# ClearPath
# System Requirements Specification
## Version <1.0>
## 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 | 10/29/2025 | Full Team | Initial Copy of the SRS document |
| | | | |
| | | | |

# 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 outlines the entire project. ClearPath aims to integrate live traffic information into a physical simulator. The live traffic shall be produced in three dimensions mirroring the information being received in real time. We will then set up a situation where a plane incursion occurs on the taxiway. The primary scope for this shall remain here. The simulator has been set up, and we have gotten the controls to work thus far. This is all we have done thus far. We have only gotten the simulator up to working conditions, we have not attempted to look at the back-end code or attempted to integrate any API. The schedule of the project was thrown off due to campus closures and waiting to receive the newest software. The information within this SRS document is currently the best idea we have of the requirements and scope of this project. All of the sections are subject to change.

## 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

There are no special document conventions.

## 1.4. Project References

We do not currently have any references.

## 1.5. Definitions, Acronyms, and Abbreviations

### 1.5.1. Definitions

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

**Table 1:**

| Term | Definition |
|---|---|
| | |
| | |
| | |

### 1.5.2. Acronyms

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

**Table 2:**

| Term | Definition |
|---|---|
| FD401cr | Force Dynamics 401cr – The Hydraulic movement and frame of the SIM |
| | |
| | |

### 1.5.3. Abbreviations

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

**Table 3:**

| Term | Definition |
|---|---|
| | |
| | |
| | |

# 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 includes the following features:

- Live traffic data integration into simulation
- Runway incursion scenarios

## 2.3. User Classes and Characteristics

The users of ClearPath include:

- Customer – The customer is a department chair and professor.
- Teachers Assistant - Rajagopal Sugumar is a teacher's assistant for the CS490 class.
- Developers – The developers consist of the group members.

### 2.3.1. Actors

This section presents the actors in the system.

- End User
  - An End User is the user which uses the system for training purposes.
- Developer
  - A Developer is the user who programs and integrates the hardware and software of the system.
- Customer

- A Customer is the user who requested the developers to develop the system.

### 2.3.2. Use Cases

This section presents the Use Cases, developed for the system.

- Live Traffic Simulation
- Integrating live traffic into the 4D simulator to fly alongside real-time traffic.
- Runway Incursion Practice
- Practicing procedures required in runway incursion situations.

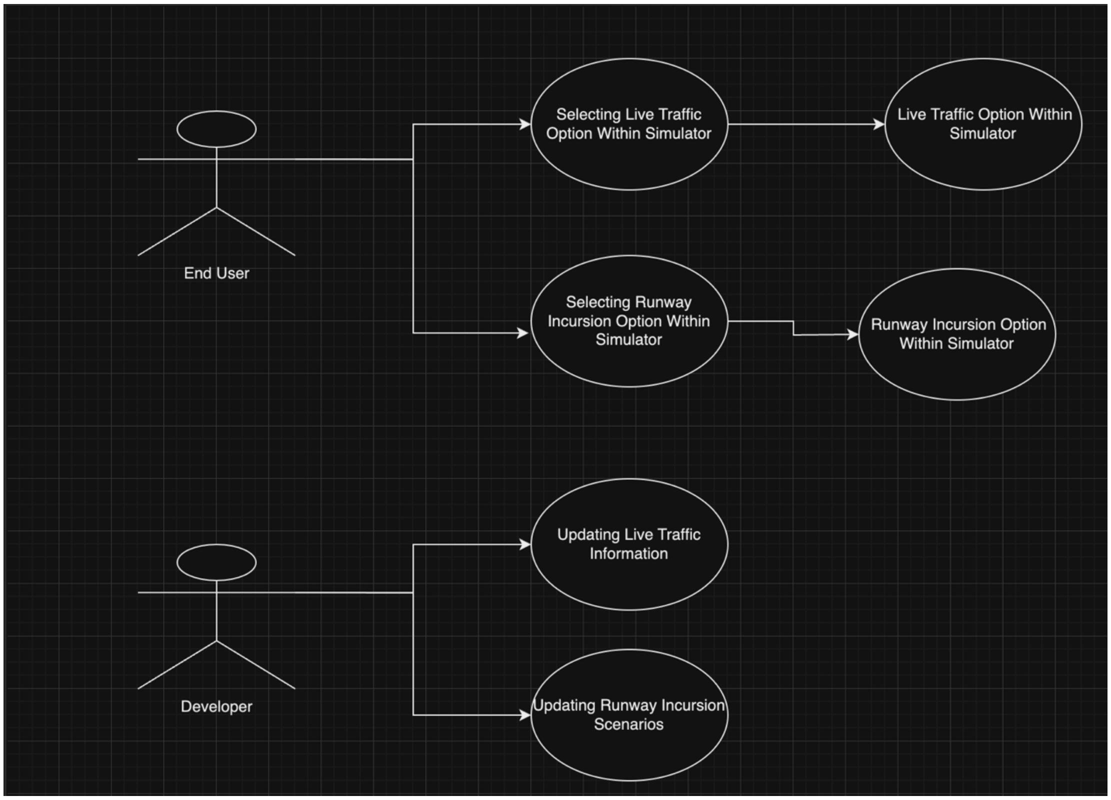
End User
Selecting Live Traffic Option Within Simulator
Live Traffic Option Within Simulator
Selecting Runway Incursion Option Within Simulator
Runway Incursion Option Within Simulator
Developer
Updating Live Traffic Information
Updating Runway Incursion Scenarios

Figure 1: Use case diagram. This figure is based solely on what we think the software looks like. We do not have a complete idea yet.

### 2.3.3. Scenarios

This section presents scenarios for **each use case**, described in the previous section.

**<u>Scenario 1: Live Traffic Simulation</u>**

**Description:** The user is integrated into the live traffic within the simulator.

**Actors:** Current User (can be a developer, customer, or end user).

**Precondition: The** system has been launched correctly and the simulator controls are working.

**Trigger Condition:** The user chooses the live traffic mode.

**Steps:**

1. Unknown at this time.

**Scenario 2: Runway Incursion Scenario**

**Description:** The system presents a runway incursion and the end user is tasked with dealing with the situation

**Actors:** Current User (can be a developer, customer, or end user).

**Precondition: The** system has been launched correctly and the simulator controls are working.

**Trigger Condition:** The user chooses to enter scenario.

**Steps:**

1. Unknown at this time

## 2.4. General Constraints

Constraints are unknown at this time.

## 2.5. Operating Environment

ClearPath is intended to be used as an application for indoor flight simulators using either full motion or stationary set-ups.

## 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 flight radar 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-3.1]*** *The system shall display real-time flight data on a dedicated screen section in a clear and legible format.*
- ***[REQ-3.2]*** *The user shall be able to toggle between different views of the flight data (e.g., map view, list view) through a menu.*
- ***[REQ-3.3]*** *The system shall provide visual alerts (e.g., flashing icons) for runway incursions.*

### 3.1.2. Bot Interaction Interface

Characteristics:

- The interface shall allow users to set parameters for the fake "bots" simulating various scenarios during training.
- Users shall be able to select difficulty levels and types of scenarios.

*Requirements:*

- ***[REQ-3.4]*** *The system shall provide a configuration panel for setting up bot behaviors, which shall be saved and loaded for future sessions.*
- ***[REQ-3.5]*** *The user shall receive feedback on bot actions through visual and audio alerts during simulations.*

## 3.2. Hardware Interfaces

*Requirements:*

- ***[REQ-3.6]*** *The power switch shall be an easily accessible toggle switch.*
- ***[REQ-3.7]*** *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:
  - 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 12
  - Mnemonic: XP12
  - Specification Number: SRS-XP12-001
  - Version Number: 12.0
  - Source: Laminar Research

- Name: FlightRadar24 API
  - Mnemonic: FR24
  - Specification Number: SRS-FR24-002
  - Version Number: 1.0
  - Source: FlightRadar24

*Requirements:*

- ***[REQ-3.8]*** *The system shall interface with the FlightRadar24 API to receive real-time flight data.*
- ***[REQ-3.9]*** *The system shall ensure that data is parsed and integrated into the X-Plane 12 environment effectively.*

## 3.4. Communications Interfaces

Local Network Protocols

*Requirements:*

- ***[REQ-3.10]*** *The system shall use TCP/IP protocol for communication between the FlightRadar24 API and the simulation environment.*
- ***[REQ-3.11]*** *The system shall implement a secure connection (HTTPS) when accessing the FlightRadar24 API to ensure data integrity and security.*

# 4. Behavioral Requirements

## 4.1. User Classes and Access Levels

***[REQ-4.1]*** *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-4.2]*** *User access levels shall be defined in the Access Table with the following codes:1: Restricted*

- *1: Basic User*
- *2: Administrator*

## 4.2. Related Real-world Objects

***[REQ-4.3]*** *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-4.4]*** *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-4.5]*** *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.*

## 4.3. System Stimuli and Responses

***[REQ-4.6]*** *When live traffic data is received from FlightRadar24, the system shall update the aircraft objects in real time to reflect their actual positions and movements.*

***[REQ-4.7]*** *Upon detection of a potential incursion on the taxiway:*

- ***4.7.1****: The system shall alert both Basic Users and Administrators of the incursion through a visible alert on the simulator screen.*

- ***4.7.2****: The system shall log the event for later review, including details on aircraft IDs, positions, and time of incursion.*

***[REQ-4.8]*** *In the event of a network interruption or data feed loss, the system shall:*

- ***4.8.1****: Notify the Administrator with an error message, while displaying the last known positions of aircraft.*
- ***4.8.2****: Attempt to reconnect to the data source every 30 seconds until the connection is restored.*

## 4.4. Scenario-Specific Requirements

***[REQ-4.9]*** *For each taxiway incursion scenario, the system shall allow the Administrator to configure variables, such as aircraft speed, proximity threshold, and alert type.*

***[REQ-4.10]*** *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-4.11]*** *The system shall connect to live data sources, such as FlightRadar24, to integrate real-time aircraft positions into the simulation.*

***[REQ-4.12]*** *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-4.13]*** *The system shall display aircraft in three dimensions, accurately reflecting altitude, speed, and heading.*

***[REQ-4.14]*** *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-4.15]*** *The system shall detect potential incursions on runways and taxiways, triggering alerts to the Administrator for intervention.*

***[REQ-4.16]** 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.*

## 4.6. Functional

### 4.6.1. Live Data Processing and Validation

***[REQ-4.17]** 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-4.18]** If data validation identifies anomalies, the system shall revert to the last valid data state and notify the Administrator.*

### 4.6.2. Aircraft Position and Movement Updates

***[REQ-4.19]** 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-4.20]** The system shall ensure smooth aircraft movement by interpolating position data to avoid abrupt transitions.*

### 4.6.3. Incursion Detection Logic

***[REQ-4.21]** 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-4.22]** 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-4.23**] The system shall handle overflow errors during data processing by queuing excess data and processing it sequentially.*

***[REQ-4.24]** 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-4.25]*** *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-4.26]*** *The system shall display default aircraft and runway positions based on configuration files, allowing the Administrator to adjust settings as needed.*

***[REQ-4.27]*** *Display window settings shall be initialized based on values specified in ______________.*

# 5. Non-behavioral Requirements

## 5.1. Performance Requirements

***[REQ-5.1]*** The system shall support one user interacting with the simulation controls.

***[REQ-5.2]*** The system shall process real-time updates of aircraft position, speed, and heading every one second, ensuring smooth transitions within the simulator.

***[REQ-5.3]*** During peak operations, the system shall handle up to 50 aircraft in the simulated environment without exceeding a 2-second delay in updates.

***[REQ-5.4]*** *95% of simulated incursion scenarios shall trigger alerts to the Administrator in less than 0.5 seconds upon detection.*

## 5.2. Safety Requirements

***[REQ-5.5]*** *The system shall restrict modifications to incursion thresholds and other scenario variables to Administrator access only to prevent accidental or unauthorized changes.*

***[REQ-5.6]*** *The system shall require user authentication for both Basic User and Administrator roles to ensure that only authorized personnel interact with the simulation.*

***[REQ-5.7]*** *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

### 5.3.1. Availability

***[REQ-5.8]*** *The system shall be operational with 99.5% availability during scheduled hours to ensure continuous access for training and simulation purposes.*

***[REQ-5.9]*** *The system shall automatically checkpoint simulation data every 5 minutes to allow for recovery and restart without significant data loss.*

### 5.3.2. Security

***[REQ-5.10]*** *The system shall require encrypted storage for user authentication details, using Windows 10 security and encryption.*

***[REQ-5.11]*** *Access to simulation controls and incursion scenario data shall be logged with timestamps, user roles, and actions performed for security auditing.*

***[REQ-5.12]*** *The system shall restrict data feed access through a secure API key, limiting access to FlightRadar24 data only to authorized users.*

### 5.3.3. Maintainability

***[REQ-5.13]*** *The software shall be designed with modular components for simulation controls, data integration, and incursion detection to simplify future maintenance and updates.*

***[REQ-5.14]*** *Each module shall contain documentation for functionality, parameters, and dependencies, ensuring that new developers can make updates with minimal onboarding.*

### 5.3.4. Portability

***[REQ-5.15]*** *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-5.16]*** *Host-dependent code shall be limited to 20% of the total codebase to facilitate easy porting to different operating systems or simulation environments.*

## 5.4. Design and Implementation Constraints

***[REQ-5.17]*** *The system shall comply with the existing XPlane simulator hardware and software setup without requiring additional hardware modifications.*

***[REQ-5.18]*** *The software shall be compatible with FlightRadar24 API standards, ensuring seamless data integration for real-time traffic.*

***[REQ-5.19]*** *The system shall store no more than 500 entries in the log for incursion detection events to prevent memory overload on the simulator hardware.*

# 6. Other Requirements

## 6.1. Database Requirements

***[REQ-6.1]*** *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-6.2]*** *Data retention for incursion events shall be limited to 30 days, after which data shall be automatically archived or deleted.*

***[REQ-6.3]*** *Access to the database shall be restricted to Administrator-level users, ensuring that only authorized personnel can view, modify, or delete event data.*

***[REQ-6.4]*** *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-6.5]*** *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.*

## *6.2.* Operations

***[REQ-6.6]*** *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.*
- *Unattended Mode: For after-hours operation, where simulation updates and data logging continue without direct user input.*

***[REQ-6.7]*** *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-6.8]*** *The system shall perform automatic backup operations every 24 hours to ensure data is preserved in case of system failure.*

***[REQ-6.9]*** *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

Because we have not gotten past downloading the X-Plane 12 software and running it on the simulator we do not currently know what is required for the state model. Reference section 1.1.

### 7.1.1. Data Sources

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

- Flight Radar
- Time & Position Data of Planes for FlightRadar24

### 7.1.2. Data Sinks

No current data sinks.

### 7.1.3. Data Dictionary

No current items needed in data dictionary.

### 7.1.4. Context Diagram (Level 0 Data Flow Diagram)

Because we have not gotten past downloading the X-Plane 12 software and running it on the simulator we do not currently know what is required for the Context Diagram. Reference section 1.1.

### 7.1.5. Level 1 Data Flow Diagram

Because we have not gotten past downloading the X-Plane 12 software and running it on the simulator we do not currently know what is required for the data flow diagram. Reference section 1.1.

### 7.1.6. Level 2 Data Flow Diagram

Because we have not gotten past downloading the X-Plane 12 software and running it on the simulator we do not currently know what is required for the state model. Reference section 1.1.

## 7.2. Class Model

Because we have not gotten past downloading the X-Plane 12 software and running it on the simulator we do not currently know what is required for the class model. Reference section 1.1.

## 7.3. State Model

Because we have not gotten past downloading the X-Plane 12 software and running it on the simulator we do not currently know what is required for the state model. Reference section 1.1.

# 8. To Be Determined List

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

- The function of creating a new 3D object in the simulator
- The function of including a incursion scenario in the simulator.
- The function of integrating live data