Range of Requirements

In this document I will analyse the requirements imposed in the Informatics Large Practical coursework specification and determine appropriate testing strategies for each. Only requirements that are directly testable or assessable within the **ILP CW2 Specification** are included. Requirements relating to user interfaces, storage, authentication, and real-world deployment are explicitly out of scope although I will briefly mention the non-functional requirements that would be needed should deployment be the true end goal.

The main types of requirements that will be covered are functional and qualitative requirements as well as measurable quality attributes. Each requirement will be given a reference ID that links them to the individual test cases in the **Test Cases** document.

Notation:
The IDs are of the form "type of requirement – subcategory of requirement – number".

FR - Functional Requirement
MA – Measurable Attribute
NFR – Non- Functional Requirement

OV – Order Validation
N – Navigational

Functional Requirements

ID	Requirement	Test Implication
FR-OV-01	Reject orders with invalid credit card expiry date	Format & boundary testing
FR-OV-02	Reject orders with invalid credit card number	Format validation
FR-OV-03	Reject orders with invalid CVV	Format validation
FR-OV-04	Reject orders containing zero pizzas	Boundary testing
FR-OV-05	Reject orders exceeding the maximum pizza count	Boundary testing
FR-OV-06	Reject orders containing pizzas that do not exist in any restaurant	Lookup validation
FR-OV-07	Reject orders containing pizzas from multiple restaurants	Consistency validation
FR-OV-08	Reject orders with incorrect total price	Arithmetic validation
FR-OV-09	Reject orders placed when the restaurant is closed	Date-based validation
FR-OV-10	Each invalid order will contain exactly one error condition	Isolation testing
FR-OV-11	For syntactically valid orders, the endpoint shall return HTTP 200 with a validation result	Robustness testing
FR-N-01	Move the drone in the 16 discrete compass directions	Structural testing
FR-N-02	Move the drone by a fixed distance per step	Boundary testing

FR-N-03	The drone will hover for one move above the restaurant or Appleton Tower when collecting or delivering an order	Path property testing
FR-N-04	The system will avoid entering no-fly zones	Integration testing
FR-N-05	Once the drone enters the Central Area, it will not leave it again	Integration testing
FR-N-06	Return HTTP 400 if no valid delivery path can be found	Negative testing
FR-N-07	For valid orders, the system will return a sequence of positions representing the delivery path	System testing
FR-N-08	The GeoJSON endpoint shall return the same path as the delivery path endpoint, excluding hover steps	Structural validation

Measurable Attributes

ID	Attribute	Metric	Test Implication
MA-01	Correctness	All specified error codes exercised	Specification-based testing
MA-02	Robustness	Correct HTTP status for invalid input	Negative testing
MA-03	Numerical accuracy	Correct handling of geometric thresholds	Boundary testing
MA-04	Execution time	Response time within reasonable limits	Lightweight performance testing

Non-functional requirements

- The system should be secure and keep user data private.
- The system should aim to have the wait time between placing an order and receiving the order be no less than 45 minutes.
- The system should not take less than 60 seconds to find the optimal paths for all orders
- The drone should have 100% availability aside from when it’s charging at Appleton Tower.
- The drone should fly above buildings and stick to rooftops as much as possible rather than flying over streets.