

List of Components

- Elastic Cloud Compute (EC2)
 - EC2 Autoscaling
 - EC2 Network Load Balancer
- Relational Database Service (RDS)
- Route 53 (Potentially)
- AWS Certification Manager (ACM) (Potentially)
- Virtual Private Cloud (VPC)
- Amazon Simple Storage Service (S3)
- Elastic File System (EFS)
- Amazon CloudWatch (Potentially)

Backend Architecture Description

The backend architecture will use Django and be composed of various controllers.

The Authentication Controller will handle login requests by redirecting the user to their specified login choice, such as Google. Once redirected, the user will be prompted to log into their account. Upon successful login, they will be redirected back to the data portal, which will process the response, grab the URL for the user's homepage and send that view to the user.

The File Controller will be responsible for handling the users' file requests which will query the database, set up a view, and send it to the user. Furthermore, it will also handle save requests that will either save a file if not present or update one if the file already exists. Another feature that the file controller will handle will be users' scripts. When a user creates a script, the file controller will query the database, package it up in an Amazon S3 bucket and pass it along to Amazon's Lambda serverless computing service. This will provide users with resources to run their scripts using high-powered machines.

The project controller will handle all requests pertaining to a user's protected system files, such as projects. It will take a user's request, process it, query the database for the selected project, and then provide resources to the user.

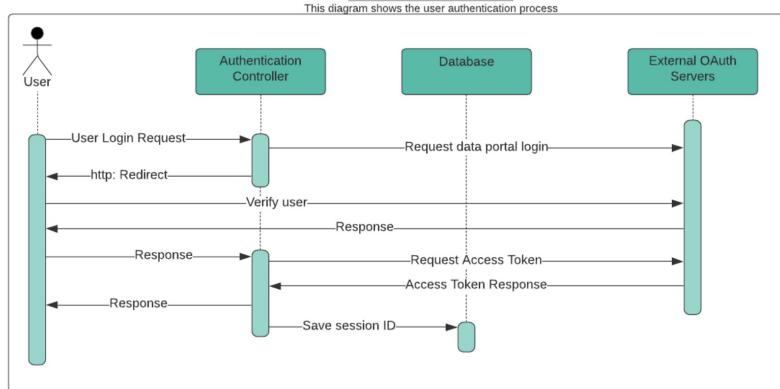
List of Components

- File systems
- File controller
- Database
- Project Controller
- Web Server
- Authentication Controller

High-level Sequence Diagrams

Authentication Controller → External OAuth → Verify User

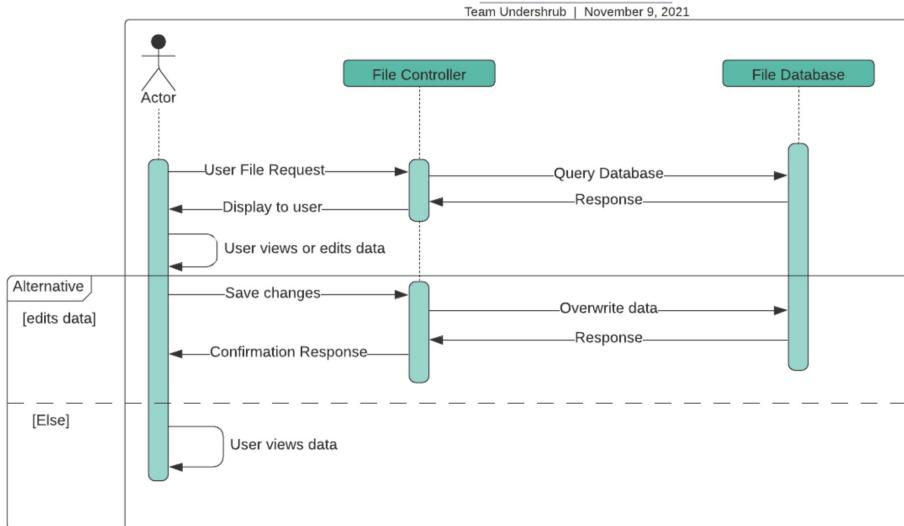
Figure 5: Authentication Sequence Diagram



caption: This diagram shows the user authentication process

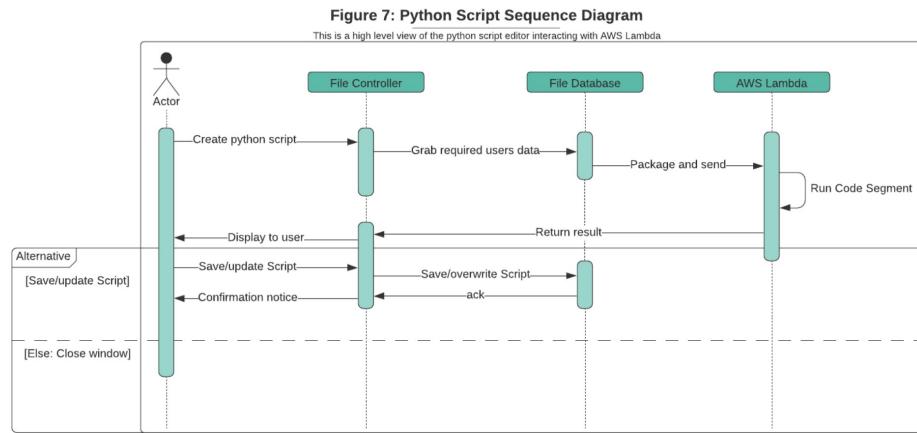
File Controller → Query Database → Provide file to user → save data to database

Figure 6: File Sequence Diagram



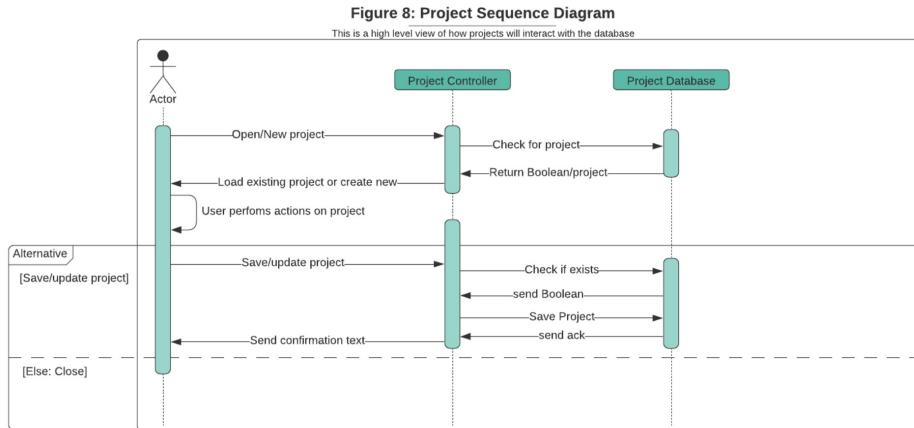
caption: This diagram shows how a file is opened, edited and saved.

File Controller → Query Database → Query AWS Lambda → Provide resources to user



caption: This figure is a high level view of the python script editor with AWS Lambda

Project Controller → Query Database → provide resources to user



caption: This figure is a high level view showing how files will interact with the database.

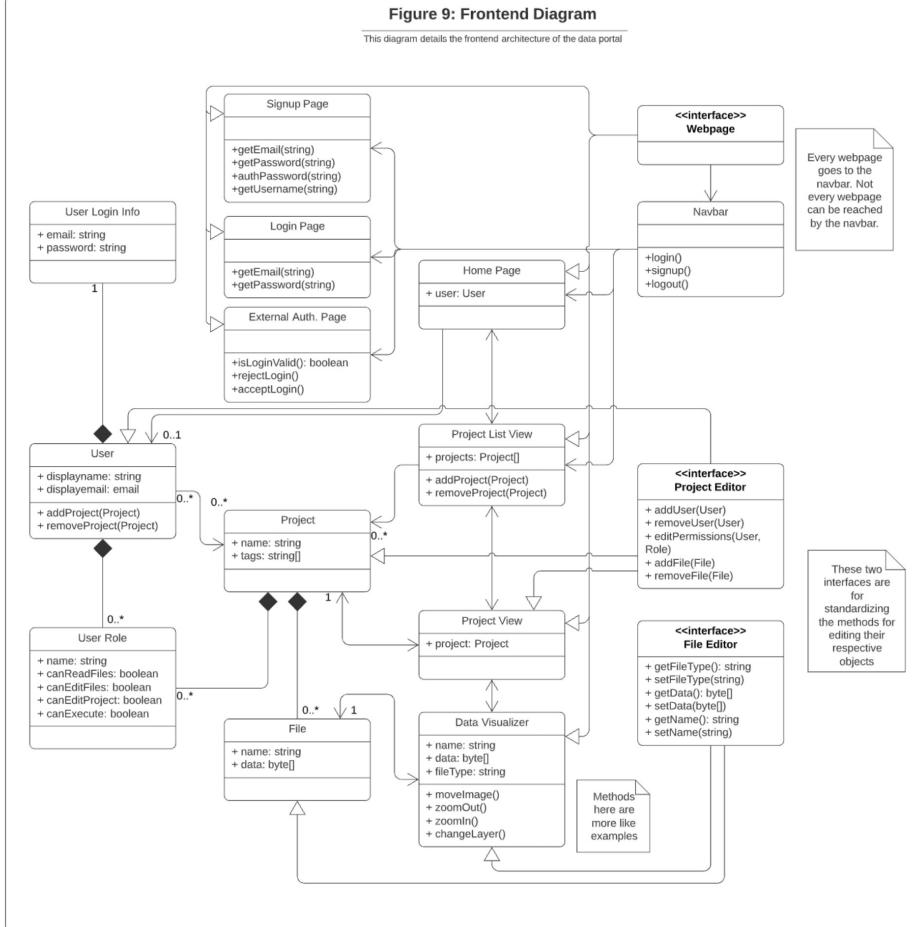
Frontend Architecture Description

The frontend architecture diagram gives a description of the system's web interface and design, detailing how users interact with the system's key components. The first component of the web interface is the homepage which contains software information, account accessibility, introductions, links to the navbar and the project list. Once you navigate off the homepage and either have a guest or user account you can access the project view which shows all related project media the user has access to, alongside allowing the user to edit or select a project if they have the right credentials. The project is viewed within the project view and allows the display of any file the user wishes to access and the directory of the structure. Within the project view is the data visualizer which can be opened to view particular information on the project such as the layering, images, or even specific documents uploaded to the project.

The frontend will also work to create "user permission" objects to be associated with specific actions in a "user project" pair structure which can associate certain permissions which any user with that object gains access to. Within the diagram the "user login info" is only designed to exist for the signup/login process pages and won't be used outside of them, while the "username" and "user" objects will exist along projects and the homepage to identify the user. There will be a simple (1 - 0) boolean which can represent a signal if the user is logged in, once the project notices a user is logged in it can then retrieve the specific information associated with the user. Notably; this system will allow multiple projects to exist alongside a single user and multiple users to exist for a single project, being viewable in the "project list view" to see which clients have access to what information.

Figure 9: Frontend Diagram

This diagram details the frontend architecture of the data portal



caption: This details how the front end will interact with the various elements of the user interface and details the data and data types of each object.

List of Components.

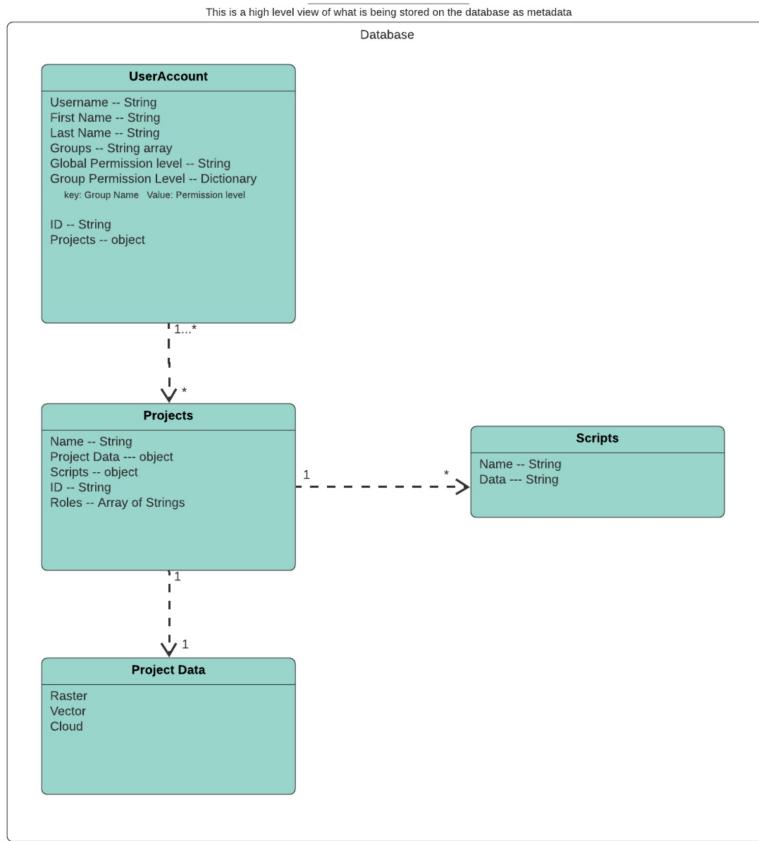
- <<interface>> Webpage
- Home Pages: Webpage
- Project List View: Webpage
- Project View: Webpage
- Data Visualizer: Webpage
- Navbar: Webpage
- Signup Page: Webpage
- Login Page: Webpage
- External Authentication Page: Webpage
- Project
- File
- User

- User Permissions
- User Login Info

B.3 Persistent Data Design

This section will detail a high level view of the database's data architecture. A MySQL instance of AWS RDS is the database that will be primarily used to store information about the customer and their data. The database description section (3.1) will be an entity relationship diagram that will display what is being stored, the data type and what it represents. The following section (3.2) will be the File description section that will detail the file structure and its contents to include the name, data types, and representation descriptions.

Figure 10: Database Diagram



Caption: A high level view of how a user's account information and data will be stored. It also shows the data being stored at each level of an account and project.

B.3.1 Database Descriptions

UserAccount will store a user's information which includes private and public data, the description of this data is detailed below.

- **Username:** This will be the user's display name.
- **First Name:** This is the user's first name.
- **Last Name:** This is the user's last name.
- **Group Name:** This is the current group(s) the user is a part of.
- **Global Permission Level:** The current permission level of the user that defines access controls
- **Group Permission Levels:** The current permission level of the user within each group the user is a member of.
- **Projects:** A current list of projects the user owns or is a member of.

Projects will store a user's current projects to include the name, its data (explained below) and any scripts attached to them.

Project Data will contain various raster, vector, and point cloud data that belong to a user. The layers are as follows:

- **Raster Imagery:**
 - The user could work with various raster information such as imagery, terrain datasets, etc.
- **Vector Data:**
 - Examples of vector data could include GIS data, parcels, demographics.
- **Point Cloud:**
 - Various examples could be lidar and 3D data collections.

Scripts will be stored by their name as a string and the data will be a string that will be the source code.

B.3.2 File Descriptions

This section will break down the file structure in terms of fields. Each field will be assigned a name, it's data type, size, and a small description of what it will represent.

UserAccount

- An object for storing information about a user within a database.
- Username: string
- FirstName: string
- LastName: string
- Groups: string[]
- GlobalPermissionLevel: string

- GroupPermissionLevels: Dictionary<Key: string, Value: string>
- ID: string
- Projects: string[]

Projects

- An object for storing information about a project within a database.
- Name: string
- Data: ProjectData
- Scripts: Script[]

Script

- An object for storing information about a script within a database.
- Name: string
- Data: string //Parsed as Python code.

ProjectData

An object for storing information about project data within a database.

- Raster Imagery: .TIFF
- Vector Data: Vertices and paths
- Point Cloud: Lidar (x, y, z)

Authentication Controller.

- Controller for authentication of users.
- UserLoginRequest(UserLoginInformation): Nullable<httpRedirect>

File Controller.

- Controller for accessing/modifying files by users.
- UserGetFileRequest(File): databaseResponse
- UserSaveFileRequest(File): databaseResponse

File Database.

- Object representing a database for storing files.
- DatabaseQuery(File): databaseResponse
- DatabaseWrite(File): databaseResponse

Project Controller.

- Controller for managing projects by users.
- OpenProject(Project): databaseResponse
- SaveProject(Project): boolean

Project Database

- Object representing a database for storing projects.
- DoesProjectExist(Project): boolean

- SaveProject(Project): boolean

<<interface>> Webpage.

- Interface to represent any webpage.

Navbar.

- A section on every webpage to help navigate to other pages.

Home Page: Webpage.

- The default page for the website.
- NavigatePages()
- ViewUser()
- Login()
- Signup()
- Logout()

Signup Page: Webpage.

- A page to create a new account.
- GetEmail(string)
- GetPassword(string)
- AuthPassword(string)
- GetUsername(string)

Login Page: Webpage.

- A page to login to an existing account.
- GetEmail(string)
- GetPassword(string)

External Authentication Page: Webpage.

- A web page to login or signup via an external authentication service.
- IsLoginValid(UserLoginInfo): boolean
- RejectLogin()
- AcceptLogin()

Project List View: Webpage.

- A web page to view all projects assigned to the user.
- ViewProject(Project)
- LeaveProject(Project)
- AddProject(Project)

Project View: Webpage.

- A web page to view a single project and its files.
- DownloadFile(File)
- UploadFile(File)
- EditFile(File, byte[])
- RemoveFile(File)
- CopyFile(File)

Data Visualizer: Webpage.

- A web page to visualize a single file's data.
- RasterData: raster_data

User Login Information

- An object to hold the information required to log in a user.
- Email: string
- Password: string

User.

- An object to represent a user.
- DisplayName: string
- DisplayEmail: string

User Permissions.

- An object to represent allowed permissions for a user.
- CanEditFiles: boolean
- CanEditProject: boolean

Project.

- An object to represent a project.
- Name: string
- Tags: string[]

File.

- An object to represent a file.
- Name: string
- data: byte[]
- getFiletype(): string

B.4 Requirements Matrix

See [the external requirements matrix](#). The reason it is external is because the formatting looks egregious if done anyway

Subappendix B.A - Agreement Between Customer and Contractor

What is being agreed to when this document is signed.

Upon signing this document, the signer agrees to the terms and conditions presented in Appendix A and Appendix B, with any future amendments requiring an updated signature. The signer also agrees to provide assistance with the creation and any tasks related to the Geospatial Data Portal as agreed by the signer and either Team Undershrub or Blue Marble Geographic®. Additionally, the signer agrees to do everything needed to ensure that this agreement is upheld in good faith, alongside agreeing to treat any designed or developed material as "work made for hire" for Blue Marble Geographic®. Except as otherwise stated within this agreement, the signer will have full control over working time, methods, and decisions related to the project until either the Blue Marble Geographic® or Team Undershrub deem to end the arrangement. However, the signer will be responsive to the reasonable needs and concerns from either aforementioned group throughout the duration of the arrangement.

Procedures to be used for future changes to this document.

When a member of Team Undershrub or an employee of Blue Marble Geographic® wishes to commit any changes to this document, they must first address them in a comment underneath their signature below, then must alert either the SDD document manager or the Geospatial Data Portal project lead. The alerted individual will notify all signing parties. Subsequently, all signing parties will discuss via email or a meeting the proposed changes to the document. Implementation of proposed changes requires support from the Geospatial Data Portal project lead along with a majority of the development team. These procedures shall be followed for any future additions, deletions or modifications of material within this document.

Name: Victor Minor	Date: 11/19/2021
Signature:	

Subappendix B.B – Team Review Sign-off

Within this section of the SDD document you will find the signatures, names and date of the documental approvals for each acting member of "Team Undershrub". These signatures acknowledge and authenticate the approval and review of this document. This authentication includes the overall content, formatting, identification of contributed material and development directions presented within. These signatures must be updated alongside the date and comments when there is a subsational update to the document, which in this case includes any formatting, editing, grammar and change of materials within.

Name: Anthony Jackson	Date: 11 - 7 - 21
Signature:	
	
Comments:	

Name: David Sincyr III	Date: 11 - 8 - 21
Signature:	
	
Comments:	

Name: Devin Carter	Date: 11 - 8 - 21
--------------------	-------------------

Signature:

Devin Carter

Comments:

Name: Stephen Kaplan	Date: 11 - 8 - 21
----------------------	-------------------

Signature:

Stephen Kaplan

Comments:

Name: Grant Shotwell	Date: 11 - 8 - 21
----------------------	-------------------

Signature:

Grant Shotwell

Comments:

Subappendix B.C – Document Contributions

Grant Shotwell	<ul style="list-style-type: none"> - Formatting - Frontend Architecture - File Descriptions - Total Contribution: 20%
Stephen Kaplan	<ul style="list-style-type: none"> - Assisted in the design of all systems - Created the Trello board - Formatted the entire document - Created cloud architecture diagrams - Wrote cloud architecture descriptions - Total Contribution: 20%
Devin Carter	<ul style="list-style-type: none"> - Created the requirements matrix - Total Contribution: 20%
David Sincyr III	<ul style="list-style-type: none"> - Migrated over portions of the SRS - Wrote the introduction section - Setup the title page - Designed the sequence diagrams - Wrote the persistent data introduction section - Created the Database entity relationship diagram and wrote their descriptions - Wrote the backend description - Total Contribution: 20%
Anthony Jackson	<ul style="list-style-type: none"> - Wrote appendix B's original formatting and SDD authentication. - Assisted with sequence diagram design. - wrote the architectural design description. - assisted with the front-end diagram by filling in functions. - rewrote the frontend architecture description. - formatted 3.1 & 3.2 - Total Contribution: 20%

Appendix C - User Interface Design Document

Table of Contents	Page
C.1 Introduction	121
Purpose of This Document	121
References	121
C.2 User Interface Standards	122
2.1 Layouts	122
Standard Layouts	122
Signup, Login, and other Authentication Pages	123
Dashboard Page	124
User Profile and Settings Pages	124
Geospatial Data Projects	125
Script Editor	126
Error Pages and Notifications	126
2.2 Common Components	126
Form Elements	126
Menus	126
Prompts and Dialogues	127
Lists of Documents, Projects, and other Objects	127
Footer	127
2.3 Branding	128
Colors	128
Typography	128
Logos and Slogans	128
C.3 User Interface Walkthrough	129
C.4. Data Validation	133
C.5 Report Formats	135
Subappendix C.A – Agreement Between Customer and Contractor	135
Subappendix C.B – Team Review Sign-off	136
Subappendix C.C – Document Contributions	138

Document History

Name	Date	Reason for Changes	Version
David Sincyr	11/8/2021	Initial Creation	1.0
All Team Members	11/27/2021	Added in all relevant information	1.1
All Team Members	11/29/2021	Finalized the document	1.2

C.1. Introduction

Blue Marble Geographics® is a software development company that specializes in geographic information systems and geodetic desktop software. Their software is aimed towards all types of users who need a better tool to manipulate and visualize their geospatial data. In order to better serve their customers, a cloud based structure is needed. The dynamic web portal will allow its users to upload their geospatial data to an online database and provide them the ability to share, manipulate, and visualize their data with others. Additionally, users will be able to automate the visualization process by using Python scripts through a built-in Python script editor. This is a capstone project for Blue Marble Geographics®, in partial fulfillment of the Computer Science BS degree for the University of Maine.

C.1.1 Purpose of This Document

The purpose of this document is to specify the user interface design elements of the Geospatial Data Portal project along with guidelines on their appropriate usage. This specification details various screen layouts, common components such as menus and texts, and branding requirements. A walkthrough of the user interface with specifications and limitations on user-supplied data fields is also provided. The specifications detailed in this document conform to and extend those laid out in the Blue Marble Geographics® Brand Guide. The intended readership for this document are the stakeholders and the design team to analyze the user interface of the application prior to deployment.

C.1.2 References

1. LiDAR data information:
https://www.bluemarblegeo.com/knowledgebase/global-mapper-20/LiDAR_Support_in_Global_Mapper.htm
2. Blue Marble Geographics Brand & Style Guide:
<https://docs.google.com/document/d/1IfIrl8nXOnt-TjAx1LiOQN-m1cjYbrng-zUS5KTnHX38/edit>
3. UI Prototype tool:
<https://framer.com>
4. Google Colors:
<https://usbrandcolors.com/google-colors/>
5. Global Mapper Pro Desktop Application

C.2. User Interface Standards

This section will detail the design standards that will be adhered to for consistency in the user interface, such as the layouts, details of what each page will contain, common components, colors, typography, and slogans. Illustrations will be used to show the more important aspects of the application.

C.2.1 Layouts

C.2.1.1 Standard Layouts

This application will utilize three primary layouts (Single Column, Multi-Column, and Modal). Additional layouts will be used as necessary and where reasonable on specific pages, such as code editor pages. The following paragraphs provide guidelines for appropriate use of the primary layouts.

Pages of the web application may consist of multiple layouts, and layouts may comprise other layouts. For example, a page may use a Single Column layout where the main Content Container consists of a Multi-Column layout. To minimize the complexity of the user interface, fewer layout compositions are preferred unless the additional compositions significantly increase clarity or provide some other traceable benefit to users.

Single Column

The single column layout consists of a series of containers stacked vertically. This layout is the wrapper for most but not all pages of the application. In a page context, the Single Column layout begins with the Page Header and ends with the Page Footer. The Navigation Menu is typically placed directly below the Page Header, followed by the main Content Container for the page, but forgoing the Navigation Menu is permissible on pages that require more specialized navigational controls.

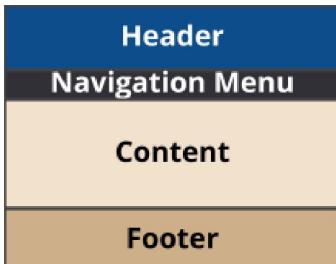


Figure 1: The standard form of the Single Column layout. This is representative of most pages of the application with variations occurring primarily in the Content Container.

Multi-Column

The multi-column layout consists of a series of containers stacked horizontally. This layout is generally (but not necessarily) a subcomponent of other layouts. Columns with widths of simple percentage values, such as 30%, are preferred over arbitrarily-sized ones.

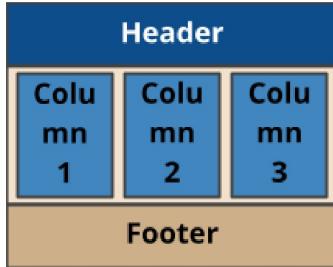


Figure 2: A variation of the Single Column layout which forgoes the Navigation Menu and includes a Multi-Column layout as a subcomponent.

Modal

The modal layout consists of a horizontally centered container overlaying a background, a page, or other elements. The container may be any width, but should generally be no more than 90% nor less than 20% of the parent container's width. Appropriate uses for modal views include displaying notifications, presenting prompts and forms, and focusing attention on a feature or component.

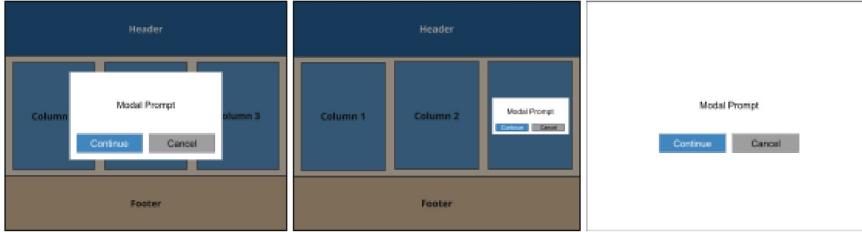


Figure 3: Examples of modal layouts (from left to right) overlaying a top-level layout, overlaying a subcomponent layout, and used as the top-level layout.

C.2.1.2 Signup, Login, and other Authentication Pages

The signup and login pages of this application will use a modal layout with horizontally centered form that includes fields for username and password text input, buttons for "Log In", "Log In With Google", "Forgot Username/Password", and "Create An Account". Other authentication pages, such as the account recovery page, will use a similar format.

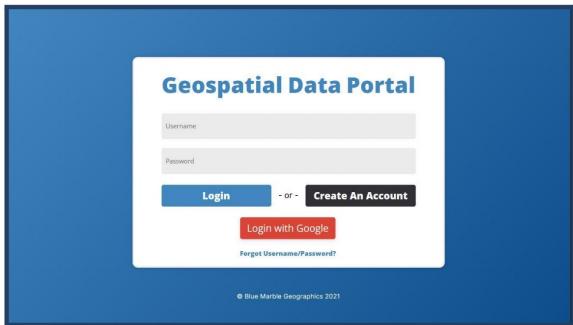


Figure 4: An example of the authentication and signup page with text fields for username and password, alongside these buttons for “Login”, “Create an Account”, “Login with Google” and an extra link for users who lose their credentials.

C.2.1.3 Dashboard Page

The dashboard page will use the standard Single Column layout and will provide users access to all other areas of the application.



Figure 5: This is an example of the dashboard page and the buttons for “Home”, “Profile”, “Settings”, “Logout” and the two lower buttons which link the list of projects and built in script editor respectively.

C.2.1.4 User Profile and Settings Pages

The user profile page will use a Multi-Column layout wrapped by the standard Single Column layout. Users will be able to adjust their information by clicking “Edit” within the profile view. The settings page, which will utilize a similar layout, contains display and permissions settings.



Figure 6: This is an example of a User Profile page with the user's Icon, Username, First Name, Last Name, Email and current groups being presented for viewing purposes.

C.2.1.5 Geospatial Data Projects

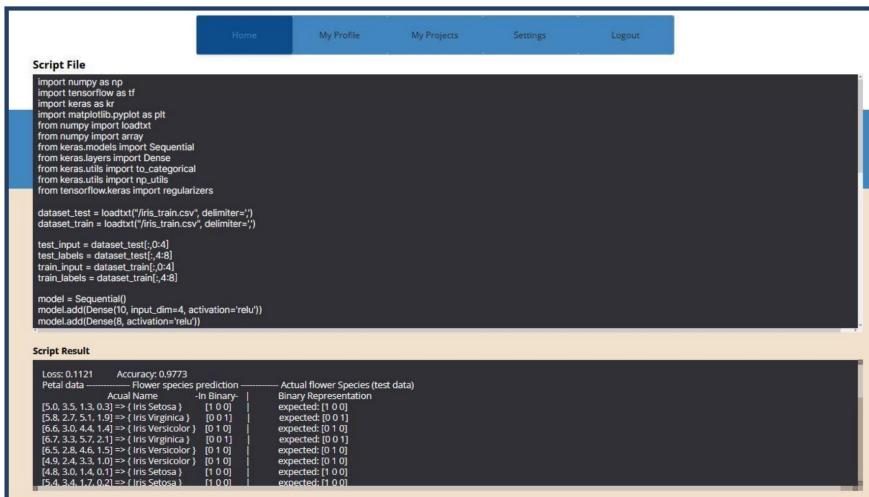
The project view will utilize a Multi-Column layout wrapped by the standard Single Column layout. The first column shows information about the currently selected project, while the second, wider column provides a list of the user's projects for them to select from.

A screenshot of a project selection screen. On the left, there is a sidebar titled "Project Info" containing a list of projects: Project A, Project B, Project C, and Project D. On the right, there is a larger column titled "Project Info" containing a list of projects: Project C, Project D, Project E, Project F, Project G, and Project H.

Figure 7: An example of the users project data selection screen. It is split up into two different images here but the project's window is scrollable.

C.2.1.6 Script Editor

Custom layout (multi-column w/ code editor).



The screenshot shows a top navigation bar with tabs: Home, My Profile, My Projects, Settings, and Logout. Below this is a 'Script File' section containing Python code for a neural network model to predict Iris flower species. The code imports numpy, tensorflow, keras, and matplotlib, loads the Iris dataset, and defines a Sequential model with two Dense layers. The 'Script Result' section below shows the output of the script, including the loss (0.1121), accuracy (0.9773), and a table comparing actual flower species with their binary representations against predicted ones. The table has columns for Actual Name, Binary Representation, and Predicted Name.

Actual Name	Binary Representation	Predicted Name
[5.0, 3.5, 1.3, 0.2] => {Iris-setosa}	[0 0 0]	expected: [0 0 0]
[5.8, 2.7, 5.1, 1.9] => {Iris-Virginica}	[0 1 0]	expected: [0 1 0]
[6.6, 3.0, 4.4, 1.4] => {Iris-Versicolor}	[0 1 0]	expected: [0 1 0]
[6.7, 3.3, 5.7, 2.1] => {Iris-Virginica}	[0 0 1]	expected: [0 0 1]
[6.5, 2.8, 4.6, 1.5] => {Iris-Versicolor}	[0 1 0]	expected: [0 1 0]
[4.9, 3.0, 4.5, 1.4] => {Iris-Versicolor}	[0 1 0]	expected: [0 1 0]
[4.8, 3.0, 4.4, 1.4] => {Iris-Virginica}	[1 0 0]	expected: [1 0 0]
[5.4, 3.4, 1.7, 0.2] => {Iris-setosa}	[1 0 0]	expected: [1 0 0]

Figure 8: This is an example of how the script editor will be laid out for the user to access, showing two boxes with actual script occurrences and then the result from said script.

C.2.1.7 Error Pages and Notifications

Error Pages such as 403 and 404 pages, shall utilize the modal layout. UI notifications of any kind, including error notifications will also be using the modal layout as its base design.

C.2.2 Common Components

C.2.2.1 Form Elements

Form Elements, best described as a container for input elements, are used for login input fields (text boxes for username and password). Form Elements are also used for the functionality of sharing permissions with individuals, groups, or to the overall public. The type of permission (read/write) is itself a Form Element, although a radio button rather than text field.

C.2.2.2 Menus

Method for moving to different areas in the Data Portal. Primary menu component is the navigation bar and its associated dropdown menus. Home, My Profile, Settings, and Logout are examples of navigation bar buttons.

C.2.2.3 Prompts and Dialogues

File Upload

Unique prompt upon selecting to upload a file. The prompt takes an input in the form of a file from the user.

Notices

Assortment of dialogues that appear in reaction to user actions. Commonly these are in response to potential errors. Example, failed login. Another example, failed file upload. These special errors are the standard use case for notice prompts in the Data Portal. Certain success notices may also appear as dialogue to confirm certain results to the other. Note that non prompt/dialogue notices do exist.

Permissions

Whenever a user attempts to add or adjust permissions, for a project, this creates a prompt. User input is taken in order to adjust/add permissions.

C.2.2.4 Lists of Documents, Projects, and other Objects

Content components for the Data Portal. These components have a range of accessibility (public, shared, private), alongside differing permission levels (i.e. if an individual was responsible for the item's creation). These components are held within "My Projects" and similar Data Portal tabs.

C.2.2.5 Footer

The footer will be empty except for a small image of the "Blue Marble Geographic" logo for identification purposes and copyright.

C.2.2.6 Buttons

Buttons are used in a variety of places. One such place is the navigation bar, while another is login. Buttons take two primary forms, static and responsive. The former of the two represent one click buttons that provide no visual feedback and instead purely perform their action. "Continue with Google" is an example of the former. The latter responds visually to users by changing from the default Blue Marble blue to a darker shade. The nav bar offers an example of this type. Buttons are used in many places to confirm inputs or in use of portal navigation.

C.2.2.7 Dropdown

Dropdowns are used primarily for the navigation bar. The dropdown menu allows for condensing standard menu features in order to simplify the visual noise users must

handle. The visual noise, in too great a quantity, would severely handicap the ability to find the desired button (or speaking for example, the desired navigation path).

C.2.3 Branding

The subsequent sections provide a brief overview of the branding guidelines and requirements that apply to this project. Permissible deviations from these branding standards are noted as necessary. For more information, refer to the Blue Marble Geographics Brand Guide.

C.2.3.1 Colors

Blue Marble Geographics uses a standard color palette consisting of five colors: Light Blue, Blue Marble, Charcoal, Earth, and Sand. In the Geospatial Data Portal project, the colors are purposed as follows:

- *Charcoal as the background.*
- *Blue Marble as the secondary background.*
- *Light Blue for primary buttons and actions.*
- *Sand for secondary buttons and actions.*
- *Earth for tertiary buttons and actions.*

Figure 9 (on right): The Blue Marble Geographics color palette, for reference.

C.2.3.2 Typography

Open Sans is the standard font for this project and shall be used for all non-code text, while *Open Sans Bold* shall be reserved for H1, H2, and H3 HTML tags. Any text representing code, code segments, or code comments shall use *Courier New*.

C.2.3.3 Logos and Slogans



Figure 10 (above): The Blue Marble Geographics current logo.

There are no current plans to include any slogans for the project, but there will be a Logo drafted in the later stages which will bear similarities to the logo used by "*Blue Marble Geographics*".



C.3. User Interface Walkthrough

This section will go over how the user will see the website. First a diagram of the navigation, then screenshots of mockup images made in Framer alongside a description of said images.

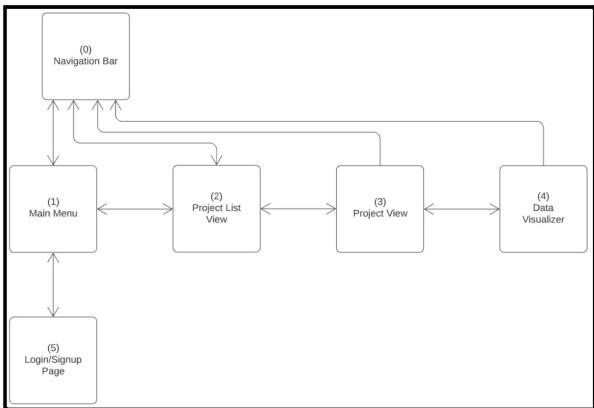


figure 11: This details how the user interfaces will flow between each other and how they are linked together

This diagram represents the navigation between screens on the website. Screens 1 through 4 are the primary screens the user will interact with. These screens will allow the user to interact with the data structure that screen is scoped to. Screen 0, the navigation bar, will appear on every screen to provide easy access to other screens. Screen 5, the login/signup page, will allow the user to login to an existing account or create a new account using an external authentication service.



Figure 12: This is a general overview of the main menu that has a scrollable area as shown on the left. The navigation bar sticks to the top of the browser window as users scroll down the page.

The Main Menu's primary purpose is to be a landing page for all traffic and user activity, within this menu is a "Profile Icon" with the "Username" viewable alongside so that users can know which account they are accessing at a glance; underneath the username is a "logout" button (link) which will allow them to sign out of their account. Alongside this is the "Projects" button which allows a user to view which projects they are currently associated with.

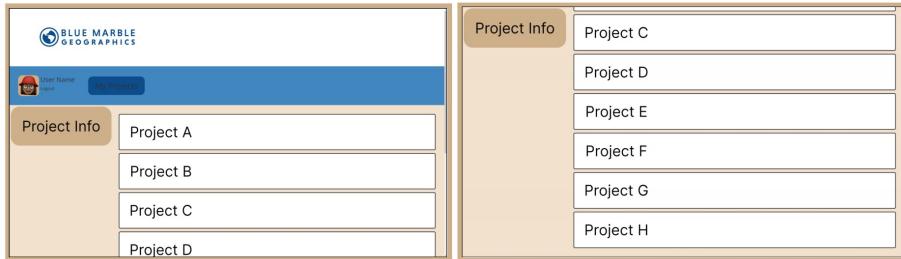


Figure 13: This shows the scrollable project list for an account.

The Project List View's primary purpose is to show the user which projects they are assigned to, in addition to providing them access to the project view screen for those projects. This is easily done through multiple boxes which show the name of the project, as an example the user could press on "Project A" and be taken straight into the project view for said project.

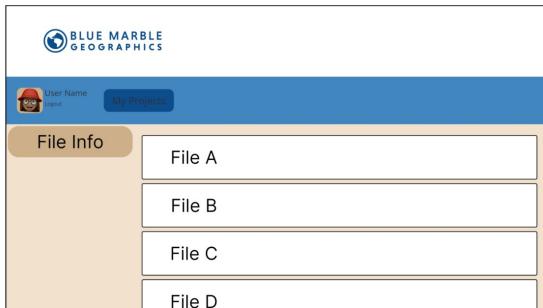


Figure 14: This shows the user interface for a users file and it will also be a scrollable window (not depicted)

The Project View's primary purpose is to show the user all files within the project, providing them access to the data visualizer screen for those files. This likewise uses a "box" method to show a list of files within a project which can then be clicked on to either view or edit depending on what the user needs to do.

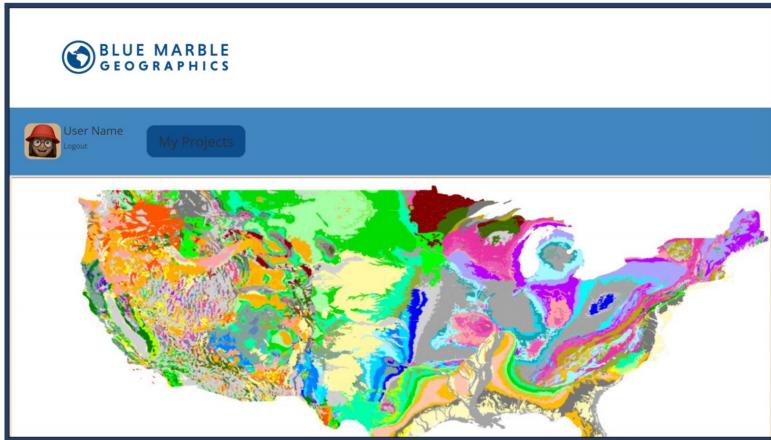


Figure 15: This shows the data visualizer user interface window

The Data Visualizer's primary purpose is to visualize whatever data is contained within a file to the user, the user will be able to modify certain elements of the file from within this screen. This will easily be done through a interactive map which allows the user to toggle between various uploaded layers as to get unique information about various thing such as topography.



Figure 16: This depicts the navigation bar that will follow the user through most user interfaces.

The Navigation Bar's primary purpose is to provide easy navigation to other screens. Not all screens can be accessed from the Navigation Bar. For example, the data visualizer cannot be reached from the navigation bar as it requires a file to visualize. The Main Menu, Project List View, and Login/Signup Page can be accessed from the navigation bar.

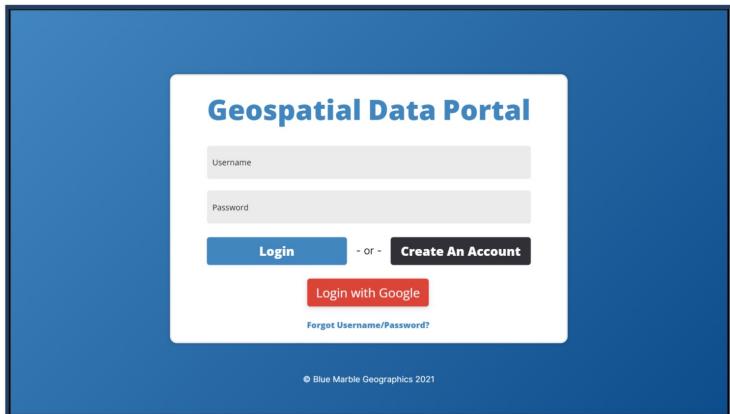


Figure 17: This shows the user login page that will be shown when initially going to the web portal

The Login/Signup Page will provide the external authentication services to the user which includes a “Google Login”, “Create An Account”, “Manual Login” and a small link for beginning the password/username troubleshooting process.

C.4. Data Validation

This section is a description of various data information that the project will need to use, modify or access as to function in its completion. This also includes who is modifying the information, how it's being modified and the constraints to its entry for its usage.

Description of Data	Data Type	Constraints	Handled or Authentication	Unique Identifier
Email	string	Valid email (includes @, ends with a domain), not empty	<i>This information will be handled by Google's authentication services.</i>	auth_email
Username	string	Includes only letters and numbers, longer than 3 characters	<i>This information will be handled by Google's authentication services.</i>	auth_username
Password	string	Longer than 8 characters	<i>This information will be handled by Google's authentication services.</i>	auth_password
FirstName	string	Not empty	<i>This information will be handled by Google's authentication services.</i>	auth_firstname
LastName	string	Not empty	<i>This information will be handled by Google's authentication services.</i>	auth_lastname
Python Script	.py file type	> 0 bytes, < 1 GB, includes python file header	This information will be stored on an Amazon S3 bucket and paired with a unique token used for authentication purposes. This information may then be accessed by users, the Project	upload_pscript

			Management subsystem, and AWS Lambda following successful authentication and permission verification.	
Raster Imagery	.tif file type .png file type	Includes tif or png file header	This information will be accessed from the user's system before being presented for viewing or augmentation in the user's database.	upload_raster
Vector Information (Vertices, Paths)	.svg file type	Includes svg file header	This information will be accessed from the user's system before being presented for viewing or augmentation in the user's database.	upload_vector
Point Cloud Information	.PCD file type .LAS file type .LAZ file type	Includes PCD, LAS, or LAZ file header	This information will be accessed from the user's system before being presented for viewing or augmentation in the user's database.	upload_pointcloud
Project Name	string	Includes only letters	This information will be given to the system for labelling purposes and viewing from users.	project_name
Project Description	string	Includes only letters and numbers	This information will be given to the system for labelling purposes and viewing from users.	project_description

C.5. Report Formats

The data access portal will not generate any hard copy reports.

Subappendix C.A – Agreement Between Customer and Contractor

What is being agreed to when this document is signed.

Upon signing this document, the signer agrees to the terms and conditions presented in Appendix A and Appendix B, with any future amendments requiring an updated signature. The signer also agrees to provide assistance with the creation and any tasks related to the Geospatial Data Portal as agreed by the signer and either Team Undershrub or Blue Marble Geographic®. Additionally, the signer agrees to do everything needed to ensure that this agreement is upheld in good faith, alongside agreeing to treat any designed or developed material as "work made for hire" for Blue Marble Geographic®. Except as otherwise stated within this agreement, the signer will have full control over working time, methods, and decisions related to the project until either the Blue Marble Geographic® or Team Undershrub deem to end the arrangement. However, the signer will be responsive to the reasonable needs and concerns from either aforementioned group throughout the duration of the arrangement.

Procedures to be used for future changes to this document.

When a member of Team Undershrub or an employee of Blue Marble Geographic® wishes to commit any changes to this document, they must first address them in a comment underneath their signature below, then must alert either the UIDD document manager or the Geospatial Data Portal project lead. The alerted individual will notify all signing parties. Subsequently, all signing parties will discuss via email or a meeting the proposed changes to the document. Implementation of proposed changes requires support from the Geospatial Data Portal project lead along with a majority of the development team. These procedures shall be followed for any future additions, deletions or modifications of material within this document.

Name: Victor Minor	Date:
Signature:	

Subappendix C.B – Team Review Sign-off

Within this section of the UIDD document you will find the signatures, names and date of the documental approvals for each acting member of "Team Undershrub ". These signatures acknowledge and authenticate the approval and review of this document. This authentication includes the overall content, formatting, identification of contributed material and development directions presented within. These signatures must be updated alongside the date and comments when there is a subsational update to the document, which in this case includes any formatting, editing, grammar and change of materials within.

Name: Anthony Jackson	Date: 11 - 29 - 21
Signature: 	
Comments:	

Name: David Sincyr III	Date: 11 - 29 - 21
Signature: 	
Comments:	

Name: Devin Carter	Date: 11 - 29 - 21
---------------------------	---------------------------

Signature:

A handwritten signature in black ink, appearing to read "Devin Carter". The signature is fluid and cursive, with a small mark resembling a bird or a flower above the letter 'i'.

Comments:

Name: Stephen Kaplan	Date: 11 - 29 - 21
-----------------------------	---------------------------

Signature:

A handwritten signature in black ink, appearing to read "Stephen Kaplan". The signature is written in a flowing, cursive script.

Comments:

Name: Grant Shotwell	Date: 11 - 29 - 21
-----------------------------	---------------------------

Signature:

A handwritten signature in black ink, appearing to read "Grant Shotwell". The signature is written in a bold, cursive style.

Comments:

Subappendix C.C – Document Contributions

Identify how each member contributed to the creation of this document. Include what sections each member worked on and an estimate of the percentage of work they contributed. Remember that each team member must contribute to the writing (includes diagrams) for each document produced.

Grant Shotwell	<ul style="list-style-type: none">- Made Framer mockups for Section 3, added screenshots.- Wrote the general overview description for Section 3.- Assisted with descriptions and design commentary for Section 3.- 20% of work completed.
Stephen Kaplan	<ul style="list-style-type: none">- Assisted with section 2.- Created multiple figures in section 2.- Helped develop mock-ups of the UI for section 2.- Assisted with section 4.- 20% of work completed.
Devin Carter	<ul style="list-style-type: none">- Assisted with definitions for Section 2.- Assisted David with Section 5.- Gave advice and assisted with Section 4.- Helped develop some mock-ups of the UI for section 2 & 3.- 20% of work completed.
David Sincyr III	<ul style="list-style-type: none">- Completed the Introduction section- Created an initial UI illustration- Assisted Devin Carter with Section 5- Created multiple mock-ups of the UI- Wrote the Introduction section to section 2- Formatted the document- 20% of work completed.
Anthony Jackson	<ul style="list-style-type: none">- Completed and updated Appendix A, B & C.- Created Section 4 and filled in a majority of the table alongside its description.- Assisted with definitions for Section 2.- Assisted with descriptions and design commentary for Section 3, also modifying the description.- 20% of work completed.