DIDforsale Simple IVR system.

Creating an IVR for the industry where companies can manage the new customer complaints/queries as well as handle the queries of non-costumers:

IVR FLOW:

Step: 1

To speak with a Sales representative, press 1. To reach a Customer Support agent, press 2. To reach our Billing department, press 3. If you would like to speak with an Operator, press 0.

Any above input here after pressing 1 for the first time will try to connect call with concerned dept. (Numbers for the dept. are preset in configuration)

Step 2:

If unable to connect it will play a prerecorded message (voice/text to speech) that will ask user to press key (9) to record and leave a voicemail for the concerned dept.

Details for recording voicemail are given in the code on keypress (e.g. press * to finish your recording).

Follow this: Set the webhook URL for DID phone number after logging in to user portal of DIDforSale.

Login> Go to DID configs under Products > Select DID number> Choose action manage routing and apply> SELECT YOUR CONFIGURATION AND CHOOSE FORWARD TO WEB URL

Web.Config Settings

```
<add key="WebsitePath" value="http://domainname.com" />
<add key="SupportEmail" value="user@domainname.com" />
```

You must set website path and email for voicemail forwarding to email address after it’s recorded.

IVRController.cs

```
using Did.didML;
using DidIVR.Services;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Configuration;
using System.Web.Mvc;

namespace DidIVR.Controllers
{
    public class IVRController : Controller
    {
        // GET: IVR

        IVRService _ivrservice = new IVRService();
        public ActionResult Start()
        {
            string response = "<Gather timeout=\"10\" action=\"" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/GetDigits\"
numDigits=\"1\"><Say>To speak with a Sales representative, press 1. To reach a Customer
Support agent, press 2. To reach our Billing department, press 3. If you would like to
speak with an Operator, press 0.</Say></Gather>";
            return this.Content("<Response>" + response + "</Response>", "text/xml");
        }
        public ActionResult GetDigits(string digits)
        {
            string response = string.Empty;
            switch (digits)
            {
                case "0":
                    response = "<Say>All Operators are busy on another call. Please hold
the line and stay with us.</Say><Dial action=\"" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/CallEvents\"
><Number>1XXXXXXXXXX</Number></Dial>";
                    break;
                case "1":
                    response = "<Say>Welcome you in sale department. All sale persons are
busy on another call. Please hold the line and stay with us</Say><Dial action=\"" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/CallEvents\"
><Number>1XXXXXXXXXX</Number></Dial>";
                    break;
                case "2":
                    response = "<Say>All Customer Support agent are busy on another call
please hold the line and stay with us.</Say><Dial action=\"" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/CallEvents\"
><Number>1XXXXXXXXXX</Number></Dial>";
                    break;
                case "3":
                    response = "<Say>Welcome you in billing department.All persons are
busy on another call. Please hold the line and stay with us</Say><Dial action=\"" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/CallEvents\"
><Number>1XXXXXXXXXX</Number></Dial>";
                    break;
                case "9":
                    response = "<Say>Please Record your message after beep, to finish
reording press *.</Say><Record maxLength=\"30\" finishOnKey=\"*\"  action=\"" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/RecordingUrl\" />
";
                    break;
                default:
                    response = "<Say>You have entered wrong input. Thanks to call us.
Good bye</Say><Hangup />";
                    //response += "<Redirect>" +
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/Start</Redirect>";
                    break;
            }
            return this.Content("<Response>" + response + "</Response>", "text/xml");
        }
        [HttpPost]
        public ActionResult CallEvents(string dialCallStatus)
        {
            string response = string.Empty;
            if (dialCallStatus == "completed")
            {
                return this.Content("<Root></Root>", "text/xml");
            }
            response = "<Gather timeout=\"10\" action=\"" + 
WebConfigurationManager.AppSettings["WebsitePath"].ToString() + "/ivr/GetDigits\" 
numDigits=\"1\"><Say>Sorry our representative is busy on another call, Press 9 to start 
reording.</Say></Gather>";
            return this.Content("<Response>" + response + "</Response>", "text/xml");
        }

        [HttpPost]
        public ActionResult RecordingUrl(string recordingUrl, string callerid)
        {
            string response = string.Empty;
            if (string.IsNullOrEmpty(recordingUrl))
            {
                response = "<Say>No message recorded, Good bye.</Say><Hangup />";
            }
            else {
                response = "<Say>Thanks, we contact you shortly. Good bye.</Say><Hangup 
/>";
                //To forward recorded voicemail to email
                _ivrservice.SendEmail("New Voicemail Message from " + callerid, "You have 
a new voicemail from " + callerid + ".<br/><br/><br/> Click this link to listen to the 
message:<br/>" + recordingUrl);

            }
            return this.Content("<Response>" + response + "</Response>", "text/xml");
        }

    }
}
```

In the above code we have used gather verb to get the keypress input. On the keypress it will redirect to the action with key. We will get the key input using `GetDigits action`**. Then the given logic above will be followed.**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Mail;
using System.Web;
using System.Web.Configuration;

namespace DidIVR.Services
{

    interface IIVRService
    {
        void SendEmail(string subject, string body);
    }
    public class IVRService : IIVRService
    {
        public void SendEmail(string subject, string body)
        {
            SmtpClient client = new SmtpClient();
            client.Port = 587;
            client.Host = "smtp.gmail.com";
            client.EnableSsl = true;
            client.Timeout = 10000;
            client.DeliveryMethod = SmtpDeliveryMethod.Network;
            client.UseDefaultCredentials = false;

            client.Credentials = new System.Net.NetworkCredential("fromemail",
"password");
            MailMessage mm = new MailMessage("fromemail",
WebConfigurationManager.AppSettings["SupportEmail"].ToString(), subject, body);
            mm.IsBodyHtml = true;
            mm.DeliveryNotificationOptions = DeliveryNotificationOptions.OnFailure;
            client.Send(mm);
        }
    }
}
```

This code show the email function used for sending recorded voicemail to a particular email address.