

```
'Allows users to create Customers and Accounts and select Customer Details
'and Account Details screens.
'Written by Bill Blackley
'First Written 23/6/05
'Last written 27/8/05

Public Class frmMainMenu
    Inherits System.Windows.Forms.Form

    Windows Form Designer generated code

    'Deletes IP Address of machine that the user logged in on from
    'Staff.localIPAddress attribute, calls the garbage collector and
    'closes the FAMS application
    Private Sub mnuLogout_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles mnuLogout.Click

        Try

            'Open Database connection
            FAMSDBConnection.Open()

            'Create Update Command Text
            Dim SQLUpdateStatement As String = "UPDATE Staff SET localIPAddress = ' " & _
            Nothing & " ' WHERE localIPAddress = ' " & frmLogin.GetIPAddress() & "'

            'Set Update Command to update Staff.localIPAddress attribute of FAMSDB
            'to NULL
            staffDataAdapter.UpdateCommand.CommandText = SQLUpdateStatement

            'Execute the Update Command
            staffDataAdapter.UpdateCommand.ExecuteNonQuery()

            'Close FAMSDB Database connection
            FAMSDBConnection.Close()

            'Call the Garbage Collector
            System.GC.Collect()

            'Close FAMS Application
            Application.Exit()

        Catch ex As Exception
            MessageBox.Show(ex.ToString())
        End Try

    End Sub 'mnuLogout_Click

    'disables the close button (to force the user to Logout) when
    'frmMainNemu is loaded
    Private Sub frmMainMenu_Load(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles MyBase.Load
        'Disable close button
        CloseButton.Disable(Me)

    End Sub 'frmMainMenu_Load

    'disables the close button (to force the user to Logout) when
    'frmMainNemu is resized
    Private Sub frmMainMenu_SizeChanged(ByVal sender As Object, ByVal e As System.
    EventArgs) Handles MyBase.SizeChanged
        CloseButton.Disable(Me)
    End Sub 'frmMainMenu_Sizechange

    'Open form frmChangePassword
    Private Sub mnuChangePassword_Click(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles mnuChangePassword.Click
        Dim winChangePassword As New frmChangePassword
        winChangePassword.Show()
    End Sub 'mnuChangePassword

    'Randomly generates a new customer number by calling NewCustomerNo() method, displays
    'the new customer number in frmMainMenu and INSERTs a record in the
    'CustomerMaintenance table to indicate which member of staff created the new
```

```

'customer.
Private Sub cmdCreateCustomer_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles cmdCreateCustomer.Click
    Try
        Dim Result As DialogResult

        'Ask user if they're sure they want to Create a New Customer
        Result = MessageBox.Show("Are you sure you want to Create New Customer?",
"Create Customer?", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

        'if user clicks Yes....
        If Result = DialogResult.Yes Then

            'Generate New Customer Number Randomly, display it in frmMainMenu
            'and INSERT new customer record in Customer Table of FAMSDB
            NewCustomerNo()

            'Open Database Connection
            FAMSDBConnection.Open()

            'Return the DataSet to it's original state
            StaffDataSet.Reset()

            'Select staff Number of staff member that is logged in
            staffDataAdapter.SelectCommand.CommandText = _
            "SELECT staffNo FROM Staff WHERE localIPAddress = '" & _
            frmLogin.GetIPAddress() & "'"

            'Fill StaffDataSet with results of query
            staffDataAdapter.Fill(StaffDataSet)

            'put result of query into datatable with which to work
            Dim datatable As DataTable = StaffDataSet.Tables(0)

            'Create Insert Command
            CustomerMaintenanceDataAdapter.InsertCommand.CommandText = _
            "INSERT INTO CustomerMaintenance " & _
            "VALUES ('" & Convert.ToString(datatable.Rows(0)(0)) & "', '" & _
            txtCustomerNo.Text & "', '1', '" & Convert.ToString(DateTime.Now.ToString(
"d")) & _
            "', 'Created')"

            'Execute Insert Command
            CustomerMaintenanceDataAdapter.InsertCommand.ExecuteNonQuery()

            'Close Database Connection
            FAMSDBConnection.Close()

        End If

    Catch ex As Exception
        MessageBox.Show(ex.ToString())
    End Try
End Sub 'cmdCreateCustomer_Click

'Randomly generates an new account number for the sort code the user has enetered,
'displays the new account number in frmMainMenu and INSERTs a record in the
'AccountMaintenance Table indicating which member of staff created the new account
Private Sub cmdCreateAccount_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles cmdCreateAccount.Click
    Try
        Dim Result As DialogResult

        'As user if they're sure they want to create a new account
        Result = MessageBox.Show("Are you sure you want to Create New Account?",
"Create New Account?", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

        'If user clicks yes.....
        If Result = DialogResult.Yes Then

            If txtSortCode.Text.Equals("") Then

```

```

        'Tell user they must eneter sort code
        MessageBox.Show("Must Enter Sort Code in the range 990000 to 990271",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

    Else

        'Check sort code that the user enetered in valid
        If (Convert.ToInt32(txtSortCode.Text) >= 990000 And _
Convert.ToInt32(txtSortCode.Text) <= 990271) Then

            'Generate New Account Number, display it in frmMainMenu and
            'INSERT new account Record in Account Table of FAMSDB
            NewAccountNo()

            'Open Database Connection
            FAMSDBConnection.Open()

            'Return the DataSet to it's original state
            StaffDataSet.Reset()

            'Select staff Number of staff member that is logged in
            staffDataAdapter.SelectCommand.CommandText = _
            "SELECT staffNo FROM Staff WHERE localIPAddress = '" & _
            frmLogin.GetIPAddress() & "'"

            'Fill StaffDataSet with results of query
            staffDataAdapter.Fill(StaffDataSet)

            'put result of query into datatable with which to work
            Dim datatable As DataTable = StaffDataSet.Tables(0)

            'Create Insert Command
            AccountMaintenanceDataAdapter.InsertCommand.CommandText = _
            "INSERT INTO AccountMaintenance VALUES(" & _
            "'" & txtSortCode.Text & "', " & _
            "'" & txtAccountNo.Text & "', " & _
            "'" & Convert.ToString(datatable.Rows(0)(0)) & "', '1', " & _
            "'" & Convert.ToString(DateTime.Now.ToString("d")) & "', " & _
            "'Created'"

            'Execute INSERT Command
            AccountMaintenanceDataAdapter.InsertCommand.ExecuteNonQuery()

            'Close Database Connection
            FAMSDBConnection.Close()

        Else

            'Display Error message if invalid sort code is enetered
            MessageBox.Show("Invalid Sort Code! Sort Code must be in the range
990000 to 990271", "Invalid Sort Code!", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If

    End If

End If

Catch ex As Exception
    MessageBox.Show(ex.ToString())
End Try
End Sub 'cmdCreateAccount

'Displays the customer's data and the accounts the customer holds in
'frmCustomerDetails
Private Sub cmdViewCustomer_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles cmdViewCustomer.Click

    Try

        'Open Database connection
        FAMSDBConnection.Open()

        'Returns the DataSet to it's original state
        CustomerDataSet.Reset()

        'Select Customer Details
        CustomerDataAdapter.SelectCommand.CommandText = _

```

```

"SELECT * FROM Customer WHERE customerNo = '" & txtCustomerNo.Text & "'"

'Fill DataSet
CustomerDataAdapter.Fill(CustomerDataSet)

'Close database connection
FAMSDBConnection.Close()

'Put results of query into databale with which to work
Dim datatable As DataTable = CustomerDataSet.Tables(0)

'If Customer record exists.....
If datatable.Rows.Count <> 0 Then

    'Open frmCustomerDetails
    Dim winCustomerDetails As New frmCustomerDetails
    winCustomerDetails.Show()

    'Fill form with customer data
    winCustomerDetails.txtCustomerNo.Text = Convert.ToString(datatable.Rows(0)
(0))
    winCustomerDetails.txtTitle.Text = Convert.ToString(datatable.Rows(0)(1))
    winCustomerDetails.txtForenames.Text = Convert.ToString(datatable.Rows(0)
(2))
    winCustomerDetails.txtSurname.Text = Convert.ToString(datatable.Rows(0)(3))
    winCustomerDetails.txtAddressLine1.Text = Convert.ToString(datatable.Rows
(0)(4))
    winCustomerDetails.txtAddressLine2.Text = Convert.ToString(datatable.Rows
(0)(5))
    winCustomerDetails.txtAddressLine3.Text = Convert.ToString(datatable.Rows
(0)(6))
    winCustomerDetails.txtCity.Text = Convert.ToString(datatable.Rows(0)(7))
    winCustomerDetails.txtCounty.Text = Convert.ToString(datatable.Rows(0)(8))
    winCustomerDetails.txtPostCode.Text = Convert.ToString(datatable.Rows(0)
(9))
    winCustomerDetails.txtTelephoneNo.Text = Convert.ToString(datatable.Rows(0)
(10))
    winCustomerDetails.txtEmailAddress.Text = Convert.ToString(datatable.Rows
(0)(11))
    winCustomerDetails.txtDoB.Text = Convert.ToString(datatable.Rows(0)(12))
    winCustomerDetails.txtDoD.Text = Convert.ToString(datatable.Rows(0)(13))

    'Open Database Connection
    FAMSDBConnection.Open()

    'Returns the DataSet to it's original state
    AccountDataSet.Reset()

    'Select Accounts Held by Customer
    AccountDataAdapter.SelectCommand.CommandText = _
"SELECT sortCode, accountNo, productType FROM Account " & _
"WHERE customerNo1 = '" & txtCustomerNo.Text & "'" & _
"OR customerNo2 = '" & txtCustomerNo.Text & "'" & _
"OR customerNo3 = '" & txtCustomerNo.Text & "'"

    'Fill AccountDataSet
    AccountDataAdapter.Fill(AccountDataSet)

    'Close Database Connection
    FAMSDBConnection.Close()

    'fill DataGridView dgdAccountsHeld
    winCustomerDetails.dgdAccountsHeld.SetDataBinding(AccountDataSet, "Account"
)

Else
    'Display warning message if customer if not found
    MessageBox.Show("Customer Does Not Exist!", "No Customer Found",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If

Catch ex As Exception
    MessageBox.Show(ex.ToString())
End Try

```

```
End Sub 'cmdViewCustomer_Click

'Displays the data relating to the customer's account in frmAccountDetails
Private Sub cmdViewAccount_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles cmdViewAccount.Click

    Try
        'Open Database Connection
        FAMSDBConnection.Open()

        'Returns the DataSet to it's original state
        AccountDataSet.Reset()

        'Select Account Details
        AccountDataAdapter.SelectCommand.CommandText = _
        "SELECT * FROM Account WHERE sortCode = '" & _
        txtSortCode.Text & "' AND accountNo = '" & _
        txtAccountNo.Text & "'"

        'Fill AccountDataSet with account data selected in query
        AccountDataAdapter.Fill(AccountDataSet)

        'Close Database Connection
        FAMSDBConnection.Close()

        'Put Data in datatable with which to work
        Dim datatable As DataTable = AccountDataSet.Tables(0)

        'If account exists.....
        If datatable.Rows.Count <> 0 Then

            '.....open form frmAccountDetails
            Dim winAccountDetails As New frmAccountDetails
            winAccountDetails.Show()

            'If ProductType field is non-empty disable it
            If Convert.ToString(datatable.Rows(0)(11)).Length <> 0 Then
                winAccountDetails.cboProductType.Enabled = False
            End If

            'If Closing Date field is non-empty and status field is set to
            '"Closed", disable most of form frmAccountDetails
            If (Convert.ToString(datatable.Rows(0)(7)).Length <> 0 And _
            Convert.ToString(datatable.Rows(0)(5)).Equals("Closed")) Then

                winAccountDetails.txtCustomerNo1.Enabled = False
                winAccountDetails.txtCustomerNo2.Enabled = False
                winAccountDetails.txtCustomerNo3.Enabled = False

                winAccountDetails.txtTitle1.Enabled = False
                winAccountDetails.txtTitle2.Enabled = False
                winAccountDetails.txtTitle3.Enabled = False

                winAccountDetails.txtForenames1.Enabled = False
                winAccountDetails.txtForenames2.Enabled = False
                winAccountDetails.txtForenames3.Enabled = False

                winAccountDetails.txtSurname1.Enabled = False
                winAccountDetails.txtSurname2.Enabled = False
                winAccountDetails.txtSurname3.Enabled = False

                winAccountDetails.cboAccountStatus.Enabled = False

                winAccountDetails.txtBalance.Enabled = False
                winAccountDetails.txtAvailableBalance.Enabled = False
                winAccountDetails.txtInterestDue.Enabled = False
                winAccountDetails.txtChargesDue.Enabled = False
                winAccountDetails.txtOverdraftLimit.Enabled = False

                winAccountDetails.txtOpeningDate.Enabled = False
                winAccountDetails.txtClosingDate.Enabled = False

                winAccountDetails.mnuDeposits.Enabled = False
            End If
        End If
    End Try
End Sub
```

```
        winAccountDetails.mnuWithdrawals.Enabled = False
        winAccountDetails.mnuUpdate.Enabled = False
    End If

    'Now display account details in form frmAccountDetails
    winAccountDetails.txtSortCode.Text = Convert.ToString(datatable.Rows(0)(0))
    winAccountDetails.txtAccountNo.Text = Convert.ToString(datatable.Rows(0)
(1)
Rows(0)(2))
    winAccountDetails.txtAvailableBalance.Text = Convert.ToString(datatable.
    winAccountDetails.txtBalance.Text = Convert.ToString(datatable.Rows(0)(3))
    winAccountDetails.txtOverdraftLimit.Text = Convert.ToString(datatable.Rows
(0)(4))
    winAccountDetails.cboAccountStatus.Text = Convert.ToString(datatable.Rows
(0)(5))
    winAccountDetails.txtOpeningDate.Text = Convert.ToString(datatable.Rows(0)
(6)
    winAccountDetails.txtClosingDate.Text = Convert.ToString(datatable.Rows(0)
(7)
    winAccountDetails.txtInterestDue.Text = Convert.ToString(datatable.Rows(0)
(8)
    winAccountDetails.txtChargesDue.Text = Convert.ToString(datatable.Rows(0)
(9)
    winAccountDetails.txtNotes.Text = Convert.ToString(datatable.Rows(0)(10))
    winAccountDetails.cboProductType.Text = Convert.ToString(datatable.Rows(0)
(11)
    winAccountDetails.txtCustomerNo1.Text = Convert.ToString(datatable.Rows(0)
(12)
    winAccountDetails.txtCustomerNo2.Text = Convert.ToString(datatable.Rows(0)
(13)
    winAccountDetails.txtCustomerNo3.Text = Convert.ToString(datatable.Rows(0)
(14)

    'Now display account holder's names

    'display first account holder's name if they exist
    If winAccountDetails.txtCustomerNo1.Text.Length <> 0 Then

        'Open Database Connection
        FAMSDBConnection.Open()

        'Returns the DataSet to it's original state
        CustomerDataSet.Reset()

        'Select customerNo1's name
        CustomerDataAdapter.SelectCommand.CommandText = _
        "SELECT title, forenames, surname FROM Customer WHERE " & _
        "customerNo = '" & winAccountDetails.txtCustomerNo1.Text & "'"

        'Fill CustomerDataSet
        CustomerDataAdapter.Fill(CustomerDataSet)

        'close database connection
        FAMSDBConnection.Close()

        'Fill datatable with results of query
        Dim datatable1 As DataTable = CustomerDataSet.Tables(0)

        'display customerNo1's name
    winAccountDetails.txtTitle1.Text = Convert.ToString(datatable1.Rows(0)
(0)
    winAccountDetails.txtForenames1.Text = Convert.ToString(datatable1.Rows
(0)(1)
    winAccountDetails.txtSurname1.Text = Convert.ToString(datatable1.Rows
(0)(2)

    End If

    'display second account holder's name if they exist
    If winAccountDetails.txtCustomerNo2.Text.Length <> 0 Then

        'Open Database Connection
        FAMSDBConnection.Open()
```

```

'Returns the DataSet to it's original state
CustomerDataSet.Reset()

'Select customerNo1's name
CustomerDataAdapter.SelectCommand.CommandText = _
"SELECT title, forenames, surname FROM Customer WHERE " & _
"customerNo = '" & winAccountDetails.txtCustomerNo2.Text & "'"

'Fill CustomerDataSet
CustomerDataAdapter.Fill(CustomerDataSet)

'close database connection
FAMSDBConnection.Close()

'Fill datatable with results of query
Dim datatable2 As DataTable = CustomerDataSet.Tables(0)

'display customerNo1's name
winAccountDetails.txtTitle2.Text = Convert.ToString(datatable2.Rows(0)
(0))
winAccountDetails.txtForenames2.Text = Convert.ToString(datatable2.Rows
(0)(1))
winAccountDetails.txtSurname2.Text = Convert.ToString(datatable2.Rows
(0)(2))

End If

'display third account holder's name if they exist
If winAccountDetails.txtCustomerNo3.Text.Length <> 0 Then

'Open Database Connection
FAMSDBConnection.Open()

'Returns the DataSet to it's original state
CustomerDataSet.Reset()

'Select customerNo1's name
CustomerDataAdapter.SelectCommand.CommandText = _
"SELECT title, forenames, surname FROM Customer WHERE " & _
"customerNo = '" & winAccountDetails.txtCustomerNo3.Text & "'"

'Fill CustomerDataSet
CustomerDataAdapter.Fill(CustomerDataSet)

'close database connection
FAMSDBConnection.Close()

'Fill datatable with results of query
Dim datatable3 As DataTable = CustomerDataSet.Tables(0)

'display customerNo1's name
winAccountDetails.txtTitle3.Text = Convert.ToString(datatable3.Rows(0)
(0))
winAccountDetails.txtForenames3.Text = Convert.ToString(datatable3.Rows
(0)(1))
winAccountDetails.txtSurname3.Text = Convert.ToString(datatable3.Rows
(0)(2))

End If
Else
'Display warning message if account record is not found
MessageBox.Show("Account Not Found!", "No Account Found", MessageBoxButtons
.OK, MessageBoxIcon.Warning)
End If

Catch ex As Exception
MessageBox.Show(ex.ToString())
End Try
End Sub 'cmdViewAccount

'Randomly generates a new customer number, displays it in frmMainMenu and
'INSERTs a new customer record in Cuustomer Table of FAMSDB database
Private Sub NewCustomerNo()

```

```

Try
    'variable to hold customer number
    Dim customerNumber As Integer

    'declare variables for generating new customer number randomly
    Dim randomObject As Random = New Random

    'assign random number to Customer Number variable
    customerNumber = randomObject.Next(1, 2147483647)

    'Open Database Connection
    FAMSDBConnection.Open()

    'Returns the DataSet to it's original state
    CustomerDataSet.Reset()

    'create query to select customer details
    CustomerDataAdapter.SelectCommand.CommandText = _
    " SELECT * FROM Customer WHERE customerNo = ' " & _
    Convert.ToString(customerNumber) & "' "

    'Fill CustomerDataSet with rows resulting from query
    CustomerDataAdapter.Fill(CustomerDataSet)

    FAMSDBConnection.Close()

    'Put results of query into datatable with which to work
    Dim datatable As DataTable = CustomerDataSet.Tables(0)

    'If customer number is found in the database recursively call
    'NewCustomerNo() method
    If datatable.Rows.Count <> 0 Then

        NewCustomerNo()

    Else

        'Open Database Connection
        FAMSDBConnection.Open()

        'Create INSERT Command
        CustomerDataAdapter.InsertCommand.CommandText = _
        "INSERT INTO Customer (customerNo) VALUES (' " & _
        Convert.ToString(customerNumber) & "') "

        'Execute INSERT Command
        CustomerDataAdapter.InsertCommand.ExecuteNonQuery()

        'Close Database Connection
        FAMSDBConnection.Close()

        'Display Customer Number in TextBox in Form frmMainMenu
        txtCustomerNo.Text = Convert.ToString(customerNumber)

    End If
Catch ex As Exception
    MessageBox.Show(ex.ToString())
End Try

End Sub 'NewCustomerNo()

'Randomly generates a new account number, displays it in frmMainMenu and
'INSERTs a new record in the Account Table of the FAMSDB database
Private Sub NewAccountNo()

    Try
        'Variable to hold new account number
        Dim accountNumber As Integer

        'Declare variables for randomly generating new account number
        Dim randomObject As Random = New Random

        'Assign random number to accountNumber Variable
        accountNumber = randomObject.Next(10000000, 100000000)
    
```



```
'Open Database Connection
FAMSDBConnection.Open()

'Return the DataSet to it's original state
AccountDataSet.Reset()

'Create query to select account details
AccountDataAdapter.SelectCommand.CommandText = _
"SELECT * FROM Account WHERE sortCode = '" & _
txtSortCode.Text & "' AND accountNo = '" & txtAccountNo.Text & "'"

'Fill AccountDataSet
AccountDataAdapter.Fill(AccountDataSet)

'Close Database Connection
FAMSDBConnection.Close()

'Put results of query into datatable with which to work
Dim datatable As DataTable = AccountDataSet.Tables(0)

'If account number is found in the database recursively call
'NewAccountNo() method
If datatable.Rows.Count <> 0 Then

    NewAccountNo()

Else
    'Open Database Connection
    FAMSDBConnection.Open()

    'Create INSERT Command
    AccountDataAdapter.InsertCommand.CommandText = _
    "INSERT INTO Account(sortCode, accountNo, customerNo1, customerNo2, " & _
    "customerNo3, availableBalance, balance, overdraftLimit, " & _
    "status, openingDate, closingDate, totalInterest, charges) VALUES(" & _
    "'" & txtSortCode.Text & "', '" & Convert.ToString(accountNumber) & "', " & _
    "'', '', '', '0', '0', '0', 'Open', " & _
    "'" & Convert.ToString(DateTime.Now.ToString("d")) & "', " & "'', '0', '0') "

    'Execute INSERT Command
    AccountDataAdapter.InsertCommand.ExecuteNonQuery()

    'Close Database Connection
    FAMSDBConnection.Close()

    'Display new account number in frm MainMenu
    txtAccountNo.Text = Convert.ToString(accountNumber)

End If

Catch ex As Exception
    MessageBox.Show(ex.ToString)
End Try
End Sub 'NewAccountNo

End Class 'frmMainMenu
```