TÀI LIỆU HƯỚNG DẪN TÍCH HỢP SDK FLUTTER

EKYC-NFC

I. Tích hợp SDK vào iOS

1. Yêu cầu

- Yêu cầu có sẵn 1 dự án **Flutter**
- Mở thư mục ios trong dự án Flutter và tích hợp SDK vào dự án theo các bước của tài liệu (phần 1 và 2 trong mục **II. Tích hợp thư viện SDK vào dự án**):
  [VNPT] Hướng_dẫn_tích_hợp_SDK_IOS_ICNFCCardReader_v2.1.6

2. Thêm cầu nối

- Cầu nối với Flutter được cấu hình trong **AppDelegate**
- Tham khảo tệp **AppDelegate** trong dự án tích hợp SDK mẫu
- Một số đoạn code lưu ý:

```
let controller : FlutterViewController = window?.rootViewController as!
FlutterViewController
/// điền tên tương ứng trong dự án Flutter
let channel = FlutterMethodChannel(name: "flutter.sdk.ekyc/integrate",
                                   binaryMessenger: controller.binaryMessenger)
```

```
/// kiểm tra tên method tương ứng bên Flutter để gọi chức năng tương ứng
/// `navigateToNfcQrCode` để gọi NFC với QRcode
/// `navigateToScanNfc` để gọi chỉ quét NFC và nhập các thông tin đầu vào
DispatchQueue.main.async {
    if call.method == "navigateToNfcQrCode" {
        self.actionOpenQRAndNFC(controller)
    } else if call.method == "navigateToScanNfc" {
        self.actionOpenOnlyNFC(controller, idNumberCard: idNumberCard,
birthdayCard: birthdayCard, expiredDateCard: expiredDateCard)
    }
}
```

- Các phương thức còn lại tương tự việc tích hợp SDK cho dự án Native iOS (tham khảo tài liệu đề cập ở phần 1)

II. Tích hợp SDK vào Android

1. Yêu cầu

- Yêu cầu có sẵn 1 dự án **Flutter**
- Mở thư mục android trong dự án Flutter và tích hợp theo các bước theo tài liệu sau:
  [VNPT] Hướng_dẫn_tích_hợp_SDK_ANDROID_ICNFCCardReader_v…

2. Thêm cầu nối

- Tạo file cầu nối tới **Flutter** thông qua **FlutterChannel** tại **MainActivity.kt** như sau:

```
import android.app.Activity
import android.content.Intent
import android.nfc.NfcManager
import com.vnptit.nfc.activity.VnptScanNFCActivity
import com.vnptit.nfc.utils.KeyIntentConstantsNFC
import com.vnptit.nfc.utils.KeyResultConstantsNFC
import com.vnptit.nfc.utils.SDKEnumNFC
import io.flutter.embedding.android.FlutterActivity
import io.flutter.embedding.engine.FlutterEngine
import io.flutter.plugin.common.MethodCall
import io.flutter.plugin.common.MethodChannel
import org.json.JSONObject

class MainActivity : FlutterActivity(), MethodChannel.MethodCallHandler {
  companion object {
     private const val CHANNEL = "flutter.sdk.ekyc/integrate"
     private const val EKYC_REQUEST_CODE = 100
     private const val ERROR_NFC_CODE = "69"
  }

  private lateinit var channel: MethodChannel
  private lateinit var result: MethodChannel.Result

  override fun configureFlutterEngine(flutterEngine: FlutterEngine) {
     super.configureFlutterEngine(flutterEngine)
     channel = MethodChannel(flutterEngine.dartExecutor.binaryMessenger,
CHANNEL)
     channel.setMethodCallHandler(this)
  }

  override fun cleanUpFlutterEngine(flutterEngine: FlutterEngine) {
     super.cleanUpFlutterEngine(flutterEngine)
     channel.setMethodCallHandler(null)
  }
```

```
  override fun onMethodCall(call: MethodCall, result: MethodChannel.Result) {
     this.result = result

     val json = parseJsonFromArgs(call)
     val intent = when (call.method) {
        "navigateToNfcQrCode" -> navigateToNfcQrCode(json)
        "navigateToScanNfc" -> navigateToScanNfc(json)
        else -> {
           result.notImplemented()
           null
        }
     }

     intent?.let {
        if (!isDeviceSupportedNfc()) {
           result.error(ERROR_NFC_CODE, "Thiết bị không hỗ trợ NFC", null)
           return
        }

        activity.startActivityForResult(it, EKYC_REQUEST_CODE)
     }
  }

  override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
     super.onActivityResult(requestCode, resultCode, data)
     if (requestCode == EKYC_REQUEST_CODE) {
        if (resultCode == Activity.RESULT_OK) {
           if (data != null) {
              /**
               * đường dẫn ảnh mặt trước trong thẻ chip lưu trong cache
               * [KeyResultConstantsNFC.IMAGE_AVATAR_CARD_NFC]
               */
              val avatarPath =
data.getStringExtra(KeyResultConstantsNFC.IMAGE_AVATAR_CARD_NFC)

              /**
               * chuỗi thông tin cua SDK
               * [KeyResultConstantsNFC.CLIENT_SESSION_RESULT]
               */
              val clientSession =

data.getStringExtra(KeyResultConstantsNFC.CLIENT_SESSION_RESULT)

              /**
               * kết quả NFC
               * [KeyResultConstantsNFC.LOG_NFC]
               */
              val logNFC = data.getStringExtra(KeyResultConstantsNFC.LOG_NFC)

              /**
```

```
                * mã hash avatar
                * [KeyResultConstantsNFC.HASH_AVATAR]
                */
               val hashAvatar =
data.getStringExtra(KeyResultConstantsNFC.HASH_AVATAR)

               /**
                * chuỗi json string chứa thông tin post code của quê quán
                * [KeyResultConstantsNFC.POST_CODE_ORIGINAL_LOCATION_RESULT]
                */
               val postCodeOriginalLocation =

data.getStringExtra(KeyResultConstantsNFC.POST_CODE_ORIGINAL_LOCATION_RESULT)

               /**
                * chuỗi json string chứa thông tin post code của nơi thường trú
                * [KeyResultConstantsNFC.POST_CODE_RECENT_LOCATION_RESULT]
                */
               val postCodeRecentLocation =

data.getStringExtra(KeyResultConstantsNFC.POST_CODE_RECENT_LOCATION_RESULT)

               /**
                * time scan nfc
                * [KeyResultConstantsNFC.TIME_SCAN_NFC]
                */
               val timeScanNfc =
data.getStringExtra(KeyResultConstantsNFC.TIME_SCAN_NFC)

               /**
                * kết quả check chip căn cước công dân
                * [KeyResultConstantsNFC.CHECK_AUTH_CHIP_RESULT]
                */
               val checkAuthChipResult =

data.getStringExtra(KeyResultConstantsNFC.CHECK_AUTH_CHIP_RESULT)

               /**
                * kết quả quét QRCode căn cước công dân
                * [KeyResultConstantsNFC.QR_CODE_RESULT_NFC]
                */
               val qrCodeResult =
data.getStringExtra(KeyResultConstantsNFC.QR_CODE_RESULT_NFC)

               result.success(
                  JSONObject().apply {
                     putSafe(KeyResultConstantsNFC.IMAGE_AVATAR_CARD_NFC,
avatarPath)
                     putSafe(KeyResultConstantsNFC.CLIENT_SESSION_RESULT,
clientSession)
                     putSafe(KeyResultConstantsNFC.LOG_NFC, logNFC)
```

```
                    putSafe(KeyResultConstantsNFC.HASH_AVATAR, hashAvatar)
                    putSafe(
                       KeyResultConstantsNFC.POST_CODE_ORIGINAL_LOCATION_RESULT,
                       postCodeOriginalLocation
                    )
                    putSafe(
                       KeyResultConstantsNFC.POST_CODE_RECENT_LOCATION_RESULT,
                       postCodeRecentLocation
                    )
                    putSafe(KeyResultConstantsNFC.TIME_SCAN_NFC, timeScanNfc)
                    putSafe(KeyResultConstantsNFC.CHECK_AUTH_CHIP_RESULT,
checkAuthChipResult)
                    putSafe(KeyResultConstantsNFC.QR_CODE_RESULT_NFC,
qrCodeResult)
                 }.toString()
              )
           }
        }
     }
  }

  private fun isDeviceSupportedNfc(): Boolean {
     val adapter = (getSystemService(NFC_SERVICE) as?
NfcManager)?.defaultAdapter
     return adapter != null && adapter.isEnabled
  }

  private fun navigateToNfcQrCode(json: JSONObject): Intent {
     return Intent(this, VnptScanNFCActivity::class.java).also {
        /**
         * Truyền access token chứa bearer
         */
        it.putExtra(KeyIntentConstantsNFC.ACCESS_TOKEN,
json.getString("access_token"))
        /**
         * Truyền token id
         */
        it.putExtra(KeyIntentConstantsNFC.TOKEN_ID, json.getString("token_id"))
        /**
         * Truyền token key
         */
        it.putExtra(KeyIntentConstantsNFC.TOKEN_KEY,
json.getString("token_key"))
        /**
         * điều chỉnh ngôn ngữ tiếng việt
         *    - vi: tiếng việt
         *    - en: tiếng anh
         */
        it.putExtra(KeyIntentConstantsNFC.LANGUAGE_NFC,
SDKEnumNFC.LanguageEnum.VIETNAMESE.value)
        /**
```

```
         * hiển thị màn hình hướng dẫn + hiển thị nút bỏ qua hướng dẫn
         * - mặc định luôn luôn hiển thị màn hình hướng dẫn
         *    - true: hiển thị nút bỏ qua
         *    - false: ko hiển thị nút bỏ qua
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_GOT_IT, true)
        /**
         * bật tính năng upload ảnh
         *    - true: bật tính năng
         *    - false: tắt tính năng
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_UPLOAD_IMAGE, true)
        /**
         * bật tính năng get Postcode
         *    - true: bật tính năng
         *    - false: tắt tính năng
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_MAPPING_ADDRESS, true)
        /**
         * bật tính năng xác thực chip
         *    - true: bật tính năng
         *    - false: tắt tính năng
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_VERIFY_CHIP, true)
        /**
         * truyền các giá trị đọc thẻ
         *    - nếu không truyền gì mặc định sẽ đọc tất cả (MRZ,Verify
Document,Image Avatar)
         *    - giá trị truyền vào là 1 mảng int: nếu muốn đọc giá trị nào sẽ
truyền
         *      giá trị đó vào mảng
         * eg: chỉ đọc thông tin MRZ
         *    intArrayOf(SDKEnumNFC.ReadingNFCTags.MRZInfo.value)
         */
        it.putExtra(
           KeyIntentConstantsNFC.READING_TAG_NFC,
           intArrayOf(
              SDKEnumNFC.ReadingNFCTags.MRZInfo.value,
              SDKEnumNFC.ReadingNFCTags.VerifyDocumentInfo.value,
              SDKEnumNFC.ReadingNFCTags.ImageAvatarInfo.value
           )
        )
        /**
         * truyền giá trị bật quét QRCode
         *    - true: tắt quét QRCode
         *    - false: bật quét QRCode
         */
        it.putExtra(KeyIntentConstantsNFC.IS_TURN_OFF_QR_CODE, false)
        // set baseDomain="" => sử dụng mặc định là Product
        it.putExtra(KeyIntentConstantsNFC.CHANGE_BASE_URL_NFC, "")
    }
```

```
  }

  private fun navigateToScanNfc(json: JSONObject): Intent {
     return Intent(this, VnptScanNFCActivity::class.java).also {
        /**
         * Truyền access token chứa bearer
         */
        it.putExtra(KeyIntentConstantsNFC.ACCESS_TOKEN,
json.getString("access_token"))
        /**
         * Truyền token id
         */
        it.putExtra(KeyIntentConstantsNFC.TOKEN_ID, json.getString("token_id"))
        /**
         * Truyền token key
         */
        it.putExtra(KeyIntentConstantsNFC.TOKEN_KEY,
json.getString("token_key"))
        /**
         * điều chỉnh ngôn ngữ tiếng việt
         *    - vi: tiếng việt
         *    - en: tiếng anh
         */
        it.putExtra(KeyIntentConstantsNFC.LANGUAGE_NFC,
SDKEnumNFC.LanguageEnum.VIETNAMESE.value)
        /**
         * hiển thị màn hình hướng dẫn + hiển thị nút bỏ qua hướng dẫn
         * - mặc định luôn luôn hiển thị màn hình hướng dẫn
         *    - true: hiển thị nút bỏ qua
         *    - false: ko hiển thị nút bỏ qua
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_GOT_IT, true)
        /**
         * bật tính năng upload ảnh
         *    - true: bật tính năng
         *    - false: tắt tính năng
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_UPLOAD_IMAGE, true)
        /**
         * bật tính năng get Postcode
         *    - true: bật tính năng
         *    - false: tắt tính năng
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_MAPPING_ADDRESS, true)
        /**
         * bật tính năng xác thực chip
         *    - true: bật tính năng
         *    - false: tắt tính năng
         */
        it.putExtra(KeyIntentConstantsNFC.IS_ENABLE_VERIFY_CHIP, true)
        /**
```

```
         * truyền các giá trị đọc thẻ
         *    - nếu không truyền gì mặc định sẽ đọc tất cả (MRZ,Verify
Document,Image Avatar)
         *    - giá trị truyền vào là 1 mảng int: nếu muốn đọc giá trị nào sẽ
truyền
         *      giá trị đó vào mảng
         * eg: chỉ đọc thông tin MRZ
         *    intArrayOf(SDKEnumNFC.ReadingNFCTags.MRZInfo.value)
         */
        it.putExtra(
           KeyIntentConstantsNFC.READING_TAG_NFC,
           intArrayOf(
              SDKEnumNFC.ReadingNFCTags.MRZInfo.value,
              SDKEnumNFC.ReadingNFCTags.VerifyDocumentInfo.value,
              SDKEnumNFC.ReadingNFCTags.ImageAvatarInfo.value
           )
        )
        /**
         * truyền giá trị bật quét QRCode
         *    - true: tắt quét QRCode
         *    - false: bật quét QRCode
         */
        it.putExtra(KeyIntentConstantsNFC.IS_TURN_OFF_QR_CODE, true)
        // set baseDomain="" => sử dụng mặc định là Product
        it.putExtra(KeyIntentConstantsNFC.CHANGE_BASE_URL_NFC, "")
        // truyền id định danh căn cước công dân
        it.putExtra(KeyIntentConstantsNFC.ID_NUMBER_CARD,
json.getString("card_id"))
        // truyền ngày sinh ghi trên căn cước công dân
        it.putExtra(KeyIntentConstantsNFC.BIRTHDAY_CARD,
json.getString("card_dob"))
        // truyền ngày hết hạn căn cước công dân
        it.putExtra(KeyIntentConstantsNFC.EXPIRED_CARD,
json.getString("card_expire_date"))
     }
  }

  private fun parseJsonFromArgs(call: MethodCall): JSONObject {
     return try {
        @Suppress("UNCHECKED_CAST")
        (JSONObject(call.arguments as Map<String, Any>))
     } catch (e: Exception) {
        JSONObject(mapOf<String, Any>())
     }
  }

  /**
   * put value to [JSONObject] with null-safety
   */
  private fun JSONObject.putSafe(key: String, value: String?) {
     value?.let { put(key, it) }
```

```
  }
}
```

II. Xây dựng cầu nối

1. Xây dựng file cầu nối chung

- Tạo file **sdk_ekyc_nfc.dart** để làm cầu nối tương tác giữa **Flutter** và **iOS**, **Android**

```dart
import 'dart:convert';

import 'package:flutter/foundation.dart';
import 'package:flutter/services.dart';

class SDKEkycNfc {
 static final SDKEkycNfc _singleton = SDKEkycNfc._internal();

 static SDKEkycNfc get instance {
   return _singleton;
 }

 SDKEkycNfc._internal();

 Future<Map<String, dynamic>> startScanNfc({
   required String cardId,
   required String cardDob,
   required String cardExpireDate,
 }) async {
   try {
     final result = await Channels.channel.invokeMethodOnMobile(
       "navigateToScanNfc",
       {
         "access_token": "<ACCESS_TOKEN> (including bearer)",
         "token_id": "<TOKEN_ID>",
         "token_key": "<TOKEN_KEY>",
         "card_id": cardId.trim(),
         "card_dob": cardDob.trim(),
         "card_expire_date": cardExpireDate.trim(),
       },
     );

     final Map<String, dynamic> json = jsonDecode(result);

     return json.isEmpty ? {} : json;
   } on PlatformException catch (e) {
     return {"error": e.message ?? ''};
```

```
    }
  }

  Future<Map<String, dynamic>> startNfcQrCode() async {
    try {
      final result = await Channels.channel.invokeMethodOnMobile(
        'navigateToNfcQrCode',
        {
          "access_token": "<ACCESS_TOKEN> (including bearer)",
          "token_id": "<TOKEN_ID>",
          "token_key": "<TOKEN_KEY>",
        },
      );

      final Map<String, dynamic> json = jsonDecode(result);

      return json.isEmpty ? {} : json;
    } on PlatformException catch (e) {
      return {"error": e.message ?? ''};
    }
  }
}

extension MethodChannelMobile on MethodChannel {
  Future<T?> invokeMethodOnMobile<T>(String method, [dynamic arguments]) {
    if (kIsWeb) {
      return Future.value(null);
    }

    return invokeMethod(method, arguments);
  }
}

/// Native channels.
class Channels {
  static const MethodChannel channel =
      MethodChannel('flutter.sdk.ekyc/integrate');
}
```

2. Ví dụ

- Ví dụ sử dụng file cầu nối **sdk_ekyc_nfc.dart** như sau:

```
// QR -> NFC
final res = await SDKEkycNfc.instance.startNfcQrCode();
if (res.containsKey('error')) {
 // xử lý trường hợp lỗi
} else {
 // xử lý trường hợp thành công với dữ liệu `res`
}

//NFC
final res = await SDKEkycNfc.instance.startScanNfc(
 cardId: _textIdController.text,
 cardDob: _textDobController.text,
 cardExpireDate: _textExpireController.text,
);
if (res.containsKey('error')) {
 // xử lý trường hợp lỗi
} else {
 // xử lý trường hợp thành công với dữ liệu
 `res`
}
```