**PizzaDronz Test Plan**

**Overview**

The **PizzaDronz** test plan outlines the testing strategy and process for the **drone-based pizza delivery system**, ensuring the service meets the **functional, non-functional, and performance** requirements specified in the system design.

This test plan follows a **Test-Driven Design (TDD)** approach and evolves as new requirements emerge during development.

**Test Plan Evolution**

As the project developed, new requirements were identified, prompting adjustments to the test plan. Key influencing factors include:

* **Order Tracking**: Introduced real-time order tracking tests to verify **order status updates** and ensure **data integrity**.
* **Error Handling**: Added tests covering **API failures, system crashes**, and edge cases (e.g., invalid orders).
* **Performance**: Implemented performance tests to assess the system’s ability to handle **high-demand periods** and maintain a **responsive user experience**.

**Test Levels and Types**

**5.1. Unit Tests**

* **Order Parsing**: Ensures valid and invalid orders are processed correctly.
* **Drone Navigation**: Verifies that the drone follows **defined flight paths**, avoids **no-fly zones**, and **bypasses obstacles**.

**5.2. Integration Tests**

* **Order Handling Integration**: Tests interactions between the **order API** and **drone delivery functionality**, ensuring correct sequencing.
* **Order Tracking Integration**: Ensures real-time updates of order tracking within the **delivery system**.
* **API Integration**: Verifies that **external APIs** (e.g., order validation, restaurant data, no-fly zones) are handled without errors.

**5.3. System Tests**

* **Full Delivery Cycle Test**: Evaluates the end-to-end process, from **order placement** to **final delivery**.
* **No-Fly Zone Compliance**: Ensures the drone avoids **restricted areas** as per system specifications.

**Instrumentation and Logging**

To ensure comprehensive **tracking and monitoring**, the following instrumentation and logging mechanisms were implemented:

* **Timestamping of Orders**: Logs timestamps for **each order and drone movement** to track lifecycle timing.
* **Movement Data Logging**: Records **flight paths** and **hover times** in structured JSON format for analysis.
* **Performance Metrics**: Logs **response times, delivery durations, and system load** to monitor performance under different conditions.

**Error Handling and Robustness Testing**

The test plan includes robustness tests for various failure conditions:

* **API Failures**: Simulates **timeouts** and **invalid data scenarios**, verifying system resilience.
* **Drone Malfunctions**: Tests how the system reacts to **battery depletion, GPS failures,** and other drone-related issues.
* **Order Validation Issues**: Ensures proper rejection of invalid orders (e.g., **expired payment methods, closed restaurants**).