**AsiaPay Mobile SDK**

**Integration Guide for Android Mobile PaySDK**

**PaySDK Architecture Flow**

**A screenshot of a cell phone

Description automatically generated**

**Easy Integration Steps :**

Import provided ***Security Key*** by PayDollar

Import ***PaySDK Library***

Collect***Payment Result***

Initialize

and create a ***Payload***

1

2

4

3

**1.0 Android SDK Integration Steps:**

**SDK REQUIREMENT:**

JAVA SDK Version: 1.8

Android SDK Version: 19 (Minimum), 28 (Target)

**SDK CONFIGURATION:**

1. Copy paydollarsdk-release.aar to**libs** folder.
2. Add below line to project’s gradle file:

repositories {

flatDir {

dirs**'libs'**

}

}

1. Add following libraries

implementation(**name**: **'paydollarsdk-release'**, **ext**: **'aar'**)

implementation **'com.squareup.retrofit2:converter-gson:2.2.0'**implementation **'com.google.code.gson:gson:2.3.1'**implementation **'org.bouncycastle:bcprov-jdk15on:1.60'**implementation **'com.android.volley:volley:1.1.1'**implementation **'com.fasterxml.jackson.core:jackson-core:2.7.3'**implementation **'com.fasterxml.jackson.core:jackson-annotations:2.7.3'**implementation **'com.fasterxml.jackson.core:jackson-databind:2.7.3'**implementation **'org.greenrobot:eventbus:3.0.0'**implementation **'com.google.android.gms:play-services-ads:11.8.0'**implementation **'com.google.android.gms:play-services-location:11.8.0'**implementation **'com.google.android.material:material:1.2.0-alpha04'**implementation **'com.google.android.material:material:1.0.0-beta01'**

**SDK INITIALIZATION:**

1. Create the paysdk.properties file in the assets folder with the

following attributes

RSA Public Key(SDK) :- Here you need to specify rsa public key value. The value will contain

will be provided when the merchant apply the SDK service.

e.g.

sdk\_rsa\_publickey=<<KEY>>

NOTE: Naming of property file and its attributes must be same.

1. Instantiate PaySDK class with context.

*PaySDKpaySDK = new PaySDK(getApplicationContext());*

**1.1 WEBVIEW PAYMENT CALL:**

**Prepare Payment Call :**

Initialize the PayData object and prepare the payment detail for the transaction.

PayDat**apayData**= **new** PayData();

**payData**.setChannel(EnvBase.PayChannel.***WEBVIEW***);

**payData**.setEnvType(EnvBase.EnvType.***SANDBOX***);

**payData**.setAmount(**"1"**);

**payData**.setPayGate(EnvBase.PayGate.***PAYDOLLAR***);

**payData**.setCurrCode(EnvBase.Currency.***HKD***);

**payData**.setPayType(EnvBase.PayType.***NORMAL\_PAYMENT***);

**payData**.setOrderRef(**“0123456789”**);

**payData**.setPayMethod(**"CC"**); //For webviewpayMethod is CC

**payData**.setLang(EnvBase.Language.***ENGLISH***);

**payData**.setMerchantId(**“123456”**);

**payData**.setRemark(**" "**);

*// Optional Parameter (For Value-Added Service)*

Map<String, String>extraDataHosted = **new** HashMap<String, String>();

**payData**.setExtraData(extraDataHosted);

**paySDK**.setRequestData(**payData**);

**Prepare Response Handler :**

Initialize a payment event handler to capture the payment response and result.

**paySDK**.responseHandler(**new** PaymentResponse() {

@Override

**public void** getResponse(PayResultpayResult) {

Toast.*makeText*(AuthActivity.**this**, payResult.getErrMsg(), Toast.***LENGTH\_SHORT***).show();

}

@Override

**public void** onError(Data data) {

Toast.*makeText*(AuthActivity.**this**,data.getError(),Toast.***LENGTH\_SHORT***).show();

}

});

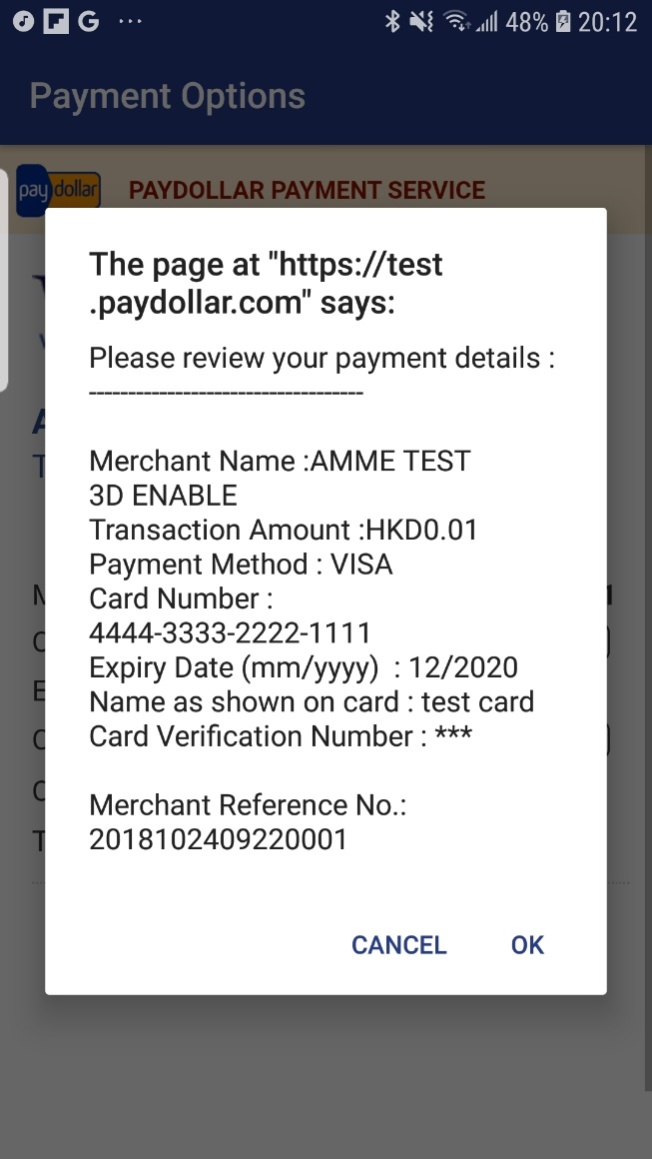
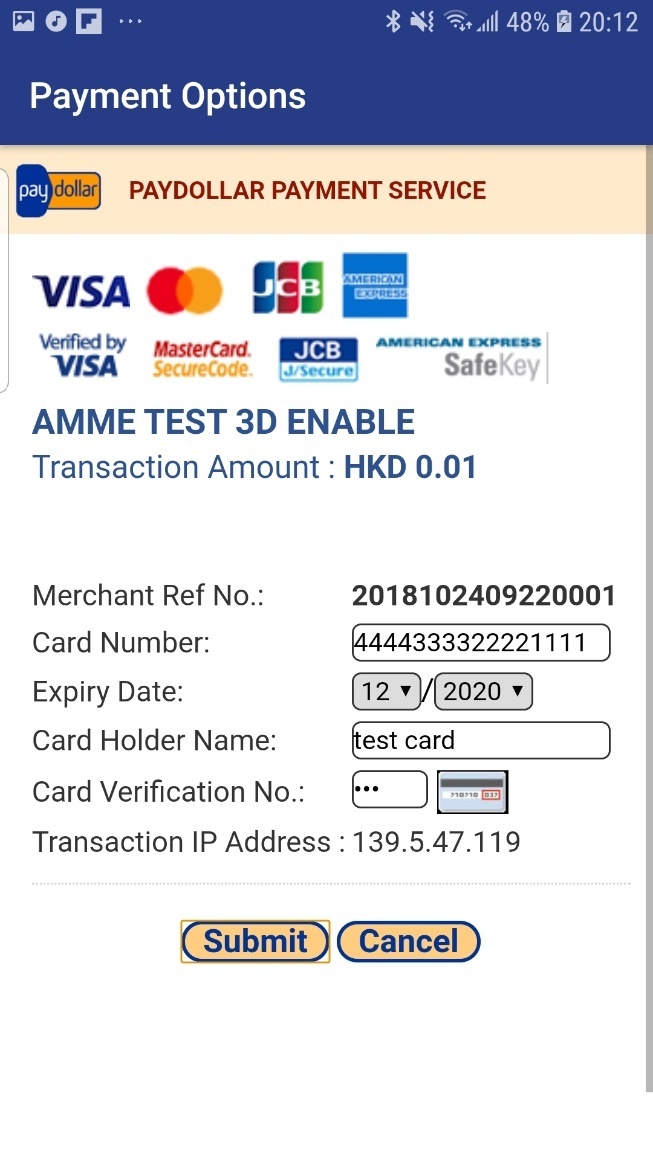
This method returns object of PayResult class.

**Trigger the Payment:**

Call the SDK to trigger the payment process. For WebView flow, it will display a In-App Webview for the payment.

***paySDK****.process();*

**WebView Sample :**



**Avoid memory leak :-**

Here all unused objects will be set to null and clean up the payData and payResult object.

*paySDK.cleanMemory();*

**1.2 Direct Payment Call:**

Prepare Payment Call:

1.2.1 Initialize the PayData class and prepare the payment detail for the transaction.

**payData= new PayData();**

**payData**= **new** PayData();

**payData**.setMerchantId(“123456”);

**payData**.setChannel(EnvBase.PayChannel.***DIRECT***);

**payData**.setEnvType(EnvBase.EnvType.***SANDBOX***);

**payData**.setAmount(“500”);

**payData**.setPayGate(EnvBase.PayGate.***PAYDOLLAR***);

**payData**.setCurrCode(EnvBase.Currency.***RMB***);

**payData**.setPayType(EnvBase.PayType.***NORMAL\_PAYMENT***);

**payData**.setOrderRef(“01234567891”);

**payData**.setPayMethod(**"VISA"**);

**payData**.setLang(EnvBase.Language.***ENGLISH***);

1.2.2 Add Card Details.

CardDetailscardDetails=**new** CardDetails();

cardDetails.setCardNo(**"4548890133258926"**);

cardDetails.setEpMonth(**"08"**);

cardDetails.setEpYear(**"2020"**);

cardDetails.setSecurityCode(**"123"**);

cardDetails.setCardHolder(**"test Card"**);

**payData**.setCardDetails(cardDetails);

**payData**.setCardDetails(cardDetails);

**payData**.setRemark(**" "**);

*// Optional Parameter (For Value-Added Service)*

Map<String, String>extraData = **new** HashMap<String, String>();

**payData**.setExtraData(extraData);

**paySDK**.setRequestData(**payData**);

**Prepare Response Handler :**

Initialize a payment event handler to capture the payment response and result.

**paySDK**.responseHandler(**new** PaymentResponse() {

@Override

**public void** getResponse(PayResultpayResult) {

Toast.*makeText*(AuthActivity.**this**, payResult.getErrMsg() + **""**, Toast.***LENGTH\_SHORT***).show();

}

@Override

**public void** onError(Data data) {

Toast.*makeText*(AuthActivity.**this**,data.getError(),Toast.***LENGTH\_SHORT***).show();

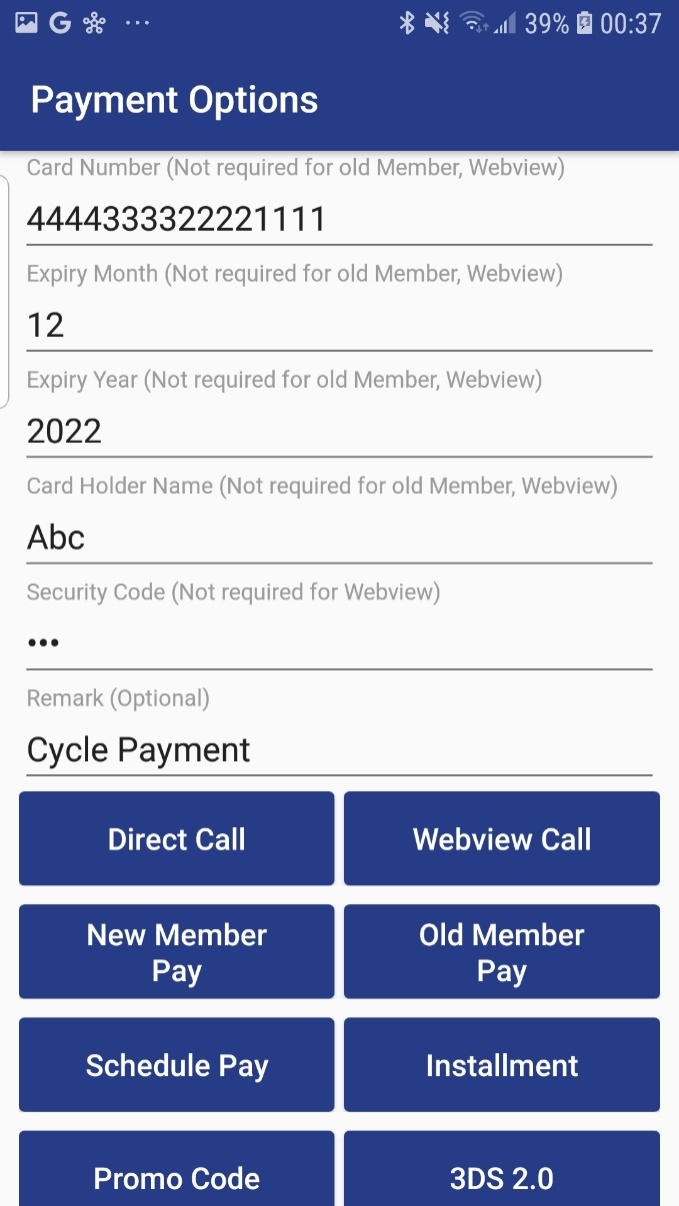
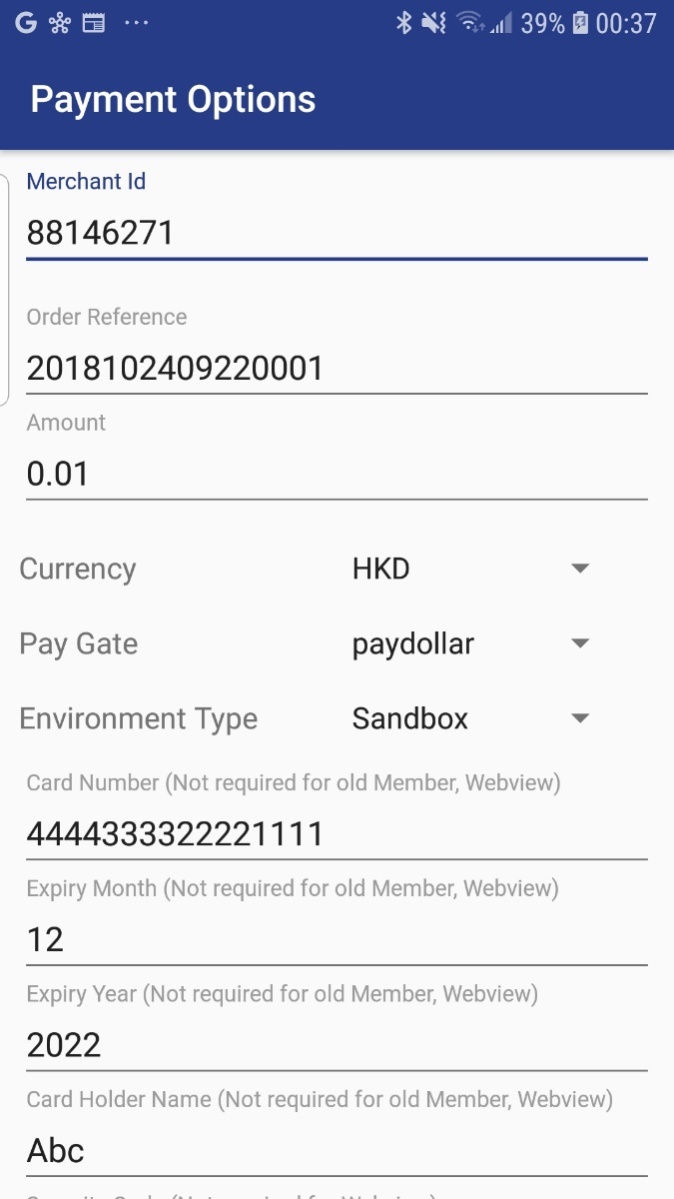
}

});

**Trigger the Payment:**

**Payment will start at this method.**

***paySDK****.process();*



**1.3 Schedule Pay**

For Schedule Pay with Webview(1.1) or Direct (1.2) Payment follow the same process & add below extra params.

extraDataSP.put**("schType","Day");** *// Value could be Day/Month/Year*

extraData.put(**"schStatus"**,**"Active"**); *// Value could be Active/Suspend*

extraData.put(**"nSch"**,**"1"**);

extraData.put(**"sMonth"**,**"4"**);

extraData.put(**"sDay"**,**"26"**);

extraData.put(**"sYear"**,**"2019"**);

extraData.put(**"eMonth"**,**""**);

extraData.put(**"eDay"**,**""**);

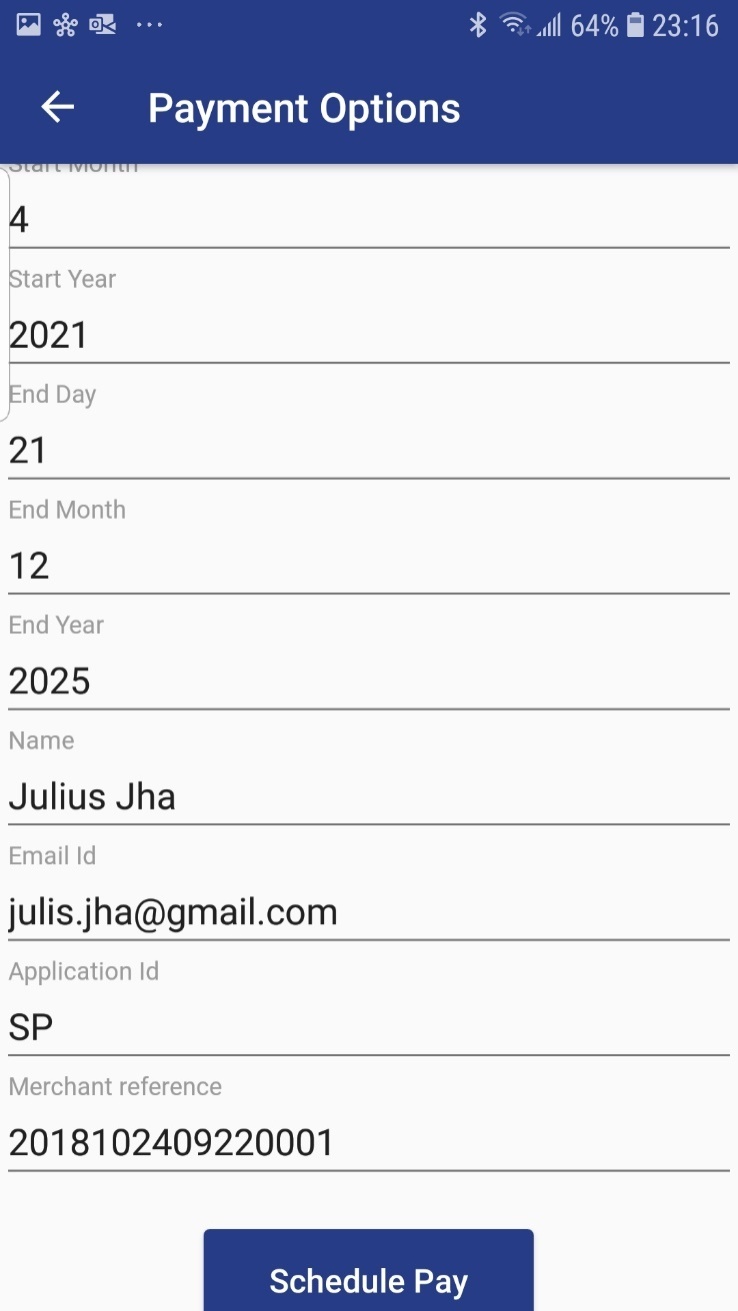
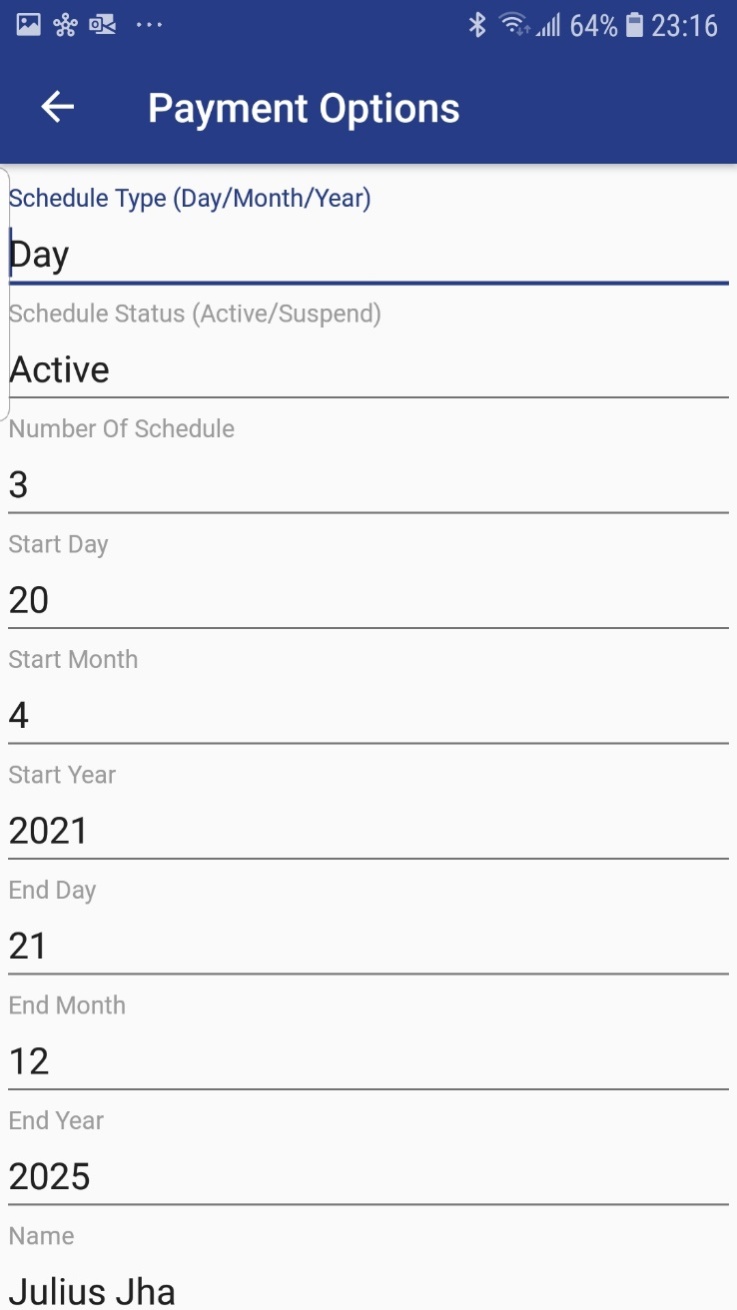
extraData.put(**"eYear"**,**""**);

extraData.put(**"name"**,**"Name"**);

extraData.put(**"email"**,**"kit@gmail.com"**);

*extraData.put(****"appId"****,****"SP"****);*

extraData.put(**"appRef"**,**"2018102409220001"**); *//appRef should be used only once*



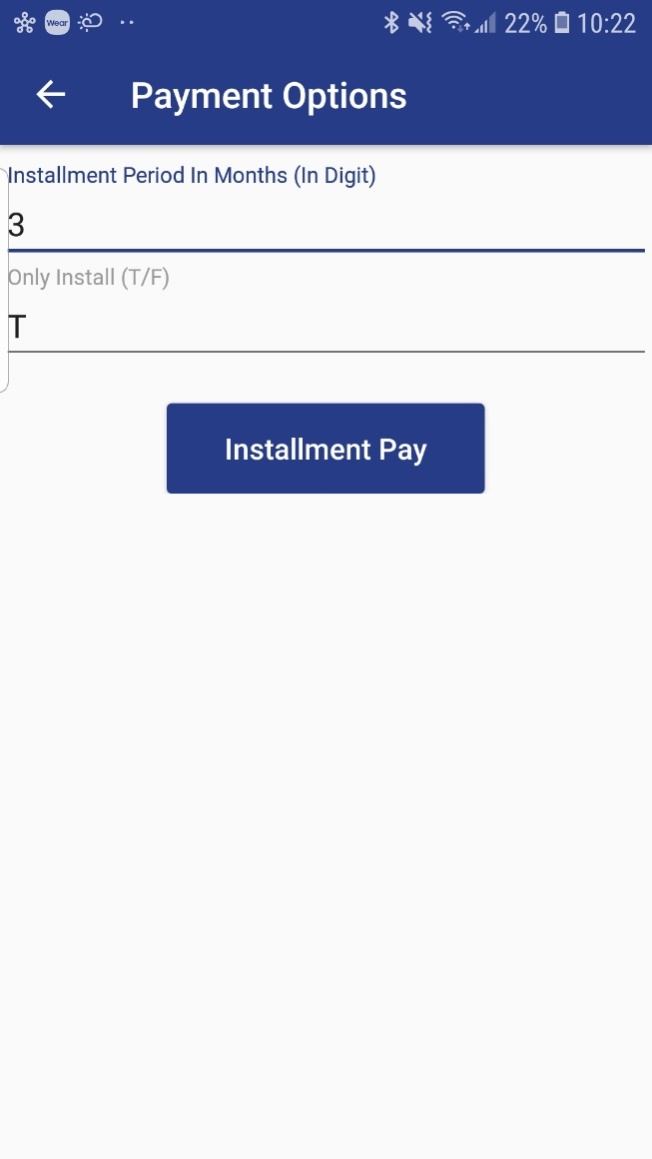
**1.4 Installment Pay**

For Installment Pay with Webview(1.1) or Direct (1.2) Payment follow the same process & add below extra params.

extraData.put(**"installment\_service","T"**);

extraData.put(**"installment\_period"**,**"6"**);

extraData.put(**"installOnly"**,**"T"**);



**1.5 Member Pay**

For Member Pay with Webview(1.1)or Direct (1.2) Payment follow the same process & add below extra params.

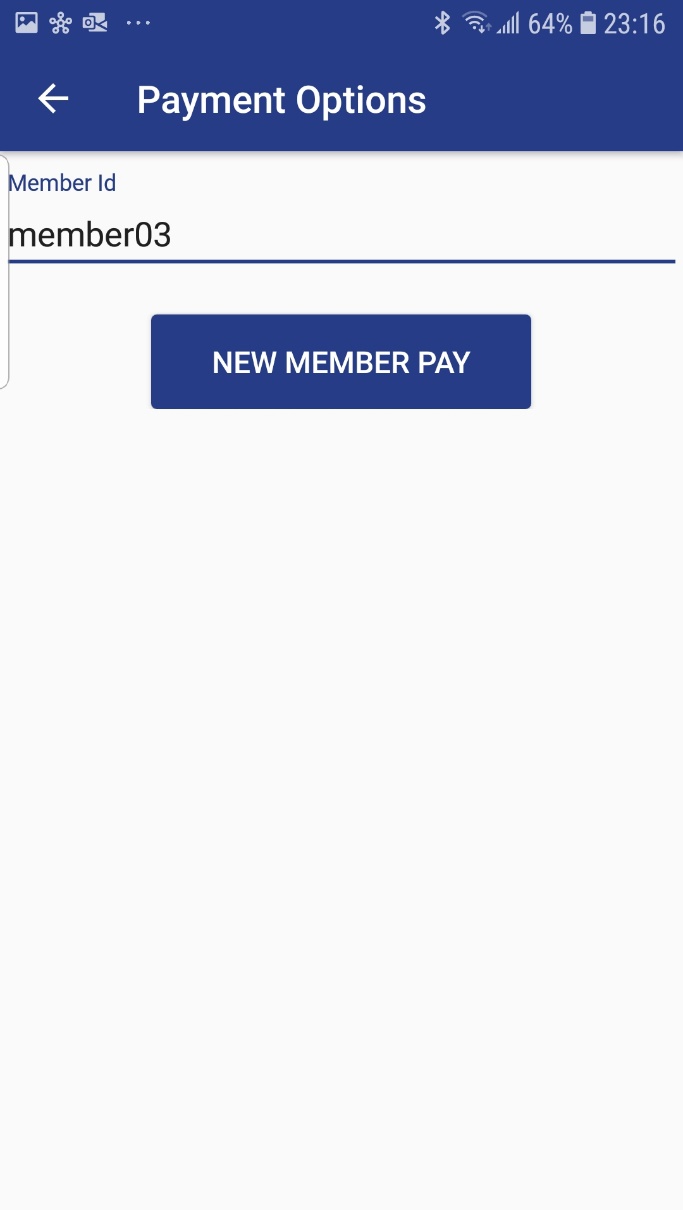
**New Member-**

extraDataMP.put**("memberPay\_service","T");**

extraDataMP.put(**"memberPay\_memberId"**,**"member01"**);

extraDataMP.put(**"addNewMember"**,**"True"**); *// If merchant making payment for first time then value should be true else false*

***NOTE : If merchant making payment for first time then add Card Details 1.2.2***



**Old Member-**

For step 1.2.2 only Security code is required.

CardDetailscardDetails=**new** CardDetails();

cardDetails1.setSecurityCode(**"123"**);

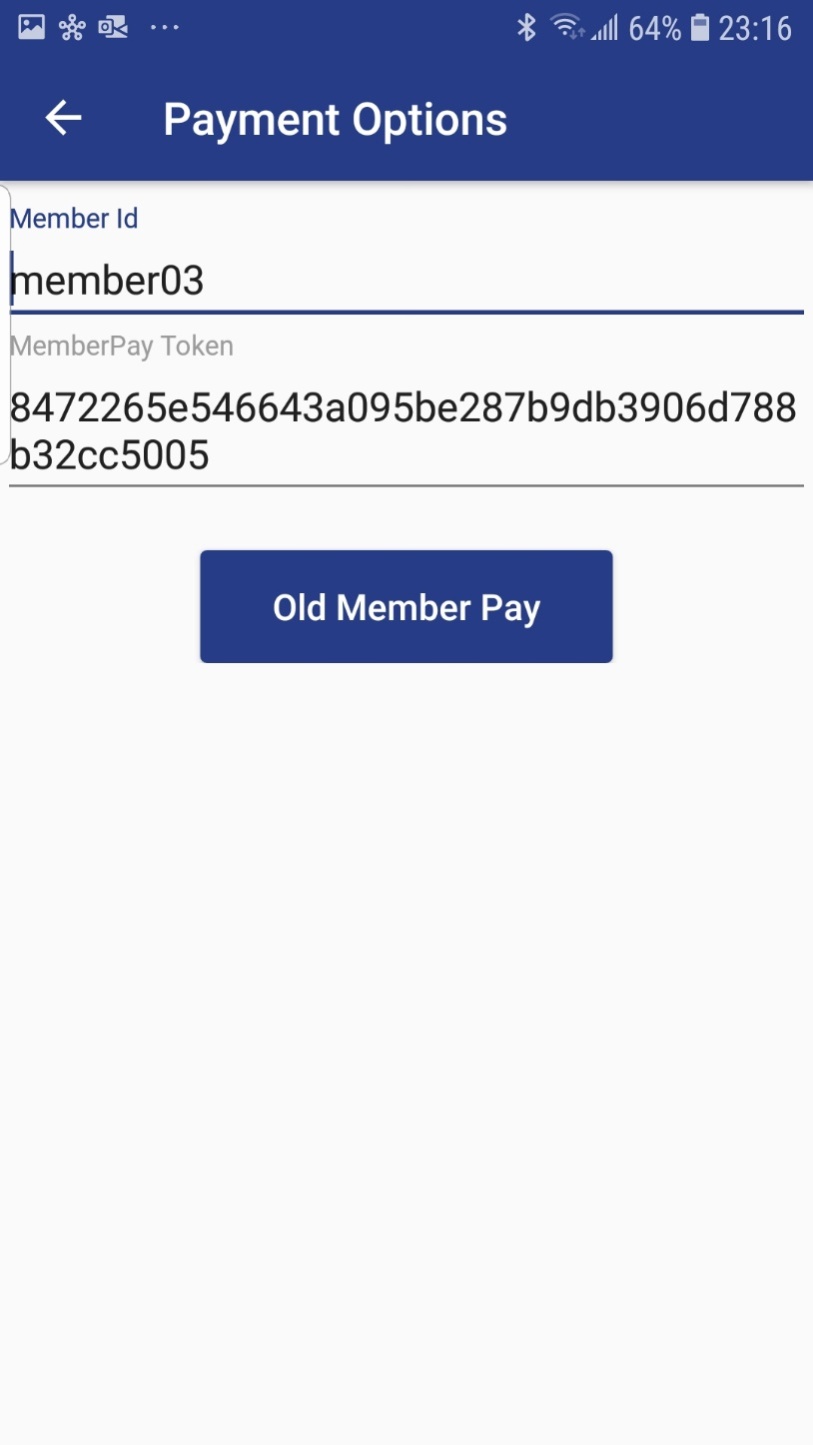
**payData**.setCardDetails(cardDetails);

extraDataOMP.put(**"memberPay\_service"**,**"T"**);

extraDataOMP.put(**"addNewMember"**,**"false"**); *// If merchant making payment for first time then value should be true else false*

extraDataOMP.put(**"memberPay\_memberId"**,**"member03"**);

extraDataOMP.put(**"memberPay\_token"**,**"d4920334a7ff564aa29b7a7732625f891c4f7d235005"**);



**1.6 Promo Pay**

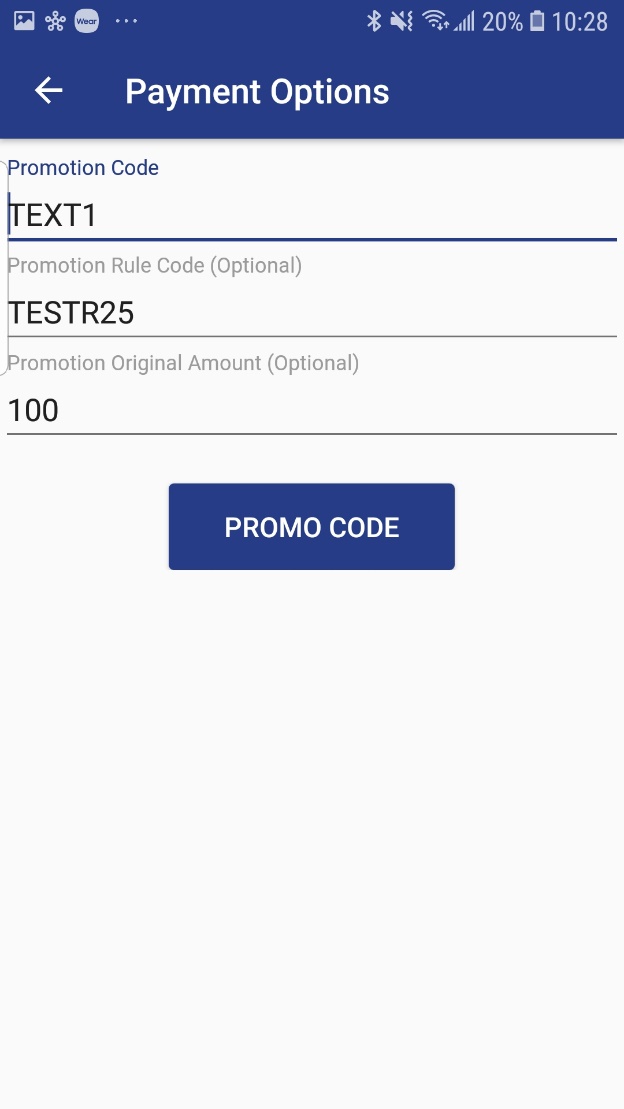
For Promo Pay with Webview(1.1) or Direct (1.2) Payment follow the same process & add below extra params.

*extraDataPC.put(****"promotion"****,****"T"****);*

extraDataPC.put(**"promotionCode"**,**"TEST1"**);

extraDataPC.put(**"promotionRuleCode"**,**""**);

extraDataPC.put(**"promotionOriginalAmt"**,**""**);



**1.7 3DS**

To use 3DS for payment add following ThreedsParam with Direct Call (1.2)

**ThreeDSParamsthreeDSParams**= **new** ThreeDSParams();

**threeDSParams**.setCustomerEmail(**"example@example.com"**);

**threeDSParams**.setMobilePhoneCountryCode(**"852"**);

**threeDSParams**.setMobilePhoneNumber(**"9000000000"**);

**threeDSParams**.setHomePhoneCountryCode(**"852"**);

**threeDSParams**.setHomePhoneNumber(**"8000000000"**);

**threeDSParams**.setWorkPhoneCountryCode(**"852"**);

**threeDSParams**.setWorkPhoneNumber(**"7000000000"**);

**threeDSParams**.setDeliveryEmail(**"example@example.com"**);

**threeDSParams**.setBillingCountryCode(**"344"**);

**threeDSParams**.setBillingState(**""**);

**threeDSParams**.setBillingCity(**"Hong Kong"**);

**threeDSParams**.setBillingLine1(**"BillingLine1"**);

**threeDSParams**.setBillingLine2(**"BillingLine2"**);

**threeDSParams**.setBillingLine3(**"BillingLine3"**);

**threeDSParams**.setBillingPostalCode(**"121245"**);

**threeDSParams**.setShippingDetails(**"01"**);

**threeDSParams**.setShippingCountryCode(**"344"**);

**threeDSParams**.setShippingState(**""**);

**threeDSParams**.setShippingCity(**"Hong Kong"**);

**threeDSParams**.setShippingLine1(**"ShippingLine1"**);

**threeDSParams**.setShippingLine2(**"ShippingLine2"**);

**threeDSParams**.setShippingLine3(**"ShippingLine3"**);

**threeDSParams**.setAcctCreateDate(**"20190401"**);

**threeDSParams**.setAcctAgeInd(**"01"**);

**threeDSParams**.setAcctCreateDate(**"20190401"**);

**threeDSParams**.setAcctAgeInd(**"01"**);

**threeDSParams**.setAcctLastChangeDate(**"20190401"**);

**threeDSParams**.setAcctLastChangeInd(**"01"**);

**threeDSParams**.setAcctPwChangeDate(**"20190401"**);

**threeDSParams**.setAcctPwChangeInd(**"01"**);

**threeDSParams**.setAcctPurchaseCount(**"10"**);

**threeDSParams**.setAcctCardProvisionAttempt(**"0"**);

**threeDSParams**.setAcctNumTransDay(**"0"**);

**threeDSParams**.setAcctNumTransYear(**"1"**);

**threeDSParams**.setAcctPaymentAcctDate(**"20190401"**);

**threeDSParams**.setAcctPaymentAcctInd(**"01"**);

**threeDSParams**.setAcctShippingAddrLastChangeDate(**"20190401"**);

**threeDSParams**.setAcctShippingAddrLastChangeInd(**"01"**);

**threeDSParams**.setAcctIsShippingAcctNameSame(**"T"**);

**threeDSParams**.setAcctIsSuspiciousAcct(**"F"**);

**threeDSParams**.setAcctAuthMethod(**"01"**);

**threeDSParams**.setAcctAuthTimestamp(**"20190401"**);

**threeDSParams**.setDeliveryTime(**"04"**);

**threeDSParams**.setPreOrderReason(**"01"**);

**threeDSParams**.setPreOrderReadyDate(**"20190401"**);

**threeDSParams**.setGiftCardAmount(**"1"**);

**threeDSParams**.setSdkMaxTimeout(**"05"**);

**threeDSParams**.setGiftCardCurr(EnvBase.Currency.***HKD***);

**threeDSParams**.setGiftCardCount(**"1"**);

**payData**.setThreeDSParams(**threeDSParams**);

Factory factory = **new** com.asiapay.sdk.integration.xecure3ds.Factory();

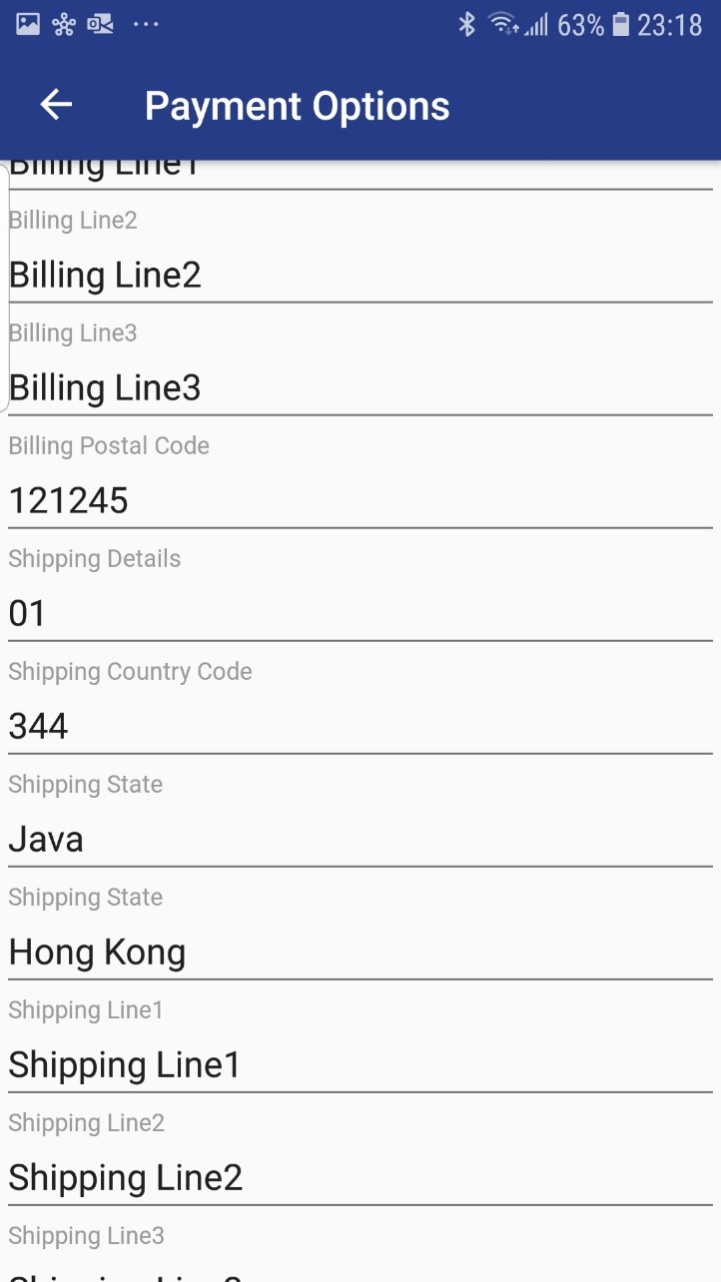
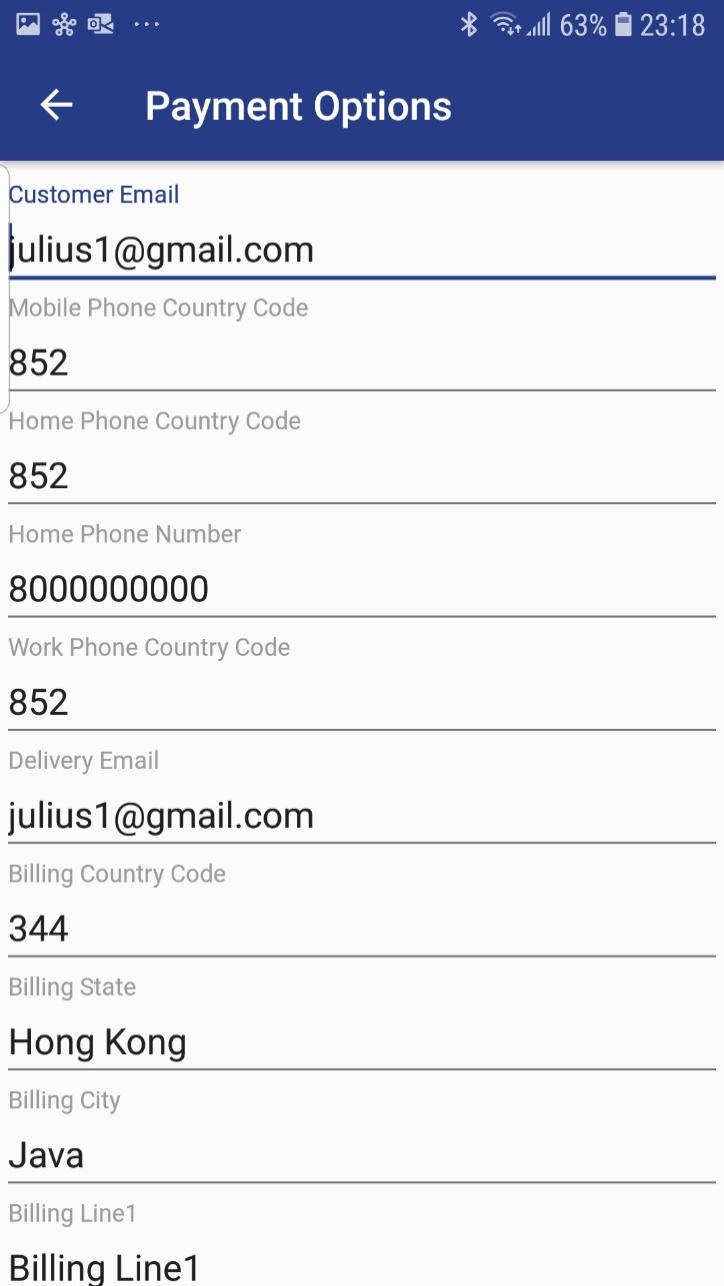
ConfigParametersconfigParameters = factory.newConfigParameters();

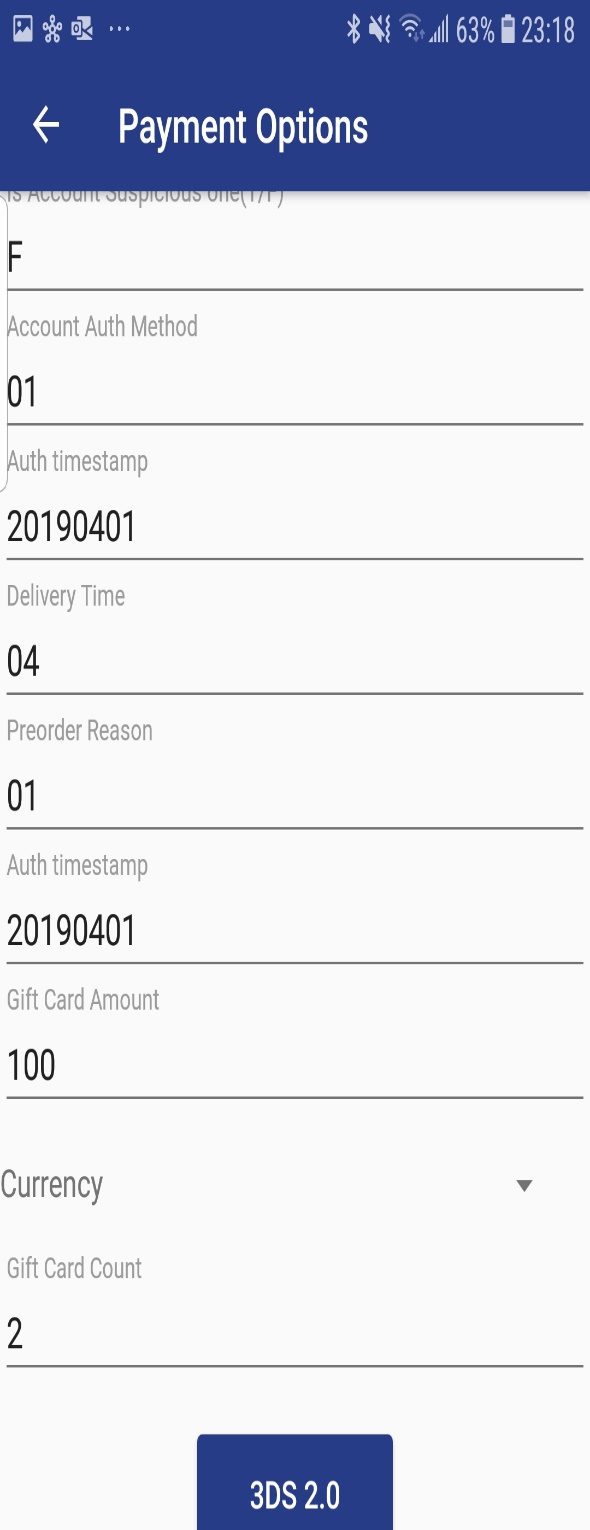
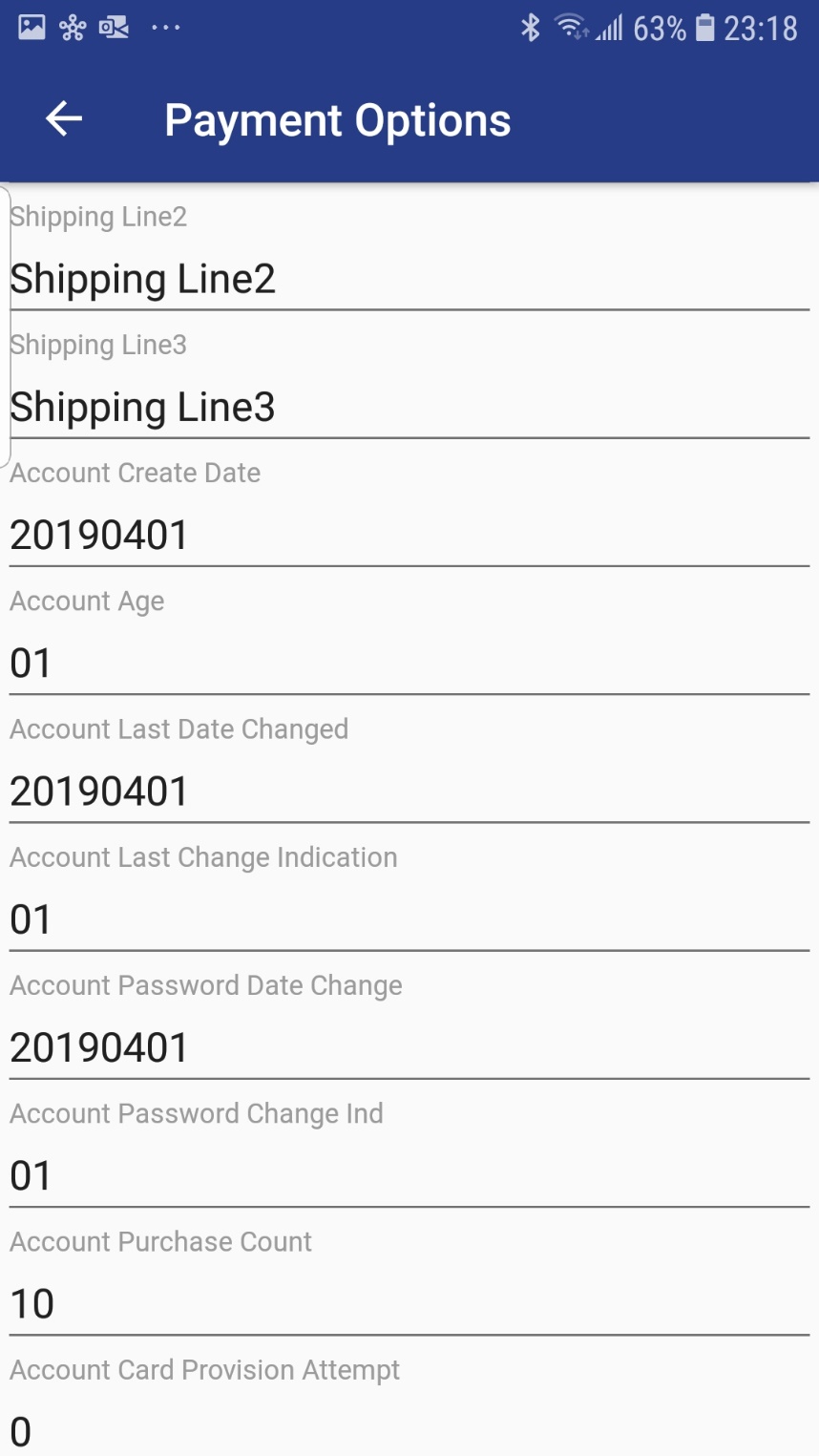
UiCustomizationuiCustomization = factory.newUiCustomization();

**payData**.setConfigParameters(configParameters);

**payData**.setUiCustomization(uiCustomization);

**payData**.setActivity(AuthActivity.**this**);





**1.8 Alipay**

1.7.1 Merge following step to 1.0 **Android SDK Integration Steps**

1. Add alipaySdk-15.5.9-20181123210601.aar to**libs** folder.
2. *Add following dependencies*

implementation(name: 'alipaySdk-15.5.9-20181123210601', ext: 'aar')

implementation "com.tencent.mm.opensdk:wechat-sdk-android-with-mta:+"

1. Sync gradle file: File  Sync with file system

1.7.2 Update below two methods for alipay payment 1.2.1.

**payData**.setActivity(AuthActivity.**this**);

**payData**.setPayMethod(**"ALIPAYAPP"**);

**NOTE: For Alipay Card Details (1.2.2) not required.**

**Class PayData properties details:**

|  |  |
| --- | --- |
| *setChannel* | Channel Type (EnvBase.PayChannel) – WebView, Direct |
| *setPayGate* | Name of payment gateway i.e. Paydollar or Pesopay or Siampay |
| *setEnvType* | Production or Sandbox |
| *setMerchantID* | ID received after registration on Asiapay merchant portal |
| *setPayType* | Payment Type (Normal / Hold) |
| *setOrderRef* | Reference number that merchant wants to attach to transaction |
| *setPayMethod* | Payment Method -  Foe Channel Type  Webview – CC ,  DIrect – VISA ,  3DS with Direct – 3DSSDK |
| *setLang* | Language of Payment Page |
| *setCurrency* | Transaction currency, e.g. HKD, INR etc. |
| *setAmount* | Transaction amount |
| *setAddNewMember* | If merchant making payment for first time then value should be true else false |
| *setRemark* | More details about transaction to tag it. This is optional. |
| *setCardDetails* | SetCardNo – Card number  setEpMonth – Expiry Month on the Card  setEpYear- Expiry Year on the Card  setSecurityCode – 3 digit Security Code on the back side of card  setCardHolder – Name on the Card |
| setExtraData | promotion – Promotion service indicator I.e T/F  promotionCode – The promotion campaign code  promotionRuleCode – The promotion rule code (Optional)  promotionOriginalAmt – The promotion original amount (Optional)  memberPay**\_**service – MemberPay service indicator I.e T/F  memberPay**\_**memberId – Member ID  installment**\_**service – Installment service indicator i.e T/F  installment**\_**period - In number of months  installOnly - Control the mandatory of installment payment i.e. T / F  appId - Application Id  appRef - appref is the reference for the whole schedule pay which will be used only one time  schType - Schedule Type could be Day/Month/Year  schStatus- Schedule Status could be Active/Suspend  nSch - Number of schedule  sMonth - Schedule start month  sDay - Schedule start day  sYear - Schedule start year  eMonth - Schedule end month  eDay - Schedule end day  eYear - Schedule end year  Extra Parameter, please reference to Integration Guide |

**Class EnvBase values details:**

|  |  |
| --- | --- |
| PayChannel | SDK payment channel  WEBVIEW: WebView redirect payment  DIRECT: Direct server payment |
| *PayGate* | SDK supported payment gateway  PAYDOLLAR: PayDollar  SIAMPAY: SiamPay  PESOPAY: PesoPay |
| *EnvType* | SDK environment configuration  PRODUCTION: Production environment  SANDBOX: Sandbox testing environment |
| *PayType* | SDK payment type  NORMAL\_PAYMENT: Sales payment  HOLD\_PAYMENT: Authorize payment |
| *Currency* | SDK supported currency list  HKD, USD, SGD, RMB, JPY, TWD, AUD, EUR, GBP, CAD, MOP, PHP, THB, MYR, IDR, KRW, BND, NZD, SAR, AED, BRL, INR, TRY, ZAR, VND, DKK, ILS, NOK, RUB, SEK, CHF, ARS, CLP, COP, CZK, EGP, HUF, KZT, LBP, MXN, NGN, PKR, PEN, PLN, QAR, RON, UAH, VEF, LKR, KWD |

**Class PayResult properties details:**

|  |  |
| --- | --- |
| *getSuccesscode* | Status of transaction. |
| *getChannel* | Channel Type (EnvBase.PayChannel) – WebView, Direct |
| *getResultCode* | Transaction status code. |
| *getResultMsg* | Message to describe more about transaction status. Message may come blank if the transaction status is Success. |
| *getOrderID* | Order ID generated by AsiaPay System |
| *getOrderRef* | Order Reference number set before initializing payment call in PayData object. |
| *getBankRef* | Bank Order Reference number |
| *getPrc* | Primary response code  Refer to Integration Guide Appendix A (PRC) |
| *getSrc* | Secondary response code  Refer to Integration Guide Appendix A (SRC) |
| *getTxTime* | Transaction Time |
| *getAmount* | Transaction amount |
| *getCurrency* | Transaction currency, e.g. HKD, INR etc. |
| *getExtraData* | Extra Response data |

**Result code For Alipay message details:**

|  |  |
| --- | --- |
| **Code** | **Message** |
| 0001 | Something went wrong! Some unhandled exception occurred. |
| 0002 | Incomplete PayData details provided. |
| 0003 | The merchant id is not valid! |
| 9000 | Success. |
| 8000 | Processing. User may have pressed back button to come out of Aplipay payment page. |
| 4000 | Alipay system error. |
| 6001 | User cancel Alipay payment |
| 6002 | Network error |
| 6004 | Unknown, need confirm with merchant |