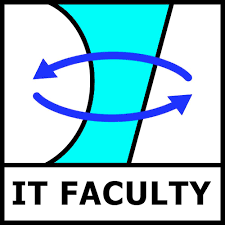
******ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**ĐỒ ÁN CƠ SỞ NGÀNH MẠNG**

**ĐỀ TÀI:**

**Phần Nguyên lí hệ điều hành*: Xây dựng chương trình client tương tác với Web API***

**Phần Lập trình mạng: *Cơ chế đa luông hệ điều hành.***

**GVHD: PGS.TS Nguyễn Tấn Khôi**

**SVTH1: Nguyễn Phước Nhâm**

**MSSV: 102202222**

**Lớp: 20TCLC\_DT5**

**SVTH1: Nguyễn Trần Thảo Vy**

**MSSV: 102202121**

**Lớp: 20TCLC\_DT2**

***Đà Nẵng, 12/2022***

**NHẬN XÉT CỦA GIẢNG VIÊN**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………..**

**…………………………………………………………………………………………**

MỤC LỤC

[MỤC LỤC 3](#_Toc122687328)

[PHẦN I: NGUYÊN LÝ HỆ ĐIỀU HÀNH 4](#_Toc122687329)

[PHẦN II: LẬP TRÌNH MẠNG 4](#_Toc122687330)

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 4](#_Toc122687331)

[*1.* NỘI DUNG ĐỀ TÀI 4](#_Toc122687332)

[*2.* MỤC TIÊU 4](#_Toc122687333)

[3. MÔI TRƯỜNG SỬ DỤNG 4](#_Toc122687334)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5](#_Toc122687335)

[1. REST API: 5](#_Toc122687336)

[1.1. Khái quát: 5](#_Toc122687337)

[1.2. Hai thành phần trong REST API 5](#_Toc122687338)

[1.3. Request và Response trong REST API 6](#_Toc122687339)

[1.4. Status Code 7](#_Toc122687340)

[2. HTTP/HTTPs 8](#_Toc122687341)

[1.1. Khái quát 8](#_Toc122687342)

[1.2. HTTP request và response 8](#_Toc122687343)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc122687344)

[*1.* BÀI TOÁN VÀ YÊU CẦU 9](#_Toc122687345)

[1.1. Bài toán: 9](#_Toc122687346)

[*2.* THIẾT KẾ CHƯƠNG TRÌNH 10](#_Toc122687347)

[1.2. Server API 10](#_Toc122687348)

[1.3. Client- winform 15](#_Toc122687349)

[CHƯƠNG 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 19](#_Toc122687350)

[1. Giao diện 19](#_Toc122687351)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 23](#_Toc122687352)

[1. Ý nghĩa đề tài: 23](#_Toc122687353)

[2. Hướng phát triển: 23](#_Toc122687354)

1. NGUYÊN LÝ HỆ ĐIỀU HÀNH

Đề tài: Ứng dụng cơ chế đồng bộ các process cho bài toán

1. LẬP TRÌNH MẠNG

Đề tài: Xây dựng chương trình client tương tác với Web API.

# GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## NỘI DUNG ĐỀ TÀI

Xây dựng chương trình quản lí học sinh, bài học của trung tâm dạy lập trình cho trẻ em.

## MỤC TIÊU

Tìm hiểu cơ chế hoạt động của Winform, sử dụng thư viện để gọi Rest API, tìm hiểu giao thức HTTP, REST API.

## MÔI TRƯỜNG SỬ DỤNG

Chương trình client: chương trình Winform được viết bằng ngôn ngữ C#.

Chương trình server: WebAPI dùng framework của C# (ABP framework).

Trong đồ án môn học này, em sử dụng phần mềm Visual Studio, Visual Studio Code để phát triển ứng dụng.

# CƠ SỞ LÝ THUYẾT

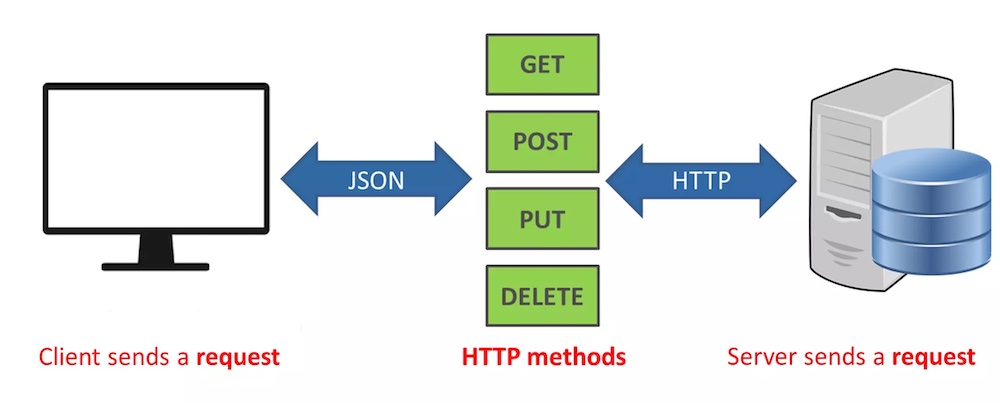
Nhờ sự phát triển của công nghệ thông tin nói chúng và Internet nói riêng, nhiều công việc, tác vụ của con người đã có thể thực hiện một cách dễ dàng hơn. Trong đó, vấn đề quản lí dữ liệu dạy học của các trung tâm dạy học bằng các công cụ truyền thống như excel, word,.. đang được chuyển dần thành các ứng dụng cụ thể cho từng trung tâm.

Trong bài đồ án này sẽ tập trung tìm hiểu về một ứng dụng desktop trong quản lí thông tin của một trung tâm dạy học: học sinh, các lớp học, bài học,… bằng Winform.

## REST API:

### Khái quát:

REST API (còn được biết với tên gọi RESTful API) là một giao diện lập trình ứng dụng (API) tuân thủ các ràng buộc và quy ước kiến trúc REST được sử dụng trong việc giao tiếp giữa client và server. REST là viết tắt của REpresentational State Transfer, nó được tạo ra bởi nhà khoa học máy tính Roy Fielding.



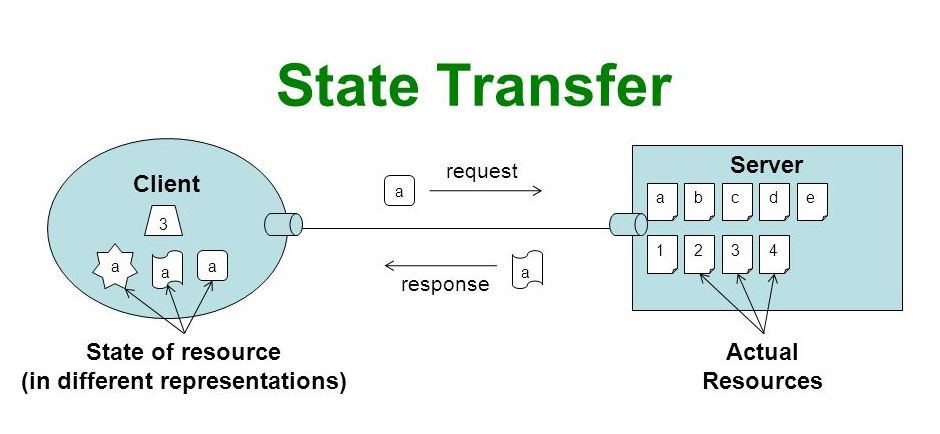
1. Rest API – Giao tiếp client và server

REST API thường vẫn sử dụng giao thức HTTP/1 kèm theo các định nghĩa trước đó mà cả client và server cần tuân thủ.

Hiện tại, các REST API dùng JSON rất phổ biến. Một ít có thể vẫn còn dùng XML.

### Hai thành phần trong REST API

**REST** (REpresentational State Transfer) là một đại diện cho sự chuyển đổi dữ liệu. Trong kiến trúc này client và server hoàn toàn độc lập, chúng không biết gì về nhau. Mỗi một request REST API đều không mang theo trạng thái trước đó (stateless). Như vậy để đôi bên trao đổi state, chúng sẽ phải thông qua các resources. Các resource này chính là phần đại diện cho sự thay đổi dữ liệu.

**API** (Application Programming Interface) là giao diện lập trình ứng dụng. Giao diện này không dành cho người dùng cuối mà dành cho các nhà phát triển (developer). Nó là cái “bề mặt”, chỉ thấy được phần khai báo (tên, tham số, kiểu trả về. 

1. Quá trình chuyển email

### Request và Response trong REST API

#### Phương thức

Như đã đề cập ở trên, để trao đổi state chúng sẽ cần giao tiếp resource thông qua việc gởi các request response thông qua HTTP/1. Cụ thể việc giao tiếp này là thế nào thì chúng cần chỉ định các method tương ứng bao gồm:

**GET**: Trả về một Resource hoặc một danh sách Resource.

**POST**: Tạo mới một Resource.

**PUT**: Cập nhật thông tin cho Resource (toàn bộ resource).

**PATCH**: Cật nhật thông tin cho resourse (một phần resource).

**DELETE**: Xoá một Resource.

#### Header: Authentication và quy định kiểu dữ liệu trả về

Mỗi một request không hề biết bất kỳ thông tin gì trước đó. Khác với khi chúng ta truy cập web, trình duyệt sẽ có session và cookie để hỗ trợ phân biệt request đấy là của ai, thông tin trước đó là gì.

Trong REST, nếu một request cần xác thực quyền truy cập, chúng sẽ phải dùng thêm thông tin trong header. Ví dụ như thông tin Authorization sẽ mang theo một user token. Hiện có 3 cơ chế Authentication chính:

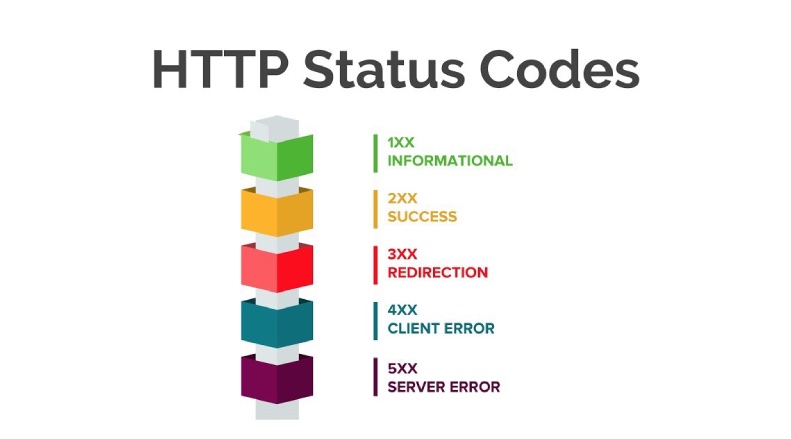
* HTTP Basic
  + - * JSON Web Token (JWT)
      * OAuth2

Ngoài ra Header còn giúp client chỉ định được loại content cần trả về từ server – content type. Việc này được thực hiện thông qua phần Accept trong header. Giá trị của nó thường là MIME type:

* image — image/png, image/jpeg, image/gif
* audio — audio/wav, audio/mpeg
* video — video/mp4, video/ogg
* application — application/json, application/pdf, application/xml, application/octet-stream

### Status Code

Response trong REST API sẽ bao gồm một status code quy định cụ thể từng trường hơp.



1. Ý nghĩa status code

Một số status phổ biến:

* 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
* 201 Created – Trả về khi một Resouce vừa được tạo thành công.
* 204 No Content – Trả về khi Resource xoá thành công.
* 304 Not Modified – Client có thể sử dụng dữ liệu cache, resource server không đổi gì.
* 400 Bad Request – Request không hợp lệ
* 401 Unauthorized – Request cần có xác thực.
* 403 Forbidden – bị từ chối không cho phép.
* 404 Not Found – Không tìm thấy resource từ URI
* 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
* 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
* 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.
* 422 Unprocessable Entity – Dữ liệu không được xác thực
* 429 Too Many Requests – Request bị từ chối do bị giới hạn

## HTTP/HTTPs

### Khái quát

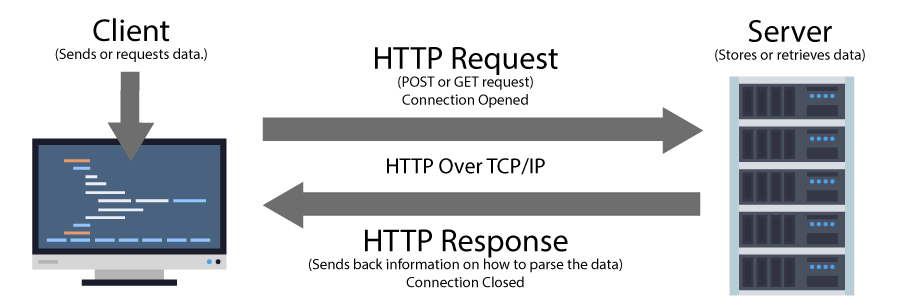
**HTTP** (HyperText Transfer Protocol) là giao thức truyền tải siêu văn bản được sử dụng trong www dùng để truyền tải dữ liệu giữa Web server đến các trình duyệt Web và ngược lại. Giao thức này sử dụng cổng 80 (port 80) là chủ yếu.

**HTTPS** (HyperText Transfer Protocol Secure) là giao thức Http có sử dụng thêm SSL (Secure Sockets Layer) để mã hóa dữ liệu trong lúc truyền tải dữ liệu nhầm gia tăng thêm tính an toàn cho việc truyền dữ liệu giữa Web server và trình duyệt Web. Giao thức Https thì sử dụng cổng 433 để truyền dữ liệu

### HTTP request và response

Sử dụng HTTP hoặc HTTPS để giao tiếp đọc thông tin từ máy chủ Web hoặc gửi thông tin đến máy chủ Web bằng trình duyệt Web.

* Giao tiếp HTTP được thực thi khi máy khách gửi một request đến máy chủ và máy chủ xử lý request rồi trả về một response cho nó.
* Các request và response HTTP về cơ bản được trao đổi dưới dạng văn bản. Ngoài thông tin dạng văn bản, thông tin nhận được từ máy chủ cũng có thể lưu trữ dữ liệu nhị phân như là hình ảnh...



1. HTTP request và response

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## BÀI TOÁN VÀ YÊU CẦU

### Bài toán:

Xây dựng ứng dụng desktop trong quản lí thông tin của một trung tâm dạy học: học sinh, các lớp học, bài học,… bằng Winform.

* Yêu cầu:
  + Quản lí user:
    - Admin
    - Giáo viên
  + Quản lí học sinh:
    - Thông tin học sinh, phụ huynh: Thông tin học sinh, phụ huynh.
    - Lớp học đang tham gia: Bài học đã học, thông tin đánh giá.
  + Quản lí Khoá học:
    - Thông tin Khoá học: Tên, mô tả, số buổi học, danh sách bài học.
    - Các chức năng CRUD cơ bản.
    - Một khoá học có thể dung cho nhiều lớp học
  + Quản lí lớp học:
    - Thông tin lớp học: tên, thời gian, khoá học, học phí…
    - Các chức năng CRUD cơ bản.
    - Mỗi lớp học thuộc 1 kì học.
    - Mỗi lớp học sử dụng 1 khoá học
  + Quản lí kì học:
    - Một kì học sẽ có nhiều lớp học
    - Thông tin lớp học: thời gian, tên kì học
  + Quản lí buổi học:
    - Một lớp học có nhiều học sinh của nhiều lớp học khác nhau.
    - Tạo 1 buổi học: Chọn thời gian => chọn học sinh => chọn bài học cho học sinh.

## THIẾT KẾ CHƯƠNG TRÌNH

### Server API

* 1. Sơ đồ trình tự tổng quát

Diagram

Description automatically generated

1. Sơ đồ trình tự tổng quát

1. Người dung truy cập vào ứng dụng winform, thực hiện đăng nhập.

2. Client-winform sẽ thực hiện gọi 1 yêu cầu đăng nhập đến server.

3. Nếu đăng nhập thành công server sẽ trả về 1 token. Dựa vào token này để định danh và phân quyền người dùng.

4. Người dùng thực hiện 1 hành động (thêm, sửa, xoá, lấy dữ liệu).

5. Client-winform gọi 1 API tương ứng với hành động của người dùng đến server.

6. Server thực hiện lấy dữ liệu hoặc cập nhật dữ liệu trên database sau đó trả về cho client kết quả(thường là 1 json, HTTP code)

7. Client nhận phản hồi từ server và hiển thị ra màn hình cho người dùng.

#### Cấu trúc project

Diagram

Description automatically generated

Server sử dụng ABP framework. Cầu trúc solution gồm:

* + **.Domain.Shared**: Chứa các hằng số, enum,.. dùng chung cho cả solution, tất cả các phần đều có thể truy cập được đến nó.
  + **.Domain**: Chứa các entity của dự án, các interface của repository. Chỉ .Application có thể gọi đến domain.

Text

Description automatically generated

1. Class entity
   * **.EntityFrameworkCore**: triển khai các interface được khai báo ở Domain. Các thao tác truy cập đến database được sử dụng ở đây thông qua dbcontext
   * Text

     Description automatically generated
2. Class repository
   * **.Application.Contract**: chứa các DTO(Data Transfer Object), các interface service.

Text

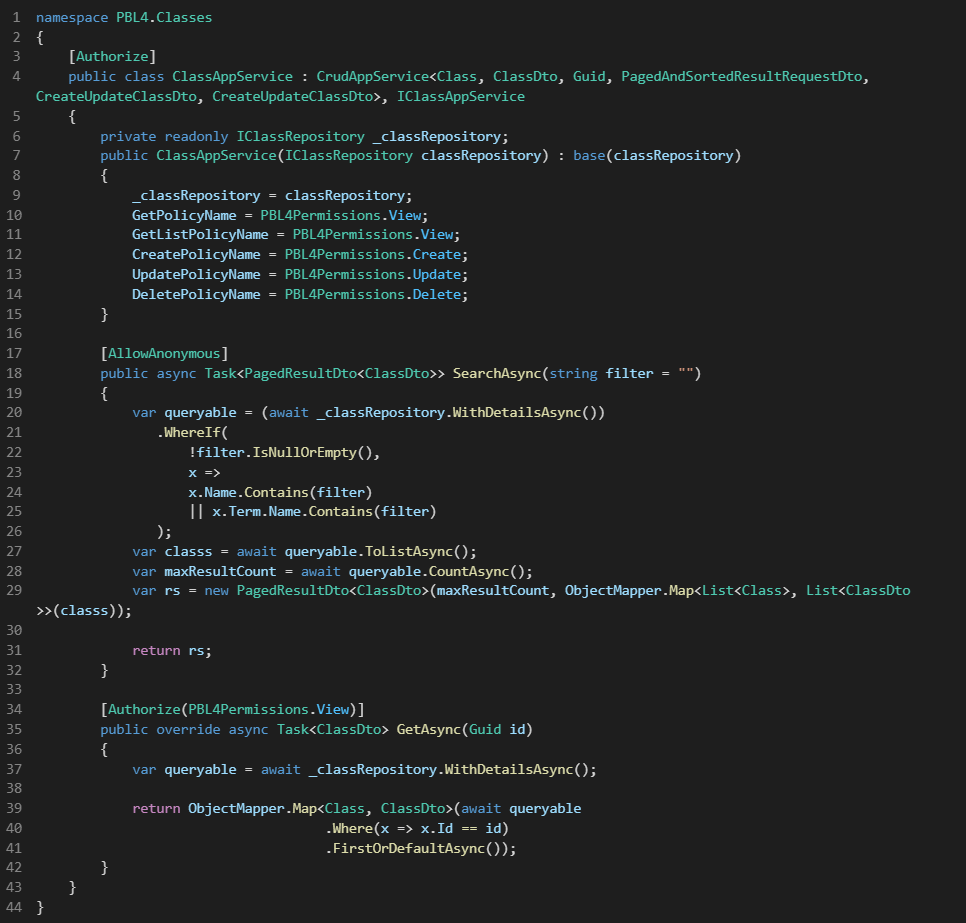
Description automatically generated

1. Class DTO

Text

Description automatically generated

1. Danh sách các permission

**.Application**: triển khai các interface được khai báo ở .Application.Contract. Chứa các service tương ứng với các API cần thiết. Gọi đến các repository được triển khai ở EntityFrameworkCore thông qua interface ở .Domain (sử dụng dependence injection pattern) để thực hiện lấy dữ liệu hoặc cập nhật dữ liệu.

1. ClassAppService
   * .**Web:** Chứa các User Interface của web, appsettings(chứa chuỗi kết nối và các cấu hình cần thiết).

#### Thiết kế cơ sở dử liệu

Graphical user interface, application, Word

Description automatically generated

#### Xử lí đăng nhập và phân quyền

Sử dụng user đăng nhập mặc định của ABP Frame work và phân quyền có sẵn

#### File appsettings.json

Text

Description automatically generated

* + Địa chỉ của Server được chạy ở local: <https://localhost:44313>
  + Chuỗi kết nối CSDL: "Server=localhost;Database=PBL4;Trusted\_Connection=True"
  + Chuỗi xác thực người dung: "https://localhost:44313"

#### Deployment

Sử dụng Docker

Docker file:

Text

Description automatically generated

Server sử dụng DigitalOcean: Server được deploy ở địa chỉ <http://143.198.84.239/>

### Client- winform

#### Các thư mục cần thiết

* + **BaseFunction**: chứa template class để gọi API
    - Text

      Description automatically generated

1. Tempalte class GetAPIByService để gọi API

**Token** được lấy lúc đăng nhập và lưu ở biến ConfigManger.Token:

Text

Description automatically generated

1. Get Token from server
   * **Dto**: chứa các entity.
   * **SdkCommon:** chứacác interface tương ứng với các function để gọi API

**Text

Description automatically generated**

1. Interface trong SDK common
   * Ví dụ gọi 1 API:

Text

Description automatically generated

1. Gọi API

classApi.GetAsync(id) tương ứng với API có url: “BaseUrl/api/app/class/{id}”

classDto sẽ chứa các thông tin của class có id là id. Sau đó hiển thị các thông tin lên view.

* + Xử lí ngoại lệ
  + Text

    Description automatically generated

1. Xử lí ngoại lệ
   * + Khi khởi tạo chương trình gọi

**Application.SetUnhandledExceptionMode(UnhandledExceptionMode.CatchException);**

để hiện thông báo khi có lỗi xảy ra

# TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

## Giao diện

Graphical user interface

Description automatically generated

1. Giao diện login

Table

Description automatically generated

1. Giao diện chính

Graphical user interface, application

Description automatically generated

1. Chi tiết, chỉnh sửa thông tin học sinh

Graphical user interface, application

Description automatically generated

1. Chi tiết khoá học

Graphical user interface, application

Description automatically generated

1. Chi tiết bài học

Graphical user interface

Description automatically generated

1. Chi tiết lớp học

Graphical user interface, application

Description automatically generated

1. Chi tiết buổi học

Graphical user interface, text, application, email

Description automatically generated

1. Danh sách các API

Graphical user interface, application

Description automatically generated

1. Quản lí quyền hạn

# 

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Ý nghĩa đề tài:

Áp dụng các kiến thức tìm hiểu được về API để tạo nên một ứng dụng có thể sử dụng được.

Tìm hiểu và sử dụng ABP Framework

## Hướng phát triển:

Tham khảo thêm cách phân quyền của các trung tâm rồi áp dụng cho chương trình.

Deploy ứng dụng và Server để có thể sử dụng trong thực tế.

TÀI LIỆU THAM KHẢO

[1] Tìm hiểu API <https://aws.amazon.com/vi/what-is/api/>

[2] Tìm hiểu ABP: <https://docs.abp.io/en/abp/latest>

[3] Tìm hiểu Refit: https://code-maze.com/using-refit-to-consume-apis-in-csharp/