

Source: C# Corner (www.c-sharpcorner.com)

PRINT

Article



Voice Login System Using Microsoft Visual Studio

By **Kishore Chowdary** on Jan 07 2017

Introduction

This article will help you to create a login system, which works with voice commands. This will work with the help of speech synthesizer class. Let us assume that you are using a automated system, which works with the help of voice commands. Subsequently, you need some sort of authentication process in it. Here, you can make use of this voice login system.

Step 1

Creating a new Windows form and adding reference file

- Open Visual Studio and create new Windows Form Application with any name you want.
- Once you have created a new form Application, open the Designer Window of Form1.
- Simply add some label to make a small impression that it is Form1.
- Now, to make your PC recognize your voice, you must add a reference file named "system.speech" from the reference files.
- This will now help to convert the voice input to text for the authentication process.

Step 2

Writing code for speech recognition and creating login block

- Now, you have to write a piece of code for speech recognition.
- Let us have the user name as "KISHORE" and password as "CHOWDARY".
- In the code which you are going to write, you can replace this user name and password with your own choice.
- Have a look over the code given below to get an idea about it.
- In the code, you can find something called KISHORE. Just replace it with your own choice of the user name.
- The code will be, as shown below. Paste the code given below by double clicking the Form1.

```
1. private void Form1_Load(object sender, EventArgs e)
2.     {
3.         speechSynthesizerObj = new SpeechSynthesizer();
4.         speechSynthesizerObj.SpeakAsync("Please say the user name.");
5.         SpeechRecognitionEngine sRecognize = new SpeechRecognitionEngine();
6.         Choices sList = new Choices();
7.         sList.Add(new string[] {
8.             "kishore"
9.         });
10.        Grammar gr = new Grammar(new GrammarBuilder(sList));
11.        try
12.        {
```

```

13.         sRecognize.RequestRecognizerUpdate();
14.         sRecognize.LoadGrammar(gr);
15.         sRecognize.SpeechRecognized += sRecognize_SpeechRecognized;
16.         sRecognize.SetInputToDefaultAudioDevice();
17.         sRecognize.RecognizeAsync(RecognizeMode.Multiple);
18.         sRecognize.Recognize();
19.     }
20.     catch
21.     {
22.         return;
23.     }
24. }

```

- There is some other piece of code, which follows the code given above. This is the code, which checks whether the given user name is correct or wrong.
- The code given below follows the code given above to check the correct user name.
- For checking purposes, we can make use of IF condition.
- Once the correct user name is given, now we have to again make use of the speech synthesizer class for the password checking.
- Hence for this purpose, we will again write the speech code and here you can replace the word "CHOWDARY" with your own password.
- Again an if condition is used to check the password and if it is verified and you can login.
- Before it, add a new form, so that it can act as the home page, which we can access after logging in successfully.
- Now, we have to make use of the navigation command for going to the next page.
- The code will be, as shown below.

```

1. if (e.Result.Text == "kishore")
2. {
3.     speechSynthesizerObj = new SpeechSynthesizer();
4.     speechSynthesizerObj.SpeakAsync("Please say the password.");
5.     SpeechRecognitionEngine sRecognize = new SpeechRecognitionEngine();
6.     Choices sList = new Choices();
7.     sList.Add(new string[] { "chowdary" });
8.     Grammar gr = new Grammar(new GrammarBuilder(sList));
9.     Thread.Sleep(500);
10.    try
11.    {
12.        sRecognize.RequestRecognizerUpdate();
13.        sRecognize.LoadGrammar(gr);
14.        sRecognize.SpeechRecognized += sRecognize_SpeechRecognized1;
15.        sRecognize.SetInputToDefaultAudioDevice();
16.        sRecognize.RecognizeAsync(RecognizeMode.Multiple);
17.        sRecognize.Recognize();
18.    }
19.    catch
20.    {
21.        return;
22.    }
23. }
24.
25. private void sRecognize_SpeechRecognized1(object sender, SpeechRecognizedEventArgs e)
26. {
27.     //Thread.Sleep(2500);
28.     if (e.Result.Text == "chowdary")
29.     {
30.         speechSynthesizerObj.SpeakAsync("Login successfull. Welcome");

```

```

31.         Form Form2 = new Form();
32.         Form2.Show();
33.         Hide();
34.     }

```

- The complete code for the whole program will look like, as shown below.

```

1. using System;
2. using System.Speech.Recognition;
3. using System.Speech.Synthesis;
4. using System.Threading;
5. using System.Windows.Forms;
6.
7. namespace LoginSystemUsingVoiceControls
8. {
9.     public partial class Form1 : Form
10.    {
11.        public Form1()
12.        {
13.            InitializeComponent();
14.        }
15.
16.        private void label1_Click(object sender, EventArgs e)
17.        {
18.
19.        }
20.        SpeechSynthesizer speechSynthesizerObj;
21.        private void Form1_Load(object sender, EventArgs e)
22.        {
23.            speechSynthesizerObj = new SpeechSynthesizer();
24.            speechSynthesizerObj.SpeakAsync("Please say the user name.");
25.            SpeechRecognitionEngine sRecognize = new SpeechRecognitionEngine();
26.            Choices sList = new Choices();
27.            sList.Add(new string[] {
28.                "kishore"
29.            });
29.            Grammar gr = new Grammar(new GrammarBuilder(sList));
30.            try
31.            {
32.                sRecognize.RequestRecognizerUpdate();
33.                sRecognize.LoadGrammar(gr);
34.                sRecognize.SpeechRecognized += sRecognize_SpeechRecognized;
35.                sRecognize.SetInputToDefaultAudioDevice();
36.                sRecognize.RecognizeAsync(RecognizeMode.Multiple);
37.                sRecognize.Recognize();
38.            }
39.            catch
40.            {
41.                return;
42.            }
43.        }
44.
45.        private void sRecognize_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
46.        {
47.            if (e.Result.Text == "kishore")
48.            {
49.                speechSynthesizerObj = new SpeechSynthesizer();
50.                speechSynthesizerObj.SpeakAsync("Please say the password.");

```

```
51.     SpeechRecognitionEngine sRecognize = new SpeechRecognitionEngine();
52.     Choices sList = new Choices();
53.     sList.Add(new string[] { "chowdary" });
54.     Grammar gr = new Grammar(new GrammarBuilder(sList));
55.     Thread.Sleep(500);
56.     try
57.     {
58.         sRecognize.RequestRecognizerUpdate();
59.         sRecognize.LoadGrammar(gr);
60.         sRecognize.SpeechRecognized += sRecognize_SpeechRecognized1;
61.         sRecognize.SetInputToDefaultAudioDevice();
62.         sRecognize.RecognizeAsync(RecognizeMode.Multiple);
63.         sRecognize.Recognize();
64.     }
65.     catch
66.     {
67.         return;
68.     }
69. }
70. else
71. {
72.     speechSynthesizerObj.SpeakAsync("Please say the valid user name.");
73.     Thread.Sleep(4000);
74. }
75. }
76. private void sRecognize_SpeechRecognized1(object sender, SpeechRecognizedEventArgs e)
77. {
78.     //Thread.Sleep(2500);
79.     if (e.Result.Text == "chowdary")
80.     {
81.         speechSynthesizerObj.SpeakAsync("Login successfull. Welcome");
82.         Form Form2 = new Form();
83.         Form2.Show();
84.         Hide();
85.     }
86. }
87. }
88. }
89.
90.
91.
92.
```

Final Step

- Now, save the code and debug the the project.
- You have successfully created a login system, which will work with the help of the voice commands.
- If you have any queries, feel free to contact me. Thank you.

Thank you for using C# Corner