Software Testing Project Requirements

James Howard

1 Functional Requirements

- Retrieve delivery orders from a remote API with pickup and drop-off locations.
- Compute an optimised path for all orders within a given day.
- Avoid no-fly zones by integrating with geospatial libraries and calculating and routing paths to avoid these zones.
- Interact with the external API to fetch no fly zones and restaurants.
- Orders must be validated. Invalid orders will be received, so invalid orders must be filtered out.
- Once the drone enters the Central Area cannot leave until delivery is made to the roof of Appleton Tower.

2 Non-Functional Requirements

- Path finding computations should take no longer than 60 seconds for all orders of the day.
- The system must be able to handle unexpected input data.
- Must adhere to aviation regulations regarding drones by avoiding no-fly zones.
- The system must be able to handle at least 50 orders within one day.
- Flight paths must be able to be visually displayed in a way that is clear and easy to understand for both operators and customers.

3 Test Approach for requirements:

- For the requirement retrieve delivery orders from a remote API with pickup and drop-off locations shall carry out functional testing with partitioning to test both valid and invalid inputs.
- For the requirement to compute an optimised path for all orders within a given day will carry out functional Testing to verify the path computation for different scenarios such as no orders,a small number of orders (e.g., 1-5) and a large amount of orders (10+ orders).
- For the requirement to avoid no-fly zones by integrating with geospatial libraries and calculating and routing paths to avoid these zones; will carry out functional testing where I validate that the system correctly responds to the following test cases: orders directly within a no-fly zone, orders near the boundaries of a no-fly zone and complex no fly zones (e.g. non-convex regions).
- For the requirement to interact with the external API to fetch no fly zones and restaurants; will use structural testing with 100% Statement Coverage to ensure all lines of the API integration logic are tested.
- For the requirement that orders must be validated will use structural testing with a high Branch Coverage to verify validation logic for different inputs.

- For the requirement Pathfinding computations should take no longer than 60 seconds for all orders of the day will carry out Combinatorial Testing with pairwise testing using the parameters number of orders and geographical spread of order restaurants, along with instrumentation to measure the time taken for the computation.
- For the requirement that the system must be able to handle unexpected input data will use robustness testing testing the system's ability to handle unexpected inputs, such as: invalid latitude/longitude, incorrect data types and missing data or incomplete information.
- For the requirement that the system must adhere to aviation regulations regarding drones by avoiding no-fly zones will use structural testing with a 100% statement coverage on code that deals with checking the path for no fly zones.
- For the requirement that system must be able to handle at least 50 orders within one day will carry out functional testing with test cases for below 50 orders, 50 orders and above 50 orders.
- For the requirement that flight paths must be able to be visually displayed in a way that is clear and easy to understand for both operators and customers; will verify that the flight path is output in a geoJson file that is able to be visually displayed on a valid website.