Bilkent University
Department of Computer Engineering

Senior Design Project

T2334

PaperAtlas

Analysis and Requirement Report

Ahmet Hakan Yılmaz - 21803399

Akın Kutlu - 21803504

Aybala Karakaya - 21801630

Selbi Ereshova - 21901326

Zehra Erdem - 21801977

Supervisor: Uğur Doğrusöz

Jury Members: Erhan Dolak and Tağmaç Topal

13.11.2022

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1. Introduction

Research papers can be reached online with the help of internet and online platforms such as Google Scholar. However, because of the increasing number of research papers, it becomes hard to keep track of information [1]. Reaching newest findings related to a topic or old papers that explains some fundamentals becomes hard and time consuming. Though lots of search algorithms are used to find papers related to a topic, finding relations between these papers is still tedious [2]. It needs to be improved. Reference relations between papers are important because if the papers that cited a given paper are known, the newest finding related to that paper can be found easily. Moreover, to understand a topic better, some old papers need to be looked at. To find the ones that have fundamental information about the topic, commonly cited papers by already known papers should be looked at. However, finding the common references can be hard when papers are looked at separately. This application, Paper Atlas, makes it easy for researchers to find correct papers by visualizing papers and their reference connections as a graph.

The innovation we are planning to implement is process innovation. Process innovation aims to implement a new or significantly improved product or service. Paper Atlas will be an application that provides services for searching papers and visualization of these papers as a graph. After deciding on the project topic, similar projects were looked at, and it was found that there is a website that provides a service for graph visualization of papers. We want to create a new application by improving this old application by adding new features. As we have an example website we want to improve, and we will add new features different from the previous websites. The right innovation type for Paper Atlas is process innovation. When we also look at other types of innovations one by one, we observe that process innovation is the innovation that resembles most of the innovations that we are going to implement in Paper Atlas. This will be a transformative innovation because we are creating an application that provides significant operational enhancement.

2. Current System

Currently there are different search applications and engines such as Google Scholar to find papers. However, these systems are mainly focused on finding a list of papers according to a searched keyword. Some of them provide finding papers that are cited by a specific paper or papers that cite a specific paper. However, the results of these searches are still a list of papers and finding the correct paper from a list can be hard. Paper Atlas will provide visualization of these reference relations and thanks to visualization, users can display deeper reference relations within a graph without confusing things.

Moreover, with the filtering, unnecessary papers can be discarded from the graph and a tidy graph visual can be obtained.

3. Proposed System

3.1 Overview

Paper Atlas will be a web application available on all modern web browsers. With Paper Atlas, users can visualize the hierarchy of research papers as a graph-based and interactive structure in order to find the most relevant paper to their topic. The goal of this project is to enhance the way academicians and students conduct their research by providing a better way to find papers that best match their research topic by showing the hierarchy of papers related to the input paper of the user. This application will show the papers as nodes with edges specifying which one references which one and which one is being referenced by which. The users will be able to read the details such as name, keywords, and abstract by clicking on the node of a paper. A link to the original paper will be available as well. The graph will be interactive; in other words, the users can move the nodes around, zoom in and out, apply layouts, and highlight nodes. Users can filter their results by time interval, keywords, author, and publication journal. Users can also search authors or keywords instead of paper titles. In this case, relative nodes will be shown for authors or keywords. The application will also provide extended features such as grouping of the nodes and ranking papers by their importance.

3.2 Functional Requirements

3.2.1 Viewing Papers as an Interactive Graph

The user can view research papers and their reference relations within a graph. Papers will be shown as circles which are called nodes. If a paper has reference to another paper, this reference relation will be indicated with an arrow between nodes of the papers. These arrows are called directed edges between nodes. It will be an interactive graph. It means the user can move the nodes of the paper and can change the appearance of the graph or apply a different layout to get a different appearance.

3.2.2 Viewing the Details of a Paper

The user can view details of a paper by clicking the node of the paper. The details are the title, authors, keywords, journal it was published and abstract of the paper. A link to the original paper will be given.

3.2.3 Viewing Papers of an Author

The user can choose an author and view the papers of the author and reference relations of the papers as a graph.

3.2.4 Viewing Papers and Related Keywords

Users can select some keywords and can see papers that are related with these as a graph visualization. Papers and keywords will be shown as nodes while the reference relation of the papers and which keywords belong to which paper will be shown as edges.

3.2.5 Grouping Papers Based on Journals

The papers can be grouped while preserving structure of the graph. They will be in the group of the journal they have been published in.

3.2.6 Searching by Different Parameters

The user can search authors, titles, keywords, journals or universities while specifying what kind of input they are providing.

3.2.7 Filtering Based On Different Values

The user can filter search results with different criteria. The user also can use these filters at the same time as filtering based on multiple values.

3.2.8 Filtering Output Based on Time Interval

The user can select a start time and an end time to obtain papers which are published between the chosen time interval. If the end date is not specified then the end date will be taken as today.

3.2.9 Ranking papers based on ‘Cited by’

The nodes that represent papers can be ranked by how many other papers have referenced this paper. The users have an option to enable or disable this feature. The ranking of nodes can be shown by the size of the nodes, by the color of the nodes or by a badge on the nodes.

3.2.10 Ranking papers based on ‘Cited’

The nodes that represent papers can be ranked by how many references it has or how strong are these references. The users have an option to enable or disable this feature. The ranking of nodes can be shown by the size of the nodes, by the color of the nodes or by a badge on the nodes.

3.2.11 Find Common References of given papers

The users can search more than one research paper and find common reference papers of these papers. The user can also specify to look down to more than one level of references. For example, if a user chooses max level as 2 then they will see a common reference of the references of the papers they have input if exists; common reference of references.

3.2.12 Displaying a Graph of Authors

The users can view a graph with nodes representing authors. This will be a directed graph. An edge from Author A to Author B will show how many references are made between authors. The users can view the list of references from Author A to Author B by clicking on the related edge.

3.3 Non-functional Requirements

3.3.1 Performance

The response time for the application besides running layout and retrieving a graph from the database should not be more than 30 milliseconds. But since a graph can be very large, running layouts and retrieving such graphs from the database can take longer but should not exceed 1 minute.

3.3.2 Usability

The user interface of the application should be user friendly. Everyone using the application should be able to understand how to handle functionalities of the application. As the application is based on graph visualization, graphs should have interactive design for everyone to use it easily.

3.3.3 Maintainability

As the initial goal of the application is to provide services for a specific topic, the application should be able to continue to function if it is decided to provide services for other topics. In addition, information that the application provides should be up to date.

3.3.4 Scalability

The application must be able to handle 10000 users at the same time.

3.4 Pseudo Requirements

3.4.1 Implementation Constraints

Node.js framework will be used for the implementation of backend with Javascript. Cytoscape.js library will be used for visualization of the graphs. Neo4j which is a graph database will be used to store information. Cypher will be used for graph query language. Response time will be less than two minutes.

3.4.2 Data Constraints

Publicly published papers will be used as data. Publicly accessible information of papers such as author, title and abstract will be stored and used.

3.4.3 Cost Constraints

Open API's will be used to gather papers. Google Scholar and other free websites will be used to gather data.

3.4.4 License Constraints

For all the used libraries, APIs and frameworks, if their licenses are appropriate for the project, they will be used.

3.4.5 Schedule Constraints

Before the first semester ends, a prototype demo will be done. Before the CS Fair 2023 the project will be done.

3.5 System Models

3.5.1 Scenarios

Scenario 1: Using Keyword Search

Zehra starts to do a research that her manager gives. The topic is Speech to Text (STT). Zehra enters the Paper Atlas and chooses graph type as keyword and paper. She selects the keyword to do the search and enters “speech to text”. She also uses time filters to find papers that have been published in the last 5 years. When she looks at the papers that are most cited she recognizes that the keyword CNN belongs most to them. She chooses one of the most cited research with the CNN keyword and starts to read it. The paper uses the terms like CNN and RCNN but does not have a detailed explanation of them. Zehra uses Paper Atlas and chooses again keyword and paper graph type. This time, she chooses paper as the search type and enters the title of the paper. By using filters, she filters keywords as “STT and CNN and RCNN” included. This time she is able to see referred papers that include these terms. She looks at the most cited ones among them and by reading these papers she can learn more about the terms she did not know before. After that she can come back to the first paper and can understand the paper more easily.

Scenario 2: Finding Common Referenced Papers

Akın who does research about web obfuscation methods finds three different methods GooPIR, TrackMeNot and Naive Query Injection. He also finds three papers that use these three methods. However, Akın needs to improve these methods. He thinks of merging these three methods into one. Therefore, he needs to find a way to merge them. He chooses graph type as Graph of Common References. He enters the three papers and looks whether there are any papers referring to three of them. From the graph, he can find if the three of them are used in one paper or maybe two of them are used in a paper and another paper uses that one with the remaining one of the three papers. He exports the graph and sends it to his group so that others can also look at the graph and read the necessary papers.

Scenario 3: Using Author Search

Hakan does research about a political topic. He wants to make sure the papers he finds are not biased. He uses Paper Atlas and chooses the graph type as author graph. He searches for the author of papers he found. In this way he can see if some authors refer to only within a group of authors. He also chooses papers grouped by journals graph type and searches the papers that he found. He looks at whether the journals of the referenced papers include the ones accepted as trustworthy globally or only some small size journals. He can use these visuals to decide how much a paper may be accepted globally or within an internal group.

3.5.2 Use-Case Model

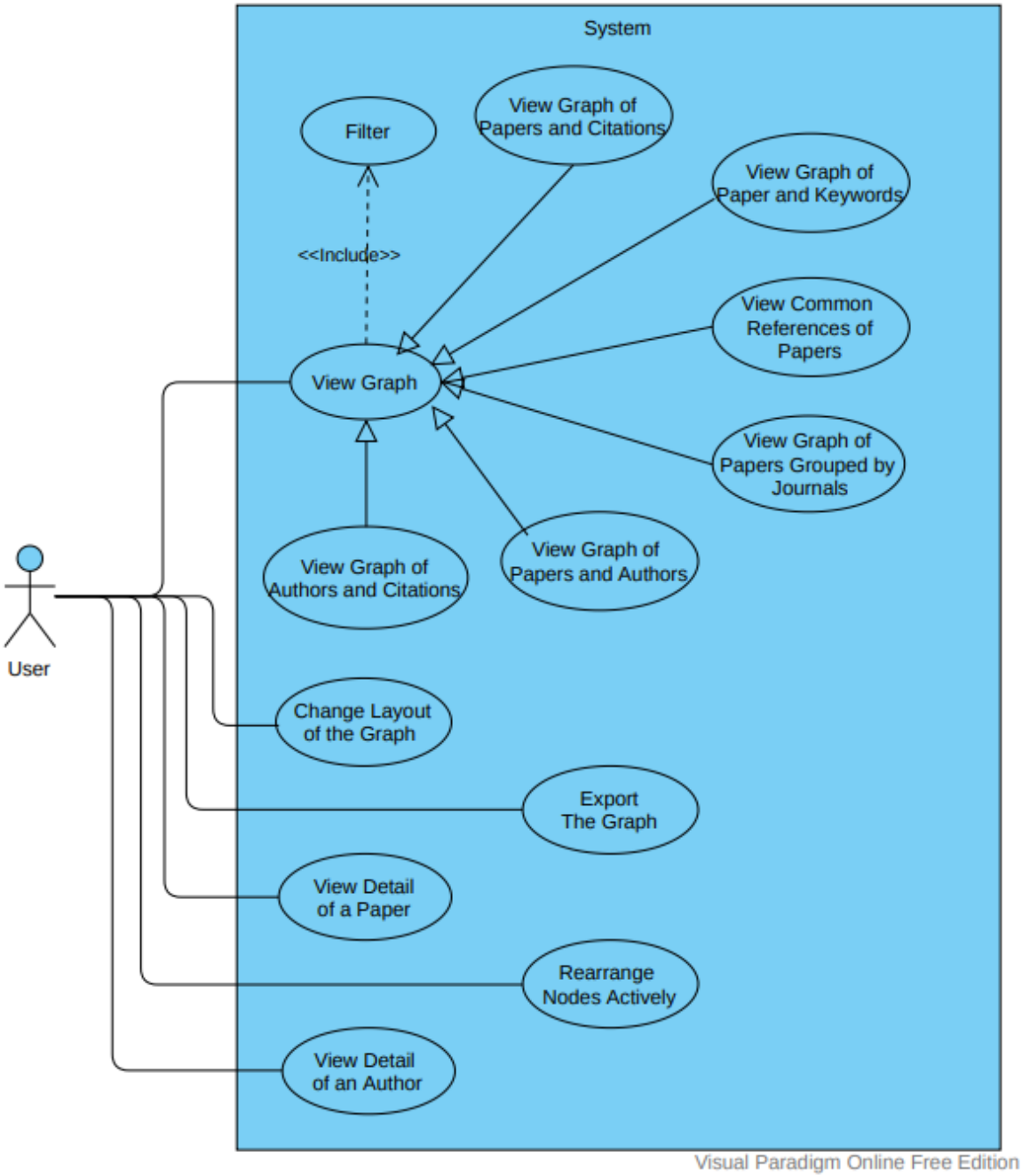

Figure 1: Use Case Diagram

1. **Name**: View Graph
2. **Participating actor**: User
3. **Entry condition:**
    a. The user enters the Paper Atlas
4. **Exit condition**:
    a. The user cancels the process, OR
    b. The users closes the Paper Atlas, OR
    c. View Graph is done successfully.
5. **Flow of events:**
    a. The user selects the type of graph which s/he wants to view
6. **Quality requirements:**
    a. At any point, View Graph use case can include Filter use case

1. **Name**: View Graph of Papers and Citations
2. **Participating actor**: Inherited from View Graph use case
3. **Entry condition:** Inherited from View Graph use case
4. **Exit condition**: Inherited from View Graph use case
5. **Flow of events:**
    a. The user selects graph type as Papers and Citations.
    b. The user searches a paper for it to be the starting node.
    c. The user selects the layer (distance) number for maximum distance selected papers and.reached papers.
    d. Paper Atlas brings the graph of papers and citations relations in wanted depth.
6. **Quality requirements:**
    a. The user should select a valid paper.
    b. The user should select layer number smaller than 6.

1. **Name**: View Graph of Papers and Authors
2. **Participating actor**: Inherited from View Graph use case
3. **Entry condition:** Inherited from View Graph use case
4. **Exit condition**: Inherited from View Graph use case
5. **Flow of events:**
    a. The user selects graph type as Papers and Authors.
    b. The user selects the search type as author or paper.
    c. The user searches for selected keywords (paper or authors).
    d. Paper Atlas brings the graph of papers and authors according to input.
6. **Quality requirements:**
    a. The user should give input at least 1 at most 5 authors OR should give input at least 1 at most 5 papers.

1. **Name**: View Graph of Common References of Papers
2. **Participating actor**: Inherited from View Graph use case
3. **Entry condition:** Inherited from View Graph use case
4. **Exit condition**: Inherited from View Graph use case
5. **Flow of events:**
    a. The user selects graph type as Graph of Common References of Papers
    b. The user searches for more than one paper.
    c. Paper Atlas brings the graph of selected papers and graph of their common references and papers that reference all of them.
6. **Quality requirements:**
    a. The user should select at least 1 at most 10 papers.

1. **Name**: View Graph of Authors and Citations
2. **Participating actor**: Inherited from View Graph use case
3. **Entry condition:** Inherited from View Graph use case
4. **Exit condition**: Inherited from View Graph use case
5. **Flow of events:**
    a. The user selects graph type as Graph of Authors and Citations
    b. The user searches authors.
    c. Paper Atlas brings the graph of selected authors and graph of their citation relations.
6. **Quality requirements:**
    a. The user should select at least 1 at most 20 authors.

1. **Name**: View Graph of Papers and Keywords
2. **Participating actor**: Inherited from View Graph use case
3. **Entry condition:** Inherited from View Graph use case
4. **Exit condition**: Inherited from View Graph use case
5. **Flow of events:**
    a. The user selects graph type as Graph of Papers and Keywords
    b. The user selects the search type as paper or keyword.
    c. The user searches for selected keywords (paper or keyword).
    d. Paper Atlas brings the graph of papers and keywords according to input.
6. **Quality requirements:**
    a. The user should select at least 1 keywords OR at least 1 paper

1. **Name**: View Graph of Papers Grouped By Journals
2. **Participating actor**: Inherited from View Graph use case
3. **Entry condition:** Inherited from View Graph use case
4. **Exit condition**: Inherited from View Graph use case
5. **Flow of events:**
    a. The user selects graph type as Graph of Papers Grouped By Journals
    b. The user selects the search type as papers or journals.
    c. The user searches for selected keywords (paper or journal).
    d. Paper Atlas brings the graph of papers and journals according to input.
6. **Quality requirements:**
    a. The user should select at least 1 journal.OR at least 1 paper

1. **Name:** Filter
2. **Participating actor:** User
3. **Entry condition:**
    a. The user views a graph
4. **Exit condition:**
    a. The user cancels the process, OR
    b. The users closes the Paper Atlas, OR
    c. The filtering is done successfully
5. **Flow of events:**
    a. The user chooses different filters
    b. The users add more filters with “and” or “or” conjunctions
    c. Paper atlas applies the filter to search ot the viewed graph
6. **Quality requirements** None

1. **Name:** Change Layout of the Graph
2. **Participating actor:** User
3. **Entry condition:**
    a. The user views a graph
4. **Exit condition:**
    a. The user cancels the process, OR
    b. The users closes the Paper Atlas, OR
    c. The layout changes successfully
5. **Flow of events:**
    a. The user choose one of the predefined layout type
    b. Paper Atlas changes the view of the graph according the chosen layout
6. **Quality requirements** None

1. **Name:** Rearrange Nodes Actively
2. **Participating actor:** User
3. **Entry condition:**
    a. The user views a graph
4. **Exit condition:**
    a. The users closes the Paper Atlas, OR
    b. The place of node changes successfully
5. **Flow of events:**
    a. The user choose one of the nodes
    b. The user moves the chosen nodes and replace it another place
    c. Paper Atlas shows the nodes at new place
6. **Quality requirements** None

1. **Name:** Export the Graph
2. **Participating actor:** User
3. **Entry condition:**
    a. The user views a graph
4. **Exit condition:**
    a. The user cancels the process, OR
    b. The users closes the Paper Atlas, OR
    c. The graph is exported successfully
5. **Flow of events:**
    a. The user chooses to export graph
    b. Paper Atlas asks to export şs as pdf or png
    c. The users choose one of formats
    d. Paper Atlas exports the graph as the chosen format
6. **Quality requirements** None

1. **Name**: View Details of a Paper
2. **Participating actor**: User
3. **Entry condition:**
    a. The user is currently in a graph with paper nodes
4. **Exit condition**:
    a. The user cancels the process, OR
    b. The users closes the Paper Atlas, OR
    c. View Details of a Paper is done successfully.
5. **Flow of events:**
    a. User selects a paper node from the graph
    b. Paper Atlas brings the details of selected paper

6. **Quality requirements:** None

1. **Name**: View Details of a Author
2. **Participating actor**: User
3. **Entry condition:**
    a. The user is currently in a graph with author nodes
4. **Exit condition**:
    a. The user cancels the process, OR
    b. The users closes the Paper Atlas, OR
    c. View Details of an Author is done successfully.
5. **Flow of events:**
    a. User selects a author node from the graph
    b. Paper Atlas brings the details of selected author
6. **Quality requirements:** None

3.5.3 Object and Class Model

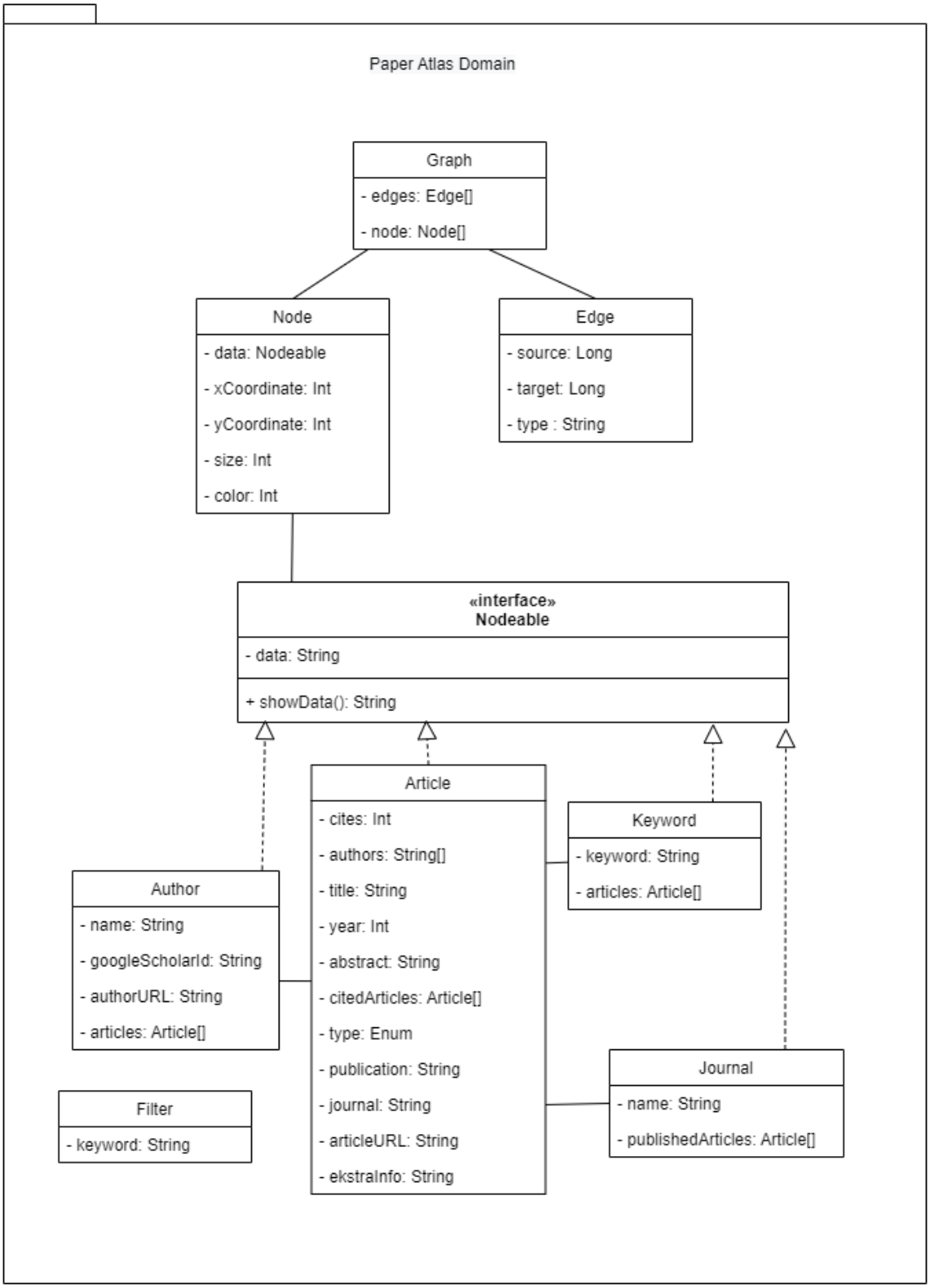

Figure 2: Class Diagram

3.5.4 Dynamic Models

3.5.4.1 Activity Diagram

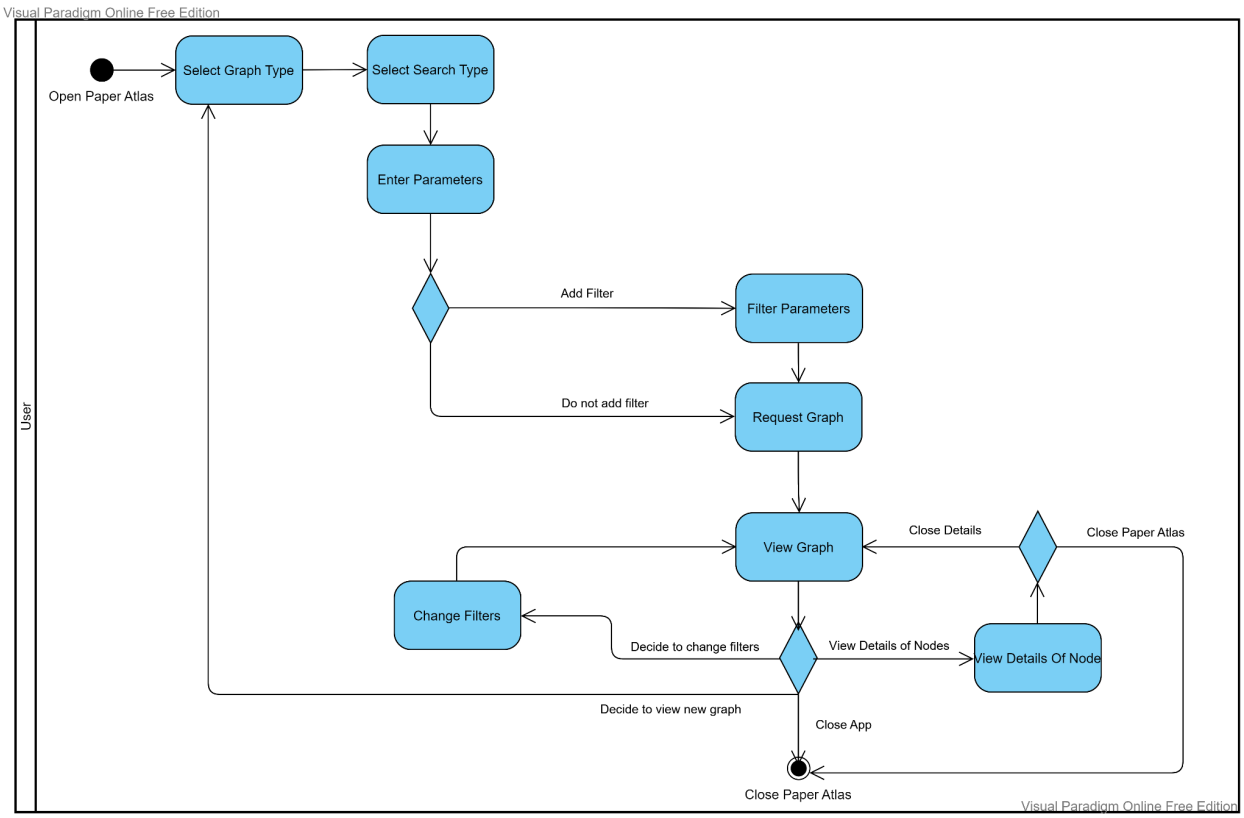

Figure 3: Activity Diagram to View Graph

3.5.4.2 Sequence Diagrams

Creating Graph of Citations:

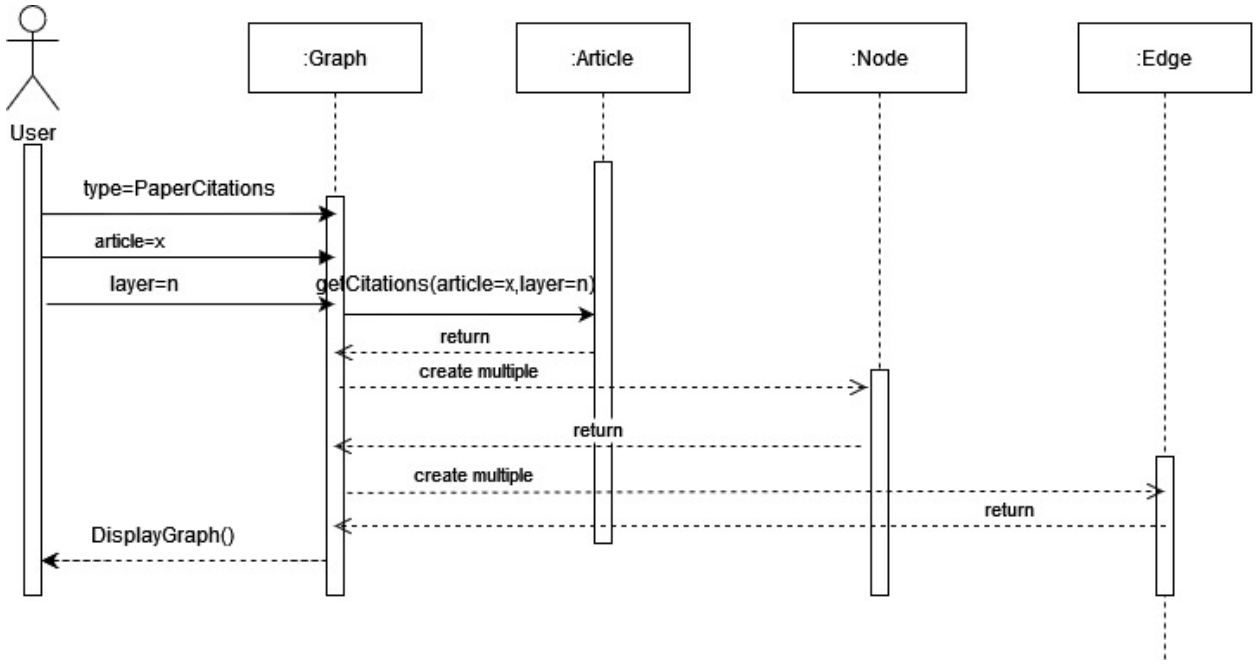

Figure 4: Creating Graph of Citations

Creating Graph of Articles and Authors:

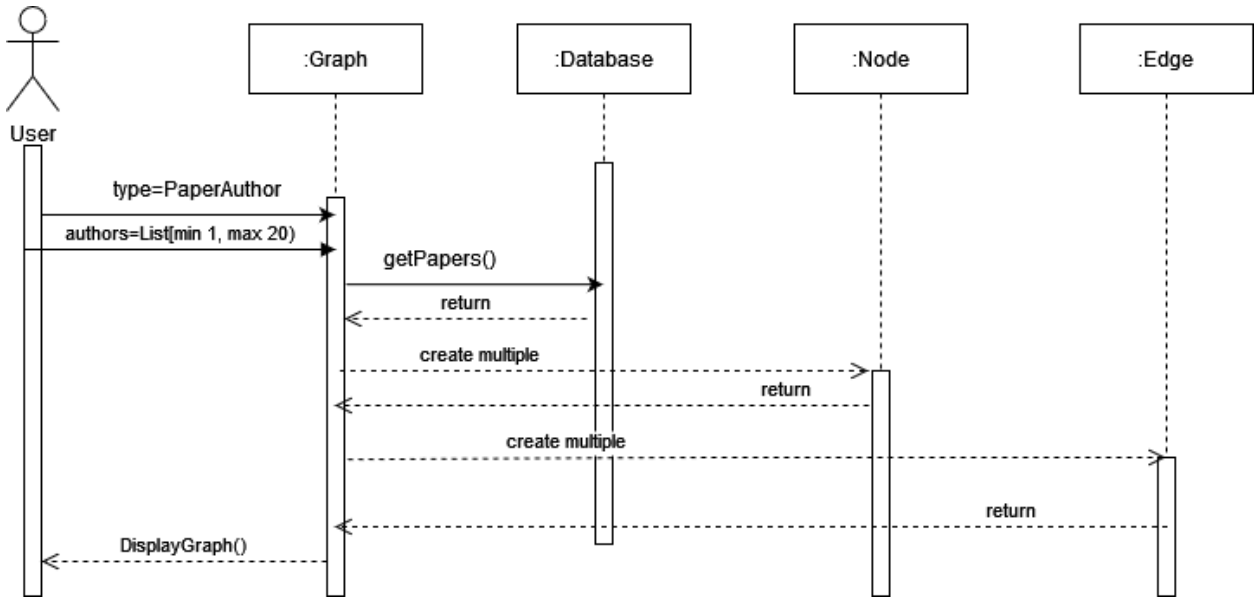

Figure 5: Creating Graph of Articles and Authors

Creating Graph of Keywords and Articles:

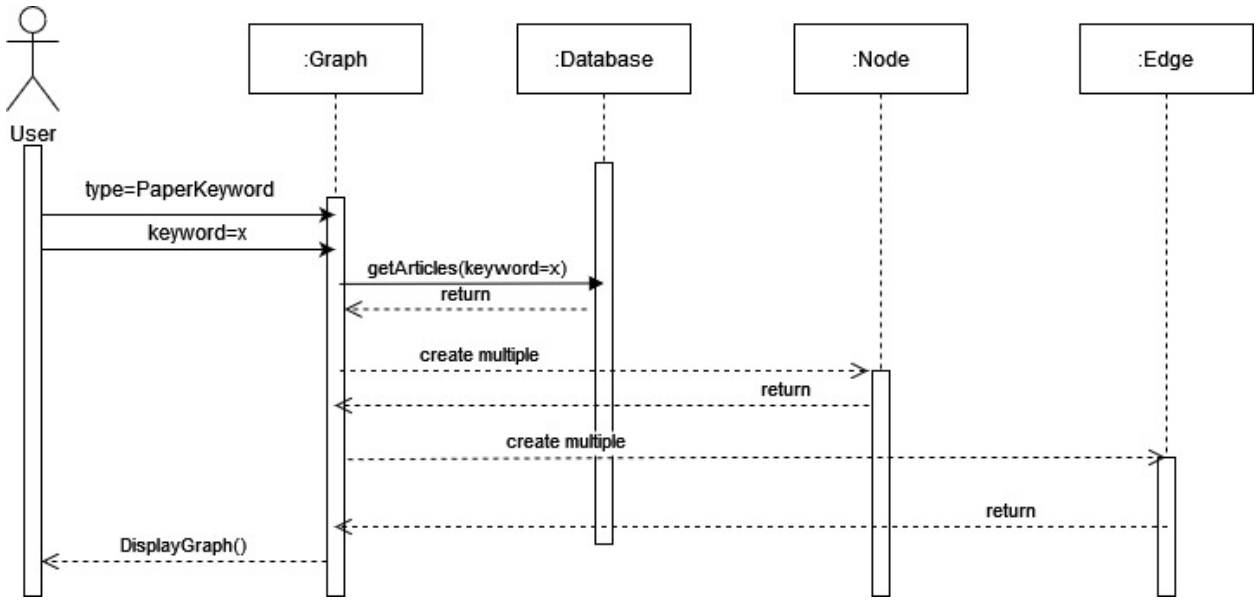

Figure 6: Creating Graph of Keywords and Articles

Creating Graph of Papers and Common Citations:

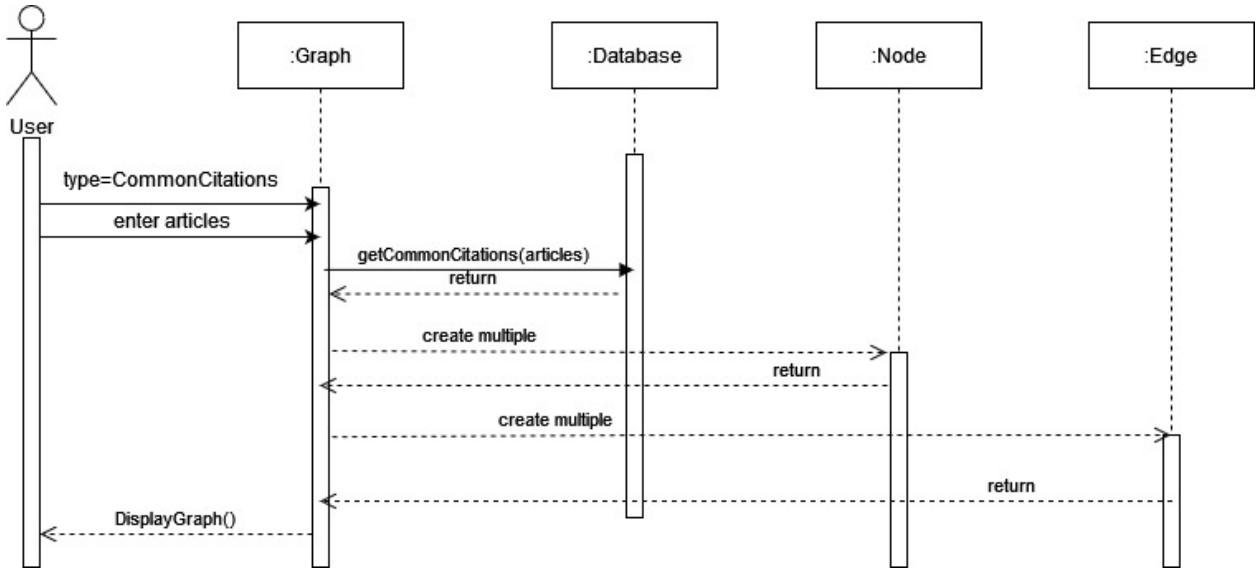

Figure 7: Creating Graph of Papers and Common Citations

Creating Graph of Papers Grouped by a Journal:

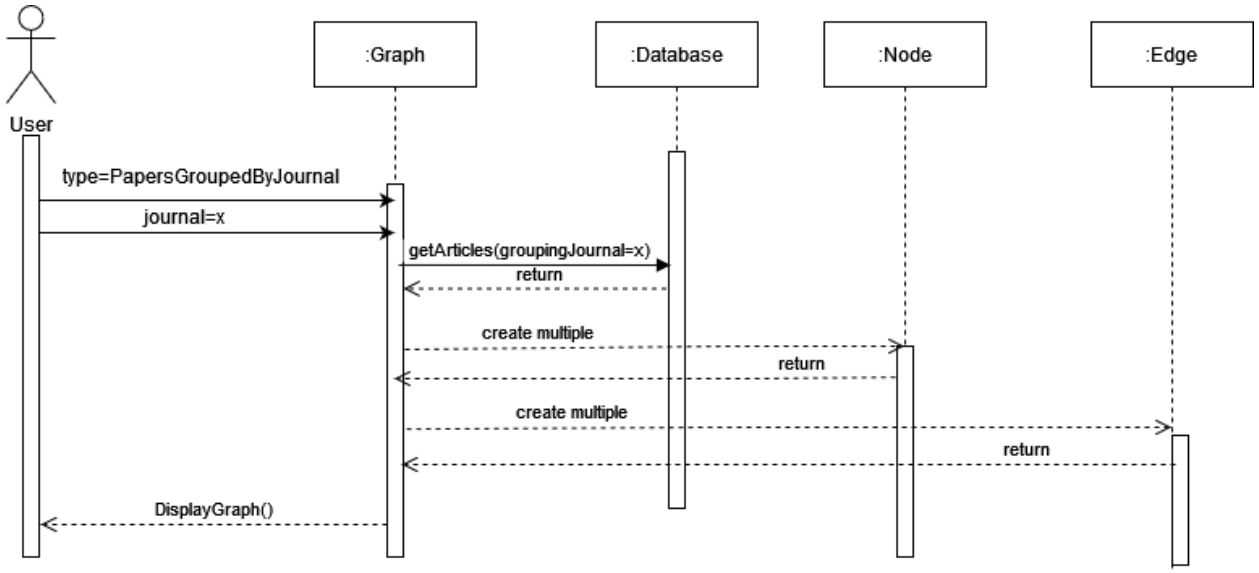

Figure 8: Creating Graph of Papers by a Journal

Creating Graph of Authors and Citations

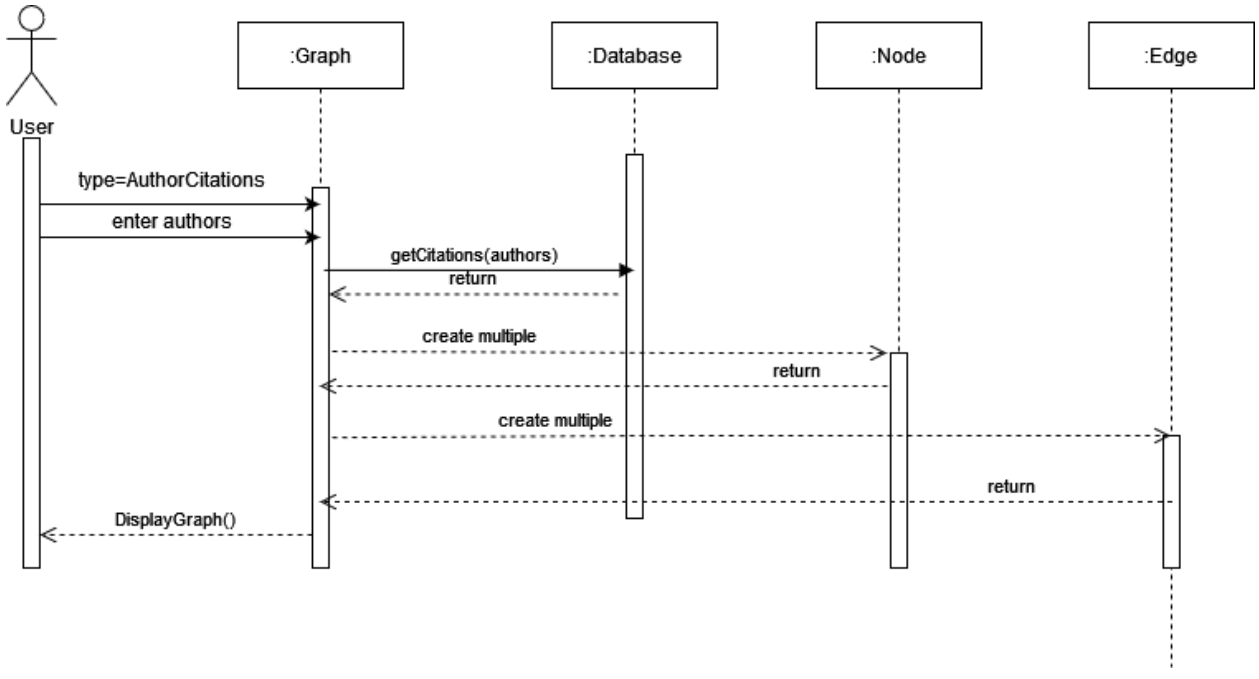

Figure 9: Creating Graph of Authors and Citations

Searching by Title

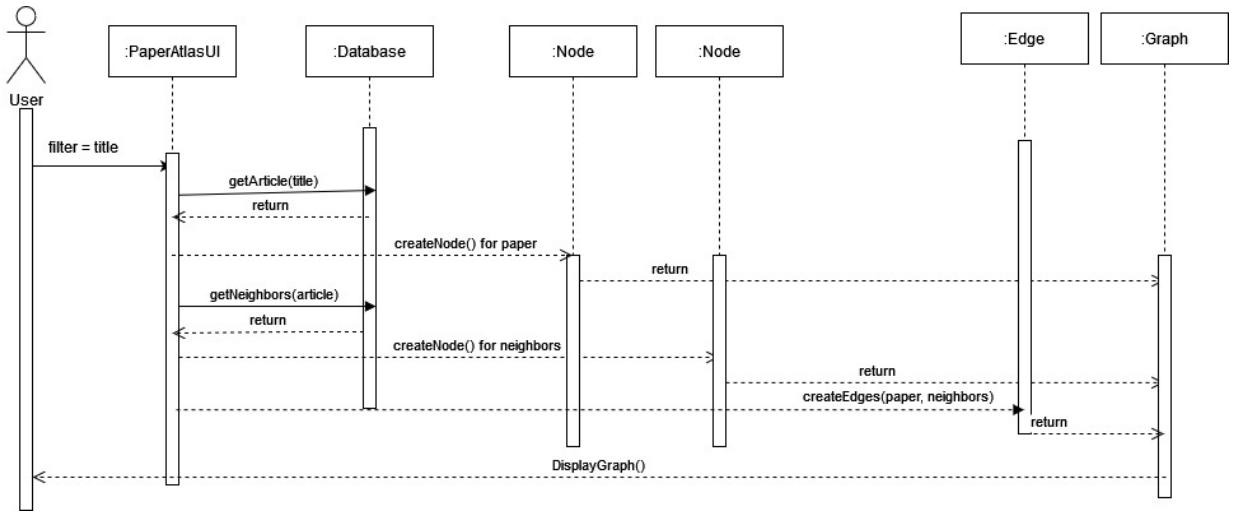

Figure 10: Searching by Title

3.5.5 User Interface

Users can search keywords in the Paper Atlas. They can do so by clicking on the button “Keyword” and write what they are searching in the search bar. The resulting graph will have the keyword as a node and the articles as nodes as well.

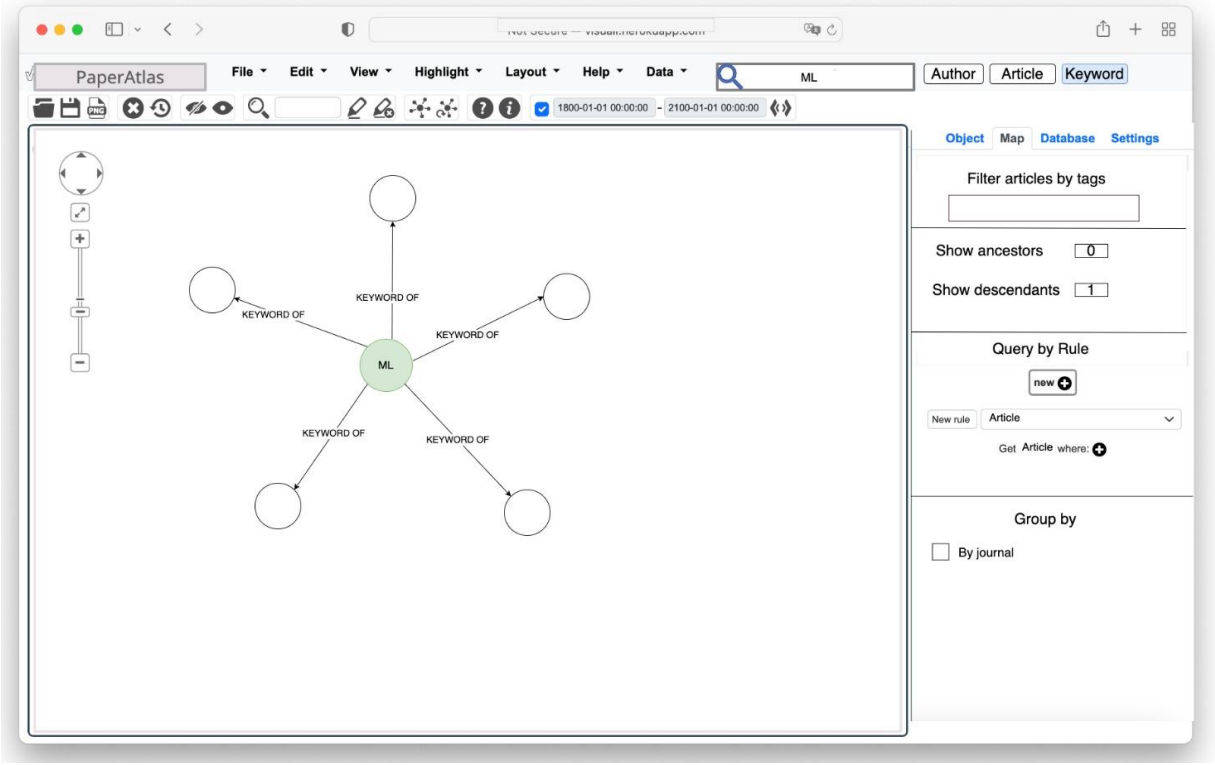

Figure 11: Searching for keywords

The user can filter the output graph by keywords. The user can do this by writing the keywords in the “Filter articles by tags” section of the “Map” panel. In this case the output graph will only show articles associated with the given keywords. The user can also choose how many ancestors, or descendants will be shown for the given input. The user can also run some user defined queries. The user can input their own queries in the “Query by Rule” section in the “Map” panel. Through this panel they can also choose to group the resulting articles by journal.

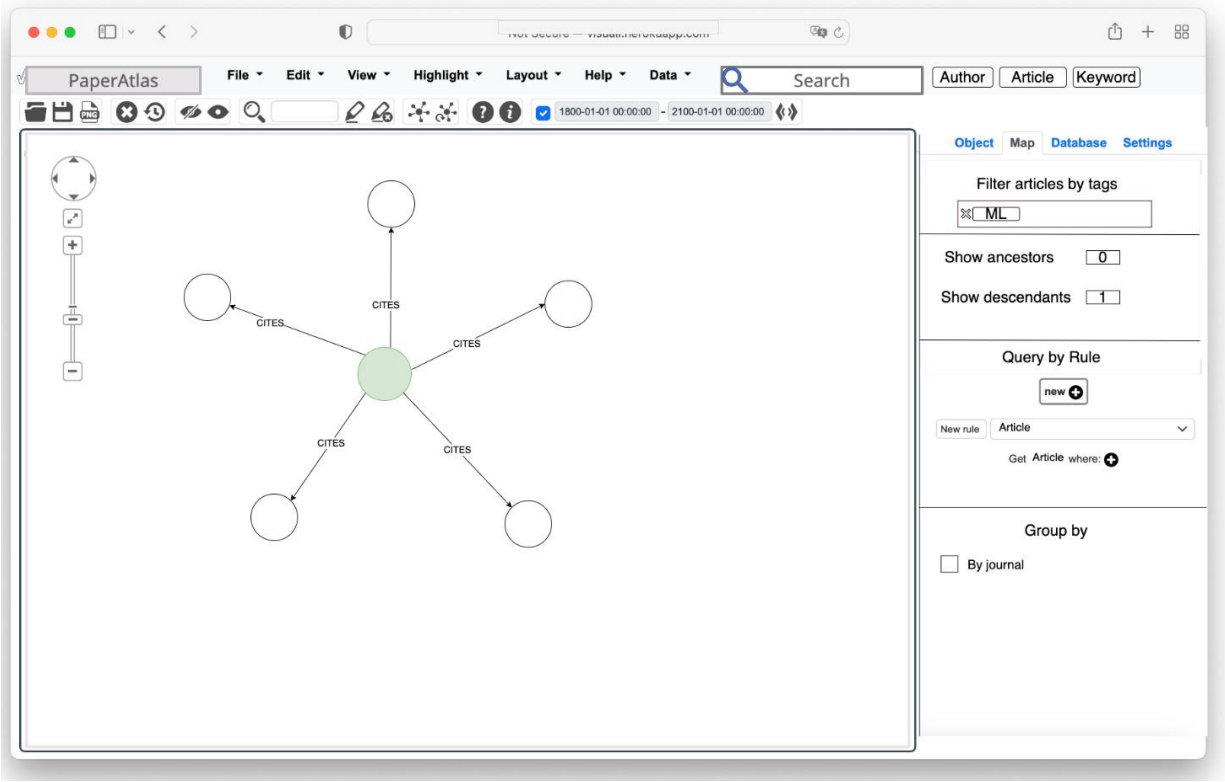

Figure 12: Searching for articles

The users can view the details of an article or an author by clicking on the article and author node in the graph. The details will be shown in the “Object” panel. The user can also search articles. To do so the user will click on the button “Article” and type the name of the article in the search bar.

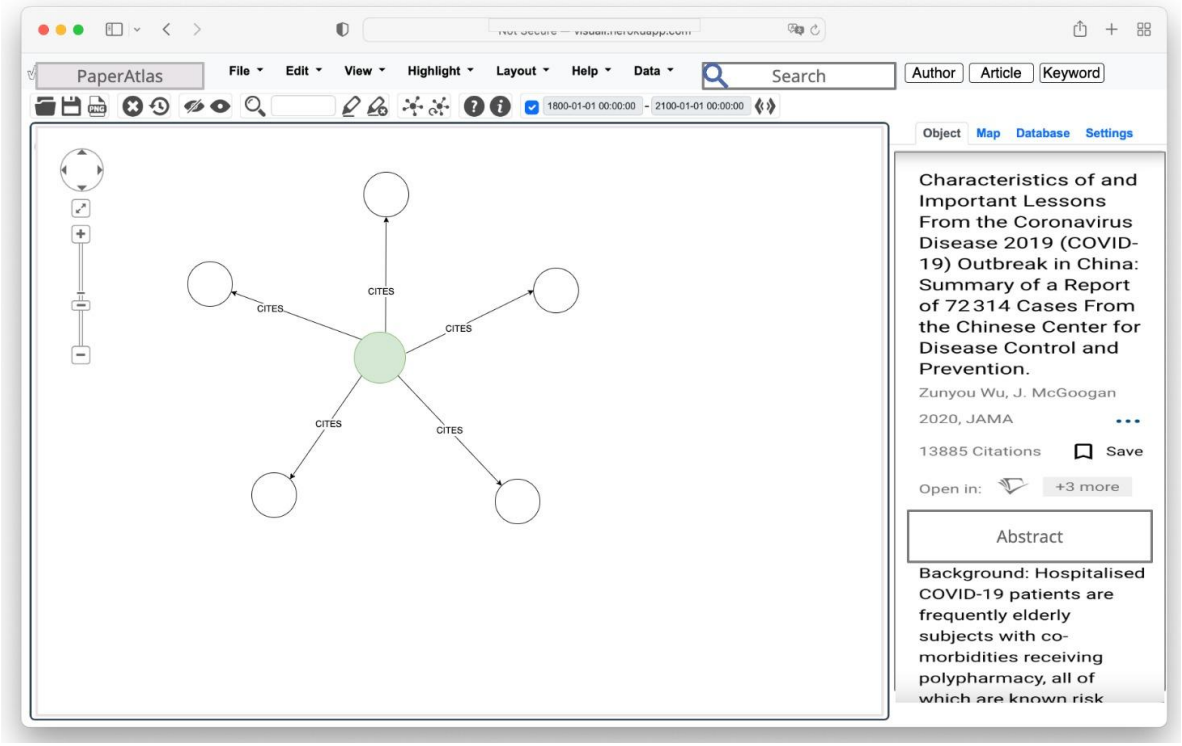

Figure 13: Seeing details of article

The user can also search for authors. To do so they will click on the button “Author” and then type the author’s name in the search bar. In the resulting graph the author will also be shown as a node along with the articles.

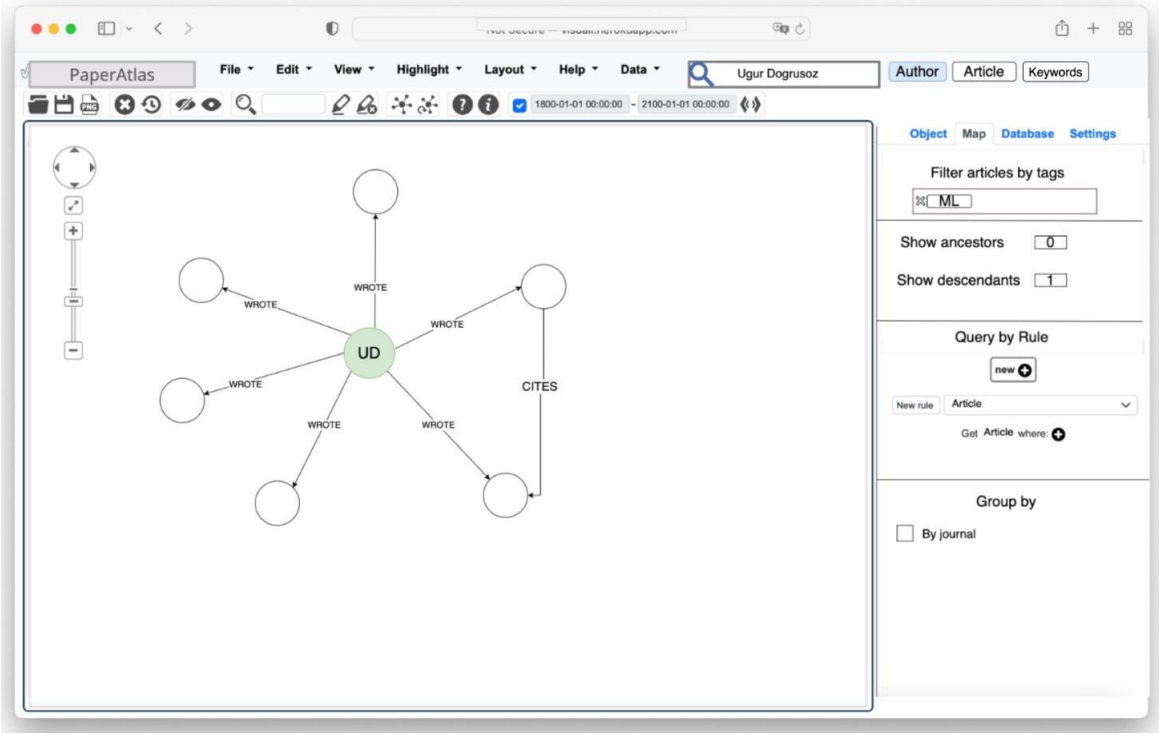

Figure 14: Searching author

4. Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

Public Health: PaperAtlas may help medical researchers make more efficient research and thus contribute to the overall public health.
Public Safety: PaperAtlas does not directly affect the safety of its users.
Public Welfare: PaperAtlas does not directly affect the welfare of its users.
Cultural Factors: PaperAtlas may affect the privacy of researchers by making it easier for adversaries to collect data about the researcher’s career. These privacy issues should be taken into consideration while building this application.
Social Factors: PaperAtlas may reveal academic groups that gain credit by only citing each other's papers. These groups are also known as academic clans. PaperAtlas may put the social status of such academics at risk.
Economic: PaperAtlas does not directly affect the economical situation of its users.

Global Factors: PaperAtlas is planned to be a software used by people all around the world. However, people speak many different languages all around the globe. Considering that most people that browse the internet speak English, having the interface in English will allow people from different countries to access the software.
Environmental: PaperAtlas will be a web application, and will not have a direct effect on the environment.

Table 1: Factors that can affect analysis and design

	Effect level	Effect
Public health	2	Can help medical research
Public safety	0	-
Public welfare	0	-
Global factors	2	Interface should be in English
Cultural factors	5	Privacy and Data Protection
Social factors	8	Risk of exposing academic dishonesty

4.2 Risks and Alternatives

The first risk is directly related to the core of the project. The main purpose of the project is to make researchers' work easier when they need to examine too many papers. The proposed solution to this problem is drawing a proper and neat graph. However, if the graph has too many nodes or edges, it leads to a more complicated graph which is contrary to the purpose of the project. To prevent this situation, the number of nodes or edges will be limited or some nodes or edges can be eliminated before showing the graph.

Another risk is finding an appropriate data source. We are planning to use an API. However, if the fetched data does not meet the requirements (inconsistent or missing fields in citations, journals, or author data) or this API has too many restrictions about request limits, the API will not be useful anymore. The alternative solution is finding an alternative API or downloading data using different tools such as Publish or Perish.

The last risk is related to storing data. If the database has too much data, searching or insertion operations will become slower, and maintaining the database will be a problem. To prevent this, we will optimize the database with different methods and in the worst-case scenario, we will replace the database with a new database that has automatic load balancing.

Table 2: Risks

	Likelihood	Effect on the project	B Plan Summary
Complex Graphs	High	Untidy and useless graph	- Adding limitations for maximum number of edges or node - Elimination of some edges or
Fetching data	Low	Inconsistent or missing data	Fetching data with alternative ways or manually
Storing too much data	Low	Difficult to maintain the database	Database optimization or changing database

4.3 Project Plan

In this project, Software Development Life Cycle will be used. Because of that, firstly requirements of the system should be decided. The system should be analyzed and some major parts should be decided such as how a user will interact with the system and which entities will be used to represent real life objects. These will be done in the Analysis and Requirement step. After that low and high level design of the projects should be decided. In this step, subsystems and components of the project should be decided. Which hardwares will be used should be decided. The solutions that include design patterns for complex software engineering should be discussed at this level as well. Implementation of the project should be done after the design step. However, in this project a small prototype will be implemented between analysis and design steps to ensure that the project can be implemented and finished on time.

Table 3: Summary of Work Packages

WP#	Work package title	Leader	Members involved
WP1	Project Specification Report	Zehra Erdem	All Members
WP2	Analysis and Requirement Report	Zehra Erdem	All Members
WP3	Collecting Data	Akın Kutlu	All Members
WP4	First Demo	Selbi Ereshova	All Members
WP5	Detailed Design Report	Ahmet Hakan Yılmaz	All Members
WP6	Final Demo	Selbi Ereshova	All Members
WP7	Final Report	Aybala Karakaya	All Members

Table 4: List of Work Packages

WP 1: *Project Specification Report*			
Start date: 01.10.2022 **End date:** 17.10.2022			
Leader:	Zehra Erdem	**Members involved:**	All Members
Objectives: *Explaining the project and deciding on requirements*			
Tasks: ***Task 1.1 Requirements*:** Deciding the functional and non functional requirements ***Task 1.2 Constraints*:** Deciding the constraint of the project ***Task 1.3 Possible Issues:*** *Looking at possible ethical and professional issues and how they can be handled.* ***Task 1.4 Risk Management:*** *Analyzing possible risks*			
Deliverables ***D1.1: Project Specification Report*** ***D1.2: Innovation Form***			

WP 2: *Analysis and Requirements Report*			
Start date: 18.10.2022 **End date:** 13.11.2022			
Leader:	Zehra Erdem	**Members involved:**	All Members
Objectives: *Analyzing the real life scenarios and deciding the details of the system according to them.*			

Tasks:

***Task 2.1 Requirements*:** Review the functional, non functional requriments of the system

***Task 2.2 Scnreios*:** Writing possible real life scenarios

Task 2.3 Use Case Diagrams: *Extracting use cases from the scenarios and requirements. Drawing the diagram of them.*

Task 2.4 UML Class Diagrams: *Deciding of the class diagram to represent real life entities and their relation*

Task 2.5 Dynamic Diagrams: *Drawing dynamic diagrams to show how system will work*

Task 2.6 Mock Ups: Drawing mock ups for user interface

Task 2.7 Risk Management: *Analyzing possible risks and how they can be handled*

Task 2.8 Planning the Project: *Planning milestones of the project*

Deliverables

D2.1: *Analysis and Requirements Report*

WP 3: *Collecting Data*			
Start date: 14.11.2022 **End date:** 30.11.2022			
Leader:	Akın Kutlu	**Members involved:**	All Members

Objectives: *Finding real life data and storing them*

Tasks:

***Task 3.1 Database*:** Creating a database to store data

***Task 3.2 APIs*:** Finding APIs that can be useful and analyzing them

Task 3.3 Storing Data: *Writing a script to send requests to APIs and storing the returned responses in the Database*

Deliverables

WP 4: *First Demo*			
Start date: 14.11.2022 **End date:** 14.12.2022			
Leader:	Selbi Ereshova	**Members involved:**	All Members

Objectives: *Starting implementation of the project and finishing some functions of it as demo*

Tasks:

***Task 4.1 Building*:** Building frontend and backend. Connecting backend to database

***Task 4.2 Backend*:** Implementing backend so that search and some filters work, returning the graphs for papers and citations work

Task 4.3 Frontend: *Implementing frontend for the functions backend provides*

Task 4.4 Presentation: *Preparing a presentation and being prepared to present both the demo and the presentation.*

Deliverables

D4.1: *Codes of the first demo*

D4.2: *Presentation*

WP 5: *Detailed Design Report*			
Start date: 30.01.2023 **End date:** 19.02.2022			
Leader:	Ahmet Hakan Yılmaz	**Members involved:**	All Members

Objectives: *Planning the high and low level design of the project*

Tasks:

***Task 5.1 Subsystems*:** Designing subsystems and their connections to each other

***Task 5.2 Database*:** Determining persistent data objects and their management

Task 5.3 Security: *Determining the authentications and authorizations.*

Task 5.4 Deployment: *Designing to how to deploy the project*

Deliverables

D5.1: *Detailed Design Report*

WP 6: *Final Demo*

Start date: 30.01.2023 **End date:** 01.05.2023

Leader:	Selbi Ereshova	**Members involved:**	All Members

Objectives: *Finishing the implementation of the project and present it*

Tasks:

***Task 7.1 Requirements*:** Finishing all functional requirements.

***Task 7.2 Test*:** *Testing all functions.*

Task 7.3 Debugging: *Finding bugs and fixing them*

Task 7.4 Presentation: *Preparing a presentation and being prepared to present both the final demo and the final presentation.*

Deliverables

D7.1: *Codes of the final demo*

D7.1: *Final Presentation*

WP 7: *Final Report*			
Start date: 01.04.2023 **End date:** 30.04.2023			
Leader:	Aybala Karakaya	**Members involved:**	All Members
Objectives: *Reporting the final point of the project and preparing user manual*			
Tasks: ***Task 6.1 Requirements*:** Reporting the matched and unmatched functional requirements and if there are new requirements reporting them as well. ***Task 6.2 Building and Deployment*:** *Reporting how the project can be built and run. Explaining if it were deployed, how it should be done and how it should be maintained.* ***Task 6.3 User Manual:*** *Preparing a user manual*			
Deliverables ***D6.1:*** *Final Report*			

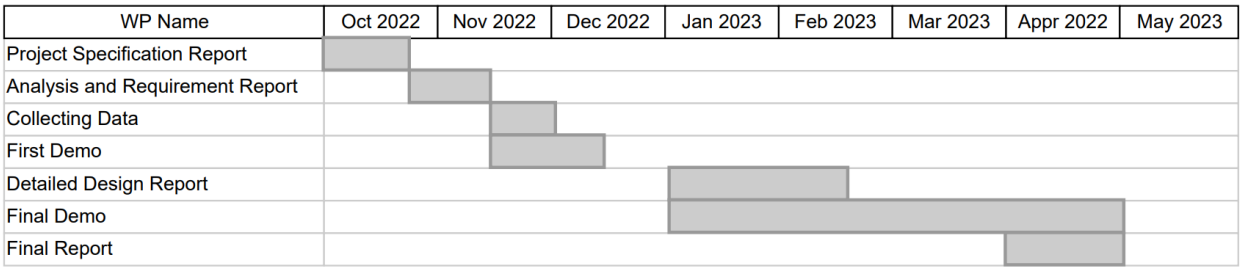

Figure 15: Gantt Chart for Project Plan

4.4 Ensuring Proper Teamwork

We are aware of the importance and benefits of teamwork. Therefore, we decided that each of the leaders of the work package is a different person according to their choices, and all team members will participate in all different work packages. We are expecting that everyone is aware of the progress of each step. Also, to follow the progress, we will communicate with each other constantly via Whatsapp. For necessary times, we will use Zoom meetings. To monitor coding progress, GitHub statistics will be used.

4.5 Ethics and Professional Responsibilities

Paper Atlas will not violate any matter of Code of Ethics and Professional Conduct (ACM). Paper Atlas will not be doing any tracking of user data or it does not require an account to be used. Therefore it does not hold any personal data of users and there is not a third party company which gets data from Paper Atlas. For users it is free to use. Paper Atlas will get its data from publicly open sources (at initial step from Google Scholar). So as Paper Atlas will collect the data from publicly open sources it will not include any personal data in it. Paper Atlas will follow the principle of “Right to be Forgotten” by removing the wanted Papers of an author in case it is removed from Google Scholar and an authorized person for the paper wants it to be removed.

4.6 Planning for New Knowledge and Learning Strategies

While developing Paper Atlas we will learn and apply different technologies or frameworks which are new to us. Some of the frameworks we will have experience but we will gain more experience using Javascript, Typescript and NodeJs. We will also use Express.js within our Node.js application.

Most of the team does not know about graph databases or libraries. So we will learn about a graph based database and libraries that can help us in graph visualization .We will use Neo4j as a database management system which is a Graph Data Platform. Neo4j is a noSQL database system and we will learn Cypher which is the language we will be using for the database. Another framework related to graphs we will be using is Cytoscape.js which allows us to work on graphs. One of our team members has used Neo4j and Cytoscape.js. We will use online resources and her knowledge to learn about them. and We will also use APIs for fetching our data from Google Scholar. There are few options for those APIs and we will decide the optimal one by experimenting with our options.

5. Discussion

As an improvement, adding user accounts and keeping track of the search history of users can be implemented. Having user accounts also would make it possible to share the graphs among the users. However, during this implementation, the written functional requirements in this report will be priority. If there is remaining time, the team can work on the improvements or they can be added in the future.

6. References

[1] M. Gusenbauer and N. R. Haddaway, “Which academic search systems are suitable for systematic reviews or Meta‐analyses? evaluating retrieval qualities of google scholar, pubmed, and 26 other resources,” Research Synthesis Methods, vol. 11, no. 2, pp. 181–217, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/jrsm.1378. [Accessed: 10-Oct-2022].

[2] A. Martín-Martín, E. Orduna-Malea, M. Thelwall, and E. D. López-Cózar, “Google scholar, web of science, and Scopus: A systematic comparison of citations in 252 subject categories,” Journal of Informetrics, 05-Oct-2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1751157718303249. [Accessed: 10-Oct-2022].