**SDK Developer Guide**

*Project Tracking*

**KAN Developments**

**MAY-2021**

#### 1.0 General Information

**USER MANUAL**

**TABLE OF CONTENTS**

Page #

### 1.0 GENERAL INFORMATION*............................................................................................ 1-1*

### 1.1 System Overview .................................................................................................................... 1-1

### 1.2 Authorized Use Permission .................................................................................................... 1-1

### 1.3 Points of Contact .................................................................................................................... 1-1

### 2.0 PRE-REQUISITE *.............................................................................................................. 2-0*

### 3.0 SDK API SUMMARY *....................................................................................................... 3-0*

**3.1** **Initializtion .................................................................................................................................... 3-1** 3.1.1 Initialization FingoSDK .......................................................................................................................... 3-1

**3.1.2 Setting Fingo SDK Parameters ………………………………………….............................** 3-1

**3.2** **Identification API Reference ........................................................................................................ 3-2**

3.2.1 Enrollment ............................................................................................................................................... 3-2

3.2.2 Identify ................................................................................................................................................... 3-2

**3.3** **Payment Api Reference ........................................................................................................... 3-3** 3.3.1 Payment ................................................................................................................................................... 3-3

3.3.2 Refund ..................................................................................................................................................... 3-3

**3.4 Logger API**.……………………………………………………………………………………………...... **3-4**

3.4.1 Usage ………………………………………………………………………………………………...…3-4

**3.5 Examples** ………………………………………………………………………………………………...... 3-5

### 4.0 SDK CODES *..................................................................................................................... 4-0*

**4.1** **DisplayRequested Msg Code ....................................................................................................... 4-1**

**4.2** **Error Codes …….......................................................................................................................... 4-2**

4.2.1 Enroll & Identify ………………………………………………………………………………………4-2

4.2.2 Payment & Refund .……………………………………………………………………………………4-2

# 1.0 GENERAL INFORMATION

# 1.1 System Overview

TBD

# 1.2 Authorized Use Permission

TBD

# 1.3 Points of Contact

TBD

# 2.0 PRE-REQUISITE

1. Knowledge in Android Development
2. FingoSDK (.aar) file
3. Set of dependencies that needs to be included in the project’s build.gradle file
   1. EventBus 'org.greenrobot:eventbus:3.2.0'
   2. Networking
      1. 'com.squareup.retrofit2:retrofit:2.9.0'
      2. 'com.squareup.okhttp3:logging-interceptor:4.2.1'
   3. GSON 'com.squareup.retrofit2:converter-gson:2.4.0'

# 3.0 SDK FEATURES

# 3.1 Initialization

At startup of the app Fingo SDK needs to be initialized to be able to proceed with api usage.

## 3.1.1 Fingo SDK Init

1. Provide Context for initializing the FingoSDK as follows:

FingoErrorCode fingoInitErrorCode = FingoSDK.*initialize*(@context);

Log.i(TAG, fingoInitErrorCode);

1. After step 1 is finished error code is returned indicating the initialization status
   * 1. H1\_OK if initialization is successful
     2. Any other error code if initialization failed

## 3.1.2 Setting FingoParams

1. Create FingoParams object and specify the required parameters

FingoParams fingoParams = new FingoParams();  
fingoParams.setCloudUrl("cloud\_url");  
fingoParams.setPartnerId("partner\_id");  
fingoParams.setMerchantId("merchant\_id");  
fingoParams.setTerminalId("terminalID");  
fingoParams.setApiKey("api\_key");  
fingoParams.setTemplateKeySeed("template\_key\_seed");

1. Pass the specified Fingoparams to the fingo sdk during to set this params into the fingo SDK.

FingoErrorCode fingoParamsErrorCode = FingoSDK.setFingoParams(fingoParams);

1. After setting “FingoParams” into the SDK the sdk will return if the params r valid or not in the fingoParamsErrorCode.

# 3.2 Identification API Reference

Identification api’s includes (Enrollment and Identify).

## 3.2.1 Enrollment

1. User can enroll a finger using the finger’s vein ID on the fingo cloud, the cloud will provide a unique deterministic finger vein ID for the finger of choice.
2. During the enrollment process if the user’s finger is already enrolled on the cloud, the stored vein ID will be returned immediately in the api response.
3. Enrollment process consists of three finger enroll captures and one verification capture, so total of 4 scans for the enrollment process to be completed.
4. Same Finger should be used during the whole enrollment process.

## 3.2.2 Identify

1. After a successful enrollment on the fingo cloud, user can be identified by the enrolled finger unique biometric template (vein\_id).
2. Identification process consists of one finger capture for the identification process to be completed.
3. If the user is already enrolled on the cloud, his unqiue vein ID will be returned from the cloud, if the user is not enrolled the cloud will return and error. (check the errors section).

# 3.3 Payment API Reference

Identification api’s includes (Enrollment and Identify).

## 3.3.1 Payment

1. After a successful enrollment on the fingo cloud, user can perform payments on the fingo cloud, given that the user already have a predefined card data stored on the cloud..
2. Payment process consists of one finger capture for the payment process to be completed.
3. If the payment failed the cloud will return error (Refer to payment error section), if the payment is successful the below payment data will be returned from the cloud.
   1. Transaction ID
   2. Gateway Transaction ID
   3. Gateway Auth Code
   4. Masked Card Number
   5. Merchant ID
   6. Transaction Timestamp

## 

## 3.3.2 Refund

1. After a successful payment on the fingo cloud, user can refund the already made transaction partial refund or full refund, given that the user already have a predefined card data stored and available payment to refund on the cloud.
2. Refund process consists of one finger capture for the refund process to be completed.
3. If the refund failed the cloud will return error (Refer to payment error section), if the refund is successful the below refund data will be returned from the cloud.
   1. Transaction ID
   2. Gateway Transaction ID
   3. Gateway Auth Code
   4. Masked Card Number
   5. Merchant ID
   6. Transaction Timestamp

# 3.4 Logger API

## 3.4.1 Usage

1. This API allow the third party app to register for fingo request and response logs at the initialization time of the SDK.
2. First third-party app needs to make use of the exposed interface of the lib FingoRequestLogger, either by implementing the interface or using it as a parameter
3. Once the FingoRequestLogger is setup you will need to initialize the FingoSDK using the logger

FingoSDK.*initialize*(@context, new FingoRequestLogger() {  
 @Override  
 public void onLogDataAvailable(String s) {  
 Log.*d*(*TAG*, "onLogDataAvailable: " + s);  
 }  
});

1. Now the third-party app will receive callbacks whenever the SDK tries to go online and contact the Fingo cloud. (THIS API IS USED FOR DEBUG PURPOSES ONLY)

# 3.5 Examples

1. First create a FingoListener that will be responsible for the events received from the SDK api’s and will provide callbacks with results.

FingoContract.FingoListener fingoListener = new FingoContract.FingoListener(){

@Override  
public void onProcessingStarted() {  
 Log.*i*(*TAG*, "onProcessingStarted");  
}  
  
@Override  
public void onDisplayTextRequested(DisplayTextRequested displayTextRequested) {  
 Log.*d*(*TAG*, "onDisplayTextRequested: " + displayTextRequested.getType());  
 Log.*d*(*TAG*, "onDisplayTextRequested: " + displayTextRequested.getText());  
 Log.*d*(*TAG*, "onDisplayTextRequested: " + displayTextRequested.getCode());  
}  
  
@Override  
public void onIdentifyData(IdentifyData identifyData) {  
 Log.*d*(*TAG*, "onOnlineIdentifyData: " + identifyData.getMemberId());  
 Log.*d*(*TAG*, "onOnlineIdentifyData: " + identifyData.getVeinId());  
 Log.*d*(*TAG*, "onOnlineIdentifyData: " + identifyData.getVerificationTemplate());  
 Log.*d*(*TAG*, "onOnlineIdentifyData: " + identifyData.getEnrolmentTemplate());  
}  
  
@Override  
public void onPaymentData(PaymentData paymentData, FingoErrorResponse fingoErrorResponse) {  
 Log.*d*(*TAG*, "onOnlinePaymentData: " + paymentData.getTransactionId());  
 Log.*d*(*TAG*, "onOnlinePaymentData: " + paymentData.getGatewayAuthCode());  
 Log.*d*(*TAG*, "onOnlinePaymentData: " + paymentData.getGatewayTransactionId());  
 Log.*d*(*TAG*, "onOnlinePaymentData: " + paymentData.getMaskedCardNumber());  
 Log.*d*(*TAG*, "onOnlinePaymentData: " + paymentData.getTimestamp());  
 Log.*d*(*TAG*, "onOnlinePaymentData: " + fingoErrorResponse.getFingoErrorList().get(0).getErrorCode());  
 Log.*d*(*TAG*, "onOnlinePaymentData: " + fingoErrorResponse.getFingoErrorList().get(0).getErrorMessage());  
}  
  
@Override  
public void onProcessingFinished(ProcessingFinished processingFinished) {  
 Log.*d*(*TAG*, "onProcessingFinished: " + processingFinished.getText());  
 Log.*d*(*TAG*, "onProcessingFinished: " + processingFinished.getErrorName());  
 Log.*d*(*TAG*, "onProcessingFinished: " + processingFinished.getErrorCode());  
 Log.*d*(*TAG*, "onProcessingFinished: " + processingFinished.getStatus());  
}

}

1. Create a presenter that will be responsible to access the SDK api’s and will provide callbacks with results

fingoPresenter = new FingoPresenter(@activity, fingoListener);

1. To invoke the enrollment API

fingoPresenter.enroll(*TIMEOUT*);

1. To invoke the identify API

fingoPresenter.identify(*TIMEOUT*);

1. To invoke the payment API

PosData posData = new PosData("2", "SomeLocation");  
fingoPresenter.payment(100, Currency.*GBP*, 0,posData, *TIMEOUT*);

1. To invoke the refund API

TerminalData terminalData = new TerminalData();  
terminalData.setLocation("Cairo");  
fingoPresenter.refund(100, "8c04ad1b-e1e8-4752-b50c-e3c9dc70ad11", "96577222", terminalData, *TIMEOUT*);

1. To cancel the finger vein capture session

fingoPresenter.cancel();

1. To check if the capture session is cancelled by the SDK

fingoPresenter.isOperationCancelled()

1. For identification and enrollment API’s, the results will be received in the

onIdentifyData(IdentifyData identifyData)

1. For payment and refund API’s, the results will be received in the

onPaymentData(PaymentData paymentData, FingoErrorResponse fingoErrorResponse)

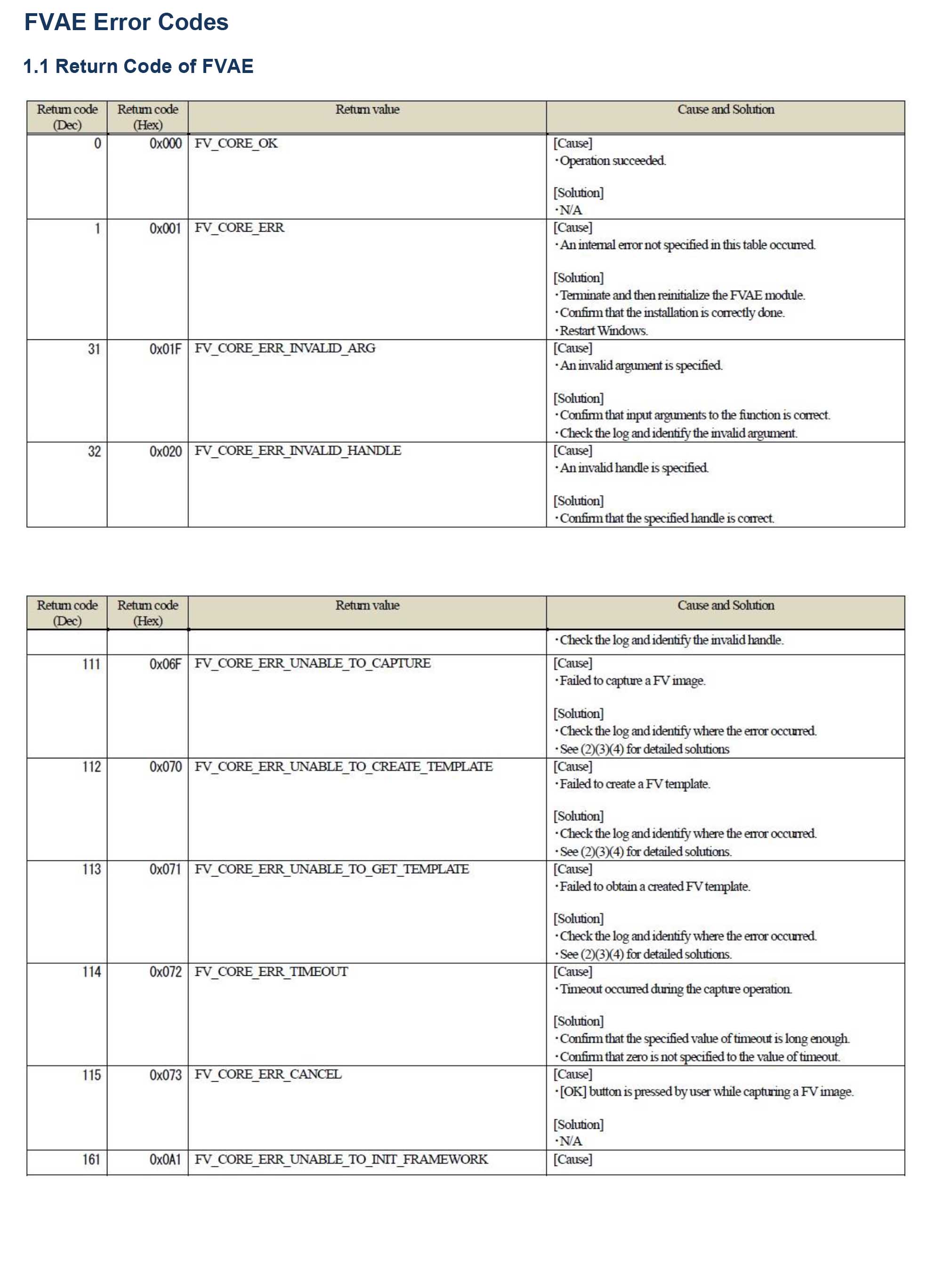
# 4.0 SDK CODES

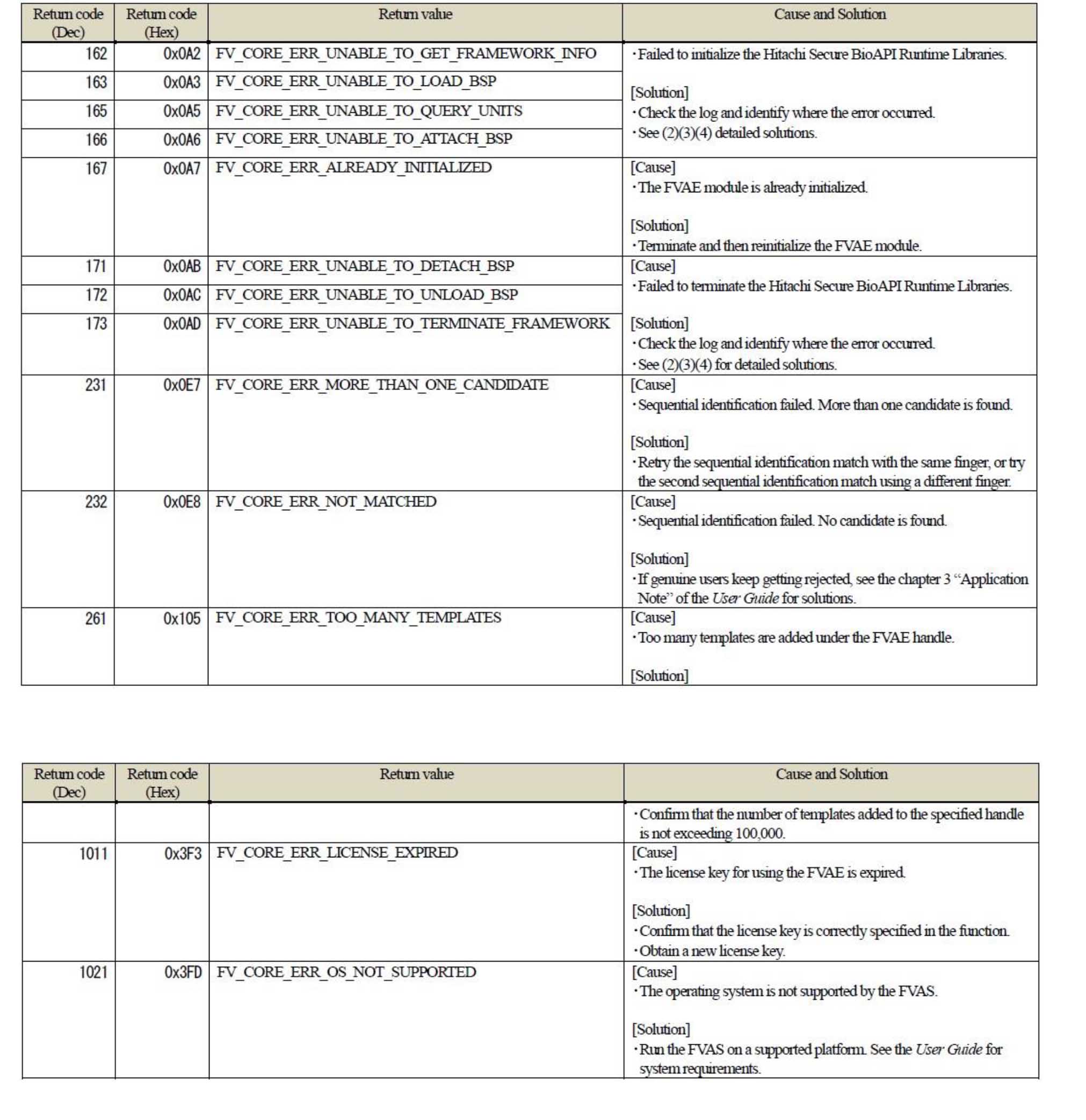
# 4.1 DisplayRequested Msg Code

The below error codes are for the onDisplayTextRequested callback, the callback returns a message in plain English and a unique code that represents this message.

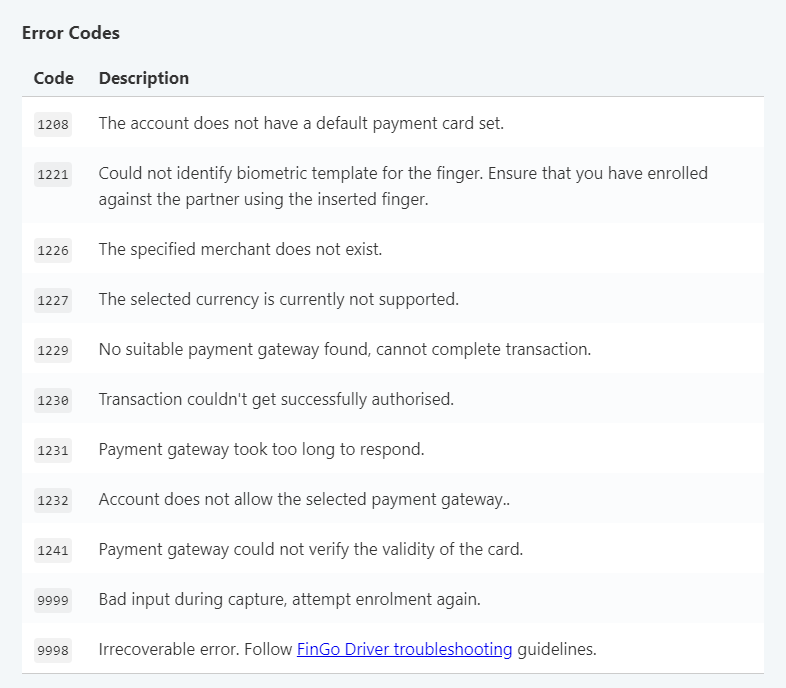
# 4.2 Error Codes

## 4.2.1 Enroll & Identify





## 4.2.2 Payment & Refund



**THE END OF THE DOCUMENT**

**THANKS**