

EnvoScholar v2.0

Student: Carlos Bravo

Mentor: Dr. Mark Finlayson,

Instructor: Dr. Masoud Sadjadi, Florida International University

Problem

Center for Aquatic Chemistry and Environment(CREST): a research center at FIU is in need for a Web-based, user-friendly front-end solution to browse through the scientific articles stored on their databases and be able to find useful literature for future and ongoing research.

There isn't a search engine dedicated to environmental science papers only.

Current System

The current system allows users to:

- Enter a search query which has autocomplete feature.
- Click on an article to view more information .
- Highlight sentences containing key concepts on the abstract.
- View the ontology in Node Graph and Tree view format.
- Go to the articles Science Direct page.
- Leave feedback on user experience and Rate articles.
- Create an account.
- Save articles.
- View the history of
 - Search queries they have entered.
 - Articles they have rated
 - Articles they have clicked on.
 - Articles they have saved.

Requirements

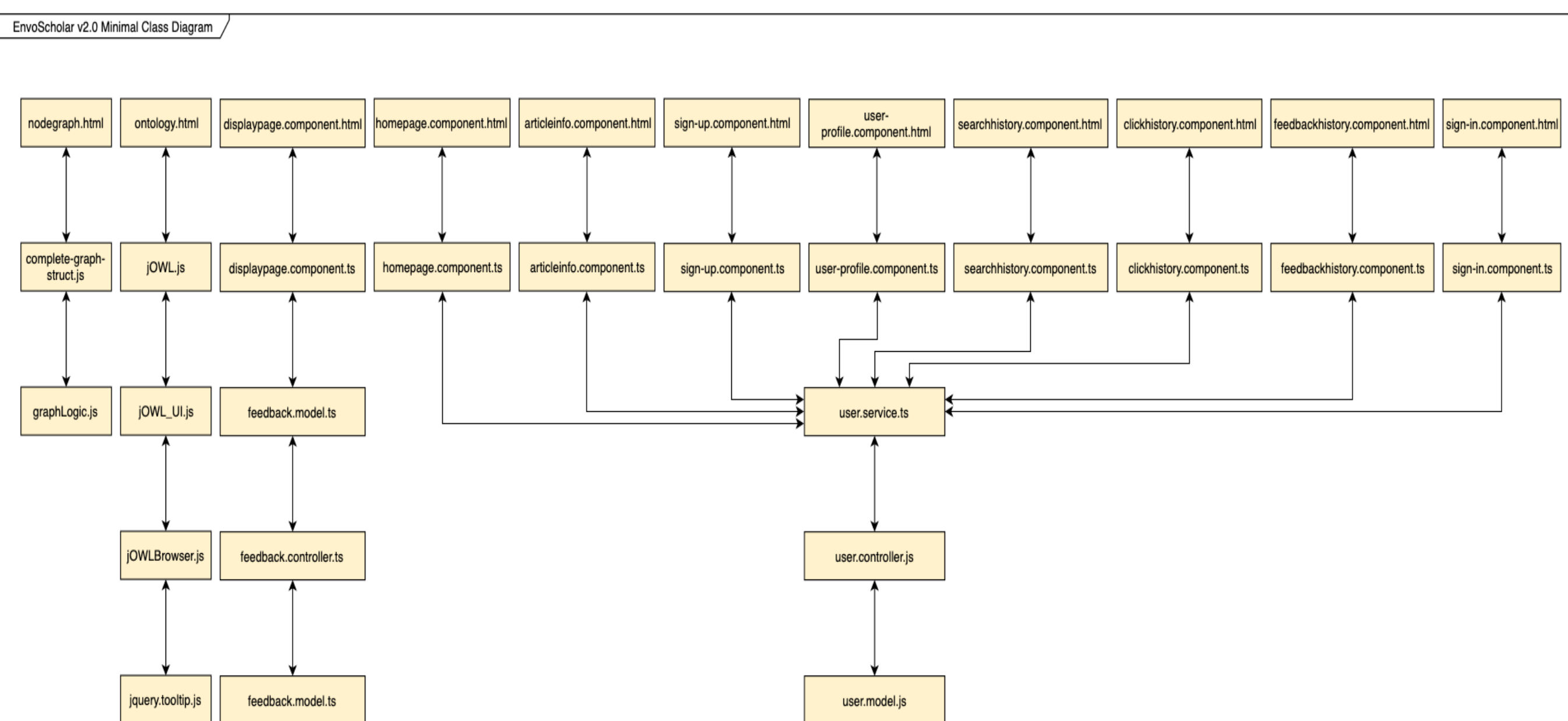
EnvoScholar v2.0 should be web-based, cross-browser and interactive.

For my contributions, a user should be:

- Able browse the ontology through a Node graph visualization.
- Able to rate scientific articles.
- Able to see feedback history.
- Search concepts and visualize them in Graph view and tree view simultaneously.

Object Design

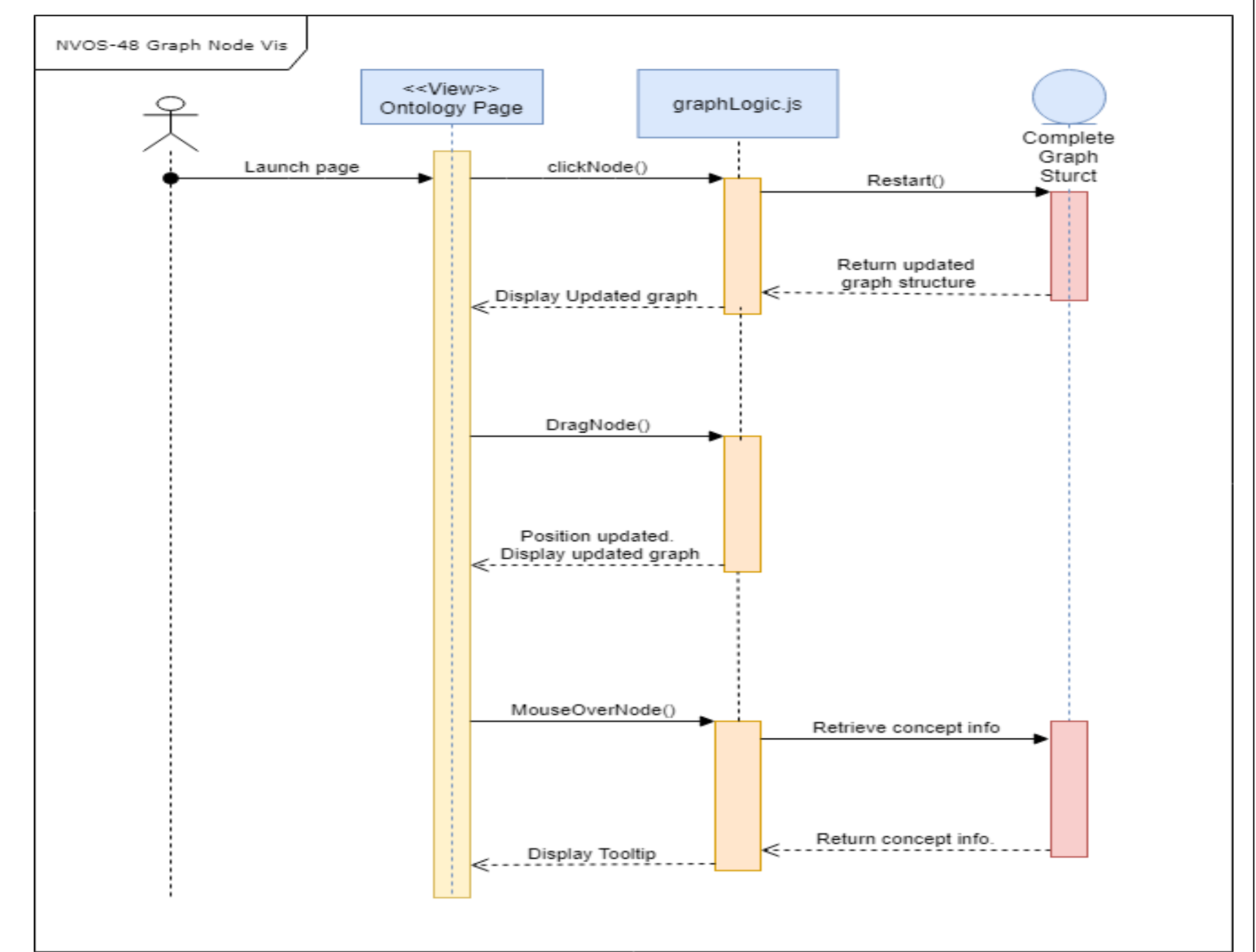
Minimal class diagram: Entire system



Use Case Diagram: Node graph visualization

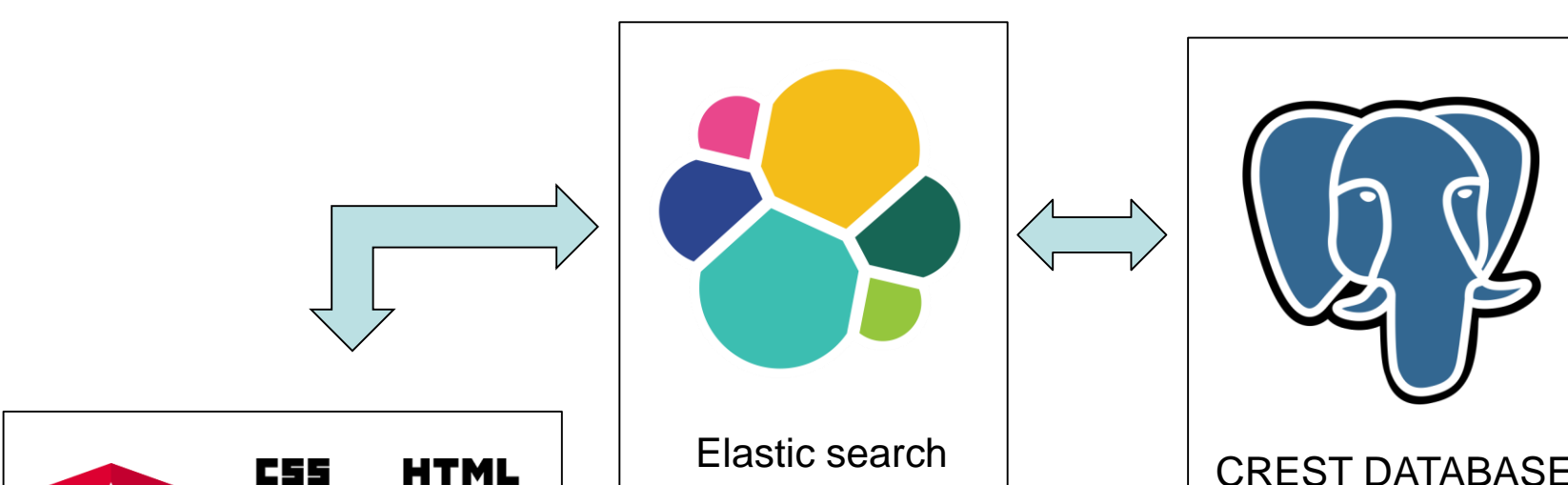


Sequence Diagram: Node graph view



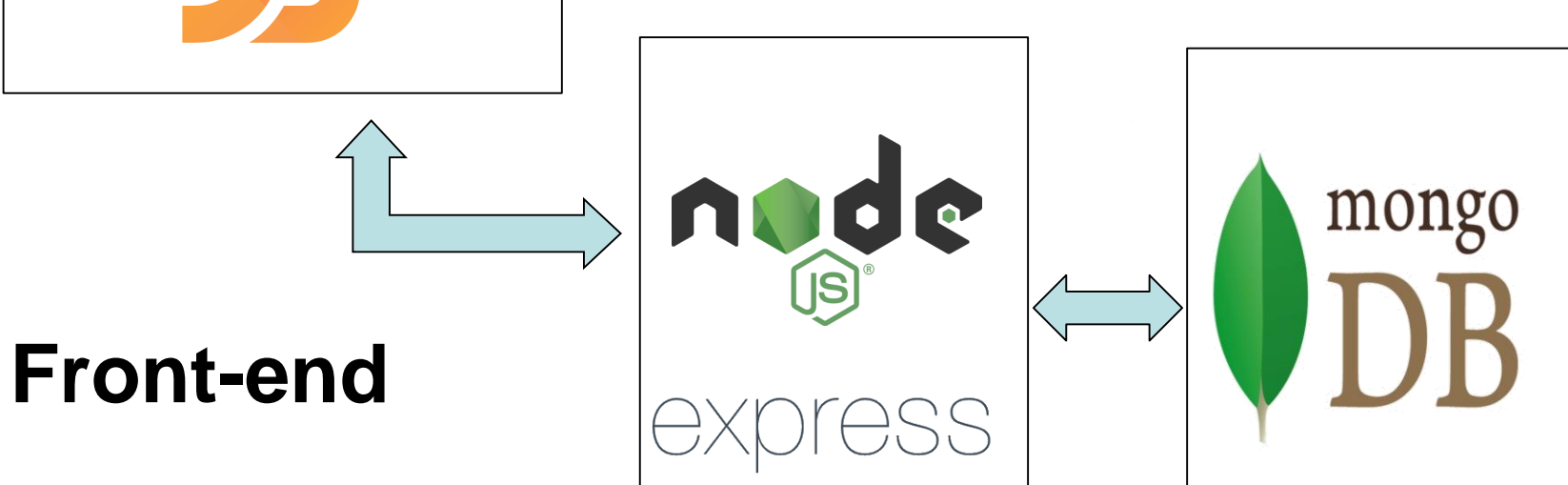
System Design

Article Information

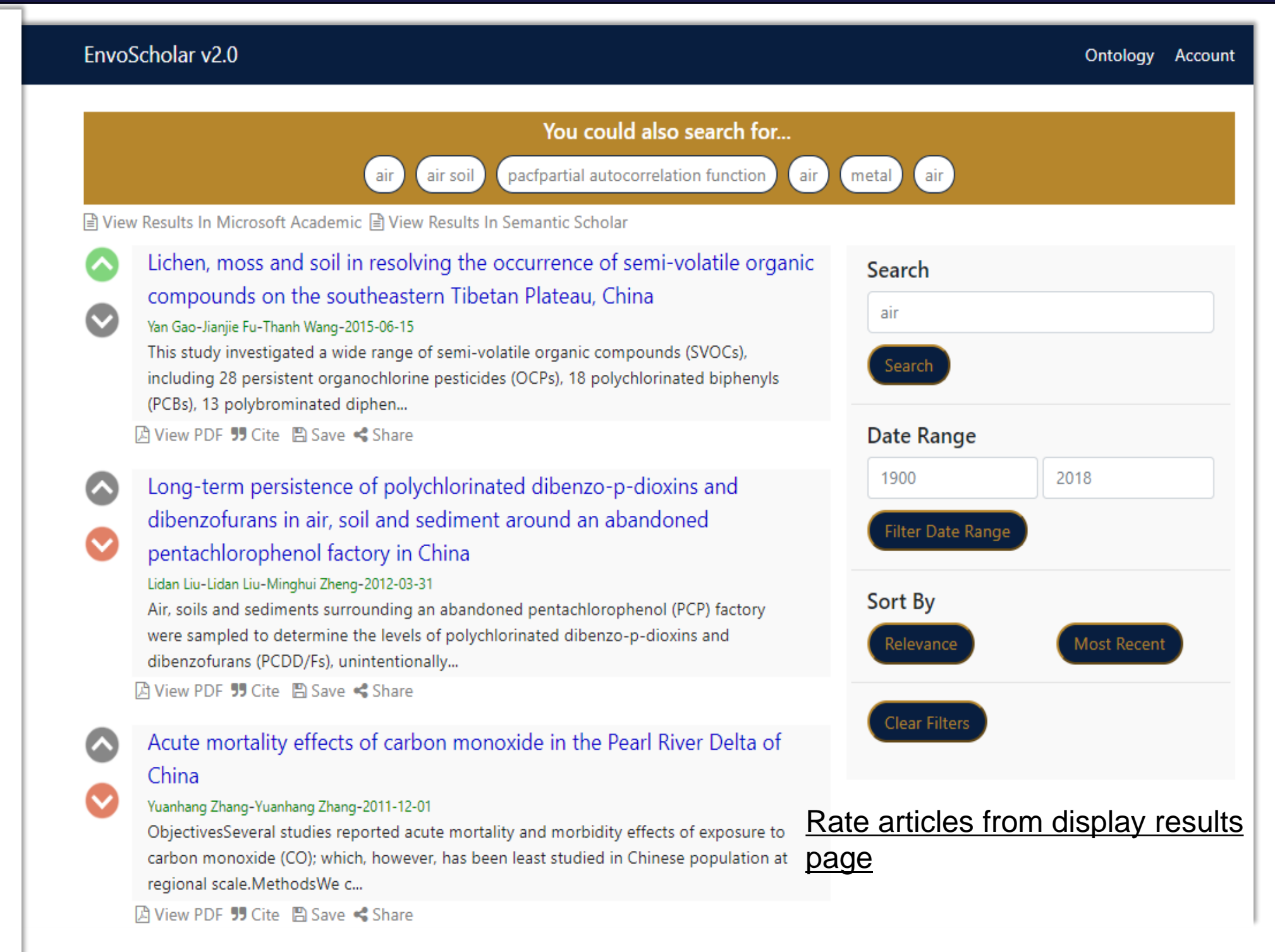
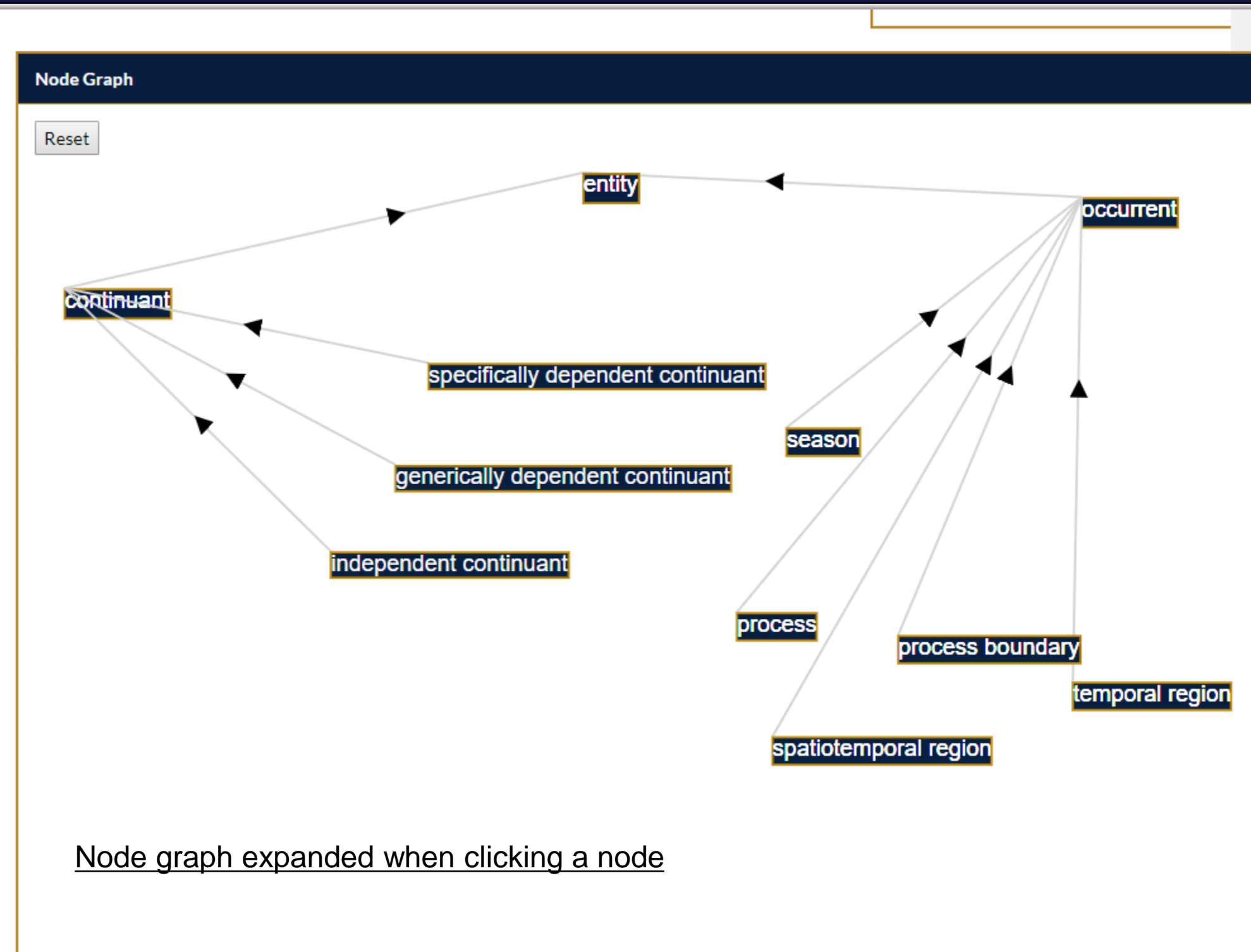


Back-end

User Information

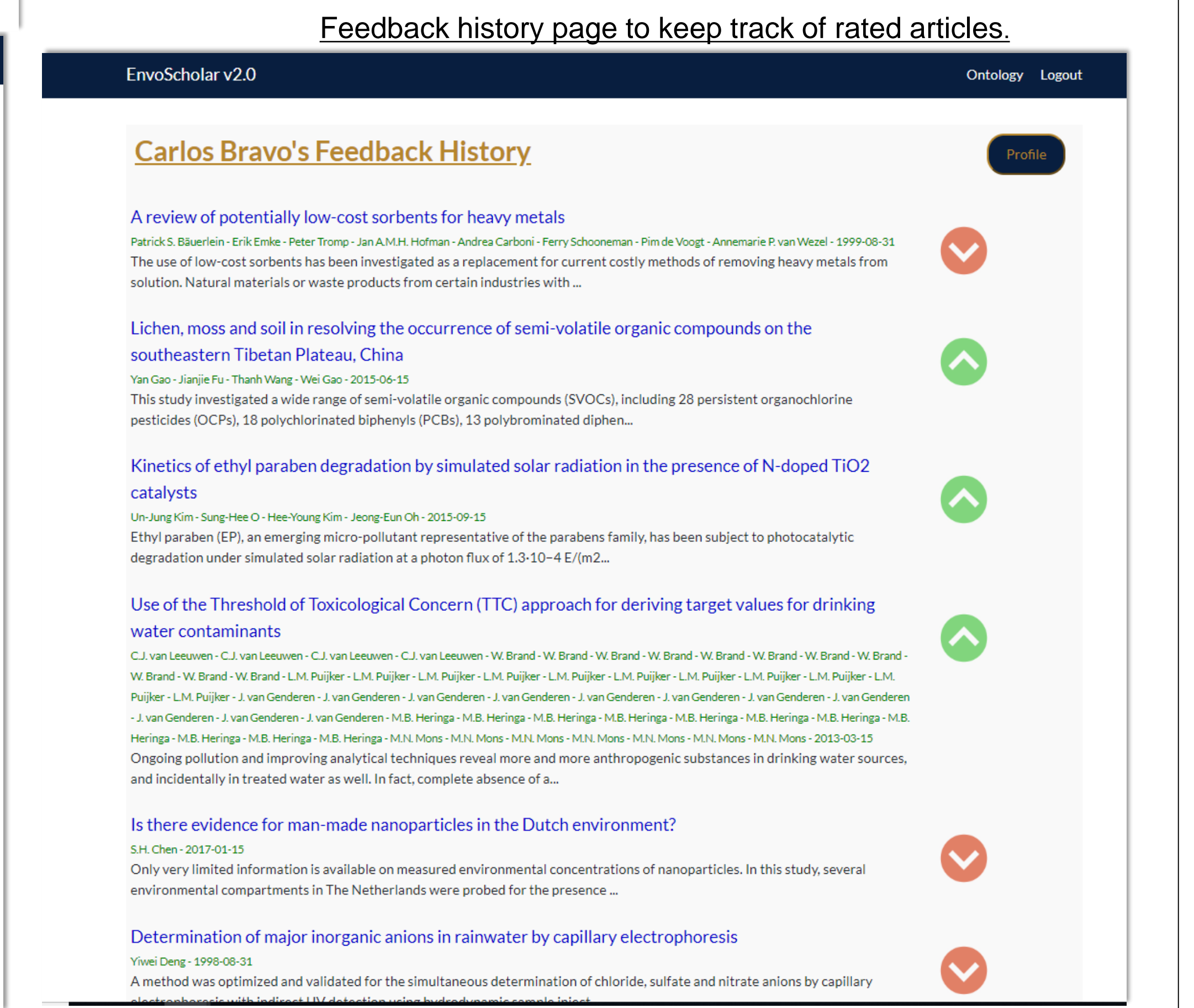
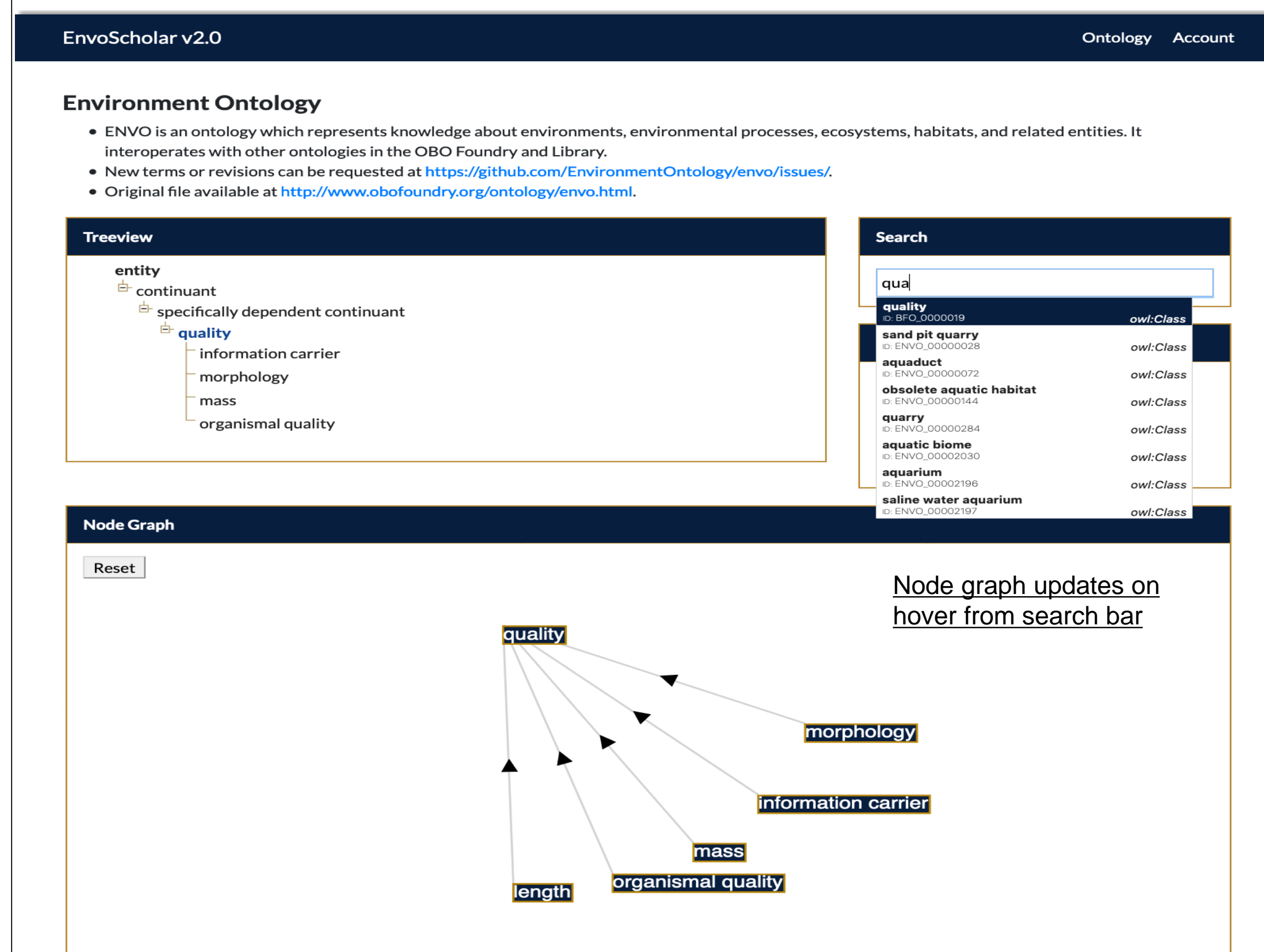


Screenshots



Implementation

- Angular 6 was used to create the functionality of the front end.
- HTML5, CSS were used for styling.
- D3.js was used to create Node graph visualization of the ontology.
- NodeJS and Express were used to create the server that connects to the Mongo Database containing the users and feedback information.
- Elasticsearch was used to enhance the GET requests to the PostgreSQL database containing the articles.
- JWT and Passport were used for user authentication



Acknowledgement

The material presented in this poster is based upon the work supported by Dr. Masoud Sadjadi and Dr. Mark Finlayson. I thank Sheila Alemany, Maria Presa, and Deya Banisakher for assistance, cooperation and mentorship that I received throughout this process