

Learning Outcome 3: Testing Techniques and Coverage

3.1 Range of Techniques:

Throughout the process, we created 3 main testing files, **ControllerTest**, **DroneServiceTest** and **OrderServiceTest**.

- **ControllerTest**: Tests for correct errors returned correct results from specific functions, appropriate Boolean responses and resistance to malformed input.
- **DroneServiceTest**: Tests for resistance against malformed input, checks accuracy of coordinate results from specific functions, correct Boolean responses, and performance test of the system.
- **OrderServiceTest**: Tests every possible combination of legal incorrect order types.

3.2 Adequacy of Testing:

ControllerTest	DroneServiceTest	OrderServiceTest
F5, NF2,	F2, F6, NF1	F1

As we can see from the above, the **test plan** (that was designed in *LearningOutcome2.pdf*) comprehensively covers all of the functional and non-functional requirements that were described in *LearningOutcome1.pdf*.

Therefore, every single requirement has been exactly and concisely fulfilled by the tests.

Additionally, the code achieved over 95% coverage. We thus know our plan was implemented adequately.

3.3 Results of Testing:

Every single test passed. Below, you can see all 636 tests passing (600 of which came from the synthetic orders).

Additionally, I also calculated a 95% code coverage using IntelliJ's code coverage calculator. The example is below:

PizzaDronz in PizzaDronz: 636 total, 636 passed

2.21 s

[Collapse](#) | [Expand](#)

InRegionRequestTest

testInRegionRequest_ValidRequest() passed 21 ms

DroneServiceTest

testIsInRegion_PointOutsideRegion() passed 390 ms

testDistanceTo_SameCoordinates_ReturnsZero() passed 3 ms

testIsCloseTo_PointsAreFar() passed 2 ms

testDistanceTo_UniqueCoordinates() passed 2 ms

testNextPosition_MoveNorthEastFromAppleton() passed 3 ms

testNextPosition_MoveNorthFromAppleton() passed 2 ms

testIsCloseTo_PointsAreClose() passed 6 ms

testDeliveryPathCalculation_PerformanceUnder20Seconds() passed 857 ms

testIsInRegion_PointInsideRegion() passed 2 ms

1 ms

LngLatPairRequestTest

testLngLatPairRequest_InvalidRequest() passed

testLngLatPairRequest_ValidRequest() passed 1 ms

OrderUnitTests

testExpiryDate_ValidExpiryDate() passed 3 ms

testCreditCardNumber_ValidNumber() passed 1 ms

testCreditCardNumber_InvalidNumber_TooLong() passed

testCVV_ValidCVV() passed

testCreditCardNumber_InvalidNumber_TooShort() passed 1 ms

testCVV_InvalidCVV_NotNumeric() passed

testOrder_OrderStatus_Valid() passed 1 ms

testExpiryDate_InvalidExpiryDate_InThePast() passed

testCVV_InvalidCVV_TooShort() passed

testCreditCardNumber_InvalidNumber_NotNumeric() passed










1 ms

NextPositionRequestTest

testNextPositionRequest_ValidRequest() passed 1 ms

PizzaDronzApplicationTests

contextLoads() passed 589 ms

        		
✓ PizzaDronz (uk.ac.ed.inf)	2 sec 211 ms	
> ✓ InRegionRequestTest	21 ms	
✓ DroneServiceTest	1 sec 267 ms	
✓ testIsInRegion_PointOutsideRegion()	390 ms	
✓ testDistanceTo_SameCoordinates_ReturnsZero()	3 ms	
✓ testIsCloseTo_PointsAreFar()	2 ms	
✓ testDistanceTo_UniqueCoordinates()	2 ms	
✓ testNextPosition_MoveNorthEastFromAppleton()	3 ms	
✓ testNextPosition_MoveNorthFromAppleton()	2 ms	
✓ testIsCloseTo_PointsAreClose()	6 ms	
✓ testDeliveryPathCalculation_PerformanceUnder20Seconds()	857 ms	
✓ testIsInRegion_PointInsideRegion()	2 ms	
> ✓ LngLatPairRequestTest	1 ms	
> ✓ OrderUnitTests	6 ms	
> ✓ NextPositionRequestTest	1 ms	
✓ PizzaDronzApplicationTests	589 ms	
✓ contextLoads()	589 ms	
✓ ControllerTest	170 ms	
✓ testNextPosition_InvalidAngle_Negative()	77 ms	
✓ testIsCloseTo_ValidRequest_False()	53 ms	
✓ testIsInRegion_InvalidRegion_AllSameLongitude()	7 ms	
✓ testIsCloseTo_InvalidRequest_MissingPosition()	3 ms	
✓ testIsInRegion_InvalidRegion_InsufficientVertices()	3 ms	
✓ testNextPosition_NullStartPosition()	3 ms	
✓ testNextPosition_InvalidAngle_OutOfBounds()	5 ms	
✓ testIsCloseTo_InvalidJsonStructure()	3 ms	
✓ testDistanceTo_NullInput()	5 ms	

3.4 Evaluation of Testing:

All 636 tests passed. Our order validation, drone path generation and functional and non-functional requirements have all been met. Additionally, the code coverage is over 95%, an exceptionally high mark that is well-over industry's benchmark (80%+). This high coverage was achieved through our comprehensive data-driven testing approach using 600 predefined test cases, as well as consistent Test-Driven Development throughout the entire creation process.

However, despite using a large synthetic dataset, we cannot be entirely certain about test diversity. While our JSON test data covers many scenarios, it's still manually curated, which could lead to potential blind spots in edge cases, particularly in drone path generation and complex order validation scenarios.

Several critical issues were identified and fixed through our testing process:

1. **Issue:** Order validation was failing to properly validate credit card numbers that were exactly at the maximum length limit.

Fix: Modified credit card validation logic to properly handle boundary conditions in length checking.

2. **Issue:** Total cost validation was failing legitimate orders because delivery charge wasn't included in calculations.

Fix: Updated price validation to include the standard £1 delivery charge in total cost calculations.

3. **Issue:** Order validation was accepting invalid pizza combinations (e.g., pizza was from multiple different restaurants).

Fix: Enhanced pizza validation to identify when checking against different restaurant's menus.

These fixes were primarily identified through our data-driven testing approach, demonstrating the effectiveness of having a large, diverse test dataset. However, the fact that these issues were found also highlights potential gaps in our initial development process that could be addressed through enhanced test-driven development practices.

The high coverage numbers, while impressive, should be viewed cautiously as they don't necessarily indicate complete testing of all possible business scenarios, particularly in complex areas like drone path generation and concurrent order processing.