National University of Singapore
Institute of System Science

Lemon Academic Knowledge Platform

System Documentation

Group 10

BIAN Weizhen A0285814W

Contents

List of Tables

List of Figures

1 Introduction

1.1 Executive Summary

Ph.D. students face challenges in navigating the vast world of scholarly articles. While tools like Google Scholar offer valuable resources, they have inherent limitations.

To address these challenges, I developed a cutting-edge academic search engine using Java and SpringBoot. This engine is specially designed for students preparing to study for PhD, bridging the gap in existing tools and providing a more personalized design and efficient search experience. It provides users with curated article suggestions, search enhancements, visual representations of keyword trends, and user-friendly features like saving articles and tracking browsing history. These integrated functionalities ensure users have a comprehensive overview of their academic journey while allowing the system to refine its user-centric recommendations.

Drawing from the methodologies and principles I acquired during coursework, I designed the engine around fundamental tenets: data structuring, algorithm optimization, and user-centric design. The combination of these principles ensured the platform was both efficient and user-friendly.

This project was more than just technical development; it was a fusion of technical acumen and a deep understanding of user needs. The experience underscored the importance of combining academic theory with real-world application and solidified my belief in the power of continuous learning. Reflecting on this endeavor, I am further inspired to apply my educational insights to address real-world challenges.

1.2 Business Problem Background

To thoroughly analyze the business problem background, we embarked on an interview survey. Recognizing that our platform primarily targets students preparing to embark on their PhD journeys, we specifically selected these individuals as our interviewees. We aimed to delve deep into their academic search habits, preferences, and potential pain points when interacting with existing platforms. The questionnaire, as detailed below, was meticulously crafted to encompass a spectrum of pertinent topics. By dissecting their responses, we sought to identify gaps in current tools and thereby tailor our platform to cater to their unique academic needs more holistically. This endeavor not only fortified our understanding of the market's requirements but also laid a strong foundation for the subsequent design and development phases of our academic search engine.

The interview results are shown in Table 2. Based on the findings from our recent survey, it has become evident that while Google Scholar serves as a widely utilized academic search engine, it has its shortcomings, especially for those about to embark on their Ph.D. journey and those unfamiliar with specific academic terminologies or the breadth of research in their areas of interest. Some participants expressed difficulty navigating the large volume of scholarly articles, often needing clarification on where to begin their search or how to delve more deeply into a topic. The process can feel overwhelming and, at times, inefficient, highlighting the necessity for a more personalized and intuitive approach.

To address these issues and offer an enhanced academic search experience, we have developed a new search engine built on Java and SpringBoot. Our unique system aims to remedy the existing limitations of Google Scholar by introducing the following features:

No.	Question
1	Do you feel that existing tools like Google Scholar meet all your needs when looking for academic articles or research materials?
2	Would you like a system that recommends related academic articles based on your interests and browsing history?
3	When determining the research direction for your PhD, do you feel that exposure to more academic article recommendations would assist your decision-making?
4	When conducting academic searches, have you ever encountered uncertainty in how to express or choose the correct keywords?
5	Do you think that displaying the core words of search results through a word cloud would help you understand and filter the content more quickly?
6	Would you prefer a platform that tracks your academic article views, likes, and bookmarks for future reference?
7	When screening academic articles, would you like a brief summary to understand the content and decide whether to delve deeper quickly?
8	When using current academic search tools, which features do you think are missing or could be further refined?
9	Do you believe that an academic search platform that offers personalized recommendations and features would be more aligned with your needs?
10	In academic research, do you feel that you spend a considerable amount of time screening and searching for relevant articles and wish there were more efficient tools to save this time?
11	Are there other platforms that address your aforementioned needs?

Table 1: Survey Questions on Academic Search Needs for PhD Students

1. Personalized Article Recommendations: Instead of the usual trial-and-error method, our platform proactively suggests academic articles to users. Recommendations span articles with the highest click rates, the latest publications, papers read by similar users, articles related to a user's browsing history, and a blend of these factors. This feature not only familiarizes users with the latest trending research areas but also guides them toward relevant content based on their unique profiles.

2. Search Term Optimization: Recognizing that newcomers to a field might need to be acquainted with all its jargon or might have a nebulous understanding of their interests, we have incorporated a term optimization chatbot. When a user keys in a term, the bot offers the five closest related terms to refine their search, helping users articulate their research intent.

3. Visual Representation via Word Clouds: To give users a bird's-eye view of their search results, our platform showcases a word cloud representing core search terms. This visual tool assists users in determining the accuracy of the results and the broader academic domain.

4. User Interaction Features: By integrating options for liking, saving, and clicking, we capture user interactions. This dual-fold feature lets users revisit their activities on their dashboard and aids the system in tailoring future article recommendations.

5. Article Summarization: A key differentiator is the feature that generates a concise summary of saved articles. Users can grasp the crux of an article in a sentence, allowing them to filter out non-relevant papers swiftly and save precious time.

No.	Question	Response
1	How often do you search for academic articles?	Daily.
2	Do you often struggle to find the right articles that align with your interests?	Yes, it's quite challenging at times.
3	How do you currently discover new research topics or areas?	Mainly through reference or random browsing or going to other social platforms such as Zhihu to find reading suggestions.
4	Would a platform recommending articles based on your interests be beneficial?	Absolutely, it would save a lot of time and help me focus on relevant research.
5	How often do you feel overwhelmed by the vast number of articles available?	Very often. It's hard to know where to start or which articles are the most relevant to my research interests.
6	Do you believe an interactive tool suggesting search terms would aid your research process?	Definitely. This helps me directly discover the general scope of the search results, and if it doesn't meet my expectations, I can directly search again. Saved me a lot of time reading them one by one.
7	How beneficial would visual representations (like word clouds) be for understanding search results?	Extremely beneficial. It provides an instant overview and helps quickly identify the relevance of search results.
8	Would you appreciate a platform that tracks your reading habits and provides insights based on them?	Yes, it would help make my research more structured and directed.
9	How important is having a summarized version of articles for you?	Very important. With the plethora of articles out there, having summarized versions would aid in efficient screening.
10	Do you feel existing platforms sufficiently cater to your academic search needs?	Not completely. There's room for improvement, especially in personalized recommendations and insights.
11	Are there any other platforms that address your aforementioned needs?	Not that I'm aware of. Most platforms offer general search capabilities, but none with the personalized approach I desire.

Table 2: Questionnaire Responses from a Prospective PhD Student

1.3 Market Research

1.3.1 Current System Description

Although Google Scholar has not publicly disclosed its algorithm, based on user experience, we can describe its core functionalities. In a typical session on Google Scholar, a search taps into a comprehensive database of academic content across various disciplines. Upon entering a query, the results display a combination of articles, conference papers, theses, and books pertinent to the search terms. An unmistakable feature is the citation count listed below each result, indicating the influence of that particular work. Clicking on an article provides access to the content and often introduces related articles. To the right, links may direct users to open-access or institutional copies of the document. Additionally, clicking on an author's name leads to their profile, presenting a list of their publications and total citations. For more specific searches, there is an advanced search feature, allowing for detailed search criteria. Overall, Google Scholar's functionalities aim to streamline and support the academic research process. The above is a description of the existing functions of Google

Scholar.

1.3.2 Current System Overview

Google Scholar offers two primary modes of operation: for visitors and logged-in users. Regardless of the mode, the primary function remains consistent: users input keywords to search for academic papers and articles. For visitors, the process is straightforward – enter keywords, and the platform returns relevant search results. However, for logged-in users, Google Scholar offers an enhanced experience. Beyond merely searching and viewing articles, logged-in users have the added benefit of saving or "starring" articles of interest. These saved articles are conveniently stored in their library on Google Scholar, allowing easy future access and organization of their academic research.

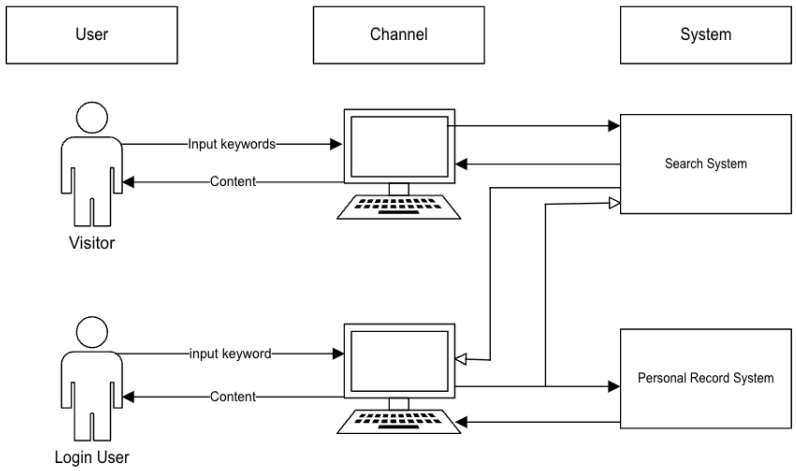


Figure 1: Current-System-overview

1.3.3 Current System Network

The flowchart showcases the HTTPS (HyperText et al.) connection process employed by Google Scholar, consistent with Google's other services. This protocol establishes a secure and encrypted connection between the user's browser and the Google Scholar server. The process begins with the user's browser sending a connect request. In response, the Google Scholar server provides its public key and a digital certificate. Subsequently, the user's browser dispatches an encrypted random key using the server's public key. With this secured key exchange in place, encrypted data transfer commences. The user's query, now securely encrypted, is sent to the server, which returns encrypted search results.

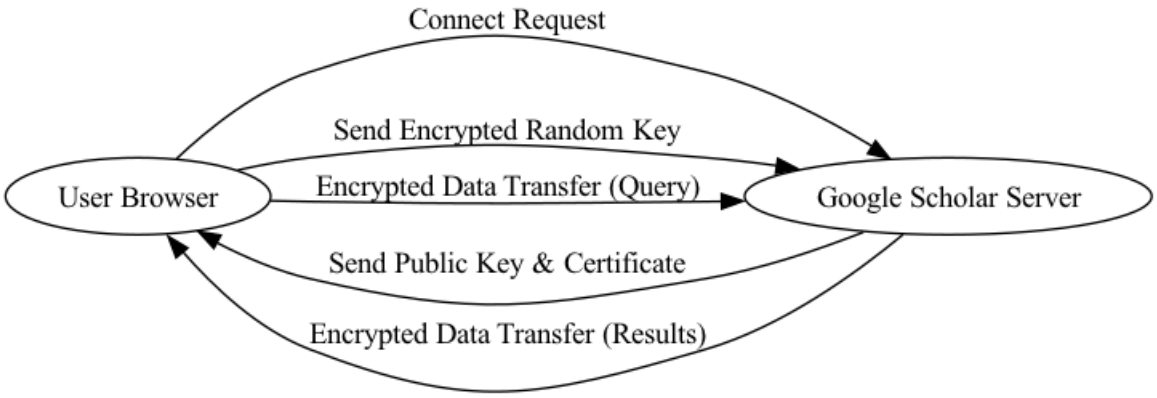


Figure 2: Network system diagram

1.3.4 User Proportion

According to the data, The proportion of users of different search engine companies is as follows:

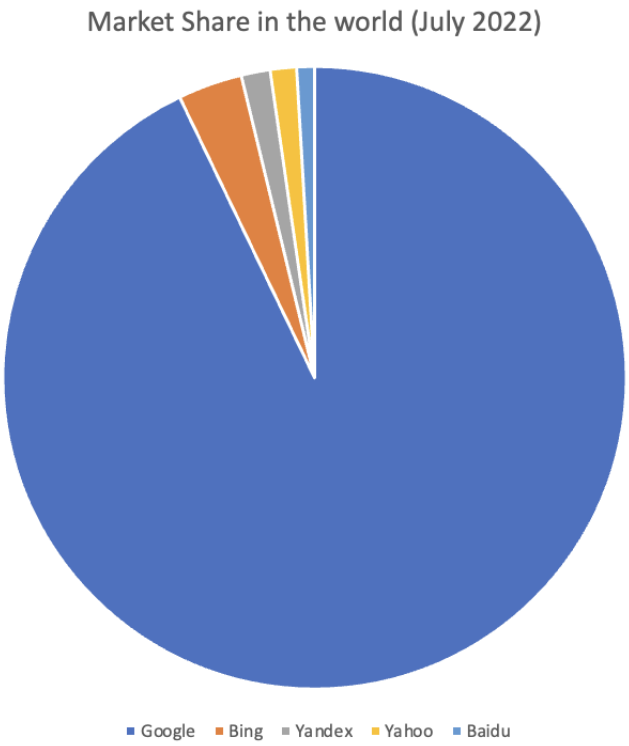


Figure 3: User Proportion

Based on the presented market share data, it is evident that Google dominates the search engine landscape, commanding 91.46% of the market. This dominance not only underscores Google's immense influence but also signifies a deep-seated trust and reliance by most users. In contrast, competitors like Bing, Yandex, Yahoo, and Baidu lag significantly behind in their market shares. Given this dominance, users are likely accustomed to the interface, functionalities, and reliability Google offers. However, this very dominance also suggests potential areas of innovation and improvement within the realm of academic search that Google Scholar might still need to cater to. In essence, this market share information illuminates the need for enhancements to Google Scholar, capitalizing on potential unmet user needs that could provide a competitive edge.

1.3.5 Competitor Research

In the rapidly evolving digital landscape, understanding the competitive environment is paramount for any venture aiming to innovate and capture market share. Recognizing the strengths and weaknesses of existing platforms allows us to discern gaps in the market and areas for potential improvement. Therefore, to gain a comprehensive perspective and ensure we tailor our offerings effectively, we conducted an in-depth analysis of our competitors. The findings of this analysis are presented below:

Name	Technology	Advantages	Disadvantages
Semantic Scholar	User preference analysis	Recommendations based on user's interest	Requires initial input if new user
Google Scholar	Statistical models	Comprehensive relevance determination	Might require extensive data like publishing history
Papers With Code	Trend analysis	Focuses on latest CS research	Only contains code-related works
Connected Papers	Knowledge Graph	Comprehensive connections via citation graph	Might require understanding of graph representation
Paper Weekly	User history analysis	Personalized AI research recommendations	Specific to AI domain

Table 3: Competitor Analysis

Next, we will analyze the areas where Google Scholar currently needs improvement and improvement based on the needs of target users and the strengths and weaknesses of market competitors.

1.3.6 Improvement Areas

In order to improve the effectiveness and user-friendliness of Google Scholar, a comprehensive analysis is necessary. By combining this with the specific needs and preferences of the target user group, we can determine which improvements will have the most profound impact on the user experience. Additionally, by considering the market landscape, we can gain insights into our competitors' strengths and potentially integrate their successful strategies into Google Scholar. Conversely, recognizing its weaknesses can provide Google Scholar with a unique opportunity to differentiate itself and offer features or functionality that its competitors lack. Therefore, we have compiled the following points that can be improved:

No.	Point	Description
1	Personalized Article Recommendations	Instead of solely relying on user-initiated searches, Google Scholar can proactively suggest academic articles. By leveraging data like articles with high click rates, latest publications, user's browsing history, and behaviors of similar users, it can provide tailored recommendations. This would help users stay updated on trending research and find relevant content effortlessly.
2	Search Term Optimization	To aid users, especially newcomers to a field, Google Scholar can incorporate a term optimization chatbot. This bot can suggest closely related terms when a user inputs a query, thus assisting in refining searches and making them more efficient.

No.	Point	Description
3	Visual Representation via Word Clouds	Upon executing a search, users can benefit from a visual representation in the form of a word cloud highlighting core terms. This not only helps determine the relevance of the results but also provides insights into related academic domains.
4	User Interaction Features	By adding features like liking, saving, and tracking clicks, Google Scholar can capture user interactions. These functionalities can enhance user engagement, allow easy revisiting of important articles, and further refine recommendation algorithms.
5	Article Summarization	By offering a feature that produces a brief summary of articles, users can quickly grasp the essence of a paper. This can save time and help users swiftly weed out non-relevant research.

Table 4: Proposed Improvements for Google Scholar based on Competitor Analysis and User Needs

1.4 Implementation Plan

In order to develop the academic search recommendation platform in a planned way, I made the following plan:

Figure 4: Gantt Chart

2 Project Solution

2.1 Functional and Non-functional Requirement

After conducting a comprehensive analysis, I meticulously listed all the functional and non-functional requirements for the system I intend to develop. This meticulous detailing is paramount as it not only ensures a clear understanding of the project's scope but also establishes a solid foundation for future development phases. Additionally, these lists serve as essential tools for communication between stakeholders, ensuring that all parties have aligned expectations and a shared vision for the project.

2.1.1 List of Functional Requirements

Table 5: Table of Functional Requirement

Req.ID	**Requirement Title**	**Description**	**Target User**
REQ-SYS-001	Login	Allows user to access their personal account	Visitor
REQ-SYS-002	Logout	Allows user to safely exit from their account	Logged-in User
REQ-SYS-003	Register	Enables new users to create a unique account	Visitor
REQ-SYS-004	Survey	Collects user's preferences via a questionnaire post-registration	Visitor
REQ-SEA-001	Search	Enables users to search for articles using keywords	All Users
REQ-SEA-002	Filter	Allows users to narrow down search results by a specific year range	All Users
REQ-REC-001	RS (popular paper)	Offers recommendations based on popular articles in the platform	All Users
REQ-REC-002	RS (latest paper)	Provides recommendations based on the latest published articles	All Users
REQ-SYS-005	Like	Allows users to show appreciation or interest in an article	Logged-in User
REQ-SYS-006	Star	Enables users to save/favorite specific articles for later	Logged-in User
REQ-SEA-003	Word cloud	Visual representation of keyword frequencies in searched articles	All Users
REQ-SEA-004	Chat robot	Assists users in refining their search terms by providing related keyword suggestions	All Users
REQ-SYS-007	Personal page	Displays a user's activity history, likes, and saved articles	Logged-in User

Continued on next page

Table 5 – continued from previous page

Req.ID	Requirement Title	Description	Target User
REQ-REC-003	RS(User-based)	Personalized recommendations based on user's interaction patterns with articles	Logged-in User
REQ-REC-004	RS (Item-based)	Offers recommendations by analyzing textual similarities in articles a user liked	Logged-in User
REQ-REC-005	RS (Mix)	A hybrid recommendation system blending item-based filtering with popularity metrics	Logged-in User
REQ-SYS-008	Paper Summary	Provides a concise summary of an article, enabling users to grasp the main idea quickly	Logged-in User

2.1.2 List of Non-Functional Requirements

Table 6: Table of Non-Functional Requirement

Req.ID	Category	Requirement Title	Target Users	Priority
REQ-PRF-001	Performance	Responsive Search	Visitor/Login-User	M
REQ-USR-001	Usability	Intuitive Navigation	Visitor/Login-User	M
REQ-ACS-001	Audit, Control, Security	User Activity Logging	Login-User	M
REQ-RLB-001	Reliability	System Uptime	login-User	M

Non-Functional Requirement Description:

Table 7: Responsive Search

Item	Description
Requirement ID	REQ-PRF-001
Category	Performance
Requirement Title	Responsive Search
Priority	Must
Non-Functional Requirement Description	*The system should provide search results instantaneously without any noticeable delay. *The search functionality should accommodate broad queries and provide relevant results.

Table 8: Intuitive Navigation

Item	Description
Requirement ID	REQ-USR-001

Category	Usability
Requirement Title	Intuitive Navigation
Priority	Must
Non-Functional Requirement Description	*The system's interface should be user-friendly and easy to navigate. *Menus, buttons, and other navigation elements should be clearly labeled and organized logically. *Any action taken by the user should have a clear path to revert or undo.

Table 9: User Activity Logging

Item	**Description**
Requirement ID	REQ-ACS-001
Category	Audit, Control, Security
Requirement Title	User Activity Logging
Priority	Must
Non-Functional Requirement Description	*The system should log user activities. *Logs should be time-stamped and associated with user IDs for traceability.

Table 10: System Uptime

Item	**Description**
Requirement ID	REQ-RLB-001
Category	Reliability
Requirement Title	System Uptime
Priority	Must
Non-Functional Requirement Description	*The system should be available for use as consistently as possible, aiming for 99.9% uptime. *In the event of system failures, the recovery process should be swift with minimal data loss. *Regular maintenance should be scheduled during off-peak hours to avoid inconveniencing users.

2.2 System Overview

2.2.1 Project Objectives & Success Measurements

This project aims to develop an academic search engine based on Java and SpringBoot to improve some of the known shortcomings of Google Scholar. The target users are those preparing to pursue a PhD but need to learn more about the academic field and personal research interests. Through this system, we aim to help these users better determine their research directions and interests. First, the system can actively recommend academic articles to users, thus solving the problem of users "not knowing where to start" when faced with massive academic resources. In addition, by optimizing search terms and using word clouds, newbies in academic fields can be helped to find the information they need more accurately. Functions such as likes, favorites, and clicks allow the system to understand users' interests better and provide them with more personalized recommendations.

The success of the project will be assessed through several vital indicators. First, users can quickly determine their research direction through the articles the system recommends; secondly, new users in the academic field find search term optimization and word cloud diagrams to be of substantial help to their searches. In addition, the system needs to regularly check whether users frequently use functions such as likes, collections, and clicks and whether they frequently visit their home pages to find their activity records. Finally, for the article summary function, it is necessary to evaluate whether users can quickly filter articles through the short summaries provided by the system, thus saving time.

2.2.2 System Structure

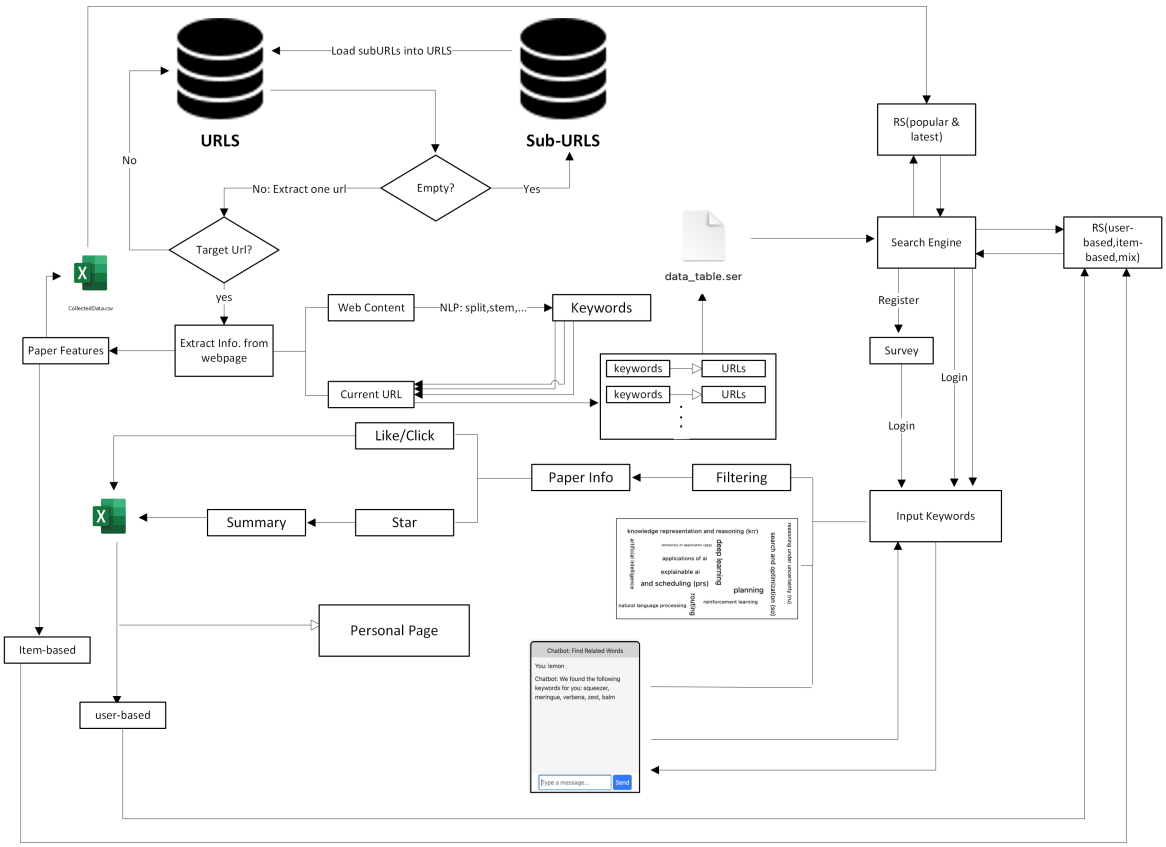


Figure 5: System Overview

2.2.3 Usecase Diagram

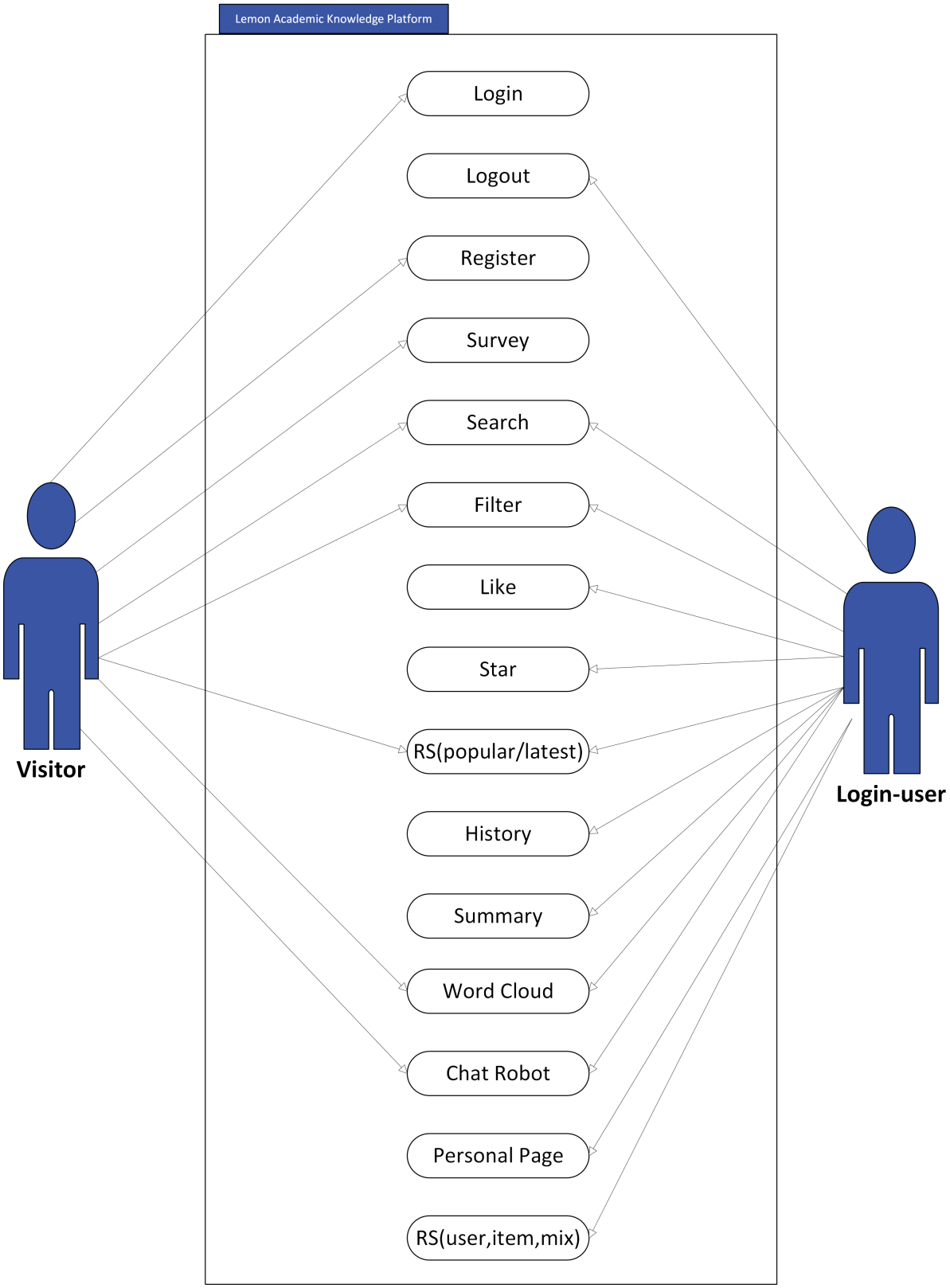


Figure 6: Usecase diagram of proposed functions

2.2.4 Human-computer Interaction Diagram

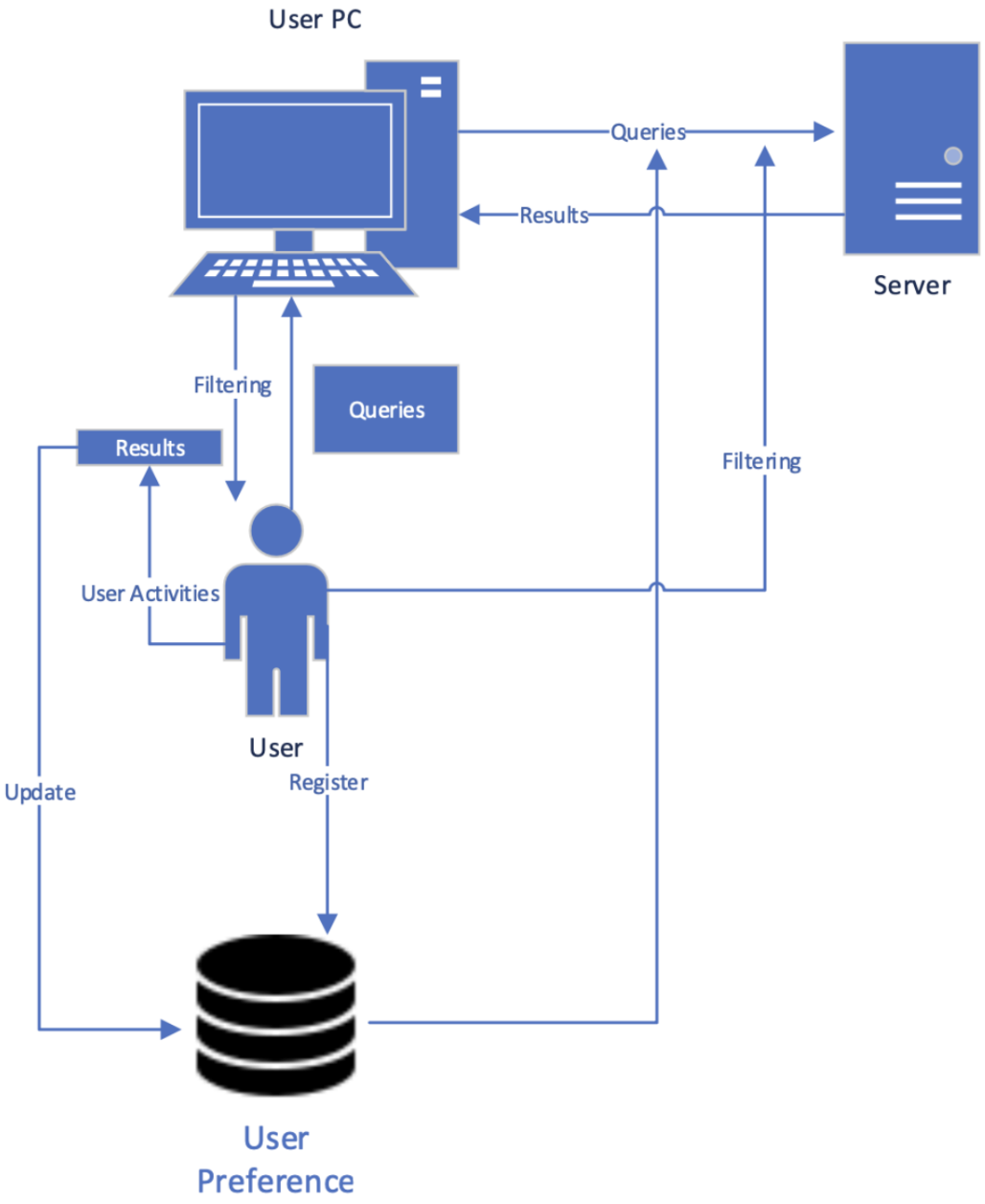


Figure 7: Human-computer Interaction Diagram

2.3 Table of Functions

Table 11: Requirement System Overview

Business Needs	**Major Features**	**System Related Function**
User Account Management	Enable users to create an account, manage their sessions, and maintain secure access.	Login, Logout, Register, Survey

Search Capability	Offer users a robust search experience with the ability to fine-tune their results.	Search, Filter,Word cloud
Recommendation Systems	Enhance user experience by suggesting articles based on various criteria.	RS (popular paper), RS (latest paper), RS(User-based), RS (Item-based), RS (Mix)
User Interaction	Allow users to interact with content, marking preferences, and keep track of interesting articles.	Like, Star, Click website, Personal Page
Assistance and Query Expansion	Assist users in refining their searches and understanding their areas of interest.	Chat robot(API)
Content Summarization	Help users quickly grasp the essence of articles and decide on their relevance.	Paper Summary(API)

2.4 Functions Details

2.4.1 User Account Management

Table 12: User Account Management Functions

Item	**Description**
Business Need	User Account Management
Register Function	
Function Name	Register
Function Description	Allows users to create a new account by providing the necessary details. This account will grant them access to the search recommendation system.
Mode	Online
Business Rules	If the provided username or email is already in use, the system should inform the user and ask for different credentials. Upon successful registration, a confirmation email will be sent to the user.
Security Requirements	The system must store passwords in a hashed and salted format. Passwords must adhere to security best practices.
Login Function	
Function Name	Login
Function Description	Allows registered users to access the system using their credentials.
Mode	Online
Business Rules	If the user inputs an incorrect id, it cannot log in successfully and back to the home page.
Logout Function	
Function Name	Logout
Function Description	Allows users to end their session and exit the system securely.

Mode	Online
Business Rules	Ensure that the user's session is properly terminated and cookies or other session identifiers are cleared.
Survey Function	
Function Name	Survey
Function Description	Enables users to provide feedback or additional information about their preferences or experience using the system.
Mode	Online
Business Rules	Only registered users should be able to complete the survey. Collect user feedback for further system improvements.

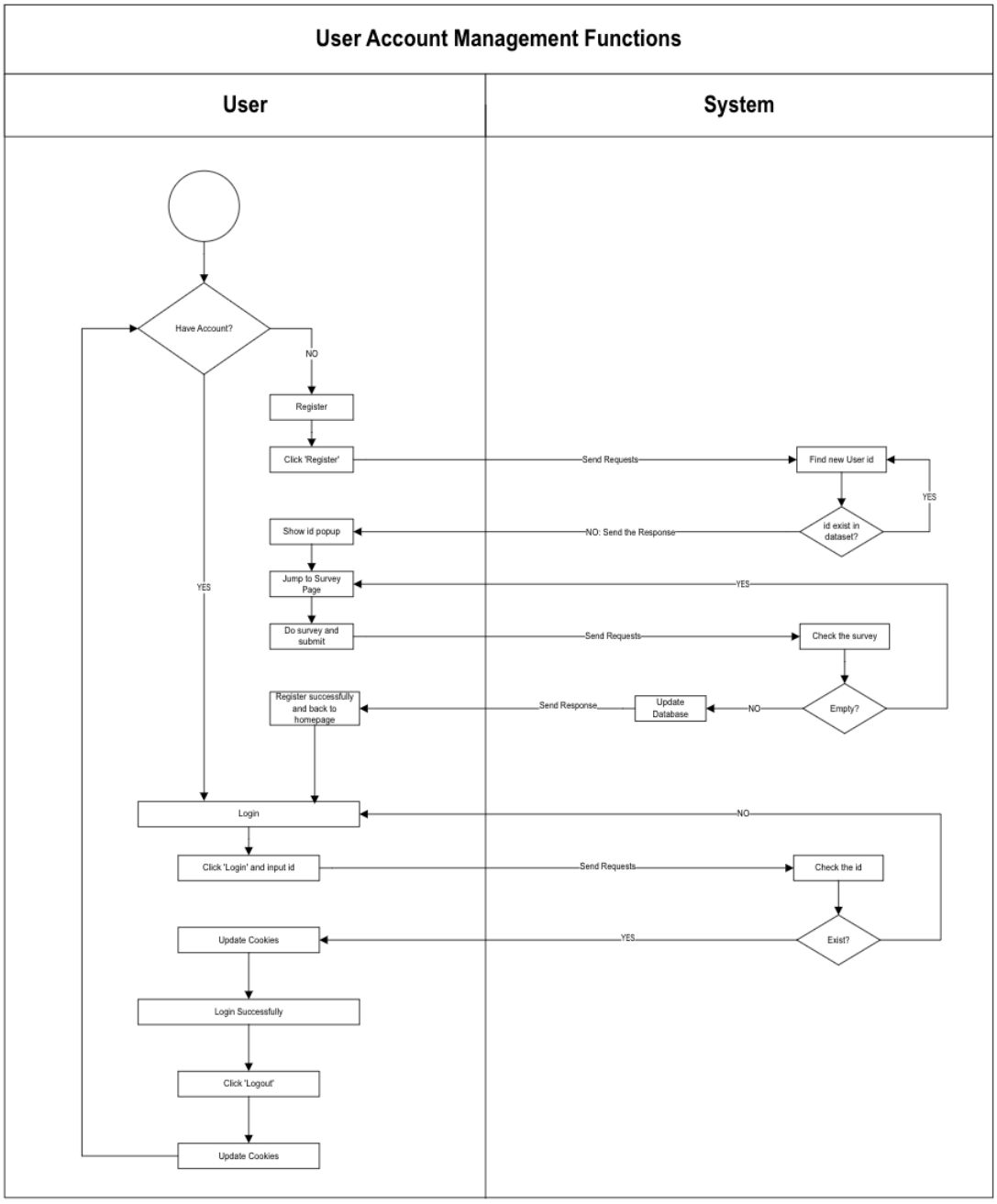


Figure 8: Account Management

System accounts are managed as follows: Within the User Account Management framework, the registration process is initiated when users without an existing account click on "Register." This action prompts the backend to generate and assign a unique user ID, ensuring no prior use. Once generated, a frontend popup informs the user of their designated ID. The system then automatically navigates to the "Survey" page, where users must answer questions and select papers that pique their interest. It is imperative to select at least one paper for successful registration, failing which, the process will not be completed. Additionally, any disruption, such as page closure or navigation back to the homepage, will abort the registration.

For returning users, the login mechanism is straightforward. By clicking "Login" and entering their unique user ID, a backend verification process determines the validity of the user credentials. A successful match grants access, saving the user's ID in a cookie for the session's duration. If no match is found, the system redirects to the main page. Lastly, the "Logout" option is available for users wishing to end their session, ensuring their ID is securely removed from the cookie. The whole process is shown in the figure 8.

2.4.2 Search Capability

Table 13: Search Capability Functions

Item	**Description**
Business Need	Search Capability
Search Function	
Function Name	Search
Function Description	Allows users to input keywords, returning articles that match the entered terms. Results are ordered based on the degree of matching.
Business Rules	Returns a list of articles matching the user's search criteria, ranked by relevance.
Filter Function	
Function Name	Filter
Function Description	Empowers users to manually refine search results manually, providing the ability to limit the range of publication years for articles.
Business Rules	Allow users to set desired date ranges, ensuring that returned articles fall within specified boundaries.
Word Cloud Visualization Function	
Function Name	Word Cloud Visualization
Function Description	Displays a word cloud of the keywords from search results, giving users a rapid overview of the themes and ensuring they align with their search intent.
Business Rules	Keywords are aggregated based on frequency from search results and represented visually in the word cloud. More frequent terms are emphasized in larger fonts.

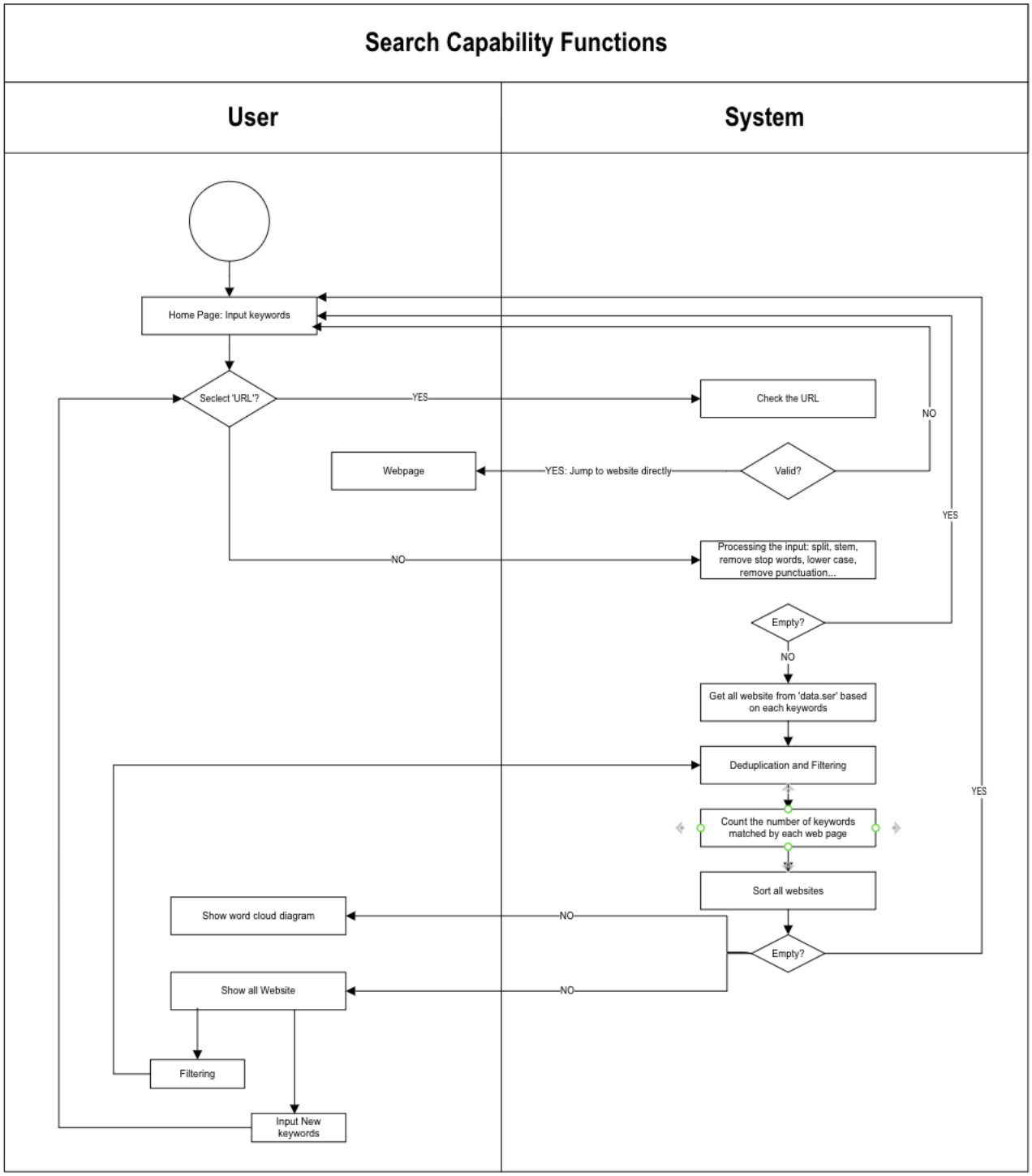


Figure 9: Search Capability

The search functionality of our system offers a seamless and user-centric experience, designed with precision and care to cater to the user's every need. When users input keywords, they are immediately presented with matching articles. These results are carefully organized, with the most pertinent articles appearing at the top, thanks to an intelligent ranking system prioritizing relevance. Enhancing this core search experience is the inclusion of the "Filter" feature. This tool empowers users to refine their results manually, such as by narrowing down to articles within a specific date range. While the left side of the user interface showcases these results, users can further hone in on their search by applying additional criteria like the publication year. Beyond just listing articles, the system provides a holistic overview of the search results through a word cloud visualization. By aggregating the keywords from the search results, this word cloud gives users a rapid snapshot of the main themes

and topics, helping them discern if the returned articles align with their initial intent.

The system's prowess is broader than just delivering search results. With the URL button, users can navigate directly to the relevant page if the linked URL is valid. However, any detour due to an invalid link smoothly redirects the user back to the main homepage, ensuring uninterrupted browsing. Behind this user-friendly interface, the system boasts advanced backend processing. Each user query undergoes a meticulous transformation: from conversion to lowercase, punctuation removal, cleansing of stop words, and stemming to tokenization. The system then scours the 'data.ser' file using these refined keywords to fetch corresponding web pages. To ensure accuracy, duplicates are removed, and the remaining web pages are ranked based on their alignment with the input terms. In the rare event of no matches or an empty query, users are gracefully led back to the main page. This intricate dance of frontend presentation and backend processing culminates in a visual treat: the word cloud. Displayed on the right side of the interface, this artistic rendering of article keywords not only augments the interface's visual appeal but also assists users in their decision-making process. Moreover, if users ever desire to start anew, the system stands ready, inviting them to input new terms and embark on a fresh search journey. The whole process is shown in the figure 9.

2.4.3 User Interaction

Table 14: User Interaction Functions

Item	**Description**
Business Need	User Interaction
Click Function	
Function Name	Click
Function Description	Allows users to select and view an article from the list of search results.
Business Rules	Every click on an article is recorded for user activity tracking.
Like Function	
Function Name	Like
Function Description	Enables users to show appreciation for an article by clicking the "like" icon.
Business Rules	The "like" status will be visible (icon highlighted) upon subsequent logins. User activity related to "likes" is saved for recommendation purposes.
Star Function	
Function Name	Star
Function Description	Allows users to save or bookmark articles of interest for future reference.
Business Rules	The bookmark status (icon highlighted) persists across user sessions. User activity related to bookmarks is saved for recommendation purposes.
Personal Page	
Function Name	Personal Page
Function Description	Provides users with a page that lists their interactions, including clicked, liked, and starred articles.
Business Rules	Only the user can access their personal page. This page aids in easy retrieval of previously viewed or liked content.

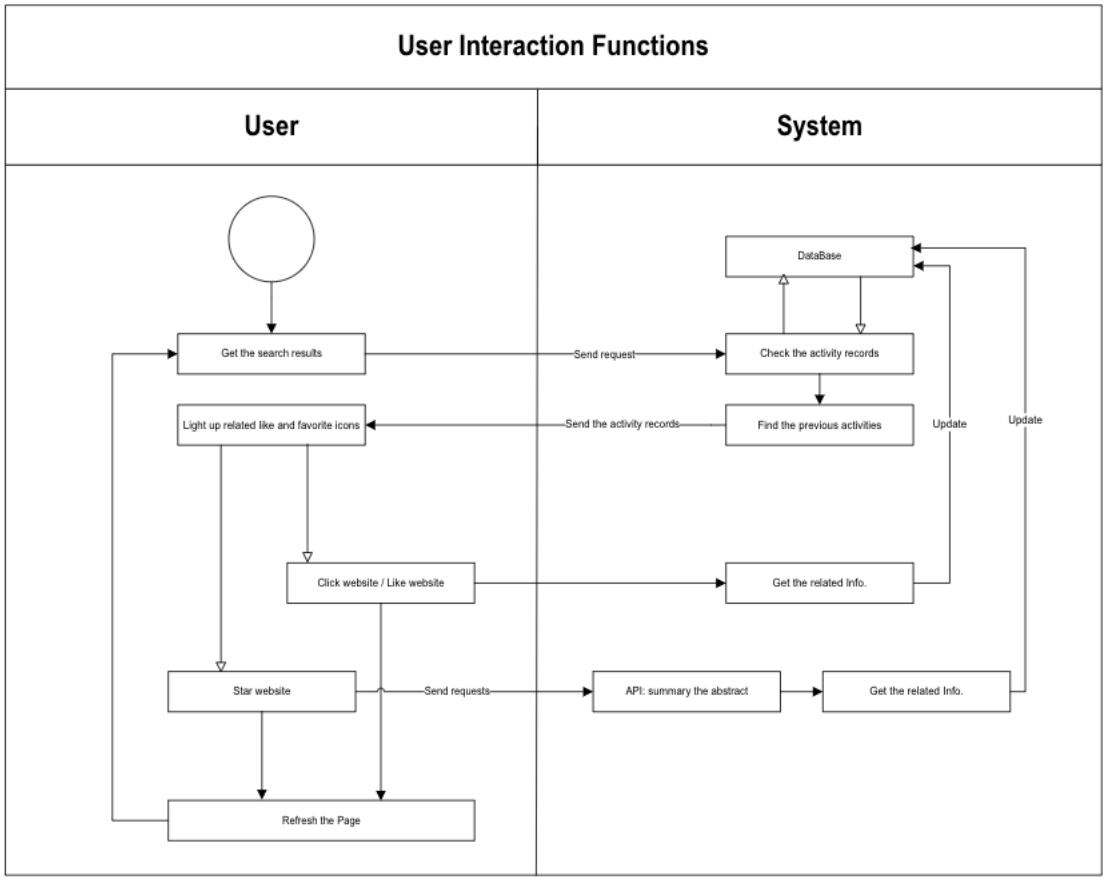


Figure 10: User Interaction

After successfully logging in and when the search engine retrieves articles, users can interact with them in several ways: by clicking on them, liking them, or bookmarking them for later reference. All these interactions are recorded for each user.

When a user clicks or likes an article, the corresponding data is immediately sent to the backend for recording. The bookmarking functionality operates slightly differently. If a user decides to bookmark an article, an API request is triggered, fetching a summarized abstract of the article. This allows users to understand the article's content in a single sentence. Both the act of bookmarking and the summarized abstract are recorded together. Every time users log into the search webpage, their previous interactions, such as likes and bookmarks, are prominently displayed, indicated by highlighted icons. For a more comprehensive overview of their interactions, users can navigate to their 'Personal Page.' This page showcases all the articles they have read, liked, and bookmarked, making it convenient for users to revisit and review their previous activities. The whole process is shown in the figure 10.

2.4.4 Recommendation Systems

Table 15: Recommendation System Functions

Item	**Description**
Business Need	Recommendation System
Article Recommendation Function	
Function Name	Article Recommendation
Function Description	Suggests articles to users based on their search history, likes, and interactions.
Business Rules	Recommendations are generated using machine learning algorithms, which factor in user activity and preferences.
Related Articles Function	
Function Name	Related Articles
Function Description	Displays articles that are similar or related to the article the user is currently viewing.
Business Rules	Similarity are determined based on article content, keywords, and user interactions with other articles.
User Behavior Analysis Function	
Function Name	User Behavior Analysis
Function Description	Analyzes user behavior to understand and predict what types of articles they might be interested in.
Business Rules	User behavior data, such as clicks, likes, and search history, is processed to generate insights and improve the recommendation engine.
Feedback Loop	
Function Name	Feedback Loop
Function Description	Allows users to provide feedback on recommended articles, helping to refine and improve the recommendation system.
Business Rules	User feedback is used to train the recommendation model, ensuring that the system evolves and adapts to user preferences over time.

1. Popularity-based Recommendation

Basic Introduction:

The popularity-based recommendation algorithm predominantly considers the click-through rate (CTR) of articles. It works by querying the database to select articles with the highest CTR, assuming that articles with more clicks generally resonate more with the broader user base.

Relevant Formula:

$$\text{Score}(\text{Article}) = \text{CTR}(\text{Article}) \tag{1}$$

Algorithm Flowchart:

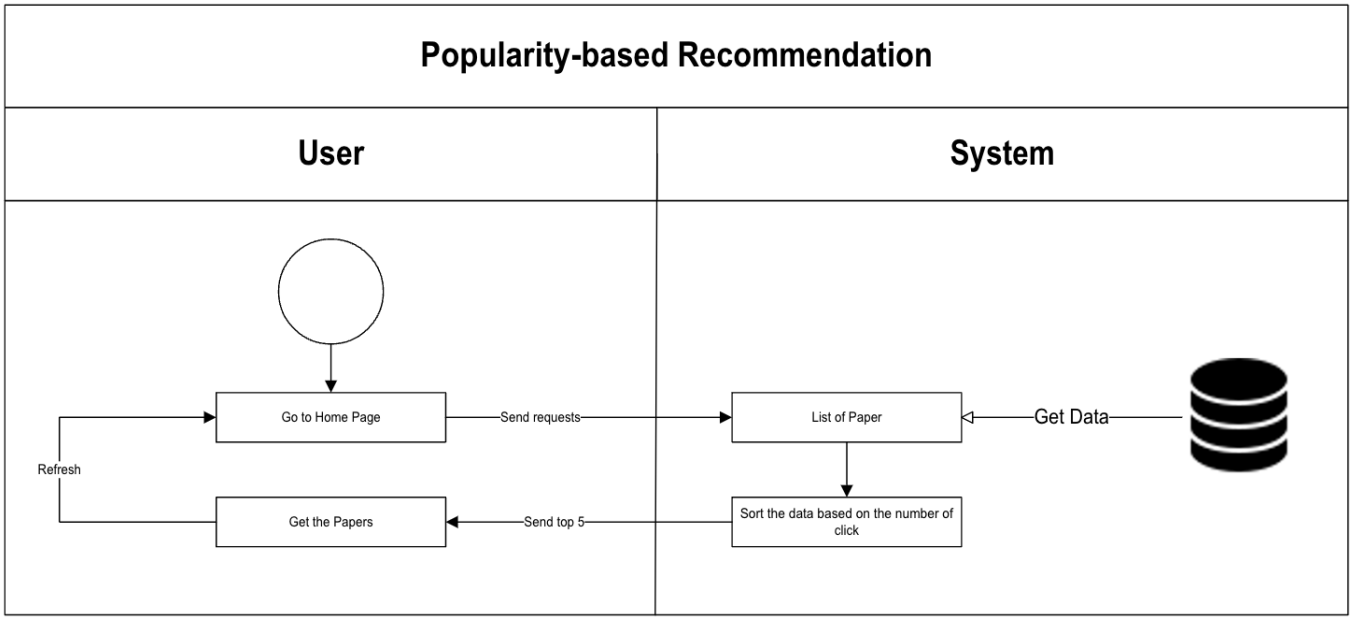


Figure 11: Popularity-based Recommendation

Reasoning for Design and Impact on User Experience:

For popularity-based recommendation algorithms, its simplicity ensures fast response times and provides immediate value, especially to users who have yet to interact with the data beforehand. If a user has just registered an account, we cannot accurately obtain his preferences. Therefore, we can use this algorithm to recommend articles that most people like to him. In this algorithm, articles are prioritized based on click-through rate. Hypothetically, if a more extensive audience finds a particular article attractive, we speculate that new users will likely find it attractive. It helps deliver trending and resonating content to users, enhancing their initial engagement with the platform.

2. Publication Year-based Recommendation

Basic Introduction:

The publication year-based recommendation algorithm emphasizes the recency of articles. This method prioritizes newly published content, assuming that users might be more interested in recent findings or discussions in the academic realm.

Relevant Formula:

$$\text{Score}(\text{Article}) = \text{Publication\_Year}(\text{Article}) \tag{2}$$

Algorithm Flowchart:

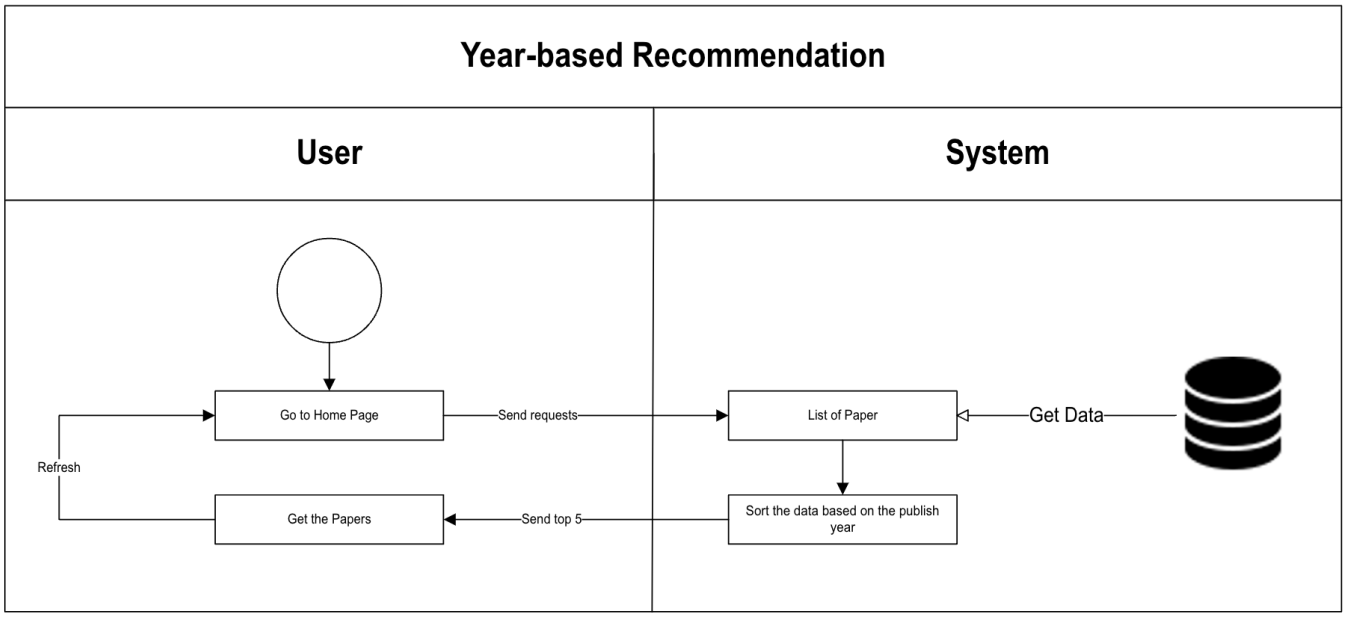


Figure 12: Publication Year-based Recommendation

Reasoning for Design and Impact on User Experience:

For recommendation methods based on the year of publication, the primary purpose is to keep users informed of the latest research and discussions. This approach is constructive for academic users or researchers keen to keep up with the latest advances in their field. Since our target users are students about to study for PhD, they need to obtain the latest scientific research papers. On the one hand, they can learn the latest knowledge, and on the other hand, they can check the progress of scientific research frontiers to determine their own future scientific research direction. By emphasizing the currency of articles, it ensures users do not miss out on contemporary discussions, thereby fostering a sense of staying updated. Therefore, we added this recommendation method to the recommendation algorithm.

3. User-based Recommendation Algorithm

Basic Introduction:

The User-based recommendation algorithm primarily focuses on a user's interaction history, particularly 'likes,' to offer content suggestions. By analyzing users' liking patterns, the system identifies other users with similar preferences. Once a likeness is established, the system recommends articles that the similar user has engaged with but that the current user still needs to explore. This process assumes that users with similar liking patterns will likely have similar content interests.

Relevant Formula:

User-Item Matrix Formation:

$$M_{ui} = \begin{cases} 1 & \text{if user } u \text{ has liked item } i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Where M_{ui} denotes the entry in the User-Item matrix for user u and item i. This matrix represents all users' liking behavior.

User Similarity Measurement:

$$\text{similarity}(u, v) = \frac{\sum_i M_{ui} \times M_{vi}}{\sqrt{\sum_i M_{ui}^2} \times \sqrt{\sum_i M_{vi}^2}} \tag{4}$$

Where u and v are different users. This equation calculates the cosine similarity between users based on their liking patterns.

Article Selection:

$$\text{Recommended\_Items}(u) = \text{Items\_Liked}(v) - \text{Items\_Already\_Seen}(u) \tag{5}$$

Where u is the current user, and v is the most similar user. This formula determines the set of articles to be recommended to the user.

Algorithm Flowchart:

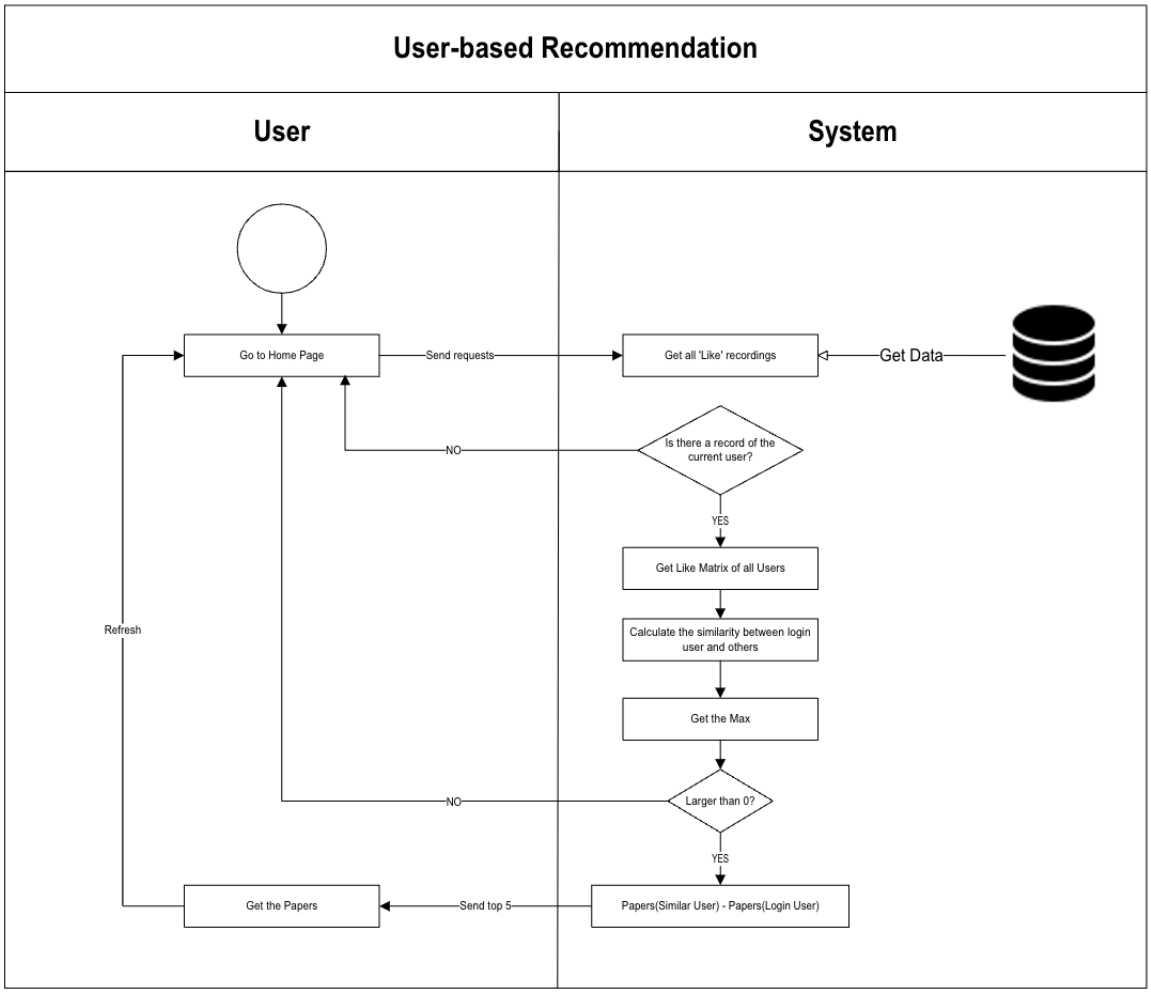


Figure 13: User-based Recommendation

Reasoning for Design and Impact on User Experience:

User-based recommendation algorithms are designed with the belief that users' preferences often reflect the preferences of similar users. By tapping into these shared interests, the platform can deliver highly personalized content closely aligned with individual tastes. Similar users read similar articles.

Furthermore, similar users should also be interested in similar research directions. Especially when users are relatively new and have yet to provide much explicit feedback, identifying patterns among users with an established history of interactions becomes extremely valuable. Therefore, we have added this recommendation method to help users quickly determine their scientific research direction. The direct benefit is an enhanced user experience where users feel the platform understands their interests, promoting long-term engagement and trust.

4. Item-based Recommendation Algorithm

Basic Introduction:

The Item-based recommendation algorithm emphasizes the characteristics of the articles themselves. After a user interacts with specific articles, notably by liking them, the system analyzes the core words and abstracts of these articles to determine critical features. By comparing these features with those of other articles in the database, the algorithm identifies and recommends new content with similar characteristics that the user still needs to explore.

Relevant Formula:

Article Feature Matrix Formation:

$$F_{ai} = \begin{cases} 1 & \text{if article } a \text{ contains feature } i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Where F_{ai} denotes the entry in the Article-Feature matrix for article a and feature i. This matrix represents all articles' features.

Article Similarity Measurement:

$$\text{similarity}(a, b) = \frac{\sum_i F_{ai} \times F_{bi}}{\sqrt{\sum_i F_{ai}^2} \times \sqrt{\sum_i F_{bi}^2}} \tag{7}$$

Where a and b are two different articles. This equation calculates the cosine similarity between two articles based on their features.

Article Selection:

$$\text{Recommended\_Articles}(a) = \text{Top\_5\_Most\_Similar\_To}(a) - \text{Articles\_Already\_Seen} \tag{8}$$

This formula selects the top 5 articles most similar to the liked article a that the user hasn't yet seen.
Algorithm Flowchart:

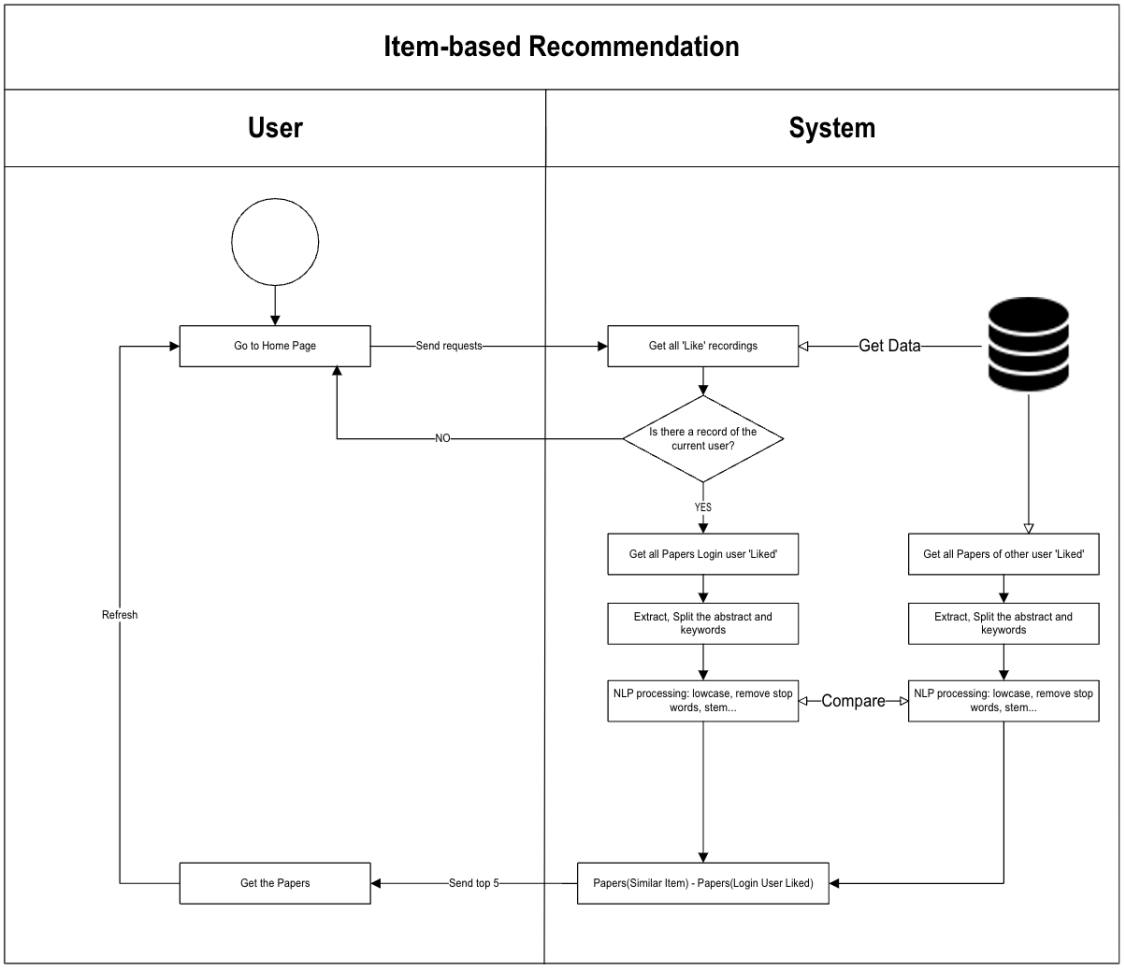


Figure 14: Item-based Recommendation

Reasoning for Design and Impact on User Experience:

Item-based recommendation methods are rooted in the idea that articles with similar characteristics will likely cater to similar user interests. Rather than relying solely on user behavior, this approach digs deeper into the content itself to ensure recommendations are content-driven. After users find the direction they are interested in through the first few methods, they need to read many articles in a particular field and conduct in-depth research. At this point, broad recommendations will no longer apply; instead, we should recommend similar articles on the same topic to allow users to gain a deeper understanding of the field. As a result, when users read specific articles, the system will identify the inherent characteristics of these articles, thereby providing more targeted content delivery. Therefore, we chose such a recommendation method. This approach ensures that recommendations remain relevant even if a user's preferences change over time. The result is a platform that continuously adapts and delivers articles matching users' reading preferences, resulting in a refined user experience.

5. Mix-based Recommendation Algorithm

Basic Introduction:

The Mix-based recommendation algorithm combines the strengths of the Popular-based and Item-based algorithms. This method first identifies the top 20 popular articles and then, from this subset, recommends the top 5 articles that are most similar to the user's preferences based on the Item-based recommendation criteria. This ensures that not only is the content tailored to the user's interest, but

it also enjoys a degree of popularity, likely indicating higher quality or relevance.

Relevant Formula:

Popular Article Selection:

$$P = \text{Top\_20\_Most\_Popular\_Articles}() \tag{9}$$

Where P denotes the set of the 20 most popular articles based on user interactions.

Mix-based Article Recommendation:

$$R(u) = \text{Top\_5\_Most\_Similar\_To}(u, P) - \text{Articles\_Already\_Seen\_by\_User}(u) \tag{10}$$

Where $R(u)$ denotes the recommended articles for user u. It selects the top 5 articles from the set P that are most similar to articles liked by user u, and that the user hasn't already seen.

Algorithm Flowchart:

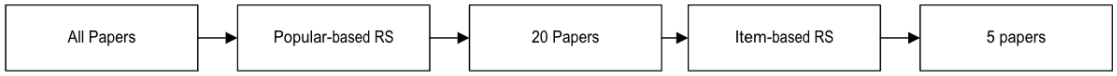


Figure 15: Mix-based Recommendation

Reasoning for Design and Impact on User Experience:

Mix-based recommendation algorithms are designed to balance broad appeal and individual relevance. By selecting well-reviewed articles first, the system ensures that a larger audience has reviewed the quality or relevance of the content. Subsequent fine-tuning using item-based criteria ensures that these popular articles match individual user preferences. The purpose of adding this algorithm is to cater to user needs while ensuring the quality of recommended articles.

2.5 APIs

APIs, or Application Programming Interfaces, have become a cornerstone of modern software design. They allow systems to communicate with one another, making it feasible to integrate complex functionalities without needing to rebuild from scratch. The choice of APIs can significantly influence a system's performance, user experience, and adaptability. In our system, we have chosen to integrate two robust APIs - the Datamuse API and the SMMRY API. These were selected not only for their operational excellence but also for their direct relevance to our platform's objectives, providing enhanced user experience and functionality.

2.5.1 Chat Robot

The Datamuse API is a versatile word-finding query engine, renowned for its ability to pull up synonyms, related terms, and context-specific words. Delving a bit deeper, a user might, for instance,

make a query using an endpoint like https://api.datamuse.com/words?rel_syn=duck, and the API would fetch words synonymous with 'duck'. Within our platform's ecosystem, this API takes on a pivotal role, fueling the capabilities of our ChatRobot feature. When users punch in a keyword, the ChatRobot, powered by Datamuse, showers them with a spectrum of word suggestions. It's not just about offering alternatives; it's about enabling users to refine their search, to get more granular with their queries, and to broaden their horizons by exploring a wider keyword landscape. The integration of Datamuse essentially redefines the search experience, making it more adaptive, intuitive, and tailored to individual user preferences.

When users search, they're immediately presented with a broader range of relevant content, enhancing their search journey. These tailored suggestions allow for more personalized browsing, ensuring content aligns with their preferences and interests. The efficiency of their search is also notably improved, reducing time spent on irrelevant content. Furthermore, this exposure to new and related terms serves as an educational tool, enriching their vocabulary and knowledge in the topic at hand.

2.5.2 Paper Summary

On the other side of the spectrum is the SMMRY API, a master at condensing vast tracts of web content into succinct, digestible summaries. It operates on a principle of efficiency – when prompted, it scans through a webpage, excising superfluous information and crafting a compressed version that captures the essence in a few articulate sentences. In our system's workflow, this API springs into action whenever users bookmark or earmark a webpage. At this juncture, the system liaises with the SMMRY API, generating a crisp summary of the content embedded in the saved page. But what truly makes this feature a standout is the sheer convenience it bestows upon users. Imagine being liberated from the chore of wading through lengthy articles. With just a brief glance at the summary, users can gauge an article's pertinence, ensuring they invest time only in content that resonates with their interests. It's not just about efficiency; it's about reshaping content consumption, making it more aligned with today's fast-paced digital rhythms.

Incorporating the SMMRY API has revolutionized how users interact with saved articles. They can now quickly discern the relevance of a piece through its summarized version, allowing them to consume more content in less time. This efficient system not only aids in decision-making about which articles warrant a thorough read but also streamlines their experience. Summarized content declutters the user's interface, fostering a cleaner, more organized reading environment.

2.6 Data Collection

2.6.1 Overview

In the world of modern search engine technology, data collection is a key component. In order to efficiently collect network data, our system uses multi-threading for data collection. The structure depicted illustrates the synergy between web servers, the Internet, and our search engine. A web server holds a large number of Internet web pages and transmits data over the Internet to search engines. First, the system extracts keywords and URLs from the acquired web data, a key step in facilitating understanding and indexing of web content. Secondly, the extracted keywords and URLs are stored in the database for subsequent retrieval and use. To improve the efficiency of data collection, we deployed 10 threads to download data from the web server simultaneously. This means our search engine is able to access data from ten different sources simultaneously, significantly increasing the speed of data processing. This strategy not only simplifies subsequent data queries but also ensures that our search engine can quickly find relevant web links when presenting search results based on user-defined keywords.

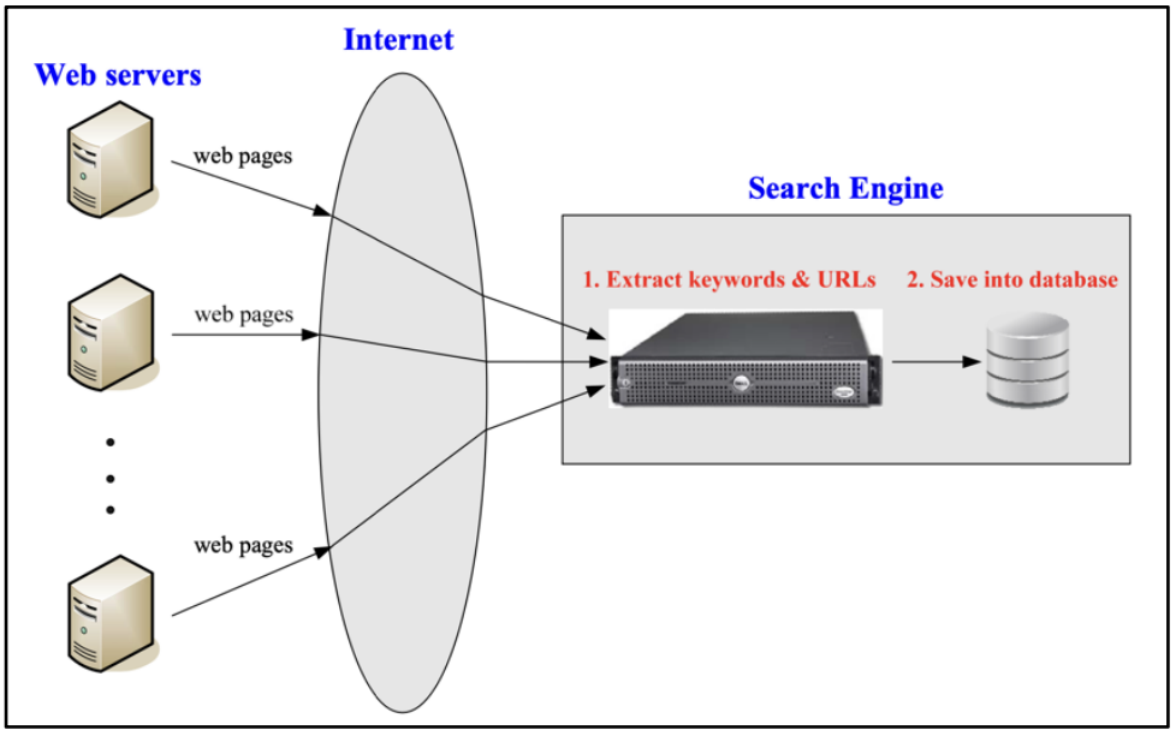


Figure 16: Data Collection: Web Structure

2.7 Data Collection Overview

The single-threaded data collection process employs a meticulously crafted algorithm to ensure accurate and efficient data retrieval within our search engine system. Our design establishes two crucial data structures: the URL Pool and the Processed URL Pool.

Commencing the collection process, the user is initially prompted to input a seed URL, which acts as the foundational point for the web search. Upon initialization, the seed URL is allocated to the URL Pool, and the Processed URL Pool is designated as empty. The algorithm then extracts a URL from the URL Pool, ensuring it's added to the Processed URL Pool and obtaining the associated file. If the procured file is in HTML format, the system delves deeper into its content. It starts by extracting all the words, ensuring that none of these words are part of a pre-defined blocklist. Any word that passes this filter retains its article features in our system. Alongside words, the algorithm also scouts for URLs within the HTML file. If these URLs adhere to specific criteria — not being present in both the URL Pool and Processed URL Pool and the total count in the URL Pool being less than U — are added to the URL Pool.

The data collection cycle is reiterated until the quantity of URLs in the Processed URL Pool reaches the threshold of the user-defined parameter V. Once this limit is hit, the algorithm ceases its operation, marking the completion of the single-threaded data collection process.

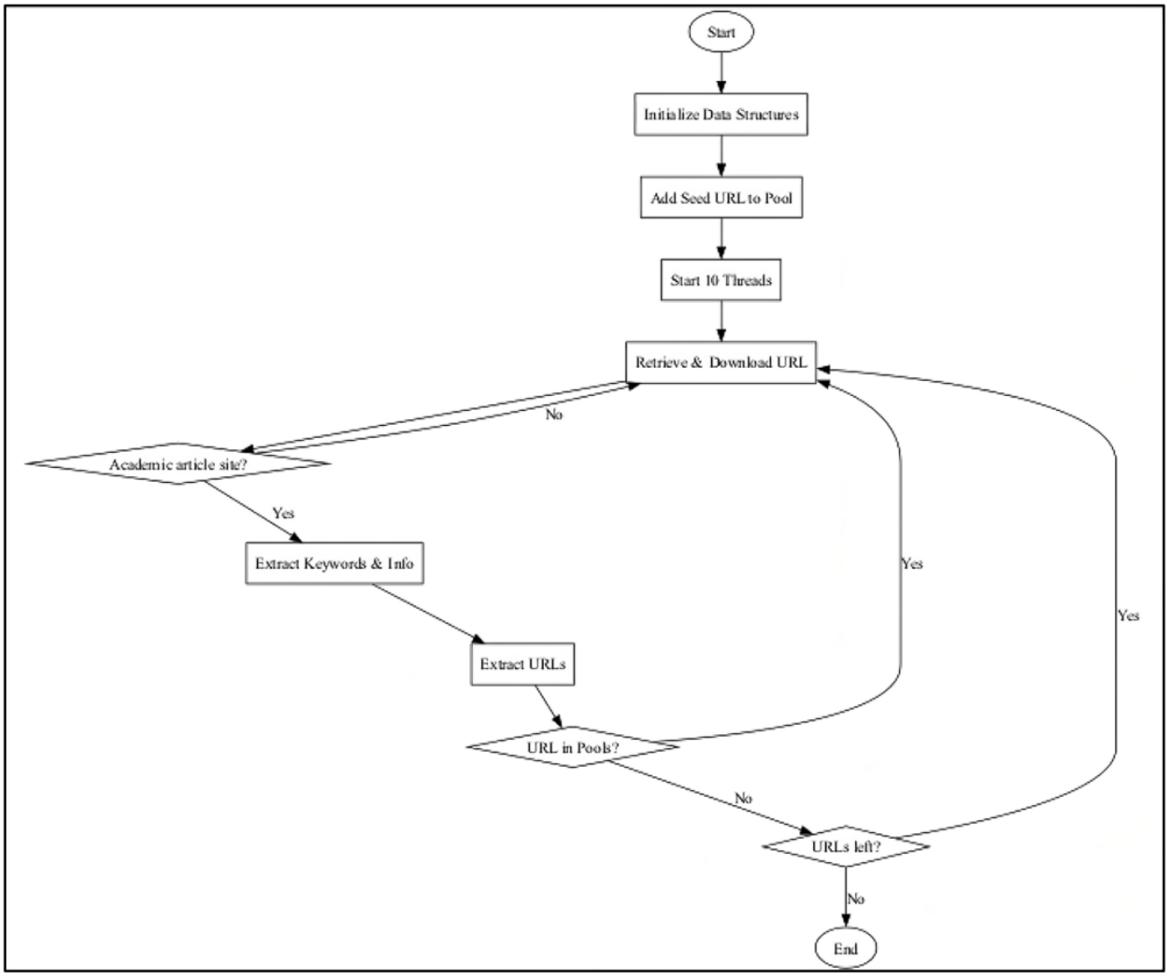


Figure 17: Data Collection: flowchart

2.7.1 Data Cleaning & Processing

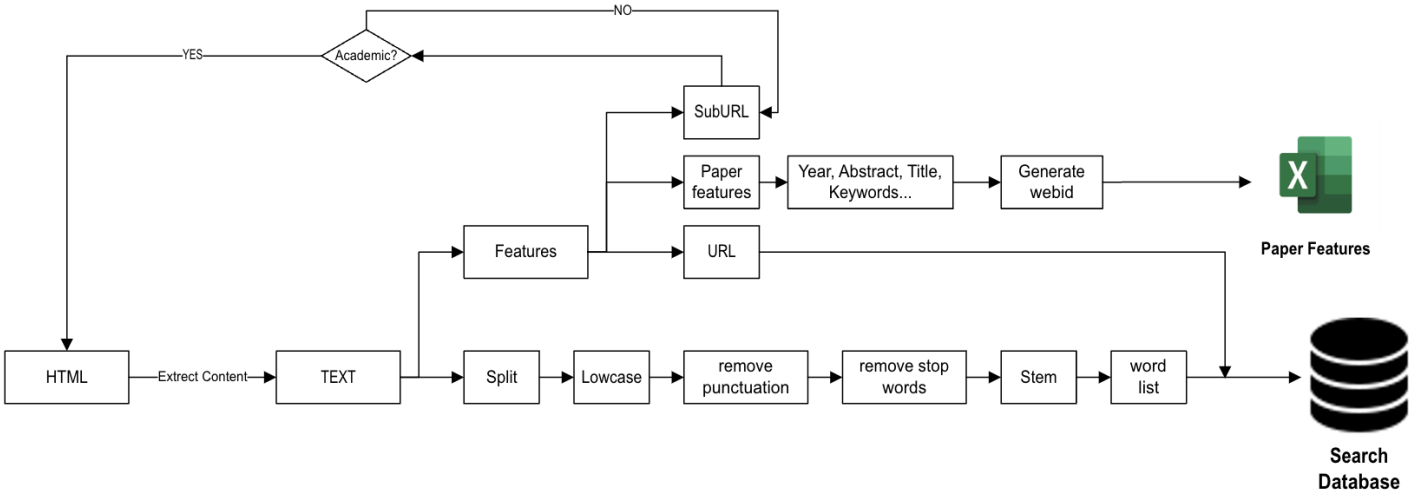


Figure 18: Data Cleaning & Processing

As shown in the figure above, the flowchart describes the process of cleaning and processing the data. First of all, the content of an HTML web page is downloaded in its entirety. On the one hand, all the contents will be subjected to a series of operations such as word splitting, lowercaseization,

etymologizing, removing punctuation, removing stop words, etc. to obtain the most critical words in the web page and save them as a list; on the other hand, the system will extract the key information of the article according to the tags of the HTML, and the sub URLs in it, if they are related to the scholarly articles will be used as the objects for the next processing; otherwise they will be discarded directly. If the sub URL is related to the academic article, it will be used as the next processing object otherwise it will be directly discarded. Other features such as year of publication, abstract, core words, etc. will also be extracted, and this information will be added to the web id and stored directly for later webpage information extraction and recommendation algorithms. Finally, the URL of the current page will be associated with all the keywords extracted in the previous step and saved in the search database, and if the user searches for these keywords, a link to the page will be returned.

2.8 UI Design

2.8.1 Design Concept

Our UI design emerges from a guiding philosophy emphasizing clarity, simplicity, and intuitiveness. We aspire to furnish users with an interface that is not only easy to navigate but also facilitates effective and rapid interactions. This commitment to user-centric design is manifested in our choices, which prioritize straightforwardness and efficiency.

2.8.2 Technology

To build the robust foundation of our interface, we leveraged the power of Java combined with SpringBoot. This combination not only ensured the reliability and performance of our backend operations but also seamlessly integrated with our frontend components. Furthermore, to enhance the presentation of our HTML pages and ensure they were both visually appealing and responsive, we incorporated Bootstrap. This integration not only beautified our pages but also guaranteed that they would maintain their elegance and functionality across various devices, ensuring a consistent experience for all users.

2.8.3 Color

Regarding aesthetics, we took inspiration from Google's classic color palette. This decision was driven by a desire to offer users a familiar and professional visual experience. The universally recognized and trusted Google colors evoke a sense of reliability and sophistication, setting users at ease and underscoring our platform's commitment to quality.

2.8.4 UI Design Showcase

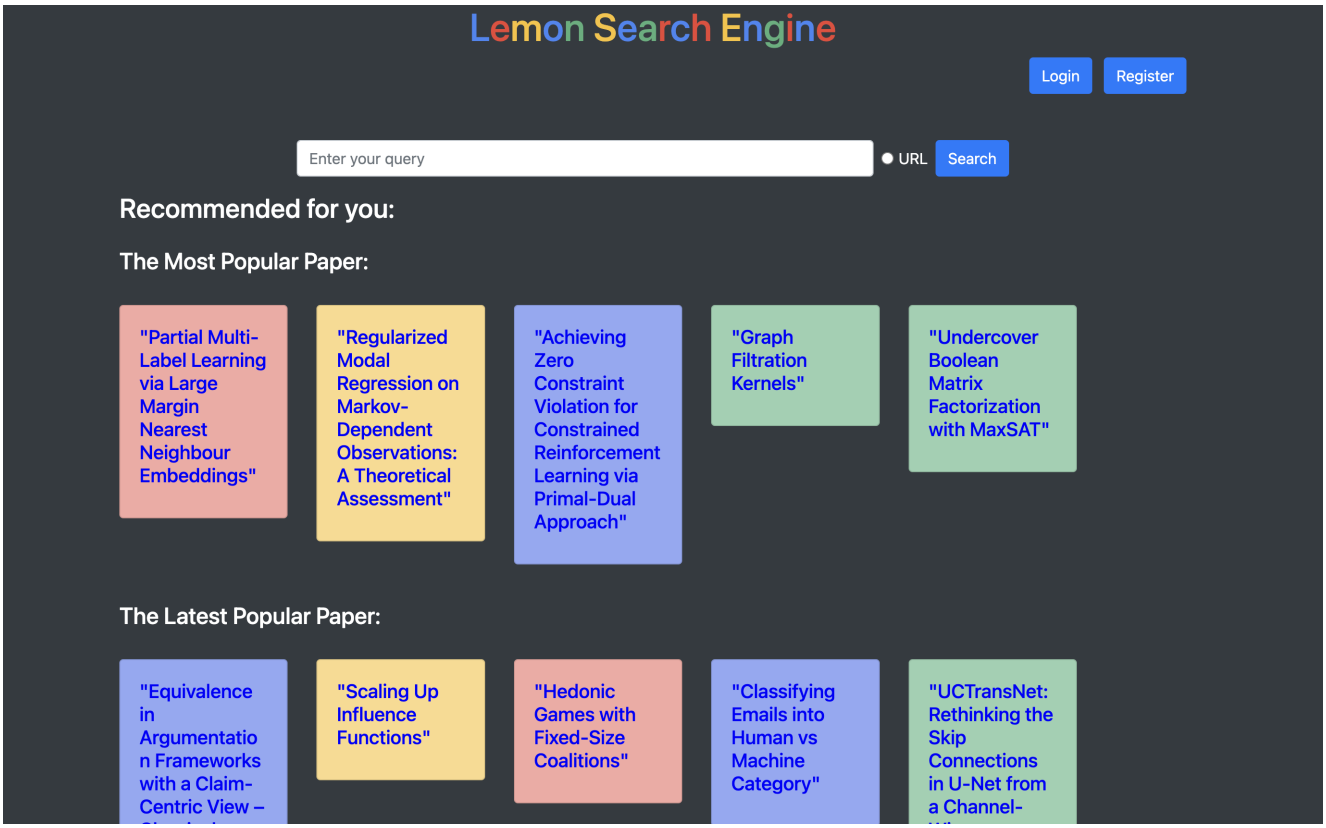


Figure 19: UI Design Showcase1

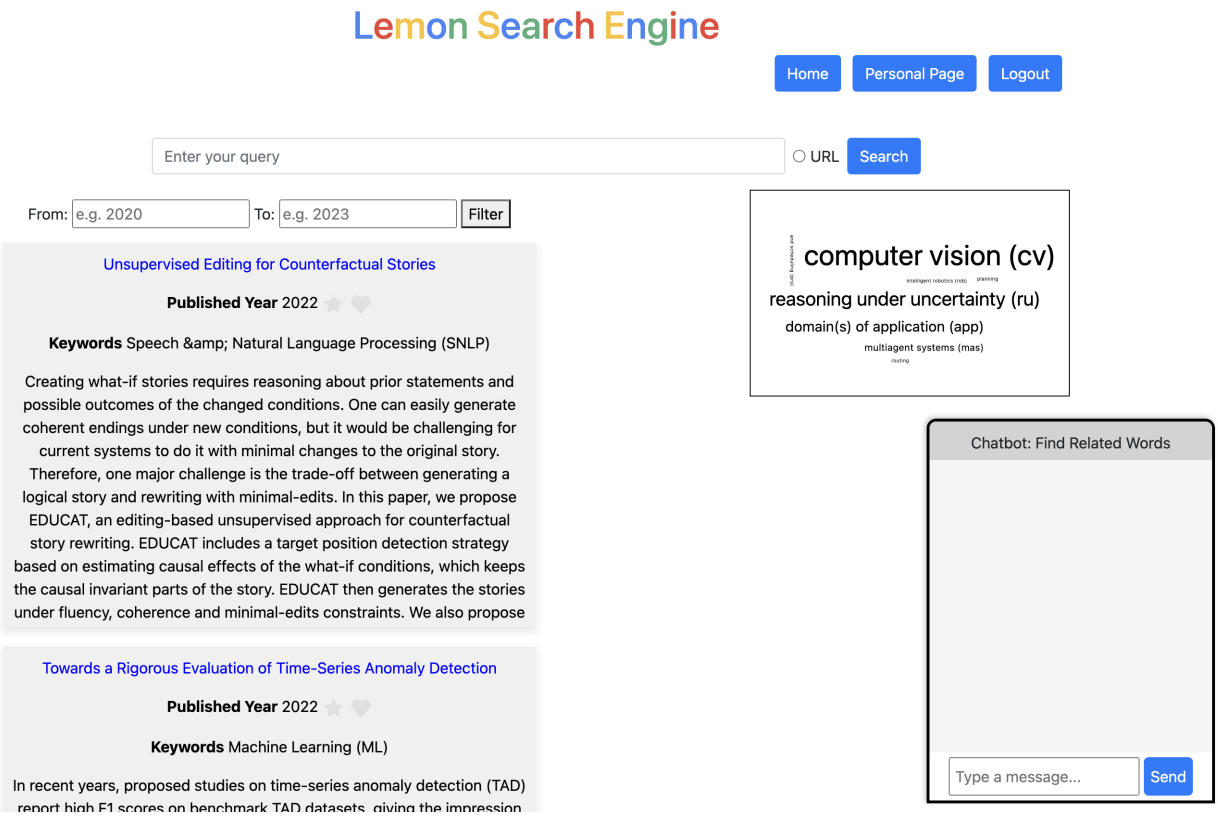


Figure 20: UI Design Showcase2

3 Project Implementation

3.0.1 Tools & Technologies

During the development of the system, Java served as the primary programming language, complemented by the SpringBoot framework to facilitate the back-end processes. The entirety of the development was conducted within the IntelliJ IDEA environment, a choice that streamlined the coding and testing processes due to its integrated features tailored for Java projects. To enhance the system's capabilities, two APIs were integrated: Datamuse API, which offered advanced word and phrase querying, and the SMMRY API, which provided automated text summarization. In terms of data handling, a conscious decision was made to prioritize data security. Rather than relying on third-party database tools, all data was self-collected and stored locally in .ser and .csv formats. This ensured total control over data access and integrity. For the front-end design, Bootstrap was incorporated to refine the user interface, allowing for a responsive and aesthetically pleasing layout. The color scheme was intentionally chosen to echo the traditional Google colors, providing a familiar and intuitive visual experience for users. For a detailed introduction, see Appendix 6.3.

3.0.2 Challenges

During the development process, I mainly encountered the following difficulties:

No.	Challenge Name	Description
1	User Requirements Analysis	Analyzing the needs and preferences of potential users posed a significant challenge. With diverse academic interests and varying familiarity with research domains among Ph.D. aspirants, determining a universal set of requirements became an intricate task. An extensive survey had to be conducted prior to the design and development phases to ensure the platform would be universally accepted and beneficial.
2	Data Collection	During data collection, an unexpected difficulty arose due to the character limitations of Excel cells. Extended data, such as detailed abstracts of academic papers, would often overflow into the adjacent cells, leading to a chaotic and disorganized dataset. To counter this, the decision was made to store abstracts at the end of the data rows, and in some instances, truncating portions of the abstracts to ensure they didn't disrupt the organization of other vital data.
Continued on next page		

Table 16 – continued from previous page

No.	Challenge Name	Description
3	Absence of Suitable AI Model	Another challenge was the lack of readily available AI models tailored for multi-dimensional analysis of academic abstracts. The ideal scenario would involve breaking down abstracts into distinct sections such as overview, advantages, limitations, and areas of potential improvement. However, without access to a pre-trained model that could accomplish this, a shift in strategy was required. Consequently, a more generic text summarization approach was adopted, although it meant sacrificing some depth in the analysis.
4	Recommendation System	Incorporating the recommendation system presented its own set of challenges. Balancing the weightage given to different parameters, like user history, click rates, and similarity of content required iterative testing. Fine-tuning the algorithm to ensure relevance while also introducing users to newer topics of potential interest was a demanding task.
5	Interface and Functionality Integration	Designing an intuitive and responsive front-end interface, while integrating functionalities like the term optimization chatbot and the word cloud visualization, tested the system's performance and user experience. Ensuring that these features ran smoothly, especially when processing large amounts of data in real time, necessitated thorough optimization and testing. The goal was to deliver a seamless experience to users without compromising on the speed or reliability of the platform.

4 Project Performance & Validation

4.0.1 Testing Strategy

No.	Test Method	Purpose
1	Black-box Testing	Primarily to test if the code runs smoothly. Start the program to test all functionalities.
2	White-box Testing	Test if there are any issues within the code, inspect the code at its level, and remove bugs, and redundant or unnecessary codes.
3	AB Testing	Divide the test users into two groups and compare the experience between the new and old system.

4	F1 Score	Test the recommendation effect of the algorithm within the system.
5	Target User Interview	Identify one target user for a direct interview to uncover system experiences and shortcomings.
6	SWOT Business Analysis	Analyze the strengths, weaknesses, opportunities, and threats of the system.

4.0.2 Test Results

1. Following rigorous scrutiny during the black-box and white-box testing phases, issues that had been detected within the code were diligently rectified. As a result of these extensive evaluations and corrections, the system is now functioning seamlessly and can operate without hitches.

2. In the case of the AB testing, it was discerned that 60% of the tested demographic favored the original version of Google Scholar over the system I developed. While my platform introduced a plethora of innovative features, the limitations of its database were evident as it couldn't cater to all the user needs. This feedback underscores the imperative for future enhancements, particularly augmenting the database capacity, to more comprehensively address user requirements.

3. In the F1 test, I found 15 testers and recommended 183 documents to them. The test results are as follows: Precision = 0.72, Recall = 0.78, F1 = 0.7475. An F1 score of 0.7475 is a reasonably good score, especially when dealing with imbalanced datasets. The score indicates that the system has a harmonious balance between precision and recall, but there is still room for improvement to reach optimal performance. Therefore, my recommendation The system algorithm is relatively good.

4. In order to further evaluate this system, I found a student who was preparing to study for a PhD to use this system and asked him to provide his experience and suggestions for improvement. Below is a summary of the interview:

He acknowledged and appreciated the extensive features integrated into the platform, noting its capabilities in offering recommendations, optimizing search terms, and visualizing core keywords via a word cloud. The students found the functionalities, such as liking, bookmarking, and viewing their activities, highly relevant and time-saving, especially the feature that summarizes articles into a single statement for efficient scanning.

However, despite the impressive suite of features, he felt that because of the small database sometimes made it difficult to find very specific or niche academic papers. Nonetheless, the overall sentiment was positive, with the agreeing that the platform's features are well-aligned with the needs of a Ph.D. student in terms of selecting, searching, and summarizing academic articles. He remains hopeful that the database size will grow, making this platform an even more powerful tool in the future.

5. In order to explore its business value and future improvement directions, we created a SWOT analysis chart. SWOT is shown below:

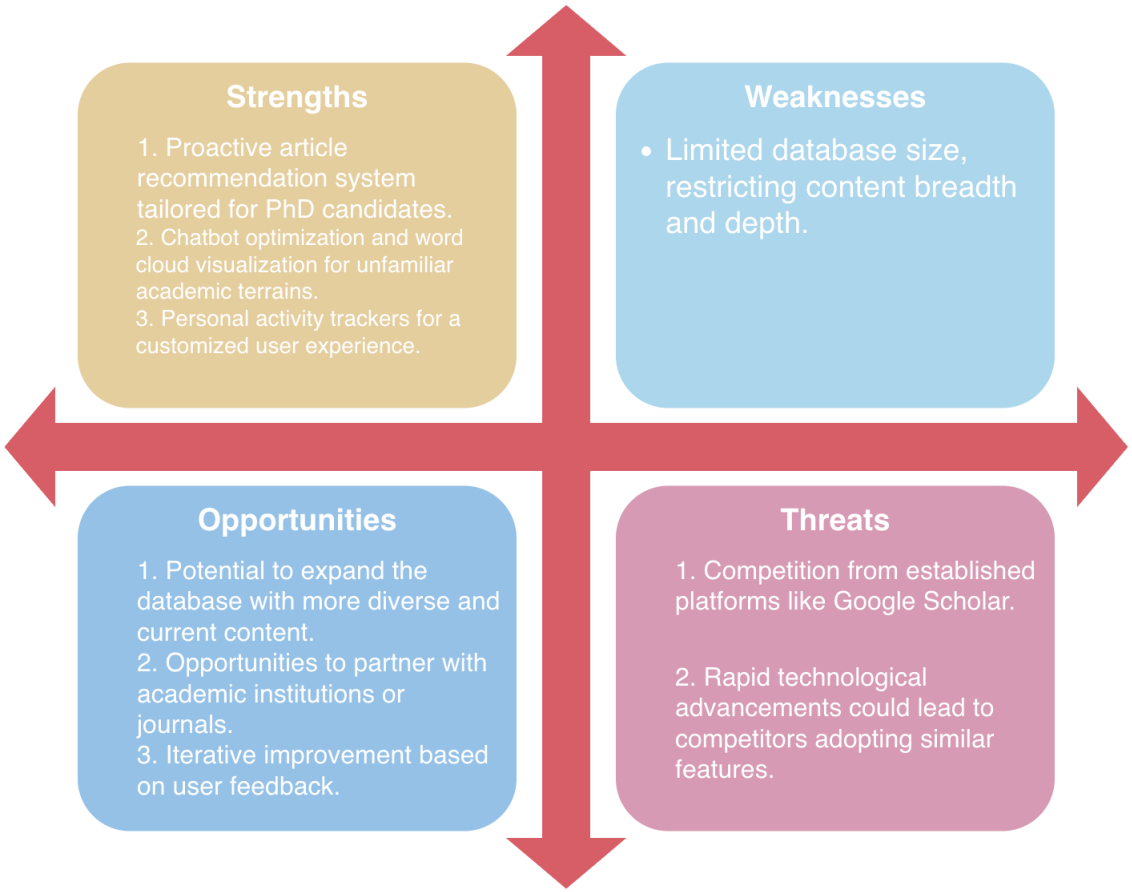


Figure 21: SWOT

5 Project Conclusions & Improvements

In reflecting on the academic search engine project, it's evident that the platform boasts a series of innovative features tailored specifically for prospective PhD candidates, filling a niche that most generic search engines overlook. The proactive recommendation system, combined with the chatbot-driven term optimization and visualizations like word clouds, provided users with an enriched and guided academic exploration experience. These features not only introduced them to trending research areas but also simplified their navigation through unfamiliar academic terrains. Moreover, integrating user engagement tools such as liking, bookmarking, and activity tracking personalized the user journey, while the automated article summary feature ensured efficient content consumption. However, a recurring concern among users was the platform's limited database size, which sometimes restricted the depth of content available. Additionally, while the recommendation system was novel, there's potential for refining and optimizing the algorithm to ensure even more relevant content suggestions. Moving forward, the immediate improvements for this platform would be expanding the database to encompass a wider range of scholarly articles and further enhancing the recommendation engine for greater accuracy and personalization.

6 Appendix

6.1 Project Proposal

Date of proposal: Oct. 2nd, 2023

Project Title: "Lemon" - An Academic Knowledge Platform

Group Members:

Bian Weizhen A0285814W

Client:

Tom, a bright and ambitious student, is gearing up to embark on his PhD journey. As he delves into the realm of academic research, he quickly realizes the vastness and complexity of available literature. With so many articles and journals at his disposal, determining where to start or which papers are most pertinent to his budding interests becomes a daunting task. While Google Scholar offers a broad database, it falls short in personalizing the experience for users like Tom who are in the nascent stages of their academic pursuits. He finds himself overwhelmed by the plethora of papers, yet unable to discern which ones are truly relevant or aligned with emerging trends. Furthermore, Tom is unfamiliar with many academic terminologies and often finds himself guessing keywords or terms to input, hoping to stumble upon the right content. What Tom truly needs is an intuitive system that not only introduces him to the current hot topics in his field but also understands and adapts to his evolving interests. A platform that can proactively recommend articles, assist in refining search terms, and provide a quick summary of papers would be invaluable for someone in Tom's shoes. Such a system would not only save time but also ensure that Tom's initial foray into academic research is productive.

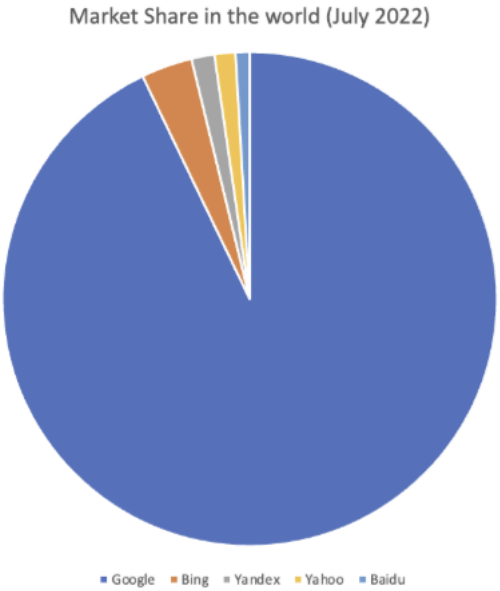
Survey & Requirements Identification

No.	Question	Response
1	How often do you search for academic articles?	Daily.
2	Do you often struggle to find the right articles that align with your interests?	Yes, it's quite challenging at times.
3	How do you currently discover new research topics or areas?	Mainly through reference or random browsing or going to other social platforms such as Zhihu to find reading suggestions.
4	Would a platform recommending articles based on your interests be beneficial?	Absolutely, it would save a lot of time and help me focus on relevant research.
5	How often do you feel overwhelmed by the vast number of articles available?	Very often. It's hard to know where to start or which articles are the most relevant to my research interests.
6	Do you believe an interactive tool suggesting search terms would aid your research process?	Definitely. This helps me directly discover the general scope of the search results, and if it doesn't meet my expectations, I can directly search again. Saved me a lot of time reading them one by one.
7	How beneficial would visual representations (like word clouds) be for understanding search results?	Extremely beneficial. It provides an instant overview and helps quickly identify the relevance of search results.
8	Would you appreciate a platform that tracks your reading habits and provides insights based on them?	Yes, it would help make my research more structured and directed.
9	How important is having a summarized version of articles for you?	Very important. With the plethora of articles out there, having summarized versions would aid in efficient screening.
10	Do you feel existing platforms sufficiently cater to your academic search needs?	Not completely. There's room for improvement, especially in personalized recommendations and insights.
11	Are there any other platforms that address your aforementioned needs?	Not that I'm aware of. Most platforms offer general search capabilities, but none with the personalized approach I desire.

Objective

Provide a literature search recommendation platform for students preparing to study for PhD.

Market Research



Name	Technology	Advantages	Disadvantages
Semantic Scholar	User preference analysis	Recommendations based on user's interest	Requires initial input if new user
Google Scholar	Statistical models	Comprehensive relevance determination	Might require extensive data like publishing history
Papers With Code	Trend analysis	Focuses on latest CS research	Only contains code-related works
Connected Papers	Knowledge Graph	Comprehensive connections via citation graph	Might require understanding of graph representation
Paper Weekly	User history analysis	Personalized AI research recommendations	Specific to AI domain

Potential Application

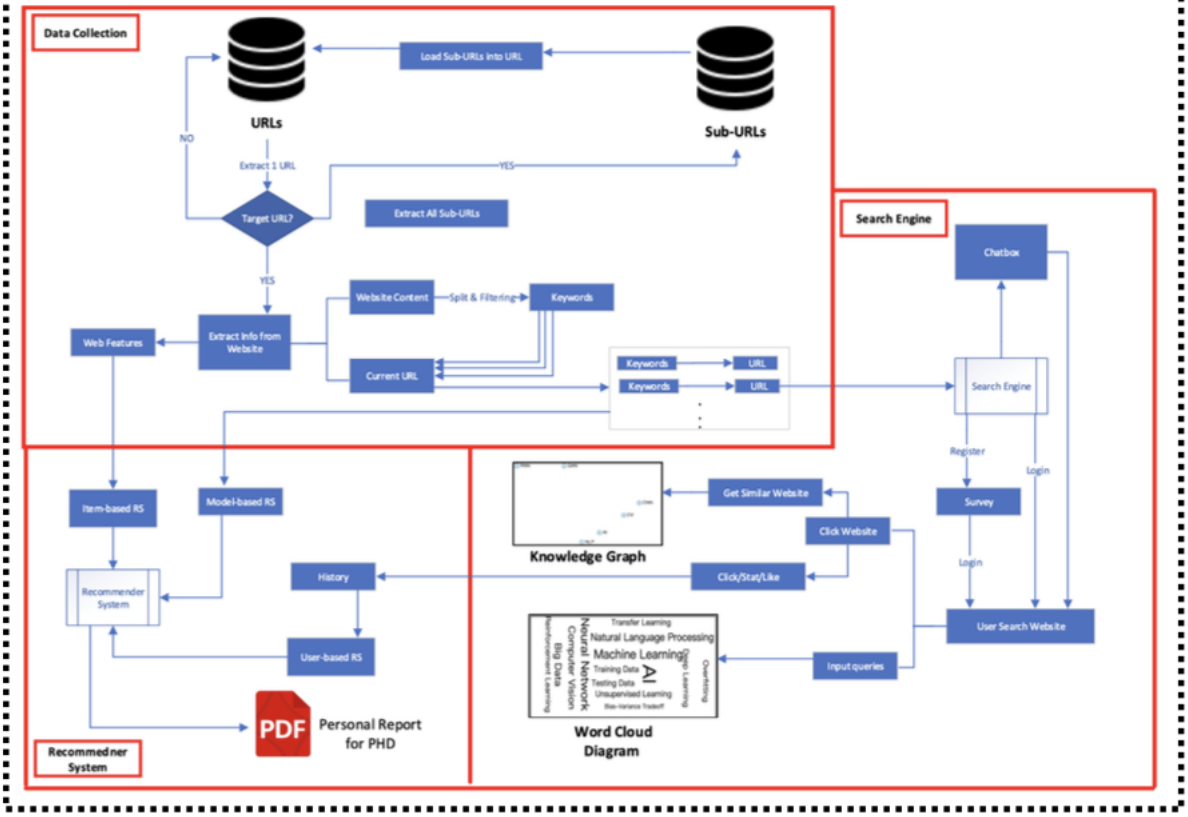
[Research Management] Students and professors at the National University of Singapore can rely on the platform to obtain literature recommendations and directly obtain further summaries of the literature, saving time. With the platform, students will focus on their areas of interest.

[PHD Guide] It can be further developed to generate a student's PHD report and find a suitable tutor for him.

Project Descriptions:

System Overview

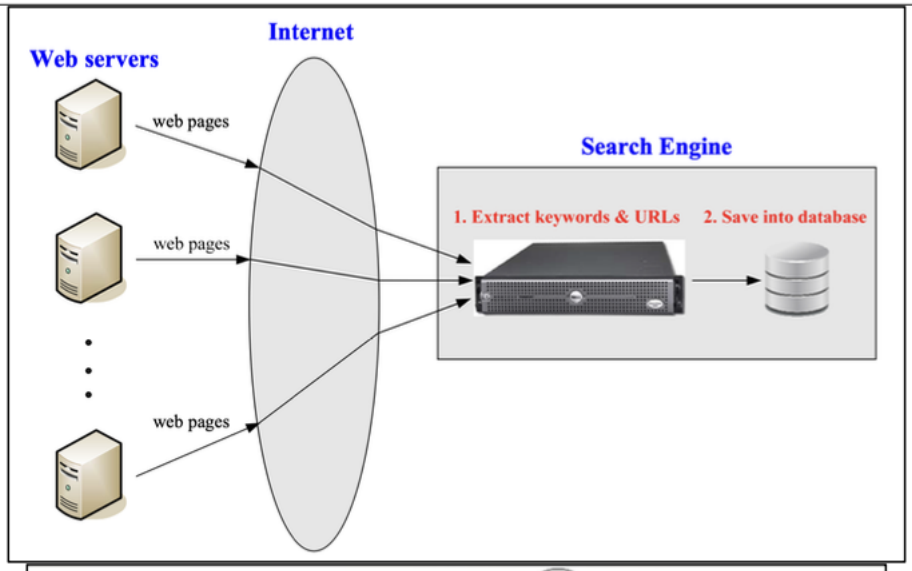
The system collects academic articles, maintaining a capacity-limited URL pool for efficiency. It enhances Google Scholar with personalized searches based on user interests and activity, integrating features like word clouds, knowledge graphs, and an assisting chatbot. Additionally, a recommendation system offers tailored article suggestions and allows users to generate a PhD recommendation report, leveraging user and item-based algorithms.

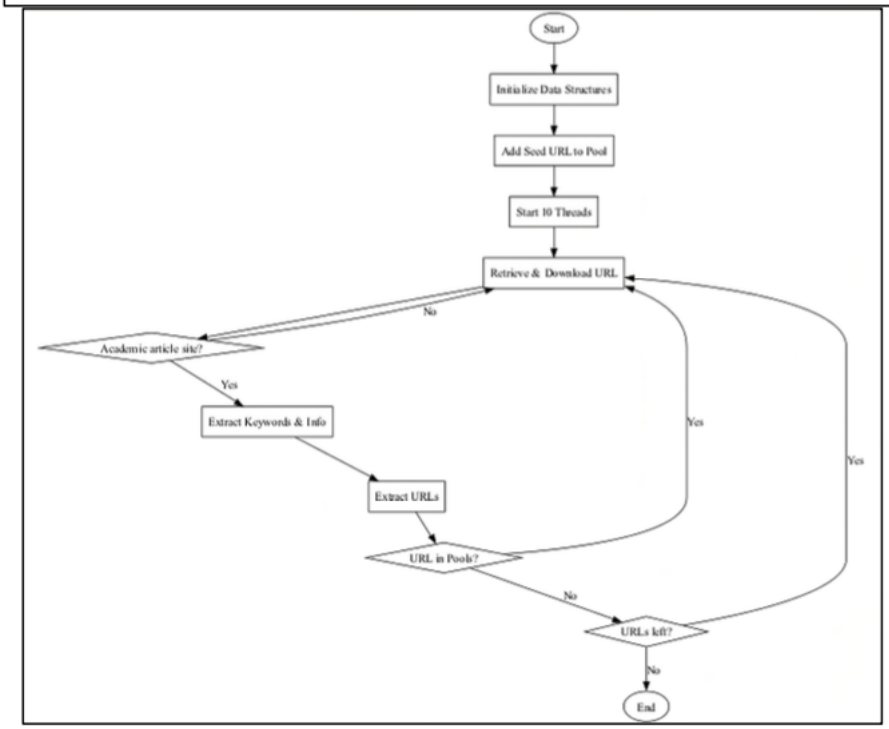


Data Collection

(1). Web Scraping

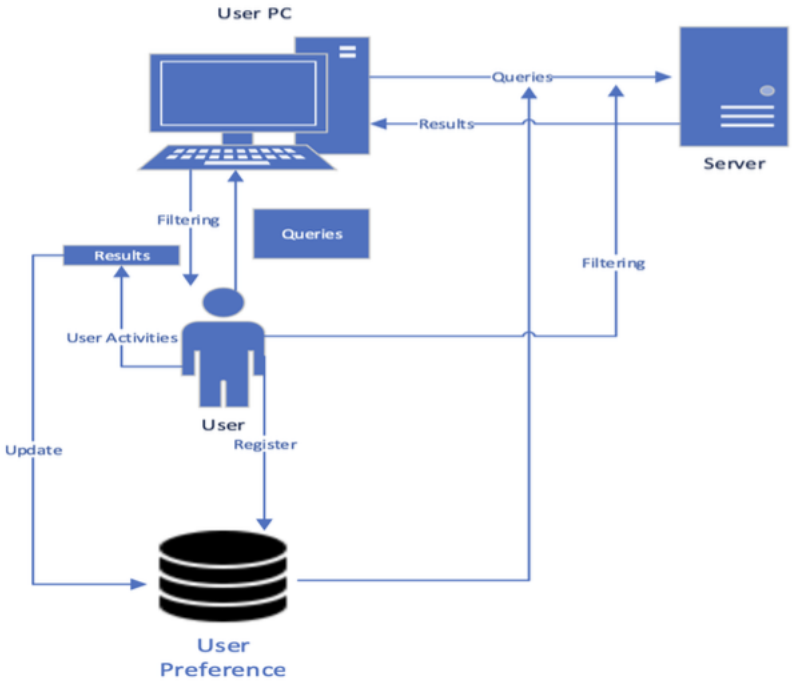
The process begins with the initialization of data structures and adding a seed URL to the pool. Ten threads are then initiated simultaneously for data capture. Each thread retrieves and downloads a URL. If the downloaded site is recognized as an academic article site, the system extracts keywords and other relevant information like year and abstract. Subsequent URLs found on the page are extracted. For each extracted URL, the system checks if it's already in the pools. If not, it is processed further. This loop continues until there are no URLs left to process. The process concludes once all URLs are processed. The procedure is shown as follow:





Search Engine

The user begins by executing a search, and in response, the search engine displays the relevant results. Recognizing the importance of a personalized search experience, our system allows the user to manually filter out certain websites from these results. After narrowing down the results with these filters, the user can further interact by clicking on a result to delve deeper, expressing interest through liking a specific result, or saving a particular result to their favorites for future reference. As these interactions occur, the system seamlessly captures them. This captured data is then processed in the backend, updating the database to reflect the user's actions and preferences, ensuring a continually improving and tailored user experience.

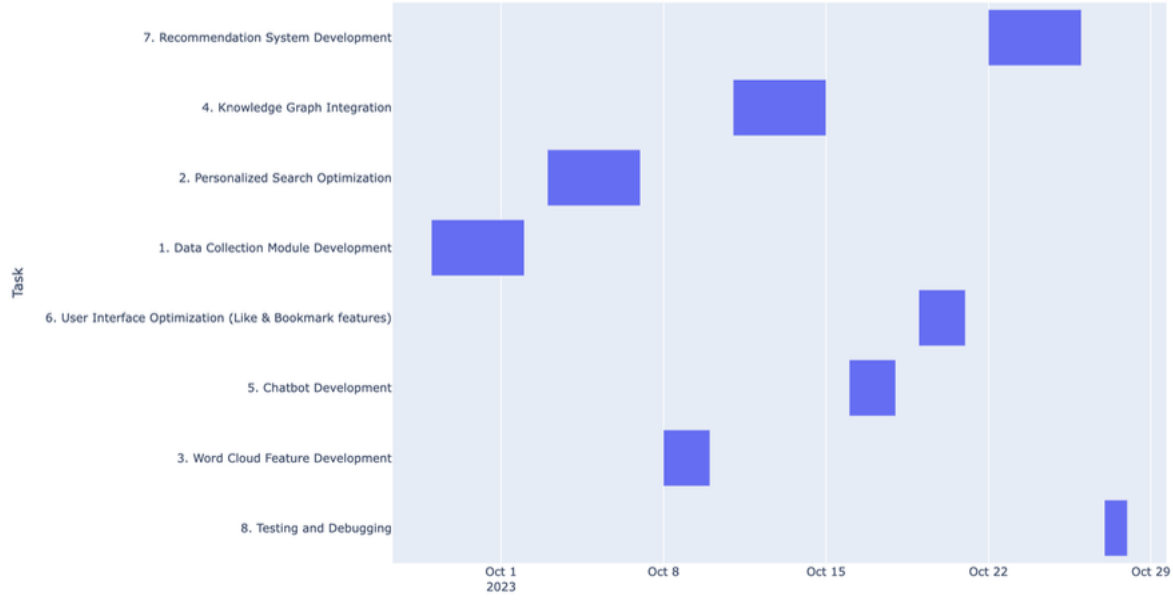


Recommender System:

1. **User-based.**
2. **Item-based.**
3. **Mixed based.**
4. **Popular based.**
5. **Year-based**

Future

Based on the tasks above and the estimated workload, we have drawn the following Gantt Chart to keep the development work on schedule:



6.2 Mapped System Functionalities

6.2.1 Topic

Introducing a groundbreaking academic search engine crafted with Java and SpringBoot that aims to revolutionize the current academic research landscape. According to project requirements, my system meets 3 topics at the same time.

Essentially, the engine is a quintessential tool for **knowledge discovery**, helping users push out top academic articles customized to their preferences. Whether users are looking for the most clicked, freshest, previously explored, or a mix of recommendations, this system has you covered. By showcasing the latest research trends, the platform provides information for students preparing to study for a PhD.

Understand the challenges newbies face in mastering academic terminology and integrate **optimization** capabilities into the search mechanism. State-of-the-art chatbots leverage this **optimization** to enhance user queries. Once entered, it suggests five closely related terms, ensuring users are always on the right track and enriching their search experience.

In addition to pure search, the system also employs recommendation algorithms by tracking user actions such as likes, bookmarks, and clicks. Through **recommendations**, it saves users the time of searching for papers and helps users find the direction they are interested in.

6.2.2 Technique Groups

Technologically, this system is a testament to the prowess of **Knowledge discovery & big data mining techniques**, ensuring that the vast corpus of academic literature is effectively indexed and retrieved. Furthermore, the incorporation of chatbots and recommendation engines stands as evidence of the system being designed using advanced **cognitive techniques or tools**. Lastly, the platform's ability to suggest closely related terms and optimize search experiences is a nod to **Business resource optimization**, ensuring that users gain the most valuable insights with the least effort.

6.2.3 Machine Reasoning

Our academic search engine epitomizes the concept of **Machine Reasoning**. By understanding user intent and bridging the gap between raw data and actionable knowledge, it intelligently recommends research paths. Whether it's suggesting related terms via the chatbot or curating paper recommendations based on user behavior, the system constantly reasons like a seasoned academic advisor.

6.2.4 Reasoning System

Our academic search engine is a prime example of a **Reasoning System** in action. It doesn't just present data, but proactively interprets user interactions to infer their academic interests and needs. Through the user's activities, the user's preferences are determined to help the user determine the research direction.

6.2.5 Cognitive System

Our academic search engine epitomizes the core principles of a **Cognitive System**. By integrating recommendation algorithms, search term optimization, and interactive user behavior tracking, the engine not only processes vast amounts of data but also learns and evolves in response to user interactions.

6.3 Installation and User Guide

6.3.1 Setup Instructions

Before you begin, ensure you meet the following requirements:

Prerequisites:

- Operating System: macOS (recommended).
- IDE: IntelliJ IDEA (recommended as it auto-identifies and installs project dependencies).
- Java: JDK 11.

Steps:

1. Install IntelliJ IDEA: Download and install from the official website.
2. Install Java JDK 11: Download JDK 11 from the Oracle official site and follow the on-screen installation instructions.
3. Clone the Repository:

```
git clone https://github.com/LuckyBian/ISY5001.git
```

After completing the above steps, your application should be up and running. If you encounter any problems, please double-check that all prerequisites are met.

6.3.2 Data Collection

For the functionality of the academic search engine, there are two primary ways to acquire the data:

- Using Pre-collected Data
- Web Scraping for New Data

Using Pre-collected Data

For those who wish to use the pre-collected dataset, here's what's included:

- `collectedData.csv`: It contains information on 1,000 academic articles, encompassing title, link, year, abstract, keywords, and more.
- `data_table.ser`: Stores the mapping between core terms and associated article information.
- **User Interactions:** `userActivity.xlsx` Records user click activities, `likeActivity.xlsx` Logs liked articles by the users, `starActivity.xlsx` Registers user's favorite or starred articles.

Web Scraping for New Data

If you want to collect more data or collect a new dataset, you can find `DataScraper.java` and adjust the parameters. For data crawling algorithms and a detailed explanation of parameters, please refer to `REPORT`. Note: The NUS campus network is unable to access some of the web pages and the data collection fails. Please connect to other networks such as cell phone hotspots for collection.

6.3.3 Running Application

If data collection is complete, ensure `data_table.ser` and `collectedData.csv` exist. Then, you can launch the application:

1. Find `SearchEngineApplication.java` in IntelliJ IDEA and click run.
2. Go to the following website:

```
http://localhost:8080/
```

6.3.4 Accessing the Application

My application is divided into visitor mode (not logged in) and user mode (logged in), which I'll describe next in relation to the interface for each.

Home Page:

Once you have successfully launched the app and accessed the web page, you will see the Home Page. As shown in the image below, you can enter keywords for a literature search, login, or register for an account. In addition, the platform will recommend some of the latest and hottest academic articles based on the FEATURES of the article (available even without login).

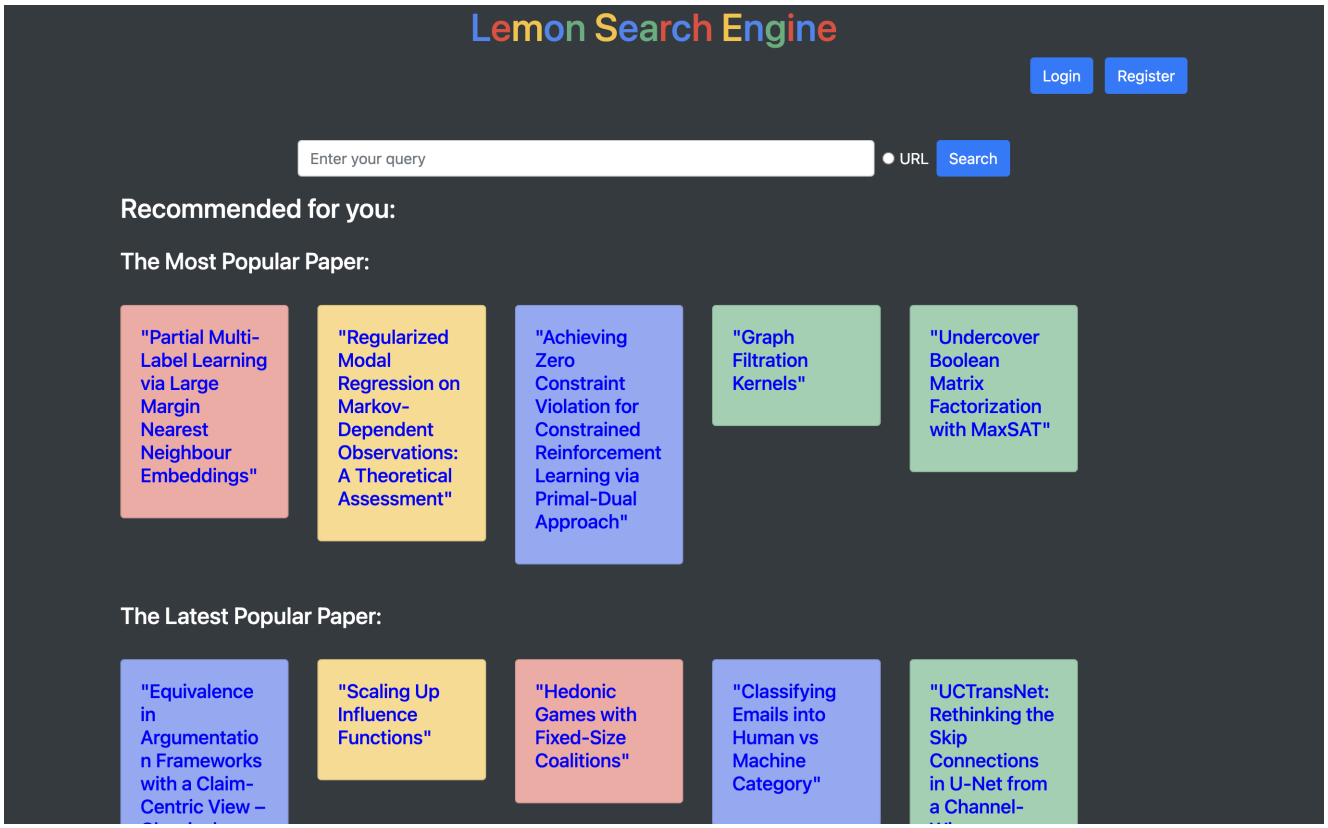


Figure 22: Home Page

Register:

If you don't have an account, you'll need to click 'Register' to sign up for a new account. The user id will be generated automatically and displayed in the popup window below. Please remember your user id as it will be used when logging in.

Figure 23: Register

Survey:

After clicking 'Register', you must complete a questionnaire to register your account successfully. As shown below, you need to select at least one article that interests you and submit it.

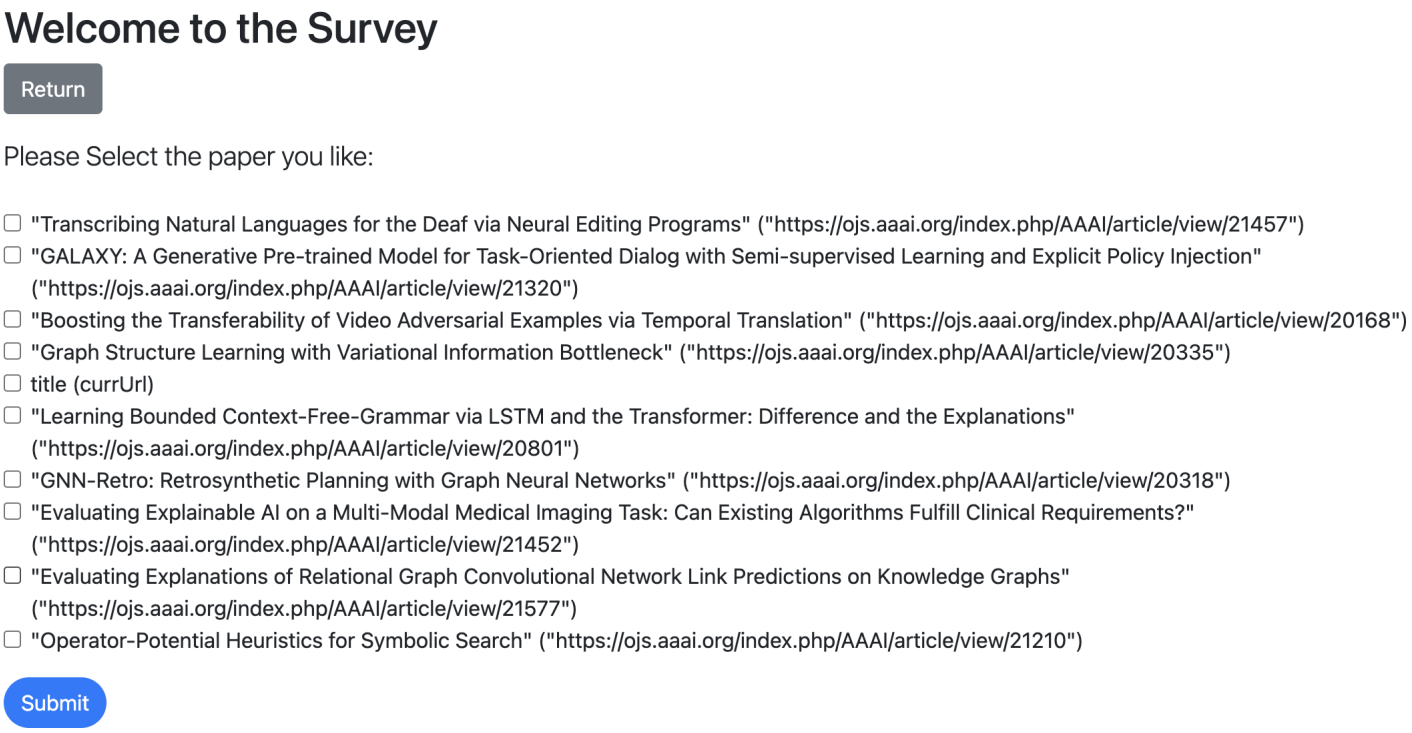


Figure 24: Survey

Search:

You can enter a keyword to search, if it matches the result, it will jump to the following page, if not, it will return to the Home Page. As shown in the figure below, the searched articles will be arranged according to the degree of match. The information about the article will be displayed on the left side of the screen. You can do some operations such as clicking on the page view, like (click on the heart icon), or favorite (click on the five-pointed star icon). All these activities will be recorded and analyzed to recommend academic articles for you. In addition, you can add a year range to re-filter the page information.

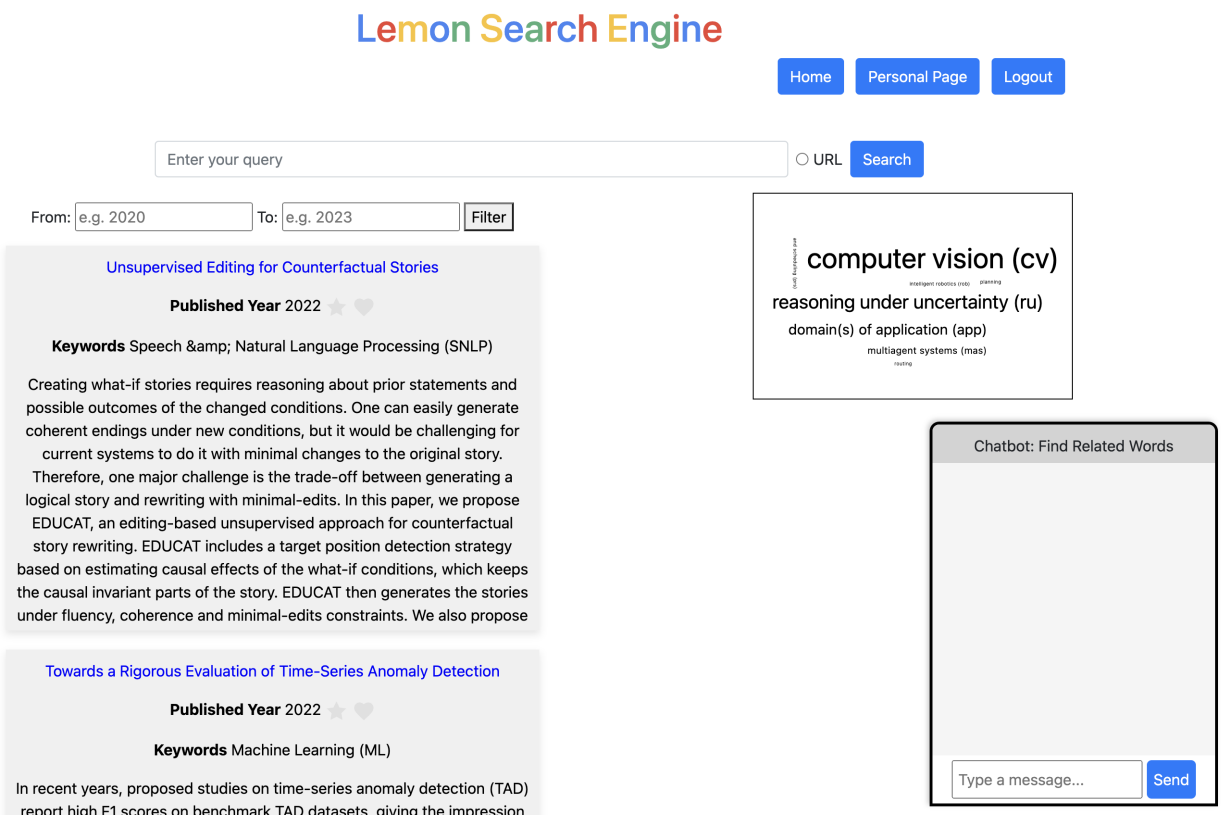


Figure 25: Search

Word Cloud Diagram:

As shown in the figure below, all the keywords in the search result articles will be organized, counted, and displayed on the word cloud map according to the frequency of occurrence (the more frequent, the larger the font size). Therefore, users can use the words in the word cloud to make a preliminary judgment whether the search terms they entered are accurate or not, and whether the matched articles are what they want or not.

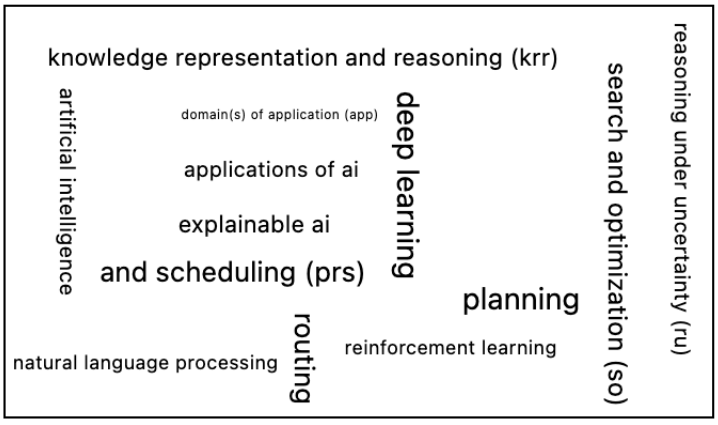


Figure 26: Word Cloud Diagram

ChatRobot:

Considering that some users are approaching an academic field for the first time, I've included the conversational bot shown below. The user can send a word to the bot and the bot will reply with 5

several words that are most relevant to the input word. With these hints, the user will find more precise search terms and thus optimize the terms.

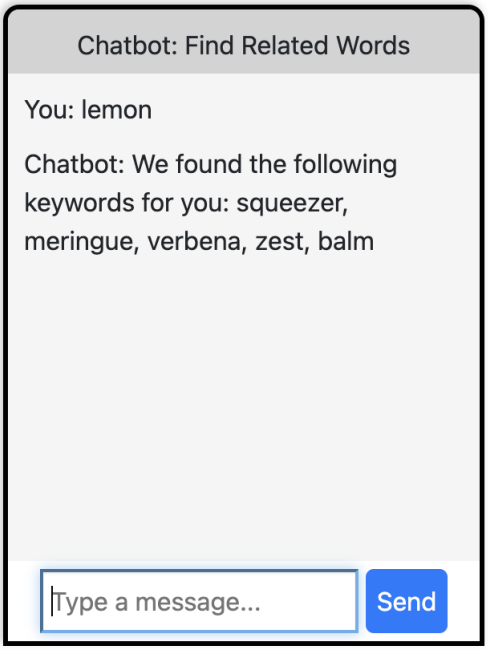
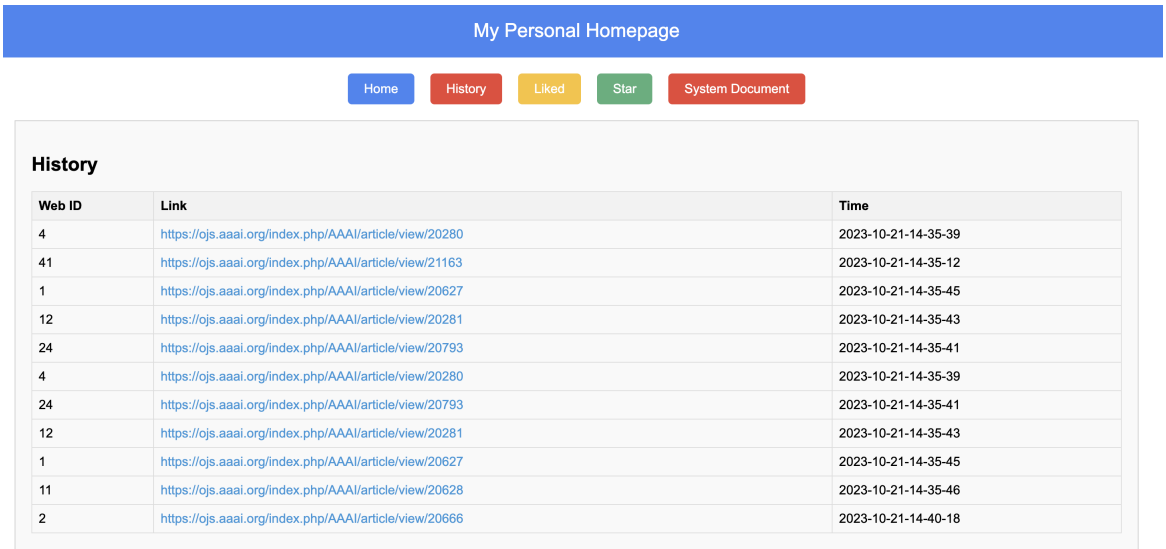


Figure 27: ChatRobot

Personal Page:

When a user has made a web page visit, liked an article or favorited an article, these records are handled differently and displayed in the 'Personal Page' (as shown below). The default display is the history and shows the time of the web page visit. Users can click on the icons above to switch between them to access Favorites, Likes, and User Guide.



Web ID	Link	Time
4	https://ojs.aaai.org/index.php/AAAI/article/view/20280	2023-10-21-14-35-39
41	https://ojs.aaai.org/index.php/AAAI/article/view/21163	2023-10-21-14-35-12
1	https://ojs.aaai.org/index.php/AAAI/article/view/20627	2023-10-21-14-35-45
12	https://ojs.aaai.org/index.php/AAAI/article/view/20281	2023-10-21-14-35-43
24	https://ojs.aaai.org/index.php/AAAI/article/view/20793	2023-10-21-14-35-41
4	https://ojs.aaai.org/index.php/AAAI/article/view/20280	2023-10-21-14-35-39
24	https://ojs.aaai.org/index.php/AAAI/article/view/20793	2023-10-21-14-35-41
12	https://ojs.aaai.org/index.php/AAAI/article/view/20281	2023-10-21-14-35-43
1	https://ojs.aaai.org/index.php/AAAI/article/view/20627	2023-10-21-14-35-45
11	https://ojs.aaai.org/index.php/AAAI/article/view/20628	2023-10-21-14-35-46
2	https://ojs.aaai.org/index.php/AAAI/article/view/20666	2023-10-21-14-40-18

Figure 28: Personal Page

Star:

It is worth noting that when the user clicks to collect, the system will further compress the article's abstract. As shown in the figure below, users can understand the general direction of the article just by reading one sentence. This saves users reading time and helps users quickly organize the article to determine the direction of interest.

Star

Web ID	Link	Summary
88	https://ojs.aaai.org/index.php/AAAI/article/view/21439	Compared with the pathological analysis and CT scans, the circulating tumor DNA methylation based approach is noninvasive and cost-effective, and thus is one of the most promising methods for early detection of lung cancer.]
53	https://ojs.aaai.org/index.php/AAAI/article/view/20637	Abstract Structural Equation/Causal Models are widely used in epidemiology and social sciences to identify and analyze the average causal effect and conditional ACE. Traditional causal effect estimation methods such as Inverse Probability Weighting and more recently Regression-With-Residuals are widely used - as they avoid the challenging task of identifying the SCM parameters - to estimate ACE and CACE. However, much work remains before traditional estimation methods can be used for counterfactual inference, and for the benefit of Personalized Public Policy Analysis in the social sciences.]
240	https://ojs.aaai.org/index.php/AAAI/article/view/20169	Abstract Vision transformers have demonstrated impressive performance on a series of computer vision tasks, yet they still suffer from adversarial examples.]

Figure 29: Star

Like:

The user's liking record is as follows.

Liked

Web ID	Link
11	https://ojs.aaai.org/index.php/AAAI/article/view/20628
18	https://ojs.aaai.org/index.php/AAAI/article/view/21285
43	https://ojs.aaai.org/index.php/AAAI/article/view/20795
88	https://ojs.aaai.org/index.php/AAAI/article/view/21439
53	https://ojs.aaai.org/index.php/AAAI/article/view/20637
205	https://ojs.aaai.org/index.php/AAAI/article/view/21186
240	https://ojs.aaai.org/index.php/AAAI/article/view/20169

Figure 30: Like

Recommendation

Finally, new recommendation methods will emerge when the user browses, likes, collects some websites, and returns to the Home Page. Beyond the previously mentioned recommendations rooted in article popularity and publication dates, we have three distinct recommendation strategies: User-based, Item-based, and mixed.

1. **User-based:** Our recommendation system adopts a user-centric collaborative filtering method, leveraging collective 'like' patterns to curate suggestions. This encompasses:
   (a) Analyzing users' 'like' actions to decipher preferences.
   (b) Forming a User-Item Matrix where rows represent users, columns denote articles, and matrix entries signify user-article interactions.

(c) Employing cosine similarity to compute the likeness in article preferences and gauge user similarity.

(d) Recognizing 'nearest neighbors' possessing the most parallel interaction patterns.

(e) Proposing articles admired by these neighbors but unseen by the target user, positing that similar users have overlapping article interests.

2. **Item-based:** Alongside the user-centric method, our platform also applies an item-driven recommendation approach. Its workings are as follows:

   (a) Extract the abstracts and keywords from articles a user expressed fondness for.

   (b) Execute a textual similarity analysis on these keywords and abstracts to pinpoint articles with aligned content.

   (c) Advocate for articles that attain the pinnacle of textual closeness, ensuring users get suggestions in line with prior preferences.

3. **Mix:** Our mechanism utilizes a hybrid recommendation modality, merging item-based filtering virtues with popularity indicators. Here's an in-depth breakdown:

   (a) Kick off using the item-centric method to discern potential articles of interest.

   (b) Classify these articles based on click frequency, prioritizing ones with heightened user involvement.

   (c) Fine-tune the recommendations by contrasting the resemblance between these sought-after articles and ones the user previously adored.

   (d) Suggest articles that echo the user's historical tastes and are also in vogue among the wider user community.

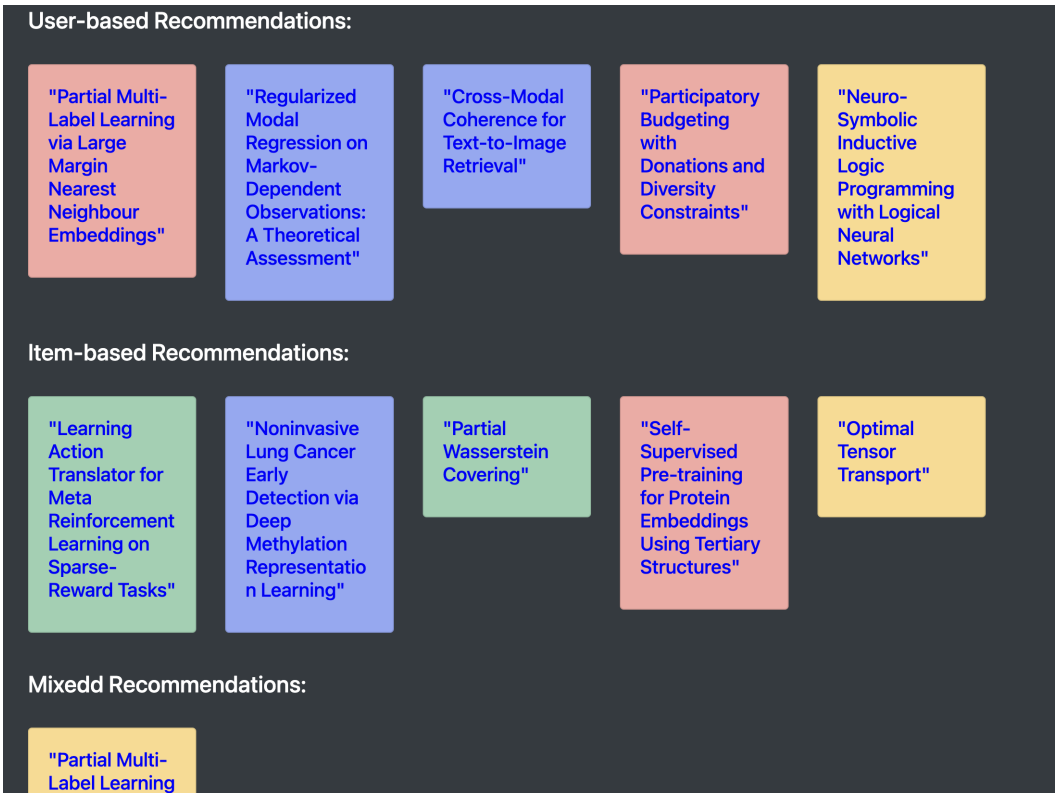


Figure 31: Recommendation

6.4 Evaluation Report

Master of Technology in Intelligent Systems

Project Peer Evaluation Form

Your Name:	Bian Weizhen
Certificate:	Graduate Certificate in Intelligent Reasoning Systems – IRS project

It is compulsory for all students to complete this form and upload (PDF only) into Canvas/LumiNUS. You will get ZERO marks for whole certificate if you do not submit.

Please rate each group member (including yourself) using the following scale in the table below:

1 = poor
2 = needs improvement
3 = satisfactory
4 = very good
5 = excellent

	Scale of 1 (poor) - 5 (excellent)						
Group Members (list by name)	Quality of work	Timelines	Task Support	Responsibility	Involvement	Leadership	Overall Performance
Bian weizhen	5	5	5	5	5	5	5

Definitions

Quality of work:	Consider the degree to which the student provides work that is accurate, complete and whether it meets the required standard
Timeliness	Consider the meeting of deadlines for submission of assigned tasks
Task Support	Consider the amount of support given/volunteered to help other team members
Responsibility	Consider the ability of the student to carry out an assigned task and, the degree to which the student can be relied upon to complete a task.
Involvement	Consider the extent to which the student participates in the exchange of information (does outside research, brings outside knowledge to group).
Leadership	Consider how the student takes leadership of assigned tasks and engages in other leadership activities.
Overall performance	Consider the overall performance of the student team member while in the group.

Additional Comments (optional), e.g. indicate the responsibilities and contributions of each team member here.

NUS National University of Singapore | ISS INSTITUTE OF SYSTEMS SCIENCE

Figure 32: Evaluation Report

6.5 Individual project report

6.5.1 Personal Contribution to Group Project

I took full responsibility throughout the project. I conceived, designed, developed, and evaluated academic search engine systems. My main work includes background research, user analysis, requirements analysis, system design, development to system evaluation, etc. My goal in creating this academic search engine using Java and SpringBoot was to address some of the shortcomings I found in Google Scholar, specifically to cater to prospective Ph.D. students who may need to become more familiar with their academic field of interest or a specific track within it.

6.5.2 What I Found Most Useful

One of the most valuable learning experiences from this project was recognizing the importance of user-centric design and development. By putting myself in the shoes of a Ph.D. aspirant like Tom, who might feel overwhelmed or unsure where to start in the vast academic landscape, I was able to tailor the system's features specifically to this user base. Crafting features such as personalized academic article recommendations, search term optimization with a chatbot, visual representation through word clouds, and a summary tool for abstracts offered practical solutions to the challenges these users face. Through this, I realized that understanding the user's challenges and preferences is crucial in creating an effective and user-friendly platform.

In addition, the biggest gain from this project is the understanding of teamwork. I started working as part of a team during the system development phase. However, I quickly realized that my teammates' abilities did not match the program's requirements, and they seemed unwilling to learn to bridge the gap. Out of commitment to the project's success, I did most of the system development myself. This is a crucial step to ensure that the quality and timeliness of the project are not compromised. I intend to assign tasks based on strengths and weaknesses; I believe that by taking on the burden of system development (which I feel more confident about), I can allow my teammates to focus on areas such as report writing, which I consider to be their strengths. However, this decision met with strong resistance and dissatisfaction from my teammates, and I eventually formed a separate group myself.

Reflecting on this situation, I recognize the challenges of team dynamics and the importance of clear communication. While my motivation stemmed from wanting the best for the project, I could have been more proactive in discussing task assignments with my team and ensuring everyone was on the same page. This could alleviate some misunderstandings and dissatisfaction. However, in the face of adversity and team conflict, I needed to stay committed to the project goals. In future teamwork, I aim to balance ensuring project quality and creating a harmonious team environment through better communication and mutual understanding.

6.5.3 Application of Knowledge and Skills in Other Contexts

The skills and knowledge I gained from this project extend far beyond the scope of academic search engines. The principles of user analysis, requirements analysis, and iterative system development can be universally applied to any software or system development initiative. The ability to solve problems, identify gaps, and create tailored solutions is a valuable skill in any work environment. Additionally, the experience of handling all aspects of a project, from conception to execution, has given me a strong appreciation for the importance of project management and adaptability. These proficiencies will be helpful in future endeavors, whether developing other software systems or leading multi-faceted projects in the workplace.