 5 - Form History

```
Private Sub butOuputHistory_Click()
    Call OutputHistory
End Sub
```

```
Private Sub Command1_Click()
    Call OutputAddressHistory
End Sub
```

```
Private Sub Form_Load()
    Call RecallHistory
End Sub
```

## A.21 Form frmSpeech



Figure 6 - Form Speech

```
Private Sub Command1_Click()  
    Me.Hide  
End Sub  
  
Private Sub Option3_Click()  
    Call male  
End Sub  
  
Private Sub Option4_Click()  
    Call female  
End Sub
```

## A.22 Form frmTodo

Start Time	End Time	Task	Location
11:30:00	11:40:00	Meeting 1	EEE
12:00:00	12:10:00	Meeting 2	EEE
12:45:00	12:55:00	Meeting 3	EEE

Tasks

- Buy Daily Mail from the guild
- Find Manual of Bridge Building, Edited by M.J. Ryall, G.A.R Parke and J.E. Harding, The Institution of Civil Engineers, TG145M in the main library, go to page 34, Paragraph 2, line 6 Word 5 Record this word.
- Go to the station and buy a train ticket on a train to New St at around 13:00
- Go to EIAC and find Proceedings of MLearn 2002
- Go for coffee in the EEE Common Room
- After your meetings go to the Ed Tech Lab and use your PC

Figure 7 - Form To Do List

```
Private Sub Form_Load()
    Call ToDo
End Sub
```

### A.23 Form frmEmail

