Requirements

Suggested test approaches appear below certain requirements.

For customers:

Functional

1.1 Have each order delivered within 0.00015 degrees of each delivery location. **UNIT**
   - Write tests to compare the calculated flightpath with the official longitude and latitude values associated with the desired What3Words addresses provided. Use Pythagoras to calculate the distance between the desired location and the location determined by the flightpath and ensure all values are less than or equal to 0.00015.

1.2 Ensure the price charged for each order is accurate to the items that were ordered. **UNIT**
   - Generate test data to store on the web server, and run multiple iterations of the system, comparing the prices in the test data with those prices determined by the system.

1.3 What3Words delivery addresses should be accurately converted to a pair of co-ordinates that are within 0.00015 degrees of the desired location. **UNIT**
   - Write tests to compare the calculated longitude and latitude values in the system with the official values stored on the What3Words website and use Pythagoras to calculate the distance between the two.

Qualitative

1.4 System should execute within 5 minutes so the drone can begin its flight as the lunch break begins (order window closes 5 minutes before lunch break). **SYSTEM**
   - Run 100 iterations of the system on various generated test data and flag any executions that run for more than 5 minutes.

For Shop Owners:

Functional

2.1 Drone should arrive within 0.00015 degrees of the shops to allow pickup. **UNIT**
   - As for 1.1

Owners of the buildings in no-fly-zones:

Functional

3.1 At no point should the final path enter the no-fly-zones. **UNIT**
   - Use the generated GeoJSON flightpath to check any intersection with the GeoJSON no-fly-zones on several iterations of the flightpath unit.

3.2 At no point should the final path exit the bounds for the system. **UNIT**
   - As above.

Qualitative

3.3 The system should be able to still calculate a path given any potential permutations of no-fly-zones, as long as they do not block off valid order locations entirely. **UNIT**

Entity who commissioned the project:

Functional

4.1 Drone must always return within 0.00015 degrees of Appleton Tower (the launch point) as its final move. **UNIT**

- As for 1.1

4.2 The flightpaths generated must not exceed 1500 moves. **UNIT**

- Run several permutations of the flightpaths on test data and count moves.

Qualitative

4.3 Path should be optimised with the goal of maximising the profit margin. A percentage monetary value of 100% is desirable, 90% and above is acceptable. **SYSTEM**

- Calculate percentage monetary value for each execution of the system, and log any results where the value falls below 90%.

4.4 Behind the above priority, the path should aim to deliver as many orders as possible. 100% is desirable, 80% is acceptable. **SYSTEM**

- Calculate delivery success rate, and log any results where the value falls below 80%.

4.5 Drone should be able to deliver at least 50 orders within the 1500 moves permitted within the constraints, in 90% of valid test cases. **SYSTEM**

System requirements:

5.1 System should connect to database and web server when given the correct ports as arguments and both the database and web server are running correctly. **INTEGRATION**

- Write test to automatically set up database and web server on various different free ports, and test that when the correct port numbers are given, that connection is always successful.

5.2 System should be able to retrieve the data comprehensively and accurately from both the database and web server. **INTEGRATION**

5.3 If data cannot be retrieved from the database or web server, or the data required is missing or unavailable, the system should not attempt to plan a flight path or update the database. **UNIT**

- Write tests to incorrectly set up database and/or web server and monitor system actions.