

## SSP1126.PcPos (C#)

ابزار ارتباط رایانه با کارتخوان (پورت سریال، USB و شبکه)

مستند فنی

نگارش ۱،۲

سابقه بازنگری			
تاریخ	نگارش	شرح	اقدام کننده
۹۵/۱۰/۱۱	۱,۰	ایجاد	اداره توسعه راهکارهای پرداخت
۹۵/۱۱/۲۵	۱,۱	ویرایش	اداره توسعه راهکارهای پرداخت
۹۶/۰۳/۱۳	۱,۲	ویرایش	اداره توسعه راهکارهای پرداخت
۹۶/۰۸/۰۱	۱,۳	ویرایش	علی احمدوند
۹۶/۰۸/۳۰	۱,۳	ویرایش	علی احمدوند
۹۷/۰۵/۰۱	۱,۴,۰,۷	ویرایش	علی احمدوند
۹۷/۰۸/۰۱	۱,۴,۱,۰	- اضافه شدن خرید چند حسابی با انتخاب شماره شبا - نمایش شماره کارت هشت شده با الگوریتم SHA256	علی احمدوند
۹۷/۰۹/۱۹	۱,۴,۲,۰	- اضافه شدن قابلیت انجام تراکنش سرویس synch پرداخت بصورت	علی احمدوند

نشر			
امضا	تاریخ	مرجع	سمت
	۹۵/۱۰/۱۱	اداره توسعه راهکارهای پرداخت	تهیه کننده
			تضمین کیفیت

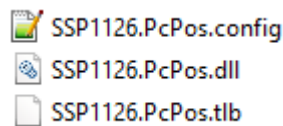
## فهرست

۵	۱. بسته حاوی نرم افزار
۵	۲. پیش نیازها
۵	۳. عملکرد هر بخش
۵	۳,۱ SSP1126.PcPos.tlb
۵	۳,۲ SSP1126.PcPos.config
۶	۳,۳ SSP1126.PcPos.dll
۷	۴. انواع روشهای ارتباطی رایانه با دستگاه کارخوان
۷	۵. کد نویسی
۷	۵,۱ متدها
۲۲	۵,۲ رویداد
۲۳	۶. مراحل اجرای تراکنشها
۲۳	۶,۱ رویه اصلی مدل Async
۲۴	۶,۲ رویه اصلی مدل Sync
۲۵	۷. ساختار اجرایی برنامه نمونه
۲۵	۷,۱ پیکربندی
۲۷	۷,۲ گزارشات
۳۰	۷,۳ شارژ
۳۱	۷,۴ قبض همراه اول
۳۲	۷,۵ قبض تلفن استان
۳۳	۷,۶ سایر قبوض

۳۴.....	۷,۷ خرید
۳۶.....	۷,۸ موجودی
۳۷.....	۷,۹ سرویس پرداخت
۳۸.....	۸. ابزار رمزنگاری (Hash Helper)
۳۹.....	۹. ایجاد تراکنش
۴۱.....	۱۰. راهنما
۴۵.....	۱۱. مقادیر کدهای پاسخ (Response Code)

## نرم افزار ارتباطی PC-POS

## ۱. بسته حاوی نرم افزار



در یک نگاه این نرم افزار در چند (فایل) ارائه شده است. فایل هایی که عدم وجود هریک باعث نقصان در کارکرد نهایی خواهد شد.

- SSP1126.PcPos.config
- SSP1126.PcPos.tlb
- Sep.Pc2Pos.config

## ۲. پیش نیازها

جهت استفاده از این نرم افزار، لازم است:

- حداقل .NET Framework 4.0. روی کامپیوتر مربوطه نصب شده باشد.
- POS در پیکربندی ارتباط با صندوق باشد.
- POS به شکل صحیح با کامپیوتر مرتبط باشد.
- درگاه ارتباطی POS با PC مشخص باشد. (بعنوان مثال COM1 یا ارتباط Ethernet)

## ۳. عملکرد هر بخش

عملکرد هریک از بخش های نامبرده، به شرح ذیل می باشد:

## ۳,۱ SSP1126.PcPos.tlb

این فایل جهت استفاده نمودن از dll در محیط هایی مانند VB6 مورد استفاده قرار می گیرد.

## ۳,۲ SSP1126.PcPos.config

حاوی تنظیمات ابزار است.

```
<appSettings>
  <add key="MinimumAmount" value="1000" />
  <add key="LogPath" value="D:\Log\\"/>
</appSettings>
```

- برای لاگ زدن در بخش **appSettings**، در **value** مربوط به **key="LogPath"**، آدرسی که می‌خواهید لاگ در آنجا ثبت شود را قرار دهید. در صورتی که این مقدار خالی باشد، در محلی که **Dll** قرار داده شده است، لاگ ثبت می‌شود.
- برای مشخص کردن حداقل مقدار ریالی هر تراکنش در بخش **appSettings**، در **value** مربوط به **key="MinimumAmount"**، حداقل مقدار تراکنش خرید را قرار دهید. این مبلغ نباید کمتر از ۱۰۰۰ ریال باشد.

### ۳,۳ SSP1126.PcPos.dll

این فایل جهت ارتباط PC با POS طراحی و تولید شده است. در این فایل با استفاده از موارد تعریف شده در **Namespace** های زیر می‌توانید با دستگاه کارخوان در ارتباط باشید:

- **SSP1126.PcPos.BaseClasses**: جهت استفاده از متدها و **Event** ها
- **SSP1126.PcPos.Infrastructure**: جهت استفاده از کلاس‌ها و **Enum** های ذکر شده در ادامه

با استفاده از این **DLL** امکان انجام تراکنش های زیر امکان پذیر می‌باشد:

- تراکنش خرید
  - با ارسال مبلغ از سمت رایانه به دستگاه کارخوان، تراکنش شروع می‌شود و پس از انجام تراکنش پاسخ به رایانه برگردانده می‌شود.
  - در ابتدا با ارسال درخواست از سمت رایانه، پیام "لطفا کارت بکشید"، روی کارخوان نمایش داده می‌شود، بعد از کشیده شدن کارت، اطلاعات ترمینال و اطلاعات کارت (به دو صورت هش شده و ماسک) و اطلاعات سگمنت‌های کارت به رایانه ارسال می‌شود، بعد از این مرحله مبلغ از رایانه به کارخوان فرستاده شده و بعد از انجام تراکنش، پاسخ به رایانه برگردانده می‌شود. از این روش خرید، برای کارت‌های سگمنتی و یا مواردی که مبلغ بر طبق شماره کارت محاسبه می‌شود، می‌توان استفاده نمود.
- تراکنش موجودی
  - تراکنش قبض همراه اول
  - استعلام قبض
  - پرداخت قبض
  - ارسال دستور پرداخت به کارخوان
- تراکنش قبض تلفن استان
  - استعلام قبض
  - پرداخت قبض
  - ارسال دستور پرداخت به کارخوان
- تراکنش سایر قبوض
  - استعلام قبض
  - پرداخت قبض

- ارسال دستور پرداخت به کارتخوان
- شارژ
  - شارژ تکی
  - شارژ مستقیم
- گزارشات
  - بر طبق سریال تراکنش
  - در یک بازه زمانی به تفکیک نوع تراکنش
  - در یک بازه زمانی به تعداد و مبلغ انواع تراکنشها
- تست ارتباط با مرکز
- دریافت وضعیت مجوز تراکنشها (کدام تراکنشها برای کاربر فعال هستند)

#### ۴. انواع روشهای ارتباطی رایانه با دستگاه کارتخوان

- Network
- RS232
- USB

#### ۵. کد نویسی

بعد از اضافه کردن SSP1126.PcPos.dll به برنامه، امکان استفاده از آن فراهم خواهد شد.

##### ۵.۱ متدها

شی اصلی **PcPosFactory** است که دارای توابع زیر می باشد و بعد از ایجاد یک نمونه در ابتدا امکان استفاده از آنها ایجاد می شود:  
در ابتدای هر تراکنش اگر نوع ارتباطی RS232 و یا USB باشد از متد زیر استفاده می شود:

• **bool SetCom(string comPort)**

در این متد:

- **string comPort**: نشان دهنده پورت Com و یا USB متصل به رایانه می باشد و دارای مقداری مانند "COM4" می باشد.

در صورتی که برای ارتباط بین کارتخوان و رایانه نوع ارتباط شبکه را انتخاب می نمایید، در ابتدای هر تراکنش از متد زیر استفاده می شود:

• **bool SetLan(string ip)**

در این متد:

- `string ip`: نشان دهنده Ip دستگاه کارتخوان می باشد. پورت ارتباطی برابر با ۱۱۹۷ به صورت ثابت در نظر گرفته شده است و قابل تغییر نمی باشد.

#### • `void Initialization(ResponseLanguage language, int timeout, AsyncType asyncType)`

بعد از تعیین نوع ارتباطی، متدی که صدا زده می شود، تابع `Initialization` است. با استفاده از این تابع ویژگی های زیر تنظیم می شوند:

۱. `ResponseLanguage language`: زبان پاسخ دریافتی از دستگاه کارتخوان می تواند فارسی یا انگلیسی باشد.
۲. `int timeout`: مشخص کننده مدت زمان انتظار برای دریافت پاسخ از دستگاه کارتخوان به ثانیه. در صورتی که `timeout` برابر "۰" مقداردهی شود مقدار آن ۱۲۰ ثانیه در نظر گرفته می شود و بعد از این مدت تراکنش منقضی می شود.
۳. `AsyncType asyncType`: توابع تعریف شده می توانند به صورت `Sync` و یا `Async` استفاده شوند. در مدل `Sync` متد بعد از دریافت پاسخ از دستگاه کارتخوان مقدار را در خروجی خود به کاربر بازمی گرداند. در مدل `Async`، رویکرد `Event-Driven` است. خروجی متد نال است و زمانی که کارت کشیده می شود `CardSwiped Event` فراخوانی می شود و هنگامی که تراکنش پایان یافت، `PosResultReceived Event` فراخوانی می شود.

#### • `PosResult Balance()`

از این متد، برای انجام تراکنش موجودی استفاده می شود. در پاسخ این درخواست، فیلدهای زیر از کلاس `PosResult` مقدار می گیرند:

- `ResponseCode`: کد پاسخ
- `ResponseDescription`: شرح پاسخ
- `CardNumberMask`: شماره کارت به صورت ماسک شده
- `CardNumberHash_Sha1`: شماره کارت به صورت هش شده
- `TraceNumber`: شماره پیگیری
- `TxnDate`: تاریخ و زمان انجام تراکنش
- `RRN`: شماره مرجع
- `TerminalId`: شماره ترمینال
- `SerialId`: شماره سریال

در پاسخ این تراکنش موجودی کارت در هیچ صورتی سمت رایانه باز نمی گردد.

#### • `PosResult GetBillProperties(string billIds, string paymentIds)`

از این متد، برای به دست آوردن هزینه و نوع سایر قبوض استفاده می شود.

- `string billIds`: شناسه قبض
- `string paymentIds`: شناسه پرداخت



در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- BillType: نوع قبض
- BillAmount: هزینه قبض

• **PosResult BillPayment(string billId, string paymentId, string additionalData, string referenceData)**

از این متد، برای انجام تراکنش سایر قبوض استفاده می شود.

- billId: string شناسه قبض
- paymentId: string شناسه پرداخت
- additionalData: string جهت چاپ داده روی پرینت استفاده می شود. به "فرمت چاپ اطلاعات روی پرینت" در راهنما مراجعه نمایید.
- referenceData: string جهت ارسال اطلاعات به سویچ استفاده می شود و حداکثر طول آن ۱۰۰ کاراکتر می باشد. مقدار این فیلد در اطلاعات تراکنش ذخیره می شود. در این فیلد از کاراکترهای @ و = نمی توان استفاده نمود. برای چاپ مقدار ذخیره شده در این فیلد روی رسید، باید این مقدار در additionalData نیز به عنوان ورودی داده شود.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- CardNumberMask: شماره کارت به صورت ماسک شده
- CardNumberHash\_Sha1: شماره کارت به صورت هش شده
- TraceNumber: شماره پیگیری
- TxnDate: تاریخ و زمان انجام تراکنش
- RRN: شماره مرجع
- TerminalId: شماره ترمینال
- SerialId: شماره سریال
- ReqAmount: مبلغ پرداختی

• **PosResult BillRequest()**

از این متد، برای انجام تراکنش سایر قبوض استفاده می شود. با استفاده از این متد شناسه قبض و پرداخت، توسط کاربر روی دستگاه کارتخوان وارد می شود. در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- CardNumberMask: شماره کارت به صورت ماسک شده
- CardNumberHash\_Sha1: شماره کارت به صورت هش شده
- TraceNumber: شماره پیگیری
- TxnDate: تاریخ و زمان انجام تراکنش
- RRN: شماره مرجع
- TerminalId: شماره ترمینال
- SerialId: شماره سریال
- ReqAmount: مبلغ پرداختی

• **PosResult** MCIBillInquiry(**string** mciNumber, **PhoneBillType** mciBillType)

از این متد، برای انجام تراکنش استعلام قبض همراه اول استفاده می‌شود.

- **string** mciNumber: شماره موبایل (به صورت کامل وارد می‌شود. به عنوان مثال ۰۹۱۲۱۲۱۰۵۴۴)
- **PhoneBillType** mciBillType: نوع پرداخت. به صورت میان دوره و یا پایان دوره.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- TerminalId: شماره ترمینال
- ReqAmount: مبلغ پرداختی
- CardNumberMask: شماره کارت به صورت ماسک شده
- CardNumberHash\_Sha1: شماره کارت به صورت هش شده
- TxnDate: تاریخ و زمان انجام تراکنش

• **PosResult** MCIBillPayment(**string** mciNumber, **PhoneBillType** mciBillType, **string** additionalData, **string** referenceData)

از این متد، برای انجام تراکنش پرداخت قبض همراه اول استفاده می‌شود.

- **string** mciNumber: شماره موبایل (به صورت کامل وارد می‌شود. به عنوان مثال ۰۹۱۲۱۲۱۰۵۴۴)
- **PhoneBillType** mciBillType: نوع پرداخت؛ به صورت میان دوره و یا پایان دوره.

- `additionalData` `string`: جهت چاپ داده روی پرینت استفاده می‌شود. به "فرمت چاپ اطلاعات روی پرینت" در راهنما مراجعه نمایید.
- `referenceData` `string`: جهت ارسال اطلاعات به سویچ استفاده می‌شود و حداکثر طول آن ۱۰۰ کاراکتر می‌باشد. مقدار این فیلد در اطلاعات تراکنش ذخیره می‌شود. در این فیلد از کاراکترهای @ و = نمی‌توان استفاده نمود. برای چاپ مقدار ذخیره شده در این فیلد روی رسید، باید این مقدار در `additionalData` نیز به عنوان ورودی داده شود.

در پاسخ این درخواست، فیلدهای زیر از کلاس `PosResult` مقدار می‌گیرند:

- `ResponseCode`: کد پاسخ
- `ResponseDescription`: شرح پاسخ
- `CardNumberMask`: شماره کارت به صورت ماسک شده
- `CardNumberHash_Sha1`: شماره کارت به صورت هش شده
- `TraceNumber`: شماره پیگیری
- `TxnDate`: تاریخ و زمان انجام تراکنش
- `RRN`: شماره مرجع
- `TerminalId`: شماره ترمینال
- `SerialId`: شماره سریال
- `ReqAmount`: مبلغ پرداختی

#### • `PosResult MCIBillRequest()`

از این متد، برای انجام تراکنش پرداخت قبض همراه اول استفاده می‌شود. با استفاده از این متد شماره تلفن همراه، توسط کاربر روی دستگاه کارخوان وارد می‌شود.

در پاسخ این درخواست، فیلدهای زیر از کلاس `PosResult` مقدار می‌گیرند:

- `ResponseCode`: کد پاسخ
- `ResponseDescription`: شرح پاسخ
- `CardNumberMask`: شماره کارت به صورت ماسک شده
- `CardNumberHash_Sha1`: شماره کارت به صورت هش شده
- `TraceNumber`: شماره پیگیری
- `TxnDate`: تاریخ و زمان انجام تراکنش
- `RRN`: شماره مرجع
- `TerminalId`: شماره ترمینال
- `SerialId`: شماره سریال
- `ReqAmount`: مبلغ پرداختی

• **PosResult TCIBillInquiry(string phoneNumber, PhoneBillType phoneBillType)**

از این متد، برای انجام تراکنش استعلام قبض تلفن استان استفاده می‌شود.

- **string phoneNumber**: شماره تلفن (بدون صفر وارد می‌شود. به عنوان مثال ۲۳۳۲۳۳۵۶۵۸ برای استان سمنان)
- **PhoneBillType mciBillType**: نوع پرداخت؛ به صورت میان دوره و یا پایان دوره.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- **ResponseCode**: کد پاسخ
- **ResponseDescription**: شرح پاسخ
- **TerminalId**: شماره ترمینال
- **ReqAmount**: مبلغ پرداختی
- **CardNumberMask**: شماره کارت به صورت ماسک شده
- **CardNumberHash\_Sha1**: شماره کارت به صورت هش شده
- **TxnDate**: تاریخ و زمان انجام تراکنش

• **PosResult TCIBillPayment(string phoneNumber, PhoneBillType phoneBillType , string additionalData, string referenceData)**

از این متد، برای انجام تراکنش پرداخت قبض تلفن استان استفاده می‌شود.

- **string phoneNumber**: شماره تلفن (بدون صفر وارد می‌شود. به عنوان مثال ۲۳۳۲۳۳۵۶۵۸ برای استان سمنان)
- **PhoneBillType mciBillType**: نوع پرداخت؛ به صورت میان دوره و یا پایان دوره.
- **string additionalData**: جهت چاپ داده روی پرینت استفاده می‌شود. به " فرمت چاپ اطلاعات روی پرینت " در راهنما مراجعه نمایید.
- **string referenceData**: جهت ارسال اطلاعات به سویچ استفاده می‌شود و حداکثر طول آن ۱۰۰ کاراکتر می‌باشد. مقدار این فیلد در اطلاعات تراکنش ذخیره می‌شود. در این فیلد از کاراکترهای @ و = نمی‌توان استفاده نمود. برای چاپ مقدار ذخیره شده در این فیلد روی رسید، باید این مقدار در **additionalData** نیز به عنوان ورودی داده شود.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- **ResponseCode**: کد پاسخ
- **ResponseDescription**: شرح پاسخ
- **CardNumberMask**: شماره کارت به صورت ماسک شده
- **CardNumberHash\_Sha1**: شماره کارت به صورت هش شده
- **TraceNumber**: شماره پیگیری

- TxnDate: تاریخ و زمان انجام تراکنش
- RRN: شماره مرجع
- TerminalId: شماره ترمینال
- SerialId: شماره سریال
- ReqAmount: مبلغ پرداختی

#### • PosResult TCIBillRequest()

از این متد، برای انجام تراکنش پرداخت قبض تلفن استان استفاده می‌شود. با استفاده از این متد شماره تلفن و نوع پرداخت، توسط کاربر روی دستگاه کارتهخوان وارد می‌شود.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- CardNumberMask: شماره کارت به صورت ماسک شده
- CardNumberHash\_Sha1: شماره کارت به صورت هش شده
- TraceNumber: شماره پیگیری
- TxnDate: تاریخ و زمان انجام تراکنش
- RRN: شماره مرجع
- TerminalId: شماره ترمینال
- SerialId: شماره سریال
- ReqAmount: مبلغ پرداختی

#### • PosResult PinCharge(string additionalData, string referenceData)

از این متد، برای انجام تراکنش شارژ رمز تکی استفاده می‌شود.

- string additionalData: جهت چاپ داده روی پرینت استفاده می‌شود. به " فرمت چاپ اطلاعات روی پرینت " در راهنما مراجعه نمایید.
- string referenceData: جهت ارسال اطلاعات به سویچ استفاده می‌شود و حداکثر طول آن ۱۰۰ کاراکتر می‌باشد. مقدار این فیلد در اطلاعات تراکنش ذخیره می‌شود. در این فیلد از کاراکترهای @ و = نمی‌توان استفاده نمود. برای چاپ مقدار ذخیره شده در این فیلد روی رسید، باید این مقدار در additionalData نیز به عنوان ورودی داده شود.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- CardNumberMask: شماره کارت به صورت ماسک شده
- CardNumberHash\_Sha1: شماره کارت به صورت هش شده
- TraceNumber: شماره پیگیری
- TxnDate: تاریخ و زمان انجام تراکنش
- RRN: شماره مرجع
- TerminalId: شماره ترمینال
- SerialId: شماره سریال
- ReqAmount: مبلغ پرداختی
- ChargePin: رمز شارژ (در صورتی که مجوز دریافت رمز سمت رایانه وجود داشته باشد این مقدار به عنوان پاسخ برگردانده می شود)
- ChargeEmergencyNumber: شماره پشتیبانی
- ChargeSerial: سریال شارژ

• **PosResult TopupCharge(string mobileNumber , string additionalData, string referenceData)**

از این متد، برای انجام تراکنش شارژ مستقیم استفاده می شود.

- string mciNumber: شماره موبایل (به صورت کامل وارد می شود. به عنوان مثال ۰۹۳۶۰۵۳۲۴۶۹)
- string additionalData: جهت چاپ داده روی پرینت استفاده می شود. به " فرمت چاپ اطلاعات روی پرینت " در راهنما مراجعه نمایید.
- string referenceData: جهت ارسال اطلاعات به سویچ استفاده می شود و حداکثر طول آن ۱۰۰ کاراکتر می باشد. مقدار این فیلد در اطلاعات تراکنش ذخیره می شود. در این فیلد از کاراکترهای @ و = نمی توان استفاده نمود. برای چاپ مقدار ذخیره شده در این فیلد روی رسید، باید این مقدار در additionalData نیز به عنوان ورودی داده شود.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- CardNumberMask: شماره کارت به صورت ماسک شده
- CardNumberHash\_Sha1: شماره کارت به صورت هش شده
- TraceNumber: شماره پیگیری
- TxnDate: تاریخ و زمان انجام تراکنش
- RRN: شماره مرجع

- TerminalId: شماره ترمینال
- SerialId: شماره سریال
- ReqAmount: مبلغ پرداختی
- ChargeEmergencyNumber: شماره پشتیبانی
- ChargeSerial: سریال شارژ

در این پروتکل خرید به ۲ صورت قابل انجام می‌باشد.

- خرید تک مرحله‌ای: با اولین دستور، مبلغ روی دستگاه کارتخوان نمایش داده می‌شود.
- خرید ۲ مرحله‌ای: در ابتدا روی دستگاه کارتخوان عبارت "لطفا کارت بکشید" نمایش داده می‌شود و بعد از کشیدن کارت و ارسال شماره کارت و شماره ترمینال، امکان دریافت مبلغ فراهم می‌شود.

برای تراکنش‌های خرید کارت‌های سگمنتی و یا مواردی که برای محاسبه مبلغ نیاز به شماره کارت می‌باشد، از متدهای PosStarterPurchaseInit و PosStarterPurchase در ۲ مرحله استفاده می‌شود. (خرید ۲ مرحله‌ای)

#### • PosResult PosStarterPurchaseInit()

از این متد، برای انجام تراکنش خرید از سمت دستگاه کارتخوان استفاده می‌شود. با صدا زدن این متد عبارت "لطفا کارت بکشید" روی دستگاه کارتخوان نمایش داده می‌شود. در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- CardNumberMask: شماره کارت به صورت ماسک شده
- CardNumberHash\_Sha1\_Sha1: شماره کارت به صورت هش شده با الگوریتم SHA1
- CardNumberHash\_Sha1\_Sha2: شماره کارت به صورت هش شده با الگوریتم SHA256
- TerminalId: شماره ترمینال
- PurchaseTypesDictionary: انواع خرید تعریف شده در کارت را نشان می‌دهد. در صورتی که کارت، عادی باشد فقط "خرید" را نشان می‌دهد و در غیر این صورت انواع خرید به صورت دیکشنری از نوع خرید و Id آن نمایش داده می‌شود. بعد از این متد باید متد PosStarterPurchase صدا زده شود.

در این مرحله روی دستگاه کارتخوان عبارت "در انتظار دریافت مبلغ" نمایش داده می‌شود و کارتخوان ۴۰ ثانیه در انتظار دریافت مبلغ باقی می‌ماند.

#### • PosResult PosStarterPurchase(string mainAmount, string amounts , string additionalData, string referenceData, int purchaseType, string ValidTerminalID = null, string PurchaseID =null)

از این متد، برای ادامه انجام تراکنش خرید از سمت کارخوان استفاده می‌شود.

- **string mainAmount**: در صورتی که تراکنش به صورت تک حسابی باشد، در این فیلد مبلغ قرار می‌گیرد. در صورتی که تراکنش به صورت چند حسابی باشد، در این فیلد جمع مبالغ قرار می‌گیرد.
- **string amounts**: در صورتی که تراکنش به صورت تک حسابی باشد، در این فیلد رشته خالی یا نال قرار می‌گیرد. در صورتی که تراکنش به صورت چند حسابی باشد، در این فیلد مبالغ به صورت یک رشته جدا شده با کاما "," قرار می‌گیرند. (به عنوان مثال ۲۰۰۰,۱۰۰۰). نوع دیگری از خرید چند حسابی قابل انجام است که در آن ترتیب مبالغ اهمیتی ندارد و می‌توان مشخص کرد که برای هر شماره شبا چه مبلغی واریز گردد که در انصورت می‌بایست این پارامتر را بصورت (شبا۱،مبلغ۱،شبا۲،مبلغ۲.....) مقدار دهی کرد. برای فعال نمودن این ویژگی لازم است با واحد پشتیبانی تماس بگیرید.
- **string additionalData**: جهت چاپ داده روی پرینت استفاده می‌شود. به " فرمت چاپ اطلاعات روی پرینت " در راهنما مراجعه نمایید.
- **string referenceData**: جهت ارسال اطلاعات به سویچ استفاده می‌شود و حداکثر طول آن ۱۰۰ کاراکتر می‌باشد. مقدار این فیلد در اطلاعات تراکنش ذخیره می‌شود. در این فیلد از کاراکترهای @ و = نمی‌توان استفاده نمود. برای چاپ مقدار ذخیره شده در این فیلد روی رسید، باید این مقدار در **additionalData** نیز به عنوان ورودی داده شود.
- **string purchaseType**: در صورتی مربوط به کارت سگمنتی باشد، **Id** نوع خرید انتخابی در این فیلد قرار می‌گیرد و در غیر این صورت با "۰" مقداردهی شود.
- **String ValidTerminalID**: در صورتی که بخواهید دستگاه پوز را ملزم به اعتبارسنجی شماره ترمینال کنید، می‌بایست به وسیله این پارامتر شماره ترمینال معتبر را به پوز ارسال نمایید. در صورتی که مقدار این پارامتر با شماره ترمینال روی دستگاه پوز تفاوت داشته باشد، پیغام خطایی اعلام می‌شود و تراکنش لغو می‌گردد. خالی بودن این پارامتر به معنای عدم نیاز به اعتبارسنجی شماره ترمینال می‌باشد.
- **String PurchaseID**: برای انجام تراکنش خرید شناسه دار می‌بایست این پارامتر را با شناسه خرید مقداردهی مقداردهی و ارسال کرد. در غیر اینصورت برای تراکنش خرید معمولی مقدار آن بصورت پیشفرض "null" است.

در پاسخ این درخواست، فیلدهای زیر از کلاس **PosResult** مقدار می‌گیرند:

- **ResponseCode**: کد پاسخ
- **ResponseDescription**: شرح پاسخ
- **CardNumberMask**: شماره کارت به صورت ماسک شده
- **CardNumberHash\_Sha1\_Sha1**: شماره کارت به صورت هش شده با الگوریتم SHA1
- **TraceNumber**: شماره پیگیری
- **TxnDate**: تاریخ و زمان انجام تراکنش
- **RRN**: شماره مرجع
- **TerminalId**: شماره ترمینال
- **SerialId**: شماره سریال
- **ReqAmount**: مبلغ اصلی
- **AffectiveAmount**: مبلغ پرداختی (بعد از اعمال تخفیف و ...)



```
• PosResult PcStarterPurchase(string mainAmount, string amounts , string additionalData,
string referenceData, string ValidTerminalID = null)
```

از این متد، برای ادامه انجام تراکنش خرید از سمت رایانه استفاده می‌شود. (همزمان با ارسال درخواست خرید، مبلغ روی دستگاه کارتخوان نمایش داده می‌شود. (خرید تک مرحله‌ای)

- **string mainAmount**: در صورتی که تراکنش به صورت تک حسابی باشد، در این فیلد مبلغ قرار می‌گیرد. در صورتی که تراکنش به صورت چند حسابی باشد، در این فیلد جمع مبالغ قرار می‌گیرد.
- **string amounts**: در صورتی که تراکنش به صورت تک حسابی باشد، در این فیلد رشته خالی یا نال قرار می‌گیرد. در صورتی که تراکنش به صورت چند حسابی باشد، در این فیلد مبالغ به صورت یک رشته جدا شده با کاما "," قرار می‌گیرند. (به عنوان مثال ۲۰۰۰,۱۰۰۰)
- **string additionalData**: جهت چاپ داده روی پرینت استفاده می‌شود. به " فرمت چاپ اطلاعات روی پرینت " در راهنما مراجعه نمایید.
- **string referenceData**: جهت ارسال اطلاعات به سویچ استفاده می‌شود و حداکثر طول آن ۱۰۰ کاراکتر می‌باشد. مقدار این فیلد در اطلاعات تراکنش ذخیره می‌شود. در این فیلد از کاراکترهای @ و = نمی‌توان استفاده نمود. برای چاپ مقدار ذخیره شده در این فیلد روی رسید، باید این مقدار در **additionalData** نیز به عنوان ورودی داده شود.
- **String ValidTerminalID**: در صورتی که بخواهید دستگاه پوز را ملزم به اعتبارسنجی شماره ترمینال کنید، می‌بایست به وسیله این پارامتر شماره ترمینال معتبر را به پوز ارسال نمایید. در صورتی که مقدار این پارامتر با شماره ترمینال روی دستگاه پوز تفاوت داشته باشد، پیغام خطایی اعلام می‌شود و تراکنش لغو می‌گردد. خالی بودن این پارامتر به معنای عدم نیاز به اعتبارسنجی شماره ترمینال می‌باشد.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- **ResponseCode**: کد پاسخ
- **ResponseDescription**: شرح پاسخ
- **CardNumberMask**: شماره کارت به صورت ماسک شده
- **CardNumberHash\_Sha1**: شماره کارت به صورت هش شده
- **TraceNumber**: شماره پیگیری
- **TxnDate**: تاریخ و زمان انجام تراکنش
- **RRN**: شماره مرجع
- **TerminalId**: شماره ترمینال
- **SerialId**: شماره سریال
- **ReqAmount**: مبلغ پرداختی
- **AffectiveAmount**: مبلغ پرداختی (بعد از اعمال تخفیف و ...)

```
PosResult GetReport(ReportAction reportAction, ReportFilterType reportFilterType, string
reportFromDate, string reportToDate, string serialId, string terminalPin, string
fromTime, string toTime)
```

از این متد، برای دریافت گزارشات استفاده می‌شود. به دلیل حجم زیاد رکوردهای گزارش ارسالی، گزارشات در بسته های ۳ تایی سمت رایانه ارسال می‌شوند و در صورت برقرار بودن ارتباط بسته بعدی ارسال می‌شود.

انواع تراکنش‌ها برای گزارشگیری به صورت زیر تعریف می‌گردند:

- خرید: کد: ۱
- قبض: کد: ۲
- شارژ رمزی: کد: ۳
- شارژ مستقیم: کد: ۴
- مجموع صورتحساب (تعداد و جمع مبلغ تراکنشهای خرید، شارژ تکی، قبض، شارژ مستقیم): کد: ۵

انواع فیلترهای گزارشگیری به صورت زیر تعریف می‌گردند:

- گزارش بر اساس سریال تراکنش
- گزارش بر اساس بازه تاریخی (حداکثر ۵ روز و تاریخ شروع و پایان باید در یک ماه باشند)
- گزارش در یک تاریخ بر اساس بازه زمانی (.... تا ۲۳:۵۹)

ورودی‌های این متد به صورت زیر می‌باشد:

- **ReportAction reportAction**: مشخص کننده نوع تراکنش برای گزارشگیری می‌باشد. انواع تراکنش‌ها برای دریافت گزارش به شرح زیر می‌باشد:
  - خرید
  - قبض
  - شارژ رمزی
  - شارژ مستقیم
  - مجموع صورتحساب
- **ReportFilterType reportFilterType**: مشخص کننده فیلتر گزارشگیری می‌باشد. انواع تراکنش‌ها جهت دریافت گزارش به شرح زیر می‌باشد:
  - بازه تاریخی (فاصله تاریخ شروع و پایان حداکثر می‌تواند ۵ روز باشد و تاریخ‌ها باید در یک ماه باشند)
  - سریال تراکنش (وقتی نوع تراکنش مجموع صورتحساب باشد، نباید فیلتر برابر سریال انتخاب شود)
- **string reportFromDate**: از تاریخ، فرمت تاریخ به صورت روز/ماه/سال (مانند ۱۳۹۵/۰۷/۰۸) باید وارد شود. در صورتی که نوع فیلتر، بازه تاریخی و یا بازه ساعتی انتخاب شود این فیلد باید حتما مقدار صحیحی داشته باشد. در غیر این صورت می‌تواند به صورت رشته خالی یا نال وارد گردد. (این فیلد در بازه ساعتی به عنوان تاریخ روز استفاده می‌شود)

- **string reportToDate**: تا تاریخ. فرمت تاریخ به صورت روز/ماه/سال (مانند ۱۳۹۵/۰۷/۰۸) باید وارد شود. در صورتی که نوع فیلتر، بازه تاریخی انتخاب شود این فیلد باید حتما مقدار صحیحی داشته باشد. در غیر این صورت می تواند به صورت رشته خالی یا نال وارد گردد.
- **string serialId**: شماره سریال. در صورتی که نوع فیلتر، سریال انتخاب شود این فیلد باید حتما مقدار صحیحی داشته باشد. در غیر این صورت می تواند به صورت رشته خالی یا نال وارد گردد.
- **string terminalPin**: رمز ترمینال. در صورتی که مقداری برای این پارامتر وارد نشود، روی دستگاه کارتخوان بعد از ارسال دستور گزارشگیری، رمز ترمینال باید وارد شود.
- **string fromTime**: از ساعت. فرمت ساعت به صورت دقیقه/ساعت (۱۳:۰۱) می باشد. در صورتی که نوع فیلتر، بازه ساعتی انتخاب شود این فیلد باید حتما مقدار صحیحی داشته باشد.
- **string toTime**: تا ساعت. فرمت ساعت به صورت دقیقه/ساعت (۱۳:۰۱) می باشد. در صورتی که نوع فیلتر، بازه ساعتی انتخاب شود این فیلد باید حتما مقدار صحیحی داشته باشد.

در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می گیرند:

- **ResponseCode**: کد پاسخ
- **ResponseDescription**: شرح پاسخ
- **TerminalId**: شماره ترمینال
- **ReportXml**: گزارش درخواستی در قالب فایل Xml

○ یک نمونه گزارش بازه تاریخی خرید:

```
<?xml version="1.0" encoding="UTF-8"?>
<Transactions>
  <Transation1>
    <TransationType>Purchase</TransationType> (نوع تراکنش)
    <Date>1395/10/12</Date>
    <Time>08:44:10</Time>
    <ShiftNum>0</ShiftNum> (شماره شیفت)
    <TraceNo>000004</TraceNo> (شماره سریال)
    <Amount>1000</Amount>
    <RRN>004970173836</RRN> (شماره مرجع)
    <Bank>سامان</Bank>
    <MaskPan>621986-##-8003</MaskPan>
    <HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB</HashPan>
    <TerminalId>10332839</TerminalId> (شماره ترمینال)
  </Transation1>
  <Transation2>
    <TransationType>Purchase</TransationType>
    <Date>1395/10/12</Date>
    <Time>08:43:51</Time>
```

```

<ShiftNum>0</ShiftNum>
<TraceNo>000003</TraceNo>
<Amount>1000</Amount>
<RRN>004970172692</RRN>
<Bank>سامان</Bank>
<MaskPan>621986-##-8003</MaskPan>
<HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB</HashPan>
<TerminalId>10332839</TerminalId>
</Transation2>
</Transactions>

```

○ یک نمونه گزارش بر اساس شماره سریال خرید:

```

<?xml version="1.0" encoding="UTF-8"?>
<Transactions>
  <Transation1>
    <TransationType>Purchase</TransationType>
    <Date>1395/10/12</Date>
    <Time>08:43:51</Time>
    <ShiftNum>0</ShiftNum>
    <TraceNo>000003</TraceNo>
    <Amount>1000</Amount>
    <RRN>004970172692</RRN>
    <Bank>سامان</Bank>
    <MaskPan>621986-##-8003</MaskPan>
    <HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB</HashPan>
    <TerminalId>10332839</TerminalId>
  </Transation1>
</Transactions>

```

○ یک نمونه گزارش قبض:

```

<?xml version="1.0" encoding="UTF-8"?>
<Transactions>
  <Transation1>
    <TransationType>Bill</TransationType>
    <Date>1395/10/13</Date>
    <Time>15:17:49</Time>
    <ShiftNum></ShiftNum>
    <TraceNo>000035</TraceNo>
    <Amount>1000</Amount>
    <RRN>004978343101</RRN>
    <Bank>سامان</Bank>
  </Transation1>
</Transactions>

```

```

<MaskPan>621986-##-8003</MaskPan>
<HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB</Hash
Pan>
<BillId>7994891706236</BillId> (شناسه قبض)
<PayId>110105</PayId> (شناسه پرداخت)
<TerminalId>10332839</TerminalId>
</Transation1>
</Transactions>

```

○ یک نمونه گزارش جمع صورتحساب:

```

<? xml version="1.0" encoding="UTF-8"?>
< Transactions>
  (تعداد قبض) <BillCount>000000</BillCount>
  (جمع مبالغ قبض) <BillSum></BillSum>
  (تعداد شارژ رمزی) <ChargeCount>000000</ChargeCount>
  (جمع مبالغ شارژ رمزی) <ChargeSum></ChargeSum>
  (تعداد شارژ مستقیم) <TopupChargeCount>000000</TopupChargeCount>
  (جمع مبالغ مستقیم) <TopupChargeSum></TopupChargeSum>
  (تعداد خرید) <BuyCount>000002</BuyCount>
  (جمع مبالغ خرید) <BuySum>2000</BuySum>
  (تعداد شارژ گروهی) <MultiChargeCount>560000</MultiChargeCount>
  (جمع مبالغ شارژ گروهی) <MultiChargeSum>16</MultiChargeSum>
</ Transactions>

```

#### • **PosResult** GetAuthorizedOperations()

این متد تراکنش‌هایی که برای آن‌ها مجوز تعریف شده و از طریق این پروتکل برای این کاربر قابل استفاده هستند را مشخص می‌نماید. در پاسخ این درخواست، فیلدهای زیر از کلاس PosResult مقدار می‌گیرند:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- TerminalId: شماره ترمینال
- IsBillValidOperation: مجوز تراکنش قبض وجود دارد یا خیر.

- IsMCIBillValidOperation: مجوز تراکنش قبض همراه اول وجود دارد یا خیر.
- IsPinChargeValidOperation: مجوز تراکنش خرید شارژ رمزی وجود دارد یا خیر.
- IsTopupChargeValidOperation: مجوز تراکنش شارژ مستقیم وجود دارد یا خیر.
- IsPurchaseValidOperation: مجوز تراکنش خرید وجود دارد یا خیر.
- IsGetReportValidOperation: مجوز گزارشگیری وجود دارد یا خیر.
- IsBalanceValidOperation: مجوز تراکنش موجودی وجود دارد یا خیر.
- IsTCIBillValidOperation: مجوز تراکنش قبض تلفن استان وجود دارد یا خیر.
- POS\_Version : نسخه اپلیکیشن دستگاه کارتخوان را بر می گرداند.

#### • bool ConnectionTest()

این متد ارتباط رایانه با دستگاه کارتخوان و دستگاه با مرکز را مشخص می نماید. در پاسخ این درخواست، فیلدهای زیر از کلاس `PosResult` زیر مقدار می گیرد:

- ResponseCode: کد پاسخ
- ResponseDescription: شرح پاسخ
- TerminalId: شماره ترمینال

#### • bool Dispose()

از این متد برای `Dispose` نمودن شی تعریف شده استفاده می شود.

#### • TransactionType GetTransactionType()

از این متد برای برگرداندن نوع تراکنش انجام شده استفاده می شود.

## ۵.۲ رویداد

در صورتی که نوع همروندی در متد `Initialization` برابر با `Aysnc` انتخاب شود، هنگامی که کارت کشیده می شود، رویداد `CardSwiped` فراخوانی می شود.

هنگامی که پاسخ نهایی از کارتخوان برگردانده می شود، متد `PosResultReceived` فراخوانی می شود.

هر دو رویداد ورودی از جنس `PosResult` دارند که متناسب با تراکنشی که باعث صدا زده شدن این رویدادها شده است فیلدها مقدار می گیرند.

## ۶. مراحل اجرای تراکنش‌ها

## ۶,۱ رویه اصلی مدل Async

Description	EFT-POS	PcPosFactory (SSP1126.PcPos.dll)	Application	
			ایجاد یک نمونه از PcPosFactory و نسبت دادن Eventها	1
			در ابتدای انجام هر تراکنش صدا نمودن متد SetCom و یا SetLan	2
			سپس در ابتدای انجام هر تراکنش صدا نمودن متد Initialization	3
		←←←←←←	صدا زدن متد مربوط به تراکنش درخواستی	4
	نمایش متن مناسب روی دستگاه کارتخوان			5
در تمامی تراکنش‌ها به جز تست ارتباط، دریافت مجوزها، گزارشگیری، استعلام سایر قبوض	کشیدن کارت روی دستگاه کارتخوان	→→→→→→		6
		↑ Raise Event CardSwiped		7
			دریافت اطلاعات کارت (۲ فرمت ماسک شده و هش) و شماره ترمینال	8
فقط در "خرید سمت کارتخوان" در این مرحله مبلغ فرستاده می‌شود.		←←←←←←	ارسال اطلاعات مورد نیاز به کارتخوان	9
در تمامی تراکنش‌ها به جز تست ارتباط، دریافت مجوزها، گزارشگیری، استعلام سایر قبوض	نمایش مبلغ و ورود رمز کارت	→→→→→→		10
		در صورت دریافت پاسخ پس از ارسال تراکنش ↑ Raise Event		12

		PosResultReceived		
--	--	-------------------	--	--

## ۶,۲ رویه اصلی مدل Sync

Description	EFT-POS	PcPosFactory (SSP1126.PcPos.dll)	Application	
			ایجاد یک نمونه از PcPosFactory و نسبت دادن Event ها	1
			در ابتدای انجام هر تراکنش صدا نمودن متد SetCom و یا SetLan	2
			در ابتدای انجام هر تراکنش صدا نمودن متد Initialization	3
		←←←←←←	صدا زدن متد مربوط به تراکنش درخواستی	4
	نمایش متن مناسب روی دستگاه کارتخوان			5
در تمامی تراکنش ها به جز تست ارتباط، دریافت مجوزها، گزارش گیری، استعلام سایر قبوض	کشیدن کارت روی دستگاه کارتخوان	→→→→→→		6
فقط در "خرید سمت کارتخوان" در این مرحله مبلغ فرستاده می شود.		←←←←←←	دریافت اطلاعات کارت در خروجی متد و ارسال اطلاعات مورد نیاز به کارتخوان	7
در تمامی تراکنش ها به جز تست ارتباط، دریافت مجوزها، گزارش گیری، استعلام سایر قبوض	نمایش مبلغ و ورود رمز کارت	→→→→→→		8
			دریافت اطلاعات تراکنش در خروجی متد	9



## ۷. ساختار اجرایی برنامه نمونه

برنامه تست شامل ۸ تب مجزاست.

## ۷،۱ پیکربندی

PC to POS (Version: 1.4.0.1) - □ ×

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.PcPos dll. PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Payment Service **Configurations** Hash Helper

Connection and Language:

Account Type: Single Media Type: Com  
 Resp Language: Persian Port Name: COM4  
 Async Type: Async POS IP: 192.168.14.106

☐ IsBillValid ☐ IsTopupChargeValid ☐ IsBalanceValid  
☐ IsPurchaseValid ☐ IsMCIBillValid ☐ IsTCIBillValid  
☐ IsPinChargeValid ☐ IsReportValid ☐ IsPaymentServiceValid

Get Authorized Operations

Echo Test:

Connection Test: Connection

2. Transaction Result:

This Result came from EFT-POS.

Main Amount: Terminal ID:  
 Affective Amount: Trace Number:  
 Response Code: Serial ID:  
 Message: RRN:  
 Card Number Mask: Transaction Date:  
 Card Number Hash:

Operations:

Reset Page Exit

Ready.

در بخش Configuration، موارد زیر قبل از انجام عملیات باید انتخاب شود:

Media Type: ارتباط از طریق LAN و یا Com (شامل RS232 و یا USB).

Account Type: وضعیت تسهیم به صورت تک حسابی و یا چند حسابی.

Resp Language: زبان پاسخ های دریافتی از دستگاه کارخوان.

Async Type: نوع همروندی. Async یا Sync.

PortName: در صورتی که Media Type برابر Com انتخاب شود فعال می شود و پورت مورد نظر باید انتخاب گردد. اما می توان با استفاده از



دکمه پورت متصل به پروتکل SSP1126 را بصورت خودکار انتخاب کرد.

Pos IP: نشان دهنده IP دستگاه کارخوان می باشد. در صورتی که Media Type برابر Network انتخاب شود فعال می شود.

در قسمت پایین با زدن کلید GetAuthorizedOperation، مجوزهایی که برای این ترمینال تعریف شده باشند، تیک می خورند.

با استفاده از کلید Connection می توان تست ارتباط با سوییچ را انجام داد و نتیجه تست در قسمت نتایج نشان داده می شود.

## ۷,۲ گزارشات

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Configurations

1. Getting Repoert from EFT-POS:

Transaction Type: Purchase

Filter Type: BySerial

Serial Number:

Terminal Pin:

From Date:

To Date:

From Time:

To Time:

Get Reports

2. Transaction Result:

This Result came from EFT-POS.

Amount:

Response Code:

Message:

Card Number Mask: #####-\*\*-####

Card Number Hash:

Terminal ID:

Trace Number:

Serial ID:

RRN:

Transaction Date:

Operations:

Reset Page Exit

Ready.

همانطور که ذکر شد، گزارش های مختلفی قابل دریافت می باشند. TransactionType نوع تراکنش را نشان می دهد:

- خرید (Purchase)
- قبض (Bill)

- شارژ رمزی (Pincharge)
- شارژ مستقیم (Topup)
- مجموع صورتحساب (Sum)
- سرویس پرداخت (Payment Service)

reportFilterType ReportFilterType: مشخص کننده فیلتر گزارشگیری می باشد. انواع تراکنش ها برای دریافت گزارش به شرح زیر می-

باشد:

- بازه تاریخی (فاصله تاریخ شروع و پایان حداکثر می تواند ۵ روز باشد و هر دو باید در یک ماه باشند) ByDate
- سریال تراکنش (وقتی نوع تراکنش مجموع صورتحساب باشد، نباید فیلتر برابر سریال انتخاب شود) bySerial
- بازه ساعتی (انتخاب یک تاریخ و وارد کردن ساعت شروع و پایان) BySerial

reportFromDate string: از تاریخ. فرمت تاریخ به صورت روز/ماه/سال (مانند ۱۳۹۵/۰۷/۰۸) باید وارد شود.

reportToDate string: تا تاریخ. فرمت تاریخ به صورت روز/ماه/سال (مانند ۱۳۹۵/۰۷/۰۸) باید وارد شود.

serialId string: شماره سریال

terminalPin string: رمز ترمینال. در صورتی که مقداری برای این پارامتر وارد نشود روی دستگاه کارخوان بعد از ارسال دستور گزارشگیری،

رمز ترمینال باید وارد شود.

- fromTime string: از ساعت. فرمت ساعت به صورت دقیقه/ساعت (۱۳:۰۱) می باشد.
- toTime string: تا ساعت. فرمت ساعت به صورت دقیقه/ساعت (۱۳:۰۱) می باشد.

با زدن کلید GetReports نتیجه گزارش در TextBox روبرو به صورت Xml نشان داده می شود:

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

tabReports tabCharge tabMCIBill tabTCIBill tabBill Purchase Balance Configurations

1. Getting Repoert from EFT-POS:

Transaction Type: Purchase

Filter Type: ByDate

Serial Number:

Terminal Pin: \*\*\*\*

From Date: 1395/11/24

To Date: 1395/11/26

From Time: \_:\_

To Time: \_:\_

Get Reports

```
<?xml version="1.0" encoding="UTF-8"?>
<Transactions><Transation1><TransationType>Purchase</TransationType><Date>1395/11/26</Date><Time>09:38:12</Time><ShiftNum>0</ShiftNum><TraceNo>000004</TraceNo><Amount>1000</Amount><RRN>005232371736</RRN><Bank>سلمان</Bank><MaskPan>621986-##-8003</MaskPan><HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB=</HashPan><TerminalId>10332839</TerminalId></Transation1><Transation2><TransationType>Purchase</TransationType><Date>1395/11/26</Date><Time>09:38:01</Time><ShiftNum>0</ShiftNum><TraceNo>000003</TraceNo><Amount>1000</Amount><RRN>005232370590</RRN><Bank>سلمان</Bank><MaskPan>621986-##-8003</MaskPan><HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB=</HashPan><TerminalId>10332839</TerminalId></Transation2><Transation3><TransationType>Purchase</TransationType><Date>1395/11/26</Date><Time>09:37:41</Time><ShiftNum>0</ShiftNum><TraceNo>000002</TraceNo><Amount>1000</Amount><RRN>005232368870</RRN><Bank>سلمان</Bank><MaskPan>621986-##-8003</MaskPan><HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB=</HashPan><TerminalId>10332839</TerminalId></Transation3><Transation4><TransationType>Purchase</TransationType><Date>1395/11/26</Date><Time>09:37:31</Time><ShiftNum>0</ShiftNum><TraceNo>000001</TraceNo><Amount>1000</Amount><RRN>005232367899</RRN><Bank>سلمان</Bank><MaskPan>621986-##-8003</MaskPan><HashPan>2DFB05001CAFECD5321715AAA14D2186CCE1E9EB=</HashPan><TerminalId>10332839</TerminalId></Transation4></Transactions>
```

2. Transaction Result:

This Result came from EFT-POS.

Amount:

Response Code: 00

Message: عملیات موفق

Card Number Mask: #####-\*\*-####

Card Number Hash:

Terminal ID:

Trace Number:

Serial ID:

RRN:

Transaction Date:

Operations:

Reset Page Exit

Successful Transaction. Trace #: ""

## ۷.۳ شارژ

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Configurations

1. Starting Charge transaction with EFT-POS:

Mobile Number:

Additional Data:

Reference Data:

Pin Charge:

Charge Serial:

Charge Pin:

Emergency Number:

2. Transaction Result:

This Result came from EFT-POS.

Amount:

Response Code:

Message:

Card Number Mask:

Card Number Hash:

Terminal ID:

Trace Number:

Serial ID:

RRN:

Transaction Date:

Operations:

Ready.

برای خرید شارژ ۲ روش شارژ رمزی و شارژ مستقیم وجود دارد.

با وارد کردن شماره موبایل و زدن کلید BuyTopupCharge شارژ مستقیم خریداری شده و برای شماره مورد نظر ارسال می شود.

برای خرید شارژ رمز دار فقط کافی است که کلید BuyPinCharge فشرده شود.

## ۷,۴ قبض همراه اول

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Configurations

1. Starting Bill transaction with EFT-POS:

Send MCI Request: MCIBill Request

Mobile Number: 09122375008

Bill Type: Periodic

Pay MCI Bill Get MCI Bill

Additional Data

```
<?xml version="1.0" encoding="utf-16"?>
<List>
<print>
<item>123</item>
<value>a</value>
<alignment>0</alignment>
</print>
</receipttype>0</receipttype>
</print>
<item>456</item>
<value>c</value>
<alignment>1</alignment>
</receipttype>1</receipttype>
</print>
</List>
```

Reference Data:

2. Transaction Result:

This Result came from EFT-POS.

Amount: Terminal ID:

Response Code: Trace Number:

Message: Serial ID:

Card Number Mask: \*\*\*\*\*-\*\*-\*\*\*\* RRN:

Card Number Hash: Transaction Date:

Operations:

Reset Page Exit

Ready.

در رابطه با قبض همراه اول، بعد از وارد نمودن شماره موبایل، ۲ عملیات استعلام قبض (Get MCI Bill) و پرداخت قبض (Pay MCI Bill) به ۲ صورت پایان دوره (FullTime) و میان دوره (Periodic) وجود دارد.

در صورتی که می‌خواهید شناسه قبض و شناسه پرداخت در دستگاه کارخوان وارد شود کافی است بدون وارد کردن هیچ گونه داده‌ای کلید MCI Bill Request را بزنید.

## ۷,۵ قبض تلفن استان

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Configurations

1. Starting Bill transaction with EFT-POS:

Send TCI Request:

TCI Number:

Bill Type:

Additional Data

```
<?xml version="1.0" encoding="utf-16"?>
<List>
<print>
<item>123</item>
<value>a</value>
<alignment>0</alignment>
<receipttype>0</receipttype>
</print>
<print>
<item>456</item>
<value>c</value>
<alignment>1</alignment>
<receipttype>1</receipttype>
</print>
</List>
```

Reference Data:

2. Transaction Result:

This Result came from EFT-POS.

Amount:  Terminal ID:

Response Code:  Trace Number:

Message:  Serial ID:

Card Number Mask:  RRN:

Card Number Hash:  Transaction Date:

Operations:

Ready.

در رابطه با قبض تلفن استان، بعد از وارد نمودن شماره کد استان بدون صفر + شماره تلفن، ۲ عملیات استعلام قبض (Get TCI Bill) و پرداخت قبض (Pay TCI Bill) به ۲ صورت پایان دوره (FullTime) و میان دوره (Periodic) وجود دارد.

در صورتی که می‌خواهید شماره تلفن و نوع پرداخت در دستگاه کارخوان وارد شود کافی است بدون وارد کردن هیچ گونه داده‌ای کلید TCI Bill Request را بزنید.



۷,۶ سایر قبوض

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Configurations

1. Starting Bill transaction with EFT-POS:

Send Bill Request: Bill Payment Request

Bill ID:

Pay ID:

Generate Bill Get Bill Properties

Bill Type:

Bill Amount:

Pay Bill

Additional Data

```
<?xml version="1.0" encoding="utf-16"?>
<List>
<print>
<item>123</item>
<value>a</value>
<alignment>0</alignment>
</print>
<receipttype>0</receipttype>
</print>
<print>
<item>456</item>
<value>c</value>
<alignment>1</alignment>
</print>
<receipttype>1</receipttype>
</print>
</List>
```

Reference Data:

2. Transaction Result:

This Result came from EFT-POS.

Amount:

Response Code:

Message:

Card Number Mask: #####-\*\*-####

Card Number Hash:

Terminal ID:

Trace Number:

Serial ID:

RRN:

Transaction Date:

Operations:

Reset Page Exit

Ready.

با وارد کردن شماره قبض و شماره پرداخت با استفاده از `GetBillProperties` می‌توان نوع و مبلغ قبض را به دست آورد.

برای تست با استفاده از `GenerateBill` شماره قبض و شماره پرداخت تستی تولید می‌شود.

بعد از وارد کردن شماره قبض و شماره پرداخت می‌توان با استفاده از `PayBill` قبض را پرداخت نمود.

در صورتی که می‌خواهید شماره تلفن و نوع پرداخت در دستگاه کارخوان وارد شود کافی است بدون وارد کردن هیچ گونه داده‌ای کلید `Bill Payment Request` را بزنید.

۷,۷ خرید

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Configurations

Start waiting:

PAN:

Hash PAN:

Terminal ID:

Purchase Types

1. Starting payment transaction with EFT-POS:

1. Send Amount

Single Account ☒ Share Account ☐

Amount:

Additional Data

```
<?xml version="1.0" encoding="utf-16"?>
<List><print><item>123</item>
<value>a</value><alignment>
0</alignment><receipttype>0</r
eceipttype></print><print><ite
m>456</item><value>c</value>
<alignment>1</alignment><rec
eipttype>1</receipttype></print>
</List>
```

Reference Data:

2. Transaction Result:

This Result came from EFT-POS.

Amount:  Terminal ID:

Response Code:  Trace Number:

Message:  Serial ID:

Card Number Mask:  RRN:

Card Number Hash:  Transaction Date:

Operations:

Ready.

خرید به ۲ صورت قابل انجام است:

- خرید از سمت رایانه:

کلید **SendAmount Pc Starter** به این منظور وجود دارد. بعد از وارد کردن مبلغ و داده اضافی برای چاپ روی رسید، با فشردن این کلید، مبلغ روی دستگاه کارتخوان نمایش داده می‌شود و از مشتری تقاضا می‌شود کارت کشیده شود. پاسخ انجام تراکنش به رایانه بازگردانده می‌شود.

○ خرید شناسه دار:

نوعی از خرید است که همراه مبلغ ارسالی به پوز می‌بایست یک شناسه نیز ارسال گردد. این شناسه توسط سوئیچ اعتبارسنجی می‌شود.

• خرید از سمت دستگاه کارتخوان:

با زدن کلید **Start**، عبارت "لطفاً کارت بکشید" روی کارتخوان نمایش داده می‌شود. بعد از کشیدن کارت اطلاعات کارت (به ۲ صورت ماسک شده و هش)، شماره ترمینال و انواع خریدهای تعریف شده روی کارت به رایانه ارسال می‌شود. کاربر بعد از انتخاب نوع خرید (در کارت های سگمنتی) و تعیین مبلغ با استفاده از کلید **SendAmount** نوع خرید و مبلغ را به دستگاه کارتخوان ارسال می‌نماید. این تنها تراکنشی است که نیاز به فراخوانی ۲ متد برای تکمیل تراکنش وجود دارد. برای ایجاد امنیت بیشتر در زمان تراکنش، بهتر است در متد **PcStarterPurchase** و **PosStarterPurchase** شماره ترمینال مورد استفاده را برای پوز ارسال کرد. در صورتیکه این مقدار با شماره ترمینال دستگاه کارتخوان تفاوت داشته باشد، در هنگام خرید یک مرحله ای و دو مرحله‌ای خطای امنیتی صادر می‌شود.

۷,۸ موجودی

PC to POS (Version: 1.0.0.0)

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.Pc2Pos dll.

PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Configurations

1. Starting Balance transaction with EFT-POS:

Starting waiting:

2. Transaction Result:

This Result came from EFT-POS.

Amount:	<input type="text"/>	Terminal ID:	<input type="text"/>
Response Code:	<input type="text"/>	Trace Number:	<input type="text"/>
Message:	<input type="text"/>	Serial ID:	<input type="text"/>
Card Number Mask:	#####-**-####	RRN:	<input type="text"/>
Card Number Hash:	<input type="text"/>	Transaction Date:	<input type="text"/>

Operations:

Ready.

با فشردن کلید Start می‌توان دستور موجودی را به کارخوان ارسال نمود.

## ۷,۹ سرویس پرداخت

PC to POS (Version: 1.3.3.0) - □ ×

Copyright © 2016 Saman Electronic Payments. All rights reserved.  
This is a Demo application for SSP1126.PcPos dll. PCPOS Protocol SSP1126

Reports Charge MCIBill TCIBill Bill Purchase Balance Payment Service Configurations

Start Service Payment:

Reference Data:

Amounts:

مبلغ گروه مواد پروتئینی	<input type="text" value="0"/>
مبلغ گروه هیدروکربن	<input type="text" value="0"/>
مبلغ گروه روغن	<input type="text" value="0"/>
مبلغ گروه لبنیات	<input type="text" value="0"/>
مبلغ گروه سایر	<input type="text" value="0"/>

2. Transaction Result:

This Result came from EFT-POS.

Main Amount:	<input type="text"/>	Terminal ID:	<input type="text"/>
Affective Amount:	<input type="text"/>	Trace Number:	<input type="text"/>
Response Code:	<input type="text"/>	Serial ID:	<input type="text"/>
Message:	<input type="text"/>	RRN:	<input type="text"/>
Card Number Mask:	#####-**-####	Transaction Date:	<input type="text"/>
Card Number Hash:	<input type="text"/>		

Operations:

Card swiped with "621986-##-8011" card number :

روش انجام این تراکنش به اینصورت است که با زدن دکمه Start دستگاه کارتخوان پیغام "لطفاً کارت را بکشید" را نشان می‌دهد. با کشیدن کارت روی دستگاه کارتخوان، منوی سرویس پرداخت نشان داده می‌شود. انتخاب یک سرویس پرداخت و انجام مراحل آن تا قبل از مرحله وارد کردن مبلغ، روی دستگاه کارتخوان انجام می‌شود. اما دریافت مبلغ از سمت رایانه صورت می‌گیرد. و با زدن دکمه Send Amounts مبلغ به دستگاه کارتخوان ارسال می‌شوند. در تراکنش سرویس پرداخت امکان ارسال داده اضافی به سمت پوز وجود ندارد.

## ۸. ابزار رمزنگاری (Hash Helper)

به منظور سهولت کاربر در رمزنگاری و مقایسه رشته های رمز شده برگه جدیدی در نسخه ۱,۴,۰,۱ اضافه شده است. این برگه شامل دو بخش است که در ادامه توضیح داده می شود.

### ۱- Get Hash

برای رمزنگاری اطلاعات می توان در قسمت Input Text متن دلخواه را وارد کرد، سپس با کلیک کردن روی دکمه Hash متن رمز شده در قسمت Output Text تولید می شود.

### ۲- Compare

در بخش امکان مقایسه کردن دو متن رمز شده فراهم شده است. هریک از متن ها را در قسمت Input Text 1 و Input Text 2 وارد کنید، پس از آن دکمه Compare را کلیک نمایید.

## ۹. ایجاد تراکنش

جهت استفاده کافی است مراحل ذیل را اجرا نمایید: ( مثال ها با استفاده از زبان C# نوشته شده است.)

- ایجاد یک نمونه از کلاس **PcPosFactory**

```
if (_PcPosFactory == null)
    _PcPosFactory = new PcPosFactory();
```

- Assign نمودن Event ها

```
_PcPosFactory.CardSwiped += _PosClient_CardSwiped;
_PcPosFactory.PosResultReceived += _PosClient_PosResultReceived;
```

- Initialize نمودن خصوصیات شی

```
private bool TransactionMediaInitialization()
{
    if (_mediaType == MediaType.Com)
    {
        SerialPort selectedPort = null;

        if (SerialPort.GetPortNames().Any(p => p == cmbPort.SelectedItem.ToString()))
            selectedPort = new SerialPort((string)cmbPort.SelectedItem);
        if (selectedPort == null)
        {
            MessageBox.Show("There is no selected Port in configurations.", "Error",
                MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            return true;
        }
        _PcPosFactory.SetCom(selectedPort.PortName, Int32.Parse(txtBaudRate.Text));
    }
    if (_mediaType == MediaType.Network)
    {
        int port;
        if (string.IsNullOrEmpty(txtPosIP.Text))
        {
            MessageBox.Show("There is no value for Pos IP in configurations.", "Error",
                MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            return true;
        }
        _PcPosFactory.SetLan(txtPosIP.Text);
    }
    _PcPosFactory.Initialization(_responseLanguage, 0, _asyncType);

    txtStatus.Text = "Waiting for EFT-POS...";
    return false;
}
```

- فراخوانی متد مناسب جهت انجام تراکنش  
به عنوان مثال موجودی:

```
private void btnStartBalance_Click(object sender, EventArgs e)
{
    progressBar.Visible = true;
    pnlStartBalance.Enabled = false;

    ResetResult();
    if (TransactionMediaInitialization()) return;

    PosResult posResult = _PcPosFactory.Balance();
    //it means that _transactionMode is sync
    if (posResult != null && _asyncType == AsyncType.Sync)
    {
        BalanceResultReceived(posResult);
    }
}
```

در صورتی که تراکنش به صورت Async باشد، posResult خروجی متد نال است و رویدادهای CardSwiped بعد از کشیدن کارت و PosResultReceived بعد از نمایش موجودی به کاربر فراخوانی می‌شود.  
برای سایر تراکنش‌ها نیز به همین صورت عمل می‌شود.



## ۱۰. راهنما

Enum های استفاده شده در برنامه

○ پاسخ زبان نتیجه تراکنش

```
public enum ResponseLanguage
{
    Persian = 0,
    English = 1
}
```

○ نوع ارتباط با رایانه

```
public enum MediaType
{
    Com = 1,
    Network = 2,
}
```

○ نوع تسهیم حساب

```
public enum AccountType
{
    Single = 0,
    Share = 1
}
```

○ نوع Alignment متن چاپ شده روی پرینت کارتخوان

```
public enum Alignment
{
    Right = 0,
    Left = 1,
    Center = 2
}
```

○ نوع رسید متن چاپ شده روی پرینت کارتخوان

```
public enum ReceiptType
{
    Customer = 0,
    Merchant = 1,
    Both = 2
}
```

○ نوع همروندی تراکنشها

```
public enum AsyncType
{
    Sync = 0,
    Async = 1,
}
```

```
}

```

○ نوع زمان قبض

```
public enum PhoneBillType
{
    Periodic = 1,
    FullTime = 2,
}
```

○ نوع تراکنش انتخابی برای گزارشات

```
public enum ReportAction
{
    Purchase = 1,
    Bill = 2,
    PinCharge = 3,
    TopupCharge = 4,
    Sum = 5, // صورت حساب مجموع
}
```

○ نوع فیلتر گزارشات

```
public enum ReportFilterType
{
    BySerial = 1,
    ByDate = 2,
    ByTime = 3,
}
```

○ نوع تراکنش

```
public enum TransactionType
{
    None = 0,
    Balance = 1,
    Purchase = 2,
    BillPay = 3,
    BillRequest = 4,
    MCIBillInquiry = 5,
    MCIBillPay = 6,
    MCIBillRequest = 7,
    PinCharge = 8,
    TopupCharge = 9,
    GetReport = 10,
    GetAuthorizedOperations = 11,
    EchoTest = 12,
    TCIBillInquiry = 13,
    TCIBillPay = 14,
    TCIBillRequest = 15,
}
```

○ فرمت چاپ اطلاعات روی پرینت

از یک String که حاوی Xml شامل نودهای item, value, alignment, receipttype است، استفاده می شود. در زیر یک مثال ارائه شده است:

```
<?xml version="1.0" encoding="utf-16"?>
<List>
  <print>
    <item>شماره پرداخت</item>
    <value>۱۲۳۴۵۶</value>
    <alignment>۰</alignment>
    <receipttype>1</receipttype>
  </print>
  <print>
    <item>شماره قبض</item>
    <value>۱۱۰۱۰۰</value>
    <alignment>۰</alignment>
    <receipttype>1</receipttype>
  </print>
</List>
```

در این مثال Item نشان دهنده عبارتی مانند شماره قبض و Value نشان دهنده عبارتی مانند ۵۱۴۴۲۴۱۸۰۶۲۳۵ می باشد. Alignment نشان دهنده نوع قرارگیری متن در هنگام چاپ رسید می باشد.

برای تعیین جهت متن از یکی از موارد زیر به عنوان مقدار می توانید استفاده نمایید:

- Right و یا ۰
- Left و یا ۱
- Center و یا ۲

ReceiptType نشان دهنده نوع رسیدی است که اطلاعات روی آن چاپ می شود:

- Customer و یا ۰
- Merchant و یا ۱
- Both و یا ۲

وقتی Item مقدار داشته باشد و Value خالی باشد می توانید از نوع Center استفاده نمایید.

در صورتی که نیاز به چاپ داده اضافی روی پوز ندارید این ورودی در متد باید برابر با NULL قرار داده شود.



## ۱۱. مقادیر کدهای پاسخ (Response Code)

مقادیر کدهای پاسخ معادل پیام‌های زیر هستند:

کد	پیام
۰	تراکنش با موفقیت انجام پذیرفت
۱	کارت کشیده شد
۲	مبلغ تراکنش نمی تواند از "حداقل مبلغ" کوچکتر باشد
۳	عدم ارتباط با دستگاه
۴	اطلاعات نامعتبر
۵	صفر ریال بدهی
۶	خطا در دریافت اطلاعات
۷	عدم دسترسی به این عملیات
۸	تراکنش یافت نشد
۹	ترمینال نامعتبر
۱۰	خطا در پاسخ
۱۲	تراکنش نامعتبر
۷۹ / ۱۳	مبلغ نامعتبر
۱۴	خطا در مقداردهی
۲۰	پاسخ نامعتبر
۲۶	خطا در تراکنش
۲۷	این قبض قبلا پرداخت شده است
۲۸	غیرقابل پرداخت

خطا در قالب اطلاعات	۳۰
تاریخ انقضای کارت سپری شده است	۳۳
اخطار امنیتی	۴۳ / ۶۳ / ۳۴
تعداد دفعات ورود رمز غلط بیش از حد مجاز است	۳۸
موجودی کافی نمی باشد	۵۱
رمز کارت نا معتبر است	۵۵
انجام تراکنش مربوطه توسط دارنده ی کارت مجاز نمی باشد	۵۷
انجام تراکنش مربوطه توسط پایانه ی انجام دهنده مجاز نمی باشد	۵۸
مبلغ تراکنش بیش از حد مجاز می باشد	۶۱
عدم دریافت پاسخ در زمان مناسب	۶۸
استفاده از ترمینال نا معتبر	۷۱
عدم دریافت پاسخ در زمان مناسب از دستگاه کارتخوان	۷۲
تعداد دفعات ورود رمز غلط بیش از حد مجاز است	۷۵ / ۶۹
کارت غیرفعال می باشد	۷۸
عدم پاسخ از سوی صادرکننده ی کارت	۹۱ / ۸۴ / ۸۰
مبالغ متفاوت	۹۲
خطای نامشخص	۹۶
عدم ارتباط با مرکز	۹۷
لغو عملیات توسط کاربر	۹۸
عدم دریافت پاسخ در زمان مناسب در کارتخوان	۹۹