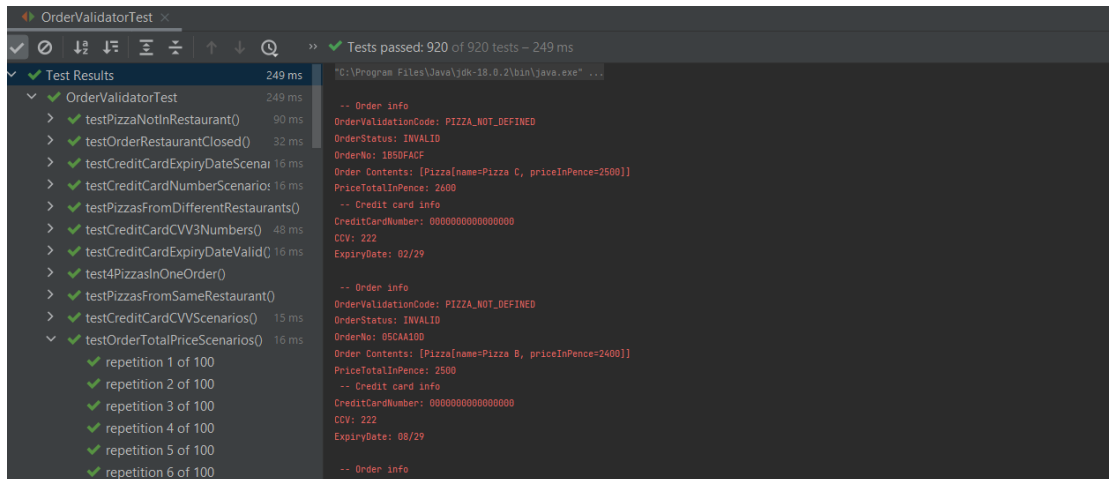
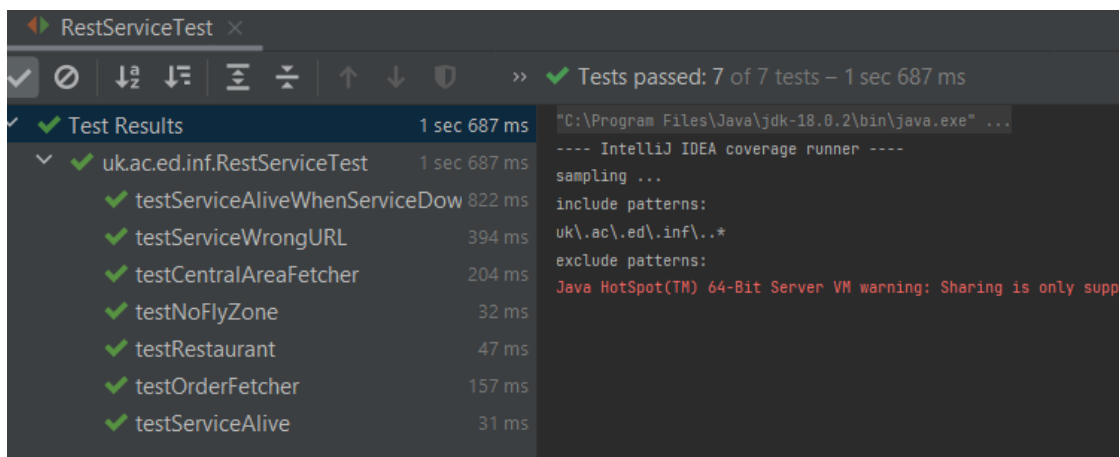


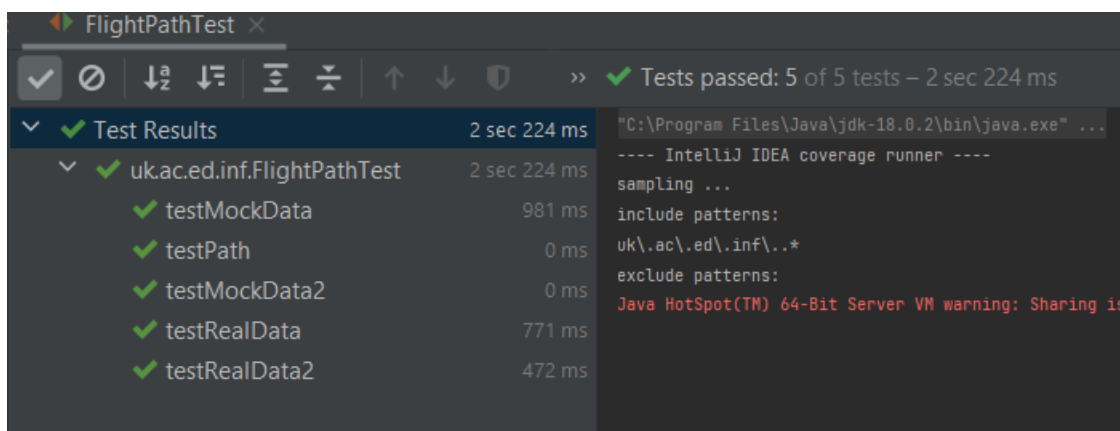
RESULTS



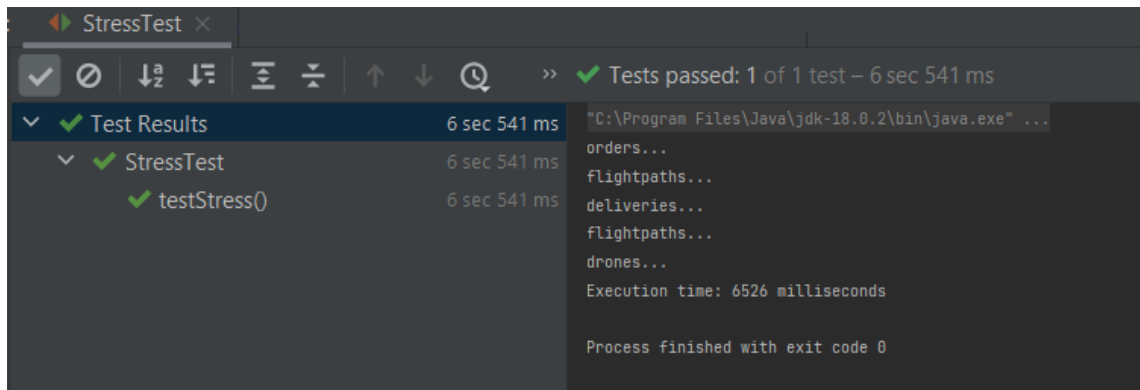
1) OrderValidatorTest passed 100%



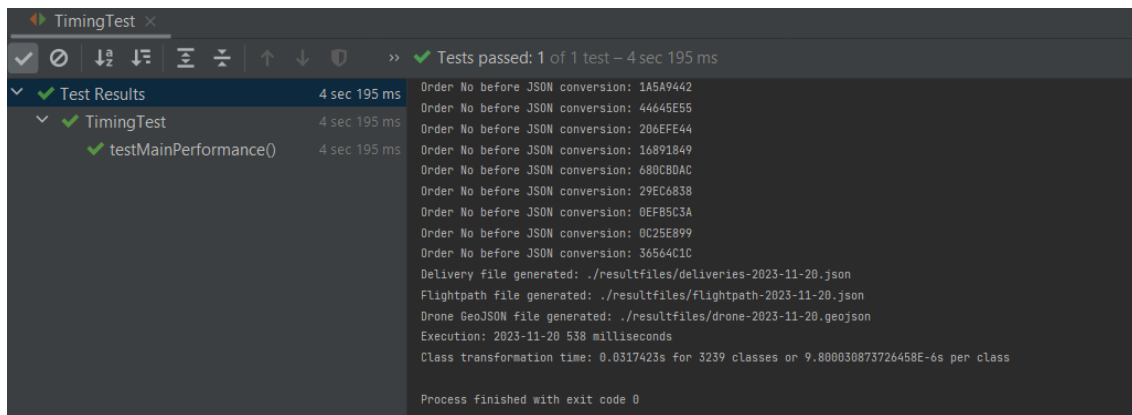
2) RestServiceTest passed 100%



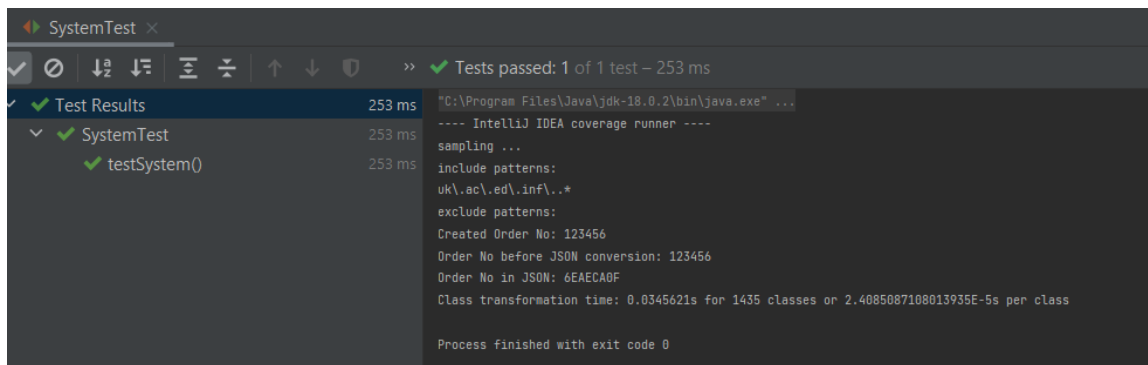
3) FlightPathTest passed 100%



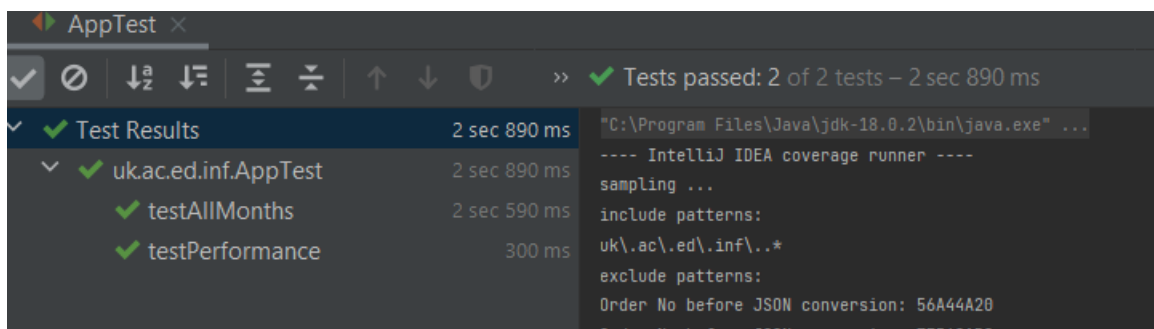
4)StressTest completed in 6526 milliseconds, passed 100%



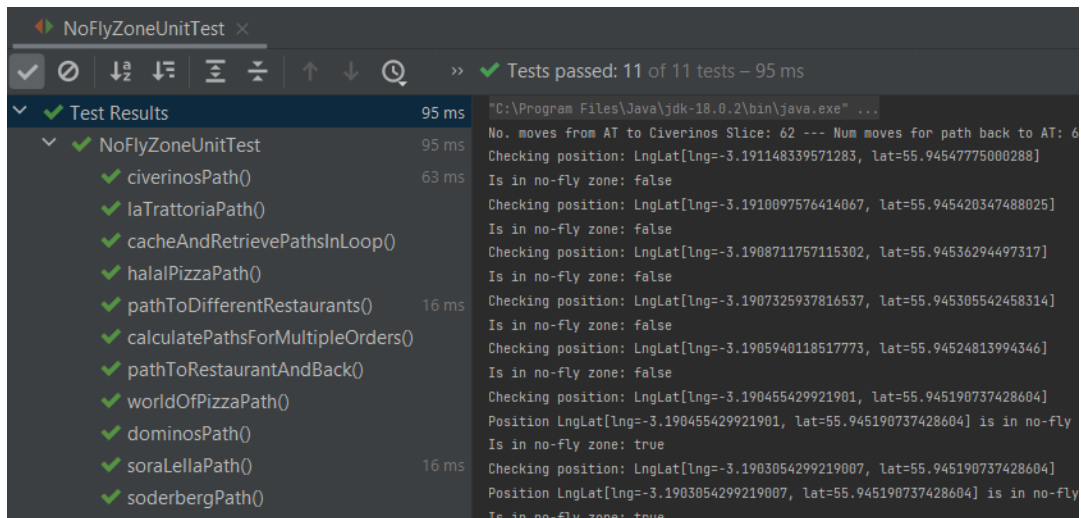
5)Timing Test completed in 538 milliseconds, passed 100%



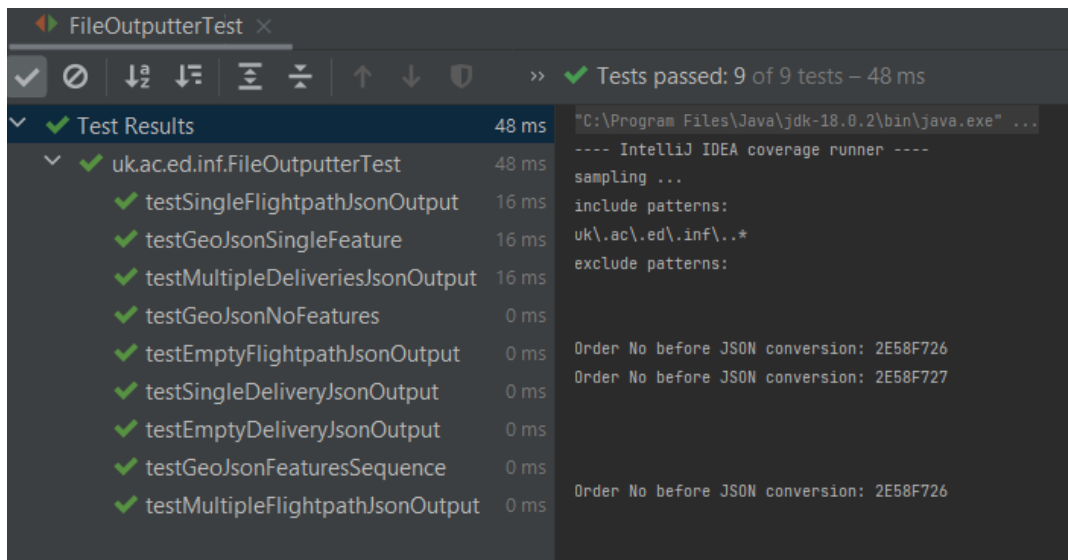
6)SystemTest, passed 100%



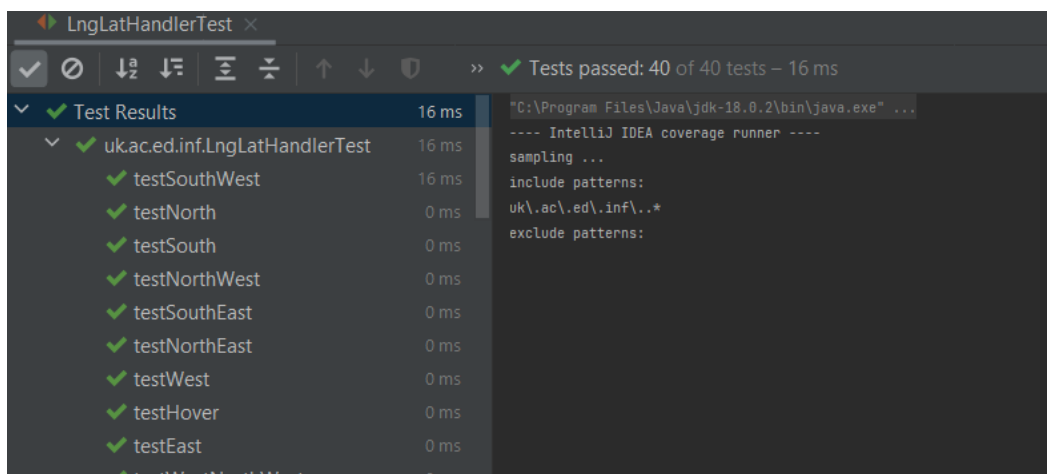
7)AppTest, passed 100%



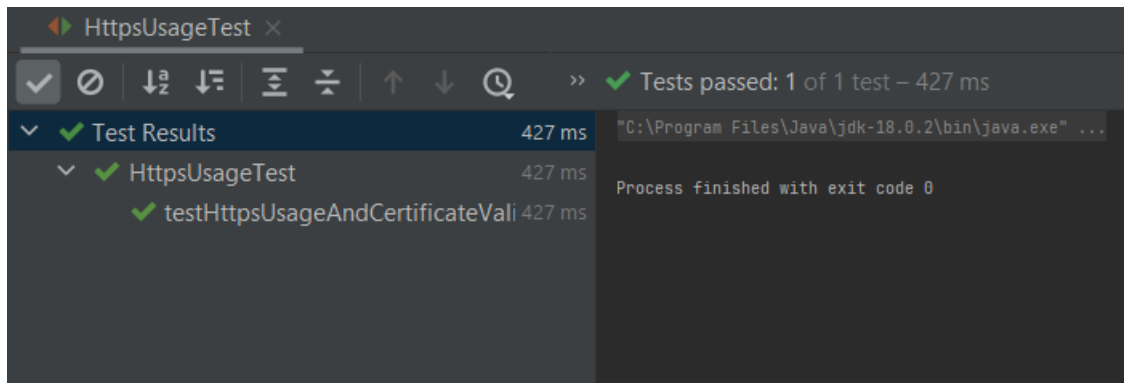
8) NoFlyZoneUnitTest, passed 100%



9) FileOutputterTest, passed 100%



10) LngLatHandlerTest, passed 100%



11) HttpsUsageTest, passed 100%

Coverage: SystemTest x

95% classes, 96% lines covered in package 'uk.ac.ed.inf'

Element	Class, %	Method, %	Line, %
ilp			
App	50% (1/2)	85% (6/7)	83% (59/71)
AppTest	100% (1/1)	100% (5/5)	100% (25/25)
FileOutputter	100% (1/1)	100% (3/3)	100% (43/43)
FileOutputterT...	100% (1/1)	100% (11/11)	100% (84/84)
FlightPath	100% (1/1)	100% (8/8)	100% (46/46)
FlightPathTest	100% (1/1)	100% (18/18)	100% (124/124)
HttpsUsageTe...	100% (1/1)	100% (1/1)	85% (12/14)
LngLatHandler	100% (2/2)	100% (10/10)	94% (52/55)
LngLatHandle...	100% (1/1)	100% (46/46)	98% (148/150)
Node	100% (1/1)	55% (5/9)	69% (9/13)
NoFlyZoneUni...	100% (1/1)	90% (20/22)	90% (190/210)
OrderValidator	100% (1/1)	100% (1/1)	100% (71/71)
OrderValidato...	100% (1/1)	100% (33/33)	99% (215/216)
RestService	100% (1/1)	100% (7/7)	100% (26/26)
RestServiceTest	100% (1/1)	100% (24/24)	97% (71/73)
StressTest	100% (1/1)	100% (8/8)	100% (34/34)
SystemTest	100% (1/1)	100% (5/5)	93% (45/48)
TestConstraints	100% (1/1)	100% (3/3)	100% (3/3)
TimingTest	100% (1/1)	100% (4/4)	91% (22/24)

12) Almost achieved perfect coverage.

EVALUATION

Passing all the tests with great code coverage is a really good indicator of a great testing where we achieved 100% pass on every test class. However, a 100% pass rate can create a false sense of security, leading to overconfidence in the system's reliability. Also a perfect test score can sometimes lead to resistance to change, as any modification in the software might disrupt the existing test results.

Though my code coverage isn't flawless, there are still areas of the code that I may have missed that could cause issues.

All these coverage metrics do have one significant flaw, though, and that is that they are simply a surrogate for the breadth and depth of the test suite. This is because it can be quite difficult to determine how suitable a test suite is for a particular system; therefore, even though these proxies are not ideal, they can provide us with quantifiable estimates and a high level of assurance that the system functions as intended.

After analyzing the data in accordance with the specified criteria, we may draw the conclusion that testing confidence has increased based on the results and their assessment. This ultimately leads to the conclusion that the testing procedure was successful and produced outcomes that satisfied the standards.