System Requirements Specification

for

RTube NeMo Team

Version 1.0 approved

Prepared by Taylor Sumlin, Maegan Lucas

Embry-Riddle Aeronautical University

September 29, 2023

Table of Contents

1. Introduction **3**
1.1 Purpose 3
1.2 Document Conventions 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Product Scope 3
1.5 References **3**
2. Overall Description **4**
2.1 Product Perspective 4
2.2 Product Functions 4
2.3 User Classes and Characteristics 4
2.4 Operating Environment 4
2.5 Design and Implementation Constraints 4
2.6 User Documentation 4
2.7 Assumptions and Dependencies 4
3. External Interface Requirements **5**
3.1 User Interfaces 5
3.2 Hardware Interfaces 5
3.3 Software Interfaces 5
3.4 Communications Interfaces 5
4. System Features **5**
4.1 Communication Database 5
4.2 Communication Transcription and Kaldi Integration
4.3 User Interface Update 6
4.4 Communication Waypoints 7
4.5 Communication Identification 8
5. Other Nonfunctional Requirements **9**
5.1 Performance Requirements 9
5.2 Safety Requirements 9
5.3 Security Requirements 9
5.4 Software Quality Attributes 9
5.5 Business Rules 10
6. Other Requirements **10**

Revision History

Name	Date	Reason For Changes	Version
TS, ML	9/29/23	SRS First Draft	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to describe and communicate the system requirements of the NeMo Team. This team will be working on the development and integration of a database to store ATC communications data and be able to effectively display the data through a web application.

1.2 Document Conventions

Font: Time New Roman
Font Size: 12

1.3 Intended Audience and Reading Suggestions

This document is intended for stakeholders and ongoing developers and future members of the project. The SRS is sectioned off to relevant areas primarily discussing the software requirements and program specifications going into detail about user classes, cases and other interactions with the system.

1.4 Product Scope

This project capsulizes two teams to work on this ongoing project. The Kaldi team, which is in charge of developing and innovating upon the existing AI technologies to be able to convert ATC (Air Traffic Control) chatter to text. The purpose of this is to be able to utilize text data to be able to better understand communications for pilots, ATC Controllers & other relevant parties. The NeMo team focuses on integrating this AI model into a web application as well as developing a database to be able to store ATC and flight data.

1.5 References

Current Github Repository: https://github.com/maeganlucas/CS490-ATC

Previous Team GitHub Repository:
https://github.com/Burnetb8/Senior-Capstone/tree/main/Documentation

2. Overall Description

2.1 Product Perspective

The product is an improvement of an existing infrastructure of a web application that utilizes AI ASR models to convert ATC chatter to text. While this is not the direct scope of the NeMo team, the application will be integrated into the existing web application framework to continuously improve upon the functioning flight tracking web app.

2.2 Product Functions

- Successfully track and display accurate flight data
- Provide recordings of ATC communications
- Transcribe ATC communication using AI models to be stored

2.3 User Classes and Characteristics

To be determined

2.4 Operating Environment

The operating environment of this project will be through a web application that will be hosted online. This will be used to integrate the Kaldi ASR models to provide data to the web app.

2.5 Design and Implementation Constraints

As there has not been significant development within the product, there are not yet of many known constraints

2.6 User Documentation

Currently user documentation is provided through text files that provide instructions and assistance for installing all project dependencies and instructions to install and run the program.

2.7 Assumptions and Dependencies

** To be determined**

3. External Interface Requirements

3.1 User Interfaces

The user interface will be through a web application. The design is similar to existing flight tracking services such as FlightAware or FlightRadar24. This user interface allows for the user to interact with specific flights and be able to listen in on recorded ATC communications. The user is additionally presented with options to filter aircraft & select individual flights.
Subject to change

3.2 Hardware Interfaces

The product is entirely based through a web application and there are no hardware interfaces being used throughout the project.

3.3 Software Interfaces

The NeMo team will be developing a database to connect Kaldi ASR models to the web application. This database will store data relating to ATC communications as well as general flight data. As it is early in the development.

3.4 Communications Interfaces

To be determined

4. System Features

4.1 Communication Database

4.1.1 Description and Priority

A database is to be developed to hold attributes of communications. This includes communication audio, transcript, time, date, location, and callsign. This is of High priority for the project. This has the highest benefit to the further development of the application as it allows for storing communications and important attributes that can be used for more features.

4.1.2 Stimulus/Response Sequences

When a user selects a flight path or airport, the database will query for the communications relevant to that flight path or airport.

4.1.3 Functional Requirements

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

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.

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.

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.

4.2 Communication Transcription and Kaldi Integration

4.2.1 Description and Priority

The web application shall integrate with the ASR models of the Kaldi team to be able to display transcriptions of communications. This is of High priority to the application. Integrating the ASR models will involve communication and collaboration with the Kadi team.

4.2.2 Stimulus/Response Sequences

When a user selects the “Transcribe” button when viewing communication audios, a transcription of that audio will be displayed on the information popup.

4.2.3 Functional Requirements

4.2.3.1 The system shall be able to integrate with speech-recognition models that will transcribe flight communication audio.

4.2.3.2 The application shall display audio transcriptions when the transcribe button is pressed by a user.

4.2.3.3 The user shall be able to press the transcribe button and view a transcript in the information popup.

4.3 User Interface Update

4.3.1 Description and Priority

The user interface will be updated and will be polished as a whole. This will include adding a Foreflight-style chart option in addition to the geographical map, adding different icons for different airplane categories and types, and smoothing the movement of the airplane icons along their flight paths. This will also include the ability to have different configurations for varying use-cases, such as different flight schools. This is of Medium priority.

4.3.2 Stimulus/Response Sequences

4.3.2.1 The user will be able to select between a geographical map and the aeronautical chart. Once selected, the application will display the user's choice.

4.3.2.2 The user will choose a configuration and the application will update accordingly.

4.3.3 Functional Requirements

4.3.3.1 The application shall display a choice of a Foreflight-style aeronautical chart or a geographical map.

4.3.3.2 The user shall be able to choose between a Foreflight-style aeronautical chart or a geographical map.

4.3.3.3 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.

4.3.3.4 The application shall display varying icons based on airplane category and type.

4.3.3.5 The application shall allow a user to choose their configuration and update accordingly.

4.3.3.6 The user shall be able to choose a configuration.

4.4 Communication Waypoints

4.4.1 Description and Priority

Waypoints, or “breadcrumbs,” will be added to flight paths to depict where communication took place. They will update with the movement of the airplane icons. This is a Medium priority to the application.

4.4.2 Stimulus/Response Sequences

When a user selects an airplane icon, the flight path will be displayed with waypoints to depict where communication took place.

4.4.3 Functional Requirements

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.

4.4.3.2 The visible waypoints shall update with airplane icon movement.

4.5 Communication Identification

4.5.1 Description and Priority

Communications will be able to be tracked across frequencies using callsigns. A user will also be able to identify when someone is communicating over a specific frequency by the color of the icon. This is a medium priority.

4.5.2 Stimulus/Response Sequences

4.5.2.1 When communication is detected on a frequency, the icon for the frequency will change to indicate live use to the user.

4.5.2.2 The user will select an airplane icon and the system will track the communications of the corresponding callsign across frequencies to display along the flight path.

4.5.3 Functional Requirements

4.5.3.1 The icon for the frequency shall change colors to indicate communication detected on the frequency.

4.5.3.2 The application shall recognize communication of a certain call sign across different frequencies.

4.6 Radio Communication Type Identification

4.6.1 Description and Priority

The application will be able to identify the type of radio communication to the user, for example: standard phraseology, plain aeronautical English, etc. This is of Low priority as this is a stretch goal for the project.

4.6.2 Stimulus/Response Sequences

The user will choose an airplane or airport icon and the transcribe button and the type of radio communication will be displayed with the transcription.

4.6.3 Functional Requirements

4.6.3.1 The application shall display the type of radio communication on the information popup once the transcribe button is clicked.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The final product needs to be able to run efficiently and without stutters as the program will be tracking live flight data from around the world as well as pulling data in from the database to assist in transcribing ATC communications.

5.2 Safety Requirements

To be determined

5.3 Security Requirements

To be determined

5.4 Software Quality Attributes

To be determined

5.5 Business Rules

To be determined

6. Other Requirements

To be determined

Appendix A: Glossary

To be determined

Appendix B: Analysis Models

To be created

Appendix C: To Be Determined List

To be determined