PROJECT: Find location to open an Italian restaurant in Ho Chi Minh city, Viet Nam.

I. Introduction: Business Problem

I'm Vietnamese and my friend is Italian , We will build **Italian restaurants** in Viet Nam . In this project we will try to find an optimal location for a restaurant. Specifically, this report will be targeted to stakeholders interested in opening an **Italian restaurant** in **Ho Chi Minh city, Viet Nam.**

Problem

Since there are lots of restaurants in Ho Chi Minh ,Viet Nam.we will try to detect **locations that are not already crowded with restaurants**. We are also particularly interested in **areas with no Italian restaurants in City** . We would also prefer locations **as close to city center as possible**, assuming that first two conditions are met.

We will use our data science powers to generate a few most promissing neighborhoods based on this criteria. Advantages of each area will then be clearly expressed so that best possible final location can be chosen by stakeholders.

Based on definition of our problem, factors that will influence our decission are:

- number of existing restaurants in the neighborhood (any type of restaurant)
- number of and distance to Italian restaurants in the neighborhood, if any
- distance of neighborhood from city center

II. DATA ACQUISITION AND CLEANING

We decided to use regularly spaced grid of locations, centered around city center, to define our neighborhoods.

Following data sources will be needed to extract/generate the required information:

- centers of candidate areas will be generated algorithmically and approximate addresses of centers of those areas will be obtained using library **geopy.geocoders**

In [2]:
```
#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values
import requests
geolocator = Nominatim(user_agent="foursquare_agent")
```

- number of restaurants and their type and location in every neighborhood will be obtained using **Foursquare API**
- coordinate of Ho Chi Minh center will be obtained using library geopy.geocoders of well known Ho Chi Minh location.(Bến thành Market)

In [3]:
```
def get_coordinates(address, verbose=False):
    try:
        location = geolocator.geocode(address)
        lat = location.latitude
        lon = location.longitude
        return [lat, lon]
    except:
        return [None, None]

address = 'Ben Thanh, Ho Chi Minh, Viet Nam'
HoChiMinh_center = get_coordinates(address)
print('Coordinate of {}: {}'.format(address, HoChiMinh_center))
```

```
Coordinate of Ben Thanh, Ho Chi Minh, Viet Nam: [10.7728665, 106.6943]
```

Let's create latitude & longitude coordinates for centroids of our candidate neighborhoods. We will create a grid of cells covering our area of interest which is aprox. 12x12 killometers centered around Ho Chi Minh city center.

Now let's create a grid of area candidates, equaly spaced, centered around city center and within ~6km from Ben Thanh Market. Our neighborhoods will be defined as circular areas with a radius of 300 meters, so our neighborhood centers will be 600 meters apart.

To accurately calculate distances we need to create our grid of locations in Cartesian 2D coordinate system which allows us to calculate distances in meters (not in latitude/longitude degrees). Then we'll project those coordinates back to latitude/longitude degrees to be shown on Folium map. So let's create functions to convert between WGS84 spherical coordinate system (latitude/longitude degrees) and UTM Cartesian coordinate system (X/Y coordinates in meters).

Importal! Zone VietName equal 48

In [6]:
```
def lonlat_to_xy(lon, lat):
    proj_latlon = pyproj.Proj(proj='latlong',datum='WGS84')
    proj_xy = pyproj.Proj(proj="utm", zone=48, datum='WGS84')
    xy = pyproj.transform(proj_latlon, proj_xy, lon, lat)
    return xy[0], xy[1]

def xy_to_lonlat(x, y):
    proj_latlon = pyproj.Proj(proj='latlong',datum='WGS84')
    proj_xy = pyproj.Proj(proj="utm", zone=48, datum='WGS84')
    lonlat = pyproj.transform(proj_xy, proj_latlon, x, y)
    return lonlat[0], lonlat[1]

def calc_xy_distance(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    return math.sqrt(dx*dx + dy*dy)

print('Coordinate transformation check')
print('-------------------------------')
print('Ben Thanh market center longitude={}, latitude={}'.format(HoChiMinh_center[1], HoChiMinh_center[0]))
x, y = lonlat_to_xy(HoChiMinh_center[1], HoChiMinh_center[0])
print('Ben Thanh Market UTM X={}, Y={}'.format(x, y))
lo, la = xy_to_lonlat(x, y)
print('Ben Thanh center longitude={}, latitude={}'.format(lo, la))
```

```
Coordinate transformation check
-------------------------------
Ben Thanh market center longitude=106.6943, latitude=10.7728665
Ben Thanh Market UTM X=685257.385186285, Y=1191377.461362694
Ben Thanh center longitude=106.6943, latitude=10.772866499999997
```

Let's create a hexagonal grid of cells: we offset every other row, and adjust vertical row spacing so that every cell center is equally distant from all it's neighbors.

364 candidate neighborhood centers generated.

In [7]:
```
HoChiMinh_center_x, HoChiMinh_center_y = lonlat_to_xy(HoChiMinh_center[1], HoChiMinh_center[0]) # City center
k = math.sqrt(3) / 2
x_min = HoChiMinh_center_x - 6000
x_step = 600
y_min = HoChiMinh_center_y - 6000 - (int(21/k)*k*600 - 12000)/2
y_step = 600 * k
latitudes = []
longitudes = []
distances_from_center = []
xs = []
ys = []
for i in range(0, int(21/k)):
    y = y_min + i * y_step
    x_offset = 300 if i%2==0 else 0
    for j in range(0, 21):
        x = x_min + j * x_step + x_offset
        distance_from_center = calc_xy_distance(HoChiMinh_center_x, HoChiMinh_center_y, x, y)
        if (distance_from_center <= 6001):
            lon, lat = xy_to_lonlat(x, y)
            latitudes.append(lat)
            longitudes.append(lon)
            distances_from_center.append(distance_from_center)
            xs.append(x)
            ys.append(y)


print(len(longitudes), 'candidate neighborhood centers generated.')
```

```
364 candidate neighborhood centers generated.
```

Detail

In [18]:
```
import pandas as pd
pd.DataFrame({'X':xs,'Y': ys,'Distance from center' : distances_from_center }).head()
```

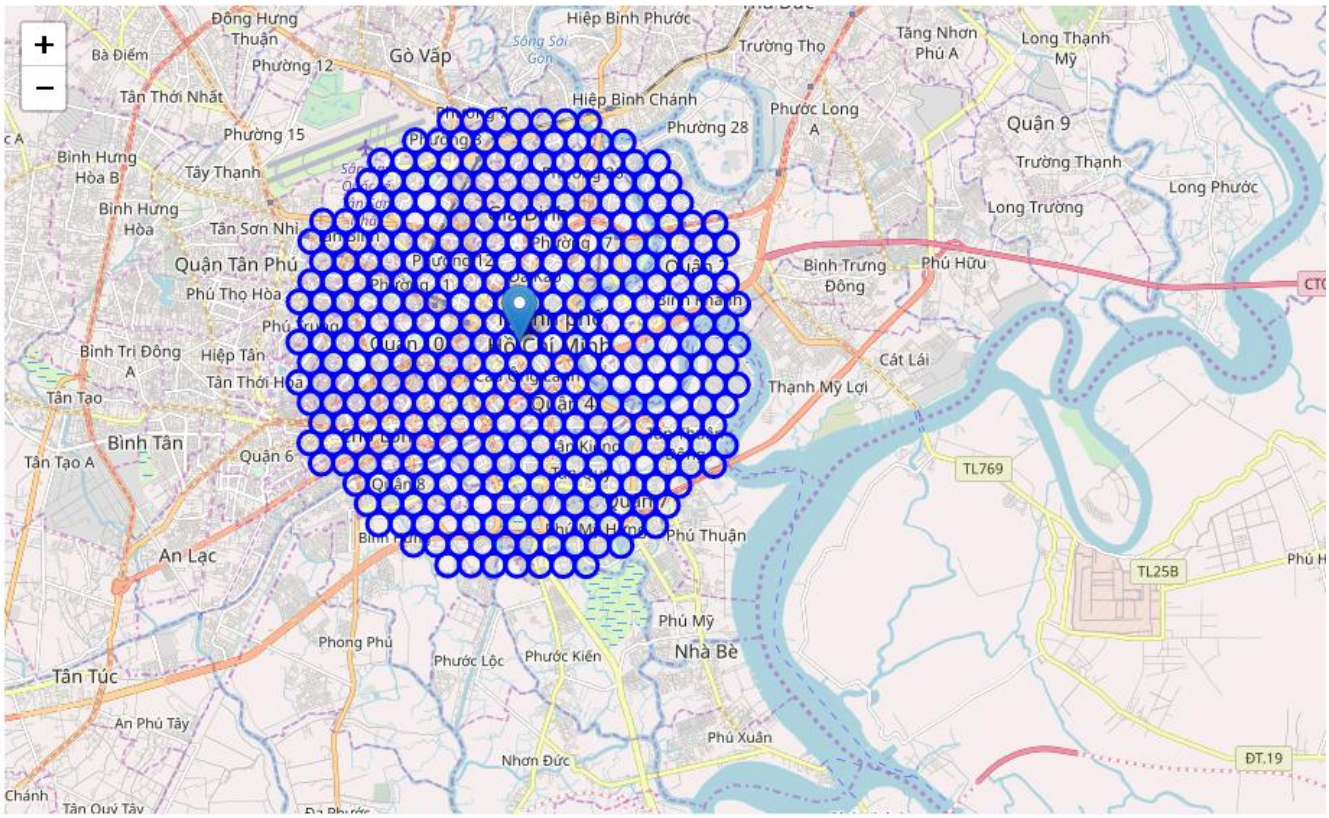
Out[18]:

	X	Y	Distance from center
0	683457.385186	1.185662e+06	5992.495307
1	684057.385186	1.185662e+06	5840.376700
2	684657.385186	1.185662e+06	5747.173218
3	685257.385186	1.185662e+06	5715.767665
4	685857.385186	1.185662e+06	5747.173218

Let's visualize the data we have so far: city center location and candidate neighborhood centers:

In [21]:
```
map_HoChiMinh = folium.Map(location=HoChiMinh_center, zoom_start=13)
folium.Marker(HoChiMinh_center, popup='Ben Thanh Market').add_to(map_HoChiMinh)
for lat, lon in zip(latitudes, longitudes):
    folium.Circle([lat, lon], radius=300, color='blue', fill=False).add_to(map_HoChiMinh)
map_HoChiMinh
```

Out[21]:



we now have the coordinates of centers of neighborhoods/areas to be evaluated, equally spaced (distance from every point to it's neighbors is exactly the same) and within ~6km from Ben

Thanh Market. Let's now use Lib geopy.geocoders to get approximate addresses of those locations.

In [22]:
```
def get_address(latitude, longitude):
    try:
        addre=''
        addre=str(latitude) + ',' + str(longitude)
        location=geolocator.reverse(addre)
        addre=location.address
        return addre
    except:
        return 'Error'

addr = get_address(HoChiMinh_center[0], HoChiMinh_center[1])
print('Reverse geocoding check')
print('-----------------------')
print('Address of [{}, {}] is: {}'.format(HoChiMinh_center[0], HoChiMinh_center[1], addr))
```

```
Reverse geocoding check
-----------------------
Address of [10.7728665, 106.6943] is: Central Park 2 Building, 117-121, Nguyễn Du, Phường Bến Thành, Quận 1, Thành phố Hồ Chí M
700000, Việt Nam
```

And get address 364 candidate neighborhood , address remove string **‘,Việt Nam’**

In [23]:
```
print('Obtaining location addresses: ', end='')
addresses = []
for lat, lon in zip(latitudes, longitudes):
    address = get_address(lat, lon)
    if address is None:
        address = 'NO ADDRESS'
    address = address.replace(r', Việt Nam', '') # We don't need country part of address
    addresses.append(address)
    print('.', end='')
print(' done.')
```

```
Obtaining location addresses:  . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . done.
```

Detail from [10:70]

In [24]:
```
addresses[10:70]
```

Out[24]:
```
['Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 72915',
 'số 12, Khu Dân Cư Ven Sông, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 72915',
 'Đường số 1, Khu 4 - Khu dân cư Ven Sông, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 72915',
 'Trạm Xử lý nước thải Phú Mỹ Hưng, Nguyễn Phan Chánh, Khu Cảnh Đồi, Phú Mỹ Hưng, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh,
72915',
 'Riverpark Residence, 81, Hà Huy Tập, Khu Cảnh Đồi, Phú Mỹ Hưng, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'Hà Huy Tập, Khu Cảnh Đồi, Phú Mỹ Hưng, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'Hồ Bán Nguyệt, Tôn Dật Tiên, The Crescent, Khu Hồ Bán Nguyệt, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 750510',
 'Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 750510',
 'Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 72000',
 'Hẻm C8, Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 756912',
 'Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 756912',
 'Thảo Loan Plaza, Đường số 9A, KDC Trung Sơn, Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 72915',
 'RMIT University Vietnam - Saigon South Campus, 702, Đường Nguyễn Văn Linh, Khu 3 - Khu Đại học, Phường Tân Phong, Quận 7, Thành phố
Hồ Chí Minh, 72915',
 'ĐH Tôn Đức Thắng, 19, Nguyễn Hữu Thọ, Khu 3 - Khu Đại học, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 72915',
 'Richlane Residence, Vivo city, Khu 2, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 72915',
 'Đặng Đại Độ, Khu Văn hóa Giải trí, Hưng Vượng 1, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'Golf Nam Sai Gon, Đường Tôn Dật Tiên, Khu Y Tế Điều Dưỡng, Phú Mỹ Hưng, Phường Tân Phú, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'SECC - Trung tâm Hội nghị và Triển lãm Sài Gòn, Hoàng Văn Thái, The Crescent, Khu Hồ Bán Nguyệt, Quận 7, Thành phố Hồ Chí Minh,
756604',
 'Tân Phú, Phú Mỹ Hưng, Phường Tân Phú, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'Trường THCS Sương Nguyệt Ánh, Dưỡng 195, Khu dân cư Hiệp Ân, Phường 5, Quận 8, Thành phố Hồ Chí Minh, 750510',
 'Hẻm 283/12/2 Bông Sao, Khu dân cư Hiệp Ân, Phường 5, Quận 8, Thành phố Hồ Chí Minh, 72000',
 'Đường 394 Tạ Quang Bửu, Khu dân cư Hiệp Ân, Phường 5, Quận 8, Thành phố Hồ Chí Minh, 72000',
 'Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 756912',
 'Trường trung cấp nghiệp vụ nam Sài Gòn, Cao Lỗ, Phường 4, Quận 8, Thành phố Hồ Chí Minh, 756912',
 'KDC 6A Intresco, Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 756912',
 'Đường số 5, KDC Trung Sơn, Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ Chí Minh, 756912',
 'Đại học Cảnh sát Nhân dân, Nguyễn Hữu Thọ, Khu 2, Khu Dân cư Kim Sơn, Quận 7, Thành phố Hồ Chí Minh, 72915',
 'hoa, 523, Lê Văn Lương, Khu 2, Khu Dân cư Kim Sơn, Quận 7, Thành phố Hồ Chí Minh, 72915',
 'Hưng Thái 2, Đường Nguyễn Đồng Chi, Khu Văn hóa Giải trí, Hưng Thái, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 72915',
 'Số 25 Tân Quy Đông, Khu Dân Cư Tân Quy Đông, Phường Tân Phong, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'Đường số 8, Khu Y Tế Điều Dưỡng, Phú Mỹ Hưng, Phường Tân Phú, Quận 7, Thành phố Hồ Chí Minh, 756604',
 'Phường Tân Phú, Quận 7, Thành phố Hồ Chí Minh, 756604',
```

Let's now place all this into a Pandas dataframe.

```
In [25]: import pandas as pd
         df_locations = pd.DataFrame({'Address': addresses,
                                      'Latitude': latitudes,
                                      'Longitude': longitudes,
                                      'X': xs,
                                      'Y': ys,
                                      'Distance from center': distances_from_center})

         df_locations.head(10)
```

Out[25]:

	Address	Latitude	Longitude	X	Y	Distance from center
0	Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành p...	10.721284	106.677557	683457.385186	1.185662e+06	5992.495307
1	Phạm Hùng, Khu 6, Khu Dân cư T30, Xã Bình Hưng...	10.721254	106.683042	684057.385186	1.185662e+06	5840.376700
2	Đường số 1, Khu 6, Xã Bình Hưng, Huyện Bình Ch...	10.721225	106.688527	684657.385186	1.185662e+06	5747.173218
3	Đường A, Khu dân cư Phước Kiển, Khu Dân cư Ph...	10.721195	106.694012	685257.385186	1.185662e+06	5715.767665
4	Xã Phước Kiển, Huyện Nhà Bè, Thành phố Hồ Chí ...	10.721165	106.699497	685857.385186	1.185662e+06	5747.173218
5	Xã Phước Kiển, Huyện Nhà Bè, Thành phố Hồ Chí ...	10.721135	106.704981	686457.385186	1.185662e+06	5840.376700
6	Trường Đinh Thiện Lý, 80, Nguyễn Đức Cảnh, Khu...	10.721105	106.710466	687057.385186	1.185662e+06	5992.495307
7	Cầu Xóm Cũi, Đường Nguyễn Văn Linh, Xã Bình Hư...	10.726026	106.669356	682557.385186	1.186181e+06	5855.766389
8	Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ C...	10.725996	106.674841	683157.385186	1.186181e+06	5604.462508
9	Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành p...	10.725967	106.680326	683757.385186	1.186181e+06	5408.326913

We can save to file csv and pickle for loading …

```
In [27]: project.save_data("hochiminh_new.csv", df_locations.to_csv())
```

```
Out[27]: {'file_name': 'hochiminh_new.csv',
          'message': 'File saved to project storage.',
          'bucket_name': 'projectcourse9finalassignment-donotdelete-pr-jihlcyvkuqijxp',
          'asset_id': 'f344f2c1-ea62-4317-9e47-cc2b540a2114'}
```

...and let's now save/persist this data into local file.

```
In [28]: df_locations.to_pickle('./locations.pkl')
```

Use Foursquare for collect data

Now that we have our location candidates, let's use Foursquare API to get info on restaurants in each neighborhood.

We're interested in venues in 'food' category, but only those that are proper restaurants - coffe shops, pizza places, bakeries etc. are not direct competitors so we don't care about those. So we will include in out list only venues that have 'restaurant' in category name, and we'll make sure to detect and include all the subcategories of specific 'Italian restaurant' category, as we need info on Italian restaurants in the neighborhood.

Category IDs corresponding to Italian restaurants were taken from Foursquare web site (https://developer.foursquare.com/docs/resources/categories):
food_category = '4d4b7105d754a06374d81259' # 'Root' category for all food-related venues

```
italian_restaurant_categories = ['4bf58dd8d48988d110941735','55a5a1ebe4b013909087cbb6','55a5a1ebe4b013909087cb7c',
                                 '55a5a1ebe4b013909087cba7','55a5a1ebe4b013909087cba1','55a5a1ebe4b013909087cba4',
                                 '55a5a1ebe4b013909087cb95','55a5a1ebe4b013909087cb89','55a5a1ebe4b013909087cb9b',
                                 '55a5a1ebe4b013909087cb98','55a5a1ebe4b013909087cbbf','55a5a1ebe4b013909087cb79',
                                 '55a5a1ebe4b013909087cbb0','55a5a1ebe4b013909087cbb3','55a5a1ebe4b013909087cb74',
                                 '55a5a1ebe4b013909087cbaa','55a5a1ebe4b013909087cb83','55a5a1ebe4b013909087cb8c',
                                 '55a5a1ebe4b013909087cb92','55a5a1ebe4b013909087cb8f','55a5a1ebe4b013909087cb86',
                                 '55a5a1ebe4b013909087cbb9','55a5a1ebe4b013909087cb7f','55a5a1ebe4b013909087cbbc',
                                 '55a5a1ebe4b013909087cb9e','55a5a1ebe4b013909087cbc2','55a5a1ebe4b013909087cbad']
```

Define function is_restaurant, get_categories, format_address, get_venues_near_location

```
def get_restaurants(lats, lons):
    restaurants = {}
    italian_restaurants = {}
    location_restaurants = []

    print('Obtaining venues around candidate locations:', end='')
    for lat, lon in zip(lats, lons):
        # Using radius=350 to meke sure we have overlaps/full coverage so we don't miss any restaurant (we're using dictionaries to
        venues = get_venues_near_location(lat, lon, food_category, foursquare_client_id, foursquare_client_secret, radius=350, limi
        area_restaurants = []
        for venue in venues:
            venue_id = venue[0]
            venue_name = venue[1]
            venue_categories = venue[2]
            venue_latlon = venue[3]
            venue_address = venue[4]
            venue_distance = venue[5]
            is_res, is_italian = is_restaurant(venue_categories, specific_filter=italian_restaurant_categories)
            if is_res:
                x, y = lonlat_to_xy(venue_latlon[1], venue_latlon[0])
                restaurant = (venue_id, venue_name, venue_latlon[0], venue_latlon[1], venue_address, venue_distance, is_italian, x,
                if venue_distance<=300:
                    area_restaurants.append(restaurant)
                restaurants[venue_id] = restaurant
                if is_italian:
                    italian_restaurants[venue_id] = restaurant
        location_restaurants.append(area_restaurants)
        print(' .', end='')
    print(' done.')
    return restaurants, italian_restaurants, location_restaurants
# Try to load from local file system in case we did this before
restaurants = {}
italian_restaurants = {}
location_restaurants = []
loaded = False
try:
    with open('restaurants_350.pkl', 'rb') as f:
        restaurants = pickle.load(f)
    with open('italian_restaurants_350.pkl', 'rb') as f:
        italian_restaurants = pickle.load(f)
```

over our neighborhood locations and get nearby restaurants; we'll also maintain a dictionary of all found restaurants and all found italian restaurants

Funtion get_restaurants will get all (version = '20190824')

```
def get_restaurants(lats, lons):
    restaurants = {}
    italian_restaurants = {}
    location_restaurants = []
    print('Obtaining venues around candidate locations:', end='')
    for lat, lon in zip(lats, lons):
        # Using radius=350 to meke sure we have overlaps/full coverage so we don't miss any restaurant (we're using dictionarie
        venues = get_venues_near_location(lat, lon, food_category, foursquare_client_id, foursquare_client_secret, radius=350,
        area_restaurants = []
        for venue in venues:
            venue_id = venue[0]
            venue_name = venue[1]
            venue_categories = venue[2]
            venue_latlon = venue[3]
            venue_address = venue[4]
            venue_distance = venue[5]
            is_res, is_italian = is_restaurant(venue_categories, specific_filter=italian_restaurant_categories)
            if is_res:
                x, y = lonlat_to_xy(venue_latlon[1], venue_latlon[0])
                restaurant = (venue_id, venue_name, venue_latlon[0], venue_latlon[1], venue_address, venue_distance, is_italian
                if venue_distance<=300:
                    area_restaurants.append(restaurant)
                restaurants[venue_id] = restaurant
                if is_italian:
                    italian_restaurants[venue_id] = restaurant
        location_restaurants.append(area_restaurants)
        print('.', end='')
    print(' done.')
    return restaurants, italian_restaurants, location_restaurants
```

And Try to load from local file system in case we did this before

```
restaurants = {}
italian_restaurants = {}
location_restaurants = []
loaded = False
try:
    with open('restaurants_350.pkl', 'rb') as f:
        restaurants = pickle.load(f)
    with open('italian_restaurants_350.pkl', 'rb') as f:
        italian_restaurants = pickle.load(f)
    with open('location_restaurants_350.pkl', 'rb') as f:
        location_restaurants = pickle.load(f)
    print('Restaurant data loaded.')
    loaded = True
except:
    pass

# If load failed use the Foursquare API to get the data
if not loaded:
    restaurants, italian_restaurants, location_restaurants = get_restaurants(latitudes, longitudes)

    # Let's persists this in local file system
    with open('restaurants_350.pkl', 'wb') as f:
        pickle.dump(restaurants, f)
    with open('italian_restaurants_350.pkl', 'wb') as f:
        pickle.dump(italian_restaurants, f)
    with open('location_restaurants_350.pkl', 'wb') as f:
        pickle.dump(location_restaurants, f)
```

```
Obtaining venues around candidate locations: . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . done.
```

OK! We will get address about restaurants and Italian restaurants ,

```
In [31]: import numpy as np

         print('Total number of restaurants:', len(restaurants))
         print('Total number of Italian restaurants:', len(italian_restaurants))
         print('Percentage of Italian restaurants: {:.2f}%'.format(len(italian_restaurants) / len(restaurants) * 100))
         print('Average number of restaurants in neighborhood:', np.array([len(r) for r in location_restaurants]).mean())

         Total number of restaurants: 1282
         Total number of Italian restaurants: 20
         Percentage of Italian restaurants: 1.56%
         Average number of restaurants in neighborhood: 3.0494505494505493
```

Total number of restaurants: 1282
Total number of Italian restaurants: 20
Percentage of Italian restaurants: 1.56%
Average number of restaurants in neighborhood: 3.0494505494505493

And we List of all restaurants (1282)

```
In [32]: print('List of all restaurants')
         print('----------------------')
         for r in list(restaurants.values())[:10]:
             print(r)
         print('...')
         print('Total:', len(restaurants))

         List of all restaurants
         ----------------------
         ('4d92f441922c6ea8b8955a79', "Edward's Bistro", 10.71983188678315, 106.70066570463338, 'Hoàng Anh Gia Lai 3   D1-09 New
         Chí Minh', 195, False, 685986.0900370291, 1185514.9320741964)
         ('52d668d8498e659c75788cc7', 'Hiền Lành Quán CN2', 10.72265925916767, 106.70127309136058, 'Việt Nam', 255, False, 686050
         1185828.046498792)
         ('5b5b1796e55d8b002c495933', 'Hong Kong Restaurant', 10.723514, 106.699715, 'Kenton, Thành phố Hồ Chí Minh, Thành phố Hồ
         262, False, 685879.8431606898, 1185921.6514310746)
         ('58a835f992ca4c07d513a30e', 'Cánh Quạt', 10.71875, 106.6991, 'Lê Văn Lương, Thành phố Hồ Chí Minh', 272, False, 685815.
         1185394.3147540337)
         ('54bb1228498e0b8a26dd558c', 'Phở Kim Hưng', 10.722993, 106.709824, '70 Ly Long Tuong', 221, False, 686985.9842873572,
         1185870.1451991696)
         ('4d528c3371548cfa40f8279a', 'Kichi Kichi Nguyen Duc Canh', 10.722551855989154, 106.710182654488, 'SC3-1 Grand View Nguy
         hành phố Hồ Chí Minh, Thành phố Hồ Chí Minh', 164, False, 687025.4886569944, 1185821.5661394093)
         ('4d1ecd18756e8cface055e54', 'Akatonbo', 10.721889630844041, 106.71304697650088, '195 (Ton Dat Thien, P. Tan Phong, Q7),
         Chí Minh, Thành phố Hồ Chí Minh', 295, False, 687339.2257659828, 1185750.0562615623)
         ('4ed1f628e3007feb7f653280', 'Pho Hung 2', 10.72224815754579, 106.71056145714186, 'Việt Nam', 127, False, 687067.112683
         1185788.2028504247)
         ('4cf8e5715698a093fdc207fc', 'Phở 24 - Grandview', 10.72150807830618, 106.71193181357592, '3SE-7-1 Nguyễn Đức Cảnh, P. 
         7, Thành phố Hồ Chí Minh', 166, False, 687217.4719304186, 1185707.1724020198)
         ('4e0881cfc65b5bf277961859', 'Quán Bên Sông', 10.71997003326531, 106.71168437972258, 'nhà bè, Thành phố Hồ Chí Minh, Thà
         Minh', 183, False, 687191.3506874996, 1185536.8915005864)
         ...
         Total: 1282
```

Let's now place all this into a Pandas dataframe.

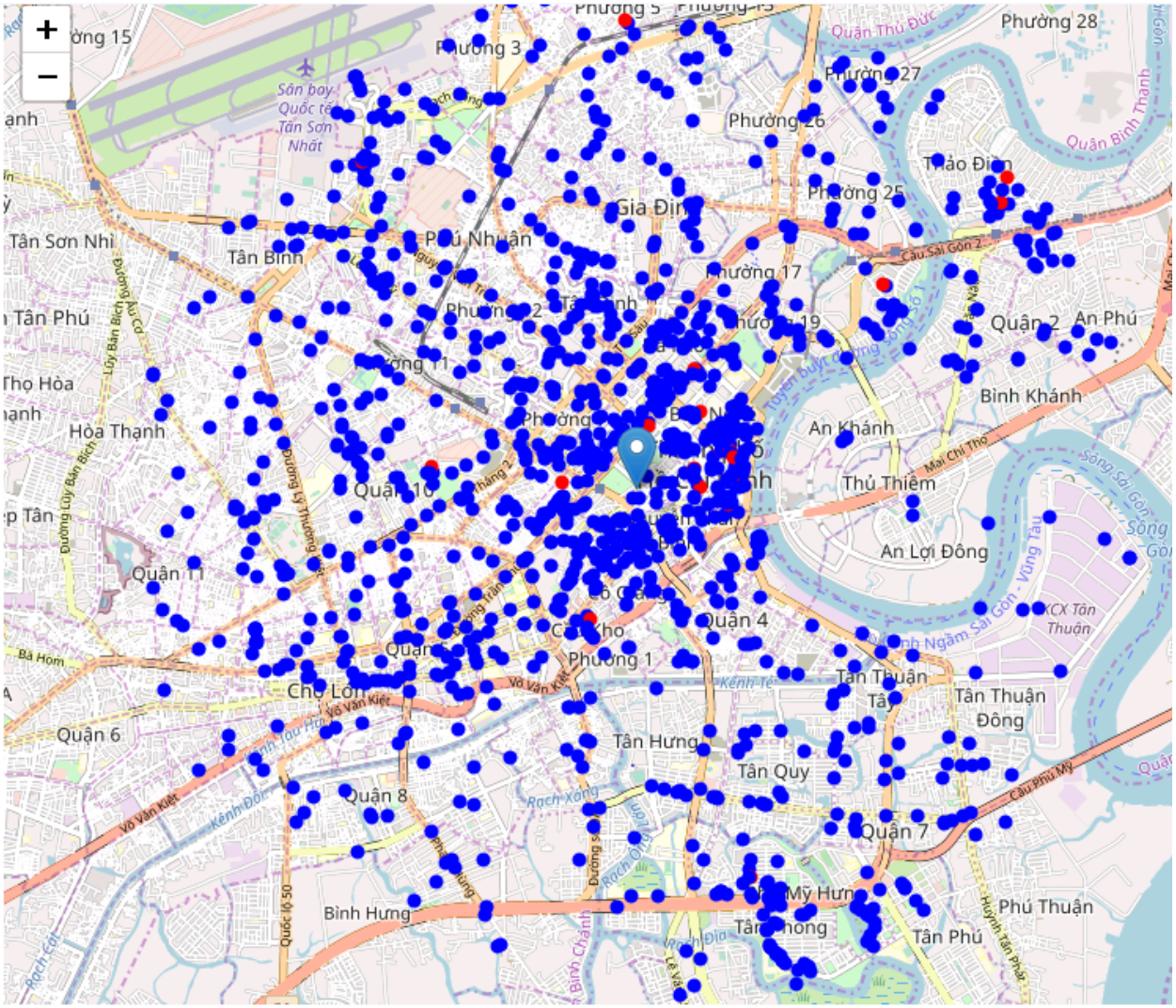
In [65]:
```
df_restaurants_hcm = pd.DataFrame(restaurants.values())
columns=['IDs',
                                    'NAME',
                                    'Latitude',
                                    'Longitude',
                                    'Address',
                                    'Coordinate',
                                    'Is_Italian',
                                    'X',
                                    'Y']
# df_restaurants_hcm.rename(columns = columns)
df_restaurants_hcm.columns = columns
df_restaurants_hcm.head()
```

Out[65]:

	IDs	NAME	Latitude	Longitude	Address	Coordinate	Is_Italian	X	Y
0	4d92f441922c6ea8b8955a79	Edward's Bistro	10.719832	106.700666	Hoàng Anh Gia Lai 3 D1-09 New Saigon, Tp Hồ ...	195	False	685986.090037	1.185515e+06
1	52d668d8498e659c75788cc7	Hiền Lành Quán CN2	10.722659	106.701273	Việt Nam	255	False	686050.804538	1.185828e+06
2	5b5b1796e55d8b002c495933	Hong Kong Restaurant	10.723514	106.699715	Kenton, Thành phố Hồ Chí Minh, Thành phố Hồ Ch...	262	False	685879.843161	1.185922e+06
3	58a835f992ca4c07d513a30e	Cánh Quạt	10.718750	106.699100	Lê Văn Lương, Thành phố Hồ Chí Minh	272	False	685815.476599	1.185394e+06
4	54bb1228498e0b8a26dd558c	Phở Kim Hưng	10.722993	106.709824	70 Ly Long Tuong	221	False	686985.984287	1.185870e+06

Let's now see all the collected restaurants in our area of interest on map, and let's also show Italian restaurants in different color.

In [94]:
```
map_HoChiMinh = folium.Map(location=HoChiMinh_center, zoom_start=13)
folium.Marker(HoChiMinh_center, popup='Ben Thanh Market').add_to(map_HoChiMinh)

for i in range(0,len(df_restaurants_hcm)-1):

    lat = float(df_restaurants_hcm["Latitude"][i])
    lon = float(df_restaurants_hcm["Longitude"][i])
    is_italian = df_restaurants_hcm["Is_Italian"][i]
    color = 'red' if is_italian else 'blue'
    folium.CircleMarker([lat, lon], radius=3, color=color, fill=True, fill_color=color, fill_opacity=1).add_to(map_HoChiMinh)
map_HoChiMinh
```



Looking good. So now we have all the restaurants in area within few kilometers from Ben Thanh Market, and we know which ones are Italian restaurants! We also know which restaurants exactly are in vicinity of every neighborhood candidate center.

This concludes the data gathering phase - we're now ready to use this data for analysis to produce the report on optimal locations for a new Italian restaurant!

III. METHODOLOGY

In this project we will direct our efforts on detecting areas of **Ho Chi Minh** that have low restaurant density, particularly those with low number **of Italian restaurants**. We will limit our analysis to area ~6km around city center. In first step we have collected the required data: location and type (category) of every restaurant within 6km from **Ben thanh Market** . We have also identified Italian restaurants (according to Foursquare categorization).

Second step in our analysis will be calculation and exploration of 'restaurant density' across different areas of **Ho Chi Minh** - we will use heatmaps to identify a few promising areas close to center with low number of restaurants in general (and no Italian restaurants in vicinity) and focus our attention on those areas.

In third and final step we will focus on most promising areas and within those create clusters of locations that meet some basic requirements established in discussion with stakeholders: we will take into consideration locations with no more than two restaurants in radius of 250 meters, and we want locations without Italian restaurants in radius of 400 meters. We will present map of all such locations but also **create clusters (using k-means clustering)** of those locations to identify general zones / neighborhoods / addresses which should be a starting point for final 'street level' exploration and search for optimal venue location by stakeholders.

IV. Analysis

Let's perform some basic explanatory data analysis and derive some additional info from our raw data. First let's count the number of restaurants in every area candidate:

In [95]:
```
location_restaurants_count = [len(res) for res in location_restaurants]

df_locations['Restaurants in area'] = location_restaurants_count

print('Average number of restaurants in every area with radius=300m:', np.array(location_restaurants_count).mean())

df_locations.head(10)
```

```
Average number of restaurants in every area with radius=300m: 3.0494505494505493
```

Out[95]:

	Address	Latitude	Longitude	X	Y	Distance from center	Restaurants in area
0	Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành p...	10.721284	106.677557	683457.385186	1.185662e+06	5992.495307	0
1	Phạm Hùng, Khu 6, Khu Dân cư T30, Xã Bình Hưng...	10.721254	106.683042	684057.385186	1.185662e+06	5840.376700	0
2	Đường số 1, Khu 6, Xã Bình Hưng, Huyện Bình Ch...	10.721225	106.688527	684657.385186	1.185662e+06	5747.173218	0
3	Đường A, Khu dân cư Phước Kiểng, Khu Dân cư Ph...	10.721195	106.694012	685257.385186	1.185662e+06	5715.767665	0
4	Xã Phước Kiển, Huyện Nhà Bè, Thành phố Hồ Chí ...	10.721165	106.699497	685857.385186	1.185662e+06	5747.173218	4
5	Xã Phước Kiển, Huyện Nhà Bè, Thành phố Hồ Chí ...	10.721135	106.704981	686457.385186	1.185662e+06	5840.376700	0
6	Trường Đinh Thiện Lý, 80, Nguyễn Đức Cảnh, Khu...	10.721105	106.710466	687057.385186	1.185662e+06	5992.495307	13
7	Cầu Xóm Củi, Đường Nguyễn Văn Linh, Xã Bình Hư...	10.726026	106.669356	682557.385186	1.186181e+06	5855.766389	0
8	Xã Bình Hưng, Huyện Bình Chánh, Thành phố Hồ C...	10.725996	106.674841	683157.385186	1.186181e+06	5604.462508	0
9	Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành p...	10.725967	106.680326	683757.385186	1.186181e+06	5408.326913	2

OK, now let's calculate the distance to nearest Italian restaurant from every area candidate center (not only those within 300m - we want distance to closest one, regardless of how distant it is).

In [96]:
```
distances_to_italian_restaurant = []

for area_x, area_y in zip(xs, ys):
    min_distance = 10000
    for res in italian_restaurants.values():
        res_x = res[7]
        res_y = res[8]
        d = calc_xy_distance(area_x, area_y, res_x, res_y)
        if d<min_distance:
            min_distance = d
    distances_to_italian_restaurant.append(min_distance)

df_locations['Distance to Italian restaurant'] = distances_to_italian_restaurant
```

In [97]:
```
df_locations.head(10)
```

Out[97]:

	Address	Latitude	Longitude	X	Y	Distance from center	Restaurants in area	Distance to Italian restaurant
0	Khu 6, Xã Bình Hưng, Huyện Bình Chánh, Thành p...	10.721284	106.677557	683457.385186	1.185662e+06	5992.495307	0	3389.717281
1	Phạm Hùng, Khu 6, Khu Dân cư T30, Xã Bình Hưng...	10.721254	106.683042	684057.385186	1.185662e+06	5840.376700	0	2831.896823
2	Đường số 1, Khu 6, Xã Bình Hưng, Huyện Bình Ch...	10.721225	106.688527	684657.385186	1.185662e+06	5747.173218	0	2295.451151
3	Đường A, Khu dân cư Phước Kiểng, Khu Dân cư Ph...	10.721195	106.694012	685257.385186	1.185662e+06	5715.767665	0	1799.597832
4	Xã Phước Kiển, Huyện Nhà Bè, Thành phố Hồ Chí ...	10.721165	106.699497	685857.385186	1.185662e+06	5747.173218	4	1388.527539

And Average distance to closest

In [98]:
```
print('Average distance to closest Italian restaurant from each area center:', df_locations['Distance to Italian restaurant'].mean())
```

Average distance to closest Italian restaurant from each area center: 1759.8628823674076

OK, so on average Italian restaurant can be found within ~1.6 km from every area center candidate. That's fairly close, so we need to filter our areas carefully!

Let's crete a map showing heatmap / density of restaurants and try to extract some meaningfull info from that. Also, let's show borders of **Ho Chi Minh** boroughs on our map and a few circles indicating distance of 1km, 2km and 3km from **Ben Thanh Market.**

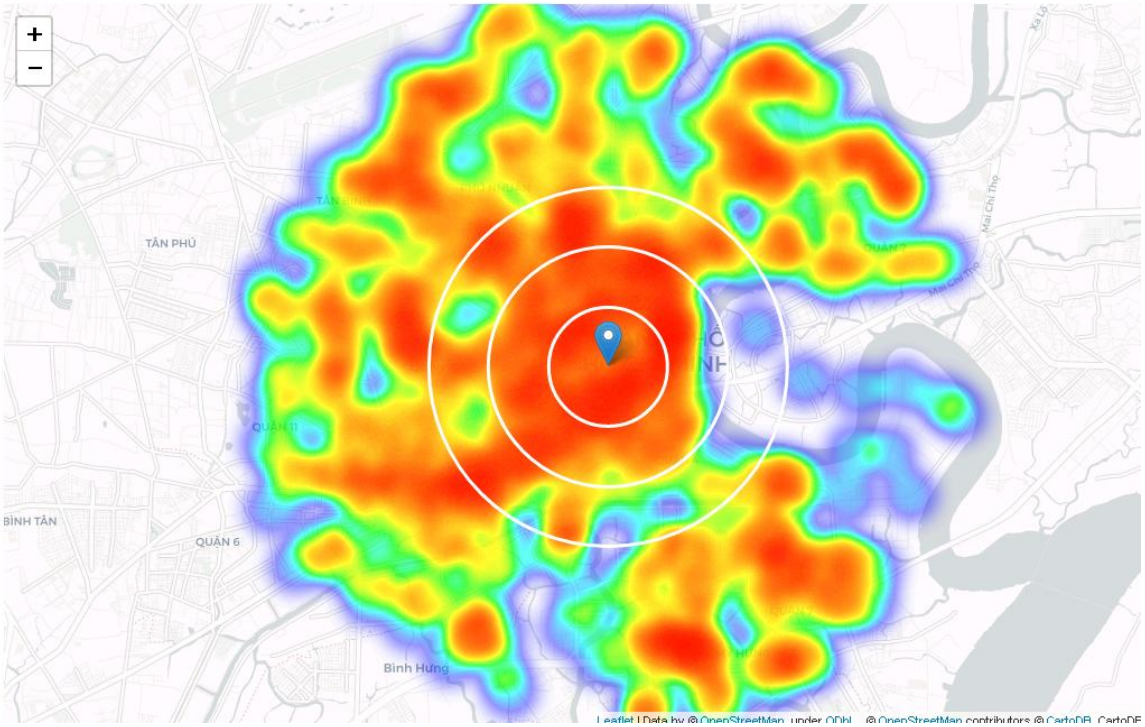
In [101]:
```
restaurant_latlons = [[res[2], res[3]] for res in restaurants.values()]

italian_latlons = [[res[2], res[3]] for res in italian_restaurants.values()]
```

In [103]:
```
from folium import plugins
from folium.plugins import HeatMap

map_HoChiMinh = folium.Map(location=HoChiMinh_center, zoom_start=13)
folium.TileLayer('cartodbpositron').add_to(map_HoChiMinh)
HeatMap(restaurant_latlons).add_to(map_HoChiMinh)
folium.Marker(HoChiMinh_center).add_to(map_HoChiMinh)
folium.Circle(HoChiMinh_center, radius=1000, fill=False, color='white').add_to(map_HoChiMinh)
folium.Circle(HoChiMinh_center, radius=2000, fill=False, color='white').add_to(map_HoChiMinh)
folium.Circle(HoChiMinh_center, radius=3000, fill=False, color='white').add_to(map_HoChiMinh)
map_HoChiMinh
```

We will get Heatmap

```
map_HoChiMinh
```

Out[103]:

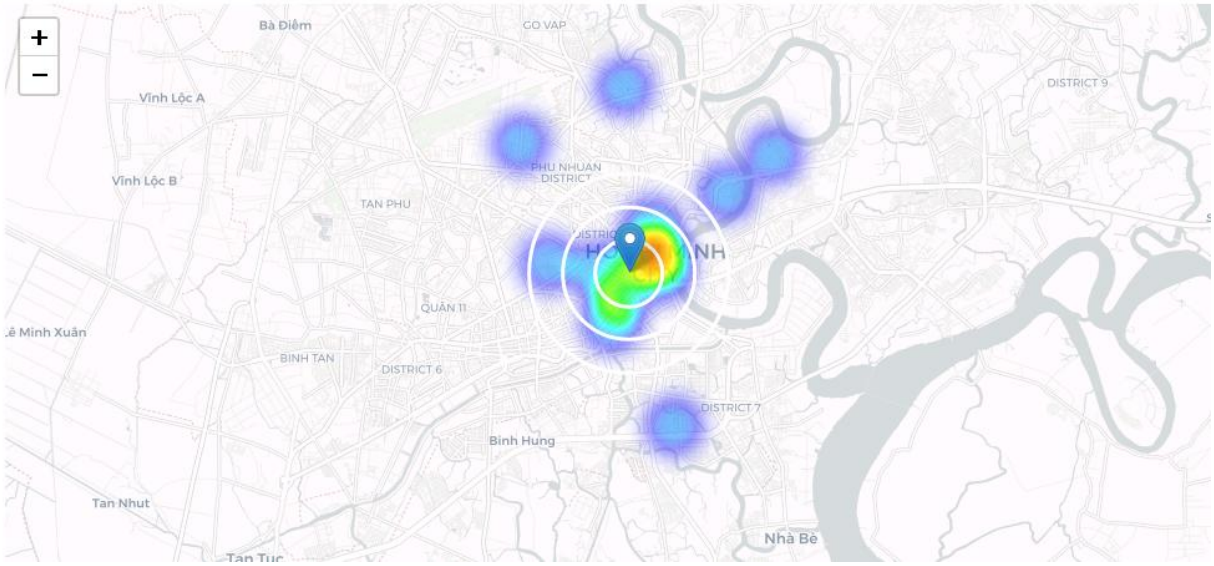


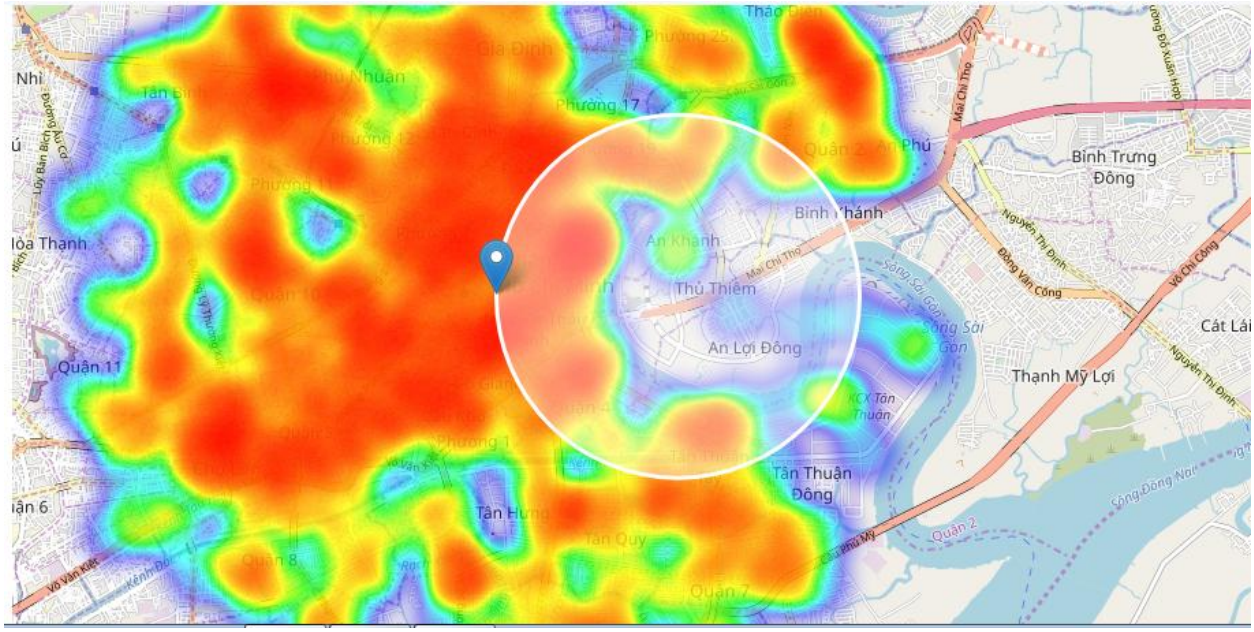
Looks like a few pockets of low restaurant density closest to city center can be found. Let's create another heatmap map showing heatmap/density of Italian restaurants only.

In [104]:

```
map_HoChiMinh = folium.Map(location=HoChiMinh_center, zoom_start=13)
folium.TileLayer('cartodbpositron').add_to(map_HoChiMinh) #cartodbpositron cartodbdark_matter
HeatMap(italian_latlons).add_to(map_HoChiMinh)
folium.Marker(HoChiMinh_center).add_to(map_HoChiMinh)
folium.Circle(HoChiMinh_center, radius=1000, fill=False, color='white').add_to(map_HoChiMinh)
folium.Circle(HoChiMinh_center, radius=2000, fill=False, color='white').add_to(map_HoChiMinh)
folium.Circle(HoChiMinh_center, radius=3000, fill=False, color='white').add_to(map_HoChiMinh)
map_HoChiMinh
```

Out[104]:





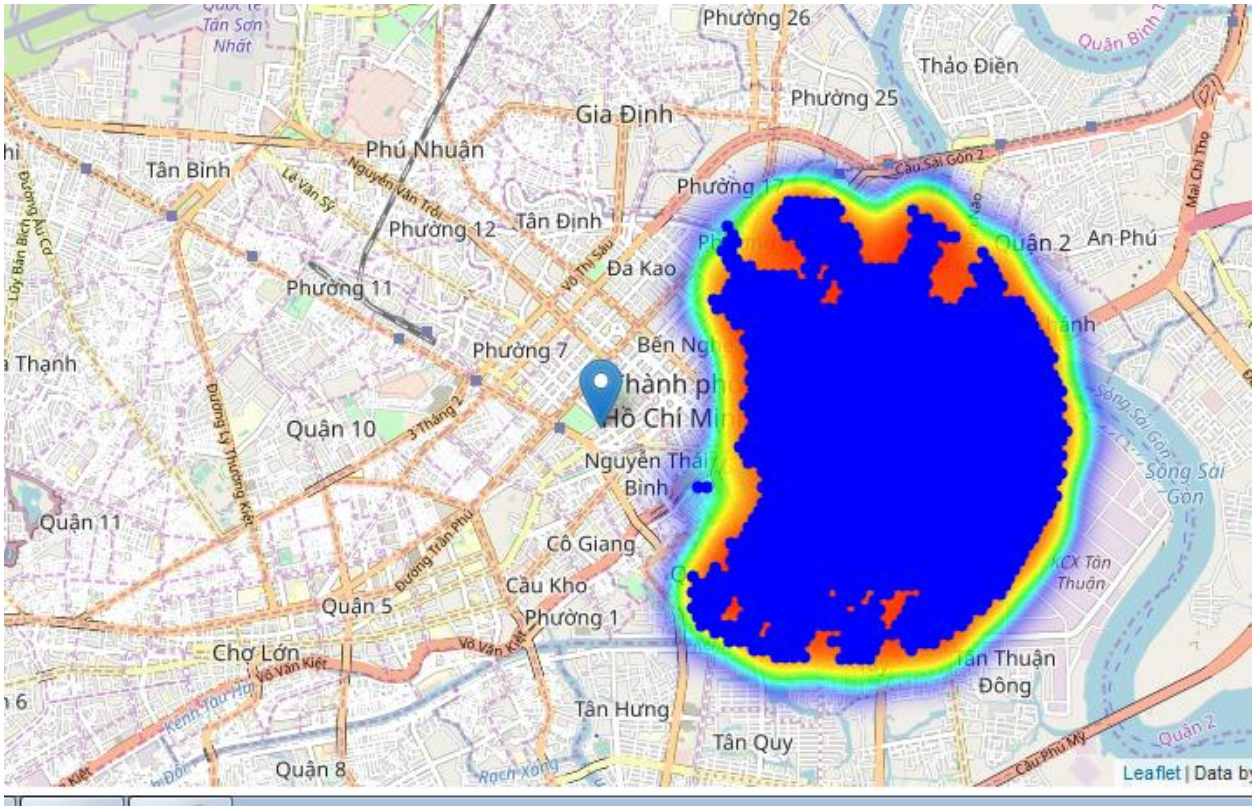
this nicely covers all the pockets of low restaurant density in An Khanh and Thu Thiêm closest to Ben Thanh Market .

Let's also create new, more dense grid of location candidates restricted to our new region of interest (let's make our location candidates 100m appart).

Now let's calculate two most important things for each location candidate: **number of restaurants in vicinity** (we'll use radius of **250 meters**) and **distance to closest Italian restaurant**.

```
In [37]: good_res_count = np.array((df_roi_locations['Restaurants nearby']<=2))
         print('Locations with no more than two restaurants nearby:', good_res_count.sum())

         good_ita_distance = np.array(df_roi_locations['Distance to Italian restaurant']>=400)
         print('Locations with no Italian restaurants within 400m:', good_ita_distance.sum())

         good_locations = np.logical_and(good_res_count, good_ita_distance)
         print('Locations with both conditions met:', good_locations.sum())

         df_good_locations = df_roi_locations[good_locations]
```

```
Locations with no more than two restaurants nearby: 1581
Locations with no Italian restaurants within 400m: 1832
Locations with both conditions met: 1567
```

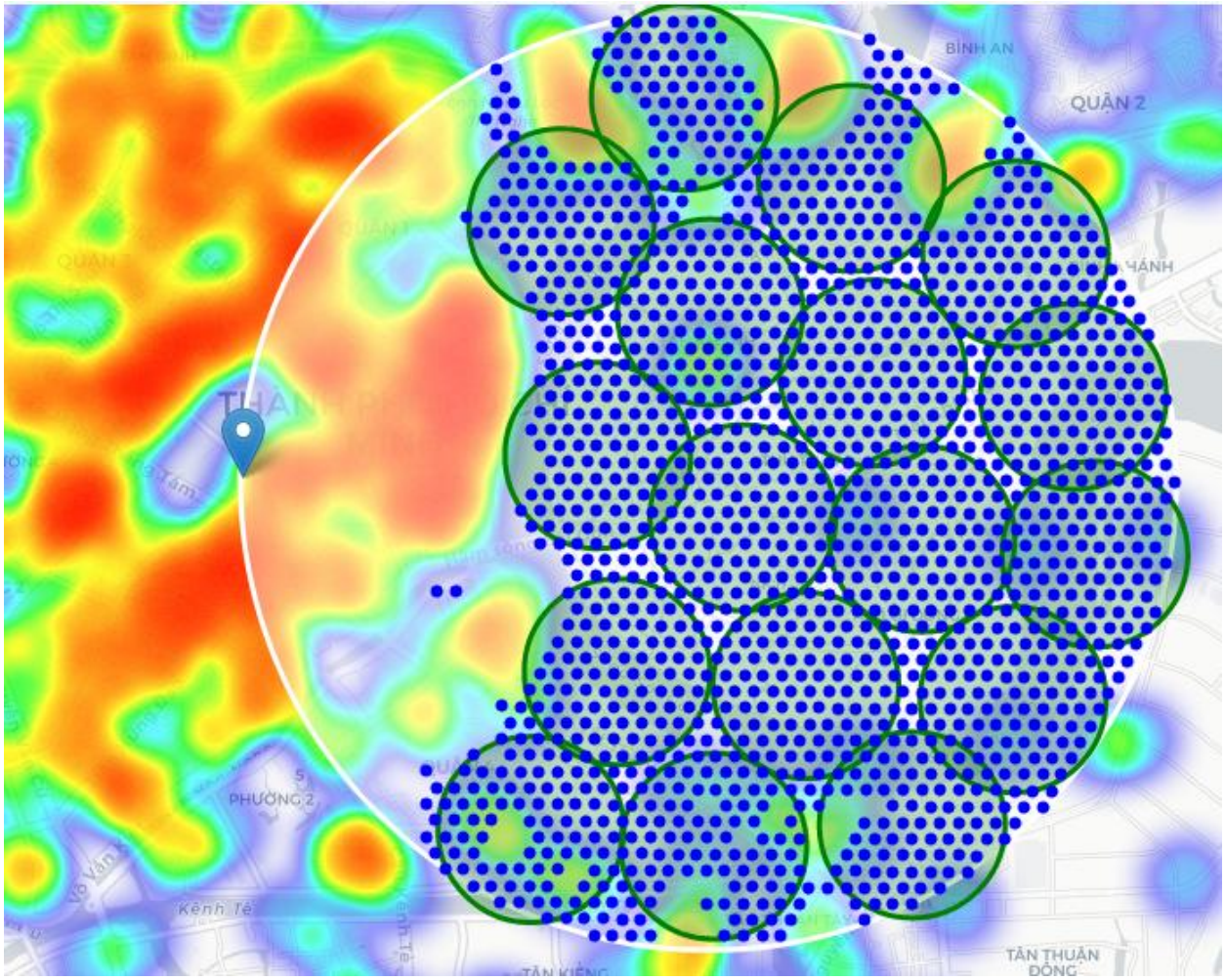


Looking good. What we have now is a clear indication of zones with low number of restaurants in vicinity, and *no* Italian restaurants at all nearby.

Let us now **cluster** those locations to create **centers of zones containing good locations**. Those zones, their centers and addresses will be the final result of our analysis.

We will use **Kmean** our clusters represent groupings of most of the candidate locations and cluster centers are placed nicely in the middle of the zones 'rich' with location candidates.

In [45]:
```
from sklearn.cluster import KMeans

number_of_clusters = 15

good_xys = df_good_locations[['X', 'Y']].values
kmeans = KMeans(n_clusters=number_of_clusters, random_state=0).fit(good_xys)

cluster_centers = [xy_to_lonlat(cc[0], cc[1]) for cc in kmeans.cluster_centers_]

map_HoChiMinh = folium.Map(location=roi_center, zoom_start=14)
folium.TileLayer('cartodbpositron').add_to(map_HoChiMinh)
HeatMap(restaurant_latlons).add_to(map_HoChiMinh)
folium.Circle(roi_center, radius=2500, color='white', fill=True, fill_opacity=0.4).add_to(map_HoChiMinh)
folium.Marker(HoChiMinh_center).add_to(map_HoChiMinh)
for lon, lat in cluster_centers:
    folium.Circle([lat, lon], radius=500, color='green', fill=True, fill_opacity=0.25).add_to(map_HoChiMinh)
for lat, lon in zip(good_latitudes, good_longitudes):
    folium.CircleMarker([lat, lon], radius=2, color='blue', fill=True, fill_color='blue', fill_opacity=1).add_to(map_HoChiMinh)

map_HoChiMinh
```



Addresses of those cluster centers will be a good starting point for exploring the neighborhoods to find the best possible location based on neighborhood specifics.

Let's see those zones on a city map without heatmap, using shaded areas to indicate our clusters.

Let's zoom in on candidate areas in **Nhà Thờ Street:**


Bến Nghé
Ba Son (dự kiến)
Nhà Hát Thành Phố (đang xây dựng)
Bến Thành
Sài Gòn
Khu phố 6
Công viên Tao Đàn
Dinh Độc Lập
Nguyễn Thái Bình
Phường 12
Cầu Ông Lãnh
Phường 9
Bến Tàu Cảnh Ngầm Sài Gòn
Hầm sông Sài Gòn
Đường Hàm Nghi
Hai Bà Trưng
Lê Duẩn
Pasteur
Nguyễn Du
Lý Tự Trọng
Lê Thánh Tôn
Võ Văn Kiệt
Đường Trần Hưng Đạo
Đường Nguyễn Thị Minh Khai


Quốc tế Tân Sơn Nhất
Phường 26
Thảo Điền
Phường 25
Gia Định
Phú Nhuận
Tân Bình
Phường 17
Phường 12
Tân Định
Đa Kao
Quận 2
An Phú
Phường 11
Phường 7
Bến Nghé
Thành phố Hồ Chí Minh
Quận 10
Nguyễn Thái Bình
Quận 11
Cô Giang
Cầu Kho
Quận 5
Phường 1
Chợ Lớn
Tân Thuận Đông
Tân Hưng
Tân Quy
Quận 8
Quận 7
Phú Mỹ Hưng
Bình Hưng
Phú Thuận
KCX Tân Thuận
Sông Sài Gòn
Cầu Sài Gòn 2
Cầu Phú Mỹ
Võ Văn Kiệt
Đường Lý Thường Kiệt
Đường Trần Phú
Nguyễn Văn Trỗi
Lê Văn Sỹ
Võ Thị Sáu
3 Tháng 2
Phạm Hùng
Kênh Tàu Hủ
Kênh Đôi
Rạch Xáng
Rạch Ông Lớn
Đường số 9A

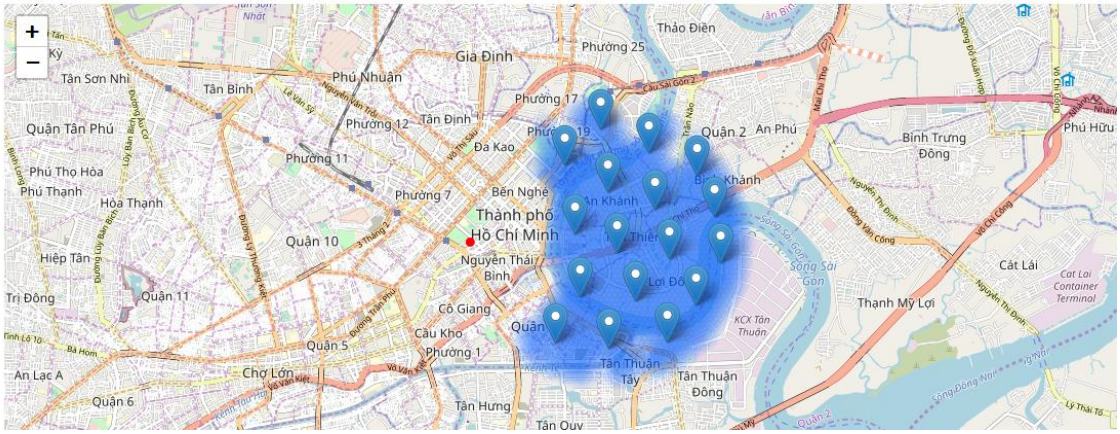
In [64]:
```
candidate_area_addresses = []
print('============================================================')
print('Addresses of centers of areas recommended for further analysis')
print('============================================================\n')
for lon, lat in cluster_centers:
    addr = get_address( lat, lon).replace(', Việt Nam', '')
    candidate_area_addresses.append(addr)
    x, y = lonlat_to_xy(lon, lat)
    d = calc_xy_distance(x, y, HoChiMinh_center_x, HoChiMinh_center_y)
    print('{}{} => {:.1f}km from Ben THành Market'.format(addr, ' '*(50-len(addr)), d/1000))
```

```
============================================================
Addresses of centers of areas recommended for further analysis
============================================================


Phường An Lợi Đông, Quận 2, Thành phố Hồ Chí Minh, 77000 => 3.4km from Ben THành Market
Phường An Lợi Đông, Quận 2, Thành phố Hồ Chí Minh, 8408 => 3.7km from Ben THành Market
Cây Bàng, Phường Thủ Thiêm, Quận 2, Thành phố Hồ Chí Minh, 8408 => 2.3km from Ben THành Market
Phường Bến Nghé, Quận 1, Thành phố Hồ Chí Minh, 77000 => 2.2km from Ben THành Market
Đường số 11, Phường An Lợi Đông, Quận 2, Thành phố Hồ Chí Minh, 71108 => 4.5km from Ben THành Market
Phường Bình An, Quận 2, Thành phố Hồ Chí Minh, NGÔ TẤT TỐ => 3.6km from Ben THành Market
Lương Định Của, Phường Thủ Thiêm, Quận 2, Thành phố Hồ Chí Minh, 8408 => 1.9km from Ben THành Market
Phường Tân Thuận Đông, Quận 7, Thành phố Hồ Chí Minh, NGÔ TẤT TỐ => 4.6km from Ben THành Market
Phường An Lợi Đông, Quận 2, Thành phố Hồ Chí Minh, 8408 => 3.2km from Ben THành Market
Hẻm 200C Xóm Chiếu, Phường 15, Quận 4, Thành phố Hồ Chí Minh, 710200 => 2.4km from Ben THành Market
Hẻm 108 Võ Duy Ninh, Phường 22, Quận Bình Thạnh, Thành phố Hồ Chí Minh, NGÔ TẤT TỐ => 3.1km from Ben THành Market
Trần Văn Khánh, Phường Tân Thuận Đông, Quận 7, Thành phố Hồ Chí Minh, 8408 => 4.0km from Ben THành Market
Phường Bình Khánh, Quận 2, Thành phố Hồ Chí Minh, NGÔ TẤT TỐ => 4.3km from Ben THành Market
Cầu Kênh 1, Mai Chí Thọ, Phường Thủ Thiêm, Quận 2, Thành phố Hồ Chí Minh, 8408 => 2.7km from Ben THành Market
Đại lộ vòng cung, Phường An Khánh, Quận 2, Thành phố Hồ Chí Minh, 77000 => 2.7km from Ben THành Market
```

This concludes our analysis. We have created 17 addresses representing centers of zones containing locations with low number of restaurants and no Italian restaurants nearby, all zones being fairly close to city center (all less than 4km from Bến Thành Market, and about half of those less than 2km from Bến Thành Market). Although zones are shown on map with a radius of ~500 meters (green circles), their shape is actually very irregular and their centers/addresses should be considered only as a starting point for exploring area neighborhoods in search for potential restaurant locations. Most of the zones are located in ****Phường An Lợi Đông, Quận 2, Thành phố Hồ Chí Minh, 77000 => 3.4km from Ben THành Market**** , which we have identified as interesting due to being popular with tourists, fairly close to city center and well connected by public transport.

```
folium.Circle(HoChiMinh_center, radius=50, color='red', fill=True, fill_color='red', fill_opacity=1).add_to(map_HoChiMinh)
for lonlat, addr in zip(cluster_centers, candidate_area_addresses):
    folium.Marker([lonlat[1], lonlat[0]], popup=addr).add_to(map_HoChiMinh)
for lat, lon in zip(good_latitudes, good_longitudes):
    folium.Circle([lat, lon], radius=250, color='#0000ff00', fill=True, fill_color='#0066ff', fill_opacity=0.05).add_to(map_HoChiMinh)
map_HoChiMinh
```

Out[66]:



V. Results and Discussion

Our analysis shows that although there is a great number of restaurants in Hồ Chí Minh City (~1282 in our initial area of interest which was 12x12km around Bến Thành Market), there are pockets of low restaurant density fairly close to city center. Highest concentration of restaurants was detected north and west from Bến Thành Market, so we focused our attention to areas east, corresponding to **District 2**. Our attention was focused on **District 2** . which offer a combination of popularity among tourists, closeness to city center, strong socio-economic dynamics *and* a number of pockets of low restaurant density.

After directing our attention to this more narrow area of interest (covering approx. 5x5km east from Bến Thành Market) we first created a dense grid of location candidates (spaced 100m appart); those locations were then filtered so that those with more than two restaurants in radius of 250m and those with an Italian restaurant closer than 400m were removed.

Those location candidates were then clustered to create zones of interest which contain greatest number of location candidates. Addresses of centers of those zones were also generated using reverse geocoding to be used as markers/starting points for more detailed local analysis based on other factors.

Result of all this is 17 zones containing largest number of potential new restaurant locations based on number of and distance to existing venues - both restaurants in general and Italian restaurants particularly. This, of course, does not imply that those zones are actually optimal locations for a new restaurant! Purpose of this analysis was to only provide info on areas close to Ho Chi Minh center but not crowded with existing restaurants (particularly Italian) - it is entirely possible that there is a very good reason for small number of restaurants in any of those areas, reasons which would make them unsuitable for a new restaurant regardless of lack of competition in the area. Recommended zones should therefore be considered only as a starting point for more detailed analysis which could eventually result in location which has not only no nearby competition but also other factors taken into account and all other relevant conditions met.

VI. Conclusion

Purpose of this project was to identify Hồ Chí Minh areas close to center with low number of restaurants (particularly Italian restaurants) in order to aid stakeholders in narrowing down the search for optimal location for a new Italian restaurant. By calculating restaurant density distribution from Foursquare data we have first identified general boroughs that justify further analysis (**District 2**), and then generated extensive collection of locations which satisfy some basic requirements regarding existing nearby restaurants. Clustering of those locations was then performed in order to create major zones of interest (containing greatest number of potential locations) and addresses of those zone centers were created to be used as starting points for final exploration by stakeholders.

Final decission on optimal restaurant location will be made by stakeholders based on specific characteristics of neighborhoods and locations in every recommended zone, taking into consideration additional factors like attractiveness of each location (proximity to park or water), levels of noise / proximity to major roads, real estate availability, prices, social and economic dynamics of every neighborhood etc.

I will be providing a other supplementary Inferential Statics in the future about on these data collected and also update in a new notebook using other categories. For now, this completes the requirements for this task.

Thank you.

Hoang Tien Thuy

Created For: COURSERA **IBM Applied Data Science Capstone Project**

List of my FourSquare Data collection saved in Github can be found in the following location:

https://github.com/hoangtienthuy83/Coursera_Capstone/blob/master/restaurants_hochiminh.csv