network-lib

A blog post documenting a 4-week sprint by the Computational Science and Informatics (CSI Team) interns to create network-lib: a proof-of-concept, software application to help answer a frequently asked and (surprisingly difficult to answer) question at engineering firms: ***What existing data do we have on our server that we can use for 'this project' or 'this proposal'?***

By Shivam Asija, Sonia Khan, and Justin Paul

Problem Statement

The standard nested file management system makes the discovery of summary information in large project folders and network drives difficult to collect in any concise way without carefully searching directories and opening many documents and data files. Further complicating the task is undocumented or inconsistent file storage protocols / structure for projects. For example:

- Similar or related files may be separated into different directories
- Files with the same name often exist in multiple directories
- File prefixes can often contain variations of "Final," "Copy of," and "Final_v2" across directories
- Zipped (and nested zipped) files can be difficult to analyze.

Common issues for engineering firms like Dewberry are accessing, understanding, and analyzing files in the millions of project folders stored on network drives. Looking back on a completed project is a tough task that requires navigating the file explorer and opening each individual file to gain insight into its contents. Here are some example filenames and the software needed to access the files if anything beyond the name of the file is desired:

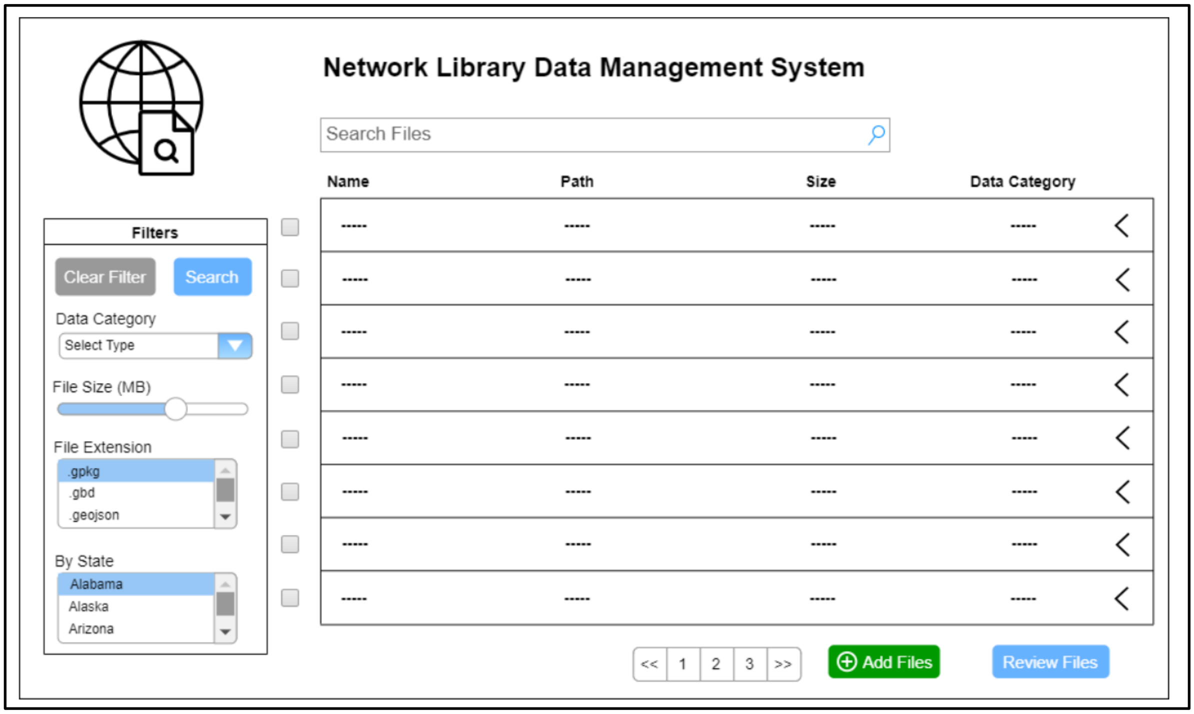
- Copy of Copy of STARRII_Hydraulics_MillCreek_2232022_LZ.xls (Microsoft Excel)
- Terrain_modified_final.Terrain.mcdtm_proj.tif (GIS software, e.g. ArcGIS)
- culv.shp (GIS software, e.g. ArcGIS)
- culv.prj (Text editor)
- mcdm_100yr_detailed_v1.prj (USACE Modeling software)

Further complicating the work done in Resilience Solutions and many other BUs is access to geospatial data. A simple question such as, "what geographic location does this dataset represent?" requires opening the file, rendering the result in an application, such as ArcGIS or QGIS, and then overlaying the data on a map for context. To do this for multiple files is a tedious process and takes up a lot of valuable time.

These issues prompted the network-lib project: a set of automated tools that can drill down into project directories, identify metadata, expose via a search engine, and aid in a rapid understanding of the next-level information stored on our servers for existing and previous studies.

Design

As inspiration for this project, we investigated websites and searched pages that provided similar functionality to our plans. A common factor amongst them was a panel that gave the user control to sort and filter the data. Our panel design used a dropdown option for sorting and checkboxes for filtering. A tabular structure of our data where each column has an attribute common to all files ensured conciseness.



Wireframe of Landing Page

We chose to have a platform containing all files within a large project folder that users can search through. Our project presents the basic information that a file explorer gives, but the client can also receive a detailed outline of specific file types. The goal is to create an easy-to-use filtering tool capable of helping industry professionals find relevant information and file specific metadata.

File Information

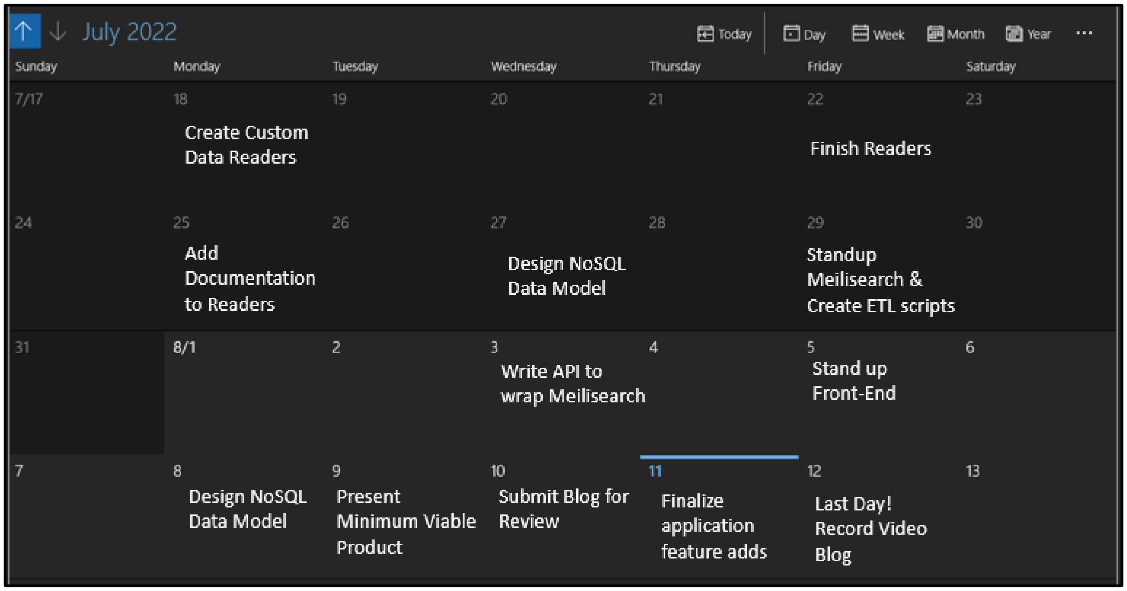
The basic information scraped from each file includes:

- filename
- data classification
- extension
- last modified
- size

The extra information is unique for each file type (raster, vector, gis-database, spreadsheet, table, archive). For example, the extra information in a spreadsheet would include the column names and the number of entries.

Project Management

Development and management require coordination and communication between team members. Agile management practices were crucial to the structure of workflow, as frequent deliverables and refinements were needed consistently. Our team's timeline was scheduled as follows:

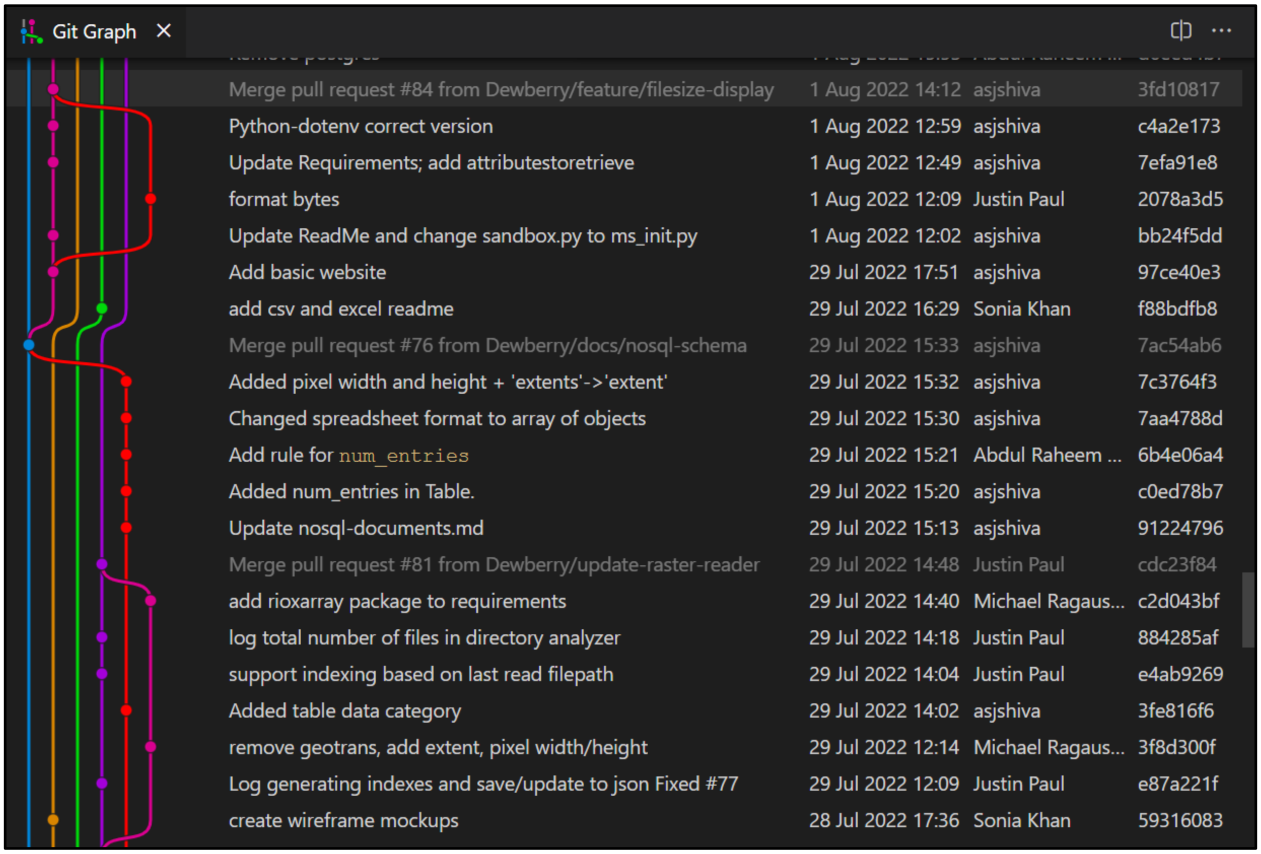


Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7/17	18 Create Custom Data Readers	19	20	21	22 Finish Readers	23
24	25 Add Documentation to Readers	26	27 Design NoSQL Data Model	28	29 Standup Meilisearch & Create ETL scripts	30
31	8/1	2	3 Write API to wrap Meilisearch	4	5 Stand up Front-End	6
7	8 Design NoSQL Data Model	9 Present Minimum Viable Product	10 Submit Blog for Review	11 Finalize application feature adds	12 Last Day! Record Video Blog	13

Development Schedule

To ensure effective communication between teammates, we held daily 15-minute SCRUMs and an hour-long update meeting to share personal progress and brainstorm ideas.

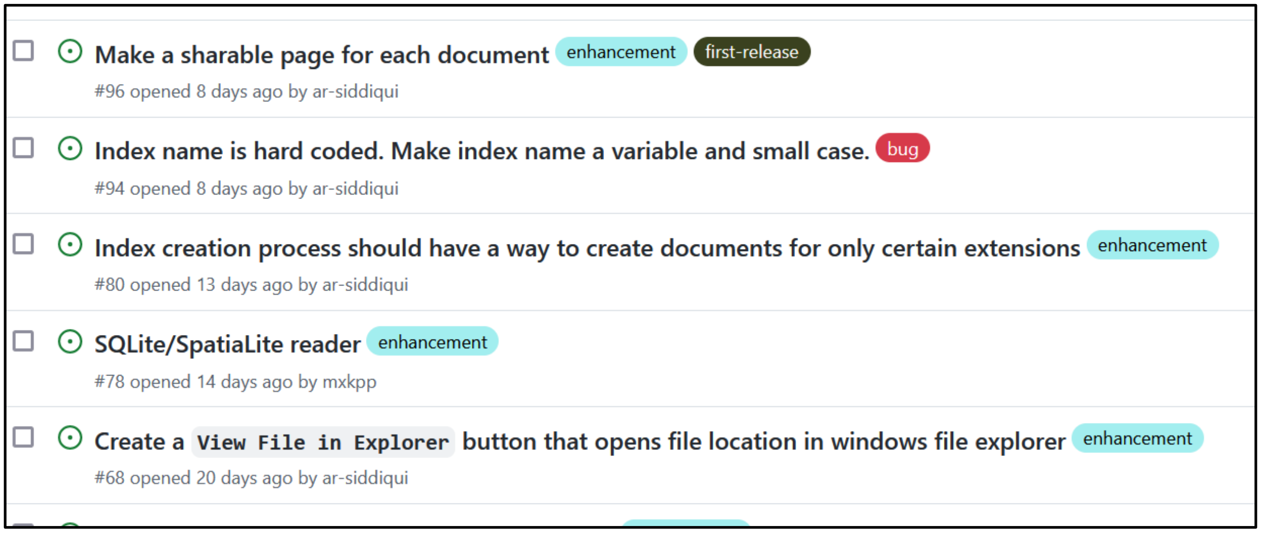
We standardized the development process by making use of GitHub and normalized format settings on our Integrated Development Environments (IDE) to keep team members up to date with ongoing revisions. Collaboration primarily took place on GitHub. For example, to maintain version control, we required new features and updates to the software to be reviewed by at least one other team member.



```
Merge pull request #84 from Dewberry/feature/filesize-display   1 Aug 2022 14:12   asjshiva          3fd10817
Python-dotenv correct version                                   1 Aug 2022 12:59   asjshiva          c4a2e173
Update Requirements; add attributestoretrieve                   1 Aug 2022 12:49   asjshiva          7efa91e8
format bytes                                                    1 Aug 2022 12:09   Justin Paul       2078a3d5
Update ReadMe and change sandbox.py to ms_init.py               1 Aug 2022 12:02   asjshiva          bb24f5dd
Add basic website                                               29 Jul 2022 17:51  asjshiva          97ce40e3
add csv and excel readme                                        29 Jul 2022 16:29  Sonia Khan        f88bdfb8
Merge pull request #76 from Dewberry/docs/nosql-schema          29 Jul 2022 15:33  asjshiva          7ac54ab6
Added pixel width and height + 'extents'->'extent'              29 Jul 2022 15:32  asjshiva          7c3764f3
Changed spreadsheet format to array of objects                  29 Jul 2022 15:30  asjshiva          7aa4788d
Add rule for num_entries                                        29 Jul 2022 15:21  Abdul Raheem ...  6b4e06a4
Added num_entries in Table.                                     29 Jul 2022 15:20  asjshiva          c0ed78b7
Update nosql-documents.md                                       29 Jul 2022 15:13  asjshiva          91224796
Merge pull request #81 from Dewberry/update-raster-reader       29 Jul 2022 14:48  Justin Paul       cdc23f84
add rioxarray package to requirements                           29 Jul 2022 14:40  Michael Ragaus... c2d043bf
log total number of files in directory analyzer                 29 Jul 2022 14:18  Justin Paul       884285af
support indexing based on last read filepath                    29 Jul 2022 14:04  Justin Paul       e4ab9269
Added table data category                                       29 Jul 2022 14:02  asjshiva          3fe816f6
remove geotrans, add extent, pixel width/height                 29 Jul 2022 12:14  Michael Ragaus... 3f8d300f
Log generating indexes and save/update to json Fixed #77        29 Jul 2022 12:09  Justin Paul       e87a221f
create wireframe mockups                                        28 Jul 2022 17:36  Sonia Khan        59316083
```

Logging of feature updates

Using the VSCode IDE, we were able to distinguish those changes from our local repositories easily. Any roadblocks that arose were specified in GitHub's *Issues* tab and assigned based on matching necessary skillsets.



☐ ⊙ Make a sharable page for each document enhancement first-release
#96 opened 8 days ago by ar-siddiqui

☐ ⊙ Index name is hard coded. Make index name a variable and small case. bug
#94 opened 8 days ago by ar-siddiqui

☐ ⊙ Index creation process should have a way to create documents for only certain extensions enhancement
#80 opened 13 days ago by ar-siddiqui

☐ ⊙ SQLite/SpatiaLite reader enhancement
#78 opened 14 days ago by mxkpp

☐ ⊙ Create a `View File in Explorer` button that opens file location in windows file explorer enhancement
#68 opened 20 days ago by ar-siddiqui

GitHub Issues tab

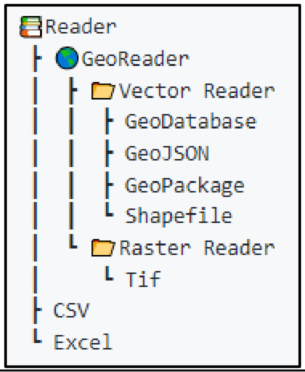
Our team prioritized these standards for code consistency and the ability to revisit logged progress throughout the project.

Activities and Obstacles

This section provides a chronological timeline for the major components of the project, and how we worked in an Agile environment to complete tasks efficiently and on schedule.

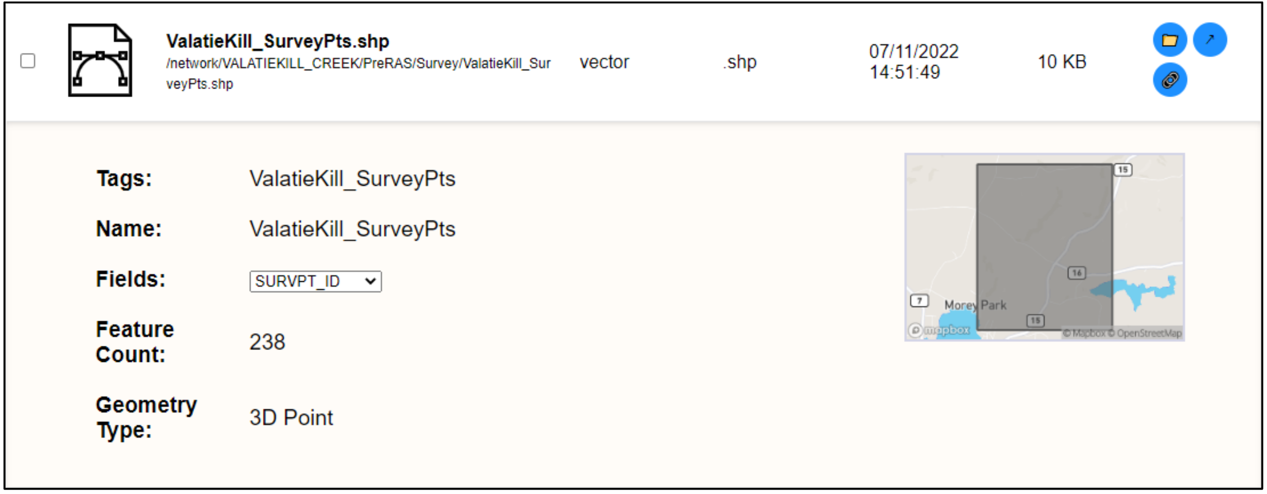
Basic Readers

The intention behind creating readers is to scrape useful information about files. When collecting data, there is general information common to files, but different file types have extra information specific to them. To address this, we made file-specific readers that scraped this extra information. To do this effectively, there was a hierarchy of readers as follows:



```
Reader
├ GeoReader
│  ├ Vector Reader
│  │  ├ GeoDatabase
│  │  ├ GeoJSON
│  │  ├ GeoPackage
│  │  └ Shapefile
│  └ Raster Reader
│     └ Tif
├ CSV
└ Excel
```

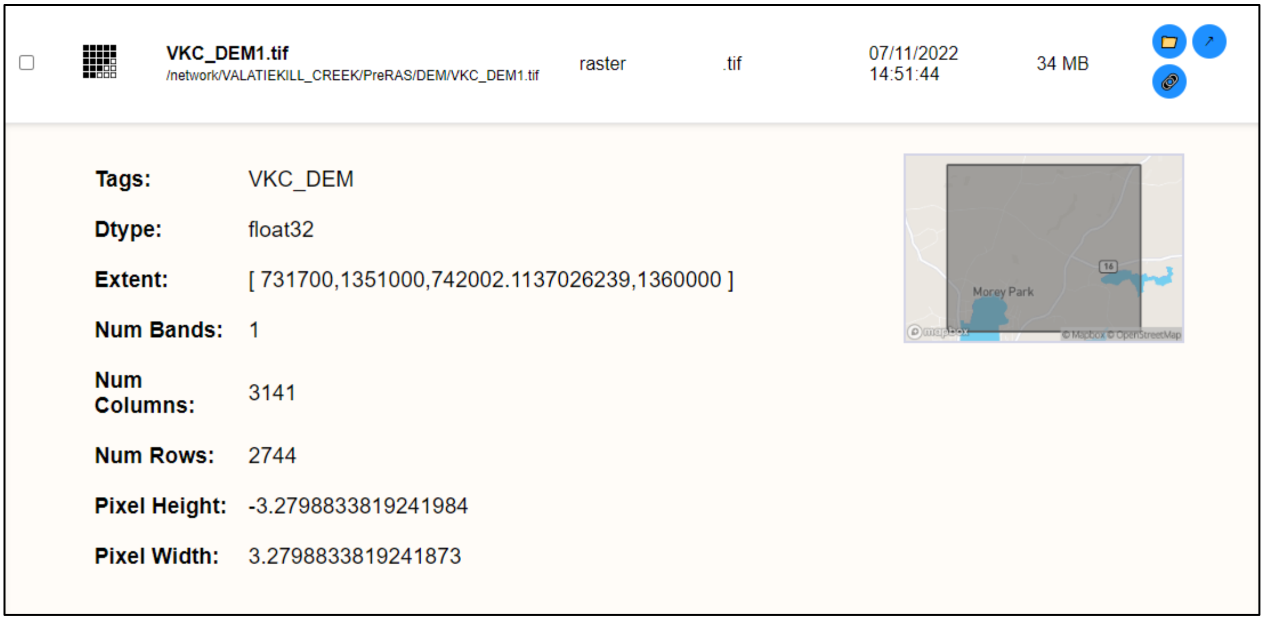
Of note, the data category for geodatabase and geopackage readers is “gis-database.” When working on the structure, an oversight was that these gis-databases can store multiple data categories comprising of “table,” “vector,” and “raster.” For the initial release, gis-databases only scrape vector and table data.



□ **ValatieKill_SurveyPts.shp**
/network/VALATIEKILL_CREEK/PreRAS/Survey/ValatieKill_SurveyPts.shp
vector .shp 07/11/2022 14:51:49 10 KB

Tags:	ValatieKill_SurveyPts
Name:	ValatieKill_SurveyPts
Fields:	SURVPT_ID
Feature Count:	238
Geometry Type:	3D Point

Data card for a geospatial dataset (vector) contained within ValatieKill_SurveyPts.shp



□ VKC_DEM1.tif
/network/VALATIEKILL_CREEK/PreRAS/DEM/VKC_DEM1.tif
raster .tif 07/11/2022 14:51:44 34 MB

Tags: VKC_DEM

Dtype: float32

Extent: [731700,1351000,742002.1137026239,1360000]

Num Bands: 1

Num Columns: 3141

Num Rows: 2744

Pixel Height: -3.2798833819241984

Pixel Width: 3.2798833819241873

Data card for a geospatial dataset (raster) contained within VKC_DEM1.tif

Directory Analyzing and Grouping

A Directory Analyzer walks through each file in a directory and groups them based on their names. These groupings are used as tags for each file.

Automatic tag cleaning methods include:

- Neglecting extensions of files (document.docx and document.pptx -> document)
- Filtering out keywords such as “Copy of” or “Final”
- Trimming leading and trailing characters (numbers and select special characters)

Another capability of the directory analyzer (which has not yet been integrated) is fuzzy tagging. The tag **“STARRII_Hydraulics”** would represent the following 14 tags:

STARRII_Hydraulics

Hoosic_US_STARRII_Hydraulics

STARRII_Hydraulics_PostenKill

LittleHoosicSTARRII_Hydraulics

STARRII_Hydraulics_TsatswassaCk

STARRII_Hydraulics_MillCreek_2232022_LZ

STARRII_Hydraulics_MillCreek_04292022_LZ

STARRII_Hydraulics_Quackenderry_Creek

LittleHoosicSTARRII_Hydraulics_5-9-2022(lz)

STARRII_Hydraulics_TsatswassaCk_9-16-2021_LZ

STARRII_Hydraulics_9262019_NB_Moordener_Kill_DC

STARRII_Hydraulics_Quackenderry_Creek_3-24-2022_LZ

STARRII_Hydraulics_Quackenderry_Creek_04-26-2022_LZ

STARRII_Hydraulics_9262019_NB_Moordener_Kill_DC_9-29-2021_BP

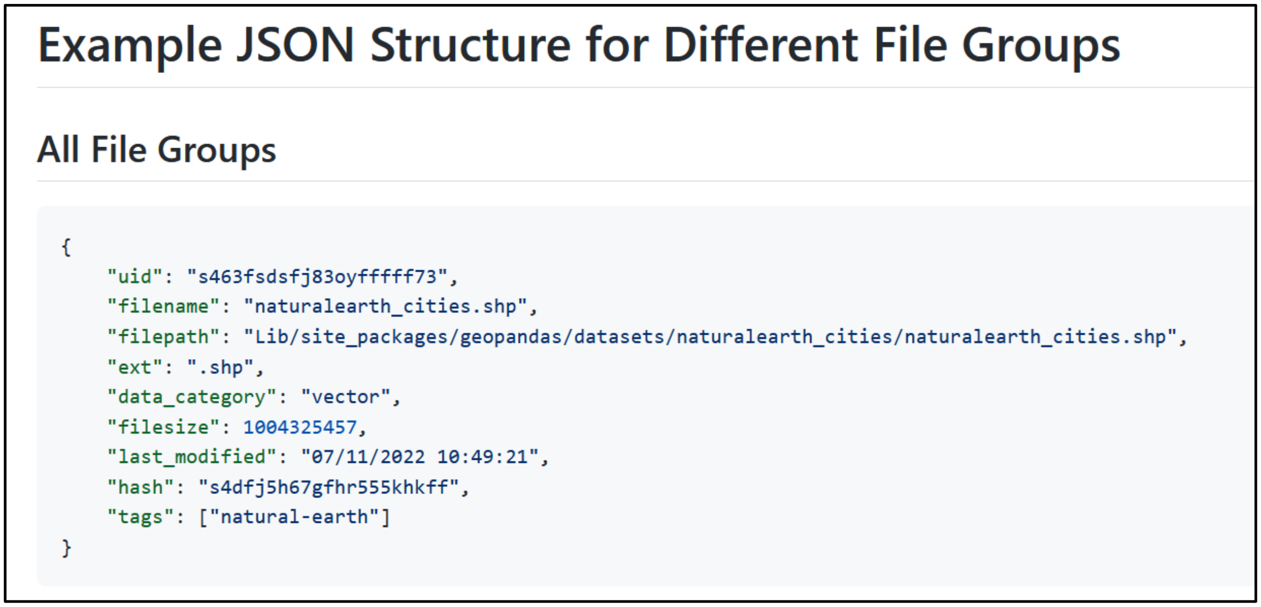
A use-case for this tagging system would be to find similar files that have inconsistent naming that could be in different folders.

Archive Reader

An Archive reader reads zipped files (zip, tar, and 7z) and “extracts” the files inside of it. This is more convenient as the extraction is done programmatically, and the actual zipped folder is only extracted into memory—not into the working directory. As the archive reader scrapes information about the file and contains a nested file system, it inherits properties from both the directory analyzer and basic reader.

Database Population

We chose to use Meilisearch as our database due to its efficiency and ease of implementation. Meilisearch is a search engine and database using a lightening memory-mapped database (LMDB) manager to store large datasets using a NoSQL json record set, which makes it ideal for our application. The directory analyzer traverses through a folder creating file tags and a list of valid files to be fed into the readers. The information gathered by our custom readers is used to populate the database. We used Python and Postman to accomplish this.



Example JSON Structure for Different File Groups

All File Groups

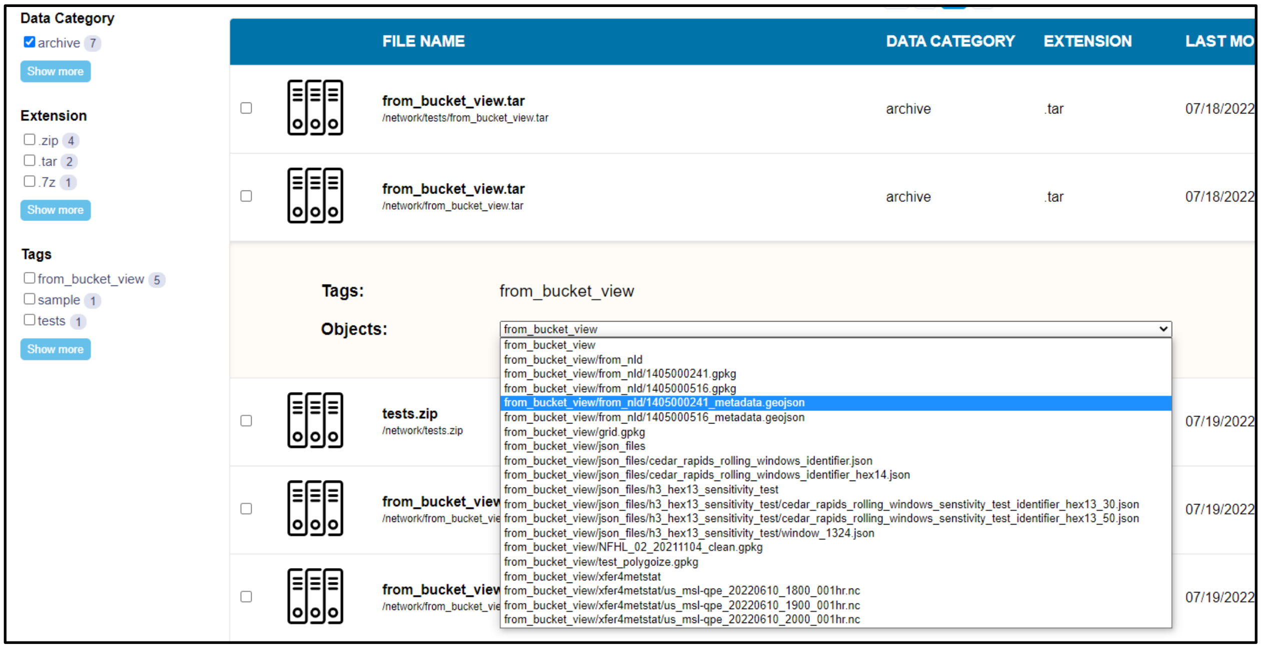
```
{
    "uid": "s463fsdsfj83oyfffff73",
    "filename": "naturalearth_cities.shp",
    "filepath": "Lib/site_packages/geopandas/datasets/naturalearth_cities/naturalearth_cities.shp",
    "ext": ".shp",
    "data_category": "vector",
    "filesize": 1004325457,
    "last_modified": "07/11/2022 10:49:21",
    "hash": "s4dfj5h67gfhr555khkff",
    "tags": ["natural-earth"]
}
```

JSON Document of Example File Group

UX/UI

The first part of website design is to determine the requirements and use-cases of the website. To achieve this, we drafted wireframes to visualize how users would interact with this system.

Looking for inspiration, our team weighed the pros and cons of existing filing system features, including the Quick Access sidebar on Windows File Explorer and the file previewing feature on Mac's Finder. Our goal for this design process was to create a mock-up of an enhanced file system. It would have qualities of a typical system but catered to the characteristics of files within Dewberry's network, such as geodatabases and raster files.



Data Category

☑ archive 7

Show more

Extension

☐ .zip 4
☐ .tar 2
☐ .7z 1

Show more

Tags

☐ from_bucket_view 5
☐ sample 1
☐ tests 1

Show more

	FILE NAME	DATA CATEGORY	EXTENSION	LAST MO
☐	**from_bucket_view.tar** /network/tests/from_bucket_view.tar	archive	.tar	07/18/2022
☐	**from_bucket_view.tar** /network/from_bucket_view.tar	archive	.tar	07/18/2022

Tags: from_bucket_view

Objects: from_bucket_view

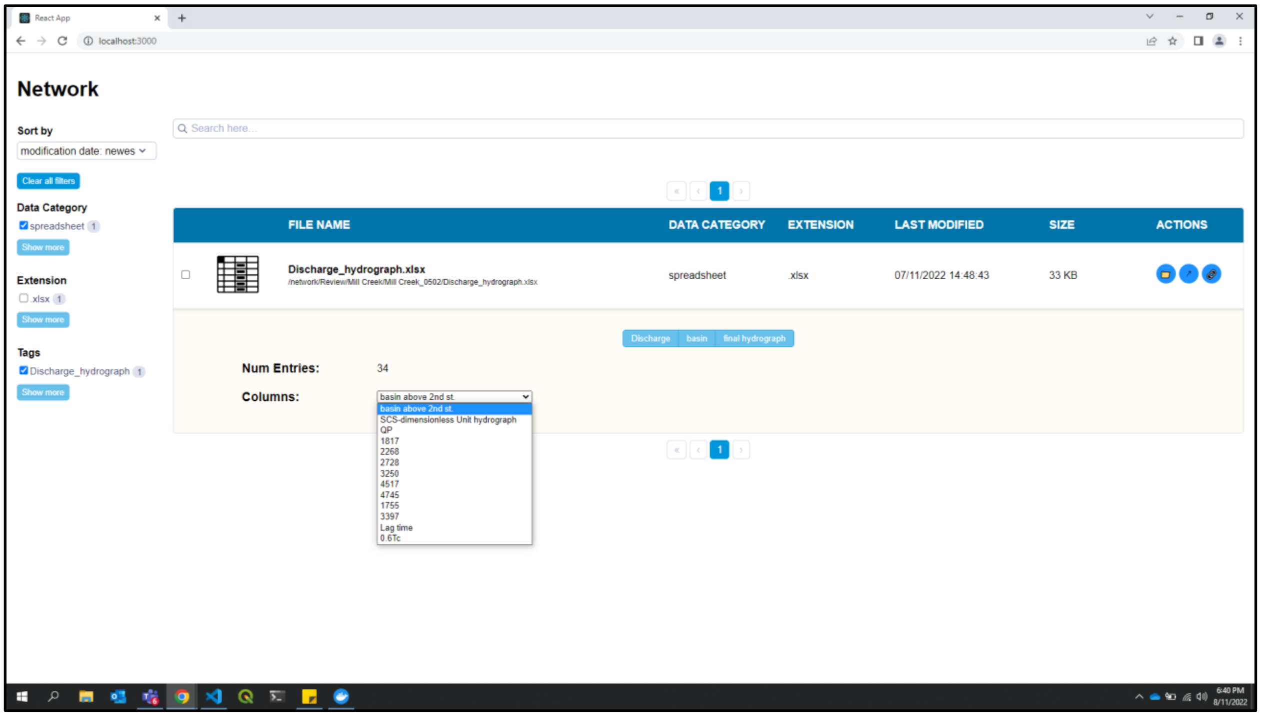
```
from_bucket_view
from_bucket_view/from_nld
from_bucket_view/from_nld/1405000241.gpkg
from_bucket_view/from_nld/1405000516.gpkg
from_bucket_view/from_nld/1405000241_metadata.geojson
from_bucket_view/from_nld/1405000516_metadata.geojson
from_bucket_view/grid.gpkg
from_bucket_view/json_files
from_bucket_view/json_files/cedar_rapids_rolling_windows_identifier.json
from_bucket_view/json_files/cedar_rapids_rolling_windows_identifier_hex14.json
from_bucket_view/json_files/h3_hex13_sensitivity_test
from_bucket_view/json_files/h3_hex13_sensitivity_test/cedar_rapids_rolling_windows_senstivity_test_identifier_hex13_30.json
from_bucket_view/json_files/h3_hex13_sensitivity_test/cedar_rapids_rolling_windows_senstivity_test_identifier_hex13_50.json
from_bucket_view/json_files/h3_hex13_sensitivity_test/window_1324.json
from_bucket_view/NFHL_02_20211104_clean.gpkg
from_bucket_view/test_polygoize.gpkg
from_bucket_view/xfer4metstat
from_bucket_view/xfer4metstat/us_msl-qpe_20220610_1800_001hr.nc
from_bucket_view/xfer4metstat/us_msl-qpe_20220610_1900_001hr.nc
from_bucket_view/xfer4metstat/us_msl-qpe_20220610_2000_001hr.nc
```

	FILE NAME	DATA CATEGORY	EXTENSION	LAST MO
☐	**tests.zip** /network/tests.zip			07/19/2022
☐	**from_bucket_view** /network/from_bucket_vie			07/19/2022
☐	**from_bucket_view** /network/from_bucket_vie			07/19/2022

Data Card of Archive File Type

We chose ours to resemble an ecommerce site for the network-lib, where files could be selected and added to a ‘cart’ to be reviewed or compared in-depth. To do this, these files would expand vertically like an accordion when clicked, displaying information unique to its data category and, if applicable, an image of the geometric bounds on a map

We incorporated Meilisearch filtering and sorting capabilities into a left panel. This allows the user to filter based on data category, file extension, and tags, as well as sort based on modification date and file size.



Network

Sort by

modification date: newes

Clear all filters

Data Category

☑ spreadsheet 1

Show more

Extension

☐ .xlsx 1

Show more

Tags

☑ Discharge_hydrograph 1

Show more

Search here...

		FILE NAME	DATA CATEGORY	EXTENSION	LAST MODIFIED	SIZE	ACTIONS
☐		**Discharge_hydrograph.xlsx** /network/Review/Mill Creek/Mill Creek_0502/Discharge_hydrograph.xlsx	spreadsheet	.xlsx	07/11/2022 14:48:43	33 KB	

Discharge basin final hydrograph

Num Entries: 34

Columns: basin above 2nd st.

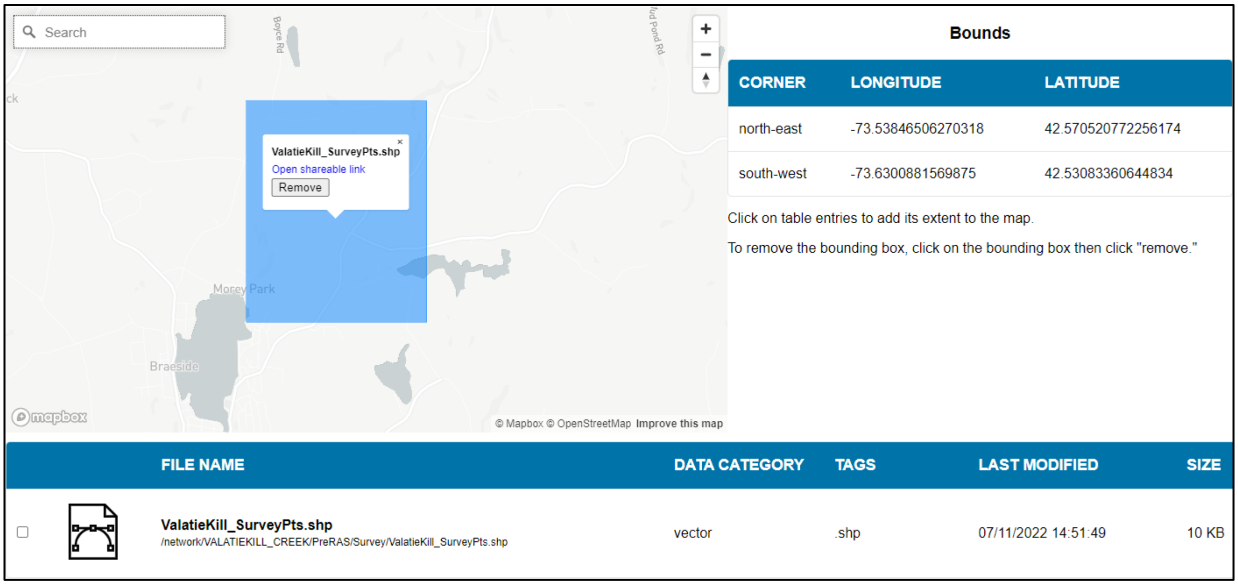
- basin above 2nd st.
- SCS-dimensionless Unit hydrograph
- QP
- 1817
- 2268
- 2728
- 3250
- 4517
- 4745
- 1755
- 3397
- Lag time
- 0.6Tc

Data Card of Excel Worksheet

Front-End

We used React to build our user interface. The instantsearch library is useful in providing components that easily work with the Meilisearch API to give useful searching and filtering functionality. This library provided checkbox filters, sorting, and a search bar for our product. As the wireframe developed, we built custom components and added new libraries to better help achieve our goal.

One of these advanced features is a geo search page capable of displaying the bounding boxes on a unified map for raster and vector files. It also performs an automatic filter on the geographic files based on what part of the map is displayed.


Search
Boyce Rd
ValatieKill_SurveyPts.shp
Open shareable link
Remove
Morey Park
Braeside
mapbox
© Mapbox © OpenStreetMap Improve this map
+
−

Bounds

CORNER	LONGITUDE	LATITUDE
north-east	-73.53846506270318	42.570520772256174
south-west	-73.6300881569875	42.53083360644834

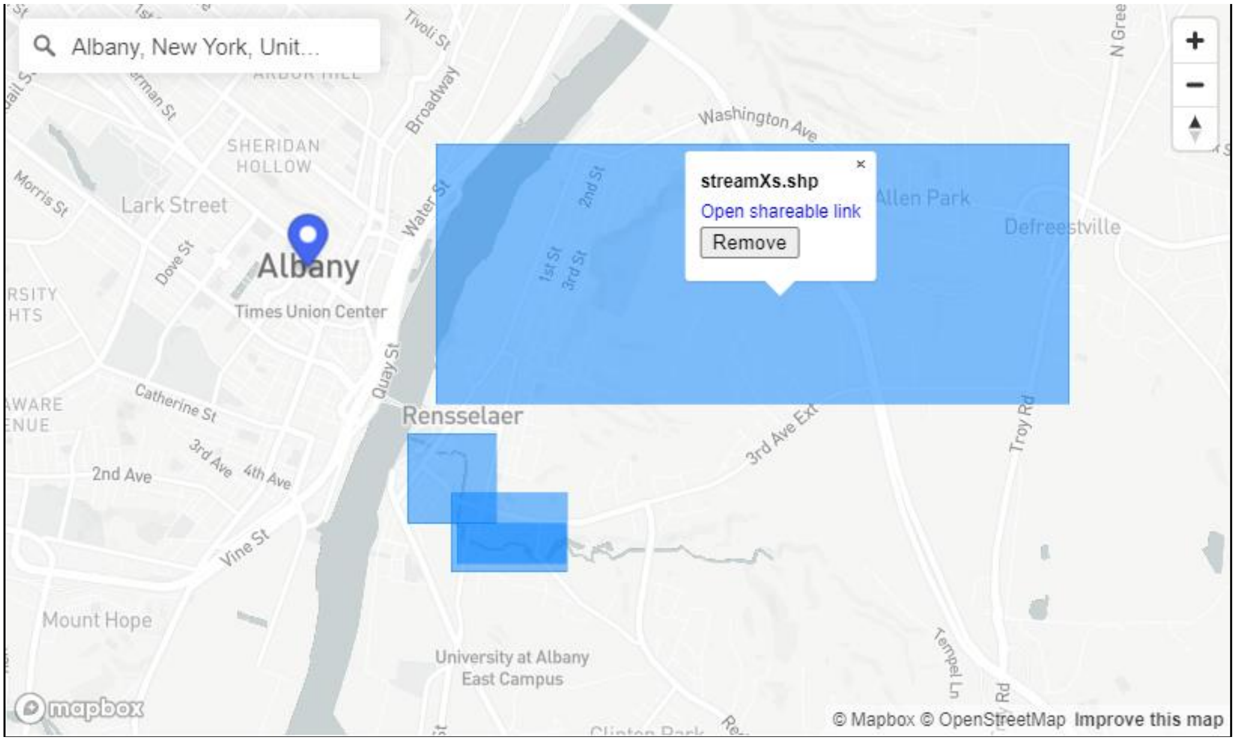
Click on table entries to add its extent to the map.

To remove the bounding box, click on the bounding box then click "remove."

		FILE NAME	DATA CATEGORY	TAGS	LAST MODIFIED	SIZE
□		**ValatieKill_SurveyPts.shp** /network/VALATIEKILL_CREEK/PreRAS/Survey/ValatieKill_SurveyPts.shp	vector	.shp	07/11/2022 14:51:49	10 KB

Geosearch Page with One Bounding Box

A great use-case for this is to use the geocoder to find all the relevant files in a particular location (Albany, NY was used above). From there, the bounding boxes of each file can be overlaid on the map to get a good sense of where the data is being represented.



Geosearch Page with Multiple Bounding Boxes

Takeaways

Lessons Learned

In future projects, we'd like to have an in-depth model of the project layout- as we weren't able to take as much time in this step because of the deadlines that were outlined for us. We've learned that communication is most important when there are roadblocks. It saves time and effort collaborating with your team to tackle those issues, rather than to attempt resolving them individually.

Future Developments

The first release of this product introduced basic functionality, including searching, sorting, and filtering on basic attributes, and advanced features. The data is presented in a tabular format using an on-click dropdown to view further information.

In the future, we hope to add a shopping cart feature, where the user will select a list of files they would like to compare and open them in a new window. Additionally, the user will also be able to compare the selected files on the Geosearch page. With this, we intend to support file types and link gis database cards to their vector and raster tables.

Special Thanks

We would like to thank the Dewberry CSI team for all their help and support throughout this project. We were given the opportunity to develop something multifaceted and challenging, while also having a long-term purpose and impact. We look forward to seeing this project's evolution and are grateful to have been a part of it. HAGS! –your favorite interns 😊

Team Members

Justin Paul

Justin Paul (He/Him) · 1st
Computer Science Student at University of Michigan

My internship was from May 31 – August 12. I primarily worked on the backend and the geo-search page.

Shivam Asija

Shivam Asija (He/Him) · 1st

Computer Science Student at California Polytechnic State University - San Luis Obispo

My internship was from June 13 – September 16.I primarily worked on the front-end portion of the website.

Sonia Khan

Sonia Khan (She/Her) · 1st

Undergraduate Student at George Mason University

My internship was from May 23 – August 12. I primarily worked on the design of the front-end portion of the website.