Test Planning Document

Project: Pizza Drone Algorithm Implementation

Introduction:
This document outlines the testing strategy for the PizzaDronz project for software testing. This project is aimed to implement an algorithm that handles and delivers pizza orders efficiently. The following document will detail the testing approach, including the methodology, techniques, and tools to be used throughout the software development life cycle.
I choose to use the Test-Driven Development (TDD) Life Cycle:

Requirements:

- R1: The pizza drone must not drop the pizza while in transit and cannot pass the no-fly zones.
- R2: The maximum planning time for the drone to plan a route must not exceed 1 minute.
- R3: The input URL must be valid for PizzaDronz Server to read data from Rest Server
- R4: The payment must be checked to ensure the validation

Priority and Pre-requisites:

- R1 is a safety requirement and has high priority. We need to allocate a considerable amount of resources to ensure the requirement is met.
- R3 is a properly higher requirement for the PizzaDronz service running properly.
- R4 is a necessary security requirement and has a relatively higher priority than the performance aspect in R2.
- R2 is a lower priority requirement but still important to measure and evaluate the performance of the drone.

Assessment of A&T needs:

- R1:
  - We need to ensure that the drone is capable of carrying the pizza securely in transit while not passing the no-fly zones in the Uni's centre area.
  - We might perform physical tests to check the stability of the drone during flight and its ability to maintain the pizza even though we set the maximum number of pizzas that the drone can carry.

    - We might also perform simulations and computer models to check the drone's performance in various weather conditions and wind speeds.
    - We will conduct a series of tests to confirm that the drone's navigation systems are functioning properly to prevent any accidents during the delivery process and make sure the pizza drone will never cross the no-fly areas.

- R2:
    - We will perform a series of tests to determine whether the algorithm will generate a plan that takes more than 1 minute to plan a route from different restaurants to the Appleton tower.
    - We need to measure the planning time accurately to evaluate the performance of the drone.
    - We might also test the drone's ability to handle multiple orders and navigate it from different restaurants to minimise delivery time.
    - We will monitor the delivery route and make any necessary adjustments to improve delivery speed by checking the flight path and see if it flies directly from Appleton to restaurants.
- R3:
    - We will need to implement the basic check for every input URL with its formats:
    - For example, an URL not ending with (/) will be directly rejected when it is used as input. Or we can implement methods to fix that URL and automatically add (/) for that URL.
    - We must try to access the URL and see if it is valid before we actually grab any data from it to avoid system crashes.
- R4:
    - We will need to check the payment detail and in that case, the CVV and card number must be valid.
    - This can be implemented in the code itself.

Scaffolding and Instrumentation

a. Test Case Design: Test cases will be designed based on the requirements. The test cases will ensure that the algorithm meets the requirements and functions as expected.
   R1: the inspection will not require any scaffolding or instrumentation, but if we plan a later exhaustive test (perhaps exercising the code many times), we will need to build scaffolding that presents randomly

generated inputs and checks that the output meets the spec, Building the scaffolding is another task to be scheduled.
R2: this is more extensive and requires data for the simulator may involve some effort that needs to be scheduled. The system test will include combining the output of single order and the daily routes and using synthetic data to observe its response

b. Code Implementation: The code will be implemented based on the test cases.

c. Test Execution: Test cases will be executed to verify that the code meets the requirements.

d. Debugging: Issues identified during test execution will be fixed.

e. Repeat: The TDD cycle will repeat until all the requirements are met, and the tests pass.

Testing Approach:
The testing approach for the Pizza Drone Algorithm Implementation project will involve the following testing techniques:
a. Unit Testing: Unit testing will be performed to verify the individual components of the algorithm. This will ensure that each component functions as expected.

b. Integration Testing: Integration testing will be performed to verify that the components of the algorithm work together as expected. This will ensure that the algorithm functions as a whole.

c. Performance Testing: Performance testing will be performed to verify that the algorithm can handle high volumes of orders efficiently. This will ensure that the algorithm meets the performance requirements.

d. Debugging: Debugging will be performed as needed to fix any issues identified during testing.

Tools and Techniques:
The following tools and techniques will be used to support the testing approach:
a. Test Automation Tools: Test automation tools will be used to automate the testing process, increasing efficiency and reducing manual effort.

b. Continuous Integration: Continuous integration will be used to automatically build and test the code whenever changes are made. This will ensure that the code is always in a releasable state.

c. Code Review: Code review will be performed to ensure that the code meets the quality standards.

Test Plan Conclusion:
In conclusion, the Pizza Drone Test-Planning Document outlines the requirements and necessary steps to ensure the safe and efficient delivery of pizzas. The high priority of R1 emphasises the importance of safety and the need to allocate sufficient resources to meet this requirement. The lowest priority of R2 highlights the importance of measuring the drone's performance to continuously improve delivery speed and efficiency, whereas R3 and R4 are quite important in making sure the system function properly.