Software Requirements Specification

for

SOApy

Version 1.0 approved

Prepared by: Aaron Simental, Andrew Jarmin, Cesar Salazar, Nathan Gonzales, Pedro Ramirez, Richard Bailon, Scott Sun, William Leung, Xico Blanco, Yuridia Ginez

CSULA/ The Aerospace Corporation

December 9th, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Richard Balion	12/05/2022		Draft
Pedro Ramirez	12/09/2022		Final

1. Introduction

1.1 Purpose

The purpose of this document is to give a high level overview of the SOApy workings. This will include Docker, API requests, Data formatting techniques, accessing and manipulating Influx Databases, and Data visualization using Pythons Django framework as well as matplotlib to allow the user to extract knowledge from all the data.

1.2 Intended Audience and Reading Suggestion

This document is intended for SOApy users, Aerospace stakeholders, as well as future developers and application testers. The SRS will serve as a starting point for future developers and testers who need to have a greater understanding of the SOApy program in order to contribute to the project or perform the necessary tests.

1.3 Product Scope

SOApy is intended to automate Aerospace's GNSS (Global Navigation Satellite Systems) constellation metrics workflow, more specifically DOP (Dilution of Precision). Currently, calculating DOP takes subject matter experts (lots of time) to calculate DOP using pre-existing methods, like SOAP. SOApy's automation will take care of finding the satellite's TLE data from a third-party service. Formatting those TLEs into a .orb, a SOAP readable file, which then calculates a .dtt file, which contains the DOP values in a grid format. The DOP values are then saved in a time series database and visualized using a web application. Once completed SOApy will be uploaded to a private GitHub repository with developers having clone and branch permission, testers and verified users will have clone permissions.

1.4 Definitions, Acronyms, and Abbreviations

SOAP- (Satellite Orbit Analysis Program) is used heavily within Aerospace to manually build and analyze satellite scenarios. Capable of performing more than 150 different analysis types.

DOP- (Dilution of Precision) is a geometric measure of GPS accuracy.

PDOP- Position Dilution of Precision

TLE - (Two Line Element) is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time.

Docker - A platform to package software to make it capable of running on any software on any machine.

Epoch - A moment in time used as a reference point

API - (Application Programming Interface) is a set of definitions and protocols to build and integrate application software.

API Query - A generic query component to read data from JSON and XML-based APIs.

DockerFile - A blueprint for creating a docker image.

Image - Template with instructions to create Docker containers.

Container(s) - Standalone, executable software package which includes applications and their dependencies.

.orb - A file extension that contains all the data for SOAP to produce a scenario.

.dtt - A file extension that contains all PDOP values.

InfluxDB - A database platform that allows to store and manage large volumes of time series data.

Grafana - A web visualization tool that can be used to analyze and visualize data from many databases.

Geomap - allows users to view and customize the world map using geospatial data.

Django - a framework that enables rapid development of secure and maintainable websites.

1.5 References

"Django." *Django Project*, docs.djangoproject.com/en/4.1.

"Docker Overview." *Docker Documentation*, 8 Dec. 2022, docs.docker.com/get-started/overview.

"Grafana Documentation | Grafana Documentation." *Grafana Labs*, grafana.com/docs/grafana/latest.

InfluxDB OSS 2.5 Documentation. docs.influxdata.com/influxdb/v2.5.

SAIC admin@space-track.org. "Space-Track.Org." *Space-Track*, www.space-track.org/documentation.

2. Overall Description

2.1 System Analysis

The goal of SOApy is to provide an automated, real-time analysis of a scenario. The capability to analyze satellites' status, behavior, and performance is a critical task that is very valuable to Aerospace stakeholders. The goals kept in mind were to create a user-friendly Web App to make sure it is easy for users to navigate, as opposed to the SOAP GUI. Other goals were to format an orb file using a template in order to create the output .dtt file, storing those PDOP values into InfluxDB, creating a visualization of the data on Grafana based on the user input from the Web App, as well as getting data from the Space Track API.

The major technical hurdles while trying to complete SOApy to its current development state were figuring out how to get started and implementing brainstormed ideas once they were approved by the team. Some of these ideas were how to use a template to format an orb file using python and how to store the longitude, latitude, and PDOP values in InfluxDB. Other hurdles we had to overcome were formatting the correct API query to get the desired TLEs, and creating a user-friendly Web App along with visualization of the PDOP values into a map.

Solutions that helped overcome the technical issues were to read the documentation and really understand how each programming language worked. Understanding our individual group goals to the smallest detail helped too. Each group understands what their end result should be and how it should piece together with the other groups. It is important to test our output to see if the results are what is desired in order to keep the order of the pipeline flowing.

2.2 Product Perspective

SOApy is a completely dependent web app that relies on the product SOAP (Satellite Orbit Analysis Program). SOApy can relate to other products that share similar interests with data visualization, web app GUI, satellite orbits, and analysis. There are similarities comparing SOAP and SOApy although SOApy will have its own GUI and be set specifically to output a data analysis on PDOP.

2.3 Product Functions

The function of this product is to visualize PDOP values analyzed through SOAP from a list of TLEs (satellites).

Main Functions -

- Input a list of Satellites, and EPOCH time
- Calculate PDOP values using containerized SOAP
- Display data of satellite information / PDOP value visualization
- "Dashboard Function" - displays a certain amount of graphs/visualizers
- "New Query Function" - be able to query a new set of satellites
- "History Function" - be able to see previous satellite queries

2.4 User Classes and Characteristics

The main users for this product will be Aerospace stakeholders and employees, frequently used for those looking for specific PDOP values for specific satellites. This product will mainly be used for those with security and privilege levels from the Aerospace Corporation.

2.5 Operating Environment

The operating systems used to develop the project:

- Windows 10
- MacOS

Software used:

- SOAP
- Docker
- Grafana
- InfluxDB

2.6 Design and Implementation Constraints

- **Windows:**
  - Hyper-V must be enabled for Docker
- **Safety and Security Considerations:**
  - Avoid uploading Docker images to Docker Hub
  - Upload only to private Git repositories

2.7 User Documentation

- **Software Requirements Specification**
  - PDF Format
- **Software Design Document**
  - PDF Format
- **User Manual**
  - Text File Format

2.8 Assumptions and Dependencies

- Users will be able to view and interact with data through the SOApy web application.
- The satellite's TLE data will be requested from the third-party component, Space-Track.
- The DOP data stored in the InfluxDB database and visualized on the web application shall display appropriately.

2.9 Apportioning of Requirements

The following requirements will be delayed and may be implemented in the future development process.

- The SOApy web application to visualize the data will be delayed. The web application is still a work in progress.
- Dockerizing the backend process

3. External Interface Requirements

3.1 User Interfaces

The SOApy web application will be developed using Django with various panels that will be implemented using Grafana which will allow users to interact with multiple dashboards that will visualize the data. Through the use of Grafana and InfluxDB, the panels will display DOP data. The web application shall be implemented using Django and then Python for the backend.

- The SOApy web application shall display DOP data visualized on a world map, satellite orbits, and satellite data.
- The SOApy web application shall display the data within panels as illustrated in Figure 1.
- The SOApy web application shall contain features that allow the user to interact with it. These features include a “Dashboard”, “New Query”, and a “History” button.

Web App Blueprint:

- The web application is currently a work in progress, but Figure 1 is a blueprint of how the team wants to design and develop the SOApy web app.

Fig. 1 (Web App. Diagram)

3.2 Hardware Interfaces

- All that’s needed is a computer screen and internet connection

3.3 Software Interfaces

- Docker version 20.10.20
- Space Track API
- Grafana

3.4 Communications Interfaces

- An Internet connection of any kind is all that is needed

4. Requirements Specification

4.1 Functional Requirements

4.4.1 Process 1 shall make a call to Space Track API, and output a file containing TLE's

4.4.2 Process 2 shall use the output of Process 1 to generate a SOAP scenario and output a DTT file containing SOAP analysis

4.4.3 Process 3 shall use the output of Process 2 and store contents of that file into InfluxDB

4.4.4 Process 4 shall query from Influxdb and output visualizations that will make analysis easy to understand

4.2 External Interface Requirements

4.2.1 Docker provides easy building, testing, and deployment of application

4.2.2 Space Track provides TLE information of the Global Positioning System

4.2.3 Grafana is a web visualization software which shall provide visualizations to data that has been stored in InfluxDb

4.2.4 InfluxDB Time Series Database which shall be used to store TLE data

4.3 Logical Database Requirements

4.3.1 The source of the information that will be placed into InfluxDb is a dtt file

4.3.2 Dtt file is the analysis output from SOAP

4.3.3 Type of information being stored in InfluxDB include a Geographical location in the form of longitude and latitude, as well as a PDOP value

4.4 Design Constraints

4.4.1 The application shall accommodate different platforms including Windows and macOS.

4.4.2 The SOAP image should be preinstalled. The request process to download SOAP shall be involved signing an NDA with Aerospace

4.4.3 The limitation of requesting satellite data per user from space-track.org shall be considered.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 The application shall stay online in order to access the online database and perform real-time user requests.

5.1.2 API queries to space-track.org shall be less than 30 requests per minute / 300 requests per hour, which might affect SOApy's performance for user requests.

5.2 Safety Requirements

5.2.1 Users shall check the SOAP CCATS G164467 Export Briefing document before distributing the software involving SOAP since the software package contains export-controlled data defined by the Export Administration Regulations(EAR). If users violate, ITAR and EAR Violations and Penalties shall be involved.

5.3 Security Requirements

5.3.1 Users will require SOAP access through the authentication of the Aerospace program where they will receive their user identity authentication requirements.

5.3.2 Access to Space-Track will also be needed with login credentials being essential for the api script to process data

5.4 Software Quality Attributes

5.4.1 The API script is adaptable with User input to process data for any given date within the correlated range. The flexibility of this will allow users to specify individual data or a mass array of data. Users will be able to re-use the software on a constant basis.

5.5 Business Rules

5.5.1 The average user will be able to simply perform basic information searches while specified individuals may be given much more authority over the product or access to key functionality options.

6. Legal and Ethical Considerations

Based on the many legal/ethical considerations of the project, one of the decisions we made to move forward with the project was to meet every Wednesday to catch up on the progress of each individual group. This decision was made so that we can have a middle point when it comes to seeing what people have worked on when compared to just meeting once a week on Fridays. Another Ethical consideration was having to request/access the API and software used by Aerospace. Since their data is confidential, we had to request access to the software (SOAP) and use access codes for each individual teammate.

Appendix A: Glossary

SOAP- (Satellite Orbit Analysis Program) Used heavily within Aerospace to manually build and analyze satellite scenarios. Capable of performing more than 150 different analysis types.

DOP- (Dilution of Precision) is a geometric measure of GPS accuracy.

PDOP- Position Dilution of Precision

TLE - (Two Line Element) is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time.

Docker - A platform to package software to make it capable of running on any software on any machine.

Epoch - A moment in time used as a reference point

API - (Application Programming Interface) is a set of definitions and protocols to build and integrate application software.

API Query - A generic query component to read data from JSON and XML-based APIs.

DockerFile - A blueprint for creating a docker image.

Image - Template with instructions to create Docker containers.

Container(s) - Standalone, executable software package which includes applications and their dependencies.

.orb - A file extension that contains all the data for SOAP to produce a scenario.

.dtt - A file extension that contains all PDOP values.

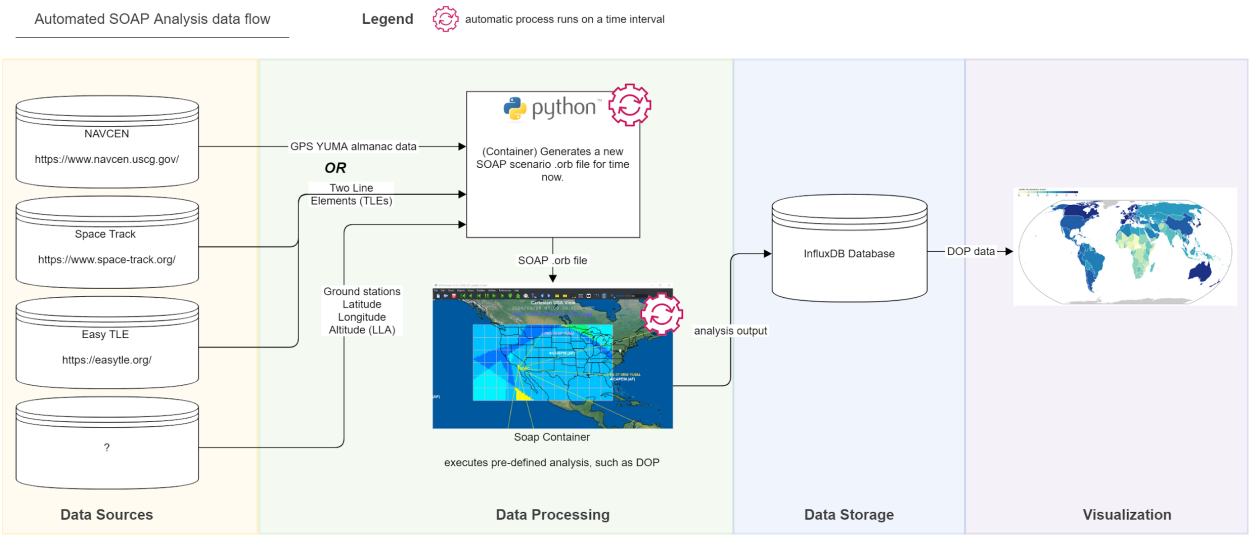
InfluxDB - A database platform that allows to store and manage large volumes of time series data.

Grafana - A web visualization tool that can be used to analyze and visualize data from many databases.

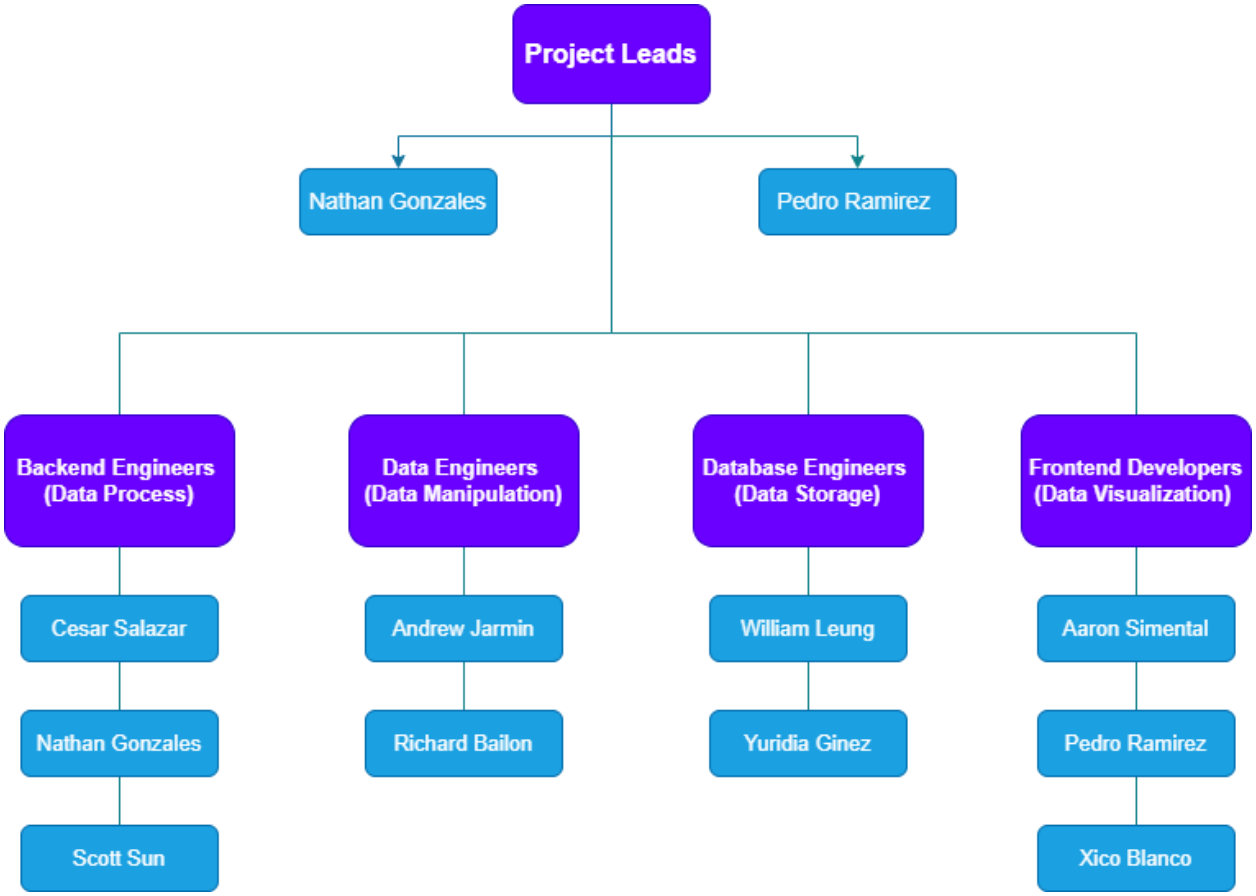
Geomap - allows users to view and customize the world map using geospatial data.

Django - a framework that enables rapid development of secure and maintainable websites.

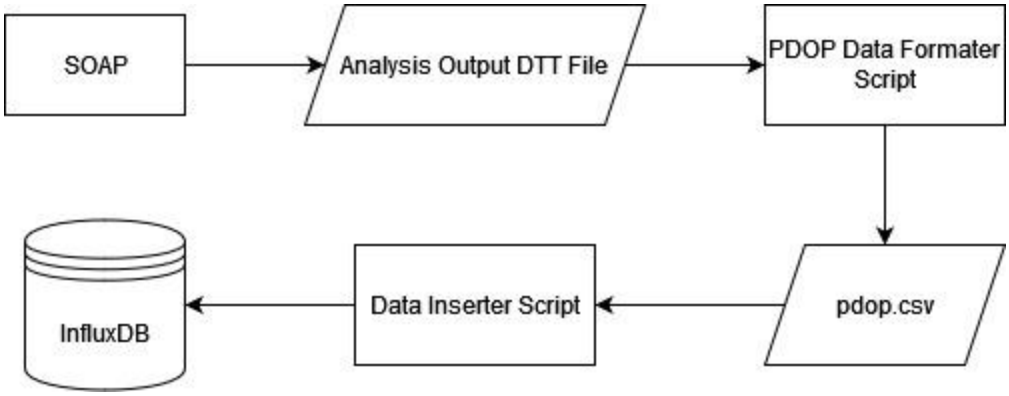
Appendix B: Analysis Models



Appendix B: Fig. 1 (SOApy Pipeline)



Appendix B: Fig. 2 (Organization Tree)



Appendix B: Fig. 3 (TLE Data Flow)

Appendix C: To Be Determined List

- GUI Connection within the user input
- API User Input
- Containerizing the TLE data
- Soapy web application