**ClearPath**

**System Requirements Specification**

**Version <2.0>**

**10/29/2024**

# 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 |

Table of Contents

[Document Control ii](#_Toc184044354)

[Distribution List ii](#_Toc184044355)

[Change Summary ii](#_Toc184044356)

[1. Introduction 1](#_Toc184044357)

[1.1. Purpose and Scope 1](#_Toc184044358)

[1.2. Intended Audience and Reading Suggestions 1](#_Toc184044359)

[1.3. Document Conventions 1](#_Toc184044360)

[1.4. Project References 1](#_Toc184044361)

[1.5. Definitions, Acronyms, and Abbreviations 1](#_Toc184044362)

[Definitions 1](#_Toc184044363)

[Acronyms 2](#_Toc184044364)

[Abbreviations 2](#_Toc184044365)

[2. General Description 3](#_Toc184044366)

[2.1. Product Perspective 3](#_Toc184044367)

[2.2. Product Features 3](#_Toc184044368)

[2.3. User Classes and Characteristics 4](#_Toc184044369)

[2.3.1 Actors 4](#_Toc184044370)

[2.3.2 Use Cases and Scenarios 4](#_Toc184044371)

[2.4. General Constraints 10](#_Toc184044372)

[2.5. Operating Environment 10](#_Toc184044373)

[2.6. User Documentation 10](#_Toc184044374)

[2.7. Assumptions and Dependencies 10](#_Toc184044375)

[3. External Interface Requirements 11](#_Toc184044376)

[3.1. User Interfaces 11](#_Toc184044377)

[3.2. Hardware Interfaces 11](#_Toc184044378)

[3.3. Software Interfaces 12](#_Toc184044379)

[3.4. Communications Interfaces 13](#_Toc184044380)

[4. Behavioral Requirements 14](#_Toc184044381)

[4.1. User Classes and Access Levels 14](#_Toc184044382)

[4.2. Related Real-world Objects 14](#_Toc184044383)

[4.3. System Stimuli and Responses 14](#_Toc184044384)

[4.4. Scenario-Specific Requirements 15](#_Toc184044385)

[4.5. Related Features 15](#_Toc184044386)

[4.6. Functional 15](#_Toc184044387)

[5. Non-behavioral Requirements 16](#_Toc184044388)

[5.1. Performance Requirements 17](#_Toc184044389)

[5.2. Safety Requirements 17](#_Toc184044390)

[5.3. Qualitative Requirements 17](#_Toc184044391)

[Availability 17](#_Toc184044392)

[Security 17](#_Toc184044393)

[Maintainability 17](#_Toc184044394)

[Portability 18](#_Toc184044395)

[5.4. Design and Implementation Constraints 18](#_Toc184044396)

[6. Other Requirements 19](#_Toc184044397)

[6.1. Database Requirements 19](#_Toc184044398)

[*6.2.* Operations 19](#_Toc184044399)

[7. Analysis Models 21](#_Toc184044400)

[7.1. Data Flow Model 21](#_Toc184044401)

[7.1.1 Data sources 21](#_Toc184044402)

[7.1.2. Data Sinks 21](#_Toc184044403)

[Data Dictionary 21](#_Toc184044404)

[7.2. Context Diagram (Level 0 Data Flow Diagram) 22](#_Toc184044405)

[A diagram of a diagram

Description automatically generated 23](#_Toc184044406)

[Level 1 Data Flow Diagram 23](#_Toc184044407)

[Level 2 Data Flow Diagrams 23](#_Toc184044408)

[*7.3.* *Class Model* 26](#_Toc184044409)

[*7.4.* *State Model* 27](#_Toc184044410)

[8. To Be Determined List 28](#_Toc184044411)

# Introduction

## 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 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. However, the integration of the back-end code and APIs has not yet been attempted. The project schedule has been affected by campus closures and delays in receiving the latest software, which has impacted the progress. The details outlined in this SRS represent our current understanding of the requirements and scope of the project, with all sections subject to change as the project evolves.

## 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.

## 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.

## Project References

We have started to use this GitHub (<https://github.com/TwinFan/LiveTraffic>) as a reference for our project. We are pulling parts of

## 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. |
| **Administrator** | 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. |

# General Description

## 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.

## Product Features

ClearPath is an advanced simulation system designed to integrate live aircraft traffic into the XPlane 12 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.
   1. Integrates real-time aircraft movement data into the simulation environment, including variables such as position, speed, altitude, and flight plans.
   2. This feature ensures the simulation reflects current traffic conditions, enabling realistic testing and training scenarios.
   3. The integration relies on data sources such as the OpenSky Network API and ADS-B feeds.
2. Runway Incursion Scenarios
   1. Simulates situations where aircraft, vehicles, or personnel enter active runways or taxiways.
   2. Designed to replicate real-world incidents, allowing for training and the development of mitigation strategies to enhance safety.
3. Motion Synchronization
   1. Synchronizes the physical motion of the Force Dynamics 401CR simulator with the simulation’s aircraft movement, providing a realistic training environment.
4. Event Logging
   1. Automatically logs key events, such as runway incursions, simulation settings, and user interactions, to facilitate future analysis and system improvements.

## 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
   1. 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. Administrator
   1. The Administrator manages the system’s logs and retrieves historical data for analysis, ensuring the integrity and security of stored information.
3. OpenSky Network API
   1. The OpenSky Network API serves as a data provider, supplying real-time live traffic data that integrates seamlessly into the simulation environment.
4. Force Dynamics 401CR
   1. 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, halts motion sync, and saves any logs generated during the session.

##### Steps:

1. The User clicks the 'Stop Simulation' button.
2. The system halts all live data integrations and motion synchronization.
3. Logs are finalized and stored for analysis.

##### 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.

Postcondition: All simulation processes are stopped, and the session logs are saved.

#### Adjust Settings

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

##### Steps:

1. The User navigates to the settings menu.
2. The User adjusts parameters (e.g., proximity threshold) via sliders or text boxes.
3. 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:

1. The system fetches live traffic updates and processes the data.
2. The User views aircraft data on the interface.
3. 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.

#### Access Historical Logs

The Administrator retrieves and analyzes historical logs of simulation events, such as detected incursions or system errors.

##### Steps:

1. The Administrator requests log access via the system interface.
2. The system validates access rights.
3. The requested logs are displayed for review.

##### Scenario:

Precondition: The Administrator has logged into the system.

Trigger Condition: The Administrator selects the log review option.

##### Steps:

1. The Administrator reviews and downloads historical logs.

2. The logs are archived if required.

Postcondition: Historical logs are accessed and reviewed.

#### Integrate Live Traffic Data

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

##### Steps:

1. The system sends a request to the OpenSky Network API for live data.
2. The API responds with live traffic data.
3. The system processes and injects the data into XPlane 12.

##### 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:

1. The system establishes a connection with the motion simulator.
2. Aircraft position and movement data are sent to the simulator.
3. 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.

#### Log Event Data

Logs simulation events such as runway incursions, user actions, and system alerts for future analysis.

##### Steps:

1. The system automatically records significant events during the simulation.
2. Logs are saved in the Event Database for future access.

##### Scenario:

Precondition: The simulation is running, or logs are being requested.

Trigger Condition: A significant event (e.g., incursion alert or user action) occurs.

##### Steps:

1. The system detects an event and logs it.

2. Logs are stored securely for later review.

Postcondition: All relevant events are logged and accessible.

A diagram of a diagram

Description automatically generated

Figure 1 Use Case Diagram

## 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.

## 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 stationary 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.

## User Documentation

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

## 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

# External Interface Requirements

## 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 (e.g., map view, list view) through a menu.
* [REQ-3] When the flight data panel is clicked the user shall be able to see all live traffic in the area, and flight data information (speed, altitude, heading)
* [REQ-4] When the map is clicked all live traffic shall be displayed in some capacity with callsigns or identifiers.
* [REQ-5] The system shall provide visual alerts (e.g., flashing icons) 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 incursion scenarios for the use, options include number of planes and location of incursion.
* [REQ-7] The user shall receive feedback on both actions through visual and audio alerts during simulations.

## 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 (e.g., 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.

## Software Interfaces

Required Software Products

* Name: X-Plane 12
  + Mnemonic: XP12
  + Specification Number: SRS-XP12-001
  + Version Number: 12.0
  + 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 parsed and integrated into the X-Plane 12 environment effectively.

## 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 12. Each interface requirement is designed to ensure seamless integration and user experience during training simulations involving runway incursions and bot interactions.

# Behavioral Requirements

## User Classes and Access Levels

* [REQ-14] The ClearPath System shall provide two user classes: Basic User and Administrator.
  + **Basic User**: Access to simulation controls and incursion scenario playback.
  + **Administrator**: Access to all system controls, including live data integration, scenario editing, and system configurations.
* [REQ-15] User access levels shall be defined in the Access Table with the following codes:1: Restricted
  + 1: Basic User
  + 2: Administrator

## 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 for the Administrator.

## 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 both Basic Users and Administrators of the incursion through a visible alert on the simulator screen.
  + **4.3.2**: The system shall log the event for later review, including details on aircraft IDs, positions, and time of incursion.
* [REQ-21] In the event of a network interruption or data feed loss, the system shall:
* **4.3.3**: Notify the Administrator 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.

## Scenario-Specific Requirements

* [REQ-22] For each taxiway incursion scenario, the system shall allow the Administrator 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).

## Related Features

**4.5.1 Live Traffic Integration**

* [REQ-24] The system shall connect to live data sources, such as OpenSky Network, 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 Administrator 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 and log the incursion event. The user should then take evasive action to remove the alert from the screen (example: stop the plane, turn off of the taxiway, allow traffic to pass).

## 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 every second.
* [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. 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 Administrator and log event details (timestamp, aircraft IDs, positions).

**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 Administrator 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 Administrator 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 Administrator to adjust settings as needed.
* [REQ-41] Display window settings shall be initialized based on values specified in Table F-1 in Appendix F.

# Non-behavioral Requirements

## 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 Administrator in less than 0.5 seconds upon detection.

## Safety Requirements

* [REQ-46] The system shall restrict modifications to incursion thresholds and other scenario variables to Administrator access only to prevent accidental or unauthorized changes.
* [REQ-47] The system shall require user authentication for both Basic User and Administrator roles to ensure that only authorized personnel interact with the simulation.
* [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.

## Qualitative Requirements

### Availability

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

### Security

* [REQ-50] The system shall require encrypted storage for user authentication details, following SHA-256 encryption standards.
* [REQ-51] Access to simulation controls and incursion scenario data shall be logged with timestamps, user roles, and actions performed for security auditing.
* [REQ-52] The system shall restrict data feed access through a secure API key, limiting access to OpenSky Network data only to authorized users.

### 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.
* [REQ-56] Host-dependent code shall be limited to 20% of the total codebase to facilitate easy porting to different operating systems or simulation environments.

## 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.
* [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.

# Other Requirements

## Database Requirements

* [REQ-60] The system shall store incursion event data in a database, including:
  + Event ID: Unique identifier for each incursion event.
  + Timestamp: Date and time the event occurred.
  + Aircraft IDs: Identifiers for the aircraft involved in the incursion.
  + Position Data: Latitude, longitude, and altitude of each aircraft at the time of the event.
* [REQ-61] Data retention for incursion events shall be limited to 30 days, after which data shall be automatically archived or deleted.
* [REQ-62] Access to the database shall be restricted to Administrator-level users, ensuring that only authorized personnel can view, modify, or delete event data.
* [REQ-63] The database shall maintain integrity constraints to ensure:
* Unique entries for each event ID.
* Consistent data types for position data (e.g., floating-point values for latitude, longitude, and altitude).
* [REQ-64] Data in the database shall be accessible for reporting purposes, allowing users to generate incursion event reports based on specific date ranges or event characteristics.

## 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.

* + Unattended Mode: For after-hours operation, where simulation updates and data logging continue without direct user input.

[REQ-65.6] The system shall require regular updates, issued by Laminar Research, some of which will be installed without user interaction.

* [REQ-66] The system shall include data processing support functions to:
  + Automatically archive incursion event data older than 30 days.
  + Generate summary reports on incursion events, accessible by Administrators.
* [REQ-67] The system shall perform automatic backup operations every 24 hours to ensure data is preserved in case of system failure.
* [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.

# Analysis Models

## 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 (Administrator only).

### 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 (Administrator only).
* Event Log Database:
  + Stores simulation logs and outputs, including:
    - Incursion Events: Aircraft ID, timestamp, and proximity details.
    - System Status Changes: Logs of user commands and adjustments to simulation settings.
    - Error Reports: Details on errors encountered during simulation or data processing.

### Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name*** | ***Description*** | ***Structure*** | ***Range*** |
| Flight Number | Identification ID for each live commercial flight in the simulation. | Alphanumeric | A-Z, 0-9 |
| Position Data | Latitude, longitude, and altitude of aircraft. | Numeric | Latitude: -90° to 90°Longitude: -180° to 180°Altitude: 0 to 100,000 ft |
| Time Data | Timestamp of aircraft movement. | ISO 8601 | Example: YYYY-MM-DDTHH:MM:SSZ |
| Proximity Threshold | Minimum distance for triggering an incursion alert. | Numeric (nautical miles) | 0.1 - 5.0 |
| Proximity Alert | Notification triggered when aircraft breach the proximity threshold. | Alphanumeric String | Examples: “Warning: Aircraft Incursion” or “Proximity Alert Triggered”. |
| Motion Data Sync | Data sent to the simulator to synchronize physical motion. | Numeric | Speed, heading, position updates. |
| Live Aircraft Data | Real-time data from OpenSky API, including position, speed, and heading. | JSON Structure | Latitude, longitude, altitude, speed, and heading values. |
| Simulation Logs | Event records for system interactions and incursions. | Alphanumeric Strings | Examples: “Incursion at Taxiway B”, “Error: Data Connection Lost”. |

## Context Diagram (Level 0 Data Flow Diagram)

A diagram of a system

Description automatically generated

Figure 2 - Context Diagram

### Level 1 Data Flow Diagram

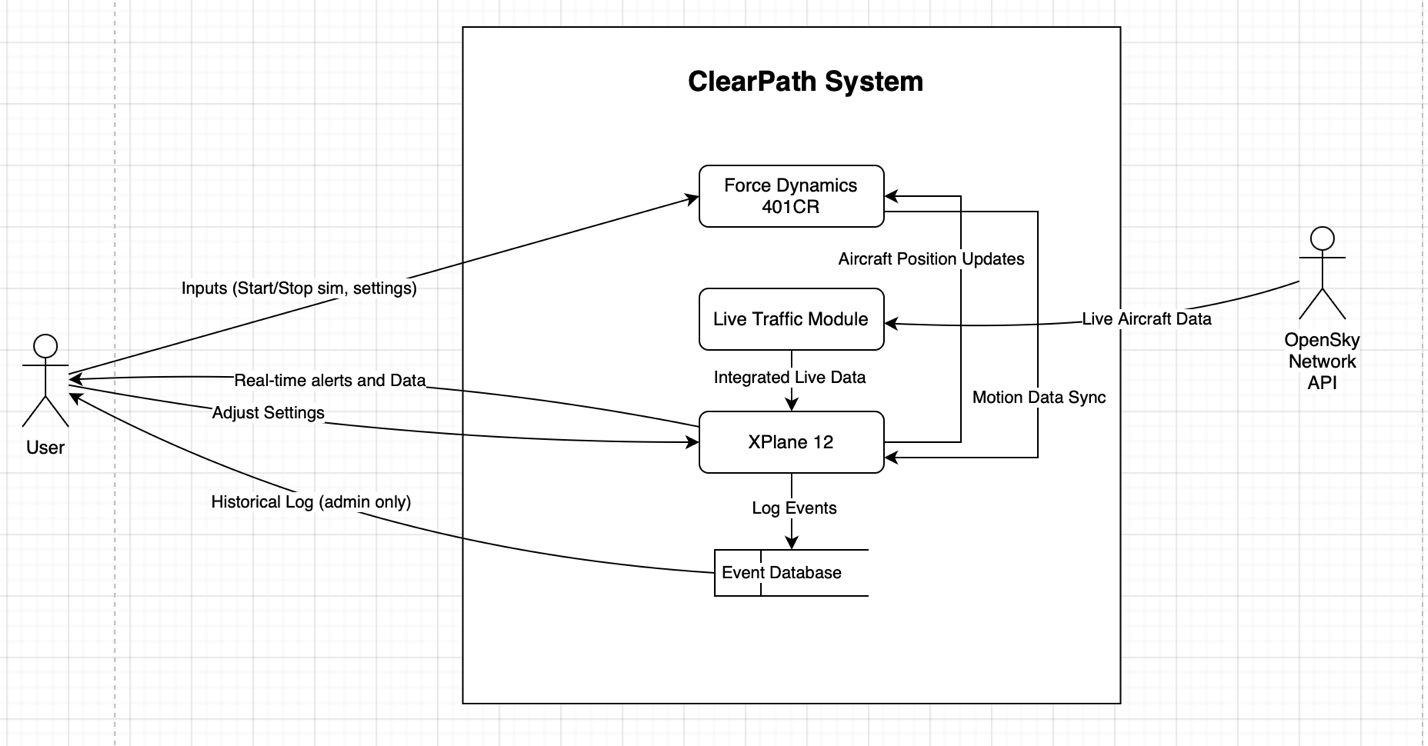


Figure 3 Level 1 Data Flow Diagram

### Level 2 Data Flow Diagrams

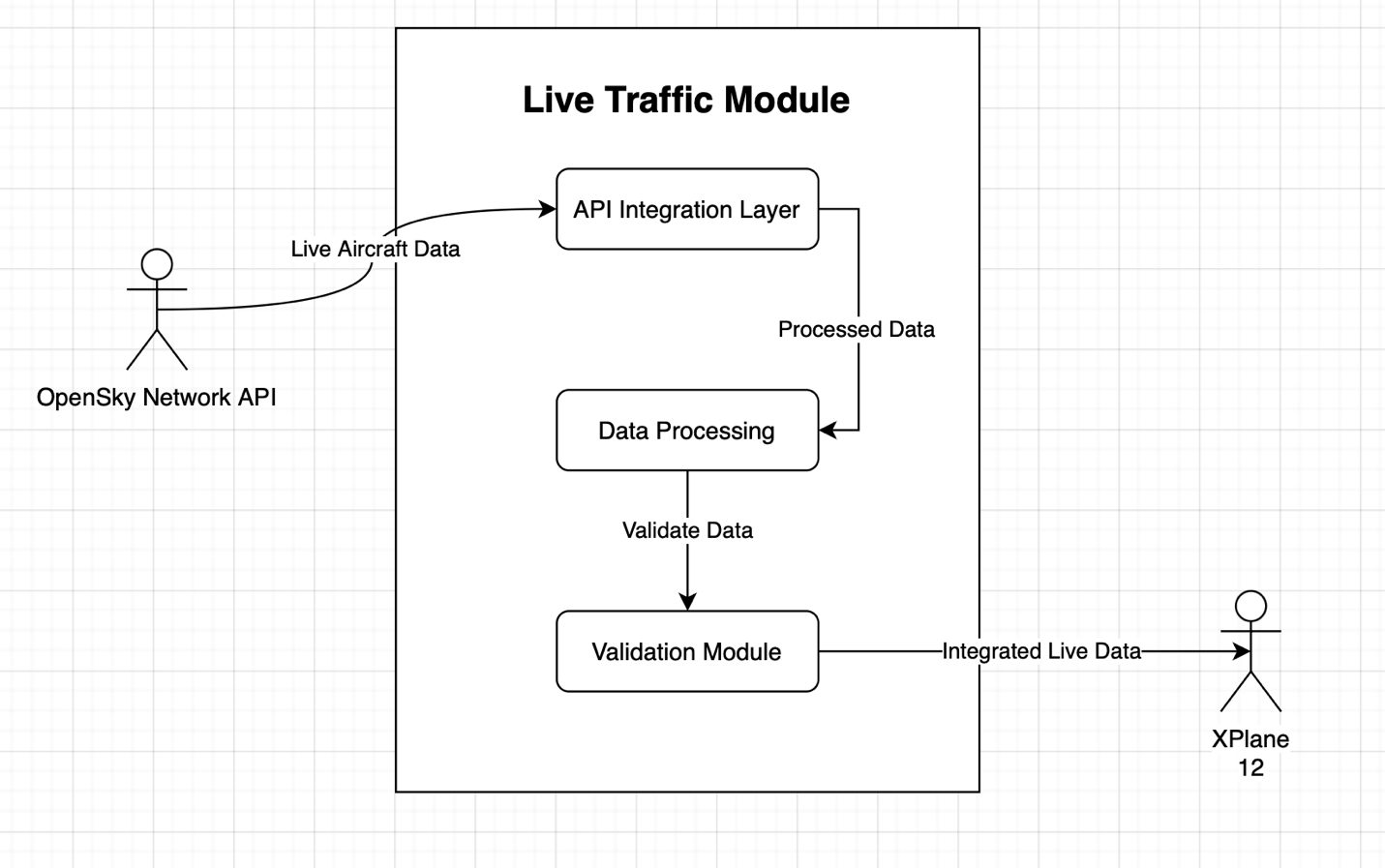


Figure 4 Live Traffic Module Data Flow Diagram

A diagram of a event database

Description automatically generated

Figure 5 Event Database Data Flow Diagram

The Force Dynamics 401CR and XPlane 12 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.

## *Class Model*

We do not have a class model diagram because the majority of the code used in our project was sourced from existing external repository and not authored by us. This code, consisting of thousands of lines, includes prewritten modules and plugins that were integrated into our system to enable functionalities such as live traffic data processing. Since we did not design or develop the underlying codebase, creating a class model diagram would require an extensive reverse-engineering effort to map the structure and relationships within the inherited code. Our primary contribution lies in the integration and configuration of this code to work seamlessly within our simulation environment.

## *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.

A diagram of a computer program

Description automatically generated

Figure 6 - State Model Diagram

# To Be Determined List

This list contains the items that are still to be determined.

* The function of including an incursion scenario in the simulator.