sUAS Safety Case Tool
Software Requirements

TEAM: GRP_3_AERIAL
AUTHORS: Brady Bargren, Blake Bryan, Benjamin Kelly,
Gabriel Perez, and Gautham Suresh

December 16, 2024

Version History

Version	Date	Author	Change
0.1	December 16, 2024	Brady Bargren	Initial draft of SRS

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to establish the functional and non-functional requirements for the sUAS Safety Case Tool, a web-based application designed to automate the generation of safety cases for small Uncrewed Aerial Systems (sUAS). This document defines the problem, describes the system design, and provides details on the project's technical architecture.

1.2 Scope

The sUAS Safety Case Tool will reduce risks associated with operating sUAS by generating safety cases programmatically. The application will address hardware, software, and environmental risks, providing pilots with a formalized mechanism to evaluate the readiness of their vehicles for safe operation in shared airspace. It also supports compliance with potential regulatory requirements by maintaining safety assessment records.

1.3 Definitions, Acronyms, and Abbreviations

Term	Description
sUAS	Small Uncrewed Aerial Systems
GSN	Goal Structuring Notation, a framework for visualizing safety cases
YAML	A data serialization language used for safety case representation
NOTAM	Notice to Air Missions, real-time aviation information

1.4 References

- FAA's Flight Risk Assessment Tool (FRAT)

- MongoDB documentation: `https://www.mongodb.com/docs/`

- React.js documentation: `https://reactjs.org/`

1.5 Overview

The document includes the system's overall description, functional and non-functional requirements, technical architecture, and team contributions. It outlines the project's role in mitigating risks associated with operating sUAS.

2 Overall Description

2.1 Product Perspective

The sUAS Safety Case Tool is a web application combining a dynamic frontend interface, a robust backend server, and a database for persistent storage. The system automates safety case generation based on user inputs, such as vehicle condition, pilot credentials, and environmental conditions.

2.1.1 Concept of Operations

- Pilots create an account and answer dynamically generated questions about their flight plan.

- Safety cases are generated in YAML format and visualized using GSN diagrams.

- Risk assessments are performed to categorize flights as low-risk or high-risk.

- The system stores safety cases for future retrieval and regulatory compliance.

2.1.2 Major User Interfaces

The tool features a user-friendly React.js interface, with sections for answering questionnaires, viewing generated safety cases, and retrieving previous assessments. The interface is optimized for clarity and usability.
Screen Flow:
Question Form (Top) (Figure 1):

Figure 1:

Question Form (Bottom) (Figure 2):

Figure 2:

Generated Safety Cases (Safety Case Pass/Fail Tree) (Figure 3):
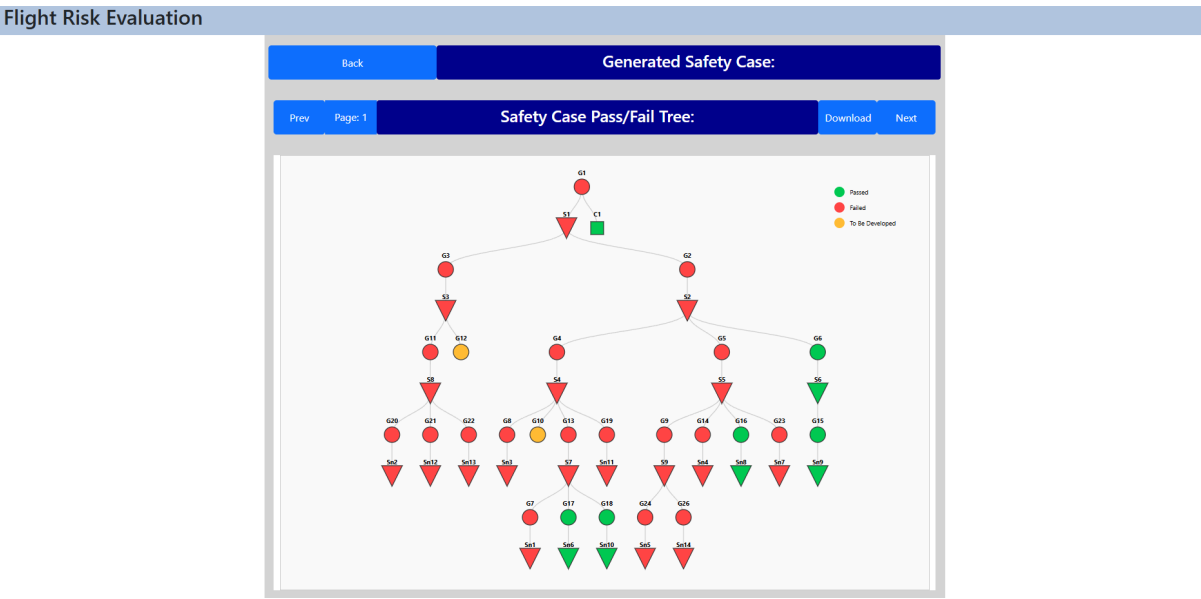

Figure 3:

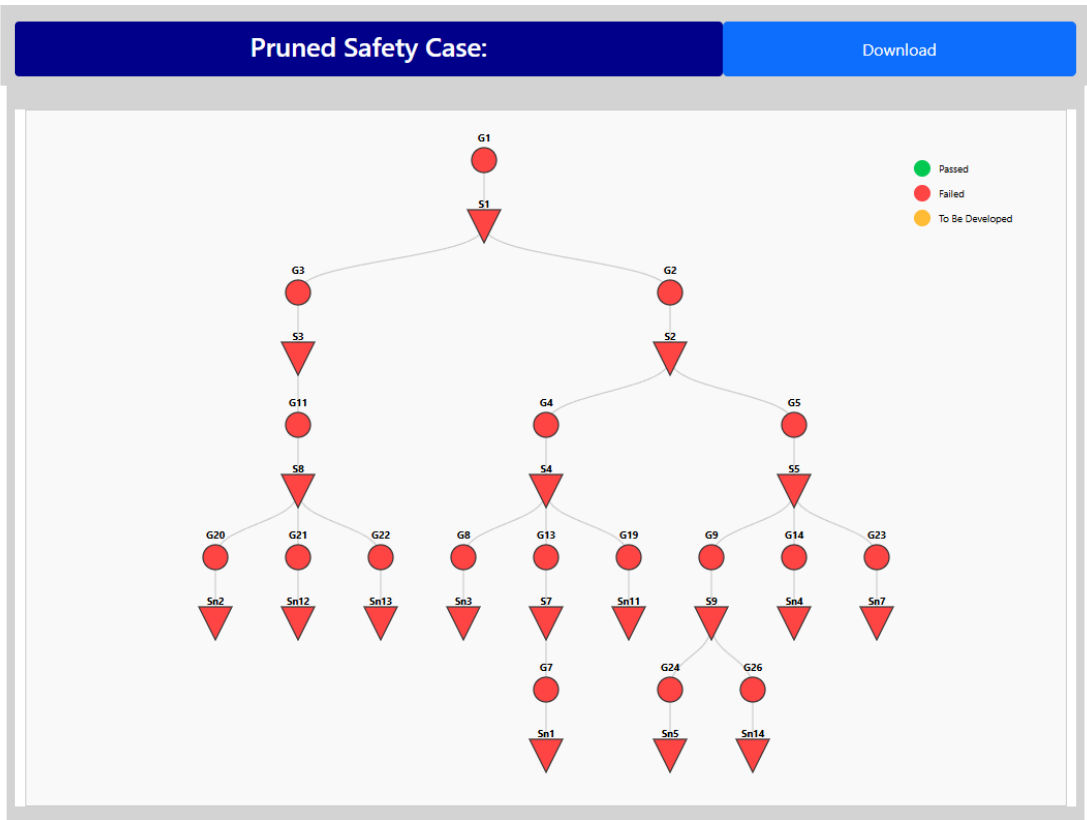**Generated Safety Cases (Safety Case Pass/Fail Tree) (Figure 4):**

Figure 4:

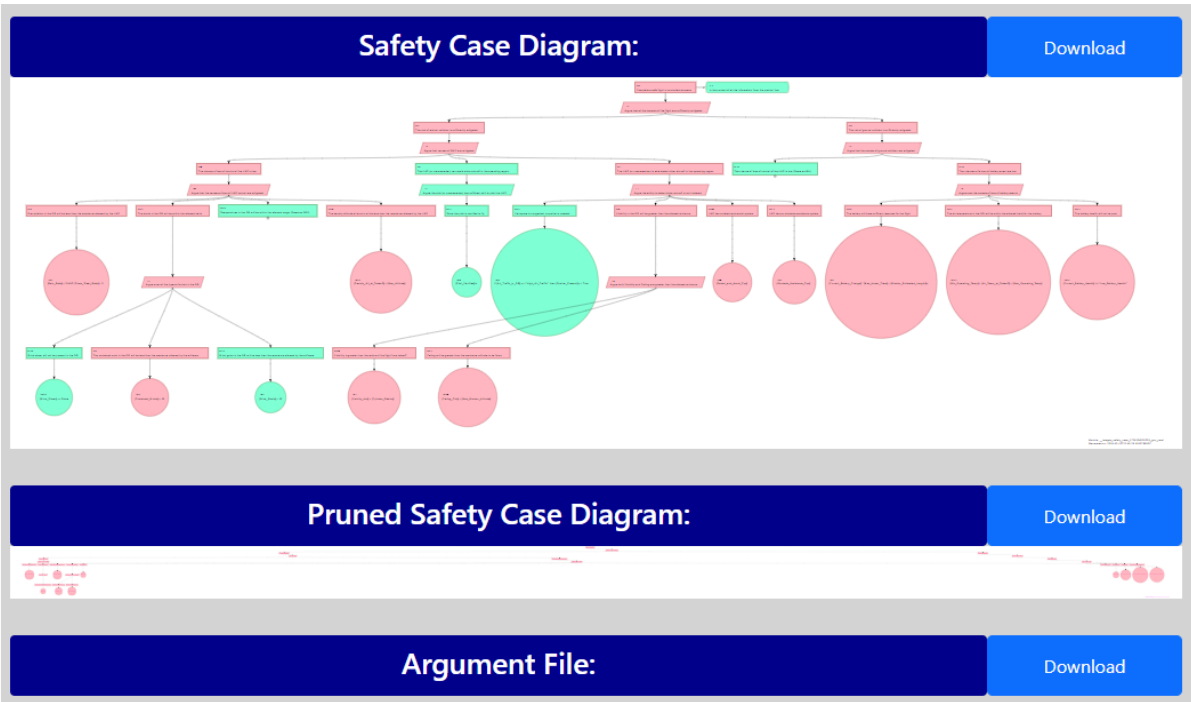**Generated Safety Cases (YAML Versions/Argument File) (Figure 5):**

Figure 5:

Survey Results (Figure 6):

Figure 6:

NOTAMs (Figure 7):

Figure 7:

Administrative Page (Figure 8):

Figure 8:

2.1.3 Hardware Interfaces

Any device with a modern web browser supporting HTML5, CSS3, and JavaScript.

2.1.4 Software Interfaces

- Frontend: React.js with TypeScript

- Backend: Node.js with Express

- Database: MongoDB

- Parsing libraries: xml2js, js-yaml

2.1.5 Communication Interfaces

The application communicates with external APIs, such as NOTAM services, and performs client-server interactions over HTTP.

2.1.6 Memory Constraints

The system uses efficient in-memory operations for dynamic form generation and risk assessments.

2.1.7 Site Adaptation Requirements

The system supports multiple user roles (pilots, advisers, administrators) and accommodates localization for different regulatory environments.

2.2 Product Functions

- Dynamic questionnaire generation based on XML flight models.

- Safety case visualization using GSN diagrams.

- Risk assessments with color-coded diagrams.

- Storage and retrieval of historical safety case data.

- Integration with NOTAMs for real-time aviation updates.

2.3 User Characteristics

Target users include:

- Pilots: To assess flight safety and compliance.

- Advisers: To guide pilots in creating safe flight plans.

- Administrators: To manage system configurations and thresholds.

2.4 Constraints

- Real-time data retrieval from NOTAM APIs may be limited by external API access.

- The system must maintain compatibility with future aviation regulations.

- Memory usage must be optimized for large-scale XML and YAML files.

2.5 Assumptions and Dependencies

- Users have internet access and modern devices.

- MongoDB and Node.js are installed on the server.

3   Specific Requirements

3.1   Features

3.1.1   Dynamic Questionnaire Generation

- Parses XML flight models to generate tailored questions.

- Adapts dynamically to different sUAS configurations.

3.1.2   Risk Assessment and Safety Case Generation

- Assigns risk levels based on user input.

- Generates YAML files and visualizes safety cases using GSN.

3.1.3   NOTAM Integration

- Fetches real-time airspace updates.

- Displays NOTAM data within the application.

3.2   Performance Requirements

- Generate safety cases within 5 seconds for typical inputs.

- Support at least 100 simultaneous users.

3.3   Design Constraints

- The system must use MongoDB for database operations.

- Safety cases must adhere to GSN standards.

3.4   Software System Attributes

3.4.1   Reliability

- Ensure 99.9% uptime for critical features.

3.4.2   Availability

- The system should be accessible 24/7.

3.4.3   Maintainability

- Modular codebase for easier updates.

3.4.4   Portability

- Support deployment on cloud platforms like AWS or Azure.

Appendix

- Demonstration video: `https://youtu.be/DIim2_m4Qaw?si=pD5d5uUXd5JJXZwp`

- YAML template file: Available in the `templates/` directory.