ClearPath
System Requirements Specification
Version <3.5>
02/4/2025

Document Control

Distribution List

The following list of people will receive a copy of this document every time a new version of this document becomes available:

Teaching assistants:

Rajagopal Sugumar

Customer(s):

Dr. Masood Towhidnejad

Project team members:

Michael Yun

Cannon Newbury

Sadeed Khan

Isaac Hewitt

Change Summary

The following table details changes made between versions of this document:

Version	Date	Modifier	Description
1.0	10/29/2024	Full Team	Initial Copy of the SRS document
2.0	11/24/2024	Full Team	Update for the end of first semester
2.1	11/26/2024	Sadeed Khan	Updated sections 1-5 in accordance with comments provided by TA
2.2	11/26/2024	Michael Yun	Completed all diagrams associated with SRS v2
2.2.1	12/2/2024	Michael Yun	Updating Sections 2.2 – 2.7
2.2.2	12/2/2024	Cannon Newbury	Formatting before Submission of V2
3.0	1/14/2025	Cannon Newbury	Addition of Section 8 and 9
3.1	1/16/2025	Sadeed Khan	Starting section 8 with economic impact
3.2	1/21/2025	Full Team	Finished Section 8 and revised reqs.
3.3	1/23/2025	Sadeed Khan, Cannon Newbury	Added changes to the requirements and added justifications for changes.
3.4	2/2/2025	Sadeed Khan, Cannon Newbury, Michael Yun	Resolved comments, adjusted diagrams, create class model
3.5	2/4/2025	Sadeed Khan	Final touches for turning in

Table of Contents

1. Introduction

1.1. Purpose and Scope

The purpose of the SRS is to outline the requirements and tests needed for the ClearPath System. This SRS provides a comprehensive overview of the project, which aims to integrate live traffic data into the XPlane simulator in LB131. The integration will feature live traffic displayed in both the 2D map view and the 3D simulation environment, mirroring real-time traffic data. Additionally, we will develop/implement a plugin that enables users to initiate incursion scenarios, customizable by the number of planes and the location of the incursion. These features define the primary scope of the project, and any further customizations, such as additional 3D objects or bots, are excluded from this scope.

Thus far, the simulator has been set up, and we have verified that the controls are functional. We have completed the implementation of live traffic data via plug-ins and crowd-sourced databases through ADSB-Fi and OpenSky. We are now working on developing the incursion scenario plug-in. We are going to develop the plug-in through another plug-in called FlyWithLua. This allows us to code and implement the plug-in the way we intended.

1.2. Intended Audience and Reading Suggestions

This document is intended for the customer and any teachers' assistants that would like to know the scope or tests we plan to run on ClearPath. The rest of the document contains our requirements, diagrams, and tests.

1.3. Document Conventions

To ensure clarity and consistency throughout the document, all section headers are formatted in bold, while figure numbers and captions are italicized. This formatting standard aids readability and maintains uniformity.

1.4. Project References

We have started to use this GitHub (https://github.com/TwinFan/LiveTraffic) as a reference for our project. We have worked to implement this plug-in into our system, but we adapted it to our needs. We specifically had the traffic limited to the nearest 50 planes as outlined in the rest of this document. This is because we still want the user to be able to fly the simulator easily and not experience frame drops because of the traffic plug-in.

1.5. Definitions, Acronyms, and Abbreviations

Definitions

This section lists terms used in this document and their associated definitions.

Table 1:

Term	Definition
Taxiway	A designated path at an airport that connects runways with terminals, parking areas, and other facilities. Used for aircraft to taxi to and from runways.
Runway	A defined rectangular area at an airport used for the takeoff and landing of aircraft.
Incursion	An incident where an unauthorized aircraft, vehicle, or person enters an active runway or taxiway, potentially creating a collision hazard.
Live Traffic	Real-time data representing the movement of aircraft, including position, altitude, speed, and heading, often sourced from ADS-B systems.
ADS-B (Automatic Dependent Surveillance–Broadcast)	Surveillance technology in which an aircraft determines its position via satellite navigation and periodically broadcasts it, enabling real-time tracking by air traffic controllers and other aircraft.
Simulation	The process of replicating real-world scenarios in a controlled virtual environment, often for training, testing, or research purposes.
Scenario	A predefined set of conditions or events created in the simulation to train users or evaluate specific behaviors or responses.
Proximity Threshold	A system-defined limit representing the minimum allowable distance between objects (e.g., aircraft) before triggering an alert or event.
OpenSky Network	A collaborative platform and API providing real-time and historical aircraft tracking data derived from ADS-B and other systems.
Force Dynamics 401CR	A motion simulator used for recreating realistic flight movements by synchronizing physical motion with virtual aircraft behavior in the simulation.
Event Logging	The automatic recording of significant events, such as incursions, system alerts, or user actions, for later review and analysis.
Synchronization	The alignment of motion data between the XPlane simulator and the Force Dynamics 401CR to ensure realistic physical responses during a simulation.
Data Flow Diagram (DFD)	A graphical representation of data movement within a system, illustrating how data flows between components, processes, and external entities.
XPlane	A flight simulation software developed by Laminar Research, used as the foundation for the ClearPath simulation environment.
Visualization	The graphical representation of data, such as aircraft movements or simulation metrics, for user interaction and monitoring purposes.
User	A user role in the ClearPath system with elevated privileges for managing settings, retrieving logs, and maintaining system functionality.
API (Application Programming Interface)	A set of protocols and tools for building software applications, enabling ClearPath to interact with external systems like the OpenSky Network.

Acronyms

This section lists the acronyms used in this document and their associated definitions.

Table 2:

Term	Definition

FD401CR	Force Dynamics 401CR – The hydraulic motion simulator used for realistic training.
ATC	Air Traffic Control – A service provided to guide and manage aircraft movements.
ADS-B	Automatic Dependent Surveillance-Broadcast – A system that broadcasts real-time aircraft data.
API	Application Programming Interface – A set of rules and protocols for interacting with external systems or data.
TCP/IP	Transmission Control Protocol/Internet Protocol – A suite of protocols for internet communication.
HTTPS	Hypertext Transfer Protocol Secure – A protocol for secure communication over a network.
XPL	XPlane – A flight simulation software used in the ClearPath system.
OSKN	OpenSky Network – A data source providing real-time flight information.
3D	Three-Dimensional – Representing data in three spatial dimensions for visualization.

Abbreviations

This section lists the abbreviations used in this document and their associated definitions.

Table 3:

Term	**Definition**
Sim	Simulator – Refers to the XPlane or Force Dynamics 401CR simulation setup.
API	Application Programming Interface – A protocol for accessing external systems.
TA	Teaching Assistant – Supports the project team with academic oversight.
Inc	Incursion – Refers to runway or taxiway incursions simulated in the system.
DB	Database – Stores simulation logs and event data.
UI	User Interface – Visual components of the simulation system for user interaction.
Alt	Altitude – Refers to the height of aircraft relative to sea level.
Lat	Latitude – The geographic coordinate indicating north-south position.
Lon	Longitude – The geographic coordinate indicating east-west position.
OS	Operating System – The software environment hosting the ClearPath system.

2. General Description

2.1. Product Perspective

ClearPath aims to develop a simulation system that integrates with the existing XPlane simulator in LB 131 to model and analyze potential runway and taxiway incursions. This system should:

1. Utilize the physical simulator as a simulated aircraft
2. Navigate the simulated aircraft through runways and taxiways
3. Create scenarios that demonstrate potential runway or taxiway incursions
4. Provide a platform for studying and preventing such incursions in real-world situations

ClearPath aims to enhance the way ATC and pilots communicate during runway and taxiway incursions.

2.2. Product Features

ClearPath is an advanced simulation system designed to integrate live aircraft traffic into the XPlane 11 flight simulation environment. It leverages real-time data sources like the OpenSky Network API and synchronizes with the Force Dynamics 401CR motion simulator to provide realistic training and testing scenarios. The system is focused on simulating and analyzing runway and taxiway incursions to enhance safety and operational protocols. The key goals of ClearPath are to develop an accurate, responsive, and user-friendly system for aviation professionals and researchers.

Key Features:

1. Real-time integration of live traffic data.
    a. Integrates real-time aircraft movement data into the simulation environment, including variables such as position, speed, altitude, and flight plans.
    b. This feature ensures the simulation reflects current traffic conditions, enabling realistic testing and training scenarios.
    c. The integration relies on data sources such as the OpenSky Network API and ADS-B feeds.
2. Runway Incursion Scenarios
    a. Simulates situations where aircraft, vehicles, or personnel enter active runways or taxiways.
    b. Designed to replicate real-world incidents, allowing for training and the development of mitigation strategies to enhance safety.
3. Motion Synchronization
    a. Synchronizes the physical motion of the Force Dynamics 401CR simulator with the simulation's aircraft movement, providing a realistic training environment.

2.3. User Classes and Characteristics

ClearPath supports multiple user classes, each interacting with the system in distinct ways. These user classes include operators, developers, and stakeholders involved in simulation, development, and analysis activities.

2.3.1 Actors

This section presents the actors in the system.

1. User
    a. The User is responsible for interacting with the simulation system by starting or stopping simulations, adjusting settings, and viewing real-time data during operational sessions.
2. OpenSky Network API
    a. The OpenSky Network API serves as a data provider, supplying real-time live traffic data that integrates seamlessly into the simulation environment.
3. Force Dynamics 401CR
    a. The Force Dynamics 401CR motion simulator ensures that aircraft movement in the simulation is synchronized with physical motion, enhancing realism.

2.3.2 Use Cases and Scenarios

Start Simulation

This use case allows the User to initialize the simulation environment. It starts all required subsystems, including live traffic data integration and motion sync.

Steps**:**

1. The User clicks the 'Start Simulation' button.
2. The system fetches live traffic data from the OpenSky Network API.
3. Motion sync with the Force Dynamics 401CR is initiated.

Scenario**:**

Precondition: The system is powered on, and the simulator setup is complete.

Trigger Condition: The User clicks the 'Start Simulation' button.

Steps:

1. The User initiates the simulation.
2. The system establishes connections with the required components.

Postcondition: The simulation environment is running with live traffic data and synchronized motion.

Stop Simulation

This use case allows the User to end a simulation session. It stops live traffic updates and halts motion sync

Steps:

4. The User clicks the 'Stop Simulation' button.
5. The system halts all live data integrations and motion synchronization.

Scenario:

Precondition: A simulation session is currently running.

Trigger Condition: The User clicks the 'Stop Simulation' button.

Steps:

1. The User ends the simulation session.
2. The system terminates all active connections.

Adjust Settings

This use case allows the User to modify simulation parameters such as proximity thresholds for incursion detection.

Steps:

6. The User navigates to the settings menu.
7. The User adjusts parameters (e.g., proximity threshold) via sliders or text boxes.
8. The system validates and applies the new settings.

Scenario:

Precondition: The system is powered on and accessible.

Trigger Condition: The User opens the settings menu.

Steps:

1. The User modifies simulation parameters.
2. The system validates the inputs and updates the parameters.

Postcondition: The updated settings are applied to the simulation.

View Real-Time Data

Displays live aircraft positions, headings, and incursion alerts. The User monitors the simulation and responds as needed.

Steps:

9. The system fetches live traffic updates and processes the data.
10. The User views aircraft data on the interface.
11. The system alerts the User to any potential runway incursions.

Scenario:

Precondition: The simulation is running.

Trigger Condition: The User monitors real-time data.

Steps:

1. The User observes live updates and responds to alerts.

2. The system logs the observed data and alerts.

Postcondition: The User has monitored real-time data and responded as necessary.

Integrate Live Traffic Data

The system fetches live aircraft data, processes it, and integrates it into the simulation.

Steps:

12. The system sends a request to the OpenSky Network API for live data.
13. The API responds with live traffic data.
14. The system processes and injects the data into XPlane 11.

Scenario:

Precondition: The OpenSky Network API is accessible.

Trigger Condition: The system initiates a request for live traffic data.

Steps:

1. The system retrieves and processes the data.

2. The data is visualized within the simulation environment.

Postcondition: Live traffic data is successfully integrated into the simulation.

Sync Motion Data

Synchronizes the simulation’s aircraft movement with the Force Dynamics 401CR motion simulator for a realistic experience.

Steps:

15. The system establishes a connection with the motion simulator.
16. Aircraft position and movement data are sent to the simulator.
17. The Force Dynamics 401CR mimics aircraft motion in real-time.

Scenario:

Precondition: The motion simulator is powered on and connected to the system.

Trigger Condition: The simulation begins, and motion synchronization is initiated.

Steps:

1. The system sends movement data to the motion simulator.
2. The simulator mirrors the motion in real time.

Postcondition: Aircraft motion data is synchronized with the Force Dynamics 401CR.

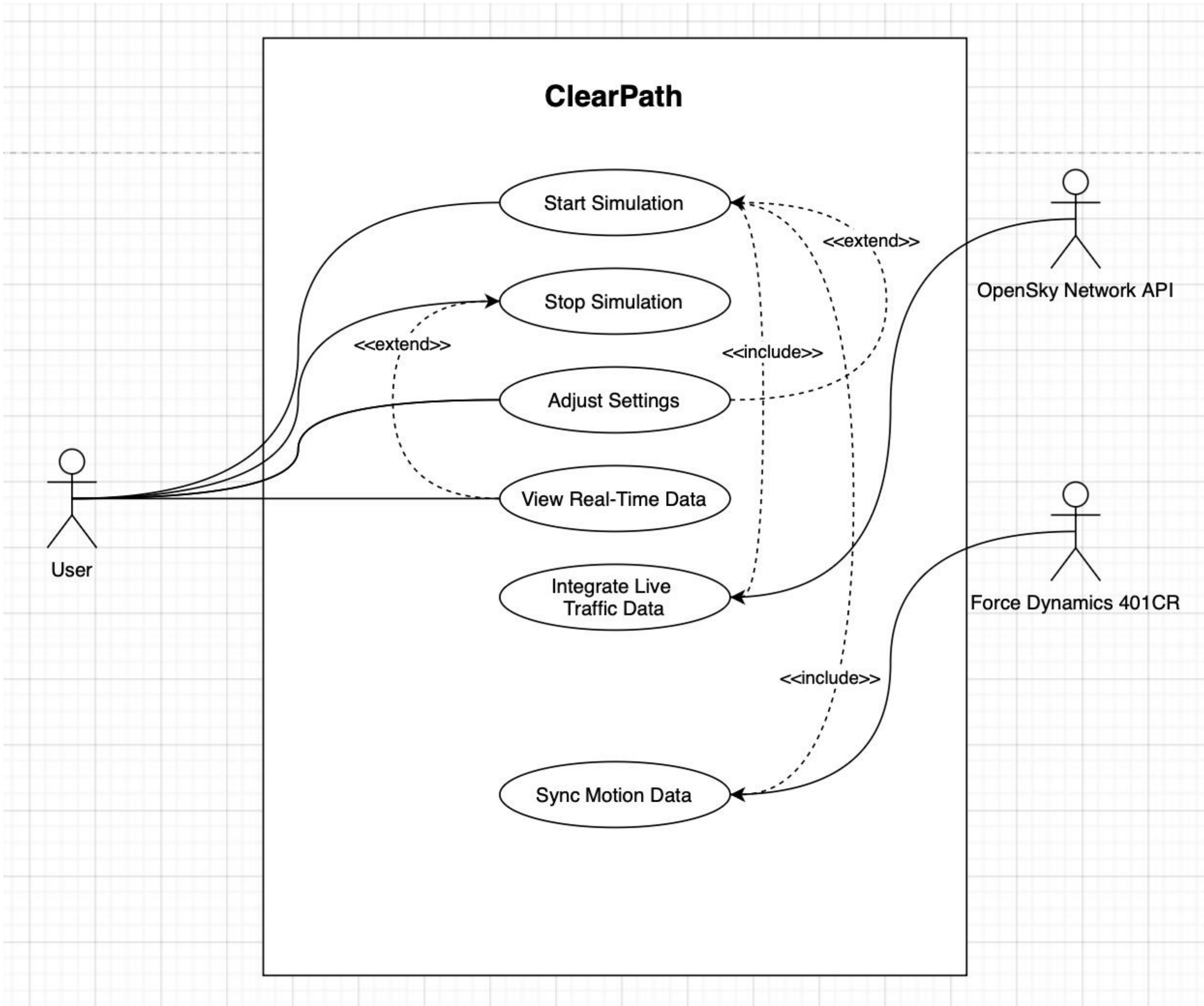


Figure 1 Use Case Diagram

2.4. General Constraints

The project faces several operational constraints that impact development and implementation. One significant constraint is the reliance on the physical simulator located on campus, which necessitates in-person work in the Lehman building. This limitation restricts the ability to work on the project remotely and imposes logistical challenges for team collaboration.

Another constraint is the workload associated with the project. The requirement for continuous documentation, combined with the demands of other academic courses, places a significant burden on team members. Additionally, commitments to AFROTC responsibilities further limit the time and energy available for project-related tasks, creating challenges in maintaining consistent progress.

2.5. Operating Environment

ClearPath is intended to be used as an application for the indoor flight simulator within LB131. It may be used either with full motion or a stationery set-up. Within the simulated environment, we are focused on runways and taxiways for incursion scenarios. The live traffic should be displayed throughout the simulated environment.

2.6. User Documentation

- ClearPath execution files and ReadMe document
    - ReadMe will provide insight on installation and XPlane interaction

2.7. Assumptions and Dependencies

The development team has made the following assumptions:

- ClearPath will be used in conjunction with XPlane software
- Live OpenSky Network readings can be read and injected into XPlane
- There are no OS conflictions and across Linux, Windows and Apple products
- Hardware plugins do not conflict with ClearPath plugin

3. External Interface Requirements

3.1. User Interfaces

3.1.1. Flight Simulation User Interface

Characteristics:

- The interface shall provide a visual display of real-time flight data, including flight paths, altitudes, and aircraft types.
- The user interface shall consist of panels that can be customized to show relevant information, including maps, flight status, and alerts.

Requirements:

- [REQ-1] The system shall display real-time flight data on a dedicated screen section in a clear and legible format.
- [REQ-2] The user shall be able to toggle between different views of the flight data. These views are the list view, and map view. These are accessed through the “Live Data” menu.

- [REQ-3] When the flight data panel is clicked the user shall be able to see the closest 50 planes in the area, and flight data information such as speed, altitude, heading, and tail number,
- [REQ-4] When the map is clicked the closest 50 planes shall be displayed with callsigns or identifiers.
- [REQ-5] The system shall provide visual alerts via a pop-up message for runway incursions.

3.1.2. Incursion Scenario Selection Interface

Characteristics:

- The interface shall allow users to set parameters for the incursion scenario.
- Parameters include, number of planes, location, etc.
- Users shall be able to select difficulty levels and types of scenarios.

Requirements:

- [REQ-6] The system shall provide options for the number of planes and location of incursion.
- [REQ-7] The User shall receive feedback on incursions through visual and audio alerts.

3.2. Hardware Interfaces

Characteristics of Hardware Components

Baseplate Power Switch: A white power switch that activates the motion control system.

Requirements:

- [REQ-8] The power switch shall be an easily accessible toggle switch.
- [REQ-9] The motion control system shall use standard connectors (RJ45 for network, USB for peripheral connections).

Connector Specifications

- Physical Connectors:
  - RJ45 connectors for Ethernet communication.
  - USB connectors for peripheral devices.
- Communication Protocols:
  - Ethernet for network communication.
  - I2C for communication between internal hardware components.
- Voltage Ranges:
  - The power supply must provide a voltage range of 12-24V to the motion control system.

3.3. Software Interfaces

Required Software Products

- Name: X-Plane 11
    - Mnemonic: XP11
    - Specification Number: SRS-XP11-001
    - Version Number: 11.0
    - Source: Laminar Research

- Name: OpenSky Network API
    - Mnemonic: OSKN
    - Specification Number: SRS-OSKN-002
    - Version Number: 1.0
    - Source: OpenSky Network

Requirements:

- [REQ-10] The system shall interface with the OpenSky Network API to receive real-time flight data.
- [REQ-11] The system shall ensure that data is inputted through the OpenSky Network APU and integrated into the X-Plane 11 environment.

3.4. Communications Interfaces

Local Network Protocols

Requirements:

- [REQ-12] The system shall use TCP/IP protocol for communication between the OpenSky Network API and the simulation environment.
- [REQ-13] The system shall implement a secure connection (HTTPS) when accessing the OpenSky Network API to ensure data integrity and security.

Summary

This section outlines the external interface requirements necessary for merging real-time flight data into X-Plane 11. Each interface requirement is designed to ensure seamless integration and user experience during training simulations involving runway incursions and bot interactions.

4. Behavioral Requirements

4.1. User Classes and Access Levels

- [REQ-14] The ClearPath System shall provide one User class
    - **User:** Access to simulation controls and incursion scenario playback. Access to all system controls, including live data integration, scenario editing, and system configurations.
- [REQ-15] All users should have access to the program fully, no user should be restricted.

4.2. Related Real-world Objects

- [REQ-16] The system shall model the following real-world objects:
    - **Aircraft**: Simulated aircraft that respond to real-time flight data and can be observed in 3D space.
    - **Taxiways and Runways**: Virtual representations of taxiways and runways where incursion scenarios will occur.
    - **ATC (Air Traffic Control):** Represents virtual ATC controls that trigger and monitor incursion scenarios.
- [REQ-17] Each aircraft object shall have attributes such as position (latitude, longitude, altitude), velocity, and heading, which are updated based on real-time data from the live traffic feed.
- [REQ-18] The incursion detection functionality shall identify when two aircraft objects come within a specified proximity on a taxiway and trigger an alert.

4.3. System Stimuli and Responses

- [REQ-19] When live traffic data is received from OpenSky Network, the system shall update the aircraft objects in real time to reflect their actual positions and movements.
- [REQ-20] Upon detection of a potential incursion on the taxiway:
    - **4.3.1**: The system shall alert Users of the incursion through a visible alert on the simulator screen.
- [REQ-21] In the event of a network interruption or data feed loss, the system shall:
    - **4.3.3**: Notify the User with an error message, while displaying the last known positions of aircraft.
    - **4.3.4**: Attempt to reconnect to the data source every 30 seconds until the connection is restored.

4.4. Scenario-Specific Requirements

- [REQ-22] For each taxiway incursion scenario, the system shall allow the User to configure variables, such as aircraft speed, proximity threshold, and alert type.

- [REQ-23] The 3D simulation shall mirror real-time aircraft positions and incursion scenarios as they occur, with visual indicators (e.g., flashing red for potential incursions).

4.5. Related Features

4.5.1 Live Traffic Integration

- [REQ-24] The system shall connect to the OpenSky Network and ADSB-Fi APi's, to integrate real-time aircraft positions into the simulation.
- [REQ-25] When live traffic data is received, the system shall update aircraft positions and headings in the simulation environment to match real-time movements.

4.5.2 3D Aircraft Behavior Visualization

- [REQ-26] The system shall display aircraft in three dimensions, accurately reflecting altitude, speed, and heading.
- [REQ-27] When a new aircraft enters the simulation or an existing one changes position, the system shall render it in 3D, updating position and heading in real time.

4.5.3 Incursion Detection and Alert System

- [REQ-28] The system shall detect potential incursions on runways and taxiways, triggering alerts to the user for intervention.
- [REQ-29] When two aircraft approach within a defined proximity threshold on a taxiway, the system shall trigger a visual alert on the simulator screen (example: stop the plane, turn off the taxiway, allow traffic to pass).

4.6. Functional

4.6.1 Live Data Processing and Validation

- [REQ-30] The system shall validate live traffic data to ensure completeness and accuracy before updating the simulation, checking for data anomalies such as missing coordinates or incorrect altitude values.
- [REQ-31] If data validation identifies anomalies, the system shall revert to the last valid data state.

4.6.2 Aircraft Position and Movement Updates

- [REQ-32] The system shall process incoming data to update each aircraft's position, speed, and heading in the simulator, recalculating and rendering new coordinates as received.
- [REQ-33] The system shall ensure smooth aircraft movement by interpolating position data to avoid abrupt transitions.

4.6.3 Incursion Detection Logic

- [REQ-34] The system shall calculate proximity between aircraft on taxiways and runways.
- [REQ-34.1] If two aircraft come within a specified threshold, the system will recognize this as a potential incursion.
- [REQ-35] When an incursion is detected, the system shall display a warning to the user.

4.6.4 Error Handling and Recovery

- [REQ-36] The system shall handle overflow errors during data processing by queuing excess data and processing it sequentially.
- [REQ-37] If a connection to the live data source is lost, the system shall notify the user and continue displaying the last known aircraft positions.
- [REQ-39] The system shall attempt to reconnect to the live data source every 30 seconds, notifying the User upon successful reconnection.

4.6.5 Data Flow and Display Defaults

- [REQ-40] The system shall display default aircraft and runway positions based on configuration files, allowing the user to adjust settings as needed.

5. Non-behavioral Requirements

5.1. Performance Requirements

- [REQ-42] The system shall support one user interacting with the simulation controls.
- [REQ-43] The system shall process real-time updates of aircraft position, speed, and heading every one second, ensuring smooth transitions within the simulator.
- [REQ-44] During peak operations, the system shall handle up to 50 aircraft in the simulated environment without exceeding a 2-second delay in updates.
- [REQ-45] 95% of simulated incursion scenarios shall trigger alerts to the user in less than 0.5 seconds upon detection.

5.2. Safety Requirements

- [REQ-48] In case of a software crash or data feed loss, the system shall revert to a safe state displaying the last known positions of all aircraft to avoid misinterpretation by users.

5.3. Qualitative Requirements

Availability

- [REQ-49] The system shall be operational with 100% availability when XPlane is started.
    - **5.3.1**: If the system is unavailable, a message should be displayed to notify the user.

Security

- [REQ-51] Access to simulation controls and incursion scenario data shall be logged with timestamps, user roles, and actions performed for security auditing.

Maintainability

- [REQ-53] The software shall be designed with modular components for simulation controls, data integration, and incursion detection to simplify future maintenance and updates.
- [REQ-54] Each module shall contain documentation for functionality, parameters, and dependencies, ensuring that new developers can make updates with minimal onboarding.

Portability

- [REQ-55] The system software shall be implemented in a cross-platform compatible language (e.g., Python or Java) to support potential deployment on other simulators or training environments.

5.4. Design and Implementation Constraints

- [REQ-57] The system shall comply with the existing XPlane simulator hardware and software setup without requiring additional hardware modifications.
- [REQ-58] The software shall be compatible with OpenSky Network API standards, ensuring seamless data integration for real-time traffic.

6. Other Requirements

6.1. Database Requirements

- [REQ-60] The system shall store live event data in a database, including:
  - ID: Unique identifier for each Plane.
  - Registration: the tail number registered to the aircraft.
  - Type: The type of aircraft.
  - Model: Specific model of aircraft
  - Callsign: Any alternative callsign used to identify the aircraft
  - Altitude
  - KN: Air Speed in knots
  - Heading

6.2. Operations

- [REQ-65] The system shall support the following operational modes:
  - Interactive Mode: During user-initiated sessions, where simulation and data processing occur in real-time based on user interactions.
    [REQ-65.1] The system shall require the user to be in the simulator chair.
    [REQ-65.2] The system shall require for the safety features, being a seatbelt and door, be buckled and closed respectively.
    [REQ-65.3] The system shall require the user to enable motion controls for the simulation.
    [REQ-65.4] The system shall require the user to start XPlane 11.
    [REQ-65.5] The system shall then start using motion to simulate the plane in-game position.

- [[REQ-68] In the event of an unplanned shutdown or restart, the system shall restore the last known database state and resume operation from the last checkpoint to prevent data loss.

7. Analysis Models

7.1. Data Flow Model

7.1.1 Data sources

The data sources and their inputs to the system identified in the data flow model are as follows:

- OpenSky Network API:
    - Provides real-time aircraft data, including:
        - Time Data: Timestamps of aircraft positions.
        - Position Data: Latitude, longitude, and altitude of planes.
        - Speed and Heading Data: Current speed and direction of aircraft in flight.
- User Input:
    - Commands and settings input by the user, including:
        - Start/Stop Simulation: Initiates or terminates the simulation session.
        - Proximity Threshold Adjustment: Allows the user to set or modify thresholds for incursion detection.
        - Authentication Details: For accessing historical logs.

7.1.2. Data Sinks

The data sinks and their outputs identified in the data flow model

- End User:
    - Outputs provided to the user include:
        - Real-time Alerts: Notifications of proximity breaches or other simulation events.
        - Simulation Data: Visual representation of aircraft positions, headings, and movement.
        - Historical Logs: Access to stored data on incursions, errors, and system interactions.

7.2. Context Diagram (Level 0 Data Flow Diagram)

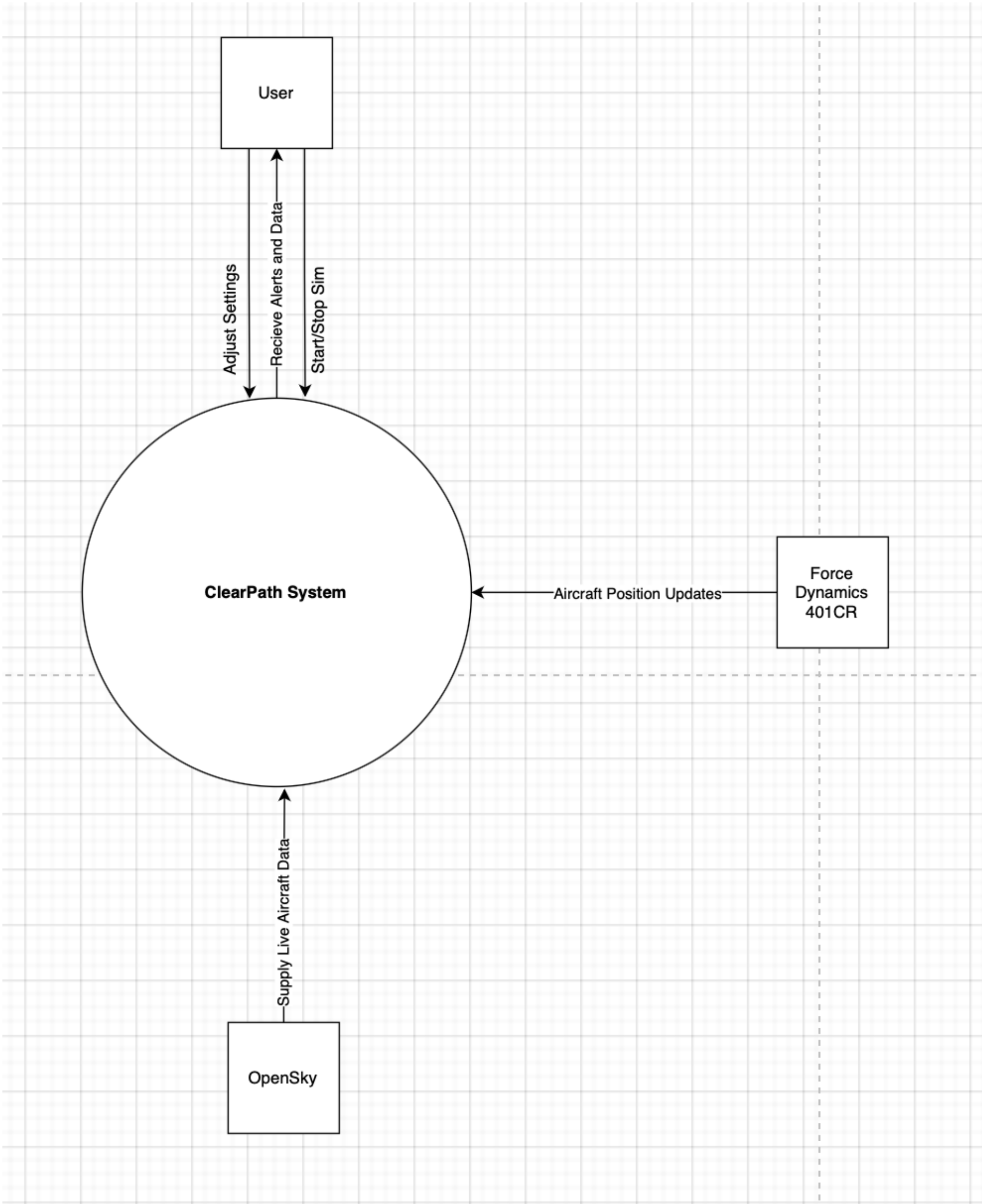


Figure 2 - Context Diagram

Level 1 Data Flow Diagram

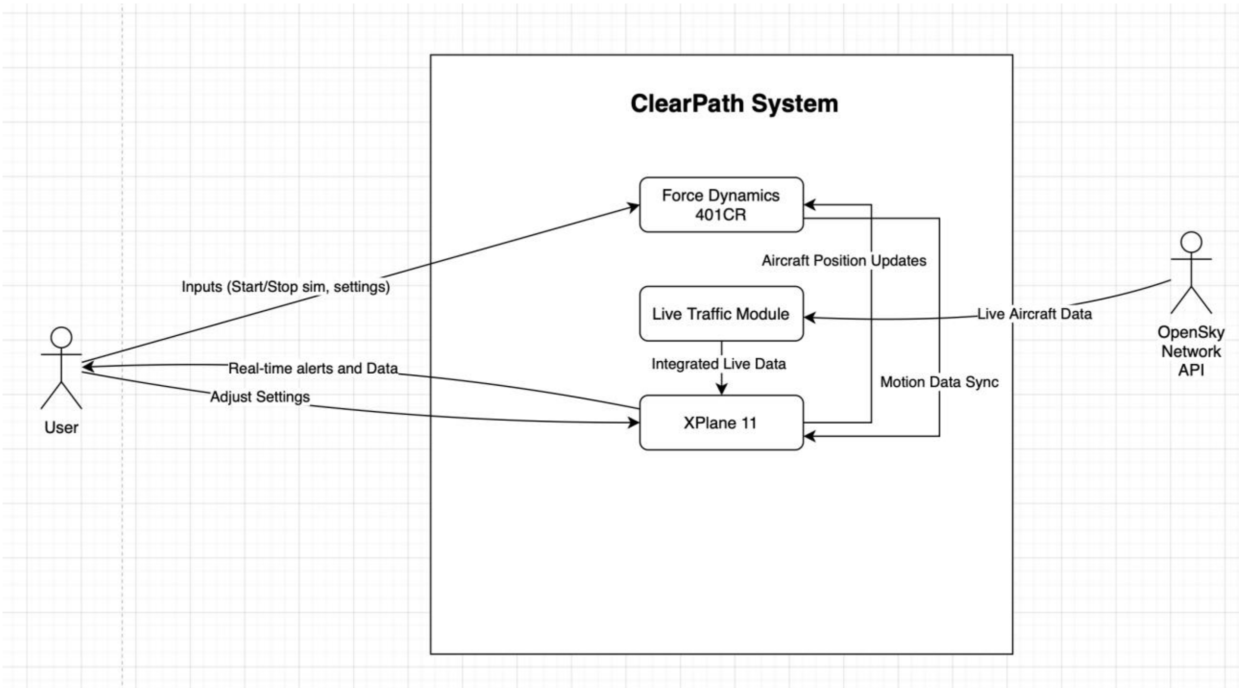


Figure 3 Level 1 Data Flow Diagram

Level 2 Data Flow Diagrams

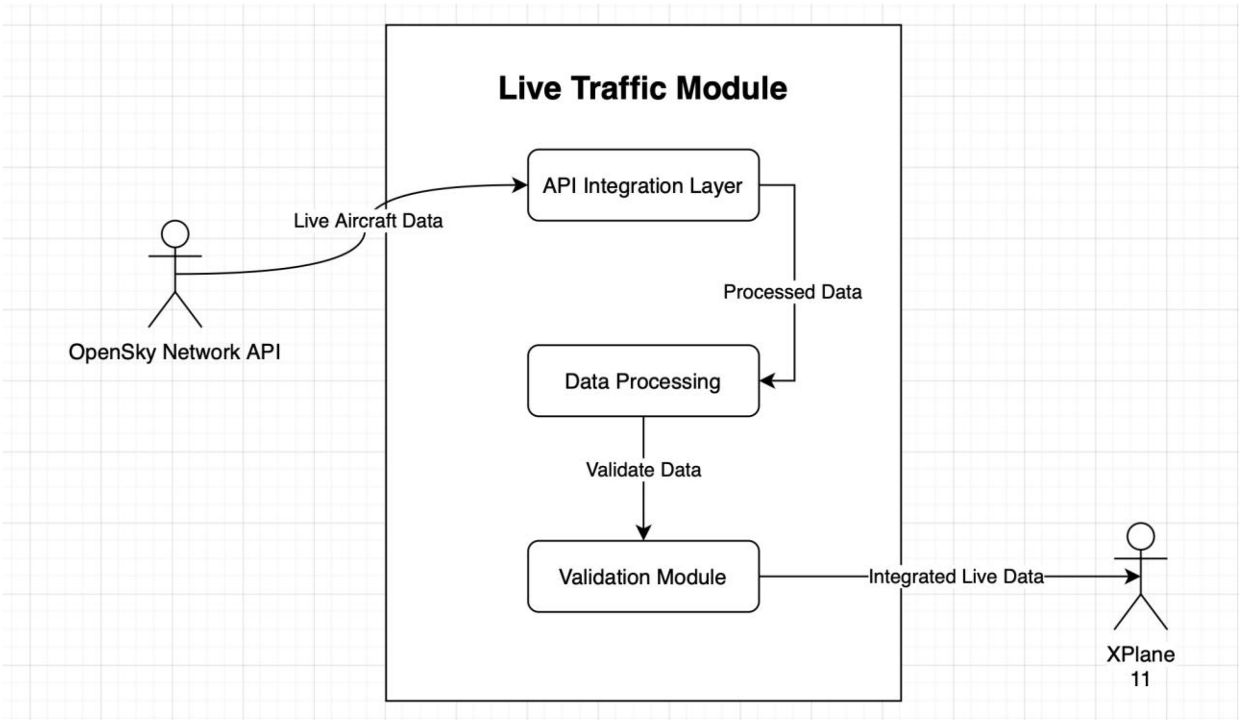
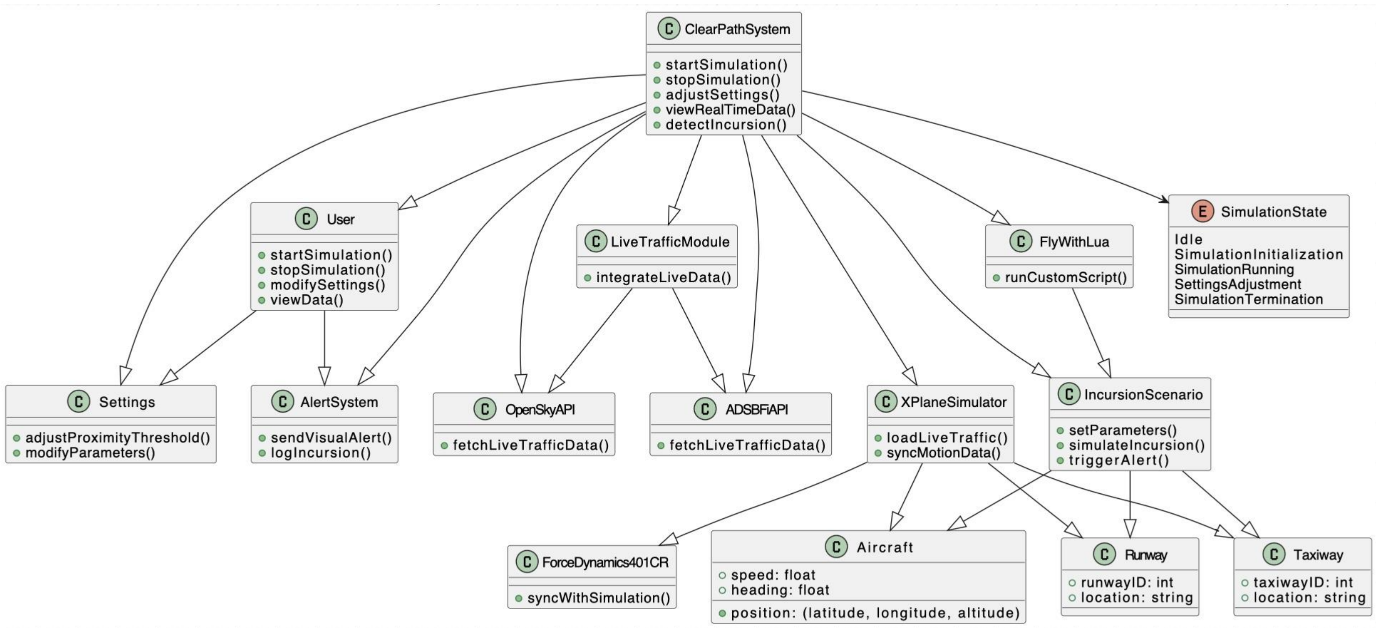


Figure 4 Live Traffic Module Data Flow Diagram

The Force Dynamics 401CR and XPlane 11 Level 2 Data Flow Diagrams (DFDs) are not included in this documentation as these modules were not designed or developed by the ClearPath team. They are external systems integrated into the ClearPath project and operate as off-the-shelf components.

7.3. Class Model



The ClearPath System class model is designed at a high level due to the integration of multiple external plugins and pre-existing software modules, such as XPlane 11, OpenSkyAPI, ADSB-Fi, and FlyWithLua. Rather than developing a system from scratch, ClearPath primarily acts as an integration framework that connects real-time air traffic data, motion synchronization, and incursion scenario scripting, written by the team, within the simulator environment. Because many of the components we use—such as XPlane's built-in functionalities and the LiveTraffic plugin—are third-party libraries with their own internal class structures, our class model does not attempt to detail the internal implementation of these systems. Instead, it focuses on defining ClearPath's interactions with these components and their roles in data flow, simulation control, and incursion detection.

By maintaining a high-level abstraction, our design ensures modularity. The class diagram emphasizes how data flows between system components rather than implementing intricate object class models that may be subject to change in the third-party plugins. For example, rather than modeling individual aircraft objects in detail, the system interacts with LiveTrafficModule, which fetches and processes live traffic from OpenSkyAPI and ADSB-Fi API. Similarly, incursion scenarios are managed through FlyWithLua with scripting done by the team. This level of abstraction allows us to focus on the integration portion rather than reverse-engineering the code and work of pre-existing software.

7.4. State Model

Below is our state model diagram. This diagram shows the different states of the program thus far. In the future we hope to add a new state or plugin that allows the user to add in runway incursion scenarios.

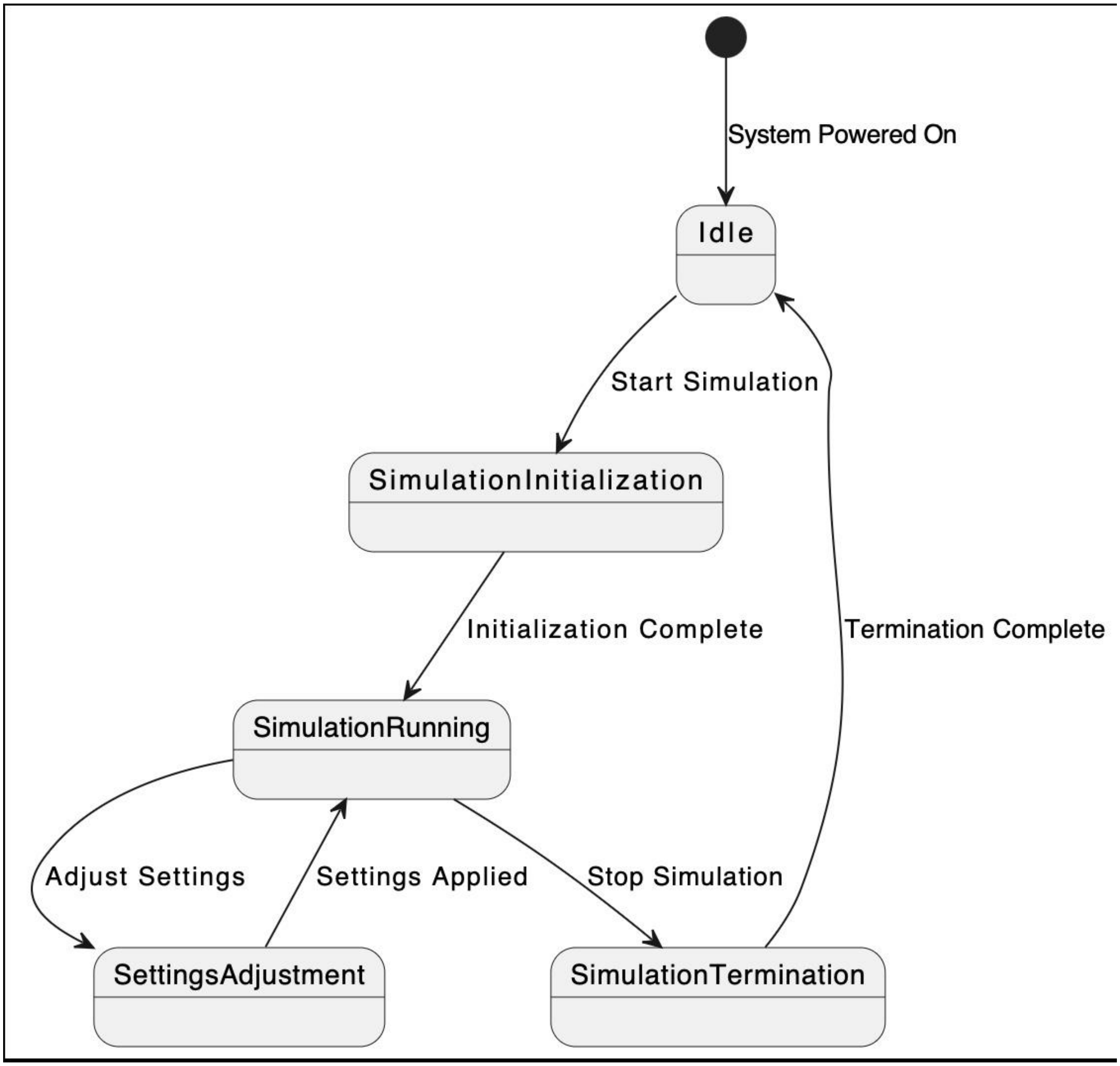


Figure 5 - State Model Diagram

8. Changes From SRS V1/2

- **[REQ-3]**: Updated live traffic count to nearest 50 planes. This makes sure the software is still usable, and performance does not drop.
- **[REQ-4]**: Updated live traffic count to nearest 50 planes. This makes sure the software is still usable, and performance does not drop.
- **[REQ-11]**: in V2 specified **X-Plane 12**, but in V3, it was changed to **X-Plane 11**.
- **[REQ-14]**: Changed to reflect removal of Administrator class
- **[REQ-15]**: The access level breakdown was simplified, and the specific breakdown of Administrator and User roles was reduced.
- **[REQ-18]**: The requirement specifying an alert going to an "Administrator" was changed to a general alert instead.
- **[REQ-20]**: Removed mention of Administrator
- **[REQ-20 4.3.3]**: Removed mention of Administrator
- **[REQ-20 4.3.2]**: System will no longer store data and will not be able to store flights for review.
- **[REQ-22]**: Removed mention of Administrator
- **[REQ-28]**: The incursion alert was previously directed to the **Administrator**, but now it is directed to the **user**.
- [REQ-29]: Removed logging the incursion event and stipulation that the incursion warning disappears when user performs evasive action.
- [REQ-32]: The live traffic data updates every 1-3 minutes. Live traffic displayed in-game will still follow flight paths frame-by-frame in real time
- [REQ-33]: Information about logging has been removed.
- [REQ-41]: Removed; No appendix F, table is represented via information given by API. Information set in [REQ-60].
- **[REQ-46]**: The restriction on scenario modifications changed from **Administrator-only** to **general user restrictions**.
- [REQ-47]: All users with access to simulator have access to the ClearPath system. There shall be no further login or authentication required.
- [REQ-56]: Removed; the Only operating system that ClearPath is intended to run on is Windows on the system lactated in LB131.
- [REQ-59]: Removed; We are no longer mandating databases to be stored containing flight information because of the restriction from X-Plane11.
- [REQ-61]: Removed: No databasing is [REQ-59] The system shall store no more than 500 entries in the log for incursion detection events to prevent memory overload on the simulator hardware.
- **[REQ-62]**: Removed; We are no longer mandating databases to be stored containing flight information because of the restriction from X-Plane11.
- [REQ-63]: Removed; We are no longer mandating databases to be stored containing flight information because of the restriction from X-Plane11.
- [REQ-64]: Removed; We are no longer mandating databases to be stored containing flight information because of the restriction from X-Plane11.

- [REQ-66]: Removed; We are no longer mandating databases to be stored containing flight information because of the restriction from X-Plane11.
- [REQ-67]: Removed; We are no longer mandating databases to be stored containing flight information because of the restriction from X-Plane11.

  The changes to requirements [REQ-11], [REQ-15], [REQ-18], [REQ-28], [REQ-46], and [REQ-62], [REQ-63], [REQ-64], [REQ-66], [REQ-67] were made to streamline the system architecture and reduce the need for additional licensing and hardware. Implementing a new terminal for the Administrator to communicate with the user terminals would require significant resources, including additional software licenses and hardware. This terminal would also need to interact seamlessly with the user terminal, introducing added complexity in data transfer and insertion processes. Additionally, X-Plane does not maintain a continuous flight data log; it only logs events when a flight is created, crashes, or is saved. This limitation necessitated simplifying the access levels and alert system to make better use of the existing capabilities of X-Plane. We are no longer mandating databases to be stored containing flight information because of the restriction from X-Plane11. The program does not save this information to be able to be accessed and processed into a database. By reducing the distinction between Administrator and User roles, alerts and access are now more manageable, avoiding the process of integrating a dedicated Administrator terminal. These adjustments allow the system to remain functional.

- **[REQ-50]** (Encryption standard requirement) was **removed**.
- **[REQ-52]** (API key security restriction) was **removed**.
- **[REQ-65.6]** (Unattended mode) was **removed**.

The removal of [REQ-50] (Encryption Standard Requirement) was due to the system not handling sensitive data, making encryption unnecessary and potentially introducing performance overhead. [REQ-52] (API Key Security Restriction) has been removed because the OpenSky Network API provides public access, making API key management redundant or overly restrictive. [REQ 65.6] was eliminated to give the user control over the entire system. This prevents compatibility issues and avoid disruptions in a stable simulation environment. These removals simplify system design, optimize performance, and maintain administrative control.

9. Project Impacts

Economic Impact (Sadeed Khan):

From an economic perspective, ClearPath offers cost savings for flight schools and airlines. Without live traffic integration, pilots can train to avoid traffic based solely on visual identifiers. This means they do not have radio contact with other traffic. This may seem like a far-fetched idea; however, this is very much possible and ClearPath gives pilots a chance to see scenarios like this on the ground first. ClearPath's runway incursions can have a similar economic impact. With this low-cost option, students can train to be familiar with extraneous circumstances avoiding costly closures, delays, or cancellations when in an operational situation.

Social Impact (Michael Yun):

On the social front, ClearPath reinforces the paramount responsibility of pilots: ensuring passengers arrive at their destinations safely. By offering realistic and practical training scenarios, such as runway incursions and visual-only traffic navigation, ClearPath equips pilots to handle high-pressure situations effectively. This preparation enhances overall safety, reducing the likelihood of accidents or errors in live operations. ClearPath not only sharpens pilots' skills but also bolsters public confidence in air travel, emphasizing the critical importance of safety in every flight.

Global Impact (Cannon Newbury):

The ClearPath project has the potential to significantly enhance global air travel safety by addressing one of the most critical concerns in aviation: runway and taxiway incursions. By integrating live traffic data into simulators and creating realistic incursion scenarios, ClearPath could reduce incidents worldwide, promoting safer and more efficient operations. This innovation could set a new standard for ground traffic management, influencing international aviation protocols and practices.

ClearPath's principles and methodologies might also extend beyond aviation to other industries, such as maritime and railway traffic management. These sectors could adopt similar systems for traffic monitoring and simulation, further advancing global transportation safety. Additionally, international collaboration in developing and deploying ClearPath could foster stronger relationships between countries, encouraging shared advancements in technology and safety protocols.

Environmental Impact (Isaac Hewitt):

From an environmental point of view, ClearPath minimizes the ecological footprint of pilot training by reducing the need for fuel-intensive flight hours during early stages of skill development. By simulating complex scenarios on the ground, ClearPath helps flight schools and airlines lower carbon emissions associated with traditional training methods. Additionally, improved pilot proficiency leads to more efficient flight operations, reducing unnecessary fuel consumption and emissions during real-world flights. ClearPath supports a greener aviation industry by promoting sustainability without compromising the quality of pilot training.

10. Amendments