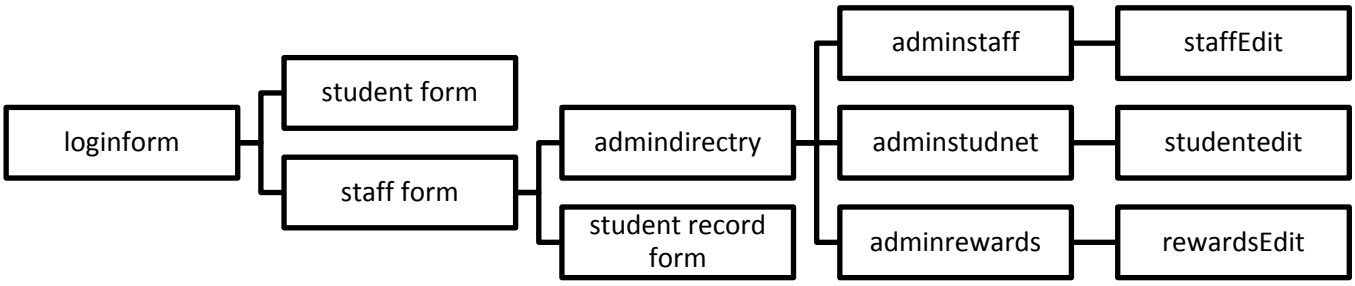
Testing and evaluation

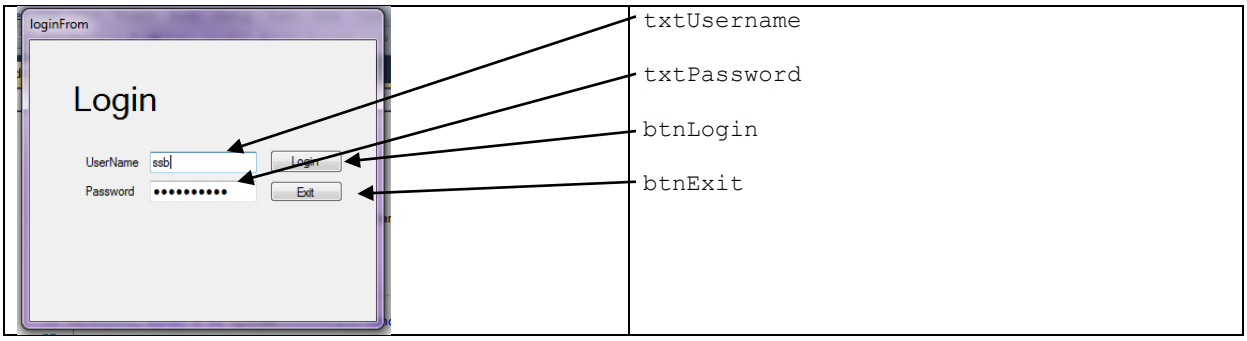
Contents

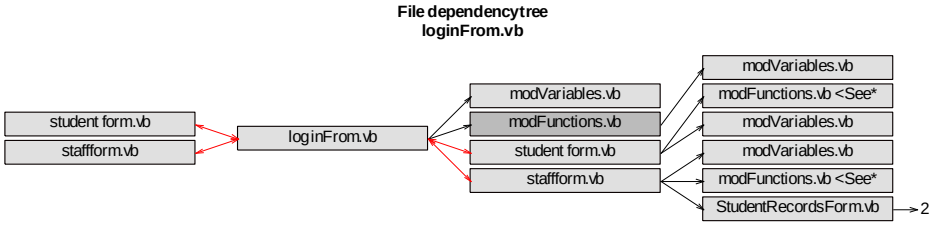
Procedure list

Full structure



loginFrom

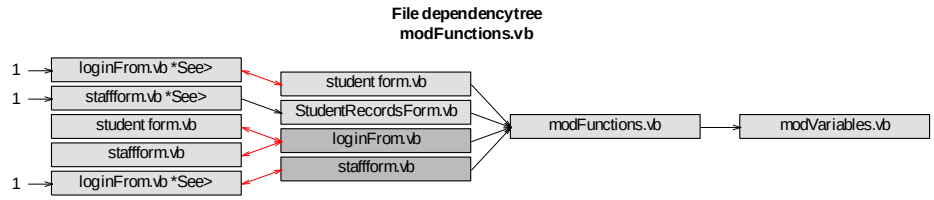





```
Private Sub btnExit_Click
   Closes the login form
Private Sub btnLogin_Click
   Checks if the user is trying to logon as staff or student, if the username is
   numeric then the user is a student, if it is alpha then the user is staff and if it
   is neither then the username is incorrect. The username is compared against the
   usernames of the students or staff respectively and once a match is found the
   password is checked against the password of the record if they are correct the user
   is logged on to the corresponding form, staff for staff student for students.
Private Sub loginFrom_Load
```

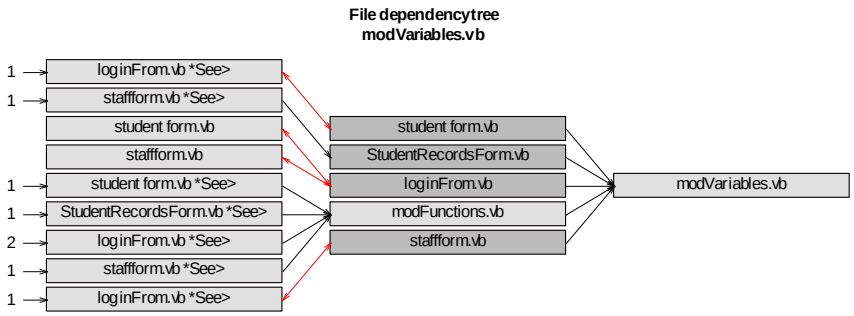
Checks if the student, staff and reward dat files exist, if not then they are created from csv files. It also intialise the numbers of each type of record in the different dat files

modFunctions

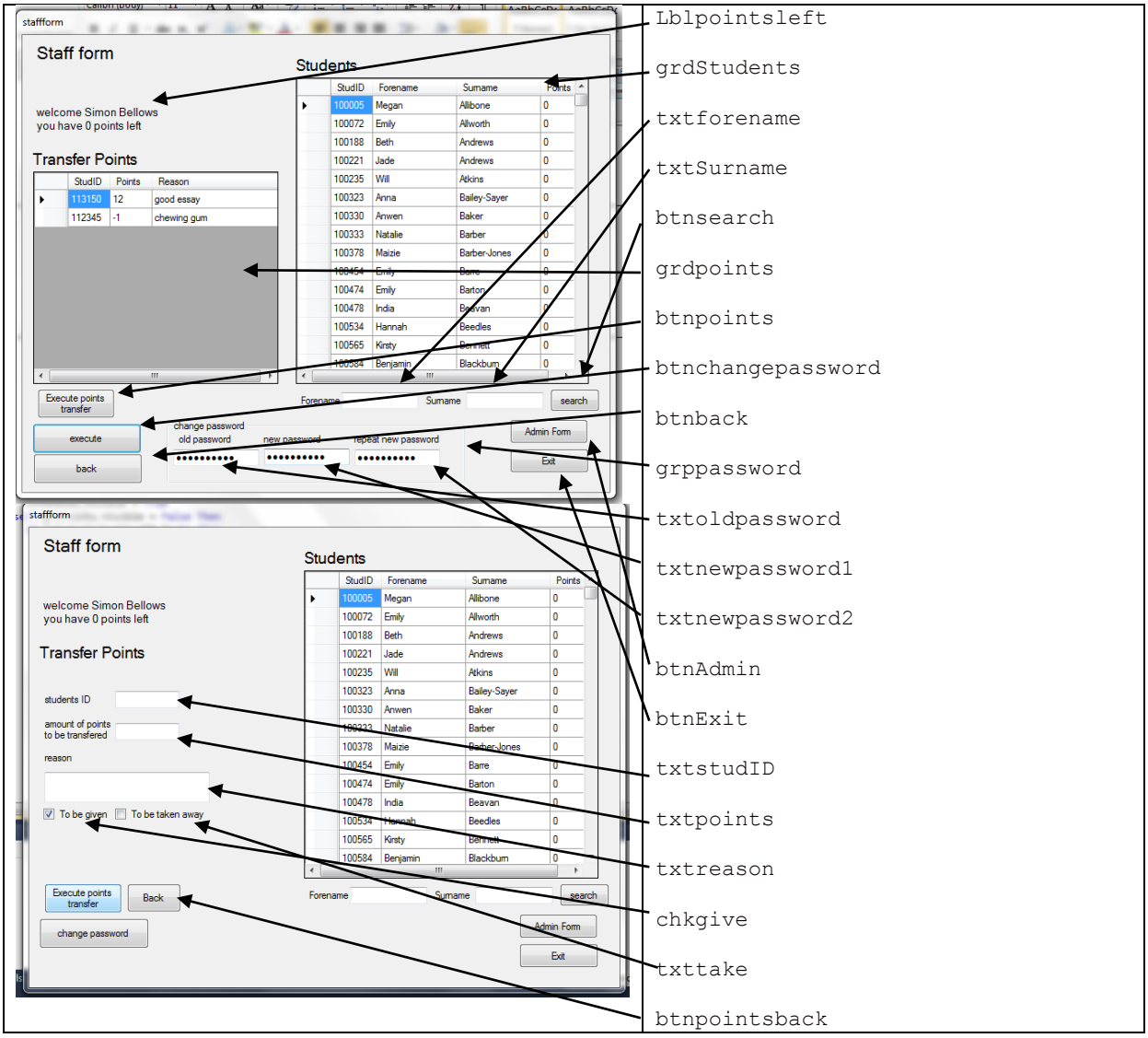


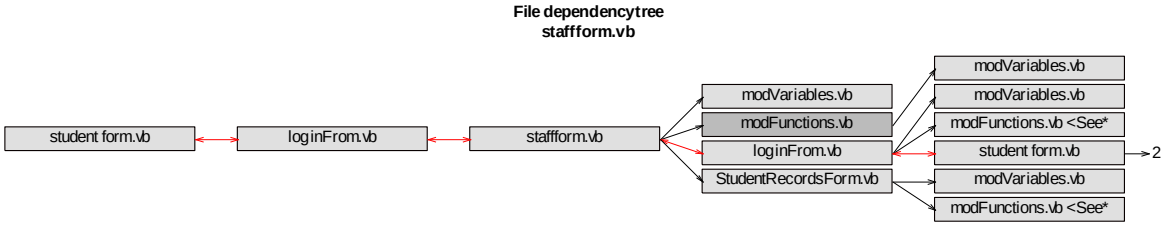
Public Function Decrypt
From Microsoft website. Encrypts strings
Public Function Encrypt
From Microsoft website. Decrypts strings
Public Function GetPointsP
Based on code used in class. Gets a specific record from the pointsProcedure dat file
Public Function GetReward
code used in class. Gets a specific record from the rewardsdat file
Public Function GetRewardP
Based on code used in class. Gets a specific record from the rewardsProcedure dat file
Public Function GetStaff
code used in class. Gets a specific record from the staff dat file
Public Function GetStudent
code used in class. Gets a specific record from the students dat file
Public Sub ImportRewards
Code used in class. Converts a csv file to a dat file using the rewards structure
Public Sub ImportStaff
Code used in class. Converts a csv file to a dat file using the staff structure
Public Sub ImportStudents
Code used in class. Converts a csv file to a dat file using the students structure
Public Function IsAlpha
Code used in class. Checks if a string is made up solely of letters by seeing if each ones ascii code is less then A or greater then Z if so then the string isn't alpha
Public Function IsName
Code used in class. Checks if a string is a name by seeing if each character is alpha or a hyphin, if so then a name.
Public Sub PutPointsP
Based on code used in class. Puts data from the pointsprocedure structure into the pointsprocedure dat file
Public Sub PutReward
code used in class. Puts data from the rewards structure into the rewards dat file
Public Sub PutRewardP
Based on code used in class. Puts data from the rewardsprocedure structure into the rewardsprocedure dat file
Public Sub PutStaff
code used in class. Puts data from the staff structure into the staff dat file
Public Sub PutStudent
code used in class. Puts data from the student structure into the students dat file

modVariables


File dependencytree
modVariables.vb
1
loginFrom.vb *See>
1
staffform.vb *See>
student form.vb
staffform.vb
1
student form.vb *See>
1
StudentRecordsForm.vb *See>
2
loginFrom.vb *See>
1
staffform.vb *See>
1
loginFrom.vb *See>
student form.vb
StudentRecordsForm.vb
loginFrom.vb
modFunctions.vb
staffform.vb
modVariables.vb

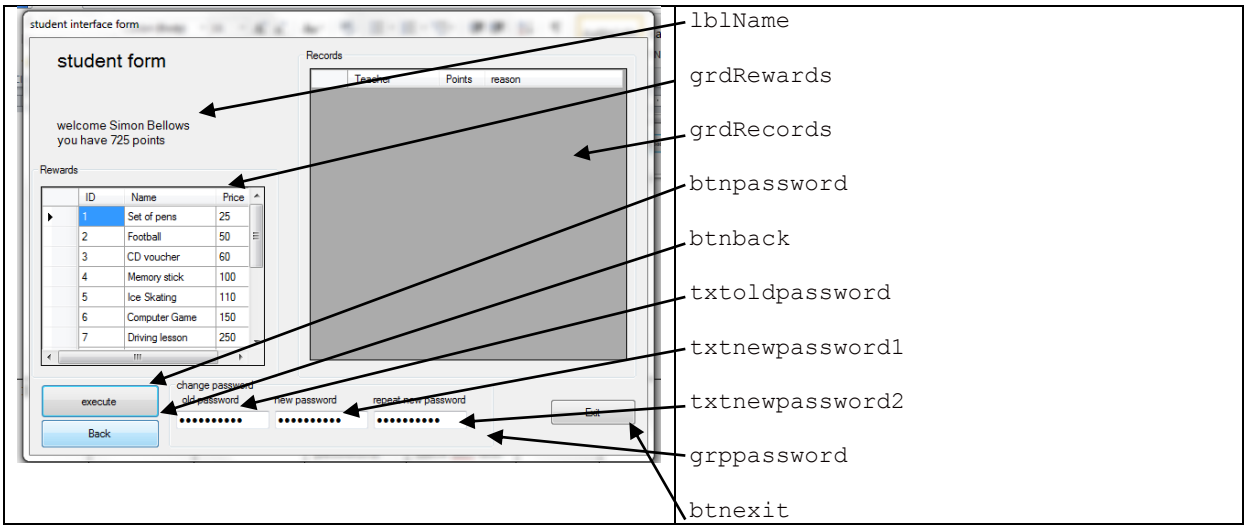
staffform


staffform
Staff form
Students
welcome Simon Bellows
you have 0 points left
Transfer Points
Execute points transfer
execute
back
change password
old password
new password
repeat new password
Admin Form
Exit
search
Forename
Surname
Lblpointsleft
grdStudents
txtforename
txtSurname
btnsearch
grdpoints
btnpoints
btnchangepassword
btnback
grppassword
txtoldpassword
txtnewpassword1
txtnewpassword2
btnAdmin
btnExit
txtstudID
txtpoints
txtreason
chkgive
txttake
btnpointsback
students ID
amount of points to be transfered
reason
To be given
To be taken away
Back

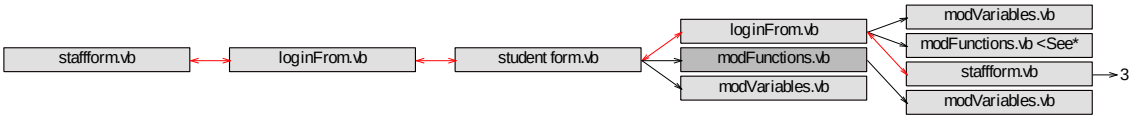


Private Sub btnAdmin_Click
Opens the admindirectory form and closes the staffform
Private Sub btnBack_Click
Hides the grppassword and btnback and changes the text of btnchangepassword to change password
Private Sub btnchangePassword_Click
Checks if grppassword is visible, if it is visible it then checks if the txtoldpassword is inputted and if it matches the oldpassword of the member of staff, if the new password is inputed, greater then 5 characters, is alpha numeric and if txtnewpassword1 is equal to txtnewpassword2. If all these are correct then it changes the password. Makes grp password and btnback invisble and changes the text of btnchangepassword to change password.
If grppassword is invisible then it makes it and btnback visible and changes the text of btnchangepassword to execute.
Private Sub btnExit_Click
Closes the staffform and opens the loginform
Private Sub btnPoints_Click
Checks if grdpoints is visible if so then it makes it invisible and makes all the different points procedure tools visible.
If grdpoints is invisible then it checks if the txtpoints is numeric, does a presence check for it and txt reason, checks if the staff member can afford to give the amount away if give is checked and depending on whether or not chkgive or chktake is checked either gives or takes the points away from the student and if giving the points are taken away from the member of staff, if the student has less points then the amount being taken away his total goes to zero. Then it puts a new pointsprocedure record in the pointsprocedure dat file.
Makes grdpoints visible and makes all the pointsprocedure tools invisible.
Private Sub btnPointsBack_Click
Makes grdpoints visible and makes all the pointsprocedure tools invisible.
Private Sub btnSearch_Click
By presence checks it first works out which of the 4 senarios it is working out, first searching by both forename and surname, second forename only, third surname only, forth neither. For the first it using a for loop checks for each record if the forename begins with txtforename and if the surname begins with txtsurname, if it does then the record is added to grdStudents. For the second it’s the same but with out checking the surname. For the third it’s the same but with out the forename. If it’s the forth then the grid just loads all the students.
Private Sub chkGive_CheckedChanged
When checked changes chktake to unchecked and when unchecked changes chktake to checked
Private Sub chkTake_CheckedChanged
When checked changes chkgive to unchecked and when unchecked changes chkgive to checked
Private Sub grdStudents_CellContentdoubleClick
Finds the content of the cell 0 of the row of the cell clicked on, which is the studID. Uses a for loop to check the studID against all the students till it finds a match, gets the data of the student from the students dat file. It then opens the studentrecordsform.
Private Sub staffform_Load
Populates grdstudents with all of the students on the student dat file
It also populates a lblpointsleft with the name and points of the member of staff

StudentForm1

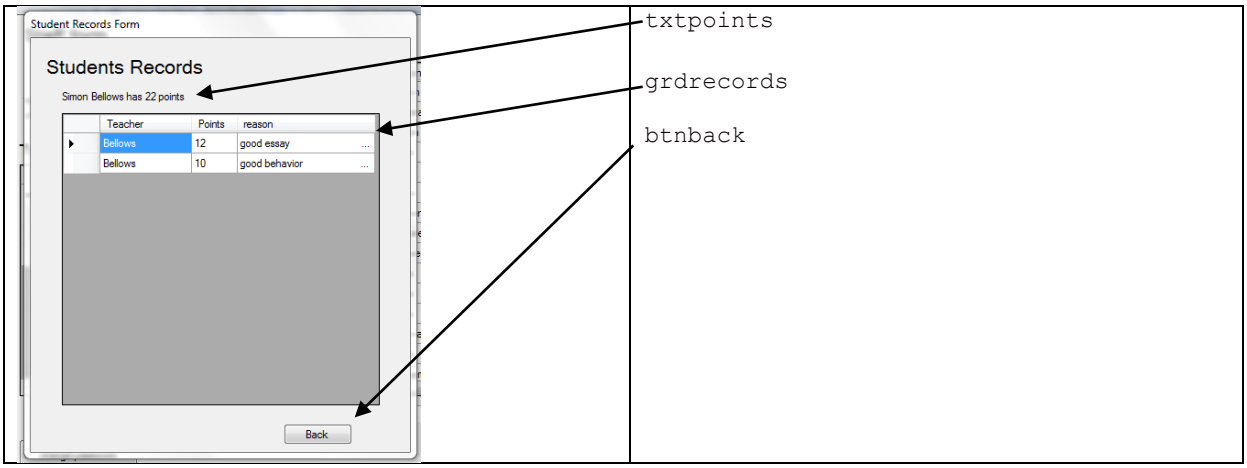


File dependencytree
student form.vb




```
Private Sub btnBack_Click
   Hides the grppassword and btnback and changes the text of btnpassword to change
   password
Private Sub btnExit_Click
   Opens the loginform and closes the student form
Private Sub btnPassword_Click
   Checks if grppassword is visible, if it is visible it then checks if the
   txtoldpassword is inputed and if it matches the oldpassword of the student, if the
   new password is inputed, greater than 5 characters, is alpha numeric and if
   txtnewpassword1 is equal to txtnewpassword2. If all these are correct then it
   changes the password. Makes grppassword and btnback invisible and changes the text
   of btnpassword to change password.
   If grppassword is invisible then it makes it and btnback visible and changes the
   text of btnpassword to execute
Private Sub grdRewards_CellContentDoubleClick
   Finds the content of the cell 0 of the row of the cell clicked on, which is the
   rewardID. Gets the data of the reward from the rewards dat file. It then opens a
   message box which asks if the student wants to buy the rewards yes/no. if no then
   the sub ends. If yes then the cost of the reward is taken away from the students
   points, the grdrewards is refreshed and a rewardsprocedure record is created and put
   into the rewardsprocedure dat file
Private Sub StudentForm1_Load
   Populates grdrewards with rewards that can be afforded by the student. This is done
   by using a for loop to go through all of the different rewards and adding those that
   have a cost smaller or equal to the points of the student.
   It also populates a label with the name and points of the student
```

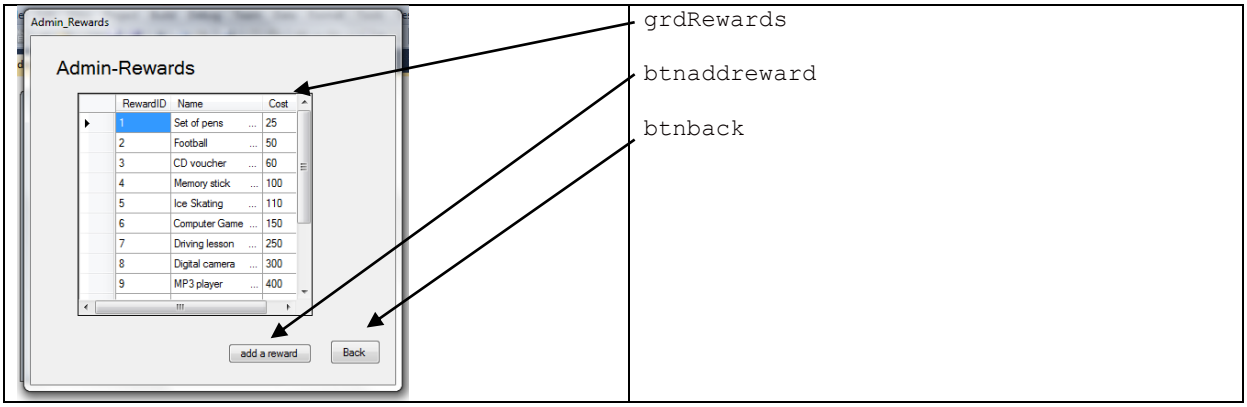
StudentRecordsForm



File dependencytree
StudentRecordsForm.vb


```
Private Sub btnBack_Click
   Closes the studentrecordsform
Private Sub StudentRecordsForm_Load
   Populates grdrecords with record that pertain to the student selected. This is done
   by using a for loop to go through all of the different pointsprocedure records and
   adding those that involve the student by checking if the studID matches that of the
   record.
   It also populates a label with the name and points for the student selected
```

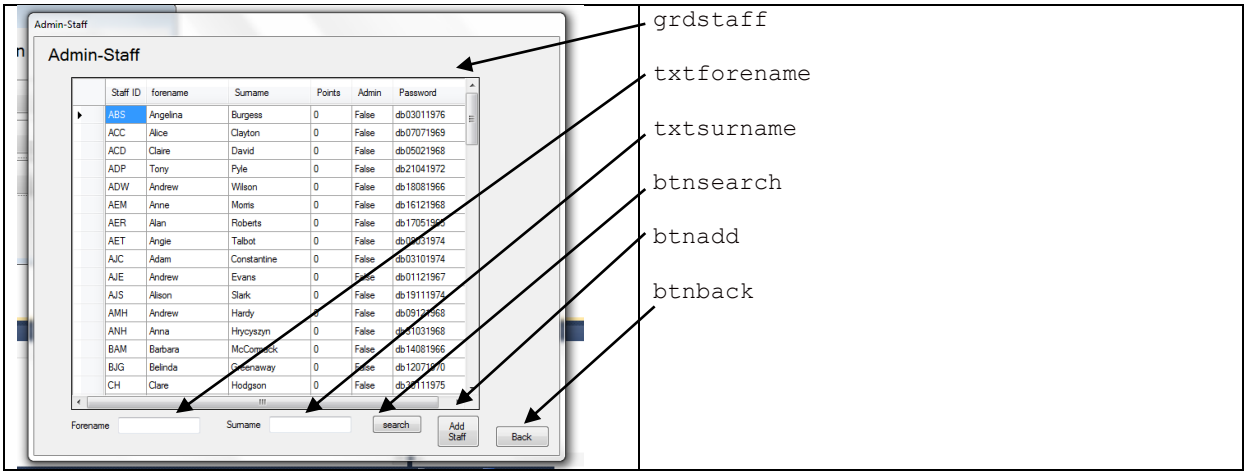
Admin_Rewards



File dependencytree
Admin-Rewards.vb


```
Private Sub Admin_Rewards_Load
   Populates grdRewards by getting the rewards from the rewards dat and then putting
   them in the data grid 1 at a time using a for loop
Private Sub btnAddReward_Click
   opens the rewardsEdit form and closes the admin_rewards form. Also sets the global
   variable adding to true that determines how the rewardsEdit form opens
Private Sub btnBack_Click
   Opens the admindirectory form and closes the admin_rewards form.
Private Sub grdRewards_CellContentdoubleClick
   Finds the content of the cell 0 of the row of the cell clicked on, which is the
   rewardID. Gets the data of reward from the rewards dat file. Sets the global
   variable adding to false and opens the rewardsEdit form.
```

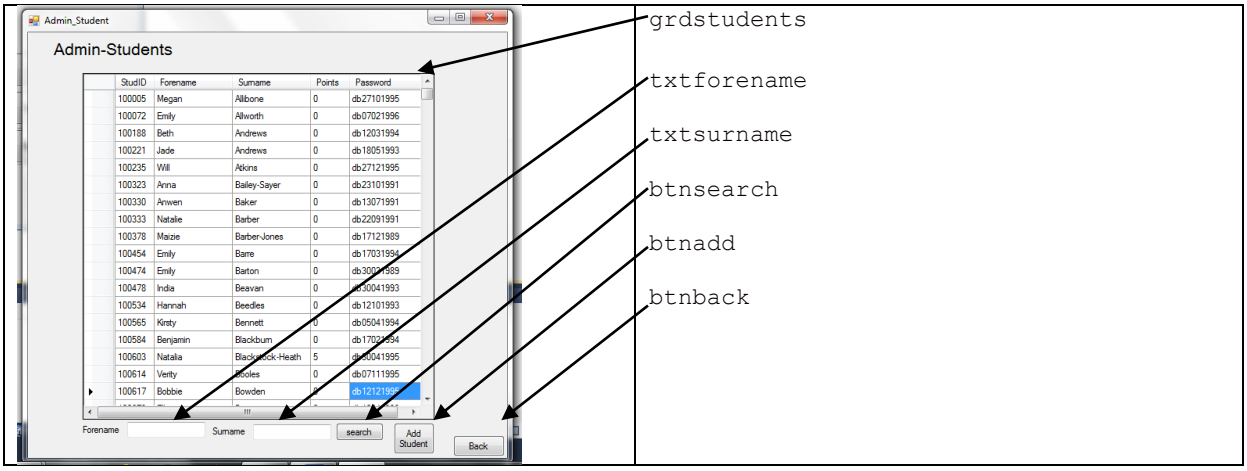
Admin_Staff



File dependencytree
Admin-Staff.vb


```
Private Sub Admin_Staff_Load
   Populates grdstaff by getting the staff from the staff dat file and then putting
   them in the data grid 1 at a time using a for loop
Private Sub btnAdd_Click
   Opens the staffedit form and close the admin_staff form. Also sets the global
   variable adding to true that determines how the staffEdit form opens
Private Sub btnBack_Click
   Opens the admindirectory form and closes the admin_staff form
Private Sub btnSearch_Click
   By presence checks it first works out which of the 4 senarios it is working out,
   first searching by both forename and surname, second forename only, third surname
   only, forth neither. For the first it using a for loop checks for each record if the
   forename begins with txtforename and if the surname begins with txtsurname, if it
   does then the record is added to grdStaff. For the second it's the same but with out
   checking the surname. For the third it's the same but with out the forename. If it's
   the forth then the grid just loads all the staff.
Private Sub grdStaff_CellContentdoubleClick
```

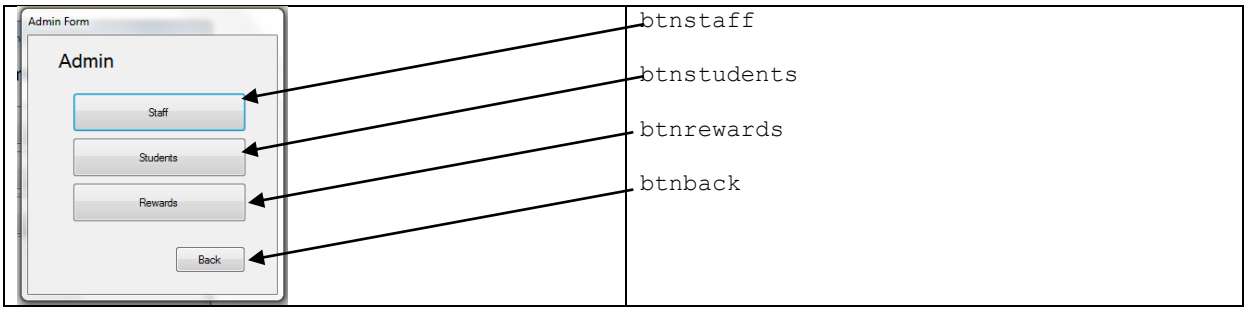
Admin_Student



File dependencytree
Admin-Student.vb


```
Private Sub Admin_Student_Load
   Populates grdstudents by getting the student from the student dat file and then
   putting them in the data grid 1 at a time using a for loop
Private Sub btnAdd_Click
   Opens the studentedit form and close the admin_studnet form. Also sets the global
   variable adding to true that determines how the studentsEdit form opens
Private Sub btnBack_Click
   Opens the admindirectory form and closes the admin_student form
Private Sub btnSearch_Click
   By presence checks it first works out which of the 4 senarios it is working out,
   first searching by both forename and surname, second forename only, third surname
   only, forth neither. For the first it using a for loop checks for each record if the
   forename begins with txtforename and if the surname begins with txtsurname, if it
   does then the record is added to grdStudent. For the second it's the same but with
   out checking the surname. For the third it's the same but with out the forename. If
   it's the forth then the grid just loads all the students.
Private Sub grdStudents_CellContentdoubleClick
   Finds the content of the cell 0 of the row of the cell clicked on, which is the
   studentID. Then it uses a for loop to check the studentID against that of every
   student until a match is found. Gets the data of the student from the student dat
   file. Sets the global variable adding to false and opens the studentEdit form.
```

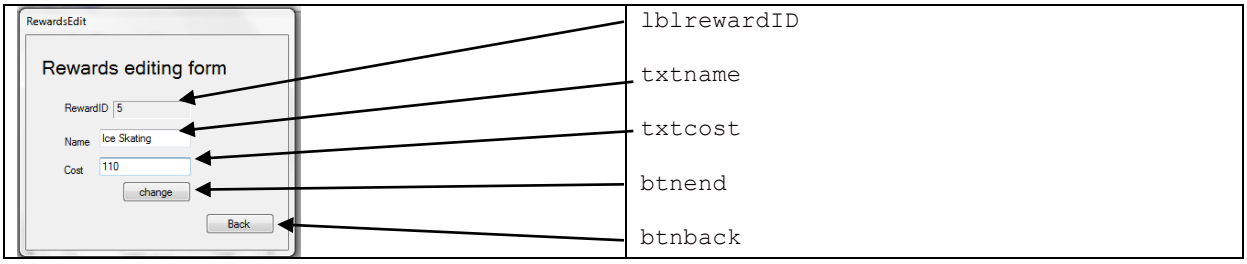
AdminDirectory



File dependencytree
AdminDirectory.vb


```
Private Sub btnAdminRewards_Click
   opens admin_rewards and hides adminDirectory
Private Sub btnAdminStaff_Click
   opens admin_staff and hides adminDirectory
Private Sub btnAdminStudents_Click
   opens admin_students and hides adminDirectory
Private Sub btnBack_Click
   opens staffform and closes admindirectory
```

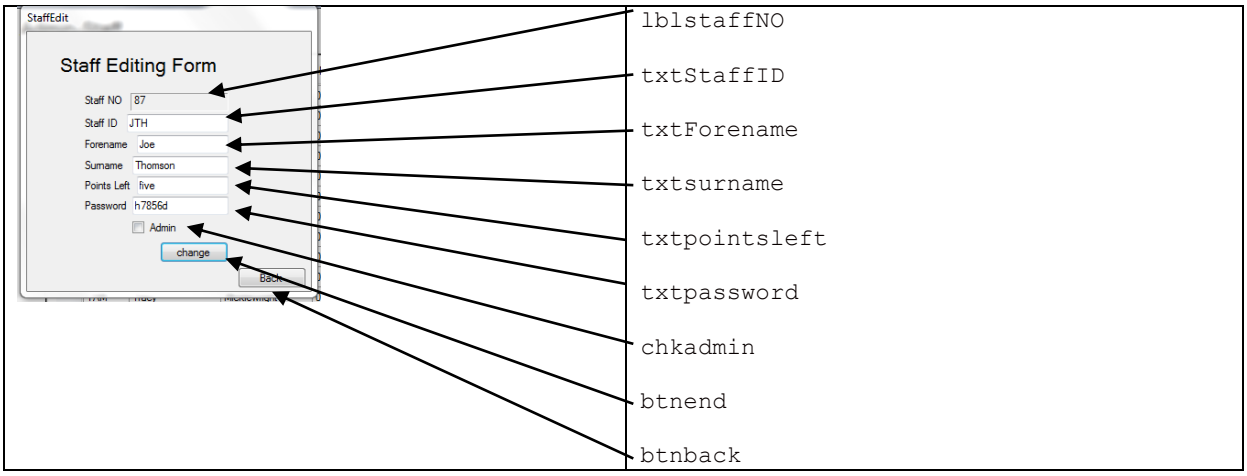
RewardsEdit

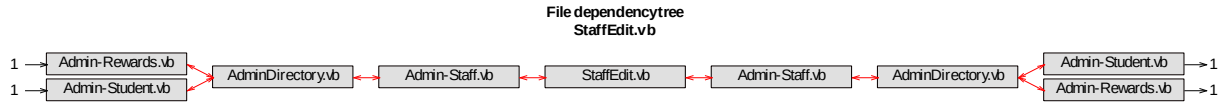


File dependencytree
RewardsEdit.vb


```
Private Sub btnBack_Click
   Opens admin_rewards and close rewardsEdit
Private Sub btnEnd_Click
   Validates the text boxes; makes sure a cost is input, it is a number and makes sure
   a name is input. The data is then put into the reward structure which is in tern put
   into the rewards dat file. The data grid on admin_rewards is then updated and the
   rewardsEdit form is closed
Private Sub RewardsEdit_Load
   First determines whether it is in adding mode or editing mode by checking the adding
   Boolean. If adding then a new reward ID is put in to the lblRewardID and the text
   boxes are left empty and the text for btnEnd reads "add record". If adding is false
   then txtcost contains the cost of the reward selected, txtname contains the name of
   the reward selected, lblrewardID contains the reward selecteds ID and btnEnd's text
   is "Change"
```

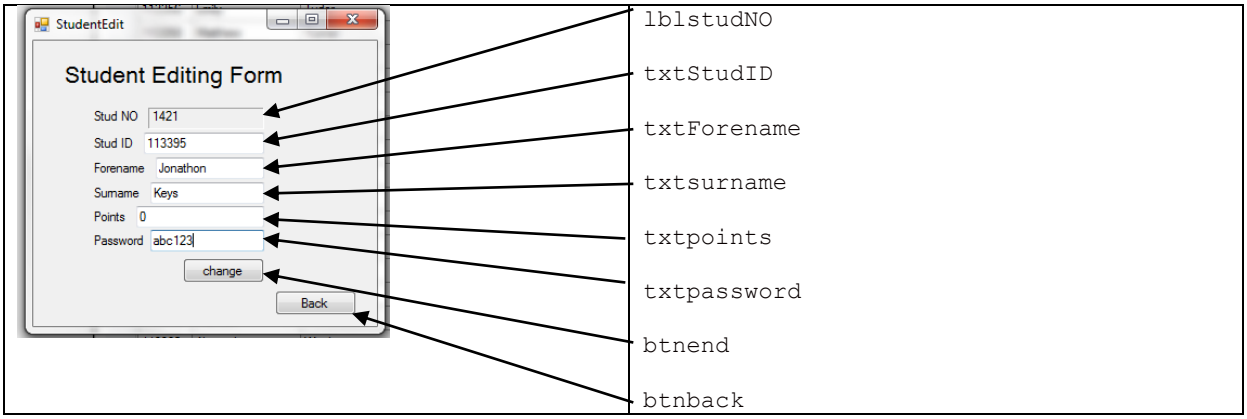
StaffEdit

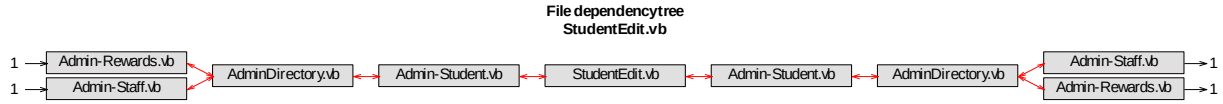





```
Private Sub btnBack_Click
   Opens admin_staff and closes staffEdit
Private Sub btnEnd_Click
   Validates the text boxes; makes sure a ID is input, it is made of letters, 3 or less
   characters in length, that txtforename is input and that it is made of letters, the
   same for txtsurname, that txtpassword is greater then 5 characters and txtpointsleft
   is a number. The data is then put into the staff structure which is in tern put into
   the staff dat file. The data grid on admin_staff is then updated and the staffEdit
   form is closed.
Private Sub StaffEdit_Load
   First determines whether it is in adding mode or editing mode by checking the adding
   Boolean. If adding then a new staffNO is put in to the lblStaffNO and the text boxes
   are left empty and the text for btnEnd reads "add record". If adding is false then
   txtforename contains the forename of the staff selected, txtsurname contains the
   surname of the staff selected, lblstaffNO contains the staffNO of the staff selected
   and so on. also btnEnd's text is "Change"
```

StudentEdit






```
Private Sub btnBack_Click
   Opens admin_student and closes studentEdit
Private Sub btnEnd_Click
   Validates the text boxes; makes sure a ID is input, it is made of numbers, 6 or less
   characters in length, that txtforename is input and that it is made of letters, the
   same for txtsurname, that txtpassword is greater then 5 characters and txtpoints is
   a number. The data is then put into the student structure which is in tern put into
   the student dat file. The data grid on admin_student is then updated and the
   studentEdit form is closed.
Private Sub StudentEdit_Load
   First determines whether it is in adding mode or editing mode by checking the adding
   Boolean. If adding then a new staffNO is put in to the lblStaffNO and the text boxes
   are left empty and the text for btnEnd reads "add record". If adding is false then
   txtforename contains the forename of the student selected, txtsurname contains the
   surname of the student selected, lblstudNO contains the studNO of the student
   selected and so on. also btnEnd's text is "Change"
```

User interface

I believe that my user interface is suitable due to the fact that all of the colours are muted and kept clean and simple; they are also constant throughout the program. All of the text boxes and buttons are labelled in an uncomplicated manner which explains easily and quickly. Also the layout of the forms is equally simple with everything in a generic format so the user does not have to search the forms for buttons they want and the orders of text boxes in the procedures is very flowing.

Variable list

Local Variables

```
Dim onrec as interger = 0
   Loginfrom
Is used in the do loops as a counter that holds how many times the loop been done and
is used as the file number to check in the loops.

For Counter as interger = 1 to Npoints
   Staffform
Is used as a counter to store which loop the for loop is on so as check the file of
that loop

Dim Counter as interger
   Staffform
Is used as a counter to store which loop the for loop is on so as check the file of
that loop

Dim ID as integer
   Staffform
Is used to store the content of cell 0 of the row clicked

Dim Counter as interger
```

Staffform
Is used as a counter to store which loop the for loop is on so as check the file of that loop

For Counter as interger = 1 to Nrewards
Studentform1
Is used as a counter to store which loop the for loop is on so as check the file of that loop

Dim onreward = getreward(counter)
Studnetform1
Is used to store the reward that has been extracted from the rewards dat file

Dim rewardselected as integer
Studentform1
Is used to store the ID of the reward selected

For Counter as interger = 1 to NpointsP
Studentrecordsform
Is used as a counter to store which loop the for loop is on so as check the file of that loop

Dim OnPointsPRec = getPointsP(counter)
Studentrecordsform
Is used to store the pointsprocedure that has been extracted from the pointsprocedure dat file

For Counter as interger = 1 to Nstaff
Admin-rewards
Is used as a counter to store which loop the for loop is on so as check the file of that loop

Dim Counter as interger
Admin-rewards
Is used as a counter to store which loop the for loop is on so as check the file of that loop

Dim ID as integer
Admin-rewards
Is used to store the content of cell 0 of the row clicked

For Counter as interger = 1 to Nstaff
Admin-staff
Is used as a counter to store which loop the for loop is on so as check the file of that loop

Dim Counter as interger
Admin-staff
Is used as a counter to store which loop the for loop is on so as check the file of that loop

Dim ID as integer
Admin-staff
Is used to store the content of cell 0 of the row clicked

For Counter as interger = 1 to Nstudents
Admin-student
Is used as a counter to store which loop the for loop is on so as check the file of that loop

Dim Counter as interger
Admin-students

```
Is used as a counter to store which loop the for loop is on so as check the file of
that loop

Dim ID as integer
   Admin-students
Is used to store the content of cell 0 of the row clicked

For Counter as interger = 1 to Nrewards
   rewardsEdit
Is used as a counter to store which loop the for loop is on so as check the file of
that loop

For Counter as interger = 1 to Nstaff
   staffEdit
Is used as a counter to store which loop the for loop is on so as check the file of
that loop

For Counter as interger = 1 to Nstudents
   studentsEdit
Is used as a counter to store which loop the for loop is on so as check the file of
that loop
```

Global variables

```
Public Adding As Boolean = True
  modVariables
adding was used as a variable to determine how the edit forms were going to open, if
true then the form is in adding mode, if false the form opens in editing mode.

Public NpointsP As Integer = 0
  modVariables

stores the number of records in the pointsprocedure dat file

Public NRewards As Integer = 0
  modVariables

stores the number of records in the rewards dat file

Public NrewardsP As Integer = 0
  modVariables

stores the number of records in the rewardsprocedure dat file

Public Nstaff As Integer = 0
  modVariables

stores the number of records in the staff dat file

Public NStudents As Integer = 0
  modVariables

stores the number of records in the students dat file

Public pointsP As pointsprocedure = Nothing
  modVariables

stores data in the shape of the pointsprocdure structure

Public reward As Rewards = Nothing
```

modVariables

stores data in the shape of the rewards structure

```
Public RewardsP As Rewardsprocedure = Nothing
```
modVariables

stores data in the shape of the rewardsprocdure structure

```
Public staff As Staffrec = Nothing
```
modVariables

stores data in the shape of the staff structure

```
Public student As studentrec = Nothing
```
modVariables

stores data in the shape of the student structure

Module-level variables

modFunctions

```
Dim DES As New System.Security.Cryptography.TripleDESCryptoServiceProvider
   Used in code from the Microsoft website
Dim Hash As New System.Security.Cryptography.MD5CryptoServiceProvider
   Used in code from the Microsoft website
```

modVariables.pointsprocedure

variables used to store pointsprocedure records that are currently being used and is used as the structure for retrieving and putting records into the pointsprocedure dat file

```
Public pointstransfered As Short
   Is used to store the amount of points used in the points transfered
Public reason As String
   Is used to store the reason for a points transfer
Public StaffNO As Short
   Is used to store the staff making the transfer's staffNO
Public StudNO As Integer
   Is used to store the student in the transfer's studNO
```

modVariables.Rewards

variables used to store rewards records that are currently being used and is used as the structure for retrieving and putting records into the rewards dat file

```
Public cost As Short
   Is used to store the cost of the reward
Public name As String
   Is used to store the name of the reward
Public RewardID As Byte
   Is used to store the id of the reward
```

modVariables.Rewardsprocedure

variables used to store rewardsprocedure records that are currently being used and is used as the structure for retrieving and putting records into the rewardsprocedure dat file

```
Public RewardID As Byte
   Is used to store the id of the reward bought
Public studNO As Short
   Is used to store the studNO of the student buying the reward
```

modVariables.Staffrec

variables used to store staff records that are currently being used and is used as the structure for retrieving and putting records into the staff dat file

```
Public admin As Boolean
   Used to store as a Boolean where the member of staff is an admin, true means they
   are false means they are not
Public DOB As Date
   Used to store the DOB of the member of staff, only used when dat file is created to
   manufacture the staff members password
Public Forename As String
   Used to store the forename of the member of staff
Public password As String
   Used to store an encrypted version of the staff members password
Public PointsLeft As Short
   Used to store the points that the staff member has left
Public staffID As String
   Used to store ID of the staff member which also acts as their username
Public staffNO As Short
   Used to store the place number of the staff member in the staff dat file
Public Surname As String
   Used to store surname of the staff member
```

modVariables.studentrec

variables used to store student records that are currently being used and is used as the structure for retrieving and putting records into the student dat file

```
Public DOB As Date
   Used to store the DOB of the student, only used when dat file is created to
   manufacture the student's password
Public Forename As String
   Used to store the forename of the member of staff
Public password As String
   Used to store an encrypted version of the students password
Public points As Short
   Used to store the points of the student
Public studID As Integer
   Used to store the ID of the student
Public studNO As Short
   Used to store the place number of the student in the student dat file
Public Surname As String
   Used to store the username of the member of staff
```

```
00001  Public Class AdminDirectory
00002
00003      Private Sub btnBack_Click(sender As System.Object, e As System.EventArgs)
           Handles btnBack.Click
00004          'opens staffform and closes admindirectory
00005          staffform.Show()
00006          Me.Close()
00007      End Sub
00008
00009      Private Sub btnAdminStaff_Click(sender As System.Object, e As System.EventArgs)
           Handles btnAdminStaff.Click
00010          'opens admin_staff and hides adminDirectory
00011          Admin_Staff.Show()
00012          Me.Hide()
00013      End Sub
00014
00015      Private Sub btnAdminStudents_Click(ByVal sender As System.Object, ByVal e As
           System.EventArgs) Handles btnAdminStudents.Click
00016          'opens admin_students and hides adminDirectory
00017          Admin_Student.Show()
00018          Me.Hide()
00019      End Sub
00020
00021      Private Sub btnAdminRewards_Click(ByVal sender As System.Object, ByVal e As
           System.EventArgs) Handles btnAdminRewards.Click
00022          'opens admin_rewards and hides adminDirectory
00023          Admin_Rewards.Show()
00024          Me.Hide()
00025      End Sub
00026  End Class
```

```
00001  Public Class Admin_Rewards
00002
00003      Private Sub Admin_Rewards_Load(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles MyBase.Load
00004          'populate grdRewards
00005          For counter As Integer = 1 To NRewards
00006              reward = GetReward(counter)
00007              grdRewards.Rows.Add(reward.RewardID, reward.name, reward.cost)
00008          Next
00009      End Sub
00010
00011      Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnBack.Click
00012          'show admindirectory and close adminrewards
00013          AdminDirectory.Show()
00014          Me.Close()
00015      End Sub
00016
00017      Private Sub grdRewards_CellContentdoubleClick(ByVal sender As System.Object,
           ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles grdRewards.
           CellContentClick
00018          Dim ID As Integer
00019          'intialise variable but if it doesnt work because content sellected is a
                header it will exit sub
00020          Try
00021              ID = grdRewards.Rows(e.RowIndex).Cells(0).Value
00022          Catch
00023              Exit Sub
00024          End Try
00025          'get reward
00026          reward = GetReward(ID)
00027          'set global variable adding to false to tell the rewardsEdit form to open in
                change mode
00028          Adding = False
00029          'open the rewardsEdit form
00030          RewardsEdit.Show()
00031
00032      End Sub
00033
00034      Private Sub btnAddReward_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnAddReward.Click
00035          'set global variable adding to true to tell the rewardsEdit form to open in
                add mode
00036          Adding = True
00037          'open the rewardsedit form
00038          RewardsEdit.Show()
00039      End Sub
00040  End Class
```

```
00001  Public Class Admin_Staff
00002
00003      Private Sub Admin_Staff_Load(sender As System.Object, e As System.EventArgs)
           Handles MyBase.Load
00004          'populate the grdstaff data grid
00005          For counter As Integer = 1 To Nstaff
00006              staff = GetStaff(counter)
00007              grdStaff.Rows.Add(staff.staffNO, staff.staffID, staff.Forename, staff.
                   Surname, staff.PointsLeft, staff.admin, Decrypt(staff.password))
00008          Next
00009      End Sub
00010
00011      Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnSearch.Click
00012          Dim counter As Integer
00013          'checks if there is a word in txtforename, serves as presence check and word
                check
00014          If IsAlpha(txtForname.Text.Trim) = True Then
00015              'checks if there is a word in txtsurname, serves as presence check and
                    word check
00016              If IsAlpha(txtSurname.Text.Trim) = True Then
00017                  'clears grdstaff populates grdstaff with only staff whose first name
                        starts with the
00018                  'text in txtforename and whose last name starts with the text in
                        txtsurname
00019                  grdStaff.Rows.Clear()
00020                  For counter = 1 To Nstaff
00021                      staff = GetStaff(counter)
00022                      If staff.Forename.ToUpper.StartsWith(txtForname.Text.Trim.
                           ToUpper) And staff.Surname.ToUpper.StartsWith(txtSurname.Text.
                           ToUpper.Trim) Then
00023                          grdStaff.Rows.Add(staff.staffNO, staff.staffID, staff.
                               Forename, staff.Surname, staff.PointsLeft, staff.admin,
                               Decrypt(staff.password))
00024                      End If
00025                  Next
00026
00027                  'if only the forname box contains letters and txtsurname is empty
00028              ElseIf txtSurname.Text = ""  Then
00029                  'clears grdstaff then populates the grdstaff with only staff whose
                        first name
00030                  'starts with the text in txtforename
00031                  grdStaff.Rows.Clear()
00032                  For counter = 1 To Nstaff
00033                      staff = GetStaff(counter)
00034                      If staff.Forename.ToUpper.StartsWith(txtForname.Text.Trim.
                           ToUpper) Then
00035                          grdStaff.Rows.Add(staff.staffNO, staff.staffID, staff.
                               Forename, staff.Surname, staff.PointsLeft, staff.admin,
                               Decrypt(staff.password))
00036                      End If
00037                  Next
00038                  'txtsurname contains invalid data so error message sent
00039              Else :MsgBox("the surname must be only letters with no spaces" )
00040              End If
00041          ElseIf txtForname.Text = ""  Then
00042              'checks if there is a word in txtsurname, serves as presence check and
                    word check
00043              If IsAlpha(txtSurname.Text) = True Then
00044                  'clears grdstaff then populates the grdstaff with only staff whose
                        second name
```

```
00045                'starts with the text in txtsurname
00046                grdStaff.Rows.Clear()
00047                For counter = 1 To Nstaff
00048                    staff = GetStaff(counter)
00049                    If staff.Surname.ToUpper.StartsWith(txtSurname.Text.Trim.ToUpper
                          ) Then
00050                        grdStaff.Rows.Add(staff.staffNO, staff.staffID, staff.
                             Forename, staff.Surname, staff.PointsLeft, staff.admin,
                             Decrypt(staff.password))
00051                    End If
00052                Next
00053                'if neither box contains letters
00054            ElseIf txtSurname.Text = ""  Then
00055                'clears grdstaff then populates it with all staff
00056                grdStaff.Rows.Clear()
00057                For counter = 1 To Nstaff
00058                    staff = GetStaff(counter)
00059                    grdStaff.Rows.Add(staff.staffNO, staff.staffID, staff.Forename,
                          staff.Surname, staff.PointsLeft, staff.admin, Decrypt(staff.
                          password))
00060                Next
00061                'txtsurname contains invalid data so error message sent
00062            Else :MsgBox("surname must be only letters with no spaces" )
00063            End If
00064            'txtforname contains invalid data so error message sent
00065        Else :MsgBox("the forename must be only letters with no spaces" )
00066        End If
00067    End Sub
00068
00069    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
          EventArgs) Handles btnBack.Click
00070        'opens admindirectory and close admin staff
00071        AdminDirectory.Show()
00072        Me.Close()
00073    End Sub
00074
00075    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.
          EventArgs) Handles btnAdd.Click
00076        'sets global variable adding to true so that staffedit will open in adding
              mode then opens staffedit
00077        Adding = True
00078        StaffEdit.Show()
00079    End Sub
00080
00081    Private Sub grdStaff_CellContentdoubleClick(ByVal sender As System.Object, ByVal
          e As System.Windows.Forms.DataGridViewCellEventArgs) Handles grdStaff.
          CellContentClick
00082        Dim ID As Integer
00083        'intialise variable but if it doesnt work because content sellected is a
              header it will exit sub
00084        Try
00085            ID = grdStaff.Rows(e.RowIndex).Cells(0).Value
00086        Catch
00087            Exit Sub
00088        End Try
00089        'gets staff data
00090        staff = GetStaff(ID)
00091        'sets global variable adding to false so that staffedit will open in change
              mode then opens staffedit
00092        Adding = False
00093        StaffEdit.Show()
```

```
00094        End Sub
00095    End Class
```

```
00001  Public Class Admin_Student
00002
00003      Private Sub Admin_Student_Load(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles MyBase.Load
00004          'populates grdstudent
00005          For counter As Integer = 1 To NStudents
00006              student = GetStudent(counter)
00007              grdStudents.Rows.Add(student.studID, student.Forename.Trim, student.
                   Surname.Trim, student.points, Decrypt(student.password))
00008          Next
00009      End Sub
00010
00011      Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnSearch.Click
00012          Dim counter As Integer
00013          'checks if there is a word in txtforename, serves as presence check and word
                check
00014          If IsAlpha(txtForname.Text.Trim) = True Then
00015              'checks if there is a word in txtsurname, serves as presence check and
                    word check
00016              If IsAlpha(txtSurname.Text.Trim) = True Then
00017                  'clears grdstudnets populates the grdstudents with only students
                        whose first name starts with the
00018                  'text in txtforename and whose last name starts with the text in
                        txtsurname
00019                  grdStudents.Rows.Clear()
00020                  For counter = 1 To NStudents
00021                      student = GetStudent(counter)
00022                      If student.Forename.ToUpper.StartsWith(txtForname.Text.Trim.
                           ToUpper) And student.Surname.ToUpper.StartsWith(txtSurname.Text
                           .ToUpper.Trim) Then
00023                          grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                               student.Surname.Trim, student.points, Decrypt(student.
                               password))
00024                      End If
00025                  Next
00026
00027                  'if only the forname box contains letters and txtsurname is empty
00028              ElseIf txtSurname.Text = ""  Then
00029                  'clears grdstudents then populates the grdstudents with only
                        students whose first name
00030                  'starts with the text in txtforename
00031                  grdStudents.Rows.Clear()
00032                  For counter = 1 To NStudents
00033                      student = GetStudent(counter)
00034                      If student.Forename.ToUpper.StartsWith(txtForname.Text.Trim.
                           ToUpper) Then
00035                          grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                               student.Surname.Trim, student.points, Decrypt(student.
                               password))
00036                      End If
00037                  Next
00038                  'txtstudent contains invalid data so error message sent
00039              Else :MsgBox("the surname must be only letters with no spaces" )
00040              End If
00041          ElseIf txtForname.Text = ""  Then
00042              'checks if there is a word in txtsurname, serves as presence check and
                    word check
00043              If IsAlpha(txtSurname.Text) = True Then
00044                  'clears grdstudents then populates the grdstudents with only
                        students whose second name
```

```
00045                'starts with the text in txtsurname
00046                grdStudents.Rows.Clear()
00047                For counter = 1 To NStudents
00048                    student = GetStudent(counter)
00049                    If student.Surname.ToUpper.StartsWith(txtSurname.Text.Trim.
                          ToUpper) Then
00050                        grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                              student.Surname.Trim, student.points, Decrypt(student.
                              password))
00051                    End If
00052                Next
00053                'if neither box contains letters
00054            ElseIf txtSurname.Text = ""  Then
00055                'clears grdstudents then populates it with all students
00056                grdStudents.Rows.Clear()
00057                For counter = 1 To NStudents
00058                    student = GetStudent(counter)
00059                    grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                          student.Surname.Trim, student.points, Decrypt(student.password)
                          )
00060                Next
00061                'txtsurname contains invalid data so error message sent
00062            Else :MsgBox("surname must be only letters with no spaces" )
00063            End If
00064            'txtforname contains invalid data so error message sent
00065        Else :MsgBox("the forename must be only letters with no spaces" )
00066        End If
00067    End Sub
00068
00069    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
          EventArgs) Handles btnBack.Click
00070        'show admin directory and close adminStudent form
00071        AdminDirectory.Show()
00072        Me.Close()
00073    End Sub
00074
00075    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.
          EventArgs) Handles btnAdd.Click
00076        'sets global variable adding to true so that studentedit will open in adding
              mode then opens studentedit
00077        Adding = True
00078        StudentEdit.Show()
00079
00080    End Sub
00081
00082    Private Sub grdStudents_CellContentdoubleClick(ByVal sender As System.Object,
          ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles grdStudents.
          CellContentClick
00083        Dim ID As Integer
00084        'intialise variable but if it doesnt work because content sellected is a
              header it will exit sub
00085        Try
00086            ID = grdStudents.Rows(e.RowIndex).Cells(0).Value
00087        Catch ex As Exception
00088            Exit Sub
00089        End Try
00090        'compares the studID of each student against the one in the row that was
              clicked on till a match is found.
00091        For counter As Integer = 1 To NStudents
00092            student = GetStudent(counter)
00093            If student.studID = ID Then
```

```
00094                     'match has been found so global variable adding set to false so that
                           studentedit will open in change
00095                     'mode then opens studentedit
00096                     Adding = False
00097                     StudentEdit.Show()
00098                     Exit Sub
00099                 End If
00100             Next
00101         End Sub
00102     End Class
```

```
00001  Public Class loginFrom
00002
00003      Private Sub btnLogin_Click(sender As System.Object, e As System.EventArgs)
           Handles btnLogin.Click
00004          'student part from class staff part based on it.
00005          Dim onrec As Integer = 0
00006          'checks if a username was inputed if not then ends sub and sends error
               message
00007          If txtUsername.Text.Trim = ""  Then
00008              MsgBox("error please enter username" )
00009              Exit Sub
00010              'checks if a password was inputed if not then ends sub and sends error
                   message
00011          ElseIf txtPassword.Text.Trim = ""  Then
00012              MsgBox("error please enter password" )
00013              Exit Sub
00014              'checks if the username is numeric if so then the username is form as
                   student if not its a member of staff
00015          ElseIf IsNumeric(txtUsername.Text.Trim) = True Then
00016              'username was numeric so checks the username against that of all the
                   students
00017              Try
00018                  Do
00019                      onrec += 1
00020                      student = GetStudent(onrec)
00021                      If Val(txtUsername.Text) = student.studID Then
00022                          'found student that matches the username so checks if the
                               password matches
00023                          If txtPassword.Text.Trim = Decrypt(student.password.Trim)
                               Then
00024                              'password was correct so student form opens and logon
                                   form opens
00025                              StudentForm1.Show()
00026                              Me.Close()
00027                              Exit Sub
00028                          Else
00029                              'password did not match so error message sent and sub
                                   ended
00030                              MsgBox("invalid password" )
00031                              Exit Sub
00032                          End If
00033                      End If
00034
00035                  Loop
00036              Catch ex As Exception
00037                  'nothing found so error message sent and sub ended
00038                  MsgBox("invalid username" )
00039              End Try
00040          ElseIf IsAlpha(txtUsername.Text) = True Then
00041              'username was a word so it is compared againts that of all the staff
00042              Try
00043                  Do
00044                      onrec += 1
00045                      staff = GetStaff(onrec)
00046                      If txtUsername.Text.Trim.ToUpper = staff.staffID.Trim.
                           ToUpper Then
00047                          'found member of staff so checking to see if the
                               passwords match
00048                          If txtPassword.Text.Trim = Decrypt(staff.password.Trim)
                               Then
00049                              'staff's password matches so staffform opens and
```

```
                                logonform closes
00050                           staffform.Show()
00051                           Me.Hide()
00052                           Exit Sub
00053                       Else
00054                           'passwords did not match so error message sent and
                                 sub ended
00055                           MsgBox("invalid password" )
00056                           Exit Sub
00057                       End If
00058                   End If
00059
00060                   Loop
00061               Catch ex As Exception
00062                   ' matching logon not found so error message sent and sub
                         ended
00063                   MsgBox("invalid username" )
00064               End Try
00065           Else
00066               'nothing found so error message sent and sub ended
00067               MsgBox("invalid username" )
00068           End If
00069       End Sub
00070
00071       Private Sub loginFrom_Load(ByVal sender As System.Object, ByVal e As
            System.EventArgs) Handles MyBase.Load
00072           'from class
00073
00074           'checks if the dat files for the staff, students and rewards has
                been created if not then they are
00075           If My.Computer.FileSystem.FileExists("Student.dat" ) = False Then
                Call ImportStudents()
00076           If My.Computer.FileSystem.FileExists("staff.dat" ) = False Then Call
                ImportStaff()
00077           If My.Computer.FileSystem.FileExists("rewards.dat" ) = False Then
                Call ImportRewards()
00078
00079           'intialises global variables for numbers of data structures
00080           NRewards = FileLen("Rewards.dat" ) / Len(reward)
00081           Nstaff = FileLen("staff.dat" ) / Len(staff)
00082           NStudents = FileLen("student.dat" ) / Len(student)
00083           NrewardsP = FileLen("Rewardsprocedure.dat" ) / Len(RewardsP)
00084           NpointsP = FileLen("pointsprocedure.dat" ) / Len(pointsP)
00085
00086       End Sub
00087
00088       Private Sub btnExit_Click(sender As System.Object, e As System.EventArgs
            ) Handles btnExit.Click
00089           'close logonform closing the program
00090           Me.Close()
00091       End Sub
00092
00093
00094   End Class
```

```
00001  Module modFunctions
00002      'microsofts code for encryption and decryption
00003      Dim DES As New System.Security.Cryptography.TripleDESCryptoServiceProvider
00004      Dim Hash As New System.Security.Cryptography.MD5CryptoServiceProvider
00005      Private Const Key = "Some key that you think appropriate!"
00006
00007      Public Function Encrypt(ByVal strText As String) As String
00008          'microsofts code that encrypts strings
00009          Encrypt = ""
00010          Try
00011              DES.Key = Hash.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(Key)
                   )
00012              DES.Mode = Security.Cryptography.CipherMode.ECB
00013              Dim DESEncrypter As System.Security.Cryptography.ICryptoTransform = DES.
                   CreateEncryptor
00014              Dim Buffer As Byte() = System.Text.ASCIIEncoding.ASCII.GetBytes(strText)
00015              Encrypt = Convert.ToBase64String(DESEncrypter.TransformFinalBlock(Buffer
                   , 0, Buffer.Length))
00016          Catch ex As Exception
00017              Encrypt = strText
00018          End Try
00019      End Function
00020
00021      Public Function Decrypt(ByVal strText As String) As String
00022          'microsofts code that decrypts strings
00023          Decrypt = ""
00024          Try
00025              DES.Key = Hash.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(Key)
                   )
00026              DES.Mode = Security.Cryptography.CipherMode.ECB
00027              Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform = DES.
                   CreateDecryptor
00028              Dim Buffer As Byte() = Convert.FromBase64String(strText)
00029              Decrypt = System.Text.ASCIIEncoding.ASCII.GetString(DESDecrypter.
                   TransformFinalBlock(Buffer, 0, Buffer.Length))
00030          Catch ex As Exception
00031              Decrypt = strText
00032          End Try
00033      End Function
00034
00035      Public Function IsName(ByVal InString As String) As Boolean
00036          Dim Counter As Integer
00037          Dim strLetters As Char()
00038          'removes spaces
00039          InString = Trim(InString)
00040          'checks if empty
00041          If InString.Length = 0 Then
00042              IsName = False
00043          Else
00044              strLetters = InString.ToUpper.ToCharArray
00045              IsName = True
00046              For Counter = 0 To InString.Length - 1
00047                  'checks each chacater for being a letter or a hyfin
00048                  If IsAlpha(strLetters(Counter)) = True Or strLetters(Counter) = "-"
                       Then
00049                      'OK
00050                  Else
00051                      'Error
00052                      IsName = False
00053                  End If
00054              Next
```

```
00055         End If
00056     End Function
00057
00058     Public Function IsAlpha(ByVal InString As String) As Boolean
00059         Dim Counter As Integer
00060         Dim strLetters As Char()
00061         InString = Trim(InString)
00062         'checks if empty
00063         If InString.Length = 0 Then
00064             IsAlpha = False
00065         Else
00066             strLetters = InString.ToUpper.ToCharArray
00067             IsAlpha = True
00068             'checks each character for being a letter if not isalpha is returned as
                  false
00069             For Counter = 0 To InString.Length - 1
00070                 If strLetters(Counter) < "A"  Or strLetters(Counter) > "Z"  Then
00071                     IsAlpha = False
00072                 End If
00073             Next
00074         End If
00075     End Function
00076
00077     Public Sub ImportStudents()
00078         'opnes up file reader and sets it to read collegedata.csv the file in which
                  the student data is stored
00079         Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
                  "collegedata.csv" )
00080         TextFileReader.TextFieldType = FileIO.FieldType.Delimited
00081         TextFileReader.SetDelimiters("," )
00082
00083         Dim CurrentRow As String()
00084         Dim OnRec As Integer = 0
00085         Dim FileNum As Integer = FreeFile()
00086         'opens file
00087         FileOpen(FileNum, "Student.Dat" , OpenMode.Random, OpenAccess.Default,
                  OpenShare.Default, Len(student))
00088
00089         While Not TextFileReader.EndOfData
00090             Try
00091                 CurrentRow = TextFileReader.ReadFields()
00092                 If Not CurrentRow Is Nothing Then
00093                     OnRec = OnRec + 1
00094                     'puts data into the studet structure
00095                     With student
00096                         .studNO = CurrentRow(0)
00097                         .studID = CurrentRow(1)
00098                         .Surname = CurrentRow(2)
00099                         .Forename = CurrentRow(3)
00100                         .DOB = CurrentRow(4)
00101                         .password = Encrypt("db"  & Format(.DOB, "ddMMyyyy" ))
00102                     End With
00103                     'puts data from the student structure into the student dat file
00104                     FilePut(FileNum, student, OnRec)
00105                 End If
00106             Catch ex As  _
00107                     Microsoft.VisualBasic.FileIO.MalformedLineException
00108                 'if error then error message is sent and try ends
00109                 MsgBox("Line "  & ex.Message & "is not valid and will be skipped." )
00110             End Try
00111         End While
```

```
00112         'message box is sent saying that the students are imported and how many
00113         MsgBox(NStudents & " students imported" )
00114         'file is closed
00115         FileClose(FileNum)
00116         TextFileReader.Dispose()
00117     End Sub
00118 
00119     Public Sub ImportStaff()
00120         'opens microsoft file reader and sets the file to be read as staffdata.csv
00121         Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
              "staffdata.csv" )
00122         TextFileReader.TextFieldType = FileIO.FieldType.Delimited
00123         TextFileReader.SetDelimiters("," )
00124 
00125         Dim CurrentRow As String()
00126         Dim OnRec As Integer = 0
00127         Dim FileNum As Integer = FreeFile()
00128         'opens the file
00129         FileOpen(FileNum, "Staff.Dat" , OpenMode.Random, OpenAccess.Default,
              OpenShare.Default, Len(staff))
00130 
00131         While Not TextFileReader.EndOfData
00132             Try
00133                 CurrentRow = TextFileReader.ReadFields()
00134                 If Not CurrentRow Is Nothing Then
00135                     OnRec = OnRec + 1
00136                     'puts data into file structure staff
00137                     With staff
00138                         .staffNO = CurrentRow(0)
00139                         .admin = CurrentRow(1)
00140                         .staffID = CurrentRow(2)
00141                         .Forename = CurrentRow(3)
00142                         .Surname = CurrentRow(4)
00143                         .DOB = CurrentRow(5)
00144                         .password = Encrypt("db"  & Format(.DOB, "ddMMyyyy" ))
00145                     End With
00146                     'puts data in file structure staff into the staff dat file
00147                     FilePut(FileNum, staff, OnRec)
00148                 End If
00149             Catch ex As  _
00150                 Microsoft.VisualBasic.FileIO.MalformedLineException
00151                 'error in text sends error message and ends try
00152                 MsgBox("Line "  & ex.Message & "is not valid and will be skipped." )
00153             End Try
00154         End While
00155         Nstaff = OnRec
00156         'sends message box notifying student that staff have been imported and how
              many have been
00157         MsgBox(Nstaff & " Staff imported" )
00158         FileClose(FileNum)
00159         TextFileReader.Dispose()
00160     End Sub
00161 
00162     Public Sub ImportRewards()
00163         'opens microsoft file reader and sets the file to be read as rewards.csv
00164         Dim TextFileReader As New Microsoft.VisualBasic.FileIO.TextFieldParser(
              "rewards.csv" )
00165         TextFileReader.TextFieldType = FileIO.FieldType.Delimited
00166         TextFileReader.SetDelimiters("," )
00167 
00168         Dim CurrentRow As String()
```

```
00169         Dim OnRec As Integer = 0
00170         Dim FileNum As Integer = FreeFile()
00171         'opens the file
00172         FileOpen(FileNum, "rewards.Dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(reward))
00173
00174         While Not TextFileReader.EndOfData
00175             Try
00176                 CurrentRow = TextFileReader.ReadFields()
00177                 If Not CurrentRow Is Nothing Then
00178                     OnRec = OnRec + 1
00179                     With reward
00180                         'puts data into file structure rewards
00181                         .RewardID = CurrentRow(0)
00182                         .name = CurrentRow(1)
00183                         .cost = CurrentRow(2)
00184                     End With
00185                     'puts data from file structure rewards into the rewards dat file
00186                     FilePut(FileNum, reward, OnRec)
00187                 End If
00188
00189             Catch ex As Microsoft.VisualBasic.FileIO.MalformedLineException
00190                 'error found so error message sent and try ended
00191                 MsgBox("Line "  & ex.Message & "is not valid and will be skipped." )
00192             End Try
00193         End While
00194         NRewards = OnRec
00195         'import completed so message notafying user sent also saying how many
               imported
00196         MsgBox(NRewards & " rewards imported" )
00197         FileClose(FileNum)
00198         TextFileReader.Dispose()
00199     End Sub
00200     Public Function GetStudent(ByVal RecNo As Integer) As studentrec
00201         'function for getting data from student dat file
00202         Dim Filenum As Integer = FreeFile()
00203         GetStudent = Nothing
00204         'opens the student dat file
00205         FileOpen(Filenum, "Student.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(Student))
00206         'gets data
00207         FileGet(Filenum, GetStudent, RecNo)
00208         'closes file
00209         FileClose(Filenum)
00210     End Function
00211
00212     Public Function GetStaff(ByVal RecNo As Integer) As Staffrec
00213         'function for getting data from staff dat file
00214         Dim Filenum As Integer = FreeFile()
00215         GetStaff = Nothing
00216         'opens the staff dat file
00217         FileOpen(Filenum, "Staff.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(staff))
00218         'gets data
00219         FileGet(Filenum, GetStaff, RecNo)
00220         'closes file
00221         FileClose(Filenum)
00222     End Function
00223
00224     Public Function GetReward(ByVal RecNo As Integer) As Rewards
00225         'function for getting data from reward dat file
```

```
00226         Dim Filenum As Integer = FreeFile()
00227         GetReward = Nothing
00228         'opens the reward dat file
00229         FileOpen(Filenum, "rewards.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(reward))
00230         'gets data
00231         FileGet(Filenum, GetReward, RecNo)
00232         'closes file
00233         FileClose(Filenum)
00234     End Function
00235 
00236     Public Function GetRewardP(ByVal RecNo As Integer) As Rewardsprocedure
00237         'function for getting data from the rewardsP dat file
00238         Dim Filenum As Integer = FreeFile()
00239         GetRewardP = Nothing
00240         'opens the rewardsP dat file
00241         FileOpen(Filenum, "rewardsP.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(RewardsP))
00242         'gets data
00243         FileGet(Filenum, GetRewardP, RecNo)
00244         'closes file
00245         FileClose(Filenum)
00246     End Function
00247 
00248     Public Function GetPointsP(ByVal RecNo As Integer) As pointsprocedure
00249         'function for getting data from the pointsp dat file
00250         Dim Filenum As Integer = FreeFile()
00251         GetPointsP = Nothing
00252         'opens the pointsp dat file
00253         FileOpen(Filenum, "pointsprocedure.dat" , OpenMode.Random, OpenAccess.
               Default, OpenShare.Default, Len(pointsP))
00254         'gets data
00255         FileGet(Filenum, GetPointsP, RecNo)
00256         'closes file
00257         FileClose(Filenum)
00258     End Function
00259 
00260     Public Sub PutStudent(ByVal EditedStudent As studentrec, ByVal RecNo As Integer)
00261         'sub for putting data into the student dat file
00262         Dim Filenum As Integer = FreeFile()
00263         'opens student dat file
00264         FileOpen(Filenum, "Student.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(student))
00265         'puts data into student dat file
00266         FilePut(Filenum, EditedStudent, RecNo)
00267         'closes student dat file
00268         FileClose(Filenum)
00269     End Sub
00270 
00271     Public Sub PutStaff(ByVal EditedStaff As Staffrec, ByVal RecNo As Integer)
00272         'sub for puttin data into the staff dat file
00273         Dim Filenum As Integer = FreeFile()
00274         'opens staff dat file
00275         FileOpen(Filenum, "Staff.dat" , OpenMode.Random, OpenAccess.Default,
               OpenShare.Default, Len(staff))
00276         'puts data into staff dat file
00277         FilePut(Filenum, EditedStaff, RecNo)
00278         'close staff dat file
00279         FileClose(Filenum)
00280     End Sub
00281 
```

```
00282       Public Sub PutReward(ByVal EditedReward As Rewards, ByVal RecNo As Integer)
00283           'sub for putting data into teh rewards dat file
00284           Dim Filenum As Integer = FreeFile()
00285           'opens rewards dat file
00286           FileOpen(Filenum, "Rewards.dat" , OpenMode.Random, OpenAccess.Default,
                 OpenShare.Default, Len(reward))
00287           'puts data into rewards dat file
00288           FilePut(Filenum, EditedReward, RecNo)
00289           'close rewards dat file
00290           FileClose(Filenum)
00291       End Sub
00292
00293       Public Sub PutRewardP(ByVal EditedRewardP As Rewardsprocedure, ByVal RecNo As
             Integer)
00294           'sub for putting data into the rewardsp dat file
00295           Dim Filenum As Integer = FreeFile()
00296           'opens rewardsp dat file
00297           FileOpen(Filenum, "RewardsProcedure.dat" , OpenMode.Random, OpenAccess.
                 Default, OpenShare.Default, Len(RewardsP))
00298           'puts data into rewardsp dat file
00299           FilePut(Filenum, EditedRewardP, RecNo)
00300           'close rewardsp dat file
00301           FileClose(Filenum)
00302       End Sub
00303
00304       Public Sub PutPointsP(ByVal EditedPointsP As pointsprocedure, ByVal RecNo As
             Integer)
00305           'sub for putting data into the pointsp dat file
00306           Dim Filenum As Integer = FreeFile()
00307           'opens pointsp dat file
00308           FileOpen(Filenum, "pointsProcedure.dat" , OpenMode.Random, OpenAccess.
                 Default, OpenShare.Default, Len(pointsP))
00309           'puts data into pointsp dat file
00310           FilePut(Filenum, EditedPointsP, RecNo)
00311           'closes pointsp dat file
00312           FileClose(Filenum)
00313       End Sub
00314   End Module
```

```
00001  Module modVariables
00002      Public Adding As Boolean = True
00003
00004      Public Structure studentrec
00005          Public studNO As Short
00006          Public studID As Integer
00007          <VBFixedString(20)> Public Surname As String
00008          <VBFixedString(20)> Public Forename As String
00009          Public DOB As Date
00010          Public points As Short
00011          <VBFixedString(64)> Public password As String
00012      End Structure
00013      Public student As studentrec = Nothing
00014      Public NStudents As Integer = 0
00015
00016      Public Structure Staffrec
00017          Public staffNO As Short
00018          <VBFixedString(3)> Public staffID As String
00019          <VBFixedString(20)> Public Surname As String
00020          <VBFixedString(20)> Public Forename As String
00021          Public PointsLeft As Short
00022          <VBFixedString(64)> Public password As String
00023          Public DOB As Date
00024          Public admin As Boolean
00025      End Structure
00026      Public staff As Staffrec = Nothing
00027      Public Nstaff As Integer = 0
00028
00029      Public Structure Rewards
00030          Public RewardID As Byte
00031          <VBFixedString(50)> Public name As String
00032          Public cost As Short
00033      End Structure
00034      Public reward As Rewards = Nothing
00035      Public NRewards As Integer = 0
00036
00037      Public Structure pointsprocedure
00038          Public StudNO As Integer
00039          Public StaffNO As Short
00040          Public pointstransfered As Short
00041          <VBFixedString(50)> Public reason As String
00042      End Structure
00043      Public pointsP As pointsprocedure = Nothing
00044      Public NpointsP As Integer = 0
00045
00046      Public Structure Rewardsprocedure
00047          Public studNO As Short
00048          Public RewardID As Byte
00049      End Structure
00050      Public RewardsP As Rewardsprocedure = Nothing
00051      Public NrewardsP As Integer = 0
00052
00053  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("WindowsApplication1" )>
00012   <Assembly:AssemblyDescription("" )>
00013   <Assembly:AssemblyCompany("" )>
00014   <Assembly:AssemblyProduct("WindowsApplication1" )>
00015   <Assembly:AssemblyCopyright("Copyright ©  2012" )>
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("50d0828b-76bd-449c-bb4e-fa5a4b5d2ad8" )>
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
```

```
00001  Public Class RewardsEdit
00002
00003      Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnBack.Click
00004          'opens adminrewards and closes rewards edit
00005          Admin_Rewards.Show()
00006          Me.Close()
00007      End Sub
00008
00009      Private Sub RewardsEdit_Load(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles MyBase.Load
00010          If Adding = False Then
00011              'adding is false there for the form is in change mode so the data of the
                   reward selected to be changed
00012              'was put into the reward structure is now put into the text boxes and
                   the text on the btn is changed to change
00013              lblRewardID.Text = reward.RewardID
00014              txtName.Text = reward.name
00015              txtCost.Text = reward.cost
00016              btnEnd.Text = "change"
00017          Else
00018              'adding is true so the text boxes are empty and a new reward id is put
                   into lblrewardid
00019              lblRewardID.Text = NRewards + 1
00020          End If
00021      End Sub
00022
00023      Private Sub btnEnd_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnEnd.Click
00024          'checks if the cost is a number if not then an error message will be sent
                and the sub ended
00025          If IsNumeric(txtCost.Text) = False Then
00026              MsgBox("cost must be a number" )
00027              Exit Sub
00028          End If
00029          'checks if there is a name inputed if not then an error message will be sent
                and the sub ended
00030          If txtName.Text.Trim = ""  Then
00031              MsgBox("there must be a name" )
00032              Exit Sub
00033          End If
00034          'puts the data in the text boxs and lable into the reward structure
00035          reward.RewardID = lblRewardID.Text
00036          reward.cost = Int(txtCost.Text)
00037          reward.name = txtName.Text
00038          'puts the data from the reward structue into the rewards dat file
00039          PutReward(reward, reward.RewardID)
00040          NRewards += 1
00041          'updates the data grid on adminrewards
00042          If Adding = True Then
00043              Admin_Rewards.grdRewards.Rows.Add(reward.RewardID, reward.name, reward.
                   cost)
00044          Else :Admin_Rewards.grdRewards.Rows.Clear()
00045              For counter As Integer = 1 To NRewards
00046                  reward = GetReward(counter)
00047                  Admin_Rewards.grdRewards.Rows.Add(reward.RewardID, reward.name,
                       reward.cost)
00048              Next
00049          End If
00050          'closes the rewards edit form and shows the admin rewards form
00051          Admin_Rewards.Show()
```

```
00052            Me.Close()
00053
00054
00055        End Sub
00056    End Class
```

```
00001  Public Class StaffEdit
00002
00003      Private Sub StaffEdit_Load(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles MyBase.Load
00004          If Adding = True Then
00005              'adding is true so the text boxes are empty and a new staff Number is
                   put into lblstaffNo
00006              lblStaffNO.Text = Nstaff + 1
00007          Else
00008              'adding is false there for the form is in change mode so the data of the
                   staff member selected to be changed
00009              'was put into the staff structure is now put into the text boxes and the
                   text on the btn is changed to change
00010              btnEnd.Text = "change"
00011              lblStaffNO.Text = staff.staffNO
00012              txtStaffID.Text = staff.staffID
00013              txtForename.Text = staff.Forename
00014              txtSurname.Text = staff.Surname
00015              txtPointsLeft.Text = staff.PointsLeft
00016              txtPassword.Text = Decrypt(staff.password)
00017              If staff.admin = True Then
00018                  chkAdmin.Checked = True
00019              End If
00020          End If
00021      End Sub
00022
00023      Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnBack.Click
00024          'closes staffedit form and opens adminstaff
00025          Admin_Staff.Show()
00026          Me.Close()
00027      End Sub
00028
00029      Private Sub btnEnd_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnEnd.Click
00030          'checks that the length of txtstaffID is 3 or less and if not then the sub
                is ended and an error message sent
00031          If Len(txtStaffID.Text) > 3 Then
00032              MsgBox("the length of the id must be 3 or less" )
00033              Exit Sub
00034          End If
00035          'checks if the staff id is made up only of letters if not then the sub is
                ended and an error message sent
00036          If IsAlpha(txtStaffID.Text) Then
00037              'the id is made up of only letters and is now being compared against the
                   ID of other members of staff to
00038              'make sure it is unique and if it is not and it isnt due to the fact
                   that the record is being changed then
00039              'the sub is ended and an error message is sent
00040              For counter As Integer = 1 To Nstaff
00041                  staff = GetStaff(counter)
00042                  If staff.staffID = txtStaffID.Text Then
00043                      If staff.staffNO <> lblStaffNO.Text Then
00044                          MsgBox("ID already in use try another" )
00045                          Exit Sub
00046                      End If
00047                      'the id is made of letters and is unique so is being put into
                           the staff structure
00048                  Else :staff.staffID = txtStaffID.Text
00049                  End If
00050              Next
```

```
00051            Else
00052                MsgBox("the staff id must be the intials of the member of staff" )
00053            End If
00054            'the staffNo in lblstaffno is being put into the staff structure
00055            staff.staffNO = lblStaffNO.Text
00056
00057            'checks if the forname is made up of letters if not then the sub is ended
                  and an error message sent
00058            If IsName(txtForename.Text) Then
00059                'txtforename is ok so is put into the staff structure
00060                staff.Forename = txtForename.Text
00061            Else :MsgBox("forename must be a name" )
00062                Exit Sub
00063            End If
00064
00065            'checks if the surname is made of letters if not then the sub is ended and
                  an error message sent
00066            If IsName(txtSurname.Text) Then
00067                'surname is ok so is put into the staff sturcture
00068                staff.Surname = txtSurname.Text
00069            Else :MsgBox("surname must be a name" )
00070                Exit Sub
00071            End If
00072
00073            'checks if the password is 6 or more letters if not then the sub is ended
                  and an error message sent
00074            If Len(txtPassword.Text) > 5 Then
00075                'password good so is put into the staff stucture
00076                staff.password = Encrypt(txtPassword.Text)
00077            Else
00078                MsgBox("the password must be at least 6 symbols" )
00079            End If
00080
00081            'checks if the points left is numeric if not then an error messesage is sent
                  and the sub ended
00082            If IsNumeric(txtPointsLeft.Text) Then
00083                'the points is ok so is put into the staff structure
00084                staff.PointsLeft = txtPointsLeft.Text
00085            Else :MsgBox("points must be a number" )
00086                Exit Sub
00087            End If
00088
00089            'checks if the admin check box is checked or not if true then true put into
                  the staff structure else a false put in
00090            If chkAdmin.Checked = True Then
00091                staff.admin = True
00092            Else :staff.admin = False
00093            End If
00094
00095            'staff structure put into the staff dat file
00096            PutStaff(staff, staff.staffNO)
00097
00098            'opens the adminstaff form
00099            Admin_Staff.Show()
00100
00101            'updates the adminstaff forms data grid
00102            If Adding = True Then
00103                Nstaff += 1
00104                Admin_Staff.grdStaff.Rows.Add(staff.staffNO, staff.staffID, staff.
                      Forename, staff.Surname, staff.PointsLeft, staff.admin, Decrypt(staff.
                      password))
```

```
00105            Else
00106                Admin_Staff.grdStaff.Rows.Clear()
00107                For counter As Integer = 1 To Nstaff
00108                    staff = GetStaff(counter)
00109                    Admin_Staff.grdStaff.Rows.Add(staff.staffNO, staff.staffID, staff.
                          Forename, staff.Surname, staff.PointsLeft, staff.admin, Decrypt(
                          staff.password))
00110                Next
00111            End If
00112            Me.Close()
00113        End Sub
00114    End Class
```

```
00001  Public Class staffform
00002
00003      Private Sub staffform_Load(ByVal sender As System.Object, ByVal e As System.
            EventArgs) Handles MyBase.Load
00004          'puts relevent information into a lable which says who is logged on and how
                many points they have
00005          lblPointsLeft.Text = "welcome "  & staff.Forename.Trim & " "  & staff.
                Surname.Trim & vbNewLine & "you have "  & staff.PointsLeft & " points left"
00006
00007          'checks weather the staff are admin, if they are it makes the button that
                will take the member of staff to the
00008          'admin form visible
00009          If staff.admin = True Then btnAdmin.Visible = True
00010
00011          'populates the grdpoints with all the different pointsprocedures that the
                staff member has been apart of
00012          For counter As Integer = 1 To NpointsP
00013              pointsP = GetPointsP(counter)
00014              If pointsP.StaffNO = staff.staffNO Then
00015                  student = GetStudent(pointsP.StudNO)
00016                  grdPoints.Rows.Add(student.studID, pointsP.pointstransfered, pointsP
                        .reason)
00017              End If
00018          Next
00019
00020          'populates the grdstudents with all the students
00021          For counter = 1 To NStudents
00022              student = GetStudent(counter)
00023              grdStudents.Rows.Add(student.studID, student.Forename.Trim, student.
                    Surname.Trim, student.points)
00024          Next
00025      End Sub
00026
00027      Private Sub btnchangePassword_Click(ByVal sender As System.Object, ByVal e As
            System.EventArgs) Handles btnChangePassword.Click
00028          'checks if the password boxes are visible
00029          If grpPassword.Visible = False Then
00030              'they arent so that is changed also the btnchangepassword has its text
                    changed to execute
00031              grpPassword.Visible = True
00032              btnBack.Visible = True
00033              btnChangePassword.Text = "execute"
00034          ElseIf Encrypt(txtOldPassword.Text) = staff.password.Trim Then
00035              'the password boxes are visible so it checks if txtoldpassword is the
                    same as the member of
00036              'staffs current password. if not then error message sent and sub ended
00037              'it then checks if the new password has greater then 5 characters if not
                    then error message sent and sub ended
00038              If Len(txtNewPassword1.Text) >= 6 Then
00039                  'it then checks if the new password is alpha numeric if not then an
                        error message is sent and sub ended
00040                  If IsAlpha(txtNewPassword1.Text) = False And IsNumeric(
                        txtNewPassword1.Text) = False Then
00041                      'it then checks if the retype of the new password is the same as
                            the first type. if not then
00042                      'an error message is sent and the sub ended
00043                      If txtNewPassword1.Text = txtNewPassword2.Text Then
00044                          'the new password checked out so it is put into staff
                                structure and that is put into the staff dat
00045                          'file, the password changeing tools are then made invisible
                                and a message box then notifys
```

```
00046                                'the member of staff that the password has been changed
00047                                staff.password = Encrypt(txtNewPassword1.Text.Trim)
00048                                PutStaff(staff, staff.staffNO)
00049                                grpPassword.Visible = False
00050                                btnBack.Visible = False
00051                                btnChangePassword.Text = "change password"
00052                                MsgBox("password changed" )
00053                            Else :MsgBox("your repeat of your new password does not match
                                     that of your intial input" )
00054                            End If
00055                        Else :MsgBox("your password must be alpha numeric" )
00056                        End If
00057                    Else :MsgBox("your password must be at least 6 characters" )
00058                    End If
00059                Else :MsgBox("you have incorrectly input your old password" )
00060                End If
00061
00062            End Sub
00063
00064            Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
                     EventArgs) Handles btnBack.Click
00065                'makes all the different password changeing tools invisible and refreshes
                         the text boxs and
00066                'changes the text of btnchangepassword to change password
00067                grpPassword.Visible = False
00068                txtNewPassword1.Text = ""
00069                txtNewPassword2.Text = ""
00070                txtOldPassword.Text = ""
00071                btnBack.Visible = False
00072                btnChangePassword.Text = "change password"
00073            End Sub
00074
00075
00076            Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.
                     EventArgs) Handles btnExit.Click
00077                'makes logonform visible and close staffform
00078                loginFrom.Visible = True
00079                Me.Close()
00080            End Sub
00081
00082
00083
00084            Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As System.
                     EventArgs) Handles btnSearch.Click
00085                Dim counter As Integer
00086                'checks if there is a word in txtforename, serves as presence check and word
                         check
00087                If IsAlpha(txtForname.Text.Trim) = True Then
00088                    'checks if there is a word in txtsurname, serves as presence check and
                             word check
00089                    If IsAlpha(txtSurname.Text.Trim) = True Then
00090                        'clears grdstudents populates grdstudents with only students whose
                                 first name starts with the
00091                        'text in txtforename and whose last name starts with the text in
                                 txtsurname
00092                        grdStudents.Rows.Clear()
00093                        For counter = 1 To NStudents
00094                            student = GetStudent(counter)
00095                            If student.Forename.ToUpper.StartsWith(txtForname.Text.Trim.
                                     ToUpper) And student.Surname.ToUpper.StartsWith(txtSurname.Text
                                     .ToUpper.Trim) Then
```

```
00096                    grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                          student.Surname.Trim, student.points)
00097                End If
00098            Next
00099
00100            'if only the forname box contains letters and txtsurname is empty
00101        ElseIf txtSurname.Text = ""  Then
00102            'clears grdstudents then populates the grdstudents with only
                  students whose first name
00103            'starts with the text in txtforename
00104            grdStudents.Rows.Clear()
00105            For counter = 1 To NStudents
00106                student = GetStudent(counter)
00107                If student.Forename.ToUpper.StartsWith(txtForname.Text.Trim.
                      ToUpper) Then
00108                    grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                          student.Surname.Trim, student.points)
00109                End If
00110            Next
00111            'txtsurname contains invalid data so error message sent
00112        Else :MsgBox("the surname must be only letters with no spaces" )
00113        End If
00114    ElseIf txtForname.Text = ""  Then
00115        'checks if there is a word in txtsurname, serves as presence check and
              word check
00116        If IsAlpha(txtSurname.Text) = True Then
00117            'clears grdstudents then populates the grdstudents with only
                  students whose second name
00118            'starts with the text in txtsurname
00119            grdStudents.Rows.Clear()
00120            For counter = 1 To NStudents
00121                student = GetStudent(counter)
00122                If student.Surname.ToUpper.StartsWith(txtSurname.Text.Trim.
                      ToUpper) Then
00123                    grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                          student.Surname.Trim, student.points)
00124                End If
00125            Next
00126            'if neither box contains letters
00127        ElseIf txtSurname.Text = ""  Then
00128            'clears grdstudents then populates it with all students
00129            grdStudents.Rows.Clear()
00130            For counter = 1 To NStudents
00131                student = GetStudent(counter)
00132                grdStudents.Rows.Add(student.studID, student.Forename.Trim,
                      student.Surname.Trim, student.points)
00133            Next
00134            'txtsurname contains invalid data so error message sent
00135        Else :MsgBox("surname must be only letters with no spaces" )
00136        End If
00137        'txtforname contains invalid data so error message sent
00138    Else :MsgBox("the forename must be only letters with no spaces" )
00139    End If
00140End Sub
00141
00142Private Sub grdStudents_CellContentdoubleClick(ByVal sender As System.Object,
 ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles grdStudents.
 CellContentClick
00143    Dim ID As Integer
00144    'intialise variable but if it doesnt work because content selected is a
          header it will exit sub
```

```
00145            Try
00146                ID = grdStudents.Rows(e.RowIndex).Cells(0).Value
00147            Catch
00148                Exit Sub
00149            End Try
00150            'gets student by comparing the ID in the row against the studID of each
                  student till a match is made
00151            For counter As Integer = 1 To NStudents
00152                student = GetStudent(counter)
00153                If student.studID = ID Then
00154                    'match found student records form opens
00155                    StudentRecordsForm.Show()
00156                    Exit Sub
00157                End If
00158            Next
00159        End Sub
00160
00161
00162        Private Sub btnPoints_Click(sender As System.Object, e As System.EventArgs)
             Handles btnPoints.Click
00163            Dim counter As Integer
00164            'checks if grid points is visible
00165            If grdPoints.Visible = True Then
00166                'grdpoints is visible so it is made invisible and all the different
                      points transfer tools are made
00167                'visible
00168                grdPoints.Visible = False
00169                btnPointsBack.Visible = True
00170                txtStudID.Visible = True
00171                txtPoints.Visible = True
00172                Label2.Visible = True
00173                Label9.Visible = True
00174                chkGive.Visible = True
00175                chkTake.Visible = True
00176                Lblreason.Visible = True
00177                txtReason.Visible = True
00178                lblPointsLeft.Text = "welcome "  & staff.Forename.Trim & " "  & staff.
                      Surname.Trim & vbNewLine & "you have "  & staff.PointsLeft & " points
                      left"
00179            ElseIf grdPoints.Visible = False Then
00180                'grdpoints isnt visible so txtstudId is checked if it is numeric. if not
                      then an error message is sent and
00181                'the sub ended
00182                If IsNumeric(txtStudID.Text) Then
00183                    'checks if the txtpoints is numberic. if not then an error message
                          is sent and the sub ended
00184                    If IsNumeric(txtPoints.Text) Then
00185                        'checks if there is a reason inputed. if not then an error
                              message is sent and the sub ended
00186                        If txtReason.Text <> ""  Then
00187                            'checks whether the staff member is giving points or not
00188                            If chkGive.Checked = True Then
00189                                'give points was checked so it is checked wether the
                                      member of staff has enough points to give
00190                                'the amount they want to. if they dont then an error
                                      message is sent and the sub ended
00191                                If staff.PointsLeft >= txtPoints.Text Then
00192                                    'the member of staff does have enough points so the
                                          records of the student who is
00193                                    'being given the points is gotten
00194                                    For counter = 1 To NStudents
```

```
    student = GetStudent(counter)
    If student.studID = txtStudID.Text Then
        Exit For
    End If
Next
'the student is given the points
student.points += txtPoints.Text
'the students edited records are put into the student dat file
PutStudent(student, student.studNO)
'the points are then taken away from the staffs points
staff.PointsLeft -= txtPoints.Text
'the staff members edited records are put into the staff dat file
PutStaff(staff, staff.staffNO)
'the data is put into the pointsprocedure structure
NpointsP += 1
pointsP.StaffNO = staff.staffNO
pointsP.StudNO = student.studNO
pointsP.reason = txtReason.Text
pointsP.pointstransfered = txtPoints.Text
'the data in the points procedure structure is transfered to the pointsprocedure dat file
PutPointsP(pointsP, NpointsP)
' a message box notifies the member of staff that the transfer is complete
MsgBox("points transfered" )
'grdpoints is made visible again and the pointsprocedure tools are made invisible
grdPoints.Visible = True
btnPointsBack.Visible = False
txtStudID.Visible = False
txtPoints.Visible = False
Label2.Visible = False
Label9.Visible = False
chkGive.Visible = False
chkTake.Visible = False
Lblreason.Visible = False
txtReason.Visible = False
txtReason.Text = ""
txtPoints.Text = ""
txtStudID.Text = ""
'lblPointsLeft is refreshed
lblPointsLeft.Text = "welcome "  & staff.Forename.Trim & " "  & staff.Surname.Trim & vbNewLine & "you have "  & staff.PointsLeft & " points left"
'grdpoints is updated
grdPoints.Rows.Clear()
For counter = 1 To NpointsP
    pointsP = GetPointsP(counter)
    If pointsP.StaffNO = staff.staffNO Then
        student = GetStudent(pointsP.StudNO)
        grdPoints.Rows.Add(student.studID, pointsP.pointstransfered, pointsP.reason)
    End If
Next
'grdstudents is updated
grdStudents.Rows.Clear()
For counter = 1 To NStudents
    student = GetStudent(counter)
```

```
00247                                                grdStudents.Rows.Add(student.studID, student.
                                                      Forename.Trim, student.Surname.Trim, student.
                                                      points)
00248                                            Next
00249                                        Else :MsgBox("you do not have enough points left" )
00250                                        End If
00251                                    Else
00252                                        'points are being deducted
00253                                        'gets the records of the student the points are being
                                              deducted from
00254                                        For counter = 1 To NStudents
00255                                            student = GetStudent(counter)
00256                                            If student.studID = txtStudID.Text Then
00257                                                Exit For
00258                                            End If
00259                                        Next
00260                                        'checks if the student has less points then the amount
                                              being deducted if so then students
00261                                        'points become 0 instead of a minus number
00262                                        If txtPoints.Text > student.points Then
00263                                            student.points = 0
00264                                        Else
00265                                            student.points -= txtPoints.Text
00266                                        End If
00267                                        'students edited data is put into the studnet dat file
00268                                        PutStudent(student, student.studNO)
00269                                        'relevent data is put into the points procedure
                                              structure
00270                                        NpointsP += 1
00271                                        pointsP.StaffNO = staff.staffNO
00272                                        pointsP.StudNO = student.studNO
00273                                        pointsP.reason = txtReason.Text
00274                                        pointsP.pointstransfered = -(txtPoints.Text)
00275                                        'data is from the pointsprocedure structure is put into
                                              the pointsprocedure dat file
00276                                        PutPointsP(pointsP, NpointsP)
00277                                        'a message is sent notifying staff member that points
                                              transfer has been completed
00278                                        MsgBox("points deducted" )
00279                                        'grdpoints is made visible and the points transfer tools
                                              are made inivisible
00280                                        grdPoints.Visible = True
00281                                        btnPointsBack.Visible = False
00282                                        txtStudID.Visible = False
00283                                        txtPoints.Visible = False
00284                                        Label2.Visible = False
00285                                        Label9.Visible = False
00286                                        chkGive.Visible = False
00287                                        chkTake.Visible = False
00288                                        Lblreason.Visible = False
00289                                        txtReason.Visible = False
00290                                        txtReason.Text = ""
00291                                        txtPoints.Text = ""
00292                                        txtStudID.Text = ""
00293                                        'grdpoints are refreshed
00294                                        grdPoints.Rows.Clear()
00295                                        For counter = 1 To NpointsP
00296                                            pointsP = GetPointsP(counter)
00297                                            If pointsP.StaffNO = staff.staffNO Then
00298                                                student = GetStudent(pointsP.StudNO)
00299                                                grdPoints.Rows.Add(student.studID, pointsP.
```

```
                                        pointstransfered, pointsP.reason)
00300                                   End If
00301                               Next
00302                               'grdstudents are refreshed
00303                               grdStudents.Rows.Clear()
00304                               For counter = 1 To NStudents
00305                                   student = GetStudent(counter)
00306                                   grdStudents.Rows.Add(student.studID, student.
                                         Forename.Trim, student.Surname.Trim, student.points
                                         )
00307                               Next
00308                           End If
00309                       Else :MsgBox("there must be a reason" )
00310                       End If
00311                   Else :MsgBox("the points must be a number" )
00312                   End If
00313               Else :MsgBox("the student ID must be a number" )
00314               End If
00315           End If
00316
00317
00318
00319       End Sub
00320
00321
00322       Private Sub btnPointsBack_Click(ByVal sender As System.Object, ByVal e As System
            .EventArgs) Handles btnPointsBack.Click
00323           'hides the change password tools
00324           grdPoints.Visible = True
00325           btnPointsBack.Visible = False
00326           txtStudID.Visible = False
00327           txtPoints.Visible = False
00328           Label2.Visible = False
00329           Label9.Visible = False
00330           chkGive.Visible = False
00331           chkTake.Visible = False
00332           Lblreason.Visible = False
00333           txtReason.Visible = False
00334           txtStudID.Text = ""
00335           txtPoints.Text = ""
00336           txtReason.Text = ""
00337       End Sub
00338
00339       Private Sub chkGive_CheckedChanged(sender As System.Object, e As System.
            EventArgs) Handles chkGive.CheckedChanged
00340           'makes sure only one of the check boxes can be checked at one time
00341           If chkGive.Checked = True Then
00342               chkTake.Checked = False
00343           Else :chkTake.Checked = True
00344           End If
00345       End Sub
00346
00347       Private Sub chkTake_CheckedChanged(sender As System.Object, e As System.
            EventArgs) Handles chkTake.CheckedChanged
00348           'makes sure only one of the check boxes can be checked at one time
00349           If chkTake.Checked = True Then
00350               chkGive.Checked = False
00351           Else :chkGive.Checked = True
00352           End If
00353       End Sub
00354
```

```
00355        Private Sub btnAdmin_Click(sender As System.Object, e As System.EventArgs)
              Handles btnAdmin.Click
00356            'opens the admin directory form and hides the staffform
00357            AdminDirectory.Show()
00358            Me.Hide()
00359
00360        End Sub
00361    End Class
```

```
00001  Public Class StudentForm1
00002
00003      Private Sub btnExit_Click(sender As System.Object, e As System.EventArgs)
           Handles btnExit.Click
00004          'shows the logon form and closes the studnet form
00005          loginFrom.Show()
00006          Me.Close()
00007      End Sub
00008
00009      Private Sub btnPassword_Click(sender As System.Object, e As System.EventArgs)
           Handles btnPassword.Click
00010          'checks if grppassword is visible
00011          If grpPassword.Visible = False Then
00012              'password changing tools are made visible
00013              grpPassword.Visible = True
00014              btnBack.Visible = True
00015              btnPassword.Text = "execute"
00016          ElseIf Encrypt(txtOldPassword.Text) = student.password.Trim Then
00017              'txtoldpassword is compared to the students current password. if
                   incorrect then an error message will be
00018              'sent and the sub ended.
00019              'check the newpassword to make sure it is more then 5 characters. if it
                   isnt then an error message will be
00020              'sent and the sub ended
00021              If Len(txtNewPassword1.Text) >= 6 Then
00022                  'check if the new password is alphanumeric. if not then an error
                       message will be sent and the sub ended
00023                  If IsAlpha(txtNewPassword1.Text) = False And IsNumeric(
                       txtNewPassword1.Text) = False Then
00024                      'check if the newpassword is the same as the retype. if not then
                           an error message will be sent and
00025                      'the sub ended
00026                      If txtNewPassword1.Text = txtNewPassword2.Text Then
00027                          'the student password will be changed and put into the dat
                               file
00028                          student.password = Encrypt(txtNewPassword1.Text.Trim)
00029                          PutStudent(student, student.studNO)
00030                          'the password changeing tools are made invisible again
00031                          grpPassword.Visible = False
00032                          btnBack.Visible = False
00033                          btnPassword.Text = "change password"
00034                          'a message is sent telling the studnet that the password has
                               changed
00035                          MsgBox("password changed" )
00036                      Else :MsgBox("your repeat of your new password does not match
                           that of your intial input" )
00037                      End If
00038                  Else :MsgBox("your password must be alpha numeric" )
00039                  End If
00040              Else :MsgBox("your password must be at least 6 characters" )
00041              End If
00042          Else :MsgBox("you have incorrectly input your old password" )
00043          End If
00044      End Sub
00045
00046      Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnBack.Click
00047          'the password changing tools are made invisible and reset
00048          grpPassword.Visible = False
00049          txtNewPassword1.Text = ""
00050          txtNewPassword2.Text = ""
```

```
00051            txtOldPassword.Text = ""
00052            btnBack.Visible = False
00053            btnPassword.Text = "change password"
00054        End Sub
00055
00056
00057        Private Sub StudentForm1_Load(sender As System.Object, e As System.EventArgs)
             Handles MyBase.Load
00058            'puts relevent information into a lable which says who is logged on and how
                 many points they have
00059            lblName.Text = "welcome "  & student.Forename.Trim & " "  & student.Surname.
                 Trim & vbNewLine & "you have "  & student.points & " points"
00060
00061            'populate the grdrewards
00062            For counter As Integer = 1 To NRewards
00063                Dim onreward = GetReward(counter)
00064                If student.points > onreward.cost Then
00065                    grdRewards.Rows.Add(onreward.RewardID, onreward.name.Trim, onreward.
                         cost)
00066                End If
00067            Next
00068            'populates grdrecords with all the points procedure that protain to the
                 student
00069            For counter = 1 To NpointsP
00070                Dim OnPointsPRec = getPointsP(counter)
00071                If OnPointsPRec.StudNO = student.studNO Then
00072                    staff = GetStaff(pointsP.StaffNO)
00073                    grdRecords.Rows.Add(staff.Surname, OnPointsPRec.pointstransfered,
                         OnPointsPRec.reason)
00074                End If
00075            Next
00076
00077        End Sub
00078
00079        Private Sub grdRewards_CellContentDoubleClick(sender As System.Object, e As
             System.Windows.Forms.DataGridViewCellEventArgs) Handles grdRewards.
             CellContentClick
00080            Dim RewardSelected As Integer
00081            'works out which reward was clicked on
00082            RewardSelected = grdRewards.Rows(e.RowIndex).Cells(0).Value
00083            'gets the data for the reward clicked on
00084            reward = GetReward(RewardSelected)
00085            'checks if the student is sure they want to buy the reward.
00086            If MsgBox("Are you sure you would like to buy "  & reward.name.Trim & " for
                 "  & reward.cost & " points" , vbYesNo) = vbYes Then
00087                'answer was yes so the the students points total has the cost of the
                     points deducted from it and is put
00088                'back into the dat file
00089                student.points -= reward.cost
00090                PutStudent(student, student.studNO)
00091                'refreshing lblname
00092                lblName.Text = "welcome "  & student.Forename.Trim & " "  & student.
                     Surname.Trim & vbNewLine & "you have "  & student.points & " points"
00093                'adding rewardsprocedure
00094                RewardsP.RewardID = reward.RewardID
00095                RewardsP.studNO = student.studNO
00096                NrewardsP += 1
00097                PutRewardP(RewardsP, NrewardsP)
00098            End If
00099        End Sub
00100    End Class
```

```
00001  Public Class StudentEdit
00002
00003      Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnBack.Click
00004          'shows the adminstudent form and closes the studentedit form
00005          Admin_Student.Show()
00006          Me.Close()
00007      End Sub
00008
00009      Private Sub StudentEdit_Load(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles MyBase.Load
00010          If Adding = True Then
00011              'adding is true so the text boxes are empty and a new studnet Number is
                   put into lblstudNo
00012              lblStudNO.Text = NStudents + 1
00013              student = GetStudent(NStudents)
00014              txtStudID.Text = student.studID + 1
00015          Else
00016              'adding is false there for the form is in change mode so the data of the
                   student selected to be changed
00017              'was put into the student structure is now put into the text boxes and
                   the text on the btn is changed to change
00018              lblStudNO.Text = student.studNO
00019              txtForename.Text = student.Forename
00020              txtSurname.Text = student.Surname
00021              txtStudID.Text = student.studID
00022              txtPoints.Text = student.points
00023              txtPassword.Text = Decrypt(student.password)
00024              btnEnd.Text = "change"
00025          End If
00026      End Sub
00027
00028      Private Sub btnEnd_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnEnd.Click
00029          'checks if studID is numeric if not then an error message is sent and the
                sub is ended
00030          If IsNumeric(txtStudID.Text) = True Then
00031              'checks if the Id is 6 figures and positive if not then an error message
                   is sent and the sub ended
00032              If Int(txtStudID.Text) > 0 And Int(txtStudID.Text) < 1000000 Then
00033                  'checks if the studID is unique, if not then an error messages is
                        sent and the sub is ended
00034                  ' if it is then it is put into the studnet structure
00035                  For counter As Integer = 1 To NStudents
00036                      student = GetStudent(counter)
00037                      If student.studID = Int(txtStudID.Text) Then
00038                          If student.studNO <> lblStudNO.Text Then
00039                              MsgBox("ID already in use try another" )
00040                              Exit Sub
00041                          End If
00042                      Else :student.studID = Int(txtStudID.Text)
00043                      End If
00044                  Next
00045              Else :MsgBox("the id must be greater then 0 and less then 1000000" )
00046                  Exit Sub
00047              End If
00048          Else :MsgBox("the Id must be a number" )
00049              Exit Sub
00050          End If
00051
00052          'checks if the forname is a name. if not then an error message is sent and
```

```
            the sub is ended
00053       'if so it is put into the student structure
00054       If IsName(txtForename.Text) Then
00055           student.Forename = txtForename.Text
00056       Else :MsgBox("forename must be a name" )
00057           Exit Sub
00058       End If
00059
00060       'checks if the surname is a name. if not then an error message is sent and
            the sub is ended
00061       'if so it is put into the student structure
00062       If IsName(txtSurname.Text) Then
00063           student.Surname = txtSurname.Text
00064       Else :MsgBox("surname must be a name" )
00065           Exit Sub
00066       End If
00067
00068       'checks if the password is greater then 5 characters if not then an error
            message is sent and the sub ended
00069       'if so it is put into the student structure
00070       If Len(txtPassword.Text) > 5 Then
00071           student.password = Encrypt(txtPassword.Text)
00072       Else
00073           MsgBox("the password must be at least 6 characters" )
00074           Exit Sub
00075       End If
00076
00077       'checks if the points are numeric if not then an error message is sent and
            the sub ended
00078       'if so it is put into the student structure
00079       If IsNumeric(txtPoints.Text) Then
00080           student.points = txtPoints.Text
00081       Else :MsgBox("points must be a number" )
00082           Exit Sub
00083       End If
00084
00085       'lblstudNo is put into the student structure
00086       student.studNO = lblStudNO.Text
00087
00088       'the student structure is put into the studnet dat file
00089       PutStudent(student, student.studNO)
00090
00091       'refeshes grdstudents on adminstudnet
00092       If Adding = True Then
00093           NStudents += 1
00094           Admin_Student.grdStudents.Rows.Add(student.studID, student.Forename.Trim
                , student.Surname.Trim, student.points, Decrypt(student.password))
00095       Else
00096           Admin_Student.grdStudents.Rows.Clear()
00097           For counter As Integer = 1 To NStudents
00098               student = GetStudent(counter)
00099               Admin_Student.grdStudents.Rows.Add(student.studID, student.Forename.
                    Trim, student.Surname.Trim, student.points, Decrypt(student.
                    password))
00100           Next
00101       End If
00102
00103       'closes the form
00104       Me.Close()
00105   End Sub
00106 End Class
```

```
00001  Public Class StudentRecordsForm
00002
00003      Private Sub StudentRecordsForm_Load(ByVal sender As System.Object, ByVal e As
           System.EventArgs) Handles MyBase.Load
00004          'loads relevent information like name and points into a label
00005          lblPoints.Text = student.Forename.Trim & " "  & student.Surname.Trim & " has
               "  & student.points & " points"
00006
00007          'populates grdrecords with all the points procedures protaining to the
               selected student
00008          For counter As Integer = 1 To NpointsP
00009              Dim OnPointsPRec = GetPointsP(counter)
00010              If OnPointsPRec.StudNO = student.studNO Then
00011                  staff = GetStaff(pointsP.StaffNO)
00012                  grdRecords.Rows.Add(staff.Surname, OnPointsPRec.pointstransfered,
                       OnPointsPRec.reason)
00013              End If
00014          Next
00015      End Sub
00016
00017      Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.
           EventArgs) Handles btnBack.Click
00018          'closes the form
00019          Me.Close()
00020      End Sub
00021  End Class
```