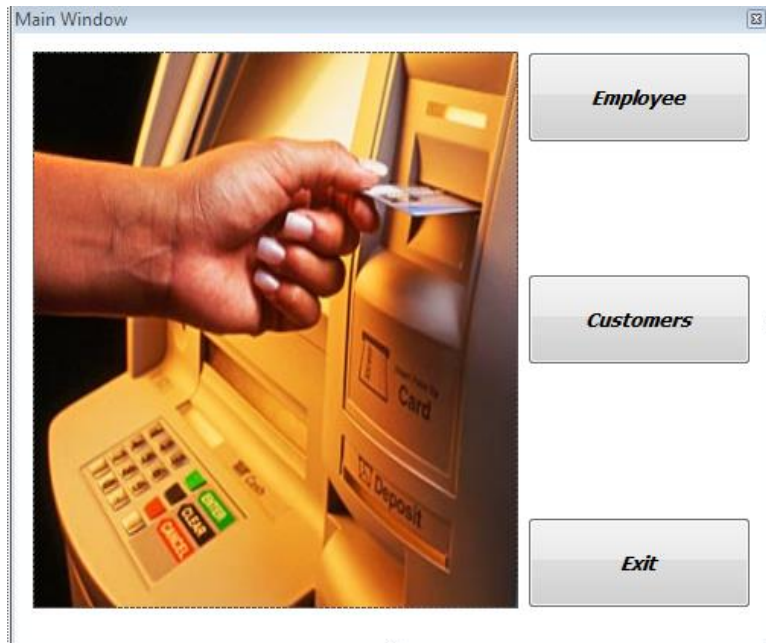


بسم الله الرحمن الرحيم

## شرح مشروع ATM

سوف نبدأ إن شاء الله شرح مشروع ATM الذى تم فيه إستخدام ADO.Net ،  
والتي تعتبر من أحدث المكتبات الخاصة بالتعامل مع قواعد البيانات ومميزاتها الكبيرة  
عن مكتبة ADO Com .

أولا : الشاشة الرئيسية :



وتعتبر هذه الشاشة هي الشاشة الرئيسية ، و تحتوى على ثلاثة أزرار ، زر يقوم  
بالدخول على شاشة تسجيل العملاء والتي يدخل عليها الموظف ، وشاشة أخرى  
للدخول على شاشة حركة السحب والإيداع والتي يدخل عليها العميل ، والزر الأخير  
للخروج من البرنامج .

ثانياً : الشاشة الخاصة بالموظفين :

عند الدخول على شاشة تسجيل العميل من قبل الموظف ، فإن المستخدم يجب أن يدخل UserName & Password الخاص به حتى يستطيع الدخول .



هذه الشاشة هي شاشة تسجيل الدخول الخاصة بالموظفين ، حتى يستطيع الموظف الدخول الى لوحة تسجيل العملاء ، وأيضاً حتى لا يجب لأي مستخدم الدخول على شاشة تسجيل العملاء .

```
Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"

Dim Comm As New SqlCommand
Comm.CommandText = "SELECT EmpUserName, EmpPass FROM EmpUser WHERE (EmpUserName = @Uname) AND (EmpPass = @Pass)"

Dim Prams(1) As SqlParameter

Prams(0) = New SqlParameter("@Uname", TxtUname.Text)
Prams(1) = New SqlParameter("@Pass", TxtPass.Text)

Comm.Parameters.AddRange(Prams)

Comm.Connection = Conn
Conn.Open()

Dim rd As SqlDataReader = Comm.ExecuteReader
If rd.HasRows Then
    CustomersForm.Show()
    rd.Read()
    CustomersForm.Text = CustomersForm.Text + " _ " + rd("EmpUserName")
    Me.Close()
Else
    MsgBox("Invalid User Name Or Password", MsgBoxStyle.Critical)
End If

Conn.Close()
```

وهذا هو الكود الخاص بزر تسجيل الدخول للموظفين ، حيث يقوم بحجز نسخة من الاتصال بقاعدة البيانات ، وكتابة نص الاتصال بين البرنامج وقاعدة البيانات ، ثم نقوم بحجز نسخة من Command المسئولة عن الاتصال بالجدول وخصائصها من حيث البحث والتسجيل والحذف والتعديل ، وذلك من خلال هذه الكوماند ، ثم نستخدم بعد ذلك البارامتر وهي تعتبر مصفوفة لإتصال المتغيرات بالحقول فى قاعدة البيانات ، وفى هذا الكود من أجل عملية القراءة ، ثم بعد ذلك نقوم بفتح قاعدة البيانات ، وتنفيذ هذا الأمر ، وسيتم قراءة البيانات من الجدول ، وفى حالة وجود سجلات وهو بالتأكيد سيكون سجل واحد فإنه يفتح شاشة الموظفين الخاصة بتسجيل العملاء ، غير ذلك فإنه يظهر رسالة إعتراض بأن اسم المستخدم أو كلمة المرور غير صحيحة .

ثالثاً : الشاشة الخاصة بالعملاء :

عند الدخول على شاشة العملاء وهي حركات السحب والإيداع ، تظهر شاشة لإدخال UserName & Password ، حتى يستطيع الدخول .

وهذه هي شاشة Login الخاصة بالعملاء ، حيث يدخل العميل اسم المستخدم الخاص به وأيضاً كلمة المرور ، حتى يستطيع الدخول ويقوم بعمليات السحب والإيداع كما شاء .

```

Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"

Dim Comm As New SqlCommand
Comm.CommandText = "SELECT UserName, Pass, ID FROM CustomerData WHERE (UserName = @Uname) AND (Pass = @Pass)"

Dim Prams(1) As SqlParameter

Prams(0) = New SqlParameter("@Uname", TxtUName.Text)
Prams(1) = New SqlParameter("@Pass", TxtPass.Text)

Comm.Parameters.AddRange(Prams)

Comm.Connection = Conn
Conn.Open()

Dim rd As SqlDataReader = Comm.ExecuteReader
If rd.HasRows Then
    Actions.Show()
    rd.Read()
    Actions.Text = Actions.Text + " _ " + rd("UserName")
    Actions.TxtDepID.Text = rd("ID")
    Actions.TxtWithID.Text = rd("ID")
    Actions.LblUserName.Text = rd("UserName")
    Actions.LblBalance.Text = SettingDB.GetAllBalance(TxtUName.Text)
    Me.Close()
Else
    MsgBox("Invalid User Name Or Password", MsgBoxStyle.Critical)
End If

Conn.Close()

```

وهذا الكود الخاص بزر Login الخاصة بالعملاء ، حيث يقوم بحجز نسخة من الاتصال بقاعدة البيانات ، وكتابة نص الاتصال بين البرنامج وقاعدة البيانات ، ثم نقوم بحجز نسخة من Command المسئولة عن الاتصال بالجدول وخصائصها من حيث البحث والتسجيل والحذف والتعديل ، وذلك من خلال هذه الكوماند ، ثم نستخدم بعد ذلك البارامتر وهي تعتبر مصفوفة لإتصال المتغيرات بالحقول في قاعدة البيانات ، وفي هذا الكود من أجل عملية القراءة ، ثم بعد ذلك نقوم بفتح قاعدة البيانات ، وتنفيذ هذا الأمر ، وسيتم قراءة البيانات من الجدول ، وفي حالة وجود سجلات وهو بالتأكيد سيكون سجل واحد فإنه يفتح شاشة العملاء الخاصة بحركة السحب والإيداع وايضاً يعمل على إظهار الرصيد الكلى للعميل عند دخوله كما سنراه لاحقاً ، غير ذلك فإنه يظهر رسالة إعتراض بأن اسم المستخدم أو كلمة المرور غير صحيحة .

رابعاً : شاشة تسجيل العملاء :

The screenshot shows a Windows-style application window titled "Customers Data". Inside the window is a form with the following fields and labels:

- ID :
- Name :  Job :
- Phone :  Mobile :
- Email :  City :
- Address :
- User Name :  Password :

At the bottom of the form, there are two groups of buttons:

- Left group: Save, Search, Delete
- Right group: Edit, Show All, Back

On the right side of the form, there is a small icon showing three stylized people (two men and one woman) in a group.

وتعتبر هذه الشاشة هي من أهم الشاشات في البرنامج ، حيث يتم تسجيل بيانات العملاء من خلالها ، وتحتوى على عدة صلاحيات ، وهي حفظ عميل جديد ، والبحث عن عميل موجود مسبقاً سواءً لغرض التعديل أو الحذف ، وأيضاً صلاحية إظهار جميع العملاء المسجلين في قاعدة البيانات .

## ١- زر الحفظ Save :

```

-----
Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"
Dim Comm As New SqlCommand
Comm.Connection = Conn

'-----

Comm.CommandText = "Select * From CustomerData Where UserName = '" + TxtUserName.Text + "'"

Conn.Open()
Dim Rd As SqlDataReader = Comm.ExecuteReader()

Do While Rd.Read()
    If Rd.HasRows Then
        MsgBox(" User Name Is Really Existing , Insert Another One", MsgBoxStyle.Critical, "Error")
        Conn.Close()
        Exit Sub
    End If
Loop

Conn.Close()

'-----

```

يقوم هذا الكود بحجز نسخة من الاتصال بقاعدة البيانات ، وكتابة نص الاتصال بين البرنامج وقاعدة البيانات ، ثم نقوم بحجز نسخة من Command الاتصال بالجدول ، وفي هذا الكود من أجل عملية القراءة ، ثم بعد ذلك نقوم بفتح قاعدة البيانات ، وتنفيذ هذا الأمر ، وسيتم قراءة البيانات من الجدول ، وغرض ذلك الكود هو جعل الجدول في قاعدة البيانات غير قابل لحدوث تكرار لإسم المستخدم ، لأن إسم المستخدم يعتبر Primary Key فى الجدول ، وذلك تجنباً لحدوث أى أخطاء قد تظهر أثناء استخدام البرنامج ، وفى حالة وجود اسم مستخدم فى الجدول فإنه يظهر رسالة تحذيرية بأنه وجد اسم مستخدم بنفس الاسم ويجب إختيار واحد آخر ، وفى حالة عكس ذلك فإنه سوف ينتقل للكود الآتى :

```
Comm.CommandText = "INSERT INTO CustomerData (Name, Job, Phone, Mobile, Email, Address, UserName, Pass, City) VALUES (@Name,@Job,@Phone,@Mob,@Mail,@Adrs,@Uname,@Pass,@City)"
Dim Prams(8) As SqlParameter
Prams(0) = New SqlParameter("@Name", TxtName.Text)
Prams(1) = New SqlParameter("@Job", TxtJob.Text)
Prams(2) = New SqlParameter("@Phone", TxtPhone.Text)
Prams(3) = New SqlParameter("@Mob", TxtMob.Text)
Prams(4) = New SqlParameter("@Mail", TxtMail.Text)
Prams(5) = New SqlParameter("@Adrs", TxtAdrs.Text)
Prams(6) = New SqlParameter("@Uname", TxtUserName.Text)
Prams(7) = New SqlParameter("@Pass", TxtPwd.Text)
Prams(8) = New SqlParameter("@City", TxtCity.Text)
Comm.Parameters.AddRange(Prams)
Conn.Open()
Dim Es As Integer = Comm.ExecuteNonQuery()
If Es > -1 Then
    MsgBox("Customer Data Saved Successfully", MsgBoxStyle.Information, "Done")

    TxtAdrs.Text = ""
    TxtCity.Text = ""
    TxtID.Text = ""
    TxtJob.Text = ""
    TxtMail.Text = ""
    TxtMob.Text = ""
    TxtName.Text = ""
    TxtPhone.Text = ""
    TxtUserName.Text = ""
    TxtPwd.Text = ""
Else
    MsgBox("Some Error Occured , Please Try Again", MsgBoxStyle.Critical, "Error")
End If
Conn.Close()
```

وفى هذا الكود يقوم بتحديد الأمر النصى الذى سوف ينفذ ، ثم نقوم بحجز بارامتر مصفوفة حتى نقوم بالاتصال بين البرنامج والحقول بالجدول ، ثم بعد ذلك نقوم بفتح قاعدة البيانات وتنفيذ الأمر وهو تسجيل بيانات العملاء ، فإذا نجحت العملية فإنه يظهر رسالة تأكيد عملية الحفظ وتنظيف التيكستات ، غير ذلك فإنه يخبره بأنه ثمة شىء خطأ حدث مع المحاولة مرة أخرى .

## ٢- زر البحث Search :

```
Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"

Dim Comm As New SqlCommand
Comm.Connection = Conn

Dim SearchUserName As String = InputBox("Input User Name :", "Search")
Comm.CommandText = "Select * From CustomerData Where UserName = '" + SearchUserName + "'"
Conn.Open()

Dim Rd As SqlDataReader = Comm.ExecuteReader()
Rd.Read()

If Rd.HasRows Then

    TxtID.Text = Rd("ID")
    TxtName.Text = Rd("Name")
    TxtJob.Text = Rd("Job")
    TxtPhone.Text = Rd("Phone")
    TxtMob.Text = Rd("Mobile")
    TxtMail.Text = Rd("Email")
    TxtAdrs.Text = Rd("Address")
    TxtUserName.Text = Rd("UserName")
    TxtPwd.Text = Rd("Pass")
    TxtCity.Text = Rd("City")

    Selected = Rd("UserName")
Else

    MsgBox("Customer User Name Not Found", MsgBoxStyle.Critical, "Error")

End If
Conn.Close()
```

ويقوم بعمل استدعاء لبيانات عميل معين ، وفي هذا الكود يقوم بالبحث عن اسم المستخدم الخاص بالعميل الذي سوف يدخله الموظف حتى تظهر البيانات ، ثم نقوم بحجز نسخة من القارئ حتى يتم قراءة البيانات من قاعدة البيانات ، وفي حالة وجود سجلات وهو بالتأكيد سجل واحد فإنه يقوم بعرض البيانات على الشاشة ، وفي حالة عكس ذلك فإنه يقول له أن اسم المستخدم غير موجود .

### ٣- زر الحذف Delete :

```
Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"
Dim Comm As New SqlCommand
Comm.Connection = Conn

Comm.CommandText = "DELETE FROM CustomerData Where UserName = '" + Selected + "'"

Conn.Open()

Dim Es As Integer = Comm.ExecuteNonQuery()

If Es > -1 Then
    MsgBox("Customer Data Deleted Successfully", MsgBoxStyle.Information, "Done")
    TxtAdrs.Text = ""
    TxtCity.Text = ""
    TxtID.Text = ""
    TxtJob.Text = ""
    TxtMail.Text = ""
    TxtMob.Text = ""
    TxtName.Text = ""
    TxtPhone.Text = ""
    TxtUserName.Text = ""
    TxtPwd.Text = ""

    Comm.CommandText = "Delete From WithdrawalsData Where CustomerUname = '" + Selected + "'"
    Dim Dp As Integer = Comm.ExecuteNonQuery()
    Comm.CommandText = "Delete From DepositsData Where CustomerUname = '" + Selected + "'"
    Dim Wi As Integer = Comm.ExecuteNonQuery()
Else
    MsgBox("Some Error Occured , Try Acaain", MsgBoxStyle.Critical, "error")
End If

Conn.Close()
```

ويقوم بعمل حذف لبيانات عميل معين ، وفي هذا الكود نقوم بحجز نسخة من قاعدة البيانات والكوماند المسؤول عن جمل Select ، وفي حالة وجود سجلات وهو بالتأكيد سجل واحد فإنه يقوم بحذف البيانات من الجدول وأيضاً حذف حركات السحب والإيداع لهذا العميل ثم يقوم بتنظيف التكتستات ، وفي حالة عكس ذلك فإنه يقول له أنه قد حدث خطأ ما مع المحاولة مرة أخرى .



## ٤- زر التعديل Edit :

```
Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"

Dim Comm As New SqlCommand
Comm.Connection = Conn

Comm.CommandText = "UPDATE CustomerData SET Name = @Name, Job = @Job, Phone = @Phone, Mobile = @Mob, Email = @Mail, Address = @Adrs, UserName = @UName, Pass = @Pass, City = @City"

Dim Prams(8) As SqlParameter
Prams(0) = New SqlParameter("@Name", TxtName.Text)
Prams(1) = New SqlParameter("@Job", TxtJob.Text)
Prams(2) = New SqlParameter("@Phone", TxtPhone.Text)
Prams(3) = New SqlParameter("@Mob", TxtMob.Text)
Prams(4) = New SqlParameter("@Mail", TxtMail.Text)
Prams(5) = New SqlParameter("@Adrs", TxtAdrs.Text)
Prams(6) = New SqlParameter("@UName", TxtUserName.Text)
Prams(7) = New SqlParameter("@Pass", TxtPwd.Text)
Prams(8) = New SqlParameter("@City", TxtCity.Text)
Comm.Parameters.AddRange(Prams)

Conn.Open()
```

هذا الزر المسؤول عن عملية التعديل حيث يقوم بحجز نسخة من الاتصال بقاعدة البيانات ، وكتابة نص الاتصال بين البرنامج وقاعدة البيانات ، ثم نقوم بحجز نسخة من Command المسنولة عن الاتصال بالجدول وخصائصها من حيث البحث والتسجيل والحذف والتعديل ، وذلك من خلال هذه الكوماند ، ثم نستخدم بعد ذلك البارامتر وهي تعتبر مصفوفة لإتصال المتغيرات بالحقول في قاعدة البيانات ، وفي هذا الكود من أجل عملية التعديل ، ثم بعد ذلك نقوم بفتح قاعدة البيانات ، وتنفيذ هذا الأمر ، وسيتم تعديل البيانات في الجدول :

```
Dim Es As Integer = Comm.ExecuteNonQuery()

If Es > -1 Then
    MsgBox("Customer Data Updated Successfully", MsgBoxStyle.Information, "Done")
    TxtAdrs.Text = ""
    TxtCity.Text = ""
    TxtID.Text = ""
    TxtJob.Text = ""
    TxtMail.Text = ""
    TxtMob.Text = ""
    TxtName.Text = ""
    TxtPhone.Text = ""
    TxtUserName.Text = ""
    TxtPwd.Text = ""
Else
    MsgBox("Some Error Occured , Try Again", MsgBoxStyle.Critical, "error")
End If

Conn.Close()

End Sub
```

وبعد عملية التعديل يقوم بتنظيف التيكستات ، وفي حالة خطأ يقول له ذلك .

## ٥- الزر Show All :

وهذا الزر مسؤول عن إظهار شاشة تقوم بعرض بيانات كل العملاء المسجلين في قاعدة البيانات .

## خامساً : شاشة عرض العملاء :

	Name	Job	Phone	Mobile	Email	Address	UserName	Pass	City
▶	Abdo Dwidar	Developer	25020347	01145758353	Abdelrahman_d...	Helwan - Egypt	Abdo.97	123456	Helwan
	Alaa Abelaziz	E Teacher	3535431	0128565600	AlaaZizo@yahoo...	Rayel St	AlaaE	nada	Helwan
	ali Zain	Student	54646464	1115565664	ali.ali77@yahoo...	tebin - Helwan	Ali80	ali	Helwan
	Amr Mousa	Programer	31354351	01150002151	AmrMousa@agc...	Rehab City - Cairo	Amr	000	Cairo
	Mohamed Rizk	Teacher	646546	011266987	Hamadarizk@Ho...	Helwam Town	anabe4	1994	Helwan
	Bassem Ata	Seller	3135351	01002654789	Bassem.Ata199...	Helwan - Egypt	BasemAta	1111	Helwan
	Saleh Montasser	Engineer	2131351	0102365489	SalehMontasser...	Masr ElGidida - E...	Eng96	saleh	Cairo
	Kareem Emad	Accountant	34435354	0111251646	Kmoemad@msn...	Domyat El gidida...	Kmo	pes13	Demiatta
	Eslam	Student	6654561553	01110325489	Eslam_Salloma9...	Giza - Egypt	Saloma	2222	Giza
	Mostafa Dwidar	Accountant	5435135	0114000000	Sasa Dwidar@y...	Tersa - Elharam	Sasa90	nana	Giza

وهذه الشاشة مسؤولة عن عرض بيانات كل العملاء ، وهذا هو الكود الخاص بها :

```
Private Sub ShowAll_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    GridResult.Rows.Clear()

    Dim Conn As New SqlConnection
    Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"

    Dim Comm As New SqlCommand
    Comm.Connection = Conn

    Comm.CommandText = "Select * From CustomerData"

    Conn.Open()

    Dim Rd As SqlDataReader = Comm.ExecuteReader()

    Do While Rd.Read()
        GridResult.Rows.Add(Rd("Name"), Rd("Job"), Rd("Phone"), Rd("Mobile"), Rd("Email"), Rd("Address"), Rd("UserName"), Rd("Pass"), Rd("City"))
    Loop

    Conn.Close()
```

أولا سوف يقوم بتنظيف كل السجلات الموجودة في أداة العرض GridResult وذلك منعاً لأي عملية تكرار ، ثم يقوم بقراءة كل السجلات من الجدول من خلال SqlDataReader ، وبعد ذلك يتم عرض كل السجلات على أداة العرض .

## سادساً : شاشة Actions :

وهى الشاشة المسؤولة عن حركة السحب والإيداع ، والتي يدخل عليها العميل حتى يقوم بهذه العمليات .

The screenshot shows a web application window titled "Actions". At the top, there is a section "Choose Your Action :" with two radio buttons: "WithDrwals" (selected) and "Deposits". Below this, there are two main panels. The left panel is for "WithDrwals" and contains fields for "ID :", "Date :" (with a calendar icon and the value "2015/08/29"), "Details :", and "Amount :". At the bottom of this panel is a button labeled "WithDrwals". The right panel is for "Deposits" and contains similar fields: "ID :", "Date :" (with a calendar icon and the value "2015/08/29"), "Details :", and "Amount :". At the bottom of this panel is a button labeled "Deposit". At the bottom left of the window, it says "Current User : Customer UserName" and "Balance : Your Balance". At the bottom right is a button labeled "Back".

وتعتبر هذه الشاشة هى من أهم الشاشات لدى العميل ، حيث تنقسم إلى جزئين الجزء الأول وهو الجزء الخاص بحركة الايداع Deposits ، والتي يقوم العميل بوضع مبلغ من المال فى حسابه الشخصى ، أما الجزء الثانى وهو المسؤول عن حركة السحب ، والتي يقوم العميل من خلالها بسحب مبلغ من المال .

## ١ - الجزء الخاص بحركة الإيداع Deposits :

```
Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"
Dim Comm As New SqlCommand
Comm.Connection = Conn

Comm.CommandText = "INSERT INTO DepositsData (CustomerUname, CustomerID, Date, Details, Amount) VALUES (@CUName,@CID,@Date,@Details,@Amount)"

Dim Prams(4) As SqlParameter
Prams(0) = New SqlParameter("@CID", TxtDepID.Text)
Prams(1) = New SqlParameter("@Date", DTPDeposits.Text)
Prams(2) = New SqlParameter("@Details", TxtDepDetails.Text)
Prams(3) = New SqlParameter("@Amount", TxtDepAmount.Text)
Prams(4) = New SqlParameter("@CUName", LblUserName.Text)
Comm.Parameters.AddRange(Prams)

Conn.Open()

Dim Es As Integer = Comm.ExecuteNonQuery()
If Es > -1 Then
    MsgBox("Deposit Action Done Successfully", MsgBoxStyle.Information, "Done")

    TxtDepAmount.Text = ""
    TxtDepDetails.Text = ""

    LblBalance.Text = SettingDB.GetAllBalance(LblUserName.Text)
Else
    MsgBox("Some error Occured , Try Again", MsgBoxStyle.Critical, "Error")
End If

Conn.Close()
```

وفى هذا الكود يقوم بتحديد الأمر النصى الذى سوف ينفذ ، ثم نقوم بحجز بارامتر مصفوفة حتى نقوم بالاتصال بين البرنامج والحقول بالجدول ، ثم بعد ذلك نقوم بفتح قاعدة البيانات وتنفيذ الأمر وهو تسجيل حركة الإيداع ، فإذا نجحت العملية فإنه يظهر رسالة تأكيد عملية الحفظ وتنظيف التيكستات ويقوم بحساب الرصيد الحالى بعد عملية الإيداع ، غير ذلك فإنه يخبره بأنه ثمة شىء خطأ حدث مع المحاولة مرة أخرى .

## ٢- الجزء الخاص بعملية السحب WithDrawals :

```
Dim AllBalance As Decimal = LblBalance.Text
Dim WithDrawalsBalance As Decimal = TxtWithAmount.Text

If WithDrawalsBalance > AllBalance Then
    MsgBox("Sorry , Your Balance Is Not Enough For WithDrawals", MsgBoxStyle.Critical, "Error")
    Exit Sub
End If
```

يعتبر هذا الكود هو أهم أمر في حركة السحب ، حيث يقوم البرنامج بحساب رصيد العميل الحالي (المتبقى) ، فإذا ادخل العميل مبلغ أكبر من رصيده الكلي فإنه لن يسمح له بعملية السحب لعدم وجود رصيد كافى لعملية السحب ، أما إذا كان أقل من الرصيد الكلي فإنه يسمح له بعملية السحب دون أى اعتراض .

```
Dim Conn As New SqlConnection
Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"
Dim Comm As New SqlCommand
Comm.Connection = Conn

Comm.CommandText = "INSERT INTO WithDrawalsData (CustomerName, CustomerID, Date, Details, Amount) VALUES (@CUName,@CID,@Date,@Details,@Amount)"

Dim Prams(4) As SqlParameter
Prams(0) = New SqlParameter("@CID", TxtWithID.Text)
Prams(1) = New SqlParameter("@Date", DTPWith.Text)
Prams(2) = New SqlParameter("@Details", TxtWithDetails.Text)
Prams(3) = New SqlParameter("@Amount", TxtWithAmount.Text)
Prams(4) = New SqlParameter("@CUName", LblUserName.Text)
Comm.Parameters.AddRange(Prams)

Conn.Open()

Dim Es As Integer = Comm.ExecuteNonQuery()
If Es > -1 Then
    MsgBox("withDrawal Action Done Successfully", MsgBoxStyle.Information, "Done")

    TxtWithAmount.Text = ""
    TxtWithDetails.Text = ""

    LblBalance.Text = SettingDB.GetAllBalance(LblUserName.Text)
Else
    MsgBox("Some error Occured , Try Again", MsgBoxStyle.Critical, "Error")
End If

Conn.Close()
```

وفى هذا الكود يقوم بتحديد الأمر النصى الذى سوف ينفذ ، ثم نقوم بحجز بارامتر مصفوفة حتى نقوم بالاتصال بين البرنامج والحقول بالجدول ، ثم بعد ذلك نقوم بفتح قاعدة البيانات وتنفيذ الأمر وهو تسجيل حركة السحب ، فإذا نجحت العملية فإنه يظهر رسالة تأكيد عملية الحفظ وتنظيف التكدسات ويقوم بحساب الرصيد الحالى بعد عملية السحب ، غير ذلك فإنه يخبره بأنه ثمة شئ خطأ حدث مع المحاولة مرة أخرى .

وأخيراً :

سوف نقوم بشرح تفصيلي لكيفية حساب الرصيد المتبقى للعميل وظهوره على الشاشة واستخدامه ايضاً في البرنامج ، وذلك من خلال عمل Class تسمى SettingDB والتي تحتوى على وظيفة لجلب مجموع المبالغ التي أودعها ، ووظيفة أخرى لجلب مجموع المبالغ التي سحبها ، ووظيفة ثالثة تقوم بعملية طرح بين القيمتين واخراج ناتج العملية هو الرصيد الكلى .

أولاً : الوظيفة GetDepositBalance : والتي تقوم بحساب مبالغ الشخص التي أودعها في البنك :

```
Public Shared Function GetDepositsBalance(ByVal CustomerUname As String) As Decimal
Try
    Dim Conn As New SqlConnection
    Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"
    Dim Comm As New SqlCommand
    Comm.Connection = Conn

    Comm.CommandText = "select SUM(Amount) as [DepBalance] From DepositsData where CustomerUname = '" + CustomerUname + "'"

    Conn.Open()

    Dim Reader As SqlDataReader = Comm.ExecuteReader()
    Reader.Read()

    If Reader.HasRows Then
        Return Reader("DepBalance")
    Else
        Return 0
    End If

Catch ex As Exception
    Return 0
End Try
End Function
```

هنا في هذه الوظيفة نقوم بقراءة السجل الخاص بالعميل الذى سيتم اختياره من قبل المستخدم عن طريق الكود ، من خلال اسم المستخدم الخاص به وسيتم البحث عنه في الجدول بناء عليه ، وبعدها يقوم بحساب كل المبالغ التي أودعها هذا العميل في البنك وبعد ذلك يتم استخراج هذه القيمة ، وفي حالة عدم وجود أى سجلات سوف يرجع القيمة بـ Null وحدوث خطأ ، فتجنباً لهذه المشكلة تم استخدام Try & Catch ، وذلك في حالة حدوث خطأ سوف ترجع القيمة بصفر أى لا يوجد عمليات إيداع ولا رصيد .

ثانياً : الوظيفة **GetWithDrawalsBalance** : والتي تقوم بحساب  
وجمع مبالغ العميل التي سحبها من البنك :

```
Public Shared Function GetWithDrawalsBalance(ByVal CustomerName As String) As Decimal

Try

    Dim Conn As New SqlConnection
    Conn.ConnectionString = "Data Source=.;Initial Catalog=ATMDB;Integrated Security=True"
    Dim Comm As New SqlCommand
    Comm.Connection = Conn

    Comm.CommandText = "select SUM(Amount) as [WithBalance] From WithDrawalsData where CustomerName = '" + CustomerName + "'"

    Conn.Open()

    Dim Reader As SqlDataReader = Comm.ExecuteReader()
    Reader.Read()

    If Reader.HasRows Then
        Return Reader("WithBalance")
    Else
        Return 0
    End If

Catch ex As Exception

    Return 0

End Try
```

هنا في هذه الوظيفة نقوم بقراءة السجل الخاص بالعميل الذي سيتم اختياره من قبل المستخدم عن طريق الكود ، من خلال اسم المستخدم الخاص به وسيتم البحث عنه في الجدول بناء عليه ، وبعدها يقوم بحساب كل المبالغ التي سحبها هذا العميل من البنك وبعد ذلك يتم استخراج هذه القيمة ، وفي حالة عدم وجود أي سجلات سوف يرجع القيمة بـ Null وحدوث خطأ ، فتجنباً لهذه المشكلة تم استخدام Try & Catch ، وذلك في حالة حدوث خطأ سوف ترجع القيمة بصفر أي لا يوجد عمليات سحب .

ثالثاً : الوظيفة **GetAllBalance** : والتي تقوم بعملية طرح بين قيمتى الایداع والسحب واستخراج الرصيد المتبقى للعميل :

```
Public Shared Function GetAllBalance(ByVal CustomerUname As String) As Decimal  
  
    Dim ResultBalance As Decimal  
  
    ResultBalance = GetDepositsBalance(CustomerUname) - GetWithdrawalsBalance(CustomerUname)  
  
    Return ResultBalance  
  
End Function
```

كما نرى فى الكود ، فإنه يتم الطرح بين القيمة التى ترجع من عملية الایداع والقيمة التى ترجع من عملية السحب ، واستخراج الناتج ، لكى تستخدم فى الكود ويتم عرضها فى شاشة العميل .