System Requirements Specification

for

Speech Recognition for Air Traffic Control

Version 3.0 approved

Prepared by Braeden Burnett, Jakob Haehre, Kira McFadden, Tyler Carr

Embry-Riddle Aeronautical University

March 10, 2023

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
TC	9/17/22	Initial document creation	1
KM	10/3/22	Added Map References, added common definitions table, added scope items	1.1
KM	10/4/22	Added definitions to table, and table description. Added General purpose statement and fall mission statement.	1.2
TC	10/4/22	Added Display Interactive Map section to system features and outline some functionality	1.2
TC	10/6/22	Added Product Perspective, Added Product Functions, Added User Classes and Characteristics, Added Operating Environment, Expanded on System Features	1.3
KM	10/7/22	Edited formatting Updated Table of Contents, Removed descriptions for sections for submission purposes	1.4
KM	11/1/22	Added User Requirements, added advanced user requirements, added software requirements, edited format of revision history table	2.0
TC	11/1/22	Added SIR requirements for website	2.0
JH	11/1/22	Added SIR requirements for model(s)	2.0
BB	11/8/22	Added Design / Implementation requirements	2.0
KM	11/8/22	Added details to previously blank sections, added use case diagram.	2.0
TC	11/8/22	Add class diagram, add to other requirements and security requirements, and update requirements	2.0
TC	12/2/22	Update requirements, use cases, figures, and formatting in response to Aaron's feedback.	3.0
BB	12/3/22	Finish changes using Aaron's feedback and finalized formatting of document	3.1
TC	3/8/23	Add missing website requirements to match with the test plan	3.2
TC	3/9/23	Fix formatting for Google Drive	3.3
KM	3/10/23	Added information to test cases, fixes of formatting issues	3.4
TC	3/10/23	Fix formatting, add new definitions, edit wording of existing requirements.	3.5
BB	3/10/23	Finish Formatting for submissions look over and add neural model case information	3.6

1. Introduction

1.1 Purpose

Flight training can be difficult when it comes to flight planning, aeronautical sectional map reading, and understanding Air Traffic Control (ATC) communications. The software should aid in flight training by allowing users to track real-time flights, display live ATC communications, and toggle between interactive maps and aeronautical sectional charts.

Fall Mission Statement
By the end of the first semester, our group plans to complete the display of an aeronautical map on addition to an interactive map and support various scales for the chart. We also plan on being able to transcribe the live speeches from LiveATC by building a special speech recognizer to transcribe these communications.

Spring Mission Statement
By the end of the second semester, our group plans to have all the basic functionality that was initially requested by the customer completed. The model will be trained and adjusted so the word error rate is as low as possible. Additionally, the website will have both the interactive and the aeronautical chart maps with plane icons that you can click to fetch data about the plane. ATC transmissions from LiveATC will be streamed by the website and transcribed by the model, which will then be displayed on the website.

1.2 Document Conventions

Font: Times New Roman
Base font size: 11

1.3 Intended Audience and Reading Suggestions

Intended Audience

Primary stakeholders: Associate Professor and Program Coordinator for M.S. in Electrical and Computer Engineering: Jianhua Liu; Aviation English Coordinator, and Adjunct Faculty: Andrew Schneider
Secondary stakeholders: ERAU Flight Department

1.4 Product Scope

The software is a web-based ASR map that displays an interactive map with the ability to toggle to an aeronautical sectional chart for the Jacksonville area. The map section of the software has several scales to allow for readability as well as a zoom function for the user to enlarge key locations on the map. The real-time flight tracker is displayed over both maps with icons representing the individual planes, and each plane can be clicked to display information about the flight. Using the LiveATC website, real-time ATC communications are run through a speech recognizer using Nvidia NeMo. The transcribed text from NeMo will then be displayed on the website alongside the plane information.

There is some advanced functionality requested by the client if the team has enough time in the semester after finishing the core functionalities. Additional audio functionalities could be included, such as replay, save, play, pause, fast-forward, rewind, and stop. Additional transcript controls could also be beneficial, such as the ability

to upload, save and edit. The editing feature would be crowdsourced, with a team of moderators needed to review the edit suggestions. Navigational marking features could be included, as well as having different modes such as replay and live modes, the ability to toggle certain layers of the VFR map on or off, displaying other sectional areas within the United States besides only the Jacksonville sectional, and allowing the users to create and save custom maps.

Things outside of the product scope include isolation of noncommunication sound from audio files and creating a help window to show the step-by-step process of how to use the product.

1.5 References

Map References

1. VFR Charts: https://www.faa.gov/air_traffic/flight_info/aeronav/digital_products/vfr/
2. Leaflet: https://leafletjs.com/
   a. https://www.earthdatascience.org/tutorials/introduction-to-leaflet-animated-maps/
   b. https://medium.com/@shachiakyaagba_41915/integrating-folium-with-dash-5338604e7c56
3. Plotly Dash: https://dash.plotly.com/layout

NeMo References

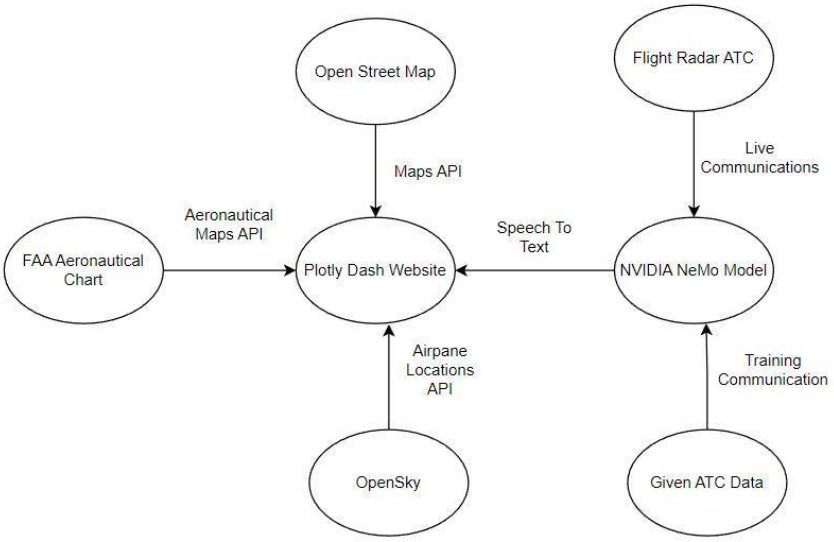
1. https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/stable/starthere/intro.html
2. https://docs.nvidia.com/deeplearning/riva/user-guide/docs/support-matrix.html

2. Overall Description

2.1 Product Perspective

The product being specified in this SRS was an idea by Dr. Liu of Embry- Riddle Aeronautical University in Daytona Beach, Florida. The product is an improvement of a concept that already exists in multiple instances on the internet. Websites such as Flightradar24 display interactive maps with icons of aircraft updates in real time as they travel around. Additional details for flights are also displayed upon request. The end goal is to recreate this functionality but extend it with adding real-time ATC communication transcriptions for each aircraft upon request. This transcription will be displayed along with the flight

information in a popup window when an aircraft is clicked.



2.2 Product Functions

There are three main groups of product functions for this system. Each group's functions are summarized below. The functions represent what is being performed on the system by the group.

User-performed:
- Clicking on an aircraft.
- Toggling between the interactive map and VFR map.
- Zooming and panning the map.

Product-performed:
- Display aircraft icons overlayed on top of the map in accurate locations.
- Refresh the aircraft icons' locations on the map.
- Retrieve the flight information and ATC communication data for a selected flight.
- Refresh the flight information and ATC communication data for a selected flight.
- Transcribe the ATC communication and display on the page.

Maintainer-performed:
- Training the model with audio and transcriptions that already have been proven accurate.
- Verifying transcriptions made by the model through created Word Error Rate scripts.

2.3 User Classes and Characteristics

The two user classes that are the most prevalent are user and maintainer. The user is just anyone who is accessing the website, and the maintainer ensures the transcriptions keep working well.

User:

- Description: The user will want to move the map, toggle the map type, browse flights, and view ATC transcriptions. The user can be anyone accessing the website, so the website should be intuitive enough for any knowledge level.
- Expertise: The user is expected to have basic knowledge about flight tracking and pilot and controller terms given they are interested in the website's content.
- Importance: High
- Description: The maintainer will perform the initial training of the model and validate future model predictions for transcriptions.
- Expertise: The maintainer is expected to have in-depth knowledge about pilot and controller terms. They should be able to recognize mistakes in the transcriptions of ATC audio data.
- Importance: Medium

2.4 Operating Environment

The website will be running on a Python 3 web server that will be hosted on a Linux machine provided to us by Dr. Liu. It must be able to open the port for visitors to access the website, and it must have internet access to be able to retrieve data from APIs. The website will be built using Plotly Dash 2.8.0 and accompanying Python packages available on the Python Package Index repository, which include, Pytz, and Dash DAQ. The maps will be built using Dash Leaflet. The plane data will be pulled using the OpenSky API.

The model training must be performed on a dedicated Nvidia graphics card. The machine being provided to us has a graphics card that meets these requirements, so remote access to this machine will be set up to perform the training even when not present. NeMo also requires Python 3.6 or higher and Pytorch 1.8.1 or higher.

No.	**ER 1**
Statement	The user shall have Python version 3.6 or higher.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Any system where the web server or model are run.

No.	**ER 2**
Statement	The user shall have Pytorch 1.8.1 or higher
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	

Where to Implement	Any system where the web server or model are run.

No.	**ER 3**
Statement	The user shall run the program on Windows, Linux, or IOS.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Any device where the user is trying to load the website.

No.	**ER 4**
Statement	The website shall pass all other tests in Chrome 107 or later.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Any device where someone is testing the website.

No.	**ER 5**
Statement	The website shall pass all other tests in Firefox 106 or later.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Any device where someone is testing the website.

No.	**ER 6**
Statement	The website shall pass all other tests in Safari 16 or later.
Source	
Dependency	
Conflicts	
Supporting Material	

Evaluation Method	
Revision History	
Where to Implement	Any device where someone is testing the website.

2.5 Design and Implementation Constraints

- The system must account for supporting multiple APIs, since they are all developed independently of one another.
- Integration of Website and Neural Modeling.
- The model must be running on a computer with a compatible Nvidia graphics card.

2.6 User Documentation

This section identifies documentation that will be delivered with our product.

A basic user guide will be delivered with the product. This will include the identification of key components of the layout of the website, such as the different maps, the components of the maps, the information included about the planes, and the purpose of the website in general. Functionality that is not immediately apparent to the user upon first impression of the website is also documented, such as the ability to click on a plane to view additional information. The user guide is not limited by these things.

There will not be a guide available for the code. However, there will be markdown files titled “README” available in key components of the code’s directories to explain how to set up the code for additional development. Code commenting will be available.

2.7 Assumptions and Dependencies

The real-time flight tracker depends on the OpenSky API. This API restricts the number of requests that can be made per day. When this request limit is reached, the website will no longer be able to fetch coordinates or information for any plane.

The Nvidia NeMo model will be required to run the website, and this requires a Nvidia GPU. The website must be hosted on a machine that contains a GPU that is capable of running the model.

3. External Interface Requirements

3.1 User Interfaces

No.	**USR 1**
Statement	The user shall click a toggle button to switch between interactive and VFR maps in 1 second.
Source	
Dependency	
Conflicts	

Supporting Material	
Evaluation Method	Select the toggle button, verify if VFR is displayed then select toggle button again and verify if VFR is no longer displayed.
Revision History	
Where to Implement	Website

No.	**USR 2**
Statement	The user shall click a plane icon to select a plane on the map.
Source	
Dependency	OpenSky API
Conflicts	
Supporting Material	
Evaluation Method	Move cursor to any plane and left click. Verify if information is displayed in the textbox.
Revision History	
Where to Implement	Website

No.	**USR 3**
Statement	The user shall click and drag the textbox to move it with the cursor.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	After a plane is selected, use the cursor to left click and hold. Move the cursor and verify that box moves with it.
Revision History	
Where to Implement	Website

No.	**USR 4**
Statement	The user shall click the plus button in the top left corner of the map to display the next most detailed level of tile size.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	When the mouse wheel is used the map will zoom in if scrolled forwards and the map will zoom

	out if the wheel is scrolled backwards.
Revision History	
Where to Implement	Website

No.	**USR 5**
Statement	The user shall click the minus button in the top left corner of the map to display the next less detailed level of tile size.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**USR 6**
Statement	The user shall scroll forwards with their cursor over the map to display the next more detailed level of tile size.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**USR 7**
Statement	The user shall scroll backwards with their cursor over the map to display the next less detailed level of tile size.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 1**
Statement	The user shall press the play button to resume the transcription text of the ATC transmission.
Source	
Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 2**
Statement	The user shall press the pause button to temporarily stop the transcribing of the ATC transmission.
Source	
Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 3**
Statement	The user shall press the fast-forward button to speed up the transcribing of the recorded ATC transmission until the transcription reaches the latest available transmission data.
Source	
Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 4**
Statement	The user shall press the rewind button to reverse the replay of the transcription data.
Source	

Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 5**
Statement	The user shall be able to submit a typed correction of the transcript if an error is found.
Source	
Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 6**
Statement	The user shall be able to upload an audio file of transcripts to the central server.
Source	
Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 7**
Statement	The user shall be able to upload old and edited versions of transcripts to the central server.
Source	
Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 8**
Statement	The user shall be able to select a specific tower to view the transcribed transmissions from.
Source	

Dependency	Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 9**
Statement	The user shall be able to select multiple navigation aids and landmarks within a set range in relation to the selected tower.
Source	
Dependency	AUSR 8, Base product completion
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

3.2 Hardware Interfaces

No.	**HIR1**
Statement	The machine running the server shall have a GPU on the list of compatible Nvidia GPUs (See NeMo Reference under Section 1.5 above)
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**HIR2**
Statement Source	The machine running the server shall have an internet connection of at least 25MB/s download speed.
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**HIR3**
Statement Source	The machine running the server shall have an internet connection of at least 25MB/s upload speed.
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

3.3 Software Interfaces

No.	**SIR1**
Statement	The system shall pull a stream of ATC audio from LiveATC.com and provide it to the Nvidia Nemo Model(s) every 15 seconds.
Source	
Dependency	LiveATC.net, Nvidia NeMo
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	NeMo Model

No.	**SIR2**
Statement	The Nvidia Nemo Model(s) shall accept a stream of audio in the form of an array of numbers representing a waveform and return a string containing the words spoken.
Source	
Dependency	Nvidia NeMo, Pytorch, Cuda
Conflicts	
Supporting Material	
Evaluation Method	The model(s) shall have a word error rate below 0.2 The model(s) shall have a word error rate per utterance below 0.2
Revision History	
Where to Implement	NeMo Model

No.	**SIR3**
Statement	The Transcript Parser shall accept a string of ATC dialog and return a list of changes to be

	made to the website.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	NeMo Model

No.	**SIR4**
Statement	The website shall display the interactive map in 1 second after the site is loaded.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	Upon launching the site, wait 1 second and see if the map is displayed.
Revision History	
Where to Implement	Website

No.	**SIR 5**
Statement	The website shall display the VFR map in 1 second after the site is loaded and the toggle button is clicked.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	Upon the selection of the toggle button, wait 1 second and see if the VFR map is displayed.
Revision History	
Where to Implement	Website

No.	**SIR 6**
Statement	The website shall display a scale in miles at the bottom left corner of the map.
Source	
Dependency	
Conflicts	

Supporting Material	
Evaluation Method	Upon launching the website, look in the bottom lefthand corner and verify if the scale is displayed.
Revision History	
Where to Implement	Website

No.	**SIR 7**
Statement	The model(s) shall accept .wav files and maintain their word error rate and word error rate per utterance
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**SIR 8**
Statement	The system shall display selected aircraft tracking information in 1 second after clicking on a plane.
Source	https://www.flightradar24.com/SWA2335/2e0fc599
Dependency	OpenSky API
Conflicts	
Supporting Material	
Evaluation Method	Upon selection of any plane, verify if the planes data is displayed in the textbox. See if displayed information lines up with data in OpenSky CSV file.
Revision History	
Where to Implement	Website

No.	**SIR 9**
Statement	The program shall always display a textbox.
Source	
Dependency	
Conflicts	
Supporting Material	

Evaluation Method	A textbox should be always present on the website.
Revision History	
Where to Implement	Website

No.	**SIR 10**
Statement	The program shall allow the user to move the textbox upon clicking and dragging with the mouse.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	Upon the selection of an aircraft, when the textbox is clicked and held when the cursor moves the box moves in the correct direction.
Revision History	
Where to Implement	Website

No.	**SIR 11**
Statement	The website shall display the interactive map and the toggle button on the initial page load.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	Upon launched the website, verify if the interactive map and a toggle button, located near the top, appear.
Revision History	
Where to Implement	Website

No.	**SIR 12**
Statement	The map shall pan with the cursor when the user clicks and holds the mouse button while moving the cursor.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	Upon click and hold sequence, when the cursor moves the map verify if the map moves in the corresponding direction.
Revision History	
Where to Implement	Website

No.	**SIR 13**
Statement	The planes shall change position to new coordinates when new information is fetched from the plane API every 15 seconds.
Source	
Dependency	OpenSky API
Conflicts	
Supporting Material	
Evaluation Method	Upon launching the website, wait 15 seconds and verify if the aircrafts move.
Revision History	
Where to Implement	Website

No.	**SIR 14**
Statement	The information panel shall load with information about the flight after the user clicks on a plane on the map.
Source	
Dependency	OpenSky API
Conflicts	
Supporting Material	
Evaluation Method	Upon a plane's selection, see if the textbox contains information. See if said information lines up with the aircraft data CSV.
Revision History	
Where to Implement	Website

No.	**SIR 15**
Statement	The information text box shall update the live transcription every 15 seconds when it is visible.
Source	
Dependency	Nvidia NeMo
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Website

No.	**SIR 17**
Statement	The website shall decrease the scale when the user uses the mouse wheel to zoom in on the map.

Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	Upon the mouse wheel being scrolled forwards, see if the map zooms in, then look in the bottom lefthand corner to see if the scale changed.
Revision History	
Where to Implement	Website

No.	**SIR 18**
Statement	The website shall increase the scale when the user uses the mouse wheel to zoom out on the map.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	Upon the mouse wheel being scrolled backwards, see if the map zooms out, then look in the bottom lefthand corner to see if the scale changed.
Revision History	
Where to Implement	Website

No.	**SIR 19**
Statement	Each plane icon shall rotate in the direction of the heading that the plane is traveling in when new plane data is fetched.
Source	
Dependency	OpenSky API
Conflicts	
Supporting Material	
Evaluation Method	Upon launching the website, select one plane and wait 15 seconds and see if the plane's nose changes position.
Revision History	
Where to Implement	Website

No.	**SIR 20**
Statement	The information box shall update the selected plane's info with new data every 15 seconds when the plane's data is fetched.
Source	
Dependency	OpenSky API
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Website

No.	**SIR 21**
Statement	The map shall prevent zooming out further if the user zooms out further than the specified threshold.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Website

No.	**SIR 22**
Statement	The map shall prevent zooming in further if the user zooms in further than the specified threshold.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Website

No.	**SIR 23**
Statement	The system shall cache the current coordinates and information of each plane if there when there is no update available from the OpenSky API.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	Website

3.4 Communications Interfaces

No.	**COMR 1**
Statement	The program shall be able to interact with the webserver.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

4. System Features

This section outlines the functional requirements for the product. It is organized by use cases.

4.1 Display Interactive Map

4.1.1 Description and Priority

4.1.1.1 On the main page of the website, an interactive map is displayed. It is a high requirement because it is the main feature of the website.

4.1.2 Stimulus/Response Sequences

4.1.2.1 The user loading the website causes the website to load first.

4.1.3 Functional Requirements

4.1.3.1 The interactive map shall be loaded within 10 seconds upon the initial load of the website.

4.2 Display Planes on Maps

4.2.1 Description and Priority

4.2.1.1 Overlayed on top of the maps, icons of aircraft for every active flight with data available are shown. They are updated every 1 second.

4.2.2 Stimulus/Response Sequences

4.2.2.1 When the map is moved, the plane icons should stay at the coordinates that they belong i.e. the planes move when the map moves.

4.2.3 Functional Requirements

4.2.3.1 The user shall be able to click on a plane in the interactive map to view a popup with any available information about the flight.

4.2.3.2 The user shall be able to click on a plane in the VFR map to view a popup with any available information about the flight.

4.2.3.3 The planes on the interactive map will stay at the actual coordinates that they belong to when the map is moved.

4.2.3.4 The planes on the VFR map will stay at the actual coordinates that they belong to when the map is moved.

4.3 Display Flight Information on Interactive Map

4.3.1 Description and Priority

4.3.1.1 After the user selects an aircraft icon on the map, the website will show a popup with available flight information. This will include things such as flight name, destination and origin.

4.3.2 Stimulus/Response Sequences

4.3.2.1 The popup is displayed when the user selects an aircraft icon on the map. If the user clicks away, the popup goes away.

4.3.3 Functional Requirements

4.3.3.1 The user shall be able to click on any aircraft icon on the map to display a popup with flight information for that flight.

4.3.3.2 The flight information shall include original airport name and location, final airport name and location, aircraft type, aircraft registration, what country the aircraft is registered to, and current latitude and longitude coordinates.

4.4 Display ATC Transcription on Interactive Map

4.4.1 Description and Priority

4.4.1.1 On the popup that appears when the user selects an aircraft icon, a live transcription of the ATC communication is displayed below the aircraft information. This transcription is updated in real time.

4.4.2 Stimulus/Response Sequences

4.4.2.1 The popup is displayed when the user selects an aircraft icon on the map. If the user clicks away, the popup goes away.

4.4.3 Functional Requirements

4.4.3.1 The user shall be able to view the live transcription of ATC communications for any aircraft that they click the icon for.

4.4.3.2 The live transcription of ATC communications shall be updated once every 3 minutes.

4.5 Toggle between Interactive and Detailed Map

4.5.1 Description and Priority

4.5.1.1 There is a toggle button on the main page of the website that, when clicked, toggles the interactive and detailed maps. The map that is shown when the website is initially loaded is the interactive map. When clicking the button, it will be switched to the detailed map.

4.5.2 Stimulus/Response Sequences

4.5.2.1 The map is toggled when the toggle button is clicked. When one map is displayed, the other is hidden.

4.5.3 Functional Requirements

4.5.3.1 The interactive map shall be replaced with the detailed map when the user clicks the toggle button.

4.6 Controls for Maps

4.6.1 Description and Priority

4.6.1.1 Both the interactive and detailed maps will have the same map controls. These include buttons to zoom in and out, and the ability to click and drag to pan around the map.

4.6.2 Stimulus/Response Sequences

4.6.2.1 Both versions of the map will be interactive. Clicking and dragging on the map in any direction will cause the map to move with the mouse. This feature is called panning. There will also be buttons to zoom in and out on the maps.

4.6.3 Functional Requirements

4.6.3.1 The user shall press the plus button to zoom in on the interactive map.
4.6.3.2 The user shall press the minus button to zoom out on the interactive map.
4.6.3.3 The user shall scroll the mouse wheel forward to zoom in on the interactive map.
4.6.3.4 The user shall scroll l the mouse wheel back to zoom out on the interactive map.

4.6.3.5 The user shall click and drag on the map in any direction to make the interactive map move in that same direction.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Performance requirements outline expectations for how fast the system should perform actions.
5.1.1 The planes shall update every 10 seconds with new coordinates.
5.1.2 The website shall fully load within 10 seconds of visiting the URL.
5.1.3 The interactive maps shall load within 10 seconds of visiting the URL.
5.1.4 The map shall load within 2 seconds when the toggle button is clicked.
5.1.5 The information panel shall update the live transcription every 15 seconds.

5.2 Safety Requirements

Safety requirements are included to reduce the possibility for damage or data loss that the product could cause.
5.2.1 The code for the website shall be committed to GitHub daily when changes are made.
5.2.2 The code for the neural model shall be committed to GitHub daily when changes are made.
5.2.3 The neural model shall implement a stopping feature if it runs out of memory to run to prevent crashing.

5.3 Security Requirements

5.3.1 The software shall be able to check if references files are corrupt.
5.3.2 The software shall use an HTTPS SSL connection for all requests.
5.3.3 PM2 shall be used to keep the web server running even after the server reboots.
5.3.4 The server shall only expose port 433 for the web server's SSL connections.

5.4 Software Quality Attributes

Software quality attributes are traits that the product must have.

5.4.1 The website shall implement all interactive map functionality to be controlled by the mouse.
5.4.2 The neural model shall make its results available via an API so the website can access them.
5.4.3 The neural model shall make predictions that are correct to an accuracy score of 80%.

5.5 Business Rules

Business rules define functions that certain roles can perform under certain circumstances.
5.5.1 Website Team: Add functionality to website.
5.5.2 ASR Team: Train models using NeMo.

6. Other Requirements

6.1 OpenSky API shall return flight information for all flights in set range.
6.2 VFR maps shall be retrieved from the faa.gov website.
6.3 OpenStreetMap shall be used as a free alternative to Google Maps.

Appendix A: Glossary

Common Definitions

Name	**Definition**
ASR	Automated Speech Recognition. Allows users to input information via speech rather than inputting information using a keyboard.
ATC	Air Traffic Control. Traffic controlling facility used in the United States for the purpose of directing air traffic
API	Application Programming Interface. Software intermediary which allows multiple applications to communicate with one another.
GUI	Graphical User Interface. Multimedia interface user interacts with to use program.
NeMo	Neural Models. An NVIDIA toolkit for building AI models with ASR, NLP, and TTS models.
NLP	Natural Language Processing. A subcategory of machine learning where a computer derives meaning from written text.
TTS	Text-to-Speech. Software that can understand text and convert it to written text.
VFR	Visual Flight Rules A set of regulations that an aircraft can use to operate under clear, sunny weather conditions.
LiveATC	A website that provides live audio streaming of ATC transmissions from major airports worldwide.
GPU	Graphical Processing Unit. Dedicated GPUs exist in many computers and are intended to handle heavy computational loads for graphics. In the case of our website, the GPU runs the model used for transcribing audio.
Interactive Map	The interactive map is a tiled map that can be panned around, zoomed in, and zoomed out. New tiles, or chunks, of map are loaded as the map is moved around. A road map is used.
Aeronautical Chart Map	Also referred to as the VFR map, this map can also be panned and zoomed, and works with similar tiling functionality. This map is different because it shows multiple layers of an aeronautical chart on top of the outline of land.

Table 1: Common definitions for acronyms and unknown terms found throughout the document.

Appendix B: Analysis Models

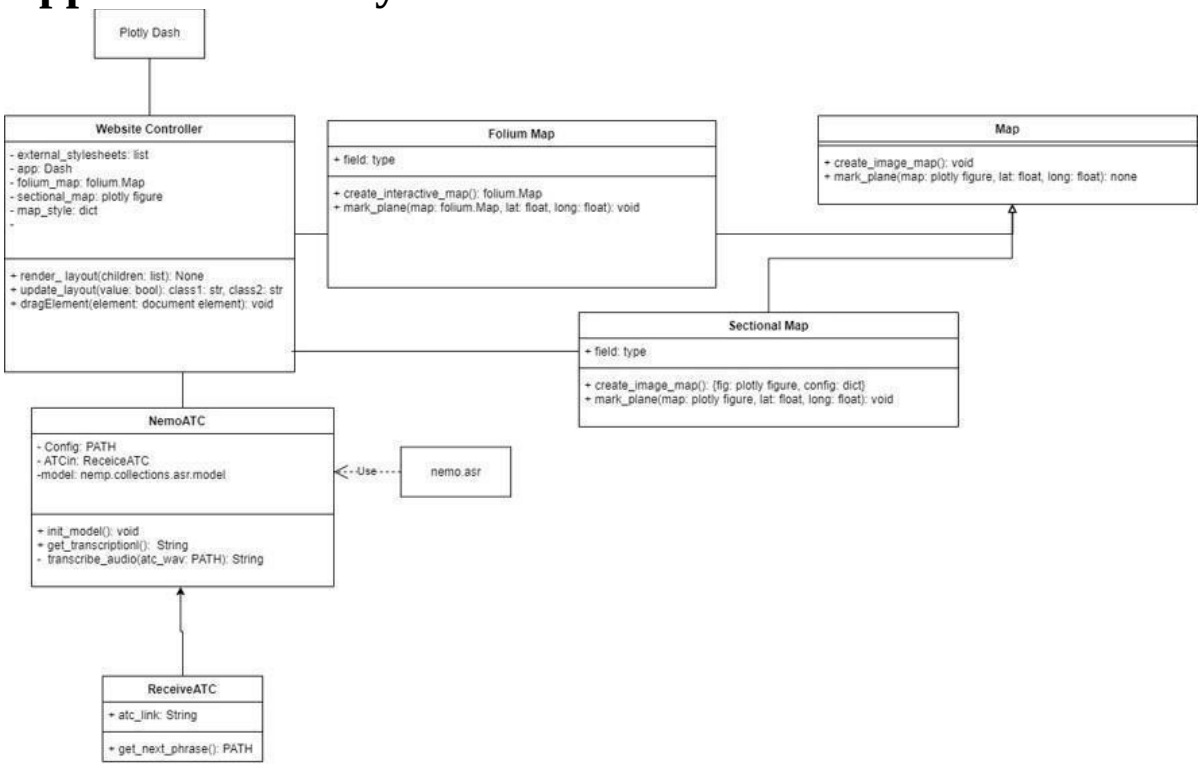


Figure 2: Combined NeMo Neural Model and Website Class Diagram

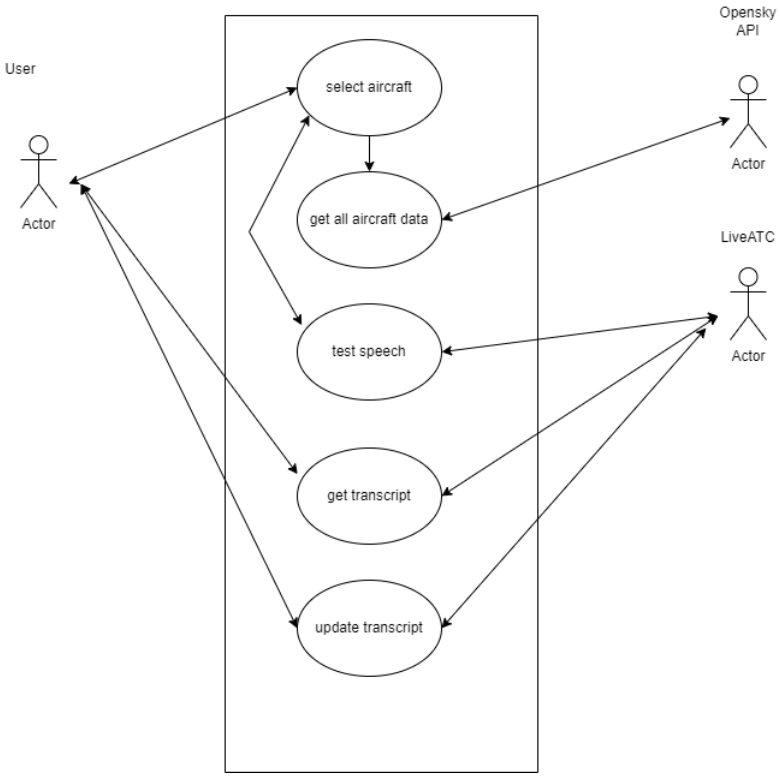


Figure 3: Use case diagram