

به نام خدا

موضوع: مستند سرویس تحت ویندوز PCPOS

مورد کاربرد: از این سرویس برای تبادل اطلاعات بین دستگاه کارت خوان و کامپیوتر استفاده می شود.

بانک های تحت پشتیبانی: در حال حاضر دو بانک شهر و پارسیان پشتیبانی می شوند.

پیاده سازی:

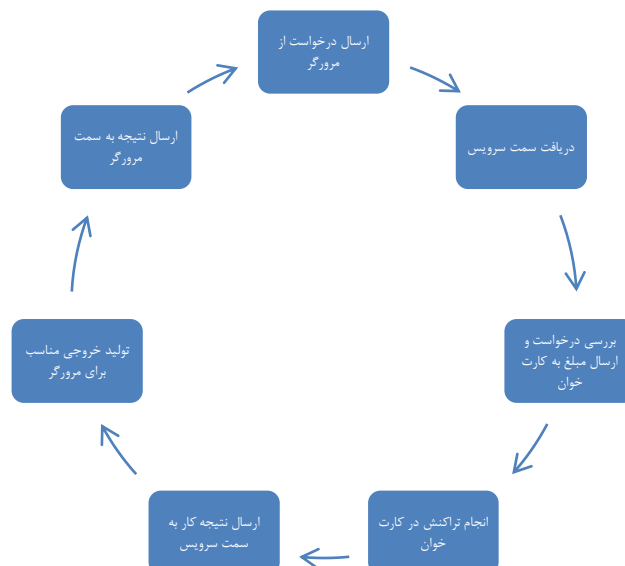
با توجه به مستندی که هر بانک از دستگاه کارت خوان در اختیار گذاشته است این سرویس تولید شده است. برای تبادل اطلاعات از سمت وب و مرورگر با دستگاه کارت خوان، این سرویس پورتی از سیستم را شنود می کند.

ماهیت این سرویس به این صورت است که بر روی پورتی از سیستم نصب شده منتظر درخواست هایی از نوع http می ماند. هر درخواستی شامل اطلاعاتی است که نوع بانک و روش اتصال به کارت خوان را مشخص می کند.

اگر دستگاه کارت خوان از طریق پورت به سیستم متصل بود درخواست به آن پورت ارسال می شود و در صورتی که کارت خوان تحت شبکه و دارای ای پی مجزا بود درخواست دریافت شده به آن آی پی ارسال می شود.

الگوریتم سرویس PCPOS:

۱. درخواست پرداخت از سمت مرورگر به آی پی و پورت سیستم ارسال می شود.
۲. سرویس مورد نظر درخواست را دریافت می کند.
۳. درخواست بررسی می شود که به چه ای پی و یا پورتی براساس نوع بانک ارسال شود.
۴. بعد از فراهم ساختن داده های مورد نیاز، درخواست از سیستم به کارت خوان ارسال می شود.
۵. بعد از انجام تراکنش در دستگاه کارت خوان اطلاعات خروجی به سمت سرویس ارسال می شود.
۶. سرویس، داده های کارت خوان را گرفته و بعد از تجمیع به سمت مرورگر ارسال می کند.



از آنجایی که ماهیت برنامه به صورت سرویس ویندوزی است؛ کدهای زیر که بیان تعریفی از یک سرویس در زبان برنامه نویسی سی شارپ است؛ ارایه می شود.

در این قطعه کد ابتدا درخواست از مرورگر دریافت می شود و با توجه به اطلاعات دریافت شده یکی از کلاس های موجود در فایل Parsian.cs و یا Shahr.cs صدا زده می شود.

برای اینکه برنامه همیشه در حال اجرا باشد؛ به صورت سرویس تحت ویندوز طراحی شده است. در این سرویس یک نخ و یک شنودگر روی دادهای سیستم، تعریف شده است. شنودگر درخواست هایی که به پورت http ارسال می شود را دریافت کرده و به نخ تعریف شده می دهد.

متدهای موجود در کدهای سرویس:

(۱) Service1 :

فراخوانی کامپوننت های سیستم برای اجرای سرویس.

(۲) OnStart :

کدهایی که در شروع کار سرویس باید اجرا شود در این قسمت قرار می گیرد. در اینجا کدهای شنودگر و نخ قرار می گیرد.

(۳) WorkerThreadFunc :

تابعی که برای اجرا به نخ پاس داده می شود. تا زمانی که درخواستی ارسال نشده نخ متوقف است و زمانی که درخواستی از سمت مرورگر به پورت مربوطه ارسال شد فعال شده و درخواست را مدیریت می کند.

(۴) ProcessRequest :

درخواست ارسال شده در این تابع بررسی می شود و مشخص می شود که به چه پورت و یا آی پی خاصی فرستاده شود و این که بانک مربوطه را مشخص می کند. اگر درخواست مربوط به بانک پارسیان بود کلاس Parsian.cs صدا زده می شود و در صورتی که بانک شهر بود کلاس Shahr.cs صدا زده می شود و اطلاعات درخواست به آن کلاس ارسال می شود.

(۵) PrintToOutPut :

بعد از پایان کار و ارسال و دریافت اطلاعات به دستگاه کارت خوان نتیجه باید به سمت مرورگر ارسال شود. این تابع وظیفه دارد اطلاعات را به صورت جی سون به مرورگر ارسال کند.

```
namespace PAX.PCPOS.WinSample
{
    public partial class Service1 : ServiceBase
    {
        public Service1()
        {
            // ویندوزی سرویس هر نیاز پیش تابع
            // ویندوز در اجرا برای نیاز مورد های کامپوننت فراخوانی
            InitializeComponent();
        }
    }
}
```

```

// دهد می پاس نخ به را سیستم های داد، رخ ابجکت این
private ManualResetEvent _shutdownEvent = new ManualResetEvent(false);
// فرایند اجرای برای نخ تعریف
private Thread _thread;
// برنامه پابی خطا برای لاگر تعریف
LogWriter logger = new LogWriter("");

سرویس های فعالیت شروع برای لازم کدهای تابع این در سرویس استارت تابعه
شود می وارد
protected override void OnStart(string[] args)
{
    // برنامه در شده تعریف متغیرهای از تنظیمات مشخصات گرفتن
    string Name = ConfigurationSettings.AppSettings.Get("MySetting");

    // حافظه در گیری فضا برای نخ، تعریف باز
    _thread = new Thread(WorkerThreadFunc);
    _thread.Name = "My Worker Thread";
    // شود اجرا سیستم وظایف زمینه پس در باید نخ این
    _thread.IsBackground = true;
    // فعالیت شروع برای نخ کردن استارت
    _thread.Start();

}

مورد های فعالیت تا شده داده بالا در شده تعریف نخ به که است تابعی این
دهد انجام را نیاز
private void WorkerThreadFunc()
{
    // شود دریافت سیگنالی که زمانی تا کن بلاک را کنونی نخ
    while (!_shutdownEvent.WaitOne(0))
    {
        // Replace the Sleep() call with the work you need to do
        // شنود برای کنونی پورت و ادرس
        var prefix = "http://localhost:8080/";
        // دریافت برای پی تی تی ای پروتکل تحت شنودگر از ای ساده تعریف
        و دریافت برای پی تی تی ای پروتکل تحت شنودگر از ای ساده تعریف
        سون جی صورت به ها داده ارسال
        HttpListener listener = new HttpListener();
        // شنودگر به ادرس الحاق
        listener.Prefixes.Add(prefix);
        try
        {
            // کن فعال را پی تی تی ای پروتکل تحت شنودگر
            listener.Start();
        }
        catch (HttpListenerException hlex)
        {
            // کن چاپ پیامی خطا وجود صورت در
            Console.WriteLine(hlex.Message);
            return;
        }
        try
        {
            // شود می داده پاس وب سمت از ای داده که زمانی تا
            while (listener.IsListening)
            {
                // بگیر را شده ارسال درخواست محتوای
                var context = listener.GetContext();

                // کن ارسال زیر تابع به را درخواست
                ProcessRequest(context);
            }
        }
    }
}

```

```

    }
    catch (HttpListenerException hlex)
    {
        // ببند را شنودگر داشت وجود خطایی اگر
        listener.Close();
        // کن لاگ خطا نوع
        logger.LogWrite(hlex.Message);
    }
}

// دهد می پاسخ را سیستم پورت به شده ارسال درخواست که اصلی تابع
private static void ProcessRequest(HttpListenerContext context)
{
    //Console.WriteLine(res2.Success.ToString());
    // Get the data from the HTTP stream
    // شده ارسال درخواست از ها داده جداسازی
    var body = new
StreamReader(context.Request.InputStream).ReadToEnd();
    // واردکن کالکشنی در را شده ارسال های داده
    NameValueCollection query = HttpUtility.ParseQueryString(body);
    //string Data;
    // تابع این خروجی ارسال برای جدید کالکشن تعریف
    Dictionary<string, string> httpResponse = new Dictionary<string,
string>();
    try
    {
        // بود پی ای شامل شده ارسال درخواست اگر
        if (query.Get("IP") != "")
        {

            try
            {
                // شود می بررسی شده ارسال درخواست نوع
                switch (query.Get("Type"))
                {
                    // بود شهر بانک برای و پی ای نوع از درخواست اگر
                    case "shahr":
                        #region PC POS Shahr bank
                        // شهر بانک کلاس به درخواست دریافت و ارسال تابع
                        شود می زده صدا

                        Shahr shahrPos = new Shahr(httpResponse);
                        httpResponse = shahrPos.pay(query);
                        #endregion
                        break;

                    default:
                        // نیست مشخص بانک نوع خروجی در خطا کردن مشخص
                        httpResponse.Add("ErrorCode", "-8000");
                        httpResponse.Add("ErrorMessage", "بانک نوع لطفا");
                        "کنید مشخص را";

                        PrintToOutPut(httpResponse, context);
                        return;
                }
            }
        }
    }
    catch (Exception e)
    {

```

```

        // اجرا زمان خطای کردن مشخص
        httpResponse.Add("ErrorCode", "-9000");
        httpResponse.Add("ErrorMessage", e.Message);
    }

}

// بود پورت فیلد شامل شده ارسال درخواست اگر
else if (query.Get("Port") != "")
{

    // شده ارسال درخواست نوع بررسی
    switch (query.Get("Type"))
    {
        // بود پارسیان بانک نوع اگر
        case "parsian":
            #region PC POS Parsian bank
            // صدا پارسیان بانک به درخواست دریافت و ارسال تابع

            Parsian parsianPos = new Parsian(httpResponse);
            httpResponse = parsianPos.pay(query);
            parsianPos = null;

            #endregion
            break;

        default:
            // باشد نمی مشخص بانک نوع - خروجی خطای تولید
            httpResponse.Add("ErrorCode", "-8000");
            httpResponse.Add("ErrorMessage", "را بانک نوع لطفا");

            PrintToOutPut(httpResponse, context);
            return;

    }

}

PrintToOutPut(httpResponse, context);
}

catch (Exception e)
{
    // می تولید مناسب پیام داد رخ خطایی دریافت و ارسال طول در اگر
    // شود می ارسال کاربر سمت به و
    httpResponse["ErrorCode"] = "200";
    httpResponse["ErrorMessage"] = e.Message;
    httpResponse["MerchantId"] = "";
    httpResponse["TerminalId"] = "";
    httpResponse["SerialNumber"] = "";
    httpResponse["MerchantName"] = "";

    httpResponse["Success"] = "False";
    httpResponse["WarningCode"] = "0";

    // بود داده رخ خطایی که صورتی در کردن لاگ
    LogWriter logger = new LogWriter("Start log in Main Thread
ProcessRequest");
    logger.LogWrite(e.Message);

    // کند می ارسال خروجی به را کار نتیجه که تابعی
    PrintToOutPut(httpResponse, context);

}

```

```

    }

    // خروجی در شده تولید های داده نوشتن برای نیاز مورد تابع
    // شود می ارسال مرورگر به و بوده سون جی صورت به خروجی
    public static void PrintToOutPut(Dictionary<string, string> dictionary,
HttpListenerContext context)
    {
        var json = new JavaScriptSerializer().Serialize(dictionary);

        // مرورگر سمت ارسال برای نیاز مورد هدرهای و داده تولید
        byte[] b = Encoding.UTF8.GetBytes(json);
        context.Response.StatusCode = 200;
        context.Response.KeepAlive = false;
        context.Response.ContentLength64 = b.Length;
        context.Response.Headers.Add("Access-Control-Allow-Origin: *");
        // System.Threading.Thread.Sleep(10000);
        var output = context.Response.OutputStream;
        output.Write(b, 0, b.Length);

        context.Response.Close();
    }

    // بده انجام را زیر موارد شد متوقف سرویس که زمانی
    protected override void OnStop()
    {
        // کن بازنشانی را کند می کنترل را ارسالی های داد رخ که ابجکتی
        _shutdownEvent.Set();
        // کن متوقف را نظر مورد نخ ثانیه سه گذشت از بعد
        if (!_thread.Join(3000))
        { // give the thread 3 seconds to stop
            _thread.Abort();
        }
    }
}
}
}

```

در ادامه کدهای مربوط به بانک پارسیان که در فایل Parsian.cs نوشته شده است ارائه می شود. توضیحات مربوط به مستندسازی کدهای بانک و نحوه ارسال و دریافت اطلاعات به کارت خوان بانک شهر در فایل "راهنمای پرداخت بانک پارسیان" ذکر شده است.

برای اطلاع از متغیرها و کدهای نوشته شده در تابع بانک به مستند بانک مراجعه کنید.

متدهای موجود در کدهای بانک پارسیان:

1) Parsian:

تعریف ابجکتی برای تولید خروجی به صورت مجموعه ای از مقادیر و ایندکس ها.

2) payMoney:

کدهای مربوط به مستند بانک پارسیان. در این قطعه کد مبلغ نهایی و پورت مربوط به دستگاه کارت خوان که به رایانه متصل است دریافت می شود و سپس با توجه به پیاده سازی که خود بانک مشخص کرده است اطلاعات به سمت کارت خوان ارسال می شود. نتیجه کار در ابجکت خروجی response وارد می شود.

۳ Pay :

این تابع درخواست دریافت شده در سرویس را به تابع **payMoney** ارسال می کند. با توجه به خروجی که در ابجکت **response** ریخته شده مشخص می شود که عملیات سمت دستگاه موفقیت آمیز بوده است و یا اینکه خطایی رخ داده است. خروجی به سمت سرویس بازگردانی می شود.

۴ translateError :

عملیاتی که سمت کارت خوان انجام می شود در صورت مواجه شدن با مشکل کد خطایی تولید می کند. برای ترجمه کد خطا به متن فارسی مناسب بانک پارسیان، این تابع را در اختیار ما قرار داده است.

```
// پارسیان بانک به مربوط کدهای
namespace PAX.PCPOS.WinSample
{
    // پارسیان خوان کارت به اطلاعات دریافت و ارسال
    class Parsian
    {
        // کلاس خروجی برای ابجکتی کردن مشخص
        private Dictionary<string, string> httpResponse;
        // تغییرات و خطاها ثبت برای لاگر تعریف
        LogWriter logger = new LogWriter("");
        string lastError = "";

        public Parsian(Dictionary<string, string> httpResponse)
        {
            // خروجی برای فضا گرفتن
            this.httpResponse = new Dictionary<string, string>();
            this.httpResponse = httpResponse;
        }

        // اطلاعات تبادل و ارتباط برقراری برای کام پورت تعریف
        private Dictionary<int, SerialPort> Coms = new Dictionary<int,
SerialPort>();
        private Dictionary<int, bool> ComStatus = new Dictionary<int, bool>();

        // خروجی گرفتن برای سراسری متغیر تعریف
        // شود می پر بانک تابع در متغیر این
        public Dictionary<string, string> response = new Dictionary<string,
string>();

        public int COM_Index;

        // خوان کارت دستگاه اطلاعات دریافت و ارسال برای بانک تابع
        // مبلغ و سیستم پورت ورودی
        public void payMoney(string AM , string COM)
        {
            BTLV tlv = new BTLV();
            tlv.AddEntry("PR", "00000");
            tlv.AddEntry("AM", AM);
            tlv.AddEntry("CU", "364");
            tlv.AddEntry("R1", "");
            tlv.AddEntry("R2", "");
            tlv.AddEntry("T1", "");
            tlv.AddEntry("T2", "");
            tlv.AddEntry("SV", "");
        }
    }
}
```

```

tlv.AddEntry("SG", "");
tlv.AddEntry("AD", "");
tlv.AddEntry("PD", "1");

String gg = tlv.ToString();
tlv = new BTLV();
tlv.AddEntry("RQ", gg);
String txt_req = tlv.ToString();
byte[] bb =
ASCIIEncoding.GetEncoding(1256).GetBytes(txt_req.Length.ToString().PadLeft(4,
'0') + txt_req);

string[] COMS = new string[] { "COM1", "COM2", "COM3", "COM4",
"COM5", "COM6" };
COM_Index = Array.IndexOf(COMS, COM);

if (!ComStatus.ContainsKey(COM_Index))
{
    ComStatus[COM_Index] = false;
    Coms[COM_Index] = new SerialPort(COMS[COM_Index], 19200,
Parity.None, 8, StopBits.One);
}

if (!ComStatus[COM_Index])
{
    //if (!Coms[COM_Index].IsOpen)
    try {
        Coms[COM_Index].Open();
        logger.LogWrite("Open Port " + COMS[COM_Index]);
    }
    catch(Exception e)
    {
        lastError = "Com Port " + (COM_Index + 1) + " Can not Be
Opened.";
        logger.LogWrite(lastError);
        logger.LogWrite(e.Message);
        return;
    }

    try
    {
        bb =
ASCIIEncoding.GetEncoding(1256).GetBytes(txt_req.Length.ToString().PadLeft(4,
'0') + txt_req);
        Coms[COM_Index].DiscardInBuffer();
        Coms[COM_Index].DiscardOutBuffer();
        Coms[COM_Index].ReadTimeout = 120000;
        Coms[COM_Index].Write(bb, 0, bb.Length);
    }
    catch
    {
        lastError = "Com Port " + (COM_Index + 1) + " Data Can not
Be Sent.";
        logger.LogWrite(lastError);
        return;
    }

    byte[] ll = new byte[4];
    byte[] ss = new byte[500];

```



```
try
{
    Coms[COM_Index].ReceivedBytesThreshold = 4;
    int tt = 0;
    while (true)
    {
        int cc = Coms[COM_Index].Read(11, tt, 4 - tt);
        tt += cc;
        if (tt == 4)
            break;
    }
    int len = int.Parse(ASCIIEncoding.ASCII.GetString(11));
    Coms[COM_Index].ReceivedBytesThreshold = len;
    ss = new byte[len];
    tt = 0;
    while (true)
    {
        int cc = Coms[COM_Index].Read(ss, tt, len - tt);
        tt += cc;
        if (tt == len)
            break;
    }
    string s1 = ASCIIEncoding.GetEncoding(1256).GetString(ss);

//Console.WriteLine("_____")
;
    //Console.WriteLine(new
    BTLV().Open(s1.Substring(5)).GetResponse()["RS"]);

    response = new BTLV().Open(s1.Substring(5)).GetResponse();
    Coms[COM_Index].Close();

}
catch (Exception ex)
{
    lastError = "Com Port " + (COM_Index + 1) + " Data Can not
Be Receive.";
    logger.LogWrite(lastError);
}

Coms[COM_Index].RtsEnable = false;
Coms[COM_Index].DtrEnable = false;
ComStatus[COM_Index] = true;
}

// بانک تابع به ارسال سپس و ها پارامتر کردن فراهم
public Dictionary<string, string> pay(NameValueCollection query)
{
    // نتیجه پرکردن و بانک تابع به ارسال
    payMoney(query.Get("Amount"), query.Get("Port"));

    //dictionary = pos.Receive;
    // از نتیجه گرفته
    Dictionary<string, string> posResponse = response;

    // this code is for error message from open or close port
    // کن اضافه را شده تولید خطای نشد داده برگشت ای نیجه که صورتی در
    if (response.Count == 0)
        httpResponse.Add("ErrorMessage", lastError);
}
```

```

    // مرورگر سمت ارسال برای پارامترها کردن فراهم
    httpResponse.Add("ErrorCode", "0");
    httpResponse.Add("WarningCode", "0");
    httpResponse.Add("MerchantId", "");
    httpResponse.Add("SerialNumber", "");
    httpResponse.Add("MerchantName", "");

    if (posResponse.ContainsKey("RS") && posResponse["RS"] == "00")
        httpResponse.Add("Success", "True");
    else
        httpResponse.Add("Success", "False");

    if (posResponse.ContainsKey("RS")) {
        httpResponse.Add("ResponseCode", posResponse["RS"]);
        httpResponse.Add("ErrorMessage",
translateError(posResponse["RS"]));

        logger.LogWrite(translateError(posResponse["RS"]));
    }

    if (posResponse.ContainsKey("AM"))
        httpResponse.Add("Amount", posResponse["AM"]);
    else
        httpResponse.Add("Amount", "");

    if (posResponse.ContainsKey("TM"))
        httpResponse.Add("TerminalId", posResponse["TM"]);
    else
        httpResponse.Add("TerminalId", "");

    if (posResponse.ContainsKey("TI"))
        httpResponse.Add("DateTime", posResponse["TI"]);
    else
        httpResponse.Add("DateTime", "");

    if (posResponse.ContainsKey("PN"))
        httpResponse.Add("PAN", posResponse["PN"]);
    else
        httpResponse.Add("PAN", "");

    if (posResponse.ContainsKey("RN"))
        httpResponse.Add("RRN", posResponse["RN"]);
    else
        httpResponse.Add("RRN", "");

    if (posResponse.ContainsKey("TR"))
        httpResponse.Add("Stan", posResponse["TR"]);
    else
        httpResponse.Add("Stan", "");

    if (posResponse.ContainsKey("SI"))
        httpResponse.Add("SVC", posResponse["SI"]);
    else
        httpResponse.Add("SVC", "");

    return httpResponse;
}

```

بانک همین مختص و میکند تبدیل آن معال به را شده تولید خطای کد که تابعی

است

```
public string translateError(string errorCode)
```

```

{
    string text = "پرداخت در خطا:";

    int errorValue = int.Parse(errorCode);

    switch (errorValue)
    {
        case 12:
            text += "است نامعتبر تراکنش";
            break;
        case 50:
            text += "مرکز با ارتباط برقراری عدم";
            break;
        case 51:
            text += "باشد نمی کافی موجودی";
            break;
        case 54:
            text += "است شده سپری کارت انقضای تاریخ";
            break;
        case 55:
            text += "است اشتباه کارت رمز";
            break;
        case 56:
            text += "است نامعتبر کارت";
            break;
        case 58:
            text += "است مجاز غیر پایانه .";
            break;
        case 61:
            text += "باشد می مجاز حد از بیش تراکنش مبلغ";
            break;
        case 65:
            text += "است مجاز حد از بیش غلط رمز ورود دفعات تعداد";
            break;
        case 75:
            text += "است اشتباه کارت رمز";
            break;
        case 99:
            text += "درخواست لغو";
            break;

        default:
            text += "..";
            break;
    }

    return text;
}

}
}

```

در ادامه کدهای مربوط به بانک شهر که در فایل Shahr.cs نوشته شده است ارایه می شود. توضیحات مربوط به مستندسازی کدهای بانک و نحوه ارسال و دریافت اطلاعات به کارت خوان بانک شهر در فایل "راهنمای پرداخت بانک شهر" ذکر شده است.

برای اطلاع از متغیرها و کدهای نوشته شده در تابع بانک به مستند بانک مراجعه کنید.

متدهای موجود در کدهای بانک شهر:

۱) Shahr:

تعریف ابجکتی برای تولید خروجی به صورت مجموعه ای از مقادیر و ایندکس ها.

۲) Pay:

این تابع با توجه به اطلاعات دریافتی در سرویس، داده های مناسب را تولید می کند و به سمت آی پی کارت خوان ارسال می کند. ابتدا با استفاده از تابع PingHost بررسی می شود که آی پی وارد شده معتبر باشد و پینگ دهد. سپس تابع BillPayment صدا زده می شود تا کارت خوان درخواست را دریافت کند.

در نهایت خروجی کارت خوان به صورت ابجکتی به سمت سرویس ارسال می شود.

۳) PingHost:

این تابع آی پی ارسالی از مرورگر را بررسی می کند که معتبر باشد و قابلیت ارسال و دریافت اطلاعات را داشته باشد.

۴) translateError:

عملیاتی که سمت کارت خوان انجام می شود در صورت مواجه شدن با مشکل کد خطایی تولید می کند. برای ترجمه کد خطا به متن فارسی مناسب بانک شهر، این تابع را در اختیار ما قرار داده است.

```
// باشد همین باید اسپیس نیم که شود توجه
// است اسپیس نیم این در بانک لایبرری
namespace PAX.PCPOS.WinSample
{
    // شهر بانک خوان کارت به اطلاعات دریافت و ارسال
    class Shahr
    {
        // کلاس خروجی برای ابجکتی کردن مشخص
        private Dictionary<string, string> httpResponse;
        // تغییرات و خطاها ثبت برای لاگر تعریف
        LogWriter logger = new LogWriter("");

        public Shahr(Dictionary<string, string> httpResponse)
        {
            // خروجی برای فضا گرفتن
            this.httpResponse = new Dictionary<string, string>();
            this.httpResponse = httpResponse;
        }

        // بانک تابع به ارسال سپس و ها پارامتر کردن فراهم
        public Dictionary<string, string> pay(NameValueCollection query)
        {
            // تنظیمات از پی ای جداسازی
            string IP = query.Get("IP");
            // شده وارد پی ای اعتبار بررسی
            // نه یا دهد می پینگ شده وارد پی ای این آیا
            if (PingHost(IP) == false)
            {
                // مناسب خطای تولید
                httpResponse.Add("ErrorCode", "200");
                httpResponse.Add("ErrorMessage", "ip نمیده جواب");
                httpResponse.Add("MerchantId", "");
                httpResponse.Add("TerminalId", "");
            }
        }
    }
}
```

```
        httpResponse.Add("SerialNumber", "");
        httpResponse.Add("MerchantName", "");

        httpResponse.Add("Success", "False");
        httpResponse.Add("WarningCode", "0");
        // PrintToOutPut(httpResponse, context);
        return httpResponse;
    }
    // خروجی گرفتن و بانک تابع به تنظیمات ارسال
    PCPOS.PosCommander.Instance.Init(IPAddress.Parse(IP),
    Int32.Parse(query.Get("Timeout")));
    Response res2 =
    PCPOS.PosCommander.Instance.BillPayment(query.Get("AdditionalData"),
    query.Get("BillId"), query.Get("PaymentId"), Boolean.Parse(query.Get("Print")));

    httpResponse.Add("ErrorCode", res2.ErrorCode.ToString());

    // if user cancel the pay operation

    if (res2.ErrorCode == 3)
        httpResponse.Add("ErrorMessage", "درخواست لغو");

    // خروجی برای ها داده کردن فراهم
    httpResponse.Add("MerchantId", res2.PosInformation.MerchantId);
    httpResponse.Add("TerminalId", res2.PosInformation.TerminalId);
    httpResponse.Add("SerialNumber", res2.PosInformation.SerialNumber);
    httpResponse.Add("MerchantName", res2.PosInformation.MerchantName);

    httpResponse.Add("Success", res2.Success.ToString());
    httpResponse.Add("WarningCode", res2.WarningCode.ToString());
    if (res2.TransactionInfo != null)
    {
        httpResponse.Add("Amount", res2.TransactionInfo.Amount);
        httpResponse.Add("DateTime", res2.TransactionInfo.DateTime);
        httpResponse.Add("PAN", res2.TransactionInfo.PAN);
        httpResponse.Add("ResponseCode",
    res2.TransactionInfo.ResponseCode);
        httpResponse.Add("ErrorMessage",
    translateError(res2.TransactionInfo.ResponseCode));
        httpResponse.Add("RRN", res2.TransactionInfo.RRN);
        httpResponse.Add("Stan", res2.TransactionInfo.Stan);
        httpResponse.Add("SVC", res2.TransactionInfo.SVC);

    logger.LogWrite(translateError(res2.TransactionInfo.ResponseCode));
    }
    else
    {
        httpResponse.Add("Amount", "");
        httpResponse.Add("DateTime", "");
        httpResponse.Add("PAN", "");
        httpResponse.Add("ResponseCode", "");
        httpResponse.Add("RRN", "");
        httpResponse.Add("Stan", "");
        httpResponse.Add("SVC", "");
    }

    return httpResponse;
}

// دهد می پینگ پی ای این آیا
public static bool PingHost(string nameOrAddress)
{

```

```

bool pingable = false;
Ping pinger = new Ping();
try
{
    PingReply reply = pinger.Send(nameOrAddress);
    pingable = reply.Status == IPStatus.Success;
}
catch (PingException)
{
    // Discard PingExceptions and return false;
}
return pingable;
}

```

بانک همین مختص و میکند تبدیل آن معال به را شده تولید خطای کد که تابعی

است

```

public string translateError(string errorCode)
{
    string text = "پرداخت در خطا:";

    int errorValue = int.Parse(errorCode);

    switch (errorValue)
    {
        case 0:
            text += "انجام موفق تراکنش";
            break;
        case 1:
            text += "شد نظر صرف تراکنش انجام از";
            break;
        case 2:
            text += "قبلا reverse است شده";
            break;
        case 3:
            text += "است نامعتبر کارت پذیرنده";
            break;
        case 5:
            text += "شد صرفنظر تراکنش انجام از";
            break;
        case 6:
            text += "داخلی خطای";
            break;
        case 9:
            text += "نمائید ارسال تراکنش بعدا لطفا - است مشغول سیستم";
            break;
        case 10:
            text += "Partial Dispense.";
            break;
        case 12:
            text += "نامعتبر تراکنش";
            break;
        case 13:
            text += "نادرست مبلغ";
            break;
        case 14:
            text += "نیست شده شناخته کارت شماره";
            break;
        case 15:
            text += "میباشد ناشناخته کننده صادر";
            break;
        case 17:
            text += "شد صرفنظر تراکنش انجام از";
            break;
    }
}

```

```
case 19:
    text += "تراکنش مجدد ورود";
    break;
case 20:
    text += "است نامعتبر پاسخ کد";
    break;
case 21:
    text += "نمیگیرد انجام عملی هیچ";
    break;
case 22:
    text += "(System Malfunction) نادرست عملکرد";
    break;
case 25:
    text += "نشد پیدا رکورد";
    break;
case 30:
    text += "است نامعتبر پیام قالب";
    break;
case 31:
    text += "است نامعتبر کارت پذیرنده";
    break;
case 32:
    text += "نشده، پرداخت مشتری به کامل بصورت وجه که آن دلیل به";
    text += "است شده صادر آن اصلاحیه تراکنش";
    break;
case 33:
    text += "است شده سپری کارت انقضای تاریخ";
    break;
case 34:
    text += "NOT approval";
    break;
case 36:
    text += "است شده محدود کارت";
    break;
case 38:
    text += "شناسائی شماره ورود (PIN) است گذشته مجاز حد از";
    text += "فردی";
    break;
case 39:
    text += "شد صرفنظر تراکنش انجام از";
    break;
case 40:
    text += "نامشخص تراکنش";
    break;
case 41:
    text += "است موقت مسدود یا شده مفقود کارت";
    break;
case 42:
    text += "است دائم مسدود یا دزدیده کارت";
    break;
case 51:
    text += "وجه کمبود";
    break;
case 54:
    text += "است رسیده پایان به کارت از استفاده تاریخ";
    break;
case 55:
    text += "است نامعتبر رمز";
    break;
case 56:
    text += "نامعتبر کارت";
    break;
case 57:
    text += "مجاز غیر تراکنش";
```

```
        break;
    case 58:
        text += ".مجاز غیر تراکنش";
        break;
    case 61:
        text += ".مجاز حد از بیش مبلغ";
        break;
    case 62:
        text += ".است شده محدود کارت";
        break;
    case 64:
        text += ".شد صرفنظر تراکنش انجام از";
        break;
    case 65:
        text += ".شد صرفنظر تراکنش انجام از";
        break;
    case 66:
        text += ".نیست فعال حساب شماره";
        break;
    case 67:
        text += ".کارت Capture شد";
        break;
    case 68:
        text += ".است آمده تاخیر با دریافتی جواب";
        break;
    case 75:
        text += ".شد صرفنظر تراکنش انجام از";
        break;
    case 76:
        text += ".شد صرفنظر تراکنش انجام از";
        break;
    case 77:
        text += ".شد صرفنظر تراکنش انجام از";
        break;
    case 78:
        text += ".نیستد فعال کارت";
        break;
    case 79:
        text += ".است نشده تعریف حساب";
        break;
    case 80:
        text += ".است اشکال دارای تراکنش پردازش";
        break;
    case 84:
        text += ".پاسخ دریافت عدم";
        break;
    case 85:
        text += ".شماره Originator است نامعتبر";
        break;
    case 90:
        text += ".پردازش درحال Cutoff";
        break;
    case 91:
        text += ".پاسخ دریافت عدم";
        break;
    case 92:
        text += ".است نامعتبر کارت کننده صادر";
        break;
    case 93:
        text += ".است نشده کامل تراکنش";
        break;
    case 94:
        text += ".است تکراری تراکنش";
        break;
```



```

        case 96:
            text += "شد. صرفنظر تراکنش انجام از";
            break;
        default:
            text += " . . ";
            break;
    }

    return text;
}

}
}

```

راهنمای تست سرویس:

برای تست سرویس از اسکریپت زیر استفاده می شود. توجه شود که فایل JQuery باید کنار این فایل باشد. این فایل شامل یک سری تنظیمات است که مشخص می کند درخواست به چه کارت خوانی ارسال شود.

تمامی فایل های مورد نیاز برای تست سرویس در پوشه test قرار داده شده است.

مشخص کردن نوع ارسال داده به سمت پورت سیستم (پورته که سرویس منتظر درخواست است)	type: "POST"
مشخص کردن نوع داده های دریافتی از پورت مورد نظر.	dataType: "json"
داده های که به سمت سرویس ارسال می شود.	data
آی پی کارت خوان - در صورتی که کارت خوان در شبکه داخلی باشد این فیلد پر می شود.	IP
پورت کارت خوان - پورته که دستگاه کارت خوان از طریق آن به سیستم متصل است.	Port
نوع بانک - برای بانک پارسیان عبارت parsian و برای بانک شهر عبارت shahr وارد می شود.	Type
مبلغی که باید در کارت خوان پرداخت شود.	Amount
مدت زمانی که سرویس منتظر می ماند که کارت خوان پاسخ دهد.	Timeout
اطلاعات و توضیحات پرداخت - مانند شهرداری همدان.	AdditionalData
شناسه قبض سند مالی قابل پرداخت	BillId
شناسه پرداخت سند مالی قابل پرداخت.	PaymentId
ارسال دستور پرینت برای دستگاه کارت خوان.	Print
آدرس و پورت سرویس - آدرسی که سرویس منتظر درخواست است.	url
خروجی حاصل شده از اجرای درخواست که توسط سرویس تولید شده.	response
فیلدی در پاسخ سرویس که مشخص می کند تراکنش به چه صورت بوده است.	'ErrorMessage'

```

<html>

<head>
<meta charset="UTF-8" />

<script src="jquery-2.1.3.js"></script>
</head>
<body>
<script>

```

```
function func() {
```

```

//InstallUtil.exe MahyapardazPOS.exe
// type: parsian or shahr
$.ajax({
    type: "POST",
    dataType:"json",
    async:false,
    data:
{"IP":"","Port":"COM5","Type":"parsian","Amount":"1000","TimeOut":10,"Additional
Data":"شهرداری
همدان","BillId":11000001165,"PaymentId":0000000195939,"Print":true},
    url: 'http://localhost:8080',
    success: function (response) {
        $('#field').html(response);

        console.log(response);
        alert(response['ErrorMessage']);
    },
    error: function (xhr, ajaxOptions, thrownError) {
        alert('در قسمت سرویس های ویندوز اجرا '+ 'Mahya Pos Bridg'+ ' لطفا سرویس '
        'کنید');
        /*alert(thrownError);*/
    }
});
}

</script>

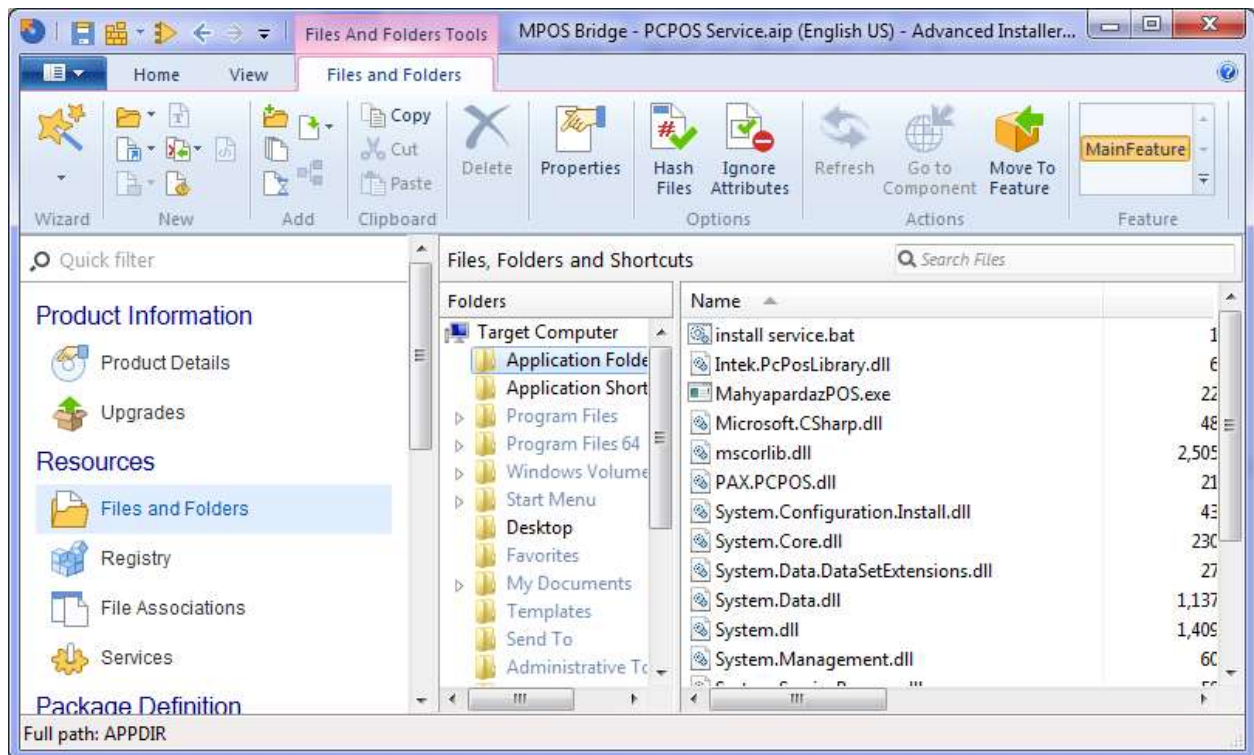
<button onclick="func();">Click Me!!!</button><hr>
Data: <div id="field"></div>

</body>
</html>

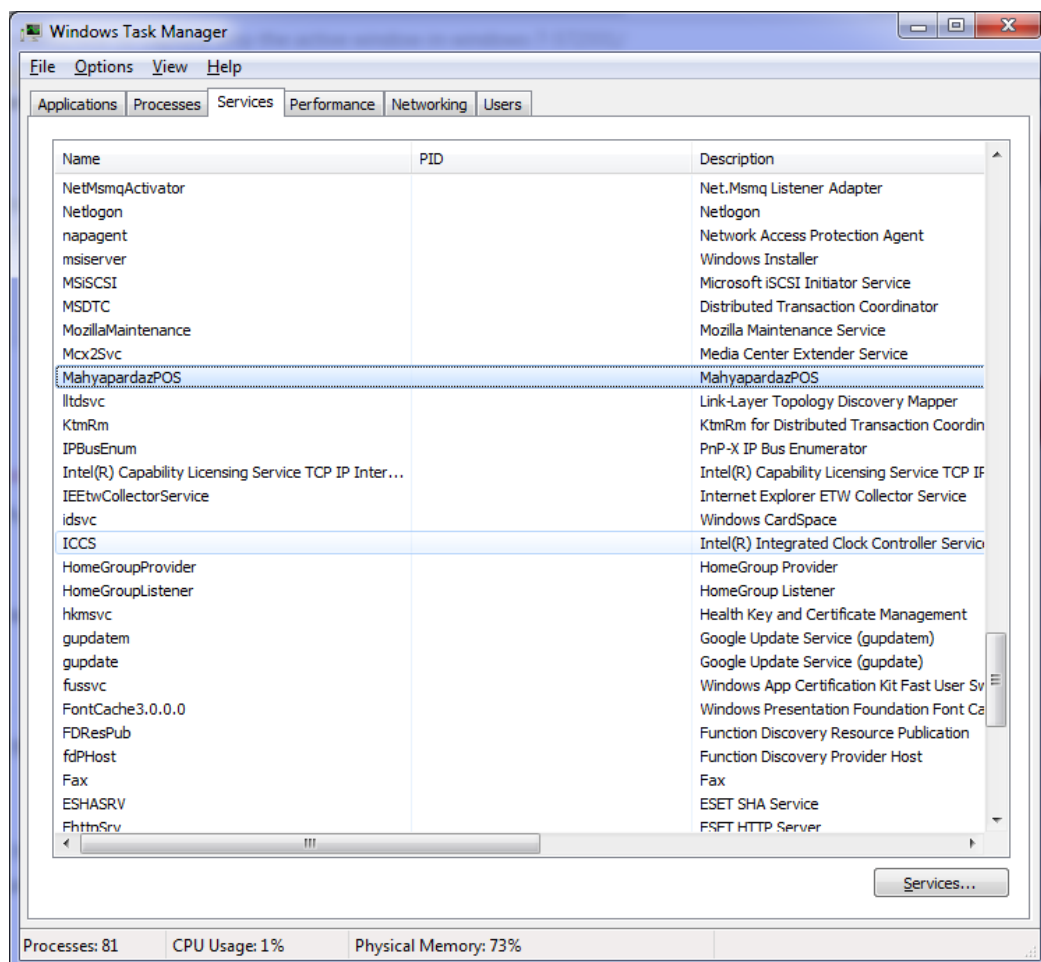
```

راهنمای ساخت برنامه نصبی تحت ویندوز / نصب / حذف

برای ساخت فایل نصبی سرویس از برنامه Advanced Installer نسخه ۱۲٫۸ استفاده شده است. کافی است پوشه release پروژه سی شارپ را در این برنامه اضافه کنید و نوع برنامه را نصب سرویس تعیین کنید.



فایل نصبی به صورت پیش فرض سرویس مورد نظر را در سیستم اجرا می کند؛ با این حال در صورتی که سرویس اجرا نشد باید سرویس را از قسمت سرویس های ویندوز اجرا کنید.



کافی است بر روی سرویس مورد نظر راست کلیک کرده و start service را انتخاب کنید.

نصب سرویس:

فایل نصبی، سرویس مورد نظر را نصب می کند؛ با این وجود می توانید با استفاده از فایل install service.bat در پوشه خود برنامه سرویس را نصب کنید.

```
@ECHO OFF

REM The following directory is for .NET 4.0
set DOTNETFX2=%SystemRoot%\Microsoft.NET\Framework\v4.0.30319
set PATH=%PATH%;%DOTNETFX2%

echo Installing IEPPAMS Win Service...
echo -----
C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil
"%~dp0MahyapardazPOS.exe"
net start "MahyapardazPOS"
echo -----
pause
echo Done.
```

حذف سرویس:

برای حذف سرویس فایل uninstall service.bat را اجرا کنید.

```
@ECHO OFF

REM The following directory is for .NET 4.0
set DOTNETFX2=%SystemRoot%\Microsoft.NET\Framework\v4.0.30319
set PATH=%PATH%;%DOTNETFX2%

echo Installing IEPPAMS Win Service...
echo -----
net stop "MahyapardazPOS"
C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil /u
"%~dp0MahyapardazPOS.exe"

echo -----
pause
echo Done.
```

طرح و تحلیل : محمد صالح صالحی

تولید : سعید دادخواه – مجتبی اشکو فراز

تهیه کننده مستند : سعید داد خواه