

TÀI LIỆU HƯỚNG DẪN TÍCH HỢP SDK REACT NATIVE EKYC-NFC

I. Tích hợp SDK vào iOS

1. Yêu cầu

- Yêu cầu một dự án **React Native**
- Mở thư mục **ios** trong dự án React Native và tích hợp theo các bước tài liệu sau:

 [VNPT] Hướng dẫn tích hợp SDK_IOS_ICNFCCardReader_v3.0.4

2. Thêm cầu nối

- Mở dự án iOS bằng Xcode và thêm tệp header (EkycNfcBridgeModule.h) và implementation (EkycNfcBridgeModule.m)

```
#ifndef EkycNfcBridgeModule_h
#define EkycNfcBridgeModule_h
#import <React/RCTBridgeModule.h>
#import "ICNFCCardReader/ICNFCCardReader.h"
@interface EkycNfcBridgeModule: NSObject <RCTBridgeModule,
ICMainNFCReaderDelegate>
@property(nonatomic, copy) RCTPromiseResolveBlock _resolve;
@property(nonatomic, copy) RCTPromiseRejectBlock _reject;
@end
#endif /* EkycNfcBridgeModule_h */
```

- Ở tệp implementation:

```
#import <Foundation/Foundation.h>
#import "EkycNfcBridgeModule.h"
#import "ICNFCCardReader/ICNFCCardReader.h"
@implementation EkycNfcBridgeModule
// To export a module named RCTCalendarModule
RCT_EXPORT_MODULE(EkycNfcBridge);
RCT_EXPORT_METHOD(navigateToNfcQrCode:(NSString *)json
resolve:(RCTPromiseResolveBlock)resolve rejecter:(RCTPromiseRejectBlock)reject)
{

    self._resolve = resolve;
    self._reject = reject;

    //....

};
```

```

RCT_EXPORT_METHOD(navigateToScanNfc:(NSString *)json
resolve:(RCTPromiseResolveBlock)resolve rejecter:(RCTPromiseRejectBlock)reject)
{

    self._resolve = resolve;
    self._reject = reject;

    //.....

};

@end

```

- Trong tệp mẫu cần chú ý đến các hàm RCT_EXPORT_METHOD, đây là các method định nghĩa tương ứng với hàm bên code react (navigateToScanNfc, navigateToNfcQrCode) có thể tạo nhiều hơn
 - navigateToNfcQrCode: mở SDK quét Qr và thực hiện quét chip
 - navigateToScanNfc: mở SDK truyền thông tin đầu vào và thực hiện quét chip
- Các phương thức con lại (nhận kết quả qua Delegate) tương tự tích hợp với dự án Native iOS (đề cập ở phần 1)

II. Tích hợp SDK vào Android

1. Yêu cầu

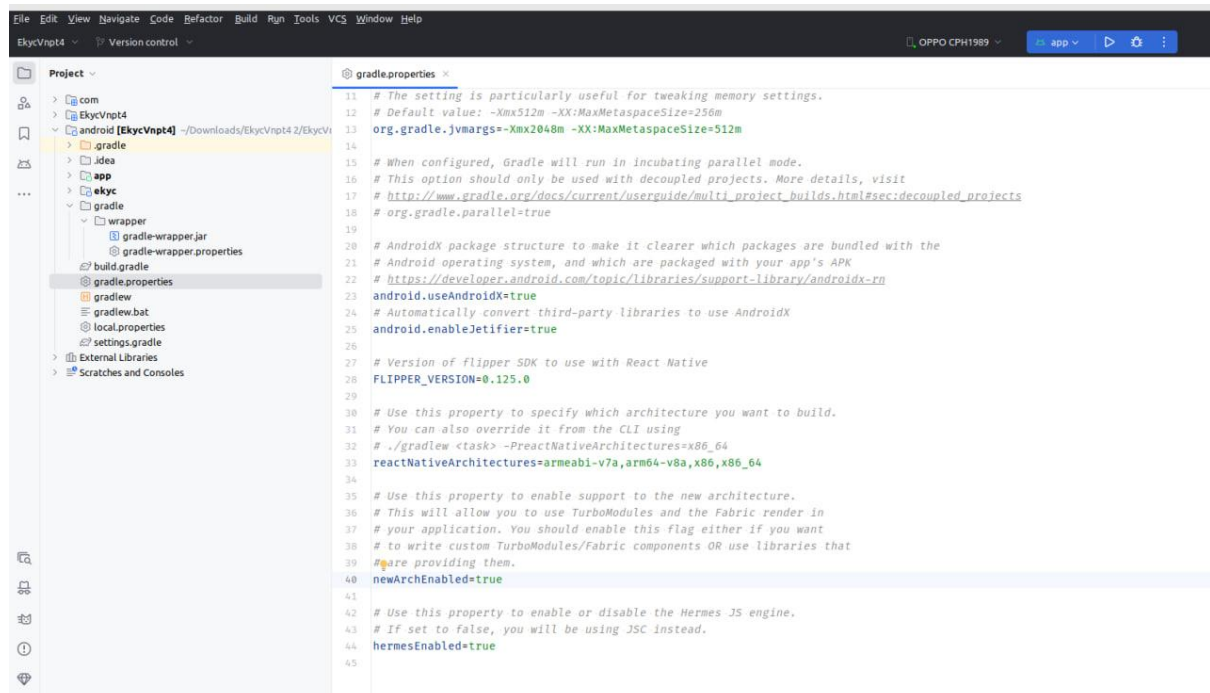
- Yêu cầu đã có sẵn một dự án React Native
- Mở thư mục android trong dự án React Native và tích hợp theo các bước theo tài liệu sau:

[VNPT] Hướng_dẫn_tích_hợp_SDK_ANDROID_ICNFFCCardReader_v...

- Nếu sử dụng React Native > v0.70 thì sẽ config như sau:

```
android {  
    ...  
    ndkVersion "23.1.7779620"
```

- Sử dụng ndk version 23.1.7779620 trong build.gradle (app)



- Enable newArchEnabled=true trong gradle.properties

2. Thêm cầu nối

- Tạo file cầu nối React Native (**EkycNfcBridgeModule.java**) kế thừa từ **ReactContextBaseJavaModule** như sau:

```
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;
```

```

import android.nfc.NfcAdapter;
import android.nfc.NfcManager;
import android.text.TextUtils;

import androidx.annotation.NonNull;

import com.facebook.react.bridge.ActivityEventListener;
import com.facebook.react.bridge.BaseActivityEventListener;
import com.facebook.react.bridge.Promise;
import com.facebook.react.bridge.ReactApplicationContext;
import com.facebook.react.bridge.ReactContextBaseJavaModule;
import com.facebook.react.bridge.ReactMethod;
import com.google.gson.JsonObject;
import com.google.gson.JsonParser;
import com.vnptit.nfc.activity.VnptScanNFCActivity;
import com.vnptit.nfc.utils.KeyIntentConstantsNFC;
import com.vnptit.nfc.utils.KeyResultConstantsNFC;
import com.vnptit.nfc.utils.SDKEnumNFC;

public class EkycNfcBridgeModule extends ReactContextBaseJavaModule {

    private static final int EKYC_REQUEST_CODE = 100;
    private static final String EKYC_REJECT_CODE = "69";
    private Promise mEkycPromise;

    public EkycNfcBridgeModule(ReactApplicationContext reactContext) {
        super(reactContext);

        final ActivityEventListener activityEventListener = new
BaseActivityEventListener() {
            @SuppressWarnings("DanglingJavadoc")
            @Override
            public void onActivityResult(Activity activity, int requestCode, int
resultCode, Intent data) {
                if (requestCode == EKYC_REQUEST_CODE && resultCode ==
Activity.RESULT_OK) {
                    if (data != null && mEkycPromise != null) {
                        /**
                         * đường dẫn ảnh mặt trước trong thẻ chip lưu trong cache
                         * {@Link KeyResultConstantsNFC#IMAGE_AVATAR_CARD_NFC}
                         */
                        final String avatarPath =
data.getStringExtra(KeyResultConstantsNFC.IMAGE_AVATAR_CARD_NFC);

                        /**
                         * chuỗi thông tin của SDK
                         * {@Link KeyResultConstantsNFC#CLIENT_SESSION_RESULT}
                         */

```

```

        final String clientSession =
data.getStringExtra(KeyResultConstantsNFC.CLIENT_SESSION_RESULT);

        /**
         * kết quả NFC
         * {@Link KeyResultConstantsNFC#LOG_NFC}
         */
        final String logNFC =
data.getStringExtra(KeyResultConstantsNFC.LOG_NFC);

        /**
         * mã hash avatar
         * {@Link KeyResultConstantsNFC#HASH_AVATAR}
         */
        final String hashAvatar =
data.getStringExtra(KeyResultConstantsNFC.HASH_AVATAR);

        /**
         * chuỗi json string chứa thông tin post code của quê quán
         * {@Link
KeyResultConstantsNFC#POST_CODE_ORIGINAL_LOCATION_RESULT}
         */
        final String postCodeOriginalLocation =
data.getStringExtra(KeyResultConstantsNFC.POST_CODE_ORIGINAL_LOCATION_RESULT
);

        /**
         * chuỗi json string chứa thông tin post code của nơi
thường trú
         * {@Link
KeyResultConstantsNFC#POST_CODE_RECENT_LOCATION_RESULT}
         */
        final String postCodeRecentLocation =
data.getStringExtra(KeyResultConstantsNFC.POST_CODE_RECENT_LOCATION_RESULT);

        /**
         * time scan nfc
         * {@Link KeyResultConstantsNFC#TIME_SCAN_NFC}
         */
        final String timeScanNfc =
data.getStringExtra(KeyResultConstantsNFC.TIME_SCAN_NFC);

        /**
         * kết quả check chip căn cước công dân
         * {@Link KeyResultConstantsNFC#CHECK_AUTH_CHIP_RESULT}
         */
        final String checkAuthChipResult =
data.getStringExtra(KeyResultConstantsNFC.CHECK_AUTH_CHIP_RESULT);

```

```

        /**
         * kết quả quét QRCode cần cước công dân
         * {@Link KeyResultConstantsNFC#QR_CODE_RESULT_NFC}
         */
        final String qrCodeResult =
data.getStringExtra(KeyResultConstantsNFC.QR_CODE_RESULT_NFC);

        final JsonObject json = new JsonObject();
        putSafe(json, KeyResultConstantsNFC.IMAGE_AVATAR_CARD_NFC,
avatarPath);
        putSafe(json, KeyResultConstantsNFC.CLIENT_SESSION_RESULT,
clientSession);
        putSafe(json, KeyResultConstantsNFC.LOG_NFC, logNFC);
        putSafe(json, KeyResultConstantsNFC.HASH_AVATAR,
hashAvatar);
        putSafe(
            json,
KeyResultConstantsNFC.POST_CODE_ORIGINAL_LOCATION_RESULT,
            postCodeOriginalLocation
        );
        putSafe(
            json,
KeyResultConstantsNFC.POST_CODE_RECENT_LOCATION_RESULT,
            postCodeRecentLocation
        );
        putSafe(json, KeyResultConstantsNFC.TIME_SCAN_NFC,
timeScanNfc);
        putSafe(json, KeyResultConstantsNFC.CHECK_AUTH_CHIP_RESULT,
checkAuthChipResult);
        putSafe(json, KeyResultConstantsNFC.QR_CODE_RESULT_NFC,
qrCodeResult);
        mEkycPromise.resolve(json.toString());
    }

    mEkycPromise = null;
}
}
};
reactContext.addActivityEventListener(activityEventListener);
}

private void putSafe(final JsonObject json, final String key, final String
value) {
    if (!TextUtils.isEmpty(value)) {
        json.addProperty(key, value);
    }
}

```

```

    }
}

@NonNull
@Override
public String getName() {
    return "EkycNfcBridge";
}

private boolean isDeviceSupportedNfc(Activity activity) {
    final NfcAdapter adapter = ((NfcManager)
activity.getSystemService(Context.NFC_SERVICE)).getDefaultAdapter();
    return adapter != null && adapter.isEnabled();
}

@ReactMethod
private void navigateToNfcQrCode(final String args, final Promise promise)
{
    final Activity activity = getCurrentActivity();
    if (activity == null) return;

    mEkycPromise = promise;
    if (!isDeviceSupportedNfc(activity)) {
        mEkycPromise.reject(EKYC_REJECT_CODE, "Thiết bị không hỗ trợ NFC");
        return;
    }

    final JsonObject json = JsonParser.parseString(args).getAsJsonObject();

    final Intent intent = new Intent(activity, VnptScanNFCActivity.class);
    /*
     * Truyền access token chứa bearer
     */
    intent.putExtra(KeyIntentConstantsNFC.ACCESS_TOKEN,
json.get("access_token").getString());
    /*
     * Truyền token id
     */
    intent.putExtra(KeyIntentConstantsNFC.TOKEN_ID,
json.get("token_id").getString());
    /*
     * Truyền token key
     */
    intent.putExtra(KeyIntentConstantsNFC.TOKEN_KEY,
json.get("token_key").getString());
    /*
     * điều chỉnh ngôn ngữ tiếng việt
     * - ví: tiếng việt

```

```

        * - en: tiếng anh
    */
    intent.putExtra(KeyIntentConstantsNFC.LANGUAGE_NFC,
SDKEnumNFC.LanguageEnum.VIETNAMESE.getValue());
    /*
    * 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
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_ENABLE_GOT_IT, true);
    /*
    * bật tính năng upload ảnh
    * - true: bật tính năng
    * - false: tắt tính năng
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_ENABLE_UPLOAD_IMAGE, true);
    /*
    * bật tính năng get Postcode
    * - true: bật tính năng
    * - false: tắt tính năng
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_ENABLE_MAPPING_ADDRESS, true);
    /*
    * bật tính năng xác thực chip
    * - true: bật tính năng
    * - false: tắt tính năng
    */
    intent.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
    * new int[]{SDKEnumNFC.ReadingNFCTags.MRZInfo.getValue()}
    */
    intent.putExtra(
        KeyIntentConstantsNFC.READING_TAG_NFC,
        new int[]{
            SDKEnumNFC.ReadingNFCTags.MRZInfo.getValue(),
            SDKEnumNFC.ReadingNFCTags.VerifyDocumentInfo.getValue(),
            SDKEnumNFC.ReadingNFCTags.ImageAvatarInfo.getValue()
        }
    );
    /*

```



```

    * truyền giá trị bật quét QRCode
    *   - true: tắt quét QRCode
    *   - false: bật quét QRCode
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_TURN_OFF_QR_CODE, false);
    // set baseDomain="" => sử dụng mặc định là Product
    intent.putExtra(KeyIntentConstantsNFC.CHANGE_BASE_URL_NFC, "");

    activity.startActivityForResult(intent, EKYC_REQUEST_CODE);
}

@ReactMethod
private void navigateToScanNfc(final String args, final Promise promise) {
    final Activity activity = getCurrentActivity();
    if (activity == null) return;

    mEkycPromise = promise;
    if (!isDeviceSupportedNfc(activity)) {
        mEkycPromise.reject(EKYC_REJECT_CODE, "Thiết bị không hỗ trợ NFC");
        return;
    }

    final JSONObject json = JsonParser.parseString(args).getAsJsonObject();

    final Intent intent = new Intent(activity, VnptScanNFCActivity.class);
    /*
     * Truyền access token chứa bearer
     */
    intent.putExtra(KeyIntentConstantsNFC.ACCESS_TOKEN,
        json.get("access_token").getString());
    /*
     * Truyền token id
     */
    intent.putExtra(KeyIntentConstantsNFC.TOKEN_ID,
        json.get("token_id").getString());
    /*
     * Truyền token key
     */
    intent.putExtra(KeyIntentConstantsNFC.TOKEN_KEY,
        json.get("token_key").getString());
    /*
     * điều chỉnh ngôn ngữ tiếng việt
     *   - vi: tiếng việt
     *   - en: tiếng anh
     */
    intent.putExtra(KeyIntentConstantsNFC.LANGUAGE_NFC,
        SDKEnumNFC.LanguageEnum.VIETNAMESE.getValue());
    /*

```

```

    * 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
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_ENABLE_GOT_IT, true);
    /*
    * bật tính năng upload ảnh
    *   - true: bật tính năng
    *   - false: tắt tính năng
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_ENABLE_UPLOAD_IMAGE, true);
    /*
    * bật tính năng get Postcode
    *   - true: bật tính năng
    *   - false: tắt tính năng
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_ENABLE_MAPPING_ADDRESS, true);
    /*
    * bật tính năng xác thực chip
    *   - true: bật tính năng
    *   - false: tắt tính năng
    */
    intent.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
    *   new int[]{SDKEnumNFC.ReadingNFCTags.MRZInfo.getValue()}
    */
    intent.putExtra(
        KeyIntentConstantsNFC.READING_TAG_NFC,
        new int[]{
            SDKEnumNFC.ReadingNFCTags.MRZInfo.getValue(),
            SDKEnumNFC.ReadingNFCTags.VerifyDocumentInfo.getValue(),
            SDKEnumNFC.ReadingNFCTags.ImageAvatarInfo.getValue()
        }
    );
    /*
    * truyền giá trị bật quét QRCode
    *   - true: tắt quét QRCode
    *   - false: bật quét QRCode
    */
    intent.putExtra(KeyIntentConstantsNFC.IS_TURN_OFF_QR_CODE, true);

```

```

        // set baseDomain="" => sử dụng mặc định là Product
        intent.putExtra(KeyIntentConstantsNFC.CHANGE_BASE_URL_NFC, "");
        // truyền id định danh căn cước công dân
        intent.putExtra(KeyIntentConstantsNFC.ID_NUMBER_CARD,
        json.get("card_id").getString());
        // truyền ngày sinh ghi trên căn cước công dân
        intent.putExtra(KeyIntentConstantsNFC.BIRTHDAY_CARD,
        json.get("card_dob").getString());
        // truyền ngày hết hạn căn cước công dân
        intent.putExtra(KeyIntentConstantsNFC.EXPIRED_CARD,
        json.get("card_expire_date").getString());

        activity.startActivityForResult(intent, EKYC_REQUEST_CODE);
    }
}

```

- Tạo một file kế thừa từ ReactPackage (**EkycNfcBridgeReactPackage.java**) khai báo cầu nối React Native (**EkycNfcBridgeModule.java**) vừa khởi tạo trong native module (hàm **createNativeModules**) như sau:

```

import androidx.annotation.NonNull;

import com.facebook.react.ReactPackage;
import com.facebook.react.bridge.NativeModule;
import com.facebook.react.bridge.ReactApplicationContext;
import com.facebook.react.uimanager.ViewManager;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class EkycNfcBridgeReactPackage implements ReactPackage {
    @NonNull
    @Override
    public List<NativeModule> createNativeModules(@NonNull
    ReactApplicationContext reactApplicationContext) {
        List<NativeModule> modules = new ArrayList<>();
        modules.add(new EkycNfcBridgeModule(reactApplicationContext));
        return modules;
    }

    @NonNull
    @Override
    public List<ViewManager> createViewManagers(@NonNull
    ReactApplicationContext reactApplicationContext) {
        return Collections.emptyList();
    }
}

```

```
}  
}
```

- Thêm package **EkycNfcBridgeReactPackage** vừa tạo vào list packages của **ReactNativeHost** trong file **MainApplication.java** như sau:

```
import android.app.Application;  
import com.facebook.react.PackageList;  
import com.facebook.react.ReactApplication;  
import com.facebook.react.ReactNativeHost;  
import com.facebook.react.ReactPackage;  
import com.facebook.react.defaults.DefaultNewArchitectureEntryPoint;  
import com.facebook.react.defaults.DefaultReactNativeHost;  
import com.facebook.soloader.SoLoader;  
  
import java.util.List;  
  
public class MainApplication extends Application implements ReactApplication  
{  
  
    private final ReactNativeHost mReactNativeHost =  
        new DefaultReactNativeHost(this) {  
            @Override  
            public boolean getUseDeveloperSupport() {  
                return BuildConfig.DEBUG;  
            }  
  
            @Override  
            protected List<ReactPackage> getPackages() {  
                @SuppressWarnings("UnnecessaryLocalVariable")  
                List<ReactPackage> packages = new PackageList(this).getPackages();  
                // Packages that cannot be autolinked yet can be added manually  
                here, for example:  
                packages.add(new EkycNfcBridgeReactPackage());  
                return packages;  
            }  
  
            @Override  
            protected String getJSMainModuleName() {  
                return "index";  
            }  
  
            @Override  
            protected boolean isNewArchEnabled() {  
                return BuildConfig.IS_NEW_ARCHITECTURE_ENABLED;  
            }  
        }  
}
```

```

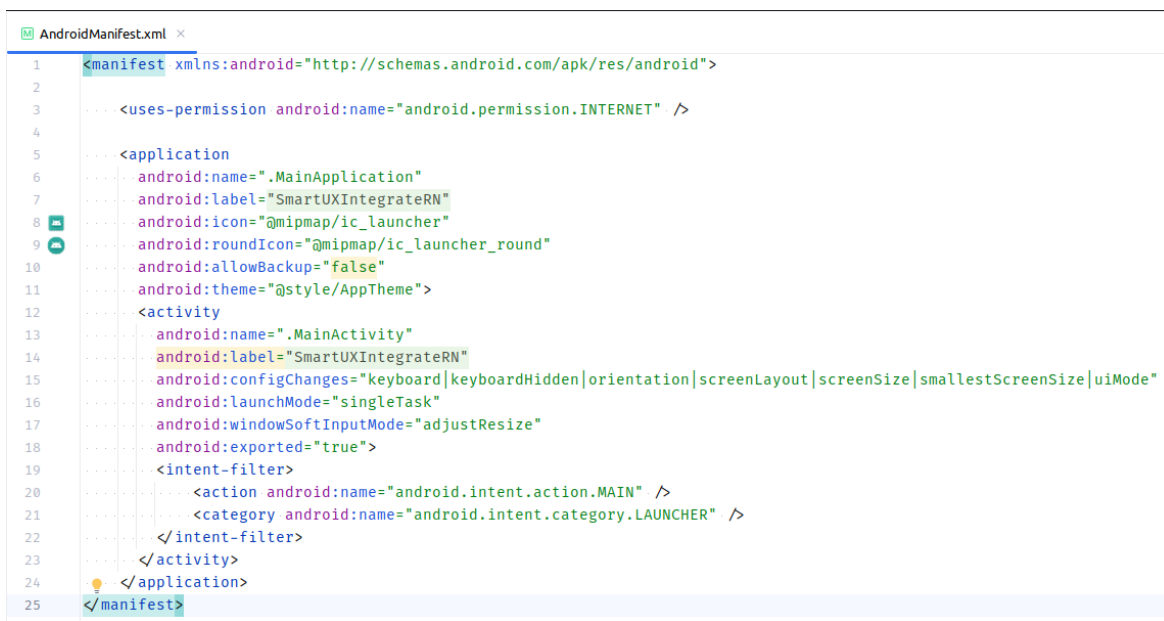
    @Override
    protected Boolean isHermesEnabled() {
        return BuildConfig.IS_HERMES_ENABLED;
    }
};

@Override
public ReactNativeHost getReactNativeHost() {
    return mReactNativeHost;
}

@Override
public void onCreate() {
    super.onCreate();
    SoLoader.init(this, /* native exopackage */ false);
    if (BuildConfig.IS_NEW_ARCHITECTURE_ENABLED) {
        // If you opted-in for the New Architecture, we load the native entry
        // point for this app.
        DefaultNewArchitectureEntryPoint.load();
    }
    ReactNativeFlipper.initializeFlipper(this,
    getReactNativeHost().getReactInstanceManager());
}
}
}

```

- Khai báo **MainApplication** trong file **AndroidManifest.xml** nếu chưa khai báo như sau:



```

1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2
3     <uses-permission android:name="android.permission.INTERNET" />
4
5     <application
6         android:name=".MainApplication"
7         android:label="SmartUXIntegrateRN"
8         android:icon="@mipmap/ic_launcher"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:allowBackup="false"
11        android:theme="@style/AppTheme">
12         <activity
13             android:name=".MainActivity"
14             android:label="SmartUXIntegrateRN"
15             android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|screenSize|smallestScreenSize|uiMode"
16             android:launchMode="singleTask"
17             android:windowSoftInputMode="adjustResize"
18             android:exported="true">
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21                 <category android:name="android.intent.category.LAUNCHER" />
22             </intent-filter>
23         </activity>
24     </application>
25 </manifest>

```

III. Xây dựng cầu nối

1. Xây dựng các file cầu nối

- Tạo file **SDKEkycNfc.js** để làm cầu nối tương tác giữa **React Native** và **iOS, Android**

```
import {NativeModules} from "react-native";

const {EkycNfcBridge} = NativeModules;
const SDKEkycNfc = {};

SDKEkycNfc.navigateToScanNfc = async function (cardId, cardDob,
cardExpireDate) {
  try {
    const json = JSON.stringify({
      access_token: '<ACCESS_TOKEN> (including bearer)',
      token_id: '<TOKEN_ID>',
      token_key: '<TOKEN_KEY>',
      card_id: cardId,
      card_dob: cardDob,
      card_expire_date: cardExpireDate,
    });
    return await EkycNfcBridge.navigateToScanNfc(json)
  } catch (e) {
    return {
      'error': e.message
    }
  }
};

SDKEkycNfc.navigateToNfcQrCode = async function () {
  try {
    const json = JSON.stringify({
      access_token: '<ACCESS_TOKEN> (including bearer)',
      token_id: '<TOKEN_ID>',
      token_key: '<TOKEN_KEY>',
    });
    return await EkycNfcBridge.navigateToNfcQrCode(json)
  } catch (e) {
    return {
      'error': e.message
    }
  }
};

export default SDKEkycNfc;
```

2. Ví dụ

```
// QR -> NFC
const response = await SDKEkycNfc.navigateToNfcQrCode();
if (response.error) {
    // xử lý trường hợp lỗi
} else {
    // xử lý trường hợp thành công với dữ liệu `response`
}

// NFC
const response = await SDKEkycNfc.navigateToScanNfc(
info.cardId,
info.cardDob,
info.cardExpireDate
);
if (response.error) {
    // xử lý trường hợp lỗi
} else {
    // xử lý trường hợp thành công với dữ liệu `response`
}
```