

System Test Plan

For

RTube NeMo Team

Team members:

Maegan Lucas, Maxwell Moolchan, Darian Hopkins, Taylor Sumlin, Adam Fitch

| Version | Author | Date | Description of Changes |
|---------|------------------|-------------------|--|
| 1.0 | Maegan Lucas | 16 October, 2023 | Initial creation of document |
| 1.1 | Maegan Lucas | 16 October, 2023 | Added Sections 3.1.1, 3.1.2 |
| 1.2 | Taylor Sumlin | 17 October, 2023 | Added 4.1 Table with information |
| 1.3 | Maegan Lucas | 17 October, 2023 | Edited Sections 2, 3.1.1, 3.1.2, 3.3.1, 3.3.2 |
| 1.4 | Maegan Lucas | 17 October, 2023 | Edited & Revised Section 4.1, Minor text edits |
| 1.5 | Maegan Lucas | 18 October, 2023 | Added Sections 5.1.1, 5.1.2, 5.1.3 |
| 1.6 | Maegan Lucas | 18 October, 2023 | Edited Sections 4.1, 5.1.1, Minor text edits, Updated Table of Contents |
| 1.7 | Maegan Lucas | 18 October, 2023 | Edited Section 3.2, Minor text edits |
| 1.8 | Maxwell Moolchan | 31 October, 2023 | Updated Revision history log to be more detailed, Minor text edits |
| 2.0 | Maegan Lucas | 01 December, 2023 | Updated all sections for final version. |
| 2.1 | Maegan Lucas | 03 December, 2023 | Updated all sections for final version. |
| 2.2 | Maegan Lucas | 04 December, 2023 | Updated all sections for final version. |
| 2.3 | Maegan Lucas | 05 December, 2023 | Updated all sections for final version. |
| 2.4 | Maegan Lucas | 12 February, 2024 | Updated section 3.3 to include new requirements. |
| 2.5 | Maegan Lucas | 25 February, 2024 | Updated sections 4.1 and 5.1 to include new requirements and tests. Updated Table of Contents. Add new requirements to section 3.3 |

| | | | |
|-----|--------------|---------------|-------------------------------|
| 2.6 | Maegan Lucas | 7 April, 2024 | Updated based on V1 feedback. |
|-----|--------------|---------------|-------------------------------|

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Objectives

2. Functional Scope

3. Overall Strategy and Approach

- 3.1 Testing Strategy
- 3.2 System Testing Entrance Criteria
- 3.3 Testing Types
 - 3.3.1 Usability Testing
 - 3.3.2 Functional Testing
- 3.4 Suspension Criteria and Resumption Requirements
 - 3.4.1 Suspension Criteria
 - 3.4.2 Resumption Requirements

4. Execution Plan

- 4.1 Execution Plan

5. Traceability Matrix & Defect Tracking

- 5.1 Traceability Matrix
- 5.2 Defect Severity Definitions

6. Environment

- 6.1 Environment

7. Assumptions

8. Risks and Contingencies

9. Appendices

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.

Please note: Some tests that will be completed this semester have not yet occurred. These will be marked with *Not Yet Tested*.

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: useability 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, 4.1.3.5: 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, 4.1.3.6: 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, 4.3.3.1: The application shall display a drop-down menu that drops down to show buttons upon clicking.

System Requirements Specification, 4.3.3.2: The application shall display a geographic map button, a Hybrid Visual Flight Rules button, a Sectionals button, a Helicopters button, a Low Instrument Flight Rules button, and a High Instrument Flight Rules button in the expanded drop down menu.

System Requirements Specification, 4.3.3.3: The application shall display a geographic map within 10 seconds of the geographic map button being selected.

System Requirements Specification, 4.3.3.4: The application shall display a Hybrid Visual Flight Rules map within 10 seconds of the Hybrid Visual Flight Rules button being selected.

System Requirements Specification, 4.3.3.5: The application shall display a Sectionals map within 10 seconds of the Sectionals button being selected.

System Requirements Specification, 4.3.3.6: The application shall display a Helicopters map within 10 seconds of the Helicopters button being selected.

System Requirements Specification, 4.3.3.7: The application shall display a Low Instrument Flight Rules map within 10 seconds of the Low Instrument Flight Rules button being selected.

System Requirements Specification, 4.3.3.8: The application shall display a High Instrument Flight Rules map within 10 seconds of the High Instrument Flight Rules button being selected.

System Requirements Specification, 4.3.3.9: The application shall display the appropriate airplane icon depending on the airplane's category.

System Requirements Specification, 4.3.3.10: The application shall allow a user to choose their configuration.

System Requirements Specification, 4.3.3.11: The application shall update to the chosen configuration within 10 seconds.

System Requirements Specification, 4.3.3.12: The application shall display a settings button on map initialization.

System Requirements Specification, 4.3.3.13: The settings panel shall be displayed on the left hand side of the screen within 5 seconds of the Settings button being selected.

System Requirements Specification, 4.3.3.14: The settings panel shall display a slider for map brightness value selection.

System Requirements Specification, 4.3.3.15: The map brightness value shall have a default of 0.35 upon page initialization.

System Requirements Specification, 4.3.3.16: The map brightness value shall not go below 0.

System Requirements Specification, 4.3.3.17: The map brightness value shall not exceed 0.7.

System Requirements Specification, 4.3.3.18: The map brightness value shall equal the user inputted value upon brightness slider interaction.

System Requirements Specification, 4.4.3.1: The application shall display a flight path when an airplane icon is selected by the user.

System Requirements Specification, 4.4.3.2: The flight path shall display visible waypoints to depict communication in their corresponding locations.

System Requirements Specification, 4.4.3.3: The visible waypoints shall update

along flight path with airplane icon movement.

System Requirements Specification, 5.1.3: The application shall display an error message when the user zooms out past the map tile zoom limit.

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 accurately store communication details in the form of callsign, time, date, longitude, latitude, audio, transcript, and a boolean value indicating whether the communication is coming from ATC.

System Requirements Specification, 4.1.3.2: The database shall accurately store flight details in the form of callsign, flight number, departure location, and arrival location.

System Requirements Specification, 4.1.3.3: The database shall be able to perform queries to recall communication details based on the callsign given by user input in the form of a flight icon selection.

System Requirements Specification, 4.1.3.4: The database shall be able to perform queries to recall communication details based on given user input in the form of an airport icon selection.

System Requirements Specification, 4.1.3.7: The database shall accurately store airport data in the form of id, type, name, longitude, latitude, elevation, country name, region name, local regio, municipality, GPS code, International Civil Aviation Code, local code, home webpage link, and a list of stream frequencies by reading an Excel file.

System Requirements Specification, 4.1.3.8: The database shall accurately recall airport information if the type of the airport is a “medium_airport”.

System Requirements Specification, 4.1.3.9: The database shall accurately recall airport information if the type of the airport is a "large_airport".

System Requirements Specification, 4.2.3.1: The application shall transcribe communication data with a 30% or better Word Error Rate (WER) when the user selects the "Transcribe" button for a specific flight within 10 seconds.

System Requirements Specification, 4.2.3.2: The application shall display transcribed audio within 15 seconds of selecting the "Transcribe" button.

System Requirements Specification, 4.2.3.3: The audio files shall be given from NearAero as .MP3 format.

System Requirements Specification, 4.2.3.4: The system shall convert .MP3 audio files to .WAV files.

System Requirements Specification, 4.2.3.5: The system shall only accept .txt files as output from NVIDIA NeMo.

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, 4.5.3.1: 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.

- NVIDIA NeMo does not transcribe audios with a 30% Word Error Rate or less.

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/Not Yet Tested |
|---|----------------------|--|---|--------------------------|
| <ul style="list-style-type: none">■ Req. 4.1.3.1 The database shall accurately store communication details in the form of callsign, time, date, longitude, latitude, audio, transcript, and a boolean value indicating whether the communication is coming from ATC. | 1.1 | callsign = '123456789', time = '12:00:00', date = '2023-12-01', longitude = '123.456789 E', latitude = '123.456789 N', audio = 'test.wav', transcript = 'test.txt', atc = 'FALSE' | Database shows callsign = '123456789', time = '12:00:00', date = '2023-12-01', longitude = '123.456789 E', latitude = '123.456789 N', audio = 'test.wav', transcript = 'test.txt', atc = 'FALSE' as an entry to Communications table. | Pass |

| | | | | |
|--|-----|--|---|------|
| <p>■ Req. 4.1.3.2 The database shall accurately store flight details in the form of callsign, flight number, departure location, and arrival location</p> | 1.2 | <p>callsign = '123456789', flight number = '12345', departure location = 'MCO', arrival location = "LAX"</p> | <p>Database shows callsign = '123456789', flight number = '12345', departure location = 'MCO', arrival location = "LAX"</p> | Pass |
| <p>■ Req. 4.1.3.3 The database shall be able to perform queries to recall communication details based on callsign given by user input in the form of a flight icon selection.</p> | 1.3 | <p>Test Airplane: callsign = '123456789'</p> | <p>Query produces: Database shows callsign = '123456789', time = '12:00:00', date = '2023-12-01', longitude = '123.45678 9E', longitude = '123.45678 9 E', audio = 'test.wav', transcript = 'test.txt', atc = 'FALSE'</p> | Pass |
| <p>■ Req. 4.1.3.4 The database shall be able to perform queries to recall communication details based</p> | 1.4 | <p>Test Airport: callsign = '987654321'</p> | <p>Query produces: Database shows</p> | Pass |

| | | | | |
|--|-----|---|--|------|
| on callsign given by user input in the form of an airport icon selection. | | | callsign = '978654321', time = '12:00:00', date = '2023-12-01', longitude = '123.456789E', latitude = '123.456789 N', audio = 'test.wav', transcript = 'test.txt', atc = 'TRUE' | |
| Req. 4.1.3.5 The user shall be able to click an airplane icon to view a popup with communication details corresponding with the selected flight path. | 1.5 | Test Airplane is clicked on map. Callsign = '123456789' | Information popup displays with communication: callsign = '123456789', time = '12:00:00', date = '2023-12-01', longitude = '123.456789E', latitude = '123.456789 N', audio = 'test.wav', | Pass |

| | | | | |
|---|-----|--|--|------|
| | | | transcript = 'test.txt', atc = 'FALSE' | |
| <p>■ Req. 4.1.3.6 The user shall be able to click an airport icon to view a popup with communication details corresponding with the selected airport.</p> | 1.6 | <p>Test Airport clicked on map: Callsign = '987654321'</p> | <p>Information popup displays with communication: callsign = '978654321', time = '12:00:00', date = '2023-12-01', longitude = '123.456789 E', latitude = '123.456789 N', audio = 'test.wav', transcript = 'test.txt', atc = 'TRUE'</p> | Pass |
| <p>Req. 4.1.3.7 The database shall accurately store airport data in the form of id, type, name, longitude, latitude, elevation, country name, region name, local region, municipality, GPS code, International Civil Aviation Code, local code, home webpage link, and a list of</p> | 1.7 | <p>id = '1234', type = "medium_airport", name = "Test Airport", longitude = "123.456789 S", latitude = '123.456789</p> | <p>Database shows: id = '1234', type = "medium_airport", name = "Test Airport",</p> | Pass |

| | | | | |
|---|-----|--|---|------|
| stream frequencies by reading an Excel file. | | E', elevation = '100', country_name = 'United States', region_name = 'Florida', municipality = 'Daytona Beach', gps_code = 'KTA', iata_code = 'KTA', local_code = 'KTA', home_link = 'testairport.com', stream_freqs = '1, 2, 3' | longitude = "123.45678 9 S", latitude = '123.45678 9 E', elevation = '100', country_name = 'United States', region_name = 'Florida', municipality = 'Daytona Beach', gps_code = 'KTA', iata_code = 'KTA', local_code = 'KTA', home_link = 'testairport.com', stream_freqs = '1, 2, 3' | |
| Req 4.1.3.8 The database shall accurately recall airport information if the type of airport is a "medium_airport". | 1.8 | type = 'medium_airport' | Query recalls: id = '1234', type = "medium_airport", name = "Test Airport", longitude = "123.45678 9 S", latitude = '123.45678 9 E', elevation = | Pass |

| | | | | |
|---|-----|--|---|------|
| | | | '100', country_name = 'United States', region_name = 'Florida', municipality = 'Daytona Beach', gps_code = 'KTA', iata_code = 'KTA', local_code = 'KTA', home_link = 'testairport.com', stream_frequencies = '1, 2, 3' | |
| Req 4.1.3.9 The database shall accurately recall airport information if the type of airport is a "large_airport" | 1.9 | type = 'large_airport' Airport = id = '4321', type = "large_airport", name = "Test Airport", longitude = "123.456789 N", latitude = '123.456789 W', elevation = '100', country_name = 'United States', region_name = 'Florida', municipality = 'Daytona | Query Results: Airport = id = '4321', type = "large_airport", name = "Test Airport", longitude = "123.456789 N", latitude = '123.456789 W', elevation = '100', country_name = 'United States', region_name = | Pass |

| | | | | |
|--|-----|---|---|------|
| | | Beach', gps_code = 'KTAL', iata_code = 'TAL', local_code = 'TAL', home_link = 'testairport.co m', stream_freqs = '1, 2, 3' | e= = 'Florida', municipality = 'Daytona Beach', gps_code = 'KTAL', iata_code = 'TAL', local_code = 'TAL', home_link = 'testairport. com', stream_freq s = '1, 2, 3' | |
| <p>■ Req. 4.2.3.1 The application shall transcribe communication data with a 30% or better WER when the user selects the “Transcribe” button for a specific flight within 10 seconds.</p> | 2.1 | Test communication seen in Test Case 1.1 excluding the transcript. | Within 10 seconds of “Transcribe” button being clicked, the transcription is produced with an error rate of 30% or less. | Pass |
| <p>■ Req. 4.2.3.2 The application shall display transcribed audio to the user within 15 seconds of selecting the “Transcribe” button.</p> | 2.2 | Test communication seen in Test Case 1.1. | Within 15 seconds of the “Transcribe” button being clicked, test.txt is displayed on the screen. | Pass |
| Req. 4.2.3.3 The audio files | 2.3 | Audio | Audio | Pass |

| | | | | |
|---|-------|---------------------------------|--|----------------|
| shall be given from NearAero as .MP3 format. | | streamed from NearAero. | should be format .MP3. | |
| Req. 4.2.3.4 The system shall convert .MP3 audio files to .WAV files. | 2.4 | Audio streamed from NearAero. | Program produces same audio streamed from NearAero in .WAV format. | Pass |
| Req. 4.2.3.5 The system shall only accept .txt files as output from NVIDIA NeMo. | 2.5.1 | Transcription with .txt format. | Program accepts file and outputs transcript. | Not Yet Tested |
| Req. 4.2.3.5 The system shall only accept .txt files as output from NVIDIA NeMo. | 2.5.2 | Transcription with .doc format. | Program rejects file and raises error. | Not Yet Tested |
| ■ Req. 4.3.3.1 The application shall display a drop-down menu that drops down to show buttons upon clicking. | 3.1.1 | Web application ran. | Drop-down menu displays. | Pass |
| ■ Req. 4.3.3.1 The application shall display a | 3.1.2 | Drop-down menu clicked. | Drop-down menu | Pass |

| | | | | |
|---|-----|---|---|------|
| drop-down menu that drops down to show buttons upon clicking. | | | expands to show buttons. | |
| <p>■ Req. 4.3.3.2 The application shall display a geographic map button, a Hybrid Visual Flight Rules button, a Sectionals button, a Helicopters button, a Low Instrument Flight Rules button, and a High Instrument Flight Rules button in the expanded drop down menu.</p> | 3.2 | Drop-down menu clicked. | Geographic map button and aeronautical map buttons are visible in the menu. | Pass |
| <p>■ Req. 4.3.3.3 The application shall display a geographic map within 10 seconds of the geographic map button being selected.</p> | 3.3 | On any aeronautical Map view. Geographic map button clicked. | Geographic map is displayed within 10 seconds of the button being clicked. | Pass |
| <p>■ Req. 4.3.3.4 The application shall display a Hybrid Visual Flight Rules map within 10 seconds of the Hybrid Visual Flight Rules button being selected.</p> | 3.4 | On any other map view. Hybrid Visual Flight Rules button clicked. | Hybrid Visual Flight Rules map displayed within 10 seconds of the button being clicked. | Pass |
| <p>■ Req. 4.3.3.5 The application shall display a Sectionals map within 10 seconds of the Sectionals button being selected.</p> | 3.5 | On any other map view. Sectionals button clicked. | Sectionals map displayed within 10 seconds of the button being clicked. | Pass |

| | | | | |
|--|------|--|---|------|
| <p>■ Req. 4.3.3.6 The application shall display a Helicopters map within 10 seconds of the Helicopters button being selected.</p> | 3.6 | On any other map view. Helicopters button clicked. | Helicopters map displayed within 10 seconds of the button being clicked. | Pass |
| <p>■ Req. 4.3.3.7 The application shall display a Low Instrument Flight Rules map within 10 seconds of the Low Instrument Flight Rules button being selected.</p> | 3.7 | On any other map view. Low Instrument Flight Rules button clicked. | Low Instrument Flight Rules map displayed within 10 seconds of the button being clicked. | Pass |
| <p>■ Req. 4.3.3.8 The application shall display a High Instrument Flight Rules map within 10 seconds of the High Instrument Flight Rules button being selected.</p> | 3.8 | On any other map view. High Instrument Flight Rules button clicked. | High Instrument Flight Rules map displayed within 10 seconds of the button being clicked. | Pass |
| <p>■ Req. 4.3.3.9 The application shall display the appropriate airplane icon depending on the airplane's category.</p> | 3.9 | Airplane category = '1' (No ADS-B Emitter Category Information) | planeCategory1.svg displayed. | Pass |
| <p>■ Req. 4.3.3.10 The application shall allow a user to choose their configuration</p> | 3.10 | Web application ran. | Choice of configuration will be shown. | Pass |

| | | | | |
|---|------|--|---|------|
| | | | | |
| <p>■ Req. 4.3.3.11 The application shall update to the chosen configuration within 10 seconds.</p> | 3.11 | Configuration choice clicked. | Configuration display properties are displayed within 10 seconds. | Pass |
| <p>■ Req. 4.3.3.12 The application shall display a settings button on map initialization.</p> <p>■</p> | 3.12 | Interactive map page is entered. | Settings button shall display. | Pass |
| <p>■ Req. 4.3.3.13 The settings panel shall be displayed on the left hand side of the screen within 5 seconds of the Settings button being selected.</p> | 3.13 | Settings button clicked. | Settings panel shown on the left hand side of the screen. | Pass |
| <p>■ Req. 4.3.3.14 The settings panel shall display a slider for map brightness value selection.</p> | 3.14 | Settings button clicked and the settings panel opened. | Map brightness slider is displayed. | Pass |
| <p>■ Req. 4.3.3.15 The map brightness value shall have a default of 0.35 upon page initialization.</p> | 3.15 | Interactive map page entered. | Map brightness value = 0.35 | Pass |

| | | | | |
|---|------|--|---|----------------|
| | | | | |
| ■ Req. 4.3.3.16 The map brightness value shall not go below 0. | 3.16 | Map brightness value = -1 | Map brightness value = 0 | Pass |
| ■ Req. 4.3.3.17 The map brightness value shall not exceed 0.7. | 3.17 | Map brightness value = 7.1 | Map brightness value = 7 | Pass |
| ■ Req. 4.3.3.18 The map brightness value shall equal the user inputted value upon brightness slider interaction. | 3.18 | Map brightness value = 0.65 given via slider selection. | Map brightness value = 0.65 | Pass |
| ■ Req. 4.4.3.1 The application shall display a flight path when an airplane icon is selected by the user. | 4.1 | Airplane is selected. | Flight path is displayed. | Pass |
| ■ Req. 4.4.3.2 The flight path shall display visible waypoints to depict communication in their corresponding locations. | 4.2 | Test Airplane data matching test communication in Test Case 1.1. | Waypoints are shown at longitude and latitude pair of test communication. | Not Yet Tested |

| | | | | |
|--|-----|--|---|----------------|
| <p>■ Req. 4.4.3.3 The visible waypoints shall update with airplane icon movement.</p> | 4.3 | <p>Test Airplane moved to new location. Second test communication with different longitude, latitude, pair.</p> | <p>Waypoint for second test communication also shown at accurate longitude, latitude pair.</p> | Not Yet Tested |
| <p>■ Req. 4.5.3.1 The application shall be able to store communication details from the same callsign with different frequencies.</p> | 5.1 | <p>Test communication from Test Case 1.1. Test communication 2: callsign = '123456789', time = '2:00:00', date = '2023-12-01', longitude = '222.222222 E', latitude = '123.456789 N', audio = 'test.wav', transcript = 'test.txt', atc = 'FALSE'</p> | <p>Both communication from Test Case 1.1 and Test communication 2 are stored in the database.</p> | Pass |
| <p>Req. 5.1.1 The application shall not refresh plane icons in less than 5 seconds.</p> | 6.1 | <p>Starting time of 6:31:25 PM.</p> | <p>Next call should be after 6:31:30 PM.</p> | Pass |

| | | | | |
|--|-----|-----------------------------------|----------------------------------|------|
| Req. 5.1.2 The application shall display an error message when the user zooms out past the map tile zoom limit. | 6.2 | Zoom level < 8, Zoom limit = 8 | Error displayed. Map blurred. | Pass |
|--|-----|-----------------------------------|----------------------------------|------|

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

○

| Communications Database | | |
|---------------------------------|-------------|-----------------|
| Requirement ID | Criticality | Test Case ID(s) |
| 4.1.3.1 | Critical | 1.1 |
| 4.1.3.2 | Critical | 1.2 |
| 4.1.3.3 | Critical | 1.3 |
| 4.1.3.4 | Critical | 1.4 |
| 4.1.3.5 | Critical | 1.5 |
| 4.1.3.6 | Critical | 1.6 |
| 4.1.3.7 | Critical | 1.7 |
| 4.1.3.8 | Critical | 1.8 |
| 4.1.3.9 | Critical | 1.9 |
| Transcription Using NVIDIA NeMo | | |
| Requirement ID | Criticality | Test Case ID(s) |
| 4.2.3.1 | Critical | 2.1 |
| 4.2.3.2 | Critical | 2.2 |

| | | |
|--------------------------------|--------------------|------------------------|
| 4.2.3.3 | Critical | 2.3 |
| 4.2.3.4 | Critical | 2.4 |
| 4.2.3.5 | Critical | 2.5.1, 2.5.2 |
| User Interface Update | | |
| Requirement ID | Criticality | Test Case ID(s) |
| 4.3.3.1 | Medium | 3.1.1, 3.1.2 |
| 4.3.3.2 | Medium | 3.2 |
| 4.3.3.3 | Medium | 3.3 |
| 4.3.3.4 | Medium | 3.4 |
| 4.3.3.5 | Medium | 3.5 |
| 4.3.3.6 | Medium | 3.6 |
| 4.3.3.7 | Medium | 3.7 |
| 4.3.3.8 | Medium | 3.8 |
| 4.3.3.9 | Low | 3.9 |
| 4.3.3.10 | Medium | 3.10 |
| 4.3.3.11 | Medium | 3.11 |
| 4.3.3.12 | Low | 3.12 |
| 4.3.3.13 | Low | 3.13 |
| 4.3.3.14 | Low | 3.14 |
| 4.3.3.15 | Low | 3.15 |
| 4.3.3.16 | Low | 3.16 |
| 4.3.3.17 | Low | 3.17 |
| 4.3.3.18 | Low | 3.18 |
| Communication Waypoints | | |
| Requirement ID | Criticality | Test Case ID(s) |
| 4.4.3.1 | Critical | 4.1 |

| | | |
|-------------------------------------|--------------------|------------------------|
| 4.4.3.2 | Critical | 4.2 |
| 4.4.3.3 | Low | 4.3 |
| Communication Identification | | |
| Requirement ID | Criticality | Test Case ID(s) |
| 4.5.3.1 | Critical | 5.1 |
| Performance Requirements | | |
| Requirement ID | Criticality | Test Case ID(s) |
| 5.1.1 | Critical | 6.1 |
| 5.1.2 | Low | 6.2 |

○

5.2 Defect Severity Definitions

| | |
|-----------------|--|
| Critical | <p>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:</p> <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 | <p>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:</p> <ul style="list-style-type: none">• Form navigation is incorrect• Field labels are not consistent with global terminology |
| Low | <p>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:</p> <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
- SQL Alchemy, version 2.0.22
- 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.
- All necessary information is being passed from OpenSky API.

8. Risks and Contingencies

No risks or contingencies have been identified.

9. Appendices