

Testing Log (Bugs Found)

1. The method for calculating the moves was wrong.
 - a) Problem: The old method was trying to use the length of the path array (i.e. how many positions in the list)
 - b) Result: This results in the calculated move being more than the actual
 - i. The real move should be the number of positions minus 1
 - ii. If a point is 495 moves away, the algorithm would return totalMoves = 993, which is wrong. Should be 901
 - c) Screenshots for old code and fixes

- i. Old code

The screenshot shows an IDE interface with several tabs open. The main tab displays the Java code for `RestServiceImplementation.java`. The code is part of a class that implements `RestService`. It contains a method `calcDeliveryPath` which iterates through a list of service points, adding deliveries to a returned path. A check is made to close the flight if there are no more points. The code then enters a loop where it adds back points to the path, calculates the total moves, and asserts that the drone's maximum moves are not exceeded. The IDE's status bar at the bottom right shows the file name `s2487866`, line count `330`, and encoding `UTF-8`.

```
public class RestServiceImplementation implements RestService
{
    public ReturnedPath calcDeliveryPath(ArrayList<MedicineDispatchRequest> queries, ArrayList<ServicePoint> servicePoints, ArrayList<RestrictedArea> restrictedAreas, ArrayList<Delivery> delivered)
    {
        utility.addDeliveriesToReturnedPath(drone.getId(),query.getId(),toDeliver,returnedPath);

        ...
        }

        // close flight if did anything
        if (!currentDronePath.isEmpty())
        {
            progress = true;
            Position last = currentDronePath.getLast();
            ArrayList<Position> back = utility.aStarSearch(last, currentDroneBase, restrictedAreas);
            utility.addDeliveriesToReturnedPath(drone.getId(), deliveryId, back, returnedPath);
            currentDroneMoves += back.size() - 1;
            totalMoves += back.size() - 1;
            totalCost += currentLandingAndTakeOffCost + currentDroneMoves * droneCapability.getCostPerMove();
            queryByDate.removeAll(delivered);

            assert(droneCapability.getMaxMoves() >= totalMoves); // assert for L01
        }

        while(progress && !queryByDate.isEmpty());
    }
}
```

The bottom panel shows a terminal window titled "Run Application" with the tab "R1Tests". It displays a log of application startup and configuration. The log includes entries for the ServletWebServerApplicationContext, the embedded Tomcat server, and the application itself. It also shows the start of the application and the configuration of various endpoints.

```
2026-01-04T18:41:45.169+08:00 INFO 65298 --- [IipTutorial_1] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1 ms
2026-01-04T18:41:45.515+08:00 INFO 65298 --- [IipTutorial_1] [main] o.s.b.a.w.EndpointLinksResolv...
2026-01-04T18:41:45.535+08:00 INFO 65298 --- [IipTutorial_1] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
2026-01-04T18:41:45.539+08:00 INFO 65298 --- [IipTutorial_1] [main] uk.ac.ed.acp.cz2.Application : Start
2026-01-04T18:41:46.046+08:00 INFO 65298 --- [IipTutorial_1] [on(1)-127.0.0.1] o.a.c.c.[Tomcat].[localhost].[/]
2026-01-04T18:41:46.046+08:00 INFO 65298 --- [IipTutorial_1] [on(1)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Initia...
2026-01-04T18:41:46.048+08:00 INFO 65298 --- [IipTutorial_1] [on(1)-127.0.0.1] o.s.web.servlet.DispatcherServlet : Compl...
```

- ii. Fixed c

```

18     public class RestServiceImplementation implements RestService
19     public ReturnedPath calcDeliveryPath(ArrayList<MedicineDispatchRequest> queries, ArrayList<ServicePoint> servicePoints, ArrayList<RestrictedArea> restrictedAreas, Array
20     {
21         start = droneBase;
22     }
23     else
24     {
25         start = currentDronePath.getLast();
26     }
27     end = query.getDelivery();
28
29     ArrayList<Position> toDeliver = utility.aStarSearch(start,end,restrictedAreas);
30     // For hover
31     toDeliver.add(toDeliver.getLast());
32     int movesTo = toDeliver.size()-1;
33
34     ArrayList<Position> toBase = utility.aStarSearch(end,droneBase,restrictedAreas);
35     int movesBack = toBase.size()-1;
36
37     if (toDeliver.isEmpty() || toBase.isEmpty())
38     {
39         // no valid path, treat as cannot deliver this query
40         continue;
41     }
42
43     int estimatedCurrentDroneMoves = currentDroneMoves + (movesTo + movesBack);
44     int estimatedNumberOfDeliveries = currentNumberOfDeliveries + 1;

```

2. The code for adding the hover point was wrong

- Problem: The old method is trying to add the last point from the toDeliver list without checking if that is empty or null
- Result: This results in a NullPointerException
 - If the delivery point is completely blocked, i.e. no path can be found, then an error will be returned

c) Screenshots for old code and fixes

i. Old code

```

19     public class RestServiceImplementation implements RestService
20     public ReturnedPath calcDeliveryPath(ArrayList<MedicineDispatchRequest> queries, ArrayList<ServicePoint> servicePoints, ArrayList<RestrictedArea> restrictedAreas, Array
21     {
22         start = droneBase;
23     }
24     else
25     {
26         start = currentDronePath.getLast();
27     }
28     end = query.getDelivery();
29
30     ArrayList<Position> toDeliver = utility.aStarSearch(start,end,restrictedAreas);
31     // For hover
32     toDeliver.add(toDeliver.getLast());
33     int movesTo = toDeliver.size()-1;
34
35     ArrayList<Position> toBase = utility.aStarSearch(end,droneBase,restrictedAreas);
36     int movesBack = toBase.size()-1;
37
38     if (toDeliver.isEmpty() || toBase.isEmpty())
39     {
40         // no valid path, treat as cannot deliver this query
41         continue;
42     }
43
44     int estimatedCurrentDroneMoves = currentDroneMoves + (movesTo + movesBack);
45     int estimatedNumberOfDeliveries = currentNumberOfDeliveries + 1;
46     double estimatedCurrentFlightCost = currentLandingAndTakeOffCost + estimatedCurrentDroneMoves * droneCapability.g
47     currentCostPerDelivery = estimatedCurrentFlightCost / estimatedCurrentNumberOfDeliveries;

```

ii. Fixed code

The screenshot shows an IDE interface with the following details:

- Project View:** Shows a tree structure of files and folders under the package `uk.ac.ed.acp.cw2`. The `RestServiceImplementation.java` file is open in the editor.
- Editor:** Displays the code for `RestServiceImplementation.java`, specifically the `calcDeliveryPath` method. The code implements an A* search algorithm to find a delivery path between a start point and a base, taking into account restricted areas and drone capabilities.
- Terminal:** Shows the output of a test run. It indicates 5 tests passed in 3 seconds, with a total execution time of 827ms. The log also includes Spring Boot version information (`v3.4.3`) and some decorative ASCII art.
- Status Bar:** Shows the file path (`s2487866 > src > main > java > uk > ac > ed > acp > cw2 > service > RestServiceImplementation.java`), line count (287), and character count (111,488).

3. The A star search cut-out moves were set too high
 - a) Problem: If there are no possible paths, it may take a long time for the algorithm to return a JSON object with no deliveries
 - b) Fix: Changed from 100000000 to 1000000