# MOLWallet SDK Specifications for Windows POS
## (Version 1.0.0)
### Last updated March 6, 2015

## **MOLPay Sdn Bhd** (948015-X)

B-13-3A, Jalan Multimedia 7/AH,
CityPark, i-City, 40000 Shah Alam,
Selangor Darul Ehsan, Malaysia.

☎ +603-55218438
🖨 +603-55218437
✉ sales@molpay.com
Ⓜ www.molpay.com

## Table of Contents

# 1. Prerequisite

Please provide below information to verify merchant identity from MOLPay Server in order for merchant's POS system to use the MOLWallet payment channel:

- merchant ID
- verify key
- appname
- username
- password

All the values above are in one word without whitespace.

The information should be submitted to MOLPay support team and the email address is **support@molpay.com**

# 2. System Requirements

- Minimum required NET.Framework 4.5
- Internet Connection is essential for the MOLWallet SDK to successfully connected with the Server

# 3. POS Pay  MOLWallet Payment Library

This section explains details about the library API, and it explains the instructions and example to be follow to develop a POS that using MOLWallet payment channel.
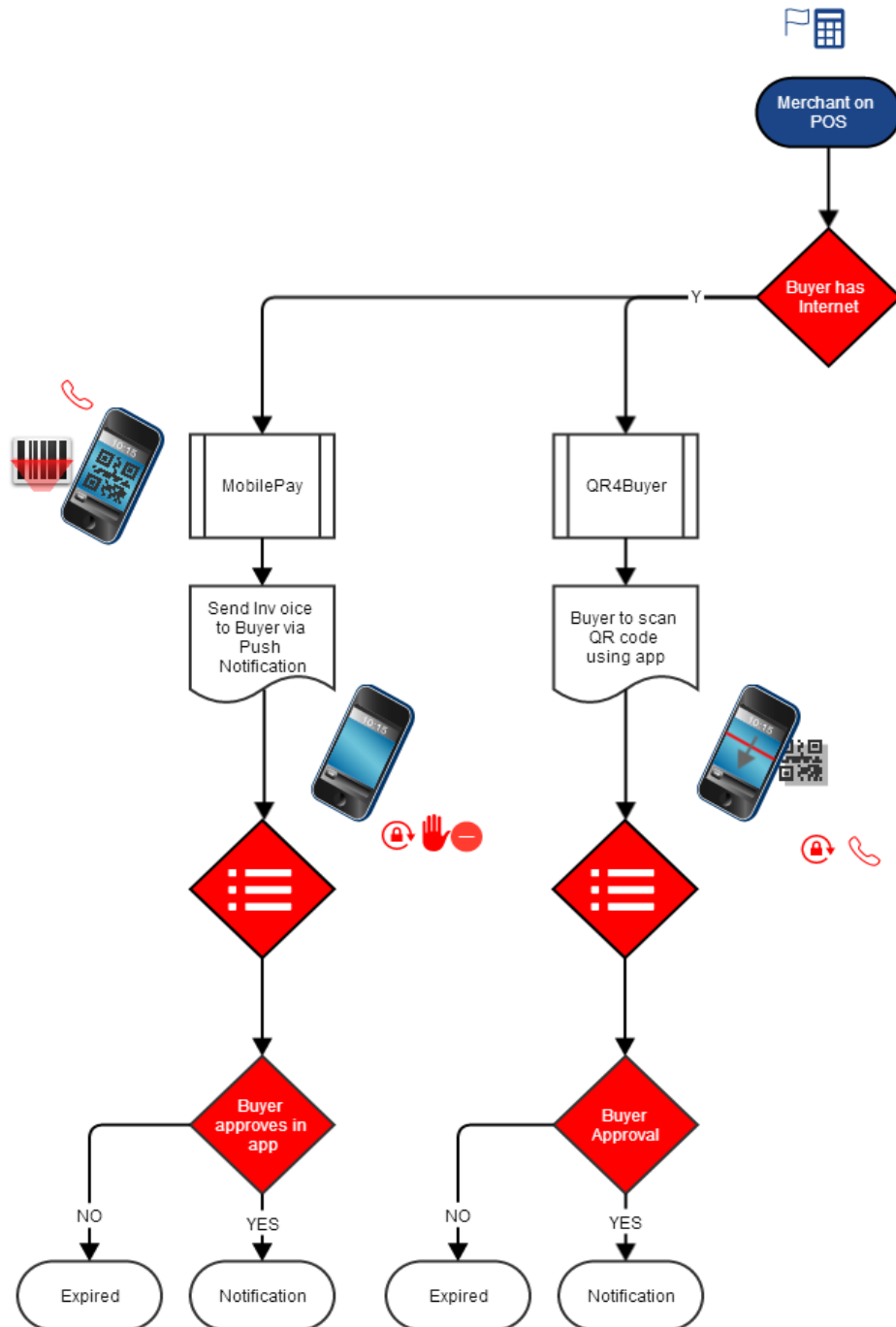
## 3.1  How the Library Works

The flow of the library is as below:
1. Library is in DLL format;
2. Initiate the library;
3. Merchant app to pass buyer's information to the SDK library and will be sent to MOLWallet server;
4. After payment is completed, MOLWallet will return the result for transaction id and payment status;
5. Library will return the result back to application.

## 3.2 Initiate the Payment

There are two types of flow which explained in provided scheme below.

## 3.2.1 Initiate the Payment to generate QR code

```csharp
private async void InitQRPayment()
    {
        try
        {
            Dictionary<MOLWallet_SDK.Keys, string> extraDic = new
                            Dictionary<MOLWallet_SDK.Keys, string>();
            extraDic.Add(MOLWallet_SDK.Keys.MerchantId, "MerchantId");
            extraDic.Add(MOLWallet_SDK.Keys.AppName, "AppName");
            extraDic.Add(MOLWallet_SDK.Keys.TerminalId, "TerminalId");
            extraDic.Add(MOLWallet_SDK.Keys.VerifyKey,
                            "verifyKey");
            extraDic.Add(MOLWallet_SDK.Keys.BillName, "Wallet user");
            extraDic.Add(MOLWallet_SDK.Keys.BillDescription,
                                            "description");
            extraDic.Add(MOLWallet_SDK.Keys.Currency, "MYR");
            extraDic.Add(MOLWallet_SDK.Keys.Country, "MY");
            extraDic.Add(MOLWallet_SDK.Keys.Amount, "amount");
            extraDic.Add(MOLWallet_SDK.Keys.UserName, "username");
            extraDic.Add(MOLWallet_SDK.Keys.Password, "password");

            molWallet = new MOLWallet();
            string result = await molWallet.GetPayment(extraDic,
                                            PaymentMode.QR);
            JObject jResult = JObject.Parse(result);
            var status = jResult["status"].ToString();
            if (status.Equals("True"))
            {
                txnId = jResult["txn_ID"].ToString();
                string url = jResult["url"].ToString();
                webBrowser1.DocumentText =
                    "<html><body>" +
                    "<img src='" +url +"' height='400' width='400'>" +
                    "</body></html>";
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.ToString());
        }

    }
```

## 3.2.2 Initiate the Mobile Payment when scan customer's phone

```csharp
public async void InitMobilePayment(string phoneNumber)
    {
        try
        {

            Dictionary<MOLWallet_SDK.Keys, string> extraDic = new
                        Dictionary<MOLWallet_SDK.Keys, string>();
            extraDic.Add(MOLWallet_SDK.Keys.MerchantId, "MerchantId");
            extraDic.Add(MOLWallet_SDK.Keys.AppName, "AppName");
            extraDic.Add(MOLWallet_SDK.Keys.TerminalId, "TerminalId");
            extraDic.Add(MOLWallet_SDK.Keys.VerifyKey,
                            "verifyKey");
            extraDic.Add(MOLWallet_SDK.Keys.BillName, "Wallet user");
            extraDic.Add(MOLWallet_SDK.Keys.BillMobile, phoneNumber);
            extraDic.Add(MOLWallet_SDK.Keys.BillDescription,
                                            "description");
            extraDic.Add(MOLWallet_SDK.Keys.Currency, "MYR");
            extraDic.Add(MOLWallet_SDK.Keys.Country, "MY");
            extraDic.Add(MOLWallet_SDK.Keys.Amount, "amount");
            extraDic.Add(MOLWallet_SDK.Keys.UserName, "username");
            extraDic.Add(MOLWallet_SDK.Keys.Password, "password");

            molWallet = new MOLWallet();
            string result = await molWallet.GetPayment(extraDic,
                                            PaymentMode.Mobile);
            JObject jResult = JObject.Parse(result);

            var status = jResult["status"].ToString();
            if (status.Equals("true"))
            {
                TxnId = jResult["order_id"].ToString();
            }

            this.Invoke((MethodInvoker)delegate
            {
                lbStatus.Text = result;
            });
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.ToString());
        }

    }
```

5

## 3.3 Payment Result (Payment Status)

You have to do manual request for the payment result.
There are four types of result as listed below:

```
//transactionId is dynamic. You will get it when you call to GetPayment
method //previously.
 Dictionary<MOLWallet_SDK.Keys, string> extraDic = new
                                Dictionary<MOLWallet_SDK.Keys, string>();
            extraDic.Add(MOLWallet_SDK.Keys.MerchantId, "MerchantId");
            extraDic.Add(Keys.TransactionId, "TransactionId");
            extraDic.Add(MOLWallet_SDK.Keys.VerifyKey,
                              "verifyKey");
            extraDic.Add(MOLWallet_SDK.Keys.Amount, "amount");
string result = molWallet.CheckPaymentStatus(extraDic);
```

The result will be in json format. Moreover there will be "status_code"
field which represents payment's current status.

Codes for status:

00 for Successful payment;

11 for failure;

22 for pending.

6

![MOLPay logo - Buy Everywhere, Pay Anywhere]

## 3.4 MOLWallet POS SDK Screenshots

## **<u>Payment page</u>**

## Payment methods page

**Scanning buyer's mobile for payment**

**Buyer scans POS QR code for payment**