Global Navigation Satellite System (GNSS) Resilience
System Requirements Specification

Version 2

02/04/2025

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:

Pate, Clay

Customer(s): Clifford, Jayson F
Neighbors, Jake C.

Project team members:

Arena, Cristina M.
Cappers, Haskell A.
Legro, Jonathan
Lofton, Tyler R.

Change Summary

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

Version	Date	Modifier	Description
1.1	10/24/2024	Jonathan Legro	Initial document creation. Includes parts of the project introduction and general description.
1.2	10/28/2024	Jonathan Legro	Edited more of the project references and terminology. (1.4)
1.3	11/10/2024	Jonathan Legro	Remove developers from Intended Audience and Terminology in the Introduction. (1.1, 1.2)
1.4	11/10/2024	Jonathan Legro	Remove developers from Actors in the General Description (2.3.1)
1.5	11/10/2024	Jonathan Legro	Remove developers from Use Cases in the General Description. (2.3.2)
1.6	11/15/2024	Haskell Cappers	Remove all SRS template information for readability and consistency. (3, 4, 5, 6, 7, 8)
1.7	11/15/2024	Jonathan Legro	Edited Scenarios to include a description, precondition, trigger condition, and alt cases. (2.3.3)
1.8	11/15/2024	Cristina Arena	Added and edited the links within the Project References and User

			Documentation to display a title not a URL.(1.4, 2.6)
1.9	11/20/2024	Jonathan Legro	Added information into Functional within Behavioral Requirements. (5.5)
1.10	11/20/2024	Jonathan Legro	Added detailed information to General Constraints and Operating Environment. (2.4, 2.5)
1.11	11/25/2024	Tyler Lofton	Added figure number for figure 1, added figure 2. Added information in (7.1.3, 7.1.4, 7.1.5.)
2.1	01/11/2025	Jonathan Legro	Added additional requirements into within Behavioral Requirements. (5.5)
2.2	01/13/2025	Jonathan Legro	Edited Table of Contents to reflect accurate page numbers for all sections and figures.
2.3	01/13/2025	Jonathan Legro	Added Microsoft Flight Simulator Developer Support as a seventh reference in 1.4 Project References.
2.4	01/16/2025	Tyler Lofton	Edited Figure 1: GPS Display Plugin Integration diagram in 2.1 Product Perspective for accuracy and clarity.
2.5	01/16/2025	Tyler Lofton	Edited Figure 2: Use Case Diagram in 2.2 Product Features to align with updated system functionality.
2.6	01/22/2025	Jonathan Legro	Updated 2.3.1 Actors to include System Tester as a new actor.
2.7	01/22/2025	Jonathan Legro	Added 2.3.2 Use Cases: Validate System Performance to reflect the new actor's role.
2.8	01/22/2025	Jonathan Legro	Added MSFS Developer Support link to 2.4 General Constraints to guide design and programming standards.
2.9	01/22/2025	Jonathan Legro	Specified display size requirements in 2.5 Operating Environment to include 800x480 (16:9) and 1024x768 (4:3) resolutions.
2.10	01/27/2025	Jonathan Legro	Revised External Interface Requirements (3.0) to reflect hardcoded GPS plugin functionality and MSFS SDK integration.
2.11	01/27/2025	Jonathan Legro	Updated User Interfaces (3.1) to specify a read-only display of pre-defined GPS navigation data.
2.12	01/27/2025	Jonathan Legro	Updated Hardware Interfaces (3.2) to clarify the absence of external hardware dependencies.

2.13	01/27/2025	Jonathan Legro	Updated Software Interfaces (3.3) to describe integration with MSFS 2020 SDK for displaying hardcoded GPS data.
2.14	01/27/2025	Jonathan Legro	Updated Communications Interfaces (3.4) to confirm the plugin operates without external communication dependencies.
2.15	01/27/2025	Jonathan Legro	Reviewed and aligned Behavioral Requirements (4.0) with system use cases and scenarios based on hardcoded data functionality.
2.16	01/30/2025	Jonathan Legro	Added Section 8: SRS Section Assignment to define stakeholder communication and societal impacts.
2.17	01/27/2025	Jonathan Legro	Added 8.1.1 Public Health to evaluate GPS plugin impacts on aviation safety and public welfare.
2.18	01/27/2025	Jonathan Legro	Added 8.1.2 Safety to address the GPS plugin's contributions to flight safety and simulation.
2.19	01/27/2025	Jonathan Legro	Added 8.1.3 Welfare to highlight the plugin's broader societal benefits.
2.20	01/27/2025	Tyler Lofton	Added 8.2.1 Global Effect to assess the GPS plugin's international impact on aviation safety and training.
2.21	01/27/2025	Cristina Arena	Added 8.2.2 Perspective of potential social impact to analyze the plugin's inclusivity and accessibility for diverse groups.
2.22	01/27/2025	Jonathan Legro	Added 8.2.3 Environmental to evaluate the plugin's environmental benefits compared to physical training.
2.23	01/27/2025	Haskell Cappers	Added 8.2.4 Economic to assess the plugin's cost-effectiveness for training and simulation.
2.24	02/02/2025	Cristina Arena	Added 4.5 REQ-029 through REQ-042 Hardware Requirements.
2.25	02/04/2025	Cristina Arena	Updated 3.2 Hardware Interfaces section in External Interface Requirements.
2.26	02/04/2025	Jonathan Legro	Added to 9. To Be Determined List to clarify scope of the project and its limitations.
2.27	02/04/2025	Tyler Lofton	Updated figures 2, 3, and 4 and added the description

Table of Contents

1. Introduction

1.1. Purpose and Scope

The System Requirements Specification (SRS) outlines the development of a GPS display plugin for Microsoft Flight Simulator 2020 (MSFS 2020). The plugin's primary purpose is to simulate real-world GPS navigation, providing pilots with a realistic tool for navigation practice. The scope includes the creation and integration of the GPS display within MSFS 2020's instrument panel, using TypeScript and HTML. This project excludes any changes to the simulator's core functions and does not involve real-world GPS data integration. The system, called the GNSS Resilience - GPS Display Plugin for MSFS 2020, addresses the absence of a GPS display in MSFS 2020 and establishes a foundation for future enhancements, such as geo-spoofing simulations. The plugin's application benefits primary users, including pilots and researchers, by enabling realistic navigation practice, supporting future aviation vulnerability testing, and providing a stable GPS diagnostic tool within the simulator.

1.2. Intended Audience and Reading Suggestions

This document is intended for:

1. **Primary Audience: Center for Aerospace Resilient Systems (CARS)** – CARS personnel can use this document to understand the project goals, scope, and technical requirements.
2. **Secondary Audience:**
   - **Federal Aviation Administration (FAA) & Commercial Airlines** – FAA and airline reps should review the Overview and System Features to assess how the plugin could support navigation training and GNSS resilience.
   - **MSFS 2020 Players** – Players may refer to the Overview and User Scenarios for insights into how the plugin enhances gameplay and navigation capabilities.

Suggested reading for:

1. **Center for Aerospace Resilient Systems (CARS)**
   - **Sections**: Review *1.1 Purpose and Scope* and *2. General Description* for an overview of project objectives, followed by *4. Behavioral Requirements* to understand functionality and performance criteria.
2. **Federal Aviation Administration (FAA) & Commercial Airlines**
   - **Sections**: Review *1.1 Purpose and Scope* and *2. General Description*, focusing on *2.2 Product Features* and *2.3 User Classes and Characteristics* to understand user scenarios and benefits.
3. **MSFS 2020 Players**

  - **Sections**: Review *1.1 Purpose and Scope*, *2.3 User Classes and Characteristics*, and *2.4 General Constraints* for purpose, user roles, and gameplay enhancements.

1.3. Document Conventions

- **Emphasis**: Key terms are bolded; requirements are tagged (e.g., [REQ-101]).
- **Priorities**: Each requirement has an individual priority.
- **Formatting**: Standard headings and numbering for easy reference.

1.4. Project References

1) **Microsoft Flight Simulator**

- **Title**: *Microsoft Flight Simulator*
- **Author**: Wikipedia Contributors
- **Source**: Wikipedia
- **Date**: Accessed 2024
- **Location**: Microsoft Flight Simulator Wiki
- *Description*: Provides general information on Microsoft Flight Simulator, relevant for understanding the simulator's framework and user environment.

2) **TypeScript Documentation**

- **Title**: *TypeScript: JavaScript With Syntax for Types*
- **Author**: Microsoft
- **Version**: Latest
- **Date**: Accessed 2024
- **Location**: TypeScript is JavaScript with syntax for types.
- *Description*: The official TypeScript documentation, useful for developers implementing the plugin within MSFS 2020.

3) **Federal Aviation Administration (FAA) - GNSS and GPS**

- **Title**: *Global Navigation Satellite System (GNSS) and GPS*
- **Author**: Federal Aviation Administration (FAA)
- **Date**: Accessed 2024
- **Location**: Satellite Navigation - Global Positioning System (GPS)
- *Description*: An FAA resource detailing GNSS and GPS functionality, supporting the project's relevance to aviation resilience.

4) **Advanced Navigation - GNSS and Satellite Navigation Explained**

- **Title**: *Global Navigation Satellite System (GNSS) and Satellite Navigation Explained*
- **Author**: Advanced Navigation
- **Date**: Accessed 2024
- **Location**: Global Navigation Satellite System (GNSS) and Satellite Navigation Explained
- *Description*: An article on GNSS concepts, providing background information on satellite navigation relevant to GPS functionality in the plugin.

5) **Microsoft Flight Simulator SDK Documentation**

- **Title**: *Microsoft Flight Simulator SDK Documentation*
- **Author**: Microsoft
- **Date**: Accessed 2024
- **Location**: SDK Documentation
- *Description*: The official SDK documentation for MSFS 2020, essential for understanding development standards and integration guidelines.

6) **Microsoft Flight Simulator FAQ**

- **Title**: *Minimum, Recommended, and Ideal PC requirements for Microsoft Flight Simulator*
- **Author**: Microsoft
- **Date**: April 23, 2020
- **Location**: General & Announcements
- *Description*: The official SDK Support Update for MSFS 2020, essential for understanding hardware constraints.

7) **Microsoft Flight Simulator Developer Support**

- **Title**: *Official Developer Support Platform for MSFS 2020*
- **Author**: Microsoft
- **Date**: Ongoing
- **Location**: MSFS DevSupport
- *Description*: Comprehensive platform offering SDK documentation, best practices, community forums, and troubleshooting resources for developing add-ons and features for Microsoft Flight Simulator 2020.

1.5. Definitions, Acronyms, and Abbreviations

1.5.1. Definitions

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

Table 1: Terms

Term	Definition
Spoofing	Mimicking signals to deceive a system, often used to mislead GPS navigation.
Display Plugin	Software component added to extend MSFS 2020 with GPS display functionality.
TypeScript	Typed JavaScript used for developing the GPS plugin in MSFS 2020.
Instrument Panel	Displays essential flight info in MSFS 2020, where the GPS plugin is integrated.
End User	Uses the GPS plugin in MSFS 2020 for navigation.
Researcher	Tests GNSS resilience using the plugin.
Simulator	Imitative representation of a process or system.
Actors	All users of the system.

1.5.2. Acronyms

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

Table 2: Acronyms

Term	Definition
SDD	System Design Document
SRS	System Requirements Specification
GNSS	Global Navigation Satellite System
MSFS	Microsoft Flight Simulator
GPS	Global Positioning System
FAA	Federal Aviation Administration
CARS	Center for Aerospace Resilient Systems
HTML	HyperText Markup Language

1.5.3. Abbreviations

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

Table 3: Abbreviations

Term	Definition
e.g.	For example

2. General Description

2.1. Product Perspective

The GNSS Resilience - GPS Display Plugin for Microsoft Flight Simulator 2020 (MSFS 2020) is a self-contained tool developed for the Center for Aerospace Resilient Systems (CARS). It adds realistic GPS navigation functionality to the simulator, allowing users to practice and test navigation as well as study GNSS resilience. This plugin supports CARS' research on aviation resilience and provides the Federal Aviation Administration (FAA) and airlines with a tool to simulate navigation and test vulnerabilities. It fits within a larger initiative focused on aviation safety and resilience.

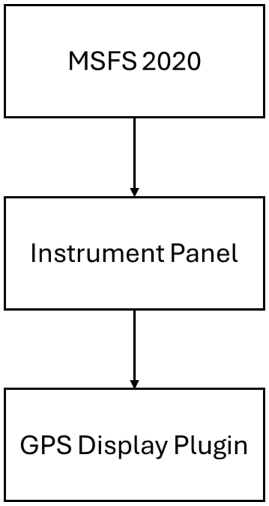

Figure 1: GPS Display Plugin Integration

2.2. Product Features

The GNSS GPS Display Plugin for MSFS 2020 includes:

1. **GPS Navigation Display**: Shows realistic GPS data within MSFS 2020.
2. **Map View**: Displays location, route, and waypoints on a visual map.
3. **Waypoint Management**: Allows users to set waypoints for navigation.
4. **Navigation Data**: Shows essential data (altitude, speed, coordinates).

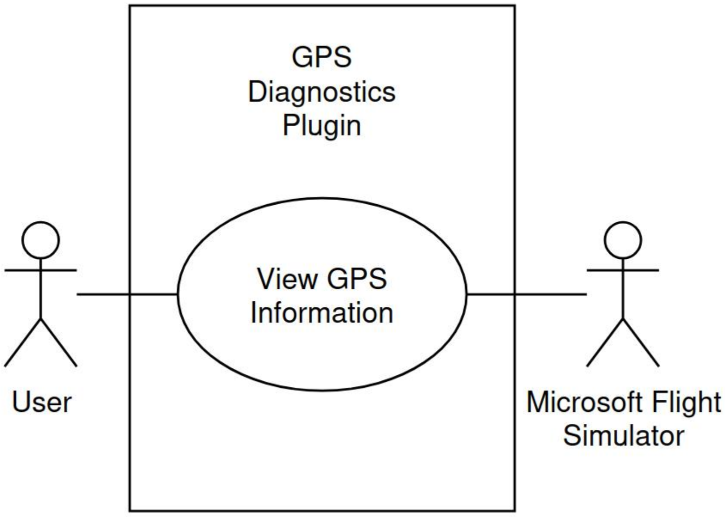

Figure 2 Use Case Diagram

2.3. User Classes and Characteristics

2.3.1. Actors

This section identifies the primary actors interacting with the GNSS GPS Display Plugin.

- **End User**
  - An End User is anyone who uses the plugin to enhance their flight simulation experience in MSFS 2020, such as pilots or aviation enthusiasts.
- **System Tester**
  - A professional or technical user responsible for evaluating the plugin's performance, diagnosing issues, and ensuring compatibility with MSFS 2020 during testing and validation phases.

2.3.2. Use Cases

This section presents the primary use cases developed for the GNSS GPS Display Plugin.

- **Enhance Flight Simulation**
  - *Users* use the plugin to improve navigation in MSFS 2020, accessing GPS data, maps, and waypoints.
- **Validate System Performance**

  - *System Testers use the plugin to ensure it functions correctly and integrates seamlessly with MSFS 2020. This includes identifying bugs, troubleshooting errors, and verifying compatibility with the simulator's systems.*

2.3.3. Scenarios

Scenario 1: Enhance Flight Simulation

- **Description:**
  - The GPS display plugin is used by an end user to enhance their flight simulation experience in MSFS 2020 by providing real-time navigation features such as waypoint management and map updates.
- **Precondition:**
  - The GPS display plugin is installed and correctly integrated into the simulator. The simulator is running, and the user has loaded a flight session.
- **Actors**:
  - End User
- **Trigger Condition**:
  - The user opens the GPS display within the instrument panel of MSFS 2020..
- **Steps**:
  - GPS interface loads.
  - User sets waypoints.
  - Map updates with navigation data.
- **Alt Cases:**
  - If the GPS interface fails to load, an error message is displayed, and the system provides a retry option.
  - If the map fails to update, the system displays a message prompting the user to reinitialize the GPS interface.

Scenario 2: Validate System Performance

- **Description:**
  - The GPS display plugin is tested by a System Tester to ensure it integrates correctly with MSFS 2020, operates as intended, and meets performance requirements.
- **Precondition:**
  - The GPS display plugin is installed and operational within MSFS 2020. The researcher has defined test parameters for GNSS simulations.
  -
- **Actors**:
  - System Tester
- **Trigger Condition:**
  - The System Tester initiates a diagnostic session for the GPS display plugin.

- **Steps**:
  - The plugin connects to the simulator environment.
  - Diagnostic tools analyze the GPS display's functionality.
  - Test results are reviewed for any errors or integration issues.
- **Alt Cases:**
  - If errors are detected, the system generates logs for troubleshooting.
  - If integration problems arise, the tester identifies solutions and reruns the diagnostics.

2.4. General Constraints

1. **Organizational**: Must align with CARS' requirements and MSFS 2020 compatibility.
2. **Hardware**: The GPS display plugin must operate within the following minimum hardware requirements, as dictated by MSFS 2020:
   - Minimum Specifications (AMD):
     1. Operating System: Windows 10 (November 2019 update, version 1909).
     2. CPU (Processor): AMD Ryzen 3 1200.
     3. GPU (Graphics Card): Radeon RX 570.
     4. VRAM (Video Memory): 2 GB.
     5. RAM (Memory): 8 GB.
     6. Storage: 150 GB HDD.
     7. Bandwidth: 5 Mbps internet connection.
   - Minimum Specifications (NVIDIA):
     1. Operating System: Windows 10 (November 2019 update, version 1909).
     2. CPU (Processor): Intel i5-4460.
     3. GPU (Graphics Card): NVIDIA GTX 770.
     4. VRAM (Video Memory): 2 GB.
     5. RAM (Memory): 8 GB.
     6. Storage: 150 GB HDD.
     7. Bandwidth: 5 Mbps internet connection.
3. **Interfaces**: Integrates with MSFS 2020 using TypeScript and HTML only.
4. **Technology**: Limited to TypeScript and HTML within MSFS 2020.
5. **Security**: Access restricted to authorized users only.
6. **Standards**: Follows MSFS 2020's design and programming guidelines available on the MSFS DevSupport Platform, which provides SDK documentation and resources.

These constraints ensure compatibility, security, and stable performance within the simulator.

2.5. Operating Environment

The GNSS GPS Display Plugin for MSFS 2020 will run on:

- **Operating System**: Windows 10 or higher.

- **Hardware Requirements (Minimum):**
    - **Processor:** AMD Ryzen 3 1200 or Intel i5-4460.
    - **Graphics Card:** Radeon RX 570 or NVIDIA GTX 770 with 2 GB VRAM.
    - **RAM:** 8 GB.
    - **Storage:** 150 GB HDD for MSFS 2020 and the plugin.
    - **Bandwidth:** Minimum 5 Mbps internet connection for MSFS 2020.
    - **Software**: Integrated within MSFS 2020, using TypeScript and HTML.
- **Display**: The plugin supports small avionics screens with resolutions like **800x480 (16:9)** or **1024x768 (4:3)**. It works with knobs or buttons for input, ensuring cockpit compatibility.

The plugin will operate smoothly alongside other MSFS 2020 components without affecting performance.

2.6. User Documentation

1. **User Manual**: Guide for installation, setup, and usage.
    - SDK Documentation
2. **Online Help**: In-app tips and troubleshooting.
    - Microsoft Flight Simulator FAQ

Formats: Delivered in PDF and accessible online; in-app help follows MSFS 2020 standards.

2.7. Assumptions and Dependencies

The project assumes:

- **Operating System**: Windows 10 or higher will be available.
- **Development Framework**: MSFS 2020 will continue supporting TypeScript and HTML integration.
- **Documentation Access**: MSFS 2020 documentation and tools will remain available and up to date.
- **Simulated Data**: Only MSFS 2020-provided simulated GPS data will be used.
- **Testing Environment**: MSFS 2020 and its instrument panel will be available for testing.

Changes to these assumptions may impact the requirements in this SRS.

3. External Interface Requirements

3.1. User Interfaces

The GPS plugin provides a graphical display within the MSFS 2020 instrument panel. It presents satellite navigation data, including waypoints, routes, and aircraft positions, but does not accept user input. The interface is designed to resemble real-world aviation GPS displays, ensuring readability and usability.

- **Display Format:** Fixed layout with standard aviation symbology.
- **Interaction:** No direct user interaction; the display updates automatically based on hardcoded navigation sequences.
- **Rendering:** Integrated within MSFS 2020 using the simulator's rendering engine.

3.2. Hardware Interfaces

The hardware components of the system interact with MSFS 2020 through a USB microcontroller, which serves as the interface between physical cockpit controls and the simulator software. This section outlines the key hardware interfaces used to achieve seamless integration with MSFS 2020.

Microcontroller Interface

The system utilizes a USB microcontroller board to process inputs from physical components such as switches, encoders, and potentiometers. This microcontroller communicates with MSFS 2020 using SimConnect to register input actions and provide feedback to the user.

- **Connection Type:** USB 2.0 or USB-C, depending on the microcontroller model.
- **Protocol:** SimConnect for data transmission between the microcontroller and MSFS 2020.
- **Power Supply:** 5V via USB or optional external 12V power supply for high-power components.

Landing Gear and Flaps Interface

The system includes mechanical controls for landing gear and flaps, which use rotary encoders, limit switches, or potentiometers to send positional data to MSFS 2020.

- **Landing Gear Lever:** Equipped with a rotary encoder or limit switch to detect up/down positions.
- **Flaps Lever:** Uses a 10KΩ linear potentiometer or rotary encoder to provide smooth analog input for flap adjustments.
- **Parking Brake Switch:** Implemented as a heavy-duty toggle switch or momentary push button with mechanical locking.

Control Panel and Switches Interface

A dedicated control panel houses various cockpit functions, including labeled toggle switches and push buttons for essential aircraft controls.

- **Battery and Alternator Controls:** Toggle switches mapped to MSFS 2020 power systems.
- **Avionics Master Switch:** A momentary or latching switch for avionics power control.
- **Rotary Encoders:** Dedicated encoders for heading, altitude, NAV, and COM radio tuning.

Indicator Lights and LED Feedback Interface

Warning and caution lights provide visual status updates based on real-time flight conditions received from MSFS 2020.

- **LED Indicators:** Bright red and yellow LEDs for landing gear status, warnings, and caution alerts.
- **Control Logic:** LED states are managed by the microcontroller based on MSFS 2020 telemetry.
- **Power Source:** 5V direct from microcontroller or external 12V supply for high-intensity lighting.

Performance and Input Handling

The hardware system is designed to ensure real-time synchronization of physical inputs with MSFS 2020.

- **Input Processing Time:** Less than 100ms latency to prevent noticeable lag.
- **Self-Test Function:** The system initializes a hardware diagnostic routine upon startup to verify the operational status of all inputs and indicators.
- **Future Expansion:** Modular design allows integration of motorized components such as linear actuators for landing gear movement.

By implementing these hardware interfaces, the system enhances the realism and interactivity of MSFS 2020, providing a robust DIY flight simulation experience.

3.3. Software Interfaces

The GPS plugin integrates with **MSFS 2020's SDK** using **TypeScript and HTML** to overlay hardcoded navigation data within the simulator's instrument panel.

- **MSFS SDK Integration:**
  - Retrieves **aircraft state data** (e.g., position, altitude, heading) from MSFS 2020.
  - Aligns hardcoded GPS waypoints with simulator-generated flight conditions.
- **Rendering and Data Handling:**
  - Uses **HTML/CSS** for the GPS display layout.
  - Implements **TypeScript logic** to simulate waypoint progression and navigation updates.
- **Compatibility:**
  - Fully contained within MSFS 2020's execution environment.
  - Does not depend on external software, databases, or third-party APIs.

3.4. Communications Interfaces

The GPS plugin operates entirely within MSFS 2020 and does not require external communication.

- **No Live Data Transmission:**

  - The system does not communicate with external GPS satellites, real-world networks, or online servers.
  - All GPS navigation data is **pre-defined and hardcoded** within the plugin.
- **No External Dependencies:**
  - The plugin does not exchange data over the internet, Bluetooth, or any other network protocol.
  - There is **no API interaction** beyond MSFS 2020's internal SDK.
- **Standalone Operation:**
  - The plugin functions **independently** within the simulator, ensuring stable operation without requiring live data sources.

4. Behavioral Requirements

4.1. Same Class of User

N/A

4.2. Related Real-world Objects

4.3. Stimulus

4.4. Related Features

4.5. Functional

The GPS display plugin must meet the following functional requirements to ensure seamless user experience within MSFS 2020:

Core Functional Requirements

- **[REQ-001]** The system shall initialize and display the GPS interface as a static image within the MSFS 2020 instrument panel.
- **[REQ-002]** The system shall successfully load and execute the JavaScript program responsible for displaying the GPS plugin image.
- **[REQ-003]** The system shall refresh the GPS display image at a rate of 2 updates per second (0.5s interval) to reflect new information.
- **[REQ-004]** The system shall ensure smooth image transitions during updates, minimizing flickering and rendering artifacts.
- **[REQ-005]** The system shall keep the GPS display visible in the instrument panel throughout the simulation session

- **[REQ-006]** The system shall keep the GPS display correctly positioned in the instrument panel throughout the simulation session.
- **[REQ-007]** The system shall not modify MSFS 2020's core simulation functionality while rendering the GPS display image.
- **[REQ-008]** The system shall not interfere with MSFS 2020's core simulation functionality while rendering the GPS display image.

Integration & Interface Requirements

- **[REQ-009]** The system shall initialize the GPS plugin within MSFS 2020 without requiring user input.
- **[REQ-010]** The system shall integrate with MSFS 2020's rendering engine to correctly display the GPS image.
- **[REQ-011]** The system shall ensure the GPS display image resizes dynamically to fit various screen resolutions.
- **[REQ-012]** The system shall refresh and replace the GPS display image only when information is available, avoiding unnecessary updates.

Performance Requirements

- **[REQ-013]** The system shall initialize the GPS display image within 2 seconds after MSFS 2020 startup.
- **[REQ-014]** The system shall maintain a frame rate impact of less than 5% relative to MSFS 2020 baseline performance.
- **[REQ-015]** The system shall ensure image updates occur efficiently without exceeding 5% CPU utilization.

Error Handling & Logging

- **[REQ-016]** The system shall display an error message if the GPS display image fails to load.
- **[REQ-017]** The system shall log errors related to image rendering failures for debugging.
- **[REQ-018]** The system shall allow debugging features to be enabled via a configuration file.
- **[REQ-019]** The system shall allow debugging features to be disabled via a configuration

Scalability & Maintainability

- **[REQ-020]** The system shall follow modular design principles to allow future enhancements, such as adding new satellite configurations.
- **[REQ-021]** The system shall maintain compatibility with MSFS 2020 updates to ensure continued functionality.

GPS Signal & Satellite Display Updates (Static)

- **[REQ-022]** The system shall update the satellite constellation image once per second (1Hz) based on precomputed aircraft position changes.

Satellite Integration & Performance Requirements

- **[REQ-023]** The system shall retrieve latitude from MSFS 2020's aircraft positioning system.
- **[REQ-024]** The system shall retrieve longitude from MSFS 2020's aircraft positioning system.
- **[REQ-025]** The system shall retrieve altitude from MSFS 2020's aircraft positioning system.
- **[REQ-026]** The system shall update satellite images without exceeding 5% CPU usage.
- **[REQ-027]** The system shall allow testers to modify key parameters through a configuration file.

Core Hardware Functional Requirements

- **[REQ-028]** The system shall include a landing gear lever equipped with a rotary encoder or limit switch to detect gear position changes.
- **[REQ-029]** The system shall include a flaps lever for controlling flap positions.
- **[REQ-030]** The flaps lever shall use either a 10KΩ linear potentiometer or a rotary encoder.
- **[REQ-031]** The flaps lever shall provide smooth control of flap positions.
- **[REQ-032]** The system shall include a parking brake switch.
- **[REQ-033]** The parking brake switch shall be a heavy-duty toggle switch or a momentary push button.
- **[REQ-034]** The parking brake switch shall have mechanical locking functionality.
- **[REQ-035]** The system shall include warning indicator lights.
- **[REQ-036]** The system shall include caution indicator lights.
- **[REQ-037]** The warning lights shall use super bright red LEDs.
- **[REQ-038]** The caution lights shall use super bright yellow LEDs.
- **[REQ-039]** The indicator lights shall be mounted in an illuminated panel.
- **[REQ-040]** The system shall include a control panel with toggle switches and push buttons.
- **[REQ-041]** The control panel shall have labeled toggle switches.
- **[REQ-042]** The control panel shall have labeled push buttons.
- **[REQ-043]** The control panel shall simulate battery power control.
- **[REQ-044]** The control panel shall simulate alternator control.
- **[REQ-045]** The control panel shall simulate avionics master switches.
- **[REQ-046]** The system shall include a rotary encoder for adjusting heading.
- **[REQ-047]** The system shall include a rotary encoder for adjusting altitude.
- **[REQ-048]** The system shall include a rotary encoder for adjusting NAV radios.
- **[REQ-049]** The system shall include a rotary encoder for adjusting COM radios.

Hardware Integration and Interface Requirements

- **[REQ-050]** The system shall include a USB microcontroller board for interfacing physical components with MSFS 2020.
- **[REQ-051]** The system shall interface with MSFS 2020 via USB connection, using SimConnect for data exchange.
- **[REQ-052]** The system shall provide LED feedback signals.

Hardware Performance Requirements

- **[REQ-053]** The system shall process and register physical inputs within 100ms to prevent noticeable input lag.

Hardware Maintainability Requirements

- **[REQ-054]** The system shall include a self-test function upon startup.
- **[REQ-055]** The self-test function shall verify that all switches are operational.
- **[REQ-056]** The self-test function shall verify that all buttons are operational.
- **[REQ-057]** The self-test function shall verify that all LEDs are operational.
- **[REQ-058]** The system shall support future enhancements, including motorized components.
- **[REQ-059]** The system shall allow the addition of linear actuators for landing gear movement.

5. Non-behavioral Requirements

5.1. Performance Requirements

- **[REQ-060]** The GPS plugin shall update the displayed navigation data every 0.5 seconds.
- **[REQ-061]** The system shall initialize GPS data within 2 seconds when MSFS 2020 starts.
- **[REQ-062]** The system shall load GPS data within 2 seconds when MSFS 2020 starts.
- **[REQ-063]** The plugin shall use no more than 5% of available system memory to minimize performance impact on MSFS 2020.
- **[REQ-064]** The plugin shall operate without causing noticeable performance degradation in MSFS 2020.

5.2. Safety Requirements

- **[REQ-065]** The plugin shall not cause MSFS 2020 to crash under normal operation.
- **[REQ-066]** The plugin shall not disrupt MSFS 2020's core functions while running.

- **[REQ-067]** The system shall notify the user of data errors without disrupting the simulator.

5.3. Qualitative Requirements

5.3.1. Availability

N/A

5.3.2. Security

N/A

5.3.3. Maintainability

- **[REQ-068]** The system shall use modular code architecture.
- **[REQ-069]** The system shall allow easy updates without requiring major code changes.
- **[REQ-070]** The system shall support efficient debugging by isolating components.

5.3.4. Portability

N/A

5.4. Design and Implementation Constraints

- **[REQ-071]** The plugin shall adhere to MSFS 2020 SDK guidelines.
- **[REQ-072]** The GPS display shall support a screen resolution of 800x480.
- **[REQ-073]** The GPS display shall support a screen resolution of 1024x768.
- **[REQ-074]** The plugin shall be built using TypeScript for compatibility with MSFS 2020.
- **[REQ-075]** The plugin shall be built using TypeScript for compatibility with MSFS 2020.

6. Other Requirements

7. Analysis Models

7.1. Data Flow Model

7.1.1. Data Sources

N/A

7.1.2. Data Sinks

N/A

7.1.3. Data Dictionary

Name	Description	Units
GROUND_VELOCITY	Speed relative to the earth's surface.	Knots

7.1.4. Context Diagram (Level 0 Data Flow Diagram)

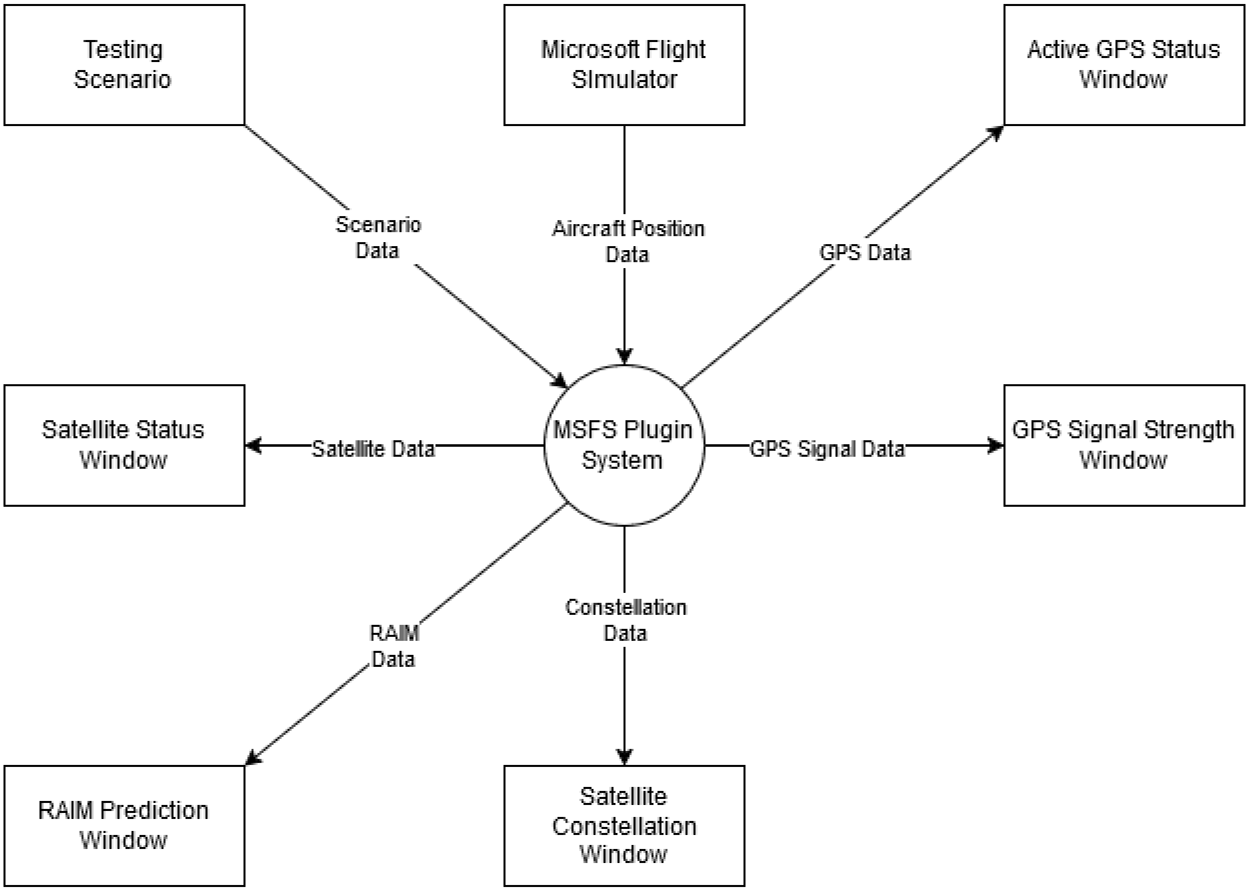

Figure 3: System Context Diagram

7.1.5. Level 1 Data Flow Diagram

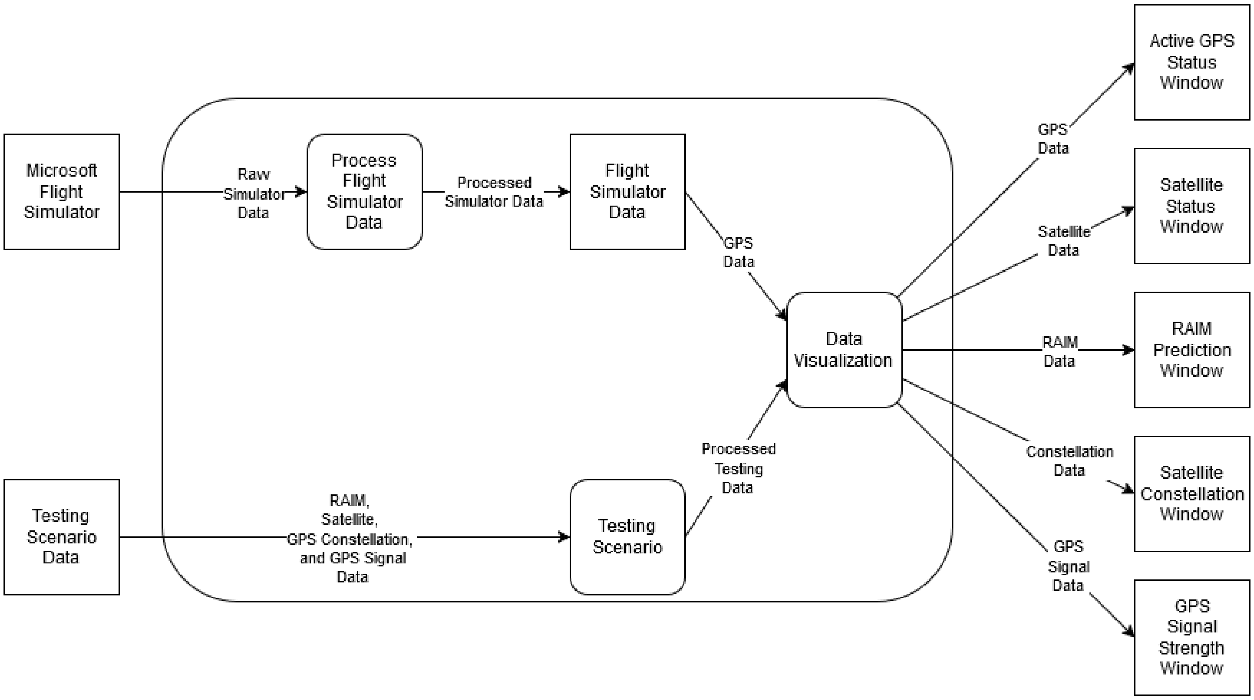

Figure 4: Level 1 Data Flow Diagram

This diagram represents the data flow of the Microsoft Flight Simulator plugin. It shows how the raw data from the simulator is processed and combined with testing scenario data to create GPS and flight information to be used. This processed data is fed into a visualization module which outputs all necessary data. This data is displayed on a module with multiple windows displaying different information for the end user.

7.1.6. Level 2 Data Flow Diagram

N/A

7.2. Class Model

N/A

7.3. State Model

N/A

8. SRS Section Assignment

This section outlines the requirements for the SRS assignment, focusing on the broader implications of engineering solutions within societal contexts. It emphasizes the importance of considering ethical and professional responsibilities and the long-term impacts of projects. The goal is to identify stakeholders and assess both positive and negative effects across various aspects, particularly global and environmental, which should be analyzed separately despite overlapping factors.

8.1. Group Section

The group will collaboratively identify and analyze stakeholders and evaluate the project's significance and impacts, focusing on global and environmental aspects.

8.1.1. Public Health

Improving flight safety is one of the key benefits of the GNSS Plugin, as it allows pilots to train for navigation and GPS use in a simulator, reducing the risk of real-world accidents caused by GPS failures. Training in a safe and controlled environment also helps lower stress and fatigue, since pilots spend less time in exhausting real-world flight scenarios. However, over-reliance on simulations could leave pilots less prepared for real-world challenges, making it essential to balance simulated training with actual flight experience.

8.1.2. Safety

Training with GPS failures and navigation errors in a simulator helps pilots practice handling emergencies without real-world risks. By simulating signal loss or interference, pilots can improve decision-making and problem-solving skills, giving them more confidence in real situations. Another benefit is that pilots can make mistakes and learn from them without the danger of crashing a real plane.

The importance of preventing crashes cannot be ignored. Recent aviation accidents have shown that proper training is critical to keeping flights safe. Saving lives is the top priority, and pilots must be fully prepared for navigation failures and unexpected problems. Training like this is not just helpful, it is necessary to improve aviation safety

Another growing issue is geo-spoofing, where GPS signals are hacked or manipulated, leading to navigation errors. These attacks are becoming more common, and pilots need to understand and respond to them. The GNSS Plugin is a useful tool for training pilots to recognize and handle these threats, making aviation safer.

While simulators are great for practice, they cannot fully replace real-world experience. Pilots must still train in actual flight conditions to be fully prepared.

8.1.3. Welfare

The GNSS Plugin helps expand access to flight training by allowing more people to learn navigation skills without the high cost of real-world flight hours. For students and aviation enthusiasts, it provides valuable GPS training, encouraging greater interest in aviation careers. Additionally, the plugin connects aviation and gaming communities, fostering collaboration, knowledge-sharing, and engagement. Despite these benefits, it cannot replace real-world flight experience, making it essential to use alongside traditional training methods for the best results.

8.2. Individual Section

Everyone will contribute a unique perspective by independently analyzing specific areas or impacts, further refining the assessment process.

8.2.1. Global Effect

Aviation training worldwide can benefit from the GNSS Plugin by making navigation practice more accessible and affordable. Pilots from different countries can learn GPS skills using MSFS 2020, even if they lack access to real-world training. A shared understanding of GPS systems could help improve safety practices and increase awareness of navigation risks like signal loss or spoofing. This may lead to more discussions on backup navigation methods within the aviation industry.

Since training rules differ by country, some regions may take longer to adopt simulation-based learning. However, flight simulators offer a cost-effective alternative to real-world training, as operating an aircraft for training can cost hundreds to thousands of dollars per hour, depending on fuel, maintenance, and instructor fees. Using MSFS 2020 with the GNSS Plugin provides a low-cost option for pilots to practice navigation skills, making training more accessible to students and professionals alike. While the plugin is a valuable tool, its impact will depend on how flight schools and aviation organizations choose to integrate it into their training programs.

8.2.2. Social

The implementation of our GNSS Plugin has the potential to provide several positive long-term social impacts, both of a direct and indirect nature. For example, in using this plugin, pilot training and safety awareness will be enhanced. Current and aspiring pilots alike using MSFS 2020 will obtain the ability to gain more realistic navigation training, therefore improving their understanding of GPS reliance and resilience.

Increased exposure to GPS failures and vulnerabilities, like geo-spoofing simulations, could help better prepare pilots and increase conversation about such challenges. The implementation of this plugin could also increase public awareness of GPS vulnerabilities, thus creating a larger social conversation among pilots and those in the aviation industry about them.

The inclusion of GNSS resilience features in a popular simulation tool like MSFS could bring about greater discussion about potential GPS failures, spoofing threats, and the importance of backup navigation systems, which encourages broader discussions about the topic. In addition, the inclusion of another tool to this simulation could encourage greater STEM interest among its users as well as higher levels of engagement, knowledge sharing, and collaboration among the gaming and aviation sectors.

However, potential negative impacts could arise from the interest of malicious actors attempting to understand and exploit GPS vulnerabilities by simulating geo-spoofing scenarios.

8.2.3. Environmental

The GNSS Plugin for MSFS 2020 can help the environment by reducing the need for real flight training, which uses fuel and creates pollution. By allowing pilots to practice navigation in a simulator, the plugin helps lower fuel use, aircraft wear, and air traffic.

Using flight simulators instead of real planes for training can also cut down on carbon emissions, helping the aviation industry become more eco-friendly. Since less fuel is burned, there is less air pollution and fewer resources used for aircraft maintenance.

However, running the simulator on computers still uses electricity, which can affect the environment. High-performance computers and gaming setups consume energy, and over time, outdated hardware may create electronic waste. To keep the plugin environmentally friendly, future improvements could focus on energy-efficient software and better use of sustainable technology.

Overall, this plugin is a greener option for training, but it is important to consider how electricity use, and computer waste can be reduced in the future.

8.2.4. Economic

Reducing training costs is one of the plugin’s biggest advantages. Pilots can practice without the high cost of fuel, aircraft maintenance, or extra flight hours, making flight training more affordable.

Flight schools and airlines may save money by incorporating simulators into their training programs. This allows pilots to build navigation skills while keeping expenses lower.

Encouraging interest in aviation and technology careers is another potential benefit. The plugin may inspire more people to explore STEM fields, leading to growth in aviation-related industries.

Despite these advantages, simulators cannot fully replace real flight experience. Hands-on training remains essential, so the plugin should be used alongside traditional training methods.

9. To Be Determined List

The following feature is postponed for future development:

- **Satellite Constellation and Visibility Simulation**
    - The plugin will display satellites statically, but they will not yet move based on orbital mechanics.
    - The radar system on top of the aircraft will not yet simulate an aperture cone to determine visible satellites.

This feature is not required for core functionality and will be revisited in a future update.