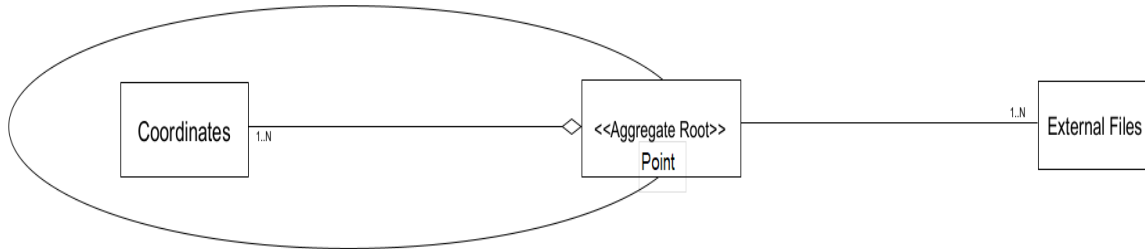


Lab8 aggregateTwo

Aggregate 2: This aggregate is present in the codestarAI/coronavirus-tracker-api/blob/app/coordinates.py



This aggregate in concept works the same way as the above aggregate, however the objects that are in play are different. The **Coordinate** object is correlated to its root object it is an aggregation of the root object and it can exist without the root object. In order for external files to access a coordinate they have to make reference to the root object.

Aggregate 2 code:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

class Coordinates:
    def __init__(self, lat_long):
        self.lat_long = lat_long

    if __name__ == "__main__":
        us_lat, us_long = 45, 60
        us = Point(us_lat, us_long)
        us_coord = Coordinates(us)
```

In the code above the **Coordinates** point is an internal object of the aggregate and the **Point** object is the aggregate root. This root object is accessed and referenced from an external file in the same logically manner of aggregate one, however the file location is different it is present in the directory.

codestarAI/coronavirus-tracker-api/blob/master/app/services/location/csbs.py

```

1  """app.services.location.csbs.py"""
2  import csv
3  import logging
4  from datetime import datetime
5
6  from asyncache import cached
7  from cachetools import TTLCache
8
9  from ...caches import check_cache, load_cache
10 from ...coordinates import Coordinates
11 from ...location.csbs import CSBSLocation
12 from ...utils import httputils
13 from . import LocationService
14
15 LOGGER = logging.getLogger("services.location.csbs")
16
17
18 class CSBSLocationService(LocationService):
19     """
20     Service for retrieving locations from csbs
21     """
22
23     async def get_all(self):
24         # Get the locations.
25         locations = await get_locations()
26         return locations
27
28     async def get(self, loc_id): # pylint: disable=arguments-differ
29         # Get location at the index equal to the provided id.
30         locations = await self.get_all()
31         return locations[loc_id]
32
33
34 # Base URL for fetching data
35 BASE_URL = "https://facts.csbs.org/covid-19/covid19_county.csv"
36

```

If you look at line 10 this line imports the coordinates method from the coordinates file. This will access the method root object method point and will then use the coordinate aggregate internal object coordinates. The whole transition of this process follows the aggregate's structure and formation. The external file csbs.py makes a reference to the root object because the coordinates method is associated with this root object to process, and this forms an aggregation. The same reasoning of cardinality that applies for aggregate one also applies to this aggregate.

Student Name: Abdullah Sarvari

GitHub User ID (case sensitive): codestarAI

Chosen Project ID: Proj_1

First Pull Request Number:

Link to the First Pull Request:

Second Pull Request Number:

Link to the Second Pull Request: