

```

1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace Exer1Solution
6 {
7     /// <summary>
8     /// Provides capabilities for managing a customers itinerary.
9     /// </summary>
10    [reference | 0 changes | 0 authors, 0 changes | 0 exceptions]
11    public class ItineraryManager
12    {
13        private readonly IDataStore _dataStore;
14        private readonly IDistanceCalculator _distanceCalculator;
15
16        [reference | 0 changes | 0 authors, 0 changes | 0 exceptions]
17        public ItineraryManager()
18        {
19            _dataStore = new SqlAgentStore(ConfigurationManager.ConnectionStrings["SqlDbConnection"].ConnectionString);
20            _distanceCalculator = new GoogleMapsDistanceCalculator(ConfigurationManager.AppSettings["GoogleMapsApiKey"]);
21        }
22
23        /// <summary>
24        /// Calculates a quote for a customers itinerary from a provided list of airline providers.
25        /// </summary>
26        /// <param name="itineraryId">The identifier of the itinerary</param>
27        /// <param name="priceProviders">A collection of airline price providers.</param>
28        /// <returns>A collection of quotes from the different airlines.</returns>
29        [reference | 0 changes | 0 authors, 0 changes | 0 exceptions]
30        public IEnumerable<Quote> CalculateAirlinePrices(int itineraryId, IEnumerable<IAirlinePriceProvider> priceProviders)
31        {
32            var itinerary = _dataStore.GetItineraryAsync(itineraryId).Result;
33            if (itinerary == null)
34                throw new InvalidOperationException();
35
36            List<Quote> results = new List<Quote>();
37            Parallel.ForEach(priceProviders, provider =>
38            {
39                var quotes = provider.GetQuotes(itinerary.TicketClass, itinerary.Waypoints);
40                foreach (var quote in quotes)
41                    results.Add(quote);
42            });
43            return results;
44        }
45
46        /// <summary>
47        /// Calculates the total distance traveled across all waypoints in a customers itinerary.
48        /// </summary>
49        /// <param name="itineraryId">The identifier of the itinerary</param>
50        /// <returns>The total distance traveled.</returns>
51        [reference | 0 changes | 0 authors, 0 changes | 0 exceptions]
52        public async Task<double> CalculateTotalTravelDistanceAsync(int itineraryId)
53        {
54            var itinerary = await _dataStore.GetItineraryAsync(itineraryId);
55            if (itinerary == null)
56                throw new InvalidOperationException();
57            double result = 0;
58            for (int i = 0; i < itinerary.Waypoints.Count - 1; i++)
59            {
60                result = result + _distanceCalculator.GetDistanceAsync(itinerary.Waypoints[i],
61                    itinerary.Waypoints[i + 1]).Result;
62            }
63            return result;
64        }
65
66        /// <summary>
67        /// Loads a Travel agents details from Storage
68        /// </summary>
69        /// <param name="id">The id of the travel agent.</param>
70        /// <param name="updatedPhoneNumber">If set updates the agents phone number.</param>
71        /// <returns>The travel agent if located, otherwise null.</returns>
72        [reference | 0 changes | 0 authors, 0 changes | 0 exceptions]
73        public TravelAgent FindAgent(int id, string updatedPhoneNumber)
74        {
75            var agentDao = _dataStore.GetAgent(id);
76            if (agentDao == null)
77                return null;
78            if (!string.IsNullOrEmpty(updatedPhoneNumber))
79            {
80                agentDao.PhoneNumber = updatedPhoneNumber;
81                _dataStore.UpdateAgent(id, agentDao);
82            }
83            return Mapper.Map<TravelAgent>(agentDao);
84        }
85    }
86 }

```

Issues found:

- Line 17 and 18, `_dataStore` and `_distanceCalculator` could be injected in constructor.
- Line 27, the method should be a `"Async Task<IEnumerable<Quote>>"` method
- Line 29 and 56, should insert `"await"` before `"="`
- Line 31 and 52, should throw a `"NullReferenceException"`
- Line 70, change the `"agentDao"` to `"agentDto"`;
- Line 68, the method `FindAgent` should divided into two methods, which are `FindAgent` and `UpdateAgent`
- Line 34, `"Parallel..."` could lead to thread safety issues, just use `"ForEach..."`
- Line 57, `(...).Result` could not get the expected value, should cast to the type of the return value of `GetDistanceAsync`