**TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM**

**TỔNG CÔNG TY CÔNG NGHỆ THÔNG TIN KHU VỰC 3 – VNPT IT3**

TÀI LIỆU HƯỚNG DẪN TÍCH HỢP SDK IOS ID CARD SCANNER

Mã số:

Tài liệu hướng dẫn tích hợp SDK IOS

Phiên bản:

V.2.3

Ngày hiệu lực:

02/04/2021

**Họ và tên**

**Chức danh**

**Ngày, tháng**

**Ký tên**

**Soạn thảo**

Nguyễn Khắc Minh

Nhân viên

**Xem xét**

**Thẩm định**

**Phê chuẩn**

**CONFIDENTIAL | All Rights Reserved****Page 1**/**21**

LỊCH SỬ THAY ĐỔI

**STT**

**Phiên bản**

**Người thực hiện**

**Người phê duyệt**

**Ngày hiệu lực**

**Nội dung thay đổi**

1

V1.0

Nguyễn Duy Hùng

Cao Thanh Hà

17/04/2019

Tạo mới

2

V1.1

Nguyễn Khắc Minh

02/04/2022

Cập nhật

**Nội dung thay đổi**

- Thay đổi nội dung phần**: Luồng chỉ So sánh khuôn mặt – compare, Luồng chỉ Đăng ký khuôn mặt – addFace**

**MỤC LỤC**

**Contents**

I.Tổng quan ......................................................................................................................... 4

**• Mục đích** ......................................................................................................................... 4

**• Phạm vi** ........................................................................................................................... 4

**• Tài liệu liên quan** ........................................................................................................... 4

**• Giải thích từ ngữ và các chữ viết tắt** ............................................................................ 4

II.Tích hợp thư viện SDK vào dự án .................................................................................. 5

**• Yêu cầu** ........................................................................................................................... 5

**• Thêm thư viện đi kèm SDK** .......................................................................................... 5

**• Cách sử dụng** .................................................................................................................. 5

**III.** **Các thuộc tính của CameraViewController .................................................................. 7** **IV.** **Dữ liệu SDK trả về sau khi hoàn thành ....................................................................... 12**

**1.** **Dữ liệu được trả về** ...................................................................................................... 12

V.Các luồng ........................................................................................................................ 13

**• Luồng đầy đủ – full** ...................................................................................................... 13

**• Luồng chỉ OCR mặt trước – ocrFront** ....................................................................... 15

**• Luồng chỉ OCR mặt sau – ocrBack** ........................................................................... 16

**4.** Luồng chỉ OCR mặt trước và mặt sau – ocrFrontAndBack **................................... 17**

**• Luồng chỉ So sánh khuôn mặt – compare** ................................................................. 18

**• Luồng chỉ Đăng ký khuôn mặt – addFace** ................................................................. 19

**• Luồng Xác thực khuôn mặt – verifyFace** .................................................................. 20

**• Code mẫu** ...................................................................................................................... 21

I. Tổng quan

1. Mục đích

 Tài liệu này nhằm mục đích mô tả chi tiết xây dựng ứng dụng iOS tích hợp SDK ID Card Scanner

2. Phạm vi

 Tài liệu này thuộc giai đoạn bàn giao sản phẩm đến khách hàng xây dựng ứng dụng iOS tích hợp SDK

**3. Tài liệu liên quan**

 N/A

4. Giải thích từ ngữ và các chữ viết tắt

**Thuật ngữ**

**Giải thích**

II. Tích hợp thư viện SDK vào dự án

1. Yêu cầu

• iOS 8.0 trở lên

• Swift 4 trở lên

2. Thêm thư viện đi kèm SDK

• Thêm thư mục có chứa các thư viện sau vào dự án:

• Alamofire.framework

• Base58Swift.framework

• BigInt.framework

• ObjectMapper.framework

• SipHash.framework

• SWMessages.framework

• TensorFlowLite.framework

• TensorFlowLiteC.framework

• Cài đặt cho các thư viện (Project  Target  Genneral)

3. Nhập các thông tin đi kèm

• Thực hiện thêm các giá trị để xác thực tài khoản như dưới đây

• Thực hiện thêm ở AppDelegate

• Dưới đây là các thông tin, mã token khách hàng cung cấp để sử dụng SDK

• SaveData.shared().SDTokenKey = ""

• SaveData.shared().SDTokenId = ""

• SaveData.shared().SDAuthorization = ""

• URL máy chủ của khách hàng, nếu không sẽ dùng mặc định của VNPT

• SaveData.shared().BaseUrl = ""

4. Cách sử dụng

• import FinalSDK trong dự án của bạn

• kế thừa CameraDelegate

• Khởi tạo CameraViewController và cài đặt các thuộc tính

• Dữ liệu sau khi được xử lý sẽ được trả về ứng dụng của bạn thông qua phương thức

getResult(), Dữ liệu được mô tả chi tiết ở phần Dữ liệu được trả về

1.

//

2.

// ViewController.swift

3.

// DemoUsingSDKSwift

4.

//

5.

// Created by Jessy on 12/30/19.

6.

// Copyright © 2019 Depper Company. All rights reserved.

7.

//

8.

9.

**import** UIKit

10.

**import** FinalSDK

11.

12.

**class** ViewController: UIViewController, CameraDelegate {

13.

14.

15.

16.

override func viewDidLoad() {

17.

**super**.viewDidLoad()

18.

// Do any additional setup after loading the view.

19.

}

20.

21.

22.

// method init, setup properties and open CameraViewController

23.

func openCameraToOCR() {

24.

25.

let objCamera = CameraRouter.createModule() as! CameraViewController

26.

objCamera.cameraDelegate = self

27.

28.

objCamera.isVersion = .Normal

29.

objCamera.followType = .full

30.

objCamera.isShowResult = **true**

31.

objCamera.isShowHelp = **true**

32.

objCamera.isType = .cmt

33.

objCamera.stepNow = .stepFront

34.

35.

objCamera.modalPresentationStyle = .fullScreen

36.

self.showDetailViewController(objCamera, sender: nil)

37.

}

38.

}

39.

40.

// MARK: - CameraDelegate

41.

extension ViewController {

42.

// method delegate when SDK return value after OCR

43.

func getResult() {

44.

//

45.

}

46.

}

III. Các thuộc tính của CameraViewController

var isValidatePostcode: Bool

Giá trị này xác định việc có xác thực số ID với mã tỉnh thành, quận huyện, xã phường tương ứng hay không. Kiểm tra quy tắc của số ID

Mặc định là **false**

var isShowHelp: Bool

Giá trị này xác định việc có hiển thị màn hình trợ giúp hay không.

Mặc định là **true**

var isShowResult: Bool

Giá trị này xác định việc có hiển thị màn hình thông tin kết quả sau khi hoàn thành hay không.

Mặc định là **false**

var isShowTrademark: Bool

Giá trị này xác định việc có hiển thị ảnh thương hiệu (logo) ở cuối màn hình hay không.

Mặc định là **true**

var isAddFace: Bool

Giá trị này xác định việc có tự động đăng ký khuôn mặt và thông tin lên hệ thống hay không. Điều kiện để thực hiện bao gồm:

• Nhận dạng ký tự quang học (OCR) thành công

• Phiên bản chức năng sử dụng là phiên bản nâng cao

• Ảnh chụp chân dung là người thật, giấy tờ thật

• Ảnh mặt trong giấy tờ khớp với ảnh chụp chân dung

• Ảnh chụp chân dung không bị che

Mặc định là **false**

var isCheckLivenessCard: Bool

Giá trị này xác định việc có thực hiện việc kiểm tra giấy tờ được chụp trực tiếp hay không.

Mặc định là **false**

var isCheckMaskFace: Bool

Giá trị này xác định có thực hiện việc kiểm tra ảnh chân dung được chụp có bị che mặt hay không.

Mặc định là **true**

var isVersion: VersionSdk

Giá trị này xác định phiên bản chức năng khi sử dụng Máy ảnh

(CameraViewController), có 2 phiên bản lựa chọn

• normal – phiên bản cơ bản. Chụp ảnh chân dung một góc độ

• pro – phiên bản nâng cao. Chụp ảnh chân dung ba góc độ

Mặc định là **pro**

var isType: TypeDocument

Giá trị này xác định kiểu giấy tờ để sử dụng. Bao gồm 4 loại:

• cmt – Chứng minh thư nhân dân, Căn cước công dân

• hochieu – Hộ chiếu

• banglaixe – Giấy phép lái xe

• cmtquandoi – Chứng minh thư quân đội

Mặc định là **cmt**

var flowType: FlowType

Giá trị này xác định việc luồng làm việc khi OCR giấy tờ và ảnh chân dung, bao gồm:

• ocrFront – Chỉ thực hiện việc Chụp ảnh và OCR mặt trước giấy tờ

• ocrBack – Chỉ thực hiện việc Chụp ảnh và OCR mặt sau giấy tờ

• ocrFrontAndBack – Chỉ thực hiện việc Chụp ảnh và OCR mặt trước và mặt sau giấy tờ (không dùng cho Hộ chiếu, bằng lái xe)

• full – Thực hiện việc Chụp ảnh mặt trước  mặt sau giấy tờ  chụp ảnh chân dung. Thực hiện OCR, Compare, CheckLiveness, CheckMaskFace, CheckLivenessCard, AddFace

• compare – Chỉ thực hiện việc Chụp ảnh chân dung, sau đó thực hiện so sánh mặt ở giấy tờ với ảnh chân dung

• addFace – Thực hiện việc chụp ảnh chân dung sau đó thực hiện Đăng ký thông tin và ảnh chân dung lên máy chủ

• verifyFace – Thực hiện việc chụp ảnh chân dung để xác thực với thông tin trên máy chủ

Mặc định là **full**

var languageApplication: String

Giá trị này xác định ngôn ngữ được sử dụng trong SDK, có 2 giá trị:

• vi – Tiếng Việt

• en – Tiếng Anh

Mặc định là **vi**

var inputIDVerifyFace: String

Giá trị được nhập vào để thực hiện chức năng xác thực khuôn mặt

Mặc định là **""**

var idTypeVerifyFace: String

Giá trị được nhập vào để thực hiện chức năng xác thực khuôn mặt, các giá trị:

• CARD\_ID – đối với Chứng minh thư nhân dân, thẻ căn cước

• PASSPORT\_ID – đối với Hộ chiếu

• DRIVER\_LICENSE\_ID – đối với Bằng lái xe

• MILITARY\_ID – đối với Chứng minh thư quân đội

Mặc định là **"CARD\_ID"**

var challengeCode: String

Giá trị này dùng để đảm bảo mỗi yêu cầu (request) từ phía khách hàng sẽ không bị thay đổi. Sau mỗi request, dữ liệu trả về sẽ bao gồm giá trị challengeCode.

Mặc định là **"11111"**

var unitCustomer: String

Giá trị này xác định tên đơn vị, tên khách hàng sử dụng SDK eKYC, được truyền vào khi khởi tạo đối tượng CameraViewController

Mặc định **""**

var resourceCustomer: String

Giá trị này xác định tên nguồn đơn vị, tên khách hàng sử dụng SDK eKYC, được truyền vào khi khởi tạo đối tượng CameraViewController

Mặc định **""**

var titleCustomer: String

Giá trị này xác định Thông tin bổ sung theo từng khách hàng, được truyền vào khi khởi tạo đối tượng CameraViewController

Mặc định **""**

var hashFrontCompareFace: String

Giá trị này là mã ảnh chứa mặt người, được truyền vào để so sánh với ảnh chân dung sau khi chụp ảnh từ SDK.

Mặc định **""**

var arrayItemPostCode: Array<PostCodeType>

Giá trị này là mảng chứa các giá trị để kiểm tra mã số trên giấy giờ và mã tỉnh thành, quận huyện, xã phường tương ứng.

Các giá trị: **[**BirthPlace, RecentLocation, OriginLocation, IssuePlace**]**

Mặc định **""**

var objectAddFace: Dictionary<String, Any>

Giá trị này là đối tượng chứa dữ liệu, được truyền vào để thực hiện Đăng ký khuôn mặt (AddFace), theo định dạng <key, value> sau:

**extra\_info** = {

"user":"tester",

"hash\_front": "<hashFront>",

"hash\_rear": "<hashBack",

"date": "<dd/MM/yyyy>"

}

**objectAddFace** = {

"card\_id": "<card\_id>",

"passport\_id": "<passport\_id>",

"driver\_license\_id": "<driver\_license\_id>",

"military\_id": "<military\_id>",

"police\_id": "<police\_id>",

"other\_id": "<other\_id>",

"fullname": "<fullname>",

"dob": "<dob>",

"gender": "<gender>",

"address": "<address>",

"hometown": "<hometown>",

"nationality": "<nationality>",

"ipfs": "<ipfs>",

"title": "<title>",

"other\_type": "<other\_type>",

"extra\_info": **extra\_info**,

}

Mặc định **{}**

var logoTrademarkName: String

Giá trị này là ảnh thương hiệu được hiển thị cuối mỗi màn hình trong SDK.

Mặc định

var stepNow: ProgessStep

Giá trị này xác định giao diện chụp ảnh giấy tờ hoặc mặt người, bao gồm các giá trị:

• stepFront – Giao diện chụp ảnh giấy tờ mặt trước

• stepBack – Giao diện chụp ảnh giấy tờ mặt sau

• stepFace – Giao diện chụp ảnh mặt người

Mặc định .**stepFront**

var isShowRotateCamera: Bool

Giá trị này xác định việc hiển thị nút bấm chức năng xoay máy ảnh trước sang máy ảnh sau.

Mặc định **false**

var isWantRotateCameraBack: Bool

Giá trị này xác định việc sử dụng máy ảnh phía trước hoặc phía sau khi chụp ảnh mặt người.

Mặc định **false**

var isCustomize: Bool

Giá trị này xác định việc chỉnh sửa giao diện theo ý khách hàng hay không.

Mặc định **false**

var isCompare: Bool

Giá trị này xác định việc có thực hiện so sánh khuôn mặt chân dung trong giấy tờ và mặt người

Mặc định **true**

var isCheckLivenessFace: Bool

Giá trị này xác định việc có thực hiện kiểm tra ảnh mặt người được chụp trực tiếp hay không.

Mặc định **true**

var isCheckMaskFace: Bool

Giá trị này xác định việc có thực hiện việc kiểm tra che mặt khi chụp ảnh mặt người hay không.

Mặc định **true**

var buttonTitleColor: UIColor

Giá trị này xác định màu nút bấm ở màn hình Hiển thị ảnh.

Mặc định **#142730**

var buttonReTakeColor: UIColor

Giá trị này xác định màu nút bấm “Chụp lại” ở màn hình Hiển thị ảnh

Mặc định **#18D696**

var buttonBackgroundColor: UIColor

Giá trị này xác định màu nền ảnh hướng dẫn, màu nền nút bấm ở màn hình Hiển thị ảnh.

Mặc định **#18D696**

var isSkipVoiceVideo: Bool

Giá trị này xác định việc có thực hiện việc chạy hết audio và video hướng dẫn rồi mới thực hiện việc chụp ảnh chân dung hay không.

Mặc định **false**

IV. Dữ liệu SDK trả về sau khi hoàn thành

1. Dữ liệu được trả về

• func getResult() {

• print("getResult():")

• print("JsonInfo: \(SaveData.shared().jsonInfo)")

• print("JsonCompareFace: \(SaveData.shared().jsonCompareFace)")

• print("JsonLiveness: \(SaveData.shared().jsonLiveness)")

• print("JsonVerifyFace: \(SaveData.shared().jsonVerifyFace)")

• print("JsonAddFace: \(SaveData.shared().jsonAddFace)")

• }

var jsonInfo: String

Giá trị sau khi bóc tách thông tin dữ liệu giấy tờ.

var jsonCompareFace: String

Giá trị sau khi so sánh ảnh mặt chân dung trên giấy tờ và ảnh chụp chân dung.

var jsonLivenessFace: String

Giá trị kết quả việc kiểm tra ảnh chân dung có được chụp trực tiếp hay không (Người thật).

var jsonVerifyFace: String

Giá trị kết quả xác thực khuôn mặt với số giấy tờ nhập vào.

var jsonCheckMask: String

Giá trị kết quả kiểm tra ảnh chân dung có bị che, đeo khẩu trang hay không.

var jsonCheckLivenessFrontCard: String

Giá trị kết quả kiểm tra ảnh giấy tờ được chụp trực tiếp hay không.

var jsonCheckLivenessBackCard: String

Giá trị kết quả kiểm tra ảnh giấy tờ được chụp trực tiếp hay không.

var jsonAddFace: String

Giá trị kết quả việc đăng ký khuôn mặt lên hệ thống.

var jsonSearchFace: String

Giá trị tìm kiếm khuôn mặt được trả về từ hệ thống.

var jsonIssuePlace: String

Giá trị postcode được trả về dựa vào địa chỉ Nơi cấp.

var jsonOriginLocation: String

Giá trị postcode được trả về dựa vào địa chỉ Nguyên quán.

var jsonRecentLocation: String

Giá trị postcode được trả về dựa vào địa chỉ Nơi ở hiện tại.

var jsonBirthPlace: String

Giá trị postcode được trả về dựa vào địa chỉ Nơi sinh.

var imageFront: UIImage

Ảnh mặt trước được trả về sau khi chụp.

var imageBack: UIImage

Ảnh mặt sau được trả về sau khi chụp.

var imageFace: UIImage

Ảnh chân dung được trả về sau khi chụp.

var imageNear: UIImage

Ảnh chân dung gần được trả về sau khi chụp.

var imageFar: UIImage

Ảnh chân dung xa được trả về sau khi chụp.

var hashImageFront: UIImage

Mã ảnh **mặt trước** giấy tờ sau khi tải ảnh lên máy chủ.

var hashImageBack: UIImage

Mã ảnh **mặt sau** giấy tờ sau khi tải ảnh lên máy chủ.

var hashImageFace: UIImage

Mã ảnh **chân dung** sau khi tải ảnh lên máy chủ.

var hashImageNear: UIImage

Mã ảnh **chân dung gần** sau khi tải ảnh lên máy chủ.

var hashImageFar: UIImage

Mã ảnh **chân dung xa** sau khi tải ảnh lên máy chủ.

var json3DLiveness: UIImage

Giá trị kết quả việc kiểm tra ảnh **chân dung 3D** có được chụp trực tiếp hay không (Người thật).

V. Các luồng

1. Luồng đầy đủ – full

• Bước 1: Thực hiện việc chụp giấy tờ mặt trước  Bước 2: Thực hiện việc chụp giấy tờ mặt sau.

• Bước 3: Chụp ảnh chân dung chủ giấy tờ.

• Bước 4: Sau khi thực hiện OCR để lấy thông tin.

• Lưu ý:

• Đối với Hộ chiếu và Bằng lái xe, sẽ chỉ có bước 1, bước 3 và bước 4. Bỏ qua chụp mặt sau của giấy tờ

• Sử dụng phiên bản cơ bản (normal) hoặc nâng cao (pro) sẽ hiển thị giao diện màn hình chụp ảnh chân dung khác nhau

Thông tin sẽ được trả về ứng dụng của bạn thông qua phương thức delegate getResult(), trong trường hợp chọn isShowResult = **true** thì màn hình Thông tin cá nhân sẽ được hiển thị và ngược lại

Chụp ảnh giấy tờ mặt trước

Chụp ảnh giấy tờ mặt sau

Chụp ảnh chân dung chủ giấy tờ

Hiển thị kết quả trả về

Code mẫu

// thực hiện việc OCR luồng đầy đủ: Chụp giấy tờ và ảnh chân dung

func showNormal() {

let objCamera = CameraRouter.createModule() as! CameraViewController

objCamera.isVersion = .Normal // hoặc .Pro

objCamera.flowType = .full

objCamera.isType = .cmt // hoặc .hochieu, .banglaixe, .cmtquandoi, cmtganchip

objCamera.cameraDelegate = self

objCamera.stepNow = .stepFront

// objCamera.arrayItemPostCode = [] as Array<PostCodeType>

objCamera.unitCustomer = ""

objCamera.challengeCode = ""

objCamera.isShowResult = false

objCamera.isShowHelp = true

objCamera.isShowTrademark = true

objCamera.isCheckLivenessCard = true

objCamera.isCheckMaskFace = true

objCamera.isAddFace = true

objCamera.isCheckLivenessFace = true

objCamera.languageApplication = self.isVietnamese ? "vi" : "en"

objCamera.logoTrademarkName = ""

objCamera.modalPresentationStyle = .fullScreen

self.present(objCamera, animated: true, completion: nil)

}

2. Luồng chỉ OCR mặt trước – ocrFront

Bước 1: Thực hiện việc chụp giấy tờ mặt trước

Bước 2: Sau khi thực hiện OCR để lấy thông tin mặt trước.

Thông tin sẽ được trả về ứng dụng của bạn thông qua phương thức delegate getResult(), trong trường hợp chọn isShowResult = **true** thì màn hình Thông tin cá nhân sẽ được hiển thị và ngược lại

Bước 1

Bước 2

Code mẫu

// thực hiện việc OCR chỉ mặt trước giấy tờ

func ocrFront() {

let objCamera = CameraRouter.createModule() as! CameraViewController

objCamera.isVersion = .Pro

objCamera.flowType = .ocrFront

objCamera.isType = self.selectedType

objCamera.cameraDelegate = self

objCamera.stepNow = .stepFront

objCamera.unitCustomer = ""

objCamera.challengeCode = ""

objCamera.isShowResult = false

objCamera.isShowHelp = true

objCamera.isShowTrademark = true

objCamera.isCheckLivenessCard = true

objCamera.isCheckMaskFace = true

objCamera.isAddFace = true

objCamera.languageApplication = self.isVietnamese ? "vi" : "en"

objCamera.logoTrademarkName = ""

objCamera.modalPresentationStyle = .fullScreen

self.showDetailViewController(objCamera, sender: nil)

}

3. Luồng chỉ OCR mặt sau – ocrBack

Bước 1: Thực hiện việc chụp giấy tờ mặt sau

Bước 2: Sau khi thực hiện OCR để lấy thông tin mặt sau.

Thông tin sẽ được trả về ứng dụng của bạn thông qua phương thức delegate getResult(), trong trường hợp chọn isShowResult = **true** thì màn hình Thông tin cá nhân sẽ được hiển thị và ngược lại

Bước 1

Bước 2

Code mẫu

// thực hiện việc OCR chỉ mặt sau giấy tờ

func ocrBack() {

let objCamera = CameraRouter.createModule() as! CameraViewController

objCamera.isVersion = .Pro

objCamera.flowType = .ocrBack

objCamera.isType = self.selectedType

objCamera.cameraDelegate = self

objCamera.stepNow = .stepBack

objCamera.unitCustomer = ""

objCamera.challengeCode = ""

objCamera.isShowResult = false

objCamera.isShowHelp = true

objCamera.isShowTrademark = true

objCamera.isCheckLivenessCard = true

objCamera.isCheckMaskFace = true

objCamera.isAddFace = true

objCamera.languageApplication = self.isVietnamese ? "vi" : "en"

objCamera.logoTrademarkName = ""

objCamera.modalPresentationStyle = .fullScreen

self.showDetailViewController(objCamera, sender: nil)

}

**4.** Luồng chỉ OCR mặt trước và mặt sau – ocrFrontAndBack

Bước 1: Thực hiện việc chụp giấy tờ mặt trước

Bước 2: Thực hiện việc chụp giấy tờ mặt sau

Bước 3: Sau khi thực hiện OCR để lấy thông tin giấy tờ.

Thông tin sẽ được trả về ứng dụng của bạn thông qua phương thức delegate getResult(), trong trường hợp chọn isShowResult = **true** thì màn hình Thông tin cá nhân sẽ được hiển thị và ngược lại

Bước 1

Bước 2

Bước 3

Code mẫu

// thực hiện việc OCR chỉ mặt trước và mặt sau giấy tờ

func ocrFrontAndBack() {

let objCamera = CameraRouter.createModule() as! CameraViewController

objCamera.isVersion = .Pro

objCamera.flowType = .ocrFrontAndBack

objCamera.isType = self.selectedType

objCamera.cameraDelegate = self

objCamera.unitCustomer = ""

objCamera.challengeCode = ""

objCamera.isShowResult = false

objCamera.isShowHelp = true

objCamera.isShowTrademark = true

objCamera.isCheckLivenessCard = true

objCamera.isCheckMaskFace = true

objCamera.isAddFace = true

objCamera.languageApplication = self.isVietnamese ? "vi" : "en"

objCamera.logoTrademarkName = ""

objCamera.modalPresentationStyle = .fullScreen

self.showDetailViewController(objCamera, sender: nil)

}

**5.** Luồng chỉ So sánh khuôn mặt – compare

Bước 1: Thực hiện việc truyền mã ảnh (mã hash) giấy tờ có chứa ảnh chân dung

Bước 2: Thực hiện việc chụp ảnh chân dung chủ giấy tờ

Bước 3: Thực hiện so sánh ảnh chân dung và ảnh mặt có trong giấy tờ.

Thông tin sẽ được trả về ứng dụng của bạn thông qua phương thức delegate getResult()

Code mẫu

// thực hiện việc Chụp ảnh chân dung sau đó sẽ so sánh với mã hash ảnh giấy tờ chứa mặt người

func showCompare(hashFront: String) {

let objCamera = CameraRouter.createModule() as! CameraViewController

objCamera.isVersion = .Normal // hoặc .Pro cho việc quay mặt 3 hướng

objCamera.flowType = .compare

objCamera.isType = self.selectedType

objCamera.cameraDelegate = self

objCamera.stepNow = .stepFace

objCamera.unitCustomer = ""

objCamera.resourceCustomer = ""

objCamera.challengeCode = ""

objCamera.hashFrontCompareFace = hashFront

objCamera.isShowResult = false

objCamera.isShowHelp = true

objCamera.isShowTrademark = true

objCamera.isCheckMaskFace = true

objCamera.isAddFace = false

objCamera.isCheckLivenessFace = true

objCamera.languageApplication = self.isVietnamese ? "vi" : "en"

objCamera.logoTrademarkName = ""

objCamera.modalPresentationStyle = .fullScreen

self.present(objCamera, animated: true, completion: nil)

}

**6.** Luồng chỉ Đăng ký khuôn mặt – addFace

Bước 1: Thực hiện việc nhập thông tin cá nhân với dữ liệu theo định dạng objectAddFace

Bước 2: Thực hiện việc chụp ảnh chân dung

Bước 3: Thực hiện việc đăng ký thông tin cá nhân kèm ảnh.

Thông tin sẽ được trả về ứng dụng của bạn thông qua phương thức delegate getResult()

Code mẫu

// thực hiện việc Chụp ảnh chân dung sau đó sẽ đăng ký thông tin lên hệ thống

// riêng phần objCamera.customerInformationAddFace phải đảm bảo đủ các key cho object như dưới đây (tham khảo phương thức: convertDetailModelToObject)

/\*

[

"card\_id": card\_id,

"passport\_id": passport\_id,

"driver\_license\_id": driver\_license\_id,

"military\_id": military\_id,

"police\_id": police\_id,

"other\_id": other\_id,

"fullname": fullname,

"dob": dob,

"gender": gender,

"address": address,

"hometown": hometown,

"nationality": nationality,

"ipfs": ipfs,

"title": title,

"other\_type": other\_type,

"extra\_info": extra\_info,

] as Dictionary<String, Any>

\*/

func callAddFaceCamera() {

let objCamera = CameraRouter.createModule() as! CameraViewController

objCamera.isVersion = .normal

objCamera.flowType = .addFace

objCamera.isType = self.selectedType

objCamera.cameraDelegate = self

objCamera.stepNow = .stepFace

objCamera.unitCustomer = ""

objCamera.resourceCustomer = ""

objCamera.challengeCode = ""

var customerInformation: Dictionary<String, Any>?

objCamera.customerInformationAddFace = customerInformation

objCamera.hashFrontCompareFace = hashFront

objCamera.isShowResult = false

objCamera.isShowHelp = true

objCamera.isShowTrademark = true

objCamera.isCheckMaskFace = true

objCamera.isAddFace = true

objCamera.isCheckLivenessFace = true

objCamera.languageApplication = self.isVietnamese ? "vi" : "en"

objCamera.logoTrademarkName = ""

objCamera.modalPresentationStyle = .fullScreen

self.present(objCamera, animated: true, completion: nil)

}

**7.** Luồng Xác thực khuôn mặt – verifyFace

Bước 1: Thực hiện nhập chuỗi số để thực hiện xác thực

Bước 2: Thực hiện việc chụp ảnh chân dung

Bước 3: Sau đó thực hiện xác thực thông tin và ảnh chân dung.

Thông tin sẽ được trả về ứng dụng của bạn thông qua phương thức delegate getResult()

Code mẫu

// thực hiện việc Chụp ảnh chân dung => mã hash, sau đó dùng ID nhập vào và mã hash để xác thực khuôn mặt

func verifyFaceNormal(verifyID: String) -> Void {

let objCamera = CameraRouter.createModule() as! CameraViewController

objCamera.cameraDelegate = self

objCamera.isVersion = .normal

objCamera.flowType = .verifyFace

objCamera.stepNow = .stepFace

objCamera.inputIDVerifyFace = verifyID

objCamera.idTypeVerifyFace = self.getTypeId(selectedType: self.selectedType)

objCamera.isType = self.selectedType

objCamera.isShowResult = false

objCamera.isShowHelp = true

objCamera.isShowTrademark = true

objCamera.isCheckMaskFace = true

objCamera.isAddFace = true

objCamera.isCheckLivenessFace = true

objCamera.languageApplication = self.isVietnamese ? "vi" : "en"

objCamera.logoTrademarkName = ""

objCamera.modalPresentationStyle = .fullScreen

self.showDetailViewController(objCamera, sender: nil)

}

8. Code mẫu

1. func openCameraViewController() {

2.

3. let objCamera = CameraRouter.createModule() as! CameraViewController

4.

• objCamera.isVersion = .Normal // lựa chọn phiên bản chức năng

• objCamera.followType = .full // lựa chọn luồng làm việc

• objCamera.stepNow = .stepFront // chọn bước chụp mặt trước giấy tờ để bắt đầu

• objCamera.cameraDelegate = self // set delegate để nhận dữ liệu ở hàm getResult() 9. objCamera.isType = self.cmt // chọn loại giấy tờ, ví dụ Chứng minh thư

10.

• objCamera.isShowResult = false // không cho phép hiển thị màn hình kết quả

• objCamera.isShowHelp = true // cho phép hiển thị màn hình trợ giúp

• objCamera.isShowTrademark = true // cho phép hiển thị logo nhãn hiệu

• objCamera.isCheckLivenessCard = true // bật tính năng kiểm tra giấy tờ chụp thật

• objCamera.isCheckMaskFace = true // bật tính năng kiểm tra có che mặt

16.

• // tên ảnh logo hiển thị ở cuối mỗi màn hình

• objCamera.logoTrademarkName = "logo\_customer\_trademark"

• objCamera.buttonTitleColor = UIColor.init(red: 125/225.00, green: 95/225.00, blu e: 5/225.00, alpha: 1) // màu chữ nút bấm

• objCamera.buttonBackgroundColor = UIColor.init(red: 250/225.00, green: 185/225.0 0, blue: 15/225.00, alpha: 1) // màu nền nút bấm

31.

• objCamera.modalPresentationStyle = .fullScreen

• self.showDetailViewController(objCamera, sender: nil)

• }