

CIS 4911

SENIOR PROJECT: ENVOSCHOLAR v3.0

Final Deliverable

Team Members:

Dairon HERNANDEZ

Jorge EUCEDA

Product Owner:

Mark FINLAYSON

Instructor:

Masoud Sadjadi

Mentors:

Maria A. PRESA REYES,

Deya BANISAKHER



The MIT License (MIT)

Copyright (c) 2019 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (EnvoSCholar), to deal in EnvoSCholar without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of EnvoSCholar, and to permit persons to whom EnvoSCholar is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of EnvoSCholar. ENVOSCHOLAR IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH ENVOSCHOLAR OR THE USE OF OTHER DEALINGS IN ENVOSCHOLAR.

1 Abstract

This document presents the information required for a thorough understanding of the academic search engine of EnvosScholar. This academic search engine is specifically for researchers in the field of environmental science to search through, and save, and provide feedback for environmental peer reviewed articles. Users are able to create profiles so they can save articles and gain access to view their saved articles, along with their search history and click history. Users can leave feedback on the overall user experience and can also browse the ENVO ontology using two visualization techniques, a treeview and node graph structure. This is version 3.0 of EnvosScholar, therefore, this document will specifically talk about features that were added to this web application. This document will contain the user stories that were completed, the user stories that were not completed (if any), the project plan, the system design, and the system validation. With the information provided in the document, one will be able to understand how EnvosScholar works with its various components and gain an idea of future plans with EnvosScholar.

Contents

1	Abstract	2
2	Introduction	5
2.1	Current System	5
2.2	Purpose of New System - EnvoScholar v3.0	6
3	User Stories	7
3.1	Implemented User Stories	7
3.1.1	User Story 68 - Website Deployment	7
3.1.2	User Story 190 - Server Side Filtering	7
3.1.3	User Story 189 - Server Side Loading and Pagination	8
3.1.4	User Story 57 - Developer Features	9
3.1.5	User Story 192 - Exportable Developer Statistics	9
3.1.6	User Story 191 - User Profile Page Enhancement	10
3.1.7	User Story 186 - User Mobile Experience	11
3.1.8	User Story 188 - Developer Mobile Experience	11
3.1.9	User Story 185 - Dynamic Ontology Visualization	12
3.1.10	User Story 187 - Embedded New Ontology Visualization	13
3.2	User Stories Pending for Next Team	13
4	Project Plan	14
4.1	Hardware and Software Resources	14
4.2	Sprints Plan	16
4.2.1	Sprint 1	16
4.2.2	Sprint 2	17
4.2.3	Sprint 3	18
4.2.4	Sprint 4	19
4.2.5	Sprint 5	20
4.2.6	Sprint 6	21
5	System Design	22
5.1	Architectural Patterns	22
5.2	System and Subsystem Decomposition	22
5.3	Data Subsystem	22
6	Appendix	24
6.1	Appendix A - UML Diagrams	24
6.1.1	Class Diagrams	24
6.1.2	Use Case Diagrams	30

6.1.3	Sequence Diagrams	36
6.2	Appendix B - User Interface Design	42
6.3	Appendix C - Sprint Review Reports	46
6.3.1	Sprint 1	46
6.3.2	Sprint 2	46
6.3.3	Sprint 3	46
6.3.4	Sprint 4	47
6.3.5	Sprint 5	47
6.3.6	Sprint 6	48
6.4	Appendix D - User Manuals, Installation/Maintenance Document, and Shortcomings/Wishlist Document, and Other Documents	48

2 Introduction

Environmental scientists conduct research in their fields using tools such as academic search engines such as Google Scholar or Academic.edu. They rely on these search engines to help them find the cutting edge work in their field, however, the existing systems are not dedicated to environmental science papers. They contain peer-reviewed articles in all research fields from medicine to computer science, and more. Users are able to create profiles so they can save articles and gain access to view their saved articles, along with their search history and click history. Users can leave feedback on the overall user experience and can also browse the ENVO ontology using two visualization techniques, a treeview and node graph structure. This is the third version of EnvoScholar, therefore, this section will briefly mention the first version and then will elaborate specifically about features that were added to this web application. With the information provided in the document, one will be able to understand how EnvoScholar works with its various components and gain an idea of future plans with EnvoScholar.

2.1 Current System

EnvoScholar was built using Angular 6 for the front end, NodeJS and Express as a server for the MongoDB which contains the user profiles and feedback space, and PostgreSQL which contains the articles. The header bar on each page contains a link called Ontology to a page which has a visualization using a treeview and node graph view from an external widget which connects to the environment ontology.

The current system allows any user, whether signed in or not, to enter a search. When a search is submitted, the user will be redirected to a page, which displays articles related to the search entry. On that page, the user is able to do a variety of actions. They will be listed below as follows:

- The user can click on any article title to be redirected to a page with more information on the article page that contains the PDF of the article.
- For each article, the user can click on View PDF to open a new tab with the Science Direct.
- For each article and if a user is logged into their account, the user can click on Save to save that article to their profile.
- For each article, the user can click on Share to share the article across social media

2.2 Purpose of New System - EnvoScholar v3.0

EnvoScholar v3.0 is an academic search engine in the field of environmental science which mainly contains newly following added features:

- Dynamic and User-Friendly Ontology Visualization that is Searchable and Expandable.
- Improved Mobile experience in major components.
- Developer Statistics that leverage user-based results.
- Overhauled User Profile capabilities.
- Complete server-side processing and filtering of search results.

If you would like to see details regarding these added features, and additional features that were also implemented, you can check Section 3, which has the details regarding to these user stories.

3 User Stories

3.1 Implemented User Stories

3.1.1 User Story 68 - Website Deployment

- **Description:** As a user, I would like to be able to access this website from the comfort of my own home, without having to depend on actively running the application on my machine. This is greatly important because it allow for greater accessibility for our researchers as well as our users.
- **Acceptance Criteria:**
 - The optimized server-side pagination enables users to quickly view results.
 - There is an indexing mechanism implemented that allows users to go to a specific page, as well as previous or next pages.
 - The aforementioned indexing mechanism also keeps track of the state of the current results, while also keeping track of the total amount of articles available.
 - The pagination processing is entirely handled by the MongoDB Server.
- **Use Case:**
 - **Name:** User Story 68
 - **Actor:** User
 - **Description <Flow of events>**
 - * The website is deployed into our custom VM provided by the product owner.
 - * The website can be seen in the following URL: envoscholar.cs.fiu.edu.
 - * The website is packaged and deployed in an optimized manner for a production environment, this requires a process to minify the HTML and JS code provided to the user.

3.1.2 User Story 190 - Server Side Filtering

- **Description:** As a user I would like the filtering to occur server side so that I can have access to more result while also having filtering occurs faster.
- **Acceptance Criteria:**
 - The user can have access to all results which can be returned by the api

- The user is able to quickly access pages without requiring a full api call of 100 results
- The website and filtering is now accessible off campus
- **Use Case:**
 - **Name:** User Story 190
 - **Actor:** User (Developer Type)
 - **Description <Flow of events>**
 - * The user enters a search query and is quickly presented with all of the available results for that query.
 - * The user is able to apply any amount of filters without causing lag on the site.

3.1.3 User Story 189 - Server Side Loading and Pagination

- **Description:** As a product owner, I would like an implementation of pagination that is done in the server, mainly so that users do not have to process the whole set of results by themselves.
- **Acceptance Criteria:**
 - The optimized server-side pagination enables users to quickly view results.
 - There is an indexing mechanism implemented that allows users to go to a specific page, as well as previous or next pages.
 - The aforementioned indexing mechanism also keeps track of the state of the current results, while also keeping track of the total amount of articles available.
 - The pagination processing is entirely handled by the MongoDB Server.
- **Use Case:**
 - **Name:** User Story 189
 - **Actor:** User
 - **Description <Flow of events>**
 - * User is able to go to the next or previous page with relatively fast response so as to not decrease user retention.
 - * User is able to accurately click on an article that has been paginated, and the corresponding page relating to that article is shown.

- * User is able to access more than 10 pages for results that have a vast amount of page results.

3.1.4 User Story 57 - Developer Features

- **Description:** As a user I would like there to be a “developer” account type which has access to unique developer stats and functionality that standard user account cannot access.
- **Acceptance Criteria:**
 - The user should be able to select one of the developer options and be redirected to a page containing a table populated with the developer stats data.
 - All data is aptly displayed and labeled in the developer tables.
 - Developers can make new developers by entering a user email in the developer page.
 - There is a graph representation of some of the data in the developer stats main page.
- **Use Case:**
 - **Name:** User Story 57
 - **Actor:** User
 - **Description <Flow of events>**
 - * User is able to see a list of options for different user activity data.
 - * User can click the buttons to access the tables for the developer data.
 - * User has the option to create a new developer by entering an email.

3.1.5 User Story 192 - Exportable Developer Statistics

- **Description:** As a user, I would like to be able to visualize ontologies in an interactive, dynamic manner. I would like this to be implemented because it would greatly help me visualize and understand Ontologies and Ontology formats.
- **Acceptance Criteria:**
 - All of the Developer statistics that have been implemented are now exportable via a CSV file.

- The ID of each query and article is stored in this CSV file.
- The CSV file is not limited by the pagination standards, and exports the whole database information.
- **Use Case:**
 - **Name:** User Story 192
 - **Actor:** User
 - **Description <Flow of events>**
 - * User is able to download all of the CSV files by clicking a button named “Export as CSV file”.
 - * User can see the article id and query id of each tuple in this CSV file, and the CSV file is not corrupted.

3.1.6 User Story 191 - User Profile Page Enhancement

- **Description:** The user should be able to select a profile option and be redirected to the appropriate view where they can access the features of that page.
- **Acceptance Criteria:**
 - The user should be able to select a profile option and be redirected to the appropriate view where they can access the features of that page.
- **Use Case:**
 - **Name:** User Story 191
 - **Actor:** User
 - **Description <Flow of events>**
 - * User should be able to select one of the profile options and be redirected to a page containing the corresponding content
 - * The user should be able to click on edit profile and be taken to a page containing the edit profile functionality
 - * Users should be able to see their saved articles and alter their list by accessing “Saved Articles:
 - * User can click on history to view their recent queries and article clicks

3.1.7 User Story 186 - User Mobile Experience

- **Description:** As a user, I would like this platform to be usable and navigable from a mobile device, because this will allow me to utilize the website from a device which is easily accessible.
- **Acceptance Criteria:**
 - The navigation bar has a collapsible animation showing all of the different pages and how to access them.
 - Page navigation is implemented for mobile browsers.
 - The interface styling is adapted to follow Gestalt's Design Principles and the components are respectively scaled for a mobile browser.
- **Use Case:**
 - **Name:** User Story 186
 - **Actor:** User
 - **Description <Flow of events>**
 - * User is able to access the main page and see all of the elements in a concise manner.
 - * User has the ability to navigate to all of the pages currently accessible via a desktop browser.
 - * User can expand and collapse the navigation bar as he sees fit, using both the search bar and the hamburger menu navigation.

3.1.8 User Story 188 - Developer Mobile Experience

- **Description:** As a user I would like to have a mobile-friendly view when I am navigating the developer page and its different components.
- **Acceptance Criteria:**
 - The user is able to access the developer stats page from the profile menu while on mobile
 - The user can access the different developer stats from the hamburger menu while in the developer stats page
 - The navigation tabs are correctly styled and scaled for a mobile browser.
- **Use Case:**

- **Name:** User Story 188
- **Actor:** User
- **Description** <Flow of events>
 - * User is able to see a list of options for different user activity data.
 - * User can click the buttons to access the tables for the developer data.
 - * User has the option to create a new developer by entering an email.

3.1.9 User Story 185 - Dynamic Ontology Visualization

- **Description:** As a user, I would like to be able to visualize ontologies in an interactive, dynamic manner. I would like this to be implemented because it would greatly help me visualize and understand Ontologies and Ontology formats.
- **Acceptance Criteria:**
 - The component implemented provides a way to visualize sections of an ontology in a specific time.
 - Allows for dynamic expansion of nodes to view neighboring nodes and their links.
 - All nodes are aptly shown on the screen and have their properties and links correctly labeled.
 - A way to search for a specific node and include it in the current graph state is provided.
- **Use Case:**
 - **Name:** User Story 185
 - **Actor:** User
 - **Description** <Flow of events>
 - * User is able to click on a node, view the node properties, and export the node properties.
 - * User can click on a node to dynamically visualize its neighboring nodes.
 - * User is able to search for a specific node via a search bar and dynamically see it being added in the graph when the search term is clicked on.

3.1.10 User Story 187 - Embedded New Ontology Visualization

- **Description:** There is a tab in the ontology page where the user can choose to navigate either to the tree-view or the node graph.
- **Acceptance Criteria:**
 - There is a tab in the ontology page where the user can choose to navigate either to the tree-view or the node graph.
- **Use Case:**
 - **Name:** User Story 187
 - **Actor:** User
 - **Description <Flow of events>**
 - * User is able to click on the node graph tab and be redirected to a full screen visualization of the ontology tree
 - * User is able to click on the treeview button from the node graph and be redirected back to the tree-view

3.2 User Stories Pending for Next Team

All of the user stories that were assigned by the Product Owner have been completed, and following user stories were created so that the next team can have a starting point. Below you will find some of these user stories that we believe will greatly enhance the product and should be a priority:

- User Story 193 - Elastic Search API used for Ontology Visualization
 - As a user, I would like for the elastic search API that is being used to gather the search results to be utilized by the custom WebVOWL Implementation of the project, so that I am able to dynamically visualize the actual Ontology present in this API over time instead of manually having to import a OWL file.

4 Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the updated components of EnvoScholar v3.0, both software and hardware, chosen for this project.

4.1 Hardware and Software Resources

- Angular 6
 - Angular 6 was used to build the front end functionality with individual components per feature excluding the treeview and node graph visualizations.
- JQuery/D3
 - JQuery was used to create the treeview and search for the ontology visualizations on the ontology page.
 - Note: On EnvoScholar v3.0, the ontology node visualization now uses the WebVOWL open source project as basis for the custom implementation.
- NodeJS/Express
 - NodeJS/Express was used to build the server that connects to the Mongo Database.
- MongoDB
 - MongoDB was used to create the EnvoScholar v3.0 database which contains a collection of users (user profile information) and a collection of feedback (user experience feedback). This Database now contains extra controllers used for providing extra developer statistics as well as providing the exportable CSV object by aggregation.
- PostgreSQL/Elasticsearch
 - PostgreSQL was used to create the database that contains the articles.
 - Elasticsearch was used in the HTTP GET and POST requests to the database to retrieve the articles.

- Note: Maria Presa Reyes, an FIU graduate student, is the main contributor for this part of the project.
- HTML5/CSS
 - HTML5 and CSS were used to structure the visual aspect of the project and work on the styling.
- Passport
 - Passport was used as the authentication middleware for NodeJS for users logging into their profile.
- JWT
 - JWT was used to create the token for login authentication.
- MacOS/Windows 10/Linux
 - All of the above platforms were used and tested for the development and deployment of EnvoScholar v3.0.

4.2 Sprints Plan

4.2.1 Sprint 1

Date: September 9, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 2:00 PM

End Time: 3:00 PM

After the discussion, the velocity of the team was estimated to be 70.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority:

- User Story 68 - Website Deployment
- User Story 190 - Server Side Filtering

The team members indicated their willingness to work on the following user stories:

- Dairon Hernandez
 - Feature 190 - Server Side Filtering
- Jorge Euceda
 - Feature 68 Website Deployment

4.2.2 Sprint 2

Date: September 23, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 2:00 PM

End Time: 3:00 PM

After the discussion, the velocity of the team was estimated to be 70. The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority:

- User Story 189 - Server Side Loading and Pagination
- User Story 57 - Developer Features

The team members indicated their willingness to work on the following features:

- Dairon Hernandez
 - Feature 57 - Developer Features
- Jorge Euceda
 - Feature 57- Server Side Loading and Pagination

4.2.3 Sprint 3

Date: October 3, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 2:00 PM

End Time: 3:00 PM

After the discussion, the velocity of the team was estimated to be 70. The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority:

-
- User Story 192 Exportable Developer Statistics
- User Story 191 User Profile Page Enhancement

The team members indicated their willingness to work on the following features:

- Dairon Hernandez
 - User Story 192 Exportable Developer Statistics
 - User Story 191 User Profile Page Enhancement
- Jorge Euceda
 - User Story 192 Exportable Developer Statistics
 - User Story 191 User Profile Page Enhancement

4.2.4 Sprint 4

Date: October 21, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 2:00 PM

End Time: 3:00 PM

After the discussion, the velocity of the team was estimated to be 70. The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority:

- User Story 186 - User Mobile Experience
- User Story 188 - Developer Experience
- Exportable Statistics

The team members indicated their willingness to work on the following features:

- Dairon Hernandez
 - User Story 186 - User Mobile Experience
 - User Story 188 - Developer Experience
- Jorge Euceda
 - User Story 186 - User Mobile Experience
 - User Story 188 - Developer Experience

4.2.5 Sprint 5

Date: November 4th, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 2:00 PM

End Time: 3:00 PM

After the discussion, the velocity of the team was estimated to be 70.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority:

- User Story 185 - Dynamic Ontology Visualization
- User Story 187 - Embedded New Ontology Visualization

The team members indicated their willingness to work on the following features:

- Dairon Hernandez
 - User Story 187 - Embedded New Ontology Visualization
 - User Story 185 - Dynamic Ontology Visualization (assisting Jorge Euceda when needed)
- Jorge Euceda
 - User Story 185 - Dynamic Ontology Visualization

4.2.6 Sprint 6

Date: November 18th, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 2:00 PM

End Time: 3:00 PM

After the discussion, the velocity of the team was estimated to be 70.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority:

- User Story 179 - Document Ontology Visualization components.
- User Story 181 - Improve Project Structure.
- User Story 182 - Perform End-to-End Unit Tests on Major Components.
- User Story 183 - Document Angular Frontend Components.

The team members indicated their willingness to work on the following features:

- Dairon Hernandez
 - User Story 182 - Perform End-to-End Unit Tests on Major Components
 - User Story 181 - Improve Project Structure
- Jorge Euceda
 - User Story 181 - Improve Project Structure
 - User Story 183 - Document Angular Frontend Components

5 System Design

This section contains information on the design decisions that went into EnvoScholar v3.0. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

5.1 Architectural Patterns

A Client-Server and Model View Controller (MVC) architectural pattern was used in this project for the presentation layer as it allows the separation of internal information from the ways that the information is actually displayed to viewer. The MVC architectural pattern includes three interconnected parts, that is, model, controller, and view. The model determines the application data structure and manages data and logic, while the view presents the actual information such as article search results to the users. The controller receives the request and then passes them to the model. Such an architectural pattern allows the coding of each part separately.

5.2 System and Subsystem Decomposition

There are three main systems in EnvoScholar, the web client subsystem, the business logic subsystem, and the data subsystem. In this section, we will describe what each of those subsections did individually to contribute to the creation of this academic search engine.

5.3 Data Subsystem

The data system is the subsystem which is responsible for the access, storage, and management of data. The data management system used in this project is MongoDB to store the user account information along with the global feedback information. The MongoDB part of this project that was manipulated in this iteration of the project by us was the storage of the feedback for an individual feedback and a global feedback value. The Elastic search is used for all the data management used with the articles in the search engine. As mentioned in Section 3, this Elastic search was minimally not accessed by us and only accessed by the graduate students of the project.

6 Appendix

6.1 Appendix A - UML Diagrams

6.1.1 Class Diagrams

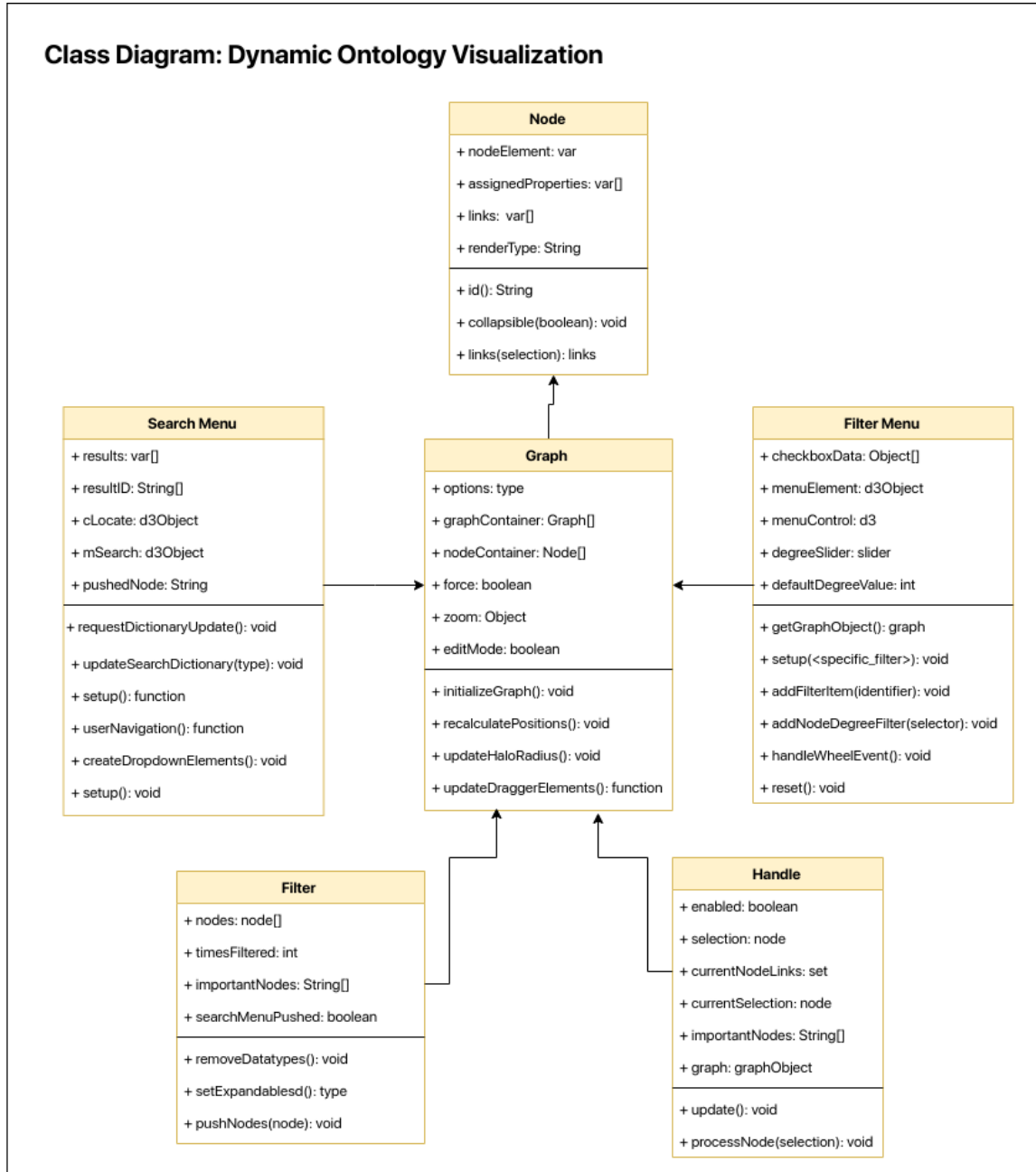


Figure 1: User Story 185 - Dynamic Ontology Visualization Class Diagram

Website Deployment Class Diagram

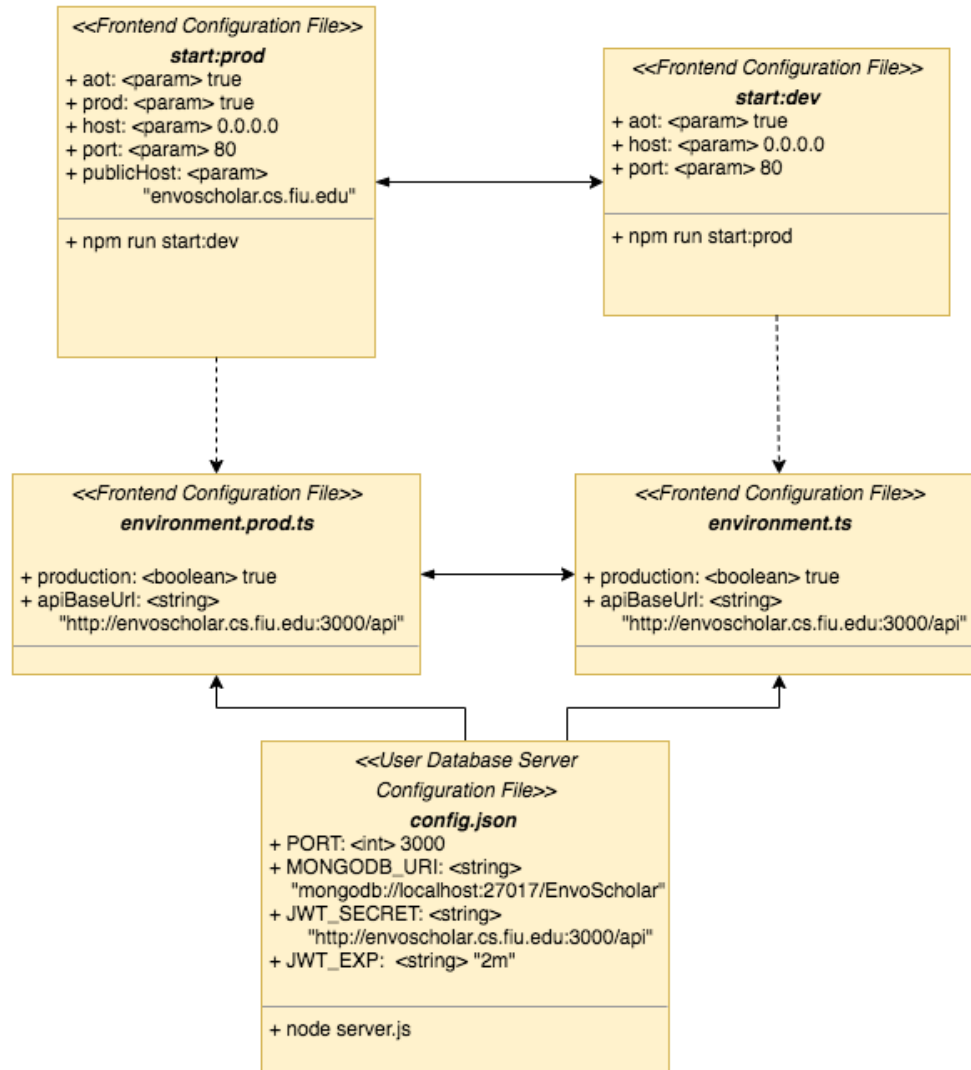


Figure 2: User Story 68 - Website Deployment Class Diagram

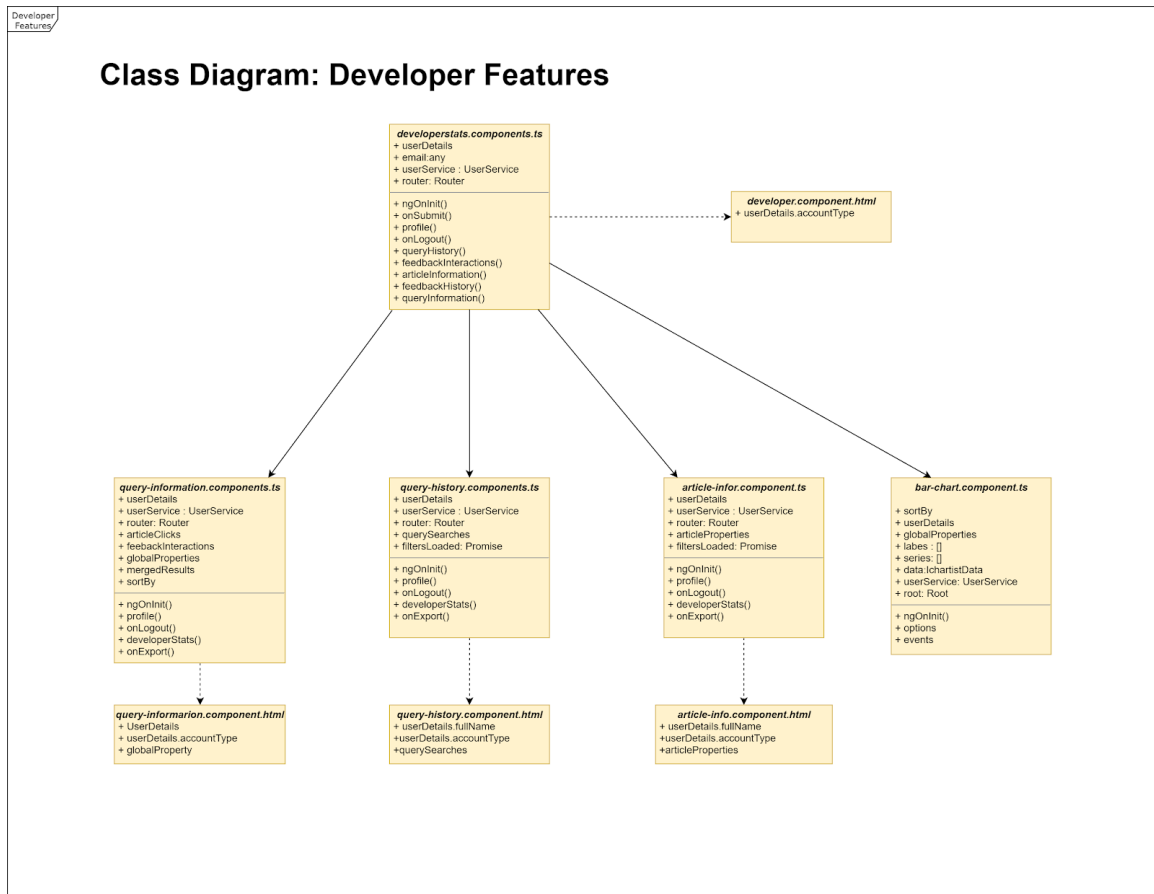


Figure 3: User Story 57 - Developer Features Class Diagram

Class Diagram: Exportable Developer Statistics

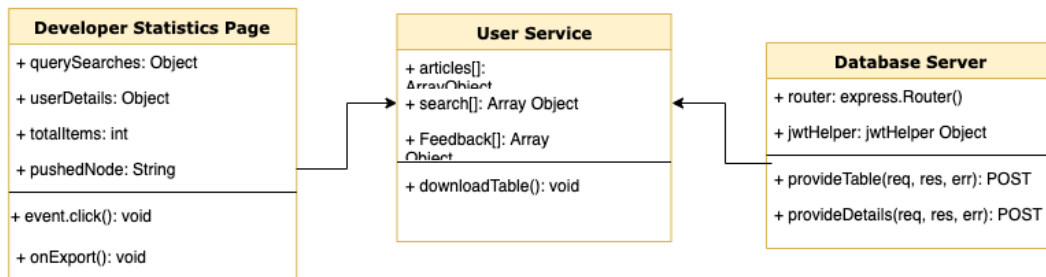


Figure 4: User Story 192 - Exportable Developer Statistics Class Diagram

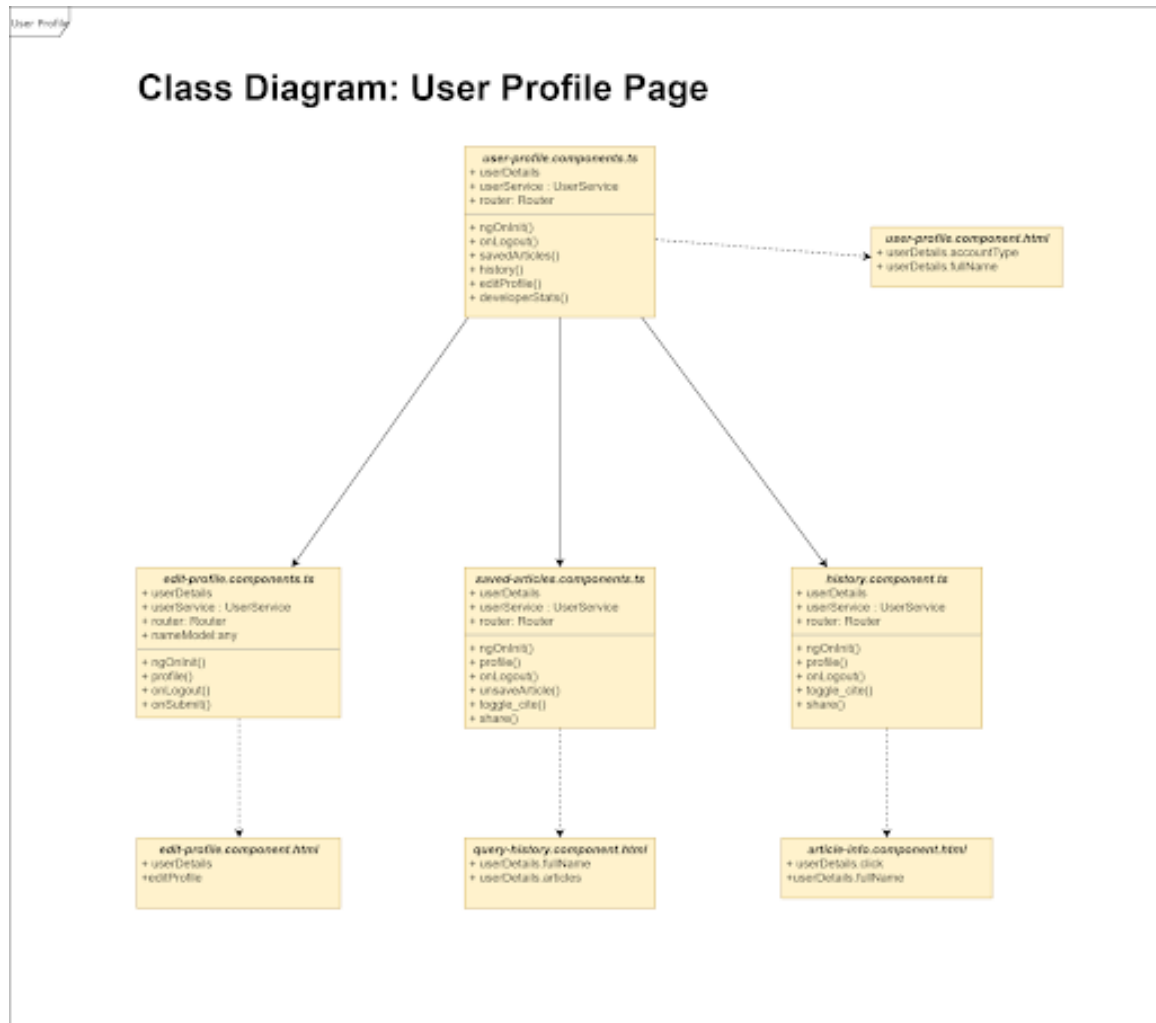


Figure 5: User Story 191 - User Profile Page Enhancement Class Diagram

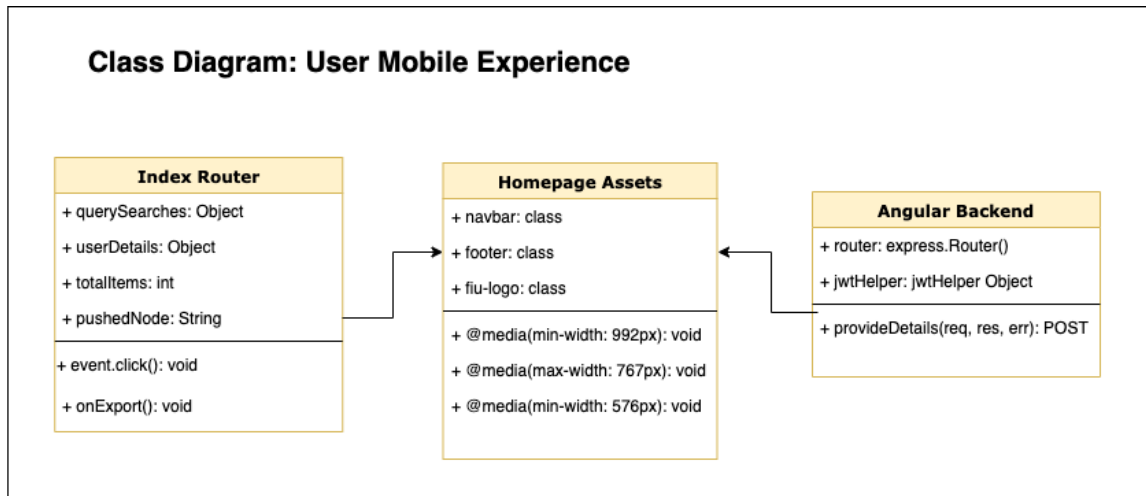


Figure 6: User Story 186 - User Mobile Experience Class Diagram

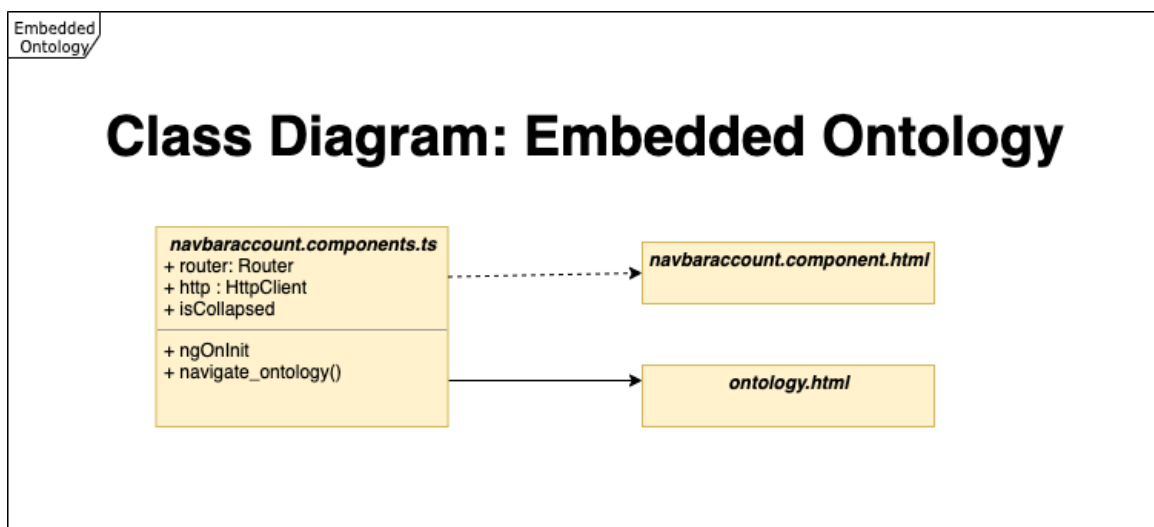


Figure 7: User Story 187 - Embedded New Ontology Visualization Class Diagram

6.1.2 Use Case Diagrams

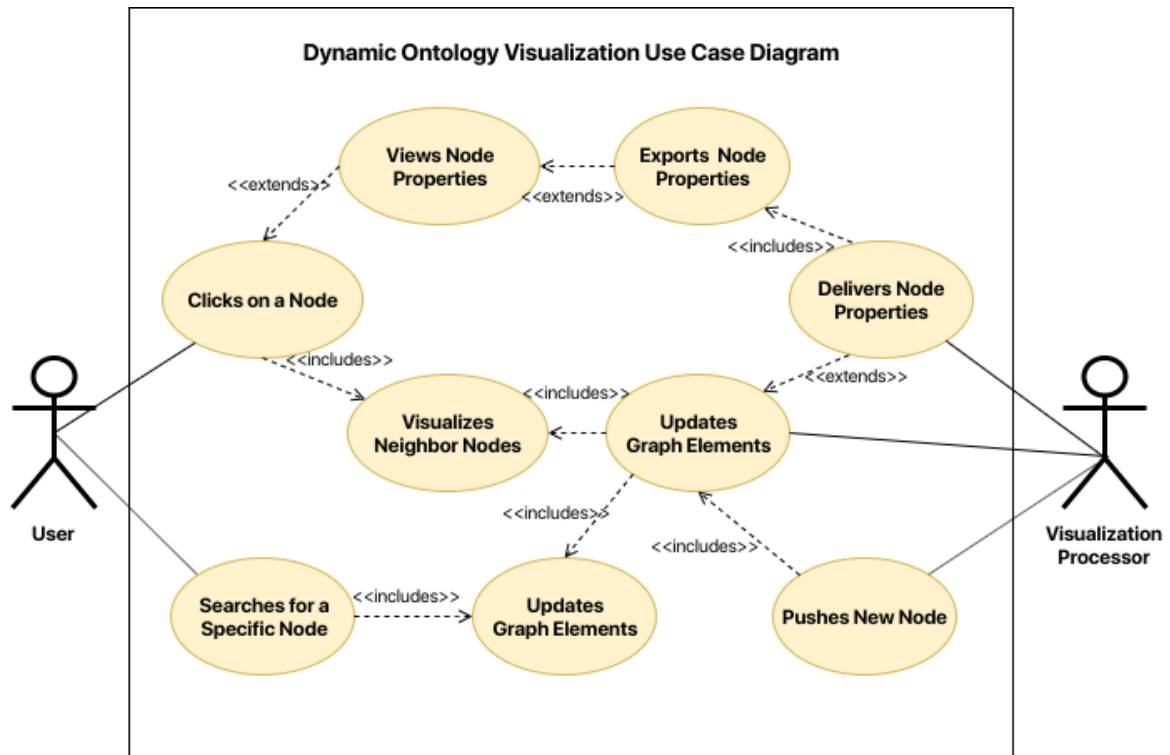


Figure 8: User Story 185 - Dynamic Ontology Visualization Use Case Diagram

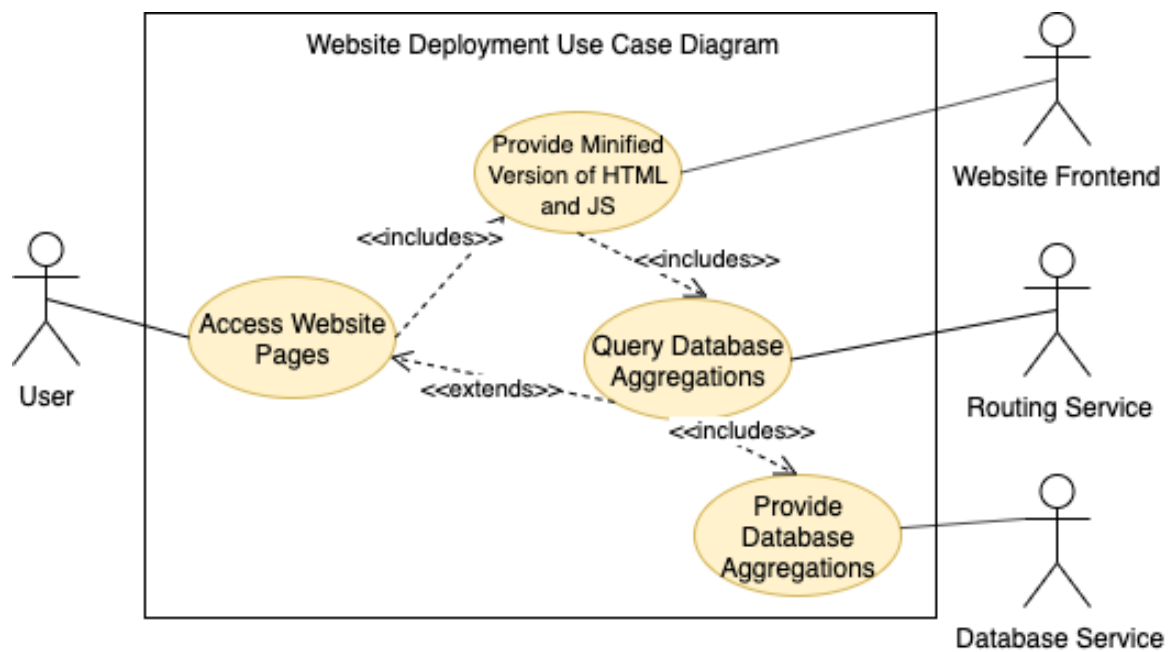


Figure 9: User Story 68 - Website Deployment Sequence Diagram

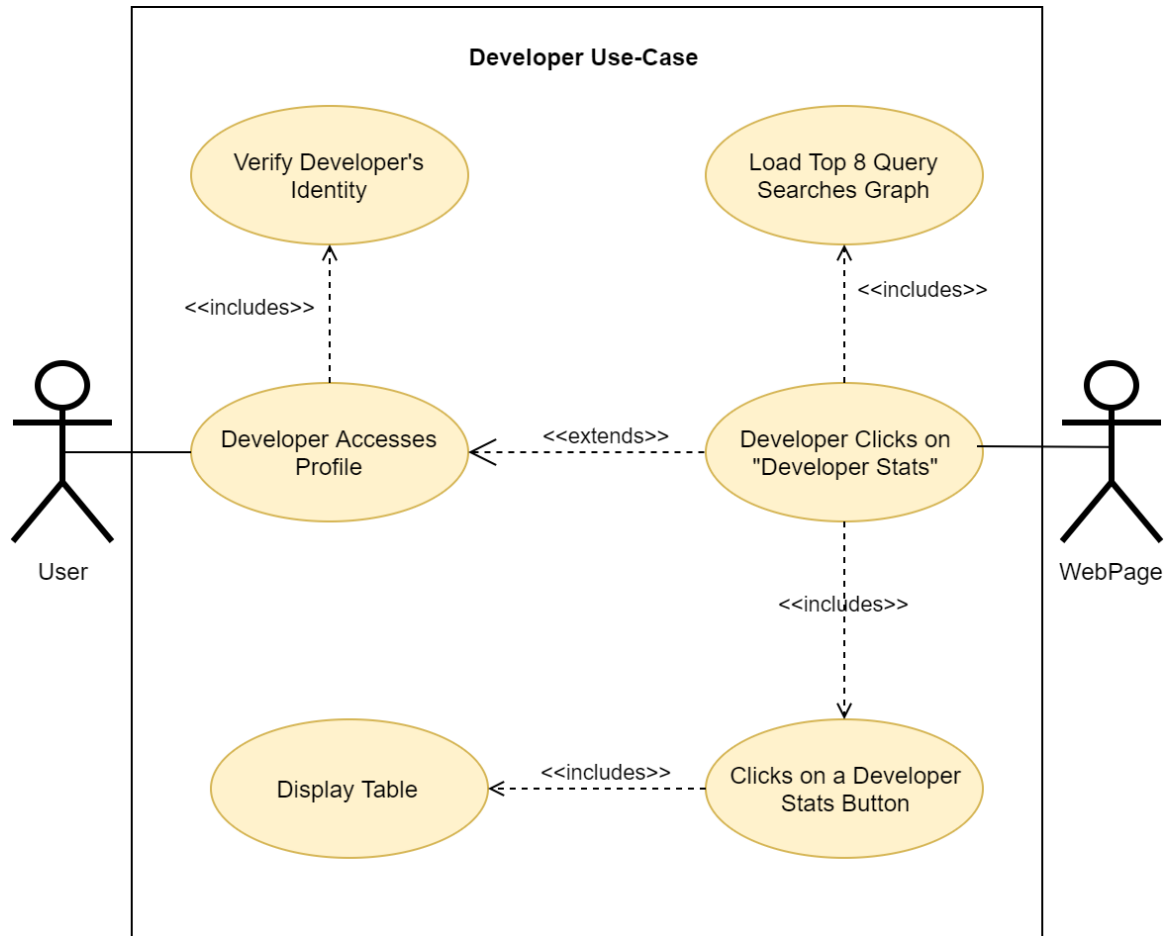


Figure 10: User Story 57 - Developer Features Use Case Diagram

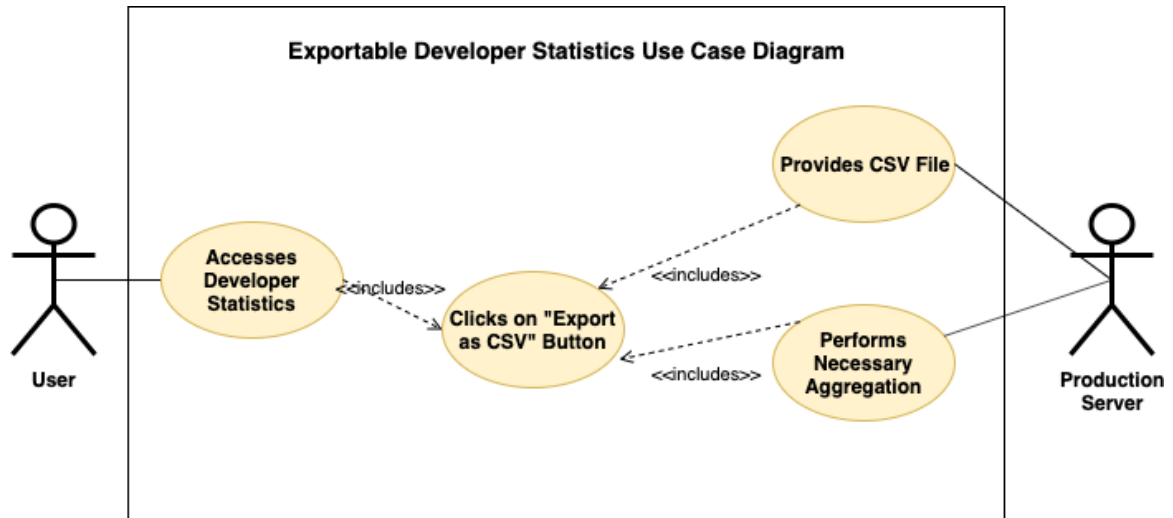


Figure 11: User Story 192 - Exportable Developer Statistics Use Case Diagram

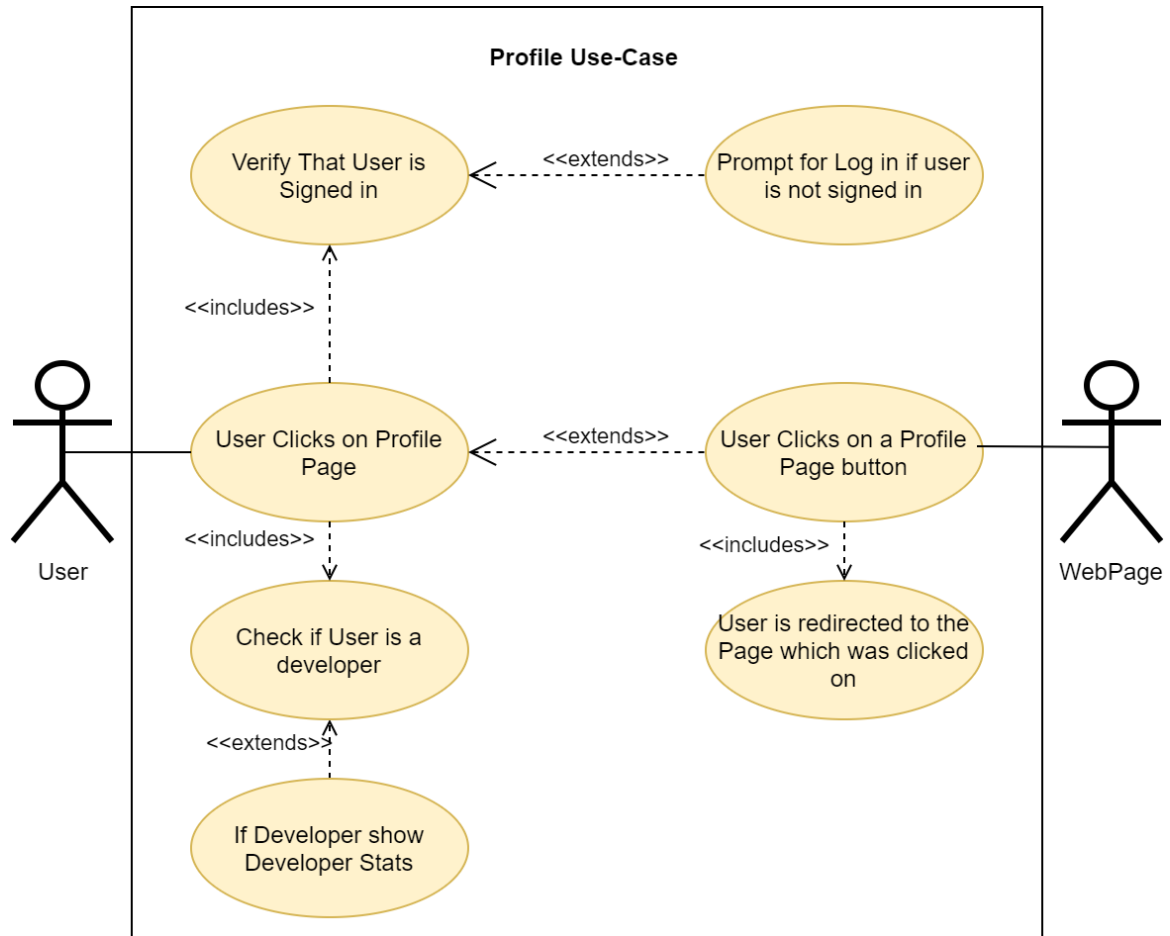


Figure 12: User Story 191 - User Profile Page Enhancement Use Case Diagram

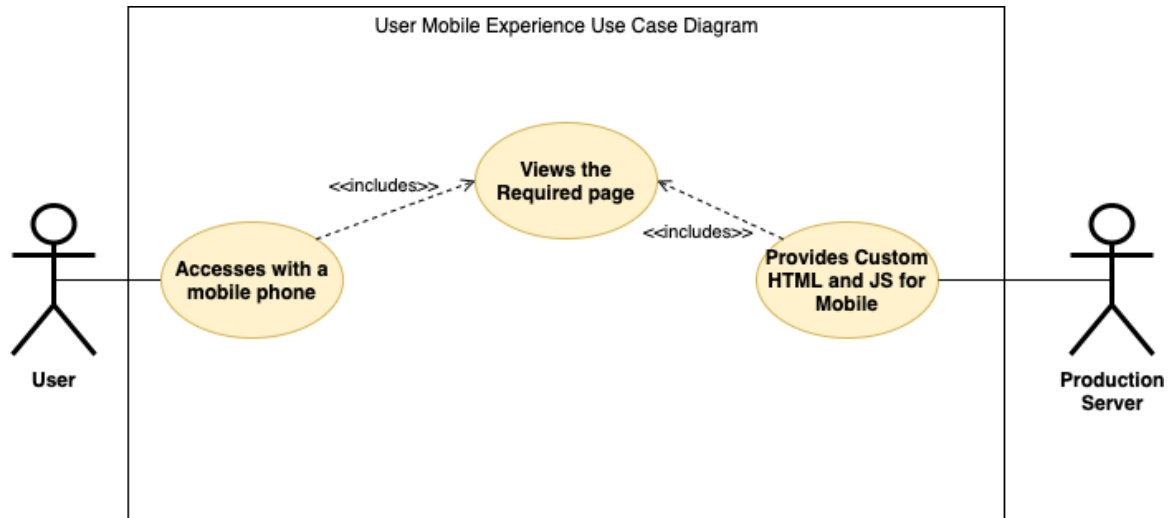


Figure 13: User Story 186 - User Mobile Experience Use Case Diagram

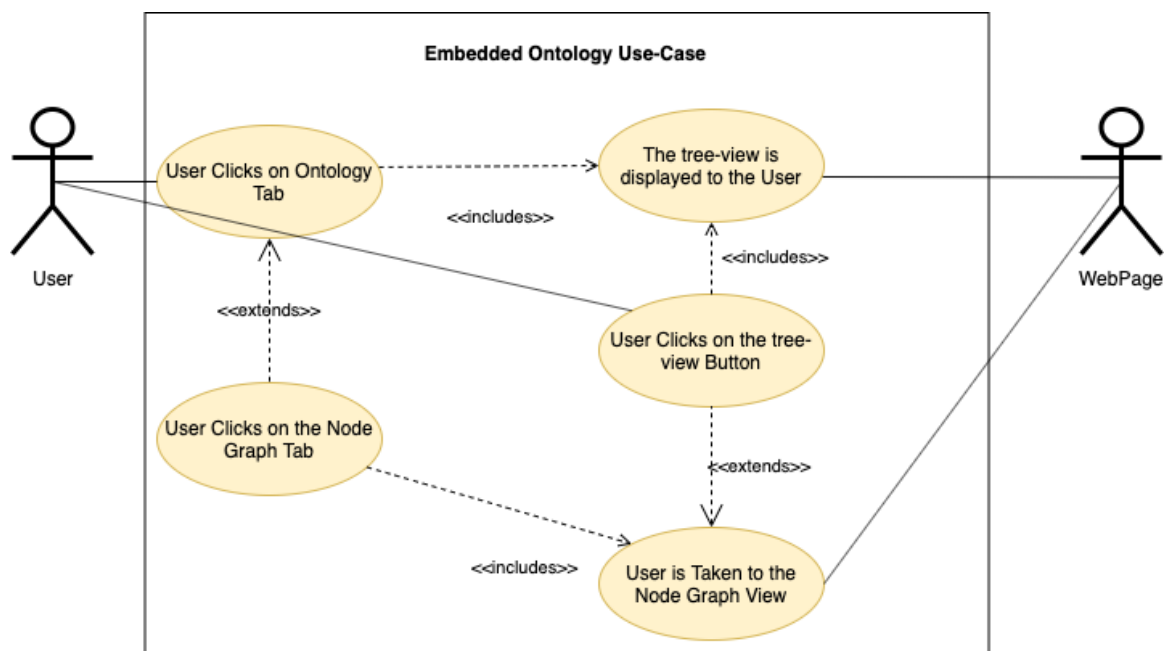


Figure 14: User Story 187 - Embedded New Ontology Visualization Use Case Diagram

6.1.3 Sequence Diagrams

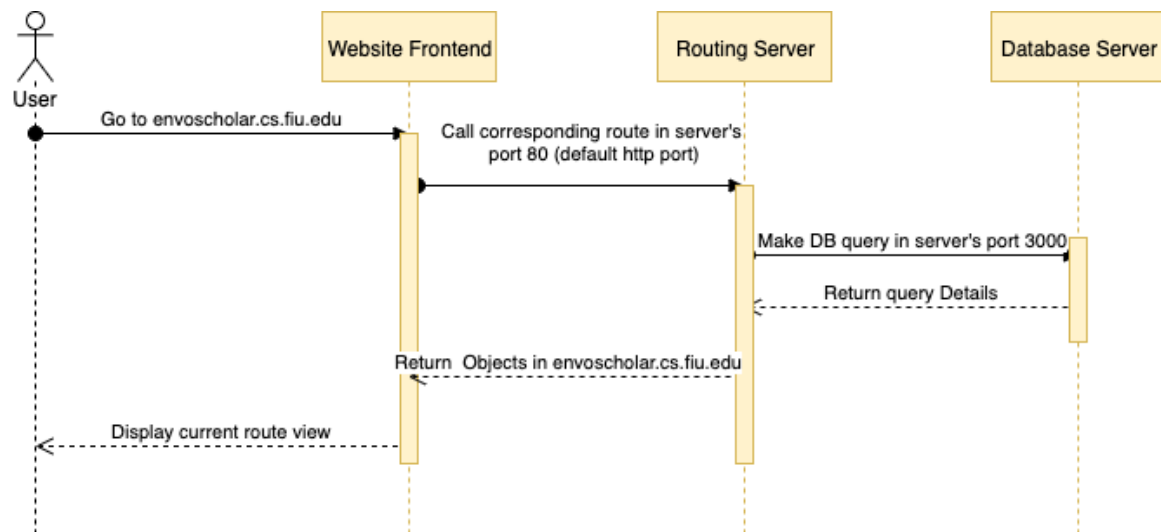


Figure 15: User Story 68 - Website Deployment Sequence Diagram

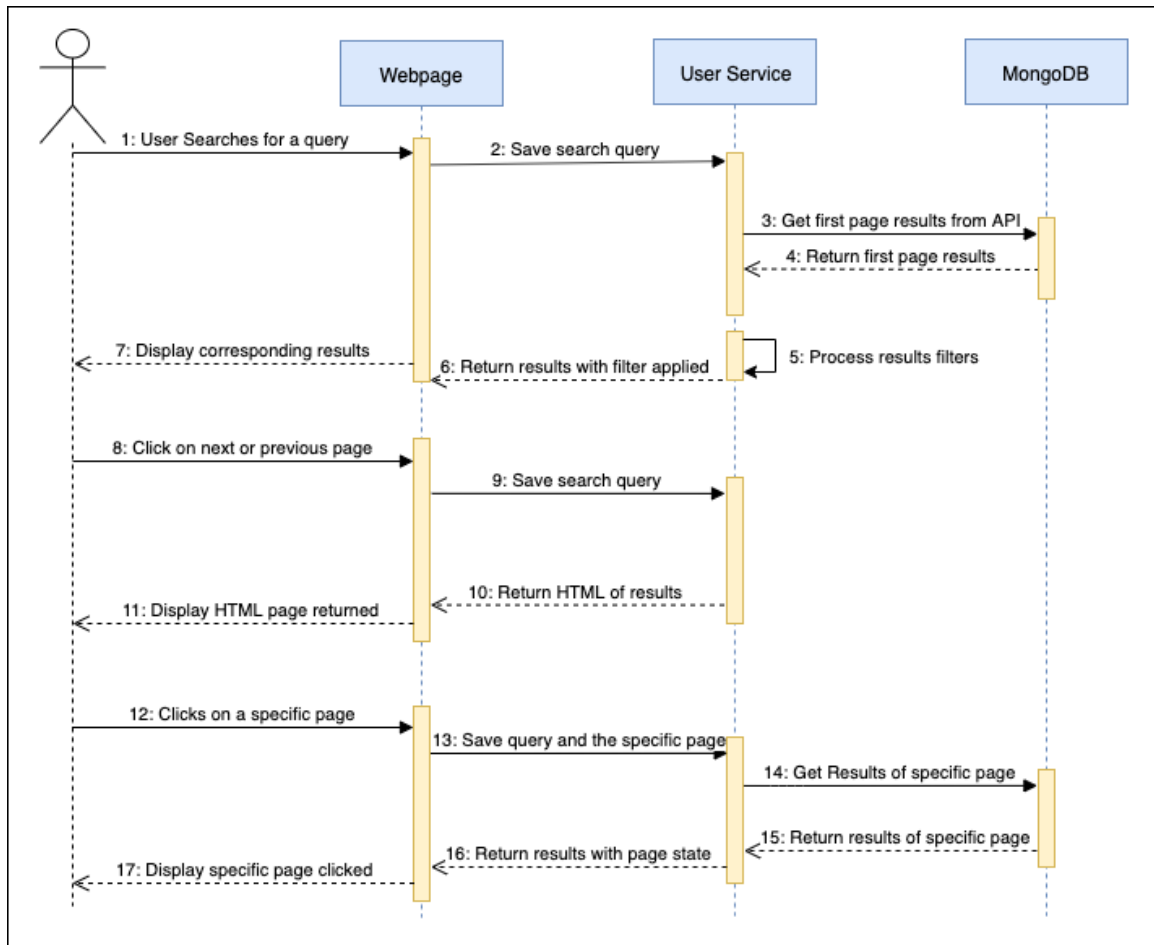


Figure 16: User Story 190 - Server Side Filtering Sequence Diagram

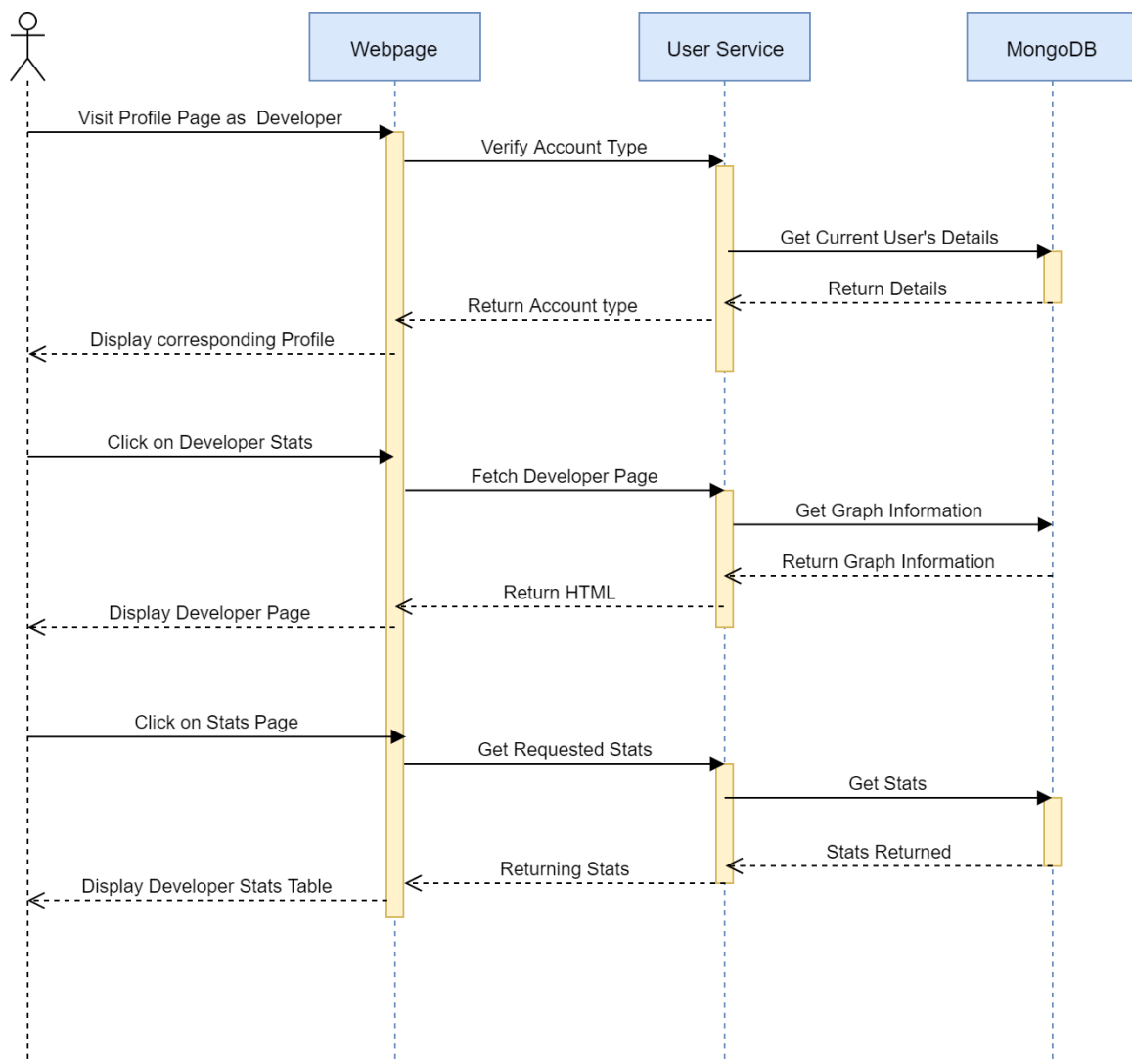


Figure 17: User Story 57 - Developer Features Sequence Case Diagram

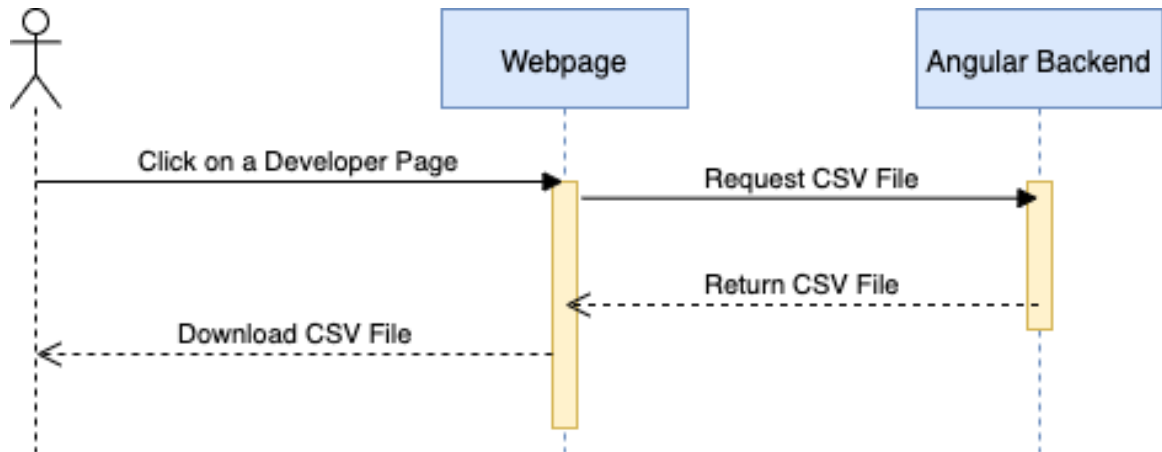


Figure 18: User Story 192 - Exportable Developer Statistics Sequence Diagram

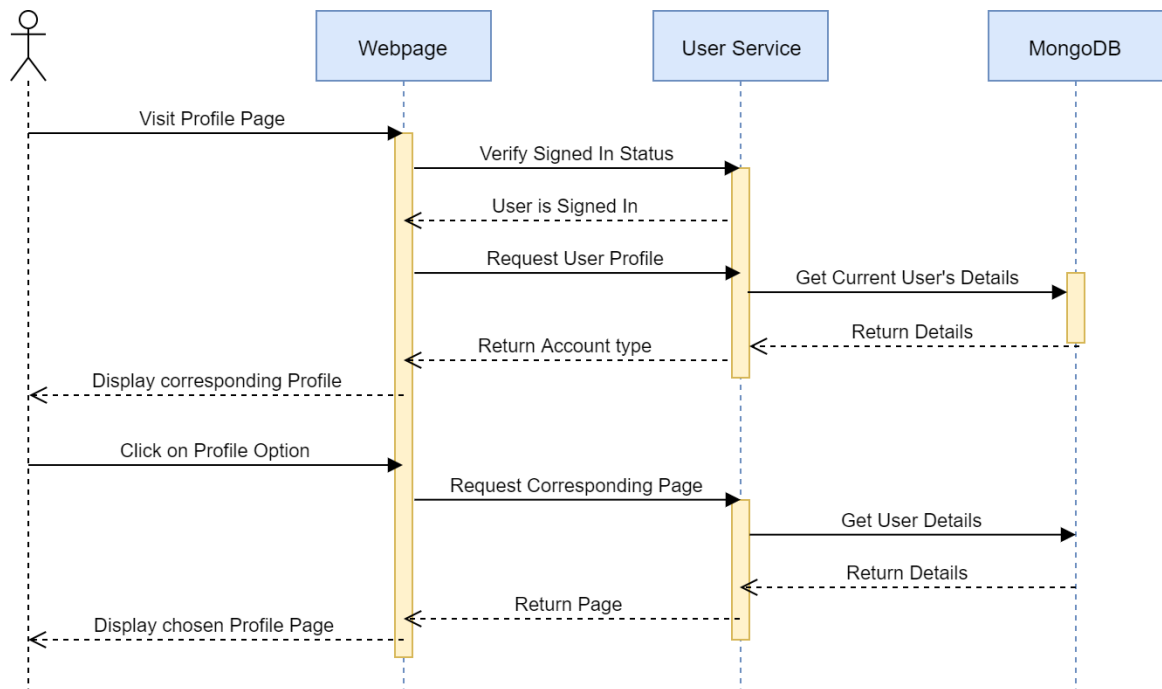


Figure 19: User Story 191 - User Profile Page Enhancement Sequence Diagram

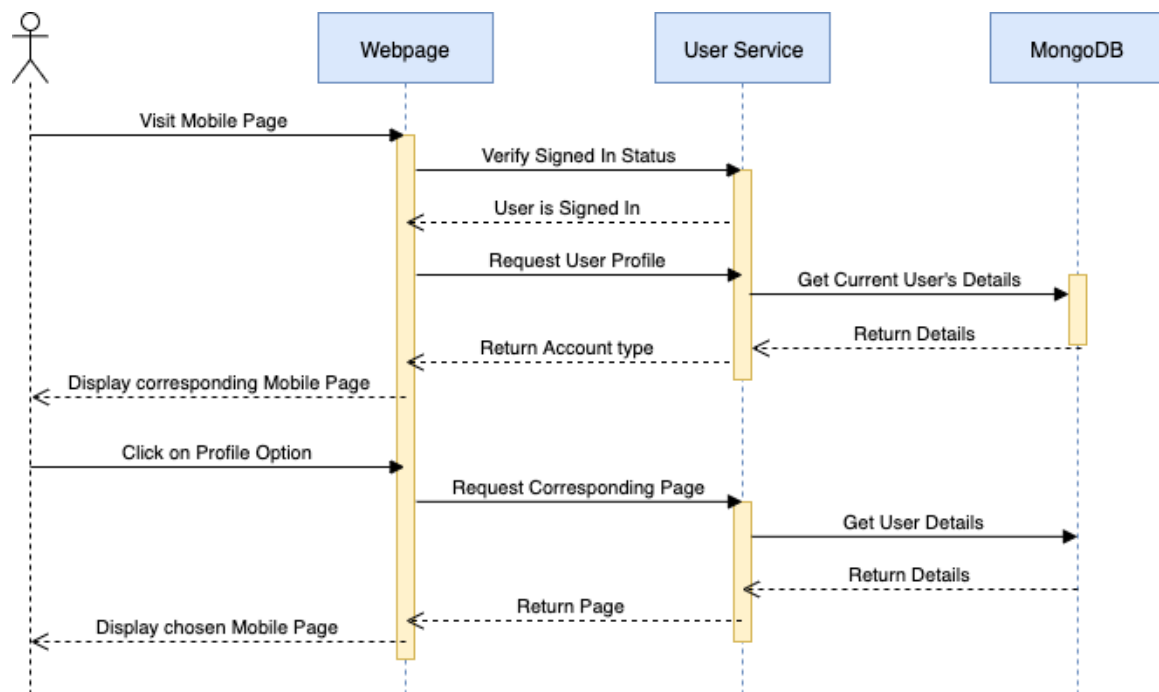


Figure 20: User Story 186 - User Mobile Experience Sequence Diagram

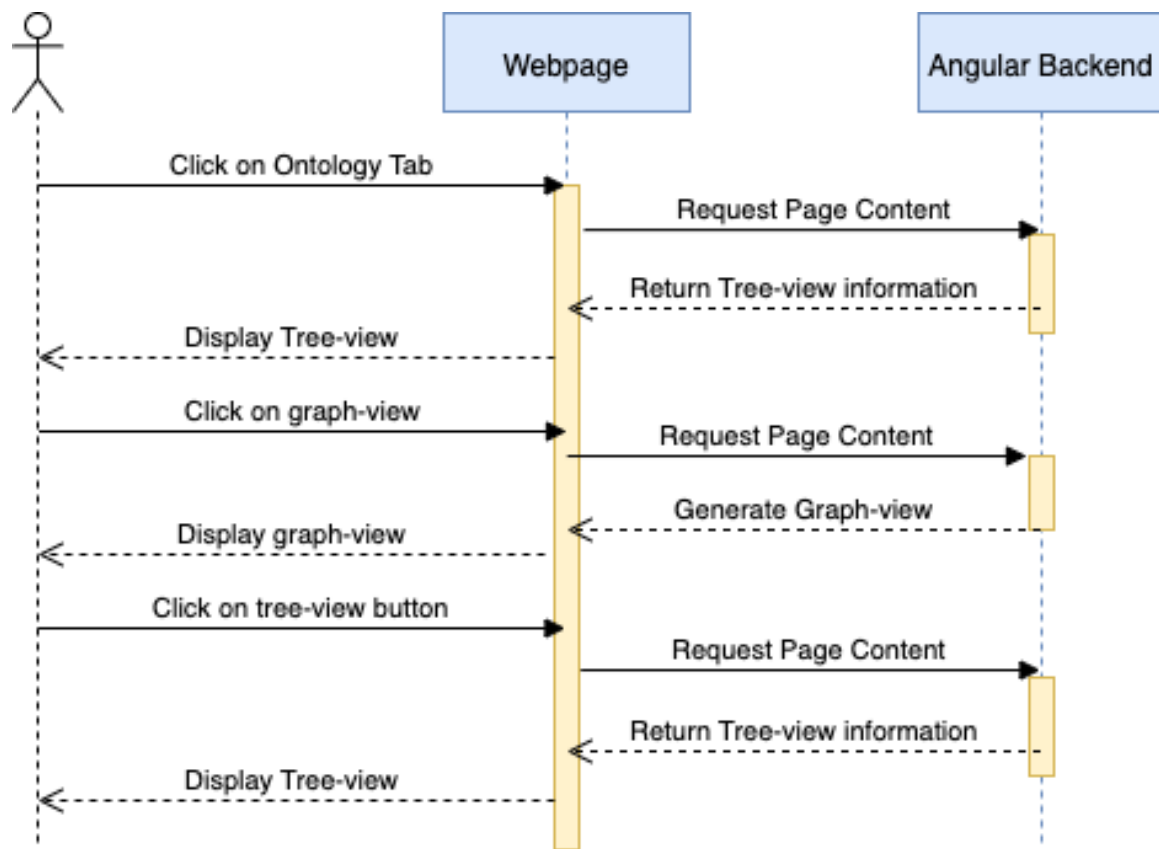


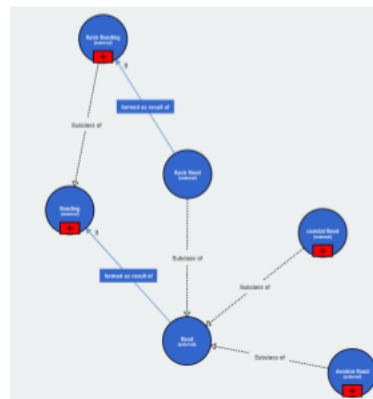
Figure 21: User Story 187 - Embedded New Ontology Visualization Sequence Diagram

6.2 Appendix B - User Interface Design

Dynamic Ontology Visualization

Dynamic node expansion

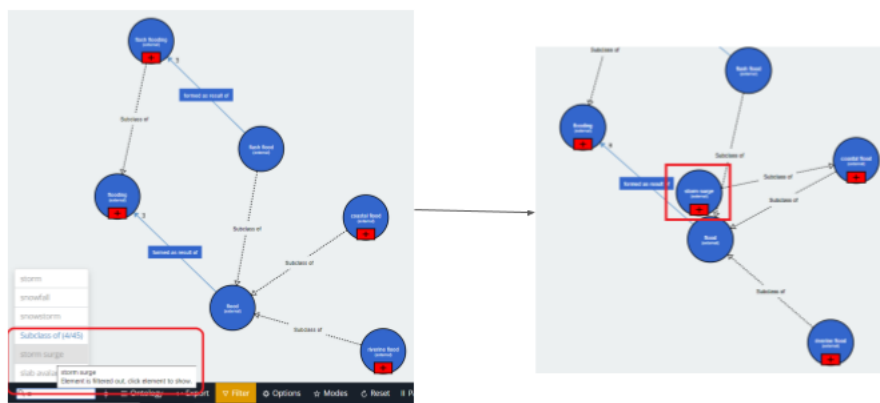
- Helps the user easily visualize the ontology and easily view neighbors of each node.
- Nodes are interactable, and expand to show the neighboring nodes in the graph when clicked.



Dynamic Ontology Visualization

Interactive Node Inclusion

- Allows the user to easily add nodes to a current view, based on their name



Developer Statistics

Query Information

- Helps determine the popularity of a query and interactions achieved in that query.

Query Name	Amount of Query Searches	Amount of Article Clicks	Amount of Relevant Articles
drugs	214	13	12
water	114	17	57
immaterial entity	39	3	6
sensitivity analysis	34	1	3

Query History

- Allows developers to view at what time a query was searched and the state of the article interactions.

Query	Time	Article Clicks	Articles Marked as Relevant
drugs	December 4, 2019, 4:09:54 PM	13	12
drugs	December 4, 2019, 10:32:46 AM	13	12
water	December 3, 2019, 1:27:11 PM	17	57

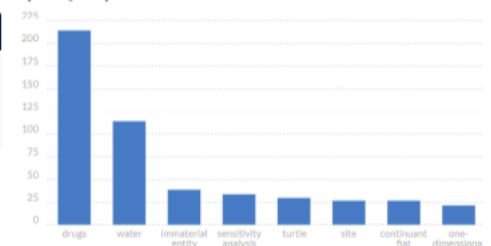
Developer Statistics (Part 2)

Article Information

Article Title	Query Name	Relevance Count	Article Clicks
Decomposition of drug mixture in Fenton and photo-Fenton processes: Comparison to singly treatment, evolution of inorganic ions and toxicity assay	drugs	4	5
Occurrence of PAHs, PCBs and organochlorine pesticides in the Tonghui River of Beijing, China	water	3	4
Measurement of pyrene in the gills of exposed fish using synchronous fluorescence spectroscopy	immaterial entity	0	3

Graph with statistical summary

Top 8 Query Searches



Mobile Experience



- Main pages are now designed with mobile devices in mind.
- The Website is now entirely navigable using a Sandwich menu with a dropdown and respective routes to each page.



Additional Features

- Search bar that retains the state of the current search is now included in the navigation bar
- Enhanced tabbing mechanism for new ontology node graph visualization



- Improved project structure, components are now organized into categories.
- Overhauled display results components to enhance user experience and interactions.



Server-Side Pagination

Pagination of results occurs in an optimized manner on the server-side

- Only one page is loaded at a specific time, allowing for optimized loading performance of results displayed.
- A theoretically infinite number of pages can now be loaded, only limited by the API call (previous limit was only 10 pages).
- Page loading times are significantly decreased, as the processing and API calling no longer occurs on the frontend. (5ms average when compared to 3 second average).

Distribution and seasonal occurrence of UV filters in rivers and wastewater treatment plants in Korea

Un-Jung Kim-Sung-Hee O-Hee-Young Kim-2016-01-15

The occurrence and distribution of eight UV filters benzophenone (BP), benzophenone triethylhexyl methoxy cinnamate (EHMC), 4-methylbenzylidene camphor (4-MBC), 2-ethylhexyl 4-dimethylaminobenz...

[View PDF](#) [Cite](#) [Save](#) [Share](#) [Relevant](#) [Similar Articles](#)

« Previous **1** 2 3 4 5 ... **990** Next »

Loaded 9900 results in 4 milliseconds

Is there evidence for man-made nanoparticles in the Dutch environment?

Patrick S. Bäuerlein-Erik Emke-Peter Tromp-2017-01-15

Only very limited information is available on measured environmental concentrations of nanoparticles. In this study, several environmental compartments in The Netherlands were probed for the presence ...

[View PDF](#) [Cite](#) [Unsave](#) [Share](#) [Relevant](#) [Similar Articles](#)

6.3 Appendix C - Sprint Review Reports

6.3.1 Sprint 1

Date: September 22, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 1:30 PM

End Time: 2:00 PM

After a show and tell presentation, the implementation of the following tasks were accepted by the product owners: All.

- Feature 68 - Website Deployment
- Feature 190 - Server Side Filtering

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning Meeting: None.

6.3.2 Sprint 2

Date: October 6, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 1:30 PM

End Time: 2:00 PM

After a show and tell presentation, the implementation of the following tasks were accepted by the product owners: All.

- Feature 189 - Server Side Loading and Pagination
- Feature 57 - Developer Features

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning Meeting: None.

6.3.3 Sprint 3

Date: October 20, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 1:30 PM

End Time: 2:00 PM

After a show and tell presentation, the implementation of the following tasks were accepted by the product owners: All.

- Feature 192 - Exportable Developer Statistics
- Feature 191 - User Profile Page Enhancement

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning Meeting: None.

6.3.4 Sprint 4

Date: November 3, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 1:30 PM

End Time: 2:00 PM

After a show and tell presentation, the implementation of the following tasks were accepted by the product owners: All.

- Feature 186 - User Mobile Experience
- Feature 188 - Developer Experience

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning Meeting: None.

6.3.5 Sprint 5

Date: November 17, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 1:30 PM

End Time: 2:00 PM

After a show and tell presentation, the implementation of the following tasks were accepted by the product owners: All.

- Feature 185 - Dynamic Ontology Visualization
- Feature 187 - Embedded New Ontology Visualization

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning Meeting: None.

6.3.6 Sprint 6

Date: December 1, 2019

Attendees: Dairon Hernandez, Jorge Euceda, Maria A. Presa Reyes, Deya Banisakher

Start Time: 1:30 PM

End Time: 2:00 PM

After a show and tell presentation, the implementation of the following tasks were accepted by the product owners: All.

- Feature 179 - Document Ontology Visualization Components.
- Feature 181 - Improve Project Structure.
- Feature 182 - Perform End-to-End Unit Tests on Major Components.
- Feature 183 - Document Angular Frontend Components.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning Meeting: None.

6.4 Appendix D - User Manuals, Installation/Maintenance Document, and Shortcomings/Wishlist Document, and Other Documents

You are able to find all of these documents in the Final Deliverable zip File. This also includes all of the Instructional Videos along with the Code.