Traffic Drones - Java Test

Scenario:

There are two automatic drones that fly around London and report on traffic conditions. When a drone flies over a tube station, it assesses what the traffic condition is like in the area, and reports on it.

Task:

Write a simulation that has one dispatcher and two drones. Each drone should "move" independently on different threads. The dispatcher should send the coordinates to each drone detailing where the drone's next position should be. The dispatcher should also be responsible for terminating the program.

When the drone receives a new coordinate, it moves, checks if there is a tube station in the area, and if so, reports on the traffic conditions there. The data should be accessible and reusable to other components.

Notes:

- The drones have limited memory, so they can only consume ten points at a time.
- The simulation should finish @ 08:10, where the drones will receive a "SHUTDOWN" signal.
- The two drones have IDs 6043 and 5937. There is a file containing their lat/lon points for their routes. The csv file layout is **drone id,latitude,longitude,time**
- There is also a file with the lat/lon points for London tube stations. **station,lat,lon**
- Traffic reports should have the following format:
    - Drone ID
    - Time
    - Speed
    - Conditions of Traffic (HEAVY, LIGHT, MODERATE). This can be chosen randomly.

Remarks:

1. Assume that the drones follow a straight line between each point, travelling at constant speed.
2. Disregard the fact that the start time is not in synch. The dispatcher can start pumping data as soon as it has read the files.
3. A nearby station should be less than 350 meters from the drone's position.

Deliverable:

1. The assignment should be delivered as a command line program that allows the user to start the simulation, and see (on the console or log file), the communications between drones and dispatcher.

2. This is a fairly open assignment in terms of how you structure the solution. You will be judged on the overall quality of the code (simplicity, presentation, performance).
3. The task should take no more than 4 hours to do. Your solution, as well as more complex ideas, improvements, and suggestions will be discussed at the interview stage.