

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỀ TÀI CUỐI KỲ
PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ

ĐỀ TÀI :

XÂY DỰNG WEBSERVICE QUẢN LÝ NHÀ HÀNG

Giáo viên hướng dẫn : ThS. HUỖNH TRUNG TRỤ

SINH VIÊN :	NGUYỄN THỊ AN	N14DCCN152
	PHAN KỲ GIANG	N14DCCN170
	NGUYỄN TRUNG ĐOÀN	N14DCCN078
	PHAN DUY VŨ	N14DCCN130

Tháng 10 năm 2017

LỜI CẢM ƠN:

Lời đầu tiên, nhóm chúng em xin gửi đến thầy giáo Huỳnh Trung Trụ - giảng viên Bộ môn Phát triển phần mềm hướng dịch vụ - lời cảm ơn chân thành nhất. Trong thời gian khóa học vừa qua thầy đã tận tình chỉ dạy, truyền đạt kiến thức một cách dễ hiểu và tạo điều kiện tốt nhất để chúng em được học hỏi và tiếp thu kiến thức một cách tốt nhất có thể. Cảm ơn thầy đã tận tình hướng dẫn cả lớp cách làm đề tài, truyền đạt kinh nghiệm làm như thế nào là tốt,... Những kiến thức, những kinh nghiệm ấy sẽ là một hành trang để em bước vào đời. Với tất cả tấm lòng chân thành, nhóm chúng em xin bày tỏ sự biết ơn chân thành đến thầy Th.S Huỳnh Trung Trụ lời cảm ơn sâu sắc nhất.

TP.Hồ Chí Minh, tháng 10 năm 2017

MỤC LỤC

GIỚI THIỆU ĐỀ TÀI	1
TỔNG QUAN:	1
CHƯƠNG 1. CÔNG NGHỆ XÂY DỰNG WEBSERVICE	2
1. Giới thiệu công nghệ , mục đích của web service	2
2. Mô hình hoạt động của dịch vụ Web	3
CHƯƠNG 2. TÌM HIỂU VỀ RESTFUL WEB SERVICE	4
1. Khái niệm:	4
2. Các nguyên tắc cơ bản để tạo ra RESTful Service	4
CHƯƠNG 3. XÂY DỰNG WEBSERVICE THÔNG QUA LARAVEL FRAMEWORK	5
1. Thiết kế database:	6
2. Tạo project Laravel :	6
3. Thiết kế API:	7
4. Cấu hình laravel:	7
5. Tạo route:	7
6. Tạo Model :	8
7. Tạo Controller :	8
CHƯƠNG 4. NỀN TẢNG WEBSITE PHP	10
1. Các bước kết nối với web service:	10
2. Giao diện menu chính :	11
CHƯƠNG 3. NỀN TẢNG ANDROID.....	13
1. Cấu trúc chương trình	13
2. Giao diện của chương trình.....	15
CHƯƠNG 4. NỀN TẢNG WINDOWS FORM.....	21
1: Các bước tạo Project:	21
2: Kết nối với Webservice.....	22

GIỚI THIỆU ĐỀ TÀI

Tổng quan:

Theo định nghĩa của W3C (World Wide Web Consortium), dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet. Dịch vụ Web cho phép client và server tương tác được với nhau ngay cả trong những môi trường khác nhau.

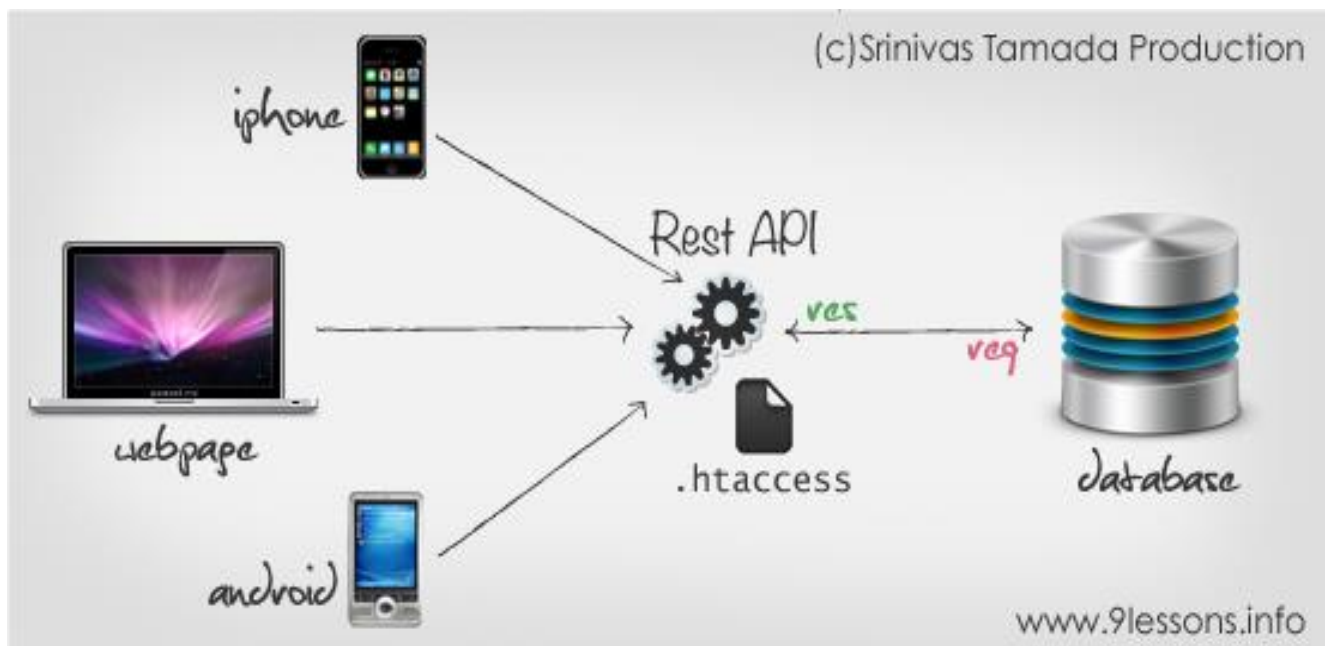
Ví dụ, , Java là một ngôn ngữ lập trình thuần hướng đối tượng và có thể giúp tạo ra các ứng dụng độc lập nền. Chính vì vậy sự kết hợp giữa web service và Java là một công nghệ thời thượng và mạnh mẽ để tạo ra những hệ thống hợp nhất với chi phí và độ phức tạp thấp trong việc tích hợp và phát triển hệ thống. Với mục đích tiếp cận và ứng dụng một cách nhanh nhất web service trong Java, bài báo cáo sẽ giới thiệu những khái niệm cơ bản và những công nghệ nền tảng của web service, được viết bởi sự hỗ trợ của Laravel Framework để từ đó có thể ứng dụng vào xây dựng một hệ thống có sử dụng web service.

Web service cung cấp dịch vụ QUẢN LÝ NHÀ HÀNG qua các nền tảng client PHP, Winform C#, Android.

CHƯƠNG 1. CÔNG NGHỆ XÂY DỰNG WEBSERVICE

1. Giới thiệu công nghệ , mục đích của web service .

Theo định nghĩa của W3C (World Wide Web Consortium), dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML. Dịch vụ Web là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu. Một dịch vụ Web được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ dịch vụ Web khác. Nó bao gồm các mô đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.



**Ưu điểm:*

- + Dịch vụ Web cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.
- + Sử dụng các giao thức và chuẩn mở. Giao thức và định dạng dữ liệu dựa trên văn bản (text), giúp các lập trình viên dễ dàng hiểu được.
- + Nâng cao khả năng tái sử dụng.
- + Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại bằng cách cho phép các tiến trình/chức năng nghiệp vụ đóng gói trong giao diện dịch vụ Web.
- + Tạo mối quan hệ tương tác lẫn nhau và mềm dẻo giữa các thành phần trong hệ thống, dễ dàng cho việc phát triển các ứng dụng phân tán.

PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ

+ Thúc đẩy hệ thống tích hợp, giảm sự phức tạp của hệ thống, hạ giá thành hoạt động, phát triển hệ thống nhanh và tương tác hiệu quả với hệ thống của các doanh nghiệp khác.

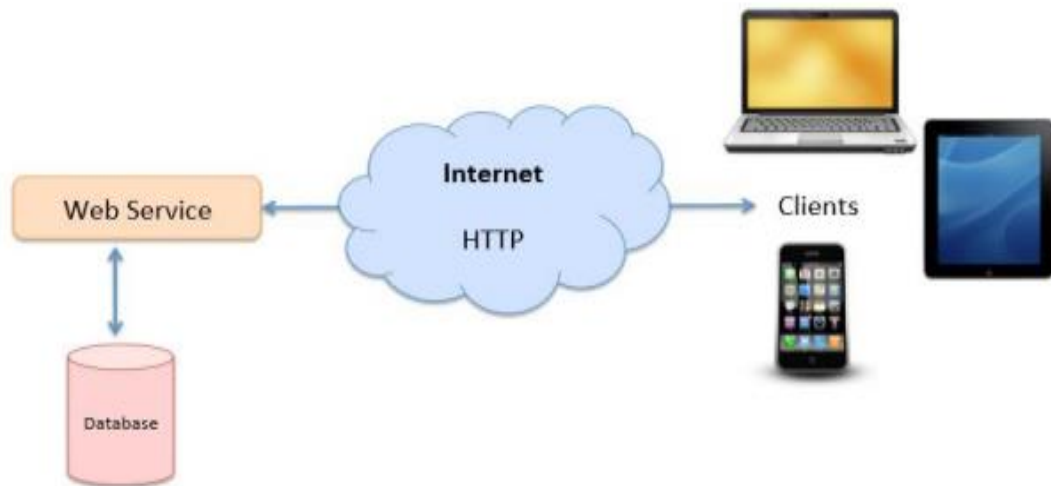
****Nhược điểm :**

+ Những thiệt hại lớn sẽ xảy ra vào khoảng thời gian chết của Dịch vụ Web, giao diện không thay đổi, có thể lỗi nếu một máy khách không được nâng cấp, thiếu các giao thức cho việc vận hành.

+ Có quá nhiều chuẩn cho dịch vụ Web khiến người dùng khó nắm bắt.

+ Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

2. Mô hình hoạt động của dịch vụ Web



CHƯƠNG 2. TÌM HIỂU VỀ RESTFUL WEB SERVICE

1. Khái niệm:

- RESTful web service là các web service được viết dựa trên kiến trúc REST. REST đã được chọn sử dụng rộng rãi thay thế cho các web service dựa trên SOAP và WSDL. RESTful web service nhẹ, có khả năng dễ mở rộng và bảo trì.

- Vậy REST là gì? REST là viết tắt của REpresentational State Transfer (chuyển trạng thái đại diện) là một kiểu kiến trúc lập trình, nó định nghĩa các quy tắc để thiết kế các web service chú trọng vào tài nguyên hệ thống. Trong kiến trúc REST mọi thứ đều được coi là tài nguyên, chúng có thể là: tệp văn bản, ảnh, trang html, video, hoặc dữ liệu động... REST server cung cấp quyền truy cập vào các tài nguyên, REST client truy cập và thay đổi các tài nguyên đó. Ở đây các tài nguyên được định danh dựa vào URI, REST sử dụng một vài đại diện để biểu diễn các tài nguyên như văn bản, JSON, XML.

2. Các nguyên tắc cơ bản để tạo ra RESTful Service

** 4 nguyên tắc thiết kế cơ bản sau:

- Sử dụng các phương thức HTTP một cách rõ ràng
- Phi trạng thái
- Hiện thị cấu trúc thư mục như URIs
- Chuyển đổi JavaScript Object Notation (JSON) và XML hoặc cả hai.

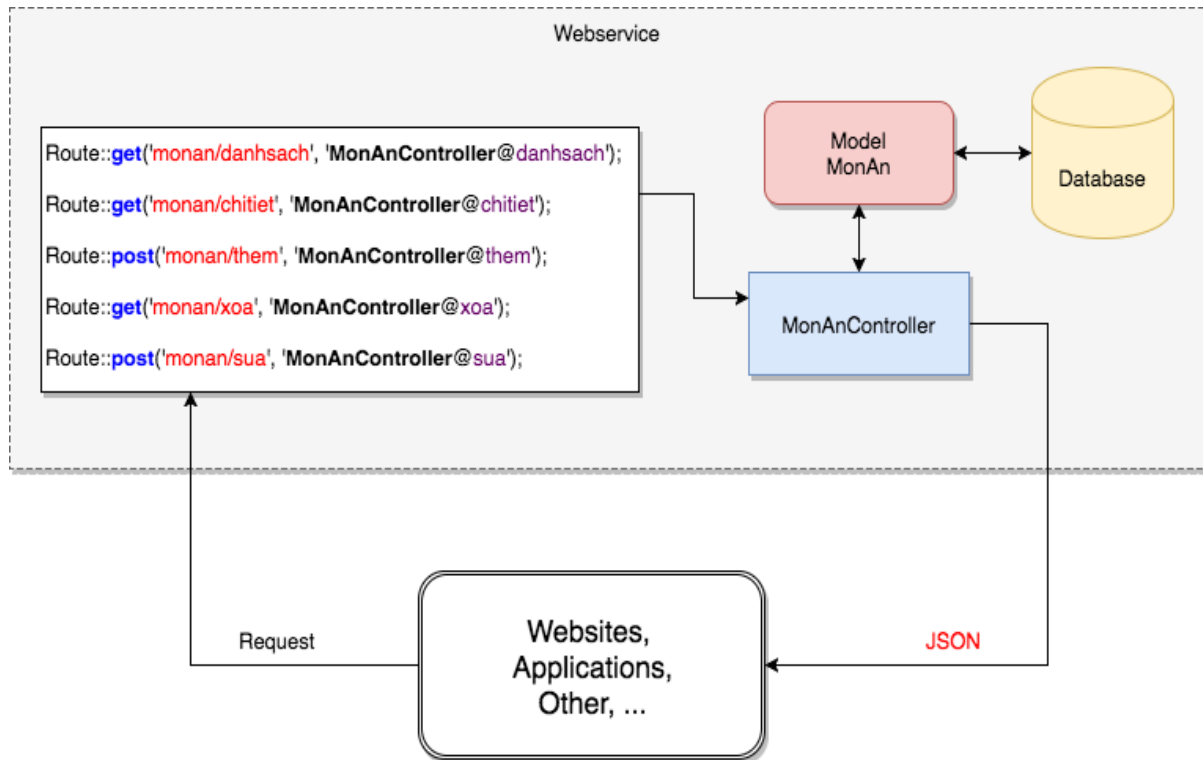
POST (HttpPost) – Tạo một tài nguyên trên máy chủ

GET (HttpGet) – Truy xuất một tài nguyên

PUT (HttpPut) – Thay đổi trạng thái một tài nguyên hoặc để cập nhật nó

DELETE (HttpDelete) – Huỷ bỏ hoặc xoá một tài nguyên

CHƯƠNG 3. XÂY DỰNG WEBSERVICE THÔNG QUA LARAVEL FRAMEWORK



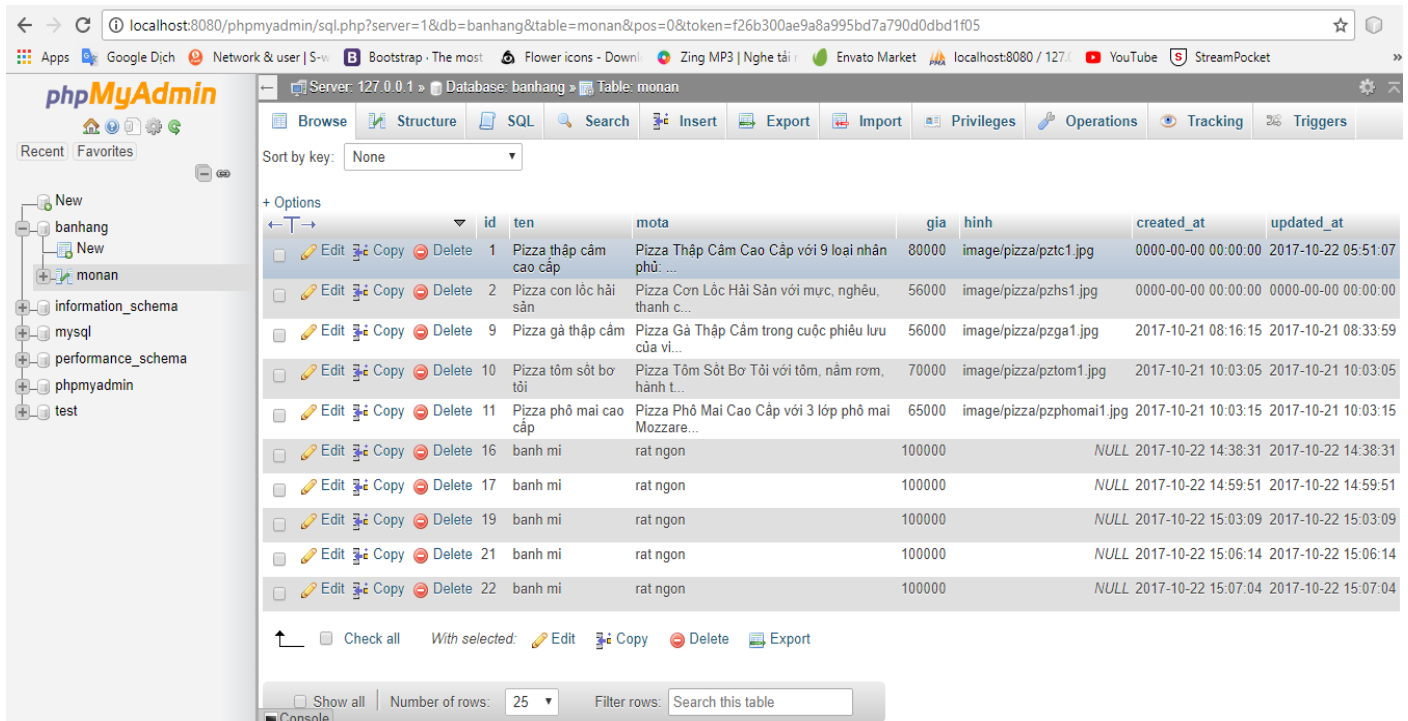
****Công cụ sử dụng :**



PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ

1. Thiết kế database:

Thiết kế trong phpmyadmin: database banhang như sau :



	id	ten	mota	gia	hinh	created_at	updated_at
<input type="checkbox"/>	1	Pizza thập cẩm cao cấp	Pizza Thập Cẩm Cao Cấp với 9 loại nhân phủ: ...	80000	image/pizza/pztc1.jpg	0000-00-00 00:00:00	2017-10-22 05:51:07
<input type="checkbox"/>	2	Pizza con lóc hải sản	Pizza Con Lóc Hải Sản với mực, nghêu, thanh c...	56000	image/pizza/pzhs1.jpg	0000-00-00 00:00:00	0000-00-00 00:00:00
<input type="checkbox"/>	9	Pizza gà thập cẩm	Pizza Gà Thập Cẩm trong cuộc phiêu lưu của vi...	56000	image/pizza/pzga1.jpg	2017-10-21 08:16:15	2017-10-21 08:33:59
<input type="checkbox"/>	10	Pizza tôm sốt bơ tỏi	Pizza Tôm Sốt Bơ Tỏi với tôm, nấm rom, hành t...	70000	image/pizza/pztom1.jpg	2017-10-21 10:03:05	2017-10-21 10:03:05
<input type="checkbox"/>	11	Pizza phô mai cao cấp	Pizza Phô Mai Cao Cấp với 3 lớp phô mai Mozzarella...	65000	image/pizza/pzphomai1.jpg	2017-10-21 10:03:15	2017-10-21 10:03:15
<input type="checkbox"/>	16	banh mi	rat ngon	100000		NULL	2017-10-22 14:38:31
<input type="checkbox"/>	17	banh mi	rat ngon	100000		NULL	2017-10-22 14:59:51
<input type="checkbox"/>	19	banh mi	rat ngon	100000		NULL	2017-10-22 15:03:09
<input type="checkbox"/>	21	banh mi	rat ngon	100000		NULL	2017-10-22 15:06:14
<input type="checkbox"/>	22	banh mi	rat ngon	100000		NULL	2017-10-22 15:07:04

2. Tạo project Laravel :

Vào thư mục htdocs của xampp → (SHIFT+ chuột phải) → Open PowerShell window here → gõ lệnh :

```
Windows PowerShell
PS C:\xampp\htdocs> composer create-project laravel/laravel NhaHang
```

PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ

3. Thiết kế API:

STT	API	Method	Param	Result	Ex
1	monan/danh sach	GET	id ten	[{ "id": 2, "ten": "banh mi", "mota": "rat ngon", "gia": 100000, "created_at": "2017-10-22 15:03:09", "updated_at": "2017-10-22 15:03:09" }, {...}]	monan/danh sach
2	monan/chitiet	GET	id	{ "id": 2, "ten": "banh mi", "mota": "rat ngon", "gia": 100000, "created_at": "2017-10-22 15:03:09", "updated_at": "2017-10-22 15:03:09" }	monan/chitiet?id=2
3	monan/them	POST	id ten gia mota		monan/them form data: id =2 ;ten = pizza ga;gia = 20000;mota = rat ngon
3	monan/sua	POST	id ten gia		monan/sua?id=2 form data: id =2 ;ten = pizza ga;gia = 20000;mota = rat ngon
4	monan/xoa	GET	id		monan/xoa?id=2

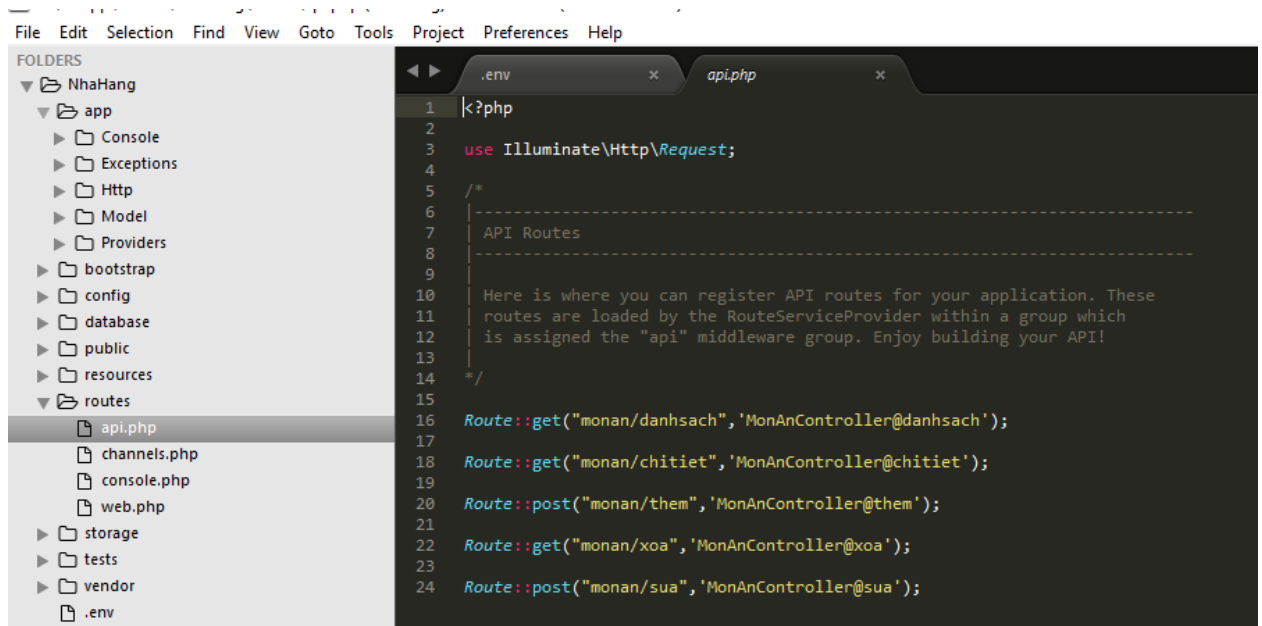
4. Cấu hình laravel:

Vào file ".env" : cấu hình database như sau :

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=banhang
DB_USERNAME=root
DB_PASSWORD=
```

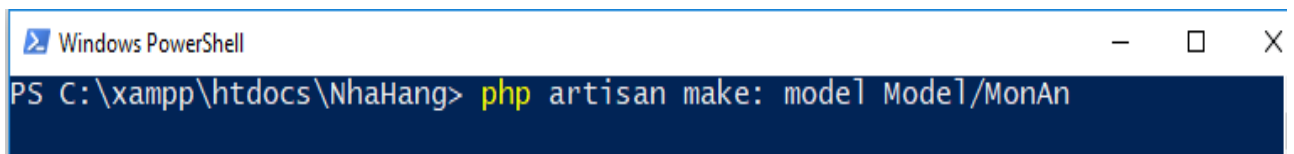
5. Tạo route:

Vào routes → api.php: thiết lập các api theo như thiết kế.



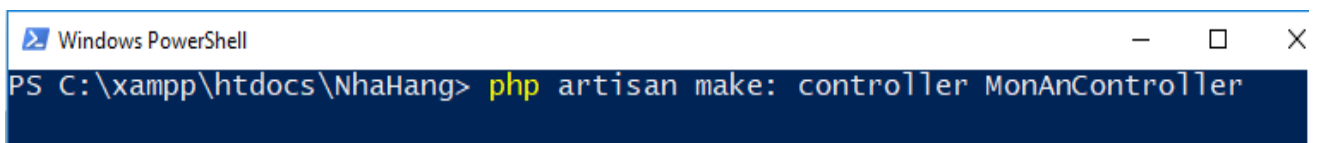
6. Tạo Model :

Vào thư mục chứa webservice → (SHIFT+ chuột phải) → Open PowerShell window here → gõ lệnh như bên dưới để tạo Model MonAn:



7. Tạo Controller :

Vào thư mục chứa webservice → (SHIFT+ chuột phải) → Open PowerShell window here → gõ lệnh như bên dưới để tạo Controller MonAn:



Và sau đó tạo các hàm tương ứng với từng route.

PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ

C:\xampp\htdocs\NhaHang\app\Http\Controllers\MonAnController.php (NhaHang) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

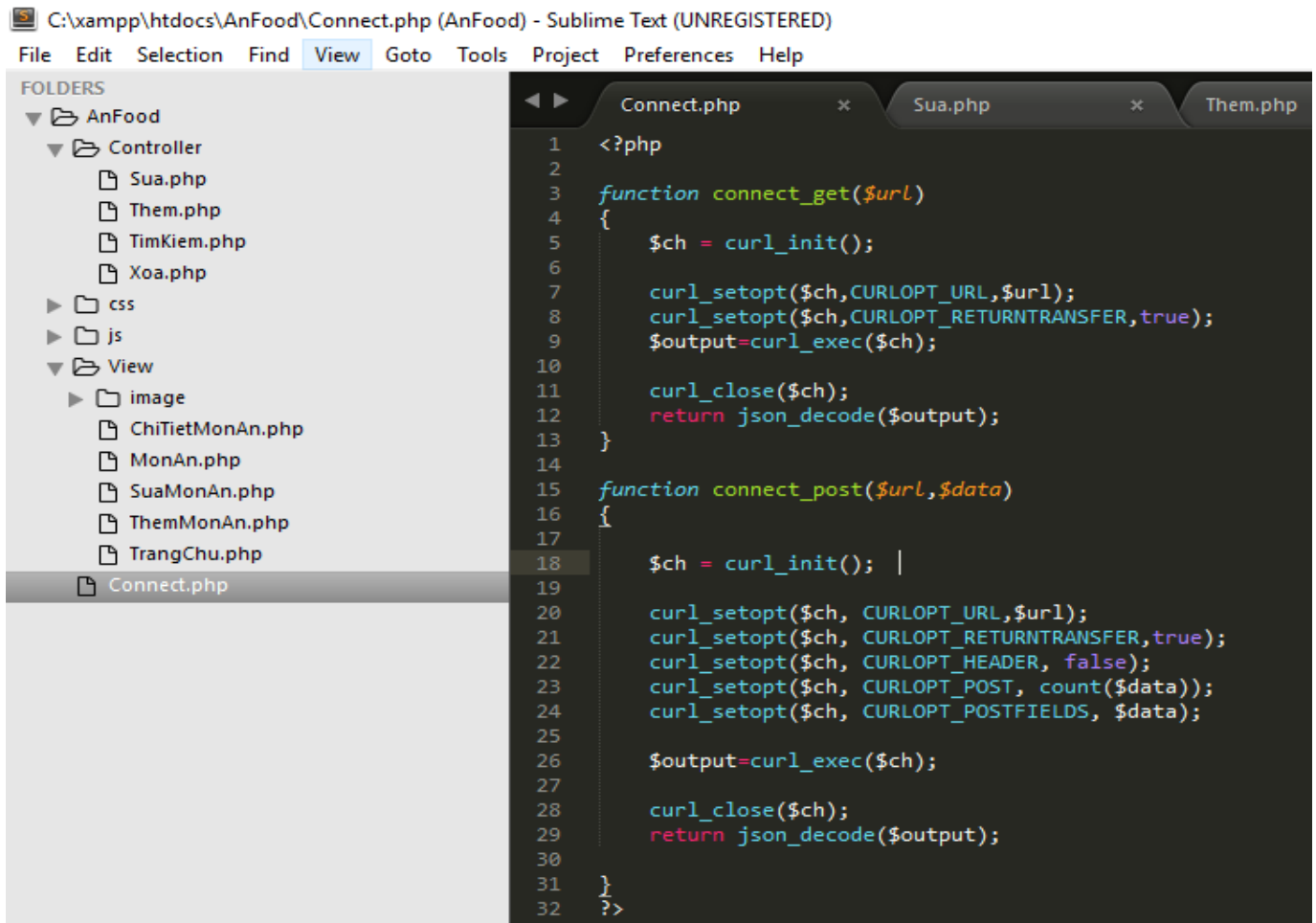
- ▼ NhaHang
 - ▼ app
 - ▶ Console
 - ▶ Exceptions
 - ▼ Http
 - ▼ Controllers
 - ▶ Auth
 - Controller.php
 - MonAnController.php
 - ▶ Middleware
 - Kernel.php
 - ▼ Model
 - MonAn.php
 - ▶ Providers
 - ▶ bootstrap
 - ▶ config
 - ▶ database
 - ▶ public
 - ▶ resources
 - ▶ routes
 - ▶ storage
 - ▶ tests
 - ▶ vendor
 - .env
 - .env.example
 - .gitattributes
 - .gitignore
 - artisan
 - composer.json

```
MonAnController.php x
1 |<?php
2
3 namespace App\Http\Controllers;
4 use App\Model\MonAn;
5
6
7 use Illuminate\Http\Request;
8
9 class MonAnController extends Controller
10 {
11     public function danh sach(Request $rq)
12     { ... }
13
14     public function chi tiet(Request $rq)
15     { ... }
16
17     public function them(Request $rq)
18     { ... }
19
20     public function xoa(Request $rq)
21     { ... }
22
23     public function sua(Request $rq)
24     { ... }
25 }
```

CHƯƠNG 4. NỀN TẢNG WEBSITE PHP

1. Các bước kết nối với web service:

Kết nối theo phương thức GET và POST:

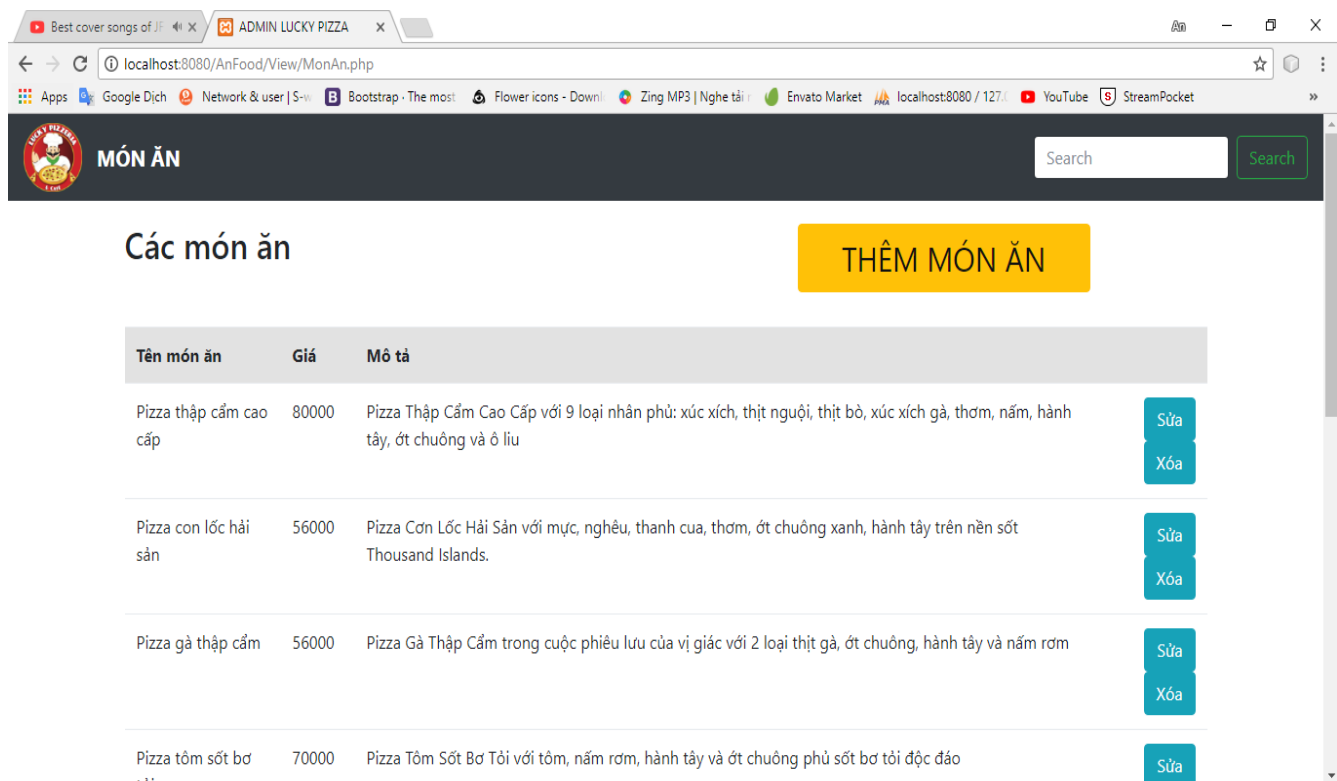


```
C:\xampp\htdocs\AnFood\Connect.php (AnFood) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
▼ AnFood
  ▼ Controller
    Sua.php
    Them.php
    TimKiem.php
    Xoa.php
  ► css
  ► js
  ▼ View
    ► image
    ChiTietMonAn.php
    MonAn.php
    SuaMonAn.php
    ThemMonAn.php
    TrangChu.php
  Connect.php

1 <?php
2
3 function connect_get($url)
4 {
5     $sch = curl_init();
6
7     curl_setopt($sch,CURLOPT_URL,$url);
8     curl_setopt($sch,CURLOPT_RETURNTRANSFER,true);
9     $output=curl_exec($sch);
10
11     curl_close($sch);
12     return json_decode($output);
13 }
14
15 function connect_post($url,$data)
16 {
17
18     $sch = curl_init(); |
19
20     curl_setopt($sch, CURLOPT_URL,$url);
21     curl_setopt($sch, CURLOPT_RETURNTRANSFER,true);
22     curl_setopt($sch, CURLOPT_HEADER, false);
23     curl_setopt($sch, CURLOPT_POST, count($data));
24     curl_setopt($sch, CURLOPT_POSTFIELDS, $data);
25
26     $output=curl_exec($sch);
27
28     curl_close($sch);
29     return json_decode($output);
30
31 }
32 ?>
```

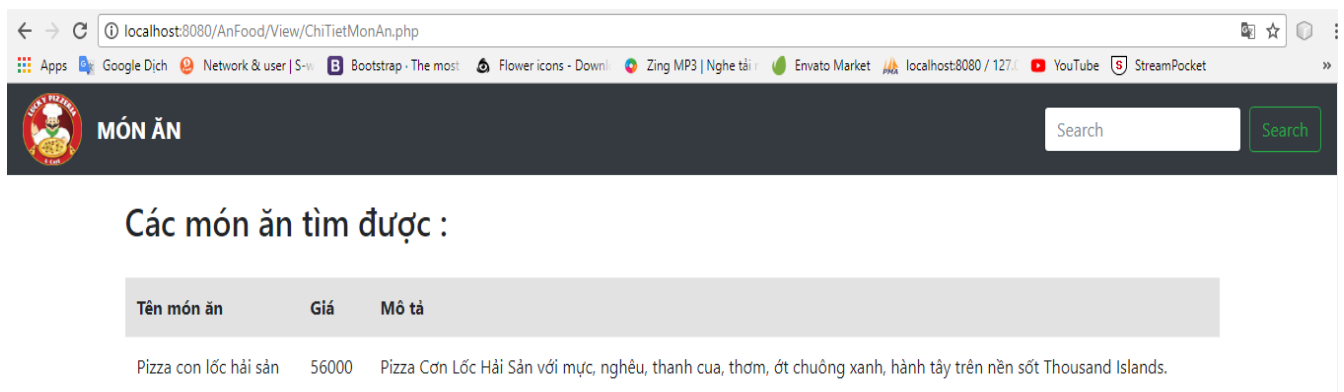
2. Giao diện menu chính :



Gồm các chức năng chính như sau :

+Tìm kiếm món ăn theo tên:

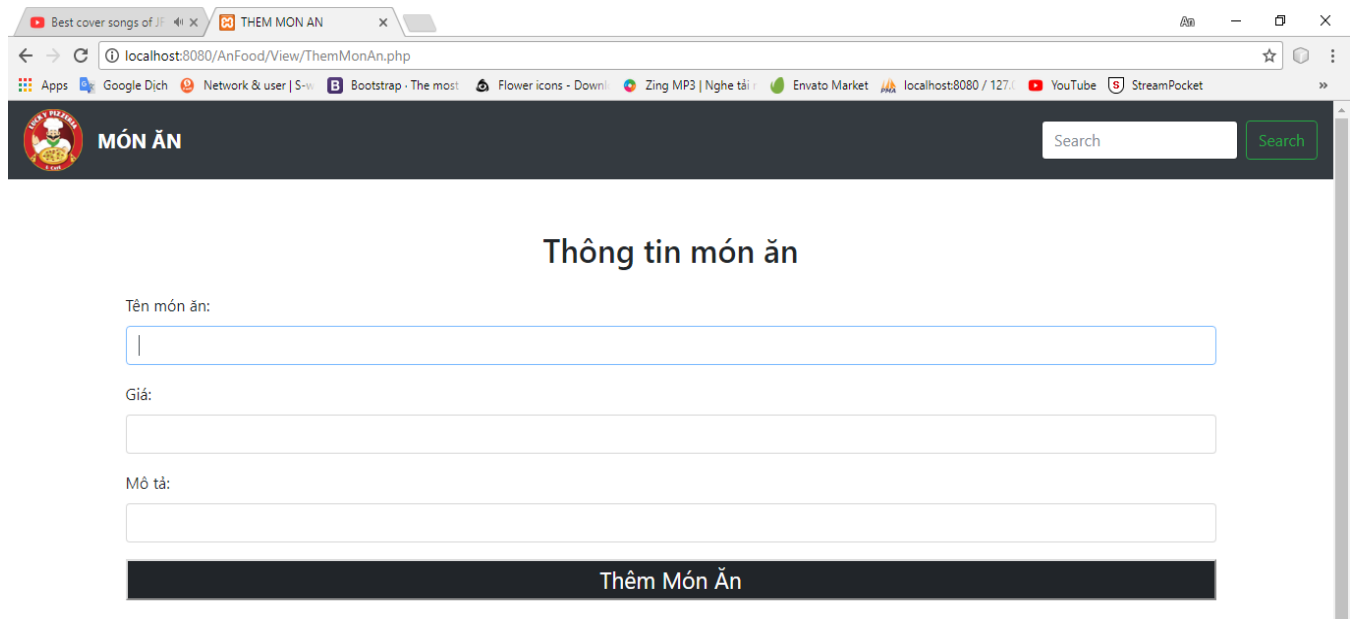
Vd: Tìm món ăn có tên “hải sản”:



+Thêm Món Ăn:

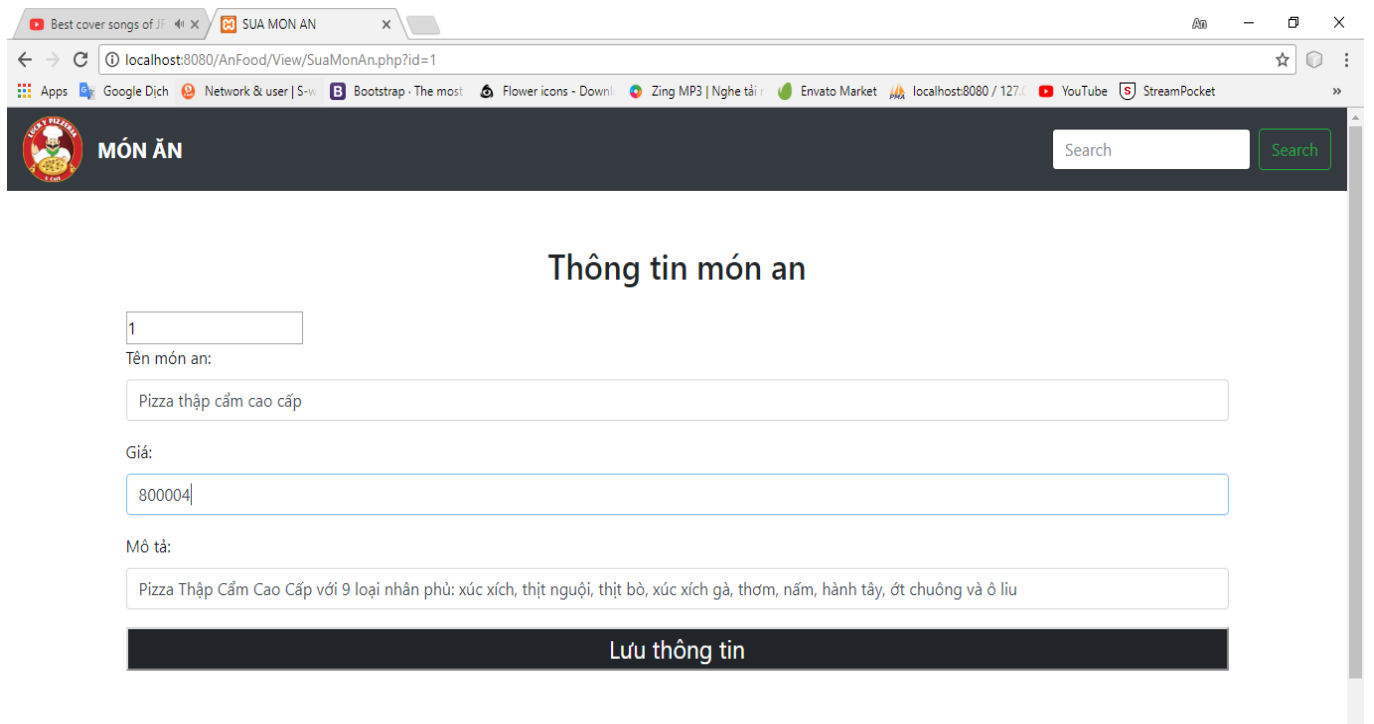
Điền thông tin món ăn cần thêm và sau đó tiến hành lưu thông tin.

PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ



The screenshot shows a web browser window with the title 'THEM MON AN'. The address bar shows 'localhost:8080/AnFood/View/ThemMonAn.php'. The page has a dark header with a logo on the left, the text 'MÓN ĂN' in the center, and a search bar on the right. The main content area is titled 'Thông tin món ăn' and contains three input fields: 'Tên món ăn:', 'Giá:', and 'Mô tả:'. Below these fields is a dark button labeled 'Thêm Món Ăn'.

+Sửa Món Ăn.



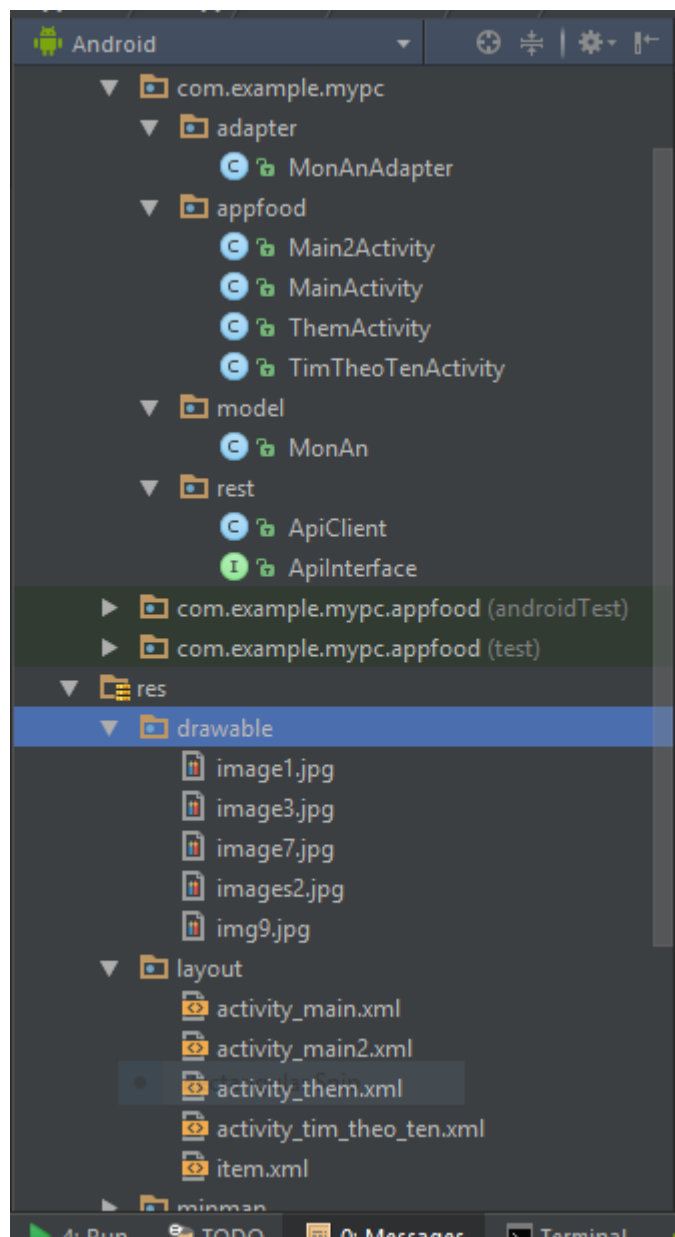
The screenshot shows a web browser window with the title 'SUA MON AN'. The address bar shows 'localhost:8080/AnFood/View/SuaMonAn.php?id=1'. The page has a dark header with a logo on the left, the text 'MÓN ĂN' in the center, and a search bar on the right. The main content area is titled 'Thông tin món an' and contains a text input field with the value '1' above the 'Tên món an:' label. Below this are three input fields: 'Tên món an:' with the value 'Pizza thập cẩm cao cấp', 'Giá:' with the value '800004', and 'Mô tả:' with the value 'Pizza Thập Cẩm Cao Cấp với 9 loại nhân phủ: xúc xích, thịt nguội, thịt bò, xúc xích gà, thơm, nấm, hành tây, ớt chuông và ô liu'. Below these fields is a dark button labeled 'Lưu thông tin'.

+Xóa Món Ăn:

Click vào button “XÓA” ở món ăn nào thì món ăn đó ngay lập tức sẽ được xóa.

CHƯƠNG 3. NỀN TẢNG ANDROID

1. Cấu trúc chương trình




```
public View getView(int position, View convertView, ViewGroup parent) {
    LayoutInflater inflater = context.getLayoutInflater();
    View item = inflater.inflate(this.resource, null);

    MonAn monAn = this.objects.get(position);
    ImageView imgHinh = (ImageView) item.findViewById(R.id.imgHinh);
    TextView txtGia = (TextView) item.findViewById(R.id.txtGia);
    TextView txtMoTa = (TextView) item.findViewById(R.id.txtMoTa);
    TextView txtTen = (TextView) item.findViewById(R.id.txtTen);

    if(monAn.getHinh() != null) {
        Glide.with(context)
            .load(monAn.getHinh().toString())
            .into(imgHinh);
    }
    txtGia.setText(" "+ monAn.getGia());
    txtMoTa.setText(monAn.getMota());
    txtTen.setText(monAn.getId()+" - "+monAn.getTen());

    return item;
}
```

Package adapter để custom layout:

Package appfood chứa các activity.

Package model chứa các parameter để lấy dữ liệu từ webservice, trong interface ApiInterface định nghĩa những phương thức để sử dụng Api như GET và POST

Folder layout chứa các giao diện UI của chương trình, folder drawable chứa các hình ảnh của project.

Sử dụng thư viện Retrofit là một HTTP client type-safe cho Android và Java. Retrofit giúp dễ dàng kết nối đến một dịch vụ REST trên web bằng cách chuyển đổi API thành Java Interface.

Sử dụng thư viện Glide để load ảnh.

Giao diện của chương trình

```
public interface ApiInterface {
    @GET("monan/danhsach")
    Call<ArrayList<MonAn>> getDSMonAn();
    @GET("monan/chitiet")
    Call<MonAn> getChiTietMonAn(@Query("id") String id);
    @GET("monan/danhsach")
    Call<ArrayList<MonAn>> getMonAnTheoTen(@Query("ten") String id);
    @GET("monan/xoa")
    Call<String> xoaMonAn(@Query("id") String id);

    @POST("monan/them")
    @FormUrlEncoded
    Call<String> addMonAn(@Field("ten") String name,
                        @Field("mota") String mota,
                        @Field("gia") int gia);
}
```

a) Màn hình chính (MainActivity)

Hiển thị Listview Món ăn (Hình, mô tả, tên, giá) và các button chức năng.

Để hiển thị Listview, sử dụng hàm loadData();

```
public void loadData() {
    dsMonAn.clear();
    ApiInterface apiService = ApiClient.getClient("http://192.168.40.49:8181/NhaHang/public/api/").create(ApiInterface.class);

    Call<ArrayList<MonAn>> call = apiService.getDSMonAn();
    Log.e("URL", call.request().url() + " ");
    call.enqueue(new Callback<ArrayList<MonAn>>() {
        @Override
        public void onResponse(Call<ArrayList<MonAn>> call, Response<ArrayList<MonAn>> response) {
            int statusCode = response.code();
            dsMonAn = response.body();
            monAnAdapter = new MonAnAdapter(MainActivity.this,
                R.layout.item, dsMonAn);
            lvMonAn.setAdapter(monAnAdapter);
        }

        @Override
        public void onFailure(Call<ArrayList<MonAn>> call, Throwable t) {
            Log.e("Error", t.toString());
            Toast.makeText(MainActivity.this, "Lỗi xảy ra, kiểm tra lại kết nối mạng hoặc server!", Toast.LENGTH_SHORT).show();
        }
    });
}
```

Giao diện hiển thị:



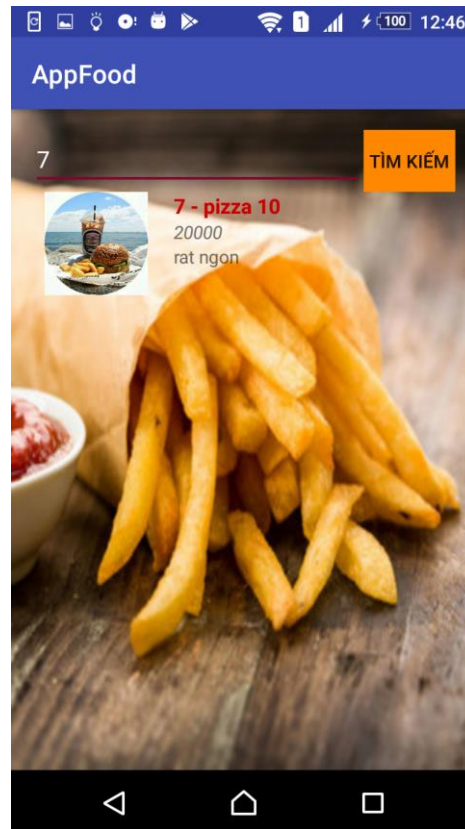
b) Màn hình Tìm kiếm theo id (Main2Activity)

Màn hình gồm Textview chứa id người dùng nhập, Button Tìm Kiếm, và kết quả hiển thị là 1 Món ăn có id bằng id đã nhập. Nếu ID không tồn tại, thông báo “ID NOT FOUND!”.

Xử lý tìm kiếm khi click vào Button Tìm Kiếm.

```
private void addEvents() {
    btnTimKiem.setOnClickListener((view) -> {
        id= txtID.getText().toString();
        ApiInterface apiService = ApiClient.getClient("http://192.168.40.49:8181/NhaHang/public/api/").create(ApiInterface.class);
        Call<MonAn> call = apiService.getChiTietMonAn(id);
        call.enqueue(new Callback<MonAn>() {
            @Override
            public void onResponse(Call<MonAn> call, Response<MonAn> response) {
                int statusCode = response.code();
                monAn = response.body();
                if(monAn.getId()==null){
                    Toast.makeText(Main2Activity.this, "ID Not Found!!! Hãy kiểm tra lại nhé!", Toast.LENGTH_SHORT).show();
                    layoutMonAn.setVisibility(View.INVISIBLE);
                }else
                {
                    txtTen.setText(monAn.getId()+" - "+ monAn.getTen());
                    txtMoTa.setText(monAn.getMota());
                    layoutMonAn.setVisibility(View.VISIBLE);
                    txtGia.setText(monAn.getGia()+"");
                    if(monAn.getHinh()!=null){
                        Glide.with(Main2Activity.this)
                            .load(monAn.getHinh().toString())
                            .into(imgHinh);
                        Log.e("Hinh",monAn.getHinh().toString());
                    }
                }
            }
        })
    })
}
```

c) Giao diện hiển thị:



d) Màn hình Tìm kiếm theo tên (TimTheoTenActivity)

Màn hình gồm 1 Textview để nhập vào tên cần tìm, Button Tìm, Listview kết quả trả về. Nếu không tìm thấy tên món ăn nào chứa chuỗi cần tìm trả về “NAME NOT FOUND!”.

Xử lý tìm kiếm theo tên khi click vào Button Tìm

```
private void addEvents() {
    btnSearch.setOnClickListener((View) -> {
        lvMonAn.setVisibility(View.VISIBLE);
        name= txtName.getText().toString();

        ApiInterface apiService = ApiClient.getClient("http://192.168.40.174:8080/NhaHang/public/api/").create(ApiInterface.class);

        Call<ArrayList<MonAn>> call = apiService.getMonAnTheoTen(name);
        Log.e("URL", call.request().url().toString());
        call.enqueue(new Callback<ArrayList<MonAn>>() {
            @Override
            public void onResponse(Call<ArrayList<MonAn>> call, Response<ArrayList<MonAn>> response) {
                int statusCode = response.code();
                dsMonAn=response.body();
                if (dsMonAn.size()==0){
                    Toast.makeText(TimTheoTenActivity.this, "Name Not Found !!!", Toast.LENGTH_SHORT).show();
                    lvMonAn.setVisibility(View.INVISIBLE);
                } else {
                    monAnAdapter = new MonAnAdapter(TimTheoTenActivity.this,
                        R.layout.item, dsMonAn);
                    lvMonAn.setAdapter(monAnAdapter);
                }
            }
        })
    })
}
```

Ví dụ khi nhập chuỗi “bánh”, sẽ tìm được kết quả như sau:

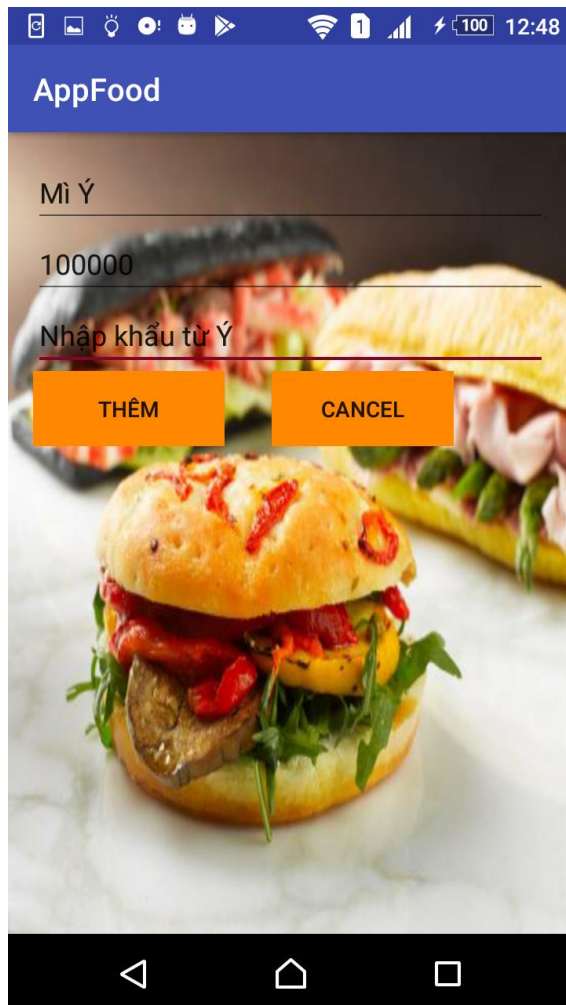


e) Màn hình Thêm món ăn (ThemActivity)

Màn hình gồm 3 ô Textview để nhập tên, giá, mô tả của món ăn. Nhấn vào Button để gửi yêu cầu thêm, Cancel để xóa chuỗi đang nhập trong Textview.

Xử lý khi click Button Thêm:

```
btnThem.setOnClickListener((view) -> {  
  
    final String name= txtName.getText()+"";  
    int gia= Integer.parseInt(txtGia.getText()+"");  
    String mota= txtMota.getText()+"";  
  
    ApiInterface apiService = ApiClient.getClient("http://192.168.40.49:8181/NhaHang/public/api/").create(ApiInterface.class);  
  
    Call<String> call = apiService.addMonAn(name,mota,gia);  
    Log.e("URL", call.request().url()+" ");  
    call.enqueue(new Callback<String>() {  
        @Override  
        public void onResponse(Call<String> call, Response<String> response) {  
            String tb= response.body();  
            Toast.makeText(ThemActivity.this, "Thêm thành công !!!!"+ name , Toast.LENGTH_LONG).show();  
            txtGia.setText("");  
            txtMota.setText("");  
            txtName.setText("");  
            Log.e("ThanhCong", tb);  
        }  
    })  
})
```



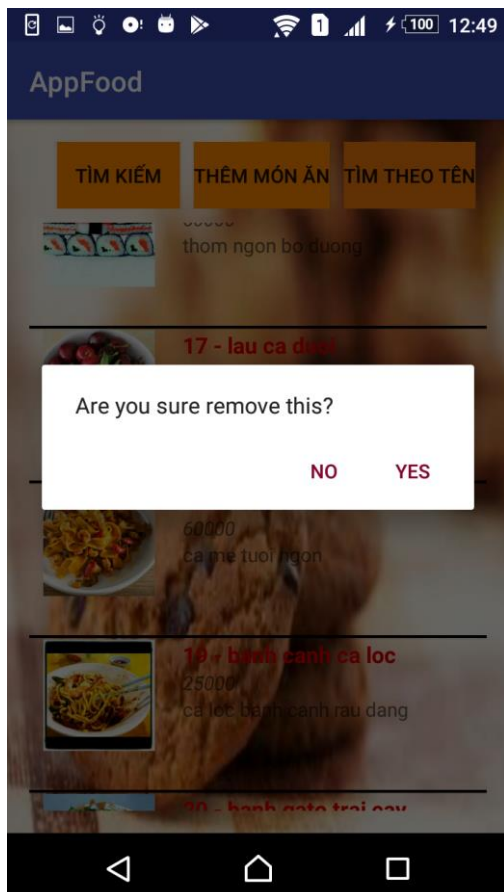
Xóa món ăn

Đây là 1 chức năng ở trong màn hình chính, cho phép người dùng xóa một 1 món ăn khi nhấn và giữ lâu.

Hàm xử lý khi click lâu vào Listview món ăn

PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ

```
lvMonAn.setOnItemClickListener((adapterView, view, pos, l) → {  
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(MainActivity.this);  
    alertDialogBuilder.setMessage("Are you sure remove this?");  
    alertDialogBuilder.setPositiveButton("Yes",  
        new DialogInterface.OnClickListener() {  
            @Override  
            public void onClick(DialogInterface arg0, int k) {  
                ApiInterface apiService = ApiClient.getClient("http://192.168.40.49:8181/NhaHang/public/api/").create()  
                Call<String> call = apiService.xoaMonAn(dsMonAn.get(pos).getId().toString());  
                Log.e("URL", call.request().url() + " ");  
                call.enqueue(new Callback<String>() {  
                    @Override  
                    public void onResponse(Call<String> call, Response<String> response) {  
                        String res=response.body();  
                        Toast.makeText(MainActivity.this, "Đã xóa thành công !!!!!", Toast.LENGTH_SHORT).show();  
                        dsMonAn.remove(pos);  
                        monAnAdapter.notifyDataSetChanged();  
                    }  
                    @Override  
                    public void onFailure(Call<String> call, Throwable t) {  
                        Log.e("Error", t.toString());  
                        Toast.makeText(MainActivity.this, "Xóa không thành công ! Kiểm tra lại", Toast.LENGTH_SHORT).sh  
                    }  
                });  
            }  
        });  
    alertDialogBuilder.setNegativeButton("No", (dialog, which) → {  
        Toast.makeText(MainActivity.this, "Click No", Toast.LENGTH_SHORT).show();  
    });  
});
```

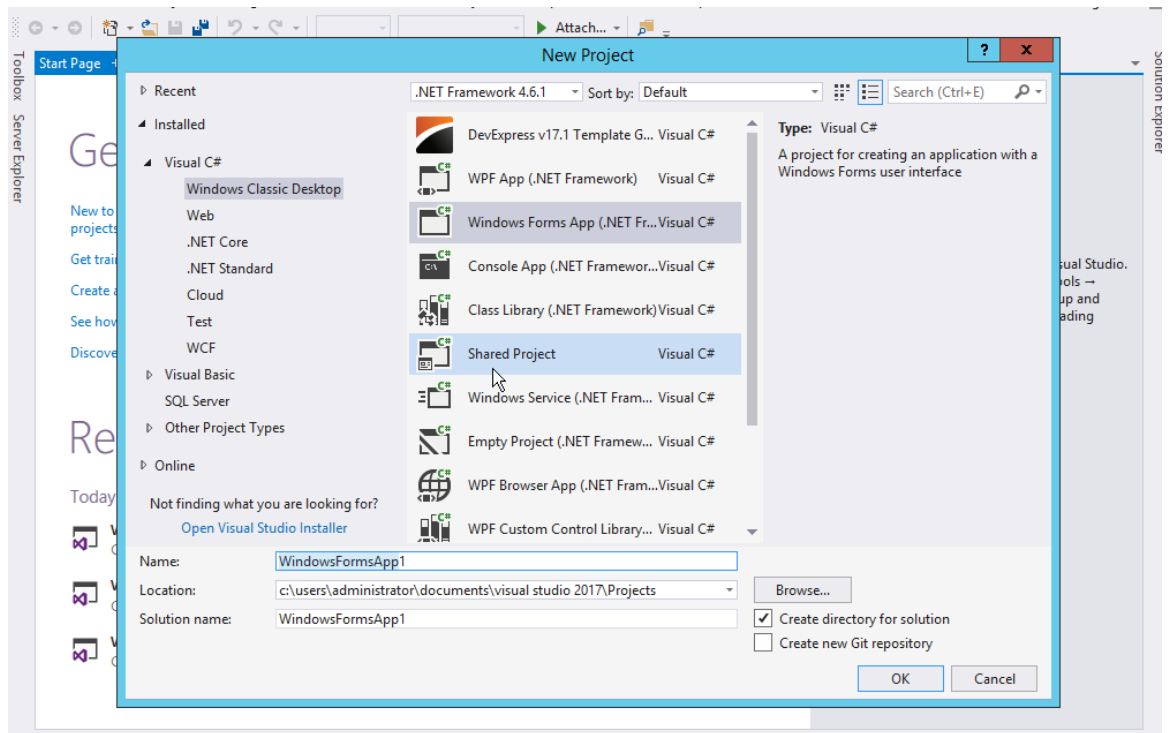


CHƯƠNG 4. NỀN TẢNG WINDOWS FORM

1: Các bước tạo Project:

Open Visual Studio lên :

Vào File->New->Project: chọn WindowsFormsApp(.Net...)



Chúng ta thiết kế một bảng mẫu tương tự trên.

id	ten	mota	gia
----	-----	------	-----

2: Kết nối với Webservice

+ Sử dụng phương thức Request và Response

Code lấy dữ liệu từ Service hiển thị lên GridView:

```
public void HienBang()
{
    HttpWebRequest WR;
    HttpWebResponse WRP = null;
    WR = WebRequest.Create("http://localhost:8080/NhaHang/public/api/monan/danhsach") as HttpWebRequest;
    WR.Method = "GET";
    WRP = (HttpWebResponse)WR.GetResponse();
    Stream dataStream = WRP.GetResponseStream();
    StreamReader reader = new StreamReader(dataStream);
    String strResponse = reader.ReadToEnd();
    var Dt = JsonConvert.DeserializeObject<List<Danhsach>>(strResponse);
    gridControl.DataSource = Dt;
    WRP.Close();
    reader.Close();
}
```

Code thực hiện lệnh xóa:

```
void Xoa(String text)
{
    HttpWebRequest WR;
    HttpWebResponse WRP = null;
    String link = "http://localhost:8080/NhaHang/public/api/monan/xoa?id="+text;
    WR = (HttpWebRequest) WebRequest.Create(link);
    WR.Method = "GET";
    WRP = (HttpWebResponse)WR.GetResponse();
    Stream dataStream = WRP.GetResponseStream();
    StreamReader reader = new StreamReader(dataStream);
    String strResponse = reader.ReadToEnd();
    var Dt = JsonConvert.DeserializeObject<List<Danhsach>>(strResponse);
    gridControl.DataSource = Dt;
    WRP.Close();
    reader.Close();
}

private void buttonXoa_Click(object sender, EventArgs e)
{
    String text = textBox1.Text;
    DialogResult result = MessageBox.Show("Bạn có muốn xóa không?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
        Xoa(text);
        Form1_Load(sender, e);
    }
    else
    {
        MessageBox.Show("Xóa không thành!");
    }
}
```

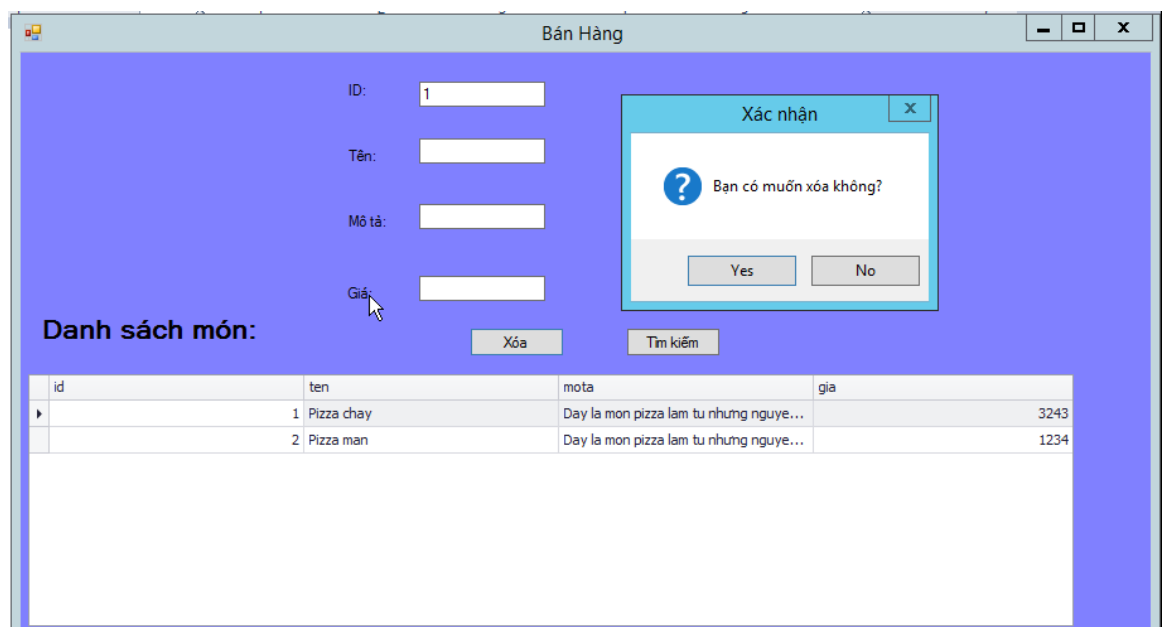
Code thực hiện lệnh tìm kiếm:

```
private void btTim_Click(object sender, EventArgs e)
{
    String text = "";
    try
    {
        text = textBox2.Text;
    }
    catch { }
    if (text != "")
    {
        Tim(text);
        MessageBox.Show("Tìm kiếm thành công!");
    }
    else
        MessageBox.Show("Quên nhập tên tìm kiếm!");
}

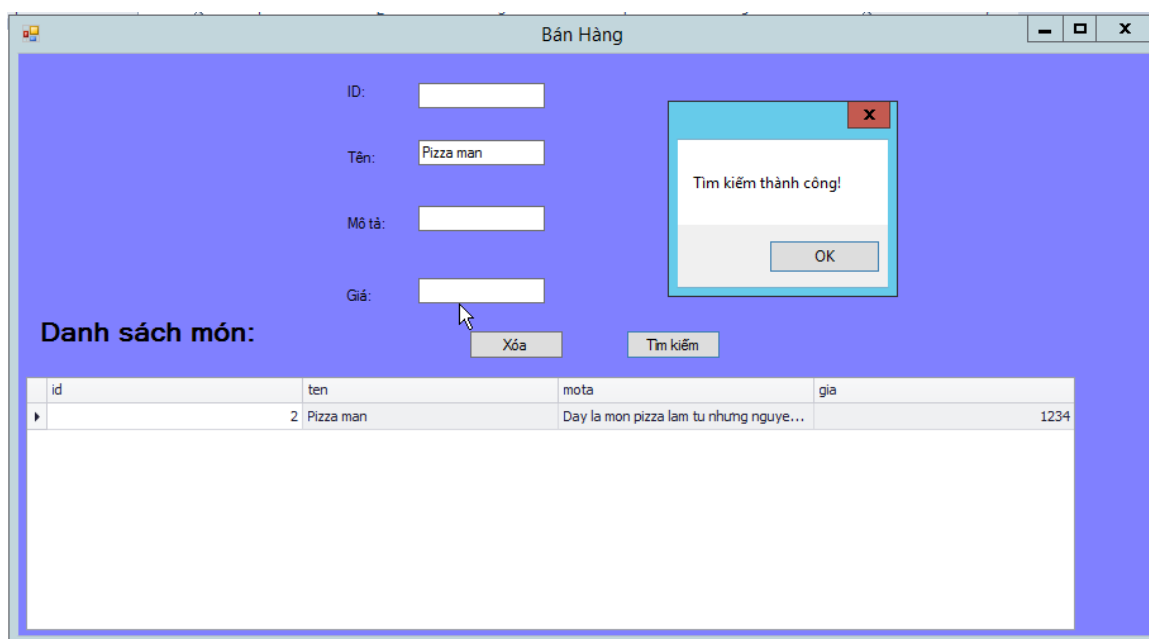
public void Tim(String text)
{
    HttpWebRequest WR;
    HttpResponseMessage WRP = null;
    String link = "http://localhost:8080/NhaHang/public/api/monan/danh sach?ten=" + text;
    WR = (HttpWebRequest)WebRequest.Create(link);
    WR.Method = "GET";
    WRP = (HttpResponseMessage)WR.GetResponse();
    Stream dataStream = WRP.GetResponseStream();
    StreamReader reader = new StreamReader(dataStream);
    String strResponse = reader.ReadToEnd();
    var Dt = JsonConvert.DeserializeObject<List<Danhsach>>(strResponse);
    gridControl.DataSource = Dt;
    WRP.Close();
}
```

Ở đây chỉ thực hiện 2 chức năng của Webservice;

Xóa:



Tim kiếm:



The screenshot shows a web application window titled "Bán Hàng". It features a search form with four input fields: "ID:", "Tên:" (containing "Pizza man"), "Mô tả:", and "Giá:". Below the form are two buttons: "Xóa" and "Tìm kiếm". A search dialog box is open, displaying "Tìm kiếm thành công!" (Search successful!) and an "OK" button. Below the form is a table titled "Danh sách món:" (Menu list:).

id	ten	mota	gia
2	Pizza man	Day la mon pizza lam tu nhung nguye...	1234