LO2

1 Test Plan

Functional Requirements (F1-F5):
F1 - Flight Path Validity (High Priority): Validating paths was essential to ensure drones avoided no-fly zones and remained in the Central Area after entry. Unit tests simulated paths near and within no-fly zones.
F2 - Order Validation (High Priority): Ensured only valid orders were processed. Each validation criterion, such as card details and pizza limits, was independently tested to confirm compliance with specifications.
F3 - Endpoint Reachability (Medium Priority): Verified availability and correct responses for all endpoints, using system tests through Postman and automated scripts.
F4 - REST Server Retrieval (High Priority): Integration tests ensured that no-fly zones, restaurant data, and Central Area coordinates were correctly fetched and handled.
F5 - GeoJSON Format Validation (Medium Priority): Checked that flight paths adhered to GeoJSON specifications. **Non-Functional Requirements (N1-N3):**
N1 - Performance (High Priority): Performance tests measured the runtime of pathfinding, ensuring compliance with the 60-second runtime constraint.
N2 - Maintainability (Medium Priority): Conducted code reviews and assessed modularity by inspecting code.
N3 - Robustness (Medium Priority): Simulated invalid inputs, edge cases, and system failures to test error-handling mechanisms and application resilience.

2 Evaluation of the Quality of the Test Plan

The test plan's quality was assessed using coverage
Coverage: Tests comprehensively addressed F1-F5 and N1-N3, ensuring all requirements were validated. Edge cases like drone entry/exit anomalies for F1 and invalid REST server data for F4 were explicitly included to capture unexpected behaviors.
Redundancy Minimization: Iterative reviews identified and eliminated duplicate or low-value tests. For example, tests for invalid order validation (F2) were consolidated.

3 Instrumentation of the Code

Logging: Flight path calculations (F1) logged node visits and validated adherence to no-fly zone and Central Area rules. For F4, logs tracked REST data retrieval attempts, documenting failures for analysis.
Assertions: Assertions verified runtime constraints, such as performance thresholds (N1) and input validity (N3).
Mocking: Mock endpoints simulated REST server responses, providing a controlled environment to test order validation.
Validation Tools: Automated GeoJSON schema validation ensured output compliance for F5. Tools like Postman further verified endpoint responses for F3.

4 Evaluation of the Instrumentation

Strengths: Logs and assertions identified issues with high precision, ensuring critical requirements like flight path validity and REST data integrity were robustly tested. Mocking minimized external dependencies, improving test reliability.
Weaknesses: Mock data lacked real-world variability, which limited input diversity.