1 Algorithm Design

The most relevant algorithm of the myTaxiService application is the one implementing the queue management. It is important to optimize its implementation as long as it is the most complex and the one that distinguishes the system.

- INIT: There is a class that initializes the queues of the taxi drivers, starting assigning one taxi to all the areas and then, basing on the statistics data, the queues are filled with a number of taxi drivers that is greater or equal than the number of the requests expected in that area.
  COMPLEXITY: O(T) with T=number of taxis (randomly assigned)
- REQUEST: The taxi drivers are now able to take requests in the designated area. When a taxi driver takes a request the system sets his availability to false and it stops monitoring his position. In fact, the server constantly monitors the position of all the taxi drivers that are declared available, connecting to the GPS on their devices, and uses this values in order to manage the taxi distribution in the different areas. If a taxi driver does not accept the request of a user, whether because he is busy or because he has not seen the call in the first minute, the system moves him to the last position of the queue, forwarding the request to the second one.
  COMPLEXITY: O(A+Q) with A=number of areas and Q=number of taxi in queue in that area(find the area where the passenger is in and take the first of the queue of that area)
- QUEUES: The areas of the city are represented by a graph with an array of adjacencies and the queue of the taxi drivers, whose position is within its boundaries, is assigned to each area. It is a useful representation in order to decide how to distribute the taxis, in fact it is possible that an area is occupied by a number of taxi that is equal to the threshold of the maximum taxis that can be present in that area. In this case the system does not put a recently arrived taxi in the queue of that area, but advise the taxi driver that he will be moved to an adjacent area that has the minimum number of taxi, searching recursively in the adjacencies of the areas. As it is necessary to guarantee that there should always be at least one taxi driver in each area, when it occurs that the last taxi driver leaves an area, the system instantly notifies the last taxi driver of the queue that he has to move from the most populated adjacent area to the needy area.
  COMPLEXITY: O(A)

ADD DOMAIN PROPERTY: ALMENO UN TAXI ACCETTA LA RICHIESTA.