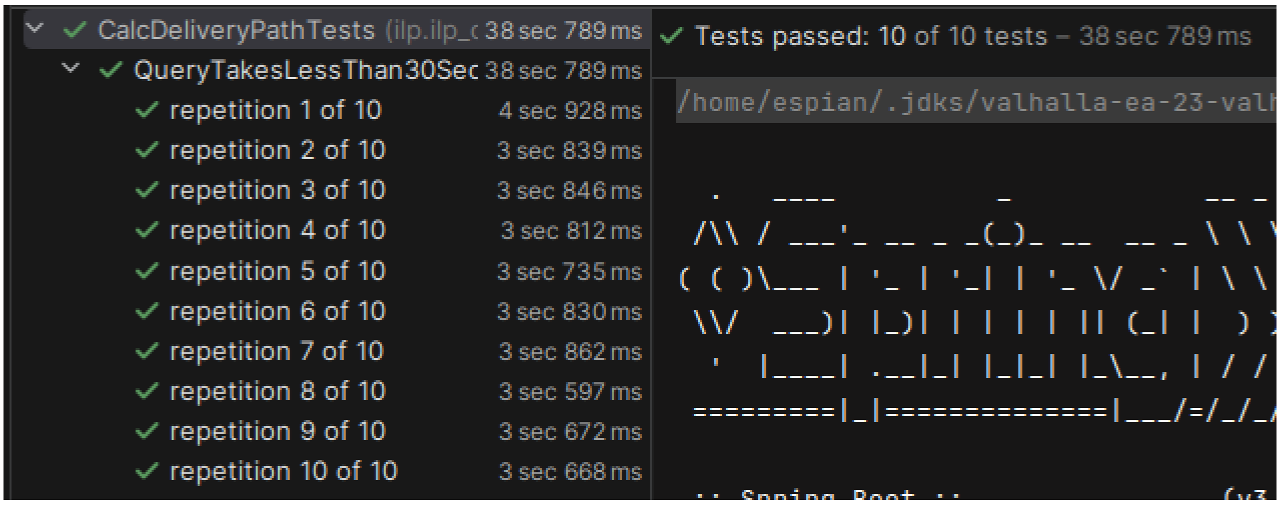
3.3 - Results of testing

Requirement 1 - *No response for an endpoint should take more than 30 seconds*

To test this requirement, a JUnit test function `QueryTakesLessThan30Seconds` was written in file `CalcDeliveryPathTests.java`, and a `@RepeatedTest(10)` annotation was added to repeat the test several times.

As seen here, all tests for this endpoint pass. These are all for 9 delivery locations.

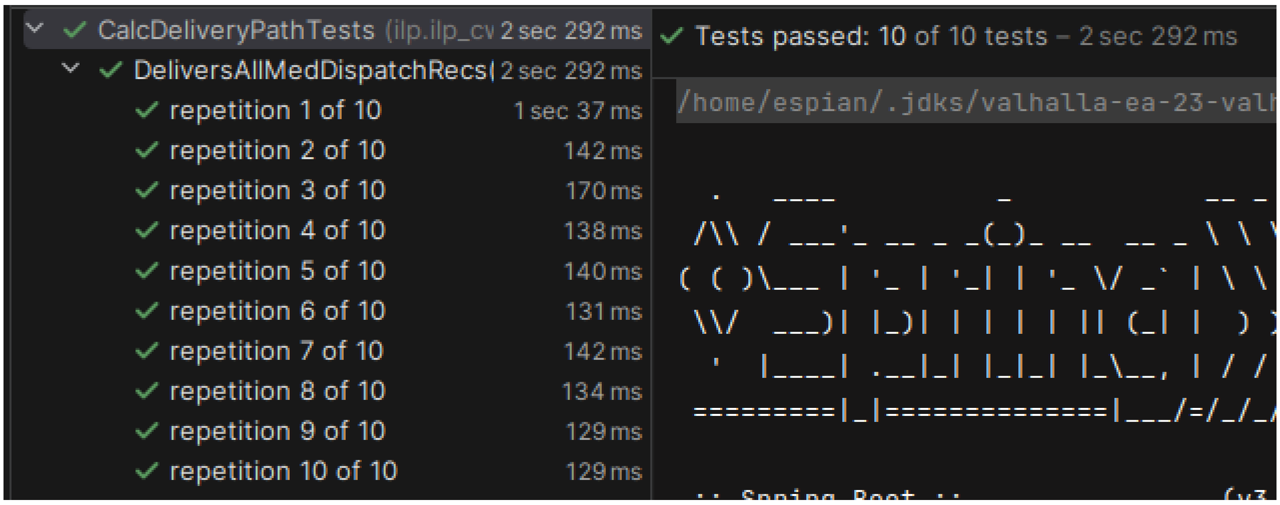


```
✓ CalcDeliveryPathTests (ilp.ilp_c 38 sec 789 ms    ✓ Tests passed: 10 of 10 tests – 38 sec 789 ms
  ✓ QueryTakesLessThan30Sec 38 sec 789 ms
    ✓ repetition 1 of 10       4 sec 928 ms          /home/espian/.jdks/valhalla-ea-23-valh
    ✓ repetition 2 of 10       3 sec 839 ms
    ✓ repetition 3 of 10       3 sec 846 ms            .   ____          _            __ _
    ✓ repetition 4 of 10       3 sec 812 ms           /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \
    ✓ repetition 5 of 10       3 sec 735 ms          ( ( )\___ | '_ | '_| | '_ \/ _` | \ \
    ✓ repetition 6 of 10       3 sec 830 ms           \\/  ___)| |_)| | | | | || (_| |  ) )
    ✓ repetition 7 of 10       3 sec 862 ms            '  |____| .__|_| |_|_| |_\__, | / /
    ✓ repetition 8 of 10       3 sec 597 ms           =========|_|==============|___/=/_/_/
    ✓ repetition 9 of 10       3 sec 672 ms
    ✓ repetition 10 of 10      3 sec 668 ms           :: Spring Boot ::                (v3
```

Requirement 2 - *A drone flight should deliver all medDispatches assigned to it*

To test this requirement, an almost identical approach was used to the above - a JUnit function `DeliversAllMedDispatchRecs` in the same file `CalcDeliveryPathTests.java`, also with the `@RepeatedTest(10)` annotation.

As seen below, all of these tests also pass. These are ran with 5 delivery locations.

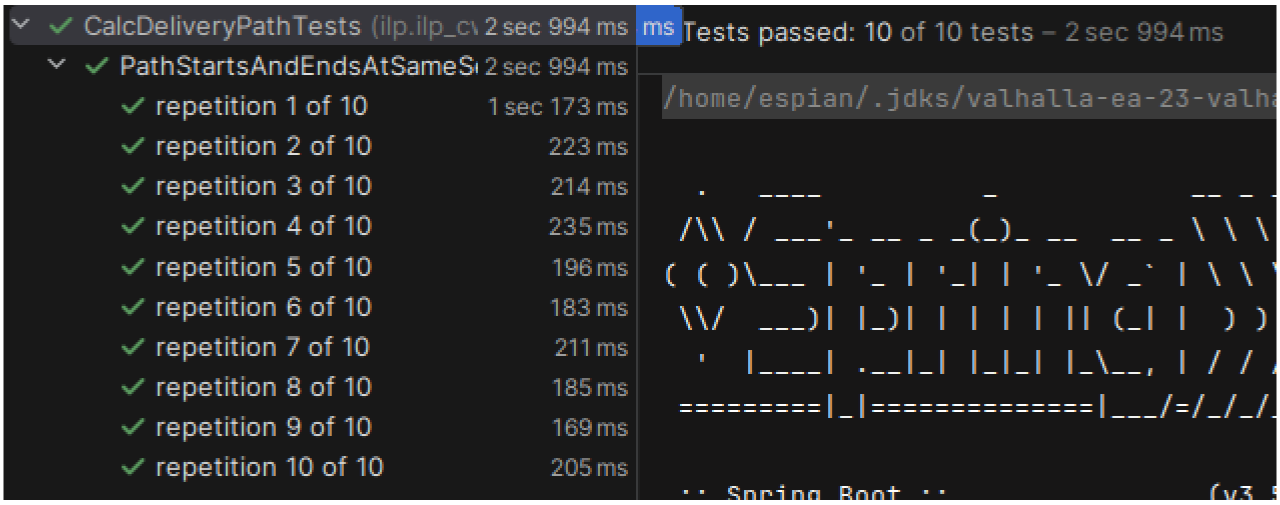


```
✓ CalcDeliveryPathTests (ilp.ilp_cv 2 sec 292 ms    ✓ Tests passed: 10 of 10 tests – 2 sec 292 ms
  ✓ DeliversAllMedDispatchRecs( 2 sec 292 ms
    ✓ repetition 1 of 10       1 sec 37 ms           /home/espian/.jdks/valhalla-ea-23-valh
    ✓ repetition 2 of 10            142 ms
    ✓ repetition 3 of 10            170 ms             .   ____          _            __ _
    ✓ repetition 4 of 10            138 ms            /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \
    ✓ repetition 5 of 10            140 ms           ( ( )\___ | '_ | '_| | '_ \/ _` | \ \
    ✓ repetition 6 of 10            131 ms            \\/  ___)| |_)| | | | | || (_| |  ) )
    ✓ repetition 7 of 10            142 ms             '  |____| .__|_| |_|_| |_\__, | / /
    ✓ repetition 8 of 10            134 ms            =========|_|==============|___/=/_/_/
    ✓ repetition 9 of 10            129 ms
    ✓ repetition 10 of 10           129 ms            :: Spring Boot ::                (v3
```

Requirement 3 - *Ensure flight paths begin and end at the same service point*

To test this requirement, the same approach is used yet again - a function `PathStartsAndEndsAtSameServicePoint`, within the same file `CalcDeliveryPathTests.java` with the same `@RepeatedTest(10)` annotation to offset the randomness.

These tests also pass, as seen below. These tests were also ran with 5 delivery locations.



```
✓ CalcDeliveryPathTests (ilp.ilp_c   2 sec 994 ms   ms Tests passed: 10 of 10 tests – 2 sec 994 ms
  ✓ PathStartsAndEndsAtSameS         2 sec 994 ms
    ✓ repetition 1 of 10             1 sec 173 ms   /home/espian/.jdks/valhalla-ea-23-valha
    ✓ repetition 2 of 10                   223 ms
    ✓ repetition 3 of 10                   214 ms     .   ____          _            __ _
    ✓ repetition 4 of 10                   235 ms    /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \
    ✓ repetition 5 of 10                   196 ms   ( ( )\___ | '_ | '_| | '_ \/ _` | \ \
    ✓ repetition 6 of 10                   183 ms    \\/  ___)| |_)| | | | | || (_| |  ) )
    ✓ repetition 7 of 10                   211 ms     '  |____| .__|_| |_|_| |_\__, | / /
    ✓ repetition 8 of 10                   185 ms    =========|_|==============|___/=/_/_/
    ✓ repetition 9 of 10                   169 ms
    ✓ repetition 10 of 10                  205 ms    :: Spring Boot ::                (v3.
```

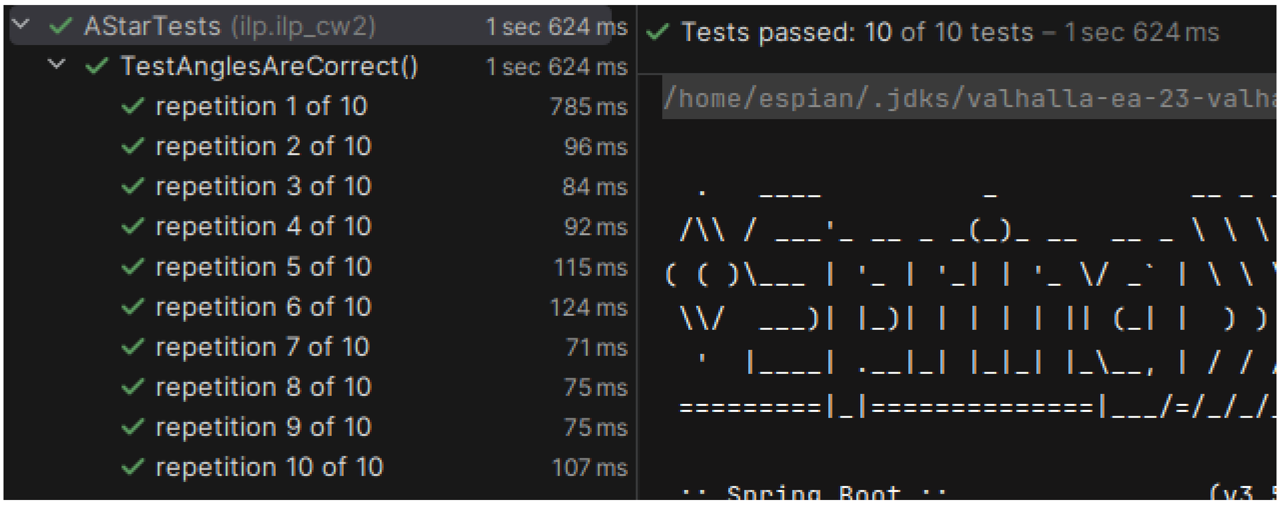
Requirement 4 - *Ensure flight paths are otherwise correct*

As this final requirement was split up into three unit level requirements, three test functions were created inside `AStarTests.java`, as these are unit tests for the A to B pathfinding.

To test this requirement, I split it into three unit-level requirements.:

1. Drones can only move with an angle that is a multiple of 22.5°
2. Drones can only move by *exactly* 0.00015° in a given direction
3. Drones should not be able to fly over no-fly zones, including corner cutting.

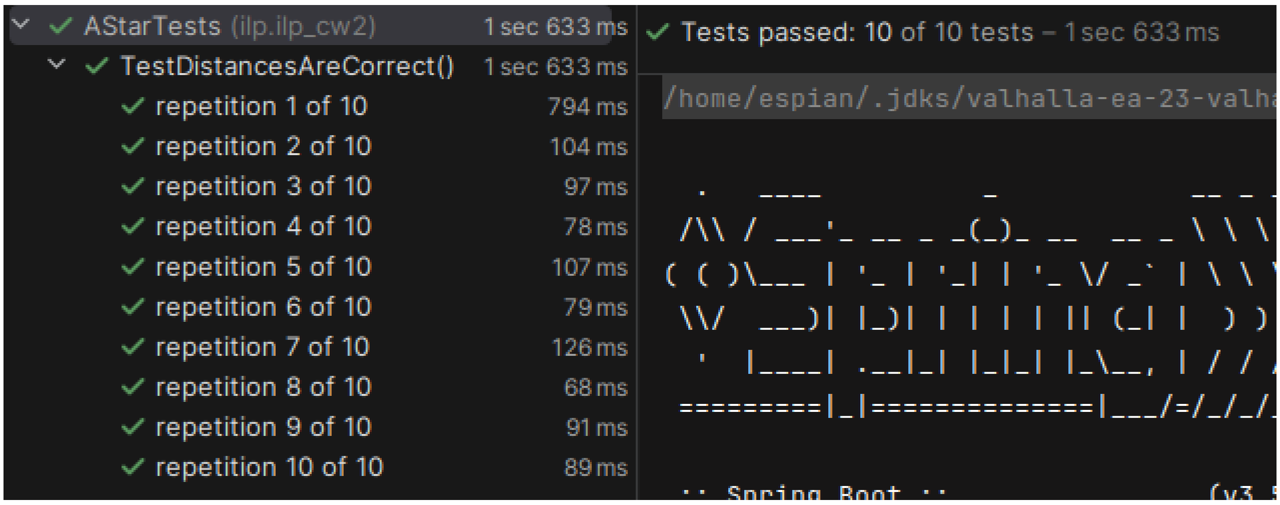
First, ***Drones can only move with an angle that is a multiple of 22.5***$^\circ$.
For this, a JUnit test function `TestAnglesAreCorrect` was created, also with the repeated test annotation.
The results of these tests are below:



```
✓ AStarTests (ilp.ilp_cw2)           1 sec 624 ms   ✓ Tests passed: 10 of 10 tests – 1 sec 624 ms
  ✓ TestAnglesAreCorrect()           1 sec 624 ms
    ✓ repetition 1 of 10                   785 ms   /home/espian/.jdks/valhalla-ea-23-valha
    ✓ repetition 2 of 10                    96 ms
    ✓ repetition 3 of 10                    84 ms     .   ____          _            __ _
    ✓ repetition 4 of 10                    92 ms    /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \
    ✓ repetition 5 of 10                   115 ms   ( ( )\___ | '_ | '_| | '_ \/ _` | \ \
    ✓ repetition 6 of 10                   124 ms    \\/  ___)| |_)| | | | | || (_| |  ) )
    ✓ repetition 7 of 10                    71 ms     '  |____| .__|_| |_|_| |_\__, | / /
    ✓ repetition 8 of 10                    75 ms    =========|_|==============|___/=/_/_/
    ✓ repetition 9 of 10                    75 ms
    ✓ repetition 10 of 10                  107 ms    :: Spring Boot ::                (v3.
```

Second, ***Drones can only move by exactly 0.00015***$^\circ$ ***in a given direction***
For this, a JUnit test function `TestDistancesAreCorrect` was created, also with the

repeated test annotation.
The results of these tests are below:



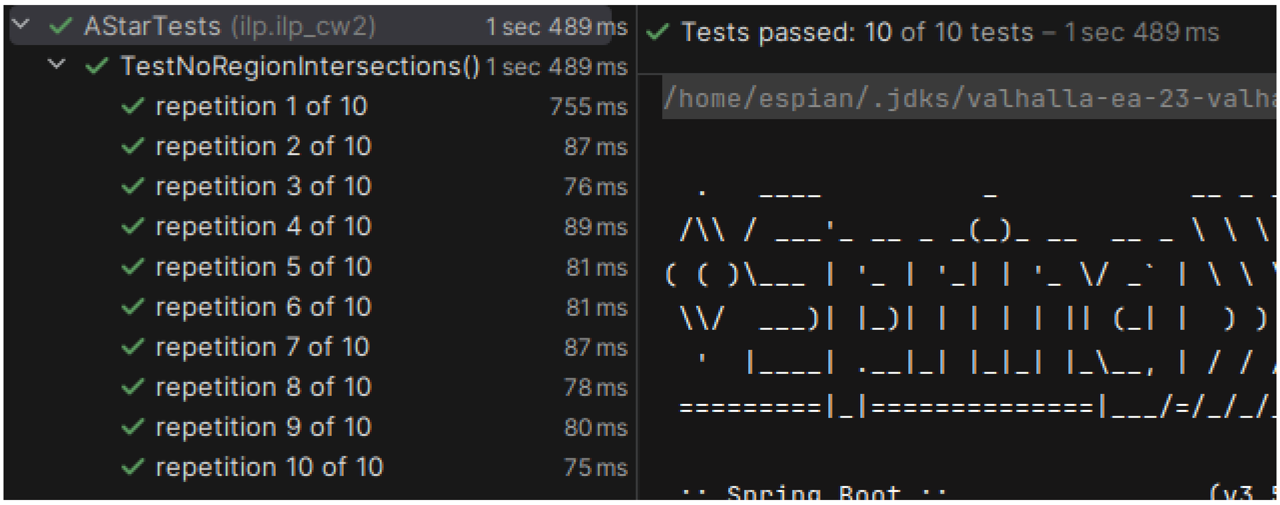
```
✓ AStarTests (ilp.ilp_cw2)              1 sec 633 ms
  ✓ TestDistancesAreCorrect()           1 sec 633 ms
    ✓ repetition 1 of 10                      794 ms
    ✓ repetition 2 of 10                      104 ms
    ✓ repetition 3 of 10                       97 ms
    ✓ repetition 4 of 10                       78 ms
    ✓ repetition 5 of 10                      107 ms
    ✓ repetition 6 of 10                       79 ms
    ✓ repetition 7 of 10                      126 ms
    ✓ repetition 8 of 10                       68 ms
    ✓ repetition 9 of 10                       91 ms
    ✓ repetition 10 of 10                      89 ms
```

```
✓ Tests passed: 10 of 10 tests – 1 sec 633 ms

/home/espian/.jdks/valhalla-ea-23-valha

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) )
  '  |____| .__|_| |_|_| |_\__, | / /
 =========|_|==============|___/=/_/_/
 :: Spring Boot ::                (v3.
```

Lastly, **Drones should not be able to fly over no-fly zones, including corner cutting.***
For this, a JUnit test function `TestNoRegionIntersections` was created, also with the repeated test annotation.
The results of these tests are below:



```
✓ AStarTests (ilp.ilp_cw2)              1 sec 489 ms
  ✓ TestNoRegionIntersections()         1 sec 489 ms
    ✓ repetition 1 of 10                      755 ms
    ✓ repetition 2 of 10                       87 ms
    ✓ repetition 3 of 10                       76 ms
    ✓ repetition 4 of 10                       89 ms
    ✓ repetition 5 of 10                       81 ms
    ✓ repetition 6 of 10                       81 ms
    ✓ repetition 7 of 10                       87 ms
    ✓ repetition 8 of 10                       78 ms
    ✓ repetition 9 of 10                       80 ms
    ✓ repetition 10 of 10                      75 ms
```

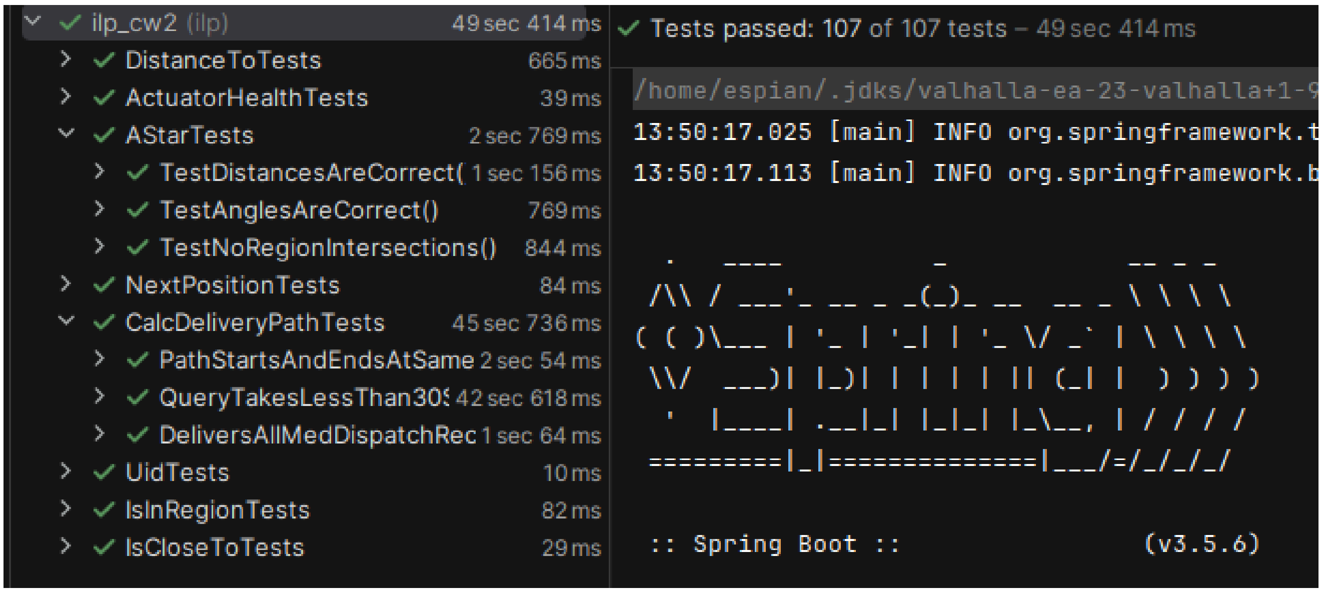
```
✓ Tests passed: 10 of 10 tests – 1 sec 489 ms

/home/espian/.jdks/valhalla-ea-23-valha

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) )
  '  |____| .__|_| |_|_| |_\__, | / /
 =========|_|==============|___/=/_/_/
 :: Spring Boot ::                (v3.
```

Overall

Here is proof that all tests pass in one run, including the results of tests written as part of ILP coursework 1:



```
✓ ilp_cw2 (ilp)                          49 sec 414 ms
  > ✓ DistanceToTests                          665 ms
  > ✓ ActuatorHealthTests                       39 ms
  ✓ AStarTests                             2 sec 769 ms
    > ✓ TestDistancesAreCorrect(  1 sec 156 ms
    > ✓ TestAnglesAreCorrect()                 769 ms
    > ✓ TestNoRegionIntersections()            844 ms
  > ✓ NextPositionTests                         84 ms
  ✓ CalcDeliveryPathTests                 45 sec 736 ms
    > ✓ PathStartsAndEndsAtSame 2 sec 54 ms
    > ✓ QueryTakesLessThan30S 42 sec 618 ms
    > ✓ DeliversAllMedDispatchRec 1 sec 64 ms
  > ✓ UidTests                                  10 ms
  > ✓ IsInRegionTests                           82 ms
  > ✓ IsCloseToTests                            29 ms
```

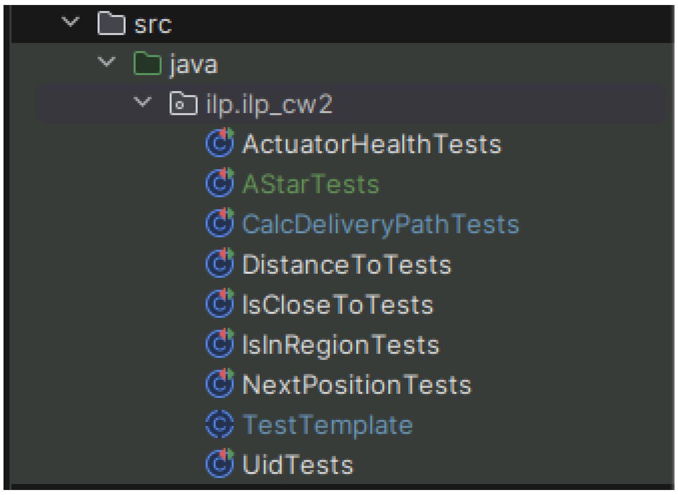
```
✓ Tests passed: 107 of 107 tests – 49 sec 414 ms

/home/espian/.jdks/valhalla-ea-23-valhalla+1-9
13:50:17.025 [main] INFO org.springframework.t
13:50:17.113 [main] INFO org.springframework.b

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.6)
```

Here is the file structure of the tests:



```
src
  java
    ilp.ilp_cw2
      ActuatorHealthTests
      AStarTests
      CalcDeliveryPathTests
      DistanceToTests
      IsCloseToTests
      IsInRegionTests
      NextPositionTests
      TestTemplate
      UidTests
```

Please note that the only test files explicitly relevant to this coursework are `CalcDeliveryPathTests` *and* `AStarTests`*, as all other files were created as part of the ILP coursework*

As all tests are written as appropriately named JUnit tests and in an understandable file structure, I consider the tests both comprehensive to read the results of, but also to add more tests - the nature of the structure makes adding tests easy, and automatically makes them force-fails for the pipeline as long as they contain the work "test" or "tests" (*Please see LO5*).