Password Checker

These guidelines show how implement password checker feature in your applications using tools available in .NET platform.

Time needed to accomplish this task: 20 - 30 minutes

GUI

- Create a new VB.NET project "Windows Form Application", change the project name and path according to your preferences.

- Create a new windows form (if it is not created by default), name it "frmPwdChkr", and change its caption/text to "Password Checker"

- In "frmPwdChkr" form, create 1 textbox, 2 labels, and 1 ProgressBar and change their properties as follows:

Old Name	New Name	Caption/Text	Other Properties
TextBox1	txtPwd		
Label1	lblPassword	Password:	
Label2	lblProgressbar	Password Strength	
ProgressBar1	pbStrength		

- By the end of this step you should have something looks like the following picture:
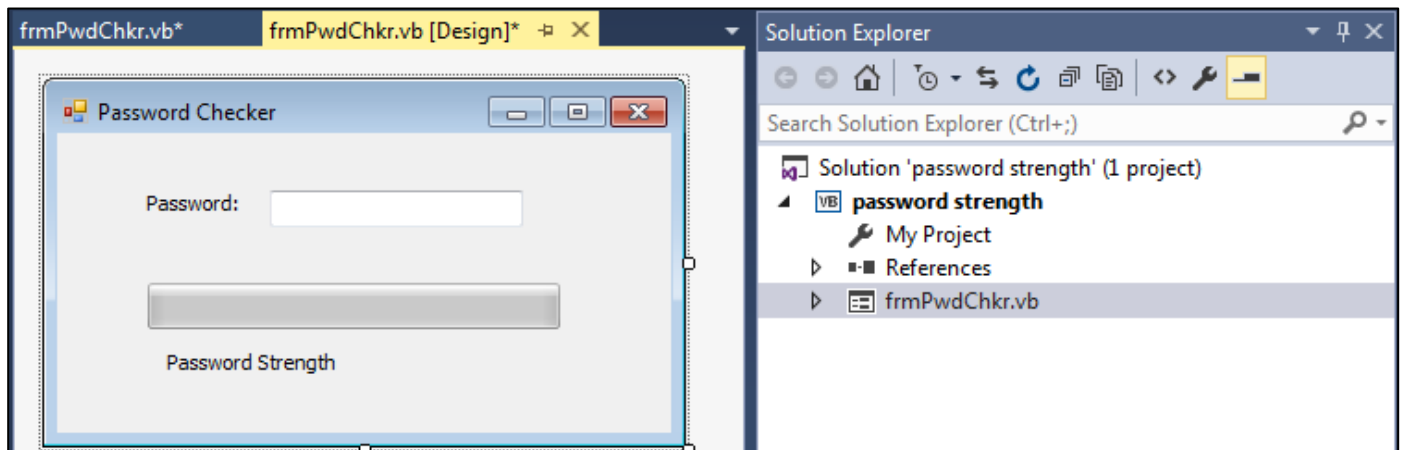
Now… Let's do some coding…

Coding

- At the beginning, let's define an array of string type to store strength levels to be used in the application. Password strength levels used in this example are: `"Invalid"`, `"Very Weak"`, `"Weak"`, `"Better"`, `"Medium"`, `"Strong"`, `"Perfect"`. To do so, write the following line

```
Private StrengthWords() As String = {"Invalid", "Very Weak", "Weak", "Better",
"Medium", "Strong", "Perfect"}
```

Calculate Password Strength:

To calculate the strength of the password entered by the user in the "txtPwd" textbox, a separate sub function `CalculateMeter()` is created. In this sub function, the password entered in txtPwd textbox will be checked for complexity using scores. This sub function will be called whenever any character is entered to txtPwd textbox. Originally, the score is zero (0), when txtPwd textbox is empty. The score is calculated as follows:

- Score is incremented by 1 if the password contains any small letter character.
- Score is incremented by 1 if the password contains any capital letter character.
- Score is incremented by 1 if the password contains any number.
- Score is incremented by 1 if the password contains any special character.
- Score is incremented by 1 if the password length is more than 6 characters
- Score is incremented by 1 if the password length is more than 10 characters
- Score is incremented by 1 if the password length is more than 15 characters

To do so, write the following code as the "`CalculateMeter`" sub:

```
Sub CalculateMeter()
    Dim score As Integer
    Dim password As String = txtPwd.Text

    If (password.Length > 6) Then score += 1 'Length more than 6
    If System.Text.RegularExpressions.Regex.IsMatch(password, "[a-z]") And
System.Text.RegularExpressions.Regex.IsMatch(password, "[A-Z]") Then
        score += 1 'upper and lower case
    End If
    If System.Text.RegularExpressions.Regex.IsMatch(password, "\d+") Then
        score += 1 'number
    End If
    If System.Text.RegularExpressions.Regex.IsMatch(password, "[,!,@,#,$,%,^,&,*,?,_,~,-
,/,"",]") Then
        score += 1 'special character
    End If
    If (password.Length >= 10) Then score += 1 'length more than 9
    If (password.Length > 15) Then score += 1 'length more than 15

    pbStrength.Value = score / 6 * 100 'finding percentage to increase
    lblProgressbar.Width = 50 * score 'label width is not auto so seeting it to show color
amount
    lblProgressbar.Text = StrengthWords(score) 'Getting strength word from string array
declarred above
    lblProgressbar.TextAlign = ContentAlignment.MiddleCenter 'alignning to center can be done
one time in design
    lblProgressbar.BackColor = GetColor(score) 'Getting color and setting
    pbStrength.ForeColor = GetColor(score) 'does not work unless you disable Visual Styles from
application properties
```

```
End Sub
```

The function GetColor is defined as follows:

```vb
Private Function GetColor(ByVal score As Integer) As Color
    Select Case score
        Case 0
            Return Color.Red
        Case 1
            Return Color.Red
        Case 2
            Return Color.Purple
        Case 3
            Return Color.LightGreen
        Case 4
            Return Color.MediumSeaGreen
        Case 5
            Return Color.Green
        Case 6
            Return Color.DarkGreen
    End Select
End Function
```

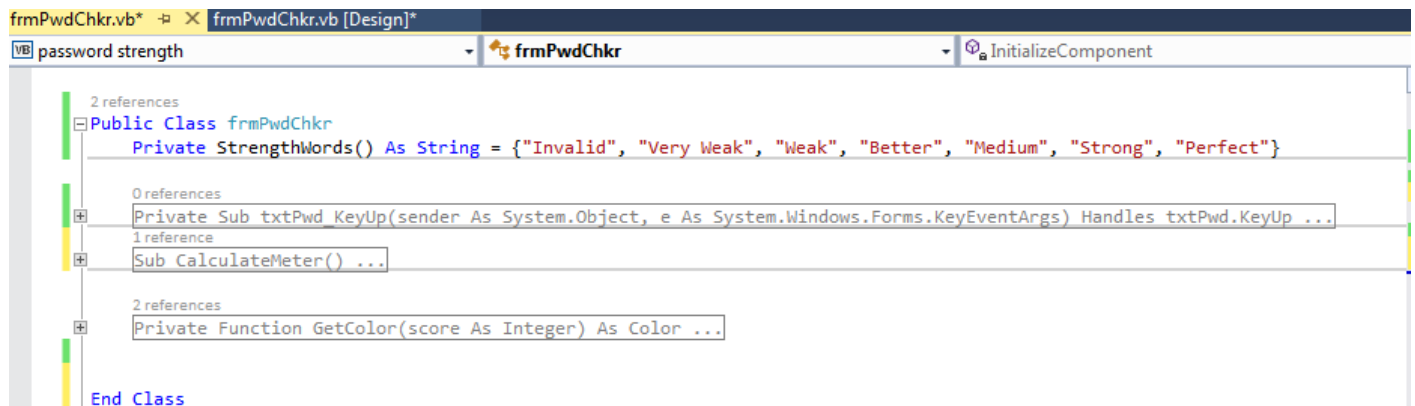## Calling CalculateMeter() when entering a character

To calculate the strength of the password entered in txtPwd textbox, the sub function CalculateMeter() should be called from the KeyUp event sub of txtPwd textbox. To do so, write the following code:

```vb
Private Sub txtPwd_KeyUp(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles txtPwd.KeyUp

    CalculateMeter()
End Sub
```

Final Product:

At the end of this session you should have the following codes:

```vb
frmPwdChkr.vb*    frmPwdChkr.vb [Design]*
password strength                    frmPwdChkr                    InitializeComponent

    2 references
  Public Class frmPwdChkr
      Private StrengthWords() As String = {"Invalid", "Very Weak", "Weak", "Better", "Medium", "Strong", "Perfect"}

      0 references
      Private Sub txtPwd_KeyUp(sender As System.Object, e As System.Windows.Forms.KeyEventArgs) Handles txtPwd.KeyUp ...
      1 reference
      Sub CalculateMeter() ...

      2 references
      Private Function GetColor(score As Integer) As Color ...

  End Class
```

The "txtPwd_KeyUp" sub should look like this:

```vb
Private Sub txtPwd_KeyUp(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles txtPwd.KeyUp
    CalculateMeter()
End Sub
```

The "CalculateMeter" sub should look like this:

```vb
Sub CalculateMeter()
    Dim score As Integer

    Dim password As String = txtPwd.Text
    If (password.Length > 6) Then score += 1 'Length more than 6
    If System.Text.RegularExpressions.Regex.IsMatch(password, "[a-z]") And System.Text.RegularExpressions.Regex.IsMatch(password, "[A-Z]") Then
        score += 1 'upper and lower case
    End If
    If System.Text.RegularExpressions.Regex.IsMatch(password, "\d+") Then
        score += 1 'number
    End If

    If System.Text.RegularExpressions.Regex.IsMatch(password, "[,!,@,#,$,%,^,&,*,?,_,~,-,/,"",]") Then
        score += 1 'special character
    End If

    If (password.Length >= 10) Then score += 1 'length more than 9
    If (password.Length > 15) Then score += 1 'length more than 15

    pbStrength.Value = score / 6 * 100 'finding percentage to increase
    lblProgessbar.Width = 50 * score 'label width is not auto so seeting it to show color amount
    lblProgessbar.Text = StrengthWords(score) 'Getting strength word from string array declarred above
    lblProgessbar.TextAlign = ContentAlignment.MiddleCenter 'alignning to center can be done one time in design
    lblProgessbar.BackColor = GetColor(score) 'Getting color and setting
    pbStrength.ForeColor = GetColor(score) 'does not work unless you disable Visual Styles from application properties

End Sub
```

The "GetColor" function should look like this:

```vb
    Private Function GetColor(ByVal score As Integer) As Color
        Select Case score
            Case 0
                Return Color.Red
            Case 1
                Return Color.Red
            Case 2
                Return Color.Purple
            Case 3
                Return Color.LightGreen
            Case 4
                Return Color.MediumSeaGreen
            Case 5
                Return Color.Green
            Case 6
                Return Color.DarkGreen

        End Select
    End Function


End Class
```