



iPay88 iOS Mobile Payment SDK v1.0.6.1

(Malaysia Gateway)

Contents

Contents	1
1 Mobile Payment SDK – iOS	2
1.1 Pre-requisites when develop iOS Apps that use iPay88 iOS Mobile SDK channel.....	2
2 iPay88 IOS Mobile Payment Library.....	2
2.1 Mobile Payments Library API Reference	2
2.2 Adding the Library and Importing Header Files	2
2.3 Start the Library UIView	3
2.4 Method in the Mobile Payments Library	4
2.5 Custom Objects in the Mobile Payments Library.....	5
2.6 Payment Response Parameters return to BackendPostURL.....	6
2.6.1 ASP sample code	8
2.6.2 PHP sample code.....	8
2.7 BackendPostURL Response page signature	9
2.7.1 Sample function code to generate iPay88 Signature	10
3 The Checkout Experience with the iPay88 IOS Mobile Payments Library	11
3.1 iPay88 IOS Mobile – Credit Card	11
3.2 iPay88 iOS Mobile – CIMB	13
3.3 iPay88 iOS Mobile – Maybank2u	16
4 Appendix I	20
4.1 PaymentId.....	20
4.2 Currency	21
4.3 Error Description.....	21

1 Mobile Payment SDK – iOS

1.1 Pre-requisites when develop iOS Apps that use iPay88 iOS Mobile SDK channel

Please provide your Merchant Code to iPay88 to verify merchant identity.

iPay88 need to configure your Merchant Code into iPay88 server to allow you can use iPay88 Mobile Payment channel.

2 iPay88 IOS Mobile Payment Library

This section provides details about the Mobile Payments Library API, and it provides instructions and examples for integrating the library with merchant iOS application.

2.1 Mobile Payments Library API Reference

The flow of the library is as follow:

1. Your application initializes the library.
2. Your UI view controller should add the UI view returned by the library as its subview.
3. After buyers complete their payments, the library returns a callback to your application with the status of the payment and the Transaction Id.

2.2 Adding the Library and Importing Header Files

1. In XCode, open your build target and go to “Build Phases” tab.
2. Under “Link Binary With Libraries”, add the “libipay88sdk.a” static library.
3. Add the headers files (“Ipay.h” & “IpayPayment.h”) into your project.

Also, import the appropriate headers into your application classes. The following classes must be imported:

```
#import "Ipay.h"  
#import "IpayPayment.h"
```

2.3 Start the Library UIView

The Library uses the native `UIWebView` to start the checkout flow, and to communicate completion back to you. You can implement `PaymentResultDelegate` to be informed immediately upon successful completion of a payment.

To start the `Ipay88` payment, you must initialize the `Ipay` object first.

You must use the checkout method and give it the `IpayPayment` object. `IpayPayment` handles simple payments, which support single receivers of payments with one transaction and a few details.

In the following example, the buyer checks out with a simple payment for a single recipient:

```
paymentsdk = [[Ipay alloc] init];
paymentsdk.delegate = self;
IpayPayment *payment = [[IpayPayment alloc] init];
[payment setId:@"6"];
[payment setMerchantKey:@"apple88KEY"];
[payment setMerchantCode:@"M09999"];
[payment setRefNo:@"ORD1188"];
[payment setAmount:@"1.00"];
[payment setCurrency:@"MYR"];
[payment setProdDesc:@"Payment for ORD1188"];
[payment setUsername:@"John Woo"];
[payment setUserEmail:@"johnwoo@yahoo.com"];
[payment setUserContact:@"0123456789"];
[payment setRemark:@"ORD1188"];
[payment setLang:@"ISO-8859-1"];
[payment setCountry:@"MY"];
[payment setBackendPostURL:@"http://www.mywebsite.com/backend_response.php"];
UIView *paymentView = [paymentsdk checkout:payment];
[self.view addSubview:paymentView];
```

2.4 Method in the Mobile Payments Library

The “checkout” method

This method returns a UIView(an instance of UIView).

```
- (UIView *)checkout:(IpayPayment *)payment;
```

The interface for providing details on when a payment is completed is defined in <PaymentResultDelegate> protocol. This interface provides you with a way to be notified immediately when a payment has completed:

```
@protocol PaymentResultDelegate <NSObject>
```

```
- (void)paymentSuccess:(NSString *)refNo withTransId:(NSString *)transId  
withAmount:(NSString *)amount withRemark:(NSString *)remark withAuthCode:(NSString  
*)authCode;
```

```
- (void)paymentFailed:(NSString *)refNo withTransId:(NSString *)transId  
withAmount:(NSString *)amount withRemark:(NSString *)remark withErrDesc:(NSString  
*)errDesc;
```

```
- (void)paymentCancelled:(NSString *)refNo withTransId:(NSString *)transId  
withAmount:(NSString *)amount withRemark:(NSString *)remark withErrDesc:(NSString  
*)errDesc;
```

```
@end
```

Requery method

```
- (void)requery:(IpayPayment *)payment
```

Note: only the refNo, merchantCode and amount properties of the IpayPayment object are required for requery(the rest can be left empty).

The interface for providing details on when a requery is completed is defined in <PaymentResultDelegate> protocol. This interface provides you with a way to be notified immediately when a requery has completed:

```
@protocol PaymentResultDelegate <NSObject>
```

```
- (void)requerySuccess:(NSString *)refNo withMerchantCode:(NSString *)merchantCode  
withAmount:(NSString *)amount withResult:(NSString *)result;
```

```
- (void)requeryFailed:(NSString *)refNo withMerchantCode:(NSString *)merchantCode  
withAmount:(NSString *)amount withErrDesc:(NSString *)errDesc;
```

```
@end
```

2.5 Custom Objects in the Mobile Payments Library

The Mobile Payments Library includes custom objects for passing information between the library and your application during checkout.

ipayPayment

This object is passed to the library. This object contains all the values for a payment.

Method	Description
setMerchantKey(String merchantKey)	[Required] – Merchant Key that provided by iPay88. E.g. apple88KEY
setMerchantCode(String merchantCode)	[Required] – Merchant Code that provided by iPay88. E.g. M09999
setPaymentId(String paymentId)	[Required] – PaymentId is the value to request payment method to appear on iPay88 landing page. Please refer to Appendix I for available PaymentId OR leave it blank by set it as [payment setPaymentId:␣"]; Note: Ensure the Merchant Account had subscribed to the payment option in iPay88 before specify it in the APP
setRefNo(String refNo)	[Required] – Reference number for merchant reference purposes, should be unique for each transaction.
setAmount(String amount)	[Required] – Final Amount to pay and in 2 decimal point, E.g. 10.90 *For testing, please use 1.00
setCurrency(String currency)	[Required] – Currency code that based on standard ISO. E.g. MYR
setProdDesc(String prodDesc)	[Required] – Simple Product Description. E.g. Ticket for Concert. Note: Special characters is not allowed.
setUserName(String userName)	[Required] – Customer name in merchant's system. E.g. John Woo
setUserEmail(String userEmail)	[Required] – Customer email address in merchant's system with valid email format. E.g. johnwoo@yahoo.com
setUserContact(String userContact)	[Required] – Customer contact number in merchant's system. E.g. 60123456789
setRemark(String remark)	[Optional] – Remark for particular transaction. Note: Special characters is not allowed.
setLang(String lang)	[Optional] – Language. *Default - ISO-8859-1
setCountry(String country)	[Required] – Set this field to connect iPay88 gateway to selected country. *Default – MY *Available values: MY / PH / ID Note: Please ensure merchant have merchant account before connect to specified country gateway
setBackendPostURL(String url)	[Required] – Specify a valid merchant callback URL when payment success. E.g. http://www.myshop.com/backend_page.php Note: Refer to section "2.6 Payment Response Parameters return to BackendPostURL" for detail of parameters return from iPay88

2.6 Payment Response Parameters return to BackendPostURL

HTTPS POST response from iPay88 to merchant BackendPostURL when payment success.

Parameter Name	Data Type	Size	M/O	Description
MerchantCode	String	20	M	Merchant Code provided by iPay88 and use to uniquely identify the Merchant. E.g. M09999
PaymentId	Integer		M	Please refer to Appendix I for possible PaymentId value return to BackPostURL.
RefNo	String	30	M	Reference number for merchant reference purposes, should be unique for each transaction.
Amount	Currency		M	Payment amount with two decimals and thousand symbols. E.g. 1,278.90
Currency	String	5	M	Currency code that based on standard ISO. E.g. MYR
Remark	String	100	0	Remark for particular transaction.
TransId	String	30	0	iPay88 Transaction ID. E.g. T019988877700
AuthCode	String	20	0	Bank reference number. Note: Sometime bank may not return reference number to gateway
Status	String	1	M	Use to indicate payment status "1" – Success "0" – Fail
ErrDesc	String	100	0	Payment status description (Refer to Appendix I)
Signature	String	100	M	SHA1 signature (Refer to "2.7 BackendPostURL Response page signature" on how the Signature generated)
CCName	String	200	0	Applicable for credit card payment only. Credit card holder name
CCNo	String	16	0	Applicable for credit card payment only. Masked credit card number. First six and last four of credit card number. Eg: 492159xxxxx4941
S_bankname	String	100	0	Applicable for credit card payment only. Credit card issuing bank name
S_country	String	100	0	Applicable for credit card payment only. Credit card issuing country

** Refer to next page for detail of BackendPostURL implementation...

BackendPostURL (Backend Post Feature)

Backend POST feature is server to server technology where it does not depend on user's web browser to return payment response data to merchant website.

With this feature implemented, merchant server still can get the payment status on the backend (asynchronously) even if merchant mobile application (app) fails to get status via iPay88 SDK which may be due to user accidentally closed web browser, internet connection timeout and etc.

Implementation

On the merchant server, create a page (BackendPostURL) to accept backend post response parameters from iPay88 gateway. The list of parameters return is listed on section "2.6 Payment Response Parameters return to BackendPostURL".

On the BackendPostURL page, write out the word 'RECEIVEOK' only (without quote) as an acknowledgement after update order to PAID on merchant system.

iPay88 will re-try post success status to merchant BackendPostURL up to 3 times on different time interval if no 'RECEIVEOK' acknowledgement from merchant server.

Example:

In ASP code → `response.write "RECEIVEOK"`

In PHP code → `echo "RECEIVEOK";`

Merchant Developer Note on BackendPostURL implementation

- a) iPay88 server will **ONLY** return status if the transaction is payment success. No status will return if the payment is failed.
- b) Ensure no HTML code/tag on merchant BackendPostURL page.
- c) Ensure merchant BackendPostURL contain coding/function to update order besides using *PaymentResultDelegate(paymentSuccess)* method.
- d) Ensure merchant BackendPostURL implement verification check on the received response data to prevent user hijack merchant system, e.g. implement Response Signature comparison check.
- e) Ensure implement **checking to determine** either "*PaymentResultDelegate(paymentSuccess)* method" or "BackendPostURL" when update the order, so merchant system will not update status **more than 1 time**.
Note: When payment success, iPay88 will return success payment status to "*SDK*" and "*BackendPostURL*".
- f) Ensure merchant BackendPostURL should only reply 'RECEIVEOK' when note (b), (c), (d) and (e) implemented properly.
- g) BackendPostURL is not a replacement for *PaymentResultDelegate* method to get payment success status thus merchant still require to implement *PaymentResultDelegate* on mobile application (app) to obtain payment status.

Sample (BackendURL Payment response)

2.6.1 ASP sample code

```

<%
MerchantCode = Request.Form("MerchantCode")
PaymentId    = Request.Form("PaymentId")
RefNo       = Request.Form("RefNo")
Amount      = Request.Form("Amount")
eCurrency   = Request.Form("Currency")
Remark      = Request.Form("Remark")
TransId     = Request.Form("TransId")
AuthCode    = Request.Form("AuthCode")
Status      = Request.Form("Status")
ErrDesc     = Request.Form("ErrDesc")
Signature    = Request.Form("Signature")
%>
<Add your programming code here>

IF Status=1 THEN
    COMPARE Return Signature with Generated Response Signature
    // update order to PAID
    response.write "RECEIVEOK"
ELSE
    // update order to FAIL

```

2.6.2 PHP sample code

```

<?PHP
$merchantcode    = $_REQUEST["MerchantCode"];
$paymentid      = $_REQUEST["PaymentId"];
$refno          = $_REQUEST["RefNo"];
$amount         = $_REQUEST["Amount"];
$currency       = $_REQUEST["Currency"];
$remark        = $_REQUEST["Remark"];
$transid       = $_REQUEST["TransId"];
$authcode      = $_REQUEST["AuthCode"];
$status        = $_REQUEST["Status"];
$errdesc       = $_REQUEST["ErrDesc"];
$signature      = $_REQUEST["Signature"];
PHP?>
<Add your programming code here>

IF ($status=1) {
    COMPARE Return Signature with Generated Response Signature
    // update order to PAID
    echo "RECEIVEOK";
}
ELSE {
    // update order to FAIL
}

```

2.7 BackendPostURL Response page signature

If the Merchant request is successful, the response message will contain as SHA-256 hashed signature. The hash signature for the response is a hash of the following fields:

1. **MerchantKey** (Provided by iPay88 OPSG and share between iPay88 and merchant only)
2. **MerchantCode**
3. **PaymentId**
4. **RefNo**
5. **Amount**
6. **Currency**
7. **Status**

The fields must be set in the following order,

(MerchantKey & MerchantCode & PaymentId & RefNo & Amount & Currency & Status)

For Example:

MerchantKey = "appleKEY"

MerchantCode = "M09999"

PaymentId = "6"

RefNo = "ORD1188"

Amount = "1.00" (**Note:** Remove the "." and "," in the string before hash)

Currency = "MYR"

Status = "1"

The hash would be calculated on the following string:

apple88KEYM099996ORD1188100MYR1

The resulting has signature value equals to (using SHA1 algorithm)

SVd5jkJYHnpDEPVmJPDbGgrG9nl=

To ensure the signature generated was correct, visit the link below for signature comparison.

https://www.mobile88.com/epayment/testing/testsignature_response.asp

2.7.1 Sample function code to generate iPay88 Signature

Using .NET you can use the available libraries to perform this:

```
Public Shared Function ComputeHash(ByVal Key As String) As String
    Dim objSHA1 As New SHA1CryptoServiceProvider()
    objSHA1.ComputeHash(system.Text.Encoding.UTF8.GetBytes(Key.ToCharArray))
    Dim buffer() As Byte = objSHA1.Hash
    Dim HashValue As String = System.Convert.ToBase64String(buffer)

    Return HashValue
End Function
```

Using PHP you can use the following code to perform this:

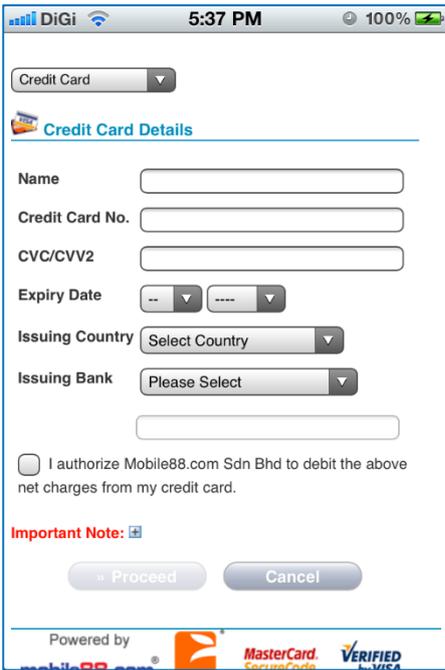
```
<?PHP
function iPay88_signature($source)
{
    return base64_encode(hex2bin(sha1($source)));
}

if (!function_exists('hex2bin'))
{
function hex2bin($hexSource)
{
    for ($i=0;$i<strlen($hexSource);$i=$i+2)
    {
        $bin .= chr(hexdec(substr($hexSource,$i,2)));
    }
    return $bin;
}
}
?>
```

3 The Checkout Experience with the iPay88 IOS Mobile Payments Library

3.1 iPay88 IOS Mobile – Credit Card

Key-in credit card detail such as Card Number, CVC, Expiry Date and Press 'Proceed' button to make payment.



DiGi 5:37 PM 100%

Credit Card

Credit Card Details

Name

Credit Card No.

CVC/CVV2

Expiry Date -- --

Issuing Country

Issuing Bank

I authorize Mobile88.com Sdn Bhd to debit the above net charges from my credit card.

Important Note:

» Proceed Cancel

Powered by   

Key-in OTP (One Time Password) received from Card Issuer SMS notification. Click 'OK' button to authorize the payment.



DiGi 5:40 PM 100%

The MSOS Code has been sent to your mobile number +60xxxx7396. Please enter the MSOS Code to authenticate this payment before session time-out.

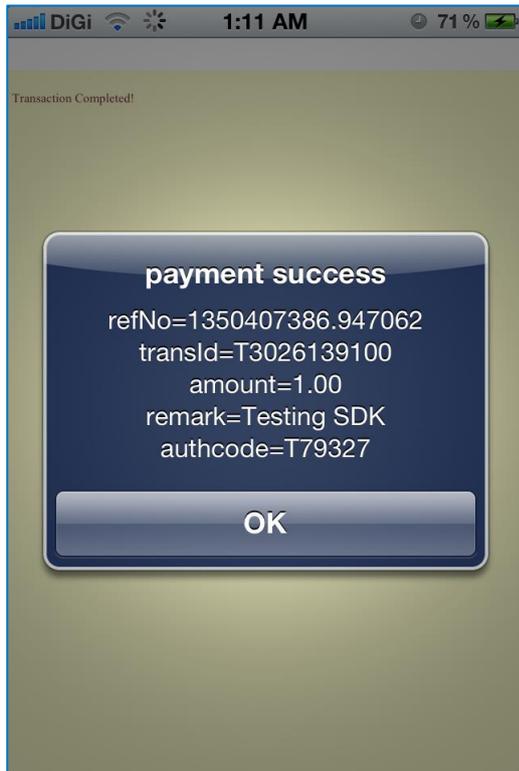
Time remaining: 4 min. 43 sec.

Merchant Name : MOBILE88.COM-IPAY88
Amount : MYR 1.00
Transaction Date : Fri Oct 12 2012 17:39:46 GMT+0800
Maybank MasterCard No. : xxxx xxxx xxxx 7665
MSOS Code :

OK Cancel Resend MSOS Code

This information is **Not Shared** with the Merchant.
 Any problem in receiving your MSOS code? Please contact our **Customer Care Hotline** at **1-300-88-6688** or **603-7844 3696** (overseas) for assistance.

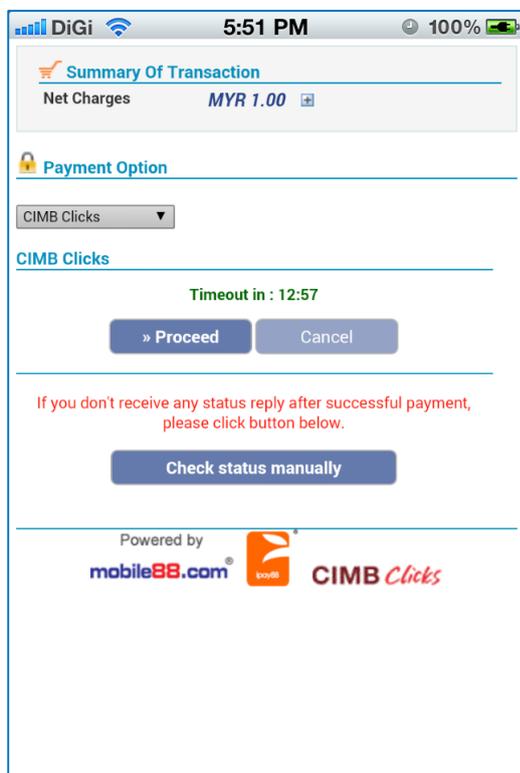
Result will sent to your Result delegate methods.



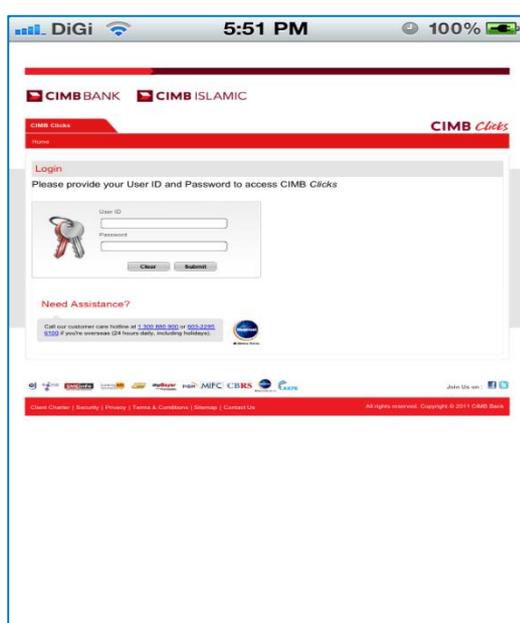
3.2 iPay88 iOS Mobile – CIMB

Customer reached to iPay88 payment page.

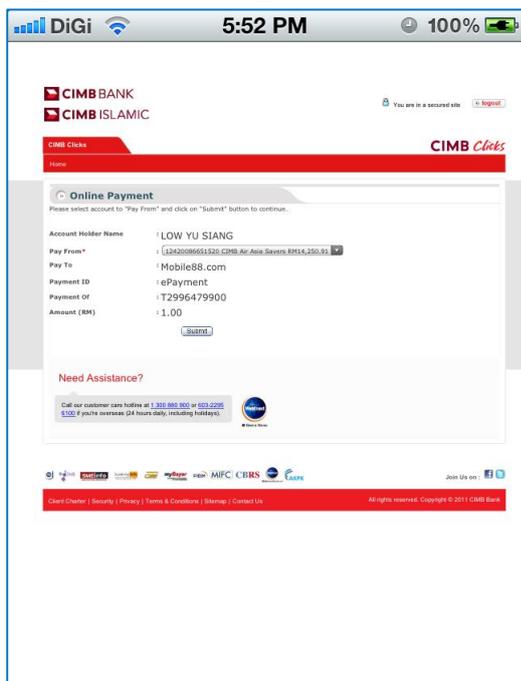
Press 'Proceed' button to Login to CIMBClicks to make payment.



Key-in CIMBClicks User ID and Password to login to CIMB.

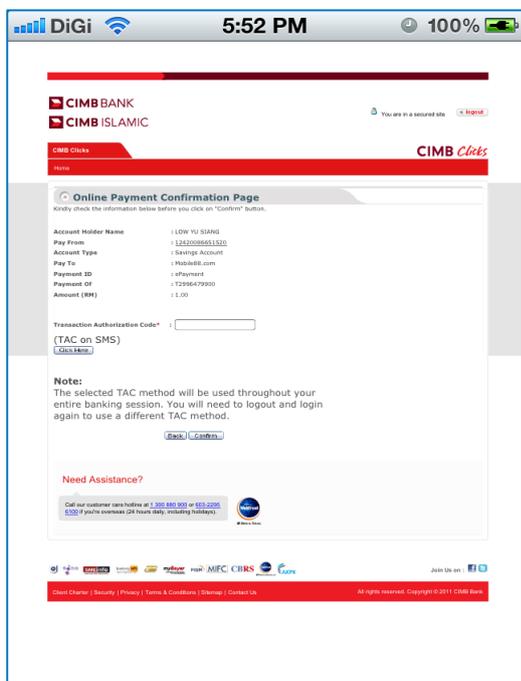


Summary of payment display on CIMB page and Press 'Submit' button to confirm.

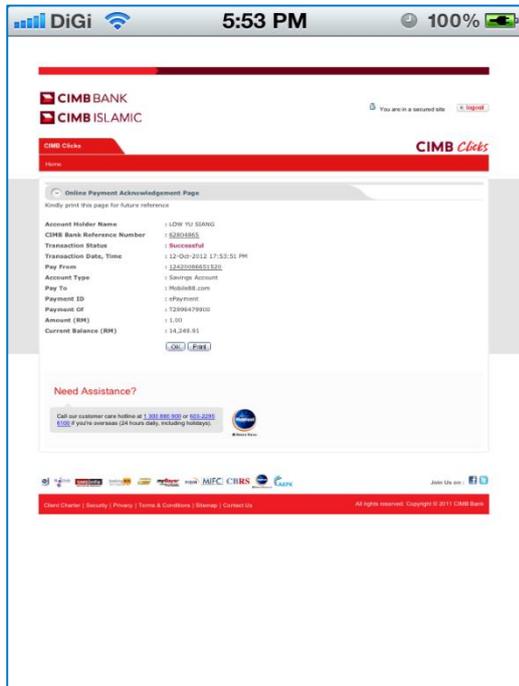


Request a TAC by press 'Click here' button.

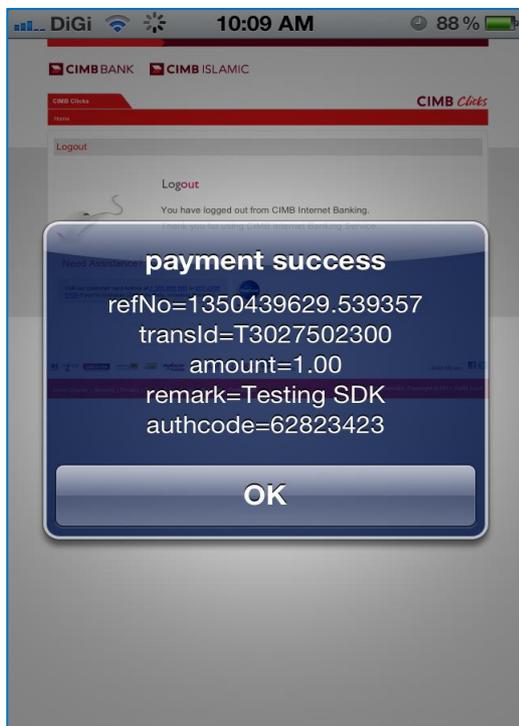
Key in the received TAC and press 'Confirm' to confirm the payment.



CIMBClicks Payment Success Summary page.

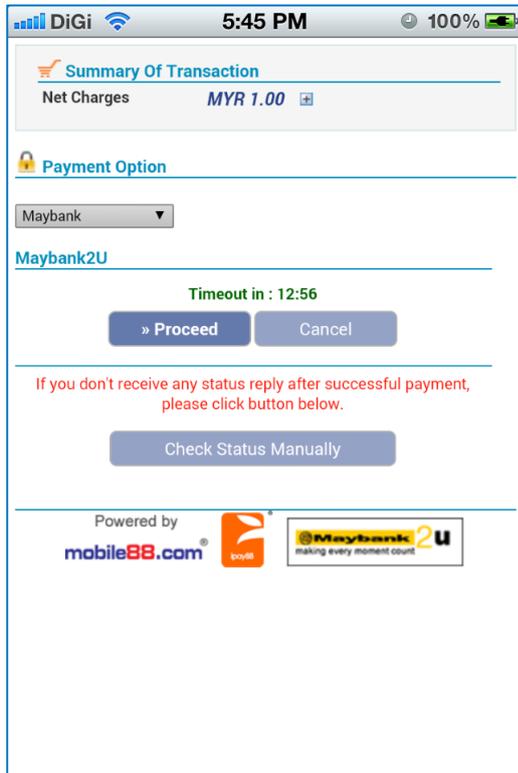


Result will send to your Result delegate methods

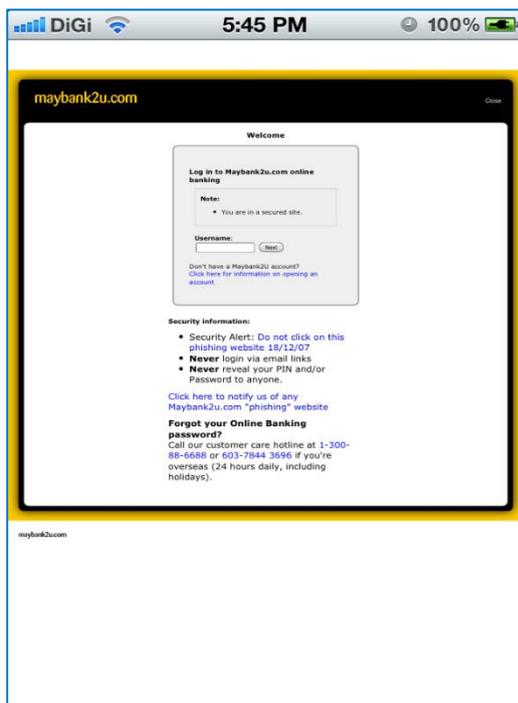


3.3 iPay88 iOS Mobile – Maybank2u

Customer reached to iPay88 payment page. Press 'Proceed' button to Login to Maybank2U to make payment.



Key-in Username and press 'Next'

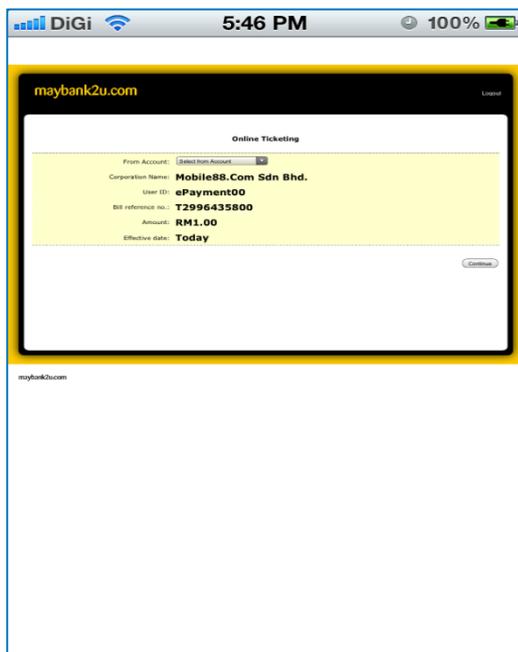


Key-in Password, and press 'Login'.



Summary of payment display on Maybank2U page.

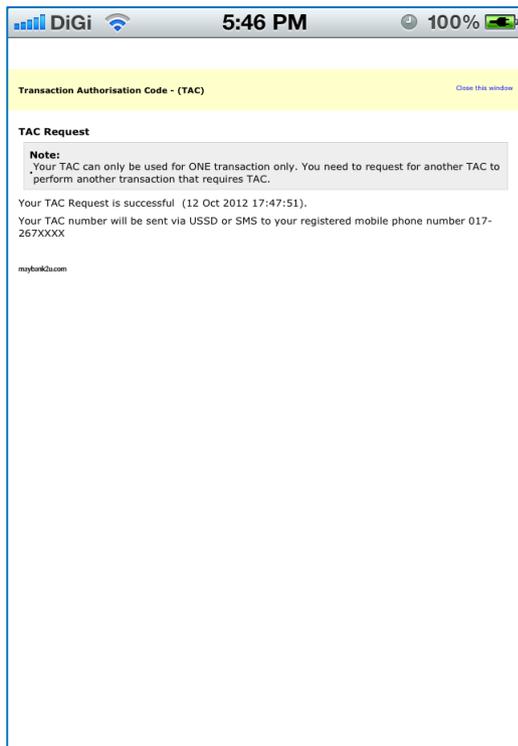
Press 'Continue' to confirm.



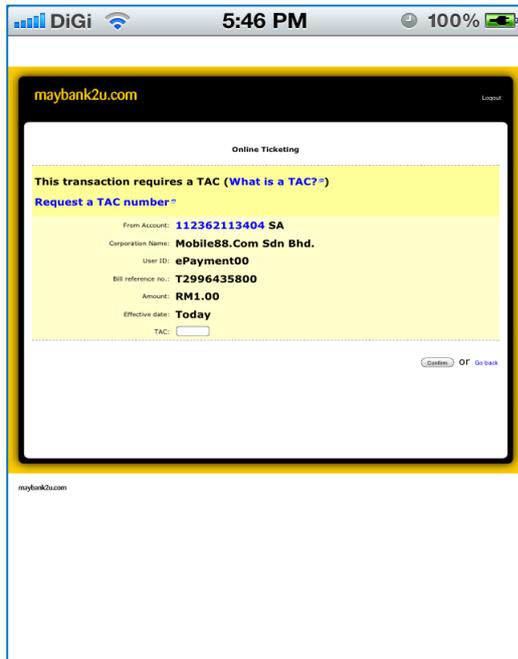
Request a TAC by press 'Request a TAC number' link.



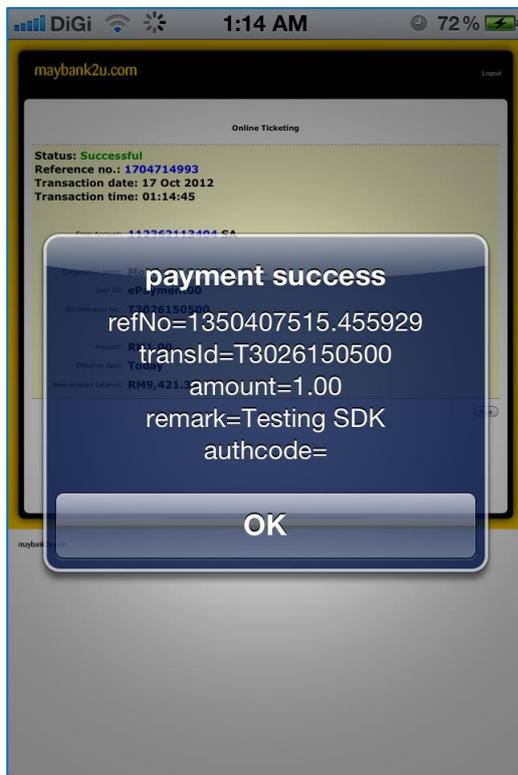
Press 'Close this Window' to close Maybank2U TAC pop up window.



Key-in TAC that received from registered mobile phone and press 'Confirm'.



Result will sent to your Result delegate methods.



4 Appendix I

4.1 PaymentId

If PaymentId not post via request, gateway will choose the predefined default payment method. By default, payment method can be re-select by customer from iPay88 payment.

All payment method below is for MYR currency ONLY.

Payment Method	Payment Id
Credit Card (MYR)	2
Maybank2U	6
Alliance Online	8
AmOnline	10
RHB Online	14
Hong Leong Online	15
CIMB Click	20
Web Cash	22
Public Bank Online	31
PayPal (MYR)	48
Credit Card (MYR) Pre-Auth	55
Bank Rakyat Internet Banking	102
Affin Online	103
Pay4Me (Delay payment)	122
BSN Online	124
Bank Islam	134
UOB	152
Hong Leong PEx+ (QR Payment)	163
Bank Muamalat	166
OCBC	167
Standard Chartered Bank	168
CIMB Virtual Account (Delay payment)	173
HSBC Online Banking	198
Kuwait Finance House	199
Boost Wallet	210

Note:

- For Pay4ME and CIMB Virtual Account payment, iPay88 will initially return Status=6 (Pending Payment) to merchant as this payment method are not real-time payment.
- Once user make payment, iPay88 will return the Status=1 (Success) to merchant BackendURL thus merchant need to ensure had implement BackendURL.
- Generated CIMB Virtual Account number is valid for 7 days and user should make payment within these 7 days.

4.2 Currency

Currency Description	Currency
Malaysian Ringgit	MYR

4.3 Error Description

Error Message	Error Description
Duplicate reference number	Reference number must be unique for each transaction.
Invalid merchant	The merchant code does not exist.
Invalid parameters	Some parameter posted to iPay88 is invalid or empty
Overlimit per transaction	You exceed the amount value per transaction. * For Testing account, only amount RM 1.00 is allowed.
Payment not allowed	The Payment method you requested is not allowed for this merchant code, please contact ipay88 to enable your payment option.
Permission not allow	Referrer URL in for your account registered in iPay88 does not match. Please register your request and response URL with iPay88.
Signature not match	The Signature generated is incorrect.
Status not approved	Account was suspended or not active.