**Detailed Business Specific Requirements:-**

**Theme**: Self Service Analytics

**Applications**: WRIS-Utilities

**Use Cases:-** GeoViewer-**WRIS-SSA-04**

**Other linked Use Case :-** All Applications with spatial data layers.

**Major Use cases:** Surface Water Bodies (WRIS-MIS-01), River Monitoring (WRIS-MIS-02), River Information (WRIS-MIS-03), Ground Water Prospects Study (2011) (WRIS-MIS-04), Forest / Tree Cover (WRIS-MIS-05), Land Degradation (2015-18) (WRIS-MIS-06), Land Use — Land Cover (WRIS-MIS-07), Soil Type (WRIS-MIS-08), Wasteland Study (WRIS-MIS-09), Rainfall (WRIS-MIS-10), Evapo-transpiration (WRIS-MIS-11), Soil Moisture (WRIS-MIS-12), Inland Navigation Waterways {(WRIS-MIS-13), Socio-Economic Census (WRIS-SSA-02), Wetland Inventory (WM-UC-01), Ramsar Sites (WM-UC-02), Wetland Catchment (WM-UC-03), Hydrometry of wetlands (WM-UC-04), Wetland monitoring (WM-UC-05), Wetland restoration (WM-UC-06), Water quality of wetlands (WM-UC-07), Trends in wetlands (WM-UC-08), Glacial Inventory (GA-UC-01), Glacial monitoring (GA-UC-03), Glacial mass balance study (GA-UC05), Snow gauging (GA-UC-06), Watershed snow cover area (GA-UC-04), Snow melt analysis (GA-UC-07), Glacial Hydrometry (GA-UC-08), Snowmelt Runoff to the River (GA-UC-10), Glacial Lakes Inventory (GA-UC-02), Change in size of glacial lakes (Near Real time) (GA-UC11), Glacial Lake Storage Status (GA-UC-12), Glacial Lake Cross Section Generation (GA-UC13), Glacial Lake Vulnerability assessment (GA-UC-14), Glacial lake outburst (GA-UC-09), District At A Glance (WRIS-MIS-16)

**Description**:- Geoviewer enables geographic visualization that deals solely with displaying information that has a geospatial component to it. It is a common window to most of the spatial layers to be seen altogether so as to get a whole picture of the data collected. it also provides rich set of tools and techniques supporting geo-spatial data analysis through enhanced visualization.

**Reframe :** Geoviewer facilitates geographic visualization by focusing exclusively on information with a geospatial element. It serves as a comprehensive interface for viewing multiple spatial layers simultaneously, allowing users to gain a complete understanding of the collected data. Additionally, it offers a robust array of tools and techniques that enhance geo-spatial data analysis through improved visualization capabilities.

**Use case of Description:**

**Title:** Enhancing Urban Planning with Geoviewer

**Actors:**

* Urban Planners
* City Officials
* Environmental Analysts
* Community Stakeholders

**Preconditions:**

* Geoviewer is installed and configured for use.
* Relevant geospatial data layers (e.g., zoning maps, population density, transportation networks, environmental impact assessments) are available and accessible.

**Main Flow:**

* **Data Integration:**
* Urban planners import various geospatial data layers into Geoviewer, including demographic data, land use, infrastructure, and environmental features.
* **Visualization:**
* Users utilize Geoviewer to visualize the integrated data layers on a single map interface. They can toggle different layers on and off to focus on specific aspects of the urban environment.
* **Analysis:**
* Planners employ the analytical tools provided by Geoviewer to assess the impact of proposed developments. They analyze factors such as population growth, traffic patterns, and environmental constraints.
* **Scenario Simulation:**
* Users create and visualize different development scenarios (e.g., residential, commercial, mixed-use) to evaluate potential outcomes and impacts on the community.
* **Stakeholder Engagement:**
* Urban planners share visualizations with city officials and community stakeholders during public meetings. The clear, interactive maps facilitate discussions and feedback on proposed projects.
* **Decision Making:**
* Based on the insights gained from the visualizations and analyses, urban planners make informed recommendations to city officials regarding zoning changes, infrastructure investments, and community development initiatives.

**Postconditions:**

* Urban planners have a comprehensive understanding of the spatial dynamics at play in the urban environment.
* Decisions regarding urban development are supported by data-driven insights, leading to more sustainable and effective planning outcomes.

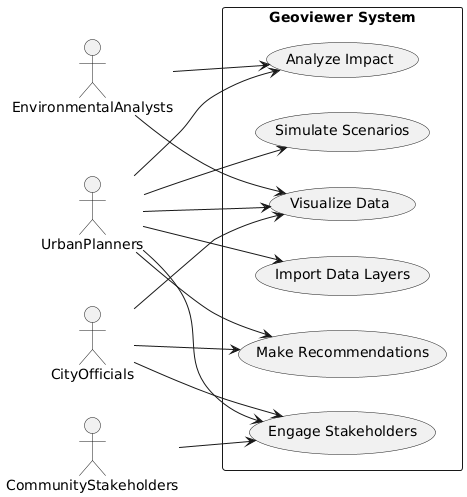
**Extensions:**

* If new data becomes available (e.g., updated census data or environmental studies), users can easily integrate this information into Geoviewer for ongoing analysis.
* Users can export visualizations and reports for documentation and further dissemination to stakeholders.

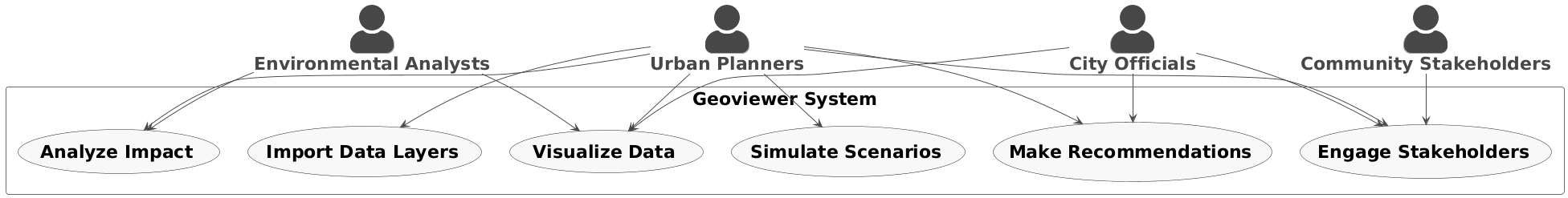
**Summary:**

The use case "Enhancing Urban Planning with Geoviewer" demonstrates how urban planners and stakeholders can leverage Geoviewer to improve urban development processes. Key participants include urban planners, city officials, environmental analysts, and community members. The scenario begins with the installation of Geoviewer and the availability of essential geospatial data layers, such as zoning maps and population density. Main activities involve integrating and visualizing these data layers, using analytical tools to assess the impacts of proposed developments, and simulating different development scenarios. Planners can share interactive visualizations with stakeholders to facilitate discussions and make informed recommendations based on the insights gained. This process leads to a comprehensive understanding of spatial dynamics and supports data-driven decisions for sustainable urban development. Additionally, the platform allows for the integration of new data and the export of visualizations for documentation. Overall, Geoviewer proves to be an invaluable tool for urban planners, enhancing their ability to analyze complex geospatial data and make effective planning decisions.

**Figure 001\_Intro\_Flowchart\_PlantUML**



**Figure 001\_Intro\_usecase\_plantUML 2**



**Code For Figure 001\_Intro\_usecase\_plantUML 2**

@startuml

!theme vibrant

skinparam actorstyle awesome

skinparam defaultfontsize 22

actor "\*\*Urban Planners\*\*" as UP

actor "\*\*City Officials\*\*" as CO

actor "\*\*Environmental Analysts\*\*" as EA

actor "\*\*Community Stakeholders\*\*" as CS

rectangle "Geoviewer System" {

usecase "\*\*Import Data Layers\*\*" as UC1

usecase "\*\*Visualize Data\*\*" as UC2

usecase "\*\*Analyze Impact\*\*" as UC3

usecase "\*\*Simulate Scenarios\*\*" as UC4

usecase "\*\*Engage Stakeholders\*\*" as UC5

usecase "\*\*Make Recommendations\*\*" as UC6

}

UP --> UC1

UP --> UC2

UP --> UC3

UP --> UC4

UP --> UC5

UP --> UC6

CO --> UC2

CO --> UC5

CO --> UC6

EA --> UC3

EA --> UC2

CS --> UC5

@enduml

**Code For Figure 001\_Intro\_usecase\_plantUML**

@startuml

left to right direction

actor UrbanPlanners as UP

actor CityOfficials as CO

actor EnvironmentalAnalysts as EA

actor CommunityStakeholders as CS

rectangle "Geoviewer System" {

usecase "Import Data Layers" as UC1

usecase "Visualize Data" as UC2

usecase "Analyze Impact" as UC3

usecase "Simulate Scenarios" as UC4

usecase "Engage Stakeholders" as UC5

usecase "Make Recommendations" as UC6

}

UP --> UC1

UP --> UC2

UP --> UC3

UP --> UC4

UP --> UC5

UP --> UC6

CO --> UC2

CO --> UC5

CO --> UC6

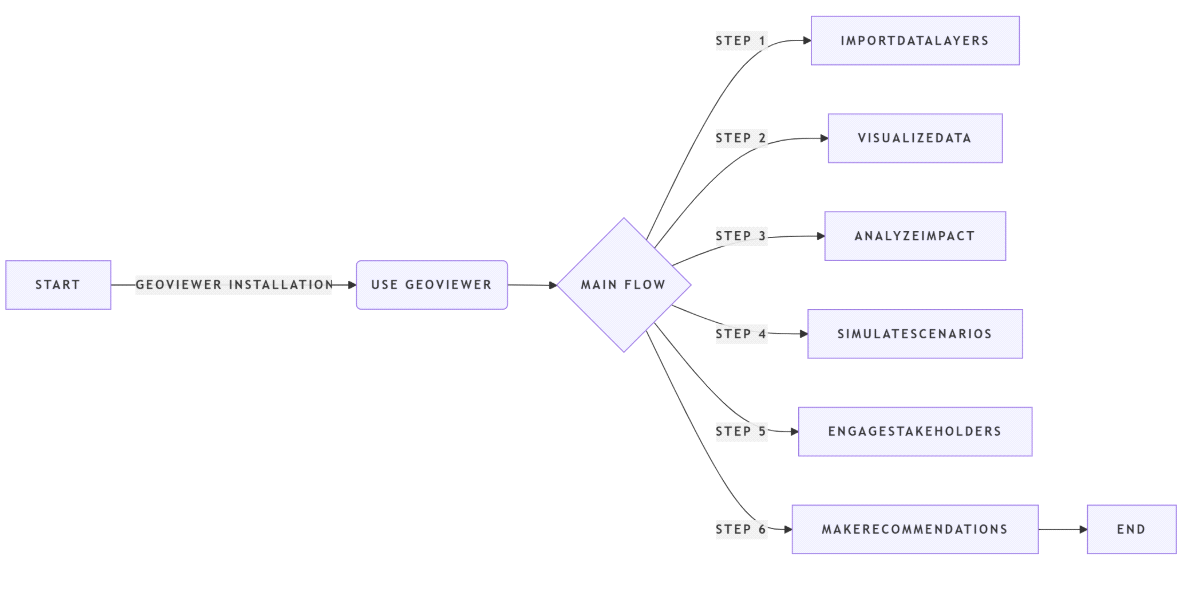
EA --> UC3

EA --> UC2

CS --> UC5

@enduml

**Figure 001\_Intro\_Flowchart\_NoteGPT**

**Code For Figure 001\_Intro\_Flowchart\_NoteGPT**

flowchart LR

A[Start] --Geoviewer Installation--> B(Use Geoviewer)

B --> C{Main Flow}

C -- Step 1 --> D[ImportDataLayers]

C -- Step 2 --> E[VisualizeData]

C -- Step 3 --> F[AnalyzeImpact]

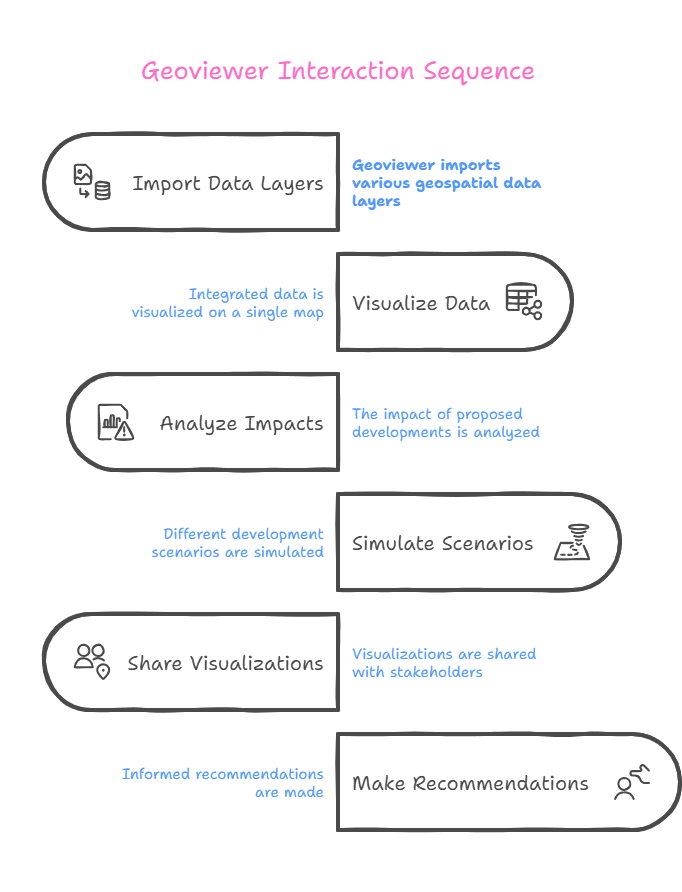
C -- Step 4 --> G[SimulateScenarios]

C -- Step 5 --> H[EngageStakeholders]

C -- Step 6 --> I[MakeRecommendations]

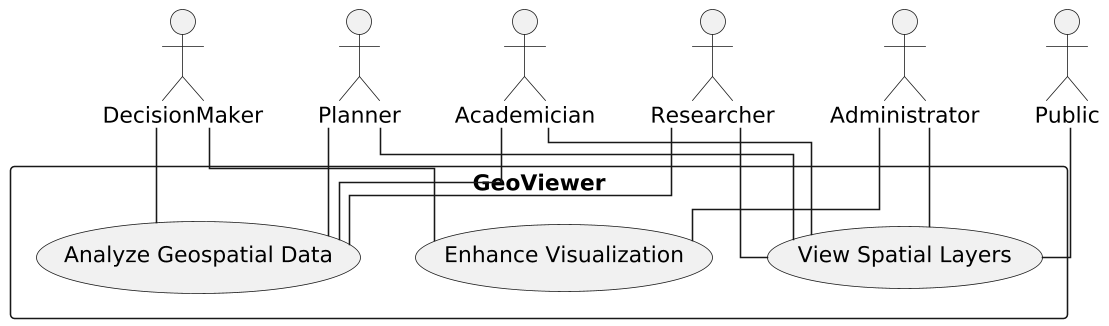
I --> J[End]

**Figure 001\_Intro\_Flowchart\_NapkinAI**

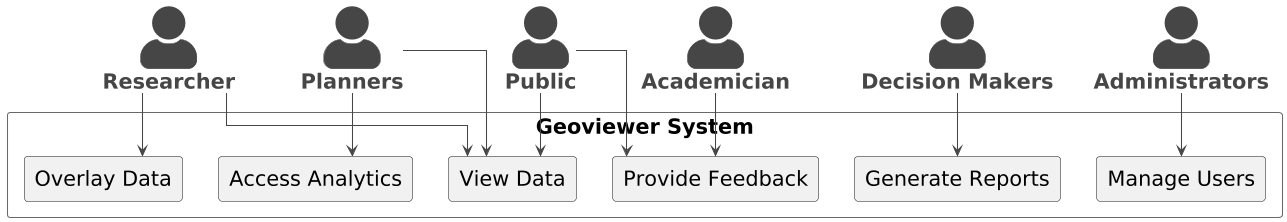


**Used By (End Users):-** Researcher, Planners, Decision makers, administrators, academicians and public.

**Figure 002\_End users\_Usecase\_PlantUML**



**Figure 002\_End users\_Usecase\_PlantUML 2**



**Code For Figure 002\_End users\_Usecase\_PlantUML**

@startuml

skinparam linetype ortho

skinparam dpi 150

actor Researcher

actor Planner

actor DecisionMaker

actor Administrator

actor Academician

actor Public

rectangle GeoViewer {

usecase "View Spatial Layers" as U1

usecase "Analyze Geospatial Data" as U2

usecase "Enhance Visualization" as U3

}

Researcher -- U1

Researcher -- U2

Planner -- U1

Planner -- U2

DecisionMaker -- U2

DecisionMaker -- U3

Administrator -- U1

Administrator -- U3

Academician -- U1

Academician -- U2

Public -- U1

@enduml

**Code For Figure 002\_End users\_Usecase\_PlantUML 2**

@startuml

!theme vibrant

skinparam linetype ortho

skinparam dpi 100

skinparam actorstyle awesome

skinparam usecasestyle rectangle

skinparam defaultfontsize 20

actor Researcher as "\*\*Researcher\*\*"

actor Planners as "\*\*Planners\*\*"

actor "\*\*Decision Makers\*\*" as DecisionMaker

actor Administrator as "\*\*Administrators\*\*"

actor "\*\*Academician\*\*" as Academician

actor Public as "\*\*Public\*\*"

rectangle "Geoviewer System" {

rectangle "(View Data)"

rectangle "(Overlay Data)"

rectangle "(Generate Reports)"

rectangle "(Access Analytics)"

rectangle "(Manage Users)"

rectangle "(Provide Feedback)"

}

Researcher --> "(View Data)"

Researcher --> "(Overlay Data)"

Planners --> "(View Data)"

Planners --> "(Access Analytics)"

DecisionMaker --> "(Generate Reports)"

Administrator --> "(Manage Users)"

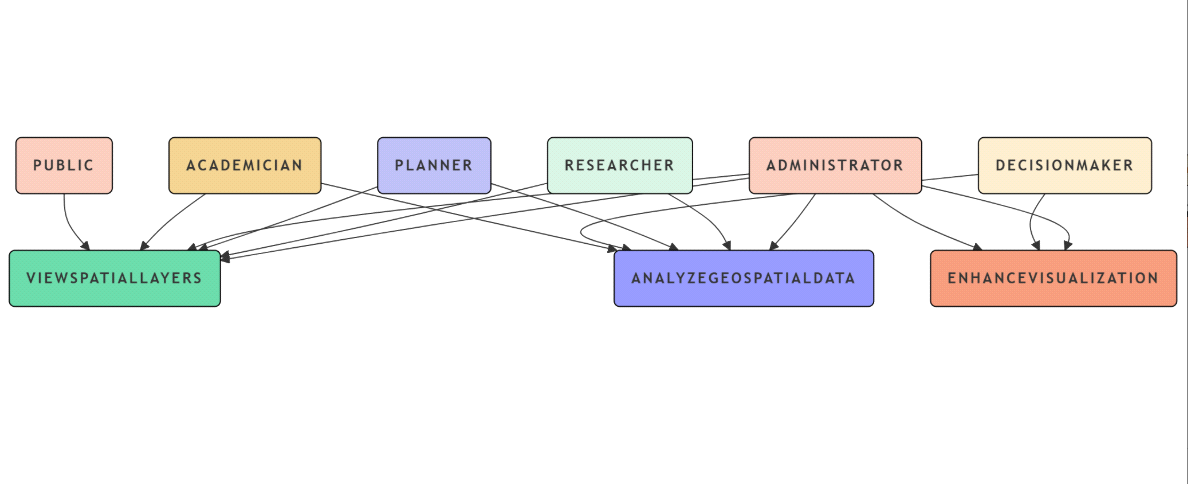
Academician --> "(Provide Feedback)"

Public --> "(View Data)"

Public --> "(Provide Feedback)"

@enduml

**Figure 002\_End users\_Flowchart\_NoteGPT**



**Code For Figure 002\_End users\_Flowchart\_NoteGPT**

flowchart

A{{"Use Cases for GeoViewer"}} --> n1("ViewSpatialLayers") & n2("AnalyzeGeospatialData") & n3("EnhanceVisualization")

R("Researcher") --> n1

R --> n2

P("Planner") --> n1

P --> n2

D("DecisionMaker") --> n2

D --> n3

A("Administrator") --> n1

A --> n3

Ac("Academician") --> n1

Ac --> n2

Pu("Public") --> n1

style A stroke:#000000,fill:#69C3D7

style n1 stroke:#000000,fill:#69D7A7

style n2 stroke:#000000,fill:#9498F8

style n3 stroke:#000000,fill:#F19A7B

style R fill:#D4EFDF ,stroke:#000000

style P stroke:#000000,fill:#BABCF1

style D stroke:#000000,fill:#F7E8CA

style A stroke:#000000,fill:#F5C9BA

style Ac stroke:#000000,fill:#EECF8F

style Pu stroke:#000000,fill:#F5C9BA

**Priority**:- **High Priority**

**Phase:-** **Phase 1** Subsumed

**Governance Need (Business Problem):-**

**Issue**:- Geoviewer requires spatial layers from other modules to be hosted on this module.

**Reframe :** Geoviewer necessitates that spatial layers from other modules be hosted within this module.

**Use case of Issue :**   
**Use Case:** Hosting Spatial Layers in Geoviewer

**Title:** Hosting Spatial Layers for Geoviewer Functionality

**Actors:**

Geoviewer System Administrator

Module Developers

Data Providers

End Users (Researchers, Planners, Decision Makers)

**Preconditions:**

Geoviewer is installed and operational.

Other modules that generate spatial layers are available and functional.

The necessary permissions and access rights are granted to the Geoviewer system.

**Main Flow:**

**Identify Required Spatial Layers:**

The Geoviewer System Administrator identifies the spatial layers needed from other modules for effective visualization and analysis.

**Coordinate with Module Developers:**

The administrator collaborates with module developers to ensure that the required spatial layers are compatible with Geoviewer.

**Host Spatial Layers:**

The identified spatial layers are hosted within the Geoviewer module, ensuring they are accessible for visualization and analysis.

**Verify Data Integrity:**

The administrator conducts checks to verify the integrity and accuracy of the hosted spatial layers.

**Enable User Access:**

End users (Researchers, Planners, Decision Makers) are granted access to the hosted spatial layers within Geoviewer.

**Utilize Spatial Layers:**

End users utilize the hosted spatial layers for various tasks, such as data visualization, impact analysis, and decision-making.

**Postconditions:**

Spatial layers from other modules are successfully hosted within Geoviewer.

End users can access and utilize the hosted spatial layers for their analysis and decision-making processes.

**Extensions:**

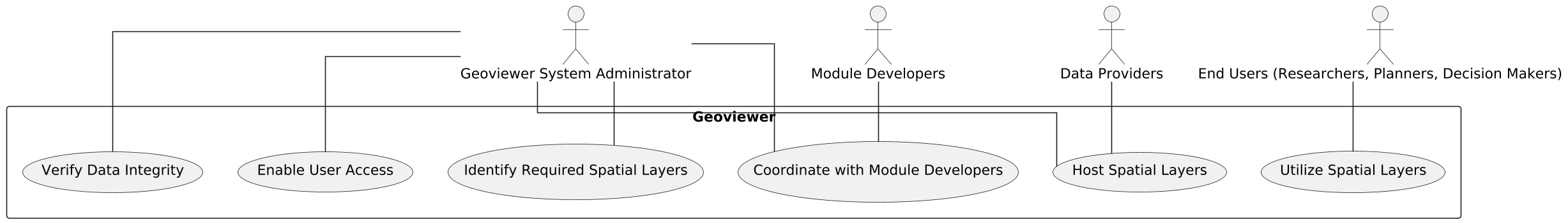
If new spatial layers are developed in other modules, the administrator can repeat the process to host these layers in Geoviewer.

If compatibility issues arise, the administrator may need to work with module developers to resolve them before hosting.

**Summary:**

The use case for hosting spatial layers in Geoviewer involves the Geoviewer System Administrator identifying necessary spatial layers from other functional modules, collaborating with module developers to ensure compatibility, and hosting these layers within Geoviewer for accessibility. The administrator verifies the integrity and accuracy of the hosted layers before granting access to end users, including researchers, planners, and decision makers, who utilize the layers for data visualization, impact analysis, and informed decision-making. The process can be repeated for new spatial layers, and any compatibility issues may require further collaboration with developers to resolve.

**Figure 003\_Issue\_Usecase\_PlantUML**



**Code for Issue\_Usecase\_PlantUML**

@startuml

skinparam linetype ortho

skinparam dpi 150

actor "Geoviewer System Administrator" as Admin

actor "Module Developers" as Devs

actor "Data Providers" as Providers

actor "End Users (Researchers, Planners, Decision Makers)" as Users

rectangle "Geoviewer" {

usecase "Identify Required Spatial Layers" as U1

usecase "Coordinate with Module Developers" as U2

usecase "Host Spatial Layers" as U3

usecase "Verify Data Integrity" as U4

usecase "Enable User Access" as U5

usecase "Utilize Spatial Layers" as U6

}

Admin -- U1

Admin -- U2

Admin -- U3

Admin -- U4

Admin -- U5

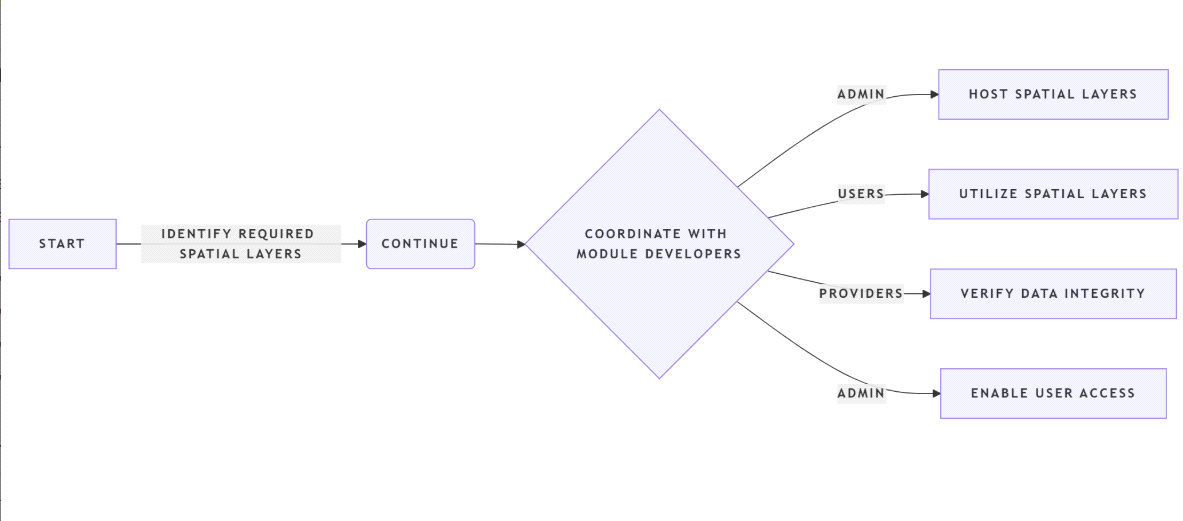
Users -- U6

Devs -- U2

Providers -- U3

@enduml

**Figure 003\_Issue\_flowchart\_NoteGPT**



**Code for Issue\_Flowchart\_NoteGPT**

flowchart LR

A[Start] --Identify Required Spatial Layers--> B(Continue)

B --> C{Coordinate with Module Developers}

C -- Admin --> D[Host Spatial Layers]

C -- Users --> E[Utilize Spatial Layers]

C -- Providers --> F[Verify Data Integrity]

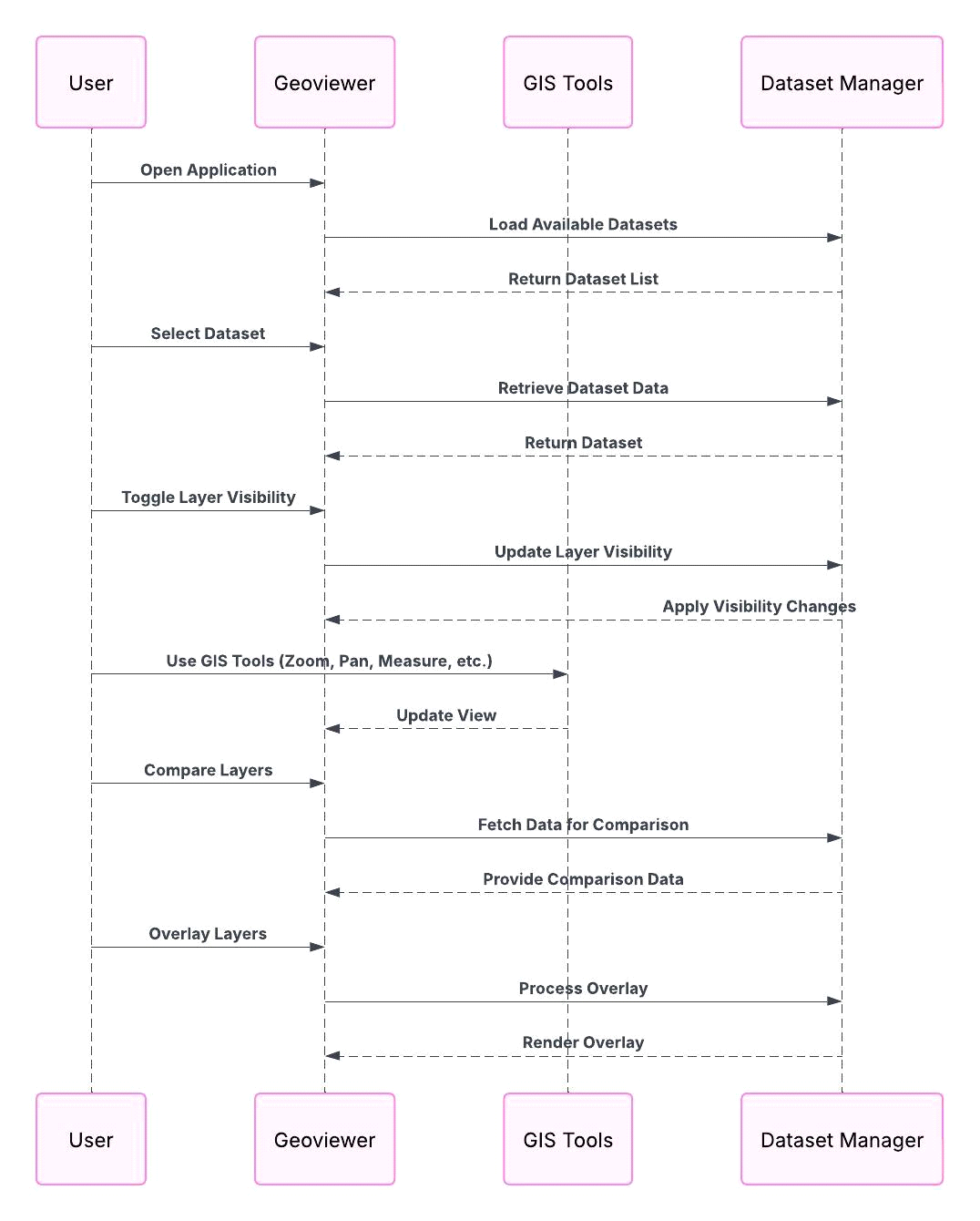
C -- Admin --> G[Enable User Access]

**Approach**:- Geoviewer from India WRIS will be subsumed with additions for new as well as updated geospatial layers of WARIMS modules.

**Output:-**

**Expected Outcome:-** Geoviewer: This module is an attempt to bring all the different sets of data on a single application for a comparative and interlinked view to derive a holistic picture with overlay. For assisting the same, many GIS based tools are provided for exploration of datasets. The user will be enabled with functionality to turn the visibility on/off for the different layers in the dataset to create user defined view.

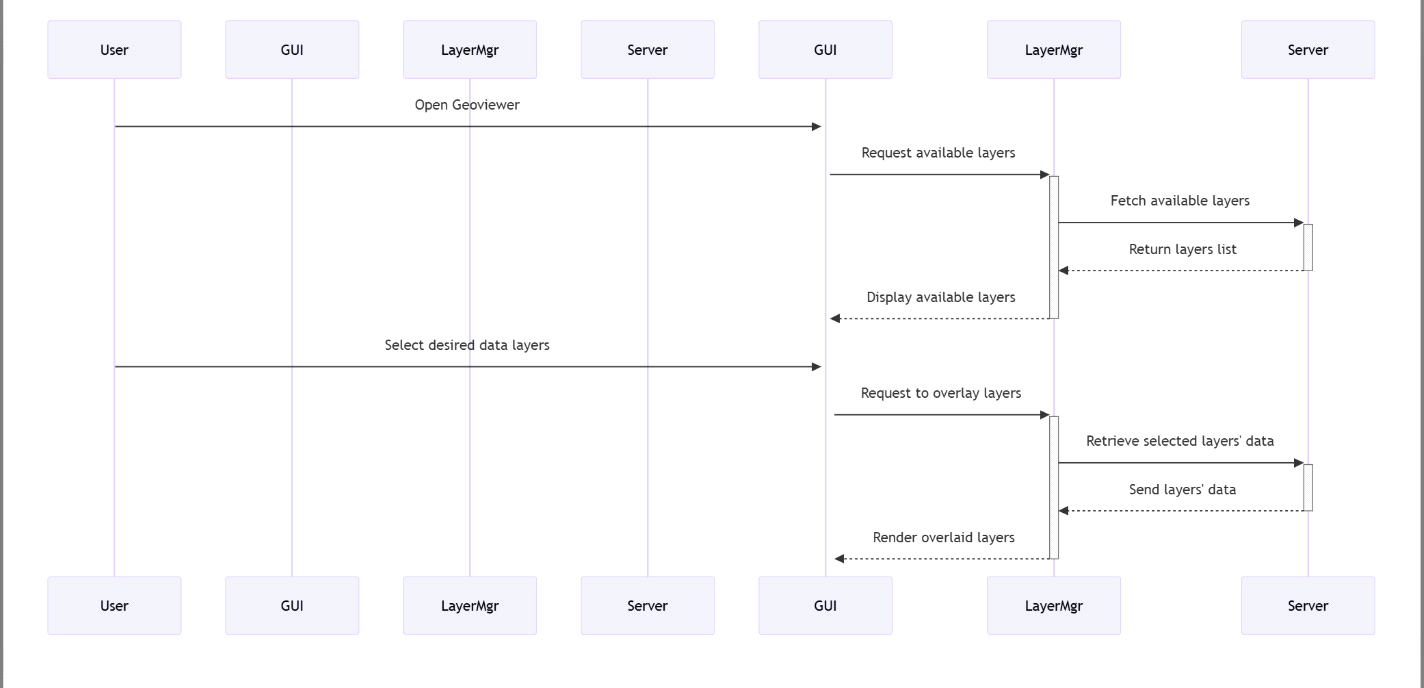
**Figure 004\_Expectedoutcome\_Sequencediagram\_Lucidchart**



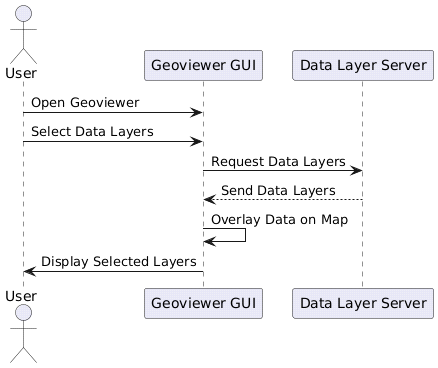
**Code for Expected outcome\_Flowchart\_NoteGP**

**Visualization:-** Geoviewer GUI can also be seen in the adjoining figure where various data layers can be overlaid as per users’ choice. (Example: Geoviewer, India WRIS)

**Figure 005\_Visualization\_Sequencediagram\_NoteGPT**



**Figure 005\_Visualization\_Sequencediagram\_PlantUML**



**Code for Visualization\_Sequencediagram\_PlantUML:**

@startuml

actor User

participant "Geoviewer GUI" as GUI

participant "Data Layer Server" as Server

User -> GUI: Open Geoviewer

User -> GUI: Select Data Layers

GUI -> Server: Request Data Layers

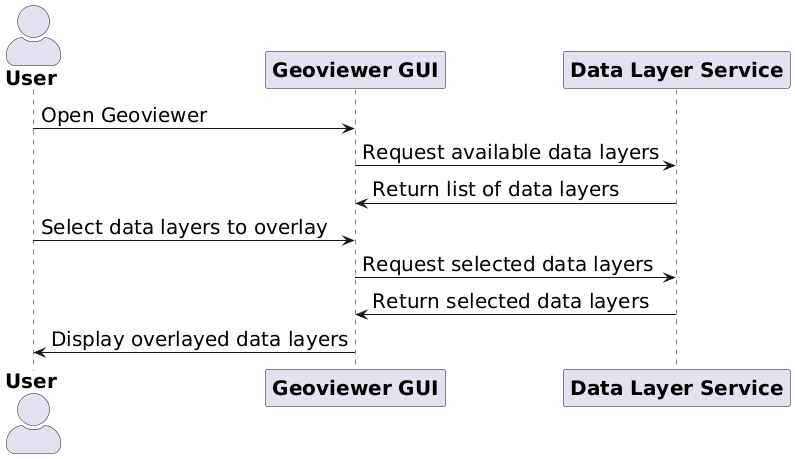
Server --> GUI: Send Data Layers

GUI -> GUI: Overlay Data on Map

GUI -> User: Display Selected Layers

@enduml

**Figure 005\_Visualization\_SequDiag\_PlantUML 2**



**Code For Figure 005\_Visualization\_SequDiag\_PlantUML 2**

@startuml

skinparam actorstyle awesome

skinparam defaultfontsize 20

actor User as "\*\*User\*\*"

participant "\*\*Geoviewer GUI\*\*" as GUI

participant "\*\*Data Layer Service\*\*" as DLS

User -> GUI: Open Geoviewer

GUI -> DLS: Request available data layers

DLS -> GUI: Return list of data layers

User -> GUI: Select data layers to overlay

GUI -> DLS: Request selected data layers

DLS -> GUI: Return selected data layers

GUI -> User: Display overlayed data layers

@enduml

**Frequency of Up-dation:-** Regular updations when any data layer from new module is added or any hosted geospatial layer is updated.

**Measure of Success:-** Hosted Data Layers are working and Web App is successfully running on the portal.

**Input Data Required:- Data Points:**

|  |  |
| --- | --- |
| **Data Point** | **Data Source** |
| Administrative Boundary Layers | NWIC |
| Hydrological Boundary Layers | NWIC |
| Socioeconomic Layers | Census of India |
| Wetland Layers | NRSC |
| Glacial Lakes | CWC |
| Ground Water Prospects Study | NRSC |
| Surface Water Bodies | NWIC |
| Land Degradation | NRSC |
| Forest/Tree Cover | NRSC/FSI |
| Land Use Land Cover | NRSC/NWIC |
| Soil | NBSS & LUP |
| Coastline | RMSI |
| Water logging/Soil Salinity | CWC |
| Water Resource Projects | CWC |
| Inland Navigation Waterways | IWAI |
| Flood Inundation | NRSC |
| Reservoir Survey | CWC |
| Agro climatic/Agro ecological Layers | NITI Ayog/ICAR |
| Infrastructure | Airport Authority of India, NHAI, NRDB, Indian Railways |
| Reported Extreme Temperature, Rainfall & Earthquake Events | IMD |

Abbreviations:

**CGWB:** Central Ground Water Board

**CWC:** Central Water Commission

**FSI:** Forest Survey of India

**IMD:** Indian Meteorological Organization

**IWAI:** Inland Waterways Authority of India

**NBSS&LUP:** National Bureau of Soil Survey & Land Use Planning

**NHAI:** National Highway Authority of India

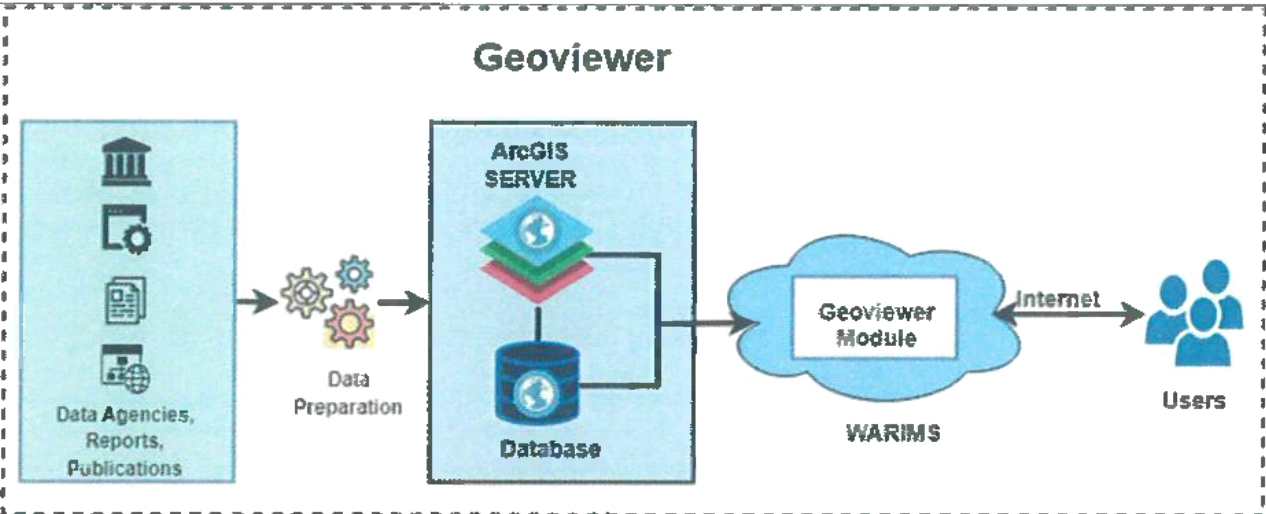
**NRDB:** National Road Database

**NWIC:** National Water Informatics Centre

**NRSC:** National Remote Sensing Centre

**Process:**

**Algorithm/Tools:-** Flowchart for various processes involved in creating Geoviewer are as follows:



Geospatial layers are generated from data received from data agencies in form of tables, pdfs, reports, shape files, geodatabases, etc. These layers are shared using ESRI QGIS server through Web app which will be hosted in the module. GIS based tools such as egend, Layer list, Base map, Print, Surface profile, Share, Add Data, Zoom in/Zoom Out, Previous/Next Zoom, Locate, Global Search, Attribute table, Select, Swipe, Measurement, Bookmark etc. will be provided. This will enable user for querying through data for desired results.

**Different data layers to be shared on Geoviewer include the following list:**

* Administrative Boundary Layers
* Hydrological Boundary Layers
* Socioeconomic Layers
* Wetland Layers
* Glacial Lakes
* Ground Water Prospects Study
* Surface Water Bodies
* Land Degradation
* Forest/Tree Cover
* Land Use Land Cover
* Soil
* Coastline
* Water logging/Soil Salinity
* Water Resource Projects
* Inland Navigation Waterways
* Flood Inundation
* Reservoir Survey
* Agro climatic/Agro ecological Layers
* Infrastructure
* Reported Extreme Temperature, Rainfall & Earthquake Events

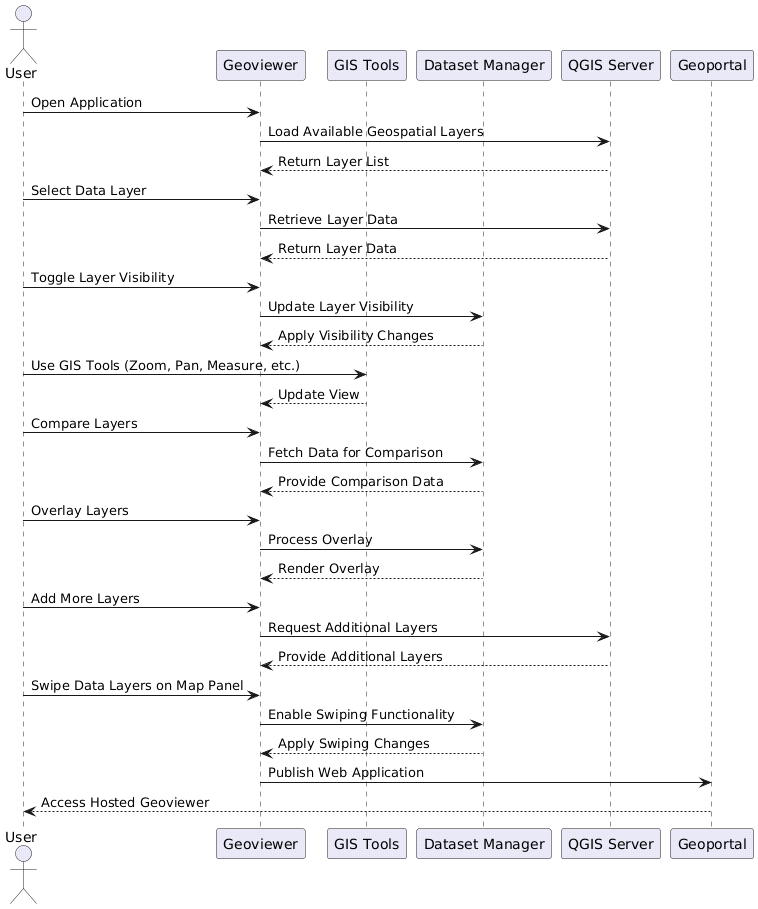
These GIS data layers will be published in QGIS Enterprise. A web-app named Geoviewer using published layers with GIS functionalities/tools for users will then be published in QGIS Enterprise. This app will then be hosted on the geoportal. Some of the GIS tools are shown.

Various Data layer list hosted on Geoviewer

Swiping various data layers on map panel in GeoViewer

Adding More layers on map panel in GeoViewer

**Figure 006\_ Algorithm/Tools \_Sequencediagram\_PlantUML**



**Code for Algorithm and tool:**

@startuml

actor User

participant Geoviewer

participant "GIS Tools" as GISTools

participant "Dataset Manager" as DatasetManager

participant "QGIS Server" as QGISServer

participant "Geoportal" as Geoportal

User -> Geoviewer: Open Application

Geoviewer -> QGISServer: Load Available Geospatial Layers

QGISServer --> Geoviewer: Return Layer List

User -> Geoviewer: Select Data Layer

Geoviewer -> QGISServer: Retrieve Layer Data

QGISServer --> Geoviewer: Return Layer Data

User -> Geoviewer: Toggle Layer Visibility

Geoviewer -> DatasetManager: Update Layer Visibility

DatasetManager --> Geoviewer: Apply Visibility Changes

User -> GISTools: Use GIS Tools (Zoom, Pan, Measure, etc.)

GISTools --> Geoviewer: Update View

User -> Geoviewer: Compare Layers

Geoviewer -> DatasetManager: Fetch Data for Comparison

DatasetManager --> Geoviewer: Provide Comparison Data

User -> Geoviewer: Overlay Layers

Geoviewer -> DatasetManager: Process Overlay

DatasetManager --> Geoviewer: Render Overlay

User -> Geoviewer: Add More Layers

Geoviewer -> QGISServer: Request Additional Layers

QGISServer --> Geoviewer: Provide Additional Layers

User -> Geoviewer: Swipe Data Layers on Map Panel

Geoviewer -> DatasetManager: Enable Swiping Functionality

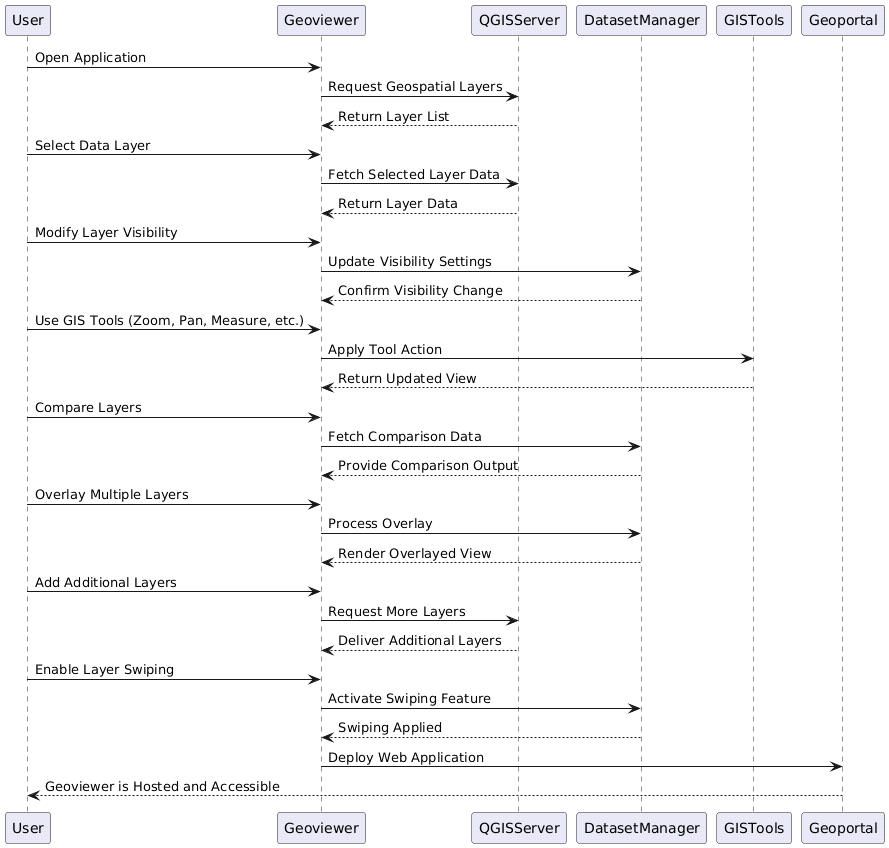
DatasetManager --> Geoviewer: Apply Swiping Changes

Geoviewer -> Geoportal: Publish Web Application

Geoportal --> User: Access Hosted Geoviewer

@enduml

**Figure 006\_ Algorithm/Tools \_Sequencediagram\_PlantUML\_img2**



**Code for Algorithm/Tools \_Sequencediagram\_PlantUML\_img2**

@startuml

User -> Geoviewer: Open Application

Geoviewer -> QGISServer: Request Geospatial Layers

QGISServer --> Geoviewer: Return Layer List

User -> Geoviewer: Select Data Layer

Geoviewer -> QGISServer: Fetch Selected Layer Data

QGISServer --> Geoviewer: Return Layer Data

User -> Geoviewer: Modify Layer Visibility

Geoviewer -> DatasetManager: Update Visibility Settings

DatasetManager --> Geoviewer: Confirm Visibility Change

User -> Geoviewer: Use GIS Tools (Zoom, Pan, Measure, etc.)

Geoviewer -> GISTools: Apply Tool Action

GISTools --> Geoviewer: Return Updated View

User -> Geoviewer: Compare Layers

Geoviewer -> DatasetManager: Fetch Comparison Data

DatasetManager --> Geoviewer: Provide Comparison Output

User -> Geoviewer: Overlay Multiple Layers

Geoviewer -> DatasetManager: Process Overlay

DatasetManager --> Geoviewer: Render Overlayed View

User -> Geoviewer: Add Additional Layers

Geoviewer -> QGISServer: Request More Layers

QGISServer --> Geoviewer: Deliver Additional Layers

User -> Geoviewer: Enable Layer Swiping

Geoviewer -> DatasetManager: Activate Swiping Feature

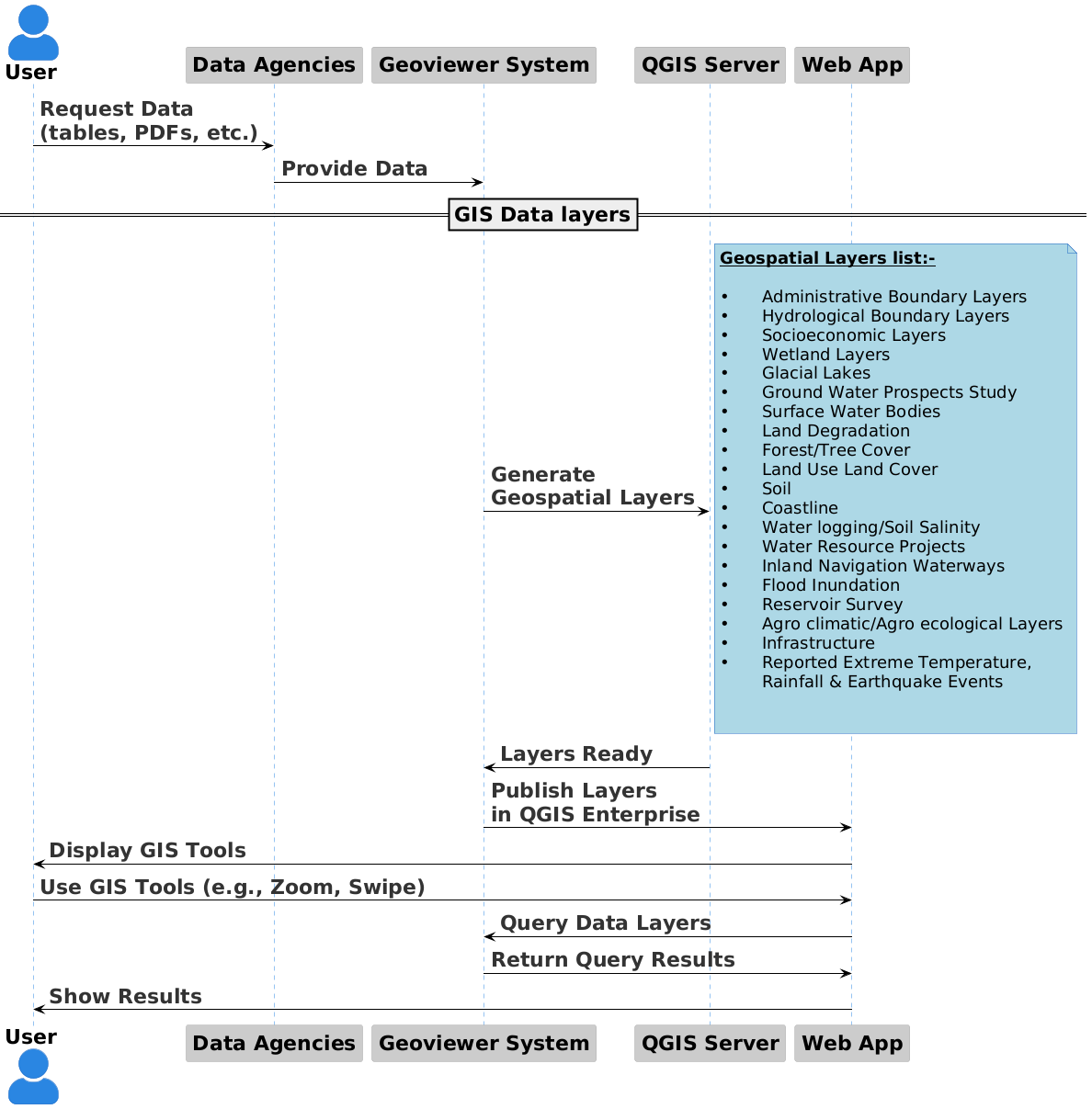
DatasetManager --> Geoviewer: Swiping Applied

Geoviewer -> Geoportal: Deploy Web Application

Geoportal --> User: Geoviewer is Hosted and Accessible

@enduml

**Figure 006\_ Algorithm/Tools \_SequDiag\_PlantUML 2**



**Code For Figure 006\_ Algorithm/Tools \_SequDiag\_PlantUML 2**

@startuml

**!theme reddress-lightblue**

**skinparam actorstyle awesome**

**skinparam defaultfontsize 22**

actor "\*\*User\*\*" as User

participant "\*\*Data Agencies\*\*" as DA

participant "\*\*Geoviewer System\*\*" as GVS

participant "\*\*QGIS Server\*\*" as QGIS

participant "\*\*Web App\*\*" as WebApp

User -> DA: **\*\***Request Data\*\* **\n**\*\*(tables, PDFs, etc.)\*\*

DA -> GVS: \*\*Provide Data\*\*

**==GIS Data layers==**

GVS -> QGIS: \*\*Generate\*\* \n\*\*Geospatial Layers\*\*

**skinparam notefontsize 18**

**note right** **#lightblue**

\*\*\_\_Geospatial Layers list:-\_\_\*\*

• Administrative Boundary Layers

• Hydrological Boundary Layers

• Socioeconomic Layers

• Wetland Layers

• Glacial Lakes

• Ground Water Prospects Study

• Surface Water Bodies

• Land Degradation

• Forest/Tree Cover

• Land Use Land Cover

• Soil

• Coastline

• Water logging/Soil Salinity

• Water Resource Projects

• Inland Navigation Waterways

• Flood Inundation

• Reservoir Survey

• Agro climatic/Agro ecological Layers

• Infrastructure

• Reported Extreme Temperature,

Rainfall & Earthquake Events

end note

QGIS -> GVS: \*\*Layers Ready\*\*

GVS -> WebApp: \*\*Publish Layers\*\* \n\*\*in QGIS Enterprise\*\*

WebApp -> User: \*\*Display GIS Tools\*\*

User -> WebApp: \*\*Use GIS Tools (e.g., Zoom, Swipe)\*\*

WebApp -> GVS: \*\*Query Data Layers\*\*

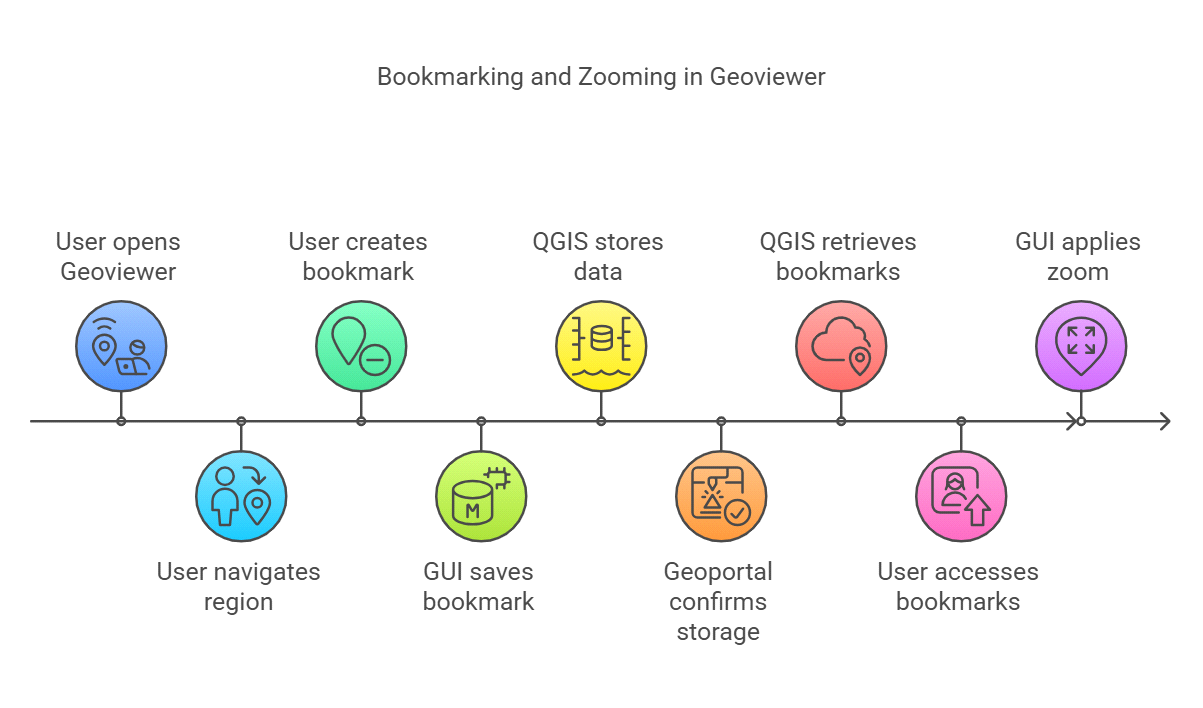
GVS -> WebApp: \*\*Return Query Results\*\*

WebApp -> User: \*\*Show Results\*\*

@enduml

**Bookmarks:** Regions at predefined Zoomed scales in Geoviewer (Can be created by users as per their requirements). Telangana & Uttar Pradesh bookmarks created by user below.

**Figure 007\_ Bookmarks \_Sequencediagram\_NoteGPT**



**Code for Bookmark:**

@startuml

skinparam linetype ortho

skinparam dpi 150

actor User

participant "Geoviewer Web App" as GUI

participant "QGIS Server" as QGIS

participant "Geoportal" as Geoportal

User -> GUI: Open Geoviewer

User -> GUI: Navigate to desired region

User -> GUI: Create Bookmark (e.g., Telangana, Uttar Pradesh)

GUI -> QGIS: Save Bookmark with predefined zoom scale

QGIS -> Geoportal: Store Bookmark Data

Geoportal --> QGIS: Confirmation of storage

QGIS --> GUI: Bookmark saved successfully

User -> GUI: Access saved Bookmarks

GUI -> QGIS: Retrieve stored Bookmarks

QGIS -> Geoportal: Fetch Bookmark Data

Geoportal --> QGIS: Return Bookmark Data

QGIS --> GUI: Display available Bookmarks

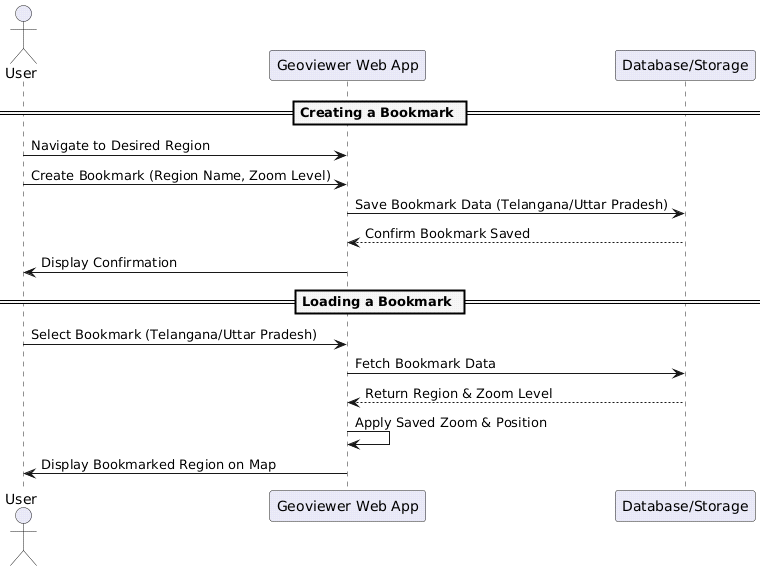
User -> GUI: Select Bookmark (e.g., Telangana)

GUI -> QGIS: Apply predefined zoom scale

QGIS --> GUI: Zoom to predefined region

@enduml

**Figure 007\_ Bookmarks \_Sequencediagram\_PlantUML**



**Code for Bookmarks \_Sequencediagram\_PlantUML :**

@startuml

actor User

participant "Geoviewer Web App" as WebApp

participant "Database/Storage" as DB

== Creating a Bookmark ==

User -> WebApp: Navigate to Desired Region

User -> WebApp: Create Bookmark (Region Name, Zoom Level)

WebApp -> DB: Save Bookmark Data (Telangana/Uttar Pradesh)

DB --> WebApp: Confirm Bookmark Saved

WebApp -> User: Display Confirmation

== Loading a Bookmark ==

User -> WebApp: Select Bookmark (Telangana/Uttar Pradesh)

WebApp -> DB: Fetch Bookmark Data

DB --> WebApp: Return Region & Zoom Level

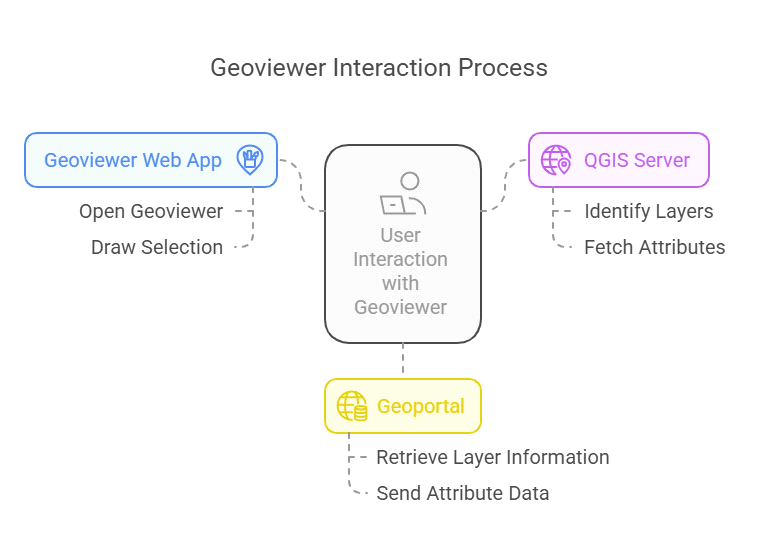
WebApp -> WebApp: Apply Saved Zoom & Position

WebApp -> User: Display Bookmarked Region on Map

@enduml

**Select Feature:** Upon drawing a rectangular selection in the Map panel, all selectable layers will be shown on a right-side window and attributes of selected features of different layers can be seen in the attribute table below.

**Figure 008\_ Select Feature \_Sequencediagram\_NoteGPT**



**Code for Select Feature :**

@startuml

skinparam linetype ortho

skinparam dpi 150

actor User

participant "Geoviewer Web App" as GUI

participant "QGIS Server" as QGIS

participant "Geoportal" as Geoportal

User -> GUI: Open Geoviewer

User -> GUI: Draw rectangular selection on Map panel

GUI -> QGIS: Identify selectable layers within selection area

QGIS -> Geoportal: Retrieve layer information

Geoportal --> QGIS: Return layer data

QGIS --> GUI: Display selectable layers in right-side window

User -> GUI: Select a specific layer

GUI -> QGIS: Fetch attributes of selected features

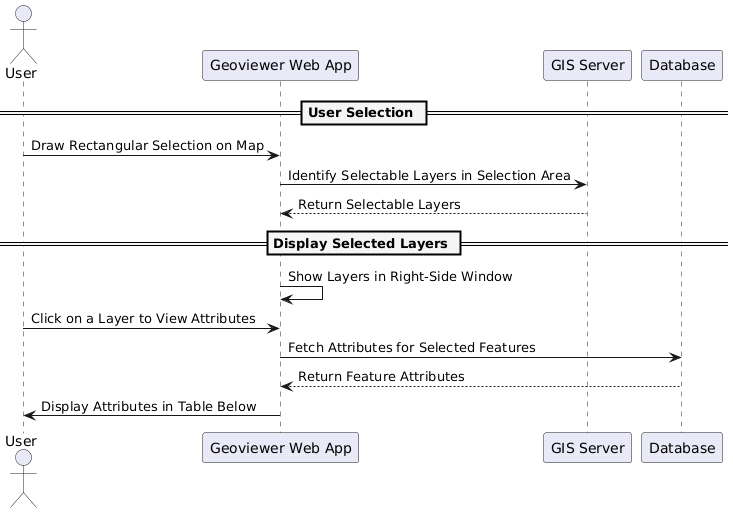
QGIS -> Geoportal: Retrieve feature attributes

Geoportal --> QGIS: Send attribute data

QGIS --> GUI: Display attributes in attribute table below

@enduml

**Figure 008\_ Select Feature \_Sequencediagram\_PlantUML**



**Code for Select Feature \_Sequencediagram\_PlantUML:**

@startuml

actor User

participant "Geoviewer Web App" as WebApp

participant "GIS Server" as Server

participant "Database" as DB

== User Selection ==

User -> WebApp: Draw Rectangular Selection on Map

WebApp -> Server: Identify Selectable Layers in Selection Area

Server --> WebApp: Return Selectable Layers

== Display Selected Layers ==

WebApp -> WebApp: Show Layers in Right-Side Window

User -> WebApp: Click on a Layer to View Attributes

WebApp -> DB: Fetch Attributes for Selected Features

DB --> WebApp: Return Feature Attributes

WebApp -> User: Display Attributes in Table Below

@enduml

**Data Validation:-** Geospatial layers need to be checked for working properly before hosting on Geoviewer.

**Software Technologies:-**  QGIS Enterprise

**Dependencies & Risks:** Ability to host data layers, QGIS Server

**User Acceptance Testing (UAT):-** NWIC

**Development Responsibility:** NWIC

**References :-**

<https://indiawris.gov.infwris/#/Geoviewer>

https.//indiawris.gov.in/downloads/Functional Requirement Specification1.pdf

<https://indiawris.gov.in/downloads/Data> Assessment Reports.pdf

**---End of Document---**