1. **TEST PLAN IDENTIFIER**

**WM-DLSP1.0**

1. **Introduction :**

This is the Test Plan for the Drone-Launch Scheduler (DLS) Program. The Main objective of this project is to Schedule the Drone Departure time and calculate the Net Promoter Score (NPS) . This Test plan document will address the process of detail testing , testing requirements, risks and contingencies and more., for validating the Drone Launch Schedules and NPS.

Each Module within this project will undergo different types of testing at different levels ,Unit , System/Integration , Functional and Non Functional . The Details of each testing is further defined in later part of the test plan document.

The estimated time line for this project will be 2 months, as such, any

delays in the development process or in the installation and verification of the third party software could have significant effects on the test plan. The acceptance testing is expected to take once the application delivery from system test .

1. **Scope of Testing**

**3.1 System Overview**

The customer interacts with order placement channels like .com or any handheld devices to place the order. Once the order is placed it will flow through the order management and to the order database. A queue server will receive the Drone orders from the orders database. This server will queue all the orders and pushes to the Drone Database once the order is ready to be delivered by the Drone.

All interactions with the database are handled via the Drone Delivery Service (DDS)API layer. The API layer is responsible to provide necessary information for the DLS Micro service Layer requesting it. The DLS Micro Service Layer which has the DLS Core ,Net promoter score business logic, Drone Delivery Time Calculator (DDTC), DLS Generator, DLS Access will in turn provide data to DLS interface. All DLS Micro service Layer calculations and information are stored in Drone DB.

Once the DLS interface fed with input having Order number, Location and Order time eg: WM001 N11W5 05:11:50 , then the interface calls necessary services to get the Drone Schedules. DLS core will do Drone schedules to dispatch the order based on the order timestamp .

NPS Business Engine gets all the necessary details from the Drone DB in order to calculate the Net Promoter score based on the business logic.

Finally , After everything is calculated and updated DLS Generator will generate a file with records in format - WM004 06:31:24 which is order number and Departure time along with NPS as the last record.

[Refer Architectural / System design document for detailed flow]

**3.2 Test Items**

Following are the list of testable items for this project’s functionality.

|  |  |
| --- | --- |
| Drone Orders DB | This Holds the Drone related order details along with customer details. This database is also responsible for maintaining drones schedule - delivery times , NPS etc., |
| Drone – Delivery Services (API Layer) | This Layer interacts DLS Micro service Layer and database in order to provide and process data. |
| DLS Micro Service Layer | This Layer has following servicesDLS Core : This core module is responsible for managing orders ,drone availability, scheduling drone departures. DDTC (Drone Delivery Time Calculator) : This module calculates the Drone Delivery times based on the business logic NPS Business Engine :Calculating the Net Promoter score based on the customer feedbacks. DLS Generator : It is responsible to generate and output the Drone Orders along with schedules and NPS . DLS Access : To fetch orders, Location and other customer details needed.Analytics : Which is for analyzing number of customers doing drone orders , the location which has more orders , frequency of orders and others. |
| D L S (Drone Launch Scheduling) Interface | This is a web interface which interacts with DLS Micro service layer for getting the Drone schedules , NPS Score etc., |

**3.3 Non-Test Items**

Following are the list of non-testable items or out of scope for this project’s functionality.

|  |  |
| --- | --- |
| Drone Order Queue Server / Feed Server | This Module is responsible to maintain a queue for the drone orders placed by customers. The queue will fetch all the necessary info for the Drone delivery system through drone databases and services. |
| Order DB | This Database will hold all the orders related data . |
| Order Management Module | Handles all the orders coming from various channels and process those orders. |
| Client Channel | Client Channel is the customer facing module for placing the orders. |

**3.4 Features to be Tested**

The following is a list of the areas to be focused during testing of the application.

1. Drone Database testing for validating all necessary order transaction updates and recovery procedures.
2. Drone Services which performs HTTP operation from and to with DLS Micro services.
3. NPS Service for Validating and Verifying score
4. DLS Core Service , which is responsible for Drone Schedules
5. DDTC Service ,Responsible for calculating Drone Deliveries
6. DLS Generator Generates the Drone Scheduler list and NPS
7. DLS Access to access records that needs to be processed
8. DLS Web interface for Testing the Drone Schedules, NPS Value

**3.5 Features Not to be Tested**

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts.

1. Queue Messaging server to verify the queued order messages that updates data to the database
2. **Approach**

**4.1 Testing Levels**

The testing for the DLS program will consist of Unit, System/Integration (combined) and Functional and Non- Functional test levels. It is hoped that there will be QE team member for system/integration testing along

with the development team member participation.

UNIT Testing for the DLS Micro Services like DLS core, DDTC , NPS Engine, DLS Generator, DLS Access will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the QE Team.

SYSTEM/INTEGRATION Testing which integrates DLS interface and Drone API’s with external modules will be performed by the QE Team and development team member of the respective modules. Programs will enter into System/Integration test after all critical defects have been corrected. Integration testing may have up to two to three Major defects as long as they do not impede testing of the program.

FUNCTIONAL Testing for DLS Interface and it’s modules is performed by the QE team based on the successful outcomes from the previous levels of testing. In this level both manual and automation scenarios which are identified both at interface and api levels are tested and validate with the System. Any Issues/ Bugs that found need to be resolved by development team on the timely basis to avoid any delays in functional testing. Any new feature updates need to undergo all the above levels of testing before it’s a part of functional testing. All the necessary Test Bed need to be set up before starting the testing. This also involves testing the service layer for validating necessary responses.

NON FUNCTIONAL Testing for DLS Program also involves non-functional aspects of testing such as Performance , Load and stress with implanting and manipulating orders and drone schedules data. This will be managed by Non Functional Test team members . This testing outcome will ensure the given program for the production readiness as per nonfunctional parameters which are not addressed by functional testing. Any issues raised will be handled by development team based on the scenarios designed at the project initiation phase.

USER ACCEPTANCE Testing is done by the product owners / Beta users for taking a final decision about the DLS Program. Given Program will enter into Acceptance test after all critical and major defects have been

Corrected.

**4.2 Testing Tools**

Following are some of the Test tools used in the process of evaluating the DLS Program.

* Test Manager
* Defect Tracker
* Log Recorder
* Version-Control System
* Continuous Integration / Delivery Tool

**4.3 Software Requirements**

For performing functional and Non Functional Testing below are the list of software’s will be used

Functional Testing libraries and tools– Java SDK , Frameworks include Webdriver ,Cucumber, Maven/Gradle, Extent Reports, GIT , Postman / RESTAssured .

Non Functional Testing software’s - HP Load Runner

**4.4 Hardware Requirements**

Mac book pro / Windows Machine with i7 Processor , 16 GB ram , and 256 GB Hard drive

Network availability with WIFI

**4.5 Meetings**

As a part of agile process test team and dev team will have daily stand ups to evaluate progress. Defect Triage meeting will be scheduled based on the particular day’s outcome of defect numbers. Additional meetings can be called as required for emergency situations.

1. **Measures and Metrics**

The following information will be collected by the Development team during the Unit testing process. This information will be provided to the test team at program turnover as well as be provided to the project team on a biweekly basis.

1. Defects by module and severity.

2. Defect Origin (Requirement, Design, Code)

3.Time spent on defect resolution by defect, for Critical & Major only. All Minor

Defects can be totaled together.

The following information will be collected by the test team during all testing phases. This information will be provided on a biweekly basis to the test manager and to the project team.

1. Defects by module and severity.

2. Defect Origin (Requirement, Design, Code)

3. Time spent on defect investigation by defect, for Critical & Major only. All Minor defects can be totaled together.

4. Number of times a program submitted to test team as ready for test.

5. Defects located at higher levels that should have been caught at lower levels of testing.

1. **Test Deliverables**

* Test Plan
* Test Design Specification
* Test Case documents
* Test Data document
* System/Integration test plan
* Unit test plans/turnover documentation
* Defect/Incident reports
* Test summary report
* Non-functional testing reports

These documents will be delivered for every build cycles in order to provide a transparent view from testing towards the program.

1. **Items Pass/Fail Criteria**

To pass the Functional and Non-functional testing following criteria should be met.

1. All Unit testing need to passed with almost having 80% code coverage
2. Integration testing should be considered as pass once all the services are validated
3. Functional testing for evaluating and verifying DLS Micro services and DLS Interface should match the expected output
4. All Non Functional tests are accepted with the if the AUT undergo the baseline metric values
5. **Defect Management**

The assignment and description of defect severity levels will be as follows:

1 -Critical Business objectives or completion of test case are impacted.

2 - High Defects which provide invalid/incorrect information. An example of a priority 2 defect could be a miscalculation of NPS.

3 – Medium Defects are esthetic in nature. An example of a priority 3 defect could be the misalignment of records or misplacement of record values . Functionality is NOT impacted.

4 - Low An item observed during testing that may require further information. This type of priority could be assigned to a work order for an item encountered that is not clear in the requirements.

1. **Suspension Criteria and Resumption Requirements**

As a general guideline, if the defect backlog continually increases over a two week period, testing should be suspended. This will allow the developers time to fix existing defects without the pressure and confusion of new defects being added to the backlog.

When a change is being migrated to the test environment, QE must be notified in advance to schedule a time for the move. After the move has been completed, a retest of previously tested functions should be performed.

If a critical processing unit is found to have severe defects (as defined by the defect reporting process), testing should be suspended until the defects have been fixed. When the fixed unit is moved back into the test environment, any previously performed tests that affect the unit should be performed again to ensure new defects were not created as a result of the fix.

1. **Planning Risks and Contingencies**
   * + - Staffing issues – Any QE unavailability for more than 2 weeks will have an impact over Deliverables , Alternatively additional hours / Weekends should be allotted to do any pending testing deliverables.
       - Environmental Issues – Any time if there is downtime in the environment, need to be handled with the use of Mocks and stubs if it exceeds more than 3 days.
       - Changes to the original requirement in DLS program might have an impact over Deliverables and modifying / adding additional Test Scenarios .
       - If number of defects raised per modules is more than 2 critical defects per sprint then there will be having an impact over meeting the testing deadline
2. **Staffing and Training Needs**

* A Test Team of 2 full time and 1contractor need to be allocated to test the project.
* Test team members should be trained on the services and databases used to test the DLS program
* All Test members should have enough knowledge on api service testing , validating database schemas and Web driver framework. Basic understanding of framework design will be given at the beginning of the project
* Test Member should be given baselines on Non-functional testing parameters

1. **Schedule**

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan time line. Coordination of the personnel required for each task, test team, development team, management and customer will be handled by the project manager in conjunction with the development and test team leaders.

1. Review of Requirements document by test team personnel (with other team members) and initial creation of automation framework and classes.
2. Review of the System design document by test team personnel. This will

provide the team with a clearer understanding of the application structure.

1. Unit test time within the development process
2. Test cases Designed and Reviewed on Every sprint.
3. Automation Development and implementation parallel to sprint work.
4. Execution of Automation once the project is matured for running the automation scripts on daily basis and sharing report
5. **Approvals**

|  |  |
| --- | --- |
| Project Manager |  |
| Product Owner |  |
| Development Team lead |  |
| Test manager |  |
| API Team Lead |  |
| Database Admin |  |
| Change Control and regression Team member |  |
| Devops Engineer |  |

1. **References**

System Design Document

Requirement Document

Developer Design Document

1. **Appendix**

|  |  |
| --- | --- |
| API | Application Programming Interface |
| DDC | Drone Delivery Calculator |
| DLS | Drone Launch Scheduler |
| NPS | Net Promoter score |
| QE | Quality Engineer |