Search Service Test

Background

We offer parking space location and ‘park and ride’ location data.
There is an api with two endpoints:

/api/search

Returns a ranked list of parking space and ‘park and ride’ locations within a given latitude and longitude.
It uses third party services to rank the parking space and ‘park and ride’ locations.

- *ParkingSpaceHttpService* - A mock http service.
- *ParkAndRideSDK* - A mock SDK dependency.

It always ranks ‘park and ride’ locations above parking space locations.

Example Response: (see below 1.)

/api/details

Returns an **unranked** list of parking space and ‘park and ride’ location details within a given latitude and longitude.

Example Response: (see below 2.)

Task

Finish the controller and gateways to complete the API and test the implementation. (You can refactor any of the code as you see fit)

Part 1 (45 mins approx)
Complete the ***/api/details*** endpoint.

The description for a ParkingSpace should follow the convention:
"Parking space with {{no_of_spaces}} bays: {{space_details}}"

The location_name for a ParkingSpace should follow the convention:
"{{street_name}}, {{city}"

The location_name for a ParkAndRide should follow the convention:
"{{location_description}}"

The description for a ParkAndRide should follow the convention:
"Park and Ride to {{attraction_name}}. (approx {{minutes_to_destination}} minutes to destination)"

With the bracketed info coming from the database model.

The test *test/Feature/SearchControllerTest::testDetailsEndpoint()* can be used to verify your implementation.
(To run the test *$ vendor/bin/phpunit tests/Feature/SearchControllerTest.php --filter testDetailsEndpoint*)

Part 2 (60 mins approx)
Finish the ***/api/search*** endpoint. This includes implementing the *ParkingSpaceRankerGateway*. *ParkAndRideRankerGateway* has already been implemented but you can refactor this as you wish.

ParkingSpaceHttpService takes a json encoded string "*[1,3,4]"* which are the ID's of the ParkingSpace models and will return a json string of the ID's in the ranked order eg. *"[1,4,3]".*

Each gateway should:

- Call it's respective ranking service.
- Log out the result.
- Use the ranking to order its collection of models.
- Return the ranked collection of models.

The search endpoint should return both parking spaces and 'park and ride' locations. (Remember that 'park and ride' locations are always ranked higher than parking spaces)

Note: Both the *ParkingSpaceHttpService* and *ParkAndRideSDK* can become very slow. Your service should handle when ThirdParty services are taking too long and always return a result.

There is a test in ParkingSpaceRankerGatewayTest that you can use to test your implementation.

Part 3 (15 mins approx)
Complete the testSearchEndpointUsesGatewayRanking test in SearchControllerTest to test the expected behaviour of the search endpoint.

Important Notes

You should only need to look in the tests/ app/ and database/ folders
Within app/ you can ignore:
*Console/**
*Exceptions/**
*Http/Middleware/**
Http/Kernel.php
*Providers/**

Do not change any code within the *app/ThirdParty/* folder

Example Responses

Example Response 1. (***/api/search***)


```
[
        {
            "id": 3,
            "name": "Park n Ride 3",
            "lat": 0.1,
            "lng": 0.1,
            "owner": {
                "name": "bob",
            }
        },
        {
            "id": 1,
            "name": "Park n Ride 1",
            "lat": 0.1,
            "lng": 0.1,
            "owner": {
                "name": "phil",
            }
        },
        {
            "id": 1,
            "name": "Parking Space 1",
            "lat": 0.1,
            "lng": 0.1,
            "owner": {
                "name": "matt",
            }
        }
]
```


Example Response 2. (***/api/details***)


```
[
   {
      "description": "Park and Ride to disneyland. (approx 10 minutes to destination)",
      "location_name": "TCR"
   },
   {
      "description":  "Parking space with 2 bays: Driveway off street",
      "location_name":  "Oxford Street, London"
   }
]
```