System Requirements Specification

for

Speech Recognition for Air Traffic Control

Version 2.0 approved

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

Embry-Riddle Aeronautical University

November 8, 2022

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

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 Google 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 a Google 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.

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 will have a web-based ASR map, that can display a version of Google Maps with the ability to toggle to an aeronautical sectional chart for the Jacksonville area. The map section of the software will have several scales to allow for readability as well as a zoom function to highlight key locations on the map. The real-time flight tracker should be displayed over both maps and be able to display information like how FlightRadar24 reflects data. Using the LiveATC website, we will take real-time ATC communications and run them through a speech recognizer using NeMo. The transcribed file from NeMo will then be displayed in the website within a movable text box.
For the time being, the current goal is to have the maps fully functional, meeting all the requirements stated above, with the real-time flights overlayed on the map. On the ASR, we want to have a minimum of three models trained with an accuracy of 0.2.
There are a variety of items that could potentially be included in the scope. 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. 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, displaying other sectional areas outside of just the Jacksonville one, 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

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

NeMo References

- https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/stable/starthere/intro.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.

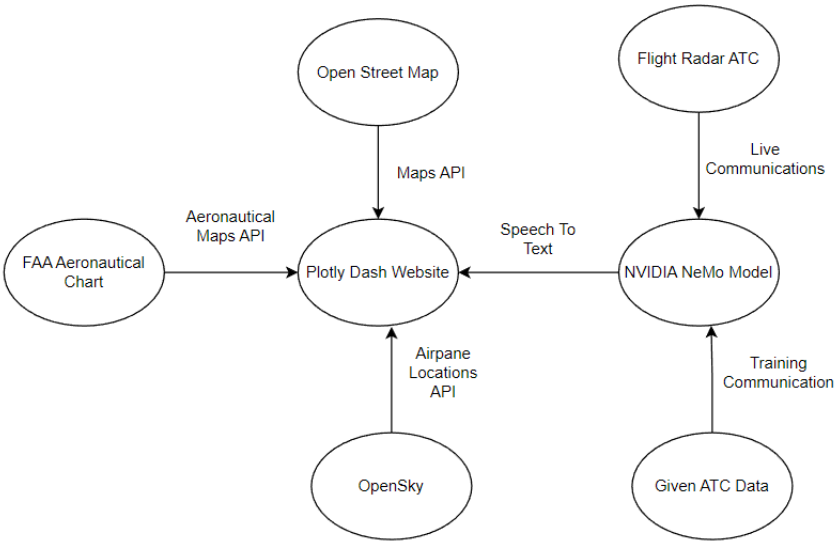

Figure 1: *This diagram shows the major components of the system and how they interact with each other in a high-level architecture diagram. The user is interacting with the Plotly Dash website, which then gets its information from the other aspects of the system.*

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 on the map
- Retrieve 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

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

Maintainer:
- 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 2.6.1 and accompanying Python packages available on the Python Package Index repository, which include Imagecodecs, Numpy, and Pillow, and Dash DAQ. The interactive map will be built using Folium, which in the background uses Leaflet and OpenStreetMap. The VFR map will be rendered using Plotly Express.

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	

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	

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	

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	

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	

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	

2.5 Design and Implementation Constraints

- Intergration of 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, but 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 work. Code commenting will be available at a meaningful level.

2.7 Assumptions and Dependencies

3. The real-time flight tracker depends on the OpenSky API, and the API restricts the number of requests that can be made per day.External Interface Requirements

3.1 User Interfaces

No.	**USR 1**
Statement	The user shall be able to use a button to switch between interactive and VFR Maps
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	Created V1
Where to Implement	

No.	**USR 2**
Statement	The user shall be able to select a plane on the map.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**USR 3**
Statement	The user shall be able to move the textbox.

Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**USR 4**
Statement	The user shall be able to zoom in and out of the maps.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**USR 5**
Statement	The user shall be able to move around the map.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 1**
Statement	The user shall be able to press the play button to resume the replay.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 2**
Statement	The user shall be able to press the pause button to stop the replay.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 3**
Statement	The user shall be able to press the fast-forward button to speed up the reply.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 4**
Statement	The user shall be able press the rewind button to reverse the reply.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 5**
Statement	The user shall be able to edit the transcript if they find an error.
Source	
Dependency	
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	
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	
Conflicts	
Supporting Material	

Evaluation Method	
Revision History	
Where to Implement	

No.	**AUSR 8**
Statement	The user shall be able to select a specific tower.
Source	
Dependency	
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
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

3.2 Hardware Interfaces

No.	**HIR1**
Statement	The machine running the program shall have a GPU on the list of compatible Nvidia GPUs
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**HIR2**
Statement	The machine running the program shall have an internet connection
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

3.3 Software Interfaces

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

No.	**SIR2**
Statement	The Nvidia Nemo Model(s) shall accept a stream of audio in wav format 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	

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	

No.	**SIR4**
Statement	The Website shall accept a list of changes and display those changes to the user
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	

Revision History	
Where to Implement	

No.	**SIR 5**
Statement	The website shall be able to display both google and sectional maps.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**SIR 6**
Statement	The website shall be able to display a variety of scales for the maps.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

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

No.	**SIR 8**
Statement	The program will be able to display selected aircraft tracking information similar to FlightRadar24/
Source	https://www.flightradar24.com/SWA2335/2e0fc599
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**SIR 9**
Statement	The program shall be able to display a textbox.
Source	

Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**SIR 10**
Statement	The program shall be able to allow the user to move the textbox.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

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	
Revision History	
Where to Implement	

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	
Revision History	
Where to Implement	

No.	**SIR 13**
Statement	The planes shall change position to new coordinates when new information is fetched from the plane API every 10 seconds.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

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	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

No.	**SIR 15**
Statement	The information panel shall update the live transcription every 5 seconds when it is visible.
Source	
Dependency	
Conflicts	
Supporting Material	
Evaluation Method	
Revision History	
Where to Implement	

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

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

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

4.1.3 Functional Requirements

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

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

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

4.2.3 Functional Requirements

2. 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.
3. 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. The planes on the interactive map will stay at the actual coordinates that they belong to when the map is moved.
5. 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

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

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

6. The user shall be able to click on any aircraft icon on the map to display a popup with flight information for that flight.
7. 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

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

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

8. The user shall be able to view the live transcription of ATC communications for any aircraft that they click the icon for.
9. The live transcription of ATC communications shall be updated once per every 3 minutes.

4.5 Toggle between Interactive and Detailed Map

4.5.1 Description and Priority

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

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

4.5.3 Functional Requirements

10. 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

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

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

11. The user shall press the plus button to zoom in on the interactive map.
12. The user shall press the minus button to zoom out on the interactive map.
13. The user shall click and drag on the map in any direction to make the interactive map move in that same direction.
14. The user shall press the plus button to zoom in on the VFR map.
15. The user shall press the minus button to zoom out on the VFR map.
16. The user shall click and drag on the map in any direction to make the VFR 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 revent 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 20%.

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

6. 5.5.2 ASR Team: Train models using NeMoOther 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.
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
TTS	Text-to-Speech
VFR	(Visual Flight Rules) A set of regulations that an aircraft can use to operate under clear, sunny weather conditions.

Table 1: *Common definitions for acronyms found throughout the document.*

Appendix B: Analysis Models

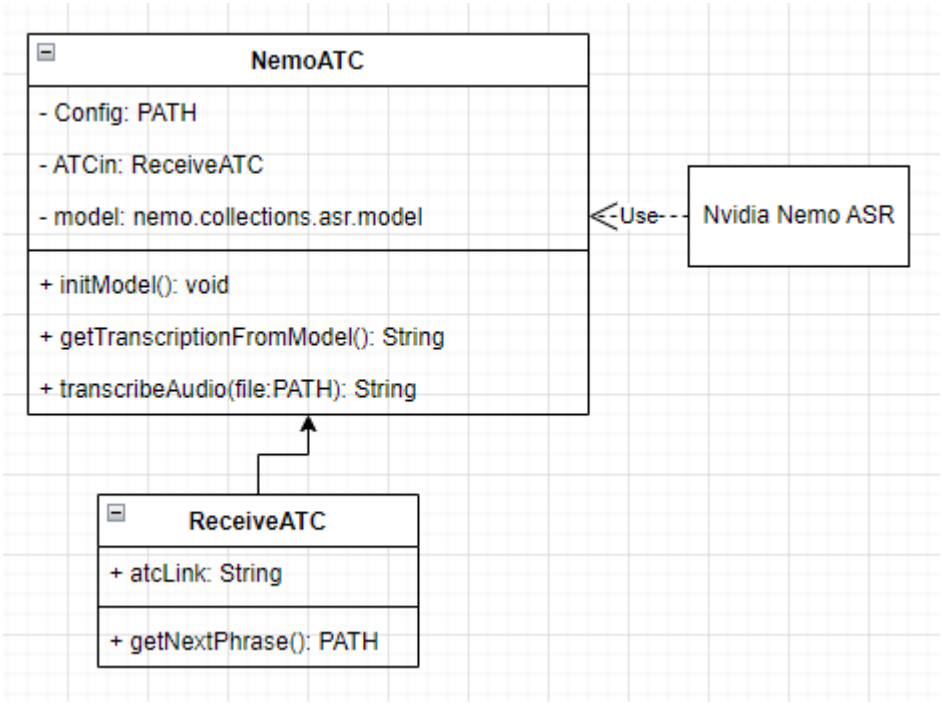

Figure 2: NeMo Neural Model Class Diagram

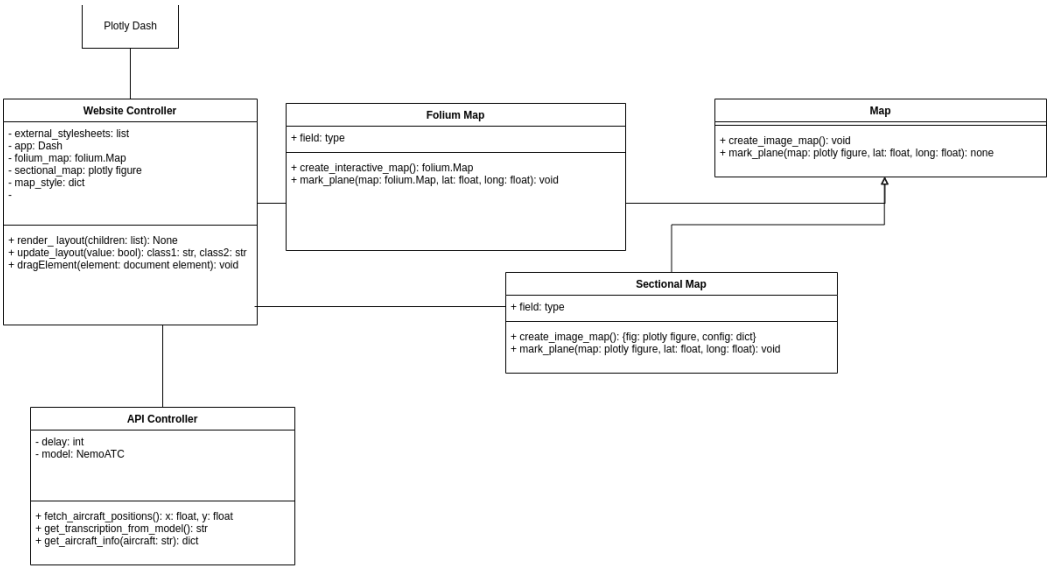

Figure 3: Website Class Diagram

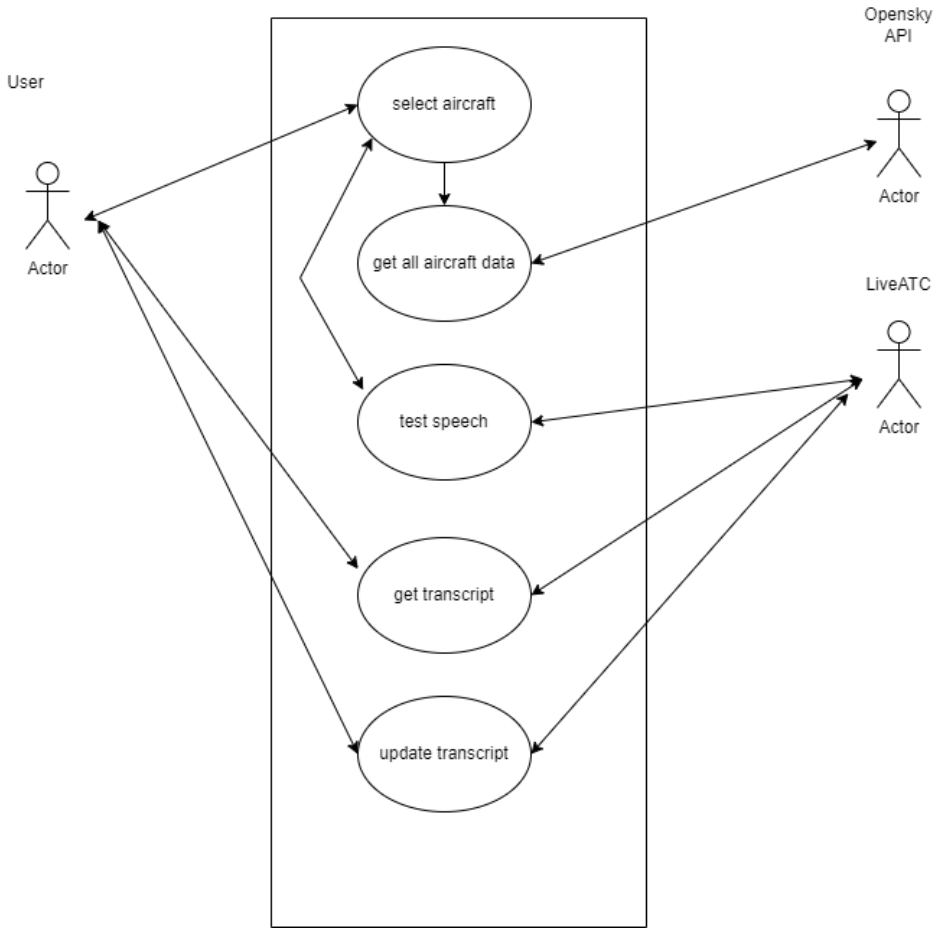

Figure 4: Use case diagram