TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN**

**CHUYÊN ĐỀ LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**

**ORDERS APP**

*Người hướng dẫn*: **ThS. VŨ ĐÌNH HỒNG**

*Người thực hiện*: **NGUYỄN DUY THANH – 51403318**

**TRẦN ĐẠI NHÂN – 51403028**

Lớp **: 14050302**

Khoá  **: 18**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN**

**CHUYÊN ĐỀ LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**

**ORDERS APP**

Người hướng dẫn: **ThS. VŨ ĐÌNH HỒNG**

Người thực hiện: **NGUYỄN DUY THANH**

**TRẦN ĐẠI NHÂN**

Lớp **: 14050302**

Khoá  **: 18**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

LỜI CẢM ƠN

Chúng em xin chân thành gửi lời cảm ơn tới toàn thể thầy, cô phụ trách bộ môn chuyên đề lập trình trên thiết bị di động, đã mang đến cho chúng em những kiến thức nền tảng cơ bản để ứng dụng vào bài tập lớn và đồ án môn học này.Chúc em xin kính chúc sức khỏe và lời cảm ơn trân trọng đến tất cả thầy cô.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của ThS. Vũ Nguyên Hồng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 4 tháng 5 năm 2017*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Duy Thanh*

*Trần Đại Nhân*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Để làm được đề tài này chúng em đã tìm hiểu ASP.NET Core để viết phần backend và tìm hiểu thêm một số cách công cụ giao diện có sẵn trong Android để hiện thực giao diện phía fontend, và một một số cách hiệu chỉnh giao diện.

Ứng dụng hoàn chỉnh, tất cả các danh mục, món ăn đều lấy được từ phía webserver. Ứng dụng chia làm 2 giao diện cho phía người dùng và giao diện phía người quản lý.

Giao diện phía người quản lý sẽ có các chức năng như sau: đăng nhập, thêm, xóa, sửa danh mục và món ăn, kiểm tra đơn hàng.

Giao diện khách hàng sẽ thực hiện các chức năng như sau: chọn danh mục, chọn món ăn, đặt mua sản phẩm.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc481787542)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc481787543)

[TÓM TẮT iv](#_Toc481787544)

[MỤC LỤC 1](#_Toc481787545)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc481787546)

[CHƯƠNG 1 – TỔNG QUAN 4](#_Toc481787547)

[1.1 Framework ASP.NET Core 4](#_Toc481787548)

[1.1.1 ASP.NET Core là gì? 4](#_Toc481787549)

[1.1.2 Đặc điểm riêng biệt của ASP.Net Core 4](#_Toc481787550)

[1.1.3 Tại sao ta nên sử dụng ASP.Net Core? 5](#_Toc481787551)

[1.2 JSON 5](#_Toc481787552)

[1.2.1 Json là gì? 5](#_Toc481787553)

[1.2.2 Tại sao ta nên sử dụng JSON. 6](#_Toc481787554)

[1.3 AsyncTack 7](#_Toc481787555)

[1.3.1 Khái niệm về đa luồng 7](#_Toc481787556)

[1.3.2 Tại sao chúng ta nên sử dụng AsynTack 8](#_Toc481787557)

[1.3.3 Một số phương thức cơ bản trong AsynTask 9](#_Toc481787558)

[1.3.3.1 onPreExecute 9](#_Toc481787559)

[1.3.3.2 doInBackground 9](#_Toc481787560)

[1.3.3.3 onPostExecute 9](#_Toc481787561)

[CHƯƠNG 2 – GIAO DIỆN PHẦN MỀM 12](#_Toc481787562)

[2.2 Giao diện danh sách món ăn 12](#_Toc481787563)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

**DANH MỤC BẢNG**

CHƯƠNG 1 – TỔNG QUAN

* 1. Framework ASP.NET Core

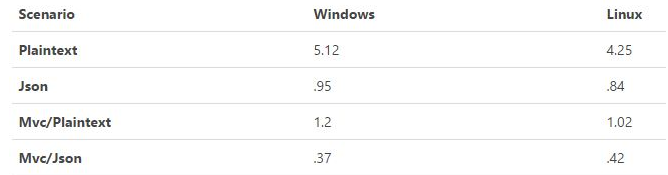
1.1.1 ASP.NET Core là gì?

ASP.NET core là một mã nguồn mở và là nền tảng mới cho xây dựng cloud trên internet kết nối các ứng dụng web, loT và mobile backends. ASP.NET Core có thể chạy trên .NET Core hoặc chạy đầy đủ trên .NET Framework. ASP.NET Core được kiến trúc để cung cấp một nền tảng phát triển tối ưu cho các ứng dụng được triển khai tới cloud hoặc chạy on-premises. ASP.NET Core bao gồm các thành phần mô-đun cần thiết tối thiểu, do đó bạn giữ lại được tính linh hoạt trong khi xây dựng các solution của bạn. Bạn có thể phát triển và chạy ASP.NET Core trên Windows, MAC và LINUX. ASP.NET Core là mã nguồn mở tại [GitHub](https://github.com/aspnet/home).

1.1.2 Đặc điểm riêng biệt của ASP.Net Core

Hỗ trợ đa nền tảng: bạn có thể phát triển và chạy ASP.NET trên cả Windows, Mac, và Linux. Mã nguồn mở rộng.

Hiệu suất: Microsoft giới thiệu máy chủ mới web Kestrel chạy trong host IIS của bạn hoặc chạy sau một host process khác. Kestrel hiện tại là máy chủ .NET chạy nhanh nhất hiện nay.



Hình 1.1: Bảng hiệu xuất đánh giá từ Microsoft.

1.1.3 Tại sao ta nên sử dụng ASP.Net Core?

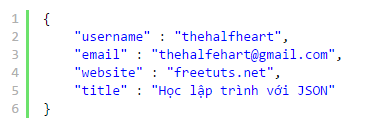
ASP.NET Core có một số thay đổi kiến trúc làm cho nó gọn nhẹ hơn. ASP.NET Core không còn dựa trên System.Web.dll mà dựa trên tập hợp các granular và các NuGet. Nhờ thế cho phép bạn tối ưu hóa các ứng dụng của bạn nhờ chỉ cần sử dụng các gói NuGet mà bạn cần. Ngoài ra nó bảo mật hơn, cải thiện hiệu suất và giảm chi phí nhờ bạn chỉ cần chỉ trả cho những gì bạn sử dụng mà thôi.

* 1. JSON

1.2.1 Json là gì?

**JSON** là chữ viết tắt của (**J**avascript **O**bject **N**otation) , đây là một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được, bạn có thể sử dụng lưu nó vào một file, một record trong CSDL rất dễ dàng. JSON có định dạng đơn giản, dễ dàng sử dụng và truy vấn hơn XML rất nhiều nên tính ứng dụng của nó hiện nay rất là phổ biến, theo tôi thì trong tương lai tới trong các ứng dụng sẽ sử dụng nó là đa số.

Ví dụ dưới đây mình định nghĩa một chuỗi JSON lưu trữ thông tin cá nhân của mình như sau, đây là ví dụ quan trọng nhất của phần  **json là gì** này.



Hình 1.2: Ví dụ về chuỗi JSON

Như vậy cú pháp của JSON rất đơn giản là mỗi thông tin dữ liệu sẽ có 2 phần đó là **key và value,** điều này tương ứng trong CSDL là tên field và giá trị của nó ở một record nào đó. Tuy nhiên nhìn qua thì đơn giản nhưng nếu ta mổ xẻ nó ra thì có một vài điều như sau:

Chuỗi JSON được bao lại bởi dấu ngoặc nhọn {}

Các key, valuecủa JSON bắt buộc phải đặt trong dấu nháy kép {"}, nếu bạn đặt nó trong dấu nháy đơn thì đây không phải là một chuỗi JSON đúng chuẩn. Nên trường hợp trong value của bạn có chứa dấu nháy kép thì hãy dùng dấu (\) để đặt trước nó nhé, ví dụ học \"json là gì\" tại website freetuts.net.

Nếu có nhiều dữ liệu (nhiều cặp key => value) thì ta dùng dấu phẩy (,) để ngăn cách

Các key của JSON bạn nên đặt chữ cái không dấu hoặc số, dấu \_ và không có khoảng trắng., ký tự đầu tiên không nên đặt là số. Điều này rất giống với nguyên tắc đặt tên [biến trong PHP.](https://freetuts.net/khai-bao-bien-va-hang-so-trong-php-2.html)

1.2.2 Tại sao ta nên sử dụng JSON.

Lưu trữ dữ liệu đơn thuần. Đó là khi bạn muốn lưu trữ dữ liệu dưới dạng metadata ở phía server. Chuỗi JSON sẽ được lưu vào database và sau đó khi cần dữ liệu thì sẽ được giải mã. Ví dụ với PHP, cung cấp các hàm liên quan đến JSON để mã và giải mã là json\_encode và json\_decode.  
 Chú ý: phương pháp này cũng tương tự như sử dụng tính năng serialize và unserialize của PHP. Nhưng trong khi serialize và unserialize sử dụng với cả dữ liệu và biến, tức là phụ thuộc vào ngôn ngữ lập trình là PHP và dĩ nhiên không thể transfer sang ngôn ngữ lập trình khác để unserialize được. Vì vậy, nếu dữ liệu của bạn chỉ đơn thuần là dữ liệu cơ bản (chuỗi kí tự, số…) thì bạn hoàn toàn không nên sử dụng serialize mà nên sử dụng JSON.  
 Sử dụng JavaScript, ActionScript để xử lý thông tin trả về từ phía server. Rất nhanh và rất dễ dàng.

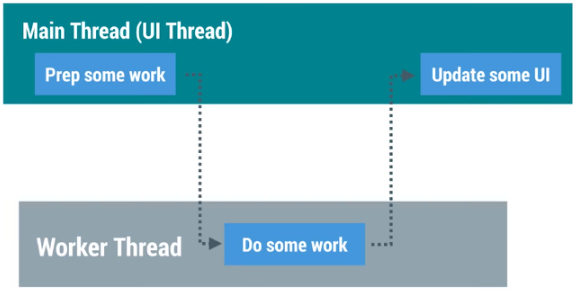
* 1. AsyncTack
     1. Khái niệm về đa luồng

MultiThreading là khái niệm dùng để nhắc đến những tác vụ chạy song song với nhau và giúp quá trình tương tác của người dùng với ứng dụng không bị gián đoạn. Khi một ứng dụng được start lên sẽ có một Thread được gọi là Main Thread hay UI Thread được chạy. Từ Thread này chúng ta có thể tạo ra những Thread khác để làm những tác vụ chiếm nhiều thời gian. Nếu chúng ta cố tình xử lý những tác vụ này trên UI Thread mà không tạo ra Thread khác có thể dẫn tới làm đơ UI và crash ứng dụng.

Những trường hợp nên sử dụng MultiThreading.

* Thao tác với hệ thống mạng (network).
* Đọc, ghi file.
  + 1. Tại sao chúng ta nên sử dụng AsynTack

Trong Android, khi chúng ta tạo một Thread để xử lý một tác vụ nào đó trong phương thức run() và gọi phương thức start() để chạy tác vụ này. Thì trong phương thức run chúng ta không được update các giá trị trực tiếp lên UI mà phải gửi message về UI Thread để UI Thread update các giá trị đó lên UI. Nếu bạn cố tình thì sẽ application sẽ ném ra một ngoại lệ.



Hình 1.3: Ví dụ mô hình làm việc MultiThreading trong Android.

* + 1. Một số phương thức cơ bản trong AsynTask

**1.3.3.1 onPreExecute**

Là phương thức được chạy đầu tiên khi task được bắt đầu. Phương thức này làm việc trên UI Thread.

1.3.3.2 doInBackground

Là phương thức chạy background thực hiện tác vụ nào đó. Phương thức này chạy trên Background Thread vì thế chúng ta không được update giá trị lên UI trong phương thức này.

1.3.3.3 onPostExecute

Là phương thức được gọi ngay sau khi phương thức doInBackground kết thúc. Đối số của phương thức này chính là kết quả return từ phương thức doInBackground. Phương thức này làm việc trên UI Thread.

Ngoài ra còn có thêm 2 phương thức khác nữa như:

**- publishProgress:** Được sử dụng trong phương thức doInBackground để cập nhật quá trình làm việc lúc runtime và gọi phương thức onProgressUpdate trên UI Thread để update giá trị trong khi doInBackground vẫn làm việc.

**- onProgressUpdate:** Update giá trị lên UI, được gọi nếu phương thức **publishProgress** được gọi trong phương thức **doInBackground**. Phương thức này làm việc trên UI Thread.

Ba đối số mà chúng ta tạo class extends từ AsyncTask là:

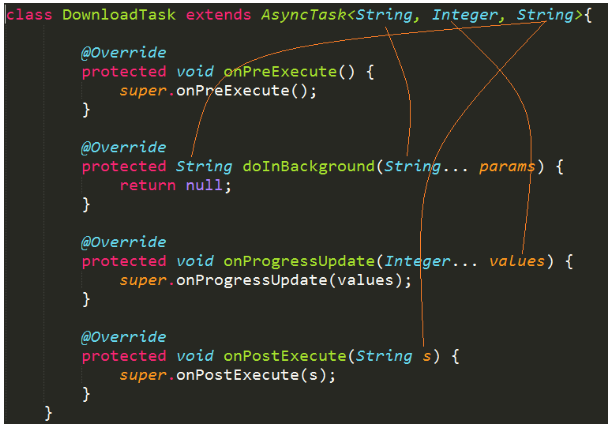
Vd: public class MyThread extends AsyncTask<Params, Progress, Result>

**- Param**: là kiểu của đối số được truyền vào phương thức doInBackground.

**-Progress:** là kiểu biến dùng để update giá trị trong quá tình **doInBackground** chạy. Giá trị này được truyền vào phương thức **onProgressUpdate.**

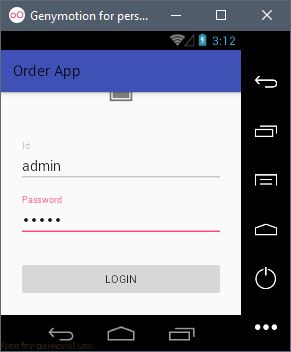
**- Result:** Là kiểu trả về của phương thức **doInBackground** và là kiểu của đối số của phương thức **onPostExcute**.

Dễ hiểu hơn ta có hình ảnh giải thích chi tiết.

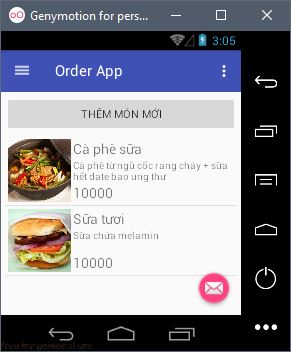


CHƯƠNG 2 – GIAO DIỆN PHẦN MỀM

2.1 Giao diện đăng nhập



2.2 Giao diện danh sách món ăn



**TÀI LIỆU THAM KHẢO**