**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**  
  
**---------------------------------------**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

ỨNG DỤNG MOBILE ĐIỂM DANH BẰNG  
NHẬN DIỆN KHUÔN MẶT

NGÀNH: Công Nghệ Thông Tin

CHUYÊN NGÀNH: Công Nghệ Thông Tin

**NGƯỜI HƯỚNG DẪN SINH VIÊN THỰC HIỆN:**

**ThS. Vũ Đình Long** **Lê Nguyên Khoa; MSSV: 1651120099**

**Phạm Kim Long; MSSV: 1651120101**

**Lê Minh Trung Nghĩa; MSSV: 1651120108**

TP. HỒ CHÍ MINH – Năm 2020

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**  
  
**---------------------------------------**



**Lê Nguyên Khoa**

**Phạm Kim Long**

**Lê Minh Trung Nghĩa**

**ỨNG DỤNG MOBILE ĐIỂM DANH BẰNG  
NHẬN DIỆN KHUÔN MẶT**

NGÀNH: Công Nghệ Thông Tin

CHUYÊN NGÀNH: Công Nghệ Thông Tin

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**NGƯỜI HƯỚNG DẪN**

ThS. Vũ Đình Long

TP. HỒ CHÍ MINH – Năm 2020

**Lời cam đoan**

Nhóm chúng em xin cam đoan những nội dung được trình bày trong đồ án tốt nghiệp “Ứng dụng Mobile điểm danh bằng nhận diện khuôn mặt” là công trình nghiên cứu của nhóm chúng em, không được sao chép từ bất kỳ Thực tập tốt nghiệp nào khác. Các thông tin được trình bày trong tài liệu đều do nhóm chúng em tìm hiểu và tổng hợp lại cũng như phần lập trình do tụi em tự thực hiện. Em xin được chịu trách nhiệm cho lời cam đoan này.

TP. Hồ Chí Minh, tháng 5 năm 2020

Sinh viên thực hiện

**Lê Nguyên Khoa**

**Phạm Kim Long**

**Lê Minh Trung Nghĩa**

**Mục lục**

[**Danh mục hình** 3](#_Toc42110088)

[**Danh mục bảng** 4](#_Toc42110089)

[**Danh mục các từ viết tắt** 5](#_Toc42110090)

[**Chương 1: Giới thiệu** 6](#_Toc42110091)

[**1. Lý do chọn đề tài** 6](#_Toc42110092)

[**2. Phương pháp nghiên cứu** 6](#_Toc42110093)

[**3. Project charter** 6](#_Toc42110094)

[**3.1. Tên dự án** 6](#_Toc42110095)

[**3.2. Mô tả dự án** 7](#_Toc42110096)

[**3.3. Mục tiêu của dự án** 7](#_Toc42110097)

[**3.4. Đối tượng sử dụng** 7](#_Toc42110098)

[**3.5. Sản phẩm bàn giao** 7](#_Toc42110099)

[**3.6. Thành viên và vai trò của từng thành viên trong nhóm** 7](#_Toc42110100)

[**3.7. Thời gian thực hiện dự án** 7](#_Toc42110101)

[**3.8. Phạm vi dự án** 8](#_Toc42110102)

[**4. Đặc tả yêu cầu** 8](#_Toc42110103)

[**5. Các công cụ hỗ trợ** 9](#_Toc42110104)

[**Chương 2: Cơ sở lý thuyết** 10](#_Toc42110105)

[**1. ASP .Net Core** 10](#_Toc42110106)

[**1.1. Khái niệm** 10](#_Toc42110107)

[**1.2. Sự khác nhau giữa .NET Core và .NET Framework** 11](#_Toc42110108)

[**1.3. Các đăc tính quan trọng của ASP.NET Core** 12](#_Toc42110109)

[**2. React Native** 19](#_Toc42110110)

[**2.1. React native là gì?** 19](#_Toc42110111)

[**2.2. Native app và Hybrid app** 19](#_Toc42110112)

[**2.3. Ưu nhược điểm của React Native là gì?** 21](#_Toc42110113)

[**2.4. Một số khái niệm React Native** 21](#_Toc42110114)

[**2.5. Công cụ phát triển React Native** 22](#_Toc42110115)

[**2.6. Một số thư viện thông dụng** 22](#_Toc42110116)

[**3. Azure Cognition Services** 25](#_Toc42110117)

[**3.1. Giới thiệu về Microsoft Cognitive Services** 25](#_Toc42110118)

[**3.2. Áp dụng Microsoft Cognitive Service** 29](#_Toc42110119)

[**4. Đặc thù của hệ thống** 36](#_Toc42110120)

[**Chương 3: Phân tích thiết kế** 37](#_Toc42110121)

[**1. Sơ đồ Use-case** 37](#_Toc42110122)

[**2. Class Diagram** 37](#_Toc42110123)

[**2.1. Mô tả thực thể** 37](#_Toc42110124)

[**2.2. Sơ đồ class** 40](#_Toc42110125)

[**3. Sequence Diagram** 42](#_Toc42110126)

[**3.1. Chức năng sinh viên** 42](#_Toc42110127)

[**3.2. Chức năng giảng viên** 43](#_Toc42110128)

[**4. Activity Diagram** 44](#_Toc42110129)

[**4.1. Chức năng sinh viên** 44](#_Toc42110130)

[**4.2. Chức năng giảng viên** 45](#_Toc42110131)

[**Chương 4: Triển khai** 46](#_Toc42110132)

[**1. Front-end** 46](#_Toc42110133)

[**1.1. Giao diện chung** 46](#_Toc42110134)

[**1.2. Giao diện sinh viên** 47](#_Toc42110135)

[**1.3. Giao diện giảng viên** 51](#_Toc42110136)

[**2. Back-end** 59](#_Toc42110137)

[**2.1. Chức năng chung** 59](#_Toc42110138)

[**2.2. Chức năng sinh viên** 59](#_Toc42110139)

[**2.3. Chức năng giảng viên** 66](#_Toc42110140)

[**Chương 5: Testing** 76](#_Toc42110141)

[**1. Front-end** 76](#_Toc42110142)

[**2. Back-end** 80](#_Toc42110143)

[**Chương 6: Kết luận và hướng phát triển** 82](#_Toc42110144)

[**1. Kết luận** 82](#_Toc42110145)

[**2. Hướng phát triển** 82](#_Toc42110146)

[**TÀI LIỆU THAM KHẢO** 83](#_Toc42110147)

# **Danh mục hình**

[Hình 2.1: Toolchain của .Net Core 11](#_Toc42110148)

[Hình 2.2: Sự khác nhau (ASP.NET Core in Action của Andrew Lock) 11](#_Toc42110149)

[Hình 2.3: Đa nền tảng 13](#_Toc42110150)

[Hình 2.4: Kiến trúc Microservices 14](#_Toc42110151)

[Hình 2.5: Mô hình của Containers 15](#_Toc42110152)

[Hình 2.6: Nuget Package 16](#_Toc42110153)

[Hình 2.7: ASP .Net Core hợp nhất MVC và Web API 16](#_Toc42110154)

[Hình 2.8 Angular và ASP .Net Core 17](#_Toc42110155)

[Hình 2.9: Sơ đồ UML về Class và Sequence cho mẫu thiết kế Dependency Injection 19](#_Toc42110156)

[Hình 2.10: Cách hoạt động của React Native 20](#_Toc42110157)

[Hình 2.11: Ví dụ về các biểu tượng trong thư viện React Native Vector Icons 23](#_Toc42110158)

[Hình 2.12: Ứng dụng React Native Image Picker 23](#_Toc42110159)

[Hình 2.13: Ví dụ minh hoạ việc xác định khuôn mặt từ ảnh 35](#_Toc42110160)

[Hình 2.14: Sơ đồ kiến trúc hệ thống 36](#_Toc42110161)

[Hình 3.1: Sơ đồ Use-case 37](#_Toc42110162)

[Hình 3.2: Sơ đồ class 41](#_Toc42110163)

[Hình 3.3: Sơ đồ tuần tự chức năng Đăng kí khuôn mặt 42](#_Toc42110164)

[Hình 3.4: Sơ đồ tuần tự chức năng Điểm danh 43](#_Toc42110165)

[Hình 3.5: Sơ đồ hoạt động chức năng Đăng kí khuôn mặt 44](#_Toc42110166)

[Hình 3.6: Sơ đồ hoạt động chức năng Điểm danh 45](#_Toc42110167)

[Hình 4.1: Giao diện bắt đầu 46](#_Toc42110168)

[Hình 4.2: Giao diện Đăng nhập 46](#_Toc42110169)

[Hình 4.3: Giao diện Trang chủ 47](#_Toc42110170)

[Hình 4.4: Giao diện Chi tiết thông báo chung 47](#_Toc42110171)

[Hình 4.5: Giao diện các chức năng 48](#_Toc42110172)

[Hình 4.6: Giao diện Thời khoá biểu 48](#_Toc42110173)

[Hình 4.7: Giao diện điểm danh theo môn 49](#_Toc42110174)

[Hình 4.8: Giao diện điểm danh theo môn 49](#_Toc42110175)

[Hình 4.9: Giao diện Cập nhật thông tin cá nhân 50](#_Toc42110176)

[Hình 4.10: Giao diện Cập nhật mật khẩu 50](#_Toc42110177)

[Hình 4.11: Giao diện Đăng kí khuôn mặt 51](#_Toc42110178)

[Hình 4.12: Giao diện Thông báo 51](#_Toc42110179)

[Hình 4.13: Giao diện các chức năng 52](#_Toc42110180)

[Hình 4.14: Giao diện Tạo thông báo 52](#_Toc42110181)

[Hình 4.15: Giao diện Danh sách thông báo 53](#_Toc42110182)

[Hình 4.16: Giao diện chi tiết thông báo 53](#_Toc42110183)

[Hình 4.17: Giao diện Thông tin cá nhân 54](#_Toc42110184)

[Hình 4.18: Giao diện Cập nhật mật khẩu 54](#_Toc42110185)

[Hình 4.19: Giao diện Xem danh sách sinh viên theo môn học 55](#_Toc42110186)

[Hình 4.20: Giao diện Xem thời khoá biểu 55](#_Toc42110187)

[Hình 4.21: Giao diện Danh sách điểm danh 56](#_Toc42110188)

[Hình 4.22: Giao diện Danh sách điểm danh 56](#_Toc42110189)

[Hình 4.23: Giao diện Điểm danh 57](#_Toc42110190)

[Hình 4.24: Giao diện khi mở camera Điểm danh 57](#_Toc42110191)

[Hình 4.25: Giao diện Cập nhật điểm danh 58](#_Toc42110192)

[Hình 4.26: Giao diện thông báo 58](#_Toc42110193)

[Hình 4.27: Giao diện thông báo kết quả điểm danh được gửi đến sinh viên 59](#_Toc42110194)

[Hình 5.1: Testing chức năng gửi thông báo 76](#_Toc42110195)

[Hình 5.2: Testing chức năng Đăng nhập 77](#_Toc42110196)

[Hình 5.3: Testing chức năng Điểm danh 78](#_Toc42110197)

[Hình 5.4: Testing chức năng Cập nhật mật khẩu 79](#_Toc42110198)

[Hình 5.5: Testing chức năng Điểm danh 80](#_Toc42110199)

[Hình 5.6: Testing chức năng Đăng kí khuôn mặt 81](#_Toc42110200)

# **Danh mục bảng**

[Bảng 1.1: Đặc tả yêu cầu 8](#_Toc42110201)

[Bảng 2.1: Vị trí của .NET trong mô hình phát triển .NET 10](#_Toc42110202)

[Bảng 2.2: So sánh sự khác nhau của .NET Framework và .NET Core 12](#_Toc42110203)

[Bảng 3.1: Chi tiết thực thể Module SinhVien 37](#_Toc42110204)

[Bảng 3.2: Chi tiết thực thể Module GiangVien 38](#_Toc42110205)

[Bảng 3.3: Chi tiết thực thể Module LopHoc 38](#_Toc42110206)

[Bảng 3.4: Chi tiết thực thể Module MonHoc 39](#_Toc42110207)

[Bảng 3.5: Chi tiết thực thể Module SinhVienTrongLopHoc 39](#_Toc42110208)

[Bảng 3.6: Chi tiết thực thể Module ChiTietNgayHoc 39](#_Toc42110209)

[Bảng 3.7: Chi tiết thực thể Module ChiTietDiemDanh 39](#_Toc42110210)

[Bảng 3.8: Chi tiết thực thể Module ThongBao 40](#_Toc42110211)

# **Danh mục các từ viết tắt**

**API:** Application Programing Interface

**MVC:** Model – View – Controller

**SQL:** Structure Query Languege

**IOT:** Internet of Things

**SDK:** Software Development Kit

**REST:** Representational State Transfer

**CRUD:** Create – Read – Update – Delete

# **Chương 1: Giới thiệu**

## **1. Lý do chọn đề tài**

- Với mục đích học tập, mong muốn tìm hiểu những nội dung mới, công nghệ mới, những chủ đề mới mẻ hơn để thực hiện nhằm có thêm những kiến thức không những chỉ là về lập trình mà là kể về những kĩ năng mềm khi thực hiện đồ án với phong cách mới mẻ, chuyên nghiệp hơn.

- Với mong muốn tạo ra 1 ứng dụng giúp các giảng viên có thể thực hiện việc điểm danh ở mỗi buổi học được nhanh chóng hơn và chính xác hơn. Tiết kiệm được nhiều thời gian của buổi học để có thể tập trung nhiều hơn về việc giảng dạy nội dung của buổi học khi mà công việc điểm danh sẽ được thực hiện chỉ trong 1 vài phút ngắn ngủi mà vẫn mang lại hiệu quả tuyệt đối cho buổi học.

- Mở rộng hơn là việc theo dõi thời khoá biểu cũng như thông tin điểm danh một cách nhanh chóng ở cả đối với giảng viên và sinh viên.

## **2. Phương pháp nghiên cứu**

- Tìm hiểu và nghiên cứu các tài liệu về ASP .Net Core, React Native, Azure Cognition Services cả về lí thuyết lẫn cách thức triển khai. Song với đó là củng cố hơn những kiến thức về cơ sở dữ liệu.

- Trong quá trình thực hiện dự án, cả nhóm thường họp lại với nhau cùng giải quyết vấn đề, bên cạnh đó là hỗ trợ lẫn nhau khi gặp khó trong quá trình triển khai.

- Tìm tòi, học hỏi lẫn nhau cũng như trên tìm kiếm các thông tin trên mạng thông qua công cụ tìm kiếm Google, kênh Youtube hay những trang hỏi đáp như Stack Overflow, …

- Sử dụng công cụ Microsoft Project để dễ dàng quản lí tiến độ dự án, phân công công việc rõ rãng để dự án có thể hoàn thành trong thời gian thực hiện dự án theo quy định.

## **3. Project charter**

### **3.1. Tên dự án**

**Ứng dụng Mobile điểm danh bằng nhận diện khuôn mặt.**

### **3.2. Mô tả dự án**

- Ứng dụng Mobile giúp giảng viên có thể điểm danh sinh viên trong lớp học bằng phương pháp nhận diện khuôn mặt, tránh học hộ, điểm danh hộ.

- Giúp sinh viên theo dõi được tình hình điểm danh của mình.

### **3.3. Mục tiêu của dự án**

- Web API (back-end) để xử lí và trả dữ liệu bao gồm quản lí thông tin, nhận diện khuôn mặt,...

- Mobile app (front-end) để giảng viên và sinh viên dễ dàng tương tác

### **3.4. Đối tượng sử dụng**

- Ứng dụng này dành cho sinh viên và giảng viên sử dụng.

### **3.5. Sản phẩm bàn giao**

- Project charter, tài liệu đặc tả, tài liệu phân tích thiết kế, WBS (Work Breakdown Structure), ứng dụng.

### **3.6. Thành viên và vai trò của từng thành viên trong nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thành viên** | **MSSV** | **Vai trò** |
| 1. | Lê Nguyên Khoa | 1651120099 | Nhóm trưởng, Back-end |
| 2. | Phạm Kim Long | 1651120101 | Thư kí, Back-end |
| 3. | Lê Minh Trung Nghĩa | 1651120108 | Front-end |

### **3.7. Thời gian thực hiện dự án**

|  |  |
| --- | --- |
| Thời gian bắt đầu dự án | 02/03/2020 |
| Thời gian thực hiện dự án | 8 tuần |

### **3.8. Phạm vi dự án**

- Tập trung chủ yếu và việc xây dụng ứng dụng với việc tiến hành xây dựng Back-end API với ASP .Net Core cùng với Face API từ Microsoft Cognition Services kết hợp với việc triển khai ứng dụng Mobile với React Native.

- Xây dụng một cơ sở dữ liệu dụng sẵn để kiểm thử và triển khai ứng dụng.

- Thực hiện tài liệu báo cáo.

## **4. Đặc tả yêu cầu**

Bảng 1.1: Đặc tả yêu cầu

| **STT** | **Chức năng** | **Mô tả** | **Độ khó** |
| --- | --- | --- | --- |
| 1 | Đăng nhập | Giảng viên và sinh viên đăng nhập bằng tài khoản và mật khẩu do nhà trường cung cấp với phần quyền khác nhau | - Dễ  - Phải có |
| 2 | Đăng kí khuôn mặt (Dành cho sinh viên) | Sinh viên đăng kí khuôn mặt của chính mình hệ thống có dữ liệu để đối chiếu để xác nhận việc điểm danh | - Khó  - Phải có |
| 3 | Điểm danh (Dành cho giảng viên) | Giảng viên chụp ảnh trực tiếp lớp học bằng điện thoại để tiến hành điểm danh | - Khó  - Phải có |
| 4 | Xem thông tin điểm danh của lớp học (Dành cho giáo viên) | Giảng viên có thể xem được danh sách lớp và tình hình điểm danh | - Trung bình  - Phải có |
| 5 | Xem thông tin điểm danh (Sinh viên) | Sinh viên có thể xem thông tin điểm danh của mình theo từng môn học, ngày học | - Trung bình  - Phải có |
| 6 | Xem thời khoá biểu | Giảng viên và sinh viên đều có thể xem thời khoá biểu theo các tuỳ chọn như xem theo tuần, xem theo môn | - Trung bình  - Phải có |
| 7 | Cập nhật thông tin | Người dùng có thể cập nhật thông tin cá nhân | - Dễ  - Phải có |
| 8 | Thông báo | Giảng viên có thể gửi thông báo cũng như những thông tin điểm danh theo từng ngày cho sinh viên theo lớp học | - Trung bình  - Có thể có hoặc không |

## **5. Các công cụ hỗ trợ**

- Back-end: Visual Studio, Microsoft SQL Server

- Front-end: Visual Code, Android Studio

- Rest API Client, Sourcetree, Microsoft Word, Microsoft Project

# **Chương 2: Cơ sở lý thuyết**

## **1. ASP .Net Core**

### **1.1. Khái niệm**

- ASP.NET Core là một nền tảng mã nguồn mở, cross-platform framework cho việc xây dựng và kết nối các ứng dụng web hiện đại chẳng hạn như ứng dụng web, Internet of Thing (IOT), Mobile Backend – Web API.

- ASP.NET Core cung cấp một kiến trúc để tối ưu hóa việc xây dựng các ứng dụng đám mây (microsoft azure) hoặc các ứng dụng độc lập. Chúng ta có thể phát triển ứng dụng ASP.NET Core đa nền tảng (Windows, Mac, Linux).

- Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework.

Bảng 2.1: Vị trí của .NET trong mô hình phát triển .NET

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Web Forms | MVC | Web API | ASP.NET Core MVC | |
| ASP.NET 4.6 (System Web) | | | ASP.Net Core 1.0 (Microsoft.AspNetCore) | |
| .Net Framework  Full .Net Framework for any scenario and Library support on Windows | | | | .NET Core  Modular libraries and runtime optimized for server and cloud workloads |

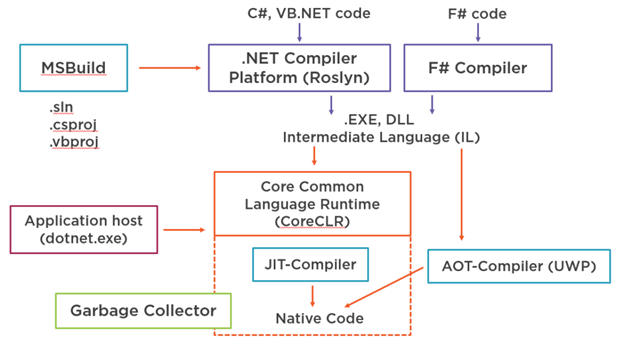
**Cách thức hoạt động của .NET Core**

- Về mặt ý tưởng nó có những điểm tương tự như trong .NET Framework. Tuy nhiên nó không liên quan đến .NET Framework.

- Thứ nhất, môi trường thực thi của .NET Core được gọi là CoreCLR. CoreCLR có thể chạy trên nhiều hệ điều hành. Hiện nay CoreCLR có thể hoạt động trên Windows, Linux và macOS. CoreCLR sử dụng một trình biên dịch trung gian tương tự JIT compiler của .NET Framework. Mặc dù tên gọi giống nhau nhưng JIT compilter của .NET Core không phải là JIT của .NET Framework. JIT của .NET Core có thể dịch mã IL sang mã máy của 3 nền tảng nó hỗ trợ.

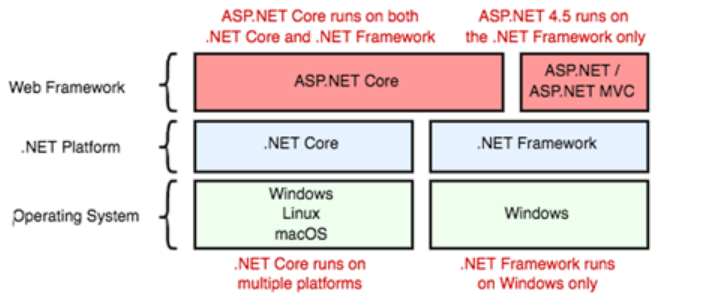
- Một sự khác biệt nữa về runtime nằm ở chỗ, CoreCLR và mã máy được tải và kích hoạt bởi một tiến trình khác, dotnet.exe, trong khi CLR được kiểm soát bởi hệ điều hành Windows. Với vai trò framework, .NET Core cung cấp khung sườn cho phát triển ứng dụng web (ASP.NET Core), phát triển ứng dựng desktop trên windows (Windows Forms và WPF, từ .NET Core 3.1). Về tính năng này, .NET Core và .NET Framework gần như tương đương nhau.

- Dưới đây là minh họa toolchain của .NET Core



Hình 2.1: Toolchain của .Net Core

### **1.2. Sự khác nhau giữa .NET Core và .NET Framework**

****

Hình 2.2: Sự khác nhau (ASP.NET Core in Action của Andrew Lock)

- .NET Core là một hệ thống độc lập với .NET Framework Khi bạn đã hiểu .NET Framework theo khía cạnh như trên, bạn sẽ cần hình dung tương tự về .NET Core. .NET Core cũng là một hệ thống trọn vẹn bao gồm nền tảng dịch mã trung gian và thực thi ứng dụng, các framework để phát triển các loại ứng dụng, và là hệ thống thư viện hỗ trợ. Hệ thống của .NET Core được xây dựng mới hoàn toàn và độc lập với .NET Framework.

Bảng 2.2: So sánh sự khác nhau của .NET Framework và .NET Core

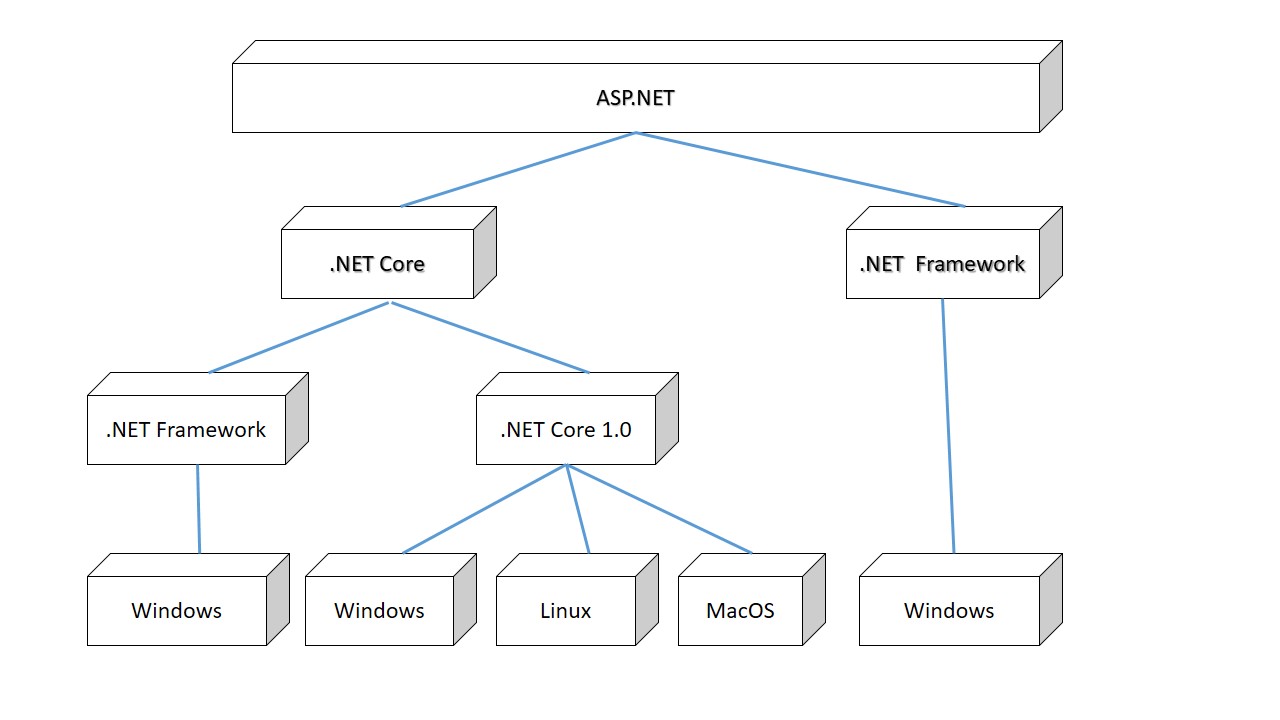
| **.NET Framework** | **.NET Core** |
| --- | --- |
| - Chỉ chạy trên windows  - Số lượng third – party lớn, chất lượng  - Docker, Microservices còn nhiều hạn chế  - Support Windows Form, WPF  - Chạy ổn định, cộng đông support lớn  - Số lượng công việc nhiều | - Đa nền tảng (Windows, Linux, MacOS)  - Cộng đồng còn non trẻ, số lượng  third – party ít  - Được sinh ra để dành cho Microservices, rất mạnh mẽ  - Windows form, WPF chỉ mới được thêm vào từ phiên bản 3.0 (9-2019)  - Là một công nghệ khá mới, chưa có tính ổn định  - Số lượng công việc không nhiều bằng .NET Framework |

### **1.3. Các đăc tính quan trọng của ASP.NET Core**

**1. Đa nền tảng**

- ASP.NET Core hỗ trợ ứng dụng của bạn (web/dịch vụ) chạy được trên nhiều nền tảng.

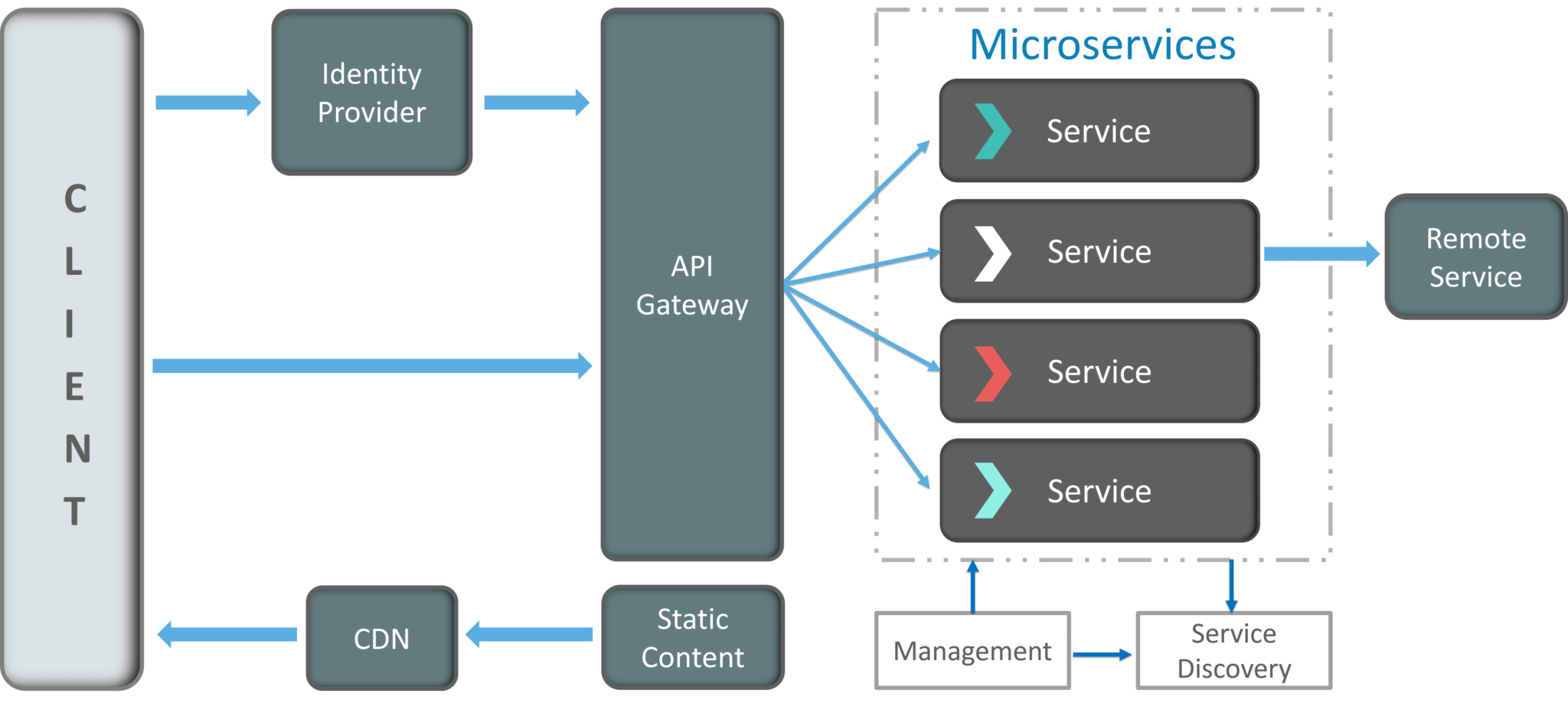
- .NET Core hỗ trợ các hệ điều hành được đề cập trước đây như là phát triển workstation của bạn. Visual Studio cung cấp một môi trường phát triển tích hợp (IDE) cho Windows và MacOS. Bạn cũng có thể sử dụng Visual Studio Code, chạy trên macOS, Linux và Windows. Visual Studio Code hỗ trợ .NET Core, bao gồm cả IntelliSense và debug. Hầu hết các IDE bên thứ ba, chẳng hạn như Sublime, Emacs và VI, làm việc với .NET Core. Các IDE bên thứ ba này nhận được trình soạn thảo IntelliSense sử dụng Omnisharp.



Hình 2.3: Đa nền tảng

**2. Sử dụng kiến trúc Microservices**

- Kiến trúc microservices cho phép kết hợp các công nghệ trên một ranh giới dịch vụ. Hỗn hợp công nghệ này cho phép sử dụng .NET Core cho các dịch vụ microservices mới hoạt động với các service hoặc micro services khác. Ví dụ, bạn có thể kết hợp các service hoặc micro services được phát triển với .NET Framework, Java, Ruby hoặc các công nghệ độc lập khác.

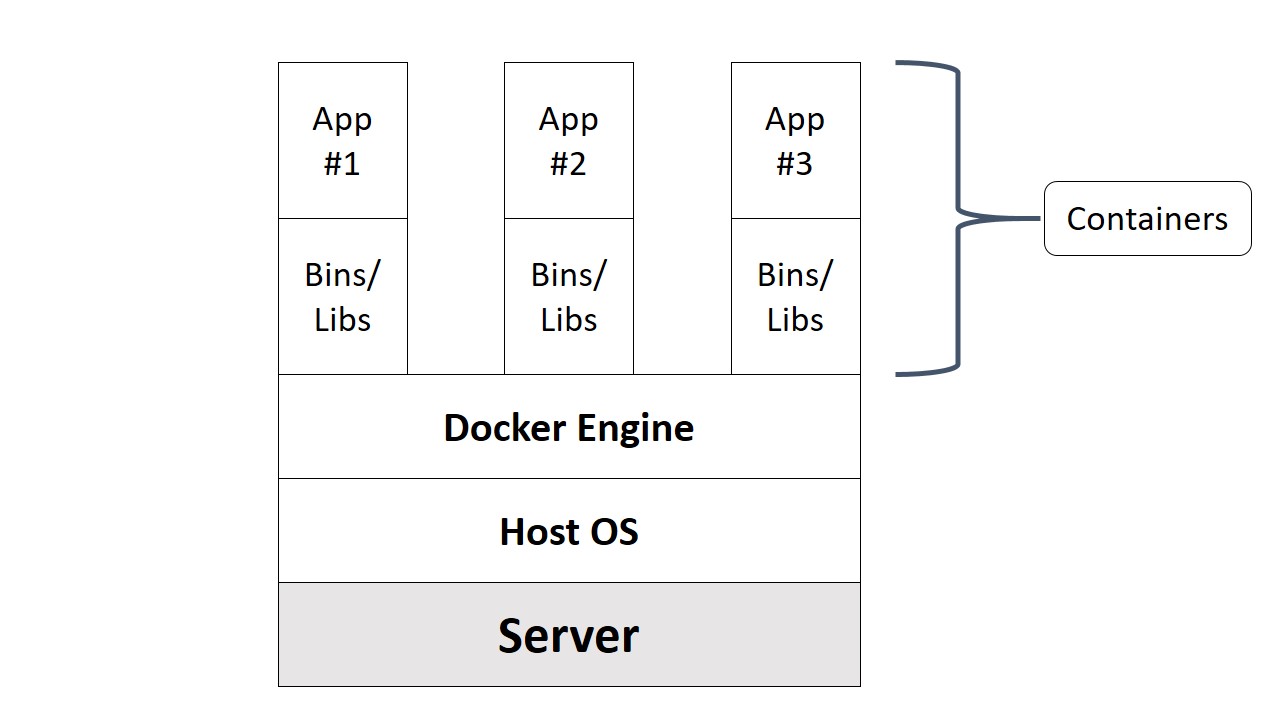


Hình 2.4: Kiến trúc Microservices

- Có rất nhiều nền tảng cơ sở hạ tầng có sẵn. Azure Service Fabric được thiết kế cho các hệ thống microservice lớn và phức tạp. Dịch vụ ứng dụng Azure là một lựa chọn tốt cho các dịch vụ microsoft đa quốc gia. Các lựa chọn thay thế Microservices dựa trên Docker phù hợp với bất kỳ phương pháp tiếp cận microservices nào, như được giải thích trong phần Containers. Tất cả các nền tảng này hỗ trợ .NET Core và làm cho chúng lý tưởng để lưu trữ các micro service của bạn.

**3. Sử dụng Docker containers.**

- Containers thường được sử dụng kết hợp với kiến trúc microservices. Các containers cũng có thể được sử dụng để chứa các ứng dụng hoặc dịch vụ web theo bất kỳ mẫu kiến trúc nào. .NET Framework có thể được sử dụng trên các containers trên Windows, nhưng tính mô đun và tính chất nhẹ của .NET Core làm cho nó trở thành một lựa chọn tốt hơn cho các containers. Khi tạo và triển khai một containers, kích thước image của nó nhỏ hơn nhiều so với .NET Core so với .NET Framework. Bởi vì nó là framework đa nền tảng, bạn có thể triển khai các ứng dụng máy chủ cho các container Linux Docker chẳng hạn.

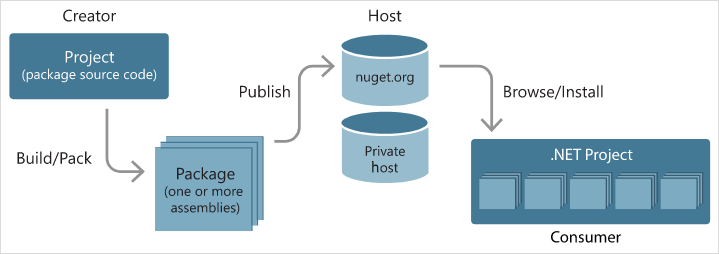


Hình 2.5: Mô hình của Containers

- Các container trên Docker có thể được lưu trữ trong cơ sở hạ tầng Linux hoặc Windows của riêng bạn hoặc trong một dịch vụ cloud như Azure Container Service. Azure Container Service có thể quản lý, dàn xếp và mở rộng các ứng dụng dựa trên container trong cloud.

**4. Dùng chung toàn bộ Nuget Package.**

- Nuget Package là 1 extension giúp bạn dễ dàng hơn trong việc quản lý, cài đạt thêm, xóa các bộ thư viện và công cụ từ bên thứ ba vào project sử dụng môi trường .NET. Nhận thấy rằng khi muốn thêm 1 thư viện vào project, bạn đều phải lặp đi lặp lại việc tải thư viện về, “gắn” nó vào project, và điều chỉnh lại project, Nuget add-on được cài vào Visual Studio cho phép quản lý vô số các thư viện mở và làm các việc trên giúp bạn.

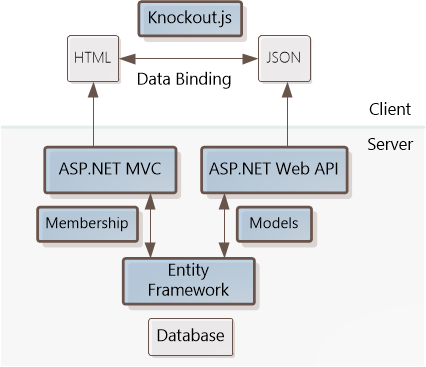


Hình 2.6: Nuget Package

- ASP.NET Core có một số thay đổi kiến trúc lớn. ASP.NET Core không còn dựa trên System.Web.dll nữa. Nó được dựa trên một tập hợp các gói, các module hay cũng được gọi là các Nuget packages. Điều này cho phép bạn tối ưu ứng dụng của bạn để chỉ bao gồm những packages nào cần thiết. Lợi ích của nó là giúp cho ứng dụng nhỏ hơn, bảo mật chặt chẽ hơn, giảm sự phức tạp, tối ưu hiệu suất hoạt động và giảm chi phí, thời gian cho việc phát triển.

**5. ASP.NET Core hợp nhất ASP.NET MVC và ASP.NET Web API.**

- Trước ASP.NET Core, các nhà phát triển thường sử dụng các khung MVC và Web API. MVC được thiết kế để tạo các ứng dụng web phục vụ HTML. ASP Web được thiết kế để tạo các dịch vụ RESTful bằng JSON hoặc XML.



Hình 2.7: ASP .Net Core hợp nhất MVC và Web API

- Với ASP.NET Core, MVC và Web API đã hợp nhất với nhau. Luôn luôn có rất nhiều sự trùng lặp với hai khung. MVC có thể trả về dữ liệu JSON thay vì HTML. Kết hợp chúng là một động thái tốt và đơn giản hóa sự phát triển.

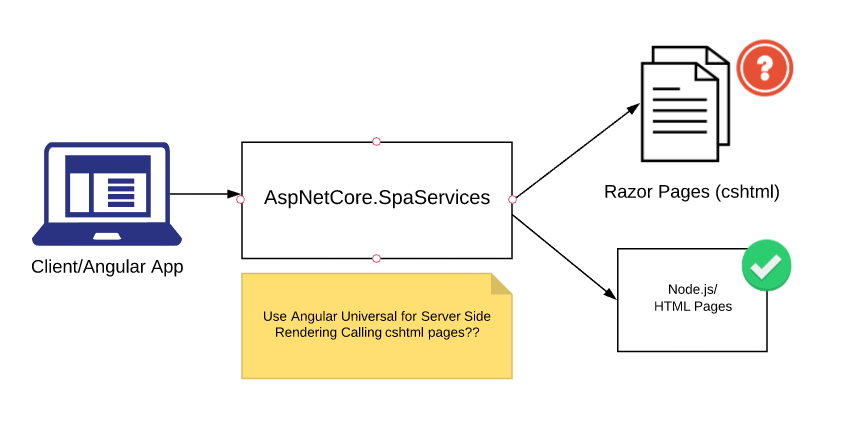
- Bạn có thể tạo ra các ứng dụng web có thể testing theo mô hình MVC(Model-View-Controller). Và bạn cũng có thể xây dựng HTTP services cái mà hỗ trợ nhiều định dạng và có đầy đủ hỗ trợ cho nội dung của dữ liệu trả về.

**6. Có thể host trên IIS hoặc tự host.**

- Internet Information Services (IIS) chính là các dịch vụ dành cho máy chủ chạy trên nền hệ điều hành Window nhằm cung cấp và phân tán các thông tin lên mạng, nó bao gồm nhiều dịch vụ khác nhau như Web Server, FTP Server…

**7. Dễ dàng tích hợp với các framework frontend như Angular, Knockout...**

- Nếu đã từng xây dựng một vài ứng dụng Web, chắc hẳn bạn đã từng nghe đến cái tên Angular, một Frameworks Javascript giúp chúng ta xây dựng ứng dụng Web đầy đủ tính năng từ phía Client. Angular giúp nâng cao năng suất của các lập trình viên, có cấu trúc phát triển rõ ràng…



Hình 2.8 Angular và ASP .Net Core

**8. Hỗ trợ cấu hình cho nhiều môi trường.**

- Cấu hình là các tham số hoặc các cài đặt cụ thể cho ứng dụng. Các cài đặt này được lưu trữ riêng biệt trong code và trong các file độc lập. Nó giúp các developer và quản trị kiểm soát và dễ dàng thay đổi cách mà ứng dụng chạy.

- ASP.NET Core hỗ trợ đọc cấu hình từ các nguồn khác nhau và các định dạng khác nhau. Một vài nguồn được sử dụng phổ biến như:

- Định dạng file (JSON, INI hoặc XML)

Ví dụ dưới đây thể hiện một cấu hình JSON đơn giản:



- Command line Arguments (tham số dòng lệnh)

Hệ thống cấu hình có thể được load từ các tham số của dòng lệnh. Phương thức CreateDefaultBuilder tải các đối số của dòng lệnh sử dụng.

- Environment variables (biến môi trường)

Phương thức config.AddEnvironmentVariables();

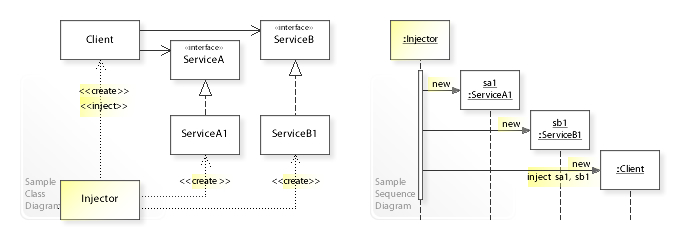
Load biến môi trường vào bộ nhớ của configuration collection. Bạn có thể đọc như sau:

C:\Users\Asus\Desktop\10.PNG

- Custom Provider (cái này là tự tạo ra provider riêng theo ý muốn)

**9. Có sẵn Dependency Injection.**

- Dependency Injection: Đây là một cách để hiện thực Inversion of Control Pattern (Có thể coi nó là một design pattern riêng cũng được). Các module phụ thuộc (dependency) sẽ được inject vào module cấp cao. DI được dùng để làm giảm sự phụ thuộc giữa các module, dễ dàng hơn trong việc thay đổi module, bảo trì code và testing.



Hình 2.9: Sơ đồ UML về Class và Sequence cho mẫu thiết kế Dependency Injection

## **2. React Native**

### **2.1. React native là gì?**

- React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.

- Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái.

### **2.2. Native app và Hybrid app**

- Native app:

+ Là tên gọi dành cho những ứng dụng được xây dựng và phát triển bằng những công cụ do chính nhà phát triển cung cấp cho lập trình viên. Hiện nay, trên thế giới có hai nhà phát triển lớn tồn tại song song và cạnh tranh lẫn nhau đó là Android và iOS.

+ Được tạo ra bằng ngôn ngữ của hệ điều hành và các tính năng có sẵn trên hệ điều hành đó. Điều này cho phép các ứng dụng vận hành với tốc độ cao nhất mà không cần thông qua bất kì ứng dụng bên thứ ba hay engine nào.

+ Các ứng dụng Android có thể được viết bằng hai ngôn ngữ là Java và Kotlin. Trong khi đó, iOs app được viết bằng ngôn ngữ hướng đối tượng Swift, trước đây là Objective C. Công cụ hỗ trợ phát triển ứng dụng cho hệ điều hành Android và iOs có thể tham khảo là Android Studio và Xcode.

- Hybrid app:

+ Có thể được hiểu là “đứa con” giữa ứng dụng web và mobile (thiết bị di động). Đây là một loại ứng dụng vừa có khả năng cài đặt trên điện thoại giống như các ***Native App*** vừa có thể sử dụng trên nền tảng web mà không cần cài đặt trên máy.

+ Bên cạnh việc các ứng dụng ***Hybrid*** được xây dựng trên ngôn ngữ HTML, CSS và JS, thông thường, rất khó để phân biệt được ***Native App*** và ***Hybrid App***. Dù vậy, Hybrid App có nhược điểm là hiệu năng sẽ bị giảm và không tương tác được hết các tài nguyên hệ thống phần mềm và phần cứng.

-Cách hoạt động của React Native:

A picture containing text, clock

Description automatically generated

Hình 2.10: Cách hoạt động của React Native

- Bằng cách tích hợp 2 thread là Main Thread và JS Thread cho ứng dụng mobile. Với ***Main Thread*** sẽ đảm nhận vai trò cập nhật giao diện người dùng(UI). Sau đó sẽ xử lý tương tác người dùng. Trong khi đó, **JS Thread** sẽ thực thi và xử lý code Javascript. Hai luồng này hoạt động độc lập với nhau.

- Để tương tác được với nhau hai Thread sẽ sử dụng một Bridge(cầu nối). Cho phép chúng giao tiếp mà không phụ thuộc lẫn nhau, chuyển đổi dữ liệu từ thread này sang thread khác. Dữ liệu từ hai Thread được vận hành khi tiếp nối dữ liệu cho nhau.

### **2.3. Ưu nhược điểm của React Native là gì?**

- Ưu điểm

* Tối ưu thời gian.
* Hiệu năng ổn định.
* Tiết kiệm chi phí.
* Đội ngũ phát triển ứng dụng không quá lớn.
* Ứng dụng tin cậy, ổn định.
* Xây dựng ứng dụng ít native code nhất cho nhiều hệ điều hành khác nhau.
* Trải nghiệm người dùng tốt hơn khi so sánh với ứng dụng Hybrid.

- Nhược điểm

* Yêu cầu Native code.
* Hiệu năng kém hơn so với Native App.
* Bảo mật chưa thật sự tốt do dùng JS.
* Quản lý bộ nhớ.
* Tùy biến chưa thật sự tốt ở một số module.

### **2.4. Một số khái niệm React Native**

- Components: là một khái niệm cơ bản của cả React và React native. Chính việc chia nhỏ ứng dụng thành các components nhỏ tạo nên tính tái sử dụng cao và khả năng mở rộng của chúng.

- Props:

+ Props là viết tắt của Properties. Một điều mà bạn cần phải nhớ khi sử dụng props đó là không bao giờ nên thay đổi giá trị của nó, hay nói cách khác, đây là một dữ liệu immutable.

+ Các component nhận props từ component cha. Bạn không được thay đổi giá trị của props trong các component này mà chỉ được phép đọc giá trị ra thôi. Trong React thì dữ liệu sẽ đi theo một chiều, có nghĩa là từ component cha đến các component con.

- State: hoạt động khác với Props. State là dữ liệu nội bộ của một Component, trong khi props là dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu mutable. Vì đặc điểm này nên chúng ta hay sử dụng State để thay đổi dữ liệu của view, binding data lại view khi có thay đổi. Nhưng chúng ta không dùng this.state để gán lại giá trị thay đổi cho nó, mà chúng ta sẽ dùng this.setState. Function này sẽ trigger cho class rằng hãy render lại component và các component con của nó, còn this.state thì không.

### **2.5. Công cụ phát triển React Native**

+ React Native cung cấp một số công cụ phát triển miễn phí cho những nhà phát triển. Có thể kể đến như:

* Hot Reloading
* Chrome Dev Tools: để kiểm tra yêu cầu mạng, hiển thị bản ghi giao diện và dừng mã trên các trình gỡ lỗi
* Redux DevTools: để kiểm tra trạng thái của store Redux

### **2.6. Một số thư viện thông dụng**

**1. Native Base:**

- NativeBase là thư viện components UI rất phổ biến cung cấp hàng chục components đa nền tảng cho React Native. Khi sử dụng NativeBase, bạn có thể sử dụng bất kỳ thư viện bên thứ ba riêng nào và dự án đi kèm với một hệ sinh thái phong phú xung quanh nó, từ bộ [starter-kits](https://startreact.com/themes/native-starter/?utm_source=nativebase&utm_medium=showcase+section&utm_campaign=nativebase) hữu ích đến các [themes](https://nativebase.io/nativebase-customizer) có thể tùy chỉnh. Tất cả đều rất đẹp và cực kỳ dễ dàng sử dụng cũng như tuỳ biến.

**2. React Native Elements**

- React-native-elements là bộ công cụ UI đa nền tảng tùy biến cao được xây dựng hoàn toàn bằng Javascript. Các tác giả thư viện nói rằng, ý tưởng với React Native Elements liên quan đến cấu trúc components hơn là thiết kế, có nghĩa là việc implement sẽ mượt mà hơn, chúng ta hoàn toàn có thể custom tuỳ theo ý thích, việc có thể custom ở mức độ cao luôn luôn là một ưu điểm lớn với chúng ta.

**3. React Native Vector Icons:**

A picture containing people, boat, group, side

Description automatically generated

Hình 2.11: Ví dụ về các biểu tượng trong thư viện React Native Vector Icons

- Thư viện này là một bộ icon có thể tùy chỉnh cho React Native với sự hỗ trợ cho NavBar / TabBar / ToolbarAndroid, source icon và kiểu dáng đầy đủ. Nó cực kỳ hữu ích và được sử dụng bởi hàng ngàn ứng dụng cũng như các thư viện component UI khác (như react-native-paper). Thư viện cung cấp các bộ icon được tạo sẵn sẵn trong hộp.

**4. React Native Image Picker**

A screenshot of a cell phone

Description automatically generated

Hình 2.12: Ứng dụng React Native Image Picker

- Một thư viện thiết yếu khác cho bất kỳ ứng dụng nào có tính năng tải lên hình ảnh hoặc xử lý hình ảnh. Nó hỗ trợ lựa chọn từ bộ sưu tập và ảnh chụp từ máy ảnh. Một tính năng hữu ích khác trong thư viện này tôi thích là tùy chọn chất lượng của hình ảnh theo ý cá nhân. Tính năng này giải quyết các sự cố bộ nhớ do hình ảnh có độ phân giải cao.

**5. React Native Router Flux**

- Điều hướng là một trong những mối quan tâm chính trong cộng đồng React Native bởi vì nó không có hệ thống định vị mặc định. Bất kể hệ thống định vị mà React Native gốc bản xuất hiện luôn có thay đổi đột ngột hoặc không ổn định.

- Thư viện này đã giúp tôi thực hiện điều hướng nhanh chóng bằng một API khai báo rất đơn giản. Nó duy trì một ngăn xếp cho các tuyến đường và chuyển hướng đến bất kỳ cảnh trong ứng dụng đơn giản như gọi một chức năng.

- Nó hỗ trợ chuyển hướng theo thẻ, ngăn kéo và Modals quá. Các modal có thể được định nghĩa là các cảnh để một Modal có thể được gọi từ bất kỳ cảnh nào.

- Hầu hết các trình diễn ứng dụng ở trên sử dụng Phản ứng-bản-router-Flux cho hệ thống định vị.

- Nếu bạn muốn xem trực tuyến, hãy tải xuống ứng dụng Helpy, nó được xây dựng bằng cách sử dụng phản ứng tự nhiên có sẵn trên cả Android và iOS. Helpy là một ứng dụng  Q n A. Nói một cách đơn giản, đó là một sự thay thế dựa trên vị trí và đơn giản cho Quora.

**6.** [**Redux**](https://redux.js.org/)

- Đây là một thư viện độc đáo giới thiệu một khái niệm mới để lưu trữ dữ liệu. Trong khi sử dụng Redux, bạn có thể truy vấn, chọn, chèn và cập nhật bản ghi trong cơ sở dữ liệu. Redux cũng có một tính năng thực sự hữu ích để chỉnh sửa mã trực tiếp.

**7.** [**React-Redux**](https://github.com/reduxjs/react-redux)

- Thư viện Reac-redux là một đoạn mã ngắn tích hợp bộ chứa trạng thái này vào ứng dụng của bạn. Nó là một ràng buộc cho thư viện Redux.

**8.** [**Redux-Thunk**](https://github.com/reduxjs/redux-thunk)

- Đây là một thư viện nổi tiếng để làm việc với các action không đồng bộ trong Redux. Trong khi sử dụng nó, các nhà phát triển có thể viết các trình tạo hành động trả về một hàm thay vì một hành động. Thư viện là một đoạn mã rất nhỏ mà dễ hiểu nhưng mạnh mẽ.

**9.** [**Axios**](https://github.com/axios/axios)

- Axios là một client HTTP nhẹ dành cho JavaScript, được xây dựng để gửi các yêu cầu HTTP không đồng bộ đến các điểm cuối REST và thực hiện các hoạt động CRUD. Tóm lại, điều đó có nghĩa là, bạn có thể sử dụng các hàm không đồng bộ và chờ các hàm để viết mã không đồng bộ dễ đọc hơn.

## **3. Azure Cognition Services**

### **3.1. Giới thiệu về Microsoft Cognitive Services**

**\* Microsoft Cognitive Services là gì?**

**- Microsoft Cognitive Services** bao gồm một bộ các API ứng dụng trí tuệ nhân tạo thông minh, cho phép lập trình viên ở mọi cấp độ từ những bạn sinh viên viết ứng dụng đầu tiên của mình hay những lập trình viên chuyên nghiệp làm việc cho những công ty, tổ chức lớn đều có thể tạo ra được những ứng thông minh hơn một cách dễ dàng.

- Các API của Cognitive Services được viết dưới dạng REST API do vậy lập trình viên có thể tích hợp các API này trên nhiều nền tảng khác nhau như iOS, Android, hay Windows, chỉ cần có kết nối Internet.

**\* Các API của Microsoft Cognitive Services:**

- Tính đến thời điểm viết bài viết này, Microsoft Cognitive Services bao gồm 21 API được chia thành 5 nhóm: **Vision**, **Speech**, **Language**, **Knowledge** và **Search**.

**1. Vision API:**

- Là nhóm API liên quan đến xử lý hình ảnh, trong nhóm Vision API này hiện Microsoft cung cấp 4 API sau:

**+ Computer Vision API**: API này cho phép trích xuất những thông tin có giá trị từ bức ảnh của bạn với khả năng xác định được kiểu đối tượng trong ảnh (là bánh mỳ, con chó, con mèo hay cây cối, …) hay nếu là người thì API này cũng xác định được giới tính của nhân vật trong ảnh. Ngoài ra, API này cũng hỗ trợ nhận diện được những nhận vật nổi tiếng hay trích xuất chữ có trong bức hình của bạn.

**+ Face API**: Cái tên nói lên tất cả, đây là API cho phép phát hiện khuôn mặt có trong bức hình của bạn. Ngoài ra, API này cũng trả về các thuộc tính của khuôn mặt như tuổi, giới tính, độ rạng ngời của nụ cười hay thậm chí là chiều dài của tóc, … Ngoài phát hiện khuôn mặt, Face API còn cho phép so sánh 2 khuôn mặt có phải là của cùng một người hay không.

**+ Emotion API**: API này cho phép xác định tâm trạng của người có trong bức hình xem họ đang vui, đang buồn hay đang giận dữ.

**+ Video API**: API này là một tập hợp các thuật toán xử lý video tân tiến của Microsoft. Với Video API, các nhà phát triển có thể tích hợp các tính năng chỉnh sửa video bao gồm chống rung, phát hiện khuôn mặt người, phát hiện chuyển động hay tạo video thumbnail.

**2. Speech API:**

- Là nhóm API liên quan đến xử lý âm thanh với công nghệ xử lý đến từ Bing. Trong nhóm Speech API này hiện Microsoft cung cấp 3 API:

**+ Bing Speech API**: API này cho phép trích xuất một tập tin âm thanh sang dạng chữ, chuyển đổi định dạng chữ sang âm thanh (tức là đọc chữ) hay đoán ý của một câu nói.

**+ Custom Recognition Intelligent Service (CRIS)**: CRIS cho phép bạn có thể tùy biến language model và acoustic model sao cho phù hợp với ứng dụng hoặc người dùng của bạn.

**+ Speaker Recognition API**: Với những thuật toán nhận dạng giọng nói tân tiến của Microsoft, API này cho phép nhận dạng giọng nói của người nói trong một tập tin âm thanh. API này bao gồm 2 thành phần: speaker verification và speaker identification tạm dịch tương ứng là xác nhận người nói và xác định người nói. **Speaker Verification** cho phép xác nhận và xác thực người dùng bằng giọng nói của họ. Lập trình viên chỉ cần cho người dùng đọc một đoạn văn bản có sẵn để lưu lại dữ liệu giọng nói của họ (enrollment) rồi ở mỗi lần cần xác thực, người dùng chỉ cần đọc lại chính xác đoạn văn bản đã được dùng để lấy dữ liệu giọng nói của mình (bước enrollment) để xác thực. **Speaker Identification** có thể xác định được người đang nói trong một tập tin âm thanh dựa trên một tập dữ liệu các người nói tiềm năng. Tính năng này cũng có thể được sử dụng để xác thực người dùng bằng giọng nói. Tuy nhiên thay vì phải đọc chính xác một đoạn văn bản cố định như Speaker Verification thì khi sử dụng tính năng Speaker Identification này, người dùng có thể đọc một đoạn văn bản bất kỳ, API sẽ phân tích và đối chiếu với tập dữ liệu giọng nói của người dùng để so sánh và xác thực.

**3. Language API:**

- Là nhóm API liên quan đến xử lý ngôn ngữ. Nhóm Language API này hiện được Microsoft cung cấp 5 API:

**+ Bing Spell Check API**: API này cho phép phát hiện và sửa các lỗi chính tả có trong một đoạn văn bản mà bạn cung cấp. API còn có khả năng phát hiện từ lóng, sửa lỗi tên riêng hay sửa các từ đồng âm, …

**+ Web Language Model API**: API này giúp hỗ trợ xử lý ngôn ngữ tự nhiên, với khả năng chèn khoảng cách vào 1 đoạn văn bản được viết liền nhau như hashtag hay đường dẫn.

**+ Linguistic Analysis API**: The Linguistic Analysis API giúp bạn hiểu sâu hơn văn bản của mình. API này sẽ giúp phân tích cú pháp của ngôn ngữ tự nhiên để dễ dàng xác định được các thực thể (danh từ) hay các hành động (động từ) có trong văn bản. Việc xử lý văn bản này có thể hữu ích cho các công việc phân tích như phân tích tâm lý.

**+ Language Understanding Intelligent Service (LUIS)**: LUIS cho phép lập trình viên xây dựng các model hiểu được ngôn ngữ tự nhiên cũng như hiểu được các câu lệnh riêng được thiết kế riêng cho ứng dụng của bạn. Ví dụ: Bạn có thể nói “bật đèn trong phòng ngủ”, gửi câu lệnh đó tới LUIS model, và thay vì trả lại chính xác các câu từ có trong câu lệnh trên, LUIS sẽ trả về dữ liệu chứa: thông tin hành động là “bật” vị trí là “phòng ngủ” và đối tượng hướng đến là “bóng đèn”, từ đó ứng dụng có thể dễ dàng xử lý được câu lệnh của bạn.

**+ Text Analytics API**: API này giúp xác định các ẩn ý, từ khóa, chủ đề hay ngôn ngữ được sử dụng có trong một đoạn văn bản.

**4. Knowledge API:**

- Là nhóm API liên quan đến tri thức. Nhóm Knowledge API này hiện bao gồm 4 API:

**+ Academic Knowledge API:**API này cho phép lập trình viên xây dựng những giải pháp tìm kiếm tài liệu học thuật với tính năng Interpret, trả về kết quả gợi ý cho từ khóa mà người dùng nhập vào dựa vào nguồn dữ liệu phong phú từ hệ thống [Microsoft Academic Graph](https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/) (MAG).

**+ Knowledge Exploration Service API**: API này cho phép lập trình viên xây dựng những giải pháp tìm kiếm sử dụng ngôn ngữ tự nhiên bằng cách dịch ngôn ngữ tự nhiên mà người dùng nhập vào sang các biểu thức truy vấn có cấu trúc mà máy tính có thể dễ dàng hiểu và xử lý được.

**+ Entity Linking Intelligence Service API**: Với một đoạn văn bản, Entity Linking Intelligence Service sẽ nhận dạng và xác định từng thực thể (entity) có trong đoạn văn dựa vào ngữ cảnh của đoạn văn đó và sẽ liên kết những entity này tới Wikipedia. Lấy ví dụ rằng bạn có một đoạn văn bản trong đó chứa từ cloud, từ cloud này có thể hiểu sang thành “Cloud Computing” (điện toán đám mây) hay “Cloud” (đám mây trên trời), dựa vào ngữ cảnh mà API này sẽ xác định được rằng từ cloud có ý nghĩa như thế nào.

**+ Recommendations API**: API này cho phép xây dựng các giải pháp khuyến nghị cho người dùng. Chẳng hạn như bạn xây dựng một ứng dụng bán hàng, sử dụng API này cho phép bạn dễ dàng xây dựng ra các tính năng khuyến nghị mua hàng như “Các sản phẩm được bán chạy”, “Các sản phẩm được mua cùng” hay “Những sản phẩm hàng đầu trong mặt hàng Đồ gia dụng” chẳng hạn, từ đó sẽ khuyến khích người dùng mua nhiều hơn.

**5. Search API:**

- Search API cung cấp các giải pháp tìm kiếm sử dụng sức mạnh của Bing.com cũng như từ nhiều đối tác như AOL, Apple, Amazon, Yahoo, …. Nhóm Search API bao gồm 5 API cho các tác vụ tìm kiếm trang web, hình ảnh, video hay tin tức.

**+ Bing Web Search API**: Đây là API chủ lực của gói Search API. Chỉ với một cú pháp lệnh gọi đến API này, lập trình viên có thể lấy được các kết quả trả về cho trang web, hình ảnh, video hay tin tức tương ứng. Nó khá tương tự với việc bạn tìm kiếm trên các công cụ tìm kiếm như Bing.com hay Google.com. Ngoài ra, lập trình viên cũng sẽ nhận được những tính năng mạnh mẽ từ công cụ tìm kiếm Bing Search như ranking kết quả tìm kiếm, phân loại kết quả tìm kiếm theo vùng, …

**+ Bing Autosuggest API**: API này cho phép lập trình viên có thể xây dựng tính năng đề xuất các từ khóa tìm kiếm liên quan kể cả khi từ khóa tìm kiếm chưa được điền đầy đủ. Ví dụ nếu người dùng gõ từ khóa tìm kiếm là “Thời tiết tại H”, API sẽ trả về danh sách các từ khóa gợi ý như “Thời tiết tại Hà Nội”, “Thời tiết tại Hồ Chí Minh” hay “Thời tiết tại Hà Giang” chẳng hạn.

**+ Bing Image Search API**: API này cho phép lập trình viên có thể tìm kiếm các hình ảnh tương ứng với từ khóa nhập vào. Ngoài trả về đường dẫn của hình ảnh, API này cũng trả về các metadata hữu ích như kích thước ảnh, màu chủ đạo của ảnh, …

**+ Bing Video Search API**: API này cho phép lập trình viên có thể tìm kiếm các video tương ứng với từ khóa nhập vào. Ngoài trả về đường dẫn của video, API này cũng trả về các metadata hữu ích khác như tên nhà sản xuất, định dạng mã hóa, ảnh thumbnail, …

**+ Bing News Search API**: API này cho phép lập trình viên có thể tìm kiếm các tin tức, bài báo tương ứng với từ khóa nhập vào. Ngoài ra, API cũng trả về các metadata hữu ích khác như thể loại, thông tin nhà xuất bản, ngày xuất bản, …

+ Có thể thấy với bộ 21 API mà dịch vụ **Microsoft Cognitive Services** cung cấp, lập trình viên có thể thỏa sức xây dựng các ứng dụng tích hợp các tính năng thông minh sử dụng sức mạnh từ machine learning mà dịch vụ này mang lại một cách dễ dàng chỉ bằng việc gọi API và xử lý kết quả trả về với định dạng JSON.

### **3.2. Áp dụng Microsoft Cognitive Service**

- Ví dụ này sẽ trình bày cách xác định các khuôn mặt chưa biết bằng cách sử dụng các objects Persongroup, được tạo từ những người đã biết trước. Được viết bằng C # bằng cách sử dụng thư viện Azure Cognitive Services Face.

- Cách tạo một Persongroup. Persongroup này chứa danh sách những người đã biết.

- Cách thêm khuôn mặt cho mỗi người. Những khuôn mặt này được sử dụng làm cơ sở để xác định từng người. Nên sử dụng hình ảnh mặt trước rõ ràng. Ví dụ là ID ảnh. Một bộ ảnh chất lượng bao gồm khuôn mặt của cùng một người trong các tư thế, màu sắc quần áo hoặc kiểu tóc khác nhau.

+ Để thực hiện việc này, hãy chuẩn bị:

- Một vài bức ảnh với khuôn mặt người. Ở đây chọn hình ảnh 1 vài siêu sao bóng đá: Ronaldo, Messi, De Bruyne.

- Một loạt các bức ảnh để kiểm tra. Các bức ảnh có thể chứa hoặc không chứa khuôn mặt của những người đã xác định ở trên. Chúng được sử dụng để kiểm tra nhận dạng. Cũng chọn một số hình ảnh mẫu từ liên kết trước.

**\* Triển khai**

**Bước 1: Phân quyền khi gọi API**

- Mỗi khi gọi API đều yêu cầu 1 subscription key. Key này có thể được chuyển thành tham số query string hoặc được chỉ định trong request của phần header. Để chuyển subscription key qua query string, hãy xem request URL cho Face - Detect làm ví dụ:

https://westus.api.cognitive.microsoft.com/face/v1.0/detect[?returnFaceId][&returnFaceLandmarks][&returnFaceAttributes]

&subscription-key=<Subscription key>

- Thay vào đó, chỉ định subscription key trong HTTP request header ocp-apim-subscription-key: <Subscription Key>. Khi bạn sử dụng thư viện, subscription key được chuyển thành hàm tạo của class FaceClient. Ví dụ:

|  |
| --- |
| **private readonly** IFaceClient faceClient = **new** FaceClient(  **new** ApiKeyServiceClientCredentials(**"<subscription key>"**),  **new** System.Net.Http.DelegatingHandler[] { }); |

- Để lấy subscription key, hãy truy cập Azure Marketplace từ cổng thông tin Azure. Để biết thêm thông tin, xem tại <https://azure.microsoft.com/en-us/try/cognitive-services/>.

**Bước 2: Tạo PersonGroup**

- Ở bước này, một Persongroup có tên "Players" chứa Ronaldo, Messi và De Bruyne. Mỗi người có vài khuôn mặt đăng ký. Các khuôn mặt phải được detect từ các hình ảnh. Sau tất cả các bước này, bạn có một Persongroup như hình ảnh sau:

|  |  |  |
| --- | --- | --- |
| Person Group: Players | Messi | A person smiling for the camera  Description automatically generatedA person smiling for the camera  Description automatically generatedA person smiling for the camera  Description automatically generated |
| Ronaldo | A person smiling for the camera  Description automatically generatedA person smiling for the camera  Description automatically generatedA person smiling for the camera  Description automatically generated |
| De Bruyne | A person smiling for the camera  Description automatically generatedA person in a blue shirt  Description automatically generatedA close up of a person  Description automatically generated |

**- Bước 2.1: Xác định từng người trong PersonGroup**

Mỗi Person là đơn vị xác định cơ bản. Một người có thể có một hoặc nhiều khuôn mặt được biết đến đã đăng ký. Một Persongroup là một tập hợp của nhiều Person. Mỗi Person được định nghĩa cụ thể trong PersonGroup. Identification được thực hiện đối với PersonGroup. Nhiệm vụ là tạo một Persongroup và sau đó tạo những người trong đó, chẳng hạn như Ronaldo, Messi, De Bruyne.

Đầu tiên, tạo một Persongroup mới bằng cách sử dụng Persongroup - Create API. API thư viện tương ứng là phương thức CreatePersongroupAsync cho class FaceClient. Group ID được chỉ định để tạo group là duy nhất cho mỗi đăng ký. Bạn cũng có thể get, update hoặc delete Persongroups bằng cách sử dụng API Persongroup khác.

Sau khi xác định group, bạn có thể xác định những người trong đó bằng cách sử dụng Persongroup Person - Create API. Phương thức là CreatePersonAsync. Bạn có thể thêm một khuôn mặt cho mỗi Person sau khi chúng được tạo.

|  |
| --- |
| // Create an empty PersonGroup  **string** personGroupId = **"players"**;  await faceClient.PersonGroup.CreateAsync(personGroupId, **"Players"**);  // Define Messi  CreatePersonResult friend1 = **await** faceClient.PersonGroupPerson.CreateAsync(  // Id of the PersonGroup that the person belonged to  personGroupId,  // Name of the person  **"Messi"**  );  // Define Ronaldo and De Bruyne in the same way |

**- Bước 2.2: Detect khuôn mặt và đăng kí đúng cho từng người.**

Việc detect được thực hiện bằng cách gửi request web "POST" đến Face - Detect API với tệp hình ảnh trong body request HTTP. Khi bạn sử dụng thư viện, việc phát hiện khuôn mặt được thực hiện thông qua một trong các phương thức Detect..Async của class FaceClient.

Đối với mỗi khuôn mặt được detect, hãy gọi Persongroup Person – Add Face để thêm nó vào đúng người.

Đoạn code sau trình bày quá trình detect khuôn mặt từ hình ảnh và thêm nó vào Person:

|  |
| --- |
| // Directory contains image files of Messi  **const string** friend1ImageDir = **@"D:\Pictures\Players\Messi\"**;    **foreach**(**string** imagePath **in** Directory.GetFiles(friend1ImageDir, **"\*.jpg"**))  {  **using** (Stream s = File.OpenRead(imagePath))  {  // Detect faces in the image and add to Messi  await faceClient.PersonGroupPerson.AddFaceFromStreamAsync(  personGroupId, friend1.PersonId, s);  }  }  // Do the same for Ronaldo and De Bruyne |

Nếu hình ảnh chứa nhiều hơn một khuôn mặt, chỉ có khuôn mặt lớn nhất được thêm vào. Bạn có thể thêm khuôn mặt khác cho Person. Truyền một chuỗi theo định dạng "targetFace = left, top, width, height" cho Persongroup Person – Add Face tham số truy vấn API’s targetFace. Bạn cũng có thể sử dụng tham số tùy chọn targetFace cho phương thức AddPersonFaceAsync để thêm các khuôn mặt khác. Mỗi khuôn mặt được thêm vào Person được cấp một ID khuôn mặt duy trì duy nhất. Bạn có thể sử dụng ID này trong Persongroup Person – Delete Face và Face – Identify.

**Bước 3: Train the PersonGroup**

- Persongroup phải được Train trước khi việc Identification có thể được thực hiện bằng cách sử dụng nó. Persongroup phải được Train lại sau khi bạn thêm hoặc xóa bất kỳ Person nào hoặc nếu bạn chỉnh sửa khuôn mặt đã đăng ký của Person. Việc Train được thực hiện bởi Persongroup - Train API. Khi bạn sử dụng thư viện, nó sẽ gọi đến phương thức TrainPersongroupAsync:

|  |
| --- |
| **await** faceClient.PersonGroup.TrainAsync(personGroupId); |

- Train là một quá trình không đồng bộ. Nó có thể không được hoàn thành ngay cả sau khi phương thức TrainPersongroupAsync return. Bạn có thể phải cần Status về Query the Training. Sử dụng Persongroup – Get Training Status API hoặc phương thức GetPersongroupTrainingStatusAsync của thư viện. Đoạn code sau đây thể hiện logic đơn giản của việc chờ việc Train Persongroup kết thúc:

|  |
| --- |
| TrainingStatus trainingStatus = **null**;  **while**(**true**)  {  trainingStatus = await faceClient.PersonGroup.GetTrainingStatusAsync(personGroupId);    **if** (trainingStatus.Status != TrainingStatusType.Running)  {  **break**;  }    await Task.Delay(1000);  } |

**Bước 4: Xác định một khuôn mặt đối một Persongroup đã xác định**

- Khi dịch vụ Face thực hiện Identification, nó sẽ tính toán sự giống nhau của khuôn mặt kiểm tra giữa tất cả các khuôn mặt trong một group. Nó trả về những Person có thể so sánh đúng nhất cho việc kiểm tra khuôn mặt. Quá trình này được thực hiện thông qua API Face - Identify hoặc phương thức IdentifyAsync của thư viện.

- Việc kiểm tra khuôn mặt phải được Detect bằng cách thực hiện các bước trước đó. Sau đó, FaceID được chuyển đến API Identification dưới dạng argument thứ hai. Nhiều faceID có thể được xác định cùng một lúc. Kết quả bao gồm tất cả các kết quả được xác định. Mặc định thì quy trình nhận dạng chỉ trả về một người phù hợp nhất với khuôn mặt kiểm tra. Nếu bạn thích, chỉ định tham số tùy chọn maxNumOfCandidatesReturned để cho phép quá trình nhận dạng trả lại nhiều ứng viên hơn.

- Đoạn code sau đây thể hiện quá trình Identification:

|  |
| --- |
| **string** testImageFile = **@"D:\Pictures\test\_img1.jpg"**;  **using** (Stream s = File.OpenRead(testImageFile))  {  var faces = **await** faceClient.Face.DetectWithStreamAsync(s);  var faceIds = faces.Select(face => face.FaceId.Value).ToArray();  var results = **await** faceClient.Face.IdentifyAsync(faceIds, personGroupId);  **foreach**(var identifyResult **in** results)  {  Console.WriteLine(**"Result of face: {0}"**, identifyResult.FaceId);  **if** (identifyResult.Candidates.Length == 0)  {  Console.WriteLine(**"No one identified"**);  }  **else**  {  // Get top 1 among all candidates returned  var candidateId = identifyResult.Candidates[0].PersonId;  var person = **await** faceClient.PersonGroupPerson.GetAsync(personGroupId, candidateId);  Console.WriteLine(**"Identified as {0}"**, person.Name);  }  }  } |

- Sau khi bạn hoàn thành các bước, hãy cố gắng xác định các khuôn mặt khác nhau. Xem các khuôn mặt của Ronaldo, Messi hoặc De Bruyne có thể được xác định chính xác theo các hình ảnh được tải lên để Detect khuôn mặt. Xem các ví dụ sau:

A person making a face for the camera

Description automatically generated

Hình 2.13: Ví dụ minh hoạ việc xác định khuôn mặt từ ảnh

**Bước 5: Request với quy mô lớn**

Persongroup có thể chứa tới 10.000 Person dựa trên giới hạn thiết kế trước đó. Để biết thêm thông tin về các kịch bản lên tới hàng triệu tỷ, hãy xem <https://docs.microsoft.com/en-us/azure/cognitive-services/face/face-api-how-to-topics/how-to-use-large-scale>.

Tóm lại: Ta đã học được quy trình tạo Persongroup và xác định Person. Các tính năng sau đây đã được giải thích và chứng minh:

Detect khuôn mặt bằng cách sử dụng [Face - Detect](https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d) API.

Create PersonGroups bằng cách sử dụng [PersonGroup - Create](https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395244) API.

Create persons bằng cách sử dụng [PersonGroup Person - Create](https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f3039523c) API.

Train a PersonGroup bằng cách sử dụng [PersonGroup – Train](https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395249) API.

Xác định những khuôn mặt lạ đối với the PersonGroup bằng cách sử dụng [Face - Identify](https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395239) API.

## **4. Đặc thù của hệ thống**



Hình 2.14: Sơ đồ kiến trúc hệ thống

# **Chương 3: Phân tích thiết kế**

## **1. Sơ đồ Use-case**



Hình 3.1: Sơ đồ Use-case

## **2. Class Diagram**

### **2.1. Mô tả thực thể**

#### 2.1.1. Module SinhVien

Bảng 3.1: Chi tiết thực thể Module SinhVien

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | int | Khoá chính |
| 2 | MSSV | Mã số sinh viên | varchar(10) |  |
| 3 | matKhau | Mật khẩu | varchar(50) |  |
| 4 | hoTen | Họ tên | nvarchar(100) |  |
| 5 | ngaySinh | Ngày sinh | date |  |
| 6 | gioiTinh | Giới tính | bit |  |
| 7 | diaChi | Địa chỉ | nvarchar(100) |  |
| 8 | soDienThoai | Số điện thoại | varchar(15) |  |
| 9 | Email | Email | varchar(50) |  |
| 10 | ghiChu | Ghi chú | nvarchar(max) |  |
| 11 | hasRegisteredFace | Xét đăng kí khuôn mặt | bit | Thể hiện sinh viên đã đăng kí khuôn mặt hay chưa. |

#### 2.1.2. Module GiangVien

Bảng 3.2: Chi tiết thực thể Module GiangVien

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | int | Khoá chính |
| 2 | MSGV | Mã số giảng viên | varchar(10) |  |
| 3 | matKhau | Mật khẩu | varchar(50) |  |
| 4 | hoTen | Họ tên | nvarchar(100) |  |
| 5 | ngaySinh | Ngày sinh | date |  |
| 6 | gioiTinh | Giới tính | bit |  |
| 7 | diaChi | Địa chỉ | nvarchar(100) |  |
| 8 | soDienThoai | Số điện thoại | varchar(15) |  |
| 9 | Email | Email | varchar(50) |  |
| 10 | ghiChu | Ghi chú | nvarchar(max) |  |

#### 2.1.3. Module LopHoc

Bảng 3.3: Chi tiết thực thể Module LopHoc

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | int | Khoá chính |
| 2 | IDMonHoc | Mã môn học | int | Khoá phụ Module MonHoc(ID) |
| 3 | IDGiangVien | Mã số giảng viên | int | Khoá phụ Module GiangVien(ID) |

#### 2.1.4. Module MonHoc

Bảng 3.4: Chi tiết thực thể Module MonHoc

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | Int | Khoá chính |
| 2 | tenMonHoc | Tên môn học | nvarchar(50) |  |

#### 2.1.5. Module SinhVienTrongLopHoc

Bảng 3.5: Chi tiết thực thể Module SinhVienTrongLopHoc

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | Int | Khoá chính |
| 2 | IDLopHoc | Mã lớp học | Int | Khoá phụ Module LopHoc (ID) |
| 3 | IDSinhVien | ID sinh viên | int | Khoá phụ Module SinhVien(ID) |

#### 2.1.6. Module ChiTietNgayHoc

Bảng 3.6: Chi tiết thực thể Module ChiTietNgayHoc

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | Int | Khoá chính |
| 2 | IDLopHoc | Mã lớp học | Int | Khoá phụ Module LopHoc(ID) |
| 3 | thoiGianBatDauBuoiHoc | Thời gian bắt đầu buổi học | smalldatetime | Ngày và giờ cụ thể bắt đầu lớp học |
| 4 | phongHoc | Phòng học | varchar(5) |  |

#### 2.1.7. Module ChiTietDiemDanh

Bảng 3.7: Chi tiết thực thể Module ChiTietDiemDanh

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | int | Khoá chính |
| 2 | IDChiTietNgayHoc | Mã chi tiết ngày học | int | Khoá phụ Module ChiTietNgayHoc (ID) |
| 3 | IDSinhVienTrongLopHoc | Mã sinh viên trong lớp học | int | Khoá phụ Module SinhVienTrongLopHoc (ID) |
| 4 | isDiemDanh | Xét điểm danh | bit | Sinh viên đã điểm danh hay chưa |

#### 2.1.8. Module ThongBao

Bảng 3.8: Chi tiết thực thể Module ThongBao

| **STT** | **Tên cột** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 1 | ID | ID | int | Khoá chính |
| 2 | IDLopHoc | Mã lớp học | int | Khoá phụ Module LopHoc (ID) |
| 3 | ngayTao | Ngày tạo | datetime |  |
| 4 | tieuDe | Tiêu đề | nvarchar(20) |  |
| 5 | noiDung | Nội dung | nvarchar(max) |  |

### **2.2. Sơ đồ class**



Hình 3.2: Sơ đồ class

## **3. Sequence Diagram**

### **3.1. Chức năng sinh viên**



Hình 3.3: Sơ đồ tuần tự chức năng Đăng kí khuôn mặt

### **3.2. Chức năng giảng viên**



Hình 3.4: Sơ đồ tuần tự chức năng Điểm danh

## **4. Activity Diagram**

### **4.1. Chức năng sinh viên**



Hình 3.5: Sơ đồ hoạt động chức năng Đăng kí khuôn mặt

### **4.2. Chức năng giảng viên**



Hình 3.6: Sơ đồ hoạt động chức năng Điểm danh

# **Chương 4: Triển khai**

## **1. Front-end**

### **1.1. Giao diện chung**

|  |  |
| --- | --- |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112233_F-At.jpg  Hình 4.1: Giao diện bắt đầu | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112240_F-At.jpg  Hình 4.2: Giao diện Đăng nhập |

### **1.2. Giao diện sinh viên**

|  |  |
| --- | --- |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112341_F-At.jpg  Hình 4.3: Giao diện Trang chủ | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112351_F-At.jpg  Hình 4.4: Giao diện Chi tiết thông báo chung |

|  |  |
| --- | --- |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112634_F-At.jpg  Hình 4.5: Giao diện các chức năng | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112441_F-At.jpg  Hình 4.6: Giao diện Thời khoá biểu |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112456_F-At.jpg  Hình 4.7: Giao diện điểm danh theo môn  (Có mặt) | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112549_F-At.jpg  Hình 4.8: Giao diện điểm danh theo môn  (Không có mặt) |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112602_F-At.jpg  Hình 4.9: Giao diện Cập nhật thông tin cá nhân | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112554_F-At.jpg  Hình 4.10: Giao diện Cập nhật mật khẩu |
| Hình 4.11: Giao diện Đăng kí khuôn mặt | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112622_F-At.jpg  Hình 4.12: Giao diện Thông báo  theo từng môn học |

### **1.3. Giao diện giảng viên**

|  |  |
| --- | --- |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112648_F-At.jpg  Hình 4.13: Giao diện các chức năng | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112656_F-At.jpg  Hình 4.14: Giao diện Tạo thông báo |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112755_F-At.jpg  Hình 4.15: Giao diện Danh sách thông báo  theo môn học | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112758_F-At.jpg  Hình 4.16: Giao diện chi tiết thông báo |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112806_F-At.jpg  Hình 4.17: Giao diện Thông tin cá nhân | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112554_F-At.jpg  Hình 4.18: Giao diện Cập nhật mật khẩu |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112820_F-At.jpg  Hình 4.19: Giao diện Xem danh sách sinh viên theo môn học | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112823_F-At.jpg  Hình 4.20: Giao diện Xem thời khoá biểu  theo môn học |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112832_F-At.jpg  Hình 4.21: Giao diện Danh sách điểm danh  theo môn học | C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112836_F-At.jpg  Hình 4.22: Giao diện Danh sách điểm danh  theo môn học (2) |
| C:\Users\franc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200518-112843_F-At.jpg  Hình 4.23: Giao diện Điểm danh | A group of people sitting at a table  Description automatically generated  Hình 4.24: Giao diện khi mở camera Điểm danh |
| A screenshot of a cell phone  Description automatically generated  Hình 4.25: Giao diện Cập nhật điểm danh | A screenshot of a cell phone  Description automatically generated  Hình 4.26: Giao diện thông báo  Điểm danh thành công |
| A screenshot of a cell phone  Description automatically generated  Hình 4.27: Giao diện thông báo kết quả điểm danh được gửi đến sinh viên |  |

## **2. Back-end**

### **2.1. Chức năng chung**

\*\*\* **Chức năng Đăng nhập**

- Mô tả: Giảng viên và sinh viên thực hiện đăng nhập với tài khoản và mật khẩu được cung cấp từ trước.

- Code Controller:

- Endpoint: /api/users/login

- Request: + Phương thức: POST

+ Body: FromForm {username, password}

- Response: JSON

{

    "username": "SV00000002",

    "password": null,

    "role": "sv",

    "token": "aaaaaaa.bbbbbbb.ccccccc"

}

### **2.2. Chức năng sinh viên**

#### **2.2.1. Xem danh sách điểm danh theo môn học**

- Mô tả: Sinh viên có thể tra cứu thông tin điểm danh của mình theo từng môn học, ngày học.

- Code Controller:

- Endpoint: /api/sinhvien/current/chitietdiemdanh

- Request: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [       {          "tenMonHoc": "Công nghệ phần mềm",          "ngayHoc": "2020-03-09T07:00:00",          "phongHoc": "A001",          "isDiemDanh": false       },       {         …  }  ] |

#### **2.2.2. Xem danh sách môn học**

- Mô tả: Sinh viên có thể xem danh sách những môn học mà hiện đang theo học.

- Code Controller:

- Endpoint: /api/sinhvien/current/lophoc

- Resquest: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "tenMonHoc": "Công nghệ phần mềm",          "tenGiangVien": "LÊ VĂN DŨNG",          "thongBao": [              {                  "id": 1,                  "idlopHoc": 1,                  "ngayTao": "2020-04-18T07:00:00",                  "tieuDe": "Nghỉ lễ(Cập nhật 5/1/2020 10:03:08 AM)",                  "noiDung": "Nghỉ học nha mấy em",                  "idlopHocNavigation": null              },              {  ...              }          ]      },      {       ...      }  ] |

#### **2.2.3. Xem chi tiết thông báo theo lớp**

- Mô tả: Sinh viên chi tiết thông báo từ lớp học.

- Code Controller:

- Endpoint: /api/sinhvien/current/lophoc/{idLopHoc}/thongbao/{idThongBao}

- Resquest: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 1,      "idlopHoc": 1,      "ngayTao": "2020-04-18T07:00:00",      "tieuDe": "Nghỉ lễ(Cập nhật 5/1/2020 10:03:08 AM)",      "noiDung": "Nghỉ học nha mấy em",      "idlopHocNavigation": {          "id": 1,          "idmonHoc": 1,          "idgiangVien": 1,          "idgiangVienNavigation": null,          "idmonHocNavigation": null,          "chiTietNgayHoc": [],          "sinhVienTrongLopHoc": [],          "thongBao": []      }  } |

#### **2.2.4. Xem danh sách thông báo theo lớp**

- Mô tả: Sinh viên xem những thông báo của lớp học phần đấy.

- Code Controller:

- Endpoint: /api/sinhvien/current/lophoc/{idLopHoc}/thongbao

- Resquest: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "id": 1,          "idlopHoc": 1,          "ngayTao": "2020-04-18T07:00:00",          "tieuDe": "Nghỉ lễ(Cập nhật 5/1/2020 10:03:08 AM)",          "noiDung": "Nghỉ học nha mấy em",          "idlopHocNavigation": {              "id": 1,              "idmonHoc": 1,              "idgiangVien": 1,              "idgiangVienNavigation": null,              "idmonHocNavigation": null,              "chiTietNgayHoc": [],              "sinhVienTrongLopHoc": [],              "thongBao": [                  {                      "id": 9,                      "idlopHoc": 1,                      "ngayTao": "2020-04-30T07:00:00",                      "tieuDe": "Nghỉ lễ",                      "noiDung": "Nghỉ nha"                  },                  {                   ...                  }              ]          }      },      {          ...      },  ] |

#### **2.2.5. Xem tất cả thông báo**

- Mô tả: Sinh viên xem danh sách tất cả những thông báo theo từng môn học.

- Code Controller:

- Endpoint: /api/sinhvien/current/lophoc/thongbao

- Resquest: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "tenMonHoc": "Công nghệ phần mềm",          "tenGiangVien": "LÊ VĂN DŨNG",          "thongBao": [              {                  "id": 1,                  "idlopHoc": 1,                  "ngayTao": "2020-04-18T07:00:00",                  "tieuDe": "Nghỉ lễ(Cập nhật 5/1/2020 10:03:08 AM)",                  "noiDung": "Nghỉ học nha mấy em",                  "idlopHocNavigation": null              },              {  ...              }          ]      },      {       ...      }  ] |

#### **2.2.6. Xem thời khoá biểu**

- Mô tả: Sinh viên xem thời khoá biểu ngày học, môn học cụ thể cùng với giảng viên và phòng học.

- Code Controller:

- Endpoint: /api/sinhvien/current/thoikhoabieu

- Resquest: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "tenMonHoc": "Công nghệ phần mềm",          "ngayHoc": "2020-03-09T07:00:00",          "tenGiangVien": "LÊ VĂN DŨNG",          "phongHoc": "A001"      },      {       ...      }  ] |

#### **2.2.7. Đăng kí khuôn mặt**

- Mô tả: Sinh viên thực hiện yêu cầu đăng kí khuôn mặt bắt buộc thông qua việc chụp ảnh khuôn mặt từ đấy có thể đối chiếu dữ liệu cho việc thực hiện điểm danh tại các lớp học. Số hình ảnh yêu cầu tối thiểu 10 thông qua việc mở camera để tiến hành chụp ảnh hay sử dụng ảnh có sẵn trong thư việc nhưng đảm bảo việc có xuất hiện khuôn mặt của mình.

- Code Controller:

- Endpoint: /api/khuonmats

- Resquest: + Phương thức: POST

+ Body: JSON

{

"MSSV":"SV00000002",

"base64ImagesStrings":["aaaaaabbbbb","ccccccccdddddd"]

}

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "Id": 2,      "Mssv": "SV00000002",      "MatKhau": "123456",      "HoTen": "LÊ VĂN MINH",      "NgaySinh": "1996-01-30T00:00:00",      "GioiTinh": true,      "DiaChi": "Tp. Hồ Chí Minh",      "SoDienThoai": "0974597417",      "Email": "levanminh@gmail.com",      "GhiChu": null,      "HasRegisteredFace": true,      "SinhVienTrongLopHoc": []  } |

#### **2.2.8. Cập nhật thông tin cá nhân**

- Mô tả: Sinh viên có thể cập nhật lại những thông tin cá nhân của mình nếu có sai sót. Chỉ được sửa những thông tin như ngày sinh, địa chỉ, số điện thoại, … Không được sửa các thông tin như Mã số sinh viên, họ tên, …

- Code Controller:

- Endpoint: /api/sinhvien/current

- Resquest: + Phương thức: PUT

+ Body: JSON

{

"NgaySinh":"1996-01-01",

"DiaChi":"Hà Nội",

… //{NgaySinh, DiaChi, SoDienThoai, GioiTinh, Email, GhiChu}

}

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 2,      "mssv": "SV00000002",      "matKhau": "123456",      "hoTen": "LÊ VĂN MINH",      "ngaySinh": "1996-01-01T00:00:00",      "gioiTinh": true,      "diaChi": "Hà Nội",      "soDienThoai": "0974597417",      "email": "levanminh@gmail.com",      "ghiChu": null,      "hasRegisteredFace": null,      "sinhVienTrongLopHoc": []  } |

#### **2.2.9. Cập nhật mật khẩu**

- Mô tả: Sinh viên có thể thay đổi mật khẩu của mình khi xác nhận chính xác mật khẩu cũ và nhập hai lần mật khẩu trùng khớp để việc đổi mật khẩu thành công.

- Code Controller:

- Endpoint: /api/sinhvien/{mssv}/matkhau

- Resquest: + Phương thức: PUT

+ Body: JSON

{

"MatKhauCu":"123456",

"MatKhauMoi":"654321",

}

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 2,      "mssv": "SV00000002",      "matKhau": "654321",      "hoTen": "LÊ VĂN MINH",      "ngaySinh": "1996-01-01T00:00:00",      "gioiTinh": true,      "diaChi": "Hà Nội",      "soDienThoai": "0974597417",      "email": "levanminh@gmail.com",      "ghiChu": null,      "hasRegisteredFace": null,      "sinhVienTrongLopHoc": []  } |

### **2.3. Chức năng giảng viên**

#### **2.3.1. Xem chi tiết lớp học**

- Mô tả: Giảng viên xem chi tiết danh sách sinh viên trong lớp học cũng như thời khoá biểu chi tiết theo từng môn học.

- Code Controller:

- Endpoint: /api/giangvien/current/lophoc

- Request: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "maLopHoc": 1,          "maMonHoc": 1,          "tenMonHoc": "Công nghệ phần mềm",          "giangVien": "GV00000001",          "sinhVienTrongLopHocs": [              {                  "mssv": "SV00000001",                  "tenSinhVien": "LÊ VĂN DŨNG"              },              {               ...              }          ],          "thoiKhoaBieu": [              {                  "id": 22,                  "idlopHoc": 1,                  "thoiGianBatDauBuoiHoc": "2020-03-09T07:00:00",                  "phongHoc": "A001",                  "idlopHocNavigation": null,                  "chiTietDiemDanh": []              },              {                  ...              }          ]      },      {          ...      }  ] |

#### **2.3.2. Xem danh sách điểm danh theo môn học**

- Mô tả: Giảng viên có thể tra cứu thông tin điểm danh của sinh viên theo ngày học cụ thể.

- Code Controller:

- Endpoint: /api/giangvien/chitietdiemdanh/{maNgayHoc}

- Request: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "mssv": "SV00000001",          "isDiemDanh": false      },      {          "mssv": "SV00000002",          "isDiemDanh": false      },      {          ...      }  ] |

#### **2.3.3. Điểm danh**

- Mô tả: Giảng viên tiến hành điểm danh thông qua việc chụp ảnh hay chọn ảnh từ thư viện với hình ảnh chứa nhiều khuôn mặt của sinh viên nhất có thể để việc điểm danh có thể diễn ra chính xác nhất.

- Code Controller:

- Endpoint: /api/giangvien/diemdanh/{maNgayHoc}

- Request: + Phương thức: POST

+ Body: JSON

|  |
| --- |
| {  "base64ImagesStringS":"aaaaabbbbbbbccccccc"  } |

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {  "danhSachKhuonMatTimThay": [  {  "externalID": "SV00000026",  "confidence": 0.51012  }  ],  "danhSachDiemDanh": [  {  "chiTietDiemDanh": {  "id": 148,  "idchiTietNgayHoc": 24,  "idsinhVienTrongLopHoc": 26,  "isDiemDanh": true,  "idchiTietNgayHocNavigation": null,  "idsinhVienTrongLopHocNavigation": null  },  "mssv": "SV00000026",  "isDiemDanh": true  },  {  "chiTietDiemDanh": {  "id": 149,  "idchiTietNgayHoc": 24,  "idsinhVienTrongLopHoc": 27,  "isDiemDanh": false,  "idchiTietNgayHocNavigation": null,  "idsinhVienTrongLopHocNavigation": null  },  "mssv": "SV00000027",  "isDiemDanh": false  },  {  ...  }  ]  } |

#### **2.3.4. Gửi thông báo đến lớp học**

- Mô tả: Giảng viên tạo thông báo để gửi đến các sinh viên thuộc lớp học mà thông báo được gửi đến.

- Code Controller:

- Endpoint: /api/giangvien/lophoc/{idLopHoc}/thongbao

- Request: + Phương thức: POST

+ Body: JSON

|  |
| --- |
| {  "TieuDe":"Không có",  "NoiDung":"Hôm nay lớp chúng ta nghỉ nha các em"  } |

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 15,      "idlopHoc": 1,      "ngayTao": "2020-05-11T09:38:45.4844599+07:00",      "tieuDe": "Không có",      "noiDung": "Hôm nay lớp chúng ta nghỉ nha các em",      "idlopHocNavigation": {          "id": 1,          "idmonHoc": 1,          "idgiangVien": 1,          "idgiangVienNavigation": null,          "idmonHocNavigation": null,          "chiTietNgayHoc": [],          "sinhVienTrongLopHoc": [],          "thongBao": []      }  } |

#### **2.3.5. Xem chi tiết thông báo theo lớp**

- Mô tả: Xem chi tiết thông báo đã gửi đến lớp học với nội dung, ngày gửi cùng như lớp học phần nhận thông báo.

- Code Controller:

- Endpoint: /api/giangvien/lophoc/{idLopHoc}/thongbao/{idThongBao}

- Request: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 1,      "idlopHoc": 1,      "ngayTao": "2020-04-18T07:00:00",      "tieuDe": "Nghỉ lễ(Cập nhật 5/1/2020 10:03:08 AM)",      "noiDung": "Nghỉ học nha mấy em",      "idlopHocNavigation": {          "id": 1,          "idmonHoc": 1,          "idgiangVien": 1,          "idgiangVienNavigation": null,          "idmonHocNavigation": null,          "chiTietNgayHoc": [],          "sinhVienTrongLopHoc": [],          "thongBao": []      }  } |

#### **2.3.6. Xem danh sách thông báo theo lớp**

- Mô tả: Xem danh sách tất cả các thông báo đã gửi theo từng lớp học.

- Code Controller:

- Endpoint: /api/giangvien/lophoc/{idLopHoc}/thongbao

- Request: + Phương thức: GET

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "id": 1,          "idlopHoc": 1,          "ngayTao": "2020-04-18T07:00:00",          "tieuDe": "Nghỉ lễ(Cập nhật 5/1/2020 10:03:08 AM)",          "noiDung": "Nghỉ học nha mấy em",          "idlopHocNavigation": {              "id": 1,              "idmonHoc": 1,              "idgiangVien": 1,              "idgiangVienNavigation": null,              "idmonHocNavigation": null,              "chiTietNgayHoc": [],              "sinhVienTrongLopHoc": [],              "thongBao": [                  {                      "id": 9,                      "idlopHoc": 1,                      "ngayTao": "2020-04-30T07:00:00",                      "tieuDe": "Nghỉ lễ",                      "noiDung": "Nghỉ nha"                  },                  {                      ...                  }              ]          }      },      {          ...      }  ] |

#### **2.3.7. Chỉnh sửa điểm danh**

- Mô tả: Giảng viên có thể tiến hành chỉnh sửa điểm danh nếu việc điểm danh qua ảnh chụp còn thiếu sót.

- Code Controller:

- Endpoint: /api/giangvien/diemdanh/{maNgayHoc}

- Request: + Phương thức: PUT

+ Body: JSON

|  |
| --- |
| [  {  "MSSV":"SV00000022",  "isDiemDanh":true  },  {  "MSSV":"SV00000023",  "isDiemDanh":true  }  ] |

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| [      {          "chiTietDiemDanh": {  "id": 148,  "idchiTietNgayHoc": 24,  "idsinhVienTrongLopHoc": 26,  "isDiemDanh": true,  "idchiTietNgayHocNavigation": null,  "idsinhVienTrongLopHocNavigation": null  },  "mssv": "SV00000022",  "isDiemDanh": true      },      {          "chiTietDiemDanh": {  "id": 148,  "idchiTietNgayHoc": 24,  "idsinhVienTrongLopHoc": 26,  "isDiemDanh": true,  "idchiTietNgayHocNavigation": null,  "idsinhVienTrongLopHocNavigation": null  },  "mssv": "SV00000023",  "isDiemDanh": true      },      {          ...      }  ] |

#### **2.3.8. Chỉnh sửa thông báo**

- Mô tả: Giảng viên có thể chỉnh sửa những thông báo đã gửi đi kèm với thời gian cập nhât thông báo cụ thể để giảng viên lẫn sinh viên dễ dàng theo dõi.

- Code Controller:

- Endpoint: /api/giangvien/lophoc/{idLopHoc}/thongbao/{idThongBao}

- Request: + Phương thức: PUT

+ Body: JSON

|  |
| --- |
| {  "TieuDe":"Không có",  "NoiDung":"Hôm nay chúng ta học lúc 15:00 nha các em"  } |

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 15,      "idlopHoc": 1,      "ngayTao": "2020-05-11T09:38:45.483",      "tieuDe": "Không có(Cập nhật 5/11/2020 9:43:56 AM)",      "noiDung": "Hôm nay chúng ta học lúc 15:00 nha các em",      "idlopHocNavigation": {          "id": 1,          "idmonHoc": 1,          "idgiangVien": 1,          "idgiangVienNavigation": null,          "idmonHocNavigation": null,          "chiTietNgayHoc": [],          "sinhVienTrongLopHoc": [],          "thongBao": []      }  } |

#### **2.3.9. Xoá thông báo**

- Mô tả: Giảng viên có thể xoá thông báo đã gửi đến các lớp học.

- Code Controller:

- Endpoint: /api/giangvien/lophoc/{idLopHoc}/thongbao/{idThongBao}

- Request: + Phương thức: DELETE

+ Authorization: Bearer Token

- Response: Status 200 OK

#### **2.3.10. Cập nhật thông tin cá nhân**

- Mô tả: Giảng viên có thể cập nhật lại những thông tin cá nhân của mình nếu có sai sót. Chỉ được sửa những thông tin như ngày sinh, địa chỉ, số điện thoại, … Không được sửa các thông tin như Mã số giảng viên, họ tên, …

- Code Controller:

- Endpoint: /api/giangvien/current

- Request: + Phương thức: PUT

+ Body: JSON

|  |
| --- |
| {  "DiaChi":"TP. Hồ Chí Minh",  "NgaySinh":"1979-01-30",  … //{NgaySinh, DiaChi, SoDienThoai, GioiTinh, Email, GhiChu}  } |

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 1,      "msgv": "GV00000001",      "matKhau": "123456",      "hoTen": "LÊ VĂN DŨNG",      "ngaySinh": "1979-01-30T00:00:00",      "gioiTinh": true,      "diaChi": "TP. Hồ Chí Minh",      "soDienThoai": "0976843167",      "email": "levandung@gmail.com",      "ghiChu": null,      "lopHoc": []  } |

#### **2.3.11. Cập nhật mật khẩu**

- Mô tả: Giảng viên có thể thay đổi mật khẩu của mình khi xác nhận chính xác mật khẩu cũ và nhập hai lần mật khẩu trùng khớp để việc đổi mật khẩu thành công.

- Code Controller:

- Endpoint: /api/giangvien/current/matkhau

- Request: + Phương thức: PUT

+ Body: JSON

|  |
| --- |
| {  "MatKhauCu":"123456",  "MatKhauMoi":"654321"  } |

+ Authorization: Bearer Token

- Response: JSON

|  |
| --- |
| {      "id": 1,      "msgv": "GV00000001",      "matKhau": "654321",      "hoTen": "LÊ VĂN DŨNG",      "ngaySinh": "0001-01-01T00:00:00",      "gioiTinh": true,      "diaChi": "TP. Hồ Chí Minh",      "soDienThoai": "0976843167",      "email": "levandung@gmail.com",      "ghiChu": null,      "lopHoc": []  } |

# **Chương 5: Testing**

## **1. Front-end**

|  |  |
| --- | --- |
| Thực hiện | Kết quả |
| - Chức năng tạo thông báo gửi đến lớp học:  + Chưa thực hiện nhập nội dung của thông báo dẫn đến chương trình báo lỗi yêu cầu nhập vào nội dung của thông báo. | Hình 5.1: Testing chức năng gửi thông báo |
| -Chức năng đăng nhập:  + Việc đăng nhập thất bại.  + Có những lí do sau:  . Không nhập đúng tài khoản hoặc mật khẩu.  . Đăng nhập không đúng theo phần quyền của tài khoản. | A screenshot of a cell phone  Description automatically generated  Hình 5.2: Testing chức năng Đăng nhập |
| - Chức năng điểm danh:  + Điểm danh thất bại  + Có những lí do sau:  . Ảnh không hợp lệ  . Ảnh không đúng định dạng  . Không có khuôn mặt trong ảnh  … | A screenshot of a cell phone  Description automatically generated  Hình 5.3: Testing chức năng Điểm danh |
| -Chức năng cập nhật mật khẩu:  + Không thể tiến hành thay đổi mật khẩu vì nhập sai mật khẩu cũ. | A screenshot of a cell phone  Description automatically generated  Hình 5.4: Testing chức năng Cập nhật mật khẩu |

## **2. Back-end**

- Chức năng điểm danh:

+ Điểm danh thất bại

+ Có những lí do sau:

. Ảnh không hợp lệ

. Ảnh không đúng định dạng

. Không có khuôn mặt trong ảnh

…

A screenshot of a social media post

Description automatically generated

Hình 5.5: Testing chức năng Điểm danh

- Chức năng Học khuôn mặt – đăng kí khuôn mặt của sinh viên

+ Đăng kí khuôn mặt thất bại:

+ Có những lí do sau:

. Ảnh không lệ

. Ảnh không đúng định dạng

. Không có khuôn mặt trong ảnh

A screenshot of a social media post

Description automatically generated

Hình 5.6: Testing chức năng Đăng kí khuôn mặt

# **Chương 6: Kết luận và hướng phát triển**

## **1. Kết luận**

- Thu nhận thêm những kiến thức cần thiết làm tiền đề cho việc tự học tập thêm những kiến thức mới, công nghệ mới trong tương lai.

- Học được cách sử dụng SDK Cognitive Service dành cho C#.

- Nắm rõ nguyên lý triển khai ứng dụng trên những môi trường khác nhau, nền tảng khác nhau.

- Tích lũy những trải nghiệm tốt từ đó có thêm kinh nghiệm về vấn đề làm việc nhóm, thực hiện đồ án (nâng cao kĩ năng mềm) cũng như việc chia sẽ kiến thức dễ dàng giữa các thành viên. Học tập được những thứ hay từ việc thực hiện đồ án như kĩ năng tìm kiếm thông tin, xây dựng tài liệu, những kĩ năng về lập trình, …

## **2. Hướng phát triển**

- Với việc dự án sử dụng một hệ CSDL phần lớn là dựng sẵn nên vẫn cần phải phát triển thêm. Để có thể đem ra sử dụng với việc phát triển import, quản lí dữ liệu, cũng như khả năng bảo mật bên cạch việc xử lý những vấn đề phát sinh, …

- Cải thiện độ chính xác của các chức năng nhận diện khuôn mặt.

- Tối ưu giao diện để thân thiện hơn với người dùng.

- Mở rộng thêm với phân quyền cho vai trò quản lí (admin) giúp dễ dàng hơn trong việc quản lý sinh viên, giảng viên bên cạnh việc quản lí về tiến trình giảng dạy và học tập ở các môn học.

- Tích lũy thêm những kiến thức nghiệp vụ tốt hơn để phần nào hoàn thiện hơn về sản phẩm. Cải thiện hơn những tính năng cũ, bổ sung thêm những tình năng cần thiết để ứng dụng trở nên tốt hơn và giúp nó trở nên tiện ích. Đưa sản phẩm ra sử dụng rộng rãi.

# **TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Tìm hiểu về Microsoft Cognitive Services – Sức mạnh của Machine Learning – Lion Pham - <https://blog.lionpham.com/2016/08/27/microsoft-cognitive-services-api/>

**Tiếng Anh**

<https://docs.microsoft.com/vi-vn/azure/cognitive-services/Face/Quickstarts/client-libraries?pivots=programming-language-csharp&fbclid=IwAR1Ge0m6eGZuTRerXy1To6nZ6wXQEYg6s_eRHyhCpYelZqPQ7B0VplpUMOs&tabs=windows#identify-a-face>