

Figure 12: This details the options an actor has regarding their projects and how it will interact with the system

<b>Number</b>	66	
<b>Name</b>	View List of Projects Owned By Self	
<b>Summary</b>	General user views the list of projects they personally own.	
<b>Priority</b>	3	
<b>Preconditions</b>	General user must be viewing the list of projects.	
<b>Postconditions</b>	General user narrows the search down to just their owned projects.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user attempts to refine the projects they're viewing.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user uses the project view window to navigate to a tab that shows purely projects the user themselves own.
	2	System retrieves a list of projects owned by that user.
	3	System displays all self owned projects.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	< No Projects Available > : If no projects are available for display the system returns a screen where the search result area shows a notice regarding the lack of results.
<b>Open Issues</b>		

<b>Number</b>	67	
<b>Name</b>	View List of Projects Shared By Others	
<b>Summary</b>	General user views all projects they themselves did not create.	
<b>Priority</b>	3	
<b>Preconditions</b>	General user must be viewing the list of projects.	
<b>Postconditions</b>	General user narrows the search down to just those they do not own.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user attempts to refine the projects they're viewing.	
<b>Main Scenario</b>	1	General user uses the project view window to navigate to a tab that shows purely projects shared with the user.
	2	System retrieves a list of projects shared with that user.
	3	System displays all shared projects.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	< No Projects Available > : If no projects are available for display the system returns a screen where the search result area shows a notice regarding the lack of results.
<b>Open Issues</b>		

<b>Number</b>	68
---------------	----

<b>Name</b>	View List of Public Projects	
<b>Summary</b>	General user views all public projects.	
<b>Priority</b>	3	
<b>Preconditions</b>	General user must be viewing the list of projects.	
<b>Postconditions</b>	General user narrows the search down to just public projects.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user attempts to refine the projects they're viewing.	
<b>Main Scenario</b>	1	General user uses the project view window to navigate to a tab that shows purely public projects.
	2	System retrieves a list of all public projects
	3	System displays all public projects.
	<b>Step</b>	<b>Branching Action</b>
<b>Extensions</b>	3a	< No Projects Available > : If no projects are available for display the system returns a screen where the search result area shows a notice regarding the lack of results.
<b>Open Issues</b>		

<b>Number</b>	69	
<b>Name</b>	View List of Projects	
<b>Summary</b>	General user views a list of all visible projects.	
<b>Priority</b>	4	
<b>Preconditions</b>	General user is logged in, and some amount of projects exist.	
<b>Postconditions</b>	General user is given a view window showing all visible projects.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user uses a UI element to view all visible projects.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user chooses to view all visible projects.
	2	System retrieves a list of all visible projects
	3	System displays all visible projects.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	3a	< No Projects Available > : If no projects are available for display the system returns a screen where the search result area shows a notice regarding the lack of results.
<b>Open Issues</b>		

<b>Number</b>	70
---------------	----

<b>Name</b>	Filter List of Projects	
<b>Summary</b>	General user filters the list of visible projects via a defined criteria.	
<b>Priority</b>	2	
<b>Preconditions</b>	General user is viewing the list of visible projects.	
<b>Postconditions</b>	Project list refined to match the filter.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user selects the filter option.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Project Database Manager fetches a running list of all filter types for the project viewer.
	2	Project Database Manager displays all filter types to the user.
	3	General user selects one or more filter types and submits them to the Project Database Manager.
	4	Project Database Manager takes the inputted filters and narrows the search result.
	5	Project Database Manager returns the new search result.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Filter Cancellation > : General user closes out of the filter selection.
	5a	< No Projects Available > : Project Database Manager displays empty result with a notice for lack of matching projects.
<b>Open Issues</b>		

<b>Number</b>	71	
<b>Name</b>	Filter by Name	
<b>Summary</b>	General user filters the list of visible projects via a defined name.	
<b>Priority</b>	2	
<b>Preconditions</b>	General user is viewing the list of visible projects.	
<b>Postconditions</b>	Project list refined in attempt to find a project with matching, or near matching, name.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user selects the filter option.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user chooses to filter by name via inputting a name into a text box.
	2	Project Database Manager takes and sanitizes input.
	3	Project Database Manager searches for matching, or near matching, projects.
	4	Project Database Manager sorts the list of results based on closest match.
	5	Project Database Manager displays the valid list of projects.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	4a	< Exact Match >: In cases of exact matches the Project Database Manager highlights the match.
	5a	< No Projects Available >: Project Database Manager displays empty result with a notice for lack of matching projects.
<b>Open Issues</b>		

<b>Number</b>	72	
<b>Name</b>	Filter by Tag	
<b>Summary</b>	General user filters the list of visible projects via one or more defined tags.	
<b>Priority</b>	2	
<b>Preconditions</b>	General user is viewing the list of visible projects.	
<b>Postconditions</b>	Project list refined in an attempt to find all projects that fit the defined tag(s).	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user selects the filter option.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Project Database Manager fetches a running list of all existing tags.
	2	Project Database Manager displays the list of tags to the user.
	3	General user looks through the list of tags to find those that match their ideal search filter.
	4	General user selects one or more tags and submits them.
	5	Project Database Manager takes the inputted tags and narrows the search result.
	6	Project Database Manager returns the new search result.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Filter Cancellation > : General user closes out of the filter selection.
	3a	< Text Search > : [ Optional feature to consider]: general user types an input into a text box, narrowing the list of tags to exactly what they're looking for.
	4a	< Filtering Method > : General user can select to narrow search result via alternative means. By default the search filters for projects that meet every tag. Other options: either or, not both, not either or.
	6a	< No Projects Available > : Project Database Manager displays empty result with a notice for lack of matching projects.
<b>Open Issues</b>	< list of issues awaiting decisions that affect the use case >	

<b>Number</b>	73	
<b>Name</b>	Delete Selected Projects	
<b>Summary</b>	General user deletes selected projects assuming it's valid for them to do so.	
<b>Priority</b>	2	
<b>Preconditions</b>	General user is viewing existing projects.	
<b>Postconditions</b>	General user deletes the project assuming it is valid for them to do so.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user selects a project.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user selects the UI option for deleting a project.
	2	System makes a check to confirm that it is valid for the user to delete the selected project.
	3	System requests confirmation for deleting the selected project.
	4	General user confirms the request to delete.
	5	Project Database Manager deletes the requested project.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Option Denied > : User does not have the appropriate level of ownership for the project, the deletion option is unavailable.
	3a	< System Denies Request > : Some change happened since the user opened the project view window, the user no longer has the appropriate ownership and is denied.
	4a	< Deletion Cancellation > : User declines to continue the process of deleting the project.
	4a2	< Cancellation Notification > : System alerts user that the deletion was cancelled.
	5a	< Error Deleting > : An error occurs during the deletion process, the user is alerted via a notice detailing the failure in deletion.
<b>Open Issues</b>		

<b>Number</b>	74	
<b>Name</b>	Select Project(s) from List	
<b>Summary</b>	User selects one or more projects from a list of projects.	
<b>Priority</b>	4	
<b>Preconditions</b>	User has viewed a list of projects.	
<b>Postconditions</b>	User has one or more selected projects.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	User attempts to interact with the list of projects.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user utilizes the project list UI to highlight a particular project.
	2	System registers the particular project highlighted.
	3	System responds to the user by visibly "selecting" the project in their UI instance.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Multi-Select >: The user picks multiple projects to be highlighted and selected.
<b>Open Issues</b>		

<b>Number</b>	75	
<b>Name</b>	Share Selected Projects	
<b>Summary</b>	General user shares the selected project assuming they have the valid ownership/permission.	
<b>Priority</b>	3	
<b>Preconditions</b>	General user has selected one or more projects.	
<b>Postconditions</b>	General user shares all selected projects.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user desires to share the project.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user selects a UI element for sharing selected projects.
	2	System retrieves the view window for sharing projects.
	3	System displays a view window for the general user.
	4	General user inputs a list of users, or a defined group, to share selected projects with.
	5	General user sets what kind of permissions will be given to the shared with individuals.
	6	System receives user input and adds permissions to inputted users.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Lack of Initial Permissions >: General user lacks the permissions to share the selected project(s).
	4a	< Non-Exist User/Group >: An inputted user/group does not exist. System highlights non-existent inputs.
	6a	< System Error >: Some system error occurs, the user is notified that the share attempt was terminated.
<b>Open Issues</b>		

<b>Number</b>	76	
<b>Name</b>	Duplicate Selected Projects	
<b>Summary</b>	General user duplicates the selected project.	
<b>Priority</b>	1	
<b>Preconditions</b>	General user has selected a specific project and has the appropriate permissions.	
<b>Postconditions</b>	Selected project is copied and duplicated.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user desires to copy a specific project.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user goes through the process to select one or more projects.
	2	General chooses the UI option for copying and pasting the project.
	3	System responds by offering a view window with settings for the duplicated project.
	4	User inputs selection of settings for the duped project(s).
	5	System duplicates projects and alerts general user with a notice regarding the duplication success.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Lack of Initial Permissions > : General user lacks the permissions to duplicate the selected project(s).
	4a	< Default Settings > : General user can skip setting specific settings, under default settings the system will duplicate each project with its specific permissions.
	5a	< System Error > : Some system error occurs, the user is notified that the duplication attempt was terminated.
<b>Open Issues</b>		

<b>Number</b>	77	
<b>Name</b>	Tag Selected Projects	
<b>Summary</b>	General user applies a tag to a selected project(s).	
<b>Priority</b>	2	
<b>Preconditions</b>	General user selects a valid project, and some amount of tags exist in the system.	
<b>Postconditions</b>	Selected project(s) given one or more tags for future search filtering.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user wishes to make searching for their project easier.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user utilizes the UI to choose the tagging option for a project.
	2	System fetches a list of tags.
	3	System displays list of tags in a new view window.
	4	General user manually looks through the list for tags.
	5	General user selects a subsection of tags.
	6	System updates the project's associated tags.
	7	System displays a success notification for the user.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Lack of Initial Permissions >: General user lacks the permissions to tag the selected project(s).
	7a	< System Error >: Some system error occurs, the user is notified that the tagging attempt was terminated.
<b>Open Issues</b>		

<b>Number</b>	78	
<b>Name</b>	Create Templates from Selected Projects	
<b>Summary</b>	General user turns a selected project into a template.	
<b>Priority</b>	1	
<b>Preconditions</b>	General user selects some number of valid projects.	
<b>Postconditions</b>	General user creates one or more templates.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user wants to streamline the creation of future projects.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user chooses a templating option for the selected project(s).
	2	System displays a new window containing template options.
	3	User picks the appropriate template options and submits.
	4	System stores new template options for the user.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Lack of Initial Permissions > : General user lacks the permissions to make templates from the selected project(s).
<b>Open Issues</b>		

<b>Number</b>	79	
<b>Name</b>	Create New Tag	
<b>Summary</b>	General user creates a new type of tag.	
<b>Priority</b>	1	
<b>Preconditions</b>	General user has a project they're attempting to tag.	
<b>Postconditions</b>	General user creates a new kind of tag.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user wishes to apply custom filter tags to their project.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user selects to add a tag to their project.
	2	General user selects to add a custom tag.
	3	System responds with a UI element for inputting a new tag, and a tag description.
	4	General user responds by inputting a tag name.
	5	System validates created tag.
	6	System responds with a success notification.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	5a	< Input Sanitation & Validation > : System sanitizes and validates user input.
	6a	< System Error > : Some system error occurs, the user is notified that the tag creation attempt was terminated.
<b>Open Issues</b>	Method for making sure users do not abusively have hundreds of fodder tags.	

<b>Number</b>	80	
<b>Name</b>	Set Tag Color	
<b>Summary</b>	General user gives a tag they made a unique color for visual identification.	
<b>Priority</b>	1	
<b>Preconditions</b>	General user has created their own tag.	
<b>Postconditions</b>	General user gives their tag a color identifier.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user wants to make their tag more visually distinct.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user chooses to edit the color of a tag.
	2	System displays the options of a color matrix, RGB input, and a hex code.
	3	General user chooses one option and inputs a color.
	4	System updates the tag's associated color based on input.
	5	System responds to the user with a success notification.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	< Lack of Initial Permissions > : General user lacks the permissions to tag the selected project(s).
	5a	< System Error > : Some system error occurs, the user is notified that the tagging attempt was terminated.
<b>Open Issues</b>		

<b>Number</b>	81	
<b>Name</b>	Select Existing Tag	
<b>Summary</b>	General user selects an existing tag for options.	
<b>Priority</b>	1	
<b>Preconditions</b>	General user is looking through the list of tags.	
<b>Postconditions</b>	General user successfully selects an existing tag, seeing either options or a description.	
<b>Primary Actor</b>	General user	
<b>Secondary Actors</b>	Project Database Manager	
<b>Trigger</b>	General user wishes to view a particular tag.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	General user tries to highlight a specific tag.
	2	System responds by fetching the description for the tag.
	3	System displays the description.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	< Higher Permissions > : If the user is the tag's creator, or someone of higher permission, the system fetches options for the tag in addition to the description.
<b>Open Issues</b>		

## Meta Data Use Case

Team Undershrub | November 1, 2021

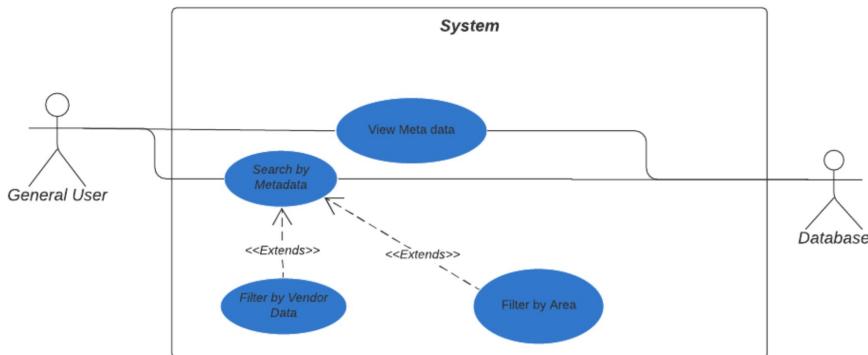


Figure 13: This shows how an actor will be able to search for projects in the database through metadata

<b>Number</b>	82	
<b>Name</b>	View Metadata	
<b>Summary</b>	General user views the metadata of a geospatial project.	
<b>Priority</b>	3	
<b>Preconditions</b>	User needs to be logged in and there needs to be some amount of data already stored in the database.	
<b>Postconditions</b>	User views the metadata of a data resource.	
<b>Primary Actor</b>	General User	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	User views available data.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User attempts to view the metadata of a data resource.
	2	Database returns the metadata components of a data resource to the user.
	3	User views the data resource's metadata detailing information on who/what/when/where/how/etc of the data resource.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Failure to return metadata : 2a1. System fails to return metadata. 2a2. System displays an error for the user.
<b>Open Issues</b>		

<b>Number</b>	83	
<b>Name</b>	Search By Metadata	
<b>Summary</b>	The user searches available data via metadata components.	
<b>Priority</b>	3	
<b>Preconditions</b>	User needs to be logged in and there needs to be some amount of data already stored in the database.	
<b>Postconditions</b>	User receives a narrowed search based on a metadata filter.	
<b>Primary Actor</b>	General User	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	General user attempts to create a filtered search.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User attempts to filter their search.
	2	System prompts the user for a filter.
	3	User returns a filter based on the metadata of stored data resources.
	4	Database returns the resulting filtered data search.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	4a	Empty Result : 2a1. No matching data can be returned based on the given metadata filter. 2a2. User receives an empty result with a note detailing the complete lack of results.
<b>Open Issues</b>		

<b>Number</b>	84	
<b>Name</b>	Filter By Area	
<b>Summary</b>	The user searches available data via the metadata component detailing where the data comes from.	
<b>Priority</b>	3	
<b>Preconditions</b>	User needs to be logged in and there needs to be some amount of data already stored in the database.	
<b>Postconditions</b>	User receives a narrowed search based on a metadata filter.	
<b>Primary Actor</b>	General User	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	General user attempts to create a filtered search.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User attempts to filter their search.
	2	System prompts the user for a filter.
	3	User returns a metadata filter that narrows the returned data to data that comes from a specific area.
	4	Database returns the resulting filtered data search.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	4a	Empty Result : 2a1. No matching data can be returned based on the given metadata filter. 2a2. User receives an empty result with a note detailing the complete lack of results.
<b>Open Issues</b>		

<b>Number</b>	85	
<b>Name</b>	Search By Vendor Data	
<b>Summary</b>	The user searches available data via vendor data.	
<b>Priority</b>	3	
<b>Preconditions</b>	User needs to be logged in and there needs to be some amount of data already stored in the database.	
<b>Postconditions</b>	User receives a narrowed search based the inputted vendor data.	
<b>Primary Actor</b>	General User	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	General user attempts to create a filtered search.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User attempts to filter their search.
	2	System prompts the user for a filter.
	3	User returns a filter based on inputted vendor data.
	4	Database returns the resulting filtered data search.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	4a	Empty Result : 2a1. No matching data can be returned based on the given metadata filter. 2a2. User receives an empty result with a note detailing the complete lack of results.
<b>Open Issues</b>		

## Use Case Data Visualization

Team Undershrub | November 1, 2021

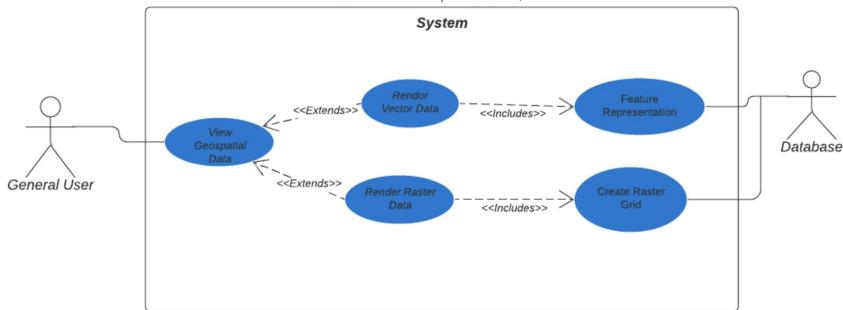


Figure 14: This details the data visualization process by an actor and through the database

<b>Number</b>	86	
<b>Name</b>	View Geospatial Data	
<b>Summary</b>	General User views geospatial data.	
<b>Priority</b>	3	
<b>Preconditions</b>	User needs to be logged in and there needs to be some amount of geospatial data that can be visualized.	
<b>Postconditions</b>	User receives a visualization of the given geospatial data.	
<b>Primary Actor</b>	General User	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	User searches for available geospatial data.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User attempts to view a select geospatial data component.
	2	Database registers what kind of geospatial data the selected component is.
	3	Database returns a visualization of the geospatial data.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Failure to determine type of geospatial data : 2a1. System fails to determine if the data is vector data or raster data. 2a2. System displays an error for the user and returns basic information in place of a visualization.
<b>Open Issues</b>		

<b>Number</b>	87	
<b>Name</b>	Render Vector Data	
<b>Summary</b>	General User requests a visualization of vector data.	
<b>Priority</b>	1	
<b>Preconditions</b>	User requested to view some geospatial data component and the component was vector data.	
<b>Postconditions</b>	User is returned a visualization of vector data.	
<b>Primary Actor</b>	General User	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	User requests for some geospatial data component to be visualized.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User requests for geospatial data to be visualized, their requested data is vector data.
	2	Database renders a visualization based on the vector data's inputted features and how those features are represented.
	3	User is returned a visualization of the vector data and its various features.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Failure to render vector data : 2a1. System fails to render the vector data due to some issue with the data itself or part of the database itself. 2a2. System displays an error for the user.
<b>Open Issues</b>		

<b>Number</b>	88	
<b>Name</b>	Render Raster Data	
<b>Summary</b>	General User requests a visualization of raster data.	
<b>Priority</b>	1	
<b>Preconditions</b>	User requested to view some geospatial data component and the component was raster data.	
<b>Postconditions</b>	User is returned a visualization of raster data.	
<b>Primary Actor</b>	General User	
<b>Secondary Actors</b>	Database	
<b>Trigger</b>	User requests for some geospatial data component to be visualized.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User requests for geospatial data to be visualized, their requested data is raster data.
	2	Database renders a pixelated raster data visualization.
	3	User is returned a visualization of the raster data and its various features.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Failure to render raster data : 2a1. System fails to render the raster data due to some issue with the data itself or part of the database itself. 2a2. System displays an error for the user.
<b>Open Issues</b>		

<b>Number</b>	89	
<b>Name</b>	Render Vector Data	
<b>Summary</b>	Vector data rendered based on what kind of vector data it is.	
<b>Priority</b>	1	
<b>Preconditions</b>	Some vector data component needs rendering.	
<b>Postconditions</b>	Vector data successfully rendered.	
<b>Primary Actor</b>	Database	
<b>Secondary Actors</b>	General User	
<b>Trigger</b>	Vector data exists.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Some type of geospatial vector data is stored in the database.
	2	General user requests the vector data to be visualized.
	3	Vector data determined to be either point, line, or polygon data.
	4	Point/line/polygon data visualized onto a map.
	5	Additional information added to visualization as required (geographic location names).
	6	Vector visualization returned to the user that requested the visualization.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Failure to properly assign meaning : 2a1. System fails to assign appropriate point/line/polygon data to the visualization. 2a2. System displays an error for the user.
<b>Open Issues</b>	How to translate point, line, and polygon data onto a visualization.	

<b>Number</b>	90	
<b>Name</b>	Create Raster Grid	
<b>Summary</b>	Cell matrix rendered and organized for future visualization to a user.	
<b>Priority</b>	1	
<b>Preconditions</b>	Some raster data component needs rendering.	
<b>Postconditions</b>	Raster data successfully rendered.	
<b>Primary Actor</b>	Database	
<b>Secondary Actors</b>	General User	
<b>Trigger</b>	Raster data exists.	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Aerial photograph, satellite imagery, scanned map, etc stored in database as raster data.
	2	General user requests the raster data to be visualized.
	3	Matrix of cells created and organized into a grid.
	4	Every cell assigned a value representing some sort of geospatial information
	5	Raster visualization returned to the user that requested the visualization.
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	4a	Failure to properly assign meaning : 2a1. System fails to assign appropriate information to each cell. 2a2. System displays an error for the user.
<b>Open Issues</b>	How to read given input, such as an aerial photograph or satellite image, and create an organized cell matrix detailing information associated with that input.	

### A.2.2 Test Cases

Test Case ID:	Test Objective:	Precondition:	Steps:	Test Data:	Expected Result:	Post-condition:
1_Create_Script	create an empty script	Authenticated user	1. A user selects the open script editor option 2. The system opens a new window and the user types in their script 3. The user selects the save script option	predefined scripts	The script window opens, the user is able to create a script and it is saved to the users account	The script appears in the users account
2_Use_Script	Uses an already created script	Authenticated User	1. A user selects or writes a script 2. A user selects run script 3. The script runs and performs its intended function	predefined scripts	The script is run through a docker container and performs the intended function.	The user can edit the script, close the script editing window or leave it open.
12_File_tests	Test the various file options: view, add, edit and delete	Authenticated user	1. select view files 2. select add file and add a file 3. Save the file then hit the edit button 4. edit the file 5. selected the edited file and hit delete	various file types	A file is addable, viewable, editable, and can be deleted.	The file is added, edited and removed from the database
10_Roles	Test the create and delete roles feature	Authenticated group or system moderator	1. Select Create Role option 2. Add in a role 3. Delete the role	NA	A role can be successfully added and deleted	The role has been added and saved or a previous one has been removed
16_Manage_role	Test the manage roles with giving and revoking permissions per role	Authenticated group or system moderator	1. Selects an already created role 2. select the give permissions option 3. Select a user in the group 4. Give them permissions 5. Test to see if they can perform the permissions granted 6. Revoke their permissions	NA	A role can be given permissions and revoked successfully	If a role is granted that role can now perform those duties until revoked.

<b>Test Case ID:</b>	<b>Test Objective:</b>	<b>Precondition:</b>	<b>Steps:</b>	<b>Test Data:</b>	<b>Expected Result:</b>	<b>Post-condition:</b>
22_Del_Data	Delete user data	Existing Data	1. Select data 2. Select/click delete data	Various data types	The user's data is deleted.	The data is removed from the users account
15_Add_Data	Save users data to the database	Authenticated user	1. User selects add data 2. User selects the data to add 3. The user selects/clicks the add button	Various data types (TIFF, JPEG, DEM, SHP, KML)	A user's data is uploaded and saved to the database	The data appears in the users data list
Valid_Login	Successful Login	Valid Account	1. Click login button 2. Enter Credentials	Valid username	A user is authenticated	The user is brought to their homepage
Invalid_Login	Unsuccessful Login	Invalid Account	1. Click login button 2. Enter Credentials 3. Hit login	Invalid username	A user is not authenticated	The user is asked for credentials again
Download_Data	Successful Download of Data	Already added data to the system.	1. Select data. 2. Select/click download data. 3. System will download the data to the user's system.	Various data types (TIFF, JPEG, DEM, SHP, KML)	The data is correctly downloaded to the user's system.	The data is marked as downloaded.
Check_Auto_Refresh	Successful completion of a refresh.	System timer of 10 minutes.	1. Create a 10 minute timer. 2. Create a check to see if the system refreshes within 10 minutes of the timer start.	The web application.	The web application completes both an automatic refresh	The system's information is refreshed.
Check_Refresh	Successful completion of a refresh.	Added functionality to refresh the page.	1. Click the refresh button. 2. Check to see if the system refreshes.	The web application.	The web application completes a manual refresh.	The system's information is refreshed.
Check_Buffer	Measure how long a loading screen takes.	System timer measuring in milliseconds.	1. Set up a timer measured in milliseconds. 2. Click onto another page.	The web application.	The web application loads in less than one second.	The system is loaded in less than a second.

### A.3. Non-Functional Requirements

Decide on a standard format for the non-functional requirements (NFRs). Included in the format should be a unique number for each NFR, a priority (1 = lowest, 5 = highest), a clear, concise description, and the test(s) that will be used during system and acceptance testing to verify that the requirement has been met. Make sure that the test numbers correspond to the NFR numbers. Note that you must include a minimum of 10 NFRs specific to product requirements, organizational requirements, and external requirements.

ID #	Priority	User Story	Requirement	Testcase
NF01	4	As a user, I need my data available to view 99% of the time.		The aftermath of an Add_Data test will allow the ability to see if any additional data is added to the system and is viewable from the menu.
NF02	1	As a user, my data must remain private to anyone not in my group's hierarchy.		Roles and Manage_Rolls are the two best test cases to see if data marked for a specific roll are viewable by others rolls if marked private.
NF03	3	As a product owner, I need the web application to work on Windows OS and MacOS browsers.		This can only be tested after a successful Valid_Login check to open the browser application.
NF04	3	As the user, I need to be able to login on the first attempt 99.9% of the time.		Valid_Login and Invalid_Login are the test cases which will see if a login has been correctly carried out by the system.
NF05	2	As a product owner, I want the web application to give (at least) limited access on mobile devices.		While there is no specific test case for this condition, you would just need to login to the application on a phone browser to see if the UI functions correctly.
NF06	3	As a user worried about security, I need my data secured from those without permission to access.		Once again, Roles and Manage_Rolls are suitable for this test as you can mark a specific variable for one subclass and see if it's viewable by another.
NF07	5	As a product owner, I want the web application to be available 99% of the time.		This can be tested through viewing the web application, with no particular test case being needed.

ID #	Priority	User Story	Requirement	Testcase
NF08	4	As a user with good payment standing, I must be able to download my data at any time.		Download_Data can be used to see if a file format can be downloaded and if any errors occur.
NF09	3	As a site administrator, I need management information to dynamically refresh every 10 minutes or when I choose to manually refresh it.		Check_Auto_Refresh and Check_Refresh can be used to see if the automatic and manual refresh functions correctly execute.
NF10	3	As a user, I would like the website to never load/buffer on a still screen for longer than a second.		You can use the Check_Buffer test to measure how long a loading screen takes to finish.
NF11	5	As a site administrator, python scripts need to be run such that they are contained without the capacity of damaging the larger systems.		This testing would rely on Docker's built in functionality and therefore cannot be easily tested with a simple test case.
NF12	4	As a user, my data uploads should not fail 99.9% of the time.		The function Add_Data can be used to test if there are any errors in data uploading, easily done through running the test case until finding a degree of accuracy.
NF13	2	As a general user, I want the UI to move/transition smoothly to improve my overall experience.		This cannot be tested until the UI development phase and is judged aesthetically rather than a setting function.

#### A.4. User Interface

See the User Interface Design Document for Geospatial Data Portal attached to this document.

## A.5. Deliverables

These deliverable artifacts are available in electronic format alongside this document and they can be printed for hard copies when needed by either a customer, client or team member.

- **System's Requirement Specification (SRS)**
  - Delivery date: Oct 25th 2021
  - *The SRS document will be available in .PDF and .DOCX formats within the shared team Github repository for the Geospatial Data Portal alongside a private drive available to any employee of Blue Marble Geographics.*
- **System Design Document (SDD)**
  - Delivery date: Nov. 10th
  - *The SSD document will be available in .PDF and .DOCX formats within the shared team Github repository for the Geospatial Data Portal alongside a private drive available to any employee of Blue Marble Geographics.*
- **User Interface Design Document (UIID)**
  - Delivery date: Nov. 19th 2021
  - *The UIID document will be available in .PDF and .DOCX formats within the shared team Github repository for the Geospatial Data Portal alongside a private drive available to any employee of Blue Marble Geographics.*
- **User Manual (UM)**
  - Delivery date: Spring 2022
  - *The user manual will be available in both the .PDF and .DOCX formats within the shared team Github repository for the Geospatial Data Portal alongside being available within a public google drive where any individual will be allowed to download the file.*
- **Administrator Manual (AM)**
  - Delivery date: Spring 2022
  - *The administrator manual will be available in both the .PDF and .DOCX formats within the shared team Github repository for the Geospatial Data Portal alongside being available within a public google drive where any individual will be allowed to download the file.*

These deliverable artifacts are **only available** in electronic form.

- **Source code for the Geospatial Data Portal.**
  - Delivery date: Continuous
  - *This source code for the Geospatial Data Portal will be available on Github within the shared team repository for Team Undershrub.*

- **The executable of the Geospatial Data Portal.**
  - Tentative delivery date: May 10th 2022
  - *The executable for the Geospatial Data Portal will be a website linked off the main portal for Blue Marble Geographics at their official web address <https://www.bluemarblegeo.com/>.*
- **A brief description and link to any required software for installation and execution of the Geospatial Data Portal.**
  - *Although there is no required software for this online application, it is recommended to have a personalized IDE for creating and editing any uploaded Python scripts to the software.*

## A.6. Open Issues

The following list is the current open issues the group has realized thus far with more to follow:

1. Define "access".
2. API documentation is probably needed.
3. Scope of API should be determined.
4. What content can be matched?
5. Where are tags created: in this action or in a separate action?
6. Use buttons or menu options
7. Allowing project owners to have some limited capacity for monitoring anonymous traffic?
8. Method for making sure users do not abusively have hundreds of fodder tags.

## Subappendix A.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 Geographics. 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 Geographics. 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 Geographics 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 Geographics 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 SRS 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:	Date:
Signature: 	

**Subappendix A.B – Team Review Sign-off**

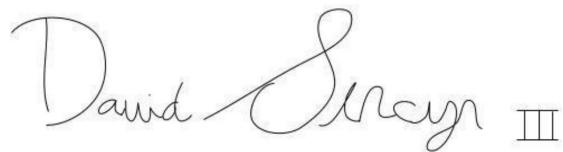
Within this section of the SRS document you will find the signatures, official typed name, date and document comments from all the team members of "Team Undershrub". These signatures authenticate the approval of the content, formatting and the identification of which material each team member has contributed. These signatures will be updated alongside the date and comments when an substantial update is to be made to the document, in this case a sustainable update would be anything that is not minor grammatical or formatting changes.

Name: Anthony Jackson	Date: 10 - 20 - 21
Signature: 	

**Name:** David Sincyr III

**Date:** 10 - 20 - 21

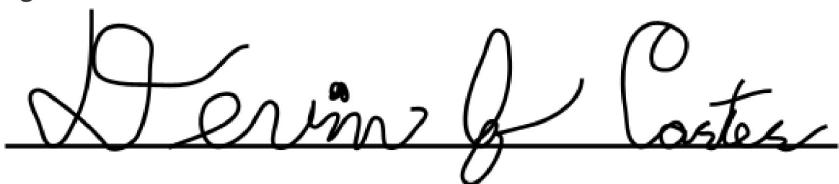
**Signature:**



**Name:** Devin Carter

**Date:** 10 - 20 - 21

**Signature:**



**Name:** Stephen Kaplan

**Date:** 10 - 20 - 21

**Signature:**



**Name:** Grant Shotwell

**Date:** 10 - 21 - 21

**Signature:**



## Subappendix A.C – Document Contributions

Grant Shotwell 20%	<ul style="list-style-type: none"> <li>- Created use cases alongside creating use case diagrams.</li> <li>- Added the team logo and formatting.</li> </ul>
Stephen Kaplan 20%	<ul style="list-style-type: none"> <li>- Created use cases alongside creating use case diagrams.</li> <li>- Created the team logo.</li> <li>- Adjusted wording throughout the document to increase clarity and consistency.</li> </ul>
Devin Carter 20%	<ul style="list-style-type: none"> <li>- Created use cases alongside creating use case diagrams.</li> <li>- Added changes requested by the client</li> </ul>
David Sincyr III 20%	<ul style="list-style-type: none"> <li>- Developed use cases</li> <li>- Created use case diagrams.</li> <li>- Wrote and edited the introduction section</li> <li>- Added dates to section 5</li> <li>- Added and modified all Use Case Diagrams</li> <li>- Created, numbered, formatted and named most use case descriptions</li> <li>- Wrote Test Cases</li> <li>- Updated the table of contents to match the format and for easier navigation</li> <li>- Edited the document for easier readability</li> <li>- Added in missing title page requirements</li> <li>- Wrote the open issues section</li> <li>- Added document version history</li> </ul>
Anthony Jackson 20%	<ul style="list-style-type: none"> <li>- Created 3 use case diagrams and around 15 test cases.</li> <li>- Added 17, 18, 50, 51, 52, 53, 54, 55, 56, 57 in official format</li> <li>- Added all the non-functional requirements to section 3, alongside setting up the formatting and creating test cases for them.</li> <li>- Worked on a majority of section 5 deliverables due dates.</li> <li>- Wrote the signature information and legal informant for Appendix A &amp; Appendix B.</li> </ul>

# Appendix B - System Design Document

## Table of Contents

<b>1 Introduction</b>	<b>98</b>
Purpose of This Document	98
References	98
<b>2 System Architecture</b>	<b>99</b>
2.1 Architectural Design	99
System Architecture Description	99
System Architecture UML Diagram	100
List of Components	100
2.2 Decomposition Description	101
Cloud Architecture Description	101
List of Components	106
Backend Architecture Description	106
List of Components	106
High-level Sequence Diagrams	107
Frontend Architecture Description	109
<b>3 Persistent Data Design</b>	<b>111</b>
Database Descriptions	112
File Descriptions	112
<b>4 Requirements Matrix</b>	<b>115</b>
<b>Appendix B.A – Agreement Between Customer and Contractor</b>	<b>116</b>
<b>Appendix B.B – Team Review Sign-off</b>	<b>117</b>
<b>Appendix B.C – Document Contributions</b>	<b>119</b>

## Document History

Name	Date	Reason for Changes	Version
David Sincyr	10/25/2021	Initial Creation	1.0
All Team Members	11/10/2021	Added in all relevant information	1.1
All Team Members	11/15/2021	Added in client changes	1.2
David Sincyr	12/9/2021	Removed issues based on feedback	1.2

## B.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. Once in the database, they will have 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.

### B.1.1 Purpose of This Document

The purpose of this System Design Document is to describe how the web portal will be designed. Specifically, it will detail the system at the architecture level, all subsystems and their services, data management systems, the overall software control structure, all hardware mappings, define access controls, and boundary conditions. The intended readership of this document is the project manager, the quality assurance manager, system developers, and system owner. Furthermore, this document is meant for all stakeholders to analyze the web portal prior to deployment to further their understanding of the system's functionalities.

### B.1.2 References

1. System Requirements Specification: [SRS\\_BlueMarbleGeographics\\_F21.docx](#)
2. User Interface Design Document: [UIDD\\_Template\\_F21.docx](#)
3. <https://fordfiringorder.com/>
4. <https://armacad.info/international-center-for-agribusiness-research-and-education--2019-09-05--certificate-program-geographic-information-systems-and-remote-sensing-rs-icare>
5. <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm>
6. System Design Document Template Adapted from Susan Mitchell

## B.2 System Architecture

This section will include a high level overview of the web portal that will detail the overall structure and design as it relates to the hardware and software requirements. It will use unified modeling language in order to better represent the design and provide the documents intended readership an easy to read format.

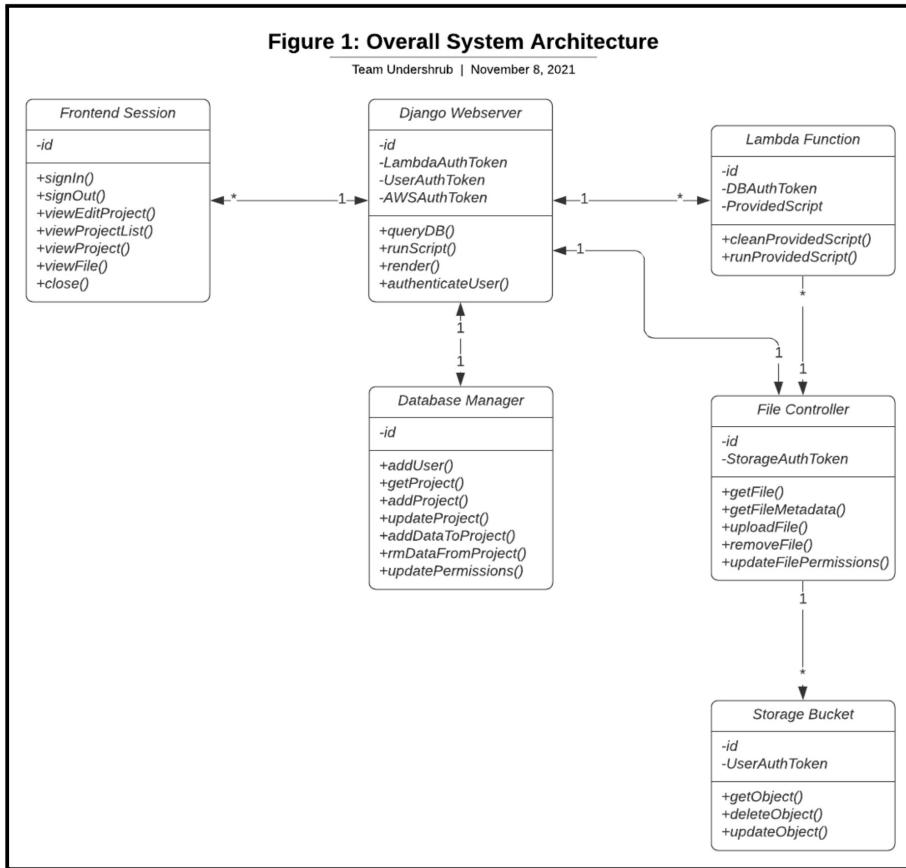
### B.2.1 Architectural Design

#### System Architecture Description

At a high level, the Geospatial Data Portal system consists of a frontend website, a database, file storage, lambda functions, and a Django backend server managing communication between each of these services. From the frontend site, users will be able to create projects, upload data, manage access to their projects, among other tasks. Queries from the frontend session to the backend Django server allow the user's actions to carry out their desired tasks. Django communicates with the Database Manager and the File Controller via authenticated API calls. The Django backend server also invokes lambda functions to carry out tasks specified in user-provided Python code. Lambda functions use API calls to query the File Controller and then return (to the Django server) some response (such as the access tokens for a newly updated dataset). In this process, the File Controller communicates with a file storage system such as Amazon S3 or AWS EFS to execute commands against user-uploaded data.

A UML diagram of the overall system architecture is provided in Figure 1. This high-level diagram highlights the relationship and distinct responsibilities of the frontend, Django backend, and cloud-based backend of the system. Section 2.2 identifies the system's subsystems and their components.

caption: This figure is a high level view of the architecture for the entire system



#### List of Components

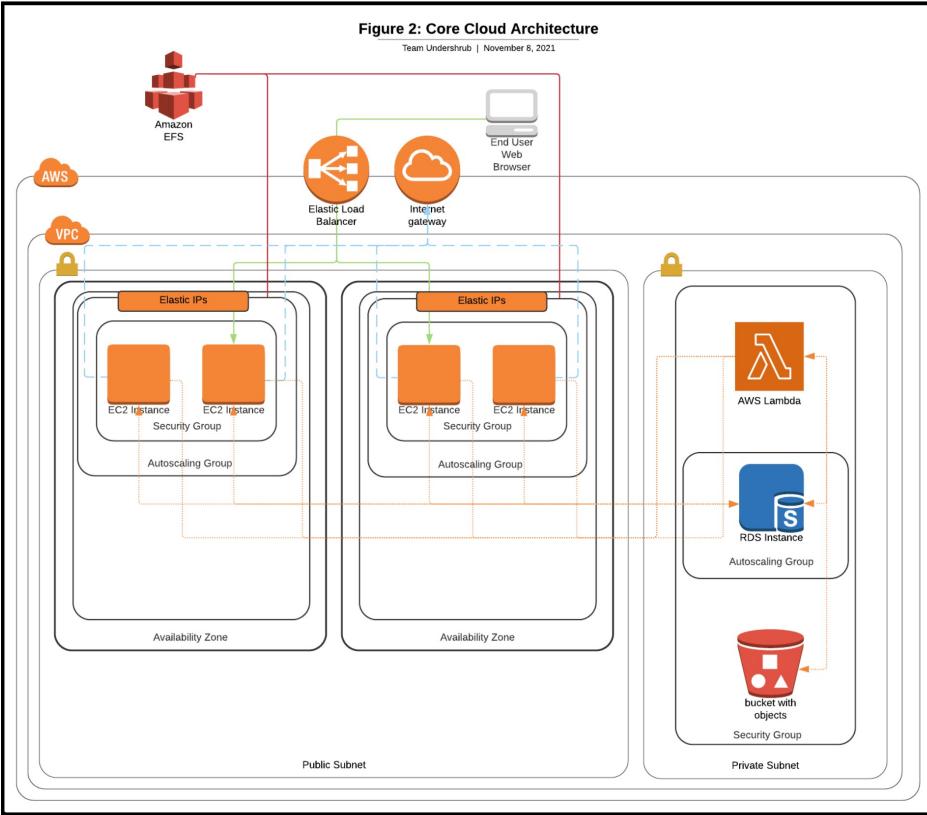
- **Front end:** Javascript React
  - Homepage
    - Project List view
    - Project view
    - File view
    - Project edit view
    - Data visualizer
- **Back end:** Django
  - AWS Database
    - MySQL
  - AWS Lambda
    - S3 Buckets
  - AWS Web Hosting Services

## **B.2.2 Decomposition Description**

This section of the system design document will break down the various subsystems, functions, objects, files, scripts and all major components of the system. The purpose and function of each component will also be detailed.

## **Cloud Architecture Description**

The cloud architecture will incorporate several services provided by Amazon Web Services. The core cloud architecture consists of one or more EC2 instances, a MySQL database hosted on RDS, data stored in S3 buckets and/or EFS, and dynamically generated AWS Lambda instances. The primary auxiliary cloud architecture consists of services such as load balancing which improve the resiliency, reliability, and availability of the system. The secondary auxiliary cloud architecture consists of services necessary for development and continued maintenance of the system. In this document, “cloud architecture” refers to the combination of the core and both auxiliary architectures.



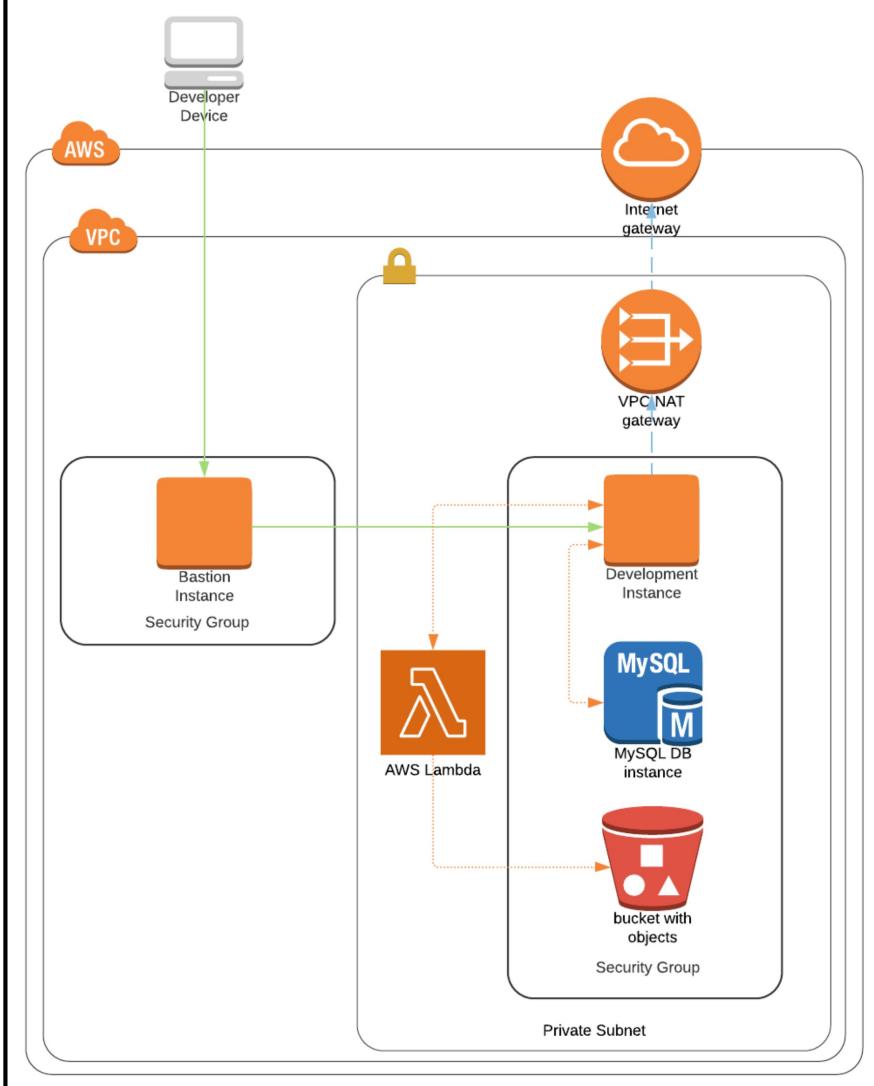
caption: This figure is a high level view of the cloud architecture the backend server will be running on

The Primary Auxiliary Cloud Architecture utilizes recommended cloud configurations to improve the performance of the system. An elastic load balancer directs traffic to web servers on different EC2 instances across multiple availability zones such that the load on any single instance is easily manageable. An autoscaling group ensures that the system is responsive to sudden increases in traffic. A VPC with a public and private subnet ensures that stored data (either in S3 or in RDS) is networked separately from the EC2 instances, decreasing the ability for malicious actors to gain unauthorized access. A separate autoscaling group for the RDS instance ensures resilience against high-traffic or large-scale data transfers. This subarchitecture is depicted in Figure 4. The Secondary Auxiliary Cloud Architecture provides a secure pathway for current developers and future maintainers of the system to connect and interact with an isolated

replica of the system. The system utilizes a MySQL database hosted on a private EC2 instance, with access restricted behind a separate bastion instance that enforces strict authentication protocols. All lambda functions and storage containers are secured in a private subnet which is depicted in figure 3.

**Figure 3: Secondary Auxiliary Cloud Architecture**

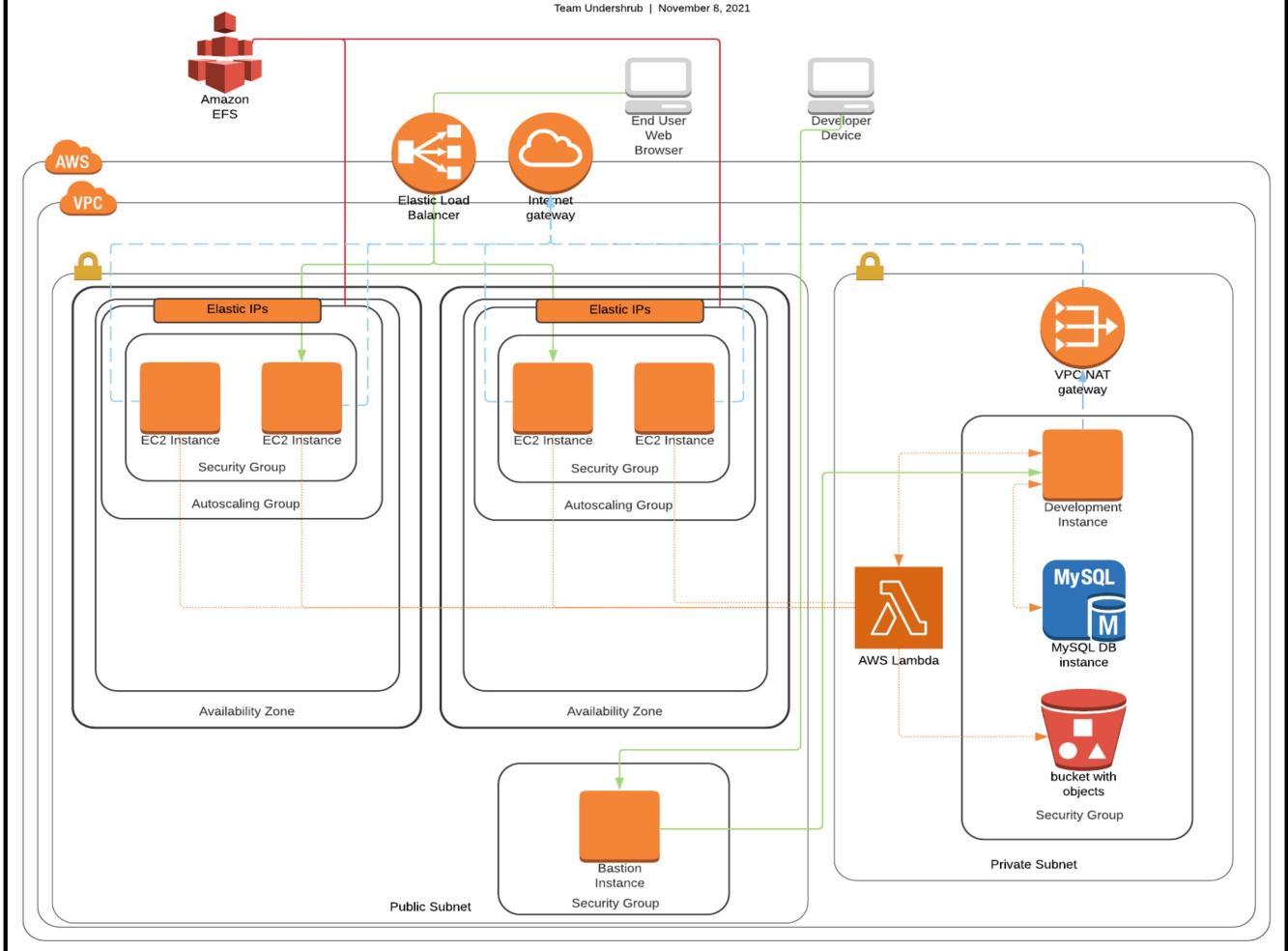
Team Undershrub | November 8, 2021



caption: The figure represents the architecture of the isolated replica of our main architecture

**Figure 4: Geospatial Data Portal Cloud Architecture**

Team Undershrub | November 8, 2021



caption: This depicts a high level view of the cloud architecture and how it interacts with various components such as the database and AWS Lambda