MapONE

Requirements Specification (Version 2)

December 3, 2021

Sponsors:
Planetary Geologic Mapping Program, USGS Astrogeology Science Center
Dr. Sarah Black, Research Physical Scientist
Marc Hunter, IT Specialist

Faculty Mentor:
Melissa D. Rose

Team Members:
Samantha Milligan
Michael Nelson
Ricardo McCrary
Jacob Stuck

Accepted as Baseline Requirements:

Client: ____________________ **Date:** 12/6/2021

Team Leader: ____________________ **Date:** 12/3/2021

Table of Contents

1. Introduction

The planetary science community develops, collects, and distributes planetary geologic mapping research. This research includes publications presenting mapping work and the development process of new map products. Similar to geologic mapping on Earth, planetary maps identify key features such as craters, canyons, and fault lines on other planets. Scientists use these shared resources within the community for numerous reasons. This includes mapping space exploration sites and learning about the elemental composition of different planets and other related scientific research. The client, the United States Geological Survey (USGS) Planetary Geologic Mapping (PGM) Program, assists the community by developing tools and resources to better access and use planetary data for these purposes. For example, PGM mapped the landing/exploration site for the National Aeronautics and Space Administration's (NASA's) Mars 2020 Perseverance Rover mission. In this case, scientists used map products to gather information about Mars's surface.

2. Problem Statement

Scientists publish their planetary maps and research either through USGS or various online sources (often in journal articles and conference papers). Because USGS mandates certain map standards, many publications are distributed throughout the internet instead. Nevertheless, these resources are valuable to the client to provide the community with all types of publications. As shown in Figure 1, USGS currently does not have an automated way to collect these existing resources. As a note, the planetary science community consists of USGS and non-USGS scientists and researchers. Overall, the client and the planetary science community lack a single, centralized system for maps published in non-USGS venues.

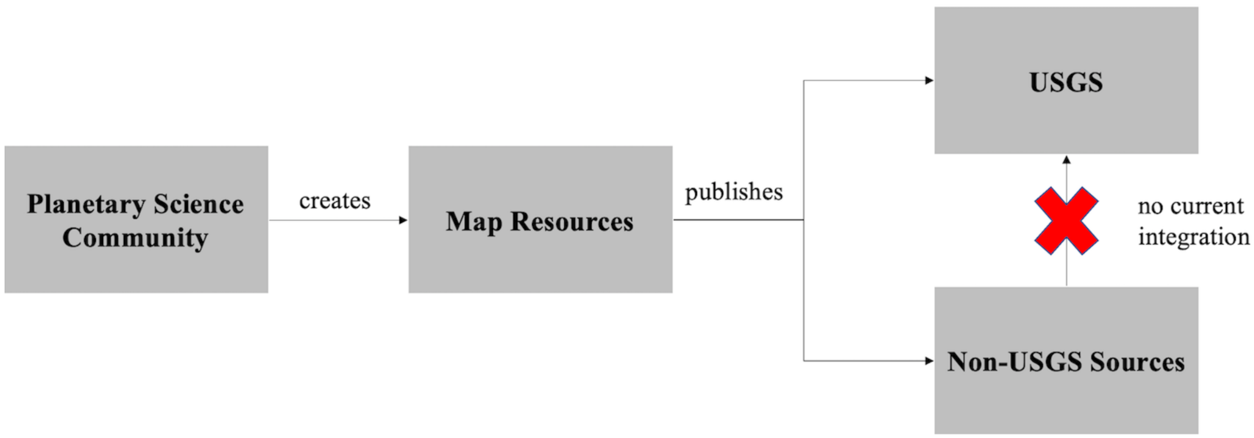


Figure 1. Diagram outlining planetary map integration issue.

Specifically, the client lacks the following features:

1. **Web Application**: Map resources published through non-USGS sources are not currently listed on a single platform.
2. **Automation**: The client has to manually collect these publications.
3. **Single Database**: The client cannot easily access these resources since non-USGS publications are stored throughout the internet.
4. **Remote Access**: The client does not have a mechanism to archive these maps.
5. **Metadata Selection**: Specific planetary metadata (author, source, region, etc.) from the non-USGS publications is not available to the client.

3. Solution Statement

Based on the needs outlined above, the project team will create a centralized system for non-USGS map publications. The envisioned product, MapONE, is a user-friendly interface accessible through the USGS's website. The interface will display map metadata for a selected region of interest and connect to one database containing non-USGS publications. The project team will also create a web scraper to identify and collect these publications from online sources. Overall, MapONE will allow researchers to access non-USGS map metadata. As shown in Figure 1, the following technologies will be used to create this product:

1. **Graphical User Interface (GUI)**: A GUI will be used to display map data on a single web application.
2. **Web Scraper**: The project team will develop a data extraction tool, known as a web scraper, to automatically detect and gather map publication metadata (author, source, region, etc.) from online non-USGS sources.
3. **Central Database**: MapONE will store publication metadata in one centralized database.
4. **Remote Storage and Servers**: Cloud computing services can be used to store map data and run the product's web application.
5. **Web Framework**: MapONE's web framework will filter metadata (author, source, region, etc.) from the database using user requests from the GUI.

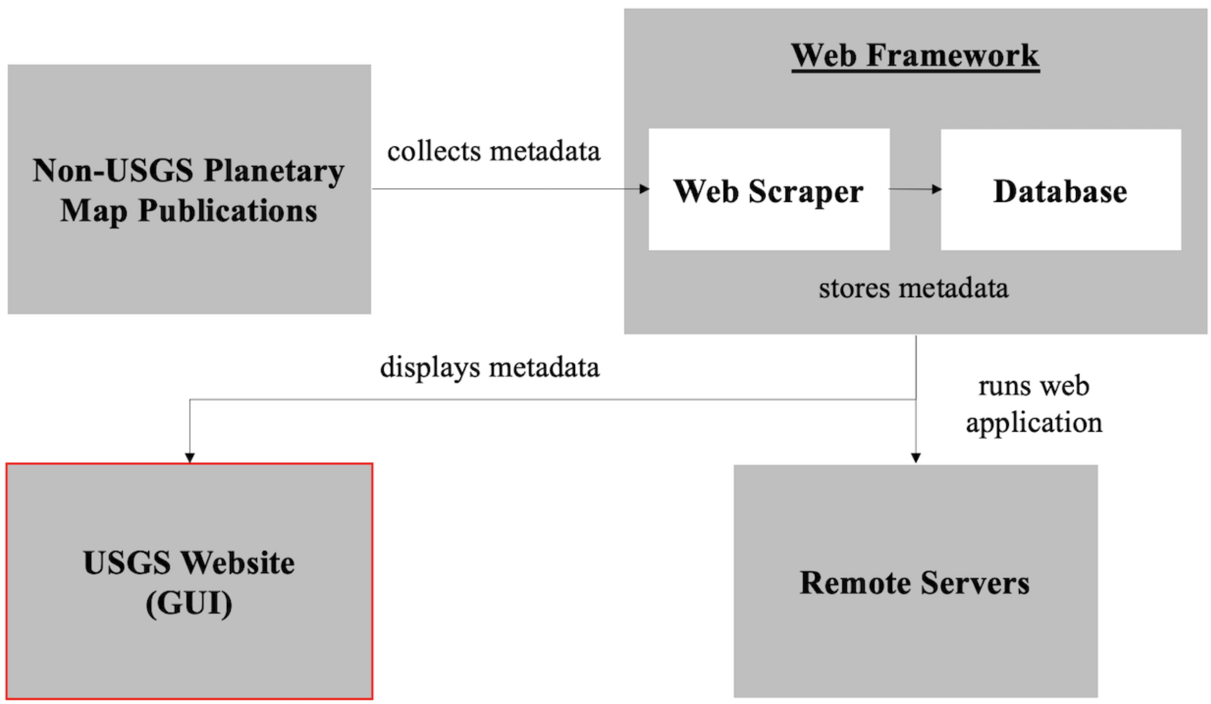


Figure 1. MapONE's solution overview.

At the focal point, the web scraper locates online map products and extracts publication metadata for the web framework to store in a database. The framework will then display the data to the client using a GUI. This way MapONE ensures these publications and content are available to the client. Together, these technologies will create a system for users to view and archive map publication data.

4. Project Requirements

Based on the solution overview, the team needs to outline MapONE's key system requirements. The product is intended for the client and other scientists/researchers in the planetary science community. Thus, the web application must be simple and allow users to effectively gather publications based on a search engine. The following domain requirements encompass the key application features the user needs:

1. Login into an account on the USGS website.
2. View and filter planetary map metadata (author, source link, region, etc.).
3. Download all publication entries.
4. View and save search history results.
5. Automate searches periodically.
6. Receive notifications on new publications from automated search results.

Based on these features, the following sections outline the related functional, performance, and environmental requirements. As a note, the project team has obtained these requirements from various client meetings and a detailed project description.

4.1 Functional Requirements

User Account System

Users should be able to log into an account when first visiting the USGS website. MapONE will show a login page with options to sign in, create a new account, or continue without one. If a user continues without an account, the system will redirect to MapONE's main page. Otherwise, the user must provide a valid email address and password.

To create a new account, the system will verify the user's email address with an email validation Application Programming Interface (API); the system will send the user's input as a request to the API. If the API returns a confirmation, the system will allow the user to then create a password. If the API returns an invalid response, the system will continue to prompt the user to enter a valid email address. Likewise, if the user submits an invalid password (does not adhere to at least 8 characters and 1 special character), the system will continue to display an error message. Once the user submits valid credentials, the system will redirect to the login page. The system will also save a generated unique identifier (ID), email address, and password in the database for new users.

Users with existing accounts must provide valid credentials. The system will confirm the email address and password are from the same existing account (checks database and verifies user ID/information). The system will display an error message for all invalid credentials. Upon successful login, the user will be directed to the main page where options to sign in/out and switch user accounts are available. Lastly, under the user profile on the main page, users can view past saved search history results; MapONE's search engine is discussed in the next section. As a note, users not logged into an account cannot receive email notifications or save automated search results.

Search Engine

Once directed to the main page, the system displays a catalog of planetary map metadata (source name, source link, map area, map scale, and author). This information along with a generated unique ID is saved in MapONE's database. The system will also display a search engine where users can search for keywords (no criteria) and filters (source name and map area). As a note, the search engine and filters are automatically turned off when a user first views the main page; the page displays all publications in the database (likely multiple pages worth).

The system uses keywords to locate all publication IDs with matching characteristics; the system searches each stored metadata field to find a match in the database. Similarly, filters will be used to locate publication IDs with the selected information. If no publications are found with specific keywords or filters, the system will display a "no results" message. If results are found, the system will return and display all fields (excluding publication ID) that meet both the keywords and filters criteria. Publications with limited information (for example, simply a URL) may not be returned.

The user will have the option to download results in a Comma-Separated Values (CSV) format. If a user is logged into an account, the system displays an option to save results. For an automated search, the user must select a frequency (weekly, monthly, etc.). Search history and automated searches/frequency are stored in MapONE's database under the user's ID; the system creates a new search class (a database class) entry with a generated unique ID and saves the user ID as a field in the entry. The search class contains a search ID, user ID, frequency (empty if not an automated search), search history keyword(s), applied filters, and number of publications returned. As previously mentioned, the user profile can view past and automated searches. The system can display this information by locating search IDs in the database under a user's ID. When the user selects to view a past search, the system directs the user to the main page with the saved keywords and filters already applied.

Web Scraper

MapONE's web scraper must first locate map products to view publication entries in the database. The system uses a machine learning model to identify publications on non-USGS web interfaces, such as the Lunar Planetary Institute (LPI) and Universities Space Research Association (USRA). The model can be trained using a dataset of publications known to contain map products provided by the client. The model will then learn what standards must be present (title, source, author, etc.) from the dataset and execute new search tests (using keywords derived from the dataset and other hardcoded planetary terms) to continuously improve accuracy. After learning the dataset, the model will be able to locate feasible map products.

Upon locating a map product, the tool extracts metadata from the publication's HTML document. All map products stored in the database must contain a source name, source link, and map area (map scale and author are optional fields). This extracted data will then be saved in the system's database under a generated unique ID for each publication entry.

Stretch Goal: The web scraper can identify figures and images within map publications, as some sources may include irrelevant information, duplicate images, etc. A web scraper that recognizes planetary figures and their features can act as another filter to the database, ensuring extraneous metadata is not included.

Notification System

The automated web scraper will continue to identify and collect new map products. Users with automated search requests will receive notifications on new publications. For all search IDs with a set frequency (classified as an automated search), the system will internally check how many publications are currently returned from the search (see "Search Engine" for more details). The system will continue to do this based on the frequency. If the number of publications returned is greater than the previous number stored in the database, the system will update the number and send an email notification to the user.

Use Case: Automated Search Notification

1. User logs into the system.
2. User searches for publications with the keyword "Mars" and filters the source name by "LPI."
3. User saves search in user profile with a frequency of "Weekly."
4. User receives an email notification *each week only if* new publications have been found.
5. User returns to the user profile and sees all results returned from the automated search.

4.2 Performance Requirements

Accessible GUI

MapONE's user interface will be easy to navigate. The product's targeted audience is researchers and scientists (non-USGS and USGS alike). For this reason, users should be able to view, save, and distribute all publications with ease. The project team's goal is for users to easily access all product features on the main web page including the user account system and search engine. Although users have access to this feature, user accounts are not required. This invites all users to easily access the GUI without a required authentication process. After the login page, users with or without an account are directed to the main page where they can view all publications and use the search engine to locate specific data. Users can also download search results and set automated searches on the main page. These features allow users to easily access and save data for later use which is crucial for research collection purposes; users do not have to frequently return to the website to view data if saved or exported.

Response Speed of Automated Searches

As discussed in the “Web Scraper" functional requirement section, all automated searches saved in the database must be executed periodically depending on the set frequency. For this reason, the web scraper must be configured to scan and pull new data. As requested by the client, data must be pulled on a monthly if not weekly basis. The project team goal is to set weekly as MapONE’s fastest available frequency to ensure frequent users have up-to-date data on publications. The system will set an internal timer each week. Once the timer ends, the web scraper’s functions will be executed to scan non-USGS web interfaces and if new publications are found, save data in the database. The timer is then reset. This ensures that all searches, most importantly, automated searches are up-to-date and easily available to users.

4.3 Environmental Requirements

MapONE’s success relies on the software integration of many technologies (GUI, web scraper, web framework, and database). The product must be a web-based, open-source Python tool as requested by the client. Python will be used for its versatility in data analytics and machine learning which is crucial for the web scraper’s development as an automated feature. Although these requirements offer flexibility, the project team needs to ensure a smooth integration for future maintenance after the product’s deployment. Thus, the team has devised three routes for integration:

1. Flutter GUI to Django Web Framework
2. Django Web Framework and PostgreSQL Database
3. Django Web Framework and Keras Web Scraper

Flutter GUI to Django Web Framework

The project team is using Flutter, a frontend web-based software to create the GUI. Flutter is written in the Dart language and can be configured with a Python plug-in to ensure compatibility.

Django, a Python full-stack web development software, will be used to structure MapONE's web framework and backend. To connect the GUI (frontend) to the web framework (backend), the two will run on separate servers. The GUI can then execute an API request to the Django server to gather information from the backend and display to users on the frontend.

Use Case: GUI API Request to Django Server

1. User searches for all publications related to Mars.
2. The GUI makes an API request to the web framework server.
3. Web framework sends the GUI a list of all publications related to Mars.
4. The GUI displays publications to users.

Django Web Framework and PostgreSQL Database

To ensure ease of integration, the database will be created within the web framework. Django automatically sets up a database in the backend and supports many languages including PostgreSQL, MapONE's database structure. This can be configured in Django's settings and is easy to set up. This overall setup allows the web framework to easily pull data from the database and output the data to the GUI. Django uses a file called *models.py* to structure the database which can be called anywhere within the framework including *views.py* where the API exists.

Django Web Framework and Keras Web Scraper

To ensure ease of integration, the web scraper will also be created within the web framework. Keras, a Python machine learning library, will be used to create the web scraper. Because Django and Keras are both written in Python, the web scraper can be easily integrated into the backend. The web scraper files can be called anywhere within the framework including *models.py* where the tool can load extracted metadata into the database. This way, the database and web scraper can interact with each other within one framework.

5. Potential Risks

Data Duplication

MapONE's most prevalent risk is the duplication of data inside the database. Without proper configuration, the web scraper may pull data from publications already listed in the database. This is likely since the same map publication may be used by multiple sources in various journals or articles. Duplicate data could strain the database's resources with limited storage and time-costly data pulls. This would harm the user's interaction with the GUI where duplicate information is displayed. The website may also experience delays with user search requests. Data retrieval time would likely increase. To mitigate this issue, the system will check all new data extracted from the web scraper with the existing database entries. Publications with the same title will not be added to the database.

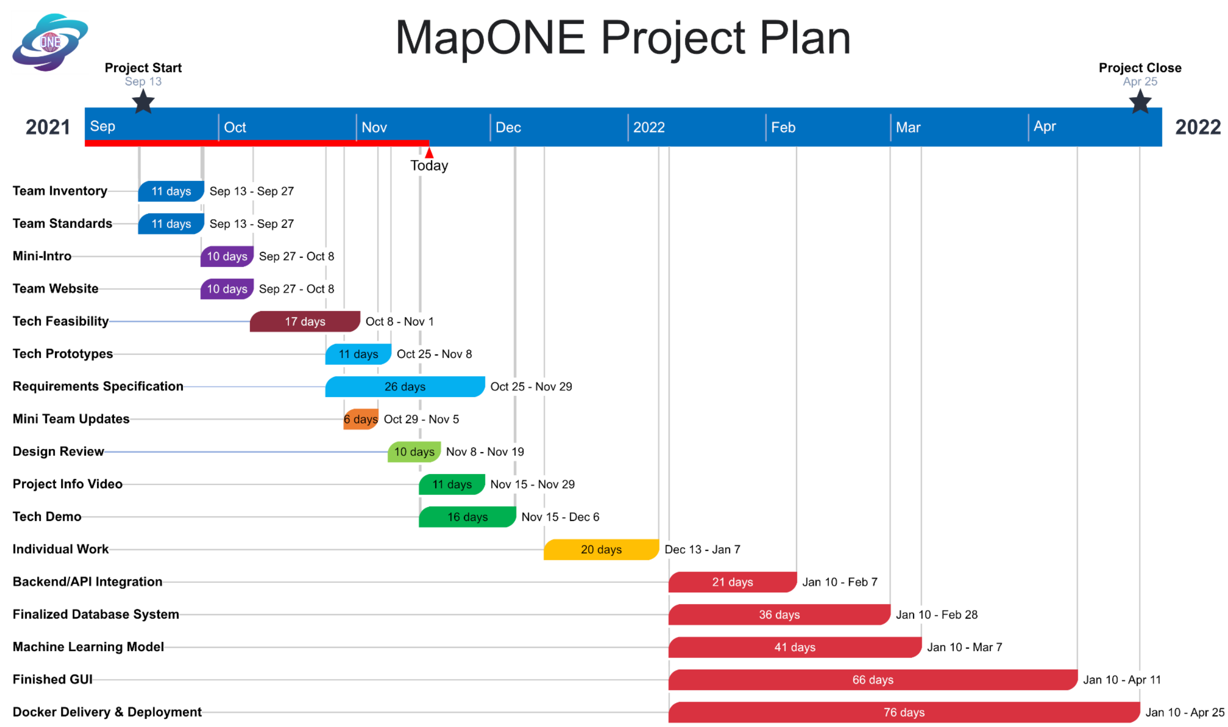
Server Shutdowns

Server crashes and potential shutdowns are common issues most data retrieval projects face. When servers crash, data may become corrupted or lost. If a server crash were to occur during a run cycle of the web scraper or a database upload, those data pulls and publications would be lost. This may cause the web scraper to rerun a data pull and add duplicate data. A server shutdown would also inhibit users' access to the product. Overall, a server crash would slow down MapONE's data acquisition. To mitigate this risk, a monthly data sanitation will be used to remove corrupted data in the database. The system will check if all publication URLs are valid and in use and remove any invalid entries; the system will send a request to the URL and view if the Hypertext Transfer Protocol (HTTP) response displays an error.

SQL Injection Attacks

SQL injection attacks are malicious attacks in which the attacker hides their identity to inject data into the server or database [1]. This would cause instability in the system and cause deterioration in the database's data and reliability. Because MapONE's server will be run and maintained by USGS, a government agency, there is an increased likelihood of these kinds of attacks. Data could be corrupted which would slow down research initiatives and possible space exploration missions. As with a server crash, the system will run a monthly data sanitation.

6. Project Plan



This figure shows the project team's current progress and future schedule. The team has completed initial documentation including the Technological Feasibility Analysis and Requirements Specification. These initial tasks laid a foundation for the technologies and system requirements needed for the product's development. Moving forward, the project team will continue to complete individual research on assigned sections. However, the team will work together to integrate the GUI, web framework, web scraper, and database to create one centralized system. The final product will be deployed in April of 2022.

7. Conclusion

The project team is set to develop MapONE, a centralized system for planetary map metadata, for final production in April 2022. The client, USGS PGM Program assists the planetary science community by developing tools to share planetary geologic mapping research. This research is often published either through USGS or various online sources. Currently, non-USGS publications are not listed on a single platform. To address this problem, the project team is developing a web application through USGS. The website will include a user-friendly interface that will display map metadata for a selected region of interest and connect to one database containing non-USGS publications. To create this product, the team must integrate a GUI, web framework, web scraper, and a database. Users will have access to view, save, and periodically automate searches to gather resources. Some of the high-level requirements include a user account system, search engine, and notification system to ensure users have access to all product features. Overall, MapONE ensures researchers have the necessary tools to collect planetary metadata for scientific research and space exploration missions alike.

References

[1] "SQL Injection" OWASP. https://owasp.org/www-community/attacks/SQL_Injection (accessed November 16, 2021).