

Sending Verification Code to Email

These guidelines show how implement a service to send verification code user email account during registration process in your application.

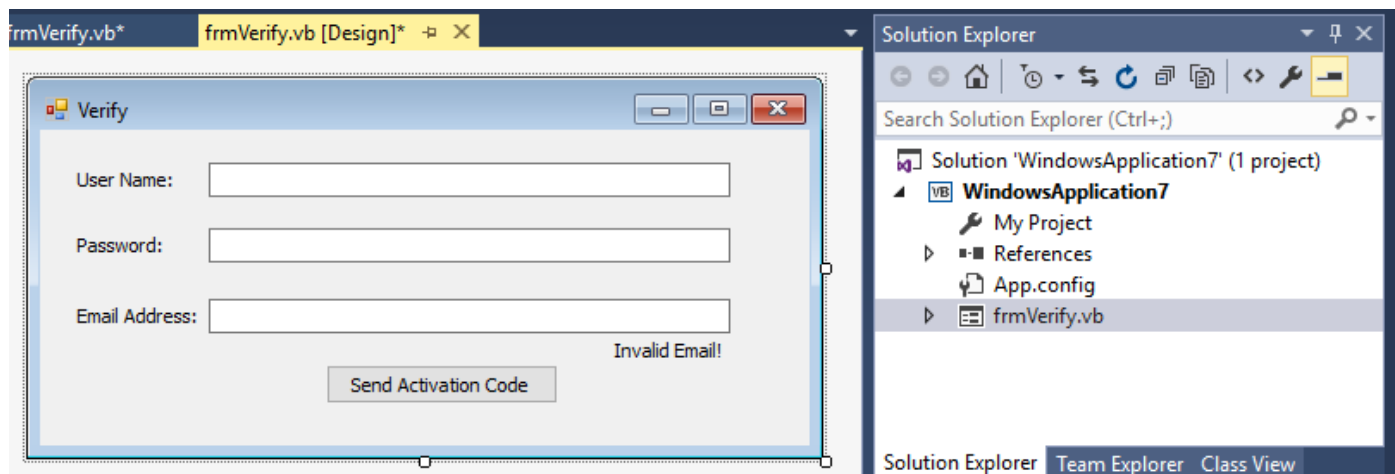
Time needed to accomplish this task: 30 - 45 minutes

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 “frmVerify”, and change its caption/text to “Verify”
- In “frmVerify” form, create 3 textboxes, 4 labels, and 1 button and change their properties as follows:

Old Name	New Name	Caption/Text	Other Properties
TextBox1	txtUserName		
TextBox2	txtPassword		PasswordChar: *
TextBox3	txtEmail		
Label1	lblUserName	User Name:	
Label2	lblPassword	Password:	
Label3	lblEmail	Email Address:	
Label4	lblVerified	Invalid Email!	
Button1	btnSendAC	Send Activation Code	

- 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 import some important references which will be used to develop this application. Write the following lines at the beginning of your code:

```
Imports System.Text.RegularExpressions
Imports System.Net.Mail
```

- If these references are not added to your .NET environment, you may go to Project>Add References.. and add above references (System.Text.RegularExpressions and System.Net.Mail) from there.

Check for valid/invalid Email address format:

Before sending verification code to the email address given by the user, it is important to validate and check whether the email given is in a valid email address or not. To determine the validity of given email address, we will create a function (EmailCheck) to do this task. This function will look for patterns associated with valid email addresses. If these patterns are found, the given email address is considered valid. Otherwise, the email address will be considered as invalid. The (*EmailCheck*) function is given below:

```
Function EmailCheck(ByVal emailAddress As String) As Boolean

    Dim pattern As String = "^([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. |(( [a-zA-Z0-9\-\ ]+\. )+))([a-zA-Z]{2,4}|[0-9]{1,3})(\ )?)$"

    Dim emailAddressMatch As Match = Regex.Match(emailAddress, pattern)

    If emailAddressMatch.Success Then
        Return True
    Else
        Return False
    End If

End Function
```

This (*EmailCheck*) function will be called whenever there is a change in the email address textbox (*txtEmail*). To do so, write the following code in the *txtEmail_TextChanged* event sub.

```
If EmailCheck(MyEmail) = True Then
    lblVerified.Text = "Valid Email!"
    btnSendAC.Enabled = True
Else
    lblVerified.Text = "Invalid Email!"
    btnSendAC.Enabled = False
End If
```

Send Email:

This application then needs to send a verification code to the email address provided by the user. To do so, write the following code as the “btnSendAC_Click” sub:

```
Dim activationCode As String = Guid.NewGuid().ToString()

Dim body As String = "Hello " + txtUsername.Text.Trim() + ","
body += vbCrLf + vbCrLf + "Please use following code to activate your account."
body += vbCrLf + "Activation Code: " + activationCode
body += vbCrLf + vbCrLf + "Thanks"

Try
    Dim Sntp_Server As New SntpClient
    Dim e_mail As New MailMessage()
    Sntp_Server.UseDefaultCredentials = False
    Sntp_Server.Credentials = New Net.NetworkCredential("SENDERemail@gmail.com",
"SENDERpassword")
    Sntp_Server.Port = 587
    Sntp_Server.EnableSsl = True
    Sntp_Server.Host = "smtp.gmail.com"

    e_mail = New MailMessage()
    e_mail.From = New MailAddress("SENDERemail@gmail.com")
    e_mail.To.Add(txtEmail.Text)
    e_mail.Subject = "SecureChat Activation Code"
    e_mail.IsBodyHtml = False
    e_mail.Body = body
    Sntp_Server.Send(e_mail)
    MsgBox("Mail Sent")

Catch error_t As Exception
    MsgBox(error_t.ToString)

End Try
```

As you may have noticed, we are going to use Gmail servers and email services from Google to produce this exercise. Remember to change the “SENDERemail@gmail.com” to your valid gmail address, and to put your gmail password instead of “SENDERpassword” in the above code.

Try running the program and see what happens.

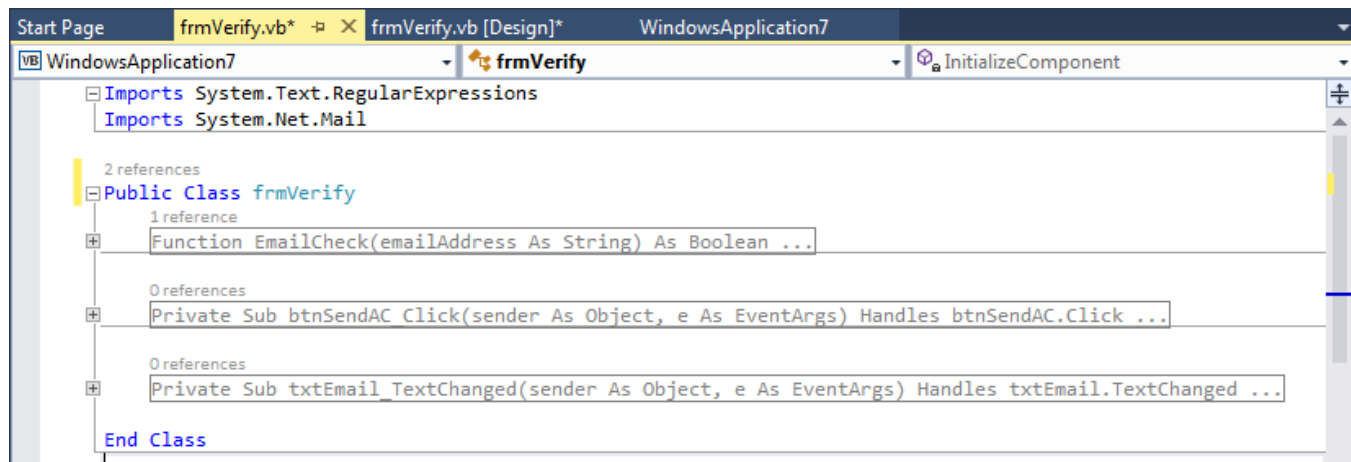
YES, you may receive an error message saying that the program couldn’t access your gmail account to send this email. This is because we are using less secure sign-in technology to access your gmail account. Therefore, you need to activate this service from your Google account. To do so, go to the following link: <https://www.google.com/settings/security/lesssecureapps> and turn on Access for less secure apps option.

Attention:

Activating Access for less secure apps option may not be a good idea for the long run. Keeping these settings off helps in safeguarding your account even more

Final Product:

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



The “btnSendAC_Click” sub should look like this:

```

Private Sub btnSendAC_Click(sender As Object, e As EventArgs) Handles btnSendAC.Click

    Dim activationCode As String = Guid.NewGuid().ToString()

    Dim body As String = "Hello " + txtUsername.Text.Trim() + ", "
    body += vbCrLf + vbCrLf + "Please use following code to activate your account."
    body += vbCrLf + "Activation Code: " + activationCode
    body += vbCrLf + vbCrLf + "Thanks"

    Try
        Dim Smtplib_Server As New SmtplibClient
        Dim e_mail As New MailMessage()
        Smtplib_Server.UseDefaultCredentials = False
        Smtplib_Server.Credentials = New Net.NetworkCredential("SENDERemail@gmail.com", "SENDERpassword")
        Smtplib_Server.Port = 587
        Smtplib_Server.EnableSsl = True
        Smtplib_Server.Host = "smtp.gmail.com"

        e_mail = New MailMessage()
        e_mail.From = New MailAddress("SENDERemail@gmail.com")
        e_mail.To.Add(txtEmail.Text)
        e_mail.Subject = "SecureChat Activation Code"
        e_mail.IsBodyHtml = False
        e_mail.Body = body
        Smtplib_Server.Send(e_mail)
        MsgBox("Mail Sent")

    Catch error_t As Exception
        MsgBox(error_t.ToString)
    End Try

End Sub
  
```

The “EmailCheck” function should look like this:

```
Function EmailCheck(ByVal emailAddress As String) As Boolean

    Dim pattern As String = "^[a-zA-Z0-9_\-\.]+\@(\[[0-9]{1,3}\.?[0-9]{1,3}\.?[0-9]{1,3}\.?\)|\([a-zA-Z0-9\-\]+\.)\)([a-zA-Z]{2,4}|[0-9]{1,3})(\)?)$"
    Dim emailAddressMatch As Match = Regex.Match(emailAddress, pattern)

    If emailAddressMatch.Success Then
        Return True
    Else
        Return False
    End If

End Function
```

The “txtEmail_TextChanged” sub should look like this:

```
Private Sub txtEmail_TextChanged(sender As Object, e As EventArgs) Handles txtEmail.TextChanged
    Dim MyEmail As String = txtEmail.Text
    If EmailCheck(MyEmail) = True Then
        lblVerified.Text = "Valid Email!"
        btnSendAC.Enabled = True
    Else
        lblVerified.Text = "Invalid Email!"
        btnSendAC.Enabled = False
    End If
End Sub
```