Unit requirements:

* There should be a way to portray a coordinate point, calculate its Euclidean distance to another point, find the next position after moving in a certain direction, and check whether if it is inside the central area.
* JSON data should be able to be read from the REST server and be deserialised into corresponding objects.
* A credit card’s details (cvv, expiry date, card number) should be able to be verified following credit card conventions.
* A list of paths that the drone takes should be able to be written out in a GeoJSON format.
* There should be a way to find out if a point is within or on the boundary of an arbitrary polygon.
* There should be a way to portray each restaurant and no-fly zone vertices as a node in a graph.

Integration requirements:

* The JSON getter subsystem should integrate with the order validation subsystem to validate any order that was received from the server
* The order validation subsystem should integrate with the graph subsystem to find the shortest path of nodes that the drone should take.
* The pathfinding algorithm should find the shortest path of nodes from node to node by integrating with the graph class.
* The JSON writer subsystem should integrate with the drone delivery and order validation subsystems to write out relevant information to different files.

System requirements:

* The system should be able to produce a flight path that maximises the sampled average number of validated pizza orders delivered before the battery is exhausted.
  + The system should be able to find the shortest distance for the drone to fly from one point to the other without entering the no fly zones
  + The system should be able to output the flight path into a file that is in accordance with the file structure specified in the documents