

```
'Form displays account details and provides facilities for updating them and
' carrying out transactions on accounts
'Written by Bill Blackley
'First Written 9/7/05
'Last Written 22/7/05
```

```
Public Class frmAccountDetails
    Inherits System.Windows.Forms.Form
```

```
Windows Form Designer generated code
```

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

```
'Opens instance of frmCashDeposit and passes sort code and account number
'to it
```

```
Private Sub mnuItmCashDeposit_Click(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles mnuItmCashDeposit.Click
```

```
    'Open frmCashDeposit
    Dim winCashDeposit As New frmCashDeposit
    winCashDeposit.Show()
```

```
    'Pass Sort Code and Account Number to instance of frmCashDeposit
    winCashDeposit.txtSortCode.Text = txtSortCode.Text
    winCashDeposit.txtAccountNo.Text = txtAccountNo.Text
```

```
End Sub 'mnuItmCashDeposit_Click
```

```
'Opens instance of frmChequeDeposit and passes sort code and account number
'to it
```

```
Private Sub mnuItmCheque_Click(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles mnuItmCheque.Click
```

```
    'Open frmChequeDeposit
    Dim winChequeDeposit As New frmChequeDeposit
    winChequeDeposit.Show()
```

```
    'Pass Sort Code and Account Number to instance of frmCashDeposit
    winChequeDeposit.txtSortCode.Text = txtSortCode.Text
    winChequeDeposit.txtAccountNo.Text = txtAccountNo.Text
```

```
End Sub 'mnuItmCheque_Click
```

```
'Allows the user to Post Interest to an account by adding the value of the
'account record's totalInterest attribute to it's balance and availableBalance
'attributes and then setting the totalInterest attribute to zero
```

```
Private Sub mnuItmPostInterest_Click(ByVal sender As System.Object, ByVal e As System.
    EventArgs) Handles mnuItmPostInterest.Click
```

```
    Try
        'First ask user if they really want to Post Interest
        Dim Result As DialogResult
```

```
        Result = MessageBox.Show("Do you really want to Post Interest?", "Post Interest
?", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
```

```
        If Result = DialogResult.Yes Then
```

```
            'Declare interest, balance, newBalance, available and
            'newAvailableBalance variables to hold amounts
            Dim interest, balance, newBalance, availableBalance, newAvailableBalance _
            As String
```

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

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

```
            'Create Select query
```

```
            AccountDataAdapter.SelectCommand.CommandText = _
            "SELECT balance, availableBalance, totalInterest 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)

'Now put values of totalInterest, balance and availableBalance
'into the appropriate variables with which to work
balance = Convert.ToString(datatable.Rows(0)(0))
availableBalance = Convert.ToString(datatable.Rows(0)(1))
interest = Convert.ToString(datatable.Rows(0)(2))

'Now calculate new balance and new availableBalance
newBalance = Convert.ToString(Convert.ToDecimal(balance) + Convert.
ToDecimal(interest))
newAvailableBalance = Convert.ToString(Convert.ToDecimal(availableBalance)
+ Convert.ToDecimal(interest))

'Now UPDATE Account Table of FAMSDB with new balance and new
'availableBalance

'Open Database Connection
FAMSDBConnection.Open()

'Create UPDATE Command
AccountDataAdapter.UpdateCommand.CommandText = _
"UPDATE Account SET balance = '" & newBalance & "', " & _
"availableBalance = '" & newAvailableBalance & "', " & _
"totalInterest = '0'"

'Execute UPDATE Command
AccountDataAdapter.UpdateCommand.ExecuteNonQuery()

'Close Database Connection
FAMSDBConnection.Close()

'Now INSERT transaction record
Transaction.InsertTransaction("Interest", interest, newBalance, _
txtSortCode.Text, txtAccountNo.Text)

'Now display confirmation message
MessageBox.Show("£" & interest & " interest has been posted", "Interest
Posted", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

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

End Sub 'mnuItmPostInterest_Click
'Opens instance of frmCashWithdrawal and passes sort code and account number
'to it
Private Sub mnuItmCashWithdrawal_Click(ByVal sender As System.Object, ByVal e As System
.EventArgs) Handles mnuItmCashWithdrawal.Click
    Dim winCashWithdrawal As New frmCashWithdrawal
    winCashWithdrawal.Show()

    winCashWithdrawal.txtSortCode.Text = txtSortCode.Text
    winCashWithdrawal.txtAccountNo.Text = txtAccountNo.Text
End Sub 'mnuItmCashWithdrawal_Click
'Opens instance of frmBankersDraft and passes sort code and account number
'to it
Private Sub mnuItmBankersDraft_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles mnuItmBankersDraft.Click

    Dim winBankersDraft As New frmBankersDraft
    winBankersDraft.Show()

    winBankersDraft.txtSortCode.Text = txtSortCode.Text

```

```

winBankersDraft.txtAccountNo.Text = txtAccountNo.Text
End Sub 'mnuItmBankersDraft_Click

'Opens instance of frmInterAccountTransfer and passes sort code and account number
'to it
Private Sub mnuItmInterAccountTransfer_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuItmInterAccountTransfer.Click
    Dim winInterAccountTransfer As New frmInterAccountTransfer
    winInterAccountTransfer.Show()

    winInterAccountTransfer.txtSortCode.Text = txtSortCode.Text
    winInterAccountTransfer.txtAccountNo.Text = txtAccountNo.Text

End Sub 'mnuItmInterAccountTransfer_Click

'Allows the user to deduct any charges due on the account by subtracting the
'value of the charges attribute in the Account Table from the value of the
'balance and availableBalance attributes of the account Table and also setting
'the value of the charges attribute to zero. Lastly, the method would credit
'BestBank's Bank of England A/c with the value of the charges - this is beyond
'the scope of this project
Private Sub mnuItmDeductCharges_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuItmDeductCharges.Click

    Try
        'First ask user if they really want to deduct charges
        Dim Result As DialogResult

        Result = MessageBox.Show("Do you really want to Deduct Charges on the Account?"
, "Deduct Charges?", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

        If Result = DialogResult.Yes Then

            'Declare variables for holding the amounts used in the calculations
            Dim charges, balance, newbalance, availableBalance, newAvailableBalance _
            As String

            'Open database Connection
            FAMSDBConnection.Open()

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

            'Create Select Query
            AccountDataAdapter.SelectCommand.CommandText = _
            "SELECT charges, balance, availableBalance FROM Account " & _
            "WHERE sortCode = '" & txtSortCode.Text & "' AND " & _
            "accountNo = '" & txtAccountNo.Text & "'"

            'Fill DataSet
            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)

            'Set charges, balance and availableBalance variables to results
            'of query, for working with
            charges = Convert.ToString(datatable.Rows(0)(0))
            balance = Convert.ToString(datatable.Rows(0)(1))
            availableBalance = Convert.ToString(datatable.Rows(0)(2))

            'Now calculate balances after charges have been deducted
            newbalance = Convert.ToString(Convert.ToDecimal(balance) - Convert.
ToDecimal(charges))
            newAvailableBalance = Convert.ToString(Convert.ToDecimal(availableBalance)
- Convert.ToDecimal(charges))

            'Now set newAvailableBalance to zero if it is negative
            If Convert.ToDecimal(newAvailableBalance) < 0 Then
                newAvailableBalance = "0"
            End If
        End Try
    End Sub

```

```

'Now UPDATE Account Table of FAMSDB Database

'Open Database Connection
FAMSDBConnection.Open()

'Create UPDATE Command
AccountDataAdapter.UpdateCommand.CommandText = _
"UPDATE Account SET charges = '0', balance = '" & _
newbalance & "', availableBalance = '" & newAvailableBalance & "'" & _
"WHERE sortCode = '" & txtSortCode.Text & "' AND " & _
"accountNo = '" & txtAccountNo.Text & "'"

'Execute UPDATE Command
AccountDataAdapter.UpdateCommand.ExecuteNonQuery()

'Close Database Connection
FAMSDBConnection.Close()

'Now INSERT transaction record in Transaction Table of FAMSDB
Transaction.InsertTransaction("Charges", charges, newbalance, _
txtSortCode.Text, txtAccountNo.Text)

'Code for Connecting to the Bank of England's server via SWIFTNet
'and Updating Bestbank's Bank of England account goes here

'Now display confirmation message to user
MessageBox.Show("£" & charges & " charges deducted", "Charges Deducted",
MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

Catch ex As Exception
    MessageBox.Show(ex.ToString())
End Try
End Sub 'mnuItmDeductCharges_Click
'Open instance of form frmStatements and passes the account's sort code
'and account number to it
Private Sub mnuStatements_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles mnuStatements.Click
    Dim winStatements As New frmStatements
    winStatements.Show()

    winStatements.txtSortCode.Text = txtSortCode.Text
    winStatements.txtAccountNo.Text = txtAccountNo.Text
End Sub 'mnuStatements_Click

'Opens the form frmCorrespondenceAddress and loads the A/c name and Correspondence
'address data into the form
Private Sub mnuCorrespondenceAddress_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuCorrespondenceAddress.Click

Try
    'Open form frmCorrespondenceAddress
    Dim winCorrespondenceAddress As New frmCorrespondenceAddress
    winCorrespondenceAddress.Show()

    'Open Database Connection
    FAMSDBConnection.Open()

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

    'Select required fields from Account Table
    AccountDataAdapter.SelectCommand.CommandText = _
"SELECT name, addressLine1, addressLine2, addressLine3, city, county, postCode,
sortCode, accountNo, closingDate, status " & _
"FROM Account WHERE sortCode = '" & txtSortCode.Text & "'" & _
"AND accountNo = '" & txtAccountNo.Text & "'"

    'Fill DataSet with results of query
    AccountDataAdapter.Fill(AccountDataSet)

    'Close database connection

```

```

FAMSDBConnection.Close()

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

'Now fill form with data
winCorrespondenceAddress.txtName.Text = Convert.ToString(datatable.Rows(0)(0))
winCorrespondenceAddress.txtAddressLine1.Text = Convert.ToString(datatable.Rows(0)(1))
winCorrespondenceAddress.txtAddressLine2.Text = Convert.ToString(datatable.Rows(0)(2))
winCorrespondenceAddress.txtAddressLine3.Text = Convert.ToString(datatable.Rows(0)(3))
winCorrespondenceAddress.txtCity.Text = Convert.ToString(datatable.Rows(0)(4))
winCorrespondenceAddress.txtCounty.Text = Convert.ToString(datatable.Rows(0)(5))
winCorrespondenceAddress.txtPostCode.Text = Convert.ToString(datatable.Rows(0)(6))
winCorrespondenceAddress.txtSortCode.Text = Convert.ToString(datatable.Rows(0)(7))
winCorrespondenceAddress.txtAccountNo.Text = Convert.ToString(datatable.Rows(0)(8))

'If Account has closingDate and status is "closed" disable
'all textboxes in form so they can't be edited
If Convert.ToString(datatable.Rows(0)(9)).Length <> 0 AndAlso Convert.ToString(datatable.Rows(0)(10)).Equals("Closed") Then

    winCorrespondenceAddress.txtName.Enabled = False
    winCorrespondenceAddress.txtAddressLine1.Enabled = False
    winCorrespondenceAddress.txtAddressLine2.Enabled = False
    winCorrespondenceAddress.txtAddressLine3.Enabled = False
    winCorrespondenceAddress.txtCity.Enabled = False
    winCorrespondenceAddress.txtCounty.Enabled = False
    winCorrespondenceAddress.txtPostCode.Enabled = False

End If

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

End Sub 'mnuCorrespondenceAddress_Click

'Updates the account record if the user edits the data and INSERTs a record in
'the AccountMaintenance table of the FAMSDB database
Private Sub mnuUpdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuUpdate.Click
    Try

        'Ask user if they're sure they want to update the account details
        Dim Result As DialogResult

        Result = MessageBox.Show("Are you sure you want to Update the Account record?", "Update Account?", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

        'If user responds Yes then.....
        If Result = DialogResult.Yes Then

            'If user is closing account, check that the A/c balance
            'is zero

            'First get A/c balance from FAMSDB
            'Open Database Connection
            FAMSDBConnection.Open()

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

            'Create SELECT query to get A/c balance
            AccountDataAdapter.SelectCommand.CommandText =
            "SELECT balance FROM Account WHERE sortCode = '" & _
            txtSortCode.Text & "' AND accountNo = '" & _
            txtAccountNo.Text & "'"

```

```

'Fill DataSet
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)

'Now check for zero balance
If ((cboAccountStatus.Text.Equals("Closed")) AndAlso _
(txtClosingDate.Text.Length <> 0) AndAlso _
(Not (datatable.Rows(0)(0).Equals("0")))) Then

    MessageBox.Show("Can Not Close Account Until Balance is zero", "Warning"
, MessageBoxButtons.OK, MessageBoxIcon.Warning)
    cboAccountStatus.Text = "Open"
    txtClosingDate.Text = ""
    txtBalance.Text = Convert.ToString(datatable.Rows(0)(0))
    Exit Sub

End If

'Replace double quote marks with underscore in Notes
'to prevent problems with Update command
txtNotes.Text = txtNotes.Text.Replace(ChrW(34), "_")

'Relace single quote marks with underscore in Notes
'to prevent problems with Update command
txtNotes.Text = txtNotes.Text.Replace(ChrW(39), "_")

'Set Available Balance = Overdraft Limit + Balance
txtAvailableBalance.Text = Convert.ToString(Convert.ToDecimal
(txtOverdraftLimit.Text) + Convert.ToDecimal(txtBalance.Text))

'Open Database Connection
FAMSDBConnection.Open()

'Create Update Command
AccountDataAdapter.UpdateCommand.CommandText = _
"Update Account SET productType = '" & cboProductType.Text & "', " & _
"customerNo1 = '" & txtCustomerNo1.Text & "', " & _
"customerNo2 = '" & txtCustomerNo2.Text & "', " & _
"customerNo3 = '" & txtCustomerNo3.Text & "', " & _
"status = '" & cboAccountStatus.Text & "', " & _
"closingDate = '" & txtClosingDate.Text & "', " & _
"availableBalance = '" & txtAvailableBalance.Text & "', " & _
"overdraftLimit = '" & txtOverdraftLimit.Text & "', " & _
"notes = '" & txtNotes.Text & "' " & _
"WHERE sortCode = '" & txtSortCode.Text & "' AND " & _
"accountNo = '" & txtAccountNo.Text & "'"

'Execute Update Command
AccountDataAdapter.UpdateCommand.ExecuteNonQuery()

'Close database connection
FAMSDBConnection.Close()

'Now INSERT Account Maintenance record

'Declare StaffNumber variable to hold the staffNo of the member of
'staff that is logged in
Dim StaffNumber As String

'Declare transactionNumber variable to hold the maximum
'transactionNo that the user has carried out on the account
Dim transactionNumber As String

'Open Database Connection
FAMSDBConnection.Open()

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

```

```

'Create Select query
StaffDataAdapter.SelectCommand.CommandText = _
"SELECT staffNo FROM Staff WHERE localIPAddress = '" & _
frmLogin.GetIPAddress() & "'"

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

'Close Database Connection
FAMSDBConnection.Close()

'Put the results of the query into a datatable with which to work
Dim datatable1 As DataTable = StaffDataSet.Tables(0)

'Put staffNo into StaffNumber variable
StaffNumber = Convert.ToString(datatable1.Rows(0)(0))

'Now select maximum transactionNo carried out by current user

'Open Database Connection
FAMSDBConnection.Open()

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

'Create Select query
AccountMaintenanceDataAdapter.SelectCommand.CommandText = _
"SELECT COUNT(*) FROM AccountMaintenance WHERE sortCode = '" & _
txtSortCode.Text & "' AND accountNo = '" & txtAccountNo.Text & "' " & _
"AND staffNo = '" & StaffNumber & "'"

'Fill AccountMaintenanceDataSet
AccountMaintenanceDataAdapter.Fill(AccountMaintenanceDataSet)

'close Database Connection
FAMSDBConnection.Close()

'Put result of query into datatable with which to work
Dim datatable2 As DataTable = AccountMaintenanceDataSet.Tables(0)

'Put highest transactionNo into transactionNumber variable
transactionNumber = Convert.ToString(datatable2.Rows(0)(0))

'Now INSERT record into AccountMaintenance Table of FAMSDB database

'Open Database Connection
FAMSDBConnection.Open()

'Create INSERT Command
AccountMaintenanceDataAdapter.InsertCommand.CommandText = _
"INSERT INTO AccountMaintenance VALUES('" & txtSortCode.Text & "', " & _
txtAccountNo.Text & "', " & _
StaffNumber & "', " & _
Convert.ToString(Convert.ToInt32(transactionNumber) + 1) & "', " & _
Convert.ToString(DateTime.Now.ToString("d")) & "', " & _
"Updated'"

'Execute INSERT Command
AccountMaintenanceDataAdapter.InsertCommand.ExecuteNonQuery()

'Close Database Connection
FAMSDBConnection.Close()

'Display confirmation message
MessageBox.Show("Account Record has been updated", "Account Updated",
MessageBoxButtons.OK, MessageBoxIcon.Information)

End If

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

```

```
End Class 'frmAccountDetails
```