**Global Navigation Satellite System (GNSS) Resilience  
System Requirements Specification**

**Version <1.0>**

**10/24/2024**

# 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:

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 | 10/24/2024 | Jonathan Legro | Initial Document Creation |
| 1 | 10/28/2024 | Jonathan Legro | Additional Inputs |
|  |  |  |  |

Table of Contents

[Document Control 1](#_Toc1810101696)

[Distribution List 2](#_Toc982065305)

[Change Summary 2](#_Toc1856683173)

[1. Introduction 3](#_Toc1969089582)

[1.1. Purpose and Scope 4](#_Toc431412786)

[1.2. Intended Audience and Reading Suggestions 4](#_Toc1007677231)

[1.3. Document Conventions 4](#_Toc1442245508)

[1.4. Project References 4](#_Toc5023274)

[1.5. Definitions, Acronyms, and Abbreviations 5](#_Toc1420885672)

[1.5.1. Definitions 5](#_Toc671045334)

[1.5.2. Acronyms 5](#_Toc358592976)

[1.5.3. Abbreviations 5](#_Toc1308458984)

[2. General Description 5](#_Toc166979391)

[2.1. Product Perspective 6](#_Toc724745889)

[2.2. Product Features 6](#_Toc992828224)

[2.3. User Classes and Characteristics 6](#_Toc1572850549)

[2.3.1. Actors 7](#_Toc1310111570)

[2.3.2. Use Cases 7](#_Toc1111211826)

[2.3.3. Scenarios 8](#_Toc524468373)

[2.4. General Constraints 8](#_Toc1755679455)

[2.5. Operating Environment 9](#_Toc76119547)

[2.6. User Documentation 9](#_Toc1408740772)

[2.7. Assumptions and Dependencies 9](#_Toc1161442934)

[3. External Interface Requirements 9](#_Toc1015839265)

[3.1. User Interfaces 10](#_Toc10589518)

[3.1.1. Mobile User Interface 10](#_Toc800747447)

[3.2. Hardware Interfaces 12](#_Toc199214897)

[3.3. Software Interfaces 12](#_Toc1752423136)

[3.4. Communications Interfaces 12](#_Toc368049004)

[4. Behavioral Requirements 13](#_Toc1439260155)

[3.1. Same Class of User 14](#_Toc551461860)

[3.2. Related Real-world Objects 14](#_Toc889452848)

[3.3. Stimulus 14](#_Toc1218294484)

[3.3.1. Mobile App Stimulus 15](#_Toc2023559863)

[3.4. Related Features 15](#_Toc581908355)

[3.5. Functional 15](#_Toc898748655)

[5. Non-behavioral Requirements 16](#_Toc1431369685)

[5.1. Performance Requirements 17](#_Toc1583225492)

[5.2. Safety Requirements 17](#_Toc1779387589)

[5.3. Qualitative Requirements 17](#_Toc1417939841)

[5.3.1. Availability 17](#_Toc496653182)

[5.3.2. Security 18](#_Toc396145983)

[5.3.3. Maintainability 18](#_Toc290257564)

[5.3.4. Portability 18](#_Toc395306289)

[5.4. Design and Implementation Constraints 18](#_Toc1348039155)

[6. Other Requirements 19](#_Toc639841530)

[6.1. Database Requirements 20](#_Toc2054216260)

[6.2. Operations 20](#_Toc30412147)

[7. Analysis Models 20](#_Toc1696989394)

[7.1. Data Flow Model 21](#_Toc1801693198)

[7.1.1. Data Sources 21](#_Toc1480749716)

[7.1.2. Data Sinks 21](#_Toc671193407)

[7.1.3. Data Dictionary 21](#_Toc2005381272)

[7.1.4. Context Diagram (Level 0 Data Flow Diagram) 22](#_Toc1093067710)

[7.1.5. Level 1 Data Flow Diagram 22](#_Toc1531040053)

[7.1.6. Level 2 Data Flow Diagram 23](#_Toc1075351054)

[7.2. Class Model 23](#_Toc911859688)

[7.3. State Model 24](#_Toc972483852)

[8. To Be Determined List 25](#_Toc392593588)

# Introduction

## 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.

## 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 Developers & Players** – MSFS developers should focus on System Specifications and Technical Requirements to ensure smooth integration, while players may refer to the Overview and User Scenarios for gameplay insights.

Suggested reading for:

1. **Center for Aerospace Resilient Systems (CARS)**
   * **Sections**: Start with *1.1 Purpose and Scope* and *2. General Description* for an overview of project objectives, followed by *4. Behavioral Requirements* and *5. Non-behavioral Requirements* to understand functionality and performance criteria.
2. **Federal Aviation Administration (FAA) & Commercial Airlines**
   * **Sections**: Begin with *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. *5.2 Safety Requirements* is also valuable for assessing system resilience.
3. **MSFS 2020 Developers**
   * **Sections**: Developers should focus on *3. External Interface Requirements*, specifically *3.2 Hardware Interfaces* and *3.3 Software Interfaces* for compatibility details. *4. Behavioral Requirements* and *7. Analysis Models* will offer insights into system behavior and data flows.
4. **MSFS 2020 Players**
   * **Sections**: Players may focus on *1.1 Purpose and Scope*, *2.3 User Classes and Characteristics*, and *2.4 General Constraints* for a general understanding of the project’s purpose, limitations, and potential gameplay enhancements.

## 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.

## Project References

1. **Microsoft Flight Simulator**
   * **Title**: *Microsoft Flight Simulator*
   * **Author**: Wikipedia Contributors
   * **Source**: Wikipedia
   * **Date**: Accessed 2024
   * **Location**: <https://en.wikipedia.org/wiki/Microsoft_Flight_Simulator>
   * *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**: <https://www.typescriptlang.org/>
   * *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**: <https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/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**: <https://www.advancednavigation.com/tech-articles/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**: <https://docs.flightsimulator.com/html/Introduction/Introduction.htm>
   * *Description*: The official SDK documentation for MSFS 2020, essential for understanding development standards and integration guidelines.

## Definitions, Acronyms, and Abbreviations

### 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. |
| Developer | Builds and maintains the GPS plugin. |
| Researcher | Tests GNSS resilience using the plugin. |
| Simulator | Imitative representation of a process or system. |
| Actors | All users of the system. |

### 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 |

### Abbreviations

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

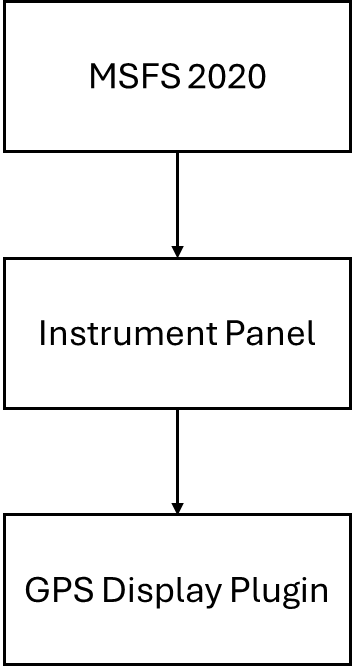
**Table 3: Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| e.g. | For example |

# General Description

## 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.



## 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).

## User Classes and Characteristics

### 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.
* **Developer**
  + A Developer is responsible for integrating, updating, and maintaining the GPS display plugin within MSFS 2020.
* **Researcher**
  + A Researcher uses the system to study GNSS resilience and conduct aviation safety testing, simulating potential vulnerabilities in a controlled environment.

Each actor has specific interactions with the system, ranging from regular use to research and development tasks.

### 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.
* **Conduct Research**
  + *Researchers* simulate GNSS vulnerabilities for aviation resilience studies.
* **Maintain System**
  + *Developers* keep the plugin updated and compatible with MSFS 2020.

### Scenarios

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

Scenario 1: Enhance Flight Simulation

* **Actors**: End User
* **Trigger**: User opens the GPS display.
* **Steps**:
  1. GPS interface loads.
  2. User sets waypoints.
  3. Map updates with navigation data.

Scenario 2: Conduct Research

* **Actors**: Researcher
* **Trigger**: Researcher starts a resilience test.
* **Steps**:
  1. GPS loads simulated data.
  2. Researcher applies interference.
  3. GPS display shows response to interference.

Scenario 3: Maintain System

* **Actors**: Developer
* **Trigger**: Developer updates the plugin.
* **Steps**:
  1. Accesses code.
  2. Makes updates.
  3. Tests and reintegrates plugin.

## General Constraints

1. **Organizational**: Must align with CARS’ requirements and MSFS 2020 compatibility.
2. **Hardware**: Requires gaming PCs that meet MSFS 2020 specs.
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.

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

## Operating Environment

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

* **Operating System**: Windows 10 or higher.
* **Hardware**: Gaming PCs meeting MSFS 2020’s requirements.
* **Software**: Integrated within MSFS 2020, using TypeScript and HTML.
* **Display**: Standard monitors or VR, for indoor use in typical room lighting and temperature.

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

## User Documentation

1. **User Manual**: Guide for installation, setup, and usage.
2. **Online Help**: In-app tips and troubleshooting.
3. **Tutorials**: Step-by-step feature guides.

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

## 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.

# External Interface Requirements

## User Interfaces

*<< Characteristics of each interface between the system and its users. This should specify the following:*

1. *The logical characteristics of each interface between the software system and its users. This includes those configuration characteristics (e.g. required screen formats, page or window layouts, content of any reports or menus, or availability of programmable function keys) necessary to accomplish the system requirements.*
2. *The logical characteristics of each interface between the hardware system and its users. This includes those configuration characteristics (layout of buttons, dials, screens, switches, or other hardware for interacting with the user) necessary to accomplish the system requirements.*
3. *All the aspects of optimizing the interface with the person who must use the system. This may simply comprise a list of do’s and don’t on how the system will appear to the user. One example may be a requirement for the option of long or short error messages. Another example may be a requirement for the inclusion of a safety cover on a particular switch. Like all others, these requirements should be verifiable, for example, “A clerk typist grade 4 shall be able to do function X in z minutes after the introductory training session” rather than “A typist shall be able to do function X.”*

*Examples:*

*[REQ-3] The system shall gray-out and disable any visible GUI elements that are not applicable to the context of the functionality being performed. See Appendix F for settings.*

*[REQ-4] Hover text shall be displayed in sentence case.*

## Hardware Interfaces

**N/A for now** – This system is currently software-only; no hardware interfaces required.

## Software Interfaces

*<< This should specify the use of other required software products (for example, a data management system, an operating system, or a mathematical package), and interfaces with other application systems (for example, the linkage between an accounts receivable system and a general ledger system). For each required software product, the following should be provided:*

1. *name,*
2. *mnemonic,*
3. *specification number,*
4. *version number, and*
5. *source >>*

## Communications Interfaces

*<< This should specify the various interfaces to communications such as local network protocols, etc. >>*

# Behavioral Requirements

*<< This template illustrates the organization of functional requirements for the product by requirement category. You may prefer to organize this section instead by use case, mode of operation, user class, object class, functional hierarchy, or combinations of any of these. Choose the format that makes the most sense for your system.* *>>*

## Same Class of User

N/A for now – No user access levels or privileges are defined yet. Requirements may be added later based on user roles.

## Related Real-world Objects

N/A for now – No specific real-world objects or models required. May be updated if user needs expand.

## Stimulus

*<< Some systems (e.g., real-time systems) can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections that include the following loss of power, wind shear, and sudden change in roll. The state diagram, event diagram, or other dynamic model is included in this section. >>*

*Examples:*

*[REQ-71] If a workspace file is not specified in the Metafile table for the scene being opened, the system shall set the parameters for each window as specified in Table F-1 in Appendix F to their corresponding default values when the scene is opened.*

*[REQ-72] If a workspace file is loaded when a scene is first opened, the system shall set the default values to those specified in the file.*

## Related Features

*<< A feature is an externally desired service provided by the system that may require a sequence of inputs to affect the desirable result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs (in such a case, Section 4.3 could be changed to a subsection of this section. The use cases form the outline of this section. >>*

## Functional

*<< Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These include:*

1. *validity checks on the inputs,*
2. *exact sequence of operations, and*
3. *responses to abnormal situations, including:*
   1. *overflow,*
   2. *communication facilities, and*
   3. *error handling and recovery.*

*In addition, data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data. Refer to data flow diagram in this section.*

*Example:*

*[REQ-89] The default settings for the display windows shall be as specified in Table F-1 in Appendix F. >>*

# Non-behavioral Requirements

## Performance Requirements

N/A – No specific requirements yet.

## Safety Requirements

N/A – No safety requirements currently.

## Qualitative Requirements

### Availability

N/A – No availability needs defined.

### Security

N/A – No security requirements yet.

### Maintainability

N/A – No maintainability needs specified.

### Portability

N/A – No portability requirements for now.

## Design and Implementation Constraints

N/A – No constraints currently set.

# Other Requirements

## Database Requirements

N/A for now – No database requirements currently needed. This may be updated if data storage or retrieval becomes necessary.

## Operations

N/A for now – No specific operational requirements defined at this stage. Operational guidelines may be added as the project develops.

# Analysis Models

## *Data Flow Model*

N/A for now – No data flow model defined. Diagrams and data handling requirements may be added if data processing needs arise.

### *Data Sources*

### *Data Sinks*

### *Data Dictionary*

### *Context Diagram (Level 0 Data Flow Diagram)*

### *Level 1 Data Flow Diagram*

### *Level 2 Data Flow Diagram*

## *Class Model*

N/A for now – No class model defined. A class structure may be developed if the system requirements expand.

## *State Model*

# To Be Determined List

*<< Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure. >>*