Creating and Evaluating a Collaborative Web Mapping Platform

Pennsylvania State University – MGIS Program
Venango County – Regional Planning Commission

Prepared by Alexandria Shreffler
Advised by Jan Oliver Wallgrün

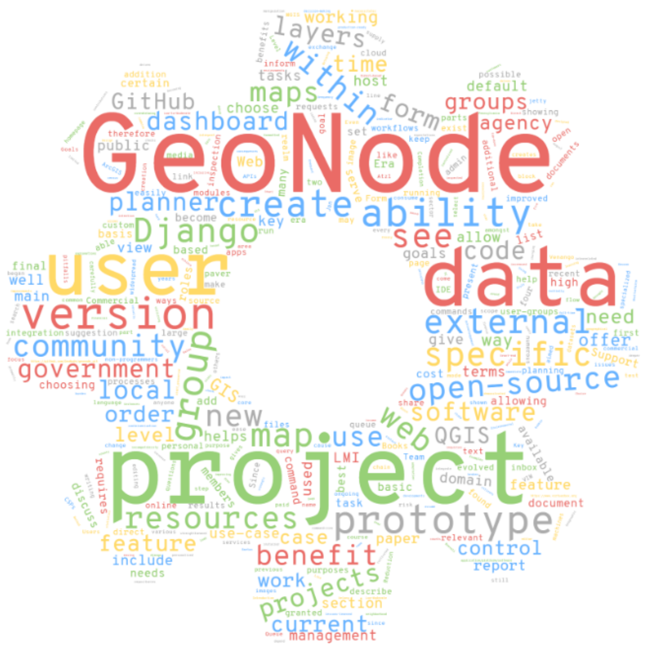
This assignment was produced on August 2nd, 2021 in order to satisfy the requirements for GEOG 596B: Capstone as part of the MGIS program.
Version 3

Abstract

If anything has been learned in pandemic times, it is the importance of remote access, virtualization, and web-based resources. Even prior to last year, collaborative mapping was recognized for its capability to facilitate synergistic data access and reduce complicated workflows. GeoNode is a web-based open-source software technology stack used to deploy geospatial content management systems to support Spatial Data Infrastructures (SDI). This paper will discuss the use and customization of GeoNode as an open-source project to support collaborative mapping in local governance through a GIS web interface. The intended audience of this paper includes active developers of GeoNode, potential developers both new and experienced who are interested in using GeoNode for data management and collaboration, and members of local government authority who have the power to organize the resources to implement a collaborative GIS system in their communities. The method for developing a customized GeoNode prototype is described within the context of local government deployment. An evaluation of GeoNode includes a critique of the platform's ability to support basic collaborative functions. Finally, recommendations are given to the audience based on the development experience with the prototype, the GeoNode community, and future of the GeoNode project.

Keywords: Collaborative Mapping; GeoNode; Community; Local Government; Web Mapping

This paper can be accessed and shared through the following link: https://tinyurl.com/GeoNodeEvaluation2021



Creating and Evaluating a Collaborative Web Mapping Platform

Introduction

Collaborative mapping has emerged as a web mapping trend capable of facilitating synergistic data access and reducing complicated workflows. Some examples of collaborative mapping functionality might include the generation of map integrated information such as task requests, external documentation, global and local communication, and queue completion. This functionality requires the adoption of a user structure designed to give each role their own permissions and access based on their individual task needs. The overarching goal of this project tested the feasibility of developing the mentioned capabilities in the form of a mapping platform to be evaluated, attempted, documented, and discussed in terms of theoretical stakeholder benefit. For this project, the use-case scenario was aimed at improving local government workflows, including both internal and external members of the typical daily processes. This broad goal was accomplished in four parts: 1) Identify any potential existing projects which could serve as a *framework* to support collaborative mapping, 2) Develop a working *prototype* with stakeholder customizations, 3) Provide an *evaluation* of the collaborative framework as a platform for providing collaborative functionality, and 4) Distribute video and written *documentation* of the user and developer experiences along with the specifications of the customized prototype.

After completing a project discovery search, the open-source project, GeoNode (https://github.com/GeoNode/geonode), was identified as a potential candidate for development to be evaluated as Free and Open-Source Software (FOSS) based on its support, investments, and ability to offer functionality for collaborative mapping platforms. Once a custom GeoNode project was created and customized, further insight into the GeoNode project was gleaned and reported. The lasting contributions of this project aimed to provide resources and ongoing support to GeoNode user groups who would benefit from demonstration literature and documentation. Recommendations will be given based on the evaluation of GeoNode in three main aspects: 1) ease of development, 2) level of support for key features, and 3) invested resources. The dual purpose of these recommendations will be to inform both potential future users and the community of GeoNode developers in an effort to encourage the exploration of collaborative solutions.

The following report will describe the project in five sections. Section 1 will describe the background and motivation for the project while Section 2 will provide the method for developing a prototype in GeoNode and illustrate the use-case scenario. The evaluation of GeoNode as a collaborative framework will be presented in Section 3. Section 4 will discuss recommendations for GeoNode users and developers, and then Section 5 will discuss the relevance of the project and future work to be done followed by some final conclusions.

Background & Motivation

Traditional Web Mapping

Since the creation of the World Wide Web in 1989, web mapping has evolved to give internet users the ability to view and interact with maps. Veenendaal, Broveli, and Li (2017) describe the history of web-mapping in nine main eras, with each subsequent era building on the accomplishments of the previous. These eras were spurred by significant events critical to the evolution of web mapping (see *Figure 1*). Maps began in the Static Era as HTML images, usually hyperlinks, which could be stored, retrieved, and shared more easily than paper maps. Eventually, they developed the ability to modify and customize maps in the Dynamic Era, which allowed users to interact with the map, such that they could manipulate their view and control the level of zoom. The Services Era ushered in the development of web-services and APIs allowed users to choose and contribute to the produced maps, and software applications could be linked to allow developers the ability to customize existing web maps. Examples of functionality gained during this era include geocoding and routing capabilities. The Interactive Era brought the use of new techniques, including image tiling, which enhanced response times for delivering web maps to users. The production of “mashups” allowed users to combine multiple map sources into one map (Cartwright, 2008; Haklay, Singleton, & Parker, 2008; Shaparev & Yakubailik, 2016).

By the time OpenStreetMap had been founded in 2004, web-mapping had reached the fifth era known as the Collaboration Era, where users could not only consume geospatial data, but they could also assist in its creation. Participatory mapping, described by terms such as crowdsourcing and Volunteered Geographic Information (VGI), evolved during this era to leverage the benefits of collective intelligence (Martin, Reynard, Pellitero Ondicol, & Ghiraldi, 2014; Gartner, 2009). After the Collaboration Era came others which launched such developments as digital globes, mobile and location-based functionality, cloud storage, semantic technologies, automation, and smart environments.

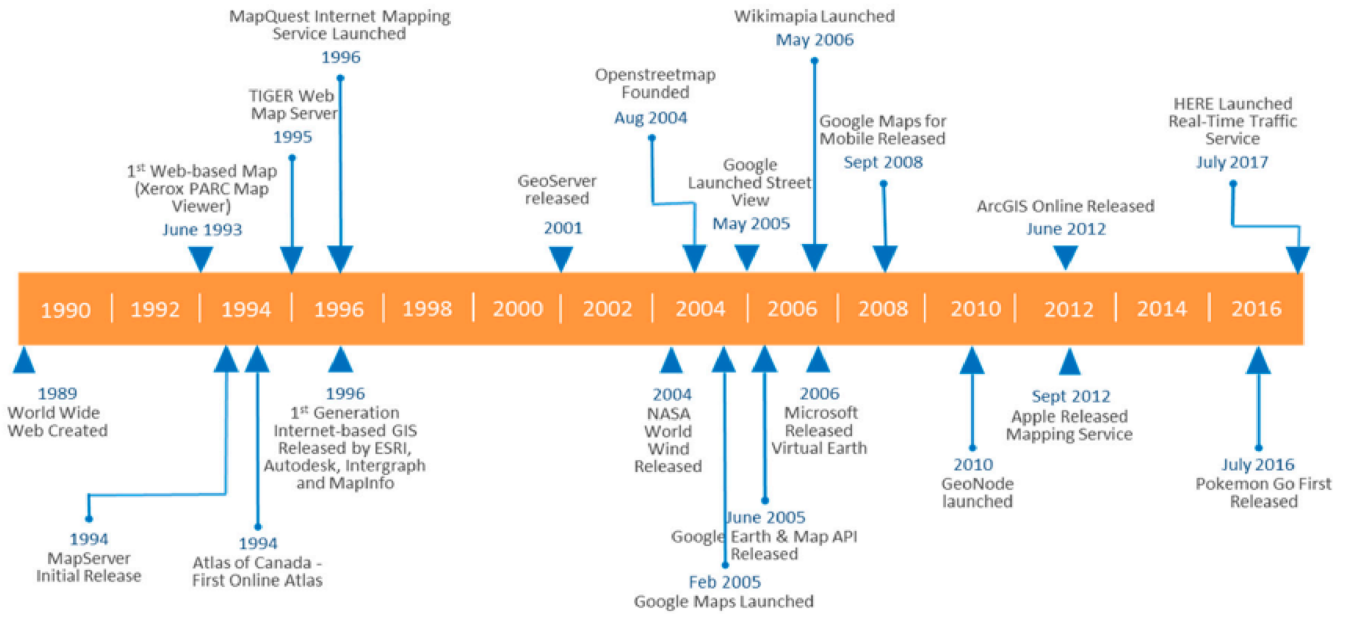

MapQuest Internet Mapping Service Launched 1996
TIGER Web Map Server 1995
1st Web-based Map (Xerox PARC Map Viewer) June 1993
GeoServer released 2001
Openstreetmap Founded Aug 2004
Google Launched Street View May 2005
Wikimapia Launched May 2006
Google Maps for Mobile Released Sept 2008
ArcGIS Online Released June 2012
HERE Launched Real-Time Traffic Service July 2017
1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016
1989 World Wide Web Created
1994 MapServer Initial Release
1994 Atlas of Canada - First Online Atlas
1996 1st Generation Internet-based GIS Released by ESRI, Autodesk, Intergraph and MapInfo
2004 NASA World Wind Released
June 2005 Google Earth & Map API Released
Feb 2005 Google Maps Launched
2006 Microsoft Released Virtual Earth
2010 GeoNode launched
Sept 2012 Apple Released Mapping Service
July 2016 Pokemon Go First Released

Figure 1. Timeline of significant web mapping events taken from Veenendaal, Brovelli, and Li (2017) for educational purposes only.

Collaborative Mapping

By recognizing that web maps can do more than just present geospatial information, developers have cultivated a community of users who understand the value of analyzing and exploring geospatial data in an online environment. Since web mapping trends show an increasing interest in developing online GIS functionality, user collaboration support, and cloud services (Veenendaal, Brovelli, & Li, 2017), it would be auspicious to offer collaborative functionality with increased personalization, cloud data management, and enhanced communication exchange.

Generally, personalization for specific roles and tasks depends on the user and their needs. Role-tailored mapping has been identified as a practical way to manage data permissions and give users the ability to view particular datasets of interest, maintain a default set-up query, submit annotations, and contribute to higher-level tasks over an integrated network (Cai, 2005; Pulsani, 2015; Tsou & Curran, 2008). As described by Foerster, Stoter, and van Oosterom (2012), this requires the development of user profiles which incorporate a combination of personal information (i.e. name, title), human-centered preferences (i.e. interface design), service-related information and preferences (i.e. access to specific data and functionality), and device-related information and preferences (i.e. PC, mobile phones, software). One example in the literature identifies the importance of developing role-tailored functionality by recognizing the negative impacts of media disruption in the supply-chain environment, which is the direct result of varying communication channels with no standardized way of sharing information (Atzl et al., 2019). The authors established a centralized application focused on easing communication and streamlining workflows within the supply chain. By creating a role-tailored map dashboard using open-source architecture and personalized applications, they were able to reduce media disruption by 80%, indicating an increase in overall information flow and task simplification.

Open-Source Development

Although Atzl et al. (2019) chose to develop their dashboard in an open-source environment, the result of their work was specific to the forest-based supply chain and their internal operations. In this case, the development of an adaptable open-source platform would encourage the widespread distribution and financial feasibility of adopting collaborative mapping tools in any field for both internal and external members. Commercial solutions offered by Esri in the ArcGIS realm exist to offer collaborative benefits to their users (Veenendaal & Dhliwayo, 2013), but they also come at an obstructive cost to some businesses and user groups. In the case of local government, it can become difficult to justify the high cost of these software products without tangible benefit to the community as a whole. Alternatively, open-source solutions often come with a monetary cost little-to-none. The benefits and consequences of open-source development have been investigated thoroughly in the literature (for example, Maurya, Ohri, & Mishra, 2015); however, one of the largest benefits of open-source is the ability to completely control specific features directly in the code and share the results freely with others.

Capstone Project Goals

The goals of the present project included four phases (see *Figure 2*). First, the identification of any potential existing projects, applications, or softwares which could serve as a *framework* to support collaborative mapping. Second, the development of a working *prototype* with stakeholder customizations. In this case, the stakeholder would be the local government sector with users both internal and external to the organization. Third, the generation of an *evaluation* of the collaborative framework as a platform for providing collaborative functionality. This evaluation will include discussion in terms of development ease, support level for key collaborative features, and resources invested in the final product. The final phase would be the distribution of a demonstration video and written *documentation* of the user and developer experiences along with the specifications of the customized prototype. Overall, this report discusses the ways that collaborative mapping can be accomplished in a single platform.

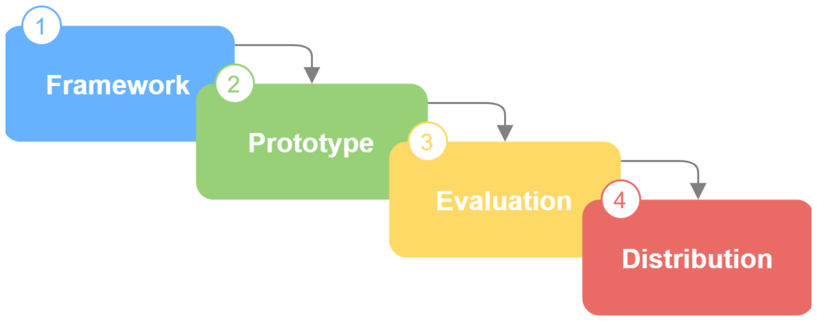


Figure 2. The flow of goals for the collaborative mapping project.

During the research process for this project, one application — GeoNode—stood out as an existing software and best candidate for development and evaluation for a few reasons. First, GeoNode has a large, responsive community of users and developers who tend to respond to questions within an hour of posting on the project forum. This presence would serve as a powerful resource when interacting with and developing a potential collaborative prototype. Second, GeoNode already utilizes a user-group structure to grant permissions and access to map layers. Finally, GeoNode's history as an open-source project demonstrates its longevity in the field of GIS web-mapping, thereby inspiring confidence in developers both new and experienced.

The GeoNode Project: Technical Organization

GeoNode is a web-based open-source software technology stack used to deploy geospatial content management systems (GeoCMS) to support Spatial Data Infrastructure (SDI). Working backwards on this definition, SDI is the entire framework used to create, exchange, and consume geospatial data, including the data itself, its metadata, the tools used to manipulate it, and the users who appreciate it. SDI is important for transparency, interoperability, cost-optimization, self-service, and avoiding the duplication of efforts (Buonanno, Zeni, Fusco, Manunta, Marsella, Carrara, & Lanari, 2019). "Content" in this case refers to objects such as images, articles, documents, commentary, and even the system users themselves. This content is rendered "geospatial" by recognizing its place in space using latitude and longitude coordinates. To meet this definition, GeoNode supports user profiles which "facilitate the use, management, and quality control of the data" (GeoNode Development Team, 2021). It combines the networking capabilities made popular by social media with the support of OGC standards, including WMS(-C), WFS(-T), ECS, and CSW as well as mass market search standards, which allow for its use with external desktop applications, such as QGIS. The application is structured using a host of open-source web mapping tools.

GeoNode was developed as a Django project, which is a Python-based Web Framework. The GeoNode project inherits principals from both Django and Python, including Don't Repeat Yourself (DRY) to improve efficiency, explicit statements and coding to improve readability and understanding, and loosely coupled architecture to prevent rigid dependencies between components (Rubio, 2017). Django terminology for applications- or "apps" for short - differs from the way it is commonly used. An application in the programming domain is typically thought of as a package of features and functionality, but in a Django project, an app is only one specific function within the project (Dauzon, Bendoraitis, Ravindran, Safari Books Online, 2016; Herbert & Safari Books Online, 2019; Lopatin & SpringerLink, 2020; Rubio & Safari Books Online, 2017). This differentiation is important to understand when learning the structure of a Django project like GeoNode. The admin dashboard is another helpful benefit of being a Django project since the GUI helps to relieve some command line development, making the GeoNode project more accessible to non-developers. The result is an intuitive user interface which integrates supported data stores, web services, and mapping clients (refer to *Figure 3*). It also offers the ability to create users and groups, further extended by granting particular users or entire groups permission to access certain resources.

In May of 2020, GeoNode rolled out a new official release, version 3.0. As of the time of this writing, the latest official release is version 3.2 (GeoNode Development Team, 2021). Some features relevant to this report include the integration of MapStore as the primary map viewer/composer, setting GeoLimits (restricting users or groups to a specific geographical area), improved message inbox and messaging, and other frontend improvements.

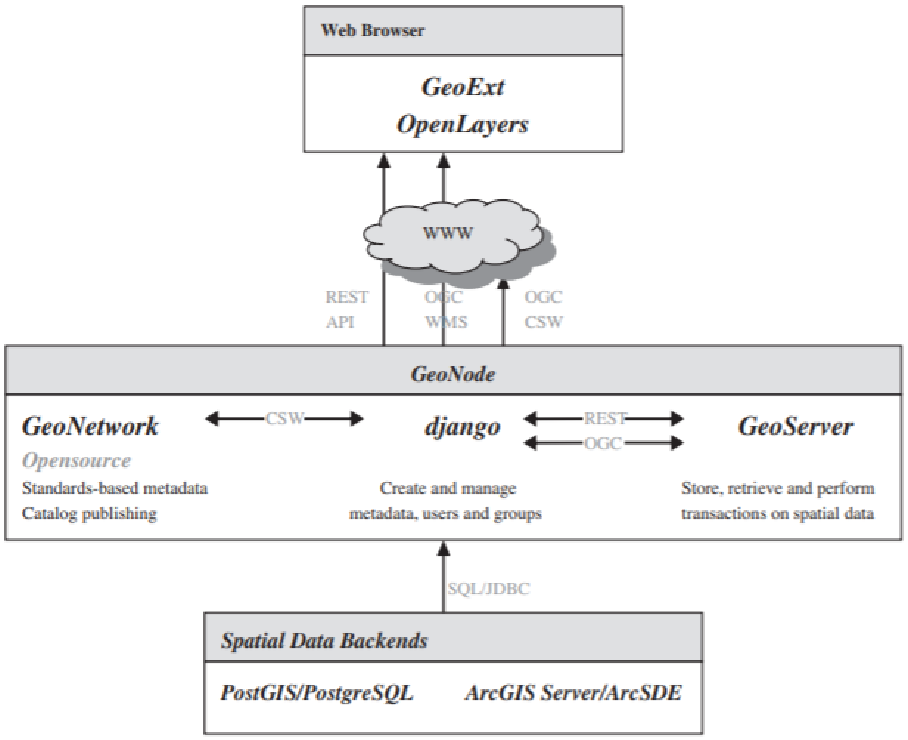


Figure 3. GeoNode architecture diagram taken from Bhattacharya et. al (2014) for educational purposes only.

GeoNode was initially developed by the founders primarily for disaster risk management, a field which necessitates the use of collaboration in order to accomplish goals quickly and efficiently. The Global Facility for Disaster Reduction and Recovery (GFDRR, 2020) used GeoNode to create the first InnovationLab, which hosted over 150 openly accessible datasets sorted into 12 different categories for better understanding disaster risk across the globe. Once registered, users can accomplish the following:

- Add, host, and share map layers, both in vector and raster format,
- Edit layer metadata,
- Style layers and geometries,
- Combine layers to create thematic web maps, and
- Assign permissions to other users and groups.

Although GeoNode offers a serviceable base for accomplishing a variety of GIS tasks (Balbo, Boccardo, Dalmasso, & Pasquali, 2014; Corti, Bartoli, Fabiani, Giovando, Kralidis, & Tzotsos, 2019; Steiniger, De La Fuente, Fuentes, Barton, & Muñoz, 2017), there are a few key elements which would need to be added to give users more collaborative control over their data. For example, how might a user be given the ability to select a feature and "assign" it to the queue of another user? As of the undertaking of this project, no such deliberate functionality has been proposed or published. One of the key purposes of this project will be to demonstrate a use-case scenario where certain key features can facilitate collaborative web-map project management.

GeoNode's Community Organization

The natural development of GeoNode can be described as bottom-up; the community of GeoNode developers, to which anyone can join at any time on a voluntary basis, drives the development of the platform. This would be juxtaposed against the development of a top-down open-source project, such as QGIS, which has a large, organized team of paid developers driving most of the project decisions. According to the most recent QGIS finance report (2019), at least 17 people were directly paid as core developers and 2 people were compensated for their work on documentation. To elaborate on the scale of this investment, that was approximately $100,000 of funding dispersal. Like QGIS, GeoNode is another one of the many projects managed under the OSGeo umbrella, but GeoNode does not have a formal budget and is therefore supported only by its own community of volunteers. Despite a lack of overhead organization, the GeoNode project does benefit from companies who choose to hire people specifically to develop GeoNode, known commonly as Commercial Support Providers (CSPs). The 2017 report lists approximately 20 commercial support providers as of 2016, though this list is noted to be incomplete (Global Facility for Disaster Reduction and Recovery, 2017). Like QGIS, the GeoNode project uses GitHub (www.github.com) to handle version control amongst a large group of developers. As a project hosted on the GitHub development platform, developers have the ability to quantify community using the GitHub platform's statistics, which counts the number of contributors, pull requests, forks, and open/closed issues. This provides a basis for comparison in terms of active community, frequency of bugs, and project turnover. A current comparison between QGIS and GeoNode is described in *Table 1*.

Table 1. Current comparison between QGIS, a well-known open-source GIS project, and GeoNode

	GeoNode	QGIS
Contributors[1]	224	439
Commercial Support Providers	12 CSPs[2]	17 core CSPs 7 commercial contributors 29 additional support providers[3]
Project Steering Committee Members	5 members[3]	6 members[3]

[1]www.github.com [2]https://geonode.org/providers/ [3]https://www.qgis.org/en/site/forusers/commercial_support.html#qgis-commercial-support

Prototype Design & Implementation

Developing in a Virtual Environment

Virtualization software divides a computer's hardware resources into two or more operating systems, allowing the user to test installations in a "bubble" environment not reliant upon, nor affecting the host computer's main OS. Virtualization can also be used to run multiple processes with potentially more efficiency. In other words, more software and processes with the same amount of hardware. Generally speaking, virtualization can be useful for disaster recovery, load balancing, scalability, running legacy applications, and addressing software-OS incompatibility. For the purposes of hosting GeoNode, virtualization is necessary in cases where the host OS is unsupported by the current installation methods (i.e. Windows was not currently a supported OS for direct GeoNode installation at the time of this project's initiation). It is also important to note that users of GeoNode will not need to know about virtualization in order to access the custom application. *Table 2* provides an overview of the prototype specifications.

Table 2. Overview of prototype specifications for the collaborative mapping project.

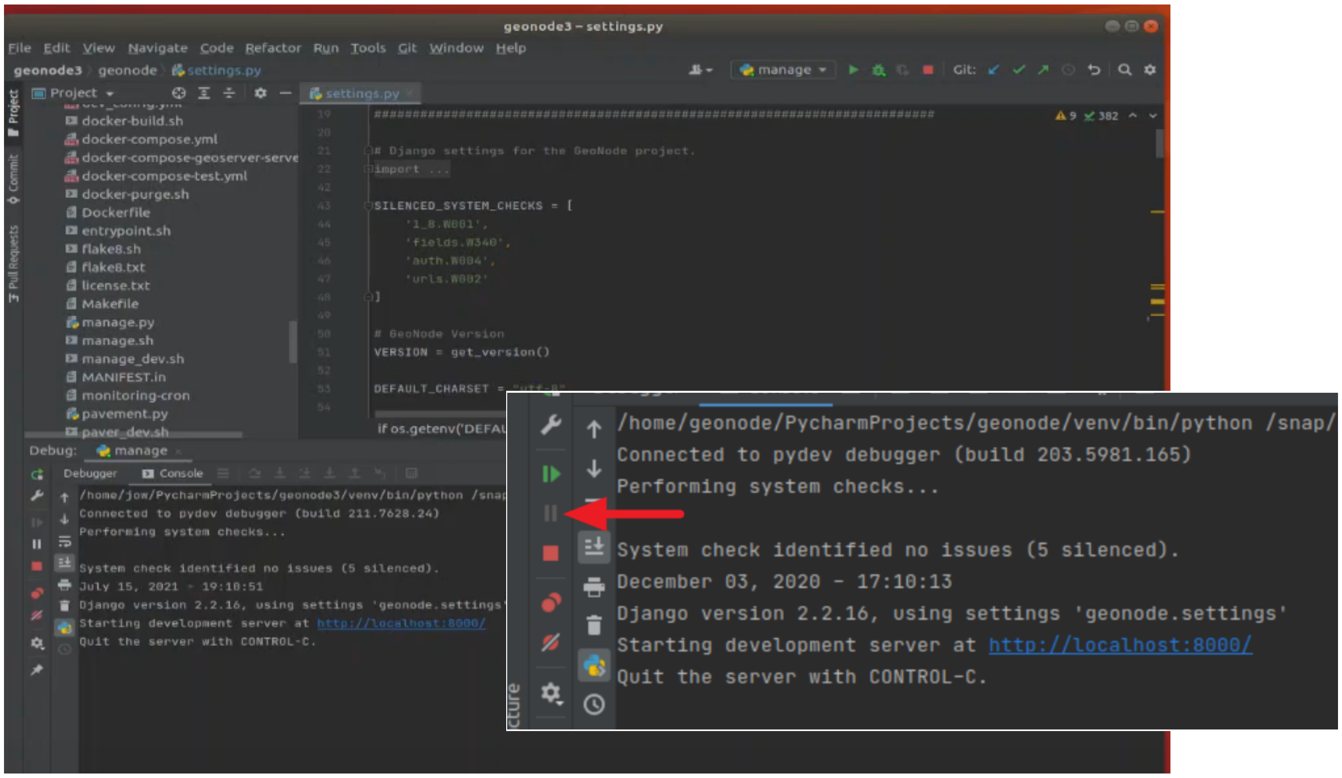
Variable	Specification
Installation Method	https://github.com/gannebamm/geonode-workshop/tree/main/00_getStarted
Host OS	Windows 10 Pro (Version 21H1), 64-bit
Virtualization Software	VirtualBox https://www.virtualbox.org/ (Version 6.1)
Virtual Machine OS	Ubuntu 18.04.5 LTS, 64-bit
Clone Version	Retrieved from https://github.com/GeoNode/geonode.git on 5/20/2021, refreshed on 7/9/2021
IDE	PyCharm Community Edition (Version 2021.1.3)

Installation Methods

Installation of the platform followed instructions posted on GitHub by a regular contributor to the GeoNode project (Hoedt, 2020). A static version of this workshop will be included in the final project documentation literature. Translator libraries allow for the manipulation of geospatial data, xml file parsing, version control, language libraries (including C and Python) which allow communication with PostgreSQL, Django, and SQLite. There are prerequisites for managing the repositories where software is being installed from, depending on the OS. The order that these commands are executed is important because each of these packages has dependencies that also need to be installed or updated. For example, GeoServer can be executed multiple different ways using different prerequisites, such as through either a jetty or tomcat Java servlet. Though there are many existing installation methods and supporting documentation, the line-by-line installation used for this project breaks down each step into parsable movements.

Test & Debug

Once GeoNode has been installed, the GeoServer instance can be run on localhost 8080 using the paver command: `paver start_geoserver`. Not to be confused with the environment created for the virtual machine, a Python virtual environment set up using PyCharm isolates GeoNode to its own project, preventing other Python projects and installation packages from becoming corrupted or changed by accident, allowing different projects to run separate installation versions. When the virtual environment is running correctly, the debug configuration can be created and activated, which starts the GeoNode installation on the specified localhost, which is port 8000 in this case (see *Figure 4*). Navigating to this destination in the browser brings up the generic GeoNode homepage (see *Figure 5*).



```
##########################################################################

# Django settings for the GeoNode project.
import ...

SILENCED_SYSTEM_CHECKS = [
    '1_8.W001',
    'fields.W340',
    'auth.W004',
    'urls.W002'
]

# GeoNode Version
VERSION = get_version()

DEFAULT_CHARSET = "utf-8"
```

```
/home/jow/PycharmProjects/geonode3/venv/bin/python /snap
Connected to pydev debugger (build 211.7628.24)
Performing system checks...

System check identified no issues (5 silenced).
July 15, 2021 - 19:18:51
Django version 2.2.16, using settings 'geonode.settings'
Starting development server at http://localhost:8000/
Quit the server with CONTROL-C.
```

```
/home/geonode/PycharmProjects/geonode/venv/bin/python /snap/
Connected to pydev debugger (build 203.5981.165)
Performing system checks...

System check identified no issues (5 silenced).
December 03, 2020 - 17:10:13
Django version 2.2.16, using settings 'geonode.settings'
Starting development server at http://localhost:8000/
Quit the server with CONTROL-C.
```

Figure 4. Debugging activation with detail on the check prompt. Sanity checks can be performed by utilizing the pause feature pointed out above.

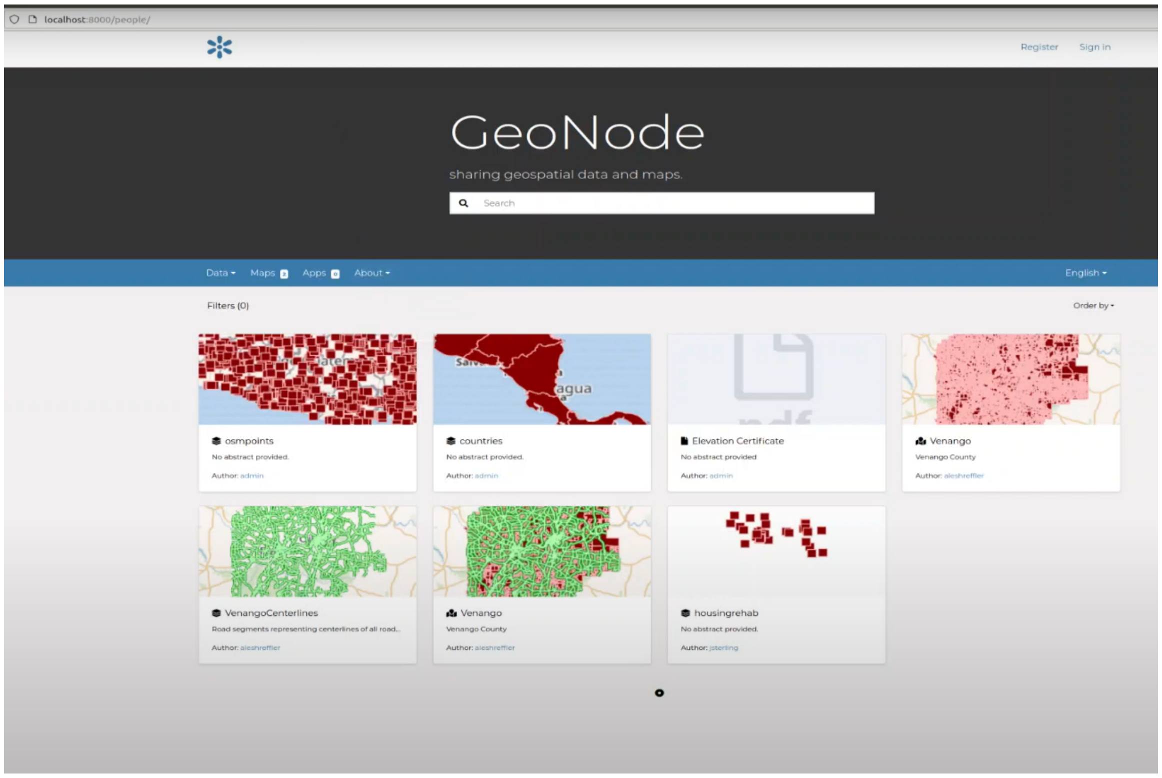


Figure 5. A screenshot of the prototype showing the GeoNode homepage with added layers.

Development Strategies

The Django admin dashboard (see *Figure 6*) allows the administrator to interact with customizable modules outside of the line-by-line code. The user interface offers such modules as theme creation, group organization, bulk operations, usage monitoring, etc. A complete list can be found on the GeoNode documentation website under the GeoNode Admins Guide section (GeoNode Development Team, 2021). However, there may still be instances where changing the code is necessary. By using an Integrated Development Environment (IDE) such as Pycharm, admins can take advantage of all the well-documented Django modules to further customize their dashboard. With reference to the code more generally, it will become necessary to merge with the main GitHub branch to integrate critical updates without rewriting all the personal code. In terms of this prototype, the Pycharm IDE offers a simple workflow to do this quickly and efficiently, allowing the user to see real-time, line-by-line differences between the branch and the main repository.

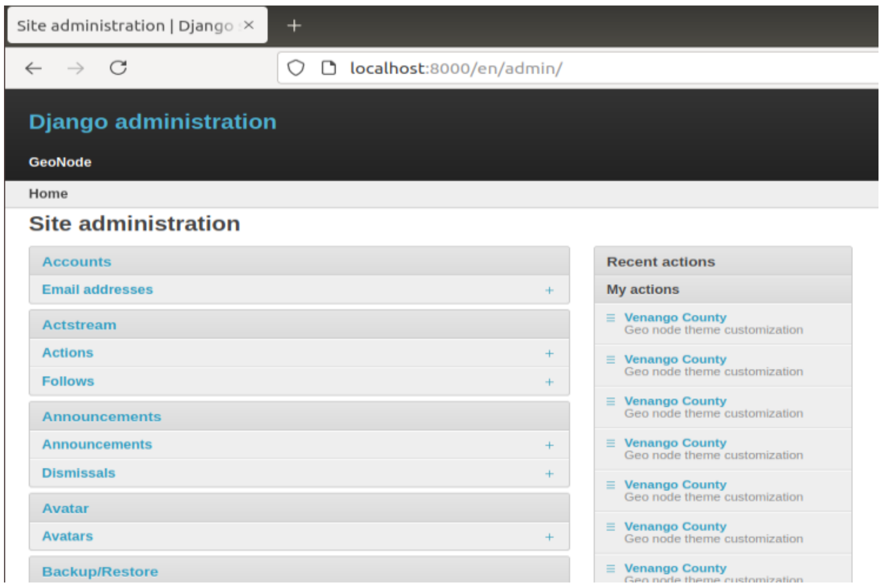


Figure 6. A screenshot of the prototype showing the Django Administration dashboard.

Use-Case Scenario: Realizing collaborative features with GeoNode

To review the general functionality offered, GeoNode users have the ability to 1) query data and metadata, 2) upload and edit a variety of data types, including vector and raster files, PDF files, images, etc., 3) customize permissions on their data, 4) perform data visualization and integration, and 5) interact with other users and user-groups. Users can create their own login roles to use the customized GeoNode platform, or the administrator can create their roles for them. In the use-case scenario for the prototype, the administrator has an idea of the user structure at the organization (Venango County) and therefore can set up the "parent" users (department heads) with their own groups (department) into which particular "child" users (employees) can subscribe. If a user does not belong to an internal department group, the administrator can either assign them to a public "citizen" group or choose not to assign them to a group and assume that the default "no group" option describes the general public. This user-group structure helps to organize levels of access to particular types of data (see *Figure 7*). For example, there may be sensitive data that should stay within the county and therefore is restricted only to certain department groups. Perhaps certain data may need to be more restrictive to only one department group. The users have their own control over who has access to their data, as well as the maps they create with that data. The structure illustrated in *Figure 7* also helps to speculatively identify how collaborative functionality can be introduced to the GeoNode platform. Collaborative functionality implies users must interact with one another in order to achieve a common goal or complete a dedicated workflow. The functionality listed in *Table 3* was selected based on the level of importance for achieving collaboration globally, the impact the feature would have on multiple users and user-groups, and the particular need when theoretically applied to the prototype use-case scenario for local government.

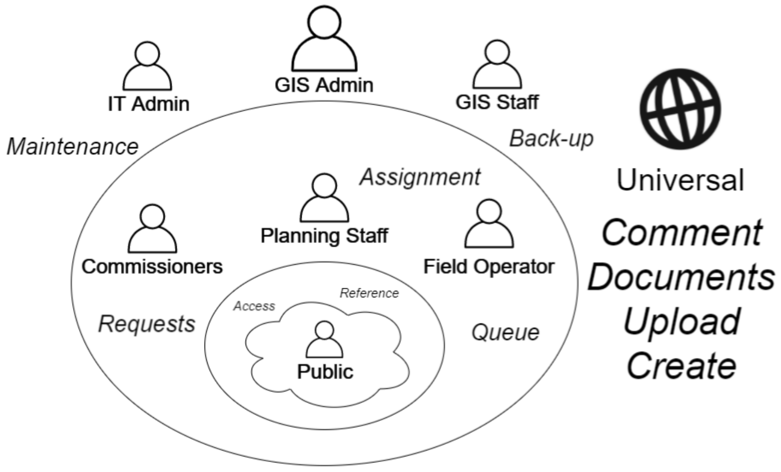


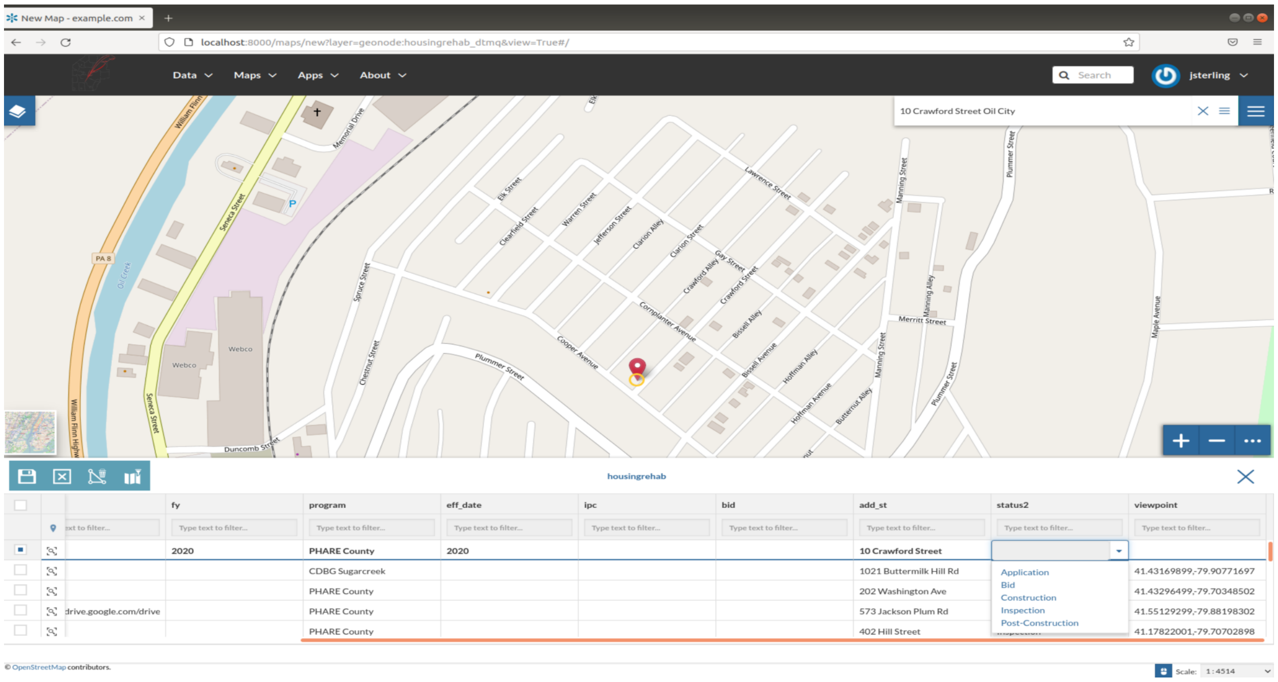
Figure 7. An overview of the theoretical user structure roles for a local government organization as well as their respective levels and proposed functionality (*italicized*).

Table 3. Summary descriptions of collaborative functionality within the scope of the collaborative mapping project.

Collaborative Feature	Justification for Inclusion
Form Submission	Collection of information from users within the platform eases the burden of data entry, directly providing basis of connection between project to-be-addressed and user-workflow structure
Feature Assignment	Integrated ability to select a feature and isolate it with the intention of bringing it to the attention of a particular user or group
Queue Completion	Bringing the feature assignments into a specialized layer for review by the assigned user or user-group
Documentation	Centralized location to store project documents where they are accessible to all relevant users and user-groups
Communication	Direct, clear mode of providing direction, scheduling updates, consolidating project notes, pinning key information, etc.

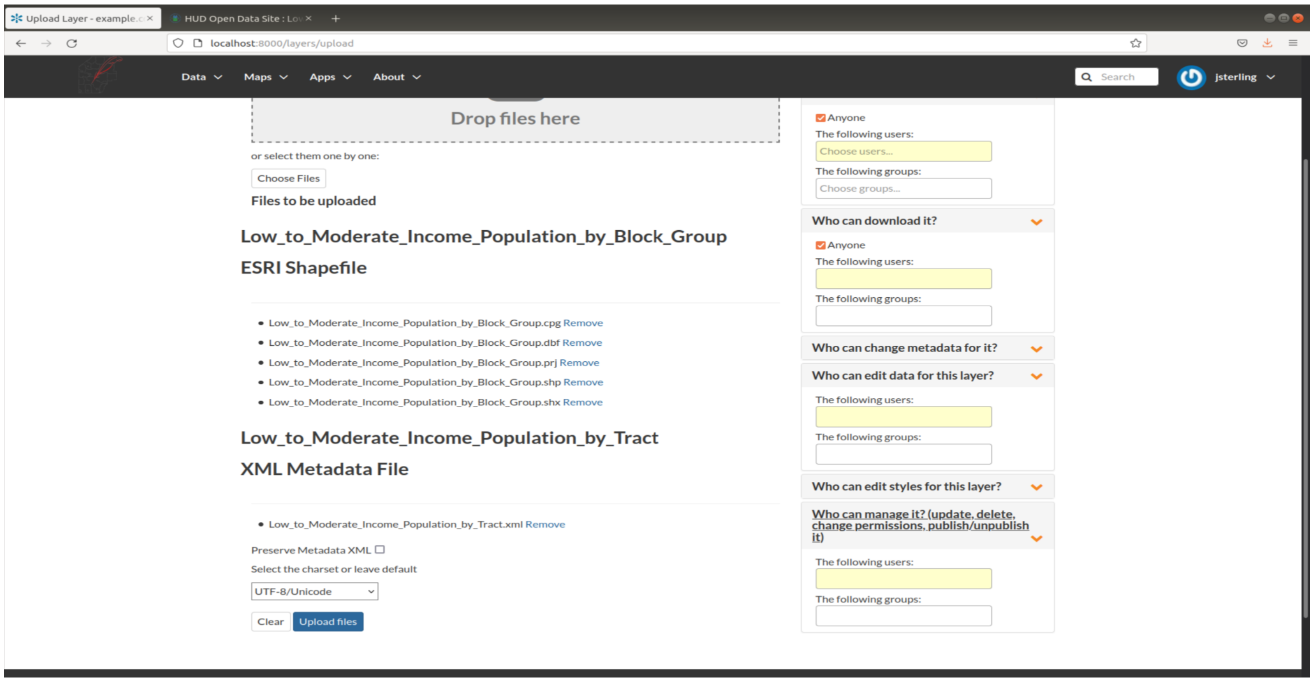
Let’s discuss a typical workflow within the application domain: the internal and external data management and collaboration between local government stakeholders and its citizens. As a member of the planning department staff, the Community Development Planner needs to document the Low-Moderate Income (LMI) percentages of a particular neighborhood to inform a Community Development Block Grant (CDBG) project. These grants have specific guidelines to awarding the grant, including eligibility for areas below a certain LMI percentage. The initial stage of project planning requires the verification of LMI below the threshold. In the established workflow, the Geospatial Analyst would be provided a spreadsheet of LMI values by block group with the directive to isolate the particular area of interest, format labels, symbolize, and return a static map for reference; this example is typical given an isolated desktop GIS environment. In a workflow optimized for collaboration with a specialized platform, users would be free to upload and create their own content. The established workflow could be improved using the online version of ArcGIS, but the subscription level has only a limited number of allowed users at a given cost; upgrading to a subscription with more allowed users would come at an additional cost. In a GeoNode project, users would be able to comment in-platform on any resource (to which they have access), including data layers and specific maps.

Instead of forming a request to the Geospatial Analyst, let's discuss how the Community Development Planner might use GeoNode instead. The Community Development Planner creates a project location and includes attributes such as contact information, grant program, and current status, as shown in *Figure 8*. Using Low-to-Moderate income census block group data uploaded from the Housing and Urban Development department (see *Figure 9*), a comparison overlay is illustrated in *Figure 10* to inform whether the project supports the improvement of communities of LMI. As part of Environmental Review, the Geospatial Analyst can overlay projects with environmental review variables (such as distance to airport, contamination sites, flood hazards, etc.) to determine whether the project is subject to regulation. Some resources of environmental review information are available as WMS, which is supported by GeoNode, allowing for seamless updates with external agency information, like the EPA contamination site map. The Home Inspector from an outside agency can create a profile and be granted access to project location data (see *Figure 11*). With additional customization, the login page could also include a dropdown menu to assign themselves to a particular group, such as a sub-role of the public group: Contractor. This would allow the Community Development Planner to map a new layer with only properties to be inspected, and then give only the Contractor group (or the particular agency) permission to view. The agency could then refer to and document status and scope of work for each property. General project documentation can be uploaded (process shown in *Figure 12*), linked to the housing rehab layer (see *Figure 13*) and made accessible by internal members of authority (planning director, commissioners, etc.) or registered external members with interest in the specific project or set of projects (homeowners, contractors, grant managers, municipal government, etc.). This accessibility can inspire communication, whether it be direction, inquiry, or acknowledgement (see *Figure 14*). In this scenario, all privileged users are empowered with the ability to identify all the projects within the program; this forms the motivation and basis for our local government use-case scenario.


New Map - example.com
localhost:8000/maps/new?layer=geonode:housingrehab_dtmq&view=True#/
Data
Maps
Apps
About
Search
jsterling
10 Crawford Street Oil City
housingrehab
© OpenStreetMap contributors.
Scale: 1 : 4514

	fy	program	eff_date	ipc	bid	add_st	status2	viewpoint
	2020	PHARE County	2020			10 Crawford Street		
		CDBG Sugarcreek				1021 Buttermilk Hill Rd	Application	41.43169899,-79.90771697
		PHARE County				202 Washington Ave	Bid	41.43296499,-79.70348502
drive.google.com/drive		PHARE County				573 Jackson Plum Rd	Construction	41.55129299,-79.88198302
		PHARE County				402 Hill Street	Inspection	41.17822001,-79.70702898
							Post-Construction	

Figure 8. The Community Development Planner creates a feature for a new housing rehab location and defines key attributes.



Drop files here

or select them one by one:

Choose Files

Files to be uploaded

Low_to_Moderate_Income_Population_by_Block_Group

ESRI Shapefile

- Low_to_Moderate_Income_Population_by_Block_Group.cpg Remove
- Low_to_Moderate_Income_Population_by_Block_Group.dbf Remove
- Low_to_Moderate_Income_Population_by_Block_Group.prj Remove
- Low_to_Moderate_Income_Population_by_Block_Group.shp Remove
- Low_to_Moderate_Income_Population_by_Block_Group.shx Remove

Low_to_Moderate_Income_Population_by_Tract

XML Metadata File

- Low_to_Moderate_Income_Population_by_Tract.xml Remove

Preserve Metadata XML ☐

Select the charset or leave default

UTF-8/Unicode

Clear Upload files

☑ Anyone
The following users: Choose users...
The following groups: Choose groups...

Who can download it?
☑ Anyone
The following users:
The following groups:

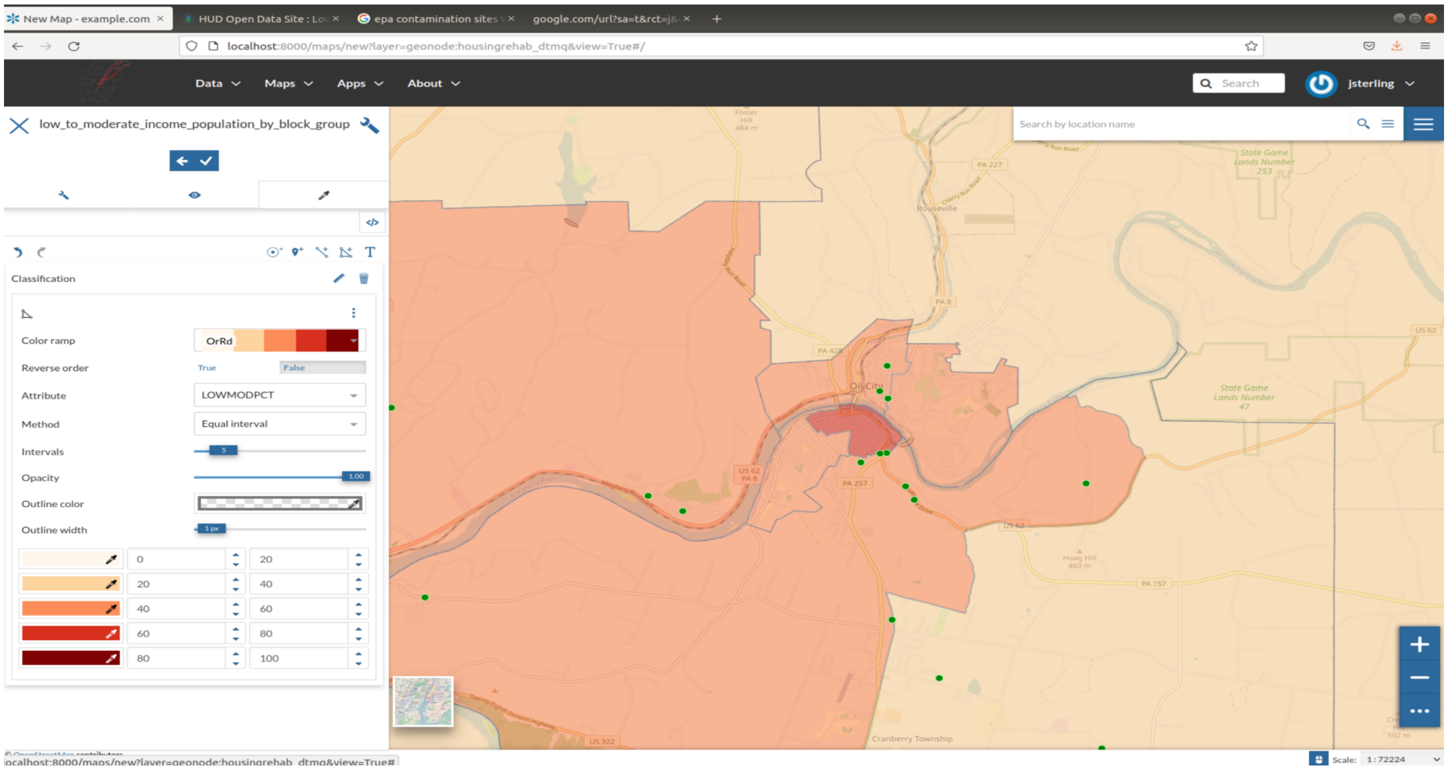
Who can change metadata for it?

Who can edit data for this layer?
The following users:
The following groups:

Who can edit styles for this layer?

Who can manage it? (update, delete, change permissions, publish/unpublish it)
The following users:
The following groups:

Figure 9. The Community Development Planner chooses a collection of files (a shapefile) to upload. Since this data is already publicly accessible, they choose not to restrict the data, allowing anyone to view or download it by default.



localhost:8000/maps/new?layer=geonode:housingrehab_dtmq&view=True#/

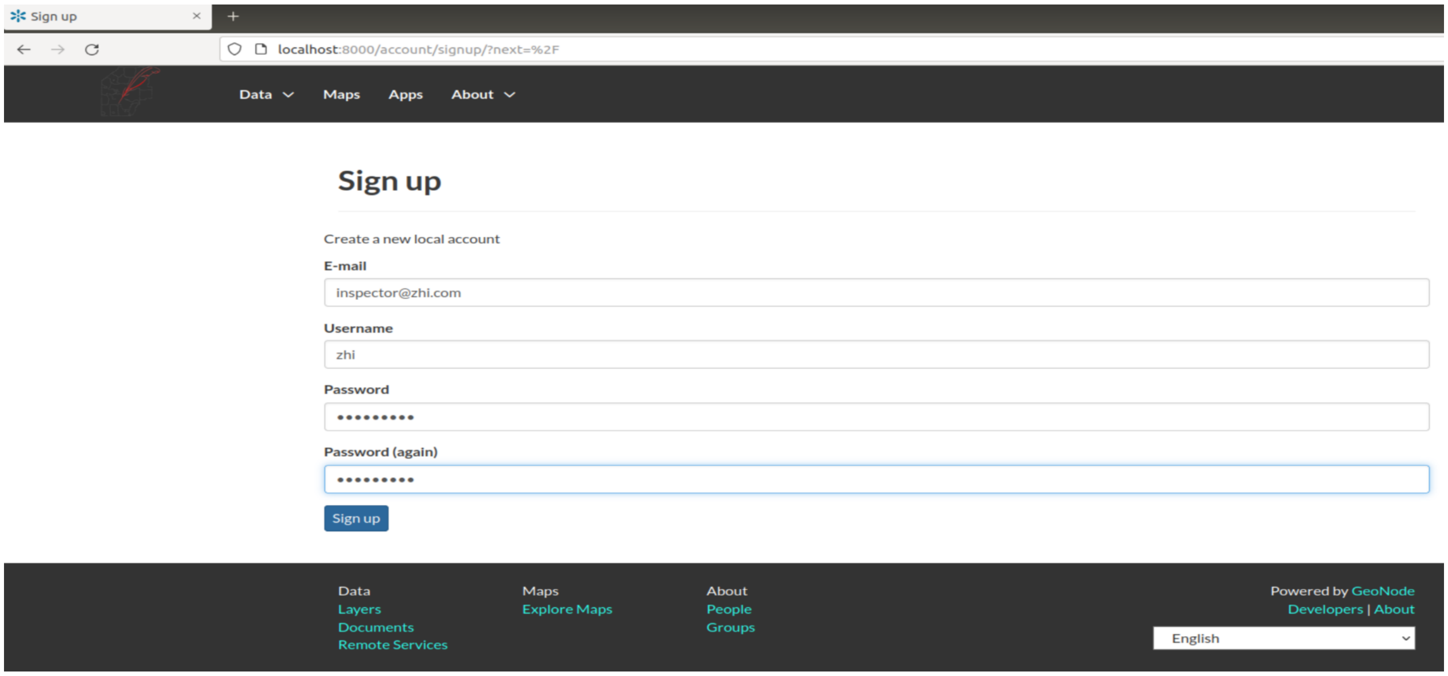
low_to_moderate_income_population_by_block_group

Classification

Color ramp	OrRd
Reverse order	True [✓ False]
Attribute	LOWMODPCT
Method	Equal interval
Intervals	5
Opacity	1.00
Outline color	
Outline width	1 px

⬜	0	20
🟧	20	40
🟧	40	60
🟥	60	80
🟥	80	100

Figure 10. An example of a symbolized classification map showing how layers can be stacked and customized in GeoNode.



localhost:8000/account/signup/?next=%2F

Sign up

Create a new local account

E-mail: inspector@zhi.com

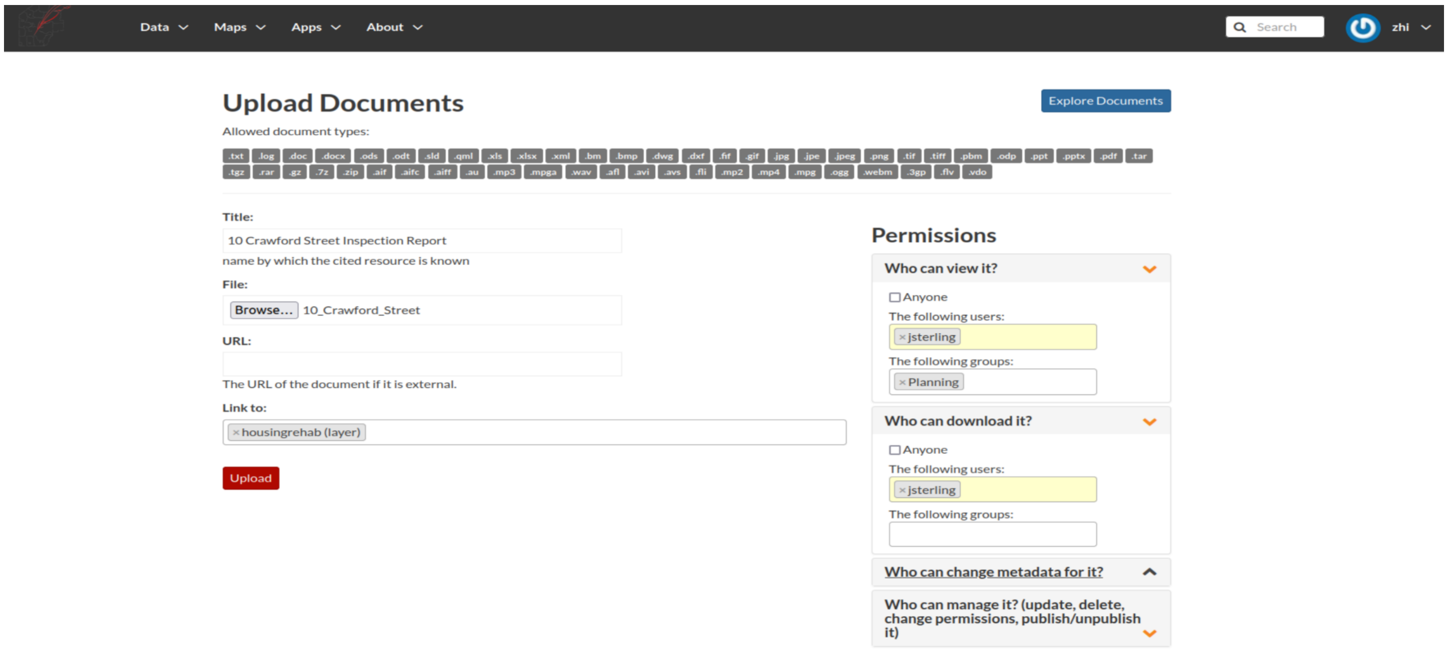
Username: zhi

Password: •••••••••

Password (again): •••••••••

Sign up

Figure 11. A screenshot of the enrollment page for a new user, the home inspection agency in this case.



Upload Documents

Allowed document types:

.txt .log .doc .docx .ods .odt .sld .qml .xls .xlsx .xml .bm .bmp .dwg .dxf .fif .gif .jpg .jpe .jpeg .png .tif .tiff .pbm .odp .ppt .pptx .pdf .tar .tgz .rar .gz .7z .zip .aif .aifc .aiff .au .mp3 .mpga .wav .afl .avi .avs .fli .mp2 .mp4 .mpg .ogg .webm .3gp .flv .vdo

Title: 10 Crawford Street Inspection Report

name by which the cited resource is known

File: 10_Crawford_Street

URL:

The URL of the document if it is external.

Link to: × housingrehab (layer)

Permissions

Who can view it?

☐ Anyone

The following users: × jsterling

The following groups: × Planning

Who can download it?

☐ Anyone

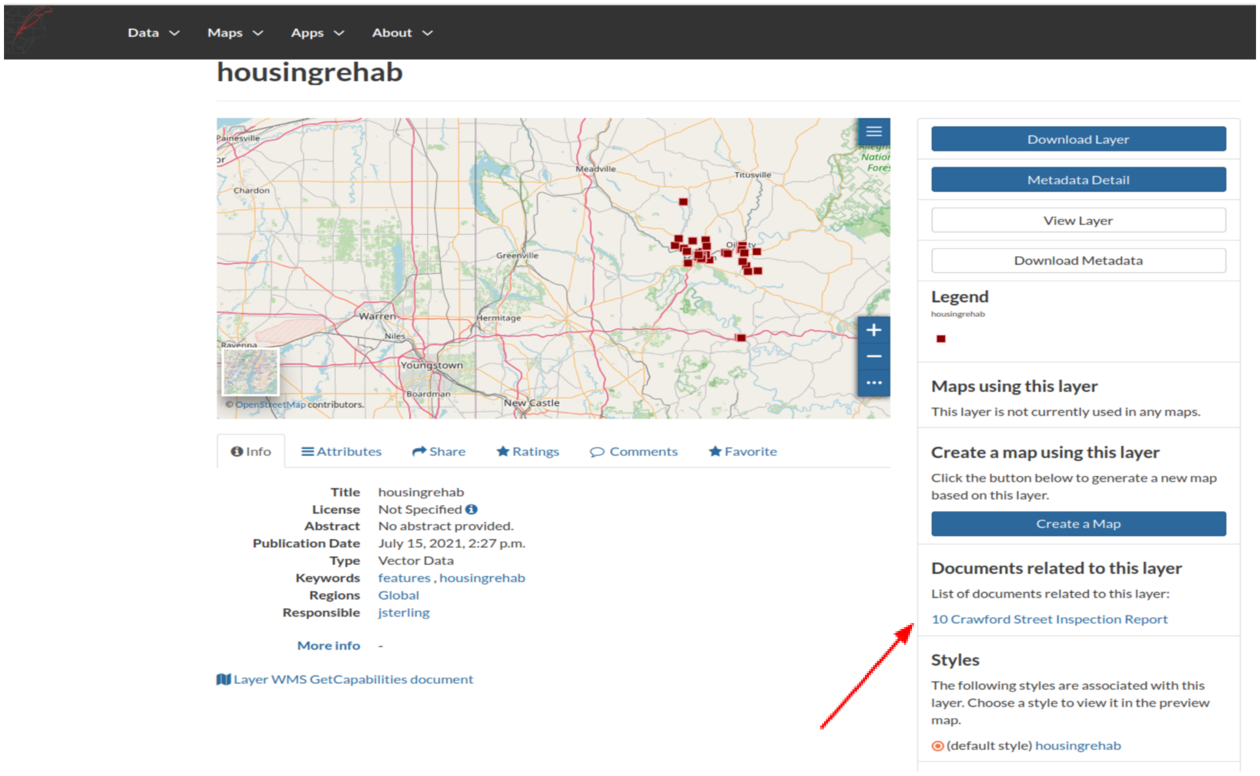
The following users: × jsterling

The following groups:

Who can change metadata for it?

Who can manage it? (update, delete, change permissions, publish/unpublish it)

Figure 12. The home inspection agency uploads a document, provides a link to the housing rehab layer, and gives the Community Development Planner (as well as the Planning department) various permissions.



housingrehab

Title	housingrehab
License	Not Specified
Abstract	No abstract provided.
Publication Date	July 15, 2021, 2:27 p.m.
Type	Vector Data
Keywords	features , housingrehab
Regions	Global
Responsible	jsterling
More info	-

Layer WMS GetCapabilities document

Legend

housingrehab

🟥

Maps using this layer

This layer is not currently used in any maps.

Create a map using this layer

Click the button below to generate a new map based on this layer.

Documents related to this layer

List of documents related to this layer:

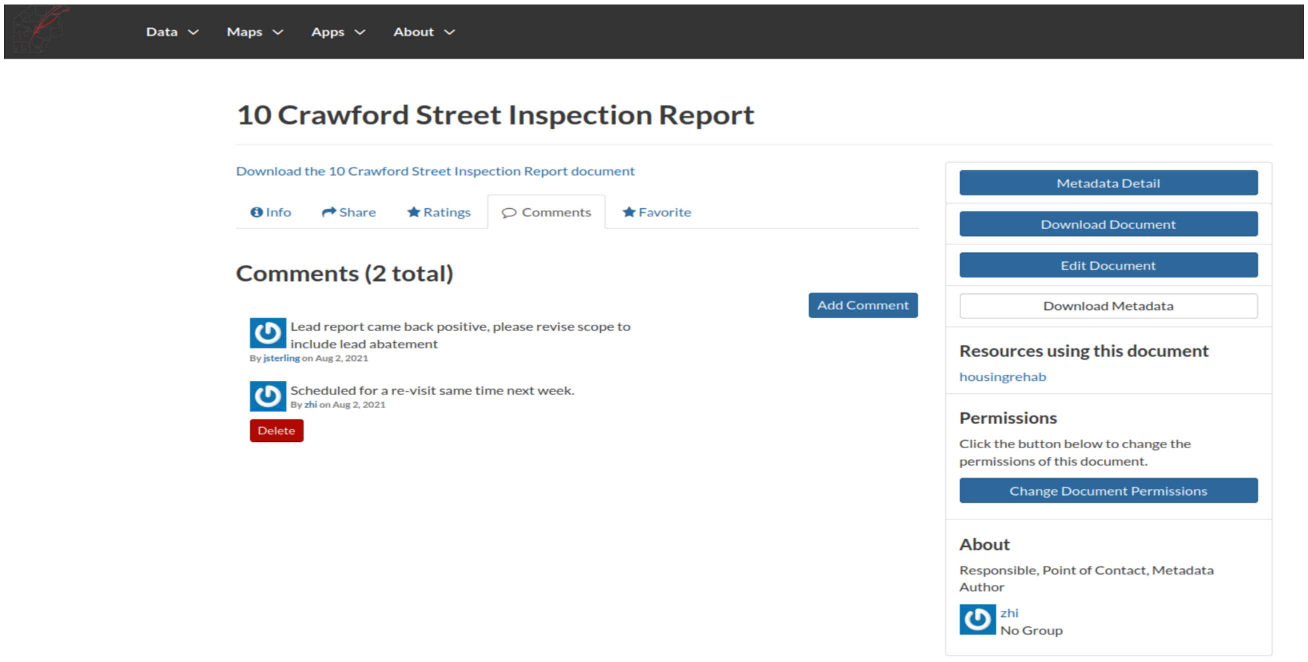
10 Crawford Street Inspection Report

Styles

The following styles are associated with this layer. Choose a style to view it in the preview map.

(default style) housingrehab

Figure 13. Landing page for the housing rehab layer that lists the newly uploaded related documents on the right-hand pane.



10 Crawford Street Inspection Report

Download the 10 Crawford Street Inspection Report document

Comments (2 total)

Lead report came back positive, please revise scope to include lead abatement
By jsterling on Aug 2, 2021

Scheduled for a re-visit same time next week.
By zhi on Aug 2, 2021

Resources using this document
housingrehab

Permissions
Click the button below to change the permissions of this document.

About
Responsible, Point of Contact, Metadata Author
zhi
No Group

Figure 14. Communication is exchanged between the Community Development Planner and the inspection agency, first with a directive and then followed by a scheduling update.

Platform Evaluation: GeoNode

The following writings about this platform experiment will serve as documentation for lessons-learned when working with an open-source project, suggestions for how open-source projects can incorporate collaborative features, explanations for how specific platforms can accomplish collaborative tasks, and recommendations for users, developers, and administrators of GeoNode projects.

Ease of Development

The GeoNode project at face-value is organized and well-written considering its more organic development pattern, as opposed to the structured, formal QGIS project. The convenience of using GitHub as a project management tool in addition to specifically formatted GeoNode Improvement Proposals (GNIPs), descriptively tagged issues, and general code formulas helps to keep a uniform development path. Even still, this project uncovered that installation requires a moderate level of programming experience, especially when troubleshooting. A project having a significant number of dependencies is an indicator of power and flexibility, but it can become problematic when some of these dependencies are unstable, meaning they are frequently updated. This can cause version conflicts and broken installations if the dependencies are no longer backwards compatible. It can also cause confusion when the installation documentation has not been updated to require the most recent version requirements, which was the case at least once for this project. In general, a high number of unstable dependencies can create problems with frequent programming maintenance, managing updates, decision-making regarding choosing to depreciate or remove a dependency.

Although it is understandably impossible to speculate every possible issue that one could encounter during the installation process, choosing to use a command line process begs explanation for users who are not programmers. For example, parts of the installation process requiring text editing could suggest VIM and explain how to use text editing commands. In addition to continuing to support containerized installation, gathering a list of references to external resources would help to prepare inexperienced users for the requirements necessary to work with GeoNode. Adding general context to the documentation would provide users with a better understanding of code mechanics; for example, why is tomcat the preferred java servlet rather than jetty? What is the purpose of the paver, and what are all the alternative options to running GeoNode? Although having a Gitter chat available for community support is helpful, this setup can be undermined by repetitive problems that are asked and addressed over and over. Over the course of this project, the Gitter activity and user-group emails posed questions regarding installation, requests for general troubleshooting, and inquiries about integrating new features.

There is another project that exists which would replace the need to develop a virtual machine, as was done for this prototype. The OSGeoLive Project (http://live.osgeo.org/en/index.html) version 14.0 has GeoNode 3.1 already installed, as well as a selection of many other open-source GIS applications. This version was released almost two years after the previous major version (2.x) and just over one year after GeoNode 3.0 was initially released. The timing of this project combined with this delay in releases created more challenges for this project than would typically be expected, but the positive outcome of this transition will benefit new users, making it much easier to test a lightweight version of GeoNode in its most evolved form.

Level of Support for Key Features

Support for collaborative functionality in GeoNode varied from well-supported to virtually unsupported. Well-supported features were ones which were functional and available at a basic capacity. Features with medium support had some structure which would allow the potential for further development. Features with little-to-no support were defined by a lack of structure in place to support the function. Overall, the potential for collaborative functionality in GeoNode exists with opportunity for improvement (see *Table 4*).

Table 4. Summary descriptions of collaborative functionality implemented and/or evaluated.

Collaborative Feature	Level of Support Provided	Implementation
Form Submission	Little to none	Requires low-level programming to add a Django form module
Feature Assignment	Medium	Assigning features to another user requires a workaround by creating special, separate layers
Queue Completion	Medium	Similar requirements as feature assignment (above), with the added challenge of updating the original layer
Documentation	Well-supported	In addition to high-level metadata documentation standards, a variety of external document types can be uploaded and linked
Communication	Well-supported	Both public (comments) and private (inbox) communication channels; improved front-end inboxing with the latest version

Form submission is a relatively simple feature which does not have a deployment module in GeoNode's customized Django administration dashboard. However, the existence of django plug-in apps that offer form handling make it relatively simple to add new forms. This could be developed in one of two ways: either add each form to the dashboard programmatically as needed, or add a form which gives the administrator the ability to create a survey form for users to complete. The latter option would be acceptable and straightforward as long as the form remains simple. However, more complicated forms (such as ones which might require spatial plotting) could benefit from the first suggestion.

Feature assignment and queue completion are interrelated functions which would need to be handled with the development of customized python functions. One way to do this might be to create special, separate layers with granted access to only the assigned user. It is worth emphasizing that this handling would not be possible if the basic user-role structure unique to GeoNode did not exist.

As expected, documentation and communication were both well-supported, just as described in the GeoNode documentation. Uploading numerous types of documents is a simple process in both the admin dashboard and the user interface. As an extended benefit, the GeoNode methods for handling metadata were more robust than originally expected, further demonstrating the powerful drive of collaboration within GeoNode. Without carefully documented projects and data, it becomes very difficult to share useful information with other people. Furthermore, GeoNode supports direct communication between users in both a public and private form through comments and inbox messages respectively.

Invested Resources

GeoNode provides the basic, default stack of apps and APIs necessary to run a customized project in a local, development environment. In order to deploy a production-ready (publically accessible) version of GeoNode, the administrator would need an internet connection, a host computer or virtual machine for GeoNode, and a designated website host/server with a custom domain for publication, which can all be hosted on the same machine or distributed across multiple machines. In the case of this prototype, one laptop hosted several iterations of installations with numerous snapshots in various states of development, which helped to keep the project development in a safe, isolated environment which could then be easily shared with others.

Over the course of nearly 2 years since this project was originally proposed, over 1000 hours have been invested into its conception, development, and documentation; This would be the equivalent of about 6-months of full-time work. In that time, GeoNode demonstrated a high level of adaptation, driven by community standards and expectations to accomplish many goals. Although the typical Windows installer is no longer available, GeoNode developers have created new streamlined documentation for installation on Windows using a containerization platform, Docker, which did not exist at the time the project was executed. Another example of change was the original interactive mapping application being changed to use MapStore as the default, which appears to have been a welcome change in the community. The fast-paced development environment helps keep GeoNode in a competitive realm, but can also be overwhelming for GeoNode users who rely on that community for support; without constant maintenance and updates, any customized GeoNode project risks the possibility of becoming obsolete and unsupported because the community resources can only afford to devote their finite resources to the most current version of GeoNode.

One of the major philosophical problems raised in program development is whether an application/platform/software should be able to do *everything* to an extent or rather specialize in just a *few* things. There are obvious trade-offs with choosing to do or offer everything versus choosing to focus on only the best (subjective) functionality. Primarily, the difficulty encountered by current installation instructions and standards, the lagging documentation, and the ongoing knowledge required to continue to maintain and update to the most recent version presents a significant barrier to widespread adoption of GeoNode. By existing as a community project, GeoNode suffers from these pitfalls, but also brings a set of unique features to the world of open-source GIS that has not otherwise been filled by an open-source project.

Recommendations

Although the application domain for this project tends to be uniquely focused on providing the most benefit to the external community, as most governments should ideally operate, the practice of prioritizing service to some particular user-group is not exclusive to government agencies. In the private sector, target customers could just as easily become the focus of inclusive platform development where access to specific data can be carefully granted or restricted and users amongst all parts of the organization can collaborate on common goals in a spatially competent environment.

It is strongly recommended that potential GeoNode developers have some basic skills and knowledge prior to attempting to work with GeoNode. The primary suggestion is to spend some time working with virtual machines in a well-documented virtualization software. Then practice basic use of the command terminal, including executing update commands and using a command-line text editor, which can be learned from resources such as the VIM Cheat Sheet (2020). If not already familiar with the Python programming language, obtain experience both reading and writing Python code. Once this foundation of knowledge is established, attempt working with the GeoNode established in the OSGeoLive virtual machine or try the Docker installation in a new machine. After spending time working with GeoNode in a prototype environment, it is finally recommended that developers conduct research on web security before moving to a production setting.

In order to continue to provide access to a high quality platform, GeoNode developers should make every effort to prioritize documentation, specifically considering how the most inexperienced user would interpret or question any particular instruction or step. One way to improve documentation might be to include plenty of crosslinks to the current GitHub repository and provide reputable, permanent links to external resources; this suggestion compliments the GeoNode-way of valuing documented metadata standards, placing it squarely within the realm of possibility for long-term project success.

Conclusion

Although some hazards of developing with open-source software can make it challenging for non-programmers to risk their current workflows, GeoNode offers methods and community support for prospective developers with attention to detail, patience to deal with troubleshooting, and ability to communicate needs with other GeoNode developers. Even with strong OSGeo support providing a convenient virtual machine image to help users avoid initial “programmer fatigue,” updated documentation will be necessary in order to bring non-programmers up to speed with GeoNode 3.0 in the OSGeoLive image. Although rewriting the internet would be a demanding task, it would be helpful to provide more intuitive literature describing the “best” installation method with specific reasoning to support this ranking. One of the most challenging tasks as a beginner-level user is trying to discern where exactly to start. Choice is the mark of versatility, but the best platforms offer simple out-of-the-box options for quick, instant results. It is difficult to identify a more satisfying accomplishment than running a few lines of code to produce immediate results, which can then be customized intuitively in a streamlined user-interface. It is not hard to imagine GeoNode approaching this vision through the integration of the recommendations provided. In fact, as more personal experience is gained with the GeoNode project, this project will have served as the first of many contributions to the GeoNode project moving forward. Additional resources supplementing this paper can be found on GitHub (https://github.com/aleshreffler/geonode-project). This paper can be accessed and shared through the following link: https://tinyurl.com/GeoNodeEvaluation2021.

Acknowledgements

I would like to recognize and praise my advisor, Jan Oliver Wallgrün, for his priceless guidance, advice, and encouragement throughout the development of this project. His enduring efforts and patience did not go unnoticed. This project would not have been possible without the support of my closest friends, family, and colleagues.

References

Atzl, C., Scholz, J., Vockner, B., Mittlböck, M., & Knoth, L. (2019). Role-tailored map dashboards: A new approach for enhancing the forest-based supply chain. *ISPRS International Journal of Geo-Information, 8*(1), 41. doi:10.3390/ijgi8010041

Balbo, S., Boccardo, P., Dalmasso, S., & Pasquali, P. (2014). A public platform for geospatial data sharing for disaster risk management. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40*(5) 189-195. doi:10.5194/isprsarchives-XL-5-W3-189-2013

Buonanno, S., Zeni, G., Fusco, A., Manunta, M., Marsella, M., Carrara, P., & Lanari, R. (2019). A GeoNode-based platform for an effective exploitation of advanced DInSAR measurements. *Remote Sensing, 11*(18), 2133. doi:10.3390/rs11182133

Cai, G. (2005). Extending distributed GIS to support geo-collaborative crisis management. *Annals of GIS, 11*, 4-14. https://doi.org/10.1080/10824000509480595

Cartwright, W. (2008). Delivering geospatial information with Web 2.0. (pp. 11-30). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved online from https://link-springer-com.ezaccess.libraries.psu.edu/book/10.1007/978-3-540-72029-4#

Corti, P., Bartoli, F., Fabiani, A., Giovando, C., Kralidis, A. T., & Tzotsos, A. (2019). GeoNode: An open source framework to build spatial data infrastructures. *Peerj Preprints.* doi:10.7287/peerj.preprints.27534v1

Dauzon, S., Bendoraitis, A., Ravindran, A., & Safari Books Online (Firm). (2016). *Django: Web development with python* (1st ed.) Packt Publishing. Retrieved from https://learning.oreilly.com/home/

Foerster, T., Stoter, J., & van Oosterom, P. (2012). On-demand base maps on the web generalized according to user profiles. *International Journal of Geographical Information Science, 26*(1), 99-121. doi:10.1080/13658816.2011.574292

Gartner, G. (2009). Web mapping 2.0. (pp. 86-100) Routledge. doi:10.4324/9780203876848-10

GeoNode Development Team. (2021). GeoNode Documentation (3.1). Zenodo. https://doi.org/10.5281/zenodo.5153169

Global Facility for Disaster Reduction and Recovery. (2020). InnovationLab [GeoNode application]. https://www.geonode-gfdrrlab.org/

Global Facility for Disaster Reduction and Recovery. (2017). Open data for resilience initiative & GeoNode: A case study on institutional investments in open source. *Washington, DC: GFDRR*. License: Creative Commons Attribution CC BY 3.0. Retrieved from https://opendri.org/wp-content/uploads/2017/03/GeoNode_Report_Final-April-4-light-map.pdf

Haklay, M., Singleton, A., & Parker, C. (2008). Web mapping 2.0: The neogeography of the GeoWeb. *Geography Compass, 2*(6), 2011-2039. doi:10.1111/j.1749-8198.2008.00167.x

Herbert, A., & Safari Books Online (Firm). (2019). *Understanding django* (1st ed.) Packt Publishing. Retrieved from https://learning.oreilly.com/home/

Lopatin, B., & SpringerLink (Online service). (2020). *Django standalone apps: Learn to develop reusable django libraries* (1st 2020. ed.). Berkeley, CA: Apress. Retrieved from https://link.springer.com/

Martin, S., Reynard, E., Pellitero Ondicol, R., & Ghiraldi, L. (2014). Multi-scale web mapping for geoheritage visualisation and promotion. *Geoheritage, 6*(2), 141-148. doi:10.1007/s12371-014-0102-3

Maurya, S., Ohri, A., & Mishra, S. (2015). Open source GIS: A review. Retrieved from https://www.researchgate.net/publication/282858368_Open_Source_GIS_A_Review

Pulsani, B. R. (2015). Implementation of open-source web mapping technologies to support monitoring of governmental schemes. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-2/W2*(2), 147-153. doi:10.5194/isprsannals-II-2-W2-147-2015

Rubio, D., & Safari Books Online (Firm). (2017). *Beginning django: Web application development and deployment with python* (1st ed.) Apress. Retrieved from https://learning.oreilly.com/home/

Shaparev, N., & Yakubailik, O. (2016). Usage of web mapping systems and services for information support of regional management. *MATEC Web of Conferences*, *79,* 01081. doi:10.1051/matecconf/20167901081

Steiniger, S., De La Fuente, H., Fuentes, C., Barton, J., & Muñoz, J. (2017). Building a geographic data repository for urban research with free software. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 42*(4) 147-153. doi:10.5194/isprs-archives-XLII-4-W2-147-2017

Tsou, M., & Curran, J. M. (2008). User-centered design approaches for web mapping applications: A case study with USGS hydrological data in the United States. In *International perspectives on maps and the internet* (pp. 301-321). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72029-4_20

Veenendaal, B., Brovelli, M. A., & Li, S. (2017). Review of web mapping: Eras, trends and directions. *ISPRS International Journal of Geo-Information, 6*(10), 317. doi:10.3390/ijgi6100317

Veenendaal, B., & Dhliwayo, M. (2013). Web mapping for promoting interaction and collaboration in community land planning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-4/W2*, 107-113. doi:10.5194/isprsarchives-XL-4-W2-107-2013

Vim Cheat Sheet. (2020). *Vim Cheat Sheet*. Retrieved from https://vim.rtorr.com/.