Software Requirements Specification

for

Displaying DOP Data

Version 2.0 approved

Prepared by John Chen, Phong Diep, Matthew Franco, Roaf Htun, Steven Lopez Carachure, Andrew Mendiola, Natalie Perales, Cindy Rodriguez, Steven Yuen

The Aerospace Corporation

5/10/24

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Version 1	9/22/23	Created UI	1.1
Version 2	10/13/23	Connected BackEnd and Database	1.2
Version 3	11/03/23	Generated Heat Map	1.3
Version 4	11/17/23	General improvements all around	1.4
Version 5	12/1/23	Finishing touches for the fall semester	1.5
Version 6	3/22/24	Connected FrontEnd Database and BackEnd	1.7
Version 7	4/5/24	FrontEnd touch ups	1.9
Version 8	5/1/24	Finishing touches for Spring semester	2.0

1. Introduction

.This document outlines at a high level the computer science senior design project for the ’23-’24 school year. The audience for this document is both the CSULA faculty and the senior design project student team. The role of the student team is to generate requirements and implement a software solution based on the feedback from Aerospace, the product owner.

1.1 Purpose

The purpose of this document is to outline the software requirements for the backend system of the DOP Data Processing and Storage project. This system is responsible for web crawling, data processing, time conversion, and storing processed data in an InfluxDB database.

1.2 Intended Audience and Reading Suggestions

This document is intended for the backend development team, the InfluxDB team, the frontend team, and project stakeholders. It provides a detailed description of the system's functionality and constraints.

1.3 Product Scope

The backend system will automate the collection of updated data from the USCG website, calculate DOP data, convert GPS time to Unix timestamp, and store the results in an InfluxDB database for further use by other teams.

1.4 Definitions, Acronyms, and Abbreviations

DOP: Dilution of Precision

USCG: United States Coast Guard

Unix Timestamp: A system for describing points in time, defined as the number of seconds elapsed since January 1, 1970

SRS: Software Requirements Specification

GPS: Global Positioning System

2. Overall Description

2.1 System Analysis

The system will be a crucial part of a larger infrastructure designed to provide accurate and timely DOP data for visualization and analysis.

2.2 Product Perspective

The backend system functions as an independent module but is integral to the overall project, interfacing with the InfluxDB database and providing data to the frontend visualization team.

2.3 Product Functions

The web application will fetch data, calculate necessary data and store them in InfluxDB. The web application will query data needed to visualize the map to display the user.

2.4 User Classes and Characteristics

Regular Web App Users:

- Frequency of Use: Infrequent users who access the system occasionally.
- Product Functions: Primarily interested in basic data retrieval and DOP calculation.
- Technical Expertise: Moderate technical proficiency.
- Security/Privilege Levels: Standard user privileges, with limited access to advanced features.
- Educational Level/Experience: Basic understanding of satellite navigation principles.
- Importance: Important for broadening the user base; their needs influence user interface simplicity.

Developers:

- Frequency of Use: Occasional, focused on system maintenance and troubleshooting.
- Product Functions: Manage user accounts, monitor system logs, and ensure system integrity.
- Technical Expertise: High technical proficiency in system administration and security.
- Security/Privilege Levels: Elevated privileges for system configuration and maintenance.
- Educational Level/Experience: Advanced technical education or equivalent experience.
- Importance: Critical for ensuring system reliability, security, and performance.

2.5 Operating Environment

The web application will be on the cloud infrastructure.

2.6 Design and Implementation Constraints

- The development process may be impacted by organizational or industry-specific regulations and policies.
- The software's functionality may be constrained by the capabilities and limitations of the hardware it runs on.
- Compatibility and interoperability considerations become crucial to enable effective communication between different software components.

2.7 User Documentation

The user will have an easy-to-understand interface on the web application, which will help the user to understand the purpose of the application.

2.8 Assumptions and Dependencies

Assumptions:

- Operating System: It is assumed that the targeted users can access a compatible operating system (e.g., Windows, Linux, macOS) that supports the required software dependencies.
- Internet Connectivity: Users are assumed to have internet connectivity to access navcen.com for data retrieval.
- Stable Navcen Website: The software assumes that the structure of the Navcen website remains relatively stable for web scraping operations.

Dependencies:

- Third-Party Libraries: The project may depend on third-party libraries such as Beautiful Soup for web scraping, InfluxDB Python client for database interaction, and other relevant libraries.
- Web Technologies: The project may depend on specific web technologies, such as HTML structure or JavaScript, for webpage interactions during data-grabbing.
- Python Runtime: The software assumes the availability of a compatible Python runtime environment on the user's machine for running the application.
- InfluxDB: The software depends on the availability and proper functioning of a locally hosted InfluxDB for data storage.

- Security Tools: Dependencies on security tools or practices for securing user data during transmission and storage.

Reusable Components:

- Algorithm Libraries: If applicable, consider reusing existing algorithm libraries or components for DOP calculations.
- User Interface Framework: Reuse established user interface frameworks or components for building a consistent and user-friendly interface.

External Services:

- Navcen Website Availability: The software relies on the continued availability and accessibility of the Navcen website for data retrieval. Changes to the website structure may require updates to the web scraping logic.

Hardware Limitations:

- Computational Power: The software's performance may be influenced by the user's hardware capabilities, especially for computationally intensive tasks like DOP calculations.

Data Quality:

- Assumed Data Quality: The software assumes a certain level of data quality and consistency from the Navcen website. Changes in data format or quality may require adjustments to the scraping and processing logic.

Legal and Regulatory Compliance:

- Data Usage Compliance: The software assumes compliance with legal and ethical standards related to web scraping, data processing, and user privacy. Changes in regulations may necessitate updates to the software.

Scalability Requirements:

- Current User Base: The software assumes a specific user base size. If there is an expectation of a significantly larger user base, scalability considerations may need to be addressed.

3. External Interface Requirements

Our software seamlessly interacts with users through an intuitive graphical interface powered by leaflet.js. It enables users to visualize and explore satellite-collected data on a map, offering a user-friendly experience. Key functionalities include selecting historical data via a calendar and saving screenshots. The design strictly follows GUI standards, ensuring consistency across screens. For a more detailed understanding, specific interface characteristics and design details are outlined in a dedicated user interface specification document.

3.1 User Interfaces

The user interface of our project utilizes leaflet.js to visually represent satellite-collected data on a map. Users can interact by selecting dates from a calendar to view historical data and save screenshots. The design adheres to GUI standards with consistent screen layouts, standard buttons, and functions such as "save screenshot." Specifics of the user interface design, including screen images, layout constraints, and error message standards, are detailed in a separate user interface specification document.

3.2 Hardware Interfaces

Our project does not have any hardware interface requirements since it is a web application, therefore as long as the user is on a supported browser it will function.

3.3 Software Interfaces

- Python: The primary programming language for the backend and some of the frontend.
- Libraries:
- Numpy (v1.25.2)
- Matplotlib (v3.7.2)
- Plotly (v5.16.1)
- BeautifulSoup4 (v4.10.0)
- Requests
- Ephem
- Pytz
- InfluxDB

3.4 Communications Interfaces

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

Web Scraping (Grabbing SEM file)

- Communication Protocol: Send an HTTP request to the navcen website to fetch the SEM file.
- Library: ‘requests’ in Python to make HTTP requests.
- Message Formatting: send GET requests to the navcen website and retrieve HTML contents.
- Security Considerations: Ensure compliance with the website’s terms of service.

Data Parsing (BeautifulSoup4)

- Communication Protocol: Internal to application. No specific external communication is required for parsing the SEM file.
- Library: Use BeautifulSoup to parse HTML content obtained from the website.
- Message Formatting: No message format for parsing

Calculating DOP data

- Communication Protocol: Internal to application.
- Message Formatting: Internal to application
- Security: Ensure the calculation process is secure and does not introduce vulnerabilities

Database:

- Communication Protocol: use HTTPS for communication with influxDB
- Security: Ensure that the connection to InfluxDB is secure

Webpage Display:

- Communication Protocol: HTTPS
- Message Formatting: HTML, CSS, and JavaScript
- Security Considerations: Ensure webpage secure

4. Requirements Specification

- The system shall allow users with a valid repository code to initiate data retrieval from the Navcen website.
- The system shall perform web scraping to grab SEM data from the Navcen website.
- The system shall handle changes in the structure of the Navcen website gracefully, providing informative error messages when necessary.
- The system shall provide a DOP calculation module that takes the grabbed data as input.
-
- The system shall implement an algorithm to calculate Dilution of Precision (DOP) accurately.
- The system shall be designed to optimize DOP calculation performance, considering potential hardware limitations.
- The system shall interact with a locally hosted InfluxDB for storing the calculated DOP data.
- The system shall use HTTP/HTTPS protocols to communicate with InfluxDB.
- The system shall include error-handling mechanisms for database interactions, logging errors and providing informative messages.
- The system shall provide a user-friendly interface for users to initiate data retrieval and view calculated DOP data.
- The system shall include mechanisms for user authentication using repository codes.
- The system shall follow secure coding practices to protect user data during interactions with the user interface.

4.1 Functional Requirements

FR1: Web Crawling

The system shall automatically crawl the USCG website for updated data using Python and BeautifulSoup4.

FR2: DOP Data Calculation

The system shall calculate DOP data using the crawled information, including measurements and satellite positions, utilizing Numpy for computations.

FR3: Time Conversion

The system shall convert GPS week and seconds data into Unix timestamps using Python's time manipulation libraries.

FR4: Data Storage

The system shall store the calculated DOP data and corresponding timestamps in the InfluxDB database.

4.2 External Interface Requirements

Repository Code Input:

- Purpose: Initiates data retrieval from the Navcen website.
- Source: Entered by the user.
- Valid Range: Alphanumeric code (e.g., 10 characters).
- Units: None.
- Timing: User-initiated.
- Relationships: Trigger for data retrieval.

Navcen Website Response:

- Purpose: Holds the response obtained by web scraping.
- Source: Navcen website.
- Valid Range: HTML or structured data.
- Units: None.
- Timing: Generated during web scraping.
- Relationships: Directly used as input for DOP calculation.

SEM Data for Calculation:

- Purpose: Input for the DOP calculation algorithm.
- Source: Obtained from Navcen website response.
- Valid Range: Structured data compatible with DOP calculation.
- Units: Relevant to satellite positioning.
- Timing: Immediate after data retrieval.
- Relationships: Directly used as input for DOP calculation.

InfluxDB Insertion Result:

- Purpose: Indicates the success or failure of the data insertion into InfluxDB.
- Destination: Logging and user interface.
- Valid Range: Success/Failure indicators.
- Units: None.
- Timing: Immediate after attempting data insertion.
- Relationships: Influences logging and user messages.

User Interface Messages:

- Purpose: Provides feedback to the user on system actions.
- Destination: Displayed on the user interface.

- Valid Range: Text messages indicating success, errors, or other status.
- Units: None.
- Timing: Immediate after user actions.
- Relationships: Directly linked to user interactions.

DOP Calculation Results Display:

- Purpose: Displays calculated DOP data.
- Destination: User interface.
- Valid Range: Numeric values and graphical representations.
- Units: Relevant to Dilution of Precision.
- Timing: After DOP calculation.
- Relationships: Directly linked to DOP calculation.

4.3 Logical Database Requirements

The system shall interface with InfluxDB for storing processed data.

The schema of the database shall be designed to efficiently store time-series data related to DOP calculations.

4.4 Design Constraints

Specify design constraints that can be imposed by other standards, hardware limitations, etc. This should be a more technical description of the overview given in section 2.5.

Hardware Limitations:

- Run Time Constraints: The calculation of DOP can be computationally intensive, and the runtime may be significantly affected by the capabilities of the user's hardware. Low-end computers with limited processing power may experience longer calculation times.

Computational Complexity:

- Algorithm Efficiency: The algorithm for calculating DOP should be designed with efficiency in mind. While it's important to maintain accuracy, efforts should be made to optimize the algorithm's performance, considering the potential limitations of the user's hardware.

User Experience:

- Real-Time Requirements: If the application requires real-time DOP calculations (for example, in navigation systems), the hardware constraints become even more critical. The system must provide timely and accurate DOP information to the user.

User Interface:

- Graphical Representation: If the DOP results are presented graphically, the design should consider the potential limitations in rendering capabilities on low-end computers. Avoid overly complex visualizations that might strain the system.

Memory Constraints:

- RAM Usage: The DOP calculation may require significant memory, and low-end computers may have limited RAM. The algorithm and data structures should be designed to minimize memory usage where possible.

Compatibility:

- Operating System Compatibility: The application should be compatible with various operating systems, and the design should account for any specific constraints imposed by different platforms.

Data Availability:

- Availability of Satellite Data: The accuracy of DOP calculations relies on the availability of accurate satellite data. If the application is used in areas with limited satellite visibility, the design should account for potential data gaps.

Regulatory Constraints:

- Compliance with Standards: The design should adhere to relevant standards and regulations related to positioning systems and satellite navigation.

Security and Privacy:

- Protection of User Data: Incorporate measures to protect user privacy and comply with relevant data protection regulations.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

This section specifies any numerical/statistical requirements imposed on the software such as:

- The number of terminals to be supported : 1 Terminal
- The number of simultaneous users to be supported : All users are supported to the software
- Amount and type of information to be handled : Real- Time Data will be handled when the user includes the calendar day to revise current or previous data.

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

All of these requirements should be stated in measurable terms. For example, "95% of the transactions shall be processed in less than 1 second" rather than, "An operator shall not have to wait for the transaction to complete."

(Note: Numerical limits applied to one specific functional requirement are normally specified as part of that requirement and should be listed in section 4. This part is more for performance / statistical requirements imposed on the entire system as a whole.)

5.2 Safety Requirements

The frontend does not collect explicit user data. Visualization is based solely on user input without compromising user data.

5.3 Security Requirements

In the software requirements we do not sell any user data that is in the use of this product. The software is purely for information purposes with no data from the user's machine.

5.4 Software Quality Attributes

Reliability: The system must perform consistent and accurate calculations.

Performance: The system should process and store data in a timely manner, suitable for real-time applications.

Maintainability: The code shall be well-documented and modular to facilitate future updates and maintenance.

5.5 Business Rules

Access Control:

- Principle: Only users with the repository code are allowed to grab data from navcen.com.
- Functional Requirement: Implement a secure authentication and authorization mechanism to restrict access to users with valid repository codes. This could involve user authentication using tokens or API keys.

Data Retrieval:

- Principle: Users with the repository code have the ability to grab data from navcen.com.
- Functional Requirement: Develop a mechanism for users to initiate data retrieval from navcen.com, potentially using a web scraping tool or API calls.

Calculation of DOP Data:

- Principle: Users are allowed to calculate DOP data using the retrieved information.
- Functional Requirement: Provide a computation module that takes the grabbed data as input and performs the necessary calculations to derive DOP data.

User Roles and Permissions:

- Principle: Different roles may exist with varying levels of access (e.g., admin, regular user).
- Functional Requirement: Implement role-based access control (RBAC) to assign specific roles to users. Admins may have additional privileges compared to regular users.

Error Handling:

- Principle: Gracefully handle errors or exceptions during the data grabbing or DOP calculation process.
- Functional Requirement: Implement error handling mechanisms to provide informative messages to users and log errors for system administrators to review.

Logging and Auditing:

- Principle: Maintain logs of user activities, especially data grabbing and DOP calculation.
- Functional Requirement: Implement logging functionalities to record user actions, errors, and other relevant events. This supports accountability and troubleshooting.

Data Integrity:

- Principle: Ensure the integrity of grabbed data and calculated DOP data.

- Functional Requirement: Implement data validation checks to ensure the accuracy and reliability of the grabbed data. Include checksums or other integrity verification mechanisms.

User Interface:

- Principle: Provide a user-friendly interface for initiating data grabbing and viewing calculated DOP data.
- Functional Requirement: Develop a user interface that allows users to interact with the system easily. This could involve creating a web-based dashboard or command-line interface.

Documentation:

- Principle: Provide clear documentation on how to use the repository code, grab data, and calculate DOP.
- Functional Requirement: Create comprehensive documentation that guides users on the proper usage of the product, including examples and troubleshooting tips.

Security Measures:

- Principle: Implement security measures to protect the repository code, user data, and the overall system.
- Functional Requirement: Incorporate encryption mechanisms, secure coding practices, and regular security audits to safeguard user information and prevent unauthorized access.

6. Legal and Ethical Considerations

In this section, we will explore the legal and ethical considerations relevant to the DOP Data Processing and Storage System project. This analysis is crucial to ensure compliance with legal standards and adherence to ethical principles.

6.1 Data Privacy and Protection

Given the nature of data collection from the USCG website, it is imperative to comply with data privacy laws and regulations. The project shall adhere to relevant legislation such as the General Data Protection Regulation (GDPR) for European users and other local data protection laws. Personal data, if any, collected during web crawling must be handled with the utmost care, ensuring anonymity and security.

6.2 Intellectual Property Rights

All software libraries and tools used in the project, such as Python and its libraries (Numpy, Matplotlib, Plotly, BeautifulSoup4, Requests, Ephem, Pytz), are subject to their respective licenses. The project shall use these libraries in accordance with their licenses, ensuring no infringement of intellectual property rights occurs. Open-source licenses for each tool will be thoroughly reviewed to ensure compliance.

6.3 Compliance with Web Crawling Ethics and Laws

Web crawling, while a powerful tool for data collection, must be conducted ethically and in accordance with legal standards. The project shall respect the terms of service of the USCG website, ensuring that the crawling activities do not overload their servers or violate any access restrictions. The use of web-crawled data will be for analysis and processing purposes only, aligning with fair use principles.

6.4 Transparency and Accountability

The project shall maintain transparency in its operations, especially concerning data processing methods and storage. Clear documentation and open communication about how data is handled will be maintained. This ensures accountability in case of any data breaches or misuse.

6.5 Ethical Use of Data

The data processed and stored shall be used solely for the intended purpose of DOP data analysis and visualization. There will be strict measures to prevent the misuse of data for unauthorized purposes, such as commercial gain or unauthorized surveillance activities.

6.6 Safeguarding Against Misinformation

Accuracy of information is paramount. The project shall implement rigorous validation methods to ensure the correctness of the DOP data processed. This is crucial to prevent the dissemination of misinformation, which can have serious consequences in navigational and other critical applications.

Appendix A: Glossary

DOP: Dilution of Precision

USCG: United States Coast Guard

Unix Timestamp: A system for describing points in time, defined as the number of seconds elapsed since January 1, 1970.

Appendix B: Analysis Models

Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.