

System Test Plan

For

RTube NeMo Team

Team members: Maegan Lucas, Taylor Sumlin, Adam Fitch, Darian Hopkins,
Maxwell Moolchan

Version/Author	Date
Version 1/ML,TS	10/16/2023

Table of Contents

1.	Introduction	2
1.1	Purpose	2
1.2	Objectives	2
2.	Functional Scope	2
3.	Overall Strategy and Approach	2
3.1	Testing Strategy	2
3.2	System Testing Entrance Criteria	2
3.3	Testing Types	2
3.4	Suspension Criteria and Resumption Requirements	4
4.	Execution Plan	5
4.1	Execution Plan	5
5.	Traceability Matrix & Defect Tracking	8
5.1	Traceability Matrix	8
5.2	Defect Severity Definitions	10
6.	Environment	10
6.1	Environment	10
7.	Assumptions	10
8.	Risks and Contingencies	11
9.	Appendices	11

1. Introduction

1.1 Purpose

This document is a test plan for RTube NeMo System Testing, produced by the RTube NeMo team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

1.2 Objectives

- Meets the requirements, specifications and the Business rules.
- Supports the intended business functions and achieves the required standards.
- Satisfies the Entrance Criteria for User Acceptance Testing.

2. Functional Scope

The Modules in the scope of testing for the RTube Nemo System Testing are mentioned in the document attached in the following path :

1. [System Requirement Specification](#)

3. Overall Strategy and Approach

3.1 Testing Strategy

RTube NeMo System Testing will include testing of all functionalities that are in scope (Refer Functional Scope Section) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, work flows, functionality access, testing of internal & external interfaces.

3.2 System Testing Entrance Criteria

In order to start system testing, certain requirements must be met for testing readiness. The readiness can be classified into: usability and functional testing.

3.3 Testing Types

3.3.1 Usability Testing

User interface attributes, cosmetic presentation and content will be tested for accuracy and general usability. The goal of Usability Testing is to ensure that the User Interface is comfortable to use and provides the user with consistent and appropriate access and navigation through the functions of the application (e.g., access keys, consistent tab order, readable fonts etc.)

System Requirements Specification, 4.3.3.1: The application shall display a choice of a Foreflight-style aeronautical chart or a geographic map.

System Requirements Specification, 4.3.3.2: The application will display the user's chosen map type with the airplane icons, airport icons, and flight paths, when applicable, over the selected map type without changing the functionality of the map.

System Requirements Specification, 4.3.3.3: The application shall display varying icons based on airplane category.

System Requirements Specification, 4.3.3.4: The application shall allow a user to choose their configuration and update accordingly.

System Requirements Specification, 4.4.3.2: The visible waypoints shall update along flight path with airplane icon movement.

System Requirements Specification, 4.5.3.1: The icon for the frequency shall change colors to indicate communication detected on the frequency.

3.3.2 Functional Testing

The objective of this test is to ensure that each element of the component meets the functional requirements of the business as outlined in the:

- Business / Functional Requirements
- Business rules or conditions
- Other functional documents produced during the course of the project i.e. resolution to issues/change requests/feedback

System Requirements Specification, 4.1.3.1: The database shall store communication details such as audio, transcript, date, time, location, and callsign.

System Requirements Specification, 4.1.3.2: The database shall be able to perform queries to recall communication details corresponding with the selected flight path.

System Requirements Specification, 4.1.3.3: The user shall be able to click an airplane icon to view a popup with communication details corresponding with the selected flight path.

System Requirements Specification, 4.1.3.4: The user shall be able to click an airport icon to view a popup with communication details corresponding with the selected airport.

System Requirements Specification, 4.4.3.1: The application shall display a flight path when an airplane is selected by the user with visible waypoints to depict communication in their corresponding locations.

System Requirements Specification, 4.5.3.2: The application shall recognize communication of a certain call sign across different frequencies.

3.4 Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

3.4.1 Suspension Criteria

Testing will be suspended if the incidents found will not allow further testing of the system/application under-test. If testing is halted, and changes are made to the software or database, it is up to the team to determine whether the whole test plan will be re-executed or part of the plan will be re-executed.

Some possible incidents that would halt testing are:

- The waypoints do not line up with the proper location on the flightpath where communication occurred.
- The database does not save data properly.
- The database is unable to query data properly.

3.4.2 Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been retested successfully.

4. Execution Plan

4.1 Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day.

Requirement	Test Case Identifier	Input	Expected Behavior	Pass/Fail
4.1.3.1: The database shall store communication details such as audio, transcript, date, time, location, and callsign.	1.1	An example communication with all necessary details.	Communication information is stored properly in the database.	
4.1.3.2: The database shall be able to perform queries to recall communication details based on user input in the form of a flight path or airport selection.	1.2.1	The callsign of the airplane icon clicked by the user.	All communications with the same callsign are displayed as waypoints on the flightpath in the proper locations.	

4.1.3.2: The database shall be able to perform queries to recall communication details based on user input in the form of a flight path or airport selection.	1.2.2	The callsign of the airplane icon clicked by the user and the location of the waypoint clicked by the user.	The specific communication details with matching input is displayed properly in the information popup.	
4.1.3.3: The user shall be able to click an airplane icon to view a popup with communication details corresponding with the selected flight path.	1.3	The callsign of the airplane icon clicked by the user.	All corresponding communication is properly displayed along with a flight path.	
4.1.3.4: The user shall be able to click an airport icon to view a popup with communication details corresponding with the selected airport.	1.4	The airport information of the icon clicked by the user.	The communication information that correspond to the airport are displayed in a popup	
4.3.3.1: The application shall display a choice of a Foreflight-style aeronautical chart or a geographical map.	3.1	No input, website is ran.	Two buttons displayed that allow choice of map.	

4.3.3.2: The application will display the user's choice map type with the airplane icons, airport icons, and flight paths when applicable over the selected map type without changing the functionality of the map.	3.2	User input via the geographic or aeronautic map option buttons.	Correct map is displayed with plane and airport icons overlaid.	
4.3.3.3: The application shall display varying icons based on airplane category and type.	3.3	Integer category from OpenSky API.	Proper image corresponding to category is displayed for the plane on the map.	
4.3.3.4: The application shall allow a user to choose their configuration and update accordingly.	3.4	Configuration from user input via selection.	Map and features update within 3 seconds to the proper configuration.	
4.4.3.1: The application shall display a flight path when an airplane icon is selected by the user with visible waypoints to depict communication in their corresponding locations.	4.1	Airplane icon selection.	A flight path is displayed from departure location to current airplane location (depicted by airplane icon), with waypoints in corresponding locations to communications.	

4.4.3.2: The visible waypoints shall update with airplane icon movement.	4.2	Selected airplane icon's given location via OpenSky API.	Waypoints update and are added as plane icon moves along flight path/	
4.5.3.1: The icon for the frequency shall change colors to indicate communication detected on the frequency.	5.1	Audio input on radio frequency.	The frequency's icon changes to the selected color.	
4.5.3.2: The application shall recognize communication of a certain call sign across different frequencies.	5.2	Audio from radio frequencies and call sign from selected airplane icon.	All communications involving the given call sign are able to be queried no matter the frequency.	

Table 4.1 The execution plan with detail about tests for each requirement from the System Requirements Specification document, what the expected result should be, and whether the test for the requirement has passed or failed.

5. Traceability Matrix & Defect Tracking

5.1 Traceability Matrix

5.1.1 Critical Requirements

- *System Requirements Specification, 4.1.3.1:* The database shall store communication details such as audio, transcript, date, time, location, and callsign.
 - *Test Case(s):* 1.1
- *System Requirements Specification, 4.1.3.2:* The database shall be able to perform queries to recall communication based on user input in the form of a flight path or airport selection.
 - *Test Case(s):* 1.2.1, 1.2.2

- *System Requirements Specification, 4.1.3.3:* The user shall be able to click an airplane icon to view a popup with communication details corresponding with the selected flight path.
 - *Test Case(s):* 1.3
- *System Requirements Specification, 4.1.3.4:* The user shall be able to click an airport icon to view a popup with communication details corresponding with the selected airport.
 - *Test Case(s):* 1.4
- *System Requirements Specification, 4.3.3.2:* The application will display the user's choice map type with the airplane icons, airport icons, and flight paths, when applicable, over the selected map type without changing the functionality of the map.
 - *Test Case(s):* 3.2
- *System Requirements Specification, 4.4.3.1:* The application shall display a flight path when an airplane icon is selected by the user with visible waypoints to depict communication in their corresponding locations.
 - *Test Case(s):* 4.1
- *System Requirements Specification, 4.5.3.2:* The application shall recognize communication of a certain call sign across different frequencies.
 - *Test Case(s):* 5.2

5.1.2 Medium Requirements

- *System Requirements Specification, 4.3.3.1:* The application shall display a choice of a Foreflight-style aeronautical chart or a geographic map.
 - *Test Case(s):* 3.1
- *System Requirements Specification, 4.3.3.4:* The application shall allow a user to choose their configuration and update accordingly.
 - *Test Case(s):* 3.4

5.1.3 Low Requirements

- *System Requirements Specification, 4.3.3.3:* The application shall display varying icons based on airplane category.
 - *Test Case(s):* 3.3
- *System Requirements Specification, 4.4.3.2:* The visible waypoints shall update with airplane icon movement.
 - *Test Case(s):* 4.2
- *System Requirements Specification, 4.5.3.1:* The icon frequency shall change colors to indicate communication detected on the frequency.
 - *Test Case(s):* 5.1

5.2 Defect Severity Definitions

Critical	The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows: <ul style="list-style-type: none">• System abends• Data cannot flow through a business function/lifecycle• Data is corrupted or cannot post to the database
Medium	The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows: <ul style="list-style-type: none">• Form navigation is incorrect• Field labels are not consistent with global terminology
Low	The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows: <ul style="list-style-type: none">• Repositioning of fields on screens• Text font on reports is incorrect

6. Environment

6.1 Environment

In order to conduct the testing, the tester needs to have the following installed on their computer:

- Python, version 9 or greater
- Cython
- OpenSky Python API
- Leaflet, version 1.9.3
- JQuery, version 3.7.0
- Leaflet Rotated Marker Plug-In, version 0.2.0
- Flask, version 2.3.2
- Pandas, version 2.0.1
- Openpyxl, version 3.1.2

7. Assumptions

- All communication information being stored in the database are the correct file format.
- OpenSky API server is running properly.

8. Risks and Contingencies

Risks and contingencies will be added as they are identified.

9. Appendices