CEREO Living Atlas

By the CEREO Living Atlas Development Team:
Zachary Garoutte, Yaru Gao, Jonathan Simmons

Sponsor: Center for Environmental Research, Education, and Outreach

center for environmental research, education & outreach
CEREO

Project Overview and Requirements

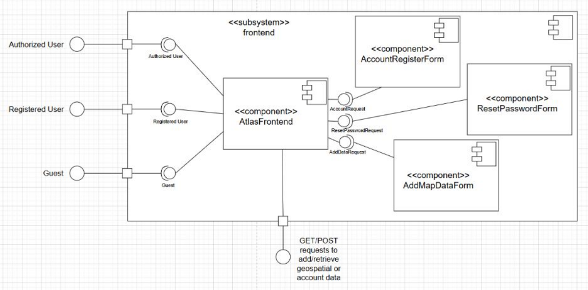
- Environmental datasets are sets of data that track and analyze the world around us. This could include tracking water quality, monitoring restoration work, or detailing the spread of invasive species.
- Organizations like researchers, tribal groups, educators, students, and government agencies, all use these environmental datasets to locate patterns, highlight case studies, and make clear decisions on future environmental work.
- However, these environmental datasets are often fragmented among separate systems making data difficult to locate, and some of these systems may be difficult to use

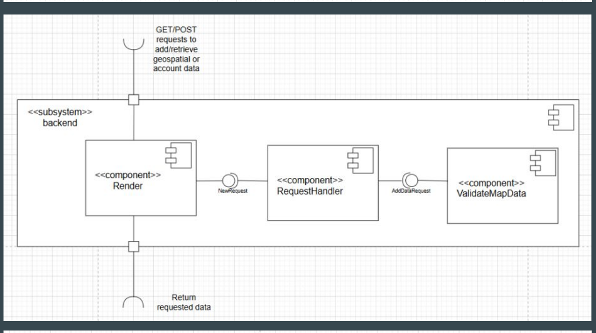
Project Overview and Requirements

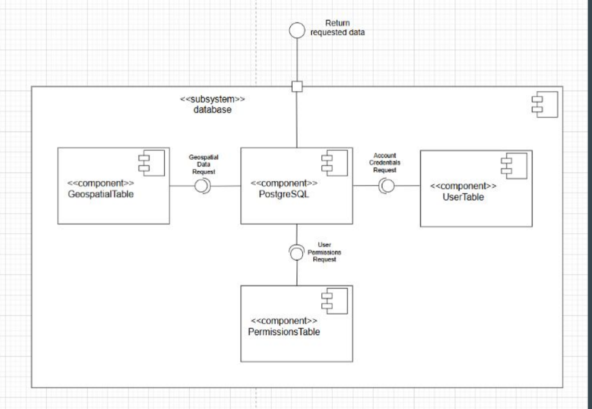
- The CEREO Living Atlas aims to solve this problem by creating a single, map-focused interface where users can upload and view environmental datasets
- The Living Atlas had already been built by previous development team, but there were still several problems with the Atlas that CEREO wanted solved to reach the original vision for the application. These include:
  - Limited features with uploading data: Datasets had to be linked as an external URL instead of a file attachment, and all data cards had a default thumbnail
  - Lack of options for searching, sorting, and filtering through the available data, as well as not being able to bookmark datasets
  - Lack of a 2-factor authentication system for users to access their accounts
  - Technical issues with accessing data including slow loading times

Project Solution Approach

- The system is a web framework that is structured into three main subsystems: Frontend, backend, and database
- The frontend uses React.js hosted on Netlify, which includes a user-friendly interface and an interactive map powered by Mapbox
  - We chose React for its ease of use allowing for faster development
- The backend uses FastAPI hosted on Render as an intermediary that handles API requests, authentication, data processing, and any other communication between the frontend and database
  - We chose FastAPI for its high performance and potential scalability to handle a greater number of requests
- The database uses PostgreSQL hosted on Microsoft Azure to store the application's geospatial data, user credentials, and metadata.
- What are the tools, frameworks, platforms, libraries, etc.?
  - We also chose PostgreSQL for its high performance and potential scalability







Project's Current Status

- Added external ArcGIS layers from WA Dept. of Ecology and other organizations that can be loaded and overlaid over the map
- User accounts have a password reset option for account recovery via an email verification link
- Data cards can now have file attachments that can be downloaded by other users
- Data cards can now have custom thumbnails
- Added ability for users to bookmark cards to be easily viewed later
- Data cards can be sorted by location or upload date and filtered by tag for easier searching
- Updated card upload feature allows users to select a location for their data directly on the map rather than manually entering coordinates
- Expanded on the layers tab by allowing hydrological and city boundaries that are shown by default to be toggled on and off
- Clicking on pins to zoom in to a pin's location
- UI refinement to reduce redundancy, promote user-friendliness, and scale to different devices
- Fixing bugs where data would not load

Project's Future Work

- Performance could be improved upon, as data can sometimes be slow to load into the application. This may be fixed through a computational upgrade to the backend.
- Encryption of user data could be added to protect against cyberattacks.
- Adding additional types of ArcGIS data such as FeatureServer would allow for more interactivity with the external GIS layers.
- An in-app tutorial could be added that would guide new users on how to use the more complex features of the application, such as the external GIS layers.
- Accessibility features could be added so that the application complies with the university's accessibility policy.

Thank You