ProjectFlow_Analytics.txt

When user wants to know the most Energy Efficient Path

---

1. User requests most efficient Path from Point1 to Point2

2. UserApplication sends Point1 and Point2 to AllPathsFinderService.

3. AllPathsFinderService (all_paths.py)finds all the paths between Point1 and Point2

4. AllPathsFinderService returns all the paths and total distance of each path (all_paths.csv) and a dictionary(path_vertices_coordinates.csv) of the coordinates of each path vertex*.

5. UserApplication forwards all the possible paths(all_paths.csv) and the dictionary(path_vertices_coordinates.csv) to PollingAndHistoryService.

6. PollingAndHistoryService gives the ambient light values along each of the paths (ambient_light_all_paths.csv)

7. UserApplication integrates the values along each path and divides by total distance to find average ambient light along each path.

8. Path with highest average ambient light wins.

*path vertex - A path vertex is a vertex in the graph representation of the map. For reference see mapGraph.png (A to U are the path vertices)

-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-

---

When user gives Live Ambient Light Data

---

1. UserApplication receives request for Live Ambient Light Data

2. UserApplication Checks policy to see if it can give Data. If no, request is ignored.

3. If yes, user sends current GPS location and LiveAmbientLightData to requesting service.

4. If RequestingService is PollingAndHistoryService, this service finds the closest waypoint** to the user's current location and updates the AmbientLightValue at this waypoint.

** waypoint - Points between each pair of adjacent path vertices, interspaced by a distance of 5 meters. For reference see N_to_M.png (The Red Pins are all waypoints between N & M interspaced by 5 meters)

-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-