Test Planning Document

James Howard

1 Requirements being tested

1.1 Retrieve delivery orders from a remote API with pickup and drop-off locations.

1.2 Compute a flight path for all orders within a given day.

1.3 Avoid no-fly zones by integrating with geospatial libraries and calculating and routing paths to avoid these zones.

1.4 Orders must be validated. Invalid orders will be received, so invalid orders must be filtered out.

1.5 Path finding computations should take no longer than 60 seconds for all orders of the day.

1.6 The system must be able to handle at least 50 orders within one day.

2 Priority and Pre-requisites

- 1.1 Retrieve delivery orders from a remote API: Priority - High

  - Essential for order processing, retrieval is a foundational task.

  - Requires access to the remote API or a dummy API

  - Requires test data with valid and invalid delivery orders to verify the system's behavior.

- 1.2 Compute a flight path for all orders within a given day: Priority - High

  - Routing is a fundamental part of the system so testing whether it is functioning properly will be key.

  - Requires test data including multiple orders with varying pick-up locations.

  - Requires simulated edge cases, for example with all locations clustered together or spread far apart.

- 1.3 Avoid no-fly zones: Priority - High

  - Compliance with no-fly zones is necessary for legal and safety requirements, making this a top priority.

  - Requires access to geospatial libraries or API

  - Also requires test data for example no-fly zones.

  - Needs test orders that intersect no-fly zones to validate the rerouting logic.

- 1.4 Orders must be validated: Priority - Medium

  - Proper order validation ensures the reliability of the system by preventing processing invalid data, which is expected.

  - Requires a clearly defined format for a valid order.

  - Requires test data with both valid and invalid orders (e.g., with missing fields, or incorrect data types).

  - Will require the implementation of validation logic and error-handling mechanisms.

- 1.5 Pathfinding computations within 60 seconds: Priority - Medium

  - Performance constraints are key for user experience, however not completely necessary for the system too operate.

  - Will require a simulated workload with various order compositions and volumes.

  - To validate this will require a way to measure the performance of the system.

- 1.6 Able to handle at least 50 orders within a day: Priority - Medium

  - System scalability is essential to meet business requirements and handle expected workloads. If the system cannot cope with expected workloads

  - To fully test this will need the whole system to function.

  - For early tests can have scaffolding to simulate how different parts of the system work so that early parts of the code can be tested with this requirement.

  - Can feedback how early tests perform with this requirement.

3 Scaffolding and Instrumentation

- 1.1 Retrieve delivery orders from a remote API:

  - Will need to set up some tool to simulate API responses during testing, and also to provide test order data.

- 1.2 Compute a flight path for all orders within a given day

  - Will need to build tools or scripts to generate synthetic orders with varying pickup locations.

  - May want tools to visualize paths for manual verification.

- 1.3 Avoid no-fly zone

  - Will need to use a mapping tool like geojson.io to visually validate routes avoiding no-fly zones.

- 1.4 Orders must be validated:

  - Will need to implement logging for invalid orders, including reasons for rejection.

  - Need instrumentation to measure how much of the code is executed during tests.

- 1.5 Pathfinding computations within 60 second:

  -Need instrumentation that can measure execution time.

- 1.6 Able to handle at least 50 orders within a day:

  - For early tests will require scaffolding to simulate parts of the system

4 Process

- 1.1 Retrieve Delivery Orders: Start - Early

  - Foundational to the system, required for all other tasks.

- 1.2 Compute a Flight Path: Start -Early to Middle

  -Critical functionality that must be tested as soon as order retrieval works.

- 1.3 Avoid No-Fly Zones: Start - Middle - Depends on pathfinding; compliance testing should start after basic routing is functional.

- 1.4 Orders Must Be Validated: Stat - Early Protects the system from invalid data. This is necessary for testing and debugging for later processes.

- 1.5 Pathfinding Within 60 Seconds: Start - Middle to Late

  -Testing performance should occur after functional correctness of pathfinding is verified, otherwise code being tested could be changed dramatically.

- 1.6 Handle at Least 50 Orders: Start - Middle

  - Whilst full scalability testing will require a fully functional system, earlier testing can be performed to make sure individual components can also perform the task.